

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM 590014



Internet of Things Project Report on
“Smart Doorbell”

By

YASHAS N (1BM16CS155)
SUMUKHA R NADIG (1BM16CS110)
VANDITHA SHIVA KUMAR (1BM16CS122)
SMARAN S RAO (1BM16CS153)

Under the Guidance of

Mrs. Sheethal

Assistant Professor, Department of CSE
BMS College of Engineering

IoT Application Development carried out at



Department of Computer Science and Engineering
BMS College of Engineering
(Autonomous college under VTU)
P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019
2018-2019

BMS COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Internet of Things project titled “Smart Doorbell” has been carried out by YASHAS N (1BM16CS155), SUMUKHA R NADIG (1BM16CS110), VANDITHA SHIVA KUMAR (1BM16CS122), SMARAN S RAO (1BM16CS153) during the academic year 2018-2019.

Signature of the guide

Mrs. Sheethal

Assistant Professor

Department of Computer Science and Engineering

BMS College of Engineering, Bangalore

Examiners

Name

Signature

1.

2.

BMS COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



DECLARATION

We, YASHAS N (1BM16CS155), SUMUKHA R NADIG (1BM16CS110),

VANDITHA SHIVA KUMAR (1BM16CS122), SMARAN S RAO (1BM16CS153) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this IoT application development work entitled "Smart Doorbell" has been carried out by us under the guidance of Sheetal V.A, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Aug-Dec 2018.

We also declare that to the best of our knowledge and belief, the development reported here is not from any other work/report by any other students.

Signature

YASHAS N (1BM16CS155)

SUMUKHA R NADIG (1BM16CS110)

VANDITHA SHIVA KUMAR (1BM16CS122)

SMARAN S RAO (1BM16CS153)

Introduction

1. Objective of the project

The Smart doorbell project allows users to remotely control access to their homes when there is a visitor.

2. Description of the project

The Smart doorbell essentially alerts the owner of the house when someone rings the doorbell by sending them a camera captured image of the visitor to their email and telegram account via a chatbot. The owner can open the door to let the visitor in or play a prerecorded voice clip through the doorbell indicating their absence. In the latter case, the visitor can record a voice message regarding their purpose of visit which is sent to the owner's telegram chat.

The image is captured using a pi-camera and in the case of insufficient lighting which is detected by an LDR sensor, a light is switched on to provide a higher quality image of the visitor. A Raspberry Pi is used to send the image and transfer the voice recordings to and from the owner's telegram chat and email.

Literature Survey

Sln.	Name of product	Commercial status	Features
1.	Normal Doorbell	Commercial	When pressed, a sound is made within the house.
2.	Speech assisted bell	Commercial	Allows people at home to talk to visitor through a speaker and microphone.
3.	Door Alarm system	Commercial	Sounds an alarm if someone tries to break in to the house.

Our Proposed Project and the advantages:

The Smart Doorbell allows users to have remote control through telegram and they also get to see the visitor's face before.

Hardware and Software Requirements

1. Hardware Requirements

Components used:

1. Raspberry Pi
2. Bread board
3. Pi Camera
4. LDR sensor
5. Capacitor
6. Resistor
7. 2 LEDs
8. Jumper wires
9. Push Button

Cost Analysis:

Slno.	Component	Cost (Rs.)
1.	Raspberry Pi (rented)	400
2.	Bread Board	30
3.	Pi Camera	450
4.	LDR sensor	10
5.	Capacitor	7
6.	Resistor	3
7.	LED	2
8.	Jumper Wire	3
9.	Push Button	15

2. Software Requirements

Operating System – Raspbian (linux)

IDE – Idle 3.5 (for python)

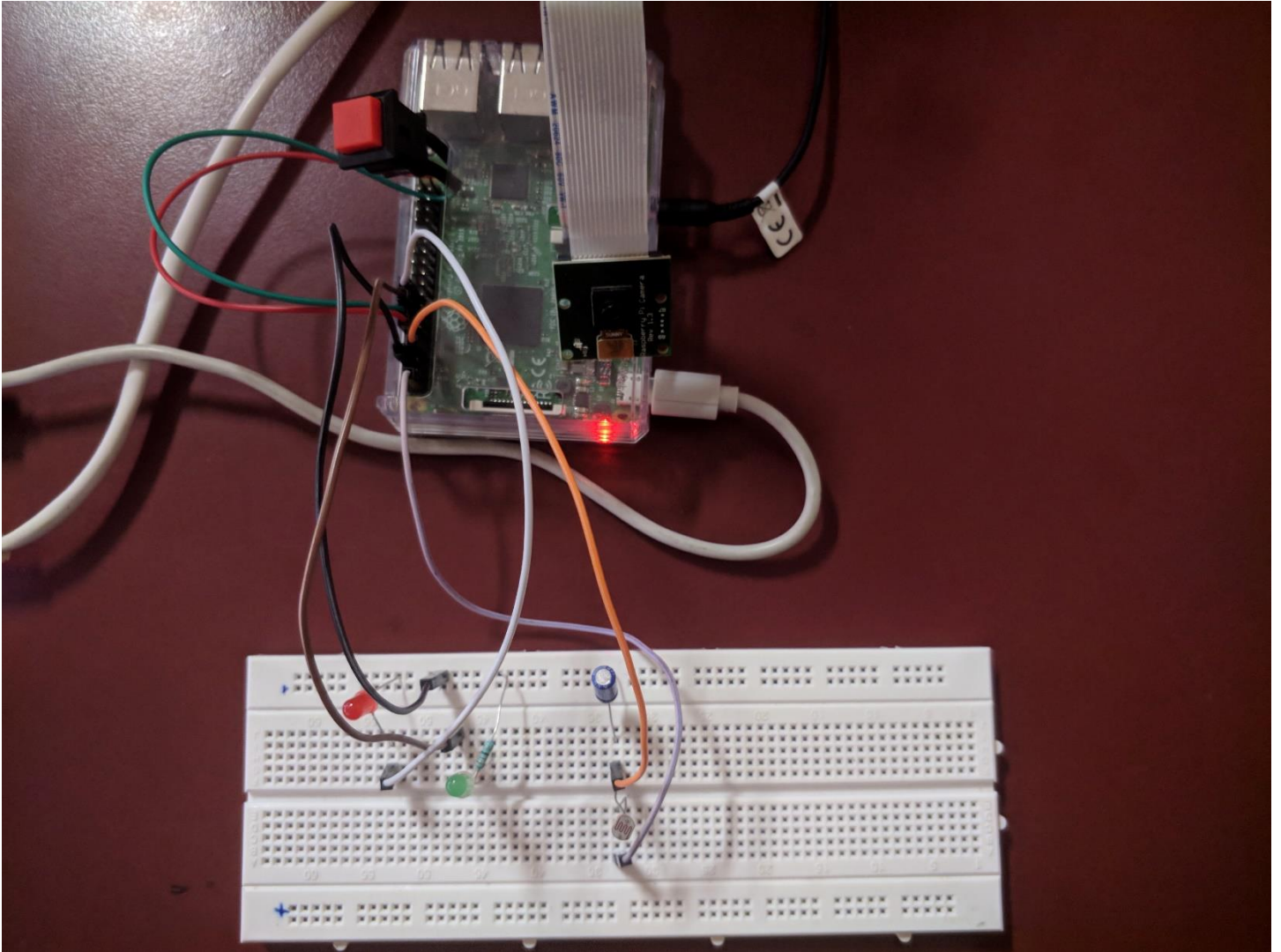
E-mail service – Gmail

Chat service – Telegram (chatbot)

Cloud service - Firebase

Design

1. Circuit Diagram



2. Interaction between elements

1. Raspberry pi is connected to a computer via an ethernet cable and USB power cable.
2. Pi Camera is fitted onto the Raspberry Pi
3. An LDR sensor is connected in series with a 1microFarad capacitor. The short end of the capacitor is connected to pin 6 (gnd). The Other end of the capacitor in series with the LDR sensor is connected to pin 7. The other end of the LDR sensor is connected to pin 7. This connection is done using the bread board.
4. A LED is connected, in series with the 330ohm resistor to ground connection. The other end of the LED is connected to pin 8.
5. Another LED is used with the short end connected to ground on the bread board and the longer end connected to pin 10.
6. One end of the push button is connected to pin 12 (gnd) and the other end to pin 14.

Implementation

1. Steps to implement

1. Connect the Raspberry Pi to your computer and start the OS through a SSH session. Open and run the python script.
2. Press the button (doorbell) and wait for the registered email address and telegram account to receive a message and image through email and the chatbot respectively. The LDR sensor prompts an LED to switch on (simulating a light bulb to provide better lighting) before the image is taken by the Pi Camera in case there is insufficient lighting.
3. The owner will send either a /open or a /out command to the chatbot. The former causes an LED to glow simulating the door being unlocked. The latter plays a prerecorded customized message informing the visitor of the owner's absence. This is followed by the visitor sending a voice message describing their purpose of visit, which is sent to the registered telegram account from the chatbot. This is implemented through Firebase.
4. The owner can then send a /close command to the chatbot, causing the LED that was turned on for the /open command to stop glowing, simulating closing of the door when a visitor leaves.

2. Source Code – Python script

```
File Edit Format Run Options Window Help
from picamera import PiCamera
from time import sleep, gmtime, strftime
import RPi.GPIO as gpio
import datetime
import time
import sys
import os
import smtplib
from shutil import copyfile
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email.mime.text import MIMEText
from email.utils import COMMASPACE, formatdate
from email import encoders
import telepot
from telepot.loop import MessageLoop
import pyrebase
import urllib

doorled=18
pin_to_circuit=7
led=16
gpio.setmode(gpio.BOARD)
gpio.setup(doorled,gpio.OUT)
gpio.output(18,gpio.LOW)

gpio.setup(12, gpio.IN, pull_up_down=gpio.PUD_UP)

camera=PiCamera()
camera.exposure_mode= 'antishake'

USERNAME = "iotdoorbellding@gmail.com"
PASSWORD = "iotdoorbell4"
MAILTO = "yashasvm3@gmail.com"
recentsnapshot="/home/pi/Documents/recent.png"

def sendmail():
    print('Sending mail')
    msg = MIMEMultipart()
    text = "Hi, \n\nSomeone knocked on your door at " + strftime("%l:%M %p on %d-%m-%Y")
    msg['to'] = MAILTO
    msg['from'] = "iotdoorbellding@gmail.com"
    msg['subject'] = "Ding Dong at " + strftime("%l:%M %p on %d-%m-%Y")

    msg.attach(MIMEText(text))
    part = MIMEBase('application', "octet-stream")
    part.set_payload(open(recentsnapshot, "rb").read())
    encoders.encode_base64(part)

Ln: 179 Col: 14
```

```
File Edit Format Run Options Window Help
encoders.encode_base64(part)

part.add_header('Content-Disposition', 'attachment; filename="photo.jpg"')

msg.attach(part)
server = smtplib.SMTP('smtp.gmail.com:587')
server.ehlo_or_helo_if_needed()
server.starttls()
server.ehlo_or_helo_if_needed()
server.login(USERNAME, PASSWORD)
server.sendmail(USERNAME, MAILTO, msg.as_string())
print('sent mail')
server.quit()

def audio():
    config = {
        "apiKey": "AizasyB3p_5jR-cJrHT2aY2R7a_KJedyHz-j6eg",
        "authDomain": "doorbell-7152c.firebaseio.com",
        "databaseURL": "https://doorbell-7152c.firebaseio.com",
        "storageBucket": "doorbell-7152c.appspot.com",
        "serviceAccount": "doorbell-7152c-firebase-adminsdk-b8ypk-1e8b5934a6.json"
    }

    firebase = pyrebase.initialize_app(config)
    storage = firebase.storage()

    storage.child("images/piii.mp3").download("downloaded.mp3")

def handle(msg):
    chat_id = msg['chat']['id'] # Receiving the message from telegram
    command = msg['text'] # Getting text from the message

    print ('Received:')
    print(command)

    if command == '/close':
        gpio.output(doorled,gpio.LOW)
        bot.sendMessage(chat_id, str("Door closed"))

    # Comparing the incoming message to send a reply according to it
    if command == '/open':
        gpio.output(doorled,gpio.HIGH)
        bot.sendMessage(chat_id, str("Door open"))

    if command == '/out':
        os.system('omxplayer pii.mp3 &')
        time.sleep(5)
        bot.sendAudio(chat_id,open('downloaded.mp3','rb'))

Ln: 50 Col: 0
```


Untitled*

File Edit Format Run Options Window Help

```

gpio.setup(led,gpio.OUT)
gpio.output(led,gpio.LOW)
def rc_time(pin_to_circuit):
    count=0
    gpio.setup(pin_to_circuit, gpio.OUT)
    gpio.output(pin_to_circuit, gpio.LOW)
    time.sleep(0.1)

    gpio.setup(pin_to_circuit, gpio.IN)

    while(gpio.input(pin_to_circuit)==gpio.LOW):
        count=count+1
    if count>2500:
        return 2
    else:
        return 3

try:
    while True:

        button_status=gpio.input(12)

        if button_status==False and count==0:
            count=1
            print('Bell pressed')
            os.system('omxplayer doorbell-5.mp3 &')
            sleep(1)
            print("Your photo is being taken. Please stand in front of the camera")
            newfile='/home/pi/Documents/'+ datetime.datetime.now().strftime('%Y-%m-%d%H:%M:%S')+'.png'
            val=rc_time(pin_to_circuit)
            if val==2:
                gpio.output(led, gpio.HIGH)
            else:
                gpio.output(led, gpio.LOW)
                camera.capture(newfile)
                gpio.output(led, gpio.LOW)
                copyfile(newfile,recentssnapshot)
                sendmail()
                print("Sending mail")
                print("sent mail")
                print("Sending to telegram")
                audio()
                bot=telepot.Bot('698352448:AAHHLrJyeCNOnusuGz85EofEoEd0SnFMj18')
                bot.sendMessage(-1001441479475,str("Hi, \n\nSomeone knocked on your door at " + strftime("%l:%M %p on %d-%m-%Y")))
                bot.sendPhoto(-1001441479475,open(recentssnapshot,'rb'))
                MessageLoop(bot, handle).run_as_thread()
                print ('Listening....')
                print("Sent to telegram")
                sleep(.2)

```

Ln: 148 Col: 11

Untitled*

File Edit Format Run Options Window Help

```

        count=count+1
    if count>2500:
        return 2
    else:
        return 3

try:
    while True:

        button_status=gpio.input(12)

        if button_status==False and count==0:
            count=1
            print('Bell pressed')
            os.system('omxplayer doorbell-5.mp3 &')
            sleep(1)
            print("Your photo is being taken. Please stand in front of the camera")
            newfile='/home/pi/Documents/'+ datetime.datetime.now().strftime('%Y-%m-%d%H:%M:%S')+'.png'
            val=rc_time(pin_to_circuit)
            if val==2:
                gpio.output(led, gpio.HIGH)
            else:
                gpio.output(led, gpio.LOW)
                camera.capture(newfile)
                gpio.output(led, gpio.LOW)
                copyfile(newfile,recentssnapshot)
                sendmail()
                print("Sending mail")
                print("sent mail")
                print("Sending to telegram")
                audio()
                bot=telepot.Bot('698352448:AAHHLrJyeCNOnusuGz85EofEoEd0SnFMj18')
                bot.sendMessage(-1001441479475,str("Hi, \n\nSomeone knocked on your door at " + strftime("%l:%M %p on %d-%m-%Y")))
                bot.sendPhoto(-1001441479475,open(recentssnapshot,'rb'))
                MessageLoop(bot, handle).run_as_thread()
                print ('Listening....')
                print("Sent to telegram")
                sleep(.2)

except KeyboardInterrupt:
    camera.close()
except Exception:
    print("Not connected to internet")
    camera.close()
finally:
    try:
        os.remove("downloaded.mp3")
    except:
        print('Exiting')
gpio.cleanup()

```

Ln: 148 Col: 11

3. Console Output

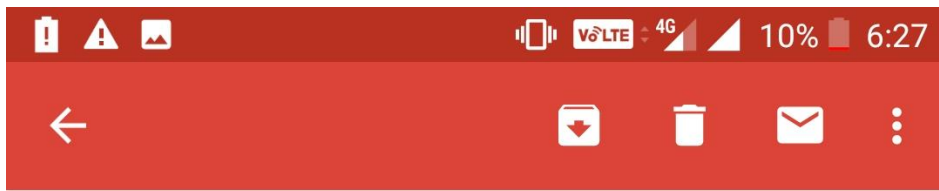
```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Desktop/withldr.py =====
Bell pressed
Your photo is being taken. Please stand in front of the camera
Sending mail
sent mail
Sending to telegram
Listening...
Sent to telegram
Received:
/start
Received:
/open
Received:
/close
>>> |
```

Ln: 19 Col: 4

```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Desktop/withldr.py =====
Bell pressed
Your photo is being taken. Please stand in front of the camera
Sending mail
sent mail
Sending to telegram
Listening...
Sent to telegram
Received:
/out
```

Ln: 6 Col: 0

4. Email Received



Ding Dong at 6:24 PM on
17-12-2018 Inbox



iotdoorbellding@gmail.com
to me
6:25 PM [View details](#)



Hi,

Someone knocked on your door at 6:24 PM on 17-12-2018



photo.jpg



[Got it, thanks!](#)

[What is this?](#)

[Thank you!](#)



Reply

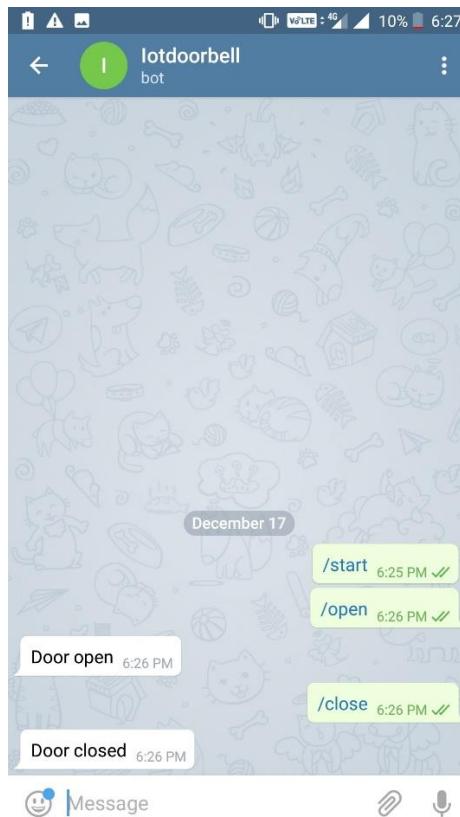


Reply all



Forward

5. Telegram Chatbot



Conclusion

The Smart Doorbell project can be used to mass produce a commercialized product with high level of security and functionality. There is scope for scaling and improvement in terms of reducing the visitor's waiting time with better server and internet facilities. This simulation of a real-life practical commodity makes use of various concepts under Internet of Things.