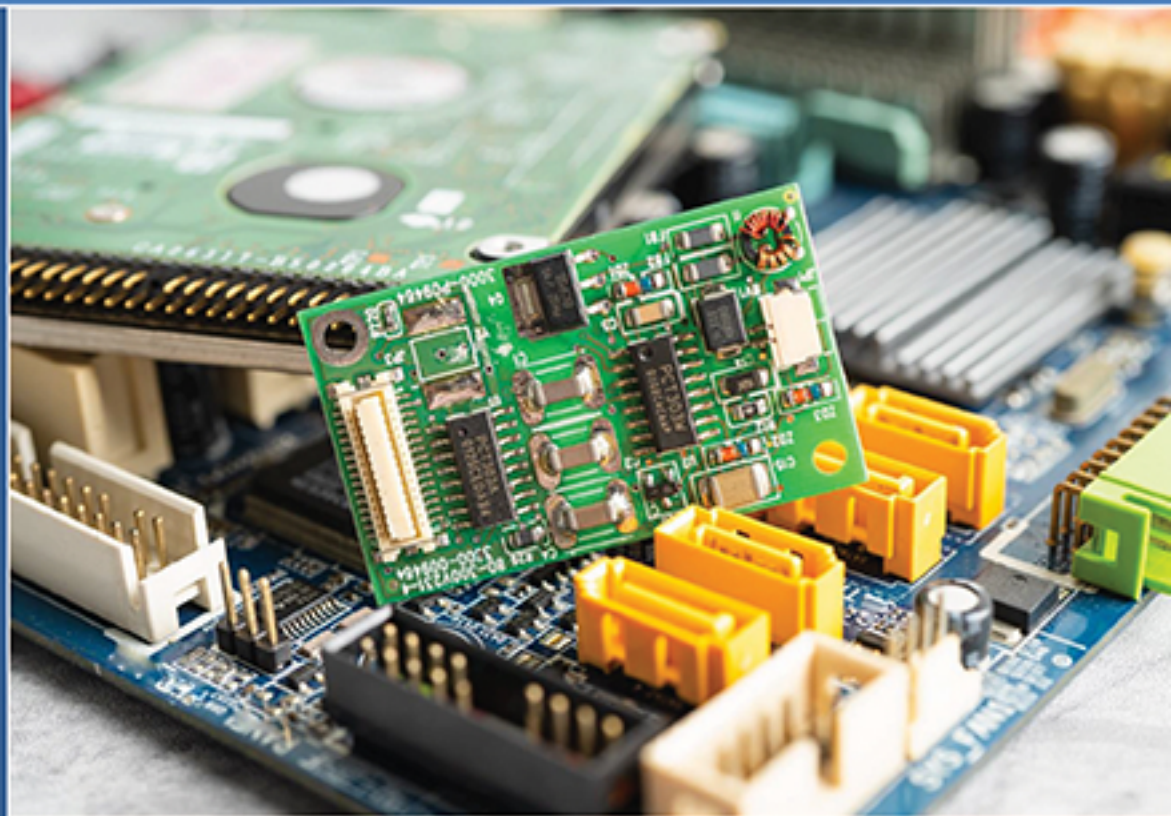


Lubica Markudova

Sistemet kompjuterike



Profesioni elektroteknik

**Profili elektroteknik për
elektronikë dhe
telekomunikacione**

Viti III

Lubica Markudova

Sistemet kompjuterike

Viti III

2025

SISTEMET KOMPJUTERIKE

viti III

Autore:

Lubica Markudova

Recenzentë

Danijela Efnusheva

Poliksena Miteva

Dejan Antonovski

Titulli origjinal:

КОМПЈУТЕРСКИ СИСТЕМИ

III година

Љубица Маркудова

Redaktore: Refail Sulejmani

Përkthyer nga gjuha maqedonase: Riza Etemi

Redaktor profesional i botimit në gjuhën shqipe: Riza Etemi

Lektor: Refail Sulejmani

Botues: Ministria e Arsimit dhe Shkencës e Republikës së Maqedonisë së Veriut
rr. "Shën Kirili dhe Metodi" nr. 54 1000 Shkup

Përpunimi kompjuterik: Lubica Markudova

Rregullimi grafik dhe teknik: Eli Vasilevska Ilievska – ARS STUDIO

Ilustrimi për faqen kryesore: <https://www.vecteezy.com/free-photos>

Vendi dhe viti i botimit: Shkup, 2025

Me Vendimin për miratimin e tekstit shkollor për lëndën Sistemet kompjuterike për vitin III (e tretë), profesioni elektroteknik, profili elektroteknik për elektronikë dhe telekomunikacione, nr. prot. 26-977/1 të datës 11.07.2024, nga Komisioni nacional për tekste shkollore.

Parathënie

Ky tekst shkollor është i destinuar për nxënësit e vitit të tretë në shkollat e mesme profesionale, drejtimi i Elektroteknikës, për profilin arsimor Elektroteknik për elektronikë dhe telekomunikacione. Teksti është përgatitur sipas programit mësimor të dizajnuar në mënyrë modulare dhe është në përputhje me përmbajtjet mësimore të lëndës Sistemet kompjuterike për vitin e tretë. Qëllimi i programit të ri mësimor është modernizimi i mësimdhënies dhe programeve mësimore në arsimin profesional dhe përmbushja e plotë e kërkesave të tregut të punës.

Lënda mësimore Sistemet kompjuterike për vitin e tretë është e pranishme me një fond prej 2 orësh në javë ose 72 orë në vit. Gjysma, pra 36 orë nga fondi i përgjithshëm i orëve, duhet të realizohen si mësim praktik ose ushtrime. Orët për mësim praktik ose ushtrimet realizohen në grupe, në përputhje me Udhëzimin e Ministrisë së arsimit dhe shkencës. Ky koncept ofron mundësinë që njohuritë teorike të fituara të aplikohen praktikisht, duke e bërë mësimin më interesant dhe argëtues. Atraktiviteti dhe interesi që nxit zhvillimi i kompjuterëve dhe programimi i tyre kanë kontribuar shumë në dëshirën e autorit për ta shkruar këtë tekst shkollor. Përmbajtjet e tij ndryshojnë nga përmbajtjet e librave shkollorë klasikë për informatikë, sepse aplikacionet e përpunuara implementohen në sisteme të integruara të programueshme, siç janë mikrokompjuterët e pllakës Arduino Uno dhe Raspberry Pi. Këta mikrokompjuterë, së bashku me kompjuterët personalë, do të jenë të domosdoshëm për realizimin e ushtrimeve.

Teksti përmban katër njësi modulare.

Njësia e parë modulare, Bazat e sistemeve kompjuterike, shpjegon konceptin e një sistemi kompjuterik, organizimin dhe arkitekturën e tij, llojet e ndryshme të sistemeve kompjuterike. Kjo njësi modulare ofron një numër të madh shembujsh për aplikimin e kompjuterëve. Në varësi të qëllimit të vendosur, zgjidhet sistemi kompjuterik dhe softueri i përshtatshëm për përpunimin e aplikacioneve.

Në njësinë e dytë modulare, Hardueri i sistemit kompjuterik personal (PC), nxënësit përforcojnë ose zgjerojnë njohuritë e tyre për karakteristikat dhe funksionin e komponentëve bazë të harduerit. Theksi vihet në montimin e kompjuterit personal dhe zgjedhjen e komponentëve, me qëllim që të arrihet një kompjuter funksional me cilësi të mirë dhe kosto të përballueshme.

Me instalimin, optimizimin dhe riinstalimin e sistemit operativ Windows 10, do të njihemi në njësinë e tretë modulare, Instalimi i sistemit operativ dhe programeve të tjera standarde për sistemin kompjuterik personal. Do të sqarohen disa programe të dobishme dhe programe të shfrytëzuesve. Gjithashtu do të shpjegohen edhe hapat për instalimin e sistemit

operativ Ubuntu 17 Linux në një makinë virtuale, ndërsa qëllimi i kësaj njësie mësimore është pasurimi i njohurive të nxënësve që dëshirojnë të punojnë edhe me sistemin operativ Linux.

Njësia e katërt modulare, Mikrokompjuterët në pllakë, është më e zgjeruara, sepse jo vetëm që studion gjuhët programuese C++ dhe Python, por i përdor ato për programimin e mikrokompjuterëve në pllakë, Arduino Uno dhe Raspberry Pi. Sigurisht, më parë duhet të njihemi me harduerin dhe programet zhvillimore të këtyre mikrokompjuterëve në pllaka.

Në tekstin shkollor, në çdo temë ka pyetje, detyra dhe aktivitete, në mënyrë që të aplikohen njohuritë dhe të kontrollohet arritja e nxënësve.

Autori i këtij teksti shkollor është përpjekur që materialin mësimor ta prezantojë në një mënyrë sa më të thjeshtë dhe të kuptueshme, me shpjegime të sakta të procedurave dhe proceseve dhe me përgatitjen e vizatimeve dhe figurave të qarta. Për përgatitjen e vizatimeve dhe ilustrimeve janë përdorur dy programe: Microsoft Visio dhe Fritzing. Fritzing është aplikacion me burim të hapur, i cili ofron mbështetje të shkëlqyer për përgatitjen e skemave elektrike me platformat Arduino Uno, Raspberry Pi dhe protoplakat. Gjatë programimit të mikrokontrollorit të pllakës mund të përdoren simulatorë kompjuterikë, siç është simuluesi falas Arduino Uno në web-platformën Tinkercad.

Këtë tekst shkollor ua dedikoj nxënësve tanë dhe shpresoj sinqerisht që do t'i ndihmojë ata në arritjen e qëllimeve të tyre profesionale. Sot, më shumë se kurrë, programuesit janë profesioni më i kërkuar, por arrijnë sukses vetëm më këmbëngulësit dhe më të guximshmit.

Ju falënderohem famijes time dhe kolegët për mbështetjen e tyre të pakursyer dhe gjithëpërfshirëse në përgatitjen e këtij teksti shkollor. Gjithashtu, falënderoj komisionin e recensentëve, i cili me sugjerimet e tij të dobishme kontribuoi në realizimin më të mirë të këtij teksti shkollor.

Nga autorja

Përmbajtja

Pjesa teorike

1. Bazat e sistemeve kompjuterike.....	1
1.1. Koncepti dhe ndarja e sistemeve kompjuterike.....	1
1.2. Struktura dhe organizimi i sistemit kompjuterik.....	3
1.2.1. Bllok struktura e sistemit kompjuterik.....	3
1.2.2. Struktura e magjistraleve e sistemit kompjuterik.....	5
1.3. Arkitektura dhe funksionimi i sistemit kompjuterik.....	7
1.4. Sistemet mikrokompjuterike dhe karakteristikat e tyre.....	10
1.5. Ngjashmëritë dhe dallimet në llojet e ndryshme të sistemeve mikrokompjuterike..	12
1.6. Zbatimi i sistemeve mikrokompjuterike.....	15
Përfundime.....	18
Pyetje dhe detyra.....	19
2. Hardueri i sistemit kompjuterik personal (PC).....	21
2.1. Struktura, roli dhe funksioni i komponentëve harduerikë.....	21
2.1.1. Mikroprocesori.....	22
2.1.2. Memoret.....	24
2.1.3. Pllaka Amë.....	27
2.1.4. Pajisja e furnizimit.....	31
2.2. Pajisjet periferike dhe karakteristikat e tyre.....	33
2.3. Zgjedhja e komponentëve harduerike dhe llogaritja e kostos për kohën dhe financat e shpenzuara	38
2.4. Përmirësimi i konfigurimit të kompjuterit.....	41
Përfundime.....	43
Pyetje dhe detyra.....	45
3. Instalimi i sistemit operativ dhe programeve tjera standarde në sistemin kompjuterik personal.....	47
3.1. Llojet e sistemeve operative.....	47
3.2. Veglat softuerike për konfigurimin e sistemit operativ Windows 10.....	50
3.3. Programe të dobishme.....	54
3.3.1. Programet antivirus.....	54
3.3.2. Programet për kompresimin e skedarëve.....	55
3.3.3. Programet për matjen e performancës së kompjuterit.....	57

3.4. Programe aplikative.....	59
Përfundime.....	62
Pyetje dhe detyra.....	64
4. Mikrokompjuteri në pllakë	
4.1. Mikrokompjuteri në pllakë - Arduino.....	67
4.2. Pjesët përbërëse të mikrokompjuterit në pllakë Arduino Uno.....	67
4.3. Elementet periferike dhe komponentët elektronikë për lidhjen me mikrokompjuterin Arduino Uno.....	70
4.4. Shtesa për mikrokontrolluesin Arduino në pllakë.....	73
4.5. Programimi i mikrokompjuterit në pllakë Arduino në gjuhën programuese C/C++	78
4.6. Variablat dhe operatorët në gjuhën programuese C/C++ për mikrokompjuterin Arduino Uno	81
4.7. Instruksionet në gjuhën programuese C/C++ për mikrokompjuterin Arduino Uno.....	82
4.7.1. Instruksionet për punën me pinat hyrje-dalje	86
4.7.2. Instruksionet për komunikim serik	87
4.7.3. Instruksionet për kontrollin e kohës.....	
4.7.4. Instruksionet matematikore.....	91
4.7.5. Instruksionet për Bit dhe Bajt.....	93
4.8. Strukturat në gjuhën programore C/C++ për mikrokompjuterin në pllakë Arduino Uno.....	93
4.8.1. Strukturat për zgjedhjen e mundësive.....	
4.8.2. Strukturat për përsëritje (strukturat ciklike).....	96
4.8.3. Strukturat për degëzim.....	96
4.9. Instruksionet për punën me bibliotekat.....	99
4.10. Mikrokompjuteri në pllakë Raspberry Pi.....	102
4.11. Pjesët përbërëse të mikrokompjuterit në pllakë Raspberry Pi.....	103
4.12. Aksesorët për Raspberry Pi.....	109
4.13. Programimi i mikrokompjuterit Raspberry Pi me gjuhën e programuese Python.....	111
4.14. Lidhja e mikrokompjuterit Raspberry Pi me pajisje hyrëse-dalëse.....	114
4.15. Klasa të pajisjeve hyrëse nga biblioteka GPIOZERO.....	117
4.15.1. Tasteri (angl Button).....	119
4.15.2. Moduli reflektues infrared për monitorim (TRCT5000).....	120
4.15.3. Senzori i distancës (HC-SR04).....	122
4.15.4. Senzori optik ose fotorezistori (angl. LDR-Light Depended Resistor).....	123
4.16. Klasa të pajisjeve dalëse nga biblioteka GPIO Zero.....	124
4.16.1. Dioda LED.....	125
4.16.2. Dioda LED e kontrolluar me impulse të moduluara në gjerësi (PWMLED).....	125
4.16.3. Dioda LED me ngjyrë të ndryshueshme (RGB-LED)	126

4.16.4. Motori i rrymës njëkahëshe.....	127
4.16.5. Servo motori.....	
Përfundime.....	128
Pyetje dhe detyra.....	129
	130
	133

Ushtrime praktike për programin mësimor Sistemet kompjuterike

1. Masat për mbrojtje dhe siguri gjatë punës.....	141
2. Ushtrime praktike për instalimin e harduerit në sistemin kompjuterik personal (PC).....	142
2.1. Masat për mbrojtje dhe siguri gjatë punës me komponentë harduerikë të kompjuterit personal.....	142
2.2. Ushtrim praktik për instalimin e pjesëve përbërëse të kompjuterit personal.....	142
3. Ushtrime praktike për instalimin e sistemit operativ dhe programeve të tjera standarde në një sistem kompjuterik personal.....	159
3.1. Përgatitja për instalimin e softuerit.....	159
3.2. Ushtrim praktik: Instalimi i sistemit operativ Windows 10.....	159
3.3. Ushtrim praktik: Konfigurimi i sistemit operativ Windows 10.....	166
3.4. Riinstalimi i sistemit operativ Windows 10.....	168
3.5. Ushtrim praktik: Instalimi i sistemit operativ Ubuntu 17 Linux në makinë virtuale.....	170
3.6. Ushtrim praktik: Përdorimi i programit antivirus Avira.....	176
3.7. Ushtrim praktik: Instalimi dhe përdorimi i programit për kompresimin e dosjeve.....	178
4. Ushtrime praktike për punën me mikrokompjuterin në pllakë Arduino dhe programimin e tij.....	181
4.1. Masat për mbrojtje dhe siguri gjatë punës me mikrokompjuterin në pllakë Arduino.....	181
4.2. Udhëzime për përdorimin e protoplakës.....	182
4.3. Simulimi kompjuterik për Arduino Uno R3.....	183
4.4. Ushtrim praktik: Instalimi i mjedisit zhvillimor për mikrokompjuterin Arduino dhe vënia në përdorim	186
4.4.1. Udhëzim për instalimin e mjedisit zhvillimor për mikrokompjuterët në pllakë Arduino.....	186
4.4.2. Menyja dhe veglat e mjedisit zhvillimor dhe shkrimi i programi në Arduino Uno R3.....	187

4.5. Ushtrime praktike për programimin e Arduino Uno R3 në gjuhën programuese C/C++.....	191
4.5.1. Ushtrim praktik: Kyçja e diodës led nëpërmjet tasterit.....	191
4.5.2. Ushtrim praktik: Kontrollimi i diodës Led me potenciometër.....	194
4.5.3. Ushtrim praktik: Rregullimi i intensitetit të dritës së diodës LED.....	198
4.5.4. Ushtrim praktik: Fotorezistori për kontrollin e intensitetit të dritës.....	200
4.5.5. Ushtrim praktik: Lidhja e Arduino Uno R3 me servo motor.....	204
4.5.6. Ushtrim praktik: Lidhja e Arduino Uno R3 me ekran LCD.....	207
5. Ushtrime praktike për punë me mikrokompjuterin në pllakë Raspberry Pi dhe programimi i tij.....	211
5.1. Masat për mbrojtje dhe siguri gjatë punës me Raspberry Pi.....	211
5.2. Ushtrim praktik: Instalimi i sistemit operativ Raspbian për Raspberry Pi.....	212
5.2.1. Shkarkimi i softuerit NOOBS për instalim.....	213
5.2.2. Instalimi i sistemit operativ Raspbian.....	213
5.3. Ushtrime praktike për programimin e Raspberry Pi 3B+ në gjuhën programuese Python.....	215
5.3.1. Ushtrim praktik: Kyçja e diodës led me taster.....	216
5.3.2. Ushtrim praktik: Semafori.....	219
5.3.3. Ushtrim praktik: Koha e reagimit.....	221
5.3.4. Ushtrim praktik: kyçja dhe shkyçja e diodës led me fotorezistor.....	224
5.3.5. Ushtrim praktik: Kontrollimi i distancës nga objekti më i afërt.....	226
5.3.6. Ushtrim praktik: Ndryshimi i drejtimin të një motori DC.....	229

Bazat e sistemeve kompjuterike

1.1. Koncepti dhe ndarja e sistemeve kompjuterike

Sot, sistemet kompjuterike përdoren kudo dhe pa to nuk mund të mendohet jeta moderne. Pothuajse nuk ekziston pajisje që nuk përmban një mikrokompjuter në përbërjen e saj. Të tilla janë: pajisjet shtëpiake, pajisjet multimediale dhe pajisjet për argëtim, automatët, pajisjet e komunikimit, aparatet mjekësore, blloqet industriale, mjetet e transportit, pajisjet e zyrave, etj.



Figura 1.1. Ndarja e sistemeve kompjuterike

Me **sistem kompjuterik** nënkuptojmë një sistem të përbërë nga komponentë harduerikë dhe softuerikë, i cili kryen pranimin, përpunimin, ruajtjen dhe shfaqjen e të dhënave dhe informacioneve. Çdo komponent ka funksion të përcaktuar saktësisht. Mikroprocesori, memoriet dhe njësitë hyrëse-dalëse janë komponentë harduerikë. Mikroprocesori kryen përpunimin e të dhënave. Memoriet i ruajnë të dhënat, në mënyrë të përhershme ose të përkohshme. Njësitë hyrëse kryejnë pranimin e të dhënave për përpunim, ndërsa njësitë dalëse i tregojnë informacionet. Që një kompjuter të kryejë një detyrë, shfrytëzuesi duhet t'i japë instruksione

kompjuterit, cilat operacione t'i ekzekutojë dhe me çfarë renditje. Shfrytëzuesi instruksionet i jep nëpërmjet programeve përkatëse.

Në përgjithësi, kompjuterët ndahen **në katër grupe**: mikro, mini, makro dhe superkompjuterë. Kjo ndarje është bërë sipas shpejtësisë së punës, madhësisë, portabilitetit, integritit, kapaciteteve të memories dhe kompleksitetit të operacioneve që mund të kryejnë.

- **Mikrokompjuterët** janë kompjuterë të destinuar vetëm për një përdorues. Në këtë grup përfshihen **kompjuterët personalë**, si: desktopët, laptopët, tabletët, telefonat inteligjentë, etj. Një shembull tjetër i sistemit mikrokompjuterik janë **mikrokompjuterët në pllaka**.

Mikrokompjuterët në pllaka janë mikrokompjuterë të integruar dhe përdoren kryesisht në sistemet e kontrollit automatik. Mikrokompjuterët në pllaka marrin informacione për ambientin që i rrethon përmes sensorëve, të lidhur me pinat hyrëse të tyre. Më pas, këto të dhëna përpunohen në mikroprocesor dhe, pas ekzekutimit të programit, aktivizohen njësitë ekzekutuese të lidhura me pinat dalës të mikrokompjuterit në pllakë (si elektromotorë, pompa, ekrane, etj.). Pjesa më e rëndësishme e mikrokompjuterëve në pllakë janë mikrokontrollorët. Këto janë qarqe të integruara speciale të dizajnuara të cilat në vete përmbajnë njësinë qendrore të procesimit (CPU), RAM dhe ROM. Në vend që secila komponentë harduerikë të jetë çip i veçantë, të gjitha komponentët janë të përfshira në një çip të vetëm.

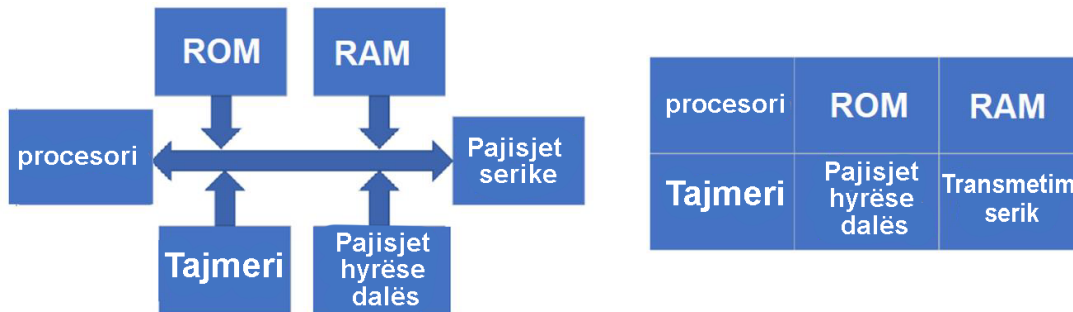


Figura 1.2. Krahasimi mes kompjuterit personal dhe mikrokontrollorit

Figura 1.2 paraqet një krahasim mes kompjuterit personal dhe mikrokontrollorit. Në kompjuterët personalë, secila prej komponentëve harduerike të përmendur është çip i veçantë në pllakën amë dhe për këtë arsye secila paraqitet me një bllok të veçantë, ndërsa shigjetat që lidhin blloqet paraqesin linjat në pllakën amë që shërbejnë për transmetimin e të dhënave. Komponentët harduerike në përbërje të mikrokontrollorit paraqiten me një bllok të përbashkët. Përveç procesorit, memories së përhershme dhe memories së punës, mikrokontrollori në përbërjen e tij mund të përmbajë edhe timer për matjen e kohës, një konvertues analogo-digjital dhe një njësi për komunikimin serik. Për shkak të dimensioneve të tij të vogla dhe kostos së ulët, mikrokontrollori mund të integrohet në shumë pajisje elektronike. Për

dallim nga kompjuterët personalë, mikrokontrollorët kanë frekuencë pune shumë më të vogël, kapacitet të memories më të vogël dhe në rrjet lidhen më vështirë.

- **Minikompjuterët** mund t'i përdorin më tepër përdorues njëkohësisht. Përdoren për aplikacione në kohë reale, si kontrolli i trafikut ose automatizimi i proceseve industriale. Këto janë kompjuterë të fuqishëm me shpejtësi të mëdha, të cilët mund të përmbajnë një ose më shumë mikroprocesorë. Në këtë grup përfshihen serverët, të cilët përdoren për menaxhimin e trafikut telefonik, rrjetave kompjuterike lokale ose globale.
- **Makrokompjuterët** janë të destinuar për aplikacione me shkallë më të lartë ndërveprimi. Një makrokompjuter mund të lidhet me shumë terminale. Përdoren në organizata më të mëdha, si banka ose institucione shtetërore ku është e nevojshme përpunimi dhe ruajtja e të dhënave të miliona përdoruesve.
- Më të fuqishmit dhe më të shtrenjtët janë **superkompjuterët**. Superkompjuterët përdoren për simulim dhe modelimin e fenomeneve komplekse, siç janë reaksionet kimike, shpërthimet bërthamore, parashikimet e motit, kërkimet kozmike, nevojat ushtarake.

Natyrisht, ne nuk jemi në gjendje t'i studiojmë të gjitha këto sisteme kompjuterike. Do t'i studiojmë kompjuterët personalë dhe mikrokontrollerët në pllaka në platformat Arduino dhe Raspberry Pi.

1.2. Struktura dhe organizimi i sistemit kompjuterik

1.2.1. Bllok struktura e sistemit kompjuterik

Tashmë përmendëm se kompjuteri është i përbërë nga hardueri dhe softueri. Në figurën 1.3 është paraqitur bllok struktura e sistemit kompjuterik, komponentët e tij harduerike dhe softuerike, llojet e tyre dhe pjesët përbërëse. [1]

Me harduer nënkuptohen të gjitha pjesët mekanike, fizike nga të cilat është i ndërtuar kompjuteri.

Mikroprocesori i ekzekuton instruksionet dhe i përpunon të dhënat. Pjesët përbërëse të një mikroprocesori janë: regjistrat, njësia aritmetiko-logjike dhe njësia kontrolluese me dekodues. Regjistrat janë lokacione të shpejta të memories brenda mikroprocesorit, të cilat përmbajnë të dhënat dhe kodet për ekzekutimin e instruksioneve. Vetë ekzekutimi ndodh në njësinë aritmetiko-logjike. Njësia

kontrolluese gjeneron sinjale kontrolli në përputhje me kodet e instruksioneve aktuale.

Memoriet i ruajnë programet dhe të dhënat. Memoria RAM është memorie e përkohshme dhe ajo i ruan dokumentet që përpunohen për momentin, ndërsa ROM-i dhe hard disku janë memorie të përhershme.

Magjistralat i transmetojnë të dhënat, dokumentet, programet dhe me to do të njihemi në detaje më vonë.

Pajisje periferike janë: njësitë hyrëse, dalëse dhe hyrëse-dalëse. Për një kompjuter personal, njësitë hyrëse janë: tastiera, miu, skaneri, kamera, lexuesi optik, ndërsa njësitë dalëse janë monitori, printeri, ploteri etj. Njësitë hyrëse shndërrojnë informacionet e përdoruesit në sinjale digjitale, zero dhe njësha, ndërsa pajisjet dalëse bëjnë të kundërtën. Pajisjet hyrëse-dalëse shërbejnë si për futjen, por edhe për nxjerrjen e të dhënave. Njësitë hyrëse-dalëse më të njohura janë modemmat dhe kartelat e zërit.

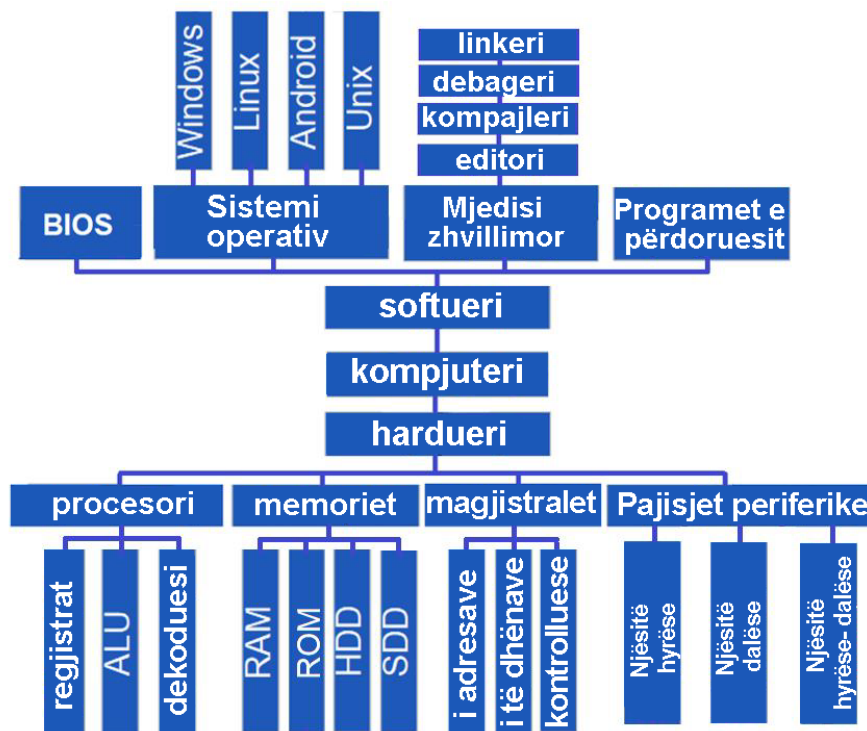


Figura 1.3. Bllok diagram i modelit bazë të kompjuterit

Softueri është grup programesh përmes të cilëve hardueri kryen detyra të caktuara. Softueri mund të ndahet në dy grupe: sistemor dhe shfrytëzues. Sistemi operativ paraqet softuer sistemor që kontrollon dhe koordinon funksionimin e komponentëve harduerikë. Ai është softueri më i afërt me harduerin. Me ndërmjetësimin e sistemit operativ, të gjitha programet e tjera kanë qasje te hardueri. Sistemet operative mundësojnë që resurset e mëdha harduerike të jenë lehtësisht të qasshme për njerëzit dhe të sigurojnë një përdorim më të rehatshëm. Sisteme operative më të përdorura janë Linux dhe Windows. Linux njihet për stabilitetin dhe sigurinë e tij, ndërsa Windowsi për thjeshtësinë në përdorim dhe për qasjen e gjerë

në softuerin aplikativ. Zakonisht, serverët dhe superkompjuterët përdorin Linux, ndërsa Windowsi është sistemi operativ më i popullarizuar për kompjuterët personalë. Microsoft Windows Serveri dhe UNIX/Linux janë sisteme operative të përdorura në rrjeta.

Mjedisi zhvillimor është softuer i cili shërben për krijimin e softuerëve të rinj ose për mbështetje në zhvillimin e programeve aplikative. Për shembull, përpara se një mikrokompjuter pllakë të integrohet në një pajisje elektronike, ai duhet të programohet. Për të shkruar një program aplikativ dhe për ta futur atë në memorien programore, nevojitet një mjedisi zhvillimor i integruar (IDE – Integrated Development Environment). Për të kuptuar më mirë mjedisin zhvillimor, do të sqarojmë disa vegla që janë pjesë e të njëjtës:

- Editorët mundësojnë futjen, ndryshimin, formatimin, ruajtjen dhe printimin e tekstit të programit.
- Përkthyesit i përkthejnë programet e shkruara në gjuhë programuese të nivelit më të lartë në gjuhën e makinës. Përkthyesit njihen edhe me emrin kompajlerë.
- Debagerët janë programe për mënjanimin e gabimeve. Vendosen pika kontrolli në vende të caktuara të programit dhe gjatë ekzekutimit ndalojnë në këto pika, me ç'rast testohet saktësia e programit.
- Linkerët janë programe që lidhin softuerin e ri me atë ekzistues që është shumë me rëndësi për funksionimin e tij.

Programet aplikative ose shfrytëzuese janë programe që mundësojnë kryerjen e detyra konkrete sipas nevojave të përdoruesve. Ato mund të jenë për edukim, punë zyrash, menaxhim biznesi dhe financiar, komunikim, argëtim etj. Në softuerin aplikativ përfshihen programet për përpunimin e tekstit, punën me tabela, menaxhimin e bazave të të dhënave, përpunimin e incizimeve video dhe audio, si dhe përpunimin e fotografive dhe vizatimeve.

1.2.2. Struktura e magjistraleve e sistemit kompjuterik

Magjistralet janë përçues bakri në pllakën amë që shërbejnë për transmetimin e të dhënave. Kompjuterët e parë personal kishin një magjistrale të vetme të quajtur magjistralla sistimore. Kjo magjistrale përbëhej nga 50 deri në 100 tela bakri të integruar në pllakën amë, me konektorë të vendosur në distanca të barabarta për lidhjen e çipave të memories dhe pajisjeve hyrëse-dalëse. Kompjuterët modernë përmbajnë më tepër magjistrale me destinime të specializuara, për shembull, një magjistrale për lidhjen e procesorit me memorien dhe një magjistrale tjetër për lidhjen e procesorit me pajisjet hyrëse-dalëse.

Në Figurën 1.4 është paraqitur se si magjstralet lidhin pjesët e ndryshme të kompjuterit, siç janë mikroprocesori, RAM, ROM dhe disa pajisje periferike.

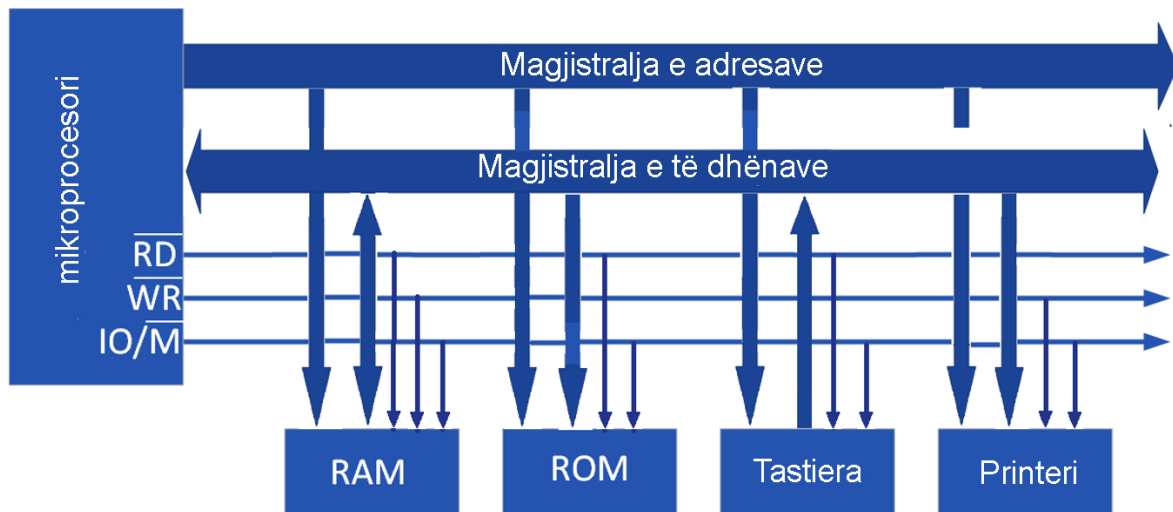


Figura 1.4. Struktura e magjstraleve e sistemit kompjuterik

Që të kryhet transmetimi i të dhënave pa gabime, përdoren rregulla të posaçme të definuara, të cilave duhet t'u përmbahen të gjitha pajisjet në sistemin kompjuterik. Ky grup i rregullave quhet protokoli i magjstrales (ang. bus protocol). **Karakteristika themelore të një magjstrale** janë frekuenca e punës dhe gjerësia. Me gjerësi të magjstrales nënkuptohet se nga sa linja bakri është e përbërë, domethënë sa bitë mund të transmetohen njëkohësisht. Frekuenca e punës është madhësi e rendit të MHz dhe na tregon sa bitë transferohen në një linjë bakri për sekondë. Nëse shumëzohen frekuenca e punës dhe gjerësia e magjstrales, fitohet **brezi i transmetimit** të magjstrales. Njësia matëse për brezin e transmetimit është numri i biteve për sekondë.

Sipas llojit të sinjaleve që transmetohen, magjstralet mund të ndahen në 3 grupe kryesore: linja të adresave (ang. ADDRESS), linja të të dhënave (ang. DATA) dhe linja të kontrollit (ang. CONTROL).

Magjistralla e adresave transmeton bitët e adresave nga mikroprocesori deri te memoria dhe pajisjet periferike. Adresa është kombinim unik i zerove dhe njëshave dhe shërben për njohjen, identifikimin e vetëm një lokacioni të memories ose pajisje periferike. Kur memoria e pranon adresën nga mikroprocesori, ajo tenton që të gjejë lokacionin e dëshiruar të memories.

Linjat e të dhënave shërbejnë për transmetimin e informacionit të përdoruesit nga një pajisje në tjetrën në kompjuter. Magjistralla e të dhënave është dykahëshe. Kur lexohet e dhëna nga memoria, magjistralla e të dhënave transmeton të dhëna nga memoria deri te mikroprocesori. Kur regjistrohet e dhënë në memorie, transmetimi i të dhënave kryhet në drejtimin nga mikroprocesori drejt memories. Magjstralet mund të jenë 8, 16, 32 ose 64-bitëshe. Gjerësia më e madhe do të thotë

shpejtësi më e madh e punës. Kështu për shembull, nëse duhet të transmetohet një e dhënë 32-bitëshe përmes një magjistrale 8-bitëshe, atëherë janë të nevojshme 4 cikle për transmetimin e të dhënave.

Linjat e kontrollit ose komandimit janë më të shumta dhe për secilën linjë të komandimit ekziston shenjë speciale, e cila, në fakt, është shkurtësë e fjalës në anglisht për sinjalin kontrolluese përkatës. Linjat e kontrollit shërbejnë për definimin e operacionin që duhet të ekzekutohet. Disa bitë të kontrollit transmetohen nga mikroprocesori deri te pajisjet tjera dhe përfaqësojnë një lloj urdhëri. Bitët të tjerë kontrolli transmetohen nga pajisjet te mikroprocesori dhe paraqesin një lloj informacioni kthyes se si janë zbatuar urdhrat e mëparshëm të dhënë. Nëse do të shkruhet apo lexohet një e dhënë nga memoria, vendosin dy linjat e kontrollit të magjistraleve të kontrollit: \overline{RD} dhe \overline{WR} . Këto janë shkurtime nga fjalët READ (lexo) dhe WRITE (shkruaj). Negacioni mbi to tregon se janë aktive në nivelin logjik zero. Nuk guxojnë që të dyja linjat të jenë aktive njëkohësisht (ose do të lexojmë, ose do të shkruajmë). Një linjë kontrolli tjetër e rëndësishme është linja me shenjën IO/\overline{M} . Kjo shenjë është shkurtësë për Input Output/Memory, që në përkthim do të thotë pajisje hyrëse-dalëse/memorie. Kur sinjali $IO/\overline{M}=1$, atëherë mikroprocesori komunikon me ndonjë njësi hyrëse-dalëse (pajisje periferike), ndërsa kur $IO/\overline{M}=0$, mikroprocesori komunikon me memorien.

Magjistralet janë të integruara në vetë pllakën amë të kompjuterit dhe numri i tyre është i pandryshueshëm. Rritja e numrit të linjave për çdo magjistrale shkakton një rritje të konsiderueshme të kostos të kompjuterit. Duhet të bëhet një kompromis mes çmimit dhe gjerësisë së magjistraleve. Për shembull, sa më shumë linja adresash të kemi në dispozicion, aq më shumë hapësirë memorie mundet drejtpërdrejtë të adresohet. Nëse magjistralla ka n linja adresash, atëherë mund të adresohen gjithsej **2ⁿ lokacione memoriesh**. Me çdo gjeneratë të re kompjuterësh, numri i linjave të magjistraleve rritet, si të adresave ashtu edhe të të dhënave dhe të kontrollit.

1.3. Arkitektura dhe funksionimi i sistemit kompjuterik

Bloku skema që është paraqitur në figurën 1.3 i jep të gjitha komponentët më të rëndësishme harduerike dhe softuerike në përbërjen e një kompjuteri. Struktura e magjistraleve e paraqitur në figurën 1.4 e tregon lidhjen mes komponentëve të ndryshme dhe transferin e të dhënave. Megjithatë, bllok struktura dhe ajo e magjistraleve nuk janë të mjaftueshme që të kuptohet se si funksionojnë në të vërtetë kompjuterët. Arkitektura kompjuterike është komplekse dhe kërkohet shumë punë dhe kohë për analizën e saj.

Arkitektura bazë e mikrokompjuterit është treguar në figurën 1.5. Për të dhënë një përshkrim funksional, është e nevojshme që të futemi në brendësinë e mikroprocesorit dhe memorien e tij të punës, RAM. Kur do të hapim një program, i njëjti transferohet nga memoria e përhershme në memorien RAM dhe në **numëruesin e programit** regjistrohet adresa fillestare e programit. Nga figura 1.5 mund të shihet se programi tashmë ndodhet në RAM. Çdo lokacion në RAM ka numër të vetin rendor, i cili quhet **adresë**. Pa adresën, mikroprocesori nuk mund të qaset te lokacioni dhe të lexojë përmbajtjen e tij. Kur do të filloj ekzekutimi i programit, mikroprocesori e merr adresën nga numëruesi i programit dhe e dërgon në RAM. Ai kërkohet, gjen lokacionin dhe përmbajtja e tij transferohet deri te mikroprocesori përmes magjstrales së të dhënave. Nëse përmbajtja e lokacionit është instruksion, atëherë ajo vendoset në **regjistrin e instruksioneve**, prej ku dërgohet në njësinë e kontrollit.

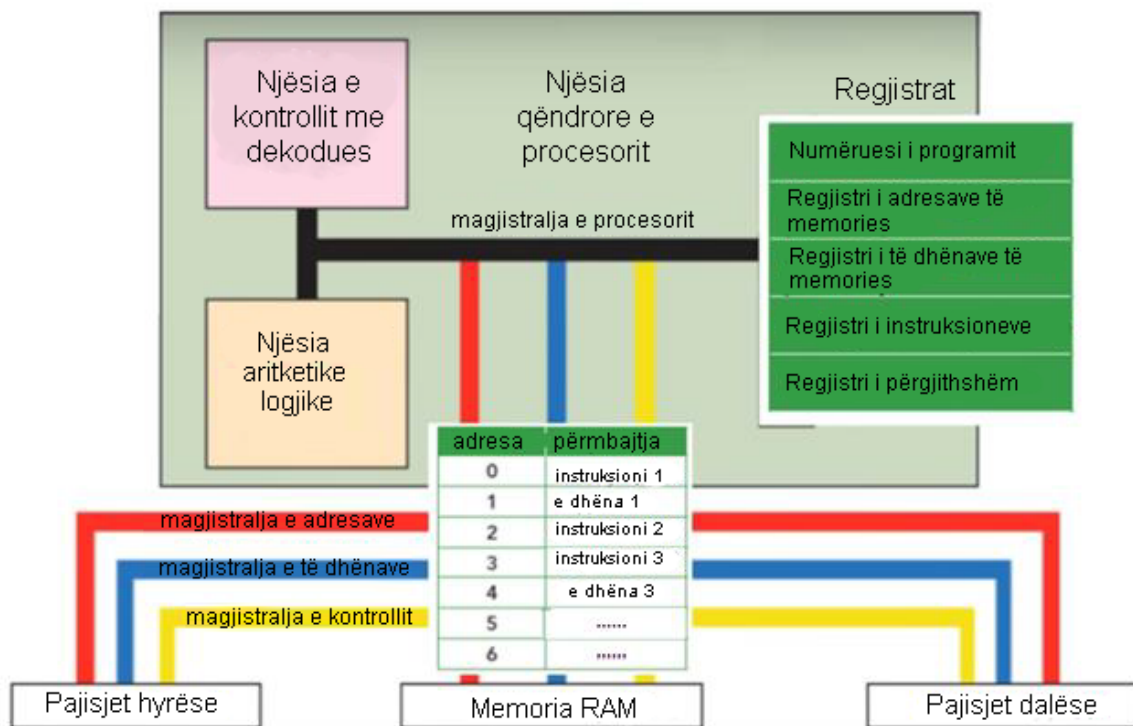


Figura 1.5. Organizimi dhe përshkrimi funksional i kompjuterit

Në **njësinë e kontrollit** ndodh magjia. Softueri e aktivizon harduerin, gjegjësisht instruksioni i koduar përcakton se cilët pina do të aktivizohen ose nuk do të aktivizohen, cilat resurse harduerike do të kyçen ose jo. Instruksioni i koduar, në fakt, zgjedh të ashtuquajturin mikroinstruksion. Numri i mikroinstruksioneve varet nga numri i instruksioneve të mundshme që mikroprocesori mund të ekzekutojë. Vetë mikroinstruksioni përbëhet nga mijëra **bitë kontrolli**, zero dhe njësha. Për shembull, zero mund të realizojë shkyçjen, ndërsa njëshi mund të realizojë kyçjen e ndonjë komponenti. Bitët e kontrollit

dërgohen përmes magjstrales së kontrollit. Në figurën 1.5. tregohet që pas instruksioneve 1 dhe 3 pasojnë të dhënat 1 dhe 3, ndërsa për ekzekutimin e instruksionit 2 nuk kërkohet ndonjë e dhënë. Kur instruksioni 1 do të transferohet nga memoria RAM në mikroprocesor, numëruesi i programit rritet me një, gjegjësisht ai do të përmbajë adresën e të dhënës 1. Pasi të dekodohet instruksioni 1, mikroprocesori përsëri dërgon adresën që ndodhet në numëruesin e programit deri në memorien RAM, me qëllim që të marrë të dhënë nr. 1. Kur e dhëna numër 1 mbërrin në mikroprocesor, ajo vendoset në regjistrin e përgjithshëm dhe nga aty kalon në **njësine aritmetiko-logjike** e cila përfundimisht e ekzekuton instruksionin. Nëse rezultati i fituar duhet të dërgohet në memorien RAM, atëherë rezultati vendoset në regjistrin e të dhënave të memories, ndërsa adresa e lokacionit ku duhet ruajtur e dhëna vendoset në regjistrin e adresës së memories. Pasi të ruhet rezultati i fituar, mikroprocesori kalon te instruksioni nr. 2, përmes adresës që ndodhet në numëruesin e programit.

Cikli **transfero-dekodo-ekzekuto** përsëritet për çdo instruksion deri sa mikroprocesori të arrijë fundin e programit. Gjatë ekzekutimit të cikleve transfero-dekodo-ekzekuto, shumë me rëndësi janë koha dhe përshtatja (sinkronizimi) e shpejtësisë së komponentëve harduerik. Timing nënkupton ndarjen e kohës dhe definimin e aktiviteteve që duhet të ekzekutohen, për secilin komponent veçmas. Tajmingu i punës së mikroprocesorit shpjegohet përmes diagrameve kohore, të cilat tregojnë se cilët pina kyçen, ndërsa cilët shkyçen në çdo nanosekundë të kohës.

Programi futen në memorien e kompjuterit përmes **pajisjeve hyrëse**. Për shembull, programi shkruhet përmes tastierës dhe editorit të tekstit. Automatikisht ruhet në RAM -memorien që përfundimisht të ruhet në memorien e përhershme, p.sh. hard disk. Shpesh here të dhënat që janë të nevojshme për ekzekutimin e programit nuk janë pjesë e tij, por futen përmes ndonjë pajisjeje hyrëse. Rezultatet e fituara nga ekzekutimi i programit, shfrytëzuesi i ndjek përmes **pajisjeve dalëse**. Për kompjuterin, informacione të dobishme janë kodet – kombinime të zerove dhe njëshave. Për shembull, një karakter (numër, shkronjë ose simbol alfanumerik) për kompjuterin përfaqëson informacion prej një bajti ose një piksel i ekranit është informacion prej tre bajtave.

Arkitektura e mikroprocesorit është komplekse dhe shpeshherë abstrakte. Komponentët harduerike formojnë një tërësi funksionale, sistem efikasiteti funksional i të cilit varet nga shumë faktorë: hierarkia, lidhja, komunikimi, protokollet, tajmingu, sinjalizimi, sinkronizimi, definimi i rrymave dhe tensjoneve hyrëse dhe dalëse për pinat e qarqeve të integruara, etj. Njohja e këtyre koncepteve mund të ndihmojë shumë gjatë programimit dhe përdorimit të mikrokompjuterëve në procesin e komandimit.

1.4. Sistemet mikrokompjuterike dhe karakteristikat e tyre

Nga sistemet kompjuterike, mikrokompjuterët janë me performance më të dobëta dhe me madhësi më të vogla, por për njeriun janë të një rëndësie më të madhe sepse pa to shfrytëzuesit fundor nuk mund të shfrytëzojnë beneficinet e kompjuterizimit dhe digjitalizimit global. Ndarja themelore e sistemeve mikrokompjuterike është në kompjuterë personalë dhe kompjuterë në pllakë (angl. SBC-Single Board Computer). Në grupin e kompjuterëve personalë përfshihen: desktopat, laptopat, tabletat dhe telefonat celular. Të gjithë këta mikrokompjuterë të numëruar janë të destinuar për përdorim nga një përdorues.



Figura 1.6. Llojet e mikrokompjuterëve personalë

Karakteristikat kryesore të kompjuterëve personalë janë: madhësia dhe transportueshmëria, konfigurimi dhe performancat, kostoja, mundësitë e lidhjes, përditësimi, softueri i disponueshëm dhe të tjera. **Madhësia dhe transportueshmëria** janë karakteristika të ndërlidhura. Nëse kompjuteri është me përmasa dhe peshë të madhe, ai nuk mund të transportohet dhe zakonisht vendoset në një vend dhe nuk zhvendoset. Gjatë zgjedhjes së vendit të punës duhet pasur kujdes që kompjuteri të jetë i mbrojtur nga papastërtitë dhe lagështia dhe të ketë hapësirë të mjaftueshme për ventilim të duhur. Fjala **konfigurim** nënkupton përzgjedhjen e komponentëve harduerikë që do të jenë kompatibil dhe do të formojnë një tërësi funksionale. Përdoruesit zakonisht kërkojnë informacione për tipin dhe karakteristikat e mikroprocesorit, llojin dhe kapacitetin e memories së përkohshme dhe asaj të përhershme, shpejtësinë e punës dhe memorien e kartelës grafike, rezolucionin e ekranit, etj. Nga konfigurimi varen **performancat** e kompjuterit, siç janë shpejtësia, besueshmëria, lloji i softuerit që mund ta ekzekutojë, qëndrueshmëria dhe jetëgjatësia e tij. Nga konfigurimi varet edhe kostoja e tij. Ndërfaqja -interfejsi nënkupton mënyrën e lidhjes së kompjuterit me pajisje të tjera elektronike, siç janë sistemet muzikore, video sistemet, televizorët, rrjetat kompjuterike, rrjetat e internetit, pajisje elektronike të ndryshme, etj. Për një lidhje të suksesshme të kompjuterit, është e nevojshme ai të disponojë me konektorët e duhur dhe softuer për komunikim.

Përditësimi varet para së gjithash nga mundësitë e pllakës amë. Për shembull, mund të përmirësohet RAM-memoria, të instalohen dy memorie të

përhershme (hard-disk dhe SSD), të zëvendësohet kartela e rrjetit me një të re dhe më të shpejtë, etj.



Figura 1.7. Mundësitë e lidhjes së kompjuterit

Përparimi i shpejtë i teknologjisë e ka ndryshuar ose më saktë, ka zgjeruar kuptimin e nocionit mikrokompjuter. Më herët, ky nocion ishte ngushtë i lidhur me nocionin kompjuteri personal. Por sot, kjo ka ndryshuar dukshëm, kryesisht për shkak të paraqitjes dhe zhvillimit të sistemeve të integruara (angl. Embedded System). Sistemet e integruara mund të jenë të bazuar në mikrokontrollues ose në platforma si Arduino dhe Raspberry Pi. Mikrokompjuterët në pllakë janë të bazë për sistemet e integruara.



Figura 1.8. Sisteme mikrokompjuterike të integruara

Krijimi i sistemeve të integruara të bazuar në mikrokontrollues mund të ndahet në pjesën harduerike dhe pjesën softuerike. Së pari duhet të përpunohet pjesa harduerike, gjegjësisht elektronika e pajisjes. Sipas destinimit të pajisjes dhe funksionalitetit përzgjidhen komponentët elektronikë: rezistorë, kondensatorë, dioda, qarqe të integruara, sensorë, njësi ekzekutuese, furnizimi me energji, lloji i mikrokontrolluesit dhe bazamenti i përshtatshëm për të. Më pas, përpunohet pllaka me qarqe të shtypura, e cila lidh komponentët elektronikë në një tërësi funksionale. Pjesa harduerike përfundon pasi komponentët të lidhen në

pllakë. Pastaj fillon pjesa softuerike. Që programi i shfrytëzuesit të implementuar në mikrokontrollues, nevojitet një pajisje e veçantë e quajtur programator i cili lidhet me kompjuterin përmes një kabloja USB. Mikrokontrolluesi vendoset në bazamentin e programatorit dhe pasi të programohet, transferohet në bazamentin e pajisjes së përdoruesit.

Krahas sistemeve integruese të bazuara të mikrokontrolluesve, gjithnjë e më shpesh përdoren edhe sistemet e bazuara në Arduino dhe Raspberry Pi. Arduino paraqet platformë zhvillimore e bazuar në mikrokontrollues me përmasat e një kartele kreditore. Platformat Arduino paraqesin pllaka elektronike të cilat përveç mikrokontrolluesit përmbajnë edhe një programator serik, oscilator kristalor dhe hyrje/dalje analoge dhe digjitale. Programuesi nuk ka nevojë të shqetësohet për zgjedhjen e programatorit, ndërsa pinat e integruar në hyrje-dalje lejojnë lidhje të thjeshtë me sensorët dhe pajisjet ekzekutuese. Për ndërtimin e pajisjes elektronike, të cilën duam ta komandojmë, në vend të pllakës së shtypur mund të përdoret një protoplakë eksperimentale (angl. Protoboard), e cila na lejon të eksperimentojmë me dizajnin e pajisjes.

Edhe pse për nga madhësia janë pothuajse të njëjtë me Arduino, Raspberry Pi ka një procesor shumë më të fuqishëm dhe kapacitet më të madh të memories RAM. Në vend të mikrokontrolluesit ai përmban një çip të veçantë me procesor dhe kartelë grafike të integruar dhe memoria RAM është çip i veçantë në pllakën elektronike. Ndryshe nga Arduino, Raspberry Pi ka sistemin e vetë operativ të integruar.

Ne do të studiojmë platformën Arduino dhe Raspberry Pi dhe do të njihemi me pjesët përbërëse të tyre, funksionalitetin, lidhjen me pajisjet hyrëse dhe dalëse dhe mënyrën e programimit të tyre.

1.5. Ngjashmëritë dhe dallimet në llojet e ndryshme të sistemeve mikrokompjuterike

Shkurtimisht do të sqarojmë karakteristikat e secilit lloj të mikrokompjuterit.

Desktopi është lloji më i vjetër i kompjuterëve personalë. Mangësi kryesore e tij është se nuk mund të transportohet. Megjithatë, për shkak të madhësisë së tij, desktopi ka shumë avantazhe. Për një çmim më të vogël fitojmë kompjuter me performancat dhe cilësitë e dëshiruara. Përdoruesi mund të zgjedhë vetë konfigurimin e desktopit të tij. Ka mundësi për përmirësim dhe është shumë e lehtë të zëvendësohen komponentë të caktuar në rast defekti. Desktopi ka të gjitha llojet e portave kompjuterike (VGA, LAN, HDMI, USB, slotet e brendshme PCI) dhe përfaqëson kompjuterin me mundësitë më të shumta për t'u lidhur me pajisje të tjera elektronike. Konsumi i energjisë është më i madh në krahasim me laptopin, por ftohja dhe ventilimi i çipave është në një nivel më të lartë, çka bën që jetëgjatësia e tij të jetë më e madhe.

Përparësia kryesore e **laptopit** është portabiliteti i tij. Kjo do të thotë se gjatë përdorimit të laptopit, përdoruesi nuk është i kufizuar nga hapësira gjatë përdorimit, por vetëm nga koha, pasi laptopi ka një bateri me kohë të kufizuar përdorimi. Pothuajse për çdo konfigurim të desktopit ekziston edhe konfigurim i ngjashëm i laptopit. Disavantazhi i laptopit është konfigurimi i gatshëm, domethënë blerësit e zakonshëm nuk mund të zgjedhin vetë komponentët e harduerit sipas nevojave të tyre. Kuptohet, konfigurimi sipas porosisë është i mundshëm, por kjo e rrit shumë çmimin, i cili edhe ashtu është i lartë. Po ashtu, edhe servisimi i laptopit është më i komplikuar dhe kushton më shumë krahasuar me desktopin. Në tabelën 1.1 është bërë një krahasim mes laptopit Dell 12.5" Latitude 7290 dhe mikro desktopit Dell OptiPlex 5070. Edhe pse kanë performanca të ngjashme, laptopi është dy herë më i shtrenjtë se desktopi.

	Dell 12.5" Latitude 7290	Dell OptiPlex 5070
Procesori	1,7 GHz Intel Core i5-8350U Quad-Core	2 GHz Intel Core i7-9700T 8-Core
Shpejtësia maksimale	3,6 GHz	4,3 GHz
RAM-i instaluar	8 GB	8 GB
Kapaciteti maksimal i memories	16 GB	32 GB
Kartela grafike	Intel UHD Graphics 620	Intel UHD Graphics 630
Memoria SSD	1 x 256 GB	1 x 256 GB
Portat	2 x USB Type-A 1 x HDMI 1.4	5 x USB Type-A 2 x DisplayPort 1.2
Altoparlant dhe mikrofon i integruara	√	×
Web kamerë e integruara	√	×
Konsumi i energjisë	65 W	90 W
Pesha	1,4 Kg	5,8 kg

Tabela 1.1. Krahasimi i mikrokompjuterit desktop dhe laptop

Tableti ka shumë pak mundësi për lidhje me pajisje të tjera. Zakonisht, tabletët kanë vetëm një port mini USB- konektor. Për lidhje me internetin ekziston vetëm një lidhje Wi-Fi dhe nuk mund të lidhet në rrjetin kompjuterik lokal. Dimensionet e vogla, transportimi i lehtë i tij dhe konsumi i vogël i energjisë janë përparësitë kryesore të tabletave. Për shkak të ekranit me prekje, ka mundësi të përdorimit të stilolapsit inteligjent, gjegjësisht përdoruesi mundet me dorë të vizatojë ose shkruaj në ekran.

Sot thuajse nuk ka njeri në botën moderne i cili nuk shfrytëzon **telefon inetelegjent**. Përdorimi kryesor i këtyre pajisjeve është për shërbimet e telefonisë celulare, por njëkohësisht ato përdoren për kryerjen e shumë aplikacioneve të tjera: si kërkimi në internet, prodhimi dhe riprodhimi audio dhe video, ndërsa shumë programe të tjera të dobishme. Shpejtësia e vogël e punës, kapacitetet e kufizuara të memories, ekranit i vogël i bëjnë telefonat celular të papërshtatshëm për punë

profesionale. Ata mund të ekzekutojnë shumë aplikacione, por janë të papërshtatshëm për programim dhe mundësitë e tyre për krijimin e softuerit të ri janë të pakrahasueshme me ato të desktopëve dhe laptopëve.

Mikrokompjuterët në pllakë janë bërë të disponueshëm për përdoruesit e zakonshëm gjatë pesëmbëdhjetë viteve të fundit. Më parë, ata përdoreshin në industri dhe sistemet për komandim automatik. Në vitin 2005 u lëshua platforma harduerike **Arduino** me softuer të hapur, me qëllim që të thjeshtohet procesi i inovacioneve elektronike prototipike. Kjo u mundësoi njerëzve të zakonshëm, edhe atyre me pak ose pa paranjohuri teknike të krijojnë produkte interaktive. Raspberry Pi janë konkurrentët më të mëdhenj të Arduino-s. Gjenerata e parë e Raspberry Pi doli në treg në vitin 2012 në Angli. Dëshmi e popullaritetit të tij është se në gjashtë muajt e parë u shitën rreth gjysmë milioni copë.

	Raspberry Pi	Arduino
përparësitë	Procesor 64-bitësh me frekuencë pune 1.5GHz dhe mundësi për ekzekutimin e disa detyrave njëkohësisht	Lidhje e lehtë me sensorë analogë dhe kontroll i saktë i punës së motorëve
	Portë rrjeti e integruar, mundësi lidhjeje pa tela përmes Wi-Fi ose Bluetooth	Zgjedhje e madhe e shtesave për të rritur funksionalitetin
	Sistemi operativ funksional	Programet i ekzekuton menjehë pas kyçjes së furnizimit pa përshtatje shtesë
	Përmban dalje audio, lidhje për kamerë, dalje HDMI dhe disa porta USB	Kostoja e ulët e prodhimit
	Zgjedhje e shkëlqyer për projekte si robotë dhe video-lojëra	Ekzekuton aplikacione të thjeshta në kohë reale
	Mund të programohet në më tepër gjuhë programuese	Harduer dhe softuer më i thjeshtë
mangësitë	Për lidhjen e komponentëve elektronike nevojitet softuer shtesë	Frekuencë pune prej 16 MHz, shpejtësi e ulët e punës dhe ekzekuton vetëm një program
	Kostoja e lartë e prodhimit	Pa shtesë Ethernet nuk mund të lidhet me rrjetin e internetit
	Ndrohje më e madhe gjatë përdorimit	Për programim përdoret vetëm gjuha programuese C++

Tabela 1.2. Krahasimi mes Arduino dhe Raspberry Pi

Sistemet mikrokompjuterike të integruara janë shumë të popullarizuara për shkak të çmimit të ulët, fleksibilitetit dhe lehtësisë së programimit. Është shumë e rëndësishme që të zgjidhet në mënyrë të mençur platforma zhvillimore për realizimin e ushtrimit praktik të dhënë ose detyre projektuese. Në këtë mund të ndihmojë tabela

1.2, në të cilën në mënyrë krahasuese janë dhënë avantazhet dhe disavantazhet e Arduino dhe Raspberry Pi. Raspberry Pi është orientuar nga softueri, që do të thotë se detyrat kryhen kryesisht duke përdorur programe, ndërsa Arduino është orientuar nga hardueri. Për shembull, nëse dëshirojmë të bëjmë disa foto, t'i përpunojmë ato dhe t'i publikojmë në një faqe interneti, atëherë Raspberry Pi ofron zgjidhje më të thjeshta. Por, nëse dëshirojmë të kontrollojmë shpejtësinë e rrotullimit të një motori me rrymë njëkahëshe, atëherë platforma Arduino Uno është më e përshtatshme për shkak të shpejtësisë më të lartë të reagimit dhe saktësisë më të madhe. Arduino dhe Raspberry Pi janë pjesë e platformës Interneti i gjërave (angl. IoT - Internet of Things). Kjo platformë është tendencë e re në teknologji dhe merret me lidhjen e sensorëve dhe pajisjeve, lidhjen në rrjet të tyre duke përdorur protokolle të ndryshme komunikimi dhe analizimin e të dhënave. Kjo platformë nuk është objekt i analizës tonë, por lidhja e tij me Arduino dhe Raspberry Pi është e pashmangshme.

1.6. Zbatimi i sistemeve mikrokompjuterike

Gjatë përzgjedhjes së kompjuterit personal, është shumë e rëndësishme të dihet **qëllimi** i tij. Aplikacionet për përpunimin e tekstit, përpunimin e zërit ose videos, krijimi i softuerëve aplikativë, dizajni grafik, lloje të ndryshme të lidhjes në rrjet dhe komunikimeve, automatizimi etj., kërkojnë kompjuterë me performanca të ndryshme.

Zgjedhja e konfigurimeve është e madhe, nga procesorët Pentium deri te Intel i9, RAM nga 4GB deri në 128GB, memorie të përhershme HDD ose SDD nga 500GB deri në 20TB, kartela grafike nga NVIDIA, Intel, AMD. Për shkak të zhvillimit të shpejtë të kompjuterëve, këto të dhëna vazhdimisht ndryshojnë dhe rriten.

Vetë kompjuterët klasifikohen në disa kategori: për përdorim shtëpiak dhe për studentë, kompjuterë biznesi, kompjuterë për profesionistë dhe kompjuterë për video lojëra. **Kompjuterët për përdorim shtëpiak** rekomandohen për përdoruesit që zakonisht përdorin aplikacionet si Microsoft Office ose për kërkime në internet. **Kompjuterët profesional** janë të destinuar për inxhinierë, arkitektë, dizajnerë. Janë të shkëlqyer për animacione dhe simulime. Këta kompjuterë përdoren gjithashtu si serverë për lidhjen në rrjet të më tepër kompjuterëve personal në kompani të vogla. **Dizajni grafik dhe lojërat video** kërkojnë procesorë të shpejtë, kartela grafike shumë të fuqishme dhe memorie me kapacitet të lartë. Rekomandohet që procesori të jetë me gjashtë bërthama me frekuencë minimale të punës prej 3GHz, memoria RAM të ketë kapacitet prej 16GB, kartela grafike të ketë të paktën 2GB memorie video RAM dhe në vend të hard diskut të përdoret memoria SSD. Çdo piksel i ekranit zë memorie prej 3 bajtësh, nga një bajt për çdo ngjyrë bazë RGB (red-green-blue). Rezolucioni HD

(high definition) do të thotë 1920×1080 piksela, ose përafërsisht dy milionë piksela. Frekuenca e punës prej 60 Hz do të thotë rifreskim i imazhit 60 herë në sekondë. Nëse shumëzohet këto shifra, do të marrim një brez lëshimi prej 360MB/s. Këto shifra shpjegojnë çmimin e lartë të kompjuterëve për lojëra video dhe për përpunimin e materialeve video. Këto kompjuterë janë me dizajn të mirë, efekte drite dhe ndriçimi, me më tepër pjesë metalike sesa pjesë plastike dhe janë të ndërtuar nga materiale me cilësi më të lartë, kështu që duhet të kenë jetëgjatësi më të gjatë.

Ekzistojnë një numër i pafundëm i shembujve të përdorimit të mikrokompjuterëve. Do të përmendim disa prej tyre:

- **Mjekësi.** Në fillim, kompjuterët u përdorën për ruajtjen dhe përpunimin e të dhënave mjekësore për shkak të qartësisë dhe qasjes më të mirë. Në këtë kategori bie edhe telemjekësia, e cila mundëson kujdesin ndaj pacientëve nga distanca. Sot, nuk mund të imagjinohet diagnostikimi dhe zbulimi i sëmundjeve pa zbatimin e tomografisë kompjuterike, rezonancës magnetike dhe rrezeve rentgen 3D. Kirurgët gjithnjë e më shumë përdorin modele virtuale për simulimin e procedurave kirurgjikale. Vendosja e çipave në trupin e njeriut tashmë është procedurë e thjeshtë. Për shembull, pacemaker është çip i cili gjeneron impulse elektrike me amplitudë dhe frekuencë të përcaktuar saktë dhe simulon funksionimin e zemrës. Ky çip ka furnizimin e vet dhe mund të përdoret gjithashtu për monitorimin e punës së zemrës.
- **Industria automobilistike.** Në një makinë moderne ka mbi 70 njësi kontrolli dhe 100 sensorë. Nga inovacionet, 80% mbështeten në përdorimin e mikrokompjuterëve në pllakë. Mikrokontrolluesi më i rëndësishëm është njësia elektronike e kontrollit (ECU - Electronic Control Unit). Madhësitë hyrëse kryesore janë fuqia e motorit dhe numri i rrotullimeve. Madhësi tjera hyrëse janë: temperatura dhe rrjedhën e ajrit të thithur, temperatura e lëngut për ftohje, pozicioni i boshtit etj. Madhësi dalëse janë sasia e karburantit që injektohet në një cikël dhe momenti i ndezjes. Njësia kontrolluese përcakton se sa kohë injektorët do të mbeten të hapur (afërsisht 6 deri 9 milisekonda, nga 600 deri 3000 herë në minutë), me çka kontrollohet sasia e karburantit. Kohëzgjatja e ndezjes së karburantit varet nga ajo se sa shpesh ndizen dhe fiken kandelat në motor. Të dhënat e këtyre parametrave ruhen në të ashtuquajturën matricë, në bazë të të cilave fitohen koeficientët dalës. Matrica, në fakt është memorie EEPROM dhe ajo është e ndryshme për lloje të ndryshme të automobilave. Pas llogaritjes, mikrokompjuteri në pllakë transferon vlerat e fituara në aktuator – injektorë. Nëse temperatura është më e lartë se e parashikuara, koha e hapjes së injektorëve zgjatet (rrjedhë më e madhe e ajrit).
- **Automatizimi i industrisë.** Komandimi automatik është proces kompleks dhe mund të ndahet në disa nivele. Niveli i parë përbëhet nga sensorët dhe njësitë ekzekutuese që ndjekin dhe ndryshojnë parametrat e prodhimit. Niveli i dytë janë kontrolluesit logjikë të programueshëm, të cilët në përputhje me programin,

veprojnë menjëherë, pa analizuar. Niveli i tretë i supervizionit i përpunon të dhënat, përpunon modele matematikore, studion makineritë dhe bën optimizimin, përgatit raporte. Kërkohej një kompjuter i shpejtë, i cili do të përpunojë shpejt të dhënat, një memorie e madhe e përhërshme për ruajtjen e vlerave të kaluara dhe grafikë të dukshme për monitorim më të lehtë të parametrave.

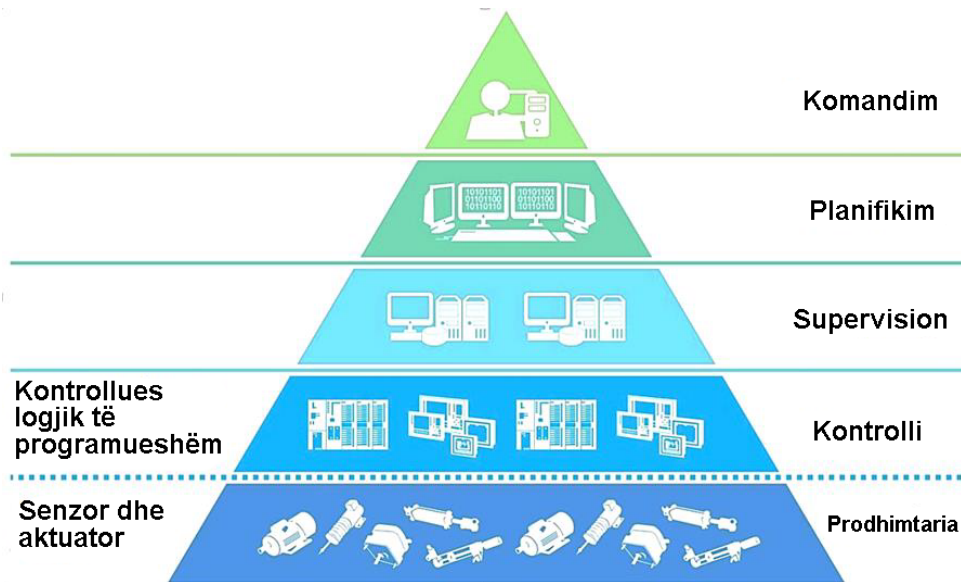


Figura 1.9. Zbatimi i mikrokompjuterëve në automatikë

- **Banka dhe financa.** Shumë institucione financiare po investojnë kapital në teknologjitë digjitale. Transaksionet bankare realizohen përmes platformave mobile dhe web, pavarësisht në cilin vend qëndroni. Kompjuterët përdoren për përpunimin e raporteve statistikore, ndjekjen e trendeve globale dhe për modernizim të vazhdueshëm të sistemit bankar elektronik. Këto tendenca përfshijnë realizimin e transaksioneve me telefon celular në vend të kartelave për pagesë dhe përdorimin e autorizimit biometrik për më shumë siguri dhe mbrojtje nga keqpërdorimi (përdorimi i shenjës së gishtit ose njohja kompjuterike e fytyrës).

Çdo fushë ka karakteristikat e veta dhe kërkon specializim për t'u bërë ekspert kompjuteri në atë fushë. Ne do të njihemi me bazat e harduerit dhe softuerit të kompjuterëve personal dhe mikrokompjuterëve në pllaka, që është shumë e rëndësishme për zhvillimin e mëtejshëm profesional, sipas dëshirave dhe mundësive të individit.

Përfundime

Me **sistem kompjuterik** nënkuptojmë sistem të përbërë nga komponentë harduerikë dhe softuerike, i cili kryen marrjen, përpunimin, ruajtjen dhe paraqitjen e të dhënave dhe informacioneve.

Në përgjithësi, kompjuterët ndahen në katër grupe: **mikro, mini, makro dhe superkompjuterë**. Mikrokompjuterët janë kompjuterë të destinuar vetëm për një përdorues. Minikompjuterët mund të përdoren nga më tepër përdorues njëkohësisht. Makrokompjuterët janë të dedikuar për aplikacione me shkallë më të lartë interaktiviteti.

Komponentë më të rëndësishme harduerike janë: mikroprocesori, memoriet, magjistralet dhe pajisjet periferike. Mikroprocesori përpunon të dhënat, memoriet i ruajnë ato, magjistralet i transmetojnë dhe pajisjet periferike kryejnë konvertimin në sinjale për përdoruesit.

Komponentë më të rëndësishme softuerike janë: sistemi operativ, programet zhvillimore dhe programet aplikative.

Struktura e magjistrave tregon se si janë të lidhura komponentët harduerike dhe si kryhet transmetimi i të dhënave.

Si funksionon mikrokompjuteri mund të shpjegohet përmes ciklit **transmeto-dekodo-ekzekuto**. Instruksionet dhe të dhënat transferohen nga RAM-i në mikroprocesor. Me dekodimin krijohen bitë kontrollues që aktivizojnë dhe çaktivizojnë harduerin. Ekzekutimi i instruksioneve kryhet në njësinë aritmetiko-logjike.

Karakteristika themelore të mikrokompjuterëve janë: madhësia dhe portabiliteti, konfigurimi dhe performancat, kostoja e prodhimit, mundësitë për lidhje, mundësia për t'u integruar, përditësimi, programimi dhe softueri i disponueshëm, etj.

Përparësia e desktopit është çmimi i tij, mundësia për përditësimin dhe mirëmbajtja e lehtë. **Përparësia e laptopit** është portabiliteti i tij.

Sistemet mikrokompjuterike të integruar janë shumë popullore për shkak të kostonë së ulët të prodhimit, fleksibilitetit dhe programimit të thjeshtë.

Mikrokontrolluesi **Raspberry Pi** ka sistemin e tij operativ, kartelën grafike dhe mundësinë për lidhje me rrjetin.

Sipas destinimit, kompjuterët personalë ndahen në kompjuterë për përdorim shtëpiak dhe kompjuterë profesionalë.

Pyetje dhe detyra

1. Çfarë nënkuptojmë me termin "sistem mikrokompjuterik"?

2. Cili është dallimi thelbësor mes një mikrokompjuteri dhe një mikrokontrolluesi?
3. Numëro disa shembuj të zbatimit të minikompjuterit!

4. Numëro blloqet strukturore kryesore të sistemit kompjuterik!

5. Cili është funksioni i sistemit operativ?

6. Çfarë lloj të magjistraleve ekzistojnë dhe çfarë bitësh transmetojnë ata?

7. Shpjego ciklin transmeto-dekodo-ekzekuto!

8. Pse është e nevojshme adresimi i lokacioneve të memories RAM?

9. Cila është funksioni i njësisë kontrolluese (Control Unit)?

10. Numëro karakteristikat kryesore të mikrokompjuterit!

11. Cilat janë avantazhet dhe disavantazhet e laptopit krahasuar me llojet tjera të kompjuterëve personal?

12. Cilat janë avantazhet dhe disavantazhet e mikrokompjuterit në pllakë Arduino Uno në raport me mikrokompjuterin Raspberry Pi?

13. Shpjego funksionin dhe rëndësinë e mikrokompjuterit ECU (Electronic Control Unit) të integruar në makinat moderne!

14. Cilat elemente duhet të përfshijë konfigurimi i një kompjuteri personal?

15. Numëro disa shembuj praktikë për mikrokompjuterë të integruar!

2. Hardueri i sistemit kompjuterik personal (PC)

2.1. Struktura, roli dhe funksioni i komponentëve harduerikë

Në krahasim me të kaluarën, sot zgjedhja e komponentëve harduerikë dhe montimi i kompjuterëve desktop është proces i thjeshtë. Për të krijuar një kompjuter personal standard janë të nevojshme komponentët bazë të mëposhtme: shtëpiza, pajisja për furnizim me energji, pllaka amë, mikroprocesori, ftohësi për procesorin, memoria RAM, hard disku ose memoria SSD, kartela grafike (e cila mund të jetë e integruar në vetë procesorin), kartela e zërit, kartela e rrjeti, pajisja CD ROM, tastiera, miu, monitori, kabllo për lidhje dhe foletë për vidat (figura 2.1).



Figura 2.1. Komponentët harduerike të kompjuteri personal desktop

Ne do të njihemi me funksionin, pjesët përbërëse, karakteristikat dhe mënyrën e lidhjes së të gjitha këtyre komponentëve harduerikë. Më vonë do të njihemi me llojet e sistemeve operative, instalimin e tyre dhe shkurtimisht me disa prej programeve më të përdorura nga përdoruesit. Këto paranjohuri janë të domosdoshme nëse dëshirojmë të ndërtojmë me sukses një kompjuter personal.

2.1.1. Mikroprocesori

Mikroprocesori është "truri" i kompjuterit. Ai ka dy funksione kryesore: ekzekuton programet dhe menaxhon me pajisjet tjera në pllakën amë. Për mikroprocesorin, programet janë një grup instruksionesh, të renditura në lokacione të njëpasnjëshme të memories. Mikroprocesori i **ekzekuton programet** instruksion pas instruksioni. Pasi të përpunojë instruksionin momental, e merr instruksionin e ardhshëm nga memoria RAM, e përpunon, ndërsa ky proces përsëritet derisa procesori të arrijë në fund të programit. **Funksioni menaxhues** e bën mikroprocesorin zotëri të pllakës amë (angl. master). Ai mbledh informacion nga të gjitha pajisjet në kompjuter, i përpunon, i proceson dhe më pas nxit aktivitete të caktuara, me qëllim të sigurojë funksionim të besueshëm të të gjithë kompjuterit. Në figurën 2.2 paraqitet organizimi bazë i mikroprocesorit. **Pjesë përbërëse** të një mikroprocesori janë: regjistrat, njësia aritmetiko-logjike dhe njësia menaxhuese me dekoderin. Regjistrat janë lokacione të shpejta të memories në brendësi të mikroprocesorit të cilat i përmbajnë të dhënat dhe kodin operacional të instruksionit momental. Regjistrat e përgjithshëm përmbajnë të dhënat për përpunim dhe të dhënat e përpunuara, ndërsa kodet e nevojshme për ekzekutimin e instruksionit momental vendosen në një regjistër specifik – regjistrin e instruksioneve.

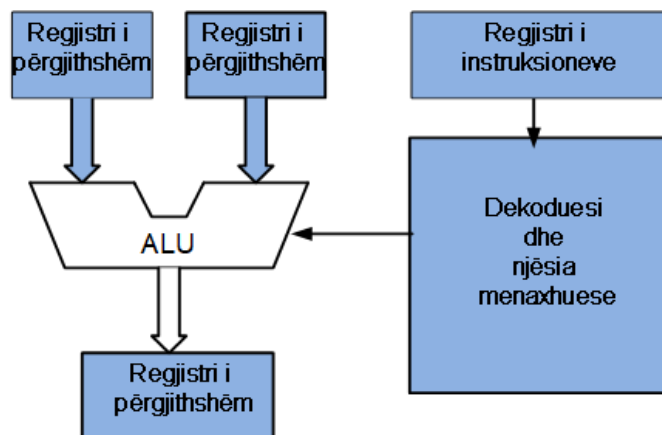


Figura 2.2. Bllok-shema e modelit bazë të mikroprocesorit

Në njësinë aritmetiko-logjike (ALU) ekzekutohen të gjitha instruksionet aritmetike dhe logjike. Njësia menaxhuese merr informacion nga të gjitha pajisjet dhe më pas vendos, ndryshon përmbajtjen e regjistrave, ndryshon gjendjen logjike të pinave, thirr programe, aktivizon pajisjet periferike etj. Zerot dhe njëshat që dalin nga njësia menaxhuese aktivizojnë ose çaktivizojnë komponentët harduerike. Numri dhe funksioni i bitëve kontrollues duhet të planifikohet gjatë fazës së dizajnit të mikroprocesorit.

Karakteristika më të rëndësishme të mikroprocesorit janë: Frekuenca e punës, gjerësia e të dhënave, koncepti i rrjedhës, kompleksiteti i grupit të

instruksioneve, numri i bërthamave, renditja e pinave, etj.

Frekuenca e punës është frekuenca e sinjalit të taktit. Taktet janë të rëndësishëm sepse kohëzgjatja e çdo operacioni matet me numër të plotë të takteve të sinjaleve. Për shembull, që të lexohet një bajt nga memoria RAM, janë të nevojshme tre takte. Në taktin e parë, mikroprocesori dërgon adresën, në të dytin kryhet kërkimi në memorie dhe në të tretin mikroprocesori merr të dhënën e dëshiruar. Ajo që për ne është sekondë, për mikroprocesorin është nanosekondë (e një-miliardta pjesë e sekondës).

Në varësi të gjerësisë së të dhënave (angl. data width), mikroprocesorët mund të jenë 8, 16, 32, 64 dhe 128-bitësh. Për shembull, në procesorët 64-bitësh ekzistojnë 64 pina për të dhënat, në regjistrat e përgjithshëm mund të vendosen 64 bitë, njësia aritmetiko-logjike kryen operacione me të dhëna prej 64-bitëve dhe në të njëjtën kohë magjistralla e të dhënave e pllakës amë përmban 64 linja. Të theksojmë se me multipleksimin e gjerësisë së të dhënave multipleksohen edhe resurset harduerike të procesorëve.

Procesorët mund të kenë grup instruksionesh komplekse ose të reduktuara. Sot, pothuajse të gjithë procesorët janë RISC (angl. Reduced Instruction Set Computer) dhe këta procesorë kryejnë instruksione të thjeshta dhe janë shumë të shpejtë.

Procesorët shumëbërthamësh nënkuptojnë integrim të më shumë procesorëve në të njëjtin qark të integruar. Thënë më thjeshtë, dy koka mendojnë më mirë se një, ndërsa katër duar janë më të vlefshme se dy. Dy bërthamat i ndajnë aplikacionet dhe kështu shkurtohet koha e ekzekutimit.

Pinat janë zgjatime metalike të linjave të brendshme të qarqeve të integruara. Gjeneratat më të reja të procesorëve kanë rreth 2000 pina me një diametër prej 0.3 mm. Në procesorët më të vjetër, pinat ndodheshin në procesor, ndërsa hapjet për to ndodheshin te bazamentin e pllakës amë. Ky koncept është i njohur me shkurtesën **PGA** (angl. Pin Grid Array). Koncepti kundërt është **LGA** (angl. Land Grid Array), ku hapjet ndodhen në procesor, ndërsa pinat ndodhen në bazament.

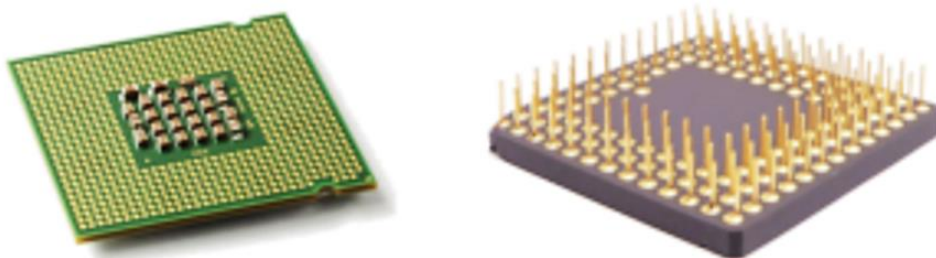


Figura 2.3. Renditja e pinave të procesorëve

Prodhuar më të mëdhenj të procesorëve janë kompanitë Intel dhe AMD. Procesorët e AMD janë shumë të popullarizuar mes adhuruesve të lojërave kompjuterike. Për një çmim më të ulët, ato ofrojnë performancë të mirë, kryesisht falë kartelës grafike të integruar në të njëjtin çip së bashku me mikroprocesorin. Procesorët e Intelit janë përpunuar me një teknologji më të avancuar, prandaj janë më rezistentë ndaj nxehtësisë dhe kanë një jetëgjatësi më të madhe.

2.1.2. Memoret

Të gjithë të dhënat dhe programet që i ekzekuton procesori duhet të jenë të vendosura në memorie. **Karakteristikat më të rëndësishme të memorieve** janë kapaciteti dhe shpejtësia e tyre. Kapaciteti i memorieve matet në bajta (B) ose njësi më të mëdha për kapacitet janë: kilobajti (KB), megabajti (MB), gigabajti (GB), terabajti (TB). Shpejtësia e memories matet përmes kohës së qasjes ose frekuencës së punës së saj. Koha e qasjes është koha që i duhet memories për të kërkuar dhe gjetur të dhënën e dëshiruar. Frekuenca e punës së memories na tregon se sa bitë mund të pranojë ose dërgojë memoria në një sekondë.

Kompjuterët përmbajnë lloje të ndryshme të memorieve. Organizimi i memorieve nënkupton ndarjen e funksioneve për çdo lloj memorieje dhe mënyrën e qasjes në të dhënën e dëshiruar. Në figurën 2.4 është paraqitur piramida e memorieve. Ashtu siç zbresim poshtë, kapaciteti i memorieve rritet, ndërsa zvogëlohet shpejtësia e tyre e punës. Shpejtësia më e madhe e punës sjell një rritje të çmimit për një bajt. Sa më afër të jenë memoret me majën e piramidës, aq më afër mikroprocesorit janë.

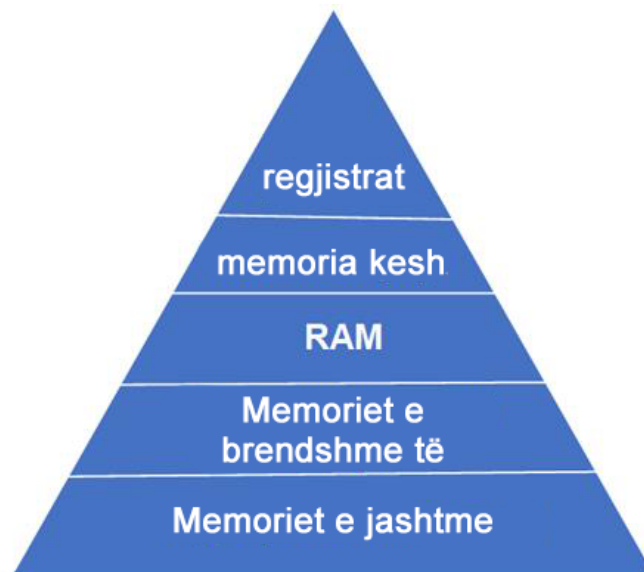


Figura 2.4. Piramida e memorieve

Regjistrat shërbejnë për ruajtjen e përkohshme të të dhënave, para se ato të përpunohen, por edhe për vendosjen e rezultateve menjëherë pas marrjes së tyre. Regjistrat janë lokacionet të memorieve më të shpejta në kompjuter, por kapaciteti i tyre është vetëm disa bajta.

Mer qëllim që të zvogëlohet koha e pritjes dhe të rritet shpejtësia e procesorit, përdoret memoria kesh (cache). Memoria kesh është memorie RAM shumë e shpejtë. Të dhënat e nevojshme nga memoret më të ngadalta dërgohen paraprakisht dhe vendosen në memorien kesh, e cila paraqet një lloj “dhomë pritjeje”

për mikroprocesorin. Në memorien kesh mund të ruhen të dhëna që mikroprocesori i përdor më shpesh, por që nuk mund t'i vendos në regjistrat e tij. Memoria kesh mund të jetë brenda vetë çipit të memories ose jashtë tij.

RAM është shkurtesë nga fjalët në anglisht Random Access Memory, që në përkthim do të thotë memorie me qasje të rastësishme. Kjo do të thotë që procesori ka qasje të barabartë në të gjitha lokacionet e memories, ose asnjë lokacion nuk është me prioritet më të ulët ose më të lartë. RAM është memorie nga e cila njëkohësisht mund të lexohen, por edhe të regjistrohen të dhëna të reja. Ajo është memorie me përmbajtje të ndryshueshme dhe paraqet një lloj **memorie pune**, një lloj listë pune në të cilën mikroprocesori i regjistron përkohësisht të gjitha të dhënat që do t'i nevojiten për realizim të suksesshëm të programit të dhënë. Në memorien RAM ruhen programet që momentalisht ekzekutohen në procesor. Memoria RAM është e njohur edhe si memoria primare, për shkak të rëndësisë së saj. Memoria RAM është **memorie e përkohshme**. Nëse shkyçet furnizimi, të dhënat nga RAM nuk mund të rinovohen kur kompjuteri kyçet përsëri. Për këtë arsye, para se të shkyçim kompjuterin, të dhënat nga RAM-i duhet të vendosen në hard disku, përmes shtypjes (angl. click) të ikonës **SAVE**. Kjo është paraqitur në figurën 2.5.

Memoria RAM e furnizon procesorin me informacione, ndërsa ai merr informacione nga ndonjë memorie e përhershme, siç është hard disku. Programet ruhen në memorien e përhershme dhe ato transferohen në RAM kur duhet të përpunohen. Nëse memoria RAM është më e madhe, atëherë ajo do të marrë një pjesë më të madhe të programit aplikativ dhe ai do të përpunohet në mikroprocesor. Kjo redukton numrin e transferimeve nga RAM në memorien e përhershme, e kjo do të thotë kursim në kohë. Mikroprocesori i ekzekuton programet **instruksion pas instruksioni**. Pasi të përfundojë me instruksionin momental, thirret memoria RAM për të gjetur instruksionin e ardhshëm. Nëse bëhet fjalë për program linear, atëherë instruksionet janë të renditura në lokacione të njëpasnjëshme (rradhë) të memories RAM, gjë që nuk ndodh nëse bëhet fjalë për një program të degëzuar.



Figura 2.5. Funksioni i memories RAM si memorie pune të mikroprocesorit

Memoria RAM është memorie primare, ndërsa hard disku dhe memori të tjerë të jashtëme janë memorie sekondare. Për dallim nga RAM-i, memoria kesh dhe regjistrat, memoria sekondare është **memorie e përhershme**. Memorie të përhershme janë **hard disku**, **memoria SSD** dhe memori të jashtëme (kompakt disqet, memori USB, kartelat SD etj.). Hard disku është memorie magnetike dhe përbëhet prej disa pllakave magnetike që rrotullohen, ndërsa midis këtyre pllakave vendoset një krehër me koka për lexim. Gjatë zgjedhjes së hard diskut, karakteristika

më të rëndësishme janë kapaciteti dhe shpejtësia e transferimit të të dhënave. Sot në treg ofrohen disqe të forta me kapacitet nga 500MB deri në 18TB, por këto të dhëna vazhdimisht ndryshojnë. Shpejtësia e transferimit të të dhënave është një nga 70Mb/s deri në 150Mb/s. Ajo varet nga faktorë të ndryshëm siç janë: shpejtësia e rrotullimit, dendësia e të dhënave, madhësia e kesh memorisë së saj dhe përmasat fizike. Shpejtësia e rrotullimit mund të jetë 5400, 7200, 10000 por ka edhe disqe me 15000 rrotullime në një minutë. Dendësia e të dhënave matet në numrin e bitëve për inç katror dhe ajo rritet me rritjen e kapacitetit të hard diskut. Mund të ndodhë që hard disku me shpejtësi më të vogël të rrotullimit dhe kapacitet më të madh të ketë shpejtësi më të madhe të transferimit të të dhënave krahasuar me një disk me shpejtësi më të lartë rrotullimi dhe kapacitet më të vogël. Për të rritur shpejtësinë e transferimit, hard disqet e sotshëm përmbajnë edhe memorie kesh dhe ata janë të njohur me emrin hard disqe hibride. Kapaciteti i kësaj memorie kesh është një nga 32MB deri në 256MB. Sa i përket përmasave fizike, hard disqet mund të jenë 2.5 dhe 3.5 inç. Zakonisht, disqet më të mëdha janë për kompjuterët desktop, ndërsa ato më të vegjël për laptopë.

Sot, hard disqet kanë filluar të zëvendësohen me memorie të reja SSD (angl. Solid State Drive). Ato janë memorie elektronike, diçka e ngjashme me memorien RAM, por me furnizim vetiak të energji për ruajtjen e përhershme të të dhënave. Për kompjuterët personalë, zakonisht zgjidhen pajisje memoruese SSD me shpejtësi transferimi të të dhënave nga 200MB/s deri në 500MB/s, por ka edhe të tilla me shpejtësi mbi 3000MB/s. Sot ofrohen pajisje SSD me kapacitet nga 128GB deri në 4TB. Tabela 2.1 paraqet krahasimin midis pajisjeve SSD dhe hard disqeve.

Hard diku	Disku SSD
Pajisje memoruese me disk të fortë (ang. HDD=Hard Disk Drive)	Pajisje me memorie të përhershme me gjysmëpërçues (ang. SSD=Solid State Drive)
Memorie magnetike	Memorie elektronike
Kapacitet maksimal 15TB	Kapacitet maksimal 1TB
Paraqitet ngrohja	Nuk ka ngrohje
Memorie e lire	Memorie e shtrenjtë
Shpejtësi të vogël	Shpejtësi të madhe
Jetëgjatësi më të vogël	Jetëgjatësi më të madhe

Tabela 2.1. Krahasimi i pajisjeve memorues HDD dhe SSD

Konfiguracionet kompjuterike më të reja për kompjuterë personal janë me hard disqe me kapacitet prej 1TB ose 2TB dhe pajisje memoruese SSD me kapacitet 256GB ose 512GB. Memoria SSD përdoret për ruajtjen dhe nisjen e sistemit operative çka rritet ndjeshëm shpejtësia e funksionimit të kompjuterit. Ndërsa, në hard disk ruhen skedarët më të mëdhenj të përdoruesit.

2.1.3. Pllaka amë

Pllaka amë i **lidh** të gjitha komponentët harduerikë në një tërësi funksionale. Komponentët me njëra tjetrën janë të lidhura fizikisht me linja bakri të quajtura magjistrale. Sot, pllakat amë prodhohen me tetë deri në dhjetë shtresa linjash prej bakri. ATX (angl. Advanced Technology eXtended) është teknologji e avancuar, me zgjerim, në dizajnimin e pllakave amë. Ekzistojnë tri madhësi kryesore të pllakave amë: ATX (305 mm × 244 mm), Micro ATX (244 mm × 244 mm) dhe Mini ITX (170 mm × 170 mm). Pllakat amë me madhësi më të mëdha kanë numër më të madh të slot-eve dhe kapacitet më të madh për memorien RAM. Përveç linjave prej bakri, pllaka amë përmban një numër të madh të konektorëve të ndryshëm dhe çipave që kontrollojnë transferimin e të dhënave nga një pajisje në tjetrën. Shkurtimisht do të njihemi me secilin prej tyre.

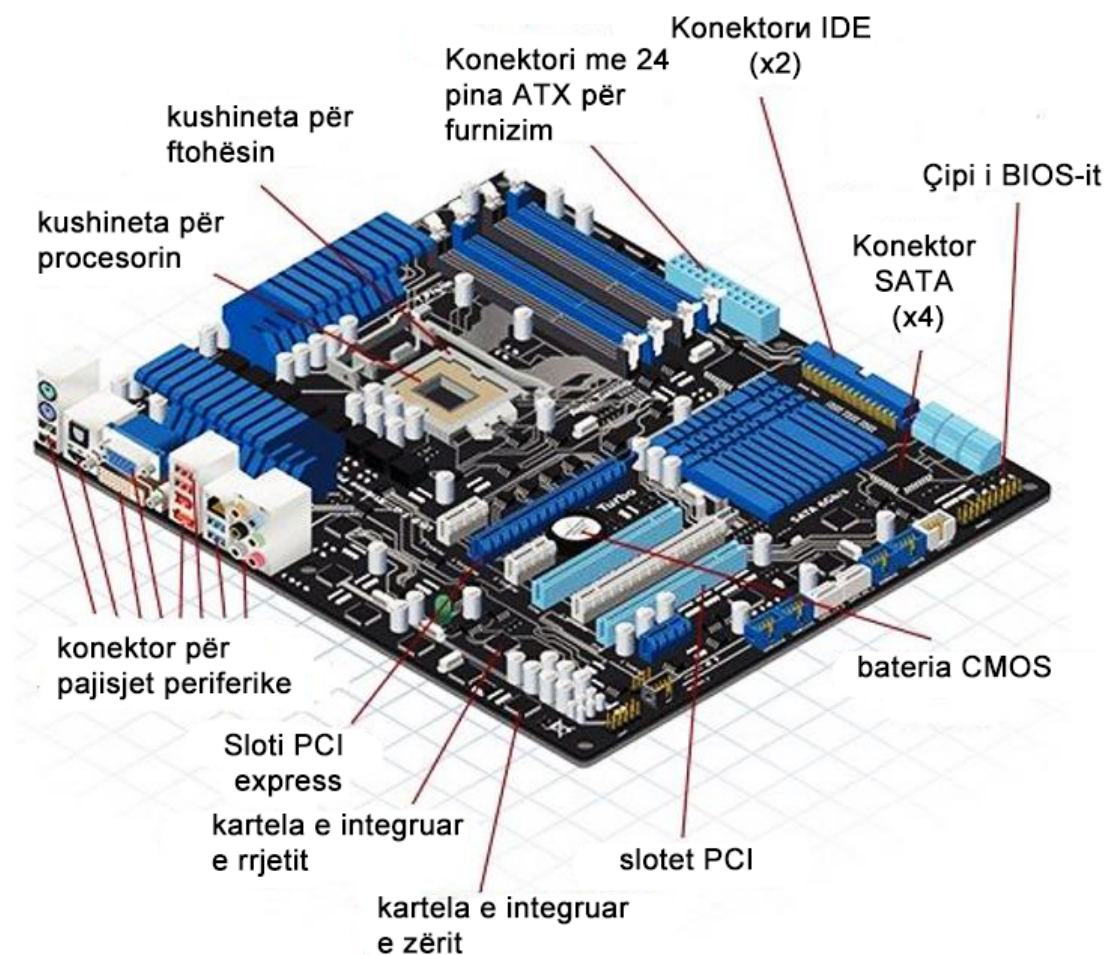


Figura 2.6. Pjesët përbërëse të pllakës amë

Procesori → Zgjedhja e pllakës amë varet nga zgjedhja e procesorit, pasi secili lloj i procesorit ka konfigurim unik të pinave, që kërkon një kushinetë adekuate.

Çipseti → Çipseti është **grup i qarqeve të integruar të specializuar të cilët kontrollojnë rrjedhën e të dhënave**. Madje edhe vetë pllaka amë emërohet sipas llojit të çipsetit të integruar në të. Çdo çipset është i përbërë prej **dy komponentëve kryesore**: ura veriore dhe jugore. Ura verior është kontrollues i grafikës dhe memories, ajo është shumë e shpejtë dhe është e lidhur drejtpërdrejt me mikroprocesorin. Ura jugor është kontrollues i hyrje-daljeve dhe ai komunikon me mikroprocesorin, me ndërmjetësimin e urës veriore.

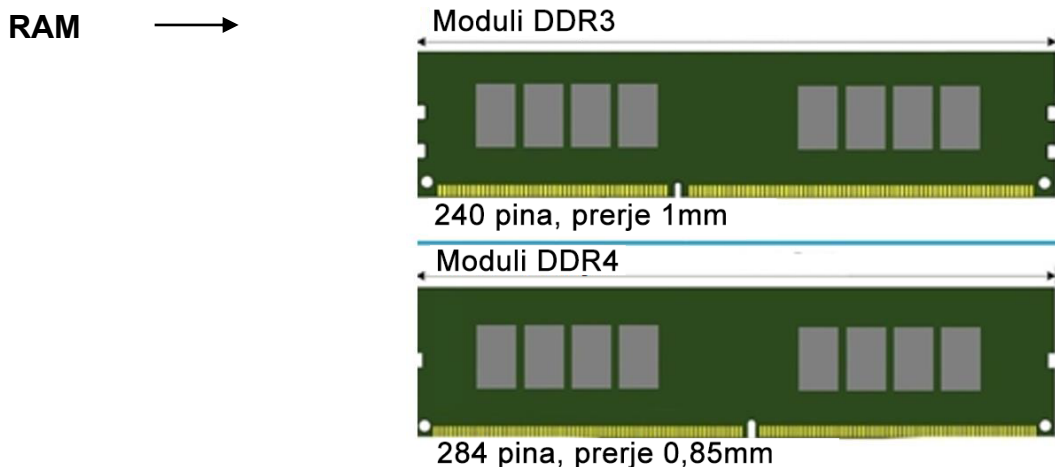


Figura 2.7. Lloje të memorieve RAM

Nga zgjedhja e pllakës varet edhe lloji dhe kapaciteti i memories RAM. Sot kryesisht përdoren llojet e mëposhtme të memories RAM: DDR4, DDR3, DDR2 dhe DDR. Për shembull, një modul RAM DDR4 nuk mund të vendoset në një slot DDR3. DDRAM (angl. Double Data Rate RAM) është lloj memorie RAM që transferon dyfishin e të dhënave për një takt. Ato dallohen për nga numri dhe renditja e pinave, furnizimi dhe frekuenca e punës, gjegjësisht shpejtësia. Një nga elementet për të identifikuar llojin e memories RAM është prerja që gjendet midis pinave.

Kartela grafike → **PCI Expressx16 dhe AGP** janë konektorët më të mirë për lidhjen e kartelave grafike. Shkurtesa PCI Express x16 (ang. Peripheral Component Interconnect Express) do të thotë lidhje eksprese të komponentëve periferike me gjerësi brezi prej 16 bitëve për një takt. Shkurtesa AGP (ang. Accelerated Graphics Port) do të thotë portë grafike me përsheptim. Slotet PCI janë me shpejtësi më të vogël dhe me qasje të kufizuar në memorie në krahasim me AGP. Të theksojmë se disa pllaka amë kanë

kartela grafike të integruara, por është mire që ajo të ketë edhe slotë shtesë, në rast të përmirësimit të video - sistemit.

- PCIx1** → Sloti PCI x 1 përdoret për lidhjen e **kartelave të zërit, kartelave të rrjetit, kontrolluesve të disqeve** dhe kontrolluesve të tjerë për pajisjeve periferike të ndryshme.
- BIOS** → BIOS (angl. Basic Input Output System) dhe CMOS (angl. Complementary Metal Oxide Semiconductor) janë dy lloje të ndryshme të memorieve në pllakën amë, të cilat nuk mundën pa njëra tjetrën, por që dallohen për nga karakteristikat dhe funksionet e tyre. BIOS-i është memorie ROM që përmban një program i cili ekzekutohet kur kyçet kompjuteri dhe nga aty rrjedh edhe shkurtesa që do të thotë baza e hyrje-daljes së sistemit. BIOS-i kryen kontrollin e komponentëve harduerikë, lokalizon sistemin operativ dhe e ngarkon në memorie, kujdeset për sigurinë e sistemit dhe kryen kontrollin e fjalëkalimeve. CMOS-i është memorie RAM e cila përdor një bateri të veten për t'u furnizuar. Në CMOS ruhen të dhënat për orën dhe datën, si dhe për konfigurimin e komponentëve të harduerit gjatë startimit të kompjuterit. Shkurtesa CMOS shërben për përshkrimin e teknologjisë së gjysmëpërçuesve për prodhimin e qarqeve të integruara.
- Hard disku dhe SSD** → Numri më i madh i pllakave amë përmbajnë **dy lloje të konektorëve SATA (angl. Serial Advanced Technology Attachment), SATA2 dhe SATA3**. Shkurtesa do të thotë teknologji e avancuar serike për lidhje. Konektori SATA2 mbështet shpejtësi transmetimi të të dhënave deri në 3Gb/s ose afërsisht 300MB/s, ndërsa SATA3 mbështet shpejtësi deri në 6Gb/s ose 600MB/s. Konektori SATA2 përdoret për lidhjen e hard disqeve dhe në të lidhet kabloja e të dhënave i hard diskut. Hard disku ka edhe një kablo SATA për furnizim me energji dhe ajo lidhet me njësinë për furnizimin me energji. Konektori SATA3 përdoret për lidhjen e memories SSD.
- Moduli M.2 SSD** → Modelet më të reja të pllaka amë përmbajnë konektor të veçantë me shenjën M.2 SSD për lidhjen e moduleve të veçantë SSD. Këta ndryshojnë nga pajisjet memoruese SSD sipas pamjes fizike dhe lidhjes së tyre dhe kjo është treguar në figurën 2.8.
Madje edhe vetë modulet SSD mund të kenë dy lloje të konektorëve: PCIe ose SATA. Moduli me konektim PCIe ka një

prerje të vendosur mes pinave, ndërsa moduli me konektim SATA ka dy prerje. Prandaj, gjatë përzgjedhjes së modulit duhet pasur kujdes se çfarë lloji është konektori M.2 SSD në vetë pllakën amë. Moduli M.2 SSD me konektim PCIe ka shpejtësi më të madhe të transmetimit të të dhënave.

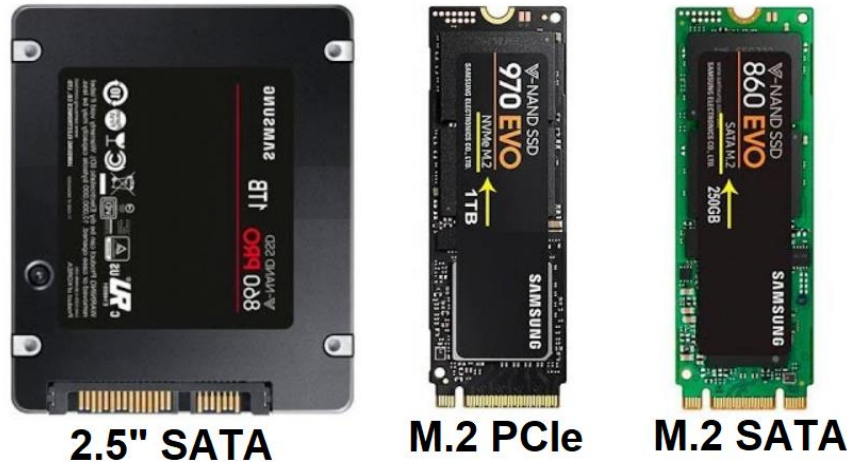


Figura 2.8. Krahasimi mes pajisjes memoruese SSD SATA dhe modulit M.2 SSD

Portat serike dhe paralele

→ Portat serike dhe paralele nuk hasen në gjeneratat më të reja të kompjuterëve personal, por ato akoma kanë zbatimin e tyre. Për shembull, porta serike përdoret për lidhjen e modemeve, ruterëve ose pajisjeve RS-232, siç janë kontrolluesit industrialë. Nëse kompjuteri nuk ka portë serike, atëherë e nevojshme është të përdoret një konvertor USB në serik. Porta paralele akoma përdoret për lidhjen e printerëve.

Konektorët për pajisjet video

→ VGA (angl. Video Graphics Array) është standard video analog. Vetë shkurtesa nënkupton varg i video grafikës dhe ka të bëjë me vargun dydimensional të ngjyrave të definuara nga vetë standardi. Përdoret për lidhjen e monitorëve, televizorëve, projektorëve dhe pajisjeve të tjera video. **HDMI (angl. High Definition Multimedia Interface)** është konektor për pajisje video dhe audio dhe ai me kualitetin e transmetimit ka zëvendësuar konektorin VGA. Shkurtesa HDMI do të thotë lidhjen e pajisjeve multimediale me rezolucion të lartë.

Pajisjet USB

→ Konektorët USB janë ndër konektorët më të përdorurat për lidhjen e një sërë pajisjesh periferike siç janë: miu, tastiera, playerat, hard disqet e jashtme, kamerat dhe aparatet fotografike, si dhe platformat zhvillimore siç janë Arduino dhe Raspberry Pi. Zakonisht funksionojnë në principin li dhe

funksiono (angl. plug and play), që do të thotë se softueri për këto pajisje inatsalohet vetvetiu.

2.1.4. Pajisja e furnizimit

Që të funksionojnë komponentët harduerike në përbërje të një sistemi kompjuterik, kanë nevojë për furnizim me energji. Pajisja e furnizimit me energji përdor tensionin e rrjetit prej 220V dhe detyra e saj është **të sigurojë nga tensioni** i rrjetit më **tepër tensione të vazhduara** për furnizim të komponentëve harduerikë. Në përbërjen e saj, kjo pajisje përmban transformator të rrjetit, drejtues dhe stabilizator.



Figura 2.9. Pamja e jashtme e pajisjes për furnizim dhe kablot për furnizim

HNë figurën 2.9 është paraqitur pamja e jashtme e pajisjes për furnizim dhe konektorët për lidhjen e kablove të furnizimit. Kablot më të rëndësishme për furnizim me energji janë paraqitur në figurën e njëjtë. Një pjesë e komponentëve harduerikë furnizohen drejtpërdrejt nga pajisja, ndërsa pjesa tjetër indirekt, përmes pllakës amë. Për furnizimin e pllakës amë përdoret kabllloja **ATX (20+4)** dhe kabllloja **ATX 12V (4+4)** për shkallën e furnizimit të bërthamës së mikroprocesorit. Për këtë arsye, kabllloja e dytë njihet edhe si kabllloja për furnizimin e procesorit. Për furnizimin e procesorit mjaftojnë vetëm 4 pina, por nëse mikroprocesori punon me frekuencë më të lartë se frekuenca nominale, atëherë lidhen të gjithë 8 pinat. Kablllo 8-pinësh është gjithashtu edhe kabllloja **PCI-E (6+2)**, i cili përdoret për furnizimin e kartelës grafike. Duhet pasur kujdes që kjo kablllo të mos ngatërrohet me kabllon ATX 12 (4+4). Kabllot SATA përdoren për furnizimin e hard disqeve dhe SSD memories.

Në figurën 2.10 është paraqitur skema e pllakës amë dhe janë shënuar konektorët dhe kushinetat e kablove për furnizim.



Figura 2.11. Kontrollimi i tensionit të pinave të daljes në kabllon për furnizim ATX (20+4)

2.2. 1.1. Pajisjet periferike dhe karakteristikat e tyre

Pajisjet periferike njihen edhe si njësi hyrëse-dalëse. Përmes njësive hyrëse, përdoruesi fut të dhëna në kompjuter, ndërsa përmes njësive dalëse ai merr informacion nga kompjuteri. Në thelb, mund të thuhet se përdoruesi lidhet me kompjuterin përmes të dhënave hyrëse dhe dalëse. Për shembull, tingujt në natyrë janë të vazhdueshme, sinjale analoge. Pasi ato të shndërrohen në sinjale elektrike, është e nevojshme që ato të digjitalizohen për t'u përpunuar nga kompjuteri. Nga sinjalet e vazhdueshme merren mostra në intervale të përcaktuara saktë, mostrat e mara rumbullakohen në numra të plotë dhe pastaj vlerat e marra kodohen, gjegjësisht ato paraqesin kombinime të zerove dhe njëshave. Ne do të njihemi me parimet bazë të funksionimit dhe karakteristikat e pajisjeve periferike më të rëndësishme.

Pajisjet hyrëse janë: tastiera, miu, skaneri, lexuesi magnetik, kamera dhe të tjera.

Tastiera përdoret për futjen e tekstit, numrave, shenjave alfanumerike si dhe për kontrollimin e operacioneve. Tastiera klasike përmban nga 101 deri në 104 tasta dhe standardi më i zakonshëm i renditjes së trastave është QWERTY. Për lidhjen e tastierës me kompjuterin përdoren konektorët PS2 ose porta USB. Ekzistojnë edhe tastiera pa tela, gjegjësisht tastiera Bluetooth. Vetë tastat mund të jenë mekanike dhe me membranë. Edhe pse tastiera mekanike është lloji më i vjetër, ajo ende është aktuale për shkak të sigurisë së kontakteve mekanike. Me secilën shtypje të tastierës dërgohet informacion prej një bajti tek mikroprocesori. Mikroprocesori e sheh tastierën si matricë, si rrjet i rreshtave dhe kolonave dhe në çdo pikë të takimit të matricës ka një ndërprerës. Matrica me ndërprerësit është paraqitur në figurën 2.12. Secili tast ka koordinatat e tij që përpunohen me numrin e rreshtit dhe numrin e kolonës. Nëse shtypet një tast i tastierës, atëherë përcaktohen koordinatat e tastit

dhe duke u bazuar në këtë informacion, koduesi në tastierë gjeneron kod unik, i cili pastaj dërgohet tek mikroprocesori.

komponenta e ndërfaqes

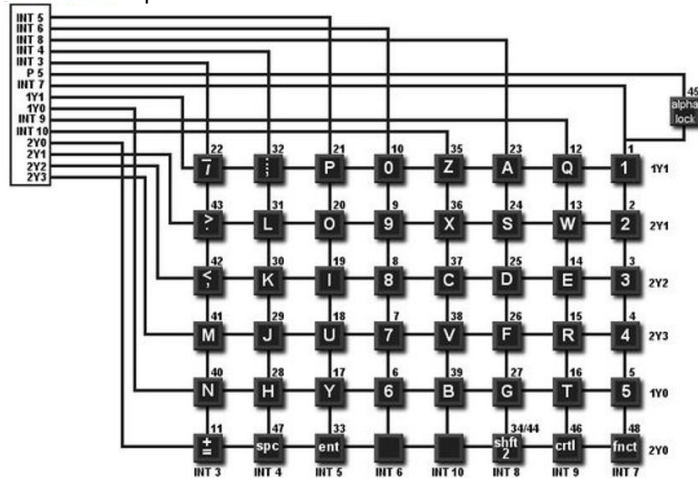


Figura 2.12. Struktura e brendshme e tastierit

Miu i kompjuterit përdoret për lëvizjen e treguesit nëpër ekranin e monitorit. Me klikimin e majtë selektohen objektet, ndërsa me klikimin e djathtë hapet menyuja për objektin e selektuar.

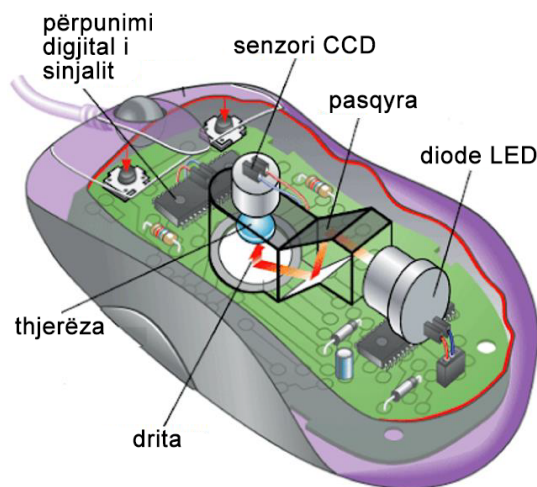


Figura 2.13. Struktura e brendshme e miut

Sipas mekanizmit, mausët e kompjuterëve ndahen në elektromekanikë, optikë dhe laserik. Në pjesën e poshtme të mausit optik janë të vendosur dy elemente, **dioda LED dhe senzori CCD**, siç tregohet në figurën 2.13. Sensori CCD përfaqëson një matricë elementesh fotosensitiv. Rrezja e drejtuar e dritës reflektohet nga sipërfaqja dhe i simulon elementet fotosensitiv. Kur drita e reflektuar bie mbi një elementin fotosensitiv, ai gjeneron tension elektrik. Mikroprocesori e identifikon elementin fotoelektrik të simuluar sipas koordinatave të tij (numrit të rreshtit dhe numrit të kolonës). Pastaj mikroprocesori e vendos treguesin e ekranit në të njëjtën pozitë si edhe elementi fotosensitiv i

simuluar në matricën e sensorit CCD. Mausët laserik kanë saktësi më të madhe, domethënë rreze drite më të fokusuar dhe më të ngushtë në krahasim me mausët optik.

Skeneri optik ose lexuesi është pajisje që i shndërron tekstet, imazhet ose fotografitë në formën e printuar në sinjal elektrik. Skeneri përbëhet nga një burim drite dhe një sensor optik i cili e detekton dritën e reflektuar. Nëse duam që tekstin e skanuar ta përpunojmë si tekst dhe jo si fotografi të nevojshme janë programe të posaçme. Karakteristika bazë të skenerit janë rezolucioni dhe thellësia e ngjyrave. Rezolucioni matet me numrin e pikave për një inç dhe sillet në kufijtë nga 75dpi (angl. dots per inch) deri në 960dpi. Thellësia e ngjyrave matet sipas sasisë së informacionit që skeneri mbledh për një pikë të imazhit. Për imazhe bardhë e zi mjafton një skener 8-bitësh, ndërsa skenerët për imazhe me ngjyra mund të jenë 24 ose 36-bitësh.

Lexuesi magnetik është i njohur si lexues i kartelave, siç janë kartelat e bankave.

Tableti grafik është pajisje hyrëse që i shndërron vizatimet e bëra me dorë, grafiqet, animacionet në formë elektrike, të përshtatshëm për përpunim kompjuterik. Trashësia e linjave, saturimi i ngjyrave, transparencja dhe cilësitë e tjera të vizatimit varen nga forca me të cilën stilolapsi shtypet mbi sipërfaqen punuese të tabletit grafik.

Njësi dalëse janë: monitori, printeri, kufjet ose altoparlantët.

Monitori dhe kartela grafike përbëjnë sistemin video të kompjuterit. Monitori na mundëson që në mënyrë vizuale të ndjekim aktivitetet në brendësi të mikroprocesorit.

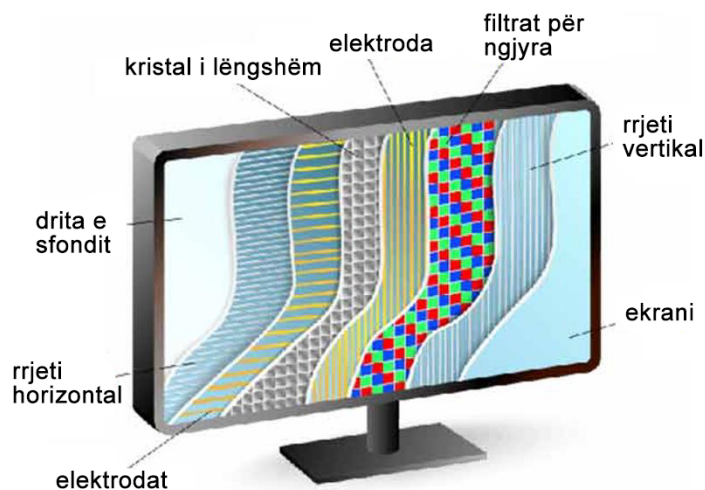


Figura 2.14. Struktura e brendshme e LCD ekranit

Pikseli është pika më e vogël e ekranit me një ngjyrë saktësisht të përcaktuar. Gjithashtu, pikselat janë të organizuar në formë të matricës. Rezolucioni është numri i pikselave në një vijë horizontale ose vertikale të ekranit. Për shembull, rezolucioni Full HD nënkupton $1920 \times 1200 = 2\,304\,000$ piksela të ekranit. Shpejtësia e rifreskimit (angl. Refresh Rate) matet në herc dhe

tregon numrin e imazheve për një sekondë. Madhësia e ekranit jepet me numër në inç dhe madhësi standarde janë 15, 17, 19, 21 dhe 27 inç. Çdo piksel përbëhet nga tre nënpiksela, nga një nënpiksel për secilën nga tri ngjyrat kryesore: e kuqe, e gjelbër dhe blu (RGB). Për shkak se nënpikselat ndodhen në distanca shumë të vogla, vjen deri te përzierja e ngjyrave. Në varësi të intensitetit të ngjyrave RGB, mund të fitohet çdo ngjyrë e spektrit të dukshëm. Ekrane më shpesh të përdorur janë LCD (angl. Liquid Crystal Display), që në përkthim do të thotë **ekran me kristale të lëngëta**. Molekulat e kristaleve të lëngëta lejojnë kalimin e dritës së sfondit kur nuk janë nën ndikimin e tensionit elektrik. Nënpikselat e kuq, blu dhe të gjelbër fitohen përmes filtrimit të dritës. Struktura e ekranit LCD është paraqitur në figurën 2.14. Karakteristika më të rëndësishme të ekraneve LCD janë: rezolucioni natyror, shpejtësia e reagimit (angl. Response Rate), këndi i shikimit, ndriçueshmëria dhe kontrasti.

Printerët janë pajisje dalëse që informacionet nga kompjuteri i transferojnë në letër ose ndonjë medium tjetër, gjegjësisht dokumentet nga forma elektronike i shndërrojnë në formë të shtypur.

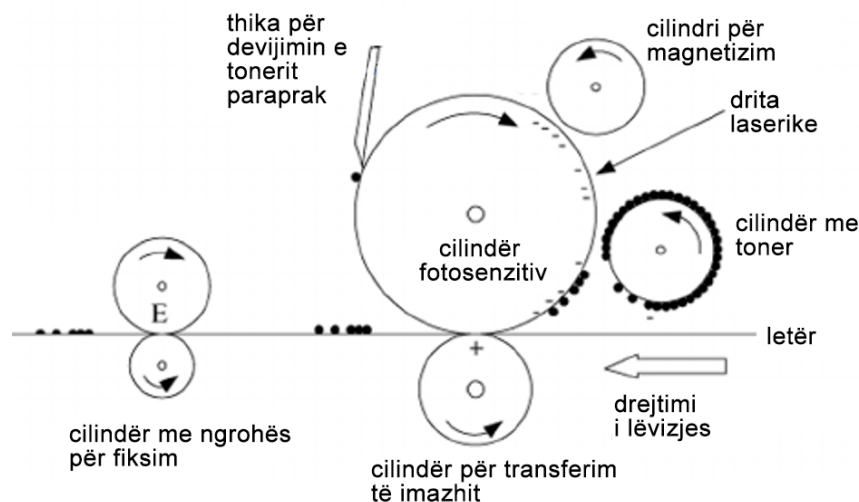


Figura 2.15. Struktura e brendshme e printerit laserik

Ekzistojnë tre lloje printerësh: me matricë, ink-jet dhe laserik. Pjesët përbërëse të një printeri laserik janë: kasete e tonerit, moduli laserik, sistemi për transferimin e imazhit dhe ngrohësi për fiksimit. Pjesa më e rëndësishme në printerin laserik është **cilindri fotosenzitiv** në brendësi të kasetës së tonerit. Në fillim, e gjithë sipërfaqja e cilindrit është me elektricitet negativ. Drita laserike e fokusuar e largon sasinë e elektricitetit negativ në pjesët e cilindrit në të cilat duhet të printohet. Pastaj, imazhi elektrostatisht i formuar shndërrohet në një imazh të dukshëm. Grimcat e tonerit në formë pluhuri janë të elektrizuara negativisht dhe largohen nga pjesët e elektrizuara negativisht të cilindrit fotosenzitiv, ndërsa tërhiqen nga pjesët e elektrizuara pozitivisht (pjesët e rrezatuara me dritë laserike). Në këtë mënyrë formohet një imazh “pozitiv” në

cilindrin fotosensitiv. Më pas, toneri nga cilindri transferohet në letër dhe i njëjti fiksohet përmes nxehtësisë. Procesi i krijimit të imazhit të printuar është paraqitur në figurën 2.15. Rezolucioni është parametri më i rëndësishëm për marrjen e një imazhi cilësor të printuar dhe tek printerët laserik sillet nga 600 deri në 2400 pika për inç (angl. dots per inch). Gjithashtu, një tjetër karakteristikë është shpejtësia e printimit, e cila arrin deri në 40 imazhe në minutë për printerët për punë kancelarie, ndërsa ekzistojnë edhe printerë profesionalë me shpejtësi nga 12 deri në 60 faqe në minutë.

Nga pajisjet hyrëse-dalëse do të përmendim kartelën grafike, kartelën zanore dhe kartelën e rrjetit.

Kartela grafike ose video përpunon imazhet dhe gjithçka që shihet në ekranin e monitorit është rezultat i punës së kësaj karte. Kartela grafike ka RAM memorien e saj video, kapaciteti i së cilës mesatarisht sillet nga 4GB deri në 8GB, edhe pse në treg ofrohen edhe kartela grafike me kapacitet prej 24GB RAM memorie. Në këtë memorie ruhen imazhet e papërpunuara të marra nga procesori, si dhe imazhet e përpunuara që dërgohen drejt monitorit.



Figura 2.16. Dy modele të kartelave grafike: MSI GeForce GT710 dhe MSI GeForce GTX1050 Ti

Nga kapaciteti i video RAM memories varet rezolucioni i imazhit të marrë. Kartelat grafike mesatare kanë një shpejtësi përpunimi prej rreth 50 imazhe për sekondë. Për monitorët analog përdoret konvertori RAMDAC, i cili shndërron sinjalet digjitale në analoge. Gjatë përzgjedhjes së kartelës grafike, duhet të merren parasysh tre parametra: kapaciteti i video RAM memories, gjatësia fizike dhe konsumi i energjisë elektrike. Gjatësia e kartelës mund të jetë nga 120mm deri në 290mm. Në figurën 2.16 është paraqitur pamja e jashtme e dy modeleve të kartelave grafike: MSI GeForce GT 710 dhe MSI GeForce GTX 1050 Ti. [12]

Pjesët përbërëse të **kartelës së zërit** janë procesori për përpunimin e zërit dhe konvertori CODEC (konvertori analogo/digjital dhe digjitalo/analog). Hyrje standarde janë: Line in (për lidhjen me sistem muzikor), Mic in (mikrofoni) dhe CD in (për lidhjen me një CD player të integruar). Dalje standarde janë: Speaker out, Line

out dhe S/PDIF (Sony/Philips Digital Interface). Dalja e fundit mund të jetë për lidhje koaksiale ose optike.

Kartelat e rrjetit shërbejnë për lidhjen e kompjuterit në një rrjet lokal (LAN - Local Area Network) ose në një rrjet pa tela (WAN - Wireless Area Network). Disa pllaka amë kanë kartela rrjeti të integruara.

2.3. Zgjedhja e komponentëve harduerikë dhe llogaritja e kostos për kohën dhe financat e shpenzuara

Teknologjia kompjuterike zhvillohet me një shpejtësi të madhe. Çdo ditë paraqiten lloje të reja të mikroprocesorëve, kartelave grafike, ekraneve, aplikacione që ndryshojnë mënyrën e komunikimit, mënyrën e marrjes së informacioneve, mënyrën me të cilën punojmë, mësojmë, argëtohemi, funksionojmë etj. Pa pajisje kompjuterike adekuate nuk mund të plotësojmë nevojat dhe ambiciet tona. Nga ana tjetër, zgjedhja e harduerit mund të jetë shpesh herë e kufizuar për shkak të buxhetit të vogël dhe njohja e pamjaftueshme e komponentëve harduerikë. Në temën e parë, kur folëm për llojet e mikrokompjuterëve, theksuam se një laptop me të njëjtin konfigurim si një desktop kushton pothuajse dyfish më shumë. Nëse mund të heqim dorë nga portativiteti, atëherë është shumë më mirë të zgjedhim një desktop. Gjithashtu, gjatë përzgjedhjes së kompjuterit është shumë e rëndësishme destinimi i kompjuterit. Shumë njohës të teknologjisë kompjuterike hezitojnë nëse është më mirë të blejnë një produkt të gatshëm ose të konfigurjnë dhe montojnë vetë një desktop kompjuter. Të dyja opsionet kanë përparësitë dhe disavantazhet e tyre. **Blerja e një kompjuteri të gatshëm është më e lehtë dhe më e shpejtë.** Nuk harxhojmë kohë për zgjedhjen e komponentëve, nuk investojmë mund për montimin dhe lidhjen e tyre, ndërsa kompjuteri është gati për përdorim sapo e blejmë. Kur bëhet fjalë për kompjuter të gatshëm, përfitohet mbështetje e plotë për përdoruesin, garanci, servisim dhe ndonjëherë edhe sistem operativ i instaluar falas. Disavantazhi është **mungesa e qasjes individuale** gjatë përzgjedhjes së kompjuterit, domethënë vetë përdoruesi nuk ka mundësi vetë të zgjedhë komponentët harduerike. Më lehtë është të blihet konfigurimi më i shtrenjtë ose të porositet i njëjti, por a është kjo zgjidhja më e mençur nga pikëpamja e kostos së investimit. Nga ana tjetër, konfigurimet e gatshme, zakonisht janë me cilësi dhe siguri më të dobët. Në botë ekzistojnë shumë prodhues të komponentëve harduerikë, disa prej të cilëve janë të panjohur ose me kualitet të dyshimtë. Numri më i madh i produkteve të tyre përfundojnë në konfigurime të gatshme dhe jo si produkte të veçanta për shitje. Vetë kompanitë, siç janë Dell ose HP tarifojnë për shërbimet e tyre të konfigurimit dhe montimit të kompjuterit, gjë që natyrisht ndikon në çmimin e kostos. Nëse vetë zgjedhim

komponentët harderike, sigurt që përfitojmë një kompjuter më cilësor, me një jetëgjatësi më të madhe.

Konfigurimi	Produkt i gatshëm	Montim vetiak
Procesori	AMD Ryzen 3 3300X Quad-Core 3.8 GHz Socket AM4 65W 100-100000159BOX Desktop Processor	7750 denarë
Pllaka amë	MSI B450 TOMAHAWK MAX II AM4 AMD B450 SATA 6Gb/s ATX AMD Motherboard	5980 denarë
Memoria RAM	G.SKILL Ripjaws V Series 16GB (2 x 8GB) 288-Pin DDR4 SDRAM DDR4 3200 (PC4 25600) Desktop Memory Model F4-3200C16D-16GVKB	4316 denarë
Kartela grafike	GIGABYTE GeForce GT 1030 Low Profile D4 2G DirectX 12 GV-N1030D4-2GL Video Card	6180 denarë
Shtëpiza	DIYPC Shadow-H3-ARGB Black Steel / Tempered Glass ATX Mid Tower Computer Case w/ 3 x 120mm Halo ARGB LED Fans Pre...	3580 denarë
Burimi i ushqimit	Rosewill Glacier Series 500W Modular Gaming Power Supply with Silent Aero-Diversion Fan, 80 PLUS Bronze Certified,...	4160 denarë
SSD memoria	Intel 670p Series M.2 2280 512GB PCIe NVMe 3.0 x4 QLC Internal Solid State Drive (SSD) SSDPEKNU512GZX1	3380 denarë
Hard disku	Team Group CX1 2.5" 480GB SATA III 3D NAND Internal Solid State Drive (SSD) T253X5480G0C101	2600 denarë
Ftohës për procesor	PURE ROCK SLIM 2	1250 denarë
Çmimi i përgjithshëm	43500 denarë	39196 denarë
Vëretje	Me produktin e gatshëm kemi garanci 3-vjeçare	Në çmim nuk është llogaritur koha e harxhuar dhe munda i dhënë

Tabela 2.2 Llogaritja e harxhimeve për konfiguracionin

Pyetje më shpesh e shtruar është nëse është financiarisht e levërditshme të montojmë vetë kompjuterin tonë. Përgjigjja është: relativisht më lirë. Çmimi i kompjuterit që ne vetë e montojmë është pothuajse i njëjtë me atë të një kompjuteri të gatshëm. Megjithatë, **shikuar për një periudhë më të gjatë kohore, montimi i kompjuterit është më i levërditshëm**, sepse kualiteti më i mirë dhe përshtatja e kompjuterit sipas nevojave tona do të thotë jetëgjatësi më të gjatë të përdorimit, ose me fjalë të tjera, për një kohë më të gjatë nuk duhet të mendojmë për një investim të

ri. Në tabelën 2.2 është treguar përlllogaritja e harxhimeve nëse blejmë një kompjuter të gatshëm ose e montojmë atë vetë.

Një avantazh tjetër i montimit të pavarur të kompjuterit është se përdoruesi shumë lehtë do të mund ta mirëmbajë dhe ta përmirësojë atë. Me njohuritë që do të fitojë, do të regojë në kohë në rast defekti dhe do të mund vetë të zëvendësojë pjesën e dëmtuar.

Nëse vendosim për montim të kompjuterin vetë, gjatë përzgjedhjes së komponentëve harduerikë duhet pasur kujdes për **kompatibilitetin** e komponentëve. Për shembull, nëse zgjedhim një procesor dhe një pllakë amë më të shtrenjtë për llogari të një kartele grafike më të lirë, sistemi kompjuterik do të jetë me shpejtësi të vogël, sepse kartela grafike ngadalë do t'i përpunojë të dhënat video. Është veçanërisht e rëndësishme që, pllaka amë, mikroprocesori dhe kartela grafike të kenë **performanca të ngjashme (shpejtësi)**. Procesorët më të shitur janë Intel dhe AMD. Procesorët Intel janë më cilësorë, më rezistentë ndaj tejngruhjes, janë më të sigurt dhe kanë jetëgjatësi më të madhe. AMD procesorët kanë fuqi më të madhe ekzekutimi dhe janë më të popullarizuar midis lojtarëve të video-lojërave dhe dizajnerëve të videove. Gjatë zgjedhjes së kartelës grafike, duhet të vendosim midis një kartele të integruar ose jo të integruar. Kartela grafike e integruar është e implementuar në të njëjtin çip, bashkë me mikroprocesorin. Prodhuesi i AMD për këtë lloj ët kartelës përdor nocionin APU (angl. Accelerated Processing Unit). Kartelat grafike më të shitura janë GeForce nga kompania NVIDIA dhe Radeon nga AMD. Aplikacionet e reja janë më të avancuara dhe kërkojnë kapacitete më të mëdha të memories së punës. Për shembull, për punë me dokumente, siç është aplikacioni Microsoft Office, është e mjaftueshme që RAM-i të ketë kapacitet prej 2GB, por nëse bëhet fjalë për përpunimin e videove (aplikacioni Adobe Premiere), atëherë janë të nevojshme të paktën 8GB memorie pune. Përveç kapacitetit, rëndësi ka edhe shpejtësia e memories e cila matet në megaherc. Në treg sot ofrohen tre lloje memoriesh: DDR3, DDR4 dhe DDR5. Sa u përket memorieve të përherëshme, memoriet SSD janë më të shpejtë, më të qëndrueshme, por edhe më të shtrenjta krahasuar me hard disqet. Por nëse duam të investojmë në të dy llojet e memorieve, atëherë mundet që sistemin operativ ta instalojmë në një memorie SSD që të kursejmë, ndërsa hard disku të na shërbejnë për dokumente dhe aplikacione të përdoruesit. Për këtë qëllim, e mjaftueshme do të jetë një memorie SSD me kapacitet prej 120GB. Paraprakisht, duhet të sigurohemi nëse në shtëpi ka dy kuti për SSD dhe hard diskun. Në fillim, me pllakën amë lidhim vetëm memorien SSD dhe instalojmë sistemin operativ. Pas montimit të memories SSD lidhim hard diskun. Para përdorimit, i njëjti duhet të formatohet duke përdorur programin Disk Management që është pjesë e sistemit operativ Windows 10. Pas hapjes së programit Disk Management, në ekran do të shfaqen të gjitha disqet që janë të lidhur siç duhet. Duke klikuar në butonin e djathtë, zgjedhim opsionin Format. Për të jemi të sigurt që dokumentet dhe aplikacionet do të ruhen në hard disk, nga vegla Settings zgjedhim "Change where new contents is saved" dhe e përzgjedhim hard diskun.

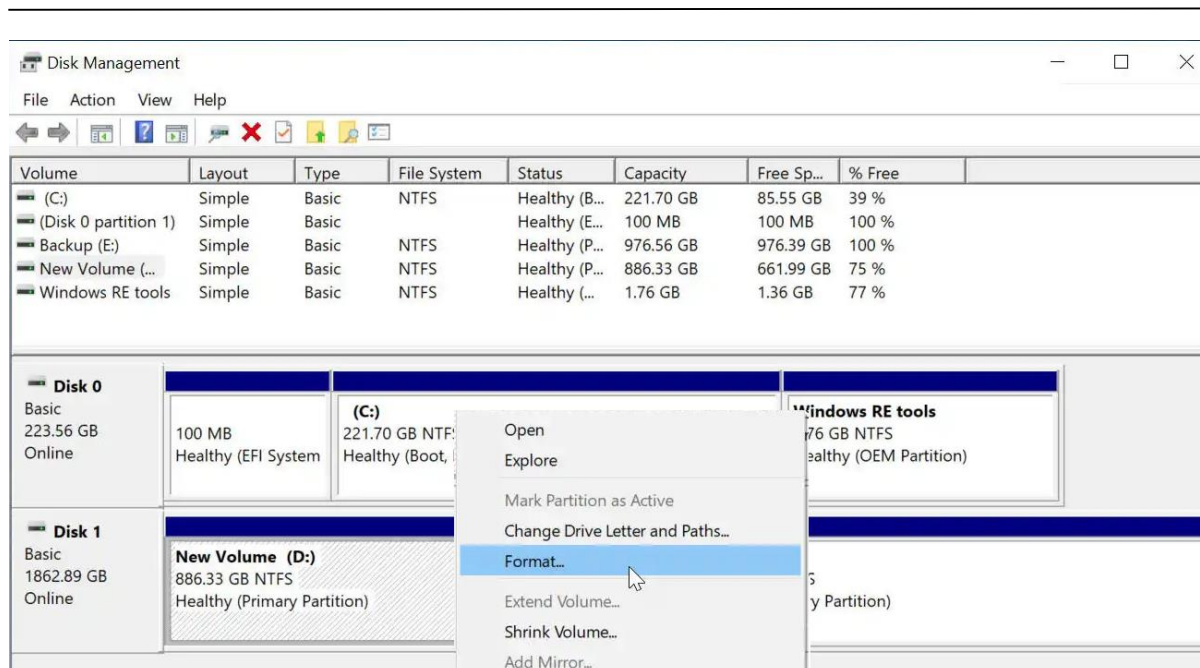


Figura 2.17. Formatimi i hard diskut

Hapi i fundit është përshtatja e ngarkimit të sistemit operativ në BIOS. Në menynë rënëse të dritares Boot order, memoria SSD duhet të jetë në vendin e parë.

Nga kjo mund të përfundohet se përzgjedhja e komponentëve harduerikë është shumë e gjerë dhe është shumë e rëndësishme që përdoruesi t'i njohë mirë ato për të marrë vendimin e duhur.

2.4. Përmirësimi i konfigurimit të kompjuterit

Përmirësimi i lehtë është avantazh i desktopit krahasuar me laptopin. Komponenti më i rëndësishëm harduerik për përmirësimin e kompjuterit është pllaka amë, pra duhet të dimë se cilët **konektor janë të lirë** për lidhjen e komponentëve harduerikë të reja.

Që të rritet shpejtësia e punës së kompjuterit, më e lehtë është **përmirësimi i memories RAM**. Para se të marrim vendimin për përmirësim, duhet të kontrollojmë sa memorie po përdor kompjuteri ynë. Për këtë do të na ndihmojë programi **Performance Monitor**. Nëse shfrytëzimi është mbi 50%, atëherë përmirësimi i RAM-it është i nevojshëm. Pastaj duhet të kontrollojmë nëse ka **slote DIMM të lira** ku do të vendosim modulet e reja të memories. Nëse nuk e njohim llojin e memories RAM në kompjuterin tonë, mund ta mësojmë nëse në Windows 10 e hapim veglën Task Manager dhe zgjedhim opsionin Performance, siç është treguar në figurën 2.18. Nëse bëhet fjalë për sistemin operativ Windows 7, kontrollimi i memories RAM mund të bëhet përmes Command Prompt duke përdorur komandën "wmic MEMORYCHIP get BankLabel,

DeviceLocator, MemoryType, TypeDetail, Capacity, Speed”. Vetë montimi i memories RAM është shpjeguar në pjesën praktike të librit, njësia mësimore 2.2. Ushtrim praktik për instalimin e komponentëve përbërëse të kompjuteri personal.

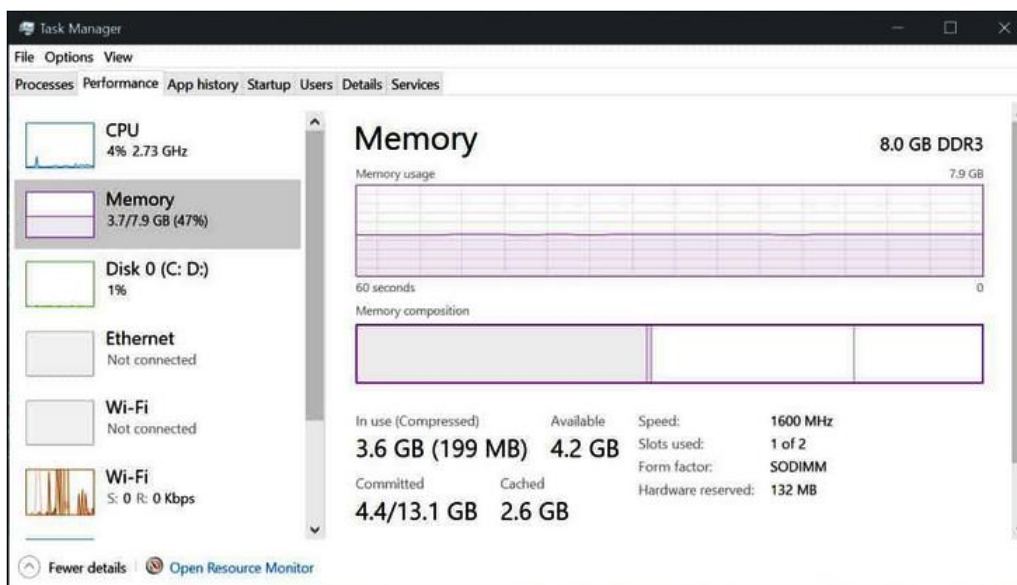


Figura 2.18. Kontrollimi i llojit dhe kapacitetit të memories RAM

Sa i përket memories së përhershme, tashmë shpjeguar se zgjidhja më e mirë është të investohet në të dy llojet e memorieve, me atë që në memorien SSD do të ruajmë sistemin operativ, ndërsa në hard disk do të ruajmë dokumentet e përdoruesit. Nëse kryejmë përmirësim, para formatimit të hard diskut, është e nevojshme që të njëjtin ta ndajmë në dy pjesë, ndërsa në ndarjen e dytë të ruajmë të gjitha dokumentet e përdoruesit për të mos i humbur ato. Në ushtrimin praktik nr. 2.5, do të fokusohemi në instalimin e një pajisjeje memoruese SATA SSD dhe një moduli memorues M.2 SSD.

Krahas përmirësimit të memories së punës dhe asaj të përhershme, konfigurimi i kompjuterit mund të zgjerohet edhe duke shtuar një kartelë grafike ose rrjeti të re. Kompjuteri mund të punojë me dy kartela, por pas montimit të tyre, duhet të kryhen përshtatje të caktuara.

Për shembull, shpejtësia dhe stabiliteti i lidhjes së internetit nuk varen vetëm nga lloji i paketës së internetit të zgjedhur, por edhe nga performanca e kartelës së rrjetit.



Figura 2.19. Adapteri i rrjetit me konektor kabllloja dhe antenë

Nëse të gjitha pajisjet e rrjetit në shtëpinë tonë kanë internet cilësor, përveç desktopit tonë me konfigurim më të vjetër, atëherë ndoshta duhet të zëvendësojmë kartelën e tij të rrjetit. Edhe një herë, zgjedhja e kartelave të rrjetit është shumë e larmishme. Ato dallohen para së gjithash në shpejtësinë, e cila mund të jetë 10Mbit/s, 100Mbit/s ose 1000Mbit/s. Mund të jenë të integruara, të jashtme ose të brendshme. Kartelat e brendshme të rrjetit lidhen përmes konektorëve PCI ose PCI Express. Kartelat e jashtme zakonisht janë adapter rrjeti me USB dhe të njëjtit mund të jenë me kablo (me lidhje kabloja RJ45) ose pa kablo (me lidhje për antenë). Këto dy lloje të adapterëve të rrjetit janë paraqitur në figurën 2.19.

Mundësitë për konfigurimin dhe përmirësimin e kompjuterëve janë shumë të mëdha. Çdo ditë dalin komponentë harduerikë të reja dhe që të përshtatemi me sfidat e reja teknologjike, duhet gjithmonë të kërkojmë zgjidhje të reja për kompjuterët tanë personalë.

Përfundime

Mikroprocesori ka **dy funksione kryesore**: ekzekuton programe dhe menaxhon me pajisjet tjera në pllakën amë. Pjesë përbërëse të një mikroprocesori janë: regjistrat, njësia aritmetiko-logjike dhe njësia e kontrollit me dekodues. Karakteristika më të rëndësishme të mikroprocesorit janë: frekuenca e punës, gjerësia e të dhënave, koncepti i qarkullimit, kompleksiteti i grupit të instruksioneve, numri i bërthamave, renditja e pinave, etj.

Memoria RAM e furnizon procesorin me informacione, ndërsa vetë ajo merr informacion nga ndonjë memorie e përhershme, siç për shembull hard disku. Të gjitha programet vendosen në hard disk dhe ato transferohen në RAM kur duhet të përpunohen.

Memorie të përhershme janë hard disqet, memoria SSD dhe memoriet e jashtme (kompakt disqet, USB memoriet, kartelat SD, etj.).

Përzgjedhja e pllakës amë varet nga zgjedhja e procesorit sepse çdo lloj i procesorit ka renditje unike të pinave të cilët kërkojnë edhe bazament përkatës.

Çipseti është grup i qarqeve të integruara speciale të cilët kontrollojnë qarkullimin e të dhënave.

DDR4, DDR3, DDR2 dhe **RAM modulet** DDR ndryshojnë sipas numrit dhe renditjes së pinave, furnizimit me energji dhe frekuencës së punës përkatësisht shpejtësisë.

PCI Express x16 dhe AGP janë konektorët më të mirë për lidhjen e kartelave grafike. Sloti PCI përdoret për lidhjen e kartelave audio dhe kartelave të rrjetit. Konektorët SATA2 dhe SATA3 janë për lidhjen e hard disqeve dhe memories SSD.

Para montimit të kompjuterit personal, detyrimisht duhet **të shkyçet furnizimi me energji** dhe të mbrohem nga elektriciteti statik.

Gjatë montimit të procesorit dhe memories RAM nuk duhet të shtypen dhe të preken pinat e artë.

Shumica e **kabllove të furnizimit kanë dizajn unik**, kështu që nuk mund të gabohet gjatë lidhjes së tyre në pllakën amë me pajisjet e tjera.

Tastiera, sensori CCD dhe ekrani LCD janë të organizuar në formën e **matricës**, ndërsa çdo element i matricës ka koordinatat e tij (numrin rendor të rreshtit dhe kolonës). Mikroprocesori i identifikon elementet sipas koordinatave të tyre.

Karakteristika themelore të kartës grafike janë kapaciteti i video memories RAM dhe shpejtësia e punës e cila matet me numrin e imazheve të përpunuara për një sekondë.

Konektori më i rëndësishëm **për furnizimin me energji është ATX 20+4** i cili përdoret për furnizimin e pllakës amë. Shumica e komponentëve harduerikë përdorin furnizimin me energji nga pllaka amë për furnizimin e tyre dhe lidhen përmes konektorëve të veçantë për furnizim. Tensionet dalëse të konektorit ATX 20+4 mund të kontrollohen me ndihmën e një multimetri.

Para përmirësimit të memprjes RAM është e nevojshme të kontrollohet korrektësia, lloji dhe kapaciteti i memories dhe numri i slotëve të lira DIMM.

Gjatë përmirësimit të memories së përhershme, **sistemi operativ duhet të instalohet në memorien SSD, ndërsa në hard disk duhet të ruhet dokumentet dhe aplikacionet e përdoruesit**. Gjatë blerjes së memories SSD duhet pasur kujdes në mënyrën e lidhjes me pllakën amë dhe renditjen e pinave të saj.

Blerja e një kompjuteri të gatshëm është më e lehtë dhe më e shpejtë. Nuk humbasim kohë për të zgjedhur komponentët, as nuk investojmë mund për lidhjen

dhe montimin e tyre dhe kompjuteri është i gatshëm për përdorim me blerjen e tij. Kur bëhet fjalë për kompjuter të gatshëm, ofrohet mbështetje e plotë për përdoruesin, garanci, servisim, ndërsa ndonjëherë edhe sistem operativ të instaluar falas.

Montimi vetanak i kompjuterit vetë do të thotë se fitojmë një produkt më kualitativ i cili plotësisht korrespondon me nevojave tona, me një jetëgjatësi më të gjatë përdorimi, mirëmbajtje dhe përmirësim më të lehtë.

Gjatë kohës së **procesit start-up**, kompjuteri lëshon sinjale zanore dhe vizuale të cilat mund të na ndihmojnë në diagnostikimin e gabimeve gjatë montimit të kompjuterit.

Për arsye të sigurisë personale dhe për mbrojtje të komponentëve nga dëmtimi, para fillimit të çdo montimi duhet të shkyçet burimi i furnizimit, pa marrë parasysh nëse bëhet fjalë për bateri ose kompjuter.

Pyetje dhe detyra:

1. Cilët janë tre komponentët përbërëse të mikroprocesorit dhe cili është funksioni i tyre?

2. Numëro karakteristikat më të rëndësishme të mikroprocesorit!

3. Cilat janë dy karakteristikat më të rëndësishme të memorieve?

4. RAM-i është memoria e punë, e përkohshme me qasje të rastësishme? Saqroje!

5. Cilat janë përparësitë dhe mangësitë e hard disk memories krahasuar me memorien SSD

6. Çfarë përfaqëson dhe për çfarë përdoret çipseti në pllakën amë të mikrokompjuterit?

7. Si mund të identifikohet lloji i RAM modulit DDR?

8. Cilat slotet në pllakat amë përdoren për lidhjen e kartelës grafike?

9. Çfarë ekzekuton BIOS-i pas çdo kyçje të kompjuterit?

10. Për çfarë përdoren konektorët SATA2 dhe SATA3 në pllakën amë?

11. Cili është konektori më i mirë për lidhjen e pajisjeve video?

12. Në katër hapa, shpjego instalimin e procesorit në kushinetën e tij.

13. Çfarë duhet të kemi kujdes gjatë instalimit të memories RAM në slotet e pllakës amë?

14. Për çfarë shërbejnë konektorët F (Front), SYS_FAN dhe PWR-FAN në pllakën amë?

15. Emërto kabllo të furnizim me energji për pllakën amë dhe procesorin? Nga sa pin ka secila kablo?

16. Shpjego procesin e montimit të memories SSD dhe hard disk memories, me renditjen e lidhjes së tyre, instalimin e sistemit operativ në memorien SSD dhe përshtatjen e hard diskut për ruajtjen e të dhënave të përdoruesit!

17. Zgjidhni një konfigurim të gatshëm të kompjuterit sipas dëshirës tuaj dhe llogaritni koston e rastësishme për montimin tuaj.

18. Demonstroni si komunikon mikroprocesori me tastierën!

19. Për çfarë shërben sensori CCD në përbërjen e mausit të kompjuterit?

20. Shpjego procedurën për kontrollimin e funksionalitetit të pajisjes së furnizimit me energji të kompjuterit!

21. Kontrolloni sa shfrytëzohet, llojin dhe kapacitetin e memories RAM në kompjuterin tuaj!

22. Kontrolloni llojin e kartelës grafike në kompjuterin tuaj dhe kërkoni në internet për karakteristikat e saj!

23. Çfarë lloj sinjalesh të zërit lëshon kompjuteri në rastin kur dështon ndonjë komponentë harduerikë? Së bashku me mësuesin kontrolloni se çfarë ndodh nëse e largojmë RAM-in nga sloti i tij në pllakën amë.

24. Zgjidhni një pllakë amë, sipas zgjedhjes tuaj, ndërsa kërkoni informacione në internet për shpërndarjen e konektorëve dhe slotëve!

3. 1. Instalimi i sistemit operativ dhe programeve tjera standarde në sistemin kompjuterik personal

3.1. Llojet e sistemeve operative

Sistemi operativ është softuer i nevojshëm për ekzekutimin e programeve aplikative dhe për koordinimin e aktiviteteve të sistemit kompjuterik. Ai siguron një mjedis ku programet e tjera mund të kryejnë punë të dobishme. Ai është program si çdo program tjetër, por shumë më kompleks, më i fuqishëm, më i madh. Për këtë arsye, ai duhet të krijohet pjesë-pjesë, me funksione të definuara mirë, hyrje dhe dalje për secilën pjesë. Sistemet operative mund të studiohen nga dy këndvështrime: nga ai i përdoruesit dhe ai i sistemit [2].

Nga këndvështrimi i përdoruesit sistemi operativ është dizajnuar për të minimizuar punën e përdoruesit gjatë shfrytëzimit të burimeve harduerike, duke siguruar përdorim të rehatshëm të sistemeve mikrokompjuterike dhe përmirësimin e performancës së tyre. Kompjuteri pranon instruksione nga përdoruesi përmes tastierës dhe dërgon mesazhe përmes monitorit. Në varësi të formës së këtyre komandave dhe metodës së futjes së tyre, dallojmë dy lloje ndërfaqesh përdoruesish: përmes linjës së komandës (angl. command line interface) ose ndërfaqes grafike (angl. graphics user interface). Në ndërfaqen me linjë komande përdoren komanda tekstuale, ndërsa në ndërfaqen grafike përdoret një sistem dritësh, menysh dhe listash për zgjedhje. Për të hyrë në ndërfaqen me linjë komande në sistemin operativ Windows, në fushën e kërkimit (angl. Search) duhet të shkruhet cmd (angl. Command Prompt).

Nga këndvështrimi i sistemit, sistemi operativ është softuer kontrollues dhe menaxhues, i lidhur ngushtë me harduerin. Sistemi operativ vendos se cili program do të ekzekutohet dhe kur, sa memorie do t'u caktohet programeve, kujdeset për

ekzekutimin e saktë të komandave të dhëna nga përdoruesi dhe të tjera. Shkurtimisht do të njihemi me funksionet e sistemit operativ:

- Ekzekutimin e programeve. Sistemi operativ është përgjegjës për transferimin e instruksioneve nga memorja RAM në mikroprocesor, dekodimin dhe ekzekutimin e tyre.
- Programet për shpërndarjen e burimeve harduerike. Për funksionimin efikas të sistemit mikrokompjuterik, sistemi operativ duhet të vendosë se si të shpërndajë burimet harduerike (mikroprocesorin, hapësirën e memories, njësitë hyrëse-dalëse) për përdoruesit dhe programet specifike. Për shembull, mikroprocesori mund të më tepër detyra njëkohësisht duke kaluar vazhdimisht nga njëra detyrë te tjetra, por kalimi bëhet aq shpesh sa që përdoruesit dhe programet e tyre janë gjithmonë në ndërveprim.
- Programet për menaxhimin e memories. Sistemi operativ vendos se cilat programe dhe të dhëna do të transferohen nga memoria sekondare në atë primare, ndan dhe liron hapësirë të memories sipas nevojës, ndjek vazhdimisht pjesët e memories që janë në përdorim, shërben për formatimin, pastrimin dhe mirëmbajtjen e hard diskut, ndërsa kujdeset për funksionimin e kesh memories.
- Programet për menaxhimin e operacioneve dhe pajisjeve hyrëse-dalëse. Për kontrollin e transferimit të të dhënave nga ose drejt pajisjeve periferike dhe për ruajtjen e tyre të përkohshme përdoren qarqe të integruara të veçanta të quajtura kontrollues të pajisjeve. Për të filluar transferimin e të dhënave, është e nevojshme që në regjistrat e kontrolluesit të shkruhen kode binare me vlera saktësisht të caktuara. Këtë e kryen drajveri, i cili në fakt është në fakt "lëvizësi i pajisjes". Drajverët janë programe ndihmëse që ndërmjetësojnë midis sistemit operativ dhe kontrolluesve.
- Programet për mbrojtje dhe siguri. Mbrojtja është mekanizëm për kontrollin e aksesit të përdoruesve në burimet harduerike dhe përdorimin e autorizuar të tyre. Mbrojtja e sistemit nga sulmet e brendshme dhe të jashtme është funksioni i sigurisë. Në këtë kategori përfshihen programet antivirus.
- Programet për komunikim. Këto programe mundësojnë transferimin e skedarëve në distancë nga një kompjuter në tjetrin, kërkimin e faqeve të internetit, dërgimin e postës elektronike, etj.
- Programet për menaxhim me skedarëve. Këto programe krijojnë, zgjedhin, fshijnë, kopjojnë, riemërtojnë skedarë dhe dosjeve.
- Programet për punë me informacione të statusit. Informacione statusi janë koha, data, sasia e memories së disponueshme, numri i përdoruesve, etj.
- Mbështetje për gjuhët programuese. Sistemi operativ i ofron përdoruesit editorë, kompajlerë, assemblerë, debagerë dhe interpretues për gjuhët

programuese më shpesh të përdorura. Këto vegla janë të domosdoshme për krijimin e softuerit aplikativ.

Sistemi operativ Windows është sistemi i parë operativ me ndërfaqe grafike dhe është produkt i kompanisë Microsoft. Ai është një nga sistemet operative më të përdorura dhe më të njohura. Sot, rreth 80% e përdoruesve të kompjuterëve personalë përdorin sistemin operativ Windows. Fillimisht, Windows ishte thjesht një përmirësim i sistemit operativ të atëhershëm të njohur DOS. Që në fillim, kishte mbështetje softuerike për të punuar me maus, ikona, meny dhe kalim të thjeshtë nga një aplikacion në tjetrin. **Përparësi të sistemit operativ Windows** janë: thjeshtësia në funksionim (user friendly), mbështetje e gjerë për lidhjen me pajisje periferike, qasje në softuer aplikativ, instalim automatik (angl. plug and play), performance më të mirë për video lojëra, kompatibilitet me shumicën e ueb faqeve. **Mangësi** janë: nevojën për konfigurime kompjuterike të shtrenjta, kdo të mbyllur, siguri të dobët kundrejt sulmeve në internet dhe mbrojtjes administrative, ndjeshmëri ndaj viruseve kompjuterike, është e nevojshme licenca, mbështetje të dobët nga ekspertë dhe shpeshherë për të rritur shpejtësinë e punës, është e nevojshme të riinstalohet sistemi operativ.

Linux-i është sistem operativ më i ri. Idenë e dha student finlandez Linus Torvalds, i cili në vitin 1991 publikoi një punim për krijimin e një bërthame të një sistemi operativ të ri falas të qëndrueshëm, me kdo të hapur i cili mund të përdoret në platforma të ndryshme. **Kodi i hapur ose i lire** do të thotë që të gjithë përdoruesit të kenë akses në kodin burimor dhe ta modifikojnë atë në varësi të nevojave të tyre. Kodi i hapur është sistem zhvillimor më i sigurt. Gjatë vetë analizës, programuesit i zbulojnë gabimet dhe viruset, i rregullojnë dhe zgjidhjet e veta i ndajnë me komunitetin e Linux-it. Vetë bërthama e sistemit operativ ka mbrojtje me disa nivele. Përpara se të krijohet një version i ekzekutueshëm i ndonjë programi, ai kontrollohet. Gjithashtu, shfletuesi i internetit është i pavarur nga sistemi operativ. Në figurën 3.1, në mënyrë simbolike është paraqitur organizimi i sistemit operativ Linux, krahasuar me organizimin e sistemit operativ Windows.

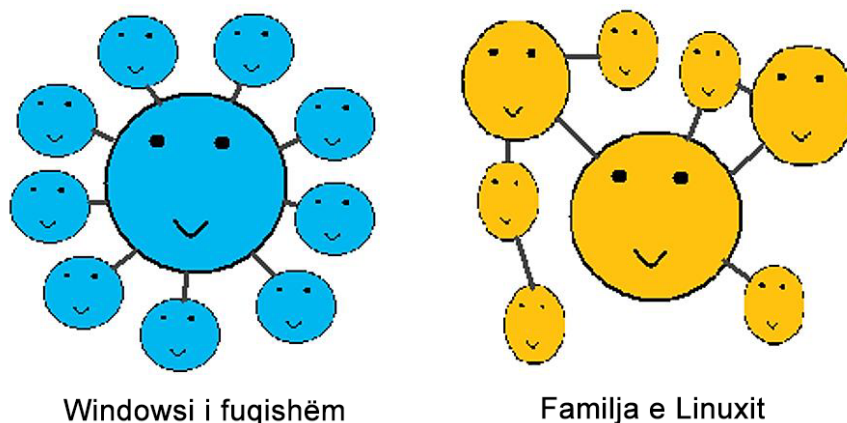


Figura 3.1. Dy koncepte të ndryshme por të suksesshme

Arkitektura e sistemit operativ Linux është si një pemë e degëzuar, ku çdo degë përfaqëson një anëtar të komunitetit dhe secili mund të kontribuojë në krijimin e një sistemi më të shpejtë, më të fuqishëm dhe më të sigurt. Ndoshta Windows është mbreti në botën e sistemeve operative, por revolucioni i Linux përfundimisht ka ndryshuar mënyrën e krijimit të sistemeve operative. Fillimisht, sistemi operativ Linux ishte i dedikuar për përdorues më të avancuar, por me shfaqjen e ndërfaqes GUI (angl. Graphical User Interface), Linux po bëhet gjithnjë e më popullor. Sot, Linux përdoret si sistem operativ për serverë në kompani të mëdha, si Google, Facebook, Twitter. Shumë kompani marrin bërthamën e Linux si bazë, e zhvillojnë dhe krijojnë sisteme operative të reja të licencuara. I tillë është sistemi operativ Android i kompanisë Google dhe sistemi operativ **Raspbian**, i krijuar nga një grup entuziastësh, adhurues të mikrokompjuterëve Raspberry Pi. Më vonë do të njihemi me instalimin dhe mënyrën e përdorimit të tij.

Sot, 74.3% e përdoruesve të telefonave celularë përdorin sistemin operativ **Android**. **Përparësitë e sistemit operativ Android** janë: Një numër tepër i madh aplikacioneve falas (Google Play Store është aplikacion i integruar në vetë sistemin operativ), kompatibil me pajisje të prodhuesve të ndryshëm (Samsung, Huawei, Motorola), kod i hapur, disponon me mjedise zhvillimore të thjeshta për krijimin e aplikacioneve të reja, hapësira e memories lehtë mund të zgjerohet, mundësia e ndarjes në internet, komunikim i thjeshtë me shumë pajisje, liri gjatë konfigurimit dhe përzgjedhje të aplikacioneve. **Mangësi** të këtij sistemi operativ janë: aplikacionet janë më të vështira për t'u krijuar për shkak të përmasave të ndryshme të ekraneve, shpejtësisë më të vogël për shkak të aplikacioneve që punojnë në sfond, siguri më e vogël dhe ndjeshmëri ndaj viruseve.

3.2. Veglat softuerike për konfigurimin e sistemit operativ Windows 10

Pas instalimit të sistemit operativ, por edhe më vonë gjatë përdorimit, është e nevojshme që **i njëjti të konfigurohet sipas kërkesave dhe nevojave personale**. Mjetet softuerike për konfigurim mundësojnë: rritjen e shpejtësisë së punës, menaxhimin e skedarëve, zgjatjen e jetëgjatësisë së komponentëve, marrjen e informacioneve për gjendjen e komponentëve harduerikë dhe softuerike si dhe për drajverët e instaluar, organizimin e të dhënave të ruajtura në hard disk ose memorien SSD, menaxhimin me aplikacioneve që punojnë në sfond, mbështetje për rikuperimin e dokumenteve të fshira ose përmbajtjeve të tyre të mëparshme, siguri dhe mbrojtje më të madhe etj. Vetë sistemi operativ Windows përmban një numër të madh veglash softuerike sistemore për konfigurim, por disa parametra mund të vendosen edhe me përdorimin e programit BIOS.

Të rikujtojmë se **BIOS-i** është softuer i veçantë që mundëson komunikimin ndërmjet komponentëve harduerikë dhe sistemit operativ. Atij i qasemi duke shtypur njërin nga tastat F1, F2, F10, F12 ose Del pas kyçjes së furnizimit. Pas hapjes së BIOS-it, paraqitet menyuja bazë për konfigurim (angl. Setup) në pjesën e sipërme të ekranit. BIOS-i nuk mbështet përdorimin e miut dhe për navigim përdoren tastat e tastierës: lart, poshtë, majtas, djathtas. Menyuja për konfigurim përbëhet nga nënmenytë si: kryesore, avancuar, siguria dhe nënmeny për instalimin e sistemit operativ dhe kyçjen e furnizimit. Shkurtimisht do të njihemi me secilën nga nënmenytë. Të theksojmë se menyuja e konfigurimit është e ndryshme në kompjuterë të ndryshëm dhe varet nga lloji i pllakës amë.

Nën menyuja →
kryesore
(angl. Main) Kjo nënmeny jep informacione për versionin e BIOS-it, llojin e procesorit dhe frekuencën e tij të punës, kapacitetin e memories. Ajo përmban vegla për ndryshimin e kohës dhe datës së sistemit.

Nën menyuja →
e avancuar
(angl. Advanced) Me këtë nënmeny kryhet konfigurimi i pajisjeve memoruese (hard diskut ose pajisjes SSD), portave për kartelën grafike, të zërit ose të rrjetit, pajisjeve USB dhe pajisjeve të tjera që janë të lidhura direkt me pllakën amë. Duhet të jemi shumë të kujdesshëm dhe të përgatitur, sepse konfigurimi i gabuar i disa parametrave në BIOS mund të shkaktojë mosfunksionimin e të gjithë sistemit.

Siguria →
(angl. Security) BIOS-i përmban dy fjalëkalime: atë të supervizorit dhe atë të përdoruesit. Fjalëkalimi i supervizorit quhet fjalëkalim i konfigurimit sepse shërben për të pasur qasje në vetë BIOS-in. Fjalëkalimi i përdoruesit quhet fjalëkalim sistemor dhe shërben për qasje në sistemin operativ. Nënmenyuja për siguri përmban vegla për ndryshimin ose largimin e fjalëkalimit të konfigurimit. Në rast se fjalëkalimi harrohet, atëherë të gjitha fjalëkalimet mund të fshihen duke përdorur një lidhje të shkurtër të veçantë në vetë pllakën amë.

Nënmenyuja →
për konfigurimin
e sistemit
operativ
(angl. Boot) Të rikujtojmë se kjo nënmeny përdoret për renditjen dhe përzgjedhjen e pajisjes për instalimin e sistemit operativ Windows.

Dalje →
(angl. Exit) Kjo nënmeny shërben për ruajtjen e ndryshimeve të bëra dhe daljen nga programi i BIOS-it.

Në pesë vitet e fundit, në kompjuterët e rinj, BIOS-i është zëvendësuar me një program të ri të njohur me shkurtesën UEFI (angl. Unified Extensible Firmware Interface), që do të thotë softuer i unifikuar i zgjeruar i implementuar për ndërfaqen. Grafika është shumë më e avancuar dhe përfshin monitorimin e performancës së kompjuterit: temperaturën, shpejtësinë e ventilatorëve për ftohje, frekuencën e punës së procesorit, etj. (figura 3.2.).

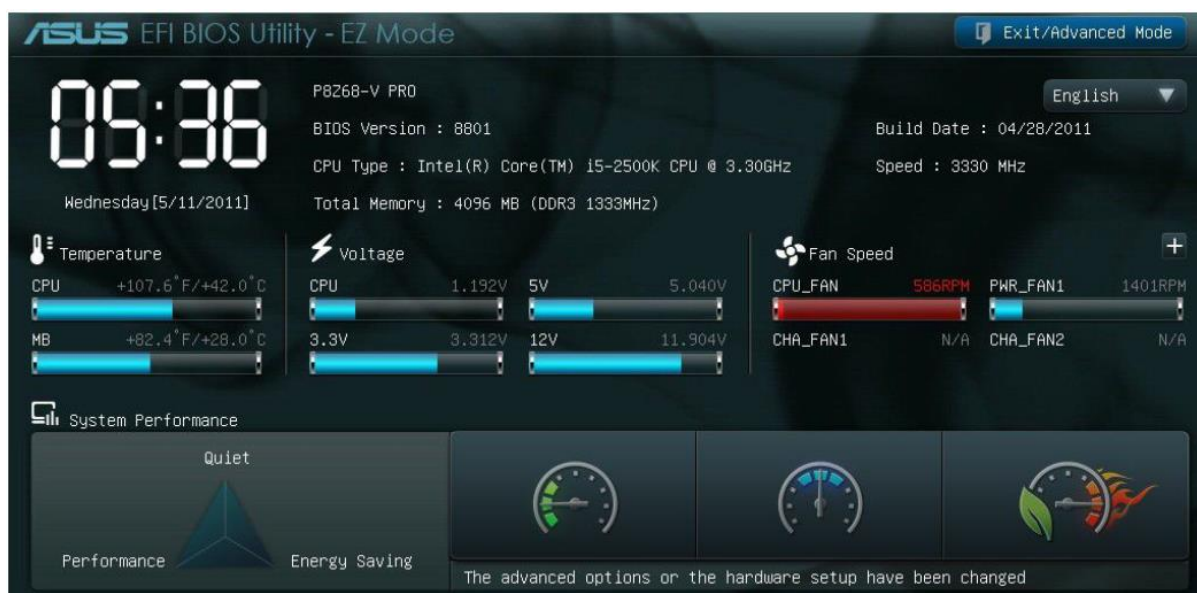


Figura 3.2. Grafika e programit UEFI

Përparësia më e madhe e programit UEFI është ajo se nuk ka kufizime në kapacitetin e hard diskut ose të memories SSD, siç është rasti me BIOS-in, i cili mbështet një kapacitet maksimal prej 2TB.

Vegla më të njohura sistimore për konfigurimin e sistemeve operative Windows janë Control Paneli dhe aplikacioni më i ri Settings. Më poshtë janë numëruar dhe shpjeguar kategoritë në aplikacionin Settings:

- Sistemi** (angl. System) → Me veglat e kësaj kategorie mund të ndryshohet rezolucioni dhe ndriçimi i ekranit, të lirohet hapësirë për ruajtjen e të dhënave, të zgjidhet regjimi i furnizimit me energji, etj.
- Pajisjet** (angl. Devices) → Kjo kategori jep listën e pajisjeve me të cilat mund të komunikojë kompjuteri: si pajisje Bluetooth, ekrane pa tela, ekrane me prekje, tableta grafik, printerë, etj.
- Telefoni** (angl. Phone) → Mundëson qasje në përmbajtjen e telefonit tonë Android ose iPhone.

- Rrjeti dhe interneti (angl. Network & Internet)** → Përdoruesit mund të zgjedhin adapterin e rrjetit Wi-Fi ose adapterin e rrjetit Ethernet, të aktivizojnë ose çaktivizojnë regjimin e punës airplane, të shtohet në një rrjet privat virtual (VPN), etj.
- Personalizimi (angl. Personalization)** → Për çdo profil mund të zgjidhet sfondi, të ndryshohen ngjyrat e temave, të ndryshojë imazhin e mbrojtësit të ekranit (screen saver), të ndryshohet pamja e menysë startuese ose e shiritit të detyrave.
- Aplikacionet (angl. Apps)** → Kjo kategori i kontrollon programet: fshirjen, ndryshimin e programeve startuese (angl. start-up), zëvendësimin e programeve standard, karakteristikat e ekranit për lojëra video (angl. video mode).
- Llogaritë e përdoruesve (angl. Account):** → Vetëm administratori mund të krijojë profile përdoruesish. Në Windows 10, mundësitë për mbrojtje dhe monitorim familjar janë shumë më të mëdha.
- Koha dhe gjuha (angl. Time & Language)** → Konfigurohen koha, data, rajoni, gjuha dhe opsionet për zërim.
- Qasja e lehtë (angl. Ease of Access)** → Kjo kategori ofron mundësi për qasje të ndryshme deri te ekranit, tastiera, miu, oratorit për audio sqarime të elementeve të ekranit.
- Ndihmësi virtual personal (angl. Cortana):** → Kjo është kategori e re dhe paraqet një asistent virtual personal për kërkim në internet dhe shfrytëzim të komponentëve.
- Privatësia, përditësimi dhe siguria (angl. Privacy, Update & Security)** → Shfrytëzohet për përditësimin e sistemit operativ, krijimin e kopjeve rezervë (angl. backup) të dokumenteve personale, skanimin dhe zbulimin e viruse ose softuerit potencial të dëmshëm, mbrojtje ndaj qasjes së paautorizuar dhe abuzimit përmes internetit, si dhe për opsione familjare. Për shembull, në kategorinë Privacy → Camera, mund të përzgjidhen programet që do të kenë qasje në kamerën e integruar të laptopit.

3.3. Programe të dobishme

3.3.1. Programet antivirus

Viruset kompjuterike përhapen nga njëri kompjuter në tjetrin dhe kanë aftësinë të vetëkopjohen, të ndryshojnë dhe të shkatërrojnë skedarët operativë dhe ata të përdoruesve. Këto skedarë janë mikpritës për viruset kompjuterike dhe ata nuk mund të ekzistojnë jashtë tyre. Kur përdoruesi do të hapë ose ekzekutojë ndonjërin nga këta skedarë, kodi i virusit fillon të ekzekutohet. Viruset kompjuterike shkaktojnë ngadalësim të kompjuterit, mbingarkim dhe dëmtim të hard diskut, fshirje, riemërtim dhe zhvendosje të skedarëve, përcjellje në të gjitha adresat nga adresari i postës elektronike, abuzim me fjalëkalimet, qasje në kompjuterin tuaj përmes internetit, etj. Viruset përhapen përmes: pajisjeve memoruese, CD disqeve nga burime të dyshimta, rrjetave kompjuterike, internetit, përmes postës elektronike. Ekzistojnë **disa mënyra për t'u mbrojtur nga viruset kompjuterike**. Nuk duhet të hapen mesazhet dhe reklamata e dyshimta elektronike, është e nevojshme përditësimi i rregullt (angl. update) i sistemit operativ, fshirja e historikut dhe kesh memories së shfletuesit të internetit, skanimi i memorieve të jashtme para përdorimit të tyre, skanim të hard diskut pas instalimit të një programi të ri. Për skanimin e memories është e nevojshme paraprakisht të instalohet një program antivirus. Antiviruset janë pako softuerike të cilat përbëhen nga **programe kompjuterike që mundohen t'i zbulojnë dhe t'i largojnë viruset kompjuterike** dhe programet e tjera të dëmshme. Programet antivirus përmbajnë baza për viruset dhe vazhdimisht përditësohen për të zbuluar viruse të rinj. Programi antivirus Windows Defender, si pjesë e sistemit operativ, nuk është i mjaftueshëm për të mbrojtur kompjuterin nga viruset dhe kërcënimet në internet.



Figura 3.3. Programet antivirus dhe logot e tyre

Sot, zgjedhja e programeve antivirus është shumë e madhe. Disa prej tyre janë falas, por nëse duam mbrojtje premium, është e nevojshme që programi të

paguhet. Nortoni është njëri prej provajderëve më të njohur në botë dhe njëri prej programeve antivirus më të sigurta. Me një tarifë vjetore të caktuar dhe me një llogari përdoruesi të krijuar paraprakisht, programi mund të shkarkohet nga faqja e tyre zyrtare (<https://us.norton.com/antivirus>) dhe injëti të instalohet.

Programe antivirus pa pagesë, me rezultate të mira gjatë testimit janë të shumta dhe ato janë paraqitur në tabelën 3.1, së bashku me linqet për shkarkimin e tyre.

Programi antivirus	Linku për shkarkim
Avira	https://www.avira.com
AVG	https://www.avg.com
Bitdefender	https://www.bitdefender.com/solutions/free.html
Panda	https://www.pandasecurity.com/en/homeusers/free-antivirus/
TotalAv	Free Antivirus 2023 – Download Free Antivirus Software & Internet Security - TotalAV
Kasperski	https://www.kaspersky.com/free-antivirus

Tabela 3.1. Programe antivirus

Përveç mbrojtjes dhe detektimit të viruseve, këto programe ofrojnë edhe shërbime të tjera si: qasje VPN (angl. Virtual Private Network), mbikëqyrje nga prindërit, menaxhim të fjalëkalimeve, mbrojtje të ueb kamerës, mbrojtje të identitetit dhe të dhënave personale, ruajtje të të dhënave në cloud etj. Për shembull, me qasjen VPN sigurojmë anonimitet, adresa jonë IP zëvendësohet me adresën e serverit, fshihet lokacioni jonë, ndërsa të dhënat për transmetim kodohen, duke i bërë të paaksesueshme. Disa nga këto funksione mund të ngadalësojnë punën e kompjuterit, kështu që të gjitha funksionet nuk është e nevojshme të instalohen. Karakteristikë e programeve antivirus të reja është se ato mund të përpunojnë të dhënat në cloud dhe në këtë mënyrë nuk ngarkohen komponentët harduerike të kompjuterit tonë.

3.3.2. Programet për kompresimin e skedarëve

Kompresimi i skedarëve është proces për zvogëlimin e hapësirës fizike përkatësisht kapacitetit të memories, të nevojshme për ruajtjen e tyre. Në përputhje me një algoritëm të caktuar zvogëlohet numri i bitëve, të cilët shërbejnë për paraqitjen binare të të dhënave dhe në këtë mënyrë zvogëlohet madhësia e skedarëve, e shprehur me numrin e bajtave. Për shembull, në një skedar tekstual, nëse një fjalë e caktuar përsëritet disa herë, atëherë e njëjta mund të dërgohet vetëm një herë, por bashkë me numrin e përsëritjeve. Me kompresimin **arrihet kursim i hapësirës memoruese dhe kursim të kohës** që është e nevojshme për transferimin e skedarit. Ekspansioni ose dekompresimi është operacioni invers që ka për qëllim rikthimin e përmbajtjes origjinale të skedarëve. Nëse bëhet fjalë për kompresim pa humbje të të dhënave, atëherë dekompresimi do të japë të dhëna identike me origjinalin, ndërsa në rastin e kompresimit me humbje të të dhënave, atëherë nuk fitohen të dhëna origjinale.

Në varësi të llojit të algoritmit të përdorur, skedarët e kompresuar mund të jenë me format të ndryshëm, me ç'rast kanë prapashtesa të ndryshme si: .zip, .rar, .7zip, .zipx, .arc dhe të tjera.

Programet më të njohura për kompresim për sistemin operativ Windows janë: WINZIP, WINRAR, 7-Zip, PeaZip, BandZip dhe të tjera. WINZIP dhe WINRAR nuk janë falas, ndërsa 7-Zip, PeaZip dhe BandZip mund të shkarkohen falas.

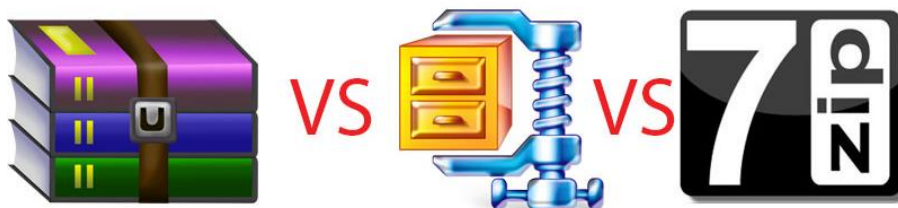


Figura 3.2. Programe për kompresim dhe simbolet e tyre

Ato dallohen për nga shkalla e kompresimit, madhësia dhe lloji i skedarëve, mbështetja për sisteme të ndryshme operative. Të gjitha programet për kompresim janë kompatibil me njëri tjetrin. Për shembull, me programin WINZIP mund të dekompresohet një dokument RAR dhe anasjelltas, por përsëri gjithçka varet nga formati i skedarit dhe nga versioni i programit. Çdo program ka veçoritë e veta. Programi WINZIP kompreson çdo skedar veçmas pa përzierje të të dhënave dhe mund të përdoren algoritme të ndryshme në varësi të llojit të skedarit. Programi WINRAR i kompreson të gjitha skedarët e selektuar njëherazi, një proces që kërkon më shumë kohë, por fitohet një shkallë më e lartë e kompresimit. Programet WINZIP, WINRAR dhe BandZip janë kompatibil vetëm me sistemin operativ Windows, ndërsa 7-Zip dhe PeaZip edhe me sistemin operativ Linux.

Në tabelën 3.2. janë dhënë rezultatet nga testimi i kryer mbi pesë programe kompresimi. Si të dhëna hyrëse janë përdorur 42 skedarë me përmbajtje të ndryshme, të vendosura në katër dosje, me kapacitet total prej 303 MB. Parametri më i rëndësishëm është shkalla e kompresimit e shprehur në përqindje. Atë e fitojmë me formulën:

$$\frac{\text{madhësia e skedarit origjinal}}{\text{madhësia e skedarit të kompresuar}} \cdot 100$$

Madhësia e shkallës së kompresimit varet jo vetëm nga lloji i programit, por edhe nga lloji i formatit të skedarit. Të theksojmë se shkalla e kompresimit varet edhe nga lloji i skedarit: tekstual, grafik, muzikor apo video. Gjithashtu, nga tabela mund të përfundohet se shpejtësia e kompresimit dhe dekompresimit varet në masë të madhe nga lloji i memories së përhershme (SSD apo hard disk) nga ku e marrim skedarin origjinal dhe ku e ruajmë skedarin origjinal. Pavarësisht se për cilin lloj të kompresimit të përcaktohem, me rëndësi është

aplikimi i saj sepse pa kompresim disa skedarë thjesht nuk mund të transferohen ose të arkivohen si skedarë rezervë (angl. backup).

Programi dhe formati i skedarit	Shpejtësia e kompresimit për SSD (sek.)	Shpejtësia e kompresimit për HDD (sek.)	Madhësia e skedarit të kompresuar (MB)	Shkalla e kompresimit (%)	Shpejtësia e dekompresimit për SSD (sek.)	Shpejtësia e dekompresimit për HDD (sek.)
PeaZip, ZIP	12,1	16,0	97,70	32,24%	1,6	13,4
PeaZip, 7Z	39,6	42,2	73,60	24,69%	1,0	13,0
PeaZip, ARC	17,2	18,3	71,70	23,66%	7,1	16,8
7-Zip, ZIP	11,9	16,0	97,70	32,24%	1,2	13,1
7-Zip, 7Z	38,0	40,3	73,60	24,29%	0,9	13,0
WinRar, ZIP	3,0	6,0	100,00	33,00%	1,0	76,6
WinRar, RAR	13,8	25,5	80,40	26,53%	1,0	18,9
Bandizip, ZIP	3,2	7,0	101,00	33,33%	2,0	13,4
Bandizip, 7Z	63,4	69,7	72,80	24,03%	5,2	17,0
WinZip, ZIP	20,5	26,0	97,10	32,05%	2,0	12,3
WinZip, ZIPX	34,7	37,2	70,70	23,33%	46,2	47,4

Tabela 3.2. Krahasimi i programeve për kompresim me formate të ndryshme të skedarëve

3.3.3. Programet për matjen e performancës së kompjuterit

Programet për matjen e performancës së kompjuterit i mundësojnë edhe një përdoruesi të zakonshëm të vlerësojë nëse kompjuteri i tij funksionon si duhet. Programet që do të shqyrtojmë janë shumë të thjeshta dhe praktike. Të gjitha këto vegla mund të shkarkohen dhe instalohen falas, ndërsa përdoruesi vetë mund të zgjedhë cilat vegla i nevojiten, pa qenë e domosdoshme t'i përdorë të gjitha.

Në vetë sistemin operativ Windows 10 ekziston një funksion Performance Monitor, që mundëson matjen e performancës së kompjuterit. I njëjti mund të hapet nëse në shiritin e kërkimit shkruajmë "**Performance Monitor**". Ky program ofron ndjekjen e shkallës së përdorimit të procesorit, memories RAM, hard diskut dhe lidhjes në rrjet në kohë reale, por ekziston edhe mundësia për mbledhjen e të dhënave dhe analizën e tyre. Edhe pse është pjesë e sistemit operativ, megjithatë, ky funksion është për përdoruesit më të avancuar dhe kërkon njohuri të mira për karakteristikat e komponentëve harduerikë.

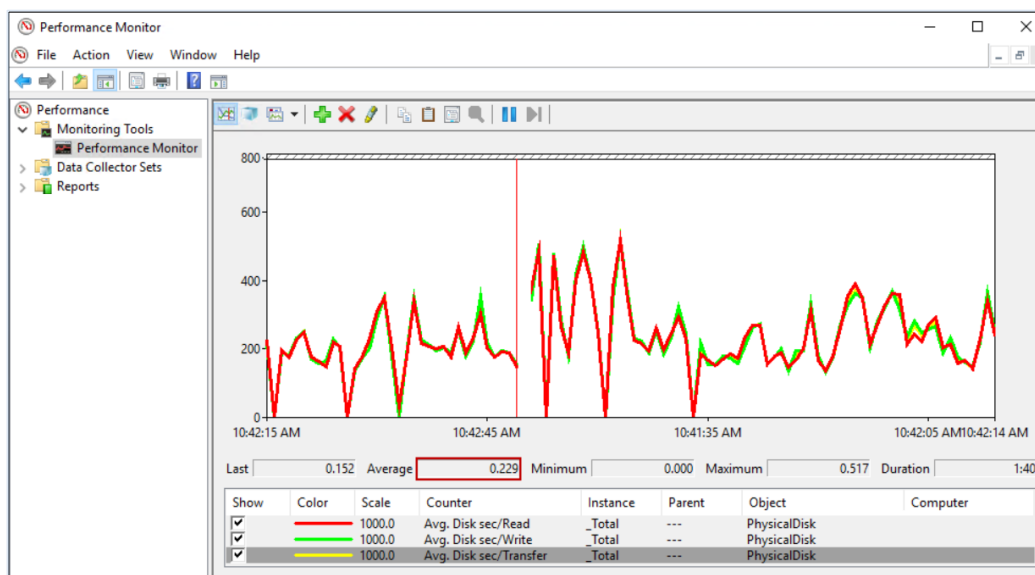


Figura 3.5. Paraqitja grafike e punës së procesorit me funksionin Performance Monitor

Aplikacioni **CPU-Z** mund të shkarkohet dhe instalohet falas nga faqja zyrtare e shpërndarësit, CPUID. Ky aplikacion monitoron performancën e njësisë qendrore të procesorit, por edhe të komponentëve tjera harduerike, siç janë memoria RAM, pllaka amë dhe kartela grafike. Përdoruesi i zakonshëm mund të marrë informacione për llojin e procesorit, prodhuesin e tij, shpejtësinë e punës për çdo bërthamë, madhësinë e kesh memories dhe kushinetës.



Figura 3.6. Matja e performancave të kompjuterit me aplikacionin CPU-Z

SpeedFan, siç tregon edhe vetë emri i saj, është program që i mundëson përdoruesit të monitorojë shpejtësinë e rrotullimit të ventilatorëve në brendësi të

shtëpizës së kompjuterit dhe temperaturën e procesorit, hard diskun, kartelën grafike dhe pajisjeve të tjera të jashtme. Programi SpeedFan lejon rregullimin e shpejtësisë së rrotullimit të ventilatorëve dhe kjo është vegël shumë e dobishme nëse kemi të bëjmë me shtëpizë me hapësirë të kufizuar.

CrystalDiskMark dhe CrystalDiskInfo janë dy programe të ndryshme nga i njëjti shpërndarës për matjen e performancës së hard diskut, siç janë shpejtësia e leximit dhe shkrimit dhe organizimin e ruajtjes së të dhënave në secilin disk në vaçanti.

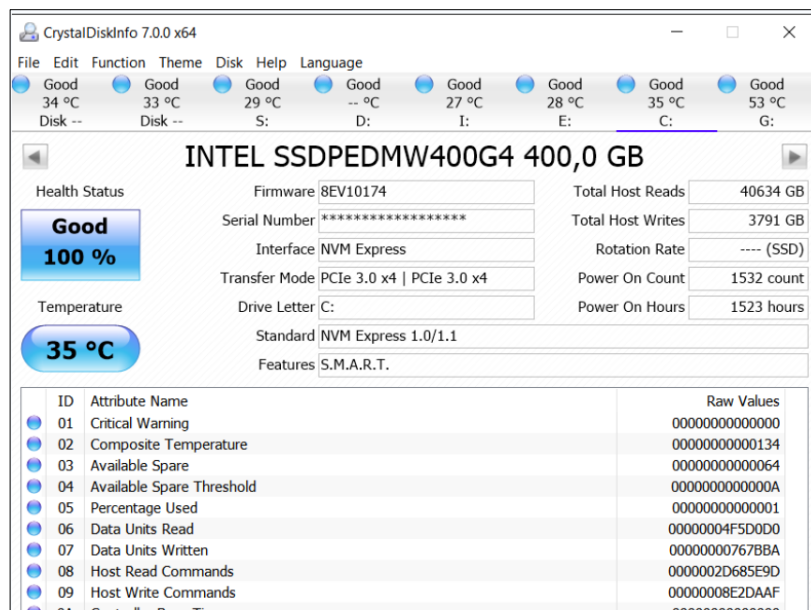


Figura 3.7. Matja e performancave të hard diskut me aplikacionin CrystalDiskInfo

MSI Afterburner është vegël shumë e dobishme e cila mundëson matje të lehtë dhe të thjeshtë të performancave të kartelës grafike në veçanti, nëse përdoruesi dëshiron të rrisë frekuencën e punës së kartelës grafike (angl. overclock). Me aplikacionin MSI Afterburner, mund të rregullohen sinjali i taktit të kartelës grafike, temperatura dhe furnizimi maksimal me energji, shpejtësia e ventilatorit dhe shumë parametra të tjerë.

Programet për matjen e performancave të kompjuterit i mundësojnë përdoruesit të shfrytëzojnë maksimalisht burimet harduerike dhe të marrë masa paraprake në kohë në rast problemesh.

3.4. Programe aplikative

Programet aplikative u mundësojnë përdoruesve të kryejnë detyra të caktuara duke përdorur kompjuterin. Duke qenë se janë të destinuara për shfrytëzuesit e fundit, ato quhen programe për përdoruesit. Përveç kompanive të specializuara për softuer, kreator të programeve aplikative mund të jenë edhe vetë përdoruesit. Sipas qëllimit, programet aplikative mund të ndahen në disa

grupe. Më poshtë janë numëruar këto grupe, qëllimi i tyre dhe përzgjedhja e programeve.

Softuer për përpunimin e tekstit	→	<ul style="list-style-type: none"> • Futja e tekstit. Modifikimi i tekstit ekzistues, • Rregullimi (formatimi) i tekstit 	→	<ul style="list-style-type: none"> • Notepad • WordPad • MS Word • Word Perfect • OpenOffice – Writer
Softuer për punë me tabela	→	<ul style="list-style-type: none"> • Krijimi i tabelave për futjen, llogaritjen dhe analizën e të dhënave. • Krijimi i grafikoneve dhe paraqitja grafike e të dhënave • Formatimi, përzgjedhja dhe filtrimi i të dhënave 	→	<ul style="list-style-type: none"> • MS Excel • OpenOffice – Calc • Lotos • QuatroPro
Softuer për krijimin e prezantimeve	→	<ul style="list-style-type: none"> • Prezantim të produkteve dhe shërbimeve • Mbajtja e mësimi dhe ligjëratave • Krijimi i reklamave dhe efekteve multimediale 	→	<ul style="list-style-type: none"> • MS PowerPoint • OpenOffice – Impress • Adobe Flash • Adobe Dreamweaver
Softuer për përpunimin e imazheve	→	<ul style="list-style-type: none"> • Dizajnimi i logove • Krijimi i plakatave dhe broshurave • Dizajnimi i librave dhe revistave 	→	<ul style="list-style-type: none"> • Grafika vektoriale (CorelDraw, Adobe Illustrator) • Grafika raster (Adobe PhotoShop, CorelPaint, Paint)
Softuer për shfletues interneti	→	<ul style="list-style-type: none"> • Qasje e shpejtë në faqe të ndryshme interneti, • Shikimi i skedarëve PDF, • Mundësia për riprodhim videosh, lojërash dhe animacionesh. • Mbrojtje nga ueb lokacionet e rrezikshme dhe shfletim anonim 	→	<ul style="list-style-type: none"> • Opera • Chrome • Mozilla Firefox • Internet Explorer • MS Outlook • Outlook express

Programet që kryejnë detyra të ngjashme quhen **paketa programore**. Më të përdorurat janë paketat për punë zyre. Paketa Microsoft Office është kompatible me sistemin operativ Windows, ndërsa paketa OpenOffice është kompatible me sistemin operativ Linux, si p.sh. versioni Edubuntu. Për punë me paketën Microsoft Office nevojitet licencë.

Në vitin 2013 në internet u shfaq platforma **Microsoft Office 365**. Ajo është aplikacion në internet e cila në vete përfshin të gjitha programet për punë zyre. Në vitin 2020, të gjithë nxënësit në Maqedoni morën profilin e tyre në këtë platformë për realizimin e mësimit në distancë, përmes programit Microsoft Teams. Nëse kyçemi në platformën Microsoft Office 365 me profilin shkollor, mund të shkarkojmë dhe instalojmë falas paketën Microsoft Office në kompjuterin tonë. Duhet të shtypim butonin **Install Office** dhe të zgjedhim opsionin Office 365 apps.

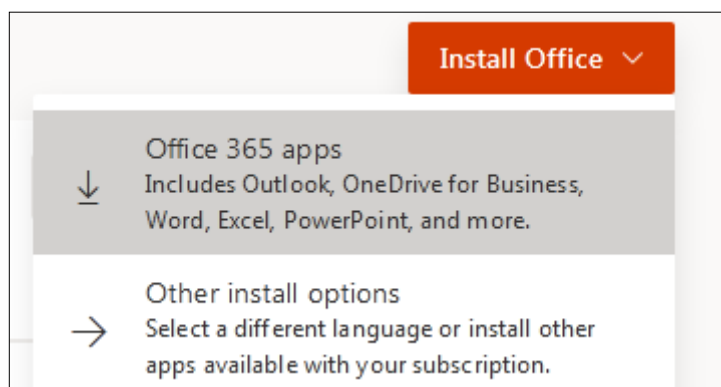


Figura 3.8. Dritarja për instalimin e paketës Microsoft Office 365

Më pas shfaqet një dritare e re, e cila na udhëzon nëpër procesin e instalimit.

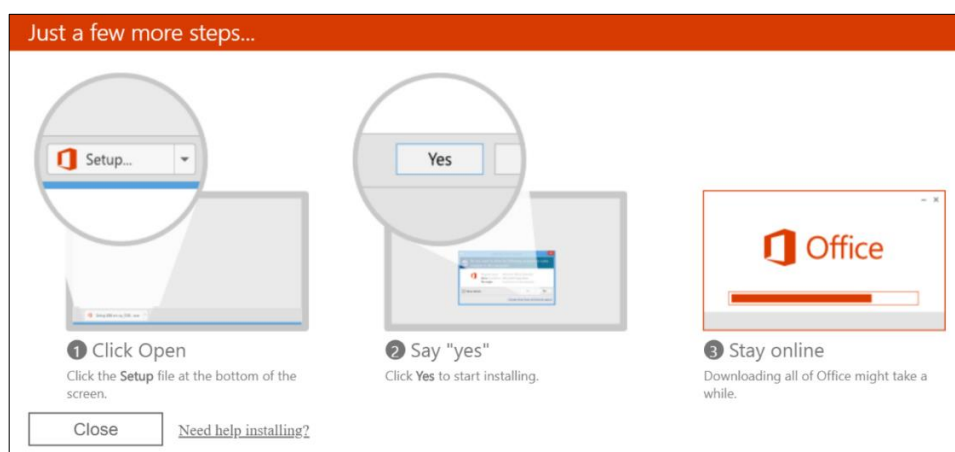


Figura 3.9. Hapat gjatë instalimit të paketës Microsoft Office 365

Fillon procesi i instalimit, siç është paraqitur në figurën 3.10. Gjatë instalimit të kësaj pakete, mund të zgjedhim një nga dy versionet, 32-bitësh dhe 64-bitësh. Nëse duam ta zëvendësojmë versionin 32-bitësh me versionin 64-bitësh, atëherë duhet të denstalojmë më parë versionin 32-bitësh.

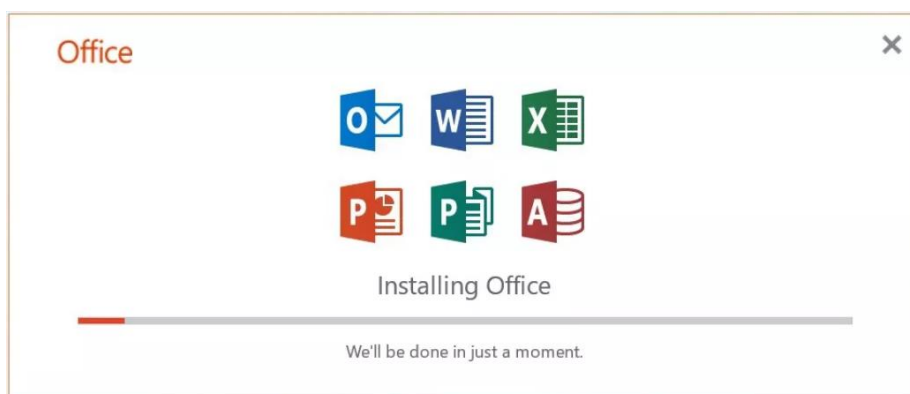


Figura 3.10. Instalimi i paketës Microsoft Office 365

Pavarësisht nëse përdorim aplikacionin në internet ose e kemi instaluar atë në kompjuterin tonë, paketa Microsoft Office 365 ofron shumë vegla dhe mostra të gatshme që do të na ndihmojnë shumë në krijimin e dokumenteve tona.

Përfundime

Sistemi operativ është i njohur si softuer menaxhues - kontrollues. Ai përmban programe për shpërndarjen e burimeve harduerike, kontrollin e operacioneve hyrëse-dalëse, menaxhim me memorien, menaxhim me të dhënat, përkthimin e gjuhëve programuese.

Përparësitë e **sistemit operativ Windows** janë: thjeshtësia në përdorim (angl. user friendly), mbështetje më e madhe për lidhjen me pajisje periferike, qasje në softuerin aplikativ, instalim i automatizuar (angl. plug and play), performanca më të mira për lojërat kompjuterike, kompatibilitet me shumicën e ueb faqeve. mangësi janë: nevoja për konfiguracione kompjuterike të shtrenjta, kod i mbyllur, siguri e dobët në lidhje me sulmet në internet dhe mbrojtjen administrative, ndjeshmëri ndaj viruseve kompjuterike, nevojitet licencë, mbështetje e dobët teknike nga ekspertë dhe shpesh here për të rritur shpejtësinë, është e nevojshme riinstalimi i sistemit operativ.

Kod i hapur ose i lirë do të thotë që të gjithë përdoruesit kanë qasje në kodin burimor dhe mund ta modifikojnë atë sipas nevojave të tyre.

Sistemi operativ **Android** dhe **Raspbian** nga kompania Google janë shembuj të zbatimit dhe zgjerimit të bërthamës së sistemit operativ Linux.

Përparësitë e sistemit operativ Android janë: numër shumë i madh i aplikacioneve falas, kompatibiliteti me pajisje të prodhuesve të ndryshëm, kodi i hapur, disponon me mjedise zhvillimore të thjeshta për krijimin e aplikacioneve të reja, hapësira memoruese lehtë zgjerohet, mundësia e ndarjes në internet,

komunikim i thjeshtë me shumë pajisje, liri në konfigurim dhe zgjedhjen e aplikacioneve. Disavantazhe të këtij sistemi operativ janë: aplikacionet janë më të vështira për t'u zhvilluar për shkak të dimensioneve të ndryshme të ekraneve, shpejtësi më e ulët për shkak të aplikacioneve në sfond, siguri më e vogël dhe ndjeshmëri ndaj viruseve.

Antiviruset janë paketa softuerike që përbëhen nga programe kompjuterike që përpiqen të gjejnë dhe të largojnë viruset kompjuterikë dhe programet tjera të dëmshme. Programe antivirus falas me rezultate të mira janë: Avira, Bitdefender, Panda, TotalAv, Kaspersky dhe të tjerë.

Programe më të njohura për kompresimin e skedarëve në sistemin operativ Windows janë: **WINZIP, WINRAR dhe version më i ri 7-Zip**. Me kompresim arrihet kursimi i hapësirës memoruese dhe kursimi i kohës që është e nevojshme për transferimin e skedarët.

Programe për matjen e performancave të kompjuterit janë: CPU-Z, SpeedFan, MSI Afterburner, CrystalDiskMark, CrystalDiskInfo dhe Performance Monitor në përbërje të vetë sistemit operativ Windows.

Në kategorinë e programeve për përdoruesit përfshihen: softueri për përpunimin e tekstit, softueri për punën me tabela, softueri për krijimin e prezantimeve, softuer për përpunimin e imazheve, softueri për shfletimin e internetit.

Instalimi i sistemit operativ Windows mund të realizohet në mënyra të ndryshme, përmes një moduli memorie (USB ose DVD) dhe përmes internetit (PHE boot). Nëse shfrytëzojmë një modul memorie, është e nevojshme që pas lidhjes së modulit dhe kyçjes së kompjuterit, duhet të **aktivizojmë programin BIOS**.

Çelësi i produktit është kod prej 25-shifrave me të cilin konfirmohet licenca. Këtë kod e rekomandon Microsoft së bashku me dorëzimin e sistemit operativ Windows, ose mund të dërgohet përmes e-mailit nëse programi për instalim është shkarkuar nga interneti.

Ekzistojnë dy mundësi për instalimin e sistemit operativ Windows: **Upgrade ose Custom**. Opsioni i parë do të thotë përditësim i sistemit operativ ekzistues Windows. Ky instalim është falas, nuk kërkon çelës produkti, me ç'rast nuk fshihen të dhënat e përdoruesit.

Windows 10 përmban **dy vegla për konfigurim**: Control Panelin dhe aplikacionin e ri Settings.

Kategoritë në aplikacionin Settings janë: System, Devices, Phone, Network&Internet,

Personalization, Apps, Account, Time & Language, Ease of access, Cortana dhe Privacy, Update & Security.

Mënyrat për **optimizimin e sistemit operativ**, gjegjësisht rritjen e shpejtësisë së punës së kompjuterit janë: largimi i aplikacioneve të pashfrytëzuara dhe i aplikacioneve vetëstartuese, lirimi i hapësirës së memories në hard disk dhe defragmentimi i saj, përdorimi i programeve antivirus, përditësimi i sistemit operativ dhe rritja e faqeve virtuale.

Makina virtuale është sistem kompjuterik mik (guest) i cili i përdor resurset e kompjuterit nikoqir (host). Në kompjuterin nikoqir do të instalohet sistemi operativ Windows, ndërsa në makinën virtuale, gjegjësisht kompjuterin mik do të instalohet sistemi operativ Ubuntu.

Në dritaren Create Virtual Machine, në fushën Name futet emri i makinës virtuale (p.sh. Ubuntu), në fushën Type futet lloji i sistemit operativ që do të instalohet në makinën virtuale (Linux), ndërsa në fushën Version futet versioni i sistemit operativ (32-bitësh ose 64-bitësh).

Pyetje dhe detyra

1. Cilat programe futen në përbërje të sistemit operativ?

2. Cilat janë përparësitë dhe mangësitë e sistemit operativ Windows?

3. Cilat janë fillimet e zhvillimit të sistemit operativ Linux?

4. Përmend shembuj të përdorimit të sistemit operativ Linux!

5. Sqaro procedurën për zgjedhjen e pajisjes për instalimin e sistemit operativ Windows 10!

6. Çfarë përfaqëson i ashtuquajtura çelës i produktit për sistemin operativ Windows?

7. Shpjego mundësitë e instalimit Upgrade dhe Custom për sistemin operativ Windows 10!

8. Cilat janë dy aplikacionet për konfigurimin e sistemit operativ Windows 10?

9. Numëro kategoritë në aplikacionin Settings!

10. Cila kategori e aplikacionit Settings përdoret për të ndryshuar rezolucionin dhe ndriçimin e ekranit?

11. Për çfarë shërben kategoria Account në aplikacionin Settings?

12. Në cilën kategori të aplikacionit Settings gjenden programet për të zbuluar viruse dhe mbrojtjen nga hyrjet e paautorizuara?

13. Cili është qëllimi kryesor i optimizimit të sistemit operativ?

14. Çfarë nënkuptojmë me nocionin "aplikacione startuese" (angl. start-up)?

15. Shpjego procedurën për defragmentimin e hard diskut?

16. Në cilën kategori përditësohen driverat për pajisjet periferike?

17. Shpjego procedurën për resetimin e sistemit operativ Windows 10 pa humbur informacionet e përdoruesit?

18. Cilat shërbime shtesë fitohen me programin antivirus

19. Përmend disa shembuj të përdorimit të programeve për kompresimin e skedarëve!

20. Cilat karakteristika të kompjuterit mund të ndiqen me përdorimin e aplikacionit CPU-Z?

21. Bëni një krahasim mes tre programeve për kompresim WINZIP, WINRAR dhe 7-Zip!

22. Numëro llojet e programeve për përdoruesit dhe jepni nga një shembull për secilin!

4. Mikrokompjuteri në pllakë

4.1. Mikrokompjuteri në pllakë - Arduino

Në temën e parë, Bazat e sistemit kompjuterik, theksuam se mikrokompjuterët më të shumtë janë kompjuterët personalë dhe mikrokompjuterët në pllaka. Mikrokompjuteri Arduino është **platforma mikrokontrolluese** e parë e madhe me kod programi të hapur (angl. open source). Kjo do të thotë qasje e lirë në shumë programe dhe skedarë të gatshëm për përdoruesit, të cilët mund të shkarkohen falas nga interneti dhe të ndryshohen për të përmirësuar performancën e pajisjeve elektronike.

Ne do të njihemi me komponentët harduerike të mikrokompjuterit në pllakë Arduino Uno R3. Do të studiojmë arkitekturën e tij, njësitë hyrëse-dalëse, mënyrën e lidhjes, veglat në mjedisin zhvillimor dhe disa programe të gatshme që do t'i implementojmë në memorien programuese. Në hyrjen e mikrokompjuterit Arduino mund të lidhen: tasterë, ndërprerës, tastiera të gatshme, sensorë (për temperaturë, presion, rrjedhë, lëvizje, etj.). Daljet mund të lidhen me pajisje dalëse të ndryshme siç janë: diodat led, llamba, sirena, motorë, ekrane. Për shembull, nëse do të ndizet diode led ose motori do të fillojë të punojë, varet nga ajo nëse është shtypur tasteri ose nëse sensori ka zbuluar një ndryshim të një madhësie fizike. Mburojet shtesë (angl. Shields) janë pllaka elektronike që montohen mbi mikrokompjuterin kryesor Arduino, me çka zgjerohen mundësitë e tij bazë dhe mund të ekzekutojnë funksione plotësuese si kontrolli i motorëve, lidhja me sensorë, komunikimi pa tel, etj.

Për programimin e Arduino përdoret gjuha programuese C/C++, një modifikim i C dhe C++. Ekziston një numër i ndryshimeve të instruksioneve për Arduino dhe për kompjuterin personal, pasi Arduino manipulon me pajisjet hyrëse-dalëse. Po ashtu, vetë programimi kërkon njohuri të mira të harduerit, veçanërisht në lidhje me sensorët dhe njësitë ekzekutuese.

Ekzistojnë më tepër se 15 modele të ndryshme të mikrokompjuterëve Arduino në pllakë, por aktualisht më të popullarizuar janë modelet Arduino Uno R3, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due. Shkurtimisht do t'i shpjegojmë dallimet ndërmjet tyre [3].

- Versioni i tretë i **Arduino Uno** (Rev3 ose R3) është mikrokompjuteri më i mirë për fillestarët dhe do ta shpjegojmë dhe përdorim gjatë realizimit të ushtrimeve praktike. Ky mikrokompjuter në pllakë ka një mikrokontrollues 8-bitësh ATmega 328, 14 pina digjital hyrje-dalje (nga të cilët gjashtë mund të përdoren si dalje me impulse të moduluara me gjerësi), 6 hyrje analoge, një konektor USB dhe hyrje për furnizim me energji. Ai mund të furnizohet me bateri ose nëpërmjet një konvertori AC/DC. Të gjitha modelet e tjera të Arduino-s do t'i krahasojmë me Arduino Uno R3.
- **Arduino Nano**. Arduino Uno R3 dhe Arduino Nano kanë procesorë, kapacitete memorie dhe pina të njëjtë. Por ekzistojnë tre ndryshime kryesore. Arduino Nano është me përmasa dy herë më të vogla, çka është avantazh gjatë montimit. Së dyti Arduino Nano ka pina mashkull nga të dyja anët dhe mund të vendoset lehtësisht në një protoplakë. Pasi që Arduino Nano ka mini USB, ai nuk mund të lidhet me shumë pajisje periferike, për dallim nga Arduino Uno R3, i cili ka konektor USB të llojit B.

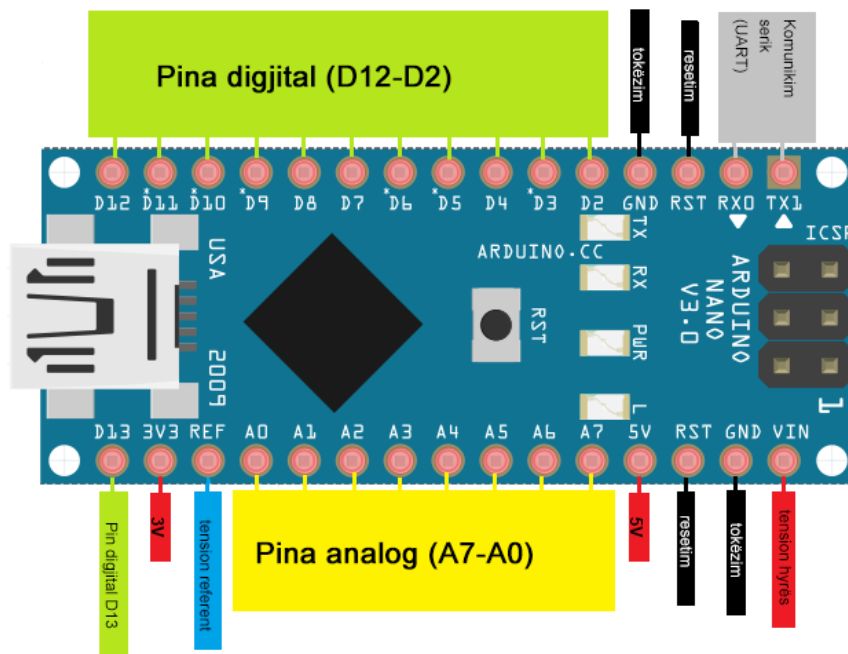


Figura 4.1. Diagrami i pinave i platformës Arduino Nano

- **Arduino Pro Mini** është sa një e gjashtë e madhësisë së Arduino Uno R3. Tensioni i furnizimit është më i ulët dhe është 3.3 V, ndryshe nga 5 V i Arduino Uno R3. Ka dy herë frekuencë pune më të vogël dhe nuk ka mundësi për komunikim në seria. Megjithatë, avantazh i madh është konsumi i tij i ulët i energjisë, gjë që e bën të përshtatshëm për vendosje në vende me qasje të vështirë.
- Arduino Mega 2560 është si vëlla më i vjetër i Arduino Uno R3. Diagrami i pinave të Arduino Mega 2560 është paraqitur në figurën 4.2. Arduino Mega 2560 ka 54 pina digjital hyrje-dalje, nga të cilat 15 mund të përdoren si dalje të impulseve të moduluara me gjerësi. Gjithashtu ka 14 hyrje analoge. Mundëson komunikim serial me katër pajisje njëkohësisht për shkak të katër portave të

integruara universale asinkrone për marrje dhe dërgim të të dhënave (angl. UART - Universal Asynchronous Receiver-Transmitter).

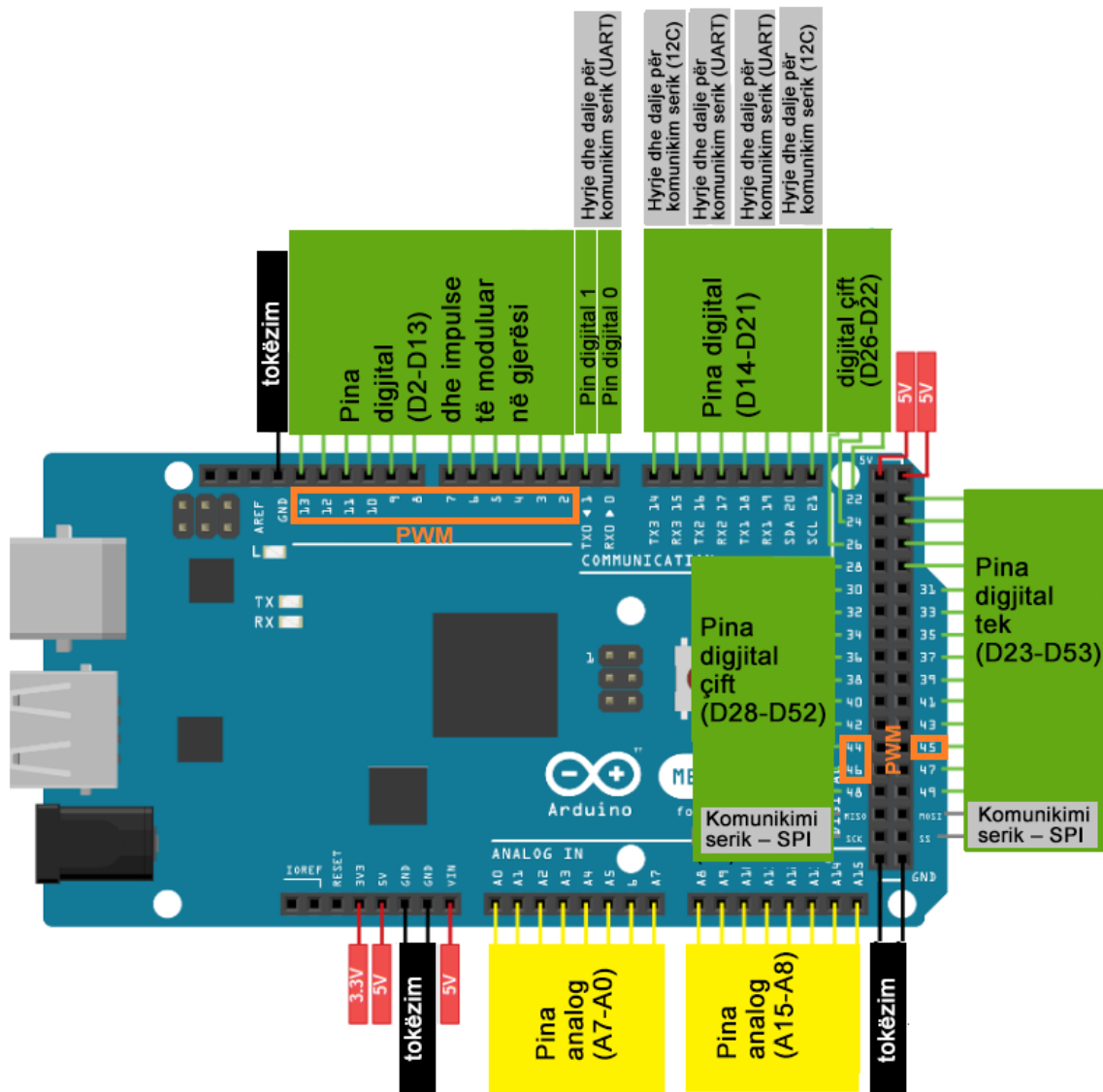


Figura 4.2. Diagrami i pinave të Arduino Mega 2560

Ka 256 KB memorie programimi, 8 KB memorie RAM dhe 4 KB memorie për të dhënat. Vërejmë se memorja e punës e Arduino Mega është tetë herë më e madhe se ajo e Arduino Uno R3.

- **Arduino Leonardo.** Dallimi kryesor qëndron te procesorët dhe numri paksa i ndryshëm i pinave. Procesori i Arduino Leonardo nuk mund të zëvendësohet në rast të një defekti, por ai ka mbështetje USB për lidhje me pajisje të tjera, ndërsa Arduino Uno R3 i duhet kontrollues i veçantë. Por, nga ana tjetër, Arduino Leonardo nuk është i kompatibil me shumë shtesa të Arduino Uno R3.
- **Arduino Due** ka një procesor shumë më të fuqishëm se Arduino Uno R3, por çmimi i tij është pothuajse dyfish më i lartë. Për shembull, frekuenca e punës e

Arduino Due është tetë bitëshe, ndërsa Arduino Uno është 8 bitëshe. Mund të ekzekutojë 104 milionë instruksione në një sekondë. Për këtë arsye, Arduino Uno kryesisht përdoret për kryerjen e llogaritjeve me numra decimal, të njohur ndryshe edhe si numra me presje të lëvizshme (angl. floating-point numbers).

4.2. Pjesët përbërëse të mikrokompjuterit në pllakë Arduino Uno

Fillimisht, do të njihemi me harduerin e mikrokompjuterit Arduino Uno R3 në pllakë: pjesët përbërëse, sensorët, aktuatorët, komponentët elektronikë për lidhjen e tyre dhe shtesave për rritjen e funksionalitetit. Më pas, do të njihemi me grupin e instruksioneve, softuerin dhe mjedisin zhvillimor të mikrokompjuterit Arduino Uno R3 në pllakë.

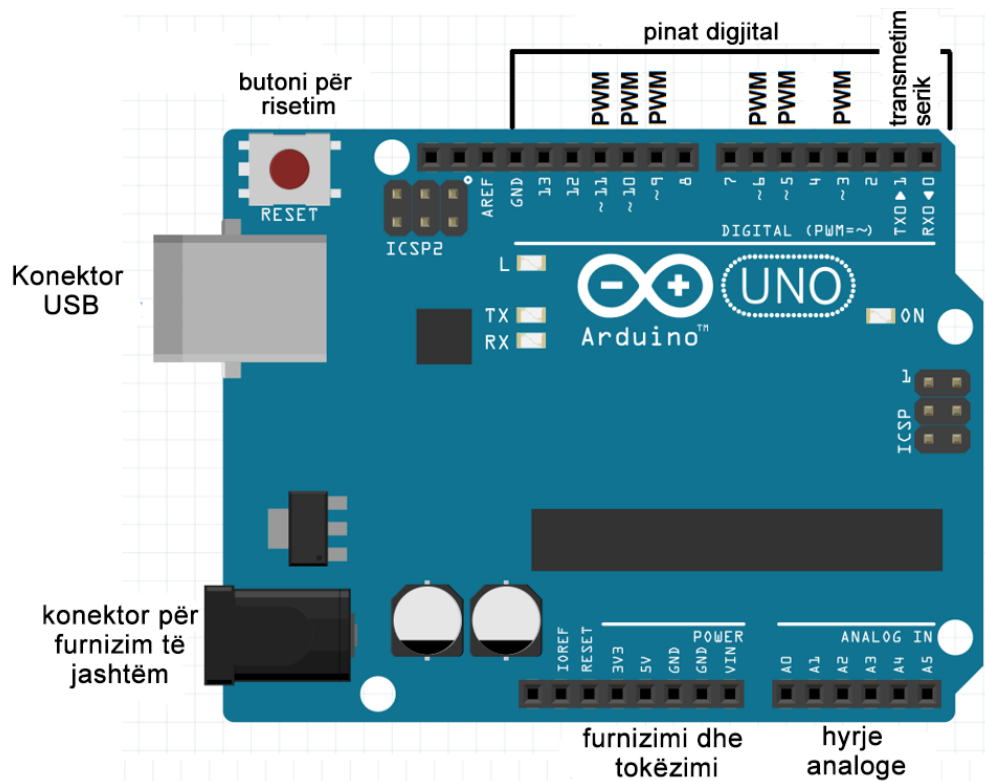


Figura 4.3. Pjesët përbërëse të Arduino Uno R3 (pamje nga lart)

Komponenti kryesor në Arduino Uno R3 është mikrokontrolluesi ATmega328 nga prodhuesi Atmel. Ky mikrokontrollues është çip 8-bitësh me 28 pina, i ndërtuar në teknologji RISC. Memoria programore është me kapacitet prej 32 KB, memoria e të dhënave EEPROM është prej 1 KB dhe memoria e brendshme RAM është 2 KB. Frekuenca maksimale është 20 MHz. Arduino Uno R3 lidhet me kompjuter përmes një kabllorja USB, ku vetë Arduino ka një mini

konektor USB. Përmes kësaj kabloje transferohen programet nga kompjuteri në Arduino Uno R3 dhe të njëjtit shkruhen në memorien programore të mikrokontrolluesit ATmega328. Pasi të jetë programuar, Arduino Uno R3 shkëputet nga kompjuteri, kyçet në një pajisje tjetër elektronike dhe përdoret për menaxhimin e procesit.

Arduino Uno R3 përmban tri dioda led sinjalizuese. Led dioda me shenjën ON për furnizimin me energji. Led dioda me shenjën TX dhe RX (Transmit-Receive) janë tregues për komunikimin serik. Këto dy led dioda pulsojnë më intensivisht kur bartet programi në mikrokontrollues. Përveç led diodave të sipërme, ekziston edhe një led diodë tjetër e integruar, e shënuar me shkronjën L, e cila është vendosur në pinin digjital të 13-të dhe përdoret për testimin e funksionalitetit të mikrokompjuterit Arduino Uno përmes ekzekutimit të programit për pulsime (angl. Blink).

Kur është i lidhur me kompjuter Arduino Uno R3, furnizohet me energji përmes portës USB. Në të kundërtën, mund të furnizohet përmes baterisë ose një adapteri të jashtëm, me tension nga 7 deri në 12 V.

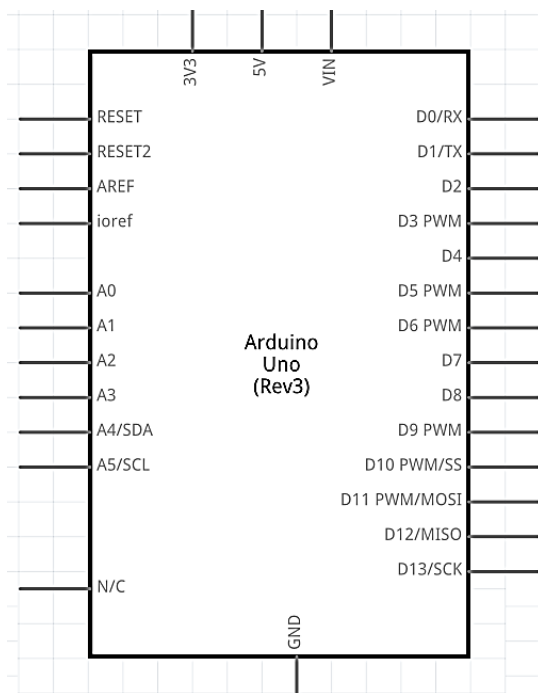


Figura 4.4. Skema funksionale e Arduino Uno R3

Në figurën 4.4 është paraqitur skema funksionale e Arduino Uno R3. Ai posedon me **14 pina digjital**. Pinat digjital mund të jenë si hyrje ose dalje, por jo dykahëshe. Elemente më të thjeshtë për punë me pinat digjital janë tasteri si element hyrës dhe led dioda si element dalje. Sinjalet digjitale mund të kenë vetëm dy vlera: zero logjike ose njësh logjik. Për shembull, kur tasteri do të jetë i shtypur, mikrokontrolluesi merr të dhënë njësh logjik (lidhet në tension prej 5 V), ndërsa kur është i lëshuar, ajo është gjendja e zeros logjike (e lidhur me tokëzimin).

Pinat me numra rendor 3, 5, 6, 9, 10 dhe 11 kanë funksion të dyfishtë. Ata mund të përdoren si pina digjital ose dalje analoge. Daljet analoge janë të shënuara me shenjëm PWM (angl. Pulse Width Modulation) që në përkthim do të thotë modulim në gjerësi të impulsit. Me këtë metodë sinjali analog konvertohet në një seri impulsesh dhe pauzash. Gjatë programimit të Arduino Uno R3 dhe punës me dalje analoge, programuesi cakton (shkruan) vlera të plota në intervalin nga 0 deri në 255. Varësia e gjerësisë së impulseve nga madhësia e vlerës së plotë

është paraqitur në Figurën 4.5. Gjerësia, gjegjësisht kohëzgjatja e impulseve dhe madhësia e tensionit të daljes varen në proporcion të drejtë nga vlera e plotë e shkruar. Për shembull, vlera 65 përfaqëson 25% të vlerës maksimale të lejuar 255, prandaj gjerësia e impulseve do të jetë 25% e periudhës dhe tensioni i daljes do të jetë 25% i vlerës së tij maksimale (5V), që është ekuivalent me 1.25 volt. Modulimi i impulseve në gjerësi përdoret edhe për hyrjet analoge, por me renditje të kundërt, pra në vend të shkrimit bëhet leximi. Tensioni i hyrjes, i cili silltet në brezin nga 0V deri në 5V, konvertohet në seri të impulseve me gjerësi të ndryshueshme, ndërsa gjerësia e impulseve përcakton vlerën e plotë që do të lexohet si madhësi hyrëse. Vlerat e plota hyrëse sillen në brezin nga 0 deri në 1023, gjegjësisht në brez katër herë më të vogël nga ai i daljes. [4]

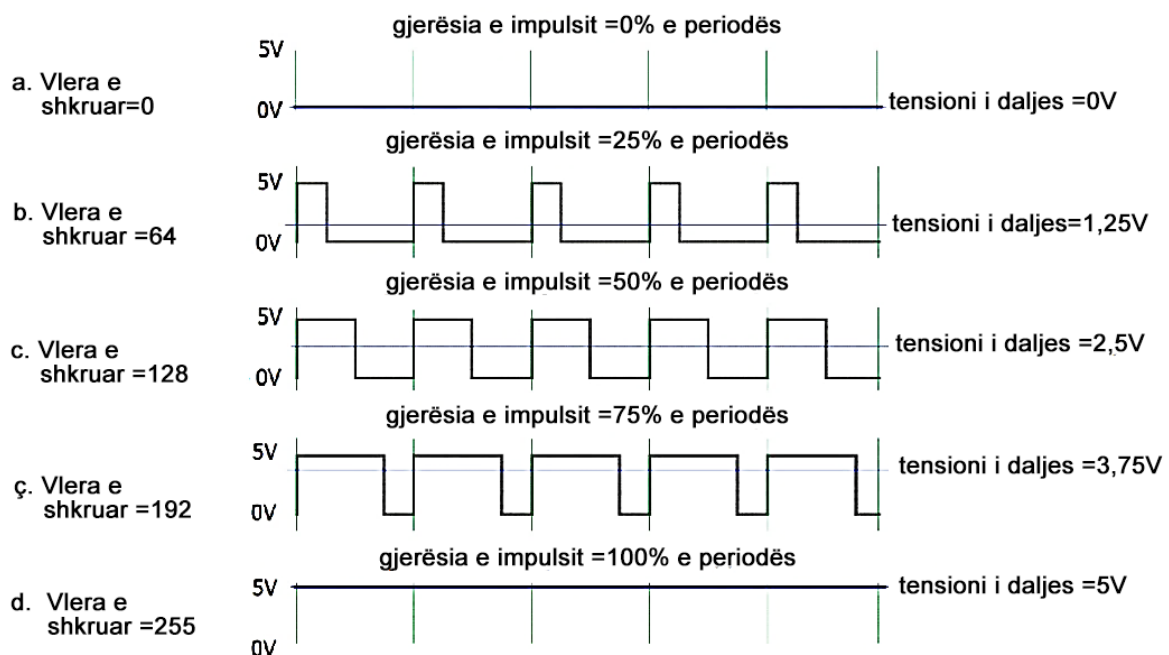


Figura 4.5. Përdorimi i modulimit në gjerësisë së impulseve në pinat dalës analog

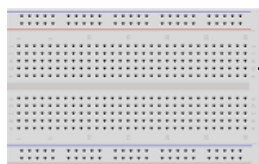
Gjithashtu, pini i parë dhe pini zero i Arduino Uno R3 kanë një funksion të dyfishtë. Përveçse si pina digjital të njëjtit mund të përdoren si pina për komunikim serik, për shembull gjatë lidhjes së dy platformave Arduino ose Arduino me mikrokompjuterin në pllakë Raspberry Pi.

Pini për tokëzim (angl. GND, Ground) dhe pini për tensionin prej 5V përdoren për furnizimin me energji të protoplakës, në të cilën saktë janë të lidhura të gjitha komponentët elektronikë sipas një skeme elektrike të caktuar.

4.3. Elementet periferike dhe komponentët elektronikë për lidhjen me mikrokompjuterin Arduino

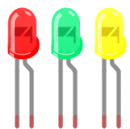
Që Arduino Uno të komunikojë me botën e jashtme dhe të marrë informacione me rrethinën përreth, janë të nevojshme njësi hyrëse: sensorë (piezozensorë, tilt **sensorë**, fotorezistorë, sensorë temperature), taster, potenciometra. Që platforma Arduino të reagojë ndaj një procesi, janë të nevojshme njësi dalëse, të njohur me emrin si aktuatorë. Në këtë kategori përfshihen: ekranet, motorët dhe sirenat.

Që njësitë hyrëse dhe dalëse të lidhen në mënyrë efikase me mikrokompjuterin Arduino Uno R3, janë të nevojshme komponentë elektronikë shtesë si: rezistorë, kondensatorë, dioda drejtuese, optokaplerë, drejtues, etj. Shkurtimisht do ta përshkruajmë funksionin e tyre. [4]



protopllaka

Protopllaka është zgjedhje e shkëlqyer për fillestarët që nuk kanë shumë njohuri paraprake në elektronikë dhe saldimit. Nën pllakën plastike janë të vendosur përçues vertikalë dhe horizontalë. Në vrimat e pllakës plastike vendosen daljet e komponentëve elektrike dhe tela për lidhjen e tyre, lidhje të shkurtëra. Udhëzimet për punë me protopllakë në pjesën praktike të njësisë modulare.



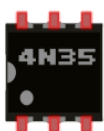
diodat led

Diodat led janë elemente dalëse më të thjeshta. Për të ndezur një diode led, është e nevojshme që kontakti më i gjatë (anoda) të lidhet me potencialin më të lartë në krahasim me kontaktin më të shkurtër (katoda).



Tasterët

Butonat kanë katër kontakte, nga dy në çdo anë të kundërt. **Kontaktet në të njëjtën anë nuk janë të lidhura elektrikisht** dhe ato duhet të vendosen në vrimat e dy përçuesve të ndryshëm të protopllakës, që qarku elektrike të mbyllet kur shtypet tasteri.



Optokapleri

Optokapleri përbëhet nga një diode led dhe një fotodiodë. Nëpër qarkun e fotodiodës kalon rryma vetëm derisa dioda led ndriçon. Dioda led është e lidhur me burimin e furnizimit, ndërsa fotodioda me ngarkesën. Me ndihmën e optokaplerit qarqet elektrike të burimit dhe të ngarkesës janë të izoluar elektrikisht.



Potenciometri



Potenciometri është rezistor i ndryshueshëm dhe zakonisht është i lidhur me njërën nga hyrjet analoge të Arduino Uno R3 për të krijuar një tension të ndryshueshëm.



Fotorezistori



Fotorezistori ka një rezistencë të ndryshueshme, e cila ndryshon në varësi të intensitetit të dritës që bie mbi sipërfaqen e tij.



Motori i rrymës njëkahëshe



Motori i rrymës njëkahëshe e konverton energjinë elektrike në energji mekanike, duke kryer lëvizje rrethore. Nëse ndryshohet drejtimi i rrymës, do të ndryshoj edhe drejtimi i rrotullimit të motorit.



Servo motori



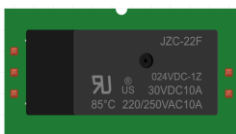
Servo motori nuk rrotullohet për një rreth të plotë, por vetëm deri në 180 shkallë. Në varësi të tensionit të hyrjes, ai do të zhvendoset për një kënd të caktuar, deri në një pozicion të caktuar, ndërsa do të mbetet në atë pozicion derisa tensioni i hyrjes të ndryshojë. Këndi i rrotullimit varet nga gjerësia e impulseve pas modulimit në gjerësi të impulseve.



Motori me hap



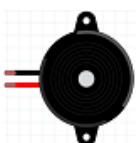
Motori me hap bëjnë lëvizje diskrete, gjegjësisht me çdo impuls hyrës motori zhvendoset për një kënd të caktuar. Kjo zhvendosje quhet hap. Sa hapa janë të nevojshëm për të bërë një rreth të plotë varet nga konstruksioni i motorit (numri i poleve).



Reletët



Reletët mundësojnë kyçjen dhe shkyçjen e konsumatorëve me fuqi të lartë përmes përdorimit të sinjaleve me tension të ulët ose me rrymë të ulët. Releja mund të ketë kontakte normale të mbyllura dhe kontakte normale të hapura. Në rastin e kontakteve normale të hapura, qarku i konsumatorit është i ndërprerë edhe kur qarku i kontrollit është i ndërprerë. Në rastin e kontakteve normale të mbyllura, konsumatori është i lidhur me burimin e energjisë kur qarku i kontrollit është i ndërprerë.



Piezo komponenta



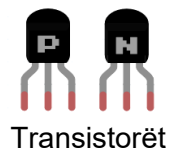
Piezo komponenta mund të përdoret si sensor për vibracione ose si gjenerator i toneve me frekuenca të ndryshme.



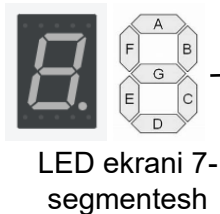
Rezistorët përdoren për rregullimin e intensitetit të rrymës së punës që të mos vij deri te dëmtimi i elementeve.



Tilt senzori është sensor për pozicionin. Në brendi të tij ekziston një top metalik i cili zhvendoset dhe në këtë mënyrë hap ose mbyll kontaktin, gjegjësisht i ndërpret ose i lidh kontaktet e sensorit. Në cilën pozitë, horizontale ose vertikale, do të mbyllet kontakti varet nga lloji i sensorit.



Transistorët përdoren si përforcues të tensionit ose të rrymës dhe si ndërprerës. Transistori 2N2222 është i përshtatshëm për rryma deri në 500 mA, ndërsa transistori TIP120 për rryma deri në 5 A.



Ekrani 7-segmentesh përdoret për paraqitjen vizuale të shifrave decimale. Shtatë pina përdoren për të eksitim të segmenteve të shënuar me shkronja latine nga a deri në f, një pin është për pikën dhe dy të tjerë janë për tokëzimin. Ekrani 7-segmentesh mund të jetë me anodë të përbashkët ose katodë të përbashkët dhe kjo është e rëndësishme për eksitim të pinave.



Ura-H (angl. H-Bridge) është qark elektronik i cili mundëson ndryshimin e drejtimit të rrymës që kalon përmes një motori për rrymë njëkahëshe. Ndryshimi i drejtimit të rrymës shkakton ndryshimin e drejtimit të rrotullimit të motorit. Realizohet me elemente ndërprerës ose në formën e një qarku të integruar siç është L239D.



Nëse tensioni i kondensatorit është më i ulët se tensioni i burimit, atëherë kondensatori mbushet, ndërsa në të kundërtën ai zbrazet. Zakonisht kondensatorët lidhen paralelisht me sensor ose motor, me qëllim që të parandalohet ndryshi i shpejtë i tensionit, shpesh i shkaktuar nga komponentët tjerë në qark.

Para se të fillojmë me lidhjen e komponentëve në protoplakë është e nevojshme të kryhet analiza e karakteristikave të tyre dhe zgjedhja e të njëjtave. Çdo prodhues i komponentëve elektronike përgatit dokumentacion tekniko-teknologjik i cili përmban informacion për performancat, mënyrën e përdorimit dhe karakteristikat, siç është vlera e tensionit minimal që është i nevojshëm për funksionimin e komponentit dhe tensioni maksimal i lejuar. Disa komponentë siç janë rezistori dhe kondensatori i zakonshëm janë komponentë pa polarizim dhe

tek ata nuk është e rëndësishme se cili konektor do të jetë në potencial elektrik më të lartë, e cili në më ulët.

Me Arduino Uno R3 mund të lidhni komponentë me tension të punës deri në 5V dhe rrymën maksimale të lejuar prej 40mA. Këto vlera nuk janë të njëjta për të gjitha platformat Arduino. Për shembull, Arduino Uno WiFi Rev2 punon me një rrymë maksimale të lejuar prej 20mA, ndërsa Arduino Zero ka një rrymë maksimale prej 7mA.

Në figurën 4.6 dhe figurën 4.7 janë paraqitur skema funksional dhe e montimit për lidhjen e motorit me rrymë njëkahëshe me mikrokontrolluesin Arduino Uno R3. Në skemën funksionale simbolet paraqesin komponentët elektronikë, ndërsa linjat mënyrën e lidhjes së tyre. Nëse qarku elektrik është i thjeshtë, me numër të vogël të komponentëve, ndoshta skema e montimit është më e lehtë për t'u kuptuar. Por, në rastin e një numri të madh të komponentëve dhe telave për lidhjen e tyre, atëherë është e domosdoshme skema funksionale sepse ajo na ofron ma tepër qartësi [5].

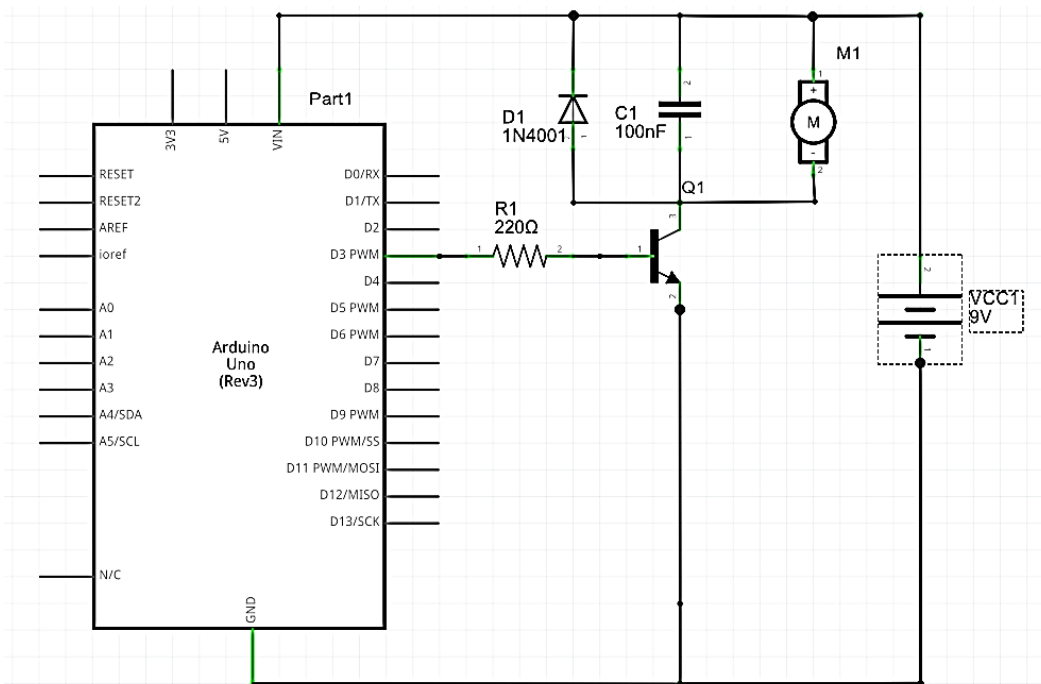


Figura 4.6. Skema funksionale për lidhjen e motorit të rrymës njëkahëshe me Arduino Uno R3

Në skemën e paraqitur në figurën 4.6, transistori ka funksionin e një ndërprerësi. Baza e transistorit është lidhur me pinin e tretë të Arduino Uno R3, i cili përdoret si dalje analoge me impulse të moduluara në gjerësi dhe nga kohëzgjatja e impulseve varet madhësia e tensionit për furnizimin e motorin, gjegjësisht intensiteti i rrymës që kalon përmes tij. Në dokumentacionin tekniko-teknologjik të transistorit duhet të kontrollojmë nëse tensioni maksimal i lejuar kolektor-emiter është më i madh se tensioni i furnizimit të motorit elektrik dhe intensiteti i rrymës së kolektorit duhet të jetë 25% më e madhe se rryma që kalon

nëpër motor. Rryma e bazës varet nga rryma e kolektorit dhe koeficienti i përforcimit të rrymës. Për shembull, nëse koeficienti i përforcimit të rrymës është 100, ndërsa rryma e dëshiruar e kolektorit është 1A, atëherë rryma e bazës do të jetë $1A / 100 = 0,01A = 10mA$. Me ndihmën e ligjit të Omit mund të llogarisim vlerën e rezistencës së lidhur me pinin e tretë digjital. Për shembull: $5V / 0,01A = 500\Omega$ dhe pasi që kjo nuk është vlerë standarde, zgjedhim 470Ω .

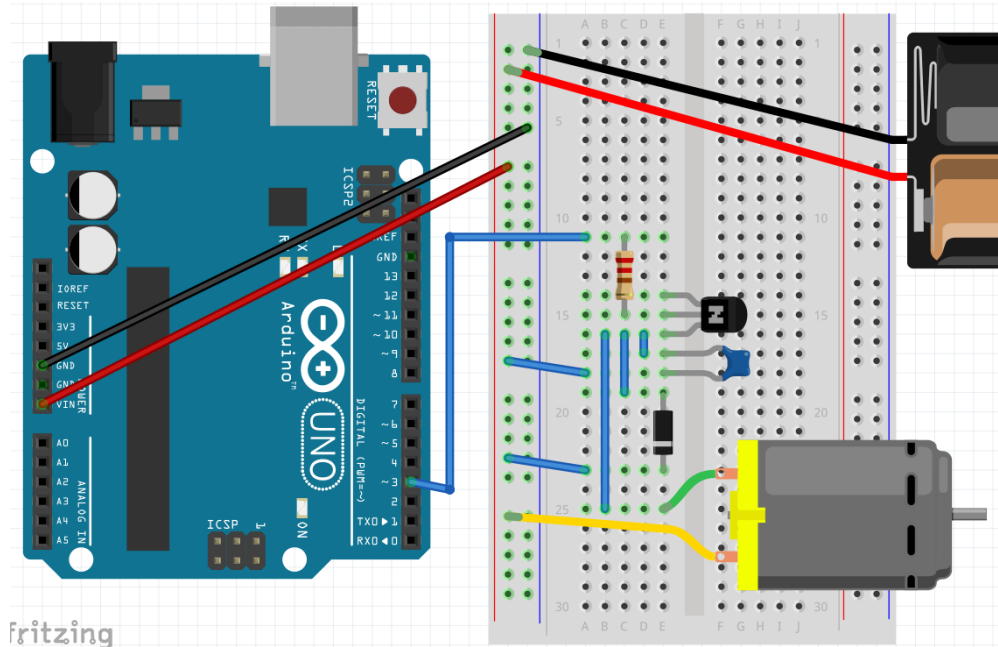


Figura 4.7. Skema e montimit për lidhjen e motorit të rrymës njëkahëshe me Arduino Uno R3

Dioda është e lidhur paralelisht me motorin dhe i mbron pinat e Arduino Uno R3 nga paraqitja e rrymave induktive të larta, të cilat paraqiten gjatë kyçjes dhe shkyçjes së motorit. Ndryshimi i gjendjes së pinave digjital mund të jetë shumë i shpejtë dhe në qark paraqiten oscilacione në tensionin e furnizimit dhe të njëjtit filtrohen me përdorimin e kondensatorit.

Furnizimi me energji i Arduino Uno është karakteristikë tjetër që duhet marrë parasysh. E dimë tashmë se Arduino Uno R3 mund të furnizohet në dy mënyra: përmes USB-së ose me furnizim të jashtëm. Pas regjistrimit të programit, nuk është e nevojshme lidhja e Arduino Uno R3 me kompjuterin dhe atëherë mund të përdoret furnizim i jashtëm, një bateri ose adapter i cili tensionin alternative e konverton në tension njëkahësh me vlerë nga 7V deri në 12V. Tensionet më të larta se 12V mund të shkaktojnë dëmtime të pllakës elektronike. Tensioni i baterisë gjithashtu duhet të jetë në brezin prej 7V deri në 12V. Kapaciteti i baterisë matet në mAh. Sa do të zgjat funksionimi i baterisë varet nga konsumimi i energjisë elektrike së komponentëve që janë të integruar në pajisjen elektronike.

Analiza e skemës së lidhjes së Arduino Uno R3 me motor të rrymës njëkahëshe është vetëm një shembull i asaj se si duhet të studiohen funksioni dhe karakteristikat e komponentëve elektronike para se ato të lidhen me mikrokontrolluesin Arduino Uno R3.

4.4. Shtesa për mikrokontrolluesin Arduino në pllakë

Mikrokontrolluesi Arduino është bërë shumë i popullarizuar mes përdoruesve të tij, për çka një kontribut shumë të madh ka dhënë koncepti i burimit të hapur, si për harduerin ashtu edhe për softuerin. Shumë përdorues që eksperimentojnë me dizajnimin e pajisjeve të tyre ju dhanë ide prodhuesve të komponentëve elektronike për të krijuar dhe lëshuar në shitje shtesa - aksesore të gatshëm me të cilët shumë u lehtësua puna me Arduino dhe dukshëm është rritur funksionaliteti i tyre. Aksesoret janë në fakt pllaka elektronike të gatshme me një qëllim të caktuar. Lidhja është e thjeshtë dhe zakonisht aksesoret përmbajnë pina mashkullore të gjatë që vendosen në pinat femërorë të Arduino. Mburojat (angl. Shields) janë aksesore që vendosen mbi pllakën bazë të Arduino, me ç'rast të dy pllakat janë të vendosura një mbi tjetrën, por nuk preken falë pinave mashkullore të gjatë. Emri mburojë ka një kuptim simbolik. Pasi që softueri i Arduino është i llojit të hapur, shfrytëzuesit kanë qasje në një numër të madh të kodeve programeve të gatshme për aksesoret të cilat mund t'i modifikojnë sipas nevojave të tyre. Po ashtu, shumica e këtyre aksesoreve kanë mbështetje softuerike në vetë mjedisin zhvillimor të integruar të Arduinos. Gjatë shfrytëzimit të një aksesori, në kodin programor duhet të thirret e ashtuquajtura bibliotekë pa të cilën aksesori nuk mund të komunikojnë me Arduino-n. Bibliotekat përmbajnë instruksione shtesë me të cilat lehtësohet manipulimi me harduerin. Siç do ta shohim më vonë, madhësia e aksesoreve është e shumëllojshme, me dimensione prej 1-2 cm dhe 4 pina, e deri në aksesori që kanë të njëjtën madhësi dhe numër pinash si edhe pllaka zhvillimore e Arduino-s, siç janë mburojat. Zakonisht aksesoret për Arduino nuk janë për fillestarët dhe nuk do t'i përdorim në pjesën praktike, por është mirë të njohim mundësitë që ato ofrojnë. Ne do të njihemi me një aksesori dhe pesë mburoja.

Pa internet ose lidhje Wi-Fi, Arduino nuk mund të komunikojë me botën e jashtme, të dërgojë të dhënat e marra nga sensorët ose të komandohet nga distanca. Për të lidhur Arduino-n në rrjet nevojitet një aksesori i veçantë si aksesori Wi-Fi ose mburojë Ethernet. Aksesori ESP8266-01 është modul Wi-Fi me çmim shumë të ulët dhe dimensione 25mmx15mm, që e shumë të përshtatshëm për lidhjen në rrjet të Arduino Uno R3 në aplikacione të platformës Internet of Things.

Pamja e jashtme dhe skema e montimit për lidhjen e aksesorit ESP8266-01 me Arduino Uno R3 janë paraqitur në figurën 4.8. Ai përmban tetë pina dhe ato nuk mund të vendosen drejtpërdrejt në pinat e Arduino Uno R3 prandaj përdoren tela për lidhje. Aksesori ESP8266-01 nuk ka të integruar portë USB por përdoret komunikimin serik asinkron përmes dy pinave TX dhe RX. Dy pinat hyrës-dalës (angl. IO - Input Output) mund të përdoren për lidhjen për shembull të një sensori [5].

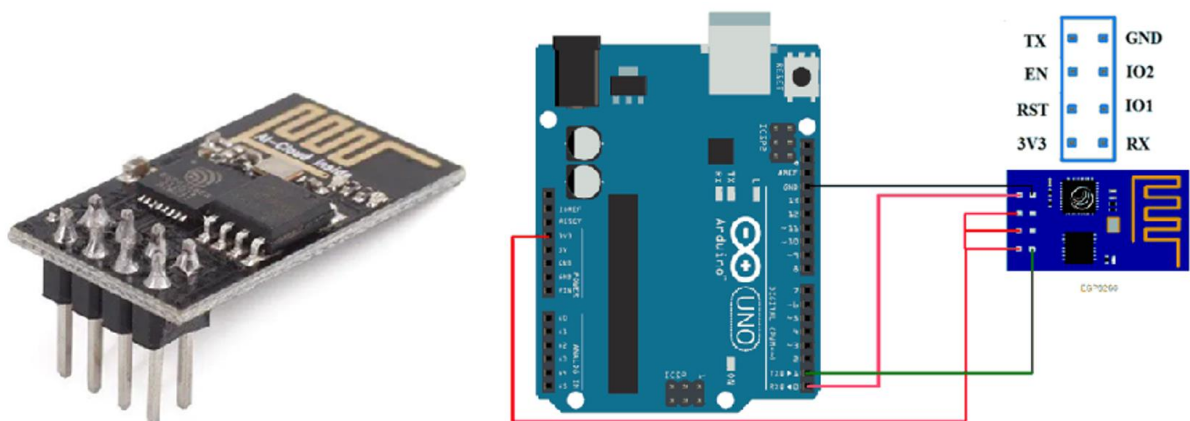


Figura 4.8. Pamja e jashtme dhe skema e montimit për lidhjen e aksesorit ESP8266-01

Distanca maksimale për lidhje Wi-Fi është një nga 100 deri në 250 metra. Biblioteka për këtë modul nuk është pjesë e mjedisit zhvillimor të integruar të Arduino por është i nevojshëm instalim shtesë.

Në figurën 4.9 është paraqitur mburoja Arduino Ethernet 2 i cili mbështet tetë lidhje të njëkohshme. Lidhet me Arduino Uno R3 përmes pinave mashkullorë të gjatë dhe



Figura 4.9. Shield-i Arduino Ethernet 2

pasi që diagrami i pinave është i njëjtë, është e mundur që mbi të të vendoset edhe një mburojë tjetër. Për lidhjen me rrjetin internet përdoret konektori RJ-45 dhe ka një slot të integruar për kartelë memoruese për ruajtjen e dokumenteve që duhet të shërbehen në rrjet.

Mburoja GSM/GPRS Arduino është aksesor që mundëson lidhjen e Arduino në rrjetin celular.



Figura 4.10. Mburoja GSM/GPRS Arduino

Përdoruesi mund të dërgojë dhe marrë mesazhe SMS, të bëjë dhe të pranojë thirrje si dhe të ketë qasje në internet përmes rrjetit global mobil dhe paketave të serviseve të orientuara për transmetimin e të dhënave (angl. Global System Mobile/ General Packet Radio Service).

Mburoja posedon slot të integruar për kartelën SIM, konektorë për kufje, antenë, Bluetooth 3.0 me antenë, ndërsa gjithçka që është e nevojshme për të mundësuar që Arduino Uno R3 të funksionojë si një telefon celular.

Në figurën 4.11 është paraqitur mburoja Arduino për motorë, e cila mund të kontrollojë katër motorë të rrymës njëkahëshe ose dy servo motorë dhe dy motorë me hap.

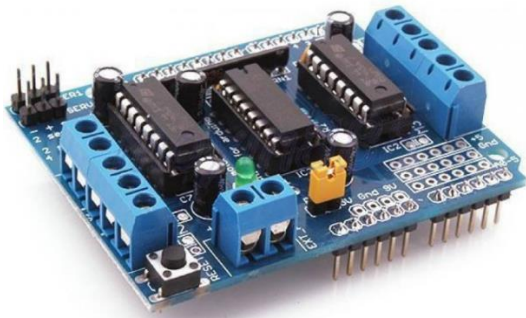


Figura 4.11. Mburopa Arduino për motor

Këta motorë nuk mundet të jenë direkt të kontrolluar vetëm me Arduino për shkak të rrymave të forta dhe fuqisë së madhe. Mburopa për motorë Arduino kryen ndarjen e rrymës për furnizimin me energji të motorit dhe përdor logjikën e platformës Arduino. Përmban dy qarqe të integruara L293D dhe paraqet një H-bridge me dy kanale për kontrollin e rrotullimeve të motorit.

Arduino nuk është i dizajnuar për kontrollin e konsumatorëve me fuqi të madhe, tensioni maksimal është 5V, me rrymë të punës deri në 40mA.



Figura 4.12. Mburopa rele Arduino

Reletë ofrojnë ndarje galvanike të pjesës së tensionit të ulët nga pjesa e tensionit të lartë të sistemit për komandim me konsumatorë të fuqishëm. Në figurën 4.12 është treguar një mburojë rele Arduino me dy kanale me tension dalës alternativ prej 250V për qarkun e konsumatorit ose tension njëkahësh 30V dhe rrymë maksimale prej 10A.

Nuk është e mundur të numërohen të gjithë aksesorët të cilët përdoren për zgjerimin e funksionalitetin e mikrokompjuterëve në pllakë Arduino. Pa marrë parasysh zgjedhjen, gjithmonë së pari duhet të lexohet dokumentacioni tekniko-teknologjik i cili i përmban të gjitha informacionet për mënyrën e lidhjes dhe mbrojtjen e pajisjeve nga dëmtimet e mundshme.

4.5. Programimi i mikrokompjuterit në pllakë Arduino në gjuhën programuese C/C++

Sot, programimi është sfidë për shumë të rinj, pikërisht për shkak të paraqitjes së shumë veglave softuerike, mjediseve zhvillimore të cilat janë të lehta për t'u përdorur dhe që mundësojnë krijimin e aplikacioneve pa përdorimin e modeleve të ndërlikuara matematikore. Për programim të suksesshëm të mikrokompjuterit në pllakë Arduino, është e nevojshme njohja e dy veçorive softuerike: mjedisi i tij zhvillimor dhe grupi i instruksioneve të tij.

Platforma Arduino nuk posedon me sistem operativ, prandaj atë nuk mundet që ta programojmë pa e **lidhur me një kompjuter personal**. Për shkruarjen dhe futjen e programet në mikrokontrolluesin Arduino, është i nevojshëm programi zhvillimor, i njohur edhe si **mjedisi zhvillimor** (angl. IDE – Integrated Development Environment).

Me mënyrën e instalimit të mjedisit zhvillimor të integruar për mikrokompjuterin Arduino dhe programet sistimore të tij do të njihemi pjesën praktike të kësaj njësie modulare.

Arduino mund të ekzekutojë vetëm një program dhe ekzekutimi i tij fillon menjëherë pas kyçjes së furnizimit. Për ta mësuar më lehtë programimin e Arduino rekomandohet përdorimi i simulatorit Tinkercad, i cili gjithashtu do të shpjegohet në pjesën praktike të kësaj njësie modulare.

Për programet e shkruara për Arduino përdoret emërtimi skica (angl. sketch). Në këtë mënyrë theksohet struktura e thjeshtë e programeve, për realizimin e shpejtë dhe të lehtë të ideve. Skica është e përbërë prej instruksioneve. Instruksionet janë shprehje që nxisin një veprim të caktuar për t'u fituar rezultati i dëshiruar. Nëpërmjet instruksioneve, ne e kontrollojmë mikrokompjuterin Arduino dhe kryejmë llogaritje.

Skica përmban dy struktura të detyrueshme: setup dhe loop. Shembulli 4.1 është përshkrim i përgjithshëm i këtyre dy strukturave të detyrueshme në skicë:

setup() → Setupi është funksion pa argumente hyrëse. Me këtë **jepen vlerat fillestare të variablave, konfigurohen pinat** si hyrës ose dalës, **zgjidhet shpejtësia për transmetim për monitorin serik, specifikohen bibliotekat e përfshira**. Ky funksion ekzekutohet vetëm një herë, kur kyçet furnizimi ose pas ristartimit të mikrokompjuterit Arduino.

loop() → Me këtë funksion, në fakt, kryejmë kontrollin e mikrokompjuterit Arduino. Fjala në anglisht loop në përkthim do të thotë lak dhe siç tregon edhe vetë emri, ky funksion vazhdimisht ekzekutohet,

rrotullohet në rreth, **vazhdimisht i ndjek ndryshimet e gjendjes së pinave të hyrjes dhe reagon në mënyrë përkatëse**. Në fakt funksioni loop paraqet një cikël pa limit i cili përsëritet deri sa Arduino ka furnizim.

Shembulli 4.1:

```

1 int buttonPin = 3;           // Fillimisht deklarohen variablat.
2 void setup() {              // setup është gjithmonë funksioni fillestar do të
                               // thotë se funksioni nuk do të jap vlerë
                               // kthyesë.
3   Serial.begin(9600);       // Definojmë shpejtësinë e transmetimit për monitorin
                               // serik.
4   pinMode(buttonPin, INPUT); // Konfigurimi i pinit hyrës.
5 }
6 void loop() {               // Fillimi i funksionit loop.
  .....
}                               // Fundi i funksionit loop.
```

Në fillim të programimit të Arduino, mund të përdoren kode programore të gatshme, por për një punë të mëtutjeshme më të suksesshme është e domosdoshme analiza e tyre, me qëllim të ndryshimit të tyre dhe përshtatjes. Sfidë është që nxënësit të mësojnë që vetë të krijojnë programet e sipas karakteristikave të komponentëve elektronike që do të përdorin. Por, para se të fillojmë krijimin dhe analizën e programeve për mikrokompjuterin Arduino, do të njihemi me elementet përbërëse bazë të gjuhës programuese C/C++: variablat, instruksionet dhe strukturat. Është e rëndësishme të kuptohet kuptimi i tyre për kompjuterin dhe të mbahet mend sintaksa e tyre. Sintaksa në programim përbëhet nga një grup i rregullave për renditjen e shenjave, emrave simbolikë, operatorëve, shenjave të pikësimit, komenteve, me qëllim që të fitohet shprehje e kuptueshme për kompjuterin.

4.6. Variablat dhe operatorët në gjuhën programuese C/C++ për mikrokompjuterin Arduino

Variablat i ruajnë të dhënat që përpunohen në kompjuter. Për çdo variabël parashihet dhe rezervohet vend në memorie. Çdo vend i rezervuar ka adresën e tij. Pasi adresat janë të vështira për t'u mbajtur mend, variablat u japin emra simbolikë. Përveç emrit simbolik, secilës variabël duhet t'i caktohet edhe një lloj shenje për llojin e të dhënës. Madhësia e hapësirës së ruajtur në memorie varet nga tipi i të dhënave. Gjithashtu, kompjuteri në mënyrë të ndryshme sillet ndaj llojeve të ndryshme të të dhënave. Për shembull, me

instruksione aritmetike mund të përpunohen numra, por jo edhe të dhëna tekstuale. Më poshtë janë dhënë shenjat e **llojeve themelore të të dhënave**.

- int → Numrat e plotë janë të dhëna më shpesh të përdorura. Platforma Arduino rezervon hapësirë memoruese prej dy bajtave për çdo numër të plotë. Brezi i numrave të plotë sillet prej -32768 deri në +32,767.

- float → Numrat decimal përdoren për paraqitjen e vlerave analoge. Numrat decimal me 7 vende decimale zënë hapësirë prej 4 bajtave në memoriën e Arduinos dhe brezi i numrave sillet nga 3,4028235E+38 deri në -3.4028235E+38. Numri pas shkronjës E është eksponent i shkallës me bazë 10 (për shembull, 35E3=35·10³=35·1000=35000).

- double → Numrat decimal me 15 vende decimale janë më të saktë se sa numrat me 7 vende decimale dhe ato zënë hapësirë memoruese prej 8 bajta.

- char → Një shenjë (angl. character) paraqet një shkronjë, shifër ose shenjë alfanumerike. Një karakter zë një bajt dhe ka një gamë prej 256 karakteresh. Një shenjë zen një memorie prej një bajti dhe brezi i vlerave është 256 shenja. Shenjat shkruhen mes apostrofave teke (përr shembull, 'A').

- string → Teksti parawet varg prej simboleve dhe ai shkruhet mes thonjzave (për shembull "mirdita").

- bool → Të dhënat logjike kanë vetëm dy vlera, zero dhe njësh, e pasaktë (angl. flase) ose e saktë (angl. true). Një variabël logjike zen hapësirë prej një bajti.

Çdo variabël që e përdorim në program duhet patjetër ta deklarojmë (lajmërojmë). Së pari theksohet lloji i të dhënës dhe më pas emrin e variablës. Nëse gjatë deklarimit të variablës i caktojmë edhe ndonjë vlerë, themi se bërë inicializimin. Më vonë vlera e variablës së inicializuar mund të ndryshohet kudo qoftë në program. Variablave u ndahet vlerë me shenjën e barazisë (=) dhe kjo shenjë është e njohur edhe me emrin operator për caktim. Konstantat janë të dhëna vlera e të cilave nuk ndryshon gjatë rrjedhës së programit. Nëse bëhet gjalë për konstantë, atëherë para llojit të të dhënës duhet të qëndroj fjala const.

Shembulli 4.2:

```

1 int countUp = 0;           // Variabël e numrit të plotë me emrin simbolik
                             // 'countUp'.
2 const float pi = 3.14;    // Numër decimal me vlerë konstante.
3 bool isCodingFun = true;  // Variabël logjike me emrin simbolik
                             // isCodingFun dhe vlerë të saktë
    
```

```
4 string stringOne = "Hello String"; // stringOne është emri i variablës tekstuale,
// kurse Hello String është vlera e tij.
//
```

Vargjeta janë grupe të variablave të llojit të njëjtë. Deklarimi i vargut përmban: llojin e të dhënave, emrin e vargut dhe numrin e elementeve. Kllapa e mesem përmban numrin e elementeve, ndërsa në kllapa të mëdha numërohen elementet. Për qasje deri te elementet përsëri përdoret kllapa e mesme gjegjësisht shkruhet emri i vargut dhe numri rendor i elementit në kllapa të mesme.

Shembulli 4.3:

```
1 string avto[3]={"Volvo", "BMW", "Ford"}; // Vargu grup i variablave tekstuale, auto
// është emri i vargut, numri në kllapa të
// mesme do të thotë se vargu përmban 3
// elemente, ndërsa elementet janë në
// kllapa të mëdha.
2 avto[0] = "Opel"; // thirim elementin me numër rendor zero
// dhe e ndryshojmë vlerën e tij.
```

Vargjet mund të jenë edhe dydimensional dhe atëherë elementet janë të shpërndarë në rreshta dhe kolona, ndërsa numrat në dy kllapa të mesme shënojnë numrin e kolonave dhe numrin e rreshtave.

Shembulli 4.4:

```
1 int Table [2][3] = {{3, 42,1}, {7, 3, 12}}; // Vargu dydimensional ka 2 rreshta dhe
// 3 kolona
```

Operatorët janë shenja me të cilat programuesit, sipas rregullave të përcaktuara saktë, formon shprehje, instruksione. Ekzistojnë më tepër lloje të operatorëve: operatorë matematikorë, logjikë, krahasues dhe operatorë për caktimin e vlerave. Shkurtimisht do të njihemi me secilin prej tyre.

Operatorët matematikorë janë të njohur edhe me emrin operatorë aritmetikë [6]

+	→	Operator për mbledhje
-	→	Operator për zbritje
*	→	Operator për shumëzim
/	→	Operator për pjesëtim
%	→	Operator për llogaritjen e mbetjes gjatë pjesëtim

Shembulli 4.5:

```
1 float a = 5.5;
2 float b = 6.6;
3 int c = 0;
4 c = a - b; // Pasi që 'c' është variabël numër i plotë, vlera e tij do të jetë -1
// edhe pse rezultati i pjesëtim është -1,1.
```

Shembulli 4.6:

```
1 int a = 5;
```

```

2 int b = 10;
3 int c = 0;
4 c = a * b;          // Variabla 'c' fiton vlerën 50

```

Shembulli 4.7:

```

1 float a = 55.5;
2 float b = 6.6;
3 int c = 0;          // Pasi që variabla 'c' është variabël numër i plotë, ajo fiton
4 c = a / b;          // vlerën 8 edhe pse rezultati i pjestimit është 8.409
                       // [6]

```

Shembulli 4.8:

```

1 int x = 0;
2 x = 7 % 5;          // x = 2
3 x = 9 % 5;          // x = 4

```

Shprehjet që përmbajnë **instruksione krahasuese** japin rezultat true (e saktë) ose false (e pasaktë). Zakonisht përdoren në funksionet për kusht. Shpesh herë, me qëllim që të zvogëlohet gjatësia e shprehjeve, përdoren operatorë të veçantë.

- != → Ky është operator për "nuk është i barabartë". Krahasohen dy variabla (x dhe y) ose variabël dhe konstantë dhe nëse janë të ndryshme, shprehja është e vërtetë. Variablat mund të jenë numra të plotë, numra decimal ose binar. Përdorimi i këtij operatori është treguar në shembullin 4.10.
- < → Operatori për më i vogël. Variabla në anën e majtë të operatorit duhet të ketë vlerë më të vogël nga variabla e anës së djathtë.
- > → Operatori për më i madh
- <= → Operatori për më i vogël ose i barabartë
- >= → Operatori për më i madh ose i barabartë
- == → Operator për i barabartë

Shembulli 4.10:

```

1 if (x != y) {          // A është x i ndryshëm nga y?
2 .....                // Kodi programor mes kllapave të mëdha ekzekutohet
: }                    // vetëm nëse shprehja në kllapat e mesme është e vërtetë.

```

Shembulli 4.11:

```

1 if (x < y) {          // A është x më i vogël se y?
2 .....                // Ekzekutimi i kodit programor mes kllapave të mëdha varet
: }                    // nga plotësimi i kushtit të mësipërm.

```

Instruksionet logjike quhen edhe instruksione boolean sepse ato mund të zbatohen në variabla binare (angl. bool). Rezultati gjithashtu është vlerë binare, zero ose njësh, e pasaktë (angl. false) ose e saktë (angl. true).

- &&** → Ky është operatori për operacioni logjik DHE (angl. and). Që të fitohet rezultat logjik njësh të dy variablat duhet të jenë njësha, gjegjësisht të sakta (të vërteta). Nëse njëra nga variablat është zero, atëherë edhe rezultati është zero.
- ||** → Operatori për operacionin logjik OSE (angl. or). Që të fitohet njëshi logjik si rezultat, të paktën njëra nga dy variablat duhet të jetë njësh, gjegjësisht e saktë (e vërtetë).
- !** → Operatori për negacion. Ky operator ka një ndryshore. Me atë ndryshohet gjendja. Nëse variabla ka qenë zero, rezultati do të jetë njësh dhe anasjelltas.

Shembulli 4.12:

```

1  if (!x) {           // Kodi programor në kllapa të mëdha ekzekutohet nëse x është
                        // zero logjike, sepse si negacion do të fitojmë rezultat të
                        // barabartë me njësh.
2  .....
:  }
```

Më poshtë janë dhënë **shprehje të shkurtëra** dhe instruksionet e tyre ekuivalente aritmetike dhe logjike.

x+=y	→	x=x+y
x-=y	→	x=x-y
x*=y	→	x=x*y
x/=y	→	x=x/y
x++	→	x=x+1
x--	→	x=x-1
x&=y	→	x=x&&y
x =y	→	x=x y

4.7. Instruksionet në gjuhën programuese C/C++ për mikrokompjuterin Arduino Uno

Sipas funksionit, instruksionet do t'i ndajmë në disa grupe:

- Instruksione për punën me pinat hyrje-dalje
- Instruksione për kontrollin e kohës
- Instruksione matematikor
- Instruksione për bitë dhe bajta
- Instruksione për komunikim serik
- Instruksione për punën me bibliotekat

4.7.1. Instruksionet për punën me pinat hyrje-dalje

Instruksionet për punën me pinat hyrje-dalje shërbejnë për konfigurimin e pinave (hyrje-dalje) dhe për regjistrimin ose leximin e vlerave të tyre.

pinMode(pin,mode) → Me regjim të pinit digjital nënkuptojmë regjim hyrje ose dalje të punës. Nëse pini është hyrje atëherë Arduino pranon të dhëna nga komponenta elektronike, por nëse është dalje atëherë Arduino dërgon të dhëna. Për daljen shënimi është OUTPUT, ndërsa për hyrjen INPUT. Të theksojmë se ekziston edhe regjim i tretë i punës me shenjën INPUT_PULLUP dhe të njëjtin do ta sqarojmë pak më vonë.

digitalRead(pin) → Me këtë instruksion **lexohet vlera** e pinit hyrës. Vlera mund të jetë HIGH (niveli i lartë) ose LOW (niveli i ulët). Në kllapën e mesme shkruhet numri i pinit ose emri i tij simbolik, nëse paraprakisht e kemi deklaruar pinin si variabël me vlerë të plotë. Nëse pini është dalje, atëherë instruksioni lexim është i pamundur.

Shembulli 4.13:

```

1 int taster=2; // taster është emri simbolik i pinit të dytë.
2 int sostojba=0; // Deklarojmë variabël me emër simbolik
/* ... */ // gjendja dhe vlerë fillestare zero.
3 sostojba=digitalRead(taster); // Leximi i vlerës së pinit të dytë.
```

Skema funksionale e paraqitur në figurën 4.13. ka të bëjë me kodin programor të shembullit 4.13. Si komponentë hyrëse përdoret tasteri i cili është i kyçur në pinin e dytë të Arduino Uno R3. Kur tasteri shtypet, pini i dytë lidhet me tokëzimin dhe vlera e lexuar është LOW. Nëse tasteri nuk është i shtypur, pini i dytë është i lidhur me furnizimin dhe vlera e lexuar është HIGH. Nëse rezistenca dhe tasteri në skemën e figures 4.13 i ndrojnë vendet atëherë kur tasteri do të shtypet, pini i dytë do të lidhet me furnizimin dhe vlera e lexuar do të jetë HIGH, gjegjësisht njësh logjik.

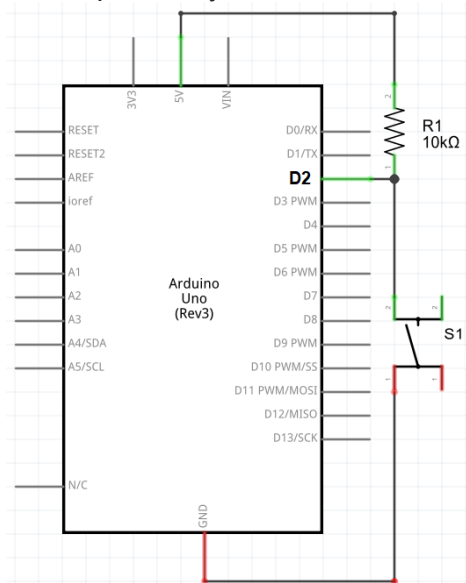


Figura 4.13. Lidhja e tasterit me Arduino Uno R3

Nëse ndërmjet furnizimit prej 5V dhe pinit D2 nuk ka të lidhur një rezistor, atëherë kur butoni nuk shtypet, vlera e lexuar do të jetë vlerë e rastit, gjegjësisht do të ndryshojë vazhdimisht ndërmjet zeros logjike dhe njëshit logjik. Në këtë rast, për të arritur funksionim të qëndrueshëm, përdoret regjimi INPUT_PULLUP me ç'rast aktivizohet rezistori i brendshëm për tërheqje (angl. pull-up), i cili është i integruar në vetë platformën Arduino dhe ndodhet ndërmjet pinit të të dhënave dhe pinit të furnizimit prej 5V (figura 4.14). Në këtë mënyrë, kur tasteri nuk shtypet, vlera e lexuar do të jetë njësh logjik.

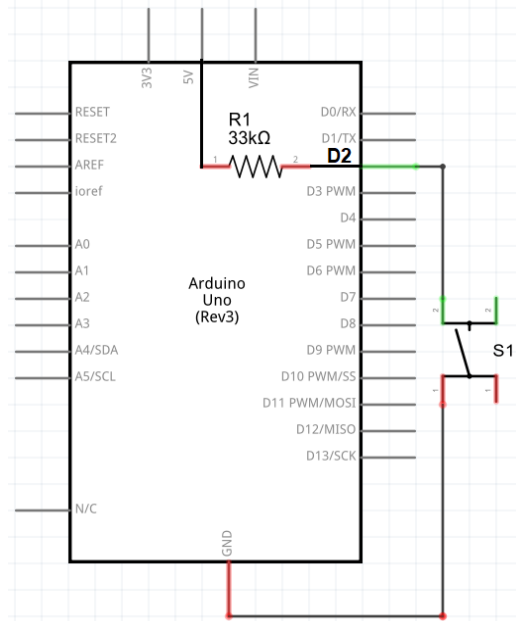


Figura 4.14. Aktivizimi i rezistorit të brendshëm pull-up në Arduino

`digitalWrite(pin,vlerë)` →

Me këtë instruksion, pini dalës digjital vendoset në nivelin e ulët ose të lartë. Për dallim nga `digitalRead`, instruksioni `digitalWrite` përmban dy parametra: emrin simbolik të pinit (ose numrin e tij) dhe vlerën HIGH ose LOW.

Skema funksionale e treguar në fig. 4.15 ka të bëjë me kodin e programit nga shembulli 4.14. Si komponentë dalëse është përdorur dioda led e cila është e lidhur në pinin e katërt dixhital të Arduino Uno R3. Kur pini i katërt do të vendoset në nivel të lartë (HIGH), atëherë dioda led do të ndriçoj. Nëse diodën led e vendosim në mënyrë të kundërt, me katodën e kthyer drejt pinit të katërt, atëherë ajo do të jetë aktive në nivelin e ulët (LOW).

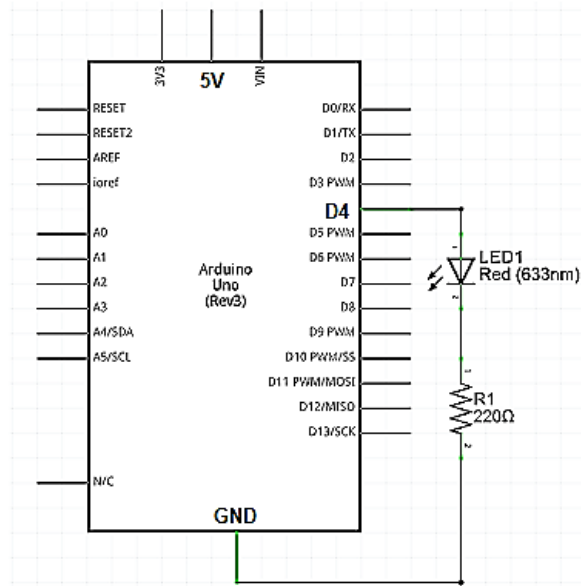


Figura 4.15. Lidhja e diodës led me Arduino Uno R3

Shembulli 4.14.

```

1 int ledDioda=4;           // ledDioda është emri simbolik i pinit të katërt.
                           //
2 digitalWrite(ledDioda,HIGH); // Pini i katërt vendoset në nivel të lartë dhe dioda
                           // led ndriçon.

```

Shembulli 4.15. paraqet program për përdorimin e tre instruksione për punën me pinat digjital. Vetëm kur shtypet tasteri, dioda led ndriçon.

Shembulli 4.15.

```

1 int ledPin = 13;         // Dioda led është e lidhur në pin/-in e trembëdhjetë
                           // të Arduino Uno.
2 int inPin = 7;          // Tasteri është i lidhur në pinin e shtatë.
3 int val = 0;            // Variabla numër i plotë val e ruan vlerën e
                           // lexuar.
4 void setup() {          //
5   pinMode(ledPin, OUTPUT); // Konfigurimi i pinit të trembëdhjetë si
                           // dalës
6   pinMode(inPin, INPUT); // Конфигурација на седмиот пин како влезен
7 }                        //
8 void loop() {           //
9   val = digitalRead(inPin); // Leximi i pinit hyrës.
10  digitalWrite(ledPin, val); // Pini në të cilin është e lidhur dioda led ka vlerë
                           // të njëjtë si edhe pini i lidhur me
                           // tasterin.

```

Në njësinë mësimore 4.2. Pjesët përbërëse të mikrokompjuterit në pllakë Arduino Uno u njohëm me hyrjet dhe daljet analoge të tij. Theksuam se për pinat analog me rëndësi të madhe është modulimi në gjerësi i impulsit.

analogRead(pin)

→ Me këtë instruksion lexohet **vlera e një hyrjeje analoge**. Tensioni në hyrjen analoge mund të ketë pafundësisht shumë vlera të ndryshme dhe lëviz në brezin nga 0 deri në 5V. Konvertori analogo-digjital, që gjendet në përbërjen e Arduino Uno R3, i shndërron vlerat analoge në numra të plotë nga 0 deri në 1023. Numër i plotë më pas paraqitet si kod binar me 10 bitë. Nëse vlera 5V ndahet në 1024 pjesë, atëherë një pjesë do t'i korrespondojë një tensioni prej 4.9 mV. Kështu, të gjitha vlerat e tensionit nga 0V deri në 4.9mV do të korrespondojnë me numrin 0, nga 4.9mV deri në 9.8mV me numrin 1, nga 9.8mV deri në 14.7mV me numrin 2 dhe

kështu me radhë, deri te tensioni 5V, i cili do t'i korrespondojë numrit 1023.

analogWrite(pin,value) → Me daljet analoge mund të kontrollohet intensiteti i dritës së diodës LED ose shpejtësia e rrotullimit të motorit. **Në pinin dalës regjistrohet vlerë analoge dhe e njëjta shndërrohet në një seri zerosh dhe njëshash.** Gjerësia, përkatësisht kohëzgjatja e impulseve dhe madhësia e tensionit të daljes varen në proporcion të drejtë nga vlera e regjistruar numër i plotë, e cila sillet në brezin prej 0 deri në 255 (figurën 4.5).

Në shembullin 4.16, në pinin dalës numër 9 është i lidhur dioda led, intensiteti i dritës së të cilit varet nga pozicioni i rrëshqitësit të potencialometrit që është i lidhur me pinin e tretë të Arduino Uno R3. Në kodin programor me numër rendor 9 të të njëjtit shembull, vlera e lexuar nga potencialometri pjesëtohet me katër, pasi sinjali dalës ka një brez vlerash katër herë më të vogël se ai i sinjalit hyrës.

Shembulli 4.16.

```

1  int ledPin = 9;           // Dioda led është e lidhur me pinin e nëntë.
2  int analogPin = 3;       // Potencialometri është i lidhur me pinin e
                             // tretë.
3  int val = 0;             // val është variabli për ruajtjen e vlerës së
                             // lexuar.
4  void setup () {
5    pinMode(ledPin, OUTPUT); // Konfigurimi i pinin dalës.
6  }
7  void loop() {
8    val = analogRead(analogPin); // Leximi i vlerës së sinjalit hyrës.
                                     // Vlera e lexuar është në brezin prej 0 deri në
                                     // 1023.
9    analogWrite(ledPin, val / 4); // Vlera e regjistruar është në brezin prej 0 deri
                                     // në 255.
10 }
```

tone(pin, frequency) → Në pinin dalës 3 ose 11 të Arduino Uno R3 mund të lidhet një sinjalizues i cili do të gjeneroj një ton të njëtrajtshëm. Frekuencën e tonit e zgjedh programuesi duke përdorur instruksionin tone (pin, frequency, duration). Ekziston mundësia që të zgjidhet edhe kohëzgjatjen e sinjalit të prodhuar. Në

të kundërtën, ai do të zgjasë derisa të ekzekutohet instruksioni `noTone()`.

4.7.2. Instruksionet për komunikim serik

Pinat RX (angl. Receive) dhe TX (angl. Transmit) në Arduino Uno R3 janë pina për komunikim serik. Në figurën 4.16 është paraqitur mënyra e lidhjes së dy platformave Arduino Uno R3, ku pini RX i platformës së parë lidhet me pinin TX të platformës së dytë dhe anasjelltas.

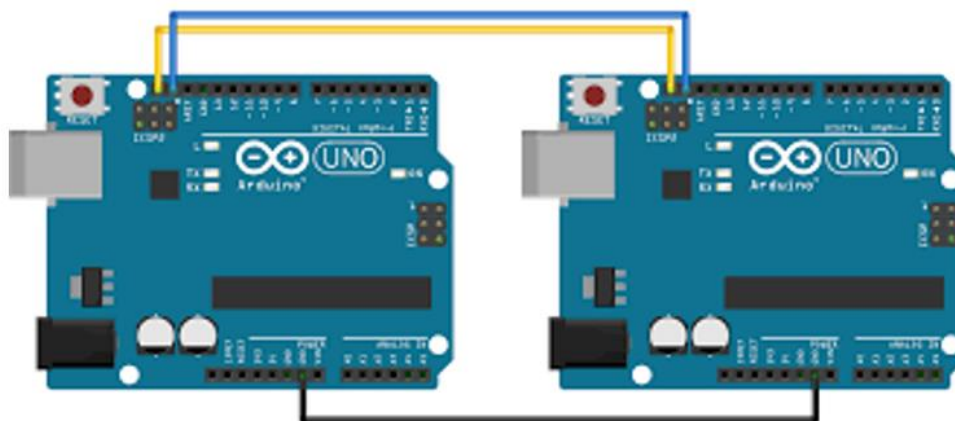


Figura 4.16. Lidhja serike ndërmjet dy platformave Arduino Uno

Përmes këtyre pinave, Arduino mund të lidhet me mikrokompjuterin Raspberry Pi, modul Wi-Fi, modul Bluetooth dhe me pajisje të tjera që mbështesin transmetimin serik. Një avantazh i madh i komunikimit serik është se shkëmbimi i të dhënave mund të monitorohet përmes monitorit të integruar serik. Të gjitha të dhënat që Arduino Uno R3 i pranon ose i dërgon me pajisjet tjera mund t'i shohim në ekran. Kuptohet, për këtë është e nevojshme që Arduino të jetë i lidhur me kompjuter përmes kablos USB dhe të jetë i hapur monitori serik në kuadër të mjedisit zhvillimor të integruar (IDE). Monitori serik është paraqitur në figurën 4.17.

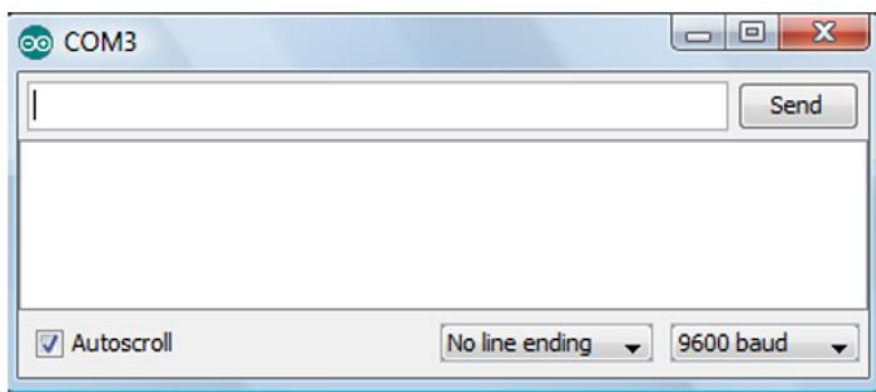


Figura 4.17. Monitori serik i mjedisit zhvillimor për mikrokompjuterin Arduino

Ekzistojnë rreth njëzet instruksione për punën me monitorin serik, por ne do të përmendim vetëm dy.

Serial.begin(shpejtësia) → Me këtë instruksion vendoset gjerësia e brezit të lëshimit përkatësisht shpejtësia e transmetimit serik, ndërsa ajo është e theksuar në këndin e poshtëm të djathtë të monitorit serik. Shpejtësia standarde është 9600 baud, që do të thotë 9600 bit për sekondë, por përdoruesi mund të ndryshojë shpejtësinë e transmetimit serik nëse këtë e kërkon pajisja e lidhur me pinat RX dhe TX.

Serial.print (x) → Në ekranin e monitorit serik **paraqitet vlera e variablit**. Pas instruksionit Serial.println(x) hapet një rresht i ri në dritaren e monitorit, ndërsa me Serial.print(x) të dhënat shfaqen në të njëjtin rresht.

Në shembullin 4.17, monitori serik tregon kohën e kaluar, të shprehur në milisekonda.

Shembulli 4.17

```

1 unsigned long koha;
2 void setup() {
3   Serial.begin(9600);
4 }
5 void loop() {
6   Serial.print("Koha e kaluar: ");
7   koha = millis();
8   Serial.println(koha);           // Pauza prej një sekonde me qëllim të zvogëlimit
9   delay(1000);                   // të numrit të të dhënave të paraqitura në
10  }                                // monitorin serik.
```

Në shembullin 4.18 Arduino Uno R3 e mat vlerën e tensionit analog të pinit A0, ndërsa monitori serik e shfaq të njëjtin. Të theksojmë se tensioni maksimal për pinat hyrës-dalës të Arduino Uno R3 është 5V dhe nuk lejohet të tejkalohet kjo vlerë. Në shembull është përdorur instruksioni analogRead. Pasi që konvertuesi analogo-dijital konverton vlerat analoge në numra të plotë nga 0 deri në 1023, për të marrë vlerën e tensionit të matur aplikohet formula:

$$U[V] = \frac{\text{Vlera e plotë i lexuar}}{1023} \cdot 5[V]$$

Shembulli 4.18

```

1 const float maxVolti = 5.0;           // Tensioni maksimal për pinin A0
4 void setup() {
5   Serial.begin(9600);                 // Shpejtësia e transmetimit serik
```

```

7 }
8 void loop() {
9   int vlera = analogRead(A0);           // Leximi i pinit hyrës.
10  float tensioni = (vlera /1023) * maxVolti; // Llogaritja e tensionit të matur
    Serial.println(tensioni);           // Shfaqja e rezultatit
11 }

```

4.7.3. Instruksionet për kontrollin e kohës

Kodet programore ekzekutohen me një shpejtësi shumë të madhe. Nëse në dalje lidhim një diodë led dhe ajo duhet të pulsojë, atëherë midis dy gjendjeve të saj (angl. ON/OFF) patjetër të futet një vonesë kohore (angl. delay). Në të kundërtën, përdoruesi nuk do ta vërejë që dioda led ndizet dhe fiket pandërprerë.

delay (ms) → **Koha e vonesës** është dhënë në milisekonda. Në një sekondë ka 1000 milisekonda ose 1,000,000 mikrosekonda.

Shembulli 4.19

```

1 int ledPin = 13;           // Dioda led është e lidhur me pinin 13-të.
2 void setup() {
3   pinMode(ledPin, OUTPUT); // Pini për diodën led konfigurohet si
4 }                           // dalje.
5 void loop() {
6   digitalWrite(ledPin, HIGH); // Dioda led kyçet.
7   delay(1000);                // Pauza prej një sekonde.
8   digitalWrite(ledPin, LOW);  // Dioda led shkyçet.
9   delay(1000);                // Pauza prej një sekonde.
10 }

```

time=micros() → Me këto dy instruksione matet koha e shprehur në mikrosekonda dhe milisekonda nga momenti kur Arduino Uno fillon të ekzekutojë programin. Për të shfaqur kohën e kaluar, duhet të aktivizohet monitori serik.

time=millis()

4.7.4. Instruksionet matematikore

max(x,y) → Instruksioni zgjedh numrin më të madh midis dy numrave x dhe y. Në shembullin 4.21, instruksioni nuk lejon që vlera e sensorit të jetë më e vogël se 20.

min(x,y) → Instruksioni zgjedh numrin më të vogël midis dy numrave x dhe y. Në shembullin 4.20, instruksioni nuk lejon që vlera e sensorit të jetë më e madhe se 100.

constraim(x,a,b) → Instruksioni kufizon brezin e variablës x; a është vlera minimale, ndërsa b është vlera maksimale.

Shembulli 4.20.

```
1 sensVal = min(sensVal, 100); // Nëse variabla sensVal ka një vlerë më të vogël
// se njëqind atëherë ajo e ruan atë vlerën e saj.
// Nëse vlera është më e madhe se njëqind,
// atëherë vlera e variablës sensVal barazohet me
// vlerën 100.
```

Shembulli 4.21

```
1 sensVal=max(sensVal, 20); // Nëse variabla sensVal ka një vlerë më të madhe
// se 20, atëherë ajo e ruan vlerën e saj. Nëse vlera
// është më e vogël se 20, atëherë variabla sensVal
// barayohet vlerën 20.
```

map(vlera, fromLow,fromHihg,toLow, toHigh) → Instruksioni e ndryshon brezin e vlerave të variablës. fromLow dhe fromHigh janë vlerat minimale dhe maksimale të brezit të vjetër, ndërsa toLow dhe toHigh janë vlerat minimale dhe maksimale të brezit të ri.

Në shembullin 4.22. ndryshohet brezi i vlerave të sinjalit analog. Në fillim, kodimi i mostrave hyrëse kryhet me 10 bitë dhe brezi është një nga 0 deri në 1023, por më pas e ndryshojmë brezin nga 0 deri deri në 255, që kodimi të jetë me 8 bitë.

Shembulli 4.22.

```
1 void setup(){
2 }
3 void loop() {
4   int val = analogRead(0); // Lexohet vlera analoge nga pini 0 dhe ruhet
// në variablën val. Kur do të ndryshohet brezi,
5   val=map(val, 0, 1023, 0, 255); // ndryshon edhe variabla Pasi që brezi u
// zvogëlua, presim që edhe vlera të
// zvogëlohet proporcionalisht.
//
6   analogWrite(9, val); // Vlera e re e fituar ruhet në pinin dalës
// numër 9.
7 }
```

4.7.5. Instruksionet për Bit dhe Bajt

bitClear (x,n) → Instruksioni riseton (e vendos në zero) bitin me numrin rendor n në variablën x. Nëse variabla e numrit të plotë është në sistemin numerik decimal, duhet të konvertohet në sistemin numerik binar. Biti me peshën më të vogël (i pari nga e djathta) është në pozicionin zero. Rezultati nga ky instruksion do të jetë variabla x me bitin n të risetuar.

bitSet (x,n) → Instruksion e riseton (e vendos në nivel të lartë) bitin me numrin rendor n në variablën x.

bitRead (x,n) → Lexohet vlera e bitit të n nga variabla x. Rezultati do të jetë një bit 0 ose 1, në varësi të vlerës së lexuar.

bitWrite (x,n,b,) → Ky instruksion ka tre parametra. X është variabël numër i plotë, n është numri rendor i bitit dhe b vlera e bitit që duhet të shkruhet. Në shembullin 4.23., me instruksionin `bitWrite (x,0,1)` ndryshohet gjendja e bitit me peshën më të vogël nga zero në një.

bit (n) → Me këtë instruksion llogaritet pasha e bitit në varësi nga numri rendor i tij, gjejjësisht pozicioni i tij në sistemin numerik binar. Pasha e bitit llogaritet sipas shprehjes $2n$. Kështu, duke filluar nga e majta në të djathtë, biti i parë ka peshën $2^0=1$, i dyti $2^1=2$, i treti $2^2=4$, i katërti $2^3=8$ etj.

Shembulli 4.23

```

1 void setup() {
2   Serial.begin(9600); // Përshtatja e brezit të lëshimit të monitorit serik.
3   byte x = 0b10000000; // Prefiksi 0b përdoret për variabla binare.
4   Serial.println(x, BIN); // Monitori serik e tregon numrin binar
                          // 10000000.
5   bitWrite(x, 0, 1); // Shkrimi i 1 në pozicionin e bitit me peshë më të
6   Serial.println(x, BIN); // vogël.
                          // Monitori serik e tregon numrin binar
                          // 10000001.
7 }
8 void loop(){
9 }

```

highByte (x) → Nëse e dhëna është 16-bitëshe, me këtë instruksion ndahen 8 bitët më të rëndësishëm.

lowByte (x) → Nëse e dhëna është 16-bitëshe, me këtë instruksion ndahen 8 bitët më pak të rëndësishëm.

Shembulli 4.24.

```

1 int test = 0xABCD;           // Deklarojmë një variabël 16-bitëshe në
                               // sistemin numerik heksadecimal.
2 byte hi, lo;                // Deklarojmë variabla dy bajtëshe.
3 hi = HighByte(test);        // Fitojmë rezultat 0xAB.
4 lo = LowByte(test);         // Fitojmë rezultatat 0xCD.
```

4.8. Strukturat në gjuhën programore C/C++ për mikrokompjuterin në pllakë Arduino

Strukturat janë blloqe instruksionesh me funksione të definuara saktë. Blloqet e instruksioneve janë të shkruajtura mes kllapave të mëdha { }. Kjo ndarje e programit në struktura dhe përdorimi i kllapave të mëdha e bën analizën e programeve më të lehtë.

4.8.1. Strukturat për zgjedhjen e mundësive

Strukturat për zgjedhjen e mundësive përfshijnë deklaratat: if, else if, ndërsa else.

if (kushti) { → **Deklarata if e kontrollon kushtin dhe nëse ai është i plotësuar (i vërtetë), atëherë instruksionet brenda bllokut ekzekutohen.** Kushti mund të jetë një shprehje logjike ose një shprehje me operatorë krahasimi. Sintaksa është e tillë që kushti shkruhet në kllapa të vogla, ndërsa blloku i instruksioneve shkruhet në kllapa të mëdha. Në figurën 4.18. është treguar bllok diagram ii kësaj strukture.

```

instruksioni 1
instruksioni 2
instruksioni 3
.....
}
```

if (kushti1) { → Struktura if... else mundëson kontroll më të mirë të kodit sepse kryen **kontrollin e shumë kushteve**. Nëse kushti në deklaraten if nuk plotësohet, kontrollohet kushti në deklaraten else if. Numri i deklaratave else if mund të jetë i pakufizuar. Çdo kusht jo i vërtetë rezulton me një kontroll të ri, derisa nuk merret një kusht i vërtetë. Kur do të merret një kusht i vërtetë, ekzekutohen instruksionet e tij dhe ndërpritet kontrolli i kushteve të ardhshme. Nëse nuk merret asnjë kusht i vërtetë, ekzekutohen instruksionet nën deklaraten else. Deklarata else është deklarata e fundit në strukturën për zgjedhjen e

```

//blloku i
//instruksioneve 1
}
else if (kushti2) {
//blloku i
//instruksioneve 2
}
else {
```

```
//blloku i
//instruksioneve 3
}
```

mundësive dhe mund të vërehet se pas deklaratës else nuk ka kushte për kontroll. Në figurën 4.18 është paraqitur bllok diagrami i strukturës if, ndërsa në figurën 4.19 është paraqitur bllok diagrami i strukturës if...else dhe për dallim nga struktura if, ekziston bllok instruksionesh që ekzekutohen edhe kur kushti if nuk është plotësuar.

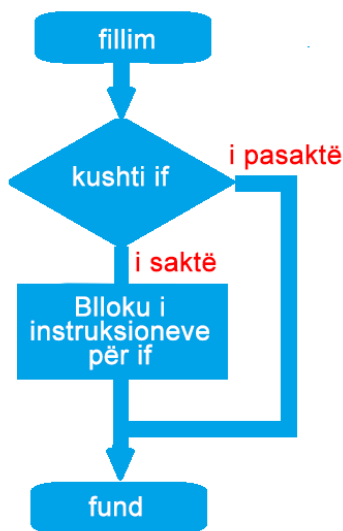


Figura 4.18. Bllok diagrami i strukturës if

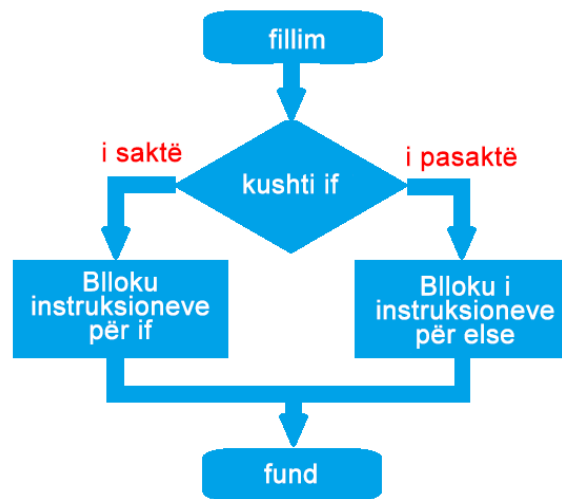


Figura 4.19. Bllok diagrami i strukturës if...else

Shembulli 4.25. është shembull për zbatimin e strukturës për zgjedhje të tri mundësive: temperatura më e madhe se 70 shkall, temperatura mes 60-70 shkallësh dhe temperatura më e vogël se 60 shkallë.

Shembulli 4.25.

```
1 if (temperature >= 70) {
2   // Rrezik! Shkyçe sistemin.
3 }
4 else if (temperature >= 60 && temperature < 70) {
5   // Kujdes! Kontrolllo vlerat e parametrave të tjerë.
6 }
7 else {
8   // temperatura është më e ulët se 60 shkallë.
9   // Sistemi funksionon në mënyrë të sigurt.
10 }
```

Në shembullin 4.26., kur tasteri nuk është i dioda led ndriçon, ndërsa kur i njëjti do të shtypet dioda ndalon të ndriçoj. Supozojmë se kur tasteri është i shtypur, pini i tij është në nivel të lartë dhe se dioda është me anodë aktive.

Shembulli 4.26.

```

1 int val = digitalRead(taster); // Kontrolli gjendjes së tasterit.
2   if (val == LOW) {           // E testojmë vërtetësinë e kushtit
3     digitalWrite(led, HIGH); // Pini i diodës vendoset në nivel të lartë.
4   }

```

Në shembullin 4.27. si komponentë hyrëse janë përdorur dy tasterë. Në kushtin e strukturës if është përdorur instruksioni logjik DHE, ndërsa për t'u ekzekutuar struktura if pinat e të dy tasterëve duhet të jenë të vendosur në nivel të lartë.

Shembulli 4.27.

```

1   if (tasterA == HIGH && tasterB ==HIGH) {
    // Kontrolli i gjendjes së tasterëve dhe testimi i i saktësisë së kushtit .
2     digitalWrite(led, HIGH); // Dioda led ndriçon.
3   } //
4   else { // Kodi programor nën strukturën else
    // ekzekutohet nëse nuk plotësohet kushti
    // nën strukturën if.
5     digitalWrite(led, LOW); // Dioda led nuk ndriçon.
6   }

```

Struktura për zgjedhjen e mundësive mund të krijohen edhe përmes deklaratave **switch...case**.

<pre> switch (variabla) { case 1: // bllok instruksionesh break; case 2: // bllok instruksionesh break; default: // bllok instruksionesh break; } </pre>	<p>→ Ngjashëm si deklarata if, deklarata switch...case kontrollon rrjedhën e programit, duke lejuar që të krijohen blloqe të ndryshme të instruksioneve për raste të ndryshme (angl. case). Ngjitur me deklaratën switch, në kllapa të vogla është theksuar emri i ndryshores nfa vlera e të cilës varet në cilin rast do të kalohet (angl. switch) ekzekutimi i programit. Fjala në anglisht default në përkthim do të thotë “standard “ ndërsa kodi pas deklaratës default ekzekutohet nëse nuk ekzekutohet asnjëri prej rasteve të më sipërme. Deklarata break përdoret së bashku me deklaratën switch dhe shënon fundin e çdo rasti. Pa deklaratën break, struktura do të vazhdoj të ekzeklutohet deri në përfundimin e saj.</p>
--	--

Në shembullin 4.28. si komponentë hyrëse është përdorur sensor, fotorezistor i cili është lidhur me hyrjen analoge A0. Të theksojmë, sinjalet analoge nga sensorët janë sinjale hyrëse për konvertorin analogo-digjital i cili në daljen e tij gjeneron vlera me numër të plotë në brezin nga 0 deri në 1023. Me supozimin që fotorezistorin do ta përdorim në hapësirë të mbyllur, vlera maksimale është zvogëluar në 600. Në këtë shembull struktura switch zëvendëson diodstruktura if. Në program lexohet vlera e sensorit, pastaj me përdorimin e instruksionit map() ndryshohet numri në

vlera të mundshme në gjithsej tre dhe në fund shfaqet njëri prej tre mesazheve në monitorin serik, në varësi të vlerës së fituar y.

Shembulli 4.28.

```

1  const int sensorMin=0;           // Definohet vlera maksimale dhe
2  const int sensorMax=600;        // minimale e sensorit.
                                   //
3  void setup() {
4      Serial.begin(9600);         // Përshtatet brezi i lëshimit imonitorit
5  }                                // serik.
6  void loop() {
7      int x=analogRead(A0);       // Lexohet vlera e sensorit
8      int y=map(x, sensorMin , sensorMax, 0,2);
                                   //Kryhet zvogëlimi i brezit të vlerave të sensorit. Numri i vlerave të
                                   //mundshme zvogëlohet nga 0-600 në 0-2.
9      switch(y) {                 // Fillon struktura switch.
10     case 0:
11         Serial.println("errësirë");
12     break;
13     case 1:
14         Serial.println("gjysmëerrësirë");
15     break;
16     case 2:
17         Serial.println("dritë");
18     break;
19 }
20 delay(10);                       // Për punë stabile futet vonesë prej
21 }                                  // 10 milisekonda mes çdo leximi të
                                   // dytë të sensorit.

```

4.8.2. Strukturat për përsëritje (strukturat ciklike)

Strukturat për përsëritje të ciklit përbëhen nga deklaratat: while, do... while dhe for. Së bashku me këto deklarata mund të përdoret edhe deklarata continue.

while (kushti) { → Fjala angleze while do të thotë "derisa", ndërsa mendohet në instruksioni 1 vërtetësinë e kushtit në kllapa të vogla. Derisa kushti është i vërtetë, cikli do të përsëritet, gjegjësisht blloku i instruksioneve poshtë deklaratës while do të ekzekutohet disa herë. Duhet pasur kujdes që cikli while të mos jetë i pafund. Kjo do të ndodhë nëse variabla në kushtin nuk ndryshon me instruksionet në kllapën e madhe të strukturës.

```

instruksioni 1
instruksioni 2
instruksioni 3
...
}

```

Për këtë qëllim, variabla në kushtin duhet të ndryshohet përmes zvogëlimit ose zmadhimit ose të jetë e varur nga gjendja e ndonjë sensori.

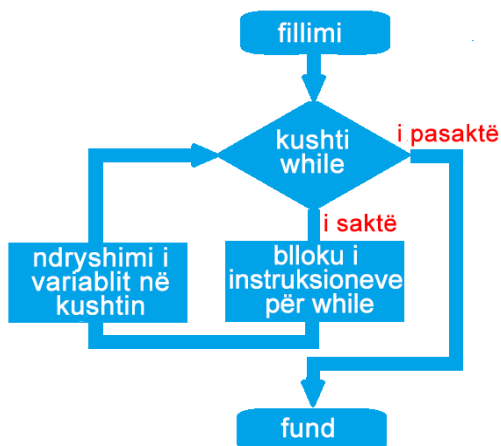


Figura 4.20. Bllok diagrami i strukturës while



Figura 4.21. Bllok diagrami i strukturës do.... While

Në shembullin 4.29. dioda pulson derisa sa vlerën që e ka matur sensori është më e madhe se 100.

Shembulli 4.29.

```
// Së pari kryhet shndërrimi analogo-digjital, pastaj verifikohet
// saktësia e kushtit në ciklin while.
1 while(analogRead(sensor) > 100) {
2   digitaWrite(LED, HIGH);           // Dioda led ndriçon
3   delay(100);                       // Pritet 100 ms.
4   digitaWrite(LED, LOW);            // Dioda led nuk ndriçon.
5   delay(100);                       // Pritet 100 ms.
6 }
```

do{ → Dallimi midis deklaratave do...while dhe while është se në instruksioni 1 deklaratën do...while, vërtetësia e kushtit kontrollon në instruksioni 2 fund të ciklit. Kjo do të thotë që cikli duhet të ekzekutohet një instruksioni 3 herë, pavarësisht vërtetësisë së kushtit. Figura 4.20. dhe figura 4.21. paraqesin krahasim midis dy strukturave while
} while (kushti) dhe do...while.

Në shembullin 4.30. përsëri si komponentë hyrëse shfrytëzohet sensori i lidhur në pinin analog A3. Cikli do...while përsëritet derisa në dalje të konvertorit analogo-digital nuk fitohet vlerë më e madhe se 100.

Shembulli 4.30.

```
1 int x = 0;
```

```

2 int Sensor=3;
3 do {
4 delay(50);           //  Sensorit i jepet koha e nevojshmeqë të kryej
                        //  detektimin.
5 x = analogRead(Sensor); //  Leximi i vlerës së sensorit.
6 } while (x < 100);   //  Verifikimi i saktësisë së kushtit.
    
```

for (deklarimi i variable; kushti; hapi me të cilin bëhet ndryshimi)
{

instruksioni 1
instruksioni 2
instruksioni 3

.....
}

→ Deklarata for përdoret për të përsëritur disa herë një bllok instruksionesh të shkruara në kllapa të mëdha. Zakonisht përdoret një numërues, vlera e të cilit vazhdimisht rritet dhe kur do të arrij maksimumin, ndalon ekzekutimi i ciklit. Shpesh herë struktura për përsëritje të ciklit me deklaratën for përdoret për konfigurim të njëjtë të më tepër pinave ose për përpunimin e më tepër të dhënave në mënyrë identike.

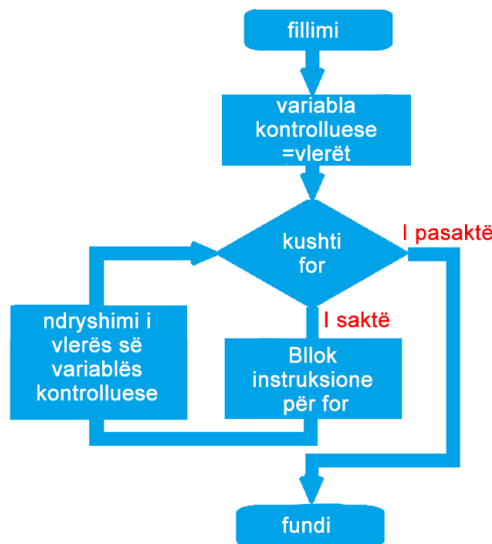


Figura 4.22. Bllok diagrami për strukturën for

Në shembullin 4.31. dioda led lidhet me një të daljet analoge që ofrojnë impulse të moduluara në gjerësi. Ashtu siç rritet vlera e daljes, nga 0 në 255, ashtu rritet edhe kohëzgjatja e impulseve dhe vlera mesatare e tensionit të daljes, ndërsa rrjedhimisht edhe intensiteti i dritës.

Shembulli 4.31.

```

1 int PWMpin = 10;           //  Dioda led është e lidhur në pinin e dhjetë.
2 void loop() {             //
3   for (int i = 0; i <= 255; i++) { //  Deklarimi i variablës për strukturën for,
                                //  kushti për ekzekutimin e saj dhe instruksioni
                                //  për hapin me të cilin kryhet ndryshimi i
                                //  variablës.
    
```

```

4   analogWrite(PWMPin, i);    // Vlera e variablës i regjistrohet në pinin e
                                   // daljes.
5   delay(10);
6   }
7 }

```

continue —————> Me deklaratën continue me kusht mund të anashkalohen disa cikle në strukturat for, while dhe do...while.

Shembulli 4,32, është shembull për zbatimin e urdhërit continue. Në këtë shembull anashkalohen vlerat nga 41 deri në 119, krahasuar me shembullin 4.31.

Shembulli 4.32.

```

1   for (int x = 0; x <= 255; x ++ ) {
2     if (x > 40 && x < 120) {           // Anashkalimi i vlerave.
3       continue;
4     }
5     analogWrite(PWMPin, x);
6     delay(50);
7   }

```

Shembulli 4.33. është kombinim i dy strukturave, for dhe if. Cikli for mund të përsëritet katër herë, por kusht është që pini i dytë digjital të jetë në nivel të ulët.

Shembulli 4.33.

```

1   for (int i=0; i < 4;)                // Deklarimi i variablës dhe kushti i përsëritjes së
                                   // ciklit.
2   {                                    // Fillimi i strukturës for.
3     if(digitalRead(2) == LOW) {      // Kushti dhe fillimi i strukturës if.
4       i++;
5     }                                  // Fundi i strukturës if.
6   }                                    // Fundi i strukturës for.

```

4.8.3. Strukturat për degëzim

Deri tani, programet janë ekzekutuar linearisht, gjegjësisht instruksionet ishin të renditur në mënyrë të njëpasnjëshme, njëri pas tjetrit. Me strukturat për degëzim ndryshon rrjedha e programit, mund të anashkalohen pjesë të kodit programor dhe në atë mënyrë të kalohet para- prapa nëpër program.

goto shenja; —————> Strukturat për degëzim realizohen përmes deklaratës goto, që në përkthim do të thotë "shko në". Deklarata goto shenja: pasohet nga një shenjë (angl. label), e cila në fakt është emër simbolik i vendit ku duhet të bëhet kërcimi.

Në shembullin 4.34. janë përdorur më tepër struktura: diodstruktura for, një strukturë if dhe një strukturë goto. Ky është shembull i ndërthurjes së një strukture në tjetrën. Nëse plotësohet kushti i strukturës if, atëherë do të ekzekutohet struktura për degëzim, do të tejkalohej blloku i instruksioneve me numër 1 dhe drejtpërdrejtë do të ekzekutohet blloku i instruksioneve numër 2. Nëse kushti i strukturës if nuk plotësohet, atëherë së pari ekzekutohet blloku i instruksioneve me numër 1 edhe atë disa here. Sa here do të ekzekutohet blloku i instruksioneve me numër 1 varet nga parametrat në strukturat for për përsëritje dhe është i barabartë me $r \times g \times b$ herë.

Shembulli 4.34.

```

1 for (byte r = 0; r < 255; r++) {
2   for (byte g = 255; g > 0; g--) {
3     for (byte b = 0; b < 255; b++) {
4       if (analogRead(0) > 250) {
5         goto bailout;
6       }
7       // blloku i instruksioneve numër 1
8     }
9   }
10  }
11  bailout:
12  // blloku i instruksioneve numër 2

```

4.9. Instruksionet për punën me bibliotekat

Ashtu si edhe shumica e platformave të programimit, edhe mjedisi zhvillimor Arduino mund të zgjerohet duke përdorur biblioteka. **Bibliotekat janë nënprograme që mundësojnë komunikimin midis pajisjeve periferike dhe mikrokompjuterit Arduino.** Ato përmbajnë instruksione me të cilat reduktohet shkrimi i kodeve programore të gjata dhe dukshëm lehtësohet puna. Sipas funksionalitetit, bibliotekat mund të ndahen në disa grupe: komunikim (shembull Wi-Fi, Ethernet, GSM), përpunim të të dhënave, ruajtje të të dhënave (shembull Cloud Data, puna me kartelat SD), displej (shembull LCD), komandimi me pajisje (shembull motorë), sensorë (shembull Temperature, optic, UV, shtypje), tajming etj. Mjedisi zhvillimor Arduino përmban disa biblioteka standard dhe ato mund t'i përdorim duke klikuar (angl. click) në **Sketch** → **Import Library** në vetë menynë. Nëse mjedisi zhvillimor nuk e përmban bibliotekën e dëshiruar atëherë ajo mund të merret në internet.

#include < emri > → Me këtë instruksion thirret biblioteka në programin kryesor, me ç'rast duhet të theksohet emri i bibliotekës.

Ne do të njihemi me bibliotekat në vazhdim: [LiquidCrystal](#), Servo dhe CapacitiveSensor .

Ekranet LCD (angl. Liquid Crystal Display) janë komponentë dalëse dhe shërbejnë për paraqitjen vizuale të rezultateve të fituara nga ekzekutimi i programit. Në figurën 4.23. është paraqitur pamja e LCD ekranit alfanumerik me dy rreshta me nga 16 simbole. Në të njëjtën figure është paraqitur edhe pin diagrami i tij.

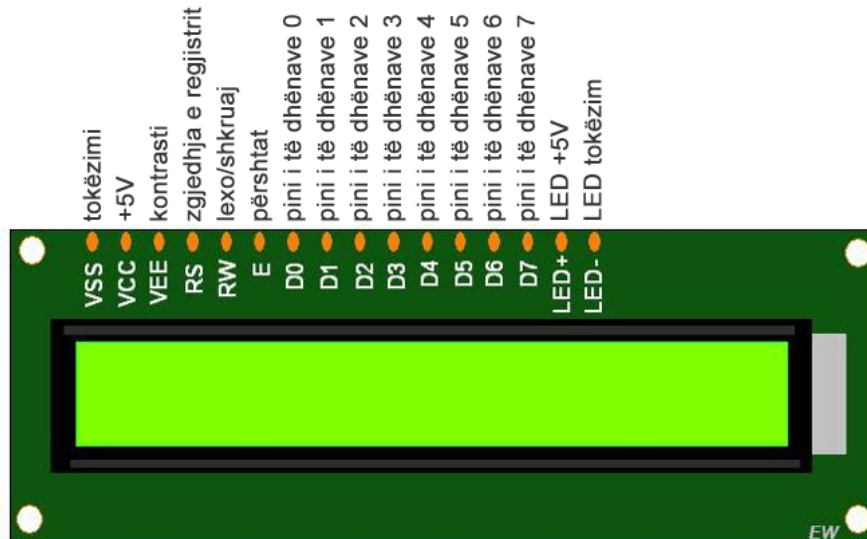


Figura 4.23. pamja e jashtme e LCD ekranit dhe pin diagrami i tij

Biblioteka LiquidCrystal është bibliotekë standarde në përbërje të mjedisit zhvillimor dhe përmes saj Arduino menaxhon me LCD-ekranin. Kjo bibliotekë përmban 20 instruksione të ndryshme, por ne do të sqarojmë vetëm tre, të cilat janë më të përdorura. më poshtë janë përshkruar vetëm tre më të përdorurat:

LiquidCrystal (rs, rw, enable, d4, d5, d6, d7) → Me këtë instruksion definohepinat e Arduino-s të cilët përdoren për lidhjen me daljen me të njëtin emër të LCD ekranit. Ky instruksion vjen pas kyçjes së bibliotekës në programin siç është theksuar në shembullin 4.35.

lcd.begin(kol,red) → Me këtë instruksion fillon komunikimi me ekranin dhe shënohet numri i kolonave dhe rreshtave prej simboleve të cilët mund të paraqiten në ekran.

lcd.print (tekst) → Ky instruksion ka vetëm një parametër, e ky është teksti që duhet të paraqitet në LCD ekranin.

Shembulli 4.35.

```
1 #include <LiquidCrystal.h> // Thirja e bibliotekës për punë me LCD
// ekranin.
```

```

2 LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2); // Zgjedhja e pinave për lidhjen e
// Arduino Uno R3 me LCD ekranin.
//
3 void setup() {
4   lcd.begin(16,1); // Ekranin ka 16 lkolona dhe një
// rresht.
5   lcd.print("hello, world!");
6 }
7 void loop() {}

```

Në pjesën praktike të kësaj njësie modulare është paraparë një ushtrim praktik me përdorimin e një ekran LCD dhe atëherë do të njihemi më në detaje me mënyrën e lidhjes së tij me Arduino Uno R3 dhe programin për shfaqjen e tekstit në ekran.

Servo motori është lloj i veçantë i motorit që nuk rrotullohet në rreth, por bën zhvendosje këndore dhe qëndron në një pozicion të caktuar deri në instruksionin e ardhshëm. Zakonisht, servo motori rrotullohet për 180 shkallë. Servo motorët shumë shpesh përdoren në robotikë, për shembull, për lëvizjen e krahut të robotit.

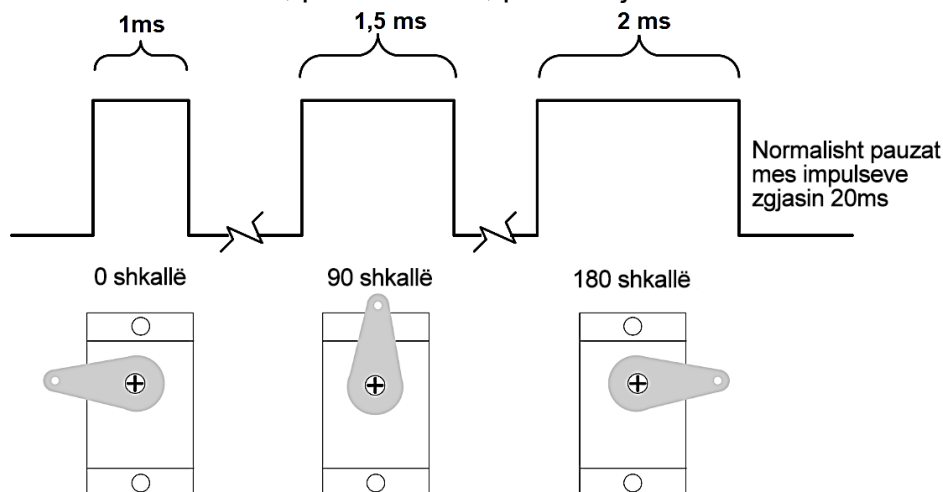


Figura 4.24. Varësia e këndit të rrotullimit të servo motorit nga gjerësia e impulseve

Që një servo motori të rrotullohet për një kënd të caktuar, ai ka nevojë për impulse të moduluara në gjerësi, gjerësia e të cilave varion nga një deri në dy milisekonda. Për shkak të ndjeshmërisë së servo motorëve, në vend të instruksionit analogWrite dhe impulseve të tij të moduluar në gjerësi shfrytëzohet biblioteka Servo dhe instruksionet e saj.

include (Servo.h) → Me këtë instruksion përfshihet biblioteka që lejon Arduino Uno të menaxhojë me servo motorin.

Servo emri → Krijojmë një objekt servo në të cilin do të ekzekutohen funksionet që përfshihen në bibliotekën Servo. Ashtu si

edhe variablat, edhe objekteve u jepen emra simbolikë, unikë.

- servo.attach (pin, min, max)** → Pin është numri i pinit të mikrokompjuterit Arduino në të cilin është i lidhur servo motori. **Vetëm pinat 9 dhe 10 kanë mbështetje për lidhje me servo motorit.** Min dhe max nuk janë parametra të domosdoshëm dhe paraqesin gjerësinë e impulseve, të shprehur në mikrosekonda, për këndin minimal dhe maksimal të rrotullimit të servo motorit. Sa më e madhe të jetë gjerësia e impulsit, aq më i madh është edhe këndi.
- servo.write(agol)** → Këndi i rrotullimit të servo motorit mund të ketë vlerë nga 0 deri në 180 gradë. Kur këndi është 0, atëherë zhvendosja është më e madhe në njërin drejtim, ndërsa për kënd prej 180° zhvendosja është më e madhe në drejtimin tjetër. Për 90 shkallë nuk ka zhvendosje.

Në shembullin 4.36. janë përdorur instruksionet e mësipërme, me ç'rast për lidhjen e servo motorit është zgjedhur pini i nëntë dhe servo motori është në mes, në pozicionin 90 shkallë.

Shembulli 4.36.

```

1 #include <Servo.h>
2 Servo myservo;           // Krijojmë servo objekt me emrin simbolik myservo
                           //
3 void setup(){
4   myservo.attach(9);
5   myservo.write(90);     // Servo motori është i pozicionuar saktë në mes.
6 }                         //
7 void loop() {}

```

Në pjesën praktike është paraparë një ushtrim praktik duke përdorur një servo motor.

Biblioteka CapacitiveSensor lejon dy ose më shumë pina të Arduino Uno R3 të shndërrohen në një sensor kapacitiv për zbulimin e kapacitetit të trupit të njeriut. Për këtë qëllim është e nevojshme që mes pinit të hyrjes dhe daljes të lidhet një rezistor dhe një copë folio prej alumini në fund të pinit të hyrjes. Nëse afër copës së folisë afrohet njeriu, trupi i tij do të absorbojë pjesë të elektricitetit, me ç'rast do të ndryshojë gjendja e njërit prej pinave. Arduino Uno e mat kohën e nevojshme që të barazohet gjendja e të dy pinave. Nëse trupi absorbon sasi më të madhe të elektricitetit, atëherë edhe koha do të jetë më e gjatë.[4]

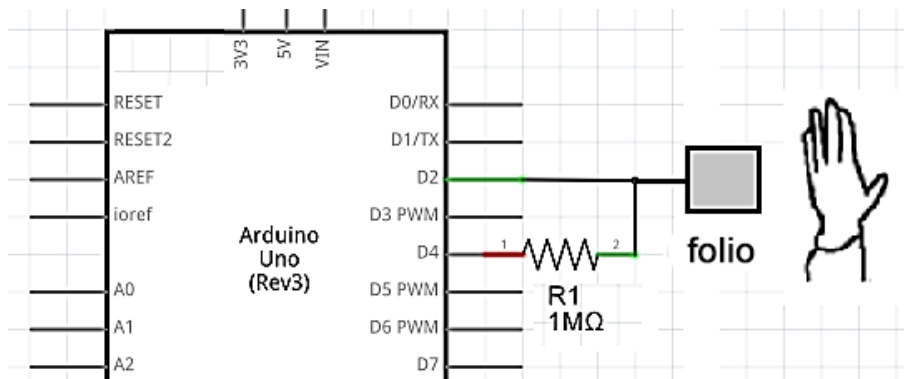


Figura 4.25. Aplikimi i bibliotekës CapacitiveSensor

Do të njihemi me tre instruksione të bibliotekës CapacitiveSensor.

include (CapacitiveSensor.h) → Me këtë instruksion kyçet biblioteka e cila mundëson leximin e vlerës së kapacitetit të sensorit.

CapacitiveSensor ime (pin sender, pin reciver) → Krijojmë një objekt me emër simbolik. Emri i objektit specifikohet përpara çdo instruksioni të përdorur nga biblioteka CapacitiveSensor. Gjithashtu, me këtë instruksion konfigurojmë pinat që do të lidhen me folion e aluminit. Midis tyre vendoset një rezistor me rezistencë nga 1MΩ deri në 40MΩ. Ndjeshmëria e sensorit rritet me rritjen e rezistencës. Me rezistencë prej 1MΩ, sensori funksionon si një sensor prekjeje dhe me një rezistencë 10MΩ si një sensor afërsie, me një distancë maksimale deri në 15 cm.

CapacitiveSensor (numri i mostrave) → Ky është instruksioni për leximin e vlerës së sensorit. Para instruksionit theksohet emri i objektit, i ndarë me pike nga instruksioni. Në kllapa të vogla shkruhet numri i mostrave i cili zakonisht është 30.

Për shembullin 4.37. është përdorur sensor kapacitiv, i cili është i lidhur me pinin e dytë dhe të katërt të Arduino Uno R3. Nëse e prekim folion, atëherë vlerat në ekranin e monitorit serik duhet të jenë më të mëdha.

Shembulli 4.37.

```

1 #include <CapacitiveSensor.h> // Thirja e bibliotekës për punë me sensor
// kapacitiv.
2 CapacitiveSensor senPrekje = CapacitiveSensor(4,2);

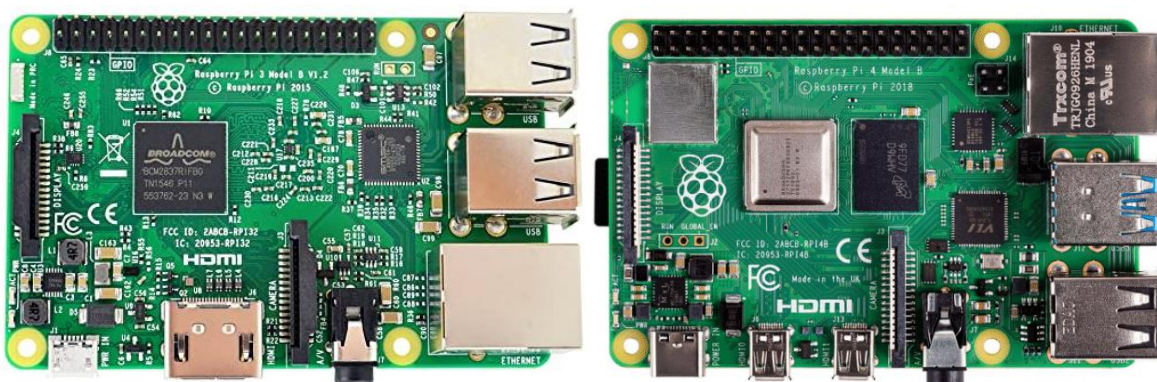
```

```
3 //Krijimi i objektit me emrin simbolik senPrekje. Zgjedhja e pinave për lidhjen
//Arduino Uno me sensor kapacitiv
4 void setup() { // Zgjedhim brez lëshimi për monitorin serik
5     Serial.begin(9600);
6 }
7 void loop() {
8     long sensorValue = senDopir.capacitiveSensor(30);
9     // Читање на вредност на сензор.
10    Serial.println(sensorValue); // Paraqitja e vlerës së sensorit në monitorin
11 } // serik.
```

4.10. Mikrokompjuteri në pllakë Raspberry Pi

Raspberry Pi është konkurrenti më i madh i Arduino. Gjenerata e parë e Raspberry Pi doli në treg në vitin 2012, në Angli. Dëshmi e popullaritetit të tij është se rreth gjysmë milioni kopje u shitën në gjashtë muajt e parë. Ngjashëm si Arduino, Raspberry Pi ishte destinuar për studentët e shkencave kompjuterike në Universitetin e Kembrixhit, me qëllimin që programimi të ishte më i lehtë. Sot, Raspberry Pi ka një rëndësi të madhe shkencore, edukative dhe praktike.

Edhe pse janë konkurrentë, Raspberry Pi dallohet shumë nga Arduino. Sipas karakteristikave të tij Raspberry Pi është shumë i ngjashëm me kompjuterin personal dhe mund të lidhet me një monitor, tastierë dhe maus. Raspberry Pi disponon me një **sistem operativ** funksional (zakonisht Linux), prandaj mund të ekzekutojë programe komplekse. Disavantazh është se softueri për Raspberry Pi nuk është me burim të hapur, siç është rasti me Arduino. Mjedisi zhvillimor i Raspberry Pi nuk përmban programe të gatshme falas dhe biblioteka me drajverë, por për këto kërkohen licenca të veçanta. Arduino lidhet shumë lehtë me sensorë dhe pajisje ekzekutuese, ndërsa që Raspberry Pi të lidhet në mënyrë efikase me një sensor, kërkohet softuer shtesë. Nga ana tjetër, të gjitha modelet e Raspberry Pi kanë një **kartelë grafike** e cila e bën shumë të lehtë lidhjen me pajisjet video. Komponenti më i rëndësishëm në të gjitha modelet është çipi Broadcom SoC (angl. System on a Chip) me një procesor ARM dhe kartelë grafike të integruar. Interesante është se Raspberry Pi nuk ka memorie të përhershme të integruar. Zakonisht përdoret kartelë SD e jashtme për të ruajtur sistemin operativ dhe si memorie programimi. Nëpërmjet konektorit 40 pinësh (angl. GPIO-General Purpose Input Output) mund të lidhen deri në 27 pajisje të ndryshme. Raspberry Pi posedon **konektor rrjeti** dhe mundësinë e lidhjes nëpërmjet Wi-Fi, gjë që e lehtëson dukshëm qasjen në rrjetin e internetit. Disa modele përmbajnë edhe Bluetooth të integruar, si një mundësi tjetër për komunikim pa tela.



Raspberry Pi 3 Model B

Raspberry Pi 4 Model B

Figura 4.26. Pamja e jashtme e dy modeleve Raspberry Pi

Raspberry Pi mund të programohet pothuajse në të gjitha gjuhët e programimit. Gjuha programuese më e përdorur është gjuha programore Python, por mund të përdoret edhe cilado gjuhë tjetër programuese siç është C/C++, Java Script, Scratch.[7] Deri tani janë prodhuar **katër gjenerata** të Raspberry Pi dhe në çdo gjeneratë ka nga disa modele. Gjenerata e parë e vitit 2012 ka dy modele A dhe B. Modeli B ka një konfigurim më të mirë, ndërsa modeli A ka më pak memorie dhe numër më të vogël të portave, por edhe konsum më të ulët të energjisë. Raspberry Pi 2, Modeli B doli në treg në vitin 2015, ndërsa vitin tjetër u paraqit edhe gjenerata e tretë. Në vitin 2019 doli gjenerata e fundit, Raspberry Pi 4. Në tabelën 4.1, janë dhënë specifikat për secilën gjeneratë.

	Raspberry Pi 4 Modeli B	Raspberry Pi 3 Modeli B	Raspberry Pi 2 Modeli B	Raspberry Pi Modeli B
Procesor	QUAD Core me 1.5 GHz	QUAD Core me 1.2 GHz	QUAD Core co 900MHz	Single Core me 700MHz
Kartela grafike	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Memoria e përhershme	MicroSD	MicroSD	MicroSD	MicroSD
RAM	Deri në 4GB SDRAM në 400 MHz	1GB SDRAM @ 400 MHz	1GB SDRAM në 400 MHz	512MB SDRAM me 400 MHz
Frekuenca e punës	1,5GHz	1,4GHz	900MHz	700MHz
Chipset	Broadcom BCM2838 64Bit	Broadcom BCM2837 64Bit	Broadcom BCM2836 32Bit	Broadcom BCM2837 32Bit
USB 2.0	4 x USB porta	4 x USB porta	4 x USB porta	4 x USB porta
Lidhja e rrjetit	RJ45	RJ45	RJ45	RJ45
Wi-Fi	E integruar	E integruar	Nuk ka	Nuk ka
Bluetooth	E integruar	E integruar	Nuk ka	Nuk ka
Hyrje dhe dalje për qëllime të përgjithshme	Konektor 40 pinësh	Konektor 40 pinësh	Konektor 40 pinësh	Konektor 40 pinësh
Furnizimi	5 V	5 V	5 V	5 V

Tabela 4.1. Krahasimi i karakteristikave të modeleve të ndryshme të Raspberry Pi

.Nga tabela mund të konkludohet se Raspberry Pi, për sa i përket aftësive të tij, është shumë më afër një kompjuteri personal sesa Arduino.

Ne do të njihemi me mënyrën e instalimit të sistemit operativ Raspbian, me komandat dhe funksionet bazë të gjuhës programuese Python dhe do të realizojmë disa projekte që i shfrytëzojnë performancat e jashtëzakonshme të ndërfaqes së Raspberry Pi.

4.11. Pjesët përbërëse të mikrokompjuterit në pllakë Raspberry Pi

Raspberry Pi është mikrokompjuter në pllakë i mrekullueshëm me madhësinë e kartelës së kreditit dhe me një kosto shumë të ulët. Me të mund të kërkojmë në internet, të luajmë video lojëra, të shkruajmë programe kompjuterike, të krijojmë qarqe elektrike etj. Ne do të njihemi me modelin Raspberry Pi 3 B+. Raspberry Pi 4 Modeli B ka një kapacitet të memories RAM shumë më të madhe (1 GB, 2 GB ose 4 GB) dhe diodë deri në katër herë shpejtësi më të madhe se paraardhësi i tij. Ky model është ideal nëse duam të përdorim Raspberry Pi si një kompjuter personal në shtëpi ose për përpunimin e ueb imazheve (angl. Computer Vision). Për dallim nga modelet tjera, Raspberry Pi 4 Modeli B ka probleme me tejngrohjen, madje rekomandohet përdorimi i një ftohësi. Nëse destinimi i Raspberry Pi është për projekte në fushën e elektronikës ose automatizimit të shtëpive tona, atëherë rekomandohet përdorimi i një Raspberry Pi 2 ose 3 modeli B.[8] Të theksojmë se të gjitha modelet Raspberry Pi janë kompatibil, gjegjësisht softuerin e shkruar për një model mund ta ekzekutojë cilido model tjetër.

Në figurën 4.27. është paraqitur version i tretë i modelit Raspberry Pi 3 B+. Si edhe çdo kompjuter tjetër, Raspberry Pi është i përbërë nga më tepër komponentë, nga të cilët më i rëndësishëm çipi sistemor Broadcom BCM2837 SoC (angl. System on Chip).

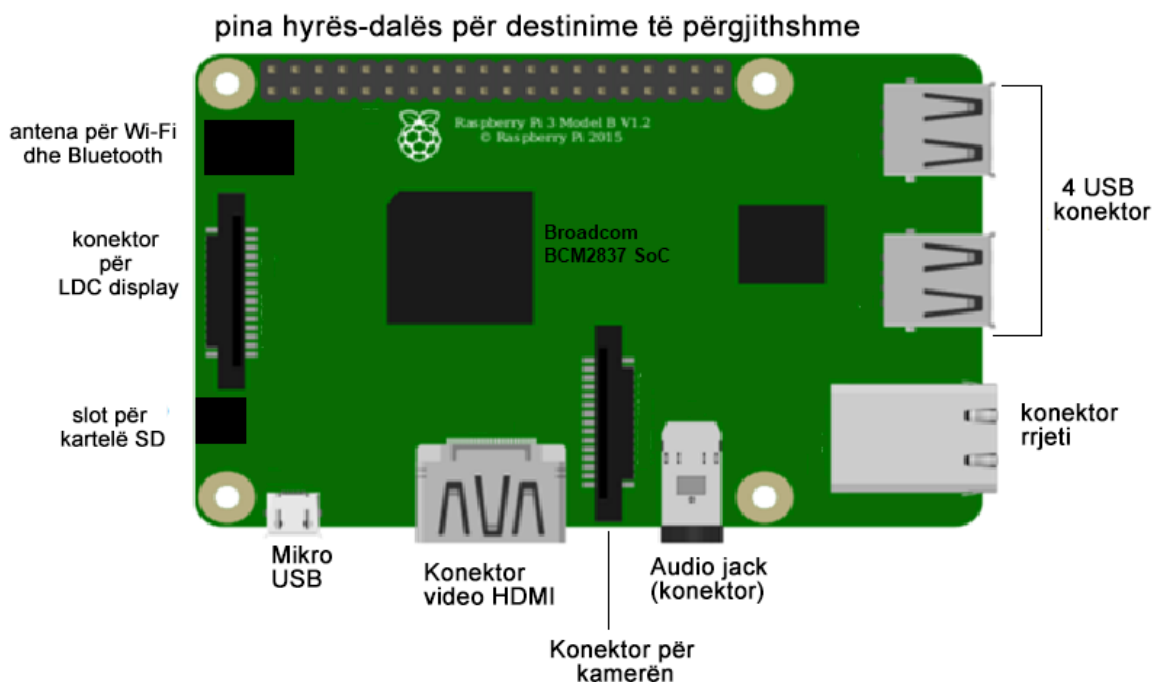


Figura 4.27 Pjesët përbërëse të Raspberry Pi 3B+ (pamja nga lart)

Ai përmban: procesor 64-bit ARM Cortex-A53 me katër bërthama me një frekuencë funksionimi prej 1.4 GHz, 32 KB cache memorie L1, 512 KB L2 cache memorie dhe një procesor grafik VideoCore IV. Memoria RAM nuk është i integruar në të njëjtin çip me procesorin, siç ishte rasti me Arduino Uno R3. Ajo është çip i veçantë në pllakën zhvillimore, me kapacitet maksimal prej 1 GB. Memorien RAM nuk mund ta shohim në figurën e mëposhtme sepse ajo ndodhet në anën e pasme. [7]

Poshtë kapakut metalik me logon e gdhendur të Raspberry Pi ndodhet radio-transmetuesi. Ai përdoret për lidhjen në rrjetet të Raspberry Pi në rrjetat kompjuterike lokale dhe në rrjetin e internetit nëpërmjet Wi-Fi ose për lidhje me pajisje të tjera inteligjente (senzorë, telefona celularë) përmes Bluetooth. Kontrolluesi i rrjetit dhe i USB është çip përgjegjës për transmetimin përmes **portës së rrjetit dhe katër konektorëve USB**. Shumë afër mini konektorit USB ndodhet një çip më i vogël i cili kujdeset për furnizimin e të gjitha komponentëve të pllakës zhvillimore.

Avantazhi më i madh i Raspberry Pi në krahasim me Arduino është mundësia e lidhjes së tij, pra numri i madh i konektorëve. Të rikujtojmë se Arduino Uno R3 kishte vetëm një mini konektor USB. Raspberry Pi ka katër konektor USB 2.0 dhe një konektor mikro USB. Në figurën 4.27. shihen vetëm dy konektor USB 2.0, por në fakt janë katër, dy prej të cilave janë të vendosur njëri mbi tjetrin. Këta konektor përdoren për lidhjen e një numri të madh të pajisjeve të jashtme: tastiera, mause kompjuterësh, kamera digjitale, hard disqe të jashtme dhe memorie tjera. Konektori mikro USB përdoret për furnizimin me energji elektrike, ngjyshëm siç furnizohen telefonat celular. Ky konektor mund të përdoret gjithashtu edhe për lidhjen me kompjuter për transferimin e të dhënave.

Përveç përmes Wi-Fi, kërkimi në internet mund të bëhet edhe nëpërmjet portës së rrjetit. Për këtë qëllim na nevojitet kablo me konektor RJ45. Fundi tjetër i kablos lidhet me ruter. Në pjesën e poshtme të portës së rrjetit janë dy dioda led që shërbejnë si indikator për komunikim të suksesshëm.

Siç na tregon edhe vetë emri **HDMI** (angl. High Definition Multimedia Interface), ky konektor ofron transmetim audio dhe video më cilësor. Nëpërmjet tij, Raspberry Pi lidhet me monitor ose televizor. Konektori 3,5 milimetërsh AV (angl. Audio Video) para së gjithash është i destinuar për lidhjen me kufje ose një audio përforcues. Ai mund të përdoret edhe si video konektor, por atëherë duhet të sigurohet një adapter i veçantë.

Në vetë pllakën zhvillimore, ekzistojnë **dy konektor të veçantë** për lidhjen e moduleve të dizajnuar special për Raspberry Pi. Njëri njihet me shkurtesën CSI (angl. Camera Serial Interface) dhe shërben për lidhjen e kamerës, ndërsa i dyti DSI (angl. Display Serial Interface) për lidhjen e ekranit sensitive (angl. touch screen).

Raspberry Pi posedon me konektor me 40 pina për t'u lidhur me pajisjet hyrëse-dalëse për qëllime të përgjithshme. Shkurtesa për këtë konektor është GPIO (angl. General Purpose Input Output). Njëzet e gjashtë pina të këtij konektori janë

për lidhje me pajisjet hyrëse-dalëse, dh etë tjerët janë për tokëzim dhe furnizim, prej 3.3 V dhe 5 V.

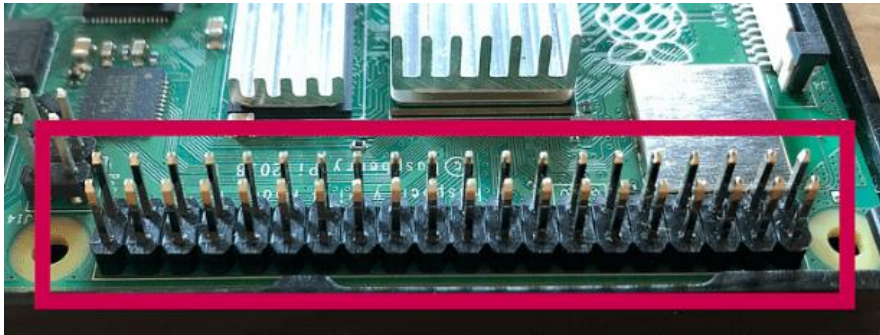


Figura 4.28. Pinat GPIO për lidhjen e pajisjeve hyrëse-dalëse

Me funksionin dhe shënimin e elementeve hyrëse-dalëse u njohëm kur folëm për pjesët përbërëse të Arduino Uno R3. Kuptohet, protoplaka është zgjidhja më e thjeshtë për lidhjen e këtyre pajisjeve.

4.12. Aksesorët për Raspberry Pi

Aksesorët për zgjerimin e funksionalitetit të Raspberry Pi njihen si harduer i ngjitur nga sipër (angl. HAT- Hardware attached on top). Zakonisht për lidhjen përdoren të gjithë 40 pinat për qëllime të përgjithshme të Raspberry Pi, por shumica e aksesorëve lejojnë që pinat t'i shfrytëzojnë edhe pajisje tjera, madje është e lejuar edhe lidhja e më shumë aksesorëve njëri mbi tjetrin. Numri më i madhi aksesorëve lidhen sipas parimit li dhe funksiono (angl. Plug and Play), gjegjësisht ato konfigurohen automatikisht. [7]

Në figurën 4.29. është paraqitur njëri nga aksesorët më të popullarizuar, aksessori me senzore dhe ekran (angl. Sense hat). Ai në vetvete përmban një ekran matricor LED 8x8, një xhoistik të vogël me pesë butona dhe gjashtë senzore të integruar: një xhiroskop (sensor për shpejtësi këndor), akselerometër

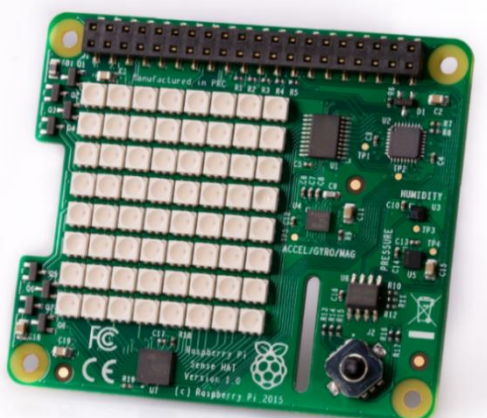


Figura 4.29. Aksesor Raspberry Pi me sensor dhe ekran

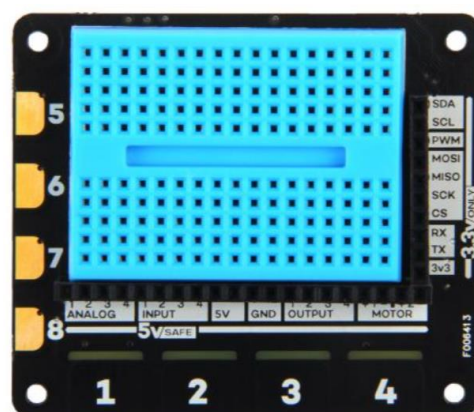


Figura 4.30. Aksesor për hulumtim Raspberry Pi

(sensori linear për nxitimin), magnetometri (duke matur fushën magnetike të tokës, magnetometri përcakton drejtimin e veriut gjeografik), barometër (sensor për shtypjen), sensor për temperaturën dhe sensor për lagështinë relative.

Aksesori për hulumtim, i paraqitur në figurën 4.30. përdoret në robotikë para së gjithash për shkak të mundësisë për kontrollin e motorëve të rrymës njëkahëshe. Ky aksesori është kompatibil edhe me platformat e mikrokontrolluese Arduino. Aksesori për hulumtim përmban katër hyrje dhe dalje me vlerë maksimale të tensionit prej 5V, katër butonakapacitiv me prekje, katër butona për lidhjen e kapëseve në formë aligatorit, katër hyrje analoge, dy ura-H për kontrollin e rrotullimit të motorëve të rrymës njëkahëshe dhe një mini protoplakë e cila vendoset nga lart.

Raspberry Pi mund të përdoret edhe si një instrument matës falë aksesorit për matjen dhe grumbullimin e të dhënave (angl. DAQ MCC 128 - Data Acquisition Measurement Computing).

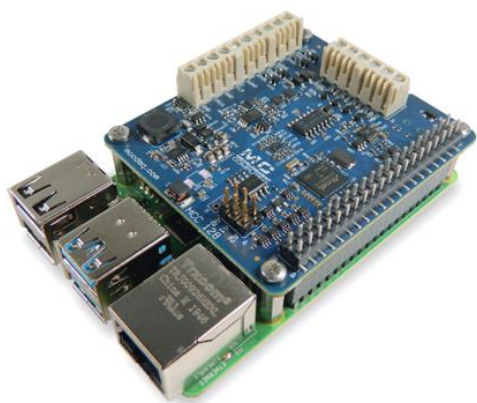


Figura 4.31. Aksesori për matjen dhe mbledhjen e të dhënave

Kjo pllakë elektronike përmban tetë hyrje analoge për matjen e tensionit. Ka dy mënyra për të matur tensionin e hyrjes. Ekzistojnë dy mënyra për matjen e tensionit të hyrjes. Në matjen me njërin skaj (angl. Single ended) matet tensioni i hyrjes në raport me pikën me potencial zero, tokëzimin.

Matja diferenciale jep ndryshimin midis dy tensioneve hyrëse. Në një sekondë merren 100 mijë. Konvertori analogo - digjital ka rezolucion prej 16 bitëve.

Mund të renditen tetë aksesore për matjen e tensionit të vendosur njëri mbi tjetrin dhe në atë rast fitohet kanal me 64 të dhëna me brez lëshimi maksimal prej 320 mijë mostra në një sekondë.

4.13. Programimi i mikrokompjuterit Raspberry Pi me gjuhën e programuese Python

Për dallim nga Arduino, Raspberry Pi është mikrokompjuter me sistem operativ. Sistemi operativ lejon instalimin dhe menaxhimin e aplikacioneve nga më të ndryshme, duke e bërë Raspberry Pi shumë të fuqishëm për sa i përket softuerit. Për Raspberry Pi ekzistojnë disa llojet e sistemeve operative, por për fillim do të përdorim **sistemin operativ Raspbian**. Në pjesën praktike të kësaj njësie modulare do të

mësojmë për instalimin e sistemit operativ Raspbian dhe desktopin e tij. Në menynë, ose më saktë në kategorinë Programming, gjendet mjedisi zhvillimor i integruar Thonny Python IDE, i cili përdoret për programimin e Raspberry Pi në gjuhën programuese Python. Në figurën 4.32. është paraqitur dritarja pas hapjes së programit zhvillimor.

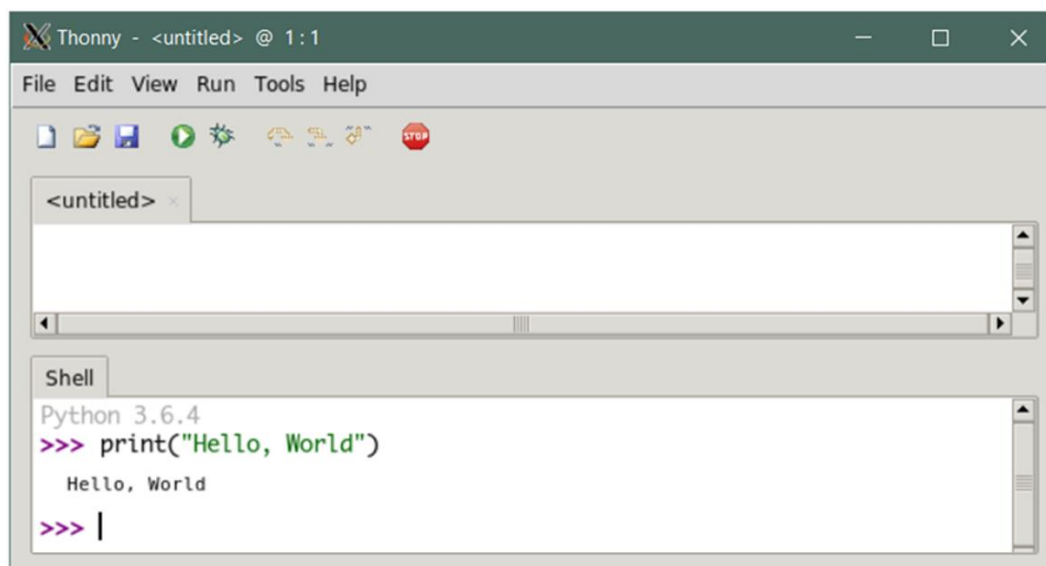


Figura 4.32. Programi zhvillimor Thonny për programimin e Raspberry Pi në gjuhën programuese Python

Ashtu si mjedisi zhvillimor Arduino, ashtu edhe programi zhvillimor Thonny Python IDE përmban editor teksti dhe një konzolë për debugim, korrigjim. Në editorin e tekstit e shkruajmë programin, ndërsa në konzolën për debugimin e shohim rezultatin e fituar me ekzekutimin e programit. Duke shtypur butonin New nisim një program të ri dhe pas krijimit e ruajmë duke shtypur butonin Save as.

Gjuha programuese Python është një nga gjuhët më të njohura të programimit, veçanërisht në mesin e gjeneratës së re. Arsyeja për këtë është **kuptueshmëria e saj e lehtë**, e cila është për shkak të sintaksës e cila është shumë e ngjashme me sintaksën e gjuhës angleze. Njohja e gjuhës angleze është parakusht për përvetësimin e shpejtë të gjuhës programuese Python. Kodi i programimit është shumë më i kapshëm. Në deklaratat nuk përdoren shkurtesa për përshkrimin e deklaratave, por fjalë të plota. Deklaratat nuk mbarojnë me pikëpresje siç ishte rasti me gjuhën programuese C/C++. Gjuha programuese Python është e një niveli më të lartë se gjuha programuese C/C++. Përkthimin e programit të shkruar në gjuhën programuese Python e realizon programi sistemor **interpretuesi**, e jo kompajleri. Kompajleri e konverton kodin e programit në gjuhën e makinës të gjithin menjëherë, ndërsa interpretuesi i përkthen deklaratat një nga një, derisa programi është duke u ekzekutuar. Kodet e programit të përkthyer me kompajler ekzekutohen më shpejt.

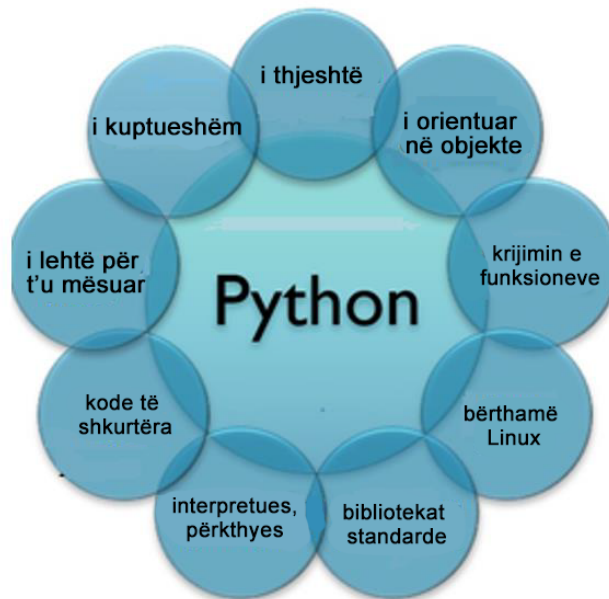


Figura 4.33. Karakteristikat e gjuhës programuese Python

Gjuha e programuese Python është më e kompatibël me sistemin operativ Linux dhe instalimi në sistemin operativ Windows është pak më i komplikuar. Të rikujtojmë, se bërthama (angl. kernel) e sistemit operativ Raspbian është Linux.

Shembulli 4.38. paraqet një **kod programi të shkurtër** për Raspberry Pi, në pinin e daljes së të cilit është lidhur një diodë.

Shembulli 4.38.

```

1 from gpiozero import LED
2 from time import sleep
3 led = LED(17)
4 while True:
5     led.on()
6     sleep(1)
7     led.off()
8     sleep(1)

```

Kodet programore në gjuhën programuese Python janë më të shkurtër sepse **nuk ekziston deklarimi i variablave**. Në gjuhën programuese C/C++, për variablin x shkruajmë `int x=5;`, ndërsa në gjuhën programuese Python vetëm `x=5`. Gjithashtu, nuk ka kufizim në llojin e të dhënave, d.m.th. nëse pjestojmë dy numra të plotë, mund të marrim menjëherë një rezultat njëzërisht, jo vetëm numër të plotë.

Kur shkruajmë një bllok instruksionesh në kuadër të një strukture, siç janë `if...else` ose `while`, nuk përdoren kllapat e mëdha, por instruksionet tërhiqen për katër boshe përpara. Pasi të deklarojmë strukturën, programuesi duhet të shtypë

dy pika (:) dhe editorit i tekstit vetë e tërheq tekstit. Komentet në kodin e programit nuk ndahen me dy prerje, por me shenjë të thurjes "#".

Në gjuhën programuese Python, **programuesi mund të krijojë funksionin e tij** duke përdorur instruksionin `def` dhe më pas të njëjtën ta thirr në vende të ndryshme në program. I tillë është kodi programor i paraqitur në shembullin 4.39.

Shembulli 4.39.

```
1 def my_function(fname, lname):
2     print(fname + " " + lname)
3 my_function("Bill", "Gates")
```

Gjuha programuese C/C++ është më e mirë për programimin e sistemeve të integruara, ndërsa gjuha programuese Python është më universale dhe me të mund të krijojmë edhe aplikacione grafike. Gjuha C/C++ është e orientuar drejt harduerit, ndërsa Python është e orientuar drejt softuerit. Gjuha programuese Python posedon me një numër të madh bibliotekash për qëllime nga më të ndryshmet. Çdo bibliotekë ka një grup të madh instruksionesh. Biblioteka `turtle` përdoret për dizajn grafik (për shembull, të vizatojmë një yll që rrotullohet). Biblioteka `pygame` përdoret për të krijuar video lojëra me efekte vizuale dhe zanore. `OpenCV` është bibliotekë për përpunimin e imazheve. `Cloud4rpi` është bibliotekë për ruajtjen e të dhënave në teknologjinë cloud. Mjafton të theksojmë se në momentin ekzistojnë 137,000 biblioteka për gjuhën programuese Python. Nuk është e mjaftueshme vetëm të zgjidhet biblioteka e duhur, por është e nevojshme të mësohen edhe komandat në kuadër të saj. Ne do të mësojmë mundësitë që i ofron biblioteka `GPIO Zero`, e cila përdoret kur `Raspberry Pi` lidhet me pajisjet hyrëse-dalëse (p.sh. dioda, tasterë, sensorë, motorë, ekrane).

4.14. Lidhja e mikrokompjuterit Raspberry Pi me pajisje hyrëse-dalëse

Për lidhjen e mikrokompjuterit `Raspberry Pi` me pajisje hyrëse dalëse përdoret **konektori 40 pinësh**, i njohur me shkurtesën `GPIO` (angl. General Purpose Input Output). **26 pina të këtij konektori janë për lidhjen me pajisjet hyrëse dalëse, ndërsa pjesa tjetër janë për tokëzim dhe furnizim me energji 3.3V dhe 5V.**

Që të mundet mikrokompjuteri `Raspberry Pi` të komunikojë me pajisjet hyrëse-dalëse të lidhura në konektorin `GPIO`, është e nevojshme të kyçet biblioteka e quajtur `GPIO Zero`. Kjo është bibliotekë standarde për sistemin operativ `Raspbian` dhe nuk ka nevojë për instalime shtesë. [9]

Numrat e pinave sipas kësaj biblioteke janë të ndryshëm nga numrat fizik të pinave. Në figurën 4.34. është treguar numërimi i të ashtuquajturave

pina GPIO. Për shembull, pinit me numër fizik 15 i korrespondon pini GPIO me numrin 22.



Figura 4.34. Shënimi dhe përshkrimi funksional i pinave GPIO

Ngjashëm si programimi i Arduino Uno R3 në C/C++, ashtu edhe gjatë programimit të Raspberry Pi 3B+ në Python, në fillim të programit duhet të kyçen bibliotekat që do të përdoren në të. Për këtë qëllim përdoret komanda import, e cila ka funksion të njëjtë si komanda include gjatë programimit të platformës zhvillimore Arduino Uno. Në shembullin 4.40, gjatë konfigurimit të pinit për lidhjen e tasterit, është shtuar prefiksi gpiozero.

Shembulli 4.40.

```
1 import gpiozero
2 button = gpiozero.Button(2)
```

Në gjuhën programuese Python, lejohet që në vend të gjithë bibliotekës, të kyçet vetëm një klasë e saj. Në shembullin 4.41, kyçet vetëm pjesa e bibliotekës e cila i përmban komandat për menaxhimin me tasterin. Në këtë rast mund të përdorim drejtpërdrejtë tasterë në program, pa prefiksin gpiozero.

Shembulli 4.41.

```
1 from gpiozero import Button
2 button = Button(2)
```

Ekziston mundësia që të punojmë me tasterë individual të punojmë me tastierë të tërë. Në atë rast duhet të thirret klasa nga biblioteka gpiozero e njohur me emrin ButtonBoard, që në përkthim do të thotë pllaka me tasterë.

Shembulli 4.42.

```
1 from gpiozero import ButtonBoard
```

Tashmë theksuam se **biblioteka GPIO është e ndarë në klasa në varësi të llojit të pajisjeve hyrëse-dalëse**. Më poshtë janë paraqitur klasat dhe pajisjet në të cilat aplikohen.

Pajisje hyrëse	→	Tsaser, senzori infra i kuq, senzori për lëvizje, senzori optik, senzori për distancë
Pajisje dalëse	→	Diodë led (RGB ose e kontrolluar me impulse të moduluar në gjerësi), sirena, motorë, servo motorë
Pajisje periferike serike pajisje	→	Konvertues analogo-digjital
Platforma dhe aksesorë	→	Ekrane led, tastiera, robotë, kontrollorë për përdorim shtëpiak përmes telekomandës
Komponentë të brendshme	→	Akordimi i kohës, temperaturës së procesorit, llogaritja e vlerave mesatare, amortizimi i diskut etj
Gjeneratorë të toneve	→	Gjenerimi i toneve muzikore, të thjeshta ose komplekse

4.15. Klasa të pajisjeve hyrëse nga biblioteka GPIOZERO

Sinjalet digjitale kanë vetëm dy vlera, një ose zero, LOW ose HIGH. Sinjalet analoge mund të marrin një vlera nga më të ndryshmet që variojnë nga 0 deri në tensionin maksimal të lejuar, duke përfshirë edhe numrat decimal. Raspberry Pi nuk ka të integruar konvertor analogo – digjital si platforma Arduino. Nëse duam të lexojmë vlerat analoge nga hyrjet e Raspberry Pi, atëherë duhet të lidhim një konvertues të jashtëm analogo-digjital, siç është për shembull qarku i integruar MCP3008. Për punë me pajisje hyrëse analoge, mund të përdoret edhe i ashtuquajtimi tension i pragut (angl. threshold). Nëse tensioni i hyrjes është më i madh se tensioni i pragut, atëherë ajo vlerë do të konsiderohet njësh logjik (LARTË), ndërsa nëse tensioni i hyrjes është më i vogël se tensioni i pragut, atëherë kjo vlerë do të konsiderohet zero logjike (LOW). Në figurën 4.35. në mënyrë krahasuese janë paraqitur tensionet e pragut për Arduino Uno dhe Raspberry Pi. Tensioni në pinat hyrës të Raspberry Pi nuk duhet të jetë më i madh se 3.3V. Në të njëjtën figurë mund të shohim se tensioni i pragut dallohet në varësi të faktit nëse tensioni i hyrjes zvogëlohet apo rritet. Kur tensioni i hyrjes rritet nga zero në vlerën maksimale të lejuar, tensioni i pragut është 1.34V për Raspberry Pi në krahasim me 2.47V për Arduino Uno. Kur tensioni i hyrjes zvogëlohet nga vlera maksimale e lejuar drejt

zeros atëherë tensioni i pragut është 1.17V për Raspberry Pi n[krahasim me 2.22V për Arduino Uno.

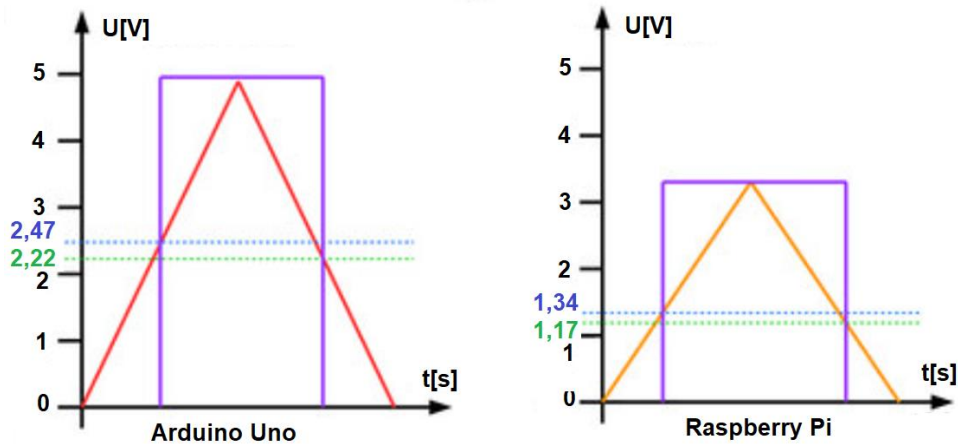


Figura 4.35. Krahasimi i tensioneve të pragut për Arduino Uno dhe Raspberry Pi

Në figurën 4.36. Tregohet është treguar ndarja e pajisjeve hyrëse sipas llojit të sinjaleve që i gjenerojnë.

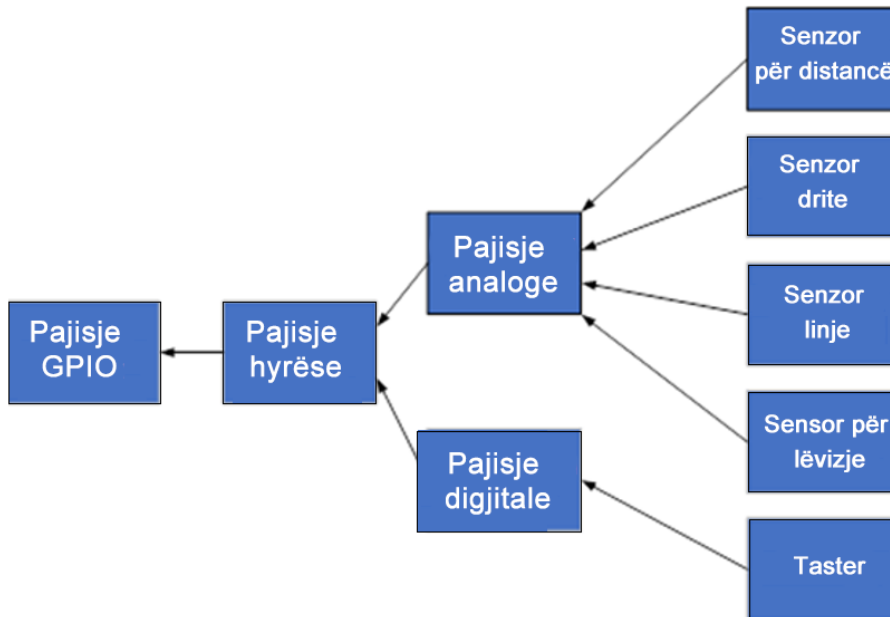


Figura 4.36. Pajisje hyrëse për Raspberry Pi

Të gjitha klasat në përbërje të bibliotekës GPIOZERO paraqesin një degëzim të tillë. Kjo qasje në masë të madhe e thjeshton programimin e Raspberry Pi.

4.15.1. Tasteri (angl Button)

Të rikujtojmë se konektorët e tasterit që shtrihen në të njëjtën linjë, por në anët e kundërta, janë të lidhur me qark të shkurtër. Kjo është paraqitur në figurën 4.37.

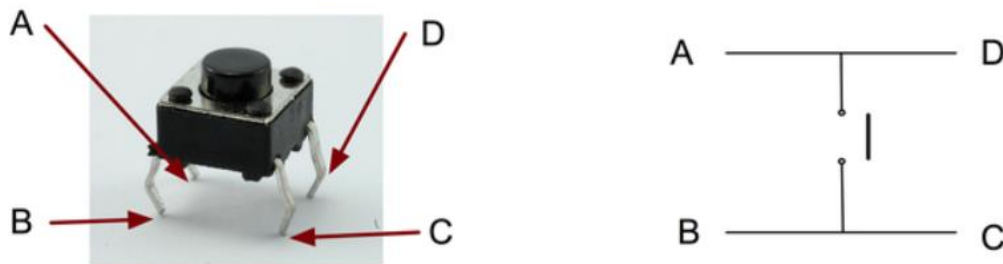


Figura 4.37. Lidhja e brendshme e konektorëve të tasterit

Shembulli 4.43. paraqet kodin programor për paraqitjen e një linje teksti me kusht që tasteri të jetë i shtypur.

Shembulli 4.43.

```

1 from gpiozero import Button
2 button = Button(4)
3 button.wait_for_press()
4 print("The button was pressed!")

```

Në këtë kod programor, janë përdorur dy instruksione për punë me taster: `Button()` dhe `wait_for_press()`. Ekzistojnë edhe 12 instruksione të tjera për punë me taster. Do të shpjegojmë instruksionet më të rëndësishme dhe parametrat e tyre, të cilat janë shkruhen në kllapa të vogla.

- Button()** → Në shembullin 4.43, numri 4 përfaqëson numrin e **pinit GPIO**. Ky parametër është i domosdoshëm. Përndryshe, konzola për debugim do të tregojë gabim. Do të përmendim edhe parametrin `pull_up`. Ky parametër mund të ketë dy gjendje, `true` ose `false`. Vlera standarde `true` do të thotë nivel i lartë (HIGH) kur shtypet tasti. Në këtë rast, njëri pin i tasterit është i lidhur me tokëzimin, ndërsa tjetri në pinin GPIO. Nëse parametri `pull_up` është `false`, atëherë fitohet nivel i ulët (LOW) kur tasteri është i shtypur dhe pini i parë në që të lidhet me masën lidhet me furnizimin prej 3,3V.
- wait_for_press ()** → Ekzekutimi i **kodit programor ndërpritet** deri sa të kalojë koha e caktuar ose tasteri nuk shtypet.
- wait_for_release()** → Ekzekutimi i kodit programor ndërpritet deri sa të kalojë koha e caktuar ose tasteri nuk lirohet.

held_time	→	Nëse tasterin e mbajmë të shtypur një kohë të gjatë, ky instruksion e tregon kohën e shtypjes të shprehur në sekonda.
hold_time	→	Ky instruksion përcakton se sa sekonda duhet pritur pas shtypjes së tasterit.
is_held	→	Pas kalimit të kohës të përcaktuar me instruksionin hold_time vlera bëhet e vërtetë (true).
is_pressed	→	Rezultati i këtij instruksioni është vlerë boolean, true ose false. Vlera është e vërtetë (true) vetëm kur tasteri është i shtypur.

4.15.2. Moduli reflektues infrared për monitorim (TRCT5000)

Moduli reflektues për monitorim infra i kuq është paraqitur në figurën 4.38. Ai mund të përdoret për detektimin e linjës së plotë. Moduli ka katër konektor, por për lidhjen me Raspberry Pi përdorim tre: Vcc, GND dhe DO (angl. Data Out). Pini Vcc lidhet me furnizimin prej 3,3V, GND është tokëzimi. Pini DO është pini për sinjalizim dhe lidhet me njërin nga pinat GPIO.

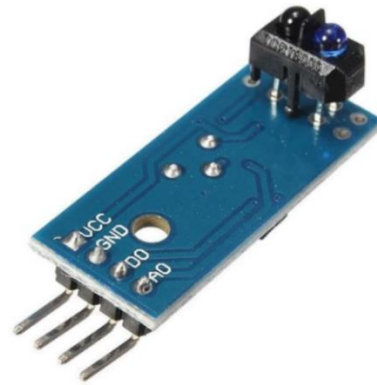


Figura 4.38. Moduli reflektues infra i kuq për monitorim

Shembull 4.44. paraqet kodin programor për paraqitjen e tekstit, në varësi të vlerës që do të lexojë sensorin.

Shembulli 4.44.

```

1 from gpiozero import LineSensor
2 from signal import pause
3 sensor = LineSensor(4)
4 sensor.when_line = lambda: print('Line detected')
5 sensor.when_no_line = lambda: print('No line detected')
6 pause()

```

LineSensor(4) → Me këtë instruksion **konfigurohet pini** për lidhjen me sensorin

when_line → Instruksioni për ekzekutim kur sensorin është aktiv.

`when_no_line` —→ Instruksioni ekzekutohet kur sensori nuk është aktiv

`lambda` —→ Ky është instruksion standard i Python i cili përdoret si nëninstruksion i një instruksioni tjetër, për të treguar ndonjë veprime. Nuk përdoret si instruksion i pavarur.

4.15.3. Senzori i distancës (HC-SR04)

Senzori i distancës është sensor me ultratingull që dërgon valë të drejtuar, e cila reflektohet nga objekti i vendosur përpara sensorit dhe kthehet në sensor. Matet koha nga momenti kur transmetuesi dërgon valën deri në momentin kur marrësi merr valën e reflektuar. Sipas këtij parametri, llogaritet distanca midis sensorit dhe objektit. **Vala e transmetuar quhet triger (aktivator), ndërsa vala e reflektuar quhet jehonë.**



Figura 4.39. Senzori i distancës (HC SR04)

Shembull 4.45. paraqet një kod programor për HC SR04 i cili jep informacion për distancën e shprehur në cm. Pini për trigerim është i lidhur me pinin GPIO17.

Shembulli 4.45.

```

1 from gpiozero import DistanceSensor
2 from time import sleep
3 sensor = DistanceSensor(echo=18, trigger=17)
4 while True:
5     print('Distance: ', sensor.distance / 100)
6     sleep(1)
    
```

DistanceSensor() —→ **Të detyrueshëm janë dy parametra.** I pari është numri i pinit GPIO për jehonën, pastaj për trigerin. Ato shkruhen në kllapa të vogla.

wait_for_in_range —→ Ekzekutimi i kodit programor ndërpritet derisa distanca të jetë mbi distancën e pragut (angl. `threshold_distance`) ose të ketë kaluar koha e specifikuar në kllapat e vogla. Distanca e pragut zakonisht është 0.3 m dhe si parametër mund të akordohet me instruksionin `DistanceSensor()`.

- wait_for_out_of_range** → Ekzekutimi i kodit programor ndërpritet derisa distanca të jetë nën distancën e pragut ose të kalojë koha e specifikuar.
- distance** → Rezultati i këtij instruksioni është distanca e matur nga sensori dhe kjo distancë është dhënë në metra.
- when_in_range** → Pas këtij instruksioni, pasohet shenja e barazimit dhe një instruksion tjetër ndihmës që ekzekutohet nëse plotësohet kushti.
- when_out_of_range** → Instruksioni përcakton një veprim nëse distanca është mbi distancën e pragut.

4.15.4. Senzori optik ose fotorezistori (анг. LDR-Light Depended Resistor)

Fotorezistori është sensori më i përdorur për zbulimin e dritës. Rezistenca e tij ndryshon nga 1KΩ në dritë dhe 100KΩ në errësi, pra e njëjta zvogëlohet me rritjen e intensitetit të dritës. Më vonë do të kryejmë një ushtrim praktik me zbatimin e tij, ndërsa tani do të njihemi me disa instruksione për të punuar me këtë sensor.

Shembulli 4.46.

```

1 from gpiozero import LightSensor
2 ldr = LightSensor(18)
3 ldr.wait_for_light()
4 print("Light detected!")

```

- LightSensor()** → Në kllapën e vogël tregohet numrin e pinit GPIO. Në shembullin 4.46, ldr është emër simbolik i sensorit dhe emrin e jep programuesi.
- wait_for_light()** → Me këtë instruksion, kodi i programit ndalet derisa sensori të aktivizohet. Koha që duhet të kalojë përpara se sensori të aktivizohet mund të specifikohet gjithashtu në kllapa të vogla.
- wait_for_dark()** → Me këtë instruksion ndalohet kodi i programit derisa të shfaqet errësira.

4.16. Klasa të pajisjeve dalëse nga biblioteka GPIO Zero

Në figurën 4.40. është treguar ndarja e pajisjeve dalëse sipas llojit të sinjaleve që i gjenerojnë.

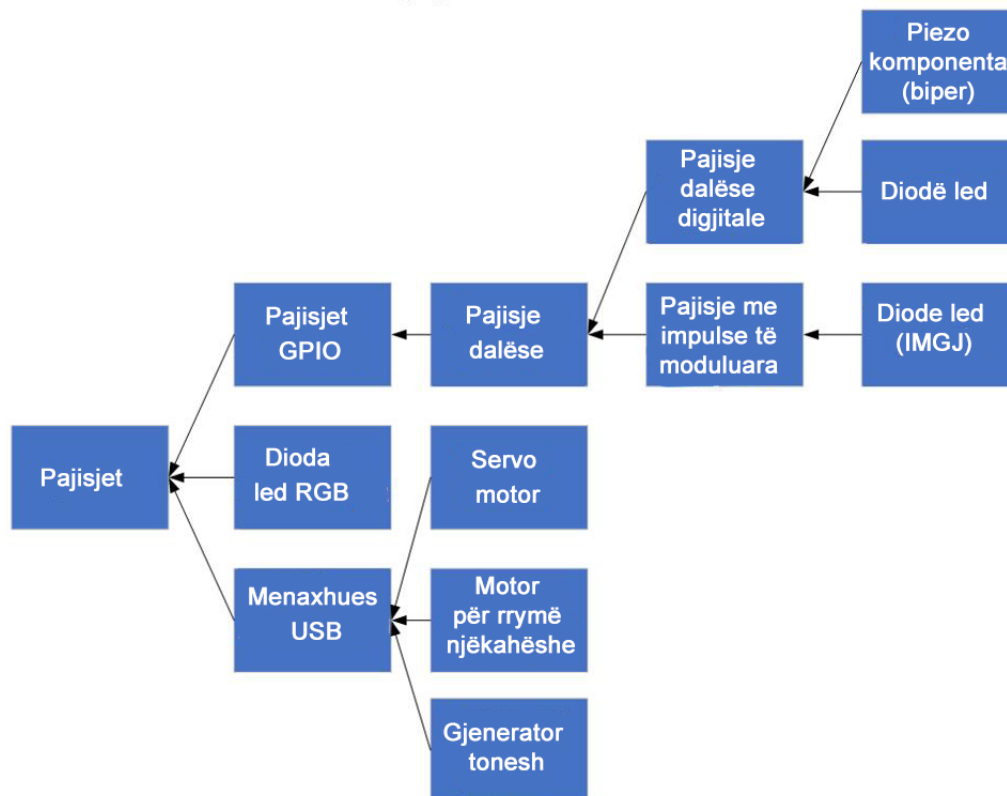


Figura 4.40. Pajisjet dalëse për Raspberry Pi

Pema e pajisjes dalëse është e degëzuar edhe më tej. Sipas llojit të konektorit, është bërë edhe një ndarje tjetër, pajisje dalëse me konektor GPIO dhe pajisje me konektor USB. Pajisjet me konektor GPIO ndahen në pajisje dalëse digjitale dhe pajisje të komanduara me impulse të moduluara në gjerësi (PWM).

Për të kyçjen dhe shkyçjen e pajisjeve digjitale, janë të nevojshme dy vlera, LOW ose HIGH. Me impulse të moduluara në gjerësi kryejmë ndryshimin e intensitetit të dritës të diodës led. Ngjashëm si te pajisjet hyrëse, ashtu edhe te pajisjet dalëse, biblioteka gpiozero përmban një grup të veçantë të instruksioneve për secilën pajisje.

4.16.1. Dioda LED

Gjatë lidhjes së diodës LED me Raspberry Pi, duhet të kujdesemi që **pini më i gjatë (anoda) të lidhet me pinin GPIO, ndërsa pini më i shkurt (katoda)**

me tokëzimin. Gjithashtu duhet të përdoret një rezistencë për të kufizuar rrymën.

Kodi programor në shembullin 4.47. kryen kyçjen e diodës LED..

Shembulli 4.47.

```
1 from gpiozero import LED
2 led = LED(17)
3 led.on()
```

LED() —→ Në kllapat e vogla theksohet numri i pinit GPIO, vlera aktive dhe fillestare. Nëse vlera aktive është HIGH, atëherë dioda do të ndriçoj si në shembullin e mësipërm. Nëse `active_high=false`, atëherë që të ndriçoj dioda led na duhet instruksioni `off()`. `led` është emri simbolik për diodën që e vendos programuesi.

blink (on_time=1, off_time=1) —→ Me këtë instruksion, dioda fillon të pulsojë. Mund të ndryshojmë **kohën e kyçjes dhe shkyçjes. Sitandard, kjo kohë është sekondë.**

off() —→ Ky insrtuksion e shkyç diodën led

on() —→ Ky instruksion kyç diodën led

toggle() —→ Ky instruksion **ndryshon gjendjen e diodës led**, LED-it, nëse ka qenë e kyçur e shkyç dhe anasjelltas.

4.16.2. Dioda led e kontrolluar me impulse të moduluara në gjerësi (PWMLLED)

Nëse duam të ndryshojmë intensitetin e dritës, në program duhet të përfshihet klasa `PWMLLED` nga biblioteka `gpiozero`. Drita është më e madh kur impulset nga vargu periodik kanë gjerësi më të madhe. Gjerësinë e impulsive mund ta ndryshojmë me ndihmën e potenciometrit.

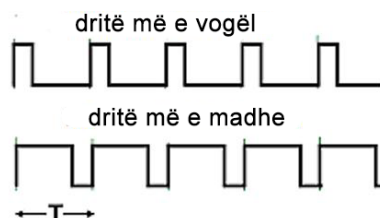


Figura 4.41. Varësia e intensitetit të dritës nga gjerësia e impulsive

Në instruksionin `PWMLLED()`, në kllapa të vogla theksohet numri i pinit GPIO për lidhje, frekuenca e vargut periodik dhe vlera fillestare. Nëse nuk specifikohet

frekuenca, atëherë do të vlej frekuenca standarde prej 100 Hz. Për të menaxhuar me intensitetin e dritës shfrytëzohen vlera vlerat nga zero deri në një.

Shembulli 4.48.

```

1 from gpiozero import PWMLED
2 from time import sleep
3 led = PWMLED(17)
4 while True:
5     led.value = 0
6     sleep(1)
7     led.value = 0.5
8     sleep(1)
9     led.value = 1
10    sleep(1)

```

4.16.3. Dioda led me ngjyrë të ndryshueshme (RGB-лед)

Shkurtesa RGB vjen nga fjalët red, green, blue, që në përkthim do të thotë e kuqe, e gjelbër, e kaltër. Me përzjerjen e këtyre tri ngjyrave kryesore fitohen të gjitha të tjerat. Ekzistojnë 256 nuanca prej secilës ngjyrë primare dhe duke përzjer këto nuanca, mund të përftohen 16,777,216 ngjyra të ndryshme. Secila ngjyrë primare paraqet një variabël vlera e së cilës është numër dhjetor me vlerë nga 0 deri në 1. **Ngjyrat tjera fitohen si kombinim i atyre primare.** Për shembull, kombinimi (1, 0, 0) përfaqëson ngjyrën e kuqe, kombinimi (0, 1, 0) përfaqëson ngjyrën e gjelbër, kombinimi (1, 1, 0) do të jetë ngjyra e verdhë dhe kombinimi (1, 0.5, 0) do të jetë portokalli. Për përdorim më të lehtë të ngjyrave, mund të përdoret biblioteka color, e cila është bibliotekë standarde në gjuhën programuese Python.

Dioda led RGB ka katër konektor, nga një konektor për secilën ngjyrë dhe një konektor të përbashkët që është më i gjatë. Në rastin e diodës led RGB me katodë të përbashkët, ajo lidhet me tokëzimin, ndërsa anodat me tre pinat e GPIO, nga një për secilën nga tri ngjyrat kryesore. Me kodin e programit në shembullin 4.49. dioda led do të ndriçoj në të verdhë. Shembulli 4.50. është kod programor për përdorimin e bibliotekës color.

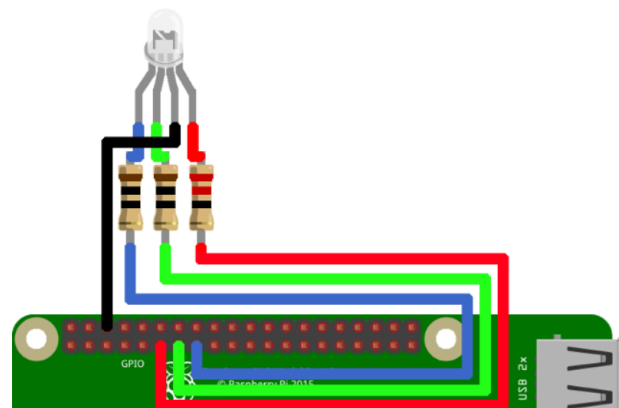


Figura 4.42. Lidhja e diodës RGB me Raspberry Pi

Shembulli 4.49.

```

1 from gpiozero import RGBLED
2 led = RGBLED(2, 3, 4)
3 led.color = (1, 1, 0)

```

Shembulli 4.50.

```

1 from gpiozero import RGBLED
2 from colorzero import Color
3 led = RGBLED(2, 3, 4)
4 led.color = Color('yellow')

```

Për diodën RGB vlej në të njëjtat instruksione si edhe për diodën led normale, me atë që duhet të konfigurohen të tre pinat GPIO dhe të definohet ngjyra e diodës led me instruksionin `color()`. Ndryçimi i diodës RGB mund të ndryshohet sipas intensitetit nëse në instruksionin `RGBLED()` kyçen impulse me modulim në gjerësi duke përdorur parametrin `pwm=True`.

4.16.4. Motori i rrymës njëkahëshe

Nëse drejtimi i rrymës ndryshon, do të ndryshoj edhe drejtimi i rrotullimit të motorit njëkahësh. Ai lidhet me **Raspberry Pi nëpërmjet një ure H**. Parimi i funksionimit të urës H është paraqitur në figurën 4.43.

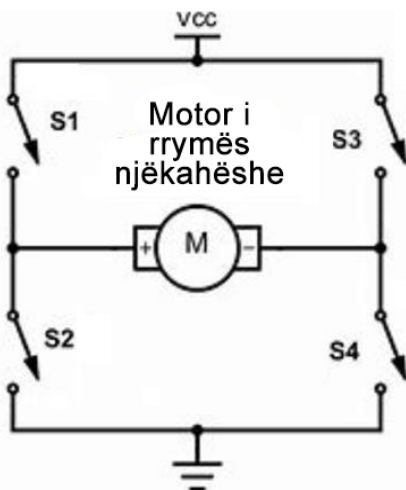


Figura 4.43. Parimi i punës së urës H

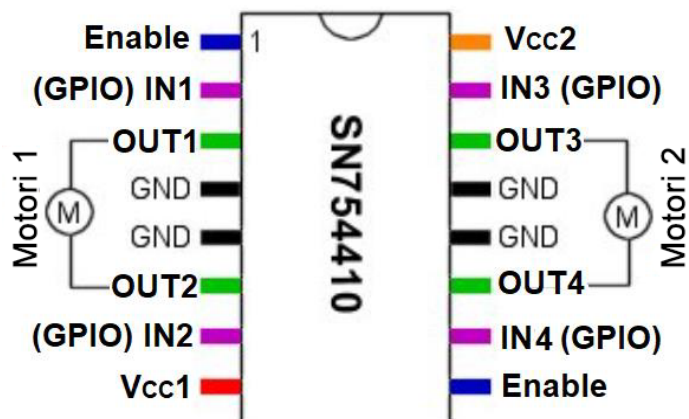


Figura 4.44. Diagrami i pinave i urës H

Ai është drejtues dhe mund të ndryshojë polaritetin e tensionit. Nëse ndërprerësit S1 dhe S4 janë të mbyllur dhe ndërprerësit S2 dhe S3 janë të hapur, atëherë motori rrotullohet në drejtim të akrepave të orës. Nëse S1 dhe S4 janë të hapura dhe ndërprerësit S2 dhe S3 janë të mbyllur, atëherë motori rrotullohet në drejtim të kundërt. Që Raspberry Pi të funksionojë si një urë H, më shpesh përdoren qarqet e integruara L293D ose SN754410. Në figurën 4.44. është treguar diagrami i pinave i qarkut të integruar SN754410 dhe i njëjti mund të përdoret për të menaxhuar dy motorë të rrymës së vazhduar. Nëse me motorët janë të lidhur rrota, atëherë ky element mund të përdoret për të lëvizur robotin. Daljet e urës H lidhen me motorin e rrymës njëkahëshe, ndërsa hyrjet me dy pina të GPIO.

Në ushtrimin praktik 4.8.2.7. ndryshimi i drejtimimit të rrotullimit të motorit të rrymës njëkahëshe është përdorur qarku i integruar SN754410 dhe në të njëjtin është shpjeguar mënyra e lidhjes së motorit, urës H, Raspberry Pi dhe protoplakës.

Më poshtë janë dhënë disa instruksione për punë me motor-DC.

- Motor()** → Në kllapa të vogla shkruhen numrat e pinave GPIO për lidhjen e urës H..
- backward(speed=1)** → Motori i rrymës së vazhduar rrotullohet në drejtim të kundërt me drejtimin e akrepave të orës. **Vlera e shpejtësisë së rrotullimit (speed) mund të ndryshohet në intervalin nga 0 deri në 1**, me kusht që të përdoren sinjale të moduluara në gjerësi. Njëshi është vlera maksimale e shpejtësisë së rrotullimit.
- forward(speed=1)** → Motori DC rrotullohet në drejtim të akrepave të orës.
- reverse()** → Ndërrohet drejtimi i rrotullimit të motorit.
- stop()** → Motori ndalon së rrotulluari.

Kodi programor në shembullin 4.51. e rrotullon motorin përpara, në drejtim të akrepave të orës.

Shembulli 4.51.:

```
1 from gpiozero import Motor
2 motor = Motor(17, 18)
3 motor.forward()
```

4.16.5. Servo motori

Le të rikujtojmë, se servo motori nuk rrotullohet në rreth, por ai zhvendoset për një kënd të caktuar, majtas ose djathtas nga pozicioni i tij qëndror, në varësi të madhësisë së sinjalit komandues. Mënyra e funksionimit të servo motorit është paraqitur në figurën 4.24.

Në figurën 4.45. është paraqitur pamja e jashtme e servo motorit, ndërsa në figurën 4.46. është paraqitur mënyra e lidhjes së tij me pinat GPIO të Raspberry Pi. Servo motori ka tre konektorë: i kuq, i zi dhe portokalli. Kablloja e kuqe lidhet me furnizimin prej 5 volt, ndërsa kablloja e zezë lidhet me tokëzimin. Kablo portokalli lidhet me njërin nga pinat GPIO dhe nëpërmjet saj Raspberry Pi komunikon me motorin, pra e kontrollon këndin e rrotullimit.



Figura 4.45. Pamja e jashtme e servo motorit

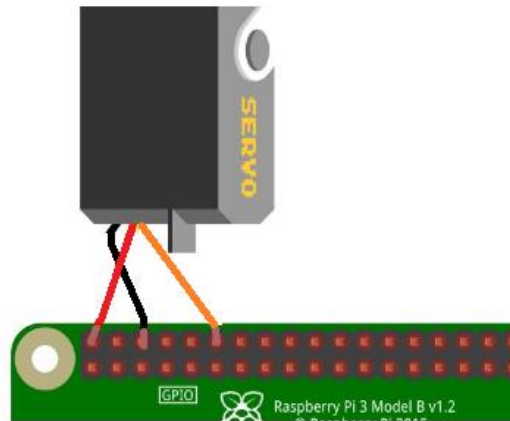


Figura 4.46. Lidhja e servo motori me Raspberry Pi

Me kodin programor në shembullin 4.52, servo motori do të rrotullohet për 45 shkallë.

Shembulli 4.52.

```
1 from gpiozero import Servo
2 servo = Servo(17)
3 servo.value = 0.5
```

Do të fokusohemi vetëm në instruksionin `servo.value()`, me të cilin definohet vlera e këndit të rrotullimit. Kjo vlerë mund të ndryshohet në intervalin midis -1 (këndi minimal prej 0 shkallë) dhe 1 (këndi maksimal prej 180 shkallë). Nëse vlera është zero, atëherë servo motori është në pozicionin e mesëm, gjegjësisht 90 shkallë.

Përfundime:

Mikrokompjuter në pllakë është tabelë elektronike në të cilën janë integruar: një mikrokontrollues, një programues serik, një oshilator kristalor, porta, hyrje dhe dalje analoge dhe digjitale. Ai përmes një kabloja USB mund të lidhet me kompjuter dhe të futet një program i gatshëm në memorien e tij programuese.

Në hyrjen e mikrokompjuterit në pllakë mund të lidhen: tasterë, ndërprerës, tastiera të gatshme, sensorët (për temperaturën, presionin, rrjedhën, lëvizjen etj.). Daljet mund të lidhen me një sërë pajisjesh dalje, si: dioda led, llamba, sinjalizues, motorë, ekrane.

Ekzistojnë më tepër se 15 modele të ndryshme të platformave Arduino, ndërsa momentalisht më të popullarizuara janë Arduino Uno, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due.

Mikrokompjuteri Arduino Uno posedon 14 pina digjitale, gjashtë hyrje analoge, tre indikatorë me dioda led dhe një LED të integruar.

Për mbrojtje personale dhe për të mbrojtur elementet nga dëmtimi, para fillimit të secilit montim duhet të shkyçet burimi i furnizimi, pavarësisht nëse bëhet fjalë për bateri apo kompjuter.

Programi zhvillimor për mikrokompjuterin Arduino Uno shkarkohet nga ueb faqja zyrtare e Arduino, më saktë nga linku i më poshtëm arduino.org/download. Pas shkarkimit të instalimit, të njëjtën e instalojmë. Shtypim (click) në Run. Nëse nuk fillon instalimi, është e nevojshme të hapet programi Control Panel, të selektohet kategoria Device Manager, pastaj nënkategoria Other Devices ose Unknown Device dhe të shtypet (click) Update Drivers ose Update Driver Software.

Elemente themelore përbërëse të gjuhës programuese C/C++ janë: variablat, instruksionet dhe strukturat.

Sintaksa në programim nënkupton një grup rregullash për rregullimin e shenjave, emrave simbolikë, operatorëve, shenjave të pikësimit dhe komenteve në mënyrë që të fitohet një deklaratë e kuptueshme për kompjuterin.

Variablat i rruajnë të dhënat që përpunohen në kompjuter. Përveç emrit dhe llojit të të dhënave, variablat kanë edhe vlerat e tyre. Kur deklarohen (paraqesim) variabla për mikrokompjuterin Arduino Uno, fillimisht shkruhet lloji i variablës, më pas emri simbolik, operatori për përcaktim dhe vlera e variablës.

Llojet bazë të të dhënave janë: numrat e plotë, numrat decimal, karakteret, vargjet dhe të dhënat logjike.

Operatorët janë shenja me të cilët programuesi, sipas rregullave të përcaktuara saktësisht, ndërton deklaratat, instruksione. Ekzistojnë disa lloje të operatorëve: operatorë matematikorë, logjikë, krahasues dhe për caktim.

Instruksionet e pinat për hyrje-dalje shërbejnë për konfigurimin e pinave (hyrje ose dalje) dhe për të shkruar ose lexuar vlerat e tyre. Në këtë grup përfshihen instruksionet: `digitalRead(pin)`, `digitalWrite(pin, vlerë)`, `pinMode(pin,mode)`, `analogRead(pin)` dhe `analogWrite(pin,vlera)`.

Konvertuesi analogo-digjital, në përbërje të mikrokompjuterin Arduino Uno, i konverton vlerat analoge në numra të plotë nga 0 deri në 1023. Numri i plotë, më pas paraqitet si kod binar prej 10 bitëve.

Instrukcioni `delay (ms)` përdoret për të futur një vonesë në kohë të shprehur në milisekonda.

Instrukcioni `map (vlera, fromLow, fromHigh, toLow, toHigh)` ndryshon brezin e vlerave të variablës. `fromLow` dhe `fromHigh` janë vlerat minimale dhe maksimale të brezit të vjetër, ndërsa `toLow` dhe `toHigh` janë vlerat minimale dhe maksimale të brezit të ri.

Instrukcioni `bitClear(x, n)` e rivendos (e vendos në zero) bitin me numër rendor `n` në variablën `x`. Instrukcioni `bitSet(x,n)` e vendos (e vendos në nivel të lartë) bitin me numër rendor `n` në variablën `x`.

Me instruksionin `Serial.begin(speed)` përcaktohet brezi i lëshimit, gjegjësisht shpejtësia e transferimit serik të të dhënave, njësia e matëse e të cilës është numri i bitëve në një sekondë. Me instruksionin `Serial.print(x)` në ekranin e monitorit serik shtypen vlerat e variablave.

Strukturat për zgjedhjen e mundësive përbëhen nga deklaratat: `if`, `if...else` dhe `else`. Deklarata `if` kontrollon kushtin dhe nëse plotësohet (`true`), atëherë i ekzekuton instruksionet.

Bibliotekat janë nënprograme që mundësojnë komunikimin ndërmjet pajisjeve periferike dhe mikrokompjuterit Arduino Uno. Mjedisi zhvillimor Arduino përmban disa biblioteka standarde dhe ne mund t'i shohim të njëjtat duke shtypur (`click`) në **Sketch > Import Library** në vetë menynë.

Brezi i vlerave që mund t'i jap senzori është shumë më e madhe se brezi i vlerave që mund të maten në mjedisin real. Prandaj, duhet të përshtaten vlerat maksimale dhe minimale të sensorit brenda sekondave të parë pas lëshimit në punë të mikrokompjuteri dhe sensorit. Kjo procedurë quhet kalibrim.

Pjesë përbërëse të mikrokompjuterit Raspberry Pi janë: çipi sistemor Broadcom BCM2837 SoC (System on Chip), memoria RAM, radio transmetuesi, katër konektorët USB, konektori RJ45, HDMI, konektorët special për kamerën dhe ekranin LCD dhe një prizë 40-pinëshe për pajisjet hyrëse-dalëse.

Raspberry Pi nuk ka memorie të përhershme të integruar, por për ruajtjen e të dhënave dhe të sistemit operativ përdoret një kartelë SD me një memorie prej minimum prej 16 GB. NOOBS (New Out-Of-Box Software) është softuer special që lejon zgjidhjen e njërit prej disa sistemeve operative për mikrokompjuterin Raspberry

Pi dhe instalim automatik me disa klikime të mausit. Ndër sistemet operative të ofruara është edhe sistemi operativ Raspbian.

Kategoria e Programing në menynë e sistemit operativ Raspbian përmban programe zhvillimore, mes të cilëve është ThonnyPython IDE.

Instruksionet për punë me taster janë: Button(), wait_for_press(), wait_for_release(), held_time, hold_time, is_held dhe is_pressed.

Instruksionet për të punuar me diodë LED janë: LED(), blink(on_time=1, off_time=1), off(), on() dhe toggle().

Pyetje dhe detyra:

1. Cilët komponentë harduerikë i përmban platforma Arduino Uno?

2. Numëro disa modele të ndryshme të mikrokompjuterëve nga seria Arduino?

3. Cilat janë tre ndryshimet kryesore midis mikrokompjuterëve Arduino Uno dhe Arduino Nano?

4. Përshkruaj mundësitë e lidhjes me pajisje tjera të mikrokompjuterit Raspberry Pi, krahasuar me mikrokompjuterin Arduino Uno!

5. Cilat janë avantazhet dhe disavantazhet e mikrokompjuterit Arduino Uno në krahasim me mikrokompjuterin Raspberry Pi?

6. Cilat janë dy mënyrat e fuqizimit të mikrokompjuterit Arduino Uno?

7. Si mund të shfrytëzohen pinat digjital të mikrokompjuterit Arduino Uno për transmetimin e sinjaleve analoge?

8. Sa indikator me dioda LED përmban mikrokompjuteri Arduino Uno dhe për çfarë përdoret secili?

9. Numëro disa pajisje hyrëse për mikrokompjuterin Arduino Uno!

10. Cilat hapje në sipërfaqen e protoplakës janë të lidhura elektrikisht?

11. Sa pina ka një taster dhe cilët prej tyre janë të lidhur elektrikisht?

12. Çfarë duhet pasur kujdes gjatë lidhjes së diodës LED në qark elektrik?

13. Shpjego si funksionon servo motori?

14. Çfarë destinimi ka komponenta piezo?

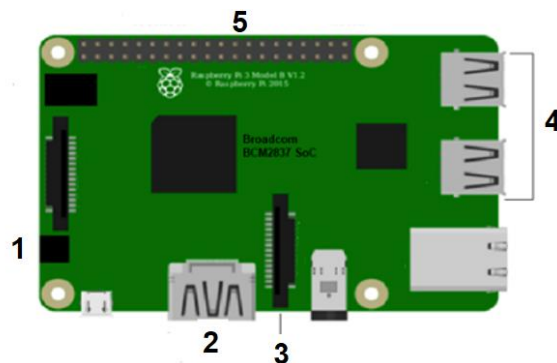
15. Për çfarë shërbejnë kondensatorët dhe rezistorët gjatë lidhjes së pajisjeve hyrëse-dalëse me mikrokompjuterin Arduino Uno ose Raspberry Pi?

16. Numëro disa funksione të mburojave Arduino!

17. Sqaro si lidhet protoplaka me furnizimin e mikrokompjuterit Arduino Uno?

18. Çfarë aplikacionesh mund të krijohen me mikrokompjuterin Raspberry Pi?

19. Emërto pjesët përbërëse të mikrokompjuterit Raspberry Pi në figurën e mëposhtme, të shënuar me numra!



20. Bazamenti GPIO (General Purpose Input Output) në mikrokompjuterin Raspberry Pi 3 ka 40 pina, por numrat GPIO ndryshojnë nga numrat fizik të pinave. Sqaro!

21. Numëro cilat kategori ndodhen në menynë e sipërfaqes punuese të sistemit operativ Raspbian?

22. Shpjego procedurën e instalimit të një drajveri pas instalimit të programit zhvillimor dhe lidhjes së mikrokompjuterit Arduino Uno me një kompjuter personal?

23. Numëro pjesët përbërëse të mjedisit zhvillimor të mikrokompjuterit Arduino Uno!

24. Për çka shërben monitori serik në përbërje të mjedisit zhvillimor të mikrokompjuterit Arduino Uno?

25. Cilat janë elementet përbërëse themelore të gjuhës programuese C++?

26. Çka nënkuptojmë me nocionin “sintaksë në programim”?

27. Numëro llojet themelore të të dhënave që përdoren në gjuhënprogramuese C++!

28. Çka paraqet emri, vlera dhe lloji i të dhënës `bool isCodingFun = true;`?

29. Cilat janë shenjat për të dhënat tekstuale dhe për vargun?

30. Sa vlera mund të kenë të dhënat logjike?

31. Çka paraqesin operatorët në gjuhën programuese C++?

32. Cili është operatori për caktimin e vlerës së variablës?

33. Çfarë rezultati do të merret pas ekzekutimit të instruksionit `x=13%5;`?

34. Shkruaj shprehjen e shkurtuar për instruksionin aritmetik `x=x-1!`

35. Cili është kushti për ekzekutimin e bllokut të instruksioneve nën strukturën `if (x!=y) {...}` ?

36. Cilët janë parametra të instruksionit `digitalWrite ()`?

37. Për çka shërben konvertuesit analogo-digjital në përbërje të mikrokompjuterit Arduino Uno?

38. Komento kodin e mëposhtëm:

```
void loop() {  
    val = analogRead(analogPin);  
    analogWrite(ledPin, val / 4);  
}
```

39. Për çfarë shërben instruksioni `tone (pin, frequency);`?

40. Me cilin instruksion matet koha nga fillimi i ekzekutimit të programit në mikrokompjuterin Arduino Uno?

41. Sqaro instruksionin constrain (x, a, b)!

42. Cili instruksion përdoret për resetimin e një biti të caktuar në variablën e dhënë?

43. Si quhet vegla për monitorimin e të dhënave të transmetuara përmes pinave RX dhe TX?

44. Cilat janë struktura të domosdoshme në programet për mikrokompjuterin Arduino Uno dhe cili është funksioni i tyre?

45. Çfarë shkruhet në kllapat e vogla dhe të mesme të strukturës if () {...} ?

46. Emërto strukturën për zgjedhjen e mundësive!

47. Cili është ndryshimi midis strukturave while...do dhe do....while?

48. Sa herë do të përsëritet cikli në kodin e mëposhtëm:

```
int x = 0;
do {
  delay(50);
  x = readSensors();
} while (x < 100);
```

49. Cilat tri deklaratat shkruhen në kllapa të vogla të strukturës for (...){...}?

50. Çfarë paraqet secili parametër në instruksionin LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)?

51. Komento instruksionin servo.attach (pin, min,max);?

52. Komento kodin e mëposhtëm:

```
void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(9, val);
}
```

53. Cilat janë avantazhet dhe disavantazhet e gjuhës programuese Python?

54. Me çka dallohen veglat kompajler dhe interpreter?

55. Pse në gjuhën programuese Python nuk ka nevojë për deklarimin e variablave?

56. Cila komandë e gjuhës programuese Python përdoret për krijimin e funksioneve të reja?

57. Pse themi se gjuha programuese Python është e orientuar drejt softuerit?

58. Cilën bibliotekë nga gjuha programuese Python e përdorim për të punuar me pajisjet hyrëse-dalëse-излезни уреди?

59. Si janë të shënuar pinat GPIO me numrat fizik 1,2,3 dhe 5? Shpjego funksionin e tyre!

60. Sqaro se në çka ndryshojnë instruksionet e mëposhtme: import gpiozero, from gpiozero import Button, from gpiozero import ButtonBoard!

61. Numëro disa pajisje hyrëse analoge për mikrokompjuterin Raspberry Pi!

62. Shpjego, se si mikrokompjuteri Raspberry Pi vendos nëse një pajisje hyrëse analoge është aktive apo jo!

63. Cilat parametra janë karakteristike për instruksionin Button()?

64. Ku haset dhe për çka shërben instruksioni wait_to_release?

65. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import Button
from signal import pause
def say_hello():
    print("Hello!")
    button = Button(2)
button.when_pressed=say_hello
pause()
```

66. Pse nevojitet kondensator gjatë lidhjes së fotorezistorit me mikrokompjuterin Raspberry Pi?

67. Për çfarë shërbejnë dy konektorët e të dhënave në sensorin e distancës?

68. Cila bibliotekë dhe klasë përdoret për të vendosur pajisjen në regjimin e pauzës?

69. Numëro disa pajisje dalëse që janë të lidhura me mikrokompjuterin Raspberry Pi nëpërmjet konektorit USB!

70. Në cilën klasë përdoret dhe çfarë rezultati jep instruksioni distance?

71. Cili instruksion përdoret për ndryshimin e gjendjes së diodës LED?

72. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import LED
from signal import pause
red =LED(17)
red.blink()
pause()
```

73. Si ndikon gjerësia e impulseve të moduluara në gjerësi në intensitetin e dritës të diodës PWMLED?

74. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import PWMLED
from time import sleep
led =PWMLED(17)
while True:
led.value=0
sleep(1)
led.value=0.5
sleep(1)
led.value=1
sleep(1)
```

75. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import LEDBoard
from signal import pause
leds=LEDBoard(5, 6, 13, 19, 26, pwm=True)
leds.value= (0.2, 0.4, 0.6, 0.8, 1.0)
pause()
```

76. Cilat janë tri ngjyrat kryesore dhe çfarë vlerash duhet të kenë ato për të fituar të bardhënë dhe të zezën?

77. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import TrafficLights
from time import sleep
from signal import pause
lights =TrafficLights(2, 3, 4)
def traffic_light_sequence():
    while True:
        yield (0, 0, 1)
        sleep(10)
        yield (0, 1, 0)
        sleep(1)
        yield (1, 0, 0)
        sleep(10)
        yield (1, 1, 0)
        sleep(1)
lights.source=traffic_light_sequence()
pause()
```

78. Me cilin instruksion mund të ndryshohet shpejtësia e rrotullimit të motorit DC?

79. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import MotionSensor, LED
from signal import pause
pir=MotionSensor(4)
led =LED(16)
pir.when_motion=led.on
pir.when_no_motion=led.off
pause()
```

80. Cili parametër përcakton shpejtësinë e motorit DC kur lidhet me një mikrokompjuter Raspberry Pi?

81. Për çfarë shërben instruksioni value() gjatë punës me servo motor?

82. Komento instruksionet dhe funksionin e kodit!

```
from gpiozero import Servo
```

```
from time import sleep
servo =Servo(17)
while True:
    servo.min()
    sleep(2)
    servo.mid()
    sleep(2)
    servo.max()
    sleep(2)
```

Ushtrime praktike për programin mësimor Sistemet kompjuterike

1. Masat për mbrojtje dhe siguri gjatë punës

Nxënësit në mënyrë të pavarur i kryejnë ushtrime praktike në grupe, jo më shumë se 2-3 nxënës, nën mbikëqyrjen e mësuesit lëndor. Për hir të kryerjes efikase dhe të sigurt të ushtrimeve, si dhe për trajtimin e duhur të pajisjeve elektronike, nxënësit duhet t'u përmbahen rregullave të mëposhtme.

- Lidhja e komponentëve elektronike kryhet ekskluzivisht në gjendjen pa tension.
- Pajisjet elektrike nuk duhet të preken me duar të lagura ose të njoma.
- Nuk lejohet mbajtja e njëkohshme e një përçuesi, instrumenti ose elementi tjetër nën tension me një dorë dhe një përçuesi tjetër me dorën tjetër.
- Veglat e punës duhet të kenë doreza të izoluara.
- Zjarri i shkaktuar nga rryma elektrike nuk shuhet me ujë.
- Pas shkëputjes nga qarku elektrik, kondensatorët elektrolitikë duhet të shkarkohen me lidhje të shkurtër të kontakteve të tyre.
- Kur punoni nën tension mos prekni përçues, rezistorë ose bateri për shkak të ngrohjes së tyre.
- Mbetjet elektronike të grumbullohen në vende të posaçme, të caktuara.
- Para fillimit të çdo ushtrimi praktik, nxënësi duhet me kujdes të lexojë kërkesat për kryerjen e ushtrimit dhe të përsërisë për mënyrën e funksionimit të komponentëve elektronike të cilat përdoren për ushtrimin.
- Komponentët elektronikë duhet të lidhen me kujdes, ngadalë dhe pa përdorimin e forcës.
- Telat për lidhjen e komponentëve elektronike të mos ndërthuren dhe të shtrëngohen së tepërmi.
- Pas lidhjes së komponentëve elektronike në një qarkun elektrik të thirret mësimdhënësi që të kryhet kontrolli, pas miratimit të të cilit kyçen tensionet e nevojshme.
- Për të gjitha defektet, dëmtimet ose mangësitë në komponentë ose pajisjet menjëherë të informohet mësuesi lëndor.

2. Ushtrime praktike për instalimin e harduerit në sistemin kompjuterik personal (PC)

2.1. Masat për mbrojtje dhe siguri gjatë punës me komponentë harduerikë të kompjuterit personal

- Me qëllim të parandalimit të goditjes elektrike, para se të filloj instalimi të shkyçet furnizimi me energji i kompjuterit.
- Të kontrollohet korrektësinë e të gjitha kablove dhe konektorëve përpara se të kyçet furnizimi.
- Të përdoret pajisje e furnizimit me energji me certifikatë përkatëse nga vetë prodhuesi.
- Detyrimisht të lexohet dokumentacioni tekniko-teknologjik për modelin konkret të pllakës amë sepse mund të ekzistojnë dallime të caktuara në parametra dhe shenja.
- Meqenëse çipat elektronikë janë të ndjeshëm ndaj elektricitetit statik, është e nevojshme të preknin ndonjë sipërfaqe metalike, për shembull vetë shtëpiza. Në këtë mënyrë vjen deri te shkarkimi i trupit të njeriut.
- Që të shmangët qarku i shkurtër, të mos preken konektorët dhe bazamenti me objekte metalike si vidha ose kaçavida.
- Instalimi i komponentëve harduerikë të kryhet në sipërfaqe pune të palëvizshme dhe stabile.
- Kompjuteri duhet të mbrohet nga lagështia dhe papastërtitë sepse ato ndikojnë dëmshëm në kontaktet e pllakës amë dhe e reduktojnë ftohjen dhe ajrosjen, kështu që mund të vij deri te mbinxehja.
- Përdoruesi duhet të shkyç në mënyrë të rregullt kompjuterin për të parandaluar dëmtimin e mundshëm të pjesëve të tij.

2.2. Ushtrim praktik për instalimin e pjesëve përbërëse të kompjuterit personal

1. Qëllimi i ushtrimit

Qëllimi i ushtrimit është instalimi i komponentëve harduerikë në brendësi të shtëpizës, gjegjësisht lidhjen e tyre me pllakën amë dhe pajisjen e furnizimit me energji elektrike përmes kablove dhe konektorëve përkatës. Pas instalimit të komponentëve dhe mbylljes së shtëpizës, duhet të lidhen tastiera, mausi dhe monitori që të kontrollohet korrektësia e lidhjes. Kuptohet, që të fitohet një kompjuter

funksional është e nevojshme të instalohet një sistem operativ dhe programe aplikative, e cila është temë studimi në njësinë modulare të radhës.

2. Koha për realizim: 4 orë mësimore

3. Komponentët e nevojshme

Për realizimin e këtij ushtrimi është i nevojshëm një kompjuter personal me komponentët harduerike të mëposhtme: procesor, ftohës i procesorit, pllakë amë, memorie RAM, pajisje për furnizim me energji elektrike, hard disk, pajisje CD ROM dhe kartelë grafike. Pasi që gjatë trajtimit të komponentëve harduerikë mund të vij deri te dëmtimi i tyre, rekomandohet një sistem kompjuterik me konfigurim dhe performancë më të dobët, por të jetë akoma funksional. Gjithashtu, nga veglat janë të nevojshme dy kaçavida (të sheshta dhe plus). Mundet të përdoret edhe një rrip dore antistatike, kapëse plastike, pastë termike, alkool dhe një leckë pastrimi mikrofibër.

4. Përgatitja për ushtrimin

- Çdo kompjuter ka specifikat e veta, prandaj para fillimit të ushtrimit praktik është e nevojshme të rishikohet dokumentacioni tekniko-teknologjik për secilin komponent në veçanti.
- Në figurën 2.1. është paraqitur pamja e modelit më të vjetër ASUS P5Q Pro Turbo LGA 755 Intel, ndërsa në figurën 2.2. pamja e modelit më të ri të pllakës amë ASRock Z59p Extreme WiFi GE LGA 1200 Intel. Për të dy modelet, në tabelën 2.1. shkruaj emrin e komponentëve harduerikë të shënuara.

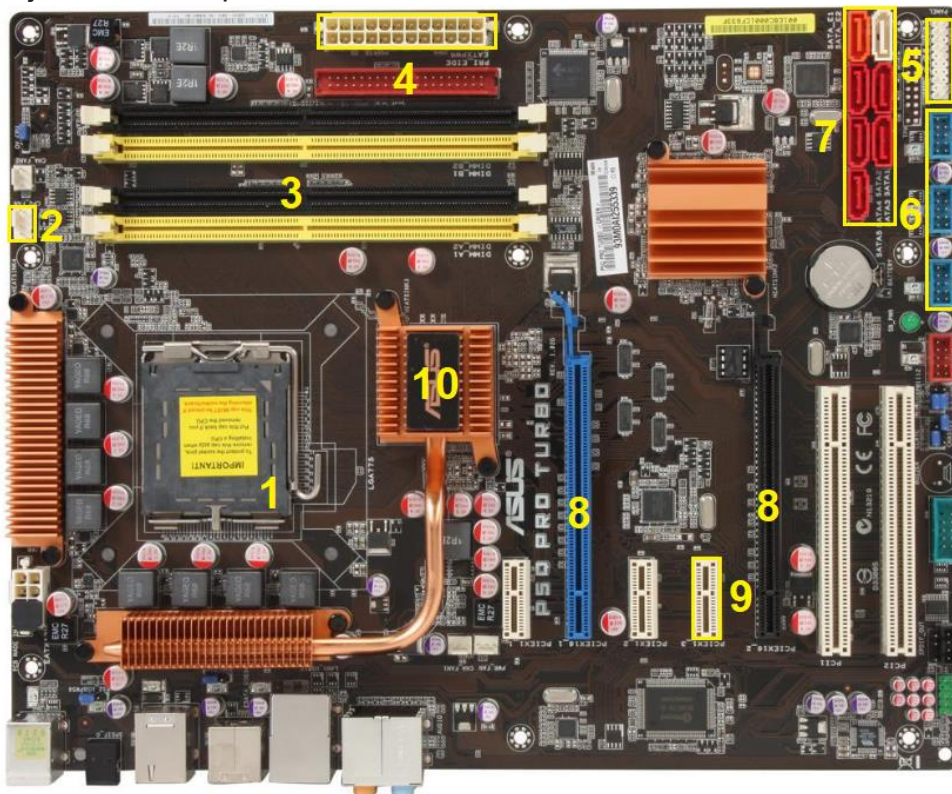


Figura 2.1. Pllaka amë, modeli ASUS P5Q Pro Turbo LGA 755 Intel

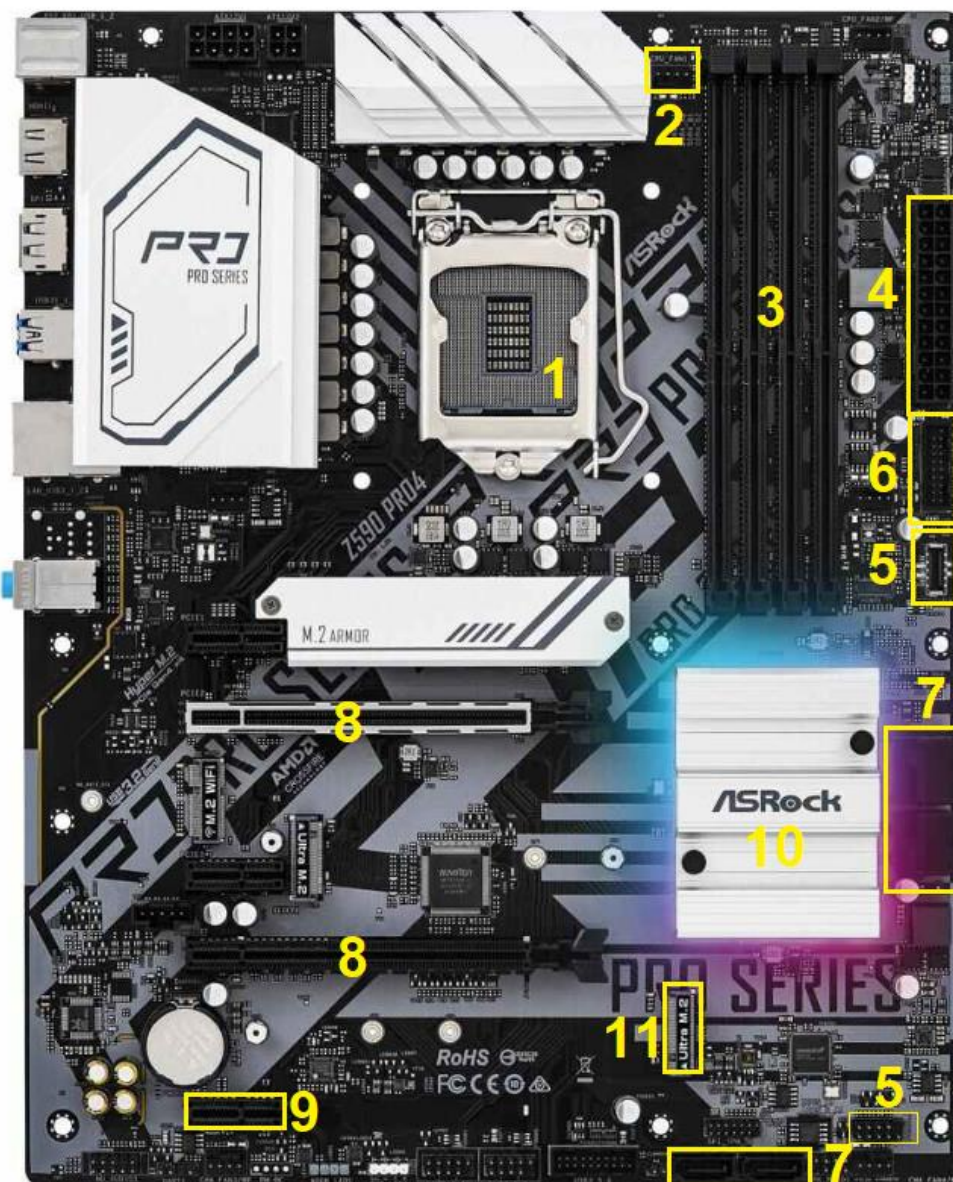


Figura 2.2. Pllaka amë, modeli ASRock Z59p Extreme WiFi GE LGA 1200 Intel

Nr. ren.	ASUS P5Q Pro Turbo LGA 755 I	ASRock Z59p Extreme WiFi GE LGA 1200
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11	X	

Tabela 2.1. Komponentët e pllakës amë dhe identifikimi i tyre

5. Instalimi i procesorit

(1) përgatitje

- Përpara fillimit të instalimit të procesorit në bazën e tij të kontrollohet kompatibiliteti i tyre.
- Të kryhet krahasimi i bazës së procesorit, modeli LGA 1151 (figura 2.1) dhe modeli AMD AM3 dhe të nënvizohen specifikat e tyre. Të theksohen procesorët që janë kompatibil me këto modele të bazamenteve.

(2) instalimi

Nëse bëhet fjalë për pllakë amë të re, bazamenti i procesorit duhet të jetë i mbuluar me foli mbrojtëse të verdhë. Në anën e djathtë të bazamentit ndodhet një **levë metalike**, e cila duhet të shtyhet poshtë, të tërhiqet në të djathtë dhe të ngrihet lartë, me ç'rast do të ngrihet edhe kapaku i bazamentit. Folia mbrojtëse e verdhë largohet dhe ajo duhet të ruhet në rast se nxiret procesori. E vendosim procesorin në bazament, me pinat e kthyera nga poshtë, duke u kujdesur që dy zgavrat anësore të procesorit të përkojnë me dy zgjatimet në bazament. Lëshohet kapaku deri në vidë, ulet leva metalike dhe tërhiqet në të majtë, që të vendoset në vendin e vet. [10]

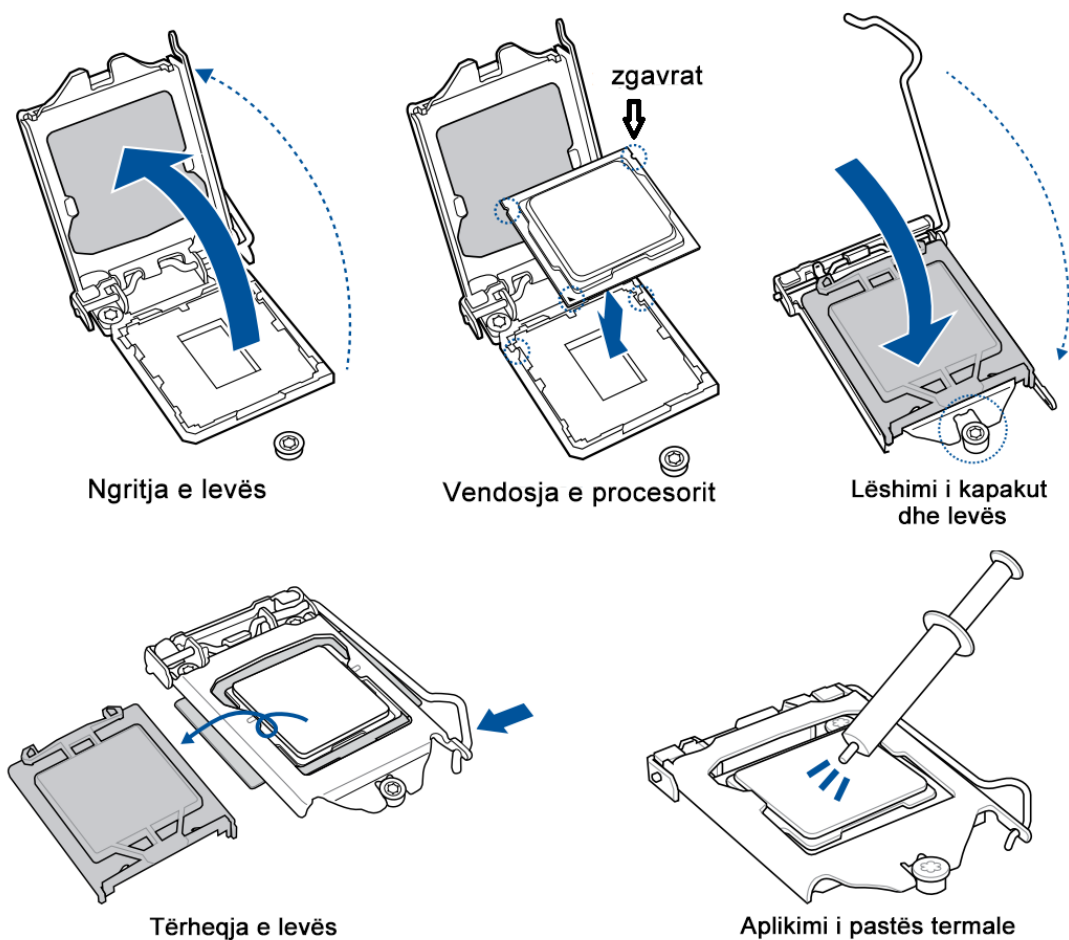



Figura 2.3. Hapa gjatë instalimit të procesorit

- (3)  Procesori duhet të mbahet nga skajet dhe të mos preken pinat e artë. Procesori nuk duhet të shtypet.

6. Instalimi i ftohësit

(1) përgatitje

- Gjatë zgjedhjes së ftohësit për një procesor, duhet pasur kujdes nëse i njëjti është kompatibil me procesorin përkatës. Ftohësit mund të jenë me: dimensione të ndryshme, llojin e ftohjes (me ajër ose me lëng), mënyrën e montimi (me vida ose ulje pistoni).
- Për shembull, të hulumtohet se cilët ftohës janë kompatibil me modelin LGA 1151 të bazamentit të procesorit.

Komenti: _____

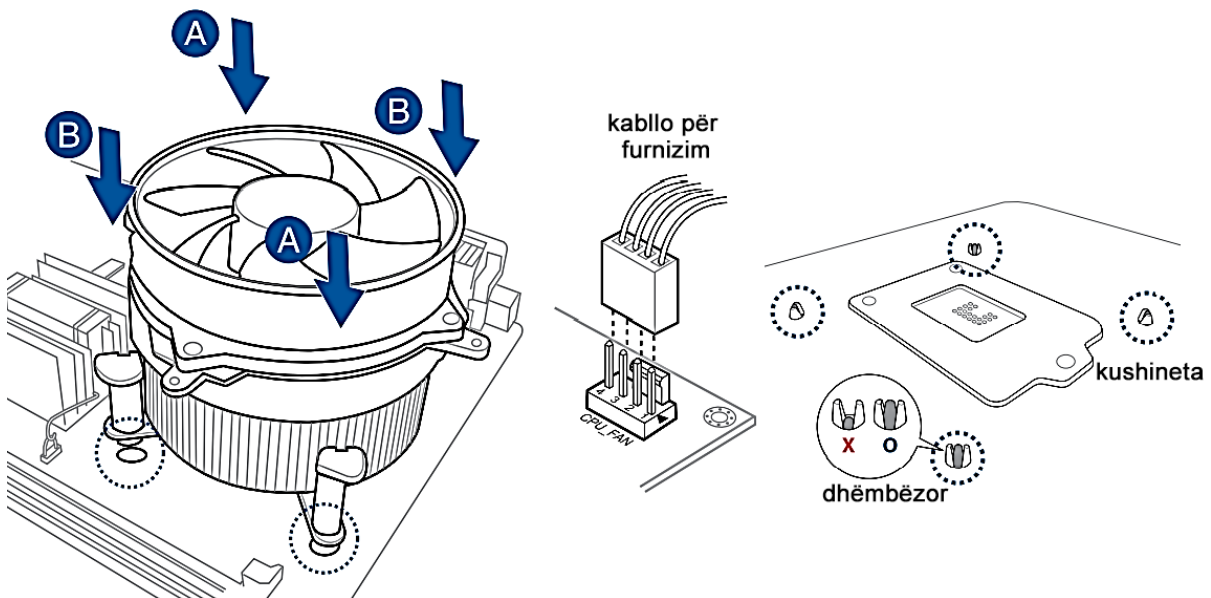


Figura 2.4. Hapa gjatë instalimit të ftohësit për procesor

(2) instalimi

Para instalimit të ftohësit duhet të kontrollohet nëse ka nevojë për aplikimin e **pastës termike** mbi pjesën e sipërme të procesorit. Pasta termike e zvogëlon temperaturën nga 20° deri në 30°. Para se të aplikohet pastë e re, duhet të largohet e vjetra dhe për këtë është mirë të përdoret alkool 90% dhe një leckë mikrofibër. Procesori së pari duhet të hiqet nga bazamenti. Në leckë aplikohet një sasi e vogël alkooli, për afërsisht sa madhësia e një pullë postare, ndërsa më pas butësisht kalohet mbi pjesën e sipërme të procesorit. Pasta aplikohet në mes, sa madhësia e gjysmës së bizeles. Pasta do të shpërndahet nën ndikimin e nxehtësisë së procesorit dhe peshës së ftohësit. Përpara se të vendosim ftohësin mbi procesor, kontrollojmë nëse kanalet e kokave të **dhëmbëzorëve plastikë** janë të drejtuara drejt qendrës së ftohësit. Zgjedhim dy dhëmbëzor që janë të

vendosur në diagonal dhe i shtypim. Pasi t'i kemi shtypur të katër dhëmbëzorët, nga pjesa e pasme e pllakës amë kontrollojmë nëse dhëmbëzorët janë barazuar me kushinetat e tyre.

(3)



Gjatë instalimit të ftohësit duhet pasur kujdes që të mos dëmtohet pllaka amë për shkak të aplikimit të forcës së tepërt.

Me kujdes të trajtohet pasta termike sepse e njëjta është substancë toksike dhe rekomandohet përdorimi i dorezave mbrojtëse.

7. Instalimi i memories RAM

(1) përgatitje

- Nga zgjedhja e pllakës amë varet lloji dhe kapaciteti i memories RAM. Fillimisht zgjedhim llojin (DDR2, DDR3, DDR4 ose DDR5), më pas kapacitetin e secilit prej moduleve. Për shembull, modulet DDR4 mund të kenë një kapacitet prej 2GB deri në 32GB. Në të njëjtën kohë, duhet të zgjedhim edhe shpejtësinë e modulit RAM. Për shembull, modulet DDR4 mund të jenë me shpejtësi nga 2133 MHz deri në 5000 MHz. Duhet të ketë përshtatje midis shpejtësisë së memories RAM dhe vetë procesorit.
- Për shembull, kapaciteti maksimal i memories RAM për pllakën amë ASUS P5Q Pro Turbo LGA 755 Intel (figura 2.1) është 16 GB, ajo mbështet modulet DDR2, me shpejtësi optimale prej 800 MHz. Hulumtoni memorien RAM për pllakën amë ASRock Z59p Extreme WiFi GE LGA 1200 Intel (figura 2.2).

Komenti: _____

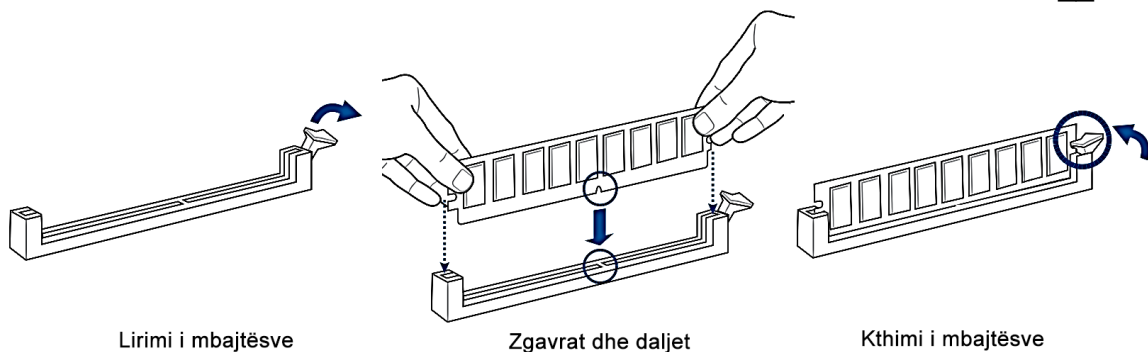



Figura 2.5. Hapa gjatë instalimit të memories RAM

(2) instalimi

Pllakat amë mund të kenë 1, 2, 4 ose 8 slote për vendosjen e moduleve të memorieve RAM. Ato mund të vendosen individualisht (lidhje me një kanal) ose në çift (me dy kanale). Lidhja me dy kanale e rrit shpejtësinë e transferimit të të dhënave për dyfishin. Për lidhjen me dy kanale, modulet RAM duhet të jenë identik (sipas llojit, shpejtësisë, kapacitetit, prodhuesit) dhe të vendosen në slote me të njëjtën ngjyrë. Nëse slotet e RAM-it nuk janë me ngjyra të ndryshme, atëherë ato vendosen njëri mbi tjetrin, për shembull

me numrat rendor 2 dhe 4 ose 1 dhe 3, por është mirë që të kontrollohet në dokumentacionin tekniko-teknologjik të vetë pllakës amë.

Mes pinave ndodhet një zgavër e vogël dhe kjo zgavër duhet të jetë në linjë me daljet në slotet DIMM (bazamentin e RAM-it). Përpara se të fusim modulën e memories, është e nevojshme të lirohen mbajtëset anësore të slotit të lirë.

- (3)  Ngjashëm si te procesori, edhe te memoret RAM duhet pasur kujdes për elektricitetin statik. Prandaj, është mirë që përpara instalimit të paktën të prekim shtëpizën metalike. Gjithashtu, modulet RAM duhet të mbahen anash dhe në asnjë mënyrë të preken pinat e artë. Nëse lloji i slotit të RAM-it nuk përputhet me llojin e modulit të memories RAM, gjatë instalimit do të vij deri te dëmtimi i komponentëve harduerikë.

8. Montimi i pllakës amë në shtëpizë

- Formatet (madhësitë) më të zakonshme për pllakat amë janë: ATX, mATX (angl. mini) dhe EATX (angl. extended). Zgjedhja e shtëpizës varet nga formati i pllakës amë. Më së miri është që procesori, ftohësi dhe memoria RAM të instalohen para se pllaka amë të montohet në shtëpizë.



Figura 2.6

(1) përgatitje

- Pastaj në pllakën amë lokalizohen vrimat për vendosjen e vidave të fiksimit. Këto vrima janë të shënuara në veçanti si në figurën 2.6. Numri dhe renditja e tyre ndryshon, në varësi të llojit të pllakës amë.
- Të lokalizohen vrimat për përforcimin e pllakave amë: ASUS P5Q Pro Turbo LGA 755 Intel (figura 2.1.) dhe ASRock Z59p Extreme WiFi GE LGA 1200 Intel (figura 2.2.)

Komenti: _____

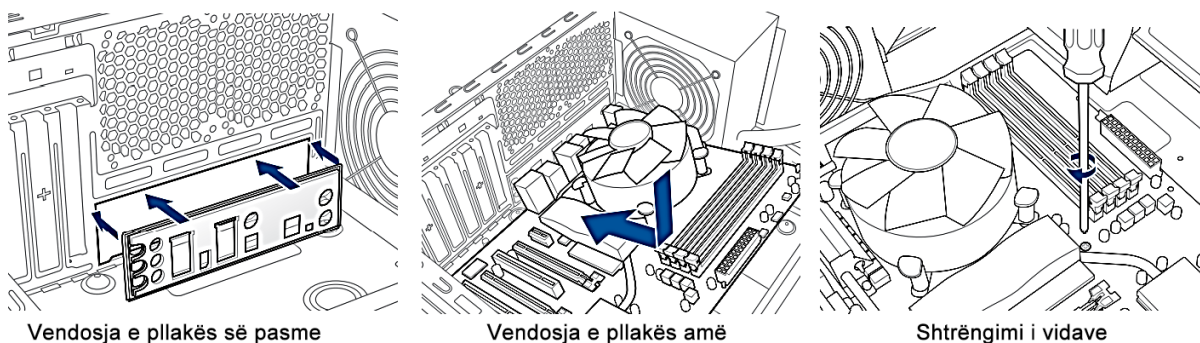


Figura 2.7. Hapa gjatë instalimit të pllakës amë

Pllakat amë vijnë të kompletuara me një pllakë të pasmë që ka vrima për llojet e ndryshme të konektorëve: USB, HDMI, konektor PS/2 për maus dhe tastierë, portë audio, konektor rrjeti RJ-45, etj. Së pari montohet pllaka e pasme.

Pastaj lokalizohen foletë për vida në brendësi të shtëpizës. Pllaka amë vendoset ashtu që vrimat për vidat e pllakës amë të përputhen me foletë për vidat e shtëpizës.

Pasi të shtrëngohen vidat, është e nevojshme të lidhen konektorët e rrymës në pllakën e përparme dhe konektori për furnizimin e ventilatorit të vetë shtëpizës. Në pllakën e përparme të shtëpizës ka një buton për furnizimin, një buton për resetimin, porta USB të përparme, konektor audio të përparme dhe dioda LED si indikator për furnizimin dhe për hard diskun. Konektorët e tyre të furnizimit duhet të lidhen me bazamentin e pllakës amë, e cila është e shënuar me shkronjën F (Front). Konektorët USB dhe audio lidhen më thjeshtë për shkak të renditjes unike të pinave të tyre. Te lidhja e butonave të furnizimit dhe resetimit nuk është me rëndësi polariteti i tyre. Te dioda LED duhet pasur kujdes në shenjat. Zakonisht, teli i bardhë dhe i zi janë plus, ndërsa telat me ngjyra të tjera janë minus ose tokëzim. Në fund, në pllakën amë duhet të lokalizohet bazamenti me shenjën SYS_FAN dhe PWR_FAN dhe në të vendoset konektori i furnizimit dhe ventilatorit të shtëpizës.

(2) Instalimi

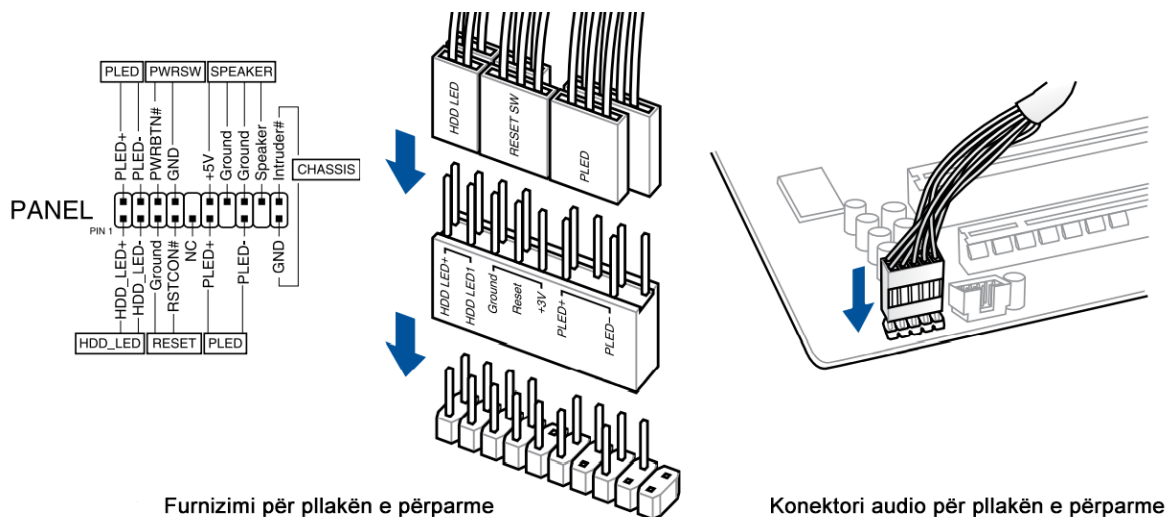


Figura 2.8. Lidhja e pllakës amë me furnizimin

(3)



Gjatë vendosjes së maskës së pasme, duhet pasur kujdes në skajet e mprehta.

Gjatë shtrëngimit të vidave nuk duhet të përdoret forcë e tepërt. Duhet të vidhoseni sa më shumë vida që të jetë e mundur për një stabilitet më të madh mekanik të pllakës amë.

9. Instalimi i pajisjeje për furnizim

Ekzistojnë pajisje furnizimi modulare dhe jo-modulare. Dallimi kryesor është se në pajisjet jo-modulare kabllot janë të lidhura në fabrikë me pajisjen dhe nuk mund të hiqen prej saj.

Karakteristika më e rëndësishme e pajisjeve të furnizimit është fuqia e tyre dhe sillet nga 300W deri në 1600W. Pa furnizim adekuat me energji elektrike, komponentët harduerit do të funksionojnë me performancë dhe stabilitet të reduktuar. Për këtë arsye është e nevojshme të kontrollohet sa është konsumi i energjisë i secilit komponent dhe të llogaritet fuqia e nevojshme e pajisjes së furnizimit. Konsumatorë më të mëdhenj të energjisë elektrike janë procesori dhe kartela grafike.

Sipas një konfigurimi kompjuterik të paracaktuar llogaritni fuqinë e pajisjes së furnizimit. Përdor kalkulatorin e linkut të më poshtëm

<https://www.newegg.com/tools/power-supply-calculator/>

(1) përgatitje

Komenti: _____

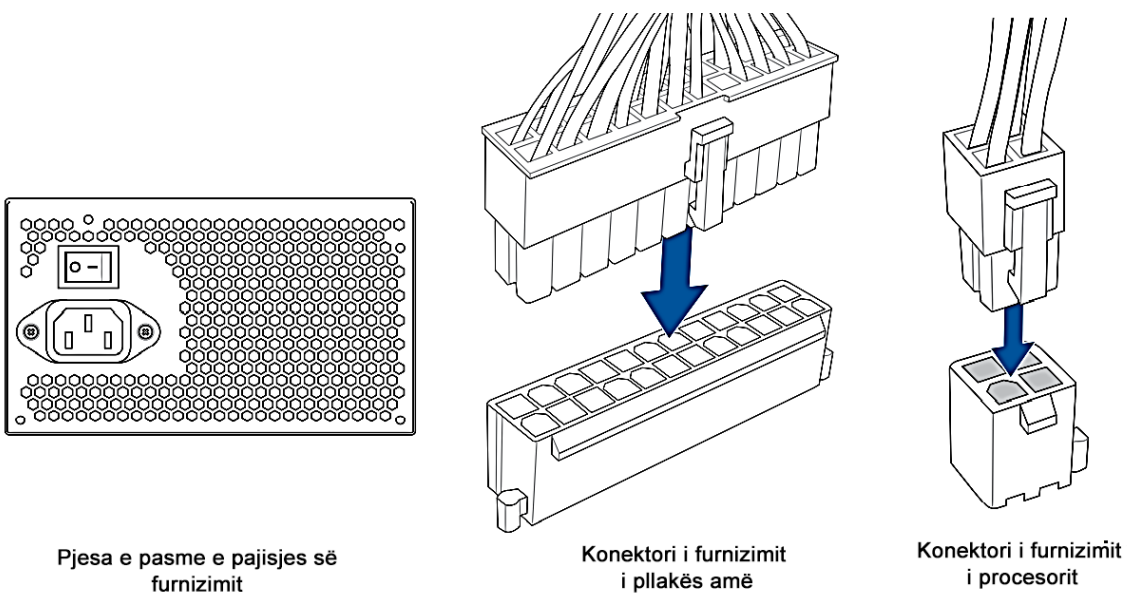



Figura 2.9. Instalimi i pajisjes së furnizimit

(2) instalimi

Njësia e furnizimit mund të vendoset në pjesën e sipërme ose të poshtme, në pjesën e pasme të shtëpizës. E vendosim pajisjen në ndarjen e brendshme, me ventilatorin e kthyer kah pllaka amë, e shtyjmë deri në pjesën e pasme dhe e njëjta përforcohet.

Njësia e furnizimit ka më tepër kabllot që duhet të lidhen me pllakën amë ose me një komponentë të caktuar. Është mirë që të gjitha kabllot për furnizim kanë një dizajn unik, prandaj nuk mund të gabohet gjatë lidhjes së tyre. Në pllakën amë lidhen dy konektor: ATX- kabllot me konektor 20+4 pina, i cili e

furnizon pllakën amë, ndërsa ATX 12V me 4 ose 8 pina, i cili e furnizon procesorin. Konektori me 8 pina siguron një fuqi prej 235W, ndërsa konektorit me 4 pina 155W. Për furnizimin e hard diskut dhe pajisjes përdoren dy lloje të kabllave: SATA dhe IDE me 4 pina, ndërsa të njëjtit do t'i shpjegojmë së shpejti. Kablloja me gjashtë pina për furnizimin e kartelës grafike është shënuar me shenjën VGA1 dhe është i njohur me emrin kabloja PCI Express.

- (3)  Përdorimi i pajisjes së furnizimit me tension të paqëndrueshëm mund të shkaktojë dëmtim të pllakës amë dhe komponentëve tjera harduerike.

10. Instalimi i hard diskut dhe pajisjes SSD

- Nxënësit duhet të rikujtojnë karakteristikat bazë të hard diskut: parimi i funksionimit, kapaciteti, dimensionet fizike, shpejtësia e transferimit të të dhënave, shpejtësia e rrotullimit, memoria kesh, konektorët për lidhje.
- Të krahasojë karakteristikat e dy modeleve të hard disqeve, me zgjedhje të rastit.

(1) përgatitje

Komenti: _____

- Të lokalizohen konektorët SATA për lidhjen e hard diskut me pllakën amë dhe hard diskut me pajisjen e furnizimit

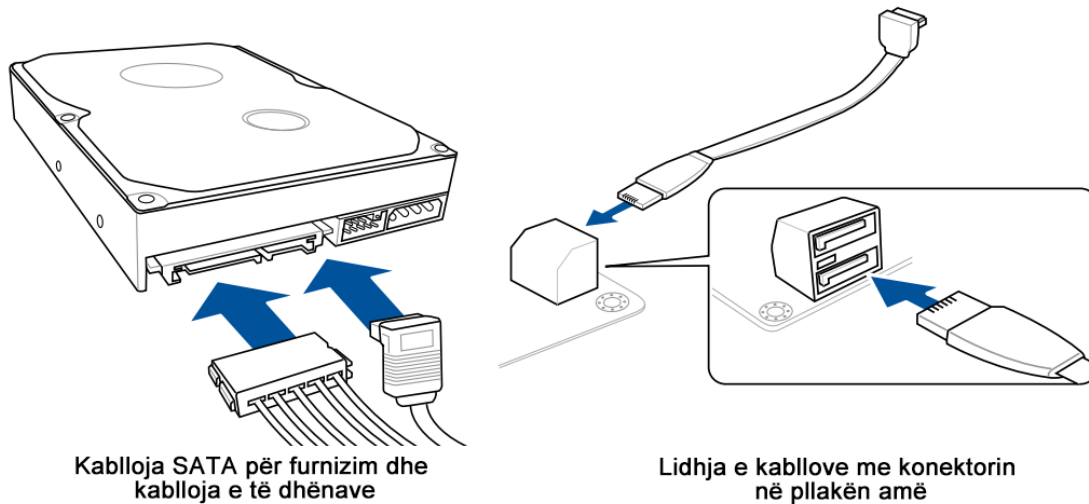


Figura 2.10. Lidhja e hard diskut me kablo SATA

Në brendësi të shtëpizës ekziston ndarje të veçantë me lartësi 9 cm. Në të vendoset hard disku, me ç'rast përpara duhet të shihen konektorët e tij në formën e shkronjës L. Në konektorin më të madh përforcohet **kablloja për**

(2) instalimi

furnizim nga pajisja e furnizimit, ndërsa në konektorin më të vogël vendoset **kabloja e të dhënave SATA**, i cili vjen së bashku me hard diskun. Skaji tjetër i kablos të të dhënave SATA lidhet në konektorin në pllakën amë të shënuar me SATA2, në formën e shkronjës L.

Pasi që pajisjet memoruese SSD SATA janë me të njëjtën madhësi dhe renditje të pinave si edhe hard disku, për lidhjen e tyre përdoren konektorët e njëjtë SATA për furnizim ose transferimin e të dhënave. Kuptohet, paraprakisht duhet të kontrollohet nëse në shtëpizë ekzistojnë dy ndarje për SSD dhe për hard disk.

(3)



Hard disku i sapo instaluar duhet së pari të inicializohet duke shtypur klikun e djathtë dhe duke zgjedhur opsionin "Initialize Disk". Pas hapjes së dritares së re siç tregohet në figurën 2.10. duhet të zgjidhet tabela e particioneve (ndarjeve), MBR (angl. Master Boot Record) ose GPT (angl. Globally Unique Identifier Partition Table). Particionet janë pjesë virtuale të hard diskut. Tabela e particioneve përmban informacione për atë se si janë të organizuar particionet, ku fillon dhe mbaron secili particion, cilët sektorë zë, ndërsa gjithashtu përmban edhe kod për startimin e sistemit operativ i njohur me emrin ngarkues i startimit.



Figura 2.11. Zgjedhja e tabelës së particioneve

Shkurtesa MBR në përkthim do të thotë regjistrim kryesor për startim, ndërsa shkurtesa GPT do të thotë tabela për particionet me identifikues global unik. Me tabelën MBR, mund të adresohen vetëm 2TB kapacitet i memories së hard diskut, ndërsa në tabelën GPT nuk ka kufizime të tilla, domethënë kapaciteti maksimal i memories është 9.7 ZB (1 zeta bajt është e barabartë me një miliard tera bajt). Tabela MBR mbështet vetëm 4 particione, njëra prej të cilave mund të konfigurohet si zgjeruar (angl

extended) dhe e njëjta mund të ndahet në 23 particione plotësuese. Tabela GPT përmban 128 particione të ndryshme, e cila është më se e mjaftueshme për aplikacionet e sotme reale. Le të theksojmë se tabela MBR është e lidhur me programin BIOS, ndërsa tabela GPT me programin UEFI. Kompjuterët me program BIOS nuk mund të kenë particione GPT.

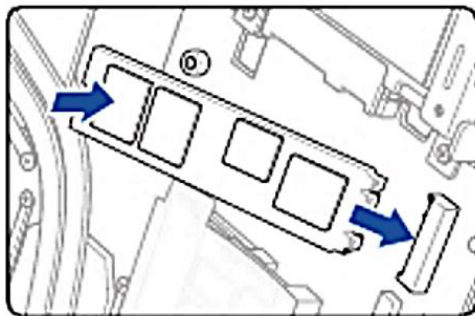
Në rast të instalimit të një hard disku të dytë, si përmirësim të konfigurimit ekzistues, në vetë sistemin operativ duhet të kryhet rishpërndarja e tij (angl allocate). Për këtë qëllim duhet të hapet programi Disk Management në përbërje të sistemit operativ Windows 10. Hard disku i sapo instaluar shënohet si hapësirë e pashpërndarë (angl. unallocated space) dhe me shtypjen e klikut të majtë duhet të zgjidhet opsioni "New Simple Volume". Gjithashtu ekziston mundësia që skedarët nga hard disku i vjetër të transferohen në hard diskun e ri. Kjo procedurë njihet si klonim dhe është e një niveli më të avancuar.

11. Instalimi i modulit M.2 SSD

- Gjatë instalimit të modulit M.2 SDD, duhet të kontrollojmë llojin e konektorit të tij, PCIe ose SATA. Nëse moduli dhe konektori i pllakës amë nuk kanë të njëjtën koneksion, ato nuk mund fizikisht të lidhen. Të lokalizohen dy konektor M.2 për lidhjen e moduleve memoruese SATA
- SSD në pllakën amë ASRock Z59p Extreme WiFi GE LGA 1200 Intel të paraqitur në figurën 2.2.

Komenti: _____

(1) përgatitje



Shtytja e modulit M.2 SSD në konektorin e tij



Përforcimi i modulit M.2 SSD me ndihmën e vidës

Figura 2.12. Instalimi i modulit M.2 SDD

(2) instalimi

Në një kënd të vogël, pinat e modulit M.2 SSD shtypen me kujdes në konektorin përkatës. Pasi të vendoset moduli në pllakën amë, ai fiksohet me ndihmën e një vide. Të theksojmë se ekzistojnë tri fole për vida në afërsi të konektorit M.2 sepse modulet M.2 SSD mund të jenë me tri gjatësi të ndryshme (22, 60 ose 80 mm) ndërsa gjerësia është gjithmonë e njëjtë.



(3) Gjatë rrotullimit të vidës për përforcimin e modulit M.2 SSD nuk duhet të përdoret forcë e tepërt.



12. Instalimi i pajisjeve optike

(1) përgatitje

- Në këtë kategori të pajisjeve përfshihen pajisjet CD, DVD dhe Blue-Ray të cilat për regjistrimin dhe leximin e të dhënave përdorin teknologjinë lazer. Ashtu si me komponentët harduerike të më mëparshme edhe këtu me rëndësi është zgjedhja. Disqe më kualitative janë disqet Blu-Ray, më pas vijjnë DVD-të dhe në fund CD-të. Kur themi kualitet, mendojmë në kapacitetin dhe rezolucionin e lartë të regjistrimeve video. Pajisjet Blu-Ray mund të lexojnë të gjitha llojet e disqeve, pajisjet DVD disqe DVD dhe CD, ndërsa pajisjet CD mund të lexojnë vetëm disqe CD. Në koston e pajisjes ndikon shpejtësia e leximit dhe shkrimit të të dhënave.
- Të bëhet krahasimi mes dy modeleve, pajisjeve DVD ose Blue-Ray, me zgjedhje të rastit. Të bëhet analiza e rentabilitetit në lidhje me furnizimin e një pajisjeje të jashtme ose të brendshme.

Komenti: _____

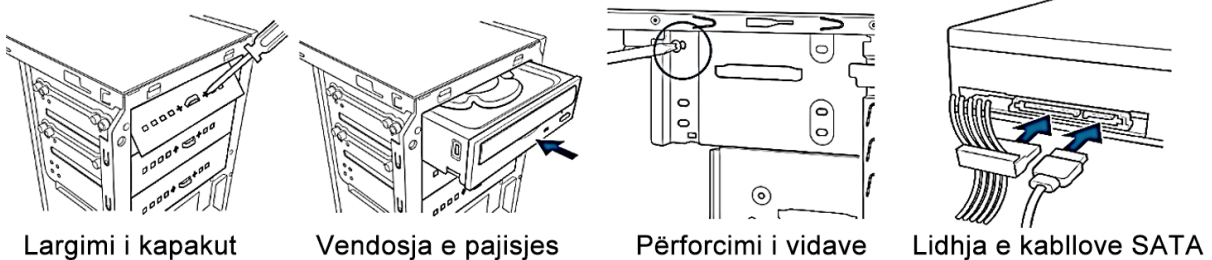


Figura 2.13. Instalimi i pajisjes optike

(2) instalimi

Pajisja optike montohet në anën e përparme të shtëpizës. Për këtë qëllim, është e nevojshme të largohet maska e përparme e shtëpizës, e cila është e përforcuar me dy dhëmbëzorë të futur në vrimat e pllakës amë. Pastaj duhet të largohet kapaku i ndarjes 14 cm të lartë. Pajisja DVD vendoset në ndarjen

dhe nga anash përforcohet ne vida. Ngashëm si edhe hard disku, pajisja DVD ka dy konektor SATA në formën e shkronjës L, njëri për furnizim dhe tjetri për të dhënat.



- (3) Disa ndarje përdorin binarë që të sigurojnë një pozitë më të qëndrueshme të pajisjes. Nëse është kështu, binarët duhet të ngjiten në secilën anë të pajisjes përpara se ajo të futet në ndarjen e shtëpizës. Pas instalimit të pajisjes duhet të instalohet drajveri i tij.

13. Instalimi i kartelës grafike

- Nxënësit duhet të rikujtojnë karakteristikat bazë të kartelave grafike: kapacitetin e video memories-RAM, gjatësinë fizike dhe konsumin e energjisë elektrike.
- Të krahasohen karakteristikat e dy modeleve të kartelave grafike, MSI GeForce GT 710 dhe MSI GeForce GTX 1050 Ti, të paraqitura në figurën 2.16. në pjesën e teorike të librit.

Komenti: _____

(1) përgatitje

- Të lokalizohen slotin PCI Expressx16 për lidhjen e kartelës grafike me pllakën amë.

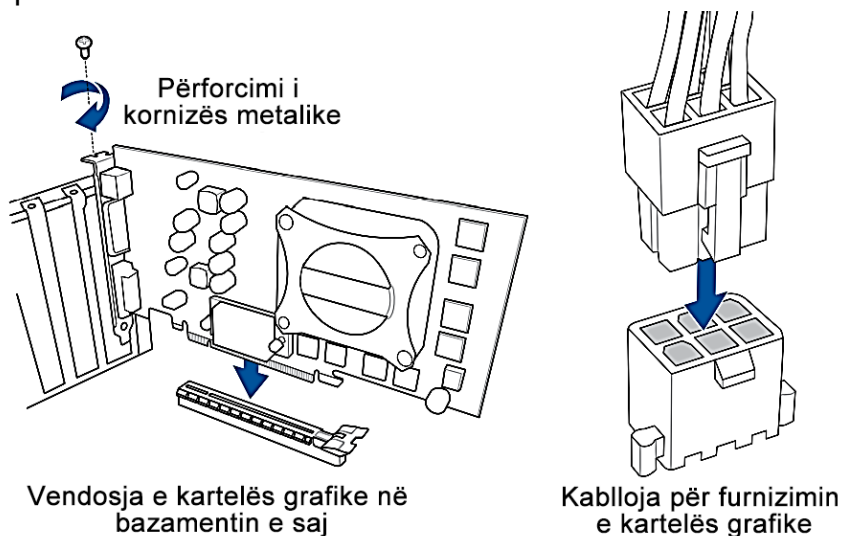



Figura 2.14. Vendosja e kartelës grafike në slotin PCI Expressx16

Karta grafike vendoset në slotin PCI Expressx16 në pllakën amë. Para fillimit të montimit, është e nevojshme të largohen kornizat metalike në pjesën e pasme të shtëpizës, duke i liruar vidat. Pasi të vendoset kartela grafike në slotin PCI Expressx16, është e nevojshme të përforcohet korniza metalike në vetë kartelën grafike, në pjesën e pasme të

(2) instalimi

shtëpizës, duke përdorur vida. Në fund, përforcohet kabloja gjashtë pinësh për furnizim i shënuar me shenjën VGA1.

- (3)**  Gjatë zgjedhjes së kartelës grafike, duhet pasur kujdes nëse ka hapësirë të mjaftueshme për montimin e saj në vetë pllakën amë, sepse modelet e reja të kartelave grafike janë me dimensione shumë më të mëdha se modelet e vjetra. Gjithashtu duhet të verifikojmë nëse pajisja e furnizimit mund të plotësojë konsumin e energjisë për kartelën grafike të zgjedhur. Ngjashëm si me instalimin e moduleve të memories RAM, përpara se të fusim kartelën grafike, është e nevojshme të lirohet mbajtësi anësor i slotit PCI Expressx16. Pas instalimit të kartelës grafike, duhet të instalohet edhe drajveri i saj. Nëse bëhet fjalë për zëvendësimin e kartelës grafike të vjetër me një të re, më së miri është që para instalimit të drajverëve të rinj, të deinstalohen drajverët e kartelës së vjetër.

14. Kontrollimi i funksionalitetit

Nëse pas instalimit të pjesëve përbërëse, kompjuteri nuk dëshiron të kyçet, duhet të kontrollohen elementet e mëposhtme:

- A janë vendosura saktë kabllo e furnizimit të kompjuterit dhe monitorit?
A ndriçon dioda LED e gjelbër për furnizimin, në pjesën e përparme të shtëpizës?
- A ndriçon dioda LED për furnizimin e monitorit?
- Përpara se të hapet shtëpiza, detyrimisht të largohet kabloja e furnizimit!
- A janë të gjithë lidhur mirë dhe të përforcuar të gjithë konektorët në pllakën amë?
- A është funksionale pajisja e furnizimi? Mund të maten tensionet e pinave të konektorit ATX 20+4.

Për diagnostifikim e problemit mund të ndihmojnë sinjalet zanore të dritës që gjenerohen nga kompjuteri gjatë kyçjes së tij. Kur dalim nga shtëpia, zakonisht sigurohemi kontrollojmë nëse kemi portofolin, çelësat, telefonin celular dhe gjërat e tjera që janë të domosdoshme për të nisur ditën me ne. E njëjta ndodh edhe kur kompjuteri fillon të punojë. Ai kontrollon nëse të gjitha pjesët harverike a janë funksional. Testi POST (Power On Self Test) bazohet në të dhënat e marra nga memoria CMOS. CMOS (angl. Complementary Metal Oxide Semiconductor) është memorie RAM me një kapacitet të vogël prej 100 deri në 200 bajt, me bateri të veten të vogël për furnizim. Për shembull, POST mat kapacitetin e memories RAM ekzistuese dhe të dhënat e mara i krahason me të dhënat e memories RAM CMOS. Të dhënat për llojin e memories së integruar janë të futura nga prodhuesi, por të njëjtat mund t'i ndryshojë edhe vetë shfrytëzuesi,

nëse zgjerohet ose ndryshohet lloji i memories ose futet një komponentë harduerikë shtesë. Nëse kompjuteri e "kalon" testin POST, ai lëshon një zë (beep) të shkurtër. Nëse kompjuteri nuk e "kalon" testin POST, ai lëshon disa zëra të shkurtër ose të gjatë në varësi të gjendjes së kompjuterit. Është e pa këndshme që **prodhues të ndryshëm përdorin kode të ndryshme zëri**. Më poshtë janë dhënë kodet zanore më të rëndësishme që i përdor kompania Dell.

1 sinjal zanor i shkurtër	→	Dështimi i memories BIOS ROM
2 sinjal zanor i shkurtër	→	Nuk është detektuar memoria RAM
3 sinjal zanor i shkurtër	→	Dështimi i pllakës amë
4 sinjal zanor i shkurtër	→	Dështimi i memories RAM
5 sinjal zanor i shkurtër	→	Dështimi i baterisë CMOS
6 sinjal zanor i shkurtër	→	Dështimi i kartelës grafike
7 sinjal zanor i shkurtër	→	Dështimi i procesorit

Për shembull, nëse sinjali zanor tregon se nuk ekziston memorie RAM, atëherë më e pakta që mund të bëjmë është të largojmë modulet RAM nga bazamenti i tyre, të kontrollojmë pinat dhe përsërit t'i kthejmë në vendet e tyre.

Krahas sinjaleve zanore, ekzistojnë edhe sinjale drite. Dioda LED e gjelbër dhe e kuqe në maskën e përparmë të shtëpizës, të cilat në fakt janë indikator të furnizimit, mund të ndiçojnë me dinamika të ndryshme.

Kompjuterët desktop janë shumë më të thjeshtë për t'u mirëmbajtur, për diagnostifikim dhe zëvendësim të komponentëve harduerikë të dëmtuara. Për të zgjatur jetëgjatësinë kompjuterit, shumë e rëndësishme është që përdoruesi të kujdeset siç duhet për të. Kompjuteri duhet të mbrohet nga lagështia dhe papastërtia sepse ato ndikojnë dëmshëm në kontaktet e pllakës amë dhe reduktojnë ftohjen dhe ajrosjen, kështu që mund të vij deri te mbinxehja. Gjithashtu, përdoruesi duhet të shkyç kompjuterin siç duhet për të parandaluar dëmtimin e mundshëm të pjesëve të tij.

3. Ushtrime praktike për instalimin e sistemit operativ dhe programeve të tjera standarde në një sistem kompjuterik personal

3.1. Përgatitja për instalimin e softuerit

Sot instalimi i softuerit aplikativ dhe softuerit sistemor është shumë i thjeshtë, kryhet në disa hapa dhe çdo hap shpjegohet në detaje gjatë vetë procedurës. Megjithatë, për një instalim të suksesshëm, përdoruesi duhet t'u përmbahet rregullave të mëposhtme.

- Duhet të jemi të sigurt se programi që po instalojmë do të jetë i dobishëm për ne. Çdo program i instaluar, pavarësisht nëse e përdorim apo jo, merr memorie dhe e ngadalëson punën e kompjuterit.
- Për shkarkimin e softuerit për instalim duhet të përdorim linqe të besueshme dhe të verifikuara.
- Duhet të kontrollojmë nëse konfigurimi i kompjuterit plotëson kërkesat për funksionimin korrekt të softuerit.
- Duhet të kontrollojmë nëse kemi hapësirë të mjaftueshme të lirë në memorien e përhershme.
- Duhet të kontrollojmë nëse ka opsion për ta deinstaluar.
- Rekomandohet që përdoruesi të zgjedhë një instalim sipas dëshirës së tij (angl. Custom) sepse në këtë mënyrë ai përzgjedh mundësitë që i ofrohen.
- Duhet të sigurohet një furnizim i qëndrueshëm dhe qasje e sigurt në internet në mënyrë që të mos ketë ndërprerje gjatë instalimit.
- Ushtrimet praktike të instalimit të softuerit të mos kryhen pa miratimin e mësuesit të lëndës.

3.2. Ushtrim praktik: Instalimi i sistemit operativ Windows 10

1. Qëllimi i ushtrimit

Sot instalimi i sistemit operativ është procedurë e thjeshtë. Pas fillimit të procesit të instalimit, hapen dritare me instruksione të cilat duhet të lexohen dhe ekzekutohen me kujdes. Gjatë kohës së këtij ushtrimi praktik nxënësit do të njihen

me tri procedura: krijimi i një moduli instalimi, thirrja e tij në programin BIOS dhe instalimi i sistemit operativ Windows 10.

2. Koha e realizimit: 2 orë mësimore

3. Përgatitje për instalimin e sistemit operativ Windows 10.

Së pari duhet të zgjidhet lloji i sistemit operativ Windows 10. Windows 10 Home është për përdorim në shtëpi, Windows 10 Pro është për profesionistët, Windows 10 Enterprise është për kompanitë dhe Windows 10 Education është për studentët dhe nxënësit. Të gjithë llojet e sistemeve operative janë në dispozicion në versionet **32 dhe 64 bitësh**. Të theksojmë, versioni 64-bitësh nuk mund të instalohet në një kompjuter me procesor 32-bitësh. Të dhënat përpunohen më shpejtë me procesorët 64-bitësh, të cilët mundësojnë përdorimin më efikas të memories RAM mbi 4 GB.

Sistemi operativ Windows 10 mund të funksionojë në të njëjtin konfiguracion kompjuterik si edhe Windows 7: procesor me frekuencë pune mbi 1 GHz, RAM memorie prej 2GB, një hard disk me kapacitet prej 16 GB për versionin 32-bitësh dhe 20 GB për versionin 64-bitësh, kartelë grafike me WDDM (angl. Windows Display Driver Model), ekran me rezolucion 800 × 600 piksela.

Nëse kompjuteri posedon me një version më të vjetër të licencuar të Windows-it, ai mund të përditësohet në Windows 10 dhe vetë përditësimi është falas. Nëse bëhet fjalë për kompjuter të ri, atëherë sistemi operativ Windows 10 patjetër të instalohet. Për këtë qëllim është i nevojshëm modul memorie instalimi (USB ose DVD).

Për aktivizimin e Windows 10 kërkohet çelës i produktit për aktivizim. Çelësi i produktit është kod 25-shifror për të verifikuar licencën. Një ueb faqen e Microsoftit mund të gjendet lista zyrtare e shitësve të Windows 10 të cilët e dorëzojnë kodin së bashku me sistemin operativ, ose i njëjti mund të dërgohet nëpërmjet emailit, nëse programi i instalimit shkarkohet nga Interneti. Një mënyrë tjetër për të marrë licencë digjitale është përmes llogarisë suaj të përdoruesit në Microsoft. Për këtë qëllim, zgjedhim Settings → Update and Security → Activation. E shtypim butonin Shto një llogari (angl. Add an account) dhe qasemi me emrin e përdoruesit dhe fjalëkalimin tonë. Procedurën e instalimit të Windows 10 do ta ndajmë në tri pjesë: krijimin e një moduli memorie USB instaluese, thirrjen e tij përmes programit të menaxhimit të BIOS-it dhe në fund instalimin e vetë sistemit operativ. [11]

4. Krijimi i një moduli memorie USB instaluese

- (1) Modulën e memojes USB e lidhim me kompjuterin. Ai duhet të ketë një kapacitet minimal prej 8 GB.
- (2) <https://www.microsoft.com/en-us/software-download/windows10%20> është linku për të shkarkuar veqlën për krijimin e modulit instalues (angl. Media

Creation Tool). Klikohet butoni Download tool now (Shkarkoje veglën tani) dhe brenda pak sekondash mjeti do të shfaqet në dosjen e shkarkimeve (angl. Download).

- (3) Me klikim të dyfishtë mbi veglën fillon krijimi i modulit instalues. I pranojmë kushtet e theksuar (angl. Accept)
- (4) Në dritaren e ardhshme e zgjedhim opsionin për krijimin e modulit instalues.

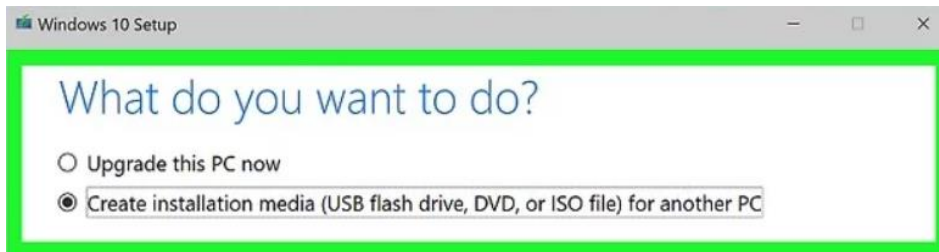


Figura 3.1. Fillimi i krijimit të modulit instalues për Windows-in

- (5) Zgjedhim gjuhën, sistemin operativ dhe arkitekturën
- (6) Zgjedhim llojin e modulit dhe klikojmë I ardhshëm (angl. Next)

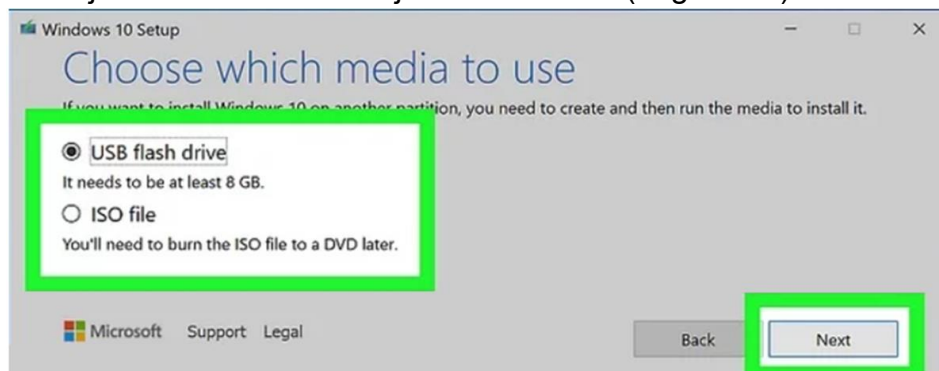


Figura 3.2. Zgjedhja e modulit memorues, USB ose DVD

- (7) Nëse ka më tepër module memorie USB, zgjedhim njërin.

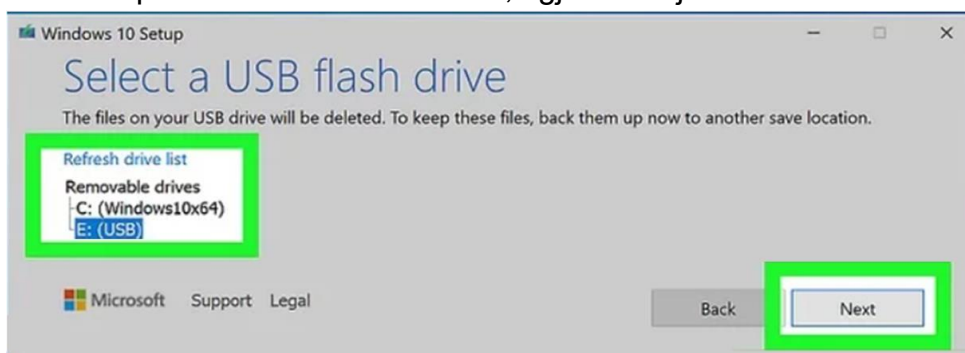


Figura 3.3. Zgjedhja e modulit USB për ruajtjen e instalimit

- (8) Presim që të instalohen dokumentet dhe në fund shtypim Përfundo (angl. Finish).

Komenti: _____

5. Thirrja e modulit memorues USB instalues

- (1) Pas lidhjes së modulit dhe kyçjes së kompjuterit, hapim programin BIOS, ku ndodhet menyuja për zgjedhje e pajisjes që përmban instalimin. Për këtë qëllim, menjëherë pas kyçjes së kompjuterit duhet të shtypim tastin **F1**, **F2**, **F10**, **F12** ose **Del**. Ky buton është i ndryshëm për prodhues të ndryshëm të kompjuterëve.
- (2) Nga menyuja e sipërme, zgjedhim opsionin Boot dhe më pas me ndihmën e tastierës zhvendosemi lart-poshtë, e zgjidhim pajisjen dhe shtypim Enter.

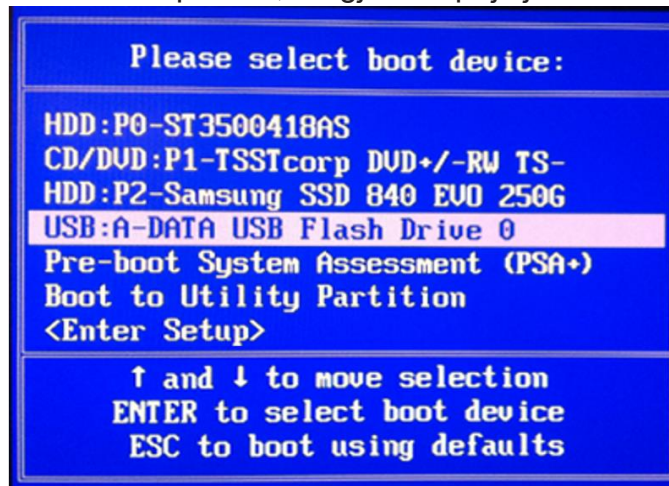


Figura 3.4. Zgjedhja e pajisjes për instalim

- (3) Pas shtypjes së Enter, hapet dritarja për instalim (angl. Windows Setup Wizard).

Komenti: _____

6. Instalimi i sistemit operativ Windows 10



Figura 3.5. Zgjedhja e gjuhës standarde, kohës dhe gjuhës për futjen e të dhënave

Me dritaren e parë zgjidhet gjuha, koha dhe formati i datës, gjuha e tastierës për futjen e të dhënave (USA ose UK). Shtypim Next dhe hapet dritarja e dytë.

- (2) Duke zgjedhur opsionin "Instalo Tani" (Install now), fillon procesi i instalimit.
- (3) Në fillim të instalimit, hapet një dritare që i kërkon përdoruesit të vendosë të ashtuquajturin çelës produkti. Kemi të bëjmë me një kod 25-shifror me të cilin konfirmohet licenca. Këtë kod e jep Microsoft, së bashku me dorëzimin e sistemit operativ, ose mund të dërgohet me email, nëse programi i instalimit është shkarkuar nga Interneti.
- (4) Dritarja e tretë është për pranimin e kushteve që lidhen me licencën e sistemit operativ.

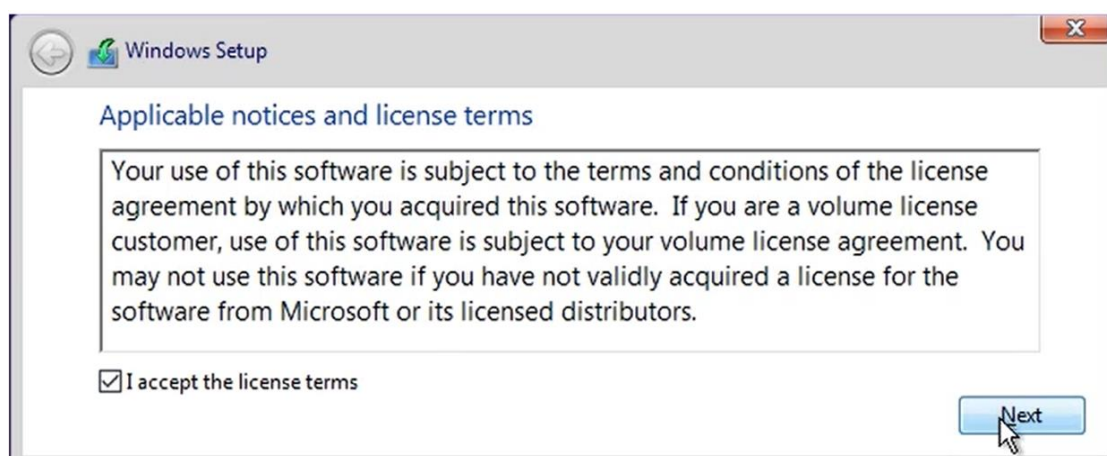


Figura 3.6. Pranimi i kushteve për licencën

Kushtet i pranojmë me përzgjedhjen e katrorit nga ana e majtë dhe shtypim Next.

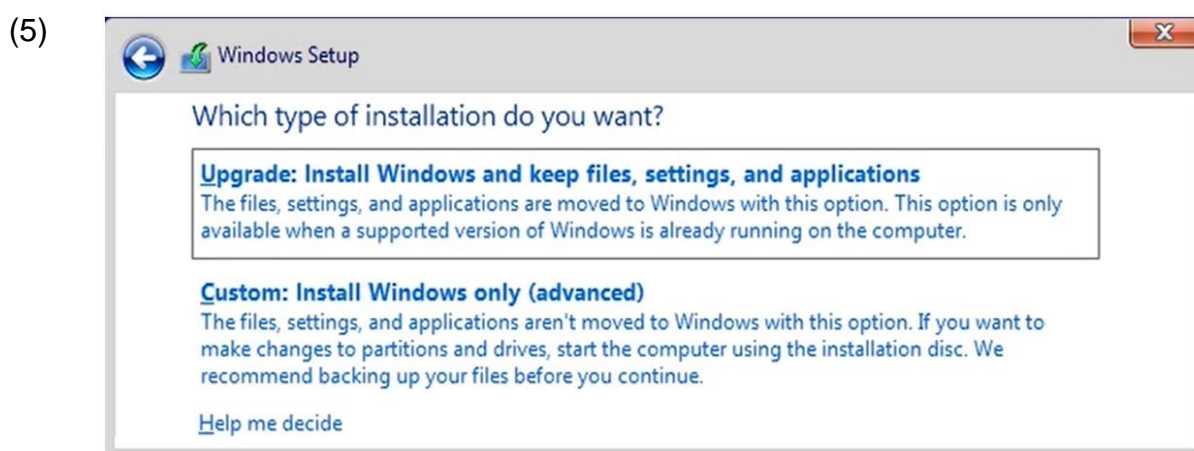


Figura 3.7. Zgjedhja mes përditësimit ose një instalimi të ri të Windowsit. Dritarja e katërt është për zgjedhjen e llojit të instalimit. Ka dy opsione: Upgrade ose Custom. Opsioni i parë do të thotë përditësim i sistemit operativ ekzistues Windows. Ky instalim është falas, nuk kërkon çelës produkti dhe nuk fshihen të dhënat e përdoruesit. Me opsionin e dytë fshihen të gjitha të dhënat e ruajtura në kompjuter dhe kjo nënkupton një instalim i ri.

- (6) Dritarja e pestë është për zgjedhjen e diskut në të cilin do të instalojmë sistemin operativ.

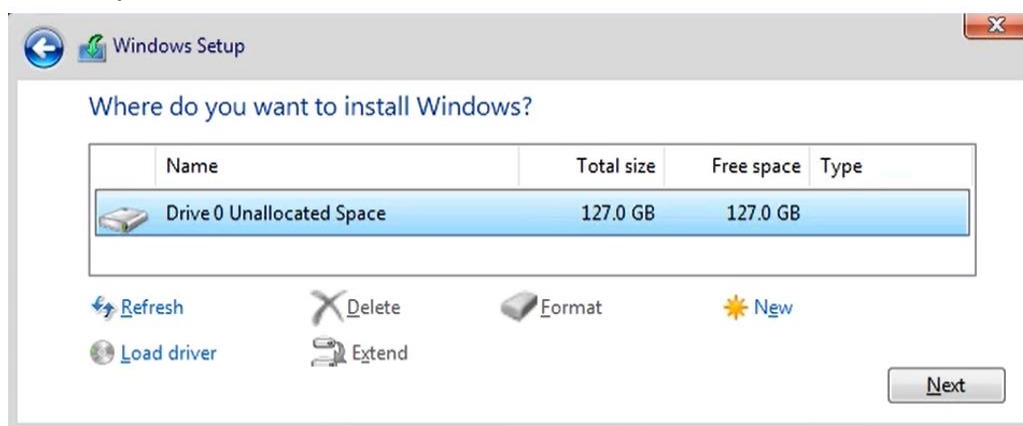


Figura 3.8. Zgjedhja e partitionit të hard diskut

Duke shtypur butonit New mund të krijohet një particion i në disk, me ç'rast duhet të përcaktohet edhe kapaciteti i particionit të ri. Në particionin e parë vendoset sistemi operativ, ndërsa në të dytin ruhen të dhënat e përdoruesit. Të theksojmë se me krijimin e particionit të dytë, me automatizëm formohet edhe një particion tjetër sistemor. Të rikujtojmë se particionet mund të jenë të llojeve të ndryshme, MBR ose GPT, ndërsa të njëjtat u shpjeguan në 2.2. Ushtrim praktik për instalimin e pjesëve përbërëse të kompjuterit personal, në pjesën nr. 10 kur folëm për inicializimin e hard disqeve. Është e logjikshme që nëse kompjuteri posedon program UEFI, ndarjet duhet të jenë GPT. Nëse kompjuteri ka një program BIOS, atëherë të mundshme janë vetëm particionet MBR dhe nëse hard disku ka një kapacitet mbi 2TB, atëherë gjithçka që është mbi 2TB nuk do të shfrytëzohet.

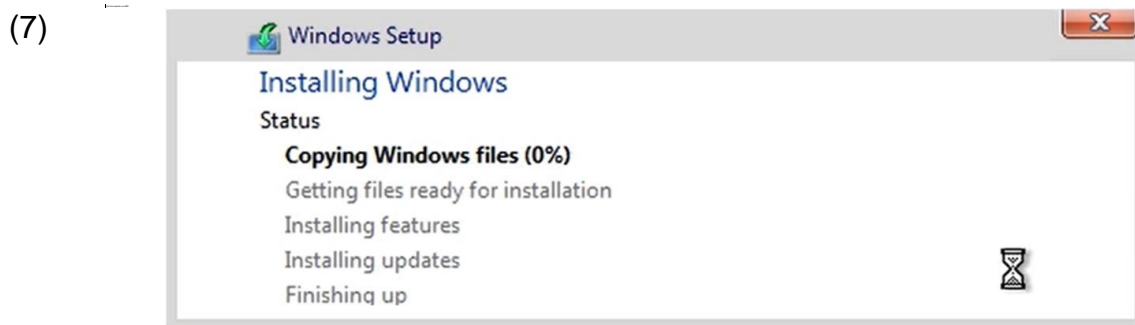


Figura 3.9. Fillimi i instalimit të Windowsit

Fillon instalimi i sistemit operativ dhe pasi e njëjta të përfundojë, kompjuteri do të ristartohet.

- (8) Pastaj pasojnë disa dritare me të cilat përshtaten më tepër gjëra: shteti ose rajoni dhe gjuha e tastierës për futjen e të dhënave.
- (9) Dritarja për lidhjen me rrjetin mund të anashkalohe dhe këtë përzgjedhje do ta bëjmë pas instalimit të Windows 10.
- (10) Me dy dritaret e ardhshme përcaktohen emri i përdoruesit, fjalëkalimin dhe sugjerimi (angl. hint) i cili duhet të na ndihmojë të mbajmë mend fjalëkalimin. Pa fjalëkalim, nuk do të mund të hapim sipërfaqen punuese, desktopin e sistemit operativ. Në instalime të caktuara, përveç emrit të përdoruesit, duhet të specifikohet edhe profili në Microsoft dhe nëse përdoruesi nuk ka një profil të tillë, ai mund ta krijojë. Në sistemin operativ Windows 10 ka opsion që të dhënat dhe skedarët e përdoruesit të ruhen me automatizëm në cloud (sngl. OneDrive). Në këtë mënyrë, përdoruesi do të ketë qasje në të dhënat e tij edhe nëpërmjet telefonit celular.
- (11) Dritarja e parafundit është për zgjedhjen e aplikacionit Kortana, si asistent personal.

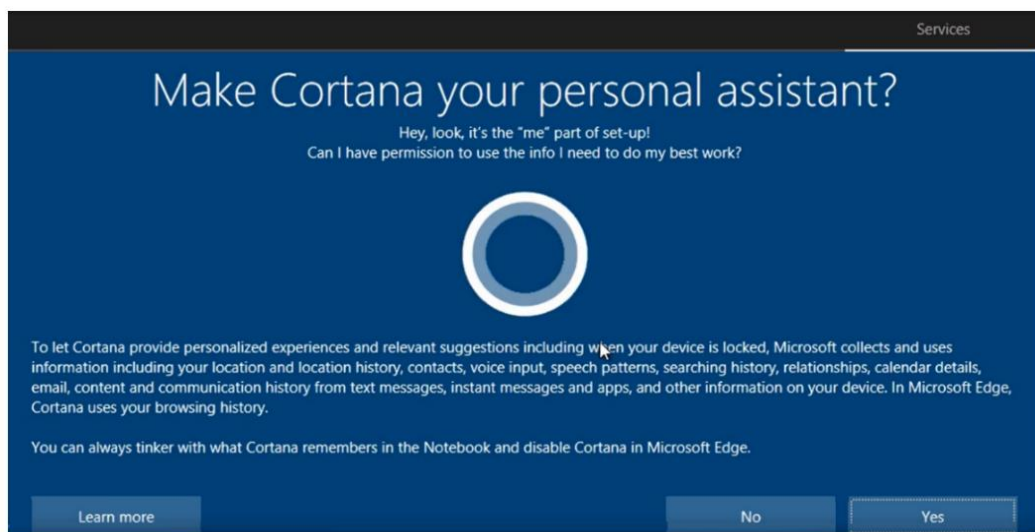


Figura 3.10. Zgjedhja e aplikacionit për asistent personal

Ky aplikacion është risi në Windows 10 dhe përdoret për kërkim në internet dhe për menaxhim me komponentët harduerike, por nuk është i disponueshëm në të gjitha shtet.

- (12) Hapi i fundit i instalimit është konfigurimi i attributeve private, si përcaktimi i vendndodhjes, diagnostifikimi i softuerit dhe aplikacioneve, njohja e të folurit dhe të ngjashme.

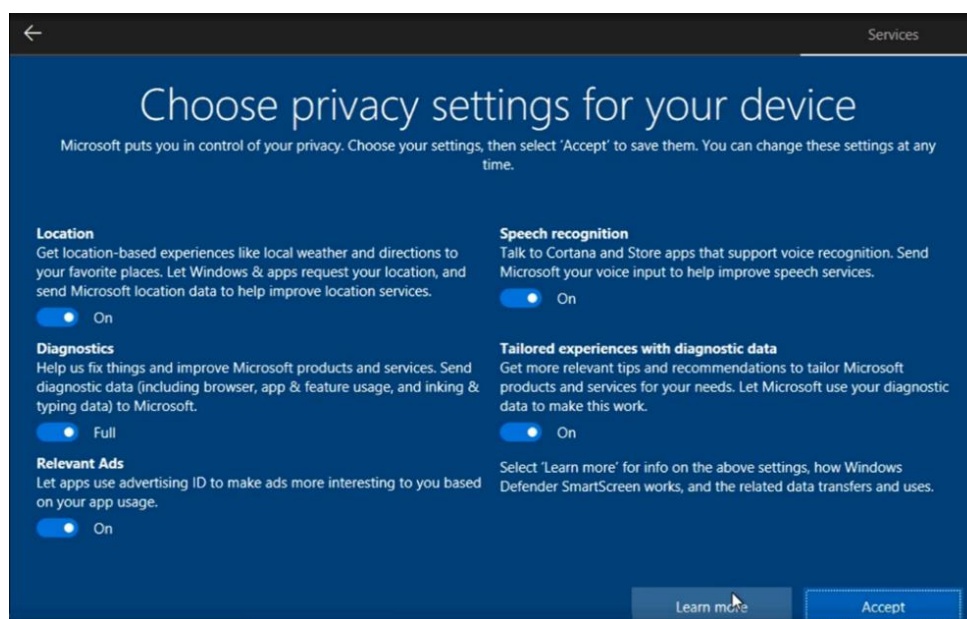


Figura 3.11. Vendosija e mundësisë së përcaktimit të vendndodhjes dhe diagnostifikimit

Instalimi ka përfunduar dhe duhet të presim disa minuta që të hapet sipërfaqja punuese e Windows 10.

Komenti: _____

3.3. Ushtrim praktik: Konfigurimi i sistemit operativ Windows 10

Qëllimi i ushtrimit: Me kalimin e kohës, performanca e sistemit operativ zvogëlohet për një sërë arsyesh: jokompatibiliteti, viruset, aplikacionet që nuk përdoren etj. **Qëllimi i optimizimit është rritja e shpejtësisë së sistemit operativ.** Koha e nevojshme për realizimin e ushtrimit është orë mësimore. Më poshtë janë dhënë disa mënyra për arritjen e këtij qëllimi, rëndësinë e tyre dhe hapat për realizim.

- | | | |
|--|---|---|
| <p>(1) Aplikacionet startuese (angl. Startup) →</p> | <p>Startup aplikacionet aktivizohen automatikisht gjatë kyçjes së kompjuterit dhe e risin kohën e pritjes. Për këtë arsye është e nevojshme të reduktohet numri i tyre.</p> | <ul style="list-style-type: none"> • Setting • Apps • Startup • Sort by: Startup impact • Zgjedhja e aplikacioneve • Ristartimi i kompjuterit |
|--|---|---|

- | | | |
|---|---|---|
| <p>(2) Aplikacionet e papërdorura</p> | <p>→ Sistemet operative Windows nuk ofrojnë njoftime (notification) për ngarkesën e sistemit, prandaj vetë përdoruesit duhet të largojnë programet që nuk i përdorin.</p> | <ul style="list-style-type: none"> • Settings • Apps • Apps & feature • Zgjedhja e aplikacionit • Uninstall |
| <p>(3) Lirimi i memories së hard diskut</p> | <p>→ Zakonisht pasi të përdoret 70% e hapësirës memoruese të hard diskut. Në Panelin e kontrollit (Control Panel) kjo mundësi është nën kategori e veglave administrative- sistemore (System-Administrative Tools).</p> | <ul style="list-style-type: none"> • Settings • System • Storage • Temporary files • Zgjedhja e skedarëve • Removes files |
| <p>(4) Organizimi i të dhënave në hard disk (angl. defragmentations)</p> | <p>→ Kur në hard disk do të zvogëlohet hapësira e lirë, atëherë të dhënat regjistrohen në vende të ndryshme në hard disk, me ç'rast rritet koha e leximit. Me defragmentim, këto të dhëna vendosen pranë njëra-tjetrës dhe lirohet hapësira e memories.</p> | <ul style="list-style-type: none"> • Settings • System • Storage • More settings • Optimize Drivers • Zgjedhim diskun (C: ose D:) • Optimize |
| <p>(5) Programet antivirus</p> | <p>→ Nëse sistemi operativ funksionon me një shpejtësi të reduktuar, është e nevojshme të thirret dhe të aktivizohet programi antivirus. Nëse kompjuteri nuk ka një program të veçantë antivirus, mund të përdoret programi Windows Defender.</p> | <ul style="list-style-type: none"> • Control Panel • Security • Windows Defender • Scan |
| <p>(6) Përditësimi i sistemit operativ</p> | <p>→ Është e gabuar kur përdoruesit çaktivizojnë përditësimet automatike sepse përditësimet janë shumë të rëndësishme për sigurinë e sistemit.</p> | <ul style="list-style-type: none"> • Settings • Update & Security • Windows Update • Check for Update |

	Drajverët e pajisjeve mund të përditësohen në kategorinë System-Device manager.	<ul style="list-style-type: none">• Ristartimi i kompjuterit
(7) Zmadhimi i faqeve virtuale	→ Memoria virtuale nënkupton zgjerimin e memories parësore përmes memories sekondare. Kështu, memoria RAM merr të dhënat nga hard disku në formën e faqeve virtuale me madhësi fikse dhe i fut ato në një kornizë virtuale. Nëse rritet kapaciteti i faqeve virtuale, do të reduktohet numri i transferimeve ndërmjet RAM-it dhe hard diskut.	<ul style="list-style-type: none">• Control Panel• System• Advanced System (Performance)• Settings• Advanced• Change (Virtual memory)

Sistemi operativ Windows 10 ofron shumë mundësi dhe ne nuk mund t'i shpjegojmë dhe eksplorojmë të gjitha. Por, gjithmonë duhet të hulumtohet dhe të përpiqemi të konfigurujmë në mënyrë të pavarur sistemet kompjuterike.

Komenti: _____

3.4. Riinstalimi i sistemit operativ Windows 10

1. Qëllimi i ushtrimit:

Për shkak të mbrojtjes së pamjaftueshme nga viruset, mbingarkesës me shumë aplikacione ose përdorimit të tepruar të kapacitetit të hard diskut, kompjuteri ngadalë e humbet shpejtësinë e tij të funksionimit. Një nga zgjidhjet më të thjeshta është riinstalimi i sistemit operativ. Ekzistojnë më tepër opsione për riinstalim dhe qëllimi i këtij ushtrimi është të njihemi me to.

2. Koha e realizimit: 1 orë mësimore

3. Riinstalimi me humbje të të dhënave të përdoruesit (dokumenteve, fotografive dhe videove) dhe aplikacioneve të instaluara

Nuk ekziston asnjë dallim midis procedurës për këtë lloj të riinstalimit dhe instalimit. Pasi të krijohet një modul memorie instalimi, të njëjtën mund ta përdorim në

një numër të pakufizuar për të riinstaluar sistemin operativ Windows 10. Për të mos ardhur deri te humbja e të dhënave, është më mirë të bëjmë kopje rezervë të dokumenteve të përdoruesit përpara se të fillojmë me riinstalimin. Ne mund t'i ruajmë ato në memorie të jashtme ose të krijojmë një particion të veçantë në hard disk përmes funksionit të Disk Management. Gjatë kohës së riinstalimit, kjo pjesë (particion) i hard diskut nuk do të formatohet.

4. Risetimi i sistemit operativ Windows 10 në mënyrë që të dhënat e përdoruesit do të ruhen, por aplikacionet e instaluara do të fshihen.

Përparësia e kësaj procedure është se nuk kërkohet asnjë modul memorie USB dhe e gjithë procedura kryhet automatikisht. Risetimi i sistemit operativ kërkon hapësirë memoruese të lirë në hard disk prej 3-4GB. Nëse nuk ka kaq hapësirë të lirë, do të marrim një njoftim

Për risetimin e sistemin operativ Windows 10, zgjedhim opsionet Settings → Update&Security → Recovery → Reset this PC (Get Started). Paraqitet dritare për zgjedhjen e opsionit, nëse dokumentet e përdoruesit do të ruhen ose gjithçka do të fshihet (figura).

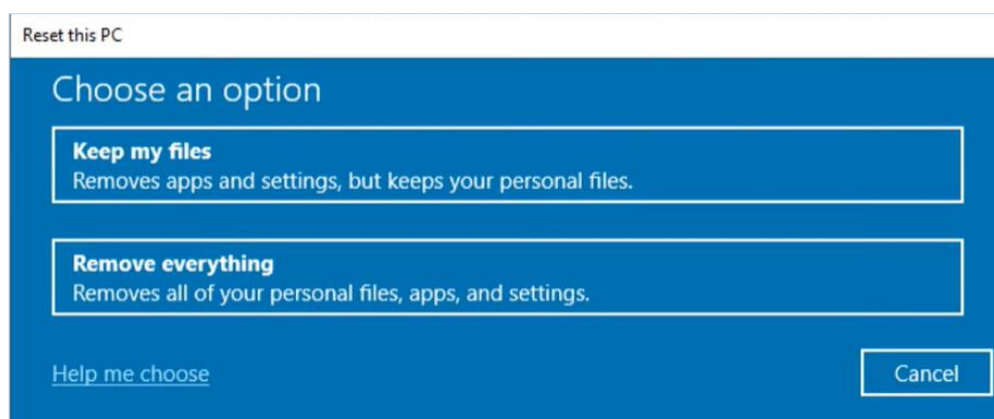


Figura 3.12. Gjatë risetimit të Windows 10 mund të rruhen dokumentet e përdoruesit

Pas një përgatitjeje të shkurtër, hapet një dritare për konfirmimin e risetimit. Në këtë dritare, ekziston një opsion për të listuar aplikacionet që do të fshihen. Pas kësaj fillon risetimi.

5. Riinstalimi i sistemit operativ Windows 10 nëpërmjet një moduli memorie USB instalimi në mënyrë që të dhënat e përdoruesit dhe aplikacionet e instaluara të ruhen.

Në ushtrimin praktik 3.2. Instalimi i sistemit operativ Windows 10 u njohëm me krijimin e modulit të memories USB instaluese. Për të kryer këtë ushtrim, është e nevojshme që modulën instalues ta lidhim me kompjuter. E selektojmë modulën, shtypim në klikun e djathtë dhe zgjedhim opsionin **Mount** dhe në këtë mënyrë përmbajtja e modulit të memories do të bëhet e dukshme. Në dosjen.

This PC, selektojmë dokumentin e ri, e shtypim klikun e djathtë dhe e zgjedhim opsionin **Open in new window**. Hapet një dritare e re dhe zgjedhim opsionin **Setup.exe**. Fillon përgatitja për risetimin e sistemit operativ. Në dritaren "Get importante updates", zgjedhim opsionin **Not right now**, me qëllim që të thjeshtohet procesi i riinstalimit. I pranojmë kushtet për licencën.

Pas disa minutash pritje, shfaqet dritarja **"Ready to install"** që në përkthim do të thotë "Gati për instalim". Duhet t'i kushtojmë vëmendje më të madhe kësaj dritareje. Opsionet **Install Windows 10 Home/Pro dhe Keep personal files and apps** duhet të jetë të selektuar. Nëse nuk e zgjedhim opsionin Keep personal files and apps, të njëjtën mund ta marrim përmes opsionit **Change what to keep**.

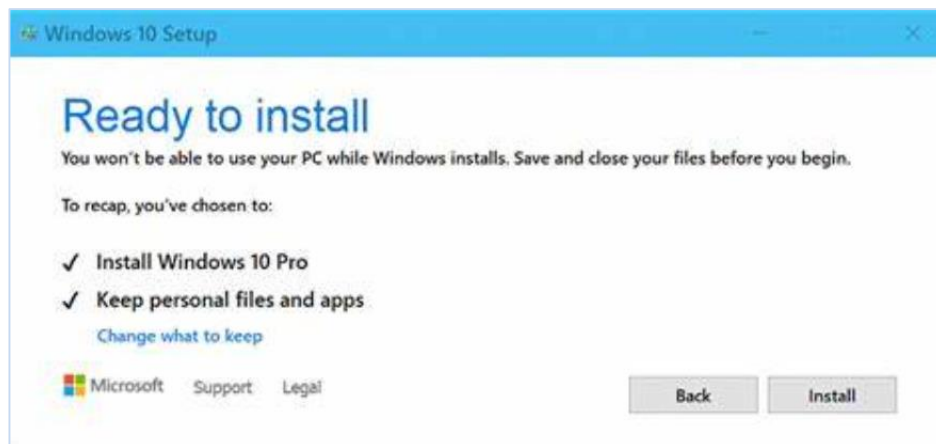


Figura 3.13. Mbrojtja e dokumenteve dhe aplikacioneve personale nga fshirja

Në këtë mënyrë kryejmë mbrojtjen e dokumenteve dhe aplikacioneve personale. Me shtypjen e butonit Install fillon riinstalimi i sistemit operativ Windows 10.

Komenti: _____

3.5. Ushtrim praktik: Instalimi i sistemit operativ Ubuntu 17 Linux në makinë virtuale

1. Qëllimi i ushtrimit:

Në pjesën teorike të njësisë modulare Instalimi i një sistemi operativ dhe programeve të tjera standarde në një sistem kompjuterik personal, tashmë jemi njohur me shumë përparësi dhe mundësi që i ofron sistemi operativ Linux. Qëllimi i këtij ushtrimi është përdorimi i dy sistemeve operative në të njëjtin kompjuter, duke përdorur një makinë virtuale për të instaluar sistemin operativ Linux. **Ubuntu është shpërndarës shumë i popullarizuar** i sistemeve operative Linux. Për të shkarkuar

instalimin e sistemit operativ, është e nevojshme të hapet ueb faqja www.ubuntu.com e këtij shpërndarësi dhe më pas të klikohet në Download, ndërsa zgjedhim versionin desktop.

2. Koha e realizimit: 2 orë mësimore

3. Udhëzime për krijimin e makinës virtuale

VirtualBox është aplikacion për krijimin, konfigurimin dhe administrimin e makinave virtuale. Makina virtuale është mysafir (guest) i sistemit kompjuterik që përdor resurset e kompjuterit nikoqir (host). Në kompjuterin nikoqir do të instalohet sistemi operativ Windows, ndërsa në makinën virtuale, gjegjësisht kompjuteri mysafir do të instalohet sistemi operativ Ubuntu. Në këtë mënyrë, **në kompjuterin e njëjtë** do të kemi të instaluar **dy sisteme operative** pa pasur ndikim njëri në tjetrin.

Programi për instalimin e makinës virtuale shkarkohet nga faqja e internetit www.virtualbox.org, duke klikuar opsionin Download dhe duke zgjedhur linkun Windows host. Pas instalimit të aplikacionit VirtualBox, e hapim dhe klikojmë në opsionin New për të krijuar një makinë të re virtuale.

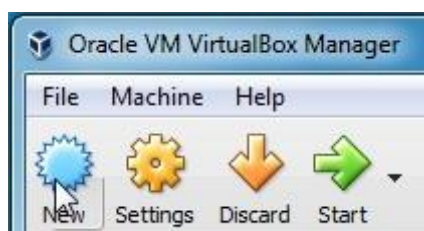


Figura 3.14. Krijimi i makinës virtuale

Hapet dritarja **Create Virtual Machine**. Në fushën **Name**, vendoset emrin e makinës virtuale (për shembull Ubuntu), në fushën **Type** futet lloji i sistemit operativ që do të instalohet në makinën virtuale (Linux) dhe në fushën **Version**, futet versioni i sistemit operativ (32-bitësh ose 64-bitësh). Në fundi klikohet Next.

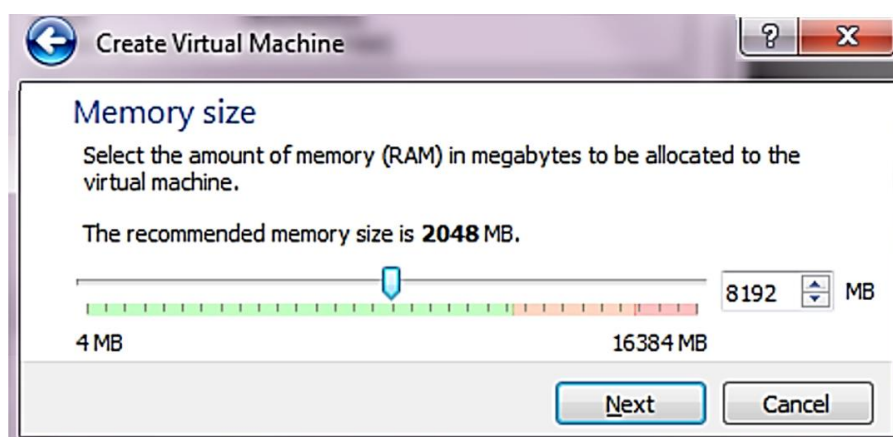


Figura 3.15. Përcaktimi i kapacitetit të memories RAM për makinën virtuale

Dritarja e ardhshme shërben për përcaktimin e kapacitetit të memories RAM për makinën virtuale (figura). Është mirë të ndiqen rekomandimet, por duhet të merret parasysh edhe kapaciteti total i memories.

Tri dritaret e ardhshme përdoren për të krijuar **hard diskun virtual**, definimin e kapacitetit të tij dhe përcaktimin e mënyrës së menaxhimit me hapësirën memoruese.

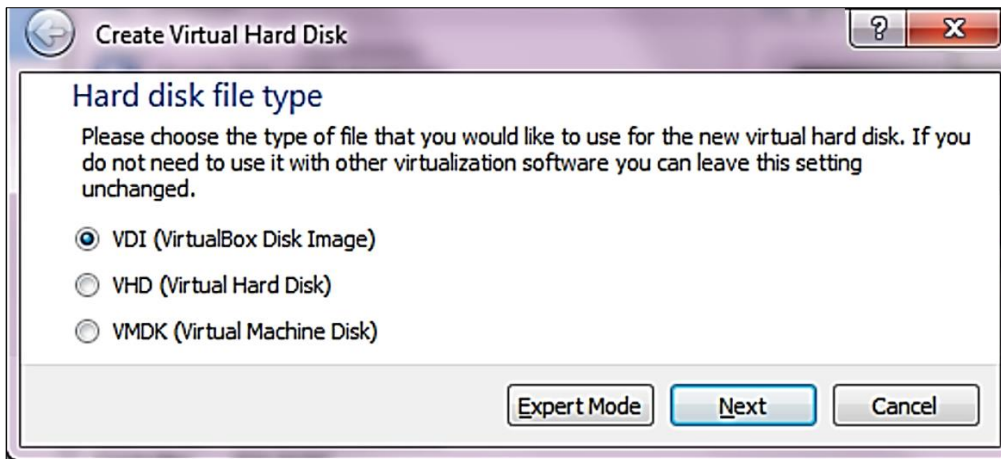


Figura 3.16. Zgjedhja e llojit të hard diskut



Figura 3.17. Zgjedhja e kapacitetit të hard diskut

Krijimin e makinës virtuale e konfirmojmë duke klikuar Create. Në këtë hap të fundit kemi mundësinë të ndryshojmë emrin dhe lokacionin e makinës virtuale.

Komenti: _____

4. Instalimi i sistemit operativ Ubuntu

Për të instaluar sistemin operativ Ubuntu, klikojmë opsionin Start në dritaren VirtualBox Manager, të paraqitur në figurën 3.18.



Figura 3.18. Fillimi i instalimit të sistemit operativ Ubuntu

Në dritaren Select start-up disk, duke klikuar në dosjen (folderin) e verdhë, gjejmë lokacionin ku ndodhet instalimi i sistemit operativ, e zgjedhim atë dhe klikojmë në butonin Start.

Në dritaren e Welcome, janë dhënë **dy mundësi për instalimin** e sistemit operativ Ubuntu. Me opsionin Try Ubuntu, sistemi operativ mund të përdoret pa instalim. Me këtë mënyrë, në fakt, e instalojmë sistemin operativ në hard diskun virtual i cili është krijuar paraprakisht dhe para instalimit është bosh, kështu që nuk ka fshirje të sistemit operativ ekzistues.

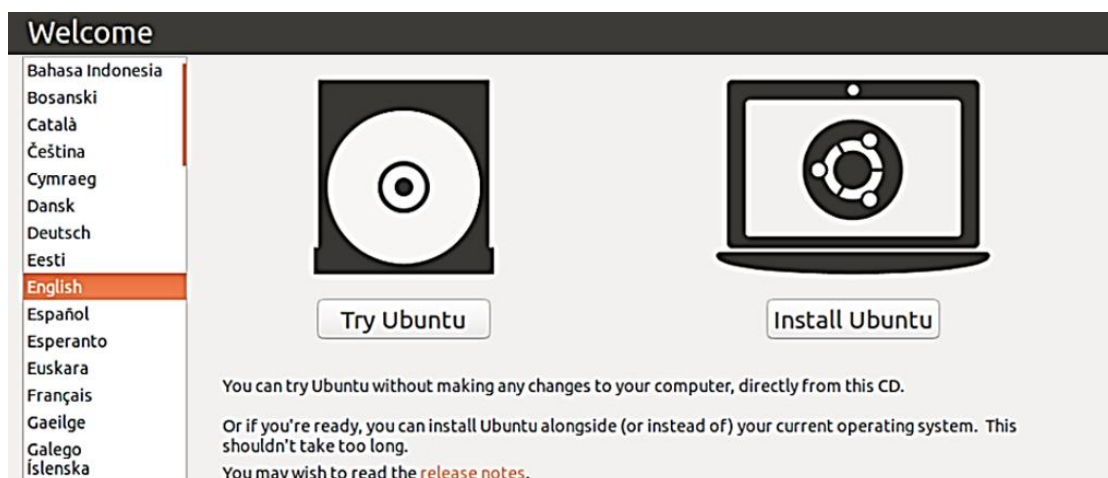


Figura 3.19. Zgjedhja e metodës së instalimit të sistemit operativ Ubuntu

Në dritaren **Installation type**, duke klikuar në butonin **Install Now** fillon instalimi. Opsioni **Erase disk and install Ubuntu** është i selektuar paraprakisht, por tashmë kemi shpjeguar se pasi që bëhet fjalë për hard disk virtual, nuk do të vij deri te fshirja ose humbja e të dhënave.

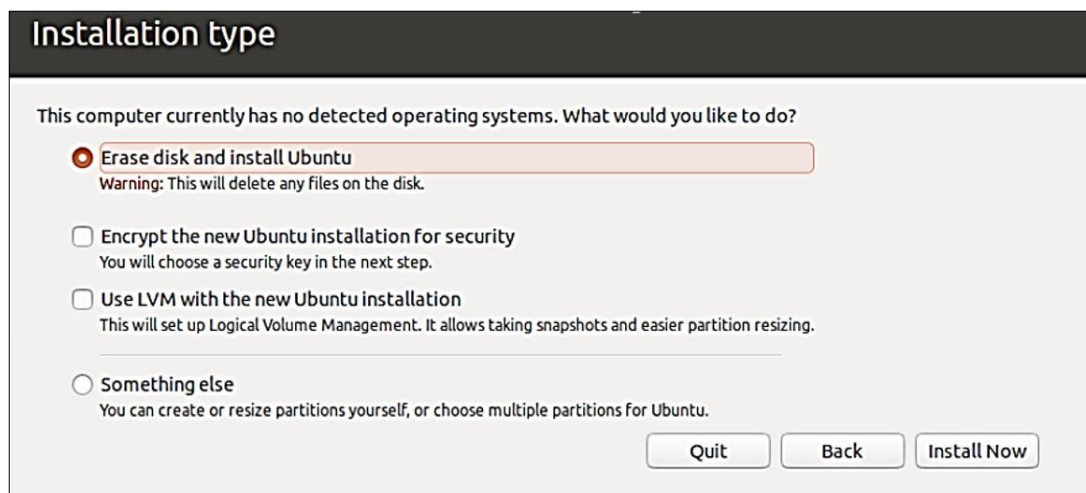


Figura 3.20. Për shkak të makinës virtuale, sistemi operativ ekzistues nuk do të fshihet.

Dritarja e ardhshme është **Preparing to install Ubuntu** dhe në të ekziston mundësia e zgjedhjes së përditësimit automatik të sistemit operativ dhe instalimin e softuerit shtesë për multimedia. Në fund klikojmë në butonin **Continue**.

Dy dritaret e ardhshme janë për zgjedhjen e shtetit ose rajonit dhe gjuhën e tastierës për futjen e të dhënave, ngjashëm si me rastin e instalimit të sistemit operativ Windows 10.

Pason dritarja për përcaktimin e emrit të përdoruesit dhe fjalëkalimit. Duke klikuar në **Continue**, fillon instalimi.

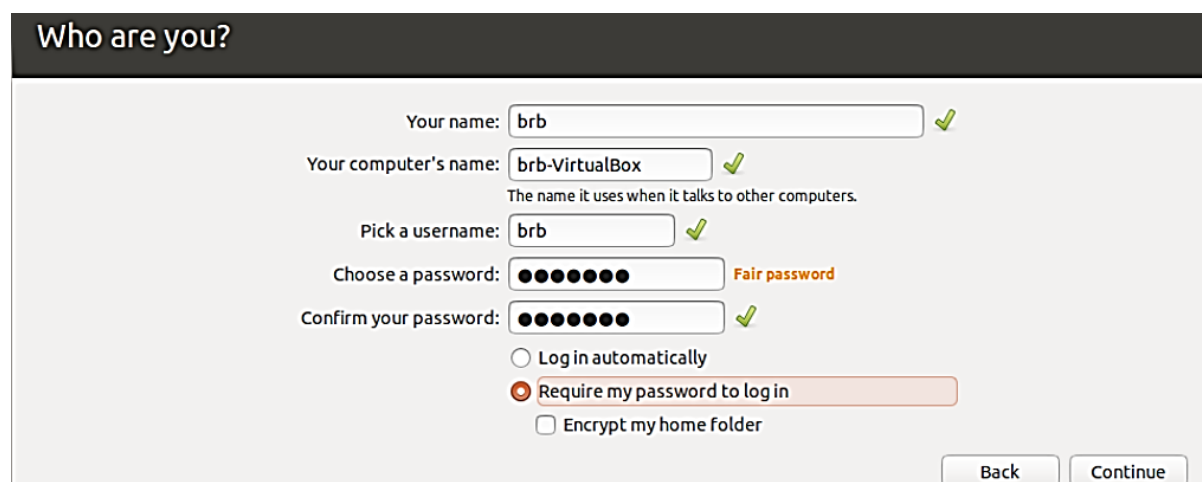


Figura 3.21. Zgjedhja e emrit të përdoruesit dhe fjalëkalimi

Pas informacionit për instalimin e suksesshëm, e ristartojmë sistemin dhe e fitojmë sipërfaqen punuese të Ubuntu.

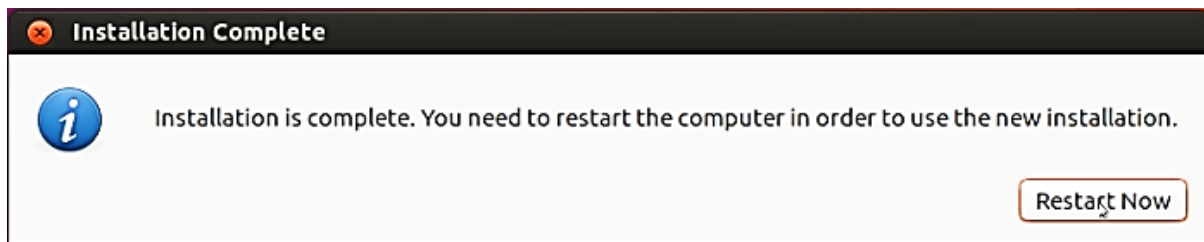


Figura 3.22. Ristartimi i kompjuterit për të fituar hapësirën punuese të Ubuntu

Pas përfundimit të instalimit, është e nevojshme të fshihet skedari instalues. Për këtë qëllim zgjedhim sistemin operativ Ubuntu në makinën virtuale, klikojmë në opsionin Storage në kategorinë Settings. Më pas, në pjesën Storage Tree duhet të zgjidhet dokumenti instalues dhe klikohet në Remove selected storage attachment. Në fund shtypim butonin OK.

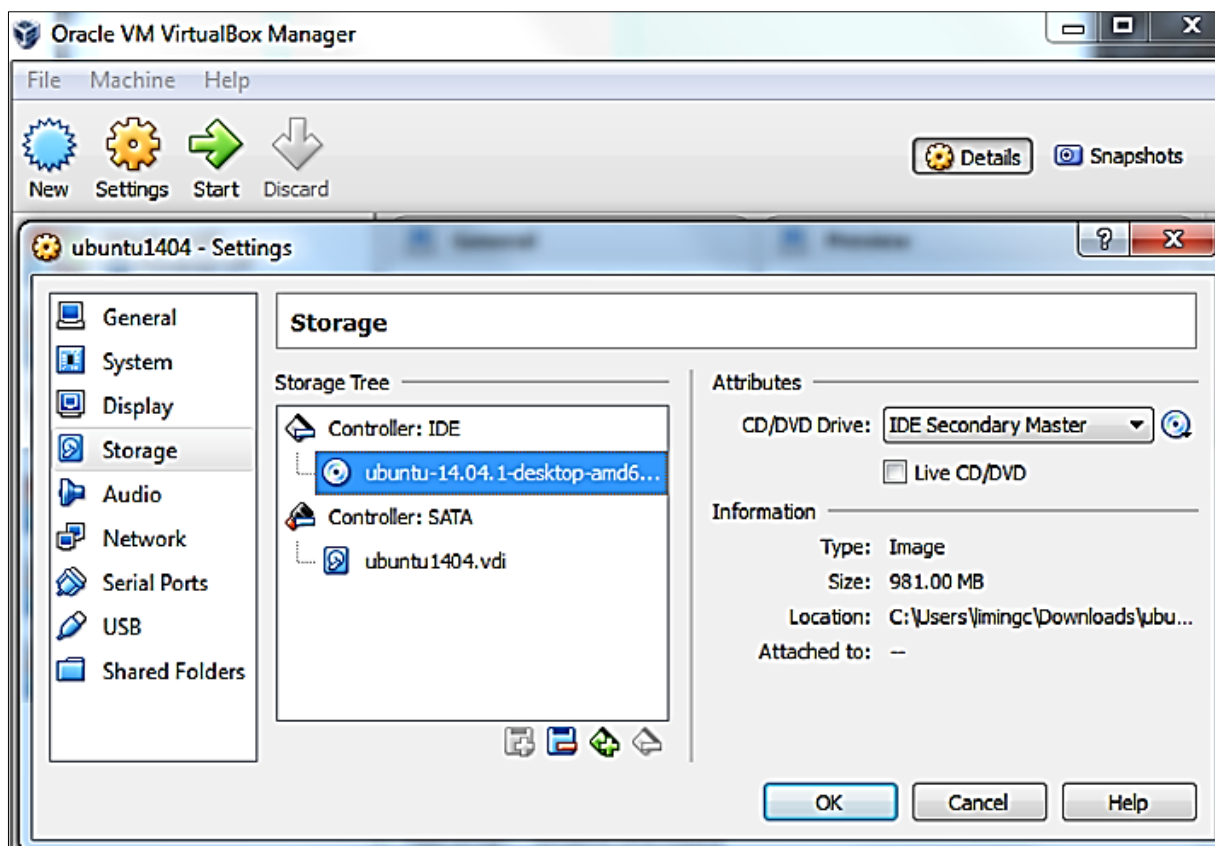


Figura 3.23. Fshirja e dokumentit instalues

Komenti: _____

3.6. Ushtrim praktik: Përdorimi i programit antivirus Avira

1. Qëllimi i ushtrimit

Tashmë kemi përmendur se Windows Defender është program antivirus sistemor, por për mbrojtje më të madhe një numër i madh i përdoruesve përdorin programe shtesë antivirus. Qëllimi i këtij ushtrimi është instalimi dhe deinstalimi i programit antivirus Avira dhe përdorimi i veglave softuerike në kuadër të tij.

2. Koha e realizimit: 1 orë mësimore

3. Shkarkimi dhe instalimi i programit antivirus Avira

- Nëse duam ta shfrytëzojmë këtë program, versionin e tij falas mund ta shkarkojmë (1) nga faqja zyrtare <https://www.avira.com> duke shtypur butonin Free Download. Do të paraqitet njoftimi për shkarkim të përfunduar të dokumentit instalues, zakonisht në dosjen për shkarkime (angl. Download).

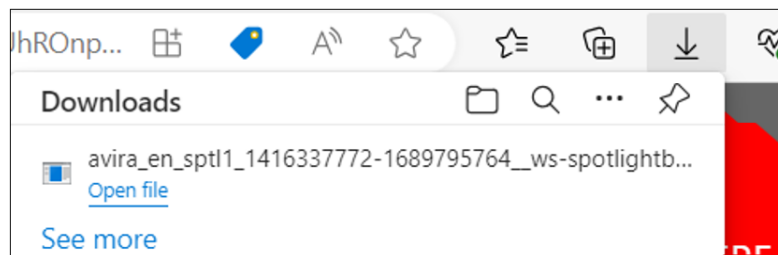


Figura 3.24. Shkarkimi i dokumentit instalues për programin Avira

- (2) Që të filloj instalimi, është e nevojshme që dokumenti të hapet, zgjedhim **Run**, shtypim **Agree dhe Install** dhe presim 4-5 minuta që procesi të përfundojë. Pas instalimit paraqitet ikonë me logon programit antivirus Avira dhe tani është gati për t'u përdorur.



Figura 3.25. Instalimi i programit antivirus Avira

4. Përdorimi i veglave softuerike të programit antivirus Avira

- Veglat softuerike në përbërje të programit antivirus Avira ndahen në tri kategori: Siguria (angl. Security), Privatësia (angl. Privacy) dhe Performansat (angl. Performance). Në kategorinë "Siguria" përfshihen:
- (1) Performansat (angl. Performance). Në kategorinë "Siguria" përfshihen: skanimi i viruseve, niveli i mbrojtjes, korigjimi dhe fshirja e dokumenteve të dyshimta, përditësimi i aplikacionit (angl. upgrade), mbrojtja nga sulmet e viruseve (angl. Firewall). Duke shtypur opsionin Skanim i viruseve (angl. **Virus scan**) fillon kontrolli. Ekzistojnë tre opsione: kontroll i plotë, kontroll i shpejtë ose planifikim i skanimit. Nëse kompjuteri nuk është i sigurt, do të paraqitet një njoftim për rrezik dhe është e nevojshme të shtypet butoni **Fix problem**.

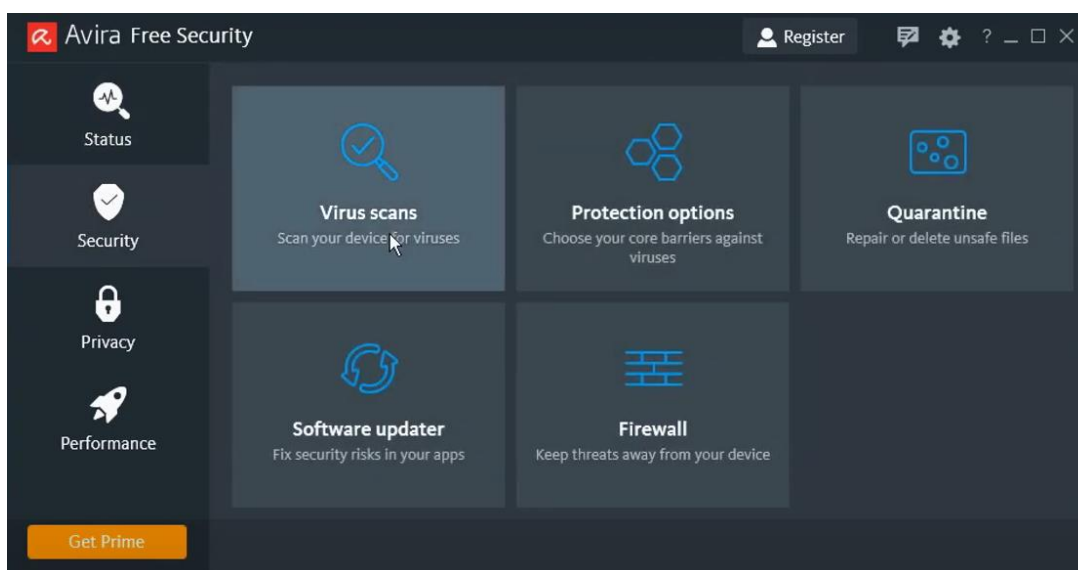


Figura 3.26. Zbulimi dhe largimi i viruseve me programin Avira

- (2) Kategoria "Privatësia" përmban veglat e mëposhtme: siguria gjatë kërkimit në internet (angl. Browser Safety), qasja në VPN, fshirja e të dhënave që rrezikojnë sigurinë gjatë punës, menaxhimi me fjalëkalimet, pengimi i ndarjes së të dhënave mes sistemit operativ Windows ose ndonjë aplikacioni tjetër.
- (3) Kategoria "Performansat" mundëson: optimizimin (lirimin e hapësirës së memories dhe përmirësimin e shpejtësisë së punës), optimizimin e aplikacioneve Startup, kursimin e baterisë, përditësimin e drajverëve të pajisjeve, gjetjen e dokumenteve të dyfishta.

5. Deinstalimi i programit antivirus Avira

Programet antivirus mund të ngarkojnë kompjuterin dhe të ulin shpejtësinë e funksionimit të tij. Prandaj nuk rekomandohet përdorimi i njëkohshëm i dy programeve antivirus. Nëse duam të ndryshojmë programin antivirus, atëherë ai duhet të deinstalohet.

- (1) Për këtë qëllim zgjidhet opsioni Settings → Apps → Apps and Features dhe në listën me programe zgjedhim programin Avira. Duke shtypur në klikun e

djathtë paraqitet dritare e re, shtypim Uninstall/Change dhe zgjedhim Uninstall.

- (2) Është e mundur që të marrim një njoftim se programi antivirus Avira nuk mund të deinstalohet sepse është i lidhur me nënprograme të tjera.

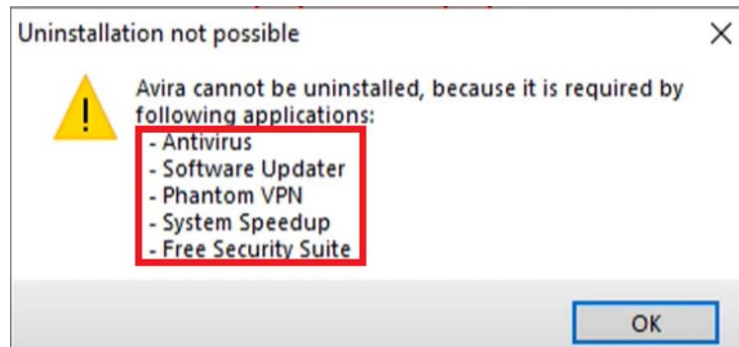


Figura 3.27. Nënprogramet e lidhura me programin antivirus Avira

Për të mundësuar deinstalimin e programit Avira, është e nevojshme që fillimisht të deinstalohen të gjithë nënprogramet që lidhen me të. Në listën e programeve, tek Apps and Features i gjejmë të njëjtit dhe duke shtypur në klikun e djathtë zgjedhim opsionin Uninstall.

- (3) Gjatë kohës së deinstalimit, është e nevojshme të konfirmojmë disa njoftime. Ristartojmë kompjuterin dhe kontrollomë në Program and Features nëse programi Avira është fshirë.

Komenti: _____

3.7. Ushtrim praktik: Instalimi dhe përdorimi i programit për kompresimin e dosjeve

1. Qëllimi i ushtrimit:

Qëllimi i këtij ushtrimi është të kompresojë dhe dekompresojë një dosje me më shumë dokumente, duke përdorur programe të ndryshme për kompresim. Pasi të kryhet kompresimi është e nevojshme të bëhet krahasimi i madhësisë së dosjeve të kompresuar dhe kohës së kompresimit dhe dekompresimit.

2. Koha e realizimit: 1 orë mësimore

3. Kompresimi dhe dekompresimi i një dosje me program për kompresim që është pjesë e vetë sistemit operativ Windows

Zgjedhim një dosje. Duke shtypur klikun e djathtë hapet një meny rënëse, zgjedhim Send to dhe opsionin Compressed (zipped) folder. Krijohet dosja e re me shtesën.zip edhe atë në të njëjtin vend ku ka qenë dosja burimore.

Dosja burimore dhe e kompresuar do të kenë të njëjtin emër, por do të ndryshojnë me shtesën.

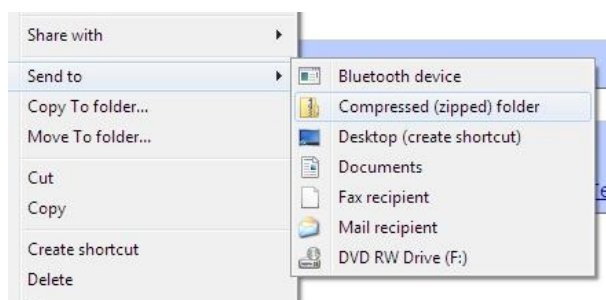


Figura 3.28. Kompresimi i dosjes

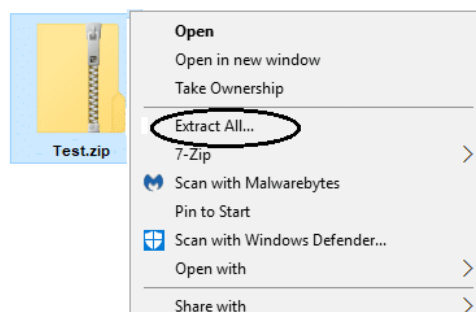


Figura 3.29. Dekompresimi i dosjes

Deri sa zgjat kompresimi duhet të matet koha e nevojshme për kryerjen e procedurës. Pasi përfundimit të kompresimit, kontrollojmë madhësinë e dokumentit të kompresuar duke shtypur në klikun e djathtë dhe zgjedhje të opsionit Properties.

Nëse duam të dekompressojmë një dokument ose dosje, atëherë bëjmë selektimin me klikun e majtë dhe me një klikim të djathtë zgjedhim **Extract All**. Hapet një dritare e re, zgjedhim destinacionin e dokumentit të ri të dekompresuar me opsionin Browse dhe klikojmë edhe një herë Extract.

4. Kompresoni një skedar me programin 7-ZIP

Programin 7-Zip falas mund ta shkarkohet nga faqja zyrtare e internetit www.7-Zip.com. Në dispozicion janë dy versione, 32 dhe 64 bitësh. Zgjidhet njëri nga opsionet dhe fillon shkarkimi i dokumentit për instalim.

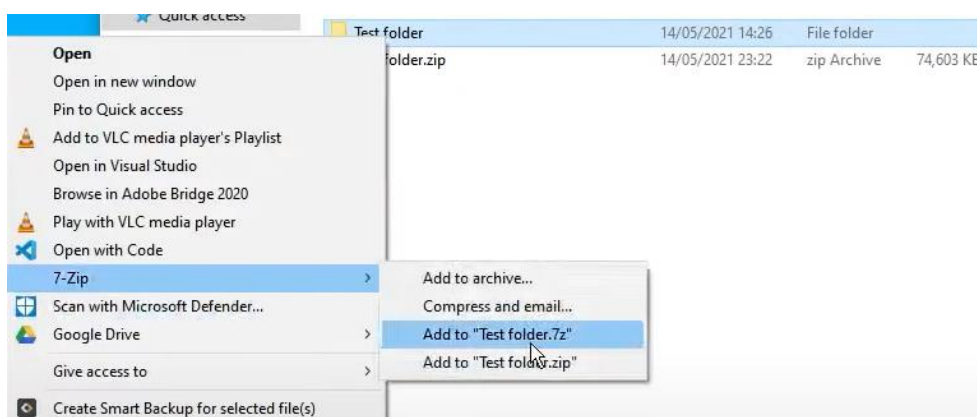


Figura 3.30. Zbatimi i programit 7-Zip

Pasi të keni shkarkuar dokumentin, me klik të dyfishtë, zgjidhim Install dhe fillon instalimi i tij. Zgjedhim të njëjtën dosje si në hapin e mëparshëm dhe me klik të djathtë e zgjedhim programin 7-Zip dhe në menynë e re rënëse, zgjedhim opsionin Add to archive (figura 3.30). Hapet një dritare e re e cila jep mundësinë

të zgjedhim formatin, shkallën dhe metodën e kompresimit. Metodatat më të përdorura janë LZMA ose LZM2. Pas shtypjes së butonit OK, fillon kompresimi, matim kohën e vetë procedurës dhe kontrollojmë madhësinë e dosjes së re të fituar të kompresuar me shtesën 7zip.

5. Krahasimi i rezultateve të fituara

Në tabelën numër 3.1. fusim rezultatet e fitara. Tabela mund të zgjerohet me programe të reja kompresimi si WINZIP, WINRAR, PeaZip ose BandZip, me të cilët u njohëm në njësinë mësimore 3.3.2. Programet për kompresimin e skedarëve nga pjesa teorike e librit.

Programi për kompresim	Koha e kompresimit (sek.)	Madhësia e skedarit të kompresuar	Shkalla e kompresimit	Koha e dekompresimit
Programi i kompresimit i Windowsit				
7-Zip				

Tabela 3.1. Krahasimi i programeve të ndryshme të kompresimit

Në të njëjtën njësi mësimore 3.3.2. Programet për kompresim të skedarëve, në pjesën teorike është shpjeguar mënyra për llogaritjen e shkallës së kompresimit.

Komenti: _____

4. Ushtrime praktike për punën me mikrokompjuterin në pllakë Arduino dhe programimin e tij

4.1. Masat për mbrojtje dhe siguri gjatë punës me mikrokompjuterin në pllakë Arduino

Ushtrimet praktike kryhen duke përdorur mikrokompjuterin në pllakë Arduino Uno R3 i cili punon me sinjale me tension të ulët dhe me rrymë të vogël. Për ta mbrojtur atë nga dëmtimi, duhet të merren disa masa paraprake.

- Përpara fillimit të çdo montimi, Arduino Uno R3 duhet të shkyçet nga burimi i tij i furnizimit.
- Bateria, si burim i furnizimit, lidhet me Arduino Uno R3 nëpërmjet pinave Vin dhe GND. Vlera maksimale e lejuar e tensionit është 20V, por rekomandohet përdorimi i baterive me tension nga 9 deri në 13V. Gjatë lidhjes duhet kushtuar vëmendje polaritetit të tensionit.
- Tensioni maksimal për pinat hyrës-dalës të Arduino Uno R3 është 5.5V.
- Rryma maksimale e hyrjes për vetëm një pin të Arduino Uno R3 është 40mA. Shuma e rrymave hyrëse për të gjithë pinat nuk duhet të jetë më e madhe se 200 mA.
- Nuk lejohet qarku i shkurtër i pinave.
- Konektorët e komponentëve të polarizuar duhet të identifikohen dhe të lidhen në mënyrë të duhur.
- Para fillimit të lidhjes të llogaritet intensiteti i rrymës përmes komponentëve elektrikë dhe të kontrollohet nëse e njëjta është e barabartë ose më e vogël se rryma maksimale e lejuar sipas dokumentacionit tekniko-teknologjik për atë komponent.
- Për lidhjen e motorëve ose konsumatorëve të tjerë induktivë, duhet të përdoren kontrollorë të përshtatshëm (angl. motor driver).
- Për kufizimin e rrymës nëpër diodat led të përdoren rezistorë.

4.2. Udhëzime për përdorimin e protoplakës

Saldimi është proces për prodhimin e pajisjeve elektronike në një pllakë të printuar me një dizajn të përhershëm, funksionim të qëndrueshëm dhe kontakte elektrike të sigurta. Por shpesh herë ndodh që dizajni i qarkut elektrik të ndryshohet me qëllim të përmirësimit të funksionalitetit dhe cilësisë së pajisjes. Kjo nënkupton ndryshimin e mënyrës së lidhjes së komponentëve, deri edhe në zëvendësimin e tyre. Protopllakat janë ideale për ndryshimin e shpejtë dhe të lehtë të dizajnit të qarqeve elektrike.

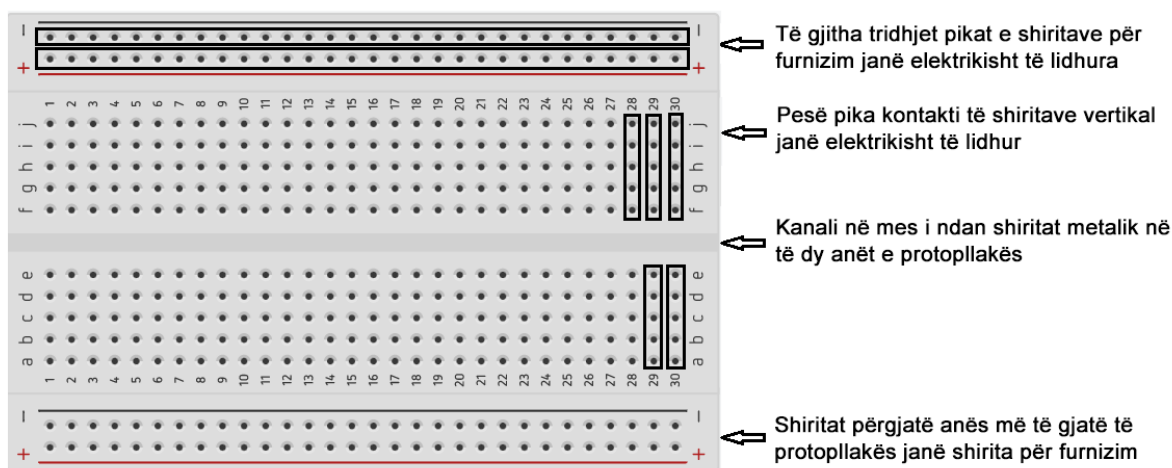


Figura 4.1. Pamja e jashtme e protoplakës

Protopllaka paraqet pllakë e përpunuar nga plastika e cilësisë së lartë me vrima në të që janë në të vërtetë pika kontakti ose pika lidhjeje. Në pikat e kontaktit vendosen terminalët e elementeve elektronike ose telat për lidhjen e tyre. Në figurën 4.1. është paraqitur pamja e jashtme e protoplakës. Nën maskën plastike janë përçuesit, të vendosur paralelisht në disa grupe, të cilët lidhin elektrikisht elementet elektronike. Në vetë përçuesit ndodhen kontakte në formë të kapseve dhe elementet elektronike mekanikisht përforcohen, kapen në protoplakë me shtypje të butë të terminaleve.

Elementet elektronike vendosen në shiritat vertikal, në mes të pllakës. Përgjatë anëve më të gjata të pllakës, në të dy anët ndodhen dy palë shiritat (shiritat horizontal në figurën 4.1.). Shiriti i shënuar me + ose i kuq lidhet me furnizimin, ndërsa shiriti i shënuar me - ose i zi lidhet me tokëzimin.

Për lidhje të rregullt të elementeve është shumë e rëndësishme të njihet shpërndarja e shiritave metalikë. Të gjitha pikat e kontaktit që shtrihen në të njëjtin shirit janë të lidhura në mënyrë elektronike me njëra-tjetrën dhe duhet pasur kujdes gjatë vendosjes së elementeve që të mos vij deri te lidhja e shkurtër e terminaleve, sepse atëherë nëpër to nuk do rrjedh rrymë. Për shembull, në figurën 4.2, dioda led e parë është e vendosur në mënyrë jo të rregullt sepse elektrodën e saj, anoda dhe katoda, janë vendosur në pikat e kontaktit që i përkasin të njëjtit shirit. Dioda e dytë në figurën 4.2. paraqet

lidhje të saktë të diodës led me protoplakën. Nëse duam të lidhim dy elemente, për shembull një diodë led dhe një rezistencë, atëherë terminalet e tyre duhet të vendosen në pikat e kontaktit që i përkasin të njëjtit shirit.

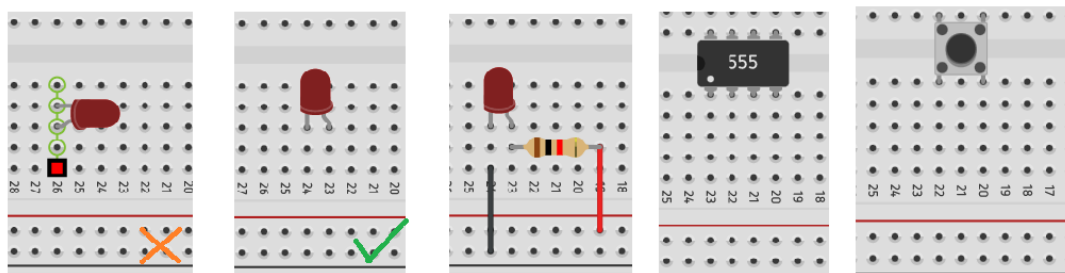


Figura 4.2. Lidhja e komponentëve elektrike në protoplakë

Në mesin e protoplakës ka një zgavër, kanal i cili elektrikisht i ndan përçuesit vertikalë në dy pjesë. Kjo lejon lidhjen e qarqeve të integruara me dy rreshta terminalesh (angl. DIP-Dual in Line). Qarku i integruar vendoset pikërisht mbi kanal në mënyrë që pinat e tij të vendosen në anë të ndryshme të kanalit dhe të mos jenë të lidhur elektrike. Pikat e kontaktit të protoplakës të integruara. Në kanal in e qendror vendosen edhe tasterë. Tasterët kanë katër terminale, nga dy në dy anët e kundërta. **Terminalet e të njëjtës anë nuk janë të lidhur elektrikisht** dhe ato duhet të vendosen në pikat e kontaktit që u përkasin dy përçuesve vertikalë, në mënyrë që qarku elektrik të mbyllet kur të shtypet tasteri.

4.3. Simulimi kompjuterik për Arduino Uno R3

Simulatorët kompjuterikë janë bërë vegla shpesh të përdorura gjatë mësimin dhe punës në fushën e elektronikës dhe teknikës kompjuterike. Ata nxisin kuriozitetin, shpikjen, të menduarit intuitiv dhe janë më të sigurt për punë.

Ushtrimet praktike të planifikuara duhet të kryhen në një kabinet për mësim praktik dhe për këtë kërkohet pajisje përkatëse, harduer dhe elemente elektronike. Disa nga ushtrimet mund të realizohen plotësisht edhe me aplikimin e simulimit kompjuterik. Tinkercad është platformë internet falas që ofron simulime të shkëlqyera kompjuterike për Arduino Uno R3.

Përdoruesi duhet të ketë një llogari përdoruesi në Google ose Apple dhe ta përdorë atë për t'u identifikuar në faqen zyrtare të platformës www.Tinkercad.com. Pas **kyçjes** me sukses hapet faqja kryesore me menyne kryesore. Mundësitë janë shumë të mëdha. Të rikujtojmë se platforma Tinkercad ofron punë me dizajn 3D, krijimin e kodeve programuese në gjuhën programuese Codeblocks dhe dizajnimin e qarqeve elektrike. Neve na duhet të simulojmë qarqe elektrike me përdorimin e Arduino Uno R3 dhe gjuhës programuese C/C++.

Duke klikuar në opsionet Resources → Learning Center → Learn Circuit (shq. Resusret → Qendra për mësim → Mësimi i qarqeve elektrike) hapen udhëzimet për zgjedhjen e komponentëve, lidhjen e tyre, shkrimin e programeve dhe simulimin e tyre. Udhëzimet për punë janë të shkruar në dhe janë interaktiv, që do të thotë se mund të aplikohen menjëherë sepse me hapjen e udhëzimeve hapet edhe sipërfaqja punuese dhe menya për zgjedhjen e komponentëve.

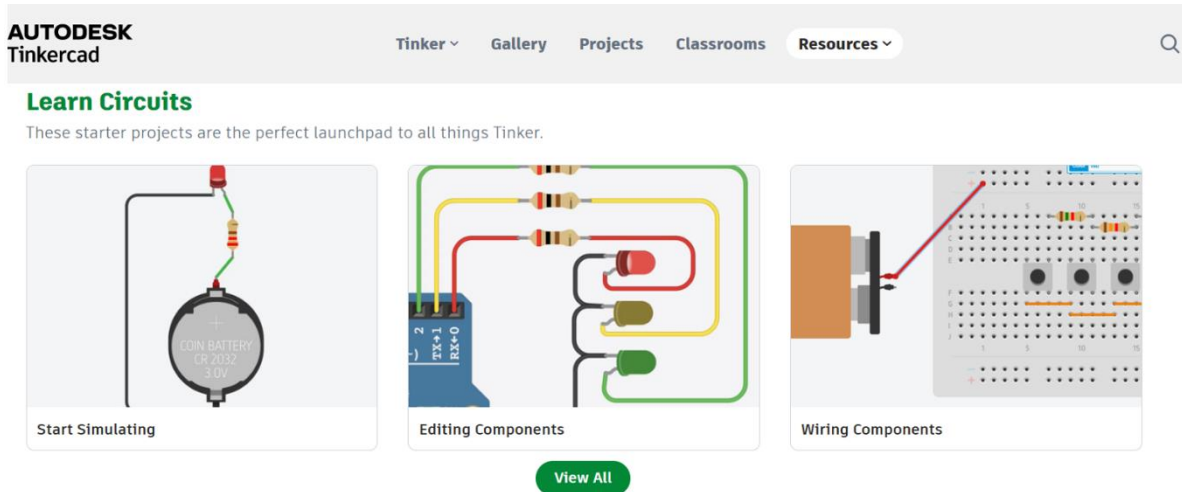


Figura 4.3. Hapja e udhëzimeve për krijimin e qarqeve elektrike duke përdorur Arduino Uno R3

Duke shtypu (klikuar) në figurën e profilit dhe zgjedhje të opsionit Dizajni im (angl. My Designs) hapen të gjitha projektet që i kemi punuar më parë. Ato ruhen nga vetë ueb platforma. Nëse duam të krijojmë një qark elektrik të ri, atëherë zgjedhim opsionin I ri (angl. New).

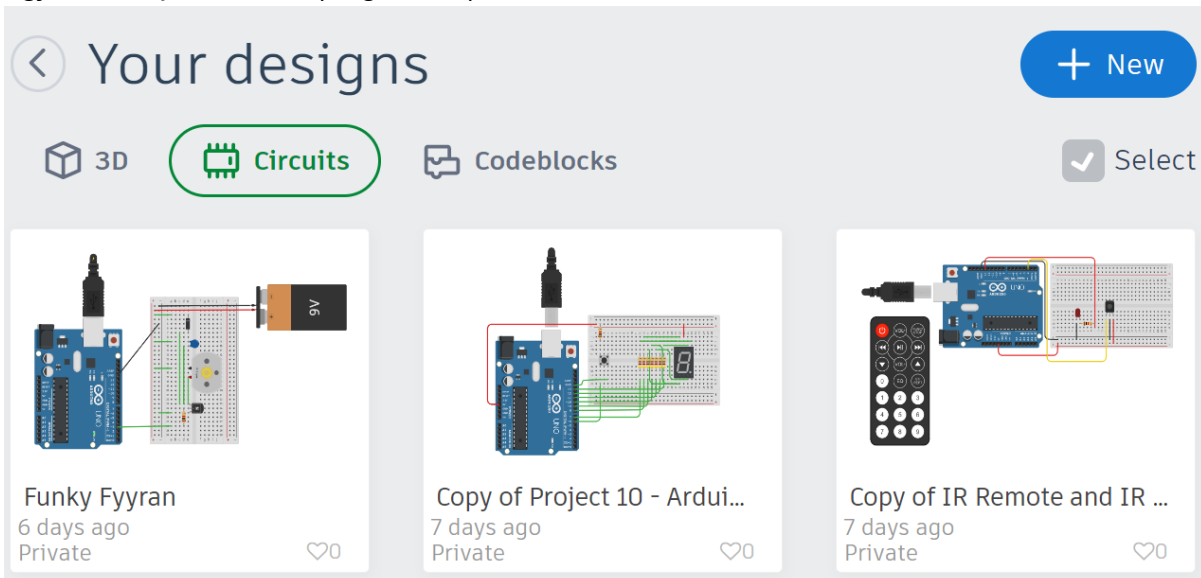


Figura 4.4. Hapja e dizajneve të reja dhe të vjetra të qarqeve elektrike

Në figurën 4.5. është paraqitur sipërfaqja punuese për formimin e qarqeve elektrike dhe programimin e Arduino Uno R3. Qarku elektrik formohet shumë lehtë, me tërheqjen dhe lëshimin e komponentëve elektronike, ndërsa më pas

ato lidhen. Programin e punës mund ta shikojmë ose ta shkruajmë duke klikuar butonin **Code** dhe duke zgjedhur opsionin **Text**. Pasi të formojmë qarkun elektrik dhe të shkruajmë programin, shtypim butonin **Start Simulation** dhe vëzhgojmë nëse qarku funksionon siç është menduar. Nëse ka gabime në kod, të njëjtit do të shënohen me ngjyrë të kuqe.

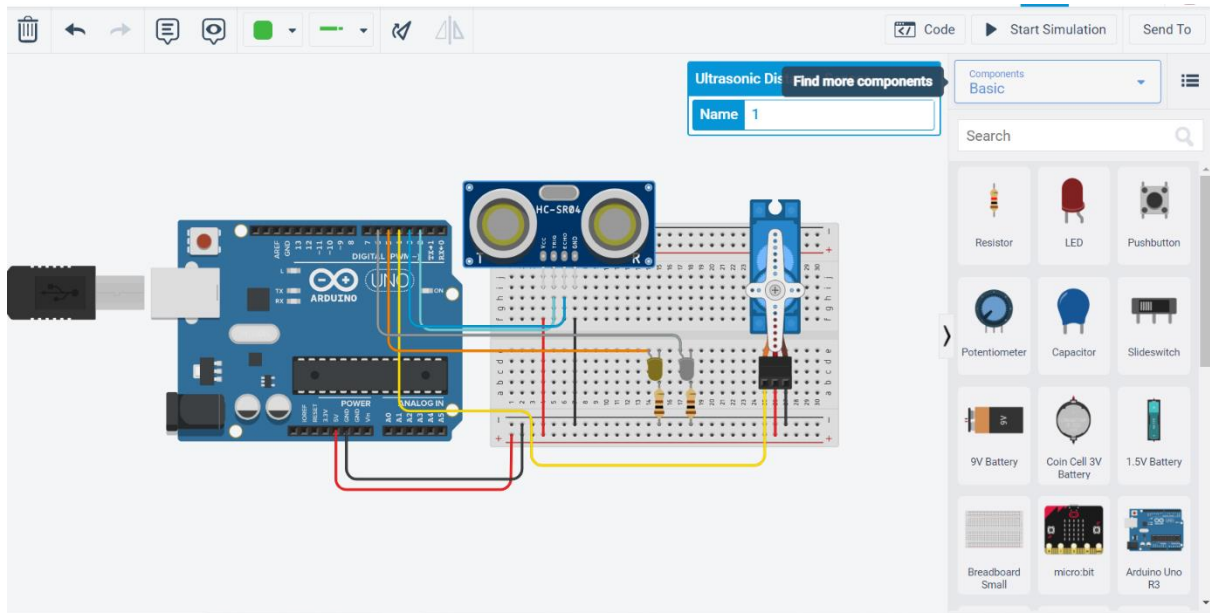


Figura 4.5. Pamja e sipërfaqes punuese për punë me qarqe elektrike në web platformën Tinkercad

Web platforma e internetit Tinkercad është me burim të hapur dhe lejon shpërndarjen e projekteve. Duke klikuar në opsionin Dërgo te (angl. Send To) në vetë projektin hapet një dritare e re, zgjedhim Fto njerëz (angl. Invite people), e kopjojmë (angl. Copy) linkun për shpërndarje dhe më pas ngjitet (angl. Paste) dhe ia dërgojmë projektin personit të dëshiruar përmes emailit ose një rrjeti social. Në fakt, duke kërkuar në internet mund të gjejm shumë projekte të gatshme të përpunuar në Tinkercad dhe me hapjen e tyre dhe zgjedhje të opsionit Mendo (angl. Tinker this) web shfletuesi e hap drejtpërdrejt projektin e zgjedhur në platformën Tinkercad. Kjo mund të ndihmojë shumë në përmirësimin e projekteve tona.

4.4. Ushtrim praktik: Instalimi i mjedisit zhvillimor për mikrokompjuterin Arduino dhe vënia në përdorim

4.4.1. Udhëzim për instalimin e mjedisit zhvillimor për mikrokompjuterë në pllakë Arduino

Programi zhvillimor shkarkohet nga faqja zyrtare e Arduinos, konkretisht nga linku i mëposhtëm <https://www.arduino.cc/en/software>. Mund të zgjedhim njërin prej disa versioneve të ndryshme të programeve zhvillimore Arduino, por nëse përdorim sistemin operativ Windows 10, është mirë të shkarkojmë versionin më të fundit. Duhet të theksojmë se ky version nuk do të funksionojë në sistemet operative më të vjetra, si Windows XP ose Windows 7. Pas shkarkimit të dokumentit për instalim, e instalojmë programin zhvillimor. Shtypim (click) në "Run", i pranojmë kushtet e licencimit, zgjedhim një dosje për të ruajtur programin dhe në fund klikojmë "Install".[5]

Pas instalimit, e lidhim Arduino Uno R3 me kompjuterin personal nëpërmjet një kabloja USB me dy konektor të ndryshëm, të tipit A dhe B. Arduino Uno R3 merr energji nga kompjuteri personal dhe ndriçon dioda e integruar e gjelbër e shënuar me shkronjën L. Kur pajisjen Arduino Uno R3 e lidhim për herë të parë me kompjuter, **automatiksht instalohet drajveri**. Drajveri është softuer i cili i mundëson kompjuterit të komunikojë me pajisje të re.

Nëse blhet fjalë për version më të vjetër të sistemit operativ Windows, mund të shfaqet një dritare që ju kërkon të specifikoni lokacionin nga i cili duhet të instalohet drajveri. Nëse nuk fillon instalimi, është e nevojshme të hapet programi **Control Panel**, të zgjidhim kategorinë **Device Manager**, më pas nënkategorinë **"Other Devices"** ose **"Unknown Devices"** dhe të shtypet (angl. click) **"Update Drivers"** ose **"Update Driver Software"**. Më pas specifikojmë drajverin për lidhjen e mikrokompjuterit në pllakë Arduino, kështu zgjedhim dosjen "Drivers" në përbërje të vetë dosjes Arduino.

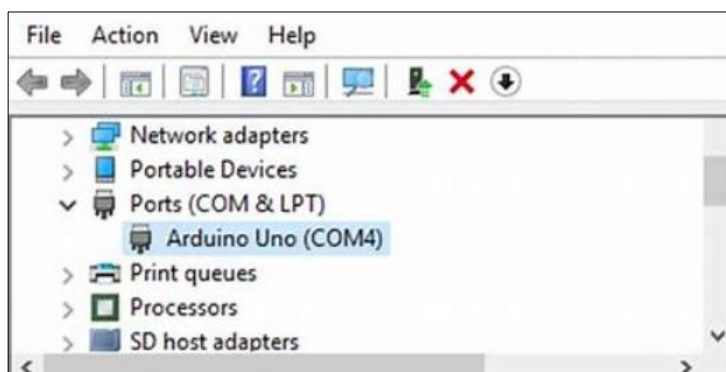


Figura 4.6. Gjetja e drajverit për Arduino Uno

Si konfirmim për instalim të suksesshëm, në kategorinë Device Manager, duke klikuar në opsionin "PORT (COM&LPT)", duhet të paraqitet Arduino Uno. Nëse nuk e dimë se me cilën portë është i lidhur mikrokompjuteri Arduino, atëherë mund ta shkyçim Arduino-n nga kompjuteri, pastaj ta kyçim përsëri dhe të shohim se cila nga portat serike shfaqet si e re.

Arduino përdor kode programore me burim të hapur, që do të thotë se programet janë falas. Çdo programues mund të publikojë kodet programuese të tij dhe kështu të kontribuojë në zhvillimin e kësaj platforme. Vetë mjedisi zhvillimor i Arduino përmban shumë programe të gatshme dhe listën e këtyre programeve mund ta shohim duke klikuar File → **Examples**. Program të gatshëm mund të shkarkojmë edhe nga ndonjë faqe interneti Arduino dhe ta hapim duke klikuar në File → Open → Look in dhe në fund dosjet në të cilën e kemi ruajtur programin.

4.4.2. Menyja dhe veglat e mjedisit zhvillimor dhe shkrimi i programi në Arduino Uno R3

Në figurën 4.7. është paraqitur programi zhvillimor i Arduino. **Pjesë përbërëse më kryesore janë: menyja, shiriti me vegla, editore i tekstit, monitori serik, bibliotekat dhe konzola për debugim, korigjim.**

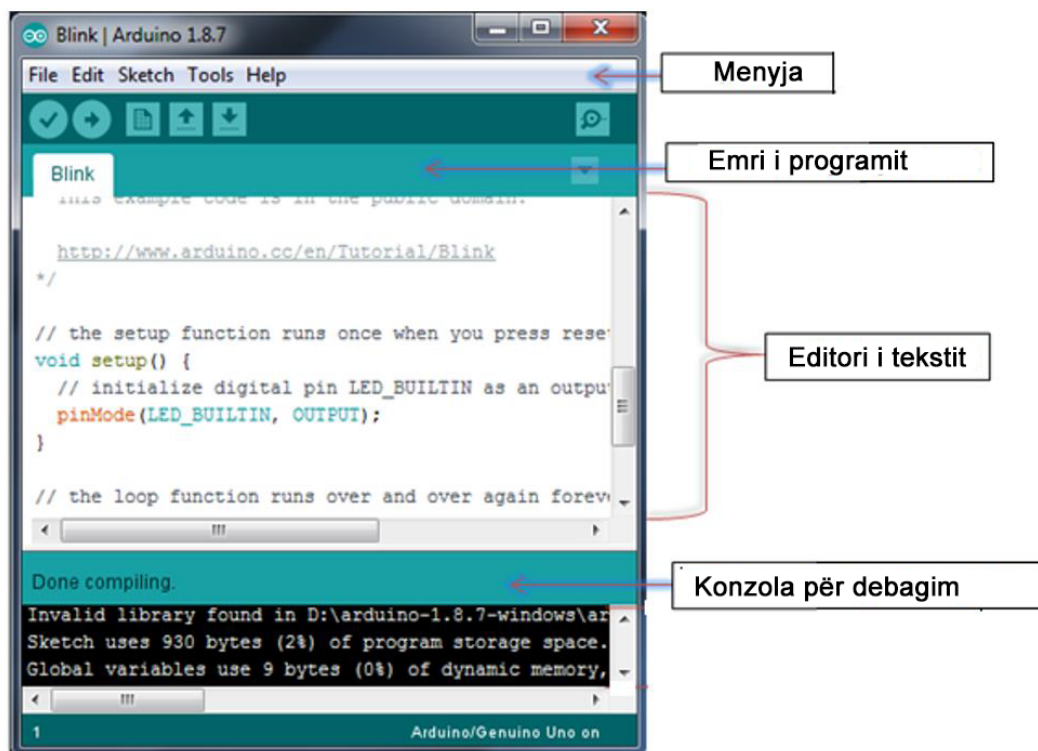


Figura 4.7. Pamja e mjedisit zhvillimor për mikrokompjuterin në pllakë Arduino

Menyja → Në meny ndodhen kategoritë e mëposhtme: File, Edit, Sketch, Tools, Help.

Shiriti me vegla —→ Në shiritin me vegla ndodhen butonat: Verifiko/Kompajlo (angl. Verify/Compile), Ngarko domethënë shkruaj (angl. Upload), hapni një dokument të ri (angl. New), hapni një dokument të vjetër (angl. Open) dhe ruaje (angl. Save). Me veglën Verify kontrollohet kodi, rresht pas rreshti, me qëllim që të zbulohen gabimet eventuale. Me veglën Upload, kodi i verifikuar shkruhet në memorien programuese të mikrokontrolluesit të ATmega328.

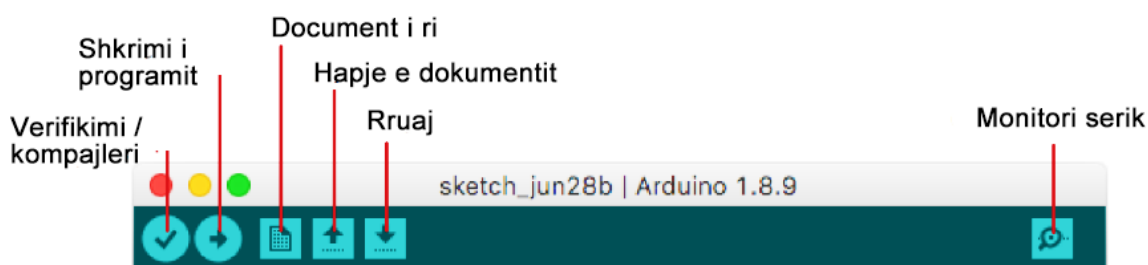


Figura 4.8. Shiriti me vegla për mjedisin zhvillimor të rduino

Editori i tekstit —→ Kjo është pjesa më e rëndësishme e programit zhvillimor sepse në të shkruajmë kodet, programet në gjuhën programuese C/C++.

Monitori serik —→ Me shtypjen e tij mundësohet shikimi i të dhënave që dërgohen ose merren nga Arduino. Ka një zbatim të madh tek aplikacionet në kohë reale.

Konzola për debugim —→ Kjo dritare na jep informacion për kompajlimin e suksesshëm ose tregon numrin rendor të rreshtit që shkakton gabim gjatë kompajlimit. Kompajlimi është procesi i përkthimit të programit nga një gjuhë programuese e nivelit më të lartë në gjuhën e makinës.

Bibliotekat (angl. libraries) —→ Me shtypjen (angl. click) e kësaj kategorie, hapet lista e nënprogrameve për punë me pajisje të ndryshme hyrësdalëse.

Pas instalimit të mjedisit zhvillimor dhe njohjes me menynë dhe veglat do të shpjegojmë procedurën e shkrimit të një programi në memorien e Arduino Uno R3. Për këtë qëllim do të përdorim diodën led të integruar e cila është e lidhur me pinin 13-të të Arduino Uno R3. Gjithashtu do të përdorim një program të gatshëm i cili nuk kërkon përdorimin e komponentëve tjera elektronike. Ky ushtrim është një nga ushtrimet më të thjeshta dhe është lloj testi i funksionimit të Arduino Uno R3.

1. Pas instalimit, e lidhim Arduino Uno R3 me kompjuterin personal nëpërmjet një kabloje USB me dy konektor të ndryshëm, të tipit A dhe B.
2. Hapi i parë është zgjedhja e llojit të mikrokompjuterit Arduino dhe zgjedhje e portës serike. Të rikujtojmë se familja Arduino përbëhet prej më tepër platformave të ndryshme: Uno, Mega, Nano, Leonardo, etj. Për të zgjedhur Arduino Uno R3 duhet të shtypim në opsionet Tools → Board → Arduino Uno. Kjo është paraqitur në figurën 4.9.

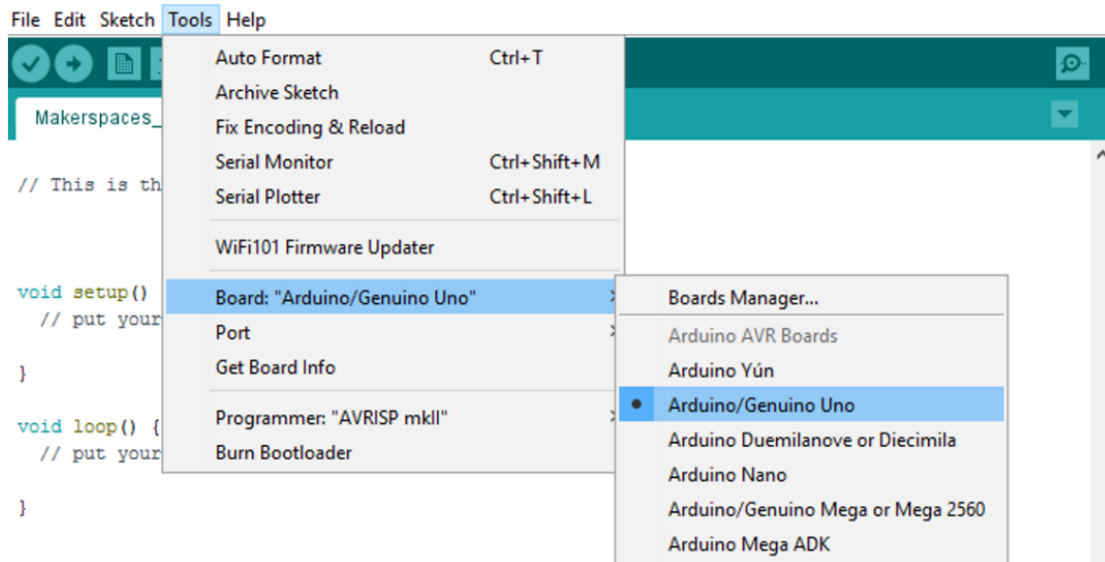


Figura 4.9. Procedura e përzgjedhjes së mikrokompjuterit Arduino

3. Për zgjedhjen e portës, është e nevojshme të shtypet opsionin Tool → Serial Port. Kjo është paraqitur në figurën 4.10. Gjatë kësaj paraqitet një meny rënëse e portave serike të disponueshme dhe kjo ndryshon në varësi të pajisjeve që janë të lidhura me kompjuterin.

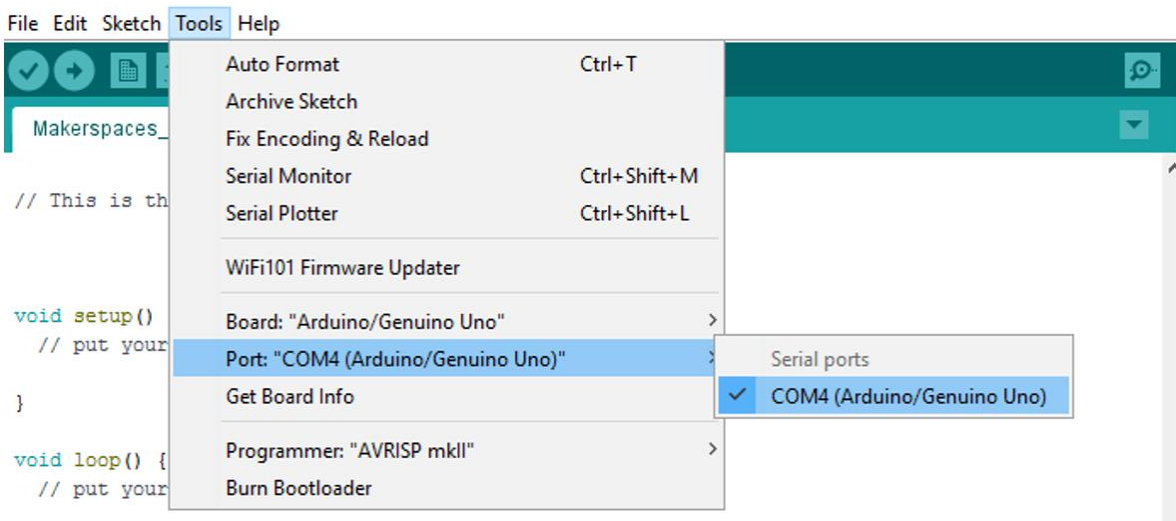


Figura 4.10. Procedura e përzgjedhjes së portës për Arduino Uno

4. Kodin programor e gjejmë duke shtypur File → Example, ndërsa më pas zgjedhim 01. Basic → Blink. (figura 4.11.)
5. Përpara se programi të shkruhet në memorien e Arduino Uno R3, ai duhet të përkthehet në gjuhën e makinës, domethënë të kompajlohet. Për këtë qëllim në listën e veglave shtypim butonin Verify/Compile ose në meny zgjedhim Sketch → Verify/Compile.

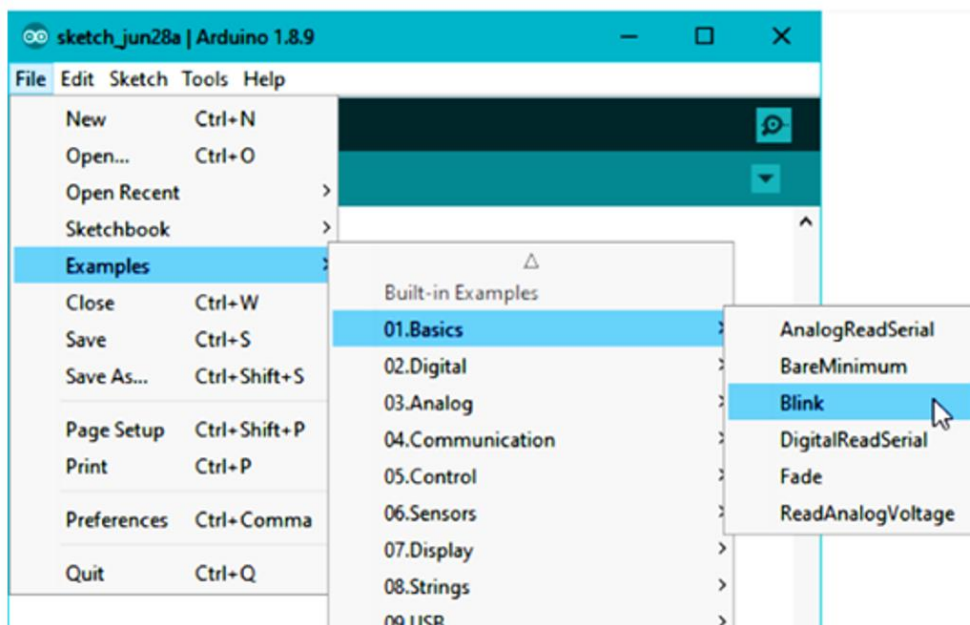


Figura 4.11. Hapja e programit Blink për Arduino Uno R3

6. Në konzolën për debugim marrim njoftimin "Compajling Sketch....". dhe paraqitet shiriti për progresin e procedurës dhe në fund duhet të marrim njoftimin "Done Compiling" (figura 4.12.).



Figura 4.12. Njoftim për kompajlimin e suksesshëm

7. Nëse nuk ka gabime në kodin programor, në konzolën për debugim marrim informacionin për madhësinë e programit që e kemi hapur të shprehur në bajta dhe informacion për madhësinë e hapësirës memoruese të mbetur të lirë.
8. Me shtypjen e butonit Bashkangjit (angl. Upload), fillon shkrimi i programit në memorie. Dioda led e integruar me shenjën L do të ndalojë së ndriçuari, ndërsa do të fillojë të ndriçojë dy diodat për transmetimin serik, me shenjat RX dhe TX. Regjistrimi zgjat disa sekonda, ndërsa më pas fillon të pulsojë (angl. blink) dioda led me shenjën L.

4.5. Ushtrime praktike për programimin e Arduino Uno R3 në gjuhën programuese C/C++

Të gjitha ushtrimet janë të thjeshta dhe të destinuara për fillestarët, me qëllim të zbatimit praktik të njohurive teorike nga programimi i Arduino Uno R3. Për programim të suksesshëm të Arduino Uno R3 është e nevojshme njohja e harduerit dhe softuerit të tij. Para fillimit të ushtrimeve praktike duhet të përsëritet materiali mësimor i mëposhtëm:

- 4.3. Elementet periferike dhe komponentët elektronikë për lidhje me mikro-kompjuterin në pllakë Aduino, nga pjesa teorike e librit
- 4.1. Masat për mbrojtje dhe siguri gjatë punës me mikrokompjuterin në pllakë Arduino, nga pjesa praktike e librit
- 4.2. Udhëzime për punën me protopllakën, nga pjesa praktike
- 4.3. Simulimi kompjuterik për Arduino Uno R3, nga pjesa praktike
- 4.4.2. Menyja dhe veglat e mjedisit zhvillimor dhe shkrimi i programit në Arduino Uno R3, nga pjesa praktike

4.5.1. Ushtrim praktik: Kycja e diodës led nëpërmjet tasterit

1. Qëllimi i ushtrimit

Qëllimi i ushtrimit është lidhja e diodës ledh dhe një tasteri, si komponentë hyrëse dhe dalëse, me Arduino Uno R3, sipas skemës së dhënë funksionale dhe të montimit dhe përdorimi i protopllakës. Nxënësit duhet të shkruajnë një program në mjedisin zhvillimor, ta vendosin programin në memorien e Arduino Uno R3 dhe ta testojnë funksionimin e tij.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për zbatimin e këtij ushtrimi janë:

- Mikrokompjuter në pllakë Arduino Uno R3
- protopllakë
- taster për pllakën e printuar
- diodë led
- rezistorë me rezistenca $R1=220\Omega$ dhe $R2=1K\Omega$
- tela për lidhje (qarqet të shkurtëra)
- kompjuter me mjedisin zhvillimor Arduino të instaluar

Tasteri dërgon vlerat (HIGH ose LOW) deri te Arduino Uno R3, ndërsa më pas Arduino komandon me diodën led. Rezistenca R1 nevojitet për të përshtatur rrymën e funksionimit të diodës, me qëllim që intensiteti i dritës të jetë optimal, pa ngrohje të

dëmshme. Rezistenca R2 kryen mbrojtjen e pinit në të cilin është i lidhur tasteri nga ndryshimet e shpejta të tensionit të tij.

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- diagramin e pinave të Arduino Uno R3
- mënyrën e lidhjes së diodës led dhe tasterit në protoplakë. Të njëjtat shpjegohen në udhëzimet për punë me protoplakën.
- instruksionet hyrëse-dalëse për Arduino Uno R3, strukturat e domosdoshme (setup dhe loop) dhe struktura if...else për zgjedhjen e mundësive.

5. Përshkrimi i skemës elektrik dhe mënyra e lidhjes

Në figurën 4.13. është paraqitur skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

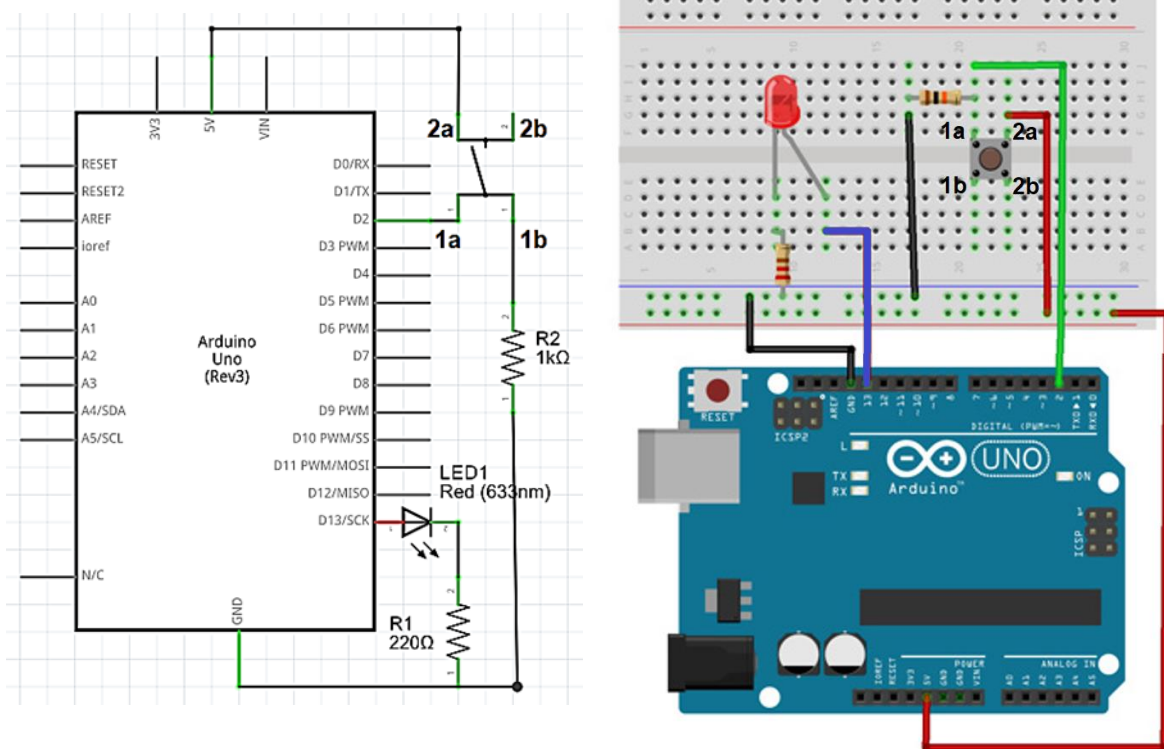


Figura 4.13. Lidhja e diodës led dhe tasterit me Arduino Uno R3

Kur butoni nuk është i shtypur, pini i dytë i Arduino Uno R3 është i lidhur me tokëzimin që korrespondon me nivelin logjik të ulët (LOW) dhe tensioni i pinit të dytë është 0V. Kur tasteri është i shtypur pini i dytë i Arduino Uno R3 është i lidhur me furnizimin gjë që korrespondon me nivelin logjik të lartë (HIGH) dhe tensioni i pinit të dytë është i ndryshëm nga zero. Dioda led ndriçon kur pini i 13 është në nivel të lartë logjik, gjegjësisht kur tensioni i daljes i pinit është 5V.

Lidhja e komponentëve realizohet në këtë mënyrë:

- (1) E vendosim tasterin në mesin e protoplakës. Për lidhjen e tasterit përdoren tre qarqe të shkurtëra. Me qarkun e shkurtër me ngjyrë të kuqe

i lidhim terminalet 2a-2b me shiritin për furnizim (të shënuar me + ose ngjyrë të kuqe). Me qarkun e shkurtër të zi dhe përmes rezistencës R2 i lidhim terminalet 1a-1b të tasterit me shiritin për tokëzim (të shënuar me minus ose ngjyrë të kaltër). Në të njëjtën kohë, me qarkun e shkurtër e gjelbër, lidhim terminalet 1a-1b i lidhim me pinin e dytë digjital të Arduino Uno R3. Qarku i shkurtër i gjelbër është lloj teli sinjalizues dhe përmes tij tasteri i dërgon vlerat HIGH ose LOW deri në pinin e dytë.

- (2) E vendosim diodën led në protoplakë me ç'rast duhet pasur kujdes që anoda të jetë e lidhur me pikën e kontaktit me potencial më të lartë se potenciali i katodës. Me qark të shkurtër të zi dhe përmes rezistencës R1 e lidhim katodën e diodës led me shiritin për tokëzim, ndërsa qark të shkurtër të kaltër e lidhim anodën me pinin e 13-të të Arduino Uno R3.
- (3) Me qark të shkurtër të kuq e lidhim pinin 5V të Arduino Uno R3 me shiritin për furnizim të protoplakës, ndërsa me telin e zi e lidhim pinin GND me shiritin për tokëzim.

6. Shkrimi dhe ngarkimi i programit për Arduino Uno R3

Arduino Uno R3 e kontrollon gjendjen e tasterit duke përdorur instruksionin `digitalRead()`. Dioda led kyçet dhe shkyçet nëpërmjet softuerit duke përdorur instruksionin `digitalWrite()`. Programi përmban një strukturë `if...else` për zgjedhjen e njëjës prej dy mundësive (dioda led ndriçon ose dioda led nuk ndriçon). Gjendja e tasterit është kusht për ekzekutimin e strukturës `if...else`.

Më poshtë është dhënë programi me të cilin dioda led kyçet dhe shkyçet përmes softuerit, në varësi të gjendjes së tasterit (i shtypur ose i lëshuar).

```
// Bëhet deklarimi i konstantave dhe variablave përpara se të shkruhen
// strukturat setup dhe loop
// Konstantat nuk ndryshohen dhe përdoren për zgjedhjen e pinave për lidhje
1  const int buttonPin = 2;           // Tasteri është i lidhur në pinin digjital
                                       // numër 2.
2  const int ledPin = 13;             // Dioda led është e lidhur me pinin
                                       // digjital me numër 13
// Variablat ruajnë të dhëna që vazhdimisht ndryshojnë.
3  int buttonState = 0;               // buttonState është emri për vlerën që
                                       // fitohet me leximin e pin it numër 2, ku
                                       // është i tasteri.
4  void setup() {                     // Fillimi i strukturës setup.
5    pinMode(ledPin, OUTPUT);         // Pini në të cilin është e lidhur dioda led
                                       // konfigurohet si pin dales, që do të thotë
                                       // se Arduino do të dërgojë të dhëna.
6    pinMode(buttonPin, INPUT);       // Pini në të cilin është i lidhur tasteri
                                       // konfigurohet si pin hyrës, që do të thotë
                                       // se Arduino do të pranojë të dhëna.
```

```

7   } // Fundi i strukturës setup.
8   void loop() { // Fillimi i strukturës loop.
9     buttonState=digitalRead(buttonPin); // Lexohet vlera e pinit numër 2 dhe ajo
// regjistrohet në variablën
// buttonState.
10    if (buttonState == HIGH) { // Kontrollohet nëse tasteri është i
// shtypur.
11      digitalWrite(ledPin, HIGH); // Nëse është i shtypur, atëherë
// dioda led ndriçon.
12    } // Fundi i deklaratës if.
13    else { // Fillimi i deklaratës else.
14      digitalWrite(ledPin, LOW); // nëse nuk i shtypur tasteri,
// dioda nuk ndriçon.
15    } // Fundi i deklaratës else.
16  } // Fundi i strukturës loop.

```

Procedura për futjen e një programi në memorien e Arduino Uno R3 është e njëjtë për të gjitha programet, ndërsa e njëjta është shpjeguar hap pas hapi në ushtrimin praktik 4.4.2. Menytë dhe veglat e mjedisit zhvillimor dhe shkrimi i një programi në Arduino Uno R3.

7. Rezultatet nga realizimi i ushtrimit

Menjëherë pas futjes së programit duhet të kontrollohet nëse Arduino Uno R3 funksionon siç duhet. Kur të shtypim tasterin dioda led duhet të ndriçojë. Nëse e lëshojmë tasterin atëherë dioda led ndalon së ndriçuari.

Komenti: _____

8. Bëni një ndryshim!

- Çfarë do të ndodhte nëse të dy instruksionet DigitalWrite në program i zëvendësojnë vendet?

Komenti: _____

- Çfarë do të ndodhë nëse vendosim një kohë vonese prej 5s pas të dy instruksioneve DigitalWrite? Të përdoret instruksionin delay().

Komenti: _____

4.5.2. Ushtrim praktik: Kontrollimi i diodës led me potenciometër

1. Qëllimi i ushtrimit

Ky ushtrim paraqet zbatim praktik të hyrjeve analoge të Arduino Uno R3. Nxënësit do të mësojnë se si madhësia e sinjaleve të hyrjes analoge ndikon në kohëzgjatjen e

sinjaleve në dalje të pinave digjital të Arduino Uno R3. Me ndihmën e një potenciometri do të ndryshojmë shpejtësinë e pulsimit të diodës led.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Mikrokompjuter Arduino Uno R3
- protoplakë
- potenciometër - trimer me rezistencë 10 K Ω
- diodë led
- rezistorë me rezistencë R1=220 Ω
- tela lidhës (qarqe të shkurtëra)
- kompjuter me mjedisi zhvillimi Arduino të instaluar

Në këtë ushtrim praktik potenciometri është komponenti kryesor që është i lidhur me njërën nga hyrjet analoge të Arduino Uno R3. Potenciometri është me rezistencë të ndryshueshme dhe me të gjenerohet tension i ndryshueshëm me vlerë maksimale prej 5V dhe një vlerë minimale 0V. Konvertuesi analogo-digjital i integruar në vetë Arduino i konverton vlerat analoge në numra të plotë nga 0 deri në 1023. Sa më e madhe të jetë vlera në dalje të konvertorit analogo-digjital, aq më rrallë dioda do të kyçet dhe shkyçet, domethënë dioda led do të pulsojë me shpejtësi më vogël.

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen përa:

- modulimin në gjerësi të impulseve dhe rëndësinë e tij për hyrjet analoge të Arduino Uno R3
- instruksionet për punë me pinat hyrës analog të Arduino Uno R3
- kuptimin e instruksionit delay() dhe zgjedhjen e kohës së vonesës

5. Përshkrimi i skemës elektrike dhe mënyra e lidhjes

Në figurën 4.14. tregohet skema funksionale dhe i montimi për realizimin e këtij ushtrimi. Potenciometri ka tre terminale prej të cilëve dy terminale janë furnizimi dhe tokëzimi. **Pini i mesëm i potenciometrit** është pini i kontrollit ose i të dhënave dhe atë e lidhim me pinin analog A0 të Arduino Uno R3.

Mënyra e lidhjes është si vijon:

- (1) E vendosim potenciometrin në protoplakë me ç'rast duhet pasur kujdes që terminalet e tij t'i vendosim në pika kontakti që u përkasin përçuesve të ndryshëm. Terminalet e potenciometrit janë me numra nga 1 deri në 3. Terminalet me numër 1 dhe 3 lidhen me shiritat për furnizim dhe tokëzim. Pini me numër 2, përmes qarkut të shkurtër të verdhë, lidhet drejtpërdrejt me pinin analog A0.

- (2) E vendosim diodën led në protoplakë ashtu siç u shpjegua në ushtrimin e mëparshëm, duke e lidhur katodën me shiritin e tokëzimit, ndërsa anodën me pinin e 13-të.
- (3) Pinat e shënuar me GND dhe 5V të Arduino Uno R3 i lidhim me shiritat për tokëzim dhe furnizim të protoplakës.

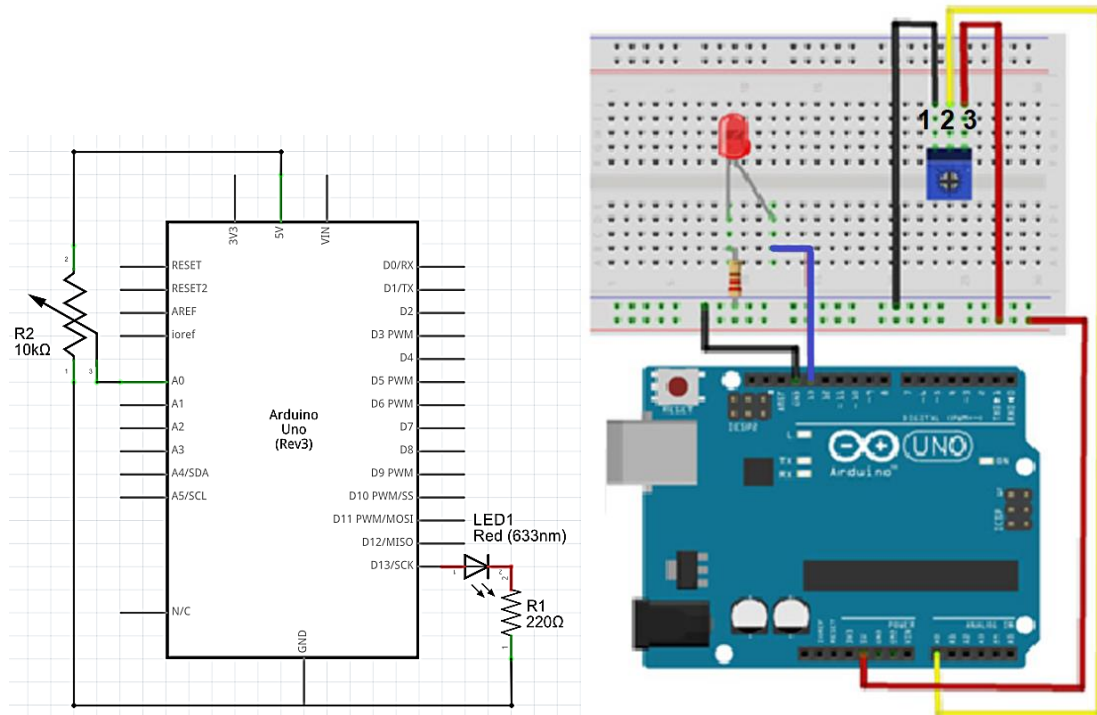


Figura 4.14. Lidhja e diodës led dhe e potenciometrit- trimer me Arduino Uno

6. Shkrimi dhe ngarkimi i programit për Arduino Uno R3

Arduino Uno R3 kontrollon vlerën e sinjalit në pinin e hyrjes analoge A0 duke përdorur instruksionin analogRead(). Vlera e lexuar paraqet një variabël me emrin simbolik sensorValue. Kjo variabël është shkruar si kohë vonesë në të dy instruksionet delay() që vijnë pas dy instruksioneve digitalWrite(), me të cilët ndryshohet gjendja e diodave led.

Më poshtë është dhënë programi për kontrollin e frekuencës së pulsimit të diodës led me potenciometër.

```

1 int sensorPin = A0; // Zgjedhja e pinit hyrjes për
// potociometrin.
2 int ledPin = 13; // Zgjedhja e pinit për lidhjen e
// diodës led.
3 int sensorValue = 0; // Deklarojmë një variabël për r
// uajtjen e vlerës së fituar nga
// potociometri.
4 void setup() { //
5 pinMode(ledPin, OUTPUT); // Pini ledPin konfigurohet
// si dalës.
6 } // Fundi i strukturës setup.
```

```

7 void loop() { // Fillimi i ciklit.
8   sensorValue =analogRead(sensorPin);
//Vlera e lexuar nga potenciometri vendoset në variablin me emrin simbolik
//"sensorValue".
9   digitalWrite(ledPin, HIGH); // Dioda led kyçet.
10  delay(sensorValue); // Dioda ndriçon aq milisekonda sa
// është vlera e marrë nga
// potenciometri.
11  digitalWrite(ledPin, LOW); // Dioda led shkyçet.
12  delay(sensorValue); // Dioda nuk do të ndriçoj për aq
// milisekonda sa është vlera e
// marrë nga potenciometri.
// Pastaj sërish kthehemi në fillim
// të ciklit.
//
13 } // Fundi i ciklit.
```

Pas shkrimit të programit, i njëjti ngarkohet në memorien e Arduino Uno R3 siç u shpjegua më parë.

7. Rezultatet nga realizimi i ushtrimit

Kur do të kyçet furnizimi për Arduino Uno R3 është i ndezur, me ndihmën e një kaçavide rregullohet rezistenca e potenciometrit nga minimumi në maksimum. Kur rezistenca është më e lartë atëherë vlera e sinjalit të hyrjes dhe koha e vonesës janë më të ulëtat dhe dioda led do të pulsojë me shpejtësinë më të madhe. Nëse koha e vonesës është tepër e vogël, është e mundur që të mos vërehet pulsimi i. Të theksojmë se koha maksimale e vonesës është 1023.

Komenti:_____

8. Bëni një ndryshim!

- Çfarë do të ndodhë nëse në instruksionet delay() variabla sensorValue shumëzohet me pesë?

Komenti:_____

- Çfarë do të ndodhë nëse pas leximit të vlerës së sinjalit analog të hyrjes në kodin programues futet instruksion i ri procent=map(sensorValue,0,1023,0,100) siç tregohet në shembullin 4.53?

Shembulli 4.53.

```

7 void loop() {
8   sensorValue =analogRead(sensorPin);
9   procent = map(sensorValue,0,1023,0,100);
10  digitalWrite(ledPin, HIGH);
11  delay(procent);
12  digitalWrite(ledPin, LOW);
```

```
13   delay(100-procent);  
14 }
```

Komenti: _____

4.5.3. Ushtrim praktik: Rregullimi i intensitetit të dritës së diodës led

1. Qëllimi i ushtrimit

Nxënësit do të përdorin njërin nga pinat e daljes analoge të Arduino Uno R3 dhe në mënyrë softuerike do të ndryshojnë madhësinë e tensionit të daljes. Ndryshimet në tension do të monitorohen nëpërmjet përdorimit të një diode led, me ç'rast intensiteti i dritës që ndryshojë gradualisht nga minimumi drejt maksimumit dhe anasjelltas.

2. Koha e realizimit: 1 orë mësimore

3. Komponentët e nevojshëm për realizimin e këtij ushtrimi janë:

- Mikrokompjuter Arduino Uno R3
- protoplakë
- dioda led
- rezistor me rezistencë $R1=220\Omega$
- dy tela për lidhje (qarqe të shkurtëra)
- kompjuter me mjedisin zhvillimor Arduino të instaluar

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- modulimin në gjerësi i impulseve, rëndësia e tij për daljet analoge të Arduino dhe numri i vlerave të mundshme
- instruksionet për punë me pinat e daljes analog të mikrokompjuterit në pllakë Arduino
- strukturën for për përsëritjen e cikleve

5. Përshkrimi i skemës elektrike dhe mënyra e lidhjes

Në figurën 4.15. janë treguar skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

Qarku është shumë i thjeshtë. Katoda e diodës led, përmes rezistencës R1, duhet të lidhet me shiritin për tokëzim, ndërsa anoda me pinin e nëntë të Arduino Uno R3. Në këtë ushtrim nuk mund të përdorim pinin e trembëdhjetë si në ushtrimet e mëparshme, sepse në të nuk mund të rregjistrohen vlerat analoge.

Tensioni i daljes, me të cilin kontrollohet intensiteti i dritës së diodës led, e shtojmë duke e përdorur instruksionin analogWrite() me ç'rast në kllapa të mesme vendoset vlera analoge. Vlera minimale analoge është 0, ndërsa maksimale është 255.

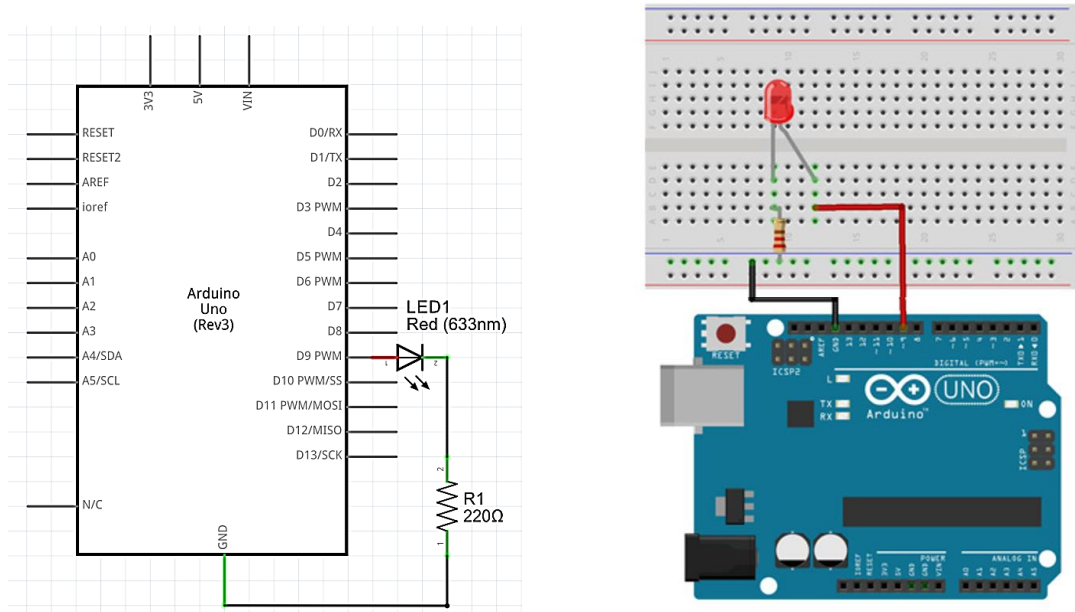


Figura 4.15. Lidhja e diodës led në pinin PWM të Arduino Uno R3

Programi për rregullimin e intensitetit të dritës përdor dy struktura "for" për përsëritjen e ciklit. Në strukturën e parë for, variablën e zmadhojmë për pesë dhe cikli përsëritet derisa nuk marrim vlerën 255. Në strukturën e dytë for, variabla zvogëlohet për pesë derisa nuk arrihet vlera 0.

```

1  int ledPin = 9; // Dioda led është e lidhur në pinin
                        // numër 9.
2  void setup() {   // Nuk ka konfigurim të pinave
3  } //
4  void loop() {   // Fillimi i ciklit.
5  for(int fadeValue = 0; fadeValue <= 255; fadeValue +=5)
  {

```

// Për strukturën for, deklarojmë variabël me emrin simbolik "fadeValue" dhe vlerë
// fillestare zero. Atë e rrisim për 5 derisa të marrim një numër të barabartë me 255.
// Pas çdo rritjeje ose zvogëlim pritet 30 milisekonda.

```

6  analogWrite(ledPin, fadeValue); // E zmadhojmë intensitetin e
7  delay(30); // dritës
8  } //
9  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5)
  {

```

//E zvogëlojmë intensitetin e dritës. Variablin e zvogëlojmë për pesë derisa të marrim
//vlerën zero. Pastaj kthehemi në fillim të ciklit.

```

10 analogWrite(ledPin, fadeValue);
11 delay(30);
12 }
13 }

```

Ashtu si programet e mëparshme, edhe këtë program e ngarkojmë në Arduino Uno R3 me procedurën e mëposhtme:

- (1) E lidhim Arduino Uno R3 me kompjuterin nëpërmjet USB kablos
- (2) Në menynë kryesore zgjedhim opsionin Tools dhe zgjedhim llojin e mikro-kompjuterit Arduino dhe portën serike.
- (3) E përkthejmë programin në gjuhën e makinës duke përdorur veglën Verify/Compile.
- (4) Dhe në fund e ngarkojmë programin duke shtypur veglën Upload.

Menjëherë pas ngarkimit të programit, qarku elektrik do të fillojë të punojë gjegjësisht intensiteti i dritës i diodës led do të fillojë të zvogëlohet dhe rritet në mënyrë alternative.

Komenti: _____

- Çfarë do të ndodhë nëse e ndryshojmë hapin e rritjes ose zvogëlimit të vlerës analoge nga pesë në ndonjë vlerë tjetër?

Komenti: _____

- Çfarë do të ndodhë nëse rrisim kohën e vonesës në instruksionin delay()?

Komenti: _____

4.5.4. Ushtrim praktik: Fotorezistori për kontrollin e intensitetit të dritës

1. Qëllimi i ushtrimit

Në këtë ushtrim do të ndryshojmë intensitetin e dritës së një diode led duke përdorur një fotorezistor. Gjegjësisht, dioda led do të shërbejë si tregues i ndriçimit të mjedisit në të cilin është vendosur Arduino Uno R3 së bashku me fotorezistorin. Gjithashtu në këtë ushtrim nxënësit do të mësojnë se çfarë do të thotë dhe si kryhet kalibrimi i senzorit.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Mikrokompjuter Arduino Uno R3
- protoplakë
- diodë led
- fotorezistor
- rezistorë me rezistencë $R1=220\Omega$ dhe $R2=10K\Omega$
- tela për lidhje (qarqe të shkurtëra)
- kompjuter me mjedisin zhvillimor Arduino të instaluar

Fotorezistori është element gjysmëpërçues, rezistenca elektrike e të cilit ndryshon nën ndikimin e dritës që bie mbi sipërfaqen e tij. Për shkak të kësaj vetie

gjen zbatim si sensor elektronik. Para se të integrojmë fotorezistorin në qark, me ndihmën e një multimetri, mund të matim rezistencën e tij duke ndryshuar intensitetin e dritës. Kur nuk ka dritë, rezistenca e tij është më e madhe. Sa më shumë të afrojmë fotorezistorin afër burimit të dritës, aq më shumë zvogëlohet rezistenca e tij.

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- karakteristikat themelore të fotorezistorëve
- diagramin e pinave të Arduino Uno R3
- instruksionet për punë me pinat analog dhe brezin e vlerave të mundshme
- kuptimin e strukturës bazë if
- zbatimin e instruksionit map() për të ndryshuar brezit të vlerave.

5. Përshkrimi i skemës elektrik dhe mënyra e lidhjes

Në figurën 4.16. tregohen skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

- (1) E vendosim fotorezistorin në protoplakë me ç'rast duhet të kemi kujdes që terminalet e tij t'i vendosim në pika kontakti që u përkasin përçuesve të ndryshëm. Me një qark të shkurtër të kuq njërin skaj të fotorezistorit e lidhim me shiritin e furnizimit. Skajin tjetër të fotorezistorit, nëpërmjet rezistorit R2 dhe qarkut të shkurtër të zi e lidhim me shiritin e tokëzimit. Në fakt, fotorezistori dhe rezistenca R2 formojnë një ndarës të tensionit. Me qarkun e shkurtër të kaltër pika e kontaktit ndërmjet fotorezistorit dhe rezistencës R2 lidhet me hyrjen analoge A0.
- (2) Ashtu si në ushtrimin e mëparshëm, katoda e diodës led, nëpërmjet rezistorit R1 dhe qarkut të shkurtër të zi, e lidhim me shiritin për tokëzim. Me qarkun e shkurtër të gjelbër e lidhim anodën me pinin e nëntë me shenjën PWM, që do të thotë se është dalje analoge.
- (3) Pinat GND dhe 5V të Arduino Uno R3 i lidhim me shiritat për furnizim dhe tokëzim.

Arduino Uno R3 mat tensionin e fituar nga ndarësi i tensionit dhe konverton vlerat e matura i konverton proporcionalisht në vlera të numrave të plota në brezin nga 0 deri në 1023. Vlera e plotë 1023 korrespondon me tensionin prej 5V, ndërsa zero me tensionin prej 0V. Por, për shkak të rënies së tensionit në rezistencën R2, tensioni i hyrjes në pinin A0 asnjëherë nuk do të arrijë vlerën 5V, ndërsa as sinjali i hyrjes nuk do të arrijë vlerën e tij maksimale. Gjithashtu, fotorezistori është i projektuar që të zbuloj ndryshimet në intensitetin e dritës në një brez shumë më të gjerë sesa janë ndryshimet e dritës në kushte të ambientit të mbyllur. Për këto arsye është e nevojshme të bëhet kalibrimi i fotorezistorit. Kalibrimi është i domosdoshëm që sensori të funksionojë me saktësi. Në teknikën e matjeve me nocionin kalibrim i

i një instrumenti matës nënkuptojmë përcaktimin e devijimit të vlerës së lexuar në instrumentin matës në raport me vlerën e vërtetë (vlera e etalonit). Në lidhje me Arduino Uno R3 dhe fotorezistorin, devijimi paraqet ndryshimin midis vlerës maksimale ose minimale të sinjalit hyrës që e jep vetë Arduino dhe vlerës maksimale ose minimale që mund ta detektojë vetë fotorezistori. Vlerat përfundimtare në brezin nga 0 deri në 1023 nuk do të arrihen asnjëherë, që automatikisht do të thotë se dioda led gjithmonë do të ndriçoj, por asnjëherë nuk do të arrijë maksimumin ose minimumin e saj. Që të eliminohet një devijim i tillë duhet të rregullohen vlerat maksimale dhe minimale të sensorit që në sekondat e parë pas fillimit në punë të qarkut elektrik. Kjo arrihet duke afruar dhe larguar dorën nga sensorin, me ç'rast softueri i Arduino përzgjedh brezin e vlerave maksimale dhe minimale.

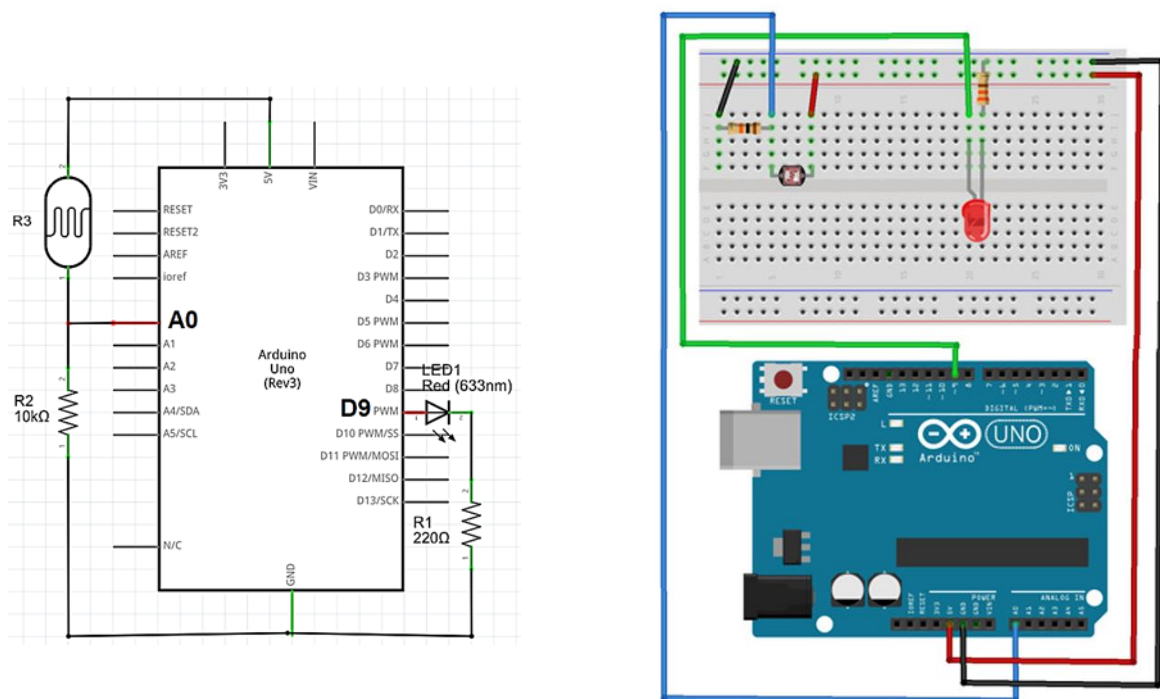


Figura 4.16. Lidhja e diodës led dhe fotorezistorit me Arduino Uno R3

6. Shkrimi dhe ngarkimi i programi për Arduino Uno R3

Kalibrimi kryhet vetëm një herë, ai nuk përsëritet dhe për këtë arsye i njëjti shkruhet si kod programues brenda strukturës së detyrueshme setup. Për shkak të nevojës për kalibrim, janë shtuar dy variabla të rej me emra simbolikë lightLow dhe lightHigh. Variabla lightHigh rritet vazhdimisht nga zero deri në vlerën maksimale që mund të detektojë fotorezistori në një mjedis të caktuar fizik. Variabla LightLow zvogëlohet vazhdimisht nga 1023 deri në vlerën minimale të fotorezistorit. Instruksioni përshtatLightLevel=map(Light Level, lightHigh, lightLow, 0, 255) kryen konvertimin e brezit të hyrjes me vlera minimale dhe maksimale të kalibruara në brezin dalës të vlerave, nga 0 deri në 255.

Më poshtë është dhënë programi me komente për instruksionet dhe strukturat e tij.

```
//Zakonisht fillojmë me zgjedhjen e pinave për lidhjen e fotorezistorit
//diodës led.
1  const int sensorPin = 0;      //  Sensori është i lidhur me pinin analog zero
                                   //
2  const int ledPin = 9;        //  Dioda led është e lidhur me pinin e nëntë në dalje
                                   //  të të cilit ka impulse të moduluara në gjerësi.
                                   //
3  int lightLevel;              //  Variabla i rruan të dhënat e fituara nga
                                   //  fotorezistori.
4  int përshtatlightLevel;     //  Variabla me emrin simbolik përshtatlightLevel
                                   //  përdoret gjatë ndryshimit të brezit nga hyrja në
                                   //  dalje.
5  int lightHigh = 0;          //  Variabli lightHigh paraqet vlerën maksimale pas
                                   //  kryerjes së kalibrimit.
                                   //
6  int lightLow = 1023;        //  Variabli lightLow paraqet vlerën minimale pas
                                   //  kryerjes së kalibrimit
                                   //
7  void setup()
8  {
9  pinMode(ledPin, OUTPUT);    //  Konfigurimi i pinit të daljes për diodën
                                   //  led.
// Kalibrimi kryhet në pesë sekondat e parë nga fillimi i ekzekutimit të .
// programit.
   while (millis() < 5000) {    //  Me instruksionin millis() matet koha.
                                   //
// Lexohet vlera pinit analog analogPin (fotorezistori) dhe fillon kalibrimi.
//
   lightLevel = analogRead(sensorPin);
// Variabla lightLow merr vlerë që korrespondon me intensitetin minimal të dritës në
// mjedisin e dhënë.
   if (lightLevel < lightLow) {
       lightLow = lightLevel;
   }
//Variabla lightHigh merr vlerë që korrespondon me intensitetin maksimal të dritës në
// mjedisin e dhënë
   if (lightLevel > lightHigh) {
       lightHigh = lightLevel;
17  }
18  }
19  }
20  void loop()
21  {
// Lexohet vlera e tensionit të pinit sensorPin (fotorezistori).
```

```

22   lightLevel = analogRead(sensorPin);
    // Kalimi nga njëri brez në tjetrin është linear.
23   përshtatlightLevel = map(lightLevel, lightHigh, lightLow, 0, 255);
24   analogWrite(ledPin, përshtatlightLevel);
    // Sinjali që korrespondon me ndriçimin e fotorezistorit e eksiton diodën
    //led.
25 } Fundi i strukturës loop //

```

Pasi të shkruajmë programin, i njëjti ngarkohet në Arduino Uno R3 sipas procedurës së shpjeguar tashmë në shembujt e mëparshëm.

7. Rezultatet nga realizimi i ushtrimit

Kur qarkun elektrik e lidhim me furnizimin, menjëherë në pesë sekondat e parë, kryejmë kalibrimin duke afuar dhe larguar dorën drejt sensorit. Më pas kontrollojmë nëse dioda ndriçon më shumë kur mbi fotorezistor bie më shumë dritë sesa kur fotorezistori mbulohet me dorë ose objekt tjetër.

Komenti: _____

8. Bëni një ndryshim!

- Si konfirmim i gjithçkaje që është thënë për kalibrimin, mund t'i shfaqim rezultatet e fituara në ekranin e monitorit serik. Për këtë qëllim, duhet të definojmë shpejtësinë e transmetimit në strukturën setup duke përdorur instruksionin Serial.begin(9600) dhe të futim instruksionet Serial.println(lightLow) dhe Serial.println(lightHigh) pasi mbarimit të kalibrimit dhe Serial.println (lightLevel) përpara hartimit, gjegjësisht para ndryshimit të vlerave të brezit hyrës në brezin dalës.

Komenti: _____

- Në vend të një fotorezistorit, mund të vendosim çdo lloj sensori dhe vlerat e tij do të reflektohen në dritën që e emeton dioda led. Gjithashtu, në vend të diodës led mund të vendosim një piezo sensor i cili do të lëshojë tone ose motor, shpejtësia e të cilit do të varet nga ndriçimi i fotorezistorit. Këto ndryshime në kodin programor kërkojnë njohuri të mira të grupit të instruksioneve të mikrokompjuterit Arduino.

Komenti: _____

4.5.5. Ushtrim praktik: Lidhja e Arduino Uno R3 me servo motor

1. Qëllimi i ushtrimit

Qëllimi i këtij ushtrimi praktik është zhvendosja e servo motorit për 180 shkallë. Për programimin e servo motorit dhe kontrollin e tij, nxënësit duhet të thirrin bibliotekën Servo.h, me të cilën u njohëm në njësinë mësimore 4.9. Instruksionet për punën me bibliotekat në pjesën teorike të librit. Ky është ushtrimi i parë praktik i

programimit të Arduino Uno R3, nga ushtrimet e realizuara deri tani, që përdor një bibliotekë.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Mikrokompjuter Arduino Uno R3
- protoplakë
- servo motor
- tela lidhës (qarqe të shkurtëra)
- kompjuter me mjedisin zhvillimor Arduino të instaluar

Parimi i funksionimit të servo motorit është paraqitur në figurën 4.24. në njësinë mësimore 4.9. Instruksione për punën me bibliotekat në pjesën teorike të librit.

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- diagramin e pinave të Arduino Uno R3
- parimin e funksionimit të servo motorit
- instruksionet që janë pjesë e bibliotekës Servo.h
- kuptimin e strukturës if për përsëritjen e ciklit

5. Përshkrimi i skemës elektrik dhe mënyra e lidhjes

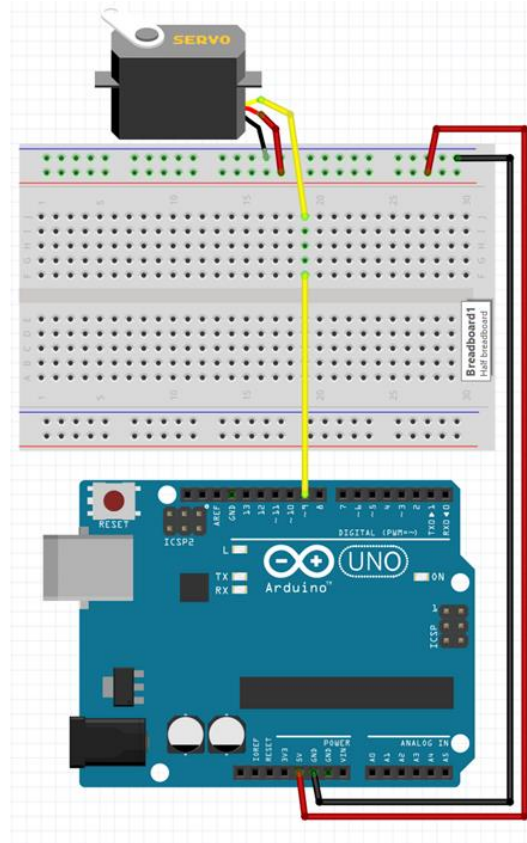
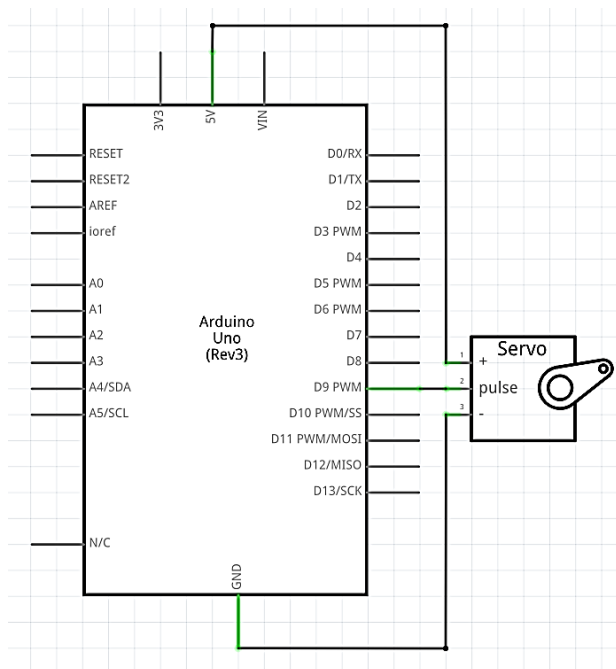


Figura 4.17. Lidhja e servo motorit me Arduino Uno R3

Në figurën 4.17. tregohet skema funksionale dhe e montimit për realizimin e këtij ushtrimi. Skema e montimit është shumë e thjeshtë. Servo motori ka tre tela.

Teli i zi është për tokëzimin, i kuqi është për furnizimin dhe teli i verdhë është për kontrollin e motorit dhe përmes tij servo motori lidhet me pinin e 9-të të Arduino Uno R3. Të rikujtojmë, vetëm pini i 9 dhe 10 ka mbështetje për lidhjen e servo motorit. Duke vendosur një kondensator 100µF në pinat për tokëzim dhe furnizim, mund të shmangen ndryshimet e shpejta të tensionit. Servo motori është pajisje dalëse. Në këtë ushtrim, nuk ka pajisje hyrëse dhe ne do të kontrollojmë servo motorin thjeshtë me softuer, vetëm me instruksione.

6. Shkrimi dhe ngarkimi i programit për Arduino Uno R3

Më poshtë është dhënë programi me të cilin servo motori do të rrotullohet për 180 shkallë, nga e majta në të djathtë, me hapa prej dy shkallësh sa herë që struktura for përsëritet.

```

1  #include <Servo.h>           // Kyçja e bibliotekës për kontrollin e servo
                                // motorit.
2  Servo myServo;              // Definojmë servo objekt me emrin simbolik
                                // mySevo.
                                // Në këtë objekt do të veprojnë komandat e
3  int position;               // bibliotekës Servo.
                                // Variabla position përmban vlerën e këndit për të
                                // cilin duhet të rrotullohet servo motori.
4  void setup()                //
5  {                            // Fillimi i funksionit setup.
6    myServo.attach(9);        // Pinin e 9-të e konfigurojmë si pin në të cilin do
                                // të lidhet servo motori.
7  }                            // Fundi i funksionit setup.
8  void loop()                 //
9  {                            // Fillimi i funksionit loop.
                                //Me strukturën for, gradualisht e rrisim këndin e rrotullimit për dy shkallë,
                                // duke filluar nga zero deri në vlerën maksimale prej 180 shkallësh.
10  for(position = 0; position < 180; position += 2) {
11    myServo.write(position);
                                //Gjatë çdo përsëritje të strukturës for, servo motori zhvendoset për dy
                                //shkallë në të djathtë.
12    delay(20);                // Koha e shkurtër e vonesës shërben për
                                // stabilizimin e pozitës para rrotullimit të ri.
                                //
13  }                            // Fundi i strukturës for.
14  }                            // Fundi i funksionit loop.

```

Bibliotekën e kyçim duke klikuar në menynë kryesore Sketch → Import Library, me ç'rast hapet menyya rënëse. Biblioteka Servo.h është bibliotekë standarde në përbërje të mjedisit zhvillimor të Arduinos. Programin e fusim me

procedurën: lidhja e Arduinos me kompjuterin → zgjedhja e mikrokompjuterit Arduino dhe portës serike → verifikimi → ngarkimi.

7. Rezultatet nga realizimi i ushtrimit

Menjëherë pas vendosjes së programit qarku elektrik do të fillojë të punojë, kështu që servo motori do të fillojë të rrotullohet nga e majta në të djathtë, për gjysmë rrotullimi. Kur do të rrotullohet për 180 shkallë, servo motori ndalon.

Komenti: _____

8. Bëni një ndryshim!

- Në programin e mësipërm mund të shtohet edhe një strukturë tjetër for me të cilën këndi i rrotullimit që do të kthehet prapa, nga 180 shkallë deri në zero. Në këtë rast nga vlera e këndit duhet të zbriten dy. Në këtë mënyrë, servo motori do të rrotullohet përpara-prapa.

```

14 for(position = 180; position >0; position -= 2) {
15     myServo.write(position);
16     delay(20);
17 }
```

Komenti: _____

- Për kontrollin e këndit të rrotullimit të servo motorit mund të përdoret edhe një potenciometër, si pajisje hyrëse e lidhur në njërin nga pinat analog. Kuptohet, në atë rast, duhet të bëhen ndryshime më të mëdha në programin e mësipërm. Në strukturën e domosdoshme loop do të duhet të përdoren instruksionet e mëposhtme: analogRead() për leximin e vlerës së potenciometrit, map() për ndryshimin e brezit të hyrjes në atë të daljes (nga 0-1023 në 0-255) dhe myServo.write() për lëvizjen e servo motorit.

Komenti: _____

4.5.6. Ushtrim praktik: Lidhja e Arduino Uno R3 me ekran LCD

1. Qëllimi i ushtrimit

Qëllimi i këtij ushtrimi është shfaqja e tekstit "Sistemet kompjuterike" në një ekran LCD dhe koha e kaluar e shprehur në milisekonda.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- mikrokompjuter Arduino Uno R3
- protopllakë
- ekran LCD, modeli GDM1602K
- potenciometër - trimer me rezistencë 10KΩ

- xjtela për lidhje (qaerqe të shkurtëra)
- kompjuter me mjedisin zhvillimor Arduino të instaluar

Në njësinë mësimore 4.9. Instruksione për punë me bibliotekat në pjesën teorike të librit janë përshkruar diagrami i pinave të një ekranit LCD dhe disa instruksione në kuadër të bibliotekës LiquidCrystal. Gjatë përzgjedhjes së ekranit LCD duhet të kontrollojmë nëse kontrolluesi në përbërje të ekranit është kompatibil me bibliotekën LiquidCrystal, gjë për të cilën do të na ndihmojë dokumentacioni tekniko-teknologjik i ekranit. Për shembull, kontrollorë standardë kompatibil me këtë bibliotekë janë HD44780 ose ST7066U. Potenciometri-trimer shërben për rregullimin e kontrastit.

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- diagramin e pinave të Arduino Uno R3
- diagramin e pinave të ekranit LCD, modeli GDM1602K
- instruksionet për përdorimin e ekranit LCD

5. Përshkrimi i skemës elektrik dhe mënyra e lidhjes

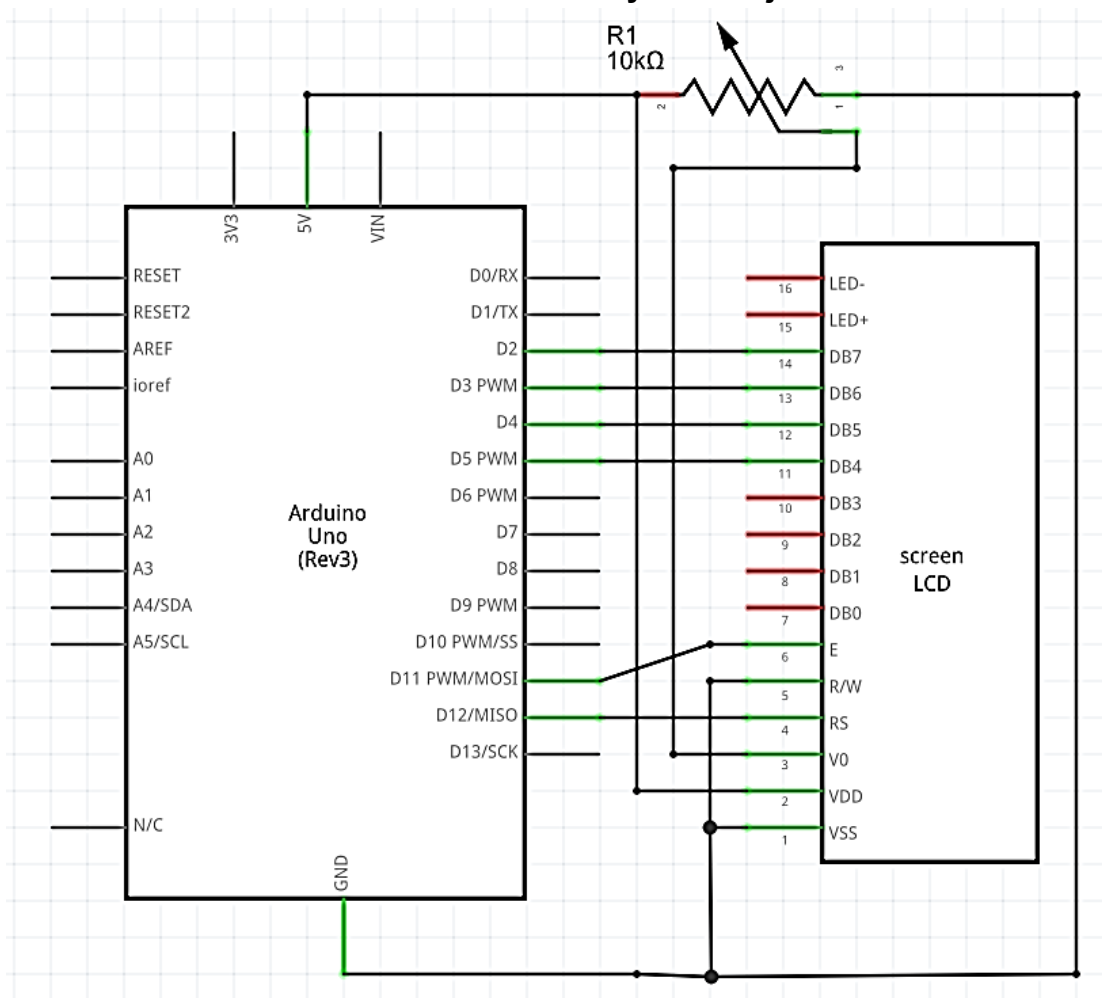


Figura 4.18. Skema elektrik për lidhjen e ekranit LCD me Arduino Uno R3

Pinat e ekranit LCD duhet të lidhen me pinat e Arduino Uno R3 si më poshtë:

Terminale të ekranit LCD	Pina të Arduino R3
(1) RS	pinin digjital 12
(2) Enable pini me	pinin digjital 11
(3) Pini D4 me	pinin digjital 5
(4) Pini D5 me	pinin digjital 4
(5) Pini D6 me	pinin digjital 3
(6) Pini D7 me	pinin digjital 2
(7) R/W	me pinin për tokëzim
(8) V _{SS}	me pinin për tokëzim
(9) V _{CC}	me pinin për furnizim prej 5V

Në figurën 4.18. është treguar mënyra e lidhjes së ekranit LCD me Arduino Uno R3.

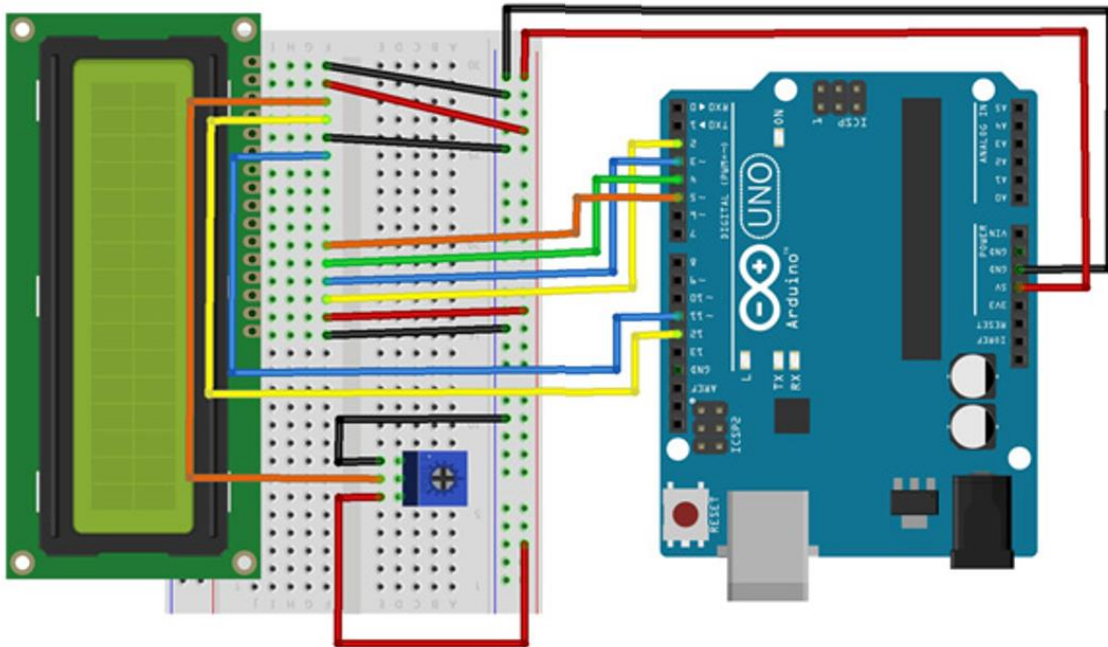


Figura 4.34. Skema e montimit për lidhjen e ekranit LCD me Arduino Uno R3

Pini V_{EE} është i lidhur me terminalin e dytë të potenciometrit nëpërmjet qarkut të shkurtër portokalli. Terminali i parë dhe i tretë të potenciometrit janë të lidhur me shiritat për furnizim dhe tokëzim.

6. Shkrimi dhe ngarkimi i programi për Arduino Uno R3

Pason përpunimi i softuerit në mjedisin zhvillimor.

```

1  #include <LiquidCrystal.h>    // Kyçet biblioteka për punë me ekranin LCD, e
                                // cila është bibliotekë standard
                                //
2  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
   // Deklarojmë pinat e Arduino Uno R3 për lidhje me ekranin.
3  void setup() {                //

```

```

4   lcd.begin(16, 2);           // Fillon komunikimi me ekranin LCD nëpërmjet
   //                          // vendosjes së numrit të kolonave dhe
   //                          // rreshtave prej simboleve.
5   lcd.print("Sistemet kompjuterike"); //
   Në ekran shtypet teksti. }
6   void loop() {
7
8   lcd.setCursor(0, 1);       // Zgjidhet fusha aktive prej gjithsejt 32,
   //                          // përmes zgjedhjes së kolonave dhe
   //                          // rreshtave. Të theksojmë, kolonat dhe
   //                          // rreshtat numërohen duke filluar nga zero.
9   lcd.print(millis()/1000);  // Në ekran shtypet koha e shprehur në
   //                          // sekonda, e kaluar nga ristartimi i fundit i
   //                          // Arduino Uno R3
10  }

```

Pasi ta shkruajmë programin, të njëjtin e fusim në memorien e Arduino Uno me procedurën: lidhjen e Arduino Uno R3 me kompjuterin → zgjedhjen e mikrokompjuterit Arduino dhe portës serike → verifikimi → ngarkimi.

7. Rezultatet nga realizimi i ushtrimit

Menjëherë pas futjes së programit qarku elektrik do të fillojë të punojë, kështu që në ekran do të paraqitet teksti “Sistemet Kompjuterike” dhe koha e matur.

Komenti: _____

8. Bëni një ndryshim!

Në vetë mjedisin zhvillimor ekzistojnë një numër i madh i programeve të gatshme për ekranet LCD. Ato mund të hapen dhe të analizohen duke klikuar File → Examples → LiquidCrystal. Do të numërojmë disa prej tyre: zhvendosja automatike e tekstit, pulsimi i tekstit, shtypja e formave tekstuale, zgjedhja e drejtimit të tekstit, etj.

Komenti: _____

5. Ushtrime praktike për punë me mikrokompjuterin në pllakë Raspberry Pi dhe programimi i tij

5.1. Masat për mbrojtje dhe siguri gjatë punës me Raspberry Pi

Për realizim të suksesshëm dhe të sigurt të ushtrimeve praktike duhet të respektohen rregullat e përgjithshme të theksuara në njësinë mësimore 1. Masat për mbrojtje dhe siguri gjatë punës në pjesën praktike të këtij libri. Por, gjithashtu duhet të njihen edhe karakteristikat specifike të pajisjeve elektronike të cilat zakonisht ndodhen në dokumentacionin tekniko-teknologjik të tyre. Shkaqet më të shpeshta të dëmtimit të Raspberry Pi janë furnizimi joadekuat dhe mosnjohja e karakteristikave të komponentëve hyrëse-dalëse të lidhura. Masat më të rëndësishme të mbrojtjes gjatë punës me Raspberry Pi janë:

- Zgjedhja e burimit të furnizimit është shumë e rëndësishme. Përdoret një adapter USB me tension maksimal të furnizimit prej 5V. Të theksojmë se Raspberry Pi 4 Model B përdor një adapter me konektor USB-C, ndërsa modelet tjera përdorin mikro USB.
- Rrymën që duhet ta sigurojë burimi i furnizimit varet nga modeli i Raspberry Pi dhe konsumi i energjisë elektrike të komponentëve hyrëse-dalëse të lidhura. Rryma maksimale e lejuar është një nga 1.5A deri në 2A. Për shkak të konsumit më të lartë të energjisë elektrike të Raspberry Pi 4 Modeli B, i nevojitet rrymë me intensitet prej 3A.
- Bazamenti GPIO përmban katër pina për furnizim dhe tetë pina për tokëzim. Dy nga pinat e furnizimit kanë tension konstant prej 5V, ndërsa dy të tjerët 3.3V.
- Tensioni maksimal i lejuar për pinat hyrëse për GPIO është 3.3V.
- Përpara fillimit të çdo montimi, Raspberry Pi duhet të shkëputet nga burimi i tij i furnizimit.
- Meqenëse Raspberry Pi është i ndjeshëm ndaj elektricitetit statik, para fillimit të montimit është e nevojshme të preket ndonjë sipërfaqe metalike. Në këtë mënyrë vjen deri te shkarkimi i trupi të njeriut.
- Përpara se të kyçet burimi i furnizimit, detyrimisht të kontrollohen nëse të gjithë komponentët elektronikë janë të lidhur siç tregohet në skemën funksionale ose të montimit.
- Pinat asnjëherë nuk duhet të lidhen me njëri-tjetrin sepse mund të vijë deri te dëmtimi i tyre.

- Për të rregulluar rrymën e punës duhet të përdoren rezistorë. Për shembull, nëse diodën led e lidhim direkt me pinin për furnizim 5V, do të vij deri te nxehja dhe do të dëmtohet.
- Për të lidhur motorë ose ngarkesa të tjera induktive, të përdoren urat H (angl. motor drive).

5.2. Ushtrim praktik: Instalimi i sistemit operativ Raspbian për Raspberry Pi

5.2.1. Shkarkimi i softuerit NOOBS për instalim

Për të instaluar sistemin operativ Raspbian do të përdorim softuer me emrin simbolik Softuer i ri nga kutia (angl. NOOBS-New Out Of the Box Software). Ky është softuer special që mundëson zgjedhjen e njërit prej disa sistemeve operative dhe me instalim automatic vetëm me disa klikime të mausit. Softueri për instalim ruhet në një kartelë micro SD. Tashmë kemi përmendur se Raspberry Pi nuk ka memorie të përhershme të integruar, por për ruajtjen e të dhënave dhe të sistemit operativ përdoret **kartelë SD me minimum 16 GB memorie**.

Për të shkarkuar softuerin NOOBS, në web kërkuesin duhet të shkruhet rpf.io/download. Në faqen që do të hapet, në kategorinë Download, fillimisht shtypim në NOOBS me logon e Raspberry, ndërsa më pas shtypim (click) në Download Zip, që ndodhet nën "NOOBS offline and network install".



CFigura 5.1. Faqja e internetit për shkarkimin e dokumentit instalues për NOOBS

E vendosim kartelën micro SD në slotin e duhur të kompjuterin personal. Të rikujtojmë se nëse nuk është kartela e re, duhet të formatohet por duhet pasur kujdes për të dhënat që janë të ruajtura në të. Më pas në dosjen Shkarkime (angl. Download) e kërkojmë skedarin e kompresuar të instalimit. Ky skedar

njihet si skedar arkivi (angl. archive) dhe përmban një numër të madh dokumentesh të veçanta. Me një klikim të dyfishtë e “depaketojmë” dokumentin e kompresuar, i selektojmë të gjithë dokumentet në të dhe i transferojmë duke i kopjuar në kartelën micro SD.

5.2.2. Instalimi i sistemit operativ Raspbian

Pastaj **kartelën micro SD** me softuerin e instaluar NOOBS **duhet ta largojmë nga kompjuteri personal dhe ta fusim atë në slotin e mikrokompjuterit Raspberry Pi**. Kur kartelën për here të parë do ta lidhim me Raspberry Pi, shfaqet menyja NOOBS e paraqitur në Figurën 5.2. Me maus **selektojmë** kuadratin e sistemit operativ që duam ta instalojmë, që në këtë rast është **Raspbian**. Do të vërejmë se ikona Install do të marrë ngjyrë, që do të thotë se sistemi operativ është gati për instalim. [8]

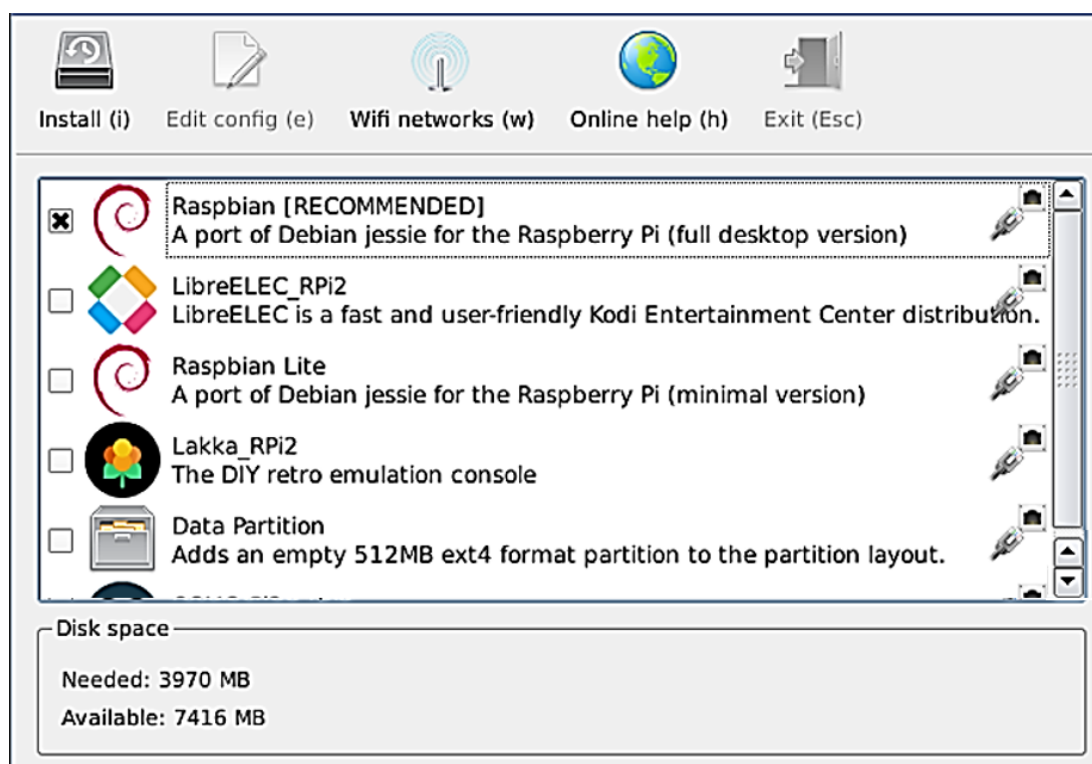


Figura 5.2. Zgjedhja e sistemit operativ Raspbian

Duke shtypur ikonën Install, fillon procesi i instalimit me ç’rast fshihen të gjitha të dhënat nga kartela micro SD, përveç softuerit NOOBS. Vetë instalimi mund të zgjasë nga 10 deri në 30 minuta. Pas përfundimit të instalimit shtypim (angl. click) OK dhe pajisja Raspberry Pi do të ristartohet. Vetëm gjatë hapjes së parë të sistemit operativ, paraqiten boot mesazhet të cilët zgjasin një deri në dy minuta. Përfundimisht paraqitet sipërfaqja punuese e Raspbian dhe në të dritarja me të cilën njoftohet konfigurimi i sistemit operativ. Klikojmë në Next. Ngjashëm si gjatë instalimit të sistemeve operative Windows ose Linux, ashtu edhe gjatë

instalimit të sistemit operativ Raspbian zgjedhim: shtetin, gjuhën, zonën kohore, fjalëkalimin, zgjedhjen e rrjetit Wi-Fi dhe përditësimin e sistemit operativ. Në dritaren e fundit, duhet të shtypim Reboot, pas së cilës sistemi do të ristartohet dhe konfigurohet me ndryshimet e bëra.

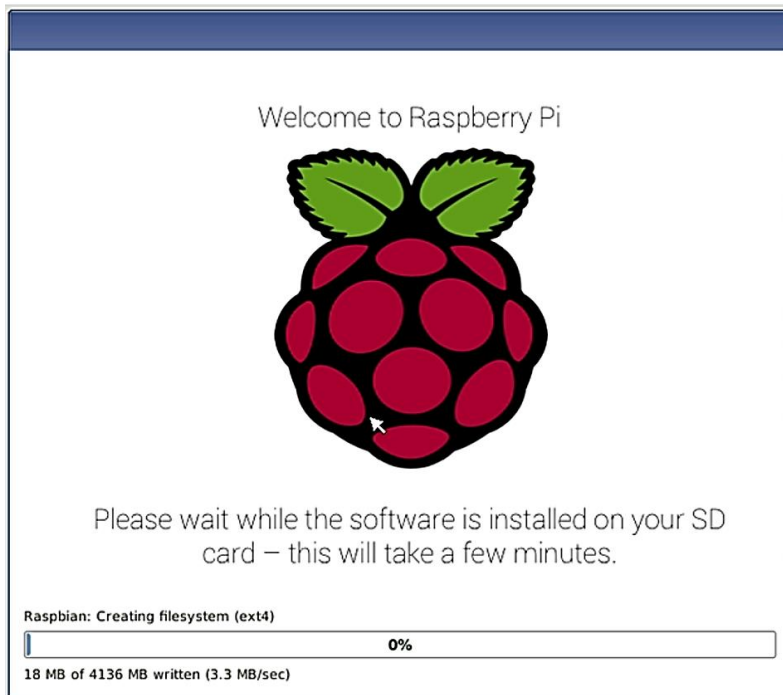


Figura 5.3. Instalimi i sistemit operativ Raspbian

Sipërfaqja punuese e sistemit operativ Raspberry duket shumë e ngjajshme me sipërfaqen punuese të sistemit operativ Windows.

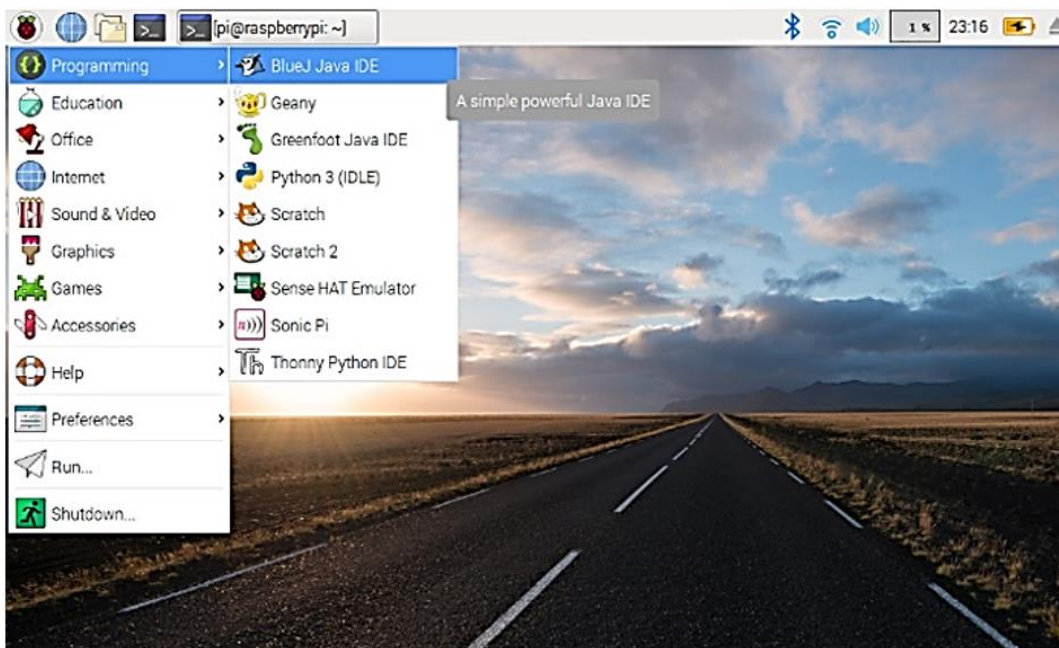


Figura 5.4. Pamja e sipërfaqes punuese të sistemit operativ Raspbian

Në këndin e sipërm të djathtë ndodhet shiriti me vegla për: Bluetooth, zgjedhjen e rrjetit, volumin e zërit, monitorimin e procesorit, ora, kyçjen dhe shkyçjen e pajisjeve USB. Në këndin e sipërm majtas ndodhet Raspberry dhe duke klikuar mbi të hapet menyja rënëse me kategori të shumta. Për shembull, kategoria **Programing** përmban programe zhvillimore mes të cilëve edhe **Thonny Python IDE**. Për kërkim në internet, në kategorinë Internet zgjedhim Chromium Web Browser. Për të punuar me dokumente dhe skedarë, na duhet File Manager dhe për ta aktivizuar atë zgjedhim kategorinë Accessories. Në kategorinë Office ndodhet programi për përpunimin e tekste LibreOffice Writer. Mund të instalohet softueri shtesë duke klikuar në Recommended Software në kategorinë Preferences. Për të shtuar harduer të ri (për shembull, softuer për të punuar me kamerë, angl. Raspberry Pi Camera Module) ose për t'u lidhur në rrjet (angl. VNC- Virtual Network Computer), mund të përdoret opsioni Interface, deri te i cili vijmë duke shtypur (click) në Raspberry Pi Configuration në kategorinë Preferences.

5.3. Ushtrime praktike për programimin e Raspberry Pi 3B+ në gjuhën programuese Python

Për realizimin me sukses të ushtrimeve praktike, janë të nevojshme këto paranjohuri: instalimi i sistemit operativ Raspbian, programimi në gjuhën programuese Python, shfrytëzimi i bibliotekës GPIOZERO dhe lidhja dhe karakteristikat e pajisjeve bazë hyrëse-dalëse.

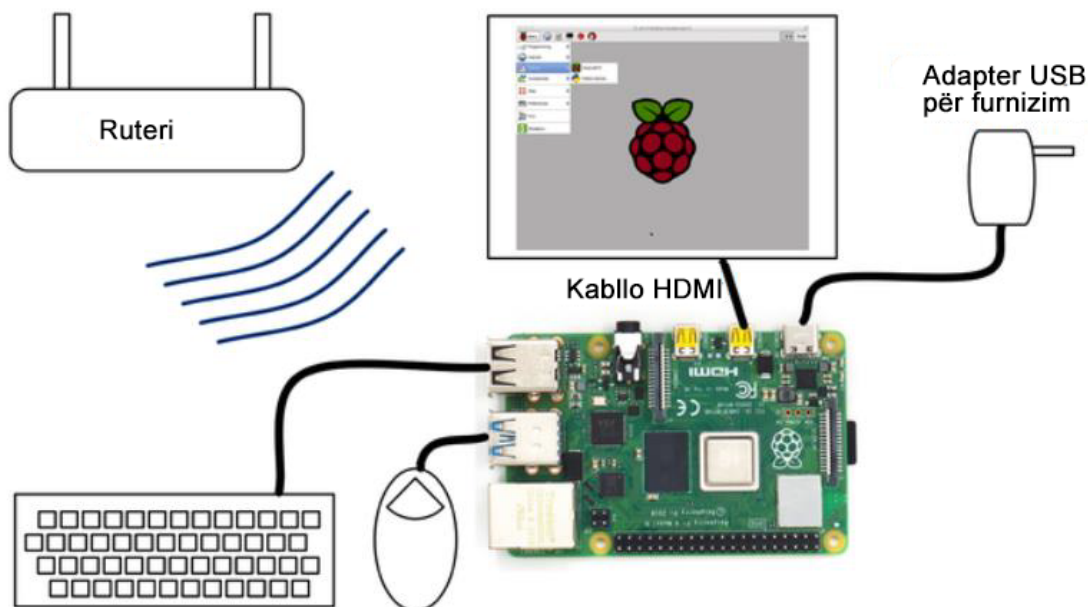


Figura 5.5. Sistemi Raspberry Pi

Gjatë ekzekutimit të ushtrimeve, Raspberry Pi 3B+ duhet të jetë i lidhur me tastierë për futjen e tekstit të kodit programor dhe me monitor për të shfaqur rezultatet

tekstuale të marra pas ekzekutimit të programit. Në figurën 5.5. është paraqitur një sistem i plotë Raspberry Pi. Të rikujtojmë se, për lidhjen me tastierë përdoret porta USB kurse për lidhjen me monitor konektori HDMI. Gjithashtu nuk ka kufizime në përzgjedhjen e modelit të Raspberry Pi, sepse kodet e dhëna janë kompatibil me të gjitha modelet.

Në çdo ushtrim praktik, është dhënë skema për lidhjen së Raspberry Pi 3B+ me pajisjet hyrëse-dalëse, duke përdorur një protoplakë. Pastaj pason kodi programor me komentet përkatëse. Me kodet programore mund të eksperimentohet duke ndryshuar vlerat e disa parametrave: shpejtësinë e rrotullimit të motorit, frekuencën e pulsimit të diodës, ngjyrën e diodës RGBLED, tekstit të shfaqur, etj.

5.3.1. Ushtrim praktik: Kyçja e diodës led me taster

1. Qëllimi i ushtrimit

Qëllimi i ushtrimit është lidhja e diodës led dhe tasterit me Raspberry Pi 3B+ duke përdorur një protoplakë, shkrimin e programit në gjuhën programuese, ngarkimin e programit në Raspberry Pi duke përdorur veglat e mjedisit zhvillimor të integruar Thonny Python dhe testimi i funksionimit të qarkut elektrik.

2. Koha e realizimit: 1 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistem operativ Raspbian të instaluar. Sistemi përbëhet nga një Raspberry Pi 3B+ me adapter për furnizim, tastierë, maus dhe monitor.
- protoplakë
- taster për pllakën e printuar
- Ldiodë led
- rezistor me rezistencë $R1=220\Omega$
- tela lidhës (qarqe të shkurtëra)

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- diagramin e pinave për bazamentin GPIO
- mënyrën e lidhjes së diodës led dhe tasterit në protoplakë. E njëjta është shpjeguar në udhëzimet për punë me protoplakën
- instruksionet nga klasat e bibliotekës GPIO Led dhe Button për të punuar me taster dhe diodën led.

- ushtrimin praktik 4.8.1.1. Kyçja e diodës led me taster me përdorimin e Arduino Uno R3 dhe të bëhet krahasimi skemës elektrike dhe kodit programor

5. Përshkrimi i skemës elektrike dhe mënyra e lidhjes

Në figurën 5.6. janë paraqitur skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

- (1) Tasteri vendoset në mesin e protoplakës. Terminali i djathtë i tasterit lidhet me shiritin për tokëzimi, ndërsa terminali i majtë me pinin numër 2 të GPIO.
- (2) Katoda e diodës led lidhet me shiritin për tokëzim, ndërsa anoda me pinin numër 17 të GPIO.
- (3) Shiriti i tokëzimit të protoplakës lidhet me pinin GND të Raspberry Pi 3B+.

Skemat elektrike në figurën 5.6. dhe figurën 4.13. nga pjesa praktike e librit janë të ndryshme. Te Raspberry Pi tasteri nuk është i lidhur me shiritin për furnizim të protplakës. Kështu, kur tasteri nuk është i shtypur atëherë pini numër 2 i GPIO nuk është i lidhur, çarku është i hapur. Kur shtypet tasteri, pini numër 2 i GPIO lidhet me shiritin për tokëzim. Nivel i lartë logjik (HIGH) i pinit numër 2 të GPIO është tregues se tasteri është shtypur. Dioda led ndriçon kur pini numër 2 i GPIO është në nivel të lartë logjik.

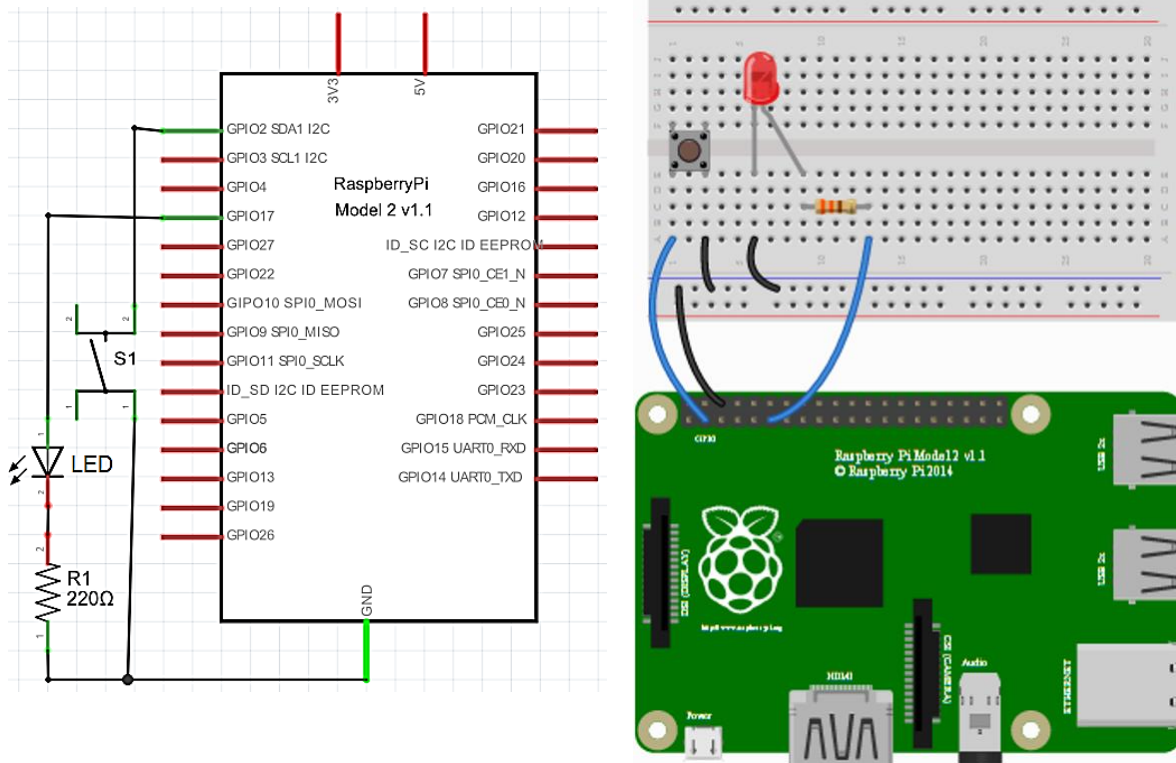


Figura 5.6. Lidhja e diodës led dhe tasterit me Raspberry Pi 3B+

6. Shkrimi dhe ekzekutimi i programi për Raspberry Pi 3B+

Në figurën 4.32. nga pjesa teorike e librit është treguar pamja e mjedisit zhvillimor të integruar Tonny Python IDE të cilin do ta përdorim për të shkruar programin dhe për ta ekzekutuar. Hapim një projekt të ri me veglën New, e shkruajmë kodin e programit në editorin e tekstit dhe pasi e shkruajmë e ruajmë (angl. Save). Kur do të ruajmë emrin e programit, ndryshon nga untitled në emrin që kemi vendosur. Pjesa e emërtuar si Shell shërben për të paraqitjen e mesazheve tekstuale duke përdorur instruksionin print(). Në kllapa të vogla shkruhet teksti që duhet të shfaqet, i shkruar midis thonjëzave. Më poshtë është dhënë kodi i programit për realizimin e këtij ushtrimi.

```

1 from gpiozero import LED, Button # Kyçim dy pjesë nga biblioteka
                                     # GPIOZERO për funksionimin e diodës led
                                     # dhe #tasterit.
2 from signal import pause          # Kyçim pjesë nga biblioteka signal për futjen e
                                     # kohës së vonesës.
3 led = LED(17)                     # Konfigurojmë pin për lidhjen e diodës
                                     #
4 button = Button(2)                 # Konfigurojmë pin për lidhjen e tasterit
                                     #
5 button.when_pressed = led.on      # E kyçim diodën led kur tasteri shtypet
                                     #
6 button.when_released = led.off    # E shkyçim diodën led kur tasteri lëshohet
                                     #
7 pause()                            # Fusim kohë vonese. Zakonisht koha e
                                     # vonesës është sekondë. Kohën e ndryshojmë
                                     # duke future vlerë në kllapat e vogla.
                                     #

```

Që të ekzekutohet programi i shkruar është e nevojshme të shtypet ikona Run. Krahasuar me platformën Arduino, Raspberry Pi është mikrokompjuter me sistemin e vet operativ. Interpreti e përkthen programin, instruksion pas instruksioni madje edhe gjatë shkrimit të vetë programit dhe nuk janë të nevojshme veglat Verify dhe Upload.

7. Rezultatet nga realizimi i ushtrimit

Menjëherë pas fillimit të programit, duhet të shtypim tasterin dhe dioda led do të ndriçoj. Kur tasterin dom ta lëshojmë, dioda led do të ndalojë së ndriçuari.

Komenti: _____

8. Bëni një ndryshim!

- Duam që dioda led të ndriçojë edhe për një kohë të caktuar pas lëshimit të tasterit. Cili instruksion do të përdoret dhe ku do të futet i njëjti?

Komenti _____

- Duam që dioda led të ndriçojë kur tasteri lëshohet dhe të mos ndriçoj kur tasteri shtypet. Cilat ndryshime duhet bërë në kodin e programit?

Komenti: _____

5.3.2. Ushtrim praktik: Semafori

1. Qëllimi i ushtrimit

Dritat e semaforit kyçen dhe shkyçen sipas kohës së përcaktuar saktë dhe cikli është i pafund, gjegjësisht përsëritet vazhdimisht. Qëllimi i këtij ushtrimi praktik është komandimi me dritat e semaforit duke përdorur një Raspberry Pi 3B+. Në bibliotekën gpiozero, ekziston një klasë e veçantë pikërisht për të punuar me semafor (TrafficLights) dhe kjo e thjeshton ndjeshëm shkrimin e kodit të programit.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistem operativ Raspbian të instaluar
- protoplakë
- tre dioda led
- tre rezistorë me rezistencë $R=220\Omega$
- tela për lidhje (qarqe të shkurtëra)

4. Përgatitja për ushtrimin

Nga hardueri i Raspberry Pi 3B+, nxënësit duhet të rikujtohen në diagramin e pinave të bazamentit GPIO dhe mënyrën e lidhjes së diodave led në protoplakë. I njëjti është shpjeguar në udhëzimet për punë me protoplakën.

Gjithashtu, para fillimit të ushtrimit, nxënësit duhet të njihen me klasën TrafficLights nga biblioteka e gpiozero dhe me instruksionin sleep nga biblioteka time.

Shembulli 4.54.

```
1 semafor=TrafficLights(25, 8, 7)
```

Shembulli 4.54. është shembull i deklaramit të objektit i cili do të jetë bartës i klasës TrafficLights(). Në programin e këtij ushtrimi praktik është deklaruar një objekt i kësaj klase sepse ka vetëm një semafor. Nëse duam ta zgjerojmë sistemin edhe me një semafor tjetër, i cili do të kontrollojë trafikun në rrugën e dytë të kryqëzimit, atëherë do të duhet të deklarojmë edhe një objekt tjetër nga klasa TrafficLights(). Numrat 25, 8 dhe 7 në kllapat e vogla janë numrat e pinave të GPIO në të cilët janë të lidhur tri dritat gjegjësisht tri diodat. Nga biblioteka e

kontrollit të kohës përdoret vetëm instruksioni sleep(). Në kllapa të vogla theksohet koha e shprehur në sekonda për të cilën sistemi do të jetë në pauzë.

5. Përshkrimi i skemës elektrike dhe mënyra e lidhjes

Lidhja e diodave led me Raspberry Pi 3B+ është shumë e thjeshtë. Gjatë vendosjes së diodave led në protoplakë duhet pasur kujdes që konektorët e tyre të vendosen në pika kontakti që u përkasin përçuesve të ndryshëm. Katoda e çdo diode led është e lidhur me shiritin e tokëzimit përmes një rezistence 220Ω. Anoda e diodës led të kuqe është e lidhur me pinin me numër 2 të GPIO, anoda e diodës led të verdhë me pinin numër 3 të GPIO dhe anoda e diodës led të gjelbër me pinin numër 4 të GPIO. Të mos harrojmë që pinin GND të Raspberry Pi 3B+ ta lidhim me shiritin e tokëzimit nëpërmjet një qarku të shkurtër të zi.

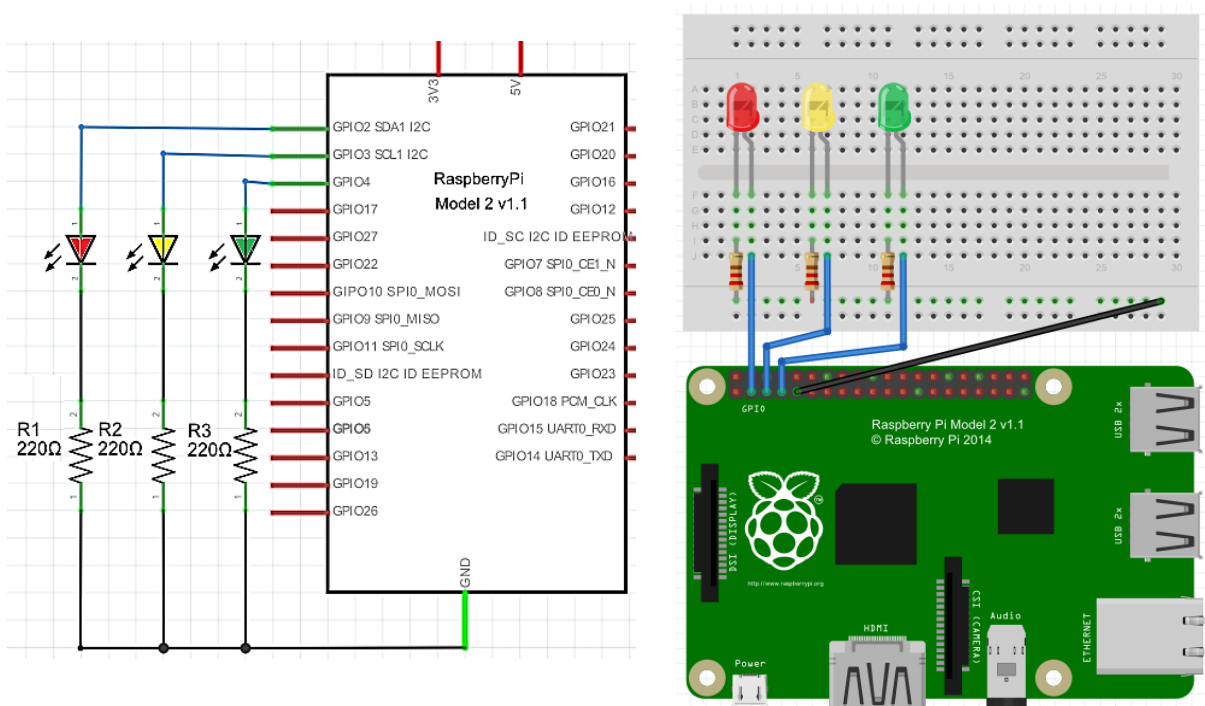


Figura 5.7. Lidhja e semaforit me Raspberry Pi 3B+

6. Shkrimi dhe ngarkimi i programi për Raspberry Pi 3B+

Procedura e shkrimit dhe ngarkimit të programit është e njëjtë si në ushtrimin e mëparshëm. Fillimisht hapim një projekt të ri, e shkruajmë në editorin e tekstit, e ruajmë dhe e emërtojmë dhe në fund e ekzekutojmë duke shtypur veglën Run. Më poshtë është dhënë kodi i programit për këtë ushtrim praktik.

```

1 from gpiozero import TrafficLights # Kyçet pjesa e bibliotekës GPIOZERO për
2 from time import sleep           # punën me semafor.
3 lights = TrafficLights(2, 3, 4)  # Me instruksionin sleep e vendosim
                                   # pajisjen në regjimin e pauzës.
                                   # Konfigurimi i pinave për diodat led të
                                   # kuqe, të verdhë dhe të gjelbër.
                                   #

```

```

4 lights.green.on()           # Kyçet dioda led e gjelbër.
5 while True:                 # Fillon cikli që përsëritet pafundësisht.
                               #
6     sleep(10)               # Dioda led e gjelbër ndriçon 10 sekonda.
7     lights.green.off()      # E shkyçim diodën led të gjelbër.
8     lights.amber.on()       # Ja вклучуваме жолтата.
9     sleep(1)                 #
10    lights.amber.off()      # E shkyçim të verdhën.
11    lights.red.on()         # E kyçim diodën led të kuqe.
12    sleep(10)               # Ajo ndriçon 10 sekonda.
13    lights.amber.on()       # Kyçet dioda led e verdhë.
14    sleep(1)                 #
15    lights.red.off()        #
16    lights.amber.off()      #
17    lights.green.on()       #
    sleep(2)

```

7. Rezultatet nga realizimi i ushtrimit

Menjëherë pas shtypjes së veglës Run në shiritin e veglave të mjedisit zhvillimor të integruar të Thonny Python do të filloj të ndriçoj dioda led e gjelbër për 10 sekonda, më pas ndriçon dioda led e verdhë për një sekondë, pastaj ndriçon dioda led e kuqe për 10 sekonda. Në fund ndriçojnë diodat led e kuqe dhe e verdhë së bashku për dy sekonda dhe përsëri kthehemi në diodën led të gjelbër. Cikli përsëritet pafundësisht.

Komenti _____

8. Bëni një ndryshim!

Nxënësit mund të ndryshojnë kohën e ndriçimit të diodave led. Gjithashtu mund të shtojnë edhe një semafor tjetër që do të funksionojë anasjelltas me të parin dhe për këtë qëllim duhet të ndryshojnë kodin e programit.

Komenti _____

5.3.3. Ushtrim praktik: Koha e reagimit

1. Qëllimi i ushtrimit

Qëllimi i këtij ushtrimi është të përcaktojë se cili nga dy tasterët do të shtypet i pari. Dioda led ndriçon dhe sinjalizon fillimin e "garës". Rezultati shfaqet në pjesën me emrin Shell në formën e një mesazhi tekstual. [7]

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistem operativ Raspbian të instaluar.
- protoplakë
- një diodë led

- një rezistor me rezistencë $R=220\Omega$
- dy tasterë
- tela lidhës (qarqe të shkurtëra)

4. Përgatitja për ushtrimin

Ngjashëm si në ushtrimin e mëparshëm nxënësit duhet të rikujtohen për diagramin e pinave të bazamentit të GPIO dhe mënyrën e lidhjes së diodës led në protoplakë.

Në këtë ushtrim për herë të parë do të përdoret instruksioni uniform nga biblioteka random. Ky instruksion përdoret për zgjedhjen në mënyrë të rastësishme (angl. random) të një numri decimal në brezin nga 0 deri në 5 dhe i njëjti është përdorur për zgjedhjen e kohës (angl. time) në instruksionin sleep(time). Ndryshimi i kohës përdoret si element i surprizës për të dy lojtarët sepse nuk do të dihet saktësisht se kur do të ndriçoj dioda.

Gjithashtu, për herë të parë do të përdoret monitori serik (angl. Shell) dhe instruksioni print() për të paraqitur mesazh tekstual me informacionin për fituesin, lojtarin e parë ose të dytë.

5. Përshkrimi i skemës elektrik dhe mënyra e lidhjes

Në figurën 5.8. janë paraqitur skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

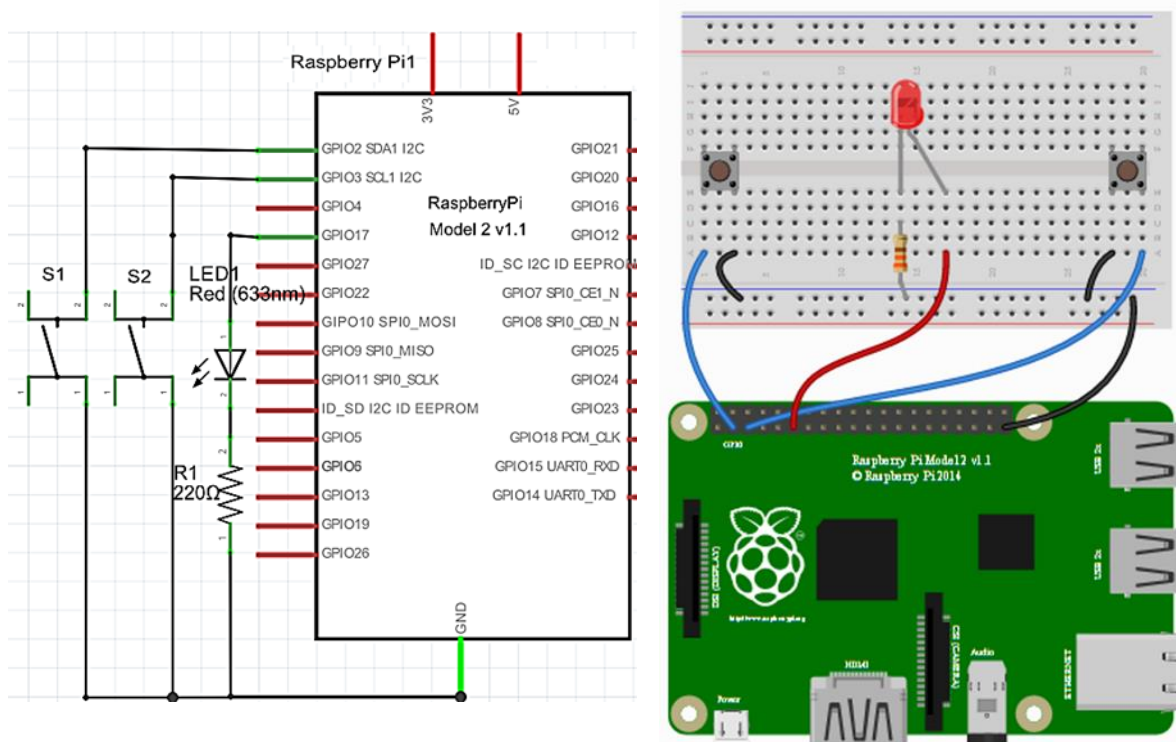


Figura 5.8. Lidhja e dy tasterëve dhe diodës led me Raspberry Pi 3B+

Lidhja është shumë e thjeshtë. Tasterët vendosen në mesin e protoplakës dhe konektorët e djathtë janë të lidhur me shiritin për tokëzim, ndërsa konektorët e majtë me pinat me numra 2 dhe 3 të GPIO me qarqe të shkurtëra blu.

Konektorët e diodës led vendosen në pikat e kontaktit që i përkasin dy përcuesve të ndryshëm në protoplakë dhe katoda nëpërmjet rezistorit lidhet me shiritin e tokëzimit, ndërsa anoda lidhet drejtpërdrejt me pinin numër 17 të GPIO me qark të shkurtër të kuq. Të mos harrojmë të lidhim shiritin e tokëzimit me njërin nga pinat GND në Raspberry Pi.

6. Shkrimi dhe ngarkimi i programit për Raspberry Pi 3B+

E shkruajmë programin në mjedisin zhvillimor të Thonny Python IDE. Të riujtojmë se i njëjti është pjesë e sistemit operativ Raspbian dhe për ta hapur atë shtypim në opsonin Programim → Tools në menynë e vetë sistemit operativ.

```

1 from gpiozero import Button,
  LED
2 from time import sleep
3 import random                # Biblioteka random jep mundësinë për
                                # zgjedhje të vlerave të rastit.
4 led = LED(17)
5 player_1 = Button(2)         # Konfigurimi i pinave për lidhjen e
                                # tasterëve dhe njëkohësisht vendosjen e
6 player_2 = Button(3)         # emrave simbolik të dy tasterave.
                                #
7 time = random.uniform(5, 10) # Definojmë një variabël me emrin
                                # simbolik time dhe një vlerë të rastit në
                                # brezin nga 5 derri në 10.
8 sleep(time)                 # Sistemi është në pauzë për kohën prej
                                # 5 deri në 10 sekonda.
9 led.on()                    # Pas kalimit të kësaj kohe kyçet dioda.
                                #
10 while True:                 # Fillon cikli.
11     if player_1.is_pressed:  # Paraqitet mesazh tekstual nëse butoni i
                                # lidhur me pinin numër 2 është shtypur i
12         print("Player 1 wins!") # pari.
13         Break                # Nëse është plotësuar kushti i mësipërm
                                # dalim nga cikli.
14     if player_2.is_pressed:  # Paraqitet mesazh tekstual nëse butoni i
                                # lidhur me pinin numër 3 është shtypur i
15         print("Player 2 wins!") # pari.
16         Break                # Dalim nga cikli.
17 led.off()                   # Dioda led shkyçet dhe kthehem në fillim
                                # të programit.

```

Gjatë kohës së shkrimit të programit, ai ngarkohet në memorien e Raspberry Pi.

7. Rezultatet nga realizimi i ushtrimit

Për të kontrolluar korrektësinë e ushtrimit duhet vetëm të shtypim Run në shiritin me vegla të mjedisit zhvillimor dhe të dy lojtarët të shtypin tasterët pasi të ndriçoj dioda led. Në monitorin e mjedisit zhvillimor duhet të paraqitet mesazh tekstual. Mes dy shtypjeve të tasterëve është vendosur kohë me vlerë të rastësishme në brezin prej 5 deri në 10 sekonda. Pasi që bëhet fjalë për cikël të pafundëm while, në përsëritjen e dytë të ciklit mund të merret një rezultat krejt i ndryshëm gjegjësisht një mesazh tekstual i ndryshëm.

Komenti: _____

8. Bëni një ndryshim!

Në ushtrim, të dy lojtarët mund të fusin emrat e tyre dhe të njëjtit të shfaqet në mesazhin tekstual. Për këtë qëllim, duhet të bëhet ndryshim në kodin e programit. Pas rreshtit programor 6, gjegjësisht pas konfigurimit të pinave për lidhjen e tasterëve duhet të shtohen instruksionet e mëposhtme.

```
lojtari_i_majtë=input("Fut emrin")  
lojtari_i_djathtë=input("Fut emrin")
```

Ndryshimi duhet të bëhet edhe në instruksionet print() dhe në vend të print("Player 1 win!") dhe print("Player 2 win!") shkruaj print("lojtari_i_majtë fiton!") dhe print("lojtari_i_djathtë fiton! ")

Komenti: _____

5.3.4. Ushtrim praktik: kyçja dhe shkyçja e diodës led me fotorezistor

1. Qëllimi i ushtrimit

Ky ushtrim praktik është i ngjashëm me ushtrimin 4.8.1.4. Fotorezistori për kontrollimin e intensitetit të dritës nëpërmjet Arduino Uno R3. Por, Raspberry Pi nuk ka të integruar një konvertues analogo-digjital siç ka Arduino Uno. Për këtë arsye nuk mund ta ndryshojmë intensitetin e dritës të diodës led ashtu siç vepruam në ushtrimin praktik me Arduino Uno. Në vend që të ndryshojë intensiteti i dritës i diodës led ajo thjesht do të kyçet dhe shkyçet, në varësi të asaj nëse është dritë apo errësirë, ditë apo natë.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistemin operativ Raspbian të instaluar.
- protoplakë
- një diodë led
- një rezistore me rezistencë $R=220\Omega$
- fotorezistor

- kondensator me kapacitet $1\mu\text{F}$
- tela për lidhje (qarqe të shkurtëra)

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- funksionimin e qarkut serik RC (lënda mësimore Elektroteknika)
- Diagramin e pinave të bazamentit të GPIO dhe mënyrën e lidhjes së diodës led në protpllakë.
- Ndryshimin e rezistencës së fotorezistorit në varësi të intensitetit të dritës dhe instruksioneve nga klasa LightSensor nga biblioteka gpio. (njësia mësimore 4.15.4. Sensori optik ose fotorezistori nga pjesa teorike e librit)

5. Përshkrimi i skemës elektrike dhe mënyra e lidhjes

Skema funksionale dhe e montimit për realizimin e këtij ushtrimi është dhënë në figurën 5.9.

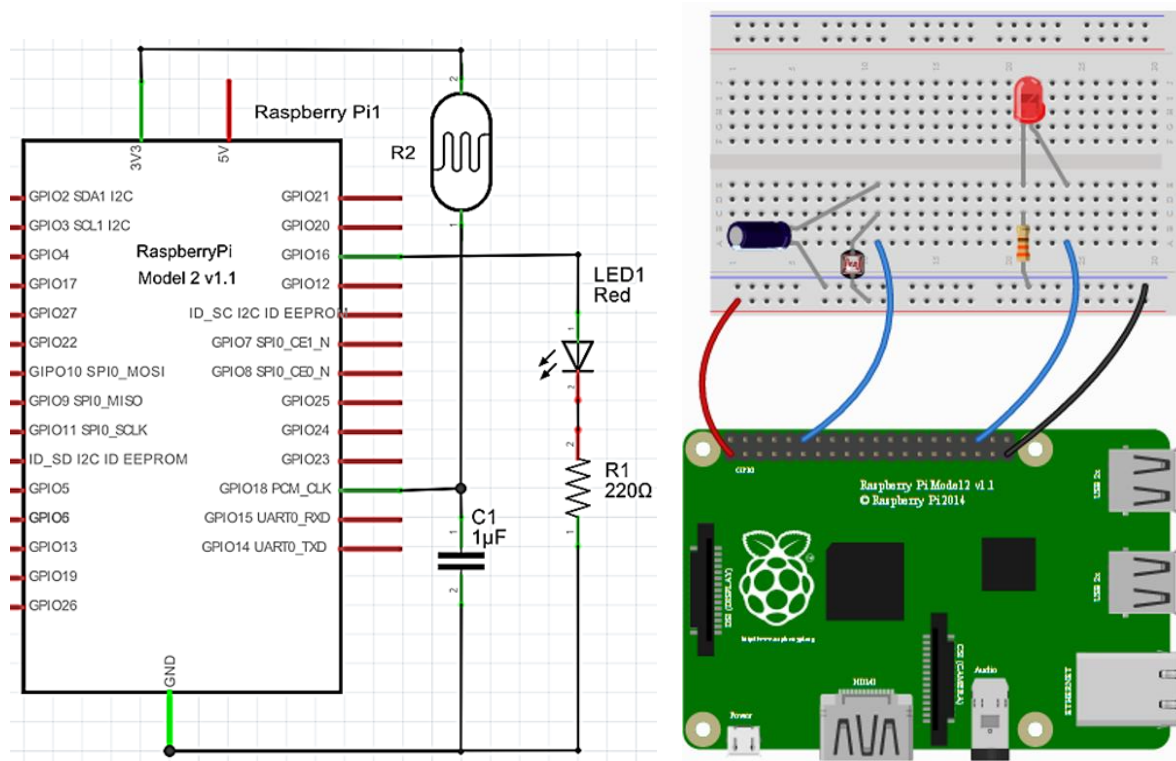


Figura 5.9. Lidhja e diodës led dhe fotorezistorit me një Raspberry Pi 3B+

Në ushtrimin 4.8.1.4. Fotorezistori për kontrollin e intensitetit të dritës, fotorezistori dhe rezistori $10\text{K}\Omega$ formojnë një ndarës të tensionit. Në këtë ushtrim praktik, në vend të një ndarësi tensioni, do të përdorim një qark serik RC i përbërë nga një rezistor dhe një kondensator, të lidhur në seri me furnizimin me energji prej 3.3V . Një konektor i fotorezistorit është i lidhur me shiritin e furnizimit, ndërsa konektori tjetër është i lidhur me pinin me numër 18 të GPIO. Në të njëjtin pin GPIO lidhet edhe kondensator me kapacitet prej $1\mu\text{F}$, ndërsa konektori tjetër i kondensatorit lidhet me shiritin për tokëzim. Nëse fotorezistori ka

rezistencë të vogël atëherë tensioni i kondensatorit ndryshon më shpejtë dhe anasjelltas. Raspberry Pi 3B+ mat kohën e nevojshme që tensioni i kondensatorit të rritet nga zero në vlerën prej 1.34V që paraqet tensionin e pragut (shiko figurën 4.35 në pjesën teorike të librit) dhe sipas kohës së matur detekton dritë ose errësi. Lidhja e diodës led është identike si në ushtrimet e mëparshme praktike, me një ndryshimin që tani përdoret pini me numër 16 i GPIO. Shiritin për furnizim e lidhim me pinin 3.3V të Raspberry Pi, ndërsa shiritin për tokëzim me pinin GND.

6. Shkrimi dhe ngarkimi i programit për Raspberry Pi 3B+

Programin e shkruajmë në mjedisin zhvillimor Thonny Python IDE. Nga biblioteka gpio e shkarkojmë klasën me emrin simbolik LightSensor e cila përmban instruksione për punë me fotorezistor. Do t'i përdorim instruksionet sensor.when_dark dhe sensor.when_light. Instruksioni sensor.when_dark ekzekutohet kur Raspberry Pi 3B+ detektor errësi (ndryshim i ngadalshëm i tensionit të kondensatorit) dhe instruksioni sensor.when_light ekzekutohet kur Raspberry Pi 3B+ detekton dritë (ndryshim i shpejtë i tensionit të kondensatorit).

```

1 from gpiozero import LightSensor, LED # Nga biblioteka GPIOZERO i thirr
# klasat e nevojshme.
2 from signal import pause
3 sensor = LightSensor(18) # Konfigurojmë pinat për lidhjen e
4 led = LED(16) # fotorezistorit dhe diodës led.
#
5 sensor.when_dark = led.on # Dioda led kyçet kur senzori detekton
# errësi.
6 sensor.when_light = led.off # Dioda led kyçet kur senzori detekton
# dritë.
7 pause() # Futet koha e vonesës.
```

Pasi të shkruajmë programin në mjedisin zhvillimor, shtypim Run në menynë e veglave.

7. Rezultatet nga realizimi i ushtrimit

Kur fotorezistori ekspozohet ndaj dritës, dioda led nuk ndizet. Kur do të mbulojmë fotorezistorin me ndonjë objekt, dioda led fillon të ndriçojë.

Komenti: _____

5.3.5. Ushtrim praktik: Kontrollimi i distancës nga objekti më i afërt

1. Qëllimi i ushtrimit

Qëllimi i këtij ushtrimi është përdorimi i një sensori të distancës (HC-SR04), lidhja e tij me Raspberry Pi 3B+ dhe shkrimi i programit i cili distancën e matur do ta

tregojë në formën e mesazhit tekstual në monitorin Shell të mjedisit zhvillimor Thonny Python IDE

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistemin operativ Raspbian të instaluar.
- protoplakë
- sensori për distance (HC-SR04)
- tela për lidhje (qarqe të shkurtëra)

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- Ushtrimin praktik 4.2. Udhëzime për punën me protoplakë nga pjesa praktike e librit
- diagrami i pinave të bazamentit GPIO (figura 4.34. në pjesën teorike)
- njësinë mësimore 4.15.3. Sensori për distancën (HC-SR04) nga pjesa teorike e librit

5. Përshkrimi i diagramit elektrik dhe mënyra e lidhjes

Skema funksionale dhe e montimit për realizimin e këtij ushtrimi është dhënë në figurën 5.10.

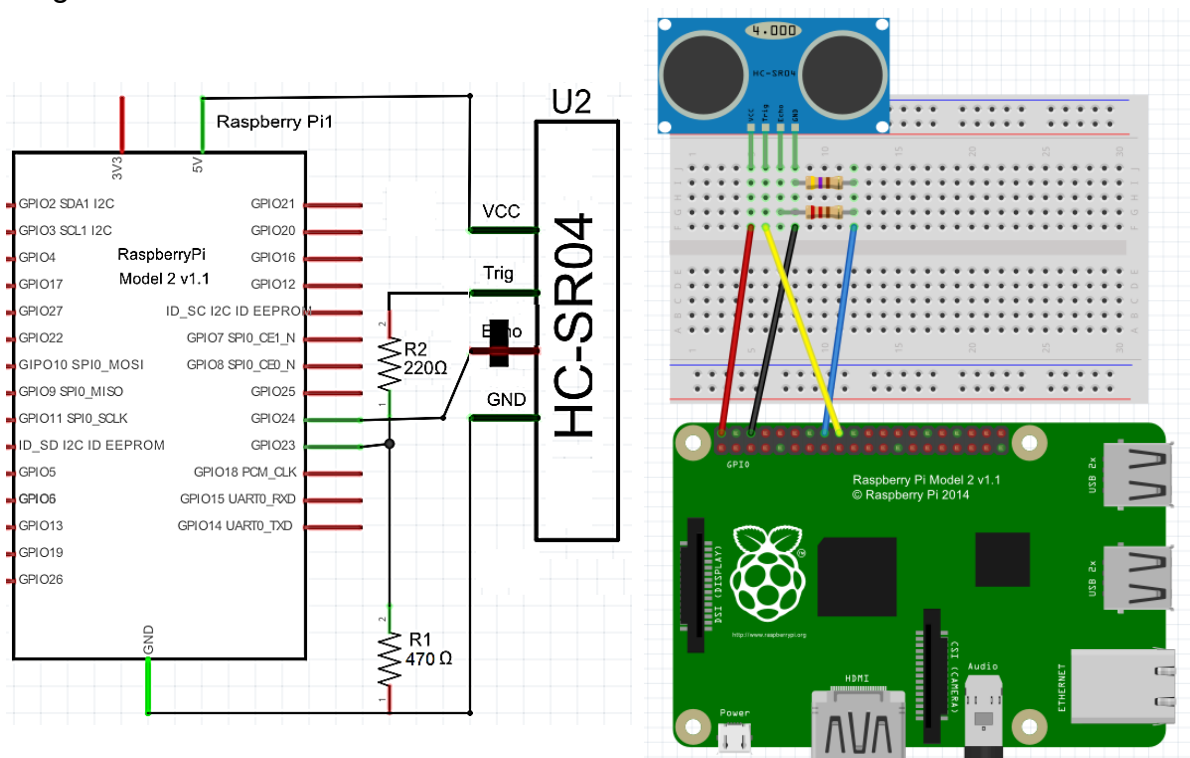


Figura 5.10. Lidhja e sensorit të distancës me Raspberry Pi 3B+

Për lidhjen e sensorit të distancës me Raspberry Pi 3B+ janë të nevojshëm dy pina GPIO, njëri për trigerin dhe tjetri për jehonën.

- (1) Konektorin për tokëzim të sensorit e lidhim me pinin GND në Raspberry Pi.
- (2) Konektorin Trig të sensorit e lidhim me pinin GPIO numër 24 në Raspberry Pi.
- (3) Njëri skaj i rezistorit me rezistencë 220Ω lidhet me konektorin Echo të sensorit.
- (4) Njëri skaj i rezistorit me rezistencë 470Ω lidhet me konektorin GND.
Dy skajet e tjera të dy rezistorëve së bashku janë të lidhur me pinin GPIO
- (5) të Raspberry Pi. Në këtë mënyrë, formohet një ndarës i tensionit i cili e zvogëlon tensionin në daljen e konektorit Echo. Ky tension është 5V, ndërsa e dimë se tensioni maksimal i hyrjes për Raspberry Pi 3B+ është 3.3V. Përndryshe, mund të vij deri te dëmtimi i Raspberry Pi.
Konektorët VCC të sensorit lidhet me pinin për furnizim prej 5V të
- (6) Raspberry Pi.

6. Shkrimi dhe ngarkimi i programit për Raspberry Pi 3B+

Programin e shkruajmë në mjedisin zhvillimor të Thonny Python IDE duke ndjekur procedurën tashmë të njohur: hapje e një projekti të ri → shkrimi në editorin e tekstit → ruajtja e programit me një emër simbolik të zgjedhur nga përdoruesi. Nga biblioteka gpio shkarkojmë klasën me emrin simbolik DistanceSensor, e cila përmban instruksione për të punuar pikërisht me sensorin e distancës (HC-SR04).

```

1 from gpiozero import          # E thirim klasën me instruksionet për
  DistanceSensor                # punë me sensorin për distancë
                                # (HC-SR04).
2 from time import sleep
3
4 sensor = DistanceSensor(23, 24) # Konfigurimi i pinave.
5
6 while True:                    # Fillon cikli i pafundëm.
7     print('Distanca nga objekti më i afërt është', sensor.distance, 'm')
  #Paraqitja e distancës së matur.
8     sleep(1)                   # Sistemi është në pauzë për 1 sekondë.
```

Programi është shumë i thjeshtë dhe i vogël. Kjo është kryesisht për shkak të përdorimit të bibliotekës gpio dhe klasës DistanceSensor. Përdoruesi nuk i detyruar të përdorë instruksione për lexim dhe shkrimi siç ishte rasti me platformën Arduino Uno. Me vetë thirrjen e klasës DistanceSensor kjo realizohet me automatizëm. Të gjitha karakteristikat harduerike të sensorit janë marrë parasysh dhe është krijuar softuer dhe i njëjti është i përfshirë në bibliotekën gpio, domethënë në klasën DistanceSensor.

7. Rezultatet nga zbatimi i ushtrimit

Para sensorit të distancës vendosim një objekt të caktuar. Distanca maksimale është 1 metër. Pasi të shkruajmë dhe ta ruajmë programin shtypim Run në menunë

me vegla dhe menjëherë në fushën e monitorit serik Shell shfaqet një mesazh tekstual me informacionin për distancën nga senzori deri në objektin e vendosur. Objekti mund të zhvendoset dhe ne mund të kontrollojmë besueshmërinë e të dhënave të marra.

Komenti: _____

5.3.6. Ushtrim praktik: Ndryshimi i drejtimit të një motori të rrymës njëkahëshe

1. Qëllimi i ushtrimit

Ky ushtrim është tjetër provë se sa e lehtë është të programosh Raspberry Pi 3B+ falë bibliotekave dhe numrit të madh të instruksioneve, të cilat e bëjnë programimin shumë të thjeshtë dhe të kuptueshëm. Qëllimi i këtij ushtrimi është të ndryshojmë drejtimin e rrotullimit të një motori - DC. Të rikujtojmë se motori për rrymë njëkahëshe nuk lidhet drejtpërdrejt me Raspberry Pi, por përdor një urë H si komponentë-ndërfaqe.

2. Koha e realizimit: 2 orë mësimore

3. Komponentët e nevojshme për realizimin e këtij ushtrimi janë:

- Sistemi Raspberry Pi 3B+ me sistemin operativ Raspbian të instaluar.
- protoplakë
- motor i rrymës njëkahëshe me furnizim prej 3V
- qark i integruar SN754410
- tela për lidhje (qarqe të shkurtër)

4. Përgatitja për ushtrimin

Nxënësit duhet të rikujtohen për:

- diagramin e pinave të bazamentit GPIO të Raspberry Pi 3B+ (figura 4.34. në pjesën teorike)
- njësinë mësimore 4.16.4. Motori i rrymës njëkahëshe nga pjesa teorike e librit
- diagramin i pinave të qarkut të integruar SN754410 (figura 4.44. në pjesën teorike)

5. Përshkrimi i diagramit elektrik dhe mënyra e lidhjes

Në figurën 5.11. dhe 5.12. është treguar skema funksionale dhe e montimit për realizimin e këtij ushtrimi.

- (1) Qarkun e integruar SN754410 e vendosim në mesin e protoplakës, duke pasur kujdes që të mos përdredhim pinat e tij. Me qarqe të shkurtëra të kuq i lidhim pinat për furnizimin dhe aktivizim (angl. Vcc, Enable) me shiritin për furnizim, ndërsa qarqe të shkurtëra të zi lidhim katër pinat GND me shiritin e tokëzimit.

- (2) Nëpërmjet protoplakës, me qarqet e shkurtëra të kaltër lidhim pinat GPIO me numër 4 dhe 14 me pinat hyrëse (angl. In1,In2) të qarkut të integruar SN754410.
- (3) Nëpërmjet protoplakës, me qarqun e shkurtër të gjelbër dhe të verdhë, lidhim terminalet e motorit të rrymës njëkahëshe me pinat e daljes (angl. Out1, Out2) të qarkut të integruar SN754410.
- (4) Në fund, shiritin e sipërm dhe të poshtëm të furnizimit e lidhim me pinin 5V të Raspberry Pi, ndërsa shiritin e furnizimit me pinin GND.

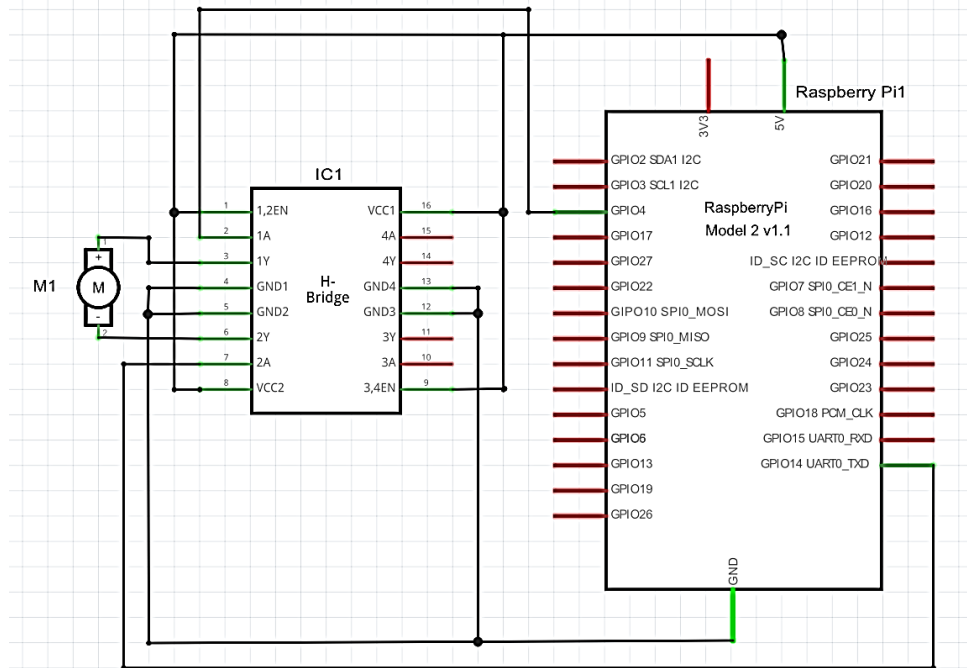


Figura 5.11. Skema funksionale për lidhjen e motorit të rrymës njëkahëshe me Raspberry Pi 3B+

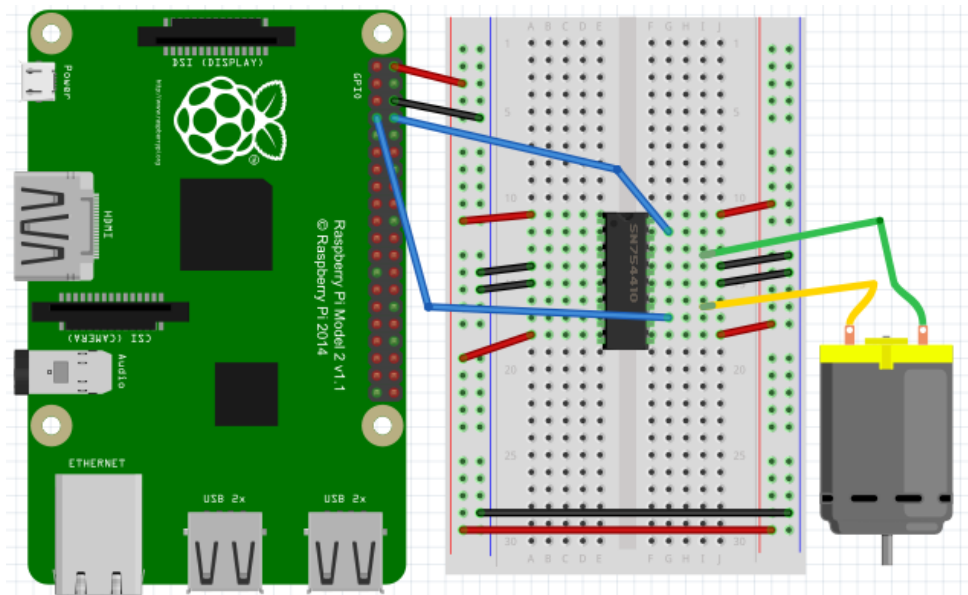


Figura 5.12. Skema e monitorimit për lidhjen e motorit të rrymës njëkahëshe me Raspberry Pi 3B+

6. Shkrimi dhe ngarkimi i programit për Raspberry Pi 3B+

E shkruajmë programin në mjedisin zhvillimor Thonny Python IDE në përbërje të sistemit operativ Raspbian. E thirrjmë klasën e instruksioneve Motor nga biblioteka gpio. I shfrytëzojmë instruksionet motor.forward() dhe motor.backward() me të cilat jemi njohur tashmë.

```

1 from gpiozero import Motor # E thirrjmë pjesën e bibliotekës gpio për punë
                                # me motor.
2 from time import sleep
3 motor = Motor(forward=4, backward=14)
4 #4 dhe 14 janë numrat e pinave GPIO.
5 while True:                    # Fillon cikëli i pafundëm.
6     motor.forward()            # Motori rrotullohet në drejtimin e shigjetës së
                                # orës. Pasi që kllapa e vogël është e zbrazët,
                                # kjo do të thotë se motori do të rrotullohet me
                                # shpejtësi maksimale. Nëse në kllapat e
                                # vogla do të ishte numri 0,5 kjo do të
                                # thoshte se motori do të rrotullohet me
                                # gjysmën e shpejtësisë së tij maksimale.
7     sleep(5)                  # Motori rrotullohet 5 sekonda në njërin drejtim.
8     motor.backward()          # Ndryshon drejtimin e rrotullimit.
9     sleep(5)                  # Motori rrotullohet 5 sekonda në drejtimin e
                                # kundërt të shigjetës së orës.

```

7. Rezultatet nga realizimi i ushtrimit

Rezultatit nga realizimi i këtij ushtrimi do të shohim nëse shtypim Run në menynë me vegla të mjedisit zhvillimor të Thonny Python IDE. Rezultate të pritshme janë që motori të fillojë të rrotullohet duke ndryshuar drejtimin e tij çdo 5 sekonda.

Komenti:_____

Literatura e shfrytëzuar:

- [1] И. Ј. Ј. Д. Љубица Маркудова, Дигитална електроника и микропроцесори-III година, Министерство за образование и наука, 2010.
- [2] Г. Силбершац, Концепти на оперативните системи, 2016: Арс Ламина.
- [3] [Online]. Available: <https://www.arduino.cc/en/hardware>.
- [4] M. S. Scott Fitzgerald, The Arduino Project Book, Torino, Italy, 2012.
- [5] B. J. N. W. Michael Margolis, Arduino Cookbook, 3rd Edition, O'Reilly Media, 2020.
- [6] "Arduino Reference," [Online]. Available: <https://www.arduino.cc/reference/en/>.
- [7] G. Halfacree, Raspberry Pi Beginner's Guide How to use your new computer, Cambridge, CB1 2JH: Raspberry Pi Trading Ltd, 2018.
- [8] S. Monk, Raspberry Pi Cookbook, 3rd Edition, O'Reilly Media, 2019.
- [9] "gpiozero-GPIO Zero 1.6.2. Documentation," [Online]. Available: <https://gpiozero.readthedocs.io/en/stable/>.
- [10] [Online]. Available: <https://www.alza.co.uk/how-to-build-your-own-PC..>
- [11] A. Warren, Writer, *Windows 10 Exam 70-698: Installing and Configuring Windows 10 LiveLessons*. [Performance]. Pearson IT Certification, 2018.
- [12] [Online]. Available: <https://www.newegg.com/>.

