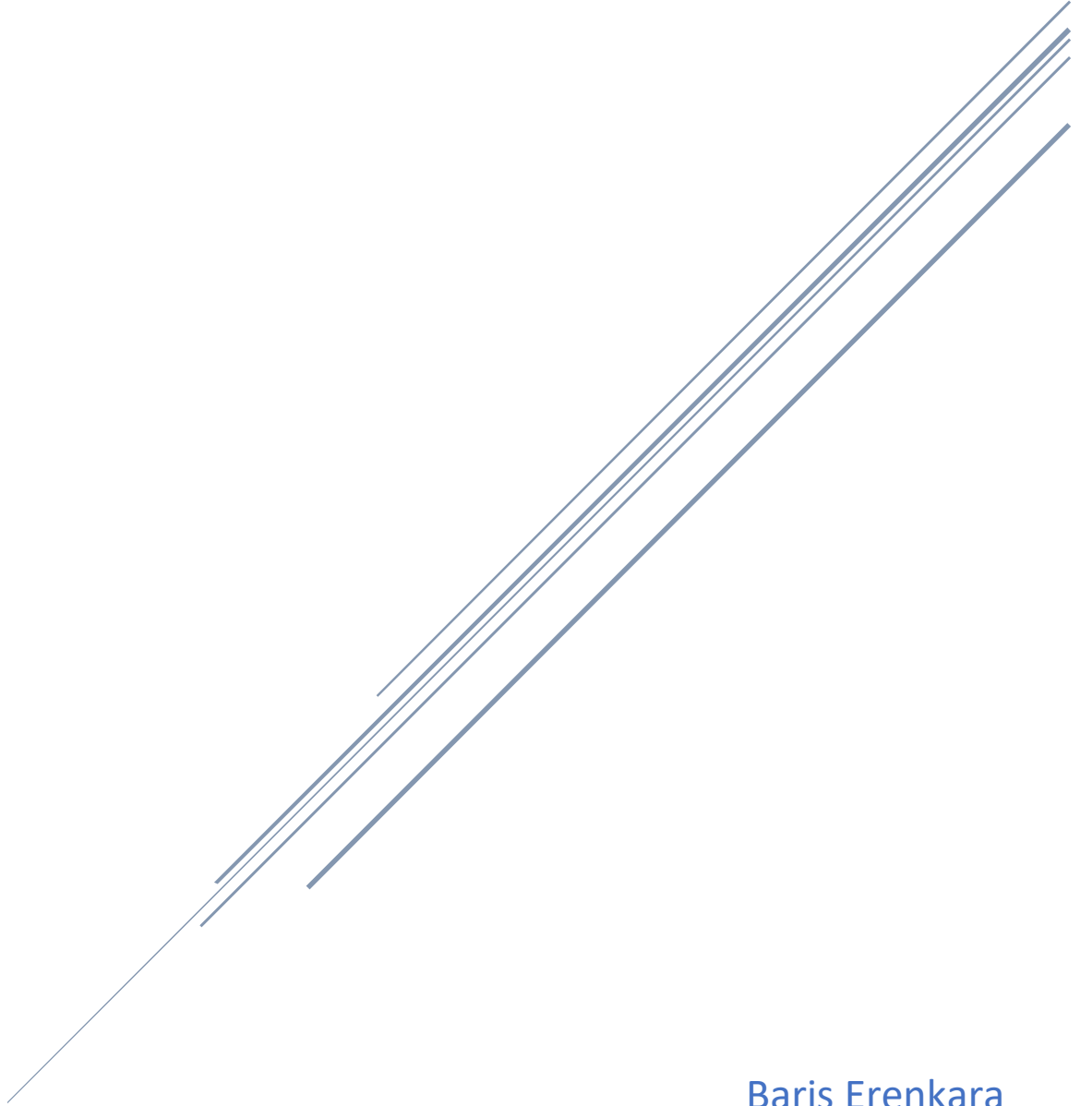


SİSTEM PROGRAMLAMA (A) (152118513)

2018-2019 BAHAR DONEM SONU PROJESİ



Baris Erenkara

152120141099

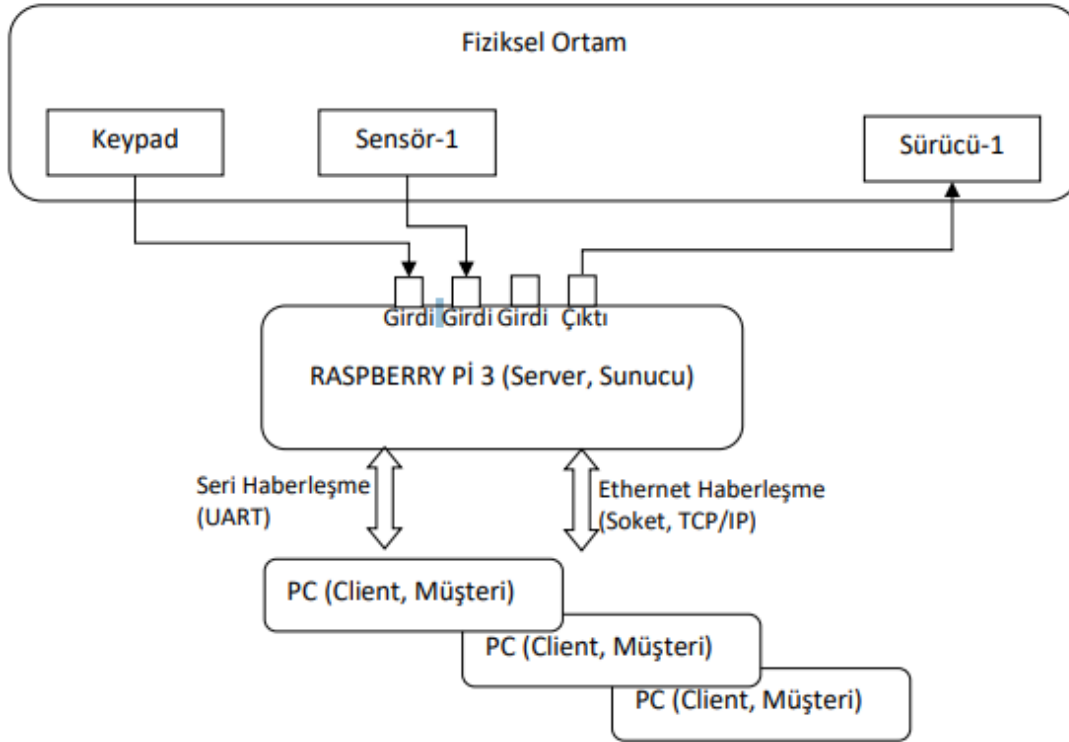
İÇİNDEKİLER

SİSTEM PROGRAMLAMA (A) (152118513).....	0
proje tanımı	2
tasarım	2
PIN DIYAGRAMLARI	2
sensor	3
sürücü (RÖLE)	3
TUŞ TAKIMI (KEYPAD).....	4
YAZILIM	4
SUNUCU	4
İSTEMCI	4
EKRAN ÇIKTILARI	5

PROJE TANIMI

Bu projede amaç, Raspberry Pi 3'ü ev otomasyonunda kullanılacak bir alt sistem haline getirmektir. Ev otomasyonunda ihtiyaç duyulan algılayıcı ve sürücüler 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.

TASARIM



PIN DIYAGRAMLARI

KEYPAD	1	2	3	4	5	6	7	8
RASPBERRY	GPIO16	GPIO20	GPIO12	GPIO12	GPIO06	GPIO13	GPIO19	GPIO26
IN/OUT	IN	IN	IN	IN	OUT	OUT	OUT	OUT
SENSOR	VCC		IN		GROUND			
RASPBERRY	PIN02		GPIO22		PIN09			
SÜRÜCÜ	VCC		IN		GROUND			
RASPBERRY	PIN04		GPIO27		PIN06			

SENSOR



İkili sistemde çıktı üreten bir sensördür. Eğim olduğu takdirde 1, düz olduğu takdirde 0 değerini geri döndürür.

SÜRÜCÜ (RÖLE)



Yüksek gerilim/ yüksek akım aygıtları kontrol kabiliyeti sağlar. Projede sadece sensöre bağlı olarak açıp kapatmak için kullanıldı.

TUŞ TAKIMI (KEYPAD)



Basılan tuşlara göre raspberry'e sinyal gönderir. Projede şifre değiştirme, röle durumu değiştirme gibi işlemler için kullanıldı.

YAZILIM

SUNUCU

İşlemlerde bulunan periyodik işlemler için iş parçaçıkları kullanıldı. Keypad dinleme işlemi için **threadListenKeypad**, sensör ve sürücü durumlarını okumak için **threadReadSensor**, ağ soket dinleme ve bağlantılara cevap verme işlemi için **threadSocketService**, Uart işlemlerini kontrol etme için **threadUARTService**, Otomatik mod işlemleri için ise **threadAutoMode** isimli iş parçaçıkları fonksiyonları kullanıldı.

Sunucudaki sabit dğeişkenler gibi birtakım bilgiler ise **config.h** isimli dosyada tutulurken, keypad işlemleri için daha önce uygulaması yapılan keypad kütüphanesinden yararlanıldı. Keypad ile girilen işlemin yapılması için ise **DoKeypadOperations** isimli fonksiyon kullanıldı.

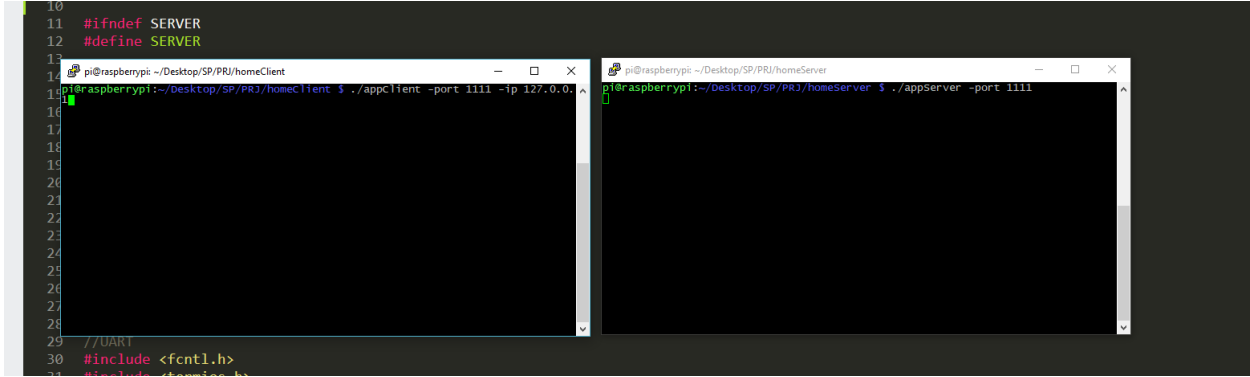
İSTEMCI

Menu için **printMainMenu** isimli fonksiyon kullanıldı. Kullanıcının Uart ya da ağ üzerinden haberleşmek istediği ise girilen argümanlar ile anlaşılıyor. Uart için **-uart** argümanı, ağ için ise **-port** ve **-ip** argümanları gerekiyor.

Uygulamanın sunucu ile haberleşmesi için ise **connection.h** kütüphanesi oluşturuldu. Bu kütüphanedeki fonksiyonların gerçekleştirilmesi ise **connection.c** isimli kaynak dosyada oluşturuldu.

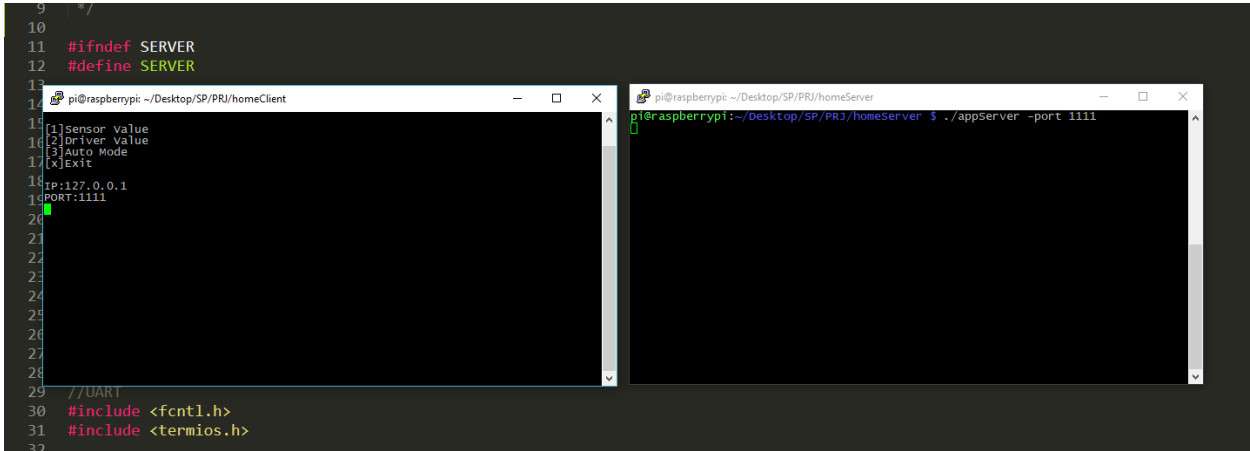
EKRAN ÇIKTILARI

Argümanlar ile uygulamaları çalıştırma.



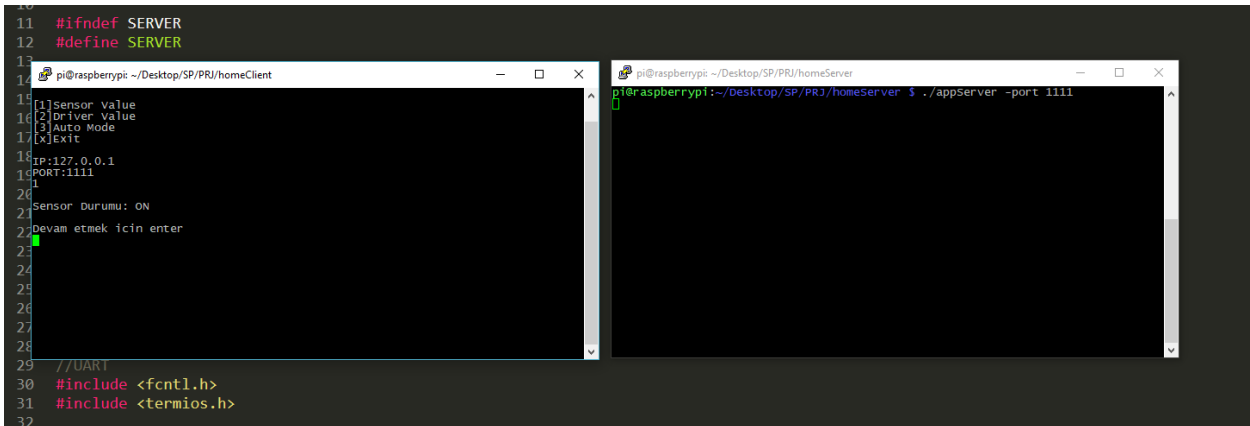
```
10
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient $ ./appClient -port 1111 -ip 127.0.0.1
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29 //UART
30 #include <fcntl.h>
31 #include <termios.h>
```

İstemci uygulamasının ip ve port üzerinden sunucuya bağlanması.



```
9
10
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient $ ./appClient -port 1111 -ip 127.0.0.1
15
16 [1]Sensor Value
17 [2]Driver Value
18 [3]Auto Mode
19 [x]Exit
20
21 ip:127.0.0.1
22 port:1111
23
24
25
26
27
28
29 //UART
30 #include <fcntl.h>
31 #include <termios.h>
```

İstemci uygulaması üzerinden sensör durumunu görüntüleme.



```
10
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient $ ./appClient -port 1111 -ip 127.0.0.1
15
16 [1]Sensor value
17 [2]Driver value
18 [3]Auto Mode
19 [x]Exit
20
21 ip:127.0.0.1
22 port:1111
23
24 Sensor Durumu: ON
25 Devam etmek için enter
26
27
28
29 //UART
30 #include <fcntl.h>
31 #include <termios.h>
```

Istemci uygulaması üzerinden sürücü durumunu görüntüleme.

```
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient
15 [1]Sensor Value
16 [2]Driver Value
17 [3]Auto Mode
18 [x]Exit
19 ip:127.0.0.1
20 PORT:1111
21
22 Surucu Durumu: OFF
23 Devam etmek için enter
24
25
26
27
28
29 //UART
30 #include <fcntl.h>
31 #include <termios.h>
```

Istemci uygulaması üzerinde otomatik mod hizmeti.

```
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient
15 [1]Sensor Value
16 [2]Driver Value
17 [3]Auto Mode
18 [x]Exit
19 ip:127.0.0.1
20 PORT:1111
21
22 Sensörün hangi durumunda sürücü ON olsun? (0-OFF/1-ON/2-Devre dışı bırak)
23
24
25
26
27
28
29 //UART
30 #include <fcntl.h>
```

Keypad aracılığıyla şifre değiştirme işlemi.

```
11 #ifndef SERVER
12 #define SERVER
13
14 pi@raspberrypi: ~/Desktop/SP/PRJ/homeClient
15 [1]Sensor Value
16 [2]Driver Value
17 [3]Auto Mode
18 [x]Exit
19 ip:127.0.0.1
20 PORT:1111
21
22 Sensörün hangi durumunda sürücü ON olsun? (0-OFF/1-ON/2-Devre dışı bırak)
23
24
25
26
27
28
29 //UART
30 #include <fcntl.h>
```

```
pi@raspberrypi: ~/Desktop/SP/PRJ/homeServer $ ./appServer -port 1111
INFO - Keypad Input: [0000*1111#], MessageSize: [10]
Password has changed!
```

Keypad aracılığıyla yanlış şifre denemesi ve sürücü durumunu değiştirme işlemleri.

```
383     /*printf("%4s\n",currentPasswd);
384     printf("%1s\n",chStar);
385     printf("%4s\n",currentPasswd);
386
387     [1]Sensor Value
388     [2]Driver Value
389     [3]Auto Mode
390     [X]Exit
391     IP:127.0.0.1
392     PORT:1234
393
394
395
396
397
398
399
400
401     else
402         printf("\nWARNING - Authentication Failed, Wrong Password");
```

```
pi@raspberrypi: ~/Desktop/SP/PRJ/homeServer
pi@raspberrypi:~/Desktop/SP/PRJ/homeServer $ ./appServer -port 1234
INFO - Keypad Input: [000011111], MessageSize: [10]
WARNING - Authentication Failed, Wrong Password
INFO - Keypad Input: [0000*1111#], MessageSize: [10]
Password has changed!
INFO - Keypad Input: [1111*1#], MessageSize: [7]
Now Relay [ON]
INFO - Keypad Input: [1111*0#], MessageSize: [7]
Now Relay [OFF]
```