



MİKRODENETLEYİCİLER (ARDUINO)

4. Hafta

Hazırlayan: Öğ. Gör. Sabri Uzuner

https://www.researchgate.net/profile/Sabri_Uzuner

NE ÖĞRENECEĞİZ?

- Fonksiyonlar



Fonksiyon nedir?

Fonksiyonlar, bir program içerisinde aynı işlem defalarca yapılıyorsa, her seferinde aynı kodları yazmak yerine o işlem için bir fonksiyon yazılıp o fonksiyonu gereken yerde çağırmaya yarar.

Fonksiyonlar **parametre alan** ya da **almayan**, **geriye değer döndüren** ya da **döndürmeyen** olarak sınıflandırılır.

```
#include <stdio.h>
void mesajyaz ()
{
    printf("Ben bir
fonksiyonum !\n");
}
```

```
int main ()
{
    mesajyaz ();
    return 0;
}
```



FONKSİYON ANATOMİSİ

- Fonksiyon tipi(çıkış parametre türü) : int
- Fonksiyon adı: toplama
- Fonksiyon giriş parametreleri: a,b

Anatomy of a C function

Datatype of data returned, any C datatype.

"void" if nothing is returned.

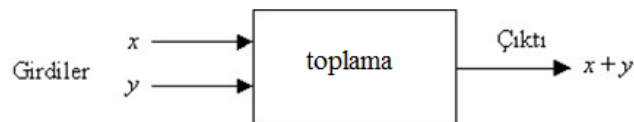
Parameters passed to function, any C datatype.

Function name

```
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Return statement, datatype matches declaration.

Curly braces required.



- // basit parametrelili fonksiyon
örneği

```
#include <stdio.h>
```

```
int toplama (int a, int b)
```

```
{  
    int r;  
    r=a+b;  
    return (r);  
}
```

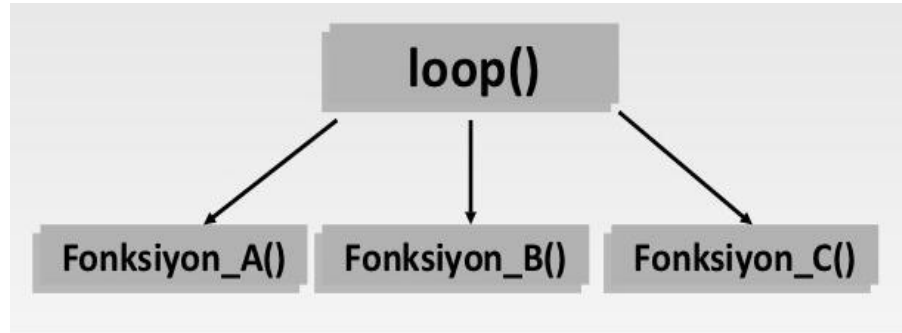
```
int main ()
```

```
{  
    int z;  
    z = toplama (5,3);  
    printf("%d", z);  
    return 0;  
}
```



ARDUINODA FONKSİYONLAR

Fonksiyonlar bazı alt işlemleri gruplara ayırıp, modüler olarak çözmek için yazılırlar. Arduino fonksiyon yazımı, C dilindeki fonksiyon yapısına çok benzer. Fonksiyonlar loop, setup veya başka bir fonksiyon içerisinde çağrılabilirler.



FONKSİYONLAR DÖNÜŞ PARAMETRELERİ

Dönüş değeri tipi

```
int ornekBirFonksiyon(int sayi1, int sayi2) {  
    /* Parametre olarak verilen iki tam sayının  
       toplamını döndüren fonksiyon */  
    int sonuc;  
    sonuc = sayi1 + sayi2;  
  
    return sonuc;  
}
```

1. Parametre **2. Parametre**

Dönüş değeri yok (void)

```
void ArduinoYazdir() {  
    /* Seri çıkış üzerinden "Arduino" yazdıran  
       fonksiyon  
       */  
    Serial.println("Arduino");  
}
```



ARDUINO PROGRAMLAMA DİLİ

FONKSİYON PARAMETRELERİ

Anatomy of a C function

Datatype of data returned,
any C datatype.

"void" if nothing is returned.

Parameters passed to
function, any C datatype.

Function name

```
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Return statement,
datatype matches
declaration.

Curly braces required.

- Arduino Programlama dili c,c++ yapısında fonksiyonlar kullanır. Burada fonksiyon parametre alabilir, ve çağrıldığı yere değer döndürebilir.



PARAMETRE ALMAYAN, GERIYE DEĞER DÖNDÜRMEYEN FONKSİYON:

Bu tip fonksiyonlara herhangi bir değer gönderilmez ve geriye değer döndürmesi beklenmez. Geriye değer döndürmediğini belirtmek için “void” komutu ile başlar.

Sözdizimi:

Fonksiyon tanımlama: `void FonksiyonAdi(){ }`

Fonksiyonu çağırma: `FonksiyonAdi();`

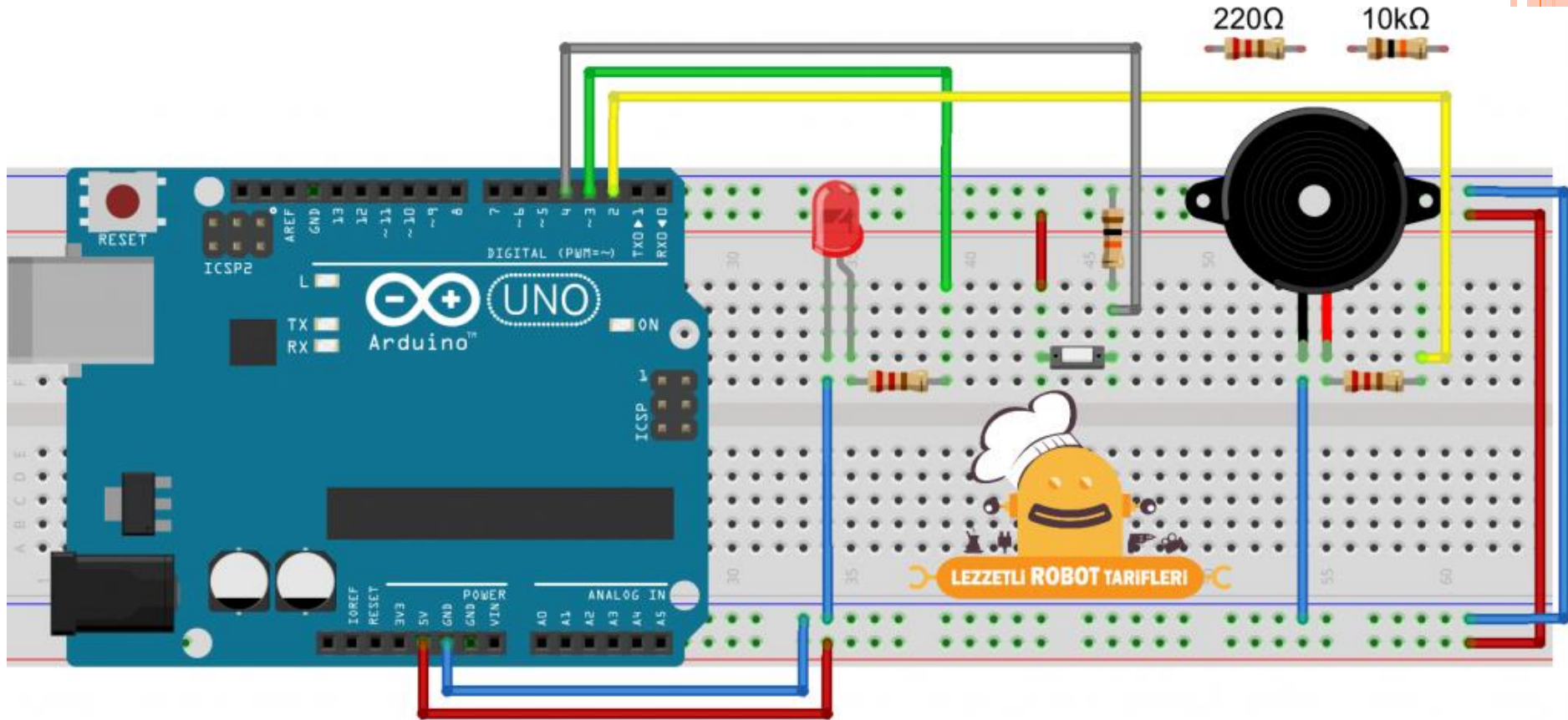


Malzeme Listesi

- 1 adet buton
- 2 adet 220ohm direnç
- 1 adet 10K direnç
- Buzzer ve kablolar

21. Parametre almayan ve değer döndürmeyen fonksiyon

Devre Şeması



Program Kodu

```
const int led_pin = 3; //pinler atandı
const int buton_pin = 4;
const int buzzer_pin = 2;

int buton_durum = 0; //butonun durumunu saklayacak olan değişken

void basla() // değer döndürmeyen parametre almayan "basla" fonksiyonu
{
    digitalWrite(led_pin, HIGH); // ledi yak
    delay(200); //bekle
    digitalWrite(led_pin, LOW); //ledi söndür
    delay(200); //bekle
}

void baslama() // değer döndürmeyen parametre almayan "baslama" fonksiyonu
{
    digitalWrite(buzzer_pin, HIGH); //buzzer çal
    delay(200); //bekle
    digitalWrite(buzzer_pin, LOW); // buzzer sustur
```



Program Kodu

```
void setup()
{
  pinMode(led_pin ,    OUTPUT); // led pini çıkış atandı
  pinMode(buton_pin,   INPUT); // btton pini giriş atandı
  pinMode(buzzer_pin,  OUTPUT); // buzzer pini çıkış atandı
}

void loop()
{
  buton_durum = digitalRead(buton_pin); // buton pinini oku, sonucu durum değişkenine kaydet
  if (buton_durum == HIGH) // buton durumu 1 ise
  {
    basla(); // basla adlı fonksiyonu çağır
  }
  else if (buton_durum == LOW) //buton durumu 0 ise
  {
    baslama(); // baslama adlı fonksiyonu çağır
  }
}
```

PARAMETRE ALMAYAN, GERIYE DEĞER DÖNDÜREN FONKSIYON

Bu tip fonksiyonlara parametre gönderilmez ancak fonksiyon geriye değer döndürür. Geriye değer döndüren fonksiyonlarda “void” komutu yerine fonksiyonun geri dönüş türü (int, string, bool vs.) yazılır. Fonksiyonun sonunda ise return komutu ile sonuç geriye döndürülür.

Sözdizimi:

Fonksiyon Tanımlama: `int fonksiyonAdi(){}`

Fonksiyonu çağırma: `int gelen = fonksiyonAdi();`

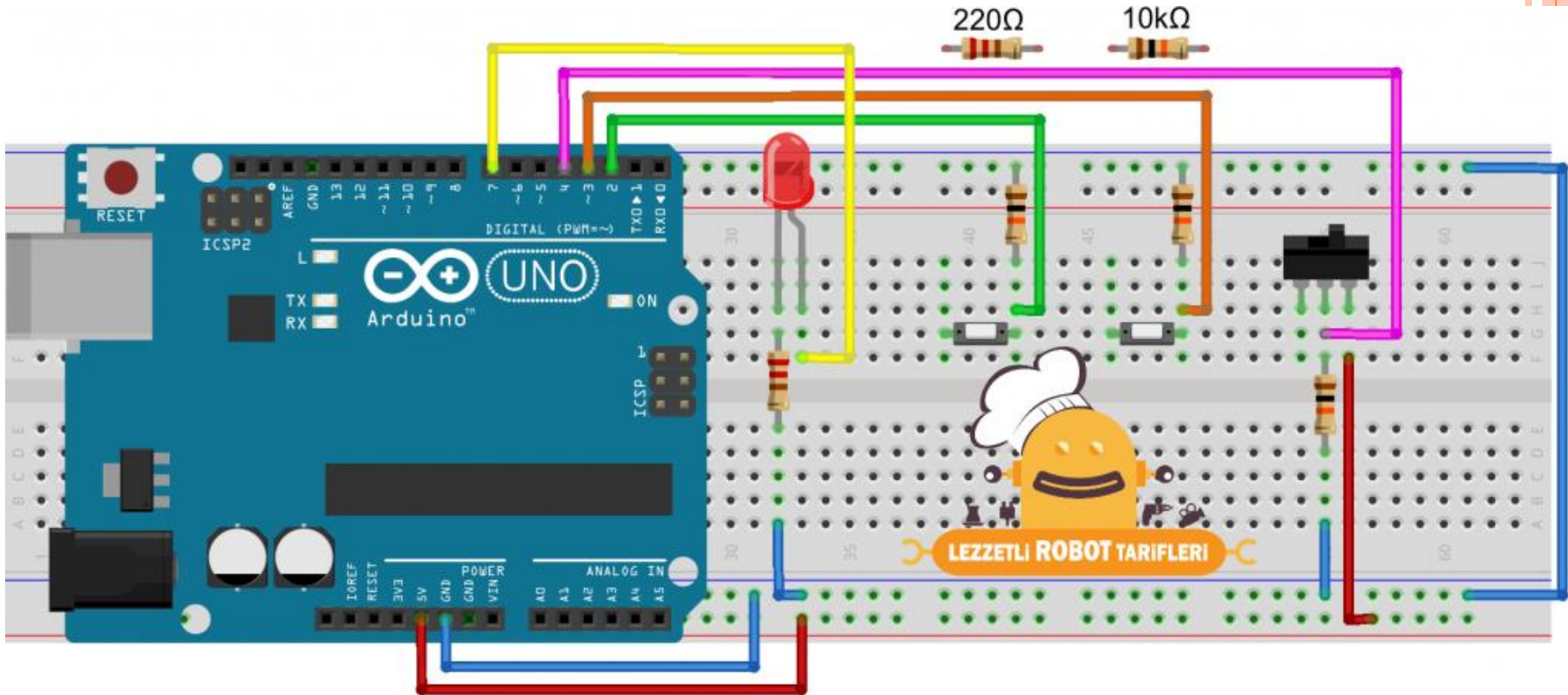


Malzeme Listesi

- 3 adet buton (biri kalıcı tip)
- 1 adet 220ohm direnç
- 3 adet 10K direnç
- LED ve kablolar

22. Parametre Almayan, Geriye Değer Döndüren Fonksiyon

Devre Şeması



Program Kodu

```
const int buton1 = 2;
const int buton2 = 3;
const int buton3 = 4;
const int led = 7;

int durum;
int toplama;
int cikarma;
int tur_sayisi;
int sayac;

int matematikselislem()
{
    durum = digitalRead (buton1);
    while (durum == HIGH)
    {
        toplama = digitalRead (buton2);
        cikarma = digitalRead (buton3);

        if (toplama == HIGH)
```

```
{
    sayac ++;
    Serial.print ("SAYAC=");
    Serial.println (sayac);

    while (1)
    {
        toplama = digitalRead (buton2);
        if (toplama == LOW)
            break;
    }
}

if (cikarma == HIGH)
{
    sayac --;
    Serial.print ("SAYAC=");
    Serial.println (sayac);
```



Program Kodu

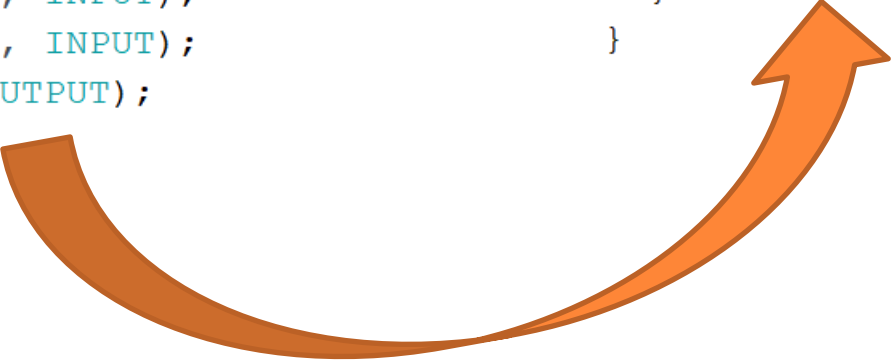
```
while (1)
{
    cikarma = digitalRead (buton3);
    if (cikarma == LOW)
        break;
}

}
durum = digitalRead (buton1);
}
return sayac;
}

void setup()
{
    pinMode (buton1, INPUT);
    pinMode (buton2, INPUT);
    pinMode (buton3, INPUT);
    pinMode (led, OUTPUT);

    Serial.begin(9600);
}

void loop()
{
    tur_sayisi = matematikselislem();
    for (int i = 0; i<tur_sayisi; i++)
    {
        digitalWrite(led, HIGH);
        delay (50);
        digitalWrite(led, LOW);
        delay (50);
    }
    while (1)
    {
        durum = digitalRead (buton1);
        if (durum == HIGH)
            break;
    }
}
```

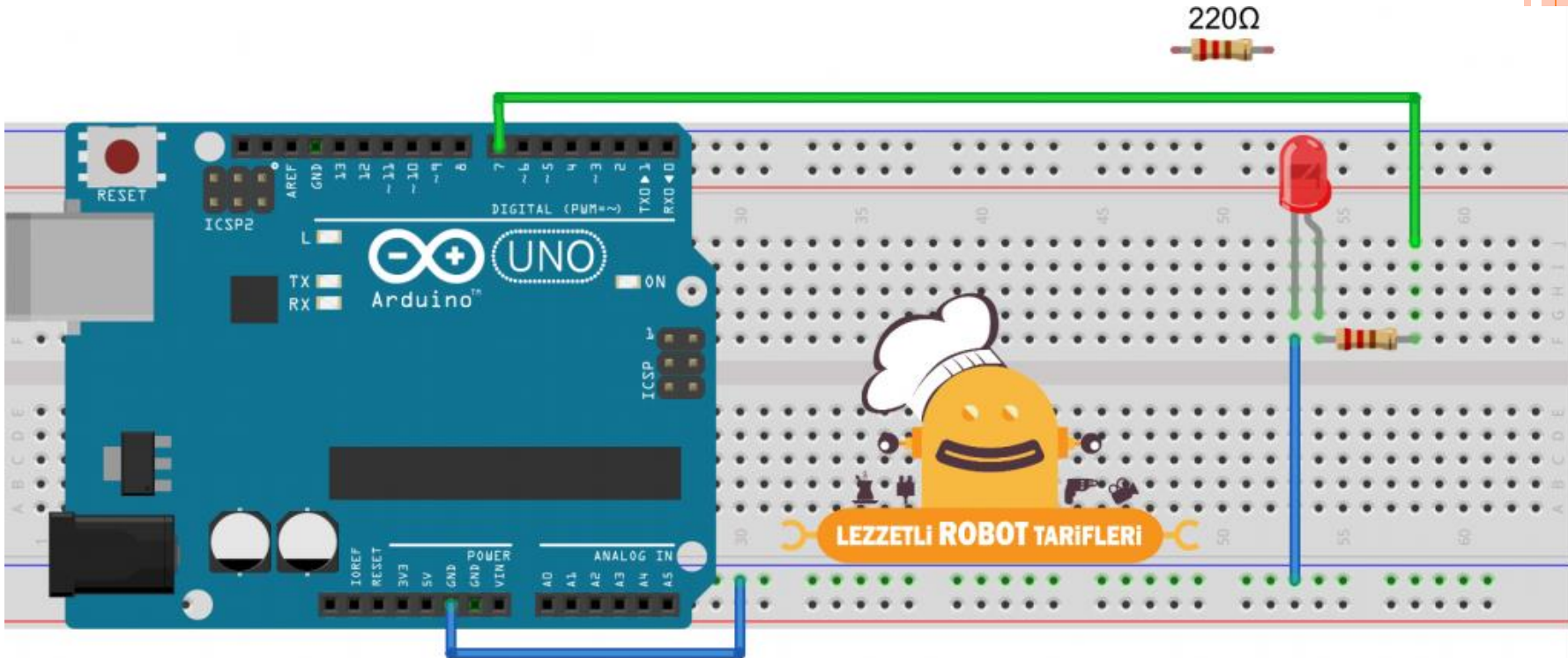


Malzeme Listesi

- 1 adet 220ohm direnç
- LED ve kablolar

23. Parametre Alan, Geriye Değer Döndürmeyen Fonksiyon

Devre Şeması



Program Kodu

```
const int led = 7; // led pini tanıtıldı

int tur_sayisi; // tur sayisi isimli değişke tanımlandı

void fonksiyon3(int a) // fonksiyon3 isimli fonksiyon
{
    for (int i = 0; i < a ; i++)
    {
        digitalWrite(led , HIGH);
        delay(100);
        digitalWrite(led , LOW);
        delay(100);
    }
}

void setup()
{
    Serial.begin(9600); // seri haberleşme başladı
}

void loop()
{
    Serial.println("Algoritma calismaya basliyor");
    delay(2500);
    tur_sayisi = 10; // tur sayisi değişkenine 10 değeri atıldı
    fonksiyon3(tur_sayisi); // tur sayisi değişkeni ile fonksiyon çağırıldı
    Serial.println("Algoritma Bitti");
    while (1);
}
```



Program Kodu 2 (toplama-cikarma)

```
int a ; // kullanılan değişkenler tanımladı
int b ;
int toplam;
int cikartma;

void topla (int h , int g) // toplama isimli fonksiyon
{
    toplam = h + g ;
}

void cikart (int z , int y) // cikart isimli fonksiyon
{
    cikartma = b - a;
}

void setup()
{
    Serial.begin(9600); // ser haberleşme başladı
}

void loop()
{
    a = 5;
    b = 6;
    topla(a, b); // topla fonksiyonu parametre ile çağırıldı
    Serial.print("toplam = ");
    Serial.println(toplam);
    cikart(b, a); // cikart fonksiyonu parametre ile çağırıldı
    Serial.print("sonuc = ");
    Serial.println(cikartma);
    while (1); // sonsuz döngü ile programı kitledik
}
```



23. Parametre Alan, Geriye Değer Döndüren Fonksiyon

Malzeme Listesi

- 1 adet 220ohm direnç
- LED ve kablolar



Program Kodu

```
int toplam;           // Sonuçların kayıt edildiği değişkenlerimiz
int cikarma;
float bolme;
int carpma;

int fonksiyonarti( int a, int b)      // Toplama fonksiyonumuz
{
    int sonuc = a + b ;
    return sonuc;
}
int fonksiyoneksi( int a , int b)      // Çıkartma fonksiyonumuz
{
    int sonuc = a - b ;
    return sonuc;
}
int fonksiyoncarpma( int a , int b)    // Çarpma fonksiyonumuz
{
    int sonuc = a * b;
    return sonuc ;
}
float fonksiyonbolme( float a , float b) // Bölme fonksiyonumuz
{
    float sonuc = a / b;
    return sonuc;
}
```



Program Kodu

```
void setup()
{
  Serial.begin(9600);
}
```

```
void loop()
{
  int a = 8;
  int b = 10;
```

```
  toplam = fonksiyonarti(a, b);      // Toplama fonksiyonumuzu çağırdık ve ilgili
  değişkene kaydettik
  cikarma = fonksiyoneksi(a, b);     // Çıkartma fonksiyonumuzu çağırdık ve ilgili
  değişkene kaydettik
  carpma = fonksiyoncarpma(a, b);    // Çarpma fonksiyonumuzu çağırdık ve ilgili
  değişkene kaydettik
  bolme = fonksiyonbolme(a, b);      // bölme fonksiyonumuzu çağırdık ve ilgili
  değişkene kaydettik
```

```
  Serial.print(" toplam = ");      // Seri monitör de görmek için istediğimiz sonuçları
  yazdırıyoruz
```

```
  Serial.println(toplam);
  Serial.print(" cikarma = ");
  Serial.println(cikarma);
  Serial.print(" carpma = ");
  Serial.println(carpma);
  Serial.print(" Bolme = ");
  Serial.println(bolme);
```

```
  while (1);                        // Yapılan işlemin sadece bir kere yapılması için "While(1)"
  ile döngümüzü kilitledik
}
```



Thank you for listening

