



Co-funded by the
Erasmus+ Programme
of the European Union



WirelessUP!

Avrupa elektroteknik sektöründe yenilik için MEÖ becerilerini geliştirmek

Proje numarası: 2017-1-HR01-KA202-035434

WirelessUP! Toolkit

Modül 3.1: Otomasyon Sistemlerinde Kablosuz Teknolojilerin Uygulanması

Arduino MKR 1000 projesi

Fikri Çıktı 3

Şubat 2019

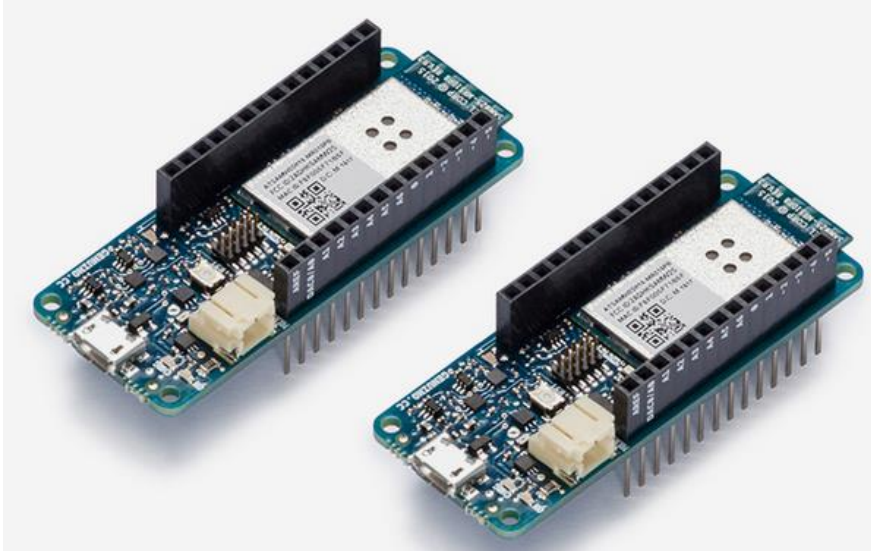
Bu yayın sadece yazarın görüşlerini yansıtır ve komisyon, burada yer alan bilgilerin herhangi bir şekilde kullanılmasından sorumlu tutulamaz.

İçerik

1. Arduino MKR 1000'e Giriş	3
2. IOT hava istasyonu - birinci bölüm	5
2.1. Meteoroloji istasyonunun montajı - sıcaklık, nem, basınç	5
2.1.1. Bağlantı adımı 1 (Şekil 4):	5
2.1.2. Bağlantı adımı 2 (Şekil 5):	6
2.1.3. Bağlantı adımı 3 (Şekil 6):	7
2.1.4. Bağlantı adımı 4 (Şekil 7):	7
3. IOT meteoroloji istasyonu - ikinci bölüm.....	8
3.1. Meteoroloji istasyonunun montajı - havadaki mikropartiküller	8
3.1.1. Bağlantı adımı 1 (Şekil 9):	8
3.1.2. Bağlantı adımı 2 (Şekil 10):	9
4. Blynk platformu	10
4.1. Uygulamayı yükleme	11
4.2. Projeyi oluşturma	11
4.3. Blink widget kutusu	14
4.3.1. Proje 1 - IOT meteoroloji istasyonu (ilk bölüm).....	14
4.3.2. Proje 2 - IOT meteoroloji istasyonu (ikinci kısım).....	15
5. Arduino IDE programı.....	15
5.1. Arduino IDE yazılımının indirilmesi.....	16
5.2. Arduino IDE yazılımının açıklaması.....	17
5.3. Arduino programının ölçüm cihazlarına yüklenmesi ve test edilmesi	18
5.3.1. Projelerimize program ekleme Proje 1 - TTVSU ölçümü ... Hata! Yer işareti tanımlanmamış.	
5.3.2. Proje 2'ye program yükleme – PMS ölçümü.....	22
5.3.3. Program kodlarının birleştirilmesi	23

1. Arduino MKR 1000'e Giriş

MKR 1000, Arduino ürün ailesinin birçok geliştirme aracından biridir. Kart, bir mikroişlemci, bir WiFi bağlantı modülü, LiPo pil güç kaynağı, USB kablosu üzerinden USB şarjı ve mikroişlemcinin faaliyetleri için gereken diğer donanımları içerir. MKR1000 kartı Şekil 1'de gösterilmektedir.



Şekil 1. MKR 100 kartı

Bir mikrodenetleyiciyi küçük bir bilgisayar olarak düşünebiliriz. Ancak, tipik bir masaüstü veya dizüstü bilgisayardan farklı bir amacı vardır. Farklı sensörlerin durumunu okuyabilir ve Arduino IDE geliştirme arayüzünde yazılan programlarla çeşitli cihazları yönetebiliriz.

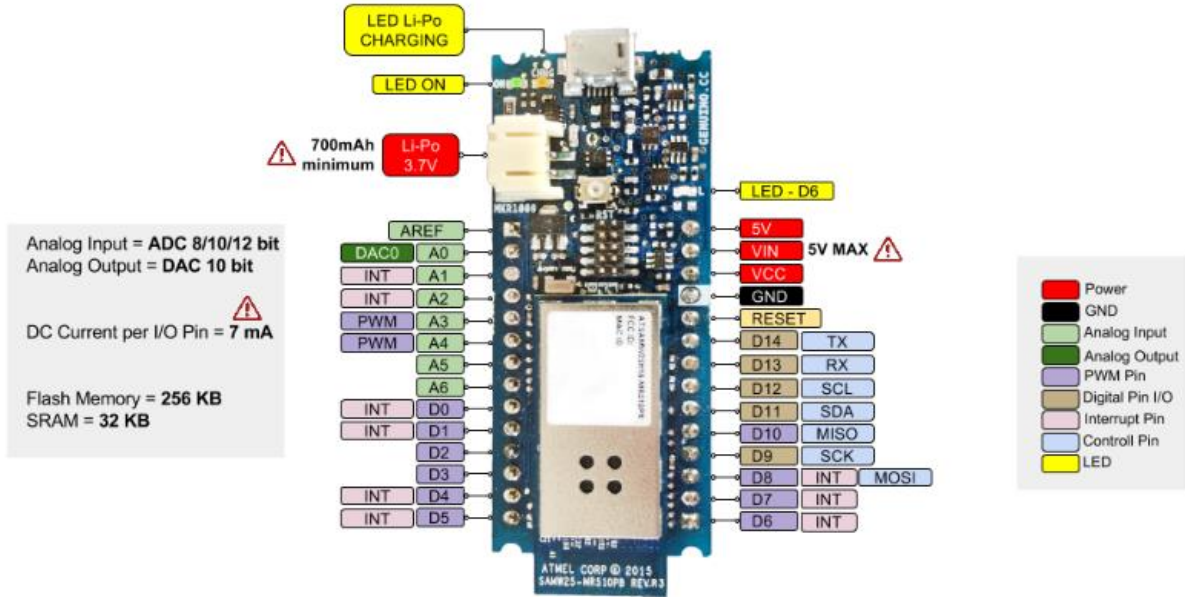
Mikrodenetleyici fonksiyonlar bazı elektronik elemanları kendisine bağlamak ve Arduino IDE programı ile kontrol etmek için kullanılır.

Mikrodenetleyicinin ne yapacağını ve ona bağladığımız şeyi nasıl yönetmesi gerektiğini bilmesi için önce programı yazmalı ve sonra onu mikrodenetleyiciye yüklemeliyiz.

Arduino döşemelerine ek elemanlar takmak için, kenar konnektörleri yardımıyla erişilebilen mikrodenetleyicileri çalıştırıyoruz (Arduino plakasının kenarındaki siyah konnektörler).

Tüm mikrodenetleyici çıkışlarına genel bakış Şekil 2'de mevcuttur.

MKR1000 PINOUT



Şekil 2. MKR 1000 pin yapısı

Görülebileceği üzere, her pinin kendine özgü bir etiketi vardır. Bu etiketler, Arduino pinlerini kontrol etmek için yazdığımız programlarda da kullanılacaktır. Arduino plakası harfler ve sayılarla işaretlenmiştir. Dijital çıkışlar (D0) ve analog çıkışlar (A0) olmak üzere iki tip pin vardır.

Analog çıkışlar A harfi ve A0, A1, A2...A6'ya kadar sayılarla işaretlenmiştir. Bu çıkışlar 0-3,3 V'luk bir analog voltajı ölçebilir.

Dijital çıkışlar 0, 1, 2....14'e kadar numaralarla veya D harfi ve bir sayıyla ifade edilir. Sadece iki durumda olabilirler- 0 veya 1 (DÜŞÜK veya YÜKSEK), veya AÇIK - KAPALI.

Analog çıkışlar dijital çıkışlar olarak da kullanılabilir, ancak dijital çıkışlar analog olamaz. Karttaki tüm pinler iki modda çalışabilir - giriş ve çıkış modu.

Çıkış modu, bu çıkışa bağlı bazı bileşenleri (LED'ler, motorlar, röleler vb.) yönetmek istediğimiz bir mikrodenetleyici kullanıldığında kullanılır. Giriş modu, tek bir çıkışın durumunu okumak istediğimizde veya düğme anahtarları, termostatlar ve benzeri gibi basit sensörler bağlanırken kullanılır.

Uygulamalı örneklere başlayabiliriz, böylece Arduino'nun çalışma şeklini en iyi şekilde anlayabiliriz.

2. IOT meteoroloji istasyonu - birinci kısım

Bu proje, MKR1000 platformunda havadaki parlaklık, sıcaklık, nem, basınç ve mikropartikülleri (hava kalitesi ölçümü) ölçmeye yarayan sensörlerden oluşur. Ayrıca, sensör değerlerinin görselleştirilmesi için bir cep telefonu uygulaması yapmayı da içerir. Meteoroloji istasyon ünitesini yapmak için birkaç adım vardır.

1. Parlaklık, sıcaklık, basınç ve nem seviyesini ölçmek için bir cihazın yapılması.
2. Havadaki mikropartikülleri ölçmek için bir tertibatın oluşturulması (hava kalitesi).
3. Blynk sunucusuna bağlanmak ve mobil uygulamalar oluşturmak için belirteçleri (tokenleri) almak.
4. Arduino programının ölçüm cihazlarına yüklenmesi ve bağlı cihazların test edilmesi.
5. Belirteçler (tokenler) hakkında geribildirim.

2.1. Hava durumu istasyonunun montajı - sıcaklık, nem, basınç

IOT meteoroloji istasyonunun ilk kısmı, ışıkları (APDS sensörü), nemi (DHT 11 sensörü) ve hava basıncını (BMP 180) ölçmeye yarayan üç sensörden oluşmaktadır. - Şekil 3.

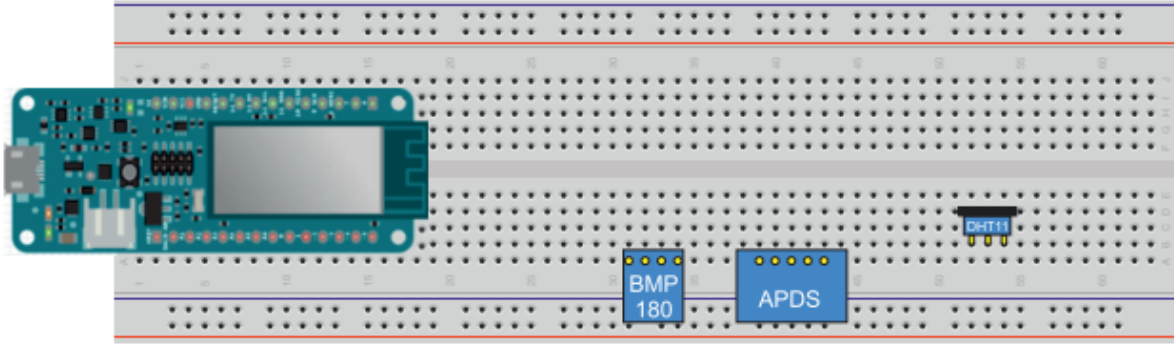
Şekil 4'teki şema, deneysel bir panoya monte edilmiş olan sensörleri göstermektedir. Sarı renkli temas noktaları, Arduino MKR1000 platformuna bağlayacağımız sensörlerin temas noktalarını belirtir.



Şekil 3. Hava istasyonu için sensörler

2.1.1. Bağlantı, 1. adım (Şekil 4):

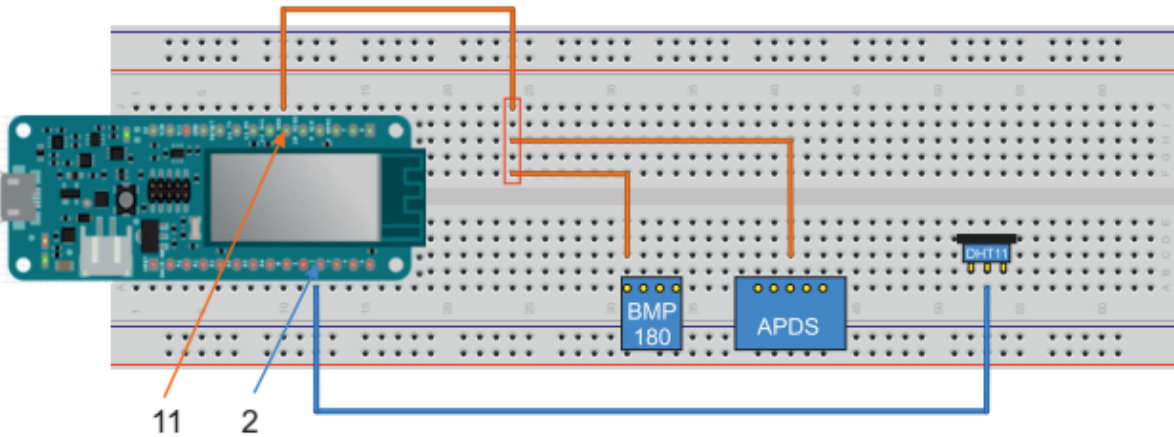
1. Arduino MKR1000 platformunu deney kartına yerleştirin.
2. BMP180 sensörünü deney kartına yerleştirin.
3. APDS sensörünü deney kartına yerleştirin.
4. DHT11 sensörünü deney kartına yerleştirin.



Şekil 4. Bağlantı, 1. adım

2.1.2. Bağlantı, 2. Adım (Şekil 5)

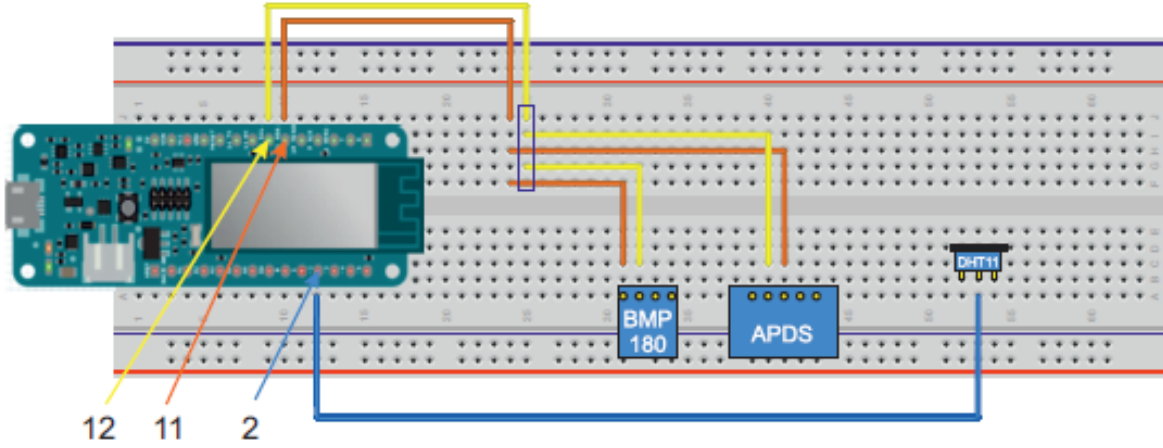
1. DHT 11 sensörünün orta bağlantı noktasını 2 numaralı Arduino kartı ile bağlayın (resimde mavi kablo).
2. Orta APDS sensörü bağlantı noktasını deney panosunun bir sütununa bağlayın (turuncu dikdörtgenle işaretli).
3. BMP180 sensörünün ilk (sol) bağlantı noktasını aynı deney panosu sütununa (turuncu bir dikdörtgen ile işaretlenmiş) bağlayın.
4. Bu sütunu (turuncu bir dikdörtgenle işaretlenmiş), 11 numaralı Arduino dijital giriş pimine ilave bir kabloyla bağlayın.



Şekil 5. Bağlantı, 2. adım - sensörleri Arduino MKR kartına bağlama

2.1.3. Bağlantı, 3. adım (Şekil 6):

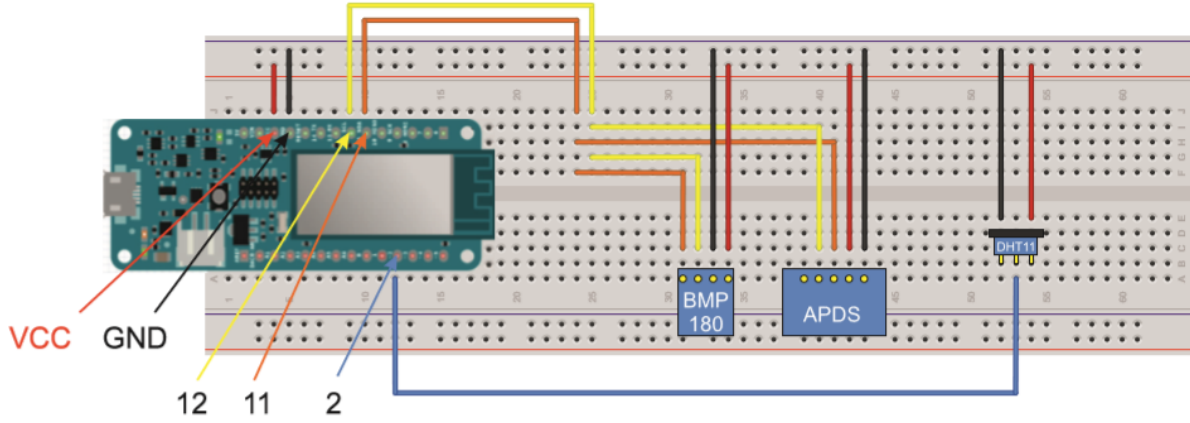
1. İkinci APDS sensörü bağlantı noktasını (sarı kablo) deney kartı sütununa (mavi bir dikdörtgen ile işaretlenmiş) bağlayın.
2. İkinci BMP180 sensör bağlantı noktasını (sarı kablo) aynı deney kartı sütununa (mavi dikdörtgen ile işaretlenmiş) bağlayın.
3. Sütunu (mavi bir dikdörtgenle işaretlenmiş) 12 numaralı Arduino dijital giriş pimine ek bir kabloyla bağlayın.



Şekil 6. Bağlantı, 3. adım

2.1.4. Bağlantı, 4. adım (Şekil 7):

1. Arduino „VCC“ (5V) pimini deney kartındaki ortak (+) sıraya bağlayın (Şekil 8'deki kırmızı kablo).
2. Arduino „GND“ pimini deney kartındaki ortak (-) hatta bağlayın (Şekil 8'deki siyah kablo).
3. BMP180 sensöründeki üçüncü pimi deney kartındaki ortak (-) hatta bağlayın (Şekil 8'deki siyah kablo).
4. BMP180 sensöründeki dördüncü pimi deney kartındaki ortak (+) hatta bağlayın (Şekil 8'deki kırmızı kablo).
5. APDS sensöründeki dördüncü pimi deney kartındaki ortak (+) hatta bağlayın (Şekil 8'deki kırmızı kablo).
6. APDS sensöründeki beşinci pimi deney kartındaki ortak (-) kabloya bağlayın (Şekil 8'deki siyah kablo).



Şekil 3. Bağlantı, 4. adım

3. IOT meteoroloji istasyonu - ikinci bölüm

Bu proje, MKR1000 platformunda, havadaki mikro partikülleri (hava kalitesi) ölçmek için bir plantower ve sensör değerlerinin görselleştirilmesi için bir cep telefonu uygulaması oluşturmayı içermektedir. Meteoroloji istasyonu ünitesini yapmak için birkaç adım vardır.

3.1. Meteoroloji istasyonunun kurulumu - havadaki mikropartiküller

IOT meteoroloji istasyonunun ikinci kısmı hava ışıklarındaki mikropartikülleri ölçmek için bir sensörden oluşur (hava kalitesi) - Şekil 8.

Şekil 10'daki şema bahsedilen sensörün deney panosuna bağlı halini göstermektedir.



Şekil 8. Havadaki mikro partikülleri ölçmek için plantower sensörü

3.1.1. Bağlantı, 1. adım (Şekil 9):

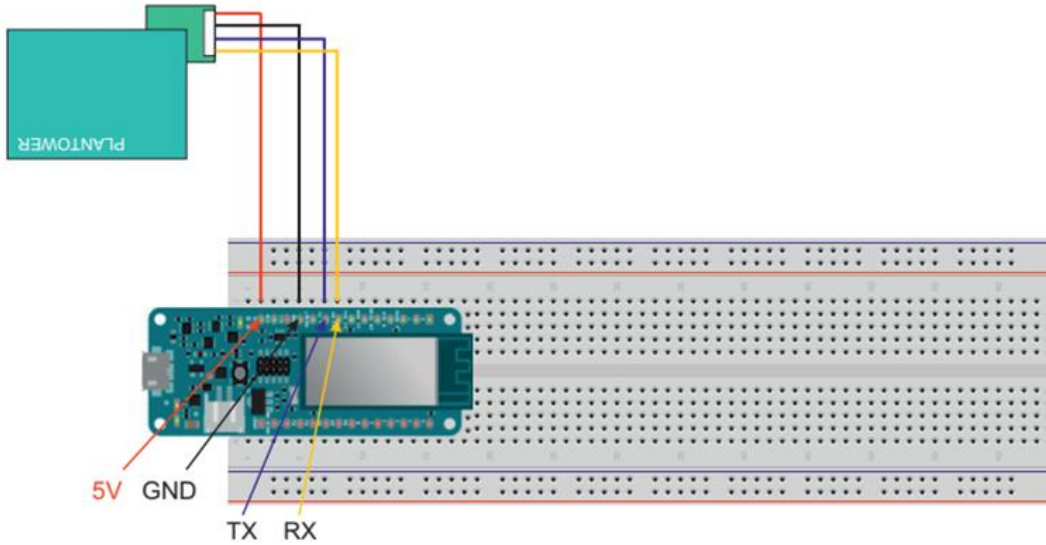
Beyaz bağlantı noktasını baskılı devre kartına takın ve plantower sensörüne monte edin. Baskılı devre kartını Plantower sensörüne uyacak şekilde yerleştirdiğinizden emin olun.



Şekil 9. baskılı devre kartını Plantower sensörüne bağlama

3.1.2. Bağlantı, 2. adım (Şekil 10):

1. Plantower sensörünün kırmızı kablosunu Arduino MKR 1000 kartındaki "VCC" (5V) pimine bağlayın.
2. Plantower sensörünün siyah kablosunu Arduino MKR 1000 anakartındaki "GND" pimine bağlayın.
3. Plantower sensörünün mavi kablosunu Arduino MKR 1000 anakartındaki dijital pim 14'e bağlayın.
4. Plantower sensörünün sarı kablosunu Arduino MKR 1000 anakartındaki dijital pim 13'e bağlayın.



Şekil 10. Plantower sensörünün Arduino MKR 1000 kartına bağlanması

Not:

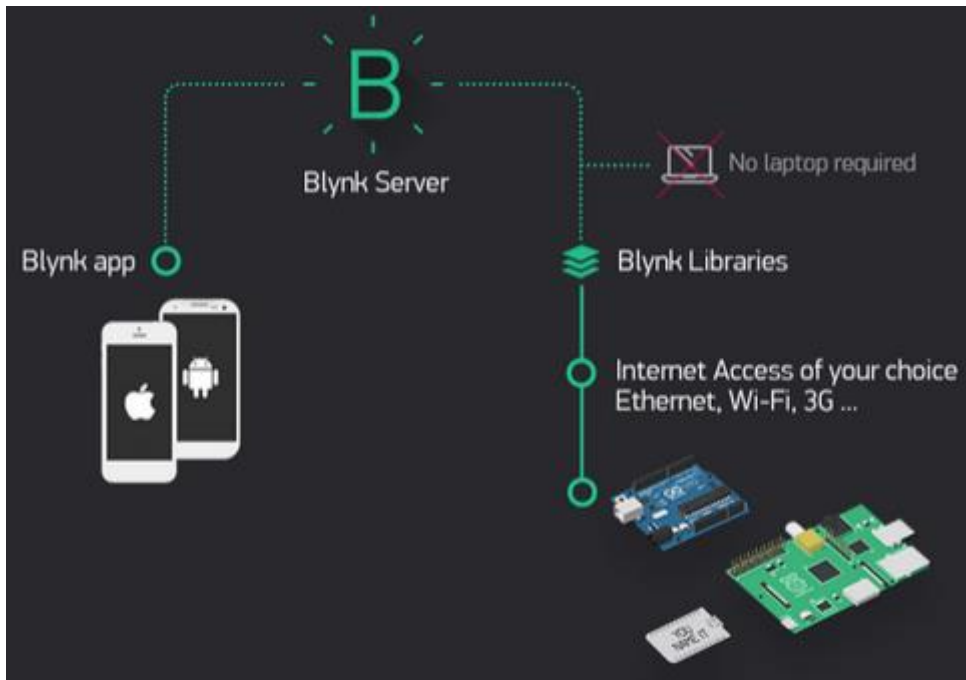
Plantower sensör kabloları, Arduino MKR 1000 panosuna bağlanmadan önce erkek-erkek kablolarla uzatılmalıdır.

4. Blynk platformu

Blynk, IoT için mobil ve web uygulamaları oluşturmayı basitleştirmeyi amaçlayan IoT açık kaynaklı bir platformdur. Donanımı uzaktan kontrol edebilir, sensör verilerini görüntüleyebilir, veri depolayabilir, görselleştirebilir ve birçok başka şey yapabilir. Blynk platformu, Arduino, ESP8266, ESP32, Raspberry Pi ve benzeri MCU'lar ve iOS ve Android için sürükle ve bırak IOT mobil uygulamaları gibi 400'den fazla donanım modelini kolayca bağlayabilir.

Platformda üç ana bileşen vardır:

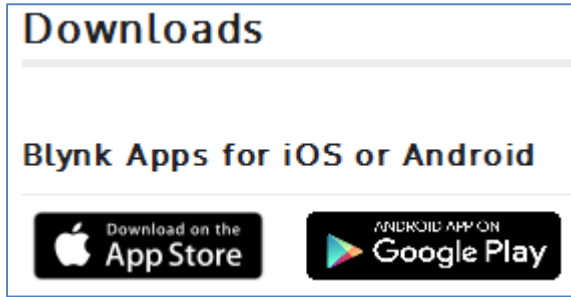
1. **Blynk Uygulaması** - çeşitli araçlar kullanarak projeler için harika arayüzler yaratmaya izin verir.
2. **Blynk Sunucusu** - akıllı telefon ve donanım arasındaki tüm iletişimden sorumludur. Ayrıca, projeler için özel Blynk sunucusunu yerel olarak çalıştırmaya izin verir. Blynk sunucusu binlerce cihazı kolayca idare edebilir ve bir Raspberry Pi'de de çalıştırılabilir.
3. **Blynk Kütüphaneleri** - sunucu ile iletişimi sağlar ve gelen ve giden tüm komutları işler.



Şekil 4. Blynk platformu

4.1. Uygulamayı İndirme

İlk projeye başlamadan önce Blynk uygulamasını bir akıllı telefona veya tablete indirmek ve kurmak gerekir. Android işletim sisteminde çalışıyorsanız, uygulama Google Mağazası'ndan (Şekil 12.) veya iOS işletim sisteminde çalışıyorsanız Uygulama Mağazasından (Şekil 12) indirilebilir. Uygulama "Blynk" anahtar kelimesi aratılarak bulunabilir.



Şekil 12. Akıllı telefonlar için Blynk işletim sistemi

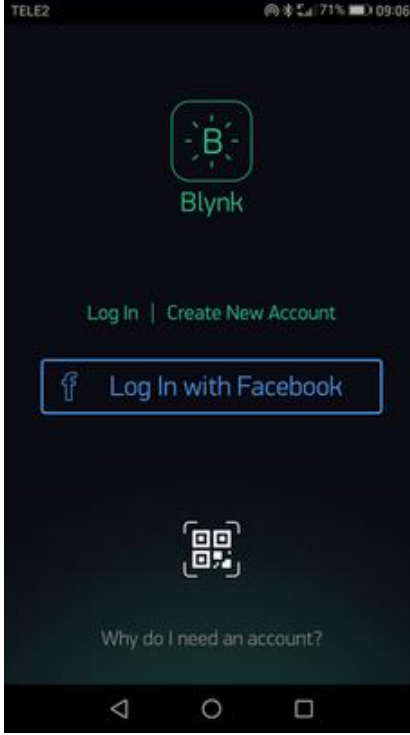
İndirmek için bağlantılar:

Android: <https://play.google.com/store/apps/details?id=cc.blynk>

IOS: <https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8>

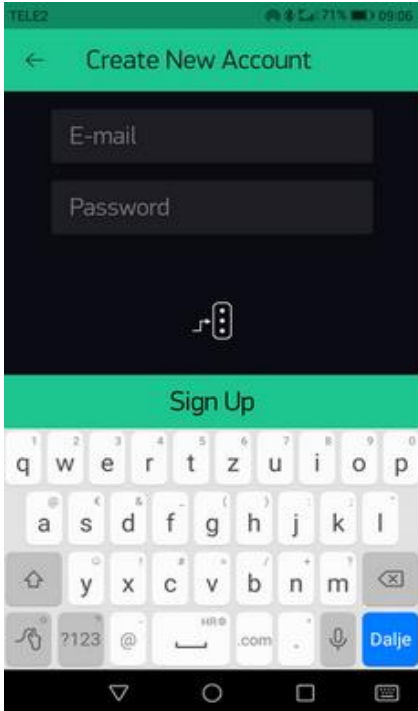
4.2. Projenin Oluşturulması

İndirmeyi tamamladıktan sonra, uygulamayı açın ve "Yeni Hesap Oluştur" u tıklayarak bir kayıt oluşturun (Şekil 13.). Projeleri doğrudan Blynk sunucusunda hazırlamak için bir kayıt gereklidir.



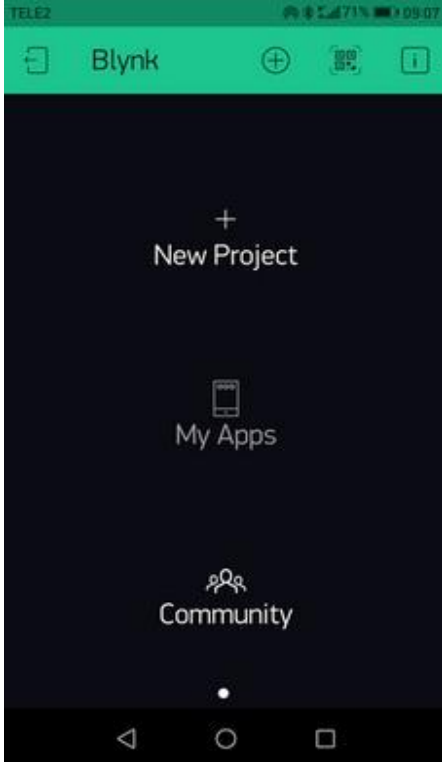
Şekil 13. Blynk uygulaması - hesap oluşturma

Kayıt için e-mail ve şifrenizi girin.



Şekil 5. Blynk uygulaması - email ve şifre

Kayıt olduktan sonra, “Yeni Proje” seçeneğini seçerek yeni bir proje yaratacağız (Şekil 15.).



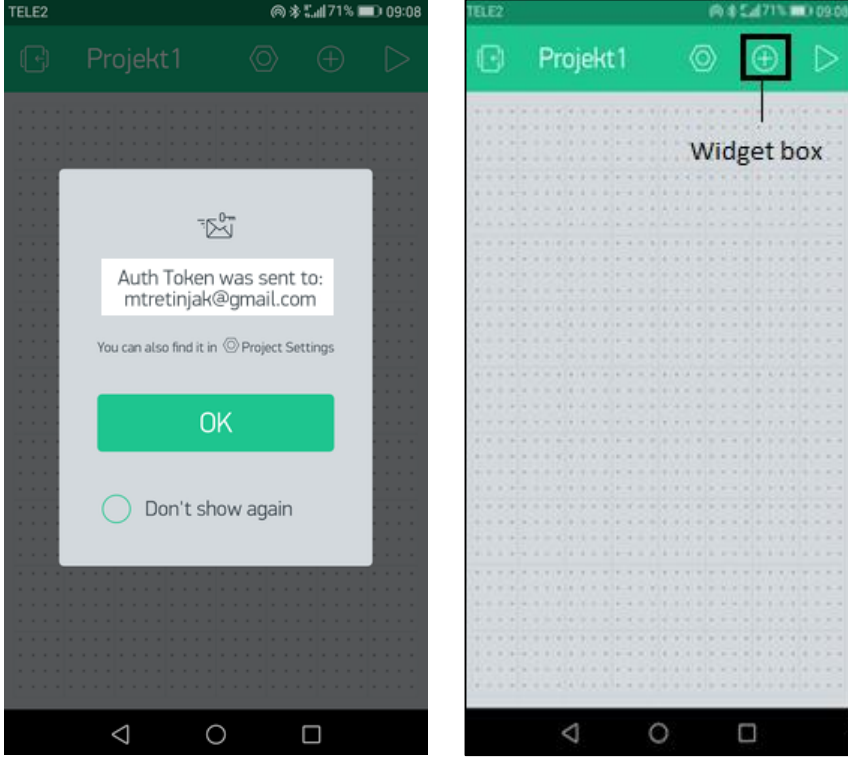
Şekil 6. Blynk uygulaması – yeni proje oluşturma

Temel proje bilgilerini girin - proje adını, kullandığınız donanımı (Arduino MKR1000), bağlantı türünü (WiFi) ve arabirimin rengini seçin – açık renkli (Şekil 16.).



Şekil 7. Blynk uygulaması – proje ayarları

Proje oluşturulduktan sonra, kimlik doğrulama belirtecini (tokeninin) oluşturulduğunu ve e-posta adresine gönderildiğini söyleyen bir bildirim ekrana gelecektir. Bu belirteç (token) her projeye özgüdür ve donanımı Blynk uygulamasına bağlamak için kullanılır. Buna daha sonra Arduino program kodunda ihtiyaç duyulacaktır.



Şekil 8. Blynk uygulaması - kimlik doğrulama belirteci (token) ve widget kutusu

Doğrulama belirtecini (token) kayıtlı e-posta adresinizden, ilk başta proje 1'de sonrasında da proje 2'de kullanmak için alın ve not edin.

4.3. Blynk widget kutusu

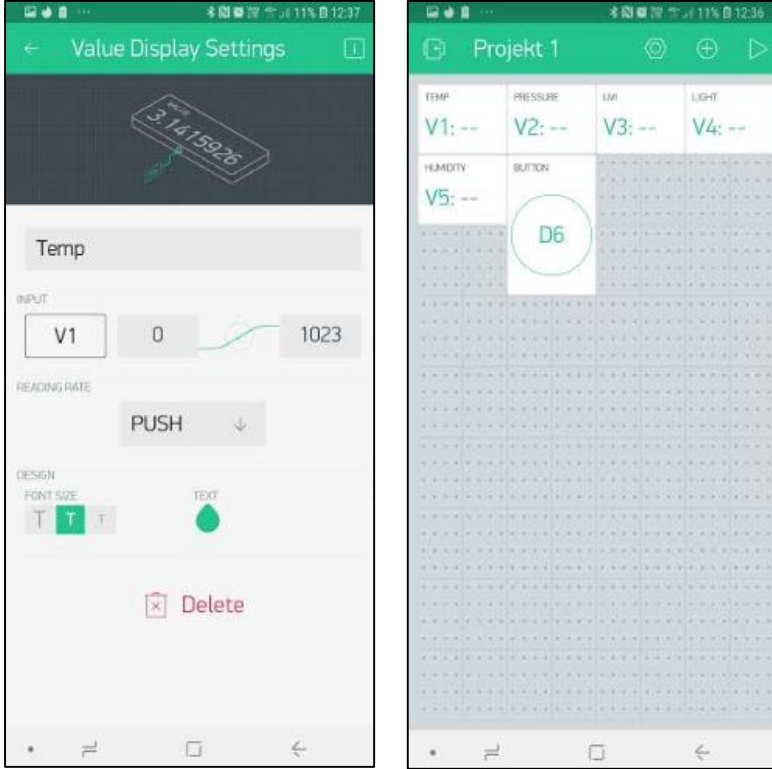
Blynk uygulama penceresinde proje 1 ve proje 2 için gerekli widget'ları aşağıdaki gibi ekleyin.

4.3.1. Proje 1 - IOT meteoroloji istasyonu (ilk bölüm)

- Değer ekran widget modülü ve Düğme widget modülünü proje 1'e ekleyin. Değer adına, ölçüm değeri metnini yazın ve değeri bir sonraki adımda aşağıdaki gibi gösterecek şekilde ayarlayın.
- 5 değer göstergesi de aşağıdaki listedeki gibi olana kadar 4. adımı birkaç kez tekrarlayın:

- ➔ Değer göstergesi - Sıcaklık - PIN: Virtual V1
- ➔ Değer göstergesi - Basınç - PIN: Virtual V2
- ➔ Değer göstergesi - UVI - PIN: Virtual V3
- ➔ Değer göstergesi - Işık - PIN: Virtual V4
- ➔ Değer göstergesi - Nem - PIN: Virtual V5

Projeye Arduino dijital pin - D6 üzerindeki LED'i açıp kapatacak bir düğme modülü ekleyin. SWITCH olarak ayarlayın.



Şekil 18. Blynk uygulaması – widget ekleme ve ayarlar

4.3.2. Proje 2 - IOT meteoroloji istasyonu (ikinci bölüm)

Üst araç çubuğundaki „+“ simgesine tıklayarak başka bir proje (proje 2) ekleyin ve ikinci proje için kimlik doğrulama kodunuzu yazmayı unutmayın. Proje adı altında, boşluk bırakmadan, "project2" yazın. Diğer veriler Proje 1 ile aynıdır.

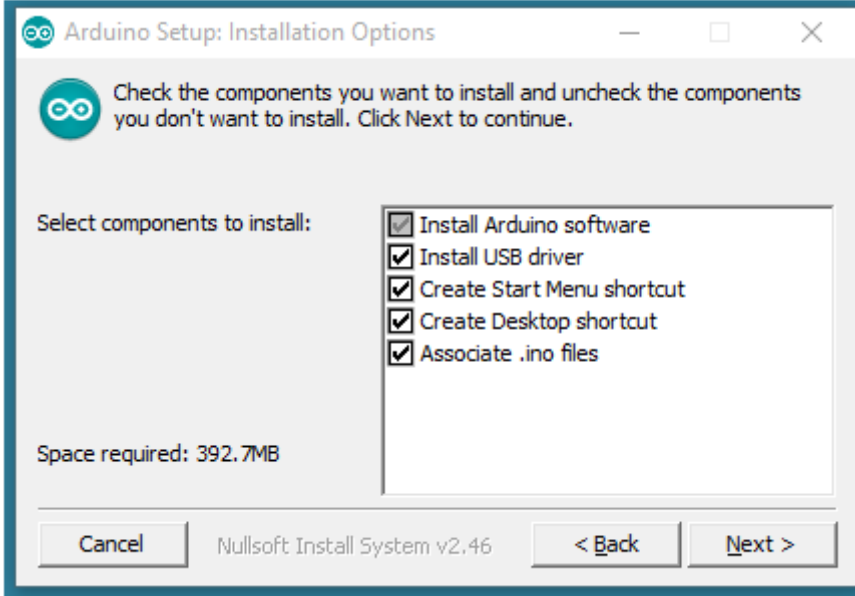
- Proje 2'ye de Proje 1'deki gibi Değer Göstergesi ekran modülü ekleyin.
 - ➔ Değer Göstergesi - PM 1 - PIN: Virtual V10
 - ➔ Değer Göstergesi - PM 2.5 - PIN: Virtual V11
 - ➔ Değer Göstergesi - PM 10 - PIN: Virtual V12

5. Arduino IDE programı

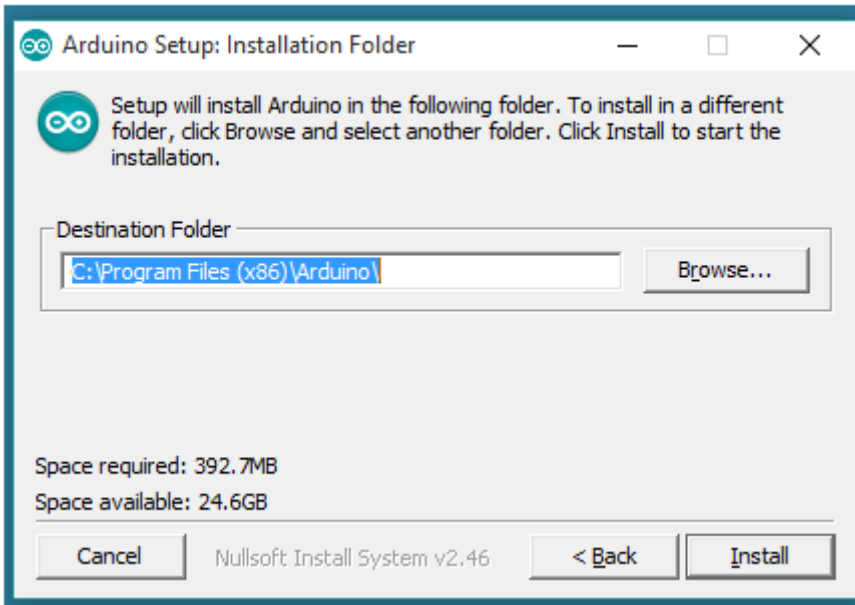
Açık kaynaklı Arduino Yazılımı (IDE) kod yazıp anakarta yüklemeyi kolaylaştırır. Windows, Mac OS X ve Linux'ta çalışmaktadır. Ortam, Java ile yazılmış olup işlemeye ve diğer açık kaynaklı yazılımlara dayanmaktadır. Bu yazılım herhangi bir Arduino kartı ile kullanılabilir.

5.1. Arduino IDE yazılımının indirilmesi

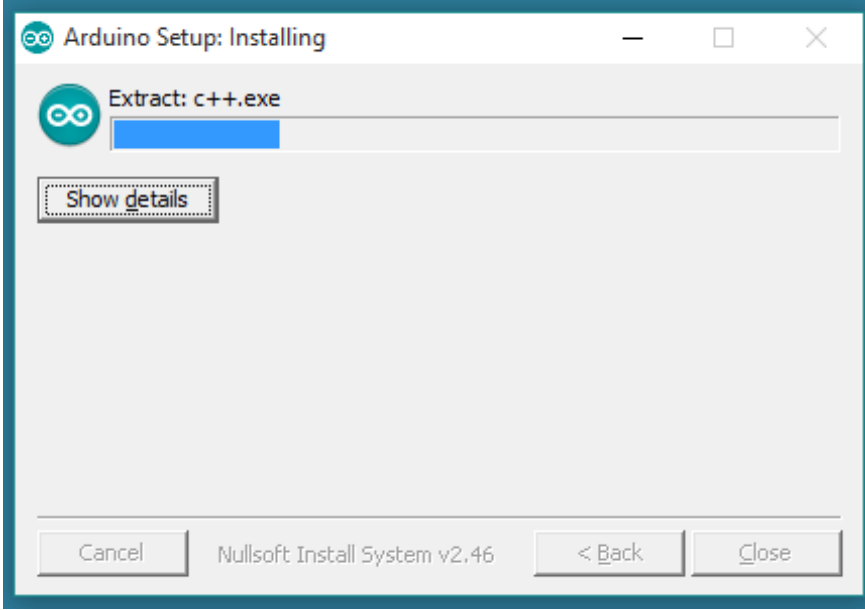
Arduino IDE yazılımının en son sürümünü indirme sayfasından edinin. Installer (.exe) aracılığıyla ya da Zip dosyası olarak indirebilirsiniz. Installer'ı kullanmak daha iyidir, çünkü sürücüler de dahil olmak üzere Arduino Yazılımını (IDE) kullanmak için gereken her şeyi doğrudan yükler. Zip paketi ile sürücülerini el ile yüklemeniz gerekir. Taşınabilir bir yükleme oluşturmak istiyorsanız, Zip dosyası da kullanışlıdır. İndirme işlemi tamamlandığında, yüklemeye devam edin ve lütfen işletim sisteminden bir uyarı aldığınızda sürücünün yükleme işleminin devam etmesine izin verin.



Yüklenecek bileşenleri seçin.



Kurulum dizinini seçin (varsayılan kalsın).



İşlem, Arduino Yazılımını (IDE) düzgün bir şekilde çalıştırmak için gerekli tüm dosyaları çıkarır ve kurar. Kurulum bittikten sonra, "Getting Started Home" kısmına geri dönebilir ve sayfanızın sağındaki listeden devre kartınızı seçebilirsiniz.

5.2. Arduino IDE yazılımının açıklaması

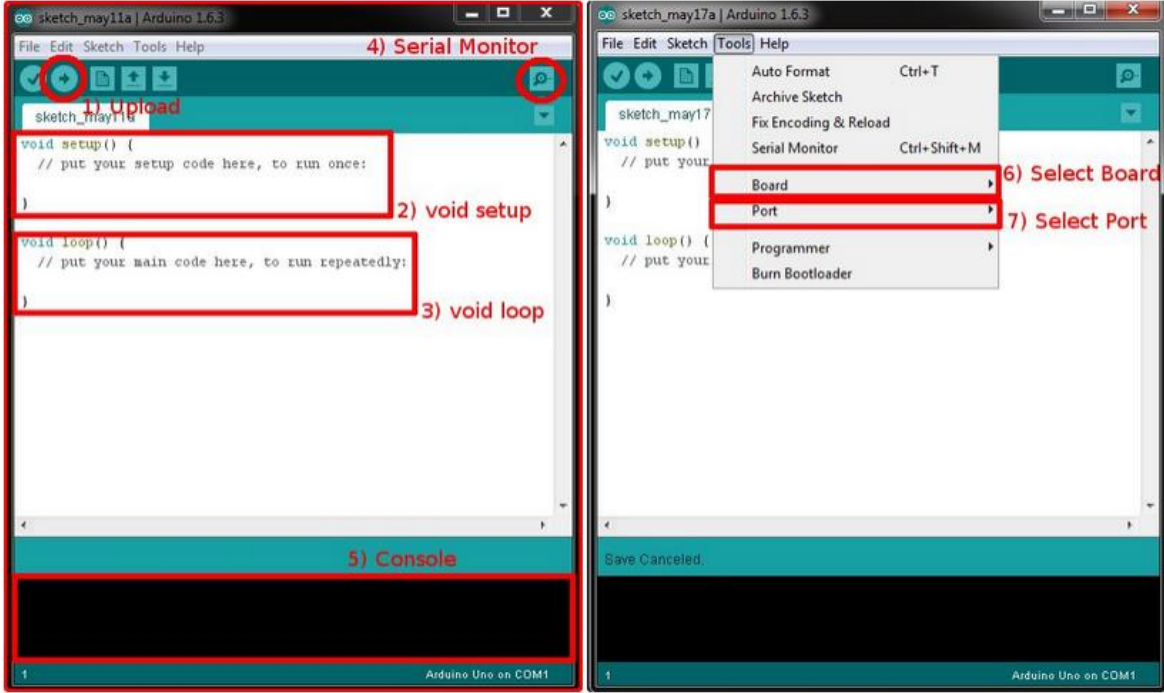
Arduino'yu kurduktan sonra, lütfen açın ve IDE'nin (Integrated Development Environment) birkaç önemli bölümünü gösterdiğim yukarıdaki fotoğraflara bakın.

1. 1 numaralı daire **yükleme(upload)** düğmesini gösteriyor. Arduino'nuz bilgisayara bağlıken, kodunuzu Arduino'ya yüklemek için bu düğmeyi tıklayın.
2. **Void setup**, kodunuzun yalnızca bir kez çalışacak kısmıdır. Arduino'nuz açıldığında veya yeniden başlatıldığında, void setup bölümüne girdiğiniz tüm kodları çalıştıracaktır.
3. **Void loop**, kodunuzun tekrar tekrar çalışacak bölümüdür. Void setup çalıştırdıktan sonra, void loop'daki tüm kodlarınız çalışır. Aradaki fark; void loop'daki tüm kodlarınız çalıştırdıktan sonra, Arduino void loop bölümünün başlangıcına geri döner ve herşeyi tekrardan çalıştırır. Bu durum defalarca tekrar etmektedir bu nedenle buna bir döngü (loop) denir.
4. **Serial monitor düğmesi**. Bu düğme serial monitor'u açar. Serial monitor, Arduino'ya komut göndermenizi ve Arduino'dan mesaj almanızı sağlar. Bunu nasıl yapacağınız hakkında daha fazla bilgi vereceğiz.

5. Burası **Arduino Konsolu**'dur. Arduino, devre kartına yükleme yaparken ya da kodu derlerken bir sorunla karşılaşırsa buraya bir hata mesajı yansıtır.

6. Üstteki **Araçlar (Tools)** menüsünde seçebileceğiniz bir "devre kartı" seçeneği vardır. Doğru devre kartı tipini seçmek için bunu kullanın. Genellikle Arduino kartının türü bir cihazınızın üzerinde bir yerde yazılıdır.

7. Burası, hangi **com port'un** Arduino'nuzu bilgisayara bağlayacağını seçme kısmıdır. Bunu bulmakta zorlanıyorsanız, Arduino'nuzu çıkarın, listeye bakın ve Arduino'nuzu tekrar takın. Yeni görünen com port, Arduino'nuz için olanıdır.



Şekil 9. Arduino IDE (Tümleşik Geliştirme) ortamı

5.3. Arduino programının ölçüm cihazlarına yüklenmesi ve test edilmesi

Projelerimiz için iki programımız bulunmaktadır; proje 1 için TVSmeasuring (TVSölçümü) ve proje 2 için PMSmeasuring (PMSölçümü). Programlar zip dosyası olarak Google Drive üzerinden paylaşılacaktır.

TTVSUmeasuring - ışık, sıcaklık, basınç ve nem ölçümü programı.

PMSmeasuring - havadaki mikropartikülleri ölçme programı.

1.3.1. 1.3.1. Projelerimize program ekleme, Proje 1 - TTVSU ölçümü

- ➔ Programı bilgisayarınıza yükleyin.
- ➔ Bilgisayarınızda Arduino IDE'yi açın.
- ➔ „TTVSUmeasuring“ programını açın.

1. Adım - taslak

Taslağın açılmasından sonra birkaç kütüphane kurmamız gerekiyor. İlk önce Sketch (taslak) sonra, Include Library (Kitaplığı Dahil Et), ardından da Manage Libraries (Kitaplıkları Yönet) ögesini tıklayın. Aşağıdaki kütüphanelerin sizde bulunması gerekmekte. Eğer yoksa, yüklemeniz gerekir:

- ➔ Blynk,
- ➔ WiFi101,
- ➔ Adafruit BMP085,
- ➔ SimpleDHT,
- ➔ PMS.

Google dokümanlarda, başka bir kütüphane daha bulacaksınız - APDS9200 (.zip dosyası). Bu kütüphaneyi de bilgisayarınızda bir yere kaydedin

Önce Sketch'e (Taslak), sonra Include Library'e (Kütüphane Ekle), ardından Add zip library'e (.zip ekle) tıklayın ve APDS9200.zip dosyasını açın.

```
#include <Wire.h> //using library wire.h
#include <Adafruit_BMP085.h> //using library for reading BMP180 sensor
#include "APDS9200.h" //using library for reading light level sensor
#include <SimpleDHT.h> //using library for DHT11 sensor

#include <SPI.h> //using library for SPI communication
#include <WiFi101.h> //using WiFi 101 library for internet connection
#include <BlynkSimpleMKR1000.h> //using library for Blynk application

int pinDHT11 = 2; //DHT11 sensor pin connected to digital pin 2

Adafruit_BMP085 bmp; //define object bmp
APDS9200 light; //define object light
SimpleDHT11 dht11; //define object dht11

BlynkTimer timer; //using Blynk timer

double temp180[6]; //variable for measuring temperature
double temp180sr = 0; //variable for measuring temperature
double tlak180[6]; //variable for measuring pressure
double tlak180sr = 0; //variable for measuring pressure

double uv[6]; //variable for measuring uv
double uvindexsr = 0; //variable for measuring uv
double svjetlo[6]; //variable for measuring level of light
double svjetlosr = 0; //varijable za mjerenje level of light

double vlagall[6]; //variable for measuring humidity
double vlagallsr = 0; //variable for measuring humidity

int brojac = 0; //varijable counter

char auth[] = " "; //enter the autentification token for project 1

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = " "; //enter the network name
char pass[] = " "; //enter the network password
```

```

void setup() {
  Serial.begin(9600); //serial connection initialization
  if (!bmp.begin()) { //checking of MBP sensor initialization
    Serial.println("BMP senzor nije pronađen"); //stop the program
    while (1) {}
  }
  Wire.begin(); //starting the wire communication

  Blynk.begin(auth, ssid, pass); //starting the Blynk servis
  timer.setInterval(10000L, myTimerEvent); //setting the values interval on 10 sec.
  brojac = 0; //setting the starting value of the counter
}

void myTimerEvent() //function active every 10 sec. using timer
{
  if (brojac < 6) {

    //----TEMPERATURE MEASUREMENT AND PRINTING THE VALUE-----TEMPERATURE MEASUREMENT AND PRINTING THE VALUE-----
    temp180[brojac] = bmp.readTemperature();
    tlak180[brojac] = bmp.readPressure();
    temp180sr = temp180sr + temp180[brojac];
    tlak180sr = tlak180sr + tlak180[brojac];

    Serial.println("-----BMP180");
    Serial.print("Temperature = ");
    Serial.print(temp180[brojac]);
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(tlak180[brojac]);
    Serial.println(" Pa");
    //-----

    //----LIGHT LEVEL MEASUREMENT AND PRINTING THE VALUE-----LIGHT LEVEL MEASUREMENT AND PRINTING THE VALUE-----
    light.enableUV();
    delay(500);
    uv[brojac] = light.getUV();
    light.enableLight();
    delay(500);
    svjetlo[brojac] = light.getLight();
    uvindexsr = uvindexsr + uv[brojac];
    svjetlosr = svjetlosr + svjetlo[brojac];
    Serial.println("-----UV&LIGHT");
    Serial.print("UV:");
    Serial.println(uv[brojac]);
    Serial.print("Svjetlo:");
    Serial.println(svjetlo[brojac]);
    //-----

    //----HUMIDITY MEASUREMENT AND PRINTING THE VALUE-----HUMIDITY MEASUREMENT AND PRINTING THE VALUE-----
    byte temperature = 0;
    byte humidity = 0;
    int err = SimpleDHTErrSuccess;
    if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
      //Serial.print("Read DHT11 failed, err="); Serial.println(err); delay(1000);
      //return;
    }

    //temp11 = (double)temperature;
    vlagall[brojac] = (double)humidity;
    vlagallsr = vlagallsr + vlagall[brojac];
    Serial.println("-----DHT11");
    Serial.print(vlagall[brojac]); Serial.println(" %");
    //-----

    brojac++;
  }
}

```

```
} else {
  //----TEMPERATURE AND PRESSURE MEASUREMENT AND PRINTING THE VALUE-----
  templ80sr = temp180sr / 6;
  tlakl80sr = tlak180sr / 600;
  Blynk.virtualWrite(V1, templ80sr);
  Blynk.virtualWrite(V2, tlakl80sr);

  Serial.print("Srednja temp = ");
  Serial.print(templ80sr);
  Serial.println(" *C");
  Serial.print("Srednji tlak = ");
  Serial.print(tlakl80sr);
  Serial.println(" hPa");

  templ80sr = 0;
  tlakl80sr = 0;
  //-----

  //----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----
  uvindexsr = uvindexsr / 6;
  svjetlosr = svjetlosr / 6;
  if ((uvindexsr >= 0) && (uvindexsr < 275)) {
    uvindexsr = 1;
  } else if ((uvindexsr >= 275) && (uvindexsr < 525)) {
    uvindexsr = 2;
  } else if ((uvindexsr >= 525) && (uvindexsr < 775)) {
    uvindexsr = 3;
  } else if ((uvindexsr >= 775) && (uvindexsr < 1050)) {
    uvindexsr = 4;
  } else if ((uvindexsr >= 1050) && (uvindexsr < 1325)) {
    uvindexsr = 5;
  } else if ((uvindexsr >= 1325) && (uvindexsr < 1675)) {
    uvindexsr = 6;
  } else if ((uvindexsr >= 1675) && (uvindexsr < 1960)) {
    uvindexsr = 7;
  } else if ((uvindexsr >= 1960) && (uvindexsr < 2240)) {
    uvindexsr = 8;
  } else if ((uvindexsr >= 2240) && (uvindexsr < 2520)) {
    uvindexsr = 9;
  } else if ((uvindexsr >= 2520) && (uvindexsr < 2800)) {
    uvindexsr = 10;
  } else if ((uvindexsr >= 2800) && (uvindexsr < 3080)) {
    uvindexsr = 11;
  } else {
    uvindexsr = 12;
  }
}

Blynk.virtualWrite(V3, uvindexsr);
Blynk.virtualWrite(V4, svjetlosr);

Serial.print("UV index = ");
Serial.println(uvindexsr);
Serial.print("Svjetlost = ");
Serial.println(svjetlosr);

uvindexsr = 0;
svjetlosr = 0;

//-----

//----HUMIDITY MEASUREMENT-----HUMIDITY MEASUREMENT-----HUMIDITY MEASUREMENT-----
vlagallsr = vlagallsr / 6;

Blynk.virtualWrite(V5, vlagallsr);

Serial.print("Srednja vlaga: ");
Serial.print(vlagallsr); Serial.println(" %");
```

```
vlagallar = 0;
//-----

brojac = 0; //setting counter to 0
}

}

void loop() {
  Blynk.run(); //function for server connection and sending data
  timer.run(); //timer function
}
```

Şekil 10. Proje 1 için program kodu

2. Adım – Arduino'nun bağlanması (proje 1)

Işık, sıcaklık, basınç ve nemi ölçmek için Arduino ölçüm aletini bağlayın. Arduino'nun programlama ortamında, Tools-> Board kısmında Arduino MKR1000'in seçildiğinden emin olun.

Kayıtlı e-postanızda, ilk proje için yetkilendirme kodunu (token) almış olacaksınız ve bu taslakta bunu yazmanız gerekecek.

```
char auth[] = ""; // insert the blink token into the quotation marks
char ssid[] = ""; // insert the wifi network name into the quotation marks
char pass[] = ""; // insert the password for connection to the wifi newtwork into
the quotation marks
```

Programı Arduino kartına yükleyin. Yüklemeyi tamamladıktan sonra, Arduino program ortamının sağ üst köşesindeki büyüteç simgesine tıklayarak Serial Monitor'u (Seri Monitör) açın.

Yeni bir pencerede, bağlı olan tüm sensörlerden gelen ölçüm değerlerini alıyor olmalısınız. Değerler her 10 saniyede bir gösterilecektir. Blynk uygulamasında yapılan ilk taslakta, değerlerin güncellendiğinden emin olun (her 60 saniyede bir).

5.3.2. 2. projeye program yükleme– *PMSmeasurement(PMSölçümü)*

1. Adım –Arduinonun bağlanması (proje 2)

Havadaki mikro partikülleri (hava kalitesi) ölçmek için Arduino MKR 1000'i bağlayın. Arduino programlama ortamında, Tools-> Board altında Arduino MKR1000'in seçildiğinden emin olun.

Kayıtlı e-postanızda, ikinci proje için yetkilendirme belgesini almış olacaksınız ve taslakta bunu yazmanız gerekecek.

```
char auth[] = ""; // insert the blink token into the quotation marks
char ssid[] = ""; // insert the wifi network name into the quotation marks
char pass[] = ""; // insert the password for connection to the wifi newtwork into
the quotation marks
```

Programı Arduino kartına yükleyin. Yüklemeyi tamamladıktan sonra, Arduino program ortamının sağ üst köşesindeki büyüteç simgesine tıklayarak Serial Monitor'u (Seri Monitör) açın..

```
#include "PMS.h"

#include <SPI.h>
#include <WiFi101.h>
#include <BlynkSimpleMKR1000.h>

PMS pms(Serial1);
PMS::DATA data;

double pm1sr=0;
double pm25sr=0;
double pm10sr=0;

char auth[] = "";
char ssid[] = "";
char pass[] = "";

void setup()
{
  Serial.begin(9600);
  Serial1.begin(9600);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  if (pms.read(data))
  {
    pm1sr = data.PM_AE_UG_1_0;
    pm25sr = data.PM_AE_UG_2_5;
    pm10sr = data.PM_AE_UG_10_0;

    Serial.println("Data:");

    Serial.print("PM 1.0 (ug/m3): ");
    Serial.println(pm1sr);

    Serial.print("PM 2.5 (ug/m3): ");
    Serial.println(pm25sr);

    Serial.print("PM 10.0 (ug/m3): ");
    Serial.println(pm10sr);

    Serial.println();

    Blynk.virtualWrite(V10, pm1sr);
    Blynk.virtualWrite(V11, pm25sr);
    Blynk.virtualWrite(V12, pm10sr);
  }
}
```

Şekil 11. Proje 2 için program kodu

Yeni bir pencerede, bağlı olan tüm sensörlerden gelen ölçüm değerlerini alıyor olmalısınız. Değerler her 10 saniyede bir gösterilecektir. Blynk uygulamasında yapılan ilk taslakta, değerlerin güncellendiğinden emin olun (her 60 saniyede bir).

5.3.3. Program kodlarını birleştirme

Ek olarak, her iki projedeki program kodları tek bir projede birleştirilebilir. Programları birleştirmek, diğer projeler için bir MKR1000 platformununun boşta kalmasını sağlar. Her iki projedeki ölçülen değerlerin birleştirilmesinden sonra, bu değerler Blynk uygulamanızda ve Arduino IDE programında serial monitor'de (seri monitör) gösterilecektir (Şekil 22).



Şekil 22. Blynk uygulaması ile ölçülen değerler