

1167 物聯網概論(英文授課)
Introduction to Internet of Things
Assignment 3 Report

Student Information

- 109550164 徐聖哲
- 109550073 陳宥安

Objectives

In the assignment 3, we integrate 2 RPI together to get information from the environment and upload collected data to ThingSpeak, which can store history data and show them on the correspondent table.

We try to simulate a greenhouse sensor system. Inside the system, the RPI will collect temperature, humidity, and lightness(LDR) continuously. On the other hand, when a person passes in front of the PIR sensor, the RPI will know it and upload it to ThingSpeak. Then ThingSpeak will do cloud computing and decide whether to ring the buzzer on the second RPI, emulating the intruder detection system. Furthermore, when the RFID reader reads the RFID card, it will also upload the signal to ThingSpeak and control the LED on the second RPI to emulate the access control system.

With the system above, we are able to manage the environment inside the greenhouse, creating a suitable place for the plant to grow faster in order to maximize the production.

Sensor and Actuator

PIR -> for Motion Detection

LED -> for informing Users start collect data

DHT-22 -> for getting real-time Temperature and Humidity

FAN -> for responding to the change of temperature or humidity

LDR -> for detecting the change of light

Buzzer -> for Reminding and Alerting Users

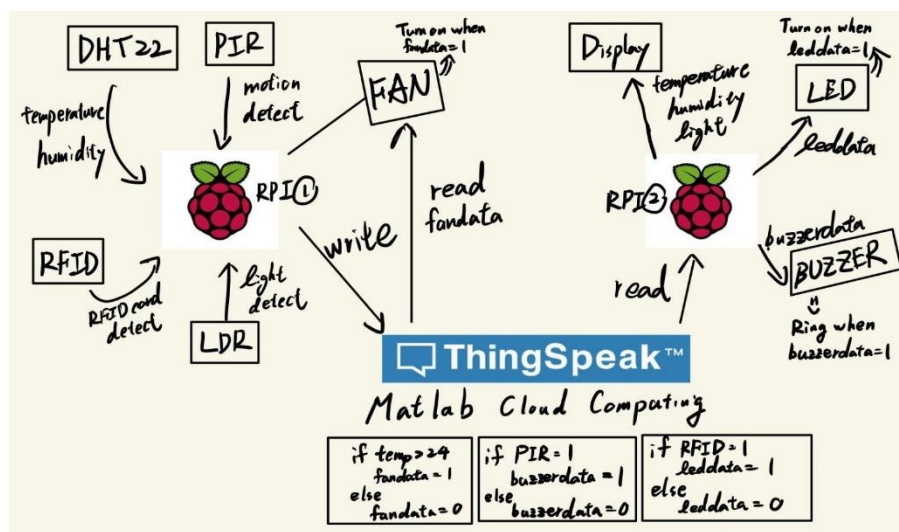
Display -> for Displaying Data collecting from the sensors

Demo Video Link

<https://youtu.be/oCPafioF0Vg>

Project Design

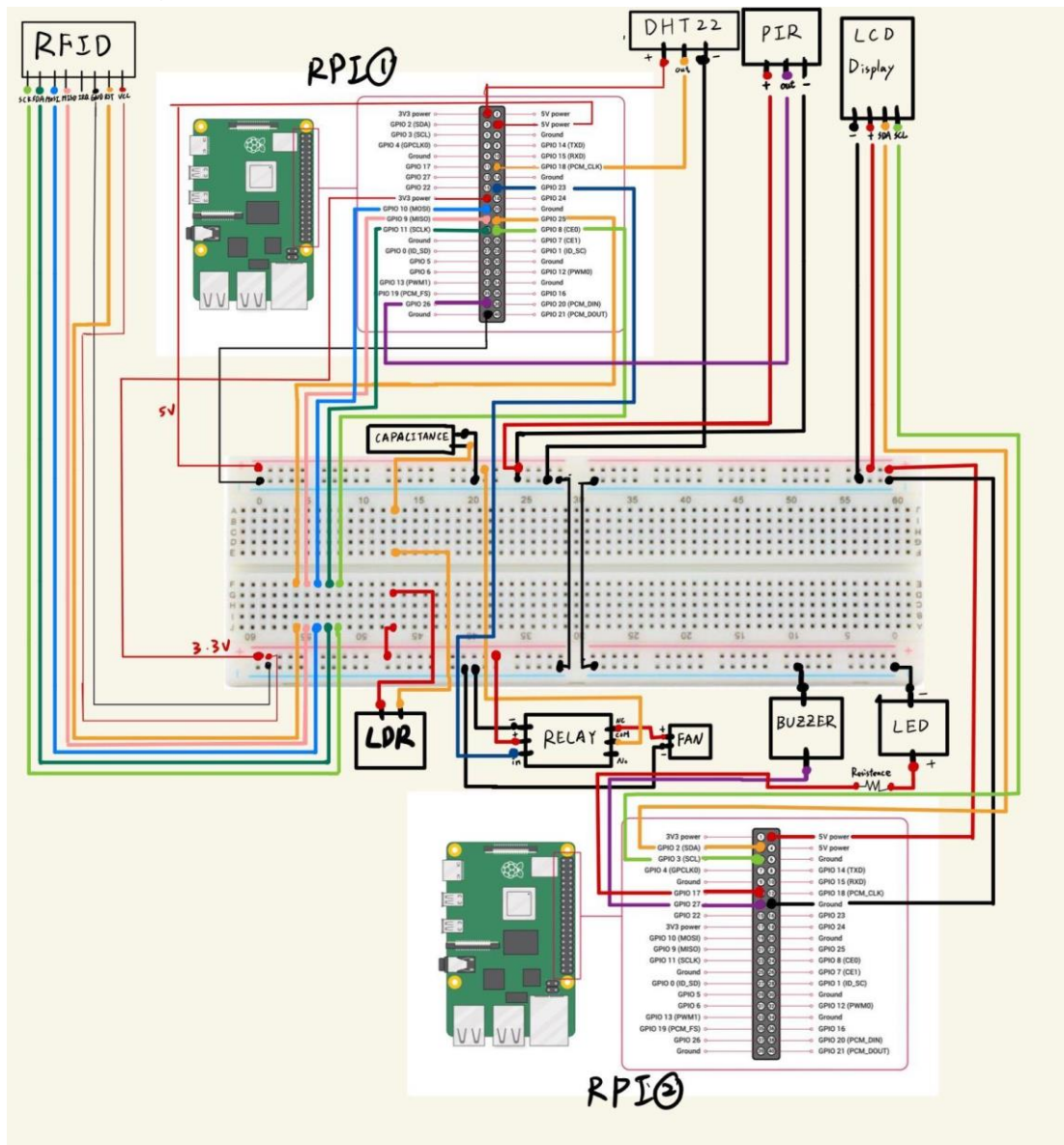
1. Flowchart Explanation



1167 物聯網概論(英文授課)

Introduction to Internet of Things

2. Circuit Diagram



Source Code

1. ThinkSpeak MATLAB Code Explanation

The code below controls the data updating on ThingSpeak and computes data on cloud.

```

1 % Enter your MATLAB Code below
2 % TODO - Replace the [] with channel ID to read data from:
3 readChannelID = 1608288;
4 % TODO - Enter the Read API Key between the '' below:
5 readAPIKey = '0SRDOB2ZRGWB47CB';
6
7 % TODO - Replace the [] with channel ID to read data from:
8 writeChannelID2 = 1625967;
9 writeAPIKey2 = 'ID4XQPE3VBPXY01';
10
11 %% Read Data %%
12 temp1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 1);
13 humidity1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 2);
14 motion1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 3);
15 rfid1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 4);
16 ldr1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 5);
17 disp(temp1);
18 disp(ldr1);
19 %% Analyze Data %%
20 % Add code in this section to analyze data and store the result in the
21 % 'analyzedData' variable.
22 modedata = 0;
23 ldrdata = ldr1;
24
25 if(temp1 > 24)
26     fandata = 1;
27 else
28     fandata = 0;
29 end
30
31 if(rfid1 == 1)
32     leddata = 1;
33 else
34     leddata = 0;
35 end
36
37 if(motion1 == 1)
38     pirdata = 1;
39 else
40     pirdata = 0;
41 end
42
43 %% Write Data %%
44 %second rpi%
45 thingSpeakWrite(writeChannelID2, [fandata, leddata, pirdata, ldrdata], 'WriteKey', writeAPIKey2);

```

Introduction to Internet of Things

The first RPI is for reading, so we imply “thingSpeakRead” with readchannelID, readAPIKey and the correct field we want to show the data. The second RPI is for writing, so we imply “thingSpeakWrite” with writechannelID, writeAPIKey and the correct field on ThingSpeak. We put the output data as an array then write it in the field. In the middle part, we calculate the fanddata with temperature, leddata with rfid value and pirdata with motion value.

2. Python Source Code Explanation

In this assignment, ThingSpeak is the specified cloud platform.

(a) FIRST RPI:

```
import adafruit_dht
import board
import time
import RPi.GPIO as GPIO
import urllib.request
import requests
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()
dht_device = adafruit_dht.DHT22(board.D18, use_pulseio = False) # DHT22 Initialization

GPIO.setwarnings(False)
GPIO.setup(26, GPIO.IN) # PIR Initialization
GPIO.setup(23, GPIO.OUT, initial=GPIO.LOW) # Fan initialization
GPIO.setup(21, GPIO.IN) # LDR initialization

def thingspeak_post_temphumd(t, h, m, r, l):
    URL='https://api.thingspeak.com/update?api_key='
    KEY='P2U9EJ5MPC39SESS'
    HEADER='&field1={}&field2={}&field3={}&field4={}&field5={}'.format(t, h, m, r, l)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)
```

The above code indicates import libraries and the initialization of different sensors. The function thingspeak_post_temphumd() updated all data on the first RPI to ThingSpeak with different fields. Key indicates the channel API key, and connect to HTTP.

```
def read_data_fan():
    URL='https://api.thingspeak.com/channels/1625967/fields/1.json?api_key='
    KEY='FYJ6KK8XYZ7ZCQ8V'
    HEADER='&results=1'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field1']
```

The function read_data_fan() reads the data from ThingSpeak and uses get_data to control the fan. We also need the API key to have the HTTP connection.

```
def ldr():
    count = 0
    GPIO.setup(21, GPIO.OUT) #Output on the pin
    GPIO.output(21, GPIO.LOW)
    time.sleep(0.1)
    GPIO.setup(21, GPIO.IN) #Change the pin back to input
    while (GPIO.input(21) == GPIO.LOW): #Count until the pin goes high
        count += 1
    return count
```

The function ldr() detects the light and returns the number it counts.

```
def fan(data):
    if data == 1:
        GPIO.output(23, GPIO.LOW)
        return 1
    else:
        GPIO.output(23, GPIO.HIGH)
        return 0
```

Introduction to Internet of Things

The function fan() returns the fan_state in boolean value

```
def read():
    global humidity, temp
    try:
        humidity= dht_device.humidity
        temp = dht_device.temperature
    except(RuntimeError):
        print("no new data this loop!")
    return temp, humidity
def pir():
    # Return PIR data
    i = GPIO.input(26)
    if i == 1:
        print("someone is here!!!")
        return i
    else:
        return i
```

The function read() gets data from DHT-22, while the function pir() returns the state true when the pir detects motion.

```
def readrfid():
    readornot = 0
    id = reader.read_id_no_block()
    if id != None:
        readornot = 1
        print("rfid detected!!!")
        return readornot
    return readornot
```

The function read rfid() returns the state true when the rfid detects an RFID card.

```
while True:
    try:
        tem, hum = read()
        p = pir()
        rfid = readrfid()
        l = ldr()
        curf = read_data_fan()
        thingspeak_post_temphumd(tem, hum, p, rfid, l)
        fop = fan(float(curf))
        time.sleep(5)
        rfid = readrfid()
        time.sleep(5)

    except KeyboardInterrupt:
        # If there is a KeyboardInterrupt (when you press ctrl+c), exit the program and cleanu
        print("Cleaning up!")
        GPIO.output(23, GPIO.LOW)
        GPIO.cleanup()
        exit(1)
```

The code above is the main function, which collect the data and upload it to the cloud.

(b) SECOND RPI

```
import drivers
import requests
import sys
import time
import RPi.GPIO as GPIO
global display
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT, initial=GPIO.LOW) #LED
GPIO.setup(27, GPIO.OUT, initial=GPIO.LOW) #BUZZER
def read_data_led():
    URL='https://api.thingspeak.com/channels/1625967/fields/2.json?api_key='
    KEY='FYJ6KK8XYZ7ZCQBV'
    HEADER='&results=2'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field2']
def read_data_buzzer():
    URL='https://api.thingspeak.com/channels/1625967/fields/3.json?api_key='
    KEY='FYJ6KK8XYZ7ZCQBV'
    HEADER='&results=3'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field3']
```

The above code indicates import libraries and the initialization of different sensors. The functions above read_data_led(), read_data_buzzer() read data collected from the first RPI on ThingSpeak we computed with field 2 and 3.

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

```
def read_data_temp():
    URL='https://api.thingspeak.com/channels/1608288/fields/1.json?api_key='
    KEY='0SRD0B2ZRGWB47CB'
    HEADER='&results=1'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field1']

def read_data_humd():
    URL='https://api.thingspeak.com/channels/1608288/fields/2.json?api_key='
    KEY='0SRD0B2ZRGWB47CB'
    HEADER='&results=2'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field2']

def read_data_ldr():
    URL='https://api.thingspeak.com/channels/1625967/fields/4.json?api_key='
    KEY='FYJ6KK8XYZ7ZCQ8V'
    HEADER='&results=4'
    NEW_URL=URL+KEY+HEADER
    get_data = requests.get(NEW_URL).json()
    print(get_data)
    return get_data['feeds'][0]['field4']
```

The functions above read_data_temp(), read_data_humd(), read_data_ldr() read data collected from the first RPI on thingspeak with field 1, 2, and the computed data from field 4.

```
def displaystring(temp, humd, ldr):
    display lcd_display_string("-----", 1)
    display lcd_display_string("Temp="+temp+"*C", 2)
    display lcd_display_string("Humidity="+humd+"%", 3)
    display lcd_display_string("ldr="+ldr, 4)

def lcd(a):
    print("a=", a)
    if a=='1':
        GPIO.output(17, GPIO.HIGH)
        print("rfid detect!!")
    else:
        GPIO.output(17, GPIO.LOW)
        print("no card yet.....")

def buzzer(b):
    print("b=", b)
    if b=='1':
        GPIO.output(27, GPIO.HIGH)
        print("buzzing!!!")
        time.sleep(1)
        GPIO.output(27, GPIO.LOW)
```

The function lcd() and buzzer() decide whether to turn the led on or ring the buzzer by the computed data we read from ThingSpeak. displaystring() displays the temperature, humidity and lightness data from ThingSpeak.

```
try:
    display = drivers.Lcd()
    display lcd_display_string("Hello from Cloud4RPI", 2)
    time.sleep(2)
    display lcd_clear()
    while (True):
        t = read_data_temp()
        h = read_data_humd()
        ldr = read_data_ldr()
        lcd(read_data_led())
        buzzer(read_data_buzzer())
        displaystring(t, h, ldr)
        time.sleep(5)
except KeyboardInterrupt:
    print("Cleaning up!")
    GPIO.output(17, GPIO.LOW)
    GPIO.output(27, GPIO.LOW)
    display lcd_clear()
```

The code above is the main function, which reads the data from the cloud.

Thoughts & Comments

Through the project, we learned how to combine two RPI with ThingSpeak, which is a cloud computing platform. However, it is hard to write and learn MATLAB in such a short time. Plus, ThingSpeak has lots of constraints that make MATLAB even harder. We suggest that ThingSpeak should not be used in the future in this course.