

**Universitatea
Transilvania
din Brașov**

**FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR**

SISTEM DE SECURITATE

Proiect Sisteme Încorporate

Autor:

Zidaru Marian George

Purcariu Vlăduț Ionel

Calculatoare anul IV

Grupa 4LF781

Marian.zidaru@student.unitbv.ro

Vladut.purcariu@student.unitbv.ro

Contents

1. Rezumat	3
1.1 Descrierea proiectului	3
1.2 Descrierea implementarii.....	3
2. Introducere	3
2.1 Prezentarea temei.....	3
2.2 Motivatia practica pentru alegerea temei	3
3. Prezentarea platformei Hardware	4
3.1 Raspberry PI 4	4
3.2 Electronica folosita.....	5
3.2.1 Senzor de distanta SR04.....	5
3.2.2 Camera p5v04a sunny.....	6
4. Schema bloc a implementarii.....	7
5. Costurile proiectului.....	8
6. Prezentarea implementarii software	8
6.1 Limbajul Python	8
6.2 RPI-GPIO	8
6.3 FTPLIB.....	8
7. Performante	9
8. Concluzii	9
9. Dezvoltari ulterioare	9
10. Bibliografie	9
11. Anexa	10

1. Rezumat

1.1 Descrierea proiectului

Proiectul propus va indeplini rolul unei camere de securitate, care ,prin intermediul unui senzor de distanță(HC-SR04) , in momentul detectării unui obstacol, va activa camera care va realiza o fotografie, care va fi încărcat pe un server FTP configurat pe Raspberry Pi, iar cu ajutorul modului Wi-Fi de pe Raspberry Pi, putem vedea capturile realizate de cameră prin intermediul unui smartphone, laptop, etc.

Elementele componente ale proiectului sunt: Placuta Raspberry Pi 4B pe care configurăm server-ul FTP și folosim modulul Wi-Fi, camera p5v04a sunny, senzor de distanta HC-SR04.

1.2 Descrierea implementarii

Piese componente vor fi conectate la placa Raspberry Pi 4B. Se configurează serverul FTP prin intermediul plăcuței, remote , folosind aplicația VNC Viewer sau remote desktop, instalând pachetul vsftpd și urmând pașii de configurare de la adresa <https://linuxize.com/post/how-to-setup-ftp-server-with-vsftpd-on-raspberry-pi/>. Senzorul de distanță împreună cu camera vor fi alimentate de către placuță, iar în momentul în care senzorul de distanță întâlnește un obstacol pe raza sa, camera sunny se va activa și va realiza o fotografie live, fotografie care prin intermediul plăcuței va fi trimisă în folderul FTP configurat anterior. Utilizarea modulului Wi-Fi integrat de pe Raspberry Pi, prin accesarea într-un browser a adresei ip a plăcuței, putem vedea capturile realizate de cameră,

2. Introducere

2.1 Prezentarea temei

În acest proiect ne propunem crearea unui sistem de securitate, ușor de utilizat și destul de eficient pentru detectarea intrusului.

2.2 Motivatia practica pentru alegerea temei

Motivul principal pentru alegerea acestei teme este crearea unei lumi cât mai sigure. Nu este un lucru nou faptul că jafurile sunt din ce în ce mai multe în ziua de azi, lipsa banilor ducând la acest lucru, ajutat și de faptul că multe proprietăți nu sunt securizate corespunzător, permițându-le hoților să scape destul de ușor. Prin crearea acestui sistem de securitate, utilizatorul își poate securiza

bunurile foarte ușor, fiind un sistem efficient și ușor de folosit. Prin simplitatea utilizării acestui sistem, vrem să încurajăm cât mai multe persoane să înceapă să folosească un astfel de sistem.

3. Prezentarea platformei Hardware

3.1 Raspberry PI 4

Noua placă de dezvoltare, Raspberry Pi 4 Model B, vine cu un nou procesor quad-core de 1,5 GHz, memorie RAM de 4 GB LPDDR4, ceea ce va rezulta într-o viteză de lucru mult mai mare. Pe partea de conectivitate avem noul standard Bluetooth 5.0, wireless LAN b/g/n/ac și un port gigabit. De asemenea, placa dispune și de 2 porturi microHDMI, astfel vă puteți bucura de un spațiu de lucru extins și datorită faptului că placa dispune și de suport 4k.



Caracteristici generale:

- Procesor:
 - **Broadcom 2711;**
 - Arhitectură procesor: **Cortex – A72, 64 biți;**
 - Frecvență procesor: **1.5 GHz;**
 - Număr nuclee fizice: **4;**
- Memorie RAM: **4 GB LPDDR4 SDRAM;**
- Conectivitate:
 - **WiFi: 2.4 GHz și 5 GHz IEE 802.11 b/g/n/ac;**
 - **Bluetooth: 5.0 BLE;**
 - **Ethernet: Gigabit;**
 - **2 x USB 2.0;**
 - **2 x USB 3.0;**
- Header GPIO cu 40 pini (compatibil cu versiunile anterioare);
- Video & audio:
 - **2 x conector micro HDMI cu suport 4k;**
 - **1 x conector MIPI DSI (Display Serial Interface);**
 - **1 x conector MIPI CSI (Camera Serial Interface);**
 - **Output audio stereo și video composite;**

- Multimedia:
 - H.265 decode (4kp60)
 - H.264 decode (1080p60)
 - H.264 encode (1080p30);
 - OpenGL ES 1.1, 2.0, 3.0 graphics;
- Slot card microSD pentru rularea sistemului de operare și stocare date;
- Alimentare USB Type C: 5V, 3A;
- Temperatură de operare: 0 - 50 grade Celsius;

3.2 Electronica folosita

3.2.1 Senzor de distanta SR04



În cadrul proiectului, am utilizat senzorul ultrasonic HC-SR04, ce reprezintă un ansamblu realizat dintr-un transmițător ultrasonic, un receptor ultrasonic și un circuit de control. Senzorul ultrasonic HC-SR04 este un senzor de distanță ce poate detecta obiecte aflate la distanțe de până la patru metri, cu un consum redus de curent și care poate fi alimentat la o tensiune de 5 V. Este un senzor ce poate fi achiziționat la un preț redus, ideal pentru aplicații precum sisteme de alarmă, uși cu deschidere automată, detecția obstacolelor, senzor de parcare etc. În cadrul proiectului, senzorul ultrasonic este folosit pentru a măsura distanța, atunci când apare în fața sa un obstacol. Senzorul este setat ca atunci când întâlnește un obstacol aflat la o distanță mai mică de 100cm, să trimită un semnal către placa Raspberry Pi, care ulterior va acționa camera.

Acesta comunică cu placa Raspberry Pi prin protocolul I²C. Protocolul de comunicație I²C este utilizat pentru comunicația dintre unul sau mai multe dispozitive de tip „master” cu unul sau mai multe dispozitive de tip „slave”. Este un protocol de comunicație serială ce utilizează doar două fire pentru transmiterea datelor între dispozitive.

Caracteristici tehnice:

- Tensiune de lucru: DC 5V
- Curent de lucru: 15mA
- Frecventa de lucru: 40 Hz
- Raza maxima: 4m
- Raza minima: 2 cm
- Unghi de măsurare: 15 grade
- Semnal de intrare de declanșare: impuls TTL 10μS
- Semnal de ieșire eco Intrare semnal TTL pârghe și interval în proporție
- Dimensiune 45 * 20 * 15mm

3.2.2 Camera p5v04a sunny

Camera este conectată la Raspberry Pi prin conectorul MIPI CSI(CAMERA SERIAL INTERFACE) care se află între portul audio și portul HDMI. Pe Raspberry Pi B original, acesta se află între portul Ethernet și portul HDMI.

Aceasta acționează în momentul în care senzorul HC-SR04 întâlnește un obstacol aflat la o distanță mai mică de 100cm, realizând o captură.



Caracteristici tehnice:

Pixel: 5 milioane

Chip fotosensibil: OV5647

Rezoluție statică: 2896*1944

Numar pixeli: 2592*1944

Dimensiune pixel: 1.4 x 1.4μm

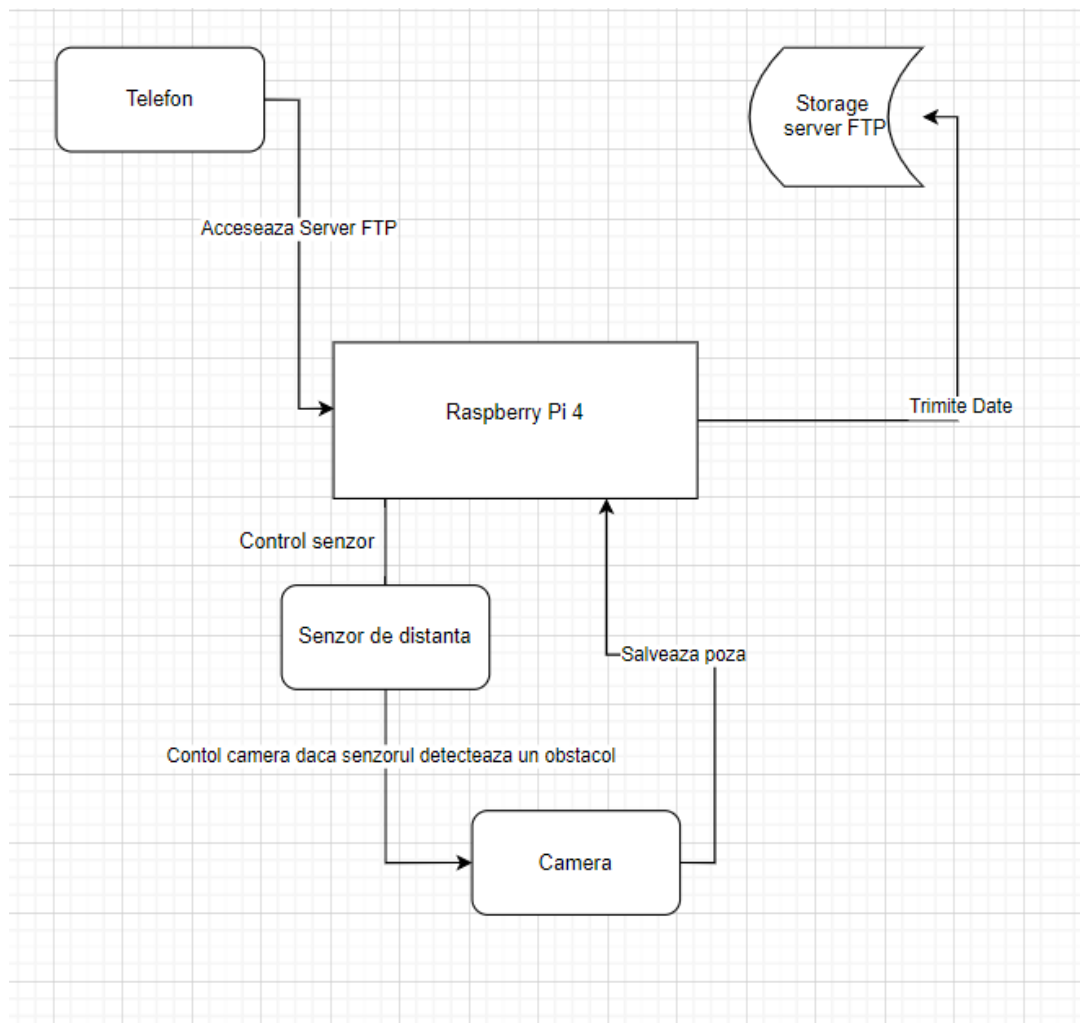
Unghi vedere: 54 x 41 grade

Accepta 1080p30, 720p60 si 640*480 60/90 video

Dimensiune: 25mm x 24mm x 9mm

Lungime banda: aprox 145mm

4. Schema bloc a implementarii



5. Costurile proiectului

Dispozitiv	Pret
Raspberry Pi 4 4 GB	430 lei
Camera p5v04a sunny	45,96 lei
Senzor HC SR04	5,76 lei
Pret total	481,72 lei

6. Prezentarea implementarii software

6.1 Limbajul Python

Python este un limbaj scriptic care are nevoie de un interpretor pentru a functiona. Acesta este un limbaj de uz general folosit de la aplicatii simple pana la aplicatii de uz industrial. Este un limbaj favorit in randul programatorilor care lucreaza in domeniul inteligentei artificiale.

6.2 RPI-GPIO

Acest pachet oferă o clasă pentru a controla GPIO pe un Raspberry Pi.

Rețineți că acest modul nu este potrivit pentru aplicații critice în timp real sau sincronizate. Acest lucru se datorează faptului că nu puteți prezice când Python va fi ocupat cu colectarea gunoiului. De asemenea, rulează sub nucleul Linux, care nu este potrivit pentru aplicații în timp real - este O/S multitasking și un alt proces poate avea prioritate față de procesor, provocând fluctuații în programul dumneavoastră.

6.3 FTPLIB

Python ftplib este un modul care implementează partea client a protocolului FTP. Conține o clasă de client FTP și câteva funcții de ajutor.

7. Performante

Performanța sistemului este una remarcabilă, acesta îndeplinind rolul unui sistem de securitate, reușind să detecteze fiecare obstacol (intrus) care se apropie la o distanță mai mică de 100cm și fotografiind live zona supravegheată.

8. Concluzii

Realizarea acestui proiect de către 2 studenți cu cunoștințe medii de programare arată faptul că se poate realiza un astfel de sistem, foarte ieftin și eficient și nu necesită cunoștințe avansate, decât dorința de a realiza ce s-a propus. Acest proiect este ușor de folosit, nici o operațiune nu este complicată și nu necesită explicații foarte multe.

9. Dezvoltări ulterioare

Acest proiect oferă posibilitatea fiecărei persoane să vină cu elemente noi pentru a face sistemul de securitate cât mai complex.

10. Bibliografie

[1] Raspberry Pi4 4GB

<https://www.optimusdigital.ro/ro/placi-raspberry-pi/8617-raspberry-pi-4-model-b-4gb-765756931182.html>

[2] Senzor HC-SR04

https://ardushop.ro/ro/home/47-modul-senzor-ultrasonic-detector-distanta.html?gclid=CjwKCAiAlfqOBhAeEiwAYi43F2xZ-H1pOIPXiE7NUFx2b97kn1VykcTV7cgpSXsTaWrvQwciA6TuRhoC0JMQAvD_BwE

[3] Camera p5v04a sunny

https://www.emag.ro/modul-camera-raspberry-pi3b-1080p-ai836-s307/pd/DX1SZTMBM/?cmpid=87002&gclid=CjwKCAiAlfqOBhAeEiwAYi43F_FmsP5ZNojz5eTLJonsGgA7W71Cy0CFrlmPhYQN8uAAPpPWt-ctURoCFXgQAvD_BwE

[4] Info Raspberry Pi

[Raspberry Pi - Wikipedia](#)

[5] RPI.GPIO

<https://pypi.org/project/RPi.GPIO/>

11. Anexa

```
1 import os
2 import ftplib
3 import glob
4 import RPi.GPIO as GPIO
5 import time
6 from datetime import datetime
7
8 GPIO.setmode(GPIO.BCM)
9 GPIO_TRIGGER = 18
10 GPIO_ECHO = 24
11 GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
12 GPIO.setup(GPIO_ECHO, GPIO.IN)
13
14 ftpUser = 'pi'
15 ftpServer = ''
16 ftppasswd = ''
17 folderproiect = '/images/'
18 folderftp = '/home/pi/ftp/files/'
19
20 session = ftplib.FTP(ftpServer, ftpUser, ftppasswd)
21
22 def take_picture():
23     os.system("libcamera-jpeg -n -t 1000 -o
24 images/poza_{}.jpeg".format(datetime.now().strftime("%d.%m.%Y_%H:%M:%S")))
25     #file = open(folderproiect, 'rb')
26     #file = glob.glob('images/*.jpeg')
27     #for image in glob.glob('images/*.jpeg'):
28         # with open(image, 'rb') as file:
29             # session.storbinary('STOR ' + folderftp + file)
30     #imageName = max(file, key = os.path.getctime)
31     #session.storbinary('STOR ' + folderftp, imageName)
32     #file.close()
33     os.system("mv images/*.jpeg {}".format(folderftp))
34
35 def distance():
36     GPIO.output(GPIO_TRIGGER, True)
37     time.sleep(0.00001)
38     GPIO.output(GPIO_TRIGGER, False)
39     StartTime = time.time()
40     StopTime = time.time()
41     while GPIO.input(GPIO_ECHO) == 0:
42         StartTime = time.time()
43     while GPIO.input(GPIO_ECHO) == 1:
44         StopTime = time.time()
```

```
45     TimeElapsed = StopTime - StartTime
46     distance = (TimeElapsed * 34300) / 2
47     return distance
48
49 if __name__ == '__main__':
50     try:
51         while True:
52             dist = distance()
53             print(dist)
54             if dist < 100:
55                 take_picture()
56                 time.sleep(1)
57     except KeyboardInterrupt:
58         print("Measurement stopped by User")
59         GPIO.cleanup()
60         session.close()
```