

Bu aşama, gerçek dünya uygulamalarının ilk adımıdır.

Burada öğrenci artık **mikrodenetleyici** (Arduino) mantığını, sensörleri, aktüatörleri ve temel kodlama prensiplerini öğrenir.

◆ AŞAMA 4: MİKRODENETLEYİCİLER VE GERÇEK SİSTEM SİMÜLASYONU

Oyun / Simülasyon: Tinkercad Circuits

Süre: 3 Hafta (8 Ders)

Amaç: Dijital ve analog verileri mikrodenetleyiciyle okuyup yönetmeyi, temel otomasyon devrelerini kurmayı öğretmek.

Seviye: Orta–İleri (dijital temel bilgisi olan öğrenci)



Genel Akış

Ders	Konu	Ana Odak	Uygulama
1	Mikrodenetleyici Nedir?	Arduino mantığı, giriş/çıkış kavramı	LED yakma devresi
2	Dijital Giriş/Çıkış	Buton okuma, LED kontrol	Butonla LED yak/söndür
3	Analog Girişler	Sensör verisi (LDR, potansiyometre)	Parlaklığa LED kontrolü
4	PWM ve Analog Çıkış	LED parlaklığa kontrolü	Fade efekti
5	Zamanlayıcılar ve Döngüler	delay(), millis()	Yanıp sönen LED ve sayaç
6	Sensör + Eyleyici	LDR, servo, buzzer kombinasyonu	İşığa göre hareket eden sistem
7	Seri Haberleşme	Serial Monitor kullanımı	Sensör verisini ekranda gösterme
8	Mini Proje: Akıllı Oda Kontrol Sistemi	Tüm sistemlerin entegrasyonu	Otomatik ışık + fan + alarm sistemi

DERS 1 — Mikrodenetleyici Nedir?

Kazanım:

Öğrenci, Arduino'nun dijital bir "beyin" olduğunu ve giriş–çıkışları yönettiğini anlar.

Konu Özeti:

- Mikrodenetleyici, sensörlerden veri alıp karar verir.
- Arduino UNO: 14 dijital, 6 analog pin içerir.
- Kod → Davranış: "donanım + yazılım" birleşimi.

Uygulama (Tinkercad Circuits):

1. Arduino + 1 LED + 220Ω direnç bağla.
2. Kodu yaz:

```
void setup() { pinMode(13, OUTPUT); }
void loop() { digitalWrite(13, HIGH); delay(500); digitalWrite(13, LOW); delay(500); }
```

1. Simülasyonu başlat → LED'in yanıp sönmesini gözlemle.

Değerlendirme:

- Kodun hangi satırı LED'i yakıyor?
- `delay()` ne işe yarıyor?

DERS 2 — Dijital Giriş / Çıkış

Kazanım:

Buton veya anahtar ile devreyi kontrol etmeyi öğrenir.

Konu Özeti:

- `pinMode(pin, INPUT)` → veriyi okur.
- `digitalRead()` ile buton durumunu alır.
- `if` koşuluyla tepki verir.

Uygulama:

1. Arduino + buton + LED bağla.

2. Kod yaz:

```
void setup() {  
    pinMode(2, INPUT);  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    if (digitalRead(2) == HIGH)  
        digitalWrite(13, HIGH);  
    else  
        digitalWrite(13, LOW);  
}
```

1. Butona bas → LED yanmalı.

Değerlendirme:

- Butona basılmadığında neden LED sönük?
- Dijital giriş “1” neyi temsil eder?

DERS 3 — Analog Girişler

Kazanım:

Analog sensörleri okumayı öğrenir.

Konu Özeti:

- `analogRead()` → 0–1023 arasında değer döndürür.
- LDR, potansiyometre gibi sensörler analog veri üretir.

Uygulama:

1. Arduino + LDR + LED devresi kur.
2. Kod yaz:

```
int sensor = A0;  
int led = 9;  
void setup() { pinMode(led, OUTPUT); }  
void loop() {  
    int value = analogRead(sensor);  
    if (value < 500) digitalWrite(led, HIGH);  
    else digitalWrite(led, LOW);  
}
```

1. Ortam ışığını değiştir → LED'in tepkisini izle.

Değerlendirme:

- LDR değerleri neye göre değişiyor?
- Analog veri neyi temsil ediyor?

DERS 4 — PWM ve Analog Çıkış

Kazanım:

Analog sinyal benzetimiyle parlaklık veya hız kontrolü yapmayı öğrenir.

Konu Özeti:

- PWM (Pulse Width Modulation): dijital pin ile “analog benzeri” çıkış üretir.
- `analogWrite(pin, değer)` → 0–255 arası sinyal.

Uygulama:

1. LED’i PWM destekli pine (9, 10, 11) bağla.
2. Kod:

```
void setup() { pinMode(9, OUTPUT); }
void loop() {
    for (int i = 0; i < 255; i++) {
        analogWrite(9, i);
        delay(10);
    }
    for (int i = 255; i > 0; i--) {
        analogWrite(9, i);
        delay(10);
    }
}
```

LED’in yavaşça parlayıp sönmesini izle.

Değerlendirme:

PWM hangi tür sinyali taklit ediyor?

`analogWrite()` ile `digitalWrite()` farkı nedir?

DERS 5 — Zamanlayıcılar ve Döngüler

Kazanım:

Kodda zaman yönetimi ve olay sırası kurmayı öğrenir.

Konu Özeti:

- `delay()` işlemi durdurur, `millis()` zamanı ölçer.
- Zaman bazlı sistemlerde kontrol akışı önemlidir.

Uygulama:

1. 2 LED kur (kırmızı–yeşil).
2. Kodla dönüşümlü yak:

```
void setup() { pinMode(8, OUTPUT); pinMode(9, OUTPUT); }
void loop() {
    digitalWrite(8, HIGH); digitalWrite(9, LOW);
    delay(1000);
    digitalWrite(8, LOW); digitalWrite(9, HIGH);
    delay(1000);
}
```

1. LED'ler sırayla yanmalı.

Değerlendirme:

- `delay()` süresini artırınca ne olur?
- `millis()` kullanmak ne işe yarar?

DERS 6 — Sensör + Eyleyici Sistemi

Kazanım:

Sensörden veri okuyup motor, servo veya buzzer gibi eyleyicileri kontrol eder.

Konu Özeti:

- Servo veya buzzer analog kontrolle çalışır.
- Giriş–çıkış ilişkisi artık otomasyona dönüşür.

Uygulama:

1. Arduino + LDR + servo bağla.
2. Kod:

```
#include <Servo.h>
Servo s;
void setup() { s.attach(9); }
void loop() {
    int light = analogRead(A0);
    int angle = map(light, 0, 1023, 0, 180);
    s.write(angle);
    delay(50);
}
```

1. Işık değişikçe servo hareket eder.

Değerlendirme:

- map() fonksiyonu ne işe yarıyor?
- Servo neden analog kontrol gerektiriyor?

DERS 7 — Seri Haberleşme

Kazanım:

Arduino'nun bilgisayar veya başka cihazlarla veri alışverişini öğrenir.

Konu Özeti:

- `Serial.begin(9600)` haberleşmeyi başlatır.
- `Serial.print()` veri gönderir.
- Bu sistem hata ayıklama ve izleme için kullanılır.

Uygulama:

1. LDR sensörü bağla.
2. Kod:

```
void setup() { Serial.begin(9600); }
void loop() {
    int val = analogRead(A0);
    Serial.println(val);
    delay(500);
}
```

1. Serial Monitor'da değerleri gözlemle.

Değerlendirme:

- Veriler hangi aralıkta geliyor?
- Bu verileri nasıl grafikleştirebiliriz?

DERS 8 — Mini Proje: Akıllı Oda Kontrol Sistemi

Kazanım:

Sensörlerden gelen veriyi işleyip çoklu karar sistemi kurar.

Konu Özeti:

- LDR (ışık sensörü) → ışık kontrolü
- DHT11 (sıcaklık sensörü) → fan kontrolü
- Buzzer → alarm fonksiyonu

Uygulama (Tinkercad):

1. Arduino + LDR + DHT11 + LED + fan (DC motor) + buzzer bağla.
2. Kod:

```
#include <DHT.h>
DHT dht(2, DHT11);
int led = 9, fan = 10, buzz = 11;
void setup() {
    pinMode(led, OUTPUT); pinMode(fan, OUTPUT); pinMode(buzz, OUTPUT);
    dht.begin(); Serial.begin(9600);
}
void loop() {
    float temp = dht.readTemperature();
    int light = analogRead(A0);
    if (light < 400) digitalWrite(led, HIGH); else digitalWrite(led, LOW);
    if (temp > 28) digitalWrite(fan, HIGH); else digitalWrite(fan, LOW);
    if (temp > 35) digitalWrite(buzz, HIGH); else digitalWrite(buzz, LOW);
    delay(1000);
}
```

1. Farklı koşullarda sistemi test et.

Değerlendirme:

- Hangi sensör hangi eyleyiciyi tetikliyor?
- Sistem hangi koşullarda aktif oluyor?

Aşama 4 Sonu – Değerlendirme

Öğrenci artık:

- Elektronik + yazılım birleşimini kavradı,
- Sensör–eyleyici mantığını anladı,
- Temel otomasyon sistemlerini kurabiliyor.

Artık sıradaki adım:

 **Aşama 5 — PCB Tasarımı, Elektronik Proje Entegrasyonu (KiCad / Fritzing)**
Yani artık sanal projeyi **gerçek devre kartına dönüştüreceğiz**.