

DENEYAP KART 1A v2 İLE İTERAKTİF LED GÖSTERİMİ

A) Deneyap Kart IDE Kurulumu:

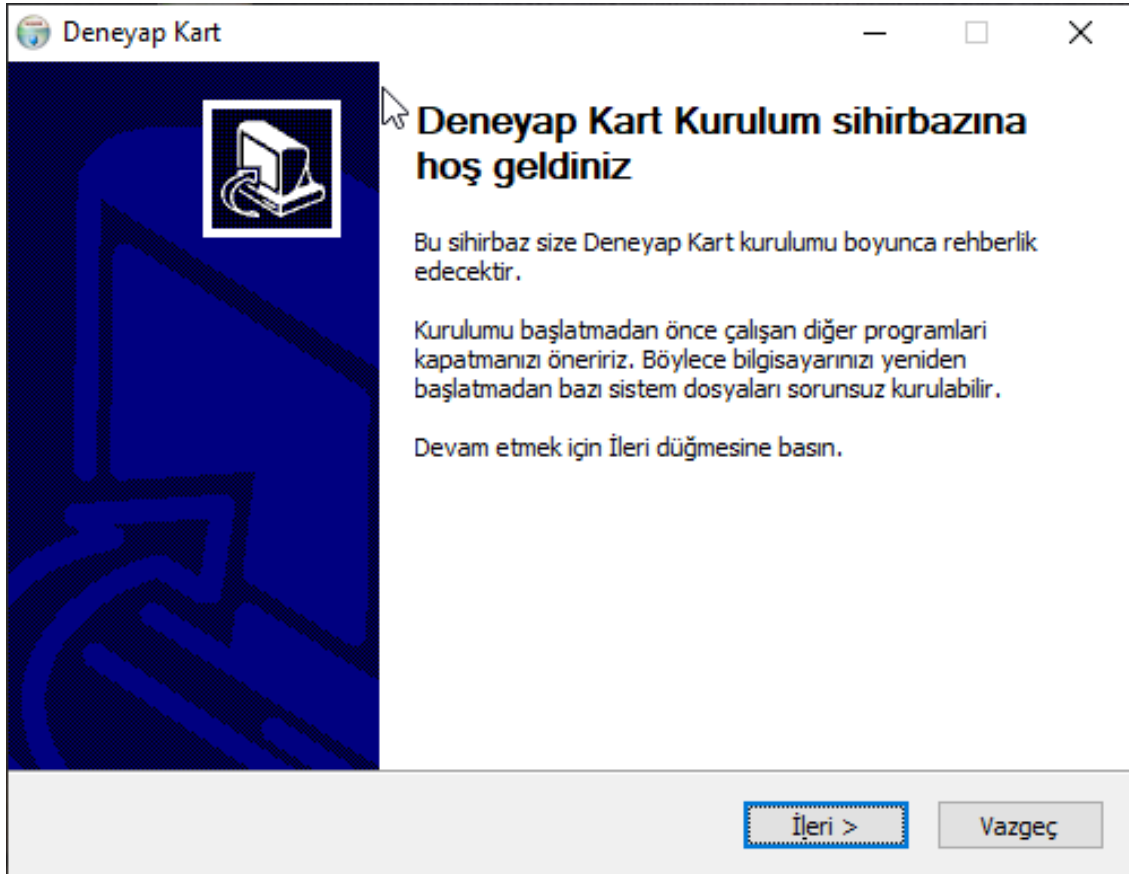
Deneyap Kart IDE, metin tabanlı programlama yapmanız için geliştirilmiş bir programlama ortamıdır. Eğer metinsel kodlama yapmak istiyorsanız, Deneyap Kart IDE'yi kullanmanız gerekir.

1-IDE'yi İndirme ve Kurma:

Deneyap Kart IDE'nin en güncel sürümünü resmi siteden indirip bilgisayarınıza kurun.

2-Uygulamayı Başlatma:

Kurulum tamamlandıktan sonra IDE'yi çalıştırın.

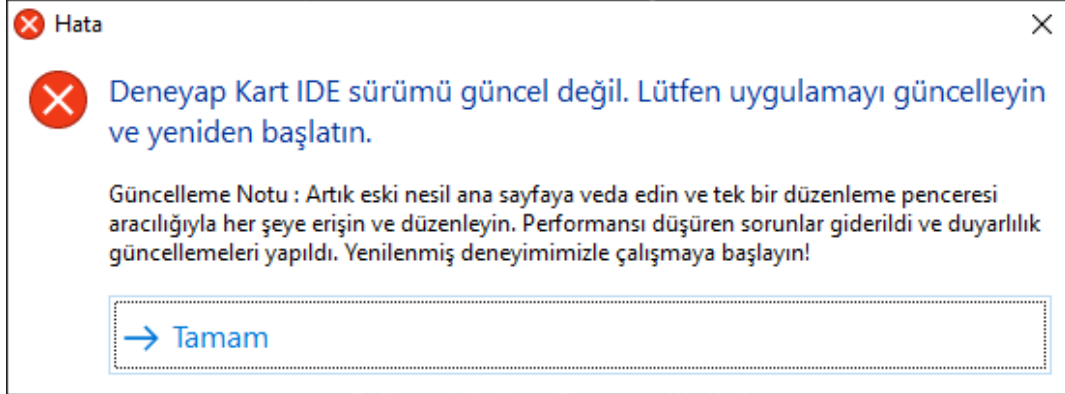


Şekil 1



3-Güncelleme Gerekirse:

“Sürüm güncel değil” uyarısı alırsanız, uygulamayı kaldırıp en son sürümü tekrar yükleyin.



Şekil 2

4-İlk Kurulum ve Gerekli Dosyalar:

IDE ilk açıldığında, donanım ve yazılım bileşenlerini otomatik olarak indirip kurar. Bu işlem sırasında internet bağlantınızın hızı etkilidir. burada belki 25-50 arası sonrası bağlantının yavaş olduğunu , sayfanın kapatılmaması gerektiğini (sunumda konuşuldu) eklenebilir.

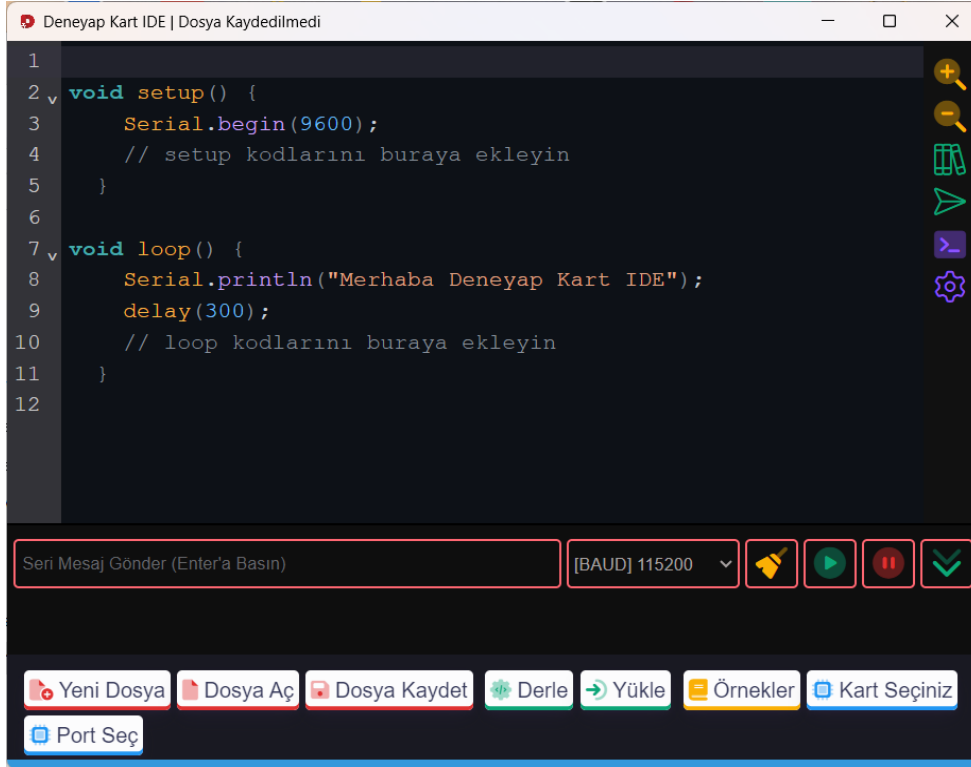


Şekil 3



5-İlk Açılış ve Arayüz:

Kurulum tamamlandığında IDE yeniden başlar ve sizi kullanıcı arayüzü karşılar.



Şekil 4

6-Kurulum Başarılı:

Tebrikler! Deneyap Kart IDE kullanıma hazır. Artık projelerinizi geliştirmeye başlayabilirsiniz.

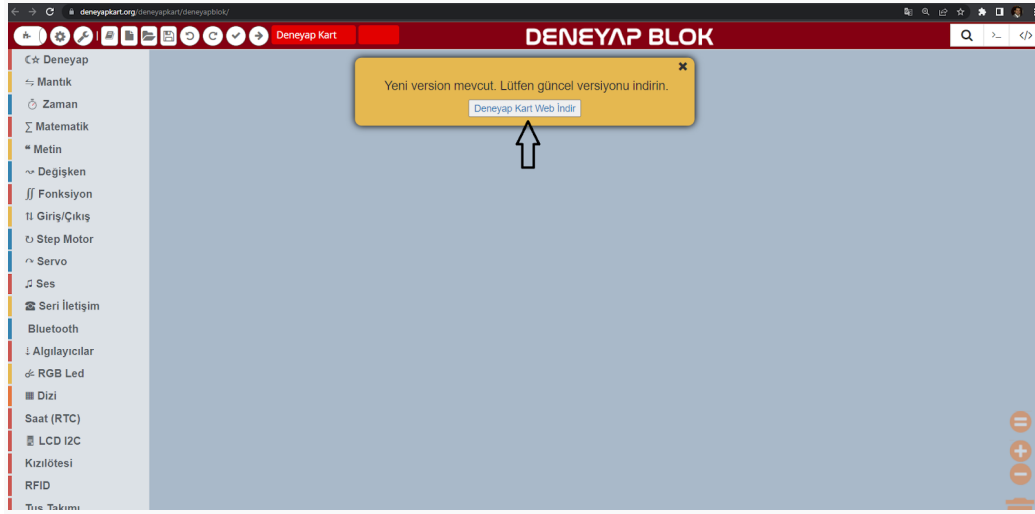
B) DENEYAP Kart Blok Web Kurulumu:

Blok tabanlı (sürükle-bırak mantığıyla çalışan) bir programlama yöntemi tercih ediyorsanız, bunun için ayrıca Deneyap Kart Blok uygulamasını yüklemeniz gerekir.

<https://blok.deneyapkart.org/> adresinden Deneyap Kart Blok sayfasına geçiş yapıyoruz.

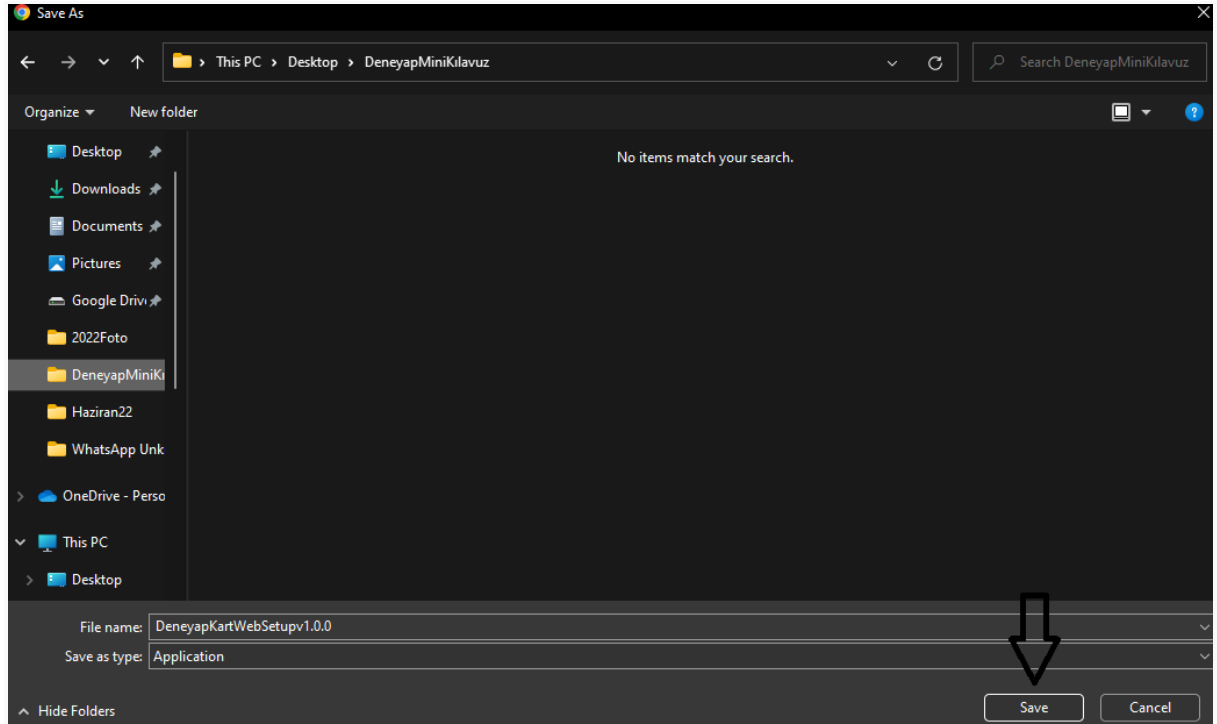


1- Deneyap Kart Blok sayfasına geiş yapıldığında “Deneyap Kart Web İndir” butonuna tıklanılır.



Şekil 5

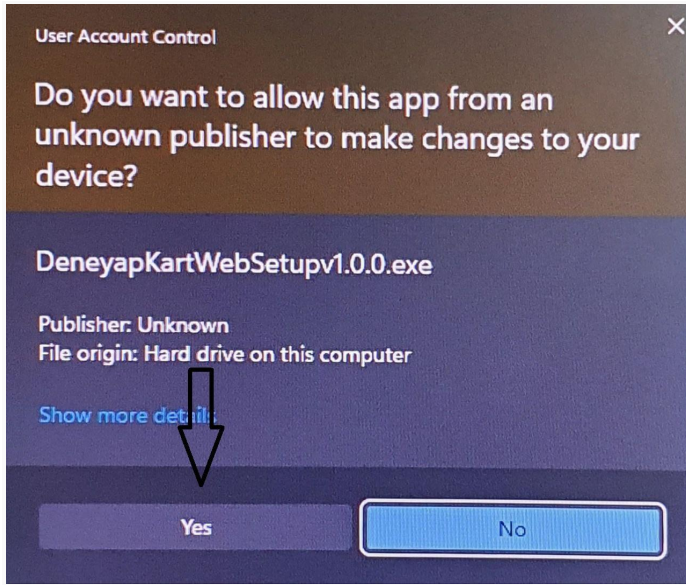
2- Açılan pencerede “Save” butonuna tıklayıp, programı bilgisayarınıza indiriniz.



Şekil 6

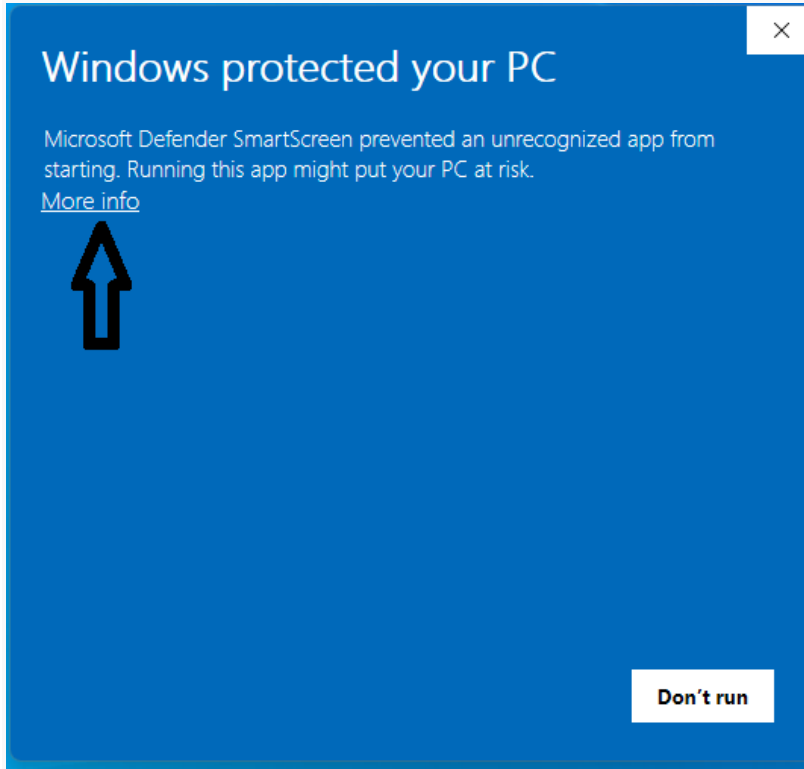


3- İndirme işlemi tamamlandıktan sonra kurulum işlemini başlatıyoruz.



Şekil 7

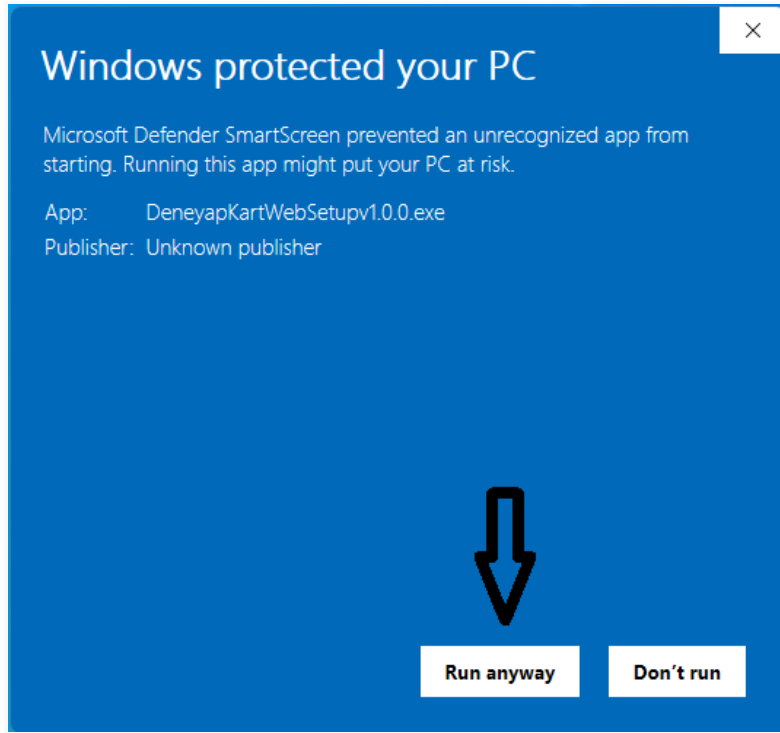
4- Kurulum başlatma sırasında Windows Defender vb. programlar aşağıdaki şekilde uyarı verebilir. Bu şekilde bir ekran ile karşılaşırsanız "More info" yazısına tıklayınız.



Şekil 8

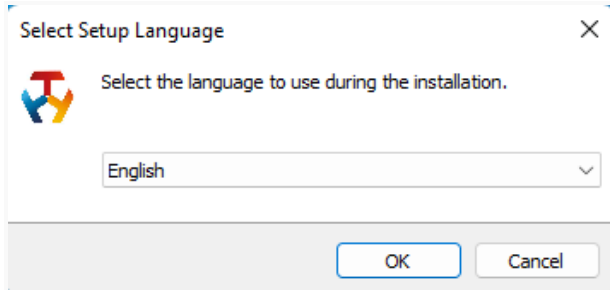


5- Yükleme adımına geçmek için “Run anyway” butonuna tıklayınız.



Şekil 9

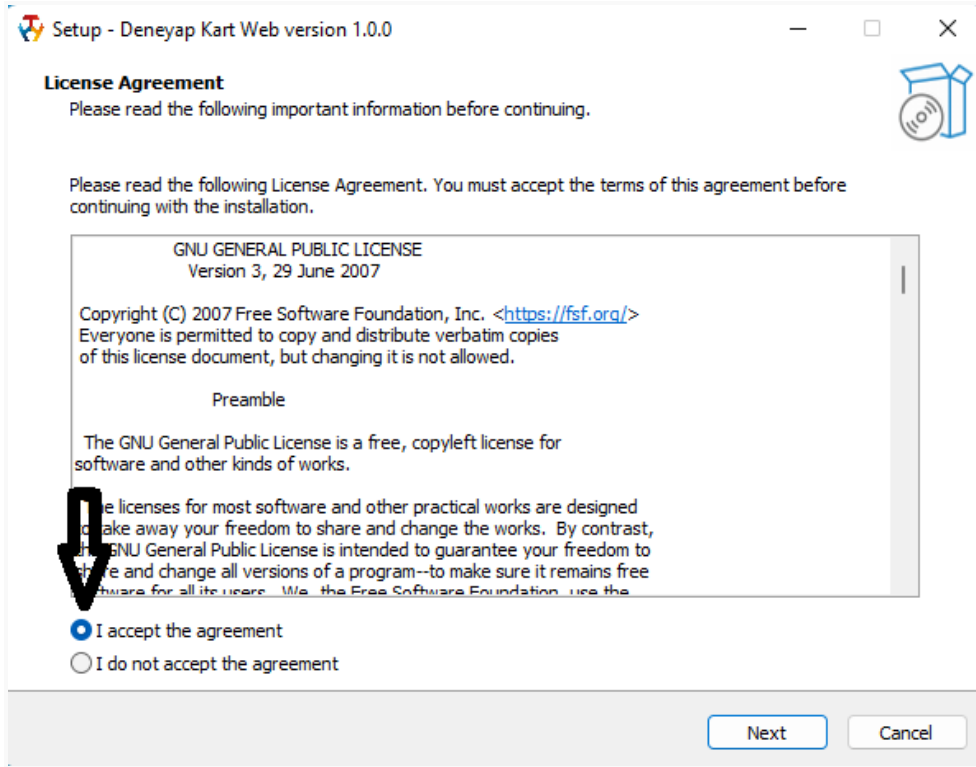
6- Yüklemenin ilk adımında dil seçimi yapmanız istenir. Dil seçimi yapıp “OK” butonuna tıklayınız.



Şekil 10

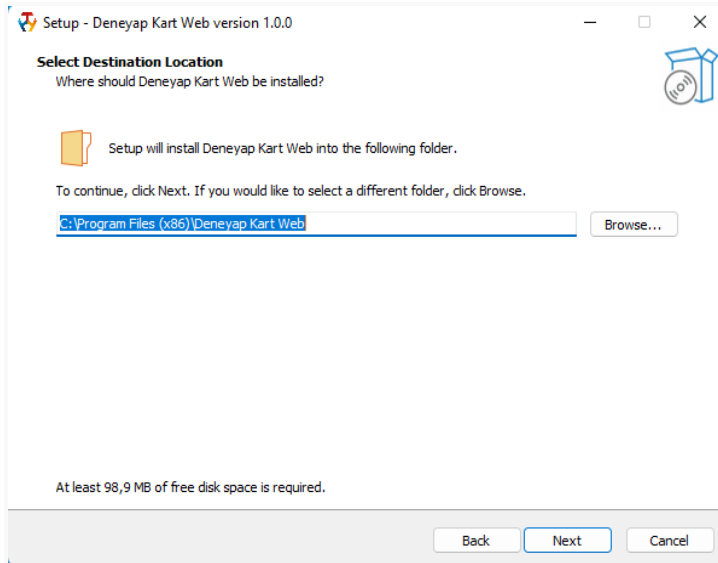


7- Ardından gelen ekranda lisans anlaşmasını onayladıktan sonra “Next” butonuna tıklayınız.



Şekil 11

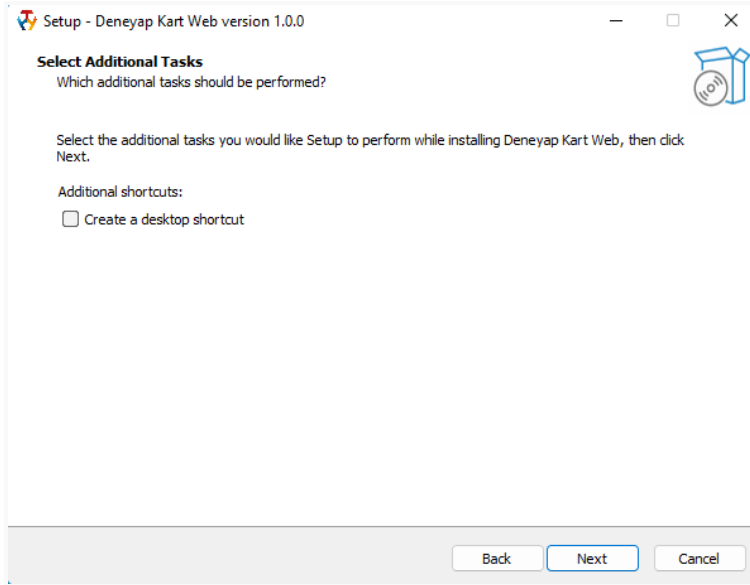
8- Gelen ekranda kurulumun nereye yapılacağı gösterilmektedir. “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 12

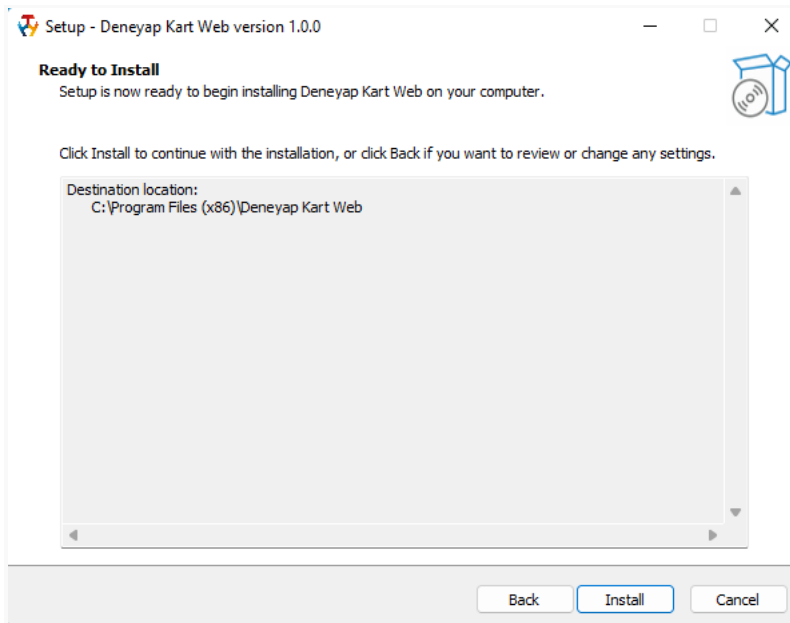


9- Bir sonraki sayfada masaüstüne kısayol oluşturulması ile ilgili seçiminizi yapıp, “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 13

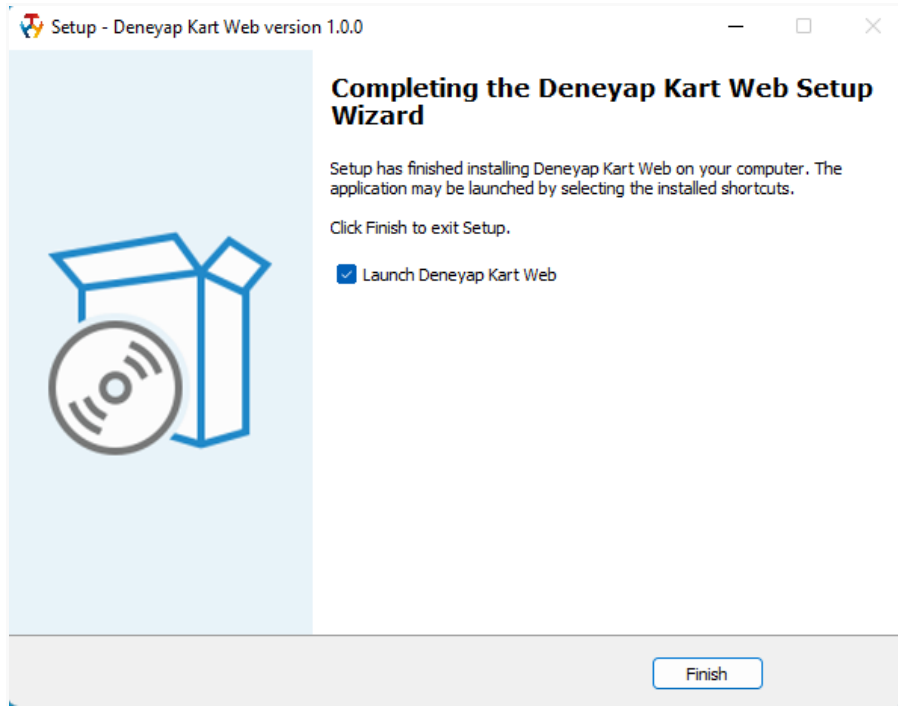
10- Gelen ekranda “Install” butonuna tıklayıp yüklemeyi başlatabilirsiniz.



Şekil 14



11- Yükleme tamamlandıktan sonra “Finish” butonuna tıklayınız.



Şekil 15

12- Windows Taskbar’da (Windows Görev Çubuğunda) sağ alt köşede Deneyap Kart Web programının ikonunu aşağıdaki gibi görebilirsiniz.



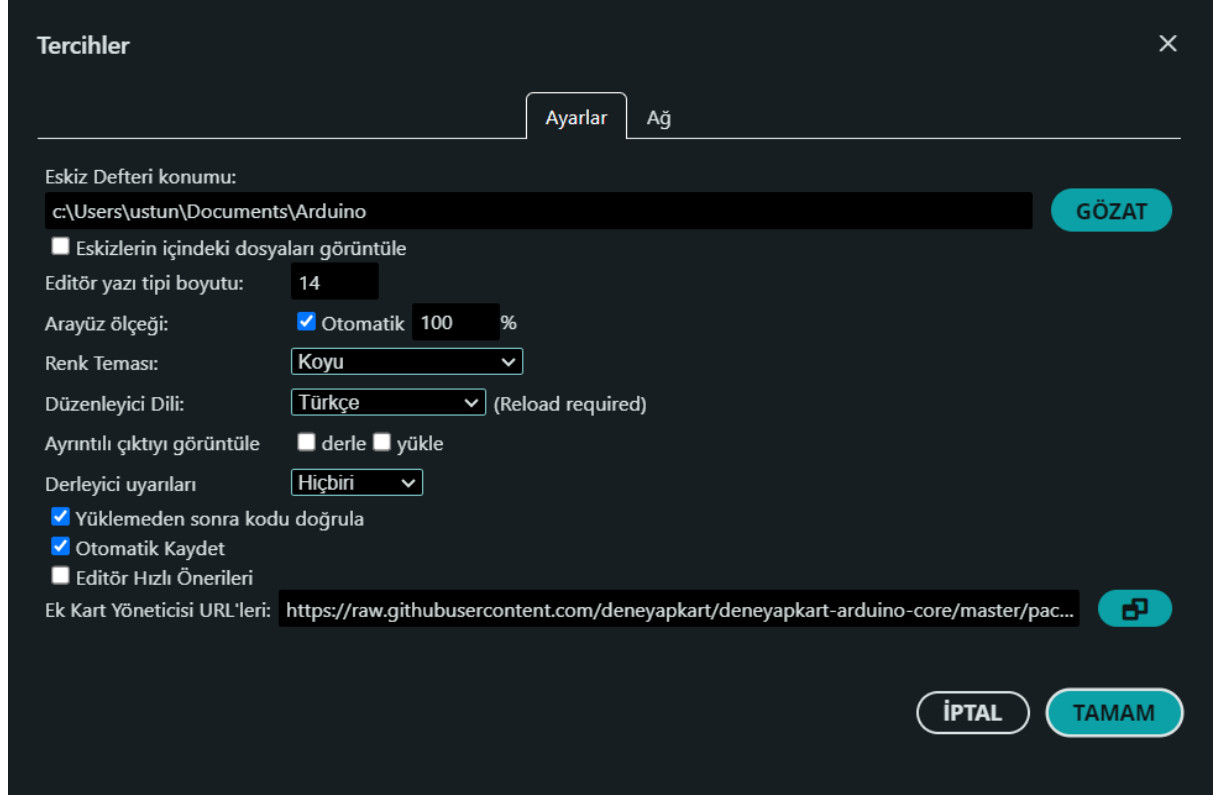
Şekil 16



C) Arduino IDE Kullanacaksanız: Deneyap Kart Nasıl Kullanılır?

1- Arduino IDE’de Dosya (File) → Tercihler (Preferences) adımını takip ederek açılan pencerede Ayarlar (Settings) sekmesinde bulunan Ek Devre Kartları Yöneticisi URL’leri (Additional Boards Manager URLs) kısmına, aşağıda paylaşılan JSON dosyasına ait adresi kopyalayınız.

https://raw.githubusercontent.com/deneyapkart/deneyapkart-arduino-core/master/package_deneypkart_index.json



Şekil 17

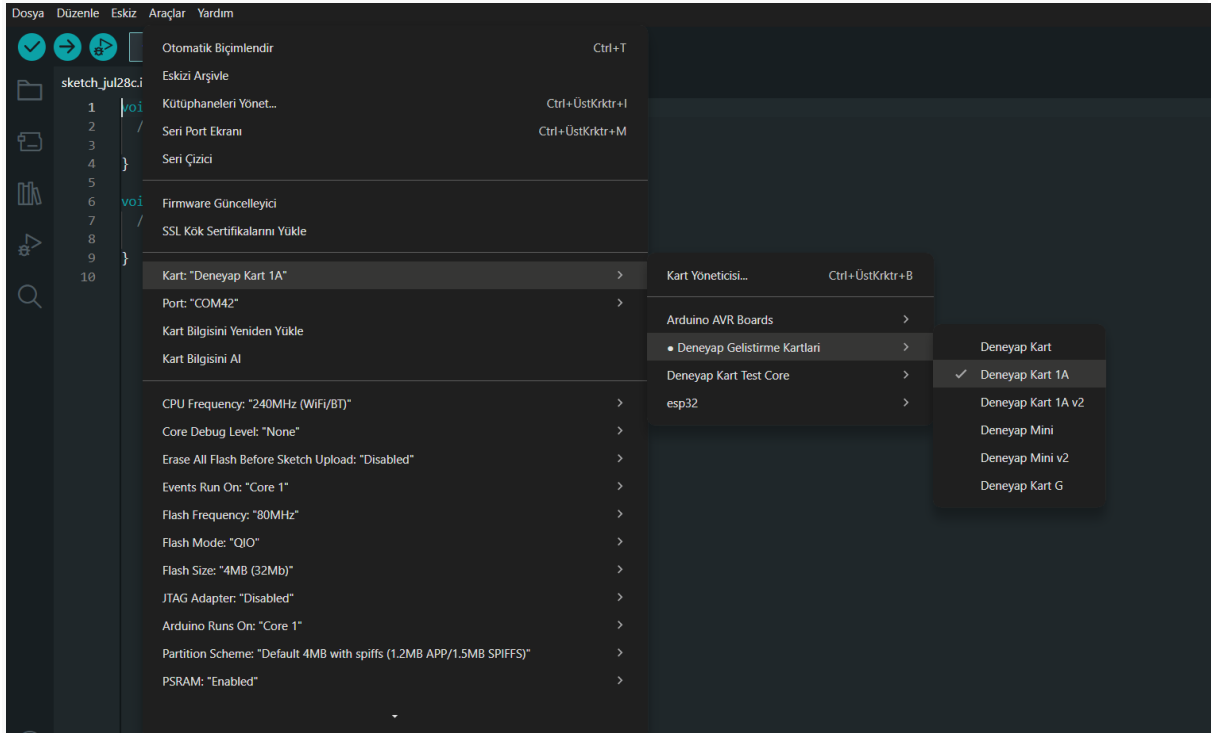


2- Ardından, Araçlar (Tools) → Kart (Board) → Kart Yöneticisi (Boards Manager) adımını takip ederek gelen ekranda arama satırına “Deneyap Gelistirme Kartlari” yazınız. En son sürüm varsayılan olarak seçilmiş halde gelir ve Kur (Install) butonuna tıklayarak yüklemeyi gerçekleştirebilirsiniz.



Şekil 18

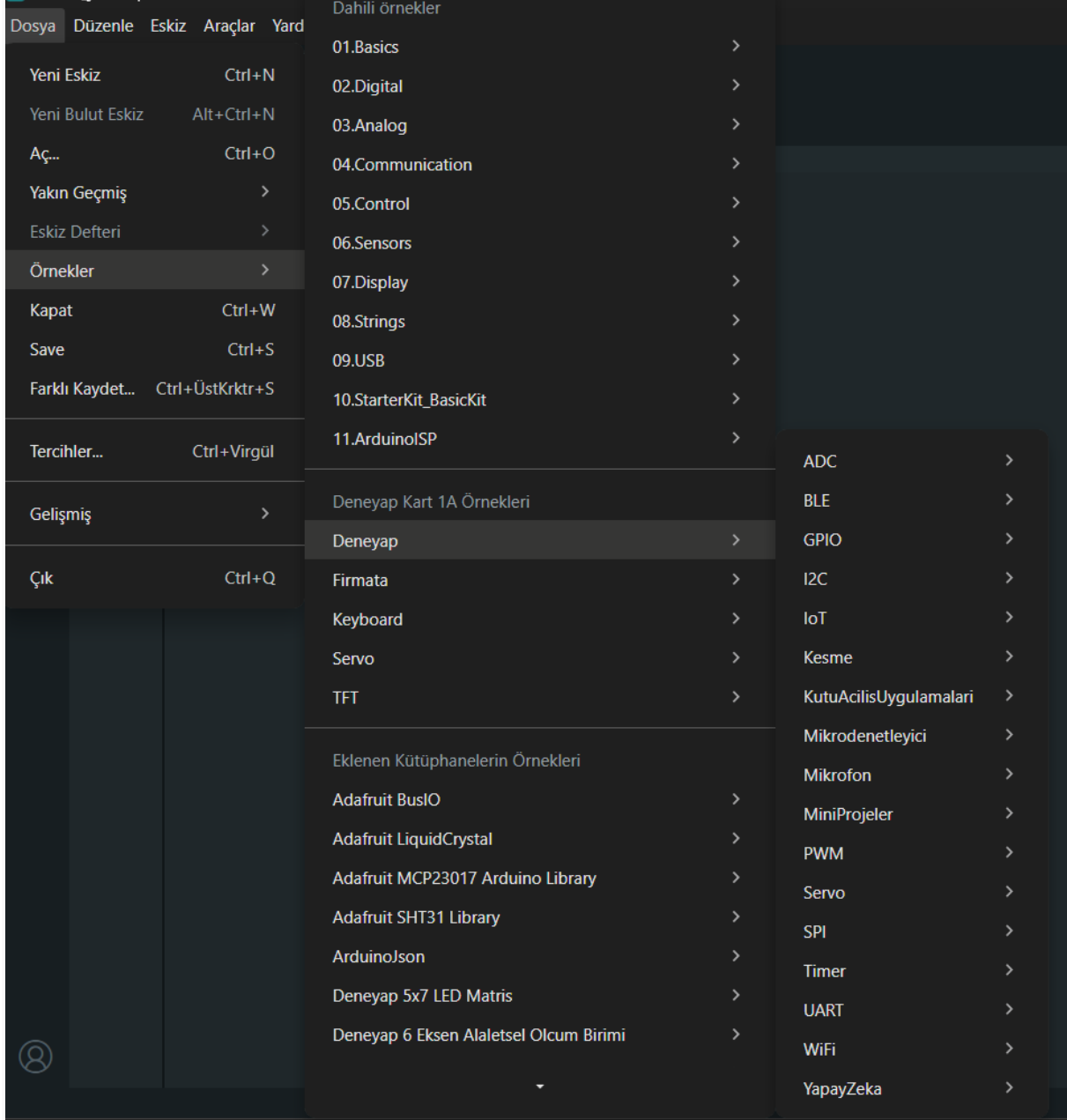
3- En son aşama olarak Araçlar (Tools) → Kart (Board) adımından “Deneyap Kart” ve kartınızın bağlı olduğu Port’u seçerek kodlama adımına geçebilirsiniz.



Şekil 19



4- Örnek kodlara erişmek için Dosya (Files) → Örnekler (Examples) başlığı altında “Deneyap Kart için Örnekler (Examples for Deneyap Kart)” yazan kısmı takip edebilirsiniz.



Şekil 20



A) Deneyap Kart IDE'ye Kütüphane Kurulumu

Deneyap Kart IDE'de bir sensör ya da modül kullanmak için ilgili kütüphaneyi kurmanız gerekir. Aşağıdaki adımları izleyerek kolayca kütüphane ekleyebilirsiniz:

1. Deneyap Kart IDE'yi Açın

Uygulamayı başlatın ve ana ekranda sol taraftaki araç çubuğunda bulunan  simgesine tıklayın.

2. Kütüphane Yöneticisine Girin

Açılan pencerede, üstteki arama çubuğuna kurmak istediğiniz kütüphanenin adını yazın. Örneğin:

- Deneyap 5x7 Led Ekran
- Deneyap 6-Eksen Ataletsel Ölçüm Birimi
- Deneyap 9-Eksen Ataletsel Ölçüm Birimi

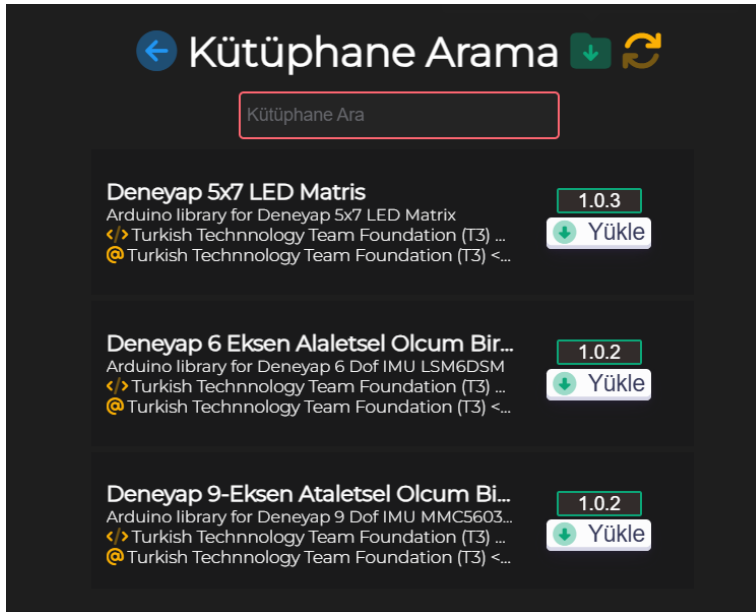
3. İlgili Kütüphaneyi Seçin ve Kurun

Arama sonucunda çıkan kütüphaneyi seçin ve sağ alt köşedeki **"Kur"** butonuna tıklayın.

Kurulum tamamlandığında, "Kuruldu" mesajı görünecektir.

4. Artık Kullanıma Hazır!

Kütüphaneyi kurduktan sonra, yeni bir proje oluşturup dosyanın en üstüne `#include <kütüphane_adi.h>` şeklinde ekleyerek doğrudan kullanabilirsiniz.



Şekil 21



B) Arduino IDE’de Deneyap Kütüphanesi Kurulumu

Arduino IDE’de bir sensör ya da modül kullanmak için ilgili kütüphaneyi kurmanız gerekir. Aşağıdaki adımları izleyerek kolayca kütüphane ekleyebilirsiniz:

1. Arduino IDE’yi Açın.

2.Sol taraftaki araç çubuğunda Kitaplık Yöneticisi simgesine tıklayın.

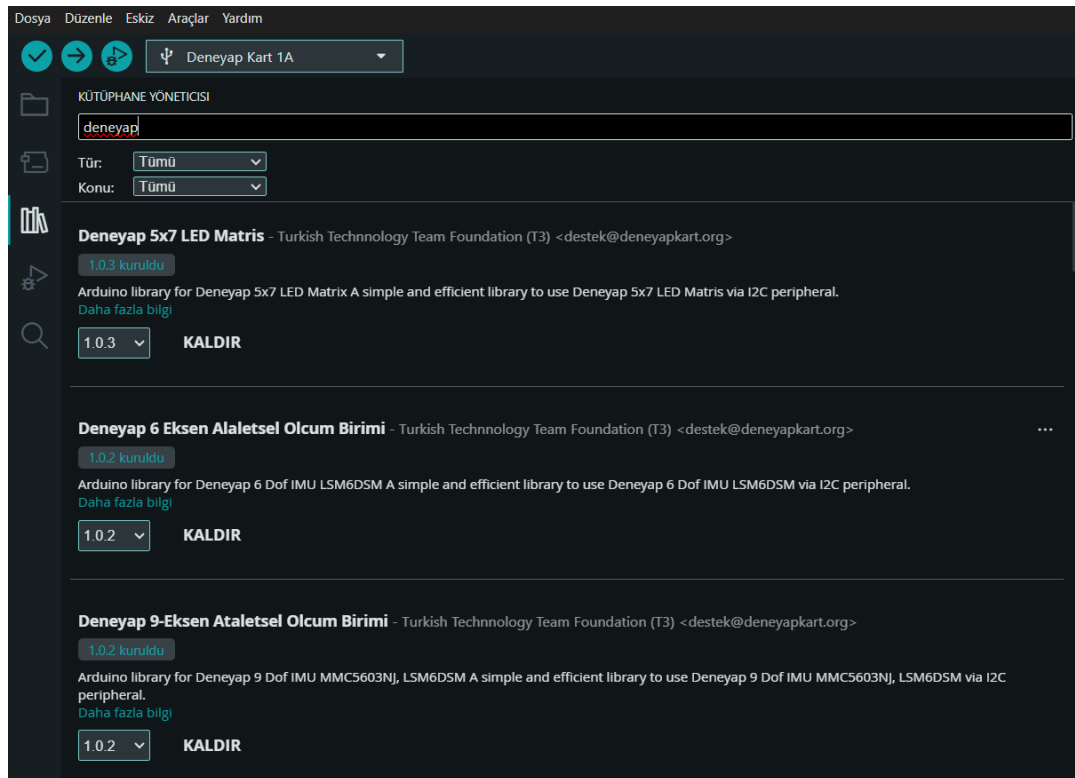
3.Üstteki arama kutusuna deneyap yazın.

- Deneyap 5x7 LED Matris,
- Deneyap 6 Eksen Ataletsel Ölçüm Birimi,
- Deneyap 9-Eksen Ataletsel Ölçüm Birimi gibi kütüphaneler çıkacak.

4.Kullanmak istediğin kütüphanenin sağ alt köşesindeki “Yükle” (veya “Install”) butonuna tıkla. Kurulum tamamlandığında, o kütüphane Arduino IDE’de kullanıma hazır hale gelir.

5.Artık projenin başında, ilgili kütüphaneyi şöyle ekleyebilirsin:

```
#include <Deneyap_5x7LedEkran.h>
veya
#include <Deneyap_DokunmatikTusTakimi.h>
```



Şekil 22



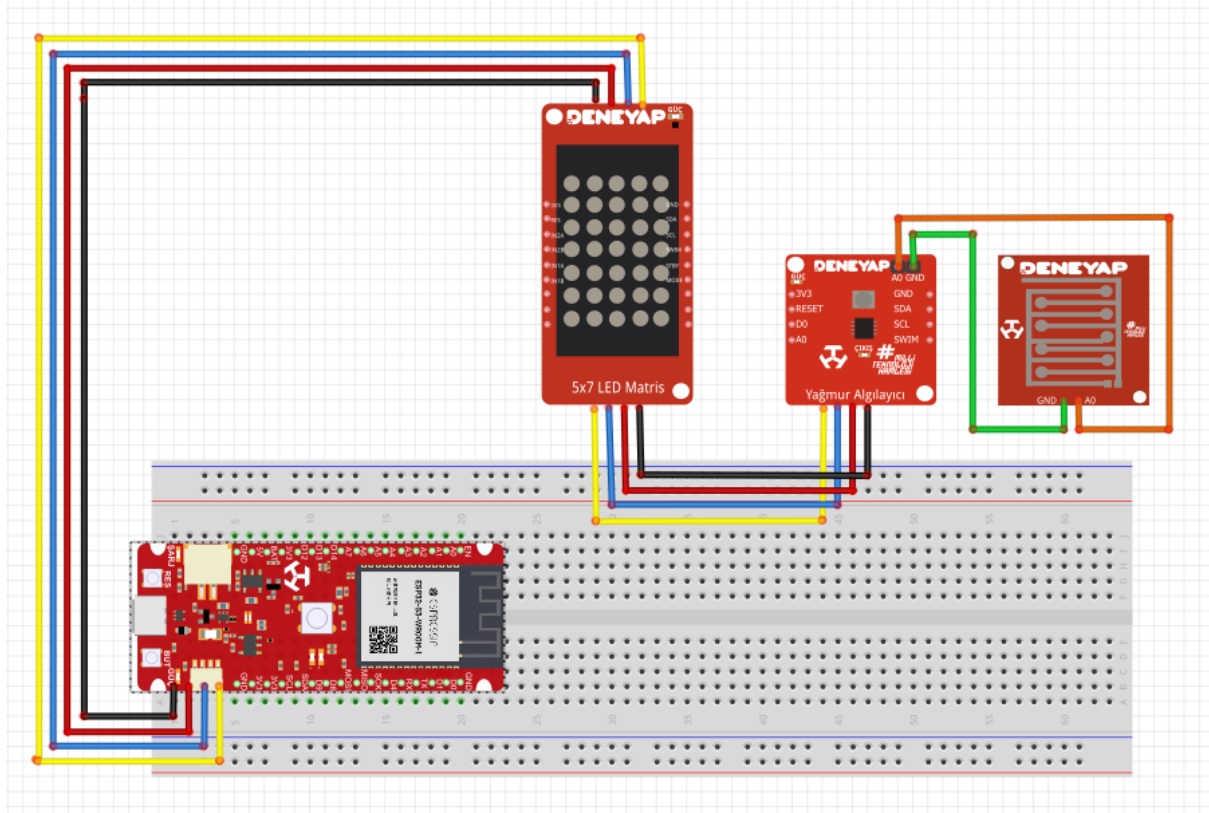
Proje Özeti: Bu proje, Deneyap Kart üzerinde bulunan yağmur algılama modülünü kullanarak yağmur durumunu tespit etmeye ve 5x7 LED matris ekranda görsel olarak göstermeye yöneliktir. Program, yağmur sensöründen hem dijital hem de analog veri okuyarak yağmurun varlığını ve yoğunluğunu ölçer. Yağmur algılandığında LED matriste özel hazırlanmış yağmur dalgası deseni görüntülenir, yağmur yoksa ekran temizlenir. Aynı zamanda sensörden alınan veriler seri monitöre yazdırılır. Böylece, gerçek zamanlı yağmur durumu hem görsel hem de sayısal olarak takip edilebilir.

Gerekli Malzemeler:

- Bir adet Deneyap Kart 1A v2
- Bir adet Deneyap 5x7 LED Matris
- Bir adet Deneyap Yağmur Algılayıcı Sensörü
- Bir adet Deneyap Yağmur Algılayıcı Plakası
- İki adet I2C kablosu
- İki adet Jumper Kablo

Devre Şeması:

Devre şeması aşağıdaki gibi kurulabilir.



Şekil 23



Proje Kodları:

- Bu kod satırları, Deneyap Kart üzerinde bulunan yağmur algılama modülü ve 5x7 LED matris ekranı kullanmak için gerekli olan kütüphaneleri programa #include komutu ile dahil eder.
- **Deneyap_YagmurAlgilama.h** dosyası, yağmur sensöründen veri okumayı sağlayan fonksiyonları içerir.
- **Deneyap_5x7LedEkran.h** dosyası ise LED matris ekranı kontrol etmek için gereken fonksiyonları sağlar.
- Böylece bu iki modül programda kullanılabilir hale gelir.

```
#include <Deneyap_YagmurAlgilama.h>

#include <Deneyap_5x7LedEkran.h>
```

Şekil 24

- Bu satırlarda **Rain** sınıfından **YagmurSensor** adında bir nesne ve **DotMatrix** sınıfından **LEDMatris** adında bir nesne oluşturuluyor.
- **YagmurSensor** nesnesi, yağmur algılama modülünü kontrol etmek ve sensörden veri almak için kullanılır.
- **LEDMatris** nesnesi ise 5x7 LED matris ekranı kontrol etmek için kullanılır.
- Böylece programda bu nesneler üzerinden sensörden veri alınabilir ve LED ekranda görüntüleme yapılabilir.

```
Rain YagmurSensor;

DotMatrix LEDMatris;
```

Şekil 25

- Bu dizi, 5 sütun ve 7 satırdan oluşan bir LED matris üzerindeki bir görüntüyü temsil eder.
- Dizideki her **1** değeri, LED matris üzerindeki ilgili noktada ışığın yanması gerektiğini belirtir.
- **0** değeri ise LED'in kapalı kalması gerektiğini gösterir.
- Bu örnekte, dizideki değerler bir "yağmur damlası" veya "dalga" şeklinde basit bir desen oluşturur.
- Böylece LED matris üzerinde bu şeklin görüntülenmesi sağlanır.

```
const byte yagmurDalga[] = {
  0, 0, 1, 0, 0,
  0, 1, 0, 1, 0,
```



```

1, 0, 0, 0, 1,
0, 1, 1, 1, 0,
0, 0, 1, 0, 0,
0, 1, 0, 1, 0,
1, 0, 0, 0, 1
};

```

Şekil 26

- **setup()** fonksiyonunda ilk olarak **Serial.begin(115200);** komutu ile seri iletişim 115200 baud hızında başlatılır.
- Bu, bilgisayar ile Deneyap Kart arasında veri alışverişi yapılmasını sağlar.
- Ardından **YagmurSensor.begin(0x2E)** fonksiyonu çağrılır; bu fonksiyon, yağmur sensörünü I2C adresi **0x2E** üzerinden başlatmayı dener.
- Eğer sensör başlatılamazsa:
 - **delay(3000);** ile 3 saniye beklenir.
 - **Serial.println("Yağmur sensörü I2C bağlantısı başarısız");** komutuyla hata mesajı seri monitöre yazdırılır.
 - **while(1);** ile program sonsuz döngüye girer ve devam etmez.
- Benzer şekilde **LEDMatris.begin(0x0A)** fonksiyonu, LED matris ekranını I2C adresi **0x0A** üzerinden başlatmayı dener.
- Eğer bağlantı başarısız olursa:
 - 3 saniye beklenir (**delay(3000);**).
 - **Serial.println("LED Matris I2C bağlantısı başarısız");** mesajı yazılır.
 - **while(1);** ile program durdurulur.
- Bu şekilde, **begin()** fonksiyonları ile donanımların I2C bağlantıları kontrol edilir.
- Hata durumunda kullanıcıya bilgi verilerek program durdurulur.
- Bu yaklaşım, donanım sorunlarının önceden tespiti için oldukça önemlidir.

```

void setup() {
  Serial.begin(115200);

  if (!YagmurSensor.begin(0x2E)) {
    delay(3000);
    Serial.println("Yağmur sensörü I2C bağlantısı başarısız");
    while (1);
  }

  if (!LEDMatris.begin(0x0A)) {
    delay(3000);
    Serial.println("LED Matris I2C bağlantısı başarısız");
    while (1);
  }
}

```

Şekil 27



- **loop()** fonksiyonunda ilk olarak **YagmurSensor.ReadRainDigital()** fonksiyonu çağrılır.
- Bu fonksiyon, yağmur sensöründen dijital (açık/kapalı - on/off) yağmur durumu bilgisini okur.
- Okunan değer **yagmurDurum** değişkenine atanır.
- Ardından **YagmurSensor.ReadRainAnalog()** fonksiyonu çağrılır.
- Bu fonksiyon, sensörden analog (sayısal ve detaylı) yağmur miktarı bilgisini alır.
- Elde edilen değer **yagmurDeger** değişkenine atanır.
- **Serial.print()** ve **Serial.println()** komutları ile bu iki değer seri monitöre yazdırılır.
- Böylece kullanıcı, sensörden gelen verileri gözlemleyebilir.
- **if (yagmurDurum == 0)** koşulu ile dijital yağmur durumu kontrol edilir.
- Eğer **yagmurDurum** değeri 0 ise, yani yağmur algılanmışsa:
 - **LEDMatris.drawLedMatrix(yagmurDalga);** komutu çalıştırılır ve önceden tanımlanmış **yagmurDalga** deseni LED matris ekranda gösterilir.
- Eğer yağmur algılanmamışsa (**else** durumu):
 - Sadece sıfırlardan oluşan boş bir dizi oluşturulur.
 - **LEDMatris.drawLedMatrix(bosDizi);** ile LED matris ekranı temizlenir.
- Son olarak **delay(500);** komutu ile 500 milisaniye beklenir.
- Bu şekilde döngü tekrar başlar ve sensör verileriyle ekran düzenli olarak güncellenmiş olur.

```
void loop() {
    bool yagmurDurum = YagmurSensor.ReadRainDigital();
    uint16_t yagmurDeger = YagmurSensor.ReadRainAnalog();

    Serial.print("Yagmur Durumu: ");
    Serial.print(yagmurDurum);
    Serial.print("\tYagmur Degeri: ");
    Serial.println(yagmurDeger);

    if (yagmurDurum == 0) {
        LEDMatris.drawLedMatrix(yagmurDalga);
    } else {
        byte bosDizi[35] = {0};
        LEDMatris.drawLedMatrix(bosDizi);
    }

    delay(500);
}
```

Şekil 28