Mikroişlemciler ve Mikrodenetleyiciler

GİRİŞ

- Mikroişlemci Nedir?
- Mikroişlemcileri Birbirinden Ayıran Özellikler
- Mikroişlemciyi Oluşturan Birimler ve Görevleri
- Bellekler
- Mikrodenetleyiciler
- Mikroişlemci ve Mikrodenetleyiciler Arasındaki
- Mikrodenetleyici Seçimi
- ARDUINO nedir?
- Neden ARDIUNO?
- DONANIM
- YAZILIM
- Örnek Uygulamalar

Mikroişlemci Nedir?

- Mikroişlemciler, bilgisayar sisteminin kalbidir.
- Bilgisayar operasyonlarını kontrol ederek veri işleme işlevlerini yerine getirir.
- CPU (Central Process Unit-Merkezi İşlem Birimi), kullanıcı ya da programcı tarafından yazılan programları meydana getiren komutları veya bilgileri yorumlamak ve yerine getirmek için gerekli olan tüm mantıksal devreleri kapsar.
- İlk mikroişlemci 1971 yılında hesap makinası amacıyla üretilen Intel firmasının 4004 adlı ürünüdür. Bir defada işleyebileceği verinin 4 bit olmasından dolayı 4 bitlik işlemci denilmekteydi.

Mikroişlemcileri Birbirinden Ayıran Özellikler

1)Kelime Uzunluğu: Mikro işlemcinin **her saat darbesinde işlem yapabileceği bit sayısına** kelime uzunluğu denir. İşlenen veriler işlemcinin özelliğine göre 4-bit, 8-bit, 16-bit, 32-bit ve 64-bit uzunluğunda olabilir. Kelime uzunluğu **veri yolu uzunluğuna** eşittir.

2)Komut İşleme Hızı: Mikro işlemcilerin çalışması için **saat sinyallerine** ihtiyaç vardır. İşlemci (CPU) her saat sinyalinde bir sonraki işlem basamağına geçer. Saat frekansı mikro işlemciye dışardan uygulanan ya da işlemcinin içinde bulunan osilatörün frekansıdır. Komut çevrim süresi ise herhangi **bir komutun görevini tamamlayabilmesi için geçen süredir**.

Mikroişlemcileri Birbirinden Ayıran Özellikler

3)Adresleme Kapasitesi: Bir işlemcinin adresleme kapasitesi, adresleyebileceği veya **doğrudan erişebileceği bellek alanının büyüklüğüdür**. Bu büyüklük işlemcinin adres hattı sayısına bağlıdır. Bu hattın sayısı tasarlanacak sistemde kullanılabilecek bellek miktarını da belirlemektedir.

4)Kaydedici Sayısı: Mikro işlemcilerde kaydediciler, genel amaçlı kaydediciler ve özel amaçlı kaydediciler olmak üzere iki grupta toplanır. Bu kaydediciler 8, 16, 32 ve 64- bitlik olabilir. Kaydedicilerin sayısının programcının işinin kolaylaştırmasının yanında programın daha sade ve anlaşılır olmasını da sağlar. Her mikro işlemcinin kendine has yapısı ve kaydedici isimleri vardır. Herhangi bir mikro işlemciyi programlamaya başlamadan önce mutlaka bu kaydedicilerin isimlerinin ve ne tür işlevlere sahip olduklarının iyi bilinmesi gerekir.

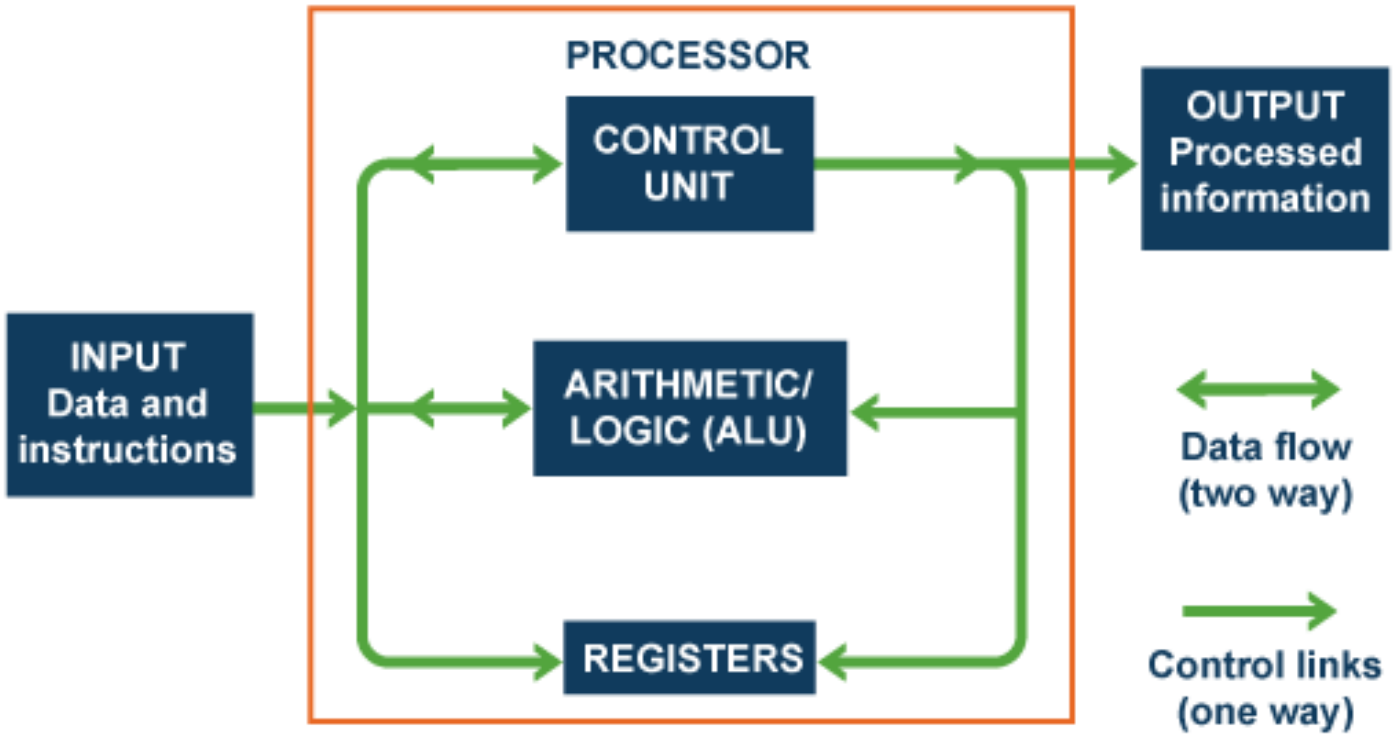
Mikroişlemcileri Birbirinden Ayıran Özellikler

5)Farklı Adresleme Modları: Bir komutun işlenmesi için gerekli verilerin bir bellek bölgesinden alınması veya bir bellek bölgesine konulması ya da bellek–kaydedici veya kaydedici–kaydedici arasında değiştirilmesi için farklı erişim yöntemleri kullanılır. Mikro işlemcinin işleyeceği bilgiye farklı erişim şekilleri, **"adresleme yöntemleri"** olarak ifade edilir. Kısaca **adresi tarif yollarıdır.**

Adresleme türleri;

- ❖Doğrudan adresleme
- ❖Dolaylı adresleme
- ❖Veri tanımlı adresleme
- ❖Kaydedici adresleme
- ❖Mutlak adresleme
- ❖Göreceli adresleme
- ❖İndisli adresleme
- ❖Akümülatör ve imalı adresleme

Mikroişlemciyi Oluşturan Birimler ve Görevleri



Mikroişlemciyi Oluşturan Birimler ve Görevleri

1)Kaydediciler (Registers)

Kaydediciler, genel ve özel amaçlı olmak üzere iki gruba ayrılır. Bunlardan başka programcıya gözükmeyen (ilgilendirmeyen) kaydediciler de vardır. Genel amaçlılara akümülatör, indis kaydedicileri girmektedir. Özel amaçlılar ise program sayıcı, yığın işaretçisi, bayraklar gibi kaydediciler girmektedir.

A)Akümülatör: Akümülatörler ,işlemcinin aritmetik ve mantık işlemleri sırasında depo görevi yapan önemli bir kaydedicidir. **Ara değerlerin üzerinde tutulması**, sisteme gelen verinin ilk alındığı yer, belleğe veya dış dünyaya gönderilecek verilerin tutulduğu yer olarak görev yapar. Bazı işlemcilerde B kaydedicisi de yardımcı akümülatör olarak kullanılır.

B)İndis Kaydedicileri: İndis kaydedicilerinin temelde üç görevi vardır. Hesaplamlarda **ara değerlerin geçici tutulmasında**, **program döngülerinde** ve zamanlama uygulamalarında **bir sayıcı olarak** ve bellekte depolanmış bir dizi verinin üzerinde bir **indisçi** olarak kullanılmaktadır.

Mikroişlemciyi Oluşturan Birimler ve Görevleri

1)Kaydediciler

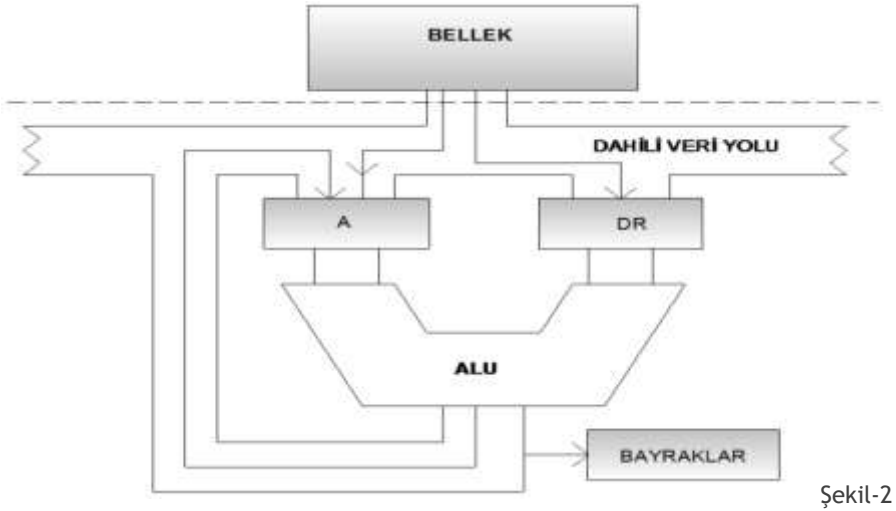
C) Program Sayıcı (PC, Program Counter) :Mikro işlemcinin yürütmekte olduğu **program komutlarının adres bilgisini** tuttuğu özel amaçlı bir kaydedicidir. Mikroişlemcinin çalışması sırasında hangi komutun hangi sırada kullanılacağının bilinmesi gerekir. Bu görevi program sayıcı (PC) yerine getirir. Bellekten alınan her komut kodundan sonra alınacak yeni komut kodunun adresi program sayıcıya otomatik olarak işlemci tarafından yüklenir. Komut çevrimi, PC'nin yeni adresi adres yoluna koyması ile başlar. Bunun ardından da ilgili kontrol sinyali gönderilir. Bellekten gelen her bilgiden sonra PC, kontrol devresinden aldığı işarete uyarak adres satırını 1 arttırır. Böylece bilgilerin bellekten işlemciye düzenli bir şekilde gelmesi sağlanır.

D)Durum Kaydedicileri (Bayraklar): Mikro işlemci içinde veya dışardan yapılan herhangi aritmetiksel, mantıksal veya kesmelerle ilgili işlemlerin sonucuna göre bitleri değer değiştirir. Bir işlem sonucunda bu bitlerin aldığı değere göre program yön bulur. Programcı bu bitlerde oluşacak değerlere göre programa yön verebilir. Elde Bayrağı-C, Sıfır Bayrağı-Z, Ondalık Bayrağı-D, Taşma Bayrağı-V, Negatif Bayrağı-N gibi çeşitleri vardır.

Mikroişlemciyi Oluşturan Birimler ve Görevleri

2)Aritmetik ve Mantık Birimi (ALU)

Mikroişlemcinin en önemli kısmını aritmetik ve lojik birimi (ALU) oluşturur. Bu ünite kaydediciler üzerinde toplama, çıkarma, karşılaştırma, kaydırma ve döndürme işlemleri yapar. Yapılan işlemin sonucu kaydediciler üzerinde saklanır. Bazen de yalnızca durum kodu kaydedicisini etkiler. ALU'daki bir işlem sonucunda durum kodu kaydedicisindeki bayrakların birkaçı etkilenebilir veya hiçbiri etkilenmez. Programcı için çoğu zaman ALU'da yapılan işlemin sonucunda etkilenen bayrakların durumu daha önemlidir. Gelişmiş mikroişlemcilerin içindeki ALU'lar çarpma ve bölme işlemlerini yapabilmektedir. ALU'nun işlem yapabileceği en büyük veri, mikro işlemcideki kaydedicilerin veri büyüklüğü ile sınırlıdır.



Şekil-2

Mikroişlemciyi Oluşturan Birimler ve Görevleri

2)Aritmetik ve Mantık Birimi (ALU)

A)Aritmetik İşlemler : ALU'da yapılan aritmetiksel işlemler mikro işlemcinin yapısına göre çeşitlilik gösterebilir. Toplama, Çıkarma, Çarpma, Bölme, Sağa veya sola kaydırma, içerik artırma veya azaltma ve döndürme işlemleri gibi işlemleri kapsar. Gelişmiş işlemcilerde büyük ondalıklı sayılarla işlem yapmak için ayrıca matematik işlemci mevcuttur.
B)Mantıksal İşlemler : VE, VEYA, Özel VEYA (XOR), Değil (NOT), Karşılaştırma (=, =<, =>, <> gibi) gibi işlemleri kapsar.

Bütün bu işlemler teknolojik yapısı değişik kapı ve **flip-flop'lardan** oluşan bir sistem tarafından yürütülmektedir.

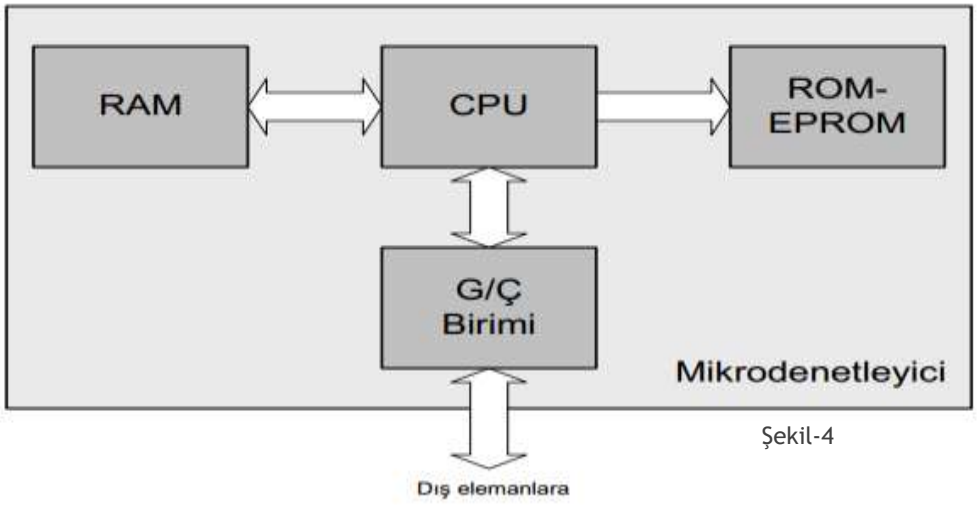
3)Kontrol Birimi

Kontrol birimi, sistemin tüm işleyişinden ve işlemin zamanında yapılmasından sorumludur. Kontrol birimi, bellekte program bölümünde bulunan komut kodunun alınıp getirilmesi, kodunun çözülmesi, ALU tarafından işlenmesi ve sonucun geri belleğe konulması için **gerekli olan kontrol sinyalleri üretir**.

Mikrodenetleyiciler

Bir mikroişlemci çekirdeğine ilave olarak, ortak bellek alanlarını kullanan, özelleştirilmiş görevler ile donatılmış çevrebirimlerin eklenmesi ile ortaya çıkan yapıya mikrodenetleyici denir. Denetim teknolojisi gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikrodenetleyiciler, mikro işlemcilere göre çok **daha basit ve ucuzdur**.

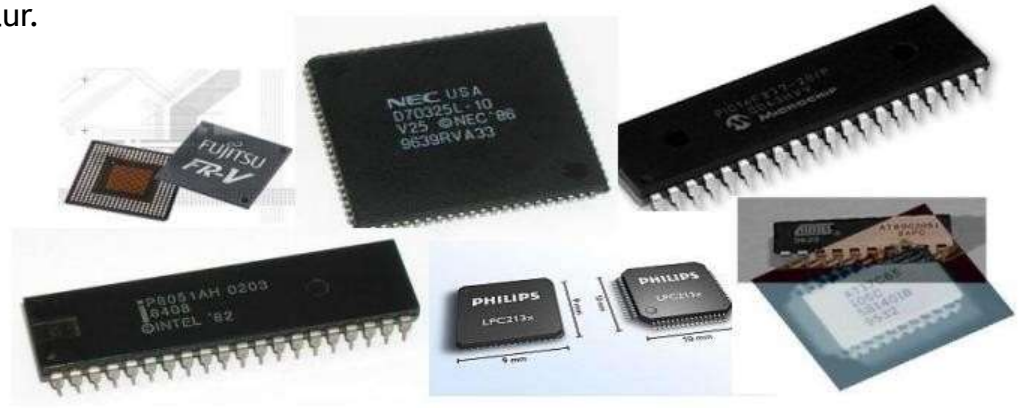
❖Mikroişlemcili sistemin tasarımı ve kullanımı mikrodenetleyicili sisteme göre daha karmaşık ve masraflıdır.
❖Mikrodenetleyicili bir sistemin çalışması için elemanın kendisi ve bir osilatör kaynağının olması yeterlidir.
❖Mikrodenetleyicilerin küçük ve ucuz olmaları, bunların tüm elektronik kontrol devrelerinde kullanılmasını sağlamaktadır.



Şekil-4

Mikroişlemci ve Mikrodenetleyiciler Arasındaki Farklar

Bir mikroişlemci görevini yerine getirebilmesi için mutlaka, verilerin saklanacağı bellek birimine, dış dünyadan veri alışverişinin düzenli yapılmasını sağlayan giriş/çıkış birimine ihtiyaç duyar. Bunlar bir mikroişlemcili sistemde ayrı ayrı birimler (entegreler) şeklinde yerini alır. Bundan dolayı mikroişlemcili sistemlere çok entegreli sistemler denilir. Bilgisayar gibi mikroişlemcili sistemlere verilen bir örnekte, bir bilgisayarın bir çamaşır makinesinde veya cep telefonunda kullanılması elbette mümkün olmayacaktır. Bilgisayar aynı anda milyonlarca işi yapabildiğinden ve çok yer kapladığından böyle yerlerde kullanılması mantıklı olmaz ve maliyetli olur. Bundan dolayı, sistemi meydana getiren elemanların birçok özelliklerinden feragat edilerek ve bir entegrede birleştirilerek mikroişlemcilerin yeni türevleri (mikrodenetleyiciler) oluşturulmuştur. Bir saydırma sinyali üreteceğimizi düşünürsek mikroişlemci ile bunu yazılımsal olarak yapmamız gerekecektir. Ancak mikrodenetleyicinin özelleştirilmiş modülleri sayesinde bu işlemleri programa paralel olarak modüllerle yapabiliriz.Böylece ana programdaki yoğunluk azalır ve işlemcimiz hızlanmış olur.



Şekil-5

Mikrodenetleyici Seçimi

Mikrodenetleyiciler ile tasarım yapmadan önce tasarlanan sisteme uygun bir denetleyici seçmek için o denetleyicinin taşıdığı özelliklerin bilinmesi gereklidir. Mikrodenetleyicinin hangi özelliklere sahip olduğu kataloglarından anlaşılabilir. Aşağıda sıralanan özellikler bunlardan bazılarıdır;

- Programlanabilir dijital paralel giriş/çıkış.
- Programlanabilir analog giriş/çıkış.
- Seri giriş/çıkış (senkron, asenkron ve cihaz yönetimi).
- Motor veya servo kontrol için pals sinyali çıkışı.
- Harici giriş vasıtasıyla kesme.
- Harici bellek arabirimi.
- Harici veri yolu arabirimi.
- Dahili bellek tipi seçenekleri (ROM, EPROM, PROM, EEPROM).
- Dâhilî RAM seçeneği.
- Kayan nokta hesaplaması.


ARDUINO

ARDUINO Nedir?

Arduino, açık kaynak kodlu yazılım ve donanıma sahip bir **mikrodenetleyici platformudur**. Açık kelimesi ile gerçek anlamda açık tasarımı ifade edilmektedir. Baskılı devresi, şematik tasarımı, PC üzerinde çalışan derleyicisi, kütüphaneleri ve **tüm detayları ile internet ortamında paylaşılmaktadır**.

Şekil-7

Neden ARDUINO?

- ❖ Hem donanımsal hem de yazılımsal olarak açık kaynaklı.
- ❖ Ucuz.
- ❖ Alt seviye mikroişlemci bilgisi gerektirmez.
- ❖ Çok zengin kütüphaneleri vardır.
- ❖ Ek bir programlama devresi gerektirmez.
- ❖ USB üzerinden haberleşebilir.
- ❖ Shield kartları sayesinde ileri teknolojiler kolayca entegre edilebilir.
- ❖ Temel elektroniğin bilinmesi uygulama geliştirmek açısından yeterlidir.

DONANIM

Arduino Uno

Arduino Leonardo

Arduino Due

Arduino Tre

Arduino Yun

Arduino Micro

Arduino Robot

Arduino Esplora

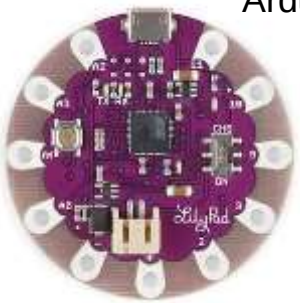
DONANIM

Arduino Mega ADK

Arduino Ethernet

Arduino Mega 2560

Arduino Mini



LilyPad Arduino USB

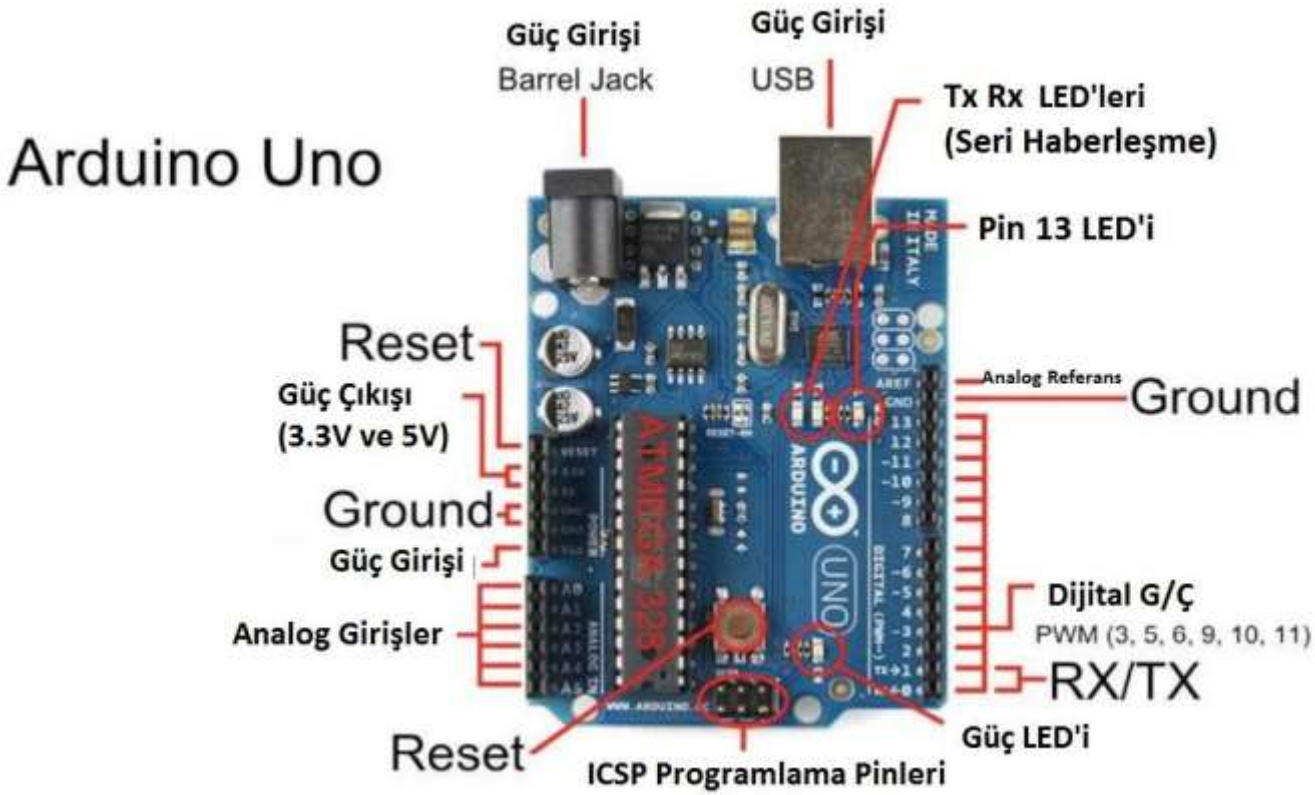
LilyPad Arduino Simple

LilyPad Arduino SimpleSnap

LilyPad Arduino

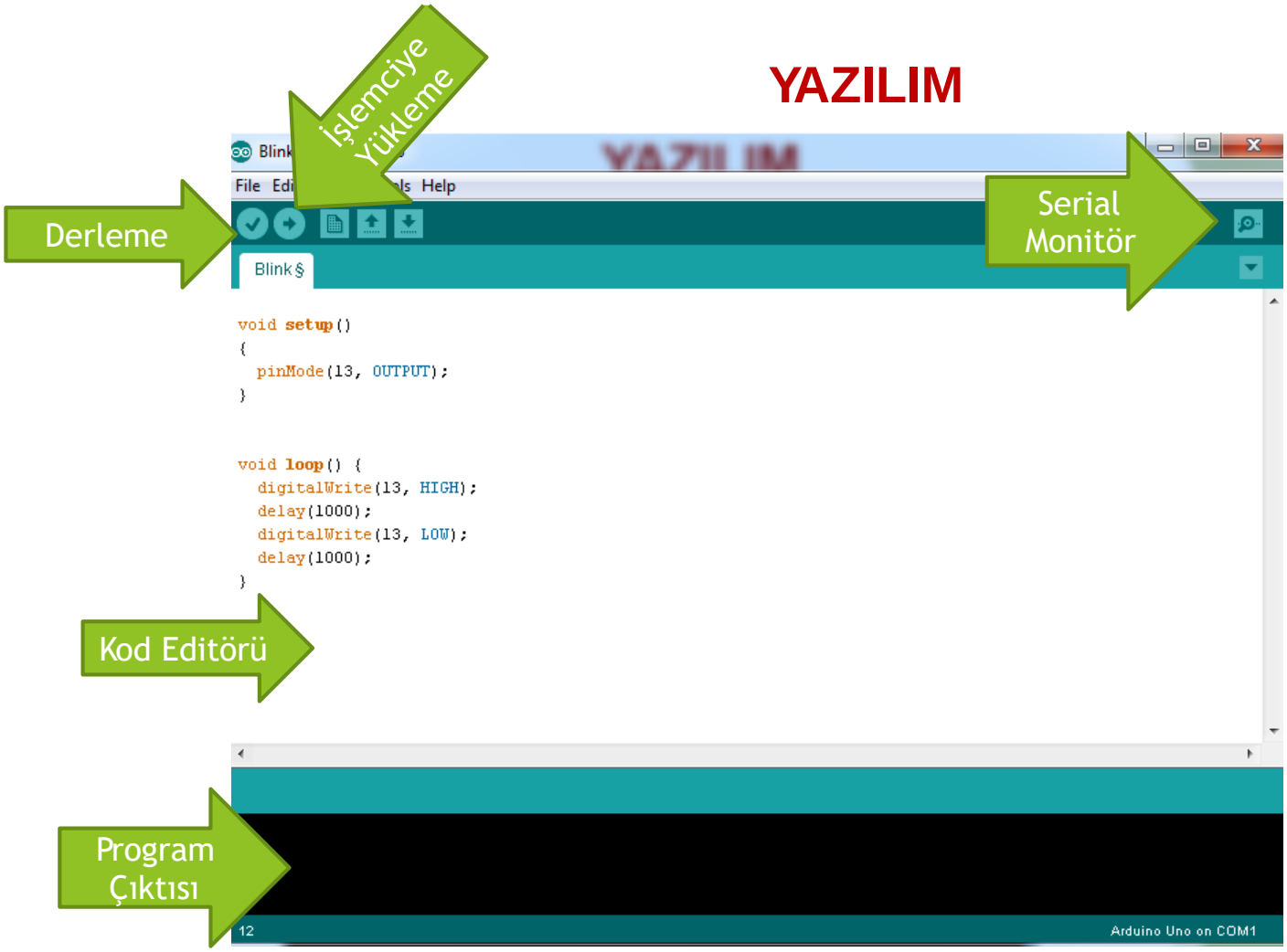
Arduino Nano

DONANIM



Şekil-8

YAZILIM



Şekil-9

YAZILIM

setup() : Programın **başlangıç** ayarları içindir sadece başlangıçta tek seferlik çalışır. Bu fonksiyon değişkenler, pin modları, seri iletişim, kütüphaneler vb. için kullanılır.

Loop() : Loop Türkçesi döngüdür. Adından anlaşılacağı gibi setup() fonksiyonundan sonra **döngü** şeklinde sürekli çalışır. Ana program kodları bu fonksiyon içine yazılır

```
//Tanımlamalar
void setup() {
// Kurulum kodları buraya yazılır. (Bir defa çalışır)
}
void loop() {
// Ana program kodları buraya yazılır. (Sürekli çalışır)
}
```

YAZILIM

- Program yazımı belirli kalıpta, bloklar halinde olur.
- Bloklar, { } parantezleri ile oluşturulur.
- Komutlar aynı veya alt alta satırlara yazılabilirler.
- Tüm komutlar, noktalı virgül (;) ile biter. Yalnız blok başlatan ifadelerden sonra noktalı virgül kullanılmaz.
- Programda kullanılan tüm değişkenler ve bilgi tipleri bildirilir.
- Programın başında kütüphaneler aktifleştirilir/çağrılır.
- Açıklamalar “//” ve “/* */ ” (Birden fazla satır için) ile yazılır.
- #define ile eşdeğer ifade atanır.
- #include ile kütüphane çağrılır.

YAZILIM

pinMode(pin no, pin türü);
Pinleri giriş veya çıkış olarak yapılandırma işlemi yapar.

digitalWrite(pin no, lojik değer);
Çıkış olarak ayarlanan pinlerin değerlerini, HIGH veya LOW olarak ayarlar.

digitalRead(Pin no);
Belirtilen digital pin değerini okur.

analogReference();
Analog giriş için referans gerilimini ayarlar.

analogRead(Pin no);
Belirtilen analog pin değerini okur.

analogWrite(pin no, dijital değer);
Ayarlanan pinden analog çıkış almayı sağlar. LED parlaklığı, motor hızı ayarlama gibi işlemlerde kullanılır.

delay(Milisaniye);
Milisaniye biriminde zaman gecikmesi verir.

Örnek Uygulamalar

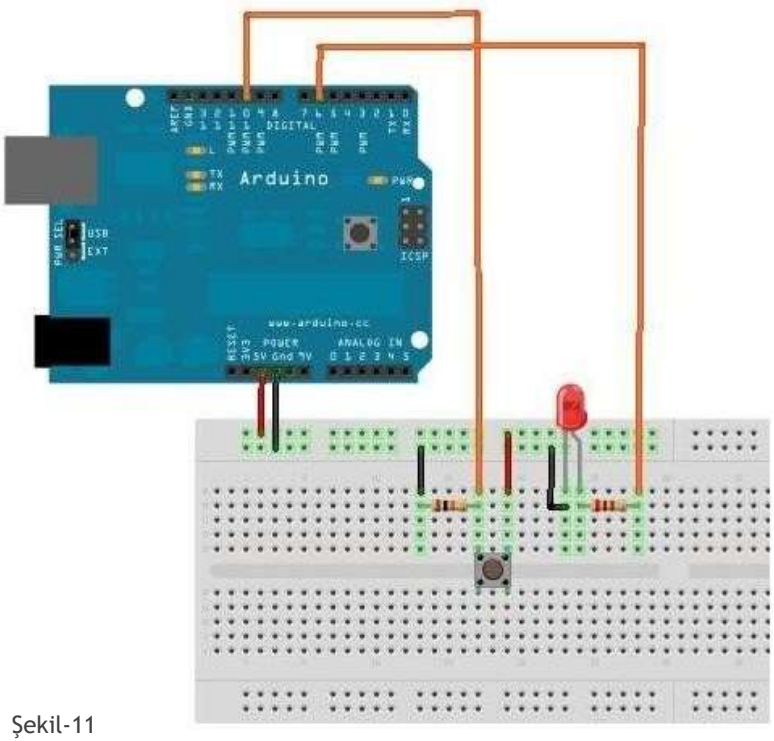
LED Uygulaması

Şekil-10

```
1.  #define LED 13
2.  void setup()
3.  {
4.    pinMode(LED, OUTPUT);
5.  }
6.  void loop() {
7.    digitalWrite(LED, HIGH);
8.    delay(1000);
9.    digitalWrite(LED, LOW);
10.   delay(1000);
11.}
```

Örnek Uygulamalar

Dijital Okuma-Seri İletişim-LED uygulaması

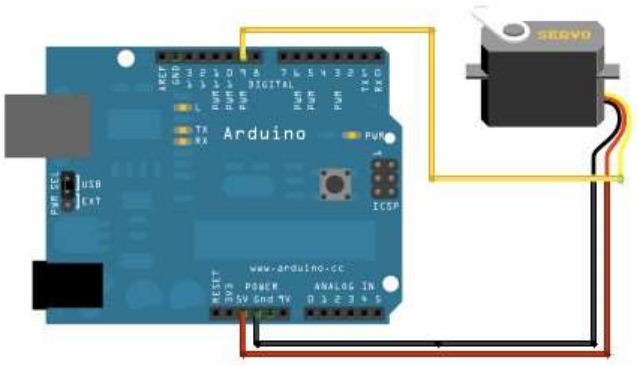


Şekil-11

```
1.  int Buton = 10;
2.  İnt LED = 6;
3.  void setup()
4.  {
5.    Serial.begin(9600);
6.    pinMode(Buton, INPUT);
7.    pinMode(LED, OUTPUT);
8.  }
9.  void loop()
10.{
11.  int butonDurumu = digitalRead(Buton);
12.  digitalWrite(LED, butonDurumu);
13.  Serial.println(butonDurumu);
14.  delay(1);
15.}
```

Örnek Uygulamalar

Servo Motor Uygulaması



Şekil-14

```
1.  #include <Servo.h>
2.  Servo SERVO1;
3.  int pozisyon = 0;
4.  void setup()
5.  {
6.    SERVO1.attach(9);
7.  }
8.  void loop()
9.  {
10.   for(pozisyon = 0; pozisyon <= 180; pozisyon++)
11.   {
12.     SERVO1.write(pozisyon);
13.     delay(15);
14.   }
15.   for(pozisyon = 180; pozisyon>=0; pozisyon--)
16.   {
17.     SERVO1.write(pozisyon);
18.     delay(15);
19.   }
20. }
```