

DENEYAP KART 1A v2 İLE

ODA HAVASI KALİTE MONİTÖRÜ

A) Deneyap Kart IDE Kurulumu:

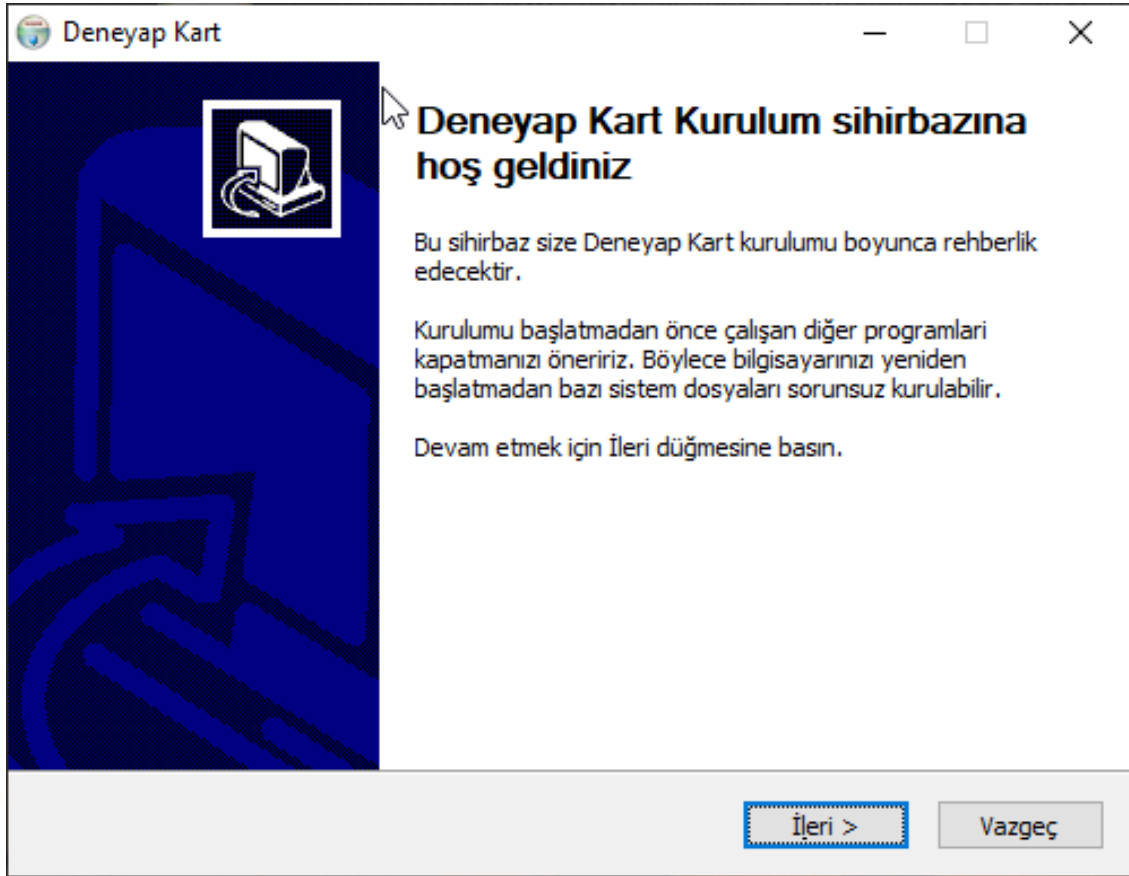
Deneyap Kart IDE, metin tabanlı programlama yapmanız için geliştirilmiş bir programlama ortamıdır. Eğer metinsel kodlama yapmak istiyorsanız, Deneyap Kart IDE'yi kullanmanız gerekir.

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.

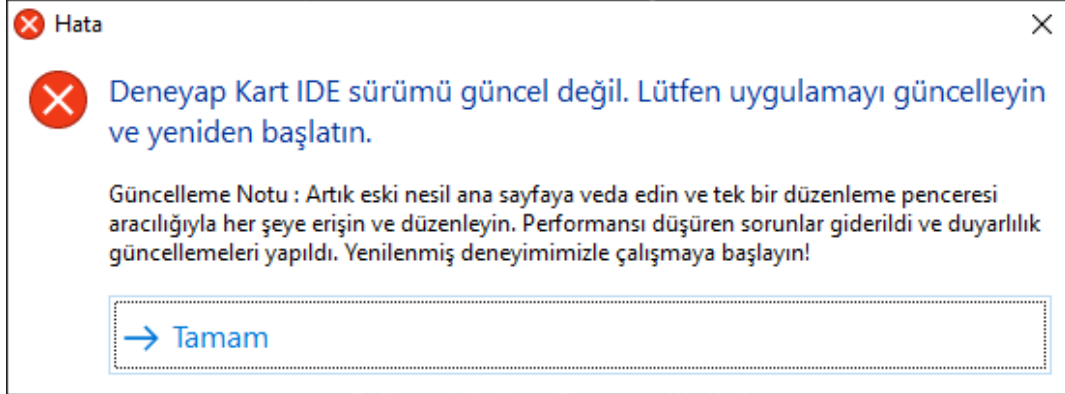


Şekil 1



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.

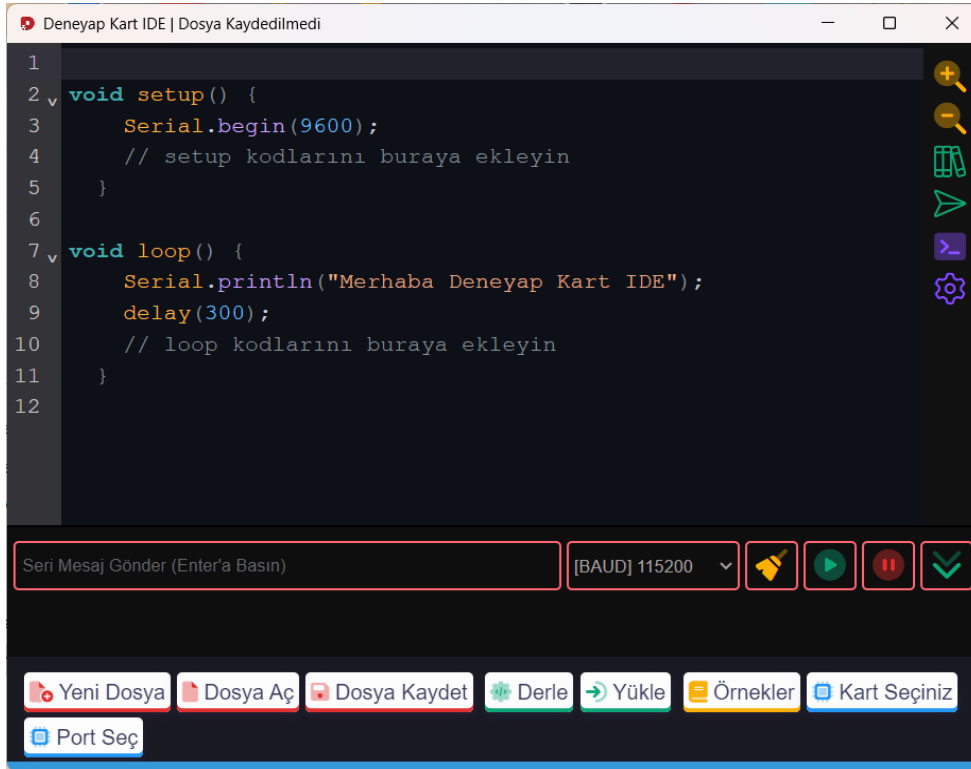


Şekil 3



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.

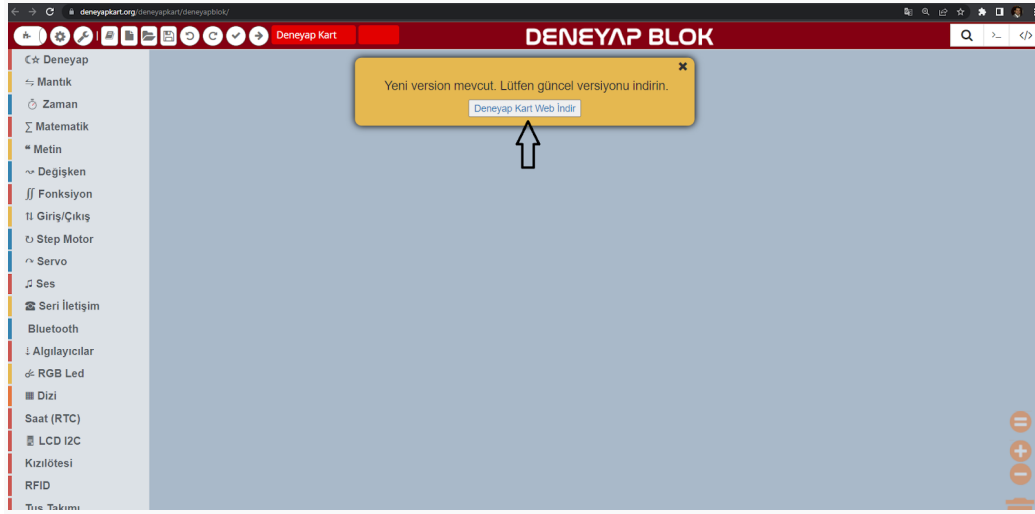
B) DENEYAP Kart Blok Web Kurulumu:

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.

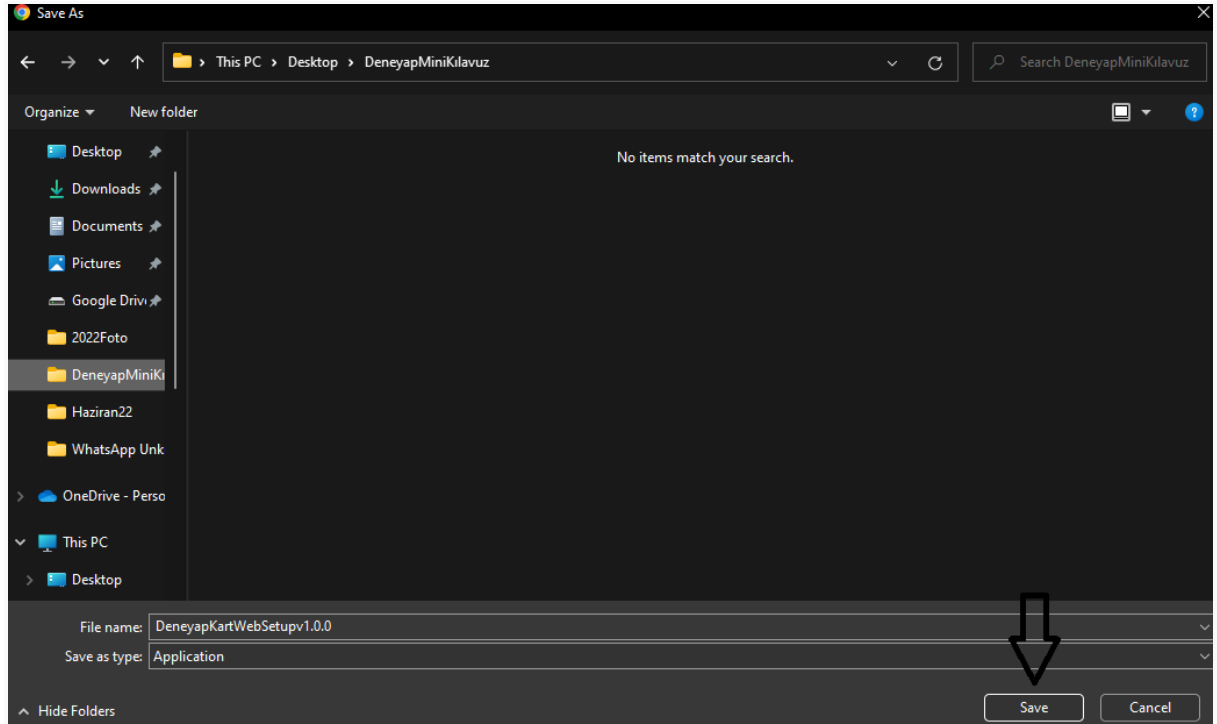


1- Deneyap Kart Blok sayfasına geçiş yapıldığında “Deneyap Kart Web İndir” butonuna tıklanılır.



Şekil 5

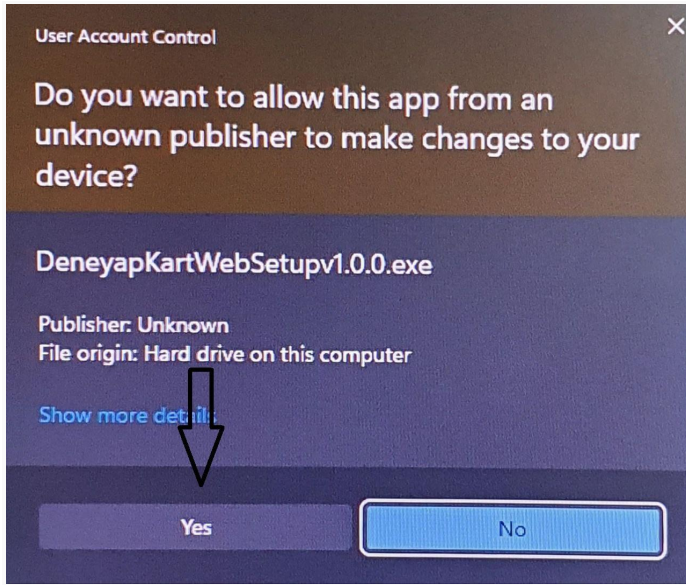
2- Açılan pencerede “Save” butonuna tıklayıp, programı bilgisayarınıza indiriniz.



Şekil 6

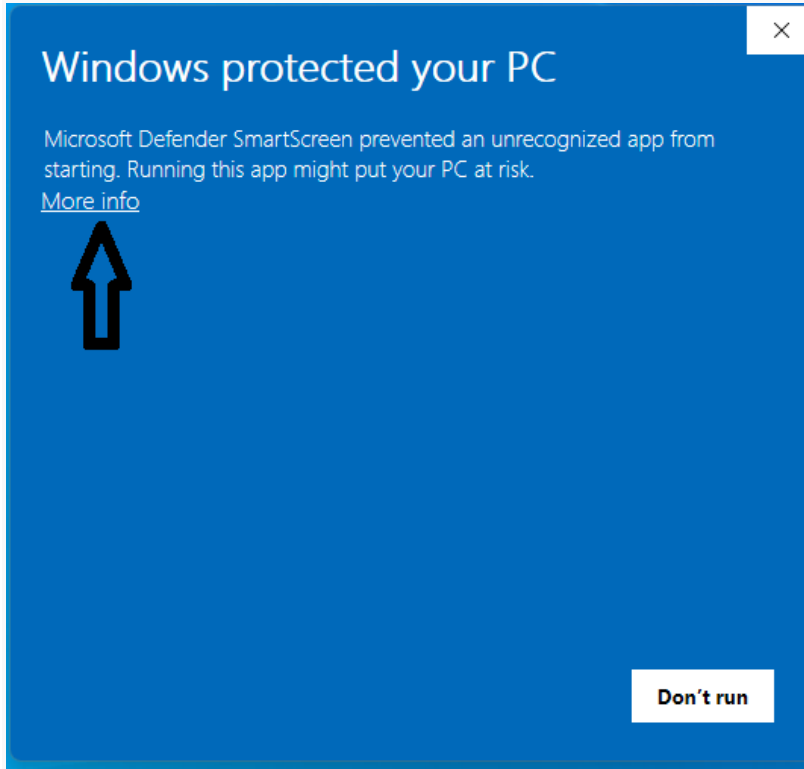


3- İndirme işlemi tamamlandıktan sonra kurulum işlemini başlatıyoruz.



Şekil 7

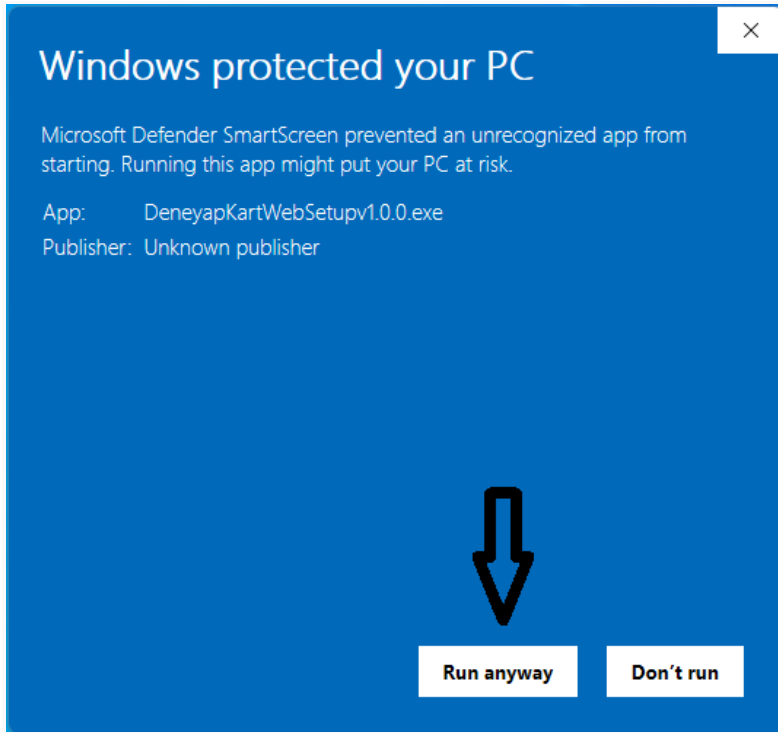
4- Kurulum başlatma sırasında Windows Defender vb. programlar aşağıdaki şekilde uyarı verebilir. Bu şekilde bir ekran ile karşılaşırsanız "More info" yazısına tıklayınız.



Şekil 8

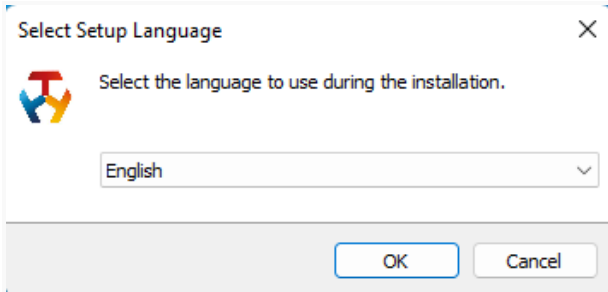


5- Yükleme adımına geçmek için “Run anyway” butonuna tıklayınız.



Şekil 9

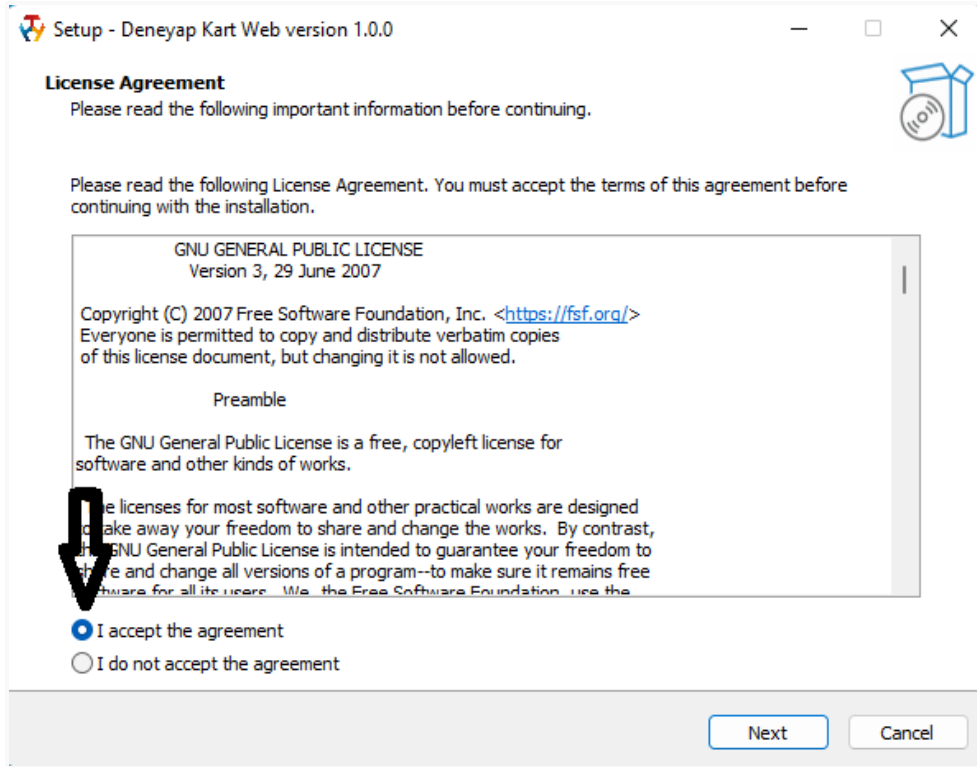
6- Yüklemenin ilk adımında dil seçimi yapmanız istenir. Dil seçimi yapıp “OK” butonuna tıklayınız.



Şekil 10

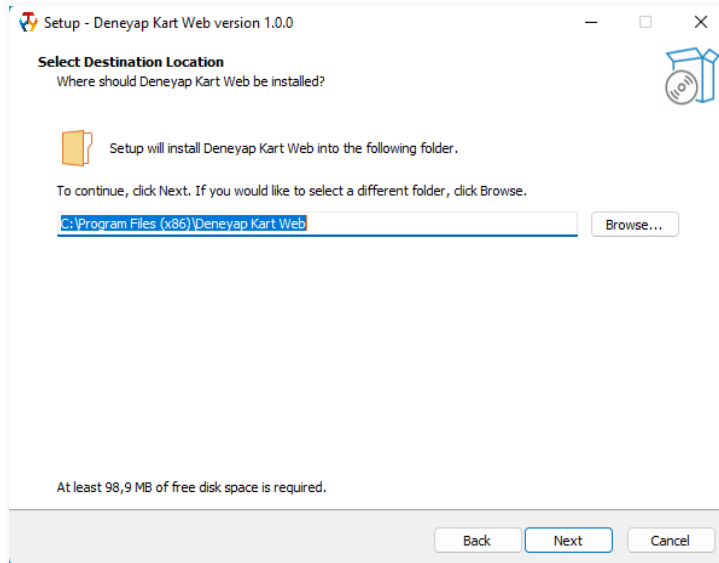


7- Ardından gelen ekranda lisans anlaşmasını onayladıktan sonra “Next” butonuna tıklayınız.



Şekil 11

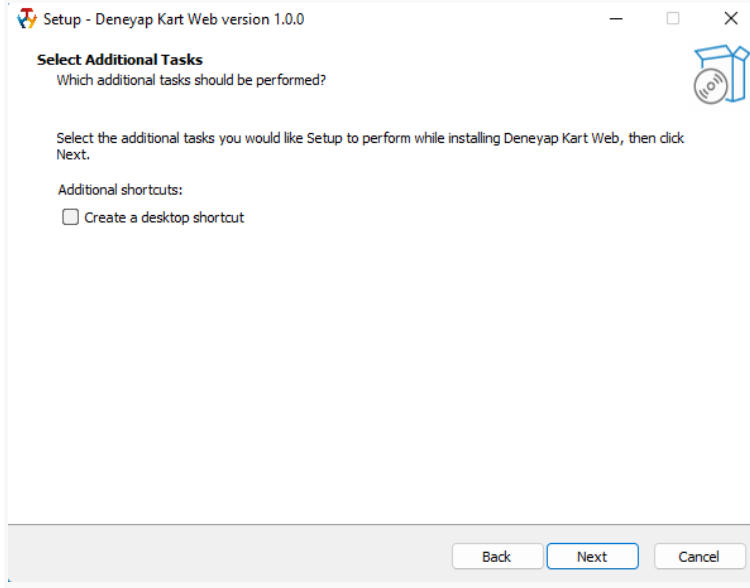
8- Gelen ekranda kurulumun nereye yapılacağı gösterilmektedir. “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 12

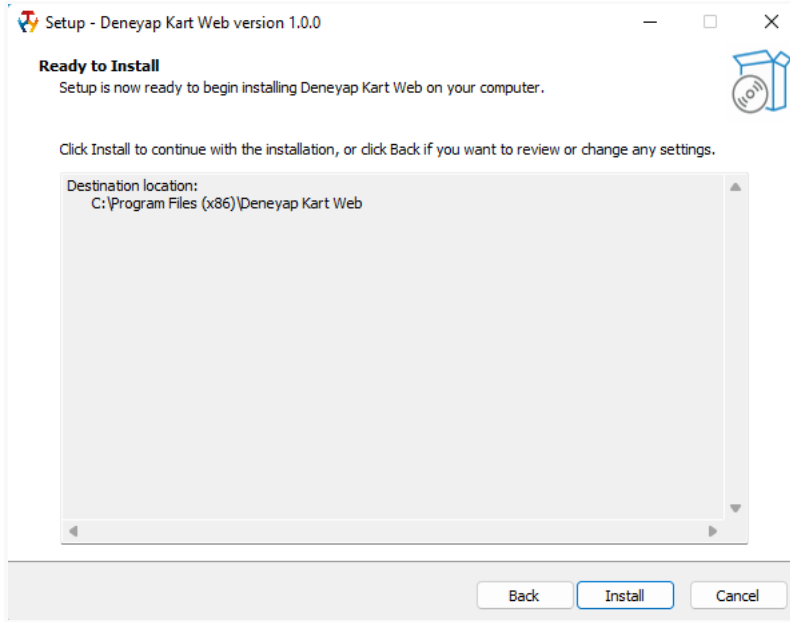


9- Bir sonraki sayfada masaüstüne kısayol oluşturulması ile ilgili seçiminizi yapıp, “Next” butonuna tıklayıp yüklemeye devam edebilirsiniz.



Şekil 13

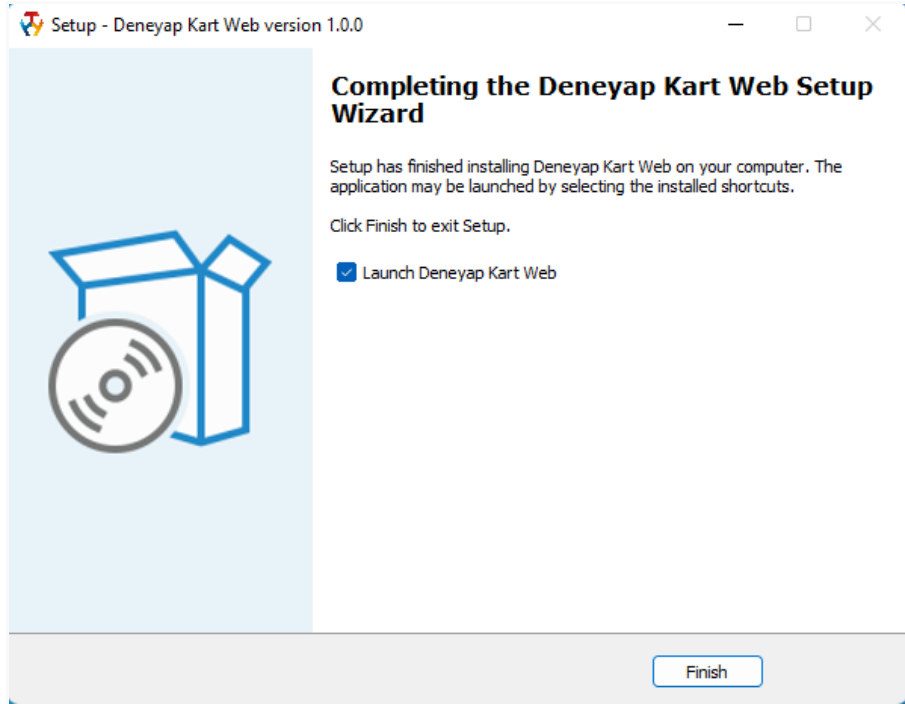
10- Gelen ekranda “Install” butonuna tıklayıp yüklemeyi başlatabilirsiniz.



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

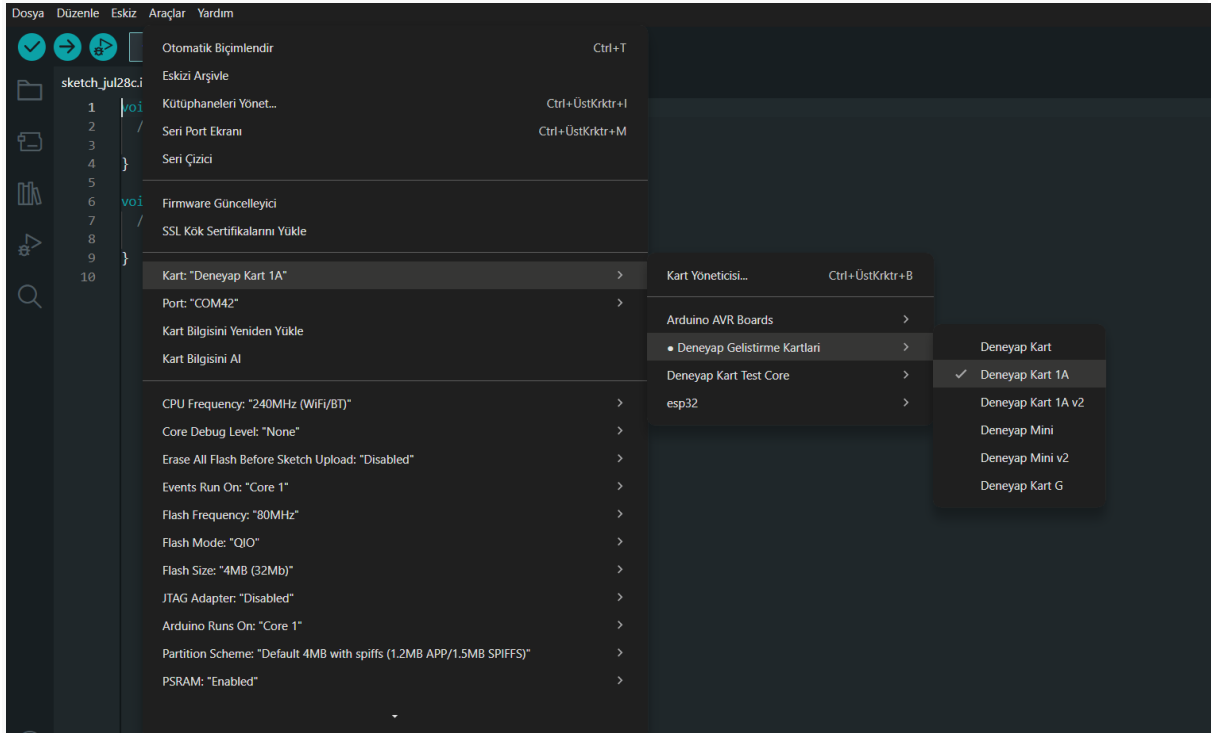


2- Ardından, Araçlar (Tools) → Kart (Board) → Kart Yöneticisi (Boards Manager) adımını takip ederek gelen ekranda arama satırına “Deneyap Gelistirme Kartlari” yazınız. En son sürüm varsayılan olarak seçilmiş halde gelir ve Kur (Install) butonuna tıklayarak yüklemeyi gerçekleştirebilirsiniz.



Şekil 18

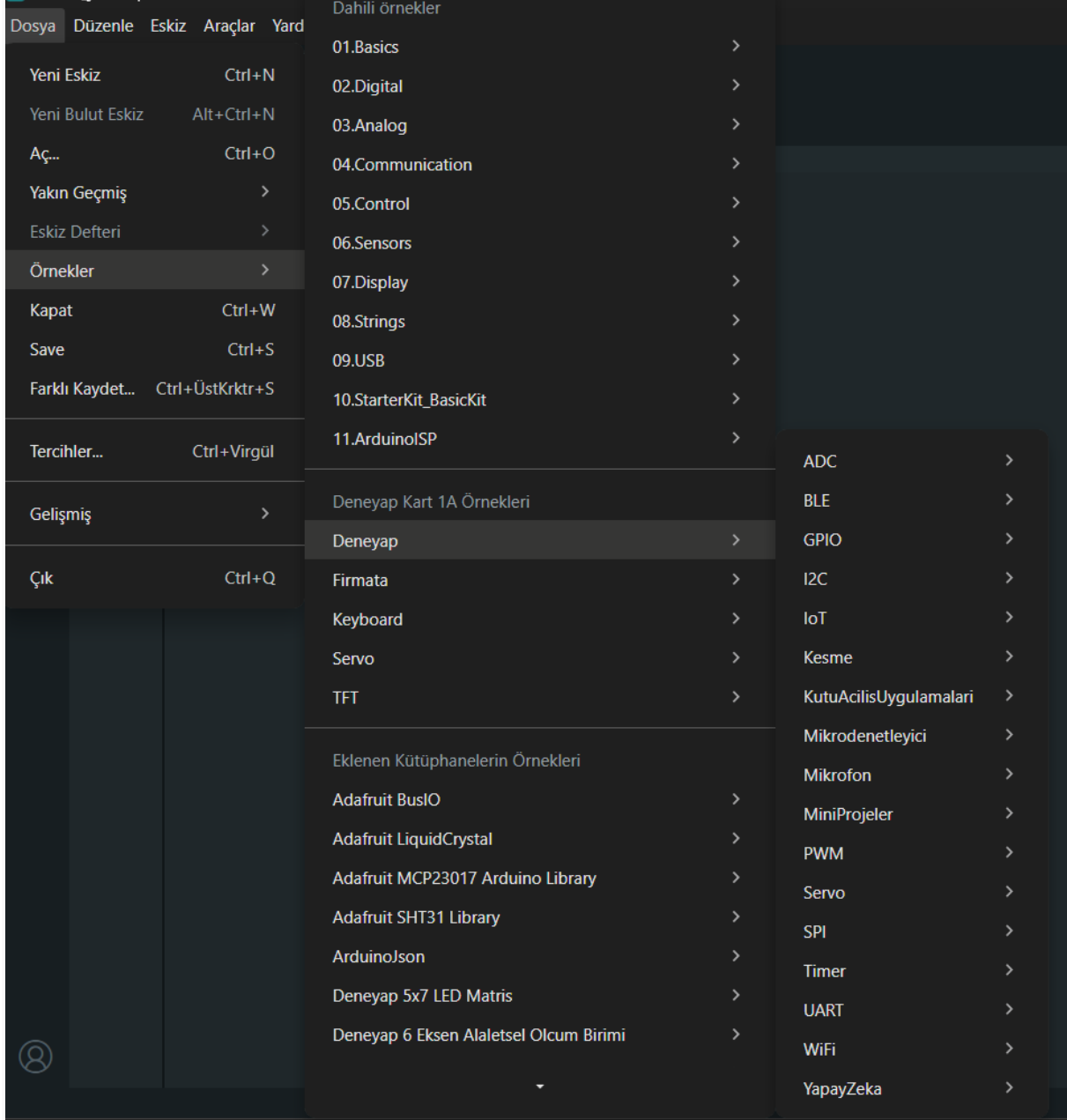
3- En son aşama olarak Araçlar (Tools) → Kart (Board) adımından “Deneyap Kart” ve kartınızın bağlı olduğu Port’u seçerek kodlama adımına geçebilirsiniz.



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

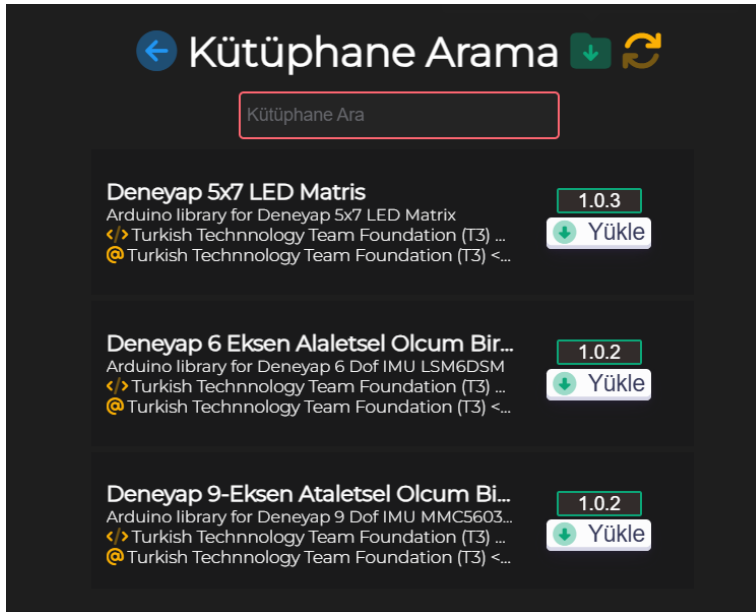
3. İlgili Kütüphaneyi Seçin ve Kurun

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.

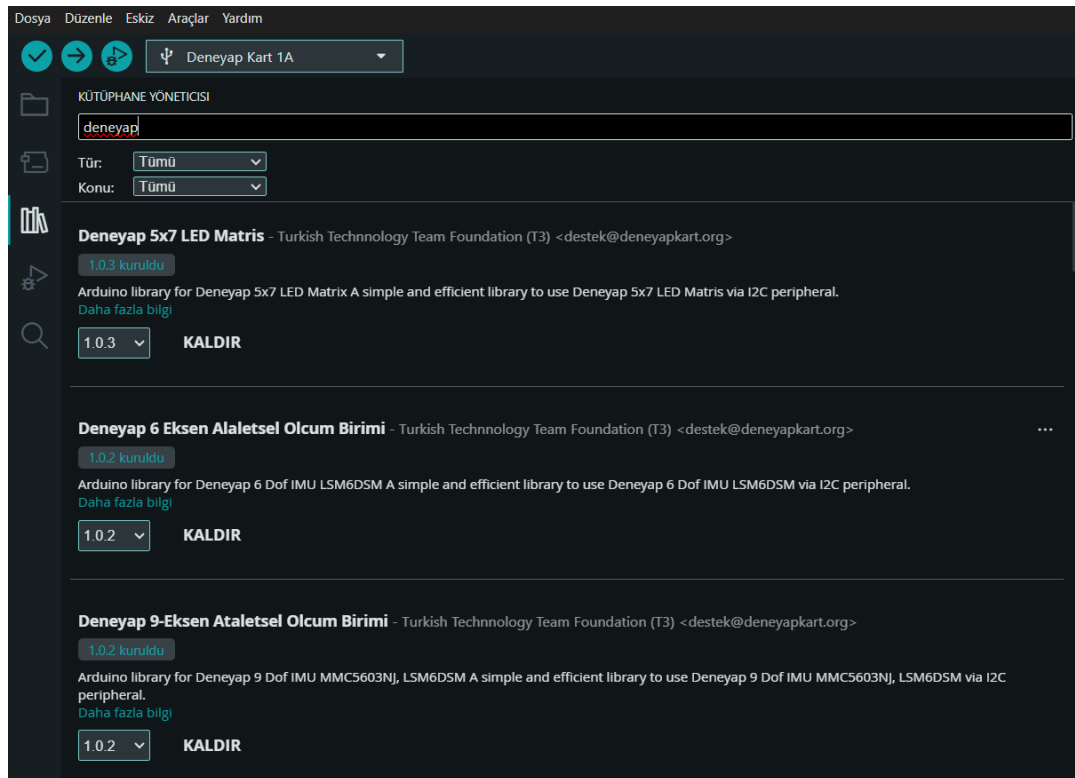
3.Üstteki arama kutusuna deneyap yazın.

- 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, Deneyap Kart 1AV2'yi ve çeşitli sensör modülleri kullanılarak çevredeki hava kalitesini izlemeyi amaçlamaktadır. Proje kapsamında;

- Gaz sensörü ile havadaki kirletici gaz miktarı (ppm cinsinden),
- Sıcaklık ve nem sensörü ile ortamın sıcaklık (°C) ve nem (%RH) değerleri ölçülmektedir.
- Elde edilen veriler OLED ekranda gösterilirken, genel hava durumu LED matris ekranında sembollerle kullanıcıya görsel olarak sunulmaktadır.

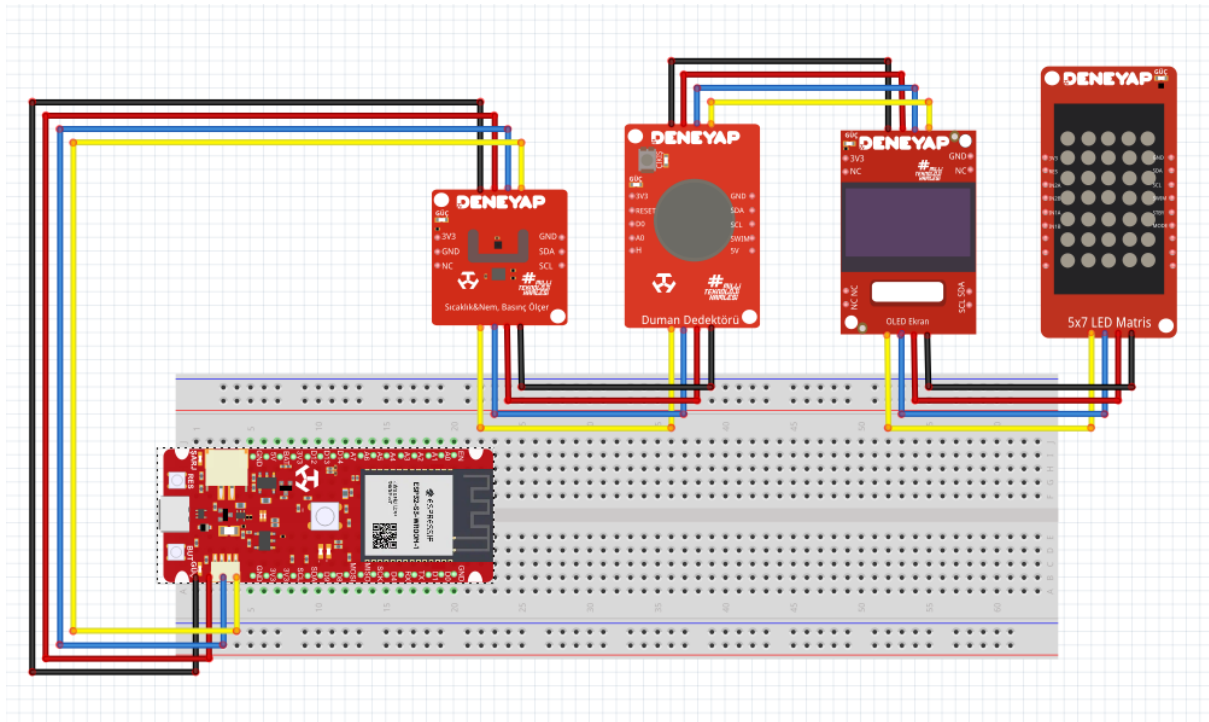
Belirlenen eşik değerlerin aşılması durumunda sistem, hava kalitesinin kötü olduğunu algılar ve LED matris üzerinde bir uyarı işareti gösterir. Aksi takdirde kullanıcıya bir gülümseme yüzü ile iyi hava durumu bilgisi verilir. Bu proje, kullanıcıların ortam koşullarını gerçek zamanlı olarak gözlemlemesini sağlar ve hava kirliliği farkındalığını artırmayı hedefler.

Gerekli Malzemeler:

- Bir adet Deneyap Kart 1A v2
- Deneyap Duman Dedektörü
- Deneyap Sıcaklık, Nem, Basınç Ölçer
- Deneyap OLED Ekran
- Deneyap 5x7 LED Matris
- Dört adet I2C kablosu

Devre Şeması:

Devre şeması aşağıdaki gibi kurulabilir.



Şekil 23



Proje Kodları:

Bu bölümde projede kullanılan Deneyap modüllerine ait kütüphaneler programa `#include` komutu ile dahil edilmiştir:

- **#include <Deneyap_DumanDedektoru.h>**
Deneyap **Duman Dedektörü** modülünü kullanmak için gereklidir.
Ortamdaki **duman yoğunluğunu algılamayı** ve bu veriye göre işlem yapmayı sağlar.
- **#include <Deneyap_SicaklikNemBasincOlcer.h>**
Deneyap **Sıcaklık, Nem ve Basınç Ölçer** modülü için kullanılır.
Ortamin **sıcaklık, nem ve basınç** verilerini ölçmeyi mümkün kılar.
- **#include <Deneyap_OLED.h>**
Deneyap **OLED ekran** modülünü projeye ekler.
Ölçülen verileri **küçük bir ekran üzerinden görsel olarak sunar**.
- **#include <Deneyap_5x7LedEkran.h>**
Deneyap **5x7 LED Matris** modülünü kullanmak için gereklidir.
Basit görseller, semboller veya uyarılar göstermek amacıyla kullanılır.

```
#include <Deneyap_DumanDedektoru.h>
#include <Deneyap_SicaklikNemBasincOlcer.h>
#include <Deneyap_OLED.h>
#include <Deneyap_5x7LedEkran.h>
```

Şekil 24

Nesne Tanımlamaları Açıklamaları

- **SmokeDetector gazSensor;**
 - Gaz/duman dedektörü modülünü temsil eder.
 - Ortamda gaz veya duman algılandığında gerekli işlemlerin yapılmasını sağlar.
- **SHT4x thpSensor;**
 - Sıcaklık ve nem sensörünü temsil eder.
 - Ortam sıcaklığı ve nem değeri gibi çevresel verilerin okunmasını sağlar.
- **OLED oled;**
 - OLED ekran modülünü temsil eder.
 - Ölçülen verilerin kullanıcıya görsel olarak sunulmasını sağlar.
 - Metin
 - Sembol veya uyarı mesajları ekranda gösterilebilir.
- **DotMatrix ledMatrix;**
 - 5x7 LED matris ekranını temsil eder.
 - Görsel bildirimler ve simgeler göstermek için kullanılır.
 - Şekiller
 - Rakamlar
 - Uyarı sembolleri gibi bilgiler gösterilebilir.



```
SmokeDetector gazSensor;  
SHT4x thpSensor;  
OLED oled;  
DotMatrix ledMatrix;
```

Şekil 25

Bu bölümde, hava kalitesini değerlendirmek için kullanılan eşik değerler tanımlanır.

- **const uint16_t gazEsigi = 200;**
 - Gaz sensöründen ölçülen değerin **200 ppm**'i aşması durumunda hava kalitesi **kötü** kabul edilir.
 - Bu eşik, olası tehlikeli gaz sızıntılarını belirlemek için referans alınır.
- **const float sıcaklikEsigi = 30.0;**
 - Ortam sıcaklığı **30.0 °C**'yi geçtiğinde sıcaklık değeri **kritik** olarak değerlendirilir.
 - Bu durum, ortamın çok sıcak olduğunu ve önlem alınması gerektiğini gösterir.
- **const float nemEsigi = 80.0;**
 - Nem oranı **%80**'i aştığında ortamın nem seviyesi **yüksek ve istenmeyen** olarak değerlendirilir.
 - Bu değer, havalandırma ya da nem alma sistemlerinin çalıştırılması gerektiğini gösterebilir.

Bu sabitler ne işe yarar?

- Sensör verilerinin karşılaştırılmasında kullanılırlar.
- Hava durumu değerlendirmesinde **temel referans** değerler olarak görev yaparlar.

```
const uint16_t gazEsigi = 200;  
const float sıcaklikEsigi = 30.0;  
const float nemEsigi = 80.0;
```

Şekil 26



Sensör Türü	Ölçülen Değer	Eşik Değeri(Limit)	Anlamı
Gaz Sensörü	Gaz Yoğunluğu(ppm)	≥ 200 ppm	Kirli hava (Uyarı göster)
Sıcaklık	Sıcaklık (°C)	≥ 30.0 °C	Uygunsuz sıcaklık koşulu
Nem	Nem (%RH)	≥ 80.0 %	Uygunsuz nem seviyesi
Genel Hava Durumu	Tüm ölçümler uygun	—	Temiz hava (Gülen yüz göster)

Bu bölümde, cihaz açıldığında yalnızca bir kez çalışan **setup()** fonksiyonu içinde yapılan işlemler şunlardır:

Seri haberleşme başlatılır:

- **Serial.begin(115200);** komutu ile bilgisayara veri gönderimi sağlanır (hata mesajları veya sensör verileri gibi).

Modüller başlatılır:

- **Gaz sensörü:** Belirtilen I2C adresiyle başlatılır.
- **Sıcaklık ve nem sensörü:** SHT4X sensörü devreye alınır.
- **OLED ekran:** Görsel bilgilendirme için başlatılır.
- **LED matris ekran:** Semboller ve uyarılar için kullanıma hazır hale getirilir.

Bağlantı hatası kontrolü yapılır:

- Her modül için bağlantı sağlanamazsa:
 - OLED ekrana ilgili hata mesajı yazdırılır.
 - Program, **while(1);** komutu ile durdurulur.

Başlangıç mesajı gösterilir:

- OLED ekran temizlenir.
- “Hava Kalitesi” ve “Analiz ediliyor...” mesajları ekrana yazdırılır.
- Bu mesajlar 2 saniye boyunca görüntülenir.

Cihaz ölçüme hazır hale gelir.



```

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

    if (!gazSensor.begin(0x20)) {
        Serial.println("Gaz sensörü I2C bağlantısı başarısız");
        while (1);
    }

    if (!thpSensor.begin(0x44)) {
        Serial.println("Sıcaklık/Nem sensörü I2C bağlantısı başarısız");
        while (1);
    }

    if (!oled.begin(0x7A)) {
        Serial.println("OLED ekran I2C bağlantısı başarısız");
    }
    oled.init();

    if (!ledMatrix.begin(0x0A)) {
        Serial.println("LED matris I2C bağlantısı başarısız");
        while (1);
    }

    oled.clearDisplay();
    oled.setTextXY(0, 0);
    oled.putString("Hava Kalitesi");
    oled.setTextXY(1, 0);
    oled.putString("Analiz ediliyor...");
    delay(2000);
}

```

Şekil 27

- Bu kod parçasında, **loop()** fonksiyonu içinde çeşitli sensör okuma ve görüntüleme işlemleri gerçekleştirilir.
- Öncelikle **thpSensor.measure()** fonksiyonu çağrılarak sıcaklık ve nem sensöründen ölçüm yapılır; bu fonksiyon başarılı sonuç döndürmezse (**SHT4X_STATUS_OK** ile karşılaştırılır) **Serial.println()** ile “Sıcaklık/Nem ölçüm hatası” mesajı seri porta yazdırılır ve **return;** komutuyla fonksiyon sonlandırılır.
- Ölçüm başarılı ise, **thpSensor.TtoDegC()** fonksiyonu ile sıcaklık değeri °C cinsinden, **thpSensor.RHtoPercent()** fonksiyonu ile nem yüzdesi okunur.



- Gaz sensöründen ise **gazSensor.ReadSmokeAnalog()** fonksiyonu ile analog gaz değeri elde edilir.
- Bu değerler **Serial.print()** ve **Serial.println()** fonksiyonları ile sırasıyla seri monitöre yazdırılır.
- OLED ekran önce **oled.clearDisplay()** ile temizlenir, ardından **oled.setTextXY()** fonksiyonu ile metin yazılacak satır ve sütun konumu ayarlanır.
- **oled.putString()** fonksiyonu ile metin, **oled.putFloat()** ile ondalıklı sayılar, **oled.putInt()** ile tam sayılar ekrana yazdırılır.
- Daha sonra, ölçülen gaz, sıcaklık ve nem değerleri önceden tanımlı eşik değerlerle karşılaştırılır ve **bool havaKotu** değişkenine atanır.
- **ledMatrix.resetDotRows()** fonksiyonu ile LED matris ekran temizlenir.
- Eğer **havaKotu** doğru ise, 5x7 LED matris üzerinde tanımlı **uyari** dizisi **ledMatrix.drawLedMatrix()** fonksiyonu ile 500 ms boyunca gösterilir ve OLED ekranda **oled.putString()** ile "HAVA KALİTESİ KOTU!" mesajı görüntülenir.
- Eğer hava iyiye, benzer şekilde **gulumseme** dizisi LED matriste gösterilir ve OLED'de "Hava Durumu: İyi" mesajı yazılır.
- Son olarak **delay(3000);** fonksiyonu ile döngü 3 saniye duraklatılır ve ölçümler yeniden yapılmak üzere döngü başa döner.

```
void loop() {

    if (thpSensor.measure() != SHT4X_STATUS_OK) {

        Serial.println("Sicaklik/Nem ölçüm hatası");

        return;

    }

    float sicaklik = thpSensor.TtoDegC();

    float nem = thpSensor.RHtoPercent();

    uint16_t gazDegeri = gazSensor.ReadSmokeAnalog();

    Serial.print("Sicaklik: "); Serial.print(sicaklik); Serial.print("
°C, ");

    Serial.print("Nem: "); Serial.print(nem); Serial.print(" %, ");

    Serial.print("Gaz: "); Serial.println(gazDegeri);
```



```

oled.clearDisplay();

oled.setTextXY(0, 0);

oled.putString("Sicaklik: ");

oled.putFloat(sicaklik, 1);

oled.putString(" C");


oled.setTextXY(1, 0);

oled.putString("Nem: ");

oled.putFloat(nem, 1);

oled.putString(" %");


oled.setTextXY(2, 0);

oled.putString("Gaz: ");

oled.putInt(gazDegeri);

oled.putString(" ppm");


bool havaKotu = (gazDegeri > gazEsigi || sicaklik > sicaklikEsigi ||
nem > nemEsigi);

ledMatrix.resetDotRows();


if (havaKotu) {

    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
};

ledMatrix.drawLedMatrix(uyari, 500);

oled.setTextXY(4, 0);
oled.putString("HAVA KALITESI KOTU!");
} else {

const byte gulumseme[35] = {

    0,0,0,0,0,
    0,0,0,0,0,
    0,1,0,1,0,
    0,0,0,0,0,
    1,0,0,0,1,
    0,1,1,1,0,
    0,0,0,0,0

};

ledMatrix.drawLedMatrix(gulumseme, 500);

oled.setTextXY(4, 0);

```



```
oled.putString("Hava Durumu: İyi");  
  
}  
  
delay(3000);  
}
```

Şekil 28

