

INF340 Mikroişlemciler Proje Raporu

GENEL AMAÇLI IOT CİHAZI

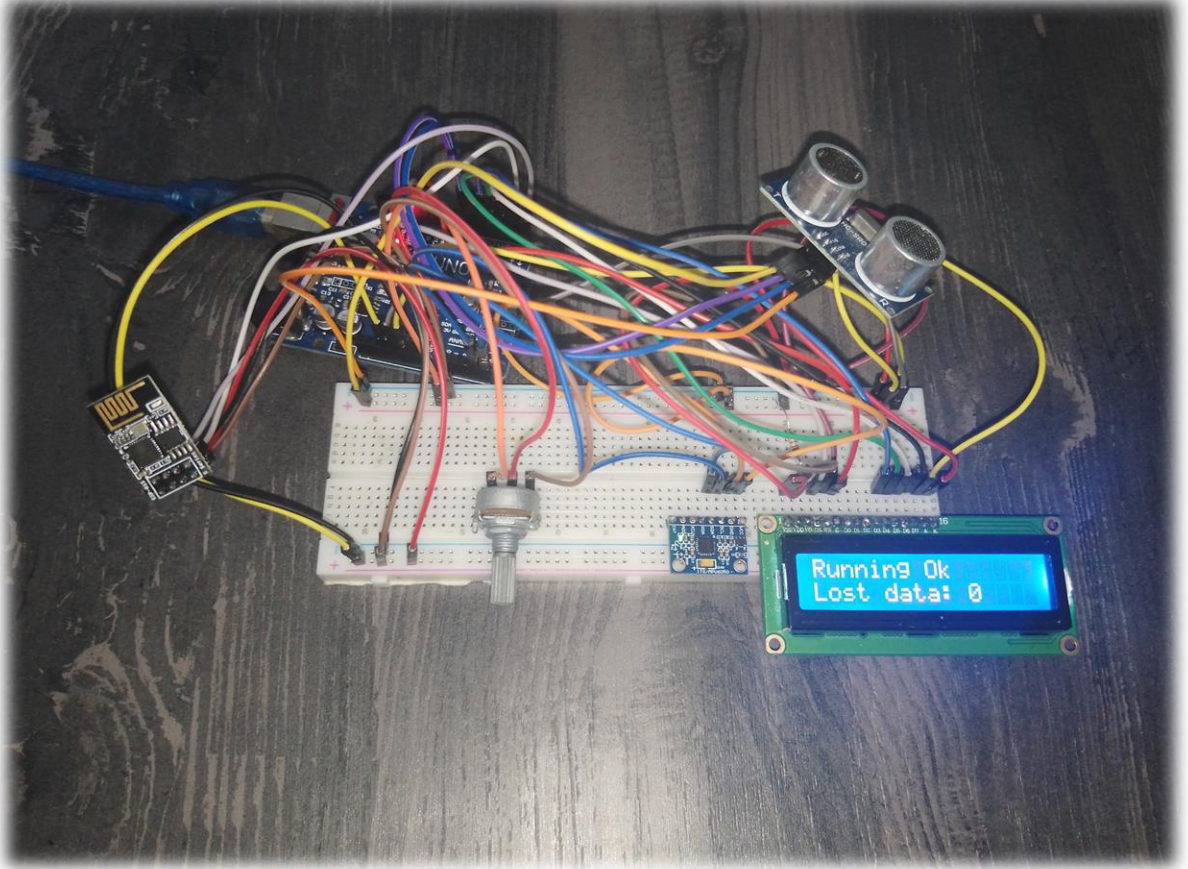


ŞÜKRÜ ÇİRİŞ 18401785

GALATASARAY ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

İçindekiler:

1. Proje Amacı
2. Projenin Çalışma Prensipleri ve Akış Şeması
3. Projede Kullanılan Malzemeler
4. Projede Kullanılan Yazılım ve Donanım Altyapısı
5. Projenin Teknik Tasarımı
6. Projede Karşılaşılan Problemler ve Çözümleri
7. Projeyi İyileştirme Önerileri



Prototipin fotoğrafı

Proje Amacı

Çeşitli fabrikalar ve atölyelerdeki çeşitli üretim makinelerinin nasıl, ne zaman, ne kadar süre çalıştığını takip etmek zor bir iştir, eğer makinenin verisini toplayan kendi içinde bir sistemi yoksa veya makine tamamen mekanik, elle çalışan bir makineyse. Genelde bu tür işler için orada çalışan insanlara kaydetme sorumluluğu verilir. Bu da verimsiz ve güvenilmez bir iştir, kaydı tutan kişi hata yapabilir ve yavaştır. Üretimin performansını ve kalitesini artırma çalışmaları yapılabilinmesi için önce doğru bir şekilde ölçülmesi gereklidir bu yüzden kaydetme adımı oldukça önemlidir. Mesela üretim hattındaki darboğazların bulunup çözülmesi üretimin performansını çok yüksek miktarlarda arttırabilir. Benim projem bu tarz makinelerin verilerini toplama sorununa bir IOT cihazı yaparak çözüm getirmeyi amaçlıyor. Projede yapılan cihaz makinenin kilit bir parçasının üstüne sabitleniyor ve istenilen adrese istenilen frekansta çeşitli sensörlerden toplanan verileri göndermeye başlıyor. Küçük bir cihazın istenilen makineye sabitlenmesiyle, bir kaç dakika içinde makinenin buluta veri göndermeye başlayıp Endüstri 4.0'a giriş yapmasını hedeflemektedir projem. Projenin çalışma videosunu buradan izleyebilirsiniz:

<https://www.youtube.com/watch?v=BQkPBPlkSHU>.

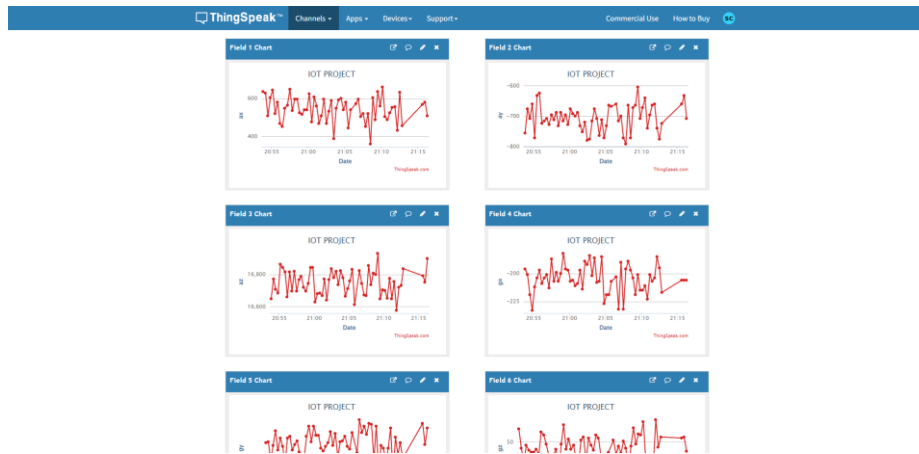
Projenin Çalışma Prensipleri ve Akış Şeması

Cihaz makinenin sensörleri ile anlamlı veri alabileceği bir yere takıldıktan sonra, istenen adrese bu verilen işlenip anlamlandırılıp kaydedilmesi için göndericek. Mesela pistonlu bir makinenin pistonuna takıldıktan sonra cihazın ölçtüğü ivme, mesafe verisi ve sinyal işleme ile makinenin ne kadar süre, ne zaman, ne kadar çalıştığı gibi sorulara cevap verebilir.

Prototipin çalışmaya başlaması için önce fişe takılması gerekiyor. Fişe takılınca 1602A LCD ekran başlatılır. Fişe takıldıktan sonra potansiyometre aracılığı ile kullanıcıdan girdi alır. Eğer girdi cihazın kapanması yönündeyse LCD ekranda IDLE yazar ve cihaz herhangi bir şey yapmaz. Eğer kullanıcı cihazı açarsa LCD ekranda STARTING yazar ve cihazın başlatma işlemleri yapılmaya başlanır.

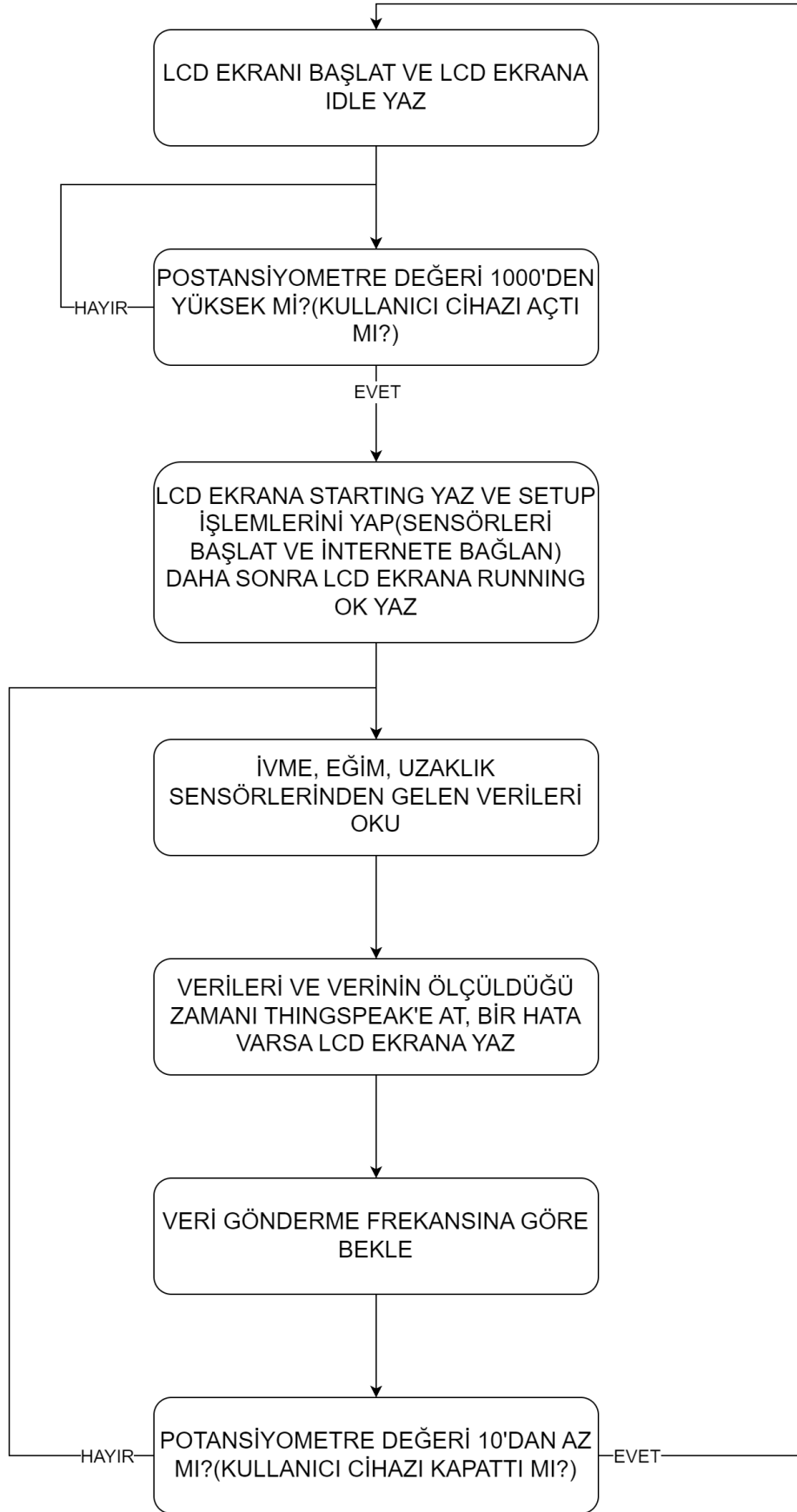
MPU6050 ivme ve eğim ölçer, ESP8266 WIFI modülü ve HC-SR04 ultrasonik mesafe sensörü başlatılır. İnternete kodda verilen bilgiler doğrultusunda bağlanır. Eğer başlatma işlemleri sırasında bir problem olursa sorun LCD ekrana yazdırılır ve cihaz orada durur. Kullanıcının cihazı yeniden başlatması gerekiyor o durumda. Eğer başlatma işlemleri hatasız bitmişse LCD ekrana RUNNING OK yazılır ve sonsuz döngü kısmına geçilir.

Projeyi fiziksel olarak gerçekleştirirken verileri Thingspeak isimli IOT projeleri için arayüz sağlayan kullanımı bedava bir siteye göndermeye karar verdim. Thingspeak en fazla on beş saniyede bir veri göndermeye izin veriyor bu yüzden prototipin kodunda veri gönderme periyodunu yirmi saniye olarak belirledim. Projenin Thingspeak kanal linki: <https://thingspeak.com/channels/2149604>. Sonsuz döngü kısmında her döngü içinde ivme, eğim, uzaklık ölçümleri ve ilk verinin gönderilme zamanını 0 kabul ederek milisaniye cinsinden ölçümün yapıldığı zaman Thingspeak'e atılır. Eğer herhangi bir aşamada hata olursa, sorun LCD ekrana yazılır ve devam edilir. Aynı zamanda kaç tane ölçümün gönderilmede başarısız olduğunun sayısı tutulur ve LCD ekranda gösterilir. Döngü içinde kullanıcıdan potansiyometre aracılığı ile kapatma girdisinin gelip gelmediğine bakılır, eğer geldiyse Arduino UNO yeniden başlatılır ve IDLE durumuna geçer. Her döngünün sonunda kodda belirtilen periyoda göre program bekler.



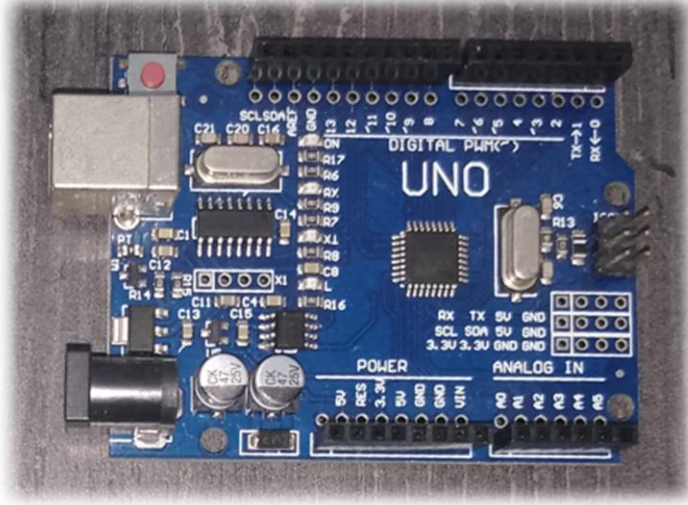
Verilerin Thingspeak'te görünümü

Sonraki sayfada projenin akış şeması yer almaktadır.



Projede Kullanılan Malzemeler

1. Arduino UNO



Projemde mikrodenetleyici olarak Arduino UNO kullandım.

2. Breadbord

Bağlantıları kolayca yapmak için bir breadbord kullandım.

3. Jumper Kablolar

Bağlantıları breadbord üzerinde yapmak için jumper kablolar kullandım.

4. 1K Ohm Direnç

LCD Ekranın V0 bacağı ile GND arasına koymak için kullandım. LCD ekranın kontrastını ayarlamak için gerekli oldu.

5. Potansiyometre



Kullanıcıdan giriş almak için potansiyometre kullandım. Okunan değer 1000'den fazla ise cihaz çalışıyor, 10'dan az ise bekleme durumuna geçiyor.

6. 1602A LCD Ekran



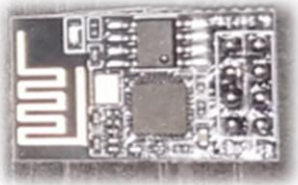
Cihazın bulunduğu durum, verdiği hata ve bunun gibi çeşitli şeyleri yazdırarak kullanıcıya cihazla ilgili bilgi verme ve karşılaştığı sorunu çözmesini kolaylaştırmak amacıyla bir LCD ekran kullandım.

7. MPU6050 İvme Ölçer ve Jiroskop



İvme ve eğim verilerini ölçmek için MPU6050 kullandım. İvme ve eğim bilgisi bir çok makinenin bulunduğu durumu tahmin etmede kullanılabileceği için oldukça gerekli bir sensör.

8. ESP8266 WIFI Modülü



Ölçülen verileri internetteki istenen adrese göndermek için cihazın internete bağlanması zorunlu. Bunun için ESP8266 kullandım.

9. HC-SR04 Ultrasonik Mesafe Sensörü

Bazı makinelerde işe yarayabileceğini düşündüğüm için HC-SR04 mesafe sensörü kullandım, çünkü ivme sensöründe hep hata payı ve drifting olur. Verileri işlerken mesafe gerekirse, direkt bu sensörün verisi kullanılabilir.



Projede Kullanılan Yazılım ve Donanım Altyapısı

Cihazın gömülü yazılımını Arduino IDE'sinde yaparken “MPU6050.h”, “LiquidCrystal.h” ve “Wire.h” kütüphanelerini kullandım. Projenin kodunu buradan görebilirsiniz:

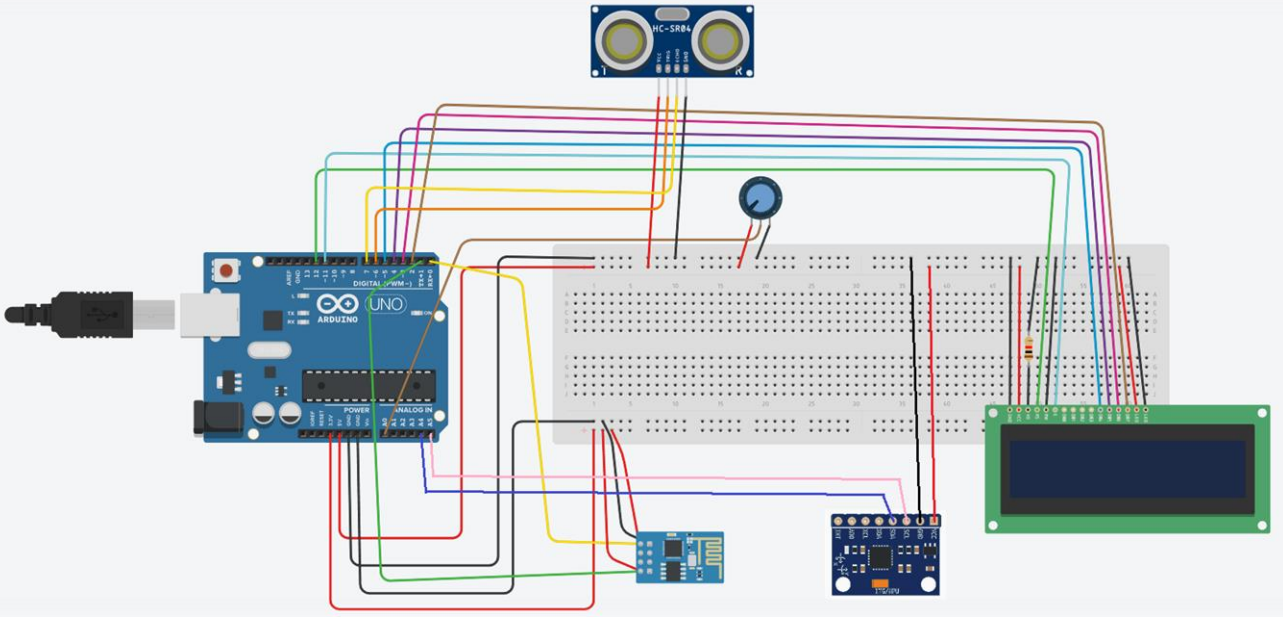
<https://github.com/SUKRUCIRIS/IOT-PROJECT-GSU>.

Verilerip gönderip görüntülemek için Thingspeak sitesini kullandım.

Donanım olarak Arduino UNO mikrodenetleyecisini ve onunla uyumlu modüller kullandım.

Projenin Teknik Tasarımı

Tüm modüller birbiriyle Arduino Uno aracılığıyla haberleşmektedir. Nasıl bağlandıklarını Tinkercad üzerinde çizdiğim aşağıdaki görselde görebilirsiniz. Ayrıca görselin altında her bağlantıyı yazdım.



Potansiyometre bağlantıları:

Sol ucu – 5V

Sağ ucu – GND

Orta ucu – A0, Arduino

ESP8266 bağlantıları:

RX ucu – TX, Arduino

TX ucu – RX, Arduino

GND ucu – GND

3V3 ucu – 3.3V

EN ucu – 3.3V

MPU6050 bağlantıları:

VCC ucu – 5V

GND ucu – GND

SCL ucu – A5, Arduino

SDA ucu – A4, Arduino

HC-SR04 bağlantıları:

VCC ucu – 5V

GND ucu – GND

TRIG ucu – 6, Arduino

ECHO ucu – 7, Arduino

1602A LCD bağlantıları:

K ucu – GND

A ucu – 5V

D7 ucu – 2, Arduino

D6 ucu – 3, Arduino

D5 ucu – 4, Arduino

D4 ucu – 5, Arduino

E ucu – 11, Arduino

RW ucu – GND

RS ucu – 12, Arduino

V0 ucu – 1K Ohm – GND

VDD ucu – 5V

VSS ucu – GND

Projede Karşılaşılan Problemler ve Çözümleri

ESP8266 modülünü Arduino UNO ile donanımsal serial haberleşme sağlayacak şekilde bağladığım için serial monitörü debug etmek için kullanamıyordum. Bu sorunu LCD ekrana cihazın durumu yanında alınan hataları da yazdırarak çözdüm. LCD ekrandaki bilgiler sayesinde kodu debug edebilmeye başladım.

LCD ekranın lehimlerini yaptıktan sonra çalıştırmayı denediğimde çalışmadığını gördüm, lehimlerin birbirine değdiğini düşünerek lehimleri temizleyip tekrar yaptım ama sorun yine çözülmedi. Bir kaç gün sonra breadbord'un bir kısmında temassızlık olduğunu ve bu yüzden LCD ekranın çalışmadığını farkettim. Bu sorunu yeni bir breadbord alarak çözdüm.

Projeyi İyileştirme Önerileri

Sensör eksikliği olduğu aşikar. Manyetik alan, sıcaklık veya ses gibi şeyler için üstünde fazladan sensörler olsa durumunu çözebileceği makinelerin sayısı artar. Mesela şu anki haliyle hareketli parçası olmayan bir fırının üstüne koyulsa onun bulunduğu durumları bu cihazın gönderdiği verilerle belirlemek imkansız olur diye düşünüyorum.

Kullandığım LCD ekran 16x2 boyutunda, daha büyük olsa karşılaşılan sorunu anlamak ve debug etmek daha kolay olabilirdi çünkü bazı hata yazıları LCD ekrana sığmıyor.