

DENEYAP KART 1A v2 İLE

SESLE YANAN LAMBA

A) Deneyap Kart IDE Kurulumu:

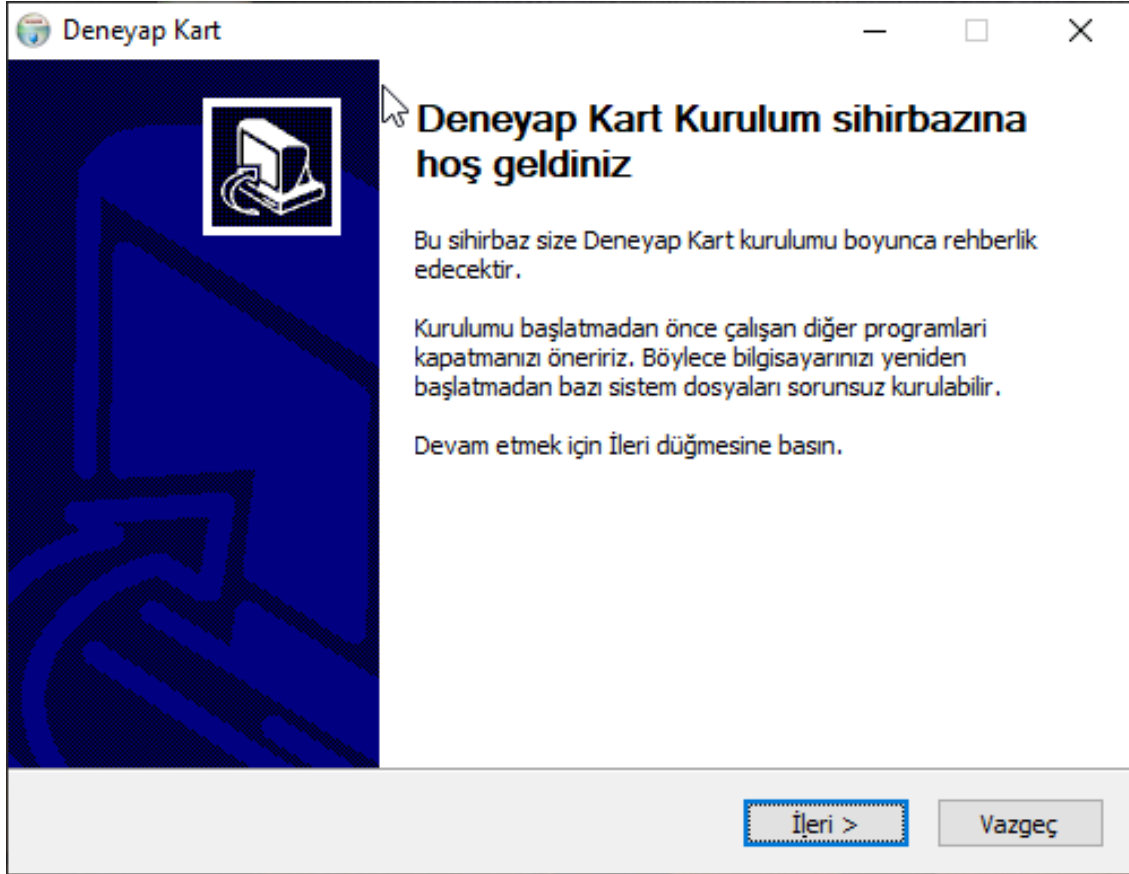
Deneyap Kart IDE, metin tabanlı programlama yapmanız için geliştirilmiş bir programlama ortamıdır. Eğer metinsel kodlama yapmak istiyorsanız, Deneyap Kart IDE'yi kullanmanız gerekir.

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.

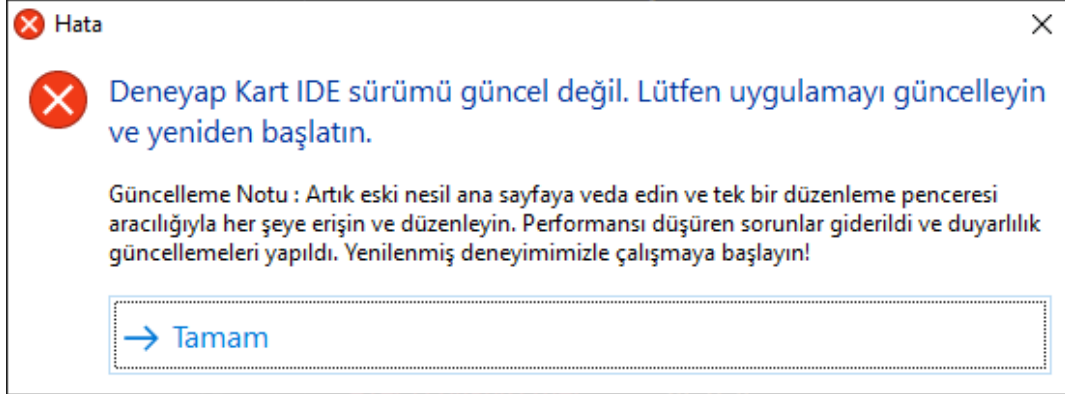


Şekil 1



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.

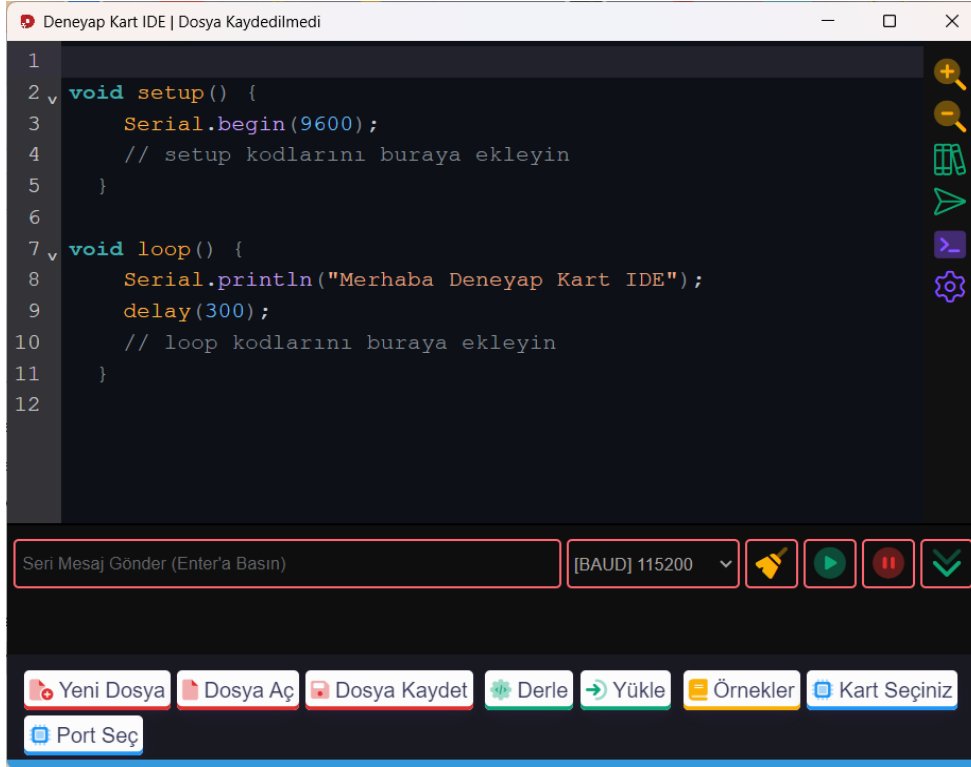


Şekil 3



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.

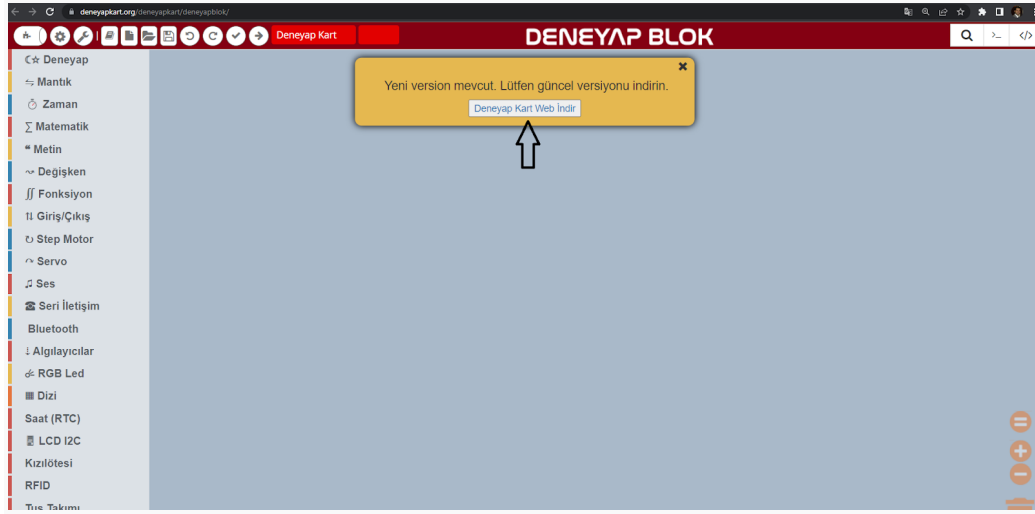
B) DENEYAP Kart Blok Web Kurulumu:

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.

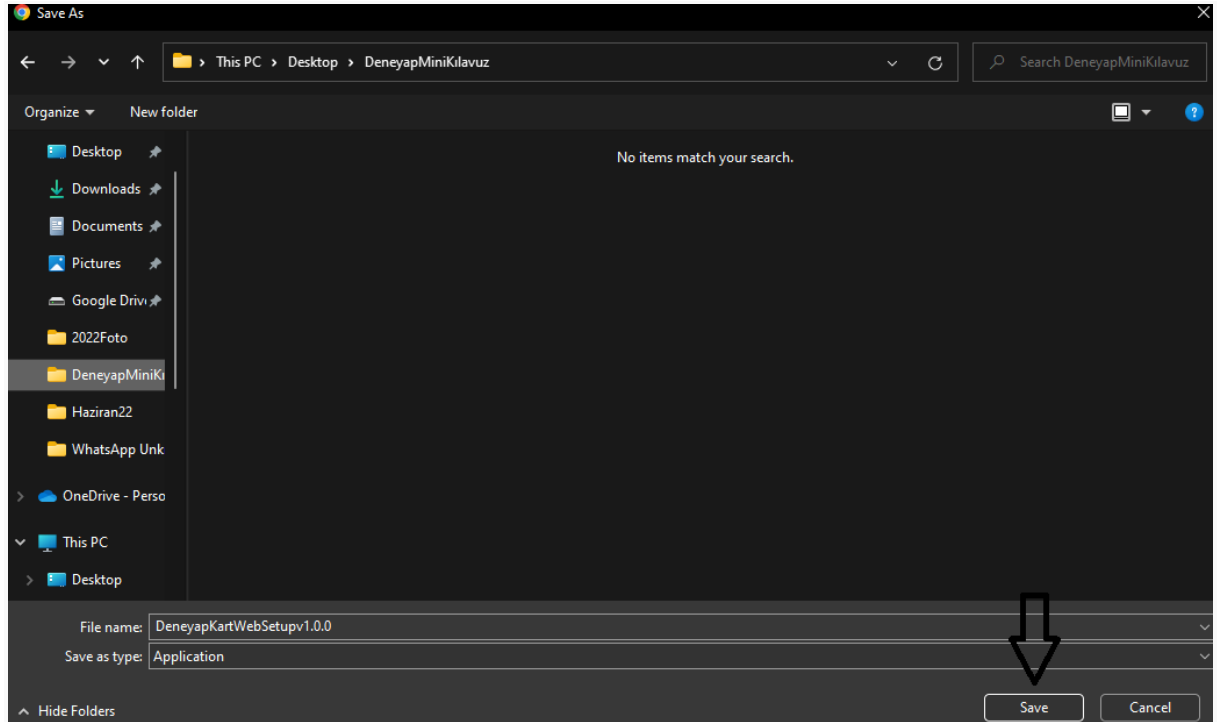


1- Deneyap Kart Blok sayfasına geiş yapıldığında “Deneyap Kart Web İndir” butonuna tıklanılır.



Şekil 5

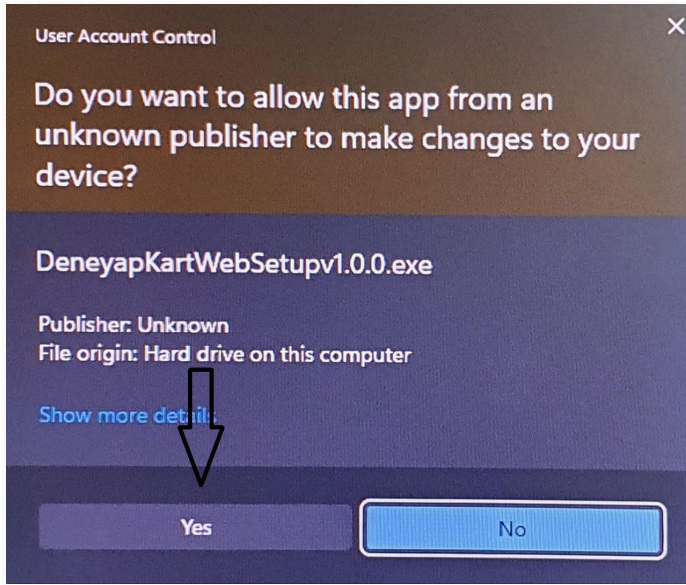
2- Açılan pencerede “Save” butonuna tıklayıp, programı bilgisayarınıza indiriniz.



Şekil 6

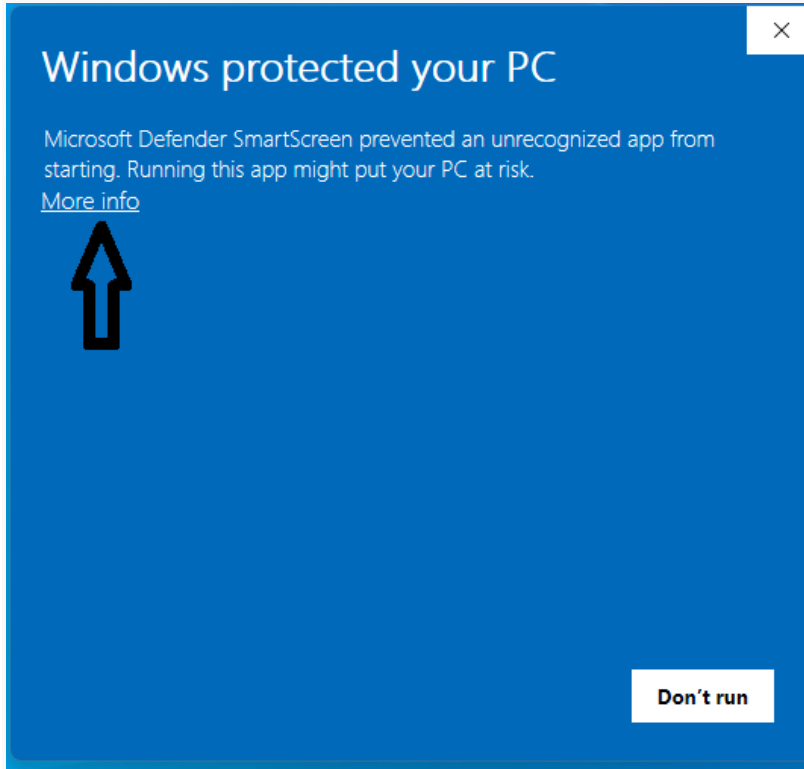


3- İndirme işlemi tamamlandıktan sonra kurulum işlemini başlatıyoruz.



Şekil 7

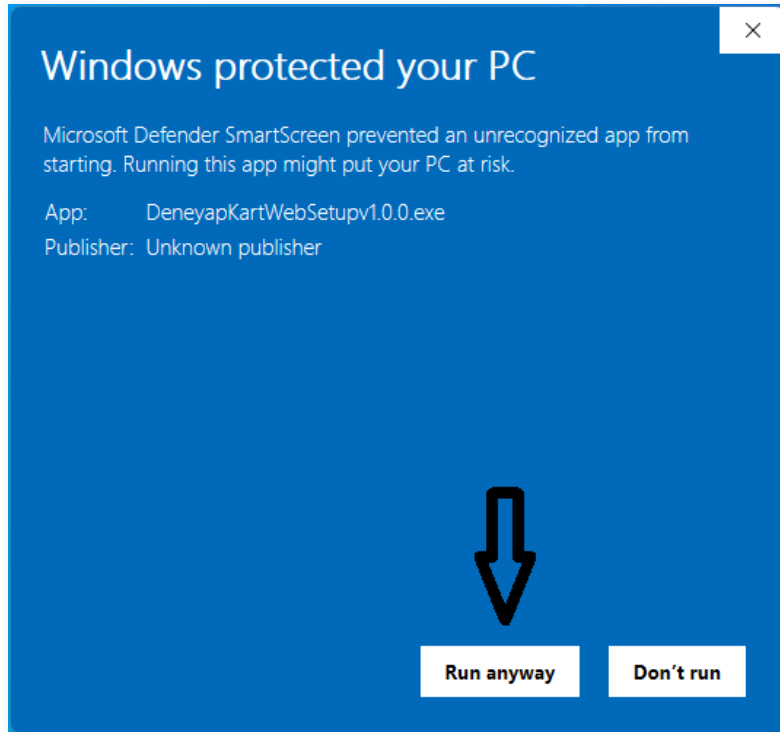
4- Kurulum başlatma sırasında Windows Defender vb. programlar aşağıdaki şekilde uyarı verebilir. Bu şekilde bir ekran ile karşılaşırsanız "More info" yazısına tıklayınız.



Şekil 8

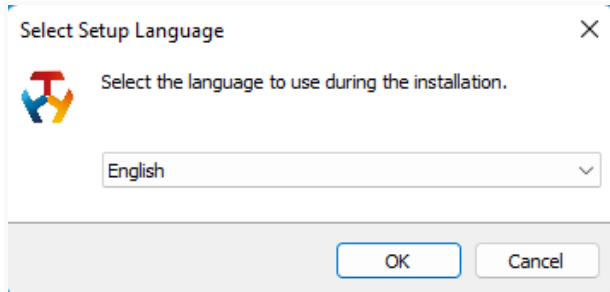


5- Yükleme adımına geçmek için “Run anyway” butonuna tıklayınız.



Şekil 9

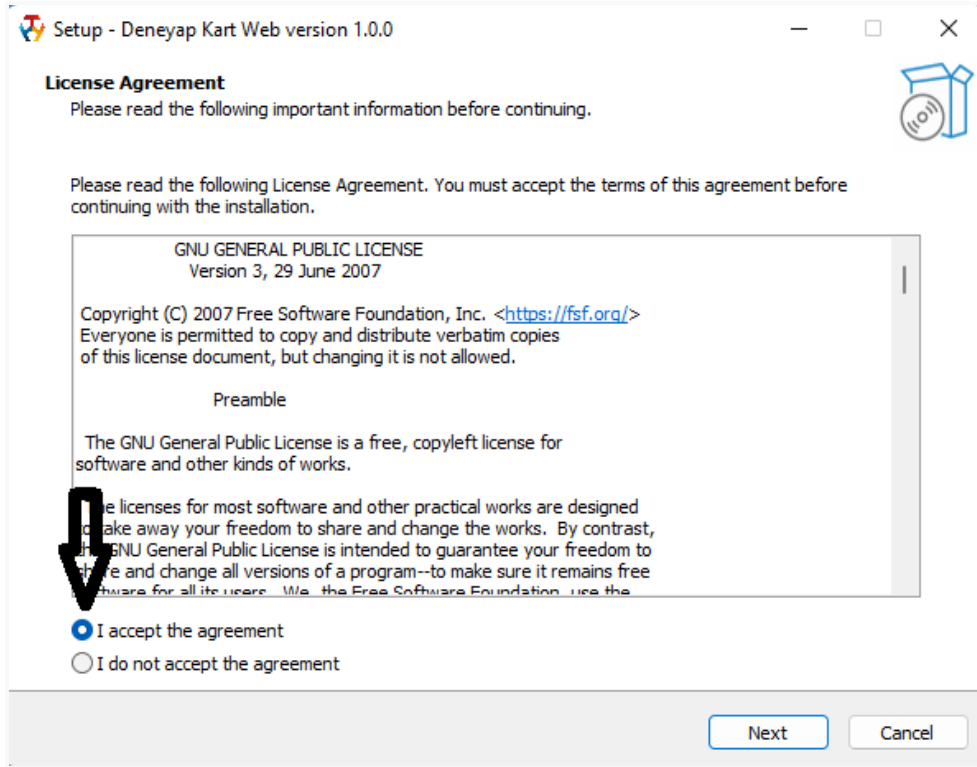
6- Yüklemenin ilk adımında dil seçimi yapmanız istenir. Dil seçimi yapıp “OK” butonuna tıklayınız.



Şekil 10

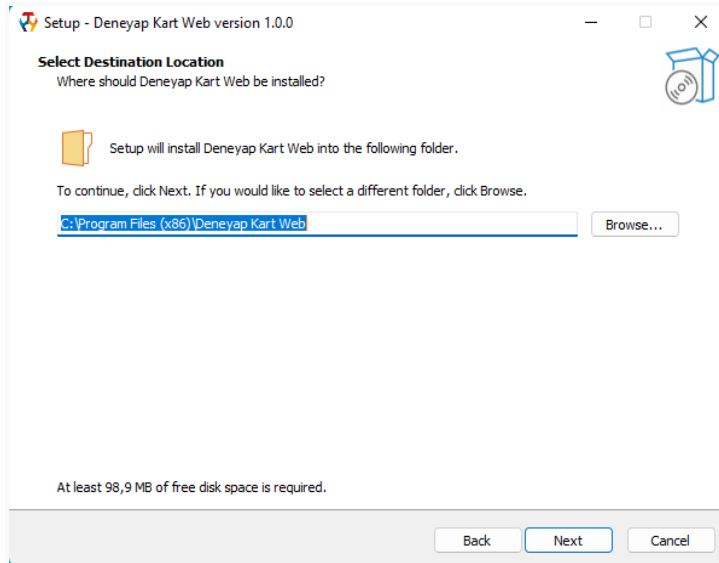


7- Ardından gelen ekranda lisans anlaşmasını onayladıktan sonra “Next” butonuna tıklayınız.



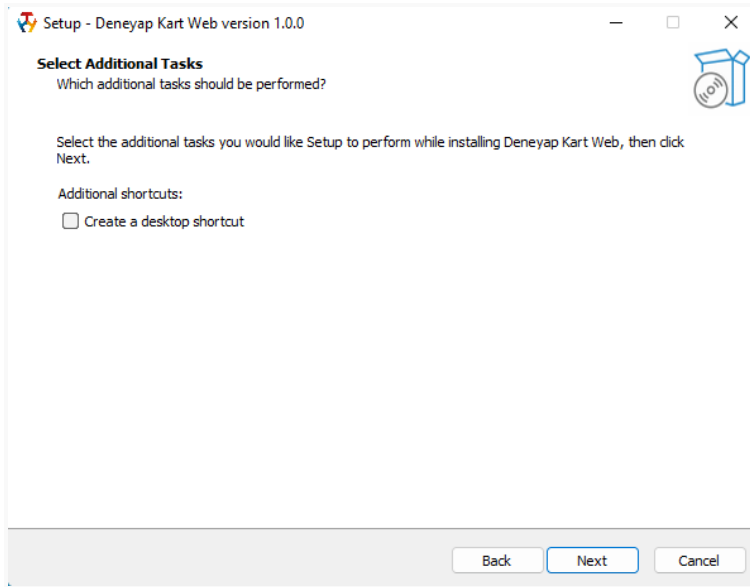
Şekil 11

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



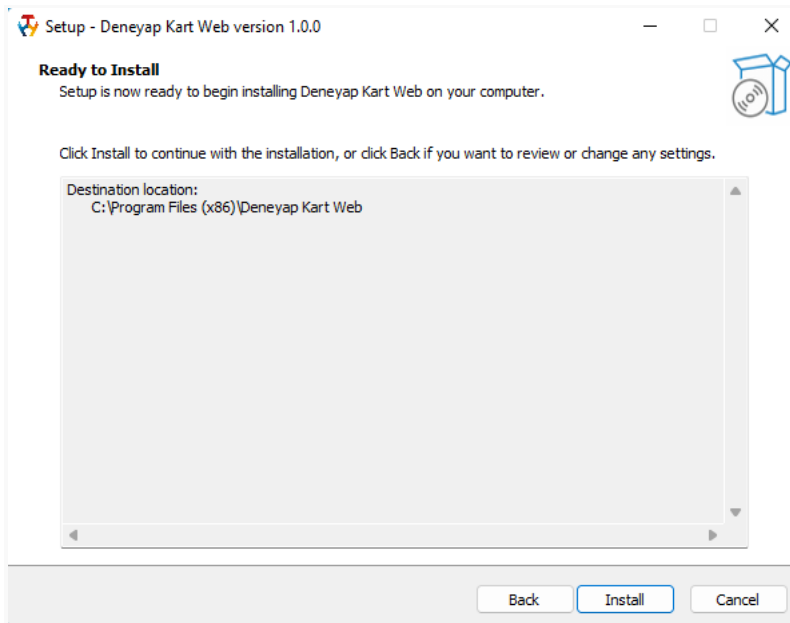
Şekil 12

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



Şekil 13

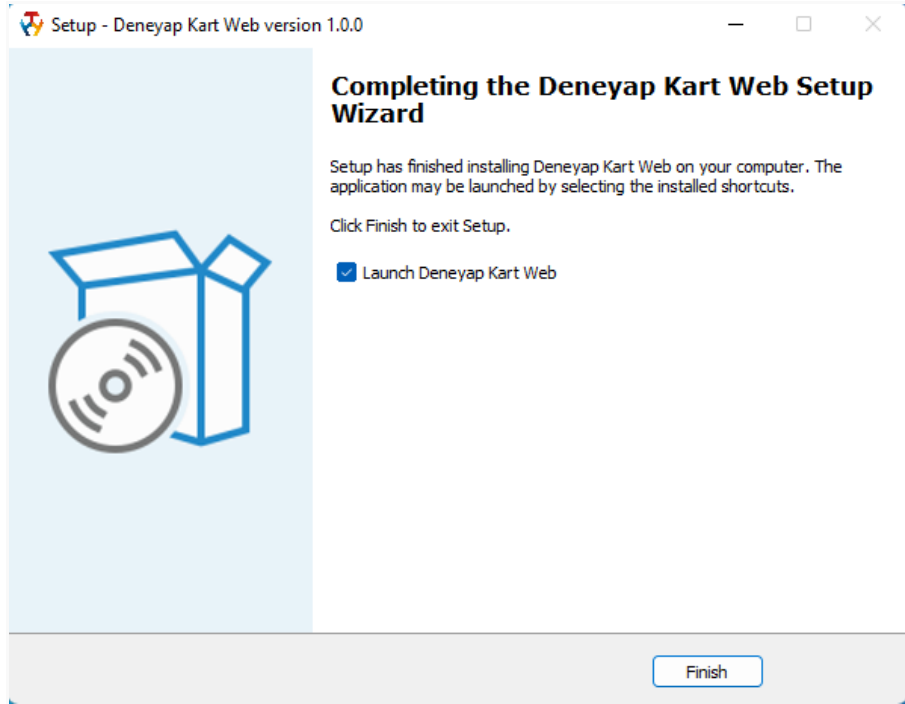
10- Gelen ekranda “Install” butonuna tıklayıp yüklemeyi başlatabilirsiniz.



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.



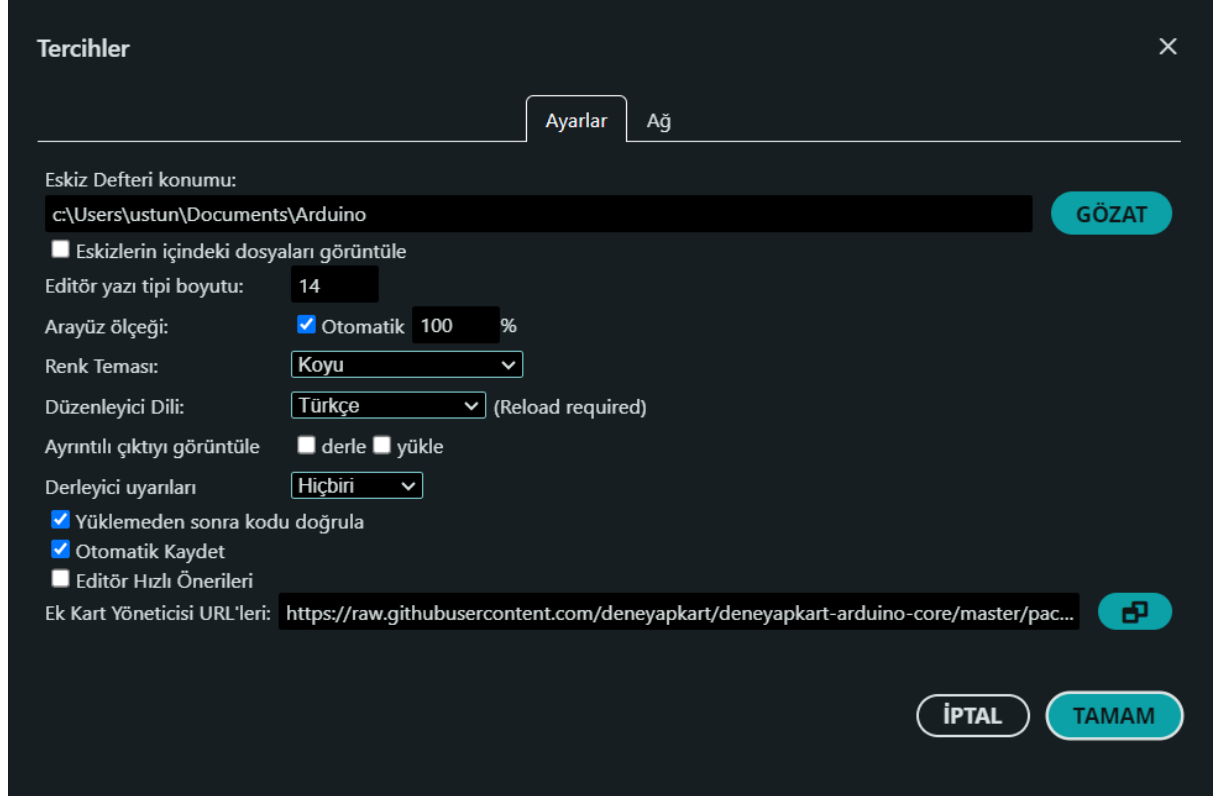
Şekil 16



C) Arduino IDE Kullanacaksanız: Deneyap Kart Nasıl 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

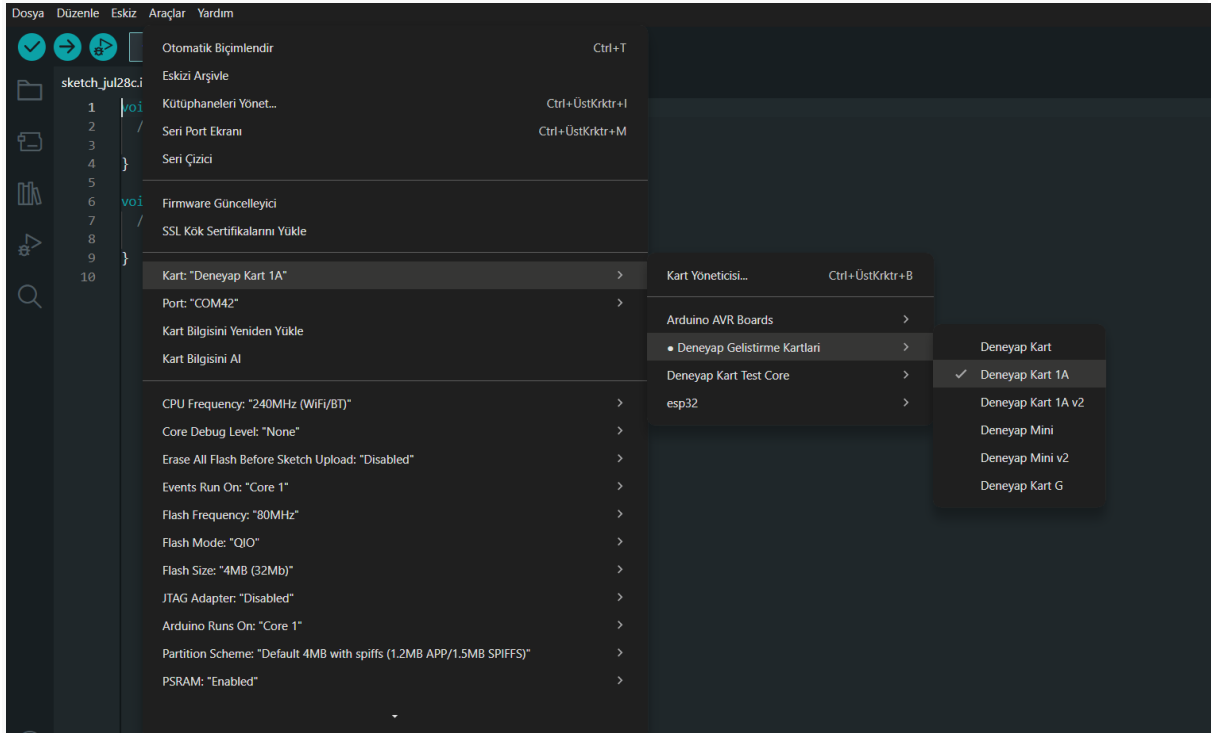


2- Ardından, Araçlar (Tools) → Kart (Board) → Kart Yöneticisi (Boards Manager) adımını takip ederek gelen ekranda arama satırına “Deneyap Geliştirme Kartları” yazınız. En son sürüm varsayılan olarak seçilmiş halde gelir ve Kur (Install) butonuna tıklayarak yüklemeyi gerçekleştirebilirsiniz.



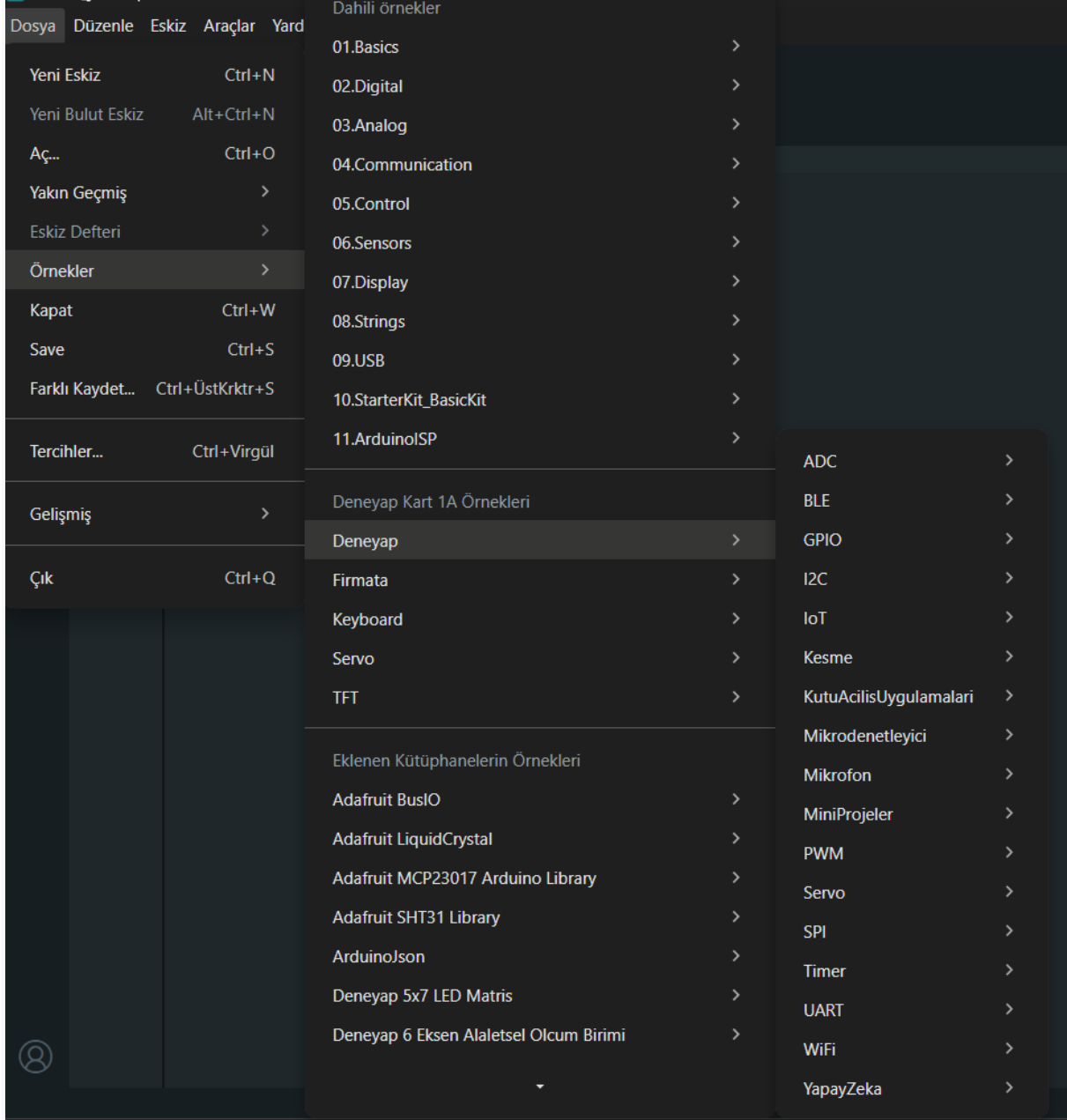
Şekil 18

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

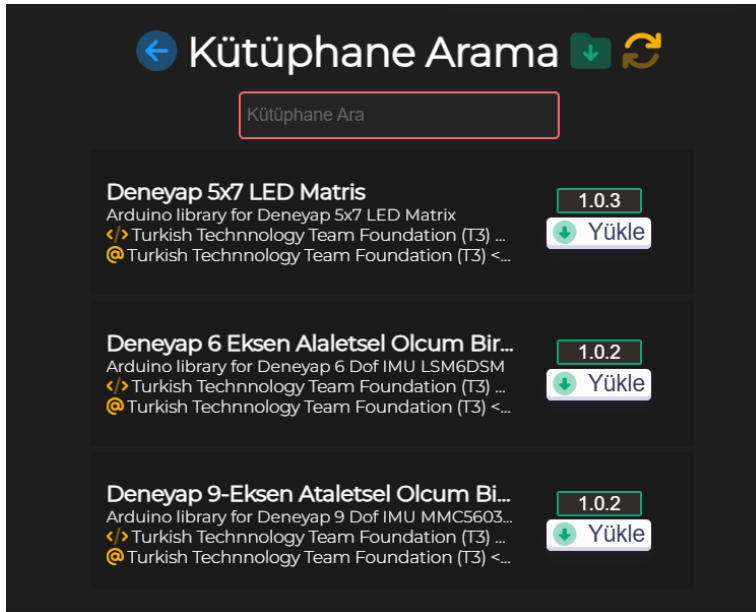
3. İlgili Kütüphaneyi Seçin ve Kurun

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.

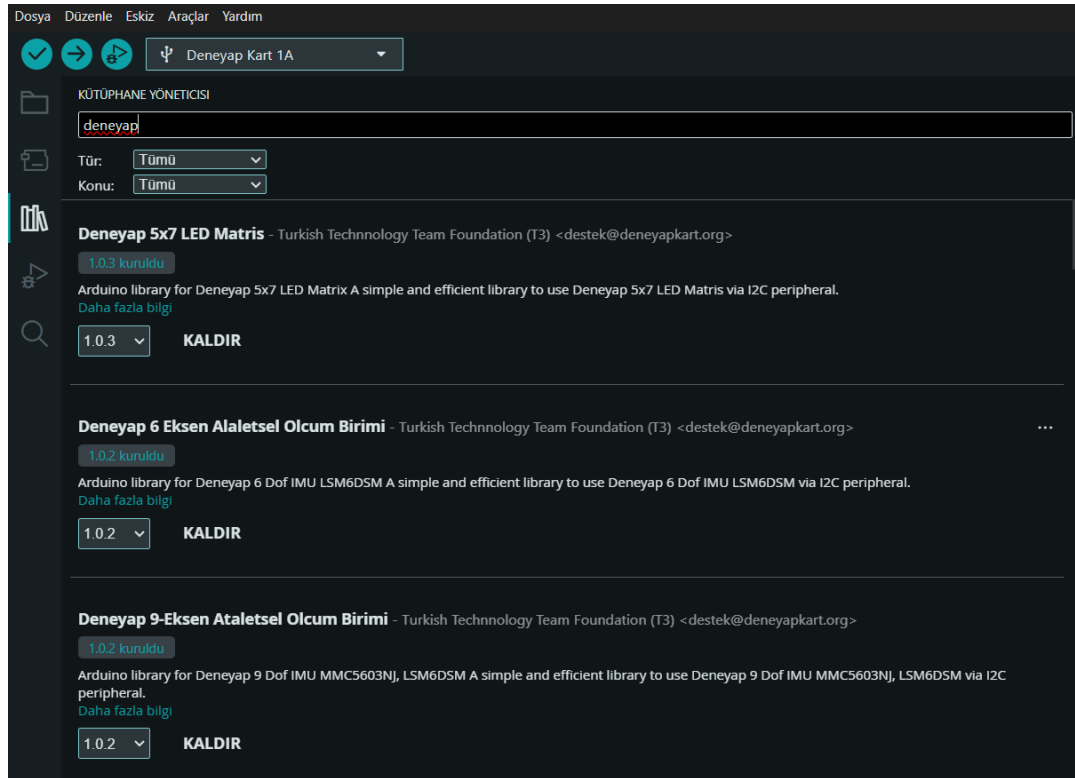
3.Üstteki arama kutusuna deneyap yazın.

- 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



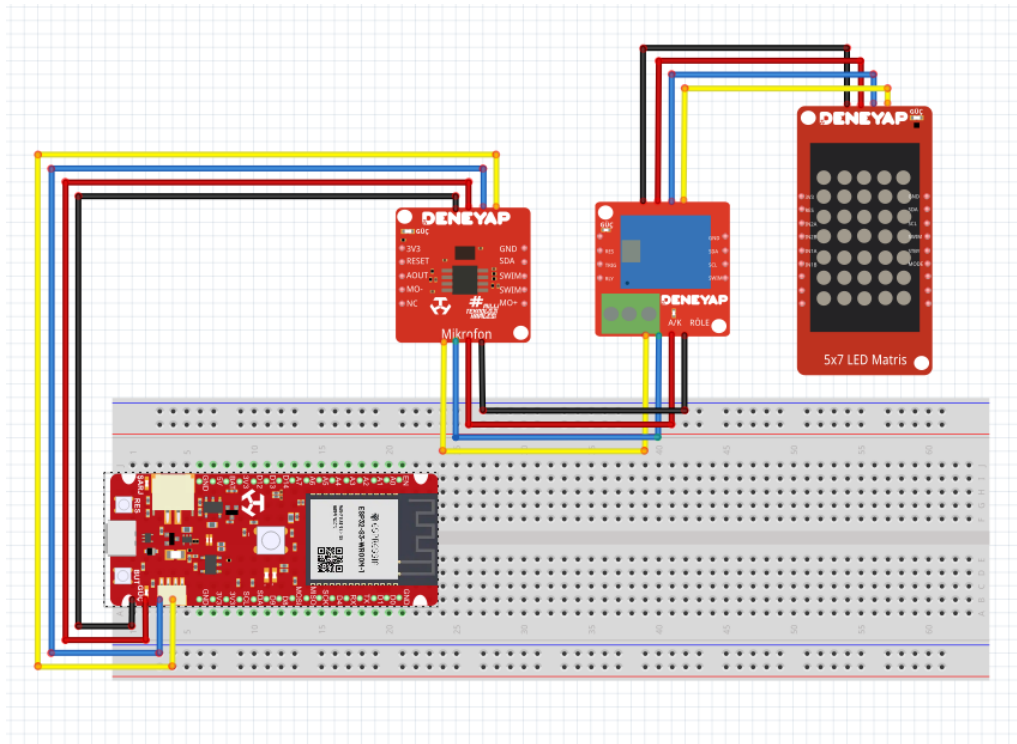
Proje Özeti: Bu proje, sesle çalışan bir lamba kontrol sistemidir. Mikrofon yardımıyla ortam sesi sürekli olarak algılanır. Eğer ortamda ani ve yüksek bir ses oluşursa (örneğin el çırpma, bağırma gibi), sistem bu sesi algılar. Sesin şiddeti belirli bir eşik değerin (örneğin 400) üzerine çıktığında, röle tetiklenerek lamba açılır veya kapanır. Sistem ilk çalıştığında lamba kapalıdır. LED matris üzerinde sistemin hazır olduğunu belirten bir sembol gösterilir. Kullanıcı tarafından çıkarılan yüksek bir ses algılandığında, rölenin durumu değiştirilir. Yani lamba açıksa kapanır, kapalıysa açılır. Her sesli komut sonrası sistem bir süre bekleyerek (örneğin 800 milisaniye) yanlışlıkla tekrar tetiklenmeyi önler. Röle durumu değiştiğinde LED matris ekranında da bu durum görsel olarak gösterilir ve seri monitörde bilgi mesajı yazdırılır. Ayrıca röle çıkışına sadece lamba değil, **motor gibi farklı elektrikli cihazlar da bağlanabilir**. Örneğin röleye bağlı bir motor yerleştirilirse, sistem yüksek ses algılandığında bu motor da çalıştırılabilir. Böylece bu sistem yalnızca aydınlatma değil, sesle çalışan **otomatik perde açma, havalandırma başlatma, kapı motoru tetikleme** gibi çok amaçlı uygulamalar için de kullanılabilir.

Gerekli Malzemeler:

- Bir adet Deneyap Kart 1A v2
- Bir adet Deneyap Mikrofon
- Bir adet Deneyap Role
- Bir adet Deneyap 5x7 LED Matris
- Üç adet I2C kablosu

Devre Şeması:

Devre şeması aşağıdaki gibi kurulabilir.



Şekil 23

Proje Kodları:

Aşağıda verilen üç satır, Deneyap Kart ile kullanılan modülleri programa dahil eder.

- **#include <Deneyap_Mikrofon.h>** satırı, mikrofon modülünü kullanmak için gerekli kütüphaneyi ekler.
- Bu sayede ses seviyesini ölçebiliriz.
- **#include <Deneyap_5x7LedEkran.h>** satırı, 5x7 LED matris ekran modülünü kullanmak için gereklidir.
- Bu ekran üzerinde bilgi gösterebiliriz.
- **#include <Deneyap_Role.h>** satırı, röle modülünü kontrol etmek için gerekli olan kütüphaneyi ekler.
- Röle sayesinde ses algılandığında bağlı bir cihaz (örneğin lamba) açılıp kapatılabilir.

```
#include <Deneyap_Mikrofon.h>

#include <Deneyap_5x7LedEkran.h>

#include <Deneyap_Role.h>
```

Şekil 24

- Kodun bu kısmında mikrofon, LED matris ve röle modüllerini kullanmak için sırasıyla **mic**, **ledMatrix** ve **role** adında nesneler tanımlanmıştır.
- **mic** nesnesi, mikrofon üzerinden ses düzeylerini okumak için kullanılır.
- **ledMatrix** nesnesi, 5x7 LED matris ekranına veri göndermek için kullanılır.
- **role** nesnesi ise bağlı olan röleyi kontrol etmek için kullanılır.
- Bu tanımlar, modüllerin kod içinde kullanılabilmesini sağlar.

```
Microphone mic;
DotMatrix ledMatrix;
Relay role;
```

Şekil 25

- Bu kodda ses algılama ve röle kontrolü için bazı sabitler ve durum değişkenleri tanımlanmıştır.
- **SES_ESIGI**, mikrofon üzerinden algılanan sesin röleyi tetiklemesi için gereken minimum ses seviyesidir; burada **400** olarak ayarlanmıştır.
- **DEBOUNCE_SURESI**, istenmeyen çift algılamaları önlemek için ses algılamaları arasına konulan zaman aralığıdır ve milisaniye cinsinden **800** olarak belirlenmiştir.
- **lambaDurumu**, lambanın açık mı kapalı mı olduğunu izlemek için kullanılan bir durum değişkenidir.



- **sonTetiklemeZamani** ise en son sesle tetikleme yapılan zamanı saklayarak debounce süresinin(bir butona basıldığında oluşan **titreşimli sinyallerin yok sayılması için beklenen kısa zaman aralığı**) takip edilmesini sağlar.
- Bu sayede sistem, belirli bir zaman aralığı geçmeden tekrar tetiklenmez.

```
const int SES_ESIGI = 400;
const unsigned long DEBOUNCE_SURESI = 800;

bool lambaDurumu = false;
unsigned long sonTetiklemeZamani = 0;
```

Şekil 26

- Bu kod parçasında, 5x7 LED matris ekranında görüntülenmek üzere üç farklı piksel düzeni tanımlanmıştır.
- Her dizi, LED matris üzerindeki piksellerin açık ya da kapalı olma durumlarını gösterir.
- **HAZIR_BMP** dizisi, cihazın hazır olduğunu göstermek için kullanılan bir görseli temsil eder.
- **ACILDI_BMP** dizisi, lambanın açıldığını belirtmek için tasarlanmıştır.
- **KAPANDI_BMP** dizisi ise lambanın kapandığını göstermek amacıyla kullanılır.
- Bu bitmapler LED matrise çizilerek kullanıcıya farklı durumlar görsel olarak aktarılır.

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

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

const byte KAPANDI_BMP[35] = {
  1, 0, 1, 0, 1,
```

```

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

```

Şekil 27

- **setup()** fonksiyonu, Arduino programının başlangıcında bir kez çalıştırılır ve cihazın çalışması için gerekli ayarları yapar.
- **Serial.begin(115200);** ile seri haberleşme başlatılır; bu, bilgisayara veri göndermek için kullanılır ve 115200 baud hızında iletişim sağlar.
- **mic.begin(0x35)** ile mikrofon sensörü başlatılır. 0x35 mikrofon sensörünün I2C adresidir.
- Eğer mikrofon başlatılamazsa, "**Mikrofon baslatilamadi!**" mesajı seri monitöre yazdırılır ve **while(1);** ile program burada sonsuza kadar durur.
- **ledMatrix.begin(0x0A)** ile 5x7 LED matris ekranı başlatılır. 0x0A LED matris modülünün I2C adresidir.
- Başlatılamazsa "**LED Matris baslatilamadi!**" mesajı çıkar ve program durur.
- **role.begin(0x0C)** ile röle modülü başlatılır. 0x0C modülün I2C adresidir.
- Başlatılamazsa hata mesajı verilir ve program durur.
- Başarıyla başlatılan röle, **role.RelayDrive(0);** ile kapalı konuma getirilir.
- LED matris ekran temizlenir (**ledMatrix.resetDotRows();**).
- **HAZIR_BMP** adlı önceden tanımlı bitmap desen, ekranda 500 ms süreyle gösterilir.
- Bu genellikle cihazın hazır olduğunu kullanıcıya bildirir.

Özetle, bu fonksiyon mikrofon, LED matris ve röle modüllerini hazırlar, röleyi kapalı konuma alır ve LED ekranda hazır olduğunu gösterir.

```

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

    if (!mic.begin(0x35)) {
        Serial.println("Mikrofon baslatilamadi!");
        while (1);
    }

    if (!ledMatrix.begin(0x0A)) {
        Serial.println("LED Matris baslatilamadi!");
    }
}

```

```

    while (1);
}

if (!role.begin(0x0C)) {
    Serial.println("Role baslatilamadi!");
    while (1);
}

role.RelayDrive(0); // Röleyi kapalı başlat
ledMatrix.resetDotRows();
ledMatrix.drawLedMatrix(HAZIR_BMP, 500);
}

```

Şekil 28

- Buradaki **loop()** fonksiyonu, programın sürekli çalıştığı ana döngüdür.
- Her döngüde **sesAlgila()** fonksiyonu çağrılır.
- **sesAlgila()** fonksiyonu ile ortamdan ses verisi alınır ve işlenir.
- Ardından 100 milisaniye beklenir (**delay(100);**).
- Bu sayede ses algılama işlemi sürekli tekrar eder ve ses seviyesi düzenli olarak kontrol edilir.

```

void loop() {
    sesAlgila();
    delay(100);
}

```

Şekil 29

- **sesAlgila()** fonksiyonu, mikrofon modülünden gelen ses değerlerini okur ve sesi algılar.
- **mic.ReadMicrophoneAnalog()** komutu ile mikrofonun analog ses değeri okunur ve **sesDegeri** değişkenine atanır.
- Bu ses değeri, seri monitöre "**Ses:** " etiketiyle birlikte yazdırılır.
- Eğer okunan ses değeri **SES_ESIGI** (ses eşiği) değerinden büyükse ve önceki tetiklemeden (**sonTetiklemeZamani**) itibaren belirlenen gecikme süresi (**DEBOUNCE_SURESI**) geçmişse:
 - **lambaDegistir()** fonksiyonu çağrılır; bu fonksiyon lambanın durumunu değiştirir (açma/kapama).
 - **sonTetiklemeZamani millis()** fonksiyonu ile güncellenir.
- Bu sayede debounce (gürültü önleme) mekanizması uygulanır ve kısa aralıklarla yanlış tetiklemeler engellenir.

- Özetle, fonksiyon mikrofona ses değerini kontrol eder, belirli bir eşik üzerindeki ses algılandığında lambayı açıp kapatır ve hızlı tepkileri önlemek için tetiklemeler arasında minimum süre bırakır.

```
void sesAlgila() {
    uint16_t sesDegeri = mic.ReadMicrophoneAnalog();
    Serial.print("Ses: ");
    Serial.println(sesDegeri);

    if (sesDegeri > SES_ESIGI && (millis() - sonTetiklemeZamani) >
    DEBOUNCE_SURESI) {
        lambaDegistir();
        sonTetiklemeZamani = millis();
    }
}
```

Şekil 30

- Bu fonksiyon, lambanın açık veya kapalı durumunu değiştirir.
- **lambaDurumu** değişkeni tersine çevrilir (açık ise kapatılır, kapalı ise açılır).
- Yeni durum, röle modülüne iletilir ve lambanın durumu değiştirilir.
- LED matrisi temizlenir.
- Yeni duruma göre ekranda "**Lamba AÇILDI**" veya "**Lamba KAPANDI**" mesajı görsel olarak gösterilir.
- Durum aynı zamanda seri monitöre yazdırılır.
- Böylece lambanın durumu hem donanım hem de görsel olarak güncellenmiş olur.

```
void lambaDegistir() {
    lambaDurumu = !lambaDurumu;
    role.RelayDrive(lambaDurumu);
    ledMatrix.resetDotRows();

    if (lambaDurumu) {
        Serial.println("Lamba AÇILDI");
        ledMatrix.drawLedMatrix(ACILDI_BMP, 500);
    } else {
        Serial.println("Lamba KAPANDI");
        ledMatrix.drawLedMatrix(KAPANDI_BMP, 500);
    }
}
```



Şekil 31

