

Lyubiça Markudova

Mikrobilgisayarlar ve Onların Programlanması



Elektroteknik mesleđi
Bilgisayar tekniđi ve
otomasyon elektrik
teknisyeni

II.yıl



Lyubiça Markudova

Mikrobilgisayarlar ve Onların Programlanması

II.yıl

2024

MİKROBİLGİSAYARLAR VE ONLARIN PROGRAMLANMASI - II yıl
Sektör/meslek: ELEKTROTEKNİK/ELEKTROTEKNİK
Nitelikler/eğitim profilleri: Bilgisayar tekniği ve otomasyon elektrik teknisyeni

Yazar:

Lyubiça Markudova

İnceleyenler:

Daniyela Efnuşeva

Janeta Servini

Maryan Malinov

Bilgisayar yapımı:

Lyubiça Markudova

Kapak çizimi:

Kristina Başoska

Orijinal baskının başlığı:

МИКРОКОМПЈУТЕРИ И НИВНО ПРОГРАМИРАЊЕ -

II година

Sektor/struka: ELEKTROTEHNIKA / ELEKTROTEHNIČKA

Kvalifikacii/obrazovni profili: Elektrotehničar za kompjuterska tehnika i avtomatika

Љубица Маркудова

Makedonca'dan Türkçe'ye çeviri: Ervin Salih

Lektör: İbrahim Yusuf

Mesleki redaksiyon: Aybeyan Selim

Editör: İlker Ali

Grafik ve teknik düzenleme: Vladimir Mladenovski - ARS STUDIO

Yayıncı

Kuzey Makedonya Cumhuriyeti Eğitim ve Bilim Bakanlığı

sok. "Sv. Kiril ve Metodij" No. 54, 1000 Üsküp

Baskı: Evropa 92 – Koçani,

Tiraj:

Kuzey Makedonya Cumhuriyeti Eğitim ve Bilim Bakanlığı İlk ve Orta Öğretim Ders Kitapları Ulusal Komisyonu'nun 19.07.2023 tarih ve 26-1916/1 sayılı kararıyla bu ders kitabının kullanımı onaylanmıştır.

Önsöz

Bu ders kitabı, ortaöğretim meslek okullarında, elektro teknik mesleği, bilgisayar tekniği ve otomasyon elektrik teknisyeni eğitim profilinden ikinci yıl öğrencilere yöneliktir. Ders kitabı modüler olarak tasarlanmış öğretim programına göre yapılmıştır ve Mikrobilgisayarlar ve onların programlanması dersinin ikinci yıl öğretim içeriğine uygundur. Yeni öğretim programının amacı, mesleki eğitimde öğretimi ve öğretim programlarını modernleştirmek ve işgücü piyasasının ihtiyaçlarını tam olarak karşılamaktır.

İkinci yıl Mikrobilgisayarlar ve onların programlanması öğretim dersi haftada 2 ders veya yılda 72 derslik fonla gerçekleştirilmektedir. Toplam ders sayısının yarısı, yani 36 ders uygulamalı öğretim veya alıştırmalar olarak gerçekleştirilmelidir. Uygulamalı öğretime veya alıştırmalara yönelik dersler, Eğitim ve Bilim Bakanlığı Talimatına göre gruplar halinde gerçekleştirilir. Böyle bir kavram, edinilen teorik bilgilerin pratik olarak uygulanmasına olanak sunarak öğrenmeyi daha ilginç ve eğlenceli yapar. Mikrobilgisayarların ve onların programlanmasının gelişmesinin yarattığı çekicilik ve ilgi, yazarın bu ders kitabını yazma isteğine büyük ölçüde katkıda bulunmuştur. İçerikler, klasik bilişim ders kitaplarının içeriklerinden farklıdır, çünkü geliştirilen uygulamalar Arduino mikrodenetleyici platformu ve Raspberry Pi mikrobilgisayar gibi programlanabilir gömülü sistemlerde uygulanmaktadır. Alıştırmaların gerçekleştirilmesi için kişisel bilgisayarlarla birlikte bunlara ihtiyaç duyulacaktır.

Ders kitabı dört modüler birimi kapsamaktadır.

İlk modüler birimi olan Mikrobilgisayarların Kullanımı, bilgisayar uygulamalarına ilişkin büyük sayıda örnek vermektedir. Belirlenen hedefe bağlı olarak uygun bilgisayar sistemi ve uygulama geliştirme yazılımı seçilir.

İkinci modüler birimi olan Mikrobilgisayar Bileşenleri'nde öğrenciler, temel donanım bileşenlerinin özellikleri ve işlevlerine ilişkin bilgiler oluşturur ve genişletir. Kişisel bilgisayarın kurulum ve montajının yanı sıra Arduino tabanlı mikrodenetleyici platformunun elektronik bileşenlerine ve onların bağlanmasına önem verilmektedir.

Üçüncü modüler birimi olan İşletim sisteminin kurulumu'nda Windows 10, Ubuntu 17 Linux ve Raspbian işletim sistemlerinin kurulumunu tanıyacağız. Arduino tabanlı mikrodenetleyici platform geliştirme ortamını kurma ve onunla çalışma hakkında talimatlar da verilmiştir.

Dördüncü modüler birim olan Mikrobilgisayarların Programlanması en kapsamlı modüler birimdir, çünkü yalnızca C++ ve Python programlama dillerini incelemekle kalmayıp aynı zamanda bunları Arduino Uno geliştirme platformunu ve Raspberry Pi mikrobilgisayarını programlamak için kullanmaktadır.

Ders kitabında her konu, teorik bilgilerin uygulanmasına yönelik uygulamalı alıştırmalar ve öğrencilerin başarılarını kontrol etmek için sorular içermektedir.

Bu ders kitabının yazarı, öğretim malzemesini mümkün olduğu kadar basit ve anlaşılır bir şekilde, prosedür ve süreçlerin kesin açıklamalarıyla ve anlaşılır çizim ve resimlerle sunmaya çalışmıştır. Çizimlerin ve ilüstrasyonların yapımı için Microsoft Visio ve Fritzing olmak üzere iki program kullanılmıştır. Fritzing, Arduino Uno, Raspberry Pi ve devre tahtaları ile elektrik şemalar oluşturmak için çok iyi destek sunan açık kaynaklı uygulamadır.

Bu ders kitabını öğrencilerimize ithaf ediyorum ve onların mesleki hedeflerine ulaşmalarına yardımcı olacağını içten ümit ediyorum.

Bu ders kitabının hazırlanmasında özverili ve her yönlü desteklerinden dolayı aileme ve meslektaşlarıma teşekkür etmek istiyorum. Ayrıca, yararlı önerileriyle bu ders kitabının daha iyi yapılmasında katkıda bulunan inceleme komisyonuna da teşekkür etmek istiyorum.

Yazar'dan

İçindekiler

1. Mikrobilgisayarların Kullanımı	1
1.1. Mikrobilgisayar Terimi ve Türle	1
1.2. Mikrobilgisayarların Uygulanması	3
1.3. Arduino-Mikrodenetleyiciye Dayalı Geliştirme Platformu	5
1.4. Raspberry Pi Mikrobilgisayar Sistemi	8
Sonuçlar	11
Sorular ve ödevler	12
2. Mikrobilgisayarların Bileşenleri.....	13
2.1. Mikrobilgisayar Sisteminin Bileşenleri.....	13
2.1.1. İki işlemci.....	14
2.1.2. Bellek Organizasyonu.....	16
2.2. Anakart - Açıklama ve Bağlantıları	20
2.3. Arduino Uno R3'ün Bileşenleri	24
2.4. Arduino Uno R3'ün Elektronik Bileşenleri ve Onların Bağlanması	26
2.5. Arduino Platformu Eklentileri.....	31
2.6. Raspberry Pi 3B+'nın Bileşenleri	34
2.7. Raspberry Pi Eklentileri.....	36
2.8. Modüler Birimi için Uygulamalı Alıştırmalar: Mikrobilgisayarların Bileşenleri	38
2.8.1. İşyerinde Koruma ve Güvenlik Önlemleri	38
2.8.2. Kişisel Bilgisayar Bileşenlerin Kurulumuna İlişkin Uygulamalı Alıştırma.....	39
2.8.2.1. Kişisel Bilgisayarın Donanım Bileşenleri ile Çalışırken Koruma ve Güvenlik Önlemleri.....	39
2.8.2.2. İşlemcinin Kurulumu	39
2.8.2.3. Soğutucunun Kurulumu.....	40
2.8.2.4. RAM Belleğin Kurulumu	41
2.8.2.5. Anakartı Kasaya Monte Etme	42
2.8.2.6. Güç Kaynağı Cihazının Kurulumu	43
2.8.2.7. Sabit disk ve SSD Cihazının Kurulumu.....	44
2.8.2.8. M.2 SSD-Modüllerinin Kurulumu	44
2.8.2.9. DVD Sürücüsünün Kurulumu	45
2.8.2.10. Grafik Kartın Kurulumu.....	45
2.8.3. Elektronik Bileşenlerin Arduino Uno R3 ile Bağlanmasına İlişkin Uygulamalı Alıştırmalar.....	47
2.8.3.1. Arduino Tabanlı Mikrodenetleyisi Platformu ile Çalışırken Koruma ve Güvenlik Önlemleri	47
2.8.3.2. Deney Tahtası Kullanım Kılavuzu	47
2.8.3.3. Arduino Uno R3 İçin Bilgisayar Simülasyonu.....	49
2.8.3.4. Uygulamalı Alıştırma: Anahtarlar veya Arduino Uno R3 Kullanarak VE Mantık Devresinin Gerçekleştirilmesi	51

2.8.3.5. Uygulamalı Alıştırma: Anahtarlar veya Arduino Uno R3 Kullanarak VEYA Mantık Devresinin Gerçekleştirilmesi	56
Sonuçlar	60
Tekrarlama Soruları	62
3. İşletim Sisteminin Kurulumu	65
3.1. İşletim Sistem Türleri.....	65
3.2. Modüler Birim İçin Uygulamalı Alıştırmalar: İşletim Sisteminin Kurulumu.....	69
3.2.1. Yazılımın Kurulumuna Hazırlanma	69
3.2.2. Uygulamalı Alıştırma: Windows 10 İşletim Sisteminin Kurulumu.....	69
3.2.3. Uygulamalı Alıştırma: Windows 10 İşletim Sistemi Yapılandırılması	76
3.2.4. Uygulamalı Alıştırma: Ubuntu 17 Linux İşletim Sistemini Sanal makineye Yükleme	81
3.2.4.1. Sanal Makine Oluşturma Talimatları	81
3.2.4.2. Ubuntu İşletim Sisteminin Kurulumu	83
3.2.5. Uygulamalı Alıştırma: Raspberry Pi İçin Raspbian İşletim Sisteminin Kurulumu ve Yapılandırılması.....	86
3.2.5.1. Kurulum için NOOBS Yazılımının İndirilmesi.....	86
3.2.5.2. Raspbian İşletim Sisteminin Kurulumu.....	87
3.2.6. Uygulamalı Alıştırma: Arduino Platformu İçin Geliştirme Ortamının Kurulumu ve Kullanıma Koyulması.....	90
3.2.6.1. Arduino Platformu İçin Geliştirme Ortamının Kurulumuna İlişkin Talimatlar	90
3.2.6.2. Geliştirme Ortamının Menüsü ve Araçları ve Arduino Uno R3'te Programın Yazılması	92
Sonuçlar	95
Tekrarlama Soruları	98
4. Mikrobilgisayarların Programlanması	101
4.1. C/C++ Programlama Dili ve Uygulaması	101
4.2. Arduino Platformu İçin C/C++ Programlama Dilindeki Değişkenler ve Operatörler	103
4.3. Arduino platformu İçin C/C++ Programlama Dilindeki Komutlar	107
4.3.1. Giriş-Çıkış Pinleriyle Çalışma Komutları	107
4.3.2. Seri İletişim Komutları.....	111
4.3.3. Zaman Kontrol Komutları.....	113
4.3.4. Matematik Komutları.....	113
4.3.5. Bit ve Bayt Komutları	115
4.3.6. C/C++ Programlama Dilindeki Yapılar	116
4.3.6.1. Olanak Seçim Yapısı	116
4.3.6.2. Tekrarlama Yapısı (Döngüsel Yapılar).....	119
4.3.6.3. Dallanma Yapıları	122
4.3.7. Kütüphanelerle Çalışma Komutları.....	123
4.4. Raspberry Pi 3B+'ı Python Programlama Dili ile Programlama	129
4.5. GPIO Zero Kütüphanesi.....	131
4.6. GPIO Zero Kütüphanesinden Giriş Cihazlar Sınıfı.....	133
4.6.1. Düğme (İng.Button)	134
4.6.2. Kızılötesi Yansıtıcı İzleme Modülü (TRCT5000).....	136
4.6.3. Mesafe Sensörü (HC-SR04).....	137
4.6.4. Optik Sensör veya Foto Rezistör.....	138

4.7. GPIO Zero kütüphanesinden Çıkış Cihazlar Sınıfı.....	139
4.7.1. LED-diyotu.....	139
4.7.2. Genişlik Modülasyonlu Darbelerle Kontrol Edilen LED-diyot (PWM LED)	140
4.7.3. Değişken Renkli LED-diyot (RGBLED).....	141
4.7.4. Doğru Akım Motoru.....	142
4.7.5. Servo Motor	143
4.8. Modüler Birimi İçin Uygulamalı Alıştırma: Mikrobilgisayarın Programlanması.....	145
4.8.1. Arduino Uno R3'ü C/C++ Programlama Dilinde Programlamak İçin Uygulamalı Alıştırmalar.....	145
4.8.1.1. Uygulamalı Alıştırma: LED diyotun Düğmeyle Açılması.....	145
4.8.1.2. Uygulamalı Alıştırma: LED diyotun Potansiyometre ile Kontrol Edilmesi.....	149
4.8.1.3. Uygulamalı Alıştırma: LED Diyotu ile Işık Yoğunluğunun Düzenlenmesi	152
4.8.1.4. Uygulamalı Alıştırma: Işık Yoğunluğunun Kontrolü İçin Fotorezistör..	154
4.8.1.5. Uygulamalı Alıştırma: Arduino Uno R3'ü Servo Motora Bağlama	159
4.8.1.6. Uygulamalı Alıştırma: Arduino Uno R3'ü LCD Ekranına Bağlama	162
4.8.2. Raspberry Pi 3B+'ı Python Programlama Dilinde Programlamak İçin Uygulamalı Alıştırmalar.....	165
4.8.2.1. Raspberry Pi ile Çalışırken Koruma ve Güvenlik Önlemleri.....	166
4.8.2.2. Uygulamalı Alıştırma: LED diyotun Düğmeyle Açılması.....	167
4.8.2.3. Uygulamalı Alıştırma: Trafik Işığı.....	169
4.8.2.4. Uygulamalı Alıştırma: Tepki Süresi.....	172
4.8.2.5. Uygulamalı Alıştırma: LED diyotun Fotorezistör ile Açılması ve Kapatılması	175
4.8.2.6. Uygulamalı Alıştırma: En Yakın Nesneye Olan Mesafeyi Kontrol Etme	177
4.8.2.7. Uygulamalı Alıştırma: Doğru Akım Motorunun Yönünü Değiştirme....	180
Sonuçlar	182
Tekrarlama Soruları	184
Ek: Raspberry Simülatörleri.....	190
Kaynakça.....	194



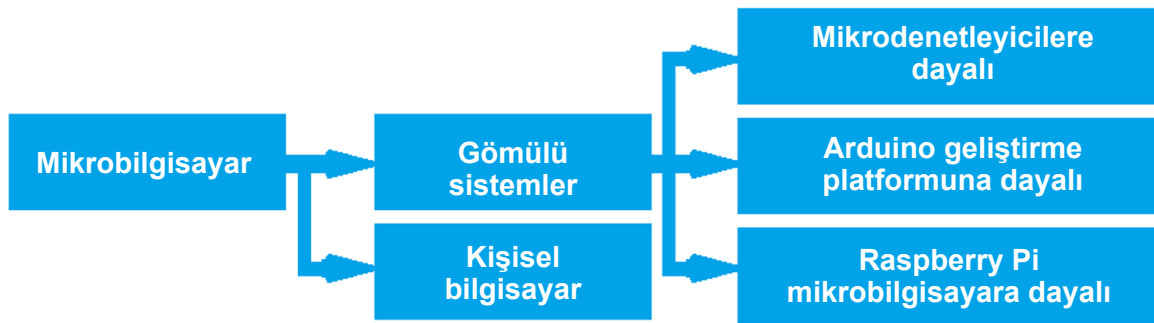
1. Mikrobilgisayarların Kullanımı

1.1. Mikrobilgisayar Terimi ve Türleri

Mikrobilgisayarlar her yerde kullanılmaktadır ve onlarsız modern yaşam düşünülemez. Bileşiminde mikrobilgisayar içermeyen neredeyse hiçbir cihaz yoktur. Bunlar: ev aletleri, multimedya cihazları ve eğlence cihazları, otomatlar, iletişim cihazları, tıbbi cihazlar, endüstriyel tesisler, trafik araçlar, ofis cihazları vb.

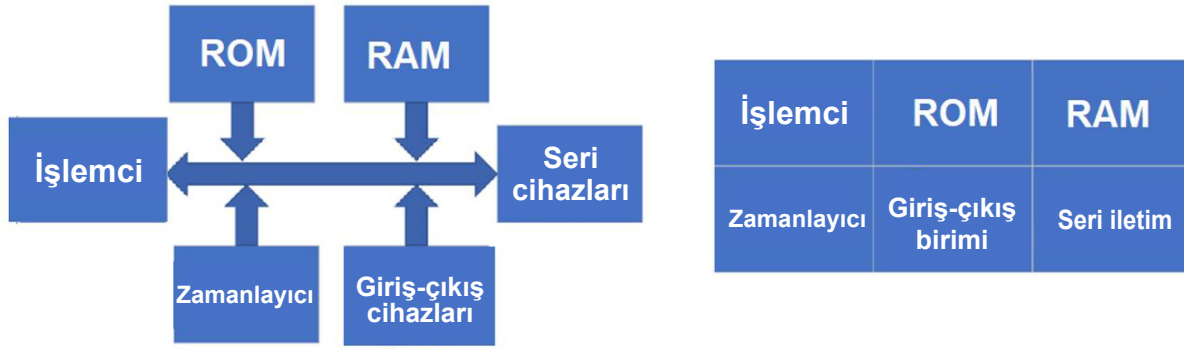
Mikrobilgisayar olarak, merkezi işlem birimi, çalışma RAM-belleği ve programları ve işlem ile elde edilen sonuçlarını kaydetmek için kalıcı bellekten oluşan sistemi tanımlıyoruz. Bu bileşenler gereklidir ve bunlar olmadan veriler işlenemez. Ardından insanın mikrobilgisayarlara bilgi vermesini, mikrobilgisayarın da insana bilgi vermesini sağlayan giriş-çıkış cihazları eklenir.

Teknolojinin hızlı gelişimi, mikrobilgisayar teriminin anlamını değiştirdi veya daha doğrusu genişletti. Eskiden bu terim kişisel bilgisayar terimine bağlıymış. Ancak bu günümüzde, öncelikle gömülü mikrobilgisayar sistemlerinin ortaya çıkması ve geliştirilmesi nedeniyle önemli ölçüde değişti. Şekil 1.1'de çağdaş mikrobilgisayarların bölümü gösterilmiştir. Mikrodenetleyiciler ve geliştirme donanım platformları gömülü sistemlerin (İng. Embedded System) temelidir. [1].



Şekil 1.1. Mikrobilgisayarların bölümü

İşlemci, kalıcı bellekte bulunan programları çalıştırır, verilen anda işlenmekte olan veriler ise RAM belleğinde saklanır. Bunların hepsi, işlemci, kalıcı ROM ve geçici RAM belleği entegre devreler tanımlamaktadır. Şu soru ortaya çıkmış: Tek bir entegre devre olabilecekken neden ayrı entegre devreler olsunlar? Bu sayede aslında bir mikrobilgisayarı entegre devre şeklinde temsil eden **mikrodenetleyici** elde edilmiştir. İçeriğinde mikroişlemci, bellekler ve veri iletimi için iletkenler içerir. Şekil 1.2. kişisel bilgisayar ile mikrodenetleyici arasındaki karşılaştırmayı temsil etmektedir. Kişisel bilgisayarlarda, verilen donanım bileşenlerinin her biri anakart üzerinde ayrı bir çiptir ve bu yüzden her biri ayrı bir blokla temsil edilir, blokları birbirine bağlayan oklar ise, anakart üzerinde veri aktarmaya yarayan hatları temsil eder. Mikrodenetleyici hepsi bir arada olacak şekilde tasarlanmıştır, bu nedenle tüm donanım bileşenleri tek bir ortak blokla temsil edilir. Mikrodenetleyici, işlemci, kalıcı ve çalışma belleği dışında, zamanı ölçmek için zamanlayıcılar, analog-dijital dönüştürücü ve seri iletişim birimi içerebilir.



Şekil 1.2. Kişisel bilgisayar şeması ile mikrodenetleyici şeması arasındaki karşılaştırma

Küçük boyutları ve düşük maliyeti nedeniyle mikrodenetleyici birçok elektronik cihaza dahil olabilir. Kişisel bilgisayarlardan farklı olarak, mikrodenetleyiciler çok daha düşük çalışma frekansına, daha küçük bellek kapasitesine sahiptir ve ağda bağlanmaları daha zordur. Kişisel bilgisayarları klavye, fare ve monitör olmadan kullanamadığımız gibi, mikrodenetleyiciler de sensörler ve yürütme cihazları bağlamadan kendi başlarına çalışamazlar. Mikrodenetleyiciler, giriş pinlerine bağlı sensörler aracılığıyla çevrelerindeki ortam hakkında bilgi alırlar. Ardından bu veriler mikroişlemcide işlenir ve programın çalıştırılmasının ardından mikrodenetleyicinin çıkış pinlerine bağlı yürütme birimleri etkinleştirilir (elektrik motorları, pompalar, ekranlar vb.). Bir elektronik cihaza yerleştirilmeden önce mikrodenetleyicinin programlanması gerekir. Bir kullanıcı programın yazılması, **entegre geliştirme ortamı** (İng. IDE-Integrated Development Environment) olarak da bilinen geliştirme programı gerekir. Kullanıcı programının mikrodenetleyicinin hafızasında uygulanabilmesi için

bilgisayara genellikle USB kablosuyla bağlanan, programlayıcı olarak adlandırılan özel bir cihaza ihtiyaç vardır. Mikrodenetleyici programlayıcıdaki tabanına yerleştirilir, kullanıcı programı kalıcı belleğe yazılır ve daha sonra mikrodenetleyici kullanıcı cihazının tabanına aktarılır.

Mikrodenetleyiciye dayalı gömülü sistemlerin yanı sıra Arduino ve Raspberry Pi'ye dayalı sistemlerin kullanımı giderek artıyor. Arduino, kredi kartı boyutunda, mikrodenetleyiciye dayalı geliştirme platformudur. Mikrodenetleyici **geliştirme platformları**, mikrodenetleyici dışında seri programlayıcı, kristal osilatör, analog ve dijital giriş-çıkışlar içeren elektronik kartları temsil eder. Programcının programlayıcı seçimi konusunda düşünmesine gerek yoktur ve yerleşik giriş-çıkış pinleri sensörlerin ve yürütme cihazların basit şekilde bağlanmasını sağlar. Donanım geliştirme platformları basit programlamayla karakterize edilir. Örneğin bunların programlanması için bit komutları kullanılır. Bit test yönergeleri ile, sensörlerin bağlı olduğu giriş pinleri kontrol edilir, bit ayarlama ve bit sınırlama yönergeleri ise çıkış cihazlarını açar ve kapatır. Yönetmemiz gereken elektronik cihazı yapmak için baskılı devre kartı yerine, cihazın tasarımını denememize olanak sağlayan deneysel devre tahtası (İng. Protoboard) kullanılabilir.

Arduino ile yakın aynı boyutta olmasına rağmen Raspberry Pi çok daha güçlü işlemciye ve daha büyük RAM belleği kapasitesine sahiptir. Mikrodenetleyici yerine işlemci ve yerleşik grafik kartı içeren özel bir çip içerir ve RAM belleği elektronik kart üzerinde bulunan özel bir çiptir. Arduino'dan farklı olarak kendi işletim sistemine sahiptir ve bu nedenle Raspberry Pi, Tek levhalı bilgisayar (İng. SBC-Single Board Computer) adı verilen mikrobilgisayar sistemidir.

Biz, Arduino platformunu ve Raspberry Pi'yi inceleyeceğiz ve bileşenlerini, işlevlerini, giriş-çıkış cihazlarıyla bağlantılarını ve onların programlama yöntemlerini tanıyacağız.

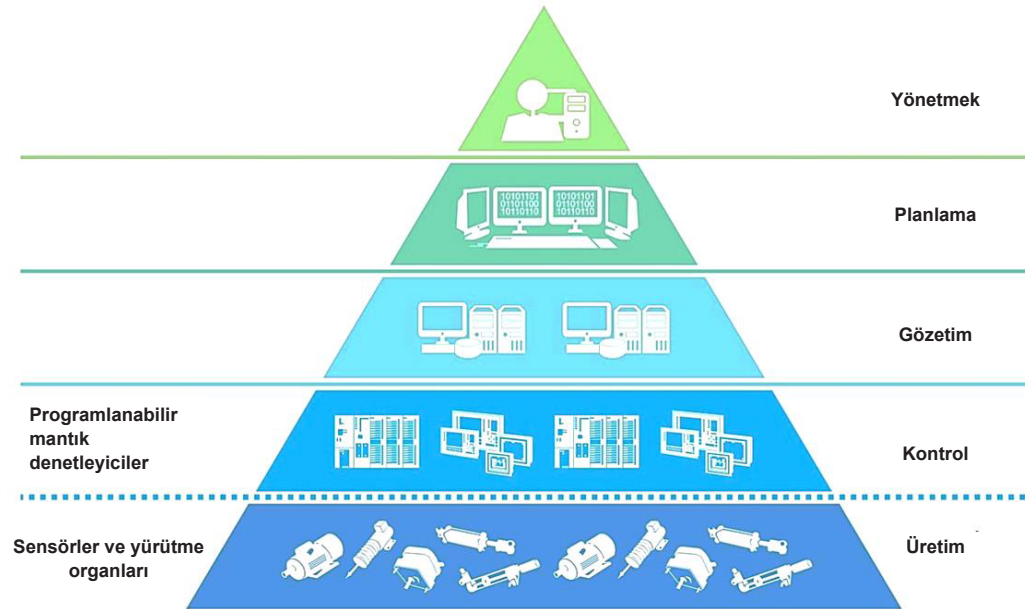
1.2. Mikrobilgisayarların Uygulanması

Mikrobilgisayar sistemi terimini tanıdıktan sonra, şimdi kullanımlarını ve uygulamalarını daha detaylı tanıyalım. Mikrobilgisayar uygulamalarının sonsuz çok örnekleri vardır. Bunlardan birkaçını belirteceğiz.

- **Tıp.** Başlangıçta bilgisayarlar, daha iyi görünürlük ve kullanılabilirlik nedeniyle tıbbi verileri depolamak ve işlemek için kullanılıyormuş. Bu kategori hastalara uzaktan bakım sağlayan teletıp hizmetini de içermektedir. Günümüzde bilgisayarlı tomografi, manyetik rezonans ve 3 boyutlu röntgen kullanılmadan hastalıkların

teşhisini ve zamanında tespitini düşünmek imkansızdır. Cerrahlar, cerrahi prosedürleri simüle etmek için sanal modelleri giderek daha fazla kullanıyor. Çiplerin insan vücuduna yerleştirilmesi artık basit bir işlemdir. Örneğin kalp pili, kesin olarak tanımlanmış genlik ve frekansta elektriksel uyarılar üreten ve kalbin çalışmasını uyaran bir çiptir. Bu çipin kendi güç kaynağı var ve aynı zamanda kalbin çalışmasını izlemek için de kullanılabilir.

- **Otomobil endüstrisi.** Modern bir otomobilde 70'in üzerinde kontrol birimi ve 100'ün üzerinde sensör bulunmaktadır. Yeniliklerin %80'i mikrodenetleyicilerin uygulanmasına dayanmaktadır. En önemli mikrodenetleyici elektronik kontrol birimidir (ECU – Electronic Control Unit). En önemli fonksiyonlar yakıt miktarı kontrolü, yakma zamanlaması ve egzoz gazı kontrolüdür. Ana giriş değerleri motor gücü ve devir sayısıdır. Emilen havanın sıcaklığı ve akışı, soğutucu sıvının sıcaklığı, şaft konumu gibi başka giriş verileri de vardır. Çıkış büyüklükler, bir çevrimde enjekte edilen yakıt miktarı ve yanma anıdır. Bu veriler, çıkış katsayılarının elde edildiği bir matrise kaydedilir. Matris aslında bir EEPROM belleğidir ve farklı araba türleri için farklıdır. Hesaplamanın ardından mikrodenetleyici elde edilen değerleri aktüatöre - enjektörlere iletir. Sıcaklık öngörülen sıcaklıktan yüksekse enjektörün açık kalma süresi uzar (daha yüksek hava akışı).
- **Endüstrinin otomasyonu.** Otomatik yönetim karmaşık bir süreçtir ve birkaç seviyeye ayrılabilir.



Şekil 1.3. Mikrobilgisayarların otomasyonda uygulanması

Birinci seviye, üretim parametrelerini izleyen ve deęiřtiren sensörler ve yürütme birimleridir. İkinci seviye, analiz yapmadan programa göre anlık hareket eden programlanabilir denetleyicilerdir. Üçüncü olan gözetim seviyesi, verileri işler, matematiksel modeller oluşturur, makineleri inceler ve optimizasyon gerçekleştirir, raporlar hazırlar. Verileri hızlı bir şekilde işleyecek hızlı bilgisayara, geçmiş deęerleri saklayacak geniş kalıcı belleęe ve parametrelerin daha kolay izlenebilmesi için kullanımı kolay(user friendly) grafięe ihtiyaç vardır.

- **Bankacılık ve Finans.** Birçok finansal kurum dijital teknolojilere yatırım yapıyor. Bankacılık işlemleri, hangi ülkede ikamet ettięinize baęlı olmadan mobil ve web platformları üzerinden gerçekleştirilmektedir. Bilgisayarlar istatistik raporlar üretmek, dünya eğilimlerini izlemek ve elektronik bankacılıęı yeniden modernleřtirmek için kullanılıyor. Bu tür eğilimler řunlardır: ödeme kartı yerine cep telefonuyla işlem yapmak ve daha fazla güvenlik ve suistimale karşı koruma için biyometrik yetkilendirmenin kullanılması (parmak izi veya bilgisayar yüz tanıma kullanılması).

Her alanın kendine has özellikleri vardır ve o alanda bilgisayar uzmanı olabilmek için uzmanlaşmak gerekir. Bireyin istek ve yeteneklerine göre daha ileri mesleki gelişim için çok önemli olan mikrobilgisayar donanım ve yazılımının temellerini tanıyacaęız.

1.3. Arduino-Mikrodenetleyiciye Dayalı Geliřtirme Platformu

Arduino, ilk büyük açık kaynaklı program kodlu (open source) **mikrodenetleyici platformudur**. Bu, internetten ücretsiz olarak indirilebilen ve elektronik cihazın performansını artırmak amacıyla deęiřtirilebilen birçok hazır kullanıcı programına ve dosyasına serbest erişim anlamına gelir. Arduino, elektronik prototip yenilikler oluřturma sürecini basitleřtirmek amacıyla 2005 yılında piyasaya sürüldü. Bu, çok az teknik bilgisi olan veya hiç teknik bilgisi olmayan sıradan insanların etkileşimli ürünler oluřturmasına olanak sağladı.

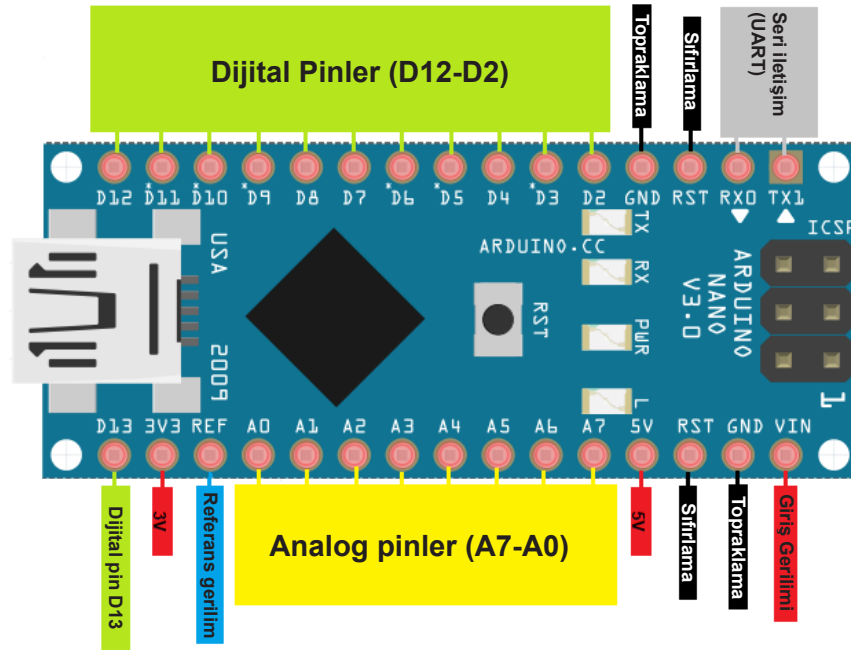
Daha sonra Arduino Uno R3 platformunun donanım bileşenlerini detaylı olarak tanıyacaęız. Mimarisini, giriş-çıkıř birimlerini, baęlanma yöntemini, geliřtirme ortamındaki araçları ve program belleęinde uygulayacaęımız birkaç hazır programı inceleyeceęiz.

Arduino platformunun girişine şunlar bağlanabilir: düğmeler, anahtarlar, hazır klavyeler, sensörler (sıcaklık, basınç, akış, hareket vb. için). Çıktılar, LED diyotlar, lambalar, sesli uyarıcılar, motorlar, ekranlar gibi çeşitli çıkış cihazlarına bağlanabilir. Örneğin, LED diyotun yanıp yanmayacağı veya motorun çalışmaya başlaması, düğmeye basılıp basılmadığına veya sensörün bazı fiziksel büyüklüğün değişip değişmediğini tespit edip etmediğine bağlıdır. Ek kalkanlar (İng.Shields), temel Arduino'ya monte edilip temel olanaklarını genişleten ve motor kontrolü, sensör bağlantısı, kablosuz iletişim gibi ek işlevleri gerçekleştirebilen elektronik kartlardır.

Arduino'nun programlanması için C ve C++ dillerinin bir modifikasyonu olan C/C++ programlama dili kullanılır. Arduino ve kişisel bilgisayar komut setleri arasında fark vardır çünkü Arduino giriş-çıkış cihazlarını yönetir. Ayrıca programlama donanım hakkında, özellikle de sensörlere ve yürütme birimlerine nasıl bağlanacağı konusunda geniş bilgiye sahip olmayı gerektirir.

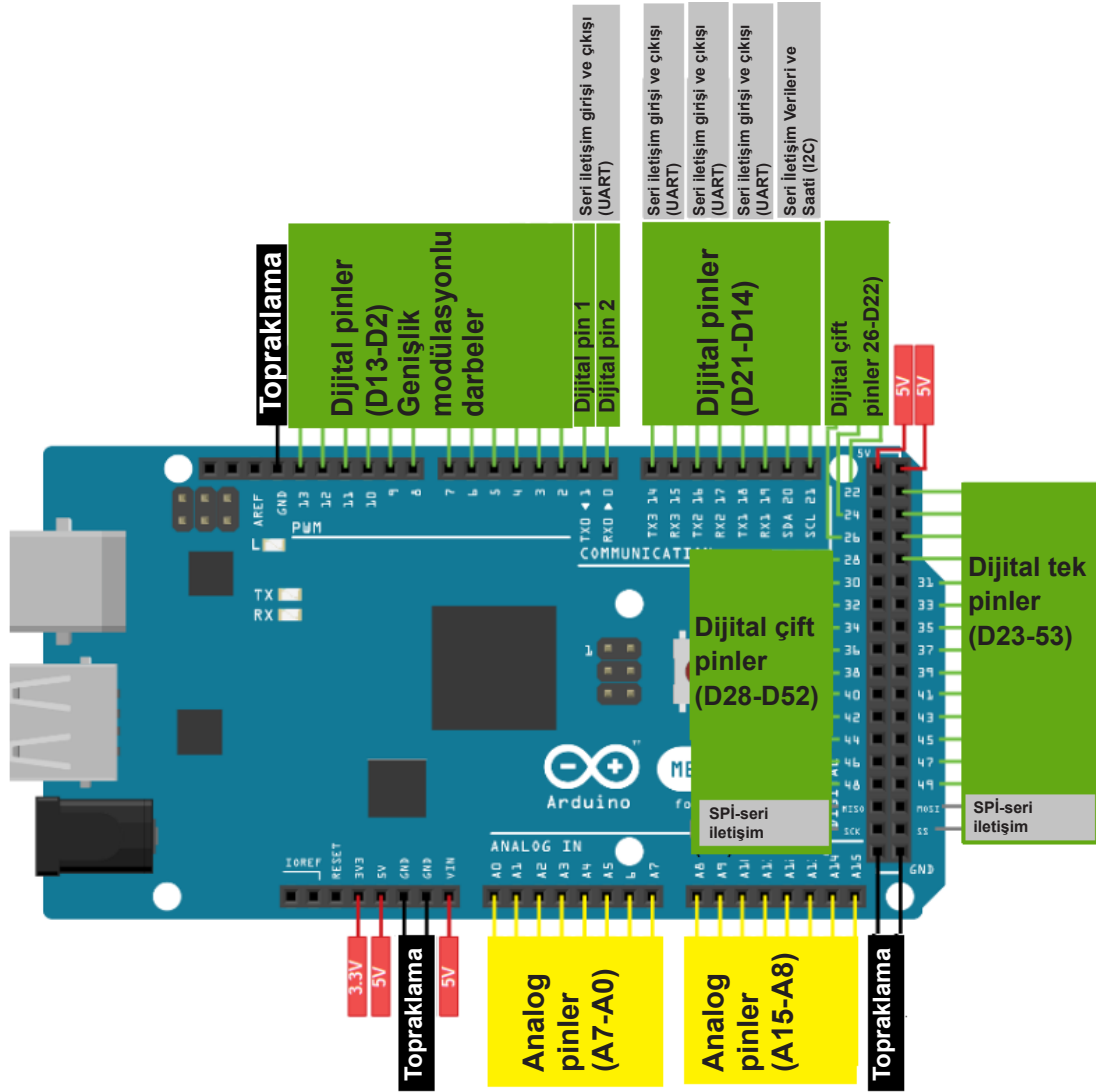
15'ten fazla farklı Arduino platform modeli bulunmaktadır ve şu anda en popüler olanlar Arduino Uno R3, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due'dir. Aralarındaki farkları kısaca açıklayacağız. [2]

- **Arduino Uno'nun** üçüncü versiyonu (Rev3 veya R3) yeni başlayanlar için en iyi platformdur ve bunu uygulamalı alıştırmaların gerçekleştirilmesinde açıklayacağız ve kullanacağız. Bu platformda sekiz bitlik ATmega 328 mikrodenetleyici, 14 dijital giriş-çıkış pini (bunlardan altısı genişlik modülasyonlu darbelerin çıkışı olarak kullanılabilir), 8 analog giriş, bir USB konektörü, güç kaynağı girişi bulunmaktadır. Platform pille veya AC/DC dönüştürücüyle çalıştırılabilir. Diğer tüm modelleri Arduino Uno R3 platformuyla karşılaştıracacağız.



Şekil 1.4. Arduino Nano platformunun pin diyagramı

- **Arduino Nano.** Arduino Uno R3 ve Arduino Nano aynı işlemciye, bellek kapasitesine ve pinlere sahiptir. Ancak üç temel fark vardır. Arduino Nano'nun iki kat daha küçük boyutta olması montaj sırasında avantaj sağlar. İkincisi, Arduino Nano'nun her iki tarafında da dişi pinler bulunur ve devre tahtasına kolayca yerleştirilebilir. Arduino Nano, mini USB'ye sahip olduğundan USB Type-C bağlantı noktasına sahip Arduino Uno R3'ün aksine çok sayıda dış cihaza bağlanamaz.
- **Arduino Pro Mini,** Arduino Uno R3'ün altıda biri boyutundadır. Arduino Uno R3'ün 5 V'una karşılık gelen gerilim kaynağı daha düşüktür ve 3,3 V'tur. İki kat daha düşük çalışma frekansına sahiptir ve seri iletişimin olanağı yoktur. Ancak düşük enerji tüketimi büyük bir avantajdır, bu da onu ulaşılması zor yerlere yerleştirmek için uygundur.
- **Arduino Mega 2560,** Arduino Uno R3'ün büyük kardeşi gibidir. Arduino Mega 2560'ın pin diyagramı şekil 1.5'te gösterilmiştir.



Şekil 1.5. Arduino Mega 2560'ın pin diyagramı

Arduino Mega 2560, 15'i darbe genişliği modülasyonlu çıkış olarak kullanılabilen 54 dijital giriş-çıkış pinine sahiptir. 14 analog girişe sahiptir. Veri almak ve göndermek için dört adet yerleşik üniversal asenkron port (UART-Universal Asynchronous Receiver-Transmitter) sayesinde aynı anda dört cihazla seri iletişime olanak sağlar. 256 KB program belleği, 8 KB RAM belleği ve 4 KB veri belleği vardır. Arduino Mega'nın çalışma belleğinin Arduino Uno R3'e göre sekiz kat daha büyük olduğunu görüyoruz.

- **Arduino Leonardo.** Temel fark işlemcilerde ve biraz farklı pin sayısındadır. Arduino Leonardo'nun işlemcisi arıza durumunda değiştirilemez ancak diğer cihazlara bağlanmak için USB desteği vardır, Arduino Uno R3'ün ise bunu için ayrı bir denetleyiciye ihtiyacı vardır. Ancak öte yandan Arduino Leonardo, birçok Arduino Uno R3 eklentileriyle uyumsuzdur.
- **Arduino Due,** Arduino Uno R3'ten çok daha güçlü işlemciye sahiptir, ancak fiyatı neredeyse iki katıdır. Örneğin Arduino Due'nin çalışma frekansı 84 MHz iken Arduino Uno R3'ün 16 MHz'idir. Arduino Uno R3 sekiz bitli, Arduino Due ise 32 bitlidir. Saniyede 104 milyon yönerge yürütebilir. Bu nedenle Arduino Due çoğunlukla kayan noktalı sayılar olarak da bilinen (İng. floating point number) ondalık sayılarla hesaplama yapmak için kullanılır.

1.4. Raspberry Pi Mikrobilgisayar Sistemi

Raspberry Pi Arduino'nun en büyük rakibidir. Raspberry Pi'nin ilk nesli piyasaya 2012 yılında İngiltere'de sunuldu. Popülaritesinin kanıtı, ilk altı ayda yaklaşık yarım milyon kopya satılmış olmasıdır. Arduino'ya benzer olarak Raspberry Pi, programlamayı kolaylaştırmak amacıyla Cambridge Üniversitesi'ndeki bilgisayar bilimi öğrencilerine yönelikti. Bugün Raspberry Pi'nin büyük bilimsel, eğitici ve pratik önemi vardır.

Rakip olmalarına rağmen Raspberry Pi, Arduino'dan çok farklıdır. Raspberry Pi, özelliklerine göre kişisel bilgisayara çok benzer ve monitör, klavye ve fare ile bağlanabilir. Raspberry Pi işlevsel **işletim sistemine** (çoğunlukla Linux) sahiptir, bu nedenle karmaşık programları çalıştırabilir. Dezavantajı, Raspberry Pi yazılımının Arduino'da olduğu gibi açık türden olmamasıdır. Raspberry Pi geliştirme ortamı ücretsiz hazır programlar ve sürücü kütüphaneleri içermez, hepsi için ayrı lisanslar gerekir. Arduino'nun, sensörlere ve yürütme cihazlarına bağlanması çok kolaydır, Raspberry Pi'nin bir sensöre etkili bir şekilde bağlanması için ise ek yazılıma ihtiyaç vardır. Öte yandan tüm Raspberry Pi modelleri video cihazlarına bağlanmayı oldukça kolaylaştıran **grafik kartına** sahiptir. Tüm modellerde en önemli bileşen, ARM

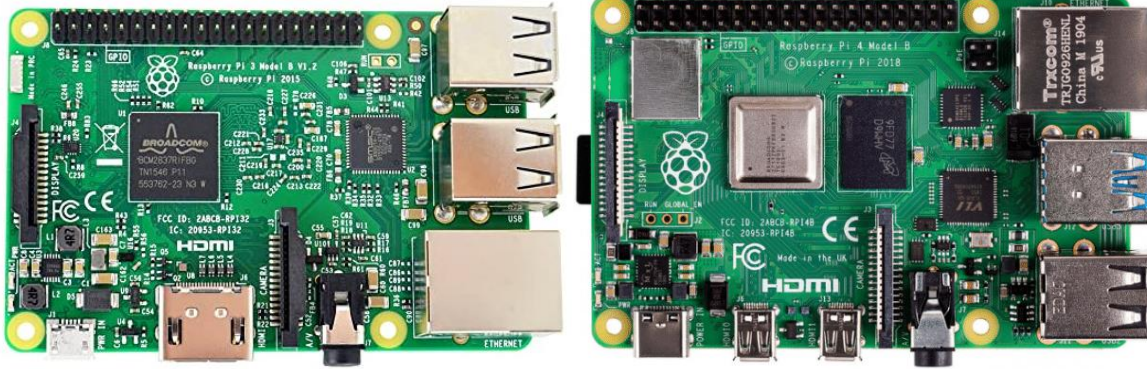
işlemcili ve entegre grafik kartına sahip Broadcom SoC (İng. System on a Chip) yongasıdır (çipidir). İlginç olan, Raspberry Pi'nin iç kalıcı belleği olmamasıdır. Genellikle işletim sistemini depolamak ve program belleği olarak dış SD-kart kullanılır. 40 pinli konektör (GPIO-General Purpose Input Output) üzerinden 27'ye kadar farklı cihaz bağlanabilir. Raspberry Pi, internete erişimi daha da kolaylaştıran **ağ konektörlerine** ve Wi-Fi aracılığıyla bağlanma olanağına sahiptir. Bazı modeller, kablosuz iletişim için başka bir yolu olarak entegre Bluetooth da içerir. Raspberry Pi hemen hemen her programlama dilinde programlanabilir. En yaygın olarak kullanılan programlama dili Python'dur ancak C/C++, Java Script, Scratch gibi diğer programlama dilleri de kullanılabilir. [3]

	Raspberry Pi 4 Model B	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Raspberry Pi Model B
İşlemci	1.5 GHz'li QUAD Core	1.2 GHz'li QUAD Core	900 MHz'li QUAD Core	700 MHz'li tek çekirdek
Grafik kart	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Kalıcı bellek	MicroSD	MicroSD	MicroSD	MicroSD
RAM	400 MHz'li 4 GB'a kadar SDRAM	1 GB SDRAM @ 400 MHz	400 MHz'li 1 GB SDRAM	400 MHz'li 512 MB SDRAM
Çalışma frekansı	1,5 GHz	1,4 GHz	900 MHz	700 MHz
Chipset	Broadcom BCM2838 64Bit	Broadcom BCM2837 64Bit	Broadcom BCM2836 32Bit	Broadcom BCM2837 32Bit
USB 2.0	4 x USB bağlantı noktası	4 x USB bağlantı noktası	4 x USB bağlantı noktası	4 x USB bağlantı noktası
Ağ bağlantısı	RJ45	RJ45	RJ45	RJ45
Wi-fi	Dahili	Dahili	Yoktur	Yoktur
Bluetooth	Dahili	Dahili	Yoktur	Yoktur
Genel amaçlı girişler ve çıkışlar	40-pinli konektör	40-pinli konektör	40-pinli konektör	40-pinli konektör
Güç kaynağı	5 V	5 V	5 V	5 V

Tablo 1.1. Farklı Raspberry Pi modellerinin özelliklerinin karşılaştırılması

Şimdiye kadar **dört nesil** Raspberry Pi üretilmiştir ve her neslin birkaç modeli vardır. 2012'den ilk nesil A ve B olmak üzere iki modele sahiptir. Model B daha iyi konfigürasyona sahiptir ve model A daha az belleğe ve daha az bağlantı noktasına sahiptir, ancak aynı zamanda daha düşük güç tüketimine sahiptir. Raspberry Pi 2, model B 2015 yılında piyasaya sunuldu ve üçüncü nesil bir sonraki yıl ortaya çıktı. 2019 yılında son nesil Raspberry Pi 4 piyasaya çıktı. Tablo 1.1'de her neslin özellik-

leri verilmiştir. Tablodan Raspberry Pi'nin olanakları açısından Arduino'dan kişisel bilgisayara çok daha yakın olduğu sonucuna varılabilir.



Raspberry Pi 3 Model B

Raspberry Pi 4 Model B

Şekil 1.6. İki Raspberry Pi modelinin dış görünümü

	Raspberry Pi	Arduino
Avantajlar	1,5 GHz çalışma frekansına 64 bit işlemciye sahip ve aynı anda birden fazla görevi gerçekleştirebilme olanağı	Analog sensörlere kolay bağlanma ve motor çalışmasının hassas kontrolü
	Dahili ağ bağlantısı, Wi-Fi veya Bluetooth üzerinden kablosuz bağlantı olanağı	İşlevselliği artırmak için geniş ek seçimi
	İşlevsel işletim sistemi	Programları, ek ayarlamalara gerek kalmadan, gücü açtıktan hemen sonra çalıştırır.
	Ses çıkışı, kamera bağlantısı, HDMI çıkışı ve bir kaç USB bağlantısı	Düşük maliyet fiyatı
	Robotlar veya video oyunları gibi projeler için çok iyi bir seçim	Basit uygulamaları gerçek zamanda çalıştırır
	Birden fazla programlama dilinde programlanabilir	Basitleştirilmiş donanım ve yazılım
Dezavantajlar	Elektronik bileşenlerin bağlanması için ek yazılım gerekir	16MHz çalışma frekansı, düşük çalışma hızı ve sadece bir program çalıştırır
	Yüksek maliyet fiyatı	Ethernet eklentisi olmadan İnternet'e bağlanamaz
	Kullanım sırasında daha fazla ısınma	Programlama için sadece C/C++ programlama dili kullanılır

Tablo 1.2. Arduino ve Raspberry Pi Karşılaştırması

Biz, Raspbian işletim sisteminin nasıl kurulacağını, Python programlama dilinin temel komutlarını ve fonksiyonlarını tanıyacağız ve Raspberry Pi'nin üstün arayüz performanslarını kullanan birkaç proje gerçekleştireceğiz.

Verilen bir uygulamalı alıştırmaların veya proje görevinin gerçekleştirilmesi için geliştirme platformunu akıllıca seçmek çok önemlidir. Bu konuda, Arduino ve Raspberry Pi'nin avantajlarını ve dezavantajlarını karşılaştıran Tablo 1.2 bize yardımcı olabilir. Raspberry Pi yazılıma yöneliktir yani ödevleri öncelikle programları uygulayarak yerine getirir, Arduino ise donanıma yöneliktir. Örneğin birden fazla fotoğraf çekmek bunları işleyip bir web sitesinde yayınlamak istiyorsak Raspberry Pi daha basit çözümler sunuyor. Ancak doğru akımlı motorun dönüş hızını kontrol etmek istiyorsak, o zaman daha yüksek reaksiyon hızı ve daha fazla hassasiyet nedeniyle Arduino Uno platformu daha uygundur.

Arduino ve Raspberry Pi, Nesnelerin İnterneti (İng. IoT-İnternet of Things) platformunun bir parçasıdır. Bu platform, teknolojiye yeni bir eğilimdir ve farklı türden iletişim protokolleri ve veri analizi uygulayarak sensörlerin ve cihazların bağlantısı, bunların ağ oluşturması ile ilgilidir. Bu platform analizimizin konusu değil ancak Arduino ve Raspberry Pi ile bağlantısı kaçınılmazdır.

Sonuçlar

Bilgisayar sistemi ile, merkezi bir işlem birimi, çalışma RAM belleği ve programların ve elde edilen işlem sonuçlarının kaydedilmesi için kalıcı bellekten oluşan bir sistem tanımlanır.

Modern bölüme göre mikrobilgisayarlar, kişisel bilgisayarların yanı sıra gömülü mikrobilgisayar sistemlerini de içerir. Bunlar mikrodenetleyicilere, donanım geliştirme platformlarına veya Raspberry Pi'ye dayalı olabilirler.

Gömülü mikrobilgisayar sistemleri düşük maliyetleri, esneklikleri ve basit programlanmaları nedeniyle oldukça popülerdir.

Mikrodenetleyici geliştirme platformu, içinde mikrodenetleyici, seri programlayıcı, kristal osilatör, kapılar, analog ve dijital girişler ve çıkışların yerleşik olduğu bir elektronik karttır. Bu kart, USB kablosuyla bilgisayara bağlanabilir ve hazır bir program, program belleğine girilebilir.

Mikrodenetleyici geliştirme platformunun girişine şunlar bağlanabilir: düğmeler, anahtarlar, hazır klavyeler, sensörler (sıcaklık, basınç, akış, hareket vb. için). Çıkışlar, LED diyotlar, ampuller, bipleyciler, motorlar, ekranlar gibi çeşitli çıkış cihazlarına bağlanabilir.

Arduino platformunun 15'ten fazla farklı modeli bulunmaktadır ve şu anda en popüler olanları Arduino Uno R3, Leonardo, Nano, Pro Mini, Mega 2560 R3, Due'dir.

Raspberry Pi, karmaşık programları çalıştırabilen işlevsel işletim sistemine (çoğunlukla Linux) sahiptir. Tüm Raspberry Pi modellerinde video cihazlarına bağlanmayı oldukça kolaylaştıran grafik kartı bulunur. Raspberry Pi, internet ağına erişimi daha da kolaylaştıran ağ konektörlerine ve Wi-Fi aracılığıyla bağlanma özelliğine sahiptir. Bazı modellerde kablosuz iletişimin başka bir yolu olarak entegre Bluetooth da bulunur

Arduino platformunun temel avantajları şunlardır: sensörlerle kolay bağlanma, çok sayıda ek cihazlar, basit uygulamaları gerçek zamanda yürütmek.

Tekrarlama soruları

1. Mikrobilgisayar sistemi terimi ile neyi tanımlıyoruz?

2. Mikrobilgisayar ve mikrodenetleyici arasındaki temel fark nedir?

3. Daha yeni otomobillerde yerleşik ECU (İng.Electronic Control Unit) mikrobilgisayarın işlevini ve önemini açıklayınız?

4. Arduino UNO platformu hangi donanım bileşenleri içerir?

5. Arduino geliştirme platformlar serisinden birkaç farklı model söyleyin?

6. Arduino Uno ve Arduino Nano geliştirme platformları arasındaki üç temel fark nedir?

7. Arduino UNO geliştirme platformuna kıyasen Raspberry Pi mikrobilgisayarının diğer cihazlara bağlanma olanaklarını açıklayınız!

8. Arduino platformunun Raspberry Pi'ye göre avantajları ve dezavantajları nelerdir?

2. Mikrobilgisayarların Bileşenleri

2.1. Mikrobilgisayar Sisteminin Bileşenleri

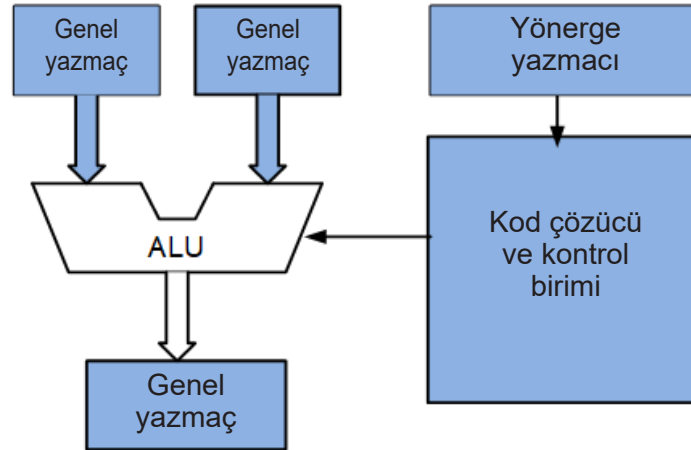
Bir mikrodenetleyicinin veya geliştirme platformunun programlanması için donanım bileşenlerinin özelliklerini bilmek gerekir. Donanım, mikrobilgisayarım yapılmış olduğu mekanik parçaları temsil etmektedir. Donanım bileşenleri şunlardan oluşmaktadır: mikroişlemci, bellekler, veri yolları ve çevresel cihazlar. **Mikroişlemci** verileri işler, **bellekler** verileri saklar, **veri yolları** ise onları iletir. Çevresel cihazlar aynı zamanda giriş-çıkış birimleri olarak da bilinir. Çevresel cihazlar kullanıcı bilgilerini dijital sinyallere, sıfırlara ve birlere veya ters yönde dönüştürür. Kullanıcı bilgileri, ses, görüntü, metin, video gibi insanlar tarafından kullanılan bilgilerdir.

Yazılım donanımı kontrol eder. **Yazılım**, donanımın kullanılmasını sağlayan programlar toplamıdır. İşlevsel olarak programlar üç gruba ayrılır: işletim sistemi, geliştirme ortamları ve uygulama yazılımı. **İşletim sistemi** donanım ve yazılım arasında bir aracıdır. İşletim sistemi, donanım bileşenlerini: işlemciyi, bellekleri, giriş-çıkış cihazları, anakart hatları boyunca veri aktarımını, veri dosyalarını vb. yöneten yazılımdır. İşletim sistemi olmadan bilgisayarlar sıradan insanlar için kullanışlı olmaz. Üçüncü başlıkta farklı işletim sistem türlerini, onların kurulumunu, özelleştirilmesini ve optimizasyonunu öğreneceğiz. **Geliştirme ortamı**, yeni yazılım geliştirmek için bir yazılımdır. Yeni programlar düzenlenir (İng. editing), makine diline çevrilir (İng. compile), hatalardan ayıklanır (İng. debug) ve diğer programlarla bağlanır. Geliştirme programlarının araçlarını ve uygulama yazılımının oluşturulmasını üçüncü ve dördüncü konuda tanıyacağız. [4]

2.1.1. Mikroişlemci

Kişisel bilgisayarlarda mikroişlemci ayrı bir donanım bileşeni, anakart üzerinde kendi yuvası (soketi) bulunan bir çiptir. Arduino platformları, mikroişlemci yerine, mikroişlemciyi ile program ve veri belleğiyle birlikte içeren gömülü bir mikrodenetleyici (Arduino Uno R3 için ATmega328P) kullanırlar. Raspberry Pi 4 mikrobilgisayar sistemi, grafik kartıyla birlikte ortak Broadcom SoC çipine yerleşik olan ARM Cortex-A53 işlemcisini kullanıyor. Mikroişlemcinin çalışma prensibi, ayrı bir çip ya da mikrodenetleyicinin parçası olmasına bağlı olmadan aynıdır.

Mikroişlemci bilgisayarın “beynidir”. Onun İki temel işlevi vardır: programları çalıştırmak (yürütmek) ve anakarttaki diğer cihazları yönetmek. Bir mikroişlemci için programlar, ardışık bellek konumlarında düzenlenmiş yönergeler dizisidir. Mikroişlemci, **programları yönerge yönerge olarak** yürütür. Mevcut yönergeyi işledikten sonra RAM belleğinden bir sonraki yönergeyi alır, işler ve bu işlem işlemci programın sonuna ulaşana kadar tekrarlanır. **Yönetme işlevi**, mikroişlemciyi anakartın efendisi (İng.master) yapar. Bilgisayarın tamamının güvenilir şekilde çalışmasını sağlamak için bilgisayardaki tüm cihazlardan bilgi toplar, işletir, işler ve ardından belirli eylemleri başlatır. Şekil 2.1.'de, mikroişlemcinin temel organizasyonu gösterilmiştir. Bir mikroişlemcinin **oluşturucu parçaları** şunlardır: yazmaçlar, aritmetik-mantık birimi ve kod çözücüyle birlikte kontrol birimi. Yazmaçlar, mikroişlemcinin içindeki mevcut yönergenin verilerini ve işlem kodunu içeren hızlı bellek konumlarıdır. Genel yazmaçlar, işlem verilerini ve işlenmiş verileri içerir, mevcut yönergeyi yürütmek için gereken kodlar ise, özel bir yazmaçta – yönerge yazmacında yerleştirilir.



Şekil 2.1. Mikroişlemcinin temel modelinin blok diyagramı

Aritmetik-mantık biriminde tüm aritmetik ve mantık yönergeleri yürütülür. Kontrol birimi tüm cihazlardan bilgi alır ve ardından kararlar verir, yazmaçların içeriğini değiştirir, pinlerin mantıksal durumunu değiştirir, programları çağırır, çevresel birimleri etkinleştirir vb. Kontrol biriminden çıkan sıfırlar ve birler, donanım bileşenlerini açar veya kapatır. Kontrol bitlerinin sayısı ve işlevi, mikroişlemcinin tasarımı sırasında öngörülmelidir.

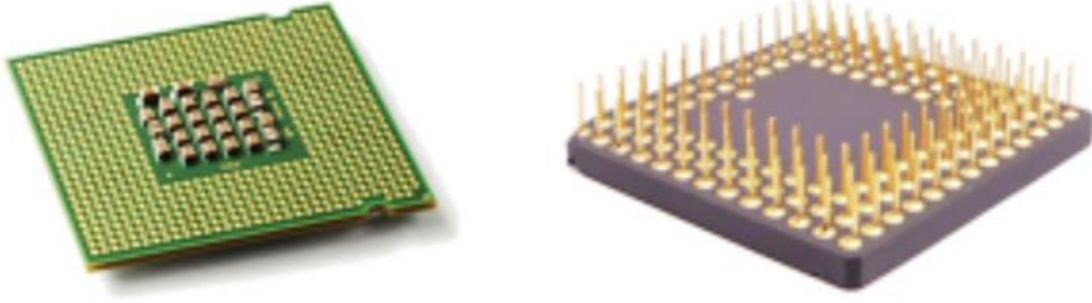
Mikroişlemcinin en önemli özellikleri şunlardır: çalışma frekansı, verilerin büyüklüğü, akış kavramı, yönergeler (komutlar) setinin karmaşıklığı, çekirdek sayısı, pinlerin düzeni vb.

Çalışma frekansı saat sinyalinin frekansıdır. Saat döngüsü önemlidir çünkü her işlemin süresi tamsayı saat döngüsüyle ölçülür. Örneğin, RAM belleğinden bir baytın okunması için üç saat döngüsü gerekir. Birinci saatte mikroişlemci adres gönderir, ikinci saatte bellek taranır ve üçüncü saatte mikroişlemci istenilen veriyi alır. Raspberry Pi'nin GHz sırasından frekansta bir saat sinyali kullandığını, Arduino Uno R3'te ise saat sinyalinin elektronik kartın üzerine yerleştirilen kristal osilatör tarafından üretildiğini ve frekasin 16MHz olduğunu daha önce vurgulamıştık. Arduino platformunda programcı, işlevi kristal osilatörden saatleri saymak olan sayaçlara çok kolay bir şekilde erişebilir. Bu, zamanın ölçülmesi hassas olmasını gerektiren uygulamalarda çok önemlidir, örneğin mutfak cihazı zamanlayıcısı veya trafik ışığı kontrolü. Arduino Uno R3'ün biri 8 bit (0'dan 255'e kadar sayar) ve ikisi 16 bit (0'dan 65535'e kadar sayar) olmak üzere üç sayaça sahip olduğunu belirtelim.

İşlenen verilerin büyüklüğüne göre mikroişlemciler 8, 16, 32, 64 ve 128 bitlik mikroişlemcilere ayrılmaktadır. Raspberry Pi'nin ARM Cortex-A53 mikroişlemcisi 64 bitlidir, Arduino Uno R3'ün ATmega328P mikrodenetleyicisi ise 8 bitlidir. Daha büyük verileri işleyen mikroişlemciler daha güçlüdür ancak onlar için daha fazla donanım kaynağı gerekir.

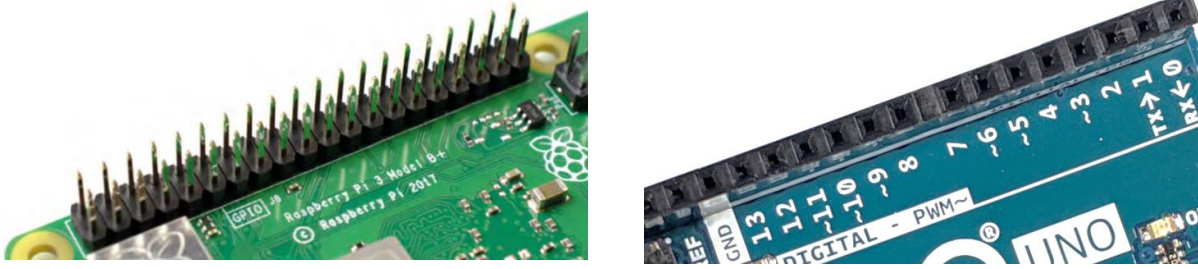
İşlemciler karmaşık veya azaltılmış yönerge (komut) setine sahip olabilir. Günümüzde hemen hemen tüm işlemciler azaltılmış yönerge setine sahiptir (İng. RISC-Reduced Instruction Set Computer) ve bu işlemciler basit donanım yönergeleri yürütür ve çok hızlıdır. ATmega328P mikrodenetleyicisi 10 bitlik işlem kodları kullanıyor ve 131 yönerge yürütebiliyor, bu da ARM Cortex-A53 işlemciyle kıyaslanamayacak kadar azdır.

Pinler, iç hatların metal uzantılarıdır ve mikroişlemciyi mikrobilgisayar sistemindeki diğer cihazlara ve bileşenlere bağlamak için kullanılır. Daha yeni nesil kişisel bilgisayar işlemcilerinde 0,3 mm çapında yaklaşık 2000 pin bulunmaktadır. Daha eski işlemcilerde pinler işlemcinin üzerindedir ve onların boşlukları anakartın yuvasındadır. Bu kavram PGA (Pin grid array) kısaltmasıyla bilinir. Bunun tersi kavramı ise boşlukların işlemcide ve pinlerin yuvada olduğu LGA (Land grid array) kavramıdır.



Şekil 2.2. LGA ve PGA konseptine göre işlemci pinlerinin görünümü

Kişisel bilgisayarlardan farklı olarak, kullanıcı yani programcı Arduino ve Raspberry Pi pinlerine erişebilir ve bunları yönetebilir. Arduino ve Raspberry Pi pinleri derken mikrodenetleyicinin kendi pinlerini veya Broadcom SoC çipini değil, sensörlere, yürütme birimlerine ve sistem ile çalışmayı kolaylaştıran çeşitli ek cihazlara bağlanabilen elektronik kartın kendi pinlerini kastediyoruz.



Şekil 2.3. Raspberry Pi ve Arduino pinlerinin dış görünümü

Örneğin, Arduino Uno R3, pinlerle çalışmak için üç tip yazmaç içerir ve bunlar aracılığıyla pinin giriş veya çıkış olup olmadıklarını belirleriz, giriş pinlerinin durumunu okuyoruz, çıkış pinlerine değerler yazdırıyoruz.

2.1.2. Bellek Organizasyonu

Veriler ve işlemci tarafından yürütülen tüm programlar belleklerde bulunmalıdır. **Belleklerin en önemli özellikleri** kapasiteleri ve hızlarıdır. Bellek kapasitesi bayt (B) veya daha büyük kapasite birimleri olan kilobayt (KB), megabayt (MB), gigabayt (GB), terabayt (TB) ile ölçülür. Bellek hızı erişim süresi veya çalışma frekansı ile ölçülür. Erişim süresi, istenilen verinin bellekte aranması ve bulunması için gereken süredir. Belleğin çalışma frekansı bize belleğin bir saniyede kaç bit alabileceğini veya gönderebileceğini gösterir.

Bilgisayarlar farklı bellek türleri içerir. Bellek organizasyonu, her bellek türü için işlevlerin bölünmesini ve istenen verilere erişme yöntemini ifade eder. Şekil 2.4.'te bellekler piramidi gösterilmiştir. Aşağıya inildikçe belleklerin kapasitesi artar, çalışma

hızları ise azalır. Daha yüksek çalışma hızı, bayt başına fiyat artışına neden olur. Bellekler tepeye ne kadar yakınsa mikroişlemciye de o kadar yakındır. Yazmaçlar, verilerin işlenmeden önce geçici olarak saklanması ve aynı zamanda sonuçların elde edildikten hemen sonra yerleştirilmesi için kullanılır. Yazmaçlar bilgisayarın en hızlı bellek konumlarıdır ancak kapasiteleri sadece birkaç bayttır.

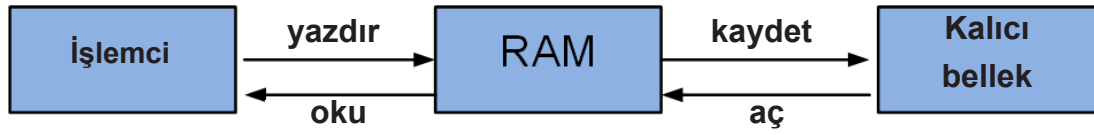
Bekleme süresini azaltmak ve işlemci hızını artırmak için önbellek kullanılır. Önbellek çok hızlı bir RAM belleğidir. Daha yavaş olan belleklerden gerekli veriler önceden getirilerek önbelleğe yerleştirilir ve bu, mikroişlemci için bir tür “bekleme odası” temsil etmektedir. Önbellekte, mikroişlemcinin daha sık kullandığı ve yazmaçlarında yerleştiremediği veriler saklanabilir. Önbellek mikroişlemcinin içinde veya dışında olabilir.



Şekil 2.4. Bellekler piramidi

RAM, İngilizce **rastgele erişim belleği** anlamına gelen **Random Access Memory** kelimelerinin kısaltmasıdır. Şöyle ki, işlemci belleğin tüm konumlarına eşit erişime sahiptir veya hiçbir konum daha düşük veya daha yüksek öncelikli değildir. RAM, aynı zamanda verilerin okunabildiği ve yeni verilerin yazılabileceği bellektir. Değişken içeriğe sahip bir bellektir ve mikroişlemcinin, verilen programın başarılı bir şekilde uygulanması için ihtiyaç duyacağı tüm verileri geçici olarak kaydettiği bir tür çalışma sayfası temsil eden çalışma belleğidir. RAM belleğinde, işlemcide o anda çalışan programlar saklanır. RAM belleği öneminden dolayı birincil bellek olarak da bilinir. RAM **geçici bellektir**. Güç kaynağı kapatılırsa RAM verileri, tekrar açıldıktan sonra geri yüklenemez. Bu nedenle, bilgisayarı kapatmadan önce, RAM belleğindeki veriler, kaydet (İng.save) simgesine basılarak kalıcı bir belleğe (örn. sabit disk) yerleştirilmelidir. Bu, Şekil 2.5'te gösterilmiştir.

RAM belleği işlemciye bilgiler sağlar ve sabit disk gibi bazı kalıcı bellekten bilgi alır. Programlar kalıcı bellekte saklanır ve işlenmeleri gerektiğinde RAM'e aktarılır. RAM belleği daha büyükse, kullanıcı programının daha büyük bir bölümünü birden alarak, mikroişlemcide işlenecektir. Bu şekilde, RAM'den kalıcı belleğe aktarım sayısı azalır, bu da zamandan tasarruf anlamına gelir. Mikroişlemci, programları **birer birer yönerge** şeklinde yürütür. Mevcut yönerge bittikten sonra RAM belleği bir sonraki yönergeyi bulmak için çağrılır. Doğrusal programı söz konusu olduğunda, yönergeler RAM belleğinin ardışık bellek konumlarında (sıralarında) düzenlenir ancak program dallanma ile olduğunda durum böyle değildir.



Şekil 2.5. RAM belleğinin mikroişlemcide çalışma belleği olarak işlevi

RAM, önbellek ve yazmaçlar geçici belleklerdir. Kişisel bilgisayarda verileri kalıcı olarak depolamak için **sabit disk**, **SSD cihazları** ve dış bellekler (CD'ler, USB bellekler, SD kartlar vb.) kullanılır. Sabit disk en büyük kapasiteye sahip bellek cihazıdır. Onda işletim sistemi dahil tüm programları kalıcı olarak saklanır. Manyetik bir bellektir ve saniyede 7.200 devir hızla dönen birkaç manyetik plakadan oluşur. Manyetik plakaların arasına okuma başlı tarak bulunmaktadır.

Sabit disk	SSD disk
Sabit disk bellek cihazı (İng. HDD = Hard Disk Drive)	Kalıcı yarıiletken bellek cihazı (İng. SSD = Solid State Drive)
Manyetik bellek	Elektronik bellek
Maksimum kapasite 15TB	Maksimum kapasite 1 TB
Isınma olur	Isınma yok
Ucuz bellek	Pahalı bellek
Düşük hız	Yüksek hız
Daha kısa yaşam süresi	Daha uzun yaşam süresi

Tablo 2.1. HDD ve SSD bellek cihazlarının karşılaştırılması

Sabit diskin en büyük dezavantajı düşük hızıdır ve bu dezavantaj, işletim sistemini etkinleştirirken en fazla ortaya çıkmaktadır. Bu nedenle, sabit diskler, büyük kapasitelere sahip elektronik çipleri temsil eden yeni SSD bellekler (ing. Solid State Drive) ile değiştirilmeye başladı. Bu bellekler manyetik bellekten neredeyse beş kat

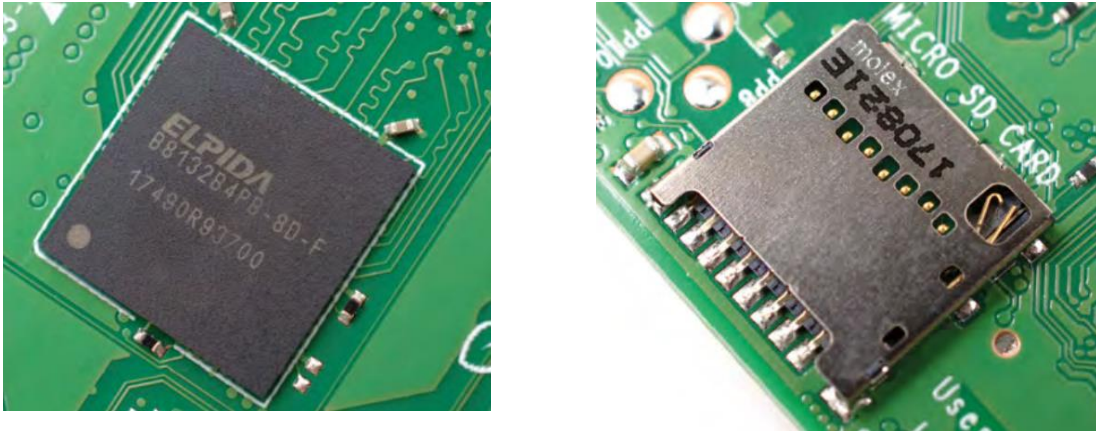
daha hızlıdır, ancak yine de kapasiteleri sabit diskten daha küçüktür. Kalıcı bellek seçerken, bizim için hız veya veri depolama kapasitesi arasında hangisinin daha önemli olup olmadığı göz önüne alınmalıdır.

Tür ve kapasite açısından Arduino ve Raspberry Pi'nin bellek organizasyonu kişisel bilgisayarlardan farklıdır, ancak aynı zamanda kendi aralarında da farklıdır. Arduino Uno R3, üç tür bellek içerir ve hepsi, mikroişlemciyle birlikte ATmega328P mikrodenetleyicisinde bulunur. Bu bellekler şunlardır: RAM belleği, program ve veri belleği. Program belleği flash (İng.flash) bellektir ve onda Arduino Uno R3 programlanırken kullanıcı programı yazılır. Yeniden programlama hızlıdır ve neredeyse sınırsız sayıda tekrarlanabilir. Veri belleği EEPROM'dur (İng. Electrical Erasable Programmable Read Only Memory). Yazma ve okuma bayt bayt yapılır, silme ise eski verinin üzerine yeni verinin yazılmasıyla yapılır. Birkaç Arduino platformu için üç belleğin kapasitesi tablo 2.2.'de verilmiştir. [2]

Arduino platformu	Mikrodenetleyici	Flash bellek	RAM belleği	EEPROM belleği
UNO Rev3	ATmega328P	32 KB	2 KB	1 KB
Mega 2560 Rev	ATmega2560	256 KB	8 KB	4 KB
Leonardo	ATmega32u4	32 KB	2,5 KB	1 KB
Nano	ATmega328P	32KB	2 KB	1 KB

Tablo 2.2. Arduino tabanlı donanım platformlarının bellekleri

Raspberry Pi'deki RAM belleği, elektronik kartın içine yerleşiktir, 1GB'tan 8GB'a kadar kapasiteye sahip özel bir çiptir. Kalıcı bellek olarak elektronik kartta özel yuvası bulunan Micro SD bellek kartını kullanılır. [3]

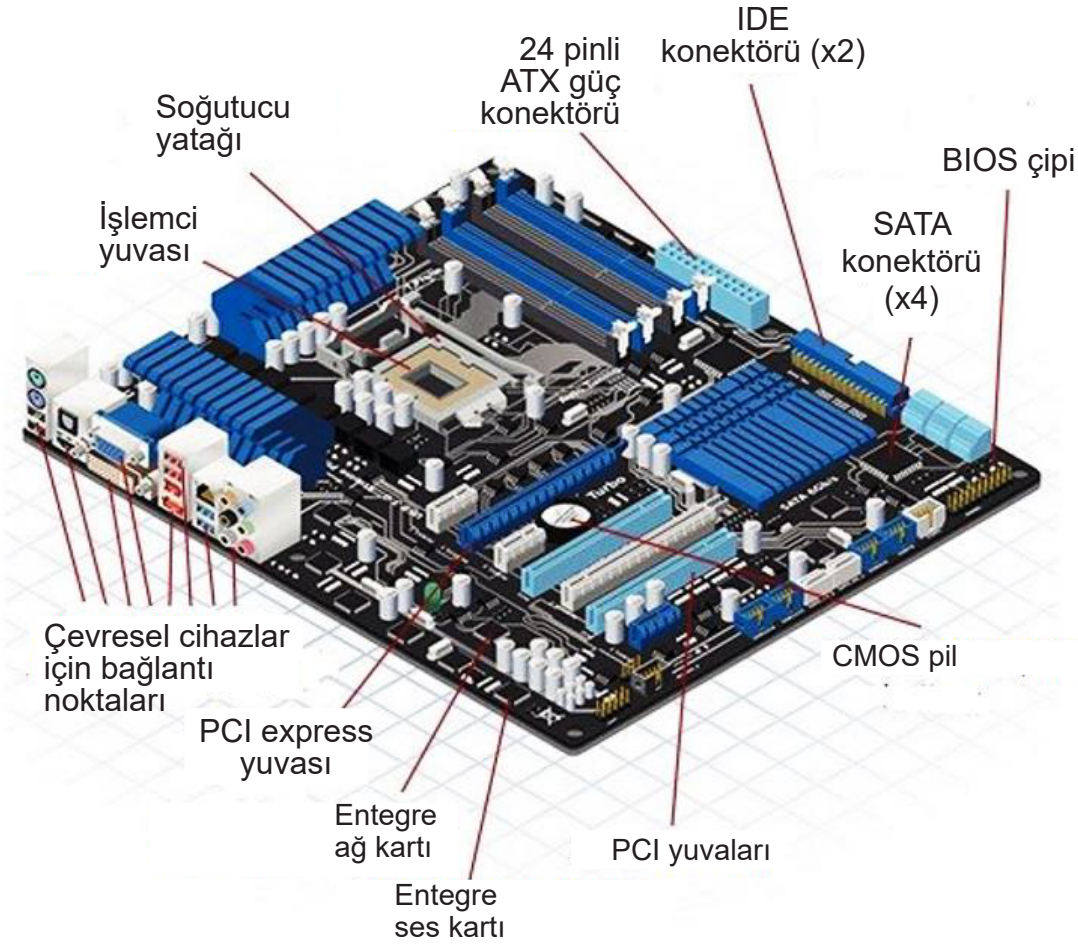


Şekil 2.6. Raspberry Pi için RAM belleğinin ve bellek kartı yuvasının dış görünümü

Bellek kartında, kelimenin tam anlamıyla her şey saklanır: işletim sistemi, programlar ve işlem verileri. Raspberry Pi, 8GB ila 32GB kapasiteli bellek kartlarını destekler. [3]

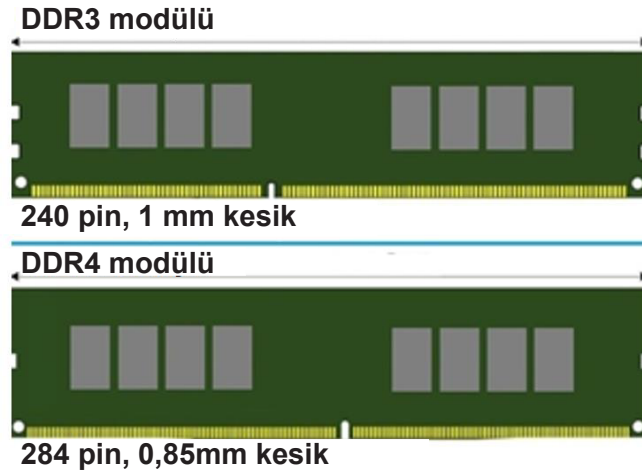
2.2. Anakart - Açıklama ve Bağlantıları

Anakart, tüm donanım bileşenlerini tek bir işlevsel bütüne **bağlar**. Bileşenler fiziksel olarak birbirlerine veriyolu adı verilen bakır hatlarla bağlıdır. Bugün anakartlar sekiz ila on bakır tabakalı hattan yapılıyor. ATX (İng. Advanced Technology eXtended), anakart tasarımında genişletilmiş ileri teknolojidir. Üç temel anakart boyutu vardır: ATX (305 mm × 244 mm), Micro ATX (244 mm × 244 mm) ve Mini ITX (170 mm × 170 mm). Daha büyük boyutlu anakartlarda daha fazla yuva ve daha fazla RAM kapasitesi bulunur. Anakart, bakır hatların yanı sıra, verilerin bir cihazdan diğer cihaza aktarımını kontrol eden çok sayıda farklı konektörler ve çipler içerir. Her birini kısaca tanıyacağız.



Şekil 2.7. Anakart bileşenleri

- İşlemci** → Anakart seçimi işlemci seçimine bağlıdır, çünkü her işlemci türü benzersiz pin düzenine sahiptir ve bu da uygun bir yuva gerektirir.
- Çipset (Yonga seti)** → Çipset, **veri akışını kontrol eden özel entegre devreler setidir**. Hata anakartın kendisi bile içinde yerleşik olan çipsetinin türüne göre adlandırılır. Her çipset **iki temel bileşenden** oluşur: kuzey ve güney köprüsü. Kuzey köprüsü grafik ve bellek denetleyicisidir, çok hızlıdır ve mikroişlemciye doğrudan bağlıdır. Güney köprüsü giriş-çıkış denetleyicisidir ve kuzey köprüsü aracılığıyla mikroişlemciyle iletişim kurar.
- RAM** → RAM belleğinin türü ve kapasitesi anakart seçimine bağlıdır. Günümüzde en yaygın olarak şu RAM bellek türleri kullanılmaktadır: DDR4, DDR3, DDR2 ve DDR. Örneğin, DDR4 RAM modülü DDR3 yuvasına takılamaz. DDRAM (İng.Double Data Rate RAM), bir döngü boyunca aktarılan veri miktarının iki katı olan RAM belleğidir. Aralarında pinlerin sayısı ve düzeni, güç kaynağı ve çalışma frekansı, yani hız bakımından farklılık gösterirler. RAM belleklerinin türünü tanımak için elemanlardan biri pinlerin arasına yerleştirilen kesiktir.



Şekil 2.8. RAM bellek türleri

- Grafik kartı** → **PCI Express×16 ve AGP**, grafik kartlarını bağlamak için en iyi konektörlerdir. PCI Express×16 kısaltması (İng. peripheral component interconnect express), döngü başına 16 bitlik bant genişliğine sahip çevresel bileşenlerin hızlı bağlantısı anlamına gelir. AGP kısaltması (İng.Accelerated Graphics Port), hızlandırılmış

grafik bağlantı noktası anlamına gelir. PCI yuvaları AGP'ye kıyasen daha düşük hız ve sınırlı bellek erişimine sahiptir. Bazı anakartlarda yerleşik grafik kartlarının bulunduğunu da belirtelim ancak video sisteminin yükseltilmesi durumunda ek yuvaların da bulunması iyi olur.

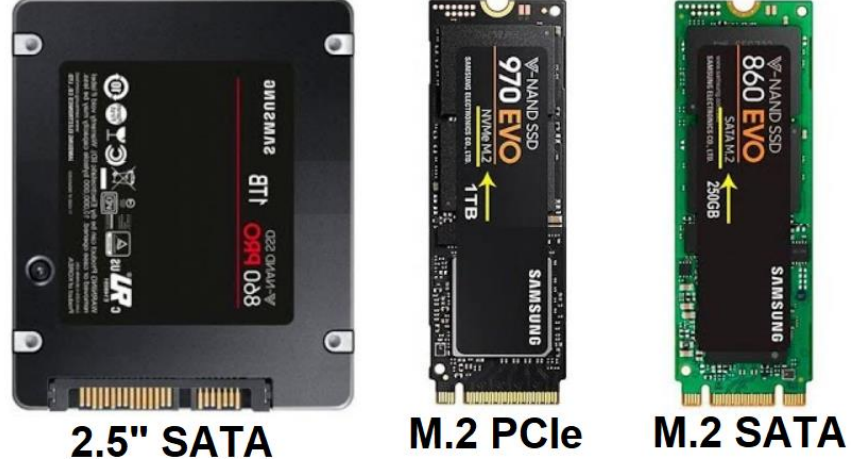
PCI×1 → PCI×1 yuvası, **ses kartlarını, ağ kartlarını, disk sürücüsü denetleyicilerini** ve diğer çeşitli çevresel cihazlar denetleyicileri bağlamak için kullanılır.

BIOS → BIOS (İng. Basic Input Output System) ve **CMOS** (İng. Complementary Metal Oxide Semiconductor) pili, anakartta bulunan, birbirleri olmadan yapamayan ancak özellikleri ve işlevleri bakımından farklı olan iki farklı bellektir. BIOS, bilgisayar açıldığında çalıştırılan programı içeren ROM belleğidir ve buradan sistemin temel giriş-çıkış anlamına gelen kısaltma gelir. BIOS, donanım bileşenlerini kontrol eder, işletim sistemini bulur ve belleğe yükler, sistem güvenliğiyle ilgilenir ve şifreleri kontrol eder. CMOS, güç için kendi pilini kullanan RAM belleğidir. CMOS'ta, saat ve tarih verileri ve bilgisayar başlatıldığında donanım bileşenleri ayarlama verileri saklanır. CMOS kısaltması, entegre devrelerin üretimi için yarı iletken teknolojisini tanımlamak için kullanılır.

Sabit Disk SDD → Çoğu anakart iki tür **SATA** (İng. **Serial Advanced Technology Attachment**) **konektörü içerir: SATA2 ve SATA3**. Kısaltma, seri gelişmiş bağlantı teknolojisi anlamına gelir. SATA2 konektörü mavi renklidir ve sabit diski bağlamak için kullanılır ve ona sabit diskin veri kablosu eklenir. Sabit diskin bir SATA güç kablosu daha vardır ve anakart üzerindeki güç kaynağı birimine bağlanır. SATA3 konektörü genellikle beyaz renktedir ve SSD belleği bağlamak için kullanılır.

M.2 SSD modülleri → Daha yeni anakart modelleri, özel SSD modüllerinin bağlanması için M.2 SSD işaretli özel konektörler içerir. SSD bellek cihazlarından, fiziksel görünüm ve bağlantı açısından farklılık gösterirler ve bu, Şekil 2.9'da gösterilmiştir. SSD modüllerinin kendileri bile iki bağlantı türünde olabilir, PCIe veya SATA. PCIe bağlantılı modülde pinler arasında bir kesik bulunurken, SATA bağlantılı

modülde iki kesik vardır. Bu nedenle modül seçilirken anakartın kendisinde bulunan M.2 SSD konektörünün türüne dikkat edilmelidir. PCIe ile bağlantılı M.2 SSD modülü daha yüksek aktarım hızına sahiptir.



Şekil 2.9. SSD SATA bellek cihazı ve M.2 SSD modülleri arasında karşılaştırma

**Seri, —→
paralel
bağlantı
noktası**

Seri ve paralel bağlantı noktaları daha yeni nesil kişisel bilgisayarlarda bulunma ancak hala kullanımları vardır. Örneğin seri bağlantı, modemleri, yönlendiricileri, endüstriyel kontrolörler gibi RSR-232 cihazlarını bağlamak için kullanılır. Bilgisayarda seri bağlantı noktası yoksa USB seri dönüştürücünün kullanılması gerekir. Paralel bağlantı noktası hala yazıcıları bağlamak için kullanılıyor.

**Video —→
cihazlar**

VGA (İng. Video Graphics Array) analog video standartıdır. Kısaltma, video grafik dizisi anlamına gelir ve standartın kendisi tarafından tanımlanan iki boyutlu renk dizisini ifade eder. Monitörlerin, TV'lerin, projektörlerin ve diğer video ekipmanlarının bağlanması için kullanılır. **HDMI (İng. High Definition Multimedia Interface)**, video ve ses cihazları için konektördür ve iletim kalitesiyle VGA konektörünün üstüne gelmiştir. Kısaltma, yüksek çözünürlüklü multimedya cihazı bağlantısı anlamına gelir.

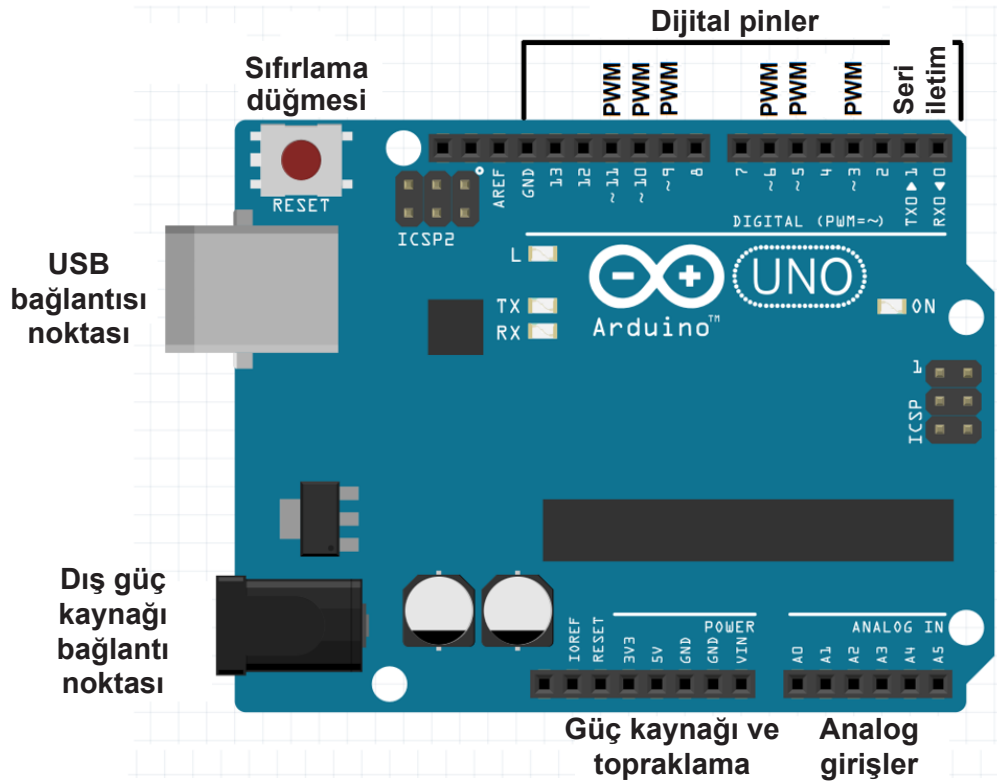
**USB —→
cihazları**

USB bağlantı noktaları, çeşitli çevresel cihazlar için en yaygın olarak kullanılan bilgisayar bağlantı noktalarıdır: fare, klavye, oynatıcılar, dış sabit diskler, kameralar ve fotoğraf makineleri, Arduino ve Raspberry Pi gibi geliştirme platformları. Çoğunlukla plug and play prensibiyle çalışırlar, yani bu cihazlara ait yazılımlar otomatik olarak yüklenir.

2.3. Arduino Uno R3'ün Bileşenleri

İlk başlıkta Arduino Uno R3 mikrodenetleyici geliştirme platformunun önemini ve uygulamasını tanıdık. Bu konuda donanımı tanıyacağız: bu platformun bileşenlerini, sensörleri ve zayıflatıcıları. Bir sonraki modül biriminde Arduino Uno R3 platformunun yazılımını ve geliştirme ortamını tanıyacağız.

Arduino Uno R3'te temel bileşen, Atmel üreticisinin ATmega328 mikrodenetleyicisidir. Bu mikrodenetleyici, RISC teknolojisiyle yapılmış, 28 pinli, 8 bitlik çiptir. Program belleği 32 KB, veri EEPROM belleği 1 KB ve iç RAM belleği 2 KB kapasiteye sahiptir. Maksimum frekans 20 MHz'dir. Arduino Uno R3, bilgisayara USB kabloyla bağlanır ve platformun kendisi de küçük bir USB bağlantı noktasına sahiptir. Bu kablo aracılığıyla programlar bilgisayardan Arduino Uno R3'e aktarılır ve ATmega 328 mikrodenetleyicinin program belleğinde yazılır. Programlandıktan sonra Arduino Uno R3 bilgisayardan çıkarılıp başka bir elektronik cihaza takılıyor ve işlem yönetimi için kullanılıyor.

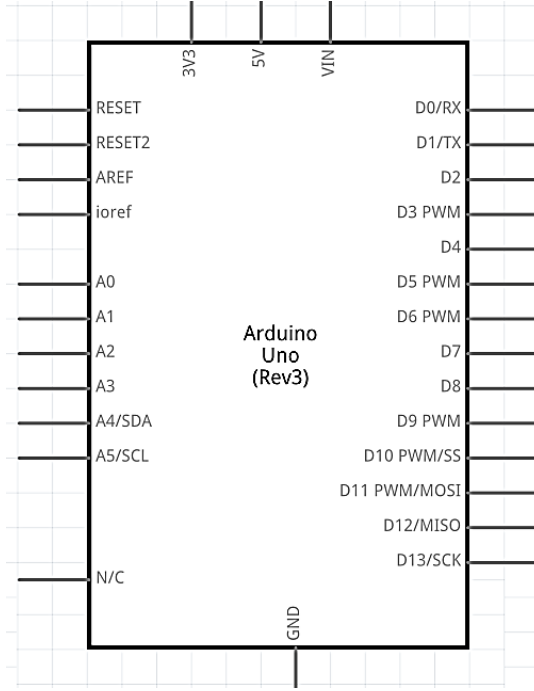


Şekil 2.10. Arduino Uno R3'ün bileşenleri (üstten bakış)

Arduino Uno R3 üç sinyalli LED diyotlar içerir. ON işaretli LED diyotu güç içindir. TX ve RX (Transmit-Receive) işaretli LED diyotlar seri iletişim göstergeleridir. Bu iki LED diyotu, mikrodenetleyiciye bir program girildiğinde daha yoğun şekilde yanıp

söner. Yukarıdaki LED diyotlara ek olarak, 13. dijital pin üzerinde yerleşmiş olan ve kırpma (İng.Blink) programını çalıştırarak donanım platformunun doğruluğunu kontrol etmek için kullanılan, L harfi ile işaretlenmiş yerleşik bir LED diyotu daha bulunmaktadır.

Arduino Uno R3, bilgisayara bağlı olduğunda USB bağlantı noktasından güç alır. Aksi takdirde, 7 ila 12 V gerilime sahip bir pil veya adaptör üzerinden güç alabilir.

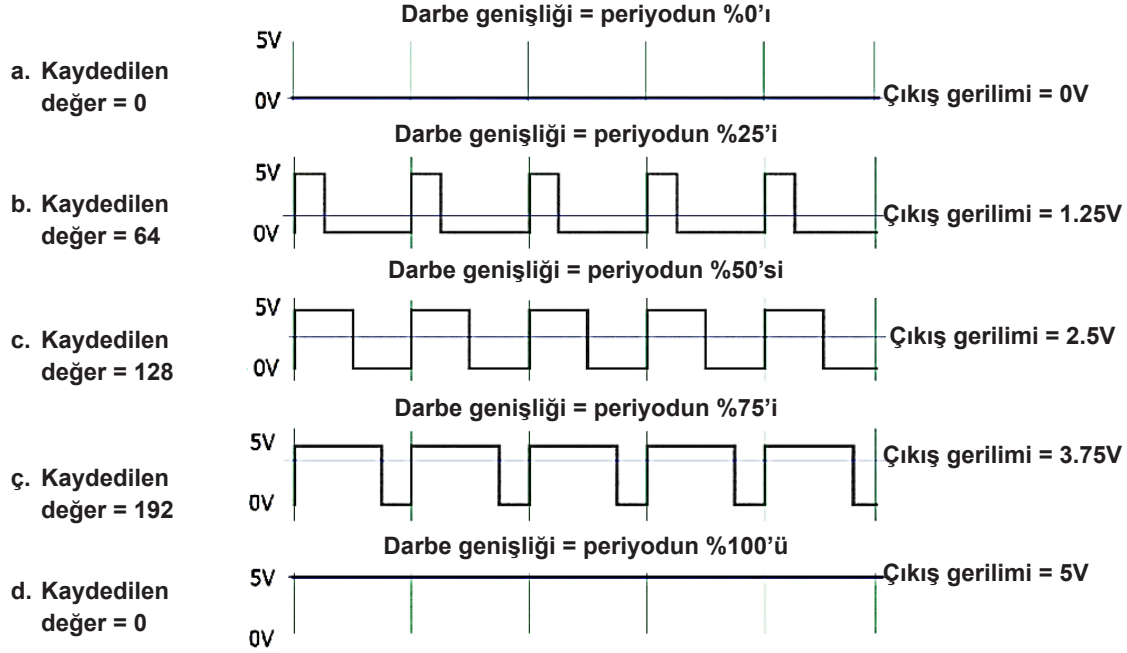


Şekil 2.11. Arduino Uno R3'ün işlevsel şeması

Şekil 2.11'de, Arduino Uno R3 platformunun işlevsel şeması gösterilmektedir. **14 dijital pine** sahiptir. Dijital pinler giriş veya çıkış olabilir ancak çift yönlü olmaz. Dijital pinlerle çalışmanın en basit elemanları giriş elemanı olarak düğme ve çıkış elemanı olarak LED diyotudur. Dijital sinyaller sadece iki değer alır; mantıksal sıfır veya mantıksal bir. Örneğin, tuşa basıldığında, mikrodenetleyici mantıksal bir verisini alır (5V'luk gerilime bağlanır) ve serbest bırakıldığında, mantıksal sıfır durumudur (topraklamayla bağlantı).

3,5,6,9,10 ve 11 numaralı pinlerin ikili işlevleri vardır. Bunlar, dijital pinler veya analog çıkışlar olarak kullanılabilirler. Analog çıkışlar, darbe genişliği modülasyonu olarak tercüme edilen PWM (İng. Pulse Width Modulation) işareti ile belirtilmiştir. Bu süreç ile analog sinyal darbeler ve duraklamalar dizisine dönüştürülür. Arduino Uno R3'ü programlarken ve analog çıkışlarla çalışırken, programcı 0 ila 255 aralığında tamsayı değerleri verir (yazar). Darbe genişliğinin tamsayı değerinin büyüklüğüne bağımlılığı Şekil 2.12'de gösterilmiştir. Darbelerin genişliği veya süresi ve çıkış geriliminin büyüklüğü kaydedilen tam sayı değerine doğru orantılı bağlıdır. Örneğin, 65 değeri, izin verilen maksimum değer olan 255'in %25'ini temsil eder, dolayısıyla darbe genişliği, periyot değerinin %25'i olacak ve çıkış gerilimi, 1,25 volta eşit olan maksimum değerinin (5V) %25'i olacaktır. Darbe genişliği modülasyonu analog girişler için de kullanılır ancak ters yönde yani yazma yerine okuma gerçekleştirilir. 0V ila 5V aralığına değişen giriş gerilimi, değişken genişlikte darbe dizisine dönüştürülür ve

darbelerin genişliği, giriş miktarı olarak okunacak tamsayı değerini belirler. Tamsayı giriş değerleri 0 ila 1023 aralığında yani çıkıştan dört kat küçük aralıktadır. [5]



Şekil 2.12. Analog çıkış pinlerinde darbe genişliği modülasyonunun uygulanması

Ayrıca Arduino Uno R3'ün birinci ve sıfırncı pinleri de ikili işleve sahiptir. Dijital pinler olarak ve seri iletişim için pin olarak da kullanılabilirler, örneğin iki Arduino platformunu veya Arduino platformunu Raspberry Pi'ye bağlarken.

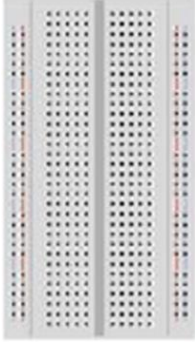
Topraklama pini (İng.GND, Ground) ve 5 V'luk gerilim pini, tüm elektronik bileşenlerin belirli elektrik şemasına göre hassas bir şekilde bağlandığı devre tahtasında güç sağlamak için kullanılır.

2.4. Arduino Uno R3'ün Elektronik Bileşenleri ve Onların Bağlanması

Arduino platformunun dış dünyayla iletişim kurabilmesi ve kendisini çevreleyen ortam hakkında bilgi alabilmesi için giriş birimlerine ihtiyaç vardır: **sensörler** (piezo, eğim sensörü, foto rezistör, sıcaklık sensörü), düğmeler, potansiyometreler. Arduino platformunun belirli bir süreç üzerinde etkilemesi için aktüatör olarak bilinen çıkış birimlerine ihtiyaç vardır. Bu kategoriye şunlar aittir: ekranlar, motorlar, sesli uyarıcılar.

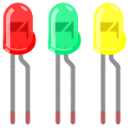
Giriş-çıkış birimlerinin Arduino Uno R3 platformuna etkili bir şekilde bağlanabilmesi için aşağıdaki verilen ek elektronik bileşenler gereklidir: Rezistörler, kapasite-

törler, doğrultucu diyotlar, optokuplörler, yönlendiriciler vb. Herbirinin işlevini kısaca anlatacağız. [5]



Devre tahtası

→ Devre tahtası, elektronik ve lehimleme konusunda fazla bilgisi olmayan yeni başlayanlar için çok iyi bir seçimdir. Plastik plakanın altına dikey ve yatay iletkenler yerleştirilmiştir. Plastik plakanın boşluklarında elektrikli bileşenlerin çıkışları ve bunların bağlanması için kablolar, jumper (atlama) kabloları yerleştirilir. Devre tahtasıyla çalışma talimatları modüler birimin uygulamalar bölümünde verilmiştir.



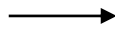
LED diyotlar



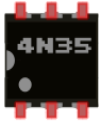
LED diyotlar en basit çıkış elemanlarıdır. LED diyotun yanması için, daha uzun bağlantısının (anot) daha kısa bağlantısından (katot) daha yüksek potansiyele bağlanması gerekir.



Düğmeler



Düğmelerin iki karşıt tarafta ikişer olmak üzere dört bağlantısı noktası vardır. Aynı **taraftaki bağlantılar elektriksel olarak bağlı değildir** ve düğmenin basıldığında elektrik devrenin kapanması için devre tahtası üzerindeki iki farklı iletkenin boşluklarına yerleştirilmeleri gerekmektedir.



Optokuplör



Optokuplör LED diyottan ve fotodiyottan oluşur. Akım fotodiyot devresinden, sadece LED diyotu yanarken akar. LED diyotu güç kaynağına bağlıdır, fotodiyot ise tüketiciye bağlanır. Optokuplör yardımıyla kaynağın ve tüketicinin akım devreleri elektriksel olarak izole edilir.



Potansiyometre



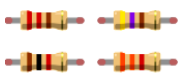
Potansiyometre değişken bir rezistördür ve değişken gerilim oluşturmak için genellikle Arduino Uno R3'ün analog girişlerinden biriyle bağlanır.



Fotorezistör



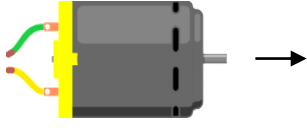
Fotorezistör, yüzeyine düşen ışığa bağlı olarak değişken dirence sahiptir.



Rezistörler

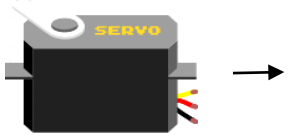


Rezistörler, elemanlara zarar vermemek için çalışma akımının gücünü ayarlamak için kullanılır.



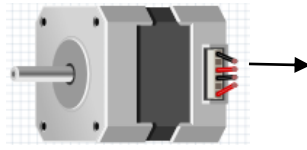
Doğru akım motoru

Doğru akım motoru, dairesel hareketler yaparak elektrik enerjisini mekanik enerjiye dönüştürür. Akımın yönü değişirse motorun dönüş yönü de değişecektir.



Servo motor

Servo motor tam bir daire boyunca dönmez, sadece 180 dereceye kadar döner. Giriş gerilimine bağlı olarak belirli bir açı için belirli bir konuma hareket edecek ve giriş gerilimi değişene kadar bu konumda kalacaktır. Dönme açısı, gerçekleştirilen darbe genişliği modülasyonundan sonra darbelerin genişliğine bağlıdır.



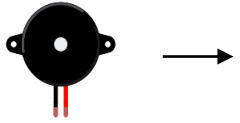
Adım motoru

Adım motorları ayırık hareketler gerçekleştirir, yani her giriş darbesinin gelişiyile motor belirli bir açı için hareket ediyor. Bu harekete adım denir. Tam bir daire yapması için gereken adım sayısı, motorun yapısına (kutupların sayısına) bağlıdır.



Röle

Röleler, düşük gerilim veya düşük akım sinyallerini kullanarak yüksek güçlü tüketicilerin açılmasını veya kapatılmasını sağlar. İki tür temas noktası vardır, Normal kapalı ve normal açık temas. Röle aktif olduğunda normal açık olan temaslar kapatılır ve tüketici güç kaynağına bağlanır.



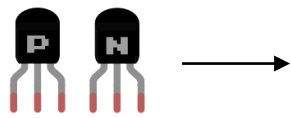
Piezo bileşeni

Piezo bileşeni, titreşim sensörü veya farklı frekanslarda ses üreticisi olarak kullanılabilir.



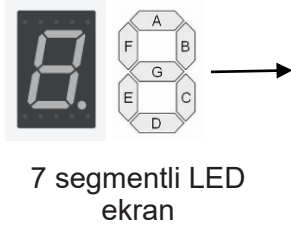
Eğim sensörü

Eğim sensörü, konum sensörüdür. İç kısmında hareket eden metal bir top bulunmaktadır ve böylece teması açar veya kapatır, yani sensörün bağlantılarını keser veya bağlar. Temasın yatay veya dikey konumda kapatılacağı sensör tipine bağlıdır.



Transistörler

Transistörler gerilim veya akım yükselteçleri ve anahtarlar olarak kullanılır. 2N2222 transistörü 500 mA'e kadar akımlar için, TIP120 transistörü ise 5 A'e kadar akımlar için uygundur.



7 segmentli LED ekran

7 segmentli ekran onluk rakamların görsel gösterimi için kullanılır. Toplam sekiz pinden yedi pini bölümleri uyarmak için kullanılır, bir pin nokta için ve iki pin topraklama içindir. 7 segmentli ekran ortak anot veya ortak katot ile olabilir ve bu, pinlerin uyarılması için önemlidir.



Doğrultucular

H köprüsü (İng. H-bridge), doğru akım motorunun yönünğn değişmesini sağlayan elektronik devredir. Akım yönündeki değişiklik, motorun dönme yönünde de değişikliğe neden olur. Anahtarlama elemanlarından veya L239D gibi entegre devre şeklinde yapılır.



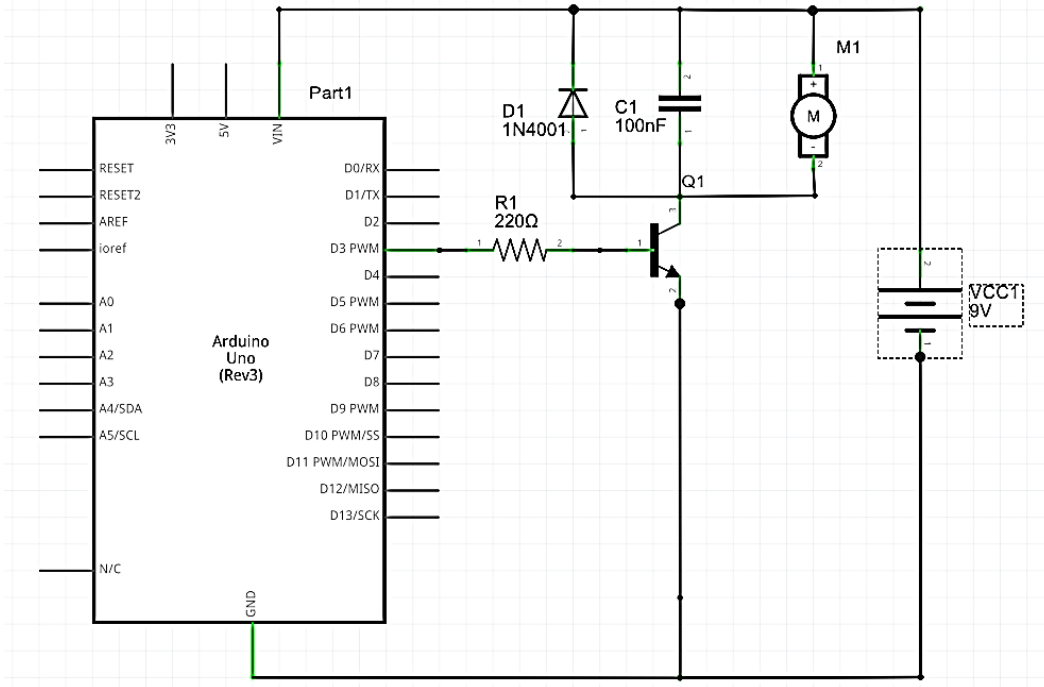
Kapasitörler

Kapasitörler gerilimi kaynak gerilimden küçükse kapasitör dolar, aksi takdirde boşanır. Kapasitörler genellikle devredeki diğer bileşenlerin neden olduğu hızlı gerilim değişikliklerini önlemek için genellikle sensörle veya motorla paralel olarak bağlanır.

Bileşenleri devre tahtasına bağlamaya başlamadan önce, bileşenleri özelliklerini analiz edip seçmeliyiz. Her elektronik bileşen üreticisi, bileşenin çalışması için gereken minimum gerilim değeri ve izin verilen maksimum gerilim gibi performans, kullanım ve özellikler hakkında bilgiler içeren teknik-teknolojik belgeler hazırlar. Rezistör ve sıradan kapasitör gibi bazı bileşenler kutuplu olmayan bileşenlerdir, dolayısıyla hangi bağlantı noktanın daha yüksek, hangisinin daha düşük elektrik potansiyelinde olacağı önemli değildir. Diyotlar, transistörler, entegre devreler kutuplanmış bileşenlerdir ve bağlantı noktalarının tanımlanması gereklidir.

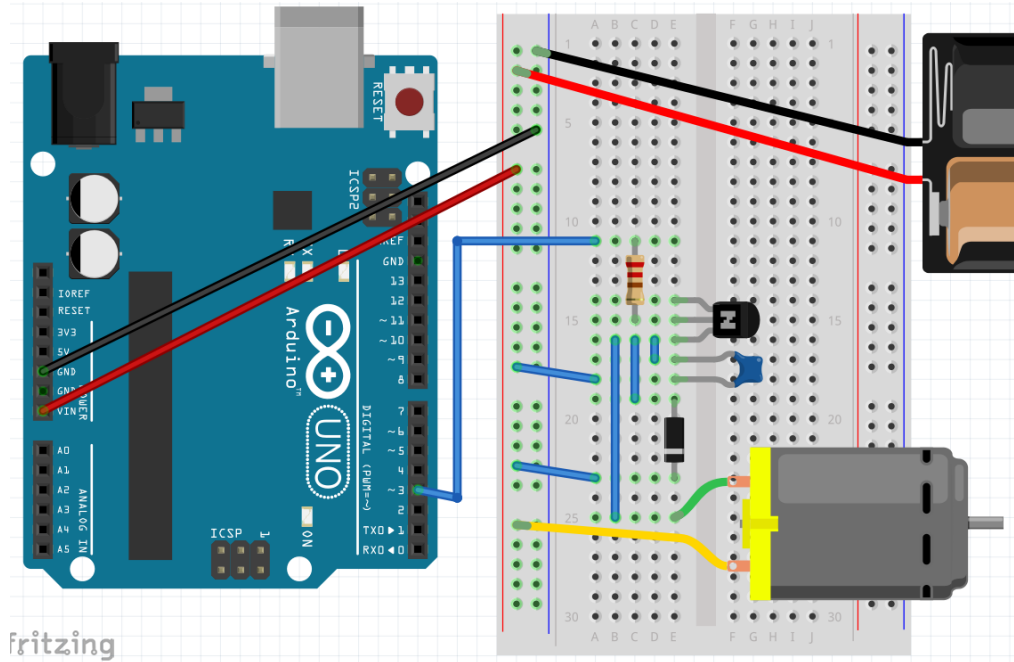
Arduino Uno R3'e, çalışma gerilimi 5V'a ve izin verilen maksimum 40mA akıma sahip bileşenler bağlanabilir. Bu değerler tüm Arduino platformları için aynı değildir. Örneğin Arduino Uno WiFi Rev2 platformu 20mA maksimum izin verilen akımla, Arduino Zero ise 7mA ile çalışmaktadır.

Şekil 2.13'te ve Şekil 2.14'te, doğru akım motorunu Arduino Uno R3 platformuna bağlamak için işlevsel ve montaj şeması gösterilmiştir. İşlevsel şemada semboller elektronik bileşenleri, çizgiler ise bunların bağlanma şeklini temsil eder. Elektrik devresi basitse, az sayıda bileşeni varsa, bağlantı şemasının anlaşılması belki daha kolay olur. Ancak çok sayıda bileşen ve bunları bağlamak için kabloların olması durumunda bize daha fazla görünürlük sağladığı için işlevsel şema gereklidir. [6]



Şekil 2.13. Doğru akım motorunun Arduino Uno R3'e bağlanmasının işlevsel şeması

Şekil 2.13'te gösterilen şemada, transistörün anahtar işlevi vardır.



Şekil 2.14. Doğru akım motorunun Arduino Uno R3'e bağlanmasının montaj şeması

Transistörün tabanı, genişlik modülasyonu darbelerle analog çıkış olarak kullanılan Arduino Uno R3'ün üçüncü pinine bağlıdır ve motoru besleyen gerilim, yani içinden akan akımın gücü darbelerin süresine bağlıdır. Transistörün teknik-teknolojik

belgelerinden izin verilen maksimum kolektör-emitör geriliminin elektrik motorunun güç kaynağının geriliminden daha yüksek olup olmadığını ve kolektör akımının büyüklüğünün motordan akan akımdan %25 daha yüksek olup olmadığını kontrol etmeliyiz. Beyz akımı, kolektör akımına ve akım yükseltme katsayısına bağlıdır. Örneğin akım yükseltme katsayısı 100 ise ve istenilen kolektör akımı 1A ise o zaman beyz akımı $1A/100=0,01A=10mA$ olacaktır. Ohm yasasını kullanarak üçüncü dijital pine bağlı direncin değerini hesaplayabiliriz. Örneğin $5 / 0.01 = 500\Omega$ ve bu standart bir değer olmadığından 470Ω seçiyoruz. Diyot motora paralel olarak bağlıdır ve Arduino Uno R3 pinlerini, motorun açılıp kapatıldığında meydana gelen büyük endüksiyon akımlarının oluşmasından korur. Dijital pinlerinde durum değişikliği çok hızlı olabilir ve devrede kaynağın geriliminde salınımlar meydana gelir ve bunlar bir kapasitör kullanılarak filtrelenmektedir.

Arduino Uno'nun güç kaynağı dikkat edilmesi gereken başka bir özelliktir. Arduino Uno R3'ün USB veya dış güç ile iki şekilde güç ile beslenebileceğini önceden biliyoruz. Programın yazdırılmasından sonra Arduino Uno R3'ün bilgisayara bağlanmasına gerek kalmaz ve o zaman dış bir güç kaynağı kullanılabilir, pil veya alternatif gerilimi 7 ila 12V değerinde doğru akıma dönüştüren adaptör. 12V'un üzerindeki gerilimler elektronik karta zarar verebilir. Pilin gerilimi de 7 ila 12V aralığında olmalıdır. Pilin kapasitesi mAh ile ifade edilir. Pilin ne kadar süreceği, elektronik cihazda yerleşik bileşenlerin elektrik enerji tüketimine bağlıdır.

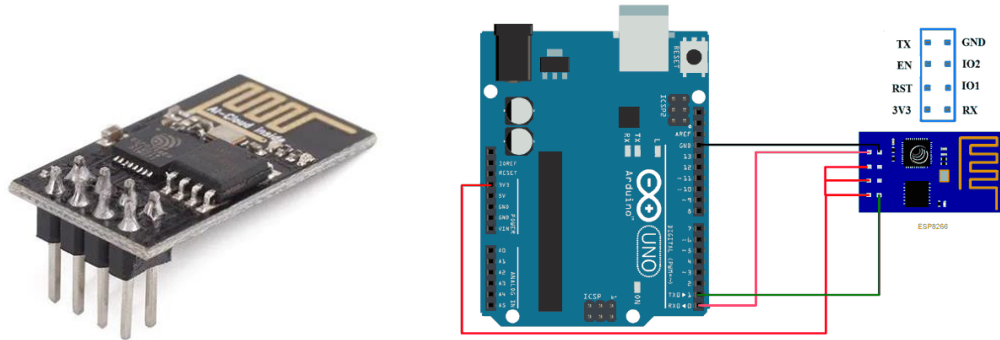
Arduino Uno R3'ü doğru akım motor ile bağlantı şemasının analizi, elektronik bileşenlerin geliştirme platformuna bağlanmadan önce işlevinin ve özelliklerinin nasıl incelenmesi gerektiğine dair sadece bir örnektir.

2.5. Arduino Platformu Eklentileri

Arduino platformu, kullanıcıları arasında oldukça popüler hale geldi ve bunun için hem donanım hem de yazılım olarak açık kaynak (İng. open source) kavramının büyük katkısı vardır. Cihazlarının tasarımını deneyen birçok kullanıcı, elektronik bileşen üreticilerine Arduino ile çalışmayı büyük ölçüde kolaylaştıran ve işlevselliğini önemli ölçüde artıran hazır eklentiler (aksesuarlar) oluşturma ve piyasaya sürme fikrini verdi. Eklentiler aslında amacı kesin olarak belirlenmiş, hazır elektronik kartlardır. Bağlanma basittir ve çoğu eklenti Arduino'nun dişi pinlerine takılan uzun erkek pinler içerir. **Kalkanlar (İng.Shields)**, temel Arduino kartının üzerine yerleştirilen, iki kartın üst üste olduğu ancak uzun erkek pinler nedeniyle birbirine değmeyen eklentilerdir. Kalkan isminin sembolik adı vardır. Arduino yazılımı açık

kaynak olduğundan dolayı kullanıcılar, ihtiyaçlarına göre değiştirebilecekleri çok sayıda hazır eklenti programlama koduna erişebilirler. Ayrıca bu eklentilerin çoğunun Arduino entegre geliştirme ortamında yazılım desteği vardır. Bir geliştirme kartı eklentisi kullanıldığında, program kodunda kütüphane dahil edilmelidir ve kütüphane olmadan eklenti Arduino ile iletişim kuramaz. Kütüphaneler, donanımla çalışmayı büyük ölçüde kolaylaştıran ek yönergeler içerir. Daha sonra göreceğimiz gibi, eklentilerin boyutu, 1-2 santimetre ve 4 pinden, kalkanlar gibi geliştirme kartının kendisi ile aynı boyut ve pin sayısına sahip olan eklentilere kadar değişmektedir. Genellikle Arduino platformu için eklentiler yeni başlayanlar için değildir ve bunları pratik kısımda kullanmayacağız ancak sundukları olanakları bilmek iyidir. Bir eklenti ve beş kalkan tanıyacağız.

İnternet erişimi veya Wi-Fi bağlantısı olmadan Arduino dış dünya ile iletişim kuramaz, sensörlerden alınan verileri gönderemez veya uzaktan kontrol edilemez. Arduino'yu ağa bağlamak için, Wi-Fi eklentisi veya Ethernet kalkanı gibi özel bir eklenti gerekir. ESP8266-01 eklentisi, 25mmx15mm boyutlarına sahip çok düşük maliyetli Wi-Fi modülüdür ve bu da onu özellikle Nesnelerin İnterneti platform uygulamalarında Arduino Uno R3'ün ağ bağlantısı için uygun yapar. [6]



Şekil 2.15. ESP8266-01 eklentinin dış görünümü ve bağlanmanın montaj şeması

ESP8266-01 eklentinin dış görünümü ve Arduino Uno R3'e bağlamak için montaj şeması Şekil 2.15'te gösterilmektedir. Sekiz pin içerir ve bunlar Arduino Uno R3 pinlerine yerleştirilemez ancak bunları bağlamak için teller kullanılır. ESP8266-01 eklentisinde yerleşik USB bağlantı noktası yoktur ancak TX ve RX pinleri aracılığıyla universal asenkron seri iletişim kullanır. İki giriş-çıkış pini bağlamak için, örneğin sensör bağlamak için kullanılabilir. Wi-Fi bağlantısı için maksimum mesafe 100-250 metredir. Bu modülün kütüphanesi Arduino'nun entegre geliştirme ortamının bir parçası değildir ve ek kurulum gerektirir.

Şekil 2.16'da, sekiz eşzamanlı bağlantıyı destekleyen Arduino Ethernet 2 kalkanı gösterilmiştir. Arduino Uno R3 ile uzun erkek pinler üzerinden bağlanır ve pin şeması aynı olduğundan üzerine başka bir kalkan eklemek mümkündür.



Şekil 2.16. Arduino Ethernet 2 kalkanı

İnternet ağına bağlanmak için RJ-45 konektörü kullanılır ve ağda bakımı yapılması gereken belgeleri depolamak için yerleşik bir bellek kartı yuvası vardır.

GSM/GPRS Arduino kalkanı, Arduino'nun mobil ağda ağa bağlanmasını sağlayan eklentidir.

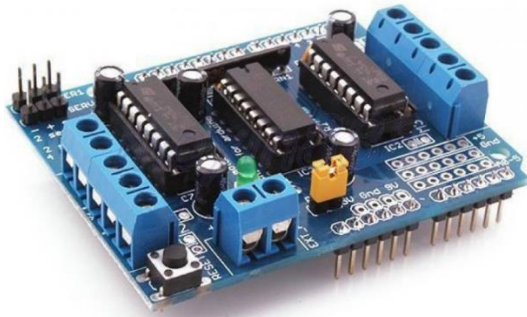


Şekil 2.17. GSM/GPRS Arduino kalkanı

Kullanıcı, SMS mesajları alıp gönderebilir, arama yapabilir ve yanıtlayabilir küresel mobil ağ ve paket odaklı veri aktarım hizmeti (İng.Global System Mobil / General Packet Radio Service) aracılığıyla İnternet'e erişebilir. Kalkanda yerleşik SIM kart yuvası, kulaklık

konektörleri, anten, antenli Bluetooth 3.0 ve Arduino Uno R3'ün bir cep telefonu işlevini yapması için gereken her şey bulunur.

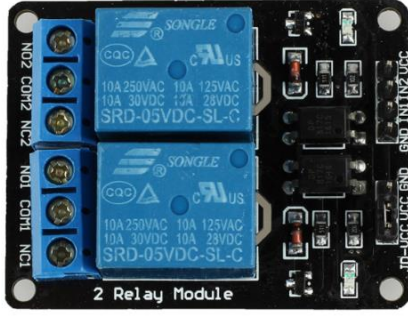
Şekil 2.18'de. Dört doğru akım motoru veya iki servo motoru veya 2 adım motoru kontrol edebilen Arduino motor kalkanı gösterilmiştir.



Şekil 2.18. Arduino motor kalkanı

Bu motorlar, yüksek akımlar ve yüksek güçler nedeniyle sadece Arduino tarafından doğrudan kontrol edilemez. Arduino motor kalkanı, motorun güç kaynağında akım bölünmesini gerçekleştirir ve Arduino platformunun mantığını kullanır. İki L293D entegre devre içerir ve motor dönme hızının kontrolü için iki kanallı bir H köprüsü temsil etmektedir.

Arduino, yüksek güçlü tüketicileri kontrol etmek için tasarlanmamıştır; 40mA'ya kadar çalışma akımla maksimum gerilim 5V'tur.



Şekil 2.19. Arduino röle kalkanı

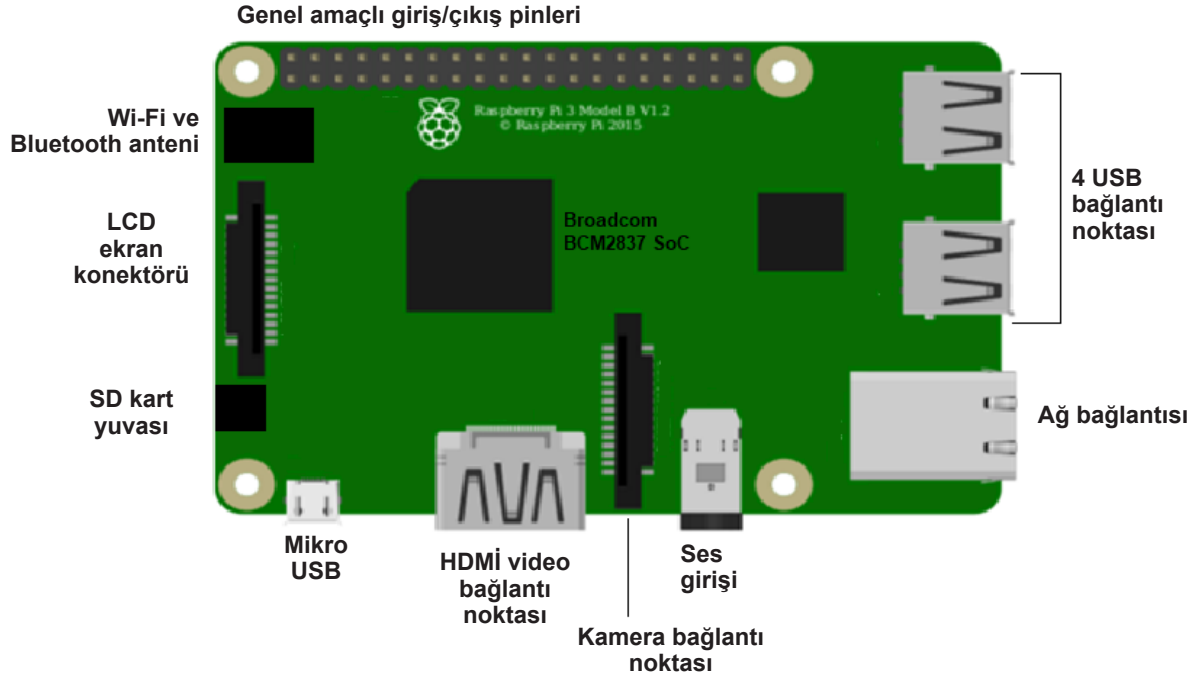
Röleler, büyük tüketiciler ile yönetim sisteminin yüksek gerilim kısmından, alçak gerilimin galvanik olarak ayrılmasını sağlar. Şekil 2.19'da, 250V çıkış alternatif gerilime veya 30V doğru gerilime ve 10A maksimum akıma sahip iki kanallı Arduino röle kalkanı gösterilmiştir.

Arduino platformunun işlevselliğini arttırmak için kullanılan eklentilerin tamamını listelemek mümkün değildir. Seçim ne olursa olsun, her zaman önce cihazların nasıl bağlanacağı ve olası hasarlardan nasıl korunacağı ile ilgili tüm bilgileri içeren teknik-teknolojik belgeler okunmalıdır.

2.6. Raspberry Pi 3B+'nın Bileşenleri

Raspberry Pi, kredi kartı boyutunda ve çok düşük maliyetli, tek kart üzerinde çok iyi bir mikrobilgisayardır. Bununla internette arama yapabiliriz, video oyunları oynayabiliriz, bilgisayar programları yazabiliriz, elektrik devreleri oluşturabiliriz vb. Biz Raspberry Pi 3 B+ modelini tanıyacağız. 2019 yılında, önceki modele göre çok daha büyük RAM kapasitesine (1GB, 2GB veya 4GB) ve üç ila dört kat daha yüksek çalışma hızına sahip yeni model Raspberry Pi 4 B modeli piyasaya çıktı. Raspberry Pi'yi evde kişisel bilgisayar olarak veya web görüntülerini işlemek (İng. Computer Vision) için kullanmak istiyorsak bu model idealdir. Diğer modellerden farklı olarak Raspberry Pi 4 B modelinde ısınma sorunu meydana gelir, hatta soğutucu kullanılması tavsiye edilmektedir. Raspberry Pi'nin amacı elektronik veya evlerimizin otomasyonu alanında projeler ise Raspberry Pi 2 veya 3 B modelinin kullanılması tavsiye edilir. [7] Tüm Raspberry Pi modellerinin uyumlu olduğunu vurgulayalım, yani bir model için yazılan yazılım diğer herhangi bir modelde çalışabilir.

Şekil 2.20'de Raspberry Pi 3 B+ modelinin üçüncü revizyonu gösterilmektedir. Diğer tüm bilgisayarlar gibi Raspberry Pi de çeşitli bileşenlerden oluşup bunlardan en önemlisi Broadcom BCM2837 SoC (System on Chip) sistem çipidir. Bu çip şunları içerir: 1,4 GHz çalışma frekansına sahip 64 bitlik ARM Cortex-A53 dört çekirdekli işlemci, 32 KB L1 önbellek, 512 KB L2 önbellek ve VideoCore IV grafik işlemcisi. RAM belleği, Arduino Uno R3'te olduğu gibi işlemci ile aynı çip üzerine entegre değildir. Geliştirme kartında maksimum 1 GB kapasiteye sahip ayrı bir çiptir. Aşağıdaki şekilde RAM belleğini göremiyoruz çünkü alt tarafta bulunmaktadır. [3]



Şekil 2.20. Raspberry Pi 3B+'nın bileşenleri (üstten bakış)

Raspberry Pi logosunun kazındığı metal kapağın altında radyo vericisi bulunur. Bu verici Raspberry Pi'yi Wi-Fi aracılığıyla yerel bilgisayar ağlarına ve İnternet ağına bağlamak veya Bluetooth aracılığıyla diğer akıllı cihazlara (sensörler, cep telefonları) bağlanmak için kullanılır. Ağ ve USB denetleyicisi, **ağ bağlantı noktası ve dört USB bağlantı noktası** üzerinden iletim kurmak için sorumlu çiptir. Mini USB bağlantı noktasının hemen yanında, geliştirme kartındaki tüm bileşenlere güç verilmesiyle ilgilenen daha küçük bir çip bulunur.

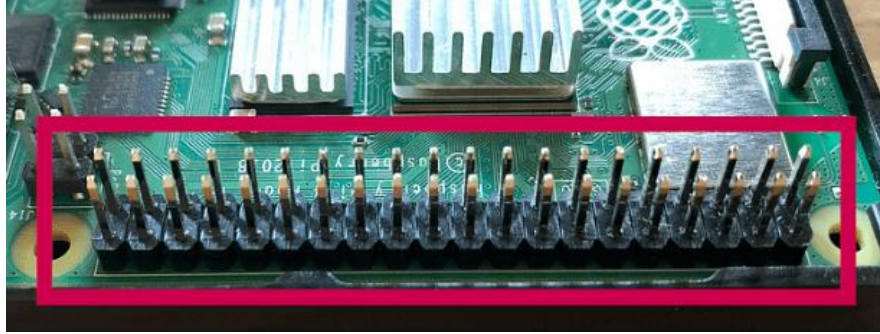
Raspberry Pi'nin Arduino'ya göre en büyük avantajı bağlanabilirliği yani büyük sayıda bağlantı noktasının olmasıdır. Arduino Uno R3'ün sadece bir mini USB bağlantısına sahip olduğunu hatırlayalım. Raspberry Pi'nin dört USB 2.0 ve bir mikro USB bağlantı noktası vardır. Şekil 2.20'de yalnızca iki USB 2.0 bağlantı noktası görünüyor, ancak ikisi üst üste yerleşmiş olup aslında dört adet vardır. Bu bağlantı noktaları büyük sayıda dış cihazı bağlamak için kullanılır: klavyeler, bilgisayar fareleri, dijital kameralar, dış sabit diskler ve diğer bellekler. Mikro USB bağlantı noktası, cep telefonlarına güç verildiği gibi güç için kullanılır. Bu bağlantı noktası aynı zamanda veri aktarımı için bilgisayara bağlanmak için de kullanılabilir.

İnternette arama Wi-Fi dışında ağ bağlantı noktası üzerinden de yapılabilir. Bunun için RJ45 konektörlü kabloya ihtiyacımız var. Kablonun diğer ucu yönlendiriciye bağlanır. Ağ bağlantı noktasının alt kısmında başarılı iletişim için gösterge görevi olan iki LED diyot bulunmaktadır.

HDMI (İng. High Definition Multimedia Interface) adından da anlaşılacağı gibi, bu bağlantı en yüksek kalitede ses ve video iletimini sunar. Onun yardımıyla Raspberry Pi monitöre veya televizyona bağlanır. 3,5 mm AV (İng. Audio Video)

bağlantı noktası öncelikle kulaklıklara veya ses yükselteçine bağlanmak için kullanılır. Video bağlantısı olarak da kullanılabilir ancak bu durumda ayrı bir adaptörün satın alınması gerekir. Geliştirme kartında, özel olarak tasarlanmış Raspberry Pi modüllerini bağlamak için iki **özel konektör** bulunmaktadır. Biri CSI (İng. Camera Serial Interface) kısaltması altında bilinir ve kameranın bağlanması için kullanılır, diğeri ise dokunmatik ekranın (İng. touch screen) bağlanması için DSI (İng. Display Serial Interface) konektörüdür.

Raspberry Pi, **genel amaçlı giriş-çıkış cihazlarına bağlanmak için 40 pinli konektöre sahiptir**. Bu konektörün kısaltması GPIO'dur (İng. General Purpose Input Output). Bu konektörün yirmi altı pini giriş-çıkış cihazlarına bağlanmak içindir, geri kalanları ise topraklama ve 3,3 V ve 5 V arası güç kaynağı içindir. Giriş-çıkış cihazları olarak şunları düşünüyoruz: LED diyotlar, düğmeler, sensörler, motorlar ve diğerleri. Onların işlevini ve anlamını Arduino Uno R3'ün bileşenlerinden bahsederken öğrenmiştik. Tabii ki, bu cihazları bağlamak için devre tahtası en basit çözümdür.



Şekil 2.21. Giriş-çıkış cihazların bağlanması için GPIO pinleri

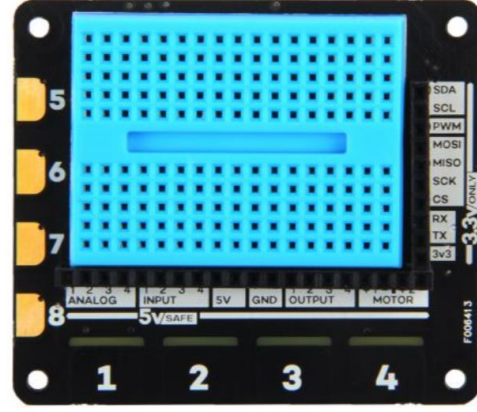
2.7. Raspberry Pi Eklentileri

Raspberry Pi'nin işlevselliğini genişletmek için eklentiler, üstte eklenen donanım (İng. HAT-Hardware attached on top) adı ile bilinmektedir. Bağlantı için çoğu zaman Raspberry Pi üzerindeki 40 genel amaçlı pinin hepsi kullanılır, ancak eklentilerin çoğu pinlerin diğer cihazlar tarafından da kullanılmasına ve hatta birden fazla eklentinin üst üste bağlanmasına izin verir. Eklentilerin çoğu tak ve çalıştır prensibine göre bağlanır (İng. Plug and play), yani otomatik olarak yapılandırılırlar. [3]

Şekil 2.22'de, en popüler eklentilerden biri olan sensörlü ve ekranlı eklenti (İng. Sense hat) gösterilmektedir. İçeriğinde, 8x8 LED matris ekranı, beş düğmeli küçük bir manevra kolu ve altı entegre sensör içerir: jiroskop (açısal hız sensörü), ivmeölçer (doğrusal ivme sensörü), manyetometre (dünyanın manyetik alanını ölçerek, manyetometre coğrafi kuzeyin yönünü belirler), barometre (basınç sensörü), sıcaklık sensörü ve bağıl nem sensörü.



Şekil 2.22. Sensörlü ve ekranlı Raspberry Pi eklentisi



Şekil 2.23. Raspberry Pi'nin araştırma eklentisi

Şekil 2.23'te gösterilen araştırma eklentisi, öncelikle doğru akım motorları kontrol etme olanağı nedeniyle robotikte kullanılır. Bu eklenti aynı zamanda Arduino mikrodenetleyici platformlarıyla da uyumludur. Araştırma eklentisi, maksimum 5V gerilim değerine sahip dört giriş ve çıkış, dört kapasitif dokunmatik düğme, bağlamak için timsah klipsli dört düğme, dört analog giriş, doğru akım motorların dönüşünü kontrol etmek için iki H-köprü ve üstten yerleştirilen mini bir devre tahtası içerir.

Raspberry Pi, Ölçüm ve veri toplama (İng.DAQ MCC 128 – Data Acquisition Measurement Computing) eklentisi sayesinde ölçüm cihazı olarak da kullanılabilir.



Şekil 2.24. Ölçüm ve veri toplama eklentisi

Bu elektronik kart gerilimi ölçmek için sekiz analog giriş içerir. Giriş gerilimi ölçmek için iki yol vardır. Tek uçlu ölçümde (İng. Single ended) giriş gerilimi sıfır potansiyel noktası olan topraklamaya göre ölçülür. Diferansiyel ölçüm, iki giriş gerilim arasındaki farkı belirtir. Saniyede 100 bin örnek alınıyor. Analog dijital dönüştürücü 16 bit çözünürlüğe sahiptir.

Sekiz gerilim ölçüm eklentisi üst üste dizilebilir ve bu durumda saniyede maksimum 320.000 örnek bant genişliğine sahip 64 veri kanalı elde ediliyor.

2.8. Modüler Birimi İçin Uygulamalı Alıştırmalar: Mikrobilgisayarların Bileşenleri

2.8.1. İşyerinde Koruma ve Güvenlik Önlemleri

Öğrenciler uygulamalı alıştırmaları ders öğretmeninin gözetiminde, en fazla 2-3 öğrenciden oluşan gruplar halinde bağımsız olarak gerçekleştirirler. Alıştırmaların verimli ve güvenli bir şekilde gerçekleştirilmesi ve elektronik ekipmanların doğru şekilde kullanılması için öğrencilerin aşağıdaki kurallara uymaları gerekmektedir.

- Elektronik bileşenlerin bağlanması sadece gerilimsiz durumda gerçekleştirilir.
- Elektrikli ekipmanlara nemli veya ıslak ellerle dokunulmaz.
- Bir elle bir iletkeni, aleti veya başka bir elemanı gerilim altında, diğer eliyle başka bir iletkeni aynı anda tutmak yasaktır.
- Çalışma aletlerinin yalıtımlı sapları olmalıdır.
- Elektrik akımından kaynaklanan yangın suyla söndürülmez.
- Akım devresi kesildikten sonra elektrolitik kapasitörlerin uçlarında kısa devre yapılarak boşaltılmalıdır.
- Gerilim altında çalışırken iletkenler, rezistörler, piller ısınmalarından dolayı dokunulmamalıdır.
- Elektronik atıklar özel olarak belirlenmiş yerde toplanmalıdır.
- Her uygulamalı alıştırma başlamadan önce öğrenci, alıştırmının gerçekleştirilmesine ilişkin gereklilikleri dikkatlice okumalı ve alıştırmada kullanılan elektronik bileşenlerin çalışma şeklini tekrarlamalıdır.
- Elektronik bileşenler dikkatli, yavaş ve kuvvet uygulanmadan bağlanmalıdır.
- Elektronik bileşenleri bağlamak için kullanılan teller çok fazla iç içe geçirilmemeli ve sıkılmamalıdır.
- Elektronik bileşenleri elektrik devresine bağladıktan sonra, öğretmeni arayarak kontrol yapılmalı ve onayının ardından gerekli gerilimler açılır.
- Bileşenlerde veya ekipmanda tüm arızalar, hasarlar veya eksiklikler derhal ders öğretmenine bildirilmelidir.

2.8.2. Kişisel Bilgisayar Bileşenlerin Kurulumuna İlişkin Uygulamalı Alıştırma

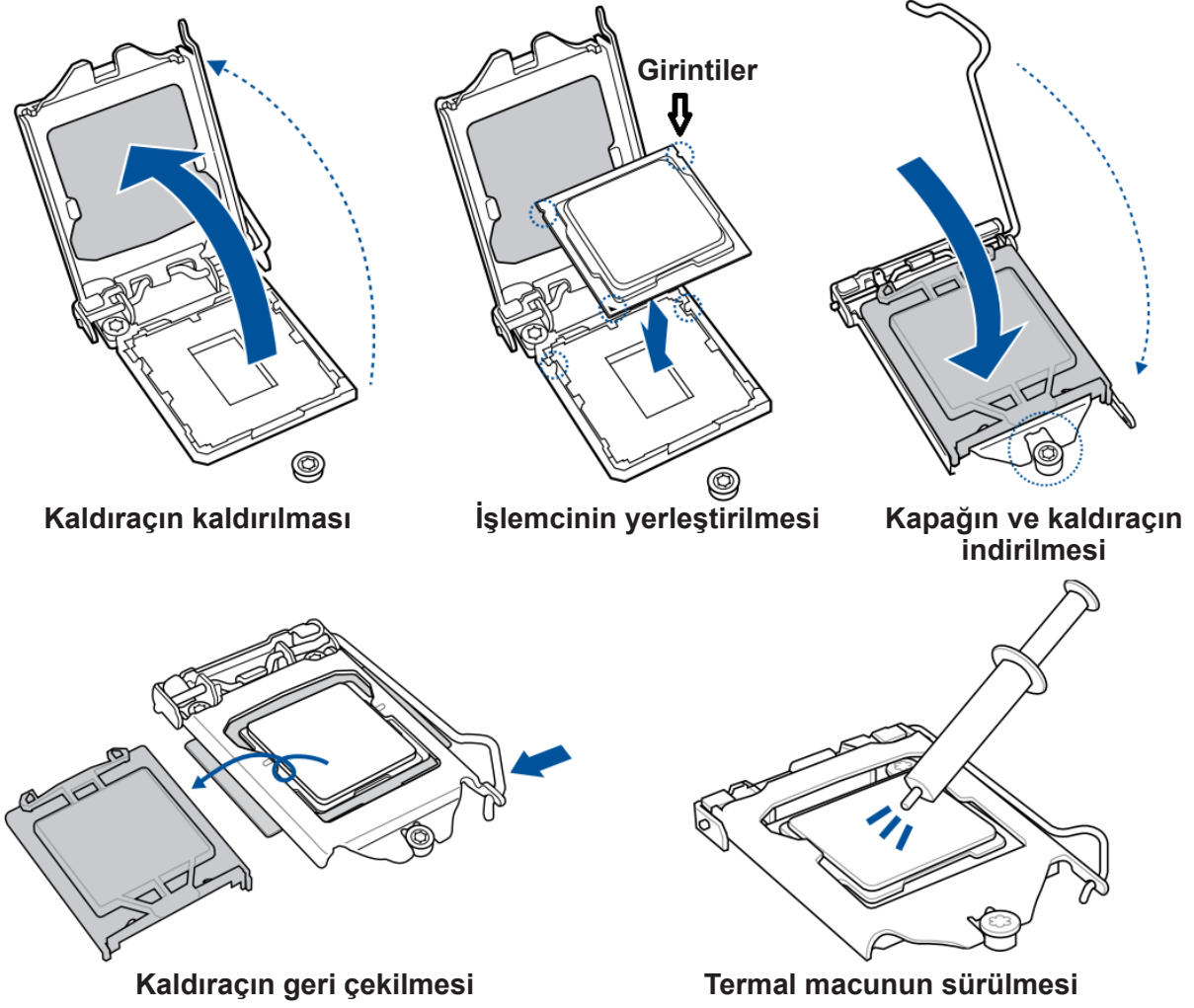
2.8.2.1. Kişisel Bilgisayarın Donanım Bileşenleriyle Çalışırken Koruma ve Güvenlik Önlemleri

- Elektrik çarpmasını önlemek için kurulumu başlamadan önce bilgisayarın güç kaynağı kapatılsın.
- Elektrik gücünü açmadan önce tüm kabloların ve konektörlerin doğruluğu kontrol edilsin.
- Üreticiden uygun sertifikaya sahip güç kaynağı cihazı kullanılsın.
- Parametrelerde ve işaretlemelerde belirli farklılıklar olabileceğinden dolayı, kullanılan anakart modeline ait teknik-teknolojik belgeler zorunlu olarak okunmalıdır.
- Elektronik çipler statik elektriğe duyarlı olduğundan dolayı, kasa gibi bir tür metal yüzeye dokunmak gerekir. Bu şekilde insan vücudu boşaltılır.
- Kısa devreyi önlemek için konektörlere ve yuvalara vida veya tornavida gibi metal nesnelere dokunmayın.
- Donanım bileşenlerinin kurulumu oynamayan ve sabit bir çalışma yüzeyinde gerçekleştirilmelidir.
- Anakartın temas noktalarına zararlı etkisi olduğundan ve soğutma ve havalandırma azalttığından dolayı, aşırı ısınma meydana gelmemesi için bilgisayar nem ve kirden korunmalıdır.
- Parçalarının zarar görmesini önlemek için kullanıcı bilgisayarı düzgün bir şekilde kapatmalıdır.

2.8.2.2. İşlemcinin Kurulumu

Yeni bir anakart ise işlemcinin yuvası (soketi) sarı koruyucu folyo ile kaplanmalı olmalıdır. Yuvanın sağ tarafında bulunan **metal kaldıraç (kol)** aşağı itilir, sağa çekilir ve yukarı kaldırılır ve bu arada yuvanın kapağı da kaldırılıyor. Sarı koruyucu folyo çıkartılır ve işlemcinin çıkarılması durumunda saklanmalıdır. İşlemci yandan tutulmalı ve **altın pinlere dokunulmamalıdır**. İşlemciyi, pinler aşağı bakacak

şekilde yuvaya yerleştiriyoruz ve işlemcinin iki yan girintisinin yuvadaki çıkıntılarla çakıştığına dikkat ediyoruz. [8]



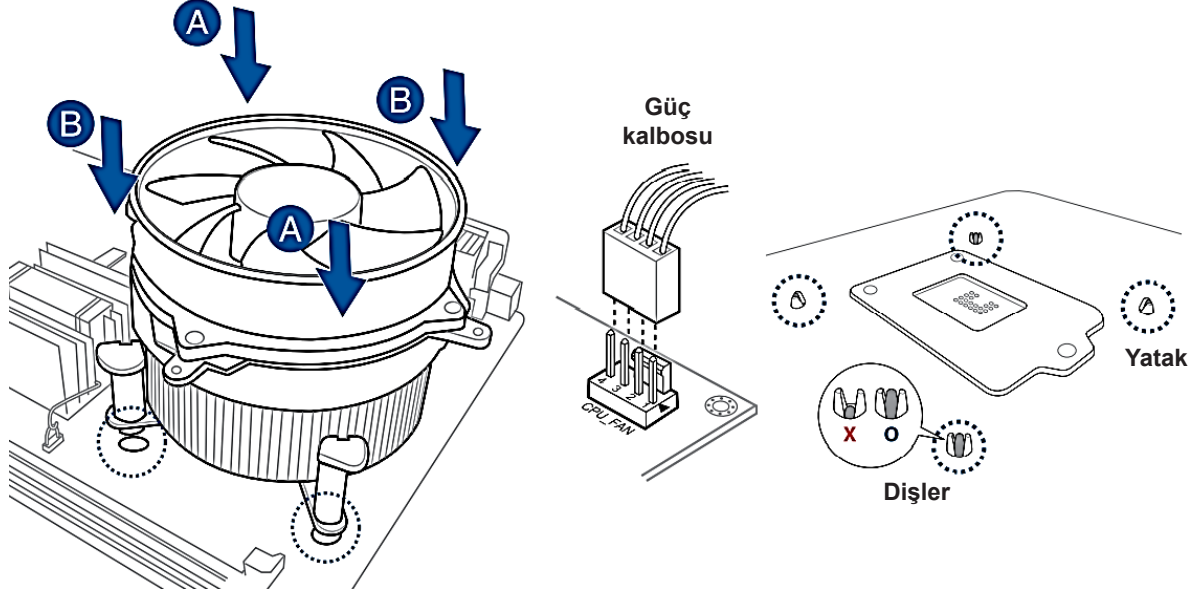
Şekil 2.25. İşlemci kurulum adımları

İşlemci bastırılmamalıdır. Kapak civata somununa indirilir, metal kol indirilir ve yerine yerleşmesi için sola doğru çekilir.

2.8.2.3. Soğutucunun Kurulumu

Soğutucunun kurulumunu başlatmadan önce işlemcinin üst kısmına **termal macunun** sürülmesine gerek olup olmadığı kontrol edilmelidir. Termal macun sıcaklığı 20° ile 30° arasında azaltır. Yeni macunu uygulamadan önce eskisi çıkarılmalı ve bunun için %90 alkol ve mikrofiber bez kullanılması en doğrudur. Öncelikle işlemcinin yuvadan çıkarılması gerekiyor. Beze yaklaşık posta pulu büyüklüğünde az miktarda alkol uygulanır ve ardından işlemcinin üst kısmı hafifçe silinir. Macun bir bezelye tanesinin yarısı kadar ortada sürülür. Macun, işlemcinin sıcaklığı ve soğutucunun ağırlığı etkisi altında bulaşacaktır. Soğutucuyu işlemcinin üzerine yerleştir-

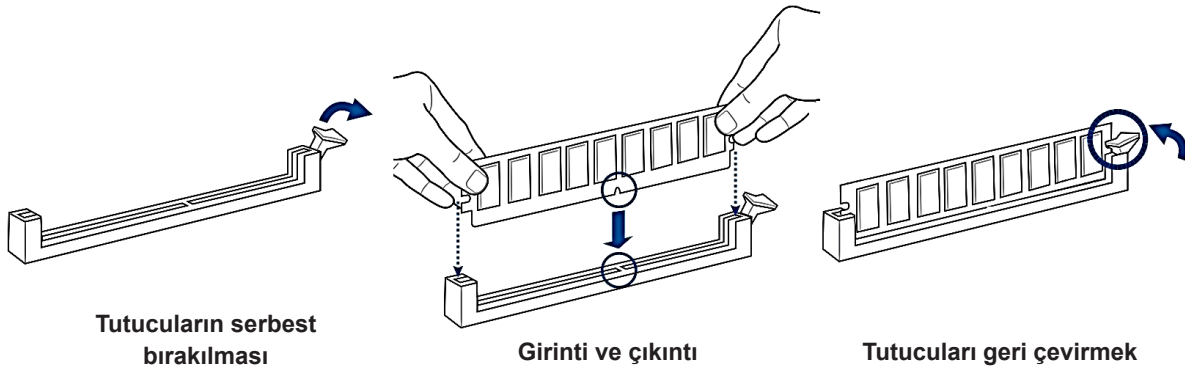
meden önce **plastik dişlerin** başlarındaki olukların soğutucunun merkezine doğru yönlendirildiğini kontrol ediyoruz. Çapraz olarak yerleştirilmiş iki diş seçip bastırıyoruz. Dört dişe de bastırdıktan sonra anakartın arkasından dişlerin yataklarına göre hizalı olup olmadığını kontrol ediyoruz.



Şekil 2.26. İşlemci soğutucu kurulum adımları

2.8.2.4. RAM Belleğin Kurulumu

İşlemcide olduğu gibi RAM belleğinde de statik elektriğe karşı dikkatli olmamız gerekiyor. Bu nedenle kurulumdan önce en azından metal kasaya dokunmak en iyisidir. Ayrıca RAM modülleri yandan tutulmalı ve altın pinler hiçbir şekilde dokunulmamalıdır.



Şekil 2.27. RAM belleği kurulum adımları

Pinler arasında **küçük bir girinti** vardır ve bu girinti, DIMM yuvalarındaki (RAM yatağı) çıkıntıyla aynı hizada olmalıdır. Bellek modülünü takmadan önce boş yuvanın **yan tutucularını** serbest bırakmamız gerekiyor.

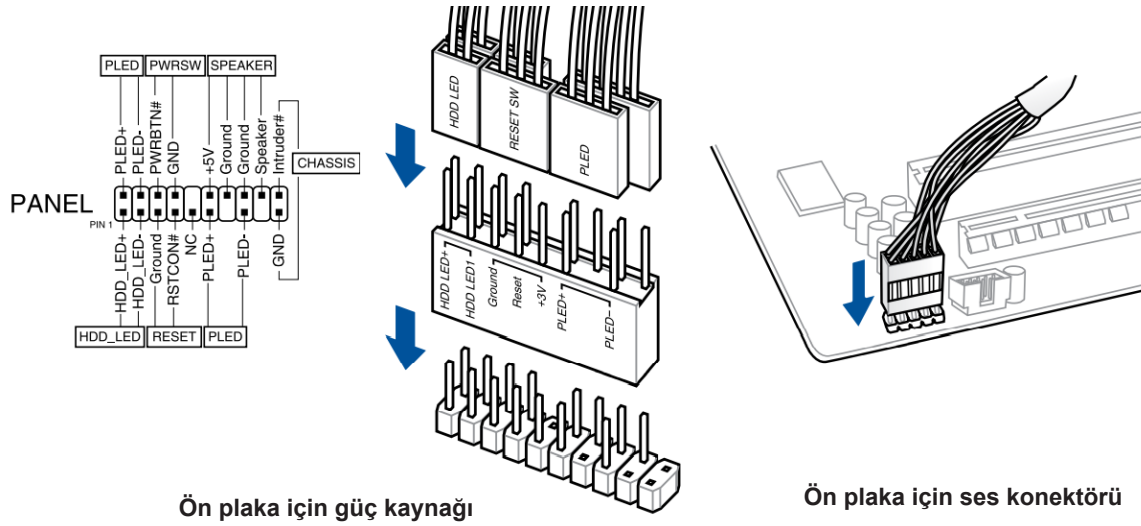
2.8.2.5. Anakartı Kasaya Monte Etme

Anakartı kasaya monte etmeden önce işlemcinin, soğutucunun ve RAM belleğinin kurulması tavsiye edilir. Anakartlar, kasanın arkasında bağlantılar için boşlukların bulunduğu **metal plakayla** birlikte gelir. Metal plakayı yerleştirirken boşlukların anakartın arkasındaki bağlantı noktalarına denk gelmesine dikkat edilmelidir. Ardından, kasadaki **vidalar** bulunmalıdır ve anakart, anakarttaki vida delikleri kasanın vidalarıyla aynı hizada olacak şekilde yerleştirilmelidir.



Şekil 2.28. Anakartı monte etme adımları

Vidaları vidalandıktan sonra, **güç bağlantılarını** ön plakaya ve soğutucunun güç bağlantısını kasaya bağlamak gerekir.



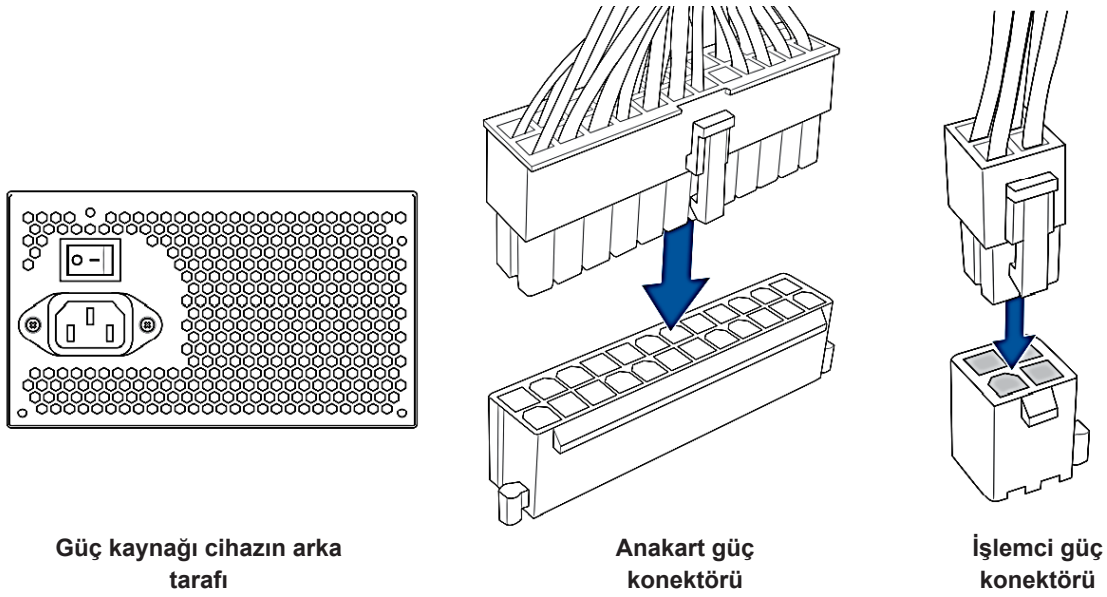
Şekil 2.29. Anakartın güç kaynağına bağlanması

Kasanın ön plakasında güç düğmesi, sıfırlama düğmesi, ön USB bağlantı noktaları, ön ses konektörleri ve güç ile sabit disk göstergeleri olarak LED diyotlar bulunur. Onların güç bağlantıları, **F (Front)** harfi ile işaretlenmiş anakartın yatağıyla

bağlanmalıdır. USB ve ses konektörleri, pinlerinin benzersiz düzeni nedeniyle en basit olarak bağlanır. Güç ve sıfırlama düğmelerinde, kutupluluk önemli değildir. LED diyotlarda işaretlere dikkat edilmelidir. Genellikle, siyah ve beyaz tel artıdır, diğer renklerdeki teller ise eksi veya topraklamadır. Son olarak, anakartta **SYS_FAN** ve **PWR_FAN** işaretli yuvaları bulunmalıdır ve onlarda kasa suğutucusunun güç kaynağı bağlantısı yerleşmelidir.

2.8.2.6. Güç Kaynağı Cihazının Kurulumu

Güç kaynağı cihazı kasanın arka tarafında üst veya alt kısma yerleştirilebilir. Cihazı soğutucusu anakarta bakacak şekilde iç bölmeye yerleştirip arka kısmına kadar itiyoruz ve vidalarla sabitliyoruz.

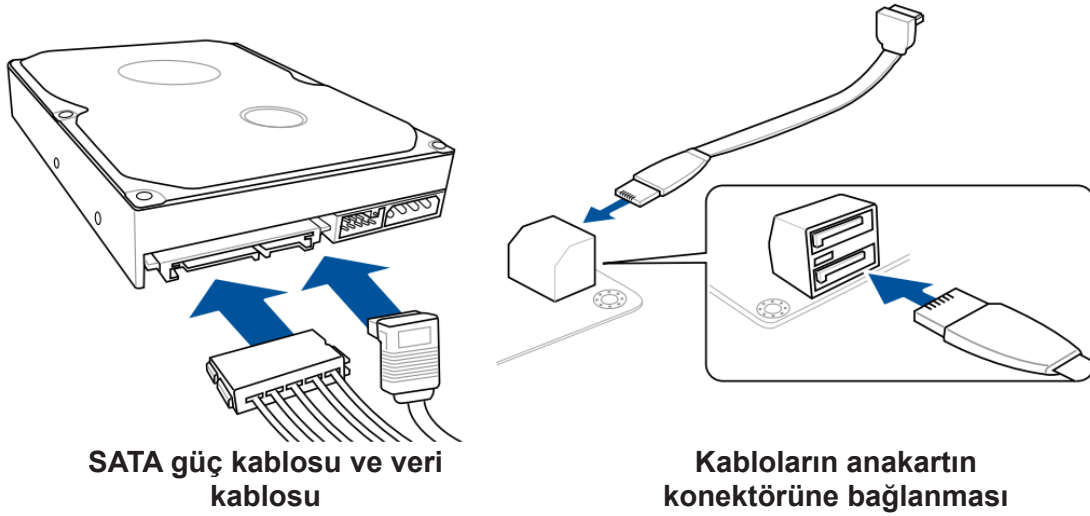


Şekil 2.30. Güç kaynağı cihazının kurulumu

Güç kaynağı cihazında anakarta veya belirli bir bileşene bağlanması için gereken birden fazla kablo bulunur. Tüm güç kablolarının benzersiz tasarıma sahip olması iyidir, böylece onları bağlarken hata yapılamaz. Anakarta iki konektör bağlanır: anakarta güç veren **20+4 pinli ATX kablosu** ve işlemciye güç sağlayan **4 pinli ATX 12V**. Sabit diske ve DVD sürücüsüne güç sağlamak için iki tür kablo kullanılır: **SATA ve 4 pinli IDE** ve devamda **bunları kısaca açıklayacağız**. Grafik kartının altı pinli güç kablosu VGA1 işareti ile belirtilmiştir ve PCI Express kablosu adıyla bilinir.

2.8.2.7. Sabit Disk ve SSD Cihazın Kurulumu

Kasanın içinde 9 cm yüksekliğinde ayrı bir bölme bulunmaktadır. Burada sabit disk, L harfi şeklinde olan konektörleri öne bakacak şekilde takılır. Daha büyük konektöre güç kaynağı cihazından gelen **güç kablosu**, daha küçük konektöre ise sabit diskin beraberinde gelen **SATA veri kablosu** takılır. SATA veri kablosunun diğer ucu anakartın SATA2 ile işaretlenmiş L harfi şeklindeki mavi konektöre takılır.



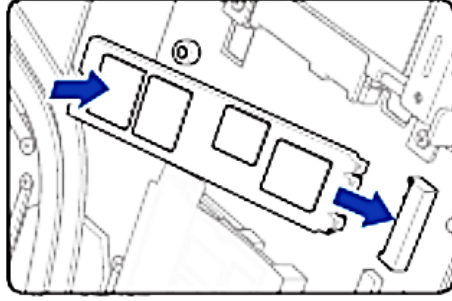
Şekil 2.31. Sabit diskin SATA kablolarıyla bağlanması

SATA SSD bellek cihazları sabit disklerle aynı boyut ve pin düzenine sahip olduğundan, bağlanırken güç veya veri aktarımı için aynı SATA konektörleri kullanılır. Tabii ki, ilk önce kasada SSD ve sabit disk için iki bölüm olup olmadığı kontrol edilmelidir. Genellikle, işletim sistemi SSD belleğine kurulur, sabit disk ise kullanıcı belgelerini ve uygulamaları depolamak için kullanılır. Bu şekilde işletim sisteminin açılması için gereken süre kısalıyor ve kişisel bilgisayarın çalışma hızı artıyor.

2.8.2.8. M.2 SSD Modülünün Kurulumu

M.2 SSD modülünün kurulumu sırasında bağlantı tipini, yani PCIe veya SATA olup olmadığını kontrol etmemiz gerekir. Modül ve anakart konektörü aynı bağlantıya sahip değilse fiziksel olarak bağlanamazlar (Şekil 2.9). M.2 SSD modülünün pinleri küçük bir açı altında uygun konektöre dikkatlice bastırılır. Modül anakart üzerine yerleştirildikten sonra vida yardımıyla sabitlenir. M.2 konektörünün yanında üç vida

yuvası bulunduğunu belirtelim, çünkü M.2 SSD modülleri üç farklı uzunlukta (22, 60 veya 80 mm) olabilir, genişlikleri ise her zaman aynıdır.



M.2 SSD modülün konektörüne
bastrılması

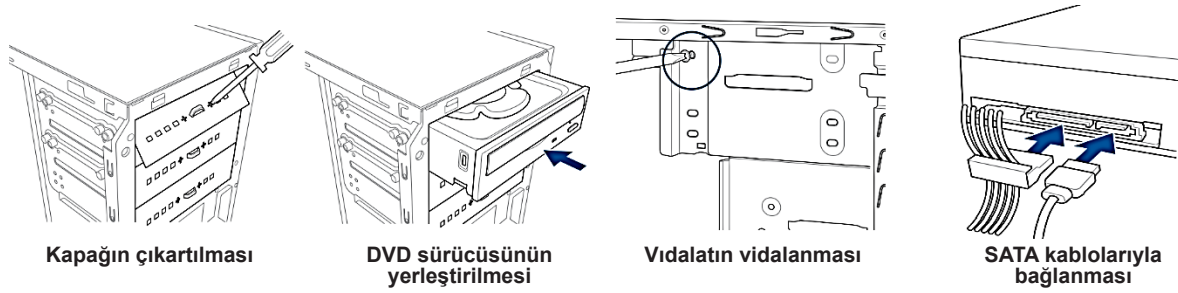


M.2 SSD modülünün vida
yardımıyla sabitlendirilmesi

Şekil 2.32 M.2 SSD modülünün kurulumu

2.8.2.9. DVD Sürücüsünün Kurulumu

DVD sürücüsü kasanın ön tarafına monte edilir. Bunun için anakartın boşluklarına yerleştirilen iki dişle sabitlenen kasanın ön **maskesini** çıkarmak gerekiyor. Ardından 14 cm yüksekliğindeki bölmenin **kapağı çıkartılmalıdır**. DVD sürücüsü bölmeye yerleştiriliyor ve yandan vidalarla sabitlenir.

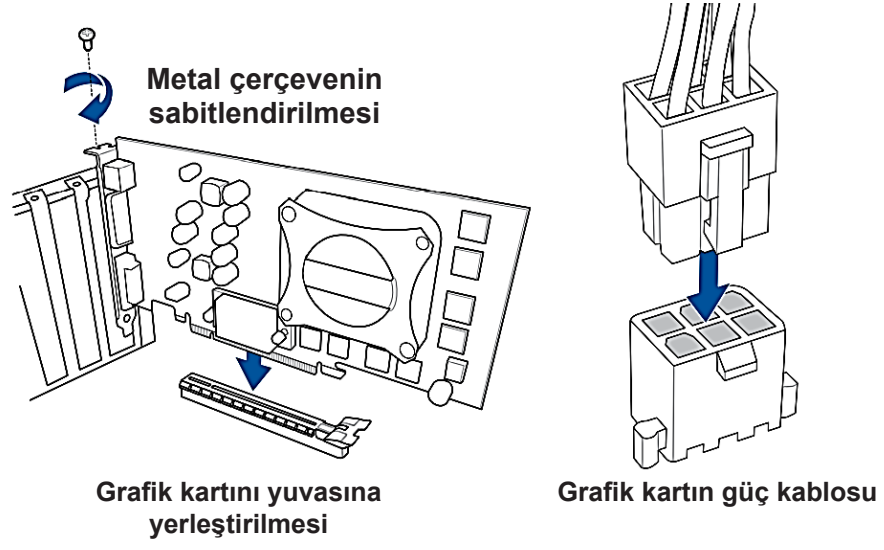


Şekil 2.33. DVD sürücüsü kurulumu

Sabit diskte olduğu gibi, DVD sürücüsünde de biri güç, diğeri veri için olmak üzere iki L şeklinde SATA konektörü bulunur.

2.8.2.10. Grafik Kartın Kurulumu

Grafik kartı anakartın **PCI Expressx16** yuvasında yerleştirilir. Monte etmeye başlamadan önce kasanın arkasındaki metal çerçeveleri vidaları sökerek çıkarmak gerekir. Grafik kartını PCI Expressx16 yuvasına taktıktan sonra grafik kartının **metal çerçevesini** kasanın arka kısmına vidalar yardımıyla sabitlemek gerekiyor. Son olarak, VGA1 işaretiyle işaretlenmiş altı pinli güç kablosunu takıyoruz.



Şekil 2.34. Grafik kartın PCI Expressx16 yuvasına yerleştirilmesi

2.8.3. Elektronik Bileşenlerin Arduino Uno R3 ile Bağlanmasına İlişkin Uygulamalı Alıştırmalar

2.8.3.1. Arduino Tabanlı Mikrodenetleyici Platformu İle Çalışırken Koruma ve Güvenlik Önlemleri

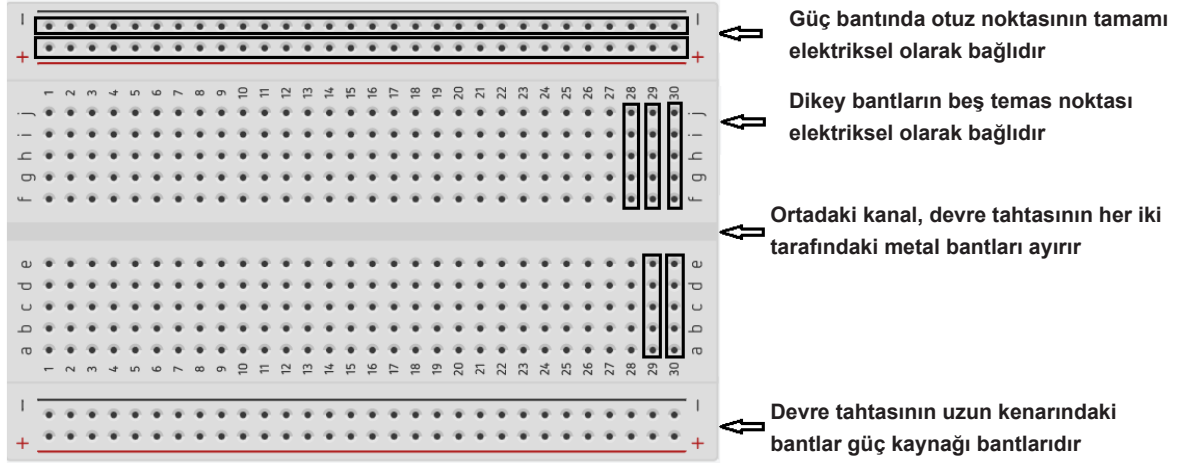
Uygulamalı alıştırmalar, alçak gerilim ve düşük akım sinyalleriyle çalışan Arduino Uno R3 platformu kullanılarak gerçekleştirilir. Bunları hasardan korumak için bazı önlemlerin alınması gerekir.

- Herhangi bir montaja başlamadan önce Arduino Uno R3 platformunun güç kaynağıyla bağlantısı kapatılır.
- Pil, güç kaynağı olarak Arduino Uno R3'e Vin ve GND pinleri aracılığıyla bağlanır. İzin verilen maksimum gerilim değeri 20 V'tur ancak 9 ila 13 V gerilimli pillerin kullanılması tavsiye edilir. Bağlarken gerilimin kutupluluğuna dikkat edilsin.
- Arduino Uno R3'ün giriş-çıkış pinleri için maksimum gerilim 5,5V'tur.
- Arduino Uno R3'ün bir pini için maksimum giriş akımı 40mA'dir. Tüm pinlerin giriş akımlarının toplamı 200 mA'yı geçmemelidir.
- Pinlerin kısa devresine izin verilmez.
- Kutuplanmış bileşenlerin bağlantıları tanımlanmalı ve uygun şekilde bağlanmalıdır.
- Bağlantıyı başlatmadan önce, elektrikli bileşenlerden geçen akımın şiddeti hesaplanmalı ve o bileşenin teknik-teknolojik belgelerine göre izin verilen maksimum akıma eşit veya daha düşük olup olmadığı kontrol edilsin.
- Motorları veya diğer endüktif tüketicileri bağlamak için uygun kontrol cihazları kullanılmalı (İng. motor driver)
- LED diyotlardan geçen akımı sınırlamak için rezistörler kullanılmalıdır.

2.8.3.2. Devre Tahtası Kullanım Kılavuzu

Lehimleme, baskılı devre kartı üzerinde kalıcı tasarıma, kararlı çalışmaya ve güvenilir elektrik kontaklarına sahip elektronik cihazlar yapmak için bir prosedürdür.

Ancak çoğu zaman, cihazın işlevselliğini ve kalitesini artırmak amacıyla elektrik devresinin tasarımı değiştiriliyor. Bu, bileşenlerin bağlanma şeklini değiştirmek, hatta onları değiştirmek anlamına gelir. Devre tahtaları elektrik devrelerinin tasarımını hızlı ve basit bir şekilde değiştirmek için idealdir. [6]

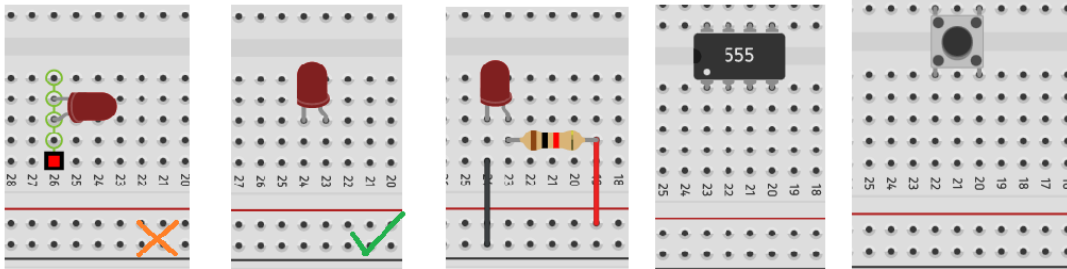


Şekil 2.35. Devre tahtasının dış görünümü

Devre tahtası, üzerinde aslında temas noktaları veya bağlantı noktaları olan boşlukları olan, yüksek kaliteli plastikten yapılmış bir plakadır. Elektronik elemanların bağlantı uçları veya bunları bağlamak için kullanılan teller temas noktalarına yerleştirilir. Şekil 2.35'te, devre tahtasının dış görünümü gösterilmiştir. Plastik maskenin altında, elektronik elemanların elektrik bağlantısını gerçekleştiren, birkaç grup halinde paralel olarak yerleştirilmiş iletkenler bulunmaktadır. İletkenlerin üzerinde kısaç şeklinde kontaklar bulunur ve elektronik elemanlar mekanik olarak takılır, bağlantı uçları hafifçe basılarak devre kartına takılır.

Elektronik elemanlar plakanın ortasındaki dikey bantlarda yerleştirilir. Plakanın uzun kenarları boyunca her iki tarafta iki çift bant bulunmaktadır (Şekil 2.35'teki yatay bantlar). + veya kırmızı renkle işaretlenen bant güç kaynağına, - veya siyah renkle işaretlenen bant ise topraklamaya bağlanır.

Elemanların doğru şekilde bağlanması için metal bantların sıralamasını bilmek çok önemlidir. Aynı bant üzerinde yatan tüm temas noktaları birbirine elektronik olarak bağlıdır ve elemanları yerleştirirken bağlantı uçlarında kısa devre yapmamaya dikkat edilmelidir, çünkü o zaman onlardan akım akmayacaktır.



Şekil 2.36. Elektrik bileşenlerin devre tahtasında bağlanması

Örneğin Şekil 2.36'da ilk LED diyotu yanlış yerleştirilmiştir çünkü elektrotları, anot ve katot, aynı banta ait temas noktalarına yerleştirilmiştir. Şekil 2.36'daki ikinci diyot. LED diyotun devre tahtasına doğru şekilde bağlantısını temsil eder. İki elemanı bağlamak istiyorsak, örneğin LED diyot ve rezistör, bunların bağlantı uçları aynı banta ait temas noktalarına yerleştirilmelidir.

Devre tahtanın ortasında dikey iletkenleri elektriksel olarak ikiye ayıran bir kanal yani girinti vardır. Bu, entegre devrelerin iki bağlantı hattıyla (İng. DIP-Dual in Line) bağlanmasına izin verir. Entegre devre, pinleri kanalın farklı taraflarına yerleşecek ve elektriksel olarak bağlı olmayacak şekilde kanalın tam üstüne yerleştirilir. Devre tahtanın temas noktaları arasındaki mesafe, entegre devrelerin pinleri arasındaki olduğu mesafe kadar 2,45 mm'dir. Merkezi kanala düğmeler de yerleştirilir. Düğmelerin iki karşıt tarafta iki bağlantı olmak üzere toplam dört bağlantısı vardır. **Aynı taraftaki bağlantılar elektriksel olarak bağlı değildir** ve onlar iki dikey iletkene ait temas noktalarına yerleştirilmelidir, böylece düğmeye basıldığında elektrik devresi kapanır.

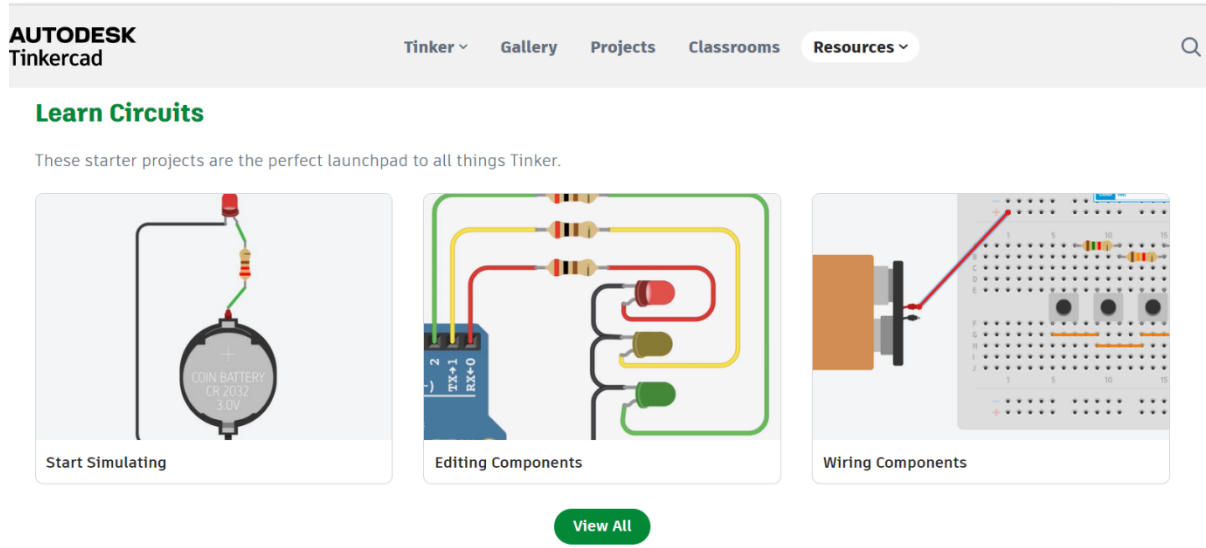
2.8.3.3. Arduino Uno R3 İçin Bilgisayar Simülasyonu

Bilgisayar simülatörleri, elektronik ve bilgisayar tekniği alanında öğrenme ve çalışmada sıklıkla kullanılan bir araç haline geldi. Simülatörler merakı, yaratıcılığı, sezgisel düşünmeyi teşvik eder ve çalışmak için daha güvenlidir.

Öngörülen uygulamalı alıştırımların uygulamalı öğretim kabinesinde gerçekleştirilmesi gerekir ve bunun için de uygun ekipman, donanım ve elektronik elemanlar gerekir. Bazı alıştırımlar tamamen bilgisayar simülasyonu uygulanarak gerçekleştirilebilir. Tinkercad, Arduino Uno R3 için mükemmel bilgisayar simülasyonu sunan ücretsiz web platformudur.

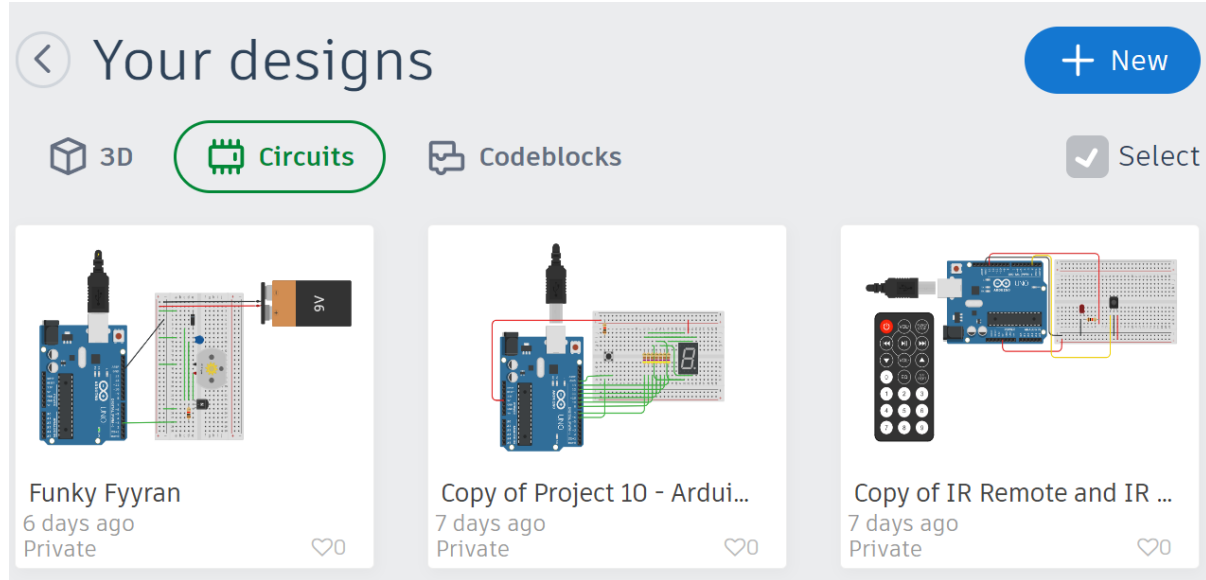
Kullanıcının Google veya Apple kullanıcı hesabına sahip olması gerekir ve bu hesaptan www.Tinkercad.com platformunun resmi web sitesine giriş yapması gerekir. Başarılı **giriş** yaptıktan sonra ana menünün bulunduğu ana sayfa açılır. Olanaklar çok büyüktür. Tinkercad platformunun 3D tasarımla çalışma, Codeblocks programlama dilinde programlama kodlarının oluşturulması ve elektrik devrelerinin tasarlanması olanağı sunduğunu belirtelim. Bizim, Arduino Uno R3 ve C/C++ programlama dilini kullanarak elektrik devrelerinin simülasyonuna geçimiz vardır.

Resources → Learning Center → Learn Circuit (Türkçe Kaynaklar → Öğrenim Merkezi→Elektrik Devrelerini Öğrenmek) seçeneklerine tıklanıldığında bileşenlerin seçilmesi, bağlanması, program yazılması ve simüle edilmesiyle ilgili talimatlar açılır. Çalışma talimatları adım adım yazılmıştır ve etkileşimlidir, yani hemen uygulanabilir, çünkü talimatların açılmasıyla çalışma alanı ve bileşenlerin seçilmesi için menü açılır.



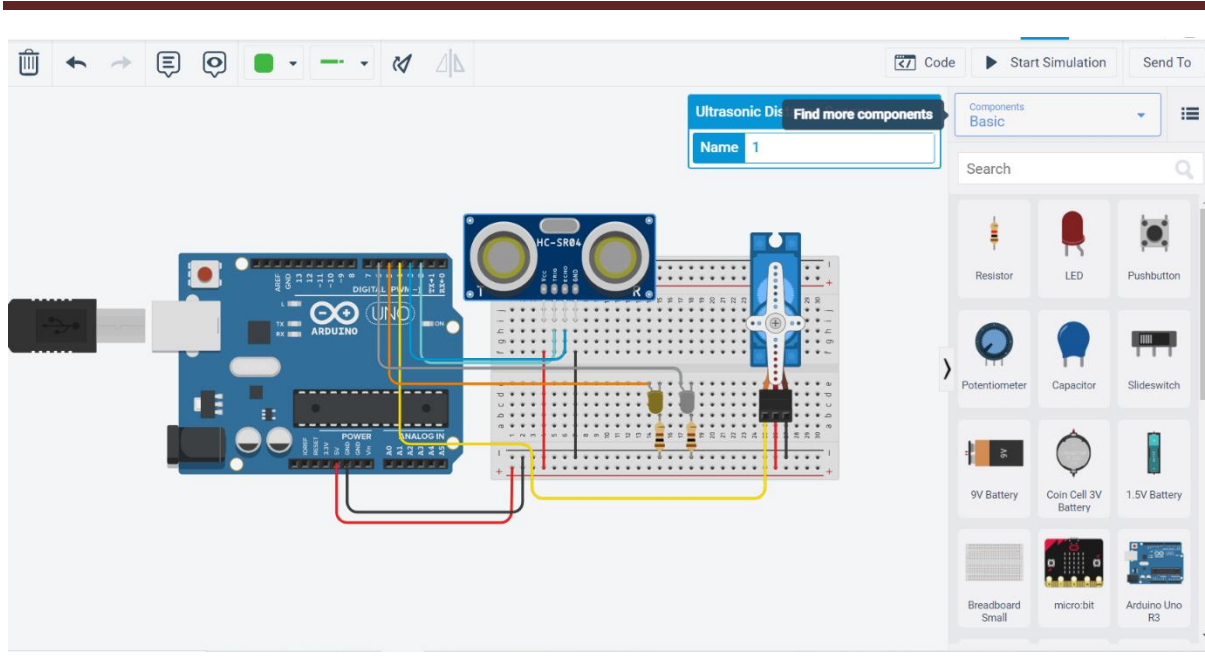
Şekil 2.37. Arduino Uno R3 kullanarak elektrik devreleri oluşturma talimatlarının açılması

Profil resmine basıp (tıklayarak) ve Tasarımlarım (İng. My Designs) seçeneğini seçerek daha önce üzerinde çalıştığımız tüm projeler açılıyor. Onların hepsi Web platformunun kendisi tarafından kaydedilir. Yeni bir elektrik devresi oluşturmak istiyorsak Yeni (İng.New) seçeneğini seçiyoruz.



Şekil 2.38. Yeni ve eski elektrik devre tasarımlarının açılması

Şekil 2.39'da elektrik devrelerinin oluşturulması ve Arduino Uno R3'ün programlanması için çalışma alanının (masaüstünün) görünümü gösterilmiştir. Elektrik devre, elektronik bileşenlerin çekilip bırakılmasıyla çok kolay bir şekilde monte edilir ve ardından bağlanır. **Code** düğmesine basıp **Text** seçeneğini seçersek çalışma programını görebiliriz veya yazabiliriz. Elektrik devreyi kurduktan ve programı yazdıktan sonra **Start Simulation** düğmesine basıyoruz ve devrenin öngörüldüğü gibi çalışıp çalışmadığını gözlemliyoruz. Kodda hatalar varsa, onlar kırmızı renkle işaretlenecektir.



Şekil 2.39. Tinkercad web platformunda elektrik devreleriyle çalışmak için çalışma alanının (masaüstünün) görünümü

Tinkercad web platformu açık kaynaktır ve projelerin paylaşılmasına olanak sağlar. Projenin kendisindeki Gönder (İng. Send to) seçeneğine tıkladığımızda yeni bir pencere açılıyor, İnsanları davet et (İng. Invite people) seçeneğini seçiyoruz, paylaşım bağlantısını kopyalıyoruz (İng. Copy) ve ondan sonra yapıştırıyoruz (İng. Paste) ve projeyi e-posta veya bazı sosyal ağlar aracılığıyla istenilen kişiye gönderiyoruz. Aslında internette arama yaparak Tinkercad’de yapılmış birçok hazır proje bulunabilir ve bunları Açık Düşün (İng. Tinker this) seçeneğini seçerek web tarayıcısı seçilen projeyi doğrudan Tinkercad platformunda açar. Bu, projelerimizi geliştirmemize çok yardımcı olabilir.

2.8.3.4. Uygulamalı Alıştırma: Anahtarlar veya Arduino Uno R3 Kullanarak VE Mantık Devresinin Gerçekleştirilmesi

1. Alıştırmanın amacı elektronik bileşenleri bağlayıp VE mantık fonksiyonunun aşağıdaki verilmiş iki şekilde gerçekleştirilmesi için elektrik devresi elde etmektir.

- Seri bağlı düğmeler (anahtarlar) kullanarak
- Arduino Uno R3 kullanarak


Bu şekilde mantıksal fonksiyonu anlamak daha kolay olacaktır ve iki yapı arasında karşılaştırma yapılacaktır. Yazılımı daha sonra tanıyacağımız için şimdi sadece elektrik devresini kurup anlatacağız ve dördüncü modül birimde Arduino platformunu programlayacağız. [9]

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- arduino uno r3
- devre tahtası
- baskılı devre kartı için iki düğme
- led diyot
- üç rezistör ($r_3=220\Omega$, $r_1=r_2=1k\Omega$)
- bağlama kabloları (jumper kablolar)

3. Alıştırma için hazırlık

Alıştırmanın gerçekleştirilmesine başlamadan önce, öğrenci devre tahtasının kullanım talimatını okumalı, Arduino Uno R3'ün pin diyagramını ve elektronik elemanlar olarak düğme ve LED diyotun özelliklerini hatırlamalıdır.

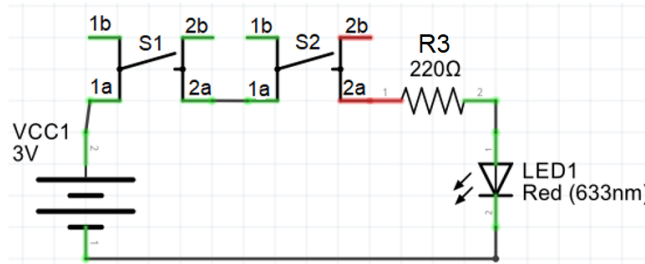
Mantık sembolü		
A		
B		
Doğruluk tablosu		
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Şekil 2.40. VE mantık devresi

Mantık devreleri dijital sinyalleri işlemek için kullanılır. Şekil 2.40'ta, VE mantık devresinin sembolü ve doğruluk tablosu verilmiştir. VE devresinin çıkışında yüksek seviye yani mantıksal bir elde etmek için tüm girişlerin yüksek seviyede olması gerekir. Çıkışta düşük seviye yani mantıksal sıfır elde etmek için girişlerden en az birinin düşük seviyede olması yeterlidir.

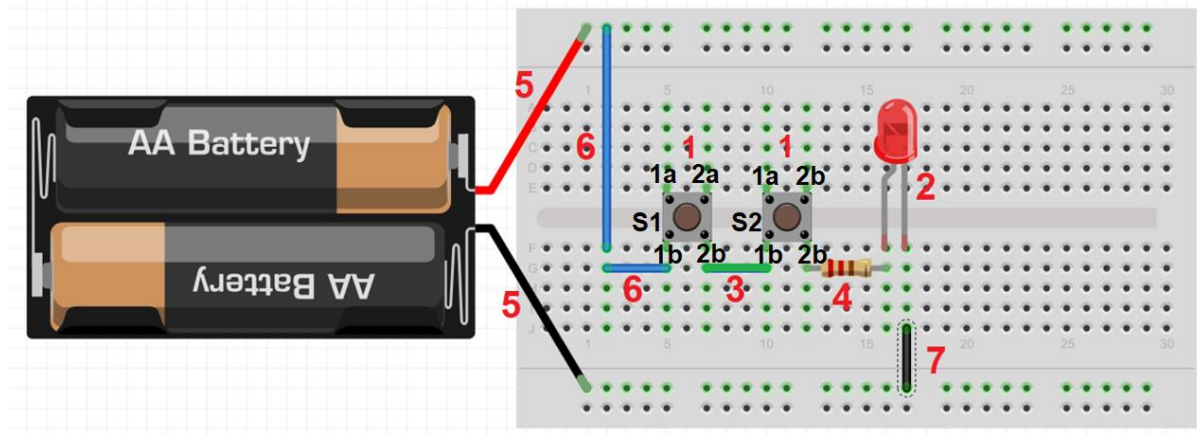
4. VE mantık fonksiyonunun düğmeler kullanılarak gerçekleştirilmesi

Şekil 2.41'de, VE mantık fonksiyonunun seri bağlı iki düğme ve bir LED diyot kullanılarak gerçekleştirilmesi için işlevsel şema gösterilmiştir.



Şekil 2.41. Seri olarak bağlı iki düğmenin işlevsel şeması

Düğmeler VE mantık devresinin girişlerini, LED diyotu ise çıkışını temsil eder. Şekil 2.42'de montaj şeması ve alıştırmanın uygulanmasının adımları verilmiştir.



Şekil 2.42. Düğmeler kullanarak VE devresinin montaj şeması

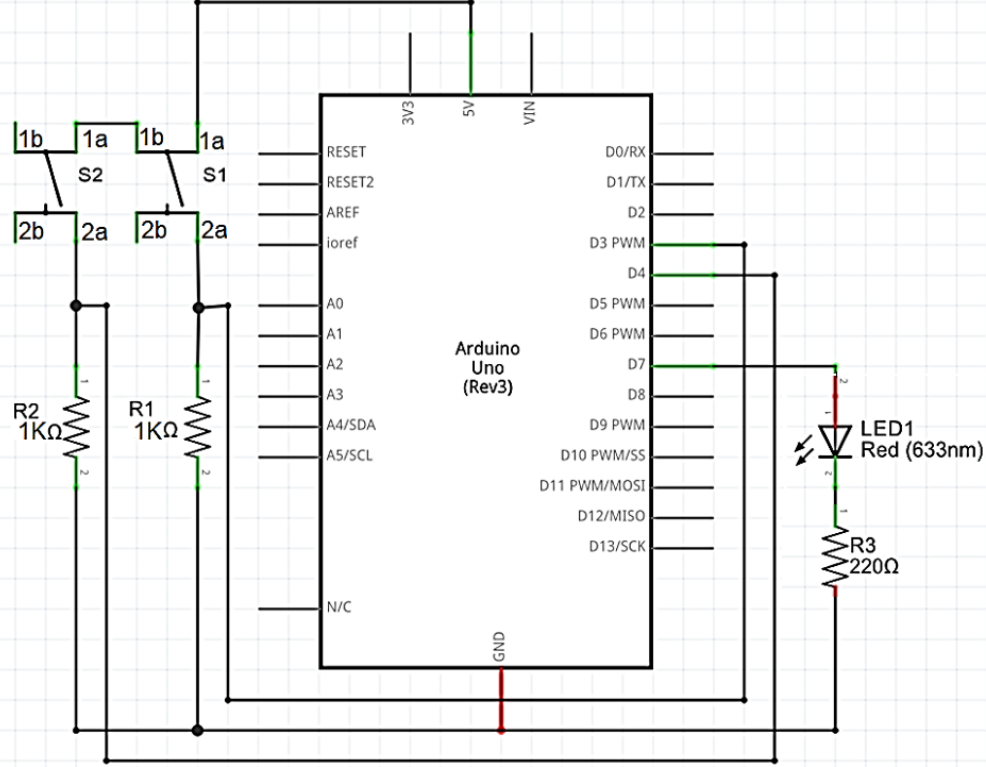
- (1) Öncelikle iki düğmeyi devre tahtasının kullanım kılavuzundan Şekil 2.36'ya uygun olarak devre tahtasının ortasına yerleştiriyoruz.
- (2) Anotun güç kaynağının pozitif kutbuna bağlandığına dikkat ederek LED diyotu takıyoruz.
- (3) Yeşil jumper kablo ile düğmelerin yan yana olann bağlantı noktalarını, S1 düğmesinin 2b bağlantı noktasını S2 düğmesinin 1b bağlantı noktasına bağlıyoruz.
- (4) R3 rezistörü ile LED diyotun çalışma akımını ayarlıyoruz. Aynı zamanda R3 rezistörü aracılığıyla LED diyotu S2 düğmesine bağlıyoruz.
- (5) Pili devre tahtasının güç bantlarıyla bağlıyoruz. Kırmızı jumper kabloyla pilin pozitif kutbunu üst güç bantına ve siyah kablo ile pilin negatif kutbunu alt bantta bağlıyoruz.
- (6) Mavi jumper kabloyla S1 düğmesinin 1a-1b bağlantısını güç kaynağının pozitif kutbuna bağlarız. 1a bağlantısının 1b ile kısa bağlı olduğunu hatırlayalım.
- (7) LED diyotun katotunu güç kaynağının negatif kutbuna bağlıyoruz.

Sadece her iki düğmeye basıldığında devreden akım akacak ve LED diyotu yanacaktır.

5. Arduino Uno R3 kullanarak Ve mantık fonksiyonunun gerçekleştirilmesi

İşlevsel (Şekil 2.43.) ve montaj şemasından (Şekil 2.44.) S1 ve S2 düğmelerinin seri bağlı olmadıkları görülebilir. S1 düğmesi bir tarafta güç kaynağına (1a-1b bağlantısı) ve diğer tarafta dijital pin D3'e (2a-2b bağlantısı) bağlıdır. Aynı durum S2 düğmesi için de geçerlidir ancak fark, D3 yerine dijital pin D4'e bağlanmasıdır. D3 ve D4 pinlerinin giriş pinleri olması gerektiğini belirtelim. Her düğme ile toplaklama arasına rezistör (R1 ve R2) bağlıdır.

Bunlar olmadan dijital pinlerin mantık seviyesi bilinemez çünkü herhangi bir düğmeye basarak 5V'luk besleme ve topraklama kısa devre yapacaktır buna da izin verilmiyor.



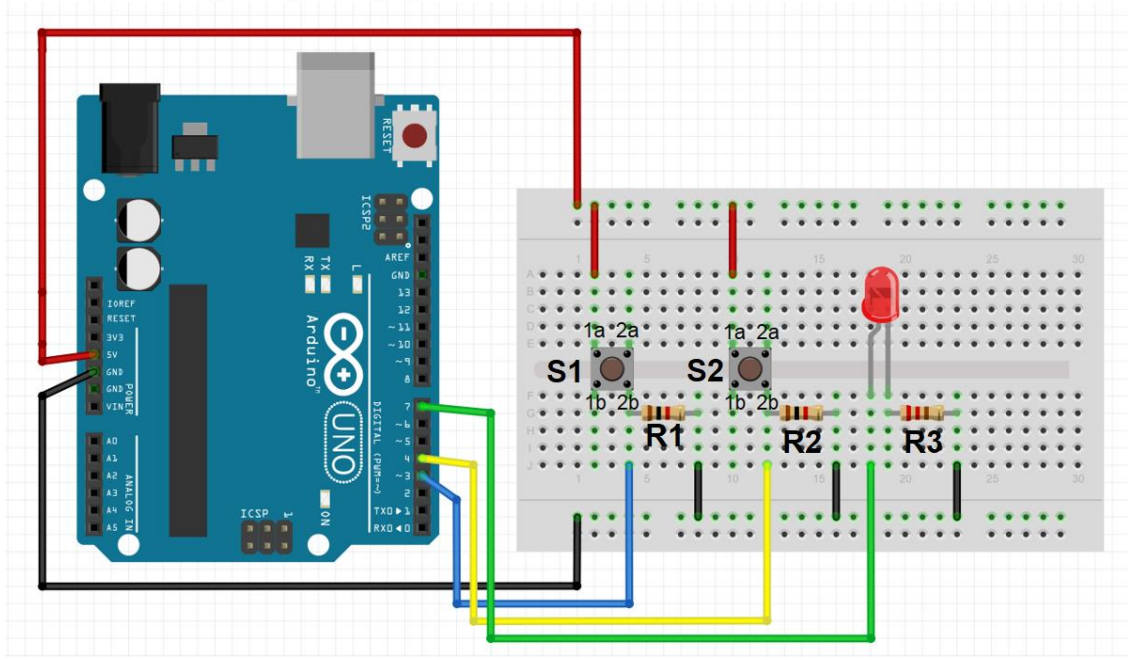
Şekil 2.43. Arduino Uno R3 ile gerçekleştirilen VE mantık devresinin işlevsel şeması

Düğmeler açık olduğunda D3 ve D4 dijital pinleri, güç kaynağı olmadığından R1 ve R2 rezistörleri aracılığıyla hiçbir akımın akmadığı topraklamaya bağlanır. Düğmeler kapanınca D3 ve D4 pinleri güç kaynağına bağlanacak ve yüksek mantık seviyesine ayarlanacaklar. LED diyotun anodu dijital çıkış pini D7 ile bağlıdır, katot ise rezistör R3 aracılığıyla topraklamaya bağlıdır. Dijital pin D7 yüksek seviyede olduğunda LED diyotu yanacaktır. Bunun gerçekleşmesi için her iki düğmenin de basılması gerekir. Düğmelerin durumu yazılım ile kontrol edilir.

Bu alıştırmamızın gerçekleştirilmesinde adımlar aşağıdaki gibidir:

- (1) Güç kaynağını devre tahtasına bağlıyoruz. Kırmızı jumper kablo ile 5V pini devre tahtasının üst güç bantına bağlıyoruz, siyah jumper kablo ile ise topraklamayı (İng. GND-Ground) alt yan banta bağlarız. Tüm elemanları tüm devre tahtasına bağladıktan sonra, alıştırmamızın sonunda iki güç kaynağının jumper kablolardan biri bağlanmalıdır.
- (2) Düğmeleri devre tahtasının ortasına yerleştiriyoruz. İki düğmenin 1a-1b bağlantı noktaları sırasıyla kırmızı jumper kablo ile güç kaynağına, 2a-2b bağlantı

- noktalarını ise R1 veya R2 rezistörü aracılığıyla topraklamaya bağlıyoruz.
- (3) Mavi jumper kablo ile S1 düğmesi ile R1 rezistörü arasındaki noktayı D3 dijital pinine bağlıyoruz. Aynısını S2 düğmesi ile R2 rezistörü ve D4 dijital pini arasındaki nokta için sarı jumper kablo ile tekrarlıyoruz.
 - (4) LED diyotu devre tahtanın üzerine yerleştiriyoruz, katodu R3 rezistörü üzerinden topraklamaya bağlıyoruz ve yeşil jumper kablo ile anodu yedinci dijital pine bağlıyoruz.



Şekil 2.44. Arduino Uno R3 ile gerçekleştirilen VE mantık devresinin montaj şeması

6. VE mantık fonksiyonunu gerçekleştirmek için Arduino Uno R3 programı

Bu uygulamalı alıştırma programlama kodunu içerir ancak programlama kodu Arduino Uno R3 platformunu programlamak için geliştirme ortamını ve programlama dilini tanıdıktan sonra kullanılabilir. Şimdilik, mantıksal VE komutunu yürütmek için sadece && operatöründen bahsedeceğiz. Program kodunda bu işlenen sarı renkle işaretlenmiştir.

Alıştırmanın tam olarak gerçekleştirilmesi için Arduino Uno R3 platformunu USB kablo ile bilgisayara bağlamak, ortamını açıp programı yazmak, Arduino Uno R3'ün belleğine yazdırıp düğmelere basmak gerekmektedir.

```
1 int S1 = 3; // S1 düğmesi üçüncü pine bağlanır.
2 int S2 = 4; // S2 düğmesi dördüncü pine bağlanır.
3 const int Led = 7; // LED diyotu yedinci pine bağlanır.
4 int S1Durum = 0; // S1Durum S1 düğmesinin durumunu
// okumak için değişkendir
```

```

5 int S2Durum = 0; // S2Durum S2 düğmesinin durumunu
// okumak için değişkendir
6 void setup() { // Pinleri yapılandırma fonksiyonunun
// başlatılması.
7     pinMode(Led, OUTPUT); // LED bağlantı pini çıkış
// pinidir
8     pinMode(S1, INPUT); // Düğmelerin bağlantı pinleri giriş.
9     pinMode(S2, INPUT); // pinleridir.
//
10 }
11 void loop(){
12     S1Durum = digitalREAD(A); // Pinlerin durumunun okunması.
13     BSTATUS = digitalREAD (B);
//If dallanma yapısının koşulunda VE mantıksal fonksiyon komutu kullanılmıştır.
//Bu komut ile düğmelerin durumu kontrol edilir.
14     if (S1Durum == HIGH && S2Durum ==HIGH( {
        digitalWrite(Led, HIGH); // Verilen koşul geçerliyse
// LED diyotu açılır.
15     }
16     else {
17         digitalWrite(Led, LOW); // Koşul geçerli değilse
// diyot açılmayacaktır.
18     }
19 }

```

7. Beklenen sonuçlar

LED diyotu sadece her iki düğme aynı anda basılmış olduğunda yanmalıdır. Düğmeler basılmadığında veya yalnızca bir düğme basıldığında LED diyotu yanmaz. Elde edilen sonuçlar beklenen sonuçlardan farklıysa ve programın derlenmesi ve yazılması sırasında geliştirme ortamı yazılım hatası bildiriyorsa elektronik elemanların bağlantısının kontrol edilmesi gerekir.

2.8.3.5. Uygulamalı Alıştırma: Anahtarlar veya Arduino Uno R3 Kullanarak VEYA Mantık Devresinin Gerçekleştirilmesi

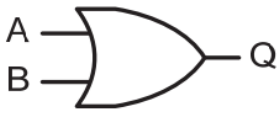
1. **Bu alıştırmanın amacı** önceki alışırmaya benzerdir, tek fark VE yerine **VEYA mantık fonksiyonunun** gerçekleştirilmesi gerekli olmasıdır. Bu alıştırma da iki bölüme ayrılmıştır, paralel bağlı düğmeler kullanılarak ve Arduino Uno R3 kullanılarak elektrik devrenin tasarımını yapmak. İki düğmenin paralel bağlı olduğu elektrik devresinin VEYA mantık fonksiyonunun donanım uygulaması olduğunu, Arduino Uno R3 ile elektrik devresinin ise yazılım uygulaması olduğunu ve VEYA mantık fonksiyonu için yeni bir programın elde edilmesi için

VE mantık fonksiyon programında sadece bir komutun değiştirilmesinin yeterli olduğunu belirtelim. Arduino Uno R3'ün elektrik devresi değişmeden kalır.

1. Bu alıştırmanın gerçekleştirilmesi için **gerekli bileşenler** şunlardır:

- Arduino Uno R3
- Devre tahtası
- Baskılı devre kartı için iki düğme
- Led diyot
- Üç rezisör ($R1=R2=1K\Omega$, $R3=220\Omega$,)
- Bağlama kabloları (jumper kabloları)

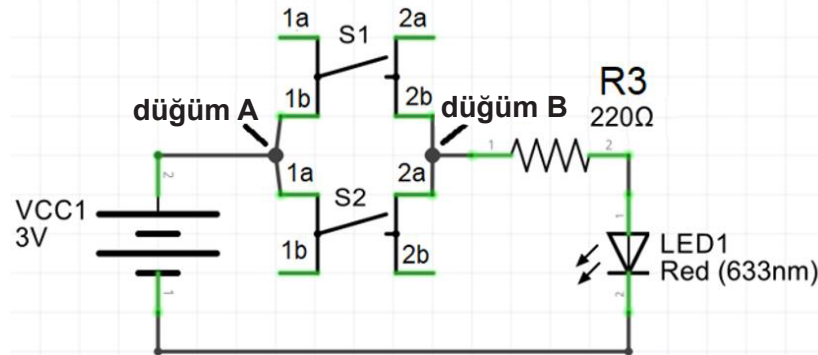
2. Alıştırma için hazırlık

Mantık sembolü		
A	B	Q
		
Doğruluk tablosu		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Şekil 2.45.VEYA devresi

VE mantıksal fonksiyonunu gerçekleştirmek için önceki alıştırmada söylenenlerin aynı bu alıştırma için de geçerlidir. Şekil 2.45'te. VEYA mantık devresinin sembolü ve doğruluk tablosu verilmiştir. VEYA devresinin çıkışında yüksek seviye yani mantıksal bir elde etmek için girişlerden birinin yüksek seviyede olması yeterlidir, tüm girişler de yüksek seviyede olabilir. Çıkışta düşük seviye yani mantıksal sıfır elde etmek için tüm girişlerin düşük seviyede olması gerekir.

3. Düğmeler kullanarak VEYA mantıksal fonksiyonunun gerçekleştirilmesi

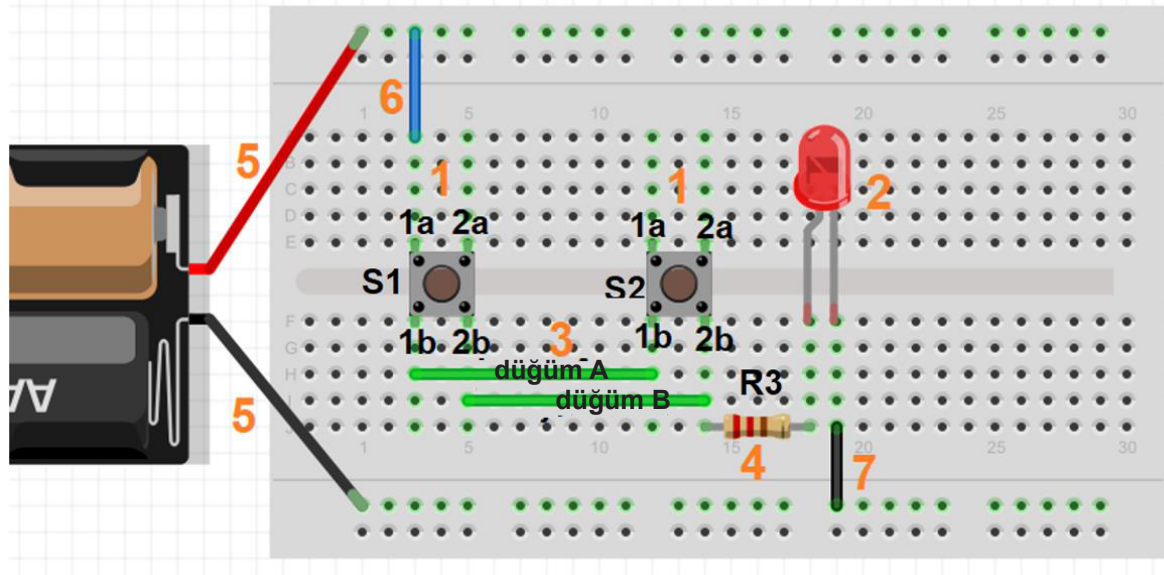


Şekil 2.46. İki paralel bağlı düğmenin işlevsel şeması

Şekil 2.44'da paralel bağlı iki düğme ve bir led diyot kullanarak VEYA mantıksal fonksiyonunun gerçekleştirilmesinin işlevsel şeması gösterilmektedir. Kapalı

devre elde etmek ve led-diyotun yanması için düğmelerden sadece birine basılması yeterlidir.

Şekil 2.47’de alıştırmamızın gerçekleştirilmesi için montaj şeması tanımlanmıştır. Elektronik bileşenleri devre tahtası kullanarak paralel bağlantıya bağlarken, temas noktalarının seçimine dikkat edilmelidir. Paralel bağlantının düğümleri, elektronik elemanların bağlantı noktaları birbirine bağlayan jumper kabloların kullanılmasıyla oluşturulur. Jumper kabloyu, paralel bağlantının parçası olması gereken elektronik elemanın konektörünün takıldığı metal banta ait bir temas noktasına yerleştirmemiz gerekiyor. Aynı metal banta ait temas noktalarının aynı numara ve farklı harf ile işaretlendiğini belirtelim.



Şekil 2.47. Düğmeler kullanılarak VEYA devresinin montaj şeması

- (1) İlk önce iki düğmeyi devre tahtasının ortasına yerleştiriyoruz
- (2) Anotun güç kaynağının pozitif kutbuna bağlandığına dikkat ederek LED diyotu yerleştiriyoruz.
- (3) 1a-1b ve 2a-2b numaralılarla işaretlenmiş iki düğmenin bağlantıları noktalarını yeşil jumper kabloyla bağlayıp A ve B düğümlerini oluşturuyoruz.
- (4) R3 rezistörü ile LED diyotun çalışma akımını ayarlıyoruz. Aynı zamanda LED diyotunu rezistör aracılığıyla A düğümüne bağlıyoruz.
- (5) Devre tahtasının güç bantlarına pili bağlıyoruz. Kırmızı jumper kablosu ile pilin pozitif kutbunu üst güç bantına, siyah kabloyla ise negatif kutbu alt banta bağlıyoruz.
- (6) Mavi jumper kablolarıyla S1 düğmesinin 1a-1b bağlantılarını güç kaynağının pozitif kutbuna bağlıyoruz.

(7) LED diyotun katotunu güç kaynağının negatif kutbuna bağlıyoruz.

LED diyotunun yanması için iki düğmeden birinin basılmış olması yeterlidir.

4. Arduino Uno R3 kullanarak VEYA mantıksal fonksiyonunun gerçekleştirilmesi

Yazılım çözümleri genellikle donanım çözümlerinden çok daha basittir. Bu alıştırmamızın uygulanması için yeni bir elektrik devresi yerine, işlevsel şeması şekil 2.43'te ve montaj şeması şekil 2.44'te gösterilen önceki alıştırmadaki elektrik devresini kullanacağız. Elektrik devre basit olduğunda kablolamada herhangi bir değişiklik yapmak kolaydır. Ancak çok sayıda elektronik bileşen ve jumper kablo olması durumunda tasarım değişikliği karmaşık olabilir.

5. VEYA mantıksal fonksiyonunun gerçekleştirilmesi için Arduino Uno R3 Programı

VEYA mantık fonksiyonunu gerçekleştirmek için, VE mantıksal fonksiyon programındaki sadece bir komut değiştirmek yeterlidir. VE mantıksal işlevi için && operatörünü VEYA mantıksal işlemi için yeni | | operatörüyle değiştireceğiz.

```
1 int S1 = 3; // S1 düğmesi üçüncü pine bağlanır.
2 int S2 = 4; // S2 düğmesi dördüncü pine bağlanır.
3 const int Led= 7; // LED diyotu yedinci pine bağlanır.
4 int S1Durum = 0; // S1Durum S1 düğmesinin durumunu
// okumak için değişkendir.
5 int S2Durum = 0; // S2Durum S2 düğmesinin durumunu
// okumak için değişkendir
6 void setup() { // Pinleri yapılandırma fonksiyonunun
// başlatılması.
7     pinMode(Led, OUTPUT); // LED bağlantı pini çıkış.
// pinidir
8     pinMode(S1, INPUT);
9     pinMode(S2, INPUT); // Düğmelerin bağlantı pinleri giriş.
// pinleridir
10 }
11 void loop(){
12     S1Durum=digitalRead(A); // Pinlerin durumunun okunması.
13     BStatus = digitalRead(B);
//If dallanma yapısının koşulunda VE mantıksal fonksiyon komutu kullanılmıştır.
//Bu komut ile düğmelerin durumu kontrol edilir.
14     if (S1Durum == HIGH | | S2Durum == HIGH) {
        digitalWrite(Led, HIGH); // Verilen koşul geçerliyse
// LED diyotu açılır.
```

```
15     }
16     else {
17         digitalWrite(Led, LOW);    // Koşul geçerli değilse
                                        // diyot açılmayacaktır.
18     }
19 }
```

6. Beklenen sonuçlar

Önceki alıştırımda olduğu gibi, Arduino Uno R3 ile VEYA mantık fonksiyonunu tam olarak gerçekleştirmek için geliştirme ortamını ve komutlar setini incelemek gerekir. Elektronik bileşenleri bağlayıp programa girdikten sonra düğmelerden en az birine veya her ikisine aynı anda basıldığında LED diyotu yanmalıdır.

Sonuçlar

Bir bilgisayarın donanım bileşenleri şunlardır: mikroişlemci, bellekler, veri yolları ve çevresel cihazlar. Mikroişlemci verileri işler, bellekler bunları saklar, veri yolları ise verileri iletir. Çevresel cihazlar aynı zamanda giriş-çıkış birimleri olarak da bilinir.

İşletim sistemi donanım ve yazılım arasında bir aracıdır. İşletim sistemi, donanım bileşenleri: işlemciyi, bellekleri, giriş ve çıkış cihazları yöneten yazılımdır. Geliştirme programı, yeni yazılım oluşturmak için bir yazılımdır.

Mikroişlemcinin iki temel işlevi vardır: programları yürütür ve anakarttaki diğer cihazları yönetir. Bir mikroişlemcinin kurucu parçaları şunlardır: yazmaçlar, aritmetik-mantık birimi ve kod çözücülü kontrol birimi. Mikroişlemcinin en önemli özellikleri şunlardır: çalışma frekansı, verilerin büyüklüğü, akış kavramı, komut kümesinin karmaşıklığı, çekirdek sayısı, pin düzeni vb.

RAM belleği işlemciye bilgi sağlar, bilgileri ise sabit diskten alır. Tüm programlar sabit diskte bulunur ve işlenmeleri gerektiğinde RAM'e aktarılır.

Sabit disk, SSD cihazları ve dış bellekler (kompakt diskler, USB bellekler, SD kartlar vb.) kalıcı veri depolama için kullanılır. Anakartın seçimi işlemci seçimine bağlıdır, çünkü

her işlemci tipi uygun yuva gerektiren benzersiz bir pin düzenine sahiptir. Çipset, veri akışını kontrol eden özel entegre devreler setidir.

DDR4, DDR3, DDR2 ve DDR RAM modülleri pin sayısı ve düzeni, güç kaynağı ve çalışma frekansı veya hız açısından birbirinden farklıdır.

PCI Express×16 ve AGP, grafik kartlarını bağlamak için en iyi konektörlerdir. PCI yuvası ses kartlarını, ağ kartlarını bağlamak için kullanılır. SATA2 ve SATA3 konektörleri sabit disk ve SSD belleği bağlamak için kullanılır.

İşlemci ve RAM belleğin kurulumu sırasında altın pinlere basılmamalı ve dokunulmamalıdır.

Çoğu güç kablosunun benzersiz bir tasarımı vardır, bu nedenle bunları anakarta ve diğer cihazlara bağlarken hata yapılamaz.

SATA2 konektörü mavi renklidir ve sabit disk bağlamak için kullanılır. SATA3 konektörü genellikle beyaz renktedir ve SSD belleği bağlamak için kullanılır.

Arduino Uno R3 platformunun temel bileşeni ATmega328 mikro denetleyicisidir. Program belleği 32KB, veri EEPROM belleği 1KB ve iç RAM belleği 2KB kapasiteye sahiptir. Maksimum frekans 20MHz'dir.

Arduino Uno R3'te 14 dijital pin, altı analog giriş, üç LED diyot göstergesi ve bir yerleşik LED diyotu vardır. Darbe genişlik modülasyonu analog sinyallerle yapılmalıdır.

Arduino Uno R3'ün giriş birimleri şunlardır: sensörler (piezo, eğim sensörü, fotorezistör, sıcaklık sensörü), düğmeler, potansiyometreler. Çıkış birimleri şunlardır: ekranlar, motorlar, bipleyciler.

LED diyotun yanması için uzun bağlantı ucu (anot) kısa bağlantı ucuncan (katot) daha yüksek potansiyele bağlamak gerekir. Düğmenin aynı tarafındaki uçlar elektriksel olarak bağlı değildir.

Kapasitörler hızlı gerilim değişimlerini önlemek için sensöre veya motora paralel olarak bağlanır. Rezistörler çalışma akımının gücünü ayarlamak için kullanılır.

Kişisel korunma ve elemanların hasar görmesini önlemek için, pil veya bilgisayar olmasında bağımsız olarak, herhangi bir kurulumu başlamadan önce güç kaynağı kapatılmalıdır.

Arduino eklentilerinden şunları belirteceğiz: ESP8266-01 Wi-fi modülü, Ethernet kalkanı, SM/GPRS Arduino kalkanı, motor ve röle kalkanı.

Arduino Uno R3'ün giriş-çıkış pinleri için maksimum gerilim 5,5V'tur, Arduino Uno R3'ün bir pini için maksimum giriş akımı 40mA'dır.

Raspberry Pi 3B+ bileşenleri şunlardır: Broadcom BCM2837 SoC (System on Chip) sistem çipi, RAM belleği, radyo vericisi, dört USB bağlantı noktası, RJ45 konektörü, HDMI, kamera ve LCD ekran için özel konektörler ve giriş ve çıkış cihazları için 40 pinli soket (yuva).

Devre tahtanın plastik maskesinin altında, elektronik elemanların elektriksel bağlantısını gerçekleştiren, birkaç grup halinde paralel olarak yerleştirilmiş iletkenler bulunmaktadır. İletkenlerin üzerinde klips şeklinde kontaklar vardır.

Tekrarlama soruları

1. Mikrobilgisayarın dört bileşeni hangileridir ve işlevleri nedir?

2. Mikroişlemcinin üç bileşeni hangileridir ve işlevleri nedir?

3. Mikroişlemcinin en önemli özelliklerini sayın?

4. Belleklerin en önemli iki özelliği nedir?

5. RAM çalışma, geçici rastgele erişim belleğidir? Açıklayınız?

6. Sabit disk bellek cihazının SDD cihazına göre avantajları ve dezavantajları nedir?

7. Kişisel bilgisayarın ana kartındaki çipset neyi temsil eder ve ne işe yarar?

8. DDR RAM modülünün türü nasıl belirlenir?

9. Grafik kartını anakarta bağlamak için hangi yuvalar kullanılıyor?

10. Her bilgisayar açıldıktan sonra BIOS ne yapar?

11. Anakart'a SATA2 ve SATA3 konektörlerinin ne işe yaradığını açıklayınız?

12. Video cihazlarını bağlamak için en iyi konektör hangisidir?

13. İşlemcinin yuvasına kurulumunu dört adımda açıklayınız?

14. RAM belleğin anakart yuvalarına kurulumu sırasında nelere dikkat edilmelidir?

15. Anakart üzerindeki F (Front), SYS_FAN ve PWR-FAN bağlantıları ne için kullanılır?

16. Anakartın ve işlemcinin güç kablolarını adlandırın? Her kabloda kaç pin vardır?

17. ATmega 328 mikrodenetleyicinin en önemli özellikleri nelerdir?

18. Arduino Uno R3 platformuna güç sağlamanın üç yolu hangileridir?

19. Arduino Uno R3'ün hangi dijital pinleri analog sinyallerini iletmek için kullanılabilir?

20. Arduino Uno R3 kaç LED diyot göstergesi içerir ve her biri ne için kullanılır?

21. Arduino platformları için birkaç giriş cihazı listeleyiniz?

22. Devre tahtanın yüzeyindeki hangi boşluklar elektriksel olarak bağlıdır?

23. Bir düğmenin kaç pini vardır ve bunlardan hangileri elektriksel olarak bağlıdır?

24. LED diyonu elektrik devresine bağlarken nelere dikkat edilmelidir?

25. Servo motorun nasıl çalıştığını açıklayınız?

26. Piezo bileşeninin uygulaması nedir?

27. Arduino Uno R3 platformunun bellek organizasyonunu açıklayınız!

28. Giriş-çıkış cihazlarını Arduino veya Raspberry Pi'ye bağlarken kapasitörler ve rezistörler ne için kullanılır?

29. Arduino kalkanlarının bazı işlevlerini listeleyiniz?

30. Arduino Uno R3'ün bir pini için maksimum giriş akımı ne kadardır?

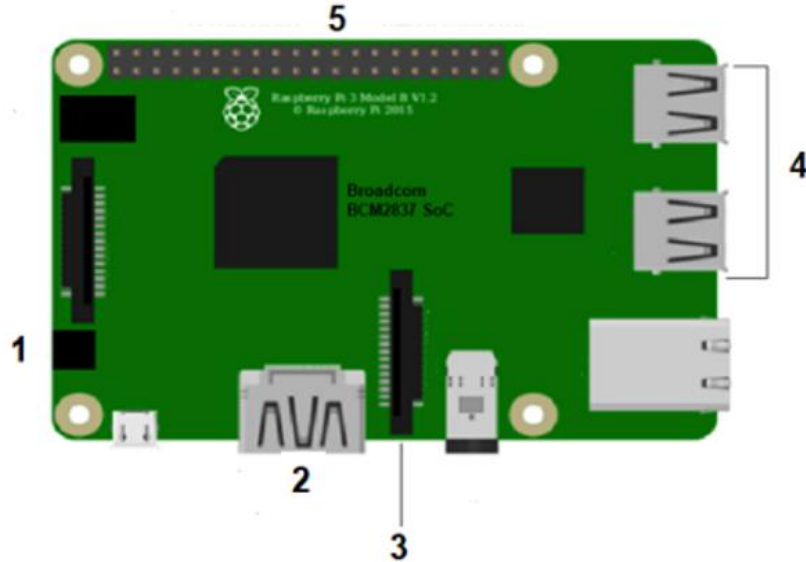
31. Devre tahtası Arduino Uno R3'ün güç kaynağı ile nasıl bağlandığını açıklayın?

32. Raspberry Pi mikrobilgisayarıyla hangi uygulamalar oluşturulabilir?

33. Raspberry Pi 3'ün GPIO (General Purpose Input Output) yuvasında 40 pin bulunmaktadır, ancak GPIO numaraları pinlerin fiziksel numaralarından farklıdır. Açıklayın!

34. Raspberry Pi için sensörlü ve ekranlı (İng. Sense hat) eklentinin bileşenlerini listeleyin!

35. Alttaki şekil 2.48'de Raspberry Pi 3B+'nın, rakamlarla işaretlenmiş bileşenleri adlandırın?



Şekil 2.48. Raspberry Pi 3B+'nın bileşenleri

3. İşletim Sisteminin Kurulumu

3.1. İşletim Sistemi Türleri

İşletim sistemi, uygulama programlarını çalıştırmak ve bilgisayar sisteminin etkinliklerini koordine etmek için gereken yazılımdır. Diğer programların faydalı işlerini yapabileceği bir ortam sağlamaktadır. İşletim sistemi de diğer programlar gibi bir programdır ancak çok daha karmaşık, daha güçlü ve daha büyüktür. Bu nedenle, her bölüm için iyi tanımlanmış fonksiyon, giriş ve çıkış ile bölüm bölüm oluşturulmalıdır. İşletim sistemleri iki açıdan incelenebilir: kullanıcı açısından ve sistem açısından.

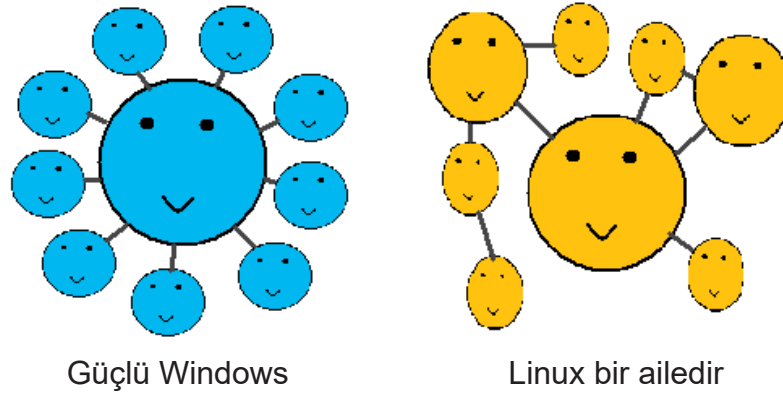
Kullanıcı açısından bakıldığında işletim sistemi, kullanıcının donanım kaynaklarının kullanımındaki işini en aza indirmek, yani mikrobilgisayar sistemlerinin rahat kullanımını sağlamak ve performanslarını artırmak için tasarlanmıştır. Bilgisayar, klavye aracılığıyla kullanıcıdan yönergeler alır ve monitör aracılığıyla mesajlar verir. Bu komutların biçimine ve giriş yöntemine bağlı olarak, iki tür kullanıcı arayüzü ayırt ediyoruz: komut satırı (İng. interface command line) veya grafik arayüzü (İng. graphics user interface). Komut satırı arayüzünde metinsel komutlar kullanılır, grafik arayüzünde ise pencereler, menüler ve seçim listelerinden oluşan sistem kullanılır. Komut satırı arayüzüne erişmek için Windows işletim sisteminde Arama alanına (İng. Search) cmd (İng. Command Prompt) yazılması gerekir.

Sistem açısından bakıldığında işletim sistemi, kontrol ve yönetim yazılımıdır ve donanımla yakından ilişkilidir. İşletim sistemi hangi programın ve ne zaman çalıştırılacağına, programlara ne kadar bellek ayrılacağına karar verir, kullanıcı tarafından verilen komutların doğru yürütülmesiyle ilgilenir vb.. Devamda işletim sisteminin fonksiyonlarını kısaca tanıyacağız. [10]

- Programların yürütülmesi. İşletim sistemi, yönergeleri RAM belleğinden mikroişlemciye aktarılmasından, kodlarının çözülmesinden ve yürütülmesinden sorumludur.
- Donanım kaynaklarını dağıtım programları. Mikrobilgisayar sisteminin verimli çalışması için işletim sisteminin donanım kaynaklarını (mikroişlemci, bellek alanı, giriş-çıkış birimleri) belirli kullanıcı ve programlara nasıl dağıtacağına karar vermesi gerekir. Örneğin bir mikroişlemci, bir görevden diğerine sürekli geçiş yaparak birden fazla görevi aynı anda gerçekleştirebilir, ancak geçiş o kadar sık gerçekleştirilir ki kullanıcılar ve programları sürekli etkileşim halinde olur.
- Belleği yönetme programları. İşletim sistemi, hangi programların ve verilerin ikincil bellekten birincil belleğe aktarılacağına karar verir, gereklere göre bellek alanı ayırır ve serbestleştirir, belleğin mevcut anda kullanılan kısımlarını izler, sabit diskin formatlanması, temizlenmesi ve bakımı için kullanılır, önbelleğinin çalışmasıyla ilgilenir.
- Giriş-çıkış işlemleri ve cihazları yönetme programları. Çevresel cihazlara veya çevresel cihazlardan veri aktarımını kontrol etmek ve bunların geçici olarak depolanmasını için cihaz denetleyicileri olarak adlandırılan özel entegre devreler kullanılır. Veri aktarımının başlayabilmesi için denetleyicilerin yazmaçlarına kesin olarak belirlenmiş bir değere sahip ikili kodların yazılması gerekmektedir. Bunu, aslında "cihazı hareket edici" olan sürücü gerçekleştirir. Sürücüler, işletim sistemi ile denetleyiciler arasında aracılık eden yardımcı programlardır.
- Koruma ve güvenlik programları. Koruma, kullanıcının donanım kaynaklarına erişimine ve bunların yetkili kullanımını kontrol etmeye yönelik mekanizmadır. Sistemin iç ve dış saldırılara karşı savunulması bir güvenlik meselesidir. Bu kategoride antivirüs programları aittir.
- İletişim programları. Bu programlar bir bilgisayardan diğerine uzaktan dosya aktarımına, web sayfalarında arama yapılmasına, e-posta gönderilmesine vb. olanak sağlar.
- Dosya yönetimi programları. Bu programlar dosya ve dizinleri oluşturur, seçer, siler, kopyalar, yeniden adlandırır.
- Durum bilgileriyle çalışma programları. Durum bilgileri saat, tarih, kullanılabilir bellek miktarı, kullanıcı sayısı vb.'dir.
- Programlama dillerine destek. İşletim sistemi kullanıcıya en sık kullanılan programlama dilleri için editörler, derleyiciler, çeviriciler, hata ayıklayıcılar ve yorumlayıcılar sağlar. Bu araçlar uygulama yazılımı oluşturmak için gereklidir.

Windows işletim sistemi grafik arayüze sahip ilk işletim sistemidir ve Microsoft firmasının ürünüdür. En çok kullanılan ve popüler işletim sistemlerinden biridir. Günümüzde kişisel bilgisayar kullanıcılarının %80'i Windows işletim sistemini kullanıyor. Başlangıçta Windows, o zamanlar popüler olan DOS işletim sisteminin yalnızca bir yükseltmesiymiş. En başından beri fareyle, simgeler, menülerle çalışmak ve bir uygulamadan diğerine basit geçiş için yazılım desteği varmış. **Windows işletim sisteminin avantajları şunlardır:** çalışması basit (user friendly), çevresel cihazlarla bağlantı için en iyi destek, uygulama yazılımının kullanılabilirliği, otomatik kurulum (plug and play), video oyunları için en iyi performanslar, web sitelerin çoğunun Windows işletim sistemiyle uyumluluğu. **Dezavantajları** şunlardır: Pahalı bilgisayar konfigürasyonlarına duyulan ihtiyaç, kapalı kod, İnternet saldırıları ve yönetimsel koruma açısından zayıf güvenlik, bilgisayar virüslerine karşı duyarlılık, lisans gerekliliği, zayıf uzman desteği ve sıklıkla çalışma hızını artırmak için işletim sisteminin yeniden yükleme gereği.

Linux daha yeni bir işletim sistemidir. Bu fikri, 1991 yılında farklı platformlarda kullanılabilecek yeni, ücretsiz, stabil, açık kaynaklı işletim sisteminin çekirdeğini oluşturmaya ilişkin çalışma yayınlayan Finlandiyalı öğrenci Linus Torvalds tarafından ortaya koyulmuş. **Açık veya serbest kod**, tüm kullanıcıların kaynak koduna erişebilmesi ve kendi ihtiyaçlarına göre değiştirebilmesi anlamına gelir. Açık kod daha güvenilir bir geliştirme sistemidir. Analiz sırasında programcılar hataları ve virüsleri keşfeder, düzeltir ve çözümlerini Linux topluluğuyla paylaşır. İşletim sisteminin çekirdeğinin çeşitli koruma düzeyleri vardır. Herhangi bir programın yürütülebilir versiyonunu yapmadan önce kontrol edilir. Ayrıca, web tarayıcısı işletim sisteminden bağımsızdır. Şekil 3.1'de. Linux işletim sisteminin organizasyonunu Windows işletim sisteminin organizasyonu ile karşılaştırmalı olarak sembolik olarak gösterilmiştir.



Şekil 3.1. İki farklı ancak başarılı kavram

Linux işletim sisteminin mimarisi dallara ayrılan bir ağaç gibidir ve her dal topluluğun bir üyesini temsil eder ve herkes daha hızlı, daha güçlü ve daha güvenilir bir sistem oluşturmaya katkıda bulunabilir. Windows, işletim sistemleri dünyasının efendisi olabilir ancak Linux devrimi, işletim sistemlerinin oluşturulma şeklini kesinlikle değiştirdi. Başlangıçta, Linux işletim sistemi daha ileri düzey kullanıcılara yönelikmiş, ancak etkili GUI (Graphical User Interface) arayüzünün ortaya çıkışıyla Linux giderek daha popüler oluyor. Günümüzde Linux, Google, Facebook, Twitter gibi büyük şirketlerin sunucularında işletim sistemi olarak kullanılmaktadır. Birçok firma temel Linux çekirdeğini alarak geliştiriyor ve yeni lisanslı işletim sistemleri oluşturuyor. Bu, Google şirketinin Android işletim sistemi ve bir grup hevesli, Raspberry Pi mikrobilgisayar sevenler tarafından oluşturulan Raspbian işletim sistemidir.

Bugün cep telefonu kullanıcılarının %74,3'ü Android işletim sistemini kullanmaktadır. **Android işletim sisteminin avantajları** şunlardır: çok sayıda ücretsiz uygulama (Google Play Store, işletim sisteminin kendisinde yerleşik bir uygulamadır), farklı üreticilerin (Samsung, Huawei, Motorola) cihazlarıyla uyumluluk, açık kaynak, basit yeni uygulamalar oluşturmaya yönelik geliştirme ortamları, bellek alanının kolayca genişletilebilmesi, İnternette paylaşım olanağı, birçok cihazla basit iletişim, yapılandırma özgürlüğü ve uygulama seçimi. Bu işletim sisteminin **dezavantajları** şunlardır: farklı ekran boyutları nedeniyle uygulamaların geliştirilmesinin daha zor olması, arka plan uygulamaları nedeniyle daha düşük hız, daha az güvenlik ve virüslere karşı duyarlılığıdır.

Raspbian sadece Raspberry Pi için işletim sistemidir. Daha sonra kurulumunu ve kullanım yöntemini tanıyacağız.

3.2. Modüler Birimi İçin Uygulamalı Alıştırmalar: İşletim Sisteminin Kurulumu

3.2.1. Yazılım Kurulumu İçin Hazırlık

Günümüzde uygulama ve sistem yazılımının kurulumu çok basittir, birkaç adımda gerçekleştirilir ve her adım prosedür sırasında ayrıntılı olarak açıklanır. Ancak başarılı kurulum için kullanıcının aşağıdaki kurallara uyması gerekir.

- Kurduğumuz programın bizim için yararlı olacağından emin olmalıyız. Kurulan her program, kullansak da kullanmasak da, bellekte yer kaplar ve bilgisayarın çalışmasını yavaşlatır.
- Kurulum yazılımını indirmek için güvenilir ve doğrulanmış bağlantılar kullanmalıyız.
- Kalıcı bellekte yeterli boş alan olup olmadığını kontrol etmeliyiz. Yazılımın doğru çalışması için bilgisayar yapılandırmasının gereklilikleri karşılayıp karşılamadığını kontrol etmemiz gerekir.
- Kaldırma seçeneği olup olmadığını kontrol etmemiz gerekir
- Kullanıcının kendi seçeceği bir kurulumu (İng. Custom) seçmesi tavsiye edilir çünkü bu şekilde kendisine sunulan olanakları seçmiş oluyor.
- Kurulum sırasında kesintiyi önlemek için kararlı bir güç kaynağı ve güvenilir İnternet erişimi sağlanmalıdır.
- Yazılım kurulumuna ilişkin uygulamalı alıştırmalar ders öğretmenin onayı olmadan yapılmamalıdır.

3.2.2. Uygulamalı Alıştırma: Windows 10 İşletim Sisteminin Kurulumu

1. Windows 10 işletim sisteminin kurulumu için hazırlıklar

İlk önce Windows 10 işletim sisteminin türünü seçmeniz gerekiyor. Windows 10 Home ev kullanımı içindir, Windows 10 Pro profesyoneller içindir, Windows 10 Enterprise şirketler için ve Windows 10 Education öğrenciler içindir. Tüm işletim sistem türleri **32 ve 64 bit sürümlerde** mevcuttur. 64-bitli sürüm,

32 bitli işlemciye sahip bilgisayara yüklenemeyeceğini vurgulayalım. 64 bitli işlemciler ile veriler daha hızlı işleniyor ve 4 GB üzeri RAM belleğin daha verimli kullanılmasına olanak sağlıyor.

Windows 10 işletim sistemi, Windows 7 ile aynı bilgisayar yapılandırmasıyla çalışabilir: 1 GHz'in üzerinde çalışma frekansına sahip işlemci, 2 GB RAM bellek kapasitesi, 32 bitli sürüm için 16 GB ve 64 bitli sürümü için 20 GB kapasiteli sabit disk, WDDM'li (İng. Windows Display Driver Model) grafik kartı, 800 × 600 piksel çözünürlüklü ekran.

Bilgisayarda Windows'un daha eski bir lisanslı sürümü varsa, Windows 10'a yükseltilebilir ve yükseltme ücretsizdir. Ancak yeni bir bilgisayar ise Windows 10 işletim sisteminin kurulması gerekir. Bunun için kurulum bellek modülü (USB veya DVD) gereklidir.

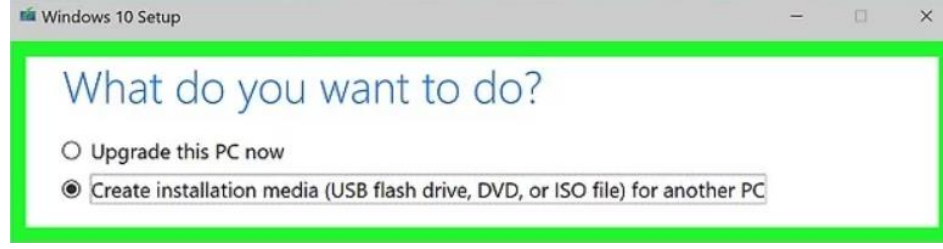
Windows 10'un aktivasyonu için ürün anahtarı gerekir. Ürün anahtarı 25 haneli lisans doğrulama kodudur. İşletim sisteminin teslimi ile birlikte kodu sağlayan Windows 10 satıcılarının resmi listesi Microsoft'un web sitesinde bulunabilir veya kurulum programı İnternet'ten indirilmişse e-posta yoluyla gönderilebilir. Dijital lisans almanın başka bir yolu Microsoft kullanıcı hesabıdır. Bu amaçla Settings → Update and Security → Activation seçeneğini seçin. Hesap ekle (İng. Add an account) düğmesine basıyoruz ve kullanıcı adımız ve şifremizle giriş yapıyoruz.

Windows 10 kurulum prosedürünü üç bölüme ayıracağız: kurulum USB bellek modülü oluşturmak, bunu BIOS yönetim programı aracılığıyla çağırmak ve son olarak işletim sistemini kurmak. [11]

2. Kurulum USB bellek modülünün oluşturulması

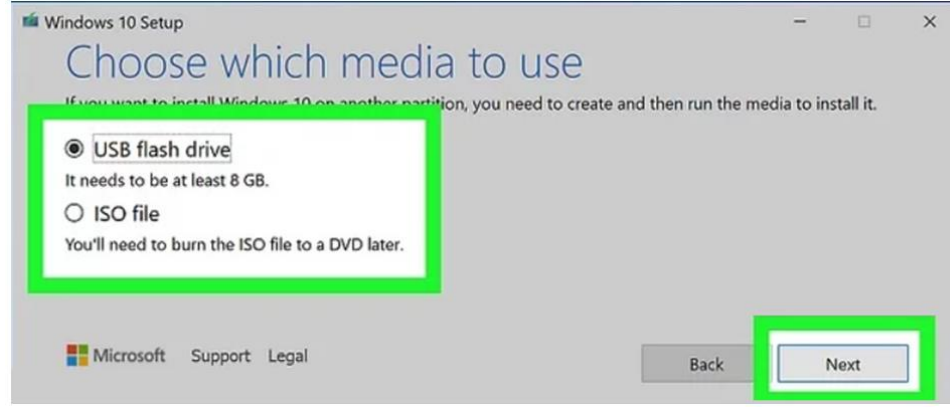
- (1) USB bellek modülünü bilgisayara bağlıyoruz. Minimum 8 GB kapasiteye sahip olması gerekir.
- (2) Kurulum modülünü oluşturma aracını (İng. Media Creation Tool) indirmek için bağlantı <https://www.microsoft.com/en-us/software-download/windows10%20> bağlantısıdır. Download tool now (aracı şimdi indir) düğmesine basılır ve birkaç saniye içinde araç İndirme klasöründe görünür.
- (3) Araca çift tıklayarak kurulum modülünün oluşturulması başlıyor. Belirtilen koşulları kabul ediyoruz (İng. Accept)

- (4) Bir sonraki pencere ile kurulum modülü oluşturma seçeneğini seçiyoruz.



Şekil 3.2. Windows kurulum modülü oluşturmaya başlama

- (5) Dil, işletim sistemi ve mimariyi seçiyoruz
(6) Modül tipini seçiyoruz ve Next'e basıyoruz.



Şekil 3.3. Bellek modülü'nün seçimi, USB veya DVD

- (7) Birden fazla USB bellek modülü varsa birini seçiyoruz.



Şekil 3.4. Kurulumu kaydetmek için USB modülünü seçme

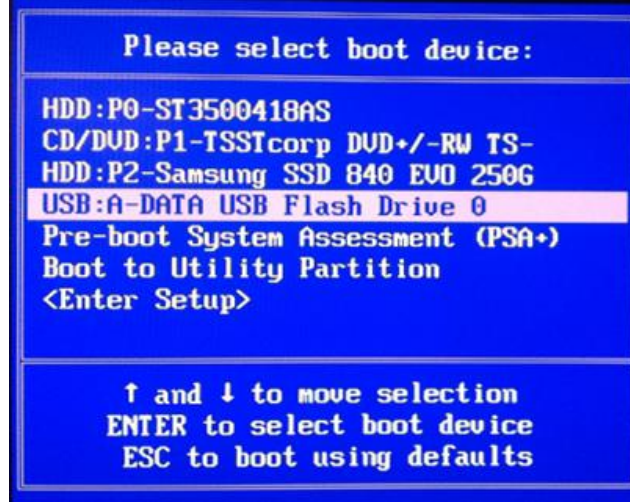
- (8) Belgelerin yüklenmesini bekliyoruz ve sonunda Son'a (Finish) basıyoruz.

3. Kurulum USB bellek modülünün çağırılması

- (1) Modülü bağladıktan ve bilgisayarı açtıktan sonra kurulumun bulunduğu

cihazın seçimi menüsünün bulunduğu BIOS programını açıyoruz. Bunun için bilgisayarı açtıktan hemen sonra **F1, F2 veya Del** tuşuna basmamız gerekiyor. Bu düğme farklı bilgisayar üreticileri için farklıdır.

- (2) Üst menüden Boot seçeneğini seçiyoruz ve ardından klavye yardımıyla yukarı aşağı hareket ederek cihazı seçiyoruz ve Enter tuşuna basıyoruz.



Şekil 3.5 Kurulum cihazının seçimi

- (3) Enter tuşuna bastıktan sonra kurulum penceresi (İng.Windows Setup Wizard) açılır.

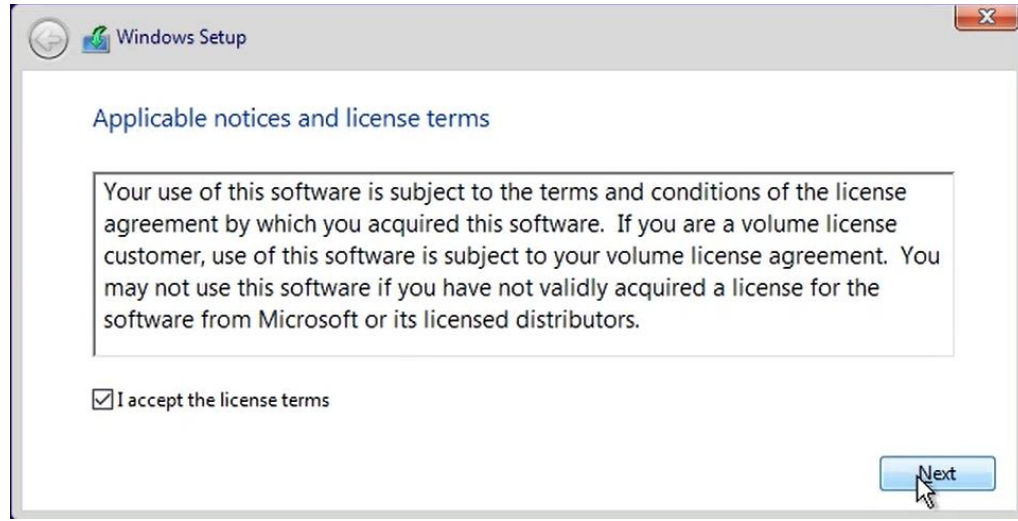
4. Windows 10 işletim sisteminin kurulumu

- (1) İlk pencerede dil, saat ve tarih formatı ve veri girişi için klavye dili (US veya UK) seçiliyor. Next'e basıyoruz ve ikinci pencere açılıyor.



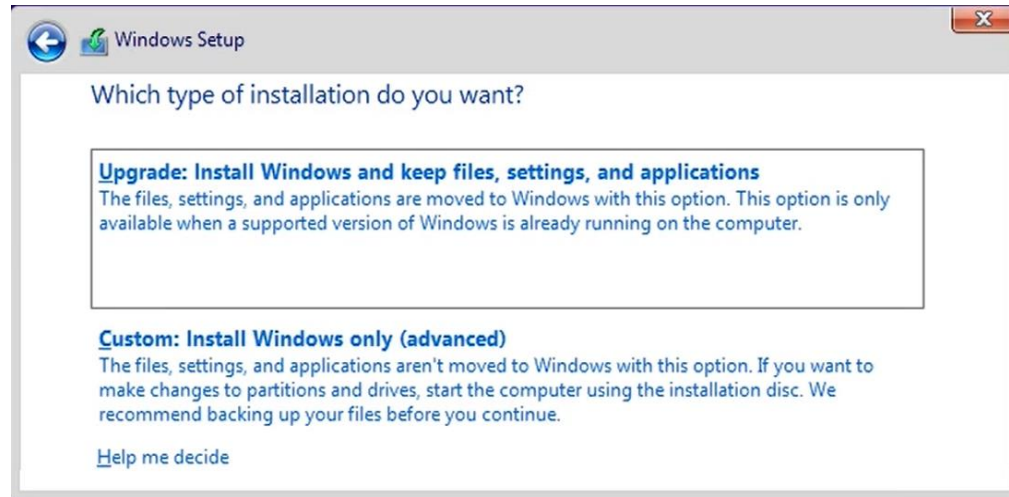
Şekil 3.6. Standart dil, saat ve veri giriş dilinin seçimi

- (2) “Şimdi Kur” (Install Now) seçeneği seçildiğinde kurulum işlemi başlar.
- (3) Kurulumun başlangıcında, kullanıcıdan ürün **anahtarını girmesinin** istendiği bir pencere açılır. Bu anahtar lisansı onaylayan 25 haneli koddur. Kod, Microsoft tarafından işletim sisteminin teslimatıyla birlikte verilir veya kurulum programının İnternet’ten indirilmesi durumunda e-postayla gönderilebilir.
- (4) Üçüncü pencere işletim sisteminin lisansı ile ilgili koşulları kabul etmek içindir.



Şekil 3.7. Lisans koşullarının kabul edilmesi

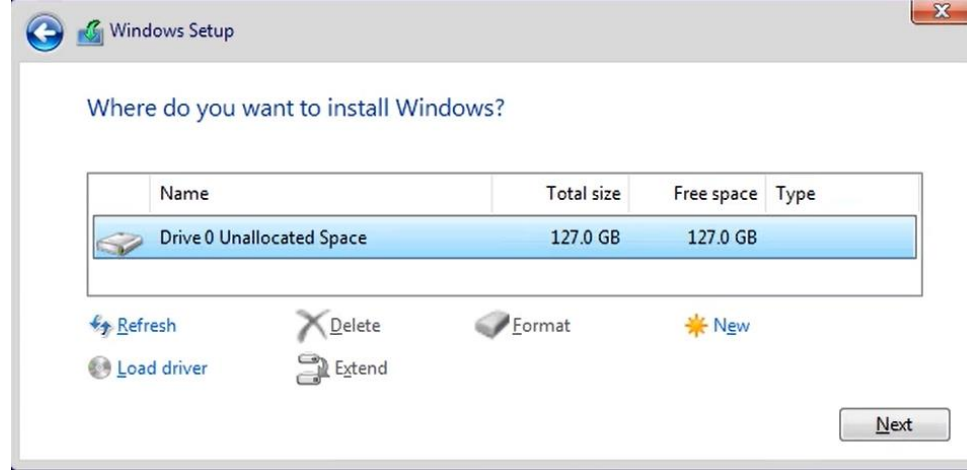
Koşulları sol taraftaki kutuyu işaretleyerek ve Next tuşuna basarak kabul ediyoruz.



Şekil 3.8. Yükseltme veya yeni Windows yüklemesi arasında seçim yapmak

Dördüncü pencere kurulum şeklini seçmek içindir. İki olasılık vardır: Upgrade veya Custom. İlk seçenek, mevcut Windows işletim sisteminin yükseltilmesi anlamına gelir. Bu kurulum ücretsizdir, ürün anahtarı gerektirmez ve kullanıcı verileri silinmez. İkinci seçenikle bilgisayarda kaydedilmiş olan tüm veriler silinir ve bu yeni kurulum anlamına gelir.

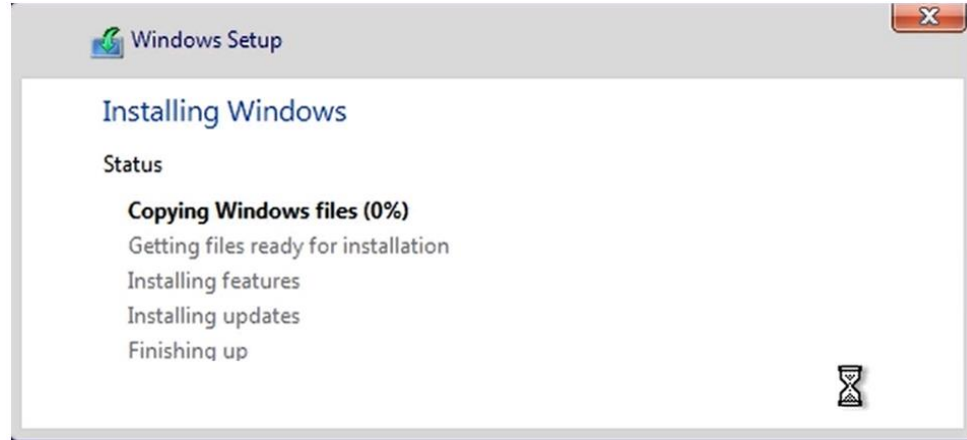
- (6) Beşinci pencere işletim sistemini kuracağımız diski seçmek içindir.



Şekil 3.9. Sabit diskten bir bölüm seçme

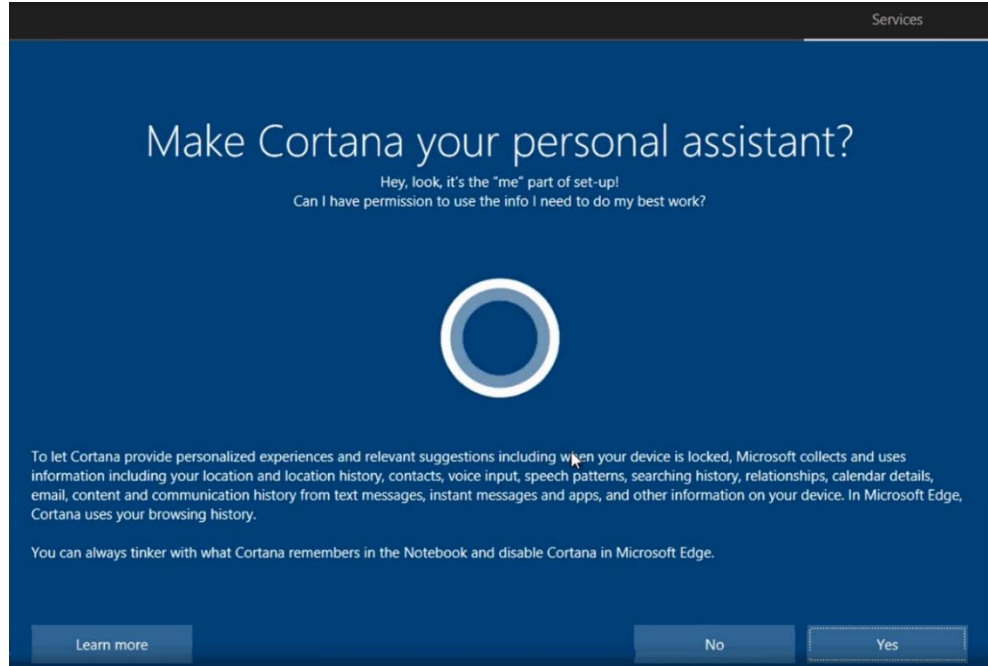
New düğmesine basılarak diskte yeni bir bölüm oluşturulabilir ve bu arada yeni bölümün kapasitesi de belirlenmelidir. İlk bölümde işletim sistemi yerleşir, ikinci bölümde ise kullanıcı verileri saklanır. İkinci bir bölümün oluşturulmasıyla otomatik olarak başka bir sistem bölümünün oluşturulduğunu vurgulayalım.

- (7) İşletim sisteminin kurulumu başlar ve tamamlandıktan sonra bilgisayar yeniden başlatılır.



Şekil 3.10. Windows kurulumun başlaması

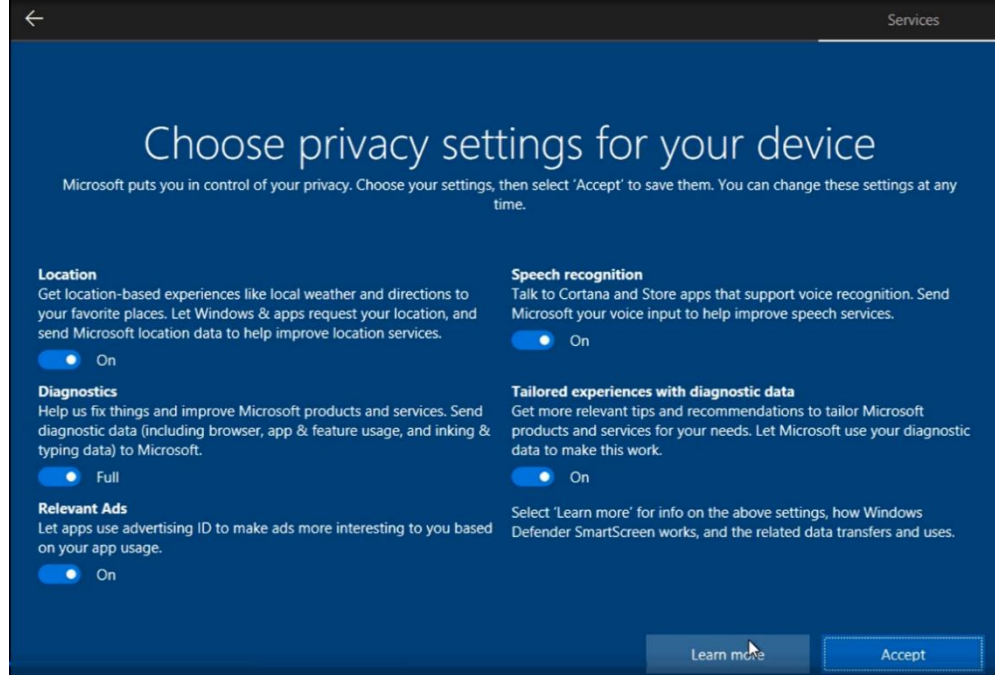
- (8) Devamda birkaç şeyin ayarlanması gerçekleştiği birkaç pencere açılır: ülke veya bölge ve veri girişi için klavye dili.
- (9) Ağ bağlantı penceresini atlayabiliriz ve bu seçimi Windows 10 kurulumundan sonra yapacağız.
- (10) Sonraki iki pencerede kullanıcı adını, şifreyi ve şifreyi hatırlamamıza yardımcı olacak bir ipucunu (İng.hint) tanımlanır. Şifre olmadan işletim sisteminin masaüstünü açamayız. Bazı kurulumlarda kullanıcı adı yanısıra Microsoft profilinin de belirtilmesi gerekir, eğer kullanıcının böyle bir profili yoksa oluşturabilir. Windows 10 işletim sisteminde, kullanıcı verilerini ve dosyalarını otomatik olarak buluta kaydetme seçeneği vardır - OneDrive. Bu şekilde, kullanıcı cep telefonu aracılığıyla verilerine ulaşabilecektir.
- (11) Sondan bir önceki pencere kişisel asistan olarak Cortana uygulamasını seçmek içindir.



Şekil 3.11. Kişisel asistan uygulaması seçimi

Bu uygulama Windows 10'da yenidir, İnternette arama yapmak ve donanım bileşenlerini yönetmek için kullanılır ancak her ülkede mevcut değildir.

- (12) Kurulumun son adımı konum belirleme, yazılım ve uygulama teşhisi, konuşma tanıma vb. gibi özel öğelerin ayarlanmasıdır.



Şekil 3.12. Konum belirleme ve teşhis olanaklarının ayarlanması

Böylece, kurulum tamamlandı ve Windows 10 masaüstü açılana kadar birkaç dakika beklemek gerekiyor.

3.2.3. Uygulamalı Alıştırma: Windows 10 İşletim Sisteminin Yapılandırılması

Windows 10 işletim sistemini kurduktan sonra **cihazları kişisel gereksinimlere göre yapılandırmak** gerekir. Bunu BIOS programını kullanarak veya işletim sisteminin kullanarak yapabiliriz.

BIOS'un donanım bileşenleri ile işletim sistemi arasındaki iletişimi sağlayan özel bir yazılım olduğunu hatırlayalım. Ona, gücü açtıktan sonra F1, F2 veya Del tuşlarından birine basarak erişiyoruz. BIOS açıldıktan sonra ekranın üst kısmında ana ayarlama menüsü (İng.Setup) karşımıza çıkar. BIOS fare ile çalışmayı desteklemez ve navigasyon için klavye tuşları kullanılır: yukarı, aşağı, sol, sağ. Ayarlama menüsü şu alt menülerden oluşur: ana, gelişmiş, güvenlik ve işletim sistemi kurulumu ve gücün açılış alt menüleri. Alt menülerin her birini kısaca tanıyacağız.

Ayarlama menüsünün farklı bilgisayarlarda farklı olduğunu ve anakart türüne bağlı olduğunu vurgulayalım.

Ana alt menü (İng.main)	→	Bu alt menü BIOS sürümü, işlemci türü ve çalışma frekansı, bellek kapasitesi hakkında bilgi sağlar. Sistem saatini ve tarihini değiştirmek için araçlar içerir.
Gelişmiş alt menü (İng. Advanced)	→	Bu alt menü ile anakarta doğrudan bağlanan bellek cihazları (sabit disk veya SSD cihazı), grafik, ses veya ağ kartı için konektörler, USB cihazlar ve diğer cihazlar ayarlanır. BIOS'ta belirli parametrelerin yanlış ayarlanması tüm sistemin arızalanmasına neden olabileceğinden çok dikkatli ve hazırlıklı olmamız gerekiyor.
Güvenlik (İng.Security)	→	BIOS iki parola içerir, gözetici parolası ve kullanıcı parolası. Gözetici parolası, BIOS'un kendisine erişmek için kullanıldığından ayarlama parolası olarak adlandırılır. Kullanıcı parolasına sistem şifresi denir ve işletim sistemine erişim için kullanılır. Güvenlik alt menüsü ayarlama parolasını değiştirmek veya kaldırmak için araçlar içerir. Şifrenin unutulması durumunda tüm şifreler anakartına özel bir jumper kablo kullanılarak silinebilir.
İşletim sistemini ayarlama alt menüsü (İng. Boot)	→	Hatırlayalım, bu alt menü Windows işletim sistemi kurulum cihazını sıralamak ve seçmek için kullanılır.
Çıkış (İng. Exit)		Bu alt menü, yapılan değişiklikleri kaydetmek ve BIOS programından çıkmak için kullanılır.

Windows 10 iki yapılandırma aracı içerir: Control Panel ve yeni Settings uygulaması. Aşağıda Settings uygulamasındaki kategoriler aşağıda listelenmiş ve açıklanmıştır.

Sistem (İng.System)	→	Bu kategorideki araçlarla ekranın çözünürlüğünü ve parlaklığını değiştirebilir, veri depolama alanında yer serbestlenebilir, güç düzenini seçebilir vb.
--------------------------------	---	---

Cihazlar (İng.Devices)	→	Bu kategori bilgisayarın iletişim kurabileceği cihazların listesini verir: Bluetooth cihazları, kablosuz ekranlar, dokunmatik ekranlar, grafik tabletler, yazıcılar vb.
Telefon (İng.Phone)	→	Cep telefonumuzdaki Android veya iPhone içeriğine erişim sağlar.
Ağ ve İnternet (İng.Network & İnternet)	→	Wi-Fi veya Ethernet ağ adaptörü seçilebilir, uçak modu açılabilir veya kapatılabilir, yeni sanal özel ağ (VPN) eklenebilir vb.
Kişiselleştirmek (İng.Personalization)	→	Her profil için arka plan seçilebilir, temanın renklerini değiştirebilir, ekran koruyucu resmini değiştirebilir (screen saver), başlat menüsünün veya çalışma çubuğunun görünümünü değiştirilebilir.
Uygulamalar (İng.Apps)	→	Bu kategori programları kontrol ediyor: başlangıç (İng. start-up) programlarının silinmesi, değiştirilmesi, standart programların değiştirilmesi, video oyunlar ekranı özellikleri (İng.video mode).
Kullanıcı hesabı (İng.Account)	→	Kullanıcı profilleri sadece yönetici oluşturabilir. Windows 10'da aile koruması ve gözetim olanakları çoktur.
Saat ve Dil (İng.Time & Language)	→	Saat, tarih, bölge, dil ve konuşma seçenekleri ayarlanır.
Kolay erişim (İng. Ease of access)	→	Bu kategori ekrana, klavyeye, fareye, ekrandaki öğelerin sesli açıklama konuşmacıya farklı erişim seçenekleri sunar.
Kişisel sanal yardımcısı (İng.Cortana)	→	Bu yeni bir kategoridir ve internette arama yapmak ve bileşenleri kullanmak için kişisel sanal yrdımcısıdır.
Gizlilik, Güncelleme ve Güvenlik (İng.Privacy, Update)	→	İşletim sistemini yükseltmek, kişisel belgelerin yedek kopyasını oluşturmak (İng. backup), virüsleri ve potansiyel olarak zararlı yazılımları taramak ve tespit etmek, İnternet üzerinden yetkisiz erişime ve suistimale karşı koruma, aile seçenekleri için

& Security)

kullanılır. Örneğin, Privacy → Camera kategorisinde dizüstü bilgisayarın yerleşik kamerasına erişimi olacak programlar seçilebilir.

Zamanla, işletim sisteminin performansları çeşitli nedenlerden dolayı azalır: uyumsuzluk, virüsler, kullanılmayan uygulamalar vb. **Optimizasyonun amacı işletim sisteminin hızını arttırmaktır.** Aşağıda bu hedefe ulaşmanın birkaç yolu, bunların anlamları ve gerçekleştirme adımları verilmiştir.

Başlangıç uygulamalar (İng.Startup)

→ Başlangıç uygulamaları bilgisayar açıldığında otomatik olarak etkinleştirilir ve bekleme süresini artırır. Bu nedenle sayılarını azaltmak gerekir.

→

- Settings
- Apps
- Startup
- Sort by: Startup impact
- Uygulamaların seçimi
- Bilgisayarın tekrar başlatılması

Kullanılmayan uygulamalar

→ Windows işletim sistemleri sistemin yüklenmesi hakkında bilgi (bildirim) sağlamaz, bu nedenle kullanıcıların kullanmadıkları programları kendileri kaldırmaları gerekir.

→

- Settings
- Apps
- Apps & feature
- Uygulamaların seçimi
- Uninstall

Sabit diskteki belleği boşaltma

→ Genellikle sabit diskteki bellek alanının %70 kullanımına ulaşıldıktan sonra yapılır. Kontrol panelinde (Control Panel) bu olasılık, sistem yönetim araçları (System-Administrative Tools) kategorisi altındadır.

→

- Settings
- System
- Storage
- Temporary files
- Dosyaların seçimi
- Remove files

Sabit diskteki verilerin organizasyonu (İng. defragmentations)

→ Sabit diskte boş alan azaldığında veriler sabit diskin farklı yerlerine yazılır, böylece verileri okuma süresi artar.

→

- Settings
- System
- Storage
- More settings

		Birleştirme ile bu veriler yan yana yerleştiriliyor ve bellek alanında yer açılır.		<ul style="list-style-type: none">• Optimize Drivers• Disk seçiyoruz (C: veya D:)• Optimize
Antivirüs programları	→	İşletim sistemi düşük hızda çalışıyorsa antivirüs programını çağırmak ve etkinleştirmek gerekir. Eğer bilgisayarda özel bir antivirüs programı yoksa Windows Defender programı kullanılabilir.	→	<ul style="list-style-type: none">• Control Panel• Security• Windows Defender• Scan
İşletim sisteminin yükseltilmesi	→	Kullanıcıların otomatik yükseltmeyi kapatmaları yanlıştır çünkü yükseltmeler sistem güvenliği için çok önemlidir. Cihaz sürücüleri Sistem-Device manager kategorisinde yükseltilebilir.	→	<ul style="list-style-type: none">• Settings• Update & Security• Windows Update• Dcheck for Update• Bilgidsayarın yeniden başlatılması
Sanal sayfaların artışı	→	Sanal bellek, birincil belleğin ikincil bellek yoluyla genişletilmesi demektir. Yani RAM belleği, verileri sabit diskten sabit boyutta sanal sayfalar halinde alır ve bunları sanal bir çerçeveye ekler. Sanal sayfaların kapasitesi artarsa RAM ile sabit disk arasındaki aktarım sayısı azalacaktır.	→	<ul style="list-style-type: none">• Control Panel• System• Advanced System (Performance)• Settings• Advanced• Change (Virtual memory)

Windows 10 işletim sistemi birçok olasılık sunmaktadır ve bunların hepsini açıklayıp inceleyemeyiz. Ancak, her zaman bilgisayar sistemlerini kendiniz araştırmalı ve yapılandırmaya çalışmalısınız.

3.2.4. Uygulamalı Alıştırma: Ubuntu 17 Linux İşletim Sistemini Sanal Makineye Yükleme

3.2.4.1. Sanal Makine Oluşturma Kılavuzu

Ubuntu, Linux işletim sistemlerinin **oldukça popüler dağıtıcısıdır**. İşletim sisteminin kurulumunu indirmek için bu dağıtıcının www.ubuntu.com adresinin açılması gerekiyor ve ardından Download seçeneğine tıklayarak masaüstü sürümünü seçiyoruz.

VirtualBox, sanal makineleri oluşturmak, yapılandırmak ve yönetmek için bir uygulamadır. Sanal makine, ana bilgisayarın (host) kaynaklarını kullanan bilgisayar sistemi konğudur (guest). Ana bilgisayara Windows işletim sistemi kurulacaktır, sanal makine yani konuk bilgisayara ise Ubuntu işletim sistemi kurulacaktır. Böylece, **aynı bilgisayarda** birbirini etkilemeden **iki işletim sistemi** kurmuş olacağız.

Sanal makine kurma programı www.virtualbox.org sitesinden Download seçeneğine tıklayarak ve Windows ana bilgisayar bağlantısını seçerek indirilir.

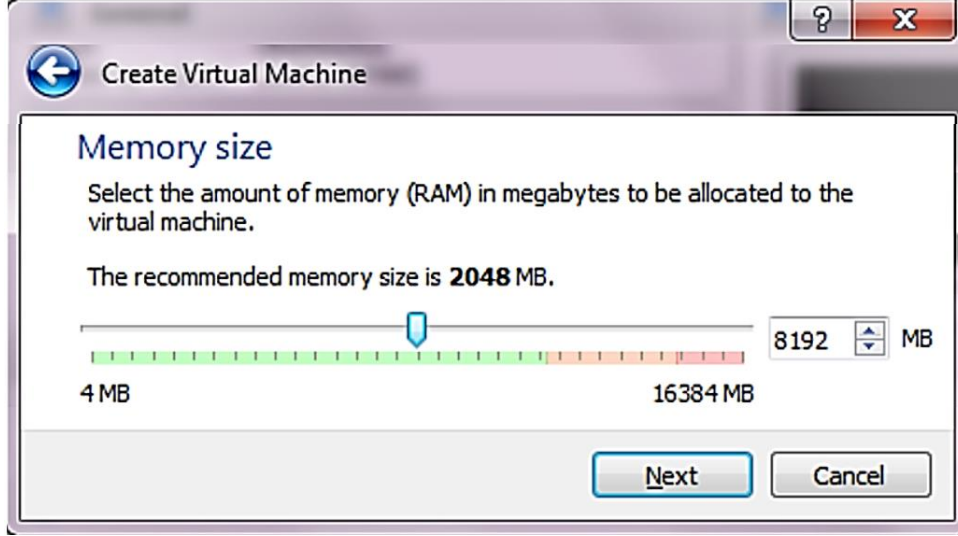


Şekil 3.13. Sanal makinenin oluşturulması

VirtualBox uygulamasını yükledikten sonra açıyoruz ve yeni bir sanal makine oluşturmak için New seçeneğine tıklanıyor. **Create Virtual Machine** penceresi açılır. **Name** alanına sanal makinenin adı giriliyor (örneğin Ubuntu), **Type** alanına sanal

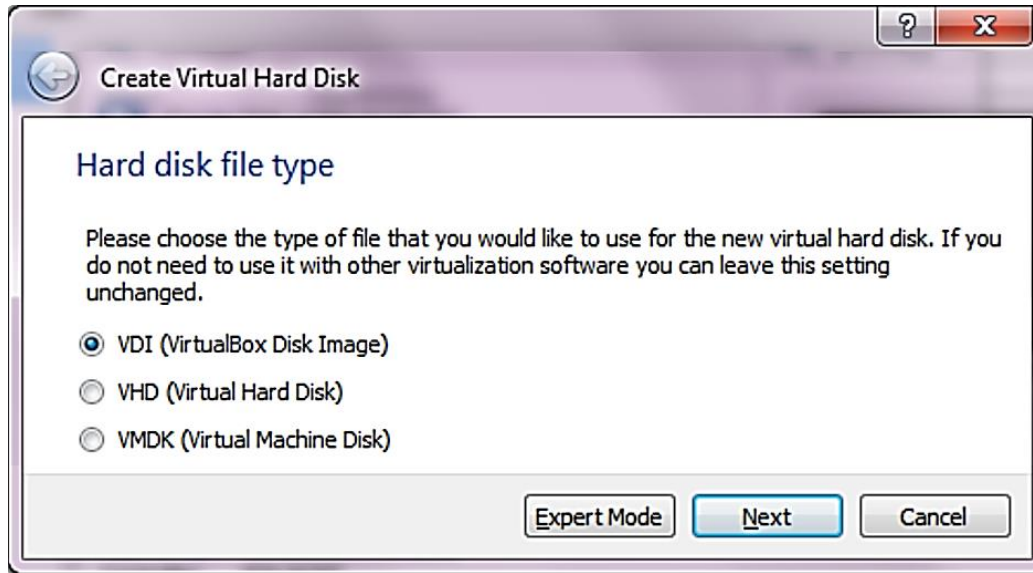
makineye (Linux) kurulacak işletim sisteminin türü girilir ve **Version** alanına işletim sisteminin sürümü giriliyor (32 bit veya 64 önemli). Sonunda Next'e tıklanıyor.

Bir sonraki pencere sanal makinenin RAM belleğinin kapasitesini belirlemek için kullanılır. En iyisi önerilere uymaktır, ancak toplam bellek kapasitesi de göz önüne alınmalıdır.



Şekil 3.14. Sanal makine için RAM bellek kapasitesinin belirlenmesi

Sıradaki üç pencere **sanal sabit disk** oluşturmak, kapasitesini tanımlamak ve bellek alanının yönetileceği şekli belirlemek için kullanılır.



Şekil 3.15. Sabit disk türü seçimi



Şekil 3.16. Sabit disk kapasitesi seçimi

Sanal makinenin oluşturulmasını Create'e tıklayarak onaylıyoruz. Bu son adımda sanal makinenin adını ve konumunu değiştirme olanağımız var.

3.2.4.2. Ubuntu İşletim Sisteminin Kurulumu

Ubuntu işletim sistemini kurmak için Şekil 3.17'de gösterilen VirtualBox Manager penceresinde Start seçeneğine tıklıyoruz.



Şekil 3.17. Ubuntu işletim sistemi kurulumunun başlatılması

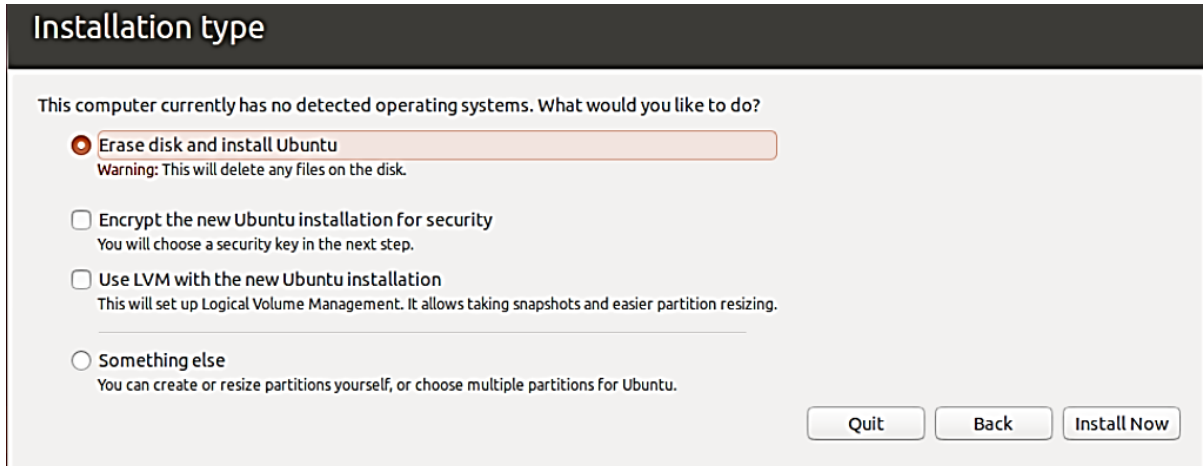
Select start-up disk penceresinde sarı klasöre (dosyaya) tıklayarak işletim sistemi kurulumunun bulunduğu konumu bulup seçiyoruz ve Start düğmesine tıklıyoruz.

Welcome penceresinde Ubuntu işletim sisteminin **kurulması için iki yöntem** sunulmaktadır. Try Ubuntu yöntemi ile işletim sistemi kurulum yapılmadan kullanılabilir. Bu şekilde, aslında işletim sistemini daha önceden oluşturulmuş ve kurulumdan önce boş olan sanal sabit disk üzerine kuruyoruz, dolayısıyla mevcut işletim sisteminin silinmesi söz konusu olmuyor.



Şekil 3.18. Ubuntu işletim sistemi kurulum yönteminin seçimi

Installation type penceresinde Install Now düğmesine tıklayarak **kurulum başlatılır**. **Erase disk and install Ubuntu** seçeneği önceden işaretlenmiştir ancak sanal bir sanal sabit disk olduğu için herhangi veri silinmesi ve kaybı olmayacağını daha önce açıklamıştık.

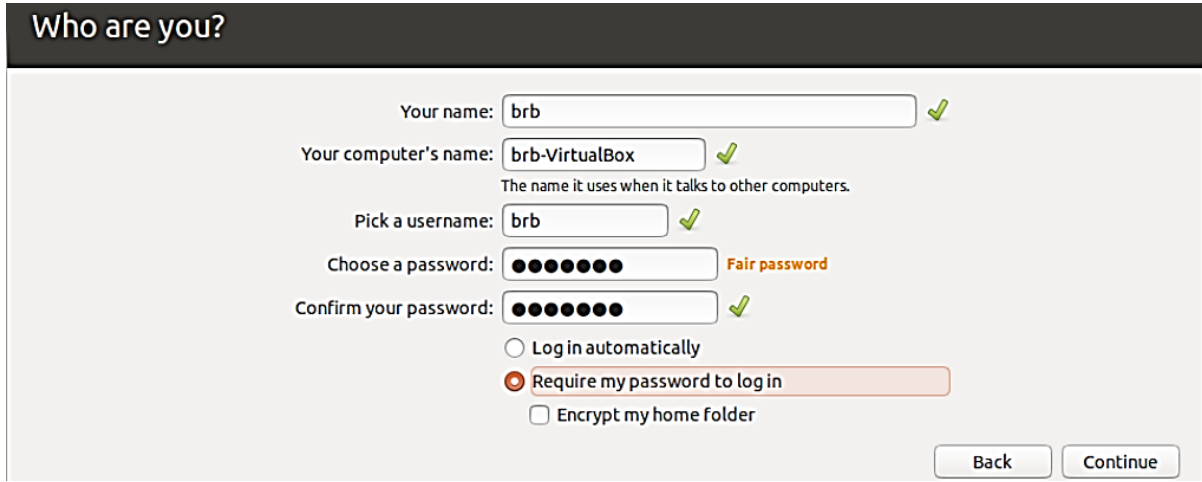


Şekil 3.19. Sanal makine olduğundan dolayı mevcut işletim sistemi silinmeyecektir.

Bir sonraki pencere Preparing to Install Ubuntu penceresidir ve içinde işletim sisteminin otomatik olarak yükseltme ve multimedya için ek multimedya yazılımının kurulumunu seçme olanağı var. Son olarak Continue düğmesine tıklıyoruz.

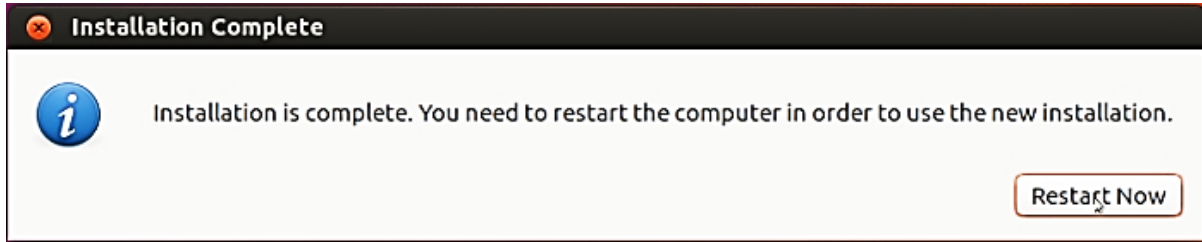
Sonraki iki pencere, Windows 10 işletim sisteminin kurulumuna benzer şekilde ülke veya bölge ve veri girişi için klavye dilini seçmek içindir.

Devamda kullanıcı adını ve şifreyi belirlemek için bir pencere açılır. Continue'ya tıklayarak kurulum başlıyor.



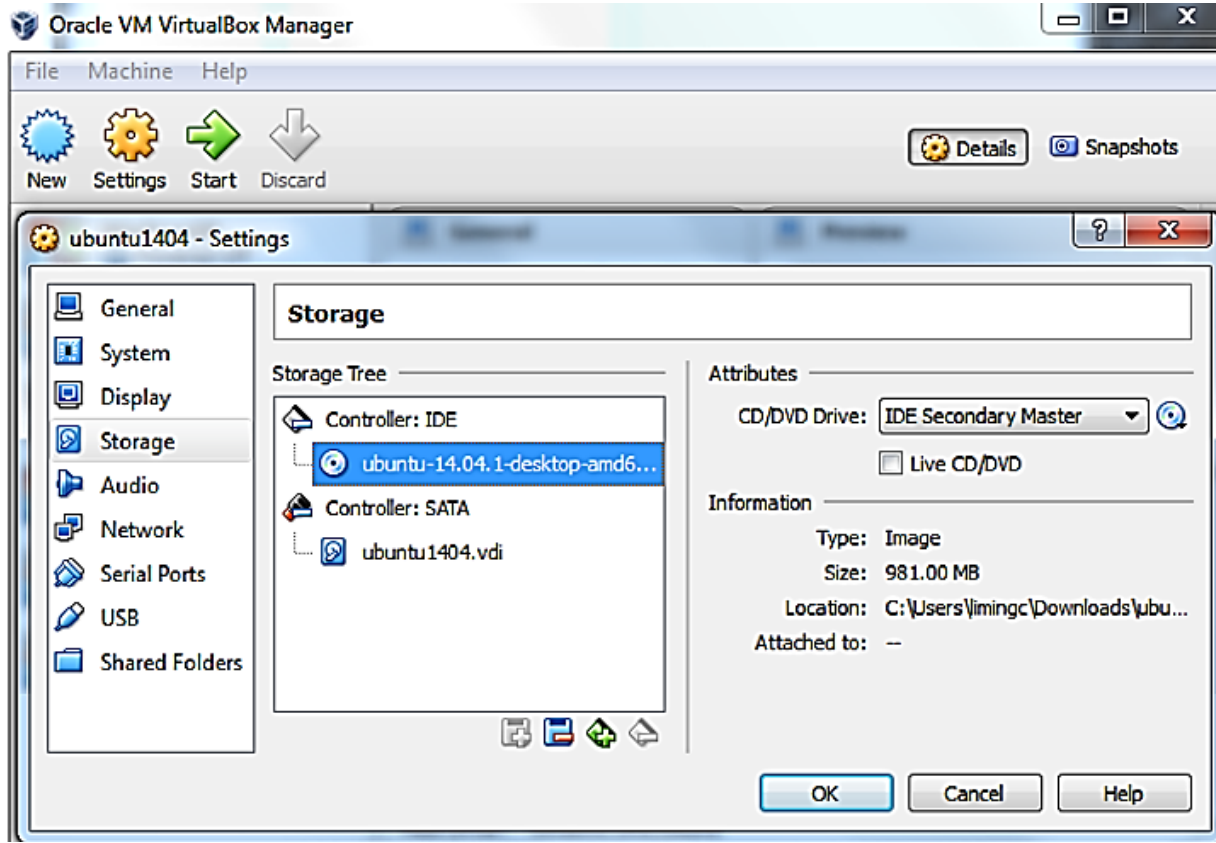
Şekil 3.20 Kullanıcı adı ve şifre seçimi

Başarılı kurulumla ilgili bilgilendirmenin ardından sistemi yeniden başlatıp karşımıza Ubuntu çalışma alanı çıkıyor.



Şekil 3.21. Ubuntu çalışma alanını elde etmek için bilgisayarın yeniden başlatılması

Kurulum gerçekleştikten sonra kurulum dosyasını silmemiz gerekiyor. Bunun için sanal makinede Ubuntu işletim sistemi seçiliyor, Settings kategorisinde Storage seçeneğine tıklanır. Ardından Storage Tree bölümünde kurulum dosyasının seçilmesi gerekir ve Remove selected storage attachment seçeneğine tıklanır (İng. Click). Son olarak OK düğmesine basıyoruz.



Şekil 3.22. Kurulum dosyasının silinmesi

3.2.5. Uygulamalı Alıştırma: Raspberry Pi İçin Raspbian İşletim Sisteminin Kurulumu ve Yapılandırması

Raspberry Pi, Arduino'dan farklı olarak işletim sistemine sahip bir mikrobilgisayardır. İşletim sistemi çeşitli uygulamaların kurulumuna ve yönetimine izin verir, bu da Raspberry Pi'yi yazılım açısından çok güçlü yapar. Raspberry Pi için birçok işletim sistemi türü vardır, ancak başlangıç için **Raspbian işletim sistemini** kullanacağız. [7]

3.2.5.1. NOOBS Kurulum Yazılımının İndirilmesi

Raspbian işletim sisteminin kurulumu için (İng.NOOBS-New Out Of The Box Software) kutudan Yeni yazılımı sembolik isimli yazılım kullanacağız. Bu, birkaç işletim sisteminden birini seçme ve birkaç fare tıklamasıyla otomatik kurulum olanağı sağlayan özel bir yazılımdır. Kurulum yazılımı mikro SD - kartta saklanır. Raspberry Pi'nin

yerleşik bir kalıcı belleğe sahip olmadığını ancak verileri ve işletim sistemini saklamak için **minimum 16 GB belleğe sahip SD kart** kullanıldığını önceden belirtmiştik.

NOOBS yazılımını indirmek için web tarayıcısında rpf.io/download yazılması gerekir. Açılan sayfada, Download kategorisinde ilk önce Raspberry logolu NOOBS'a tıklanır ve ardından "NOOBS offline and network install" altında bulunan Download Zip'e basılır (click).

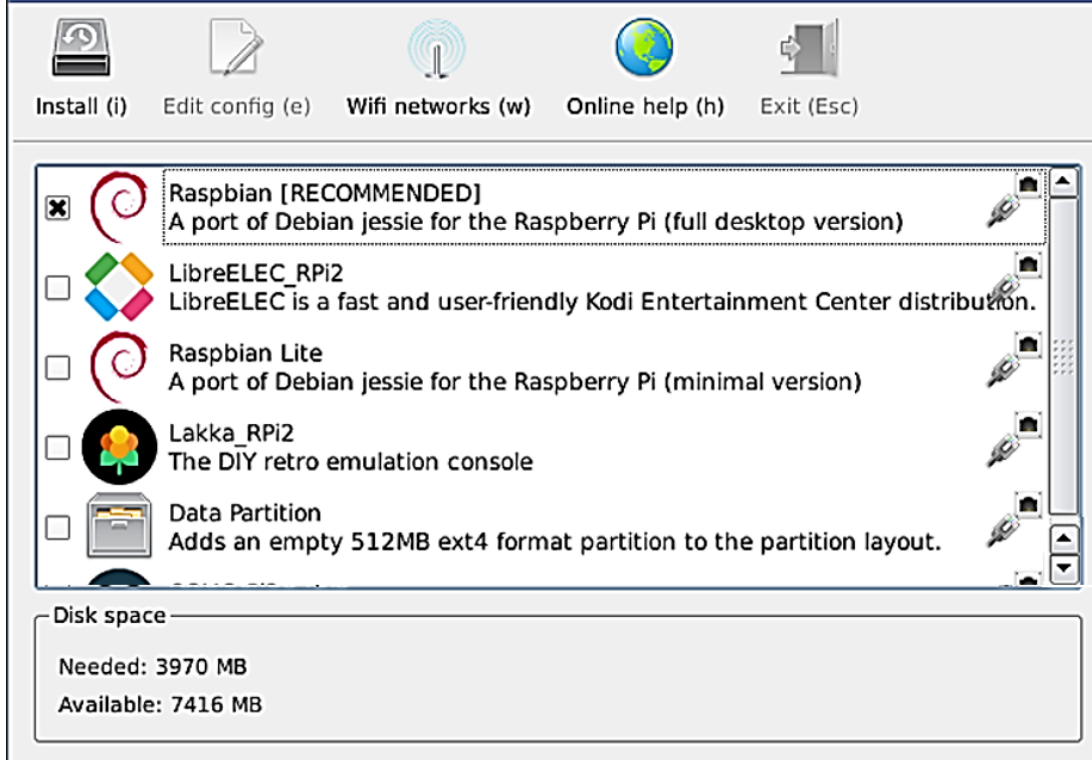


Şekil 3.23. NOOBS kurulum belgesini indirmek için web sayfası

Mikro SD kartı kişisel bilgisayarın uygun yuvasına takıyoruz. Yeni kart değilse üzerinde yazdırılmış veriler dikkate alınarak formatlanması gerektiğini hatırlatalım. Ardından İndirilenler (İng.Downloads) klasöründe sıkıştırılmış kurulum dosyasını arıyoruz. Bu dosya arşiv dosyası (İng.archive) olarak bilinir ve çok sayıda ayrı belge içerir. Çift tıklamayla sıkıştırılmış belgeyi "paketinden çıkarırız", içindeki tüm belgeleri seçiyoruz ve mikro SD karta aktarıyoruz, kopyalıyoruz.

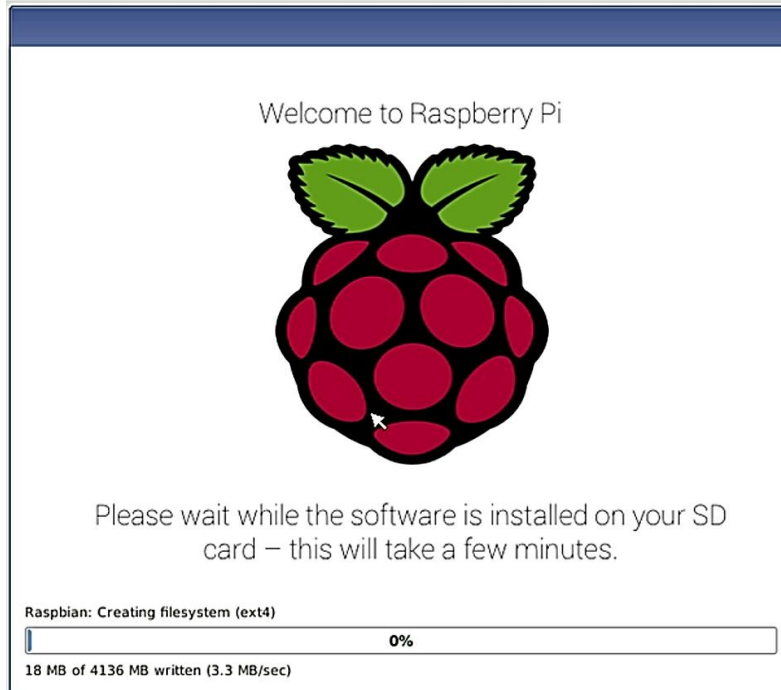
3.2.5.2. Raspbian İşletim Sisteminin Kurulumu

NOOBS yazılımının yüklü olduğu **mikro SD kartı kişisel bilgisayardan çıkar-mamız ve Raspberry Pi mikrobilgisayarının yuvasına takmamız gerekiyor**. Kartı Raspberry Pi'ye ilk bağladığımızda Şekil 3.24'te gösterilen NOOBS menüsü karşımıza çıkıyor. Fare ile kurmak istediğimiz işletim sisteminin, bu durumda **Raspbian** kutusunu **işaretliyoruz**. Install simgesinin renk değiştirdiğini fark edeceğiz, bu da işletim sisteminin kurulumuna hazır olduğu anlamına geliyor.



Şekil 3.24. Raspbian işletim sisteminin seçimi

İnstall simgesine basıldığında kurulum işlemi başlıyor ve bu arada NOOBS yazılımı dışındaki tüm veriler mikro SD karttan silinir. Kurulum 10 ile 30 dakika arasında sürebilir.



Şekil 3.25. Raspbian işletim sisteminin kurulumu

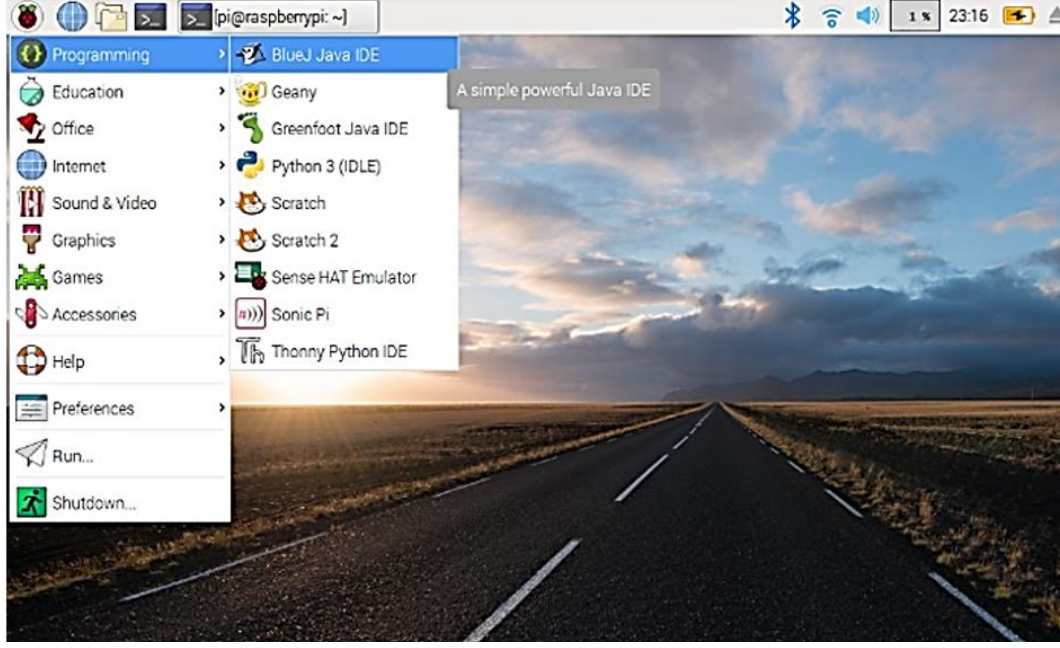
Kurulum tamamlandıktan sonra OK'ye tıklıyoruz (İng.clik) ve Raspberry Pi cihazı yeniden başlatılır. İşletim sisteminin sadece ilk açılışında, bir ila iki dakika süren boot (önyükleme) mesajları görünür. Sonunda Raspbian çalışma alanı karşımıza çıkar ve içinde işletim sisteminin yapılandırmasını bildiren pencere görünür. Next'e tıklıyoruz.



Şekil 3.26. Kurulum için giriş penceresi

Windows veya Linux işletim sistemlerinin kurulumunda olduğu gibi, Raspbian işletim sisteminin kurulumu sırasında da şunları seçiyoruz: ülke, dil, saat dilimi, şifre, Wi-Fi ağı seçimi ve işletim sisteminin yükseltilmesi. Son pencerede Reboot'a basılmalı, ardından sistem yeniden başlatılır ve yapılan değişikliklerle yapılandırılır.

Raspberry işletim sisteminin çalışma alanı (masaüstü), Windows işletim sisteminin çalışma alanına çok benzerdir. Sağ üst köşede şunlar içeren araç çubuğu bulunur: Bluetooth, ağ seçimi, ses seviyesi, işletmecisi gözetimi, saat, USB cihazlarının açılması ve kapatılması. Sol üst köşede Raspberry bulunur ve ona basıldığında birçok kategori içeren açılır menü açılır. Örneğin, **Programlama** kategorisi, **Thonny Python IDE** dahil geliştirme programlarını içerir. İnternette arama yapmak için İnternet kategorisinde Chromium Web Browser'i seçiyoruz. Belgeler ve dosyalarla çalışmak için File Manager gerekir ve onu etkinleştirmek için Accessories kategorisini seçiyoruz. Oficce kategorisinde metin işleme için LibreOffice Writer programı bulunur. Ek yazılımın yüklenmesi Preferences kategorisinde Recommended Software seçeneğine tıklayarak yapılabilir. Yeni donanım eklemek için (örneğin, kamerayla çalışma yazılımı, İng. Raspberry Pi Camera Module) veya ağa bağlanmak için (İng. VNC- Virtual Network Computer), Preferences kategorisinde Raspberry Pi Configuration tuşuna basarak (click) ulaştığımız İnterface seçeneği kullanılabilir.



Şekil 3.27. Raspbian işletim sisteminin çalışma alanının (masaüstünün) görünümü

3.2.6. Uygulamalı Alıştırma: Arduino Platformu İçin Geliştirme Ortamının Kurulumu ve Kullanıma Koyulması

3.2.6.1. Arduino Platformu için Geliştirme Ortamı Kurulum Kılavuzu

Arduino platformunun işletim sistemi yoktur ve bu nedenle **kişisel bilgisayara bağlamadan** programlayamayız. Arduino platformunun mikrodenetleyicisine program yazmak ve girmek için **geliştirme ortamı** (IDE - Integrated Development Environment) olarak da bilinen geliştirme programına ihtiyacımız var. Geliştirme ortamının daha iyi anlamak için bu tür programların kapsamına giren birkaç araç türünü açıklayacağız:

- Editörler, program metninin girilmesini, değiştirilmesini, şekillendirilmesini, kaydedilmesini ve yazdırılmasını sağlayan programlardır;
- Çeviriciler, üst düzey programlama dillerinde yazılmış programları makine diline çevirir. Çeviriciler derleyiciler adıyla da bilinir.

- Hata ayıklayıcılar hataları düzelten programlardır. Programın belirli yerlerine kontrol noktaları yerleştirilir ve program sırasında bu noktalarda durup programın durumu incelenir;
- Bağlayıcılar, yeni yazılımı eskisi ile bağlayan programlardır ve bu, programın yürütülmesi için çok önemlidir.

Geliştirme programı **resmi Arduino web sitesinden**, yani <https://www.arduino.cc/en/software> bağlantısından indirilir. Arduino geliştirme programlarının birkaç farklı sürümünden birini seçebiliriz ancak Windows 10 işletim sistemi kullanıyorsak en son sürümü indirmek en doğrudur. Bu sürümün Windows XP veya Windows 7 gibi daha eski işletim sisteminde çalışmayacağını vurgulayalım. Kurulum dosyasını indirdikten sonra geliştirme programını kuruyoruz. “**Run**”a basıyoruz (click), lisans koşullarını kabul ediyoruz, programı kaydedeceğimiz klasörü seçiyoruz ve sonunda “Install”a basıyoruz. [6]

Kurulumun ardından, Arduino Uno R3 platformunu A ve B türünden iki farklı konektör ile USB kablo aracılığıyla kişisel bilgisayara bağlıyoruz. Platform, kişisel bilgisayardan güç alır ve L harfiyle işaretlenmiş yerleşik olan yeşil diyot yanar (Şekil 2.10.). Arduino Uno R3’ü bilgisayara ilk bağladığımızda **sürücü otomatik olarak kuruluyor**. Sürücü, bilgisayarın yeni cihazla iletişim kurmasını sağlayan yazılımdır.

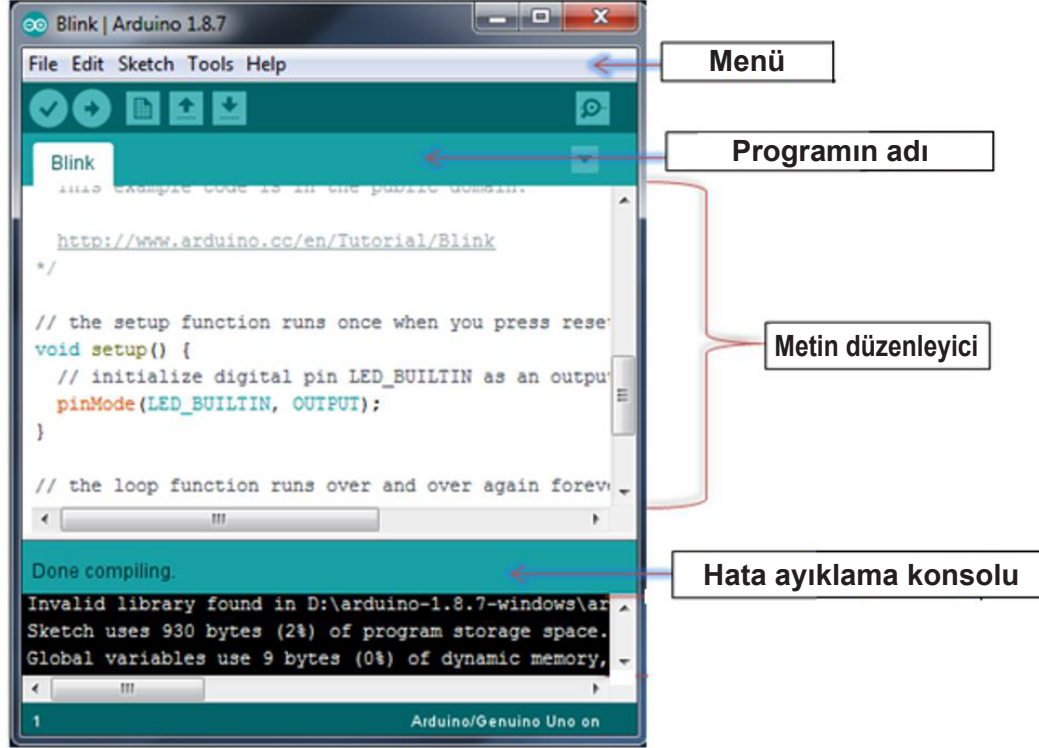
Windows işletim sisteminin daha eski bir sürümü söz konusu olunca, sürücünün yüklenmesi gereken konumun belirtilmesini arayan pencere görünebilir. Kurulum başlamazsa, **Control Panel** programını açmak, **Device Manager** kategorisini seçmek, ardından “**Other Devices**” veya “**Unknown Devices**” alt kategorisini seçmek ve “**Update Drivers**” veya “**Update Driver Software**” tuşuna basmak (İng. click) gerekir. Ardından Arduino platformunun bağlanacağı sürücüyü belirliyoruz, böylece Arduino klasörünün içeriğindeki “Drivers” klasörü seçiliyor.

Başarılı kurulumun onayı olarak, Device Manager kategorisinde “PORT (COM&LPT)” seçeneğine basınca Arduino Uno’nun ortaya çıkması gerekir. Arduino Uno platformunun hangi porta bağlı olduğunu bilmiyorsak platformun bilgisayarla olan bağlantısını kesip tekrar açarak seri portlardan hangisinin yeni görüneceğini görebiliriz.

Arduino platformu açık kaynaklı program kodları kullanıyor, bu da programların ücretsiz olduğu anlamına geliyor. Her programcı kendi programlama kodlarını yayınlabilir ve bu şekilde bu platformun geliştirilmesinde katkıda bulunabilir. Arduino geliştirme ortamını birçok hazır program içerir ve bu programların listesini File → **Examples**’a basarak görebiliriz. Ayrıca hazır bir programı Arduino web sayfasından da indirebiliriz ve File → Open → Look’a basarak ve sonunda programı kaydettiğimiz klasörden açabiliriz.

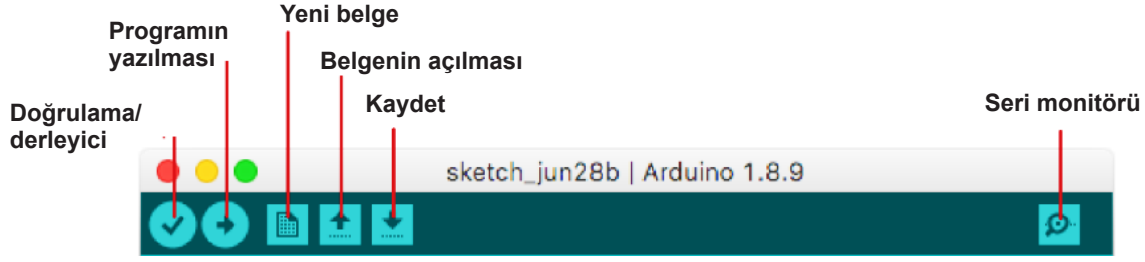
3.2.6.2. Geliştirme Ortamının Menüsü ve Araçları ve Arduino Uno R3'te Programın Yazılması

Şekil 3.29'da, Arduino platformu için geliştirme programı gösterilmektedir. Daha önemli bileşenler şunlardır: menü, araçlar çubuğu, metin düzenleyici, seri monitörü, kütüphaneler ve hata ayıklama konsolu.



Şekil 3.29. Arduino geliştirme platformu için geliştirme ortamının görünümü

Menü	→	Menüde şu kategoriler bulunmaktadır: File, Edit, Sketch, Tools, Help.
Araçlar çubuğu	→	Araçlar çubuğunda şu düğmeler vardır: doğrulama/derleyici (İng. Verify/Compile), yükleme yani yazma (İng Upload), yeni belgenin açılması (İng.New), eski belgenin açılması (İng.Open) ve kaydet (İng.Save). Doğrulama aracı ile olası hataları tespit etmek için kod satır satır kontrol edilir. Upload aracı ile doğrulanan kod mikrodenetleyici platformunun program belleğinde yazılır.

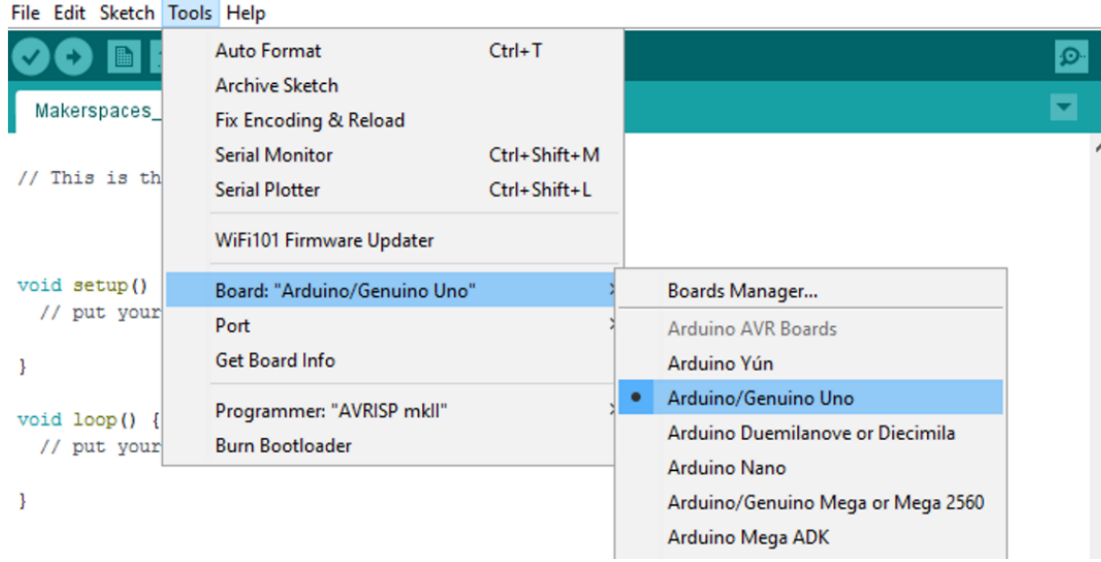


Şekil 3.28. Arduino geliştirme ortamının araçlar çubuğu

Metin düzenleyici	→	Bu, geliştirme programının en önemli parçasıdır çünkü burada C/C++ programlama dilinde kodları, programları yazıyoruz.
Seri monitör	→	Buna basarak, Arduino platformundan gönderilen veya alınan verilerin izlenmesi sağlanır. Gerçek zamanlı uygulamalarda geniş uygulaması vardır.
Hata ayıklama konsolü	→	Bu pencere bize başarılı derleme hakkında bilgi verir veya derlemede hataya neden olan satırın sıra numarasını gösterir. Derleme, programın üst düzey programlama dilinden makine diline çevrilme işlemidir.
Kütüphaneler (libraries)	→	Bu kategoriye basıldığında (İng. click) çeşitli giriş-çıkış cihazlarıyla çalışma için alt programların listesi açılır.

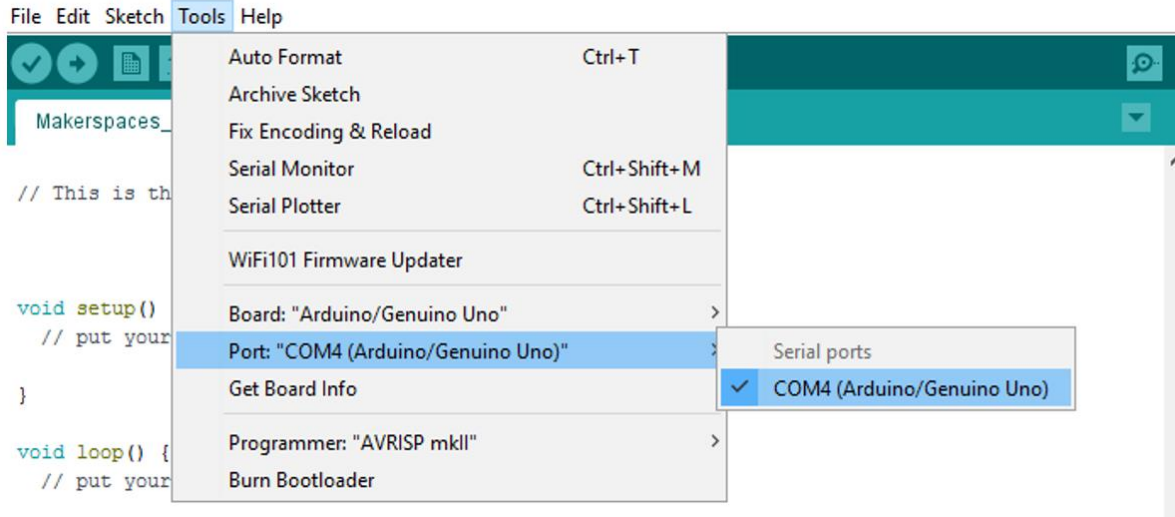
Geliştirme ortamını kurduktan ve menü ve araçları tanıdıktan sonra, Arduino Uno R3 platformunun belleğine program yazma sürecini açıklayacağız. Bu amaçla bu geliştirme platformunun 13'ncü pinine bağlanan yerleşik LED diyotu kullanacağız. Ayrıca başka elektronik bileşenlerin kullanımını gerektirmeyen hazır bir program kullanacağız. Bu alıştırma en basit alışırtmalardan biridir ve Arduino Uno R3'ün çalışması için bir tür test temsil etmektedir.

1. Kurulumdan sonra Arduino Uno R3'ü A ve B türünden iki farklı konektöre ile USB kablosu aracılığıyla kişisel bilgisayara bağlıyoruz.
2. İlk adım, mikrodenetleyici platformu ve seri portu (bağlantı noktası) seçmektir. Hatırlayalım, Arduino ailesi b,rden fazla farklı platformdan oluşuyordu: Uno, Mega, Nano, Leonardo vb. Platformun seçimi için Tools → Board → Arduino Uno seçeneklerine basılmalıdır. Bu, Şekil 3.30'da gösterilmiştir.



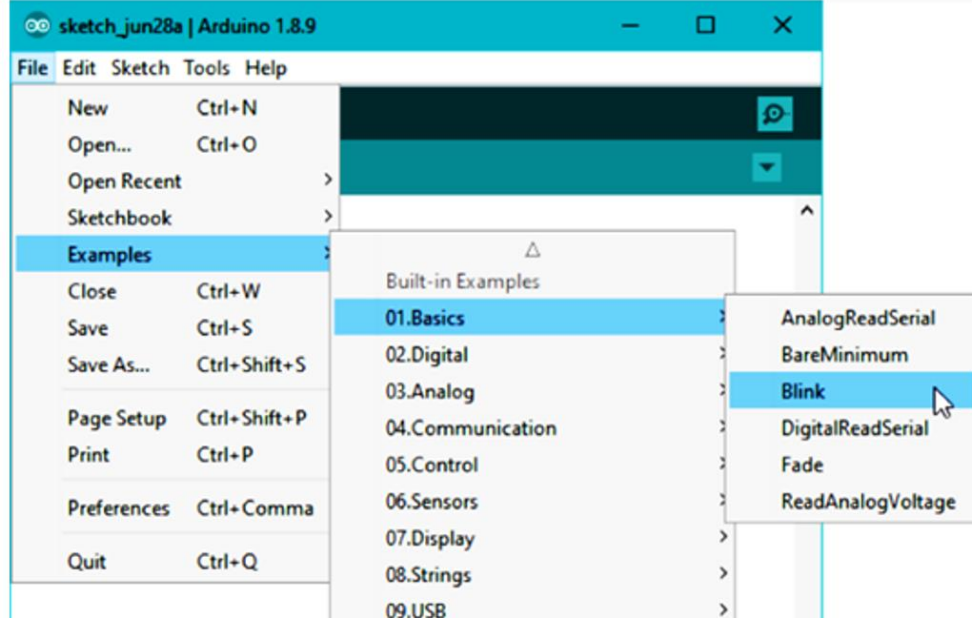
Şekil 3.30. Arduino Platform seçme işlemi

3. Port (bağlantı noktası) seçmek için Tools → Serial Port seçeneğine basılmalıdır. Bu, Şekil 3.31’de gösterilmektedir. Bu arada, kullanılabilir seri bağlantı noktalarının açılır menüsünü karşımıza çıkacaktır ve bu, bilgisayara bağlı cihazlara bağlı olarak farklıdır.



Şekil 3.31. Arduino Uno için bağlantı noktası seçme işlemi

4. Program kodunu File → Example seçeneğine basarak buluyoruz ve ardından 01. Basic → Blink seçeneğini seçiyoruz. (resim 3.32.) ,
5. Programı Arduino Uno R3 belleğine girmeden önce makine diline çevrilmesi yani derlenmesi gerekmektedir. Bunun için, araçlar listesindeki Verify/Compile düğmesine basıyoruz veya menüden Sketch → Verify/Compile seçeneğini seçiyoruz.



Şekil 3.32. Arduino Uno R3 platformu için Blink programının açılması

6. Hata ayıklama konsolunda “Compiling Sketch...” bildirimini alıyoruz ve işlemin ilerleme çubuğu ortaya çıkıyor ve sonunda “Done Compiling” bildirimi alınmalıdır (Şekil 3.33.).



Şekil 3.33. Başarılı derleme bildirimi

7. Program kodunda hata yoksa, hata ayıklama konsolunda açtığımız programın bayt olarak büyüklüğü ve kalan boş bellek alanının boyutu hakkında bilgi alırız.
8. Yükleme (İng.Upload) düğmesine basılmasıyla program belleğe yazılmaya başlıyor. L işaretli yerleşik LED diyotun yanması durdurulacak ve seri iletişim için RX ve TX işaretli iki diyot yanmaya başlayacaktır (Şekil 2.10.'a bakınız). Yazma işlemi birkaç saniye sürer ve ardından L işaretli LED diyotu yanıp sönmeye (İng.blink) başlar.

Sonuçlar

İşletim sistemi kontrol yönetim yazılımı adıyla da bilinir. Yazılım, donanım kaynaklarının dağıtımı, giriş-çıkış işlemlerinin kontrolü, bellek yönetimi, veri yönetimi, programlama dillerinin çevirisine ilişkin programları içerir.

Windows işletim sisteminin avantajları şunlardır: kullanımı basit (İng. user friendly), çevresel cihazlarla bağlantı için en büyük destek, uygulama yazılımının kullanılabilirliği, otomatik kurulum (plug and play), video oyunları için en iyi performanslar, web sayfalarının çoğu ile uyumluluk. Dezavantajları ise şunlardır: Pahalı bilgisayar yapılandırma ihtiyaçları, kapalı kod, İnternet saldırıları ve yönetimsel koruma açısından zayıf güvenlik, bilgisayar virüslerine karşı duyarlılık, lisans gerekliliği, zayıf uzman desteği ve çoğu zaman çalışma hızının artması için işletim sisteminin yeniden kurulmasının gerekliliği.

Açık veya ücretsiz kod, tüm kullanıcıların kaynak koduna erişebilmesi ve bunu kendi ihtiyaçlarına göre değiştirebilmesi anlamına gelir.

Google şirketinin Android'i ve Raspbian işletim sistemi, Linux işletim sistemi çekirdeğinin uygulanmasına ve genişletmesine ilişkin örneklerdir.

Android işletim sisteminin avantajları şunlardır: çok sayıda ücretsiz uygulama, farklı üreticilerin cihazlarıyla uyumluluk, açık kaynak, yeni uygulamalar oluşturmak için basit geliştirme ortamlarına sahip, bellek alanı kolayca genişletilebilir, İnternette paylaşım olanağı, birçok cihazla basit iletişim, yapılandırma özgürlüğü ve uygulama seçimi. Bu işletim sisteminin dezavantajları şunlardır: farklı ekran boyutları nedeniyle uygulamaların geliştirilmesinin daha zor olması, arka plan uygulamaları nedeniyle daha düşük hız, daha az güvenlik ve virüslere karşı duyarlılıktır.

Windows işletim sisteminin kurulumu, bellek modülü (USB veya DVD) ve İnternet (PHE boot) aracılığıyla farklı şekillerde gerçekleştirilebilir. Eğer bellek modülü kullanıyorsak modülü bağlayıp bilgisayarı açtıktan sonra BIOS programını etkinleştirmemiz gerekiyor.

Ürün anahtarı, lisansı doğrulayan 25 haneli bir koddur. Bu kod, Microsoft tarafından Windows işletim sisteminin teslimatıyla birlikte gönderilir veya yükleyicinin İnternet'ten indirilmesi durumunda e-postayla gönderilebilir.

Windows işletim sistemini yüklemek için iki seçenek vardır: Upgrade ya da Custom. İlk seçenek, mevcut Windows işletim sisteminin yükseltilmesi anlamına gelir. Bu kurulum ücretsizdir, ürün anahtarı gerektirmez ve kullanıcı verileri silinmez.

Windows 10 iki yapılandırma aracı içerir: Control Panel ve yeni Settings uygulaması.

Settings uygulamasındaki kategoriler şunlardır: System, Devices, Phone, Network&Internet, Personalization, Apps, Account, Time & Language, Ease of access, Cortana ve Privacy, Update & Security.

İşletim sisteminin optimizasyonu, yani bilgisayarın hızını artırmanın yöntemleri şunlardır: kullanılmayan ve başlangıç uygulamaları kaldırmak, sabit diskteki belleği boşaltmak ve birleştirmek, anti-virüs programlarını kullanmak, işletim sistemini yükseltmek ve sanal sayfaları artırmak.

Sanal makine, ana bilgisayarın (host) kaynaklarını kullanan bir konuk bilgisayar (guest) sistemidir. Ana bilgisayara Windows işletim sistemi, sanal makineye yani konuk bilgisayara ise Ubuntu işletim sistemi kurulacaktır.

Create Virtual Machine penceresinde, Name alanına sanal makinenin adını girilir (örneğin Ubuntu), Type alanına sanal makineye kurulacak işletim sisteminin türünü (Linux) ve Version alanına işletim sisteminin sürümü (32 bit veya 64 bit) girilir.

Linux işletim sistemini sanal makineye kurarken, kurulum Installation type penceresinde Install Now düğmesine tıklayarak başlatılır. Erase disk ve install Ubuntu seçeneği önceden işaretlenmiştir ancak sanal bir sabit disk olduğu için herhangi bir veri silinmesi ve kaybı olmayacağını daha önce açıklamıştık.

Raspberry Pi'nin yerleşik kalıcı belleğe sahip olmadığını ancak verileri ve işletim sistemini saklamak için minimum 16GB belleğe sahip SD kartın kullanıldığını belirtmiştik. NO-OBS (New Out-Of-Box Software), Raspberry Pi için birden çok farklı işletim sistemlerinden birinin seçilmesini ve birkaç fare tıklamasıyla otomatik kurulumla olanak sağlayan özel bir yazılımdır. Sunulan işletim sistemleri arasında Raspbian işletim sistemi de bulunmaktadır.

Raspbian işletim sistemi menüsündeki Programming kategorisi, Thonny Python IDE dahil geliştirme programlarını içerir.

Arduino mikrodenetleyici platformları için geliştirme programı, Arduino'nun resmi web sitesinden, yani arduino.org/download bağlantısından indirilir. Kurulumu indirdikten sonra kurulumu gerçekleştiriyoruz. "Run" tuşuna basıyoruz (click). Kurulum başlamazsa,

Control Panel programını açıp Device Mamanager kategorisini, ardından “Other Devices” veya “Unknown Devices” alt kategorisini seçip “Update Drivers” veya “Update Driver Software” ‘e basmak (click) gerekir.

Tekrarlama Soruları

1. Hangi programlar işletim sisteminin bir parçasıdır?

2. Windows işletim sisteminin avantajları ve dezavantajları nelerdir?

3. Linux işletim sisteminin gelişiminin başlangıcı nelerdir?

4. Linux işletim sisteminin uygulanmasına ilişkin örnekler verin!

5. Windows 10 işletim sisteminin kurulumu için cihaz seçme sürecini açıklayın!

6. Windows işletim sisteminin ürün anahtarı nedir?

7. Windows 10 işletim sistemi için Upgrade ve Custom seçeneklerinin olanklarını açıklayınız!

8. Windows 10 işletim sisteminin yapılandırması için kullanılan 2 uygulama hangileridir?

9. Settings uygulamasındaki kategorileri listeleyin!

10. Settings uygulamasının hangi kategorisi Ekran çözünürlüğünü ve parlaklığını değiştirmek için kullanılır?

11. Settings uygulamasındaki Account kategorisi ne için kullanılır?

12. Virüsleri tespit etmek ve yetkisiz erişime karşı korumak için programlar Settings uygulamasının hangi kategorisinde bulunmaktadır?

13. İşletim sistemi optimizasyonunun temel amacı nedir?

14. Başlangıç (start-up) uygulamaları terimi altında neyi tanımlıyoruz?

15. Sabit disk birleştirme sürecini açıklayın?

16. Çevresel cihaz sürücülerini hangi kategoride yükseltir?

17. VirtualBox uygulamasının uygulaması nedir?

18. Yeni bir sanal makine oluştururken Name, Tyoe ve Version alanlarına neler girildiğini açıklayınız?

19. Raspbian işletim sisteminin çalışma alanının menüsünde hangi kategorilerin bulunduğunu listeleyiniz?

20. Geliştirme programını kurup Arduino platformunu kişisel bilgisayara bağladıktan sonra sürücü kurulum prosedürünü açıklayınız?

21. Arduino platform geliştirme ortamının bileşenlerini listeleyiniz?

22. Arduino platformunun geliştirme ortamında seri monitörü ne için kullanılır?

4. Mikrobilgisayarların Programlanması

4.1. C/C++ Programlama Dili ve Uygulaması

Günümüzde programlama, birçok yazılım aracının ortaya çıkması, kullanımı kolay (İng.user friendly) ve karmaşık matematiksel modeller uygulamadan uygulamalar oluşturmaya olanak tanıyan geliştirme ortamları nedeniyle genç insanlar için ilgi çekicidir. Çok sayıda programlama dili vardır ve her birinin farklı amacı ve hedefi vardır. Biz, özellikle teknik bilimler alanında uygulama yazılımları için en yaygın kullanılan programlama dilleri olan C/C++ ve Python ile tanışacağız. C/C++ programlama dilinde yazılan **programlar** saf yapıya sahip, ihtiyaçlara göre kolaylıkla uyarlanabilen, işletim sistemlerinin, grafiksel kullanıcı arayüzlerinin ve gömülü programlanabilir sistemler **parçasıdır**.

Üçüncü konuda Arduino platformu ve sistem programları için entegre geliştirme ortamının kurulama şeklini öğrendik. Başlangıçta hazır program kodları kullanılır, ancak daha başarılı çalışma için bunları değiştirmek ve ayarlamak için bunları analiz etmek gerekir. Öğrencilerin kullanacakları elektronik bileşenlerin özelliklerine göre kendi programlarını oluşturmayı öğrenmeleri zorlu bir iştir.

Arduino platformunun yalnızca bir programı çalıştırabildiğini ve güç kaynağı açıldıktan hemen sonra çalıştırılmaya başladığını belirtelim. Arduino platformunda programlamayı daha kolay öğrenmek için 2.8.3.3 Arduino Uno R3 platformu için bilgisayar simülasyonu öğretim biriminde açıklanan Tinkercad simülatörünün kullanılması tavsiye edilir.. Programın yürütülmesinden elde edilen sonuçları izlemek için simülatörün bir parçası olan seri monitörü veya Arduino platformunun geliştirme ortamı kullanılabilir.

Arduino için yazılan programlar için taslak (İng.sketch) ismi kullanılır. Bu şekilde fikirlerin hızlı ve kolay uygulanması için programların basit yapısı vurgulanmıştır.

Taslak yönergelerden oluşmaktadır. Yönergeler (komutlar), istenen sonucu elde etmek için belirli bir eylemi başlatan ifadelerdir. Yönergeler aracılığıyla Arduino platformunu kontrol ediyoruz ve hesaplamalar yapıyoruz. [6]

Taslak iki zorunlu yapı içerir: setup ve loop. Örnek 4.1, taslağın iki zorunlu yapının genel görünümüdür.

setup() → Kurulum, giriş argümanı olmayan fonksiyondur. Bununla, **değişkenlere başlangıç değerler verilir, pinleri giriş veya çıkış olarak yapılandırılır, seri monitörün iletim hızı seçilir, dahil edilen kütüphaneler belirtilir.** Bu fonksiyon, güç kaynağı açıldıktan sonra veya Arduino platformu yeniden başlatıldıktan sonra sadece bir kez yürütülür.

loop() → Bu fonksiyonla aslında Arduino platformunu kontrol ediyoruz. İngilizce loop kelimesi bir döngü (düğüm) olarak tercüme edilir ve adından da anlaşılacağı gibi, bu fonksiyon sürekli olarak yürütülür, daire içinde döner, **giriş pinlerinin durum değişikliklerini sürekli izler ve buna göre tepki verir.** Aslında loop fonksiyonu, Arduino'ya güç olduğu sürece tekrarlanan sonsuz bir döngüdür.

Örnek 4.1.

```
1 int buttonPin = 3;           // Önce değişkenler bildirilir.
2 void setup() {              // setup her zaman başlangıç fonksiyonudur,
                              // void etiketi fonksiyonun değer
                              // döndürmeyeceği anlamına gelir.
3   Serial.begin(9600);       // Seri monitörün iletim hızını
                              // tanımlıyoruz.
4   pinMode(buttonPin, INPUT); // Giriş pinin yapılandırması.
5 }
6 void loop() {               // loop fonksiyonunun başlangıcı.
  .....
}                               // loop fonksiyonunun sonu.
```

Ancak, Arduino platformu için programlar oluşturmaya ve analiz etmeye başlamadan önce, C/C++ programlama **dilinin temel yapı elemanları** tabıtacağız: değişkenler, yönergeler ve yapılar. Bilgisayar için anlamlarını anlamak ve sözdizimlerini ezberlemek gerekir. Programlamada **sözdizimi**, bilgisayarın anlayabileceği ifade elde etmek amacıyla etiketleri, sembolik adları, operatörleri, noktalama işaretlerini, yorumları sıralamak için kurallar dizisi demektir.

4.2. Arduino Platformu İçin C/C++ Programlama Dilinde Değişkenler ve Operatörler

Değişkenler bilgisayarda işlenen **verileri saklamaktadır**. Her değişken için bellekte bir konum öngörülür ve ayrılır. Ayrılan her konumun kendi adresi vardır. Adreslerin hatırlanması zor olduğundan değişkenlere sembolik adlar verilir. Sembolik ismin yanı sıra her değişkene veri türü etiketi de atanmalıdır. Ayrılmış bellek alanının boyutu veri türüne bağlıdır. Ayrıca, bilgisayar farklı veri türlerini farklı şekilde davranır. Örneğin, aritmetik yönergeler sayıları işleyebilir ancak metin verilerini işleyemez. Aşağıda temel **veri türlerinin** tanımları verilmiştir.

- int → Tamsayılar en sık kullanılan verilerdir. Arduino platformu her tamsayı için iki bayt bellek alanı ayırır. Tam sayıların aralığı -32768 ile +32767 arasındadır.
- float → Ondalık sayılar, tamsayılardan daha fazla sayıda olası değere sahip oldukları için analog değerleri temsil etmek için kullanılır. 7 ondalık basamaklı ondalık sayılar Arduino belleğinin 4 baytını kaplar ve sayı aralığı $3.4028235E+38$ ile $-3.4028235E+38$ arasındadır. E harfinden sonraki sayı üssün 10 tabanlı üssünü temsil eder (örnek $35E3=35 \cdot 10^3=35 \cdot 1000=35000$)
- double → 15 ondalık basamaklı ondalık sayılar, 7 ondalık basamaklı ondalık sayılardan daha kesindir ve 8 baytlık bellek alanı kaplarlar.
- char → Bir sembol (İng. char) bir harfi, rakamı veya alfanümerik karakteri temsil eder. Bir sembol belleğin bir baytını kaplar. Semboller tek tırnak işareti arasına yazılır (örnek 'A').
- string → Metin semboller dizisidir ve çift tırnak işareti arasında yazılır (örnek "iyi gunler")
- bool → Mantıksal verilerin sadece iki değeri vardır; sıfır veya bir, yanlış (false) veya doğru (true). Bir mantıksal değişken bir bayt yer kaplar.

Değişkenlerin **ismi ve veri türü** yanısıra kendi değerleri de vardır. Değişkenin değeri programın başında, değişkenleri bildirdiğimizde (beyan ettiğimizde) veya daha sonra program sırasında belirlenebilir. Değişkenlere değer eşittir işaretiyle (=) atanır ve bu işaret aynı zamanda atama operatörü olarak da bilinir. Değişkenin değeri değiştirilebilir veya sabit olabilir (İng.read only). Eğer sabitse, değişken türünün önüne const kelimesi olmalıdır.

Örnek 4.2.

```
1 int countUp = 0; // Sembolik adı 'countUp' olan
// tamsayı değişkeni.
2 const float pi = 3.14; // Sabit değere sahip ondalık sayı.
3 bool isCodingFun = true; // Sembolik adı isCodingFun ve true
// değeri olan mantıksal değişken.
4 string stringOne = "Hello String"; // stringOne metinsel değişkeninin
// adıdır, Hello String ise onun
// değeridir.
```

Diziler aynı türden değişkenlerden oluşan gruplardır. Bir dizi bildirimini şunları içerir: değişken türü, dizi adı ve eleman (öge) sayısı. Orta parantez eleman sayısını içerir, büyük parantez ise elemanları listeler. Elemanlara erişmek için yine orta parantez kullanılır, yani dizi adı ve orta parantez içinde elemanın sıra numarası yazılır.

Örnek 4.3.

```
1 string araba [3]={"Volvo", "BMW", "Ford"}; // Dizi metinsel değişkenler grubudur,
// araba dizinin adıdır, orta parantez
// içindeki 3 sayısı dizinin 3 eleman
// içerdiği anlamına gelir, elemanlar
// ise büyük parantez içinde yazılır.
//
2 araba[0] = "Opel"; // Sıra numarası sıfır olan
// elemanı çağırıyoruz ve
// değerini değiştiriyoruz.
```

Diziler iki boyutlu da olabilir ve o zaman elemanlar satırlar ve sütunlar halinde düzenlenir, ortadaki iki parantez içindeki sayılar ise sütun sayısını ve satır sayısını gösterir.

Örnek 4.4.

```
1 int Table [2][3] = {{3, 42,1}, {7, 3, 12}}; // Dizi 2 sütun ve 3 satırdan
// oluşmaktadır.
```

Operatörler, programcının kesin olarak tanımlanmış kurallara göre ifadeler, yönergeler oluşturduğu işaretlerdir. Birkaç tür operatör vardır: matematiksel, mantıksal, karşılaştırma, atama operatörleri. Her biri üzerinde kısaca duracağız.

Matematiksel operatörlere aritmetik operatörler denir.

+	→	Toplama operatörü
-	→	Çıkarma operatörü
*	→	Çarpma operatörü
/	→	Bölme operatörü
%	→	Bölme işleminde kalanın hesaplanması için operatör

Örnek 4.5.

```
1 float a = 5.5;
2 float b = 6.6;
3 int c = 0;
4 c = a - b; // çıkarma sonucu -1,1 olmasına rağmen 'c' tam sayı
              // değişkeni olduğundan dolayı değeri -1 olacaktır.
```

Örnek 4.6.

```
1 int a = 5;
2 int b = 10;
3 int c = 0;
4 c = a * b; // 'c' değişkeni 50 değerini alır
```

Örnek 4.7.

```
1 float a = 55.5;
2 float b = 6.6;
3 int c = 0;
4 c = a / b; // bölme sonucu 8,409 olmasına rağmen 'c'
              // değişkeni 8 değerini alır
```

Örnek 4.8.

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

Örnek 4.9.

```
1 float a = 55.5;
2 float b = 6.6;
3 int c = 0; // 'c' değişkeni tamsayıdır.
4 c = a / b; // Bölme sonucu 8,409 olmasına rağmen
              // 'c' değişkeni 8 değerini alır
```

Karşılaştırma yönergeleri içeren ifadeler true (doğru) veya false (yanlış) sonucunu verir. Çoğu zaman koşullu fonksiyonlarda kullanılırlar. Genellikle ifadelerin uzunluğunu azaltmak için özel operatörler kullanılır.

!= → Bu “eşit değildir” operatörüdür. İki değişken (x ve y) veya bir değişken ile bir sabit karşılaştırılır ve eğer farklıysa ifade doğrudur. Değişkenler tamsayı, ondalık veya ikili sayılar olabilir. Bu operatörün uygulaması örnek 4.10’da gösterilmektedir.

- < —→ Operatori më i vogël. Ndryshorja në anën e majtë të operatorit duhet të ketë një vlerë më të vogël se ndryshorja në të djathtë.
- > —→ Operatori më i madh
- <= —→ Operatori më i vogël ose i barabartë
- >= —→ Operatori më i madh ose i barabartë
- == —→ Operatori i barazisë

Örnek 4.10.

```
1 if (x != y); {           // x, y'den farklı mıdır?
2 .....                 // Büyük parantezler arasındaki program kodu sadece
: }                     // küçük parantezler içindeki ifade doğruysa yürütülür.
```

Örnek 4.11.

```
1 if (x < y); {           // x, y'den küçük müdür?
2 .....                 // Yürütülmesi yukarıdaki koşulun yerine getirilmesine
: }                     // bağlı olan yönergeler bloğu.
                        // If yapısının sonu.
```

Mantıksal yönergelere Boolean yönergeleri de denir çünkü bunlar ikili değişkenlere uygulanabilir (İng.bool). Sonuç da aynı şekilde ikili değerdir ve sıfır veya bir, yanlış (İng.false) veya doğru (İng.true) olabilir.

- && —→ Bu, mantıksal VE işlemi için operatördür (İng. and). Mantıksal bir sonucu elde etmek için her iki değişkenin de bir olması yani tamam (doğru) olması gerekir. Değişkenlerden biri sıfır ise sonuç da sıfırdır.
- || —→ Mantıksal VEYA işlemi için operatördür (İng. or). Sonuç olarak mantıksal bir elde edebilmek için iki değişkenden en az birinin bir yani tamam (doğru) olması gerekir.
- ! —→ Olumsuzluk operatörü. Bu işlemin tek bir değişkeni vardır. Onunla durum değişiyor. Değişken sıfırsa sonuç bir olacaktır ve bunun tersi de geçerlidir.

Örnek 4.12.

```
1 if (!x) {              // Büyük parantezindeki yönergeler bloğu,
                        // eğer x mantıksal sıfır ise yürütülür, çünkü olumsuzluktan sonra
2 .....                 // sonuç olarak bir elde edeceğiz.
: }
```

Aşağıda **kısaltılmış ifadeler** ve bunların eşdeğer aritmetik ve mantık yönergeleri verilmiştir.

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

4.3. Arduino Platformu İçin C/C++ Programlama Dilindeki Yönergeler

Fonksiyona göre yönergeleri birkaç gruba ayıracağız:

- Giriş-çıkış pinleriyle çalışma yönergeleri
- Zaman kontrolü yönergeleri
- Matematiksel yönergeler
- Bit ve bayt yönergeleri
- Seri iletişim yönergeleri
- Kütüphanelerle çalışma yönergeleri

4.3.1. Giriş-Çıkış Pinleriyle Çalışma Yönergeleri

Giriş-çıkış pinleriyle çalışma yönergeleri, pinleri (giriş veya çıkış) yapılandırma ve değerlerini yazmak veya okumak için kullanılır. [12]

pinMode(pin,mode) → Dijital pin modu (düzeni) ile giriş veya çıkış çalışma modu temsil ediliyor. Eğer pin giriş piniyse Arduino elektronik bileşenden veri alır, çıkış piniyse Arduino veri gönderir. Çıkış için etiket OUTPUT' tur, giriş için INPUT'tur.

digitalRead(pin) → Bu yönerge (komut) giriş pininin **değeri okunur**. Değeri HIGH (yüksek seviye) veya LOW (alçak seviye) olabilir. Eğer pini daha önce tamsayı değişkeni olarak bildirmişsek, orta parantez içine pinin numarası veya sembolik adı yazılır. Pin çıkış piniyse okuma yönergesi mümkün değildir.

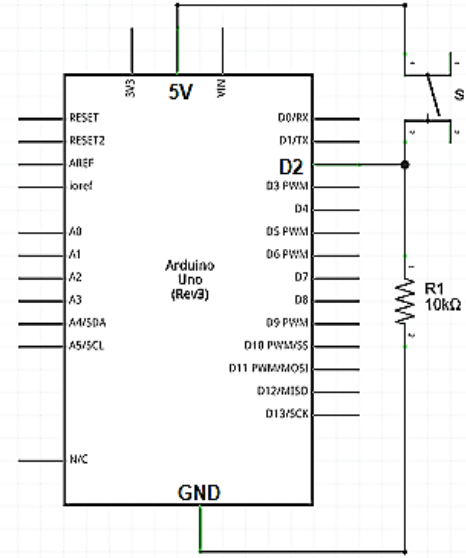
Örnek 4.13.

```

1 int dugme=2;           // dugme ikinci pinin sembolik adıdır.
2 int durum=0;          // durum sembolik adı ve başlangıç değeri
  /* ... */             // sıfır olan bir değişken bildiriyoruz.
3 durum = digitalRead (dugme); // İkinci pinin değeri olunur ve durum
  //                     // değişkenine atanır

```

Şekil 4.1'de gösterilen işlevsel şema, Örnek 4.13.'teki program koduyla ilgilidir. Giriş bileşeni olarak Arduino Uno R3'ün ikinci pinine bağlanan düğme kullanılır. Düğmeye basıldığında ikinci pin güç kaynağına bağlanır ve okunan değer HIGH olur. Düğmeye basılmadığında ikinci pin topraklamaya bağlanır ve okunan değer LOW olur.

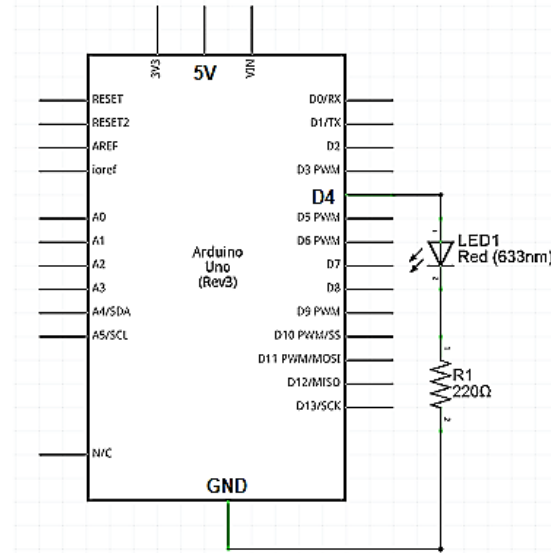


Şekil 4.1. Düğmenin Arduino Uno R3 ile bağlanması

Şekil 4.1'deki şemada rezistör ve düğme yerlerini değiştirirse, düğmeye basıldığında ikinci pin topraklamaya bağlanacak ve okunan değer LOW yani mantıksal sıfır olacaktır.

digitalWrite(pin,deger) → Bu yönerge ile dijital çıkış pini düşük veya yüksek seviyeye ayarlanır. digitalWrite yönergesinden farklı olarak, digitalWrite yönergesi iki parametre içerir: pinin sembolik adı (veya numarası) ve HIGH veya LOW değeri.

Şekil 4.2'de gösterilen işlevsel şema Örnek 4.14'ün program koduyla ilgilidir. Çıkış bileşeni olarak Arduino Uno R3'ün dördüncü dijital pinine bağlanan LED diyotu kullanılır. Dördüncü pin yüksek seviyeye (HIGH) ayarlandığında LED diyotu yanacaktır. LED diyotu katotun dördüncü pine bakacak şekilde ters yerleştirirsek düşük seviyede (LOW) aktif olacaktır.



Şekil 4.2. LED diyotun Arduino Un R3 ile bağlanması

Örnek 4.14.

```
1 int ledDiyot=4; // ledDiyor dördüncü pinin sembolik
// adıdır.
2 digitalWrite(int ledDiyot,HIGH); // Dördüncü pin yüksek seviyeye ayarlanır
// ve diyot yanar
```

Örnek 4.15. dijital pinlerle çalışmayla ilgili üç yönergenin uygulamasına ilişkin bir program göstermektedir. LED diyotu sadece düğmeye basıldığında yanar.

Örnek 4.15.

```
1 int ledPin = 13; // LED diyot Arduino Uno'nun.
// on üçüncü pinine bağlıdır.
2 int inPin = 7; // Düğme yedinci pine bağlıdır.
3 int val = 0; // val tamsayı değişkeni okunan.
// değeri saklar.
4 void setup() { //
5 pinMode(ledPin, OUTPUT); // On üçüncü pinin çıkış pini olarak
// yapılandırılması
6 pinMode(inPin, INPUT); // Yedinci pinin giriş pini olarak
7 } // yapılandırılması
8 void loop() { //
9 val = digitalRead(inPin); // Giriş pininin okunması.
10 digitalWrite(ledPin, val); // LED diyotuna bağla olan pin
// düğmeye bağlı olan pinle
// aynı değere sahiptir.
```

2.3. Arduino Uno R3'ün bileşenleri, öğretim biriminde, bu platformun analog giriş ve çıkışlarını tanımiştık. Analog pinler için darbe genişliği modülasyonunun çok önemli olduğunu vurgulamıştık.

analogRead(pin) → Bu yönerge ile **bir analog girişin değeri okunuyor**. Analog girişin gerilimi sonsuz sayıda farklı değere sahip olabilir ve 0 ila 5 V aralığında değişir. Arduino Uno R3 bileşimindeki analog-dijital dönüştürücü, analog değerleri 0 ila 1023 arasındaki tam sayılara dönüştürür. Tamsayı daha sonra 10 bitlik ikili kod olarak temsil edilir. 5V değeri 1024 bölüme bölünürse, bir bölüm 4,9 mV'luk gerilime karşılık gelecektir. Buna göre, 0 V'tan 4,9 mV'a kadar tüm gerilim değerleri 0 sayısına, 4,9 mV'tan 9,8 mV'a kadar gerilimler 1 sayısına, 9,8 mV'tan 14,7 mV'a

analogWrite(pin,value) → kadar 2 sayısına karşılık gelip, 1023 sayısına karşılık gelen 5 V gerilime kadar devam ediyor.

Analog çıkışlarla LED diyotun parlaklığı veya bir motorun hızı kontrol edilebilir. **Çıkış pinine analog bir değer yazılır ve bu değer sıfırlar ve birler dizisine dönüştürülür.** Darbelerin genişliği veya süresi ve çıkış gerilimin büyüklüğü, 0 ila 255 arasında değişen kaydedilen tamsayı değerine doğru orantılı olarak bağlıdır. (Şekil 2.12)

Örnek 4.16'da. 9 numaralı çıkış pini, ışık yoğunluğu Arduino Uno R3'ün üçüncü pinine bağlı potansiyometrenin kaydırıcısının konumuna bağlı olan LED diyot ile bağlıdır. Aynı örneğin 9 numaralı program satırında potansiyometreden okunan değer dörde bölünür, çünkü çıkış sinyali giriş sinyalinden dört kat daha küçük değer aralığına sahiptir.

Örnek 4.16.

```
1 int ledPin = 9; // LED diyot dokuzuncu
// pine bağlıdır.
2 int analogPin = 3; // Potansiyometre üçüncü
// pine bağlıdır
3 int val = 0; // val okunan değer saklanması
// için kullanılan değişkendir
4 void setup () {
5   pinMode(ledPin, OUTPUT); // çıkış pinin yapılandırılması
6 }
7 void loop() {
8   val = analogRead(analogPin); // Giriş sinyalinin değerinin
// okunması. Okunan değer
// 0 ila 1023 aralığındadır
9   analogWrite(ledPin, val / 4); // Yazılan değer 0 ila 255
// aralığındadır.
10 }
```

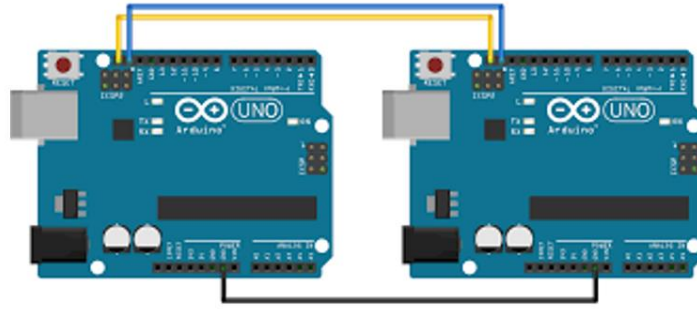
tone(pin, frequency) → Bu yönerge ile, Arduino Uno R3'ün 3 veya 11 numaralı pini için periyodik bir darbe dizisi oluşturulur. Frekans programcı tarafından seçilir. Uyarma sinyalinin süresini seçmek mümkündür. Aksi takdirde, notone() yönergesi

tone(pin, frequency,duration)

çalıştırılana kadar sürecektir. Çıkış pinine tekdüze ses üretecek bir zil bağlanabilir.

4.3.2. Seri İletişim Yönergeleri

Arduino Uno R3'ün RX (Receive) ve TX (Transmit) pinleri seri iletişim pinleridir. Şekil 4.3.'te iki Arduino Uno R3 platformunu bağlama yöntemi gösterilmiştir. Birinci platformun RX pini, ikinci platformun TX pinine bağlanmıştır ve bunun tersi de olabilir.



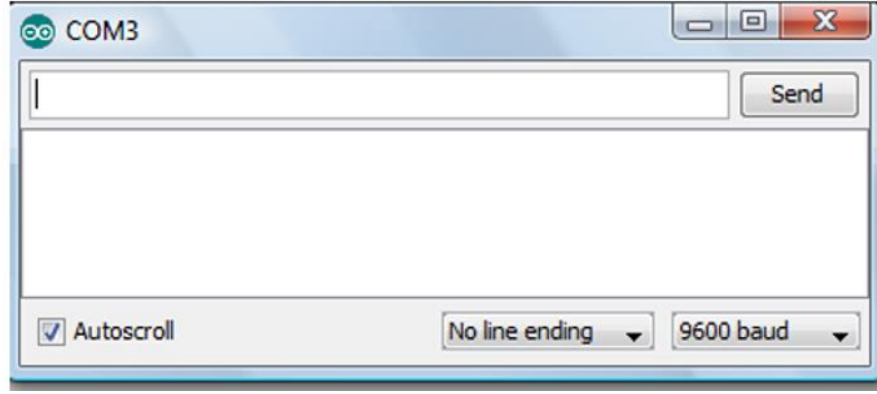
Şekil 4.3. İki Arduino Uno platformu arasında seri bağlantı

Bu pinler aracılığıyla platform, Raspberry Pi mikro bilgisayarına, Wi-Fi modülüne, Bluetooth modülüne ve seri iletimi destekleyen diğer cihazlara bağlanabilir. Seri iletişimin büyük avantajı, veri alışverişinin yerleşik seri monitör aracılığıyla izlenilmesidir. Arduino Uno R3'ün diğer cihazlarla aldığı veya gönderdiği tüm verileri ekranda yazdırılarak görebiliriz. Tabii ki, bunun için Arduino platformunun USB kablo ile bilgisayara bağlanması ve Arduino geliştirme programı çerçevesinde seri monitörün açık olması gerekiyor. Seri monitör şekil 4.4'te gösterilmiştir.

Seri monitörle çalışmak için yaklaşık yirmi yönerge vardır, ancak biz sadece ikisinden bahsedeceğiz.

Serial.begin (hiz) → Bu yönerge ile geçirme kapsamı (bant genişliği) yani seri iletim hızı ayarlanır ve bu hız **seri monitörün sağ alt köşesinde belirtilir**. Standart iletim hızı 9600 baud'dur, bu da saniyede 9600 bit anlamına gelir, ancak kullanıcı RX ve TX pinlerine bağlı cihazın gerektirdiği zaman seri iletişim hızını değiştirebilir.

Serial.print (x) → Seri monitörün ekranında **değişkenin değeri yazdırılır**.



Şekil 4.4. Arduino platformu için geliştirme ortamının seri monitör

Örnek 4.17'de seri monitör geçen süreyi milisaniye ile ifade edilmiş olarak görüntüler.

Örnek 4.17.

```

1 unsigned long sure;
2 void setup() {
3   Serial.begin(9600);
4 }
5 void loop() {
6   Serial.print("Gecen sure:");
7   sure = millis();
8   Serial.println(sure);
9   delay(1000);           // Seri monitörde görüntülenen veri sayısını
10  }                       // azaltmak için bir saniyelik
                           // duraklama.

```

Örnek 4.18'de, Arduino Uno R3, A0 pinindeki analog gerilimin değerini ölçer, seri monitör ise bu değeri görüntüler. Arduino Uno R3'ün giriş-çıkış pinleri için maksimum gerilimin 5V olduğunu ve bu değer üzerine çıkılmasına izin verilmediğini belirtelim. Örnekte analogRead yönergesi kullanılmıştır. Analog-dijital dönüştürücü, analog değerleri 0'dan 1023'e kadar tam sayılara dönüştürdüğünden dolayı, ölçülen gerilimin değerini elde etmek için aşağıdaki formül uygulanır.

$$U[V] = \frac{\text{Okunan tamsayı değer}}{1023} \cdot 5[V]$$

Örnek 4.18.

```

1 const float maxVolti = 5.0;           // A0 pini için maksimum değer
4 void setup() {
5   Serial.begin(9600);                 // Seri iletim hızı
7 }
8 void loop() {
9   int deger = analogRead(A0);         // Giriş pinin okunması.

```

```

10 float gerilim = (deger /1023) * maxVolti;      // Ölçülen gerilimin hesaplanması
    Serial.println(gerilim);                    // Sonucun görüntülenmesi
11 }

```

4.3.3 Zaman Kontrol Yönergeleri

Program kodları çok yüksek hızda yürütülür. Çıkışa LED diyotu bağlarsak ve yanıp sönmesi gerekiyorsa, iki durumu arasına (İng.ON/OFF) gecikme süresi (İng. delay) getirilmesi gerekir. Aksi takdirde kullanıcı diyotun dönüşümlü olarak açılıp kapandığını fark etmeyecektir.

delay (ms) → **Gecikme süresi** milisaniye ile ifade edilir. Bir saniyede 1.000 milisaniye veya 1.000.000 mikrosaniye vardır.

Örnek 4.19.

```

1 int ledPin = 13;                // LED diyotu onüçüncü pine
                                // bağlıdır.
2 void setup() {
3   pinMode(ledPin, OUTPUT);     // LED diyotun pini çıkış pini
                                // olarak yapılandırılıyor.
4 }
5 void loop() {
6   digitalWrite(ledPin, HIGH);  // LED diyot açılır.
7   delay(1000);                 // Bir saniyelik gecikme süresi (duraklama).
8   digitalWrite(ledPin, LOW);   // Led diyot kapanıyor.
9   delay(1000);                 // Bir saniyelik gecikme süresi.
10 }

```

time=micros() → Bu iki yönerge ile, Arduino Uno platformunun programı çalıştırmaya başladığı andan mikrosaniye ve milisaniye olarak ifade edilen süre ölçülür. Geçen süreyi görüntülemek için seri monitörü açmak gerekir.

time=milis()

4.3.4 Matematiksel Yönergeler

max(x,y) → Bu yönerge x ve y sayılarından daha büyük olanı seçer. Örnek 4.21'deki yönerge sensör değerinin 20'den küçük olmasına izin vermemektedir.

- min(x,y)** → Yönerge, x ve y sayılarından daha küçük olanı seçer. Örnek 4.20.'deki yönerge, sensör değerinin 100'den büyük olmasına izin vermez.
- constraim (x,a,b)** → Yönerge, x değişkeninin aralığını sınırlar; a minimum değer ve b maksimum değerdir.

Örnek 4.20.

```
1 sensVal = min(sensVal, 100); // Sensör değeri yüzün küçükse
// bu değer korunur.
// Sensörün değeri yüzden büyükse
// değer yüze eşitlenir.
```

Örnek 4.21.

```
1 sensVal = max(sensVal, 20); // 20 ile sensör değeri arasında
// büyük olan değeri seçer.
```

- map(deger, fromLow,fromHihg,toLow, toHigh)** → Bu yönerge, değişkenin **değer aralığını değiştirir**. fromLow ve fromHigh eski aralığın minimum ve maksimum değerleridir, toLow ve toHigh ise yeni aralığın minimum ve maksimum değerleridir.

Örnek 4.22.'de analog sinyalin değerinin aralığı değişir. Başlangıçta, giriş örneklerin kodlanması 10 bit ile yapılır ve aralık 0 ila 1023 arasındadır, ancak daha sonra aralığı 0 ila 255 arasında değiştiriyoruz ve kodlama 8 bit ile olur.

Örnek 4.22.

```
1 void loop() { //
2   int val = analogRead(0); // Sıfırıncı pinin analog değeri okunur
// ve val değişkeninde saklanır.
3   val=map(val, 0, 1023, 0, 255); // Aralık değiştiğinde değişken
// de değişir. Aralık azaldığı için
// değer de buna orantılı olarak
// azalmasını bekliyoruz.
4   analogWrite(9, val); // Elde edilen yeni değer 9 numaralı
// çıkış pinine yazılır.
5 }
```

4.3.5. Bit ve Bayt Yönergeler

- bitClear (x,n)** → Yönerge, x değişkeninde n sıra numaralı biti **sıfırlar (sıfıra ayarlar)**. Tamsayı değişkeni onlu sayı sisteminde ise ikili sayı sistemine dönüştürülmelidir. En az ağırlıklı bit (sağdan birincisi) sıfır konumundadır. Bu yönergenin sonucu, n'inci biti sıfırlanan x değişkeni olacaktır.
- bitSet (x,n)** → Yönerge x değişkenindeki n sıra numaralı n olan biti ayarlar (yüksek seviyeye ayarlar).
- bitRead (x,n)** → x değişkeninin n'inci bitinin değeri okunur. Sonuç, okunan değere bağlı olarak 0 veya 1 tek bir bit olacaktır.
- bitWrite (x,n,b,)** → Bu yönergenin üç parametresi vardır. x tam sayı değişkenidir, n bitin sıra numarasıdır ve b yazılacak bitin değeridir. Örnek 4.23.'te, bitWrite (x,0,1) yönergesi en az ağırlıklı bitin durumunu sıfırdan bire değiştirir.
- bit (n)** → Bu yönerge ile bitin, sıra numarasına yani ikili sayı içindeki konumuna bağlı olarak ağırlığı hesaplanır. Bit ağırlığı 2^n ifadesine göre hesaplanır. Buna göre, sağdan sola başlayarak, ilk bitin ağırlığı $2^0=1$, ikincinin ağırlığı $2^1=2$, üçüncünün ağırlığı $2_2=4$, dördüncünün ağırlığı $2^3=8$ 'dir vb.

Örnek 4.23.

```

1 void setup() {
2   Serial.begin(9600); // seri monitör bant genişliğinin ayarlanması
3   byte x = 0b10000000; // 0b öneki ikili değişkenler için kullanılır
4   Serial.println(x, BIN); // Seri monitör 10000000 ikili sayısını
   // görüntüler.
5   bitWrite(x, 0, 1); // En az anlamlı bit konumuna 1 yazmak
6   Serial.println(x, BIN); // Seri monitör 10000000 ikili sayısını
   // görüntüler.
7 }
```

- highByte (x)** → Veri 16 bitli ise bu yönerge en ağırlıklı sekiz biti ayırır.

lowByte (x) → Veri 16 bitli ise bu yönerge en az ağırlıklı sekiz biti ayırır.

Örnek 4.24.

```

1 int test = 0xABCD; // Onaltılık sayı sisteminde 16 bitlik bir
// değişken bildiriyoruz.
2 byte hi, lo; // İki bayt tipinden değişken bildiriyoruz.
3 hi = HighByte(test); // 0xAB sonucunu elde ediyoruz.
4 lo = LowByte(test); // 0xCD sonucunu elde ediyoruz.

```

4.3.6. Arduino Platformu İçin C/C++ Programlama Dili Yapıları

Yapılar, kesin olarak tanımlanmış işleve sahip yönerge bloklarıdır. Yönerge blokları büyük parantezlerin { } arasına yazılır. Programın yapılara bölünmesi ve büyük parantezlerin kullanılması programların analizini büyük ölçüde kolaylaştırır.

4.3.6.1. Çoklu Seçim Yapıları

Çoklu seçim yapıları if, else if ve else ifadelerinden oluşur.

if (koşul) { → **If ifadesi koşulu kontrol ediyor ve yerine getirilirse (doğruysa) yönergeler yürütülür.** Koşul, mantıksal bir ifade veya karşılaştırma operatörleri içeren bir ifade olabilir. Sözdizimi, koşulun küçük parantez içinde, yönerge bloğunun büyük parantez içinde yazılacağı şekildedir. Şekil 4.5'te bu yapının blok diyagramı gösterilmiştir.

```

if (koşul) {
yönerge 1
yönerge 2
yönerge 3
.....
}

```

if (koşul1) { → **if...else yapısı daha iyi kod kontrolü sağlar çünkü birden fazla koşulun kontrolünü gerçekleştiriyor.** If ifadesinin altındaki koşul karşılanmıyorsa, else if ifadesinin altındaki koşul kontrol edilir. else if ifadelerinin sayısı sınırsız olabilir. Her yanlış koşul, doğru bir koşul elde edilene kadar yeni kontrole yol açar. Doğru bir koşul elde edildiğinde, yönergeler yürütülür ve sonraki koşulların kontrolü durdurulur. Eğer doğru koşul elde edilmezse else ifadesinin altındaki yönergeler yürütülür. Else ifadesi, seçenek seçim yapısında son

```

if (koşul1) {
//yönerge
//bloğu 1
}
else if (koşul2) {
//yönerge
//bloğu 2
}
else {
//yönerge

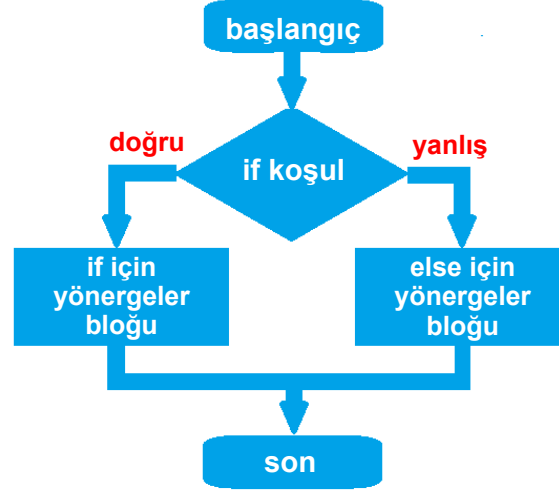
```

```
//bloğu 3
}
```

ifadedir ve else ifadesinin altında herhangi bir kontrol koşulunun olmadığı fark edilebilir. Şekil 4.6.'da, if...else yapısının blok diyagramı gösterilmektedir ve if yapısından farklı olarak, if koşulu karşılanmadığında da yürütülen yönerge bloğu vardır.



Şekil 4.5. if yapısının blok diyagramı



Şekil 4.6. if...else yapısının blok diyagramı

Örnek 4.25. üç olasılığın seçilmesi ile ilgili yapının uygulanmasına ilişkin bir örnektir. Olasılıklar şunlardır: 70 dereceden büyük bir sıcaklık, 60 ile 70 derece arasında sıcaklık ve 60 dereceden daha düşük sıcaklık.

Örnek 4.25.

```
1 if (temperature >= 70) {
2   // Tehlike! Sistemi kapatın.
3 }
4 else if (temperature >= 60 && temperature < 70) {
5   // Dikkat! Diğer parametrelerin değerlerini kontrol edin.
6 }
7 else {
8   // Sıcaklık 60 dereceden daha düşüktür.
9   // Sistem güvenli bir şekilde çalışıyor.
10 }
```

Örnek 4.26.'da düğme basılmış olmadığında LED diyotu yanar, basıldığında ise diyot yanmayı durdurur. Düğmeye basıldığında pininin yüksek seviyede olduğunu ve diyotun aktif anotla olduğunu tahmin ediyoruz.

Örnek 4.26.

```

1 int val = digitalRead(taster); // Düğme durumunun kontrol eildmesi.
2 if (val == LOW) { // Koşulun doğruluğunu tespit ediyoruz.
3     digitalWrite(led, HIGH); // Diyotun pini yüksek seviyede
// ayarlanır.
4 }

```

Örnek 4.27.'de giriş bileşeni olarak iki düğme kullanılmaktadır. If yapısı koşulunda VE mantıksal yönergesi kullanılır ve if yapısının yürütülebilmesi için her iki düğmenin pinlerinin yüksek seviyeye ayarlanması gerekir.

Örnek 4.27.

```

1 if (dugmeA == HIGH && dugmeB ==HIGH) {
// Düğmelerin durumunun kontrol edilmesi ve koşulun doğruluğunun tespit edilmesi.
2     digitalWrite(led, HIGH); // LED diyot yanıyor.
3 } //
4 else { // if yapısındaki koşul geçerli değilse
// else yapısındaki program kodu
// yürütülür.
5     digitalWrite(led, LOW); // LED diyot yanmıyor.
6 }

```

Çoklu olanaklar arasında seçim yapıları **switch...case** ifadeleri yardımıyla da oluşturulabilir.

<pre> switch (degisken) { case 1: // yönergeler bloğu break; case 2: // yönergeler bloğu break; default: // yönergeler bloğu break; } </pre>	<p>→ if ifadesi gibi, switch...case ifadesi de programın akışını kontrol ederek farklı durumlar (İng. case) için farklı program kodlarının oluşturulmasına olanak verir. switch ifadesinin hemen yanında, değeri program yürütmenin hangi duruma geçileceğine (İng. switch) bağlı olan değişkenin adı küçük parantez içinde yazılır. İngilizce default kelimesi “varsayılan (standart)” anlamına gelir ve yukarıdaki durumlardan hiçbiri yürütülmezse default ifadesinin altındaki kod yürütülür. Break ifadesi switch ifadesiyle birlikte kullanılır ve her durumun sonunu belirtir. Break ifadesi olmadan yapı sonuna kadar yürütülmeye devam edecektir.</p>
--	--

Örnek 4.28.'de giriş bileşeni olarak A0 analog girişine bağlı sensör, fotorezistör kullanılmıştır. Hatırlayalım, sensörlerden gelen analog sinyaller, çıkışında 0 ila 1023 aralığında bir tamsayı değeri üreten analog-dijital dönüştürücünün giriş sinyalleridir. Fotorezistörü kapalı bir odada kullanacağımızı tahmin ederek, maksimum değer 600'e düşürülür. Bu örnekte switch yapısı üç if yapıyı değiştirmektedir. Programda

sensörün değeri okunur, ardından map() yönergesi kullanılarak olası değer sayısı toplam üçe değişir ve son olarak seri monitörde elde edilen y değerine bağlı olarak üç mesajdan biri görüntülenir.

Örnek 4.28.

```
1 const int sensorMin=0;           // Sensörün maksimum ve minimum
2 const int sensorMax=600;        // değeri tanımlanır.
                                   //
3 void setup() {
4   Serial.begin(9600);           // Seri monitörün bant genişliği
5 }                                 // ayarlanır.
6 void loop() {
7   int x=analogRead(A0);         // Sensörün değeri okunur
8   int y=map(x,sensorMax, sensorMin, 0,2);
   //Sensör değerleri aralığının azaltılması gerçekleştirilir.
   // Olası değerlerin sayısı 0-600'den 0-2'ye düşürülür.
9   switch(y) {                   // switch yapısı başlar
10  case 0:
11    Serialprintln("karanlik");
12    break;
13  case 1:
14    Serialprintln("yarim karanlik");
15    break;
16  case 2:
17    Serialprintln("aydinlik");
18    break;
19  }
20  delay(10);                     // Daha kararlı çalışma için her iki
21 }                                 // sensör okuması arasında 10
                                   // milisaniyelik gecikme uygulanır.
```

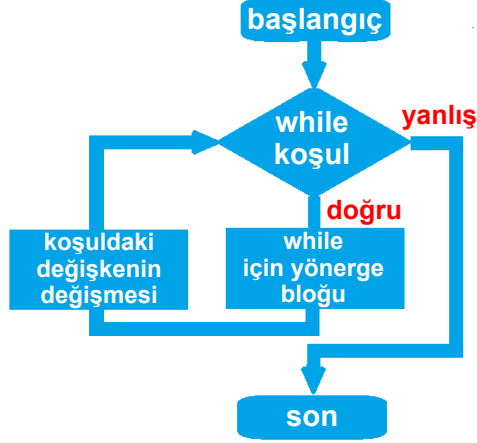
4.3.6.2. Tekrarlama Yapıları (Döngüsel Yapılar)

Döngü tekrarlama yapıları while, do...while ve for ifadelerinden oluşur. Bu ifadelerin beraberinde continue ifadesi de kullanılabilir.

while (koşul) { → İngilizce while kelimesi “kadar” anlamına gelir ve küçük parantez içindeki koşulun doğruluğu kastedilmektedir. Koşul doğru olduğu sürece döngü tekrarlanacaktır, yani while ifadesinin altındaki yönerge bloğu tekrar tekrar yürütülecektir. While döngüsünün sonsuz olmamasına dikkat edilmelidir. Bu, koşuldaki

}

değişkenin yapının büyük parantezleri içindeki yönergelerle değiştirilmemesi durumunda meydana gelecektir. Bu amaçla koşuldaki değişkenin azaltılarak veya artırılarak değiştirilmesi veya bir sensörün durumuna bağlı olması gerekir.



Şekil 4.7. while yapısının blok diyagramı



Şekil 4.8. do...while yapısının blok diyagramı

Örnek 4.29.'da sensör tarafından ölçülen değer 100'den büyük olduğu sürece diyot yanıp söner.

Örnek 4.29.

// İlk önce analog-dijital dönüşüm gerçekleştirilir ve ardından
// while döngüsündeki koşulun doğruluğu kontrol edilir.

```

1 while(analogRead(sensor) > 100) {
2   digitalWrite(LED, HIGH);           // LED diyotu yanar
3   delay(100);                         // 100 ms beklenir
4   digitalWrite(LED, LOW);            // Led diyotu yanmaz
5   delay(100);                         // 100 ms beklenir
6 }
    
```

```

do{
yönerge 1
yönerge 2
yönerge 3
....
} while (koşul)
    
```

→ do...while ve while ifadeleri arasındaki fark, do...while ifadesinde koşulun doğruluğunun döngünün sonunda kontrol edilmesidir. Bu, koşulun doğru olup olmadığına bağlı olmadan döngünün bir kez yürütülmesi gerektiği anlamına gelir. Şekil 4.7. ve Şekil 4.8. while ve do...while iki yapı arasındaki karşılaştırmayı göstermektedir.

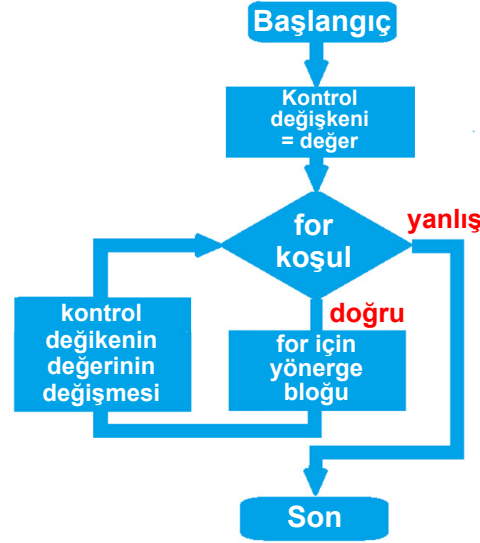
Örnek 4.30.'da giriş bileşeni olarak yine sensör kullanılır. do...while döngüsü analog-dijital dönüştürücünün çıkışında 100'den büyük bir değer elde edilene kadar tekrarlanır.

Örnek 4.30.

```

1 int x = 0;
2 do {
3   delay(50); // Sensöre algılamayı gerçekleştirmesi için
                // gerekli süre verilir.
4   x = analogRead(Sensor); // Sensörün değeri okunur
6 } while (x < 100); // Koşulun doğruluğu kontrol edilir.

```



Şekil 4.9. for yapısının blok diyagramı

```

for (değişkenin bildirimi;
koşul; artış)
{
  yönerge 1
  yönerge 2
  yönerge 3
  .....
}

```

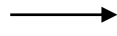
→ for ifadesi **büyük parantezler içine alınmış yönerge bloğunu birden fazla kez tekrarlamak** için kullanılır. Genelde, değeri sürekli artan bir sayaç kullanılır ve maksimuma ulaştığında döngünün yürütülmesi durur. Genellikle for ifadesine sahip tekrarlama yapısı, birden fazla pini aynı olarak yapılandırmak veya birden fazla veriyi aynı şekilde işlemek için kullanılır.

Örnek 4.31.'de LED diyot, genişlik modülasyonlu darbeler veren analog çıkışlardan birine bağlıdır. Çıkış değeri 0'dan 255'e yükseldikçe darbelerin süresi ve çıkış geriliminin ortalama değeri ve dolayısıyla ışığın yoğunluğu da artar.

Örnek 4.31.

```

1 int PWMpin = 10;           // LED diyotu onuncu pine bağlıdır.
2 void loop() {             //
3   for (int i = 0; i <= 255; i++) // for yapısı için değişkenin
   {                         // bildirim, yürütülmesinin
                               // koşulu ve değişkeni artırma
                               // yönergesi.
4   analogWrite(PWMpin, i); // Değişkenin değeri çıkış
                               // pinine yazdırılır
5   delay(10);
6   }
7 }
```

continue

continue ifadesi ile for, while veya do...while yapılarında birkaç döngü koşullu olarak atlatılabilir.

Örnek 4.32. continue ifadesinin kullanımına ilişkin bir örnektir. Bu örnekte, örnek 4.31'e kıyasen 41'den 119'a kadar olan değerler atlatılmıştır.

Örnek 4.32.

```

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

Örnek 4.33. for ve if olmak üzere iki yapının birleşimidir. for döngüsü dört kez tekrarlanabilir, ancak koşul ikinci dijital pinin düşük seviyede olmasıdır.

Örnek 4.33.

```

1 for (int i=0; i < 4;)           // Değişkenin bildirim ve döngünün
   {                               // tekrarlama koşulu.
2   {                               // for yapısının başlangıcı.
3   if(digitalRead(2) == LOW) { // if yapısının koşulu va başlangıcı.
4     i++;
5   }                               // if yapısının sonu.
6 }                               // for yapısının sonu.
```

4.3.6.3. Dallanma yapıları

Şimdiye kadar, programlar doğrusal olarak yürütülüyordu, yani yönergeler üst üste diziliyordu. Dallanma yapıları ile programın akışını değiştiriyor, prog-

ram ko dunun bazı kısımları atlatılabilir ve bu şekilde program içinde ileri geri gidilebilir.

goto etiket; → Dallanma yapıları, “gidin” anlamına gelen goto ifadesi ile gerçekleştirilir. goto ifadesinin ardından, aslında atlanacak yerin sembolik adı olan etiket (İng.label) gelir.

Örnek 4.34’te. birden fazla yapı kullanılmıştır: üç for yapısı, biri if yapısı ve biri goto yapısı. Bu, bir yapının diğerinin içine yerleştirilmesinin bir örneğidir. If yapısının koşulu geçerliyse dallanma yapısı yürütülecek, 1 numaralı yönerge bloğu atlatılacak ve 2 numaralı yönerge bloğu doğrudan çalıştırılacak. if yapısının koşulu geçerli değilse il önce 1 numaralı yönerge bloğu birden fazla kez yürütülecektir 1 numaralı yönerge bloğunun kaç kez yürütüleceği tekrarlama yapılarındaki parametrelere bağlıdır ve r×g×b’ye eşittir.

Örnek 4.34.

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

4.3.7 Kütüphanelerle Çalışma Yönergeleri

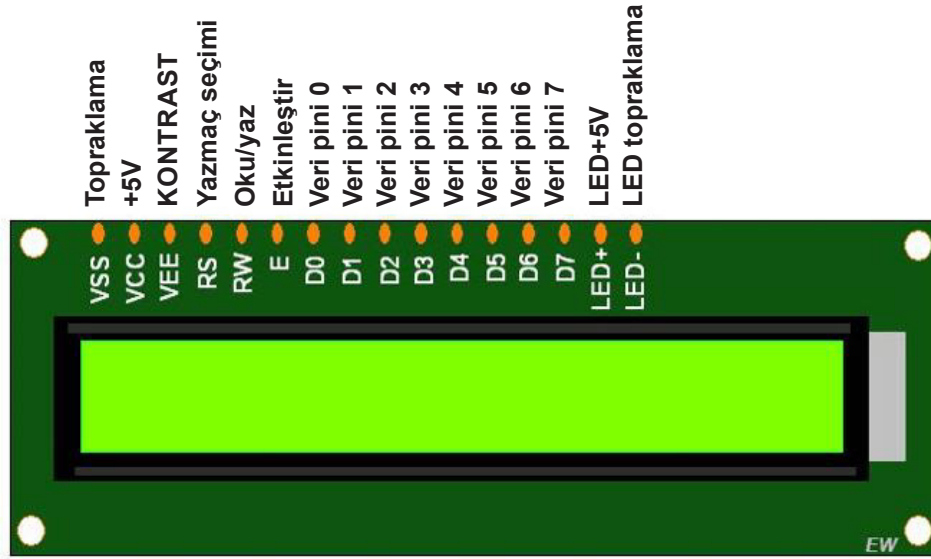
Çoğu programlama platformu gibi, Arduino geliştirme ortamı da kütüphanelerin kullanımı yoluyla genişletilebilir. **Kütüphaneler, çevresel cihazlar ile Arduino platformu arasındaki iletişimi sağlayan alt programlardır.** Kütüphaneler, uzun program kodlarının yazılmasını azaltan ve çalışmayı oldukça kolaylaştıran yönergeler içerirler. İşlevselliğe göre kütüphaneler birkaç gruba ayrılabilir: iletişim (Wi-Fi, Ethernet, GSM), veri işleme, veri depolama (Cloud Data, SD kartlarla çalışma), ekranlar (LCD), cihaz yönetimi (motorlar), sensörler (sıcaklık, optik, UV, basınç), zamanlama

vb. Arduino geliştirme ortamı birkaç standart kütüphane içerir ve bunları menüde **Sketch** → **Import Library** seçeneğine basarak (İng.click) kullanabiliriz. Geliştirme ortamı aranan kütüphaneyi içermiyorsa İnternet'ten indirilebilir.

#include < ad > → Bu yönerge ile, kütüphanenin adını belirterek ana programdaki kütüphane çağrılır.

Biz şu kütüphanelerle tanışacağız: [LiquidCrystal](#), Servo ve CapacitiveSensor.

LCD ekranlar (İng.Liquid crystal display) çıkış bileşenleridir ve programın yürütülmesinden elde edilen sonuçların görsel olarak görüntülenmesi için kullanılır. Şekil 4.10.'da, iki satırdan oluşan ve her birinin 16 sembolü olan alfanümerik LCD ekranın görünümü gösterilmiştir. Aynı şekilde onun pin diyagramı da gösterilmektedir.



Şekil 4.10. LCD ekranın dış görünümü ve pin-diyagramı

LiquidCrystal kütüphanesi, geliştirme ortamın içeriğinde olan standart bir kütüphanedir ve onun aracılığıyla Arduino platformu, LCD ekranını yönetir. Bu kütüphane 20 farklı yönerge içerir, ancak biz sadece en sık kullanılanlar üç yönerge açıklayacağız.

LiquidCrystal (rs, rw, enable, d4, d5, d6, d7) → **Bu yönerge ile** Arduino platformunun LCD ekran üzerinde aynı isimli pinlere bağlanmak için **kullanılan pinler tanımlanır**. Bu yönerge, Örnek 4.35'te gösterildiği gibi kütüphanenin programa dahil edilmesinden sonra kullanılır.

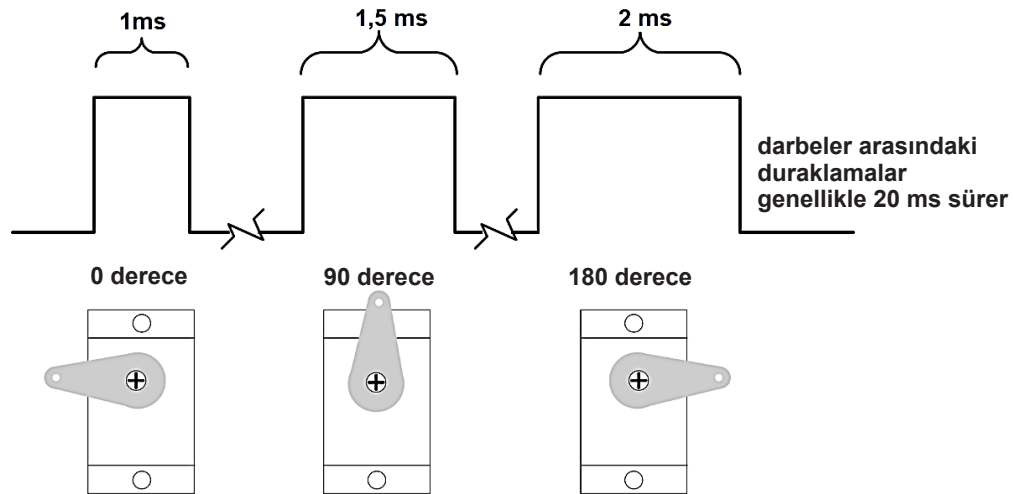
- lcd.begin(sut,sat) → Bu yönerge ile ekranla iletişim başlatılır, orta parantez içindeki sayılar ise satır ve sütun sayısını belirtir.
- lcd.print (veri) → Bu yönergenin tek bir parametresi vardır, o da LCD ekranda görüntülenecek metindir.

Örnek 4.35.

```
1 #include <LiquidCrystal.h>           // LCD ekranile çalışma kütüphanesinin
                                     // çağırılması.
2 LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2); // Arduino Uno R3'ün LCD
                                     // ekranıyla bağlanacağı pinlerin
                                     // seçimi.
3 void setup() {
4   lcd.begin(16,1);                 // Ekranın 16 sütunu ve bir
                                     // satırı vardır.
5   lcd.print("hello, world!");
6 }
7 void loop() {}
```

Bu modüler birimin pratik kısmında kullanımıyla ilgili uygulamalı alıştırma öngörülmektedir ve o zaman LCD ekranın Arduino Uno R3'e nasıl bağlanacağını ve ekranda metin görüntüleme programını daha ayrıntılı olarak öğreneceğiz.

Servo motor, daire şeklinde dönmeyen, açısal kaydırım (hareketler) gerçekleştiren ve bir sonraki yönergeye kadar belirli bir konumda kalan özel bir motor türüdür. Genelde servo motor 180 derece döner. Servo motorlar robotikte sıklıkla kullanılır, örneğin robotik kolu hareket ettirmek için.



Şekil 4.11. Servo motorun dönüş açısının darbelerin genişliğine bağlılığı

Servo motorun belirli bir açıda dönebilmesi için genişliği bir ila iki milisaniye arasında değişen genişlik modülasyonlu darbeler ihtiyacı vardır. Servo motorların hassasiyeti nedeniyle analogWrite yönergesi ve genişlik modülasyonlu darbeler yerine Servo kütüphanesi ve yönergeleri kullanılır.

- # include (Servo.h)** → Bu yönerge ile, programa servo motor kontrol kütüphanesini dahil edilir.
- Servo ad** → Servo kütüphanesinin içerdiği fonksiyonların yürütüleceği servo nesnesi oluşturuyoruz. Değişkenlerde olduğu gibi nesnelere de sembolik, benzersiz adlar verilir.
- servo.attach (pin, min, max)** → pin, servo motorun bağlı olduğu Arduino platformunun pin numarasıdır. Yalnızca pin 9 ve 10 servo motor bağlantısını destekler. Min ve max zorunlu parametreler değildir ve servo motorun minimum ve maksimum dönüş açısı için mikrosaniye olarak ifade edilen darbe genişliğini temsil eder. Darbelerin genişliği ne kadar büyükse açı da o kadar büyük olur.
- servo.write(aci)** → Servo motorun dönüş açısı 0 ila 180 derece arasında bir değere sahip olabilir. Açı 0 olduğunda yer değiştirme bir yönde en büyük olur ve 180° olduğunda yer değiştirme diğer yönde en büyüktür. 90 derece için yer değiştirme yoktur.

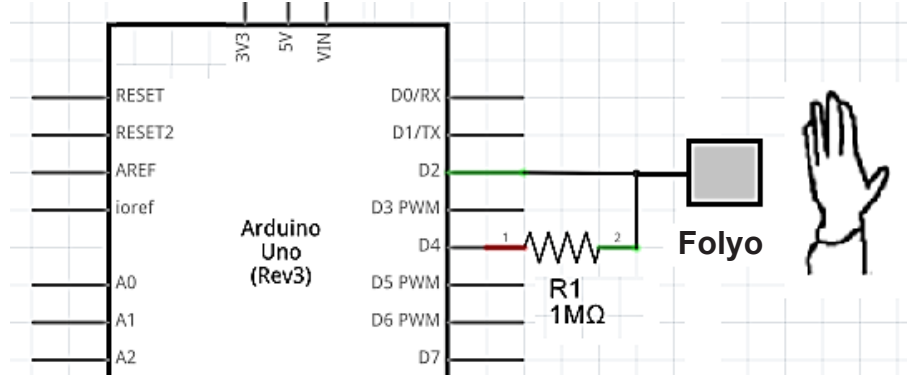
Örnek 4.36.'da yukarıdaki yönergeler kullanılmıştır ve servo motorun bağlanması için dokuzuncu pin seçilmiştir ve servo motor ortada, 90 derecelik pozisyonda bulunmaktadır.

Örnek 4.36.

```
1 #include <Servo.h>
2 Servo myservo;           // myservo embolik isimli servo nesne.
                           // oluşturuyoruz
3 void setup(){
4   myservo.attach(9);
5   myservo.write(90);     // Servo motor tam ortada
                           // konumlandırılmıştır.
6 }
7 void loop() {}
```

Pratik kısımda servo motor kullanımıyla ilgili uygulamalı alıştırmaya öngörül-
müştür.

CapacitiveSensor kütüphanesi ile Arduino Uno R3 platformunun iki veya daha fazla pini, insan vücudunun kapasitansını tespit etmek için kapasitif bir sensöre dönüştürülebilir. Bunun için giriş ve çıkış pinleri arasında yüksek dirençli rezistör ve giriş pininin ucuna bir parça alüminyum folyo bağlamak gerekir. [5]



Şekil 4.12. CapacitiveSensor kütüphanesinin uygulanması

İnsan folyo parçasına yaklaşırsa, vücudu elektriğin bir kısmını emecek, bu nedenle pinlerden birinin durumu değişecektir. Arduino Uno platformu, iki pinin durumunun eşitlenmesi için gereken süreyi ölçer. İnsan vücudu daha büyük miktarda elektrik emerse, bu süre de daha uzun olacaktır. CapacitiveSensor kütüphanesinden üç yönerge ile tanışacağız.

- # include (CapacitiveSensor.h)** → Bu yönerge ile, programda kapasitans kontrol kütüphanesi dahil edilir.
- CapacitiveSensor ad (pin sender, pin reciver)** → Sembolik adlı nesne oluşturuyoruz. CapacitiveSensor kütüphanesinden kullanılan her yönergeden önce nesnenin adı belirtilir. Ayrıca bu yönergeyle alüminyum folyoya bağlanacak pinleri yapılandırıyoruz. Aralarına 1MΩ ila 40MΩ direnci olan bir rezistör yerleştirilir. Sensörün hassasiyeti direncin artmasıyla artar. 1MΩ dirençle sensör, temas sensörü olarak çalışır, 10MΩ dirençle ise maksimum mesafe 15 cm olmak üzere yakınlık sensörü olarak çalışır.

CapacitiveSensor
(örnek sayısı)



Bu, sensörün değerini okumak için yönergedir. Yönergenin önünde, yönergeden nokta ile ayrılmış olarak nesnenin adı belirtilir. Küçük parantezin içinde, genellikle 30 olan örnek sayısı yazılır.

Örnek 4.37. için Arduino Uno R3'ün ikinci ve dördüncü pinlerine bağlı kapasitif sensör kullanılmıştır. Folyoya dokunduğumuzda seri monitörün ekranına gelen değerlerin daha yüksek olması gerekiyor.

Örnek 4.37.

```
1 #include <CapacitiveSensor.h> // Kapasitif sensörle çalışmak için
// kütüphanenin çağırılması.
2 CapacitiveSensor senDopir = CapacitiveSensor(4,2);
3 //senTemas Sembolik isimli nesnenin oluşturulması. Arduino Uno platformunun
//kapasitif sensöre bağlanması için pin seçimi
4 void setup() { // Seri monitör için bant genişliğini
5   Serial.begin(9600); // seçiyoruz.
6 }
7 void loop() {
8   long sensorValue = senDopir.capacitiveSensor(30);
9   // Sensör değerinin okunması.
10  Serial.println(sensorValue); // Sensör değerinin seri monitörde
11 } // görüntülenmesi.
```

4.4. Raspberry Pi Mikrobilgisayarın Python Programlama Dili ile Programlanması

Python programlama dili özellikle gençler arasında en popüler programlama dillerinden biridir. Bunun nedeni, sözdiziminin İngilizce dilinin sözdizimine çok benzer olması nedeniyle **kolay anlaşılır olmasıdır**. İngilizce dilinin bilinmesi, Python programlama diline hızlı bir şekilde hakim olmak için ön koşuldur. Örnek 4.38. çıkış pinine diyotun bağlı olduğu Raspberry Pi için kısa bir programlama kodunu temsil etmektedir. Programlama kodu çok daha akıcıdır. İfadelerde, ifadeleri açıklamak için kısaltmalar kullanılmaz, tam kelimeler kullanılır. C/C++ programlama dilinde olduğu gibi ifadeler noktalı virgülle bitmez.

Örnek 4.38.

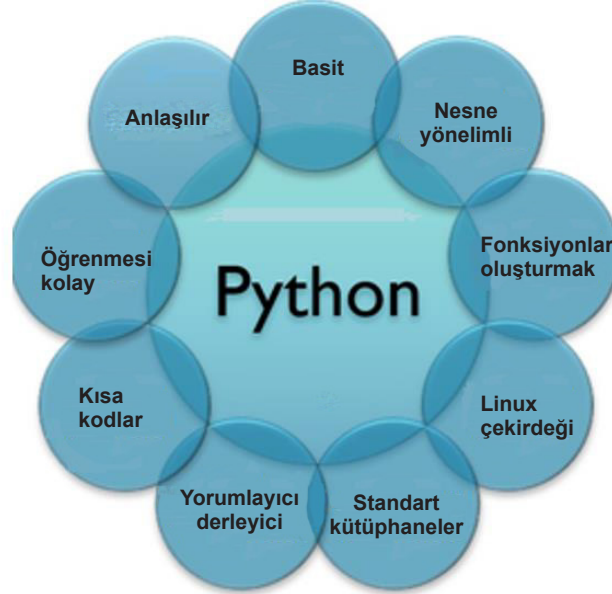
```
1 from gpiozero import LED
2 from time import sleep
3 led = LED(17)
4 while True:
5     led.on()
6     sleep(1)
7     led.off()
8     sleep(1)
```

Python programlama dili, C/C++ programlama diline göre daha üst düzeydedir. Python programlama dilinde yazılmış bir programın çevirisi derleyici tarafından değil, sistem program **yorumlayıcısı** tarafından gerçekleştirilir. Derleyici program kodunu hemen makine diline dönüştürüyor, yorumlayıcı ise program yürütülürken ifadeleri birer birer çevirir. Derleyici tarafından çevrilen program kodları daha hızlı yürütülür. Python programlama dili Linux işletim sistemiyle daha uyumludur ve Windows işletim sisteminde kurulumu biraz daha karmaşıktır. Hatırlayalım, Raspbian işletim sisteminin çekirdeği (İng. cernal) Linux'tur.

Python programlama dilindeki program kodları daha kısadır çünkü **değişken bildirim yoktur**. C/C++ programlama dilinde x değişkeni için `int x=5` yazıyoruz, Python programlama dilinde ise sadece `x=5` yazılır. Ayrıca veri türlerinde herhangi bir sınırlama yoktur, yani iki tam sayıyı bölersek, sonucu sadece bir tam sayı değil, ondalık sayı olarak hemen elde edebiliriz.

if...else veya while gibi bir yapı çerçevesinde yönerge blokları yazdığımızda büyük parantezler kullanılmaz, yönergeler içeriye doğru dört boşluk girintili olarak yazılır. Yapıyı bildirdikten sonra, programcının iki nokta üst üste (:) tuşuna basması

gerekir ve metin düzenleyicisi metni kendisi girintiler. Program kodu yorumları iki eğik çizgiyle değil, “#” diyez işaretiyle ayrılır.



Şekil 4.13. Python programlama dilinin özellikleri

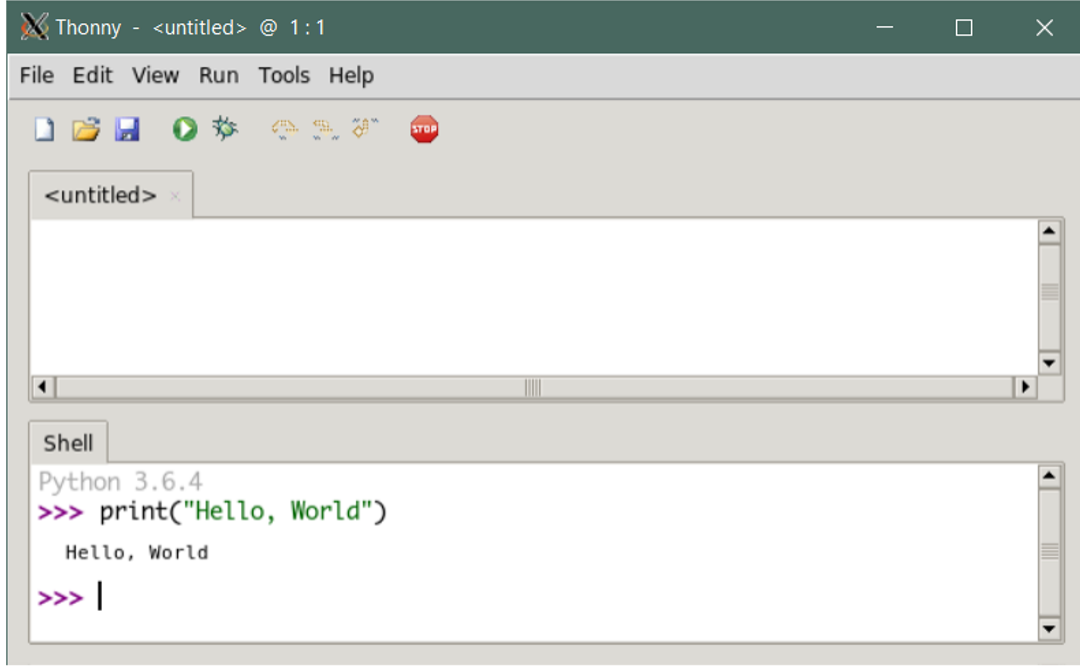
Python programlama dilinde programcı def yönergesi aracılığıyla **kendi fonksiyonunu oluşturabilir** ve daha sonra onu programın farklı yerlerinde çağırabilir. Örnek 4.39.'da gösterilen program kodu bu şekildedir.

Örnek 4.39.

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

C/C++ programlama dili gömülü sistemleri programlamak için daha iyidir, Python programlama dili ise daha evrenseldir ve onunla grafik uygulamalar da oluşturabiliriz. C/C++ donanım odaklıdır, Python ise yazılım odaklıdır. Python programlama dili, çeşitli amaçlı çok sayıda kütüphaneye sahiptir. Her kütüphanenin çok büyük yönerge seti vardır. turtle kütüphanesi grafik tasarım için kullanılır (örneğin dönen bir yıldız çizebiliriz). Pygame kütüphanesi görsel ve ses efektli video oyunlar oluşturmak için kullanılır. OpenCV görüntüler işlemek için bir kütüphanedir. Cloud4rpi, bulut teknolojisinde (İng.cloud) veri depolamak için kullanılan bir kütüphanedir. Şu anda Python programlama dili için 137.000 kütüphanenin bulunduğunu belirtmek yeterlidir. Sadece uygun kütüphaneyi seçmek yeterli değil, aynı zamanda içindeki komutları da öğrenmek gerekiyor. Biz, Raspberry Pi'ye giriş-çıkış cihazları (diyotlar, düğmeler, sensörler, motorlar, ekranlar) bağlandığında kullanılan GPIO Zero kütüphanesinin sunduğu olanakları inceleyeceğiz.

Üçüncü konuda Raspbian işletim sisteminin kurulumunu ve masaüstünü tanıdık. Menüde, daha doğrusu Programlama kategorisinde, Raspberry Pi'nin Python programlama dilinde programlanması için kullanılan entegre geliştirme ortamı Thonny Python IDE bulunmaktadır. Şekil 4.14.'te geliştirme programını açtıktan sonraki pencere gösterilmektedir.



Şekil 4.14. Raspberry pi'yi Python programlama dilinde programlamak için Thonny geliştirme programı

Arduino geliştirme ortamında olduğu gibi, Thonny Python IDE geliştirme programı da metin düzenleyici ve hata ayıklama konsolu içerir. Metin düzenleyicide programı yazıyoruz, hata ayıklama konsolunda ise programın çalıştırılmasıyla elde edilen sonucu görüyoruz. New düğmesine basarak yeni bir program başlatıyoruz ve programı oluşturulduktan sonra Save As düğmesine basarak kaydediyoruz.

4.5. GPIO Zero Kütüphanesi

İkinci konuda, donanımla tanışırken Raspberry Pi 3B+'nın giriş-çıkış cihazlarına bağlanması için GPIO (İng. General Purpose Input Output) kısaltmasıyla bilinen **40 pinli konektörün** kullanıldığını belirtmiştik. Bu konektörde, **26 pin giriş-çıkış cihazlarıyla bağlanmak için, geri kalanlar ise topraklama ve 3.3 V veya 5 V'luk güç kaynağı içindir.**

Raspberry Pi 3B+'nın GPIO pinlerine bağlı giriş-çıkış cihazlarıyla iletişim kurulabilmesi için GPIO Zero adında kütüphanenin dahil edilmesi gerekmektedir. Bu

kütüphane Raspbian işletim sistemi için standart bir kütüphanedir ve ek kurulum ihtiyacı yoktur. [13]

Bu kütüphaneye göre pin numaraları fiziksel pin numaralarından farklıdır. Aşağıdaki şekilde GPIO pinleri olarak adlandırılanların numaralandırması gösterilmiştir. Örneğin, fiziksel 15 numaralı pin, 22 numaralı GPIO pinine karşılık gelir.



Şekil 4.15. GPIO pinlerinin işaretlenmesi ve işlevsel açıklaması

Arduino Uno R3'ü C/C++ ile programlandığı gibi, Raspberry Pi 3B+'ı da Python'da programlarken, içerisinde kullanılacak kütüphanelerin programın başında dahil edilmesi gerekmektedir. Bu amaçla include yönergesi kullanılır. Örnek 4.40.'da düğme bağlantı pini yapılandırılırken, gpiozero öneki eklenmiştir.

Örnek 4.40.

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

Python programlama dilinde, kütüphanenin tamamı yerine kütüphaneden sadece sınıfın dahil edilmesine izin verilmektedir. Örnek 4.41.'de kitaplığın sadece düğme kontrol komutlarını içeren kısmı dahil edilmiştir. Bu durumda gpiozero öneki olmadan programdaki düğmeleri doğrudan kullanabiliriz.

Örnek 4.41.

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

Tek bir düğme yerine klavyenin tamamıyla çalışma imkanı vardır. Bu durumda gpiozero kütüphanesinden ButtonBoard olarak bilinen ve düğmeli pano anlamına gelen sınıf çağrılmalıdır.

Örnek 4.42.

```
1 from gpiozero import ButtonBoard
```

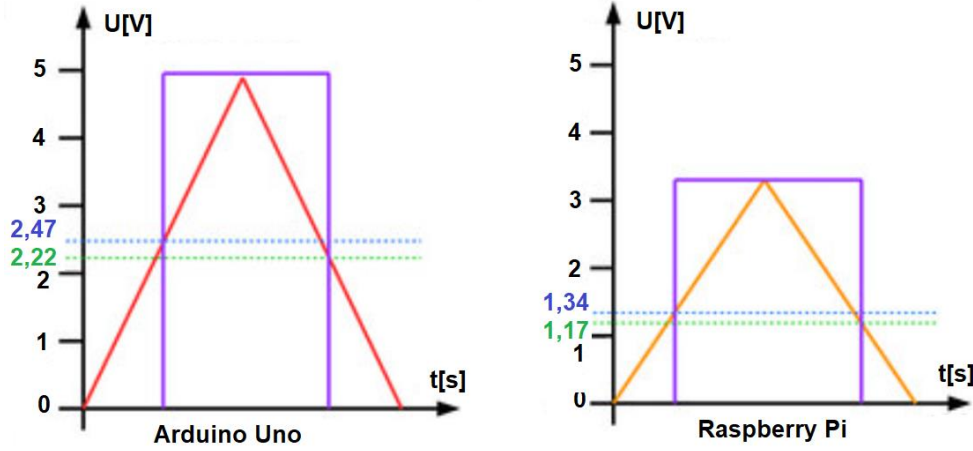
GPIO kütüphanesinin giriş-çıkış cihazlarının türüne bağlı olarak sınıflara ayrıldığından daha önce bahsetmiştik. Aşağıda sınıflar ve bunlarla ilgili cihazlar gösterilmiştir.

Giriş cihazları	→	Düğme, kızılötesi sensör, hareket sensörü, optik sensör, mesafe sensörü
Çıkış cihazları	→	LED diyot (RGB veya darbe-genişliği darbeler kontrollü), sesli uyarıcılar - ziller, motorlar, servo motorlar
Seri çevresel cihazlar	→	Analog-dijital dönüştürücü
Platformlar ve ekler	→	LED ekranlar, klavyeler, robotlar ve uzaktan kumandalı ev kullanımı için denetleyiciler
İç bileşenler	→	Zaman ayarı, işlemci sıcaklığı, ortalama değerlerin hesaplanması, disk kullanımı vb.
Ses üreticiler	→	Basit veya karmaşık müzik sesleri oluşturma

4.6. GPIO Zero Kütüphanesinin Giriş Cihazlar Sınıfları

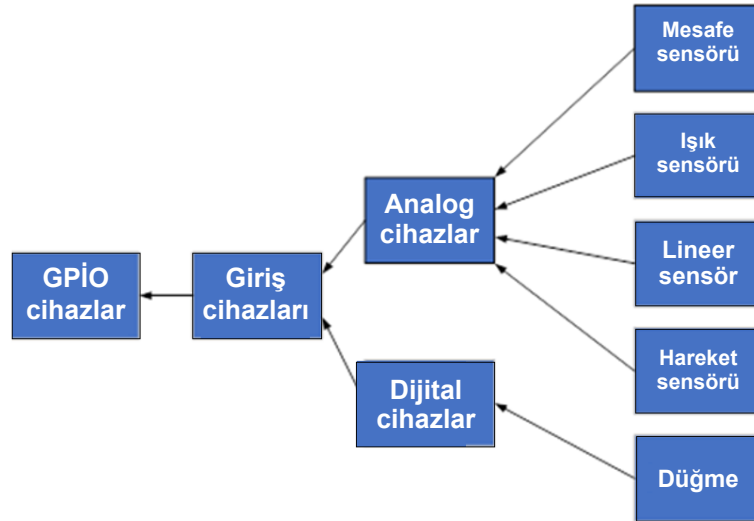
Dijital sinyallerin sadece iki değeri vardır, bir veya sıfır, LOW veya HIGH. Analog sinyaller, ondalık sayılar da dahil, 0 ile izin verilen maksimum gerilim aralığında farklı değerler alabilir. Raspberry Pi, Arduino platformu gibi yerleşik analog-dijital dönüştürücüye sahip değildir. Raspberry Pi girişlerinden analog değerleri okumak istiyorsak o zaman MCP3008 entegre devresi gibi dış analog-dijital dönüştürücü bağlamamız gerekir. Giriş analog cihazlarıyla çalışmak için eşik gerilimi (İng.threshold) olarak adlandırılan gerilim kullanılabilir. Eğer giriş gerilimi eşik gerilimden daha büyükse bu değer mantıksal bir (HIGH) olarak kabul edilir ve giriş gerilimi eşik gerilimden daha düşükse bu değer mantıksal sıfır (LOW) olarak kabul edilir. Şekil 4.16.'da Arduino Uno ve Raspberry Pi için eşik gerilimleri karşılaştırmalı olarak gösterilmiştir. Raspberry Pi'nin giriş pinlerindeki gerilim 3,3V'tan yüksek olmamalıdır. Aynı şekilde, eşik gerilimi giriş geriliminin azalmasına veya artmasına bağlı olarak değiştiğini görebiliriz. Giriş

gerilimi sıfırdan izin verilen maksimum değere yükseldiğinde eşik gerilimi Raspberry Pi için 1,34V ve Arduino Uno için karşılaştırmalı olarak 2,47V'tur. Giriş gerilimi izin verilen maksimum değerden sıfıra düştüğünde eşik gerilimi Raspberry Pi için 1,17V ve Arduino Uno için karşılaştırmalı olarak 2,22V olur.



Şekil 4.16. Arduino Uno ve Raspberry Pi için eşik gerilimlerinin karşılaştırılması

Şekil 4.17.'de giriş cihazlarının ürettikleri sinyallerin türüne göre ayırımı gösterilmiştir.

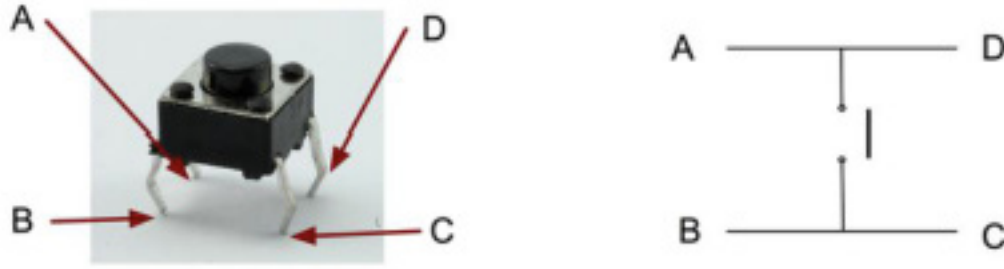


Şekil 4.17. Raspberry Pi için giriş cihazları

GPIOZERO kütüphane bileşimindeki tüm sınıflar böyle dallanma gösterir. Bu yaklaşım Raspberry Pi'nin programlanmasını büyük ölçüde basitleştirir.

4.6.1. Düğme (İng.Button)

Aynı hatta ancak ters tarafta bulunan düğme bağlantılarının kısa devre yaptığını hatırlayalım. Bu, Şekil 4.18'de gösterilmektedir.



Şekil 4.18. Düğme bağlantı noktalarının iç bağlantısı

Örnek 4.43. düğmenin basılmış olması koşuluyla bir satırı metnin görüntülenmesi için program kodunu temsil etmektedir.

Örnek 4.43.

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

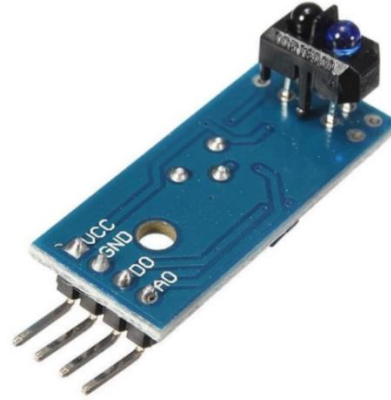
Bu program kodunda iki düğme ile işlem yönergesi kullanılmıştır: Button() ve wait_for_press(). Düğme ile çalışmak için 12 başka yönerge daha vardır. Daha önemli yönergeleri ve onların küçük parantez içinde yazılan parametrelerini açıklayacağız.

- Button()** → Örnek 4.43.'te 4 sayısı **GPIO-pin numarasını** temsil etmektedir. Bu parametre zorunludur. Aksi halde hata ayıklama konsolu hata bildirecektir. Ayrıca pull_up parametresinden de bahsedeceğiz. Bu parametrenin true veya false olmak üzere iki durumu olabilir. Standart true değeri, düğmeye basıldığında yüksek seviye (HIGH) demektir. Bu durumda düğmenin bir pini topraklamaya, diğer pini ise GPIO pinine bağlanır. Pull_up parametresi false ise düğmeye basıldığında düşük seviye (LOW) elde edilir ve ilk pin topraklama yerine 3,3V'luk gerilim kaynağına bağlanır.
- wait_for_press ()** → Belirtilen süre geçene veya düğmeye basılana kadar **program kodunun yürütülmesi durdurulur.**
- wait_for_release()** → Belirtilen süre geçene veya düğme bırakılana kadar program kodunun yürütülmesi durdurulur.

held_time	→	Eğer düğmeyi uzun süre basık tutarsak, bu yönerge bastırma süresini saniye olarak gösterir.
hold_time	→	Bu yönerge düğmeye basıldıktan sonra kaç saniye bekleneceğini belirler.
is_held	→	hold_time yönergesi ile belirlenen sürenin bitiminden sonra değer doğru (true) olur.
is_pressed	→	Bu yönergenin sonucu boolean değeridir, true veya false. Değer sadece düğme basılı olduğunda doğrudur (true).

4.6.2. Kızılötesi yansıtıcı izleme modülü (TRCT5000)

Kızılötesi yansıtıcı izleme modülü, Şekil 4.19'da gösterilmektedir. Kesintisiz çizgi tespiti için kullanılabilir. Modülün dört konektörü vardır, ancak Raspberry Pi'ye bağlantı için üç tane kullanıyoruz: Vcc, GND ve DO (İng.Data Out). Vcc pini 3,3V kaynağa bağlanır, GND topraklamadır. DO pini sinyal pinidir ve GPIO pinlerinden biriyle bağlanır.



Şekil 4.19. Kızılötesi yansıtıcı izleme modülü

Örnek 4.44. sensör tarafından okunan değere bağlı olarak bir satır metin görüntüleyen programlama kodunu temsil eder.

Örnek 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) → Bu yönerge ile sensör bağlantı **pinini yapılandırıyoruz**

- when_line → Yönerge, sensör aktif olduğunda yürütülür.
- when_no_line → Yönerge, sensör aktif olmadığında yürütülür
- lambda → Bu, bir eylemi belirtmek için başka bir yönergenin alt yönergesi olarak kullanılan standart bir Python yönergesidir. Bağımsız yönerge olarak kullanılmaz.

4.6.3. Mesafe sensörü (HC-SR04)

Mesafe sensörü, sensörün önüne yerleştirilen bir nesne tarafından yansıtılan ve sensöre geri gönderilen yönlü bir dalga gönderen ultrasonik sensördür. Vericinin dalgayı gönderdiği andan alıcının yansıyan dalgayı aldığı ana kadar geçen süre ölçülür. Bu parametreye göre sensör ile nesne arasındaki mesafe hesaplanır. **Gönderilen dalgaya tetikleyici (aktivatör), yansıyan dalgaya ise yankı denir.**



Şekil 4.20. Mesafe sensörü (HC SR04)

Örnek 4.45. HC SR04 için cm olarak ifade edilen mesafe hakkında bilgi veren bir program kodudur. Tetikleme pini GPIO17 pinine bağlıdır.

Örnek 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()** → İki parametre zorunludur. İlk parametre yankı için, ardından ise tetikleyici için GPIO pin numarasıdır. Onlar küçük parantez içinde yazılırlar.
- wait_for_in_range** → Program kodunun yürütülmesi, mesafe eşik mesafesinin (İng.threshold_distance) üzerine olduğu sürece veya küçük parantez içinde belirtilen süre geçene kadar program durdurulur. Eşik mesafesi

- genellikle 0,3 m'dir ve parametre olarak Distance-Sensor() yönergesiyle ayarlanabilir.
- wait_for_out_of_range** → Program kodunun yürütülmesi, mesafe eşik mesafesinin altına düştüğünde veya belirlenen süre geçene kadar durdurulur.
- distance** → Bu yönergenin sonucu sensör tarafından ölçülen mesafedir ve bu mesafe metre olarak verilir.
- when_in_range** → Bu yönergenin ardından, eşittir işareti gelir ve koşulun karşılanması durumunda yürütülecek başka bir yardımcı yönerge gelir.
- when_out_of_range** → Yönerge, mesafenin eşik mesafesinin üzerinde olması durumunda bir eylem tanımlar.

4.6.4. Optik Sensör veya Fotorezistör (İng. LDR-Light Dependet Resistor)

Fotorezistör, ışık algılama için en yaygın kullanılan sensördür. Direnci ışıktaki 1KΩ'dan ve karanlıkta 100KΩ'a kadar değişir, yani ışık şiddeti arttığında azalır. İlerleyen bölümde uygulamasıyla ilişkin uygulamalı alıştırma gerçekleştireceğiz, şimdi ise bu sensörle çalışmak için birkaç yönergeyle tanışacağız.

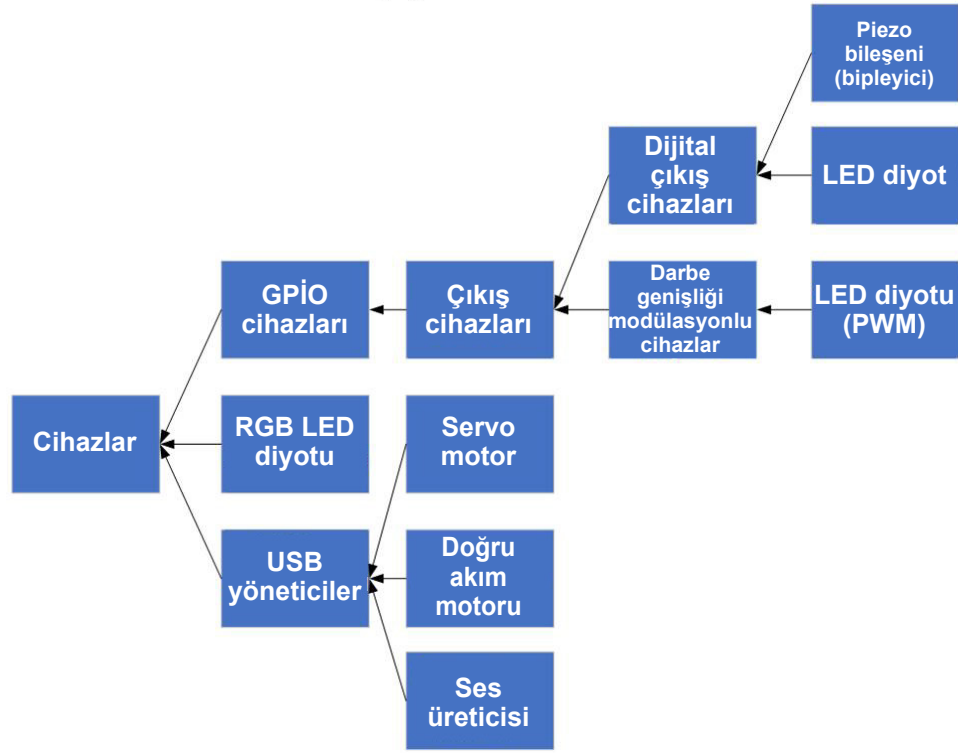
Örnek 4.46.

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

- LightSensor()** → Küçük parantez içinde GPIO pin numarası belirtilir. Örnek 4.46,'da ldr sensörün sembolik adıdır ve adı programcı tarafından verilir.
- wait_for_light()** → Bu yönerge ile sensör aktif hale gelene kadar program kodu durdurulur. Küçük parantez içinde sensörün aktif hale gelmesi için geçmesi gereken süre de belirtilebilir.
- wait_for_dark ()** → Bu yönerge ile, Karanlık durumu meydana gelene kadar program kodu durdurulur.

4.7. GPIO Zero Kitaplığından Çıkış Cihazlar Sınıfları

Şekil 4.21.'de çıkış cihazlarının ürettikleri sinyallerin türüne göre bölümü gösterilmiştir.



Şekil 4.21. Raspberry Pi için çıkış cihazları

Çıkış cihazlarının bulunduğu ağaç daha da dallanmıştır. Konektör tipine göre, GPIO konektörlü çıkış cihazları ve USB konektörlü cihazlar olmak üzere başka bir bölüm yapılır. GPIO konektörlü cihazları dijital çıkış cihazlarına ve darbe genişliği kontrolü (PWM) cihazlarına bölünmüştür.

Dijital cihazların açılması ve kapanması için iki değer gereklidir: LOW veya HIGH. Darbe genişliği kontrolü ile LED diyotunun ışık yoğunluğunu değiştiriyoruz. Giriş cihazlarda olduğu gibi çıkış cihazlarında da gpiozero kütüphanesi her cihaz için ayrıdan özel yönerge seti içerir.

4.7.1. LED Diyot

LED diyotu Raspberry Pi'ye bağlarken **uzun pinin (anot) GPIO pinine, kısa pinin (katot) ise topraklamaya** bağlandığına dikkat etmemiz gerekir. Ayrıca, akımı sınırlamak için rezistör de kullanılmalıdır.

Örnek 4.47'deki program kodu, LED diyotun açılmasını gerçekleştirir.

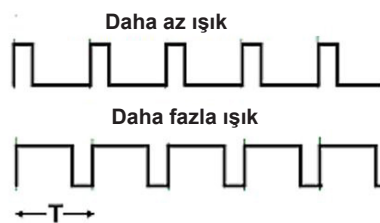
Örnek 4.47.

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

- LED()** → Küçük parantezde GPIO pin numarası, aktif ve başlangıç değeri belirtilir. Aktif değer HIGH ise diyot yukarıdaki örnekteki gibi yanacaktır. `active_high=false` ise LED diyoyun yanması için `off()` yönergesi gerekir. `led`, programcı tarafından verilen diyotun sembolik adıdır.
- blink** (`on_time=1,`
`off_time=1`) → Bu yönerge ile diyot yanıp sönmeye başlar. **Açılma ve kapanma sürelerini** değiştirebiliriz. **Standart olarak bu süre bir saniyedir.**
- off()** → Bu yönerge LED diyotu kapatıyor
- on()** → Bu yönerge LED diyotu açıyor
- toggle()** → Bu yönerge **LED diyotun durumunu değiştiriyor**, açıksa kapatıyor ve tersi.

4.7.2. Genişlik Modülasyonlu Darbeler ile Kontrol Edilen LED Diyotu (PWMLLED)

LED diyotunun ışık yoğunluğunu değiştirmek istiyorsak, GPIOZero kütüphanesinden PWMLLED sınıfını çağırmamız gerekir. Periyodik dizi darbeleri daha büyük genişliğe sahip olduğunda parlaklık daha fazladır. Darbelerin genişliğini potansiyometre yardımıyla değiştirebiliriz.



Şekil 4.22. Işık yoğunluğunun darbe genişliğine bağımlılığı

Programda ışık yoğunluğunu değiştirmek istiyorsak gpiozero kütüphanesinden PWMLLED sınıfı dahil edilmelidir. PWMLLED() yönergesinde küçük parantezde bağlanılacak GPIO pininin numarası, periyodik dizinin frekansı ve başlangıç değeri

belirtilir. Frekans belirtilmezse standart olan 100 Hz frekansı geçerli olacaktır. Işık yoğunluğunu kontrol etmek için sıfırdan bire kadar değerler kullanılır.

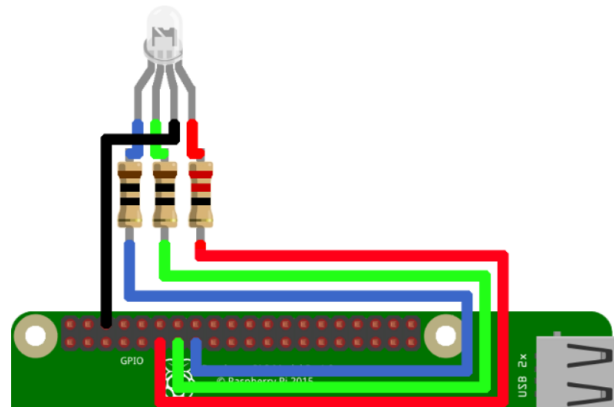
Örnek 4.48.

```
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)
```

4.7.3. Değişken Renkli LED Diyot (RGB-LED)

RGB kısaltması kırmızı, yeşil ve mavi anlamına gelen red, green, blue kelimelerinden gelir. Bu üç temel rengin karıştırılmasıyla tüm diğer renkler elde edilir. Her ana rengin 256 tonu vardır ve bu tonların karıştırılmasıyla 16.777.216 farklı renk elde edilebilir. Her bir ana renk, değeri 0'dan 1'e kadar olan ondalık sayı olan bir değişkeni temsil eder. **Diğer renkler, temel renklerin birleşimi olarak elde edilir.** Örneğin (1, 0, 0) kombinasyonu kırmızı rengi temsil eder, (0, 1, 0) kombinasyonu yeşil rengi temsil eder, (1, 1, 0) kombinasyonu sarı rengi olacak ve (1, 0.5, 0) kombinasyonu turuncu rengi olacaktır. Renklerin daha kolay kullanılması için Python programlama dilinin standart kütüphanesi olan color kütüphanesi kullanılabilir.

RGB LED diyotun dört bağlantı noktası vardır, üç anot ve daha uzun olan bir katodu vardır. Katot topraklamaya bağlanır, anotlar ise üç ana rengin her biri için birer pin olmak üzere üç GPIO pinine bağlanır. Örnek 4.49'daki program koduyla. LED diyot sarı renkte yanacaktır. Örnek 4.50. color kütüphanesinin kullanımına ilişkin program kodudur.



Şekil 4.23. RGB-diyotun Raspberry Pi ile bağlanması

Örnek 4.49.

```

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

```

Örnek 4.50.

```

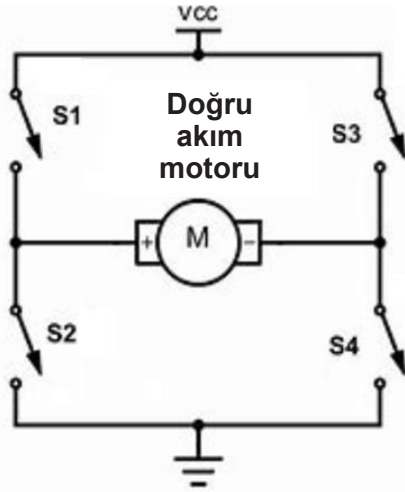
1 from gpiozero import RGBLED
2 from colorzero import Color
3 led = RGBLED(2, 3, 4)
4 led.color = Color('yellow')

```

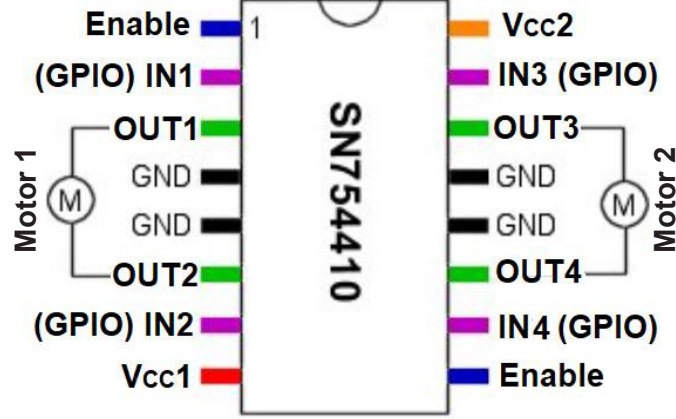
Normal LED diyot ile aynı yönergeler RGB LED için de geçerlidir, ancak üç GPIO pininin yapılandırılması ve LED diyotun renginin color() yönergesiyle tanımlanması gerekir. RGBLED() yönergesinde pwm=True parametresi kullanılarak genişlik modülasyonlu darbeler dahil edilirse RGB-LED'in ışığı yoğunluğa göre değiştirilebilir.

4.7.4. Doğru Akım Motoru

Akımın yönü değişirse, doğru akım motorunun dönüş yönü de değişecektir. Doğru akım motoru, **Raspberry Pi'ye H-köprüsü aracılığıyla bağlanır**. H-köprüsünün çalışma prensibi şekil 4.24'te gösterilmektedir.



Şekil 4.24. H-köprüsünün çalışma prensibi



Şekil 4.25. H-köprüsünün pin diyagramı

Motor bir doğrultucudur ve gerilimin kutupluluğunu değiştirebilir. S1 ve S4 anahtarları kapalı ve S2 ve S3 anahtarları açıksa motor saat yönünde döner. S1 ve S4 açık ve S2 ve S3 anahtarları kapalı ise motor ters yönde döner. Raspberry Pi ile H-köprüsü olarak çalışmak için en sıkça L293D veya SN754410 entegre devreleri

kullanılır. Şekil 4.25'te. SN754410 entegre devresinin pin diyagramı gösterilmiştir ve bu devre iki doğru akım motoru kontrol etmek için kullanılabilir. Motorlara tekerlekler bağlanırsa bu kurgu robotu hareket ettirmek için kullanılabilir. H köprüsünün çıkışları doğru akım motoruna, girişleri ise iki GPIO pinine bağlanır.

4.8.2.7. Doğru Akım Motorunun Dönme Yönünün Değişmesi uygulamalı alıştırmada, SN754410 entegre devresi kullanılmıştır ve motorun, H-köprüsünün, Raspberry Pi'nin ve devre tahtasının bağlanma şekli açıklanmıştır.

Devamda doğru akım motoruyla çalışmaya ilişkin bir kaç yönerge verilmiştir.

Motor()	→	Küçük parantez içinde, H-köprüsünün bağlanması için kullanılan GPIO pinlerinin numaraları yazılır.
backward(speed=1)	→	Doğru akım motoru saat yönünün ters yönünde döner. Genişlik modülasyonlu darbelerin kullanılması şartıyla dönüş hızının (speed) değeri 0 ila 1 aralığında değiştirilebilir . Bir, dönüş hızının maksimum değeridir.
forward(speed=1)	→	Doğru akım motoru saat yönünde döner.
reverse()	→	Motorun dönüş yönü değiştirilir.
stop()	→	Motorun dönmesi durdurulur.

Örnek 4.51'deki program kodu, motoru ileriye doğru, saat yönünde döndürür.

Örnek 4.51.:

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

4.7.5. Servo Motor

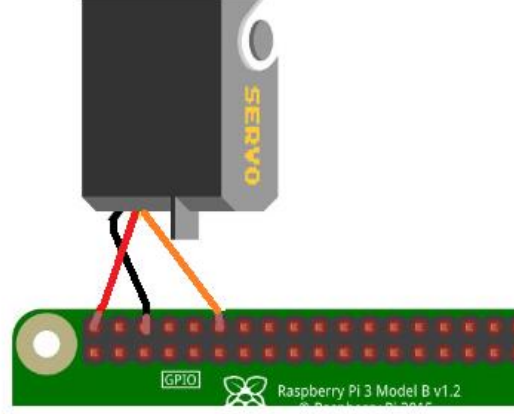
Hatırlayalım, servo motor bir daire içinde dönmez, kontrol sinyalinin büyüklüğüne bağlı olarak orta konumunun soluna veya sağına doğru, belirli bir açıyla hareket eder. Servo motorun çalışma yöntemi şekil 4.11'de gösterilmiştir.

Şekil 4.26.'da servo motorun dış görünümü gösterilmiştir, şekil 4.27.'de ise Raspberry Pi'nin GPIO pinlerine nasıl bağlanacağını göstermektedir. Servo motorun

üç konektörü vardır: kırmızı, siyah ve turuncu. Kırmızı kablo 5 volt'luk güç kaynağına, siyah kablo ise topraklamaya bağlanır. Turuncu kablo GPIO pinlerinden birine bağlanır ve bunun aracılığıyla Raspberry Pi motorla iletişim kurar, yani dönüş açısını kontrol eder.



Şekil 4.26. Servo motorunun dış görünümü



Şekil 4.27. Servo motorun Raspberry Pi ile bağlanması

Örnek 4.52.'deki program koduyla, servo motor 45 derece için dönecektir.

Örnek 4.52.

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

Sadece dönüş açısının değerini tanımlayan `servo.value()` yönergesini açıklayacağız. Bu değer -1 (minimum açı 0 derece) ile 1 (maksimum açı 180 derece) arasında değişebilir. Değer sıfır ise servo motor orta konumda, yani 90 derecede olacaktır.

4.8. Modüler Birimi İçin Uygulamalı Alıştırmalar: Mikrobilgisayarların Programlanması

4.8.1. Arduino Uno R3'ü C/C++ Programlama Dilinde Programlanmasına İlişkin Uygulamalı Alıştırmalar

Tüm alıştırmalar basittir ve Arduino geliştirme platformunda programlamaya ilişkin teorik bilgilerin pratik olarak uygulanması amacıyla yeni başlayanlara yöneliktir. Arduino Uno R3'ün başarılı bir şekilde programlanması için, donanım ve yazılım hakkında bilgi gerektirir. Uygulamalı alıştırmalara başlamadan önce aşağıdaki öğretim birimler tekrarlanmalıdır:

- 2.4. Arduino Uno R3'ün Elektronik Bileşenleri ve Onların Bağlanması
- 2.8.3.1. Arduino Tabanlı Mikrodenetleyici Platformu ile Çalışırken Koruma ve Güvenlik Önlemleri
- 2.8.3.2. Devre Tahtası Kullanım Kılavuzu
- 2.8.3.3. Arduino Uno R3 İçin Bilgisayar Simülasyonu
- 3.2.6.2. Arduino Uno R3 Platformunda Geliştirme Ortamının Menüsü ve Araçları ve Programın Yazılması

4.8.1.1. Uygulamalı Alıştırma: Düğme Aracılığıyla LED Diyotun Açılması

1. Alıştırmanın amacı

Alıştırmanın amacı, giriş ve çıkış bileşeni olarak bir LED diyotu ve bir düğmeyi, verilen işlevsel ve montaj şemasına göre ve devre tahtası kullanarak Arduino Uno R3'e bağlamaktır. Öğrencilerin geliştirme ortamında program yazmaları, programı Arduino Uno R3'ün belleğine yüklemeleri ve çalışmasını test etmeleri gerekiyor.

2. Bu alıştırmanın gerçekleştirilmesi için **gerekli bileşenler** şunlardır:

- arduino uno r3 platformu
- devre tahtası
- baskılı devre kartı düğmesi

- led diyot
- dirençleri $r_1=220\omega$ ve $r_2=1\omega$ olan rezistörler
- bağlama kabloları (jumper kablolar)
- arduino platformu için geliştirme ortamı kurulmuş bilgisayar

Düğme, Arduino Uno R3'e değerler gönderir (HIGH veya LOW), ve devamında platform LED diyotu yönetir. R1 rezistörü, zararlı ısınma olmadan ışığın yoğunluğu optimal olması amacıyla, diyotun çalışma akımını ayarlamak için gereklidir. R2 rezistörü, düğmenin bağlı olduğu pini gerilimdeki hızlı değişikliklerden korur.

3. Alıştırma için hazırlık

Öğrenciler aşağıdakileri hatırlatılmalıdır:

- Arduino Uno R3'ün pin diyagramı
- LED diyotu ve devre tahtası üzerindeki düğmeyi bağlama yöntemi. Bu, devre tahtası ile çalışma kılavuzunda açıklanmıştır.
- Arduino Uno R3 için giriş-çıkış yönergeleri, zorunlu yapılar (setup ve loop) ve olasılıklar arasında seçim yapısı if...else.

4. Elektrik şemasının ve bağlantı yönteminin açıklaması

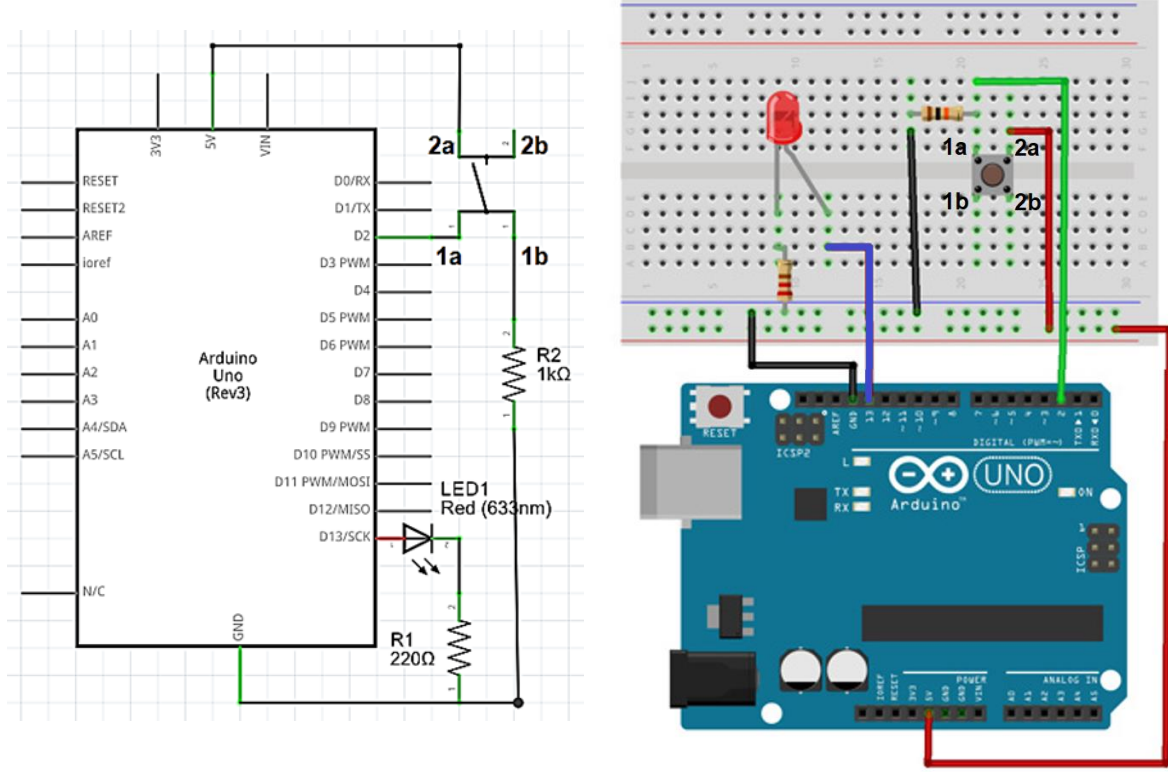
Şekil 4.28'de, bu alıştırmanın uygulanmasına ilişkin işlevsel ve montaj şeması gösterilmiştir.

Düğme basılı olmadığında Arduino Uno R3'ün ikinci pini düşük mantıksal seviyeye (İng. LOW) karşılık gelen topraklamaya bağlanır ve ikinci pinin gerilimi 0V'tur. Düğmeye basıldığında Arduino Uno R3'ün ikinci pini yüksek mantıksal seviyeye (İng. HIGH) karşılık gelen güç kaynağına bağlanır ve ikinci pinin gerilimi sıfırdan farklıdır. LED diyotu, 13. pin yüksek mantıksal seviyeye geldiğinde yani pinin çıkış gerilimi 5V olduğunda yanar.

Bileşenlerin bağlanması şu şekilde gerçekleştirilir:

- (1) Düğmeyi devre tahtasının ortasına yerleştiriyoruz. Düğmeyi bağlamak için üç jumper kablo kullanılır. Kırmızı jumper kablo ile 2a-2b bağlantı noktalarını güç kaynağı bantına bağlıyoruz (+ veya kırmızı renkle işaretli). Siyah jumper kablosu ve R2 rezistörü aracılığıyla düğmenin 1a-1b bağlantı noktalarını topraklama bantına (eksi veya mavi renkle işaretli) bağlıyoruz. Aynı zamanda yeşil jumper kablo ile 1a-1b bağlantı noktalarını Arduino Uno R3'ün ikinci dijital pinine bağlıyoruz. Yeşil jumper kablosu bir tür sinyal kablosudur ve bu kablo aracılığıyla düğme HIGH veya LOW değerlerini ikinci pine gönderir.
- (2) Anotun katottan daha yüksek potansiyele sahip temas noktasına bağlandığına dikkat ederek LED diyotu devre tahtası üzerine yerleştiriyoruz. Siyah jumper kablo ile R1 rezistörü aracılığıyla LED diyotun katotunu topraklama

bantına ve mavi jumper kablo ile anodu Arduino Uno R3'ün 13. pinine bağlıyoruz.



Şekil 4.28. Arduino Uno R3'e LED diyotun ve düğmenin bağlanması

(3) Kırmızı jumper kablo ile Arduino UnoR3'ün 5V pinini devre tahtanın güç kaynağı bantına bağlıyoruz, siyah kablo ile GND pinini topraklama bantına bağlıyoruz.

5. Arduino Uno R3 için programın yazılması ve girilmesi

Arduino Uno R3, digitalRead() yönergesini kullanarak düğmenin durumunu kontrol eder. LED diyot, digitalWrite() yönergesini kullanarak yazılımsal olarak açılır ve kapatılır. Program, iki olasılıklı bir seçim için if...else yapısı içerir (LED diyot yanıyor veya LED diyot yanmıyor). Düğmenin durumu if...else yapısının yürütülmesinin koşuludur

Yukarıdaki devreyi kontrol etme programı aşağıda verilmiştir.

```
// Sabitler ve değişkenler setup ve loop yapıları yazılmadan
// önce bildirilir
// Sabitler değişmiyor ve bağlantı pinlerini seçmek için kullanılır
1 const int buttonPin = 2;           // Düğme 2 numaralı dijital pine
                                     // bağlıdır 2.
2 const int ledPin = 13;            // LED diyotu 13 numaralı dijital
                                     // pine bağlıdır.
// Değişkenler, sürekli değişen verileri saklar.
```

```
3 int buttonState = 0; // ButtonState, düğmenin bağlı olduğu
// 2 numaralı pinin okunmasıyla elde
// edilen değerin adıdır.
4 void setup() { // setup yapısının başlangıcı.
5   pinMode(ledPin, OUTPUT); // LED diyotun bağlandığı pin çıkış
// pini olarak yapılandırılır, bu da
// Arduino veri göndereceği anlamına gelir.
6   pinMode(buttonPin, INPUT); // Düğmenin bağlı olduğu pin giriş
// pini olarak yapılandırılır, bu da
// Arduino veri alacağı anlamına gelir.
7 } // setup yapısının sonu.
8 void loop() { // loop yapısının başlangıcı.
9   buttonState=digitalRead(buttonPin); // 2 numaralı pinin değeri okunuyor
// ve bu değer buttonState
// değişkenine yazdırılır.
10  if (buttonState == HIGH) { // Düğmenin basılı olup olmadığı
// kontrol edilir.
11    digitalWrite(ledPin, HIGH); // Eğer basılıysa, LED diyot
// yanar.
12  } // if ifadesinin sonu.
13  else { // else ifadesinin başlangıcı.
14    digitalWrite(ledPin, LOW); // Eğer basılı değilse, diyot
// yanmaz.
15  } // Else ifadenin sonu.
16 }
```

Programın Arduino Uno R3 belleğine girilmesi tüm programlar için aynıdır ve 3.2.6.2 “Arduino Uno R3 Platformunda Geliştirme Ortamının Menüsü ve Araçları ve Programın Yazılması” öğretim biriminde adım adım açıklanmıştır.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Programı girdikten hemen sonra Arduino Uno R3’ün düzgün çalışıp çalışmadığı kontrol edilmelidir. Düğmeye bastığımızda LED diyot yanmalıdır. Düğmey, serbest bırakırsak, LED diyotun yanması durdurulur.

Yorum: _____

7. Değişiklik yapın!

- Programda iki digitalWrite yönergesi yerini değiştirirse ne olacak?

Yorum: _____

- İki digitalWrite yönergesinin her birinden sonra 5s’lik gecikme süresi girersek ne olacak? delay () yönergesi kullanılsın.

Yorum: _____

4.8.1.2. Uygulamalı Alıştırma: LED Diyotun Potansiyometre ile Kontrol Edilmesi

1. Alıştırmanın amacı

Bu alıştırma, Arduino Uno R3'ün analog girişlerinin pratik uygulamasını tanımlamaktadır. Öğrenciler, analog giriş sinyallerinin büyüklüğünün, Arduino Uno R3'ün dijital pinlerinin çıkış sinyallerinin süresini nasıl etkilediğini öğrenecekler. Potansiyometre yardımıyla LED diyotun yanıp sönme hızını değiştireceğiz.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Arduino Uno R3 platformu
- Devre tahtası
- 10 K Ω dirençli düzeltici potansiyometre
- LED diyot
- R1=220 Ω dirençli rezistör
- Bağlama kabloları (jumper kablolar)
- Arduino platformu için geliştirme ortamı kurulmuş bilgisayar

Bu uygulamalı alışırmada potansiyometre, Arduino Uno R3'ün analog girişlerinden birine bağlı olan ana bileşendir. Potansiyometre değişken dirençlidir ve onunla maksimum değeri 5 V ve minimum değeri 0 V olan değişken gerilim üretilir. Platformun içinde yerleşik analog-dijital dönüştürücü, analog değerleri 0'dan 1023'e kadar tam sayılara dönüştürür. Analog-dijital dönüştürücünün çıkış değeri ne kadar yüksekse, diyot o kadar seyrek açılıp kapanır, yani LED diyot daha düşük bir hızla yanıp sönecektir.

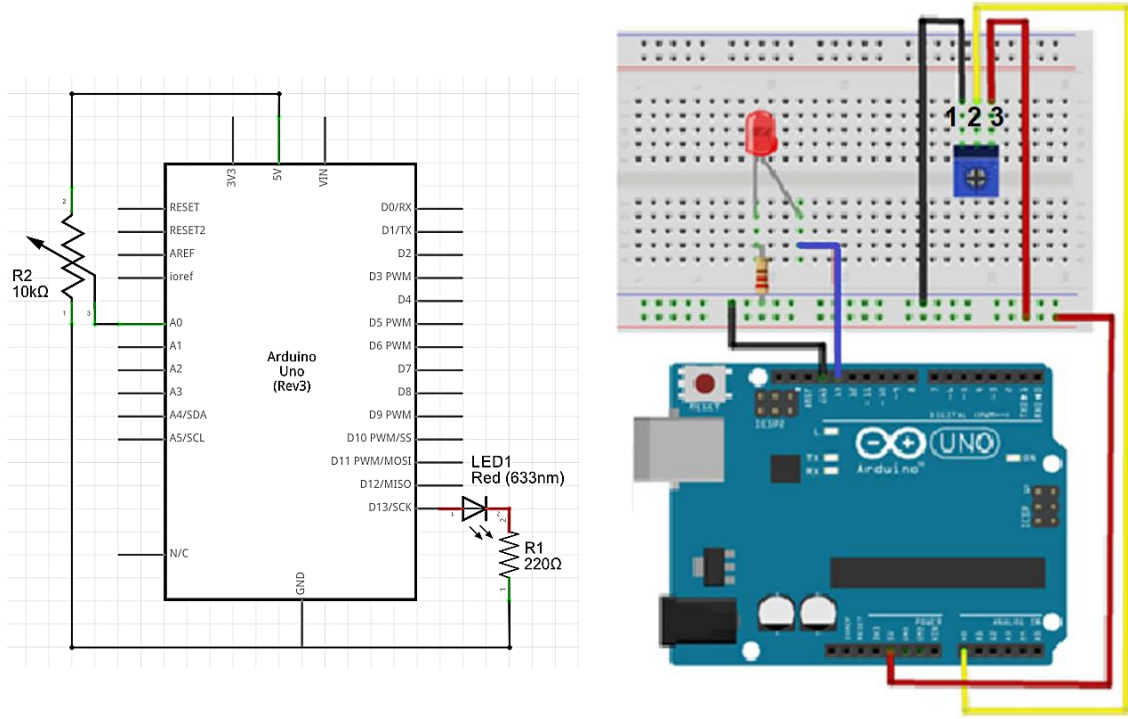
3. Alıştırma için hazırlık

Öğrenciler şunlar hatırlatılmalıdır:

- Darbe genişliği modülasyonu ve onun Arduino Uno R3'ün analog girişleri için önemi
- Arduino Uno R3'ün analog giriş pinleriyle çalışma yönergeleri
- delay() yönergenin anlamı ve gecikme süresinin seçimi

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.29.'da bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmiştir. Potansiyometrenin üç bağlantı noktası vardır ve bunlardan ikisi güç ve topraklama bağlantılarıdır. **Potansiyometrenin ortancı pini kontrol veya veri pinidir** ve onu Arduino Uno R3'ün **analog pini A₀'a** bağlıyoruz.



Şekil 4.29. LED diyotun ve düzeltici potansiyometrenin Arduino Uno R3'e bağlanması

Bağlanma yöntemi aşağıdaki gibidir:

- (1) Potansiyometreyi, bağlantı noktaları farklı iletkenlere ait temas noktalarına yerleştirmeye dikkat ederek, devre tahtası üzerine yerleştiriyoruz. Potansiyometrenin bağlantı noktaları 1'den 3'e kadar numaralandırılmıştır. 1 ve 3 numaralı bağlantı noktaları güç kaynağı ve topraklama bantlarına bağlanır. 2 numaralı pin, sarı jumper kablosu aracılığıyla doğrudan A0 analog pinine bağlanır.
- (2) LED diyotu önceki alıştırmada açıklandığı gibi aynı şekilde devre tahtası üzerine yerleştiriyoruz, yani katodu topraklama bantına, anodu ise 13. pine bağlıyoruz.
- (3) Arduino Uno R3'teki GND ve 5V işaretli pinleri devre tahtasının toprak ve güç kaynağı bantlarına bağlıyoruz.

5. Arduino Uno R3 için programın yazılması ve girilmesi

Arduino Uno R3, analogRead() yönergesini kullanarak A0 analog giriş pinin sinyalinin değerini kontrol ediyor. Okunan değer, sembolik adı sensorValue olan değişkeni temsil etmektedir. Bu değişken, LED diyotların durumunu değiştiren iki digitalWrite() yönergesinin ardından gelen iki delay() yönergesinde gecikme süresi olarak yazılır.

Aşağıda LED diyotun yanıp sönme sıklığını potansiyometre ile kontrol etme programı verilmiştir.

```
1 int sensorPin = A0;           // Potansiyometre için giriş pini
                                // seçimi.
2 int ledPin = 13;             // LED diyotun bağlanması için
                                // pin seçimi.
3 int sensorValue = 0;        // Potansiyometre ile elde
                                // edilen değerin saklanması
                                // için değişken bildirimi
4 void setup() {               //
5   pinMode(ledPin, OUTPUT);   // ledPin pini çıkış pini olara
                                // yapılandırılır
6 }                             // setup yapısının sonu
7 void loop() {                // döngünün başlangıcı.
8   sensorValue =analogRead(sensorPin);
// Potansiyometre ile okunan değer sembolik adı
// "sensorValue" olan değişkene yerleştirilir
9   digitalWrite(ledPin, HIGH); // LED diyot açılır.
10  delay(sensorValue);         // Diyot, potansiyometreden elde
                                // edilen değer kadar milisaniye
                                // boyunca yanar.
11  digitalWrite(ledPin, LOW);  // LED diyot kapanır
12  delay(sensorValue);         // Diyot, potansiyometre ile elde
                                // edilen değer kadar milisaniye
                                // boyunca yanmayacaktır.
                                // Daha sonra tekrar döngünün
                                // başlangıcına dönüyoruz.
                                //
13 }
```

Program yazıldıktan sonra daha önce anlatıldığı gibi Arduino Uno R3'ün belleğine girilir.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Arduino Uno R3 için güç kaynağı açıldığında, potansiyometrenin direncini minimumdan maksimuma ayarlamak için tornavida kullanılır. Direnç en yüksek olduğunda giriş sinyalinin değeri ve gecikme süresi en küçük olur ve LED diyot en yüksek hızda yanıp sönecektir. Gecikme süresi çok kısa ise diyotun yanıp sönmeye fark edilmeyebilir. Maksimum gecikme süresinin 1023 olduğunu vurgulayalım.

Yorum: _____

7. Değişiklik yapın!

- delay() yönergelerinde sensorValue değişkeni beş ile çarpılırsa ne olacaktır?

Yorum: _____

- Analog giriş sinyalinin değerini okuduktan sonra, Örnek 4.53'te gösterildiği gibi program koduna yeni bir `yuzde=map(val,0,1023,0,100)` yönergedi eklenirse ne olacaktır?

Örnek 4.53.

```
7 void loop() {
8   sensorValue =analogRead(sensorPin);
9   yuzde = map(val,0,1023,0,100);
10  digitalWrite(ledPin, HIGH);
11  delay(yuzde);
12  digitalWrite(ledPin, LOW);
13  delay(100-yuzde);
14 }
```

Yorum: _____

4.8.1.3. Uygulamalı Alıştırma: LED Diyotun Işık Yoğunluğunun Ayarlanması

1. Alıştırmanın amacı

Öğrenciler Arduino UnoR3'ün analog çıkış pinlerinden birini kullanacak ve yazılım üzerinden çıkış gerilimin büyüklüğünü değiştirecekler. Gerilim değişiklikleri LED diyot kullanılarak izlenecektir ve bu arada ışık yoğunluğu kademeli olarak minimumdan maksimuma kadar değişecektir ve bunun tersi.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Arduino Uno R3 platformu
- Devre tahtası
- LED diyot
- $R1=220\Omega$ dirençli rezistör
- İki bağlantı kablosu (jamper kablolar)
- Arduino platformu için geliştirme ortamı kurulmuş bilgisayar

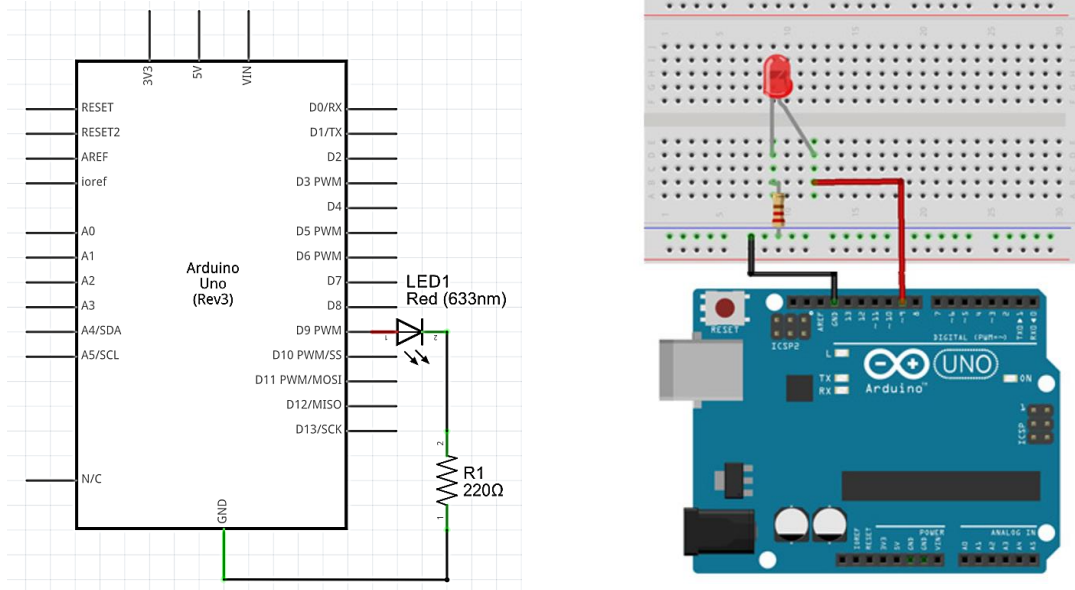
3. Alıştırma için hazırlık

Öğrenciler şunları hatırlatılmalıdır:

- Darbe genişliği modülasyonu, onun Arduino platformunun analog çıkışları için önemi ve olası değerlerin sayısı
- Arduino platformunun analog çıkış pinleriyle çalışma yönergeleri
- for döngü tekrarlama yapısı

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.30.'da bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmiştir.



Şekil 4.30. LED diyotun Arduino Uno R3'ten PWM pinine bağlanması

Devre çok basittir. LED diyotun katodu, R1 rezistörü aracılığıyla, topraklama bantına, anot ise Arduino Uno R3'ün dokuzuncu pinine bağlanmalıdır. Bu alıştırmada on üçüncü pini önceki alıştırmalarda olduğu gibi kullanamıyoruz çünkü ona analog değerler yazılamıyor.

5. Arduino Uno R3 için programın yazılması ve girilmesi

LED diyotun ışık yoğunluğunu kontrol eden çıkış gerilimi, orta parantez içine analog değer girildiği `analogWrite()` yönergesini kullanarak veriyoruz. Minimum analog değer 0, maksimum değer ise 255'tir.

Işık yoğunluğunu ayarlama programı, döngüyü tekrarlamak için iki "for" yapısı kullanır. İlk for yapısında, değişkeni beş için artırıyoruz ve 255 değerini elde edene kadar döngü tekrarlanıyor. İkinci for yapısında, değişkenin değerini sıfır değere ulaşana kadar beş için azaltılır.

```

1  int ledPin = 9;           // Led diyot 9 numaralı pine
                               // bağlıdır
2  void setup() {           // Pinlerin yapılandırılması
3  }                           // yapılmıyor
4  void loop() {            // Döngünün başlangıcı
5  for(int fadeValue = 0; fadeValue <= 255; fadeValue +=5)
   {
   // for yapısı için, sembolik adı "fadeValue" olan ve başlangıç değeri sıfır
   // olan bir değişken bildiririz. Bu değişkenin değerini 255'e eşit bir sayı elde edene
   // kadar 5'er artırıyoruz. Her artış veya azalmadan sonra 30 milisaniye beklenir.

6  analogWrite(ledPin, fadeValue); //

```

```
7     delay(30);                // Işık yoğunluğunu
8     }                        // artırıyoruz
9     for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5)
    {
    .
    // Işığın yoğunluğunu azaltıyoruz. Sıfır değerini elde edene kadar değişkeni
    // beş için azaltıyoruz. Ondan sonra döngünün başlangıcına dönüyoruz

10    analogWrite(ledPin, fadeValue);
11    delay(30);
12    }
13    }
```

Önceki programlarda olduğu gibi bu programı da Arduino Uno R3'e aşağıdaki prosedürle giriyoruz:

- (1) Arduino Uno R3'ü USB kablosuyla bilgisayara bağlıyoruz
- (2) Ana menüde Tools seçeneğini seçip platform türünü ve seri bağlantı noktasını seçiyoruz
- (3) Verify/Compile aracını kullanarak programı makine diline çeviriyoruz
- (4) Ve son olarak Upload aracına basarak programı giriyoruz.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Programı girdikten hemen sonra elektrik devresi çalışmaya başlayacak yani led diyotun ışık şiddeti dönüşümlü olarak azalmaya ve artmaya başlayacaktır.

Yorum: _____

7. Değişiklik yapın!

- Analog değer artışını veya azalmasını beşten farklı bir değer için değiştirirsek ne olacaktır?

Yorum: _____

- delay() yönergesinde gecikme süresini arttırsak ne olacaktır?

Yorum: _____

4.8.1.4. Uygulamalı Alıştırma: Işık Yoğunluğunun Kontrolü İçin Fotorezistör

1. Alıştırmanın amacı

Bu alıştırmada, LED diyotun ışık yoğunluğunu fotorezistör kullanarak değiştireceğiz. Şöyle ki, LED diyotu Arduino Uno R3'ün fotorezistörle birlikte yerleştirildiği ortamın aydınlığını göstergesi olarak kullanılacaktır. Ayrıca bu alıştırmada

öğrenciler sensör kalibrasyonunun ne demek olduğunu ve nasıl yapıldığını öğrenecekler.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Arduino Uno R3 platformu
- Devre tahtası
- LED diyot
- Fotorezistör
- $R1=220\Omega$ ve $R2=10K\Omega$ dirençli rezistörler
- Bağlama kabloları (jumper kablolar)
- Arduino platformu için geliştirme ortamı kurulmuş bilgisayar

Fotorezistör, yüzeyine düşen ışığın etkisi altında elektrik direnci değişen yarı iletken bir elemandır. Bu özelliği nedeniyle elektronik sensör olarak uygulama bulur. Fotorezistörü devreye dahil etmeden önce, multimetre kullanarak ışığın yoğunluğunu değiştirerek direncini ölçebiliriz. Işık olmadığında direnci en yüksektir. Fotorezistörü ışık kaynağına ne kadar yaklaştırsak direnci da o kadar azalır.

3. Alıştırma için hazırlık

Öğrenciler şunları hatırlatılmalıdır:

- Fotorezistörlerin temel özellikleri
- Arduino Uno R3'ün pin diyagramı
- Analog pinlerle çalışma yönergeleri ve olası değer aralığı
- koşullu if yapısının anlamı
- Değer aralığını değiştirmek için map() yönergenin uygulanması.

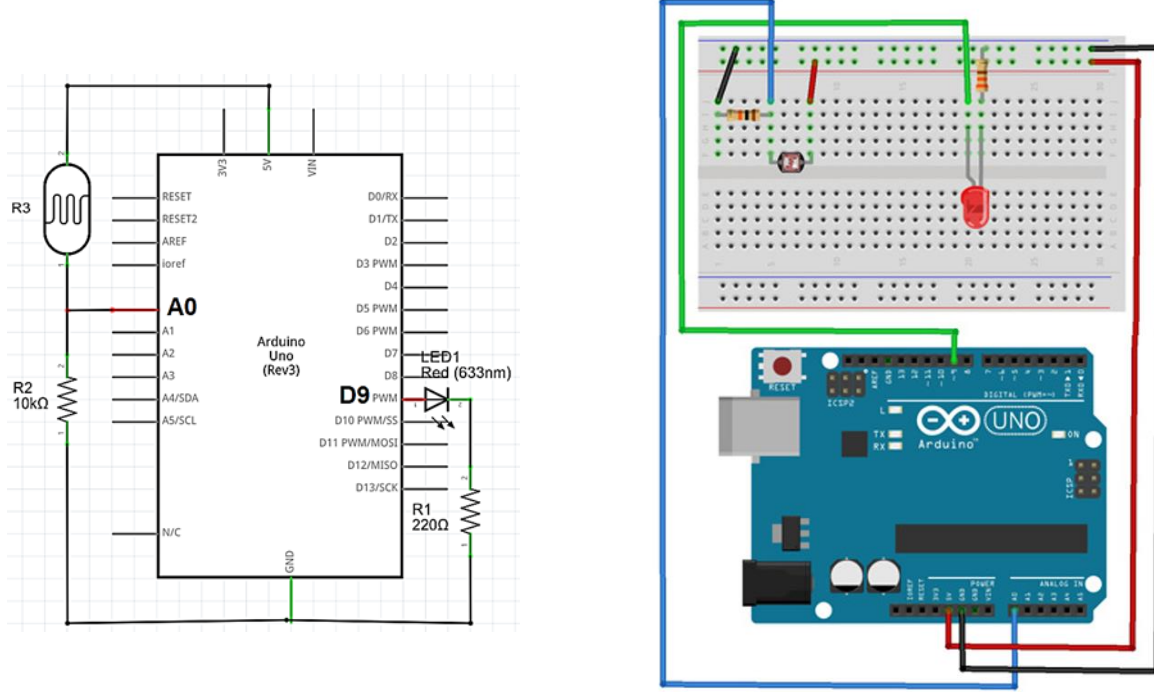
4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.31.'de bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmektedir.

- (1) Fotorezistörü devre tahtasına yerleştirerek, bağlantı noktalarının farklı iletkenlere ait temas noktalarına yerleştirmeye dikkat etmemiz gerekiyor. Kırmızı jumper kablosu ile fotorezistörün bir ucunu güç kaynağı bantına bağlıyoruz. Fotorezistörün diğer ucunu R2 rezistörü ve siyah jumper kablosu aracılığıyla topraklama bantına bağlıyoruz. Aslında fotorezistör ve R2 rezistörü bir gerilim bölücü oluştururlar. Mavi jumper kablo ile fotorezistör ile R2 rezistörü arasındaki temas noktası A0 analog girişine bağlanır.
- (2) Önceki alıştırma gibi, LED diyotun katotunu R1 rezistörü ve siyah jumper kablosu aracılığıyla topraklama bantına bağlıyoruz.

Yeşil jumper kablosu ile anotu PWM işaretli dokuzuncu pine bağlıyoruz, bu da bunun analog bir çıkış olduğu anlamına gelir.

(3) Arduino Uno R3'ün GND ve 5V pinlerini güç kaynağı ve topraklama bantlarına bağlıyoruz.



Şekil 4.31. LED diyotun ve fotorezistörün Arduino Uno R3'e bağlanması

Arduino Uno R3, gerilim bölücüden elde edilen gerilimi ölçer ve ölçülen değerleri doğru orantılı olarak 0 ila 1023 aralığında tamsayı değerlerine dönüştürür. 1023 tamsayı değeri, 5V'luk gerilime, sıfır ise 0V'luk gerilime karşılık gelir. Ancak, R2 rezistörünün gerilim düşüşü nedeniyle A0 pinindeki giriş gerilimi hiçbir zaman 5V değerine ulaşmayacak ve giriş sinyali maksimum değerine ulaşamayacaktır. Ayrıca, fotorezistör iç mekanda ışık değişikliklerinden çok daha geniş bir aralıkta ışık yoğunluğundaki değişiklikleri tespit etmek için tasarlanmıştır. Bu nedenlerden dolayı fotorezistörün kalibre edilmesi gerekir. Kalibrasyon, sensörün doğru çalışması için gereklidir. Ölçme tekniğinde, ölçüm cihazının kalibrasyonu terimi, ölçüm cihazı tarafından okunan değer gerçeğe (etalon değerine) göre sapmasının belirlenmesi anlamına gelir. Arduino Uno R3 ve fotorezistör ile ilgili olarak sapma, platformun verdiği giriş sinyalinin maksimum veya minimum değeri ile fotorezistörün algılayabileceği maksimum veya minimum değer arasındaki farktır. 0 ila 1023 aralığındaki uç değerlere hiçbir zaman ulaşamayacaktır, bu da otomatik olarak LED diyotun her zaman açık olacağı, ancak maksimum veya minimum değerlerine hiçbir

zaman ulaşamayacağı anlamına gelir. Bu sapmayı ortadan kaldırmak için elektrik devresi açıldıktan sonra, ilk birkaç saniye içinde **sensörün maksimum ve minimum değeri ayarlanmalıdır**. Bu, platform yazılımının maksimum ve minimum değer aralıklarını seçmesiyle elin sensöre doğru ve sensörden uzaklaştırarak hareket ettirilmesiyle elde edilir.

5. Arduino Uno R3 için programın yazılması ve girilmesi

Kalibrasyon sadece bir kez yapılır, tekrarlanmaz ve bu yüzden zorunlu setup yapısının bileşiminde program kodu olarak yazılır. Kalibrasyon ihtiyacı nedeniyle lightLow ve lightHigh sembolik adlarına sahip iki yeni değişken tanımlanmıştır. lightHigh değişkeni, sürekli olarak sıfırdan belirli bir fiziksel ortamda fotorezistörün algılayabileceği maksimum değere kadar artar. lightLow değişkeni sürekli olarak 1023'ten fotorezistörün minimum değerine kadar azalır. `ayarlightLevel = map(lightLevel, lightHigh, lightLow, 0, 255)` yönergesi, kalibre edilmiş minimum ve maksimum değerlere sahip giriş aralığını 0 ile 255 arasında değer çıkış aralığına dönüştürür.

Devamda yönergeleri ve yapıları hakkında yorumların bulunduğu program verilmiştir.

```
// Genellikle fotorezistörün ve LED diyotun bağlanması için pinlerin
// seçimiyle başlanır.
1  const int sensorPin = 0;           // Sensör sıfırıncı analog pine bağlıdır.
//
2  const int ledPin = 9;              // LED diyot, çıkışında genişlik modülasyonlu darbeleri
// olan dokuzuncu pine bağlıdır.
//
3  int lightLevel;                   // Değişken fotorezistöre elde edilen değerleri
// saklıyor
4  int prilagodilightLevel;         // Sembolik adı ayarlightLevel olan
// değişken aralığın girişten çıkışa
// değiştiğinde kullanılır.
5  int lightHigh = 0;                // lightHigh değişkeni kalibrasyon
// gerçekleştirildikten sonraki maksimum değeri
// temsil eder.
6  int lightLow = 1023;              // lightLow değişkeni kalibrasyon
// gerçekleştirildikten sonraki minimum değeri
// temsil eder.
7  void setup()
8  {
9  pinMode(ledPin, OUTPUT);         // LED diyotu için çıkış pinin
// yapılandırılması.
// Kalibrasyon program çalıştırdıktan sonra ilk beş saniye içinde
// gerçekleştirilir.
10 while (millis() < 5000) {        // millis() yönergesi ile süre ölçülür
//
```

```
// SensorPin (fotorezistör) pininin geriliminin değeri okunuyt ve
// kalibrasyon başlatılır.
11   lightLevel = analogRead(sensorPin);
// lightLow değişkeni verilen ortamdaki minimum ışık yoğunluğuna
// karşılık gelen bir değer alır.
12   if (lightLevel < lightLow) {
13       lightLow = lightLevel;
14   }
// lightHigh değişkeni verilen ortamdaki maksimum ışık yoğunluğuna
// karşılık gelen bir değer alır.
15   if (lightLevel > lightHigh) {
16       lightHigh = lightLevel;
17   } // if yapısının sonu.
18 } // while yapısının sonu.
19 } // setup yapısının sonu.
20 void loop()
21 { // loop yapısının başlagıcı.
// sensorPin (fotorezistör) pininin geriliminin değeri okunuyor.
22   lightLevel = analogRead(sensorPin);
// Bir aralıkta diğerine geçiş doğrusaldır.
23   prilagodilightLevel = map(lightLevel, lightHigh, lightLow, 0, 255);
24   analogWrite(ledPin, prilagodilightLevel);
// Fotorezistörün aydınlığına karşılık gelen sinyal
// LED diyotu uyarır.
25 } //loop yapısının sonu.
```

Programı yazdıktan sonra daha önceki örneklerde anlatılan prosedüre göre Arduino Uno R3'e girilir.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Devreyi güç kaynağına bağladığımızda hemen ilk beş saniye içinde elimizi sensöre doğru hareket ettirerek ve uzaklaştırarak kalibrasyon gerçekleştiriyoruz. Ardından, fotorezistöre daha fazla ışık düştüğünde, ya da fotorezistörü elle veya başka bir nesneyle kapatıldığında LED'in daha parlak yanıp yanmadığını kontrol ediyoruz.

Yorum: _____

7. Değişiklik yapın!

- Kalibrasyon hakkında söylenen her şeyin onayı olarak elde edilen sonuçları seri monitörün ekranında görüntüleyebiliriz. Bunun için setup yapısında Serial.begin(9600) yönergelerini kullanarak iletim hızını tanımlamamız ve gerçekleştirilen kalibrasyondan sonra Serial.println(lightLow) ve Serial.println(lightHigh) ve eşlemeden önce, yani giriş değer aralığının çıkış aralığına değiştirilmesinden önce Serial.println(lightLevel) yönergelerini girmemiz gerekiyor.

Yorum: _____

- Fotorezistör yerine herhangi bir sensör yerleştirebiliriz ve değerleri LED diyotun yaydığı ışığa yansıtılacaktır. Ayrıca, LED diyot yerine sesler yayacak bir piezo sensör veya hızı fotorezistörün parlaklığına bağlı olan bir motor yerleştirebiliriz. Programlama kodundaki bu değişiklikler, Arduino platformunun yönerge seti hakkında iyi bilgi sahibi olmayı gerektirir.

Yorum: _____

4.8.1.5. Uygulamalı Alıştırma: Arduino Uno R3'ü Servo Motora Bağlamak

1. Alıştırmanın amacı

Bu uygulamalı alıştırmanın amacı servo motoru 180 derece için hareket ettirmektir. Servo motoru programlamak ve kontrolü için öğrenciler, 4.3.7 "Kütüphanelerle Çalışma Yönergeleri" öğretim biriminde tanıştığımız Servo.h. kütüphanesini çağırmalıdır. Bu, şimdiye kadar gerçekleştirdiğimiz uygulamalardan, Arduino Uno R3'ün programlanması için kütüphane kullanan ilk uygulamalı alıştırma dır.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Arduino Uno R3 platformu
- Devra tahtası
- Servo motor
- Bağlama kabloları (jumper kablolar)
- Arduino platformu için geliştirme ortamı kurulmuş bilgisayar

Servo motorun çalışma prensibi şekil 4.11'de gösterilmiştir.

3. Alıştırma için hazırlık

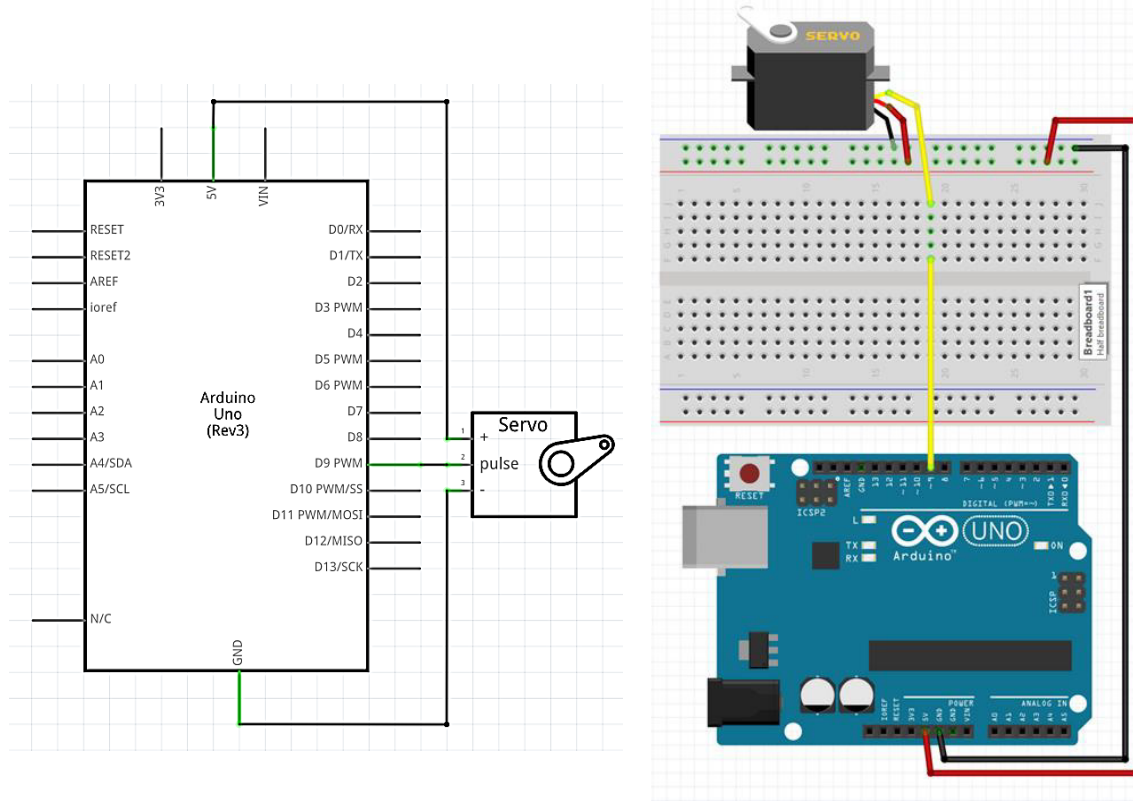
Öğrenciler şunları hatırlatılmalıdır:

- Arduino Uno R3'ün pin diyagramı
- Servo motorun çalışma prensibi
- Servo.h kütüphanesinin parçası olan yönergeler
- Döngüleri tekrarlama için for yapısının önemi

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.32.'de bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmektedir. Montaj şeması çok basittir. Servo motorun üç kablosu vardır. Siyah kablo topraklama içindir, kırmızısı güç kaynağı içindir ve sarı kablo motorun kontrolü içindir ve bunun aracılığıyla servo motor Arduino Uno R3'ün 9. pinine bağlanır. Hatırlayalım, sadece 9 ve 10 numaralı pinlerin servo motor bağlama desteği vardır. Topraklama ve güç kaynağı pinlerine kapasitesi 100 µF olan kapasitör yerleştirilerek hızlı gerilim değişiklikleri engellenebilir. Servo motor çıkış cihazıdır. Bu alıştırma

herhangi bir giriş cihazı yoktur ve servo motoru sadece yazılımla ve sadece yönergelerle kontrol edeceğiz.



Şekil 4.32. Servo Motorun Arduino Uno R3'e bağlanması

5. Arduino Uno R3 için programın yazılması ve girilmesi

Aşağıda for yapısının her tekraramasında servo motoru iki derecelik adımlarla soldan sağa 180 derece döndürecek program verilmiştir.

```

1 #include <Servo.h>           // Servo motor kontrolü için kütüphane
                                // dahil ediyoruz.
2 Servo myServo;              // MyServo sembolik adıyla bir servo nesnesi
                                // tanımlıyoruz. Servo kütüphanesi komutları
                                // bu nesneyle ilgili olacaktır.
3 int position;                // position değişkeni, servo motorun
                                // döndürülmesi gereken açının değerini
                                // içerir.
4 void setup()                 // setup fonksiyonunun başlangıcı.
5 {                             // Dokuzuncu pini servo motora bağlanacak
6   myServo.attach(9);         // pin olarak yapılandırıyoruz.
7 }                             // Setup fonksiyonunun sonu.
8 void loop()                  // Loop fonksiyonunun başlangıcı.
9 {                             // for yapısıyla dönüş açısını sıfırdan başlayıp maksimum 180 derece
                                // değerine kadar kademeli olarak iki için derece artırıyoruz.

```

```
10 for(position = 0; position < 180; position += 2) {
11     myServo.write(position);
    // for yapısının her tekrarlanmasında servo
    // motor iki derece sağa doğru hareket ediyor.
12     delay(20);           // Kısa gecikme süresi, yeni dönüşten
                           // önce konumun stabilize edilmesine
                           // yarar.
13 }                       // for yapısının sonu.
14 }                       // loop fonksiyonunun sonu.
```

Kütüphaneyi, Sketch → İmport Library ana menüsüne tıklayarak dahil ediyoruz, ardından açılır menü açılıyor. Servo.h kütüphanesi, Arduino platformunun geliştirme ortamında yer alan standart bir kütüphanedir. Programı şu prosedürle giriyoruz: Arduino platformunu bilgisayara bağlama → platform ve seri bağlantı noktası seçimi → doğrulama → girme.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Programı girdikten hemen sonra elektrik devresi çalışmaya başlayacak ve servo motor soldan sağa yarım daire kadar dönmeye başlayacaktır. 180 derece döndüğünde servo motor durur.

Yorum: _____

7. Değişiklik yapın!

- Dönüş açısını 180 dereceden sıfıra geri çevirmek için yukarıdaki programa bir for yapısı daha eklenebilir. Bu durumda açının değerinden iki çıkarılmalıdır. Bu şekilde servo motor ileri geri dönecektir.

```
14 for(position = 0; position < 180; position - = 2) {
15     myServo.write(position);
16     delay(20);
17 }
```

Yorum: _____

- Servo motorunun dönüş açısını kontrol etmek için analog pinlerden birine giriş cihazı olarak bağlanan potansiyometre de kullanılabilir. Tabi ki, bu durumda yukarıdaki programda büyük değişiklikler yapılması gerekir. Zorunlu loop yapısında, şu yönergelerin kullanılması gerekecektir: potansiyometrenin değerini okumak için analogRead(), giriş aralığını çıkış aralığına değiştirmek için map() (0-1023'ten 0-255'e) ve servo motoru hareket ettirmek için myServo.write().

Yorum: _____

4.8.1.6. Uygulamalı Alıştırma: Arduino Uno R3'ü LCD Ekranı Bağlamak

1. Alıştırmanın amacı

Bu alıştırmanın amacı "Hello world" metnini LCD ekrana ve geçen süreyi milisaniye ile ifade ederek yazdırmaktır.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Arduino Uno R3 platformu
- Devre tahtası
- LCD ekran, model GDM1602K
- 10KΩ dirençli düzeltici potansiyometre
- Bağlamı kabloları (jumper kablolar)
- Arduino platformu için geliştirme ortamı kurulmuş bilgisayar

Şekil 4.10.'da LCD ekranının pin diyagramı gösterilmiştir. 4.3.7. "Kütüphanelerle Çalışma Yönergeleri" öğretim biriminde LiquidCrystal kütüphanesinde yer alan birkaç yönerge açıklanmıştır. LCD ekran seçerken ekranın bileşimindeki denetleyicinin ekranın LiquidCrystal kütüphanesiyle uyumlu olup olmadığını kontrol etmemiz gerekiyor ve bunun için teknik-teknolojik belgelerden yararlanabiliriz. Örneğin bu kütüphaneyle uyumlu standart denetleyiciler HD44780 veya ST7066U denetleyicileridir. Düzeltici potansiyometre kontrastı ayarlamak için kullanılır.

3. Alıştırma için hazırlık

Öğrenciler şunlar hatırlatılmalıdır:

- Arduino Uno R3'ün pin diyagramı
- LCD ekran, model GDM1602K'nın pin diyagramı
- LCD ekranla çalışma yönergeleri

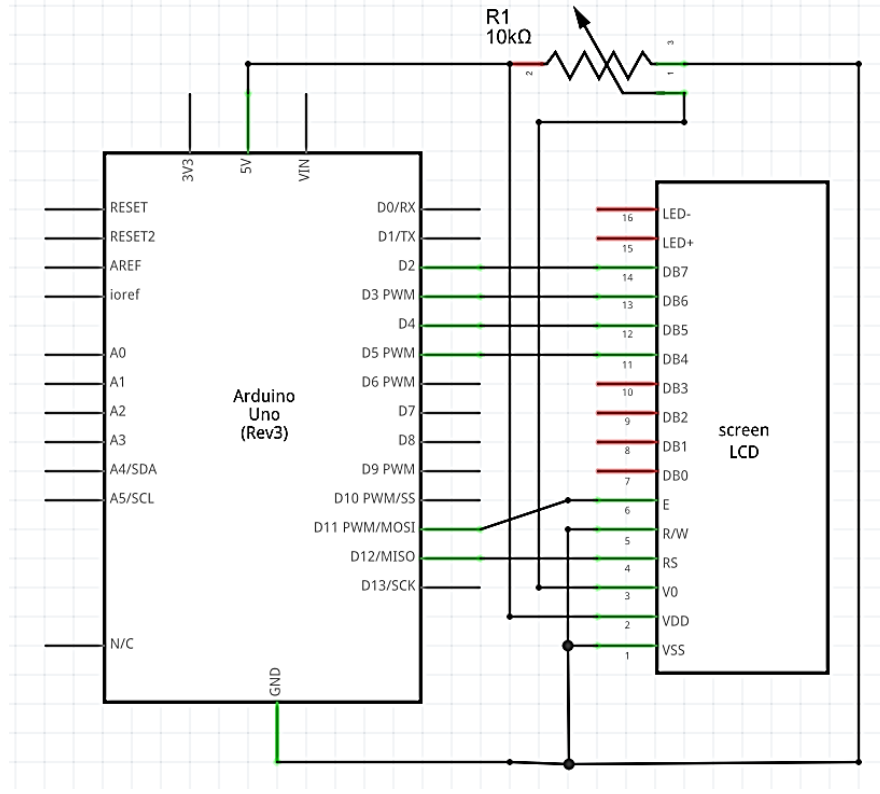
4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.33'te. LCD ekranın Arduino Uno R3'e bağlanma şekli gösterilmektedir.

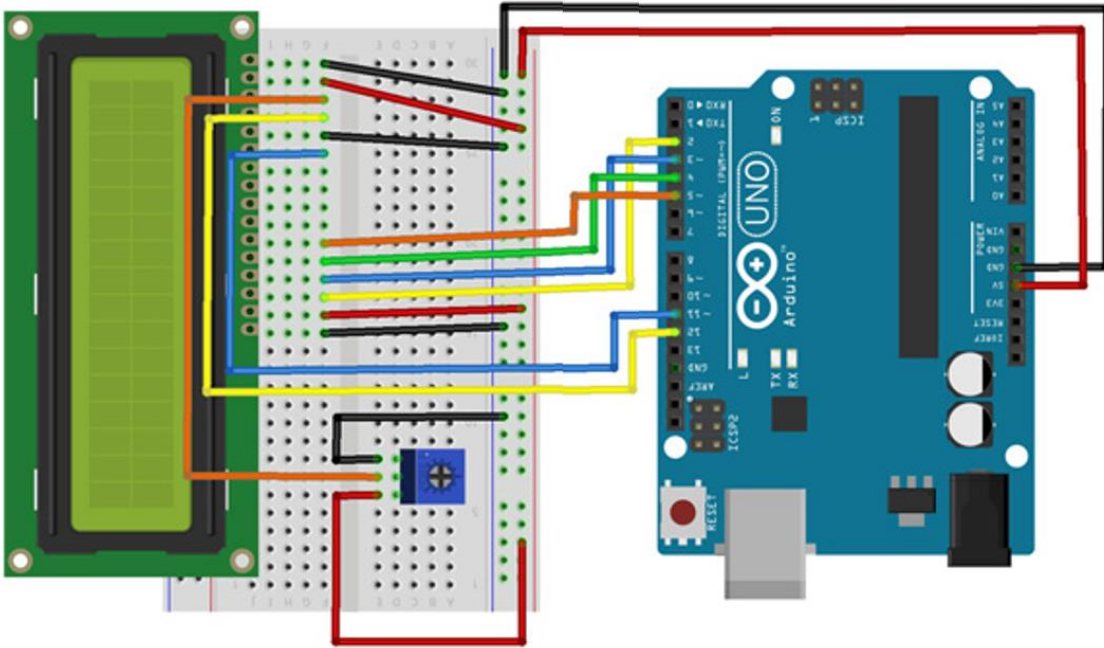
LCD ekran pinleri Arduino Uno R3 pinlerine aşağıdaki şekilde bağlanmalıdır:

LCD ekranın bağlantı noktaları	Arduino R3 pinleri
(1) RS	Dijital pin 12
(2) Enable pini	Dijital pin 11
(3) D4 pini	Dijital pin 5
(4) D5 pini	Dijital pin 4
(5) D6 pini	Dijital pin 3
(6) D7 pini	Dijital pin 2

- (7) R/W → Topraklama pini
(8) V_{ss} → Topraklama pini
(9) V_{cc} → 5V'luk güç kaynağı pini



Şekil 4.33. LCD ekranı Arduino Uno R3'e bağlamak için elektrik şeması



Şekil 4.34. LCD ekranı Arduino Uno R3'e bağlamak için montaj şeması

V_{EE} pini turuncu jumper kablosu aracılığıyla potansiyometrenin ikinci bağlantı noktasına bağlanır. Potansiyometrenin birinci ve üçüncü bağlantı noktaları güç kaynağı ve topraklama bantlarına bağlanır.

5. Arduino Uno R3 için programın yazılması ve girilmesi

Devamda geliştirme ortamında yazılımın oluşturulması verilmiştir.

```
1  #include <LiquidCrystal.h>    // Standart kütüphane olan LCD ekranla
                                   // çalışma kütüphanesi dahil edilir.
                                   //
2  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
   // Ekranla bağlanacak Arduino Uno R3 pinlerini bildiriyoruz.
3  void setup() {                //
4    lcd.begin(16, 2);           // Satır ve sütun sayısının
                                   // ayarlanmasıyla LCD ile
                                   // iletişim başlıyor.
5    lcd.print("hello world!");  // Metin ekrana yazdırılır.
6  }
7  void loop() {
8    lcd.setCursor(0, 1);        // Toplam 32 alandan bir sütun ve bir satır
                                   // seçilerek aktif alan seçilir. satırlar ve
                                   // sütunlar sıfırdan başlayarak sayıldığını
                                   // vurgulayalım
9    lcd.print(millis()/1000);   // Arduino Uno R3'ün son
                                   // sıfırlanmasından hesaplayarak
                                   // saniye olarak ifade edilen süre ekrana yazdırılır
10 }
```

Programı yazdıktan sonra şu prosedürle Arduino Uno platformunun belleğine giriyoruz: Arduino Uno R3'ü bilgisayara bağlamak → platform ve seri bağlantı noktasını seçimi → doğrulama → girilme.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Programı girdikten hemen sonra elektrik devresi çalışmaya başlayacak ve ekranda "hello world" yazısı ve ölçülen süre görüntülenecek.

Yorum: _____

7. Değişiklik yapın!

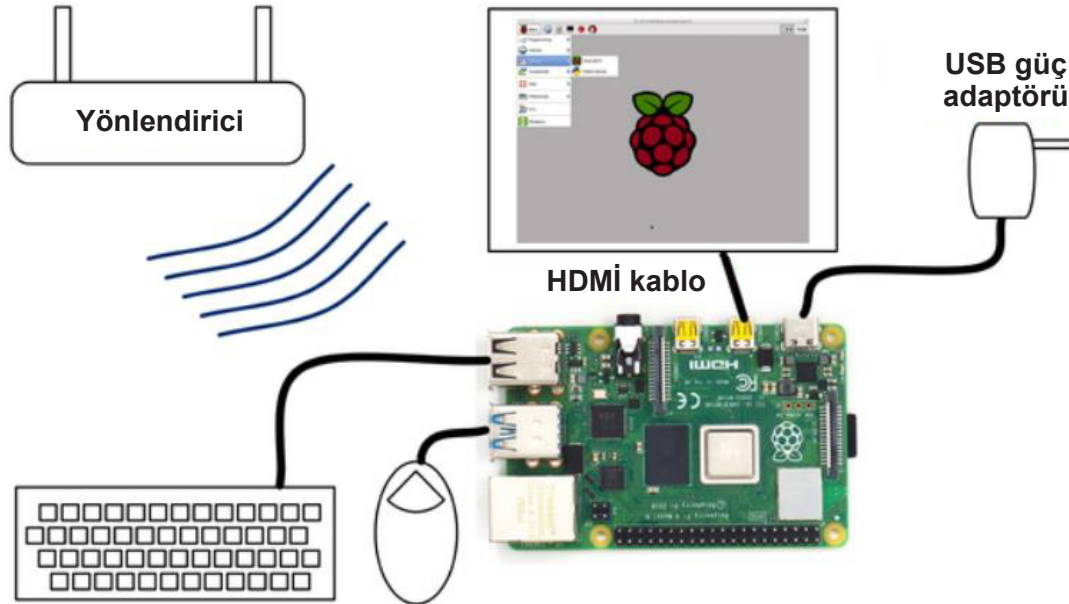
Geliştirme ortamında LCD ekranı için çok sayıda hazır program vardır. Bunlar, File → Example → LiquidCrystal seçeneğine basılarak açılabilir ve analiz edilebilirler. Bunlardan birkaçını listeleyeceğiz: metnin otomatik hareketi, metnin yanıp sönmesi, metin biçimleri yazdırma, metin yönü seçimi vb.

Yorum: _____

4.8.2. Raspberry Pi 3B+'ı Python Programlama Dilinde Programlanmasına İlişkin Uygulamalı Alıştırmalar

Uygulamalı alıştırmaların başarılı bir şekilde gerçekleştirilmesi için aşağıdaki ön bilgiler gereklidir: Raspbian işletim sisteminin kurulumu, Python programlama dilinde programlama, GPIOZERO kütüphanesinin kullanılması ve temel giriş-çıkış cihazlarının bağlanması ve özellikleri.

Alıştırmaları gerçekleştirirken Raspberry Pi 3B+, program kodu metnini girmek için klavyeye ve programın yürütülmesinden sonra elde edilen metin sonuçlarını görüntülemek için monitöre bağlanmalıdır. Şekil 4.35.'te bir Raspberry Pi sisteminin tamamı gösterilmektedir. Klavyenin bağlanması için USB bağlantısı, monitöre bağlanmak için ise HDMI bağlantısının kullanıldığını hatırlayalım. Ayrıca, Raspberry Pi modeli seçiminde herhangi bir sınırlama yoktur çünkü verilen kodlar tüm modellerle uyumludur.



Şekil 4.35. Raspberry Pi sistemi

Her uygulamalı alıştırmada, devre tahtası kullanarak Raspberry Pi 3B+'ı giriş-çıkış cihazlarına bağlama şeması verilmiştir. Ardından uygun açıklamalarla program kodu gelir. Program kodlarıyla, bazı parametrelerin değerleri değiştirilerek denebilir: motorun dönüş hızı, diyot yanıp sönme frekansı, RGBLED diyot rengi, görüntülenen metin vb.

4.8.2.1. Raspberry Pi ile Çalışırken Koruma ve Güvenlik Önlemleri

Uygulamalı alıştırımların başarılı ve güvenli bir şekilde gerçekleştirilmesi için 2.8.1. "İşyerinde Koruma ve Güvenlik Önlemleri" öğretim biriminde belirtilen genel kurallara uyulmalıdır. Ancak aynı zamanda, genellikle teknik-teknolojik belgelerde yer alan elektronik ekipmanın spesifik özellikleri de bilinmelidir. Raspberry Pi'nin en yaygın hasar nedenleri uygun olmayan güç kaynağı ve bağlı olan giriş-çıkış bileşenlerinin özelliklerinin bilinmemesidir. Raspberry Pi ile çalışırken en önemli koruma önlemleri şunlardır:

- Güç kaynağı seçimi çok önemlidir. Güç kaynağı için maksimum 5V gerilime sahip USB adaptörü kullanılır. Raspberry Pi 4 model B'nin USB-C bağlantılı adaptör kullandığını, diğer modellerin ise mikro USB kullandığını vurgulayalım.
- Güç kaynağının sağlamak gerektiği akım, Raspberry Pi modeline ve bağlı giriş-çıkış bileşenlerinin güç tüketimine bağlıdır. İzin verilen maksimum akım 1,5A ila 2A arasındadır. Raspberry Pi 4 Model B'nin daha yüksek güç tüketimi nedeniyle 3A akıma ihtiyacı vardır.
- GPIO yuvası (soketi) 4 güç pini ve sekiz topraklama pini içerir. Güç pinlerinden ikisi 5V sabit gerilime, diğer ikisi ise 3,3V gerilime sahiptir.
- GPIO giriş pinleri için izin verilen maksimum gerilim 3,3V'tur.
- Herhangi bir montaja başlamadan önce Raspberry Pi'nin güç kaynağıyla bağlantısı kesilmelidir.
- Raspberry Pi statik elektriğe duyarlı olduğundan dolayı montaja başlamadan önce bazı metal yüzeylere dokunmak gerekmektedir. Bu şekilde İnsan vücudu boşaltılır.
- Güç kaynağını açmadan önce, tüm elektronik bileşenlerin işlevsel veya montaj şemasında gösterildiği gibi bağlandığı kontrol edilmelidir.
- Pinler kesinlikle birbirine bağlanmamalıdır çünkü zarar görebilirler.
- Çalışma akımını ayarlamak için rezistörler kullanılmalıdır. Örneğin LED diyotu doğrudan 5V güç kaynağı pinine bağlarsak ısınır ve hasar meydana gelir.
- Motorları veya diğer endüktif tüketicileri bağlamak için H köprüleri kullanılmalıdır (İng. motor driver)

4.8.2.2. Uygulamalı Alıştırma: LED Diyotun Düğmeyle Açılması

1. Alıştırmanın amacı

Alıştırmanın amacı, devre tahtası kullanarak LED diyot ve düğme Raspberry Pi 3B+'a bağlamak, Python programlama dilinde program yazmak, Thonny Python entegre geliştirme ortamını araçları aracılığıyla programı Raspberry Pi'ye girmek ve elektrik devresinin çalışmasını incelemektir.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Raspbian işletim sisteminin kurulmuş olduğu Raspberry Pi 3B+ sistemi. Sistem Raspberry Pi 3B+ ile güç adaptörü, klavye, fare ve monitörden oluşmaktadır.
- Devre tahtası
- Baskılı devre kartı düğmesi
- LED diyot
- $R1=220\Omega$ dirençli rezistör
- Bağlama kabloları (jumper kablolar)

3. Alıştırma için hazırlık

Öğrenciler şunları hatırlamalıdır:

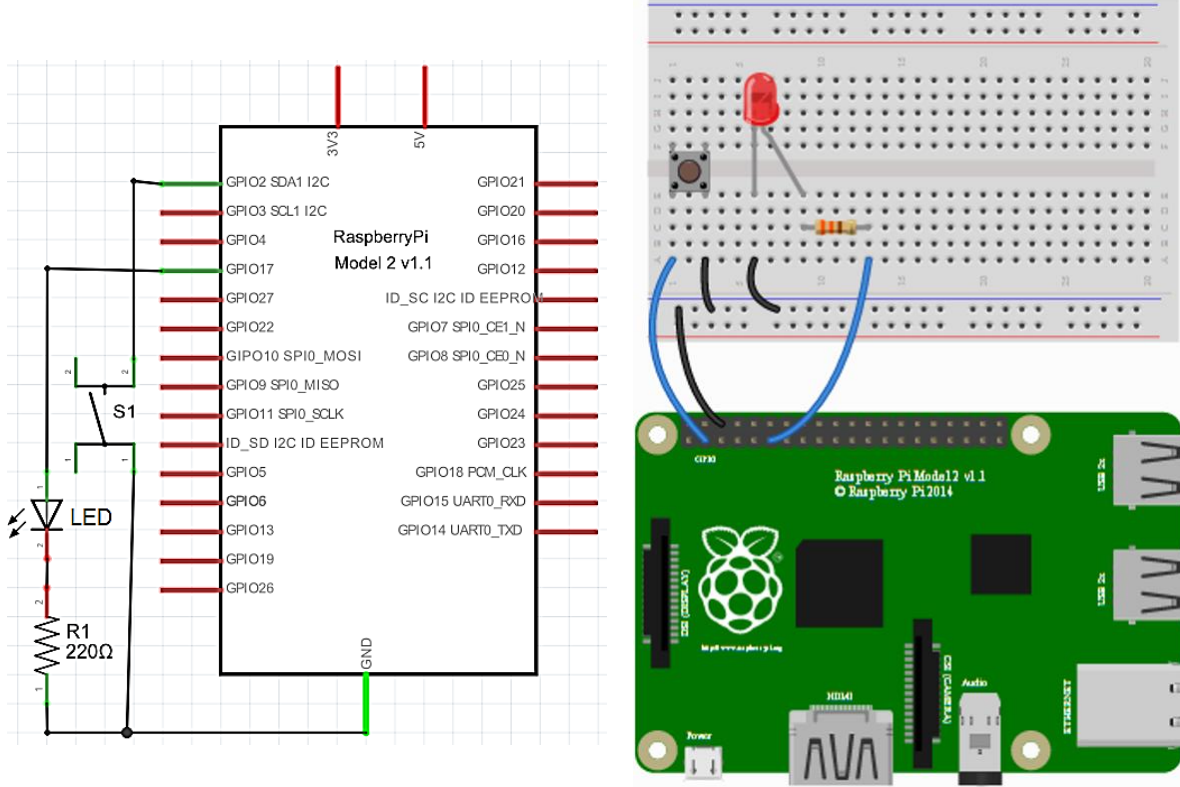
- GPIO yuvasının pin diyagramı
- LED diyotun ve düğmenin devre tahtasına bağlanma şekli. Bu, devre tahtası ile çalışma kılavuzunda da açıklanmıştır.
- Düğme ve LED diyotla çalışmak için GPIO kütüphanesinden Led ve Button sınıflarındaki yönergeler.
- 4.8.1.1. Arduino Uno R3 kullanarak "Düğme Aracılığıyla LED Diyotun Açılması" uygulamalı alıştırma ve elektrik şemanın ile program kodunun karşılaştırılması yapılsın

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.36'da. bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmektedir.

- (1) Düğme devre tahtanın ortasına yerleştirilir. Düğmenin sağ bağlantı noktası topraklama bantına, sol bağlantı noktası ise 2 numaralı GPIO pinine bağlanır.
- (2) LED diyotun katodu topraklama bantına, anodu ise 17 numaralı GPIO pinine bağlanır.
- (3) Devre tahtanın topraklama bantı Raspberry Pi 3B+'ın GND pinine bağlanır.

Şekil 4.28. ve Şekil 4.36.'daki elektrik şemaları birbirinden farklıdır. Raspberry Pi'de düğme, devre tahtanın güç kaynağı bantına bağlı değildir. Şöyle ki düğmeye basılmadığında 2 numaralı GPIO pini bağlı değildir ve devre açıktır. Düğmeye basıldığında 2 numaralı GPIO pini topraklama bantına bağlanır. 2 numaralı GPIO pinindeki düşük mantıksal seviye (İng. Low), düğmenin basıldığıнын bir göstergesidir. 2 numaralı GPIO pini yüksek mantıksal seviyede olduğunda LED yanar.



Şekil 4.36. LED diyotun ve düğmenin Raspberry 3B+'ya bağlanması

5. Raspberry Pi 3B+ için programın yazılması ve yürütülmesi

Şekil 4.14.'te programı yazmak ve yürütmek için kullanacağımız Tonny Python IDE entegre geliştirme ortamının görünümü gösterilmiştir. New aracı ile yeni proje açıyoruz, metin düzenleyicide program kodunu yazıyoruz ve yazdıktan sonra kaydediyoruz (İng.Save). Kaydettiğimizde programın adı untitledan belirttiğimiz isme değişiyor. Shell olarak adlandırdığımız bölüm, print() yönergesini kullanarak metin mesajlarını görüntülemek için kullanılır. Küçük parantez içinde görüntülenecek metin tırnak işaretleri arasında yazılır. Aşağıda bu alıştırmanın gerçekleştirilmesi için program kodu verilmiştir.

```
1 from gpiozero import LED, Button # GPIOzero kütüphanesinden LED diyot
# ve düğme ile çalışmak için iki bölümü
# dahil ediyoruz.
```

```
2 from signal import pause          # Gecikme süresini girmek için signal
                                     # kütüphanesinden bir bölüm dahil ediyoruz

3 led = LED(17)                      # Diyotun bağlanacağı pini
                                     # yapılandırıyoruz.
4 button = Button(2)                 # Diyotun bağlanacağı pini
                                     # yapılandırıyoruz.
5 button.when_pressed = led.on       # Düğmeye basıldığında LED diyotu
                                     # açıyoruz.
6 button.when_released = led.off     # Düğme basılı olmadığında
                                     # LED diyotu kapatıyoruz.
7 pause()                            # Gecikme süresine giriyoruz. Standart
                                     # gecikme süresi bir saniyedir. Bu süreyi
                                     # küçük parantezde bir değer girerek
                                     # değiştiriyoruz.
```

Yazılan programı yürütmek için Run simgesine basmamız gerekiyor. Arduino platformundan farklı olarak Raspberry Pi, kendi işletim sistemine sahip bir mikro bilgisayardır. Derleyici, programın yazılması sırasında programı yönerge yönerge olarak çevirir ve Verify ve Upload araçlarına ihtiyaç yoktur.

6. Alıştırmanın gerçekleştirilmesi sonuçları

Programın başlatılmasından hemen sonra düğmeye basmamız gerekir ve LED diyot yanacaktır. Düğmeyi bıraktığımızda LED diyotun yanması duracaktır.

Yorum: _____

7. Değişiklik yapın!

- Düğme bırakıldıktan sonra LED diyotun belirli bir süre yanmasını istiyoruz. Hangi yönerge kullanılacak ve nereye eklenecek?

Yorum _____

- Düğmeyi bıraktığınızda LED diyotun yanmasını ve düğmeye basıldığında yanmamasını istiyoruz. Program kodunda ne gibi değişiklikler yapılmalıdır?

Yorum _____

4.8.2.3. Uygulamalı Alıştırma: Trafik Işığı

1. Alıştırmanın Amacı

Trafik ışığının ışıkları kesin olarak belirlenen zamanlamaya göre açılıp kapanır ve döngü sonsuzdur yani sürekli tekrarlanır.

Bu uygulamalı alıştırmanın amacı, trafik ışığının ışıklarını Raspberry Pi 3B+ kullanarak yönetmektir. Gpiozero kütüphanesinde, özellikle trafik ışıklarıyla çalışmak için özel bir sınıf vardır (İng.TrafficLights) ve program kodunun yazılmasını önemli ölçüde basitleştirir.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Raspbian işletim sistemi kurulmuş Raspberry Pi 3B+ sistemi
- Devre tahtası
- Üç LED diyot
- R=220Ω dirençli üç rezistör
- Bağlama kabloları (jumper kablolar)

3. Alıştırma için hazırlık

Öğrenciler Raspberry Pi 3B+ donanımından GPIO yuvasının pin diyagramını ve LED diyotların devre tahtasına bağlamanın şeklini hatırlamalıdır. Aynısı, devre tahtası ile çalışma kılavuzunda açıklanmıştır.

Ayrıca, alışırmaya başlamadan önce öğrencilerin gpiozero kütüphanesindeki TrafficLights sınıfı ve time kütüphanesindeki sleep yönergesi ile tanışmaları gerekir.

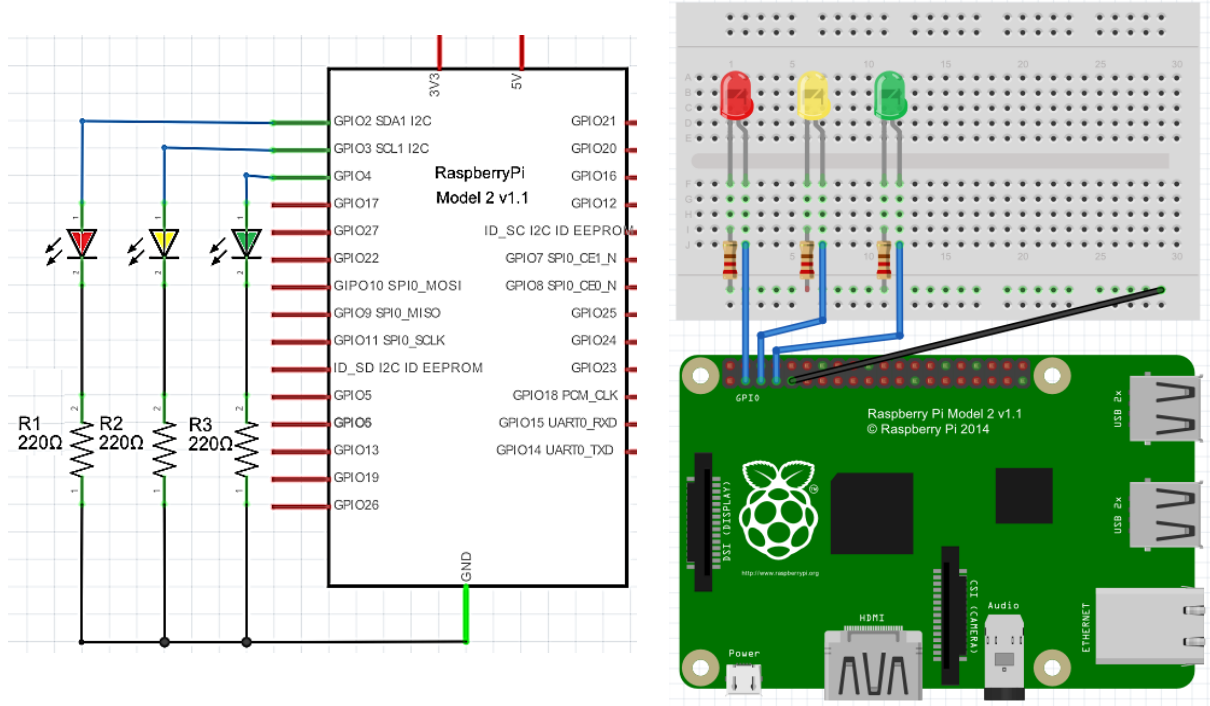
Örnek 4.54.

```
1 semafor=TrafficLights(25, 8, 7)
```

Örnek 4.54. TrafficLights() sınıfının taşıyıcısı olacak nesne bildiriminin bir örneğidir. Bu uygulamalı alıştırmanın programında, yalnızca bir trafik ışığı olduğundan bu sınıftan bir nesne bildirilir. Sistemi kavşaktan ikinci cadde boyunca trafiği yönetecek başka bir trafik ışığıyla genişletmek istiyorsak TrafficLights() sınıfının başka bir nesne daha bildirmemiz gerekecektir. Küçük parantezdeki 25, 8 ve 7 sayıları, üç ışığın yani üç diyotun bağlı olduğu GPIO pinlerinin numaralarıdır. Zaman kontrol kütüphanesinden sadece sleep() yönergesi kullanılır. Küçük parantez içinde sistemin duraklama durumunda kalacağı süreyi saniye olarak ifade etmektedir.

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

LED diyotların Raspberry Pi 3B+'a bağlanması çok basittir. LED diyotları devre tahtanın üzerine yerleştirirken bağlantı noktalarının farklı iletkenlere ait kontak noktalarına yerleştirilmesine dikkat etmeliyiz. Her LED diyotun katodu, 220Ω'luk rezistör aracılığıyla topraklama bantına bağlıdır. Kırmızı LED diyotun anodu 2 numaralı GPIO pinine, sarı LED diyotun anodu 3 numaralı GPIO pinine ve yeşil LED diyotun anodu 4 numaralı GPIO pinine bağlıdır. Raspberry Pi 3B+'ın GND pinini siyah jumper kablosu ile topraklama bantına bağlamayı unutmamalım.



Şekil 4.37. Trafik ışığının Raspberry Pi 3B+'ya bağlanması

5. Raspberry Pi 3B+ için programın yazılması ve girilmesi

Programı yazma ve girme prosedürü önceki alıştırmada olduğu gibi aynıdır. İlk önce yeni bir proje açıyoruz, metin düzenleyicide yazıyoruz, kaydedip adlandırıyoruz ve son olarak Run aracına basarak çalıştırıyoruz. Bu uygulamalı alıştırmamızın program kodu aşağıda verilmiştir.

```

1  from gpiozero import TrafficLights # GPIOZERO kütüphanesinden trafik
2  from time import sleep           # ışığıyla çalışma kısmı dahil ediliyor.
3  lights = TrafficLights(2, 3, 4)   # sleep yönergesiyle cihazı duraklama
4  lights.green.on()                # moduna getiriyoruz.
5  while True:                      # Kırmızı, sarı ve yeşil
6      sleep(10)                    # LED diyot pinlerinin
7      lights.green.off()            # yapılandırılması.
8      lights.amber.on()             # Yeşil diyot açılıyor.
9      sleep(1)                     # Sonsuz tekrarlanan döngü
10     lights.amber.off()            # başlatıyoruz.
11     lights.red.on()               # Yeşil LED diyot 10 s. yanıyor.
12     sleep(10)                    # Yeşil LED diyotu kapatıyoruz
13     lights.amber.on()             # Sarıyı açıyoruz.
14     sleep(1)                     #
15     lights.red.off()              # Sarıyı kapatıyoruz.

```

```
16 lights.amber.off()
17 lights.green.on()
   sleep(2)
```

6. Alıştırmanın gerçekleştirilmesi sonuçları

Thonny Python geliştirme ortamının araç çubuğunda Run aracına bastıktan hemen sonra yeşil LED diyot 10 saniye boyunca yanacak, ardından sarı LED diyot bir saniye boyunca yanar ve ardından kırmızı LED diyot 10 saniye boyunca yanacaktır. Sonunda kırmızı ve sarı LED diyotlar iki saniye boyunca birlikte yanar ve tekrar yeşil LED diyota döneriz. Döngü sonsuza kadar tekrarlanıyor.

Yorum _____

7. Değişiklik yapın!

Öğrenciler LED diyotların yanma sürelerini değiştirebilirler. Ayrıca, birinci trafik ışığıyla ters çalışacak başka bir trafik ışığı da ekleyebilirler ve bunun için programlama kodunu değiştirmeleri gerekir.

Yorum _____

4.8.2.4. Uygulamalı Alıştırma: Tepki Süresi

1. Alıştırmanın amacı

Bu alıştırmanın amacı iki düğmeden hangisine ilk önce basılacağını belirlemektir. LED diyot yanınca “yarışın” başladığı belirtir. Sonuç, Shell olarak adlandırılan bölümde metinsel mesaj şeklinde görüntülenir. [3]

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Raspbian işletim sisteminin kurulmuş olduğu Raspberry Pi 3B+ sistemi
- Devre tahtası
- bir LED diyot
- R=220Ω dirençli bir rezistör
- İki düğme
- Bağlama kabloları (jumper kablolar)

3. Alıştırma için hazırlık

Önceki alıştırmaya benzer şekilde, öğrenciler GPIO yuvasının pin diyagramını ve LED diyotları devre tahtasına bağlama şeklini hatırlamalıdır.

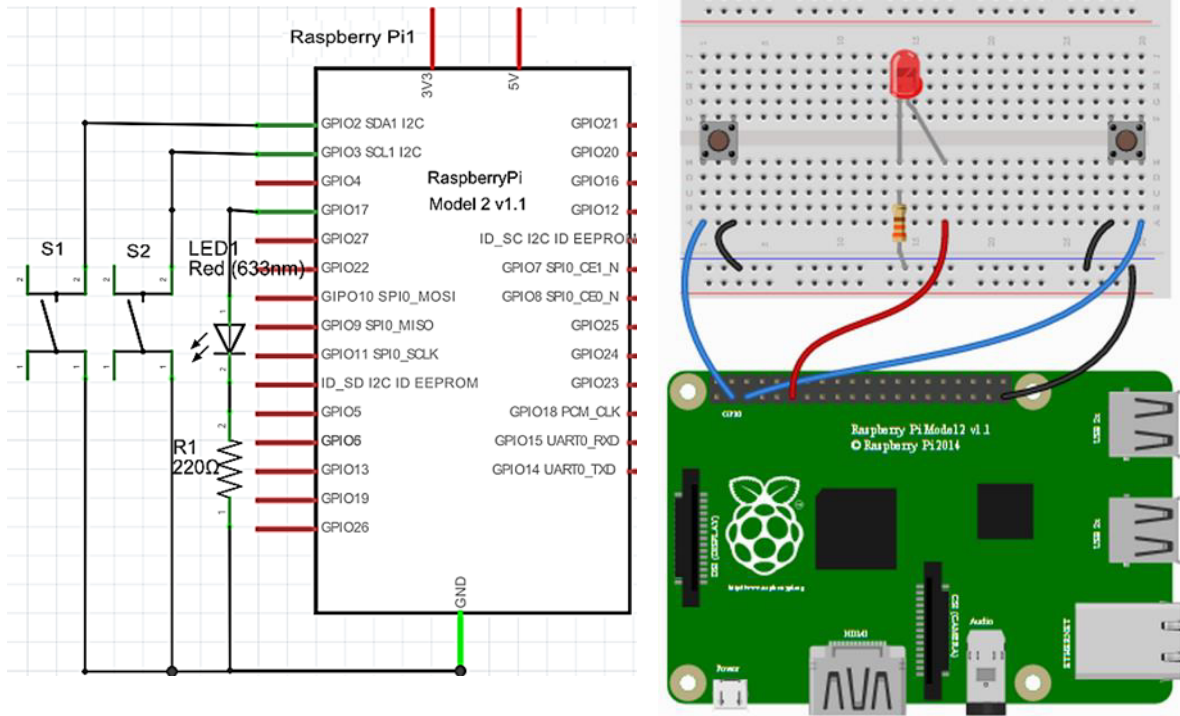
Bu alıştırmada, random kütüphanesinden uniform yönergesi ilk kez kullanılacaktır. Bu yönerge, 0 ila 5 aralığındaki bir ondalık sayının rastgele seçimi (İng.random) için kullanılır ve aynı zamanda sleep(time) yönergesinde zamanın (İng. time)

seçimi için kullanılmıştır. Zaman değişikliği, her iki oyuncu için de sürpriz elemanı olarak kullanılıyor çünkü diyotun ne zaman yanacağı tam olarak bilinmiyor.

Ayrıca, ilk kez seri monitör (İng.Shell) ve kazanan, birinci veya ikinci oyuncu hakkında bilgi içeren metinsel mesajı görüntülemek için print() yönergesi kullanılacaktır.

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.38.'de bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmektedir. Bağlanma çok basittir. Düğmeler devre tahtasının ortasına yerleştirilir ve sağ bağlantı noktaları topraklama bantına, sol bağlantı noktaları ise mavi jumper kablolarla 2 ve 3 numaralı GPIO pinlerine bağlanır. LED diyotun bağlantıları devre tahtasının iki farklı iletkenine ait kontak noktalarına yerleştirilir ve katot rezistör üzerinden topraklama bantına, anot ise kırmızı jumper kablosu ile doğrudan 17 numaralı GPIO pinine bağlanır. Topraklama bantını Raspberry Pi'de GND pinlerinden birine bağlamayı unutmayalım.



Şekil 4.38. İki düğmenin ve LED diyotun Raspberry Pi 3B+'ya bağlanması

5. Raspberry Pi 3B+ için programın yazılması ve girilmesi

Programı Thonny Python IDE geliştirme ortamında yazıyoruz. Thonny Python IDE'nin Raspbian işletim sisteminin bir parçası olduğunu hatırlatalım ve açmak için işletim sisteminin menüsünde Programming → Tools seçeneğine basıyoruz.

```
1 from gpiozero import Button,
  LED
```

```
2 from time import sleep
3 import random # random kütüphanesi, rastgele
# değerler seçme olanağı verir.
4 led = LED(17)
5 player_1 = Button(2) # Düğmeleri bağlama pinlerinin
# yapılandırması ve aynı anda
6 player_2 = Button(3) # her iki tuşa sembolik adların
# verilmesi
7 time = random.uniform(5, 10) # Sembolik adı time olan ve
# 5 ila 10 aralığında rastgele değere
# sahip bir değişken tanımlıyoruz.
8 sleep(time) # Sistem 5 ila 10 saniye boyunca
# duraklama durumundadır.
9 led.on() # Bu süre geçtikten sonra
# diyot açılıyor
10 while True: # Döngü başlatılıyor.
11     if player_1.is_pressed: # 2 numaralı pin'e bağlı
# düğme ilk basıldıysa
12         print("Player 1 wins!") # metinsel mesaj görüntülenir.
13         Break # Yukarıdaki koşul geçerliyse
# döngüden çıkıyoruz.
14     if player_2.is_pressed: # 3 numaralı pin'e bağlı
# düğme ilk basıldıysa
15         print("Player 2 wins!") # metinsel mesaj görüntülenir.
16         Break # Döngüden çıkıyoruz.
17 led.off() # LED diyot kapanıyor de
# programın başına dönüyoruz.
```

Yukarıdaki programın yazılmasıyla, program Raspberry Pi'nin belleğine girer.

6. Alıştırmanın gerçekleştirilmesinin sonuçları

Alıştırmanın doğruluğunu kontrol etmek için geliştirme ortamının araç kutusunda sadece Run'a basmalıyız ve LED diyotu yandıktan sonra her iki oyuncu da düğmeye basar. Geliştirme ortamının monitöründe metinsel mesaj görünmelidir: İki düğme basılışı arasında, 5 ila 10 saniye aralığında rastgele değere sahip bir süre tanıtılır. Sonsuz while döngüsü söz konusu olduğu için döngünün ikinci tekrarında farklı sonuç yani farklı metinsel mesaj elde edilebilir.

Yorum: _____

7. Değişiklik yapın!

Alıřtırmada her iki oyuncu kendi adını girebilir ve kısa mesajda adı görünmesi sağlanabilir. Bu amaçla program kodunda deęişiklik yapılması gerekmektedir. 6. program satırından sonra yani düğmelerin bağlanması için pinlerin yapılandırılmasından sonra ařağıdaki yönergeler eklenmelidir:

```
sol_oyuncu=input ('Adi girin')
sag_oyuncu=input ('Adi girin')
```

Print() yönergesinde da deęişiklik yapılmalı ve print("Player 1 wins!") ve print("Player 2 wins!") yerine print("sol_oyuncu wins!") ve print("sag_oyuncu wins! ") yazılabilir.

Yorum: _____

4.8.2.5. Uygulamalı Alıřtırma: LED Diyotun Fotorezistör ile Açılması ve Kapanması

1. Alıřtırmanın amacı

Bu uygulamalı alıřtırma, 4.8.1.4. Arduino uno R3 ile "İřık Yoęunluęunun Kontrolü İçin Fotorezistör" alıřtırmasına benzerdir. Ancak, Raspberry Pi'nin Arduino Uno olduęu gibi yerleşik analog-dijital dönüřtürücüsü yoktur. Bu nedenle, Arduino Uno ile uygulamalı alıřtırmada yaptığımız gibi LED diyotun ıřık yoęunluęunu deęiřtirmiyoruz. LED diyot ıřığının yoęunluęunu deęiřtirmek yerine, aydınlık veya karanlık, gündüz veya gece olup olmadıęına baęlı olarak sadece açılacak ve kapatılacaktır.

2. Bu alıřtırmanın gerçekleştirilmesi için gerekli bileşenler řunlardır:

- Raspbian iřletim sistemi kurulmuş Raspberry Pi 3B+ sistemi
- Devre tahtası
- Bir LED diyot
- $R = 220\Omega$ dirençli bir rezistör
- Fotorezistör
- $1\mu F$ kapasitanslı kapasitör
- Baęlama kabloları (jumper kablolar)

3. Alıřtırma için hazırlık

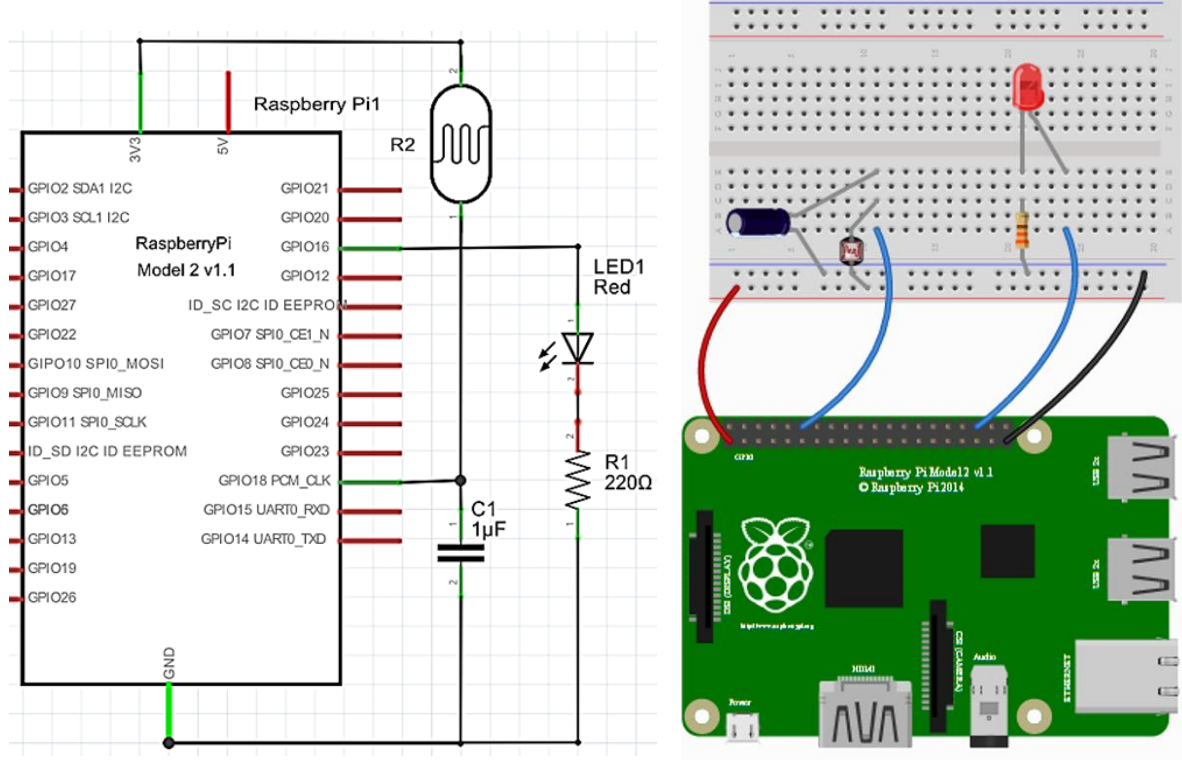
Öęrenciler řunları hatırlatmalıdır:

- Seri RC devrenin çalıřması (Elektroteknik öęretim dersi)
- GPIO yuvasının pin-diyagramı ve LED diyotu devre tahtasına baęlama yöntemi.
- Fotorezistörün ıřık yoęunluęuna baęlı olarak direncinin deęiřmesi ve gpio

kütüphanesinden LightSensor sınıfı yönergeleri (4.6.4. “Optik Sensör veya Fotorezistör” öğretim birimi)

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması Şekil 4.39.'da verilmiştir.



Şekil 4.39. LED diyotun ve fotorezistörün Raspberry Pi 3B+'ya bağlanması

4.8.1.4. “İşık Yoğunluğunun Kontrolü İçin Fotorezistör” uygulamalı alıştırma fotorezistör ve 10KΩ'lık rezistör gerilim bölücü oluştururlar. Bu uygulamalı alıştırma, gerilim bölücü yerine, 3,3V'luk güç kaynağına seri bağlı, bir rezistör ve bir kapasitörden oluşan seri RC devresi kullanacağız.

Fotorezistörün bir bağlantısı güç kaynağı bantına, diğer bağlantısı ise 18 numaralı GPIO pinine bağlanır. Aynı GPIO pinine 1 µF'lik kapasitör bağlanır ve kapasitörün diğer pini topraklama bantına bağlanır. Fotorezistörün direnci düşüğe kapasitörün gerilimi daha hızlı değişir ve tersi. Raspberry Pi 3B+, kapasitör geriliminin sıfırdan eşik gerilimi olan 1,34V değerine artması için gereken süreyi ölçer (bkz. şekil 4.16.) ve ölçülen süreye göre karanlık veya aydınlık algılar.

LED diyotun bağlanması önceki uygulamalı alıştırma ile aynıdır, tek fark şimdi 16 numaralı GPIO pininin kullanılmasıdır.

Güç kaynağı bantı Raspberry Pi'nin 3.3V pinine, topraklama bantını ise GND pinine bağlarız.

5. Raspberry Pi 3B+ için programın yazılması ve girilmesi

Programı Thonny Python IDE geliştirme ortamında yazıyoruz. gpio kütüphanesinden, fotorezistörle çalışmak için yönergeler içeren LightSensor sembolik adlı sınıfı alıyoruz. sensor.when_dark ve sensor.when_light yönergelerini kullanacağız. Raspberry Pi 3B+ karanlık algıladığında (yavaş kapasitör gerilimi değişimi) sensor.when_dark yönergesi yürütülür ve Raspberry Pi 3B+ ışık algıladığında (hızlı kapasitör gerilimi değişimi) sensor.when_light yönergesi yürütülür.

```
1 from gpiozero import LightSensor, LED # GPIOZERO kütüphanesinden
# gerekli sınıfları çağırıyoruz.
2 from signal import pause
3 sensor = LightSensor(18) # Fotorezistör ve LED diyotun
4 led = LED(16) # bağlanması için pinleri
# yapılındırıyoruz.
5 sensor.when_dark = led.on # Sensör karanlık algıladığında
# LED diyot açılır
6 sensor.when_light = led.off # Sensör aydınlık algıladığında
# LED diyot kapanır.
7 pause() # Gecikme süresi girilir.
```

Programı geliştirme ortamında yazdıktan sonra araçlar menüsünde Run tuşuna basıyoruz.

6. Alıştırmayı gerçekleştirilmesinin sonuçları

Fotorezistör ışığa açık kaldığında LED diyot yanmaz. Fotorezistörü bir nesneyle kapattığımızda LED diyot yanmaya başlar.

Yorum: _____

4.8.2.6. Ugulamalı alıştırma: En Yakın Nesneye Olan Mesafeyi Belirtme

1. Uygulamanın amacı

Bu alıştırmamızın amacı, mesafe sensörü (HC-SR04) uygulamak, onu Raspberry Pi 3B+'a bağlamak ve ölçülen mesafeyi Thonny Python IDE geliştirme ortamında Shell monitöründe metinsel mesaj olarak görüntüleyecek program yazmaktır.

2. Bu alıştırmamızın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Raspbian işletim sisteminin kurulmuş olduğu Raspberry Pi 3B+ sistemi
- Devre tahtası
- Mesafe sensörü (HC-SR04)
- Bağlama kabloları (jumper kablolar)

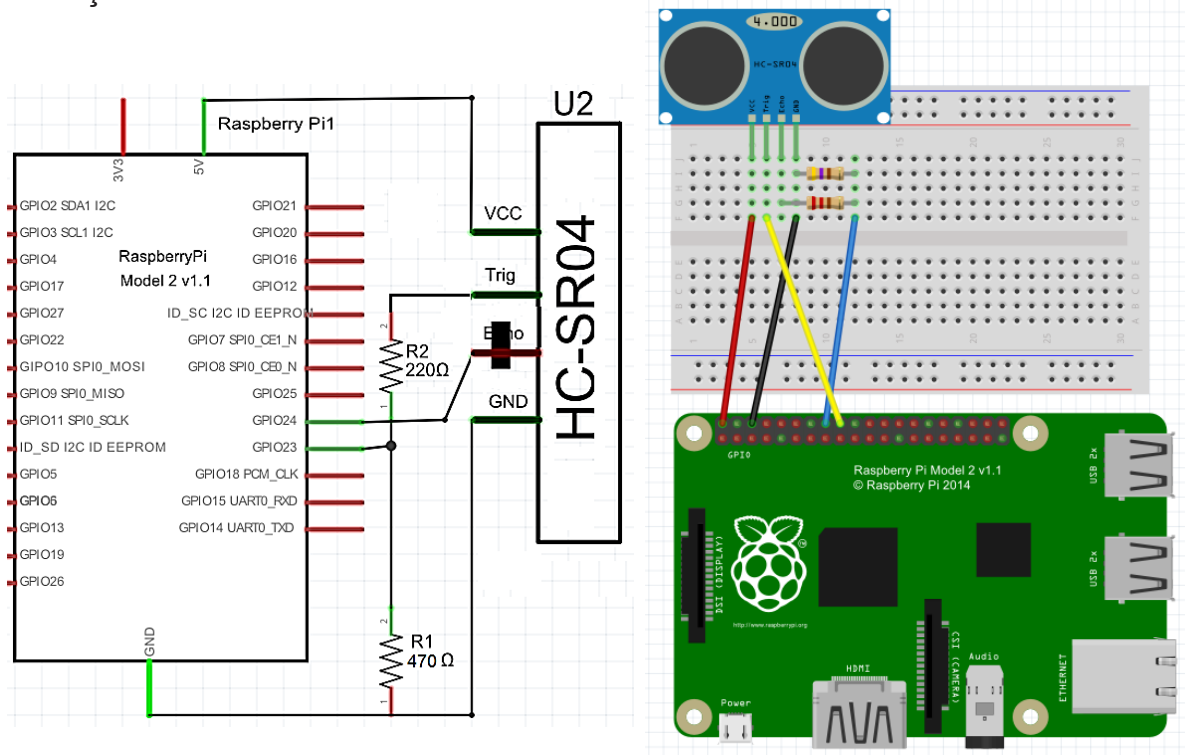
3. Alıştırma için hazırlık

Öğrenciler şunları hatırlatılmalıdır:

- 2.8.3.2. “Devre Tahtası Kullanım Kılavuzu” öğretim birimi
- GPIO yuvasının pin diyagramı (şekil 4.15.)
- 4.6.3 “Mesafe sensörü (HC-SR04)” öğretim birimi

4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması şekil 4.40.’ta verilmiştir.



Şekil 4.40. Mesafe sensörünün Raspberry Pi 3B+'ya bağlanması

Mesafe sensörünü Raspberry Pi 3B+'ya bağlamak için, biri tetikleyici, diğeri yankı için olmak üzere iki GPIO pini gerekir.

- (1) Sensörün topraklama bağlantı noktasını Raspberry Pi'nin GND pinine bağlıyoruz
- (2) Sensörün Trig bağlantı noktasını Raspberry Pi'nin 24 numaralı GPIO pinine bağlıyoruz.
- (3) 220Ω dirençli rezistörün bir ucu sensörün Echo bağlantısına bağlanır
- (4) 470Ω dirençli rezistörün bir ucu GND bağlantısına bağlanır.
- (5) İki rezistörün diğer iki ucu Raspberry Pi'nin GPIO pinine birlikte bağlanır. Bu şekilde Echo bağlantısının çıkış gerilimini azaltan gerilim bölücü oluşturulur. Bu gerilim 5V'tur, ancak Raspberry Pi 3B+ için maksimum giriş geriliminin

3,3V olduğunu biliyoruz. Aksi takdirde Raspberry Pi'de hasar meydana gelebilir.

(6) Sensörün VCC bağlantı noktası Raspberry Pi'nin 5V'luk güç kaynağı pinine bağlanır.

5. Raspberry Pi 3B+ için programın yazılması ve girilmesi

Programı, artık bilinen prosedüre göre Thonny Python IDE geliştirme ortamında yazıyoruz: yeni proje açmak → metin düzenleyicide yazmak → programı kullanıcı tarafından seçilen sembolik bir adla kaydetmek. gpio kütüphanesinden, tam olarak mesafe sensörüyle (HC-SR04) çalışmak için yönergeleri içeren DistanceSensor sembolik adını taşıyan sınıfı indiriyoruz.

```
1 from gpiozero import          # Mesafe sensörü (HC-SR04)
  DistanceSensor                # ile çalışma yönergeleri içeren
                                # sınıfı çağırıyoruz.
2 from time import sleep
3
4 sensor = DistanceSensor(23, 24) # Pinlerin yapılandırılması
5
6 while True:                    # Sonsuz döngü başlıyor
7     print(('Ne yakın nesneye olan mesafe: ', sensor.distance, 'm')
  #Ölçülen mesafenin görüntülenmesi
8     sleep(1)                   # Sistem 1 saniye duraklama durumunda kalıyor.
```

Program çok basit ve kısadır. Bunun nedeni, herşeyden önce gpio kütüphanesinin ve DistanceSensor sınıfının kullanılmasıdır. Kullanıcının Arduino Uno platformunda olduğu gibi okuma ve yazma yönergelerini kullanmasına gerek yoktur. Sadece DistanceSensor sınıfını çağırmak bunu otomatik olarak yapar. Sensörün tüm donanım özellikleri dikkate alınarak yazılım oluşturulur ve gpio kütüphanesinin yani DistanceSensor sınıfının içeriğinde yer alır.

6. Alıştırmayı gerçekleştirilmesinin sonuçları

Mesafe sensörünün önüne belirli bir nesne yerleştiriyoruz. Maksimum mesafe 1 metredir. Programı yazıp kaydettikten sonra araçlar menüsünde Run'a basıyoruz ve Shell seri monitör alanında hemen sensörden yerleştirilen nesneye olan mesafe hakkında bilgi içeren metinsel mesaj beliriyor. Nesne hareket ettirilebilir ve elde edilen verilerin güvenilirliğini kontrol edebiliriz.

Yorum: _____

4.8.2.7. Uygulamalı Alıştırma: Doğru Akım Motorunun Dönme Yönünün Değişmesi

1. Alıştırmanın amacı

Bu alıştırma, programlamayı çok basit ve anlaşılır yapan kütüphaneler ve yönerge setleri sayesinde Raspberry Pi 3B+'nın programlanmasının ne kadar kolay olduğuna dair başka bir kanıttır. Bu alıştırmanın amacı DC motorun dönüş yönünü değiştirmektir. Doğrudan akım motorunun Raspberry Pi'ye doğrudan bağlanmadığını ancak arayüz bileşeni olarak H-köprüsü kullanıldığını hatırlayalım.

2. Bu alıştırmanın gerçekleştirilmesi için gerekli bileşenler şunlardır:

- Raspbian işletim sisteminin kurulmuş olduğu Raspberry Pi 3B+ sistemi
- Devre tahtası
- 3V güç kaynaklı doğru akım motoru
- SN754410 entegre devresi
- Bağlantı kabloları (jumper kablolar)

3. Alıştırma için hazırlık

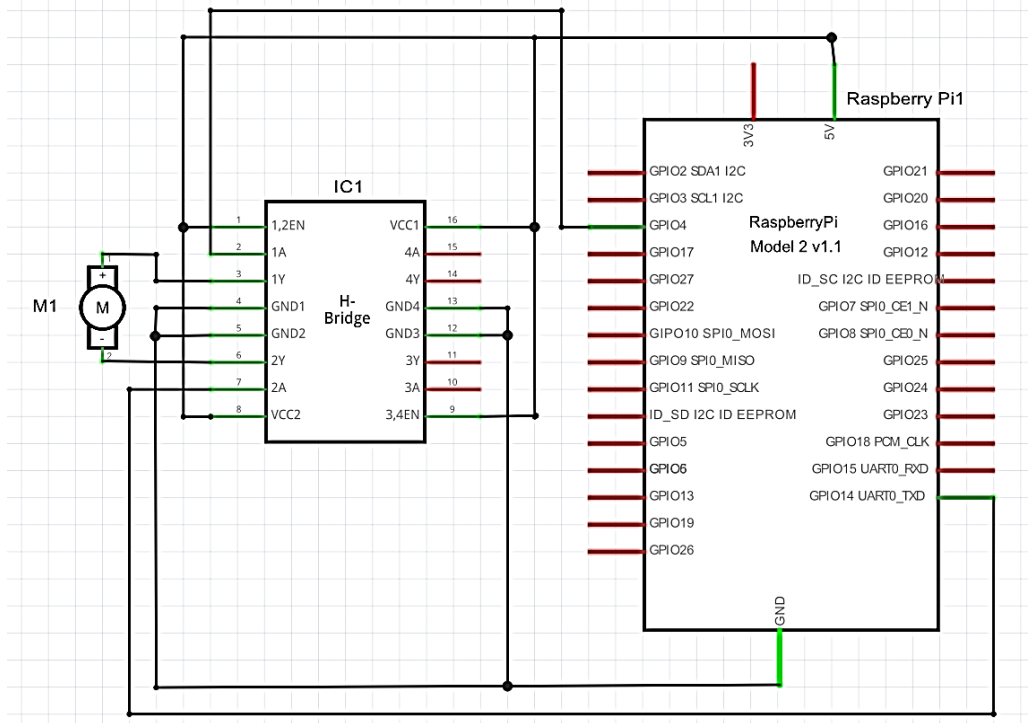
Öğrenciler şunları hatırlatılmalıdır:

- Raspberry Pi 3B+'de GPIO yuvasının pin diyagramı (Şekil 4.15.)
- 4.7.4. "Doğru Akım Motoru" öğretim birimi
- SN754410 entegre devresinin pin diyagramı (şekil 4.25.)

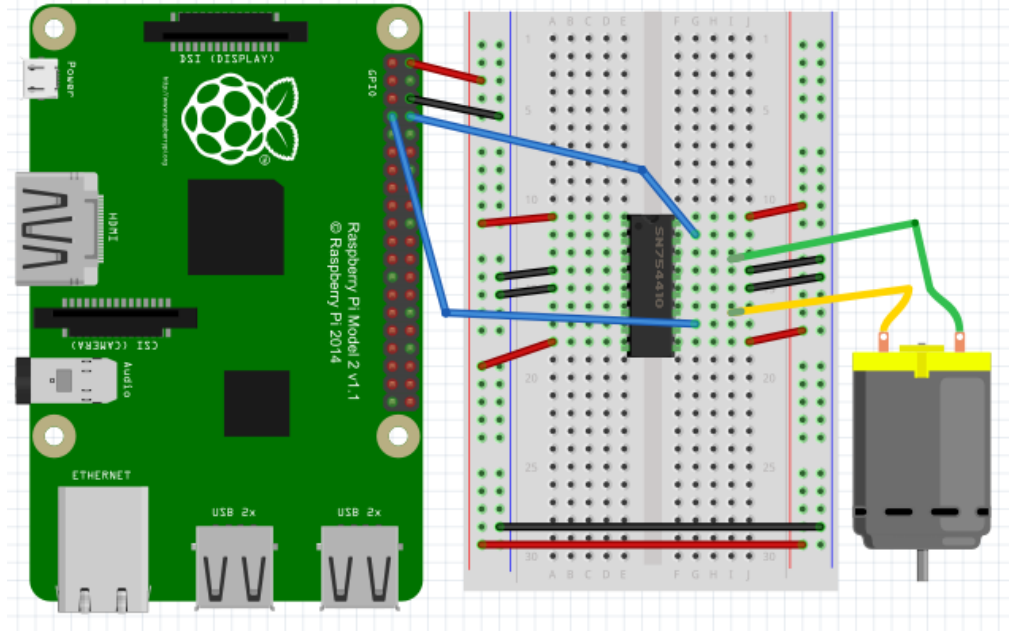
4. Elektrik şemanın açıklaması ve bağlanma yöntemi

Şekil 4.41. ve 4.42.'de bu alıştırmanın gerçekleştirilmesi için işlevsel ve montaj şeması gösterilmiştir.

- (1) SN754410 entegre devresini devre tahtasının ortasına, pinlerinin bükülmesine dikkat ederek yerleştiriyoruz. Kırmızı jumper kablolarla güç kaynağı ve etkinleştirme pinlerini (İng. Vcc, Enable) güç kaynağı bantına bağlıyoruz, siyah jumper kablolarla ise dört GND pinini topraklama bantına bağlıyoruz.
- (2) Devre tahtası üzerinden, mavi jumper kablolarla 4 ve 14 numaralı GPIO pinlerini SN754410 entegre devresinin giriş pinlerine (İng. In1, In2) bağlıyoruz.
- (3) Devre tahtası üzerinden yeşil ve sarı jumper kablolarla doğru akım motorunun bağlantı noktalarını SN754410 entegre devresinin çıkış pinlerine (Out1, Out2) bağlıyoruz.
- (4) Son olarak üst ve alt güç kaynağı bantını Raspberry Pi'nin 5V pinine, güç kaynağı bantı ise GND pinine bağlıyoruz.



Şekil 4.41. Doğru akım motorunun Raspberry Pi 3B+'ya bağlanmasının işlevsel şeması



Şekil 4.42. Doğru akım motorunun Raspberry Pi 3B+'ya bağlanmasının montaj şeması

5. Raspberry Pi 3B+ için programın yazılması ve girilmesi

Programı Raspbian işletim sisteminin içeriğindeki Thonny Python IDE geliştirme ortamında yazıyoruz. gpio kütüphanesinden Motor yönerge sınıfını çağırıyoruz. Daha önce tanıştığımız motor.forward() ve motor.backward() yönergelerini kullanıyoruz.

```
1 from gpiozero import Motor # gpio kütüphanesinden motorla
                                # çalışma kısmını giriyoruz.
2 from time import sleep
3 motor = Motor(forward=4, backward=14)
4 #4 ve 14 GPIO pinlerinin numaralarıdır.
5 while True: # Sonsuz döngü başlıyor. Motor
6     motor.forward() # saat yönünde dönüyor. Küçük
                    # parantez boş olduğundan bu, motorun
                    # maksimum hızda döneceği anlamına
                    # gelir. Eğer küçük parantez içinde
                    # 0,5 sayısı olsaydı bu, motorun
                    # maksimum hızının yarısı kadar
                    # döneceği anlamına gelirdi.
7     sleep(5) # Motor 5 saniye bir yönde dönüyor.
8     motor.backward() # Dönme yönü değişiyor
9     sleep(5) # Motor beş saniye saat yönünün
                # ters yönünde dönüyor.
```

6. Alıştırmanın gerçekleşmesinin sonuçları

Bu uygulamalı alıştırmanın gerçekleşmesinin sonuçlarını, Thonny Python IDE geliştirme ortamının araçlar menüsünde Çalıştır tuşuna basarsak göreceğiz. Beklenen sonuç, motorun her 5 saniyede yön değiştirerek dönmeye başlamasıdır.

Yorum: _____

Sonuçlar

C/C++ programlama dilinin temel yapı elemanları şunlardır: değişkenler, yönergeler ve yapılar.

Programlamada sözdizimi, bilgisayarın anlayabileceği ifade elde etmek amacıyla etiketleri, sembolik adları, operatörleri, noktalama işaretlerini, yorumları sıralamak için bir dizi kurallar anlamına gelir.

Değişkenler bilgisayarda işlenen verileri saklar. Değişkenlerin adı ve veri türünün yanı sıra kendi değerleri de vardır. Arduino platformu için değişkenleri bildirdiğimizde (deklare ettiğimizde) ilk önce değişkenin türü, ardından sembolik adı, atama operatörü ve değişkenin değeri yazılır.

Temel veri türleri şunlardır: tamsayılar, ondalık sayılar, karakterler, dizeler ve mantıksal veriler.

Operatörler, programcının kesin olarak tanımlanmış kurallara göre ifadeler ve yönergeler oluşturduğu işaretlerdir. Birkaç tür operatör vardır: matematiksel, mantıksal, karşılaştırma, atama operatörleri.

Giriş-çıkış pin yönergeleri, pinleri (giriş veya çıkış) yapılandırmak ve değerlerini yazmak veya okumak için kullanılır. Bu gruba şu yönergeler aittir: `digitalRead(pin)`, `digitalWrite(pin,deger)`, `pinMode(pin,mode)`, `analogRead(pin)` ve `analogWrite(pin,deger)`.

Arduino Uno R3'te yerleşik olan analog-dijital dönüştürücü, analog değerleri 0'dan 1023'e kadar tamsayıya dönüştürür. Tamsayı daha sonra 10 bitlik ikili kod olarak temsil edilir.

`delay(ms)` yönergesi, milisaniye olarak ifade edilen gecikme süresini girmek için kullanılır.

`map(deger,fromLow,fromHigh,toLow,toHigh)` yönergesi, değişkenin değer aralığını değiştirir. `fromLow` ve `fromHigh` eski aralığın minimum ve maksimum değerleri, `toLow` ve `toHigh` ise yeni aralığın minimum ve maksimum değerleridir.

`bitClear(x,n)` yönergesi, `x` değişkenindeki `n` sıra numaralı olan biti sıfırlar (sıfıra ayarlar). `bitSet(x,n)` yönergesi, `x` değişkenindeki `n` sıra numaralı olan biti 1 bitine ayarlar (yüksek seviyeye ayarlar).

`Serial.begin(hiz)` yönergesi ile bant genişliği, yani ölçü birimi bir saniyedeki bit sayısı olan seri veri aktarım hızı ayarlanır. `Serial.print(x)` yönergesi ile seri monitörün ekranında değişkenin değeri yazdırılır.

Olanaklardan seçim yapıları `if`, `if...else` ve `else` ifadelerinden oluşur. `if` ifadesi koşulu kontrol eder ve yerine getirilirse (doğruysa) yönergeleri yürütür.

Kütüphaneler, çevresel cihazlar ile Arduino platformu arasındaki iletişimi sağlayan alt programlardır. Arduino geliştirme ortamı bir kaç standart kütüphane içerir ve bunları menüde `Sketch > Import Library` seçeneğine tıklayarak (click) görebiliriz.

Sensörün verebildiği değer aralığı, gerçek ortamda ölçülebilen değer aralığından çok daha büyüktür. Bu nedenle sensörü çalıştırdıktan hemen sonra, ilk bir kaç saniye içinde sensörün maksimum ve minimum değerleri ayarlanmalıdır. Bu prosedüre kalibrasyon denir.

Python programlama dilinde değişkenlerin bildirimini olmadığı için kodlar daha kısadır. if...else veya while gibi bir yapı çerçevesinde ifade blokları yazdığımızda büyük parantezler kullanılmaz, onun yerine ifadeler içeriye doğru dört boşluk girintili olarak yazılır.

Düğme ile çalışma yönergeleri şunlardır: Button(), wait_for_press (), wait_for_release(), held_time, hold_time, is_held ve is_pressed

LED diyotu ile çalışma yönergeleri şunlardır: LED(), blink(on_time=1, off_time=1) , off(), on () ve toggle().

Tekrarlama soruları

1. C++ programlama dilinin temel yapı elemanları hangileridir?

2. Programlamada sözdizimi terimiyle neyi tanımlıyoruz?

3. C++ programlama dilinde kullanılan temel veri türlerini listeleretiniz?

4. bool isCodingFun = true değişkeninin adı, değeri ve türü nedir?

5. Metinsel verileri ve dize için etiketler nelerdir?

6. Mantıksal veriler kaç değere sahip olabilir?

7. C++ programlama dilinde operatörler neyi temsil eder?

8. Değişkene değer atama operatörü hangisidir?

9. x=13%5; yönergesi yürütüldükten sonra nasıl sonuç elde edilecektir?

10. x=x-1 aritmetik yönergesinin kısaltılmış ifadesini yazınız.

11. if (x!=y) {...} yapısı altında yönerge bloğunun çalıştırılmasının koşulu nedir?

12. digitalWrite () yönergesinin parametreleri nedir?

13. ArduinoUno geliştirme platformunun bileşimindeki analog-dijital dönüştürücü ne için kullanılır?

Aşağıdaki kodu yorumlayın:

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

14. tone (pin, frequency) yönergesi ne için kullanılır?

15. Arduino Uno geliştirme platformunda programın çalıştırılmaya başlanmasından itibaren geçen süre hangi yönerge ile ölçülür?

16. constrain(x, a, b) yönergesini açıklayınız?

17. Verilen değişkende belirli bir biti sıfırlamak için hangi yönerge kullanılır?

18. Arduino Uno mikrobilgisayarı için programlarda bulunması gereken yapılar hangileridir ve onların işlevleri nedir?

19. if () {...} yapısının küçük ve orta parantezlerinde ne yazılıyor?

20. Olanaklardan seçim yapısını adlandırın!

21. while...do ve do....while yapıları arasındaki fark nedir?

Aşağıdaki kodda döngü kaç kez tekrarlanacaktır:

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

22. for (...) {...} yapısının küçük parantezlerinde hangi üç ifade yazılır?

23. yönergede her parametre neyi tanımlar?

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)?

24. Servo.attach (pin, min, max); yönergesini yorumlayın?

Aşağıdaki kodu yorumlayın:

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

25. Python programlama dilinin avantajları ve dezavantajları nedir?

26. Derleyici ve yorumlayıcı sistem programları neye göre farklıdır?

27. Python programlama dilinden hangi yönerge yeni işlevler oluşturmak için kullanılır?

28. Python programlama dilinde giriş-çıkış cihazlarla çalışmak için hangi kütüphaneyi kullanıyoruz?

29. 1, 2, 3 ve 5 fiziksel sayılı GPIO pinleri nasıl işaretlenmiştir? Onların işlevlerini açıklayınız!

30. Aşağıdaki yönergelerin neye göre farklı olduklarını açıklayınız: import gpiozero, from gpiozero import Button, from gpiozero import ButtonBoard!

31. Raspberry Pi mikrobilgisayar için bir kaç analog giriş cihazı listeleyiniz!

32. Raspberry Pi mikrobilgisayarının bir analog giriş cihazının aktif olup olmadığına nasıl karar verdiğini açıklayın!

33. C/C++ ve Python programlama dilleri arasında karşılaştırma yapınız? Bunlardan hangisi daha büyük sayıda kütüphaneye sahiptir ve onların kullanımından elde edilen kazanç nedir?

34. Buton() yönergesi için hangi parametreler karakteristiktir?

35. wait_to_release yönergesine nerede rastlanır ve ne için kullanılır?

36. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

```
from gpiozero import Button  
from signal import pause  
def say_hello():  
  print("Hello!")
```

```
button = Button(2)
button.when_pressed = say_hello
pause()
```

37. Raspberry Pi 3B+'a fotorezistör bağlarken kapasitör neden gerekir?

38. Mesafe sensöründeki iki veri bağlantı noktası ne için kullanılır?

39. Cihazı duraklama moduna ayarlamak için hangi kütüphane ve sınıf kullanılır?

40. Raspberry Pi'ye USB bağlantısı üzerinden bağlanan bir kaç çıkış cihazı sayınız?

41. distance yönergesi hangi sınıfta kullanılır ve nasıl sonuç verir?

42. LED diyotun durumunu değiştirmek için hangi yönerge kullanılır?

43. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

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

44. Genişlik modülasyonlu darbelerin genişliği PWMLED diyotunun ışık yoğunluğunu nasıl etkiler?

45. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

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

```
sleep(1)
```

46. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

```
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()
```

47. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

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

48. Üç ana renk hangileridir ve beyaz ve siyah rengi elde etmek için hangi değerlere sahip olmaları gerekir?

49. Hangi yönerge ile DC motorun dönüş hızı değiştirilebilir?

50. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

```
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()
```

51. DC, Raspberry Pi'ye bağlanırken dönüş hızını hangi parametre belirler?

52. Servo motorla çalışırken value() yönergesi ne için kullanılır?

53. Aşağıdaki kodun yönergelerini ve işlevini yorumlayın!

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

Ek: Raspberry Simülatörleri

2.8.3.3. “Arduino Uno R3 için bilgisayar simülasyonu” öğretim biriminde Tinkercad platformunun sunduğu olağanüstü olanaklarla tanıştık. Maalesef bu platform Raspberry Pi için simülasyon desteklemiyor. Raspberry Pi için Python programlama dilinde ücretsiz simülatör şu anda yoktur. Ancak son modüler birime ek olarak, şu anda mevcut olan dört Raspberry simülatörünü kısaca tanıtacağız.

Microsoft şirketinin çerçevesinde Azure IoT platformu oldukça popüler bir simülatördür. Azure, dijital dönüşümler için yazılım ve donanım desteği sunan bulut teknolojisidir. İlk başlıkta Nesnelerin İnterneti'nin sensörlerin ağ oluşturmasını sağlayan bir platform olduğunu belirtmiştik. Kullanıcının iletişim protokollerinin seçimi ve yapılandırılması konusunda düşünmesine gerek yok çünkü platform onu kendi yapar. Açık tipte yani kullanıcıları program kodlarını paylaşabilecek şekilde olması düşünülmüştür. Bu simülatörün yapım aşamasında olduğunu ve deneme sürümünün kullanıldığını belirtelim. Maalesef Microsoft bir yıl boyunca eğitim amaçlı ücretsiz kullanıma izin vermesine rağmen ücretsiz değildir. Kullanıcının kendi Azure profiline sahip olması gerekir.

Wylidrin STUDIO, Raspberry Pi için ücretsiz bir simülatördür. Uygulama, <https://wylidrin.studio/> bağlantısıyla resmi web sitesinden indirilebilir, ancak kurulum gerektirmeyen, sadece Google kullanıcı profiliyle oturum açmayı gerektiren web sürümü de vardır. Azure'e benzer şekilde Wylidrin platformu, sensörlerin ağ oluşturmasını sunuyor. Proje oluştururken Fritzing uygulamasında elektrik şemasını derlemek gerekir ve ardından kaydedilen belgeler Wylidrin platformu tarafından indirilir. Çok sayıda avantajına rağmen, bu simülatör bu ders kitabının öğretim amaçlarına uygun değildir çünkü Raspberry Pi'yi programlamak için Python değil JavaScript programlama dili kullanılır.

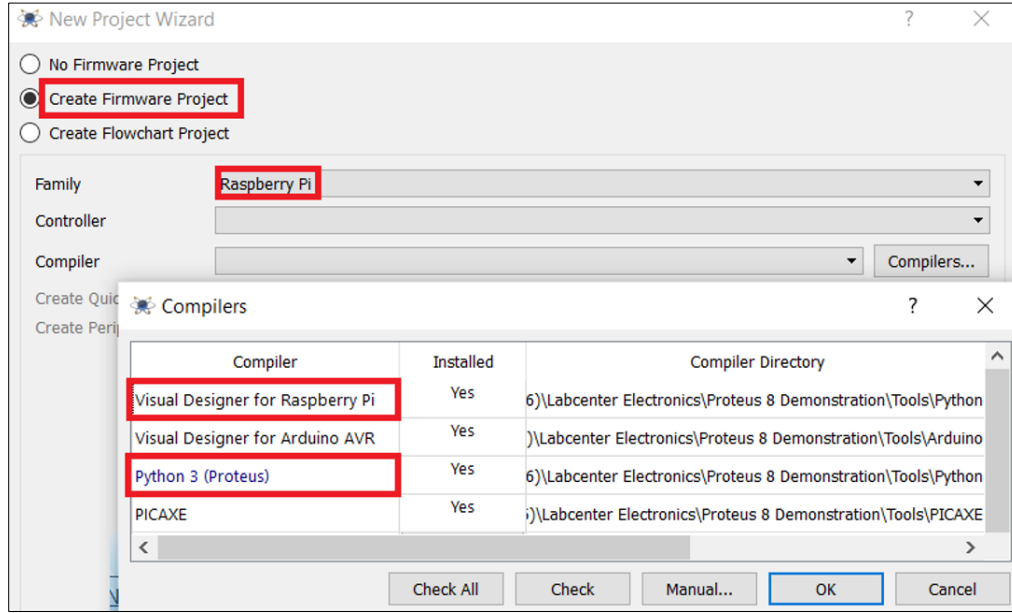
Wokwi başka bir ücretsiz simülatördür, ancak Raspberry Pi 3V+'ı değil Raspberry Pico'yu destekler. Bu bir mikrobilgisayar değildir, işletim sistemi olmayan mikrodenetleyicidir ve GPIO kütüphanesindeki yönergeler kullanılamaz.

Proteus, 1989'dan beri çok iyi bilinen bir elektrik devresi simülasyon uygulamasıdır. Bu uygulamanın daha yeni sürümleri Arduino ve Raspberry Pi simülasyonunu destekler. Raspberry Pi'yi programlamak için Python 3 programlama dili kullanılabilir, ancak blok diyagramlarını oluşturmak için çoğunlukla Visual Designer düzenleyicisi kullanılır.

Bu simülatör ücretsiz değildir. Sunduğu olanakları tanımak istiyorsak, simülatörün ücretsiz deneme sürümünü Proteus'un resmi web sitesinden <https://www.labcenter.com/simulation/> bağlantısından indirebiliriz. Deneme sürümünde zaman sınırı yoktur ancak yeni projelerin kaydedilmesine izin vermez ve Raspberry Pi'yi

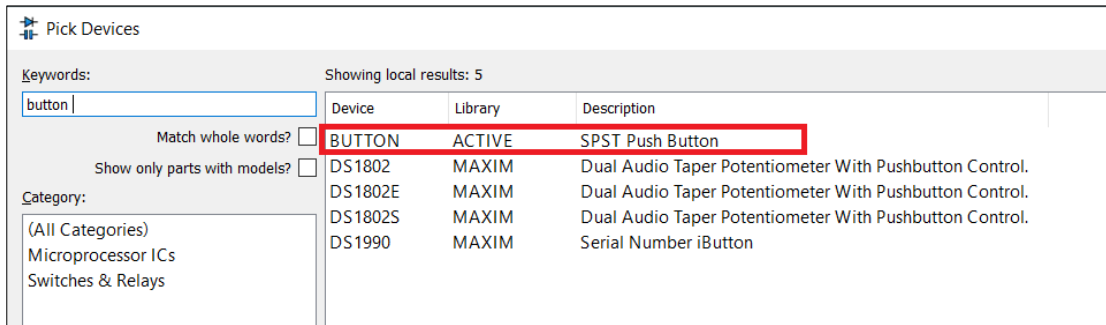
programlamak için sadece Visual Designer düzenleyicisine izin verir. Proje oluşturma prosedürünü kısaca tanıyacağız.

Öncelikle yeni bir proje açılmalı (İng. New Project), isimlendirilmeli ve seçilen dosyaya kaydedilmelidir. Ardından, genel bir tasarım modeli seçtiğimiz (İng. Create a schematic design from the selected template-DEFAULT), elektrik kartı için bir tasarım seçeneğini seçmeyip (İng. Do not create a PCB layout) ve gömülü sistemler için yazılımı (İng. Create Firmware Project) seçtiğimiz birkaç pencere açılır ve ardından denetleyiciyi ve derleyici türünü seçiyoruz. Bu, Şekil 4.43'te gösterilmektedir.



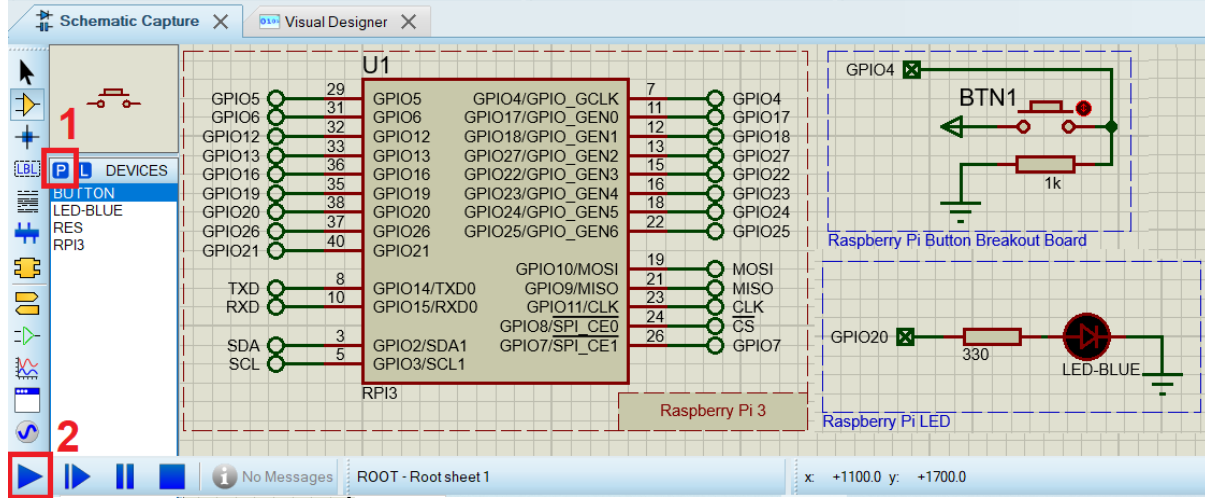
Şekil 4.43. Raspberry Pi 3B+ için derleyici seçimi

Bir sonraki pencerede Finish düğmesine basıldığında, elektrik devresini derlemek için bir pencere (İng. Schematic Capture) ve programı oluşturmak için bir pencere (İng. VSM Studio veya Source Code) olmak üzere iki pencereden oluşan çalışma alanı (masaüstü) açılır. Penceresinin sol üst köşesinde P (İng. Pick Devices) harfiyle işaretlenmiş elektrik devresi kurma için düğme bulunmaktadır (Şekil 4.45.'de 1 numaralı düğme) ve bu düğmeye basıldığında elektrik bileşenlerin seçimi için yeni bir pencere açılır. Bunları anahtar kelimeye göre arayarak buluyoruz (Şekil 4.44).



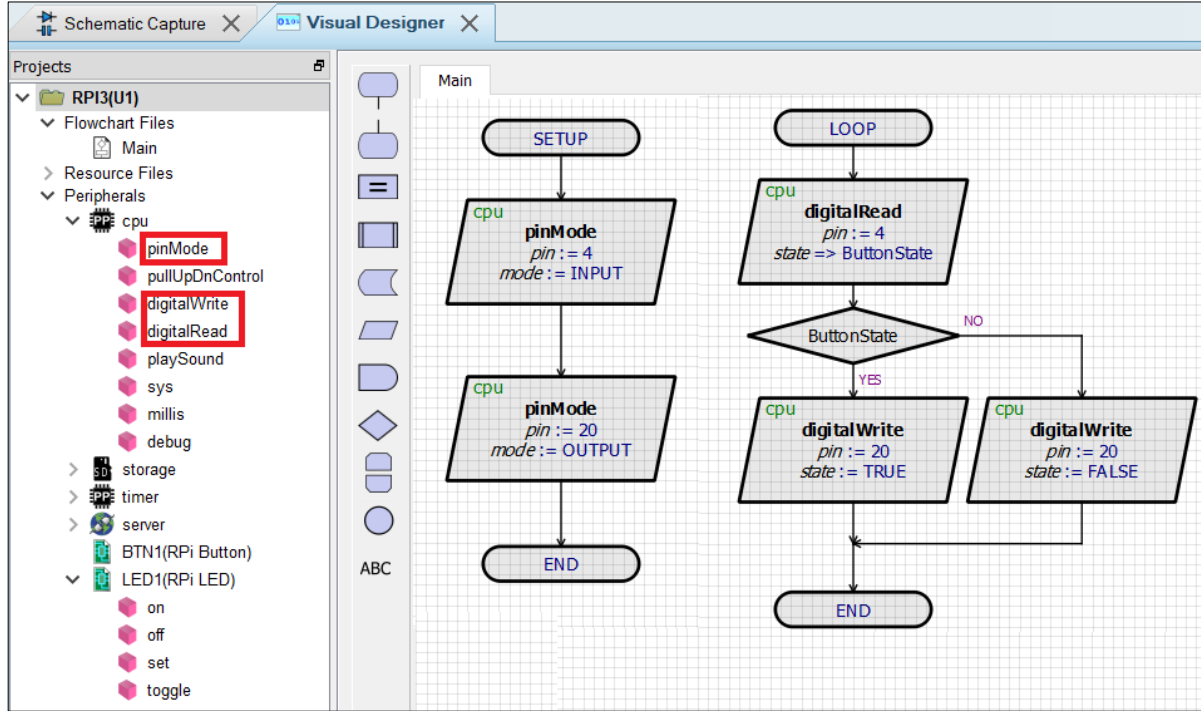
Şekil 4.44. Elektrik devresi için donanım bileşenlerin seçimi

Şekil 4.45.'te düğme, rezistör, LED diyotu ve Raspberry Pi 3B+'dan oluşan elektrik devresi gösterilmektedir. Simülatörde Raspberry Pi 3B+'ın GPIO pinlerine sahip bileşenler hatlarla değil, onların aynı isimlendirilmesiyle birbirine bağlanır.



Şekil 4.45. Elektrik devresi oluşturmak için çalışma alanının görünümü

Elektrik devresini kurduktan sonra Visual Designer metin düzenleyici ile veya Python programlama dilinde programı oluşturmaya geçiyoruz. Şekil 4.46.'da, yukarıdaki elektrik devresinin Visual Designer metin düzenleyicisinde oluşturulan program blok diyagramı gösterilmiştir. Programlama çok basittir.



Şekil 4.46. Visual Designer metin düzenleyicinin çalışma alanının görünümü

Çalışma alanının sol tarafında Project isimli ayrı bir pencerede, kullanılan tüm donanım bileşenleri listelenir ve üzerlerine tıklandığında ilgili bileşenle ilgili tüm yö-

nergelerin bulunduğu açılır menüler açılır. Onların seçimi ve “Sürükle ve bırak” yöntemiyle blokları diyagrama giriyoruz ve program oluşturuyoruz. “Run” düğmesine basıldığında (Şekil 4.45.’de 2 numaralı düğme) simülasyon başlatılır.

Simülatörlerin kullanımı öğrenmeyi kolaylaştırabilir ve öğretim içeriklerini zenginleştirebilir. Simülatörler, elektrik devrelerinin tasarımında kolay değişiklik ve deneme yapılmasına, program kodlarının değiştirilmesine ve olası hataların tespit edilmesine olanak verir ve en önemlisi, tüm bunlar donanım bileşenlerinin satın alınmasını gerektirmez.

Каунакча:

- [1] M. Rafiqzaman, "INTRODUCTION TO MICROPROCESSORS AND MICROCOMPUTER-BASED APPLICATIONS," in *Microprocessors and Microcomputer-Based System Design, 2nd Edition*, CRC Press, 2021.
- [2] [Online]. Available: <https://www.arduino.cc/en/hardware>.
- [3] G. Halfacree, *Raspberry Pi Beginner's Guide How to use your new computer*, Cambridge, CB1 2JH: Raspberry Pi Trading Ltd, 2018.
- [4] И. J. J. Д. Љубица Маркудова, *Дигитална електроника и микропроцесори-III година*, Министерство за образование и наука, 2010.
- [5] M. S. Scott Fitzgerald, *The Arduino Project Book*, Torino, Italy, 2012.
- [6] B. J. N. W. Michael Margolis, *Arduino Cookbook, 3rd Edition*, O'Reilly Media, 2020.
- [7] S. Monk, *Raspberry Pi Cookbook, 3rd Edition*, O'Reilly Media, 2019.
- [8] [Online]. Available: <https://www.alza.co.uk/how-to-build-your-own-PC>.
- [9] D. Wilcher, *Basic Arduino Projects*, Make:Community, 2014.
- [10] Г. Силбершац, *Концепти на оперативните системи*, 2016: Арс Ламина.
- [11] A. Warren, Writer, *Windows 10 Exam 70-698: Installing and Configuring Windows 10 LiveLessons*. [Performance]. Pearson IT Certification, 2018.
- [12] "Arduino Reference," [Online]. Available: <https://www.arduino.cc/reference/en/>.
- [13] "gpiozero-GPIO Zero 1.6.2. Documentation," [Online]. Available: <https://gpiozero.readthedocs.io/en/stable/>.