

1167 物聯網概論(英文授課)
Introduction to Internet of Things

Assignment 3 Report

Student Information

- 0816142 卓灝辰
- 0816106 陳秉劼
- 0816189 戴珮珊

Objectives

To simulate a plant monitoring iot system that can monitor its temperature, the state of irrigation system(water is enough or not) and maybe intruders(sound and distance).

To know how to send data from sensors from raspberry pi to ThingSpeak platform and how to use ThingSpeak platform to analyze data and set the actuators states, then send the states data to the other raspberry pi which the actuators are on from the platform.

Sensor and Actuator

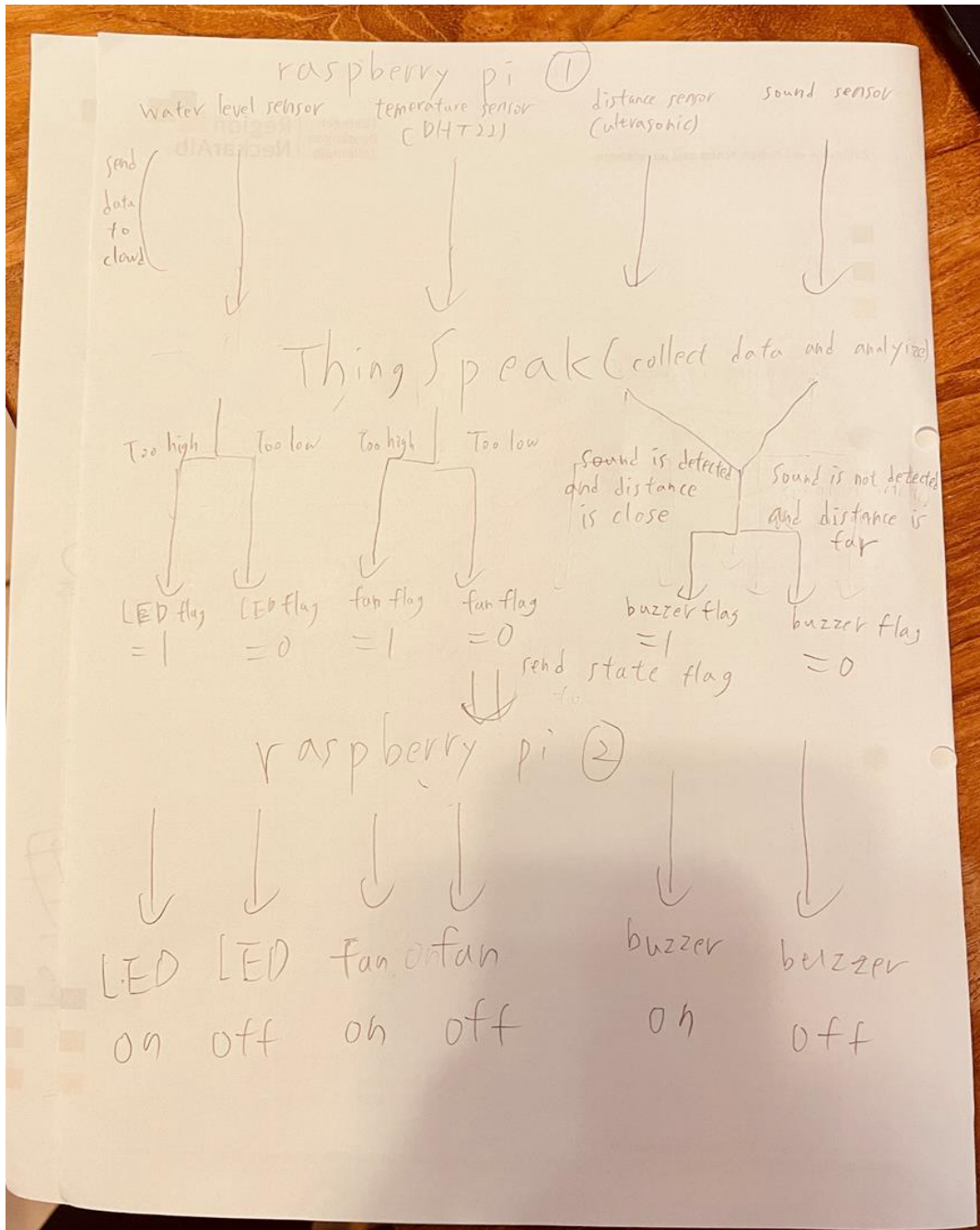
Water level sensor --- LED

Ultrasonic sensor(distance) & Im386 sound sensor--- buzzer

DHT22 --- fan

Project Design

1167 物聯網概論(英文授課) Introduction to Internet of Things



2. Diagram

Raspberry pi 1:

We set all of our sensors On this raspberry pi to collect the data(water level, temperature,distance,sound on or not) of the plant we want to monitor and upload to thingspeak.

ThingSpeak:

We read the uploaded data and set thresholds to determine the time to activate or deactivate actuators. if water level is higher than a threshold we set the LED flag on, lower we turn off; if temperature is higher than a threshold we set the fan flag on, lower we turn off; if distance is closer than a threshold and sound is detected we set the LED flag on, and if distance is further than a threshold and sound is not detected we turn off.

1167 物聯網概論(英文授課)

Introduction to Internet of Things

Raspberry pi 2:

We retrieve the flags of actuators from thingspeak and if flag =1 we turn on the actuator, and if flag =0 or none we don't turn on. This raspberry pi is to alert the owner of plant.

Source Code

1st raspberry pi code:

```
import RPi.GPIO as GPIO
import time
import Adafruit_DHT
import RPi.GPIO as GPIO
import sys
import random
import urllib.request
import requests
import threading
#import lineTool
DEVICE_TOKEN = "7aLP5TWE2gwfDn16Zyzo6fGdK"
DHT_SENSOR = Adafruit_DHT.DHT22
waterlevel = None
DHT_PIN = 17
# change these as desired - they're the pin
# SPI port on the ADC to the Cobbler
SPICLK = 11
SPIMISO = 9
SPIMOSI = 10
SPICS = 8
LM386_PIN=26
PIN=20
# photoresistor connected to adc #0
photo_ch = 0
writekey='B7N0T7Q0YC1P8NAS'
readkey='HN7KQ6WSXM04R230'

def init():
    GPIO.setwarnings(False)
    GPIO.cleanup() #clean up at the end of your script
    GPIO.setmode(GPIO.BCM) #to specify which pin numbering
    # set up the SPI interface pins
    GPIO.setup(SPIMOSI, GPIO.OUT)
    GPIO.setup(SPIMISO, GPIO.IN)
    GPIO.setup(SPICLK, GPIO.OUT)
    GPIO.setup(SPICS, GPIO.OUT)
    GPIO.setup(LM386_PIN, GPIO.IN)

def get_distance():
    TRIG = 18
    ECHO = 24
    i=0
    PIN=20
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.setup(ECHO, GPIO.IN)
    GPIO.output(TRIG, False)
    print("Starting.....")
    time.sleep(2)
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)
    while GPIO.input(ECHO)==0:
        pulse_start = time.time()
    while GPIO.input(ECHO)==1:
        pulse_stop = time.time()
    pulse_time = pulse_stop - pulse_start
    distance = pulse_time * 17150
    return distance

def sound(channel):
    print('noise')
    time.sleep(0.5)
    return 1

def get_sound():
    GPIO.add_event_detect(LM386_PIN, GPIO.RISING, callback=sound, bouncetime=10)
    return 1

def readadc(adcnun, clockpin, mosipin, misopin, cspin):
    if ((adcnun > 7) or (adcnun < 0)):
        return -1
    GPIO.output(cspin, True)
    GPIO.output(clockpin, False) # start clock on cspin
    GPIO.output(cspin, False) # bring cspin down for transmit
    commandout = adcnun
    commandout |= 0x18 # start bit + single bit mode
    commandout <= 3 # we only need to send four bits
    for i in range(5):
        if (commandout & 0x80):
            GPIO.output(mosipin, True)
        else:
            GPIO.output(mosipin, False)
        commandout <<= 1
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)
    adcout = 0
    # read in one empty bit, one null bit and 10 ADC bits
    for i in range(12):
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)
        adcout <<= 1
        if (GPIO.input(misopin)):
            adcout |= 0x1
    GPIO.output(cspin, True)
    adcout >>= 1 # first bit is 'null' so drop it
    return adcout

def get_temp():
    temp = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    return temp

def thingspeak_post_humd(humidity,distance,temp,soundd):
    URL='https://api.thingspeak.com/update?api_key='
    KEY= writekey
    HEADER='&field1={}&field2={}&field3={}&field4={}'.format(humidity,distance,temp,soundd)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)

def main():
    init()
    print("will start detec water level\n")
    j=0
    while True:
        adc_value=readadc(photo_ch, SPICLK, SPIMOSI, SPIMISO, SPICS)
        useless,temperature=get_temp()#get temp
        distance=get_distance()
        soundd=get_sound()
        print(adc_value)
        print(temperature)
        print(distance)
        print(soundd)
        thingspeak_post_humd(adc_value,distance,temperature,soundd)
        time.sleep(20)
```

1167 物聯網概論(英文授課)

Introduction to Internet of Things

2nd raspberry pi code:

```
import sys
import time
import Adafruit_DHT
import RPi.GPIO as GPIO
import random
import urllib.request
import requests
import threading
DHT_SENSOR = Adafruit_DHT.DHT22
LM386_PIN = 18
LED_PIN = 23
DHT_PIN = 21
BUZZER_PIN = 20
FAN_PIN = 14
DATA_SENDING_INTERVAL = 10
POOL_INTERVAL = 0.5
def init():
    GPIO.setwarnings(False)
    GPIO.cleanup()
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(FAN_PIN,GPIO.out)
    GPIO.setup(LED_PIN,GPIO.out)
    GPIO.setup(BUZZER_PIN,GPIO.out)
```

initialize and import library we need

```
def bell():
    C4 = 262 # Do
    E4 = 330 # Mi
    music = [C4, E4]
    M_1 = C4
    M_3 = E4
    GPIO.setup(BUZZER_PIN, GPIO.OUT)
    p = GPIO.PWM(BUZZER_PIN, 50)
    p.start(15) # 0 <= DV <= 100
    play(p, M_3, 1)
    play(p, M_1, 1)
    p.stop
def play(p, frequency, tempo):
    p.ChangeFrequency(frequency)
    time.sleep(0.5 * tempo)
```

two functions that make buzzer work

```
def read_data_led():
    NEW_URL = 'https://api.thingspeak.com/channels/1625035/fields/5.json?api_key=HN7KQ6WSXMO4R23Q&results=2'
    get_data = requests.get(NEW_URL).json()
    channel_id = get_data['channel']['id']
    feild_l = get_data['feeds']
    t = []
    for x in feild_l:
        t.append(x['field5'])
    return(t[1])
def read_data_buzzer():
    NEW_URL = 'https://api.thingspeak.com/channels/1625035/fields/6.json?api_key=HN7KQ6WSXMO4R23Q&results=2'
    get_data = requests.get(NEW_URL).json()
    channel_id = get_data['channel']['id']
    feild_l = get_data['feeds']
    t = []
    for x in feild_l:
        t.append(x['field6'])
    return(t[1])
def read_data_fan():
    NEW_URL = 'https://api.thingspeak.com/channels/1625035/fields/7.json?api_key=HN7KQ6WSXMO4R23Q&results=2'
    get_data = requests.get(NEW_URL).json()
    channel_id = get_data['channel']['id']
    feild_l = get_data['feeds']
    t = []
    for x in feild_l:
        t.append(x['field7'])
    return(t[1])
```

3 functions that get value of led,buzzer and fan(which are either true or false)

1167 物聯網概論(英文授課) Introduction to Internet of Things

```
def main():
    init()
    fan=read_data_fan()
    led=read_data_led()
    buzzer=read_data_buzzer()
    print(fan)
    print(led)
    print(buzzer)
    time.sleep(0.5)
    if fan == '1':
        GPIO.output(FAN_PIN,True)
        print('fan on')
    else:
        GPIO.output(FAN_PIN,False)
        print('fan off')
    if led == '1':
        GPIO.output(LED_PIN,True)
        print('led on')
    else:
        GPIO.output(LED_PIN,False)
        print('led off')
    if buzzer == '1':
        bell()
        print('buzzer on')
    else:
        print('buzzer off')
while True:
    main()
    time.sleep(5)
```

main function is to
get fan,led,and
buzzer value and
then turn on/off
actuators

MATLAB code:

```
readkey='HN7KQ6WSXM04R23Q';
writekey='B7N0T7Q0YC1P8NAS';
```

set up keys and
channel id

```
readChannelID = 1625035;
writeChannelID=1625035;
```

```
i=0;j=0;k=0;
```

```
humi= thingSpeakRead(readChannelID,'ReadKey', readkey, 'Fields', 1);
distance=thingSpeakRead(readChannelID,'ReadKey', readkey, 'Fields', 2);
temperature=thingSpeakRead(readChannelID,'ReadKey', readkey, 'Fields', 3);
sound=thingSpeakRead(readChannelID,'ReadKey', readkey, 'Fields', 4);
```

get value from
thingspeak(from
field1~field4)

```
if humi>400
```

```
    i=1
```

```
else
```

```
    i=0
```

```
end
```

```
if distance<10 && sound==1
```

```
    j=1
```

```
else
```

```
    j=0
```

```
end
```

```
if temperature>28
```

```
    k=1
```

```
else
```

```
    k=0
```

```
end
```

```
thingSpeakWrite(writeChannelID, [i,j,k],'WriteKey', writekey, 'Fields', [5,6,7]);
```

i,j,k representing actuators' true
false , after that we post the
value on thingspeak at once

1167 物聯網概論(英文授課)
Introduction to Internet of Things

Demo video

<https://youtu.be/LlgxRJ0D5xc>