SİSTEM PROGRAMLAMA

2017 - 2018 BAHAR DÖNEMİ

DÖNEM PROJESİ : Ev Otomasyon Sistemi

152120141003 - Şafak AKINCI
152120141014 - Ahmet Gökçe BOZAN

İÇİNDEKİLER

1 - PROJE AMACI:

Raspberry Pi 3'ü, ev otomasyonunda kullanılacak bir alt sistem haline getirmektir. Ev otomasyonunda ihtiyaç duyulan algılayıcı(sensörler) ve sürücüler (röle) doğrudan Raspberry Pi 3'e bağlanacaktır. Bu sistem bir sunucu modül gibi çalışarak, üst seviyede yazılan uygulamalara ağ üzerinden ve seri bağlantı üzerinden algılayıcı ve sürücülere erişim imkanı sağlayacaktır.

2 - PROJE DETAYI:

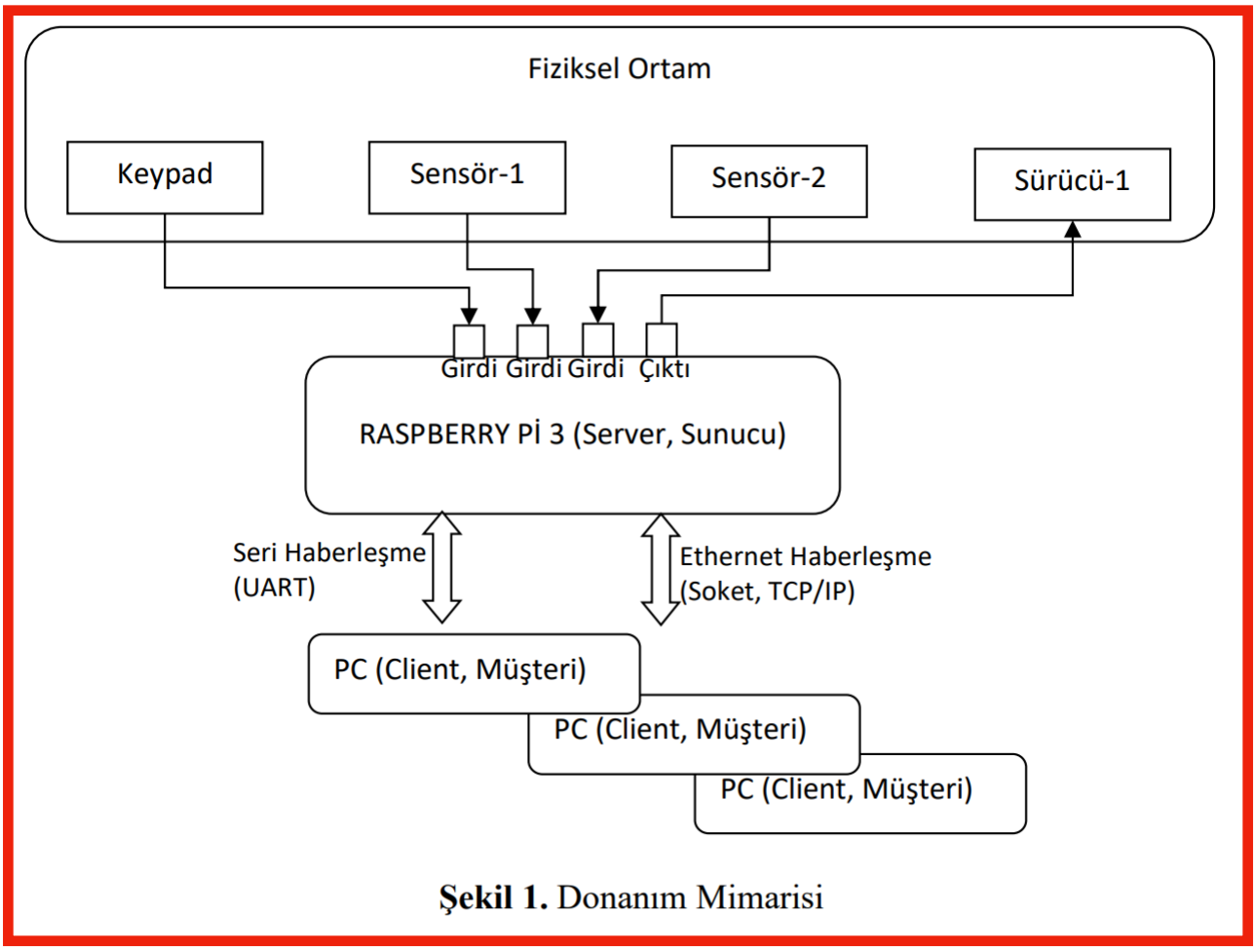
Adım 1. Donanım Tasarımı ve Gerçeklemesi:
Tasarımda, aşağıda belirtilen giriş çıkış birimleri kullanılmıştır.

İkilik sistemde (0 veya 1) çıktı üreten sensörler:	**Eğim Sensörü, Mesafe Sensörü**
Sürücü olarak:	**Röle**
Klavye olarak:	**4x4 Membrane Keypad**

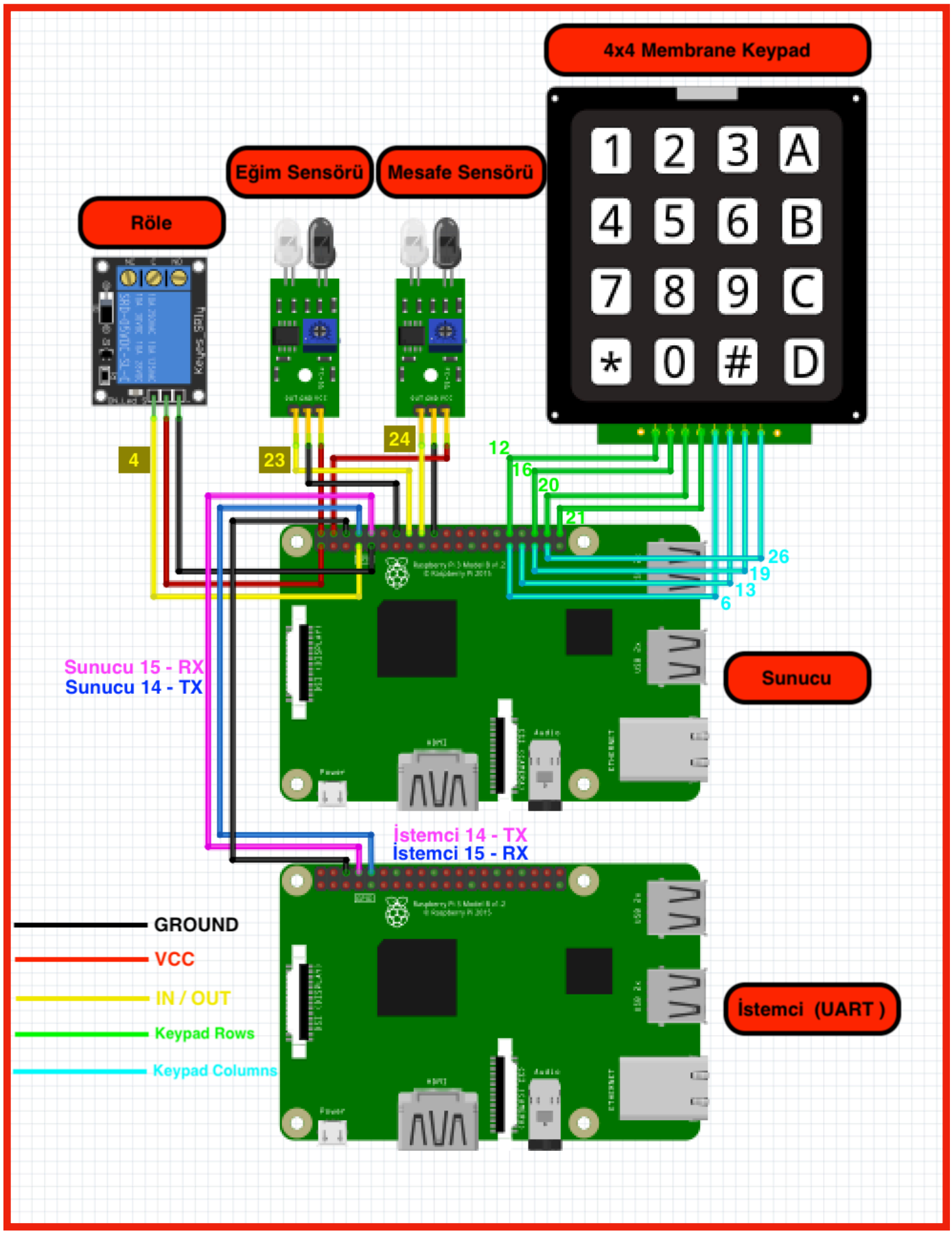
Eğim sensörü, eğimi algılayabilmek için içlerinde civa damlacığı ya da metal bilye bulundurur. Bu sensörler bulundukları konuma göre içlerindeki civa damlacığının ya da metal bilyenin sensör içerisindeki anahtarları açması ya da kapamasıyla çalışır.

Mesafe sensörleri, gelişmiş yansımalı sensörlerdir. Bu sensörler bir IR verici led tarafından yayılan IR ışığın IR alıcı modül ile toplanması mantığıyla çalışırlar. IR ışınların yayılımı kodlanmıştır ve kesik kesiktir.

Röle, düşük akımlar kullanarak yüksek akım çeken cihazları anahtarlama görevinde kullanılan devre elemanıdır. Rölenin bobinine enerji verildiğinde mıknatıslanan bobin bir armatürü hareket ettirerek kontakların birbirine temasını sağlar ve devrede iletim sağlanmış olur. Örnek kullanım alanları sigortalar, alarm sistemleri, elektrikli ev aletleri vb.



Şekil 1. Donanım Mimarisi

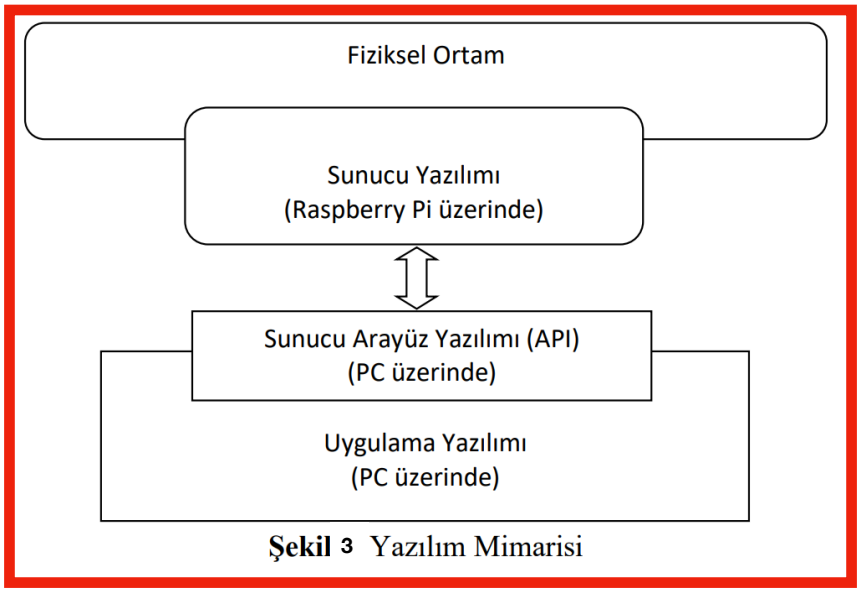


Şekil 2: Donanım Tasarımı ve Gerçeklemesi

Adım 2. Yazılım Tasarımı ve Gerçeklemesi

Şekil 3'de gösterildiği gibi, sistemde üç farklı yazılım geliştirilmiştir.

1. Raspberry Pi üzerinde koşacak sunucu yazılım.
2. Raspberry Pi'ye Ethernet (Socket) ya da Seri Port (UART) üzerinden erişecek olan, PC tarafından uygulama geliştirmede kullanılacak olan kütüphane. Raspberry pi ile haberleşme ve mesaj denetimleri bu kütüphanedeki fonksiyonlarla sağlanacaktır.
3. İlk iki yazılımı test etmek üzere çalıştırılacak uygulama yazılımıdır.



Şekil 3 Yazılım Mimarisi

Adım 2. 1. Sunucu Yazılım

- *Sürücü için açma ya da kapama talebi geldiğinde, buna göre sürücünün durumu değiştirilecektir.*

İstemcilerden gelen istekleri bağlantı türüne göre (UART ya da SOKET) dinlemek için iki adet iş parçacığı kullanılmaktadır. (**threadReadUART, threadCheckSocketMessage**)

- *Program çalışması ile birlikte, soket yaratarak bağlantı istekleri için bekleyecek, bağlantı isteği alırsa bağlantıyı kuracak, bağlantı kesildiğinde yeni bir bağlantı için beklemeye devam edecektir.*

Bir iş parçacığı (**threadMakeConnection**) sürekli olarak, soket üzerinden gelen bağlantı isteklerini dinler ve kabul eder (Sunucuya en fazla 5 istemci bağlanabilir.)

- *Sensörler periyodik olarak okunarak, lokal değişkenlerde güncel değerleri tutulacaktır.*

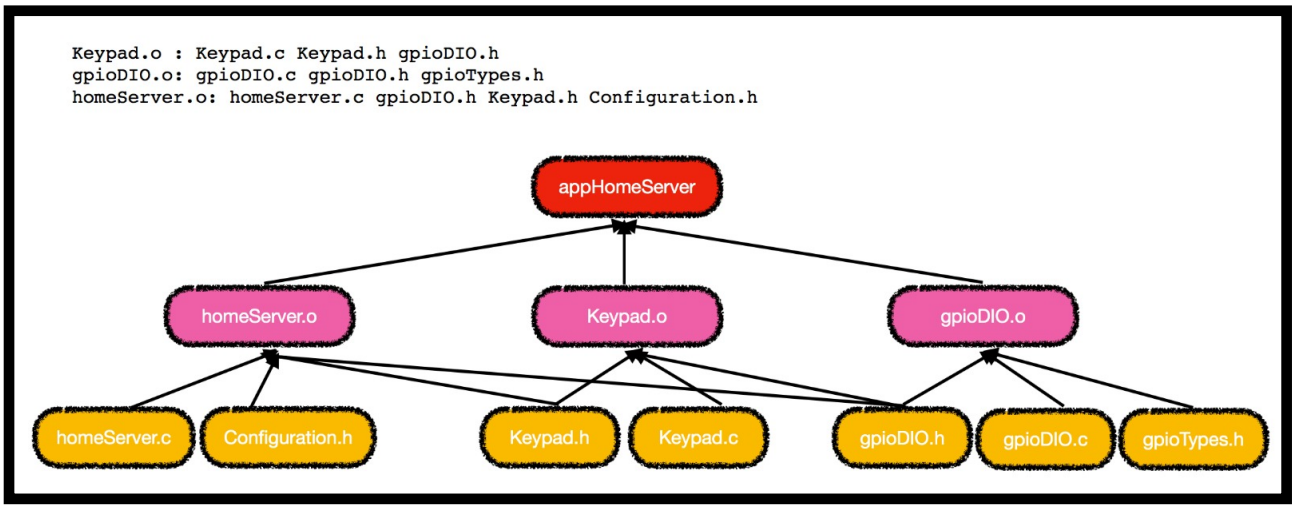
Bir iş parçacığı **(threadReadSensorValues)** sürekli olarak, sensörlerden gelen değerleri okur ve lokal değişkenlerde tutar.

- *Keypad sürekli dinlenecektir (Kapı şifresi gibi düşünülebilir). Keypad üzerinden, şifre değişimi ve şifre ile giriş yaparak role açma ve kapama işlemi yapılabilecektir. Buna göre; XXXX*YYYY# (tuşlandığında, XXXX doğru şifre ise, YYYY olarak değiştirilecektir.) XXXX*0# (tuşlandığında, XXXX doğru şifre ise, röle kapatılacaktır.) XXXX*1# (tuşlandığında, XXXX doğru şifre ise, röle açılacaktır.) Sistem ilk çalıştığında şifre 0000 olarak tanılanacaktır. Kullanıcı isterse, şifreyi istediği bir şifre ile yukarıdaki tuşlamaları kullanarak gerçekleştirecektir.*

Bir iş parçacığı, (**threadListenKeypad**) sürekli olarak, sunucuya bağlı olan keypadi dinler ve gerekli işlemleri yapar.

ALINAN MESAJ	CEVAP MESAJI	EYLEM
sensorDurum:	sensorDurum <durum>:	Güncel sensör durumları paketlenerek gönderilir.
sensorSayi:	sensorSayi <sensor sayisi>:	Sunucuya bağlı toplam sensör sayısını gönderir.
sensorList:	sensorList <ad1, ad2>:	Sunucuya bağlı sensör isimlerini gönderir.
surucu <0 ya da 1>:	surucu <ok veya error>:	Sürücü istenen duruma getirilir ve istemciye durum bilgisi iletilir.
surucuDurum:	surucuDurum <0 veya 1>:	Sürücünün güncel durumu istemciye iletilir.
autoMod <on sensorId sensorDeğeri>:	autoMod <ON>:	İstemci, otomatik moddan çıkana kadar, belirtilen sensörün durumuna göre röle kontrolü yapılır.
autoMod <off>:	autoMod <OFF>:	Otomatik moddan çıkıldığı istemciye iletir.
close:	close <OK>:	Sunucu ile istemci arasındaki bağlantı kapatılır.

Sunucu yazılımının derlenmesi için gerekli olan makefile dosyasını bağlılık ağacı Şekil 4'de gösterilmiştir.



Şekil 4: Server Makefile

Adım 2. 2. Sunucu Arayüz Yazılımı

Sunucu ile istemci arasında iletişimi sağlayacak kütüphane yazılmıştır ve raporla beraber sunulacaktır. Aşağıda belirtilen dosyaların içinde, iletişimi sağlayacak gerekli fonksiyonlar kodlanmıştır.

CommunicationAPI.c
CommunicationAPI.h

Sunucu arayüz yazılımının terminal üzerinden **"libCommunicationAPI.so"** isimli dinamik kütüphane olarak derlenmesi için gerekli olan adımlar aşağıdaki gibidir.

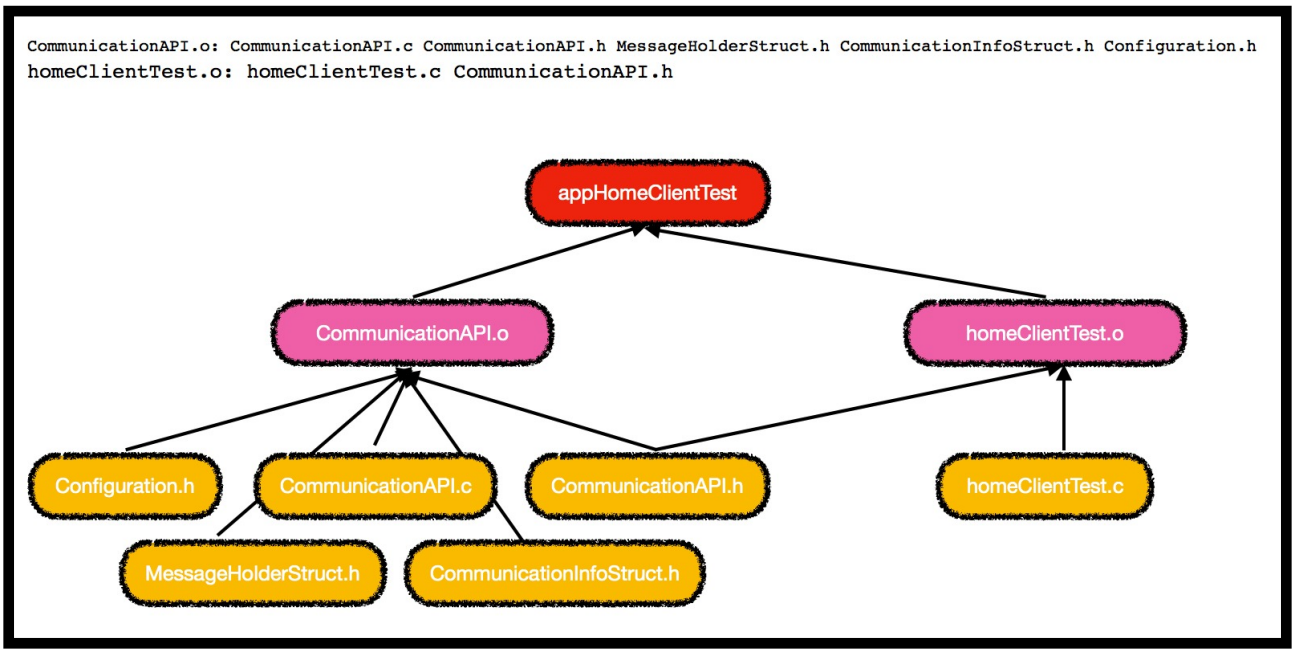
```
$ gcc -c -Wall -fpic CommunicationAPI.c -o CommunicationAPI.o

$ gcc -shared -o libCommunicationAPI.so CommunicationAPI.o

$ export LD_LIBRARY_PATH=/home/pi:$LD_LIBRARY_PATH

$ gcc -L./ -I./ -Wall -o appHomeClientTest homeClientTest.c -lCommunicationAPI
```

Adım 2. 3. Uygulama Yazılımı

Sunucu ile geliştirilen arayüz yazılımı kullanılarak oluşturulan "appHomeClientTest" uygulama yazılımı, kullanıcıya bir menü sunarak aşağıda listelenen işlemleri gerçekleştirecektir.

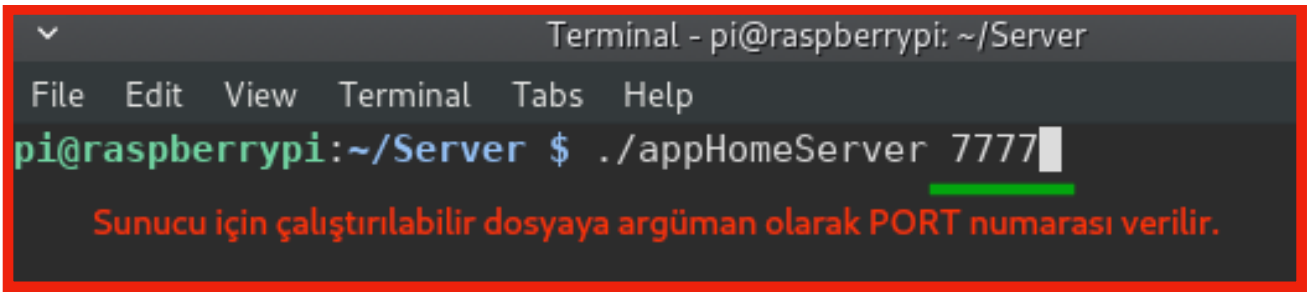
1. Güncel sensör değerlerinin öğrenilmesi
2. Toplam sensör sayısının öğrenilmesi
3. Sunucuda mevcut olan tüm sensörlerin isimleri ve id'lerinin listelenmesi
4. Röle durumunun değiştirilmesi
5. Güncel röle durumunun öğrenilmesi
6. Automod'a geçilmesi
7. Sunucu ile bağlantının kapatılması ve programın sonlandırılması

Uygulama yazılımı için gerekli olan makefile dosyasına ait bağlılık ağaçı Şekil 5'de gösterilmiştir.

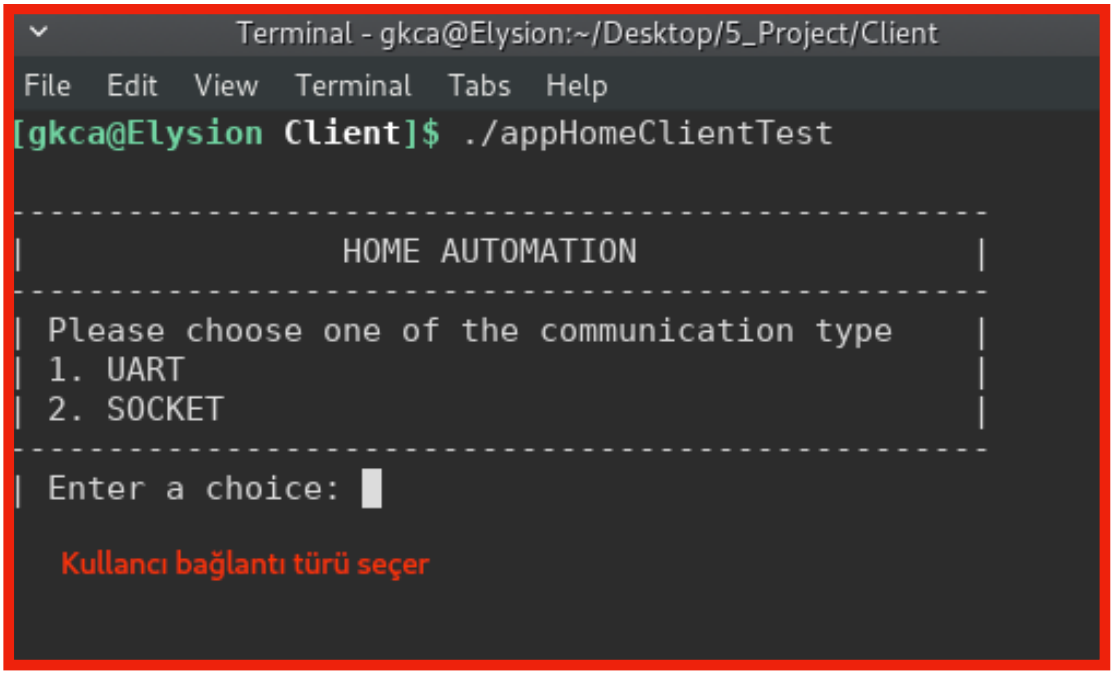


Şekil 5 : Client Makefile

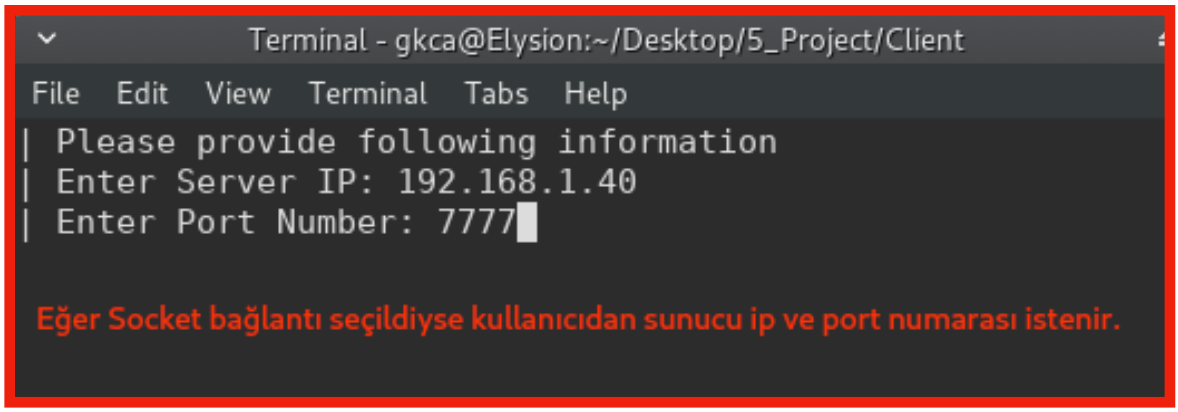
UYGULAMA ÇIKTILARI



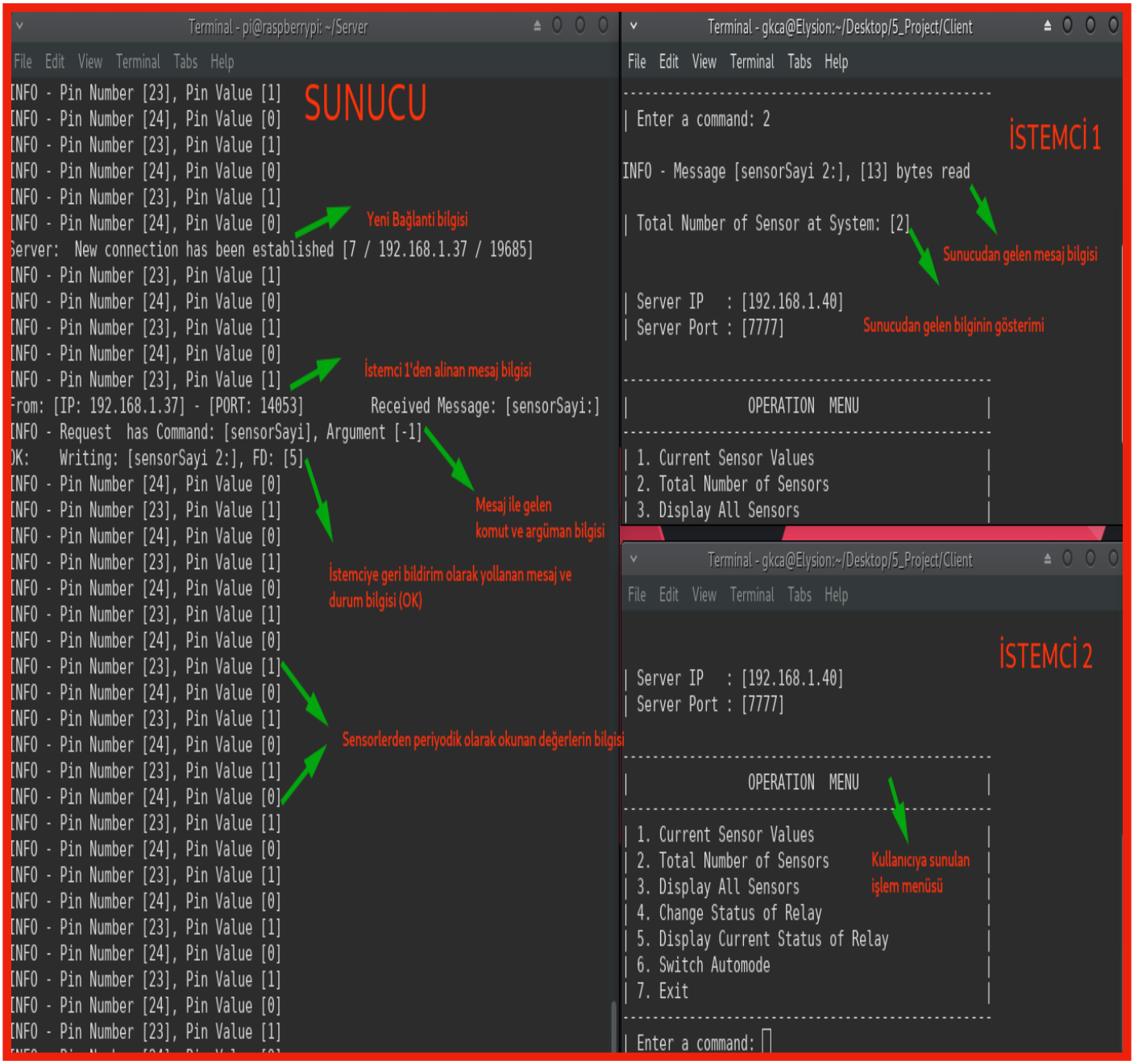
Şekil 6 : Sunucu Başlatma



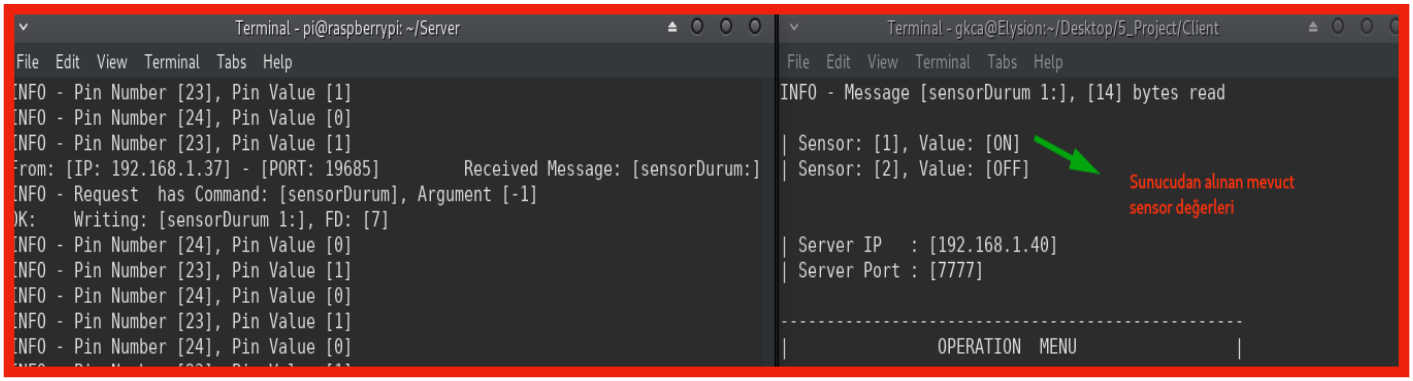
Şekil 7: Kullanıcı Başlatma



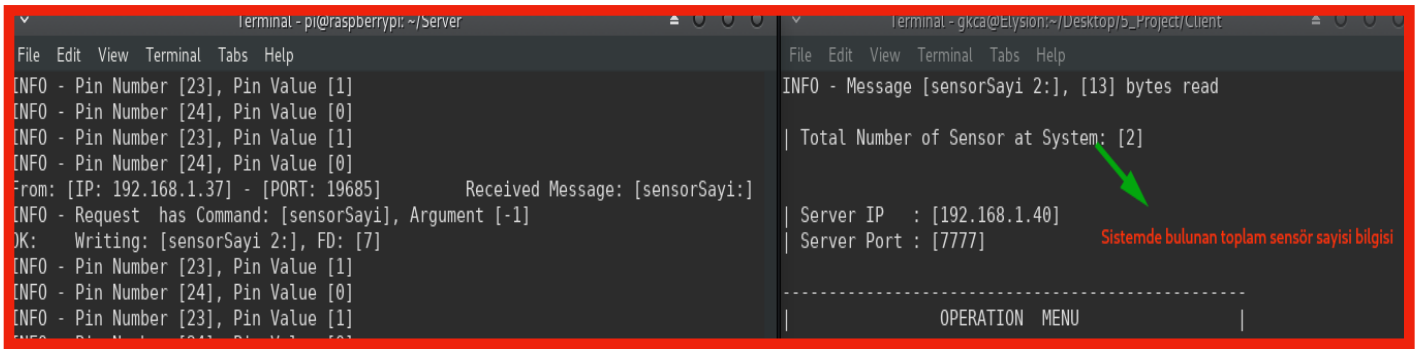
Şekil 8: Socket Başlatma



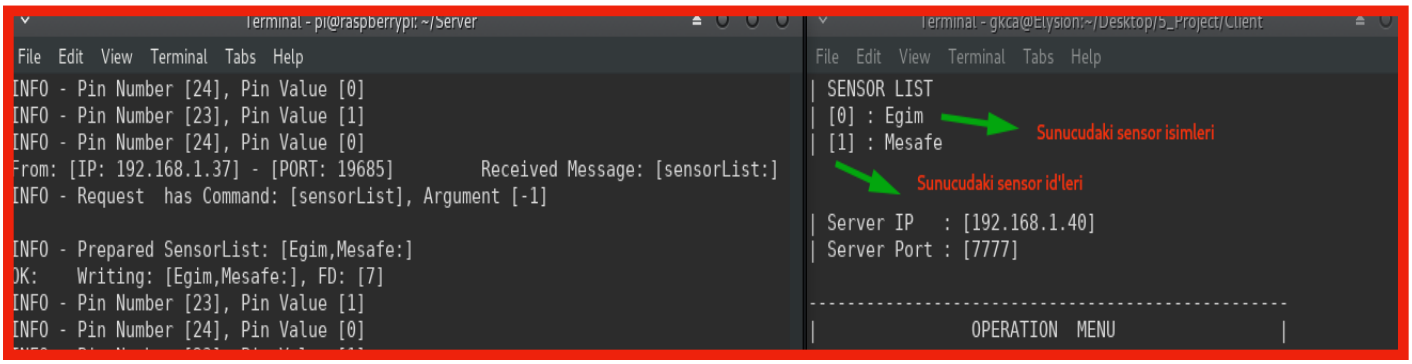
Şekil 9 : Sunucu ve İki İstemci



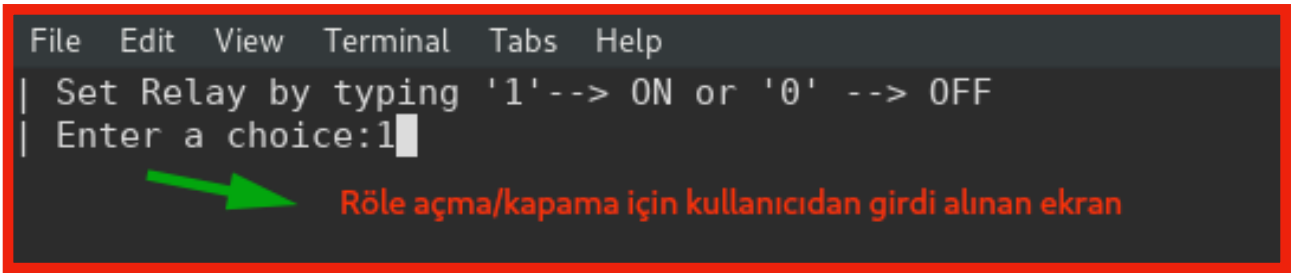
Şekil 10: Güncel Sensör Değerlerinin Öğrenilmesi



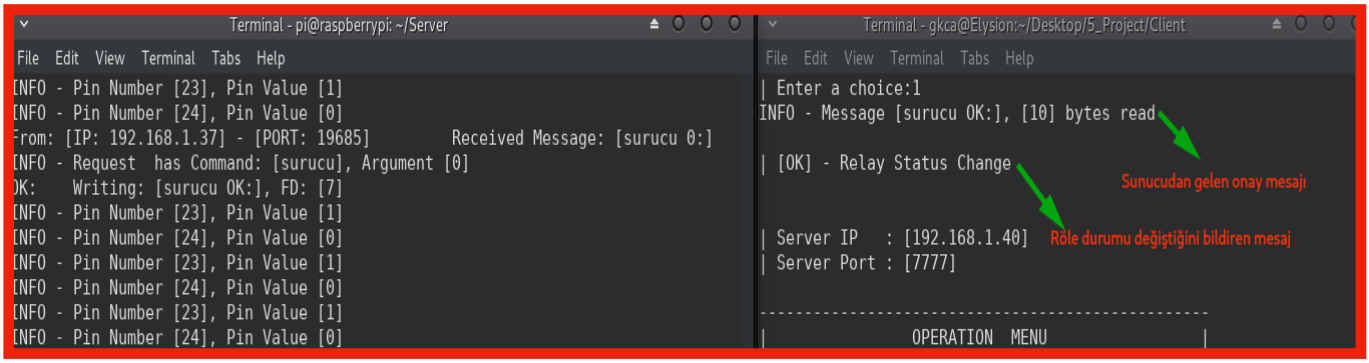
Şekil 11: Toplam sensör sayısının öğrenilmesi



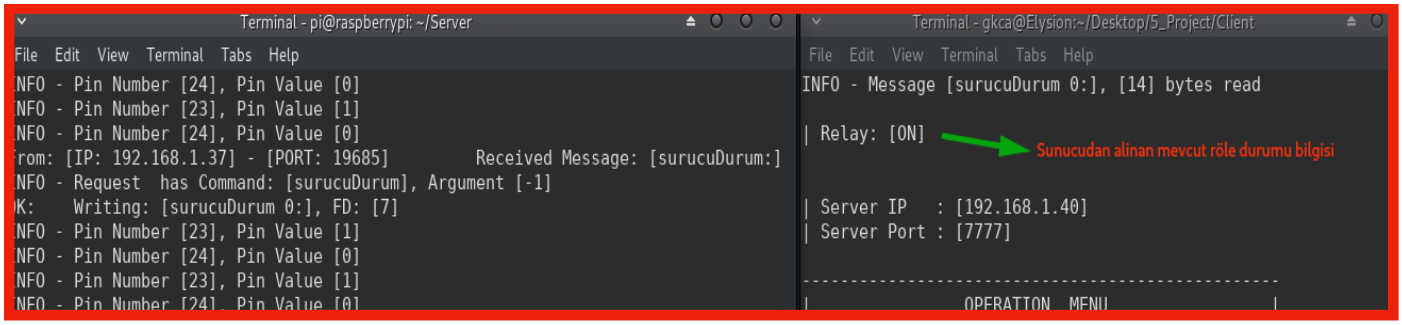
Şekil 12: Sunucuda mevcut olan tüm sensörlerin isimleri ve id'lerinin listelenmesi



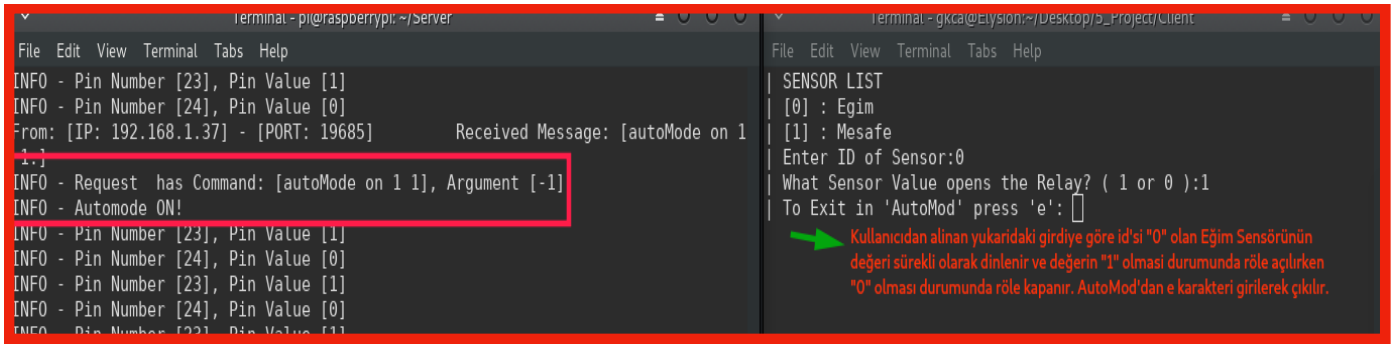
Şekil 13: Röle durumunun değiştirilmesi



Şekil 14: Röle durumunun değiştirilmesi



Şekil 15: Güncel röle durumunun öğrenilmesi



Şekil 16: Automod'a geçilmesi