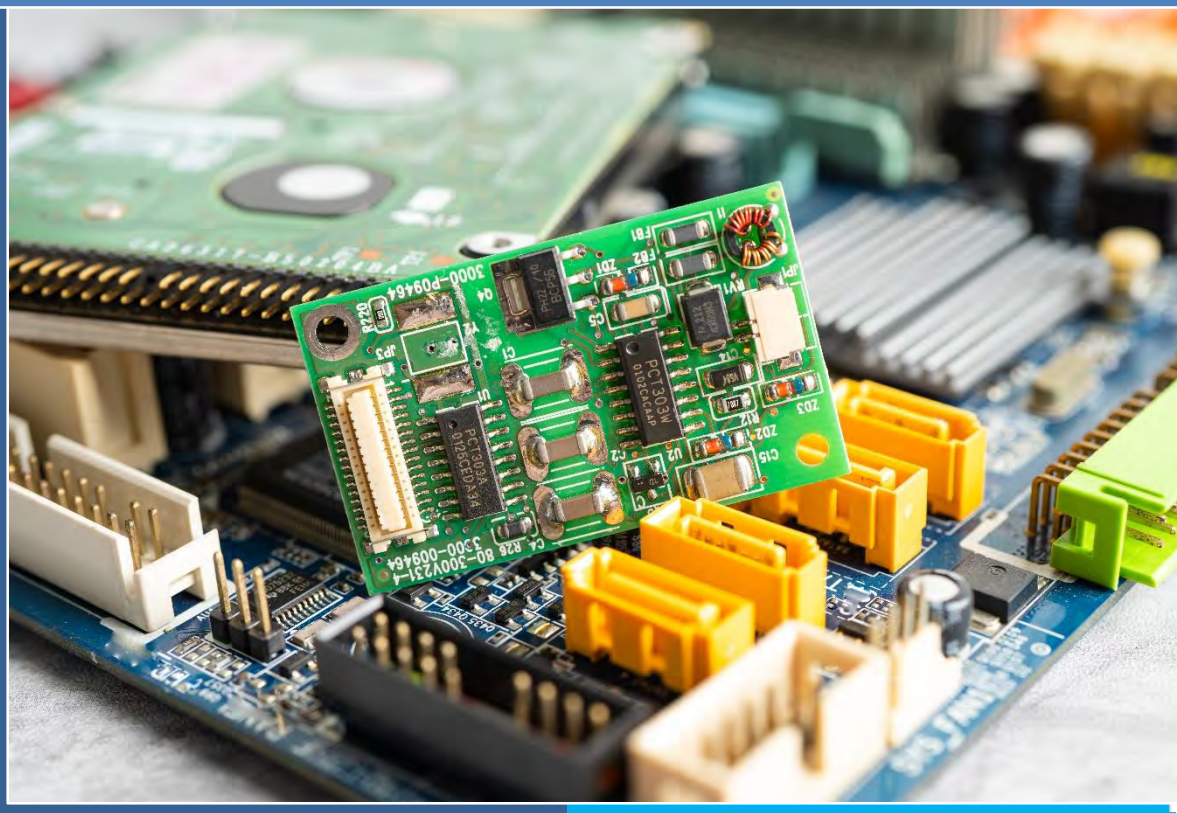
Љубица Маркудова

Компјутерски системи



Електротехничка струка

Електротехничар за електроника и телекомуникации

Трета година

Љубица Маркудова

Компјутерски системи

III година

2024

КОМПЈУТЕРСКИ СИСТЕМИ
III година

Автор:
Љубица Маркудова

Рецензенти:
Данијела Ефнушева
Поликсена Митева
Дејан Антоновски

Лектор:
Елена Саздовска

Стручна редакција:
Риза Етеми

Уредник:
Риза Етеми

Компјутерска подготовка:
Љубица Маркудова

Илустрација за насловна страна: https://www.vecteezy.com/free-photos

Издавач: Министерство за образование и наука на Република Северна Македонија, ул. „Св. Кирил и Методиј“ бр. 54, 1000 Скопје

Графичко и техничко уредување: Ели Василевска Илиевска – АРС СТУДИО

Место и година на издавање: Скопје, 2024 година

Со Одлука за одобрување на учебникот по предметот Компјутерски системи за III (трета) година, струка електротехничка, профил електротехничар за електроника и телекомуникации, бр. 26-977/1 од 11.07.2024 година, од Националната комисија за учебници.

Предговор

Овој учебник е наменет за учениците од трета година во средните стручни училишта, Електротехничка струка, за образовниот профил Електротехничар за електроника и телекомуникации. Учебникот е работен според модуларно дизајнираната наставна програма и е во согласност со наставните содржини по предметот Компјутерски системи од трета година. Цел на новата наставна програма е осовременување на наставата и наставните програми во стручното образование и целосно задоволување на барањата на пазарот на труд.

Наставниот предмет Компјутерски системи од трета година е застапен со фонд од 2 часа неделно или 72 часа годишно. Половина, односно 36 часа од вкупниот фонд на часови треба да се реализираат како практична настава или вежби. Часовите за практична настава или вежбите се реализираат во групи, согласно со Упатството на Министерството за образование и наука. Ваквиот концепт нуди можност стекнатите теоретски знаења да се применат практично, што учењето го прави поинтересно и забавно. Атрактивноста и интересот што го предизвикува развојот на компјутерите и нивното програмирање многу придонесоа за желбата на авторот да го напише овој учебник. Неговите содржини се разликуваат од содржините на класичните учебници по информатика бидејќи изработените апликации се имплементираат во програмабилни вградливи системи, како што се микрокомпјутерите на плочка Arduino Uno и Raspberry Pi. Овие микрокомпјутери, заедно со персоналните компјутери, ќе бидат неопходни за реализација на вежбите.

Учебникот опфаќа четири модуларни единици.

Првата модуларна единица, Основи на компјутерски системи, го објаснува поимот компјутерски систем, неговата организација и архитектура, различните видови компјутерски системи. Оваа модуларна единица дава голем број примери за примената на компјутерите. Во зависност од поставената цел, се избира соодветен компјутерски систем и софтвер за изработка на апликации.

Во втората модуларна единица, Хардвер на персонален компјутерски систем (PC), учениците ги утврдуваат или прошируваат своите знаења за карактеристиките и функцијата на основните хардверски компоненти. Ставен е акцент на монтажата на персонален компјутер и изборот на компоненти, со цел да се добие функционален компјутер со добар квалитет и прифатлива цена на чинење.

Со инсталацијата, оптимизацијата и преинсталацијата на оперативниот системи Windows 10 ќе се запознаеме во третата модуларна единица, Инсталација на оперативен систем и други стандардни програми за персонален компјутерски систем. Ќе бидат објаснети неколку корисни и кориснички програми. Исто така ќе бидат

објаснети чекорите при инсталација на оперативен систем Ubuntu 17 Linux во виртуелна машина и цел на оваа наставна единица е збогатување на знаењата на учениците кои сакаат да работат и во оперативен систем Linux.

Четвртата модуларна единица, Микрокомпјутери на плочка, е најобемна бидејќи не само што ги проучува програмските јазици C++ и Python, туку нив ги користи за програмирање на микрокомпјутерите на плочка Arduino Uno и Raspberry Pi. Се разбира, претходно мора да се запознаеме со хардверот и развојните програми на овие микрокомпјутери на плочка.

Во учебникот, во секоја тема има прашања, задачи и активности, со што се настојува да се применат знаењата и да се направи проверка на постигнувањата на учениците.

Авторот на овој учебник се обиде наставниот материјал да го презентира на колку што е можно поедноставен и разбирлив начин, со прецизни објаснувања на постапките и процесите и изработка на јасни цртежи и слики. За изработка на цртежите и на илустрациите беа користени две програми, Microsoft Visio и Fritzing. Fritzing е апликација од отворен тип, која нуди одлична поддршка за изработка на електрични шеми со платформите Arduino Uno и Raspberry Pi и протоплочка. При програмирање на микроконтролер на плочка можат да се користат компјутерски симулатори, каков што е бесплатниот Arduino Uno симулатор на веб-платформата Tinkercad.

Овој учебник им го посветувам на нашите ученици и искрено се надеваме дека ќе им помогне во остварувањето на нивните професионални цели. Денес, како никогаш досега, програмерите се најбарана професија, но успеваат само најупорните и најхрабрите.

Им се заблагодарувам на моето семејство и на колегите за несебичната и сестрана поддршка во изработката на овој учебник. Исто така, се заблагодарувам на рецензентската комисија, која со своите корисни сугестии придонесе за подобра реализација на овој учебник.

Од авторот

Содржина

Теоретски дел

Практични вежби за наставната програма Компјутерски системи

Основи на компјутерски системи

1.1. Поим и поделба на компјутерски системи

Денес компјутерските системи се користат насекаде и без нив не може да се замисли модерниот живот. Речиси не постои уред што во својот состав не содржи микрокомпјутер. Такви се: апаратите за домаќинство, мултимедијалните уреди и уредите за забава, автоматите, уредите за комуникација, медицинските апарати, индустриските погони, сообраќајните средства, канцелариските уреди итн.

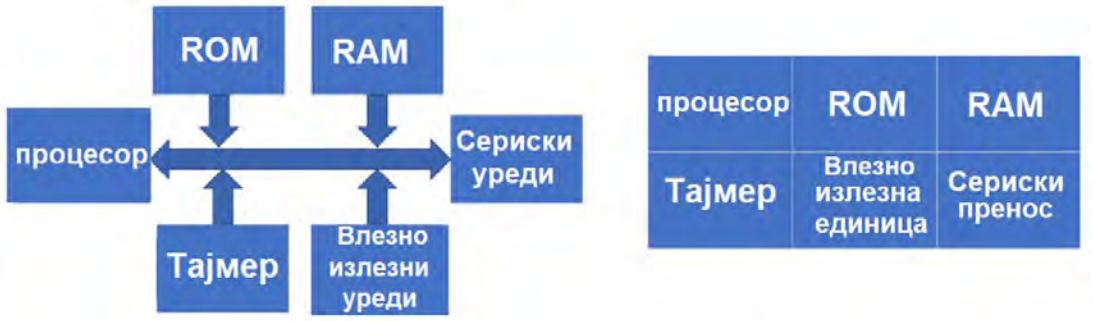
Слика 1.1. Поделба на компјутерските системи

Под **компјутерски систем** подразбираме систем составен од хардверски и софтверски компоненти, кој врши прием, обработка, чување и прикажување на податоци и информации. Секоја компонента има точно определена функција. Микропроцесорот, мемориите и влезно-излезните единици се хардверски компоненти. Микропроцесорот врши обработка на податоците. Мемориите ги чуваат податоците, трајно или привремено. Влезните единици вршат прием на податоци за обработка, а излезните единици ги прикажуваат информациите. За да компјутерот изврши некоја задача корисникот мора да му зададе инструкции

на компјутерот, кои операции да ги изврши и по кој редослед. Корисникот инструкциите ги задава преку соодветни програми.

Генерално, компјутерите се поделени **во четири групи**: микро, мини, макро и суперкомпјутери. Оваа поделба е направена според брзината на работа, големината, преносливоста, вградливоста, мемориските капацитети и комплексноста на операциите што можат да ги извршуваат.

- **Микрокомпјутерите** се компјутери наменети само за еден корисник. Во оваа група спаѓаат **персоналните компјутери**, како што се: десктоп, лаптоп, таблет, паметни телефони др. Друг пример за микрокомпјутерски систем претставуваат **микрокомпјутерите на плочка**.
  Микрокомпјутерите на плочка се вградливи микрокомпјутери и најмногу се користат во системите за автоматско управување. Микрокомпјутерите на плочка примаат информации за средината што ги опкружува преку сензорите, приклучени на нивните влезни пинови. Потоа овие податоци се процесираат во микропроцесорот и, по извршувањето на програмата, се активираат извршните единици приклучени на излезните пинови од микрокомпјутерот на плочка (електромотори, пумпи, дисплеи итн.). Најважен дел на микрокомпјутерите на плочка се микроконтролерите. Тоа се специјално дизајнирани интегрирани кола што во себе содржат централна процесорска единица, RAM и ROM. Наместо секоја хардверска компонента да биде посебен чип, сите компоненти се имплементирани во еден ист чип.



Слика 1.2. Споредба меѓу персонален компјутер и микроконтролер

Слика 1.2. претставува споредба меѓу персонален компјутер и микроконтролер. Во персоналните компјутери секоја од наведените хардверски компоненти е посебен чип на матичната плоча и затоа секоја е претставена со посебен блок, а стрелките кои ги поврзуваат блоковите ги претставуваат водовите на матичната плоча кои служат за пренос на податоци. Хардверските компоненти во составот на микроконтролерот се прикажани со еден заеднички блок. Освен процесор, трајна и работна меморија микроконтролерот во својот состав може да содржи тајмери за мерење на време, аналогно-дигитален претворувач и единица за сериска комуникација. Поради своите мали димензии и ниската цена на чинење микроконтролерот може да се вгради во многу електронски уреди. За

разлика од персоналните компјутери, микроконтролерите имаат многу помала работна фреквенција, помал мемориски капацитет и мрежно потешко се поврзуваат.

- **Миникомпјутерите** можат да ги користат повеќе корисници истовремено. Се користат за апликации во реално време, како контрола на сообраќај или автоматизација на индустриските процеси. Тоа се моќни компјутери со огромни брзини, кои можат да содржат еден или повеќе микропроцесори. Во оваа група спаѓаат серверите, кои се користат за управување со телефонскиот сообраќај, локални или глобални компјутерски мрежи.
- **Макрокомпјутерите** се наменети за апликации со поголем степен на интерактивност. Еден макрокомпјутер може да биде поврзан со повеќе терминали. Се користат во поголемите организации, како банките или државните институции каде што е потребна обработка и складирање на податоците од милиони корисници.
- Најмоќни и најскапи се **суперкомпјутерите.** Суперкомпјутерите се користат за симулација и за моделирање на комплексни феномени, како што се хемиски реакции, нуклеарни експлозии, временски прогнози, космички истражувања, воени потреби.

Се разбира, ние не сме во можност да ги проучиме сите овие компјутерски системи. Ќе ги проучиме персоналните компјутери и микрокомпјутерите на плочка од платформите Arduino и Raspberry Pi.

1.2. Структура и организација на компјутерски систем

1.2.1. Блоковска структура на компјутерски систем

Веќе спомнавме дека компјутерот е составен од хардвер и софтвер. На слика 1.3. е прикажана блоковска структура на компјутерски систем, неговите хардверски и софтверски компоненти, нивните видови и составни делови. [1]

Под хардвер се подразбираат сите механички, физички делови од кои е направен компјутерот.

Микропроцесорот ги извршува инструкциите и ги обработува податоците. Составни делови на еден микропроцесор се: регистри, аритметичко-логичка единица и управувачка единица со декодер. Регистрите се брзи мемориски локации во внатрешноста на микропроцесорот, кои ги содржат податоците и кодовите за извршување на инструкциите. Самото извршување се

случува во аритметичко-логичката единица. Управувачката единица генерира контролни сигнали согласно кодовите на тековните инструкции.

Мемориите ги чуваат програмите и податоците. RAM-меморијата е привремена меморија и таа ги чува документите кои моментално се обработуваат, а ROM и хард-дискот се трајни мемории.

Магистралите ги пренесуваат податоците, документите, програмите и со нив детално ќе се запознаеме подоцна.

Периферните уреди се: влезните, излезните и влезно-излезните единици. За персонален компјутер влезни единици се: тастатура, глушец, скенер, камера, оптички читач, а излезни се монитор, печатач, цртач или плотер итн. Влезните единици ги претвораат корисничките информации во дигитални сигнали, нули и единици, а излезните уреди обратно. Влезно-излезните уреди служат за влез, но и за излез на податоци. Најпознати влезно-излезни единици се модемите и звучните картички.

Слика 1.3. Блок-шема на основен модел на компјутер

Софтверот е множество од програми врз основа на кои хардверот извршува одредени задачи. Софтверот може да се подели во две групи: системски и кориснички. Оперативниот систем претставува системски софтвер која ја контролира и ја координира работата на хардверските компоненти. Тој е софтвер најблизок до хардверот. Со посредство на оперативниот систем сите останати програми пристапуваат до хардверот. Оперативните системи овозможуваат огромните хардверски ресурси да бидат лесно достапни за човекот и обезбедуваат поудобна работа. Најмногу користени оперативни системи се Linux и Windows. Linux е познат по својата стабилност и сигурност,

а Windows по едноставноста во работењето и достапноста на апликативен софтвер. Најчесто серверите и суперкомпјутерите користат Linux, а Windows е најпопуларен оперативен систем за персонални компјутери. Microsoft Windows Server и UNIX/Linux се мрежни оперативни системи.

Р**азвојна околина** е софтвер што служи за создавање на нов софтвер или поддршка во изработката на апликативни програми. На пример, пред да биде вграден во некој електронски уред микрокомпјутерот на плочка треба да се испрограмира. За пишување на корисничка програма и нејзино впишување во програмската меморија потребна интегрирана развојна средина (анг. IDE-Integrated Development Environment). Заради подобро сфаќање на развојната околина, ќе објасниме неколку алатки што се дел од истата:

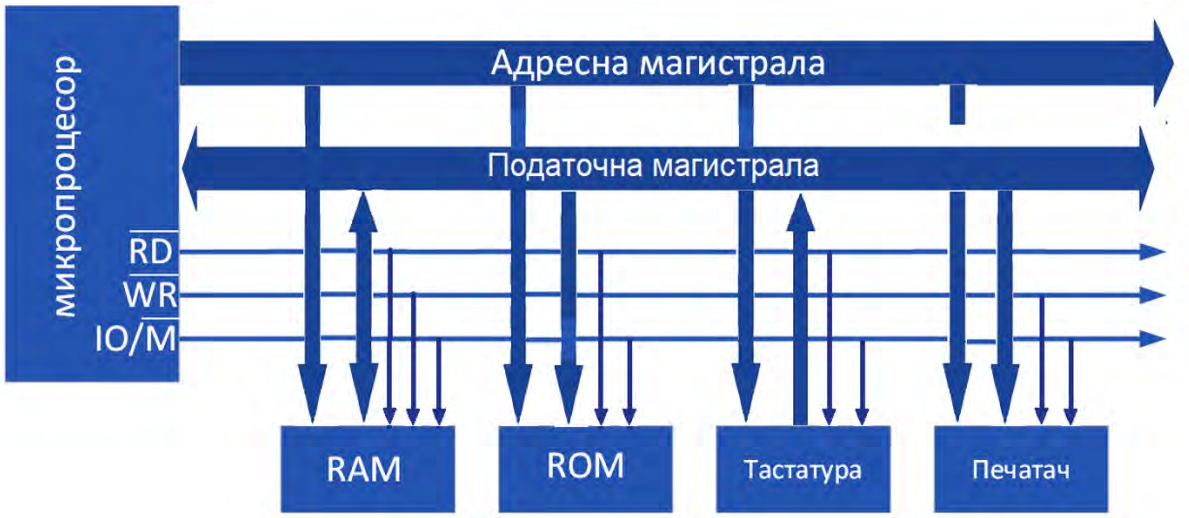
- Едиторите овозможуваат внесување, менување, обликување, памтење и печатење текст на програмата.
- Преведувачите ги преведуваат програмите напишани во програмски јазици од повисок ред во машински јазик. Преведувачите се познати и под името компајлери.
- Дебагери се програми за отстранување грешки. Се поставуваат контролни точки на одделни места во програмата и во текот на програмата се застанува на овие точки, при што се испитува точноста на програмата.
- Поврзувачи се програми што вршат поврзување на новиот софтвер со стариот што е многу важно за негово извршување.

Апликативните или корисничките програми се програми кои овозможуваат извршување на конкретни задачи според потребите на корисниците. Тие може да бидат наменети за едукација, канцелариско работење, деловно и финансиско работење, комуникација, забава итн. Во апликативен софтвер спаѓаат програмите за обработка на текст, работа со табели, работа со бази на податоци, обработка на видео и звук записи, обработка на слики и цртежи итн.

1.2.2. Магистрална структура на компјутерски систем

Магистралите се бакарни водови на матичната плоча и служат за пренос на податоци. Првите персонални компјутери имале една единствена магистрала наречена системска магистрала. Таа се состоела од 50 до 100 бакарни жици вградени во матична плоча, со конектори поставени на еднакви растојанија за приклучување на мемориските чипови и на влезно-излезните уреди. Модерните компјутери содржат повеќе магистрали со специјална намена, на пример, една магистрала за поврзување на процесорот со меморијата и друга магистрала за поврзување на процесорот со влезно-излезните уреди.

На слика 1.4. е прикажано како магистралите ги поврзуваат различните делови од компјутерот, како што се микропроцесорот, RAM, ROM и неколку периферни уреди.



Слика 1.4. Магистрална структура на компјутерски систем

За да се одвива преносот на податоци без грешки, се користат специјално дефинирани правила, до кои мора да се придржуваат сите уреди во компјутерскиот систем. Ова множество од правила се нарекува магистрален протокол (анг. bus protocol). **Основни карактеристики на една магистрала** се работната фреквенција и ширината. Под ширина на магистрала подразбираме од колку бакарни водови е составена, т.е. колку битови може да се пренесат истовремено. Работната фреквенција е од редна величина на MHz и ни покажува колку битови се пренесуваат по еден бакарен вод во една секунда. Ако се помножат работната фреквенција и ширината на магистралата, се добива **пропусниот опсег на магистралата**. Единица мерка за пропусен опсег е број на битови во секунда.

Според видот на сигналите што се пренесуваат, магистралите можат да се групираат во 3 основни групи: адресни линии (анг. ADDRESS), податочни (анг. DATA) и контролни (анг. CONTROL).

Адресната магистрала ги пренесува адресните битови од микропроцесорот до мемориите и периферните уреди. Адресата е единствена комбинација од нули и единици и служи за препознавање, идентификација на само една мемориска локација или периферен уред. Кога меморијата ќе ја добие адресата од микропроцесорот, таа се пребарува за да ја пронајде саканата мемориска локација.

Податочните линии служат за пренос на корисничката информација од еден до друг уред во компјутерот. Податочната магистрала е двонасочна. Кога се чита податок од меморијата, податочната магистрала пренесува податоци од меморијата кон микропроцесорот. Кога се впишува податок во меморијата, преносот на податоци се врши во насока од микропроцесорот кон меморијата. Магистралите можат да бидат 8, 16, 32 и 64-битни. Поголема ширина значи и

поголема брзина на работа. Така, на пример, ако треба да се пренесе 32-битен податок по 8-битна магистрала, тогаш се потребни 4 циклуси за пренос на податоци.

Контролните или управувачките линии се најмногубројни и за секоја управувачка линија постои специјална ознака, што, всушност, е скратеница од англискиот збор за дадениот контролен сигнал. Контролните линии служат за дефинирање на операцијата што треба да се изврши. Некои контролни битови се пренесуваат од микропроцесорот кон други уреди и претставуваат еден вид наредби. Други контролни битови се пренесуваат од уредите кон микропроцесорот и претставуваат еден вид повратна информација за тоа како се извршени претходно зададените наредби. Дали ќе се пишува или ќе се чита податок од меморијата, одлучуваат две контролни линии од контролната магистрала: $\overline{RD}$ и $\overline{WR}$**.** Тие се кратенки од зборовите READ (читај) и WRITE (пишувај). Негацијата врз нив значи дека се активни на логичка нула. Не смее двете линии да бидат активни истовремено (или ќе читаме или ќе пишуваме). Друга поважна контролна линија е линијата со ознака $IO/\overline{M}$ **.** Оваа ознака е кратенка од Input Output/ $\overline{\text{Memory}}$ што во превод значи влезно-излезни уреди/меморија. Кога овој сигнал $IO/\overline{M}$ =1, тогаш микропроцесорот комуницира со некоја влезно-излезна единица (периферен уред), а кога $IO/\overline{M}$ = 0, тогаш микропроцесорот комуницира со меморијата.

Магистралите се вградени во самата матична плоча на компјутерот и нивниот број е неменлив. Зголемувањето на бројот на линии по магистрала предизвикува големо зголемување на цената на компјутерот. Мора да се прави компромис меѓу цената и ширината на магистралите. На пример, колку повеќе адресни линии имаме на располагање толку поголем мемориски простор може директно да се адресира. Ако магистралата има n адресни линии, тогаш може да се адресираат вкупно **2^n мемориски локации.** Со секоја нова генерација на компјутери бројот на линии по магистрала расте, како адресните така и податочните и контролните.

1.3. Архитектура и функционирање на компјутерски систем

Блоковската шема што е прикажана на слика 1.3. ги дава сите поважни хардверски и софтверски компоненти во составот на еден компјутер. Магистралната структура прикажана на слика 1.4. ја дава врската меѓу различните компоненти и трансферот на податоци. Но, сепак, блоковската и магистралната структура не се доволни за да се разбере како всушност функционираат компјутерите. Компјутерската архитектура е сложена и потребни се многу труд и време за анализа на истата.

Основната архитектурата на микрокомпјутер е прикажана на слика 1.5. За да се даде функционален опис, мора да се навлезе во внатрешноста на микропроцесорот и неговата работна, RAM-меморија. Кога ќе отвориме некоја програма, истата се пренесува од трајната меморија во RAM-меморијата и во **програмскиот бројач** се запишува почетната адреса на програмата. Од слика 1.5. може да се види дека програмата е веќе во RAM-от. Секоја локација од RAM-от има свој реден број кој се нарекува **адреса.** Без адреса микропроцесорот не може да пристапи до локацијата и да ја прочита нејзината содржина. Кога ќе започне извршувањето на програмата, микропроцесорот ја зема адресата од програмскиот бројач и ја испраќа до RAM-от. Тој се пребарува, ја пронаоѓа локацијата и нејзината содржина се пренесува до микропроцесорот преку податочната магистрала. Ако содржината на локацијата е инструкција, тогаш таа се сместува во **инструкцискиот регистар**, од каде се носи до управувачката единица.

Слика 1.5. Организација и функционален опис на компјутер

Во **управувачката единица** се случува магијата.Софтверот го активира хардверот, односно кодирана инструкција одредува кои пинови ќе се активираат или нема да се активираат, кои хардверски ресурси ќе се вклучат или не. Кодираната инструкција, всушност, избира една таканаречена микроинструкција. Бројот на микроинструкции зависи од бројот на можни инструкции кои микропроцесорот може да ги извршува. Самата микроинструкција е составена од илјадници **контролни битови**, нули и единици. На пример, нулата може да изврши исклучување, а единицата може да изврши

вклучување на некоја компонента. Контролните битови се испраќаат преку контролната магистрала. На слика 1.5. се прикажува дека после инструкциите 1 и 3 следуваат податоци 1 и 3, а за извршување на инструкцијата 2 не е потребен податок. Кога инструкцијата 1 ќе се пренесе од RAM-меморијата во микропроцесорот, програмскиот бројач се зголемува за еден, односно тој ќе ја содржи адресата на податокот 1. Откако ќе се декодира инструкцијата 1, микропроцесорот повторно ја испраќа адресата содржана во програмскиот бројач до RAM-меморијата, со цел да го добие податокот број 1. Кога податокот број 1 ќе дојде до микропроцесорот, тој се внесува во општиот регистар и од таму влегува во **аритметичко-логичката единица** која конечно ја извршува инструкцијата. Доколку добиениот резултат треба да се пренесе во RAM-от, тогаш резултатот се внесува во меморискиот податочен регистар, а адресата на локацијата во која треба да се смести податокот се внесува во меморискиот адресен регистар. Откако ќе се меморира добиениот резултат, микропроцесорот пристапува кон инструкцијата број 2, преку адресата содржана во програмскиот бројач.

Циклусот **пренеси-декодирај-изврши** се повторува за секоја инструкција сè додека микропроцесорот не дојде до крајот на програмата. При извршувањето на циклусите пренеси-декодирај-изврши, многу се важни времето и усогласеноста на брзината (синхронизација) на хардверските компоненти. Тајминг значи распределба на време и дефинирање на активности кои треба да се извршат, за секоја компонента посебно. Тајмингот за работа на микропроцесорот се објаснува преку временски дијаграми кои, всушност, покажуваат кои пинови се вклучуваат, а кои се исклучуваат во секоја наносекунда од времето.

Програмата се внесува во меморијата на компјутерот преку **влезните уреди**. На пример, програмата ја пишуваме преку тастатурата и текст едиторот. Таа автоматски се меморира во RAM-меморијата за да ја зачуваме на крајот во трајната меморија, на пример хард-дискот. Често пати податоците кои се потребни за извршување на програмата не се дел од самата програма, туку се внесуваат преку некој влезен уред. Резултатите добиени од извршувањето на програмата, корисникот ги следи преку **излезните уреди**. За компјутерот, кориснички информации се кодови, комбинации од нули и единици. На пример, еден знак (бројка, буква или алфанумерички симбол) за компјутерот претставува информација од еден бајт или еден пиксел од екранот е информација од три бајти.

Архитектурата на микропроцесорот е комплексна и понекогаш апстрактна. Хардверските компоненти формираат една функционална целина, систем чија ефикасност во работењето зависи од многу фактори: хиерархија, поврзување, комуникација, протоколи, тајминг, сигнализација, синхронизација, дефинирање на влезни и излезни струи и напони за пиновите на интегрираните кола итн. Познавањето на овие поими може многу да помогне при програмирањето и употребата на микрокомпјутерите во процесното управување.

1.4. Микрокомпјутерски системи и нивни карактеристики

Од компјутерските системи, микрокомпјутерите се со најслаби перформанси и најмала големина, но за обичниот човек се од најголемо значење бидејќи без нив крајните корисници не можат да ги користат бенефициите на глобалната компјутеризација и дигитализација. Основна поделба на микрокомпјутерските системи е на персонални компјутери и компјутери на плочка (анг. SBC-Single Board Computer). Во групата на персонални компјутери спаѓаат: десктоп, лаптоп, таблет и мобилен телефон. Сите набројани микрокомпјутери се наменети за еден корисник.

Слика 1.6. Видови персонални микрокомпјутери

Основни карактеристики на персоналните компјутери се: големината и преносливоста, конфигурацијата и перформансите, цената на чинење, можностите за поврзување, надградбата, достапен софтвер и др. **Големината и преносливоста** се заемно зависни карактеристики. Доколку компјутерот е со големи димензии и тежина, тој не може да се пренесува туку се поставува на едно место и не се поместува. При изборот на работното место треба да се внимава компјутерот да биде заштитен од нечистотии и влага и да има доволно простор за правилна вентилација. Зборот **конфигурација** значи избор на хардверски компоненти кои ќе бидат компатибилни и ќе формираат функционална целина. Корисниците најчесто бараат информации за типот и карактеристиките на микропроцесорот, видот и капацитетот на привремената и трајната меморија, брзината на работа и меморијата на графичката картичка, резолуцијата на екранот итн. Од конфигурацијата зависат **перформансите** на компјутерот, како што се: брзината, сигурноста, видот на софтверот што може да го извршува, издржливоста и векот на употреба. Од конфигурацијата зависи и цената на чинење. Интерфејсот значи начин на поврзување на компјутерот со други електронски уреди, како што се музички системи, видеосистеми, телевизори, компјутерски мрежи, интернет мрежи, електронски уреди и слично. За успешно поврзување на компјутерот, потребно е тој да располага со соодветни конектори и софтвер за комуникација.

Надградбата зависи пред сè од можностите на матичната плоча. На пример, може да се надгради RAM-меморијата, да се вградат две трајни

меморији (хард-диск и SSD), да се замени мрежната картичка со нова побрза картичка итн.

Слика 1.7. Можности за поврзување на микрокомпјутер

Брзиот напредок на технологијата го смени или поточно го прошири значењето на поимот микрокомпјутер. Порано овој поим тесно се поврзувал со поимот персонален компјутер. Но, денес ова е значително променето, што се должи пред сè на појавата и развојот на вградливите системи (анг. Embedded System). Вградливите системи можат да бидат базирани на микроконтролери или Arduino и Raspberry Pi. Микрокомпјутерите на плочка се основа на вградливите системи.

Слика 1.8 Вградливи микрокомпјутерски системи

Изработката на микроконтролерските базирани вградливи системи може да се подели на хардверски и софтверски дел. Прво треба да се изработи хардверскиот дел односно електрониката на уредот. Според намената на уредот и неговата функционалност се избираат електронските компоненти: отпорници, кондензатори, диоди, интегрирани кола, сензори, извршни единици, напојување, вид на микроконтролер и соодветното подножје за истиот. Потоа се изработува плочката со печатеното коло, која ги поврзува електронските компоненти во една функционална целина. Хардверскиот дел завршува откако

елементите ќе се поврзат на плочката. Потоа започнува софтверскиот дел. За да корисничката програма се имплементира во микроконтролерот потребен е специјален уред таканаречен програматор кој се поврзува со компјутер преку USB кабел. Микроконтролерот се поставува во подножјето на програматорот и откако ќе се испрограмира се префрла во подножјето на корисничкиот уред.

Покрај микроконтролерските базирани вградливи системи се почесто се користат Arduino и Raspberry Pi базираните системи. Arduino претставува микроконтролерска базирана развојна платформа со големина на кредитна картичка. Arduino платформите претставуваат електронски плочки коишто освен микроконтролер содржат сериски програматор, кристален осцилатор и аналогни и дигитални влезови и излези. Програмерот не треба да се грижи за изборот на програматор, а вградените влезно-излезни пинови дозволуваат едноставно поврзување со сензорите и извршните уреди. За изработка на електронскиот уред, кој треба да го управуваме, наместо печатена плочка може да се користи експериментална протоплочка (анг. Protoboard), која ни дозволува да експериментираме со дизајнот на уредот.

Иако по големина се речиси исти со Arduino, Raspberry Pi има многу помоќен процесор и поголем капацитет на RAM меморијата. Наместо микроконтролер тој содржи специјален чип со процесор и вградена графичка картичка и RAM меморијата е посебен чип на електронската плочка. За разлика од Arduino тој има свој оперативен систем.

Ние ќе ги проучиме Arduino платформата и Raspberry Pi и ќе се запознаеме со нивните составни делови, функционалноста, поврзувањето со влезно-излезните уреди, начинот на нивно програмирање.

1.5. Сличности и разлики кај различни типови микрокомпјутерски системи

Накратко ќе ги објасниме карактеристиките на секој тип на микрокомпјутер.

Десктопот е најстар тип на персонален компјутер. Негов главен недостаток е непреносливоста. За сметка на својата големина, десктопот има многу предности. За помала цена добиваме компјутер со посакуваните перформанси и квалитет. Корисникот може сам да ја избере конфигурацијата на својот десктоп. Постои можност да се надградува и многу лесно да се заменат одредени компоненти во случај на дефект. Десктопот ги има сите видови компјутерски порти (VGA, LAN, HDMI, USB, внатрешни PCI-слотови) и претставува компјутер со најмногу можности за поврзување со други електронски уреди. Потрошувачката на енергија е поголема во однос на лаптопот, но ладењето и вентилацијата на чиповите е на повисоко ниво поради што неговиот животен век е подолг.

Основна предност на **лаптопот** е неговата преносливост. Ова значи дека при употребата на лаптоп, корисникот воопшто не е ограничен со просторот туку само со времето бидејќи лаптопот има батерија со ограничено времетраење. Речиси за секоја конфигурација на десктоп постои иста таква конфигурација на лаптоп. Недостаток на лаптопот се готовите конфигурации, односно обичните купувачи не можат сами да ги изберат хардверските компоненти според своите потреби. Се разбира, конфигурација по нарачка е возможна, но тоа многу ја зголемува и така високата цена. Исто така, самото сервисирање на лаптопот е покомплицирано и чини повеќе во однос на десктопот. Во табелата 1.1 е направена споредба меѓу лаптопот Dell 12.5" Latitude 7290 и микро десктопот Dell OptiPlex 5070. Иако се со слични перформанси, лаптопот е двапати поскап во однос на десктопот.

	Dell 12.5" Latitude 7290	Dell OptiPlex 5070
Процесор	1,7 GHz Intel Core i5-8350U Quad-Core	2 GHz Intel Core i7-9700T 8-Core
Максимална брзина	3,6 GHz	4,3 GHz
Инсталиран RAM	8 GB	8 GB
Максимален капацитет на меморија	16 GB	32 GB
Графичка картичка	Intel UHD Graphics 620	Intel UHD Graphics 630
SSD-меморија	1 x 256 GB	1 x 256 GB
Порти	2 x USB Type-A 1 x HDMI 1.4	5 x USB Type-A 2 x DisplayPort 1.2
Вграден звучник и микрофон	√	×
Вградена веб-камера	√	×
Потрошувачка	65 W	90 W
Тежина	1,4 Kg	5,8 kg

Табела 1.1. Споредба на микрокомпјутер десктоп и лаптоп

Таблетот има многу малку можности за поврзување со други уреди. Вообичаено, таблетите имаат само еден мини USB-приклучок. За интернет врска постои само Wi-Fi поврзување и тој не може да се приклучи во локална компјутерска мрежа. Малите димензии, неговата лесна преносливост и малата потрошувачка на енергија се главни предности на таблетите. Поради екранот на допир постои можност за користење на паметно пенкало, односно корисникот може рачно да црта и да пишува на екранот.

Денес речиси да нема човек во модерниот свет кој не користи **паметен телефон**. Нивна основна намена е користење на услугите на мобилната телефонија, но паралелно се користат за извршување на многу други апликации: пребарување на интернет, аудио и видеопродукција и репродукција и многу други кориснички програми. Малата брзината на работа, ограничените мемориски капацитети, малиот екран ги прават мобилните телефони непогодни

за професионална работа. Тие можат да извршуваат многу апликации, но се непогодни за програмирање и нивните можности за создавање нов софтвер се неспоредливи со оние на десктопот и лаптопот.

Микрокомпјутерите на плочка станаа достапни за обичните корисници во последниве петнаесет години. Претходно, истите се користеа во индустријата и системите за автоматско управување. Во 2005 година беше лансирана **Ардуино** хардверската платформа со слободен софтвер, со цел да се поедностави процесот на електронски прототипни иновации. Тоа им овозможи на обичните луѓе со малку или без технички предзнаења да изградат интерактивни производи. Raspberry Pi се најголемите конкуренти на Arduino. Првата генерација на Raspberry Pi излезе на пазарот во 2012 година, во Англија. Доказ за неговата популарност е тоа што во првите шест месеци беа продадени околу половина милион примероци.

	Raspberry Pi	Arduino
предности	64- битен процесор со работна фреквенција од 1.5GHz и можност за истовремено извршување на повеќе задачи	Лесно поврзување со аналогни сензори и прецизна контрола на работа на мотори
	Вграден мрежен приклучок, можност за безжично поврзување преку Wi-Fi или Bluetooth	Голем избор на додатоци за и зголемување на функционалноста
	Функционален оперативен систем	Програмите ги извршува веднаш по вклучување на напојувањето без дополнителни нагодувања
	Содржи аудио излез, приклучок за камера, HDMI излез и неколку USB	Ниска цена на чинење
	Одличен избор за проекти како што се роботи или видеоигри	Извршува едноставни апликации во реално време
	Може да се програмира во повеќе програмски јазици	Поедноставен хардвер и софтвер
недостатоци	За поврзување на електронски компоненти потребен е дополнителен софтвер	Работна фреквенција од 16MHz, мала брзина на работа и извршува само една програма
	Висока цена на чинење	Без Ethernet додаток не може да се поврзе со интернет мрежата
	Поголемо загревање при употреба	За програмирање се користи само програмскиот јазик C/C++

Табела 1.2. Споредба меѓу Arduino и Raspberry Pi

Вградливите микрокомпјутерски системи се многу популарни поради нивната мала цена на чинење, флексибилност, едноставно програмирање. Многу е важно паметно да се избере развојната платформа за реализација на дадена практична вежба или проектна задача. Во тоа може да ни помогне табела

1.2. во која споредбено се дадени предностите и недостатоците на Arduino и Raspberry Pi. Raspberry Pi е софтверски ориентиран односно задачите ги извршува пред сè со примена на програми, а Arduino е хардверски ориентиран. На пример, доколку сакаме да направиме неколку фотографии, истите да ги обработиме и објавиме на некоја веб страна тогаш Raspberry Pi нуди поедноставни решенија. Но, доколку сакаме да контролираме брзина на вртење на мотор на еднонасочна струја тогаш платформата Arduino Uno е посоодветна поради поголемата брзина на реакција и поголемата прецизност. Arduino и Raspberry Pi се дел од платформата Интернет на нештата (анг. IoT- Internet of Things). Оваа платформа е нов тренд во технологијата и се занимава со поврзување на сензори и уреди, нивно вмрежување со примена на различни видови комуникациски протоколи и анализа на податоци. Оваа платформа не е предмет на наша анализа, но нејзината поврзаност со Arduino и Raspberry Pi е неминовност.

1.6.Примена на микрокомпјутерски системи

При изборот на персонален компјутер, многу е важно да се знае неговата **намена.** Апликациите за обработка на текст, обработка на звук или видео, создавање апликативен софтвер, графички дизајн, различни видови вмрежувања и комуникации, автоматизација и други, бараат компјутери со различни перформанси.

Изборот на конфигурации е огромен, од Pentium до Intel i9 процесори, RAM од 4GB до 128GB, трајни мемории HDD или SDD, од 500GB до 20TB, графички картички од NVIDIA, Intel, AMD. Поради брзиот развој на компјутерите овие податоци постојано се менуваат односно зголемуваат.

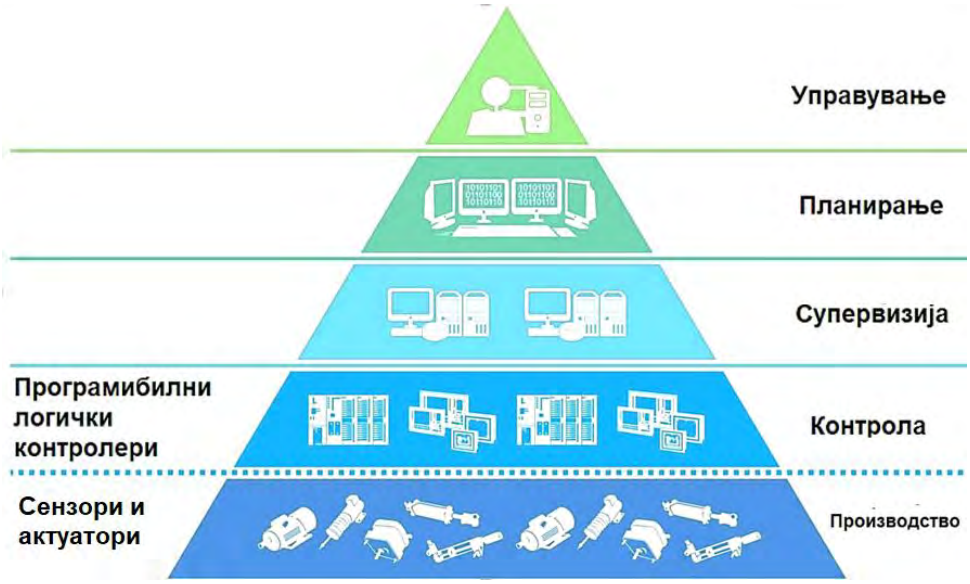
Самите компјутери се класифицираат во неколку категории: за домашна употреба и за студенти, бизнис-компјутери, компјутери за професионалци и компјутери за видеоигри. **Компјутерите за домашна употреба** се препорачуваат за корисници кои најчесто користат апликации како што се Microsoft Office или пребарување на интернет. **Професионалните компјутери** се наменети за инженери, архитекти, дизајнери. Одлични се за анимации и за симулации. Овие компјутери се користат и како сервери за вмрежување на повеќе персонални компјутери во помали компании. **Графичкиот дизајн и видеоигрите** бараат брзи процесори, многу моќни графички картички и меморија со голем капацитет. Се препорачува процесорот да биде шест јадрен со минимална работна фреквенција од 3GHz, RAM меморијата да има капацитет од 16GB, графичката картичка да има минимум 2GB видео RAM меморија и наместо хард-диск да се користи SSD меморија. Секој пиксел од екранот зафаќа меморија од 3 бајти, по еден бајт за секоја од основните бои RGB (red-green-blue). HD (high definition) резолуцијата значи 1920×1080 пиксели, или приближно

два милиона пиксели. Работната фреквенција од 60 Hz значи обновување на сликата 60 пати во една секунда. Ако ги помножиме овие бројки, ќе добиеме пропусен опсег од 360MB/s. Овие бројки ја објаснуваат високата цена на компјутерите за видеоигри и за обработка на видео материјали. Овие компјутери се со подобар дизајн, звучни и светлосни ефекти, со повеќе метални од пластични делови и изработени се од поквалитетни материјали, така што треба да имаат подолг век на траење.

Постојат бесконечно многу примери за примена на микрокомпјутерите. Ќе наведеме неколку од нив.

- **Медицина**. На почетокот, компјутерите биле користени за складирање и обработка на медицински податоци поради подобра прегледност и достапност. Во оваа категорија спаѓа и телемедицината, која дозволува грижа за пациентите од далечина. Денес не може да се замисли дијагностика и навремено откривање на болести без примена на компјутерска томографија, магнетна резонанца и 3D-рендгенски зраци. Хирурзите сè повеќе користат виртуелни модели за симулација на хируршки процедури. Вградувањето чипови во човечкото тело е веќе едноставна постапка. На пример, пејсмејкерот претставува чип што генерира електрични импулси со точно дефинирана амплитуда и фреквенција и ја стимулира работата на срцето. Овој чип има сопствено напојување и истовремено може да се користи и за следење на работата на срцето.
- **Автомобилска индустрија**. Во еден современ автомобил има над 70 управувачки единици и 100 сензори. Од иновациите, 80% се засноваат на примена на микрокомпјутери на плочка. Најважен микроконтролер е електронската управувачка единица (анг. ECU – Electronic Control Unit). Главни влезни големини се моќноста на моторот и бројот на вртежи. Други влезни величини се: температурата и протокот на вшмуканиот воздух, температурата на течноста за разладување, положбата на вратилата итн. Излезни величини се количеството на гориво што се вбризгува во еден циклус и моментот на палење. Управувачката единица одредува колку долго вбризгувачите ќе останат отворени (околу 6 до 9 милисекунди, од 600 до 3000 пати во минута), со што се контролира количеството на гориво. Времето на палење на горивото зависи од тоа колку често се вклучуваат и исклучуваат свеќичките во моторот. Сите овие податоци се запишуваат во таканаречената матрица, врз основа на што се добиваат излезните коефициенти. Матрицата е, всушност, EEPROM-меморија и таа е различна за различни типови на автомобили. По пресметката, микрокомпјутерот на плочка ги пренесува добиените вредности до актуаторот – вбризгувачите. Доколку температурата е повисока од предвидената, се продолжува времето на отвореност на вбризгувачот (поголем воздушен проток)
- **Автоматизација на индустријата.** Автоматското управување е сложен процес и може да се подели на неколку нивоа. Првото ниво се сензорите и извршните единици што ги следат и ги менуваат производствените

параметри. Второто ниво се програмабилните логички контролери кои во согласност со програмата, делуваат моментално, без да анализираат. Третото ниво на супервизија ги обработува податоците, изработува математички модели, ги проучува машините и врши оптимизација, изготвува извештаи. Потребен е брз компјутер што брзо ќе ги обработува податоците, голема трајна меморија за чување на минатите вредности и прегледна графика за полесно следење на параметрите.



Слика 1.9. Примена на микрокомпјутери во автоматиката

- **Банкарство и финансии**. Многу финансиски институции вложуваат капитал во дигиталните технологии. Банкарските трансакции се извршуваат преку мобилни и веб-платформи, без оглед во која земја престојувате. Компјутерите се користат за изработка на статистички извештаи, следење на светските трендови и повторна модернизација на електронското банкарство. Такви трендови се вршење трансакции со мобилен телефон наместо со платежна картичка и користење биометриска авторизација заради поголема сигурност и заштита од злоупотреба (користење отпечаток од прст или компјутерско препознавање на лик).

Секоја област си има свои карактеристики и потребна е специјализација за да се стане експерт за компјутери во таа област. Ние ќе се запознаеме со основите на хардверот и софтверот на персоналните компјутери и микрокомпјутерите на плочка, што е многу важно за натамошниот професионален развој, според желбите и можностите на поединецот.

Заклучоци

Под **компјутерски систем** подразбираме систем составен од хардверски и софтверски компоненти, кој врши прием, обработка, чување и прикажување на податоци и информации.

Генерално, компјутерите се поделени во четири групи: **микро, мини, макро и суперкомпјутери**. Микрокомпјутерите се компјутери наменети само за еден корисник. Миникомпјутерите можат да ги користат повеќе корисници истовремено. Макрокомпјутерите се наменети за апликации со поголем степен на интерактивност.

Најважни хардверски компоненти се: микропроцесор, мемории, магистрали и периферни уреди. Микропроцесорот обработува податоци, мемориите ги чуваат, магистралите ги пренесуваат и периферните уреди вршат претворање на корисничките сигнали.

Најважни софтверски компоненти се: оперативен систем, развојни програми и кориснички програми.

Магистралната структура ни покажува како се поврзани хардверските компоненти и како се врши трансферот на податоци.

Како функционира микрокомпјутерот може да се објасни преку **циклусот пренеси-декодирај-изврши**. Инструкциите и податоците се пренесуваат од RAM-от до микропроцесорот. Со декодирањето се генерираат контролни битови кои го вклучуваат и исклучуваат хардверот. Извршувањето на инструкциите се врши во аритметичко-логичката единица.

Основни карактеристики на микрокомпјутерите се: големината и преносливоста, конфигурацијата и перформансите, цената на чинење, можностите за поврзување, вградливоста, надградбата, програмибилноста и достапен софтвер и др.

Предност на десктопот е неговата цена, надградба и лесно одржување. **Предност на лаптопот** е неговата преносливост.

Вградливите микрокомпјутерски системи се многу популарни поради нивната мала цена на чинење, флексибилност, едноставно програмирање.

Raspberry Pi микроконтролерот на плочка има свој оперативен систем, графичка картичка и можност за мрежно поврзување.

Според намената, персоналните компјутери се делат на компјутери за домашна употреба и професионални компјутери.

Прашања и задачи

1. Што подразбираме под поимот „микрокомпјутерски систем“?
2. Која е суштинската разлика меѓу микрокомпјутер и микроконтролер?
3. Набroј неколку примери за примена на миникомпјутер!
4. Набroј ги основните структурни блокови на компјутерски систем!
5. Која е функцијата на оперативниот систем?
6. Какви видови магистрали постојат и какви битови пренесуваат истите?
7. Објасни го циклусот пренеси-декодирај-изврши!
8. Зошто е потребно адресирање на мемориските локации на RAM-меморијата?
9. Која е функцијата на управувачката единица!
10. Набroј ги основните карактеристики на микрокомпјутер!
11. Кои се предностите и недостатоците на лаптопот во однос на другите видови персонални компјутери?
12. Кои се предностите и недостатоците на микрокомпјутерот на плочка Arduino Uno во однос на микрокомпјутерот Raspberry Pi?
13. Објасни ја функцијата и значењето на микрокомпјутерот ECU (Electronic Control Unit),вграден во поновите автомобили!
14. Кои елементи треба да ги содржи конфигурацијата на еден персонален компјутер?
15. Набroј неколку практични примери за вградливи микрокомпјутери!

2. Хардвер на персонален компјутерски систем (PC)

2.1. Структура, улога и функција на хардверски компоненти

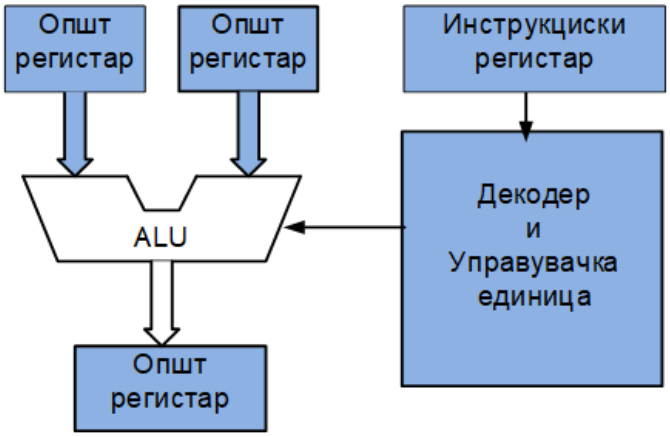
Во однос на минатото, денес изборот на хардверски компоненти и монтажата на десктоп компјутерите е едноставна постапка. За составување на еден стандарден персонален компјутер потребни се следниве основни компоненти: куќиште, уред за напојување, матична плоча, микропроцесор, ладилник за процесорот, RAM меморија, хард-диск или SSD меморија, графичка картичка (која може да биде интегрирана во самиот процесор), звучна картичка, мрежна картичка, CD ROM уред, тастатура, глувче, монитор, кабли за поврзување и навртки (слика 2.1).

Слика 2.1. Хардверски компоненти на десктоп персонален компјутер

Ние ќе се запознаеме со функцијата, составните делови, карактеристиките и начинот на поврзување на сите овие хардверски компоненти. Подоцна ќе се запознаеме со видовите на оперативни системи, нивната инсталација и накратко ќе се запознаеме со неколку најчесто користени кориснички програми. Овие предзнаења се неопходни доколку сакаме успешно да изградиме еден персонален компјутер.

2.1.1. Микропроцесор

Микропроцесорот е „**мозокот**“ на компјутерот. Тој има две основни функции: извршува програми и управува со другите уреди на матичната плоча. За микропроцесорот програмите се множество од инструкции, наредени во последователни мемориски локации. Микропроцесорот ги **извршува програмите** инструкција по инструкција. Откако ќе ја обработи тековната инструкција, ја презема следната инструкција од RAM-меморијата, ја обработува, и оваа постапка се повторува сè додека процесорот не дојде до крајот на програмата. **Управувачката функција** го прави микропроцесорот господар на матичната плоча (анг. master). Тој прибира информации од сите уреди во компјутерот, ги обработува, ги процесира и потоа поттикнува одредени активности, со цел да се обезбеди сигурна работа на целиот компјутер. На слика 2.2. прикажана е основната организација на микропроцесорот. **Составни делови** на еден микропроцесор се: регистри, аритметичко-логичката единица и управувачката единица со декодерот. Регистрите се брзи мемориски локации во внатрешноста на микропроцесорот кои ги содржат податоците и операцискиот код на тековната инструкција. Општите регистри ги содржат податоците за обработка и обработените податоци, а кодовите потребни за извршување на тековната инструкција се сместуваат во специфичен регистар – инструкциски регистар.



Слика 2.2. Блок-шема на основен модел на микропроцесор

Во аритметичко-логичката единица се извршуваат сите аритметички и логички инструкции. Управувачката единица прима информации од сите уреди и потоа одлучува, менува содржина на регистри, менува логичка состојба на пинови, повикува програми, активира периферни уреди итн. Нулите и единиците што излегуваат од управувачката единица ги вклучуваат или ги исклучуваат хардверските компоненти. Бројот и функцијата на контролните битови треба да се предвиди уште при дизајнирањето на микропроцесорот.

Најважни карактеристики на микропроцесорот се: работната фреквенција, ширината на податоците, проточниот концепт, комплексноста на

инструкциското множество, бројот на јадра, распоредот на пиновите и др.

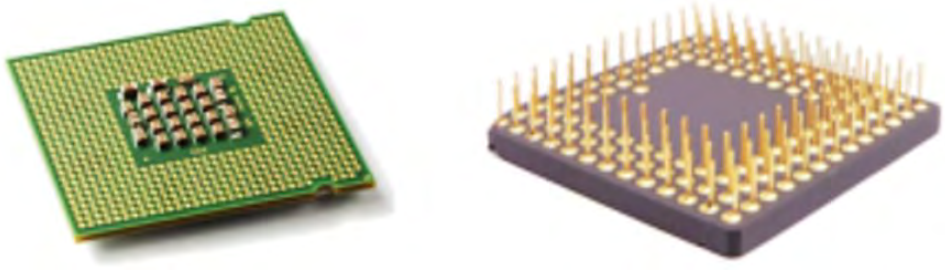
Работна фреквенција е фреквенцијата на такт сигналот. Тактовите се важни бидејќи времетраењето на секоја операција се мери во цел број на такт сигнали. На пример, за да се прочита еден бајт од RAM-меморијата, потребни се три такта. Во првиот такт, микропроцесорот испраќа адреса, во вториот се пребарува меморијата и во третиот такт микропроцесорот го добива саканиот податок. Тоа што за нас е секундата, за микропроцесорот е наносекундата (милијардити дел од секундата).

Во зависност од ширината на податоците (анг. data width), микропроцесорите може да бидат 8, 16, 32, 64 и 128-битни. На пример, кај 64-битните процесори постојат 64 податочни пинови, во општите регистри може да се сместат 64 битови, аритметичко-логичката единица извршува операции со 64-битни податоци и исто така податочната магистрала на матичната плоча содржи 64 водови. Да споменеме дека со мултиплицирање на широчината на податоците се мултиплицираат и хардверските ресурси на процесорите.

Процесорите можат да имаат комплексно или редуцирано инструкциско множество. Денес речиси сите процесори се RISC (анг. Reduced Instruction Set Computer) и овие процесори хардверски извршуваат едноставни инструкции и се многу брзи.

Повеќејадрените процесори значат вградување на повеќе процесори во едно исто интегрирано коло. Едноставно кажано, две глави размислуваат подобро од една, четири раце се повредни од две. Двете јадра си ги делат апликациите и на таков начин се скратува времето на извршување.

Пиновите се метални продолженија на внатрешните водови на интегрираните кола. Поновите генерации процесори имаат околу 2000 пинови со пречник од 0,3 mm. Кај постарите процесори, пиновите се на процесорот, а отворите за нив се на подножјето на матичната плоча. Овој концепт е познат под кратенката **PGA** (анг. Pin Grid Array). Спротивен концепт е **LGA** (анг. Land Grid Array),каде што отворите се на процесорот, а пиновите се на подножјето.



Слика 2.3. Распоред на пиновите на процесори

Најголеми производители на процесори се компаниите Intel и AMD. Процесорите на AMD се многу популарни меѓу љубителите на видеоигри. За помала цена нудат добри перформанси, што, пред сè, се должи на вградената графичка картичка на ист чип заедно со микропроцесорот. Процесорите на Intel се изработени во подобра технологија и поради тоа се поотпорни на загревање и имаат подолг век на траење.

2.1.2. Мемории

Сите податоци и програми што ги извршува процесорот мора да бидат сместени во меморииte. Најважни **карактеристики на меморииte** се нивниот капацитет и брзината. Капацитетот на меморииte се мери во бајти(B) или поголеми единици за капацитет се: килобајти(KB), мегабајти(MB), гигабајти(GB), терабајти(TB). Брзината на меморијата се мери преку времето на пристап или нејзината работна фреквенција. Времето на пристап е потребното време меморијата да се пребара и да го пронајде саканиот податок. Работната фреквенција на меморијата ни покажува колку битови може да прими или да испрати меморијата во една секунда.

Компјутерите содржат различни видови мемории. Мемориската организација подразбира поделба на функциите за секој вид меморија и начинот на пристап до саканиот податок. На сликата 2.4. е прикажана пирамидата од мемории. Како се симнуваме надолу, така се зголемува капацитетот на меморииte, а се намалува нивната брзина на работењето. Поголемата брзина на работата повлекува зголемување на цената по еден бајт. Колку се поблиску меморииte до врвот, толку се поблиску до микропроцесорот.

Слика 2.4. Пирамида на мемории

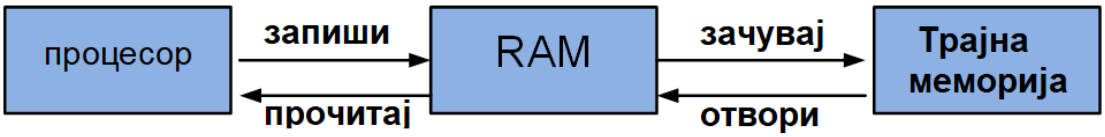
Регистрите служат за привремено чување на податоците, пред тие да бидат обработени, но и за сместување на резултатите веднаш по нивното добивање. Регистрите се најбрзи мемориски локации во компјутерот, но нивниот капацитет изнесува само неколку бајти.

Со цел да се намали времето на чекање и да се зголеми брзината на процесорот, се користи **кеш-меморија.** Кеш-меморијата е многу брза RAM-меморија. Потребните податоци од побавните мемории однапред се донесуваат и се сместуваат во кеш-меморијата и таа претставува еден вид „чекалница“ за

микропроцесорот. Во кеш-меморијата може да се чуваат податоци што микропроцесорот ги користи почесто, а не може да ги смести во своите регистри. Кеш-меморијата може да биде во самиот мемориски чип или надвор од него.

RAM е кратенка од англиските зборови Random Access Memory, што во превод значи **меморија со случаен пристап**. Имено, процесорот има ист пристап до сите локации од меморијата или ниедна локација не е со помал или со поголем приоритет. RAM е меморија од која може истовремено да се читаат, но и да се пишуваат нови податоци. Тоа е меморија со променлива содржина и претставува **работна меморија**, еден вид работен лист на кој микропроцесорот привремено ги запишува сите податоци што ќе му бидат потребни за успешно реализирање на дадената програма. Во RAM-меморијата се чуваат програмите што моментално се извршуваат во процесорот. RAM-меморијата е позната и под името примарна меморија, поради нејзината важност. RAM е **привремена меморија**. Доколку се исклучи напојувањето, податоците од RAM не можат да се обноват по повторното вклучување. Поради тоа, пред да го исклучиме компјутерот, податоците од RAM треба да се сместат на хард-дискот, преку притискање (анг. click) на иконата SAVE. Ова е прикажано на слика 2.5.

RAM-меморијата го снабдува процесорот со информации, а пак таа добива информации од некоја трајна меморија, како на пример хард дискот. Програмите се чуваат во трајната меморија и тие се пренесуваат во RAM кога треба да бидат обработени. Доколку RAM-меморијата е поголема, тогаш одеднаш ќе земе поголем дел од корисничката програма и таа ќе се обработи во микропроцесорот. Со ова се намалува бројот на преноси од RAM до трајната меморија, а тоа значи и заштеда на време. Микропроцесорот ги извршува програмите **инструкција по инструкција**. Откако ќе заврши со тековната инструкција, се повикува RAM-меморијата да ја пронајде следната. Ако се работи за линиска програма тогаш инструкциите се наредени во последователни мемориски локации (редици) од RAM-меморијата, што не е случај доколку се работи за програма со разгранување.



Слика 2.5. Функција на RAM-меморијата како работна меморија на микропроцесорот

RAM-меморијата е примарна меморија, а хард-дискот и другите надворешни мемории се секундарни. За разлика од RAM-от, кеш-меморијата и регистрите, **секундарната меморија** претставува трајна меморија. Трајни мемории **се хард-дискот, SSD-меморијата** и надворешните мемории (компакт-дискови, USB-мемории, SD-картички и др.). Хард-дискот е магнетна меморија и составена е од неколку магнетни плочи коишто ротираат и помеѓу магнетните плочи поставен е чешел со глави за читање. При изборот на хард-диск најважни

карактеристики се капацитетот и брзината за пренос на податоци. Денес на пазарот се нудат хард-дискови со капацитет од 500MB до 18TB, но овие податоци постојано се менуваат. Брзината за пренос на податоци изнесува од 70Mb/s до 150Mb/s. Таа зависи од повеќе фактори како што се: брзината на ротација, густината на податоци, големината на неговата кеш меморија, физичката големина. Брзината на ротација може да изнесува 5400, 7200, 10000, а постојат и хард-дискови со 15000 вртежи во една минута. Густината на податоци се мери во број на битови по инч квадратен и таа се зголемува како што се зголемува и капацитетот на хард-дискот. Може да се случи хард-диск со помала брзина на ротација и поголем капацитет да има поголема брзина за пренос на податоци во однос на хард-диск со поголема брзина на ротација и помал капацитет. Со цел да се зголеми брзината за пренос, денешните хард-дискови содржат и кеш меморија и тие се познати по името хибридни хард-дискови. Капацитетот на оваа кеш меморија изнесува од 32MB до 256MB. Што се однесува до физичката големина хард-дисковите можат да бидат 2,5 и 3,5 инчни. Најчесто поголемите се наменети за десктоп компјутери, а помалите за лаптопови.

Денес хард-дисковите почнаа да се заменуваат со нови SSD-мемории (анг. Solid State Drive). Тие претставуваат електронски мемории, нешто слично како RAM меморија, но со сопствено напојување за трајно чување на податоците. За персонални компјутери најчесто се избираат SSD мемориски уреди со брзина за пренос на податоци од 200MB/s до 500MB/s, а постојат и такви со брзина над 3000MB/s. Денес се нудат SSD уреди со капацитет од 128GB до 4TB. Табела 2.1. претставува споредба меѓу SSD мемориски уреди и хард-диск.

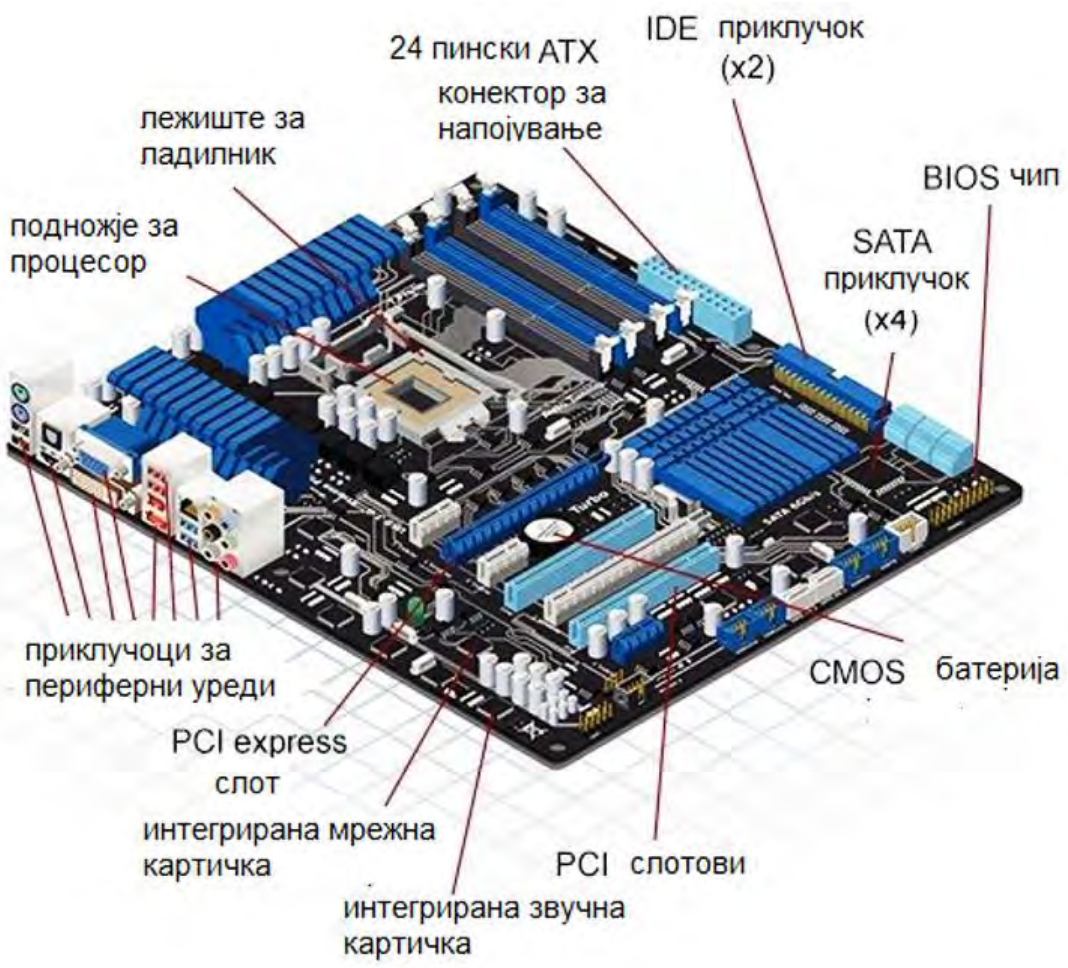
Хард диск	**SSD диск**
Мемориски уред со тврд диск (анг. HDD=Hard Disk Drive)	Уред со трајна полупроводничка меморија (анг. SSD=Solid State Drive)
Магнетна меморија	Електронска меморија
Максимален капацитет 15TB	Максимален капацитет 1TB
Појава на греење	Нема греење
Евтина меморија	Скапа меморија
Мала брзина	Голема брзина
Пократок век на траење	Подолг век на траење

Табела 2.1. Споредба на HDD и SDD мемориски уреди

Поновите компјутерски конфигурации за персонален компјутер се со хард-диск со капацитет 1TB или 2TB и SSD мемориски уред со капацитет 256MB или 512MB. SSD меморијата служи за чување и подигање на оперативниот систем, со што значително се зголемува брзината на работа на компјутерот. Во хард-дискот се чуваат поголемите кориснички датотеки.

2.1.3. Матична плоча

Матичната плоча ги **поврзува** сите хардверски компоненти во една функционална целина. Компонентите меѓу себе се физички поврзани со бакарни водови наречени магистрали. Денес матичните плочи се изработуваат во осум до десет слоеви од бакарни водови. ATX (анг. Advanced Technology eXtended) е напредна технологија, со проширување, во дизајнирањето на матичните плочи. Постојат три основни големини на матични плочи: ATX (305 mm × 244 mm), Micro ATX (244 mm × 244 mm) и Mini ITX (170 mm × 170 mm). Матичните плочи со поголеми димензии имаат поголем број на слотови и поголем капацитет на RAM-меморијата. Освен бакарни водови, матичната плоча содржи голем број различни конектори и чипови што го контролираат преносот на податоци од еден до друг уред. Накратко ќе се запознаеме со секој од нив.

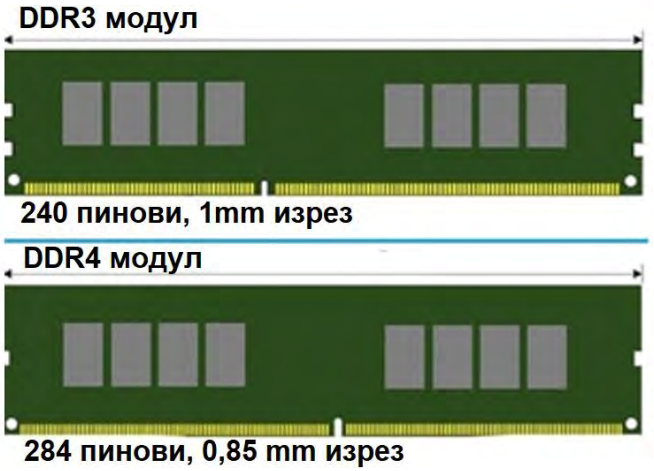


Слика 2.6. Составни делови на матичната плоча

Процесор ⟶ Изборот на матична плоча зависи од изборот на процесор, бидејќи секој тип на процесор има уникатен распоред на пиновите, што бара и соодветно подножје.

Чипсет ⟶ Чипсетот е **множество од специјални интегрирани кола коишто го контролираат протокот на податоци**. Дури и самата матична плоча се именува според видот на чипсетот вграден во неа. Секој чипсет е составен од **две основни компоненти**: северен и јужен мост. Северниот мост е графички и мемориски контролор, тој е многу брз и е директно поврзан со микропроцесорот. Јужниот мост е влезно-излезен контролор и тој комуницира со микропроцесорот, со посредство на северниот мост.

RAM ⟶



Слика 2.7. Видови RAM-мемории

Од изборот на матичната плоча зависи видот и капацитетот на RAM-меморијата. Денес најчесто се користат следниве видови на RAM-меморија: DDR4, DDR3, DDR2 и DDR. На пример, не може DDR4 RAM модул да се постави во DDR3 слот. DDRAM (анг. Double Data Rate RAM) е RAM меморија со двојно поголема количина на податоци кои се пренесуваат за времетраење на еден такт. Тие се разликуваат по бројот и распоредот на пиновите, напојувањето и работната фреквенција, односно брзината. Еден од елементите за препознавање на видот на RAM-мемориите е изрезокот поставен помеѓу пиновите.

Графичка картичка ⟶ **PCI Express×16 и AGP** се најдобри конектори за поврзување на графичките картички. Кратенката PCI Express×16 (анг. peripheral component interconnect express) значи експресна конекција на периферни компоненти со пропусен опсег 16 битови по такт. Кратенката AGP (анг. Accelerated Graphics Port) значи графичка порта со забрзување. PCI-слотовите се со помала брзина и ограничен пристап до меморијата во однос на AGP. Да споменеме дека некои матични плочи имаат

вградени графички картички, но добро е таа да содржи и дополнителни слотови, во случај на надградба на видео-системот.

PCIx1 → PCIx1-слотот се користи за поврзување на **звучни картички, мрежни картички, диск-драјв контролери** и други контролери за најразлични периферни уреди.

BIOS

CMOS → BIOS (анг. Basic Input Output System) и CMOS (анг. Complementary Metal Oxide Semiconductor) се две различни меморији на матичната плоча, кои не можат една без друга, но се разликуваат по своите карактеристики и функција. BIOS-от е ROM-меморија што во себе содржи програма којашто се извршува кога се вклучува компјутерот и од таму доаѓа кратенката што значи основен влез-излез на системот. BIOS-от врши проверка на хардверските компоненти, го лоцира оперативниот систем и го внесува во меморијата, се грижи за сигурноста на системот и врши проверка на лозинките. CMOS-от е RAM-меморија што користи сопствена батерија за напојување. Во CMOS-от се чуваат податоци за времето и датумот и податоци за нагодување на хардверските компоненти при стартување на компјутерот. Кратенката CMOS се користи за опис на полупроводничка технологија за производство на интегрирани кола.

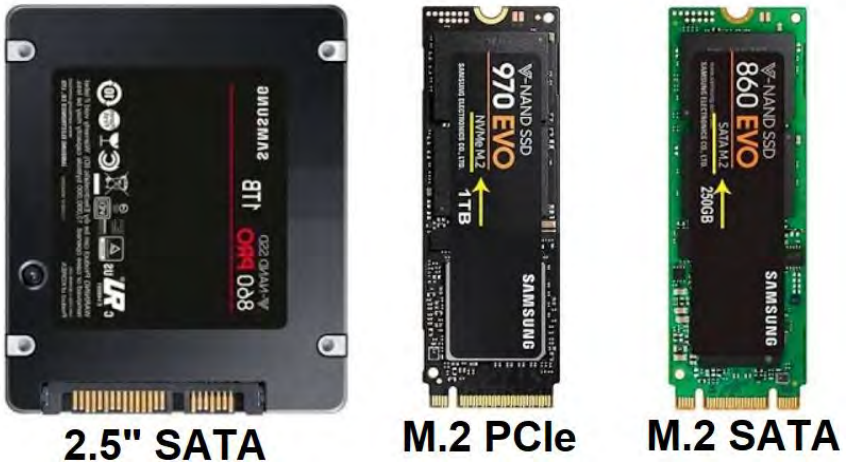
Хард диск

SDD → Поголем број матични плочи содржат два **вида SATA (анг. Serial Advanced Technology Attachment) конектори, SATA2 и SATA3**. Кратенката значи сериска напредна технологија за поврзување. Конекторот SATA2 поддржува брзини за пренос на податоци од 3Gb/s или приближно 300MB/s, а SATA3 6Gb/s или 600MB/s. Конекторот SATA2 се користи за поврзување на хард дискот и во него се приклучува податочниот кабел на хард дискот. Хард дискот поседува уште еден SATA-кабел за напојување и тој се поврзува со единицата за напојување. Конекторот SATA3 се користи за приклучување на SSD-меморијата.

M.2 SSD модули → Поновите модели на матични плочи содржат специјални конектори со ознака M.2 SSD за поврзување на специфични SSD-модули. Тие се разликуваат од SSD мемориските уреди по физичкиот изглед и конекцијата и ова е прикажано на слика 2.8.

Дури и самите SSD-модули можат да бидат со два типа на конекција, PCIe или SATA. Модулот со PCIe конекција има

еден изрез поставен меѓу пиновите, а модулот со SATA конекција има два изрези. Поради тоа при самиот избор на модул треба да се внимава од кој тип е M.2 SSD конекторот на самата матична плоча. M.2 SSD-модулот со PCIe конекција има поголема брзина на пренос.



Слика 2.8. Споредба меѓу SSD SATA мемориски уред и M.2 SSD-модули

Серискатa и паралелната порта → Серискатa и паралелната порта не се среќаваат кај поновите генерации персонални компјутери, но тие сè уште имаат своја примена. На пример, серискатa порта се користи за поврзување модеми, рутери или RSR-232 уреди, како што се индустриските управувачи. Доколку компјутерот не поседува сериска порта, тогаш потребно е да се користи USB сериски конвертор. Паралелната порта сè уште се користи за поврзување на печатачи.

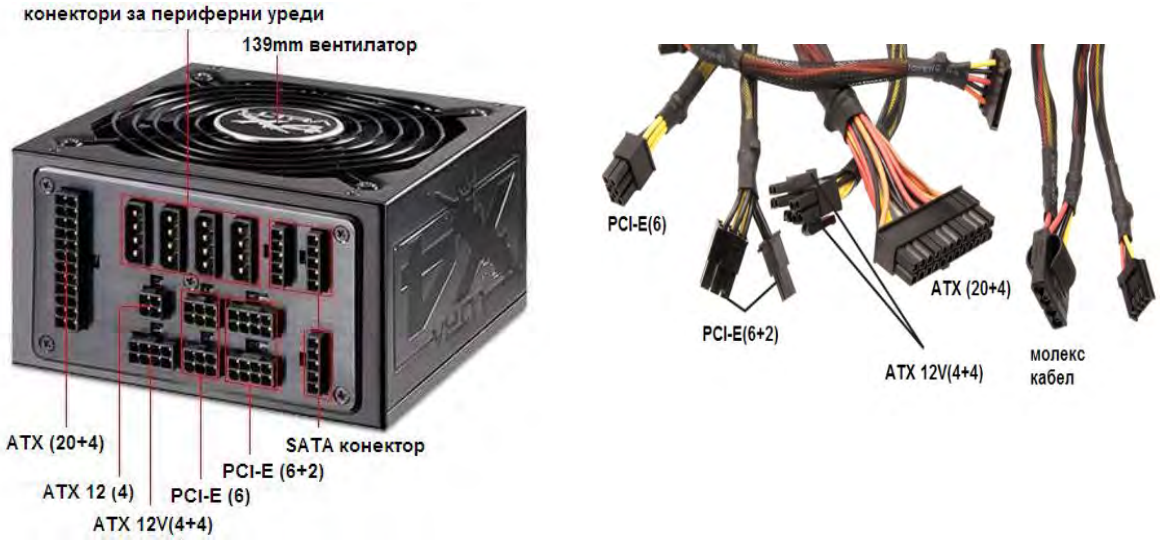
Приклучоци за видеоуреди → VGA (анг. Video Graphics Array) е аналоген видео стандард. Самата кратенка значи видео графичка низа и се однесува на дводимензионалната низа од бои дефинирана со самиот стандард. Се користи за поврзување на монитори, телевизори, проектори и друга видео-опрема. **HDMI (анг. High Definition Multimedia Interface)** е приклучок за видео и звучни уреди и тој со својот квалитет на пренос го потисна VGA-приклучокот. Кратенката HDMI значи поврзување на мултимедијални уреди со висока резолуција.

USB-уреди → USB-приклучоците се најчесто користени компјутерски приклучоци за голем број периферни уреди: глувче, тастатура, плеери, надворешни хард-дискови, камери и фотоапарати, развојни платформи како што се Arduino и Raspberry Pi. Најчесто функционираат на принципот вклучи и работи

(анг.plug and play), што значи софтверот за овие уреди се инсталира автоматски.

2.1.4. Уред за напојување

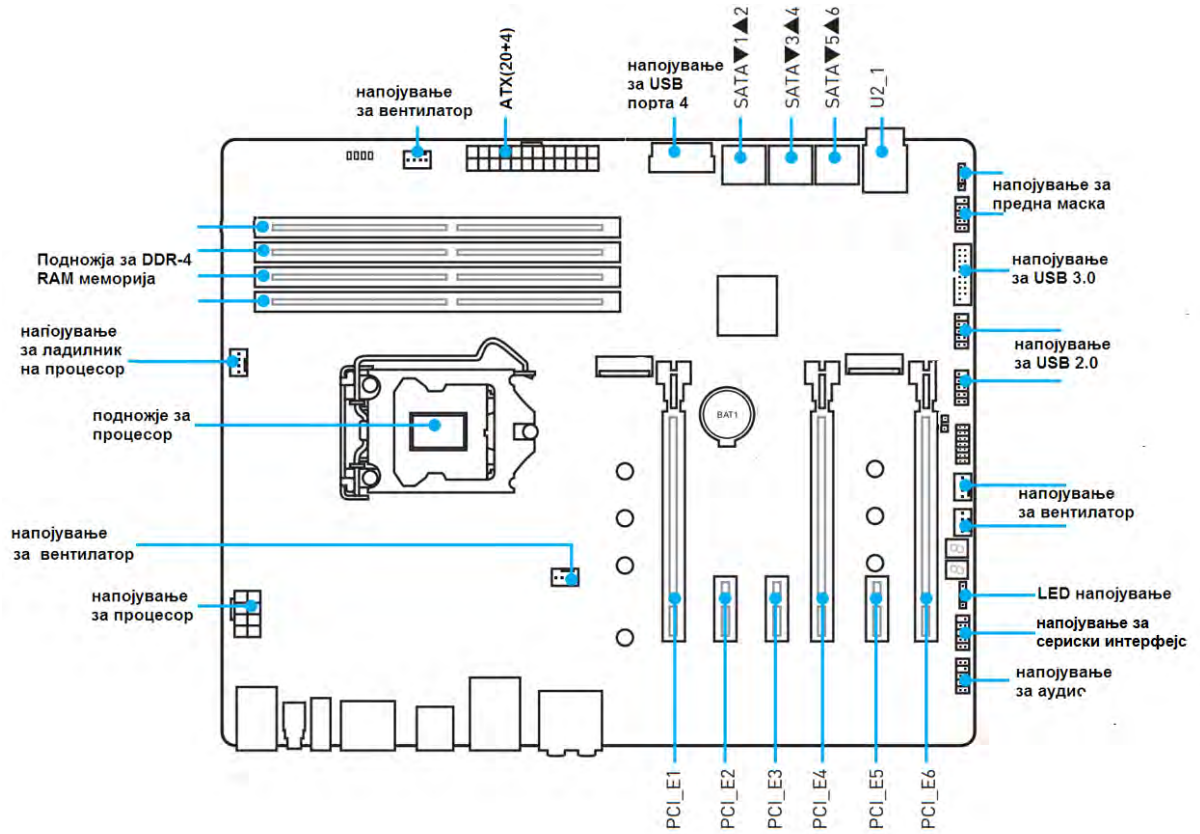
За да работат хардверските компоненти во составот на еден компјутерски систем, потребно им е напојување. Уредот за напојување користи мрежен напон од 220V и негова задача е да обезбеди **од мрежниот напон** повеќе различни **еднонасочни напони за напојување** на хардверските компоненти. Во својот состав, овој уред содржи мрежен трансформатор, насочувач и стабилизатор.



Слика 2.9. Надворешен изглед на уред и кабли за напојување

На слика 2.9. е прикажан надворешниот изглед на уредот за напојување и конекторите за поврзување на каблите за напојување. Позначајните кабли за напојување се прикажани на истата слика. Еден дел од хардверските компоненти се напојуваат директно од уредот, а друг дел индиректно, преку матичната плоча. За напојување на матичната плоча се користи кабелот **ATX (20+4)** и кабелот **ATX 12V(4+4)** за степенот за напојување на јадрото на микропроцесорот. Поради тоа, вториот кабел е познат и под името процесорски кабел за напојување. За напојување на процесорот се доволни само 4 пинови, но доколку микропроцесорот работи со фреквенција поголема од номиналната фреквенција, тогаш се поврзуваат сите 8 пинови. Осумпински кабел е и кабелот **PCI-E** (6+2), кој се користи за напојување на графичката картичка. Треба да се внимава, овој кабел да не се помеша со кабелот ATX 12 (4+4). SATA-каблите служат за напојување на хард-дискот и SSD-меморијата.

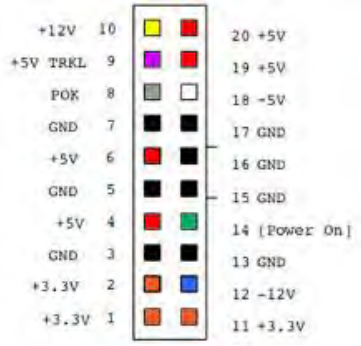
На слика 2.10. е прикажана шема на матична плоча и се означени конекторите и подножјата на каблите за напојување.



Слика 2.10. Шема на матична плоча со означени конектори за напојување

Подоцна, практично ќе се запознаеме со начинот на нивното поврзување. Од сликата гледаме дека матичната плоча може да има повеќе конектори за напојување на вентилатори за ладење. Преку матичната плоча се напојуваат и елементите поставени на предната маска од куќиштето, како што се LED-индикатори, USB-порти и копчето за вклучување и исклучување на десктопот. Да нагласиме дека подножјата за напојување на портите USB 3.0 и USB 2.0 се различни и обично подножјето за портата USB 3.0 е со сина боја. Матичните плочи прилично се разликуваат по распоредот на конектори и подножја и поради тоа **се препорачува користење упатство**, техничка документација која доаѓа заедно со матичната плоча.

Денешните компјутери се поотпорни на дефекти од порано. Но, доколку компјутерот не сака да се вклучи, најдобро е прво да се провери исправноста на уредот за напојување. Со помош на мултиметар можат да се измерат напоните на пиновите на излез од ATX 20+4 кабелот. **Тест-мерењето на излезните напони** и пин-дијаграмот на ATX20+4 кабелот е прикажано на слика 2.11. Краткоспојницата е поставена меѓу 15 и 16 пин бидејќи уредот за напојување нема да обезбеди излезни напони доколку нема оптоварување. Пред поставувањето на краткоспојницата, задолжително да се исклучи уредот за напојување од мрежниот напон како заштита од струен удар. Да нагласиме дека ваквото испитување може да биде опасно и не треба да се извршува без надзор на предметен наставник.



Слика 2.11.Проверка на напон на излезни пинови на кабел за напојување ATX (20+4)

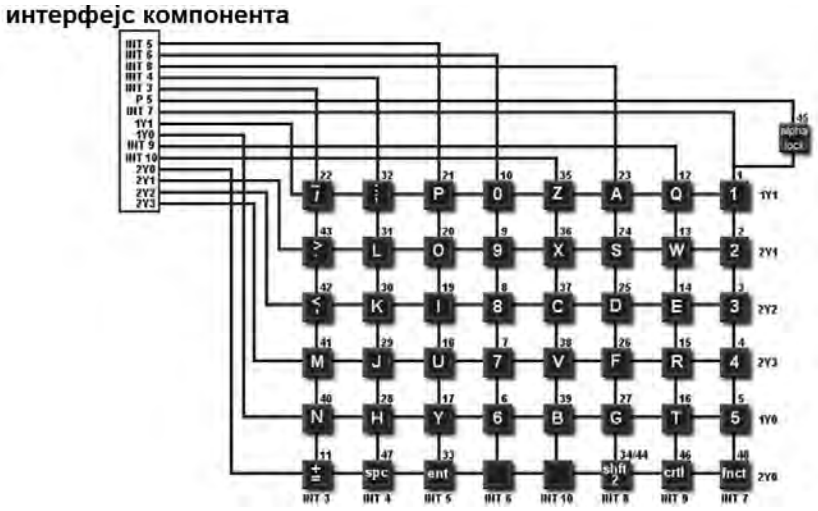
2.2. Периферни уреди и нивни карактеристики

Периферните уреди се познати под името влезно-излезни единици. Преку влезните единици човекот внесува податоци во компјутерот, а преку излезните единици тој добива информации од компјутерот. Всушност, може да се каже дека човекот се поврзува со компјутерот преку влезно-излезните податоци. На пример, звуците во природата се континуирани, аналогни сигнали. Откако тие ќе се претворат во електричен сигнал, потребно е да се дигитализираат за да бидат компјутерски обработени. Од континуираните сигнали се земаат примероци во точно определени интервали, земените примероци се заокружуваат на цели броеви и потоа добиените вредности се кодираат, односно претставуваат комбинации од нули и единици. Ние ќе се запознаеме со основните принципи на работа и карактеристиките на поважните периферни уреди.

Влезни уреди се: тастатура, глувче, скенер, магнетен читач, камера и други.

Тастатурата се користи за внесување текст, бројки, алфанумерички знаци, како и за контрола на операции. Класичната тастатура содржи од 101 до 104 копчиња и најчест стандард за распоредување на копчињата е QWERTY. За поврзување на тастатурата со компјутер се користат PS2 конектор или USB-порта. Постојат и безжични тастатури, односно Bluetooth тастатури. Самите копчиња можат да бидат механички и со мембрана. Иако механичката тастатура е постар тип, таа сè уште е актуелна поради сигурноста на механичките контакти. Со секое притискање на тастатурата се испраќа информација од еден бајт до микропроцесорот. Микропроцесорот ја гледа тастатурата како матрица, мрежа од редици и колони и во секоја спојна точка на матрицата има по еден прекинувач. Матрицата со прекинувачите е прикажана на слика 2.12. Секое копче има свои координати кои одговараат на бројот на редица и бројот на колона. Ако биде притиснато некое копче од тастатурата,тогаш се утврдуваат

координатите на тоа копче и врз основа на таа информација, кодерот во тастатурата генерира уникатен код кој потоа се испраќа до микропроцесорот.



Слика 2.12. Внатрешна структура на тастатура

Компјутерското глувче се користи за движење на покажувачот по екранот на мониторот. Со левиот клик се селектираат објекти, а со десниот клик се отвора менито за селектираниот објект.

Слика 2.13. Внатрешна структура на глувче

Според механизмот, компјутерските глувчиња се поделени на електромеханички, оптички и ласерски. На долната страна од оптичкото глувче се сместени два елементи, **LED-диода и CCD-сензор**, како што е прикажано на слика 2.13. CCD-сензорот претставува матрица од фотоосетливи елементи. Насоченият светлосен зрак се одбива од подлогата и ги побудува фотоосетливите елементи. Кога рефлектираниот зрак ќе падне врз фотоосетливиот елемент, тој генерира електричен напон. Микропроцесорот го идентификува побудениот фотоелектричен елемент според неговите координати (број на редица и број на колона). Потоа микропроцесорот го

поставува покажувачот на екранот во истата положба како и побудениот фотоелектричен елемент во матрицата на CCD-сензорот. Ласерските глувчиња имаат поголема прецизност, т.е понасочен и потенок светлосен зрак во однос на оптичките глувчиња.

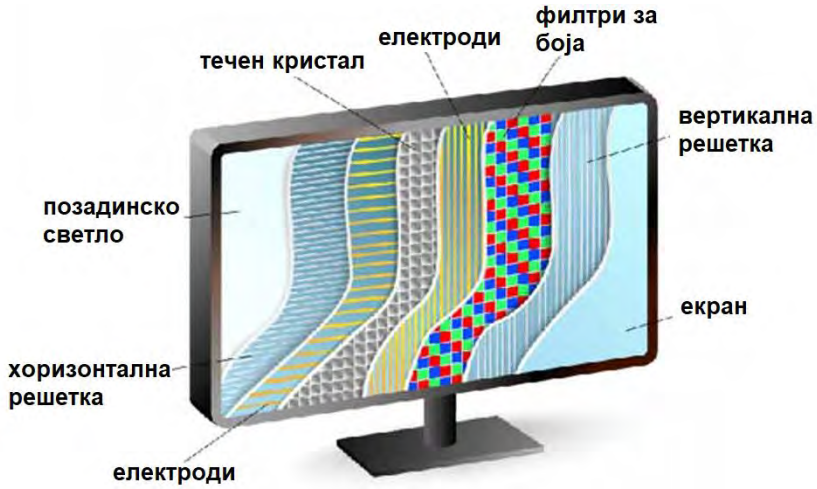
Оптичкиот читач или скенер е уред што ги претвора текстовите, сликите или фотографиите во печатена форма во електричен сигнал. Скенерот се состои од извор на светлина и оптички сензор кој ја детектира рефлектираната светлина. Доколку сакаме скенираниот текст да го обработуваме како текст, а не како слика потребни се специјални програми. Основни карактеристики на скенерот се резолуцијата и длабочината на бојата. Резолуцијата се мери во број на точки на еден инч и се движи во границите од 75dpi (анг. dots per inch) до 960dpi. Длабочината на бојата се мери според количеството на информација што скенерот ја собира за една точка од сликата. За црно-бели слики доволен е 8-битен скенер, а скенерите за слики во боја може да бидат 24 или 36 битни.

Магнетниот читач е познат под името читач на картички, како што се банкарските картички.

Графичкиот таблет е влезен уред кој ги претвора рачно нацртаните цртежи, графици, анимации во електрична форма, погодна за компјутерска обработка. Дебелината на линиите, сатурацијата на бојата, транспарентноста и другите својства на цртежот зависат од силата со која пенкалото притиска врз работната површина на графичкиот таблет.

Излезни единици се: монитор, печатач, слушалки или звучници.

Мониторот и графичката картичка го сочинуваат видео-системот на компјутерот. Мониторот ни овозможува визуелно да ги следиме активностите во внатрешноста на микропроцесорот.

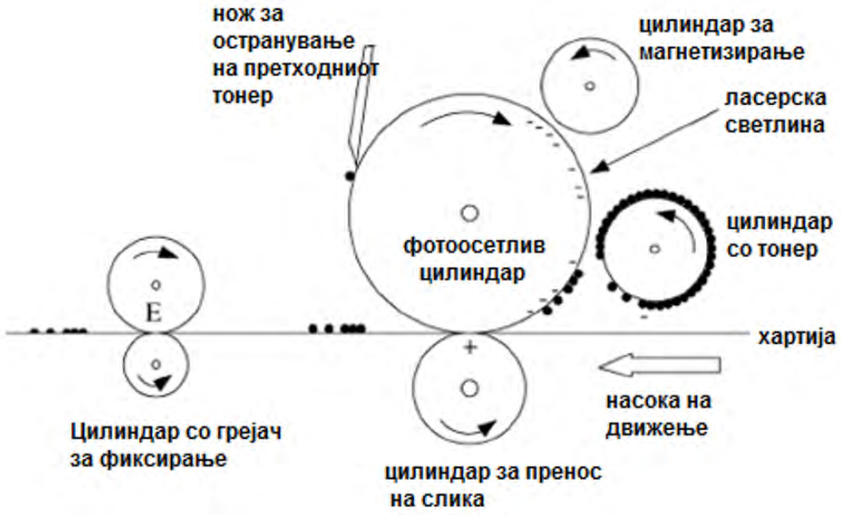


Слика 2.14. Внатрешна структура на LCD-екран

Пиксел е најмалата точка на екранот со точно определена боја. Исто така, пикселите се организирани во форма на матрица. Резолуција е бројот на пиксели во една хоризонтална или вертикална линија на екранот. На пример, Full HD-резолуција значи 1920x1200= 2304 000 пиксели на екранот. Брзината на

обновување (анг. Refresh Rate) се мери во херци и го дава бројот на слики во една секунда. Големината на екранот е дадена во број на инчи и стандардни големини се 15, 17, 19, 21 и 27 инчи. Секој пиксел е составен од три потпиксели, по еден потпиксел за трите основни бои: црвена-зелена-сина (RGB). Бидејќи потпикселите се на многу мали растојанија, доаѓа до мешање на боите. Во зависност од интензитетот на RGB боите, може да се добие која било боја од видливиот спектар. Најчесто користени екрани се LCD (анг. Liquid Cristal Display), што во превод значи **екран со течни кристали.** Молекулите на течните кристали го пропуштаат позадинското светло кога не се под влијание на електричен напон. Црвените, сините и зелените потпиксели се добиваат со филтрирање на светлината. Структурата на LCD-екранот е прикажана на слика 2.14. Поважни карактеристики на LCD-екраните се: природната резолуција, брзината на одзив (анг. Response Rate), аголот на гледање, осветленоста и контрастот.

Печатачите се излезни уреди кои информациите на излез од компјутерот ги пренесува на хартија или некој друг медиум, односно документите од електронска форма ги претвора во печатена форма.



Слика 2.15. Внатрешна структура на ласерски печатач

Постојат три вида печатачи: матрични, инк-џет и ласерски. Составни делови на ласерскиот печатач се: тонер-касета, ласерски модул, систем за пренос на слика и грејач за фиксирање. Најважен дел во ласерскиот печатач е **фотоосетливиот цилиндар** во внатрешноста на тонер-касетата. На почетокот, целата површина на цилиндарот е негативно наелектризирана. Фокусираната ласерска светлина го отстранува негативното количество електрицитет на оние делови од цилиндарот кои всушност треба да бидат отпечатени. Потоа, формираната електростатичка слика се претвора во видлива слика. Честичките на прашкастиот тонер се негативно наелектризирани и тие се одбиваат од негативно наелектризираните делови на фотоосетливиот цилиндар, а се привлекувани од позитивно наелектризираните делови (делови озрачени со ласерска светлина). На ваков начин се формира „позитив“ слика на

фотоосетливиот цилиндар. Потоа, тонерот од цилиндарот се пренесува на хартија и на крај, истиот се фиксира со помош на топлина. Процесот на формирање печатена слика е прикажан на слика 2.15. Резолуцијата е најважен параметар за добивање квалитетна печатена слика и кај ласерските принтери таа се движи од 600 до 2400 точки по инч (анг. dots per inch). Исто така, друга карактеристика е брзината на печатење која изнесува до 40 слики во една минута кај печатачите за канцелариско работење, а постојат и професионални печатачи со брзина од 12 до 60 страни во една минута.

Од влезно-излезните уреди ќе ги споменеме графичката, звучната и мрежната картичка.

Графичката или видео-картичката обработува слики и сè она што се гледа на екранот од мониторот е резултат на работата на оваа картичка. Самата графичка картичка има своја видео RAM-меморија, чиј капацитет во просек се движи околу 4GB до 8GB, иако на пазарот се нудат и графички картички со 24GB капацитет на RAM-меморија. Во неа се чуваат необработените слики добиени од процесорот, но и обработените слики кои се испраќаат кон мониторот.

Слика 2.16. Два модели на графички картички: MSI GeForce GT 710 и MSI GeForce GTX 1050 Ti

Од капацитетот на видео RAM-меморијата зависи резолуцијата на добиената слика. Просечните графички картички имаа просечна брзина на обработка околу 50 слики во секунда. За аналогни монитори се користи RAMDAC конвертор што ги претвора дигиталните сигнали во аналогни. При изборот на графичка картичка треба да се земат во предвид три параметри: капацитетот на видео RAM-меморијата, физичката должина и потрошувачката на електрична енергија. Должината на графичката може да биде од 120mm до 290mm. На слика 2.16. е прикажан надворешниот изглед на два модели графички картички: MSI GeForce GT 710 и MSI GeForce GTX 1050 Ti. [12]

Составни делови на **звучната картичка** се процесор за обработка на звук и CODEC (аналогно/дигитален и дигитално/аналоген) конвертор. Стандардни влезови се: Line in (за поврзување со музички систем), Mic in (микрофон) и CD in (за поврзување со вграден CD-плеер). Стандардни излези се: Speaker out, Line

out и S/PDIF (Sony/Philips Digital Interface). Последниов излез може да биде за коаксијален или оптички приклучок.

Мрежните картички служат за поврзување на компјутерот во локална (LAN-Local Area Network) или безжична мрежа (WAN-Wireless Area Network). Некои матични плочи имаат вградена мрежна картичка.

2.3. Избор на хардверски компоненти и калкулирање на трошоците за потрошеното време и финансиите

Компјутерската техника се развива со огромна брзина. Секојдневно се појавуваат нови типови на микропроцесори, графички картички, екрани, апликации кои го менуваат начинот на комуникација, начинот на примање информации, начинот на кој работиме, учиме, се забавуваме, функционираме и сл. Без соодветна компјутерска опрема не можеме да ги задоволиме нашите потреби и амбиции. Од друга страна, изборот на хардвер може да биде прилично ограничен поради малиот буџет и недоволните познавања на хардверските компоненти. Во првата тема, кога зборувавме за видови микрокомпјутери, истакнавме дека лаптоп со иста конфигурација, како и десктоп чини речиси двојно повеќе. Доколку можеме да се откажеме од можноста за преносливост, тогаш е многу подобро да се одлучиме за десктоп. Исто така, при изборот на компјутер е многу важна неговата намена. Многу познавачи на компјутерската технологија се двоумат дали е подобро да купат готов производ или самите да го конфигурираат и асемблираат својот десктоп компјутер. Двете можности имаат свои предности и недостатоци. **Купувањето на готов компјутер е полесно и побрзо**. Не трошиме време околу изборот на компоненти, не вложуваме труд за нивно составување и поврзување, со самото купување компјутерот е спремен за работа. Кога се работи за готов компјутер, се добива целосна корисничка поддршка, гаранција, сервис, а понекогаш и бесплатно инсталиран оперативен систем. Недостаток е **непостоењето на индивидуален пристап** при изборот на компјутер, односно самиот корисник нема можност сам да ги избере хардверските компоненти. Најлесно е да се купи најскапата конфигурација или да се нарача истата. Но, дали е тоа најпаметно решение во поглед на исплатливоста на инвестицијата. Од друга страна, готовите конфигурации, во просек, се со послаб квалитет и сигурност. Во светот постојат многу производители на хардверски компоненти, од кои некои се непознат или со сомнителен квалитет. Поголем дел од нивните производи завршуваат во готови конфигурации, а не како посебни производи за продажба. Самите компании, како што се Dell или HP наплаќаат за своите услуги за конфигурирање и составување на компјутерот, што секако мора да се одрази врз

цената на чинење. Доколку сами ги избереме хардверските компоненти, со сигурност добиваме поквалитетен компјутер, со подолг век на траење.

Конфигурација	**Готов производ**	**Сопствено составување**
Процесор	AMD Ryzen 3 3300X Quad-Core 3.8 GHz Socket AM4 65W 100-100000159BOX Desktop Processor	**7750 денари**
Матична плоча	MSI B450 TOMAHAWK MAX II AM4 AMD B450 SATA 6Gb/s ATX AMD Motherboard	**5980 денари**
RAM-меморија	G.SKILL Ripjaws V Series 16GB (2 x 8GB) 288-Pin DDR4 SDRAM DDR4 3200 (PC4 25600) Desktop Memory Model F4-3200C16D-16GVKB	**4316 денари**
Графичка картичка	GIGABYTE GeForce GT 1030 Low Profile D4 2G DirectX 12 GV-N1030D4-2GL Video Card	**6180 денари**
Куќиште	DIYPC Shadow-H3-ARGB Black Steel / Tempered Glass ATX Mid Tower Computer Case w/ 3 x 120mm Halo ARGB LED Fans Pre...	**3580 денари**
Напојување	Rosewill Glacier Series 500W Modular Gaming Power Supply with Silent Aero-Diversion Fan, 80 PLUS Bronze Certified,...	**4160 денари**
SSD-меморија	Intel 670p Series M.2 2280 512GB PCIe NVMe 3.0 x4 QLC Internal Solid State Drive (SSD) SSDPEKNU512GZX1	**3380 денари**
Хард-диск	Team Group CX1 2.5" 480GB SATA III 3D NAND Internal Solid State Drive (SSD) T253X5480G0C101	**2600 денари**
Ладилник за процесор	PURE ROCK SLIM 2	**1250 денари**
Вкупна цена	**43500 денари**	**39196 денари**
Забелешка	**Со готовиот производ имаме гаранција од 3 години**	**Во цената не е пресметано потрошеното време и вложениот труд**

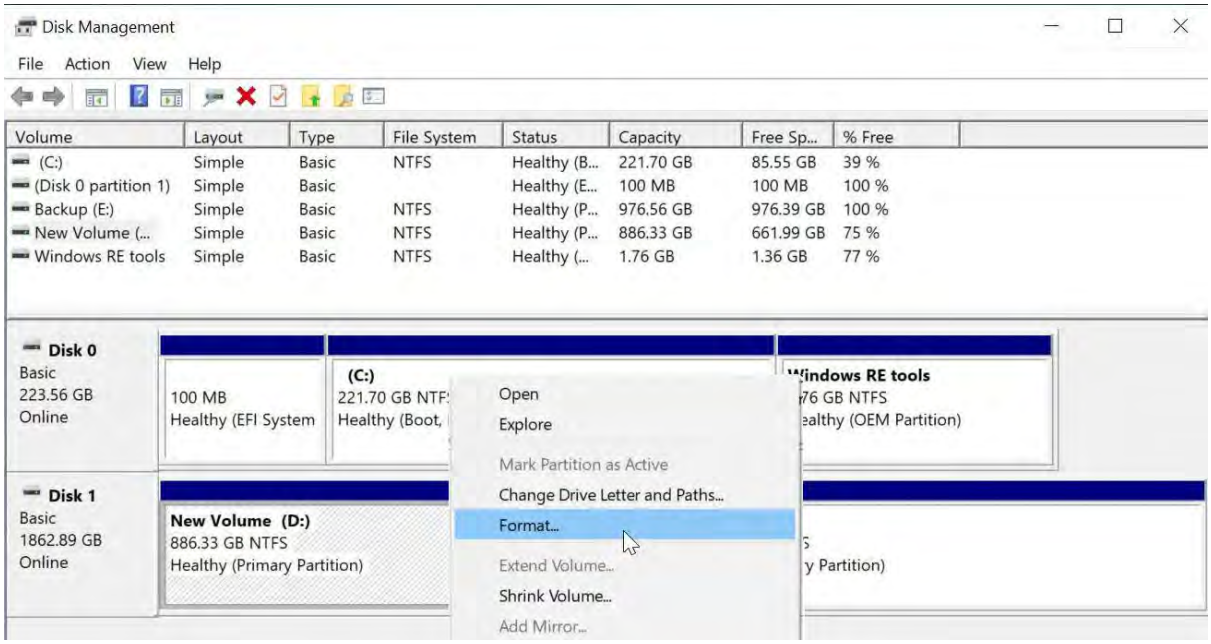
Табела 2.2 Калкулирање на трошоци за конфигурација

Најчесто поставувано прашање е дали е финансиски исплатливо самите да го составиме нашиот компјутер. Одговорот е: релативно поевтино. Цената на чинење на компјутер кој самите го составуваме е речиси иста со онаа на готов компјутер. Но, **гледано на подолг временски период, составувањето на компјутер е поисплатливо** бидејќи подобриот квалитет и приспособувањето на компјутерот кон нашите потреби значи подолг век на употреба или со други зборови, подолго време не мора да размислуваме за нова инвестиција. На

табела 2.2. е прикажана пресметка на трошоците доколку купиме готов компјутер или истиот да го составиме самите.

Друга предност при самостојното составување на компјутер е што корисникот многу полесно ќе може да го одржува и да го надоградува истиот. Со знаењето што ќе го стекне, навремено ќе реагира во случај на дефект и самиот ќе може да изврши замена на некој оштетен дел.

Доколку се одлучиме за сопствено составување на компјутер, при изборот на хардверски компоненти треба да се внимава за нивната **компатибилност.** На пример, доколку одбереме поскап процесор и матична плоча за сметка на поевтина графичка картичка, компјутерскиот систем ќе биде со мала брзина бидејќи графичката картичка бавно ќе ги обработува видео-податоците. Посебно е важно, матичната плоча, микропроцесорот и графичката картичка да бидат со **слични перформанси (брзина**). Најпродавани процесори се Intel и AMD. Intel процесорите се со поквалитетна изработка, поотпорни се на загревање, посигурни се и имаат поголем животен век. AMD процесорите имаат поголема извршна моќ и се попопуларни кај играчите на видео-игри и видео-дизајнерите. При изборот на графичка картичка треба да се одлучиме меѓу интегрирана или неинтегрирана картичка. Интегрираната графичка картичка е имплементирана на ист чип, заедно со микропроцесорот. AMD производителот за овој вид картичка го користи терминот APU (анг. Accelerated Processing Unit). Најпродавани графички картички се GeForce на компанијата NVIDIA и Radeon на AMD. Новите апликации се понапредни и бараат поголеми капацитети на работната меморија. На пример, за работа со документи, каква што е апликацијата Microsoft Office, доволно е RAM-от да има капацитет од 2GB, но доколку се работи за видео-обработка (апликацијата Adobe Premiere), тогаш потребни се 8GB работна меморија. Освен капацитетот, важна е и брзината на меморијата која се мери во мегахерци. Денес на пазарот се нудат три вида мемории DDR3, DDR4 и DDR5. Што се однесува до тајните мемории, SSD-мемориите се побрзи, поиздржливи, но и поскапи во однос на хард-дискот. Но, доколку сакаме да инвестираме во двата типа на мемории, тогаш можеме оперативниот систем да го инсталираме во SSD-меморијата за да заштедиме, а хард-дискот да ни служи за кориснички документи и апликации. За таа цел, доволна ќе ни биде SSD- меморија со капацитет од 120GB. Претходно, треба да провериме дали во куќиштето постојат две прегради за SSD и за хард-дискот. На почетокот, со матичната плоча ќе ја поврземе само SSD-меморијата и ќе го инсталираме оперативниот систем. После монтажата на SSD-меморијата треба да го поврземе хард-дискот. Пред употребата, истиот треба да се форматира, за што ќе ја употребиме програмата Disc Management која е дел од оперативниот систем Windows 10. Откако ќе ја отвориме програмата Disk Management, на екранот ќе се појават сите дискови кои се правилно поврзани. Со притискање на десен клик се избира опцијата Format. За да бидеме сигурни дека корисничките документи и апликации ќе се меморираат во хард-дискот, од алатката Settings избираме „Change where new contents is save“ и го избираме хард-дискот.



Слика 2.17. Форматирање на хард-диск

Последен чекор е да го нагодиме подигањето на оперативниот систем во BIOS-от. Во паѓачкото мени на прозорецот Boot order, SSD-меморијата треба да биде на прво место.

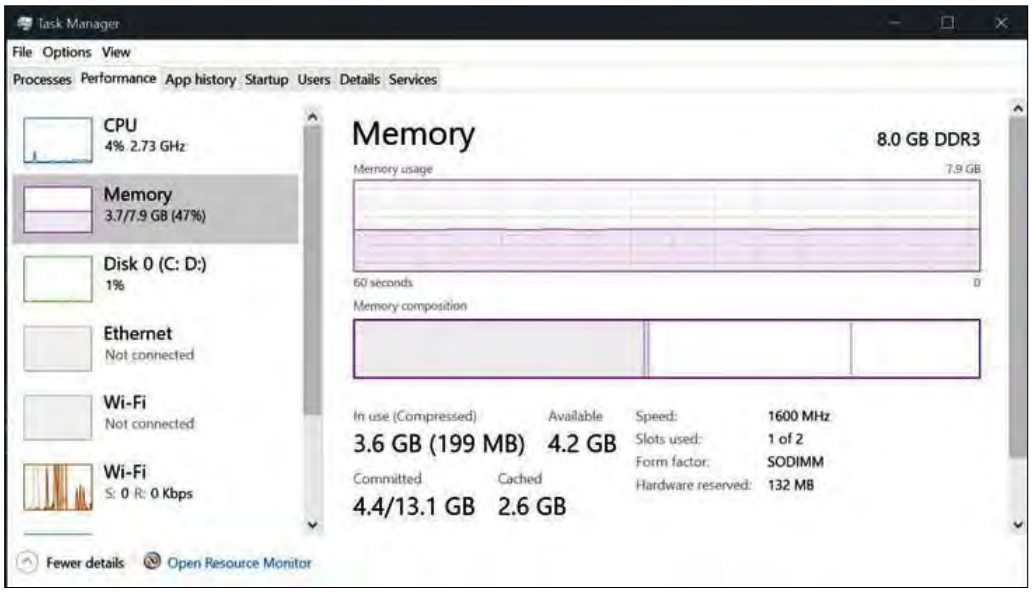
Од ова може да се заклучи дека изборот на хардверски компоненти е огромен и е многу важно корисникот добро да ги познава истите за да донесе вистинска одлука.

2.4. Надградба на конфигурација на компјутер

Лесната надградба е предност на десктопот во однос на лаптопот. Најважна хардверска компонента за надградба на компјутерот е матичната плоча, односно треба да знаеме кои се **слободни конектори** за поврзување на нови хардверски компоненти.

За да се зголеми брзината на работа на компјутерот, најлесно е да се **надгради RAM-меморијата**. Пред да донесеме одлука за надградба, треба да провериме колку меморија користи нашиот компјутер. Во тоа ќе ни помогне програмата **Performance Monitor**. Доколку искористеноста е поголема од 50%, тогаш е потребно надградба на RAM-меморијата. Потоа треба да провериме дали има **слободни DIMM- слотови** на кои ќе ги поставиме новите мемориски модули. Доколку не го познаваме видот на RAM-меморија во нашиот компјутер, можеме да дознаеме ако во Windows 10 ја отвориме алатката Task Manager и избереме Performance, како што е прикажано на слика 2.18. Доколку се работи за оперативен систем Windows 7, проверката на RAM-меморијата може да се направи со command prompt командата „wmic MEMORYCHIP get BankLabel,

DeviceLocator, MemoryType, TypeDetail, Capacity, Speed*".* Самата монтажа на RAM-меморијата е објаснета во практичниот дел на учебникот, наставна единица 2.2.Практична вежба за инсталација на составни делови на персонален компјутер.



Слика 2.18. Проверка на вид и капацитет на RAM-меморија

Што се однесува до трајната меморија веќе објаснивме дека најдобро решение е да инвестираме во двата видови мемории, со тоа што во SSD-меморијата ќе го чуваме оперативниот систем, а во хард-дискот ќе ги чуваме корисничките документи. Доколку вршиме надградба, пред форматирањето на хард-дискот, потребно е истиот да го поделиме на две партиции и на втората партиција да ги мемориграме сите кориснички документи за да не ги изгубиме истите. Во практичната вежба број 2.5. ќе се задржиме на инсталацијата на SATA SSD мемориски уред и M.2 SSD мемориски модул.

Покрај надградбата на работната и трајната меморија, конфигурацијата на компјутерот може да се прошири и со додавање нова графичка или мрежна картичка. Компјутерот може да работи со две картички, но после монтажата на истите, треба да се извршат одредени приспособувања.

На пример, брзината и стабилноста на интернет врската не зависат само од видот на избраниот интернет пакет, туку и од перформансите на мрежната картичка.

Слика 2.19. Мрежен адаптер со кабелски и антенски приклучок

Доколку сите мрежни уреди во нашиот дом имаат квалитетен интернет, со исклучок на нашиот десктоп со постара конфигурација, тогаш можеби треба да се замени неговата мрежна картичка. Повторно, изборот на мрежни картички е најразновиден. Тие се разликуваат пред сè во брзината, која може да изнесува 10Mbit/s, 100Mbit/s или 1000Mbit/s. Можат да бидат интегрирани, надворешни или внатрешни.Внатрешните мрежни картички се поврзуваат преку PCI или PCI Express конекторите. Надворешните картички се најчесто USB-мрежни адаптери и истите можат да бидат жичени (со кабелски RJ45 приклучок) или безжични (со антенски приклучок). Овие два типа на мрежни адаптери се прикажани на слика 2.19.

Можностите за конфигурирање и надградба на компјутерите се многу големи. Секојдневно излегуваат нови хардверски компоненти и за да се приспособиме кон новите технолошки предизвици, мора секогаш да бараме нови решенија за нашите персонални компјутери.

Заклучоци

Микропроцесорот има **две основни функции**: извршува програми и управува со другите уреди на матичната плоча. Составни делови на еден микропроцесор се: регистри, аритметичко-логичката единица и управувачката единица со декодер. Најважни карактеристики на микропроцесор се: работната фреквенција, ширината на податоците, проточниот концепт, комплексност на инструкциско множество, број на јадра, распоред на пинови и др.

RAM-меморијата го снабдува процесорот со информации, а пак таа добива информации од некоја трајна меморија, како на пример хард дискот**.** Сите програми се сместени на хард-дискот и тие се пренесуваат во RAM кога треба да бидат обработени.

Трајни мемории се хард-дискот, SSD-меморијата и надворешните мемории (компакт-дискови, USB-мемории, SD-картички и др.).

Изборот на матична плоча зависи од изборот на процесор бидејќи секој тип на процесор има уникатен распоред на пинови кои бараат и соодветно подножје.

Чипсетот е множество од специјални интегрирани кола кои го контролираат протокот на податоци.

DDR4, DDR3, DDR2 и DDR **RAM-модулите** се разликуваат по бројот и распоредот на пинови, напојувањето и работната фреквенција односно брзината.

PCI Express×16 и AGP се најдобри конектори за поврзување на графички картички. PCI слотот се користи за поврзување на звучни картички, мрежни картички. Конекторите SATA2 и SATA3 се за поврзување на хард-дискот и SSD-меморијата.

Пред монтажа на персоналниот компјутер, задолжително треба **да се исклучи напојувањето** и да се заштитиме од статички електрицитет.

При монтажа на процесорот и RAM-меморијата не смее да се притиска и да се допираат златните пинови.

Повеќето **кабли за напојување имаат уникатен дизајн**, па не може да се згреши при нивното поврзување на матичната плоча со другите уреди.

Тастатурата, CCD-сензорот и LCD-екранот се организирани во форма на **матрица** и секој елемент од матрицата си има свои координати (реден број на редица и колона). Микропроцесорот ги идентификува елементите според нивните координати.

Основни **карактеристики на графичката картичка** се капацитетот на видео RAM- меморијата и брзината на работа која се мери во број на обработени слики во една секунда.

Најважен **конектор за напојување е ATX 20+4** кој се користи за напојување на матичната плоча. Повеќето хардверски компоненти го користат напојувањето на матичната плоча за сопствено напојување и се поврзани преку специјални конектори за напојување. Излезните напони на ATX 20+4 конекторот можеме да ги провериме со помош на мултиметар.

Пред надградувањето на RAM-меморијата потребно е да се провери искористеноста, видот и капацитетот на меморијата и бројот на слободни DIMM-слотови.

При надградување на трајната меморија, **оперативниот систем треба да се инсталира во SSD-меморијата, а во хард-дискот да се чуваат корисничките документи и апликации**. При набавка на SSD-меморија треба да се внимава на начинот на поврзување со матичната плоча и распоредот на нејзините пинови.

Купувањето на готов компјутер е полесно и побрзо.Не трошиме време околу изборот на компоненти, не вложуваме труд за нивно составување и поврзување

и компјутерот е спремен за работа со самото купување. Кога се работи за готов компјутер, се добива целосна корисничка поддршка, гаранција, сервис, а понекогаш и бесплатно инсталиран оперативен систем.

Сопственото составување на компјутер значи добивање поквалитетен производ кој целосно одговара на нашите потреби, со подолг век на употреба, полесно одржување и надоградување.

За време на **start-up процесот**, компјутерот испушта звучни и светлосни сигнали кои можат да ни помогнат при дијагностика на грешка во монтажата на компјутерот.

Заради лична заштита и заштита на елементите од оштетување, пред почетокот на секоја монтажа треба да се исклучи изворот за напојување, без оглед дали се работи за батерија или компјутер.

Прашања и задачи

1. Кои се трите составни делови на микропроцесорот и која е нивната функција?
2. Набroj ги најважните карактеристики на микропроцесорот!
3. Кои се двете најважни карактеристики на меморииte?
4. RAM е работна, привремена меморија со случаен пристап. Објасни!
5. Кои се предности, а кои се недостатоци на хард-диск меморијата во однос на SDD-меморијата?
6. Што претставува и за што служи чипсетот на матичната плоча на микрокомпјутерот?
7. Како може да се препознае видот на DDR RAM-модул?
8. Кои слотови од матичната плоча се користат за поврзување на графичката картичка?
9. Што извршува BIOS-от после секое вклучување на компјутерот?
10. За што се користат SATA2 и SATA3 конекторите на матичната плоча?

11. Кој е најдобар приклучок за поврзување на видео-уредите?

12. Во четири чекори, објасни ја инсталацијата на процесорот во неговото подножје.

13. На што треба да се внимава при инсталацијата на RAM меморијата во слотовите од матичната плоча?

14. За што служат F (Front), SYS_FAN и PWR-FAN приклучоците на матичната плоча?

15. Именувај ги каблите за напојување на матичната плоча и процесорот? По колку пинови има секој кабел?

16. Објасни ја постапката на монтажа на SSD-меморија и хард-диск меморија, со редоследот на нивно поврзување, инсталација на оперативен систем на SSD-меморијата и приспособување на хард-дискот за чување на кориснички податоци!

17. Одбери една готова конфигурација на компјутер, по твоја желба, и пресметај ги трошоците во случајна сопствено составување.

18. Демонстрирај како комуницира микропроцесорот со тастатурата!

19. За што служи CCD-сензорот во состав на компјутерското глувче?

20. Објасни ја постапката на проверка на исправноста на уредот за напојување на компјутерот!

21. Провери ја искористеноста, видот и капацитетот на RAM-меморијата во твојот компјутер!

22. Провери го видот на графичка картичка во твојот компјутер и пребарај на интернет за неговите карактеристики!

23. Какви звучни сигнали испушта компјутерот во случај на откажување на некоја хардверска компонента? Заедно со наставникот проверете што ќе се случи доколку го извадите RAM-от од неговиот слот на матичната плоча.

24. Одбери една матична плоча, по твој избор, и пребарај на Интернет информација за распоредот на конекторите и слотовите!

3. Инсталација на оперативен систем и други стандардни програми на персонален компјутерски систем

3.1. Видови оперативни системи

Оперативниот систем е софтвер потребен за извршување на апликативни програми и за координирање на активностите на компјутерскиот систем. Тој обезбедува околина во која другите програми можат да извршуваат корисна работа. Тој е програма како и секоја друга програма, но многу покомплексна, помоќна, поголема. Поради тоа тој мора да се креира дел по дел, со добро дефинирана функција, влез и излез за секој дел. Оперативните системи може да се проучуваат од две гледни точки: од онаа на корисникот и онаа на системот [2].

Од кориснички поглед оперативниот систем е дизајниран со цел да се минимизира работата на корисникот при искористувањето на хардверските ресурси односно да се обезбеди удобна употреба на микрокомпјутерските системи и подобрување на нивните перформанси. Компјутерот прима инструкции од корисникот преку тастатурата и дава пораки преку мониторот. Во зависност од обликот на овие наредби и методот на нивно внесување разликуваме два вида кориснички интерфејс, преку командна линија (анг. interface command line) или графички интерфејс (анг. graphics user interface). Кај интерфејсот со командна линија се употребуваат текстуални наредби, а кај графичкиот интерфејс се користи систем од прозорци, менија и листи за селекција. За пристап до интерфејсот со командна линија, во Windows оперативниот систем, во полето за пребарување (анг. Search) треба да се напише cmd (анг. Command Prompt).

Од системски поглед, оперативниот систем е контролно-управувачки софтвер и е тесно поврзан со хардверот. Оперативниот систем одлучува која програма ќе се изврши и кога, колку меморија ќе им се додели на програмите, се

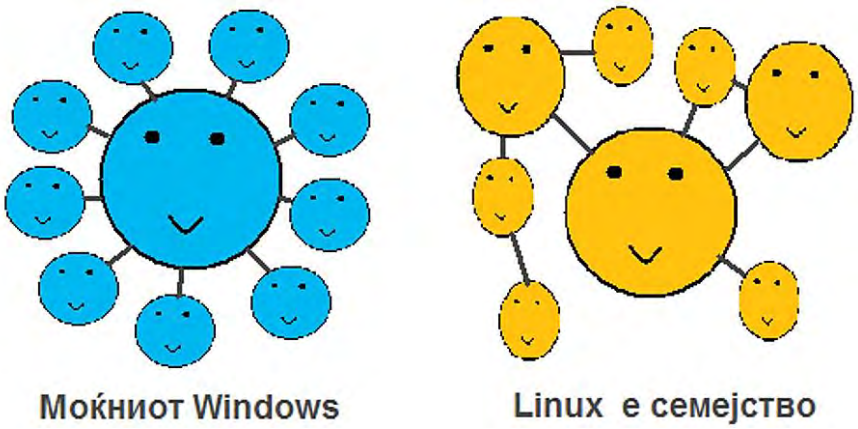
грижи за правилно извршување на командите дадени од корисникот и слично. Накратко ќе се запознаеме со функциите на оперативниот систем.

- Извршување на програмите. Оперативниот систем е задолжен за преносот на инструкциите од RAM меморијата во микропроцесорот, нивно декодирање и извршување.
- Програми за распределба на хардверски ресурси. За ефикасна работа на микрокомпјутерскиот систем оперативниот систем мора да одлучи како да ги додели хардверските ресурси (микропроцесор, мемориски простор, влезно-излезни единици) на специфичните корисници и програми. На пример, микропроцесорот може да извршува повеќе задачи истовремено така што постојано ќе се префрла од една на друга задача, но префрлувањето се извршува толку често што корисниците и нивните програми се во постојано заемно дејство.
- Програми за управување со меморијата. Оперативниот систем одлучува кои програми и податоци ќе се пренесат од секундарната во примарната меморија, доделува и ослободува мемориски простор по потреба, ги следи тековно употребуваните делови на меморијата, служи за форматирање, чистење и одржување на хард-дискот, се грижи за работата на кеш меморијата.
- Програми за управување со влезно-излезни операции и уреди. За контрола на преносот на податоци од или кон периферните уреди и нивно привремено складирање се користат специјални интегрирани кола наречени контролери за уреди. За да започне преносот на податоци потребно е во регистрите на контролерот да се запишат бинарни кодови со точно определена вредност. Ова го извршува драјверот кој всушност е „двигател на уредот”. Драјверите се помошни програми кои посредуваат меѓу оперативниот систем и контролерите.
- Програми за заштита и безбедност. Заштитата е механизам за контрола на пристап на корисниците до хардверските ресурси и авторизирана употреба на истите. Одбраната на системот од внатрешни и надворешни напади е работа на безбедност. Во оваа категорија спаѓаат антивирусните програми.
- Програми за комуникација. Овие програми овозможуваат пренос на датотеки на далечина од еден на друг компјутер, пребарување на веб-страници, испраќање на електронска пошта и др.
- Програми за управување со датотеки. Овие програми креираат, селектираат, бришат, копираат, преименуваат датотеки и директориуми.
- Програми за работа со статусни информации. Статусни информации се време, датум, количество на достапна меморија, број на корисници и слично.
- Поддршка за програмски јазици. Оперативниот систем му обезбедува на корисникот едитори, компајлери, асемблери, дебагери и толкувачи за

најчесто користените програмски јазици. Овие алатки се неопходни за создавање на апликативен софтвер.

Оперативниот систем Windows е првиот оперативен систем со графички интерфејс и е производ на компанијата Microsoft. Тој е еден од најкористените и најпопуларните оперативни системи. Денес дури 80% од корисниците на персонални компјутери го користат оперативниот систем Windows. Во почетокот, Windows бил само надградба на тогаш популарниот оперативен систем DOS. Уште на самиот почеток постоела софтверска поддршка за работа со глувче, икони, мени и едноставен премин од една во друга апликација. **Предности на оперативниот систем Windows** се: едноставност во работењето (user friendly), најголема поддршка за поврзување со периферни уреди, достапност до апликативен софтвер, автоматска инсталација (анг. plug and play), најдобри перформанси за видео-игри, компатибилност со најголем број на веб-страници**. Недостатоци** се: потреба од скапи компјутерски конфигурации, затворен код, слаба сигурност во поглед на интернет-напади и административна заштита, чувствителност на компјутерски вируси, потребна е лиценца, слаба експертска поддршка и честопати за да се зголеми брзината на работата, потребна е преинсталација на оперативниот систем.

Linux е понов оперативен систем. Идејата ја дал финскиот студент Линус Торвалдс, кој во 1991 година објавува труд за создавање јадро на нов бесплатен, стабилен оперативен систем со отворен код, кој ќе може да се применува на различни платформи. **Отворен или слободен код** значи сите корисници да имаат пристап до изворниот код и да можат да го менуваат во зависност од своите потреби. Отворениот код е посигурен развоен систем. При самата анализа, програмерите ги откриваат грешките и вирусите, ги поправаат и своите решенија ги споделуваат со заедницата на Linux. Самото јадро на оперативниот систем има заштита од неколку нивоа. Пред да се направи извршна верзија на која било програма, таа се проверува. Исто така, веб-пребарувачот е независен од оперативниот систем. На слика 3.1. симболично е прикажана организацијата на оперативниот систем Linux, споредена со организацијата на оперативниот систем Windows.



Слика 3.1. Два различни, но успешни концепти

Архитектурата на оперативниот систем Linux е како разгрането дрво, секоја гранка претставува еден член од заедницата и секој може да придонесе во создавањето на побрз, посилен и посигурен систем. Можеби Windows е господар во светот на оперативните системи, но револуцијата на Linux дефинитивно го промени начинот на генерирање на оперативните системи. На почетокот, оперативниот систем Linux бил наменет за понапредни корисници, но со појавата на ефикасниот интерфејс GUI (анг. Graphical User Interface) Linux станува сè повеќе популарен. Денес Linux се користи како оперативен систем за сервери на големи компании, како што се Google, Facebook, Twitter. Многу компании го земаат јадрото на Linux како основа, го развиваат и создаваат нови лиценцирани оперативни системи. Таков е оперативниот систем Android на компанијата Google и оперативниот систем **Raspbian,** креиран од група ентузијасти, љубители на Raspberry Pi микрокомпјутерите. Подоцна ќе се запознаеме со неговата инсталација и начин на користење.

Денес 74,3% од корисниците на мобилни телефони користат **Android** оперативен систем. **Предности на оперативниот систем Android** се: огромен број бесплатни апликации (Google Play Store е апликација вградена во самиот оперативен систем), компатибилност со уреди од различни производители (Samsung, Huawei, Motorola), отворен код, располага со едноставни развојни средини за создавање нови апликации, меморискиот простор лесно се проширува, можност за споделување на интернет, едноставна комуникација со многу уреди, слобода при конфигурацијата и избор на апликации**. Недостатоци** на овој оперативен систем се: апликациите се потешки за изработка поради различните димензии на екраните, помала брзина поради заднинските апликации, помала сигурност и осетливост на вируси.

3.2. Софтверски алатки за конфигурација на Windows 10 оперативен систем

По инсталацијата на оперативниот систем, но и подоцна во текот на работата, потребно е да се **конфигурира истиот според личните барања и потреби**. Софтверските алатки за конфигурација овозможуваат: зголемување на брзината на работа, менаџирање со датотеки, продолжување на животниот век на компонентите, добивање информации за состојбата на хардверските и софтверските компоненти и инсталираните драјвери, организирање на податоците сочувани во хард-дискот или SSD меморијата, управување со задински апликации, поддршка за обновување на избришани документи или нивни стари содржини, поголема сигурност и заштита итн. Самиот Windows оперативен систем содржи голем број системски софтверски алатки за конфигурација, но одредени параметри можат да се постават и со употреба на BIOS програмата.

Да се потсетиме **BIOS-от** е специјален софтвер кој овозможува комуникација меѓу хардверските компоненти и оперативниот систем. До него пристапуваме со притискање на едно од копчињата F1, F2, F10, F12 или Del после вклучувањето на напојувањето. После отворањето на BIOS-от се појавува основното мени за поставување (анг. Setup) во горниот дел од екранот. BIOS-от не поддржува работа со глувче и за навигација се користат копчињата од тастатурата: горе, долу, лево, десно. Менито за поставување е составено од следните подменија: главно, напредно, безбедност и подмени за инсталација на оперативниот систем и вклучување на напојувањето. Накратко ќе се запознаеме со секое од подменијата. Да нагласиме дека менито за поставување е различно кај различни компјутери и зависи од видот на матична плоча.

Главно подмени (анг. Main) ⟶ Ова подмени дава информација за BIOS верзијата, видот на процесор и неговата работна фреквенција, капацитетот на меморијата. Тоа содржи алатки за менување на системското време и датум.

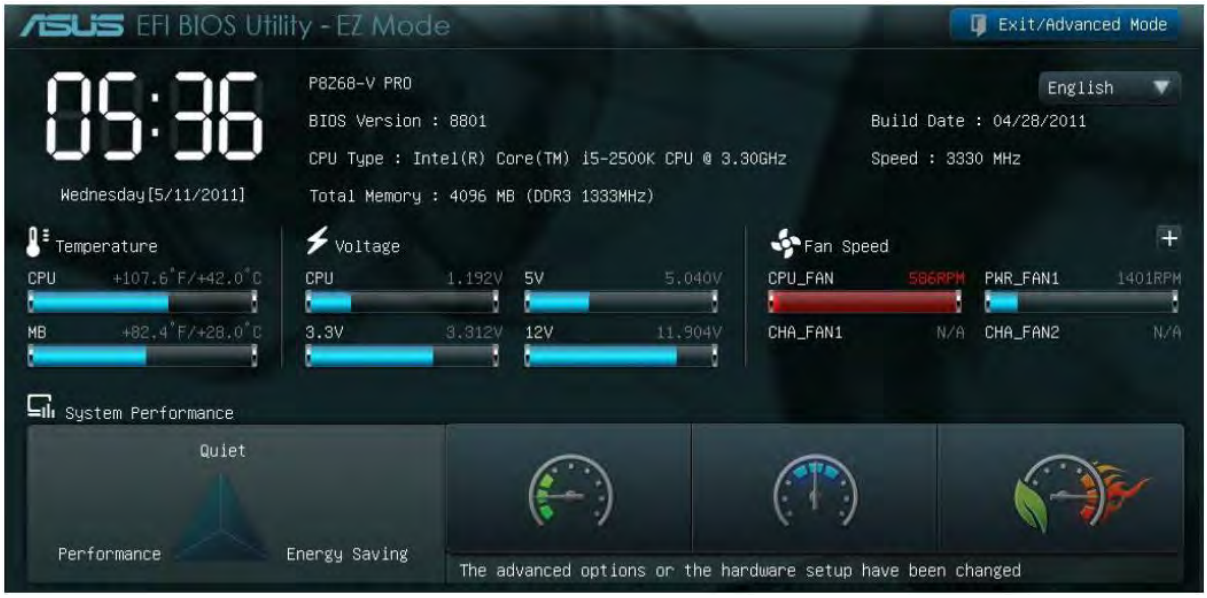
Напредно подмени (анг. Advanced) ⟶ Со ова подмени се врши поставување на мемориските уреди (хард-диск или SSD уред), приклучоците за графичката, звучната или мрежната картичка, USB уредите и другите уреди кои се директно поврзани на матичната плоча. Треба да бидеме многу внимателни и подготвени бидејќи неправилното поставување на одредени параметри во BIOS-от може да предизвика неработење на целиот систем.

Безбедност (анг. Security) ⟶ BIOS-от содржи две лозинки, супервизорска и корисничка. Супервизорската лозинка се нарекува лозинка за поставување бидејќи служи за пристап до самиот BIOS. Корисничката лозинка се нарекува системска лозинка и служи за пристап до оперативниот систем. Подменито за безбедност содржи алатки за промена или отстранување на лозинката за поставување. Во случај да се заборави лозинката тогаш сите лозинки можат да се избришат со употреба на посебна краткоспојница на самата матична плоча.

Подмени за поставување на оперативен систем (анг. Boot) ⟶ Да се потсетиме ова подмени се користи за подредување и избор на уред за инсталација на Windows оперативен систем

Излез (анг. Exit) ⟶ Ова подмени служи за сочувување на направените измени и излез од BIOS програмата.

Во последните пет години, во новите компјутери BIOS-от е заменет со нова програма позната под кратенката UEFI (анг. Unified Extensible Firmware Interface) што значи унифициран проширен имплементиран софтвер за интерфејс. Графиката е многу подобра и истата вклучува следење на перформансите на компјутерот: температура, брзина на вентилаторите за ладење, работна фреквенција на процесор итн. (слика 3.2.)



Слика 3.2. Графика на UIEF програмата

Најголема предност на UEFI програмата е тоа што нема ограничување во капацитетот на хард-дискот или SSD меморијата како што е тоа случај со BIOS-от кој поддржува мемориски капацитет од максимум 2TB.

Најпознати системски алатки за конфигурација на Windows оперативните системи се Control Panel и поновата апликација Settings. Подолу се набројани и објаснети категориите во апликацијата Settings.

Систем (анг. System) ⟶ Со алатките на оваа категорија може да се менува резолуцијата и осветленоста на екранот, да се ослободува простор за складирање податоци, да се избира режим на напојување итн.

Уреди (анг. Devices) ⟶ Оваа категорија дава листа на уреди со коишто може да комуницира компјутерот: Bluetooth-уреди, безжични екрани, екрани на допир, графички таблети, печатачи и слично.

Телефон (анг. Phone) ⟶ Овозможува пристап до содржината на нашиот Android или iPhone мобилен телефон.

Мрежа и интернет (анг. Network & Internet) → Може да се избере Wi-Fi или Ethernet мрежен адаптер, да се вклучи или да се исклучи авионскиот режим на работа, да се додаде нова виртуелна приватна мрежа (VPN) итн.

Персонализација (анг. Personalization) → За секој профил може да се одбере заднина, да ги промени боите на темата, да ја смени сликата за заштита на екранот (screen saver), да го измени изгледот на стартното менито или работната лента.

Апликации (анг. Apps) → Оваа категорија ги контролира програмите: бришење, менување на стартирачки (анг. start-up) програми, замена на стандардни програми, карактеристики на екран за видеоигри (анг. video mode).

Корисничка сметка (анг. Account) → Само администраторот може да креира кориснички профили. Kaj Windows 10, можностите за семејна заштита и набљудување се многу поголеми.

Време и Јазик (анг. Time & Language → Се нагодуваат времето, датумот, регионот, јазикот и опциите за говор.

Лесен пристап (анг. Ease of access) → Оваа категорија нуди можности за различен пристап до екранот, тастатурата, глувчето, оратор за аудио објаснување на елементите на екранот.

Личен виртуелен помошник (анг. Cortana) → Ова е нова категорија и претставува личен виртуелен помошник за пребарување на интернет и искористување на компонентите.

Приватност, ажурирање и безбедност (анг. Privacy, Update & Security) → Се користи за надградба на оперативниот систем, креирање резервна копија на личните документи (анг. backup), скенирање и откривање на вируси и потенцијално штетен софтвер, заштита од недозволен пристап и злоупотреба преку интернет, семејни опции. На пример, во категоријата Privacy → Camera може да се изберат програмите што ќе имаат пристап до вградената камера на лаптопот.

3.3. Корисни програми

3.3.1. Антивирусни програми

Компјутерските вируси се пренесуваат од еден на друг компјутер и имаат способност да се самокопираат, да ги менуваат и да ги уништуваат оперативните и корисничките датотеки. Овие датотеки се домаќини за компјутерските вируси и тие не можат да опстанат надвор од нив. Кога корисникот ќе отвори или стартува некоја од тие датотеки, вирусниот код почнува да се извршува. Компјутерските вируси предизвикуваат успорување на компјутерот, преполнување и оштетување на хард-дискот, бришење, преименување и разместување на датотеки, препраќање на сите адреси од адресарот на електронската пошта, злоупотреба на лозинки, пристап до самиот компјутер преку интернет итн. Вирусите се пренесуваат преку: меморискиот стик, CD дискови од сомнителни извори, компјутерска мрежа, интернет, преку електронска пошта. Постојат неколку **начини за заштита од компјутерски вируси.** Не треба да се отвораат сомнителни електронски пораки и реклами, редовна надградба (анг. update) на оперативниот систем, бришење на историјата и кеш-меморијата на интернет- пребарувачот, скенирање на екстерните меморији пред нивната употреба, скенирање на хард-дискот после инсталацијата на нова програма. За скенирање на мемориите е потребна претходна инсталација на антивирусна програма. Антивирусите се софтверски пакети кои се состојат од компјутерски **програми што се обидуваат да ги пронајдат и да ги отстранат компјутерските вируси** и другите штетни програми. Антивирусните програми содржат бази на вируси и постојано се надополнуваат со откривањето на нови вируси. Антивирусната програма Windows Defender, како дел од самиот оперативен систем, не е доволна за да го заштити компјутерот од вируси и интернет-закани.

Слика 3.3. Антивирусни програми и нивните логоа

Денес, изборот на антивирусни програми е многу голем. Некои од нив се бесплатни, но доколу сакаме премиум заштита, потребно е програмата да се

плати. Norton е светски познат провајдер на една од најсигурните антивирусни програми. Со соодветен годишен надоместок и претходно креирање на своја корисничка сметка, програмата може да се преземе од нивната официјална веб-страна (https://us.norton.com/antivirus) и истата да се инсталира.

Бесплатни антивирусни програми, со добри резултати при тестирањето има повеќе и истите се дадедени во табела 3.1., заедно со линкот за преземање на истите.

Антивирусна програма	Линк за преземање
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

Табела 3.1. Антивирусни програми

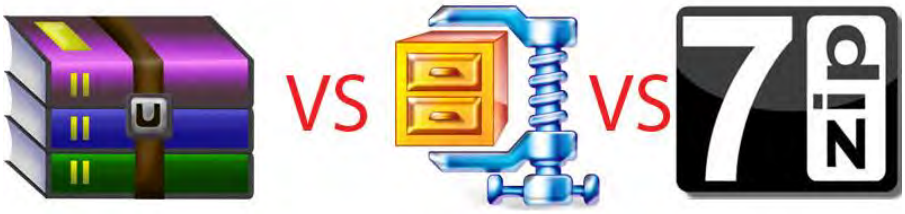
Освен заштита и детекција на вируси, овие програми нудат и други услуги како што се: VPN (анг. Virtual Private Network) пристап, родителски надзор, менаџер на лозинки, заштита на веб-камера, заштита на индентитет и лични податоци, складирање на податоци во облак и друго. На пример, со VPN пристапот стекнуваме анонимност, нашата IP-адреса се заменува со адресата на серверот, се прикрива нашата локација, податоците за пренос се кодираат и не е можен пристап до истите. Некои од овие функции може да ја забават работата на компјутерот, така што сите функции не мора да бидат инсталирани. Карактеристика на новите антивирусни програми е што можат да процесираат од облак и на таков начин не се оптоваруваат хардверските компоненти на нашиот компјутер.

3.3.2. Програми за компресија на датотеки

Компресија на датотеки е постапка за намалување на физичкиот простор односно меморискиот капацитет, потребен за нивно зачувување. Според даден алгоритам се намалува бројот на битови, кои служат за бинарно прикажување на податоците и на таков начин се намалува големината на датотеките, изразена во број на бајти. На пример, во текстуална датотека, доколку одреден збор се повторува неколку пати последователно тогаш истиот може да се пренесе само еднаш, но заедно со бројот на повторувања. Со компресијата се постигнува **заштеда на мемориски простор и заштеда на време** кое е потребно за пренесување на датотеките. Експанзија или декомпресија е инверзна операција која има за цел да ја врати оригиналната содржина на датотеките. Доколку се работи за компресија без губење на податоци тогаш декомпресијата дава податоци идентични на оригиналот, а во случај на компресија со губење на

податоци тогаш не се добиваат оригинални податоци. Во зависност од употребениот алгоритам, компресираните датотеки можат да бидат со различен формат, за што имаат различна наставка: .zip, .rar, .7zip, zipx, arc и други.

Најпознати програми за компресија за Windows оперативниот систем се: WINZIP, WINRAR, 7-Zip, PeaZip, BandZip и други. WINZIP и WINRAR не се бесплатни, а 7-Zip, PeaZip и BandZip се бесплатни за преземање.



Слика 3.4. Програми за компресија и нивните симболи

Тие се разликуваат по степенот на компресија, големината и видот на датотеките, брзината на компресија, поддршката за различни видови оперативни системи. Сите програми за компресија се поддржуваат меѓу себе. На пример, со програмата WINZIP можеме да декомпресираме RAR документ и обратно, но повторно сè зависи од форматот на датотеката и од верзијата на програмата. Секоја програма си има своја специфичност. Програмата WINZIP ја компресира секоја датотека посебно, без мешање на нивните податоци и може да се користат различни алгоритми според видот на датотеката. Програмата WINRAR ги компресира сите селектирани датотеки одеднаш, тоа трае подолго време, но се добива поголем степен на компресија. Програмите WINZIP, WINRAR и BandZip се компатибилни само со Windows оперативниот систем, а 7-Zip и PeaZip и со Linux оперативниот систем.

Во табела 3.2. се дадени резултатите од извршеното тестирање на пет програми за компресија. Како влезни податоци употребени се 42 датотеки со различна содржина, сместени во четири папки, со вкупен капацитет од 303MB. Најважен параметар е степенот на компресија изразен во проценти. Тој се добива според равенката:

$$\frac{\text{големина на оригинална датотека}}{\text{големина на компресирана датотека}} \cdot 100$$

Големината на степенот на компресија зависи не само видот на програмата туку и видот на формат. Да нагласиме дека степенот на компресија зависи и од видот на датотеката: текстуална, графичка, музичка или видео. Исто така, од табелата може да се заклучи дека брзината на компресија и декомпресија во многу зависи од видот на трајна меморија (SSD или хард-диск) од каде ја повикуваме оригиналната датотека и каде ја сочувуваме компресираната датотека. Без оглед за кој тип на компресија ќе се одлучиме, важна е нејзината примена бидејќи без компресија некои датотеки едноставно

не можат да се пренесат или да се архивираат како заштитни (анг. back up) датотеки.

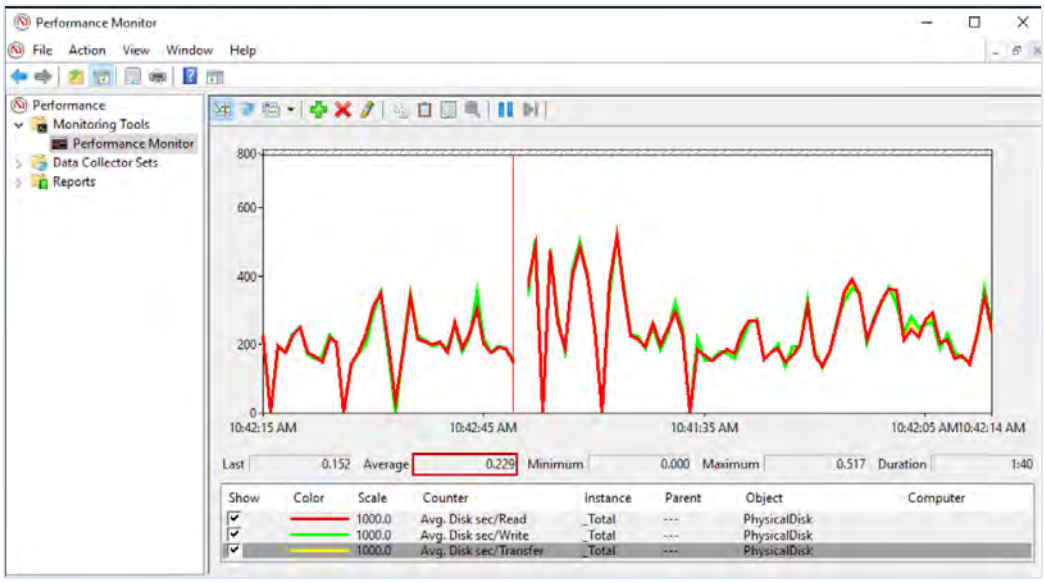
Програма и формат на датотека	Брзина на компресија за SSD (сек.)	Брзина на компресија за HHD (сек.)	Големина на компресирана датотека (MB)	Степен на компресија	Брзина на декомпресија за SSD (сек.)	Брзина на декомпресија за HDD (сек.)
PeaZip, ZIP	12,1	16,0	97,70	32,24%	1,6	13,4
PeaZip, 7Z	39,6	42,2	73,60	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

Табела 3.2. Споредба на програми за компресија со различни формати на датотеки

3.3.3. Програмите за мерење на перформансите на компјутерот

Програмите за мерење на перформансите на компјутерот му дозволуваат дури и на обичниот корисник да утврди дали неговиот компјутер работи како што треба. Програмите што ќе ги разгледаме се многу едноставни и практични. Сите алатки може бесплатно да се преземат и да се инсталираат и самиот корисник може да направи свој избор на алатки доколку нема потреба од сите.

Во самиот оперативен систем Windows 10 постои функција Performance Monitor која овозможува мерење на перформансите на компјутерот. Истата може да се отвори доколку во лентата за пребарување напишеме „**Performance Monitor“**. Оваа програма нуди следење на степенот на употреба на процесорот, RAM меморијата, хард-дискот и мрежната поврзаност во реално време, а постои можност за собирање на податоци и нивна анализа. Иако е дел од оперативниот систем, сепак, оваа функција е за понапредни корисници и потребно е добро познавање на карактеристиките на хардверските компоненти.



Слика 3.5 Графички приказ на работата на процесорот со функцијата Performance Monitor

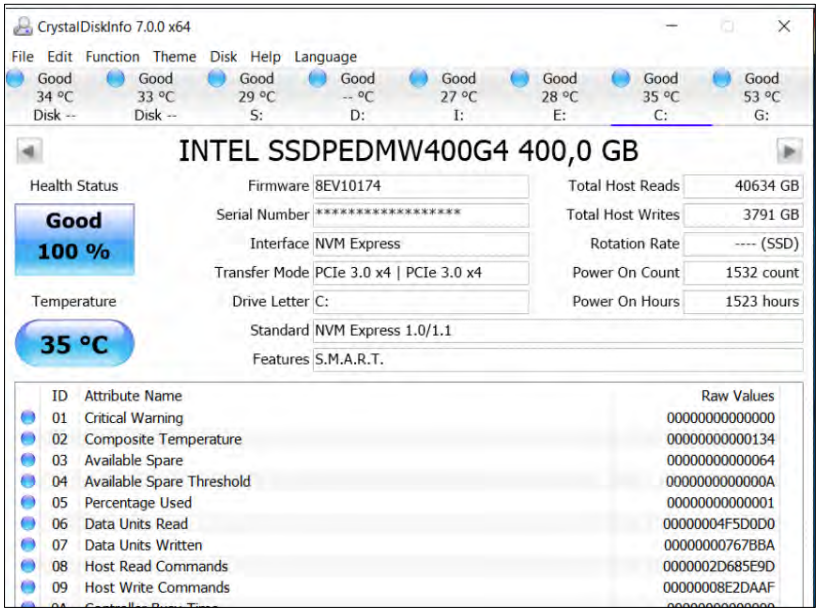
Апликацијата **CPU-Z** може бесплатно да се преземе и да се инсталира од официјалната страница на дистрибутерот CPUID. Оваа апликација врши мониторинг на перформансите на централната процесорска единица, но и другите хардверски компоненти, како што се RAM-меморијата, матичната плоча и графичката картичка. Обичниот корисник може да добие информација за типот на процесор, неговиот производител, брзината на работа на секое јадро, големината на кеш-меморијата, подножјето.

Слика 3.6. Мерење на перформансите на компјутерот со апликацијата CPU-Z

SpeedFan, како што кажува самото име, е програма со која корисникот може да ја следи брзината на вртење на вентилаторите во внатрешноста на

куќиштето и температурата на процесорот, хард-дискот, графичката картичка и други екстерни уреди. Програмата SpeedFan дозволува регулација на брзината на вртење на вентилаторите и ова е многу корисна алатка доколку се работи за куќиште со ограничен простор.

CrystalDiskMark и CrystalDiskInfo се две различни програми од ист дистрибутер за мерење на перформансите на хард-дискот, како што се брзината на читање и пишување и организацијата на складирање на податоци во секој диск посебно.



Слика 3.7. Мерење на перформансите на хард-дискот со апликацијата CrystalDiskInfo

MSI Afterburnerе многу корисна алатка која овозможува лесно и едноставно мерење на перформансите на графичката картичка посебно, доколку корисникот сака да ја зголеми работната фреквенција на графичката картичка (анг. overclock). Со апликацијата MSI Afterburner може да се приспособи такт сигналот на графичката картичка, температурата и максималното напојување, брзината на вентилаторот и друго.

Програмите за мерење на перформансите на компјутерот му овозможуваат на корисникот максимално искористување на хардверските ресурси и навремено преземање мерки на претпазливост во случај на проблем.

3.4. Кориснички програми

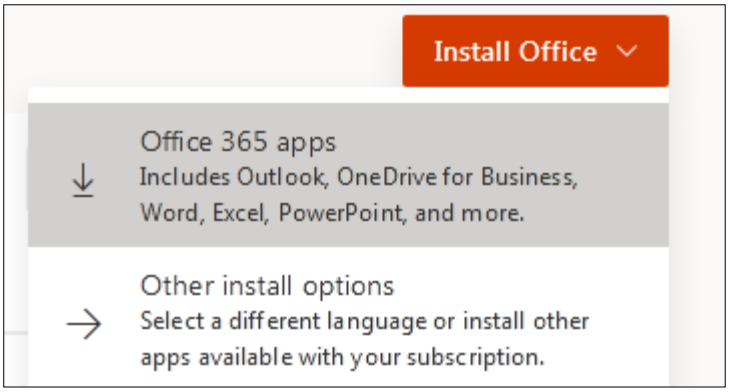
Апликативните програми им овозможуваат на корисниците да извршат одредени задачи со примена на компјутер. Бидејќи се наменети за крајните корисници, се наречени кориснички програми. Покрај специјализираните софтверски компании, креатори на апликативен софтвер можат да бидат и самите корисници. Според намената, корисничките програми можат да се

поделат на повеќе групи. Подоле се набројани овие групи, нивната намена и изборот на програми.

Софтвер за обработка на текст →	• Внесување на текст. Модифицирање на веќе внесениот текст, • Обликување (форматирање) на текстот →	• Notepad • WordPad • MS Word • Word Perfect • OpenOffice – Writer
Софтвер за работа со табели →	• Креирање на табели за внесување, пресметки и анализа на податоци. • Изработка на графикони и графичко претставување на податоци • Форматирање, сортирање и филтрирање на податоци →	• MS Excel • OpenOffice – Calc • Lotos • QuatroPro
Софтвер за изработка на презентации →	• Презентација на производи и услуги • Одржување на настава и предавања • Креирање на реклами и мултимедијални ефекти →	• MS PowerPoint • OpenOffice – Impress • Adobe Flash • Adobe Dreamweaver
Софтвер за обработка на слика →	• Лого дизајн • Дизајнирање на плакати и брошури • Дизајнирање на книги и списанија →	• Векторска графика (CorelDraw, Adobe Ilustrator) • Растерска графика (AdobePhotoShop, CorelPaint, Paint)
Софтвер за интернет-прелистувачи →	• Брз пристап до различни веб-локации, • Прегледување на PDF датотеки, • Можност за репродукција на видео, игри и анимација. • Заштита од опасни веб-локации и анонимно пребарување →	• Opera • Chrome • Mozilla Firefox • Internet Explorer • MS Outlook • Outlook express

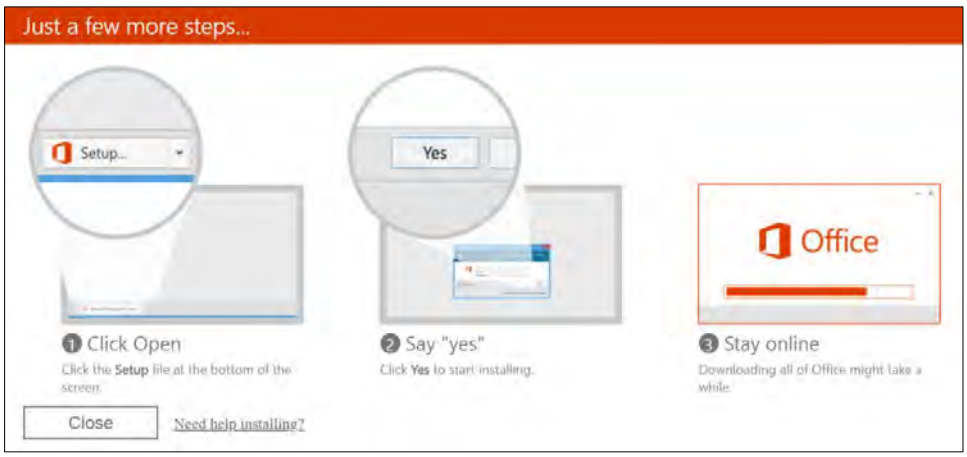
Програмите кои извршуваат слични задачи се нарекуваат **програмски пакети.** Најмногу користени се пакетите за канцелариско работење. Пакетот Microsoft Office е компатибилен со оперативниот систем Windows, а пакетот OpenOffice е компатибилен со оперативниот систем Linux,како што е верзијата Edubuntu. За работа со пакетот Microsoft Office потребна лиценца.

Во 2013 година на интернет се појави платформата **Microsoft Office 365.** Таа е веб-апликација која во себе ги содржи сите програми за канцелариско работење. Во 2020 година сите ученици во Македонија добија свој профил на оваа платформа заради реализација на настава на далечина, преку програмата Microsoft Teams. Доколку се најавиме на платформата Microsoft Office 365 со училишниот профил, можеме бесплатно да го преземеме и да го инсталираме пакетот Microsoft Office на нашиот компјутер. Потребно е да го притиснеме копчето **Install Office** и да ја одбереме опцијата Office 365 apps.



Слика 3.8. Прозорец за инсталација на пакетот Microsoft Office 365

Потоа се појавува нов прозорец кој, всушност, нè води низ процесот на инсталација.



Слика 3.9. Чекори при инсталација на пакетот Microsoft Office 365

Започнува процесот на инсталација, како што е прикажано на слика 3.10. При инсталацијата на овој пакет можеме да избереме една од двете верзии, 32 и 64 битна. Доколку сакаме 32-битната верзија да ја замениме со 64- битната, тогаш мора претходно да ја деинсталираме 32-битната верзија.

Слика 3.10. Инсталација на пакетот Microsoft Office 365

Без оглед дали користиме веб-апликација или истата ја имаме инсталирано на нашиот компјутер, пакетот Microsoft Office 365 нуди многу алатки и готови шаблони кои многу ќе ни помогнат при креирањето на нашите документи.

Заклучоци

Оперативниот систем е познат под името контролно управувачки софтвер. Тој опфаќа програми за распределба на хардверските ресурси, контрола на влезно-излезните операции, управување со меморијата, управување со податоците, преведување на програмски јазици.

Предности на **оперативниот систем Windows** се: едноставност во работењето (ang. user friendly), најголема поддршка за поврзување со периферни уреди, достапност до апликативен софтвер, автоматска инсталација (ang.plug and play), најдобри перформанси за видеоигри, компатибилност со најголем број веб-страници. Недостатоци се: потреба од скапи компјутерски конфигурации, затворен код, слаба сигурност во поглед на интернет-напади и административна заштита, чувствителност на компјутерски вируси, потребна е лиценца, слаба експертска поддршка и често пати за да се зголеми брзината на работа, потребна е преинсталација на оперативниот систем.

Отворен или слободен код значи сите корисници да имаат пристап до изворниот код и да можат да го менуваат во зависност од своите потреби.

Android на компанијата Google и оперативниот систем **Raspbian** се примери за примена и проширување на јадрото на оперативниот систем Linux.

Предности на оперативниот систем Android се: огромен број бесплатни апликации, компатибилност со уреди од различни производители, отворен код, располага со едноставни развојни средини за создавање нови апликации, меморискиот простор лесно се проширува, можност за споделување на

интернет, едноставна комуникација со многу уреди, слобода при конфигурација и избор на апликации. Недостатоци на овој оперативен систем се: апликациите се потешки за изработка поради различните димензии на екраните, помала брзина поради позадинските апликации, помала сигурност и осетливост на вируси.

Антивирусите се софтверски пакети кои се состојат од компјутерски програми што се обидуваат да ги пронајдат и да ги отстранат компјутерските вируси и другите штетни програми. Бесплатни антивирусни програми со добри резултати при тестирањето се: Avira, Bitdefender, Panda, TotalAv, Kasperski и други.

Најпознати програми за компресија во оперативниот систем Windows се: **WINZIP, WINRAR и поновата 7-Zip**. Со компресијата се постигнува заштеда на мемориски простор и заштеда на време кое е потребно за пренесување датотеки.

Програми за мерење на перформансите на компјутерот се: CPU-Z, SpeedFan, MSI Afterburner, CrystalDiskMark, CrystalDiskInfo и Performance Monitor во состав на самиот оперативен систем Windows.

Во кориснички програми спаѓаат: софтвер за обработка на текст,софтвер за работа со табели,софтвер за изработка на презентации,софтвер за обработка на слика и софтвер за интернет-прелистувачи.

Инсталацијата на оперативниот систем Windows може да се изврши на различни начини, преку мемориски модул (USB или DVD) и преку интернет (PHE boot). Доколку користиме мемориски модул, потребно е после приклучувањето на модулот и вклучувањето на компјутерот, да ја **активираме програмата BIOS.**

Клуч на продукт е код од 25 цифри со кои се потврдува лиценцата. Овој код го испорачува Microsoft заедно со доставата на оперативниот систем Windows или може да биде испратен преку е-пошта, доколку програмата за инсталација е преземена од интернет.

Постојат две можности за инсталација на оперативен систем Windows: **Upgrade или Custom.** Првата опција значи надградба на веќе постоечкиот оперативен систем Windows. Оваа инсталација е бесплатна, не бара клуч на продукт, при што не се бришат корисничките податоци.

Windows 10 содржи две **алатки за конфигурирање**: Control Panel и новата апликација Settings.

Категории во апликацијата Settings се: System, Devices, Phone, Network&Internet,

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

Начини за **оптимизација на оперативниот систем**, односно зголемување на брзината на работа на компјутерот се: отстранување на неискористените и стартирачки апликации, ослободување меморија во хард-дискот и негова дефрегментизација, употреба на антивирусни програми, надградба на оперативен систем и зголемување на виртуелните страници.

Виртуелна машина претставува компјутерски систем-гостин (guest) кој ги користи ресурсите на компјутерот-домаќин (host). Во компјутерот-домаќин ќе биде инсталиран оперативниот систем Windows, а во виртуелната машина, односно компјутерот-гостин ќе биде инсталиран оперативниот систем Ubuntu.

Во прозорецот Create Virtual Machine, во полето Name се внесува името на виртуелната машина (на пример Ubuntu), во полето Type се внесува видот на оперативен систем кој ќе се инсталира во виртуелна машина (Linux) и во полето Version се внесува верзијата на оперативен систем (32- битен или 64-битен).

Прашања и задачи

1. Кои програми влегуваат во состав на оперативниот систем?
2. Кои се предностите и недостатоците на оперативниот систем Windows?
3. Кои се почетоците на развој на оперативниот систем Linux?
4. Наведи примери за примена на оперативниот систем Linux!
5. Објасни ја постапката за избор на уред за инсталација на оперативен систем Windows 10!
6. Што претставува таканаречениот клуч на продукт за оперативен систем Windows?
7. Објасни ги инсталациските можности Upgrade и Custom за оперативен систем Windows 10!
8. Кои се двете апликации за конфигурација на оперативен систем Windows 10?
9. Набroj ги категориите во апликацијата Settings!

10. Која категорија од апликацијата Settings се користи за промена на резолуција и осветленост на екран?
11. За што служи категоријата Account во апликацијата Settings?
12. Во која категорија од апликацијата Settings се наоѓаат програмите за откривање вируси и заштита од недозволен пристап?
13. Која е основна цел на оптимизацијата на оперативниот систем?
14. Што подразбираме под поимот „стартирачки (анг.start-up) апликации“?
15. Објасни ја постапката за дефрегментизација на хард-диск?
16. Во која категорија се надградуваат драјверите за периферните уреди?
17. Објасни ја постапката за ресетирање на оперативниот систем Windows 10 без губење на кориснички информации?
18. Кои дополнителни услуги се добиваат заедно со антивирусната програма?
19. Наведи неколку примери за примена на програми за компресија на датотеки!
20. Кои карактеристики на компјутерот можат да се следат со примена на апликацијата CPU-Z?
21. Направи споредба меѓу трите програми за компресија WINZIP, WINRAR и 7-Zip!
22. Набројги видовите кориснички програми и наведи по еден пример од секоја од нив!

4. Микрокомпјутер на плочка

4.1. Arduino микрокомпјутер на плочка

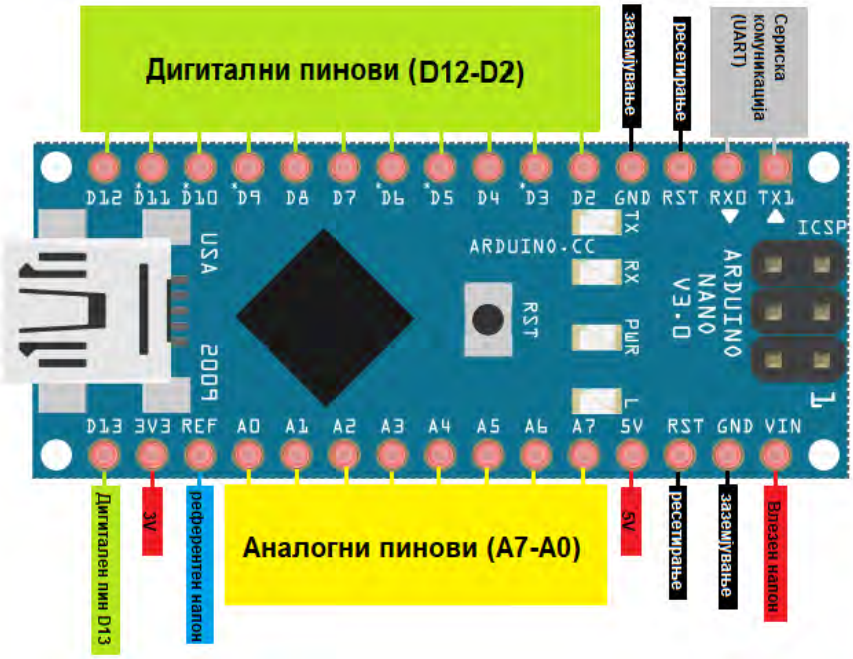
Во првата тема, Основи на компјутерски системи, истакнавме дека најмногубројни микрокомпјутери се персоналните компјутери и микрокомпјутерите на плочка. Микрокомпјутерот Arduino е првата голема **микроконтролерска платформа** со програмски код од отворен тип (анг. open source). Ова значи слободен пристап до многу готови кориснички програми и датотеки кои можат бесплатно да се преземаат од интернет и менуваат со цел да се подобрат перформансите на електронскиот уред.

Ние ќе се запознаеме со хардверските компоненти на Arduino Uno R3 микрокомпјутерот на плочка. Ќе ја проучиме неговата архитектура, влезно-излезните единици, начинот на поврзување, алатките во развојната средина и неколку готови програми што ќе ги имплементираме во програмската меморија. На влез од Arduino микрокомпјутерот можат да се поврзат: тастери, прекинувачи, готови тастатури, сензори (за температура, притисок, проток, движење и др). Излезите можат да се поврзат со најразлични излезни уреди како што се: лед-диоди, светилки, зујалици, мотори, дисплеи. На пример, дали лед-диодата ќе светне или моторот ќе почне да работи зависи од тоа дали тастерот бил притиснат или дали сензорот детектирал промена на некоја физичка величина. Дополнителните штитови (анг. Shields) се електронски плочки што се монтираат врз основниот Ардуино микрокомпјутер, со што се прошируваат неговите основни можности и можат да се извршуваат дополнителни функции како контрола на мотор, поврзување со сензор, безжична комуникација итн.

За програмирање на Arduino се користи програмскиот јазик C/C++, модификација на C и C++. Постои разлика меѓу инструкциските множества за Arduino и за персонален компјутер, бидејќи Arduino манипулира со влезно-излезните уреди. Исто така, самото програмирање бара добро познавање на хардверот посебно на начинот на поврзување со сензорите и извршните единици.

Постојат повеќе од **15 различни модели** на Arduino микрокомпјутери на плочка, а моментално најпопуларни се Arduino Uno R3, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due. Накратко ќе ги објасниме разликите меѓу нив [3] .

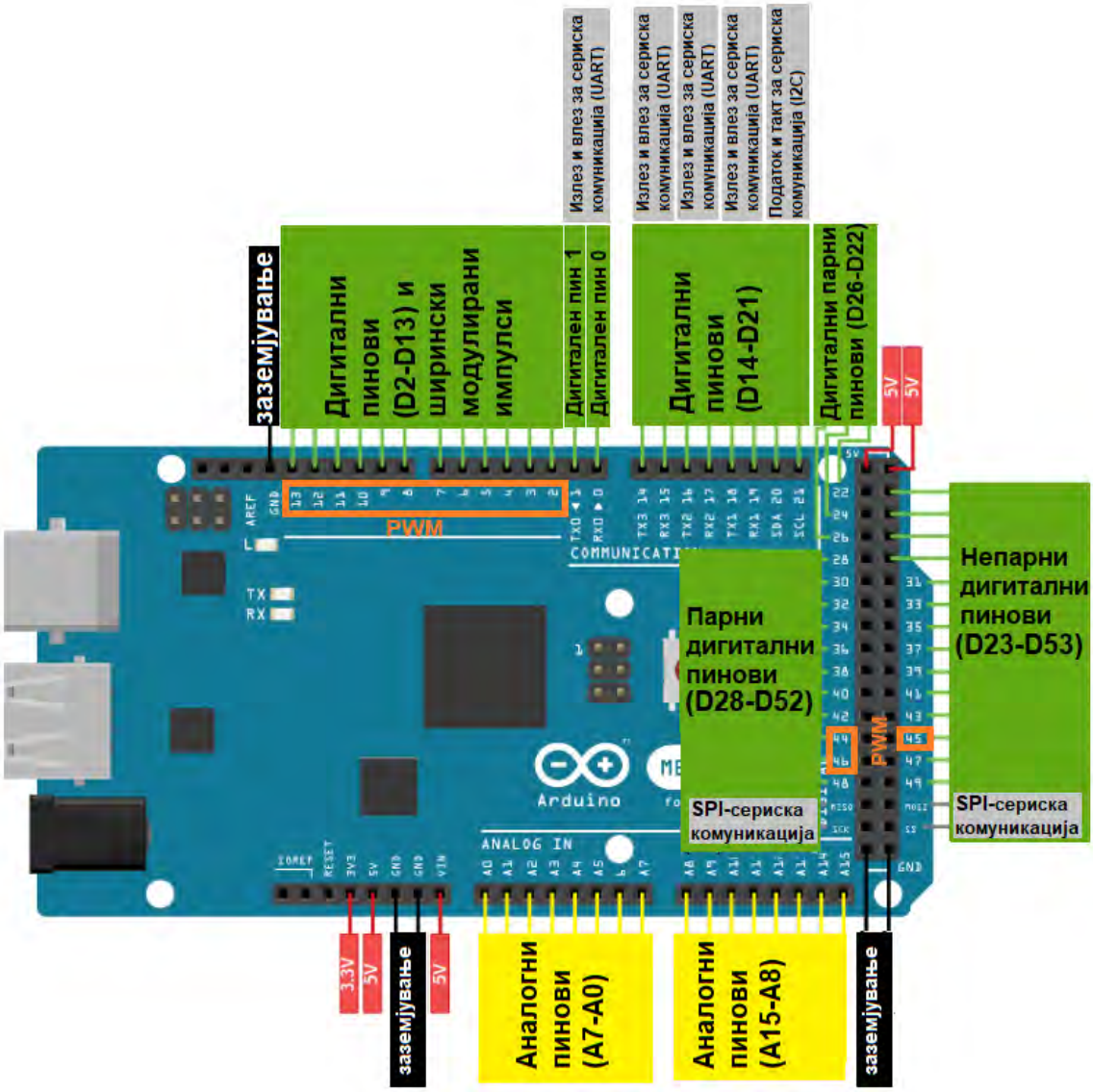
- Третата верзија на **Arduino Uno** (Rev3 или R3) е најдобар микрокомпјутер за почетници и истиот ќе го објасниме и користиме при реализацијата на практичните вежби. Овој микрокомпјутер на плочка има осумбитен ATmega 328 микроконтролер, 14 дигитални влезно-излезни пинови (од кои шест можат да се користат како излези на ширински модулирани импулси), 6 аналогни влезови, USB-конектор, влез за напојување. Тој може да се напојува со батерија или преку AC/DC-конвертор. Сите останати модели на Arduino ќе ги споредуваме со Arduino Uno R3.
- **Arduino Nano**. Arduino Uno R3 и Arduino Nano имаат исти процесори, мемориски капацитети и пинови. Но постојат три основни разлики. Arduino Nano е со двапати помали димензии, што е предност при монтажа. Второ Arduino Nano има машки пинови од двете страни и лесно може да се постави на протоплочка. Бидејќи Arduino Nano има мини USB, тој не може да се поврзе со многу периферни уреди, за разлика од Arduino Uno R3, кој има USB приклучок тип B.



Слика 4.1. Пин дијаграм на Arduino Nano платформа

- **Arduino Pro Mini** е една шестина од големината на Arduino Uno R3. Напонот на напојување е помал и изнесува 3,3 V, за разлика од 5 V на Arduino Uno R3. Има двапати помала работна фреквенција и нема можност за сериска комуникација. Но, голема предност е неговата мала потрошувачка на енергија и тоа го прави погоден за поставување на тешко достапни места.
- Arduino Mega 2560 е како постар брат на Arduino Uno R3. Пин дијаграмот на Arduino Mega 2560 е прикажан на слика 4.2. Arduino Mega 2560 има 54 дигитални влезно-излезни пинови, од кои 15 можат да се користат како излези на ширински модулирани импулси. Располага со 14 аналогни влезови. Овозможува сериска комуникација со четири уреди истовремено

поради четирите вградени универзални асинхрони порти за прием и испраќање на податоци (анг. UART-Universal Asynchronous Receiver-Transmitter).



Слика 4.2. Пин дијаграм на Arduino Mega 2560

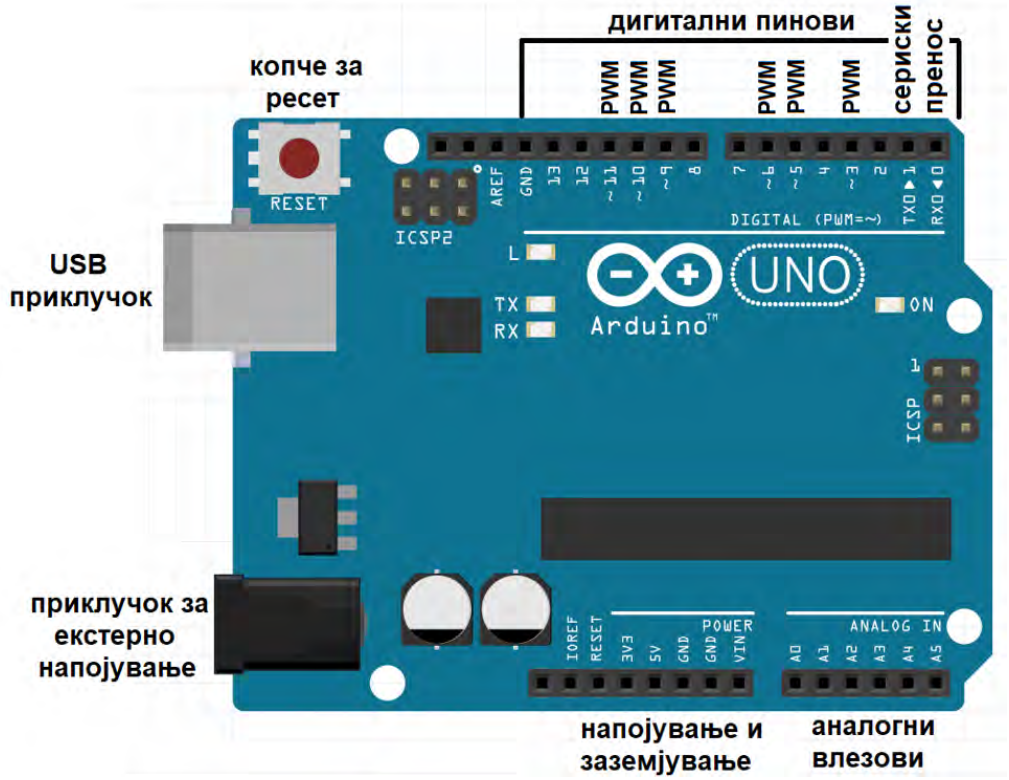
Располага со 256 KB програмска меморија, 8 KB RAM меморија и 4 KB податочна меморија. Гледаме дека работната меморија на Arduino Mega е осумпати поголема од онаа на Arduino Uno R3.

- **Arduino Leonardo**. Основната разлика е во процесорите и во незначително различниот број на пинови. Процесорот на Arduino Leoardo не може да се замени во случај на дефект, но тој има USB-поддршка за поврзување со други уреди, а на Arduino Uno R3 му треба посебен контролер. Но, од друга страна, Arduino Leonardo е некомпатибилен со многу додатоци на Arduino Uno R3.
- **Arduino Due** има многу појак процесор од Arduino Uno R3, но и неговата цена е речиси двапати поголема. На пример, работната фреквенција на

Arduino Due изнесува 84 MHz, наспроти 16 MHz на Arduino Uno R3. Arduino Uno R3 е осум битен, а Arduino Due е 32-битен. Може да извршува 104 милиони инструкции во една секунда. Поради тоа, Arduino Due најчесто се користи за вршење пресметки со децимални броеви кои уште се познати под името броеви со подвижна запирка (анг. floating point number).

4.2. Составни делови на микрокомпјутер на плочка Arduino Uno

Најнапред ќе се запознаеме со хардверот на Arduino Uno R3 микрокомпјутерот на плочка: составните делови, сензорите, актуаторите, електронските компоненти за нивно поврзување и додатоците за зголемување на функционалноста. Подоцна ќе се запознаеме со инструкциското множество, софтверот и со развојната средина на Arduino Uno R3 микрокомпјутерот на плочка.



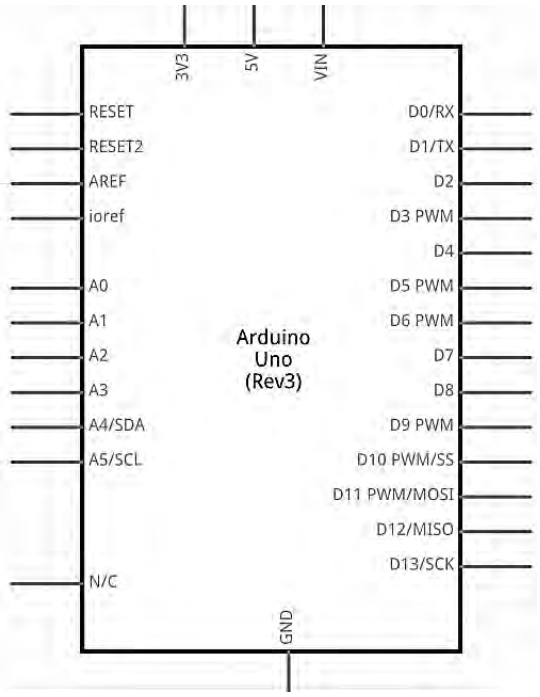
Слика 4.3. Составни делови на Arduino Uno R3 (поглед од горе)

Основна компонента во Arduino Uno R3 е микроконтролерот ATmega328 од производителот Atmel. Овој микроконтролер е 8-битен чип so 28 пинови, изработен во RISC-технологија. Програмска меморија е со капацитет 32 KB, податочната EEPROM-меморија е со 1 KB и внатрешната RAM-меморија изнесува 2 KB. Максималната фреквенција изнесува 20 MHz. Arduino Uno R3 се поврзува со компјутер преку USB-кабел, при што самиот Arduino има мини USB-

приклучок. Преку овој кабел се пренесуваат програмите од компјутерот во Arduino Uno R3 и истите се впишуваат во програмската меморија на микроконтролерот ATmega 328. Откако ќе се испрограмира, Arduino Uno R3 се исклучува од компјутерот, се вградува во друг електронски уред и се користи за процесно управување.

Arduino Uno R3 содржи три сигнални лед-диоди. Лед-диодата со ознака ON е за напојување. Лед-диодите со ознака TX и RX (Transmit-Receive) се покажувачи за сериска комуникација. Овие две лед-диоди поинтензивно трепкаат кога се внесува програма во микроконтролерот. Освен горните лед-диоди, постои уште една вградена лед-диода,означена со буквата L којашто е поставена на 13-тиот дигитален пин и се користи за проверка на исправноста на Arduino Uno микрокомпјутерот преку извршување на програмата за трепкање (анг. Blink).

Кога е поврзан со компјутер Arduino Uno R3 се напојува преку USB-портата. Во спротивно, може да се напојува преку батерија или надворешен адаптер, со напон од 7 до 12 V.

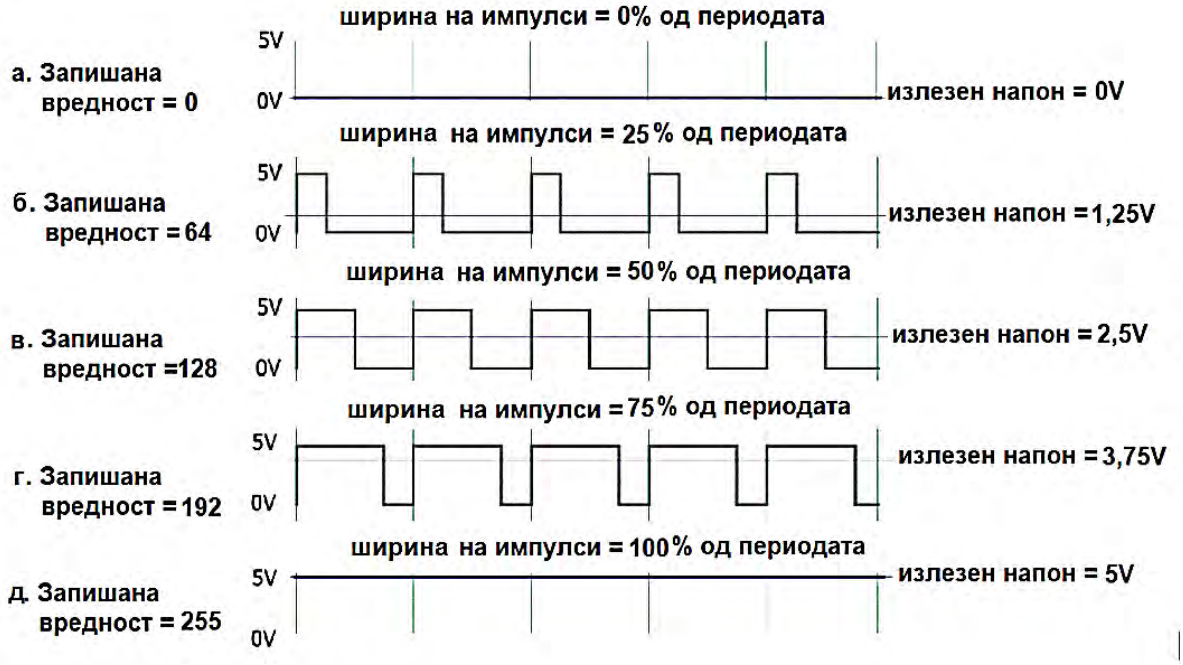


Слика 4.4. Функционална шема на Arduino Uno R3

На слика 4.4. е прикажана функционалната шема на Arduino Uno R3. Тој располага со **14 дигитални пинови.** Дигиталните пинови можат да бидат влезни или излезни, но не двонасочни. Наједноставни елементи за работа со дигиталните пинови се тастерот како влезен елемент и лед-диодата како излезен елемент. Дигиталните сигнали може да имаат само две вредности, логичка нула или логичка единица. На пример, кога тастерот ќе биде притиснат, микроконтролерот добива податок логичка единица (се поврзува на напон од 5V), а кога е отпуштен, тоа е состојба на логичка нула (поврзаност со заземјување).

Пиновите со реден број 3, 5, 6, 9, 10 и 11 имаат двојна функција. Тие може да се користат како дигитални пинови или аналогни излези. Аналогните излези се обележани со ознаката PWM (анг. Pulse Width Modulation) што во превод значи импулсна-ширинска модулација. Со оваа постапка аналогниот сигнал се претвора во низа од импулси и паузи. При програмирање на Arduino Uno R3 и работа со аналогни излези, програмерот задава (запишува) целобројни вредности во опсегот од 0 до 255. Зависноста на ширината на импулсите од големината на целобројната вредност е прикажана на слика 4.5. Ширината

односно времетрањето на импулсите и големината на излезниот напон зависат правопропорционално од запишаната целобројна вредност. На пример, вредноста 65 претставува 25 % од максималната дозволена вредност 255, затоа ширината на импулсите ќе изнесува 25% од вредноста на периодата и излезниот напон ќе биде 25% од неговата максимална вредност (5V) што е еднакво на 1,25 волти. Импулсно ширинската модулација се користи и кај аналогните влезови, но со обратен редослед односно наместо запишување се врши читање. Влезниот напон, кој се движи во опсегот од 0V до 5V, се претвора во поворка од импулси со променлива ширина и ширината на импулсите ја одредува целобројната вредност што ќе се прочита како влезна големина. Целобројните вредности на влез се движат во опсегот од 0 до 1023 односно во четири помал опсег од оној на излезот. [4]



Слика 4.5. Примена на импулсна ширинска модулација кај аналогни излезни пинови

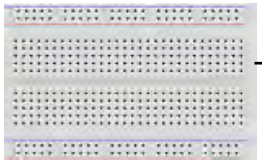
Исто така, првиот и нултиот пин од Arduino Uno R3 имаат двојна функција. Освен како дигитални пинови истите може да се користат како пинови за сериска комуникација, на пример при поврзување на две Arduino платформи или Arduino со Raspberry Pi микрокомпјутерот на плочка.

Пинот за заземјување (анг. GND, Ground) и пинот за напон од 5 V се користат за напојување на протоплочка, на која прецизно се поврзани сите електронски компоненти според одредена електрична шема.

4.3. Периферни елементи и електронски компоненти за поврзување со Aduino микрокомпјутер на плочка

За Arduino Uno да комуницира со надворешниот свет и да прима информации за средината што го опкружува, потребни се влезни единици: **сензори** (пиезо, тилт-сензор, фотоотпорник, температурен сензор), тастери, потенциометри. За Arduino платформата да делува врз одреден процес, потребни се излезни единици, познати под името актуатори. Во оваа категорија спаѓаат: дисплеи, мотори, зујалици.

За влезно-излезните единици ефикасно да се поврзат со Arduino Uno R3 микрокомпјутерот, потребни се дополнителни електронски компоненти како што се: отпорници, кондензатори, насочувачки диоди, оптокаплери, насочувачи итн. Накратко ќе ја опишеме нивната функција.[4]



протоплочка

Протоплочката е одличен избор за почетници што немаат големи предзнаења од електроника и лемење. Под пластичната плоча, поставени се вертикални и хоризонтални проводници. Во отворите на пластичната плоча се поставуваат изводите на електрични компоненти и жици за нивно поврзување, краткоспојници. Упатството за работа со протоплочка е дадено во практичниот дел на модуларната единица.

Лед-диоди

Лед-диодите се наједноставни излезни елементи. За лед-диодата да свети, потребно е подолгиот приклучок (анода) да се поврзе на повисок потенцијал во однос на пократкиот приклучок (катода).

Тастери

Тастерите имаат четири приклучоци, по два од две спротивни страни. **Приклучоците од иста страна не се електрично поврзани** и тие треба да се постават во отворите на два различни проводници на протоплочката, за електричното коло да се затвори кога тастерот ќе се притисне.

Оптокаплер

Оптокаплерот е составен од лед-диода и фотодиода. Низ колото на фотодиодата протекува струја само додека свети лед-диодата. Лед-диодата е поврзана со извор за напојување, а фотодиодата со потрошувач. Со помош на оптокаплерот струјните кола на изворот и потрошувачот се електрично изолирани.

Потенциометар

Потенциометарот е променлив отпорник и најчесто е поврзан со еден од аналогните влезови на Arduino Uno R3 со цел да се создаде променлив напон.

Фотоотпорник

Фотоотпорникот има променлива отпорност, во зависност од интензитетот на светлината што паѓа на неговата површина.

Мотор на еднонасочна струја

Моторот на еднонасочна струја ја претвора електричната енергија во механичка, вршејќи кружни движења. Ако се промени насоката на струјата, ќе се промени и насоката на вртење на моторот.

Серво мотор

Серво-моторот не се врти за полн круг, туку само до 180 степени. Во зависност од влезниот напон, тој ќе се помести за определен агол, до одредена позиција, и ќе остане во таа позиција сè додека не се промени влезниот напон. Аголот на вртење зависи од ширината на импулсите по извршената импулсно-ширинска модулација.

Чекорен мотор

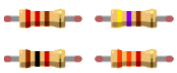
Чекорните мотори вршат дискретни поместувања односно со доаѓањето на секој влезен импулс моторот се придвижува за определен агол. Ова поместување се нарекува чекор. Колку чекори се потребни за да се направи полн круг зависи од конструкцијата на моторот (број на полови) .

Релеј

Релеите овозможуваат вклучување или исклучување на потрошувачи со големи моќности преку употреба на нисконапонски или нискострујни сигнали. Релејот може да има нормално затворени контакти и нормално отворени контакти. Во случај на нормално отворени контакти тогаш колото на потрошувачот е во прекин кога и управувачкото коло е во прекин. Во случај на нормално затворените контакти тогаш потрошувачот е поврзан со неговиот извор на напојување кога управувачкото коло е во прекин.

Пиезо-компонентата

Пиезо-компонентата може да се користи како сензор за вибрации или генератор на тонови со различна фреквенција.

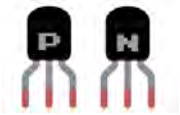


Отпорници

Отпорниците се користат за приспособување на јачината на работната струја за да не дојде до оштетување на елементите.

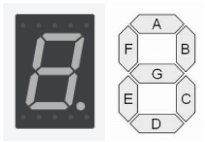
Тилт-сензор

Тилт-сензор е сензор за позиција. Во неговата внатрешност постои метално топче кое се поместува и на таков начин го отвора или затвора контактот односно ги прекинува или ги поврзува приклучоците на сензорот. Во која позиција, хоризонтална или вертикална, ќе биде затворен контактот зависи од типот на сензорот.



Транзистори

Транзисторите се користат како напонски или струјни засилувачи и прекинувачи. Транзисторот 2N2222 е погоден за струи до 500 mA, а транзисторот TIP120 за струи до 5 A.

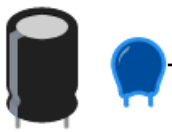


7-сегментен LED екран

7-сегментниот екран се користи за визуелно прикажување на декадни цифри. Седум пинови се користат за побуда на сегментите означени со латиничните букви од a до f, еден пин е за точката и два се заземјување. 7-сегментниот екран може да биде со заедничка анода или заедничка катода и ова е важно за побудување на пиновите.

Насочувачи

H-мостот (анг. H-bridge) е електронско коло кое овозможува промена на насоката низ моторот за еднонасочна струја. Промената на насоката на струјата предизвикува промена на насоката на вртење на моторот. Се изработува од прекинувачки елементи или во облик на интегрирано коло како што е L239D.



Кондензатори

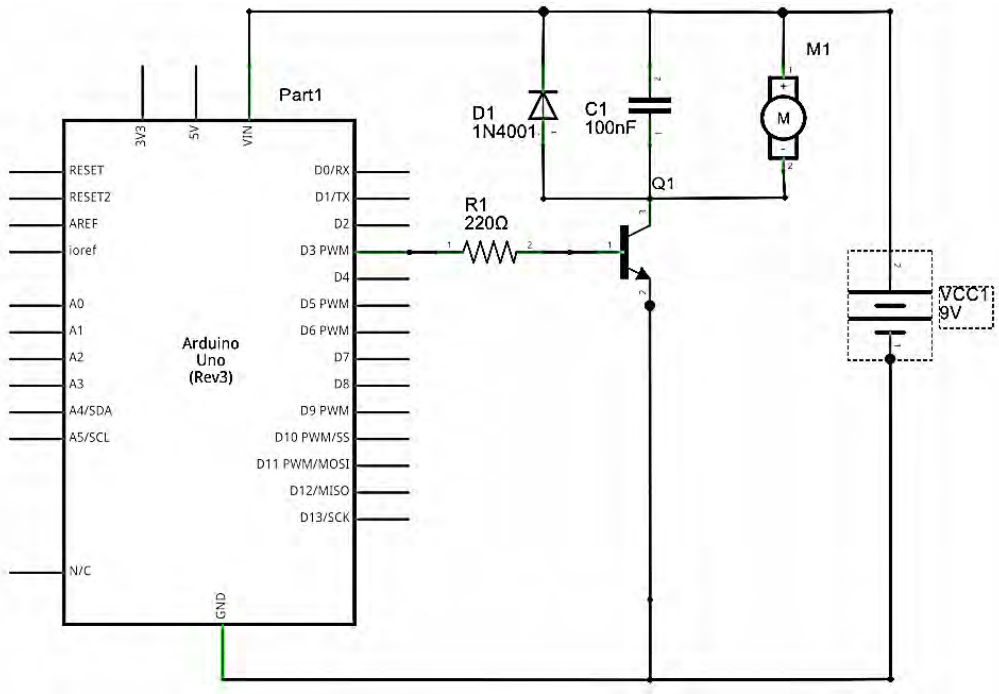
Ако напонот на кондензаторот е помал од напонот на изворот, тогаш кондензаторот се полни, а во спротивно се празни. Вообичаено кондензаторите се поврзуваат паралелно со сензор или со мотор, со цел да се спречи брза промена на напонот, најчесто предизвикана од други компоненти во колото.

Пред да започнеме со поврзувањето на компонентите на протоплочката потребно е да се изврши анализа на нивните карактеристики и избор на истите. Секој производител на електронски компоненти изготвува техничко-технолошка документација која содржи информации за перформансите, начинот на употреба и карактеристиките, како што е вредноста на минималниот напон кој е потребен за работа на компонентата и максимално дозволениот напон. Некои компоненти како што се отпорникот и обичниот кондензаторот се неполаризирани компоненти и кај нив не е важно кој од изводите ќе биде на повисок, а кој на

понизок електричен потенцијал. Диодите, транзисторите, интегрираните кола се поларизирани компоненти и е потребно да се изврши индентификација на изводите.

Со Arduino Uno R3 може да се поврзуваат компоненти со работен напон до 5V и максимално дозволена струја од 40mA. Овие вредности не се исти за сите Arduino платформи. На пример, Arduino Uno WiFi Rev2 платформата работи со 20mA максимално дозволена струја, а Arduino Zero со 7mA.

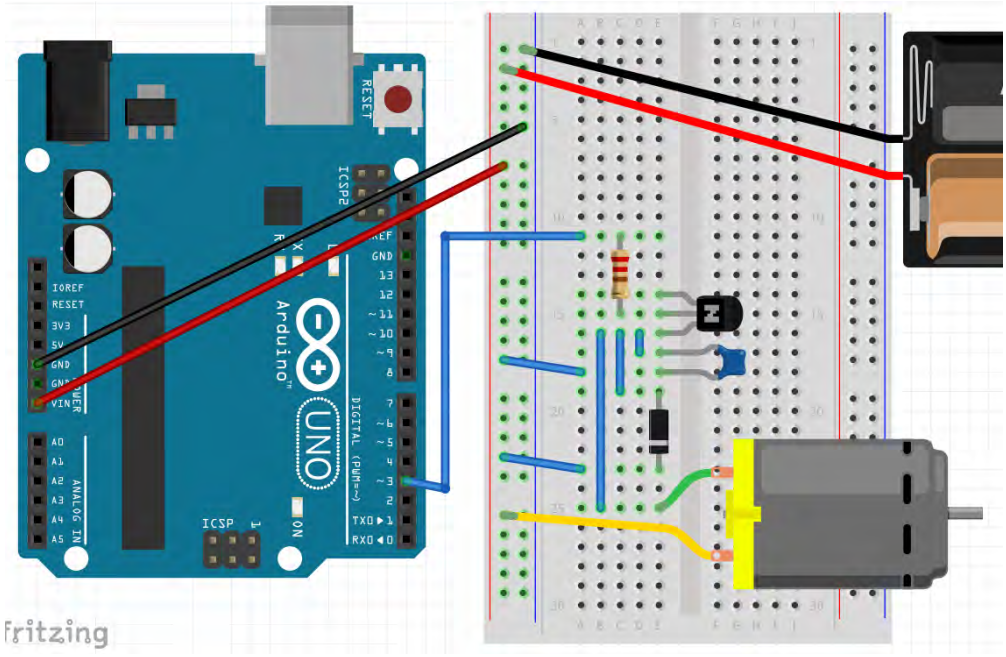
На слика 4.6. и слика 4.7. се прикажани функционалната и монтажната шема за поврзување на мотор на еднонасочна струја со Arduino Uno R3 микрокомпјутерот. Во функционалната шема симболите ги претставуваат електронските компоненти, а линиите начинот на нивно поврзување. Ако електричното коло е едноставно, со мал број на компоненти, можеби монтажната шема е полесна за разбирање. Но, во случај на голем број на компоненти и жици за нивно поврзување тогаш неопходна е функционална шема зошто таа ни нуди поголема прегледност [5].



Слика 4.6. Функционална шема за поврзување на мотор на еднонасочна струја со Arduino Uno R3

Во шемата прикажана на слика 4.6. транзисторот има функција на прекинувач. Базата на транзисторот е поврзана со третиот пин на Arduino Uno R3 кој се користи како аналоген излез со ширински модулирани импулси и од времетраењето на импулсите зависи големината на напонот за напојување на моторот односно јачината на струјата која тече низ него. Во техничко-технолошката документација на транзисторот треба да провериме дали максимално дозволениот напон колектор-емитер е поголем од напонот на напојување на електричниот мотор и јачината на колекторската струја треба да биде 25% поголема од струјата што тече низ моторот. Базната струја зависи од

колекторската струја и коефициентот на струјно засилување. На пример, ако коефициентот на струјно засилување изнесува 100, а посакуваната колекторска струја изнесува 1A тогаш базната струја ќе изнесува 1A/100=0,01A=10mA. Со помош на Омовиот закон можеме да ја пресметаме вредноста на отпорникот поврзан со третиот дигитален пин. На пример 5 / 0,01 = 500Ω и бидејќи ова не е стандардна вредност избираме 470Ω.



Слика 4.7. Монтажна шема за поврзување на мотор на еднонасочна струја со Arduino Uno R3

Диодата е поврзана паралелно со моторот и ги штити пиновите на Arduino Uno R3 од појава на големи индукциони струи, кои се јавуваат при вклучување и исклучување на моторот. Промената на состојбата на дигиталните пинови може да биде многу брза и во колото се јавуваат осцилации во напонот на напојување и истите се филтрираат со употреба на кондензатор.

Напојувањето на Arduino Uno е уште една карактеристика на која треба да се внимава. Веќе знаеме дека Arduino Uno R3 може да се напојува на два начини, со USB или екстерно напојување. После впишувањето на програмата нема потреба од поврзување на Arduino Uno R3 со компјутер и тогаш може да се користи екстерно напојување, батерија или адаптер кој наизменичниот напон го претвора во еднонасочен со вредност од 7 до 12V. Напони поголеми од 12V може да предизвикаат оштетување на електронската плоча. Напонот на батеријата исто така треба да биде во опсегот од 7 до 12V. Капацитетот на батеријата е даден во mAh. Колку ќе трае батеријата зависи од потрошувачката на електрична енергија на компонентите вградени во електронскиот уред.

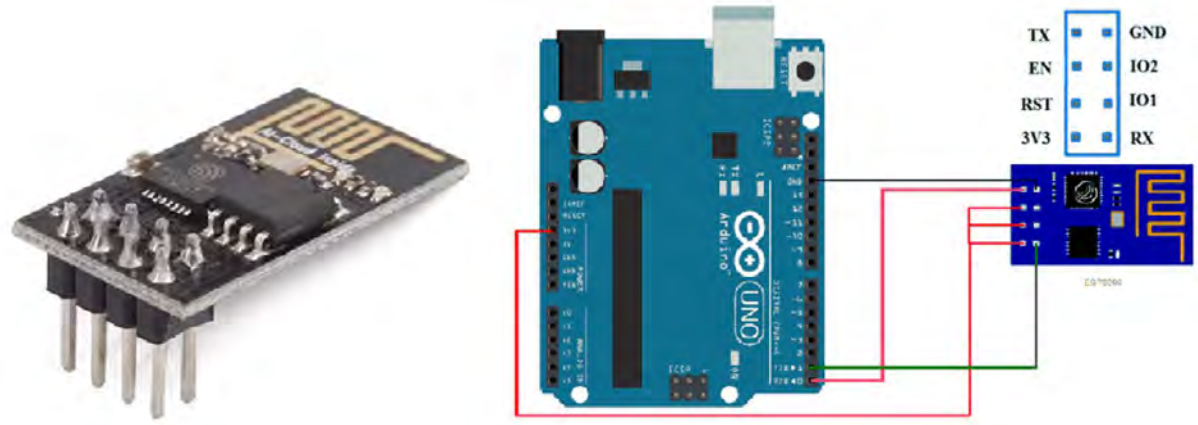
Анализата на шемата за поврзување на Arduino Uno R3 со мотор на еднонасочна струја е само еден пример за тоа како треба да се проучат функцијата и карактеристиките на електронските компоненти пред истите да се поврзат со Arduino Uno R3 микрокомпјутерот на плочка.

4.4. Додатоци за Arduino микрокомпјутер на плочка

Arduino микрокомпјутерот на плочка стана многу популарен меѓу неговите корисници, за што голем придонес даде и концептот на отворен извор, како на хардвер така и на софтвер. Многу корисници кои експериментираат со дизајнот на своите уреди им дадоа идеја на производителите на електронски компоненти да креираат и пуштат во продажба готови додатоци со кои многу се олесни работата со Arduino и значително се зголеми нивната функционалност. Додатоците се всушност готови електронски плочи со точно определена намена. Поврзувањето е едноставно и најчесто додатоците содржат долги машки пинови кои се поставуваат во женските пинови на Arduino. Штитовите (анг. Shields) се додатоци коишто се поставуваат врз основната Arduino-плоча при што двете плочи се една врз друга, но не се допираат поради долгите машки пинови. Називот штит има симболично име. Бидејќи Arduino софтверот е од отворен тип, корисниците имаат пристап до огромен број на готови програмски кодови за додатоци кои можат да ги менуваат според своите потреби. Исто така поголемиот дел од овие додатоци имаат софтверска поддршка во самата интегрирана развојна средина на Arduino. При користење на додаток, во програмскиот код мора да се повика таканаречената библиотека без која додатокот не може да комуницира со Arduino. Библиотеките содржат додатни инструкции со кои многу се олеснува манипулацијата со хардверот. Како што ќе видиме подоцна, големината на додатоците е најразлична, со димензии од 1-2 сантиметри и 4 пинови, до додатоци кои се со иста големина и број на пинови како и самата Arduino развојна плоча, какви што се штитовите. Обично додатоците за Arduino не се за почетници и во практичниот дел нема да ги користиме, но добро е да се познаваат можностите кои ги нудат истите. Ќе се запознаеме со еден додаток и пет штитови.

Без интернет пристап или Wi-Fi приклучок Arduino не може да комуницира со надворешниот свет, да испраќа податоци добиени од сензорите или да се управува од далечина. За вмрежување на Arduino потребен е посебен додаток како Wi-Fi додаток или Ethernet штит. Додатокот ESP8266-01 е Wi-fi модул со многу ниска цена на чинење и димензии 25mmx15mm што го прави посебно погоден за вмрежување на Arduino Uno R3 во апликации на платформата Интернет на нештата.

Надворешниот изглед и монтажната шема за поврзување на ESP8266-01 додатокот со Arduino Uno R3 е прикажан на слика 4.8. Тој содржи осум пинови и истите не може да се постават во пиновите на Arduino Uno R3 туку се користат жици за поврзување. Додатокот ESP8266-01 нема вграден USB приклучок туку се користи универзална асинхрона сериската комуникација преку двата пинови TX и RX. Двата влезно-излезни пинови (анг. IO – Input Otput) може да се искористат за поврзување на пример на сензор [5].



Слика 4.8. Надворешниот изглед и монтажната шема за поврзување на ESP8266-01 додаток

Максималната оддалеченост за Wi-Fi поврзаност изнесува од 100-250 метри. Библиотеката на овој модул не е дел од интегрираната развојна средина на Ардуино туку е потребна дополнителна инсталација.

На слика 4.9. е прикажан штитот Arduino Ethernet 2 кој поддржува осум истовремени конекции. Се поврзува со Arduino Uno R3 преку долги машки пинови и бидејќи пин дијаграмот е ист можно е врз него да се нададе уште еден штит. За поврзување со интернет мрежата се користи RJ-45 приклучок и постои вграден слот за мемориска картичка за чување на документи кои треба да се сервисираат во мрежата.

Слика 4.9. Arduino Ethernet 2 штит

GSM/GPRS Arduino штитот е додаток кој овозможува вмрежување на Arduino во мобилна мрежа.

Слика 4.10. GSM/GPRS Arduino штит

Корисникот може да прима и испраќа SMS пораки, да се јавува и одговара на повици и пристапува до интернет преку глобалната мобилна мрежа и пакетски ориентираниот сервис за пренесување на податоци (анг. Global System Mobile / General Packet Radio Service).

Штитот располага со вграден слот за SIM картичка, конектори за слушалки, антена, Bluetooth 3.0 со антена и се она што е потребно за да Arduino Uno R3 има функција на мобилен телефон.

На слика 4.11. е прикажан Arduino штит за мотори, кој може да контролира четири мотори на еднонасочна струја или два серво мотори или 2 чекорни мотори.

Слика 4.11. Arduino штит за мотори

Овие мотори не може да бидат директно контролирани само со Arduino поради јаките струи и големите моќности. Arduino мотор штитот врши струјно одвојување на изворот за напојување на моторот и ја користи логиката на Arduino платформата. Содржи две интегрирани кола L293D и претставува две-канален H-мост за контрола на вртежи на мотор.

Arduino не е дизајниран за контрола на потрошувачи со големи моќности, максималниот напон изнесува 5V, со работна струја до 40mA.

Слика 4.12. Arduino релеј штит

Релеите обезбедуваат галванско одвојување на нисконапонскиот од високонапонскиот дел на системот за управување со моќни потрошувачи. На слика 4.12. е прикажан две-канален Arduino релеј штит со 250V излезен наизменичен напон за потрошувачкото коло или 30V еднонасочен напон и 10A максималната струја.

Не е возможно да се набројат сите додатоци кои се користат за проширување на функционалноста на Arduino микрокомпјутерите на плочка. Без оглед на изборот секогаш прво треба да се прочита техничко-технолошката документација која ги содржи сите информации за начинот на поврзување и заштита на уредите од евентуална штета.

4.5. Програмирање на Arduino микрокомпјутер на плочка во програмскиот јазик C/C++

Денес, програмирањето е предизвик за многу млади луѓе, токму поради појавата на многу софтверски алатки, развојни средини кои се лесни за употреба и коишто овозможуваат да се создадат апликации без примена на сложени математички модели. За успешно програмирање на Arduino микрокомпјутерот на плочка потребно е познавање на две софтверски специфичности, неговата развојна средина и неговото инструкциско множество.

Arduino платформата не располага со оперативен систем и затоа неа не можеме да ја програмираме без да ја **поврземе со персонален компјутер.** За пишување и внесување на програмите во микроконтролерот на Arduino потребна ни е развојна програма, или уште позната како **развојна средина (**анг. IDE – Integrated Development Environment).

Со начинот на инсталирање на интегрираната развојна средина за Arduino микрокомпјутерот и нејзините системски програми ќе се запознаеме во практичниот дел на оваа модуларна единица.

Arduino може да извршува само една програма и нејзиното извршување започнува веднаш по вклучувањето на напојувањето. За полесно совладување на програмирањето на Arduino се препорачува употреба на Tinkercad симулаторот кој исто така ќе биде објаснет во практичниот дел на оваа модуларна единица.

За програмите напишани за Arduino се користи називот скица (анг. sketch). На ваков начин се нагласува едноставната структура на програмите, за брза и лесна реализација на идеите. Скицата е составена од инструкции. Инструкциите се искази што иницираат одредено дејство за да се добие посакуваниот резултат. Преку инструкциите, ние го контролираме Arduino микрокомпјутерот и вршиме пресметки.

Скицата содржи две задолжителни структури: setup и loop. Пример 4.1 е општ приказ на двете задолжителни структури во скицата.

setup() ⟶ Setup е функција без влезни аргументи. Со неа **се задаваат почетни вредности на променливите, се конфигурираат пиновите** како влезни или излезни, **се избира брзината за пренос на серискиот монитор, се специфицираат вклучените библиотеки**. Оваа функција се извршува само еднаш, кога ќе се вклучи напојувањето или по рестартирањето на Arduino микрокомпјутерот.

loop() ⟶ Со оваа функција, всушност, вршиме контрола на Arduino микрокомпјутерот. Англискиот збор loop во превод значи јамка и како што самото име покажува, оваа функција постојано се

извршува, се врти во круг, **постојано ги следи промените на состојбата на влезните пинови и соодветно реагира.** Всушност функцијата loop претставува бесконечен циклус кој се повторува сè додека Arduino има напојување.

Пример 4.1.

```
1  int buttonPin = 3;              //  Прво се декларираат променливите.
2  void setup() {                  //  setup е секогаш почетна функција.Ознаката void
                                   //  значи дека функцијата нема да даде повратна
                                   //  вредност.
3    Serial.begin(9600);           //   Дефинираме брзината за пренос на серискиот
                                   //  монитор.
4    pinMode(buttonPin, INPUT);    //   Конфигурација на влезен пин.
5  }
6  void loop() {                   //  Почеток на функцијата loop.
   .......
   }                               //  Крај на функцијата loop.
```

На почетокот од програмирањето на Arduino, може да се користат готови програмски кодови, но за понатамошна успешна работа неопходна е анализа на истите, со цел нивно менување и приспособување. Предизвик е учениците да научат да составуваат свои програми согласно карактеристиките на електронските компоненти кои ќе ги користат. Но, пред да започнеме со креирање и анализата на програми за Arduino микрокомпјутерот, ќе се запознаеме со основните градбени елементи на програмскиот јазик C/C++: променливите, инструкциите и структурите. Потребно е да се разбере нивното значење за компјутерот и да се запамети нивната синтакса. Синтакса во програмирањето значи множество правила за подредување на ознаките, симболичните имиња, операторите, интерпункциските знаци, коментарите, со цел да се добие исказ разбирлив за компјутерот.

4.6. Променливи и оператори во програмскиот јазик C/C++ за Arduino микрокомпјутер

Променливите ги чуваат податоците што се обработуваат во компјутерот. За секоја променлива се предвидува и се резервира место во меморијата. Секое резервирано место има своја адреса. Бидејќи адресите се тешки за паметење, на променливите им се даваат симболични имиња. Освен симболичното име, на секоја променлива мора да ѝ се додели и ознака за вид на податок. Големината на резервираниот мемориски простор зависи од видот на податокот. Исто така, компјутерот различно се однесува спрема различни типови на податоци. На пример, со аритметичките инструкции можат да се

обработуваат броеви, но не и текстуални податоци. Подолу се дадени ознаките на основните **типови на податоци**.

int ⟶ Целите броеви се најчесто користени податоци. Arduino платформата резервира два бајти мемориски простор за секој цел број. Опсегот на цели броеви изнесува од -32768 до +32 767.

float ⟶ Децималните броеви се користат за прикажување аналогни вредности. Децималните броеви со 7 децимални места зафаќаат простор од 4 бајти во меморијата на Arduino и опсегот на броеви изнесува од 3,4028235E+38 до -3.4028235E+38. Бројот по буквата E претставува експонент на степенот со основа 10 (пример $35E3=35\cdot10^3=35\cdot1000=35000$).

double ⟶ Децималните броеви со 15 децимални места се попрецизни од децималните броеви со 7 децимални места и тие зафаќаат мемориски простор од 8 бајти.

char ⟶ Еден знак (анг. character) претставува една буква, цифра или алфанумерички знак. Еден знак зафаќа меморија од еден бајт и опсегот на вредности изнесува 256 знаци. Знаците се пишуваат помеѓу единечни апострофи (пример ‘A’).

string ⟶ Текстот претставува низа од симболи и тој се пишува помеѓу наводници (пример “dobar den”)

bool ⟶ Логичките податоци имаат само две вредности, нула или еден, неточно (анг. false) или точно (анг. true). Една логичка променлива зафаќа простор од еден бајт.

Секоја променлива што ја користиме во програмата мора да ја декларираме (највиме). Најнапред се наведува типот на податок, а потоа името на променливата. Ако при декларацијата на променливата и доделиме и некоја вредност велиме дека сме направиле иницијализација. Подоцна вредноста на иницијализирана променлива може да се промени каде било во програмата. На променливите им се доделува вредност со знакот еднакво (=) и тој знак е познат и под името оператор за доделување. Константите се податоци чија вредност не се менува во текот на програмата. Ако се работи за константа, тогаш пред типот на податокот треба да стои зборот const.

Пример 4.2.

```
1  int countUp = 0;                  // Целобројна променлива со симболично
                                     // име 'countUp'.
2  const float pi = 3.14;            // Децимален број со константна вредност.
3  bool isCodingFun = true;          // Логичка променлива со симболично име
                                     // isCodingFun и вистинита вредност.
```

```
4  string stringOne = "Hello String";  // stringOne е името на текстуална
                                       // променлива, а Hello String е неговата
                                       // вредност.
```

Низите се групи од променливи од ист тип. Декларацијата на низата содржи: вид на податоци, име на низата и број на елементи. Средната заграда го содржи бројот на елементи, а во големата заграда се набројани елементите. За пристап до елементите повторно се користи средната заграда односно се пишува името на низата и редниот број на елементот во средна заграда.

Пример 4.3.

```
1  string avto[3]={"Volvo", "BMW", "Ford"};  // Низата е група од текстуални
                                             // променливи, avto е името на низата,
                                             // бројот 3 во средната заграда значи
                                             // дека низата содржи 3 елементи, а
                                             // елементите се во големата заграда.
2  avto[0] = "Opel";                         // Го повикуваме елементот со реден
                                             // број нула и ја менуваме неговата
                                             // вредност.
```

Низите може да бидат и дводимензионални и тогаш елементите се распоредени во редици и колони, а броевите во двете средни загради означуваат број на колони и број на редици.

Пример 4.4..

```
1  int Table [2][3] = {{3, 42,1}, {7, 3, 12}};   // Низата е составена од 2 колони и 3
                                                  // редици.
```

Операторите се знаци со кои програмерот, според точно определени правила, гради искази, инструкции. Постојат повеќе типови оператори: математички, логички, споредбени, оператори за доделување. Накратко ќе се задржиме на секој од нив.

Математичките оператори се познати под името аритметички оператори. [6]

+	Оператор за собирање
-	Оператор за одземање
*	Оператор за множење
/	Оператор за делење
%	Оператор за пресметка на остаток при делење

Пример 4.5.

```
1  float a = 5.5;
2  float b = 6.6;
3  int c = 0;
4  c = a - b;        // Бидејќи 'c' е целобројна променлива, нејзината вредност ќе
                     // биде -1 иако резултатот од одземањето е -1,1
```

Пример 4.6.

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

```
2  int b = 10;
3  int c = 0;
4  c = a * b;          //  Променливата 'c' добива вредност 50 .
```

Пример 4.7.

```
1  float a = 55.5;
2  float b = 6.6;
3  int c = 0;          //  Бидејќи променливата 'c'  е целобројна променлива, таа
4  c = a / b;          //  добива вредност 8 иако резултатот од делењето е  8.409.
                       //  [6]
```

Пример 4.8.

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

Исказите што содржат **споредбени инструкции** даваат резултат true (точно) или false (неточно). Најчесто се користат во условни функции. Честопати, со цел да се намали должината на исказите, се користат посебни оператори.

- != ⟶ Ова е оператор за „не е еднакво“. Се споредуваат две променливи (x и y) или променлива и константа и ако се различни, исказот е вистинит. Променливите можат да бидат целобројни, децимални или бинарни броеви. Примената на овој оператор е прикажана во пример 4.10.
- < ⟶ Оператор за помало. Променливата од левата страна на операторот треба да има помала вредност од променливата од десната страна.
- > ⟶ Оператор за поголемо
- <= ⟶ Оператор за помало или еднакво
- >= ⟶ Оператор за поголемо или еднакво
- == ⟶ Оператор за еднаквост

Пример 4.10.

```
1  if (x != y) {       //  Дали x е различно од y?
2  .....               //  Програмскиот код меѓу големите загради  се извршува
:  }                   //  само ако исказот во малите загради е вистинит.
```

Пример 4.11.

```
1  if (x <y) {         //  Дали x е помало од y?
2  .....               //  Извршување на програмскиот код меѓу големите загради
:  }                   //  зависи од исполнетоста на горниот услов
```

Логичките инструкции се нарекуваат уште и булови инструкции бидејќи тие можат да се применат на бинарни променливи (анг. bool). Резултатот е исто така бинарна вредност, нула или еден, неточно (анг. false) или точно (анг.true).

&& ⟶ Ова е оператор за логичка И операција (анг. and). За да се добие резултат логичка единица, двете променливи треба да бидат единици, односно точни (вистинити). Ако една од променливите е нула, тогаш и резултатот е нула.

|| ⟶ Оператор за логичка ИЛИ операција (анг. or). За да се добие логичка единица како резултат, барем една од двете променливи треба да биде единица, односно точна (вистинита).

! ⟶ Оператор за негација. Оваа операција има една променлива. Со неа се менува состојбата. Ако променливата била нула, резултатот ќе биде еден и обратно.

Пример 4.12.

```
1  if (!x)  {        //  Програмскиот код  во големата заграда се извршува ако x е
                     //  логичка нула, бидејќи по негацијата ќе добиеме резултат
                     //  еднаков на еден.
2  ……..
:  }
```

Подолу се дадени **скратените искази** и нивните еквивалентни аритметички и логички инструкции.

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. Инструкции во програмскиот јазик C/C++ за микрокомпјутер Arduino Uno

Според функцијата, инструкциите ќе ги поделиме во неколку групи:

- Инструкции за работа со влезно-излезни пинови
- Инструкции за контрола на време
- Математички инструкции
- Бит и бајт инструкции
- Инструкции за сериска комуникација
- Инструкции за работа со библиотеки

4.7.1. Инструкции за работа со влезно-излезни пинови

Инструкциите за работа со влезно-излезните пинови служат за конфигурирање на пиновите (влез или излез) и за запишување или читање на нивните вредности. [6]

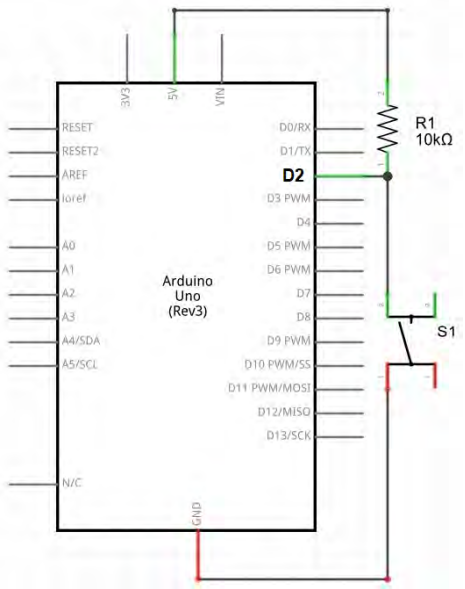
pinMode(pin,mode) ⟶ Под режим на дигитален пин подразбираме влезен или излезен режим на работа. Ако пинот е влезен тогаш Arduino прима податоци од електронската компонента, а ако е излезен тогаш Arduino испраќа податоци. За излез ознаката е OUTPUT, а за влез INPUT. Да споменеме дека постои и трет режим на работа со ознака INPUT_PULLUP и истиот ќе го објасниме малку подоцна.

digitalRead(pin) ⟶ Со оваа инструкција **се чита вредноста** на влезниот пин. Вредноста може да биде HIGH (високо ниво) или LOW (ниско ниво). Во средната заграда се запишува бројот на пинот или неговото симболично име, доколку претходно сме го декларирале пинот како целобројна променлива. Ако пинот е излезен, тогаш инструкцијата читање е невозможна.

Пример 4.13.

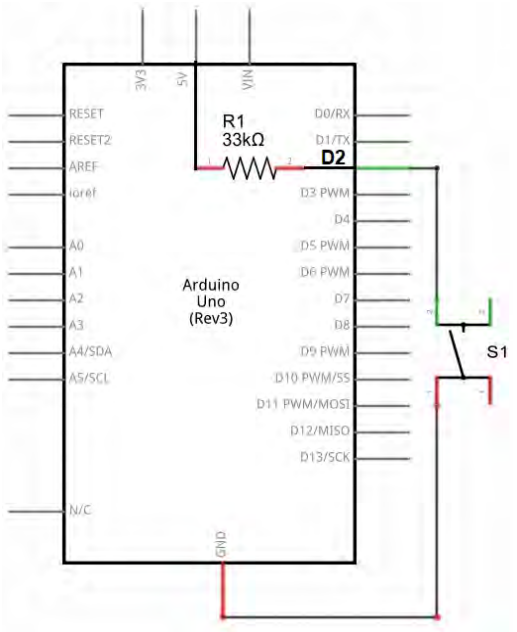
```
1  int taster=2;                    // taster е симболичното име на вториот пин.
2  int sostojba=0;                  // Декларираме променлива со симболично име
   /* ... */                        // sostojba и почетна вредност нула.
3  sostojba=digitalRead(taster);    // Читање на вредноста на вториот пин.
```

Функционалната шема прикажана на слика 4.13. се однесува на програмскиот код од пример 4.13. Како влезна компонента е употребен тастер кој е приклучен на вториот пин на Arduino Uno R3. Кога тастерот е притиснат, вториот пин се поврзува со заземјувањето и прочитаната вредност изнесува LOW. Кога тастерот не е притиснат вториот пин е поврзан со напојувањето и прочитаната вредност е HIGH. Ако отпорникот и тастерот во шемата на слика 4.13. си ги заменат местата тогаш кога тастерот ќе биде притиснат вториот пин ќе се поврзе со напојувањето и прочитаната вредност ќе биде HIGH односно логичка единица.



Слика 4.13. Поврзување на тастер со Arduino Uno R3

Доколку меѓу напојувањето од 5V и пинот D2 нема поврзан отпорник тогаш кога тастерот не е притиснат прочитаната вредност ќе биде случајна вредност односно постојано ќе се менува меѓу логичка нула и логичка единица. Во овој случај за да се постигне стабилна работа се користи режимот INPUT_PULLUP со што се активира внатрешниот отпорник за повлекување (анг. pull-up) кој е вграден во самата Arduino платформа и се наоѓа меѓу податочниот пин и пинот за напојување од 5V (слика 4.14.). На таков начин кога тастерот не е притиснат прочитаната вредност ќе биде логичка единица.

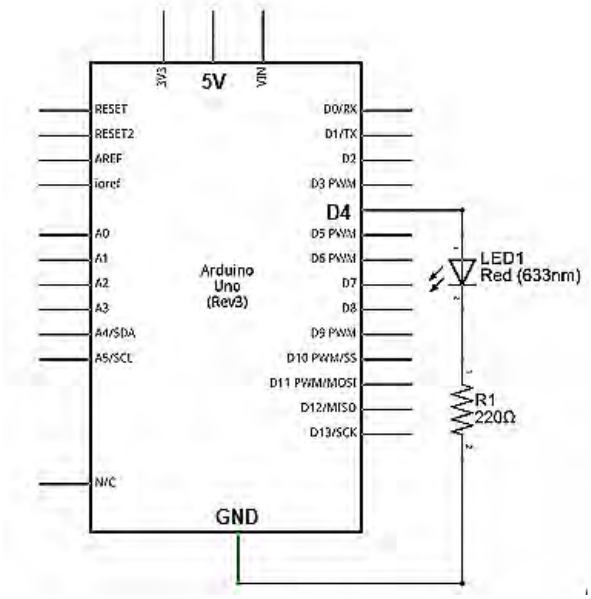


Слика 4.14. Вклучување на внатрешен pull-up отпорник во Arduino

digitalWrite(pin,vrednost) ⟶

Со оваа инструкција, дигиталниот излезен пин се поставува на ниско или високо ниво. За разлика од инструкцијата digitalRead, инструкцијата digitalWrite содржи два параметри: симболично име на пинот (или неговиот број) и вредноста HIGH или LOW.

Функционалната шема прикажана на слика 4.15. се однесува на програмскиот код од пример 4.14. Како излезна компонента е употребена лед диода кој е приклучен на четвртиот дигитален пин на Arduino Uno R3. Кога четвртиот пин ќе се постави на високо ниво (HIGH) тогаш лед диодата ќе светне. Ако лед диодата ја поставиме обратно, со катодата свртена кон четвртиот пин тогаш таа ќе биде активна на ниско ниво (LOW).



Слика 4.15. Поврзување на лед-диода со Arduino Uno R3

Пример 4.14.

```
1  int ledDioda=4;                  //  ledDioda е симболично име на четвртиот пин.
                                    //
2  digitalWrite(ledDioda,HIGH);     //  Четвртиот пин се поставува на високо ниво и
                                    //  диодата свети.
```

Пример 4.15. претставува програма за примена на трите инструкции за работа со дигитални пинови. Само кога тастерот е притиснат лед-диодата свети.

Пример 4.15.

```
1  int ledPin = 13;                 //  Лед-диодата е поврзана на тринаесеттиот
                                    //  пин на Arduino Uno.
2  int inPin = 7;                   //  Тастерот е поврзан на седмиот пин.
3  int val = 0;                     //  Целобројната променлива val ја памети
                                    //  прочитаната вредност.
4  void setup() {                   //
5    pinMode(ledPin, OUTPUT);       //  Конфигурација на тринаесеттиот пин како
                                    //  излезен
6     pinMode(inPin, INPUT);        //  Конфигурација на седмиот пин како влезен
7  }                                //
8  void loop() {                    //
9    val = digitalRead(inPin);      //  Читање на влезниот пин.
10   digitalWrite(ledPin, val);     //  Пинот на којшто е поврзана лед-диодата
   }                                //  има иста вредност како и пинот поврзан со
                                    //  тастерот.
```

Во наставната единица 4.2. Составни делови на микрокомпјутер на плочка Arduino Uno се запознавме со неговите аналогни влезови и излези. Истакнавме дека за аналогните пинови од суштинско значење е ширинско импулсната модулација.

analogRead(pin) ⟶ Со оваа инструкција **се чита вредноста на еден аналоген влез**. Напонот на аналогниот влез може да има бесконечно многу различни вредности и се движи во опсегот од 0 до 5 V. Аналогно-дигиталниот конвертор, во составот на Arduino Uno R3, ги претвора аналогните вредности во цели броеви од 0 до 1023. Целиот број потоа се претставува како бинарен код од 10 битови. Ако вредноста 5 V се подели на 1024 дела, тогаш еден дел ќе одговара на напон од 4,9 mV. Значи сите вредности на напонот од 0 V до 4,9 mV ќе одговараат на бројот 0, од 4,9 mV до 9,8 mV на бројот 1, од 9,8 mV до 14,7 mV на бројот 2 и така

натаму, сè до напонот 5 V, кој ќе одговара на бројот 1023.

analogWrite(pin,value) ⟶ Со аналогните излези може да се контролира јачината на светлината на LED-диода или брзината на вртење на мотор**. На излезниот пин се запишува аналогна вредност и истата се претвора во низа од нули и единици.** Ширината односно времетрањето на импулсите и големината на излезниот напон зависат правопропорционално од запишаната целобројна вредност, која се движи во опсегот од 0 до 255. (слика 4.5.)

Во примерот 4.16. на излезниот пин број 9 е поврзана лед-диода чиј интензитет на светлина зависи од положбата на лизгачот на потенциометарот кој е поврзан на третиот пин од Arduino Uno R3. Во програмскиот ред број 9 од истиот пример, прочитаната вредност од потенциометарот се дели со четири, бидејќи излезниот сигнал има четирипати помал опсег на вредности од оној на влезниот.

Пример 4.16.

```
1  int ledPin = 9;                  //  Лед-диодата е поврзана со деветтиот пин.
2  int analogPin = 3;               //  Потенциометарот е поврзан со третиот
                                    //  пин.
3  int val = 0;                     //  val е променлива за чување на прочитаната
                                    //  вредност.
4  void setup () {
5    pinMode(ledPin, OUTPUT);       //  Конфигурација на излезен пин.
6    }
7  void loop() {
8     val = analogRead(analogPin);  //  Читање на вредноста на влезниот сигнал.
                                    //  Прочитаната вредност е во опсегот од 0 до
                                    //  1023.
9     analogWrite(ledPin, val / 4); //  Запишаната вредност е во опсегот од 0 до
                                    //  255.
10 }
```

tone(pin, frequency) ⟶

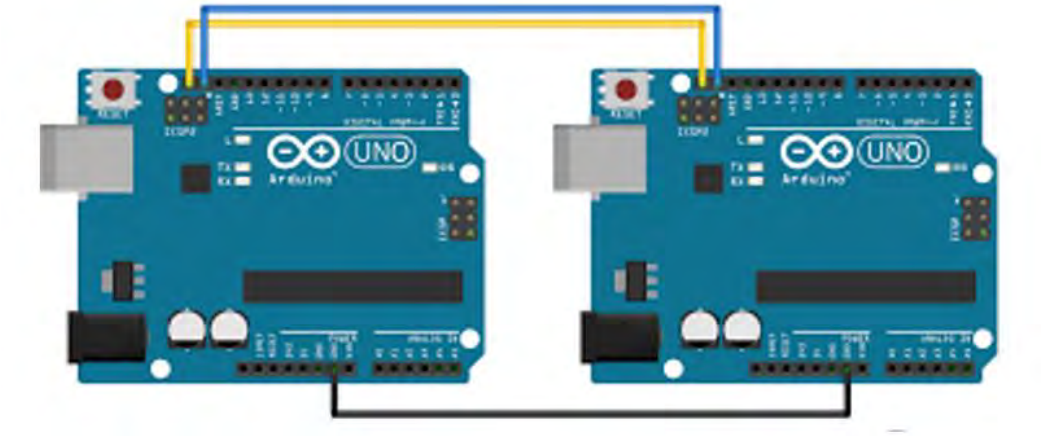
tone(pin, frequency,duration)

На излезниот пин 3 или 11 на Arduino Uno R3 може да се поврзе зујалица која ќе генерира едноличен тон. Фреквенцијата на тонот ја избира програмерот преку примена на инструкцијата tone (pin, frequency). Постои можност да се избере и времетраењето на побудниот сигнал. Во

спротивно, тој ќе трае сè додека не се изврши инструкцијата **notone()**.

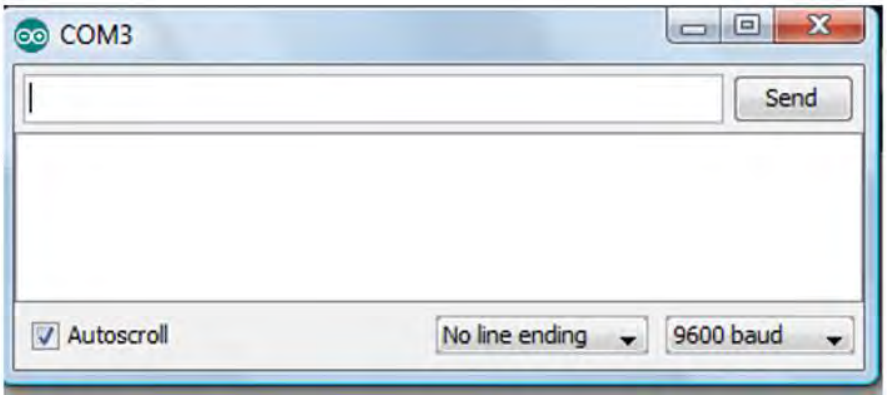
4.7.2. Инструкции за сериска комуникација

Пиновите RX (анг. Receive) и TX (анг. Transmit) на Arduino Uno R3 се пинови за сериска комуникација. На сликата 4.16. е прикажан начинот на поврзување на две Arduino Uno R3 платформи, при што RX-пинот на првата платформа е поврзан со TX-пинот на втората платформа и обратно.



Слика 4.16. Сериска врска меѓу две Arduino Uno платформи

Преку овие пинови, Arduino може да се поврзе со микрокомпјутер Raspberry Pi, Wi-Fi-модул, Bluetooth-модул и со други уреди што поддржуваат сериски пренос. Голема предност на серискатa комуникација е што размената на податоци може да се следи преку вградениот сериски монитор. Сите податоци што Arduino Uno R3 ги прима или ги испраќа со други уреди можеме да ги видиме на екран. Се разбира, за ова е потребно Arduino да биде поврзан со компјутер преку USB-кабел и да биде отворен серискиот монитор во рамките на интегрираната развојна средина. Серискиот монитор е прикажан на слика 4.17.



Слика 4.17. Сериски монитор на развојна средина за Arduino микрокомпјутер

Постојат дваесетина инструкции за работа со серискиот монитор, но ние ќе спомнеме само две.

Serial.begin(brzina) ⟶ Со оваа инструкција се поставува пропусниот опсег односно брзината на сериски пренос и таа е **наведена во долниот десен агол на серискиот монитор**. Стандардна брзина е 9600 boud што значи 9600 битови во секунди, но корисникот може да ја промени брзината на сериски пренос доколку тоа го наложува уредот поврзан на пиновите RX и TX.

Serial.print (x)
Serial.println (x) ⟶ На екранот од серискиот монитор **се прикажува вредноста на променливата**. После инструкцијата Serial.println (x) се отвара нова линија во прозорецот на мониторот, а со Serial.print (x) податоците се прикажуваат во истата линија.

Во пример 4.17. серискиот монитор го прикажува изминатото време, изразено во милисекунди.

Пример 4.17.

```
1  unsigned long vreme;
2  void setup() {
3    Serial.begin(9600);
4  }
5  void loop() {
6    Serial.print("Izminato vreme: ");
7    vreme = millis();
8    Serial.println(vreme);          //  Пауза од една секунда со цел да се
9    delay(1000);                    //  намали бројот на податоци прикажани
10 }                                 //  на серискиот монитор.
```

Во пример 4.18. Arduino Uno R3 ја мери вредноста на аналогниот напон на пинот A0, а серискиот монитор ја покажува истата. Да споменеме дека максималниот напон за влезно-излезните пинови на Arduino Uno R3 изнесува 5V и не е дозволено да се надмине оваа вредност. Во примерот е употребена инструкцијата analogRead. Бидејќи аналогно-дигиталниот конвертор ги претвора аналогните вредности во цели броеви од 0 до 1023, за да се добие вредноста на измерениот напон се применува формулата.

$$\mathrm{U[V]} = \frac{\text{Целобројна прочитана вредност}}{1023} \cdot 5\mathrm{[V]}$$

Пример 4.18.

```
1  const float maxVolti = 5.0;          //  Максимален напон за пинот A0
4  void setup() {
5    Serial.begin(9600);                //  Брзина на сериски пренос
```

```
7   }
8   void loop() {
9     int vrednost = analogRead(A0);              //  Читање на влезниот пин.
10    float napon = (vrednost /1023) * maxVolti;  //  Пресметка на измерен напон
      Serial.println(napon);                      //  Прикажување на резултат
11  }
```

4.7.3. Инструкции за контрола на време

Програмските кодови се извршуваат со многу голема брзина. Ако на излезот поврземе лед-диода и таа треба да трепка, тогаш меѓу двете нејзини состојби (анг. ON/OFF) мора да се воведе време на доцнење (анг. delay). Во спротивно, корисникот нема да забележи дека диодата наизменично се вклучува и се исклучува.

delay (ms) ⟶ **Времето на доцнење** е дадено во милисекунди. Во една секунда има 1 000 милисекунди или 1000000 микросекунди.

Пример 4.19.

```
1   int ledPin = 13;                //  Лед-диодата е поврзана со 13-от пин.
2   void setup() {
3     pinMode(ledPin, OUTPUT);      //  Пинот на лед-диодата се конфигурира како
4   }                               //  излезен.
5   void loop() {
6     digitalWrite(ledPin, HIGH);   //  Лед-диодата се вклучува.
7     delay(1000);                  //  Пауза од една секунда.
8     digitalWrite(ledPin, LOW);    //  Лед-диодата се исклучува.
9     delay(1000);                  //  Пауза од една секунда.
10  }
```

time=micros()
time=millis() ⟶ Со овие две инструкции се мери времето изразено во микросекунди и милисекунди од моментот кога Arduino Uno ќе почне да ја извршува програмата. За прикажување на изминатото време, потребно е да се вклучи серискиот монитор.

4.7.4. Математички инструкции

max(x,y) ⟶ Инструкцијата го избира поголемиот број од двата броја x и y. Во пример 4.21, инструкцијата не дозволува вредноста на сензорот да биде помала од 20.

min(x,y) ⟶ Инструкцијата го избира помалиот број од двата броја x и y. Во пример 4.20. инструкцијата не дозволува вредноста на сензорот да биде поголема од 100.

constraim(x,a,b) ⟶ Инструкцијата го ограничува опсегот на променливата x; a е минималната вредност, a b е максималната вредност.

Пример 4.20.

```
1  sensVal = min(sensVal, 100);  // Ако променливата sensVal има вредност
                                 // помала од сто тогаш таа ја задржува својата
                                 // вредност. Ако вредноста е поголема од сто
                                 // тогаш вредноста на променливата sensVal се
                                 // изедначува со вредноста 100.
```

Пример 4.21.

```
1  sensVal=max(sensVal, 20);  // Ако променливата sensVal има вредност
                              // поголема од 20 тогаш таа ја задржува својата
                              // вредност. Ако вредноста е помала од 20 тогаш
                              // вредноста на променливата sensVal се
                              // изедначува со вредноста 20.
```

map(vrednost, fromLow,fromHihg,toLow, toHigh) ⟶ Инструкцијата **го менува опсегот на вредности** на променливата. fromLow и fromHihg се минималната и максималната вредност на стариот опсег, а toLow и toHigh се минималната и максималната вредност на новиот опсег.

Во пример 4.22. се менува опсегот на вредности на аналогниот сигнал. Во почетокот, кодирањето на влезните примероци се врши со 10 битови и опсегот е од 0 до 1023, но потоа го менуваме опсегот од 0 до 255, за кодирањето да биде со 8 битови.

Пример 4.22.

```
1  void setup(){
2  }
3  void loop() {
4    int val = analogRead(0);          // Се чита аналогната вредност од нултиот пин
                                       // и се сочувува во променливата val.
5    val=map(val, 0, 1023, 0, 255);    // Кога ќе се промени опсегот, се менува и
                                       // променливата. Бидејќи опсегот се намалил,
                                       // треба да очекуваме и вредноста сразмерно
                                       // да се намали.
6    analogWrite(9, val);              // Новодобиената вредност се запишува на
                                       // излезниот пин број 9.
7  }
```

4.7.5. Бит и бајт инструкции

bitClear (x,n) → Инструкцијата го **ресетира (поставува на нула) битот** со реден број n во променливата x. Ако целобројната променлива е во декаден броен систем, треба да се изврши претворање во бинарен броен систем. Битот со најмало значење (првиот од десно) е на нулта позиција. Резултат од оваа инструкција ќе биде променливата x со ресетиран n-ти бит.

bitSet (x,n) → Инструкцијата го сетира (поставува на високо ниво) битот со реден број n во променливата x.

bitRead (x,n) → Се чита вредноста на n-от бит од променливата x. Резултатот ќе биде еден бит 0 или 1, во зависност од прочитаната вредност.

bitWrite (x,n,b,) → Оваа инструкција има три параметри. X е целобројна променлива, n е реден број на битот и b е вредноста на битот што треба да се запише. Во пример 4.23., со инструкцијата bitWrite (x,0,1) се менува состојбата на битот со најмала тежина од нула на еден.

bit (n) → Со оваа инструкција се пресметува тежината на битот во зависност од неговиот реден број, односно неговата позиција во бинарниот број. Тежината на битот се пресметува според изразот 2^n. Така, почнувајќи од десно кон лево, првиот бит има тежина 2^0=1, вториот 2^1=2, третиот 2^2=4, четвртиот 2^3=8 итн.

Пример 4.23.

```
1  void setup() {
2    Serial.begin(9600);      //  Нагодување на пропусен опсег на сериски монитор.
3    byte x = 0b10000000;     //  Префиксот 0b се користи за бинарни променливи.
4    Serial.println(x, BIN);  //  Серискиот монитор го прикажува бинарниот број
                              //  10000000.
5    bitWrite(x, 0, 1);       //  Запишување 1 на местото од најнезначајниот бит.
6    Serial.println(x, BIN);  //  Серискиот монитор го прикажува бинарниот број
                              //  10000001.
7  }
8  void loop(){
9  }
```

highByte (x) ⟶ Ако податокот е 16-битен, со оваа инструкција се издвојуваат најзначајните осум битови.

lowByte (x) ⟶ Ако податокот е 16-битен, со оваа инструкција се издвојуваат најмалку значајните осум битови.

Пример 4.24.

```
1  int test = 0xABCD;     //  Декларираме 16-битна променлива во
                          //  хексадекаден броен систем.
2  byte hi, lo;           //  Декларираме две бајт променливи.
3  hi = HighByte(test);   //  Добиваме резултат 0xAB.
4  lo = LowByte(test);    //  Добиваме резултат 0xCD.
```

4.8. Структури во програмскиот јазик C/C++ за Arduino микрокомпјутер на плочка

Структурите се блокови од инструкции со точно дефинирана функција. Блоковите од инструкции се напишани помеѓу големи загради { }. Ваквата поделба на програмата на структури и употребата на големите загради многу ја олеснува анализата на програмите.

4.8.1. Структури за избор на можности

Структурите за избор на можности се составени од исказите: if, else if и else.

```
if (услов) {
  инструкција 1
  инструкција 2
  инструкција 3
  .....
}
```

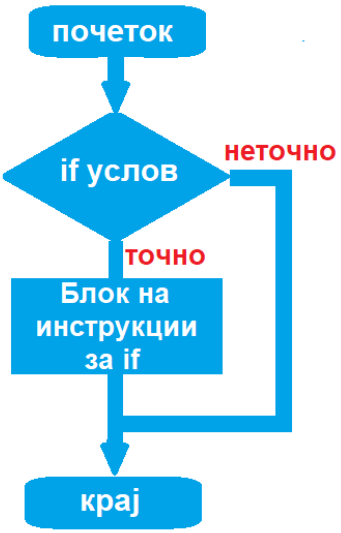
⟶ **Исказот If го проверува условот и ако тој е исполнет (вистинит) инструкциите се извршуваат**. Условот може да биде некој логички израз или израз со оператори за споредба. Синтаксата е таква што условот е запишан во мала заграда, а блокот инструкции во голема заграда. На слика 4.18. е прикажан блок дијаграмот на оваа структура.

```
if (услов1) {
 //блок
 //инструкции 1
}
else if (услов2) {
  //блок
 //инструкции 2
}
else {
```

⟶ Структурата If…else овозможува подобра контрола на кодот бидејќи врши **проверка на повеќекратни услови**. Доколку условот под исказот if не биде исполнет, се проверува условот под исказот else if. Бројот на else if искази може да биде неограничен. Секој невистинит услов предизвикува нова проверка, сè додека не се добие вистинит услов. Кога ќе се добие вистинит услов, се извршуваат неговите инструкции и се прекинува проверката на следните услови. Доколку не се добие вистинит услов, се извршуваат

```
  //блок
  //инструкции 3
}
```

инструкциите под исказот else. Исказот else е последен исказ во структурата за избор на можности и може да се забележи дека под исказот else нема услов за проверка. На слика 4.19. е прикажан блок дијаграмот на if…else структурата и за разлика од if структурата постои блок инструкции кои се извршуваат и кога if условот не е исполнет.



Слика 4.18. Блок-дијаграм на if структура

Слика 4.19. Блок-дијаграм на if…else структура

Пример 4.25. е пример за примена на структура за избор на три можности: температура поголема од 70 степени, температура меѓу 60 и 70 степени и температура помала од 60 степени.

Пример 4.25.

```
1  if (temperature >= 70) {
2    // Опасност! Исклучи го системот.
3  }
4  else if (temperature >= 60 && temperature < 70) {
5    // Внимание! Провери ги вредностите на другите параметри.
6  }
7  else {
8     // температурата е пониска од 60 степени.
9    //  Системот работи безбедно.
10 }
```

Во пример 4.26., кога тастерот не е притиснат лед-диодата свети, а кога истиот ќе се притисне диодата престанува да свети. Претпоставуваме дека кога тастерот е притиснат неговиот пин е на високо ниво и дека диодата е со активна анода.

Пример 4.26.

```
1  int val = digitalRead(taster);     //  Проверка на состојба на тастер.
2    if (val == LOW) {                //  Ја испитуваме вистинитоста на условот
3      digitalWrite(led, HIGH);       //  Пинот на диодата се поставува на високо ниво.
4    }
```

Во пример 4.27. како влезни компоненти се употребени два тастери. Во условот на if структурата е употребена логичката инструкција И и за да се изврши if структурата пиновите на двата тастери треба да бидат поставени на високо ниво.

Пример 4.27.

```
1  if (tasterA == HIGH && tasterB ==HIGH) {
   // Проверка на состојба на тастери и испитување на вистинитост на услов.
2     digitalWrite(led, HIGH);        //  Лед-диодата свети.
3    }                                //
4    else {                           //  Програмскиот код под else структурата се
                                      //  извршува ако не е исполнет условот под if
                                      //  структурата.
5     digitalWrite(led, LOW);         //  Лед-диодата не свети.
6  }
```

Структури за избор на можности може да се креираат и преку исказите **switch…case**.

```
switch (promenliva)
 {
  case 1:
   // блок инструкции
   break;
  case 2:
   // блок инструкции
   break;
  default:
   // блок инструкции
   break;
 }
```

Исто како исказот if, исказот switch…case го контролира текот на програмата, дозволувајќи да се креираат различни блокови од инструкции за различни случаи (анг. case). Веднаш до исказот switch, во мала заграда е наведено името на променливата од чија вредност зависи на кој случај ќе се префрли (анг. switch) извршувањето на програмата. Англискиот збор default во превод значи „стандардно“ и кодот под исказот default се извршува ако не се изврши ниеден од горните случаи. Исказот break се користи заедно со исказот switch и означува крај за секој случај. Без исказот break, структурата ќе продолжи да се извршува сè до нејзиниот крај.

Во пример 4.28. како влезна компонента е употребен сензор, фотоотпорник кој е поврзан со аналогниот влез A0. Да се потсетиме, аналогните сигнали од сензорите се влезни сигнали за аналогно-дигиталниот конвертор кој на својот излез генерира целобројни вредност во опсегот од 0 до 1023. Под претпоставка дека фотоотпорникот ќе го користиме во затворена просторија максималната вредност е намалена на 600. Во овој пример структурата switch заменува три if структури. Во програмата се чита вредноста на сензорот, потоа со употреба на инструкцијата map() се менува бројот на можни вредности на

вкупно три и на крај се прикажува една од трите пораки на серискиот монитор, во зависност од добиената вредност у.

Пример 4.28.

```
1   const int sensorMin=0;                    //  Се дефинираат максималната и
2   const int sensorMax=600;                  //  минималната вредност на сензорот.
                                              //
3   void setup() {
4      Serial.begin(9600);                    //  Се нагодува пропусниот опсег
5   }                                         //  на серискиот монитор.
6   void loop() {
7     int x=analogRead(A0);                   //  Се чита вредноста на сензорот
8     int y=map(x, sensorMin , sensorMax, 0,2);
    //Се врши намалување на опсегот на вредности на сензорот. Бројот на
    //можни вредности се намалува од 0-600 на 0-2.
9     switch(y) {                             //  Започнува switch структурата.
10    case 0:
11      Serial.println(“temnica”);
12      break;
13    case 1:
14      Serial.println(“polutemnica”);
15      break;
16    case 2:
17      Serial.println(“svetlina”);
18      break;
19     }
20     delay(10);                             //  За постабилна работа се внесува
21  }                                         //  доцнење од 10 милисекунди меѓу
                                              //  секои две читање на сензорот.
```

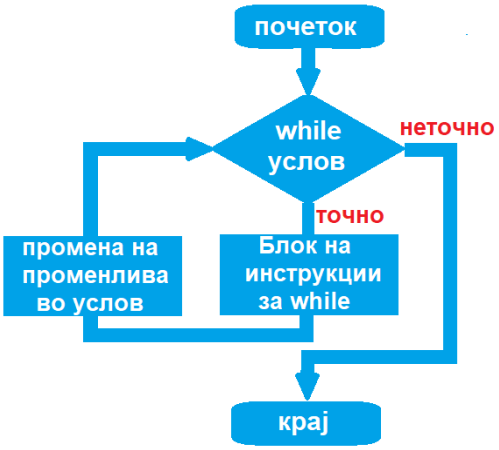
4.8.2. Структури за повторување (циклични структури)

Структурите за повторување на циклус се составени од исказите: while, do…while и for. Заедно со овие искази може да се употреби и исказот continue.

```
while (услов) {
  инструкција 1
  инструкција 2
  инструкција 3
  ...
}
```

→ Англискиот збор while во превод значи „сè додека“ и се мисли на вистинитоста на условот во малата заграда. Сè додека условот е вистинит, циклусот ќе се повторува, односно блокот инструкции под исказот while повеќекратно ќе се извршува. Треба да се внимава циклусот while да не биде бесконечен. Ова ќе се случи ако променливата во условот не се менува со инструкциите во големата заграда на структурата. За таа цел,

променливата во условот треба да се менува преку намалување или зголемување или да е зависна од состојбата на некој сензор.



Слика 4.20. Блок-дијаграм на while структура

Слика 4.21. Блок-дијаграм на do...while структура

Во пример 4.29. диодата трепка сè додека вредноста која ја измерил сензорот е поголема од 100.

Пример 4.29.

```
    // Најнапред се врши аналогно-дигитално претворање, а потоа се испитува
     // вистинитоста на условот во циклусот while.
1   while(analogRead(sensor) > 100) {
2     digitalWrite(LED, HIGH);          //  Лед-диодата свети
3     delay(100);                       //  Се чека 100 ms.
4     digitalWrite(LED, LOW);           //  Лед-диодата не свети.
5     delay(100);                       //  Се чека 100 ms.
6   }
```

```
do{
  инструкција 1
  инструкција 2
  инструкција 3
  ....
} while (услов)
```

→ Разликата меѓу исказите do…while и while е во тоа што кај исказот do…while вистинитоста на условот се проверува на крајот од циклусот. Ова значи дека циклусот мора еднаш да се изврши, без оглед на вистинитоста на условот. Слика 4.20. и слика 4.21. претставуваат споредба меѓу двете структури while и do…while.

Во пример 4.30. повторно како влезна компонента се користи сензор поврзан на аналогниот пин A3. Циклусот do…while сè повторува сè додека на излез од аналогно-дигиталниот конвертор не се добие вредност поголема од 100.

Пример 4.30.

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

```
2  int Sensor=3;
3  do {
4  delay(50);                    //  На сензорот му се дава потребното време за да
                                 //  изврши детекција.
5    x = analogRead(Sensor);     //  Читање на вредноста на сензорот.
6  } while (x < 100);            //  Испитување на вистинитост на услов.
```

for (декларација на променлива; услов; чекор со кој се врши промената)
{
инструкција 1
инструкција 2
инструкција 3
.....
}

→ Исказот for се користи за **повеќекратно повторување на блок инструкции напишани во големата заграда.** Најчесто се користи бројач чија вредност постојано се зголемува и кога ќе го достигне максимумот, престанува извршувањето на циклусот. Честопати структурата за повторување со исказ for се користи за исто конфигурирање на повеќе пинови или за обработка на повеќе податоци на идентичен начин.

Слика 4.22. Блок-дијаграм на for структура

Во пример 4.31. лед диодата е поврзана на еден од аналогните излези кои даваат ширински модулирани импулси. Како се зголемува излезната вредност, од 0 до 255, така се зголемува времетраењето на импулсите и средната вредност на излезниот напон, а со тоа и интензитетот на светлината.

Пример 4.31.

```
1  int PWMpin = 10;              //  Лед-диодата е поврзана на десеттиот пин.
2  void loop() {                 //
3     for (int i = 0; i <= 255; i++) {  //  Декларација на променлива за for
                                 //  структурата, услов за нејзино извршување и
                                 //  инструкција за чекорот со кој се врши
                                 //  промената на променливата.
```

```
4     analogWrite(PWMpin, i);     //  Вредноста на променливата i се запишува на
                                  //  излезниот пин.
5     delay(10);
6      }
7  }
```

continue ⟶ Со исказот continue можат условно да се прескокнат неколку циклуси во структурите for, while или do…while.

Пример 4.32. е пример за примена на исказот continue. Во овој пример се прескокнуваат вредностите од 41 до 119, споредбено со пример 4.31.

Пример 4.32.

```
1  for (int x = 0; x <= 255; x ++) {
2    if (x > 40 && x < 120) {              //  Прескокнување на вредности.
3       continue;
4     }
5    analogWrite(PWMpin, x);
6    delay(50);
7  }
```

Пример 4.33. е комбинација од две структури, for и if. Циклусот for може да се повтори четири пати, но услов е вториот дигитален пин да биде на ниско ниво.

Пример 4.33.

```
1  for (int i=0; i < 4;)              //  Декларација на променлива и услов за
                                      //  повторување на циклусот
2  {                                  //  Почеток на for структура.
3     if(digitalRead(2) == LOW) {     //  Услов и почеток на if структура.
4       i++;
5     }                               //  Крај на if структура.
6  }                                  //  Крај на for структура.
```

4.8.3. Структури за разгранување

Сè досега, програмите се извршуваа линеарно, односно инструкциите беа наредени последователно една врз друга. Со структурите за разгранување се менува текот на програмата, може да се прескокнуваат делови од програмскиот код и на таков начин да се оди напред-назад низ програмата.

goto oznaka; ⟶ .....
oznaka:

Структурите за разгранување се реализираат преку исказот goto, што во превод значи „оди на“. По исказот goto следува ознака (анг. label) којашто, всушност, претставува симболично име на местото каде што треба да се скокне.

Во пример 4.34. се употребени повеќе структури: три for структури, една if структура и една goto структура. Ова е пример за вгнездување на една структура во друга. Ако е исполнет условот на if структурата тогаш ќе се изврши структурата за разгранување, ќе се прескокне блокот на инструкции број 1 и директно ќе се изврши блокот на инструкции број 2. Ако условот на if структурата не е исполнет тогаш прво се извршува блокот инструкции број 1 и тоа повеќекратно. Колку пати ќе се изврши блокот на инструкции број 1 зависи од параметрите во for структурите за повторување и е еднакво на r×g×b пати.

Пример 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;
.        }
.        // блок инструкции број 1
.      }
.    }
.  }
.  bailout:
.  // блок инструкции број 2
```

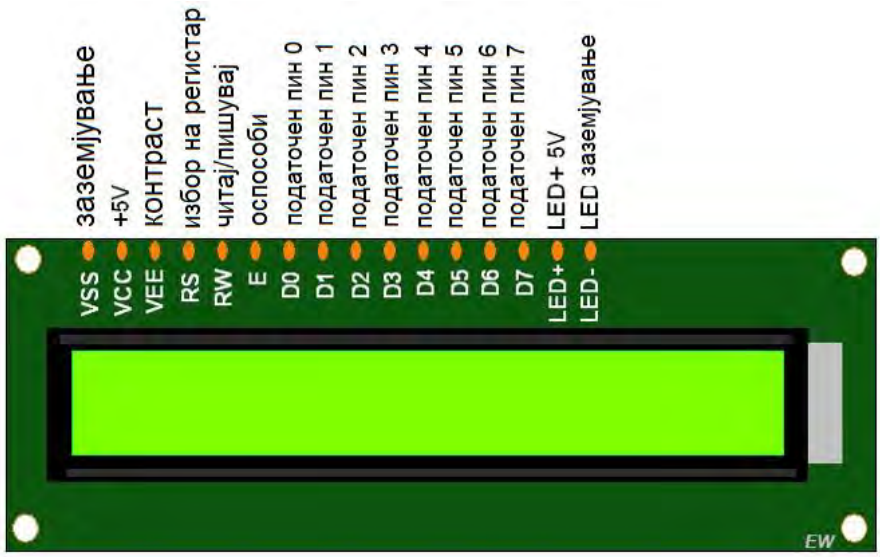
4.9. Инструкции за работа со библиотеки

Како и повеќето програмски платформи, така и развојната средина на Arduino може да се прошири преку употреба на библиотеки. **Библиотеките се потпрограми што овозможуваат комуникација меѓу периферните уреди и Arduino микрокомпјутерот**. Тие содржат инструкции со кои се намалува пишувањето на долги програмски кодови и значително се олеснува работата. Според функционалноста, библиотеките можат да се поделат во неколку групи: комуникација (пр. Wi-Fi, Ethernet, GSM), обработка на податоци, чување на податоци (пр.Cloud Data, работа со SD-картички), дисплеи (пр. LCD), управување со уреди (пр. мотори), сензори (пр. температурни, оптички, UV, притисок), тајминг итн. Arduino развојната средина содржи неколку стандардни библиотеки и нив можеме да ги употребиме со притискање (анг. click) на **Sketch→ Import Library** во самото мени. Доколку развојна средина не ја содржи бараната библиотеката тогаш таа може да се преземеме од интернет.

#include < ime > ⟶ Со оваа инструкција се повикува библиотеката во главната програма, при што треба да се наведе името на библиотеката.

Ние ќе се запознаеме со следниве библиотеки: LiquidCrystal, Servo и CapacitiveSensor .

LCD-екраните (анг. Liquid crystal display) се излезни компоненти и служат за визуелно прикажување на резултатите добиени од извршувањето на програмата. На слика 4.23. е прикажан изгледот на алфанумерички LCD-екран со две редици со по 16 симболи. На истата слика е прикажан и неговиот пин-дијаграм.



Слика 4.23. Надворешен изглед на LCD-екран и пин-дијаграм

Библиотеката LiquidCrystal е стандардна библиотека во состав на развојната средина и преку неа Arduino управува со LCD-екранот. Оваа библиотека содржи 20 различни инструкции, но ние ќе објасниме само три, кои се најчесто користени.

LiquidCrystal (rs, rw, enable, d4, d5, d6, d7) ⟶ **Со оваа инструкција се дефинираат пиновите на** Arduino кои се користат за поврзување со истоимените изводи на LCD-екранот. Оваа инструкција следува по вклучувањето на библиотеката во програмата како што е наведено во пример 4.35.

lcd.begin(kol,red) ⟶ Со оваа инструкција започнува комуникацијата со екранот и се опишува бројот на колони и редици од симболи кои можат да се прикажат на екранот.

lcd.print (tekst) ⟶ Оваа инструкција има само еден параметар, а тоа е текстот што треба да се прикаже на LCD-екранот.

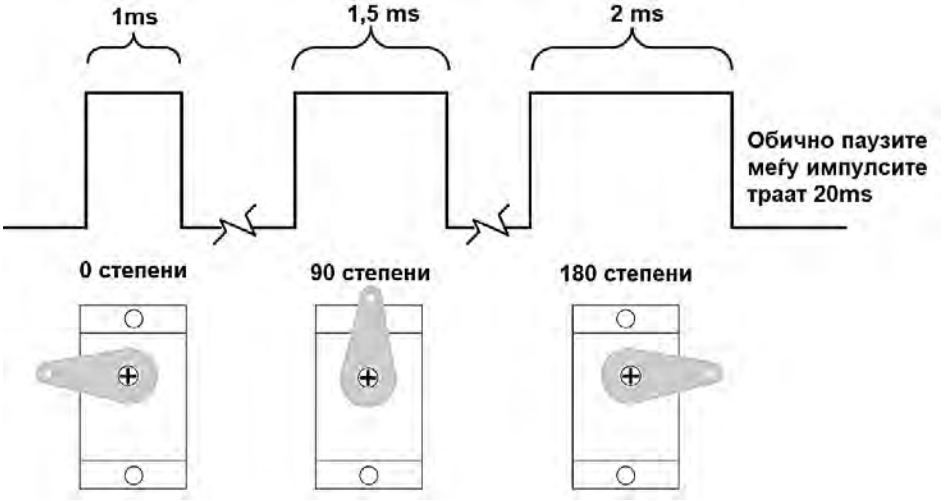
Пример 4.35.

```
1  #include <LiquidCrystal.h>          // Повикување на библиотека за работа
                                       // со LCD-екран.
```

```
2  LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);   //  Избор на пинови за поврзување на
                                                 //  Arduino Uno R3 со LCD-екранот.
                                                 //
3  void setup() {
4    lcd.begin(16,1);                            //  Екранот има 16 колони и една
                                                 //  редица.
5    lcd.print("hello, world!");
6  }
7  void loop() {}
```

Во практичниот дел на оваа модуларна единица предвидена е практична вежба со употреба на LCD-екран и тогаш подетално ќе се запознаеме со начинот на негово поврзување со Arduino Uno R3 и програмата за прикажување на текст на екранот.

Серво-моторот е посебен вид на мотор што не врти во круг, туку врши аголни поместувања и останува во одредена позиција до следната инструкција. Вообичаено, серво-моторот ротира за 180 степени. Серво-моторите многу често се користат во роботиката, на пример, за придвижување на роботска рака.



Слика 4.24. Зависност на аголот на вртење на серво моторот од ширината на импулсите

За да серво-моторот се заврти за определен агол, потребни му се ширински модулирани импулси, чија ширина варира меѓу една и две милисекунди. Поради чувствителноста на серво моторите наместо инструкцијата anologWrite и нејзините ширински модулирани импулси се користи Серво библиотеката и нејзините инструкции.

 # include (Servo.h) ⟶ Со оваа инструкција се вклучува библиотеката која овозможува Arduino Uno да управува со серво-моторот.

Servo ime ⟶ Креираме серво-објект на кој ќе се извршуваат функциите содржани во Серво библиотеката. Исто како

променливите, и на објектите им се задаваат симболични, уникатни имиња.

servo.attach (pin, min, max) ⟶ pin е бројот на пинот на Arduino микрокомпјутерот на кој е поврзан серво-моторот**. Само пиновите 9 и 10 имаат поддршка за поврзување на серво-мотор.** Min и max не се задолжителни параметри и претставуваат ширина на импулси, изразена во микросекунди, за минимален и максимален агол на завртување на серво-моторот. Колку е поголема ширината на импулсите, толку е поголем и аголот.

servo.write(agol) ⟶ Аголот на завртување на серво-моторот може да има вредност од 0 до 180 степени. Кога аголот е 0 тогаш поместувањето е најголемо во една насока, а за агол од 180˚ поместувањето е најголемо во друга насока. За 90 степени нема поместување.

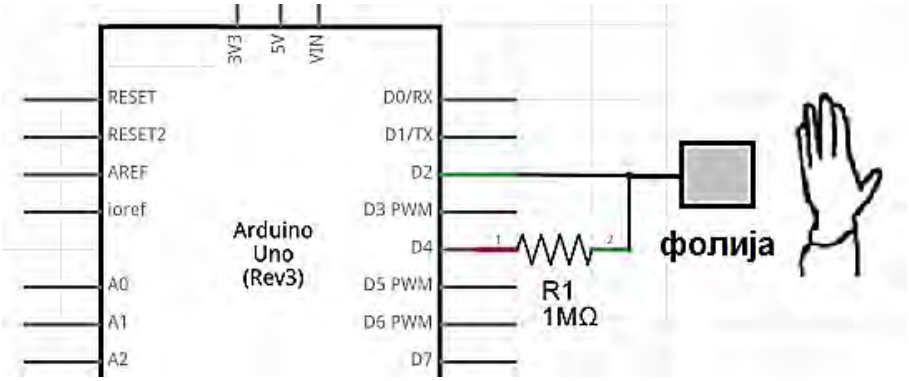
Во примерот 4.36. употребени се горните инструкции, при што за поврзување на серво-моторот е избран деветтиот пин и серво-моторот е во средината, на положба 90 степени.

Пример 4.36.

```
1  #include <Servo.h>
2  Servo myservo;          //  Креираме серво-објект со симболично име myservo
                           //
3  void setup(){
4    myservo.attach(9);
5    myservo.write(90);    //  Серво-моторот е позициониран точно на средината.
6  }                       //
7  void loop() {}
```

Во практичниот дел предвидена е практична вежба со употреба на серво мотор.

CapacitiveSensor библиотеката дозволува два или повеќе пинови на Arduino Uno R3 може да се претворат во капацитивен сензор за детектирање на капацитивноста на човечкото тело. За таа цел потребно е меѓу влезниот и излезниот пин да се поврзе отпорник со голема отпорност и парче алуминиумска фолија на крајот од влезниот пин. Ако до парчето фолија се доближи човек, неговото тело ќе апсорбира дел од електрицитетот, поради што ќе се промени состојбата на еден од пиновите. Arduino Uno го мери времето потребно за да се изедначи состојбата на двата пинови. Ако телото апсорбира поголемо количество електричество, тогаш и времето ќе биде подолго. [4]



Слика 4.25. Примена на CapacitiveSensor библиотеката

Ќе се запознаеме со три инструкции од библиотеката CapacitiveSensor.

 # include (CapacitiveSensor.h) → Со оваа инструкција се вклучува библиотеката која овозможува читање на вредноста на капацитивноста на сензорот.

CapacitiveSensor ime (pin sender, pin reciver) → Креираме објект со симболично име. Името на објектот се наведува пред секоја употребена инструкција од CapacitiveSensor библиотеката. Исто така, со оваа инструкција ги конфигурираме пиновите што ќе бидат поврзани со алуминиумска фолија. Помеѓу нив се поставува отпорник со отпорност од 1МΩ до 40МΩ. Чувствителноста на сензорот се зголемува со зголемување на отпорноста. Со отпорник од 1МΩ сензорот работи како сензор за допир, а со отпорник од 10МΩ како сензор за блискост, со максимално растојание до 15cm.

CapacitiveSensor(broj na primeroci) → Ова е инструкција за читање на вредноста на сензорот. Пред инструкцијата се наведува името на објектот, разделен со точка од инструкцијата. Во малата загради се пишува бројот на примероци кој вообичаено е 30.

За пример 4.37. употребен е капацитивен сензор, кој е поврзан на вториот и четвртиот пин на Arduino Uno R3. Ако ја допреме фолијата тогаш вредностите на екранот од серискиот монитор треба да бидат поголеми.

Пример 4.37.

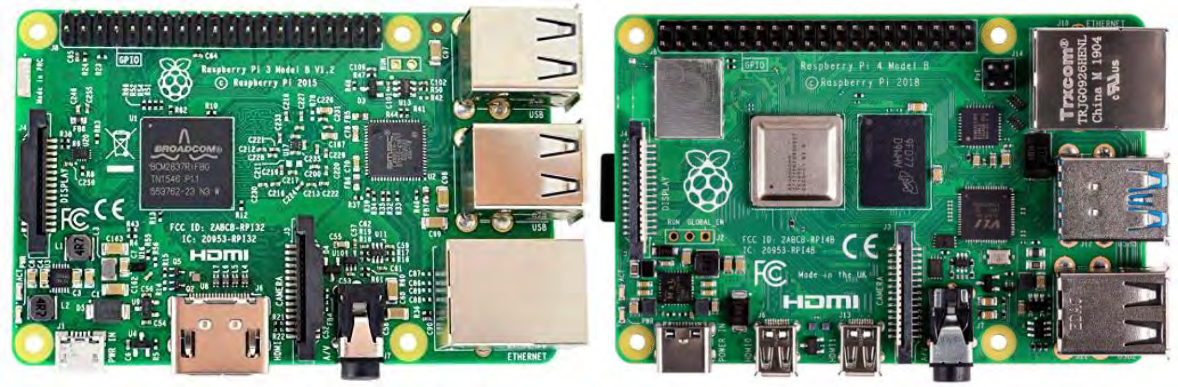
```
1  #include <CapacitiveSensor.h>      // Повикување на библиотека за работа со
                                      // капацитивен сензор.
2  CapacitiveSensor senDopir = CapacitiveSensor(4,2);
```

```
3  //Креирање на објект со симболично има senDopir. Избор на пинови за
   //поврзување на Arduino Uno со капацитивниот сензор.
4  void setup() {                    // Избираме пропусен опсег за сериски
5     Serial.begin(9600);            // монитор.
6  }
7  void loop() {
8  long sensorValue = senDopir.capacitiveSensor(30);
9  // Читање на вредност на сензор.
10 Serial.println(sensorValue);       // Прикажување на вредност на сензор на
11 }                                  // серискиот монитор.
```

4.10. Raspberry Pi микрокомпјутер на плочка

Raspberry Pi се најголемите конкуренти на Arduino. Првата генерација на Raspberry Pi излезе на пазарот во 2012 година, во Англија. Доказ за неговата популарност е тоа што во првите шест месеци беа продадени околу половина милион примероци. Слично како Adruino, Raspberry Pi беше наменет за студентите по компјутерски науки на Универзитетот во Кембриџ, со цел програмирањето да биде полесно. Денес Raspberry Pi има големо научно, образовно и практично значење.

Иако се конкуренти, Raspberry Pi многу се разликува од Arduino. По своите карактеристики Raspberry Pi е многу сличен со персонален компјутер и може да се поврзе со монитор, тастатура и глувче. Raspberry Pi располага со функционален **оперативен систем** (најчесто Linux), поради што може да извршува комплексни програми. Недостаток е тоа што софтверот за Raspberry Pi не е од отворен тип, како што е случај со Arduino. Развојната средина на Raspberry Pi не содржи бесплатни готови програми и библиотеки со драјвери, туку за тоа се потребни посебни лиценци. Arduino многу лесно се поврзува со сензори и со извршни уреди, а пак за Raspberry Pi ефикасно да се поврзе со некој сензор, потребен е дополнителен софтвер. Од друга страна, сите модели на Raspberry Pi поседуваат **графичка картичка** којашто го прави многу лесен за поврзување со видео-уреди. Најважна компонента во сите модели е Broadcom SoC (анг. System on a Chip) чипот со ARM процесор и интегрирана графичка картичка. Интересно е тоа што Raspberry Pi нема интерна трајна меморија. Најчесто се користи екстерна SD-картичка за чување на оперативниот систем и како програмска меморија. Преку 40-пинскиот конектор (анг. GPIO-General Purpose Input Output) можат да се поврзат до 27 различни уреди. Raspberry Pi поседува **мрежни конектори** и можност за поврзување преку Wi-Fi, што дополнително го олеснува пристапот до интернет-мрежата. Некои модели содржат и интегриран Bluetooth, како уште еден начин за безжична комуникација.



Raspberry Pi 3 Модел B Raspberry Pi 4 Модел B

Слика 4.26. Надворешен изглед на два модели на Raspberry Pi

Raspberry Pi може да се програмира во речиси сите програмски јазици. Најчесто користен е програмскиот јазик Python, но може да се користи и кој било друг програмски јазик како што е C/C++, Java Script, Scratch. [7] Досега се произведени **четири генерации** на Raspberry Pi и во секоја генерација има по неколку модели. Првата генерација од 2012 година е со два модела, А и В. Моделот В има подобра конфигурација, а моделот А има помала меморија и помал број на приклучоци, но и помала потрошувачка на енергија. Raspberry Pi 2, модел В излезе на пазарот во 2015 година, а следната година се појави и третата генерација. Во 2019 излезе последна генерација, Raspberry Pi 4. Во табелата 4.1, дадени се спецификациите за секоја генерација.

	Raspberry Pi 4 Модел В	**Raspberry Pi 3 Модел В**	**Raspberry Pi 2 Модел В**	**Raspberry Pi Модел В**
Процесор	QUAD Core со 1.5 GHz	QUAD Core со 1.2 GHz	QUAD Core со 900MHz	Едно јадро со 700MHz
Графичка картичка	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Трајна меморија	MicroSD	MicroSD	MicroSD	MicroSD
RAM	До 4GB SDRAM со 400 MHz	1GB SDRAM @ 400 MHz	1GB SDRAM со 400 MHz	512MB SDRAM со 400 MHz
Работна фреквенција	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 порти	4 x USB порти	4 x USB порти	4 x USB порти
Мрежен приклучок	RJ45	RJ45	RJ45	RJ45
Wi-Fi	Вграден	Вграден	Нема	Нема
Bluetooth	Вграден	Вграден	Нема	Нема
Влезови и излези со општа намена	40-пински конектор	40-пински конектор	40-пински конектор	40-пински конектор
Напојување	5 V	5 V	5 V	5 V

Табела 4.1. Споредба на карактеристики на различни модели на Raspberry Pi

. Од табелата може да се заклучи дека Raspberry Pi според своите можности е многу поблиску до персонален компјутер отколку Ардуиното.

Ние ќе се запознаеме со начинот на инсталирање на оперативниот систем Raspbian, со основните наредби и функции на програмскиот јазик Python и ќе реализираме неколку проекти што ги користат извонредните интерфејс перформанси на Raspberry Pi.

4.11. Составни делови на Raspberry Pi микрокомпјутер на плочка

Raspberry Pi е извонреден микрокомпјутер на плочка со големина на кредитна картичка и многу ниска цена на чинење. Со него можеме да пребаруваме на интернет, играме видеоигри, пишуваме компјутерски програми, креираме електрични кола и др. Ние ќе се запознаеме со моделот Raspberry Pi 3 B+. Raspberry Pi 4 модел B има многу поголем капацитет на RAM меморијата (1GB, 2GB или 4GB) и три до четири пати поголема брзина на работа во однос на неговиот претходник. Овој модел е идеален доколку сакаме Raspberry Pi да го користиме како домашен персонален компјутер или за обработка на веб-слики (анг. Computer Vision). За разлика од другите модели Raspberry Pi 4 модел B има проблеми со загревањето, па дури се препорачува и употреба на ладилник. Доколку намената на Raspberry Pi се проекти од областа на електрониката или автоматизација на нашите домови тогаш се препорачува употреба на Raspberry Pi 2 или 3 модел B. [8] Да нагласиме дека сите модели на Raspberry Pi се компатибилни, односно софтверот напишан за еден модел може да го извршува кој било друг модел.

На сликата 4.27. е прикажана третата ревизија на моделот Raspberry Pi 3 B+. Како и секој друг компјутер, така и Raspberry Pi е составен од повеќе компоненти, од кои најважен е системскиот чип Broadcom BCM2837 SoC (System on Chip).

Слика 4.27.Составни делови на Raspberry Pi 3B+ (поглед од горе)

Тој во себе содржи: 64-битен ARM Cortex-A53 четиријадрен процесор со работна фреквенција 1,4 GHz, 32 KB L1 кеш-меморија, 512 KB L2 кеш-меморија и графички процесор VideoCore IV. RAM-меморијата не е интегрирана на ист чип со процесорот, како што тоа беше случај со Arduino Uno R3. Таа е посебен чип на развојната плоча, со максимален капацитет од 1GB. RAM-меморијата не можеме да ја видиме на долната слика бидејќи се наоѓа на задната страна. [7]

Под металното капаче со изгравираното лого на Raspberry Pi се наоѓа радиопредавател. Тој се користи за вмрежување на Raspberry Pi во локални компјутерски мрежи и во интернет-мрежата преку Wi-Fi или за поврзување со други паметни уреди (сензори, мобилни телефони) преку Bluetooth. Мрежниот и USB-контролерот е чип одговорен за пренос преку **мрежната порта и четирите USB-приклучоци**. Во непосредна близина на мини USB-приклучокот се наоѓа еден помал чип којшто се грижи за напојувањето на сите компоненти на развојната плоча.

Најголема предност на Raspberry Pi во однос на Arduino е неговата поврзливост, односно големиот број на приклучоци. Да се потсетиме дека Arduino Uno R3 имаше само еден мини USB-приклучок. Raspberry Pi има четири USB 2.0 и едно микро USB-приклучоци. На слика 4.27. се гледаат само два USB 2.0 приклучоци, но всушност се четири, од кои два се поставени еден врз друг. Овие приклучоци се користат за поврзување со голем број надворешни уреди: тастатури, компјутерски глувчиња, дигитални камери, екстерни хард-дискови и други мемории. Микро USB-приклучокот се користи за напојување, исто како што се напојуваат мобилните телефони. Овој приклучок може да се користи и за поврзување со компјутер за пренос на податоци.

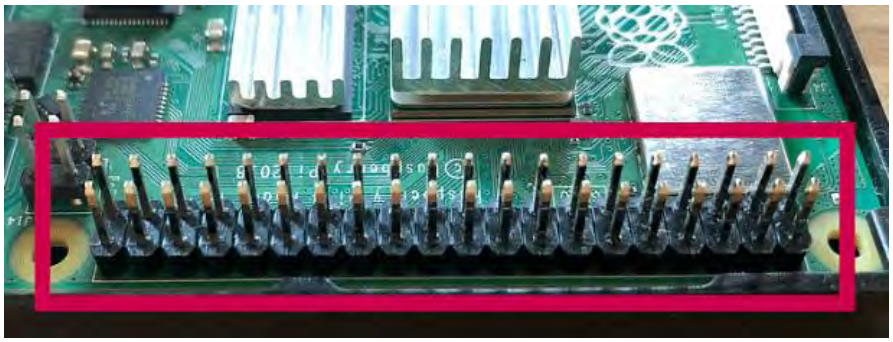
Освен преку Wi-Fi, пребарувањето на интернет може да се врши и преку мрежната порта. За таа цел ни е потребен кабел со RJ45-приклучок. Другиот крај на кабелот се поврзува со рутер. Во долниот дел на мрежната порта има две лед-диоди што служат како покажувачи за успешна комуникација.

Како што кажува и самиот назив **HDMI** (анг. High Definition Multimedia Interface), овој приклучок нуди најквалитетен аудио и видеопренос. Преку него, Raspberry Pi се поврзува со монитор или со телевизор. 3,5-милиметарскиот AV (анг. Audio Video) приклучок пред сè е наменет за поврзување со слушалки или со аудиозасилувач. Тој може да се користи и како видеоприклучок, но тогаш треба да се набави посебен адаптер.

На самата развојна плоча постојат два **специјални конектори за** поврзување специјално дизајнирани Raspberry Pi модули. Едниот е познат под кратенката CSI (анг. Camera Serial Interface) и служи за поврзување на камера, а вториот DSI (анг. Display Serial Interface) за поврзување на сензорен екран (анг. touch screen).

Raspberry Pi располага со приклучок со **40 пинови за поврзување со влезно-излезни уреди со општа намена.** Кратенката за овој приклучок е GPIO (анг. General Purpose Input Output). Дваесет и шест пинови од овој конектор се за

поврзување со влезно-излезни уреди, а останатите се за заземјување и напојување, од 3,3 V и 5 V.



Слика 4.28. GPIO-пинови за поврзување на влезно-излезни уреди

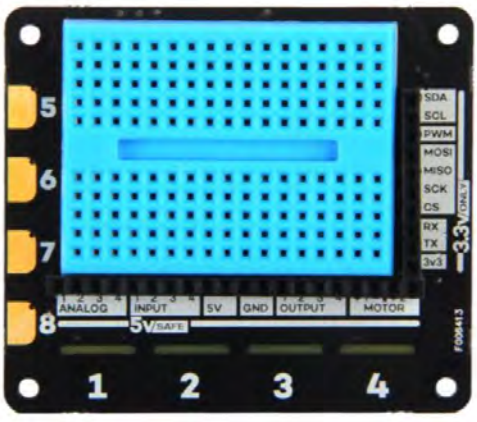
Со функцијата и значењето на влезно-излезните елементи се запознавме кога говоревме за составните делови на Arduino Uno R3. Се разбира, протоплочката е наједноставно решение за поврзување на овие уреди.

4.12. Додатоци за Raspberry Pi

Додатоците за проширување на функционалноста на Raspberry Pi се познати под името хардвер прикачен одозгора (анг. HAT-Hardware attached on top). Најчесто за поврзување се користат сите 40 пинови за општа намена на Raspberry Pi, но сепак повеќето додатоци дозволуваат пиновите да ги користат и други уреди, па дури е дозволено поврзување на повеќе додатоци еден врз друг. Најголем број додатоци се поврзуваат по принципот поврзи и работи (анг. Plug and play) односно автоматски се конфигурираат. [7]

На слика 4.29. е прикажан еден од најпопуларните додатоци, додатокот со сензори и дисплеј (анг. Sense hat). Тој во својот состав содржи 8x8 LED матричен дисплеј, мал џојстик со пет копчиња и шест интегрирани сензори: жироскоп

Слика 4.29. Додатокот за Raspberry Pi со сензори и дисплеј



Слика 4.30. Додатокот за истражување на Raspberry Pi

(сензор за аголна брзина), акцелерометар (линеарен сензор за забрзување), магнетометар (преку мерење на земјиното магнетно поле, магнетометарот го одредува правецот на географскиот север), барометар (сензор за притисок), температурен сензор и сензор за релативна влажност.

Додатокот за истражување, прикажан на слика 4.30. се користи во роботиката пред сè поради можноста за контрола на мотори на еднонасочна струја. Овој додаток е компатибилен и со Arduino микроконтролерските платформи. Додатокот за истражување содржи четири влезови и излези со максимална вредност на напонот од 5V, четири капацитивни копчиња на допир, четири копчиња за поврзување на штипки крокодилки, четири аналогни влезови, два H-мостови за контрола на вртење на мотори на еднонасочна струја и мини протоплочка која се поставува одозгора.

Raspberry Pi може да се користи и како мерен инструмент благодарение на додатокот за мерење и собирање на податоци (анг. DAQ MCC 128 - Data Acquisition Measurement Computing).

Слика 4.31. Додаток за мерење и собирање на податоци

Оваа електронска плочка содржи осум аналогни влезови за мерење на напон. Постојат два начини за мерење на влезниот напон. Кај мерењето со еден крај (анг. Single ended) се мери влезниот напон во однос на нултата потенцијална точка,заземјувањето.

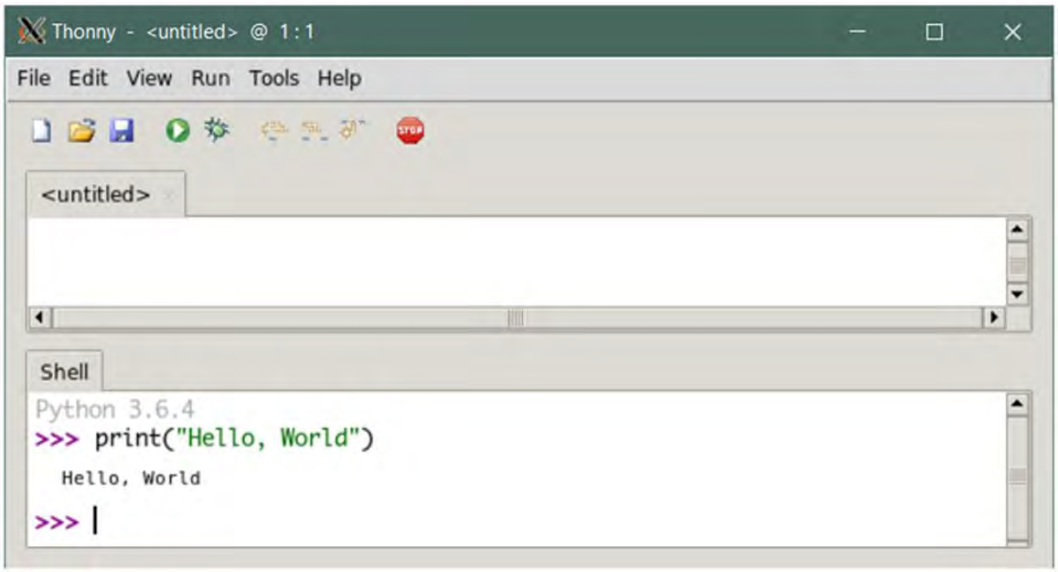
Диференцијалното мерење ја дава разликата меѓу два влезни напони. Во една секунда се земаат 100К примероци. Аналогниот дигитален претворувач има 16 битна резолуција.

Може да се наредат осум додатоци за мерење на напон еден врз друг и во тој случај се добива 64 податочни канали со максимален пропусен опсег 320К примероци во една секунда.

4.13. Програмирање на микрокомпјутер Raspberry Pi со програмски јазик Python

За разлика од Arduino, Raspberry Pi е микрокомпјутер со оперативен систем. Оперативниот систем дозволува инсталација и менаџирање на најразлични апликации, што го прави Raspberry Pi софтверски многу моќен. За Raspberry Pi постојат повеќе видови оперативни системи, но за почеток ќе го користиме **оперативниот систем Raspbian**. Во практичниот дел на оваа модуларна единица ќе се запознавме со инсталацијата на оперативниот систем

Raspbian и со неговата работна површина. Во менито, или поточно во категоријата Programming, се наоѓа интегрираната развојната средина Thonny Python IDE која служи за програмирање на Raspberry Pi во програмскиот јазик Python. На слика 4.32. е прикажан прозорецот после отворањето на развојната програма.



Слика 4.32. Развојна програма Thonny за програмирање на Raspberry Pi во програмски јазик Python

Исто како Arduino развојната средина, така и развојната програма Thonny Python IDE содржи уредувач на текст и конзола за дебагирање. Во уредувачот на текст ја пишуваме програмата, а во конзолата за дебагирање го гледаме резултатот добиен со извршувањето на програмата. Со притискање на копчето New започнуваме нова програма и по нејзиното креирање ја сочувуваме со притискање на копчето Save as.

Програмскиот јазик Python е еден од најпопуларните програмски јазици, особено меѓу младата популација. Причина за ова е неговата **лесна разбирливост**, што се должи на синтаксата која е многу слична со синтаксата на англискиот јазик. Познавањето на англискиот јазик е предуслов за брзо совладување на програмскиот јазик Python. Програмскиот кодот е многу пофлуентен. Во исказите не се користат кратенки за опишување на исказите, туку цели зборови. Исказите не завршуваат со знакот точка запирка како што тоа беше случај со програмскиот јазик C/C++. Програмскиот јазик Python е од повисок ред во однос на програмскиот јазик C/C++. Преведување на програмата напишана во програмскиот јазик Python го врши системска програма **интерпретер**, а не компајлер. Компајлерот го претвора програмскиот код во машински јазик одеднаш, додека интерпретерот ги преведува исказите еден по еден, додека се извршува програмата. Програмските кодови преведени со компајлер се извршуваат побрзо.

Слика 4.33. Карактеристики на програмскиот јазик Python

Програмскиот јазик Python е покомпатибилен со оперативниот систем Linux и инсталацијата во оперативниот систем Windows е малку посложена. Да се потсетиме, јадро (анг. kernel) на оперативниот систем Raspbian е Linux.

Пример 4.38. претставува **краток програмски код** за Raspberry Pi, на чиј излезен пин е поврзана диода.

Пример 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)
```

Програмските кодови во програмскиот јазик Python се пократки зашто **не постои декларирање на променливи.** Во програмскиот јазик C/C++ , за променливата x запишуваме int x=5;, а во програмскиот јазик Python само x=5. Исто така, нема ограничување во видот на податоци, односно ако делиме два цели броја, можеме веднаш да добиеме резултат децимален број, а не само цел број.

Кога пишуваме блок на инструкции во рамките на една структура, како што се if….else или while, не се користат големи загради, туку инструкциите се вовлекуваат за четири празни места навнатре. Откако ќе ја декларира структурата, програмерот треба да притисне две точки (:) и уредувачот на текст

самиот го вовлекува текстот. Коментарите од програмскиот код не се разделуваат со две коси црти, туку со знакот тараба „#“.

Во програмскиот јазик Python, **програмерот може да креира своја функција** преку инструкцијата def и потоа неа да ја повикува на различни места во програмата. Таков е програмскиот код прикажан во примерот 4.39..

Пример 4.39.

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

Програмскиот јазик C/C++ е подобар за програмирање вградливи системи, а програмскиот јазик Python е поуниверзален и со него можеме да креираме и графички апликации. C/C++ е хардверски, а Python е софтверски ориентиран. Програмскиот јазик Python располага со голем број библиотеки со најразлична намена. Секоја библиотека располага со огромно инструкциско множество. Библиотеката turtle се користи за графички дизајн (на пример, да нацртаме ѕвезда што ќе ротира). Библиотеката pygame се користи за креирање видеоигри со визуелни и звучни ефекти. OpenCV е библиотека за обработка на слики. Cloud4rpi е библиотека за складирање податоци во облак технологија (анг. cloud). Доволно е да истакнеме дека во моментот постојат 137 000 библиотеки за програмскиот јазик Python. Не е доволно само да се одбере соодветната библиотека, туку потребно е да се проучат и наредбите во нејзини рамки. Ние ќе ги проучиме можностите што ги нуди библиотеката GPIO Zero, која се користи кога Raspberry Pi се поврзува со влезно-излезни уреди (пр. диоди, тастери, сензори, мотори, дисплеи).

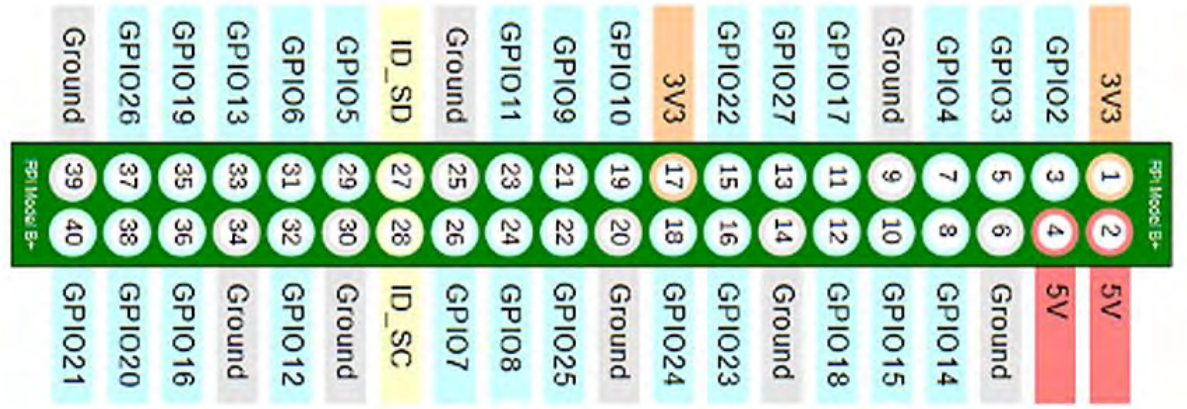
4.14. Поврзување на микрокомпјутер Raspberry Pi со влезно-излезни уреди

За поврзување на микрокомпјутерот Raspberry Pi со влезно-излезни уреди се користи **40-пински приклучок,** познат под кратенката GPIO (General Purpose Input Output). **26 пинови од овој приклучок се за поврзување со влезно-излезни уреди, а останатите се за заземјување и за напојување од 3,3V и 5V.**

За да може микрокомпјутерот Raspberry Pi да комуницира со влезно-излезните уреди поврзани на приклучокот GPIO, потребно е да се вклучи библиотека наречена GPIO Zero. Тоа е стандардна библиотека за оперативниот систем Raspbian и нема потреба од дополнителна инсталација. [9]

Броевите на пиновите според оваа библиотека се различни од физичките броеви на пиновите. На слика 4.34. е прикажано нумерирањето на

таканаречените GPIO-пинови. На пример, на пинот со физички број 15 одговара GPIO-пин со број 22.



Слика 4.34. Означување и функционален опис на GPIO пиновите

Исто како програмирањето на Arduino Uno R3 во C/C++, така при програмирањето на Raspberry Pi 3B+ во Python, на почетокот од програмата треба да се вклучат библиотеките што ќе се користат во неа. За оваа цел се користи наредбата import, која има слична функција како наредбата include при програмирање на Arduino Uno развојната платформа. Во пример 4.40., при конфигурирање на пинот за поврзување на тастер, додаден е префиксот gpiozero.

Пример 4.40.

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

Во програмскиот јазик Python, дозволено е наместо целата библиотека, да се вклучи една класа од неа. Во примерот 4.41. се вклучува само делот од библиотеката што ги содржи наредбите за управување со тастер. Во овој случај можеме директно да користиме тастери во програмата, без префиксот gpiozero.

Пример 4.41.

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

Постои можност наместо со поединечни тастери да работиме со цели тастатури. Во тој случај треба да се повика класа од библиотеката gpiozero позната под името ButtonBoard, што во превод значи плоча со тастери.

Пример 4.42.

```
1  from gpiozero import ButtonBoard
```

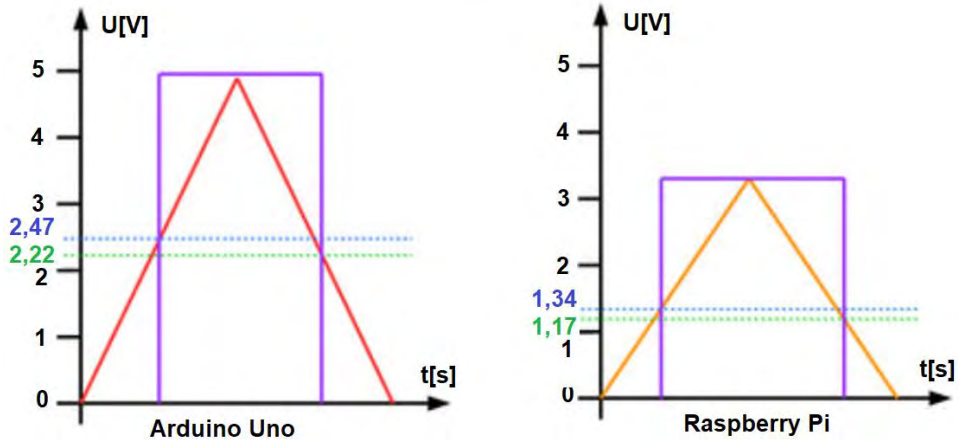
Веќе спомнавме дека **библиотеката GPIO е поделена на класи, во зависност од видот на влезно-излезните уред**и. Подолу се претставени класите и уредите на коишто тие се однесуваат.

Влезни уреди	→	Тастер, инфрацрвен сензор, сензор за движење, оптички сензор, сензор за растојание
Излезни уреди	→	Лед-диода (RGB или контролирана со импулсно-ширински импулси), зујалици, мотори, серво-мотори
Сериски периферни уреди	→	Аналогно-дигитален претворувач
Платформи и додатоци	→	LED-дисплеи, тастатури, роботи, контролери за домашна употреба преку далечинско управување
Внатрешни компоненти	→	Нагодување на време, температура на процесор, пресметка на средни вредности, искористеност на дискот и друго
Генератори на тонови	→	Генерирање музички тонови, прости или сложени

4.15. Класа на влезни уреди од библиотеката GPIOZERO

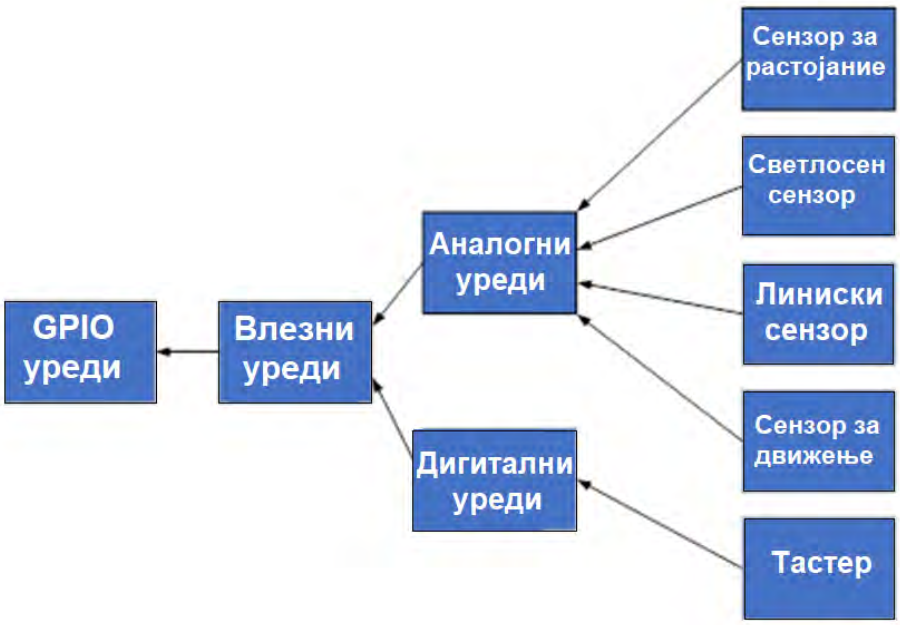
Дигиталните сигнали имаат само две вредности, еден или нула, LOW или HIGH. Аналогните сигнали можат да примаат најразлични вредности во опсегот од 0 до максимално дозволениот напон, вклучувајќи и децимални броеви. Raspberry Pi нема вграден аналогно-дигитален претворувач како Arduino платформата. Доколку сакаме да отчитуваме аналогни вредности од влезовите на Raspberry Pi тогаш треба да поврземе екстерен аналогно-дигитален претворувач како што е на пример интегрираното коло MCP3008. За работа со влезни аналогни уреди може да се искористи таканаречениот напон на праг (анг. threshold). Ако влезниот напон е поголем од напонот на праг тогаш таа вредност ќе се смета за логичка единица (HIGH), а ако влезниот напон е помал од напонот на праг тогаш таа вредност ќе се смета за логичка нула (LOW). На слика 4.35. споредбено се прикажани напоните на праг за Arduino Uno и Raspberry Pi. Напонот на влезните пинови на Raspberry Pi не смее да биде поголем од 3,3V. На истата слика може да видиме дека напонот на праг се разликува во зависност од тоа дали влезниот напон се намалува или зголемува. Кога влезниот напон се зголемува од нула кон максимално дозволената вредност напонот на праг изнесува 1.34V за Raspberry Pi и споредбено 2,47V за Arduino Uno. Кога влезниот напон се намалува од максимално дозволената вредност кон нула тогаш

напонот на праг изнесува 1.17V за Raspberry Pi и споредбено 2,22V за Arduino Uno.



Слика 4.35. Споредба на напони на праг за Arduino Uno и Raspberry Pi

На сликата 4.36. е прикажана поделбата на влезните уреди според видот на сигналите што ги генерираат.

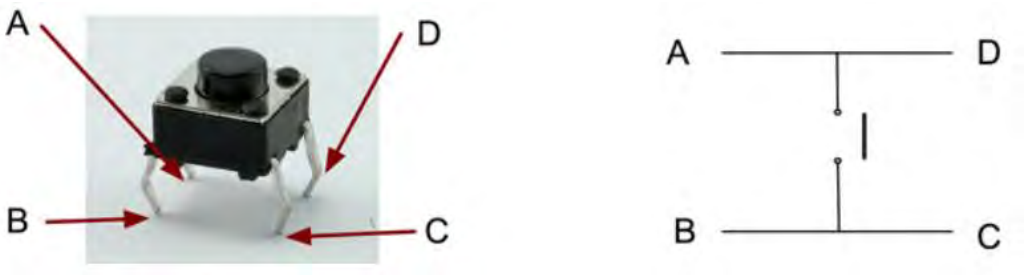


Слика 4.36. Влезни уреди за Raspberry Pi

Сите класи во составот на библиотеката GPIOZERO покажуваат вакво разгранување. Ваквиот пристап во голема мера го поедноставува програмирањето на Raspberry Pi.

4.15.1. Тастер (анг. Button)

Да се потсетиме дека приклучоците на тастерот кои лежат на иста линија, но спротивна страна се кусо поврзани. Ова е прикажано на слика 4.37.



Слика 4.37. Внатрешна поврзаност на приклучоците на тастер

Примерот 4.43. претставува програмски код за прикажување на една линија текст под услов тастерот да биде притиснат.

Пример 4.43.

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

Во овој програмски код, употребени се две инструкции за работа со тастер: Button() и wait_for_press () . Постојат уште 12 други инструкции за работа со тастер. Ќе ги објасниме поважните инструкции и нивните параметри, кои се пишуваат во малата заграда.

Button() ⟶ Во примерот 4.43, бројот 4 претставува **број на GPIO-пин.** Овој параметар е задолжителен. Во спротивно, конзолата за дебагирање ќе пријави грешка. Ќе го споменеме и параметарот pull_up. Овој параметар може да има две состојби, true или false. Стандардната вредност true значи високо ниво (HIGH) кога тастерот е притиснат. Во овој случај, едниот пин од тастерот се поврзува на заземјување, а другиот на GPIO-пинот. Ако параметарот pull_up е false, тогаш се добива ниско ниво (LOW) кога тастерот е притиснат и првиот пин наместо на маса се поврзува со напојувањето од 3,3 V.

wait_for_press () ⟶ Извршувањето на **програмскиот код се прекинува** додека не помине означеното време или тастерот не биде притиснат.

wait_for_release() ⟶ Извршувањето на програмскиот код се прекинува додека не помине означеното време или тастерот не биде ослободен.

held_time ⟶ Ако тастерот го држиме притиснат подолго време, оваа инструкција го прикажува времето на притискање изразено во секунди.

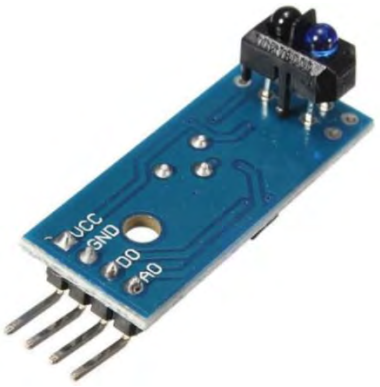
hold_time ⟶ Оваа инструкција определува **колку секунди да се чека** по притискањето на тастерот.

is_held ⟶ По истекот на времето определено со инструкцијата hold_time вредноста станува вистинита (true).

is_pressed ⟶ Резултатот од оваа инструкција е булова вредност, true или false. Вредноста е вистинита (true) само кога тастерот е притиснат.

4.15.2. Инфрацрвен рефлектирачки модул за следење (TRCT5000)

Инфрацрвениот рефлектирачки модул за следење е прикажан на слика 4.38. Тој може да се искористи за детекција на полна линија**.** Модулот има четири приклучоци, но за поврзување со Raspberry Pi користиме три: Vcc , GND и DO (анг.Data Out). Vcc-пинот се поврзува со напојувањето од 3,3 V, GND е заземјување. DO-пинот е пин за сигнализација и се поврзува со еден од GPIO-пиновите.



Слика 4.38. Инфрацрвен рефлектирачки модул за следење

Примерот 4.44. претставува програмски код за прикажување на текст, во зависност од вредноста што ќе ја прочита сензорот.

Пример 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) ⟶ Со оваа инструкција **го конфигурираме пинот** за поврзување на сензорот

when_line ⟶ Инструкцијата се извршува кога сензорот е активен.

when_no_line ⟶ Инструкцијата се извршува кога сензорот не е активен

lambda ⟶ Ова е стандардна Python инструкција којашто се користи како потинструкција на друга инструкција, за да означи некакво дејство. Не се користи како самостојна инструкција.

4.15.3. Сензор за растојание (HC-SR04)

Сензорот за растојание е ултразвучен сензор што испраќа насочен бран, кој се рефлектира од објектот поставен пред сензорот и се враќа до сензорот. Се мери времето од моментот кога предавателот ќе го испрати бранот до моментот кога приемникот ќе го прими рефлектираниот бран. Според овој параметар се пресметува растојанието помеѓу сензорот и објектот. **Испратениот бран се нарекува тригер (активатор), а рефлектираниот бран се вика ехо.**

Слика 4.39. Сензор за растојание (HC SR04)

Примерот 4.45. претставува програмски код за HC SR04 кој дава информација за растојанието изразено во cm. Пинот за тригерирање е поврзан со пинот GPIO17.

Пример 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() ⟶ **Задолжителни се два параметри. Прв е бројот на GPIO-пинот за ехото, а потоа за тригерот**. Тие се пишуваат во малата заграда.

wait_for_in_range ⟶ Извршувањето на програмскиот код се прекинува додека растојанието е над растојанието на праг (анг. threshold_distance) или не помине времето наведено во малата заграда. Растојанието на праг вообичаено изнесува 0.3m и како параметар може да се нагоди со инструкцијата DistanceSensor().

wait_for_out_of_range → Извршувањето на програмскиот код се прекинува додека растојанието е под растојанието на праг или не помине наведеното време.

distance → Резултат од оваа инструкција е растојанието што го измерил сензорот и ова растојание е дадено во метри.

when_in_range → По оваа инструкција, следува знакот еднакво и друга помошна инструкција која се извршува доколку условот е исполнет .

when_out_of_range → Инструкцијата дефинира дејство доколку растојанието е над растојанието на праг.

4.15.4. Оптички сензор или фотоотпорник (анг. LDR-Light Depended Resistor)

Фотоотпорникот е најчесто користен сензор за детекција на светлина. Неговата отпорност се менува од 1KΩ при светло и 100KΩ при мрак односно истата се намалува кога интензитетот на светлината се зголемува. Подоцна ќе реализираме практична вежба со негова примена, а сега ќе се запознаеме со неколку инструкции за работа со овој сензор.

Пример 4.46.

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

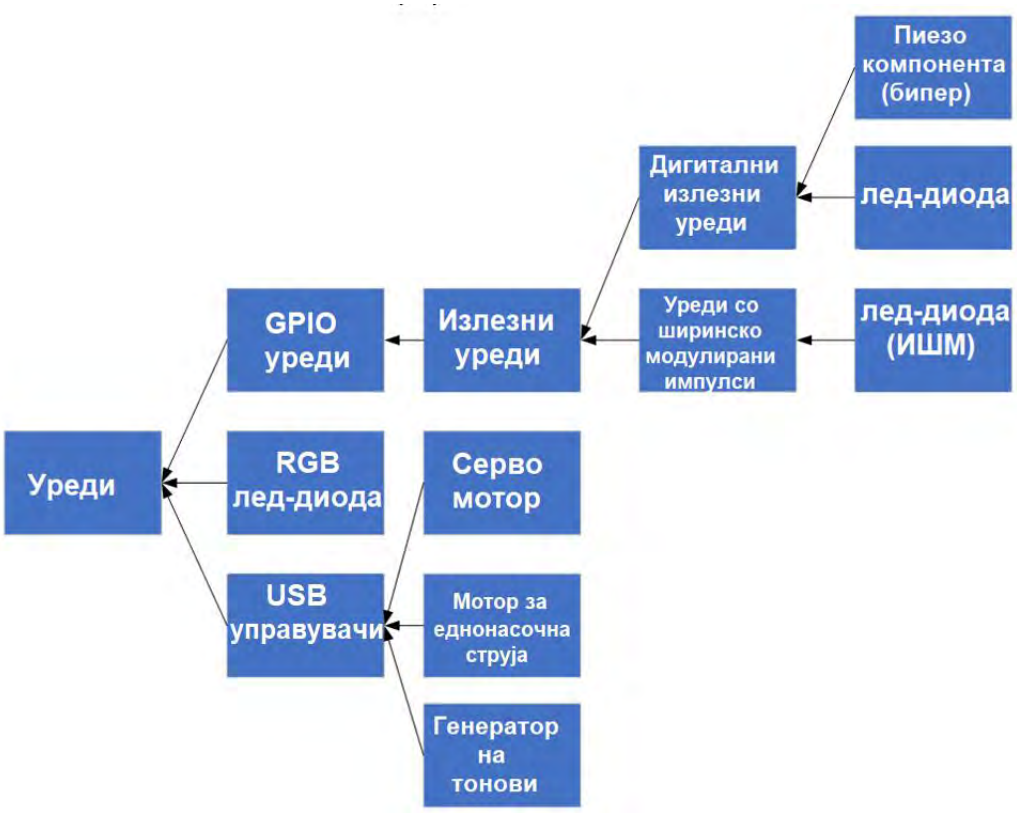
LightSensor( ) → Во малата заграда се наведува бројот на GPIO-пинот. Во примерот 4.46., ldr е симболично име на сензор и името го задава програмерот.

wait_for_light() → Со оваа инструкција, програмскиот код се стопира сè додека не се активира сензорот. Во малата заграда може да се наведе и времето што треба да помине за да сензорот се активира.

wait_for_dark () → Со оваа инструкција се стопира програмскиот код сè додека не се појави мрак.

4.16. Класи на излезни уреди од библиотеката GPIO Zero

На сликата 4.40. е прикажана поделбата на излезните уреди според видот на сигналите што ги генерираат.



Слика 4.40. Излезни уреди за Raspberry Pi

Стеблото со излезни уреди е разгрането уште повеќе. Според видот на приклучокот, извршена е уште една поделба, излезни уреди со GPIO-приклучок и уреди со USB-приклучок. Уредите со GPIO-приклучок се поделени на дигитални излезни уреди и уреди управувани со ширинско модулирани импулси (PWM).

За вклучување и исклучување на дигиталните уреди, потребни се две вредности, LOW или HIGH. Со ширинско модулираните импулси вршиме промена на интензитетот на светлина на лед-диодата. Исто како и кај влезните уреди, така и кај излезните уреди, библиотеката gpiozero содржи специјално множество од инструкции за секој уред посебно.

4.16.1. Лед-диода

При поврзувањето на LED-диодата со Raspberry Pi, треба да внимаваме **подолгиот пин (анодата) да се поврзе со GPIO-пинот, а пократкиот пин**

(катодата) со заземјувањето. Исто така треба да се употреби отпорник заради ограничување на струјата.
Програмскиот код во примерот 4.47. врши вклучување на LED-диодата.

Пример 4.47.

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

LED() ⟶ Во малата заграда се наведува бројот на GPIO-пинот, активната и почетната вредност. Ако активната вредност е HIGH, тогаш диодата ќе свети како во горниот пример. Ако active_high=false, тогаш за да светне лед-диодата ни треба инструкција off(). led е симболично име на диодата што го задава програмерот.

blink (on_time=1**,** off_time=1**)** ⟶ Со оваа инструкција, диодата почнува да трепка. Можеме да го менуваме **времето на вклучување и исклучување. Стандардно, ова време изнесува една секунда.**

off() ⟶ Оваа инструкција ја исклучува лед-диодата

on() ⟶ Оваа инструкција ја вклучува лед-диодата

toggle() ⟶ Оваа инструкција **ја менува состојбата на лед-диодата**, ако била вклучена ја исклучува и обратно.

4.16.2. Лед-диода контролирана со ширински модулирани импулси (PWMLED)

Доколку сакаме го менуваме интензитетот на светлина во програмата треба да се вклучи класата PWMLED од gpiozero библиотеката. Светлината е поголема кога импулсите од периодичната поворка имаат поголема ширина. Ширината на импулсите може да ја менуваме со помош на потенциометар.

Слика 4.41. Зависност на интензитетот на светлина од ширината на импулсите

Во инструкцијата PWMLED(), во малата заграда се наведува бројот на GPIO-пинот за поврзување, фреквенцијата на периодичната поворка и почетната вредност. Доколку не се наведе фреквенција тогаш ќе важи

стандардната фреквенција од 100 Hz. За управување со интензитетот на светлина се користат вредности од нула до еден.

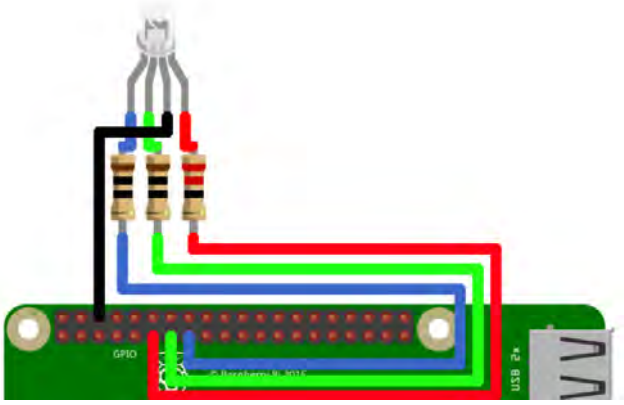
Пример 4.48.

```
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. Лед-диода со променлива боја (RGB-лед)

Кратенката RGB доаѓа од зборовите red, green, blue, што во превод значи црвена, зелена и сина. Со мешање на овие три основни бои се добиваат сите останати. Постојат 256 нијанси од секоја основна боја и со мешање на овие нијанси можат да се добијат 16 777 216 различни бои. Секоја основна боја претставува променлива чија вредност е децимален број со вредност од 0 до 1. **Останатите бои се добиваат како комбинација од основните.** На пример, комбинацијата (1, 0, 0) ја претставува црвената боја, комбинацијата (0, 1, 0) ја претставува зелената боја, комбинацијата (1, 1, 0) ќе биде жолта боја и комбинацијата (1, 0.5, 0) ќе биде портокалова боја. За полесна употреба на боите може да се користи библиотеката color, која е стандардна библиотека на програмскиот јазик Python.

RGB лед-диодата има четири приклучоци, по еден приклучок за секоја боја и еден заеднички приклучок кој е подолг. Во случај на RGB лед-диода со заедничка катода, таа се поврзува со заземјувањето, а анодите со три GPIO пинови, по еден за секоја од трите основни бои. Со програмскиот код во пример 4.49. лед-диодата ќе свети жолто. Пример 4.50. е програмски код за употреба на библиотеката color.



Слика 4.42. Поврзување на RGB-диода со Raspberry Pi

Пример 4.49.

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

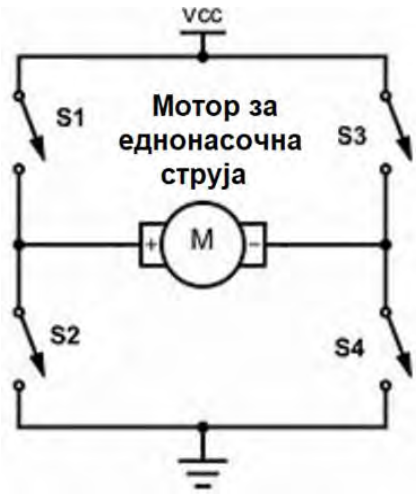
Пример 4.50.

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

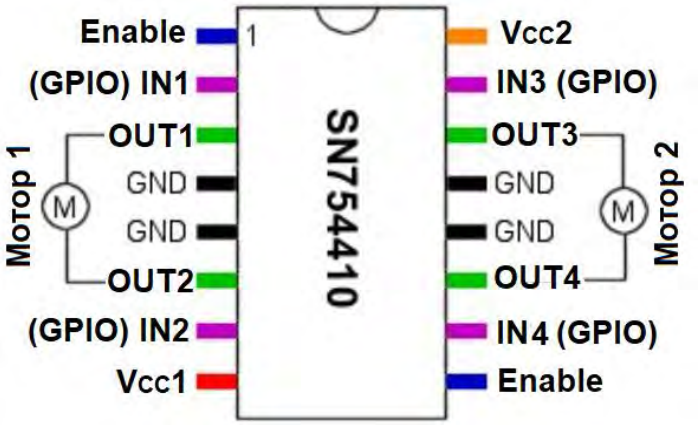
За RGB-диодата важат истите инструкции како и за обичната лед-диода, со тоа што треба да се конфигурираат три GPIO-пинови и да се дефинира бојата на лед-диодата со инструкцијата color(). Светлината на RGB-диодата може да се менува според интензитет ако во инструкција RGBLED() се вклучат ширински модулираните импулси со употреба на параметарот pwm=True.

4.16.4. Мотор на еднонасочна струја

Ако се промени насоката на струјата, ќе се промени и насоката на вртење на моторот на еднонасочна струја. Тој **се поврзува со Raspberry Pi преку H-мост.** Принципот на работа на H-мостот е прикажан на слика 4.43.



Слика 4.43. Принцип на работа на H-мост



Слика 4.44. Пин-дијаграм на H-мост

Тој е насочувач и може да го менува поларитетот на напонот. Ако прекинувачите S1 и S4 се затворени, а прекинувачите S2 и S3 се отворени тогаш моторот врти во насока на стрелката на часовникот. Ако S1 и S4 се отворени, а прекинувачите S2 и S3 се затворени тогаш моторот врти во обратна насока. За работа со Raspberry Pi како H-мост најчесто се користат интегрираните кола L293D или SN754410. На слика 4.44. е прикажан пин-дијаграмот на интегрираното коло SN754410 и истото може да се користи за управување со два мотори за еднонасочна струја. Ако со моторите се поврзат тркала тогаш овој склоп може да се искористи за движење на робот. Излезите на H-мостот се поврзуваат со моторот на еднонасочна струја, а влезовите со два GPIO-пинови.

Во практичната вежба 4.8.2.7. Промена на насока вртење на мотор на еднонасочна струја е употребено интегрираното коло SN754410 и во истата е објаснет начинот на поврзување на моторот, H-мостот, Raspberry Pi и протоплочката.

Подолу се дадени неколку инструкции за работа со DC-мотор.

Motor() ⟶ Во малата заграда се запишуваат броевите на GPIO-пиновите за поврзување на H-мостот.

backward*(speed=1)* ⟶ Моторот за еднонасочна струја се врти во насока спротивна од насоката на стрелката на часовникот**. Вредноста на брзината на вртење (speed) може да се менува во опсегот од 0 до 1,** под услов да се користат ширински модулирани импулси. Еден е максимална вредност на брзината на вртење.

forward*(speed=1)* ⟶ DC-моторот се врти во насока на стрелката на часовникот.

reverse() ⟶ Се менува насоката на вртење на моторот.

stop() ⟶ Моторот престанува да се врти.

Програмскиот код во примерот 4.51. го врти моторот нанапред, во насока на стрелките на часовникот.

Пример 4.51.:

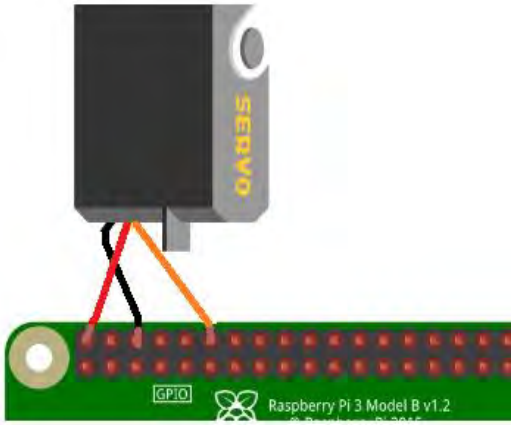
```
1  from gpiozero import Motor
2  motor = Motor(17, 18)
3  motor.forward()
```

4.16.5. Серво-мотор

Да се потсетиме, серво-моторот не се врти во круг, туку тој се поместува за одреден агол, кон лево или десно од својата средна позиција, во зависност од големината на управувачкиот сигнал. Начинот на работа на серво-моторот е прикажан на слика 4.24.

На слика 4.45. е прикажан надворешниот изглед на серво моторот, а на слика 4.46. е прикажан начинот на неговото поврзување со GPIO пиновите на Raspberry Pi. Серво-моторот има три приклучоци: црвен, црн и портокалов. Црвениот кабел се поврзува со напојувањето од 5 волти, а црниот со заземјување. Портокаловиот кабел се поврзува со еден од GPIO-пиновите и преку него Raspberry Pi комуницира со моторот, односно го контролира аголот на вртење.

Слика 4.45. Надворешен изглед на серво-мотор



Слика 4.46. Поврзување на серво=мотор со Raspberry Pi

Со програмскиот код во примерот 4.52, серво-моторот ќе се заврти за 45 степени.

Пример 4.52.

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

Ќе се задржиме само на инструкцијата servo.value(), со која се дефинира вредноста на аголот на вртење. Ова вредност може да се менува во опсегот меѓу -1 (минимален агол од 0 степени) и 1 (максимален агол од 180 степени). Ако вредноста изнесува нула, тогаш серво-моторот е во својата средна позиција односно 90 степени.

Заклучоци:

Микрокомпјутерот на плоча претставува електронска плочка во која се вградени: микроконтролер, сериски програматор, кристален осцилатор, порти, аналогни и дигитални влезови и излези. Таа преку USB-кабел може да се поврзе со компјутер и да се внесе готова програма во нејзината програмска меморија.

На влез од микрокомпјутерот на плочка можат да се поврзат: тастери, прекинувачи, готови тастатури, сензори (за температура, притисок, проток, движење и др.). Излезите можат да се поврзат со најразлични излезни уреди, како што се: лед-диоди, сијалици, бипери, мотори, дисплеи.

Постојат повеќе од 15 различни модели на Ардуино платформи, а моментално, најпопуларни се Arduino Uno, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due.

Микрокомпјутерот Arduino Uno располага со 14 дигитални пинови, шест аналогни влезови, три LED-диоди индикатори и една вградена LED.

Заради лична заштита и заштита на елементите од оштетување, пред почетокот на секоја монтажа треба да се исклучи изворот за напојување, без оглед дали се работи за батерија или компјутер.

Развојната програма за микрокомпјутерот Arduino Uno се презема од официјалната веб-страница на Arduino, поточно од следниов линк arduino.org/download. После преземањето на инсталацијата, истата ја инсталираме. Го притискаме (click) Run. Ако не започне инсталацијата, потребно е да се отвори програмата Control Panel, да се избере категоријата Device Manager, потоа поткатегоријата Other Devices или Unknown Devices и да се притисне (click) Update Drivers или Update Driver Software.

Основни градбени елементи на програмскиот јазик C/C++ се: променливите, инструкциите и структурите.

Синтакса во програмирање значи множество на правила за подредување на ознаките, симболичните имиња, операторите, интерпункциските знаци, коментарите, со цел да се добие исказ разбирлив за компјутерот.

Променливите ги чуваат податоците кои се обработуваат во компјутерот. Освен името и видот на податок, променливите имаат и свои вредности. Кога декларираме (најавуваме) променливи за микрокомпјутерот Arduino Uno, прво се пишува видот на променливата, па симболичното име, операторот за доделување и вредноста на променливата.

Основните типови на податоци се: цели броеви, децимални броеви, карактери, низи и логички податоци.

Операторите се знаци со кои програмерот, според точно определени правила, гради искази, инструкции. Постојат повеќе видови оператори: математички, логички, споредбени, оператори за доделување.

Инструкциите за влезно-излезни пинови служат за конфигурирање на пиновите (влез или излез) и за запишување или читање на нивните вредности. Во оваа група спаѓаат инструкциите: digitalRead(pin), digitalWrite(pin,vrednost), pinMode(pin,mode),analogRead(pin) и analogWrite(pin,value).

Аналогно-дигиталниот конвертор, во составот на микрокомпјутерот Arduino Uno,ги претвора аналогните вредности во цели броеви од 0 до 1023. Целиот број потоа се претставува како бинарен код од 10 битови.

Инструкцијата delay (ms) се користи за внесување време на доцнење изразено во милисекунди.

Инструкцијата map(vrednost, fromLow, fromHihg, toLow, toHigh) го менува опсегот на вредности на променливата. fromLow и fromHigh се минималната и максималната вредност на стариот опсег, а toLow и toHigh се минималната и максималната вредност на новиот опсег.

Инструкцијата bitClear (x, n) го ресетира (поставува на нула) битот со реден број n во променливата x. Инструкцијата bitSet (x,n) го сетира (поставува на високо ниво) битот со реден број n во променливата x.

Со инструкцијата Serial.begin(brzina) се поставува пропусниот опсег, односно брзината на сериски пренос на податоци чија единица мерка е број на битови во една секунда. Со инструкцијата Serial.print (x) на екранот од серискиот монитор се печати вредноста на променливата.

Структурите за избор на можности се составени од исказите: if, if...else и else. If исказот го проверува условот и ако истиот е исполнет (вистинит), тогаш ги извршува инструкциите.

Библиотеките се потпрограми кои овозможуваат комуникација меѓу периферните уреди и микрокомпјутерот Arduino Uno. Arduino развојната средина содржи неколку стандардни библиотеки и истите може да ги видиме со притискање (click) на **Sketch > Import Library** во самото мени.

Опсегот на вредности кои може да ги даде сензорот е многу поголема од опсегот на вредности кои може да се измерат во реалната околина. Поради тоа, треба да се нагодат максималната и минималната вредност на сензорот уште во првите неколку секунди по пуштањето во работа на микрокомпјутерот и сензорот. Оваа постапка се вика калибрација.

Составни делови на микрокомпјутерот Raspberry Pi се: системскиот чип Broadcom BCM2837 SoC (System on Chip), RAM-меморијата, радиопредавател, четирите USB-приклучоци, RJ45 приклучок, HDMI, специјални конектори за камера и LCD-дисплеј и 40-пинско подножје за влезно-излезни уреди.

Raspberry Pi нема вградена трајна меморија, туку за чување на податоците и оперативниот систем се користи SD-картичка со минимум 16GB меморија. NOOBS (New Out-Of-Box Software) е специјален софтвер кој овозможува избор на еден од повеќе оперативни системи за микрокомпјутерот Raspberry Pi и

автоматска инсталација со неколку клика на глувчето. Меѓу понудените оперативни системи е оперативниот систем Raspbian.

Категоријата Programming во менито на оперативниот систем Raspbian содржи развојни програми меѓу кои е ThonnyPython IDE.

Инструкции за работа со тастер се:Button(), wait_for_press(), wait_for_release(), held_time, hold_time, is_held и is_pressed.

Инструкции за работа со LED-диода се: LED(), blink(*on_time=1, off_time=1*), off(), on() и toggle().

Прашања и задачи:

1. Кои хардверски компоненти ги содржи во себе платформата Arduino Uno?
2. Наброј неколку различни модели на микрокомпјутери од серијата Arduino?
3. Кои се трите основни разлики меѓу микрокомпјутерите Arduino Uno и Arduino Nano?
4. Опиши ги можностите за поврзување со други уреди на микрокомпјутерот Raspberry Pi, споредено во однос на микрокомпјутерот Arduino Uno!
5. Кои се предностите и недостатоците на микрокомпјутерот Arduino Uno во однос на микрокомпјутерот Raspberry Pi?
6. Кои се двата начини за напојување на микрокомпјутерот Arduino Uno?
7. Како можат да се искористат дигиталните пинови на микрокомпјутерот Arduino Uno за пренесување аналогни сигнали?
8. Колку LED-диоди индикатори содржи микрокомпјутерот Arduino Uno и за што служи секој од нив?
9. Наброј неколку влезни уреди за микрокомпјутерот Arduino Uno!
10. Кои отвори на површината од протобордот се електрично поврзани?

11. Колку пинови има еден тастер и кои од нив се електрично поврзани?

12. На што треба да се внимава при поврзувањето на LED-диодата во електрично коло?

13. Објасни, како работи серво-моторот?

14. Каква намена има компонентата пиезо?

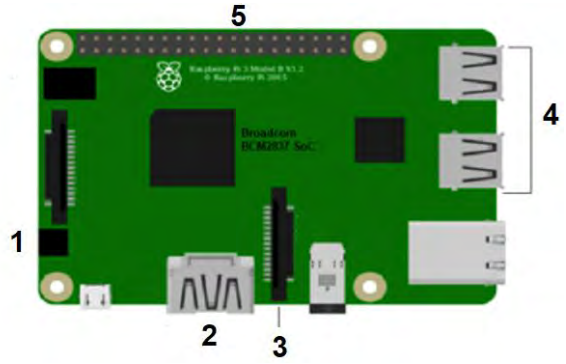
15. За што служат кондензаторите и отпорниците при поврзување на влезно-излезни уреди со микрокомпјутерот Arduino Uno или Raspberry Pi?

16. Набroј неколку функции на штитовите Arduino!

17. Објасни, како се поврзува протобордот со напојувањето на микрокомпјутерот Arduino Uno?

18. Какви апликации може да се креираат со микрокомпјутерот Raspberry Pi?

19. Именувај ги составните делови на микрокомпјутерот Raspberry Pi на долната слика означени со броевите!



20. GPIO (General Purpose Input Output) подножјето на микрокомпјутерот Raspberry Pi 3 има 40 пинови, но GPIO-броевите се разликуваат од физичките броеви на пиновите. Објасни!

21. Набroј, кои категории се наоѓаат во менито на работната површина на оперативниот систем Raspbian?

22. Објасни ја постапката за инсталација на драјвер после инсталацијата на развојната програма и поврзувањето на микрокомпјутерот Arduino Uno со персонален компјутер?

23. Набројте ги составните делови на развојната средина на микрокомпјутерот Arduino Uno!

24. За што служи серискиот монитор во состав на развојната средина на микрокомпјутерот Arduino Uno?

25. Кои се основните градбени елементи на програмскиот јазик C++?

26. Што подразбираме под поимот „синтакса во програмирањето“?

27. Набројте ги основните видови податоци што се користат во програмскиот јазик C++!

28. Што претставува име, вредност и вид на податокот bool isCodingFun = true;?

29. Кои се ознаките за текстуален податок и за низа?

30. Колку вредности можат да имаат логичките податоци?

31. Што претставуваат операторите во програмскиот јазик C++?

32. Кој е оператор за доделување вредност на променлива?

33. Каков резултат ќе се добие по извршувањето на инструкцијата x=13%5;?

34. Напишете го скратениот исказ за аритметичката инструкција x=x-1!

35. Кој е условот за извршување на блокот инструкции под структурата if (x!=y) {…} ?

36. Кои се параметри на инструкцијата digitalWrite ()?

37. За што служи аналогно-дигиталниот претворувач во составот на микрокомпјутерот Arduino Uno?

38. Коментирајте го следниот код:

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

39. За што служи инструкцијата tone (pin, frequency);?

40. Со која инструкција се мери времето од почетокот на извршувањето на програмата во микрокомпјутерот Arduino Uno?

41. Објаснете ја инструкцијата constrain (x, a, b)!

42. Која инструкција се користи за ресетирање на одреден бит во дадена променлива?

43. Како се вика алатката за следење на податоците на пренос преку пиновите RX и TX?

44. Кои се задолжителни структури во програмите за микрокомпјутерот Arduino Uno и која е нивната функција?

45. Што се запишува во малата и во средната заграда на структурата if () {…} ?

46. Именувајте ја структурата за избор на можности!

47. Која е разликата меѓу структурите while…do и do….while?

48. Колкупати ќе се повтори циклусот во следниот код:

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

49. Кои три искази се запишуваат во малите загради на структурата for (…) {…}?

50. Што претставува секој параметар во инструкцијата LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)?

51. Коментирајте ја инструкцијата servo.attach (pin, min,max);?

52. Коментирајте го следниот код:

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

53. Кои се предности и недостатоци на програмскиот јазик Python?

54. По што се разликуваат алатките компајлер и интерпретатор?

55. Зошто во програмскиот јазик Python нема потреба од декларирање на променливи?

56. Која наредба од програмскиот јазик Python се користи за креирање нови функции?

57. Зошто велиме дека програмскиот јазик Python е софтверски ориентиран?

58. Која библиотека од програмскиот јазик Python ја користиме за работа со влезно-излезни уреди?

59. Како се обележани GPIO-пиновите со физички броеви 1,2,3 и 5? Објаснете ја нивната функција!

60. Објаснете, по што се разликуваат следниве инструкции: import gpiozero, from gpiozero import Button, from gpiozero import ButtonBoard!

61. Набројте неколку аналогни влезни уреди за микрокомпјутер Raspberry Pi!

62. Објаснете, како микрокомпјутерот Raspberry Pi одлучува дали некој аналоген влезен уред е активен или не!

63. Кои параметри се карактеристични за инструкцијата Butoon()?

64. Каде се сретнува и за што служи инструкцијата wait_to_release?

65. Коментирајте ги инструкциите и функцијата на кодот!

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

66. Зошто е потребен кондензатор при поврзување на фотоотпорник со микрокомпјутер Raspberry Pi?

67. За што служат двата податочни приклучоци кај сензорот за растојание?

68. Која библиотека и класа се користи за да се постави уредот во режим на мирување?

69. Набројте неколку излезни уреди што се поврзани со микрокомпјутерот Raspberry Pi преку USB-приклучок!

70. Во која класа се користи и каков резултат дава инструкцијата distance?

71. Која инструкција се користи за промена на состојбата на LED-диодата?

72. Коментирајте ги инструкциите и функцијата на кодот!

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

73. Како влијае ширината на широчинско-модулираните импулси врз интензитетот на светлината на PWMLED-диодата?

74. Коментирајте ги инструкциите и функцијата на кодот!

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

75. Коментирајте ги инструкциите и функцијата на кодот!

```
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. Кои се трите основни бои и какви вредности треба да имаат тие за да се добие бела и црна боја?

77. Коментирајте ги инструкциите и функцијата на кодот!

```
from gpiozero import TrafficLights
from time import sleep
from signal import pause
lights =TrafficLights(2, 3, 4)
def traffic_light_sequence():
    whileTrue:
        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. Со која инструкција може да се менува брзината на вртење на DC-мотор?

79. Коментирајте ги инструкциите и функцијата на кодот!

```
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. Кој параметар ја одредува брзината на вртење на DC при негово поврзување со микрокомпјутер Raspberry Pi?

81. За што служи инструкцијата value() при работа со серво-мотор?

82. Коментирајте ги инструкциите и функцијата на кодот!

```
from gpiozero import Servo
```

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

Практични вежби за наставната програма Компјутерски системи

1. Мерки за заштита и безбедност при работа

Учениците самостојно ги извршуваат практичните вежби во групи, не повеќе од 2-3 ученици, под надзор на предметен наставник. Заради ефикасно и безбедно изведување на вежбите како и правилно ракување со електронската опрема, учениците треба да се придржуваат кон следните правила.

- Поврзувањето на електронските компоненти се извршува исклучиво во безнапонска состојба.
- Електричната опрема не се допира со влажни или мокри раце.
- Не е дозволено истовремено со едната рака да се држи проводник, инструмент или друг елемент под напон, а со друга рака друг проводник.
- Алатите за работа треба да имаат изолирани дршки.
- Пожар предизвикан од електрична струја не се гаси со вода.
- По исклучувањето од струјното коло електролитските кондензатори треба да се испразнат со кусо спојување на нивните изводи
- При работа под напон да не се допираат спроводници, отпорници, батерии поради нивно загревање.
- Електронскиот отпад да се собира на посебно предвидено место.
- Пред почетокот на секоја практична вежба ученикот треба внимателно да ги прочита барањата за изведба на вежбата и да повтори за начинот на работа на електронските компоненти кои се користат во истата.
- Електронските компоненти да се поврзуваат внимателно, пополека и без употреба на сила.
- Жиците за поврзување на електронските компоненти да не се испреплетуваат и премногу да се затегнуваат
- По поврзувањето на електронските компоненти во електрично коло да се повика наставникот за да изврши проверка, по чие одобрение се вклучуваат потребните напони.
- За сите дефекти, оштетувања или недостатоци на компонентите или приборот веднаш да се извести предметниот наставник.

2. Практични вежби за инсталација на хардвер на персонален компјутерски систем (PC)

2.1. Мерки за заштита и безбедност при работа со хардверски компоненти на персонален компјутер

- Со цел да се спречи појава на струен удар пред да започне инсталацијата да се исклучи напојувањето на компјутерот.
- Да се провери исправноста на сите кабли и конектори пред да се вклучи напојувањето.
- Да се користи уред за напојување со соодветна сертификација од самиот производител.
- Задолжително да се прочита техничко-технолошката документација за конкретниот модел на матична плоча бидејќи може да постојат извесни разлики во параметрите и обележувањето.
- Бидејќи електронските чипови се осетливи на статички електрицитет потребно е да се допре некаква метална површина, на пример самото куќиште. На таков начин доаѓа до празнење на човековото тело.
- За да се избегне кратко спојување да не се допираат конекторите и подножјата со метални предмети како завртки или шрафцигери.
- Инсталацијата на хардверските компоненти да се изведе на неподвижна и стабилна работна површина.
- Компјутерот треба да се заштити од влага и нечистотија бидејќи таа штетно влијае врз контактите на матичната плоча и го намалува ладењето и вентилацијата, па може да дојде до презагревање.
- Корисникот треба правилно да го исклучува компјутерот за да се спречи евентуално оштетување на неговите делови.

2.2. Практична вежба за инсталација на составни делови на персонален компјутер

1. Цел на вежбата

Цел на вежбата е вградување на хардверски компоненти во внатрешноста на куќиштето односно нивно поврзување со матичната плоча и уредот за напојување преку соодветните кабли и приклучоци. После инсталацијата на компонентите и затворањето на куќиштето, треба да се поврзат тастатура, глувче и монитор за да се провери исправноста. Се разбира за да се добие функционален компјутер потребна е инсталација на оперативен систем и

кориснички програми, што е предмет на проучување на следната модуларна единица.

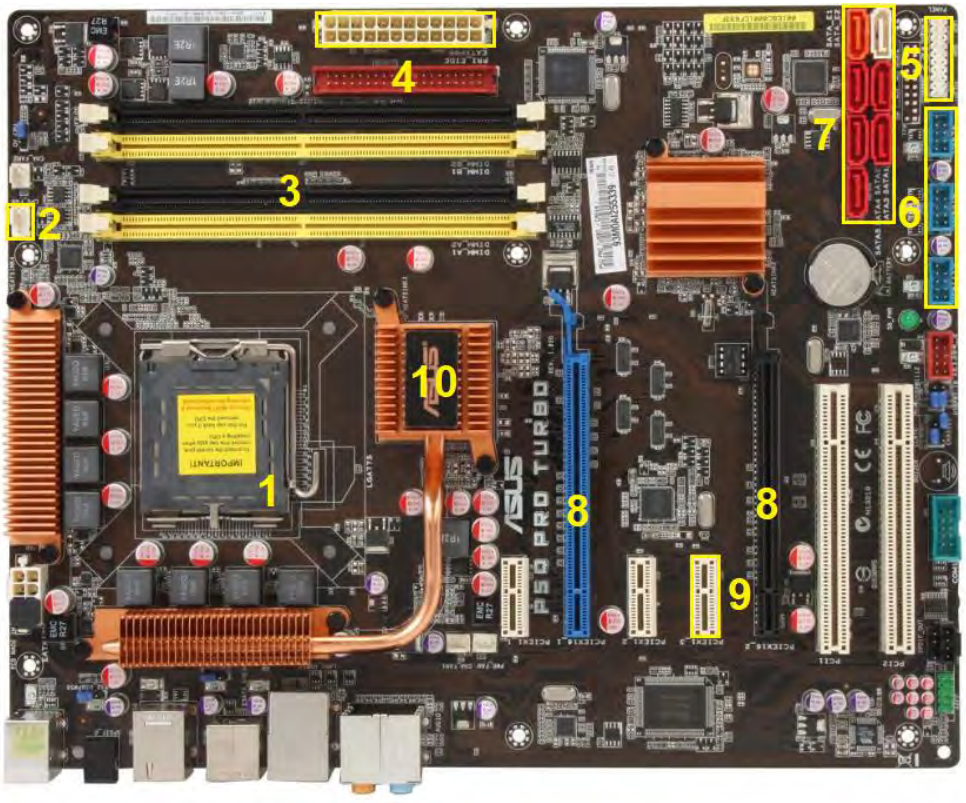
2. **Време за реализација:** 4 наставни часа

3. **Потребни компоненти**

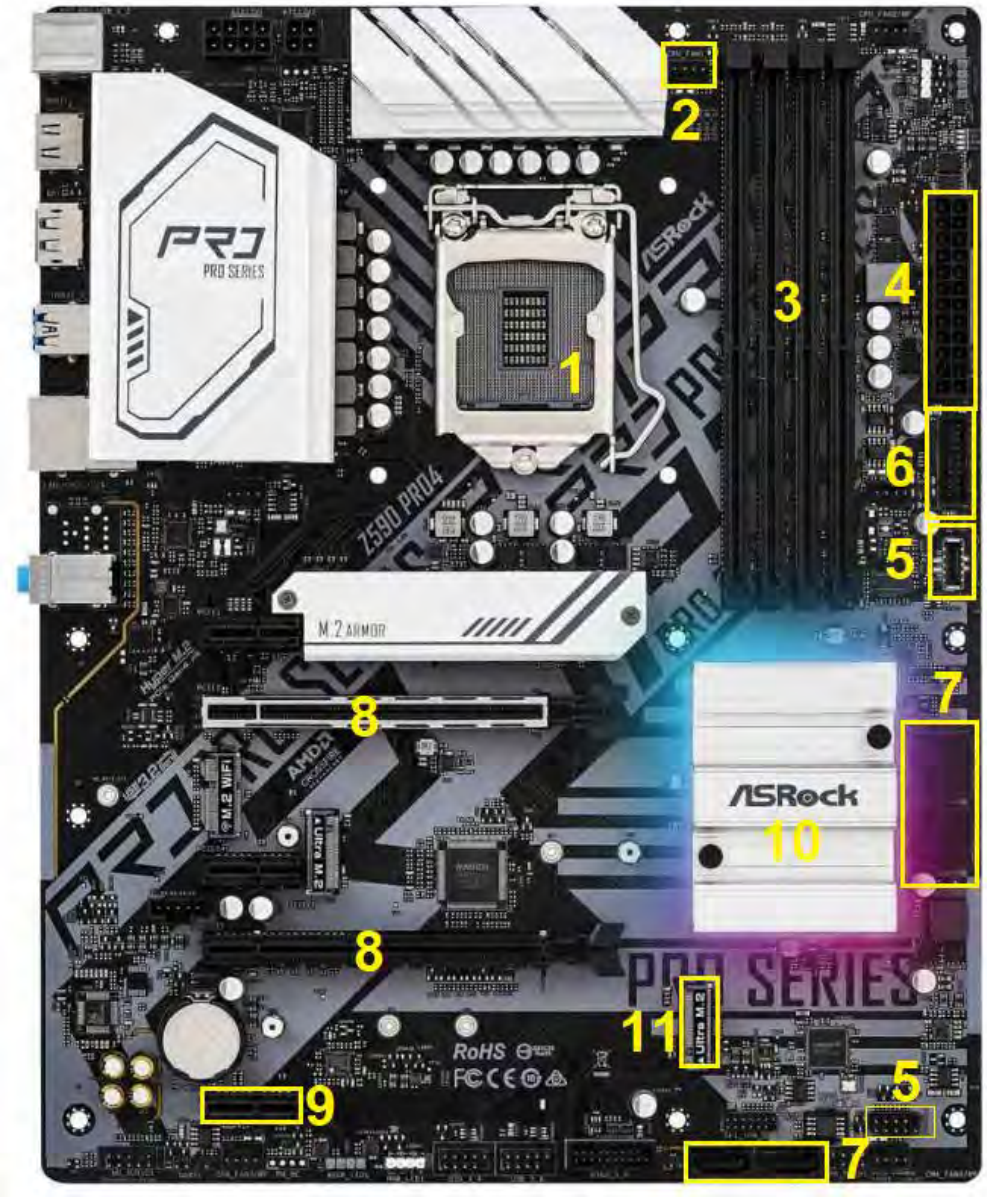
За реализација на оваа вежба потребен е персонален компјутер со следните хардверски компоненти: процесор, ладилник за процесор, матична плоча, RAM меморија, уред за напојување, хард-диск, CD ROM уред и графичка картичка. Бидејќи при ракување со хардверските компоненти може да дојде до нивно оштетување се препорачува компјутерски систем со послаба конфигурација и перформанси, но сепак да биде функционален. Исто така, од алат потребни се два шрафцигери (рамен и крстач). Опционално може да се користи антистатик алка за рака, пластични стеги, термална паста, алкохол и микрофибер крпа за чистење.

4. **Подготовка за вежбата**

- Секој компјутер си има свои специфичности и поради тоа пред почетокот на практичната вежба потребно е да се разгледа техничко-технолошката документација за секоја компонента посебно.
- На слика 2.1. е прикажан изгледот на постариот ASUS P5Q Pro Turbo LGA 755 Intel модел, а на слика 2.2. изгледот на поновиот ASRock Z59p Extreme WiFi GE LGA 1200 Intel модел на матична плоча. За двата модели да се допише називот на означените хардверски компоненти во табела 2.1 .



Слика 2.1. Матична плоча, модел ASUS P5Q Pro Turbo LGA 755 Intel



Слика 2.2. Матична плоча, модел ASRock Z59p Extreme WiFi GE LGA 1200 Intel

Реден број	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	

Табела 2.1. Компоненти на матична плоча и нивно препознавање

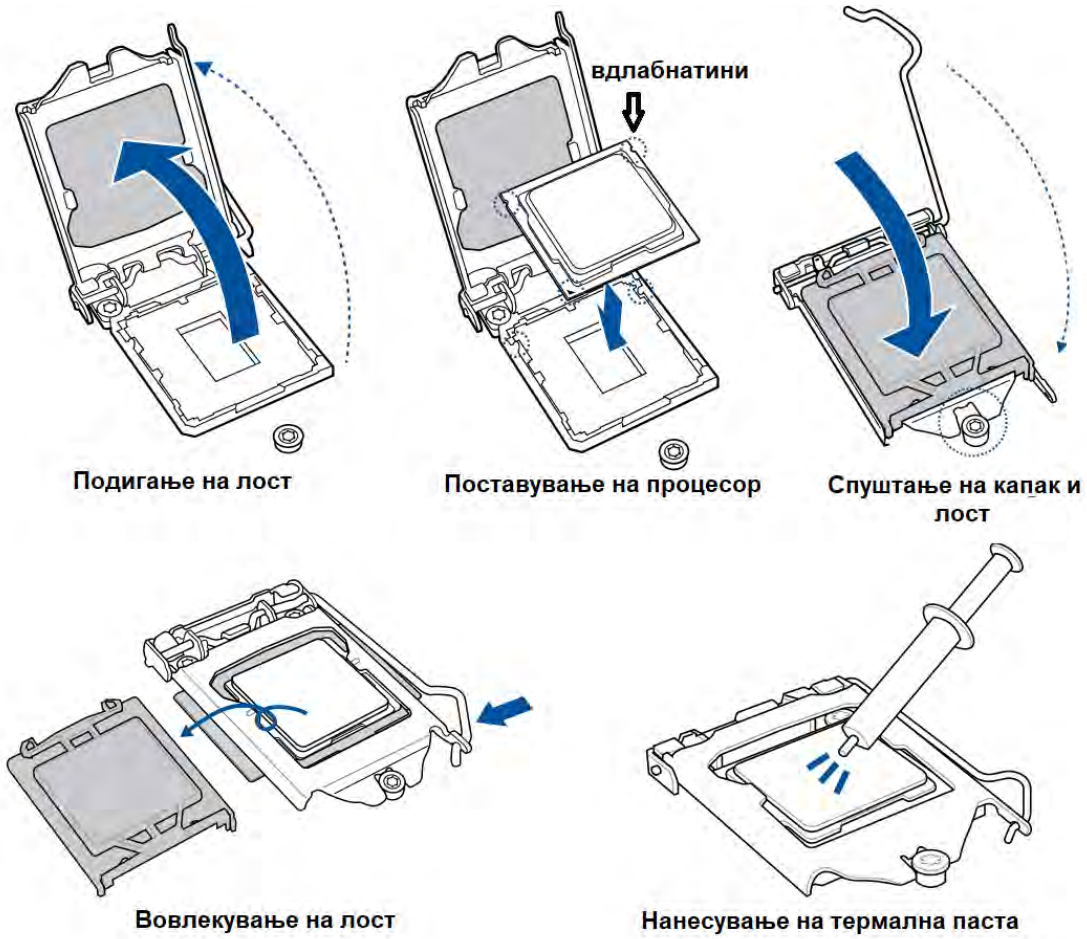
5. Инсталација на процесор

(1) подготовка

- Пред почетокот на инсталацијата на процесорот во неговото подножје да се провери нивната компатибилност.
- Да се изврши споредба на процесорските подножја, модел LGA 1151 (слика 2.1) и модел AMD AM3, и да се истакнат нивните специфичности. Да се наведат процесорите кои се компатибилни со овие модели на подножја.

(2) инсталација

Доколку се работи за нова матична плоча, подножјето на процесорот треба да биде покриено со жолта заштитна фолија. Од десната страна на подножјето се наоѓа **метален лост**, кој треба да се притисне надолу, да се повлече надесно и да се крене нагоре, при што ќе се подигне и капакот на подножјето. Жолтата заштитна фолија се вади и таа треба да се сочува во случај да се извади процесорот. Процесорот го поставуваме на подножјето, со пиновите свртени надолу, при што внимаваме двете странични вдлабнатини на процесорот да се совпаднат со двете испакнатини на подножјето. Се спушта капакот до навртката, се спушта металниот лост и тој се вовлекува во лево, за да се намести во своето место. [10]



Слика 2.3. Чекори при инсталација на процесор

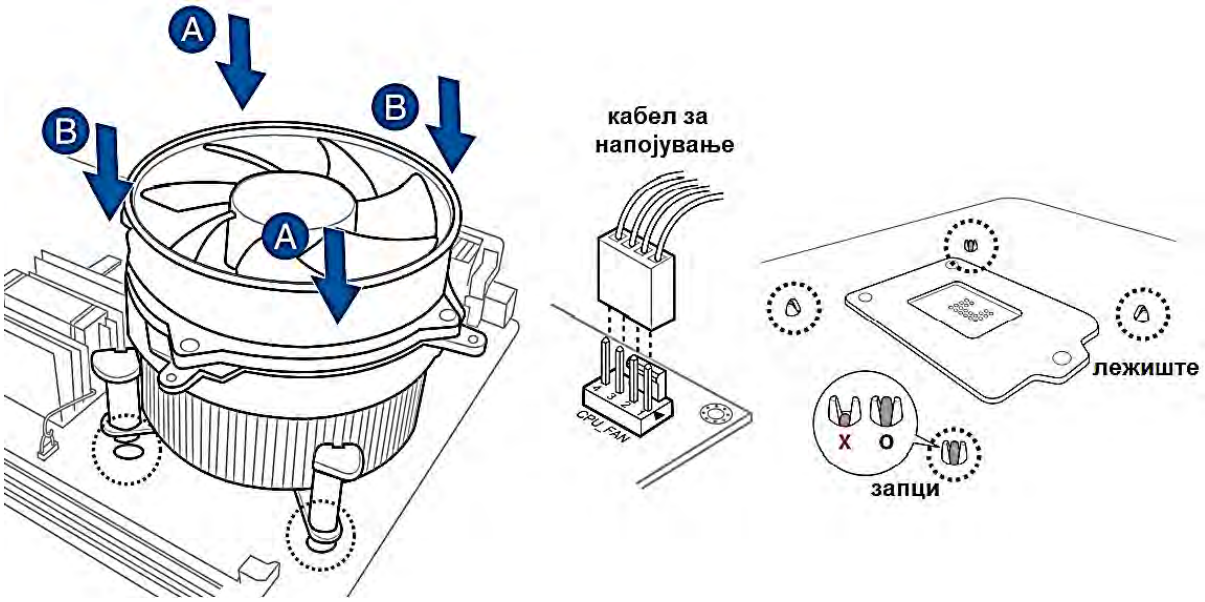
(3) Процесорот треба да се држи отстрана и да не се допираат златните пинови. Процесорот не смее да се притиска.

6. Инсталација на ладилник

(1) подготовка

- При изборот на ладилник за процесор треба да се внимава дали истиот е компатибилен со соодветниот процесор. Ладилниците можат да бидат со: различни димензии, вид на ладење (воздушно или со течност), начин на монтажа (со завртки или спуштање на клипови).
- На пример, да се истражи кои ладилници се компатибилни со LGA 1151 моделот на процесорско подножје.

Коментар:_____
_____



Слика 2.4. Чекори при инсталација на ладилник за процесор

(2) инсталација

Пред инсталацијата на ладилникот треба да провери дали има потреба од нанесување на **термална паста** врз горниот дел на процесорот. Термалната паста ја намалува температурата од 20˚ до 30˚. Пред да се нанесе нова паста, треба да се отстрани старата и за таа цел, најдобро е да се користи 90% алкохол и микрофибер крпа. Процесорот треба претходно да се извади од подножјето. На крпата се нанесува мало количество алкохол, колку големина на поштенска марка, и потоа нежно се поминува горниот дел на процесорот. Пастата се нанесува на средината, колку половина од зрно грашок. Пастата ќе се размачка под дејство на топлината на процесорот и тежината на ладилникот. Пред да го поставиме ладилникот врз процесорот, проверуваме дали жлебовите на главите **од пластичните запци** се насочени кон центарот на ладилникот. Избираме два запци што се дијагонално поставени и ги

притискаме. Откако ќе ги притиснеме сите четири запци, од задната страна на матичната плоча проверуваме дали запците се израмнети со своите лежишта.

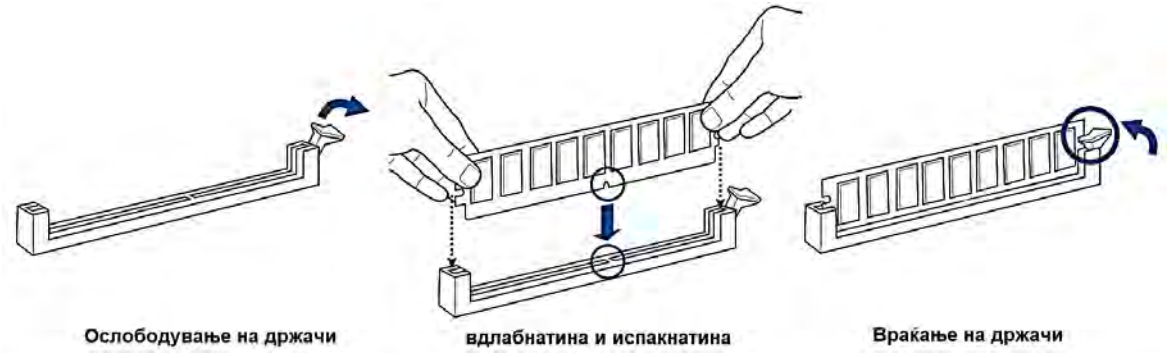
(3) При инсталацијата на ладилникот треба да се внимава да не се оштети матичната плоча поради примена на прекумерна сила.
Внимателно да се ракува со термалната паста бидејќи истата е токсична материја и се препорачува употреба на заштитни ракавици.

7. Инсталација на RAM-меморија

- Од изборот на матичната плоча зависи видот и капацитетот на RAM-меморијата. Прво се избира видот (DDR2, DDR3, DDR4 или DDR5), потоа капацитетот на секој од модулите. На пример, DDR4 модулите може да бидат со капацитет од 2GB и 32GB. Истовремено треба да ја избереме брзината на RAM-модулот. На пример, DDR4 модулите можат да бидат со брзина од 2133MHz до 5000MHz. Треба да постои усогласеност меѓу брзината на работа на RAM-меморијата и самиот процесор.
- На пример, максималниот капацитет на RAM-меморијата за матичната плоча ASUS P5Q Pro Turbo LGA 755 Intel (слика 2.1) изнесува 16GB, таа поддржува DDR2-модули, со оптимална брзина 800MHz. Истражи ја RAM-меморијата за матичната плоча ASRock Z59p Extreme WiFi GE LGA 1200 Intel (слика 2.2).

(1) подготовка

Коментар:_____
_____



Слика 2.5. Чекори при инсталација на RAM-меморија

(2) инсталација

Матичните плочи можат да бидат со 1, 2, 4 или 8 слотови за поставување на RAM-мемориски модули. Тие можат да бидат поставени поединечно (едноканално поврзување) или во пар (двоканално). Двоканалното поврзување ја зголемува брзината за пренос на податоци за двапати. За двоканално поврзување RAM-модулите треба да бидат идентични (по вид, брзина, капацитет, производител) и да се постават во слотови со иста боја. Ако RAM-слотовите не се во различна боја тогаш тие се поставуваат еден преку друг, на пример со редни броеви 2 и 4 или 1 и 3,

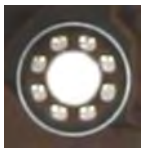
но најдобро е да се провери во техничко-технолошката документација на самата матична плоча.
Помеѓу пиновите се наоѓа мала вдлабнатина и таа вдлабнатина треба да се совпадне со испакнатината на DIMM-слотовите (RAM-подножјето). Пред да го вметнеме мемориѕкиот модул, потребно е да се ослободат страничните држачи на слободниот слот.

(3) Слично како кај процесорот, и кај RAM-меморијата треба да внимаваме на статичкиот електрицитет. Затоа, најдобро е пред инсталацијата да го допреме барем металното куќиште. Исто така, RAM-модулите треба да се држат странично и никако да не се допираат златните пинови.
Доколку видот на RAM-слот не се поклопува со видот на RAM-мемориски модул, при инсталација ќе дојде до оштетување на хардверските компоненти

8. Монтажа на матична плоча во куќиште

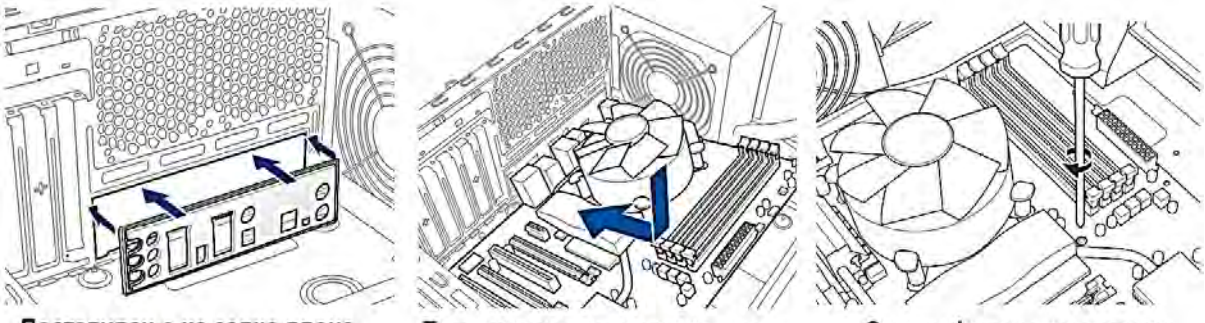
(1) подготовка

- Најчести формати (големини) за матични плочи се: ATX, mATX (анг. mini) и EATX (анг. extended). Изборот на куќиште зависи од форматот на матичната плоча.
  Најдобро е процесорот, ладилникот и RAM-меморијата да се инсталираат пред матичната плоча да се монтира во куќиштето.


Слика 2.6

- Потоа на матичната плоча ги лоцираме отворите за поставување на завртките за прицврстување. Овие отвори се посебно обележани како на слика 2.6. Нивниот број и распоред се разликува, во зависност од видот на матична плоча.
- Да се лоцираат отворите за прицврстување на матичните плочи: ASUS P5Q Pro Turbo LGA 755 Intel (слика 2.1.) и ASRock Z59p Extreme WiFi GE LGA 1200 Intel (слика 2.2.)

Коментар:_____
_____



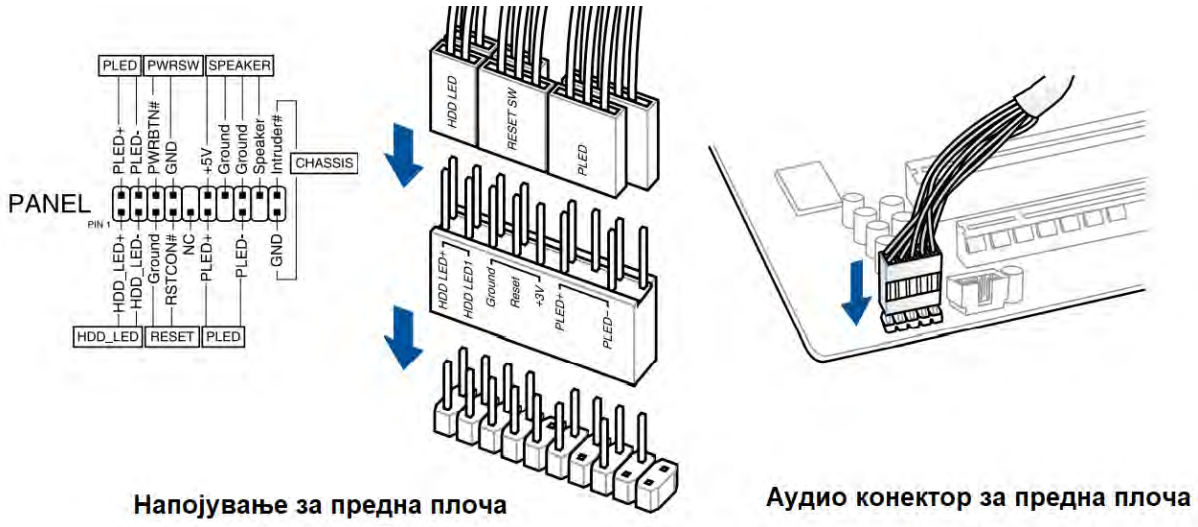
Слика 2.7. Чекори при монтажа на матична плоча

(2) Инсталација

Матичните плочи доаѓаат во комплет со задната плоча на којашто има отвори за различните видови на конектори: USB, HDMI, PS/2 конектор за глувче и тастатура, аудио џекови, мрежен RJ-45 конектор итн. Најнапред се поставува задната плоча.

Потоа се лоцираат навртките во внатрешноста на куќиштето. Матичната плоча се поставува така што отворите за завртките на матичната плоча да се совпаднат со навртките на куќиштето.

Откако ќе се зашрафат завртките, потребно е да се поврзат приклучоците за напојување на предната плоча и приклучокот за напојување на вентилаторот на самото куќиште. На предната плоча од куќиштето има копче за напојување, копче за ресетирање, предни USB-порти, предни аудио конектори и LED-диоди како индикатори за напојувањето и за хард дискот. Нивните приклучоци за напојување треба да се поврзат со подножјето на матичната плоча, кое е обележано со буквата F (Front). USB и аудио конекторите се приклучуваат наједноставно поради уникатниот распоред на нивните пинови. Кај приклучоците на копчињата за напојување и за ресетирање не е важен поларитетот. Кај LED-диодите треба да се внимава на знаците. Вообичаено, црната и белата жица се плус, а жиците во другите бои се минус односно заземјување. На крајот, на матичната плоча треба да се лоцира подножјето со ознака SYS_FAN и PWR_FAN и на него да се постави приклучокот за напојување на вентилаторот на куќиштето.



Слика 2.8. Поврзување на матичната плоча со напојувањето

(3) При поставувањето на задната маска треба да внимаваме на острите рабови.

При зашрафувањето на завртките не треба да се користи прекумерна сила. Треба да се зашрафат што е можно повеќе завртки заради поголема механичка стабилност на матичната плоча.

9. Инсталација на уред за напојување

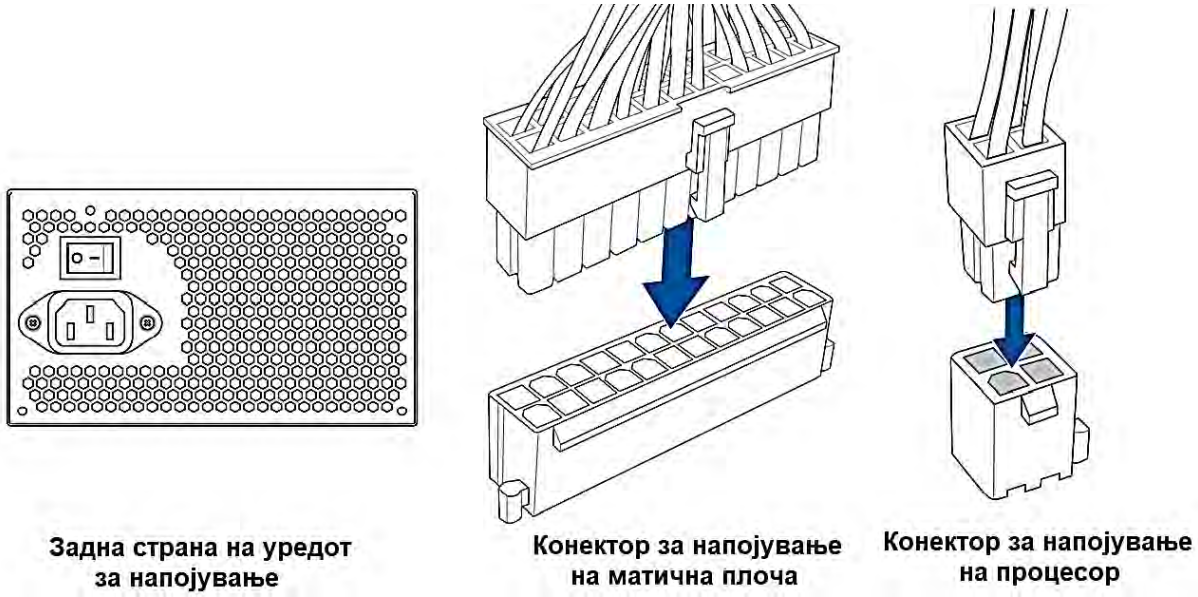
(1) подготовка

Постојат модуларни и немодуларни уреди за напојување. Основната разлика е што кај немодуларните уреди каблите се фабрички поврзани со уредот и не можат да се извадат од него.

Најважна карактеристика на уредите за напојување е нивната моќност и истата се движи од 300W до1600W. Без соодветно напојување хардверските компоненти ќе функционираат со намалени перформанси и стабилност. Поради тоа треба да се провери каква е потрошувачката на енергија на секоја компонента и да се пресмета потребната моќност на уредот за напојување. Најголеми потрошувачи на електрична енергија се процесорот и графичката картичка.

Според однапред дефинирана компјутерска конфигурација пресметај ја моќноста на уредот за напојување. Користи го калкулаторот на следниов линк https://www.newegg.com/tools/power-supply-calculator/

Коментар:_____
_____
_____



Слика 2.9. Инсталација на уред за напојување

(2) инсталација

Уредот за напојување може да се постави во горниот или во долниот дел, на задната страна на куќиштето. Го внесуваме уредот во внатрешната преграда, со вентилаторот свртен кон матичната плоча, го туркаме до задниот дел и истиот се прицврстува.

Уредот за напојување има повеќе кабли што треба да се поврзат со матичната плоча или со одредена компонента. Добро е што сите кабли за напојување имаат уникатен дизајн, па не може да се згреши при нивното поврзување. На матичната плоча се поврзуваат два конектори: ATX-

кабелот со 20+4 пински конектор, кој ја напојува матичната плоча, и ATX 12 V со 4 или 8 пина, кој го напојува процесорот. 8-пинскиот конектор обезбедува моќност од 235W, а 4-пинскиот 155W. За напојување на хард дискот и DVD-уредот се користат два вида кабли: SATA и IDE со 4 пинови, и нив ќе ги објасниме наскоро. Шест-пинскиот кабел за напојување на графичката картичка е обележан со ознаката VGA1 и тој е познат под името PCI Express кабел.

(3)

Употребата на уреди за напојување со нестабилен напон може да предизвика оштетување на матичната плоча и другите хардверски компоненти.

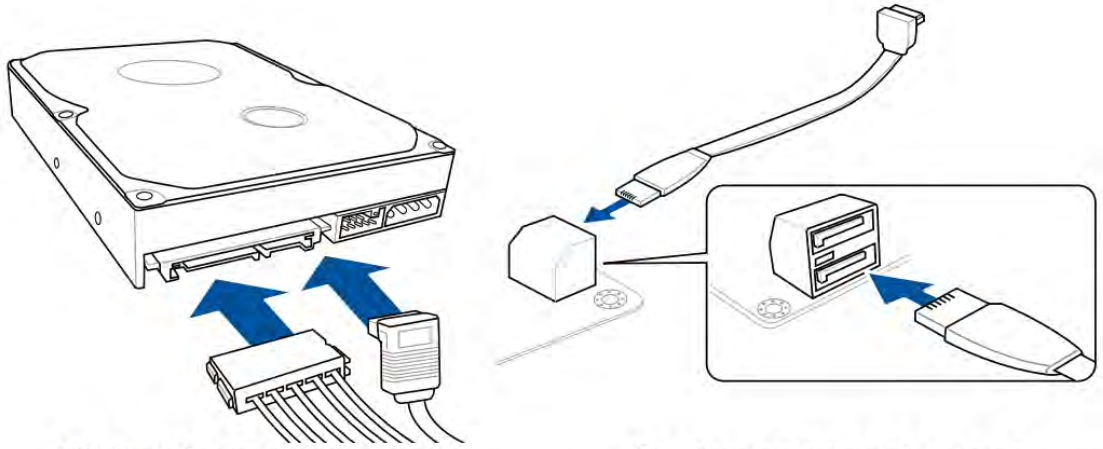
10. Инсталација на хард-диск и SSD уред

(1) подготовка

- Учениците треба да се потсетат на основните карактеристики на хард-дискот: принципот на работа, капацитетот, физичките димензии, брзината за пренос на податоци, брзината на ротација, кеш меморијата, приклучоците за поврзување.
- Да се споредат карактеристиките на два модели на хард-диск, по случаен избор.

Коментар:_____
_____

- Да се лоцираат SATA приклучоците за поврзување на хард-дискот со матичната плоча и поврзување на хард-дискот со уредот за напојување



SATA кабел за напојување и податочен кабел

Поврзување на каблите со приклучокот на матичната плоча

Слика 2.10. Поврзување на хард-диск со SATA-кабли

Во внатрешноста на куќиштето постои посебна преграда со висина од 9 cm. Во неа се вметнува хард дискот, при што напред треба да се гледаат неговите приклучоци во форма на буквата L. Во поголемиот приклучок се

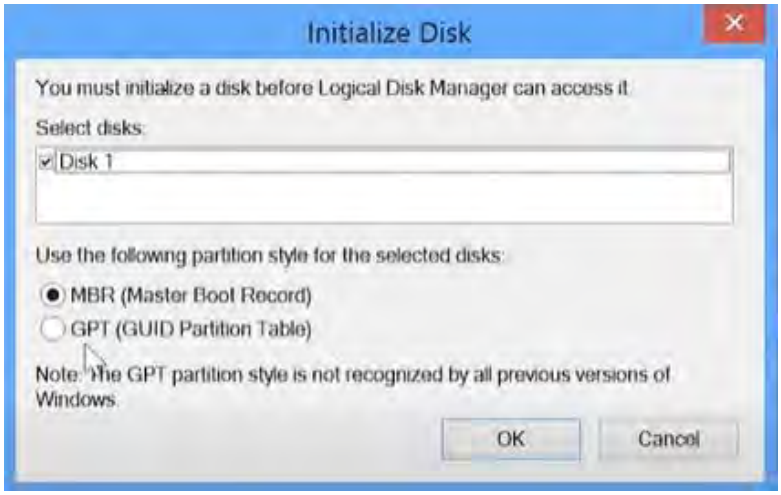
(2) инсталација

прицврстува **кабелот за напојување** од уредот за напојување, а во помалиот приклучок се поставува **SATA податочниот кабел**, кој доаѓа во комплет со хард-дискот. Другиот крај на SATA податочниот кабел се приклучува на конектор на матичната плоча обележан со SATA2, во форма на буквата L.

Бидејќи SATA SSD мемориските уреди се со иста големина и распоред на пинови како и хард-дисковите, за нивно поврзување се користат истите SATA конектори за напојување или пренос на податоци. Се разбира претходно треба да се провери дали во куќиштето постојат две прегради за SSD и хард-диск.

(3)

Новоинсталираниот хард-диск е треба претходно да се иницијализира со притискање на десен клик и избор на опцијата „Initialize Disk“. После отворањето на новиот прозорец како што е прикажано на слика 2.10. треба е да се избере табелата за партиции, MBR (анг. Master Boot Record) или GPT (анг. Globally Unique Identifier Partition Table). Партициите се виртуелни делови на хард-дискот. Табелата за партиции содржи информации за тоа како се организирани партициите, каде започнува и завршува секоја партиција, кои сектори ги зафаќа, а исто така содржи и код за стартување на оперативниот систем познат под името подигнувач (анг. Boot Loader).



Слика 2.11. Избор на табела на партиции

Кратенката MBR во превод значи главен запис за стартување, а кратенката GPT значи табела за партиции со глобален единствен идентификатор. Со MBR табелата може да се адресираат само 2TB мемориски капацитет на хард-дискот, а кај GPT табелата нема такви ограничувања односно максималниот мемориски капацитет изнесува 9,7ZB (1 зета бајт е еднаков на милијарда тера бајти). MBR табелата поддржува само 4 партиции, од кој едната може да се конфигурира како

проширена (анг. extended) и истата може да се подели на 23 дополнителни партиции. GPT табелата содржи 128 различни партиции што е повеќе од доволно за денешните реални апликации. Да нагласиме дека MBR табелата е поврзана со BIOS програмата, а GPT табелата со UEFI програмата. Компјутерите со BIOS програма не може да имаат GPI партиции.

Во случај на инсталација на втор хард-диск, како надградба на постоечката конфигурација, во самиот оперативен систем треба да се изврши негова распределба (анг. allocate). За таа цел треба да се отвори Disk Management програмата во составот на Windows 10 оперативниот систем. Новоинсталираниот хард-диск е обележан како нераспределен простор (анг. unallocated space) и со притискање на лев клик треба да се одбере опцијата "New Simple Volume".

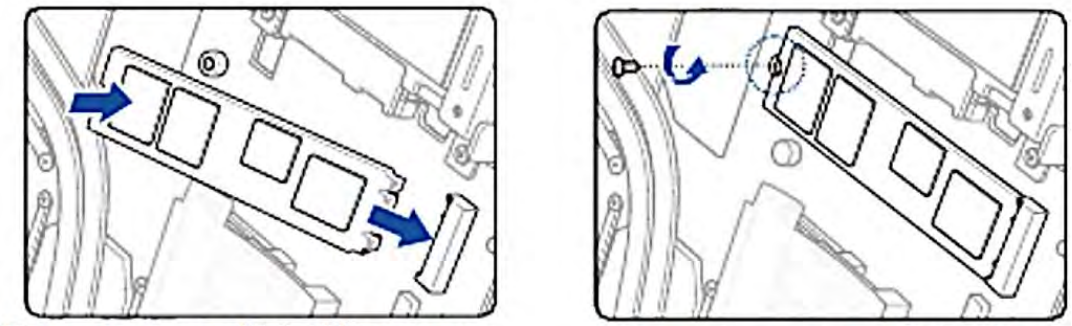
Исто така постои можност датотеките од стариот хард-диск да се пренесат и на новиот. Оваа постапка е позната под називот клонирање и истата е од понапредно ниво.

(1) подготовка

11. Инсталација на M.2 SSD модул

- При инсталацијата на M.2 SDD-модулот треба да го провериме типот на конекција на истиот, PCIe или SATA. Доколку модулот и конекторот на матичната плоча немаат иста конекција истите не можат физички да се поврзат.
- Да се лоцираат двата M.2 приклучоци за поврзување на SATA SSD мемориски модули на матичната плоча ASRock Z59p Extreme WiFi GE LGA 1200 Intel прикажана на слика 2.2 .

Коментар:_____
_____
_____



Слика 2.12. Инсталација M.2 SDD-модул

(2) инсталација

Под мал агол, пиновите на M.2 SSD-модулот внимателно се втиснуваат во соодветниот конектор. Откако модулот ќе се постави на матичната плоча истиот се прицврстува со помош на завртка. Да напомнеме дека постојат три навртки во близина на M.2 конекторот бидејќи M.2 SSD модулите можат да бидат со три различни должини (22, 60 или 80 mm) додека ширината е секогаш иста.

(3)

При зашрафувањето на завртката за прицврстување на M.2 SSD модулот не треба да се користи прекумерна сила.

(1) подготовка

12. Инсталација на оптички уреди

- Во оваа категорија на уреди спаѓаат CD, DVD и Blue-Ray уредите кои за читање и запишување на дигиталните записи користат ласерска технологија. Како кај претходните хардверски компоненти и овде е важен изборот. Најквалитетни дискови се Blu-Ray дисковите, потоа следуваат DVD и на крај CD. Кога велиме квалитет мислиме на капацитетот и високата резолуција на видео записите. Blu-Ray уредите можат да читаат секакви дискови, DVD уредите DVD и CD дискови, додека CD уредите само CD дискови. Врз цената на чинење влијаат брзината на читање и запишување на податоци.
- Да се направи споредба меѓу два модели, DVD или Blue-Ray уреди, по случаен избор. Да се направи анализа на исплатливоста во однос на набавка на екстерен или интерен уред

Коментар:_____

_____

Слика 2.13. Инсталација на оптички уред

(2) инсталаџ

Оптичкиот уред се монтира на предната страна од куќиштето. За таа цел, потребно е да се извади предната **маска** на куќиштето, која е прицврстена со два запци вметнати во отворите на матичната плоча. Потоа **треба да се отстрани капакот** на 14 cm високата преграда. DVD-

уредот се поставува во преградата и од страна се прицврстува со навртки. Исто како и хард дискот, и DVD-уредот има два SATA-приклучоци во форма на буквата L, еден за напојување и вториот податочен.

(3) Некои прегради користат шини за да обезбедат постабилна положба на уредот. Ако е таков случај тогаш шините треба да бидат закачени на секоја страна од уредот пред тој да биде внесен во преградата на куќиштето.

После инсталацијата на уредот треба да се инсталира неговиот драјвер.

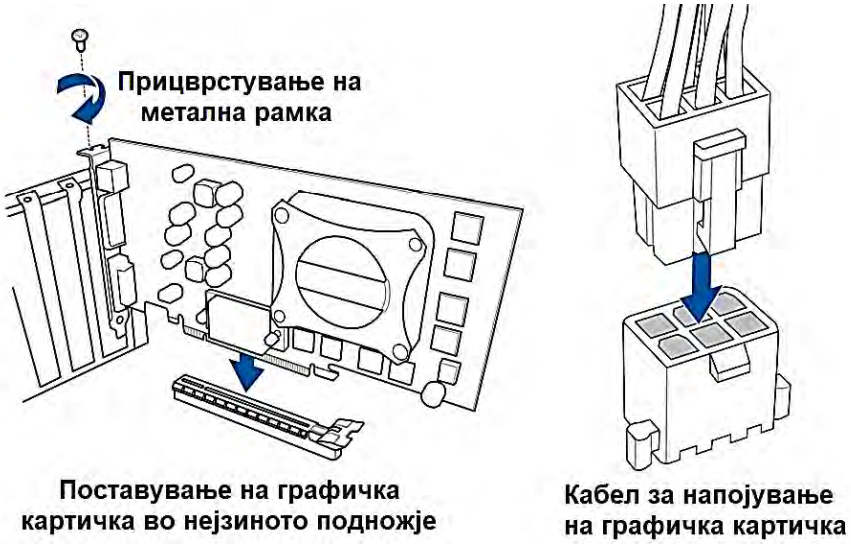
13. Инсталација на графичка картичка

(1) подготовка

- Учениците треба да се потсетат на основните карактеристики на графичките картички: капацитетот на видео RAM-меморијата, физичката должина и потрошувачката на електрична енергија.
- Да се споредат карактеристиките на два модели на графички картички, MSI GeForce GT 710 и MSI GeForce GTX 1050 Ti, прикажани на слика 2.16. во теретскиот дел на учебникот.

Коментар:_____
_____

- Да се лоцираат PCI Expressx16 слотот за поврзување на графичката картичка со матичната плоча.



Слика 2.14. Поставување на графичка картичка во PCI Expressx16 слотот

(2) инсталација

Графичката картичка се поставува на PCI Expressx16 слотот на матичната плоча. Пред почетокот на монтажата, потребно е да се отстранат металните рамки на задната страна на куќиштето, со одвртување на завртките. Откако ќе се вметне графичката картичка во PCI Expressx16 слотот, потребно е да се прицврсти металната рамка на самата графичка картичка, на задниот дел од куќиштето, со помош на

завртки. На крајот се прицврстува шест пинскиот кабел за напојување, обележан со ознаката VGA1.

(3) При изборот на графичка картичка треба да се внимава дали има доволно простор за негова монтажа на самата матична плоча бидејќи новите модели на графички картички се со многу поголеми димензии од постарите модели. Исто така треба да утврдиме дали уредот за напојување може да ја задоволи потрошувачката на енергија на избраната графичка картичка.

Слично како инсталацијата на RAM-мемориските модули, пред да ја вметнеме графичката картичка, потребно е да се ослободи страничниот држач на PCI Expressx16 слотот.

После инсталацијата на графичката картичка треба да се инсталира ѝ неговиот драјвер. Доколку се работи за замена на стара графичка картичка со нова, најдобро е пред инсталацијата на новиот драјвер да се деинсталира драјверот на старата графичка картичка.

14. Проверка на исправност

Ако после монтажата на составните делови, компјутерот не сака да се вклучи, треба да се проверат следниве елементи:

- Дали се правилно вклучени каблите за напојување на компјутерот и мониторот?
- Дали свети зелената LED-диода за напојување на предната страна од куќиштето?
- Дали свети LED-диодата за напојување на мониторот?
- Пред да се отвори куќиштето, задолжително да се исклучи кабелот за напојување!
- Дали се цврсто и правилно споени сите конектори на матичната плоча?
- Дали е исправен уредот за напојување? Може да се измерат напоните на пиновите на ATX 20+4 конекторот.

За дијагностика на проблемот може да помогнат **звучните и светлосните сигнали** кои ги генерира компјутерот при неговото вклучување. Кога излегуваме од дома, обично проверуваме дали со себе ги носиме паричникот, клучевите, мобилниот телефон и другите предмети што се неопходни за започнување на денот. Слично се случува кога компјутерот почнува да работи. Тој проверува дали сите хардверски компоненти се во функција. POST (Power On Self Test) тестот се темели на податоци добиени од CMOS-меморијата. CMOS (анг. Complementary Metal Oxide Semiconductor) претставува RAM-меморија со мал капацитет од 100 до 200 бајти, со сопствена мала батерија за напојување. На пример, POST го мери капацитетот на постоечката RAM-меморија и добиените податоци ги споредува со податоците од CMOS RAM-меморијата. Податоците за видот на вградена меморија се фабрички внесени, но истите може да ги

промени и самиот корисник, доколку се прошири или промени видот на меморија или се внесе дополнителна хардверска компонента. Ако компјутерот го „положи“ POST-тестот тој испушта еден краток послаб звук (beep). Доколку компјутерот не го „положи“ POST-тестот, тој испушта повеќе кратки или долги звуци во зависност од состојбата на компјутерот. Незгодно е што **различни производители користат различни звучни кодови**. Подоле се дадени поважните звучни кодови кои ги користи компанијата Dell.

1 краток звучен сигнал	⟶	Откажување на BIOS ROM-меморијата
2 краток звучен сигнал	⟶	Не е детектирана RAM-меморија
3 краток звучен сигнал	⟶	Откажување на матичната плоча
4 краток звучен сигнал	⟶	Откажување на RAM-меморијата
5 краток звучен сигнал	⟶	Откажување на CMOS-батеријата
6 краток звучен сигнал	⟶	Откажување на графичката картичка
7 краток звучен сигнал	⟶	Откажување на процесорот

На пример, доколку звучниот сигнал укажува на непостоење RAM-меморија, тогаш најмалку што можеме да направиме е да ги извадиме RAM-модулите од нивното подножје, да ги провериме пиновите и повторно да ги вратиме на своето место.

Покрај звучни сигнали, постојат и светлосни сигнали. Зелената и црвената LED-диода на предната маска од куќиштето, кои се всушност индикатори за напојување, можат да светат со различна динамика.

Десктоп компјутерите се многу поедноставни за одржување, за дијагностика и замена на оштетените хардверски компоненти. За да се продолжи животниот век на компјутерот, многу е важно корисникот правилно да се грижи за истиот. Компјутерот треба да се заштити од влага и нечистотија бидејќи таа штетно влијае врз контактите на матичната плоча и го намалува ладењето и вентилацијата, па може да дојде до презагревање. Исто така, корисникот треба правилно да го исклучува компјутерот за да се спречи евентуално оштетување на неговите делови.

3. Практични вежби за инсталација на оперативен систем и други стандардни програми на персонален компјутерски систем

3.1. Подготовка за инсталација на софтвер

Денес, инсталацијата на апликативниот и системскиот софтвер е мошне едноставна, истата се извршува во неколку чекори и секој чекор е детално објаснет во текот на самата постапка. Сепак за успешна инсталација корисникот треба да се придржува до следните правила.

- Треба да бидеме сигурни дека програмата што ја инсталираме ќе ни биде корисна. Секоја инсталирана програма, без оглед дали ја користиме или не, зафаќа меморија и ја успорува работата на компјутерот.
- За преземање на софтверот за инсталација треба да користиме сигурни и проверени линкови.
- Треба да провериме дали компјутерската конфигурација ги задоволува барањата за правилна работа на софтверот.
- Треба да провериме дали имаме доволно слободен простор во трајната меморија.
- Треба да провериме дали постои можност за деинсталација.
- Се препорачува корисникот да избере инсталација по негов избор (анг. Custom) бидејќи на таков начин ги селектира можностите кои му се нудат.
- Треба да се обезбеди стабилно напојување и сигурен интернет пристап за да на дојде до прекин во текот на инсталацијата.
- Практичните вежби за инсталација на софтвер да не се извршуваат без одобрение од предметен наставник.

3.2. Практична вежба: Инсталација на Windows 10 оперативен систем

1. Цел на вежбата

Денес инсталацијата на оперативен систем е едноставна постапка. Откако ќе стартува процесот на инсталација се отвораат прозорци со дадени инструкции кои треба внимателно да се прочитаат и извршат. Во текот на оваа практична

вежба учениците ќе се запознаат со три постапки: креирање на инсталациски модул, негово повикување во BIOS програмата и инсталација на Windows 10 оперативен систем.

2. **Време за реализација:** 2 наставни часа

3. **Подготовки за инсталација на Windows 10 оперативен систем**

Најнапред треба да се избере видот на Windows 10 оперативниот систем. Windows 10 Home е за домашна употреба, Windows 10 Pro е за професионалци, Windows 10 Enterprise за компании и Windows 10 Education за студенти и ученици. Сите видови оперативни системи се достапни во **32 и 64-битна верзија**. Да нагласиме, 64-битната верзија не може да се инсталира на компјутер со 32-битен процесор. Податоците се обработувани побрзо со 64-битните процесори, кои овозможуваат и поефикасно искористување на RAM-меморијата над 4 GB.

Оперативниот систем Windows 10 може да работи со истата компјутерска конфигурација како и Windows 7: процесор со работна фреквенција над 1 GHz, 2 GB капацитет на RAM-меморијата, хард-диск со капацитет 16 GB за 32-битна верзија и 20 GB за 64-битна верзија, графичка картичка со WDDM (анг. Windows Display Driver Model), екран со резолуција 800 × 600 пиксели.

Доколку компјутерот располага со некоја постара лиценцирана верзија на Windows тогаш истата може да се надгради до Windows 10 и самата надградба е бесплатна. Но доколку се работи за нов компјутер тогаш Windows 10 оперативниот систем мора да се инсталира. За таа цел потребен е инсталациски мемориски модул (USB или DVD).

За активација на Windows 10 потребен е клуч на производ. Клучот на производот е код од 25 цифри за потврда на лиценцата. На веб-страната на Microsoft може да се најде официјална листа на добавувачи на Windows 10 кои го испорачуваат кодот заедно со доставата на оперативниот систем, или истиот може да биде испратен преку е-пошта, доколку програмата за инсталација е преземена од интернет. Друг начин за добивање на дигитална лиценца е преку корисничкиот профил на Microsoft. За таа цел се избира Settings→Update and Security→Activation. Го притискаме копчето Додај сметка (анг. Add an account) и се најавуваме со нашето корисничко име и лозинка. Постапката за инсталација на Windows 10 ќе ја поделиме на три дела: креирање на инсталациски USB мемориски модул, негово повикување преку BIOS управувачката програма и конечно инсталација на самиот оперативен систем. [11]

4. **Креирање на инсталациски USB мемориски модул**

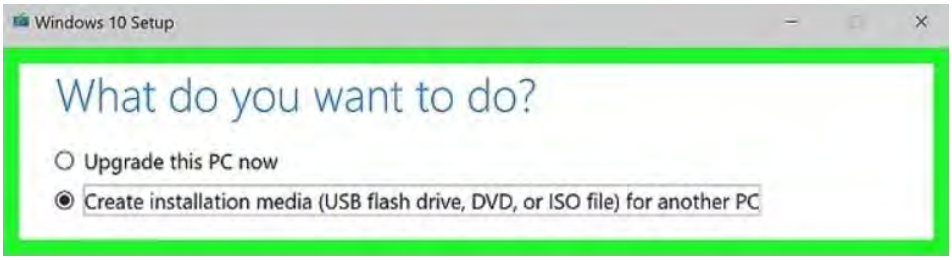
(1) USB мемориските модул го поврзуваме со компјутер. Тој треба да има минимален капацитет од 8GB.

(2) https://www.microsoft.com/en-us/software-download/windows10%20 е линк за преземање на алатката за креирање на инсталациски модул (анг. Media

Creation Tool). Се притиска копчето Download tool now (преземи ја алатката сега) и за неколку секунди алатката се појавува во папката за преземање (анг. Download).

(3) Со двоен клик на алатката започнува креирањето на инсталацискиот модул. Ги прифаќаме наведените услови (анг. Accept)

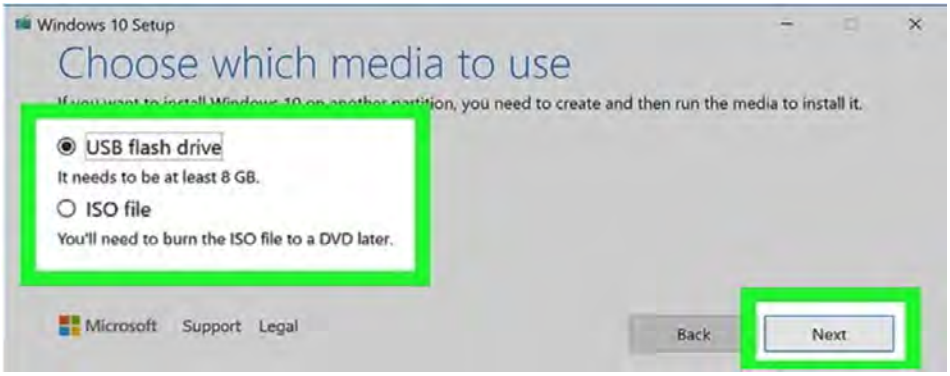
(4) Со следниот прозорец ја избираме опцијата за креирање на инсталациски модул.



Слика 3.1. Почеток на креирање на инсталациски модул за Windows

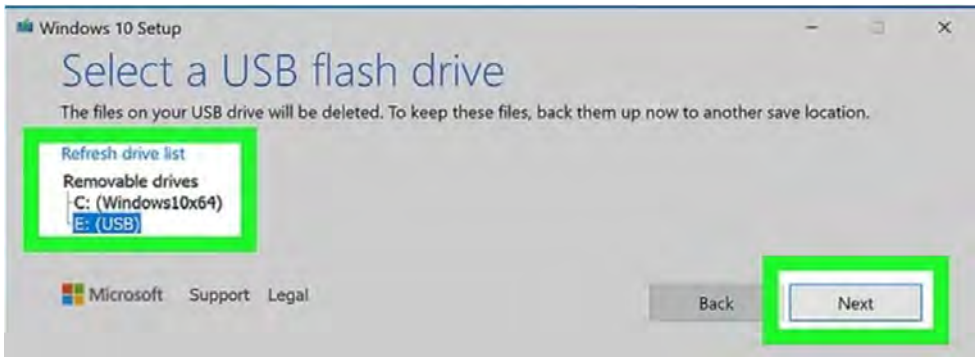
(5) Избираме јазик, оперативен систем и архитектура

(6) Избираме вид на модул и притискаме Следно (анг. Next)



Слика 3.2. Изборна мемориски модул, USB или DVD

(7) Доколку има повеќе USB мемориски модули избираме еден.



Слика 3.3. Избор на USB модул за сочувување на инсталацијата

(8) Чекаме да се инсталираат документите и на крај притискаме Заврши (анг. Finish).

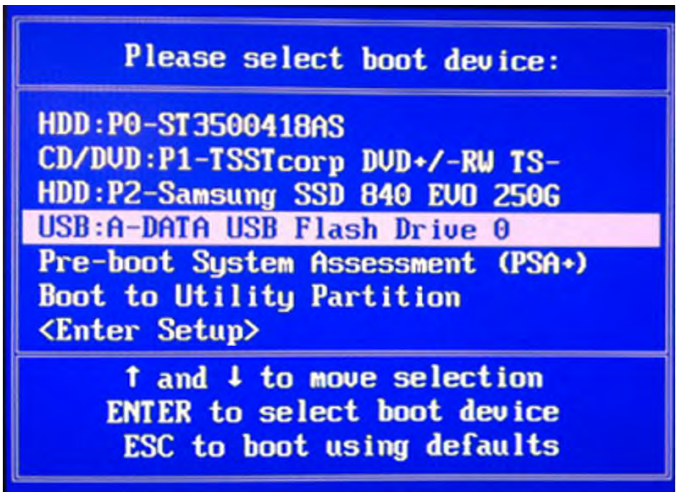
Коментар:_____

_____

5. Повикување на инсталициониот USB мемориски модул

(1) По приклучувањето на модулот и вклучувањето на компјутерот ја отвораме BIOS програмата, каде што се наоѓа менито за избор на уред што ја содржи инсталацијата. За таа цел, непосредно по вклучувањето на компјутерот треба да го притискаме копчето **F1, F2, F10, F12 или Del**. Ова копче е различно за различни производители на компјутери.

(2) Од горното мени ја избираме опцијата Boot и потоа со помош на тастатурата се движиме горе-долу, го избираме уредот и притискаме Enter.



Слика 3.4. Избор на уред за инсталација

(3) По притискањето на Enter, се отвора прозорецот за инсталација (анг. Windows Setup Wizard).

Коментар:_____

_____

_____

6. Инсталација на Windows 10 оперативен систем

(1)

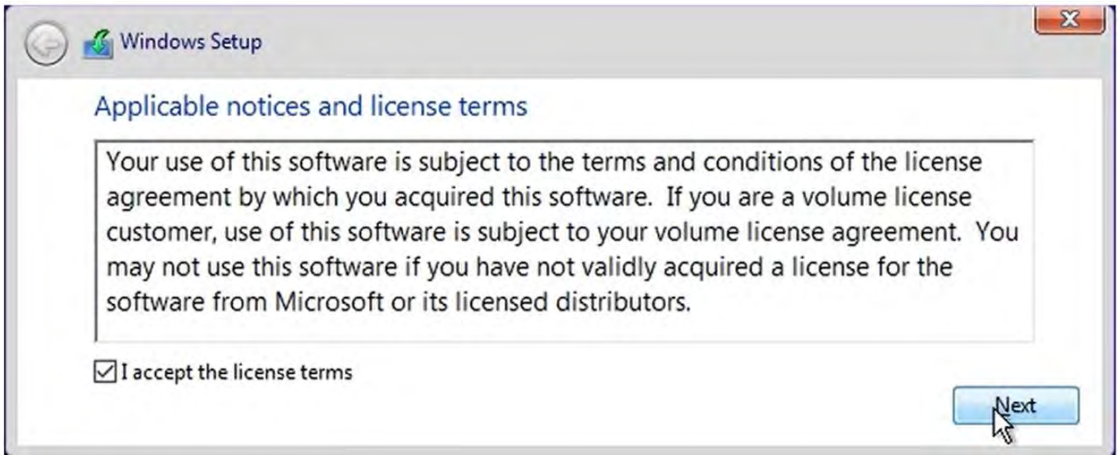
Слика 3.5. Избор на стандарден јазик, време и јазикот за внесување податоци

Со првиот прозорец се избира јазикот, времето и форматот на датумот, јазикот на тастатурата за внесување податоци (US или UK). Притискаме Next и се отвора вториот прозорец.

(2) Со изборот на опцијата „Инсталирај сега“ (Install Now), започнува процесот на инсталација.

(3) На почетокот од инсталацијата се отвора прозорец во којшто се бара од корисникот да го внесе таканаречениот **клуч на производот**. Се работи за код од 25 цифри со кои се потврдува лиценцата. Овој код го испорачува Microsoft, заедно со доставата на оперативниот систем, или може да биде испратен преку е-пошта, доколку програмата за инсталација е преземена од интернет.

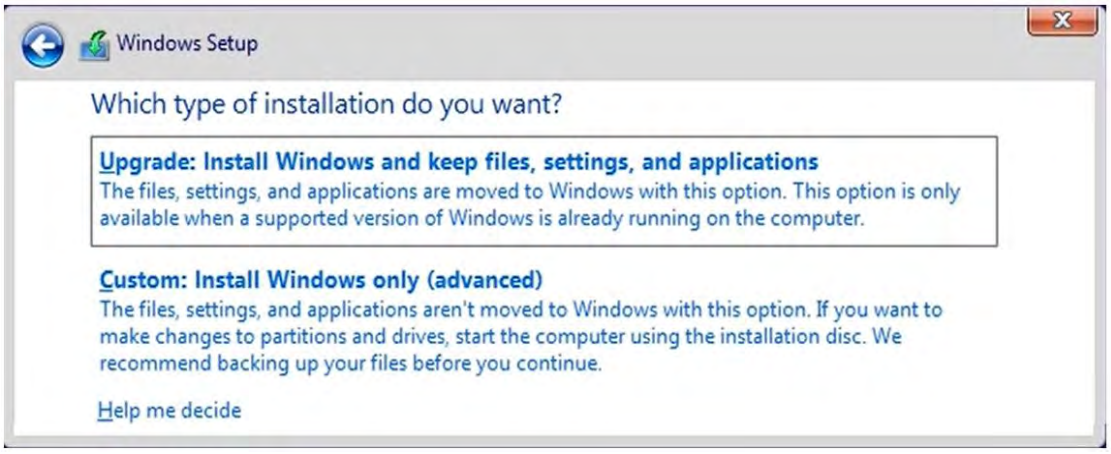
(4) Третиот прозорец е за прифаќање на условите што се поврзани со лиценцата на оперативниот систем.



Слика 3.6. Прифаќање на условите за лиценца

Условите ги прифаќаме со штиклирање на квадратчето од левата страна и притискање Next.

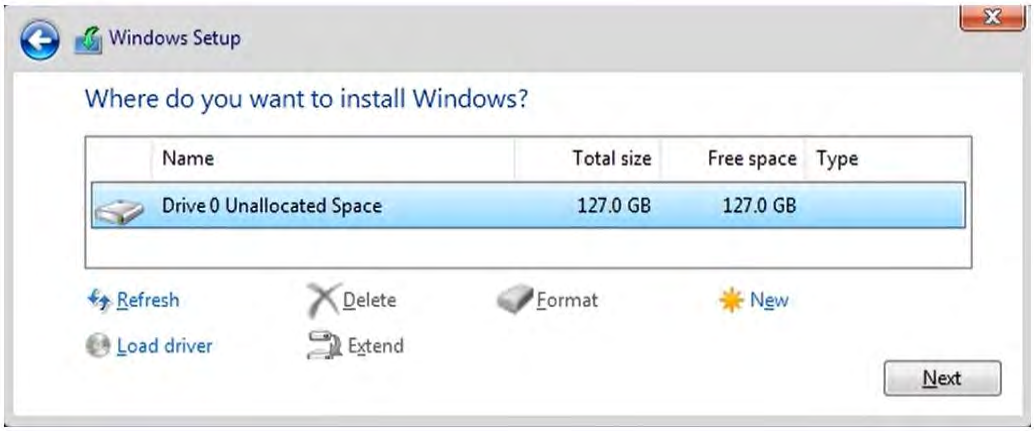
(5)



Слика 3.7. Избор меѓу надградба или нова инсталација на Windows

Четвртиот прозорец е за избор на вид на инсталација. Постојат две можности: Upgrade или Custom. Првата опција значи надградба на веќе постоечкиот оперативен систем Windows. Оваа инсталација е бесплатна, не бара клуч на производот, при што не се бришат корисничките податоци. Со втората опција се бришат сите податоци сочувани во компјутерот и ова значи нова инсталација.

(6) Петтиот прозорец е за избор на дискот на којшто ќе го инсталираме оперативниот систем.



Слика 3.8. Избор на партиција од хард-дискот

Со притискање на копчето New може да се креира нова партиција на дискот, при што треба да се одреди и капацитетот на новата партиција. Во првата партиција се сместува оперативниот систем, а во втората се чуваат корисничките податоци. Да нагласиме дека со креирањето на втора партиција, автоматски се формира уште една системска партиција. Да се потсетиме дека партициите можат да бидат од различен тип, MBR или GPT и истите беа објаснети во 2.2. Практична вежба за инсталација на составни делови на персонален компјутер, во делот бр. 10 кога зборувавме за иницијализација на хард-диск. Логично е ако компјутерот располага со UEFI програма партициите да бидат GPT. Ако компјутерот има BIOS програма тогаш можни се само MBR партиции и ако хард-дискот има капацитет над 2TB тогаш сè што е над 2TB нема да биде искористено.

(7)

Слика 3.9. Почеток на инсталација на Windows

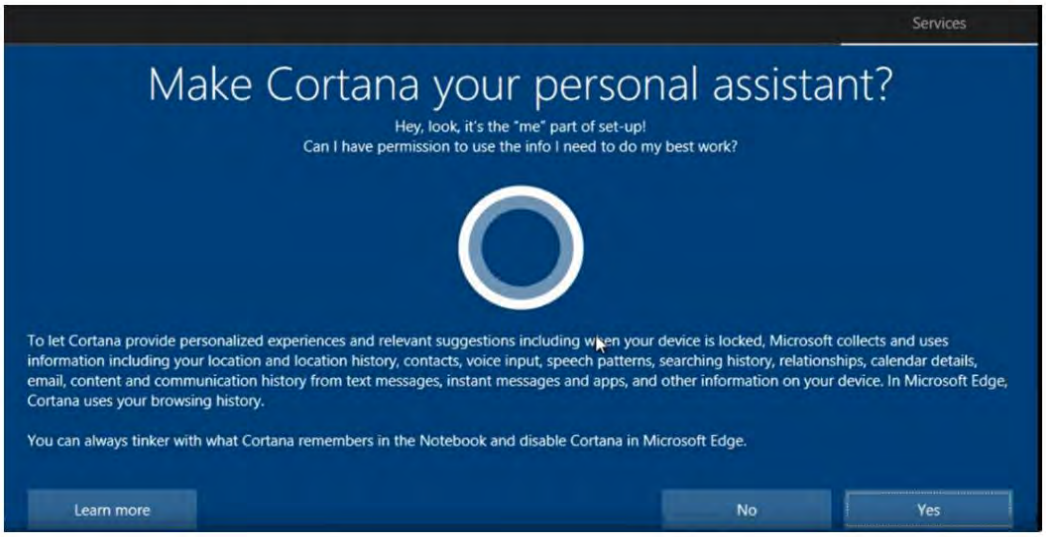
Започнува инсталацијата на оперативниот систем и откако истата ќе заврши компјутерот ќе се ресетира.

(8) Потоа следуваат неколку прозорци со коишто се врши нагодување на повеќе работи: земја или регион и јазик на тастатурата за внесување податоци.

(9) Прозорецот за поврзување во мрежа може да се прескокне и овој избор ќе го направиме по инсталацијата на Windows 10.

(10) Со следните два прозорци се дефинираат корисничкото име, лозинката и поим (анг. hint) кој треба да ни помогне да ја запаметиме лозинката. Без лозинка не ќе можеме да ја отвориме работната површина на оперативниот систем. Кај одредени инсталации, покрај корисничкото име, потребно е да се наведе и профилот на Microsoft, а доколку корисникот нема таков профил, може да го креира. Кај оперативниот систем Windows 10 има опција корисничките податоци и датотеки автоматски да се сочувуваат во облак (анг. OneDrive). На таков начин, корисникот ќе има пристап до своите податоци и преку мобилен телефон.

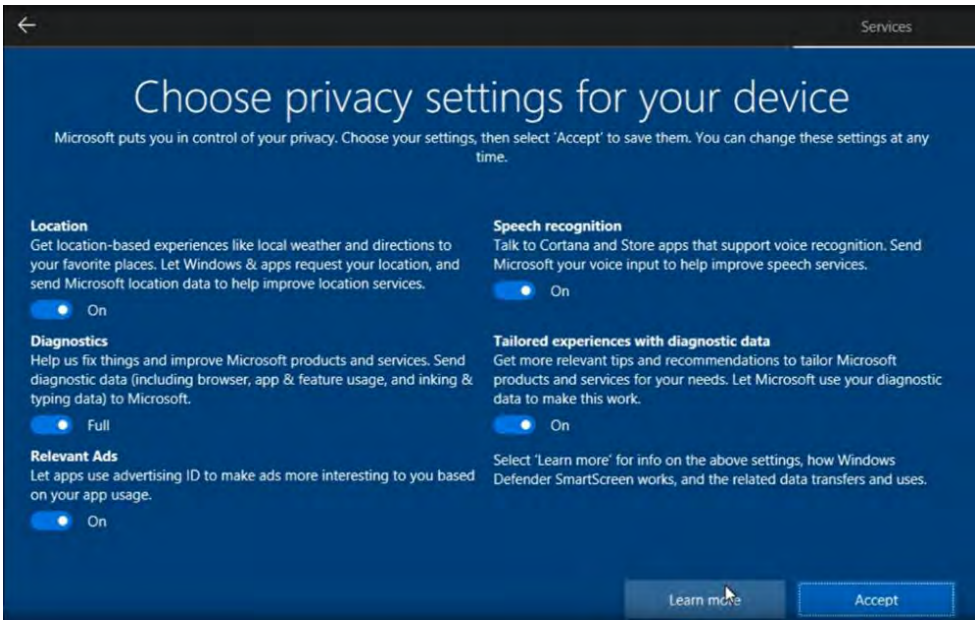
(11) Претпоследниот прозорец е за избор на апликацијата Кортана, како личен асистент.



Слика 3.10. Избор на апликација за личен асистент

Оваа апликација е новитет кај Windows 10 и се користи за пребарување на интернет и за управување со хардверските компоненти, но не е достапна за сите земји.

(12) Последниот чекор од инсталацијата е поставување приватни ставки, како одредување локација, дијагностика на софтверот и апликациите, препознавање на говор и слично.



Слика 3.11. Поставување на можности за одредување локација и дијагностика

Инсталацијата е завршена и треба да се почека неколку минути додека да се отвори работната површина на Windows 10.

Коментар:_____
_____
_____

3.3. Практична вежба: Конфигурација на Windows 10 оперативен систем

Цел на вежбата: Со текот на времето, перформансите на оперативниот систем се намалуваат од најразлични причини: некомпатибилност, вируси, апликации кои не се користат итн. **Цел на оптимизацијата е да се зголеми брзината на оперативниот систем**. Времето на реализација на вежбата изнесува еден наставен час. Подолу се дадени неколку начини за остварување на оваа цел, нивното значење и чекорите за реализација.

(1) Стартирачки апликации апликации (анг. Startup) ⟶ Startup апликациите автоматски се активираат при вклучување на компјутерот и го зголемуваат времето на чекање. Поради тоа е потребно да се редуцира нивниот број.

- Settings
- Apps
- Startup
- Sort by: Startup impact
- Избор на апликации
- Ресетирање на компјутерот

(2)	**Неискористени апликации** →	Оперативните системи Windows не даваат известувања (нотификации) за оптовареноста на системот, па поради тоа самите корисници треба да ги отстранат програмите што не ги користат.	• Settings • Apps • Apps & feature • Избор на апликации • Uninstall
(3)	**Ослободување на меморија во хард дискот** →	Вообичаено откако ќе се постигне 70% искористеност на меморискиот простор во хард дискот. Во контролниот панел (Control Panel) оваа можност е под категоријата системско-административни алатки (System-Administrative Tools).	• Settings • System • Storage • Temporary files • Избор на датотеки • Removes files
(4)	**Организација на податоци во хард дискот (анг. defragmentations)** →	Кога на хард дискот ќе се намали слободниот простор, тогаш податоците се запишуваат на различни места во хард дискот, со што се зголемува времето на вчитување. Со дефрегментизацијата, овие податоци се поставуваат еден до друг и се ослободува мемориски простор.	• Settings • System • Storage • More settings • Optimize Drivers • Избираме диск (C: или D:) • Optimize
(5)	**Антивирусни програми** →	Доколку оперативниот систем работи со намалена брзина, потребно е да се повика и да се активира антивирусната програма. Доколку компјутерот не располага со посебна антивирусна програма, може да се искористи програмата Windows Defender.	• Control Panel • Security • Windows Defender • Scan
(6)	**Надградба на оперативен систем** →	Погрешно е кога корисниците ја исклучуваат автоматската надградба бидејќи надградбите се многу важни за сигурноста	• Settings • Update & Security • Windows Update • Check for Update

		на системот. Драјверите на уредите можат да се надградуваат во категоријата System-Device manager.	• Ресетирање на компјутерот
(7)	**Зголемување на виртуелните страници**	Виртуелната меморија значи проширување на примарната меморија преку секундарната. Имено, RAM-меморијата зема податоци од хард-дискот во форма на виртуелни страници со фиксна големина и нив ги вметнува во виртуелна рамка. Ако се зголеми капацитетот на виртуелните страници, ќе се намали бројот на трансфери меѓу RAM и хард-диск.	• Control Panel • System • Advanced System (Performance) • Settings • Advanced • Change (Virtual memory)

Оперативниот систем Windows 10 нуди многу можности и не можеме сите да ги објасниме и проучиме. Но, секогаш треба да се истражува и да се настојува самостојно да се конфигурираат компјутерските системи.
Коментар:_____
_____
_____

3.4. Преинсталација на Windows 10 оперативен систем

1. **Цел на вежбата:**

Поради недоволна заштита од вируси, преоптовареност со многу апликации или преголема искористеност на капацитетот на хард-дискот компјутерот полека ја губи брзината на работа. Едно од наједноставните решенија е преинсталација на оперативниот систем. Постојат повеќе можност за преинсталација и цел на оваа вежба е да се запознаеме со истите.

2. **Време за реализација:** 1 наставен час

3. **Преинсталација со губење на корисничките податоци (документи, слики и видеа) и инсталираните апликации**

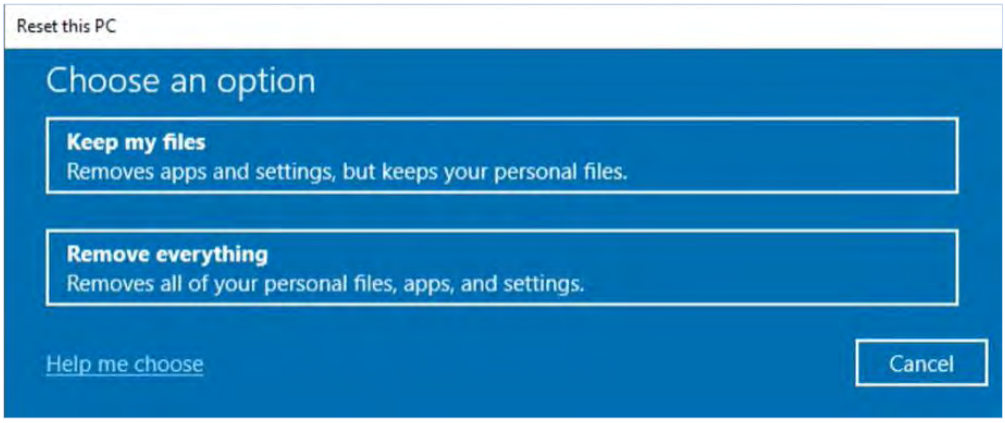
Не постои разлика меѓу постапката на овој вид на преинсталација и инсталација. Откако ќе креираме инсталациски мемориски модул истиот можеме да го користиме неограничен број пат за преинсталација на Window 10

операцискиот систем. За да не дојде до губење на податоците, најдобро е да направиме резервна копија на корисничките документи пред да започнеме со преинсталација. Нив можеме да ги сочуваме во надворешна меморија или да креираме посебна партиција на хард-дискот преку функцијата Disc Management. Во текот на преинсталацијата, овој дел (партиција) од хард-дискот нема да биде форматиран.

4. **Ресетирање на Windows 10 оперативниот систем така што корисничките податоци ќе бидат сочувани, но инсталираните апликации ќе бидат избришани.**

Предност на оваа постапка е што не е потребен USB мемориски модул и целата постапка се извршува автоматски. Ресетирањето на оперативниот систем бара слободен мемориски простор на хард-дискот од 3-4GB. Доколку не постои толку слободен простор ќе добиеме известување.

За ресетирање на Windows 10 оперативниот систем ги избираме опциите Settings→Update&Security→Recovery→Reset this PC (Get Started). Се јавува прозорец за избор на опција, дали корисничките документи ќе бидат сочувани или сè ќе биде избришано (слика).



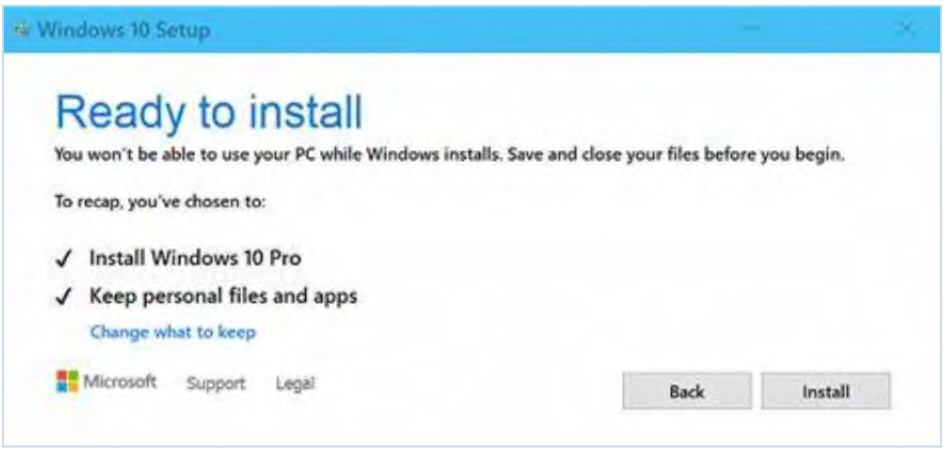
Слика 3.12. При ресетирање на Windows 10 може да се сочуваат корисничките документи

После кратката подготовка се отвора прозорец за потврда на ресетирањето. Во овој прозорец постои можност да се излистаат апликациите кои ќе бидат избришани. После тоа започнува ресетирањето.

5. **Преинсталација на Windows 10 оперативниот систем преку инсталациски USB мемориски модул така што корисничките податоци и инсталираните апликации ќе бидат сочувани.**

Во практичната вежба 3.2. Инсталација на Windows 10 оперативен систем се запознавме со креирањето на инсталациониот USB мемориски модул. За реализација на оваа вежба потребно е инсталацискиот модул да го поврземе со компјутер. Го селектираме модулот, притискаме на десен клик и ја одбираме опцијата **Mount** и на таков начин, содржината на меморискиот модул ќе стане видлива. Во папката **This PC** го селектираме новиот документ, го притискаме

десниот клик и ја избираме опцијата **Open in new window**. Се отвора нов прозорец и ја избираме опцијата **Setup.exe.** Започнува подготовка за ресетирање на оперативниот систем. Во прозорецот „**Get important updates**“ ја одбираме опцијата **Not right now,** со цел да биде поедноставен процесот за преинсталацијата. Ги прифаќаме условите за лиценца.
После неколку минути чекање, се појавува прозорецот „**Ready to install**“ што во превод значи „Спремни за инсталација“. Треба да посветиме поголемо внимание на овој прозорец. Опциите **Install Windows 10 Home/Pro** и **Keep personal files and apps** треба да бидат штиклирани. Доколку не ја гледаме опцијата Keep personal files and apps, истата може да ја добиеме преку опцијата **Change what to keep.**



Слика 3.13. Заштита на личните документи и апликации од бришење

На ваков начин вршиме заштита на личните документи и заштита на апликациите. Со притискање на копчето Install започнува преинсталирањето на оперативниот систем Windows 10.
Коментар:_____
_____
_____

3.5. Практична вежба: Инсталација на оперативен систем Ubuntu 17 Linux во виртуелна машина

1. **Цел на вежбата:**

Во теоретскиот дел на модуларната единица Инсталација на оперативен систем и други стандардни програми на персонален компјутерски систем веќе се запознавме со многуте предности и можности кои ги нуди Linux оперативниот систем. Цел на оваа вежба е употреба на два оперативни системи на ист компјутер, со тоа што за инсталацијата на Linux оперативниот систем ќе користиме виртуелна машина. **Ubuntu е многу популарен дистрибутер** на

оперативните системи Linux. За преземање на инсталацијата на оперативниот систем, потребно е да се отвори www.ubuntu.com на овој дистрибутер и потоа со кликнување на Download, ја избираме десктоп-верзијата.

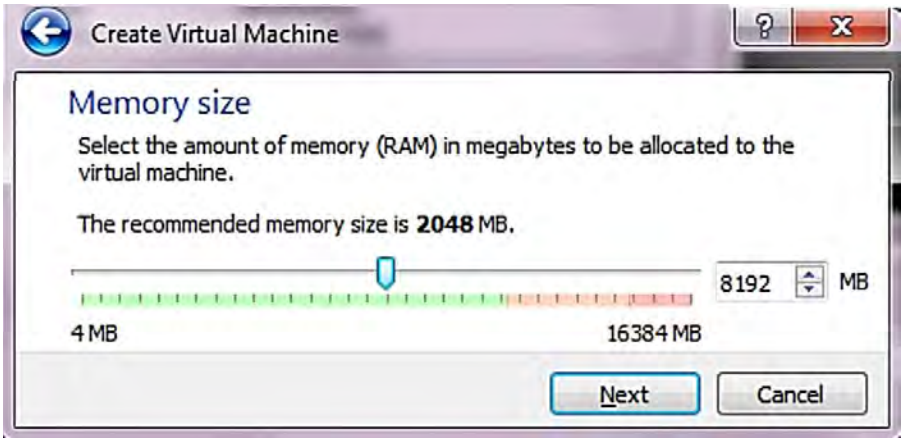
2. **Време за реализација:** 2 наставни часа

3. **Упатство за креирање на виртуелна машина**

VirtualBox е апликација за креирање, конфигурација и администрација на виртуелни машини. Виртуелна машина претставува компјутерски системски гостин (guest) што ги користи ресурсите на компјутерот-домаќин (host). На компјутерот-домаќин ќе биде инсталиран оперативниот систем Windows, а на виртуелната машина, односно guest-компјутерот ќе биде инсталиран оперативниот систем Ubuntu. На таков начин, **на ист компјутер** ќе имаме инсталирано **два оперативни система** без да си влијаат еден на друг.
Програмата за инсталирање виртуелна машина се презема од сајтот www.virtualbox.org, со кликнување на опцијата Download и избор на линкот Windows host. Откако ќе ја инсталираме апликацијата VirtualBox, ја отвораме и се кликнува на опцијата New за креирање нова виртуелна машина.

Слика 3.14. Креирање виртуелна машина

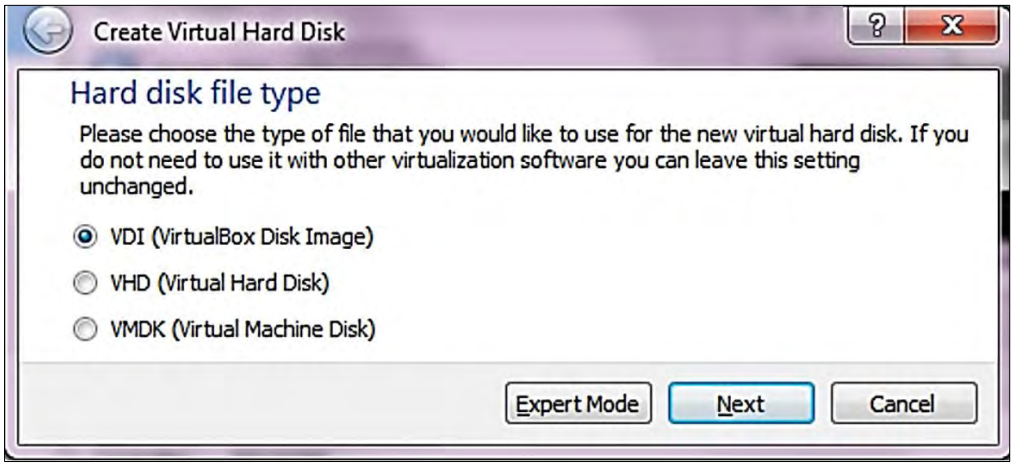
Се отвора прозорецот **Create Virtual Machine**. Во полето **Name** се внесува името на виртуелната машина (на пример Ubuntu), во полето **Type** се внесува видот на оперативниот систем што ќе се инсталира во виртуелната машина (Linux) и во полето **Version** се внесува верзијата на оперативниот систем (32-битен или 64-битен). На крајот се кликнува на Next.



Слика 3.15. Определување на капацитетот на RAM-меморијата за виртуелната машина

Следниот прозорец служи за определување на капацитетот на RAM-меморијата за виртуелната машина (слика). Најдобро е да се следат препораките, но треба да се земе предвид и вкупниот капацитет на меморијата.

Следните три прозорци служат за креирање **виртуелен хард-диск**, дефинирање на неговиот капацитет и определување на начинот на управување со меморискиот простор.



Слика 3.16. Избор на вид на хард-диск

Слика 3.17. Избор на капацитет на хард-диск

Креирањето на виртуелната машина го потврдуваме со кликање на Create. Во овој последен чекор имаме можност да ги промениме името и локацијата на виртуелната машина.

Коментар:_____

_____

_____

4. Инсталација на оперативниот систем Ubuntu

За инсталација на оперативниот систем Ubuntu кликаме на опцијата Start во прозорецот VirtualBox Menager, прикажан на слика 3.17.

Слика 3.18. Почеток на инсталација на оперативен систем Ubuntu

Во прозорецот Select start-up disk, со кликање на жолтата папка (фолдер) ја наоѓаме локацијата на којашто се наоѓа инсталацијата на оперативниот систем, ја избираме и кликаме на копчето Start.

Во прозорецот Welcome, дадени се **два начина за инсталација** на оперативниот систем Ubuntu. Со начинот Try Ubuntu, оперативниот систем може да се користи без инсталација. Со овој начин, всушност, го инсталираме оперативниот систем на виртуелниот хард-диск којшто претходно е креиран и пред инсталацијата е празен, така што нема бришење на постоечкиот оперативен систем.

Слика 3.19. Избор за начин на инсталација на оперативен систем Ubuntu

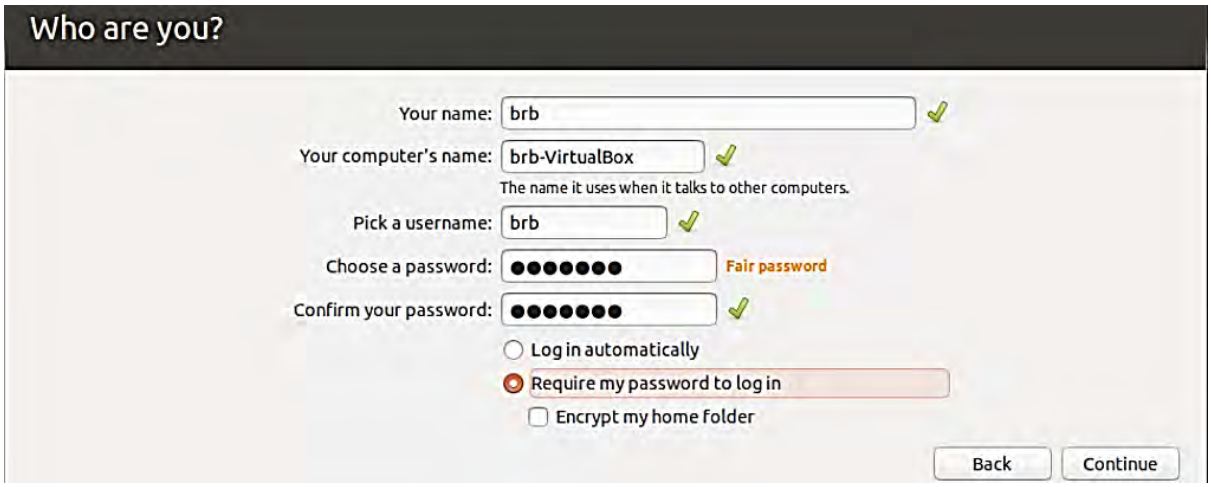
Во прозорецот **Installation type**, со кликање на копчето **Install Now започнува инсталацијата**. Опцијата **Erase disk and install Ubuntu** е однапред обележана, но веќе објаснивме дека бидејќи се работи за виртуелен хард-диск, нема да дојде до бришење и губење на податоците.

Слика 3.20. Поради виртуелната машина, нема да дојде до бришење на постоечкиот оперативен систем

Следниот прозорец е Preparing to Install Ubuntu и во него постои можност да се избере автоматска надградба на оперативниот систем и инсталација на дополнителен мултимедијален софтвер за мултимедија. На крајот кликаме на копчето Continue.

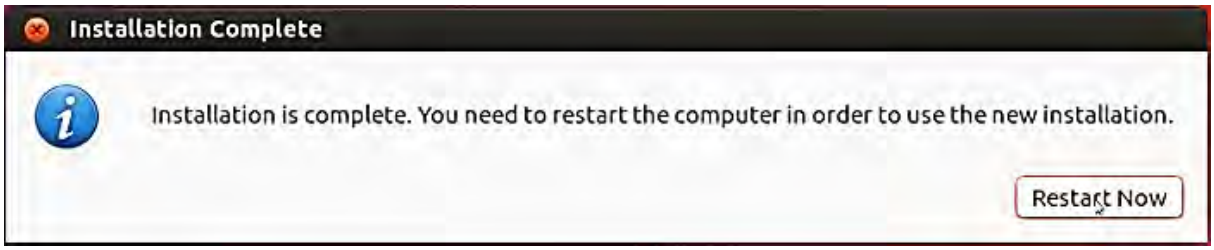
Следните два прозорци се за избор на земја или регион и јазик на тастатурата за внесување податоци, слично како и при инсталацијата на оперативниот систем Windows 10.

Следува прозорец за определување на корисничкото име и лозинката. Со кликнување на Continue, започнува инсталацијата.



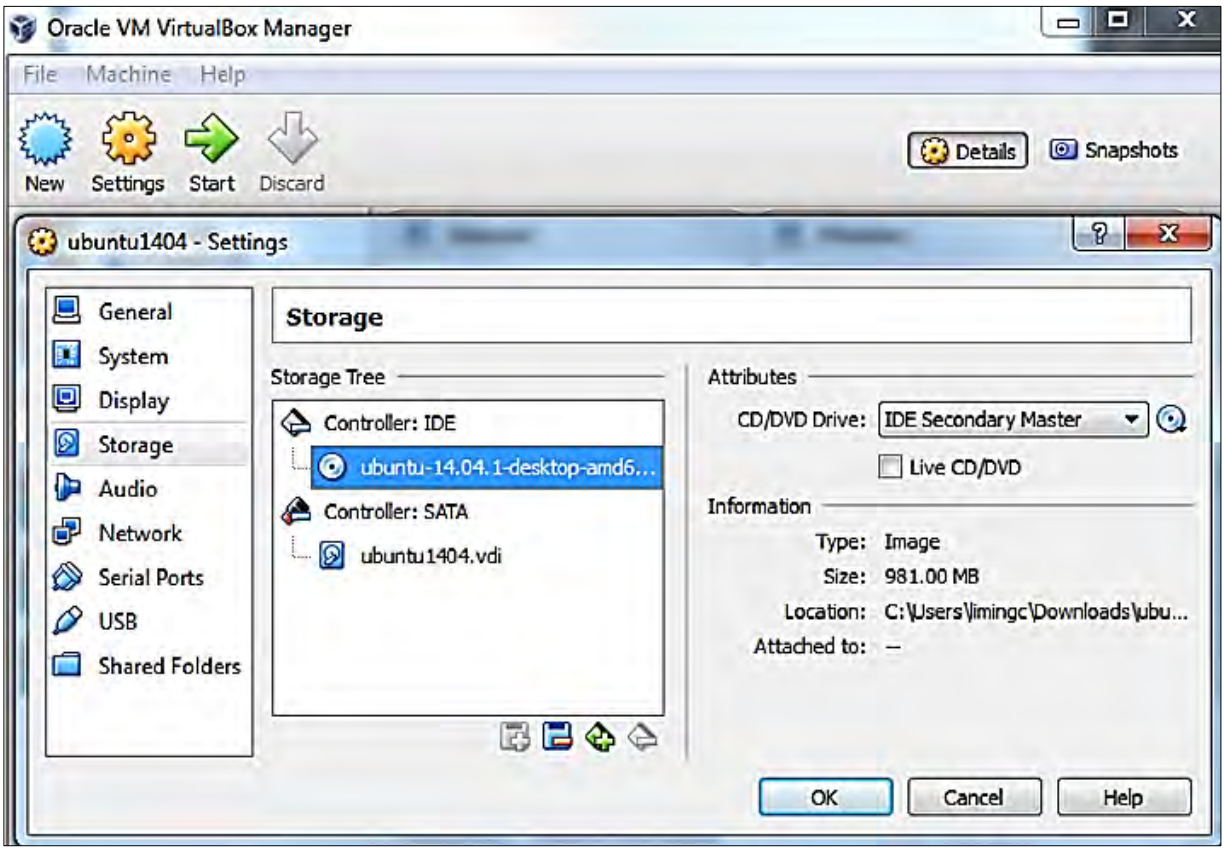
Слика 3.21. Избор на корисничко име и лозинка

По информацијата за успешната инсталација, го рестартираме системот и ја добиваме работната површина на Ubuntu.



Слика 3.22. Рестартирање на компјутерот за да се добие работната површина на Ubuntu

По извршената инсталација, потребно е да го избришеме инсталацискиот документ. За таа цел се избира оперативниот систем Ubuntu во виртуелната машина, се притиска на опцијата Storage во категоријата Settings. Потоа во делот Storage Tree треба да се избере инсталацискиот документ и да се притисне (анг. click) Remove selected storage attachment. На крајот го притискаме копчето OK.



Слика 3.23. Бришење на инсталациски документ

Коментар:_____

_____

_____

3.6. Практична вежба: употреба на антивирусна програма Avira

1. Цел на вежбата

Веќе истакнавме дека Windows Defender е системска антивирусна програма, но заради поголема заштита поголем број на корисници користат дополнителни антивирусни програми. Цел на оваа вежба е инсталација и деинсталација на антивирусната програма Avira и примена на софтверските алатки во склоп на истата.

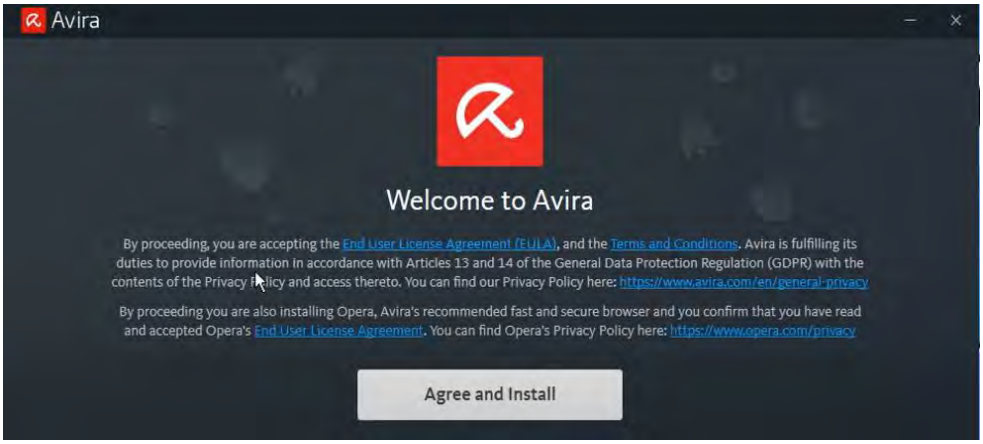
2. Време за реализација: 1 наставен час

3. Преземање и инсталација на Avira антивирусна програма

(1) Доколку сакаме да ја користиме оваа програма, нејзината бесплатна верзија можеме да ја преземеме од официјалната веб-страница https://www.avira.com со притискање на копчето **Free download.** Ќе се појави известување за завршено преземање на инсталациониот документ, најчесто во папката за преземање (анг. Download).

Слика 3.24. Превземање на инсталациски документ за Avira програма

(2) За да започне инсталацијата, потребно е документот да се отвори, избираме **Run,** притискаме **Agree and Install** и чекаме 4-5 минути за да се заврши процесот. После инсталацијата се појавува икона со логото на антивирусната програма Avira и истата е спремна за употреба.



Слика 3.25. Инсталација на Avira антивирусна програма

4. Употреба на софтверски алатки на Avira антивирусна програма

(1) Софтверските алатки во составот на Avira антивирусната програма се поделени во три категории: Безбедност (анг. Security), Приватност (анг. Privacy) и Перформанси (анг. Performance). Во категоријата „Безбедност“ спаѓа: скенирање на вируси, ниво на заштита, корекција и бришење на сомнителни документи, надградба на апликацијата (анг. upgrade), заштита од напади на вируси (анг. Firewall). Со притискање на опцијата скенирање на вируси (анг. **Virus scan)** започнува проверката. Постојат три опции: за целосна, брза проверка или закажување на скенирање. Доколку нашиот компјутер не е безбеден, ќе се појави известување за опасност и потребно е да се притисне копчето **Fix problem**.

Слика 3.26. Откривање и отстранување вируси со програмата Avira

(2) Категоријата „Приватност“ ги содржи следниве алатки: сигурност при пребарување на интернет (анг. Browser Safety), VPN пристап, бришење на податоци кои ја загрозуваат безбедноста при работа, управување со лозинки, спречување споделување на податоци преку Windows оперативниот систем или некоја друга апликација.

(3) Категоријата „Перформанси“ овозможува: оптимизација (ослободување на мемориски простор и подобрување на брзината на работа), оптимизација на Startup апликациите, заштеда на батериите, надградба на драјверите за уредите, пронаоѓање на дупликати-документи.

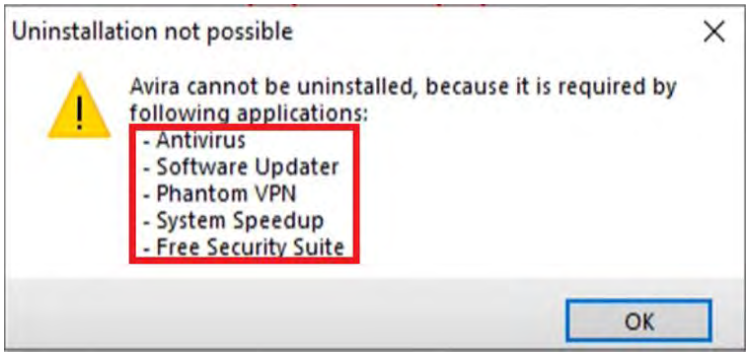
5. Деинсталација на Avira антивирусна програма

Антивирусните програми можат да го оптоварат компјутерот и да ја намалат неговата брзина на работа. Поради тоа не се препорачува истовремена употреба на две антивирусни програми. Доколку сакаме да ја смениме антивирусната програма, тогаш истата треба да се деинсталира.

(1) За таа цел се избира опцијата Settings → Apps→Apps and Features и во листата со програми ја избираме програмата Avira. Со притискање на

десниот клик се јавува нов прозорец, притискаме на Uninstall/Change и избираме Uninstall.

(2) Можно е да добиеме известување дека Avira антивирусната програма не може да се деинсталира бидејќи истата е поврзана со други потпрограми



Слика 3.27. Потпрограми поврзани со Avira антивирусната програма

За да ја овозможиме деинсталацијата на Avira програма потребна претходна деинсталација на сите потпрограми поврзани за истата. Во листата на програми, во Apps and Features, ги наоѓаме истите и со притискање на десен клик ја избираме опцијата Uninstall.

(3) Во текот на самата деинсталација, потребно е да потврдиме неколку известувања. Го рестартираме компјутерот и проверуваме во Program and Features дали е избришана програмата Avira.

Коментар:_____
_____
_____

3.7. Практична вежба: Инсталација и употреба на програма за компресија на датотеки

1. **Цел на вежбата:**

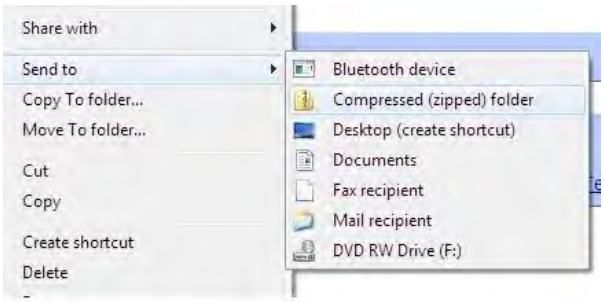
Цел на оваа вежба е компресија и декомпресија на датотека со повеќе документи, со примена на различни програми за компресија. После извршената компресија потребно е да се направи споредба на големината на компресираните датотеки и времето на компресија и декомпресија.

2. **Време за реализација:** 1 наставен час

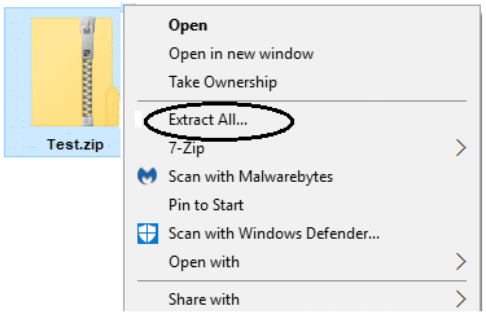
3. **Компресија и декомпресија на датотека со програма за компресија која е дел од самиот Windows оперативен систем**

Избираме една датотека. Со притискање на **десен клик** се отвора паѓачко мени, избираме **Send to** и опцијата **Compressed (zipped) folder**. Се креира нова датотека со екстензија .zip и тоа на истото место каде што се наоѓала и

изворната датотека. Изворната и компресираната датотека ќе имаат исто име, но ќе се разликуваат по екстензијата.



Слика 3.28. Компресија на датотека



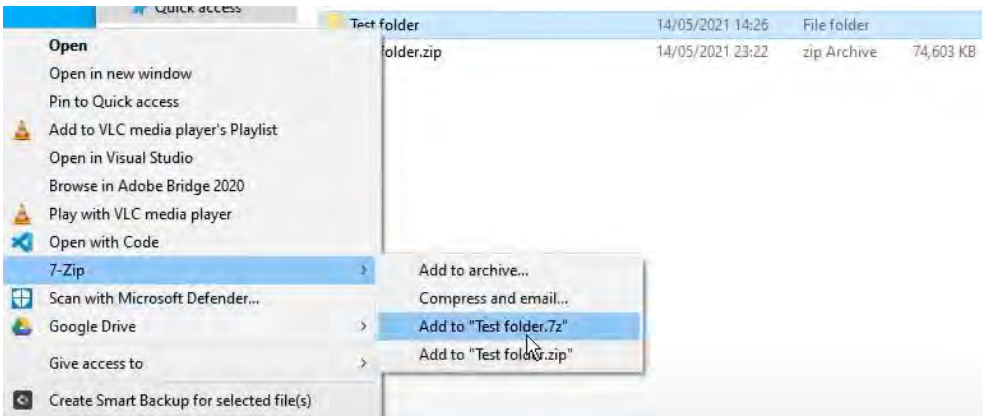
Слика 3.29. Декомпресија на датотека

Додека трае компресијата треба да се измери времето потребно за извршување на постапката. После извршената компресија ја проверуваме големината на компресираниот документ со притискање на десен клик и избор на опцијата Properties.

Доколку сакаме да декомпресираме некој документ или датотека, тогаш вршиме селекција со лев клик, а со десен клик избираме **Extract All.** Се отвора нов прозорец, ја избираме дестинацијата на новиот декомпресиран документ со опцијата Browse и уште еднаш притискаме на Extract.

4. Компресија на датотека со 7-ZIP програма

Бесплатната програма 7-Zip може да ја преземеме од официјалната веб-страница www.7-Zip.com. На располагање се две верзии, 32 и 64 битна. Се избира една од опциите и започнува преземањето на документот за инсталација.



Слика 3.30. Примена на 7-Zip програмата

После преземањето на документот, со двоен клик на истиот, избираме Install и започнува неговата инсталација. Ја избираме истата датотека како во претходниот чекор и со десен клик ја избираме програмата 7-Zip и во ново паѓачко мени опцијата Add to archive (слика 3.30). Се отвора нов прозорец кој

дава можност да го избереме форматот, степенот на компресија и методата на компресија. Најчесто користени методи се LZMA или LZM2. Откако ќе го притиснеме копчето ОК започнува компресијата, го мериме времето на самата постапка и ја проверуваме големината на новодобиената компресирана датотека со екстензија .7zip.

5. Споредба на добиените резултати

Во табелата број 3.1. ги внесуваме добиените резултати. Табелата може да се прошири со нови програми за компресија како што се WINZIP, WINRAR, PeaZip или BandZip со кои се запознавме во наставната единица 3.3.2. Програми за компресија на датотеки од теоретскиот дел на учебникот.

Програма за компресија	Време на компресија (сек.)	Големина на компресирана датотека	Степен на компресија	Време на декомпресија
Windows програма за компресија				
7-Zip				

Табела 3.1. Споредба на различни програми за компресија

Во истата наставна единица 3.3.2. Програми за компресија на датотеки, од теоретскиот дел, е објаснет и начинот за пресметка на степенот на компресија.
Коментар:_____
_____
_____

4. Практични вежби за работа со Arduino микрокомпјутер на плоча и негово програмирање

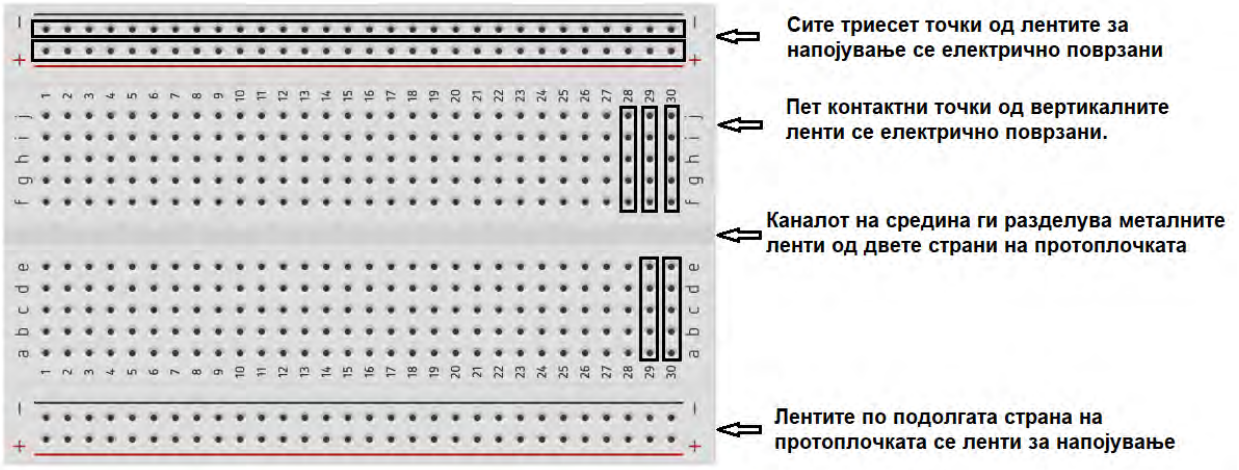
4.1. Мерки за заштита и безбедност при работа со Arduino микрокомпјутер на плоча

Практичните вежби се изведуваат со примена на Arduino Uno R3 микрокомпјутерот на плочка кој работи со нисконапонски и нискострујни сигнали. Со цел истиот да се заштити од оштетување треба да се преземат одредени мерки на претпазливост.

- Пред почетокот на секоја монтажа треба да се исклучи Arduino Uno R3 од неговиот извор за напојување.
- Батеријата, како извор за напојување, се поврзува со Arduino Uno R3 преку пиновите V_{in} и GND. Максимално дозволената вредност на напонот изнесува 20 V, но се препорачува употреба на батерии со напон од 9 до 13 V. При поврзувањето да се внимава на поларитетот на напонот.
- Максималниот напон за влезно-излезните пинови на Arduino Uno R3 изнесува 5,5V.
- Максималната влезна струја за еден пин на Arduino Uno R3 изнесува 40mA. Збирот на влезните струи за сите пинови не смее да биде поголем од 200mA.
- Не е дозволено кратко спојување на пиновите.
- Изводите на поларизираните компоненти да се идентификуваат и соодветно да се поврзат.
- Пред да се започне со поврзување да се пресмета јачината на струјата низ електричните компоненти и да се провери дали истата е еднаква или помала од максимално дозволената струја според техничко-технолошката документација за таа компонента.
- За поврзување на мотори или други индуктивни потрошувачи да се користат соодветни контролери (анг. motor driver)
- За ограничување на струјата низ лед диодите да се користат отпорници

4.2. Упатство за користење на протоплочка

Лемењето е постапка за изработка на електронски уреди на печатена плочка со постојан дизајн, стабилна работа и сигурни електрични контакти. Но, често пати се случува дизајнот на електричното коло да се менува со цел да се подобри функционалноста и квалитетот на уредот. Ова значи промена на начинот на поврзување на компонентите, па дури и замена на истите. Протоплочките се идеални за брзо и едноставно менување на дизајнот на електричните кола.



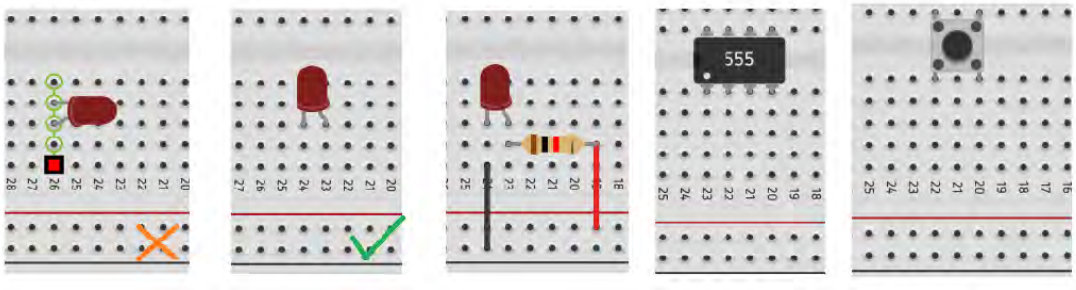
Слика 4.1. Надворешен изглед на протоплочка

Протоплочката претставува плочка изработена од висококвалитетна пластика со отвори на истата кои се всушност контактни точки или точки за поврзување. Во контактните точки се поставуваат изводите на електронските елементи или жиците за поврзување на истите. На слика 4.1. е прикажан надворешниот изглед на протоплочката. Под пластичната маска се наоѓаат проводници, поставени паралелно во неколку групи, кои вршат електрично поврзување на електронските елементи. На самите проводници се наоѓаат контакти во облик на штипки и електронските елементи механички се прицврстуваат, закачуваат на протоплочката со благо притискање на изводите.

Електронските елементи се поставуваат на вертикалните ленти, во средината на плочката. По подолгите страни на плочката, од двете страни се наоѓаат два пара ленти (хоризонталните ленти на слика 4.1.). Лентата обележана со + или црвена боја се поврзува со напојувањето, а лентата обележана со - или црна боја со заземјувањето.

За правилно поврзување на елементите многу е важно да се познава распоредот на металните ленти. **Сите контактни точки што лежат на иста лента се електронски поврзани меѓу себе и треба да се внимава при поставување на елементите да не дојде до кусо поврзување на изводите,** бидејќи тогаш низ нив нема да тече струја. На пример, на слика 4.2., првата лед диодата е неправилно поставена бидејќи нејзините електроди, анодата и катодата, се поставени во контактни точки кои припаѓаат на иста лента. Втората диода на слика 4.2. претставува правилно поврзување на лед диодата со

протоплочката. Доколку сакаме да поврземе два елементи, на пример лед диода и отпорник, тогаш нивните изводи треба да бидат поставени во контактни точки кои припаѓаат на иста лента.



Слика 4.2. Поврзување на електрични компоненти на протоплочка

На средината на протоплочката има вдлабнатина, канал кој електрично ги разделува вертикалните проводници на два дела. Ова дозволува поврзување на интегрирани кола со два реда на изводи (анг. DIP-Dual in Line). Интегрираното коло се поставува точно врз каналот така што неговите пинови се поставени на различни страни од каналот и не се електрично поврзани. Контактните точки на протоплочката се на растојание 2,45mm колку што изнесува растојанието меѓу пиновите на интегрираните кола. На средишниот канал се поставуваат и тастери. Тастерите имаат четири приклучоци, по два од две спротивни страни. **Приклучоците од иста страна не се електрично поврзани** и тие треба да се постават во контактните точки кои припаѓаат на два вертикални проводници, за да електричното коло се затвори кога тастерот ќе се притисне.

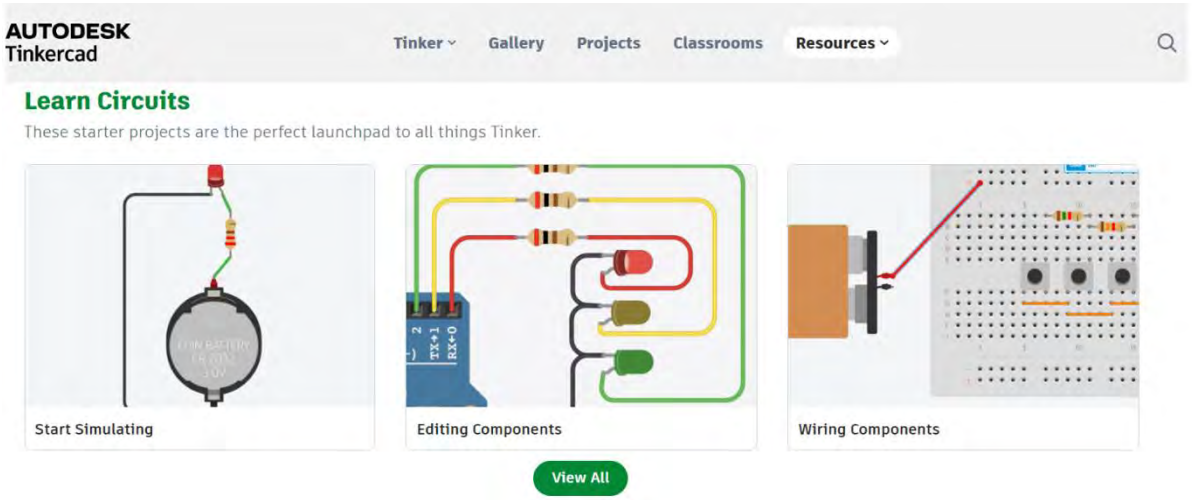
4.3. Компјутерска симулација за Arduino Uno R3

Компјутерските симулатори станаа често користена алатка при учење и работа во областа на електрониката и компјутерската техника. Тие ги поттикнуваат љубопитноста, инвентивноста, интуитивното размислување и се побезбедни за работа.

Предвидените практични вежби треба да се изведуваат во кабинет за практична настава и за тоа е потребна соодветна опрема, хардвер и електронски елементи. Дел од вежбите можат да бидат целосно реализирани и со примена на компјутерска симулација. Tinkercad е бесплатна веб-платформа која нуди одлична компјутерска симулација за Arduino Uno R3.

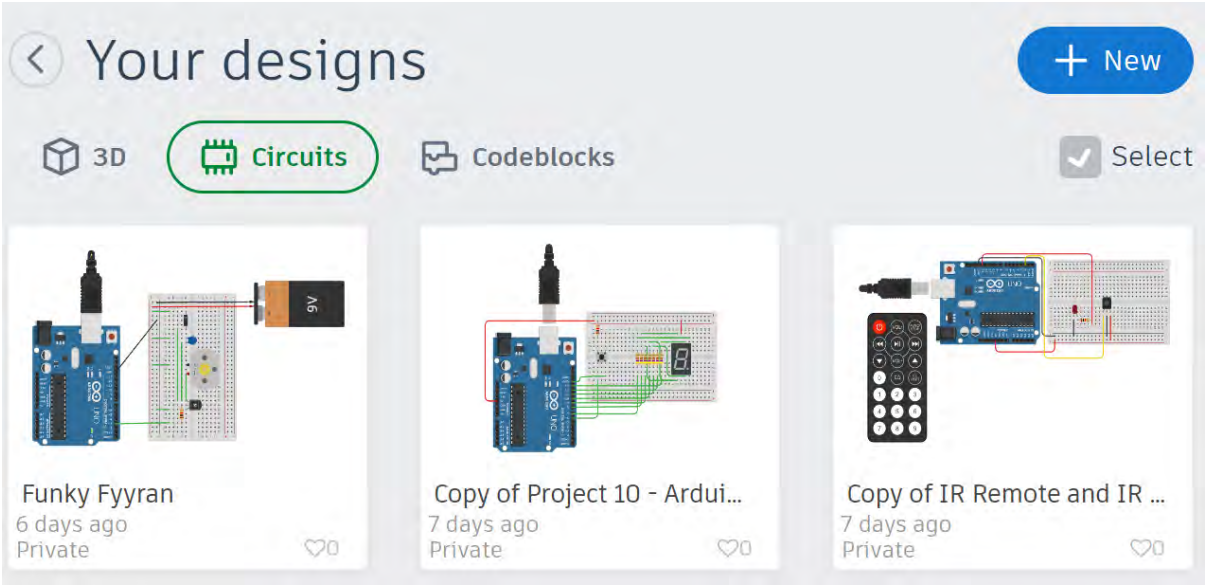
Корисникот треба да поседува Google или Apple корисничка сметка и со неа да се најави на официјалната веб-страница на платформата www.Tinkercad.com. После успешната **најава** се отвора почетната веб-страница со главното мени. Можностите се многу големи. Да напомниме дека Tinkercad платформата нуди работа со 3D дизајн, креирање на програмски кодови во програмскиот јазик Codeblocks и дизајн на електрични кола. Нам ни е потребна симулација на електрични кола со употреба со на Arduino Uno R3 и програмскиот јазик C/C++.

Со кликање на опциите Resources→Learning Center→Learn Circuit (мак. Ресурси→Центар за учење→Учење на електрични кола) се отвораат упатства за избор на компоненти, нивно поврзување, пишување на програми и симулирање на истите. Упатствата за работа се напишани во чекори и се интерактивни односно може веднаш да се применат бидејќи со отворање на упатството се отвора и работна површина и мени за избор на компоненти.



Слика 4.3. Отворање на упатства за креирање на елктрични кола со употреба на Arduino Uno R3

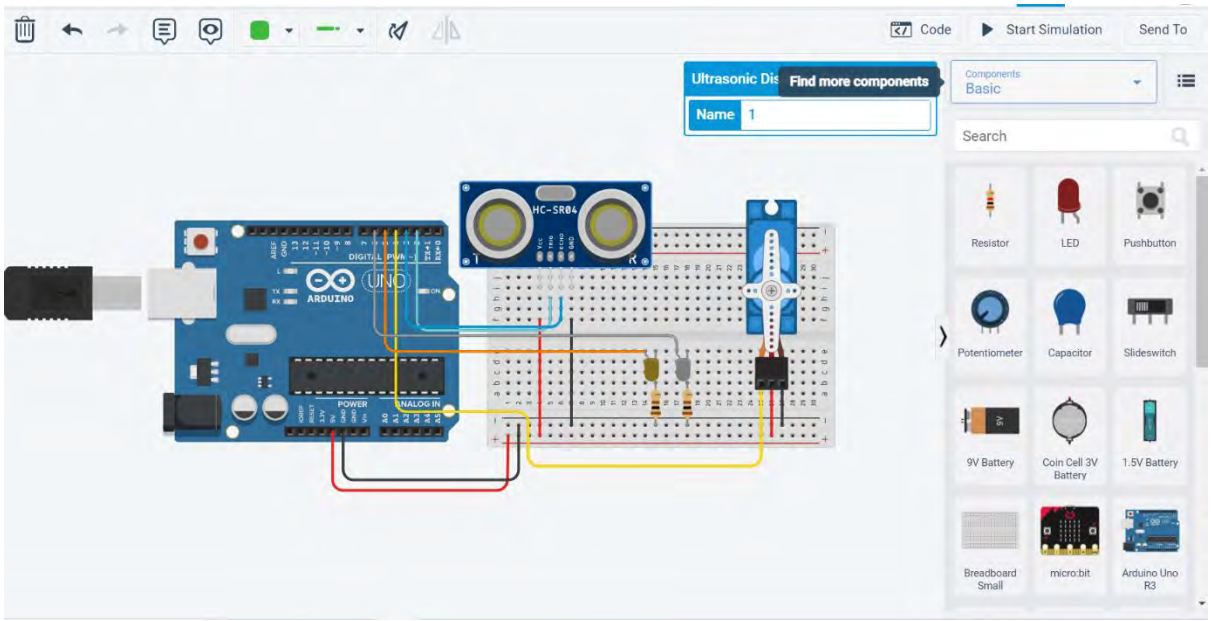
Со притискање (клик) на профилната слика и избор на опцијата Мој дизајн (анг. My Designs) се отвораат сите проекти на кои сме работеле претходно. Тие се сочувани од самата веб-платформа. Доколку сакаме да креираме ново електрично коло тогаш ја избираме опцијата Ново (анг. New).



Слика 4.4. Отворање на нови и стари дизајни на електрични кола

На слика 4.5. е прикажан изгледот на работната површина за составување на електрични кола и програмирање на Arduino Uno R3. Електричното коло се составува многу лесно, со повлекување и пуштање на електронските

компоненти, а потоа истите се поврзуваат. Програмата за работа може да ја видиме или напишеме доколку притиснеме на копчето **Code** и ја одбереме опцијата **Text.** Откако ќе го составиме електричното коло и ја напишеме програмата, го притискаме копчето **Start Simulation** и набљудуваме дали колото работи како што е предвидено. Доколку има грешки во кодот, истите ќе бидат обележани со црвена боја.



Слика 4.5. Изглед на работната површина за работа со електрични кола во веб-платформата Tinkercad

Tinkercad веб-платформата е од отворен тип и овозможува споделување на проекти. Со кликање на опцијата Испрати на (анг. Send To) во самиот проект се отвора нов прозорец, избираме Покани луѓе (анг. Invite people.), го копираме (анг. Copy) линкот за споделување и потоа со вметни (анг. Paste) го испраќаме проектот до саканата личност преку е-пошта или некоја социјална мрежа. Всушност со пребарување на интернет може да се најдат многу готови проекти изработени во Tinkercad и со нивно отворање и избор на опцијата Размисли (анг. Tinker this) веб-пребарувачот директно го отвора избраниот проект во Tinkercad платформата. Ова може многу да помогне во подобрување на нашите проекти.

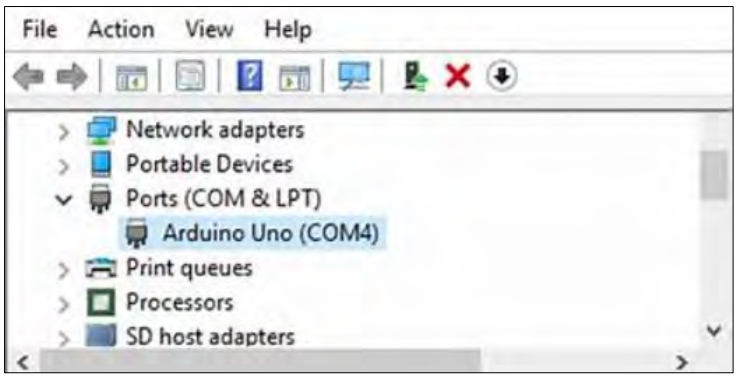
4.4. Практична вежба: Инсталација на развојна средина за Arduino микрокомпјутер и ставање во употреба

4.4.1. Упатство за инсталација на развојна средина за Arduino микрокомпјутери на плочка

Развојната програма се презема од **официјалната веб-страница на Arduino,** поточно од следниов линк https://www.arduino.cc/en/software . Може да одбереме една од неколкуте различни верзии на развојни програми Arduino, но доколку користиме оперативен систем Windows 10, најдобро е да се преземе најновата верзија. Да нагласиме дека оваа верзија нема да функционира на постар оперативен систем, како што се Windows XP или Windows 7. По преземањето на документот за инсталација, ја инсталираме развојната програма. Притискаме (click) на **„Run“,** ги прифаќаме условите за лиценцирање, избираме папка за чување на програмата и на крајот притискаме „Install“. [5]

По инсталацијата, ја поврзуваме Arduino Uno R3 со персоналниот компјутер преку USB-кабел со два различни приклучоци, од A и B-тип. Arduino Uno R3 добива напојување од персоналниот компјутер и светнува зелената вградена диода обележана со буквата L. Кога уредот Arduino Uno R3 за првпат го поврзуваме со компјутер**, автоматски се инсталира драјвер**. Драјверот е софтвер што му овозможува на компјутерот да комуницира со новиот уред.

Ако се работи за постара верзија на оперативен систем Windows, може да се појави прозорец со барање да се посочи локацијата од каде што треба да се инсталира драјверот. Ако не започне инсталацијата, потребно е да се отвори програмата **Control Panel**, да се избере категоријата **Device Manager**, потоа поткатегоријата **„Other Devices“ или „Unknown Devices“,** и да се притисне (анг. click) **„Update Drivers“ или „Update Driver Software“.** Потоа го посочуваме драјверот за поврзување на Arduino микрокомпјутерот на плочка, така што се избира папката „Drivers“ во состав на самата Arduino папка.



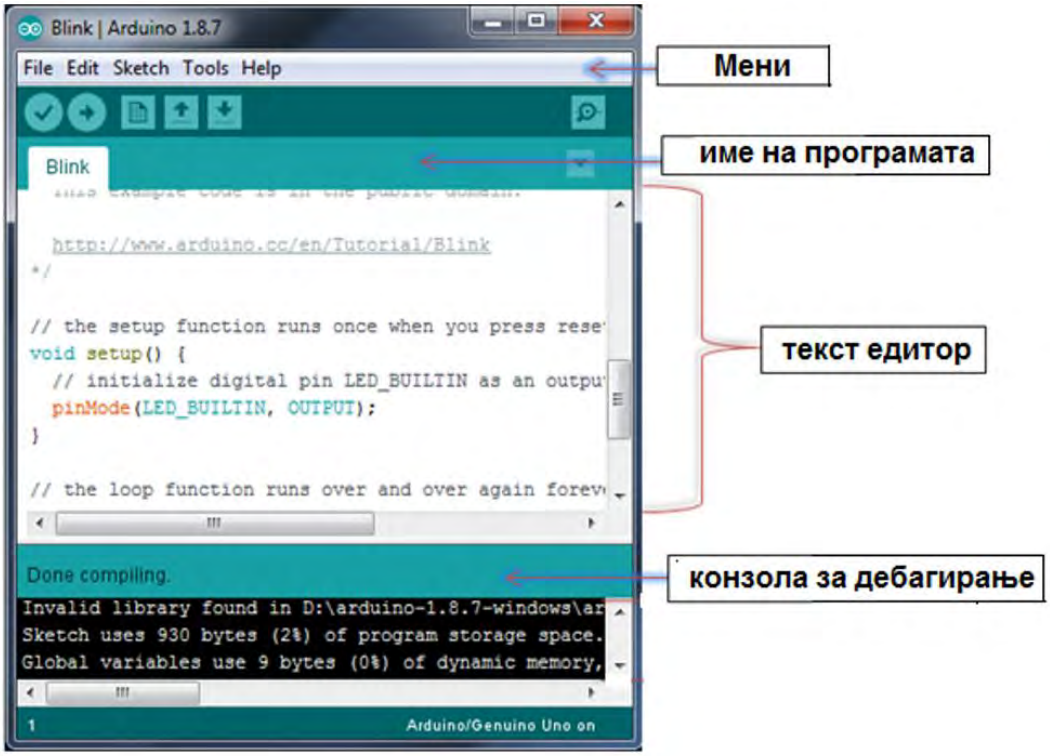
Слика 4.6. Наоѓање на драјвер за Arduino Uno

Како потврда за успешната инсталација, во категоријата Device Manager, со притискање на опцијата „PORT (COM&LPT)“ треба да се појави Arduino Uno. Ако не знаеме на која порта е поврзан Arduino микрокомпјутерот тогаш можеме да го исклучиме Arduino-то од компјутер, а потоа повторно да го вклучиме и да видиме која од сериските порти ќе се појави како нова.

Arduino користи програмски кодови од отворен тип, што значи програмите се бесплатни. Секој програмер може да ги објави своите програмски кодови и на таков начин да придонесе за развој на оваа платформа. Самата развојна средина за Arduino содржи многу готови програми и листата на овие програми можеме да ја видиме со притискање File→**Exampes.** Готова програма можеме да преземеме и од некоја Arduino веб-страница и да ја отвориме со притискање File→Open→Look in и на крај фолдерот во кој сме ја сочувале програмата.

4.4.2. Мени и алатки на развојна средина и впишување на програма во Arduino Uno R3

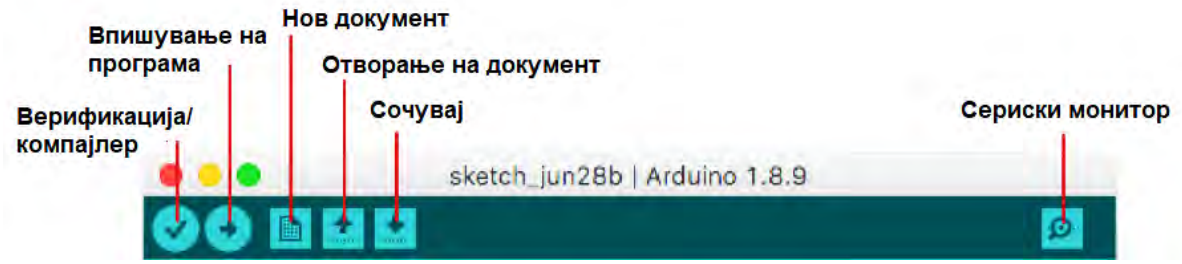
На слика 4.7. е прикажана Arduino развојната програма**. Поважни составни делови се: менито, лентата со алатки, уредувачот на текст, серискиот монитор, библиотеките и конзолата за дебагирање.**



Слика 4.7. Изглед на развојна средина за Arduino микрокомпјутери на плочка

Мени ⟶ Во менито се наоѓаат следниве категории: File, Edit, Sketch, Tools, Help.

Лента со алатки → Во лентата со алатки се наоѓаат копчињата: верификација/компајлер (анг. Verify/Compile), прикачување т.е. впишување (анг.Upload), отворање на нов документ (анг. New), отворање на стар документ (анг. Open) и сочувај (анг. Save). Со алатката Verify се проверува кодот, ред по ред, со цел да се откријат евентуалните грешки. Со алатката Upload, верификуваниот код се впишува во програмската меморија на микроконтролерот во составот на ATmega328 .



Слика 4.8. Лента со алатки за Arduino развојна средина

Уредувач на текст → Ова е најважниот дел од развојната програма бидејќи во него ги пишуваме кодовите, програмите во програмскиот јазик C/C++.

Сериски монитор → Со негово притискање се овозможува преглед на податоците што се испраќаат или се добиваат од Arduino. Има голема примена кај апликациите во реално време.

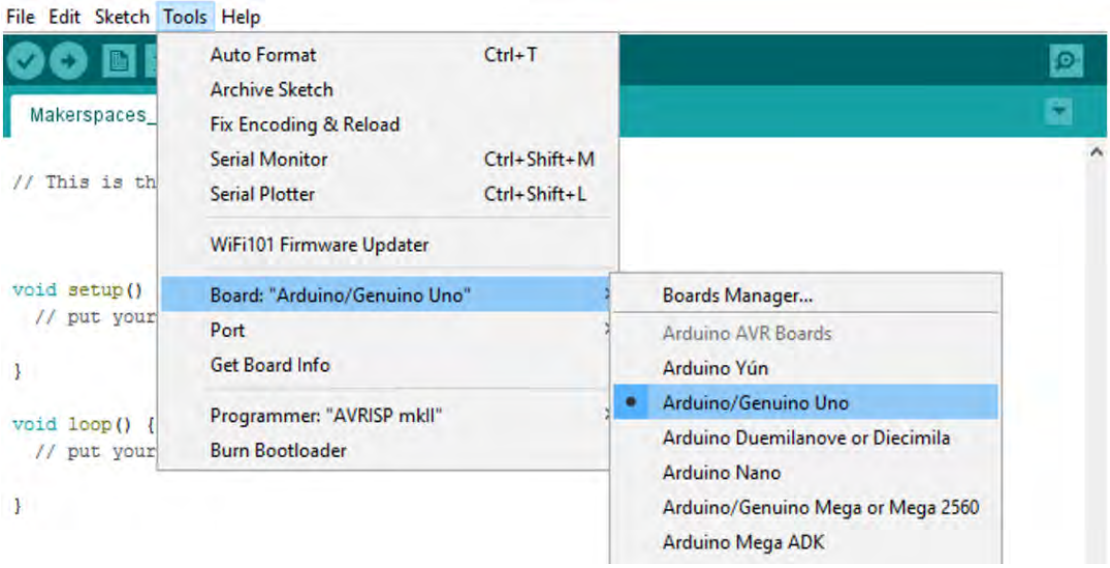
Конзола за дебагирање → Овој прозорец ни дава информација за успешното компајлирање или укажува на редниот број на редицата што предизвикува грешка во компајлирањето. Компајлирањето е постапка на преведување на програмата од програмски јазик од повисок ред во машински јазик.

Библиотеки (анг. libraries) → Со притискање (анг. click) на оваа категорија, се отвора листа на потпрограми за работа со најразлични влезно-излезни уреди.

После инсталирањето на развојната средина и запознавањето со менито и алатките ќе ја објасниме постапката за впишување на програма во меморијата на Arduino Uno R3. За таа цел ќе ја искористиме вградената лед диода која е поврзана со 13 пин на Arduino Uno R3. Исто така ќе употребиме готова програма која не бара употреба на други електронски компоненти. Оваа вежба е една од наједноставните вежби и претставува еден вид тест за работата на Arduino Uno R3.

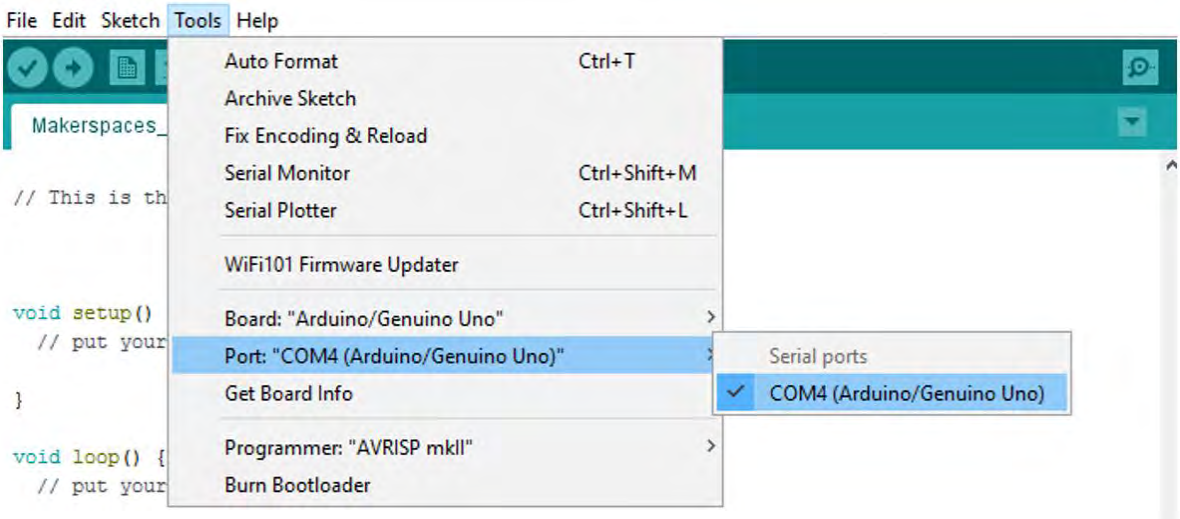
1. По инсталацијата, го поврзуваме Arduino Uno R3 со персоналниот компјутер преку USB-кабел со два различни приклучоци, од A и B-тип.

2. Прв чекор е избор на вид на Arduino микрокомпјутер и избор на сериска порта. Да се потсетиме, фамилијата Arduino беше составена од повеќе различни платформи: Uno, Mega, Nano, Leonardo и др. За избор на Arduino Uno R3 треба да се притисне на опциите Tools → Board →Arduino Uno. Ова е прикажано на слика 4.9.



Слика 4.9. Постапка за избор на Arduino микрокомпјутер

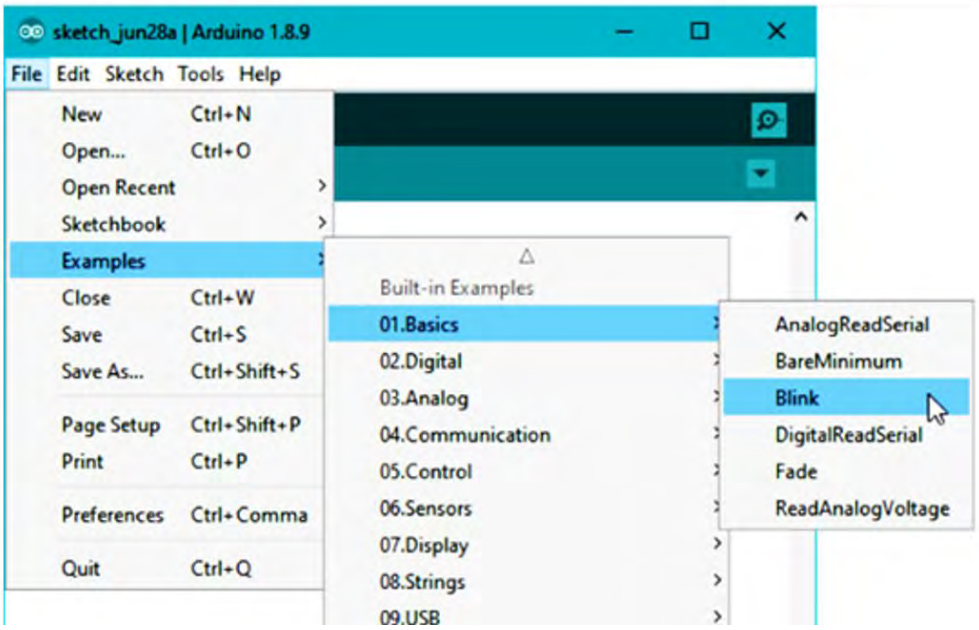
3. За избор на порта, потребно е да се притисне опцијата Tool→Serial Port. Ова е прикажано на слика 4.10. При тоа се појавува паѓачко мени на достапни сериски порти и тоа се разликува во зависност од уредите кои се поврзани со компјутерот.



Слика 4.10. Постапка за избор на порта за Arduino Uno

4. Програмскиот код го наоѓаме со притискање на File→Example, а потоа избираме 01. Basic→Blink. (слика 4.11.)

5. Пред да се впише програмата во меморијата на Arduino Uno R3 истата треба да се преведе на машински јазик односно компајлира. За таа цел во листата на алатки го притискаме копчето Verify/Compile или во менито избираме Sketch→ Verify/Compile



Слика 4.11. Отворање на програмата Blink за Arduino Uno R3

6. Во конзолата за дебагирање добиваме известување “Compiling Sketch….” и се појавува лентата за напредок на постапката и на крај треба да се добие известување “Done Compiling” (слика 4.12.).

Слика 4.12. Известување за успешно компајлирање

7. Ако нема грешки во програмскиот код, во конзолата за дебагирање добиваме информација за големината на програмата што сме ја отвориле изразена во бајти и информација за големината на останатиот слободен мемориски простор .

8. Со притискање на копчето за прикачување (анг. Upload) започнува впишувањето на програмата во меморијата. Вградената лед-диода со ознака L ќе престане да свети, а ќе започнат да светат двете диоди за сериски пренос, со ознаки RX и TX. Впишувањето трае неколку секунди после што почнува да трепка (анг. blink) лед-диодата со ознака L.

4.5. Практични вежби за програмирање на Arduino Uno R3 во програмскиот јазик C/C++

Сите вежби се едноставни и наменети за почетници, со цел практична примена на теоретските знаења од програмирање на Arduino Uno R3. За успешно програмирање на Arduino Uno R3 потребно е познавање на нејзиниот хардвер и софтвер. Пред почетокот на практичните вежби треба да се повтори следниот наставен материјал:

- 4.3. Периферни елементи и електронски компоненти за поврзување со Aduino микрокомпјутер на плочка, од теоретскиот дел на учебникот
- 4.1. Мерки за заштита и безбедност при работа со Arduino микрокомпјутер на плочка, од практичниот дел на учебникот
- 4.2. Упатство за работа со протоплочка, од практичниот дел
- 4.3. Компјутерска симулација за Arduino Uno R3, од практичниот дел
- 4.4.2. Мени и алатки на развојна средина и впишување на програма во Arduino Uno R3, од практичниот дел

4.5.1. Практична вежба: Вклучување на лед-диода преку тастер

1. **Цел на вежбата**

Цел на вежбата е поврзување на лед-диода и тастер, како влезна и излезна компонента, со Arduino Uno R3, според дадена функционалната и монтажна шема и употреба на протоплочка. Учениците треба да напишат програма во развојната средина, да ја внесат програмата во меморијата на Arduino Uno R3 и да ја испитаат нејзината работа.

2. **Време на реализација:** 2 наставни часа

3. **Потребни компоненти** за реализација на оваа вежба се:
    - Arduino Uno R3 микрокомпјутер на плоча
    - протоплочка
    - тастер за печатена плочка
    - лед диода
    - отпорници со отпорности R1=220Ω и R2=1KΩ
    - жици за поврзување (краткоспојници)
    - компјутер со инсталирана Arduino развојна средина

Тастерот испраќа вредности (HIGH или LOW) до Arduino Uno R3, а потоа Arduino управува со лед-диодата. Отпорникот R1 е потребен за да се нагоди работната струја на диодата, со цел јачината на светлината да биде оптимална, без штетно

загревање. Отпорникот R2 врши заштита на пинот на кој е поврзан тастерот од брзи промени на неговиот напон.

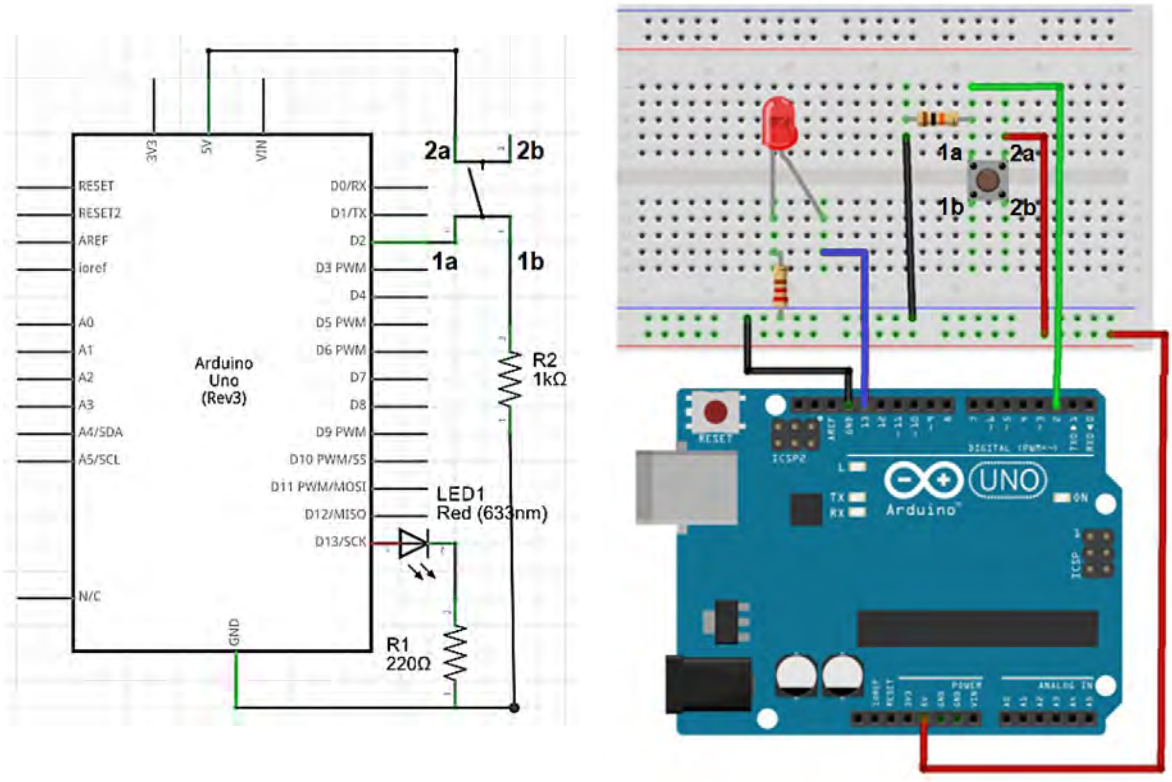
4. Подготовка за вежбата

Учениците треба да се потсетат на:

- пин-дијаграмот на Arduino Uno R3
- начинот на поврзување на лед-диода и тастер на протоплочка. Истиот е објаснет во упатството за работа со протоплочка.
- влезно-излезните инструкции за Arduino Uno R3, задолжителните структури (setup и loop) и if…else структурата за избор на можности.

5. Опис на електричната шема и начин на поврзување

На слика 4.13. се прикажани функционалната и монтажната шема за реализација на оваа вежба.



Слика 4.13. Поврзување на лед-диода и тастер со Arduino Uno R3

Кога тастерот не е притиснат вториот пин на Arduino Uno R3 е поврзан со заземјувањето што одговара на ниско логичко ниво (анг. LOW) и напонот на вториот пин изнесува 0V. Кога тастерот е притиснат вториот пин на Arduino Uno R3 е поврзан со напојувањето што одговара на високо логичко ниво (анг. HIGH) и напонот на вториот пин е различен од нула. Лед-диодата свети кога 13-от пин е на високо логичко ниво односно кога излезниот напон на пинот изнесува 5V.

Поврзувањето на компонентите се реализира на следниот начин:

(1) Го поставуваме тастерот на средината од протоплочката. За поврзување на тастерот се користат три краткоспојници. Со црвена

краткоспојница ги поврзуваме приклучоците 2a-2b со лентата за напојување (обележана со + или со црвена боја). Со црната краткоспојница и преку отпорникот R2 ги поврзуваме приклучоците 1a-1b на тастерот со лента за заземјување (обележана со минус или сина боја). Истовремено, со зелената краткоспојница приклучоците 1a-1b ги поврзуваме со вториот дигитален пин на Arduino Uno R3. Зелената краткоспојница е еден вид сигнална жица и преку неа тастерот ги испраќа вредностите HIGH или LOW до вториот пин.

(2) Ја поставуваме лед-диодата на протоплочката при што треба да се внимаваме анодата да биде поврзана на контактна точка со повисок потенцијал од онаа на катодата. Со црна краткоспојница и преку отпорникот R1 ја поврзуваме катодата на лед-диодата со лентата за заземјување, а со сина краткоспојница ја поврзуваме анодата со 13-от пин на Arduino Uno R3.

(3) Со црвена краткоспојница го поврзуваме 5V пинот на Arduino Uno R3 со лента за напојување на протоплочката, а со црната жица го поврзуваме GND пинот со лентата за заземјување.

6. Пишување и внесување на програма за Arduino Uno R3

Arduino Uno R3 ја проверува состојбата на тастерот со употреба на инструкцијата digitalRead(). Лед-диодата се вклучува и исклучува софтверски со употреба на инструкцијата digitalWrite(). Програмата содржи if...else структура за избор на една од две можности (лед-диодата свети или лед-диодата не свети). Состојбата на тастерот е услов за извршување на if...else структура

Подолу е дадена програмата со којашто лед-диодата се вклучува и исклучува софтверски, во зависност од состојбата на тастерот (притиснат или отпуштен).

```
// Се врши декларирање на константите и променливите пред да се напишат
//структурите setup и loop
// Константите не се менуваат и се користат за избор на пинови за поврзување
1  const int buttonPin = 2;        // Тастерот е поврзан на дигиталниот
                                   // пин број 2.
2  const int ledPin =  13;         // Лед-диодата е поврзана со
                                   // дигиталниот пин број 13.
// Променливите чуваат податоци што постојано се менуваат.
3  int buttonState = 0;            // buttonState е име за вредноста што се
                                   // добива со читање на пинот број 2,
                                   // каде е приклучен тастерот.
4  void setup() {                  // Почеток на структурата setup.
5    pinMode(ledPin, OUTPUT);      // Пинот на кој е поврзана лед-диодата
                                   // се конфигурира како излезен пин, што
                                   // значи Arduino ќе испраќа податоци.
6    pinMode(buttonPin, INPUT);    // Пинот на кој е поврзан тастерот се
                                   // конфигурира како влезен пин, што
                                   // значи Arduino ќе прима податоци.
```

```
7   }                                  //  Крај на структурата setup.
8   void loop() {                      //   Почеток на структурата loop.
9     buttonState=digitalRead(buttonPin);  //  Се чита вредноста на пинот број 2 и
                                       //  таа се запишува во променливата
                                       //  buttonState.
10    if (buttonState == HIGH) {       //  Се проверува дали тастерот е
                                       //  притиснат
11      digitalWrite(ledPin, HIGH);    //  Ако е притиснат, тогаш лед-диодата
                                       //  свети.
12    }                                //  Крај на исказот if.
13      else {                         //  Почеток на исказот else.
14      digitalWrite(ledPin, LOW);     //  Ако не е притиснат тастерот, диодата
                                       //  не свети.
15    }                                //  Крај на исказот else.
16  }                                  //  Крај на структурата loop.
```

Постапката за внесување на програма во меморијата на Arduino Uno R3 е иста за сите програми и истата е објаснета, чекор по чекор, во практичната вежба 3.2.6.2. Мени и алатки на развојна средина и впишување на програма во Arduino Uno R3.

7. Резултати од реализацијата на вежбата

Веднаш после внесувањето на програмата треба да се провери дали Arduino Uno R3 правилно работи. Кога ќе го притиснеме тастерот лед-диодата треба да свети. Ако го отпуштиме тастерот тогаш лед-диодата престанува да свети.

Коментар:_____

8. Направи промена!

- Што ќе се случи ако двете digitalWrite инструкции во програмата си ги заменат местата?

  Коментар:_____
- Што ќе се случи ако внесеме време на доцнење од 5s после секоја од двете digitalWrite инструкции? Да се употреби инструкцијата delay().

Коментар:_____

4.5.2. Практична вежба: Контрола на лед-диода со потенциометар

1. Цел на вежбата

Оваа вежба претставува практична примена на аналогните влезови на Arduino Uno R3. Учениците ќе научат како големината на аналогните влезни сигнали влијаат на времетраењето на сигналите на излез од дигиталните пинови

на Arduino Uno R3. Со помош на потенциометар ќе ја менуваме брзината на трепкање на лед-диода.

2. **Време на реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Arduino Uno R3 микрокомпјутер
- протоплочка
- тример-потенциометар со отпорност од 10 КΩ
- лед диода
- отпорник со отпорност R1=220Ω
- жици за поврзување (краткоспојници)
- компјутер со инсталирана Arduino развојна средина

Во оваа практична вежба потенциометарот е главна компонента која е поврзана со еден од аналогните влезови на Arduino Uno R3. Потенциометарот е со променлива отпорност и со него се генерира променлив напон со максимална вредност од 5 V и минимална вредност од 0 V. Аналогно-дигиталниот претворувач вграден во самото Arduino ги претвора аналогните вредности во цели броеви од 0 до 1023. Колку е поголема вредноста на излез од аналогно-дигиталниот претворувач, толку поретко диодата ќе се вклучува и исклучува, односно лед-диодата ќе трепка со помала брзина.

4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

- импулсно ширинската модулација и нејзиното значење за аналогните влезови на Arduino Uno R3
- инструкциите за работа со аналогните влезни пинови на Arduino Uno R3
- значењето на инструкцијата delay() и изборот на време на доцнење

5. **Опис на електричната шема и начин на поврзување**

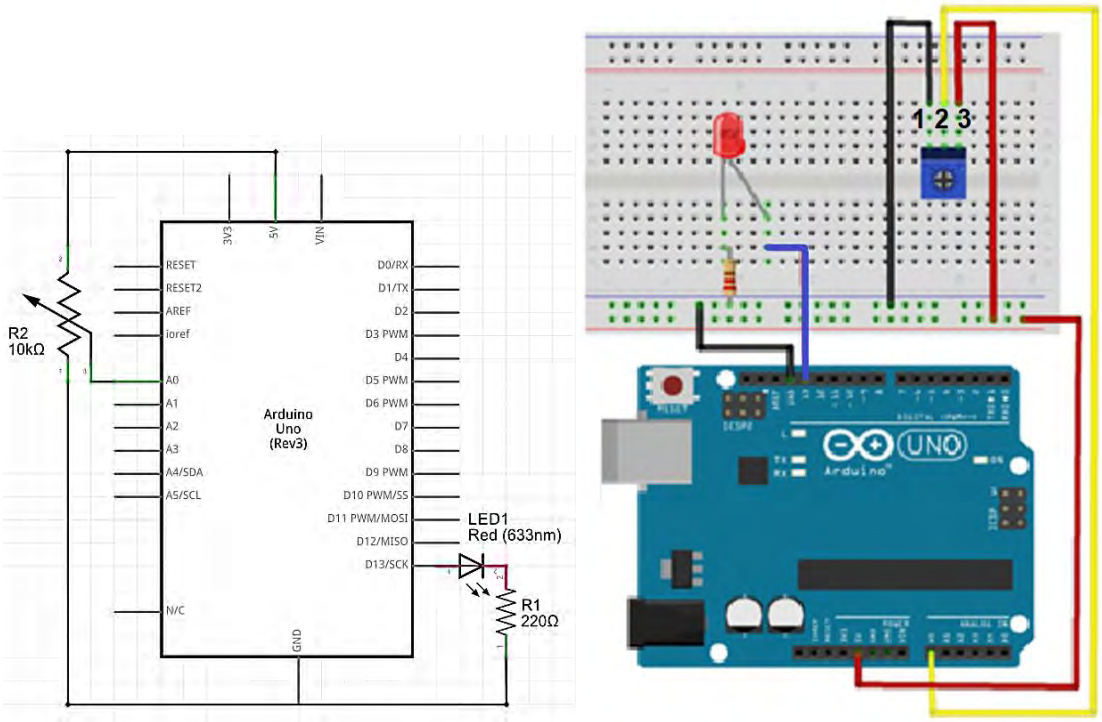
На слика 4.14. се прикажани функционалната и монтажната шема за реализација на оваа вежба. Потенциометарот има три приклучоци од кои два приклучоци се напојување и заземјување. **Средниот пин на потенциометарот** е контролен или податочен пин и него **го поврзуваме со аналогниот пин A0** на Arduino Uno R3.

Начинот на поврзување е следен:

(1) Го поставуваме потенциометарот на протоплочката при што треба да внимаваме неговите приклучоци да ги поставиме во контактни точки кои припаѓаат на различни проводници. Приклучоците на потенциометарот се нумерирани од 1 до 3. Приклучоците со број 1 и 3 се поврзуваат со лентите за напојување и заземјување. Пинот со број 2 , преку жолтата краткоспојница, директно се поврзува со аналогниот пин A0.

(2) Ја поставуваме лед-диодата на протоплочката исто како што беше објаснето во претходната вежба, катодата ја поврзуваме со лентата за заземјување, а анодата со 13-от пин.

(3) Пиновите означени со GND и 5V на Arduino Uno R3 ги поврзуваме со лентите за заземјување и напојување на протоплочката.



Слика 4.14. Поврзување на лед-диода и тример-потенциометар со Arduino Uno

6. Пишување и внесување на програма за Arduino Uno R3

Arduino Uno R3 ја проверува вредноста на сигналот на аналогниот влезен пин A0 со употреба на инструкцијата analogRead(). Прочитаната вредност претставува променлива со симболично име sensorValue. Оваа променлива се запишува како време на доцнење во двете инструкции delay() кои следат после двете инструкции digitalWrite(), со кои се менува состојбата на лед-диодите.

Подолу е дадена програмата за контрола на фреквенцијата на трепкање на лед-диодата со потенциометар.

```
1  int sensorPin = A0;      // Избор на влезен пин за
                            // потенциометарот.
2  int ledPin = 13;         // Избор на пин за поврзување на лед-
                            // диода
3  int sensorValue = 0;     // Декларираме променлива за
                            // складирање на вредноста добиена од
                            // потенциометарот.
4  void setup() {           //
5    pinMode(ledPin, OUTPUT); // Пинот ledPin се конфигурира како
                            // излезен
6  }                        // Крај на структурата setup
```

```
7   void loop() {                    //  Почеток на циклусот.
8     sensorValue =analogRead(sensorPin);
 //Прочитаната вредност од потенциометарот се сместува во променливата со
 //симболично име „sensorValue“
9     digitalWrite(ledPin, HIGH);    //  Лед-диодата се вклучува
10    delay(sensorValue);            //  Диодата свети онолку милисекунди
                                     //  колку што изнесува вредноста добиена
                                     //  од потенциометарот
11    digitalWrite(ledPin, LOW);     //  Лед-диодата се исклучува
12    delay(sensorValue);            //  Диодата нема да свети онолку
                                     //  милисекунди колку што изнесува
                                     //  вредноста добиена од потенциометарот.
                                     //  Потоа повторно се враќаме на почетокот
                                     //  од циклусот.
                                     //
13  }                                //  Крај на циклусот.
```

После пишувањето на програмата истата се внесува во меморијата на Arduino Uno R3 како што беше претходно објаснето.

7. Резултати од реализација на вежбата

Кога ќе се вклучи напојувањето за Arduino Uno R3, со помош на шрафцигер се нагодува отпорноста на потенциометарот од минимум до максимум. Кога отпорноста е најголема тогаш вредноста на влезниот сигнал и времето на доцнење се најмали и лед-диодата ќе трепка со најголема брзина . Доколку времето на доцнење е премногу мало, можно е да не се забележи трепкањето на диодата. Да нагласиме дека максималното време на доцнење изнесува 1023.

Коментар:_____

8. Направи промена!

- Што ќе се случи доколку во инструкциите delay() променливата sensorValue се помножи со пет?

Коментар:_____

- Што ќе се случи ако после читањето на вредноста на аналогниот влезен сигнал во програмскиот код се вметне нова инструкција procent=map(sensorValue ,0,1023,0,100) како што е прикажано во пример 4.53?

Пример 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    }
```

Коментар:_____

4.5.3. Практична вежба: Регулација на интензитетот на светлина на лед-диода

1. **Цел на вежбата**

Учениците ќе користат еден од аналогните излезни пинови на Arduino Uno R3 и софтверски ќе ја менуваат големината на излезниот напон. Промените на напонот ќе се следат преку употреба на лед-диода при што интензитетот на светлината постепено ќе се менува од минимум до максимум и обратно.

2. **Време за реализација:** 1 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Arduino Uno R3 микрокомпјутер
- протоплочка
- лед диода
- отпорник со отпорност R1=220Ω
- две жици за поврзување (краткоспојници)
- компјутер со инсталирана Arduino развојна средина

4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

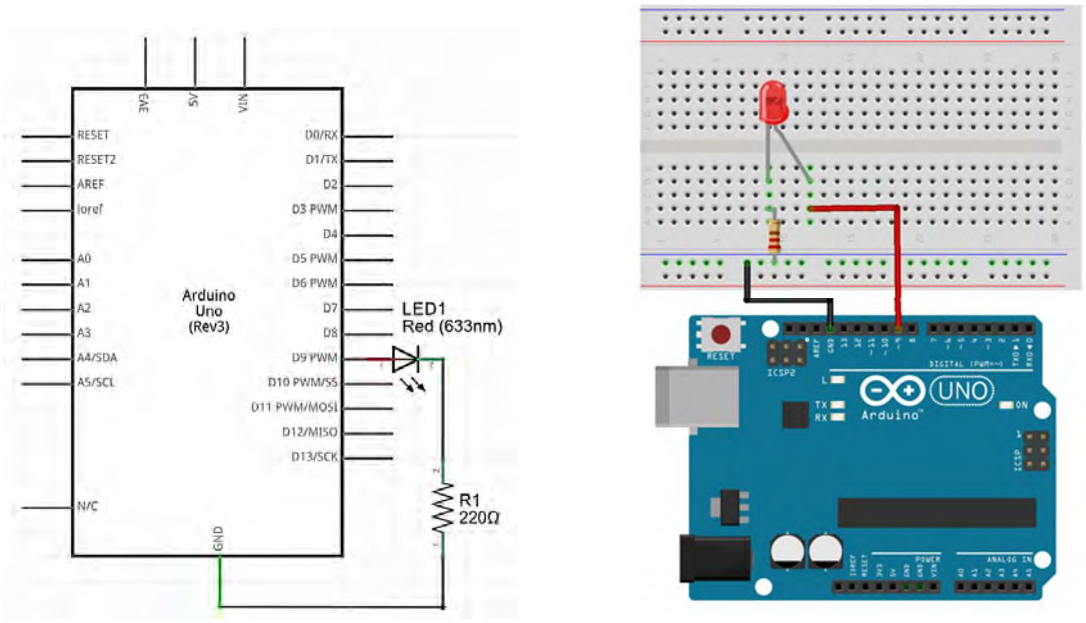
- импулсно ширинската модулација, нејзиното значење за аналогните излези на Arduino и бројот на можни вредности
- инструкциите за работа со аналогните излезни пинови на Arduino микрокомпјутер на плочка
- for структурата за повторување на циклуси

5. **Опис на електричната шема и начин на поврзување**

На слика 4.15. се прикажани функционалната и монтажната шема за реализација на оваа вежба.

Колото е многу едноставно. Катодата на лед диодата, преку отпорникот R1, треба да се поврзе со лентата за заземјување, а анодата со деветтиот пин Arduino Uno R3. Во оваа вежба не можеме да го употребиме тринаесеттиот пин како во претходните вежби, бидејќи на него не можат да се запишуваат аналогни вредности.

Излезниот напон, со кој се контролира интензитетот на светлината на лед-диодата, го задаваме преку употреба на инструкцијата analogWrite() при што во средната заграда се внесува аналогната вредност. Минималната аналогна вредност изнесува 0, а максималната 255.



Слика 4.15. Поврзување на лед-диода на PWM-пин од Arduino Uno R3

Програмата за регулација на интензитетот на светлината користи две „for“ структури за повторување на циклус. Во првата for структура, променливата ја зголемуваме за пет и циклусот се повторува сè додека не добиеме вредност 255. Во втората for структура, променливата се намалува за пет сè додека не се постигне вредност нула.

```
1   int ledPin = 9;                          //  Лед-диодата е поврзана на пин
                                             //  број 9.
2   void setup()  {                              Нема конфигурација на
3   }                                        //  пиновите
4   void loop()  {                           //  Почеток на циклусот
5     for(int fadeValue = 0; fadeValue <= 255; fadeValue +=5)
    {
// За for структурата декларираме променллива со симболично име  „fadeValue“ и
// почетна вредност нула. Неа ја зголемуваме за 5, сè додека не добиеме број
// еднаков на 255. По секое зголемување или намалување се чека 30 милисекунди.

6       analogWrite(ledPin, fadeValue);      //  Го зголемуваме интензитетот
7       delay(30);                           //  на светлината
8     }                                      //
9     for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5)
    {
//Го намалуваме интензитетот на светлината. Променливата ја намалуваме за пет
//додека не добиеме вредност нула. Потоа се враќаме на почетокот од циклусот

10      analogWrite(ledPin, fadeValue);

11      delay(30);
12    }
13  }
```

Како и претходните програми така и оваа програма ја внесуваме во Arduino Uno R3 со следнава постапка:

(1) Ја поврзуваме Arduino Uno R3 со компјутер преку USB кабел
(2) Во главното мени ја избираме опцијата Tools и избираме вид на Arduino микрокомпјутер и сериска порта
(3) Ја преведуваме програмата на машински јазик преку употреба на алатката Verify/Compile
(4) И на крај ја внесуваме програмата со притискање на алатката Upload

Веднаш после внесувањето на програмата електричното коло ќе почне да работи односно интензитетот на светлината на лед-диодата ќе почне наизменично да се намалува и зголемува.
Коментар:_____

- Што ќе се случи ако го промениме чекорот на зголемување или намалување на аналогната вредност од пет на некоја друга вредност? Коментар:_____
- Што ќе се случи ако го зголемиме времето на доцнење во инструкцијата delay() ?

Коментар:_____

4.5.4. Практична вежба:Фотоотпорник за контрола на интензитет на светлина

1. Цел на вежбата

Во оваа вежба ќе го менуваме интензитетот на светлината на лед-диода со употреба на фотоотпорник. Имено лед-диодата ќе послужи како покажувач за осветленоста на околината во која е поставена Arduino Uno R3 заедно со фотоотпорникот. Исто така во оваа вежба учениците ќе научат што значи и како се врши калибрација на сензор.

2. Време за реализација: 2 наставни часа

3. Потребни компоненти за реализација на оваа вежба се:

- Arduino Uno R3 микрокомпјутер
- протоплочка
- лед-диода
- фотоотпорник
- отпорници со отпорност R1=220Ω и R2=10KΩ
- жици за поврзување (краткоспојници)
- компјутер со инсталирана Arduino развојна средина

Фотоотпорникот е полупроводнички елемент чиј електричен отпор се менува под влијание на светлината која паѓа врз неговата површина. Заради

оваа особина наоѓа примена како електронски сензор. Пред да го вградиме фотоотпорникот во колото, со употреба на мултиметар, можеме да ја измериме неговата отпорност со тоа што ќе го менуваме интензитетот на светлината. Кога нема светлина неговата отпорност е најголема. Колку повеќе го доближуваме фотоотпорникот до изворот на светлина толку повеќе се намалува неговата отпорност.

4. Подготовка за вежбата

Учениците треба да се потсетат на:

- основните карактеристики на фотоотпорниците
- пин-дијаграмот на Arduino Uno R3
- инструкциите за работа со аналогни пинови и опсегот на можни вредности
- значењето на условната if структура
- примената на инструкцијата map() за промена на опсегот на вредности.

5. Опис на електричната шема и начин на поврзување

На слика 4.16. се прикажани функционалната и монтажната шема за реализација на оваа вежба.

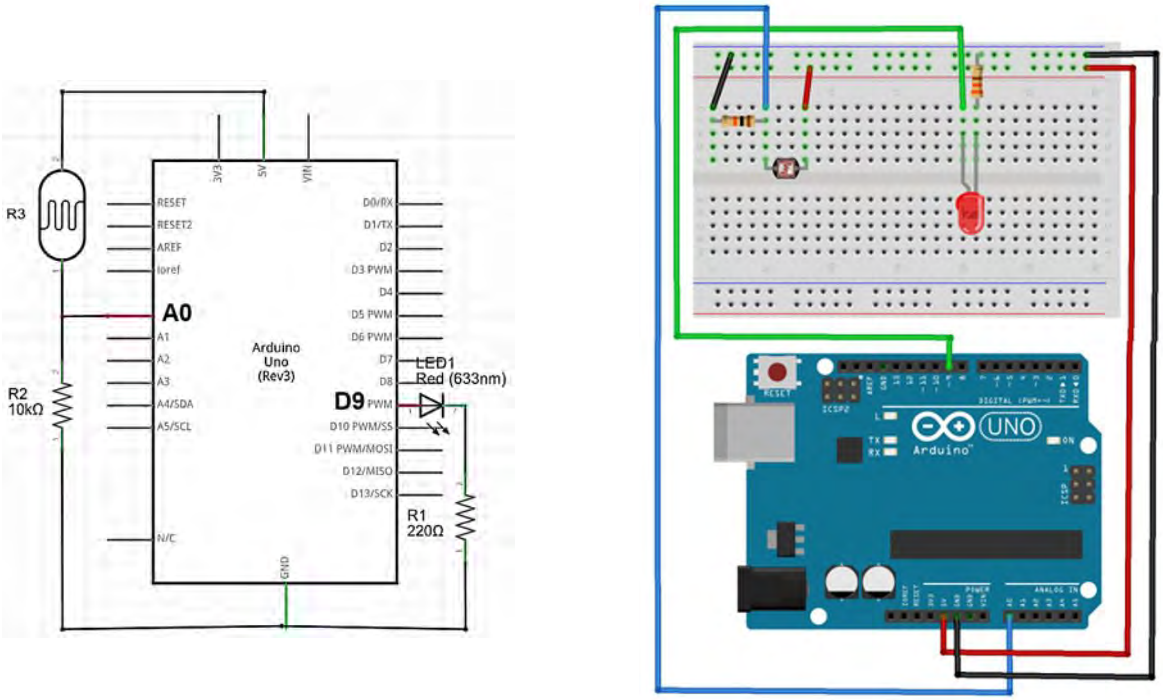
(1) Го поставуваме фотоотпорникот на протоплочката при што треба да внимаваме неговите приклучоци да ги поставиме во контактни точки кои припаѓаат на различни проводници. Со црвена краткоспојница едниот крај на фотоотпорникот го поврзуваме со лентата за напојување. Другиот крај на фотоотпорникот, преку отпорникот R2 и црна краткоспојница го поврзуваме со лентата за заземјување. Всушност фотоотпорникот и отпорникот R2 формираат напонски делител. Со сината краткоспојница контактната точка меѓу фотоотпорникот и отпорникот R2 се поврзува со аналогниот влез A0.

(2) Како во претходната вежба, катодата на лед-диодата, преку отпорникот R1 и црната краткоспојница, ја поврзуваме со лентата за заземјување. Со зелената краткоспојница ја поврзуваме анодата со деветтиот пин со ознака PWM, што значи дека е аналоген излез.

(3) Пиновите GND и 5V на Arduino Uno R3 ги поврзуваме со лентите за напојување и заземјување.

Arduino Uno R3 го мери напонот добиен од напонскиот делител и измерените вредности ги претвора правопропорционално во целобројни вредности во опсег од 0 до 1023. Целобројната вредност 1023 одговара на напон од 5V, а нулата на напон од 0V. Но, поради падот на напонот на отпорникот R2 влезниот напон на пинот A0 никогаш нема да достигне вредност 5V, ниту пак влезниот сигнал ќе добие максимална вредност. Исто така, фотоотпорникот е дизајниран да детектира промени во интензитетот на светлината во многу поширок опсег него што се промените на светлината во услови на затворен простор. Од овие причини потребно е да се изврши калибрација на фотоотпорникот. Калибрацијата е неопходна за сензорот да работи прецизно. Во мерната техника под поимот калибрација на мерен инструмент подразбираме

утврдување на отстапувањето на вредноста која се отчитува од мерниот инструмент во однос на вистинската вредност (вредноста на еталонот). Во однос на Arduino Uno R3 и фотоотпорникот отстапувањето претставува разлика помеѓу максималната или минималната вредност на влезниот сигнал што ја дава самото Arduino и максималната или минималната вредност што може да ја детектира самиот фотоотпорник. Крајните вредности во опсегот од 0 до 1023 никогаш нема да се постигнат што автоматски значи дека лед-диодата секогаш ќе свети, но никогаш нема да го достигне својот максимум или минимум. За да се елиминира ваквото отстапување треба да се нагодат максималната и минималната вредност на сензорот уште во првите неколку секунди по пуштањето во работа на електричното коло. Тоа се постигнува со движење на раката кон и од сензорот, при што Arduino софтверски ги одбира опсегот на максимална и минимална вредност.



Слика 4.16. Поврзување на лед-диода и фотоотпорник со Arduino Uno R3

6. Пишување и внесување на програма за Arduino Uno R3

Калибрацијата се извршува само еднаш, таа не се повторува и затоа истата се пишува како програмски код во составот на setup задолжителната структура. Поради потребата од калибрација се воведени две нови променливи со симболични имиња lightLow и lightHigh. Променливата lightHigh постојано се зголемува од нула до максималната вредност што може да ја детектира фотоотпорникот во дадена физичка средина. Променливата lightLow постојано се намалува од 1023 до минималната вредност на фотоотпорникот. Инструкцијата prilagodilightLevel = map(lightLevel, lightHigh, lightLow, 0, 255) врши претворање на влезниот опсег со калибрирани вредности на минимумот и максимумот во излезен опсег на вредности, од 0 до 255.

Подолу е дадена програмата со коментари за нејзините инструкции и структури.

```
//Вообичаено се започнува со избор на пинови за поврзување на фотоотпорникот и
//лед-диодата.
1   const  int sensorPin = 0;          //   Сензорот е поврзан на нултиот аналоген пин.
                                       //
2   const  int ledPin = 9;             //   Лед-диодата е поврзана на деветтиот пин на
                                       //   чиј излез има ширински модулирани импулси.
                                       //
3   int lightLevel;                    //   Променливата ги чува вредностите добиени
                                       //   од фотоотпорникот.
4   int prilagodilightLevel;               Променливата со симболично име
                                           prilagodilightLevel се користи при промена на
                                           опсегот од влез за излез
5   int lightHigh = 0;                 //   Променливата lightHigh претставува
                                       //   максимална вредност после извршената
                                       //   калибрација.
6   int lightLow = 1023;               //   Променливата lightHigh претставува
                                       //   минимална вредност после извршената
                                       //   калибрација.
7   void setup()
8   {
9     pinMode(ledPin, OUTPUT);         //   Конфигурација на излезен пин за лед-
                                       //   диодата.
// Калибрацијата се извршува во првите пет секунди од почетокот на извршувањето
// на програмата.
10    while (millis() < 5000) {        //   Со инструкцијата millis() се мери времето
                                       //
// Се чита вредноста на аналогниот пин sensorPin (фотоотпорникот) и започнува
// калибрацијата.
11        lightLevel = analogRead(sensorPin);
// Променливата lightLow добива вредност што одговара на минималниот интензитет
// на светлината во дадената средина.
12        if (lightLevel < lightLow) {
13          lightLow = lightLevel;
14        }
//Променливата lightHigh добива вредност што одговара на максималниот интензитет
// на светлината во дадената средина
15        if (lightLevel > lightHigh) {
16          lightHigh = lightLevel;
17        }                            //   Крај на if структура.
18    }                                //   Крај на while структура.
19  }                                  //   Крај на setup структура.
20  void loop()
21  {                                  //   Почеток на структурата loop.
//  Се прочитува вредноста на напонот на пинот sensorPin (фотоотпорникот).
```

```
22      lightLevel = analogRead(sensorPin);
// Преминот од еден во друг опсег е линеарен.
23      prilagodilightLevel = map(lightLevel, lightHigh, lightLow, 0, 255);
24     analogWrite(ledPin, prilagodilightLevel);
// Сигналот што одговара на осветленоста на фотоотпорникот ја побудува
// лед-диодата.
25  }                                   //     Крај на структурата loop
```

Откако ќе ја напишеме програмата истата се внесува во Arduino Uno R3 според веќе објаснетата постапка во претходните примери.

7. Резултати од реализацијата на вежбата

Кога електричното коло ќе го поврземе со напојувањето, веднаш во првите пет секунди, вршиме калибрација така што ја движиме раката кон и од сензорот. Потоа проверуваме дали лед-диодата посилно свети кога врз фотоотпорникот паѓа повеќе светлина отколку кога фотоотпорникот е покриен со рака или друг предмет.

Коментар:_____

8. Направи промена!

- Како потврда на сè што е кажано за калибрацијата, можеме да ги прикажеме добиените резултати на екранот од серискиот монитор. За таа цел треба да ја дефинираме брзината на пренос во setup структурата со употреба на инструкцијата Serial.begin(9600) и да ги внесеме инструкциите Serial.println(lightLow) и Serial.println(lightHigh) после извршената калибрација и Serial.println(lightLevel) пред мапирањето односно пред промената на влезниот опсег вредности во излезен опсег.

Коментар:_____

- Наместо фотоотпорник можеме да поставиме каков било сензор и неговите вредности ќе бидат пресликани во светлината што ја емитува лед-диодата. Исто така, наместо лед-диода можеме да поставиме пиезо-сензор кој ќе испушта тонови или мотор чија брзина ќе зависи од осветленоста на фотоотпорникот. Ваквите промени во програмскиот код бараат добро познавање на инструкциското множество на Arduino микрокомпјутерот.

  Коментар:_____

4.5.5. Практична вежба:Поврзување на Arduino Uno R3 со серво-мотор

1. Цел на вежбата

Целта на оваа практична вежба е придвижување на серво-мотор за 180 степени. За програмирање на серво-моторот и негова контрола учениците треба да ја повикаат библиотека Servo.h., со која се запознавме во наставната единица 4.9. Инструкции за работа со библиотеки во теоретскиот дел на учебникот. Ова

е прва практична вежба за програмирање на Arduino Uno R3, од досега реализираните, која користи библиотека.

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Arduino Uno R3 микрокомпјутер
- протоплочка
- серво-мотор
- жици за поврзување (краткоспојници)
- компјутер со инсталирана Arduino развојна средина

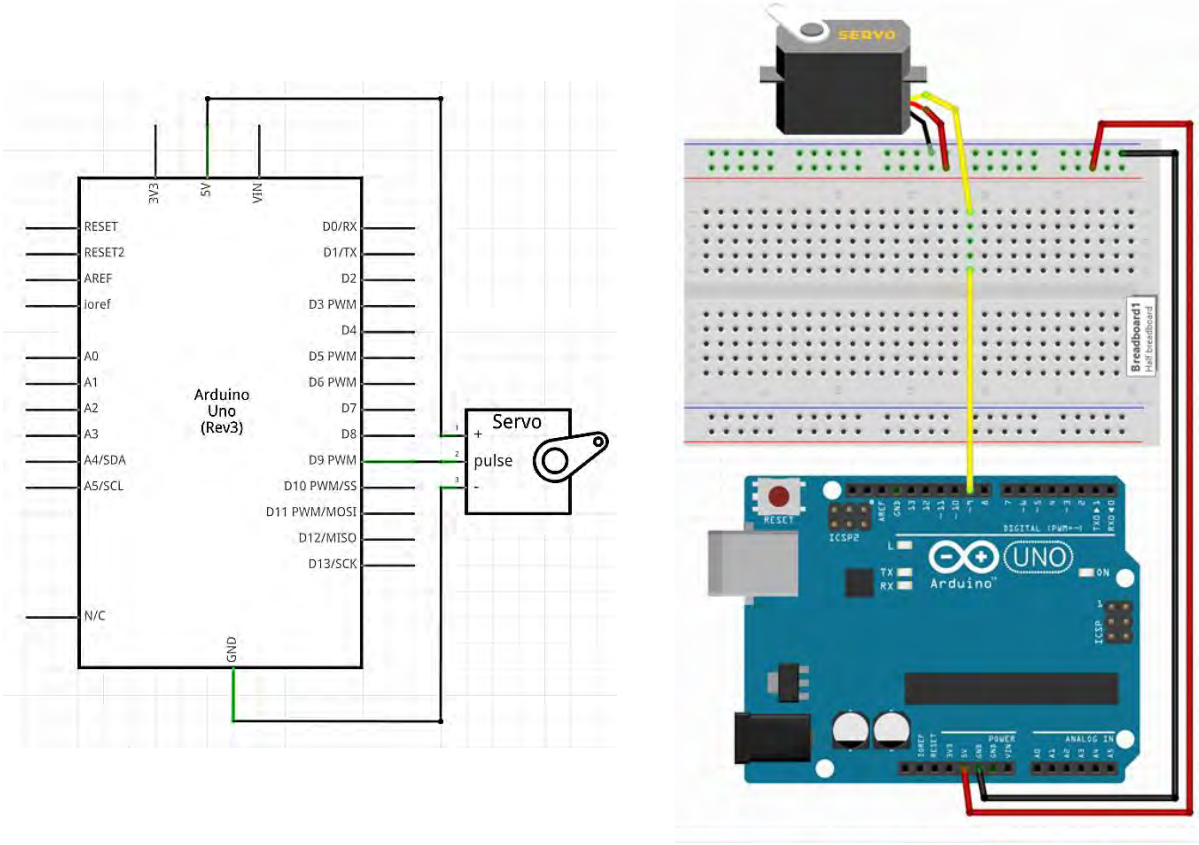
Принципот на работа на серво моторот е прикажана на слика 4.24. во оваа наставна единица 4.9. Инструкции за работа со библиотеки во теоретскиот дел на учебникот.

4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

- пин-дијаграмот на Arduino Uno R3
- принципот на работа на серво-моторот
- инструкциите кои се дел од библиотеката Servo.h
- значењето на for структура за повторување на циклуси

5. **Опис на електричната шема и начин на поврзување**



Слика 4.17. Поврзување на серво-мотор со Arduino Uno R3

На слика 4.17. се прикажани функционалната и монтажната шема за реализација на оваа вежба. Монтажната шема е многу едноставна. Серво-моторот има три жици.

Црната жица е за заземјување, црвената е за напојување и жолтата жица е за контрола на моторот и преку неа серво-моторот е поврзан со 9-тиот пин од Arduino Uno R3. Да се потсетиме, само пиновите 9 и 10 имаат поддршка за поврзување на серво-мотор. Со поставување кондензатор со капацитивност 100 μF на пиновите за заземјување и напојување, можат да се избегнат брзите промени на напонот. Серво-моторот е излезен уред. Во оваа вежба нема влезен уред и серво-моторот ќе го контролираме чисто софтверски, само со инструкции.

6. Пишување и внесување на програма за Arduino Uno R3

Подолу е дадена програма со којашто серво-моторот ќе се заврти за 180 степени, од лево кон десно, со чекор од два степена при секое повторување на структурата for.

```
1  #include <Servo.h>                 //  Вклучуваме библиотека за контрола на серво-
                                       //  мотор.
2  Servo myServo;                     //  Дефинираме серво-објект со симболично
                                       //  име myServo. На овој објект ќе се однесуваат
                                       //  наредбите од библиотеката Servo.
3   int position;                      //  Променливата position ја содржи вредноста
                                       //  на аголот за кој треба да се заврти серво-
                                       //  моторот.
4  void setup()
5  {                                   //  Почеток на функцијата setup.
6    myServo.attach(9);                //  Деветтиот пин го конфигурираме како пин на
                                       //  којшто ќе биде поврзан серво-моторот со
7  }                                   //  Крај на функцијата setup.
8  void loop()
9  {                                   //  Почеток на функцијата loop.
   //Со структурата for, аголот на вртење постепено го зголемуваме за два
   //степена,почнувајќи од нула до максималната вредност 180 степени.
10  for(position = 0; position < 180; position += 2) {
11     myServo.write(position);
   //При секое повторување на for структурата, серво-моторот се
   // поместува за два степени во десно.
12 delay(20);                         //  Краткото време на доцнење служи за
                                       //  стабилизација на положбата пред новото
                                       //  завртување.
13   }                                 //  Крај на структурата for.
14 }                                   //  Крај на функцијата loop.
```

Библиотеката ја вклучуваме со притискање на главното мени Sketch→Import Library, по што се отвора паѓачко мени. Библиотеката Servo.h е стандардна библиотека во составот на Arduino развојната средина. Програмата

ја внесуваме со постапката: поврзување на Arduino со компјутер → избор на Arduino микрокомпјутер и сериска порта → верификација → внесување.

7. **Резултати од реализацијата на вежбата**

Веднаш после внесувањето на програмата електричното коло ќе почне да работи така што серво-моторот ќе почне да се врти од лево кон десно, за пола круг. Кога ќе се заврти за 180 степени серво-моторот застанува.
Коментар:_____

8. **Направи промена!**

- Во горната програма може да се додаде уште една for структура со којашто аголот на завртување ќе се врати назад, од 180 степени до нула. Во тој случај од вредноста на аголот треба да се одземе два. На тој начин, серво-моторот ќе се врти напред-назад.

```
14  for(position = 180; position >0; position -= 2) {
15          myServo.write(position);
16          delay(20);
17  }
```

Коментар:_____

- За контрола на аголот на завртување на серво-моторот може да се користи и потенциометар, како влезен уред приклучен на еден од аналогните пинови. Се разбира, во тој случај треба да се направат поголеми промени во горенаведената програма. Во задолжителната loop структура ќе треба да се употребат инструкциите: analogRead() за читање на вредноста на потенциометарот, map() за промена на влезниот опсег во излезен (од 0-1023 кон 0-255) и myServo.write() за придвижување на серво-моторот.
  Коментар:_____

4.5.6. Практична вежба: Поврзување на Arduino Uno R3 со LCD-екран

1. **Цел на вежбата**

Целта на оваа вежба е да го испишеме текстот „ Kompjuterski sistemi “на од LCD-екран и изминатото време изразено во милисекунди.

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Arduino Uno R3 микрокомпјутер
- протоплочка
- LCD-екран, модел GDM1602K
- Тример-потенциометар со отпорност 10KΩ

- жици за поврзување (краткоспојници)
- компјутер со инсталирана Arduino развојна средина

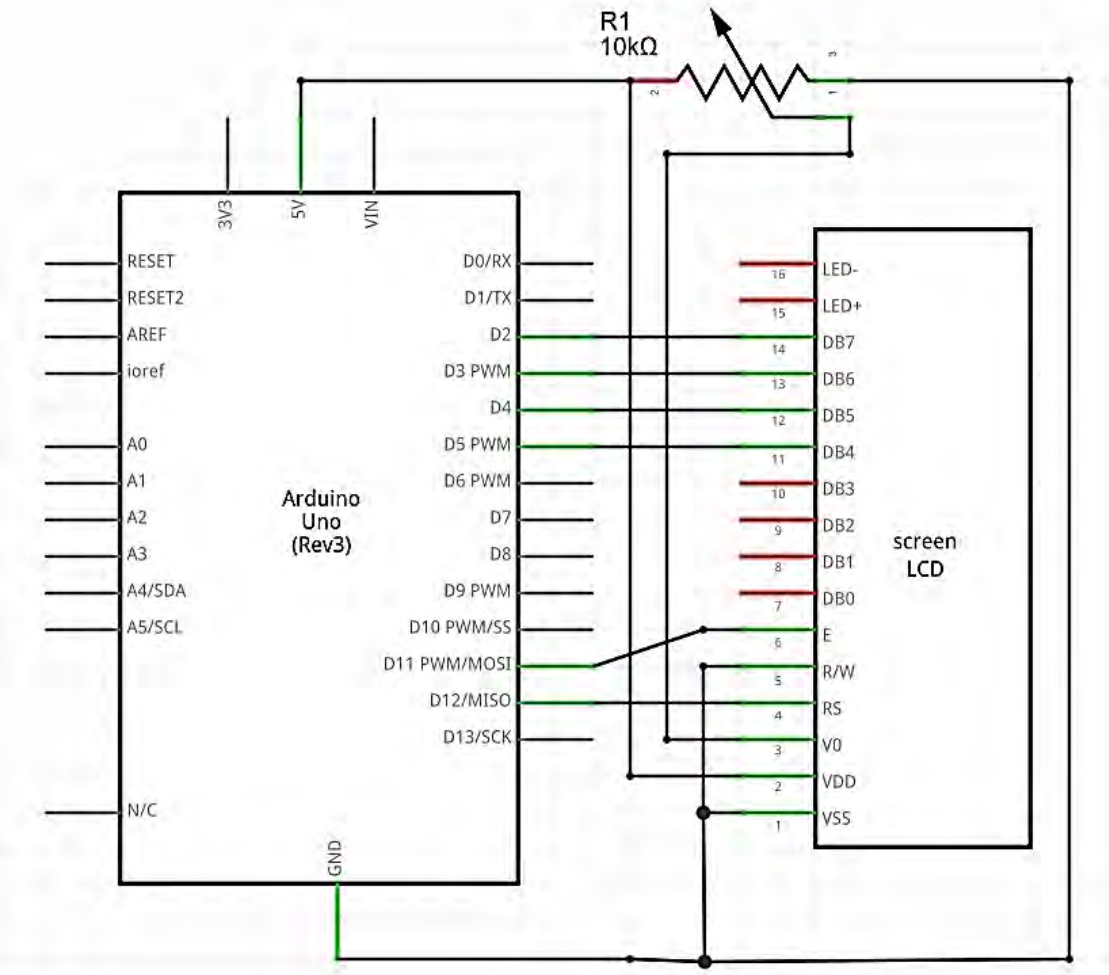
Во наставната единица 4.9. Инструкции за работа со библиотеки во теоретскиот дел на учебникот беа опишани пин-дијаграмот на LCD- екран и неколку инструкциите во состав на LiquidCrystal библиотеката. При изборот на LCD- екран треба да провериме дали контролерот во составот на екранот е компатибилен со LiquidCrystal библиотеката, за што ќе ни послужи техничко-технолошката документација на екранот. На пример, стандардни контролери компатибилни со оваа библиотека се HD44780 или ST7066U. Тример-потенциометарот се користи за нагодување на контрастот.

4. Подготовка за вежбата

Учениците треба да се потсетат на:

- пин-дијаграмот на Arduino Uno R3
- пин-дијаграмот на LCD-екранот, модел GDM1602K
- инструкциите за работа со LCD-екран

5. Опис на електричната шема и начин на поврзување

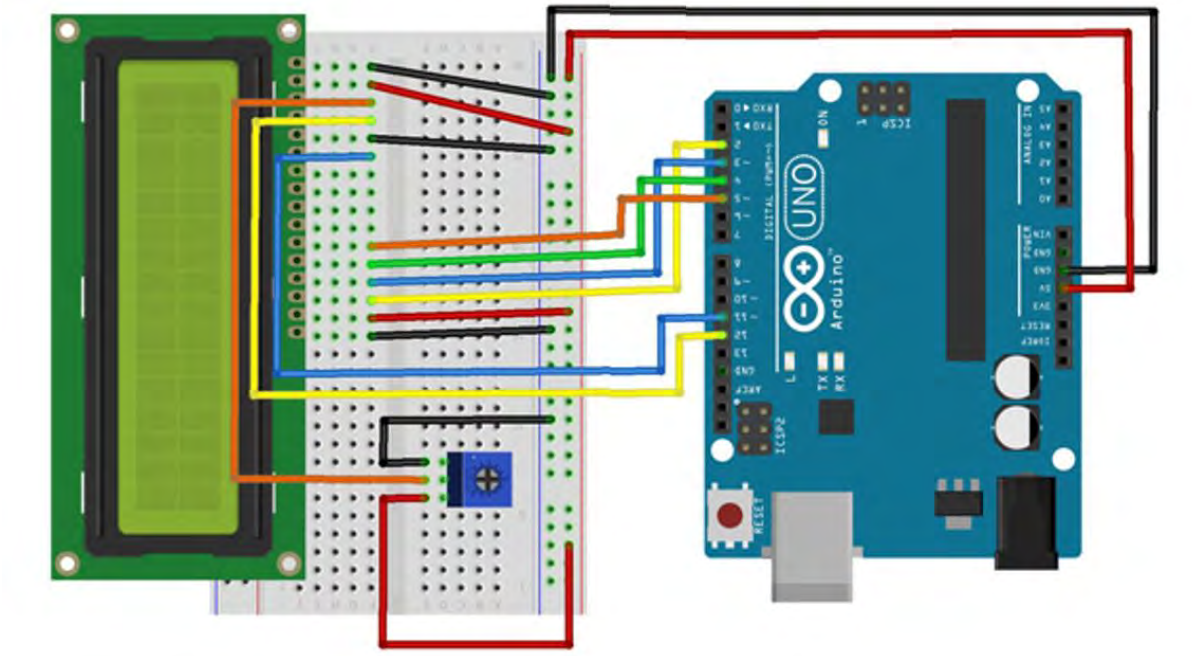


Слика 4.18. Електричната шема за поврзување на LCD-екран со Arduino Uno R3

Пиновите на LCD-дисплејот треба да се поврзат со пиновите на Arduino Uno R3 на следниот начин:

	Приклучоци на LCD-екранот		Пинови на Arduino R3
(1)	RS	→	дигиталниот пин 12
(2)	Enable пинот со	→	дигиталниот пин 11
(3)	D4 пинот со	→	дигиталниот пин 5
(4)	D5 пинот со	→	дигиталниот пин 4
(5)	D6 пинот со	→	дигиталниот пин 3
(6)	D7 пинот со	→	дигиталниот пин 2
(7)	R/W	→	пинот со заземјување
(8)	V_{SS}	→	пинот со заземјување
(9)	V_{CC}	→	пинот со напојувањето од 5 V

На слика 4.18. е прикажан начинот на поврзување на LCD-дисплејот со Arduino Uno R3



Слика 4.34. Монтажна шема за поврзување на LCD-екран со Arduino Uno R3

Пинот V_{EE} е поврзан со вториот приклучок на потенциометарот преку портокаловата краткоспојница. Првиот и третиот приклучок на потенциометарот се поврзани со лентите за напојување и заземјување.

6. Пишување и внесување на програма за Arduino Uno R3

Следува изработката на софтверот во развојната средина.

```
1  #include <LiquidCrystal.h>      //  Се вклучува библиотеката за работа со LCD-
                                   //  дисплеј, која е стандардна библиотека
                                   //
2  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
   // Ги декларираме пиновите на Arduino Uno R3 за поврзување со дисплејот.
3  void setup() {                  //
```

```
4    lcd.begin(16, 2);                  //  Започнува комуникацијата со LCD-дисплејот
                                        //  преку поставување на бројот на колони и
                                        //  редици од симболи .
5    lcd.print("Kompjuterski sistemi");
   // На екранот се печати текстот.
6  }
7  void loop() {
8    lcd.setCursor(0, 1);               //  Се избира активното поле од вкупно 32,
                                        //  преку избор на колона и редица. Да
                                        //  нагласиме, редиците и колоните се бројат
                                        //  почнувајќи од нула
9    lcd.print(millis()/1000);          //  На екранот се печати времето изразено во
                                        //  секунди, изминато од последното
                                        //  ресетирање на Arduino Uno R3
10 }
```

Откако ќе ја напишеме програмата, истата ја внесуваме во меморијата на Arduino Uno со постапката: поврзување на Arduino Uno R3 со компјутер → избор на Arduino микрокомпјутер и сериска порта → верификација → внесување.

7. Резултати од реализацијата на вежбата

Веднаш после внесувањето на програмата електричното коло ќе почне да работи така што на екранот ќе се појави текстот „Kompjuterski sistemi” и измереното време.

Коментар:_____

8. Направи промена!

Во самата развојна средина постојат голем број готови програми за LCD-екран. Тие може да се отворат и да се анализираат со притискање на File→Example→LiquidCrystal. Ќе наброиме неколку од нив: автоматско поместување на текстот, трепкање на текстот, печатење текстуални форми, избор на насока на текстот и др.

Коментар:_____

5. Практични вежби за работа со Rasperry Pi микрокомпјутер на плочка и негово програмирање

5.1. Мерки за заштита и безбедност при работа со Raspberry Pi

За успешна и безбедна реализација на практичните вежби мора да се почитуваат општите правила наведени во наставната единица 1. Мерки за заштита и безбедност при работа во практичниот дел од овој учебник. Но, исто така треба да се познаваат и специфичните карактеристики на електронската опрема кои најчесто се содржат во нејзината техничко-технолошката документација. Најчести причини за оштетување на Raspberry Pi се несоодветно напојување и непознавањето на карактеристиките на поврзаните влезно-излезни компоненти. Најважни мерки за заштита при работа со Raspberry Pi се:

- Изборот на изворот за напојување е мошне важен. Се користи USB адаптер со 5V максимален напон за напојување. Да нагласиме дека Raspberry Pi 4 модел B користи адаптер со USB-C приклучок, а останатите модели микро USB.
- Струјата која треба да ја обезбеди изворот за напојување зависи од моделот на Raspberry Pi и од потрошувачката на електрична енергија на поврзаните влезно-излезни компоненти. Максимално дозволената струја изнесува од 1,5A до 2A. Поради поголемата потрошувачка на електрична енергија на Raspberry Pi 4 модел B му е потребна струја со јачина од 3A.
- GPIO подножјето содржи 4 пинови за напојување и осум пинови за заземјување. Два од пиновите за напојување имаат константен напон од 5V, а другите два 3,3V.
- Максимално дозволениот напон за GPIO влезните пинови изнесува 3,3V.
- Пред почетокот на секоја монтажа, Raspberry Pi треба да се исклучи од неговиот извор за напојување.
- Бидејќи Raspberry Pi е осетлив на статички електрицитет, пред почетокот на монтажата потребно е да се допре некаква метална површина. На таков начин доаѓа до празнење на човековото тело.
- Пред да се вклучи изворот за напојување задолжително да се провери дали сите електронски компоненти се поврзани како што е наведено во функционалната или монтажната шема.
- Пиновите никогаш не смеат да се поврзуваат меѓусебно бидејќи може да дојде до нивно оштетување.

- За нагодување на работната струја да се користат отпорници. На пример, ако лед-диодата ја поврземе директно на пинот за напојување од 5V ќе дојде до загревање и оштетување на истата.
- За поврзување на мотори или други индуктивни потрошувачи да се користат H-мостови (анг. motor driver)

5.2. Практична вежба: Инсталација на Raspbian оперативен систем за Raspberry Pi

5.2.1. Преземање на NOOBS софтвер за инсталација

За инсталација на оперативниот систем Raspbian ќе користиме софтвер со симболично име Нов софтвер од кутија (анг. NOOBS- New Out Of the Box Software). Тоа е специјален софтвер што овозможува избор на еден од повеќе оперативни системи и автоматска инсталација со неколку кликнувања на глувчето. Софтверот за инсталација се сочувува на микро SD-картичка. Веќе спомнавме дека Raspberry Pi нема вградена трајна меморија, туку за чување на податоците и на оперативниот систем се користи **SD-картичка со минимум 16 GB меморија.**

За да го преземеме софтверот NOOBS, во веб-пребарувачот треба да се напише rpf.io/download. На страницата што ќе се отвори, во категоријата Download, прво се притиска на NOOBS со логото Raspberry, а потоа се притиска (click) на Download Zip, што се наоѓа под „NOOBS offline and network install“.

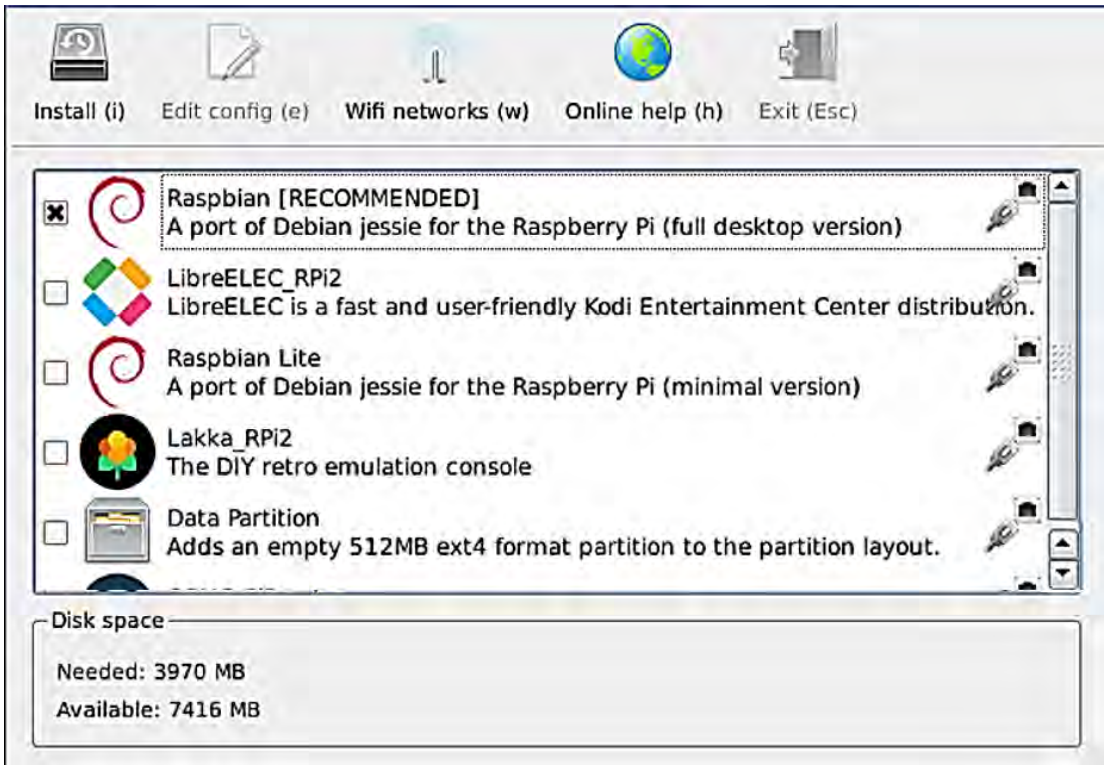
Слика 5.1. Интернет-страница за преземање на инсталациски документ за NOOBS

Ја внесуваме микро SD-картичката во соодветниот слот на персоналниот компјутер. Да напомниме дека доколку не се работи за нова картичка, таа треба да се форматира, при што треба да се води сметка за податоците што биле запишани на неа. Потоа во папката Преземања (анг. Downloads) ја бараме

компресираната датотека со инсталацијата. Овој датотека е позната како архивска датотека (анг. archive) и содржи голем број посебни документи. Со двоен клик го „отпакуваме“ компресираниот документ, ги селектираме сите документи во него и нив ги префрламе, ги копираме во микро SD-картичката.

5.2.2. Инсталација на Raspbian оперативен систем

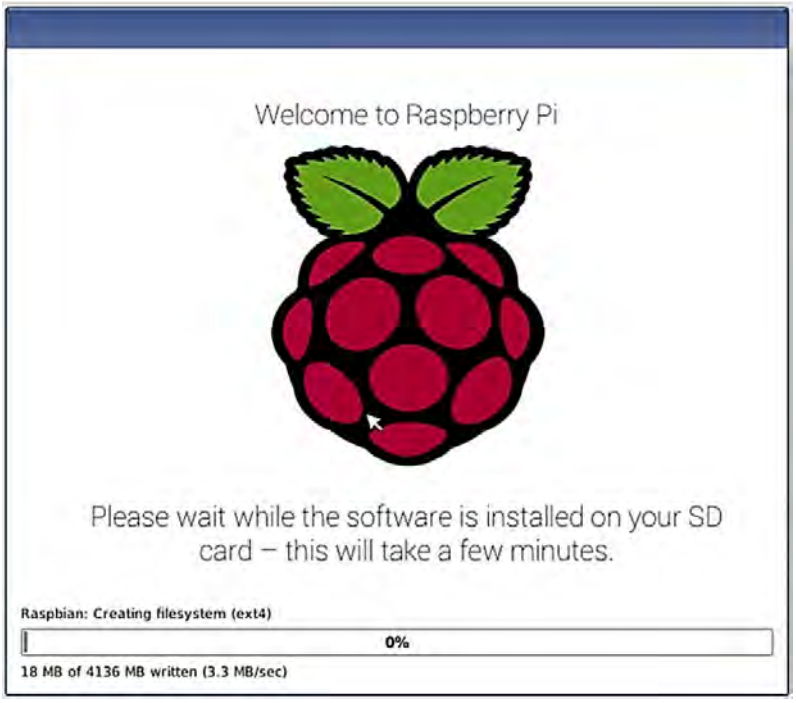
Потоа **микро SD-картичката** со инсталираниот софтвер NOOBS **треба да ја извадиме од персоналниот компјутер и да ја вметнеме во слотот на микрокомпјутерот** Raspberry Pi. Кога картичката за првпат ќе ја поврземе со Raspberry Pi, се појавува менито на NOOBS прикажано на слика 5.2. Со глувчето **го бележиме** квадратчето на оперативниот систем што сакаме да го инсталираме, а тоа во овој случај е **Raspbian**. Ќе забележиме дека иконата Install ќе добие боја, што значи дека оперативниот систем е подготвен за инсталација. [8]



Слика 5.2. Избор на оперативен систем Raspbian

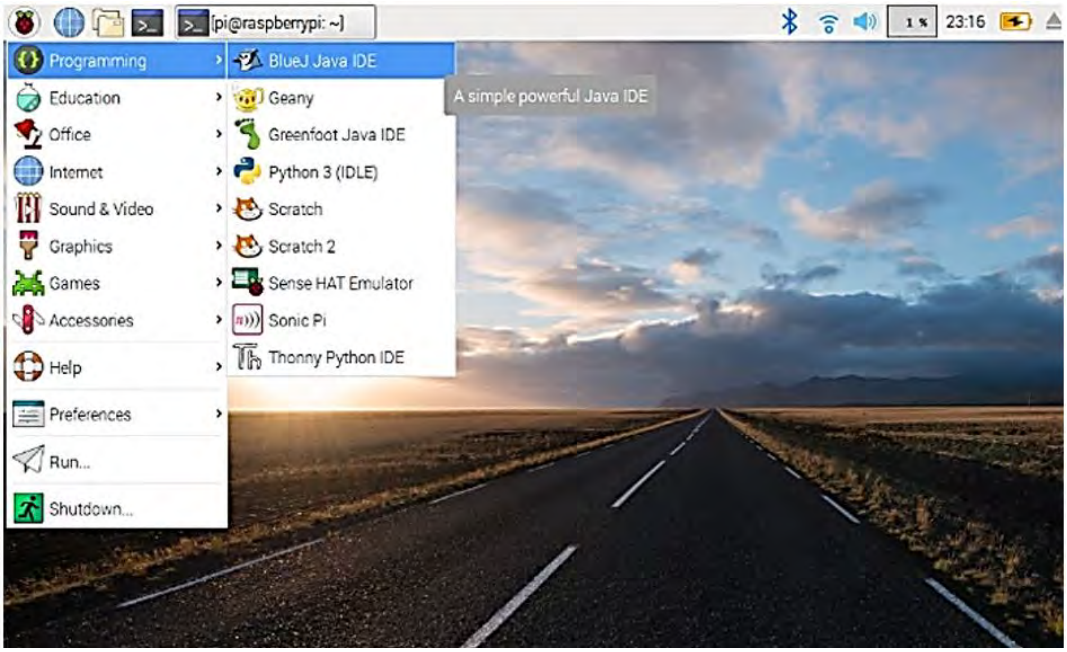
Со притискање на иконата Install, започнува процесот на инсталација, при што се бришат сите податоци од микро SD-картичката, освен софтверот NOOBS. Самата инсталација може да трае од 10 до 30 минути. По завршувањето на инсталацијата притискаме (анг. click) на OK, и уредот Raspberry Pi се рестартира. Само при првото отворање на оперативниот систем се појавуваат boot-пораки, кои траат една до две минути. Конечно се појавува работната површина на Raspbian и во неа прозорецот со којшто се најавува конфигурацијата на оперативниот систем. Кликнуваме на Next. Исто како при инсталацијата на

оперативните системи Windows или Linux, така и при инсталацијата на оперативниот систем Raspbian избираме: земја, јазик, временска зона, лозинка, избор на Wi-Fi мрежа и надградба на оперативниот систем. Во последниот прозорец треба да се притисне Reboot, по што системот се ресетира и се конфигурира со направените измени.



Слика 5.3. Инсталација на оперативен систем Raspbian

Работната површина на оперативниот систем Raspberry изгледа многу слично со работната површина на оперативниот систем Windows.

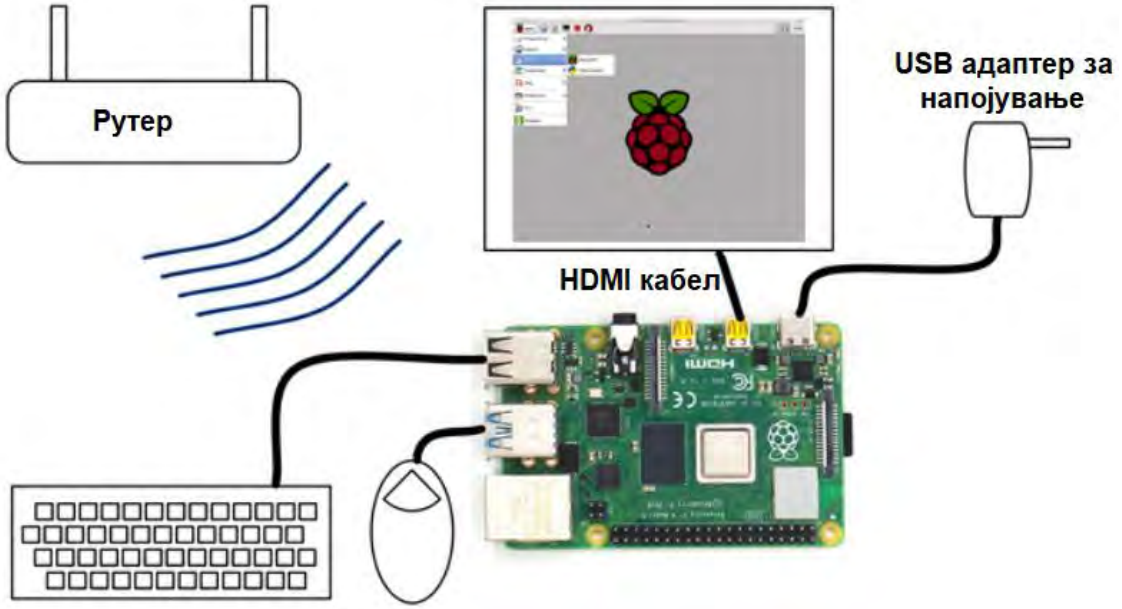


Слика 5.4. Изглед на работната површина на оперативниот систем Raspbian

Во горниот десен агол се наоѓа лентата со алатки за: Bluetooth, избор на мрежа, јачина на звук, мониторинг на процесор, часовник, вклучување и исклучување на USB-уреди. Во левиот горен агол се наоѓа Raspberry и со нејзино притискање се отвора паѓачко мени со повеќе категории. На пример, категоријата **Programming** содржи развојни програми меѓу кои е **Thonny Python IDE**. За пребарување на интернет, во категоријата Internet го избираме Chromium Web Browser. За работа со документи и датотеки, потребен ни е File Manager и за негово активирање ја избираме категоријата Accessories. Во категоријата Office се наоѓа програмата LibreOffice Writer за обработка на текст. Може да се направи инсталација на дополнителен софтвер со притискање на Recommended Software во категоријата Preferences. За додавање нов хардвер (на пример, софтвер за работа со камера, анг. Raspberry Pi Camera Module) или за поврзување во мрежа (анг. VNC- Virtual Network Computer) може да се употреби опцијата Interface, до која доаѓаме преку притискање (click) на Raspberry Pi Configuration во категоријата Preferences.

5.3. Практични вежби за програмирање на Raspberry Pi 3B+ во програмскиот јазик Python

За успешна реализација на практичните вежби, потребни се следните предзнаења: инсталација на оперативен систем Raspbian, програмирање во програмски јазик Python, користење на библиотеката GPIOZERO и поврзување и карактеристики на основни влезно-излезни уреди.



Слика 5.5. Raspberry Pi систем

При изведувањето на вежбите, Raspberry Pi 3B+ треба да биде поврзан со тастатура за внесување на текстот на програмскиот код и со монитор за

прикажување на текстуалните резултати добиени по извршувањето на програмата. На слика 5.5. е прикажан еден комплетен Raspberry Pi систем. Да се потсетиме, за поврзување со тастатура се користи USB-приклучокот, а за поврзување со монитор HDMI-приклучокот. Исто така, не постои ограничување во изборот на модел на Raspberry Pi, бидејќи дадените кодови се компатибилни со сите модели.

Во секоја практична вежба, дадена е шемата за поврзување на Raspberry Pi 3B+ со влезно-излезните уреди, преку употреба на протоплочка. Потоа следува програмскиот код со соодветни коментари. Со програмските кодови може да се експериментира преку менување на вредностите на некои параметри: брзина на вртење на моторот, фреквенција на трепкање на диодата, боја на RGBLED-диода, прикажан текст итн.

5.3.1. Практична вежба: Вклучување на лед-диода со тастер

1. **Цел на вежбата**

Цел на вежбата е поврзување на лед-диода и тастер со Raspberry Pi 3B+ со употреба на протоплочка, пишување на програма во Python програмскиот јазик, внесување на програмата во Raspberry Pi преку алатките на Thonny Python интегрираната развојна средина и испитување на работата на електричното коло.

2. **Време за реализација:** 1 наставен час

3. **Потребни компоненти за реализација на оваа вежба се:**

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем. Системот е составен од Raspberry Pi 3B+ со адаптер за напојување, тастатура, глувче и монитор.
- протоплочка
- тастер за печатена плочка
- лед-диода
- отпорник со отпорност R1=220Ω
- жици за поврзување (краткоспојници)

4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

- пин-дијаграм на GPIO подножјето
- начинот на поврзување на лед-диода и тастер на протоплочка. Истиот е објаснет во упатството за работа со протоплочка.
- инструкциите од GPIO библиотеката класи Led и Button за работа со тастер и лед-диода.

- практичната вежба 4.8.1.1. Вклучување на лед-диода со тастер со примена на Arduino Uno R3 и да се направи споредба на електричната шема и програмскиот код

5. Опис на електричната шема и начин на поврзување

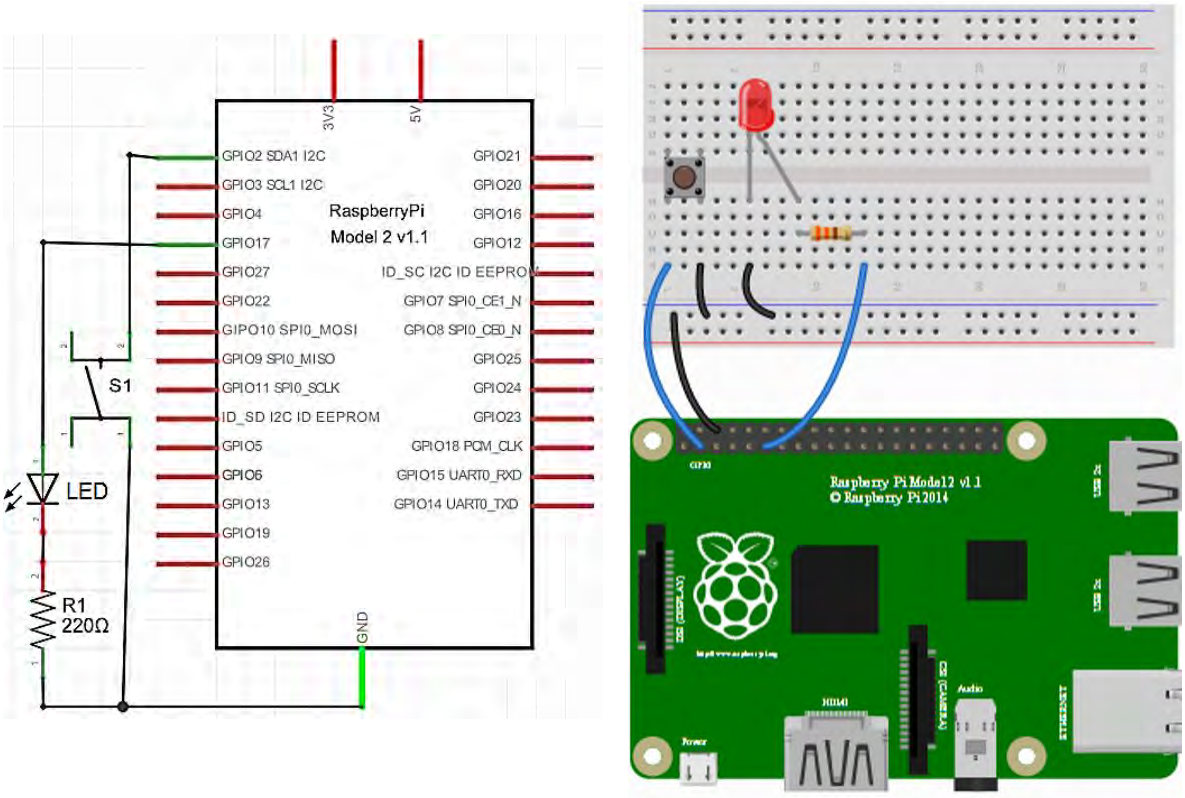
На слика 5.6. се прикажани функционалната и монтажната шема за реализација на оваа вежба.

(1) Тастерот се поставува на средината од протоплочката. Десниот приклучок на тастерот се поврзува со лентата за заземјување, а левиот приклучок со GPIO-пинот број 2.

(2) Катодата на лед-диодата се поврзува со лентата за заземјување, а анодата со GPIO-пинот број 17.

(3) Лентата за заземјување на протоплочката се поврзува со GND пинот на Raspberry Pi 3B+

Електричните шеми на слика 5.6. и слика 4.13. од практичниот дел на учебникот се различни. Кај Raspberry Pi тастерот не е поврзан со лентата за напојување на протоплочката. Имено кога тастерот не е притиснат тогаш GPIO-пинот број 2 не е поврзан, колото е отворено. Кога тастерот е притиснат GPIO-пинот број 2 се поврзува со лентата за заземјување. Ниското логичко ниво (анг. High) на GPIO-пинот број 2 е показател дека тастерот бил притиснат. Лед-диодата свети кога GPIO-пинот број 2 е на високо логичко ниво.



Слика 5.6. Поврзување на лед-диода и тастер со Raspberry Pi 3B+

6. Пишување и извршување на програма за Raspberry Pi 3B+

На слика 4.32. од теоретскиот дел на учебникот е прикажан изгледот на интегрираната развојна средина Tonny Python IDE која ќе ја користиме за пишување на програмата и нејзино извршување. Отвораме нов проект со алатката New, го пишуваме програмскиот код во уредувачот на текст и откако ќе ја напишеме ја зачувуваме (анг. Save). Кога ќе ја зачуваме називот на програмата се менува од untitled во името што сме го задале. Делот именуван како Shell служи за прикажување на текстуални пораки со употреба на инструкцијата print(). Во малата заграда се пишува текстот кој треба да се прикаже, напишан меѓу наводници. Подоле е даден програмскиот код за реализација на оваа вежба.

```
1  from gpiozero import LED, Button   # Вклучуваме два дела од библиотеката
                                      # GPIOZERO за работа на лед-диода и
                                      # тастер.
2  from signal import pause           # Вклучуваме дел од библиотеката signal за
                                      # внесување време на доцнење.
3  led = LED(17)                      # Конфигурираме пин за поврзување на
                                      # диодата.
4  button = Button(2)                 # Конфигурираме пин за поврзување на
                                      # тастерот.
5  button.when_pressed = led.on       # Ја вклучуваме лед-диодата кога тастерот е
                                      # притиснат.
6  button.when_released = led.off     # Ја исклучуваме лед-диодата кога тастерот
                                      # е отпуштен.
7  pause()                            # Внесуваме време на доцнење. Стандардно
                                      # време на доцнење е една секунда. Времето
                                      # го менуваме со внесување вредност во
                                      # малата заграда.
```

За да се изврши напишаната програма потребно е да ја притиснеме иконата Run. Споредбено со Arduino платформата, Raspberry Pi е микрокомпјутер со свој оперативен систем. Интерпретерот ја преведува програмата инструкција по инструкција дури во текот на самото пишување на програмата и не се потребни алатките Verify и Upload.

7. Резултати од реализацијата на вежбата

Веднаш после стартот на програмата треба да го притиснеме тастерот и лед-диодата ќе светне. Кога тастерот ќе го отпуштиме лед-диодата ќе престане да свети

Коментар:_____

8. Направи промена!

- Сакаме лед-диодата да свети определено време и после отпуштањето на тастерот. Која инструкција ќе се употреби и каде ќе се вметне истата?

Коментар_____

- Сакаме лед-диодата да свети кога тастерот е отпуштен и да не свети кога тастерот е притиснат. Кои промени треба да се направат во програмскиот код?

Коментар_____

5.3.2. Практична вежба:Семафор

1. Цел на вежбата

Светлата на семафорот се вклучуваат и исклучуваат според точно определен тајминг и циклусот е бесконечен односно непрекинато се повторува. Цел на оваа практична вежба е управување со светлата на семафорот преку употреба на Raspberry Pi 3B+. Во gpiozero библиотеката постои посебна класа токму за работа со семафор (анг. TrafficLights) и истата значајно го поедноставува пишувањето на програмскиот код.

2. Време за реализација: 2 наставни часа

3. Потребни компоненти за реализација на оваа вежба се:

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем
- протоплочка
- три лед-диоди
- три отпорници со отпорност R=220Ω
- жици за поврзување (краткоспојници)

4. Подготовка за вежбата

Од хардверот на Raspberry Pi 3B+ учениците треба да се потсетат на пин-дијаграмот на GPIO подножјето и начинот на поврзување на лед-диоди на протоплочка. Истиот е објаснет во упатството за работа со протоплочка.

Исто така, пред почетокот на вежбата учениците треба да се запознаат со класата TrafficLights од библиотеката gpiozero и инструкцијата sleep од библиотеката time.

Пример 4.54.

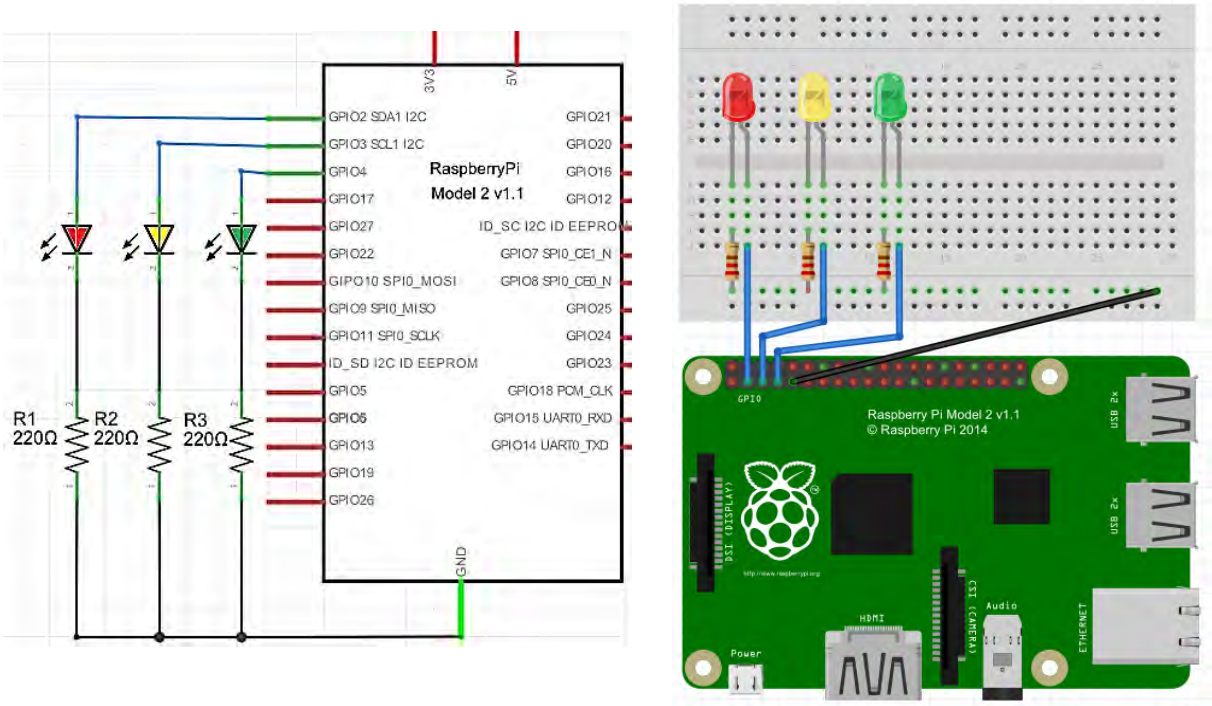
```
1 semafor=TrafficLights(25, 8, 7)
```

Пример 4.54. е пример за декларација на објект кој ќе биде носител на класата TrafficLights(). Во програмата на оваа практична вежба декларирам е еден објект од оваа класа зошто постои само еден семафор. Доколку сакаме да го проширим системот со уште еден семафор, кој ќе управува со сообраќајот по втората улица од раскрсницата тогаш ќе треба да декларираме уште еден објект од класата TrafficLights(). Броевите 25, 8 и 7 во малата зграда се броеви на GPIO пиновите на кои се поврзани трите светла односно трите диоди. Од библиотеката за контрола на време се користи само инструкцијата sleep(). Во

малата заграда се наведува времето изразено во секунди за кое системот ќе биде во мирување.

5. Опис на електричната шема и начин на поврзување

Поврзувањето на лед-диодите со Raspberry Pi 3B+ е многу едноставно. При поставувањето на лед-диодите на протоплочката треба да внимаваме нивните приклучоци да ги поставиме во контактни точки кои припаѓаат на различни проводници. Катодата на секоја лед-диода е поврзана со лентата за заземјување преку отпорник од 220Ω. Анодата на црвената лед-диода е поврзана со GPIO-пинот број 2, анодата на жолтата лед-диода со GPIO-пинот број 3 и анодата на зелената лед-диода со GPIO-пинот број 4. Да не заборавиме пинот GND на Raspberry Pi 3B+ да го поврземе со лентата за заземјување преку црна краткоспојница.



Слика 5.7. Поврзување на семафор со Raspberry Pi 3B+

6. Пишување и внесување на програма за Raspberry Pi 3B+

Постапката за пишување и внесување на програмата е иста како во претходната вежба. Прво отвораме нов проект, ја пишуваме во уредникот за текст, ја сочувуваме и именуваме и на крај ја извршуваме со притискање на алатката Run. Подоле е даден програмскиот код за оваа практична вежба.

```
1  from gpiozero import TrafficLights      # Се вклучува делот од библиотеката
                                           # GPIOZRO за работа со семафор.
2  from time import sleep                  # Со инструкцијата sleep го внесуваме
                                           # уредот во режим на мирување.
3  lights = TrafficLights(2, 3, 4)         # Конфигурација на пиновите за
                                           # црвената, жолтата и за зелената лед-
                                           # диода.
```

```
4  lights.green.on()          # Се вклучува зелената лед-диода.
5  while True:                # Започнуваме циклус што бесконечно
                              # се повторува.
6      sleep(10)              # Зелената лед-диода свети 10s .
7      lights.green.off()     # Ја исклучуваме зелената лед-диода.
8      lights.amber.on()      # Ја вклучуваме жолтата.
9      sleep(1)               #
10     lights.amber.off()     # Ја исклучуваме жолтата.
11     lights.red.on()        # Вклучуваме црвена лед-диода.
12     sleep(10)              # Таа свети десет секунди.
13     lights.amber.on()      # Се вклучува жолтата лед-диода.
14     sleep(1)
15     lights.red.off()
16     lights.amber.off()
17     lights.green.on()
       sleep(2)
```

7. **Резултати од реализацијата на вежбата**

Веднаш после притискањето на алатката Run во лентата со алатки во Thonny Python интегрираната развојна средина зелената лед-диода ќе почне да свети 10 секунди, после што светнува жолтата лед-диода за една секунда, па свети црвената лед-диода со времетраење 10 секунди. На крајот светнуваат црвената и жолтата лед-диода заедно со времетраење од две секунди и повторно се враќаме на зелената лед-диода. Циклусот се повторува бесконечно.
Коментар_____

8. **Направи промена!**

Учениците може да го менуваат времето на светење на лед-диодите. Исто така може да додадат уште еден семафор кој ќе работи инверзно во однос на првиот и за таа цел треба да го променат програмскиот код.
Коментар_____

5.3.3. Практична вежба: Времето на реакција

1. **Цел на вежбата**

Целта на оваа вежба е да се утврди кој од два тастери ќе биде прв притиснат. Лед-диодата светнува и го означува почетокот на „трката“. Резултатот се прикажува во делот именуван како Shell во облик на текстуална порака. [7]

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем
- протоплочка
- една лед-диода

- еден отпорник со отпорност R=220Ω
- два тастери
- жици за поврзување (краткоспојници)

4. Подготовка за вежбата

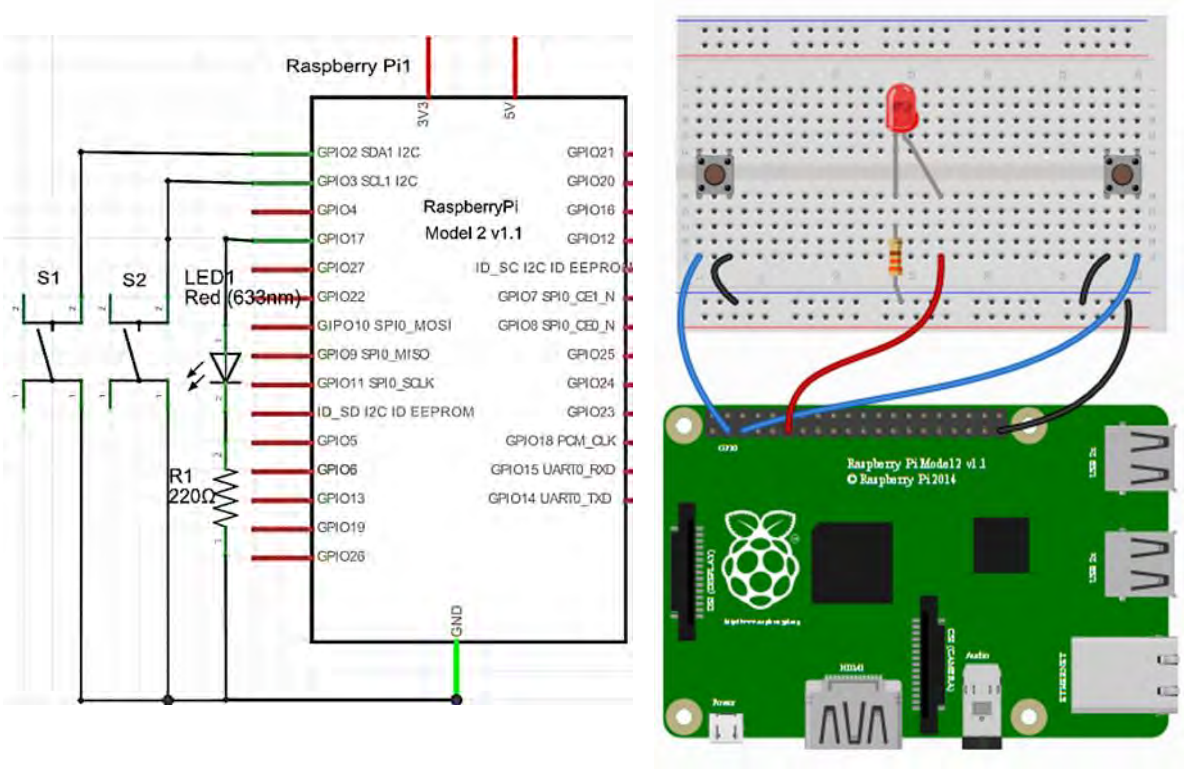
Слично како во претходната вежба учениците треба да се потсетат на пин-дијаграмот на GPIO подножјето и начинот на поврзување на лед-диоди на протоплочка.

Во оваа вежба за првпат ќе биде употребена инструкцијата uniform од библиотеката random. Оваа инструкција служи за случаен избор (анг. random) на децимален број во опсегот од 0 до 5 и истата е искористена за избор на време (анг. time) во инструкцијата sleep(time). Промената на времето се користи како елемент на изненадување за двајцата играчи бидејќи нема да се знае кога точно ќе светне диодата.

Исто така, за првпат ќе биде употребен серискиот монитор (анг. Shell) и инструкцијата print() за прикажување на текстуална порака со информација за победникот, прв или втор играч.

5. Опис на електричната шема и начин на поврзување

На слика 5.8. се прикажани функционалната и монтажната шема за реализација на оваа вежба.



Слика 5.8. Поврзување на два тастери и лед-диода со Raspberry Pi 3B+

Поврзувањето е многу едноставно. Тастерите се поставуваат на средината од протоплочката и десните приклучоци се поврзани со лентата за заземјување , а левите приклучоци со GPIO пиновите број 2 и 3 со сини

краткоспојници. Приклучоците на лед-диодата се поставуваат во контактни точки кои припаѓаат на две различни проводници од протоплочката и катодата преку отпорникот се поврзува со лентата за заземјување, а анодата директно со GPIO пинот број 17 со црвена краткоспојница. Да не заборавиме лентата за заземјување да ја поврземе со еден од GND пиновите на Raspberry Pi.

6. Пишување и внесување на програма за Raspberry Pi 3B+

Ја пишуваме програмата во развојната средина Thonny Python IDE. Да се потсетиме истата е дел од оперативниот систем Raspbian и за нејзино отворање притискаме на опцијата Programming→Tools во менито на самиот оперативен систем.

```
1  from gpiozero import Button,
   LED
2  from time import sleep
3  import random                    #  Библиотеката random дава можност
                                    #  за избор на случајни вредности.
4  led = LED(17)
5  player_1 = Button(2)             #  Конфигурација на пиновите за
                                    #  поврзување на тастери и истовремено
6  player_2 = Button(3)             #  задавање симболични имиња на
                                    #  двата тастера.
7  time = random.uniform(5, 10)     #  Дефинираме променлива со
                                    #  симболично име time и случајна
                                    #  вредност во опсегот од 5 до 10.
8  sleep(time)                      #  Системот е во мирување за време од
                                    #  5 до10 секунди.
9  led.on()                         #  По истекот на тоа време се вклучува
                                    #  диодата.
10 while True:                      #  Започнува циклус.
11    if player_1.is_pressed:       #  Се прикажува текстуална порака
                                    #  доколку тастерот поврзан со пинот
12       print("Player 1 wins!")    #  број 2 е прв притиснат.
13       Break                      #  Ако е исполнет горниот услов,
                                    #  излегуваме од циклусот.
14    if player_2.is_pressed:       #  Се прикажува текстуална порака
                                    #  доколку тастерот поврзан со пинот
15       print("Player 2 wins!")    #  број 3 е прв притиснат.
16       Break                      #  Излегуваме од циклусот.
17 led.off()                        #  Лед-диодата се исклучува и се
                                    #  враќаме на почетокот на програмата.
```

Со самото пишување на програмата истата се внесува во меморијата на Raspberry Pi.

7. **Резултати од реализацијата на вежбата**

За да ја провериме исправноста на вежбата треба само да притиснеме Run во летата со алатки на развојната средина и двајцата играчи да ги притиснат тастерите откако ќе светне лед диодата. Во мониторот на развојната средина треба да се појави текстуална порака.Меѓу две притискања на тастерите е воведено време со случајна вредност во опсегот од 5 до 10 секунди. Бидејќи се работи за бесконечен циклус while во второто повторување на циклусот може да се добие поинаков резултат односно поинаква текстуална порака.
Коментар:_____

8. **Направи промена!**

Во вежбата двајцата играчи можат да го внесат своето име и истото да стои во тесктуалната порака. За таа цел треба да се направи промена во програмскиот код. После програмскиот ред 6 односно после конфигурацијата на пиновите за поврзување на тастерите треба да се додадат следните инструкции.

```
lev_igrac=input (‘Vnesi go imeto’)
desen_igrac=input (‘Vnesi go imeto’)
```

Промена треба да се направи и во инструкциите print() и наместо print("Player 1 wins!") и print("Player 1 wins!") да се напише print("lev_igrac wins!") и print("desen_igrac wins!")
Коментар:_____

5.3.4. Практична вежба: Вклучување и исклучување на лед-диода со фотоотпорник

1. **Цел на вежбата**

Оваа практична вежба е слична со вежбата 4.8.1.4. Фотоотпорник за контрола на интензитетот на светлина преку Arduino Uno R3. Но, Raspberry Pi нема вграден аналогно-дигитален претворувач како што има Arduino Uno. Поради тоа не можеме да го менуваме интензитетот на светлината на лед-диодата како што правевме во практичната вежба со Arduino Uno. Наместо да се менува интензитетот на светлината на лед-диода таа само ќе се вклучува и исклучува, во зависност од тоа дали е светло или мрак, дали е ден или ноќ.

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем
- протоплочка
- една лед-диода
- еден отпорник со отпорност R=220Ω
- фотоотпорник

- кондензатор со капацитивност 1μF
- жици за поврзување (краткоспојници)

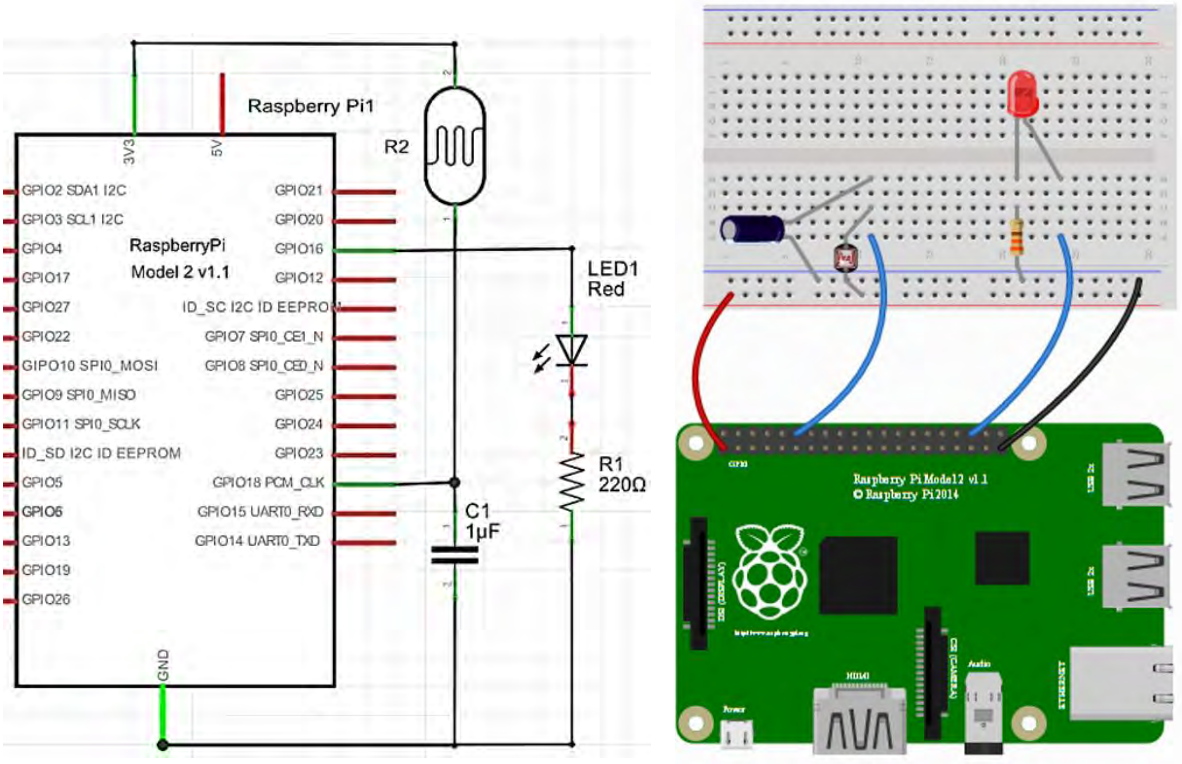
4. Подготовка за вежбата

Учениците треба да се потсетат на:

- работа на редно RC коло (наставен предмет Електротехника)
- пин-дијаграм на GPIO подножје и начин на поврзување на лед-диода на протоплочка.
- промената на отпорноста на фотоотпорникот во зависност од интензитетот на светлината и инструкциите од класата LightSensor од библиотеката gpio. (наставна единица 4.15.4. Оптички сензор или фотоотпорник од теоретскиот дел на учебникот)

5. Опис на електричната шема и начин на поврзување

Функционалната и монтажна шема за реализација на оваа вежба е дадена на слика 5.9..



Слика 5.9. Поврзување на лед-диода и фотоотпорник со Raspberry Pi 3B+

Во вежбата 4.8.1.4. Фотоотпорник за контрола на интензитетот на светлина фотоотпорникот и отпорникот од 10KΩ формираат напонски делител. Во оваа практична вежба наместо напонски делител ќе користиме редно RC коло составено од еден отпорник и кондензатор, редно поврзани со напојувањето од 3,3V. Едниот приклучок на фотоотпорникот се поврзува со лентата за напојување, а другиот приклучок се поврзува со GPIO пинот број 18. На истиот GPIO пин се поврзува кондензатор со капацитет од 1 μF, а другиот пин на кондензаторот се поврзува со лентата за заземјување. Ако фотоотпорникот има

мала отпорност тогаш напонот на кондензаторот побрзо се менува и обратно. Raspberry Pi 3B+ го мери времето потребно напонот на кондензаторот да се зголеми од нула до вредност 1,34V што претставува напон на праг (види слика 4.35. во теоретскиот дел на учебникот) и според измереното време детектира мрак или светло. Поврзувањето на лед-диодата е идентично како во претходните практични вежби со таа разлика што сега е употребен GPIO пинот број 16. Лентата за напојување ја поврзуваме со пинот 3,3V на Raspberry Pi, а лентата за заземјување со пинот GND.

6. **Пишување и внесување на програма за Raspberry Pi 3B+**

Програмата ја пишуваме во развојната средина Thonny Python IDE. Од библиотеката gpio ја преземаме класата со симболично име LightSensor која содржи инструкции за работа со фотоотпорник. Ќе ги употребиме инструкциите sensor.when_dark и sensor.when_light. Инструкцијата sensor.when_dark се извршува кога Raspberry Pi 3B+ детектира мрак (бавна промена на напонот на кондензаторот), а инструкцијата sensor.when_light се извршува кога Raspberry Pi 3B+ детектира светло (брза промена на напонот на кондензаторот),

```
1  from gpiozero import LightSensor, LED   # Од библиотеката GPIOZERO ги
                                           # повикуваме потребните класи.
2  from signal import pause
3  sensor = LightSensor(18)                # Ги конфигурираме пиновите за
4  led = LED(16)                           # поврзување на фотоотпорникот и лед-
                                           # диодата.
5  sensor.when_dark = led.on               # Лед-диодата се вклучува кога сензорот
                                           # ќе детектира мрак .
6  sensor.when_light = led.off             # Лед-диодата се исклучува кога
                                           # сензорот ќе детектира мрак .
7  pause()                                 # Се внесува време на доцнење.
```

Откако ќе ја напишеме програмата во развојната средина притискаме Run во менито со алатки.

7. **Резултати од реализацијата на вежбата**

Кога фотоотпорникот е изложен на светлина лед-диодата не свети. Кога ќе го покриеме фотоотпорникот со некаков предмет лед-диодата почнува да свети.

Коментар:_____

5.3.5. Практична вежба: Проверка на растојание до најблизок објект

1. **Цел на вежбата**

Цел на оваа вежба е примена на сензор за растојание (HC-SR04), негово поврзување со Raspberry Pi 3B+ и пишување на програма која измереното

растојание ќе го прикаже во облик на текстуална порака на Shell мониторот во развојната средина Thonny Python IDE.

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем
- протоплочка
- сензор за растојание (HC-SR04)
- жици за поврзување (краткоспојници)

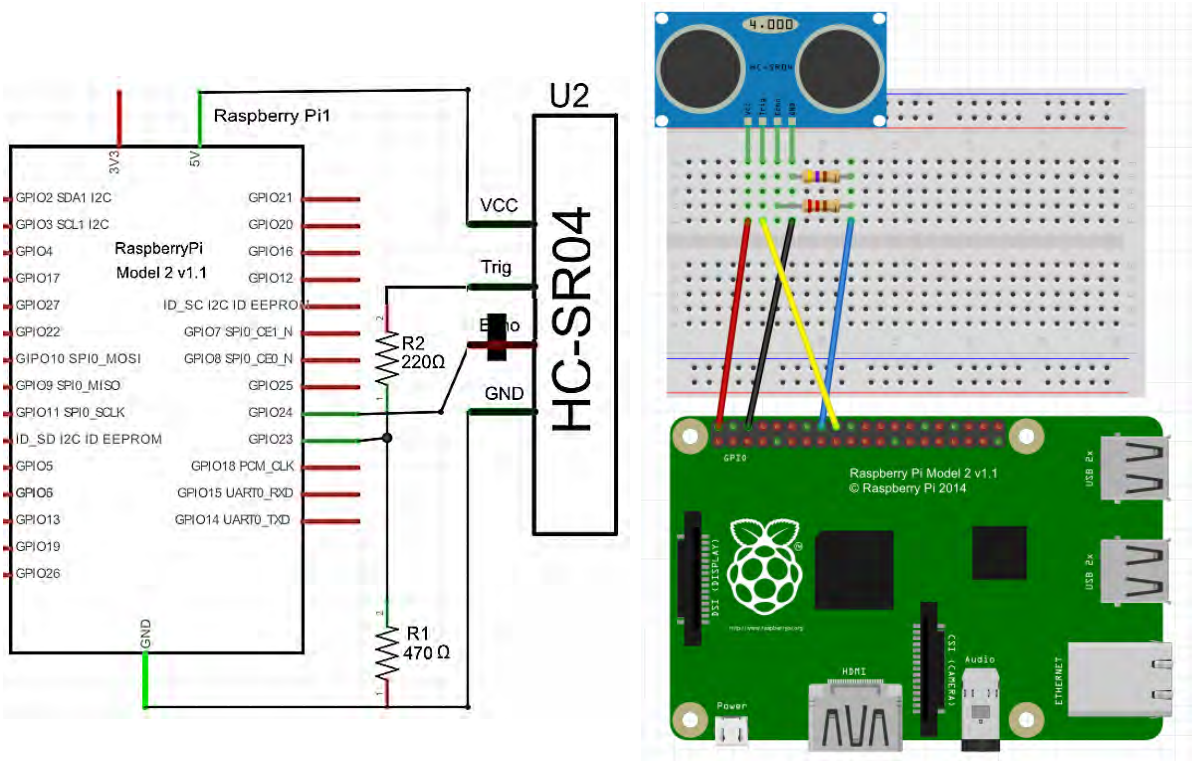
4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

- Практичната вежба 4.2. Упатство за работа со протоплочка од практичниот дел на учебникот
- пин-дијаграм на GPIO подножје (слика 4.34. во теоретскиот дел)
- наставната единица 4.15.3. Сензор за растојание (HC-SR04) од теоретскиот дел на учебникот

5. **Опис на електричната шема и начин на поврзување**

Функционалната и монтажна шема за реализација на оваа вежба е дадена на слика 5.10.



Слика 5.10. Поврзување на сензор за растојание со Raspberry Pi 3B+

За поврзување на сензорот за растојание со Raspberry Pi 3B+ потребни се два GPIO пинови, еден за тригерот и еден за ехото.

(1) Приклучокот за заземјување на сензорот го поврзуваме со GND пинот на Raspberry Pi
(2) Приклучокот Trig на сензорот го поврзуваме со GPIO пинот број 24 на Raspberry Pi.
(3) Едниот крај на отпорникот со отпорност 220Ω се поврзува со Echo приклучокот на сензорот
(4) Едниот крај на отпорникот со отпорност 470Ω се поврзува со GND приклучокот.
(5) Другите два краја на двата отпорници заедно се поврзани со GPIO пинот на Raspberry Pi. На ваков начин се формира напонски делите кој го намалува напонот на излез од приклучокот Echo. Овој напон изнесува 5V, а знаеме дека максималниот влезен напон за Raspberry Pi 3B+ изнесува 3,3V. Во спротивно може да дојде до оштетување на Raspberry Pi.
(6) Приклучокот VCC на сензорот се поврзува со пинот за напојување 5V на Raspberry Pi.

6. Пишување и внесување на програма за Raspberry Pi 3B+

Програмата ја пишуваме во развојната средина Thonny Python IDE по веќе познатата постапка: отворање на нов проект→пишување во уредникот за текст→ сочувување на програмата под симболично име избрано од корисникот. Од библиотеката gpio ја преземаме класата со симболично име DistanceSensor која содржи инструкции за работа токму со сензорот за растојание (HC-SR04).

```
1  from gpiozero import DistanceSensor      # Ja povikuvame klasata so instrukcii za
                                            # rabota so senzorot za rastojanie
                                            # (HC-SR04).
2  from time import sleep
3
4  sensor = DistanceSensor(23, 24)          # Konfiguracija na pinovi
5
6  while True:                              # Započnuva beskonečen ciklus
7      print('Rastojanieto do najbliskiot predmet iznesuva', sensor.distance, 'm')
   #Prikažuvanje na izmereното rastojanie
8      sleep(1)                             # Sistemot e vo mirување 1 sekunda.
```

Програмата е многу едноставна и малечка. Тоа се должи пред сè на употребата на библиотеката gpio и класата DistanceSensor. Корисникот не мора да користи инструкции за читање и пишување како што тоа беше случај со Arduino Uno платформата. Со самото повикување на класата DistanceSensor ова автоматски се врши. Сите хардверски карактеристики на сензорот се земени предвид и креиран е софтвер и истиот е содржан во библиотеката gpio односно класата DistanceSensor.

7. Резултати од реализацијата на вежбата

Пред сензорот за растојание поставуваме одреден предмет. Максималното растојание изнесува 1 метар. Откако ќе ја напишеме и сочуваме

програмата притискаме Run во менито со алатки и веднаш во полето на серискиот монитор Shell се појавува текстуалната порака со информација за растојанието од сензорот до поставениот објект. Објектот може да се поместува и да ја провериме веродостојноста на добиените податоци.
Коментар:_____

5.3.6. Практична вежба: Промена на насока на мотор на еднонасочна струја

1. **Цел на вежбата**

Оваа вежба е уште еден доказ колку е лесно да се програмира Raspberry Pi 3B+ благодарение на библиотеките и нивното инструкциско множество, поради кои програмирањето станува многу едноставно и разбирливо. Целта на оваа вежба е да се менува насоката на вртење на DC-моторот. Да се потсетиме моторот за еднонасочна струја не се поврзува директно со Raspberry Pi туку се користи H-мост како интерфејс-компонента.

2. **Време за реализација:** 2 наставни часа

3. **Потребни компоненти за реализација на оваа вежба се:**

- Raspberry Pi 3B+ систем со инсталиран Raspbian оперативен систем
- протоплочка
- мотор на еднонасочна струја со напојување 3V
- интегрирано коло SN754410
- жици за поврзување (краткоспојници)

4. **Подготовка за вежбата**

Учениците треба да се потсетат на:

- пин-дијаграм на GPIO подножје на Raspberry Pi 3B+ (слика 4.34. во теоретскиот дел)
- наставната единица 4.16.4. Мотор на еднонасочна струја од теоретскиот дел на учебникот
- пин-дијаграмот на интегрираното коло SN754410 (слика 4.44. во теоретскиот дел)

5. **Опис на електричната шема и начин на поврзување**

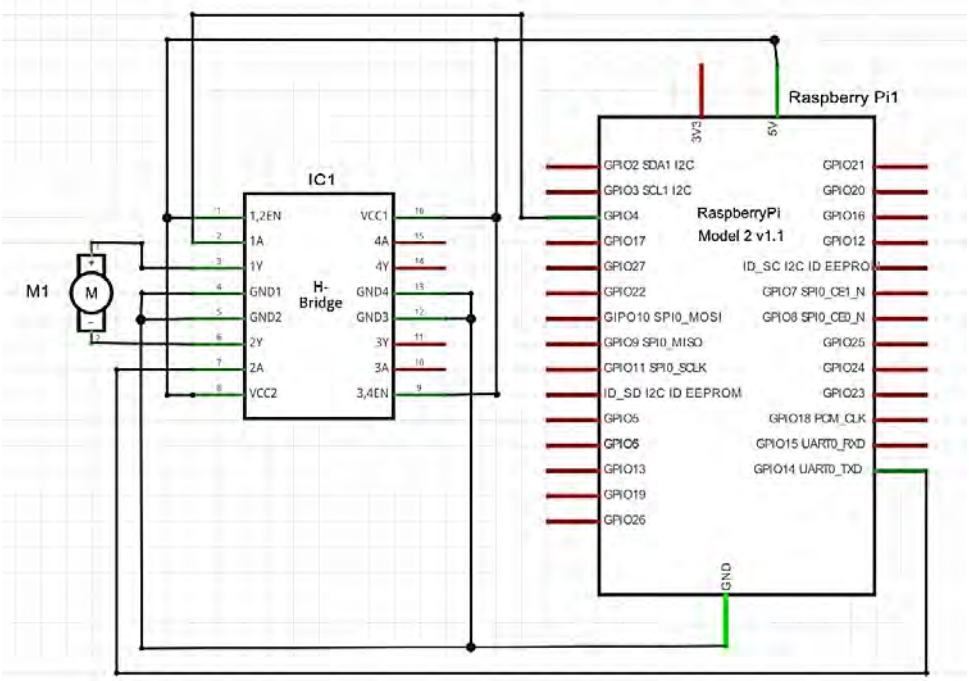
На слика 5.11. и 5.12. е прикажана функционалната и монтажната шема за реализација на оваа вежба.

(1) Интегрираното коло SN754410 го поставуваме на средината на протоплочката внимавајќи да не ги извиткаме неговите пинови. Со црвени краткоспојници ги поврзуваме пиновите за напојување и оспособување (анг. Vcc, Enable) со лентата за напојување, а со црни краткоспојници ги поврзуваме четирите GND пинови со лентата за заземјување.

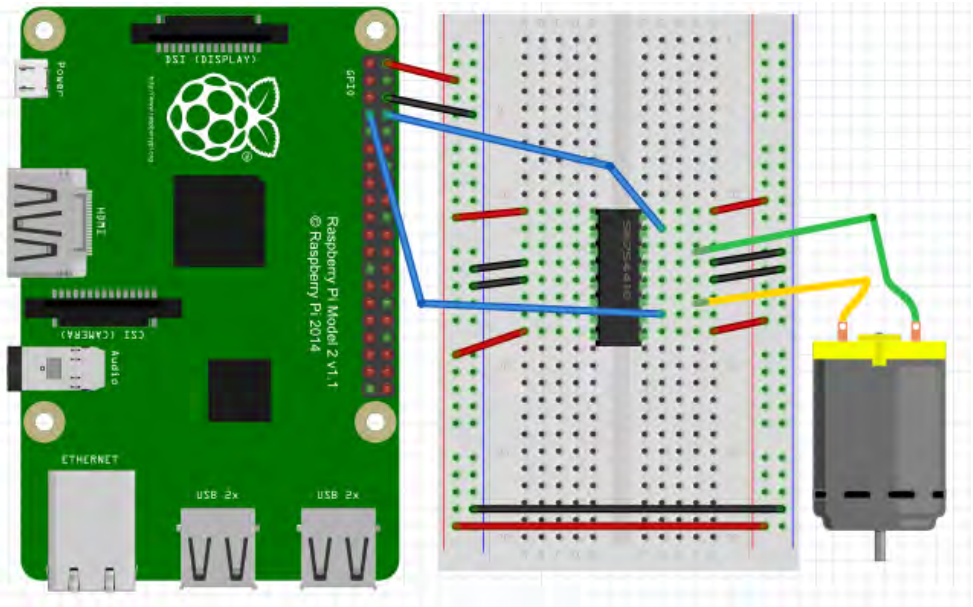
(2) Преку протоплочката, со сините краткоспојници ги поврзуваме GPIO пиновите број 4 и 14 со влезните пинови (анг. In1,In2) на интегрирано коло SN754410.

(3) Преку протоплочката, со зелената и жолтата краткоспојница ги поврзуваме приклучоците на моторот за еднонасочна струја со излезните пинови (анг. Out1, Out2) на интегрирано коло SN754410.

(4) На крај, горната и долната лента за напојување ја поврзуваме со пинот 5V на Raspberry Pi, лентата за напојување со GND пинот.



Слика 5.11. Функционална шема за поврзување на мотор на еднонасочна струја со Raspberry Pi 3B+



Слика 5.12. Монтажна шема за поврзување на мотор на еднонасочна струја со Raspberry Pi 3B+

6. Пишување и внесување на програма за Raspberry Pi 3B+

Ја пишуваме програмата во развојната средина Thonny Python IDE во состав на Raspbian оперативниот систем. Ја повикуваме класата инструкции Motor од библиотеката gpio. Ги користиме инструкциите motor.forward() и motor.backward() до кои веќе се запознавме.

```
1  from gpiozero import Motor    #  Го внесуваме делот од библиотеката gpio
                                 #  за работа со мотор.
2  from time import sleep
3  motor = Motor(forward=4, backward=14)
4  #4 и 14 се броеви на GPIO-пиновите.
5  while True:                   #  Започнува бесконечен циклус.
6     motor.forward()            #  Моторот врти во насока на стрелката #на
                                 #  часовникот. Бидејќи малата заграда е
                                 #  празна, тоа значи дека моторот ќе се врти
                                 #  со максимална брзина. Ако во малата
                                 #  заграда беше бројот 0,5 тоа ќе значеше
                                 #  дека моторот ќе врти со половина од
                                 #  својата максимална брзина.
7     sleep(5)                   #  Моторот врти 5 секунди во една насока.
8     motor.backward()           #  Се менува насоката на вртење.
9     sleep(5)                   #  Моторот врти пет секунди во насока
                                 #  обратна од стрелката на часовникот.
```

7. Резултати од реализацијата на вежбата

Резултатот од реализацијата на оваа практична вежба ќе го видиме ако притиснеме Run во менито со алатки на развојната средина Thonny Python IDE. Очекувани резултати се моторот да почне да врти со промена на својата насока на секои 5 секунди.

Коментар:_____

Користена литература:

[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/.