Lab 4: Tweet Intruder

Noah Buchanan
Derek Yocum
CS 4363: IoT Development
University of Arkansas - Fort Smith
Fall 2021

November 18, 2021

A quick note to make, our account is IoT@UAFSG8. The login used to make UAFSG3 requires portal guard login as it was a school email and this prevented us from connecting our twitter to the application. Our solution was to just make a new twitter account with a personal email.

Modifications of Code

There were only two major modifications made to the code, first was in the initialization of the program:

```
APP_KEY = 'msSEqlAM2E46GYKRVdDT5TybY'
APP_SECRET = 'u7UWElEZPyzLRf40oza623cLmiW2O0xmah3vltLMqydTGz2aXN'

auth = tweepy.OAuthHandler(APP_KEY, APP_SECRET)
auth.set_access_token('1456346752924094465-UlqKscWKOjYANwsJFVFaCb02hBM8M2', 'CVv
api = tweepy.API(auth)

pygame.init()
pygame.camera.init()
cam = pygame.camera.Camera("/dev/video0", (640,480))
scam.start()
```

Here we are authorizing our application to take control of the twitter account as well as starting our camera to be ready to take a picture of any intruders.

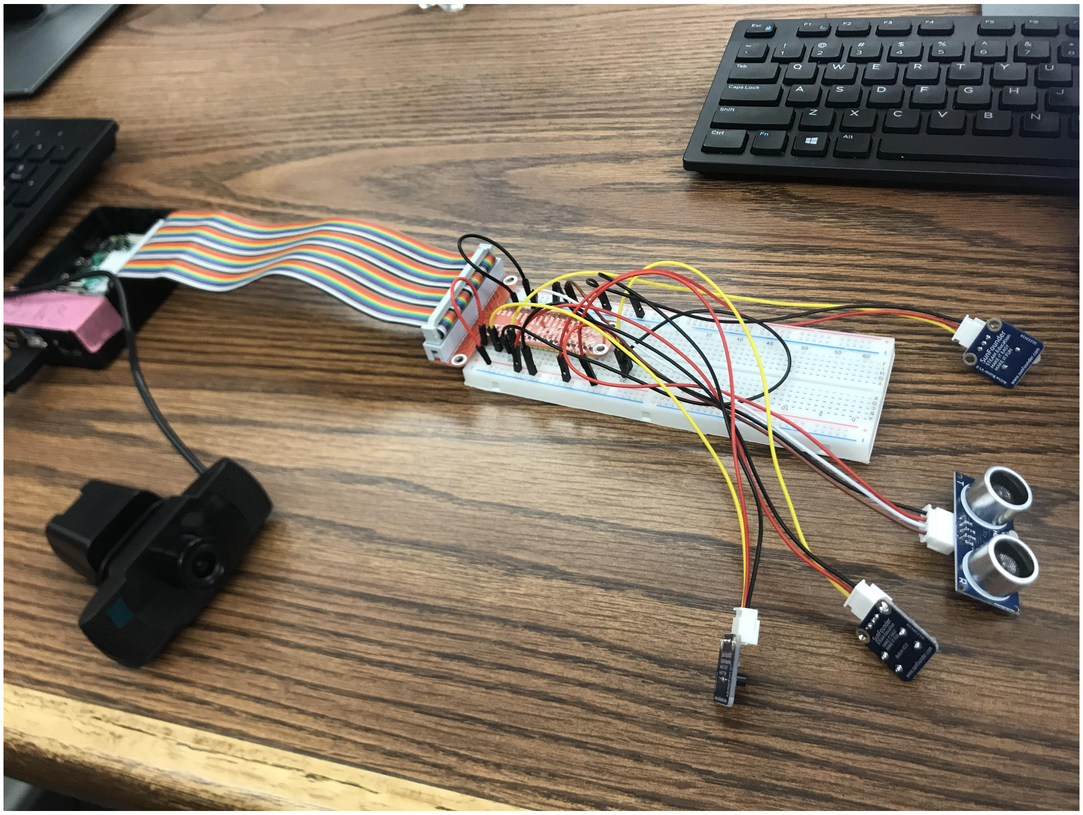
The second change is when something is actually detected in our program:

```
if(dis < 20):
  t = time.time()
  t = datetime.fromtimestamp(t)
  global alert
  alert += ('<p>Motion detected : ({dist} cm) [ '.format(dist = int(dis)))
  alert += str(t)
  alert += ']</p>'
  GPIO.setup(TRIG, GPIO.OUT)
  GPIO.output(TRIG, GPIO.HIGH)
  self.beep(0.5)
```

```
  image = cam.get_image()
  pygame.image.save(image,'image_1.jpg')
  image = 'image_1.jpg'
  media = api.media_upload(image)
  tweet = "Intruder detected!" + str(t)
  api.update_status(status = tweet, media_ids = [media.media_id])
```

I have surrounded the modified portion of what happens when an object is detected. A picture is taken and saved, and then a tweet is posted with a time stamp of the photo taken.

Hardware Connections



Source Code

```
import RPi.GPIO as GPIO
import cherrypy
import time
import random
import os
import tweepy
import pygame
import pygame.camera
from pygame.locals import *
from datetime import datetime

api = tweepy.API()
BtnPin = 11
```

```
TRIG    = 11
ECHO    = 12
ds18b20 = ’’

running = False
alert = ’’

class Page:
    @cherrypy.expose
    def index(self):
        global running
        if running:
            running = False
            self.destroy()
        return ”””<!DOCTYPE html>
                <html lang=”en”>
                <head>
                    <meta charset=”UTF−8”>
                    <title>Security System</title>
                    <link href=”/static/css/styles.css” rel=”stylesheet”>
                </head>
                <body class = ”bod”>
                <div>
                <h1 class =”title”> Security System &trade; </h1>
                </div class = ”title”>
                <form method=”get” action=”/remoteStart”>
                    <button class = ”btn”>Turn On</button>
                </form>
                <div class=”temp”>
                <h1> Current Tempature: <h1>
                Not currently on
                </div>
                <div class =”alert”><h1> ALERTS: <h1>
                ””” + alert +”””
                </div>
                </body>
                </html>”””

    @cherrypy.expose
    def remoteStart(self):
        print(alert)
        global running
        if not running:
            self.setup()
            running = True
            t = threading.Thread(target=self.sensors)
            t.daemon
            t.start()
        temp = self.read()
        return """<!DOCTYPE html>
                <html lang="en">
                <head>
                    <meta charset="UTF-8">
                    <title>Security System</title>
                    <script type = "text/javascript">
                        function autoRefreshPage()
                        {
                            window.location = window.location.href
                        }
                        setInterval('autoRefreshPage()', 1000);
                    </script>
                    <link href="/static/css/styles.css" rel="stylesheet">
                </head>

                <body class = "bod">
                <div class = "title">
                <h1 class ="title"> Security System &trade; </h1>
                </div>
                <form method="get" action="/index">
                    <button class = "btn">Turn Off</button>
                </form>
                <div class="temp"><h1> Current Tempature: <h1>
                """ + str(temp) + """ </div> <div class = "alert"> <h1> ALERTS:
                </div>
                </body>
                </html>"""


    def sensors(self):
        self.button()

    def setup(self):
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(13, GPIO.OUT)
        GPIO.output(13, GPIO.HIGH)
        GPIO.setup(ECHO, GPIO.IN)
        GPIO.setup(BtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# Set BtnPin's mode is input, and pull up to high level(3.3V)
        GPIO.add_event_detect(BtnPin, GPIO.BOTH, callback=self.detect, bouncetim
        global ds18b20
        for i in os.listdir('/sys/bus/w1/devices'):
                if i != 'w1_bus_master1':
                    ds18b20 = '28-01201f862d36'

def distance(self):
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.output(TRIG, GPIO.HIGH)

    GPIO.output(TRIG, 0)
    time.sleep(0.000002)

    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)

    while GPIO.input(ECHO) == 0:
            a = 0
    time1 = time.time()
    while GPIO.input(ECHO) == 1:
            a = 1
    time2 = time.time()
    during = time2 - time1
    return during * 340 / 2 * 100

def button(self):
    global api
    global running
    while running:
        if GPIO.input(BtnPin)==0:
            break
        dis = self.distance()
        if(dis < 20):
            t = time.time()
            t = datetime.fromtimestamp(t)
            global alert
            alert += ('<p>Motion detected : ({dist} cm) ['.format(dist = int
            alert += str(t)
            alert += ']</p>'
            GPIO.setup(TRIG, GPIO.OUT)
            GPIO.output(TRIG, GPIO.HIGH)
            self.beep(0.5)
            image = cam.get_image()
            pygame.image.save(image,'image_1.jpg')
            image = 'image_1.jpg'
            media = api.media_upload(image)
            tweet = "Intruder detected!" + str(t)
            api.update_status(status = tweet, media_ids = [media.media_id])
        GPIO.setup(BtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
            print (int(dis),'cm')
            print ('')
            time.sleep(0.5)


    def on(self):
        GPIO.output(13, GPIO.LOW)

    def off(self):
        GPIO.output(13, GPIO.HIGH)

    def beep(self,x):
        self.on()
        time.sleep(x)
        self.off()
        time.sleep(x)

    def read(self):
#       global ds18b20
        location = '/sys/bus/w1/devices/' + ds18b20 + '/w1_slave'
        tfile = open(location)
        text = tfile.read()
        tfile.close()
        secondline = text.split("\n")[1]
        temperaturedata = secondline.split(" ")[9]
        temperature = float(temperaturedata[2:])
        temperature = temperature / 1000
        return temperature

    def destroy(self):
        GPIO.setup(TRIG, GPIO.OUT)
        GPIO.setup(ECHO, GPIO.OUT)
        GPIO.output(TRIG, GPIO.HIGH)
        GPIO.output(ECHO, GPIO.HIGH)
        GPIO.cleanup()                    # Release resource

    def detect(self,chn):
        pass



if __name__ == '__main__':
    conf = {
        '/': {
            'tools.staticdir.root': os.getcwd()
            },
    '/static': {
        'tools.staticdir.on': True,
        'tools.staticdir.dir': 'public'
        }
    }
APP_KEY = 'msSEqlAM2E46GYKRVdDT5TybY'
APP_SECRET = 'u7UWElEZPyzLRf40oza623cLmiW2O0xmah3vltLMqydTGz2aXN'

auth = tweepy.OAuthHandler(APP_KEY, APP_SECRET)
auth.set_access_token('1456346752924094465-UlqKscWKOjYANwsJFVFaCb02hBM8M2',
api = tweepy.API(auth)

pygame.init()
pygame.camera.init()
cam = pygame.camera.Camera("/dev/video0", (640,480))
cam.start()

cherrypy.quickstart(Page(), '/', conf)
```