



ARDUİNO PROGRAMLAMADA KULLANILAN KONTROL YAPILARI

Mustafa NUMANOĐLU

'if - else, else - if' Yapısı

- if – else yapısı bir karşılaştırma sonucunda dönen değerın doğru olması durumunda if (eğer) parantezi içerisindeki değerin, yanlış olması durumunda else (değilse) parantezi içerisindeki değerin çalışmasını sağlar. Örnek:

sketch_apr23a \$

```
1 if (i < 4) { // Eğer i değışkeni 4'ten küçükse
2     digitalWrite(2, HIGH); // 2 numaralı dijital çıkışı 1 yap
3 } else { // değilse
4     digitalWrite(2, LOW); // 2 numaralı dijital çıkışı 0 yap
5 }
6
7
```

'if - else, else - if' Yapısı

- else yapısı isteğe bağlı olarak kullanılabilir. Yani else kısmını kullanmadan da kontrol sağlayabiliriz.
- Bu kontrol yapımızda birde else if yapımız var ki aynı yapı üzerinde birden fazla sorgu yaparak çalışabiliriz.

sketch_apr23a §

```
1 if (i < 5) { // i Değişkeni 4'den küçük ise
2     digitalWrite(2, HIGH); // 2 Numaralı dijital çıkışı 1 yap
3
4 } else if (i == 3) { // Değilse i değişkeni 3'e eşitse
5     digitalWrite(3, HIGH); // 3 Numaralı dijital çıkışı 1 yap
6
7 } else { // Yukarıdakilerin hiçbiri değilse
8     digitalWrite(4, HIGH); // 4 Numaralı dijital çıkışı 1 yap
9 }
10
```

'for' Yapısı

- Genellikle artış veya azalış sayacı olarak kullanılır. Bir değerden başka bir değere kadar artıp yada azalarak son değerden sonra döngüyü bitiren operatördür. Örneği incelersek;

```
sketch_apr23a $
```

```
1 for(int m = 1; m < 100; m = m+2)
2 {
3   println(m);
4 }
```

'for' Yapısı

- Bu örneğimizde m değerini birden başlatarak 2 şer artırımla 100 den küçük olana kadar tüm değerlerin çıktısını alıp, seri monitörde görüntülüyoruz.
1,3,5,7,9,.....97,99

'switch case' Yapısı

- 'switch' ifadesi bir nesneye atanan değerleri alır ve alt ifade 'case' operatörüne gönderir. Bu ifade de değerleri karşılaştırır ve denk gelen ifadeleri çalıştırır. Her 'case' ifadesinin sonunda 'break' komutu yer alır. Bu komutla döngü bitirilir. Yani 'case' komutuna yazılan ifadeler sadece bir kez çalışarak 'break' komutu ile sonlanır. Bu sayede algoritmada çakışmalar önlenmiş olur. Son olarak 'default' komutu ile de 'case' lerde tanımlanmamış ifadeler bu tanımlamaya girerek gerekli algoritma yapılandırılır. Örneği inceleyiniz.

'switch case' Yapısı

```
sketch_apr23a §
1 void setup()
2 {
3   pinMode(2, OUTPUT);
4   pinMode(3, OUTPUT);
5   pinMode(4, OUTPUT);
6   pinMode(5, OUTPUT);
7 }
8
9 void loop()
10 {
11   int i = 5;
12   switch (i) // i değişkeninin değerini hafızada tut
13   {
14     case 2: // i değişkeni değeri 2 ise
15       digitalWrite(2, HIGH); // 2 numaralı dijital çıkışı 1 yap
16       break; // kontrolü sonlandır
17     case 3: // i değişkeni değeri 3 ise
18       digitalWrite(3, HIGH); // 3 numaralı dijital çıkışı 1 yap
19       break; // kontrolü sonlandır
20     case 4: // i değişkeni değeri 4 ise
21       digitalWrite(4, HIGH); // 4 numaralı dijital çıkışı 1 yap
22       break; // Kontrolü sonlandır
23     default: // Yukarıdaki şartların hiçbirinde eşitlik sağlanmazsa
24       digitalWrite(5, HIGH); // 5 numaralı dijital çıkışı 1 yap
25       break; // kontrolü sonlandır
26   }
27 }
28
```

'do while' Yapısı

- do - while döngüsünde döngü do komutu ile başlar ve şart kontrol edilmeden döngü parantezine girilir. Parantez içerisindeki işlem yapılır ve döngü sonuna gelindiğinde while komutu ile şart kontrol edilir. Eğer şart hâlâ sağlanıyorsa şart false döndürene kadar işlem tekrarlanır. do while döngüsünde dikkat edilmesi gereken nokta, döngü şartı kontrol etmeksizin bir defa kesinlikle çalışmasıdır. Örneği inceleyiniz.

'do while' Yapısı

```
sketch_apr23a$  
1 void setup()  
2 {  
3   pinMode(2, OUTPUT);  
4   pinMode(3, OUTPUT);  
5   pinMode(4, OUTPUT);  
6   pinMode(5, OUTPUT);  
7  
8   for(int pin = 2; pin <= 5; pin++){  
9     pinMode(pin, OUTPUT);  
10  }  
11 }  
12  
13 void loop() {  
14   int m = 2;    // m Adında değişken tanımlayıp, başlangıç değeri olarak iki değerini atıyoruz  
15  
16   do{          // do while Döngüsüne giriyoruz  
17     digitalWrite(m, HIGH); // m Değişkeninin her döngüdeki değerini dijital pine yönlendirip HIGH olarak atıyoruz  
18     delay(1000);           // 1 Saniyelik bekleme süresi veriyoruz  
19     m++;                   // m Değişkeni değerini bir artırıyoruz  
20   }while(m <= 5);         // Döngü sonunda while komutu ile şartı belirleyip noktalı virgül ile satırı sonlandırıyoruz  
21 }  
22
```

'Continue' Yapısı

- do, while veya for döngülerinde bir aralığı atlamak istiyorsak bu ifadeyi kullanmamız gerekir. Örnek:

```
sketch_apr23a §
1 for (m = 0; m < 255; m ++)\n2 {\n3   if (m > 30 && x < 100) {\n4     continue;\n5   }\n6   analogWrite(sensor, m);\n7   delay(50);\n8 }\n9\n10
```

'Continue' Yapısı

- Bu örnekte m değeri 0 dan 255 kadar birer artarak sensör değerine yazılmaktadır. Fakat 30 ile 100 arasındaki değerleri atlayarak sensöre yazmamaktadır.

'return' Yapısı

- Bir fonksiyonu sonlandırarak yeni fonksiyon çağırarak yeni değerle döndürür. Örnek:

sketch_apr23a §

```
1 double constrain(double x, float a, float b)
2 {
3   if(x < a) return a;
4   if(x > b) return b;
5   return value;
6 }
```

- Burada x değeri a dan küçük ise x 'e a değerini atar. x b den büyük ise x 'e b değerini atar.

'return' Yapısı

sketch_apr23a \$

```
1 void setup() {
2   pinMode(2, INPUT); // 2 numaralı dijital pini giriş olarak ayarladık
3   Serial.begin(9600); // Seri haberleşme hızı 9600 BaudRate
4 }
5
6 int toplama(int sayi1, int sayi2){ // Toplama adında fonksiyon tanımladık
7   int toplam = sayi1 + sayi2;      // Toplam adında tamsayı türü değişken tanımladık ve
8                                     // 10 ile 20 sayısının toplamını değişkene atadık
9   return toplam;
10  }
11
12 void loop() {
13   if(digitalRead(2)){              // Eğer 2 numaralı pindeki butona basılı ise
14     Serial.println(toplama(10, 20)); // Toplama fonksiyonunu çağır ve dönen değeri seri porta yaz
15   }
16 }
17
```

'return' Yapısı

- Yukarıdaki kodda biz “loop” döngüsü içerisinde 2 numaralı pine bağladığımız butona bastığımızda on üçüncü satırdaki “if” sorgusu içerisine girdik, on dördüncü satırda “topla” fonksiyonunu çağırdık ve dönen değeri seri porta yazmasını istedik: `Serial.print(toplama());` .
- Altıncı satırımızda fonksiyon çalışmaya başladı. Fonksiyon kendi içerisinde 10 ve 20 sayısını toplayıp toplam değişkenine atadı. Ardından dokuzuncu satırda “return” komutu ile sonucu bize geri gönderdi.
- Program tekrar ondördüncü satıra gitti ve dönen değeri seri porta yazdırdı.

'goto' Yapısı

- Bu yapı bir döngüyü sona erdirerek başka bir ifadeye atlamayı sağlar. Döngünün içine atlanmasına, bir fonksiyondan diğerine geçilmesine izin vermez ama döngünün dışına atlanabilir. Örnek:

sketch_apr23a §

```
1 for(val m = 0; m < 255; m++){
2   for(val a = 255; a > -1; a--){
3     for(val b = 0; b < 255; b++){
4       if (analogRead(0) > 250){ goto sensor;}
5
6     }
7   }
8 }
9 sensor:
10
```