



YTÜ

BLM4021 Gömülü Sistemler Laboratuvar Projesi Final Raporu

Grup No: **24**

Proje No: **1**

Kişilerin Çalışma Yüzdesi:

Grup Sorumlusu:	Eray Zeki Sah	25	Yüz Tanıma / Raspberry Pi Kurulumları / Sistem Testleri
	Emre Kuru	25	Keypad Kurulumu / Proje Raporlama / Sistem Testleri
	Ferhat Taş	25	Hareket Sensörü Kurulumu / Proje Raporlama / Sistem Testleri
	Onur Demir	25	NodeMCU Kurulumu / Malzeme Araştırması / Sistem Testleri

İçerik

I.	Giriş ve Proje Tanıtımı.....	Sayfa: 3
II.	<i>Fritzing</i> ile Ön Tasarım.....	Sayfa: 5
III.	Kurulan Devre Detayları.....	Sayfa: 7
IV.	Yazılım Tasarımı.....	Sayfa: 11
V.	Sonuçlar, Demo Detayları ve Sunum Linki.....	Sayfa: 15
VI.	Referanslar.....	Sayfa: 18

I. Giriş ve Proje Tanıtımı

Günümüzde güvenlik sistemleri ev ve binaların güvenliği için önem arz etmektedir. Binanın içinde tanımadık biri (hırsız) binanızdaki eşyaları çalabilir. Genellikle hırsızlar binaya giren ev sahiplerinin yanında diğer ev sahiplerinin tanıdığınıymış gibi girerek binada bulunan ortak eşyaları ayakkabıları, bisikletleri çalabilmektedir. Ayrıca şüpheli davranışları tespit etmek için günümüzde sabit kameralar kullanılmaktadır ve bu da tam olarak bir gözetim sağlamak için birden çok kameranın farklı noktaları görmesi gerektiği anlamına gelir.

Projede kamera gibi pahalı bir aygıtı birçok yere yerleştirmek çok büyük maliyetler gerektirdiğinden daha düşük bütçeli bir yaklaşımın var olabileceğinin fizibilitesi yapıldı. Fizibilite sonucunda PIR (hareket sensörü) gibi ultrasonik (infrared) dalgaları kullanan sensörlerin kameralara göre çok daha ucuz bir maliyetle alınabileceği tespit edildi. Bir gömülü sistem tasarımında en önemli kriterlerden biri kuşkusuz sistemin ucuz bir şekilde elde edilebilmesidir. Bu projede de buna dikkat edilerek en ucuz olacak şekilde efektif bir proje geliştirilmiştir.

Proje kapsamında gömülü sistemde kullanılan elemanlar aşağıdaki gibidir:

- Raspberry Pi
- Kamera
- PIR (Hareket Sensörü)
- Keypad
- Buzzer (Alarm sesi için)
- NodeMCU
- Çeşitli Led'ler

Gömülü sistemde kullanılan yukarıdaki devre elemanları detaylı olarak Bölüm 3'te anlatılacaktır.

Projede geliştirilen gömülü sistemin akışı kısaca aşağıdaki gibidir:

1. Kullanıcının yüzü sisteme tanıtılacak, ayrıca keypad ile de girilebilmesi için bir şifre belirlenecek.
2. Sistem kurulduktan sonra ev içinde bulunan hareket sensörü ile hareket tespit edildiğinde kullanıcıdan şifre girmesi veya yüzünü göstermesi beklenecektir.
3. Kullanıcı evdeyse kameraya yüzünü göstererek (Yüz Tanıma) ya da keypad ile daha önceden belirlediği şifreyi girerek hareketi yapanın kendisi olduğunu sisteme bildirebilecek. Böylece alarm şiddetlenmeden susturulacak.

4. Eğer hareketi yapan kişi belli bir süre içinde yukarıdakilerden birini yapmazsa ev içinde alarm şiddetli bir şekilde çalmaya başlayacak. (Evde tanınmayan biri var uyarısı!)

Geliştirilen alarm sistemi evde kurulmadan önce yapılması gereken işlemler şunlar olacaktır:

- Kullanıcının Yüzünün Sisteme Tanıtılması: Bu işlem için yüz tanıma algoritması ile kullanıcının yüzüne veya eklemek istediği diğer yüzlere (ev içindeki diğer kişiler vs.) ait encoding'ler bulunup sistemin veritabanında saklanacaktır. Sistem devreye girdikten sonra veritabanındaki yüzlere ait encoding'ler canlı olarak gelen encoding ile karşılaştırılıp kullanıcının yüzü tespit edilerek onaylanacaktır.
- Şifre Belirlenmesi: Sistemi kullanan kişi yüz tanıma dışında şifre ile tanıma işlemi yapmayı tercih edebilecektir. Bu kapsamda kullanıcının alarm sistemini kurmadan önce yine bir şifre belirlemesi gerekecektir. Daha sonra hareket tespit edildiğinde kullanıcı şifre ile de hareketi yapanın kendisi olduğunu sisteme onaylatabilecektir.

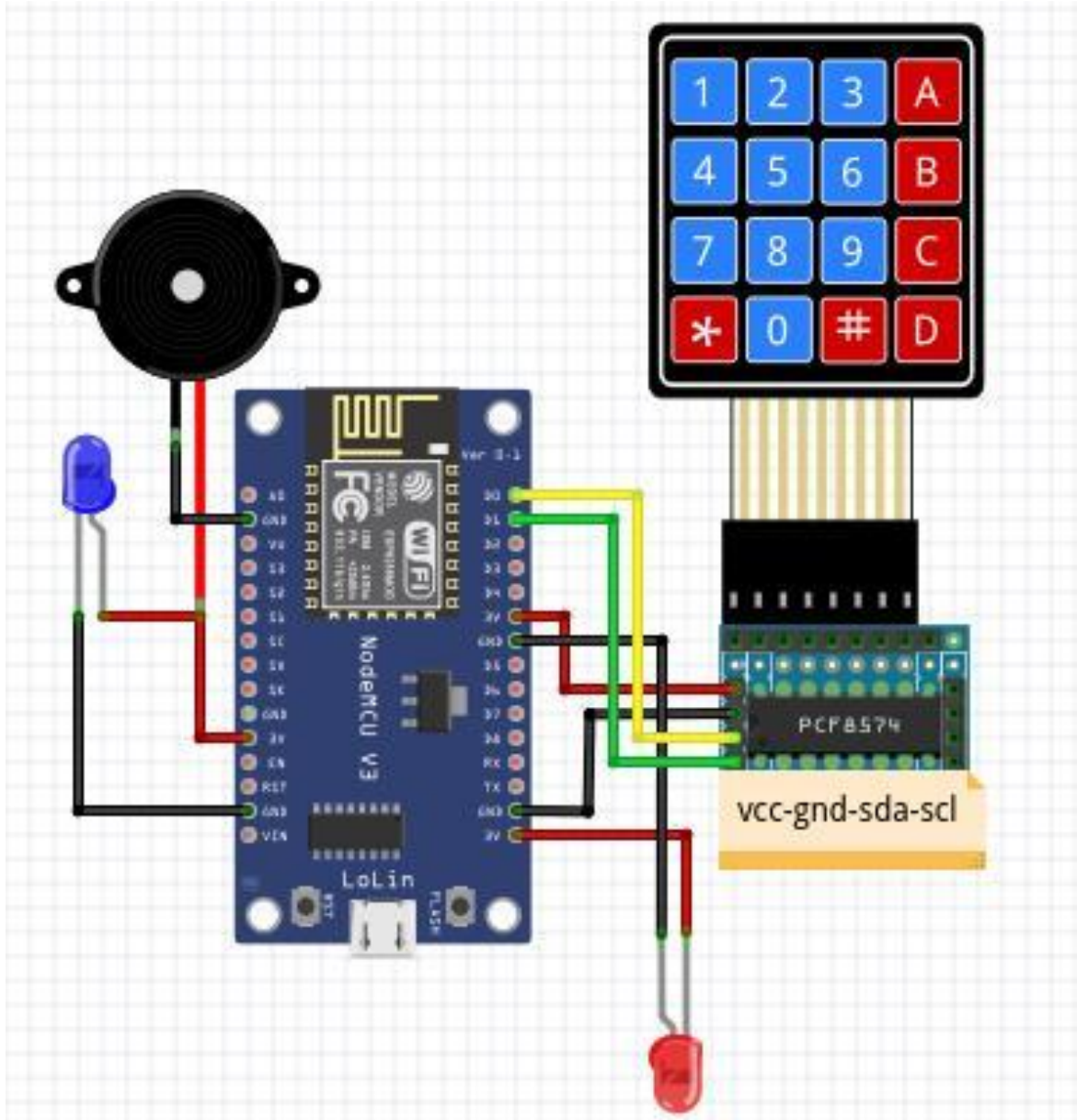
Yukarıdaki iki işlem de kullanıcı tarafından gerçekleştirildikten sonra alarm sistemi gönül rahatlığıyla kurulabilecektir.

-O-

II. *Fritzing* ile Ön Tasarım

Fritzing’te donanımın ön tasarımı yapılmadan önce devrede hangi donanımların kullanılacağına karar verildi. Buna göre sistemde donanımsal olarak diğer küçük elemanları yönetecek iki ana eleman bulunacaktır. Bunlardan biri Raspberry Pi, diğeri ise NodeMCU olacaktır.

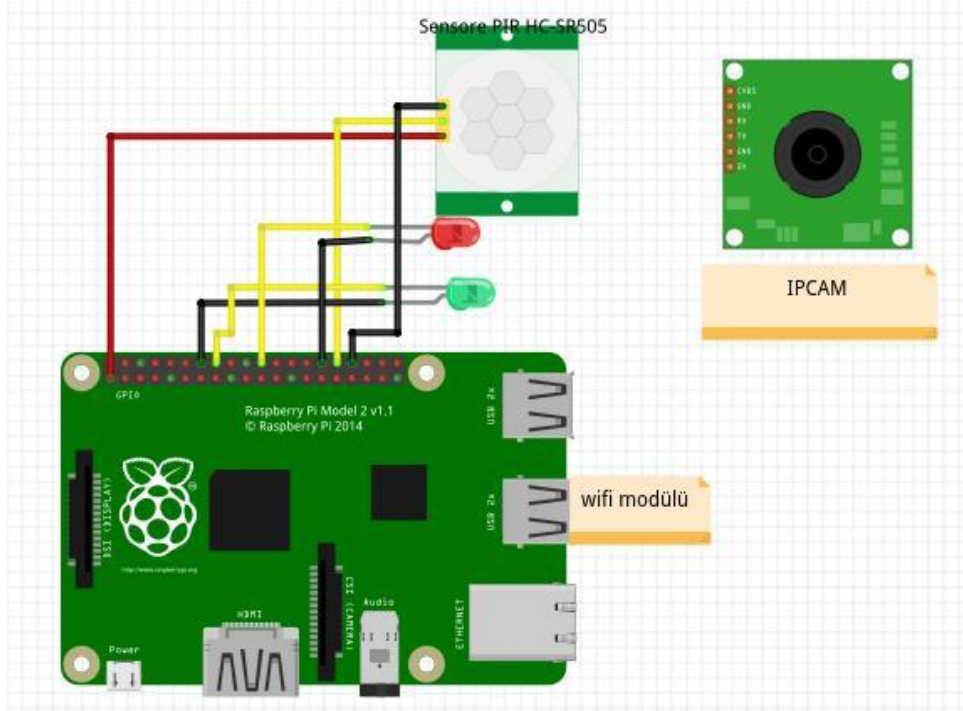
Şekil 1’de görüldüğü gibi NodeMCU’ya i2c bağlantılı keypad, kırmızı ve mavi led, bir de buzzer bağlanmıştır.



Şekil 1: NodeMCU’nun keypad, buzzer ve led’lerle bağlantısı.

Şekil 2’de görüldüğü gibi Raspberry Pi’ye buzzer (alarm çalması için), yeşil-kırmızı led ve hareket sensörü (PIR) bağlanır. Sarı kablo ile gösterilen bağlantılar veri, kırmızı ile gösterilenler gerilim siyah ile gösterilenler ise toprak bağlantılarıdır. Buna ek olarak SDA ve SCL bağlantıları sırasıyla sarı ve yeşil kablo olarak kullanılmıştır.

Yüz tanıma için kullanılacak IP Kamera ise WiFi modülü aracılığıyla Raspberry Pi ile haberleşecektir.



Şekil 2: Raspberry Pi’nin PIR ile bağlantısı ve IP kamera.

Bu bölüm ve bir sonraki bölümdeki devre çizimleri [1]’de bulunan program aracılığıyla yapılmıştır.

-O-

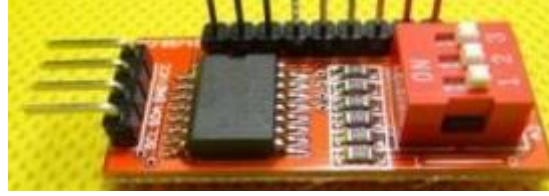
III. Kurulan Devre Detayları

Şekil 3'te Raspberry Pi Model 2'nin pin haritası gösterilmiştir. Raspberry Pi Model 2, farklı gömülü sistemlerle entegrasyon için farklı pinler barındırmaktadır. Bu pinler dış çevre cihazlarla bağlantılar sağlar. Genel amaçlı, I2C , SPI ve UART bunlardan bazılarıdır. Ayrıca farklı 'RSXXX' protokollerini kullanan bileşenler (ekran bağlantısı gibi) de Raspberry Pi 2 üzerinde bulunmaktadır. Bu bağlantılardan bazıları Raspberry Pi'nin konfigürasyon ayarlarından açılıp kapatılabilir. Bu da bazı pinlerin çoklu işlev sunmasına yaramaktadır. Örneğin kodlarken kullandığımız 'SSH' bağlantısı 'UART' pinlerinde sinyal değişimlerine yol açmaktaydı. GPIO kullanımını ortadan kaldırdı.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	Yellow	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

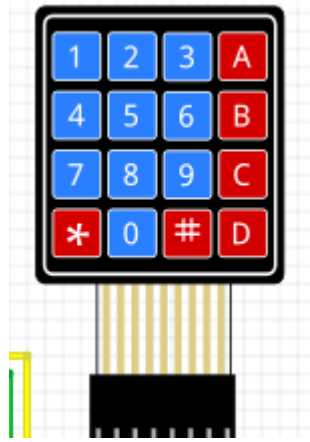
Şekil 3: Raspberry Pi Model 2'nin pin haritası.

hassasiyeti ayarlar. Sağdaki ise sinyalin gönderim süresini gecikmeyi ayarlar. IR sensörünün VCC gerilim ucu Raspberry Pi'nin Vcc'sine (5V) bağlanır. Toprak uçlarından birine ise hareket sensörünün (PIR) toprak gerilimi bağlanır. Sensörden veri alımı Raspberry Pi'nin GPIO14 ucu giriş moduna alınarak alınır.



Şekil 6: i2c bağlantısı için kullanılan çip.

Şekil 6'da i2c bağlantısı için kullanılan ve keypad'e bağlanan çip gösterilmiştir. bu çipe sda ve belli bir baud rate'e ayarlanmış scl ile sinyal gönderilebilir. Bu sinyal içteki PCF8274 çipi sayesinde uçlara iletilir. Bu çip doğal olarak 0x20 adresinde sahiptir ancak sağda bulunan pull-up ya da pull-down sinyallerle bu adres değiştirilebilir. Hepsini açılarak 0x27(hexadecimal) adresine adres çekilebilir, bu da farklı bağlantıların da i2c ile bir cihaza bağlanmasını mümkün kılar. 5 Volt gerilim en üst ucuna toprak üstten ikinci ucuna sda ve scl sırasıyla üç ve dördüncü uçlara bağlanır.



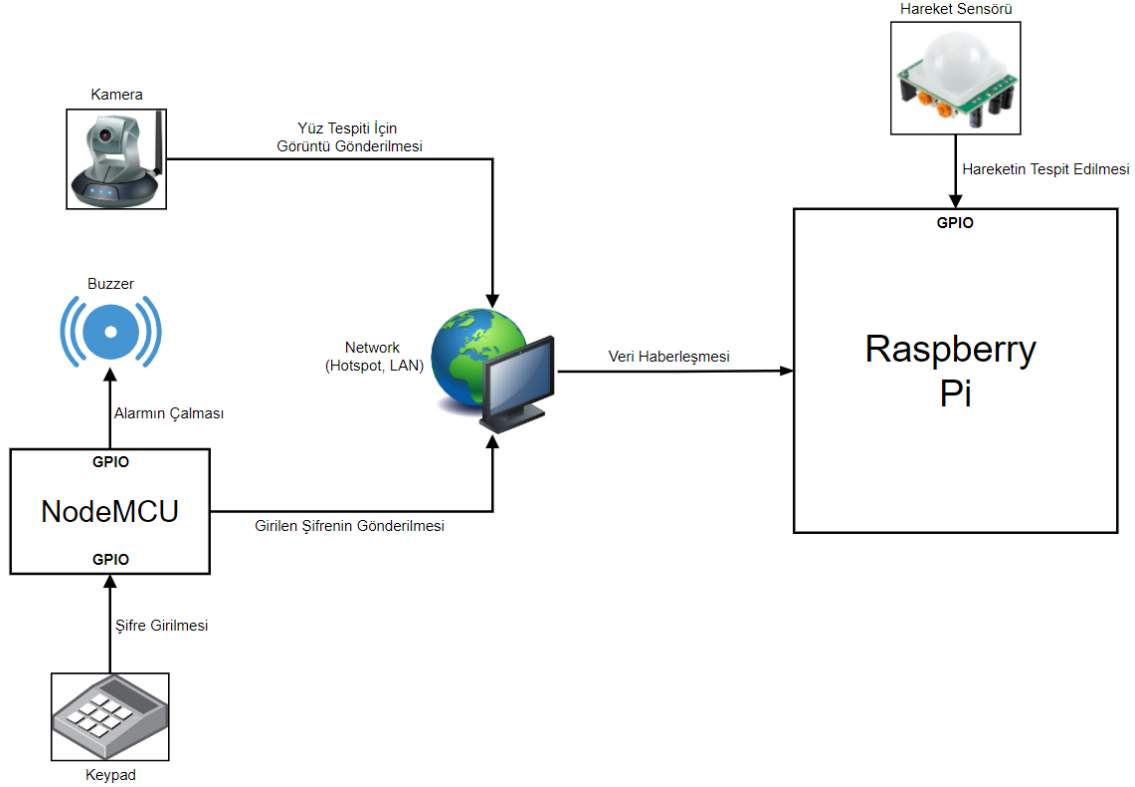
Şekil 7: Keypad modülü.

Keypad modülü karşılıklı pull-up ya da pull-down anahtarlar içerir. Bu sayede gönderilen 0xF0 ve 0x0F sinyalleri ile dikey ya da yatay sütun kapatılmış olur. Bu sayede basılan tuş iki sinyalin kullanımı ile bulunur.

-o-

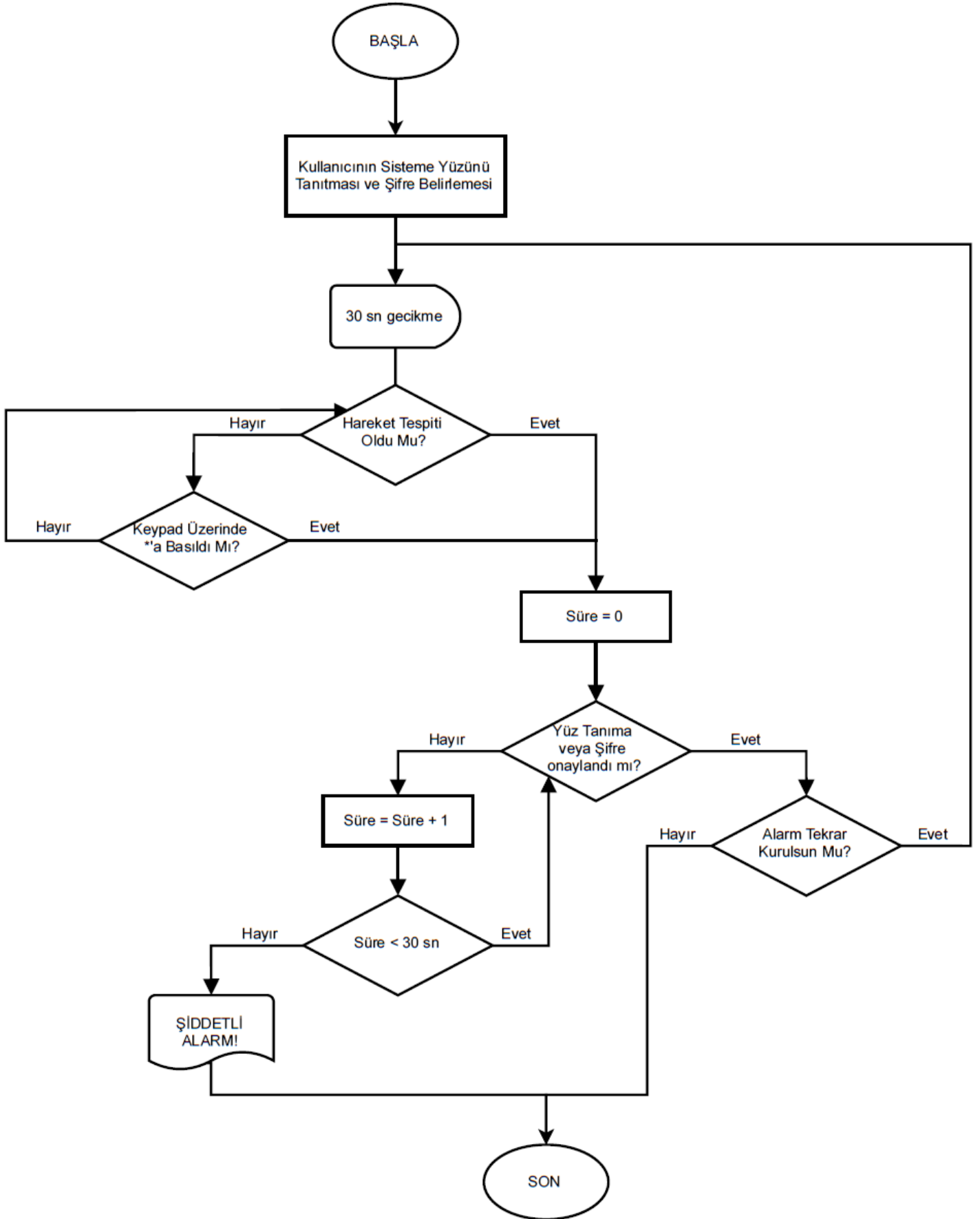
IV. Yazılım Tasarımı

Projede alarm sistemi geliştirilirken yazılım tasarımı kısmı için Şekil 8'deki sistemin genel blok diyagramı göz önüne alınmıştır. Bu blok diyagramı göz önüne alınarak sistemin ana akış diyagramı (flow chart) da Şekil 9'daki gibi oluşturulmuştur.



Şekil 8: Sistemin genel blok diyagramı.

Sistemin ana akış diyagramı Şekil 9'daki gibi çizildikten sonra Python kullanılarak gerekli yazılımlar geliştirilmiştir. Ayrıca NodeMCU ile Raspberry Pi arasındaki iletişim (Keypad ile veri iletişimi) için Arduino da kodlanmıştır.



Şekil 9: Sistemin ana akış diyagramı

```

class FaceRecognition(Thread):
    def __init__(self): pass
    def run(self): pass
    def reset_recognizer(self): pass
    def start_recognizer(self): pass
    def get_recognition_status(self): pass
    def get_recognized_people_name(self): pass

```

Şekil 10: Yüz tanıma için yazılan class.

Şekil 10’da gösterilen class’ın içinde bulunan fonksiyonlar yazılarak sistemdeki yüz tanıma işlemleri yerine getirilmektedir. Fonksiyonlar aşağıda açıklanmıştır:

- run(): Yüz tanıma flag’i (yüz tanıma algoritması devredeyken 1, devrede değilken 0 olan flag) set edildiğinde yüz tanıma işlemini başlatacaktır. Devamlı bir şekilde yüz tanıma flag’i kontrol edilir, nesne oluşturulduktan sonra start metodu ile ayrı bir thread olarak başlatılır.
- reset_recognizer(): Yüz bulunma flag’ini (kayıtlı yüzü bulursa 1, bulmazsa 0 olan flag) sıfırlar ve algoritma hazır hale gelir.
- start_recognizer(): Yüz tanıma flag’i set edilir.
- get_recognition_status(): Yüz bulunma flag’ini döndürür.
- get_recognized_people_name(): Tanınan kişinin ismini döndürür, tanınan kişi yoksa bilinmeyen döndürür.

```

class PassChecker(Thread):
    def __init__(self): pass
    def run(self): pass
    def start_pass_check(self): pass
    def set_host_and_port(self, host_set,port_set): pass
    def set_password(self, password_set): pass
    def get_pass_status(self): pass
    def get_activation_status(self): pass
    def reset_activation_status(self): pass
    def send_status(self,key): pass
    def get_check_pass(self): pass

```

Şekil 11: Şifre kontrol etmek için yazılan class.

Şekil 11’de gösterilen class’ın içinde bulunan fonksiyonlar yazılarak sistemdeki şifre kontrol etme işlemleri yerine getirilmektedir. Fonksiyonlar aşağıda açıklanmıştır:

- run(): Thread olarak çalışır, socket üzerinden Keypad’ten gelen verilerin okunduğu ve eğer start_pass_check() ile şifre kontrol flag’i aktif edilirse girilen karakterleri kaydederek şifre kontrolünü gerçekleştirir. Eğer şifre doğruysa şifre durum flag’ini set eder.

- start_pass_check(): Şifre kontrol flag'ini set eder ve şifre kontrol işlemi başlar.
- set_host_and_port(): Socketin IP ve portunu set eder.
- set_password(): Kontrol edilecek şifreyi set eder.
- get_pass_status(): Şifre durum flag'ini döndürür.
- get_activation_status(): Keypad'i devreye sokan flag'i döndürür.
- reset_activation_status(): Keypad'i devreye sokan flag'i sıfırlar.
- send_status(): Keypad'e komut gönderir, ışık aç-kapa, buzzer.
- get_check_pass(): Şifre kontrol flag'ini döndürür.

```
class GetMotion(Thread):
    def __init__(self): pass
    def run(self): pass
    def set_running_status(self): pass
    def get_motion_status(self): pass
    def stop_running(self): pass
```

Şekil 12: Hareket tespitinin yapılması için yazılan class.

Şekil 12'de gösterilen class'ın içinde bulunan fonksiyonlar yazılarak sistemdeki hareket tespit etme işlemleri yerine getirilmektedir. Fonksiyonlar aşağıda açıklanmıştır:

- run(): Thread olarak çalışır. Hareket algılama flag'i devreye sokulduğunda hareket tespit etme işlemi yapar ve hareket flag'ini set eder ve Raspberry Pi üzerindeki gerekli ışıkları yakar.
- set_running_status(): Hareket algılama flag'ini set eder.
- get_motion_status(): Hareket flag'ini döndürür.
- stop_running(): Hareket algılamayı durdurur ve ışıkları söndürür.

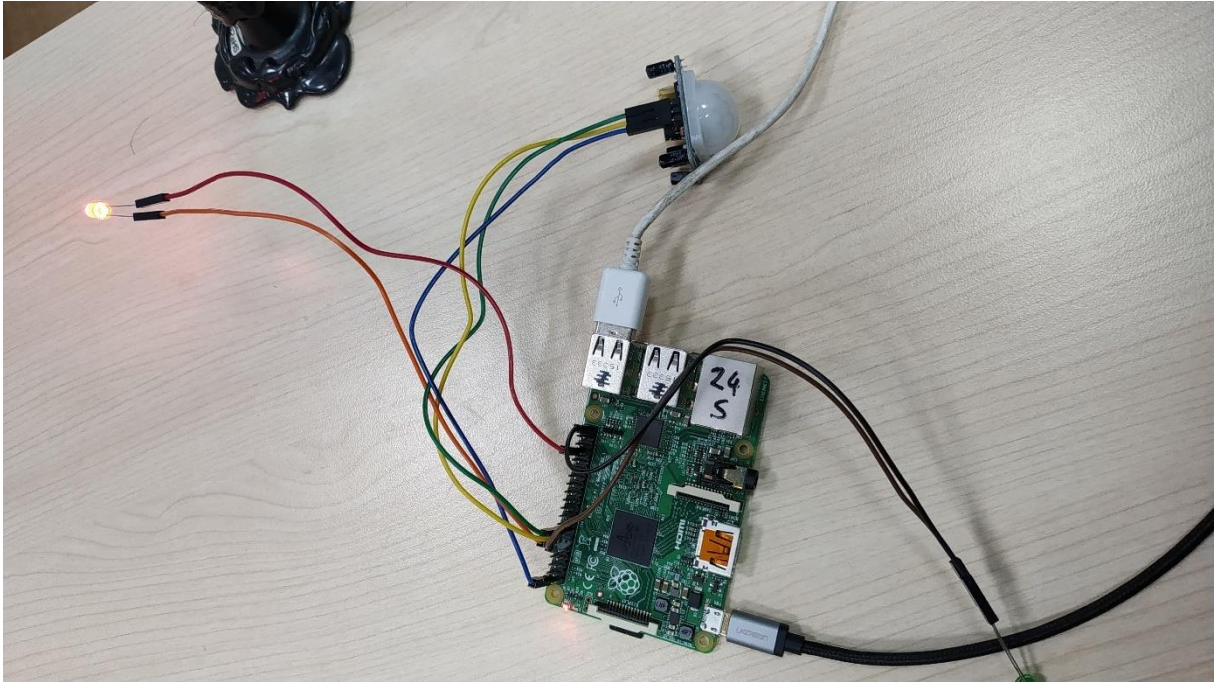
Yukarıda bulunan class'lar ve içindeki fonksiyonlar kullanılarak Şekil 9'daki akış diyagramı çerçevesinde Python ile ana program yazılmıştır. Bu program uzun olduğundan rapora koyulmamıştır. Tüm kodlar, video ve raporla birlikte teslim edilecektir.

-O-

V. Sonuçlar, Demo Detayları ve Sunum Linki

Projede sonuç olarak istenen özelliklere sahip bir gömülü sistem geliştirilmiştir. Bu gömülü sistemde;

- Kullanıcının yüzü ve bir şifre sistem kurulmaya başlanmadan önce sisteme tanıtılmaktadır.
- Sistem kurulduktan sonra ev içinde bulunan hareket sensörü ile hareket tespit edildiğinde hareketi yapan kişiden alarmın çalmaması için bir şifre girmesi ya da yüzünü göstermesi beklenmektedir.
- Kullanıcı evdeyse kameraya yüzünü göstererek (Yüz Tanıma) ya da keypad ile daha önceden belirlediği şifreyi girerek hareketi yapanın kendisi olduğunu sisteme bildirebilmektedir. Böylece alarm şiddetlenmeden susturulmaktadır.
- Eğer hareketi yapan kişi belli bir süre içinde yukarıdakilerden birini yapmazsa alarm şiddetli bir şekilde çalmaya başlamaktadır. (Evde tanınmayan biri var uyarısı!)



Şekil 13: Raspberry Pi ve ona bağlı devre elemanları.

Şekil 13'te Raspberry Pi ve ona bağlı olan hareket sensörü ve ledler gösterilmiştir. Hareket sensörü eğer bir hareket tespit ederse bu Raspberry Pi'ye bildirilmekte ve gerekli işlemler yapılmaktadır. Raspberry Pi'ye bağlı olan ledler ise gerekli olduğuna yanmakta ya da sönmektedir.

VI. Referanslar

- [1] Fritzing. <https://fritzing.org/>.

[illegible]