**T.C.**

**DOKUZ EYLÜL ÜNİVERSİTESİ**

**İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ**

**YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ**

**DÖNEM PROJESİ**

**YBS 2013:**

**BİLGİSAYAR DONANIMI**

**YANGIN ALARMI**

**2020469058**

**Kutlay Kenan TİMİN**

**Öğretim Üyesi**

**Prof.Dr. Vahap TECİM**

**İZMİR - 2023**

Raspberry Pi kullanım alanı oldukça geniş bir alet olduğu için günlük hayatımıza kolaylıkla giriş yapabilir. Bu yüzden optimum efor ve çalışma ile istediğimiz projeleri yapabilmekteyiz.

Benim Raspberry Pi ile projem yangın alarmı yapmak. Lakin sadece ışık verip ses çıkarmaktan ziyade yangın sırasında evinde bulunamayan kişilere e-mail göndererek haber vermesini sağlayacağım.

Öncelikle Raspberry Pi ile çalışmaya başlayabilmek için Raspberry OS’i elimizde bulunan Raspberry Pi’a kurmamız gerekiyor. Bu sırada kurulum aşamasında VNC ayarını aktive edip bilgisayarımın hotspot ismi ve şifresi ile eşleştirdim. Böylece Raspberry Pi boot olurken bilgisayarımın veya aynı bilgilerde olan telefon hotspotu ile internete bağlanabilecek.

Ardından breadboard ve Raspberry Pi’ımızın GPIO pinleri ile çalışmaya başlayabiliriz. Öncelikle breadboard’a ateş dedektörünü bağladım. Bunun için VCC pini ile 2. GPIO pinini dişi dişi kablo ile bağlayarak ateş dedektörüne güç sağlıyoruz. Ardından GND pini ile 6. GPIO pinini dişini topraklamak için bağlıyoruz. Son olarak da D0 pinini dişi dişi kablo ile 22. GPIO pinine haberleşmeyi sağlamak için bağlıyoruz.

İkinci adımda ise ledimizin uzun bacağını yani negatif kısmını breadboard’da negatif kutba bağlıyoruz ve direncin ile pozitif bacağına bağlantı yapıyoruz. Bunun amacı düşük voltajlı ledimizin 5 voltluk akım ile patlamamasıdır. Yine breadbord’un negatif kısmından bir erkek dişi kablo ile 15. GPIO pinine topraklıyoruz. Direncin hemen yanındaki deliğe bir erkek dişi kablo ile 8. GPIO ile Uart0 TX bağlantısı yaparak güç akışını sağlıyoruz.

Üçüncü adımda buzzer’ımız ile çalışacağız. Buzzer’ımın pozitif bacağını erkek dişi kablo ile 7. GPIO ile GPIO 4’e bağlıyoruz ve negatif bacağı ise 9. GPIO pinine bağlayarak, erkek dişi kablo ile, topraklıyoruz.

Dördüncü adımda ise yangın alarmı için bir mail adresi açıyoruz. Ben GMail’i tercih ettim. Raspberry Pi’da Python dili ile yazacağımız uygulamada bir e-mail adresimizin çalışabilmesi için bir App Password gerekiyor. Bunun için Two – Step Verification oluşturduktan sonra App Password generate ettik.

Bundan sonrasında kodlamamızı yapıyoruz Python ile. Raspberry Pi ile çalışabilmemiz için gerekli kütüphaneler: RPi.GPIO, time, smtplib, email.massage. Ekstra olarak GMail’in Python email.message kütüphanesinde çalışmalar yarattığı için false error yaratmaktadır. Bundan outputmuzun düzgün okunabilmesi için kurtulmak adına warnings kütüp hanesini çağırarak bu false error’lardan kurtulduk.

Kodumuzu yazarken öncelikle mail() isimli bir fonksiyon tanımlayarak atacağımız e-mail’in özellikleri tanımladık. Buna atacağımız e-mail’in kimden geldiği, kime gideceği, başlığı ve içeriği olarak kodladık. Sonrasında Gmail ile birlikte çalışabilmek için gerekli TTLS ayarlarını yaptık ve gönderici e-mail adresine giriş, e-mail’i gönderiş ve çıkış işlemlerini gerçekleştirdik.

Bu kısımdan itibaren artık GPIO pinleri üzerinde çalışmaya başlıyoruz. Öncelilkle kodumuzun GPIO üzerinden çalıştığımızı anlayabilmesi için GPIO.setmode(GPIO.BOARD) kodu ile kendisine GPIO’larımızı tanıttık ve ardından haberleşmeyi sağlayan pinlerimizi birer variable’a atadık. Bunun sebebi ileride kodumuz üzerinde çalışırken sürekli olarak rakamları aramak bizlere problem yaratırdı. Örnek olarak: hangi pin neye bağlıydı veya hangi pin rakamlarını değiştirmek gerekirdi verilebilir. Bu yüzden ateş dedektkörü için Pin, ledimiz için led ve buzzer’ımız içinse buzzer değişkinlerine bağlı oldukları GPIO pinlerini atadık.

Ardından yangın alarmımızın çalışabilmesi için fonksiyon tanımlaya başladık. Bunun için blink() fonksiyonunu tanımladım. Böylece ledimiz ve buzzerımız yarım saniyede bir yanıp sönerek bize uyarı verecek. Bunu ayarlamak için time.sleep(0.5) kodunu kullandık.

Sonraki tanımladığımız fonksiyon ise alert(). Bu fonksiyonun amacı şu ana kadar tanımladığımız iki fonksiyounu bir araya getirerek birlikte çalıştırmak oldu. Ayrıca ekstra bir print komutu ile output alıyoruz.

Son fonksiyonumuz olan loop() ise GPIO.add\_event\_detect kodu ile alev sensörümüzün çalıştığını algılayıp yarattığımız alert() fonksiyonuna callback yaparak kodumuzun çalışmasını sağlayacak. Böylece yangın alarmı kodumuzun sonuna geldik.

Ayrıca kodum üzerinde olmasa da Raspberry Pi üzerinde bir trick ile kodumun headless çalışmasını sağladım. Sonuçta yangın alarmlarını çalıştırmak için sürekli olarak bir monitör veya bilgisayara erişimimiz olmayacak. Terminalde sudo nano /etc/rc.local dosyasını açıp en son satıra kodumuzun path’ini yazıyoruz ve & işareti koyuyoruz. Böylece kodumuz Raspberry OS ile birlikte boot oluyor. & işaretinin sebebi ise while True döngüsü olduğu için kodumuzun döngüden çıkıp OS’imizi açmasını sağlıyor.

Gördüğünüz üzere yaptığım yangın alarmı başarılı bir şekilde çalışmaktadır. Özellikle de headless olarak çalışması sayesinde ufak bir yere monte edilerek kolayca hayatımıza girişini yapabilir. Sonraki sayfalarda ise kodumu görebilirsiniz.

import RPi.GPIO as GPIO

import time

import smtplib

from email.message import EmailMessage

import warnings

warnings.filterwarnings("ignore")

def mail():

from\_email\_addr = "xyangin.alarmi@gmail.com"

from\_email\_pass = "smoqyrrwlftnnmcp"

to\_email\_addr = "kutlaykenantimin@yandex.com"

msg = EmailMessage()

body = "Yangın var! Yangın var! Ben Yanıyorum. Yetişine Dostlar, Tutuşuyorum!"

msg.set\_content(body)

msg['From'] = from\_email\_addr

msg['To'] = to\_email\_addr

msg['Subject'] = "Yangın var!"

server = smtplib.SMTP('smtp.gmail.com', 587)

server.starttls()

server.login(from\_email\_addr, from\_email\_pass)

server.send\_message(msg)

server.quit()

Pin = 22

buzzer = 7

led = 8

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(Pin,GPIO.IN,pull\_up\_down = GPIO.PUD\_UP)

GPIO.setup(led,GPIO.OUT)

GPIO.setup(buzzer,GPIO.OUT)

def blink():

GPIO.output(buzzer,GPIO.HIGH)

GPIO.output(led,GPIO.HIGH)

time.sleep(0.5)

GPIO.output(buzzer,GPIO.LOW)

GPIO.output(led,GPIO.LOW)

time.sleep(0.5)

def alert(x):

print("Yangın var! Yangın var! Ben Yanıyorum. Yetişine Dostlar, Tutuşuyorum!")

blink()

mail()

def loop():

GPIO.add\_event\_detect(Pin,GPIO.FALLING,callback=alert)

while True:

pass

try:

loop()

except KeyboardInterrupt:

GPIO.cleanup()