|  |
| --- |
| **Politehnica univeristy of timisoara** |
| Motion detection home surveillance system |
| May,2018  Group members:  Brenduse Larisa-Patricia  Cristea Sonia |
|  |

**1. Repository**

The project history, schematics, diagrams and codebase are contained under the following git repository:

https://github.com/patricialarisa/Microprocessor-systems

**2. User requirements**

1. The system must provide information regarding the surveillance of a place.
2. The surveillance system consists of an intelligent alarm which is build using RaspberryPi.
3. This alarm contains a video camera which detects motion using a microcontroller.
4. The pictures captured by the camera are processed and sent to the user via e-mail.
5. There is no limit of information received , since it it send to an e-mail account.

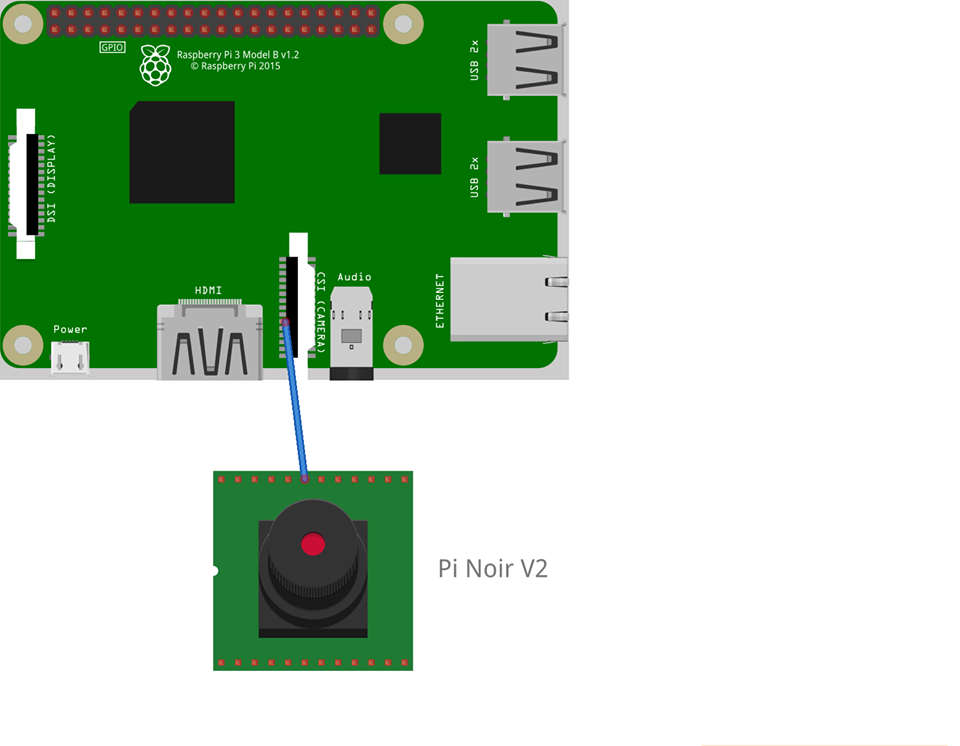
**3. System overview**

The overview of the system is depicted in Figure 1.

Figure 1: System overview diagram

**4. Circuit design**

The circuit design is depicted in Figure 2 :

Figure 2

1. **Software diagram**

****

Python elements :

* Server.py : defines the website on the localhost , along with all the necessary actions for activation/deactivaction of the alarm
* Surveillance.py : contains the code containing all the necessary code
* Utils.py : contains the e-mail address to which the images will be sent to and the necessary processes in sending the e-mail

Localhost:

For creating a database , we used wampp server . To access the database , we used PHPmyAdmin , loging in with a user which can be chosen , or just simply root.

HTML page:

The HTML page is generated from python source and it contains two buttons : one for activating the alarm and one for deactivation of the alarm.

The HTML page is created rendering a template we created in the folder called “templates” in our project. The part creating the page is shown below:

@app.route("/")

def hello():

return render\_template("index.html")

E-mail address:

This is the e-mail address all the processed images are sent to. All the processes used to sent the e-mail are found in the .py source files. The code for sending the e-mail is found below :

def send\_email(conf):

fromaddr = "pitestpi11@gmail.com"

for email\_address in conf['email\_address']:

toaddrs = email\_address

print("[INFO] Emailing to {}".format(email\_address))

text = 'Hey Someone in Your House!!!!'

subject = 'Security Alert!!'

message = 'Subject: {}\n\n{}'.format(subject, text)

msg = MIMEMultipart()

msg['From'] = fromaddr

msg['To'] = toaddrs

msg['Date'] = formatdate(localtime=True)

msg['Subject'] = subject

msg.attach(MIMEText(text))

1. **Usage and configuration**

For motion detection , the following steps are required:

* a motionless , static background image is computed , which is also used as a reference
* each frame is being captured
* each frame is transformed into a grey-toned picture and a gaussian blurr for detail reduction is added
* next step consists in computing the difference between the actual frame and the previous one to check if there is a continuous motion and the difference between the actual frame and the reference one
* if the continuous motion is present for several frames, an actual motion is taken into consideration
* after motion detection , an e-mail is sent

Other details about the files:

Conf.json is a configuration file , in which there are defined:

“show\_video” – if true , a video will be shown on the screen

“use\_email” – if true , an e-mail will be sent on motion detection

“email\_address” – the e-mail address of the recipient

“min\_upload\_seconds”- number of seconds between the frames in which motion is detected

“min\_motion\_frames”- minimum number of frames for which the motion is valid

“camera\_warmup\_time”- camera warm-up time

“delta\_tresh” – the threshold for different frames

“blur\_size” – blurr dimension

“resolution” – camera resolution

“fps” – frames per second

“min\_area” – minimum image area for motion validation

“debug\_mode”- debugging mode

In server.py file , there is defined the localhost site , along with the necessary information for activation/deactivation of the home surveillance system.

In pi\_surveillance.py file , the code containing all the necessary computation is found.

In utils.py file contains the recipient e-mail address and all the necessary processes for sending the e-mail.