

DENEYAP KART 1A v2 İLE

ENERJİ TASARRUFLU ODA KONTROLÜ

A) Deneyap Kart IDE Kurulumu:

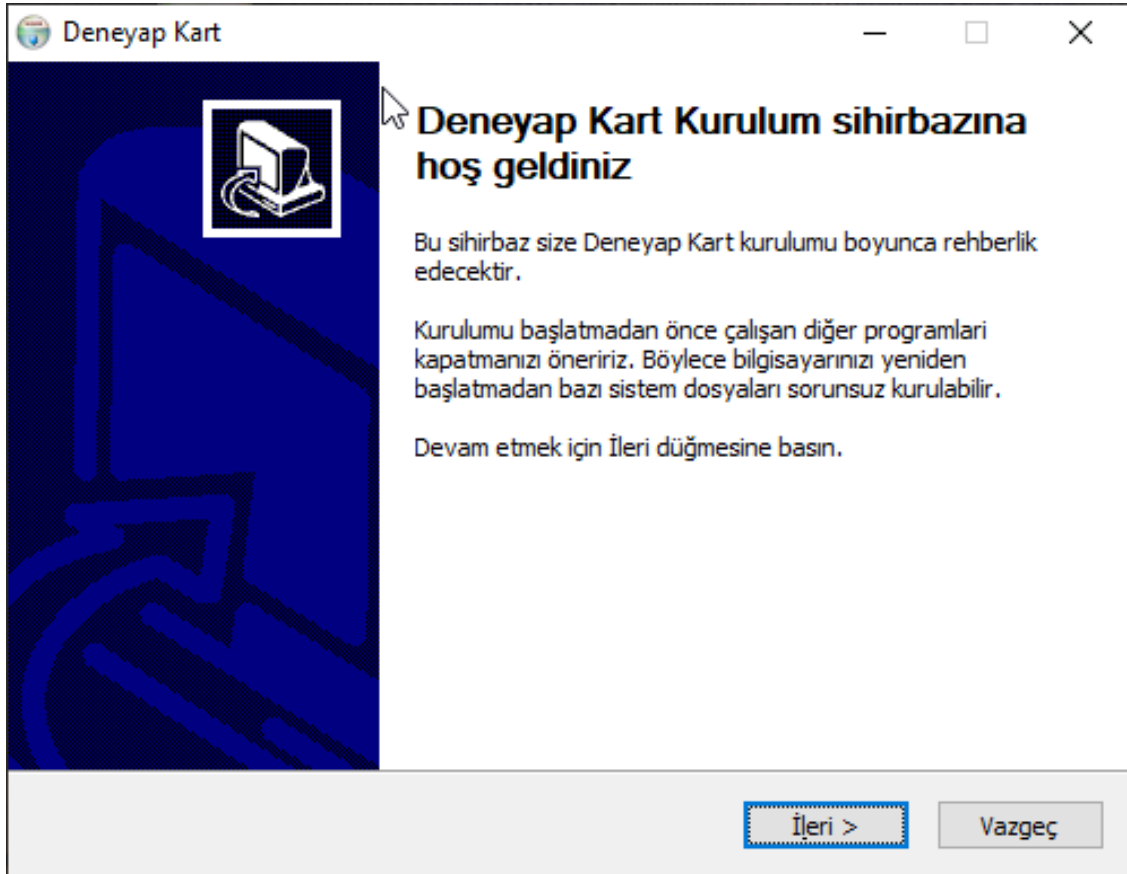
Deneyap Kart IDE, metin tabanlı programlama yapmanız için geliştirilmiş bir programlama ortamıdır. Eğer metinsel kodlama yapmak istiyorsanız, Deneyap Kart IDE'yi kullanmanız gerekir.

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.

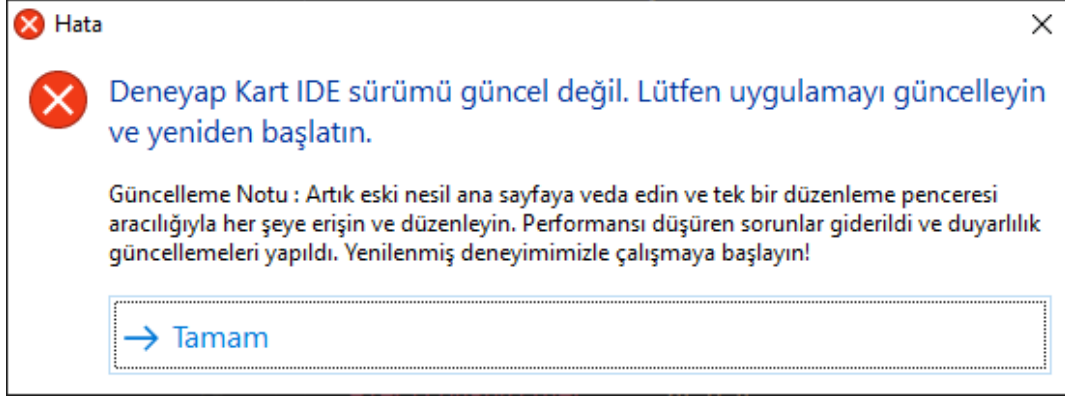


Şekil 1



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.

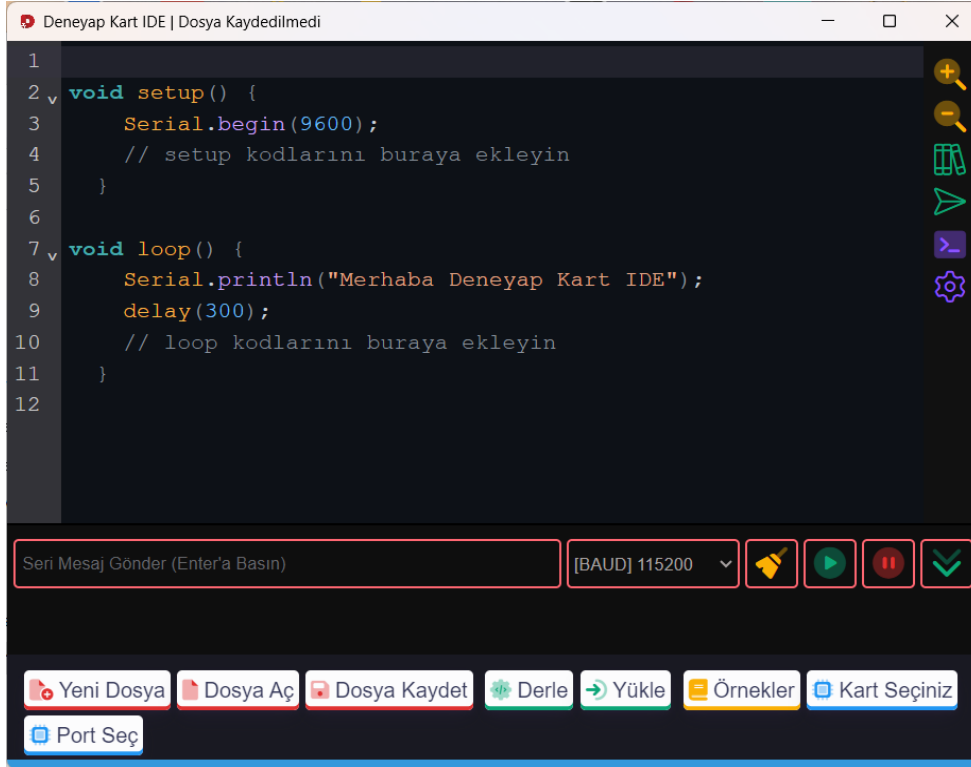


Şekil 3



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.

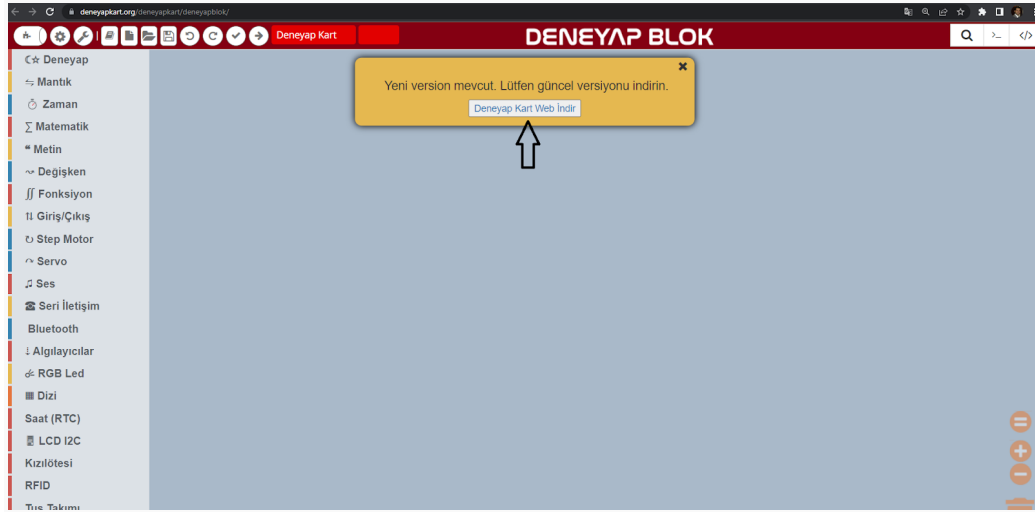
B) DENEYAP Kart Blok Web Kurulumu:

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.

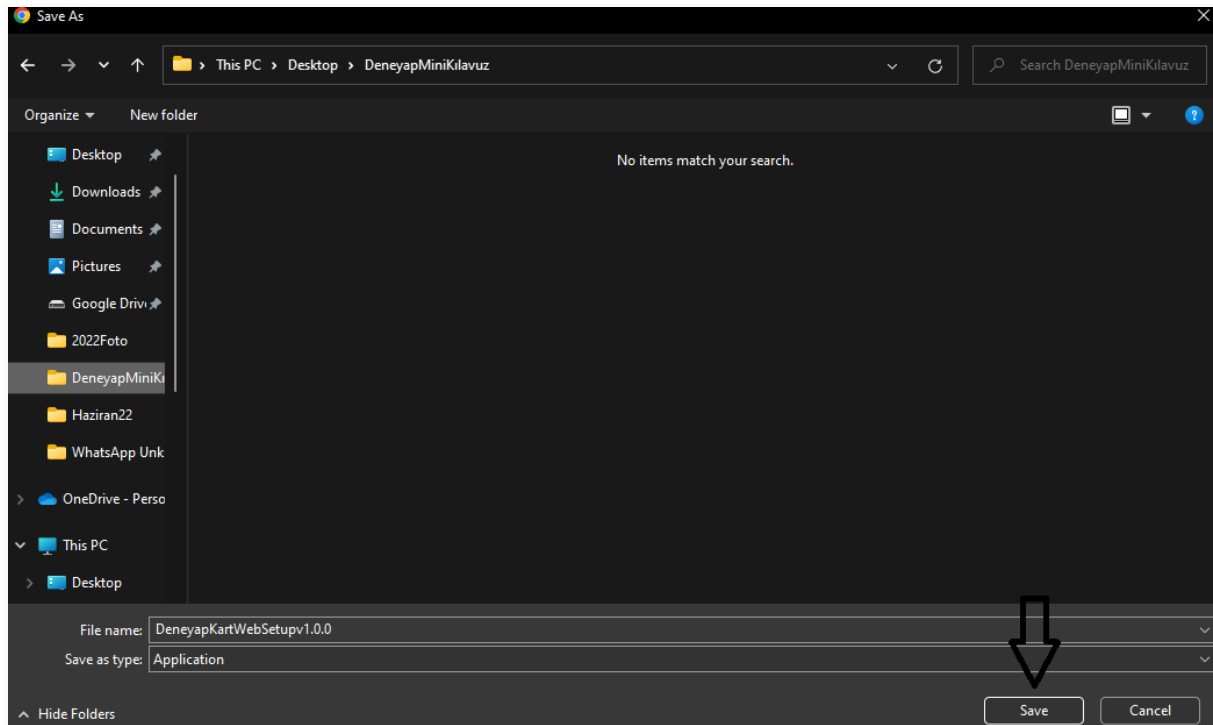


1- Deneyap Kart Blok sayfasına geçiş yapıldığında “Deneyap Kart Web İndir” butonuna tıklanılır.



Şekil 5

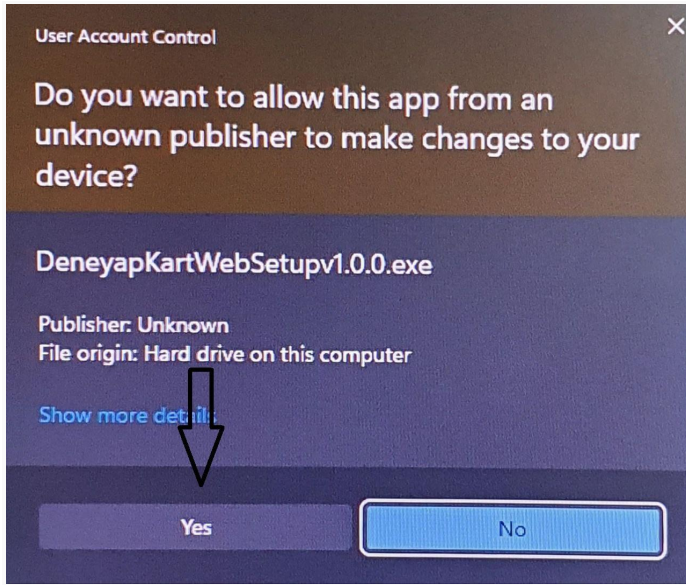
2- Açılan pencerede “Save” butonuna tıklayıp, programı bilgisayarınıza indiriniz.



Şekil 6

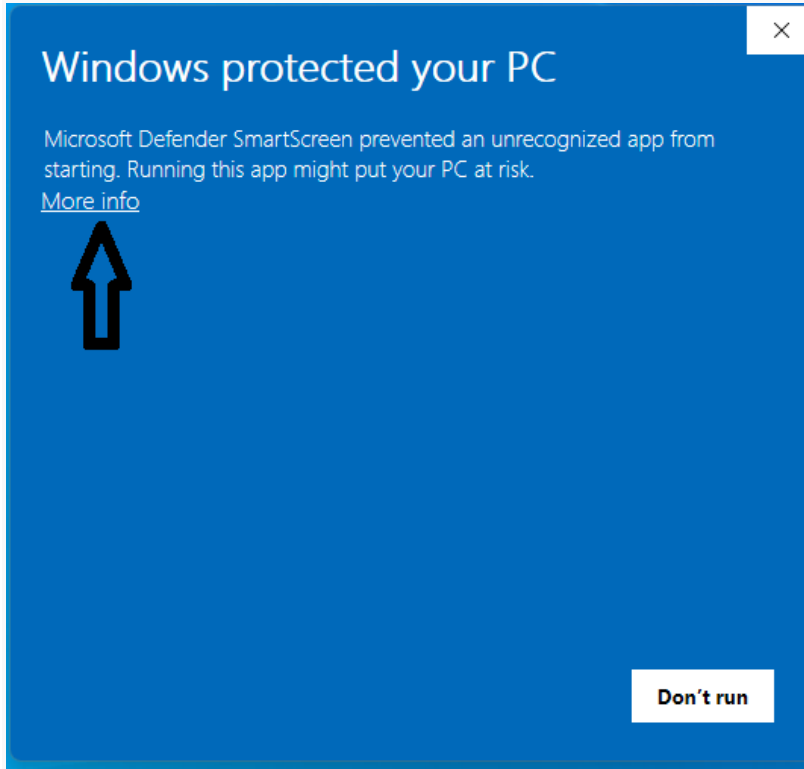


3- İndirme işlemi tamamlandıktan sonra kurulum işlemini başlatıyoruz.



Şekil 7

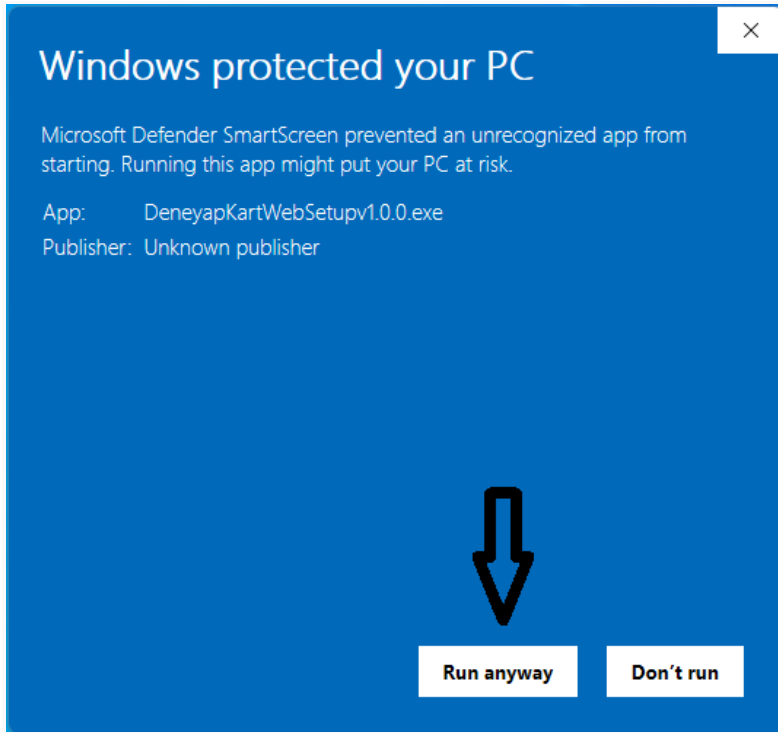
4- Kurulum başlatma sırasında Windows Defender vb. programlar aşağıdaki şekilde uyarı verebilir. Bu şekilde bir ekran ile karşılaşırsanız "More info" yazısına tıklayınız.



Şekil 8

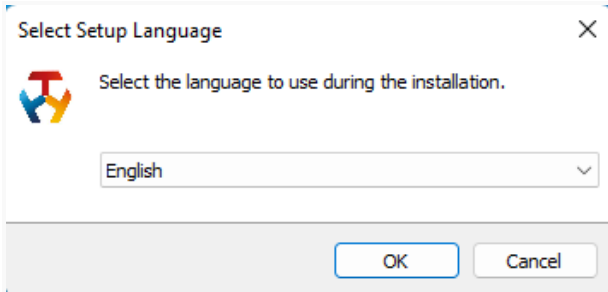


5- Yükleme adımına geçmek için “Run anyway” butonuna tıklayınız.



Şekil 9

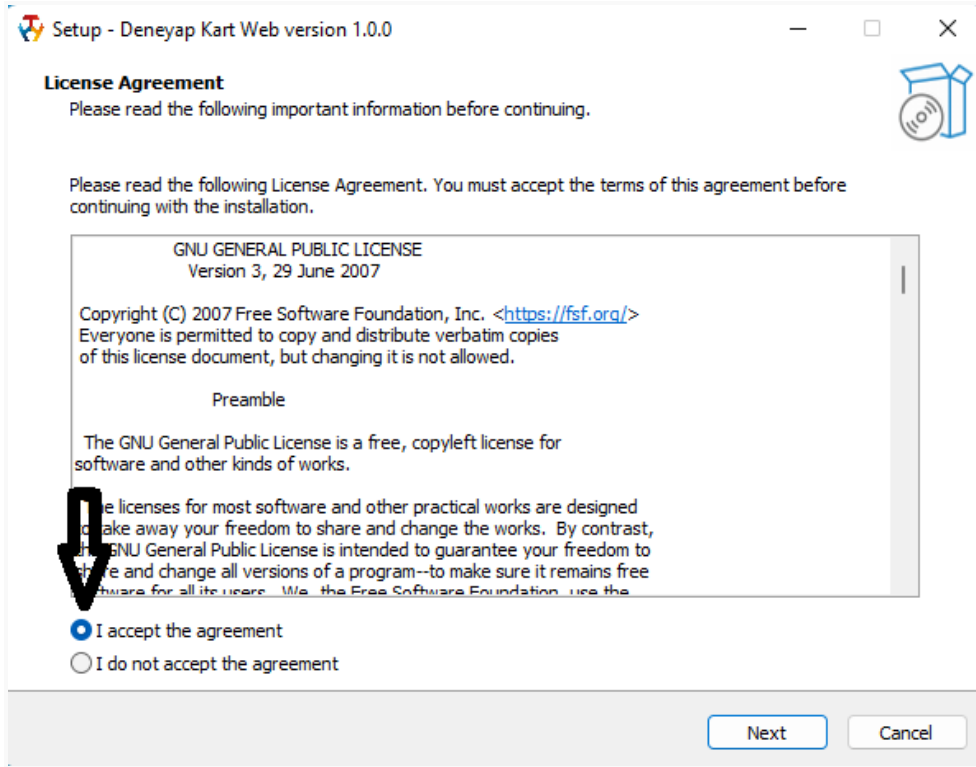
6- Yüklemenin ilk adımında dil seçimi yapmanız istenir. Dil seçimi yapıp “OK” butonuna tıklayınız.



Şekil 10

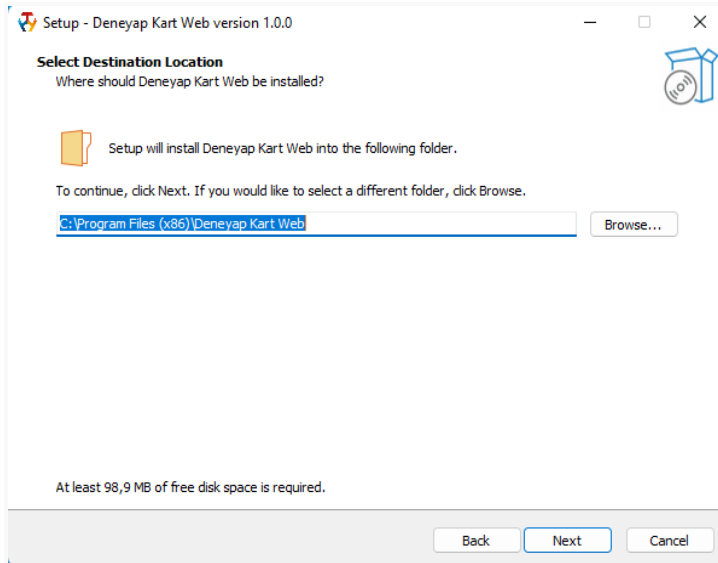


7- Ardından gelen ekranda lisans anlaşmasını onayladıktan sonra “Next” butonuna tıklayınız.



Şekil 11

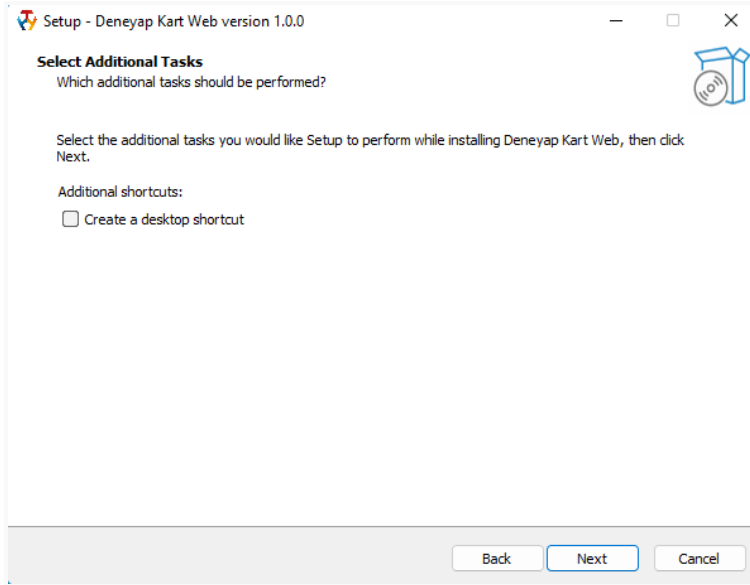
8- Gelen ekranda kurulumun nereye yapılacağı gösterilmektedir. “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 12

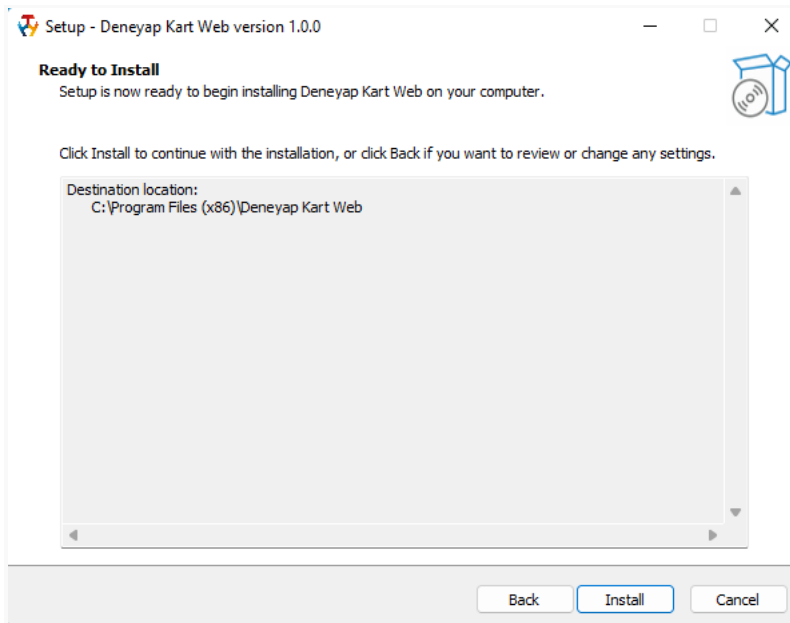


9- Bir sonraki sayfada masaüstüne kısayol oluşturulması ile ilgili seçiminizi yapıp, “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 13

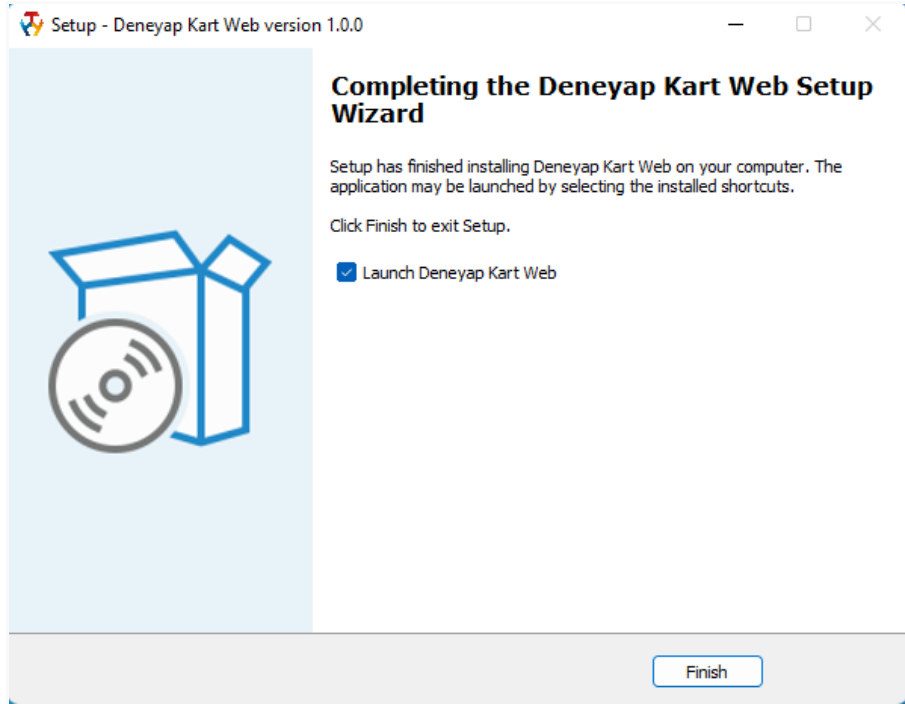
10- Gelen ekranda “Install” butonuna tıklayıp yüklemeyi başlatabilirsiniz.



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.



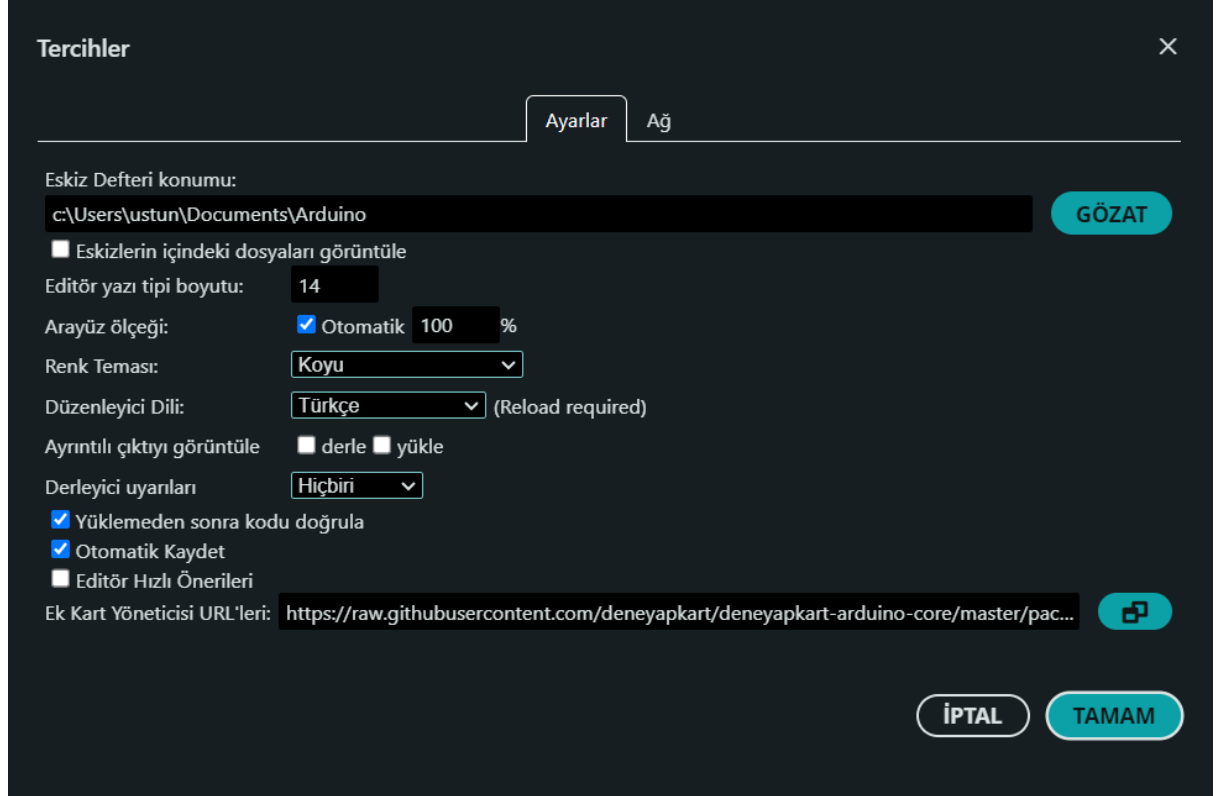
Şekil 16



C) Arduino IDE Kullanacaksanız: Deneyap Kart Nasıl Eklenir?

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

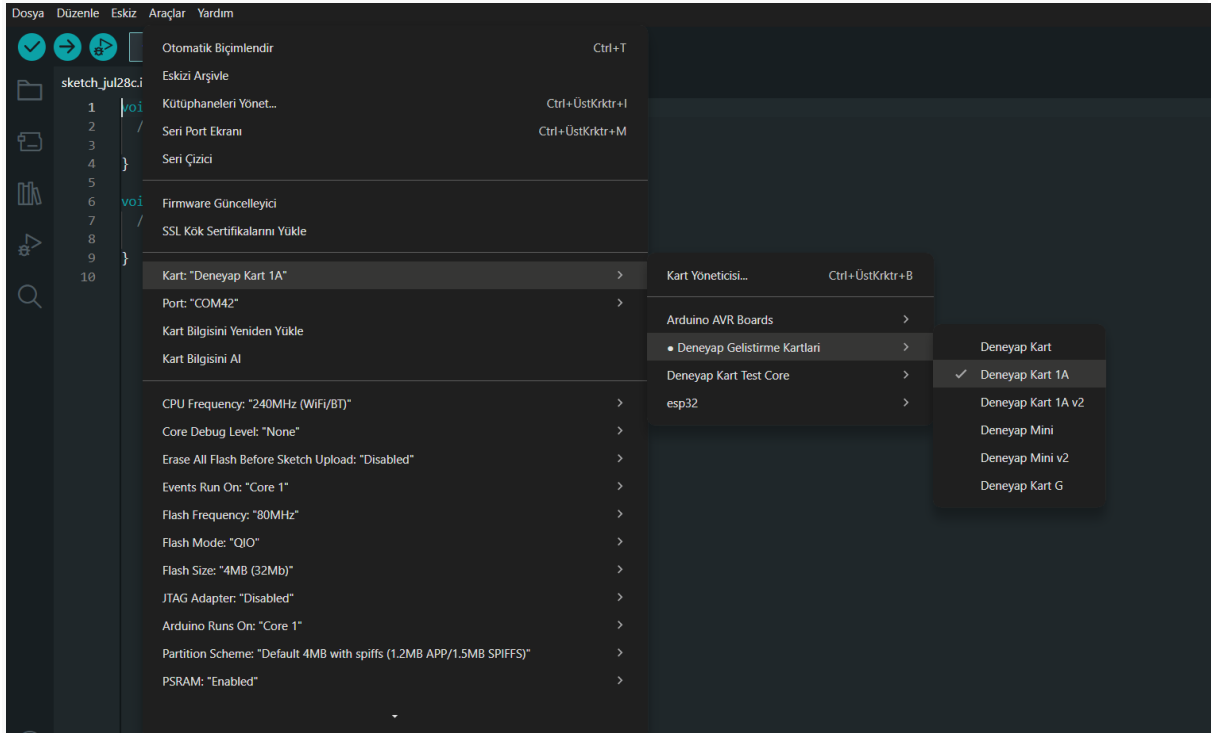


2- Ardından, Araçlar (Tools) → Kart (Board) → Kart Yöneticisi (Boards Manager) adımını takip ederek gelen ekranda arama satırına “Deneyap Gelistirme Kartlari” yazınız. En son sürüm varsayılan olarak seçilmiş halde gelir ve Kur (Install) butonuna tıklayarak yüklemeyi gerçekleştirebilirsiniz.



Şekil 18

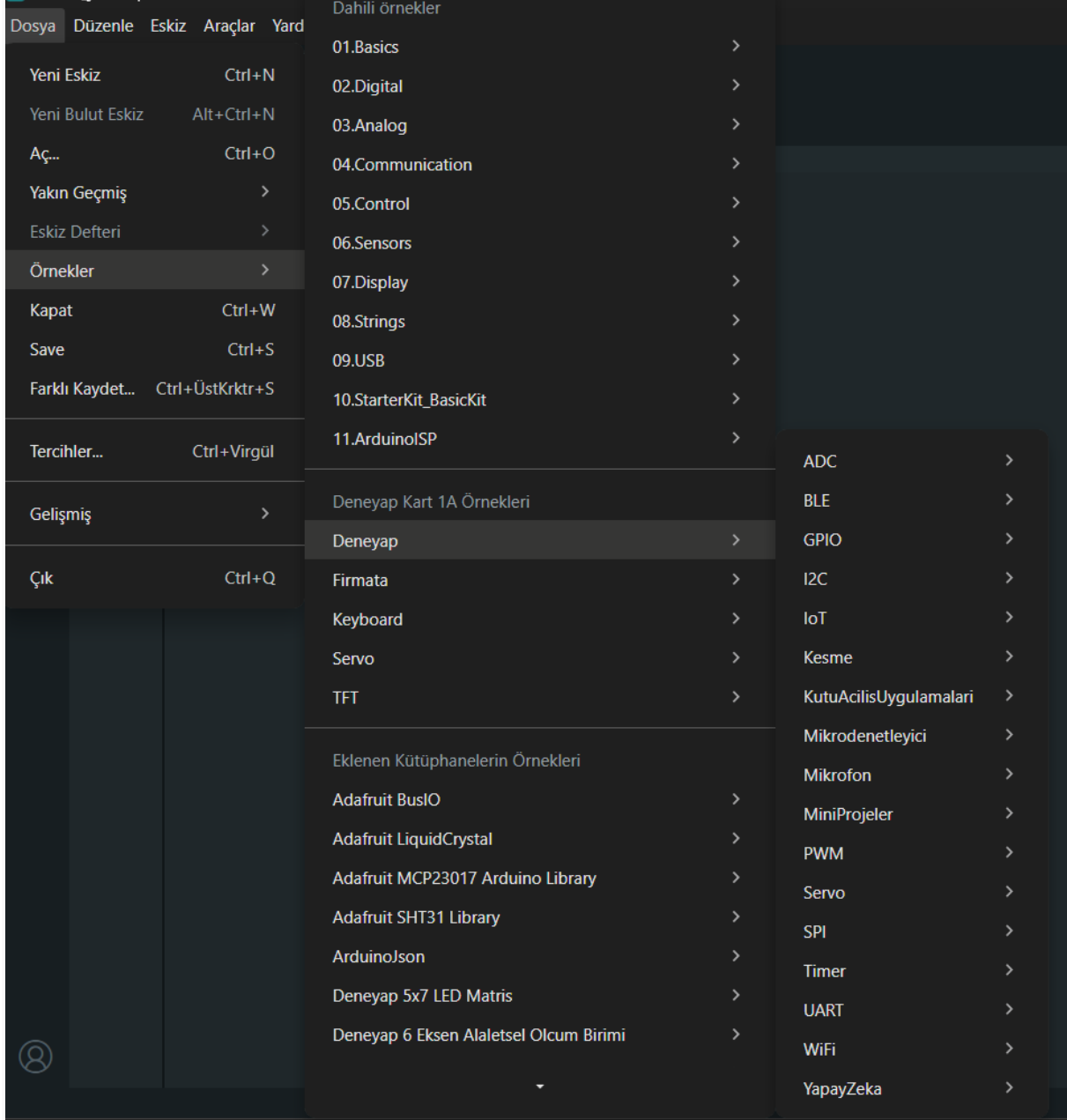
3- En son aşama olarak Araçlar (Tools) → Kart (Board) adımından “Deneyap Kart” ve kartınızın bağlı olduğu Port’u seçerek kodlama adımına geçebilirsiniz.



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

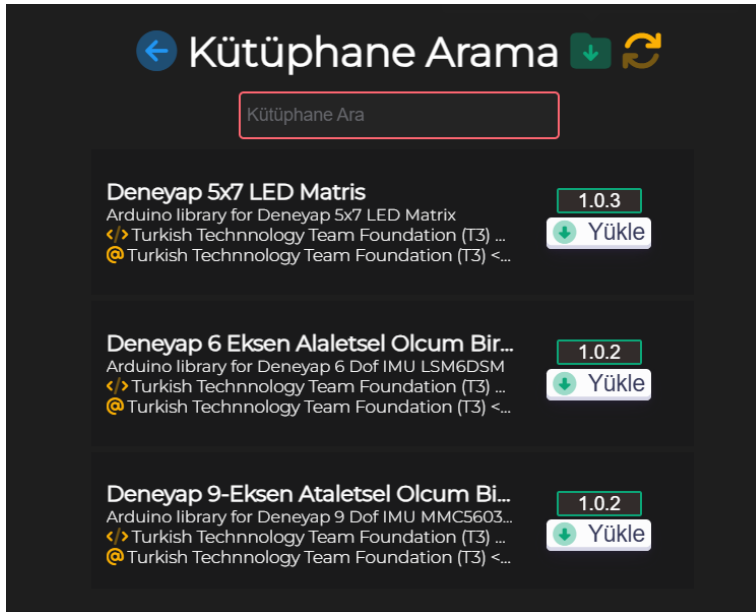
3. İlgili Kütüphaneyi Seçin ve Kurun

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.

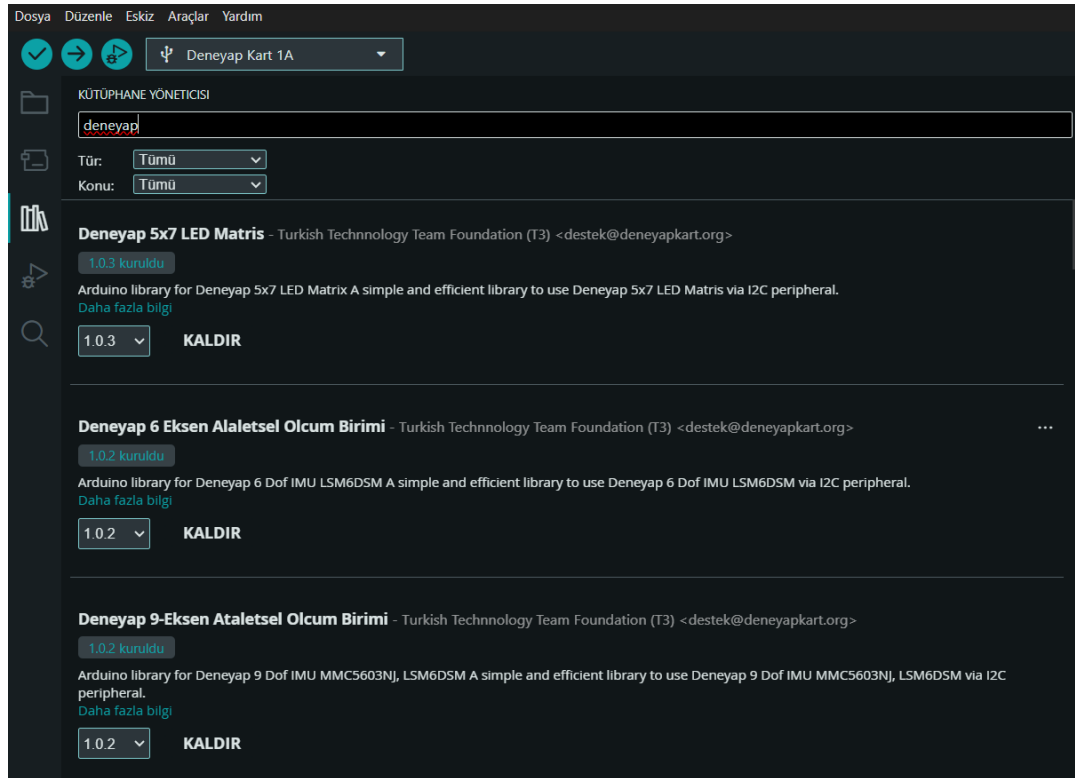
3.Üstteki arama kutusuna deneyap yazın.

- 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



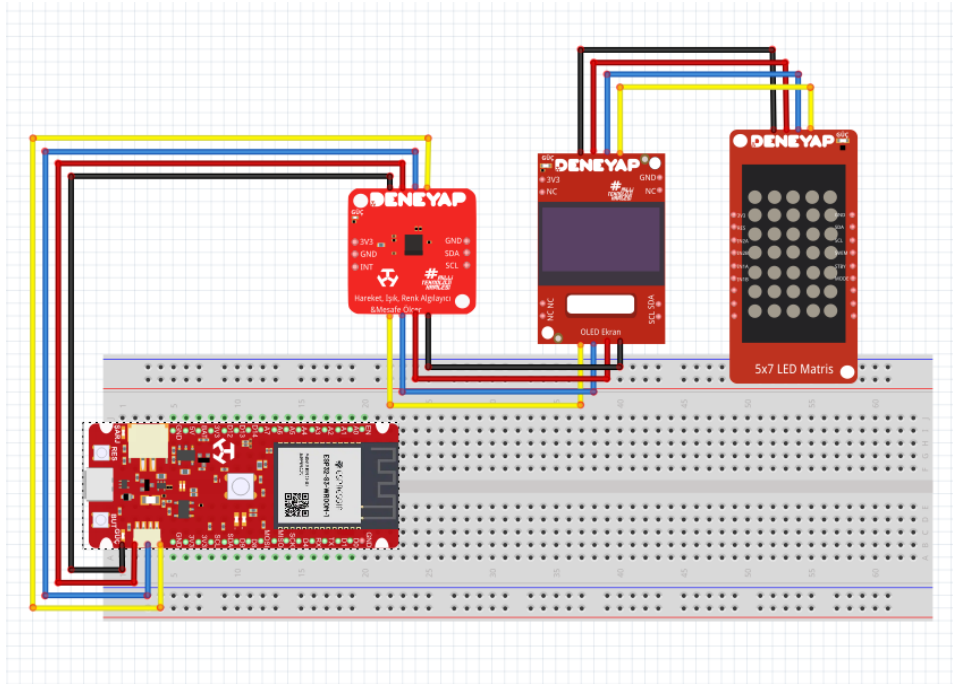
Proje Özeti: Bu proje, bir odada hareket algılandığında görsel uyarılar vererek kullanıcının bilgilendirilmesini sağlayan bir akıllı oda kontrol sistemidir. Sistem **Deneyap Kart 1Av2** ile birlikte **Deneyap Hareket, Işık, Renk Algılayıcı ve Mesafe Ölçersensörü**, **OLED Ekran**, **5x7 LED Matris**, **RGB LED** ve **Gerçek Zamanlı Saat (RTC) modülü** kullanılarak oluşturulmuştur. Amaç, özellikle güvenlik ve enerji tasarrufu gerektiren alanlarda ortamın aktif olup olmadığını izlemek ve anlık olarak kullanıcıya bildirimde bulunmaktır. Sistem çalıştığında, tüm modüller başlatılır ve OLED ekrana “Oda Kontrol” mesajı yazılır. Oda içinde bir hareket algılandığında, OLED ekran “Hareket Algılandı” mesajıyla birlikte gerçek zamanlı saat bilgisini (saat, dakika ve saniye) gösterir. Aynı anda 5x7 LED matris üzerinde bir uyarı simgesi görüntülenir ve RGB LED farklı renklerde yanıp söner. Hareketin algılandığı zaman **millis()** fonksiyonu ile kaydedilir. Eğer 10 saniye boyunca herhangi bir hareket algılanmazsa, sistem RGB LED’i kapatır ve OLED ekrana “Hareketsiz...” mesajı yazar. Bu sayede sistem, hem hareketli durumlarda hem de hareketsizlikte kullanıcıya ekran ve ışık yoluyla bilgi verir.

Gerekli Malzemeler:

- Bir adet Deneyap Kart 1A v2
- Bir adet Deneyap Hareket, Işık, Renk Algılayıcı ve Mesafe Ölçer
- Bir adet Deneyap OLED Ekran
- Bir adet Deneyap 5x7 LED Matris
- RGB LED (Deneyap Kart üzerindeki dahili LED)
- Üç adet I2C kablosu

Devre Şeması:

Devre şeması aşağıdaki gibi kurulabilir.



Şekil 23

Proje Kodları:



İLERİ SEVİYE EĞİTİM

© 2025 Deneyap Kart

Bu bölümde, projede kullanılacak Deneyap modüllerine ait kütüphaneler programa dahil edilir:

- **#include <Deneyap_GercekZamanliSaat.h>** satırı, gerçek zamanlı saat (RTC) modülünün kullanılabilmesi için gerekli kütüphaneyi projeye ekler. Bu sayede tarih ve saat bilgileri alınabilir.
- **#include <Deneyap_HareketIsikRenkAlgilayiciMesafeOlcer.h>** satırı, çoklu algılama yeteneğine sahip sensör modülünü (hareket, ışık, renk ve mesafe) projeye dahil eder.
- **#include <Deneyap_OLED.h>** satırı, OLED ekran modülünün kullanılmasını sağlar. Bu ekran, verilerin görsel olarak kullanıcıya sunulmasına olanak tanır.
- **#include <Deneyap_5x7LedEkran.h>** satırı, 5x7 LED matris ekran modülünü projeye dahil eder. Bu ekran ile karakter bazlı görsel geri bildirim sağlanabilir.

Ayrıca:

- **#define sure 400** satırı, programda **sure** adında bir sabit tanımlar ve bu sabite **400** değeri atanır. Genellikle gecikme süresi veya zamanlayıcı işlemleri için kullanılır. Sabit tanımlama, kodun okunabilirliğini ve bakımını kolaylaştırır.

```
#include <Deneyap_GercekZamanliSaat.h>

#include <Deneyap_HareketIsikRenkAlgilayiciMesafeOlcer.h>

#include <Deneyap_OLED.h>

#include <Deneyap_5x7LedEkran.h>

#define sure 400

RTC GercekZamanliSaat;
```

Şekil 24

Bu satırlarda, daha önce eklenen kütüphanelere karşılık gelen modülleri kullanabilmek için gerekli nesneler oluşturulmuştur:

- **APDS9960 Hareket;**
Bu satır, **Deneyap Hareket-Işık-Renk-Mesafe Algılayıcı modülünü** temsil eden bir nesne tanımlar. Bu nesne sayesinde hareket algılama, renk okuma, mesafe ölçme gibi işlemler yapılabilir.



- **OLED oled;**
OLED ekran modülünü temsil eden **oled** nesnesi oluşturulur. Bu nesne, ekrana yazı yazma, ekranı temizleme gibi işlemleri gerçekleştirmek için kullanılır.
- **DotMatrix ledMatrix;**
5x7 LED matris ekranını kontrol etmek amacıyla **ledMatrix** nesnesi tanımlanır. Bu nesne sayesinde LED ekranda karakterler veya şekiller gösterilebilir.
- **RTC rtc;**
Gerçek zamanlı saat (RTC) modülünü temsil eden **rtc** nesnesi tanımlanır. Bu sayede tarih ve saat bilgisi okunabilir veya ayarlanabilir.

Bu nesneler, programın ilerleyen bölümlerinde ilgili donanımları kontrol etmek için kullanılacaktır.

```
APDS9960 Hareket;

OLED oled;

DotMatrix ledMatrix;

RTC rtc;
```

Şekil 25

unsigned long sonHareketZamani = 0; satırı, en son hareketin algılandığı zamanı milisaniye cinsinden tutmak için bir değişken tanımlar. Başlangıçta sıfır olarak ayarlanır.

const unsigned long beklemeSuresi = 10000; satırı ise sabit bir süre belirler. Bu süre 10.000 milisaniye yani 10 saniyedir. Bu değer, hareket algılandıktan sonra sistemin ne kadar bekleyeceğini ifade eder.

```
unsigned long sonHareketZamani = 0;
const unsigned long beklemeSuresi = 10000;
```

Şekil 26

uyari adında tanımlanan dizi, **5 sütun ve 7 satırdan oluşan** bir LED matris ekranında gösterilecek olan görsel deseni temsil eder.

Her bir değer (**1** veya **0**), ilgili LED'in **yanıp yanmayacağını** belirler:

- **1** → LED yanar



- 0 → LED sönük kalır

Dizideki değerler, **soldan sağa, yukarıdan aşağıya** doğru sıralanır.

1. **satır:** Sadece kenarlardaki LED'ler yanar → (1, 0, 0, 0, 1)
2. **satır:** Ortası boş olacak şekilde iki yandaki LED'ler yanar → (0, 1, 0, 1, 0)
3. **satır:** Sadece ortadaki LED yanar → (0, 0, 1, 0, 0)
4. **satır:** 2. satırla aynıdır → (0, 1, 0, 1, 0)
5. **satır:** 1. satırla aynıdır → (1, 0, 0, 0, 1)
6. **ve 7. satırlar:** Tüm LED'ler sönüktür → (0, 0, 0, 0, 0)

Bu desen, **merkezde dikkat çekici bir şekil** oluşturarak genellikle **uyarı, dikkat** veya **önemli durum** gibi görsel göstergeler için kullanılır.

```
const byte uyari[35] = {
  1,0,0,0,1,
  0,1,0,1,0,
  0,0,1,0,0,
  0,1,0,1,0,
  1,0,0,0,1,
  0,0,0,0,0,
  0,0,0,0,0
};
```

Şekil 27

Programın başlangıcında çeşitli modüllerin çalışması için gerekli başlatmalar yapılır:

Seri Haberleşme Başlatma:

- **Serial.begin(115200);** → Seri monitör üzerinden bilgi aktarımı sağlanır.
- **delay(1000);** → Başlangıçta sistemin oturması için 1 saniyelik gecikme uygulanır.

RGB LED Ayarı:

- **pinMode(RGBLED, OUTPUT);** → RGB LED'in sinyal alabilmesi için çıkış (output) olarak ayarlanır.

Deneyap Hareket, Işık, Renk Algılayıcı ve Mesafe Ölçer Sensörü (APDS9960):

- **Hareket.begin(0x39);** ile I2C adresi üzerinden bağlantı kurulur.
- Başarısızsa hata mesajı verilir ve program sonsuz döngüye girer.
- Başarılıysa **Hareket.enableGesture(true);** ile hareket algılama modu aktif hale getirilir.

OLED Ekran:

- **oled.begin(0x7A);** ile ekran başlatılır.



- Başarılıysa:
 - `oled.init();` → Ekran sıfırlanır.
 - `oled.clearDisplay();` → Ekran temizlenir.
 - `oled.setTextXY(0, 0);` → Yazı pozisyonu ayarlanır.
 - "Oda Kontrol" ifadesi yazdırılır.

LED Matris:

- `ledMatrix.begin(0x0A);` → Modül başlatılır.
- Başarısızsa hata verilip durdurulur.
- Başarılıysa `ledMatrix.resetDotRows();` ile daha önceki tüm veriler temizlenir.

RTC (Gerçek Zamanlı Saat):

- `rtc.begin();` → RTC başlatılır.
- `rtc.deviceStart();` ve `rtc.adjust();` komutlarıyla saat çalıştırılır ve zaman ayarı yapılır.

Hazır Bildirimi:

- Tüm modüller doğru başlatılırsa, "**Sistem hazır.**" mesajı seri monitöre yazdırılır.

```
void setup() {
  Serial.begin(115200);
  delay(1000);

  pinMode(RGBLED, OUTPUT);

  if (!Hareket.begin(0x39)) {
    Serial.println("Hareket sensörü I2C bağlantısı başarısız!");
    while (1);
  }
  Hareket.enableGesture(true);

  if (!oled.begin(0x7A)) {
    Serial.println("OLED modülü başlatılamadı!");
    while (1);
  }
  oled.init();
  oled.clearDisplay();
  oled.setTextXY(0, 0);
  oled.putString("Oda Kontrol");

  if (!ledMatrix.begin(0x0A)) {
```



```

Serial.println("LED Matris modülü başlatılamadı!");
while (1);
}
ledMatrix.resetDotRows();

if (!rtc.begin()) {
    Serial.println("RTC modülü başlatılamadı!");
    while (1);
}
rtc.deviceStart();
rtc.adjust();

Serial.println("Sistem hazır.");
}

```

Şekil 28

Programın ana döngüsü olan **loop()** fonksiyonunda, ilk olarak **hareket algılayıcıdan gelen veri kontrol edilir**.

Eğer herhangi bir hareket algılanmışsa (**hareket != 0**), aşağıdaki işlemler sırasıyla gerçekleştirilir:

- **OLED ekran temizlenir** ve ekrana “**Hareket Algılandı.**” mesajı yazılır.
- **RTC modülünden** alınan saat, dakika ve saniye bilgisi ekranda gösterilir. Saat bilgileri tek haneli ise başına sıfır eklenir (örneğin 08:04:05 gibi), böylece daha okunabilir bir format sağlanır.
- **sonHareketZamani değişkenine**, o anki zaman (**millis()** değeri) atanır. Bu sayede en son hareketin ne zaman gerçekleştiği kaydedilmiş olur.
- **LED matris sıfırlanır** ve üzerinde bir “uyarı” deseni gösterilir. Bu desen, daha önce tanımlanmış olan **uyari[]** dizisinden alınır.
- **RGB LED**, sırayla farklı renklerde yanıp sönmek görsel bir uyarı verir. Her renkten sonra kısa bir süre beklenir (**delay(sure)**), böylece ışık geçiş efekti sağlanır.

Eğer hareket algılanmazsa:

- Program, **en son hareketin üzerinden belirli bir sürenin geçip geçmediğini kontrol eder**. Bu süre, tanımlı **beklemeSuresi** (örneğin 10 saniye) ile karşılaştırılır.
- Eğer bu süre geçmişse:
RGB LED kapatılır.
 OLED ekranda “**Hareketsiz...**” mesajı gösterilir.



```
void loop() {  
  
    int hareket = Hareket.readGesture();  
  
    if (hareket != 0) {  
  
        DateTime simdi = rtc.now();  
  
        oled.clearDisplay();  
  
        oled.setTextXY(0, 0);  
  
        oled.putString("Hareket Algilandi.");  
  
  
        oled.setTextXY(1, 0);  
  
        oled.putString("Saat: ");  
  
        if (simdi.hour() < 10) oled.putString("0");  
  
        oled.putInt(simdi.hour());  
  
        oled.putString(":");  
  
        if (simdi.minute() < 10) oled.putString("0");  
  
        oled.putInt(simdi.minute());  
  
        oled.putString(":");  
  
        if (simdi.second() < 10) oled.putString("0");  
  
        oled.putInt(simdi.second());  
  
  
        sonHareketZamani = millis();  
  
  
        ledMatrix.resetDotRows();  
  
        ledMatrix.drawLedMatrix(uyari, 500);  
  
    }  
}
```



```

    neopixelWrite(RGBLED, 64, 0, 0);

    delay(sure);

    neopixelWrite(RGBLED, 0, 64, 0);

    delay(sure);

    neopixelWrite(RGBLED, 0, 0, 64);

    delay(sure);

    neopixelWrite(RGBLED, 0, 0, 0);

    delay(sure);

    neopixelWrite(RGBLED, 64, 64, 64);

    delay(sure);

    neopixelWrite(RGBLED, 64, 64, 0);

    delay(sure);

    neopixelWrite(RGBLED, 64, 0, 64);

    delay(sure);

}

else {

    if (millis() - sonHareketZamani > beklemeSuresi) {

        neopixelWrite(RGBLED, 0, 0, 0);

        oled.clearDisplay();

        oled.setTextXY(0, 0);

        oled.putString("Hareketsiz...");

    }

}

}

```



Şekil 29

