MARMARA ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

MEKATRONİK MÜHENDİSLİĞİ

GÖMÜLÜ SİSTEMLER VE UYGULAMALARI DERSİ FİNAL RAPORU

Proje Adı:

BENDEN KAÇAMAZSIN!

Proje Sahipleri:

Emine Meltem HAMZAOĞLU 170215002

Tugay Kaan ARSLAN 171214003

BENDEN KAÇAMAZSIN!

Raspberry Pi kullanarak Parmak İzi Sensörü İle İşçilerin Bilgilerinin Alınması ve Renk Sensörü İle İş Önlüklerinin tespit edilmesi

**GİRİŞ**

Bu proje Raspberry Pi 3 B+ modeli kullanılarak ZFM60 Parmak İzi Okuyucu sensör ile parmak izi bilgileri alınıyor ve kaydediliyor. TCS3200 Renk Sensörü ile İş Önlüğünün rengi algılanıyor. Fabrikalarda İş Önlüğü giyme zorunluluğu vardır fakat bazı işçiler sıcak, kirli vs. gibi nedenlerden dolayı iş önlüklerini giymemektedir. Tasarladığımız sistem sayesinde parmak izi sensörüyle işçilerin parmak izi bilgileri alınarak kaydedilmektedir. Daha sonra Renk Sensörü sayesinde önlüklerin renkleri algılanıyor. Bu sistem İş başlangıç saati, öğle arası ve paydosta işleme alınırsa işçilerin önlük giyme zorunluluğuna uyup uymadıkları tespit edilebilir.

Projemizde Raspberry pi 3 B+ modeli kullanılmış olur Raspbian Jessie Os kuruludur.

Parmak izleri /home/pi/documents/parmakiziokuma klasörüne kaydedilmektedir.

**GEREKLİ DONANIMLAR**

1 adet Raspberry Pi 3 B+ (Raspbian Jessie Os kurulu)

1 adet TCS3200 Renk Sensörü

1 adet ZFM60 Parmak İzi Okuyucu Sensör

1 adet Breadboard

Jumper Kablolar

**GEREKLİ YAZILIM BİLEŞENLERİ**

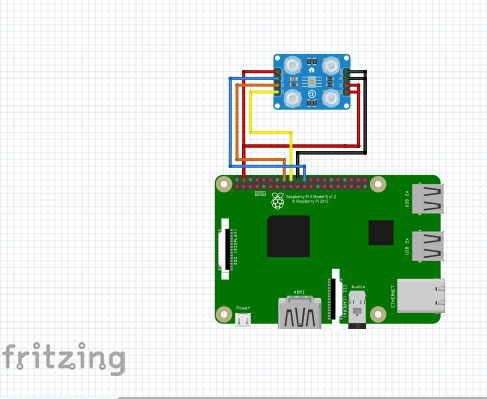
Raspbian Jessie OS

Python 3.6

**KULLANILAN BİLEŞENLERİN ÖZELLİKLERİ**

1. ZFM60 Parmak İzi Okuyucu Sensör <https://www.direnc.net/parmak-izi-okuyucu>
2. TCS3200 Renk Sensörü <https://www.robotistan.com/tcs3200-renk-sensoru-karti>
3. Breadboard
4. Jumper Kablolar

**ŞEMATİK ÇİZİM**



Renksensörü.fzz

**RENK SENSÖRÜ PYTHON KODU**

import RPi.GPIO as GPIO

import time

s2 = 23

s3 = 24

signal = 25

NUM\_CYCLES = 10

def setup():

GPIO.setmode(GPIO.BCM)

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

GPIO.setup(s2,GPIO.OUT)

GPIO.setup(s3,GPIO.OUT)

print("\n")

def loop():

temp = 1

while(1):

GPIO.output(s2,GPIO.LOW)

GPIO.output(s3,GPIO.LOW)

time.sleep(0.3)

start = time.time()

for impulse\_count in range(NUM\_CYCLES):

GPIO.wait\_for\_edge(signal, GPIO.FALLING)

duration = time.time() - start

red = NUM\_CYCLES / duration

GPIO.output(s2,GPIO.LOW)

GPIO.output(s3,GPIO.HIGH)

time.sleep(0.3)

start = time.time()

for impulse\_count in range(NUM\_CYCLES):

GPIO.wait\_for\_edge(signal, GPIO.FALLING)

duration = time.time() - start

blue = NUM\_CYCLES / duration

GPIO.output(s2,GPIO.HIGH)

GPIO.output(s3,GPIO.HIGH)

time.sleep(0.3)

start = time.time()

for impulse\_count in range(NUM\_CYCLES):

GPIO.wait\_for\_edge(signal, GPIO.FALLING)

duration = time.time() - start

green = NUM\_CYCLES / duration

if red>green and red>blue and red>12000:

print("kırmızı")

temp=1

elif red<green and blue<green and green>12000:

print("yeşil")

temp=1

elif green<blue and red<blue and blue>12000:

print("mavi")

temp=1

elif red>10000 and green>10000 and blue>10000 and temp==1:

print("objeyi yaklaştırın!")

temp=0

def endprogram():

GPIO.cleanup()

if \_\_name\_\_=='\_\_main\_\_':

setup()

try:

loop()

except KeyboardInterrupt:

endprogram()

**RENK SENSÖRÜ HTML KODU**

<p>import RPi.GPIO as GPIO</p>  
  
<p>import time</p>  
  
<p>s2 = 23</p>  
  
<p>s3 = 24</p>  
  
<p>signal = 25</p>  
  
<p>NUM\_CYCLES = 10</p>  
  
<p>def setup():</p>  
  
<p>GPIO.setmode(GPIO.BCM)</p>  
  
<p>GPIO.setup(signal,GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)</p>  
  
<p>GPIO.setup(s2,GPIO.OUT)</p>  
  
<p>GPIO.setup(s3,GPIO.OUT)</p>  
  
<p>print("\n")</p>  
  
<p>def loop():</p>  
  
<p>temp = 1</p>  
  
<p>while(1): </p>  
  
<p>GPIO.output(s2,GPIO.LOW)</p>  
  
<p>GPIO.output(s3,GPIO.LOW)</p>  
  
<p>time.sleep(0.3)</p>  
  
<p>start = time.time()</p>  
  
<p>for impulse\_count in range(NUM\_CYCLES):</p>  
  
<p>GPIO.wait\_for\_edge(signal, GPIO.FALLING)</p>  
  
<p>duration = time.time() - start </p>  
  
<p>red = NUM\_CYCLES / duration </p>  
  
<p>GPIO.output(s2,GPIO.LOW)</p>  
  
<p>GPIO.output(s3,GPIO.HIGH)</p>  
  
<p>time.sleep(0.3)</p>  
  
<p>start = time.time()</p>  
  
<p>for impulse\_count in range(NUM\_CYCLES):</p>  
  
<p>GPIO.wait\_for\_edge(signal, GPIO.FALLING)</p>  
  
<p>duration = time.time() - start</p>  
  
<p>blue = NUM\_CYCLES / duration</p>  
  
<p>GPIO.output(s2,GPIO.HIGH)</p>  
  
<p>GPIO.output(s3,GPIO.HIGH)</p>  
  
<p>time.sleep(0.3)</p>  
  
<p>start = time.time()</p>  
  
<p>for impulse\_count in range(NUM\_CYCLES):</p>  
  
<p>GPIO.wait\_for\_edge(signal, GPIO.FALLING)</p>  
  
<p>duration = time.time() - start</p>  
  
<p>green = NUM\_CYCLES / duration</p>  
  
<p>if red&gt;green and red&gt;blue and red&gt;12000:</p>  
  
<p>print("kırmızı")</p>  
  
<p>temp=1</p>  
  
<p>elif red&lt;green and blue&lt;green and green&gt;12000:</p>  
  
<p>print("yeşil")</p>  
  
<p>temp=1</p>  
  
<p>elif green&lt;blue and red&lt;blue and blue&gt;12000:</p>  
  
<p>print("mavi")</p>  
  
<p>temp=1</p>  
  
<p>elif red&gt;10000 and green&gt;10000 and blue&gt;10000 and temp==1:</p>  
  
<p>print("objeyi yaklaştırın!")</p>  
  
<p>temp=0</p>  
  
<p>def endprogram():</p>  
  
<p>GPIO.cleanup()</p>  
  
<p>if \_\_name\_\_=='\_\_main\_\_':</p>  
  
<p>setup()</p>  
  
<p>try:</p>  
  
<p>loop()</p>  
  
<p>except KeyboardInterrupt:</p>  
  
<p> endprogram()</p>

**PARMAK İZİ OKUYUCU PYHTON KODU**

from pyfingerprint.pyfingerprint import PyFingerprint

# Set fingerprint sensor to ttyUSB0 with buadrate 57600

fingerprintSensor = PyFingerprint(

'/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

# Flag for checking sensor available

fingerprintSensorStateFlag = False

i=0

while True:

def verify():

# Verify fingerprintsensor

global fingerprintSensorStateFlag

try:

if(fingerprintSensor.verifyPassword() is False):

print('Parmak izi okuyucu hatası!')

fingerprintSensorStateFlag = True

print('Parmakizi okuyucu kullanılabilir!')

except Exception as e:

print('Birşeyler Ters Gitti 1: ' + str(e))

def scan():

if(fingerprintSensorStateFlag):

try:

print('Parmak bekleniyor...')

# wait for finger

while(fingerprintSensor.readImage() is False):

pass

# save fingerprint image

print('Parmakizi kaydediliyor...')

fingerprintSensor.downloadImage(

'/home/pi/Documents/parmakiziokuma/parmakizi{}\_img.bmp'.format(i))

print('Parmakizi kaydedildi!')

except Exception as e:

print('Birşeyler ters gitti : ' + str(e))

i+=1

def main():

verify()

scan()

if \_\_name\_\_ == '\_\_main\_\_':

main()

**PARMAK İZİ OKUYUCU HTML KODU**

<p>from pyfingerprint.pyfingerprint import PyFingerprint</p>  
  
<p># Set fingerprint sensor to ttyUSB0 with buadrate 57600</p>  
  
<p>fingerprintSensor = PyFingerprint(</p>  
  
<p>'/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)</p>  
  
<p># Flag for checking sensor available</p>  
  
<p>fingerprintSensorStateFlag = False</p>  
  
<p>i=0</p>  
  
<p>while True:</p>  
  
<p>def verify():</p>  
  
<p># Verify fingerprintsensor</p>  
  
<p>global fingerprintSensorStateFlag</p>  
  
<p>try:</p>  
  
<p>if(fingerprintSensor.verifyPassword() is False):</p>  
  
<p>print('Parmak izi okuyucu hatası!')</p>  
  
<p>fingerprintSensorStateFlag = True</p>  
  
<p>print('Parmakizi okuyucu kullanılabilir!')</p>  
  
<p>except Exception as e:</p>  
  
<p>print('Birşeyler Ters Gitti 1: ' + str(e))</p>  
  
<p>def scan():</p>  
  
<p>if(fingerprintSensorStateFlag):</p>  
  
<p>try:</p>  
  
<p>print('Parmak bekleniyor...')</p>  
  
<p># wait for finger</p>  
  
<p>while(fingerprintSensor.readImage() is False):</p>  
  
<p>pass</p>  
  
<p># save fingerprint image</p>  
  
<p>print('Parmakizi kaydediliyor...')</p>  
  
<p>fingerprintSensor.downloadImage(</p>  
  
<p>'/home/pi/Documents/parmakiziokuma/parmakizi{}\_img.bmp'.format(i))</p>  
  
<p>print('Parmakizi kaydedildi!')</p>  
  
<p>except Exception as e:</p>  
  
<p>print('Birşeyler ters gitti : ' + str(e))</p>  
  
<p>i+=1</p>  
  
<p>def main():</p>  
  
<p>verify()</p>  
  
<p>scan()</p>  
  
<p>if \_\_name\_\_ == '\_\_main\_\_':</p>  
  
<p>main()</p>

**RESİMLER**