

ARDUINO

Seri Port Ekranı

- Arduinoyu USB üzerinden bilgisayarımıza bağladığımızda aslında seri port bağlantısı yapmış oluyoruz. Bu seri port üzerinden hem bilgi alışverişi yapabilir hem de uç bağlantı noktasındaki, yani Arduino'daki veriyi kontrol edebiliriz.
- `Serial.begin(9600);` «9600» Arduino ve bilgisayar arasındaki haberleşmenin hızını yani bant genişliğini (Bandwidth) belirtiyor.
- `Serial.print` seri ekrana yazı yazmamızı sağlayan komuttur.

Örnek

- `char gelenVeri=0;`
- `void setup(){`
- `Serial.begin(9600);`
- `}`
- `void loop()`
- `{`
- `if(Serial.available()>0)`
- `{`
- `gelenVeri=Serial.read();`
- `Serial.println("gelenveri:");`
- `Serial.println(gelenVeri);`
- `}`
- `}`
- `}`

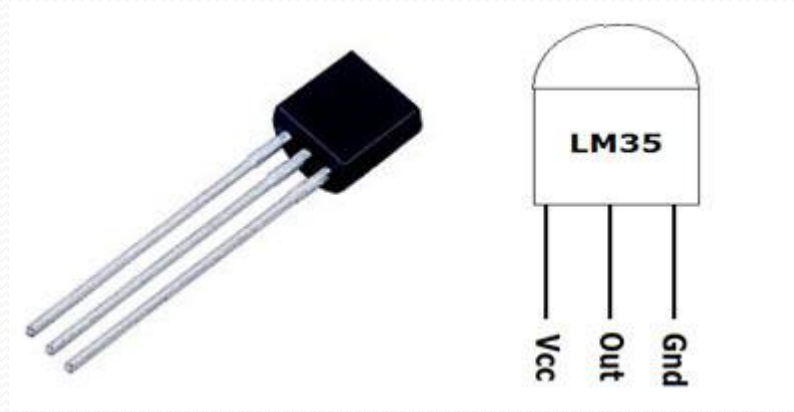
analogRead()

- Belirlenen analog pinden deęer okumak için kullanılır.Arduino,10 bitlik analog-dijital konverter(ADC) barındırır.
- 0-5 Volt arasındaki deęerleri 0-1023 arasında bir deęere atar. Her bir deęer $5 \text{ volt} / 1024 = 0.0049 \text{ volt}$ (4.9 mV)'a karşılık gelir.Arduino her 100 mikrosaniyede bir deęer okur yani saniyede 10,000 kere analog deęer okunur.
- Analog pinler için giriş,çıkış belirlenmesine gerek yoktur.

Örnek

```
int analogPin = 3;
int val = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.println(val);
}
```

LM 35



- Lm 35,her bir santigrad deęeri için 10mv gerilim üretir.
- -55°C den 150°C ye kadar ölçüm yapabilir.

Örnek

- `const int lm_35 = A0;`
- `int sensor_deger = 0;`
- `float voltaj_deger = 0;`
- `float sicaklik_deger = 0;`

- `void setup() {`

- `Serial.begin(9600);`
- `}`

- `void loop() {`

- `sensor_deger = analogRead(lm_35);`
- `Serial.print("okunan_deger = ");`
- `Serial.println(sensor_deger);`
- `voltaj_deger = (sensor_deger / 1023.0) * 5000;`
- `Serial.print("gelen voltaj=");`
- `Serial.println(voltaj_deger);`
- `sicaklik_deger = voltaj_deger / 10.0;`
- `Serial.print("sicaklik = ");`
- `Serial.println(sicaklik_deger);`
- `delay(250);`
- `}`

analogWrite()

- Bir pine herhangi bir analog değer(PWM dalgası)yazmak için kullanılır.
- Bir LED'in farklı parlaklık değerlerinde yakılması ya da bir motorun farklı hızlarda çalıştırılması gibi uygulamalarda kullanılabilir.
- Arduino'nun PWM pinlerinin frekansı yaklaşık olarak 490Hzdir ancak 5. ve 6. pinlerin frekansı yaklaşık 980 Hzdir.

- analogRead değerleri 0 ile 1023 arasındayken, analogWrite değerleri 0-255 arasındadır.
- analogWrite(pin,değer)

PWM Nedir?

- Açılımı Pulse Width Modulation yani Sinyal Genişlik Modülasyonu olan bu teknik, sinyal işleme veya sinyal aktarma gibi daha çok elektronik devrelerin yanı sıra Arduino veya elektrik makineleri gibi özel uygulama alanlarında da yer alan bir tekniktir.

- Arduino bünyesinde de PWM tekniđi kullanılabilir. Arduino bünyesinde kullanılan PWM tekniđi ile dijital sonuçlardan analog sonuçlar elde edilebilir. Bunun yanı sıra özellikle kontrol için ihtiyaç olan kare dalga üretimi de gerçekleşiyor. Kare dalga, bilindiđi gibi «on» ve «off» konumlarını sağlıyor.

- Böylece kare dalga gönderildiğinde "on" konumunda 5V uygulanırken, "off" konumunda 0V uygulanmış oluyor. İşte bu "on" kısmının aktif olduğu "**Sinyal Genişliği**" adı veriliyor. Bu doğrultuda istenilen sinyal genişliği elde etmek için de modülasyon tekniği uygulamanız gerekiyor ki bu da PWM'in temel mantığıdır.

Pulse Width Modulation

0% Duty Cycle - `analogWrite(0)`



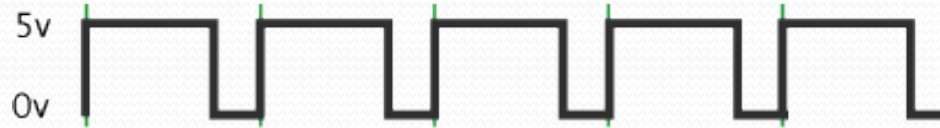
25% Duty Cycle - `analogWrite(64)`



50% Duty Cycle - `analogWrite(127)`



75% Duty Cycle - `analogWrite(191)`



100% Duty Cycle - `analogWrite(255)`



Pot ile Led Parlaklığı

- `int ledPin = 9; // LED`
- `int analogPin = 3; // Pot`
- `int val = 0;`
- `void setup()`
- `{`
- `pinMode(ledPin, OUTPUT);`
- `}`
- `void loop()`
- `{`
- `val = analogRead(analogPin);`
- `analogWrite(ledPin, val / 4); // analogRead değerleri 0 ile 1023`
`arasındayken, analogWrite değerleri 0-255 arasındadır.`
- `}`

Alev Efekti

- `int ledPin1 = 9;`
- `int ledPin2 = 10;`
- `int ledPin3 = 11;`

- `void setup() {`
- `pinMode(ledPin1,OUTPUT);`
- `pinMode(ledPin2,OUTPUT);`
- `pinMode(ledPin3,OUTPUT);`
-
- `}`

- `void loop() {`
- `analogWrite(ledPin1,random(120)+135);`
- `analogWrite(ledPin2,random(120)+135);`
- `analogWrite(ledPin3,random(120)+135);`
- `delay(100);`
- `}`

PWM ile Işık Şiddet Kontrolü

- `const int led = 3;`
- `void setup() {`
- `pinMode(led,OUTPUT);`
- `}`
- `void loop() {`
- `for(int i=0;i<=255;i++)`
- `{`
- `analogWrite(led,i);`
- `delay(20);`
- `}`
- `}`

Farklı Frekanslarda PWM

- `#include "PWM.h" //Pwm kütüphanesi program içine aktarıldı.`
-
- `int pwm_pin=9; // Pwm pinlerinden 9 numaralı pin atanmıştır.`
- `int32_t frekans =1; // Pinin frekans değeri 1 Hz olarak atanmıştır.`
-
- `void setup()`
- `{`
- `InitTimersSafe(); // Zaman tutma fonksiyonları kuruldu.`
- `SetPinFrequencySafe(pwm_pin, frekans); // Pine belirlenen frekans atanmıştır.`
-
- `}`
-
- `void loop()`
- `{`
- `// Uygulanan duty cycle değeri %50 olarak atanmıştır.`
- `// Bunun anlamı 0.5 saniye ON 0.5 saniye OFF durumudur.`
- `pwmWrite(pwm_pin,127);`
- `}`