



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

TRAFİK KAZALARI SONRASI TAKİP SİSTEMİ

Ömer Alp KIRAY - Can CİVEKOĞLU

Bilgisayar Mühendisliği Tasarım Çalışması I

Bilg. Müh. Tasarım Çalışması I Danışmanı: Öğr. Gör. Ali Osman SELVİ

BİLECİK
14 Haziran 2019



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

TRAFİK KAZALARI SONRASI TAKİP SİSTEMİ

Ömer Alp KIRAY - Can CİVEKOĞLU

Bilgisayar Mühendisliği Tasarım Çalışması I

Bilg. Müh. Tasarım Çalışması I Danışmanı: Öğr. Gör. Ali Osman SELVİ

BİLECİK

14 Haziran 2019

ÖZET

Projenin Amacı

Proje ekibinin yakınlarının yaşamış olduğu şanssız durumun başkalarının başına gelmesi halinde bu şanssız durumu şanslı duruma dönüştürmek.

Anahtar Kelimeler: Kaza - Sürücü - Takip

Projenin Kapsamı

Sorunumuz: Araç kazalarında; sürücünün, kaza anında baygınlık geçirip kazayı bildirememesidir. Nabız ve SpO2 sensörleri ile, sürücü anlık durumu; GPS Modülü ile, araç konumunu; GSM Modülü ile, "112 Acil"e çağrı göndererek sürücü durumunu ve araç konumu hakkında bilgiler iletecek. GPRS Modülü ile de verileri gerçek zamanlı veri tabanında güncel tutacak. "112 Acil" de bu bilgileri web sayfası ile gerçek zamanlı monitörleme yapabilecek.

Anahtar Kelimeler: Nabız ve SpO2 - GPS - GSM ve GPRS

Sonuçlar

Modülümüz ile kaza anında, sürücü baygınlık geçirirse "112 Acil" aracın konumunu ve kazazedenin durumunu "mobil olarak gerçek zamanlı" takip edebilecek.

Anahtar Kelimeler: 112 Acil - Konum ve Durum - Gerçek Zamanlı

ABSTRACT

Project Objective

To turn this unfortunate situation into a lucky situation if the unlucky situation experienced by the relatives of the project team happened to others.

Keywords: Accident - Driver - Tracing

Scope of Project

Our problem: In vehicle accidents; the driver is unable to report the accident by fainting at the time of the accident. With pulse and SpO2 sensors, the driver instant status; With GPS Module, the vehicle position; With GSM Module, it will send information about the driver status and vehicle location by sending a call to "112 Emergency". The GPRS module will also keep the data up to date in the real-time database. "112 Emergency" will be able to monitor this information in real time via web page.

Keywords: Pulse and SpO2 - GPS - GSM and GPRS

Results

With our module in case of an accident, if the driver faints, the "112 Emergency" will be able to follow the position of the vehicle and the status of the victim as "mobile real-time".

Keywords: 112 Emergency - Location and Status - Real Time

TEŞEKKÜR

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve bizi bu konuya yönlendiren saygıdeğer hocamız ve danışmanımız Sayın Öğr. Gör. Ali Osman SELVİ'ye tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederiz.

Ömer Alp KIRAY - Can CİVEKOĞLU

14 Haziran 2019

İÇİNDEKİLER

ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİL LİSTESİ	vi
1 GİRİŞ	1
2 PROJENİN TASARLANMASI	3
3 PROJENİN GERÇEKLEŞTİRİLMESİ	11
4 SONUÇLAR VE ÖNERİLER	14
KAYNAKLAR	15

ŞEKİL LİSTESİ

1	Projenin akış şeması	3
2	Projenin aşamaları	4
3	Projenin birinci aşaması	5
4	Projenin ikinci aşaması	5
5	Projenin üçüncü aşaması	6
6	Projenin dördüncü aşaması	6
7	RPi - Arduino - MAX30100 bağlantı şeması	7
8	MAX30100 devre şeması	8
9	RPi - TTL - SIM800C Shield bağlantı şeması	9
10	Site görünümü	10
11	MAX30100 - Arduino Serial Monitor çıktısı	11
12	MAX30100 RPi Terminal çıktısı	12
13	SIM800C Shield'in çalıştırılması	13
14	Proje son hali	14

1 GİRİŞ

Ülkemiz karayolu ağında 2017 yılında toplam 1 milyon 202 bin 716 adet trafik kazası meydana geldi. Bu kazaların 1 milyon 20 bin 47 adedi maddi hasarlı, 182 bin 669 adedi ise ölümlü yaralanmalı trafik kazasıdır. Yıl içerisinde meydana gelen ölümlü yaralanmalı trafik kazalarının yüzde 74,4'ü yerleşim yeri içinde yüzde 25,6'sı ise yerleşim yeri dışında meydana geldi.

Türkiye'de 2017 yılında meydana gelen 182 bin 669 adet ölümlü yaralanmalı trafik kazası sonucunda 3 bin 534 kişi kaza yerinde, 3 bin 893 kişi ise yaralanıp sağlık kuruluşlarına sevk edildikten sonra kazanın sebep ve tesiriyle 30 gün içinde hayatını kaybetti. Ölümlerin yüzde 48,6'sı, yaralanmaların yüzde 66,8'i yerleşim yeri içinde gerçekleşirken ölümlerin yüzde 51,4'ü yaralanmaların ise yüzde 33,2'si yerleşim yeri dışında oldu. [1]

Ülkemiz karayolu ağında 2017 yılında gerçekleşen trafik kazalarında ölen kişilerin yüzde 43,2'si sürücü, yüzde 34,2'si yolcu, yüzde 22,6'sı ise yayadır. Trafik kazalarında ölenler ve yaralananlar cinsiyetlerine göre incelendiğinde ise ölenlerin yüzde 76'sının erkek, yüzde 24'ünün kadın, yaralananların ise yüzde 68,9'unun erkek, yüzde 31,1'inin kadın olduğu görüldü.

Türkiye'de 2017 yılında ölümlü yaralanmalı trafik kazasına neden olan toplam 213 bin 325 kusura bakıldığında kusurların yüzde 89,9'unun sürücü, yüzde 8,5'inin yaya, yüzde 0,7'sinin yol, yüzde 0,5'inin taşıt ve yüzde 0,4'ünün yolcu kaynaklı olduğu görüldü. Bu projede trafik kazalarında sürücü bilgileri ile hayati değerlerinin 112'e hızlı bir şekilde gönderilmesi hayatları kurtarmada önemli bir etken olabilir.¹.

¹ Bu amaçla geliştirilen projede sürücü temelli çalışma yapılmıştır

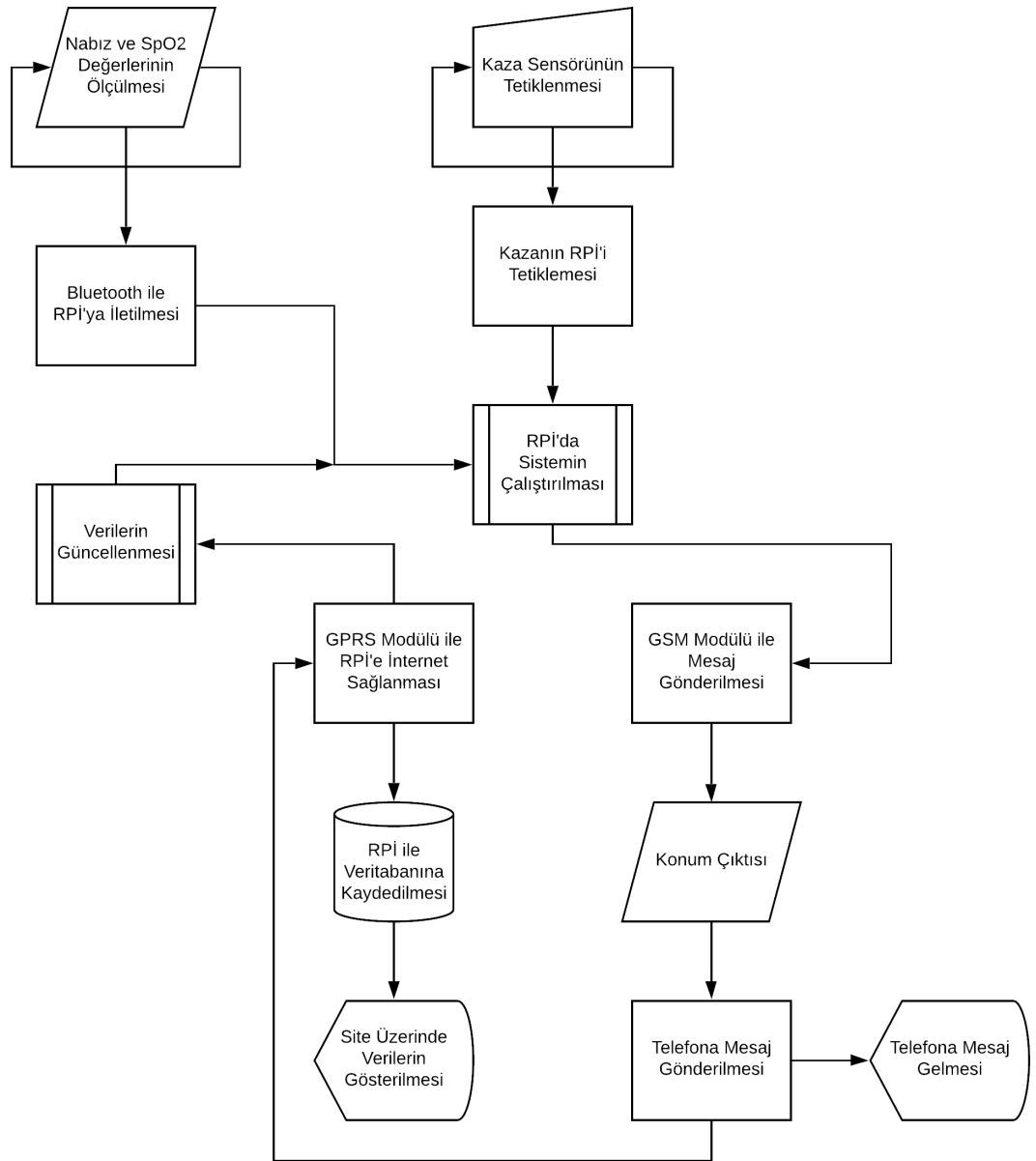
Bilgisayar Mühendisliği Tasarım Çalışması 1 için geliştirdiğimiz bu projenin tasarım aşamasında birçok araştırma ve makale inceledik. Bunlardan biri projemize yakınlığıyla dikkat çekiyordu. Smart Real-Time Healthcare Monitoring and Tracking System using GSM/GPS Technologies [2] adlı makalede sensörlerden alınan değerler ile GPS modülü aracılığıyla konumu, GPRS modülü aracılığı ile sisteme yollanmıştır. Ayrıca SIM modülü aracılığıyla bu değerleri SMS olarak da göndermiştir. Bu makalede hastalar üzerinde yapılmış olan sistem; sürücüler üzerindeki yapacağımız sisteme benzerdir.

Projemizin amacı: Proje ekibinin yakınlarının yaşamış olduğu şanssız durumun başkalarının başına gelmesi halinde bu şanssız durumu şanslı duruma dönüştürmektir. Bu projenin kapsamı: Sürücü, kaza anında baygınlık geçirirse kazayı bildirmesine yardımcı olmaktır. Nabız ve SpO2 sensörleri ile, sürücü anlık durumu; GPS Modülü ile, araç konumu; GSM Modülü ile, "112 Acil"e çağrı göndererek sürücü durumunu ve araç konumu hakkında bilgiler iletecek. GPRS Modülü ile de verileri gerçek zamanlı veri tabanında güncel tutacak. "112 Acil" de bu bilgileri web sayfası ile gerçek zamanlı monitörleme yapabilecek.

Bu projede Raspberry Pi 2 Model B kullanılmıştır. RPi kullanmamızın sebebi güçlü RAM ve işlemciye sahip olmasıdır. Bu projede maalesef Nabız ve SpO2 sensörünü RPi'ye bağlayarak kullanamadık. Çünkü RPi'de harici kesme için GPIO pini tanımlanmamıştır. Ve yazılımsal olarak çalışan kesme için sensörümüz yazılımsal kesmeyi görememektedir. Datasheet'e baktığımızda harici kesme pinleri için Arduino'yu kullanabileceğimizi gördük. Bu yüzden Nabız ve SpO2 sensörü için Arduino aracılığı ile Raspberry Pi kullanılmıştır.

2 PROJENİN TASARLANMASI

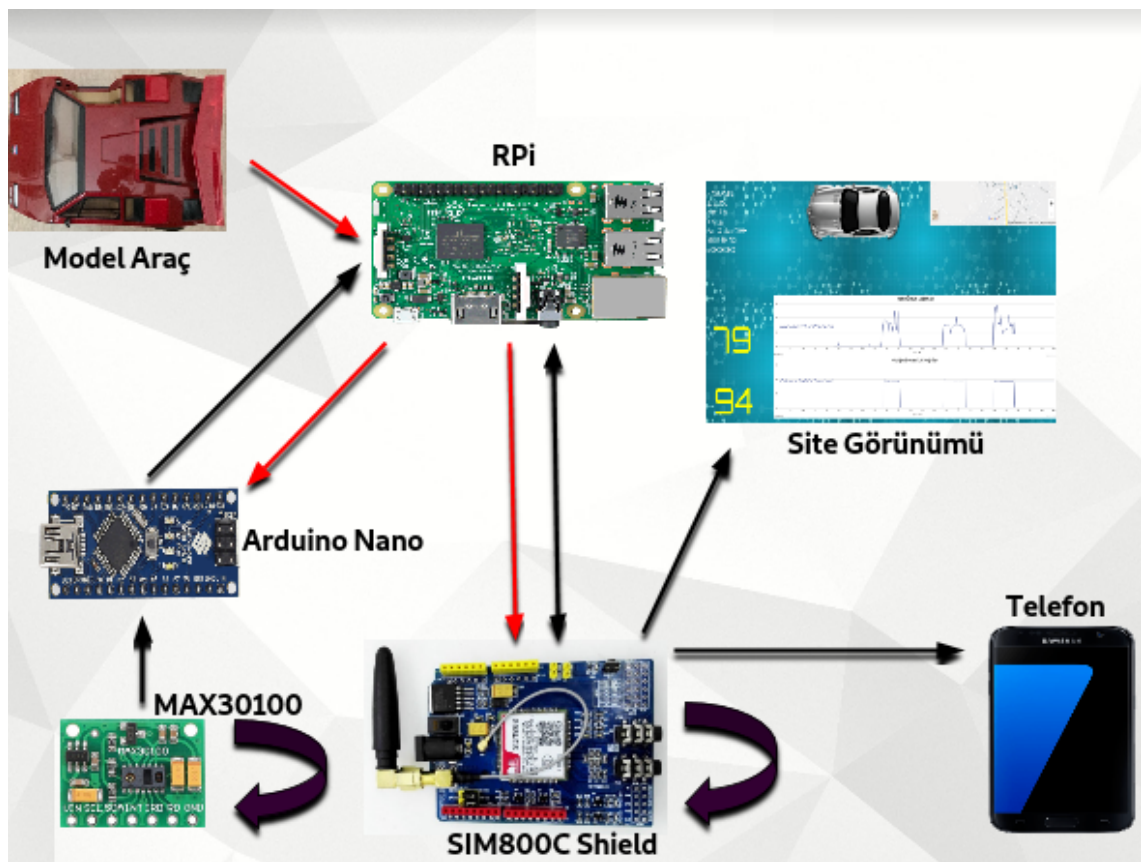
Projenin tasarlama aşamasında yapılan işlemler sırasıyla kaza tespit sisteminin belirlenmesi, Raspberry Pi'nin tetiklenmesi, MAX30100 sensörünün çalıştırılması, sensörlerin Raspberry Pi ile iletişime geçmesi ile verilerin telefona ve siteye iletilmesi şeklindedir. Proje tasarım aşamaları akış şeması üzerinden Şekil 1'de gösterilmiştir.



Şekil 1: Projenin akış şeması

Altındaki diyagramda projenin çalıştırılması esnasında gerçekleşecek olaylar gösterilmiştir.

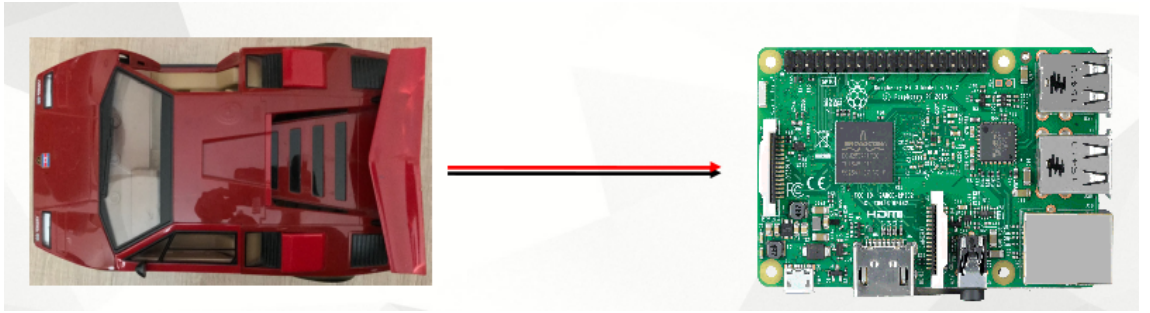
Şekil 2’de kırmızı oklar ile belirtilen işlemler güç işlemleri ile ilgilidir. Siyah renkli oklarda ise veri alış-veriş kısımlarını belirtir. Mor ile belirtilen adımlar yineleme adımlarıdır ve çalışmaya devam edildikçe kendisine bağlı işlemleri yineler.



Şekil 2: Projenin aşamaları

Kaza tespit edildiğinde Raspberry Pi tetiklenir ve böylece Raspberry Pi'ye bağlı SIM800C'ye enerji sağlanmış olur ve eldiven de tetiklenir. Böylece tasarlanan sistem çalışmaya başlar.

SIM800C'ye güç sağlayan Raspberry Pi; MAX30100 sensörü üzerindeki verileri Arduino aracılığıyla alır ve bu verileri SIM800C aracılığı ile telefona ve siteye aktarılmasını sağlar.

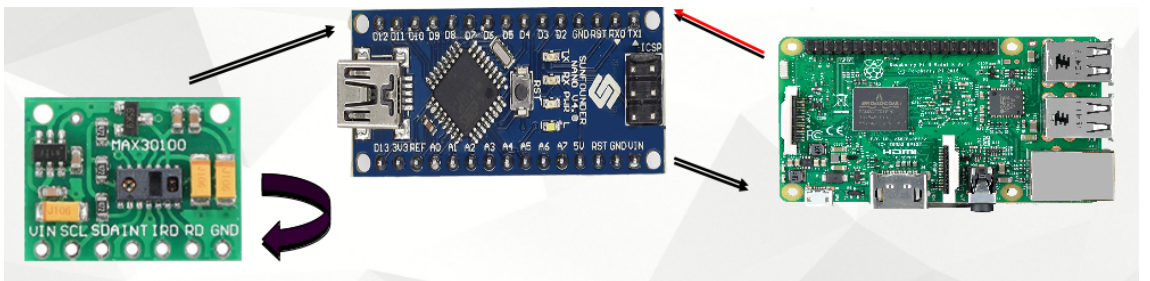


Şekil 3: Projenin birinci aşaması

Arduino Nano üzerinde çalışan sensör nabız ve SpO2 değerlerini Bluetooth seri hattı üzerinden iletir.

MAX30100 Sensörünü Raspberry Pi'ye direkt bağlayamadığımızdan dolayı Arduino aracılığı ile Raspberry Pi ile iletişimi geçiriyoruz.

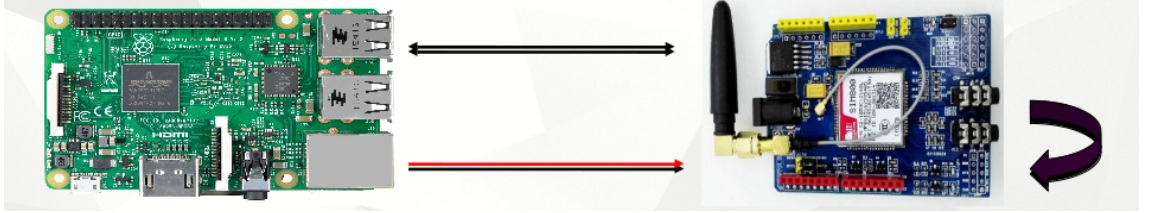
MAX30100 ile SIM800C arasında köprü sağlanmış olur.



Şekil 4: Projenin ikinci aşaması

Raspberry Pi aldığı nabız ve SpO2 değerlerini, gerekli bağlantıları gerçekleştirerek SIM800C'ye yollar. Bağlantı sağlandığı sürece işlemler tekrar edilir.

Kendisine gelen tetikleme üzerine GSM modülü ile konum bilgilerini de dahil ederek acil servise ve özel numaraya mesaj gönderir.



Şekil 5: Projenin üçüncü aşaması

SIM800C ile RPi'dan alınan veriler GPRS modülünün interneti ile veritabanına kaydedilir. Ve anlık olarak site üzerinden sürücü bilgileri, aracın durumu, aracın konumu, nabız ve SpO2 değerleri gösterilir.

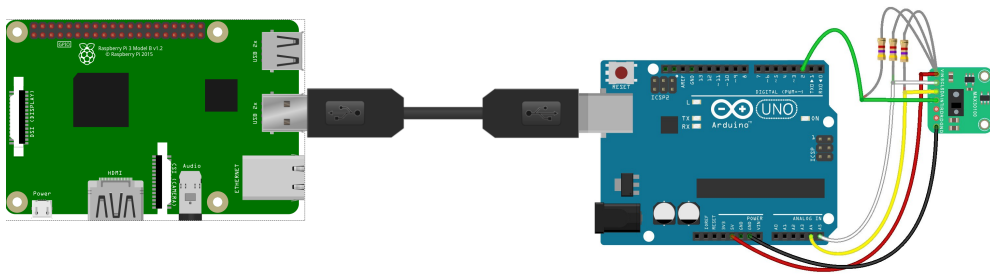


Şekil 6: Projenin dördüncü aşaması

Geliştirdiğimiz bu proje, sürücünün nabız ve SpO2 değerlerini MAX30100 sensörü ile ölçecek. SIM modülünün baz istasyonu üzerinden konum alabilmesi sayesinde GPS modülü kullanmaya ihtiyaç duymadık. SIM modülü (SIM800C) ile hem GPS konumu bulunuyor hem de GSM ve GPRS özellikleri kullanılabilir. SIM Modülü ile SMS olarak sürücü ismi, yaşı, kan grubu ve latitude (enlem), longitude (boylam) koordinatları gönderilecektir. Ayrıca bu veriler GPRS Modülü (SIM800C) ile sistemimize de gönderilecektir. Daha önceden bahsedildiği gibi maalesef MAX30100 ile RPi bağlantısını Arduino olmadan gerçekleştiremiyoruz. Bu yüzden Şekil 1'deki bağlantı yapılıyor.

Aşağıdaki Şekil 1'de MAX30100 Sensörünün RPi ile bağlantısı gösterilmiştir. Bağlantı şemasına baktığımızda MAX30100 Sensörünü RPi'a direkt bağlanıp kullanılmadığı görülmektedir. Bunun sebebi RPi'da harici kesme için GPIO pini tanımlanmamış ve yazılımsal olarak yapılmaya çalışılan kesmede MAX30100 Sensörünün dahili kesmeye ihtiyaç duymasından dolayı sistem RPi üzerinden çalıştırılamamaktadır. Arduino da ise harici kesme için GPIO pin tanımlanmıştır. Bu yüzden sistem Arduino - Pi bağlantısı aracılığıyla çalıştırılmıştır. Ve böylece Şekil 1'de görüldüğü halini almıştır.

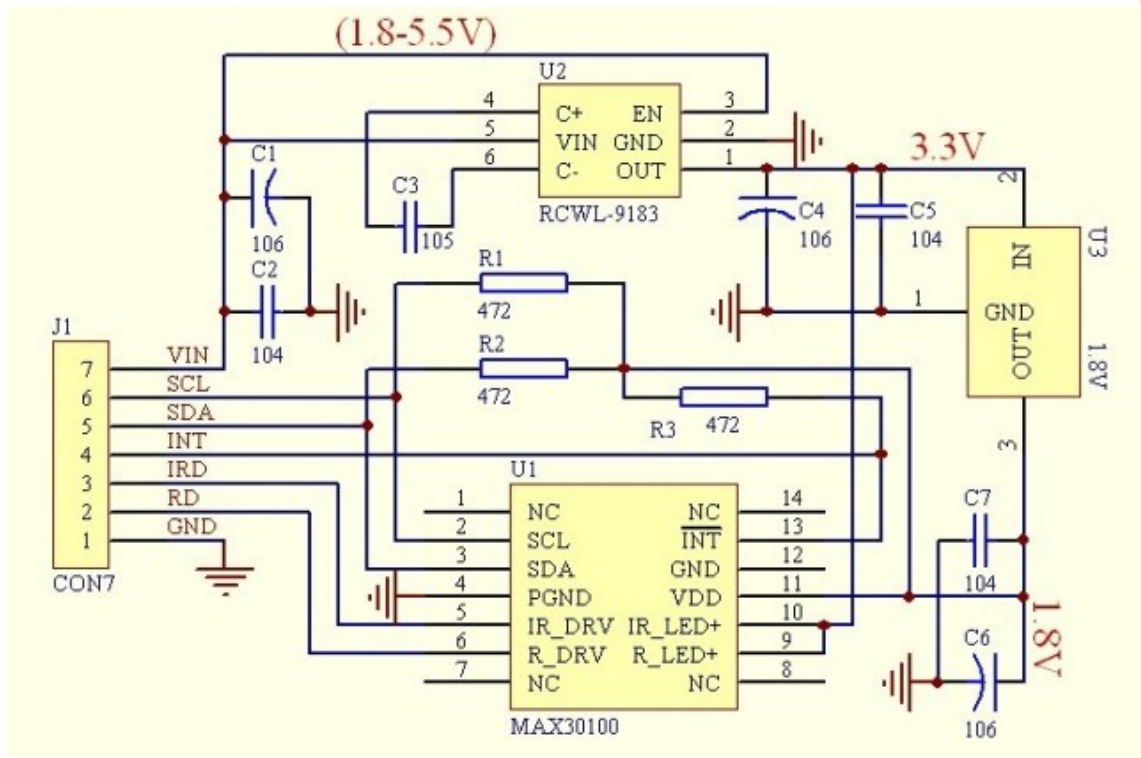
Şekil 1'de ki bir diğer nokta ise SCL ve SDA hatları üzerine harici pull-up direnç bağlantısı ile Arduino'ya bağlantı yapılmasıdır. Direnç bağlantılarının yapılma sebebi devre şeması üzerinden Şekil 2'de açıklanmıştır.



Şekil 7: RPi - Arduino - MAX30100 bağlantı şeması

Şekil 2'ye baktığımızda MAX30100 Sensörünün devre şeması gösterilmiştir. Şekil 2'de ki bağlantılarda SCL ve SDA pinleri için 4.7K Ohm harici pull-up direnç bağlantısı yapılmıştır. Bunun sebebi MAX30100'ın I2C'den 5V gerilim isteğinde bulunmasına rağmen Arduino I2C hattına 3.3V gerilim değeri sağlar. Bu yüzden MAX30100 I2C VDD için 1.8V güç değerine daha ihtiyaç duyar. Bunun için çözümümüz SCL ve SDA pinleri için harici 4.7K Ohm pull-up direnç bağlantısı yapmaktır.

Bu bağlantılar yapıldığında I2C pinlerine gelen 3.3V değeri harici 4.7K Ohm pull-up direnç bağlantısı yapılarak 5.1V seviyelerine çıkarılır. Böylece MAX30100 sensörümüz yüksek gerilim seviyesine çıkartarak sensörümüzü çalıştırabildik. Arduino'da sensörümüzden okuma yapabilmek için 9600 Baud Rate ayarında Serial Monitor ile iletişime geçirmek gerekiyor.

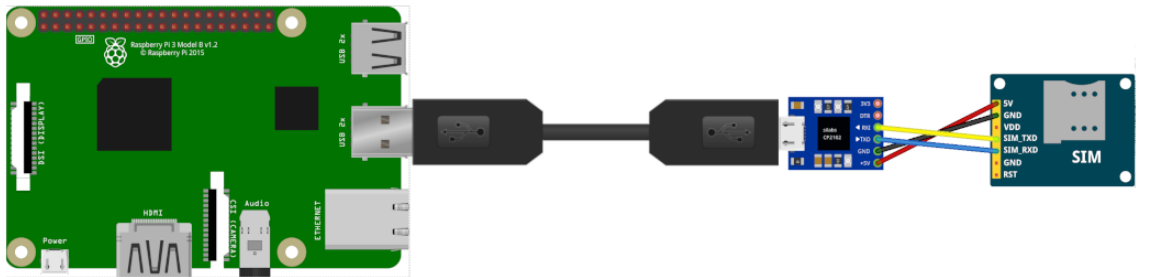


Şekil 8: MAX30100 devre şeması

Projemizin bir diğ er tarafı olan GSM, GPRS ve GPS Mod ulleri tek bir shield  zerinde toplanmıřtır. Bu shield SIM800C'dir. SIM800C Shield i in RPi'a TTL d n řt r c   zerinden baėlantı saėlanmaktadır. řekil 3'te g r ld ė  gibi SIM800C Shield'ın TX'i TTL RX'e, SIM800C Shield'ın RX'i TTL TX'e, SIM800C Shield'ın 5V'u TTL 5V'a ve SIM800C Shield GND'ı TTL GND'a baėlayarak USB  zerinden RPi ile seri haberleřme yapılması saėlanmıřtır.

řekil 3'de g r ld ė  gibi SIM800C Shield'a enerji RPi  zerinden saėlanmaktadır. Harici g   baėlantı denemelerinde gerekli akım deėerinin saėlanamaması, sens r n hassasiyeti ve sens r n kristal hızının farklı olmasından dolayı harici g   kullanılamamıřtır. SIM800C ile seri iletiřime ge ebilmeleri i in g c  TTL d n řt r c   zerinden RPi'ın USB'si  zerinden alır. RPi ile USB  zerinden iletiřime ge ebilmeleri i in bařlangı ta seri baėlantı noktasını kontrol eden varsayılan konsolu (/dev/ttyAMA0) devre dıřı bırakmamız gerekir.

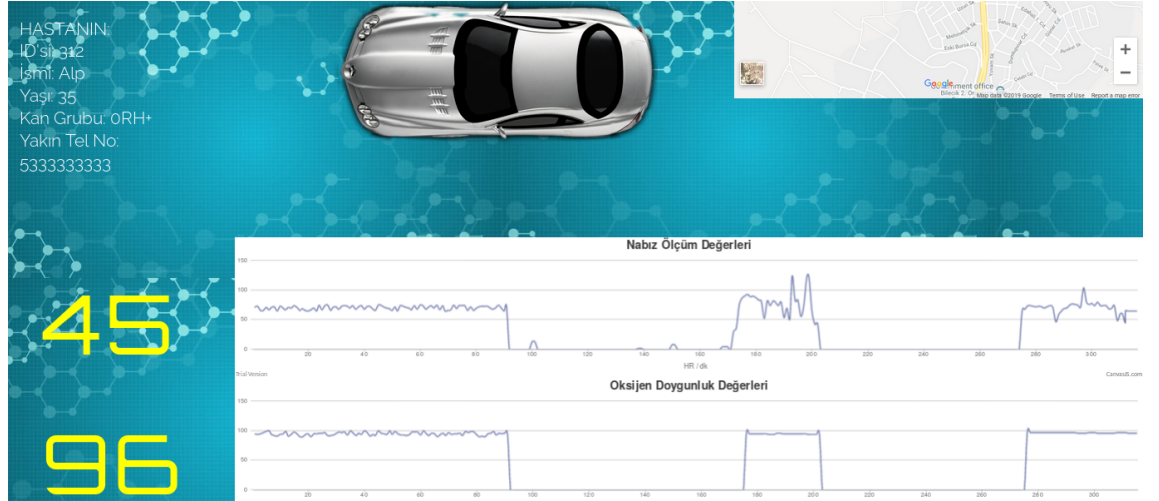
SIM800C Mod l  AT komutları aracılıėıyla baz istasyonuna istekte bulunduėu sırada 2A'e yakın akım  eker. Bazı durumlarda ( rneėin RPi'ın diėer USB portundan akım  ekme ihtiya ı duyulduėunda) RPi bu akımı saėlayamayabilir. Bu y zden 2A+ harici g   kaynaėı saėlanması gerekebilir fakat denemelerimizde sens r hassasiyetini g    n nde bulundurarak g c  RPi'ın USB portundan almaya devam edilmiřtir.



řekil 9: RPi - TTL - SIM800C Shield baėlantı řeması

Projemizin yazılım tarafı; sensörleri programlama, kodlar ve site kısmından oluşmaktadır. Sensörlerin çalışması için gerekli olan programlar kullanılmış ve kodlar yazılmıştır. Site Şekil 4'te görüldüğü gibidir ². Sol üst köşede hasta ile ilgili bilgiler yer almaktadır. Bununla birlikte aracın durumunu gösteren resim ile konumunu göstermesi için Google Maps kullanılmıştır. Alt taraflarında Nabız - SpO2 değerleri ve çizelge halinde gösterimleri vardır. Nabız ve SpO2 değerleri veritabanından alınmaktadır. RPi gerekli bağlantıları sağladığında veritabanına veri gönderecek ve PHP üzerinden site kendi kendine değerleri güncelleyecektir.

Şekil 4'de görüldüğü gibi aracın kaza yaptığı noktanın tespitinde piyasada bulunan sensörlerin bir çoğunun dijital olması ve analog olan sensörlerin hassasiyetlerinin araba kazasının noktasını belirlemede yetersiz kalması sebebiyle sistem şu anlık site üzerinden rastgele olacak şekilde başlatılmaktadır. Daha sonra sürekli çalışan MAX30100 Nabız ve SpO2 değerlerini gönderebilmesi için SIM800C Modülünün internete erişmesini bekler. Bağlantı sağlandığında SIM800C Modülü üzerinden gerekli bilgiler veritabanına gönderilir ve anlık olarak site üzerinde gösterilir.



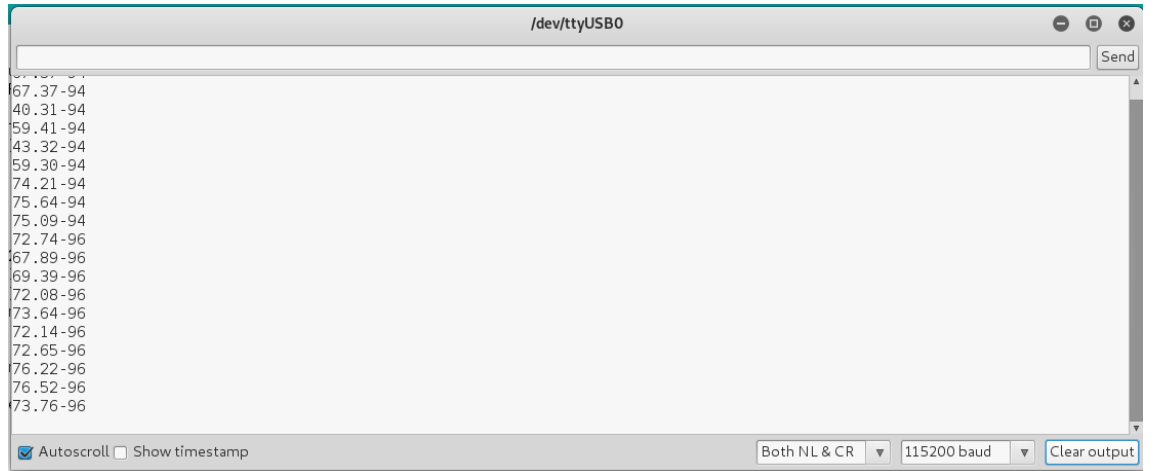
Şekil 10: Site görünümü

² Site ile ilgili detaylı bilgiler rapora eklenmemiştir.

3 PROJENİN GERÇEKLEŞTİRİLMESİ

Şekil 5’de görüldüğü gibi öncelikle MAX30100 Sensörümüzü kodlamak için Arduino IDE üzerinde bulunan MAX30100lib by OXullo Intersecans [3] kütüphanesini kullanıyoruz. Bu kütüphane yardımıyla kodlama yapıp Arduino’ya yüklüyoruz. Daha sonra Serial Monitor’den değer okumaya başlayabiliriz. Şekil 5’te ki görülen kısımda ilk değer nabız değeri, ikinci değer SpO2 değerini göstermektedir. Sensörün pozisyonuna göre değerlerde sapmalar olabilir.

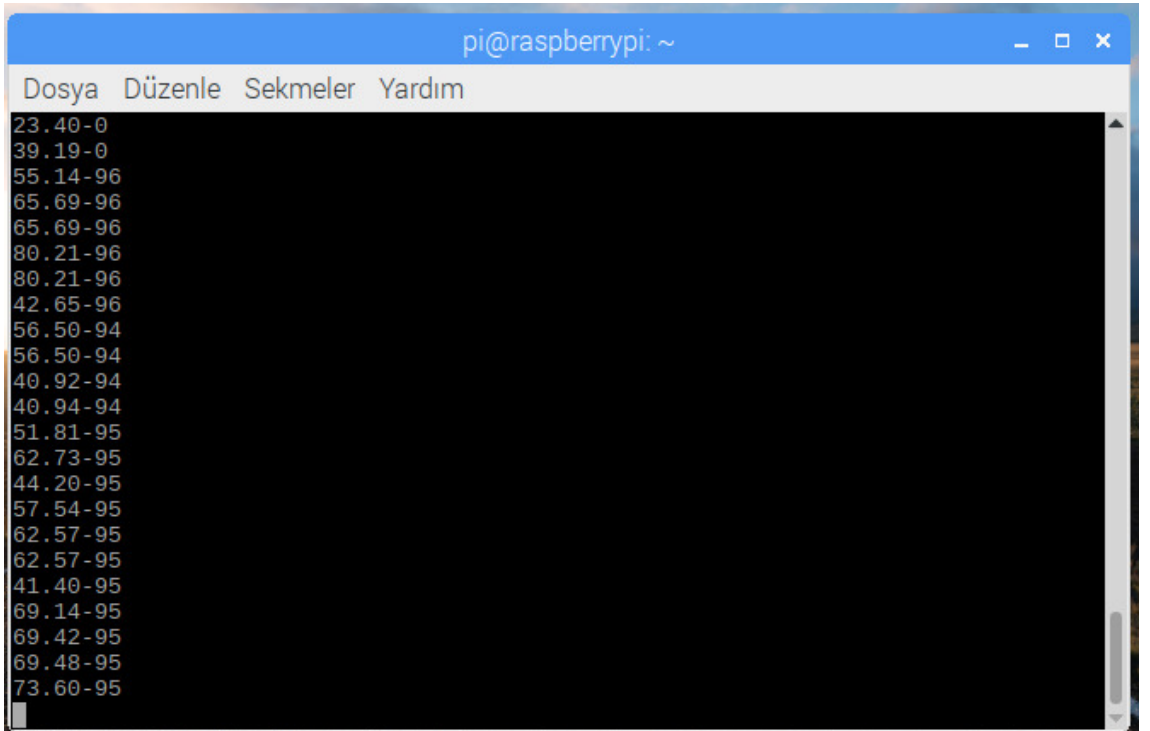
Şekil 6’da sensörümüzü RPi’a bağlanmış halini görmektesiniz. Arduino IDE üzerinden upload edilmiş sistem enerji verildiğinde çalışmaya başlar. Ve RPi’ın Terminal’i Arduino’nun Serial Monitor’unu okumaya başlar. RPi Terminale gelen değerlerin arasında "-" işareti vardır. Bu işaret değerleri ayrı ayrı işlemek için gereken karakter özelliklerinden biridir. Bu formata göre çıktı alıntısı yapıldığında Arduino Serial Monitor ile RPi Terminal’deki değerler birbiri ile eşleşmesi gerekir.



Şekil 11: MAX30100 - Arduino Serial Monitor çıktısı

Şekil 6’da görüldüğü gibi MAX30100 Sensörümüzü RPi’ın Terminal’i ile iletişime geçiriyoruz. Böylece Arduino’da olduğu gibi değerleri RPi’ın Terminali’ne getirebiliyoruz. Serial Console devre dışı bırakıldığından okumaya başlatıldığı anda Arduino Serial Monitor ile RPi Terminal eşleşir. Böylece değerler Terminal üzerinde görülür. RPi üzerinde Terminal iletişimi görebilmek için Minicom [4] programı üzerinden Arduino’da Serial Monitor’u atanan Baud Rate ile çalıştırılması gerekir.

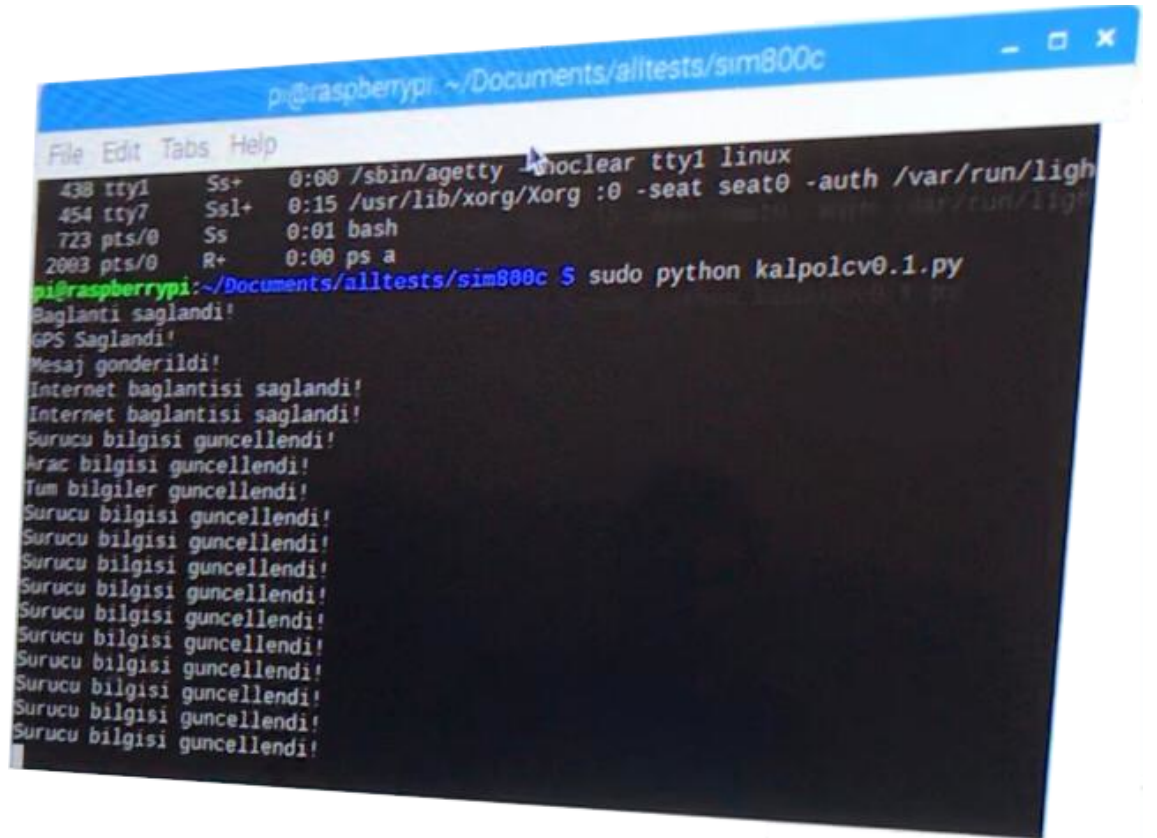
Minicom programı üzerinden aynı anda birden fazla cihaz okuma veya yazma yapabilir. Örneğin Arduino ile USB0 (/dev/ttyUSB0) üzerinden okuma yapılırken SIM800C USB1 (/dev/ttyUSB1) üzerinden okuma ve yazma yapılabilir. Burada dikkat edilmesi gereken bir nokta AT komutlarının tepki süreleridir. Bunun için SIM800C programlamada planlı gecikme uygulanmıştır.



```
pi@raspberrypi: ~
Dosya  Düzenle  Sekmeler  Yardım
23.40-0
39.19-0
55.14-96
65.69-96
65.69-96
80.21-96
80.21-96
42.65-96
56.50-94
56.50-94
40.92-94
40.94-94
51.81-95
62.73-95
44.20-95
57.54-95
62.57-95
62.57-95
41.40-95
69.14-95
69.42-95
69.48-95
73.60-95
```

Şekil 12: MAX30100 RPi Terminal çıktısı

Son olarak Şekil 7’de görüldüğü gibi SIM800C Shield RPi üzerinden çalıştırılmıştır. Öncelikle bağlantı kontrolü testi yapılıyor. SIM800C Shield bağlantıları düzgün bir şekilde yapıp RPi ile bağlantı sağlanabiliyorsa "Bağlantı sağlandı!" çıktısını verir. Daha sonra GPS için gerekli adımlar yapıldıktan sonra konum bilgisi sağlanabiliyorsa "GPS Sağlandı!" çıktısını verir. Daha sonra sürücü bilgileri, MAX30100’dan alınan nabız ve SpO2 değerleri ile konum bilgilerini belirtilen numaraya SMS olarak başarılı bir şekilde gönderebiliyorsa "Mesaj gönderildi!" çıktısını verir. Daha sonra da bu verileri veritabanına göndermek amacıyla modül üzerinden internet bağlantısı yapabiliyorsa "İnternet bağlantısı sağlandı!" çıktısını verir. İnternet bağlantısı başarılı olduktan sonra veritabanına verileri başarılı bir şekilde gönderebiliyorsa "Sürücü bilgisi güncellendi!" ile "Araç bilgisi güncellendi!" çıktılarını verir. En sonunda yapılan aşamalar tamamlandıktan sonra "Tüm bilgiler güncellendi!" çıktısı gözükür ve veriler düzenli bir şekilde tekrar güncellenir.



```
pi@raspberrypi: ~/Documents/alltests/sim800c
File Edit Tabs Help
438 tty1 Ss+ 0:00 /sbin/agetty --noclear tty1 linux
454 tty7 Ssl+ 0:15 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm
723 pts/0 Ss 0:01 bash
2003 pts/0 R+ 0:00 ps a
pi@raspberrypi:~/Documents/alltests/sim800c $ sudo python kalpolcv0.1.py
Bağlantı sağlandı!
GPS Sağlandı!
Mesaj gönderildi!
İnternet bağlantısı sağlandı!
İnternet bağlantısı sağlandı!
Sürücü bilgisi güncellendi!
Araç bilgisi güncellendi!
Tüm bilgiler güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
Sürücü bilgisi güncellendi!
```

Şekil 13: SIM800C Shield’ın çalıştırılması

4 SONUÇLAR VE ÖNERİLER

Proje çalışmamız boyunca en çok zorlandığımız aşamalar sensörlere enerji sağlamada, MAX30100 - RPi bağlantısı ve SIM800C konfigürasyonunda yaşadık. Sensörlerin gerekli gücü alamama ve yüksek gerilim seviyesine sahip mikrodenetleyici kullanmamız güç sorunlarını yaşamamıza sebep oldu. Bunun için harici pull-up direnç bağlantısı ve TTL dönüştürücüsü kullanarak üstesinden gelebildik. MAX30100 - RPi bağlantısı sorununu Arduino - RPi bağlantısı ile aştık. Son olarak SIM800C konfigürasyonu için datasheet incelemeleri ile üzerinden gelebildik.

Projemizin araba ile entegreli halinde MAX30100 Sensörü ile SIM800C Modülünü breadboard üzerinde arabanın arkasında kullandık. Şekil 8’de görüldüğü gibi arabanın çalıştırılmasında Ni-CD şarjlı pil kullanılmıştır. Şarj etme kısmında elimizde bulunan adaptörün sorunlu olmasından dolayı uzun süreli şarj etme fırsatımız olmamıştır. Ayrıca arabanın orta kısmında bulunan RPi’nin çıkarılabilir olması açısından sabitlenmemiştir.



Şekil 14: Proje son hali

KAYNAKLAR

- [1] CTAN,<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=27668> [Ziyaret Tarihi: 8 Mayıs 2019]
- [2] CTAN,https://www.researchgate.net/publication/301723832_Smart_real-time_healthcare_monitoring_and_tracking_system_using_GSMGPS_technologies [Ziyaret Tarihi: 8 Mayıs 2019]
- [3] CTAN,<https://www.arduino-libraries.info/libraries/max30100lib> [Ziyaret Tarihi: 9 Mayıs 2019]
- [4] CTAN,<https://linux.die.net/man/1/minicom> [Ziyaret Tarihi: 9 Mayıs 2019]