



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
NESNELERİN İNTERNETİ VE UYGULAMALARI DERSİ

AD – SOYAD : MERVE SARI
NUMARA : B181210074
E – POSTA : merve.sari3@ogr.sakarya.edu.tr

SAKARYA
ARALIK, 2020

İÇİNDEKİLER

- 1.1. Business Canvas İş Modeli
- 1.2. Büyük Veri
- 1.3. Proje Konusu
- 1.4. Başarı Ölçütleri
- 1.5. Sistem Mimarisi
- 1.6. Dizge (Sequence) Diyagram
- 1.7. Projede Kullanılanlar
 - 1.7.1. Nodemcu
 - 1.7.2. Load Cell (Yük Hücresi)
 - 1.7.3. HX711
 - 1.7.4. Arduino IDE
 - 1.7.5. ThingSpeak
 - 1.7.6. Blynk
- 1.8. Uygulamaya Ait Fotoğraflar
 - 1.8.1. Kabın Boş Olması Durumu
 - 1.8.2. Kabın Dolu Olması Durumu
- 1.9. Kaynaklar

1) Business Canvas İş Modeli

Temel Ortaklıklar	Temel Faaliyetler	Değer Önerisi	Müşteri İlişkileri	Müşteri Segmenti
Web/Veri Servis Sağlayıcılar Donanım Üreticileri Müşteriler Yazılım Geliştiriciler	Ürün Geliştirme Gerçekleştirme Yazılım Geliştirme Temel Kaynaklar Fiziksel Kaynaklar IoT Ağı Bulut Servisleri	Kap Dolu Olma Durumu Biliğlendirilmesi Kap Boş Olma Durumu Biliğlendirilmesi Kap Ağırlığının Ölçülmesi	Donanım Destaji Kanallar Twitter Blynk Internet	Kap Sahibi Sistem Kullanıcıları
Maliyet Yapısı		Gelir Akışı		
Yazılım Geliştirme Donanım Geliştirme		Donanım Ürün Satışı		

2) Büyük Veri

Tasarladığım IoT projesinde ürün sayısının çok fazla olduğunu ve buradan büyük veri elde edildiğini düşündüğümde öncelikle bu verileri depolamak için büyük veri tutabilecek bir veritabanı seçtim. Bunu da MongoDB olarak belirledim. Çünkü MongoDB boyut fark etmeksizin dosyaları kolayca depolayabilme özelliğine sahip. Bu özellik ilerde projenin geliştirildiği düşünülürse çok faydalı olacaktır. MongoDB'nin verinin bozulmamasını garanti etmesi, esnek bir mimari sunması, birden fazla sunucu üzerinde çalışabilmesi gibi birçok olumlu özelliğinin yanı sıra yaygın olarak kullanılması da yine ileride projenin gelişimi açısından önemlidir.

Projemde mesaj dağıtıcı olarak Apache Kafka kullanmayı düşündüm. Verilerin veya uygulama üzerinden alınan mesajların üretici ve tüketici arasında aktarılması işlemlerinde kullanabilirim. Kafka'yı seçmemde yüksek hızlı olması, verilerde bozulmamayı garanti etmesi gibi birçok olumlu özelliği etkili oldu. Ancak en önemlileri Kafka'nın diğer mesajlaşma platformlarından daha yüksek okuma ve yazma hızlarına sahip olması ve günümüzde başta Twitter(Projemde Twitter kullanıyorum.) olmak üzere birçok uygulama tarafından kullanılması. Bilindik birçok uygulama tarafından kullanılması ilerde farklı platformlarda uygulamam kullanılırsa uyumluluk açısından önemli olabilir.

Elde ettiğim verileri işlemek amacıyla Apache Spark kullanmayı düşünüyorum. Apache Spark'ı kullanma sebebim Hadoop'a göre yaklaşık 100 kat daha hızlı çalışması. Projemde kabın ağırlığını ölçüp anlık olarak veri gönderiyorum. Bu açıdan projemde hız çok önemli.

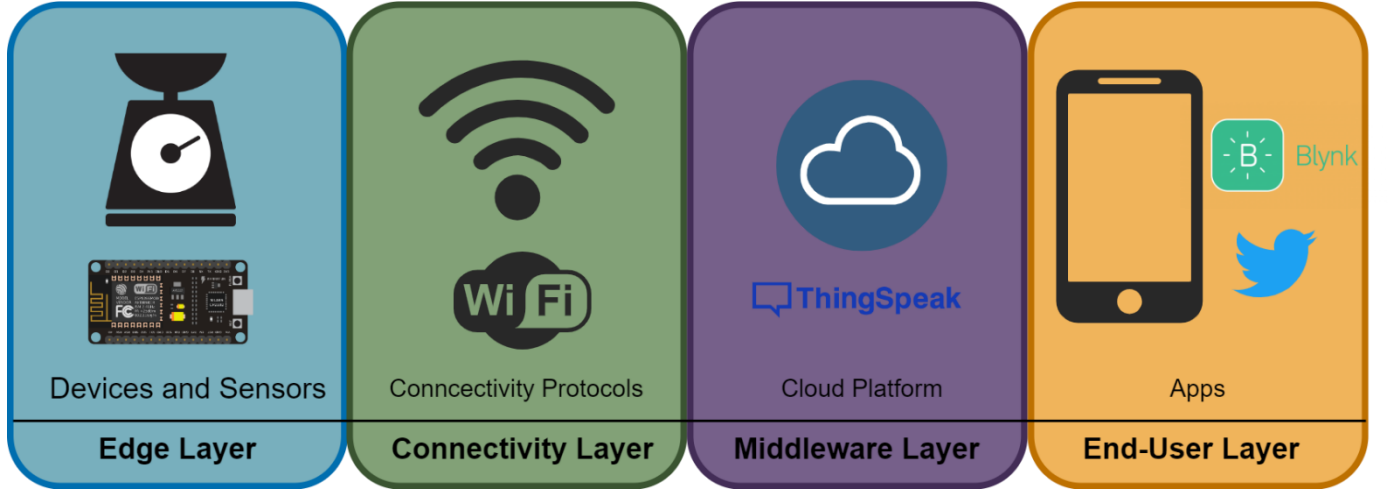
3) Proje Konusu

Tasarladığım bu proje ile sokak hayvanlarının beslenme ihtiyacına yardımcı olmak istedim. Projemde öncelikli olarak bir mama kabı tasarladım. Bu mama kabının ağırlığı sürekli olarak ölçülüyor ve ölçülen değer hem bulut platforma hem de mobil uygulamaya gönderiliyor ve böylece kabın sahibi sürekli olarak kabın durumunu görebiliyor. Kabın ölçtüğü değere göre boş olup olmasını kontrol ettim. Buna göre mobil uygulama üzerinden bildirim gönderdim. Ayrıca Twitter üzerinden de kap her boşalıp dolduğunda twit attım. Twitter kullanımı ile birlikte bu tür bir paylaşım gören diğer insanlara da teşvik olacağını ve böyle bir sosyal farkındalık ile daha fazla insanın hayvanlara yardım etmesini amaçladım.

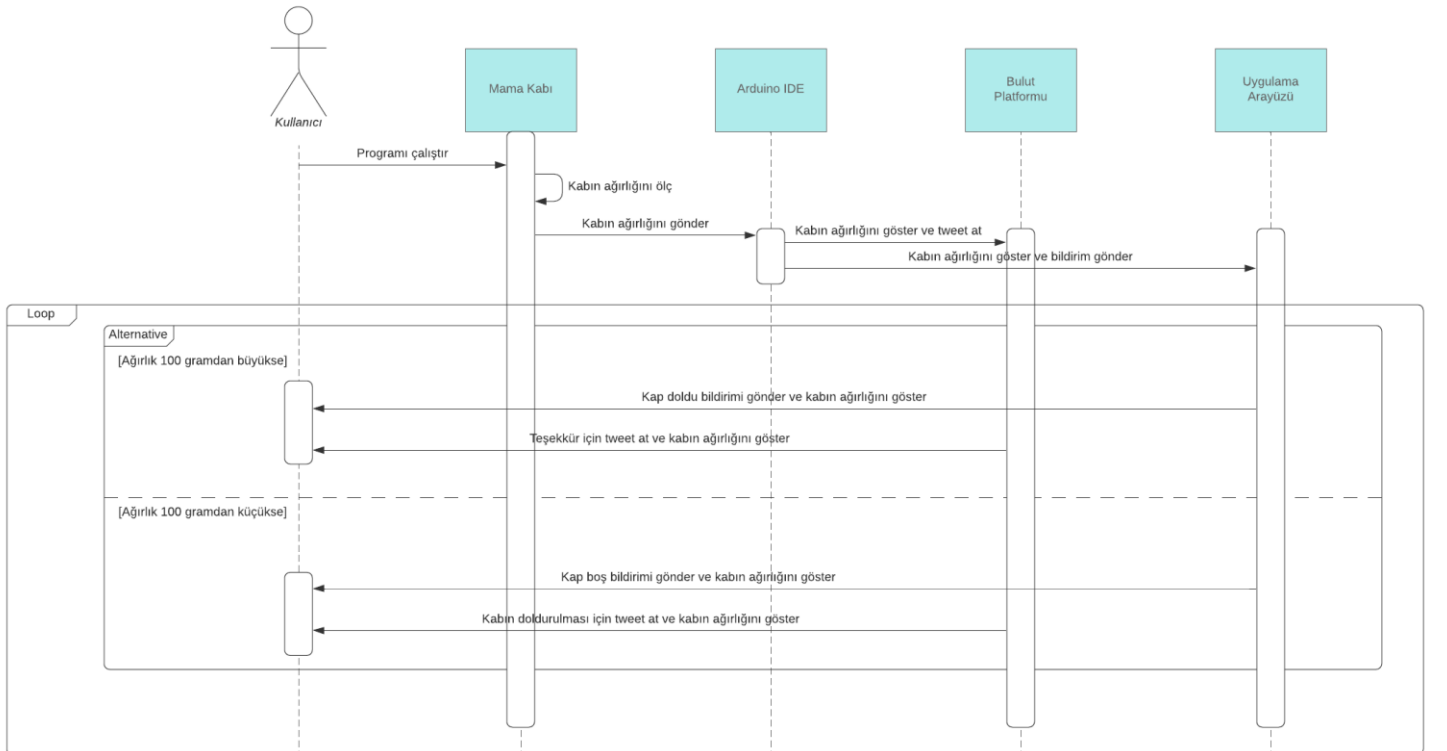
4) Başarı Ölçütleri

Başarılar	
Bir probleme çözüm getirme	%100
Elde edilen verilerden en iyi şekilde yararlanma	%90
Güvenlik	%50
Esneklik	%70
Geliştirilebilirlik	%100
Cihazlar arası haberleşme	%100
Hassasiyet ve doğruluk	%80
Programlanabilirlik	%100
Birlikte çalışabilirlik	%100

5) Sistem Mimarisi



6) Dizge (Sequence) Diyagram



7) Projede Kullanılanlar

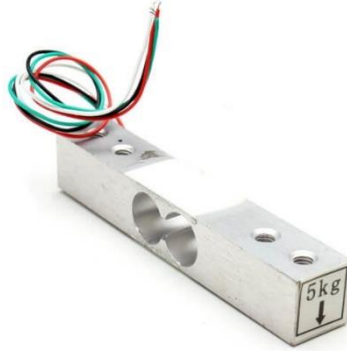
a) Nodemcu:

ESP8266 WiFi modülü üzerinde kurgulanmış, Lua script kullanan açık kaynaklı yerleşik yazılım ve geliştirme kitidir. Üzerinde çok sayıda bağlantı noktaları vardır. Bu bağlantı noktaları ile başka elektronik bileşenler yönetilebilir. WiFi sayesinde ise kolayca IoT nesneleri geliştirilebilir.



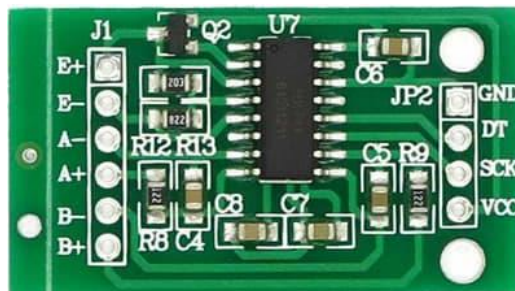
b) Load Cell (Yük Hücresi)

Load cell bir tür dönüştürücüdür. Üzerine uygulanan fiziksel kuvveti (gerilim, sıkıştırma, basınç veya tork gibi) elektrik sinyaline dönüştürür. Yük hücresine uygulanan kuvvet arttıkça, elektrik sinyali de orantılı olarak değişir. Yük hücreleri, ağırlığı ölçmek için kullanılır.



c) HX711

Ağırlığı ölçmek için yük hücrelerini kolayca okumamızı sağlayan küçük bir koparma kartıdır.



d) Arduino IDE:

Entegre geliştirme ortamıdır. Farklı işletim sistemlerinde kullanılabilir. Arduino uyumlu kartlara program yazmak ve yüklemek için kullanılır.

e) ThingSpeak:

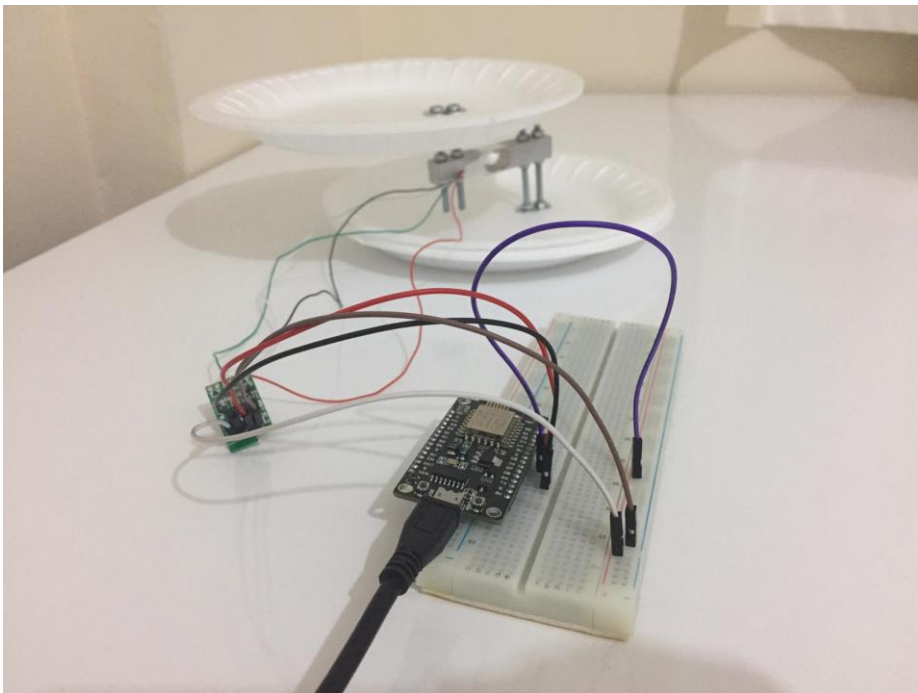
IoT nesnelerinden gerçek zamanlı veri toplama, verileri bulut ortamında depolama, analiz ve grafikler ile görselleştirme işlemlerinin gerçekleştirildiği ücretsiz kullanılabilen açık bir IoT bulut platformudur. MQTT yayın desteği vardır. Uyarı/alarm oluşturma gibi özelliklere sahiptir. Ayrıca Twitter ile birlikte kullanılabilir.

f) Blynk:

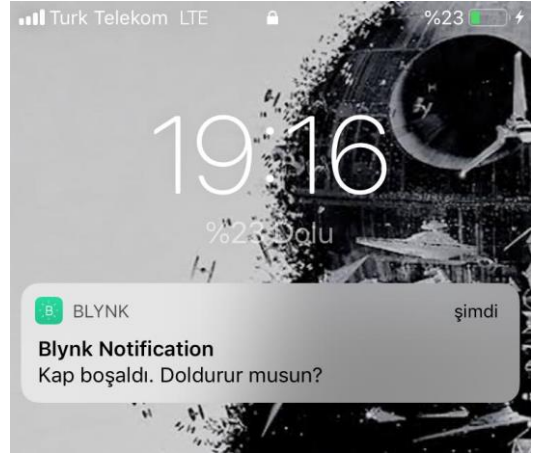
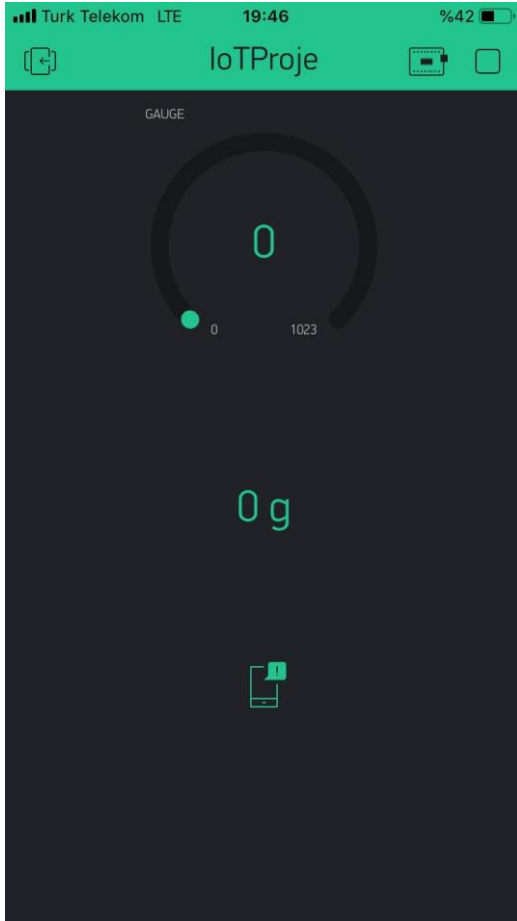
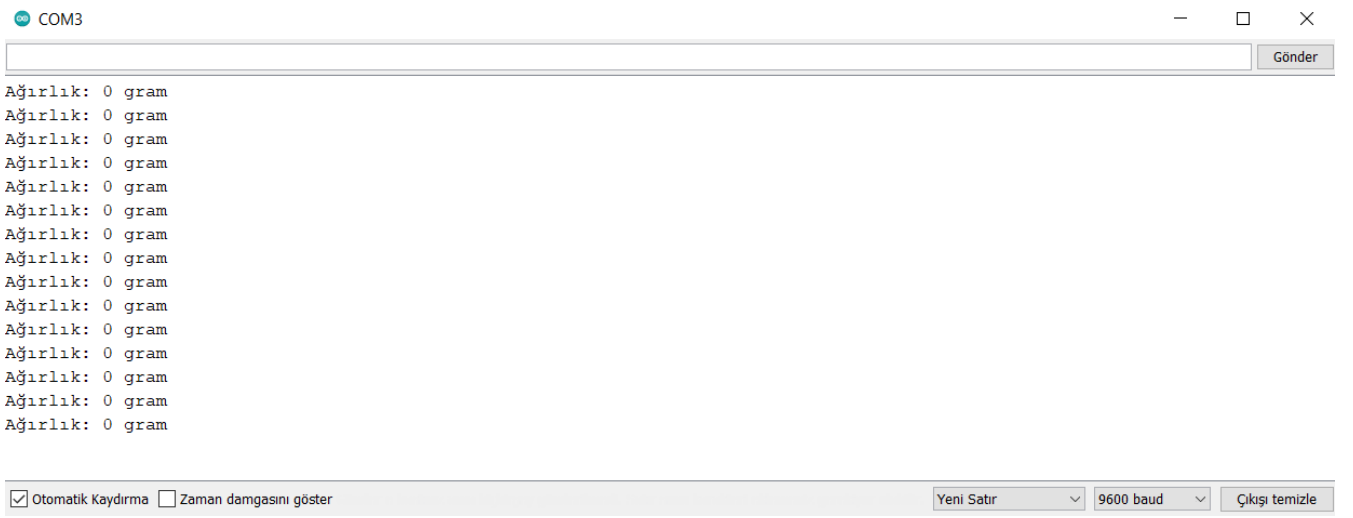
Ayrıntılı kod ve donanım bilgisi gerektirmeyen hem IOS hem de Android işletim sistemlerini destekleyen bir IoT cihaz kontrolünde ihtiyaç duyulabilecek çeşitli bileşenlere (slider, display, gauge) sahip bir mobil uygulamadır.

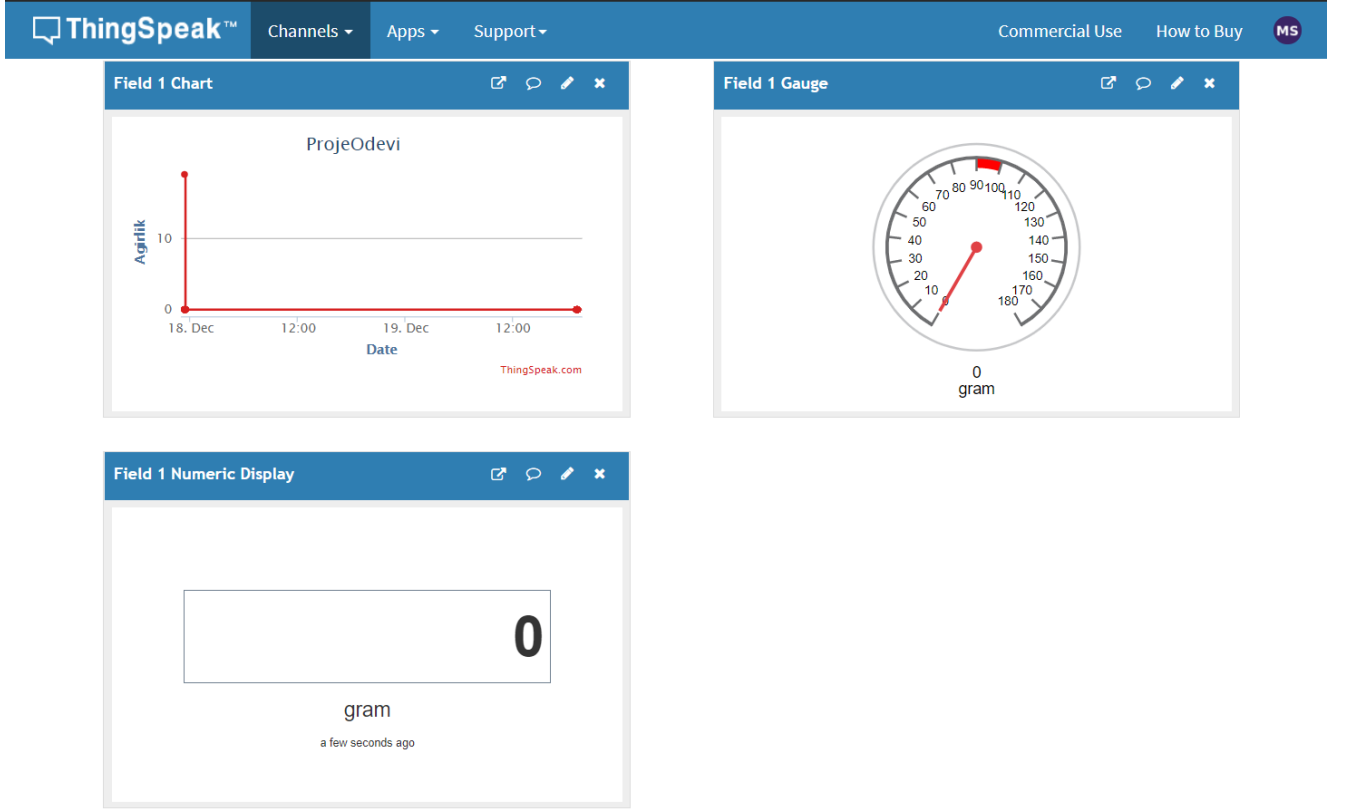
8) Uygulamaya Ait Fotoğraflar

Tasarladığım projede öncelikle ağırlık sensörü yardımıyla kabın ağırlığını sürekli olarak ölçtüm. Daha sonra bu ağırlığı Arduino IDE'ye gönderdim. Kabin boş olma durumunu kontrol etmek için ağırlığı 100 gram olarak belirledim. Yani ağırlığın 100 gramdan yüksek olduğu durumlarda kap dolu, düşük olduğu durumlarda kap boş sayılacaktır.



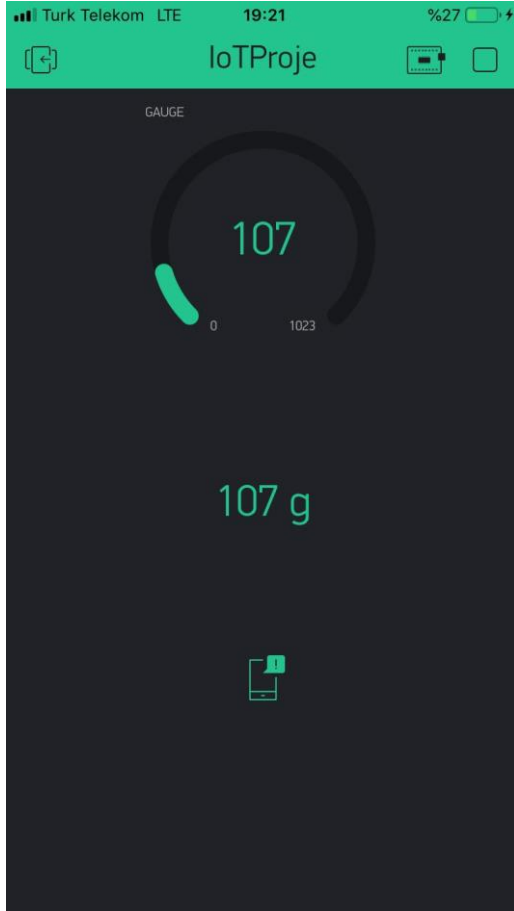
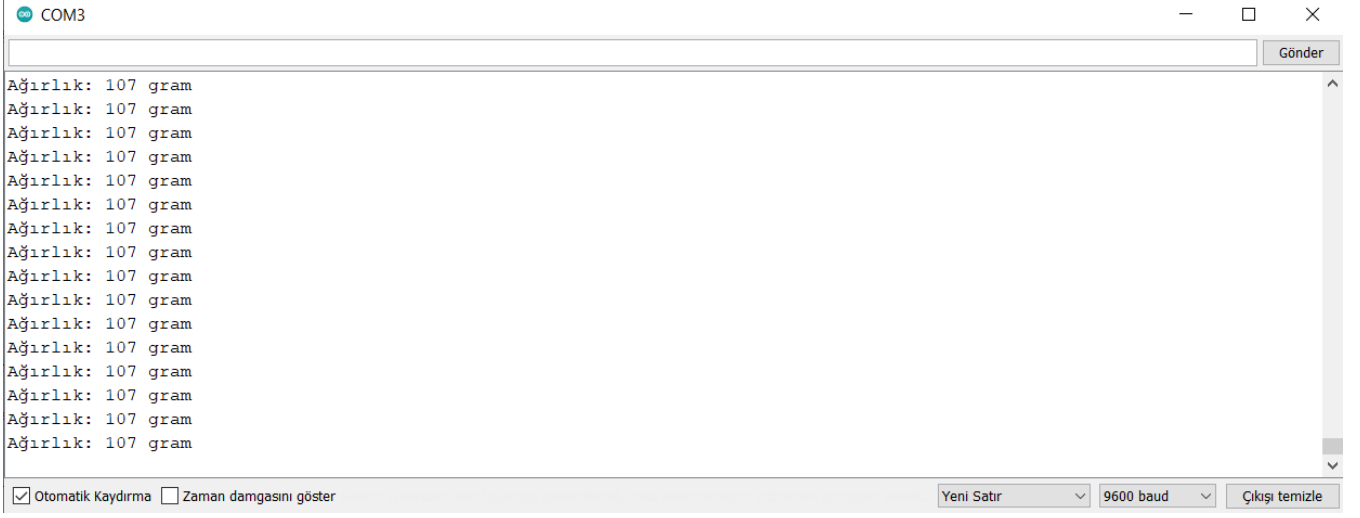
a) Kabin Boş Olması Durumu

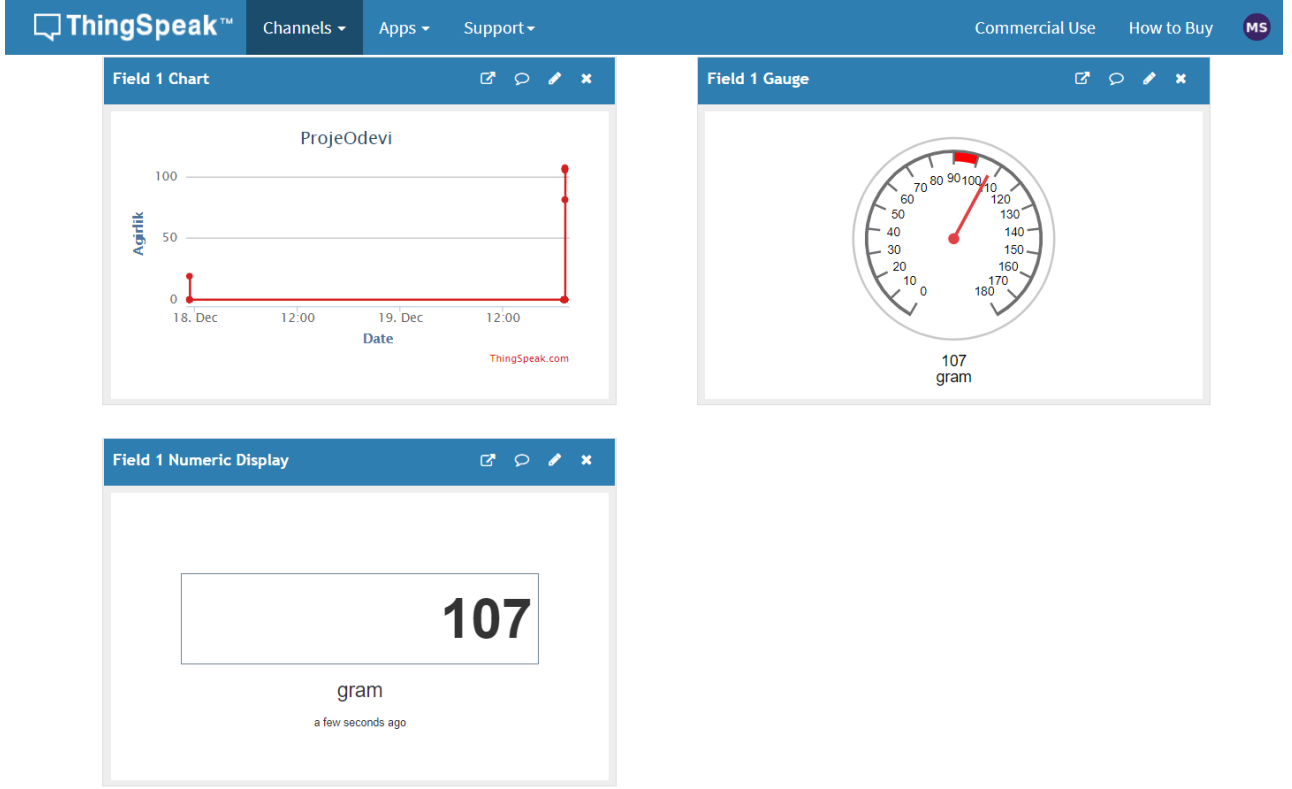




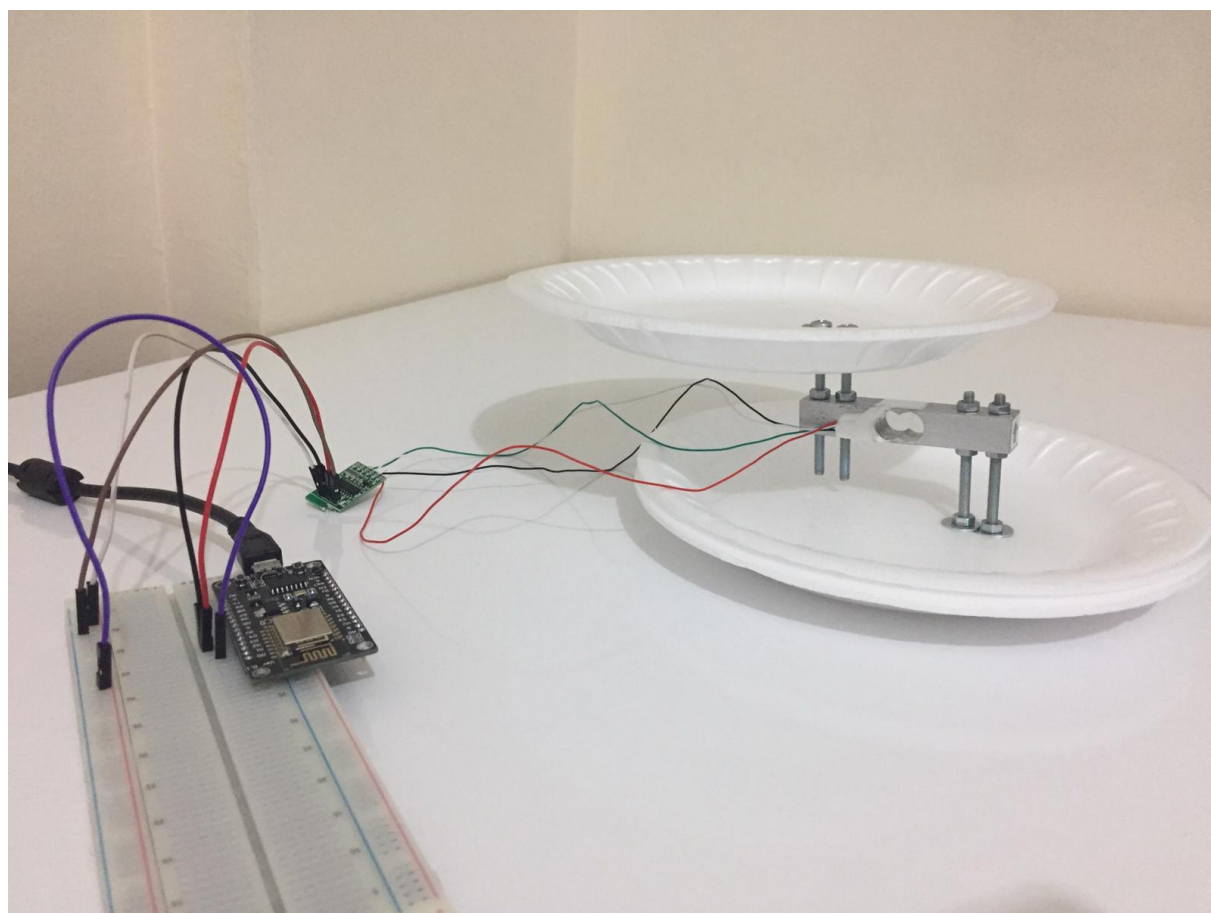
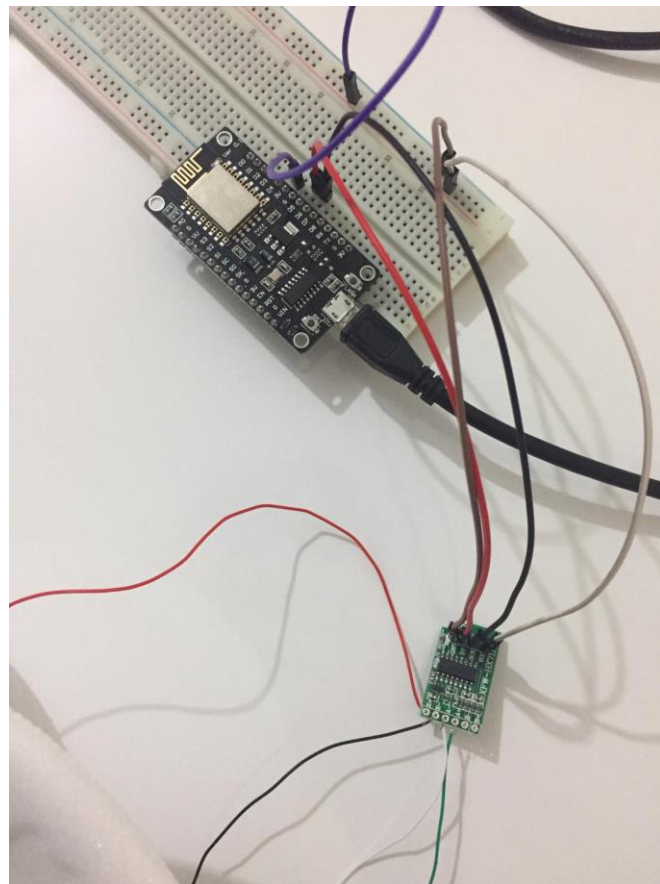
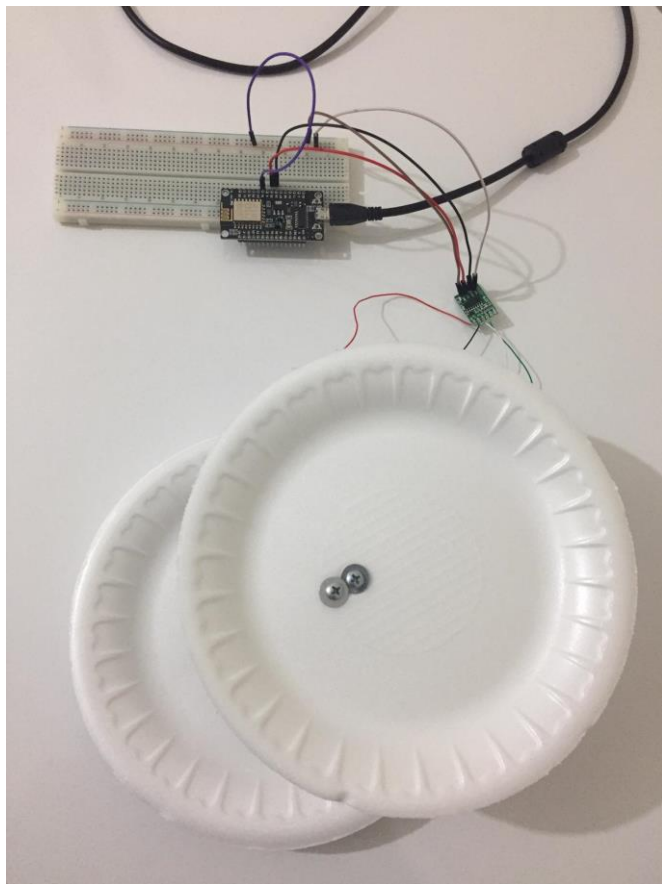
Ardunio IDE ortamında yazdığım kodlarla ağırlık değerini ThingSpeak ve Blynk platformlarına yolladım. ThingSpeak'te Field 1 tanımladım. Bu Field 1'in içine de gauge ve numeric display yerleştirdim. Böylece gelen ağırlık anlık olarak burada görüntülenebiliyor. Ayrıca ThingSpeak'in ThingTweet özelliğini kullanarak kabın boş olduğu ile ilgili de twit attım. Sonrasında Blynk uygulamasında gauge, labeled value ve notification ekledim. Gauge ve labeled value ile anlık olarak ağırlık görüntülenebilirken, notification ile de kabın boşaldığı durumlarda bildirim gönderdim. Yukarıda da kabın boş olduğu durumlar için ThingSpeak, Blynk ile gelen bildirim ve uygulama görünümü, atılan twit ile ilgili görseller yer almaktadır.

b) Kabin Dolu Olması Durumu





Ağırlığın 100 gram veya daha yüksek olduğu durumlarda, boş veya 100 gramdan düşük olduğu duruma benzer olarak öncelikle seri port ekranından ağırlık okunabiliyor. Thingspeak'te ve Blynk uygulamasında ilgili alanlara ağırlık değeri gönderiliyor. Ardından kabın dolu olması ile ilgili twit atılıyor. Günümüzde sosyal medya kullanımının çok yaygın olduğu düşünülürse Twitter kullanımı sayesinde de birçok insan bu twitleri görebilecektir. Birçok insana teşvik olması açısından bu şekilde bir uygulama hayvanların mama ihtiyacının karşılanmasında yararlı olacaktır.



KAYNAKLAR

- Doç. Dr. Cüneyt BAYILMIŞ ve Doç. Dr. Kerem KÜÇÜK, “Nesnelerin İnternet’i: Teori ve Uygulamaları”, Papatya Yayınevi, 2019.
- <https://docs.blynk.cc/>
- <https://strategyofthings.io/iot-project-success>
- <https://www.sam-solutions.com/blog/top-eight-internet-of-things-platforms/>
- <https://ciz.io/>
- <https://app.diagrams.net/>
- <https://iotdesignpro.com/projects/smart-weighing-machine-for-remote-weight-measurement>
- <https://how2electronics.com/iot-weighing-scale-hx711-load-cell-esp8266/>