

ASSIGNMENT NUMBER: E2

TITLE Develop a real time application of smart home.

PROBLEM STATEMENT /DEFINITION Develop a Real time application like a smart home with following requirements: If anyone comes at door the camera module automatically captures his image send it to the email account of user or send notification to the user. Door will open only after user's approval.

OBJECTIVE To develop real time IoT based application.

S/W PACKAGES AND HARDWARE APPARATUS USED Raspberry Pi, Pi Camera, PIR Sensor, LED, Bread Board, Resistor (1k), Connecting wires, Power supply.

Aim: Develop a Real time application like a smart home with following requirements: If anyone comes at door the camera module automatically captures his image send it to the email account of user or send notification to the user. Door will open only after user's approval.

Pre-requisite:

Basic knowledge of embedded system and IOT

Learning Objectives:

The students will be able to

- ☐ To Develop an IoT based (smart) application.

Learning Outcomes:

The students will be able to

- ☐ Implement an architectural design for IoT for specified requirement
- ☐ Solve the given societal challenge using IoT

Theory: Raspberry Pi is an ARM cortex based popular development board designed for Electronic Engineers and Hobbyists. With the processing speed and memory, Raspberry Pi can be used for performing different functions at a time, like a normal PC, and hence it is called Mini Computer in your palm. Here interfacing Pi camera with Raspberry Pi to capture the image of every visitor which has entered through the Gate or door. In this assignment, whenever any person is arrived at the Gate, he has to press a button to open the Gate, and as soon as he/she press the button, his/her picture will be captured and saved in the system with the Date and time of the entry. This can be very useful for security and surveillance purpose. This system is very useful in offices or factories where visitor entry record is maintained for visitors and attendance record is maintained for employees. This Monitoring system will digitize and automate the whole visitor entries and attendances, and there will be no need to maintain them manually. This system can be either operated by the person himself or there can be operator for pressing the button for every visitor. This is a good project for getting started with Pi camera and interface it with Raspberry Pi. Working Explanation: Working of this Raspberry Pi Monitoring System is simple. In this, a Pi camera is used to capture the images of visitors, when a push button is pressed or triggered. A DC motor is used as a gate. Whenever anyone wants to enter in the place then he/she needs to push the button.

After pushing the button, Raspberry Pi sends command to Pi Camera to click the picture and save it. After it, the gate is opened for a while and then gets closed again. The buzzer is used to generate sound when button pressed and LED is used for indicating that Raspberry Pi is ready to accept Push Button press, means when LED is ON, system is ready for operation.

Circuit Explanation: Circuit of this Raspberry Pi Visitor Surveillance System is very simple. Here a Liquid Crystal Display (LCD) is used for displaying Time/Date of visitor entry and some other messages. LCD is connected to Raspberry Pi in 4-bit mode. Pins of LCD namely RS, EN, D4, D5, D6, and D7 are connected to Raspberry Pi GPIO pin number 18, 23, 24, 16, 20 and 21. Pi camera module is connected at camera slot of the Raspberry Pi. A buzzer is connected to GPIO pin 26 of Raspberry Pi for indication purpose. LED is connected to GPIO pin 5 through a 1k resistor and a push button is connected to GPIO pin 19 with respect to ground, to trigger the camera and open the Gate. DC motor (as Gate) is connected with Raspberry Pi GPIO pin 17 and 27 through Motor Driver IC (L293D).

To connect the Pi Camera, insert the Ribbon cable of Pi Camera into camera slot, slightly pull up the tabs of the connector at RPi board and insert the Ribbon cable into the slot, then gently push down the tabs again to fix the ribbon cable.

Raspberry Pi Configuration and Programming Explanation: We are using Python language here for the Program. Before coding, user needs to configure Raspberry Pi. You should follow two tutorials for Getting Started with Raspberry Pi and Installing & Configuring Raspbian Jessie OS in Pi: **Circuit Description:** In this Intruder Alert System, we only need to connect Pi Camera module and PIR sensor to Raspberry Pi 3. Pi Camera is connected at the camera slot of the Raspberry Pi and PIR is connected to GPIO pin 18. A LED is also connected to GPIO pin 17 through a 1k resistor.

After successfully installing Raspbian OS on Raspberry Pi, we need to install Pi camera library files for run this project in Raspberry pi. To do this we need to follow given commands:

```
$ sudo apt-get install python-picamera
```

```
$ sudo apt-get install python3-picamera
```

After it, user needs to enable Raspberry Pi Camera by using Raspberry Pi Software Configuration Tool (raspi-config):

```
$ sudo raspi-config
```

Then select Enable camera and Enable it.

Then user needs to reboot Raspberry Pi, by issuing `sudo reboot`, so that new setting can take. Now your Pi camera is ready to use.

Now after setting up the Pi Camera, we will install software for sending the mail. Here we are using `ssmtp` which is an easy and good solution for sending mail using command line or using Python Script. We need to install two Libraries for sending mails using SMTP:

```
sudo apt-get install ssmtp
```

```
sudo apt-get install mailutils
```

After installing libraries, user needs to open `ssmtp.conf` file and edit this configuration file as shown in the Picture below and then save the file. To save and exit the file, Press „CTRL+x“, then „y“ and then press „enter“.

```
sudo nano /etc/ssmtp/ssmtp.conf
```

```
root=YourEmailAddress
```

```
mailhub=smtp.gmail.com:587
```

```
hostname=raspberrypi
```

```
AuthUser=YourEmailAddress
```

```
AuthPass=YourEmailPassword
```

```
FromLineOverride=YES
```

```
UseSTARTTLS=YES
```

```
UseTLS=YES
```

We can also test it by sending a test mail by issuing below command, you shall get the mail on the mentioned email address if everything is working fine:

```
echo "Hello saddam" | mail -s "Testing..." saddam4201@gmail.com
```

The Python Program of this project plays a very important role to perform all the operations. First of all, we include required libraries for email, initialize variables and define pins for PIR, LED and other components. For sending simple email, `smtplib` is enough but if you want to send mail in cleaner way with subject line, attachment etc. then you need to use `MIME` (Multipurpose Internet Mail Extensions).

```
import RPi.GPIO as gpio
```

```
import picamera
```

```
import time
```

```
import smtplib
```

```
from email.MIMEMultipart import MIMEMultipart
```

```
from email.MIMEText import MIMEText
```

```
from email.MIMEBase import MIMEBase
```

```
from email import encoders
```

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

After it, we have initialized mail and define mail address and messages:

```
fromaddr = "raspiduino4201@gmail.com"
```

```
toaddr = "saddam4201@gmail.com"
```

```
mail = MIMEMultipart()
```

```
mail['From'] = fromaddr
```

```
mail['To'] = toaddr
```

```
mail['Subject'] = "Attachment"
```

```
body = "Please find the attachment"
```

Then we have created def sendMail(data) function for sending mail:

```
def sendMail(data):
```

```
    mail.attach(MIMEText(body, 'plain'))
```

```
    print data
```

```
    dat='%s.jpg'%data
```

```
    print dat
```

```
    attachment = open(dat, 'rb')
```

```
    image=MIMEImage(attachment.read())
```

```
    attachment.close()
```

```
    mail.attach(image)
```

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

```
    server.starttls()
```

```
    server.login(fromaddr, "your password")
```

```
    text = mail.as_string()
```

```
    server.sendmail(fromaddr, toaddr, text)
```

```
    server.quit()
```

Function def capture_image() is created to capture the image of intruder with time and date.

```
def capture_image():
```

```
    data= time.strftime("%d_%b_%Y|%H:%M:%S")
```

```
    camera.start_preview()
```

```
    time.sleep(5)
```

```
    print data
```

```
    camera.capture('%s.jpg'%data)
```

```
    camera.stop_preview()
```

```
    time.sleep(1)
```

```
    sendMail(data)
```

Then we initialized the Picamera with some of its settings:

```
camera = picamera.PiCamera()
```

```
camera.rotation=180
```

```
camera.awb_mode= 'auto'
```

```
camera.brightness=55
```

And now in last, we have read PIR sensor output and when its goes high Raspberry Pi calls the capture_image() function to capture the image of intruder and send an alert message with the picture of intruder as an attachment. We have used sendmail() insdie capture_image() function for sending the mail.

```
while 1:
```

```
if gpio.input(pir)==1:
```

```
    gpio.output(led, HIGH)
```

```
    capture_image()
```

```
    while(gpio.input(pir)==1):
```

```
        time.sleep(1)
```

```
    else:
```

```
        gpio.output(led, LOW)
```

```
        time.sleep(0.01)
```

Code:

```
import RPi.GPIO as gpio
```

```
import picamera
```

```
import time
```

```
import smtplib
```

```
from email.MIMEMultipart import MIMEMultipart
```

```
from email.MIMEText import MIMEText
```

```
from email.MIMEBase import MIMEBase
```

```
from email import encoders
```

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

```
fromaddr = "raspiduino4201@gmail.com" # change the email address accordingly
```

```
toaddr = "saddam4201@gmail.com"
```

```
mail = MIMEMultipart()
```

```

mail['From'] = fromaddr
mail['To'] = toaddr
mail['Subject'] = "Attachment"
body = "Please find the attachment"

led=17
pir=18
HIGH=1
LOW=0

gpio.setwarnings(False)
gpio.setmode(gpio.BCM)
gpio.setup(led, gpio.OUT) # initialize GPIO Pin as outputs
gpio.setup(pir, gpio.IN) # initialize GPIO Pin as input
data=""

def sendMail(data):
    mail.attach(MIMEText(body, 'plain'))
    print data
    dat='%s.jpg'%data
    print dat
    attachment = open(dat, 'rb')
    image=MIMEImage(attachment.read())
    attachment.close()
    mail.attach(image)

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "your password")
text = mail.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()

def capture_image():
    data= time.strftime("%d_%b_%Y|%H:%M:%S")

```

```
camera.start_preview()
time.sleep(5)
print data
camera.capture('%s.jpg'%data)
camera.stop_preview()
time.sleep(1)
sendMail(data)
gpio.output(led , 0)
camera = picamera.PiCamera()
camera.rotation=180
camera.awb_mode= 'auto'
camera.brightness=55
while 1:
    if gpio.input(pir)==1:
        gpio.output(led, HIGH)
        capture_image()
        while(gpio.input(pir)==1):
            time.sleep(1)
    else:
        gpio.output(led, LOW)
        time.sleep(0.01)
```

Conclusion: Hence the experiment was carried out successfully.