Botezatu Matei

botezatumatei@yahoo.com

Carp Adrian-Ionuț

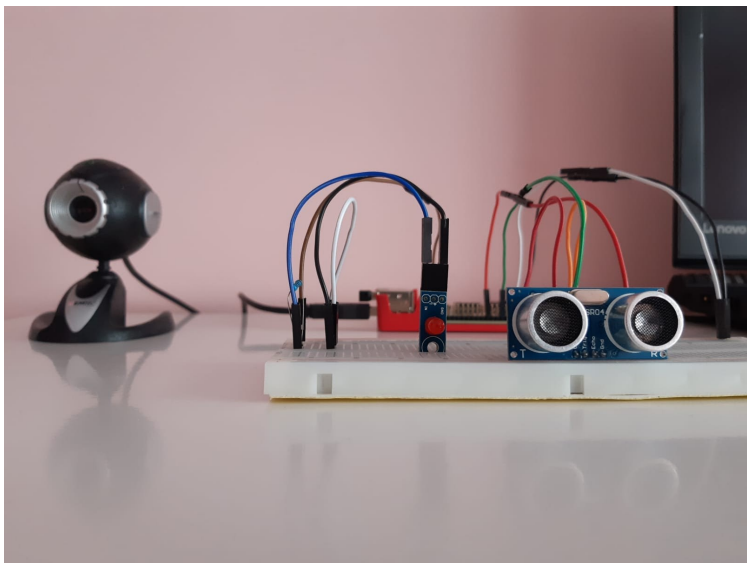
adrian_carp@ymail.com

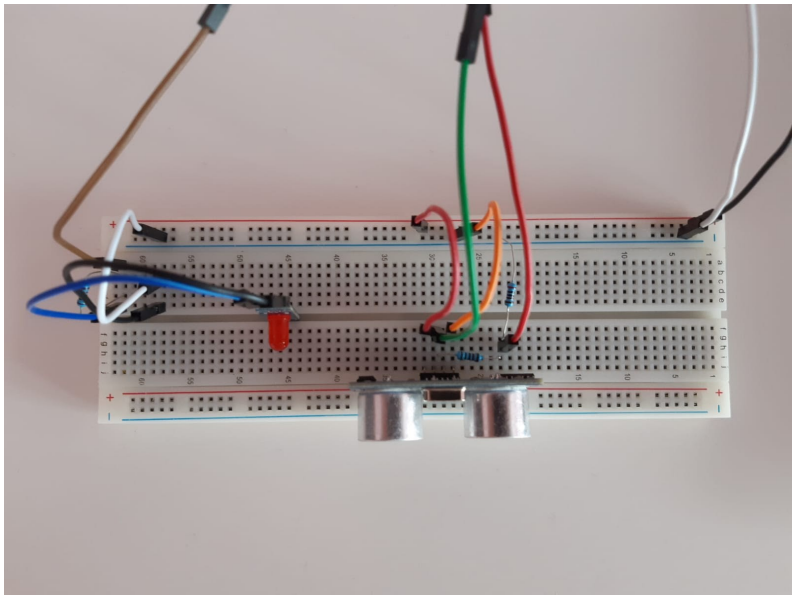
Țuțu Sebastian-Iulian

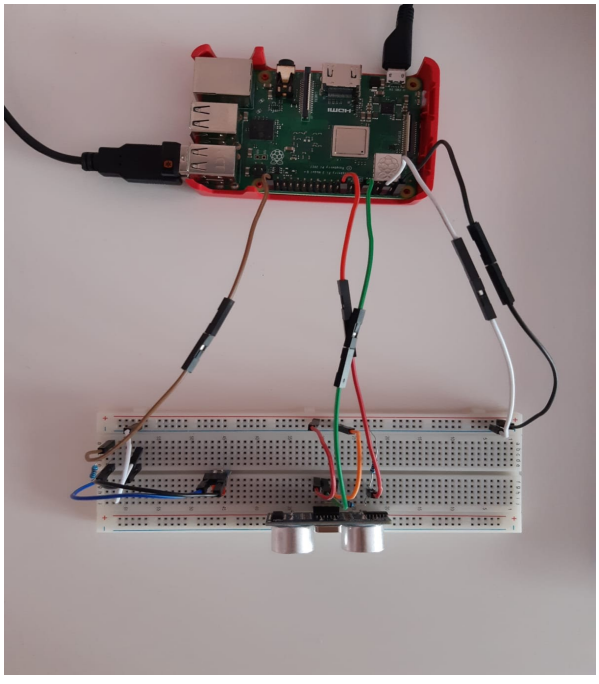
sebi_tutu@yahoo.ro

Numele proiectului

-Sistem de detectare a mișcării-







Scopul acestui proiect este de a realiza un sistem de supraveghere prin care se detectează mișcarea. Dacă un obiect/persoană se află la o distanță minimă setată în program de senzor, atunci se va activa camera WebCam și va efectua o poză pe care ulterior o va trimite pe mail.

Componente hardware:

- **Raspberry pi model 3 b+**
- **Senzor distance HC-SR04**
- **WebCam EMTEC eksvw100v**
- **Fire mama-mama x5**
- **Fire tata-tata x10**
- **Breadboard**
- **Led rosu RBFN-EU_COMP-LED-05**
- **Rezistenta 2k ohm x1**
- **Rezistenta 1k ohm x2**

Aplicație: Raspberry NOOBS

Cum funcționează proiectul nostru?

Vom folosi 1 led care dacă senzorul nu depistează niciun obiect/persoană va rămâne stins. La detecția unui obstacol camera WebCam se va declanșa și va efectua o poză care ulterior va fi trimisă pe mail. După ce mail-ul este trimis, led-ul v-a rămâne aprins 5 secunde.

Cod:

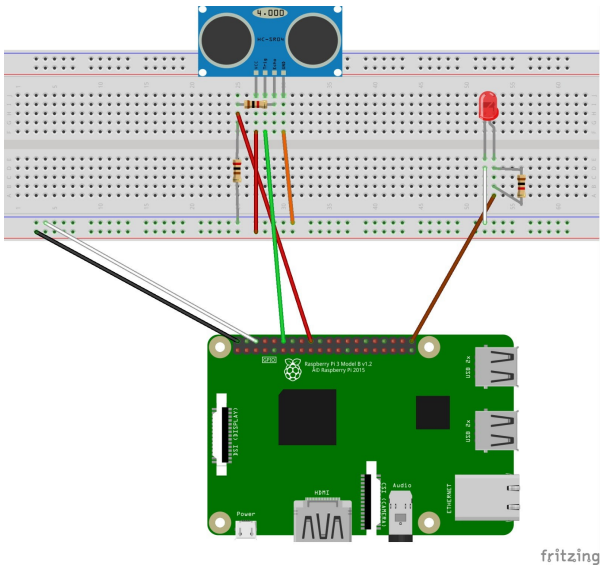
```
import RPi.GPIO as GPIO
import smtplib
import time
import signal
import sys
import os

from email.mime.multipart import MIMEMultipart
```

```
from email.mime.text import MIMEText
from email.mime.image import MIMEImage

# use Raspberry Pi board pin numbers
GPIO.setmode(GPIO.BCM)

# set GPIO Pins
pinTrigger = 18
pinEcho = 24
pinled = 21

#Email Variables
SMTP_SERVER = 'smtp.gmail.com' #Email Server (don't change!)
SMTP_PORT = 587 #Server Port (don't change!)
GMAIL_USERNAME = 'email-ul meu' #change this to match your gmail account
GMAIL_PASSWORD = 'parola'  #change this to match your gmail password
class Emailer:
    def sendmail(self, recipient, subject, content, image):
        #Create Headers
        emailData = MIMEMultipart()
        emailData['Subject'] = subject
        emailData['To'] = recipient
        emailData['From'] = GMAIL_USERNAME
        #Attach our text data
        emailData.attach(MIMEText(content))
        #Create our Image Data from the defined image
        imageData = MIMEImage(open(image, 'rb').read(), 'jpg')
        imageData.add_header('Content-Disposition', 'attachment; filename="image.jpg"')
        emailData.attach(imageData)
        #Connect to Gmail Server
        session = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
        session.ehlo()
        session.starttls()
```

```
        session.ehlo()
        #Login to Gmail
        session.login(GMAIL_USERNAME, GMAIL_PASSWORD)
        #Send Email & Exit
        session.sendmail(GMAIL_USERNAME, recipient, emailData.as_string())
        session.quit

sender = Emailer()
sendTo = 'email-ul la care trimit'
emailSubject = "Alerta!!!"
emailContent = "S-a detectat miscare in zona supravegheata, aveti atasata imaginea camerei de
supraveghere"

def close(signal, frame):
        print("\n Supraveghere oprita")
        GPIO.cleanup()
        sys.exit(0)

signal.signal(signal.SIGINT, close)

# set GPIO input and output channels
GPIO.setup(pinTrigger, GPIO.OUT)
GPIO.setup(pinEcho, GPIO.IN)
GPIO.setwarnings(False)
GPIO.setup(pinled,GPIO.OUT)

try:
        while True:
                # set Trigger to HIGH
                GPIO.output(pinTrigger, True)
                # set Trigger after 0.01ms to LOW
                time.sleep(0.00001)
                GPIO.output(pinTrigger, False)
```

```
            startTime = time.time()
            stopTime = time.time()
            # save start time
            while 0 == GPIO.input(pinEcho):
                startTime = time.time()
            # save time of arrival
            while 1 == GPIO.input(pinEcho):
                stopTime = time.time()
            # time difference between start and arrival
            TimeElapsed = stopTime - startTime
            # multiply with the sonic speed (34300 cm/s)
            # and divide by 2, because there and back
            distance = (TimeElapsed * 34300) / 2
            image = '/home/pi/image.jpg'
            if(distance<10):
                os.system('fswebcam -r 320x240 --no-banner -S 3 --jpeg 50 --save imag$
                sender.sendmail(sendTo, emailSubject, emailContent, image)
                GPIO.output(pinled,GPIO.HIGH)
                time.sleep(5)
                GPIO.output(21,GPIO.LOW)
            print ("Distance: %.1f cm" % distance)
            time.sleep(1)
except KeyboardInterrupt:
    pass
GPIO.cleanup()
sys.exit(0)
```

Schematic:



Contribuția fiecărui membru:

Botezatu Matei- Implementare senzor distanță + documentație

Carp Adrian-Ionuț- Implementare led + documentație

Țuțu Sebastian-Iulian- Implementare circuite + implementare efectuare poză și trimitere pe mail

Video: https://youtu.be/cDqgEZDfl8M

Hackster.io:https://www.hackster.io/350725/motion-detection-system-8acdbc