Assignment 2 – IoT with Cloud4RPi

Introduction to IoT

Fall 2021

Outline

- Lab Objectives
- Cloud4RPi
- How to use Cloud4RPi?
    - Create an account
    - Create a device
    - Setup control panel
- Cloud4RPi Applications
    - Store temperature data to Cloud4RPi (Send Data)
    - Control LED with Cloud4RPi (Retrieve Data)
    - Automatically generate an alert in Cloud4RPi
- Assignment 2 Specifications

Lab Objectives

- Connect Raspberry Pi to Cloud4RPi
- Capable to store the sensor data in Cloud4RPi
    - Use cloud services to collect data from sensors
        - Monitor DHT sensor from Cloud4RPi
    - Use cloud services to trigger reactions to actuators
        - Turn on LED from Cloud4RPi
- Capable to make a computing program in Cloud4RPi
    - Use cloud services to make decisions
        - Automatically generate alerts to user from Cloud4RPi

What is Cloud4RPi?

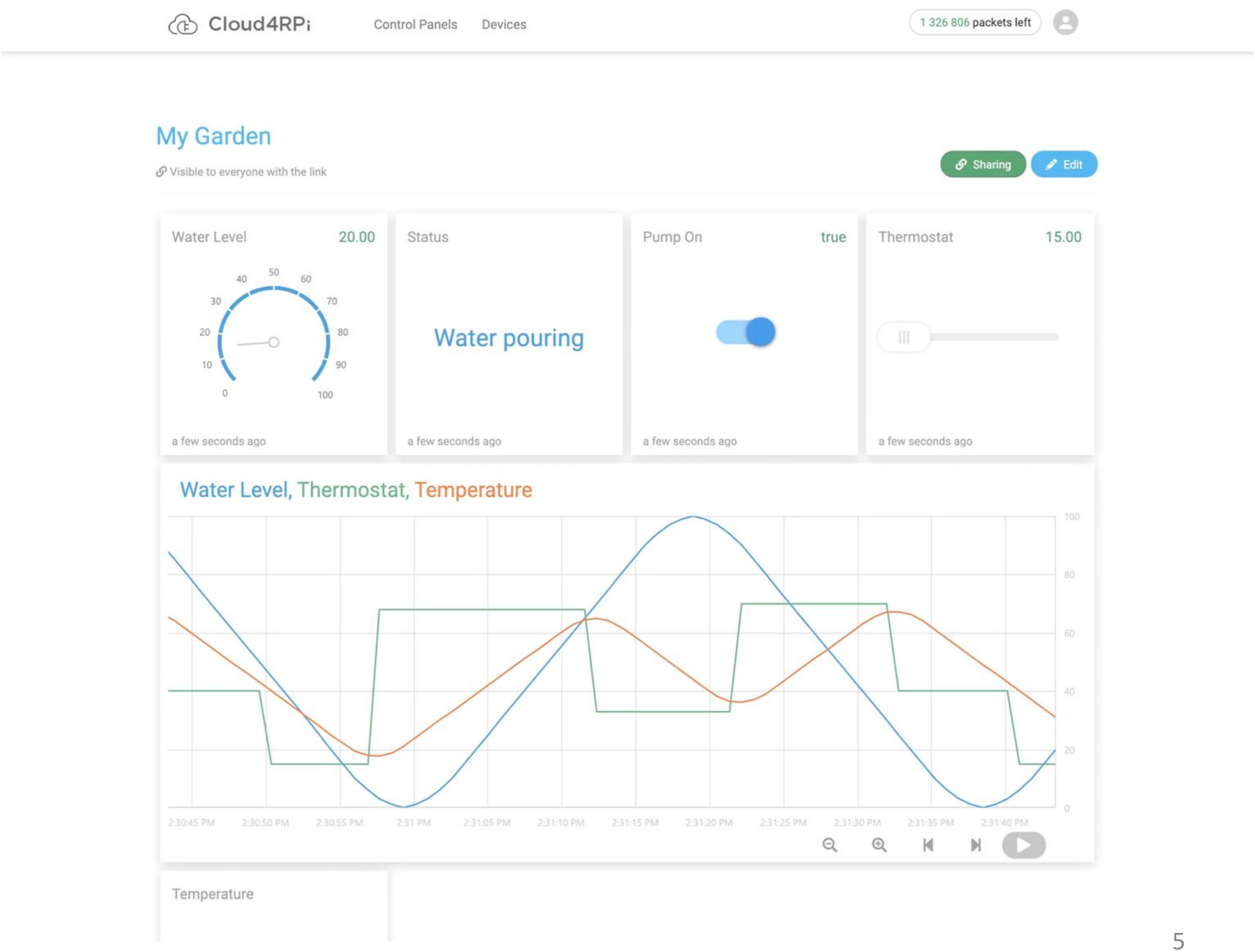
- Cloud based data service platform for IoT devices.

CLOUD CONTROL PANEL FOR RASPBERRY PI AND OTHER IOT DEVICES

Use dashboard widgets to display device data and send commands in real time. Control your IoT devices remotely.

Cloud4RPi Features (1/4)

Widgets to display real-time data



My Garden

Visible to everyone with the link

Water Level 20.00

a few seconds ago

Status

Water pouring

a few seconds ago

Pump On true

a few seconds ago

Thermostat 15.00

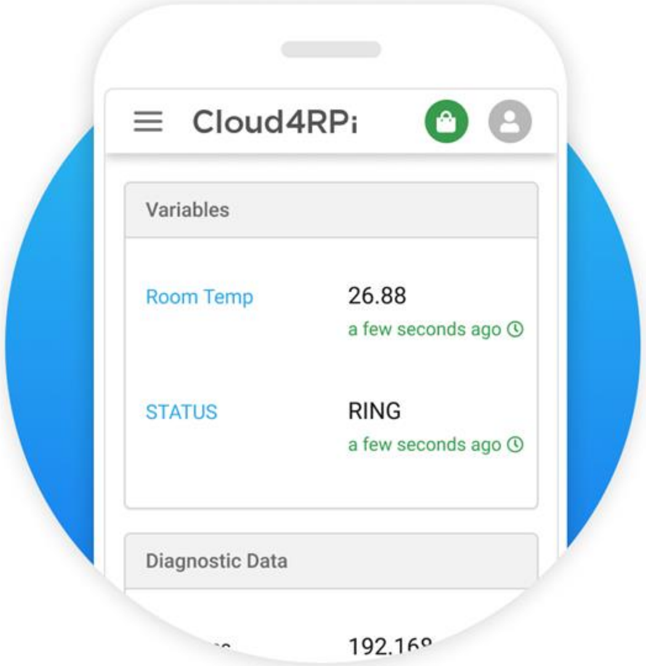
a few seconds ago

Water Level, Thermostat, Temperature

Temperature

https://cloud4rpi.io

Cloud4RPi Features (2/4)



Variables

Room Temp 26.88
a few seconds ago

STATUS RING
a few seconds ago

Diagnostic Data

192.168

Communicate with your devices in real-time

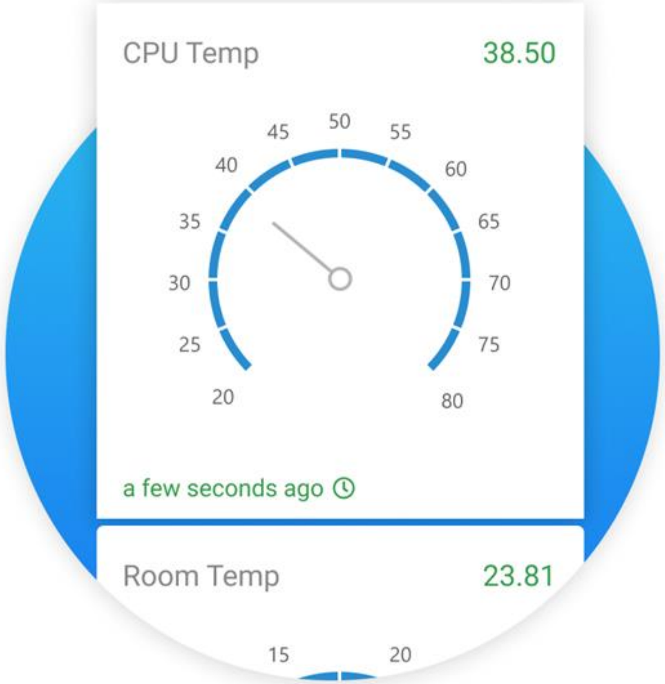
Check your country house thermometer from your mobile device and turn on the heating to prepare it for your arrival.

https://mcs.mediatek.com/

Cloud4RPi Features (3/4)

Control all your devices in one place using widgets and customizable dashboards

You have a full-fledged smart home setup with smart lightbulbs, thermostats, and sensors of all sorts? Use a single compact control panel to inspect and control all these devices.



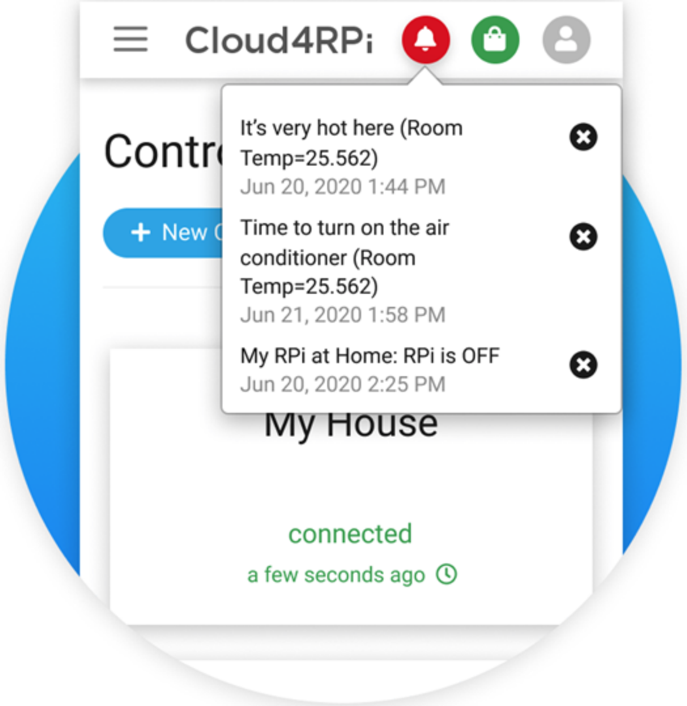
CPU Temp 38.50

a few seconds ago

Room Temp 23.81

https://mcs.mediatek.com/

Cloud4RPi Features (4/4)



It's very hot here (Room Temp=25.562)
Jun 20, 2020 1:44 PM

Time to turn on the air conditioner (Room Temp=25.562)
Jun 21, 2020 1:58 PM

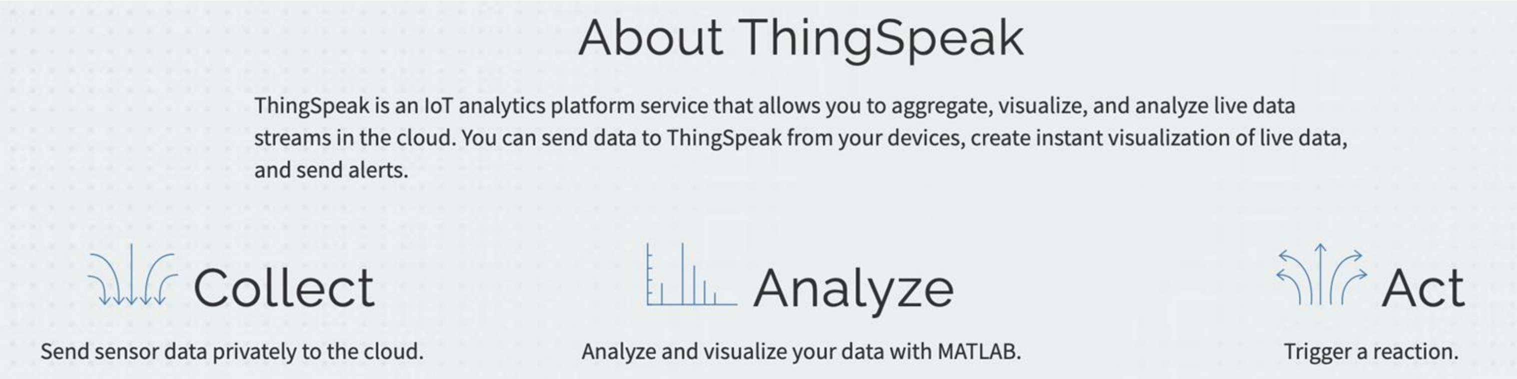
My RPi at Home: RPi is OFF
Jun 20, 2020 2:25 PM

Get notified when something important happens with your device

Don't miss critical events. Your hamster is out of drinking water or it is too dark for your orchid's comfort? You will immediately receive a notification.

https://mcs.mediatek.com/

Other Platforms



About ThingSpeak

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.

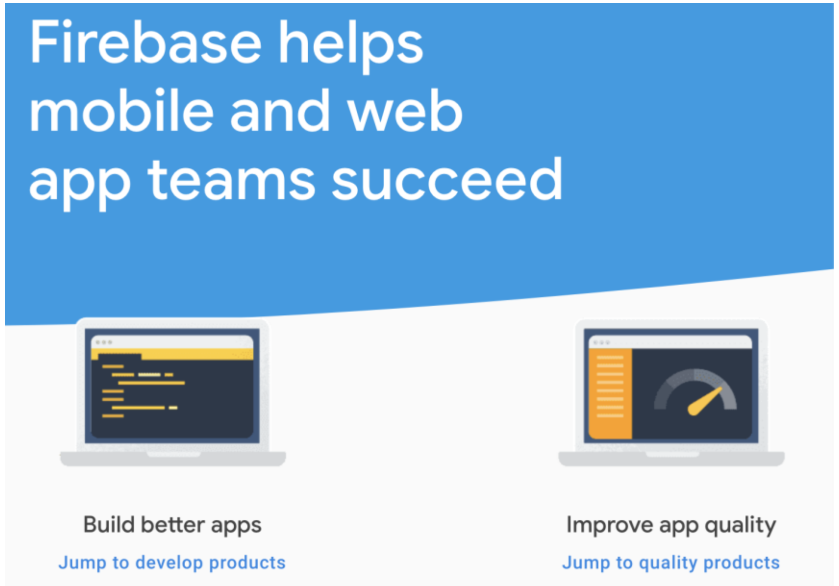
Collect
Send sensor data privately to the cloud.

Analyze
Analyze and visualize your data with MATLAB.

Act
Trigger a reaction.

AWS IoT

IoT services for industrial, consumer, and commercial solutions

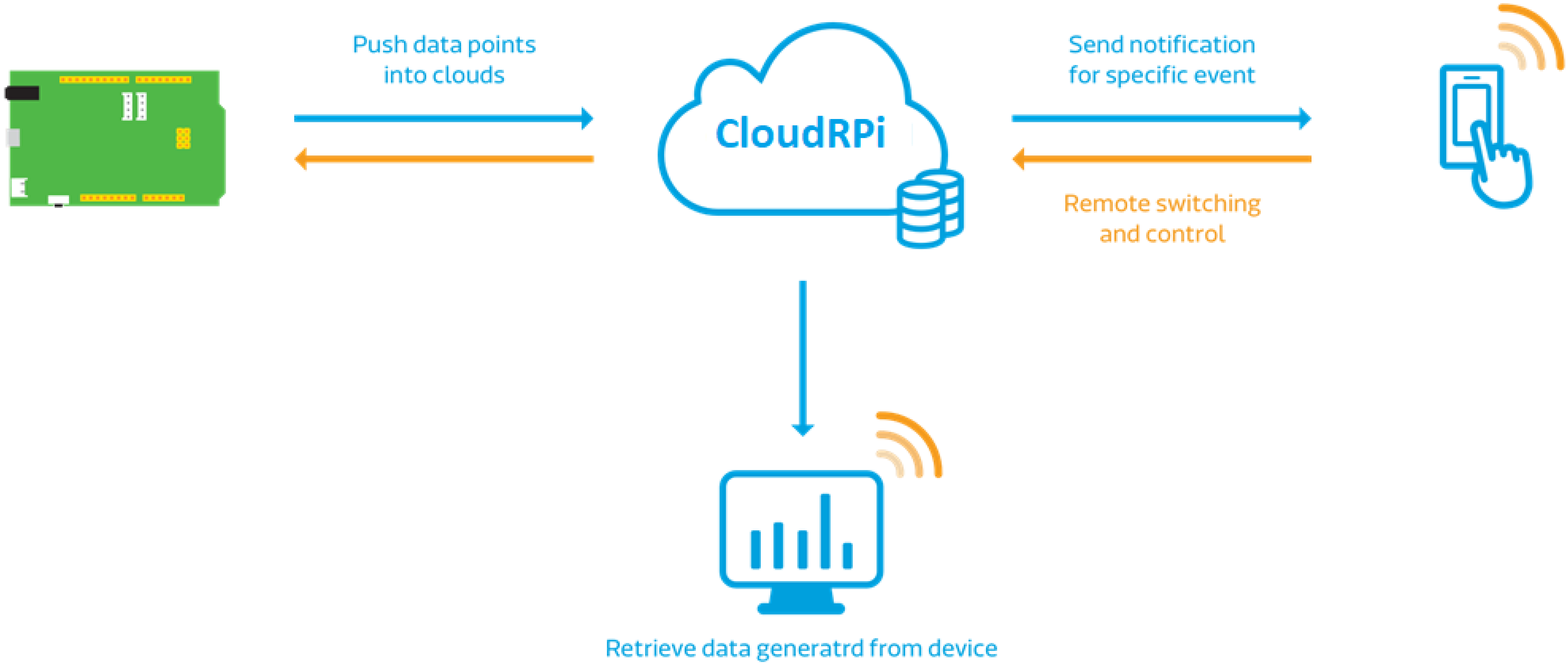


Firebase helps mobile and web app teams succeed

Build better apps
Jump to develop products

Improve app quality
Jump to quality products

Cloud4RPi – Raspberry Architecture



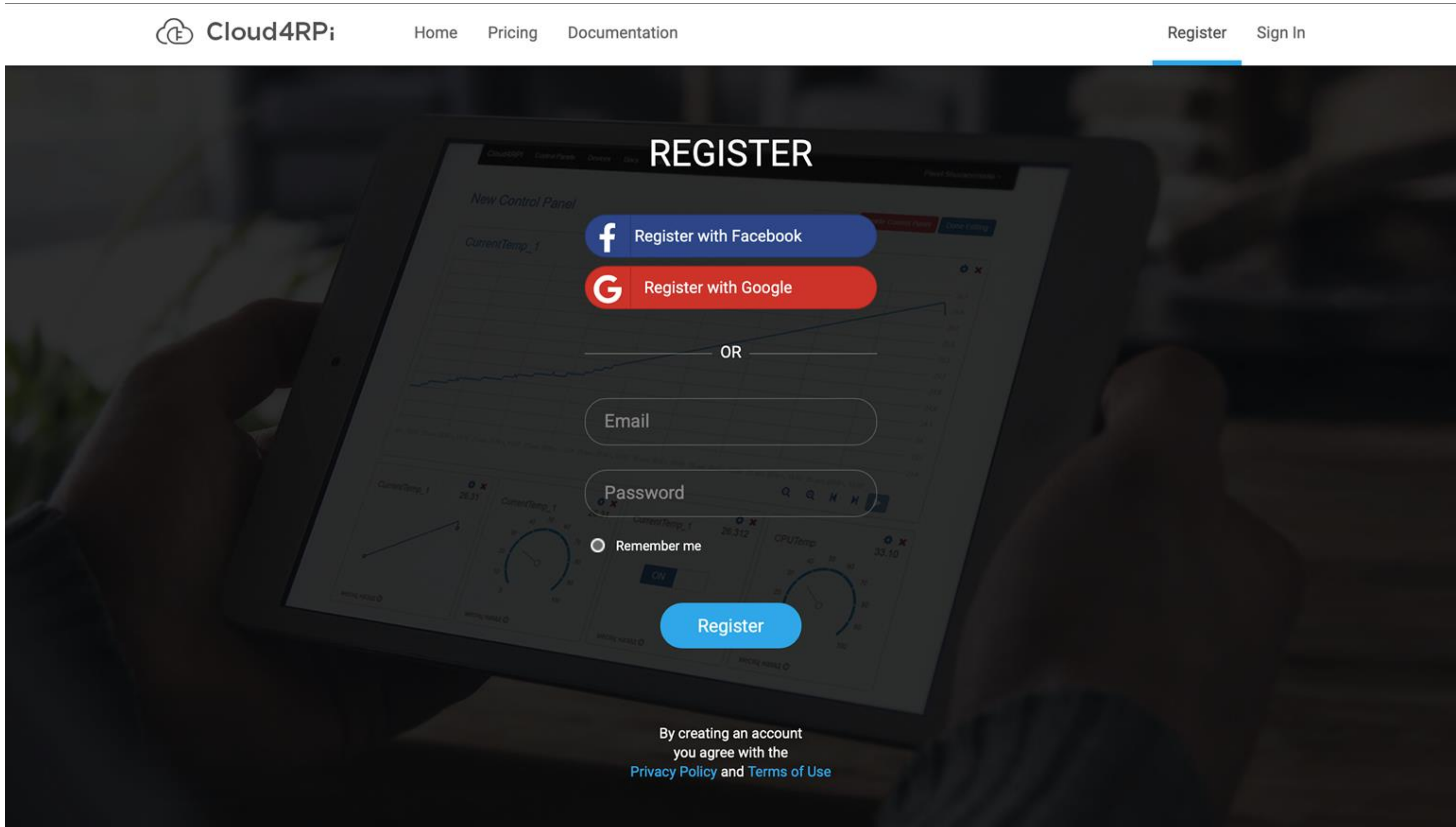
How to Use the Cloud4RPi ?

- Register an account (https://cloud4rpi.io/).
- Verify and complete the registration process
- Configure several things:
  - Add a new device,
  - Get device token,
  - Connect the device to Cloud4RPi
  - Set the environment
- Start to manage via:
  - Web console

Cloud4RPi – Create an Account

Click the follow link to create an account and sign in:

https://cloud4rpi.io/register



REGISTER

Email

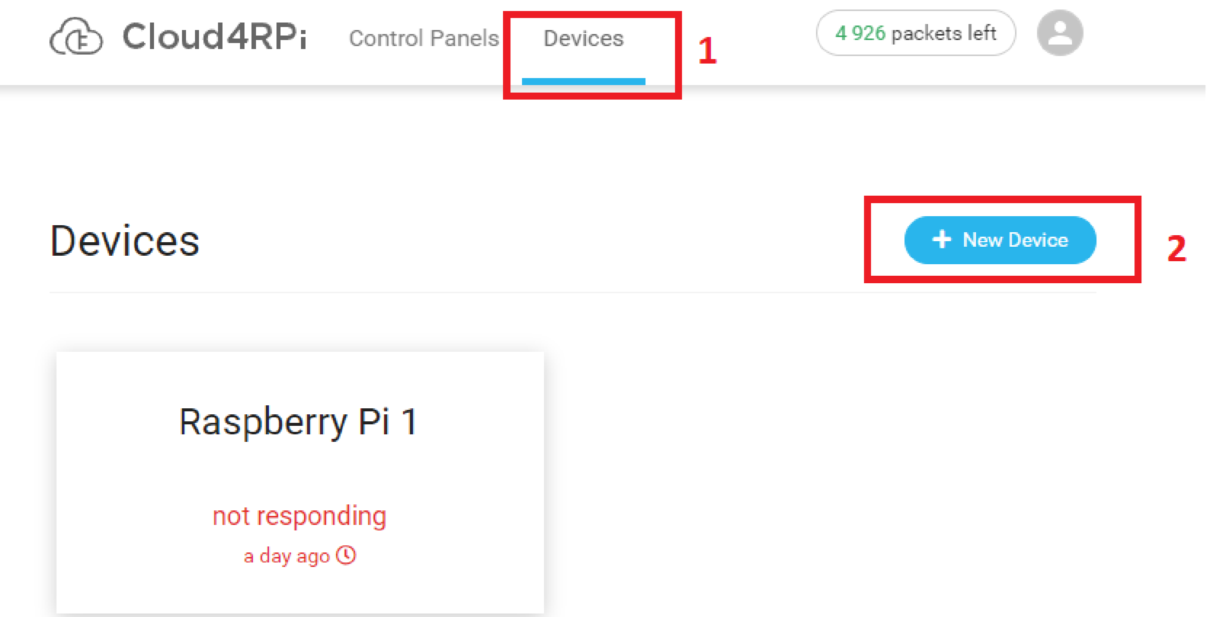
Password

○ Remember me

By creating an account you agree with the Privacy Policy and Terms of Use

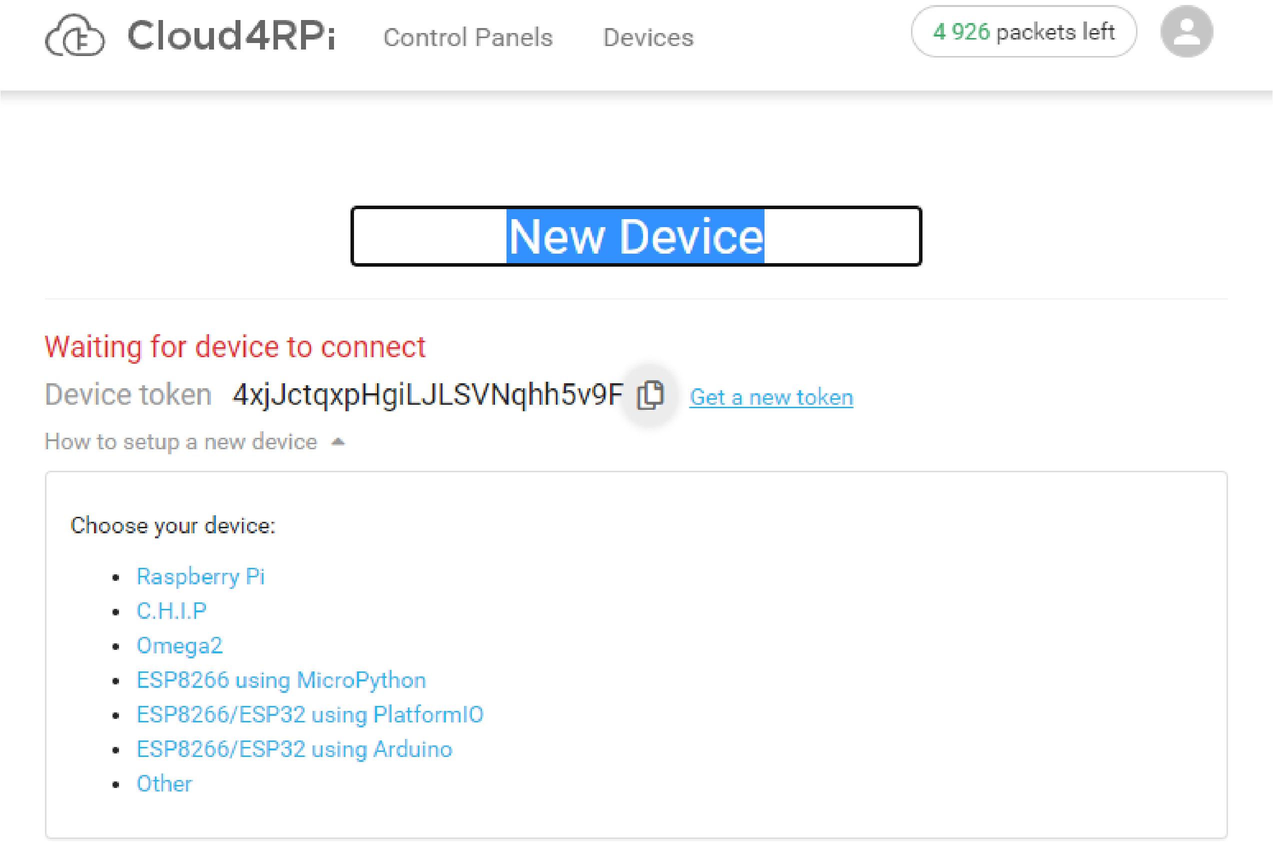
Cloud4RPi – Create a New Device (1/2)

- Dashboard -> Devices
- Add a new device
- Follow the setting on next page



Cloud4RPi – Create a New Device (2/2)

- Give a name to your device
- Save your device token
- Then, we move to your Raspberry Pi to connect it into Cloud4RPi



New Device

Waiting for device to connect

Device token 4xjJctqxpHgiLJLSVNqhh5v9F

How to setup a new device

Choose your device:

- Raspberry Pi
- C.H.I.P
- Omega2
- ESP8266 using MicroPython
- ESP8266/ESP32 using PlatformIO
- ESP8266/ESP32 using Arduino
- Other

Cloud4RPi – Connect the Device (1/2)

- Install the library in your Raspberry using your preferred Python version.
- Python 3
  - sudo pip3 install cloud4rpi
- Python 2
  - sudo python2 –m pip install cloud4rpi

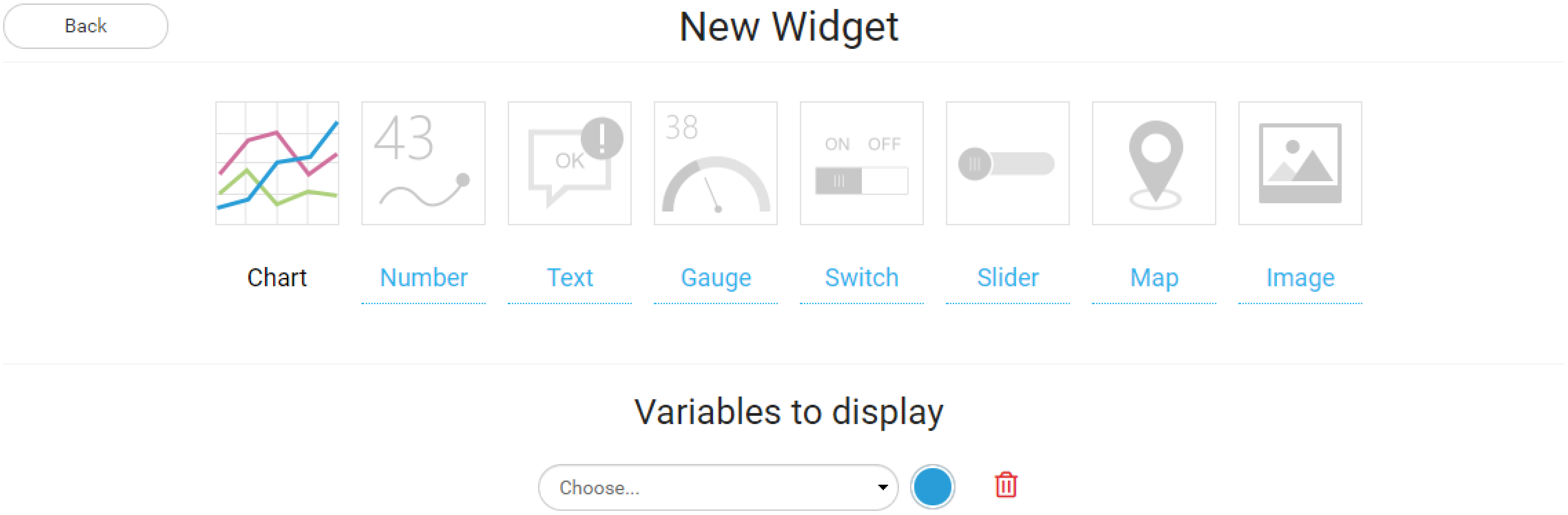
Cloud4RPi – Connect the Device (2/2)

- Basic script to connect raspberry and send data to Cloud4RPi

```
import cloud4rpi

DEVICE_TOKEN = ”YOUR_DEVICE_TOKEN”

temp, hum = None, None

def update_data():
    global hum, temp
    hum, temp = Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, DHT_PIN)
def get_t():
    update_data()
    return round(temp, 2) if temp is not None else None
def get_h():
    update_data()
    return round(hum, 2) if hum is not None else None

def main():
     variables = {
              ‘Room Temp’: {
                    ‘type’: ‘numeric’,
                    ‘bind’: get_t
              },
              ‘Room Humidity’: {
                    ‘type’: ‘numeric’,
                    ‘bind’: get_h
              }
     }
     device = cloud4rpi.connect(DEVICE_TOKEN)
     device.declare(variables)
     device.publish_config()
     Device.publish_data()
```

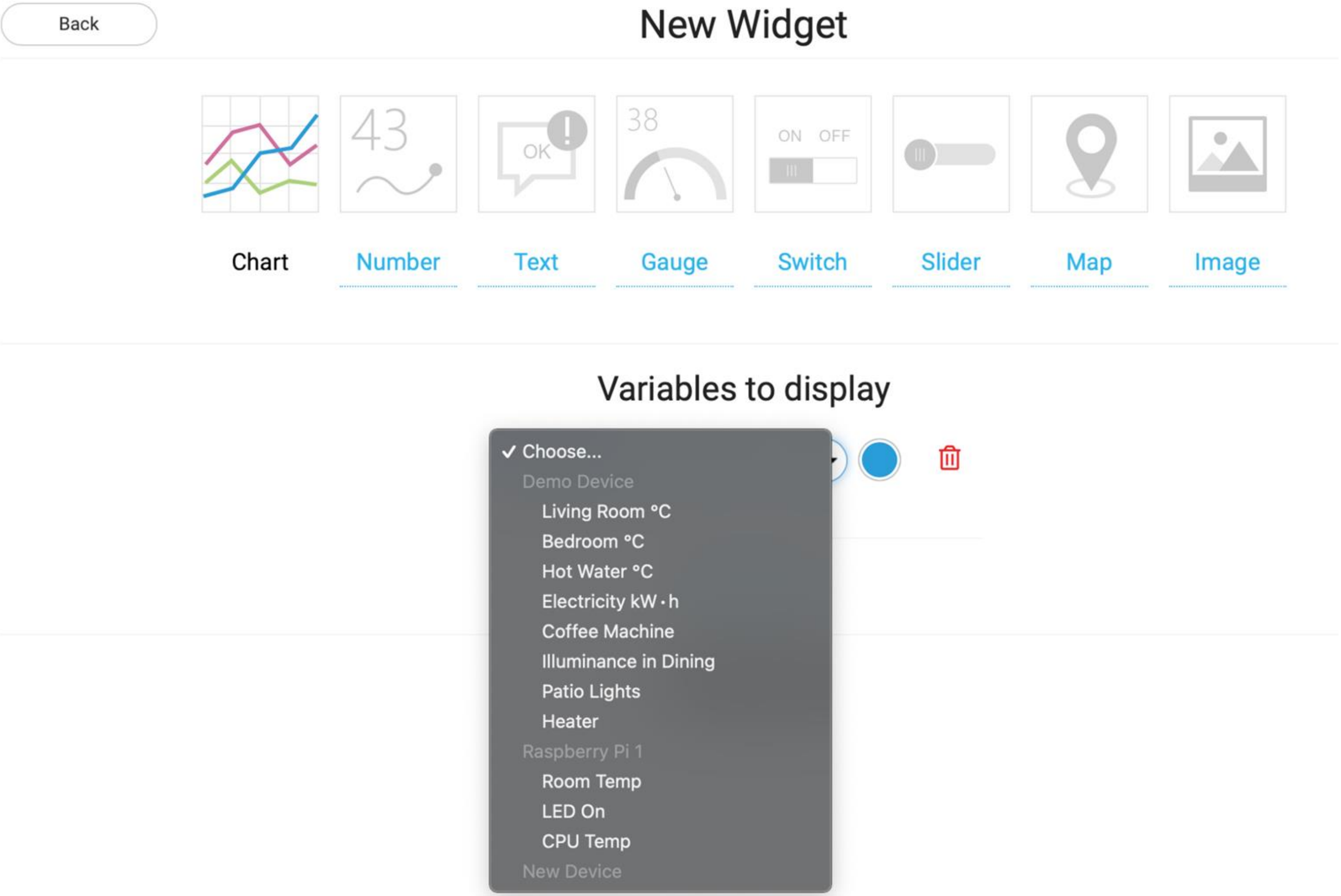
Cloud4RPi – Setup Control Panel (1/3)

- Choose the Widget



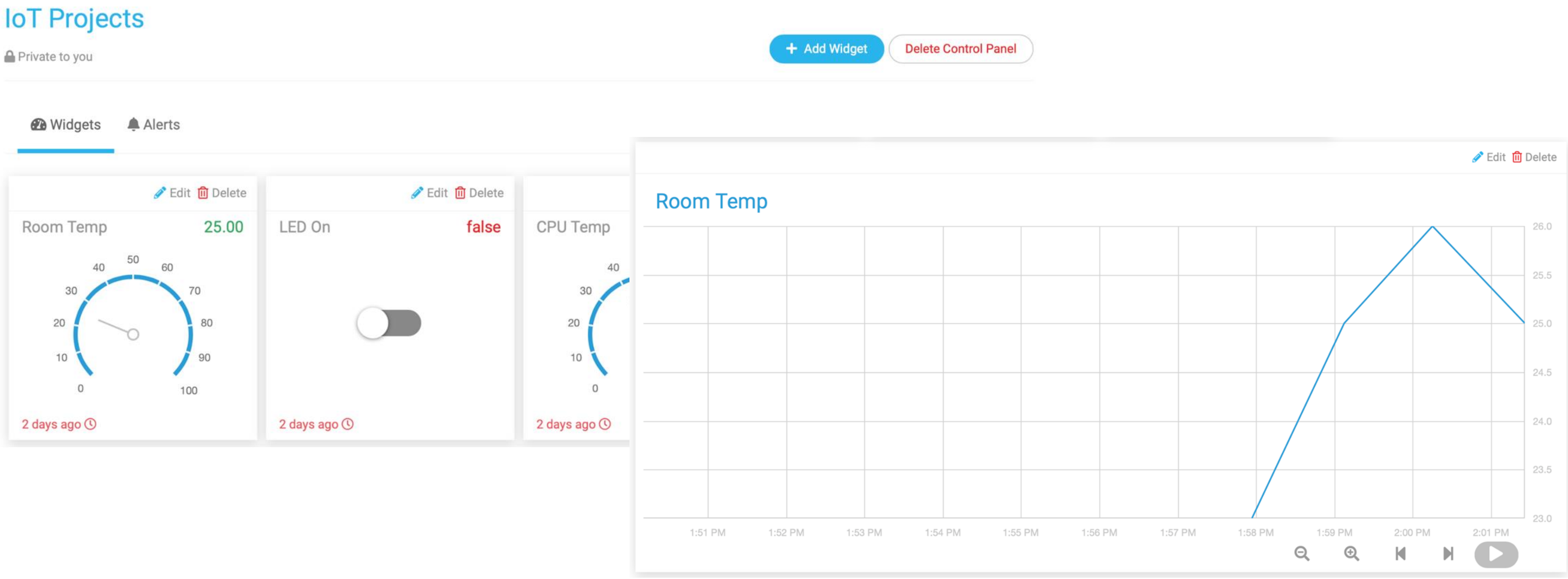
Cloud4RPi – Setup Control Panel (2/3)

- Choose the variables to display



Cloud4RPi – Setup Control Panel (3/3)

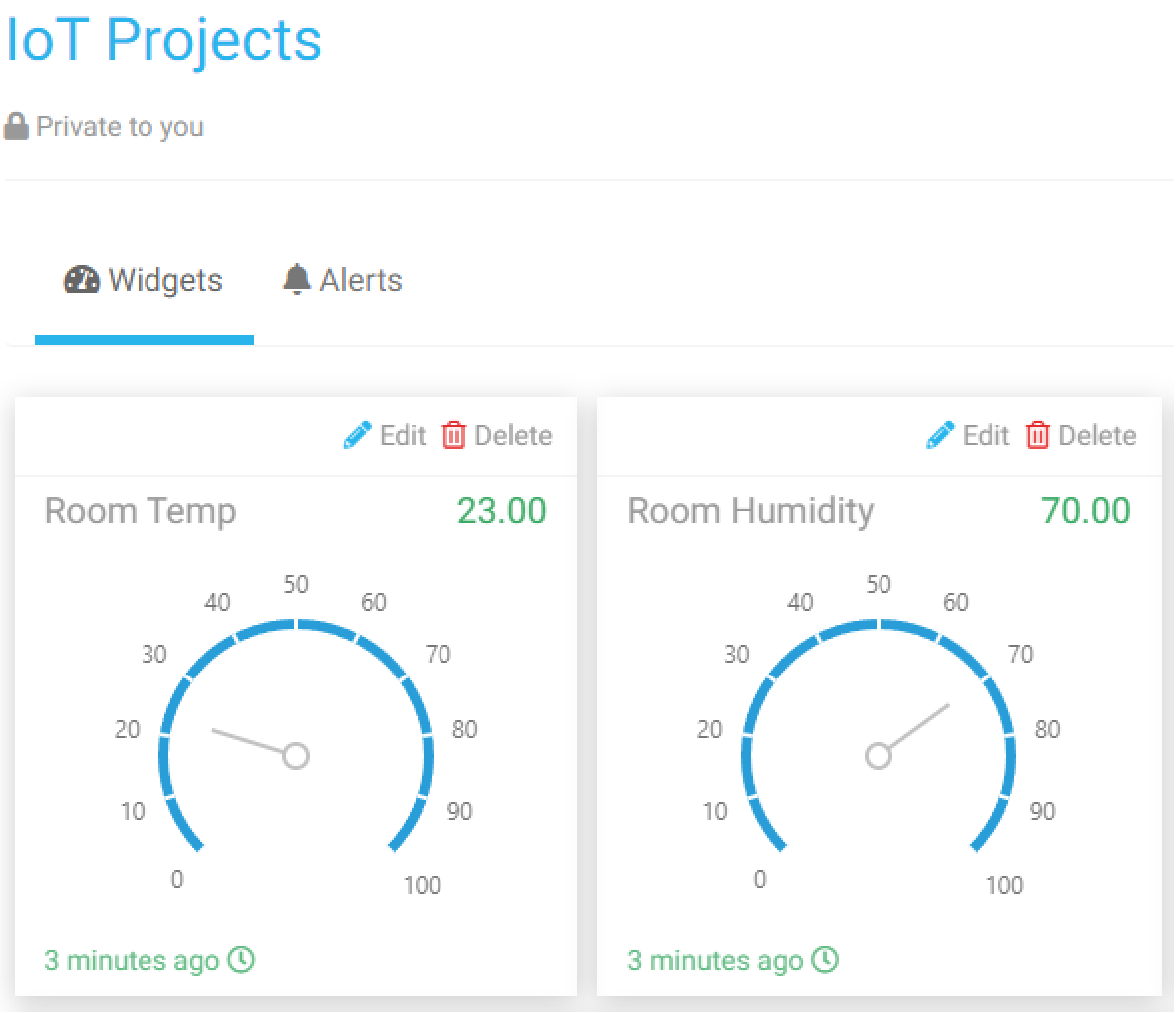
- Control Panel Example



Cloud4RPi Applications

- Storage in Cloud4RPi
  - Store temperature data to Cloud4RPi (Send Data)
  - Control LED with Cloud4RPi (Retrieve Data)
- Automatically generate an alert in Cloud4RPi
  - Generate an alert to user by e-mail and web
- Demo Video

Store Temperature Data to Cloud4RPi



IoT Projects

Private to you

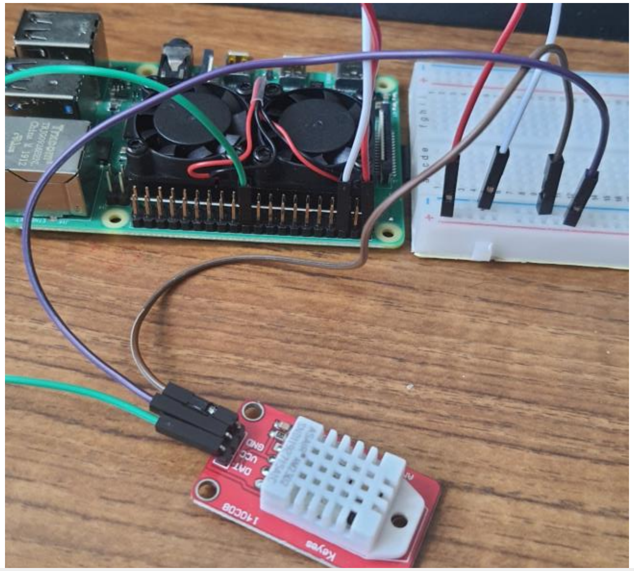
Widgets | Alerts

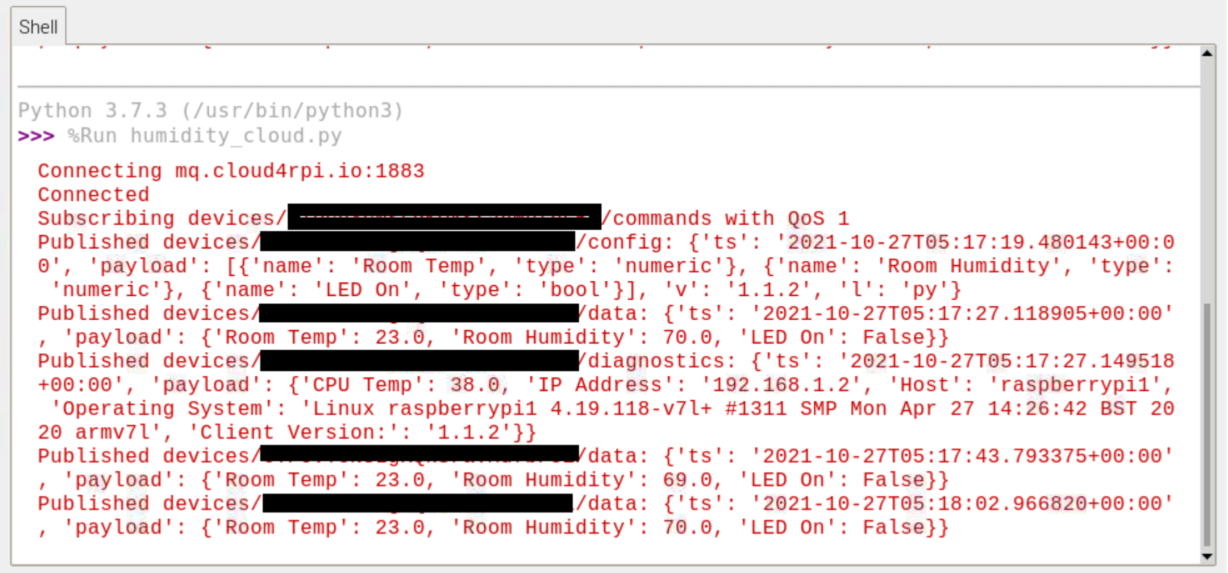
Room Temp 23.00

3 minutes ago

Room Humidity 70.00

3 minutes ago





```
Python 3.7.3 (/usr/bin/python3)
>>> %Run humidity_cloud.py

 Connecting mq.cloud4rpi.io:1883
 Connected
 Subscribing devices/[redacted]/commands with QoS 1
 Published devices/[redacted]/config: {'ts': '2021-10-27T05:17:19.480143+00:0
 0', 'payload': [{'name': 'Room Temp', 'type': 'numeric'}, {'name': 'Room Humidity', 'type':
  'numeric'}, {'name': 'LED On', 'type': 'bool'}], 'v': '1.1.2', 'l': 'py'}
 Published devices/[redacted]/data: {'ts': '2021-10-27T05:17:27.118905+00:00'
 , 'payload': {'Room Temp': 23.0, 'Room Humidity': 70.0, 'LED On': False}}
 Published devices/[redacted]/diagnostics: {'ts': '2021-10-27T05:17:27.149518
 +00:00', 'payload': {'CPU Temp': 38.0, 'IP Address': '192.168.1.2', 'Host': 'raspberrypi1',
  'Operating System': 'Linux raspberrypi1 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 20
 20 armv7l', 'Client Version:': '1.1.2'}}
 Published devices/[redacted]/data: {'ts': '2021-10-27T05:17:43.793375+00:00'
 , 'payload': {'Room Temp': 23.0, 'Room Humidity': 69.0, 'LED On': False}}
 Published devices/[redacted]/data: {'ts': '2021-10-27T05:18:02.966820+00:00'
 , 'payload': {'Room Temp': 23.0, 'Room Humidity': 70.0, 'LED On': False}}
```

Store Temperature Data to Cloud4RPi

(Source Code on Raspberry Pi)

```
import time
import Adafruit_DHT
import RPi.GPIO as GPIO

from time import sleep
import sys
import cloud4rpi
import rpi
import RPi.GPIO as GPIO

# Set Pin No, Device Key
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 21

DEVICE TOKEN = ‘YOUR_DEVICE_ID’

DATA_SENDING_INTERVAL = 10
POOL_INTERVAL = 0.5

# Get Data from Sensor
def update():
    global humidity, temperature
    humidity, temperature = Adafruit_DHT.ready_retry(DHT_SENSOR, DHT_PIN)

def humidity():
    update()
    return round(humidity, 2) if humidity is not None else None

def temp():
    update()
    return round(temperature, 2) if temperature is not None else None
```

Copy and paste your device token here

Interval time on sending data

Getting data from sensors

```
# Sending Data to Cloud
while True:
    variables = {
        ‘Room Temp’:{
                        ‘type’: ‘numeric’,
                        ‘bind’: temp
              },
        ‘Room Humidity’:{
                        ‘type’: ‘numeric’,
                        ‘bind’: humidity
              },
    }
    device = cloud4rpi.connect(DEVICE_TOKEN)

    try:
            device.declare(variables)
            device.publish_config()
            sleep(1)

      while True:
            if data_timer <= 0:
                  device.publish_data()
                  data_timer = DATA_SENDING_INTERVAL

                        sleep(POLL_INTERVAL)
                        data_timer -= POLL_INTERVAL
    except Exception as e:
            error = cloud4rpi.get_error_message(e)
            cloud4rpi.log.exception(“ERROR! %s %s, error,
sys.exc_info()[0])
    finally:
            sys.exit(0)
```

List of variables that will be used

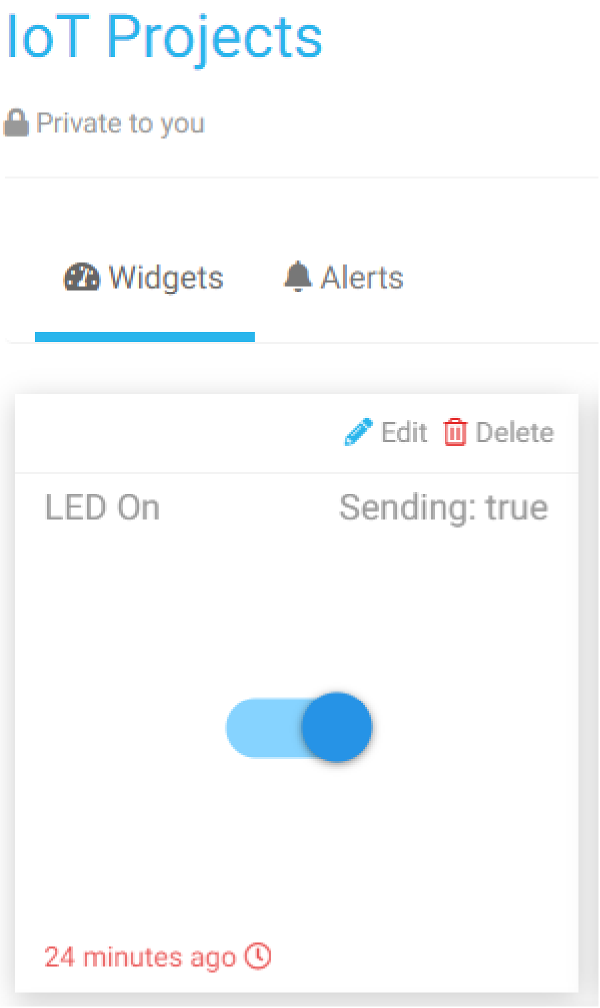
Connect RaspberryPi with cloud4rpi

Declaration of variables listed

Publish configuration to the cloud

Publish data to the cloud

Control LED with Cloud4RPi

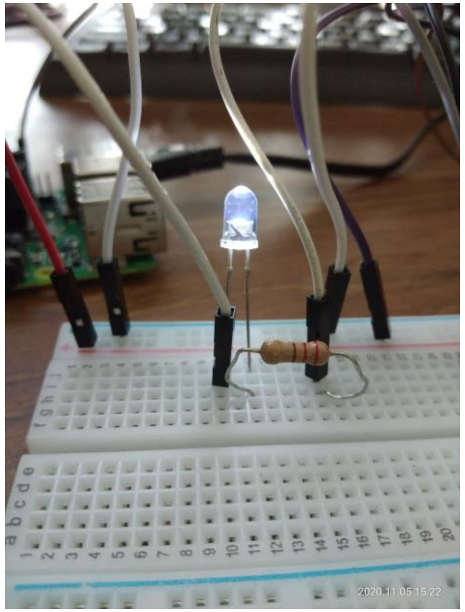


IoT Projects

Private to you

LED On Sending: true

24 minutes ago



```
Command received devices/[redacted]/commands: b'{"LED On":true}'
```

Control LED with Cloud4RPi (Source Code on Raspberry Pi)

```
import time
import Adafruit_DHT
import RPi.GPIO as GPIO

from time import sleep
import sys
import cloud4rpi
import rpi
import RPi.GPIO as GPIO

# Set Pin No, Device Key
LIGHT_PIN = 20

DEVICE TOKEN = ‘YOUR_DEVICE_ID’

DATA_SENDING_INTERVAL = 10
POOL_INTERVAL = 0.5

# Get Data from Sensor
def led_control(value=None):
            GPIO.output(LIGHT_PIN,
value)
            return
GPIO.input(LIGHT_PIN)
```

Copy and paste your device token here

Interval time on sending data

Getting data from sensors

```
# Sending Data to Cloud
while True:
    variables = {
        ‘LED On’:{
                        ‘type’: ‘bool’,
                        ‘value’: False
                        ‘bind’: led_control
              },
    }
    device = cloud4rpi.connect(DEVICE_TOKEN)

    try:
            device.declare(variables)
            device.publish_config()
            sleep(1)

      while True:
            if data_timer <= 0:
                  device.publish_data()
                  data_timer = DATA_SENDING_INTERVAL

                        sleep(POLL_INTERVAL)
                        data_timer -= POLL_INTERVAL
    except Exception as e:
            error = cloud4rpi.get_error_message(e)
            cloud4rpi.log.exception(“ERROR! %s %s, error,
sys.exc_info()[0])
    finally:
            sys.exit(0)
```

List of variables that will be used

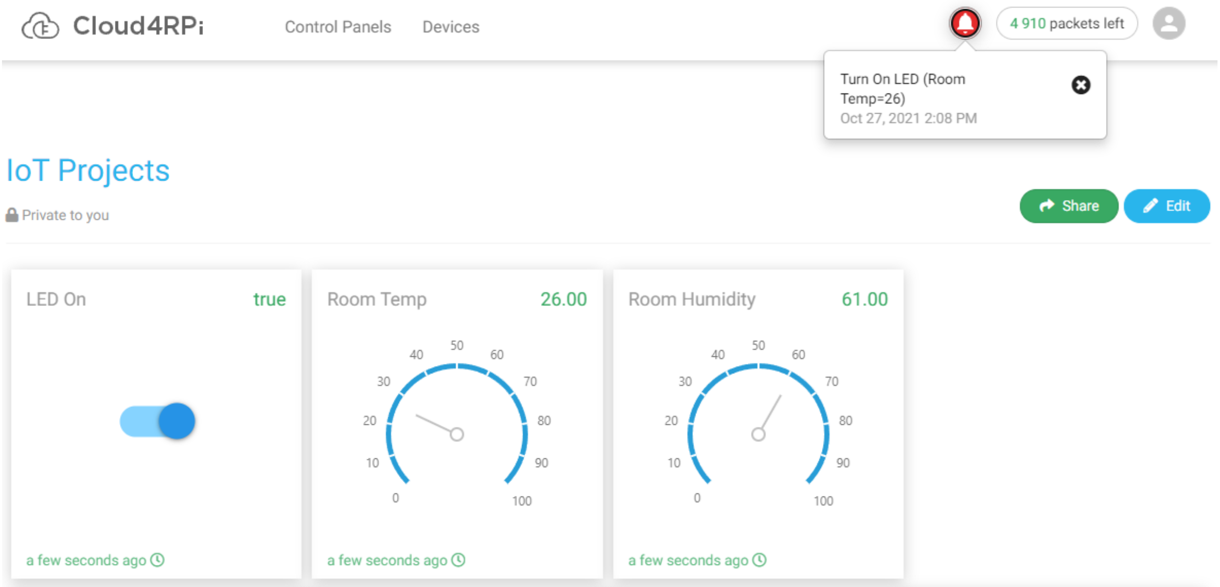
Connect RaspberryPi with cloud4rpi

Declaration of variables listed

Publish configuration to the cloud

Publish data to the cloud

Automatically Generate an Alert



Turn On LED (Room Temp=26)
Oct 27, 2021 2:08 PM

IoT Projects

Private to you

LED On true

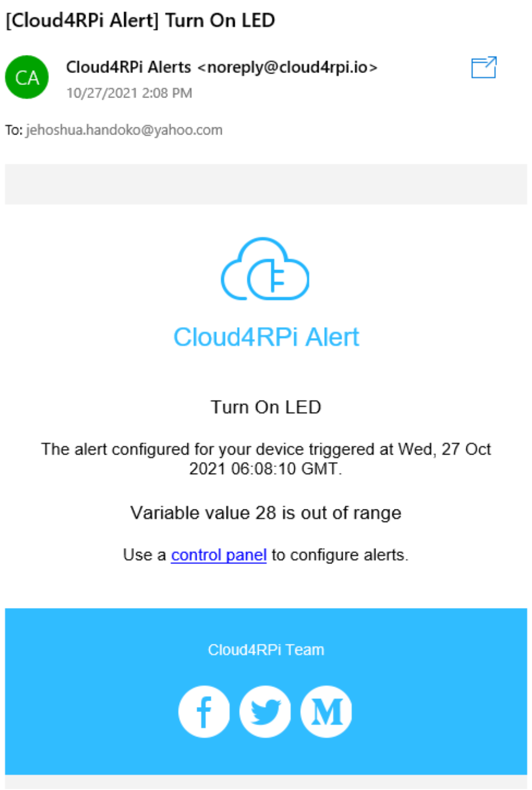
Room Temp 26.00

Room Humidity 61.00

a few seconds ago

a few seconds ago

a few seconds ago



[Cloud4RPi Alert] Turn On LED

Cloud4RPi Alerts <noreply@cloud4rpi.io>
10/27/2021 2:08 PM

To: jehoshua.handoko@yahoo.com

Cloud4RPi Alert

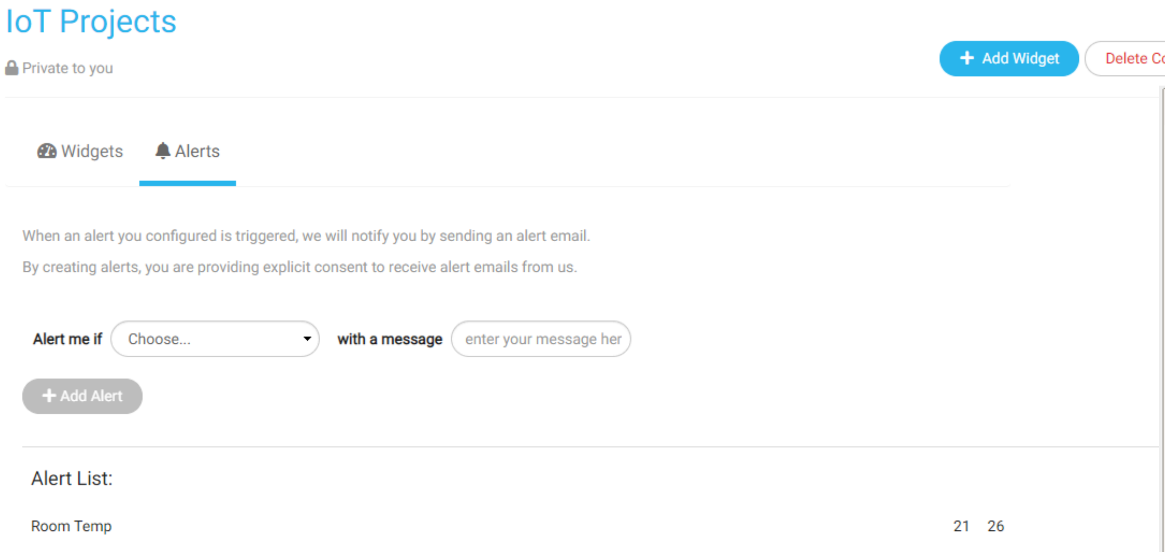
Turn On LED

The alert configured for your device triggered at Wed, 27 Oct 2021 06:08:10 GMT.

Variable value 28 is out of range

Use a control panel to configure alerts.

Cloud4RPi Team



IoT Projects

Private to you

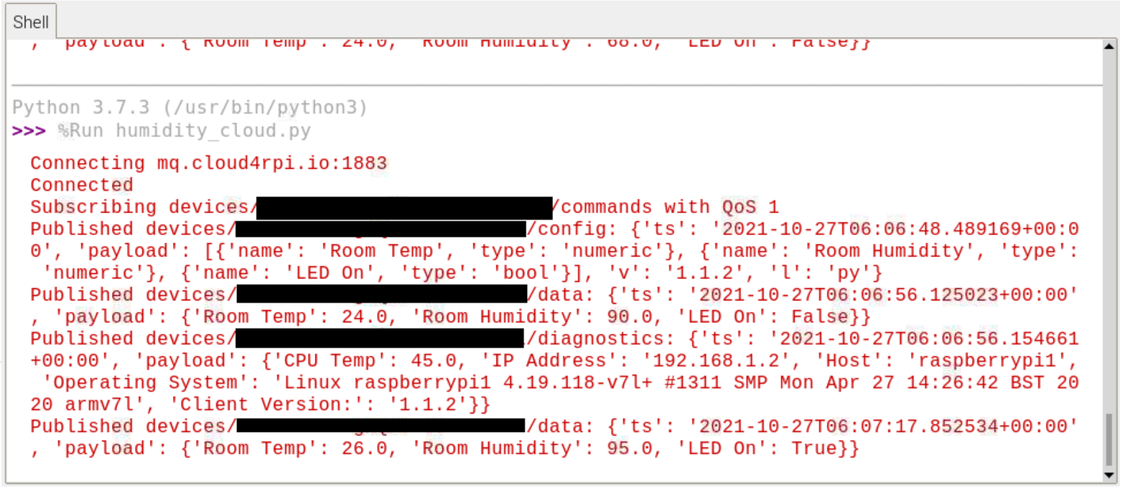
When an alert you configured is triggered, we will notify you by sending an alert email.

By creating alerts, you are providing explicit consent to receive alert emails from us.

Alert me if Choose... with a message enter your message her

Alert List:

Room Temp	21	26



```
, 'payload': {'Room Temp': 24.0, 'Room Humidity': 68.0, 'LED On': False}}

Python 3.7.3 (/usr/bin/python3)
>>> %Run humidity_cloud.py

  Connecting mq.cloud4rpi.io:1883
  Connected
  Subscribing devices/[redacted]/commands with QoS 1
  Published devices/[redacted]/config: {'ts': '2021-10-27T06:06:48.489169+00:0
  0', 'payload': [{'name': 'Room Temp', 'type': 'numeric'}, {'name': 'Room Humidity', 'type':
   'numeric'}, {'name': 'LED On', 'type': 'bool'}], 'v': '1.1.2', 'l': 'py'}
  Published devices/[redacted]/data: {'ts': '2021-10-27T06:06:56.125023+00:00'
  , 'payload': {'Room Temp': 24.0, 'Room Humidity': 90.0, 'LED On': False}}
  Published devices/[redacted]/diagnostics: {'ts': '2021-10-27T06:06:56.154661
  +00:00', 'payload': {'CPU Temp': 45.0, 'IP Address': '192.168.1.2', 'Host': 'raspberrypi1',
   'Operating System': 'Linux raspberrypi1 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 20
  20 armv7l', 'Client Version:': '1.1.2'}}
  Published devices/[redacted]/data: {'ts': '2021-10-27T06:07:17.852534+00:00'
  , 'payload': {'Room Temp': 26.0, 'Room Humidity': 95.0, 'LED On': True}}
```

Automatically Generate an Alert (Source Code on Raspberry Pi)

```
import time
import Adafruit_DHT
import RPi.GPIO as GPIO

from time import sleep
import sys
import cloud4rpi
import rpi
import RPi.GPIO as GPIO

# Set Pin No, Device Key
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 21
LIGHT_PIN = 20
threshold = 25
GPIO.setup(LIGHT_PIN, GPIO.OUT)

DEVICE TOKEN = ‘YOUR_DEVICE_ID’

DATA_SENDING_INTERVAL = 10
POOL_INTERVAL = 0.5

# Get Data from Sensor
def led_control(value=None):
    GPIO.output(LIGHT_PIN, value)
    return GPIO.input(LIGHT_PIN)

def update():
    global humidity, temperature
    humidity, temperature = Adafruit_DHT.ready_retry(DHT_SENSOR, DHT_PIN)

def humidity():
    update()
    return round(humidity, 2) if humidity is not None else None

def temp():
    update()
    return round(temperature, 2) if temperature is not None else None

def led_control_main(value=None):
    temperature = temp()
    if temperature < threshold:
                             return led_control(False)
    else:
                             return led_control(True)
```

Copy and paste your device token here

Interval time on sending data

Getting data from sensors

Automatically turn on led when above threshold

```
# Sending Data to Cloud
while True:
  variables = {
    ‘Room Temp’:{
            ‘type’: ‘numeric’,
            ‘bind’: temp
       },
    ‘Room Humidity’:{
            ‘type’: ‘numeric’,
            ‘bind’: humidity
       },
       ‘LED On’:{
            ‘type’: ‘bool’,
            ‘value’: False
            ‘bind’: led_control_main
       },
  }
  device = cloud4rpi.connect(DEVICE_TOKEN)

  try:
      device.declare(variables)
      device.publish_config()
      sleep(1)

   while True:
      if data_timer <= 0:
         device.publish_data()
         data_timer = DATA_SENDING_INTERVAL

            sleep(POLL_INTERVAL)
            data_timer -= POLL_INTERVAL
  except Exception as e:
      error = cloud4rpi.get_error_message(e)
      cloud4rpi.log.exception(“ERROR! %s %s, error,
sys.exc_info()[0])
  finally:
      sys.exit(0)
```

List of variables that will be used

Connect RaspberryPi with cloud4rpi

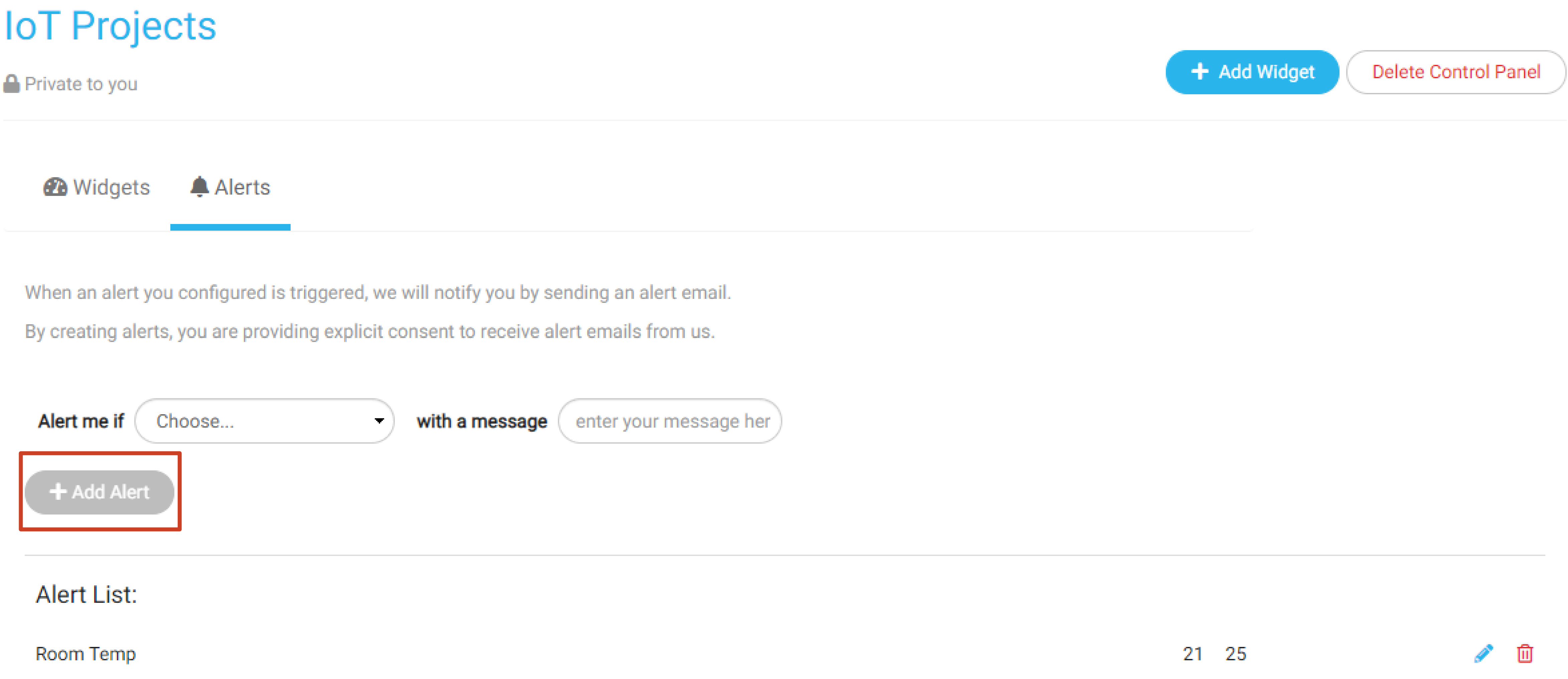
Declaration of variables listed

Publish configuration to the cloud

Publish data to the cloud

Automatically Generate an Alert (Program runs on Cloud4RPi)

Demo video:
https://youtu.be/XwUcUblRegc



IoT Projects

Private to you

When an alert you configured is triggered, we will notify you by sending an alert email.

By creating alerts, you are providing explicit consent to receive alert emails from us.

Alert me if Choose... with a message enter your message her

Alert List:

Room Temp	21	25	

Assignment 2 - Specification

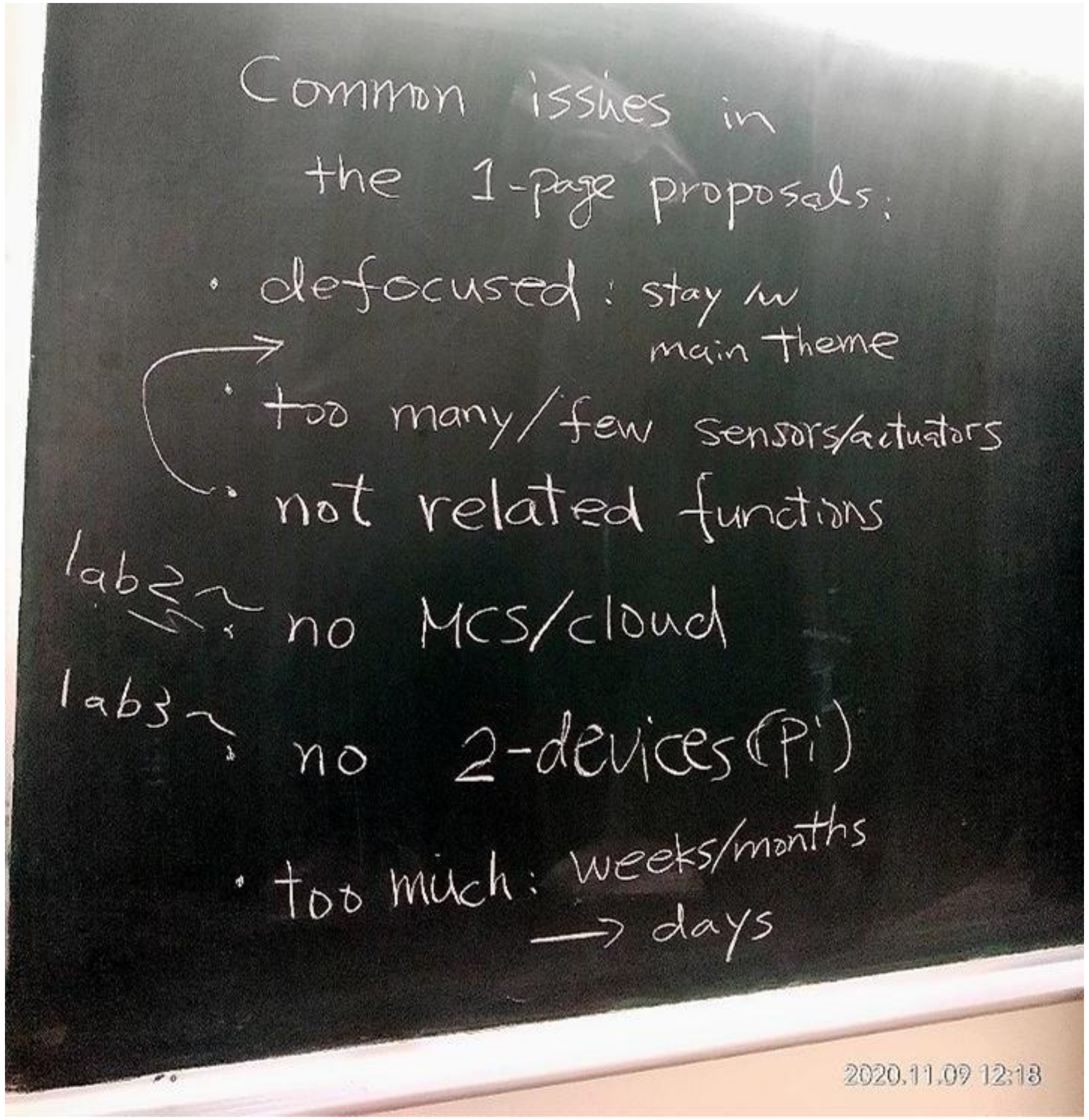
- Objectives:
  - IoT with “Your” Choices of Sensors and Actuators
  - Integration with Cloud4RPi

- Upload to E3 before 12/1 at 23:59PM
  - Assignment 2 – deliverables
    - Report (2-4 pages) using our template
      - Explain the objectives
      - Explain the specification of sensors and actuators used
      - Explain the system design
      - Explain your source code and the detail of how your script can post and get the data from Cloud4RPi to your sensors and actuators, respectively
      - Link to a 3-minute demo video on YouTube
    - Source Codes
    - Zip the above files into one compressed file and upload
  - Q&A? Post on E3 discussion board

Assignment 2 - Specification

- Note for Assignment 2:
  - You must use different combinations of sensors and actuators than the ones we present
    - If using the same combinations, your maximum score is only 65.
  - The report can be written in Chinese or English, but the video must be delivered in English.
  - In video, explain how you assemble your additional sensors and actuators, how to store and retrieve data from cloud, and show the results.
  - Upload your video to YouTube and put the link into your report. Don’t upload your video to E3.
  - In your report, make sure you have a diagram of connected sensors, actuators, and cloud.

Frequent Issues in the 1-Page Proposals



Common issues in
the 1-page proposals:

- defocused: stay w/ main theme
- too many/few sensors/actuators
- not related functions

lab2 ~ · no MCS/cloud

lab3 ~ · no 2-devices (Pi)

- too much: weeks/months → days

2020.11.09 12:18

Thank You