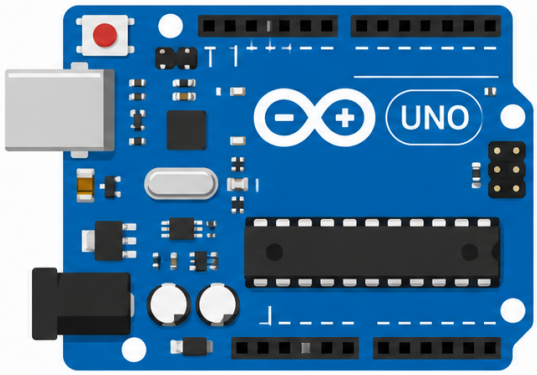
DONANIM GELİŞTİRME ATÖLYESİ

Öğren • Tasarla • Üret



Donanım Geliştirme Atölyesi

ELEKTRONİK / ROBOTİK / YAZILIM

ÜCRETSİZ REHBER

Başlangıçtan ilk çalışan projeye

15 sayfalık kaynak rehberi

2026



Mustafa Kemal Atatürk

1881 - 1938

Türkiye Cumhuriyeti’nin kurucusu

Fotoğraf: Ataturk1930s.jpg, Wikimedia Commons. Fotoğrafçı bilinmiyor.
Dosya sayfasında kamu malı olarak belirtilmiştir. Ön sayfa I.

GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen; Türk istiklalini, Türk cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi seni bu hazineden mahrum etmek isteyecek dâhilî ve haricî bedhahların olacaktır. Bir gün, istiklal ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin. Bu imkân ve şerait, çok namüsait bir mahiyette tezahür edebilir.

İstiklal ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elim ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar, gaflet ve dalalet ve hatta hıyanet içinde bulunabilirler.

Hatta bu iktidar sahipleri, şahsi menfaatlerini müstevlilerin siyasi emelleriyle tevhit edebilirler. Millet, fakruzaruret içinde harap ve bitap düşmüş olabilir.

Ey Türk istikbalinin evladı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklal ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk

1927

Metin: T.C. Gençlik ve Spor Bakanlığı, “Atatürk’ün Gençliğe Hitabesi”.

Resmî yayımın yazımı korunmuştur. Bu sayfa, 15 sayfalık rehbere dahil değildir.

01

ÖĞRENME ROTASI

Nereden başlamalı?

Bu rehber, bir projeyi yalnızca çalıştırmayı değil; bağlantısını, kodunu ve sonucunu açıklayabilmeyi hedefler. Başlangıç ve orta seviye öğrenciler için hazırlanmıştır.

Rehberi kullanmanın üç adımı

Önce oku: Bölümdeki kavramları ve üretici belgesini incele. **Sonra uygula:** Küçük deneyi tek bir değişkeni değiştirerek yap. **Son olarak kaydet:** Bağlantı tablosunu, kod sürümünü ve ölçüm sonucunu proje günlüğüne ekle.

Sayfa	Konu	Sayfa	Konu
02	Elektriğin temel dili	09	Motorlar ve robotik
03	Bileşen ve veri sayfası	10	Güç, pil ve regülatör
04	Ölçüm ve çalışma düzeni	11	Wi-Fi ve IoT
05	Arduino ile ilk adım	12	Web, C# ve veritabanı
06	Programlama mantığı	13	Simülasyon ve PCB
07	ESP32 ve kart seçimi	14	Dört haftalık proje planı
08	Sensörler ve ekranlar	15	Hata giderme ve dizin

Hangi yoldan ilerlemelisin?

İlk defa başlıyorsan: 02 → 04 → 05 → 06 sırasını izle; önce LED ve butonla çalış. **Arduino deneyimin varsa:** 07 → 08 → 10 → 11 yolunu seç. **Robot yapmak istiyorsan:** 09 ve 10’u birlikte oku; sürücü ve güç hesabını koddan önce tamamla.

> **Her proje için bir kanıt üret**
> “Çalıştı” demek yerine kısa video, seri port kaydı, ölçüm tablosu ve açıklamalı devre şeması sakla. Bölüm sonundaki görevler rehber için hazırlanmış öğrenme önerileridir; üretici sertifikası değildir.

OKU • UYGULA • DOĞRULA

Arduino Learn | EN • Temel kavramlar ve konu başlıkları.

Pro Git | TR/EN • Değişiklikleri kaydetmek için başlangıç bölümleri.

02

TEMEL ELEKTRONIK

Elektriğin temel dili

Gerilim, akım ve direnç arasındaki ilişkiyi anlamak; LED’den motor beslemesine kadar bütün projelerin temelidir.

Büyüklük	Birim	Projede ne anlatır?
Gerilim (V)	Volt, V	İki nokta arasındaki elektriksel potansiyel farkı.
Akım (I)	Amper, A	Bir iletkenden geçen elektrik yükünün akış hızı.
Direnç (R)	Ohm, Ω	Akımın geçişine gösterilen direnç.
Güç (P)	Watt, W	Birim zamanda aktarılan elektrik enerjisi.

İki temel bağıntı

V = I × R direnç üzerindeki ilişkiyi, **P = V × I** elektriksel gücü ifade eder. 10 mA, 0,010 A’dır. Hesaba başlamadan önce miliamperi ampere dönüştür; en sık yapılan hatalardan biri birimleri karıştırmaktır.

Örnek: kırmızı LED için seri direnç

5 V besleme, yaklaşık 2 V LED gerilim düşümü ve 5 mA hedef akım varsayalım. R = (5 − 2) / 0,005 = **600 Ω**. Standart bir üst değer olan **680 Ω** ile yaklaşık 4,4 mA elde edilir. LED’in gerçek ileri gerilimi veri sayfasından kontrol edilir; bu hesap bütün LED’ler için aynı sonuç vermez.

Dirençte harcanan güç yaklaşık (5 − 2) × 0,0044 = **0,013 W** olur. Bu örnekte 0,25 W direnç yeterli pay bırakır. Direnç, LED ile seri olmak şartıyla LED’in önünde veya arkasında yer alabilir.

> **Mini görev**
> Aynı LED’i 3,3 V ile beslemek için 5 mA hedefinde gereken direnci hesapla. Sonra 330 Ω seçildiğinde akımı bul. Beklenen yaklaşık sonuç: 260 Ω hesap değeri; 330 Ω ile 3,9 mA. Sonucu gerçek ölçümle karşılaştır.

OKU • UYGULA • DOĞRULA

SparkFun: Ohm Yasası | EN • Akım, gerilim, direnç ve LED hesabı.

03

BILEŞEN OKURYAZARLIĞI

Bileşeni doğru tanı

Aynı görünümlü iki modül farklı gerilim, pin sırası veya haberleşme yöntemi kullanabilir. Ürün fotoğrafı tek başına bağlantı belgesi değildir.

Bileşen	Görevi	Bağlamadan önce
Direnç / potansiyometre	Akımı sınırlar / ayarlanabilir bölücü kurar.	Değerini, gücünü ve bacak düzenini kontrol et.
Kondansatör	Enerji depolar; besleme dalgalanmasını azaltır.	Kapasite, gerilim sınırı ve varsa kutup yönü.
Diyot / LED	Tek yönde iletim / ışık üretimi.	Anot-katot, ileri akım ve gerilim.
Transistör / MOSFET	Yükü bir kontrol sinyaliyle anahtarlar.	Bacak dizilimi, akım ve sürme gerilimi.
Sensör modülü	Fiziksel büyüklüğü elektriksel veriye çevirir.	Besleme, çıkış seviyesi, ölçüm aralığı.

Veri sayfasını hangi sırayla okumalı?

1. Parça kodu: Üzerindeki tam kodu, son ekleriyle kaydet. **2. Pin açıklamaları:** Kart üzerindeki isimlerle eşleştir. **3. Önerilen çalışma koşulları:** Tasarımı bu aralıkta tut. **4. Tipik uygulama:** Gerekli ek direnç, kondansatör ve korumaları belirle.

Absolute maximum ratings, normal çalıştırma hedefi değildir; aşılmaması gereken sınırları gösterir. Ayrıca bir entegreye ait akım değeri, o entegreyi taşıyan ucuz modülün aynı akımı sürekli vereceği anlamına gelmez. PCB, soğutma ve bağlantılar da sınırlayıcıdır.

> **Kaynak okuma çalışması**
> DRV8833 belgesinde besleme aralığını, giriş mantığını ve termal koşulları bul. Bir sayfalık “parça kartı” hazırla: tam model, görev, pinler, çalışma koşulları, veri sayfası bağlantısı ve belgenin sürümü.

OKU • UYGULA • DOĞRULA

TI: DRV8833 ürün ve veri sayfası | EN • Pin tablosu ve çalışma koşullarını okuma örneği.

Arduino UNO R3 belgeleri | EN • Kart şeması, pinout ve veri sayfasını karşılaştır.

04

ATÖLYE PRATİĞİ

Ölçüm ve çalışma düzeni

Devrenin ne yaptığını anlamanın en güvenilir yolu ölçmektir. Başlangıç uygulamalarını düşük gerilimli, yalıtılmış DC kaynaklarla yap.

İşlem	Doğru yöntem	Yaygın hata
DC gerilim	Siyah prob COM, kırmızı prob V girişinde; paralel ölçüm.	Kırmızı prob A girişindeyken beslemeye dokunmak.
Direnç / süreklilik	Enerji kesilmiş ve kondansatörleri boşalmış devrede.	Enerjili devrede direnç ölçmek.
Akım	Uygun sigortalı giriş ve aralıkla devreye seri bağlamak.	Akım modunda kaynağın iki ucuna paralel bağlamak.

İlk enerji verme kontrolü

• Gücü kes; + ve GND hatlarını şemayla karşılaştır. Breadboard güç raylarının ortadan bölünüp bölünmediğini kontrol et.

• Besleme çıkışını karta bağlamadan ölç. Kablolarda açıkta iletken, ters kutup ve lehim köprüsü bulunmadığından emin ol.

• Varsa akım sınırlamalı kaynak kullan. Beklenmeyen ısınma, koku veya yüksek akımda beslemeyi kes; arızayı enerjisiz incele.

Lehim ve masa düzeni

Havyayı standında tut; göz koruması ve dumanın çalışma alanından uzaklaştırıldığı havalandırma kullan. Lehim sonrası ellerini yıka. Motor veya pil gibi yüksek akım yollarını breadboard üzerinden geçirme; uygun kablo ve bağlantı kullan.

> **Ölçüm defteri**
> Bir 5 V USB beslemenin çıkışını, ardından LED’in iki ucunu ölç. Tarih, ölçüm noktası, cihaz aralığı ve sonucu yaz. Şebeke prizleri ve açık şebeke devreleri bu başlangıç rehberinin kapsamı dışındadır.

OKU • UYGULA • DOĞRULA

Fluke: DC gerilim ölçümü | EN • Prob girişi, ölçüm kademesi ve bağlantı sırası.

05

ARDUINO

Arduino ile ilk adım

Önce bilgisayar ile kart arasındaki bağlantıyı doğrula. Sensör, motor ve ekranı aynı anda eklemek hata kaynağını bulmayı zorlaştırır.

Örnek kart: UNO R3

UNO R3, ATmega328P tabanlıdır; 14 dijital giriş/çıkış, bunların 6'sında PWM ve 6 analog giriş sunar. Mantık seviyesi 5 V'tur. Bu bilgiler **UNO R3** içindir; UNO R4, Nano ve diğer aile üyelerinin özellikleri ayrıca incelenmelidir.

Aşama	Yapılacak iş	Başarı göstergesi
1. Bağlan	Veri aktarabilen USB kablosu kullan.	Kart işletim sisteminde görünür.
2. Seç	IDE içinde doğru kartı ve portu seç.	Yükleme hedefi kartın modeliyle eşleşir.
3. Yükle	Yerleşik Blink örneğini yükle.	Yerleşik LED düzenli yanıp söner.
4. Gözle	AnalogReadSerial örneğini incele.	Seri monitörde değerler okunur.

İlk harici devre

Enerji kesikken D8 → 680 Ω direnç → LED anodu; LED katodu → GND bağlantısını kur. Örnekte pin sabitini 8 yap. Bu, tek LED için başlangıç denemesidir; GPIO pinini motor, güçlü LED veya hoparlör beslemesi gibi kullanma.

Kartın üzerinde yazan **5V**, **3.3V**, **VIN** ve **GND** birbirinin yerine kullanılamaz. Başlangıçta USB ile besle; harici beslemeye geçerken kartın güç şemasını oku ve USB ile dış kaynağın birlikte kullanım koşullarını doğrula.

> **Mini görev**
> LED'in açık ve kapalı kalma sürelerini değiştir. Daha sonra buton ekle ve bir bağlantı tablosu hazırla. Fotoğrafın yanında pin isimlerinin yazılması, projeyi başka birinin tekrar kurmasını kolaylaştırır.

OKU • UYGULA • DOĞRULA

Arduino UNO R3 | EN • Kart özellikleri ve pinout.

Arduino Built-in Examples | EN • Blink, Digital Read Serial ve Analog Read Serial.

06

PROGRAMLAMA

Kodun mantığını kur

Bir örneği kopyaladıktan sonra girişini, kararını ve çıkışını kendi cümlelerinle açıklayabiliyorsan öğrenme kalıcı hâle gelir.

setup() başlangıç ayarlarını, **loop()** tekrar eden işleri içerir. Değişkenler değer saklar; koşullar karar verir; fonksiyonlar tekrar kullanılan işleri düzenler. Pin numaraları ve eşikler için anlamlı sabitler kullan.

Örnek: UNO R3 üzerinde beklemeden LED kontrolü

```
const byte LED_PIN = 8;
unsigned long onceki = 0;
bool durum = false;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  unsigned long simdi = millis();
  if (simdi - onceki >= 500UL) {
    onceki = simdi;
    durum = !durum;
    digitalWrite(LED_PIN, durum);
  }
}
```

LED bağlantısı önceki sayfadaki gibidir. Kod, her 500 ms’de durumu tersine çevirir; tam yanıp sönme çevrimi yaklaşık 1 saniyedir. Aradaki zamanda buton okumak gibi başka işler yapılabilir. Bu öğretici örnek, fiziksel donanım üzerinde test edildiği iddiasını taşımaz.

Dört kontrol sorusu

Değişkenin türü doğru mu? Giriş pini boşta mı kalıyor? Seri monitör hızı kodla aynı mı? Uzun bir bekleme, başka bir işi engelliyor mu? Buton için **INPUT_PULLUP** kullanıldığında GND’ye basılan buton LOW okunur; mantığı buna göre kur.

OKU • UYGULA • DOĞRULA

Arduino Language Reference | EN • Fonksiyonun parametresini ve dönüşünü oku.

Blink Without Delay | EN • Digital bölümündeki zamanlama örneğini incele.

07 ESP32

ESP32: doğru kart, doğru pin

ESP32 bir kart adı olarak kullanılsa da farklı işlemci aileleri ve kart düzenleri vardır. Bağlantı tablosu tam modele göre hazırlanmalıdır.

Başlangıç sırası

Arduino IDE'ye Espressif'in kart paketini kur; kart modelini ve portunu seç. İlk denemede yalnızca seri porta kısa bir mesaj yazdır. Ardından GPIO, sensör ve son olarak Wi-Fi özelliğini ekle. Paket ve kütüphane sürümlerini proje notuna yaz.

Karar	Neyi kontrol etmelisin?
Kart adı	WROOM tabanlı DevKit, C3, S3 ve CAM kartlarının pin düzenlerini karıştırma.
GPIO seçimi	Kart şemasından erişilebilir pinleri, açılış görevlerini ve ayrılmış hatları kontrol et.
Mantık seviyesi	ESP32 GPIO'larını 3,3 V mantık olarak ele al; doğrudan 5 V sinyal uygulama.
Kablosuz özellik	Wi-Fi ve Bluetooth türünü, seçtiğin işlemci ailesinin belgesinden doğrula.

Sürüm farkını tanımak

Eski eğitimlerde kullanılan bazı fonksiyonlar yeni paketlerde değişmiş olabilir. Özellikle Arduino-ESP32 2.x ile 3.x arasında API değişiklikleri vardır. "Tanımlanmadı" hatasında rastgele kütüphane eklemek yerine kurulu sürümün belgesine ve geçiş rehberine bak.

PWM için **LEDC** belgesini kullan. Kanal sayısı ve özellikler işlemci ailesine göre değişebilir. Pin etiketinin D21 olması ile GPIO21 anlamına gelmesi de kart üreticisinin isimlendirmesine bağlıdır; tam pinout çizimi belirleyicidir.

> **Mini görev**
> Kendi kartın için 5 satırlık kimlik hazırla: işlemci ailesi, kart adı, paket sürümü, seçilen iki GPIO ve besleme yöntemi. Bir sonraki projede bu kimliği README'ye ekle.

OKU • UYGULA • DOĞRULA

Espressif: Getting Started | EN • Kurulum ve desteklenen aileler.

Espressif: 2.x → 3.0 geçişi | EN • Eski örneklerin uyumsuzluklarını kontrol et.

08

VERI TOPLAMA

Sensörler ve ekranlar

Sensör verisi, bir sayıdan fazlasıdır: birimi, ölçüm aralığı, hata payı ve güncellenme sıklığı ile anlam kazanır.

Arayüz	Temel hatlar	Tipik kullanım
Analog	Sinyal + besleme + GND	Potansiyometre veya analog sensör çıkışı.
I2C	SDA, SCL + besleme + GND	OLED, saat modülü, bazı çevre sensörleri.
SPI	Saat, veri ve cihaz seçme hatları	TFT ekran, SD kart, hızlı çevre birimleri.
UART	TX, RX ve ortak referans	Seri modüller ve cihazlar arası iletişim.

I2C ekranı çalıştırma yöntemi

Önce ekranın denetleyicisini belirle: SSD1306 ve SH1106 aynı kütüphane ayarıyla her zaman çalışmaz. Besleme ve I2C mantık seviyesini doğrula. Ardından adres taraması yap, bulunan adresi kaydet ve kütüphanenin en basit örneğini çalıştır.

Generic ESP32 için SDA GPIO21 ve SCL GPIO22 yaygın varsayılandır; diğer kartlarda değişebilir. I2C hatlarında pull-up dirençleri gerekir. Modül üzerindeki dirençlerin hangi gerilime bağlı olduğunu kontrol et; 5 V’a çekilen bir hat 3,3 V GPIO için uygun olmayabilir.

Ham veri ile karar arasına kontrol ekle

Ölçüme birim ekle; geçersiz veya eksik veriyi “0” gibi gösterme. Eşik yakınında sürekli açılıp kapanmayı önlemek için histerezis kullan: örneğin göstergeyi 30 üzerinde aç, 28 altında kapat. Bu değerler öğrenme örneğidir; gerçek uygulama gereksinimine göre belirlenir.

> **Mini görev**
> 10 sensör okumasını zaman bilgisiyle kaydet. En küçük, en büyük ve ortalama değeri hesapla. Ekranda değerin yanında birimi ve son güncelleme durumunu göster; sensör söküldüğünde hata mesajı üret.

OKU • UYGULA • DOĞRULA

Espressif: I2C API | EN • Pin seçimi, pull-up ve veri aktarımı.

Adafruit: Arduino Libraries | EN • Kütüphane kurulumu ve örnek kullanımı.

09

HAREKET KONTROLÜ

Motorlar ve robotik

Motorun dönüşünü kod belirler; güvenilir çalışmasını ise sürücü seçimi, besleme ve mekanik yük birlikte belirler.

Aktüatör	Kontrol yaklaşımı	Önemli seçim ölçütü
DC motor	Transistör veya H-köprü; hız için PWM.	Besleme gerilimi, kalkış ve kilitlenme akımı.
Hobi servosu	Uygun darbe sinyali; ayrı güç planı.	Tork, hareket aralığı ve yükteki akım.
Adım motoru	Motor tipine uygun adım sürücüsü.	Faz akımı, sürücü ayarı ve mekanik yük.

Sürücü seçerken üç soru

1. Motor gerilimi sürücünün çalışma aralığında mı? **2.** Motorun kalkış ve sıkışma akımı sürücünün sınırlarını zorluyor mu? **3.** Kontrol girişleri kartın 3,3 V veya 5 V mantığıyla uyumlu mu? “2 A” gibi bir başlık tek başına sürekli çalışma garantisi vermez; termal koşulları da oku.

Motoru GPIO’dan doğrudan besleme. Kontrol kartı ile sürücünün sinyal referansını, kullanılan devreye uygun biçimde ortaklaştır. Motor beslemesindeki darbelerin kartı yeniden başlatmasını önlemek için güç yollarını düzenle; sürücünün önerdiği kondansatörleri kullan.

İlk robot testi

- Tekerlekleri zeminden kaldır; düşük hızda ileri, dur, geri komutlarını ayrı ayrı dene.
- Hareket yönünü kablo rengine göre tahmin etme; her motoru etiketle.
- Kablosuz araçta komut kesilince durma davranışı ekle; kesinti süresini ve durmayı ölç.

> **Mini görev**
> Bir motor için gerilim, yüksüz akım, hedef yük ve sürücü modelini tabloya yaz. Kilitlenme akımını uzun süre motoru sıkıştırarak ölçme; üretici verisini kullan veya kontrollü ölçümü yetkin gözetimle planla.

OKU • UYGULA • DOĞRULA

Adafruit: DC Motors | EN • Transistörle motor kontrolünün temeli.

TI: DRV8833 | EN • H-köprü, akım ve termal koşullar.

10

ENERJI YÖNETIMI

Güç, pil ve regülatör

Çalışma süresini tahmin etmeden önce devrenin güç ihtiyacını belirle. Pil kapasitesi tek başına yeterli bir tasarım ölçütü değildir.

Güç bütçesini çıkar

Kartın, ekranın, sensörlerin ve motorların tüketimini ayrı yaz. Ortalama tüketim çalışma süresini etkiler; anlık tepe akımı ise kaynağın çöküp çökmeyeceğini belirler. Regülatör kayıplarını ve kablo üzerindeki gerilim düşümünü hesaba kat.

Dönüştürücü	İşi	Kullanım notu
Buck	Gerilimi düşürür.	Giriş gerilimi ve gerekli pay çıkıştan yeterince yüksek olmalı.
Boost	Gerilimi yükseltir.	Çıkış gücü arttıkça pil tarafından çekilen akım artar.
Buck-boost	Gerilimi yükseltir veya düşürür.	Pil gerilimi hedef çıkışın iki tarafına geçiyorsa değerlendir.

Örnek çalışma süresi hesabı

Nominal 3,7 V, 2.000 mAh pil yaklaşık **7,4 Wh** enerji taşır. Kullanılabilir enerjiyi ve dönüşüm kayıplarını birlikte yaklaşık %85 kabul edersek, 2 W ortalama yük için süre: 7,4 × 0,85 / 2 ≈ **3,1 saat**. Bu bir tahmindir; sıcaklık, hücre durumu, kesme gerilimi ve yük davranışı sonucu değiştirir.

Şarj devresi ile koruma devresi farklıdır

Şarj kontrolü hücreye uygun akım ve gerilimi yönetir; koruma/BMS ise belirli arıza durumlarını sınırlar. Biri diğerinin otomatik yerine geçmez. Seri bağlı hücreler, seri sayısına uygun şarj ve koruma düzeni ister. Tek hücrelik şarj modüllerini seri pakete rastgele bağlama.

> **Uygulama sınırı**
> Başlangıçta hazır, korumalı güç çözümü kullan. Hasarlı veya şişmiş pili kullanma. Ayarlı regülatörün çıkışını yükü bağlamadan ölç; modül üzerindeki akım iddiasını soğutma ve sürekli yük koşullarıyla doğrula.

OKU • UYGULA • DOĞRULA

Adafruit: Li-Ion & LiPoly Batteries | EN • Hücre, şarj ve kullanım esasları.

TI: LM2596 | EN • Buck regülatörün üretici veri sayfası.

11

BAĞLANTILI PROJELER

Wi-Fi ve IoT temelleri

Bir cihazın ağa bağlanması, projenin tamamlandığı anlamına gelmez. Bağlantı koptuğunda ve veri eskidiğinde ne olacağını da tasarlamalısın.

Önce yerel ağda çalıştır

STA modunda ESP32 bir erişim noktasına bağlanır. **AP modunda** ise başka cihazların bağlanacağı ağı oluşturur. İlk çalışmada yerel ağda bir durum sayfası aç; internet hizmeti, hesap ve uzaktan erişim gibi katmanları daha sonra ekle.

Yöntem	Ne zaman işe yarar?	İlk deneme
HTTP	İstek gönderip yanıt almak; web arayüzü.	Tarayıcıda bir sensör değerini göster.
MQTT	Konu başlıkları üzerinden mesaj yayınlama/abonelik.	Bir yerel broker üzerinden değer yayınla.
Yerel ekran	Ağ olmasa da kullanıcıya durum göstermek.	Bağlı / bağlantı yok / veri eski durumları.

Veri biçimini küçük ve açık tut

```
{"device":"oda-01","temperature_c":24.6,
 "sample_id":128,"status":"ok"}
```

Cihaz adı, birim ve kayıt sırası hataları ayırmayı kolaylaştırır. Ölçüm zamanı ekliyorsan cihaz saatinin doğru olup olmadığını kontrol et.

Bağlantı kesintisine hazırlık

Yeniden bağlanma denemeleri arasında bekleme bırak; ana döngüyü sonsuza kadar kilitleme. Parola ve anahtarları herkese açık kod deposuna koyma. İnternete açılan hizmetlerde kimlik doğrulama ve şifreli taşıma kullan; sertifika hatalarını doğrulamayı kapatarak çözme.

> **Mini görev**
> Wi-Fi'yi kapatıp aç; yeniden bağlantıyı test et. Eski veriye "güncel değil" işareti koy. Motor kontrolünde bağlantı kaybında durmayı doğrula.

OKU • UYGULA • DOĞRULA

Espressif: Wi-Fi API | EN • STA/AP ve bağlantı olayları.

Eclipse Mosquitto belgeleri | EN • MQTT broker, yayınlama ve abonelik araçları.

12

WEB VE UYGULAMA

Yazılım ile sistemi tamamla

Donanımdan gelen veriyi anlaşılır bir arayüze dönüştürmek, proje değerini artırır. İlk hedefin küçük ama tutarlı bir veri akışı olsun.

Katman	Öğrenilecek konu	Örnek çıktı
HTML + CSS	Anlamlı sayfa yapısı, düzen, mobil uyum.	Cihaz durumu ve ölçüm kartları.
JavaScript	Olaylar, JSON, fetch ve hata yönetimi.	Yenilenen sensör değerleri.
C#	Türler, sınıflar ve hata yakalama.	Bilgisayarda çalışan veri izleme aracı.
MySQL	Tablo, anahtar, INSERT ve SELECT.	Cihaz ölçümlerinin kayıt geçmişi.

Basit bir veri modeli

devices tablosunda cihaz kimliği ve adı; **readings** tablosunda kayıt kimliği, cihaz kimliği, ölçüm zamanı, değer ve birim tut. Tek bir metin alanına her şeyi yazmak yerine alanları ayırmak sorgulamayı ve hata kontrolünü kolaylaştırır.

Tarayıcıdan veritabanına doğrudan parola gönderme. Araya istekleri doğrulayan bir sunucu API’si koy. Sunucuda veri türünü, aralığı ve kullanıcının yetkisini denetle; SQL işlemlerinde parametreli sorgular kullan. İlk sürümde sadece gerekli verileri topla.

Arayüzde dört durum

Yükleniyor: İstek sürüyor. **Veri var:** Değer ve birim görünüyor. **Veri yok:** Henüz kayıt bulunmuyor. **Hata:** Kullanıcıya anlaşılır mesaj ve tekrar deneme yolu sunuluyor. Sadece başarılı yanıtı tasarlamak, gerçek kullanımda boş ekranlara yol açar.

OKU • UYGULA • DOĞRULA

MDN: Learn Web Development | EN • HTML, CSS ve JavaScript için öğrenme rotası.

Microsoft Learn: C# | TR • Dil temelleri ve öğreticiler.

MySQL Tutorial | EN • Tablo oluşturma ve sorgulama; sürümünü eşleştir.

13

GELIŞTIRME ARAÇLARI

Simülasyondan devreye

Simülasyon, yazılım mantığını ve bağlantı fikrini denemeyi hızlandırır. Gerçek devrenin elektriksel sınırlarını ve mekanik koşullarını ayrıca doğrulamak gerekir.

Wokwi ile küçük bir senaryo kur

Desteklenen kart ve bileşenlerle LED, buton ve sensör mantığını dene. Her denemede tek şeyi değiştir: pin, eşik veya süre. Proje bağlantısını ve kullandığın kodu birlikte sakla. Simülasyonda çalışması; gerçek besleme kalitesi, motor paraziti veya ısınma sorunlarının çözüldüğünü göstermez.

KiCad ile şemayı belgele

Önce şemayı çiz; besleme ve sinyal ağlarını isimlendir. Sembolün pin numaralarını fiziksel parçayla eşleştir. PCB aşamasında ayak izi, konnektör yönü, montaj delikleri ve kutu ölçülerini kontrol et. ERC ve DRC kontrolleri hataları bulmaya yardım eder; her tasarım kararını doğrulamaz.

Araç	Ne için kullan?	Saklanacak çıktı
Wokwi	Bağlantı ve program akışını denemek.	Proje bağlantısı + kod sürümü.
KiCad	Şema ve baskı devre hazırlamak.	Şema PDF’i + kaynak proje dosyaları.
PlatformIO	Proje ve bağımlılık ayarlarını düzenlemek.	platformio.ini + sürüm bilgisi.
Git	Değişiklik geçmişi ve geri dönüş.	Anlamlı commit mesajları + README.

İlk üretim öncesi kontrol
Basit LED devresinin şemasını çiz. Konnektörün + ve GND yönünü açıkça işaretle. Kartı 1:1 ölçekte yazdırarak parçaların oturuşunu kontrol et. Sonra ERC/DRC bulgularını tek tek değerlendir; sadece uyarıları susturmakla yetinme.

OKU • UYGULA • DOĞRULA

Wokwi Docs | EN • Başlangıç ve desteklenen donanım.

KiCad Documentation | EN • Getting Started, şema ve PCB editörü.

PlatformIO Documentation | EN • Proje ayarları ve kütüphane yönetimi.

14

UYGULAMA PLANI

Dört haftada ilk proje

Önerilen çalışma düzeni: haftada üç oturum, oturum başına yaklaşık 45-60 dakika. Süreler tahminidir; sorunları anlamaya zaman ayır.

Hafta	Odak ve uygulama	Tamamlanma ölçütü
1	Temel elektronik + LED + buton.	Bağlantı tablosu, direnç hesabı ve ölçüm kaydı hazır.
2	Sensör + seri port + ekran.	Birimli veri görünüyor; sensör hatası ayırt ediliyor.
3	Wi-Fi veya motor kontrolünden biri.	Kesinti ve yeniden başlatma senaryosu denenmiş.
4	Kutu, dokümantasyon ve sunum.	Başka biri yönergeyle projeyi tekrar kurabiliyor.

Üç proje rotasından birini seç

A / Masaüstü ortam göstergesi: ESP32, uyumlu sıcaklık-nem sensörü ve OLED. Hedef: ölçüm, birim, güncelleme ve hata durumunu göstermek. Kaynak rotası: 07 → 08 → 11.

B / Mesafe göstergesi: Geliştirme kartı, uygun mesafe sensörü ve LED/buzzer. Hedef: mesafe değiştikçe göstergeyi güncellemek. Sensör çıkış seviyesi ile kart girişini eşleştir. Kaynak rotası: 02 → 05 → 08.

C / Kontrollü mini araç: ESP32, motor sürücüsü, motorlar ve uygun hazır güç paketi. Hedef: yön/hız kontrolü ve komut kaybında durma. Kaynak rotası: 07 → 09 → 10 → 11.

Teslim dosyasında bulunması gerekenler

Amaç ve kapsam; malzeme listesi; tam kart modeli; bağlantı tablosu; şema; kurulum adımları; kod ve sürümler; test sonuçları; bilinen sınırlamalar; yararlanılan kaynaklar. Bir fotoğraf ve kısa gösterim videosu eklemek anlatımı güçlendirir.

> **Başarı ölçüsü**
> “Daha çok parça” yerine “daha iyi doğrulama” hedefle. Örneğin ekranın açılması tek testtir; sensörün sökülmesi, cihazın yeniden başlatılması ve ağın kesilmesi ayrı testlerdir.

OKU • UYGULA • DOĞRULA

Arduino Built-in Examples | EN • Projeyi küçük testlere ayırmak için örnekler.

Pro Git | TR/EN • Değişiklikleri kaydet, çalışan sürüme geri dön.

15

BAŞVURU SAYFASI

Hata giderme ve kaynak dizini

Bir hata çıktığında önce belirtisini kaydet, sonra en küçük çalışan düzene geri dön. Aynı anda birkaç şeyi değiştirmek nedeni gizler.

Belirti	İlk kontrol	Sonraki adım
Kart görünmüyor	Veri kablosu, USB portu ve işletim sistemi.	Kartın resmî kurulum belgesini incele.
Kod derlenmiyor	İlk hata satırı, kart ve kütüphane sürümü.	En küçük örneği derle; API sürümünü eşleştir.
Ekran boş	Besleme, denetleyici ve I2C adresi.	Tarama ve üretici örneğini ayrı çalıştır.
Motorla reset oluyor	Besleme düşümü ve güç kabloları.	Motor ve kart yükünü ayrı ölç.
Veri tutarsız	Birim, referans, örnekleme ve bağlantı.	Ham veriyi kaydet; filtreyi sonra ekle.

Hızlı kaynak dizini

Temeller: SparkFun Ohm Yasası (02), Fluke ölçüm kılavuzu (04).
Arduino: Kart belgeleri (05), Language Reference ve yerleşik örnekler (06).
ESP32: Espressif başlangıç ve geçiş rehberi (07), I2C API (08), Wi-Fi API (11).
Donanım: TI DRV8833 (03, 09), LM2596 ve pil kaynakları (10).
Yazılım: MDN, Microsoft Learn ve MySQL belgeleri (12).
Araçlar: Wokwi, KiCad, PlatformIO (13), Pro Git (01, 14).

Kısa sözlük

GPIO: Genel amaçlı giriş/çıkış. **ADC:** Analog değeri sayısallaştırır. **PWM:** Darbelerin görev oranıyla kontrol. **API:** Yazılımın sunduğu kullanım arayüzü. **Datasheet:** Üreticinin teknik veri belgesi. **BOM:** Malzeme listesi. **GND:** Devrenin ortak referans noktası.

Kaynak kullanma ilkesi: Önce üretici belgesi, sonra örnek proje, ardından topluluk tartışması. Kopyaladığın kodun lisansını koru; belge tarihini ve sürümünü not et. Bağlantılar 29 Ekim 2026'da kontrol edildi; "latest" sayfaları zamanla değişebilir. EN kaynaklar İngilizce, TR kaynaklar Türkçedir. Rehberdeki uygulama planları özgün önerilerdir.

OKU • UYGULA • DOĞRULA

Espressif: GPIO API | EN • Pinlerin çalışma biçimleri ve kısıtlamaları.

Espressif: LEDC API | EN • Kurulu sürüme uygun PWM kullanımını doğrula.