Remote Data Acquisition and Internet of Things

Introduction to Computer Engineering

Assoc. Prof. Anan Phonphoem
Asst. Prof. Chaiporn Jaikaeo
Asst. Prof. Aphirak Jansang

Department of Computer Engineering
Kasetsart University

IWING Team Leaders

Associate Professor

Anan Phonphoem

รศ.ดร.อนันต์ ผลเพิ่ม

Assistant Professor

Chaiporn Jaikeo

ผศ.ดร.ชัยพร ใจแก้ว

Assistant Professor

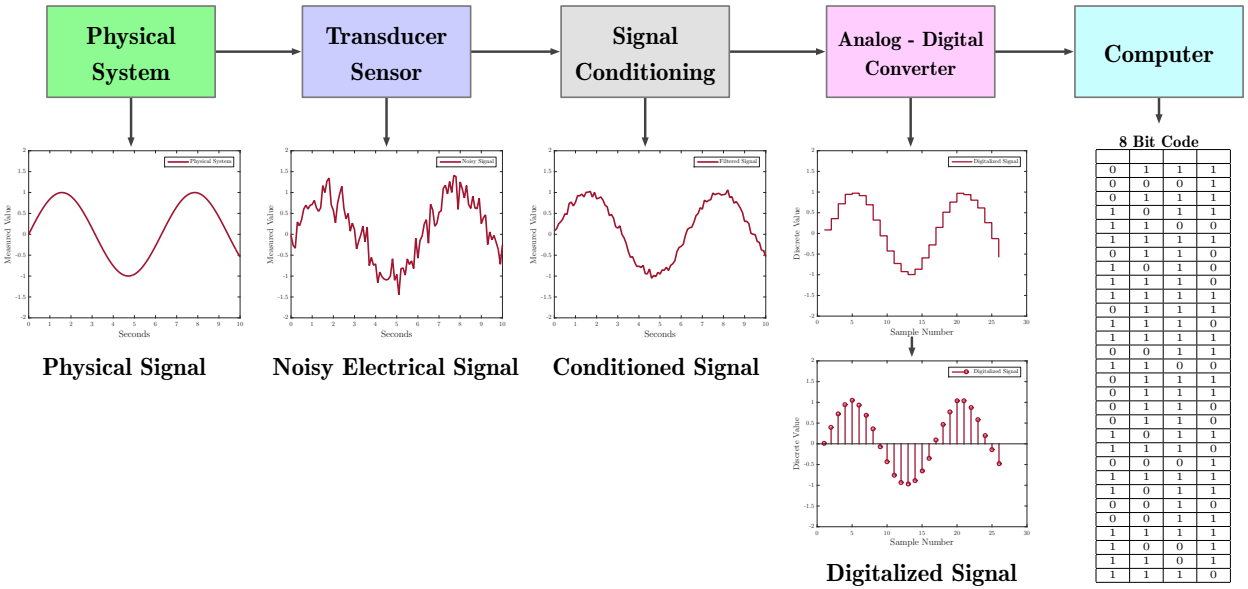
Aphirak Jansang

ผศ.ดร.อภิรักษ์ จันทร์สร้าง

Data Acquisition

The process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.

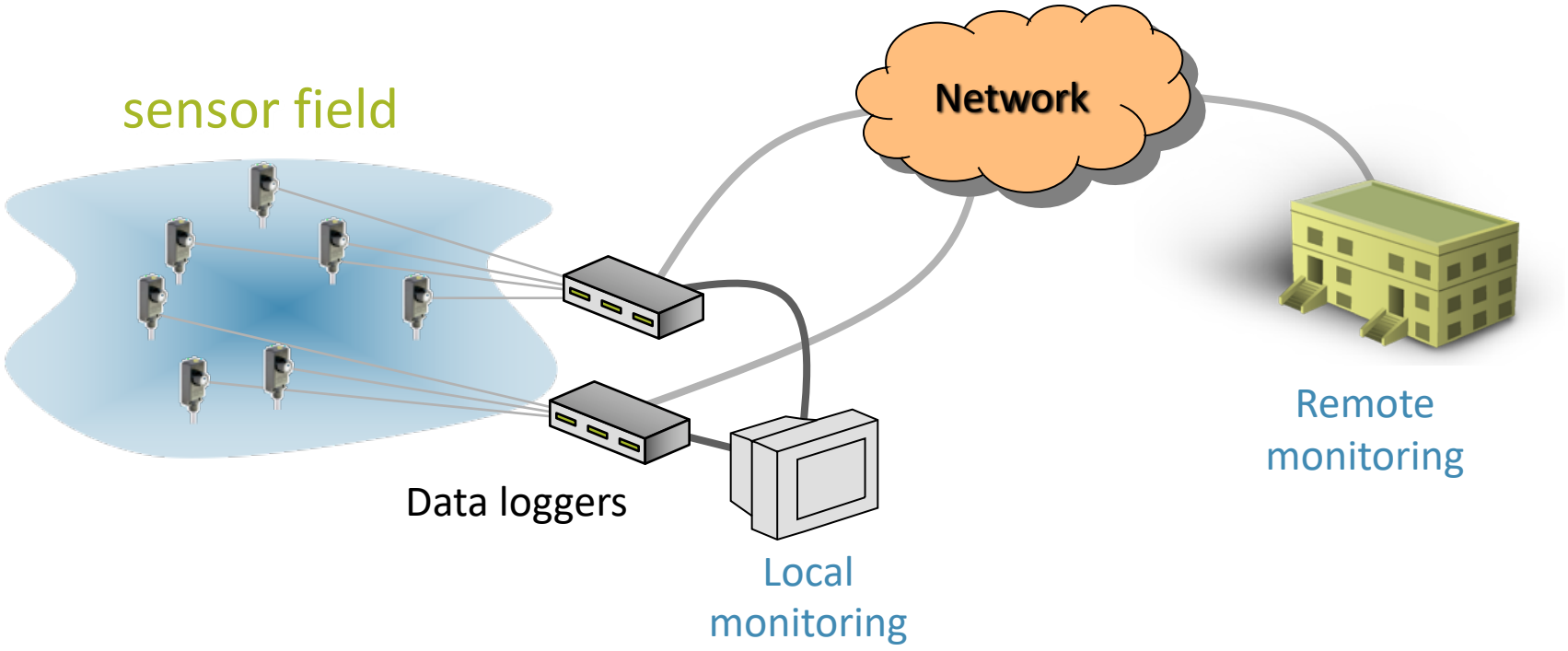
- **Data acquisition system**, also known as **DAS** or **DAQ**, is a system that performs the above process



https://en.wikipedia.org/wiki/Data_acquisition

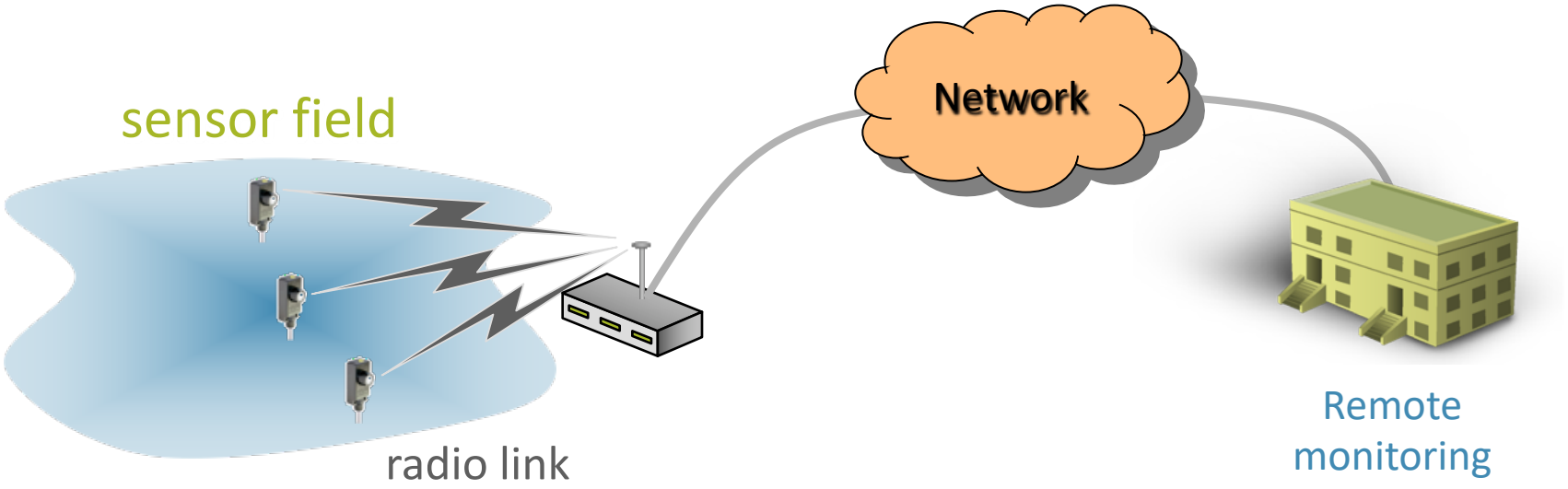
Remote Data Acquisition

- Traditional, wired sensors



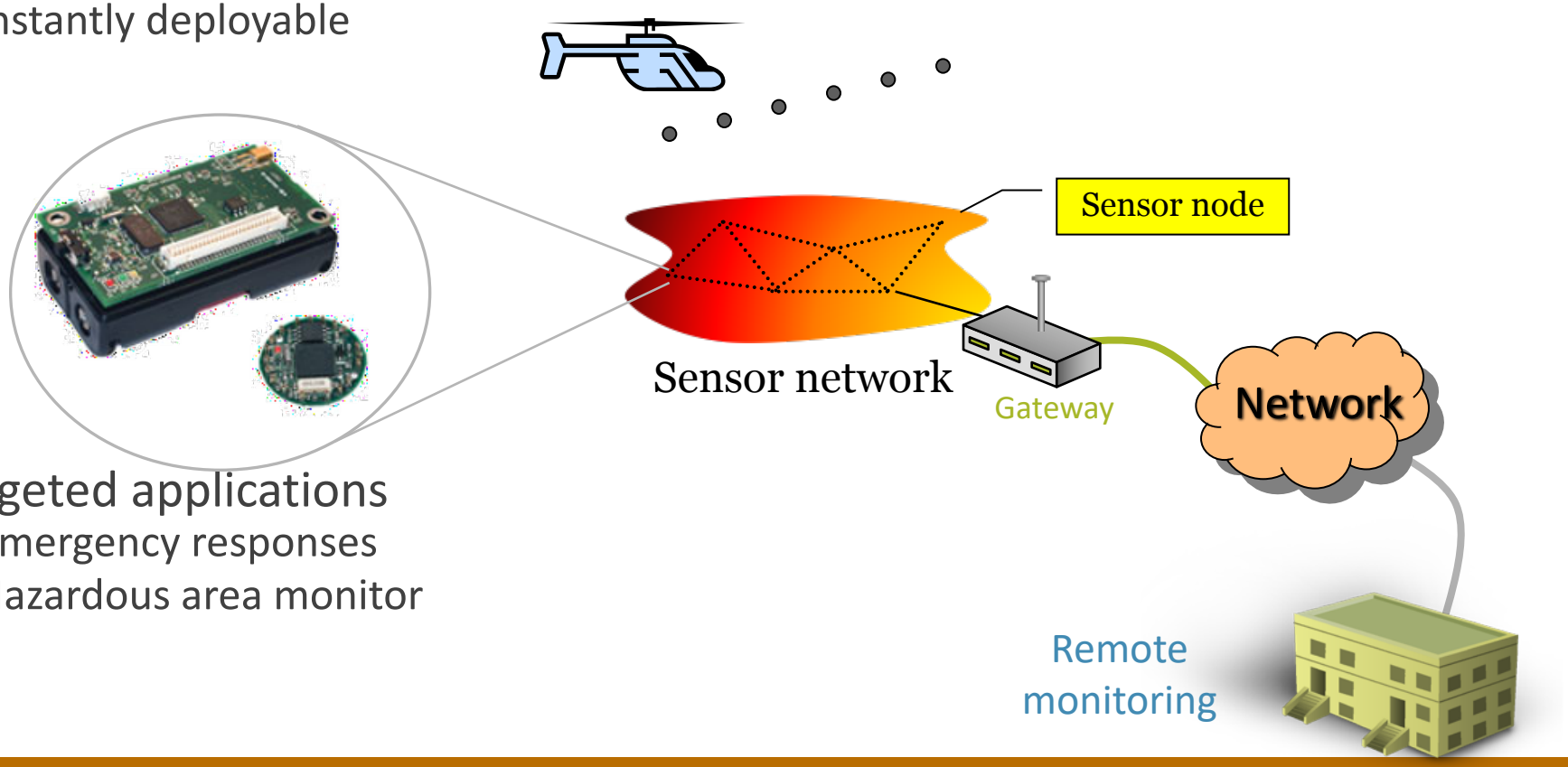
Remote Data Acquisition

- Wireless sensors
- Sensors communicate with data logger via radio links



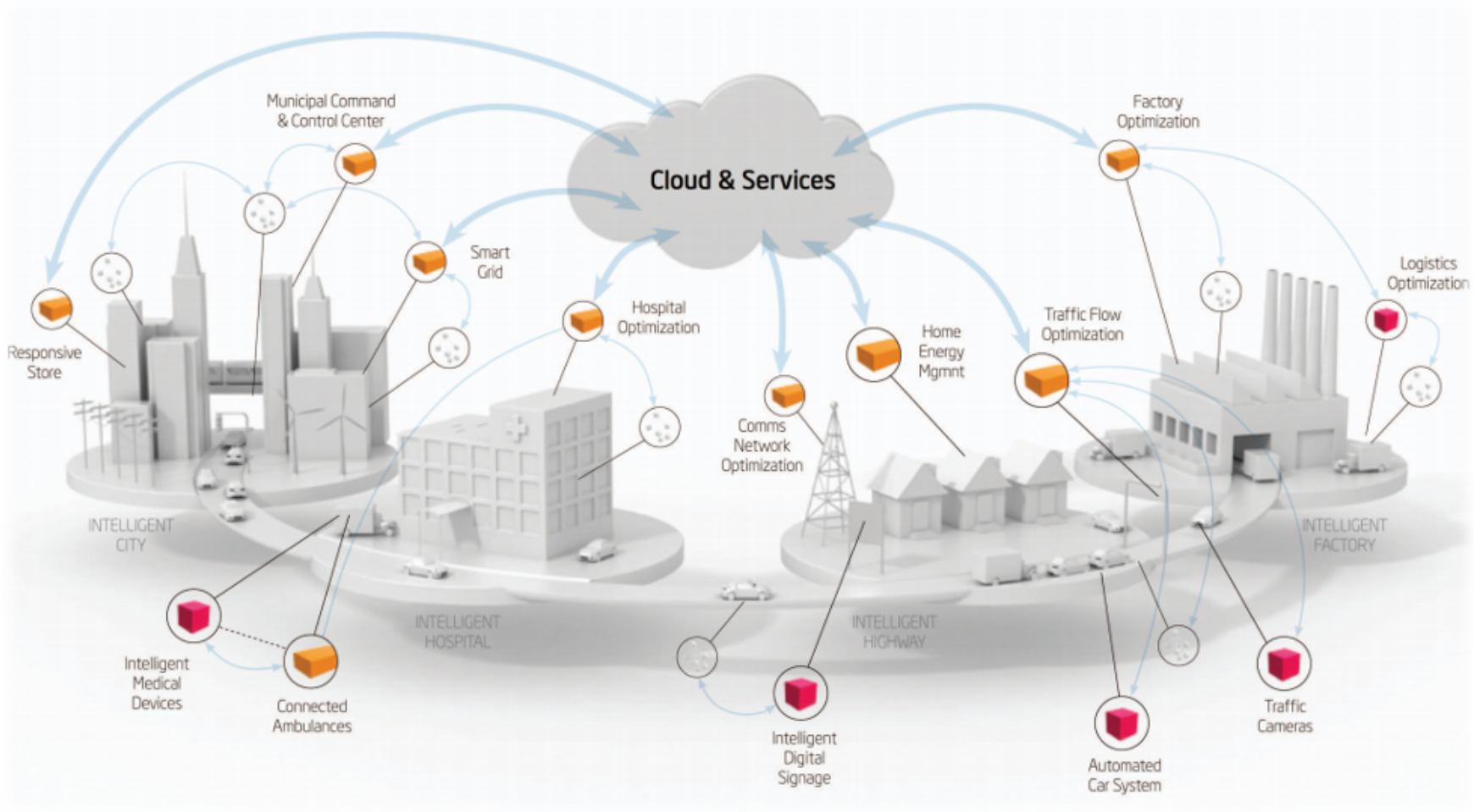
Remote Data Acquisition

- **W**ireless **S**ensor **N**etworks – WSN
- Sensor nodes deployed and forming an ad hoc network
  - Requires no hubs, access points
  - Instantly deployable



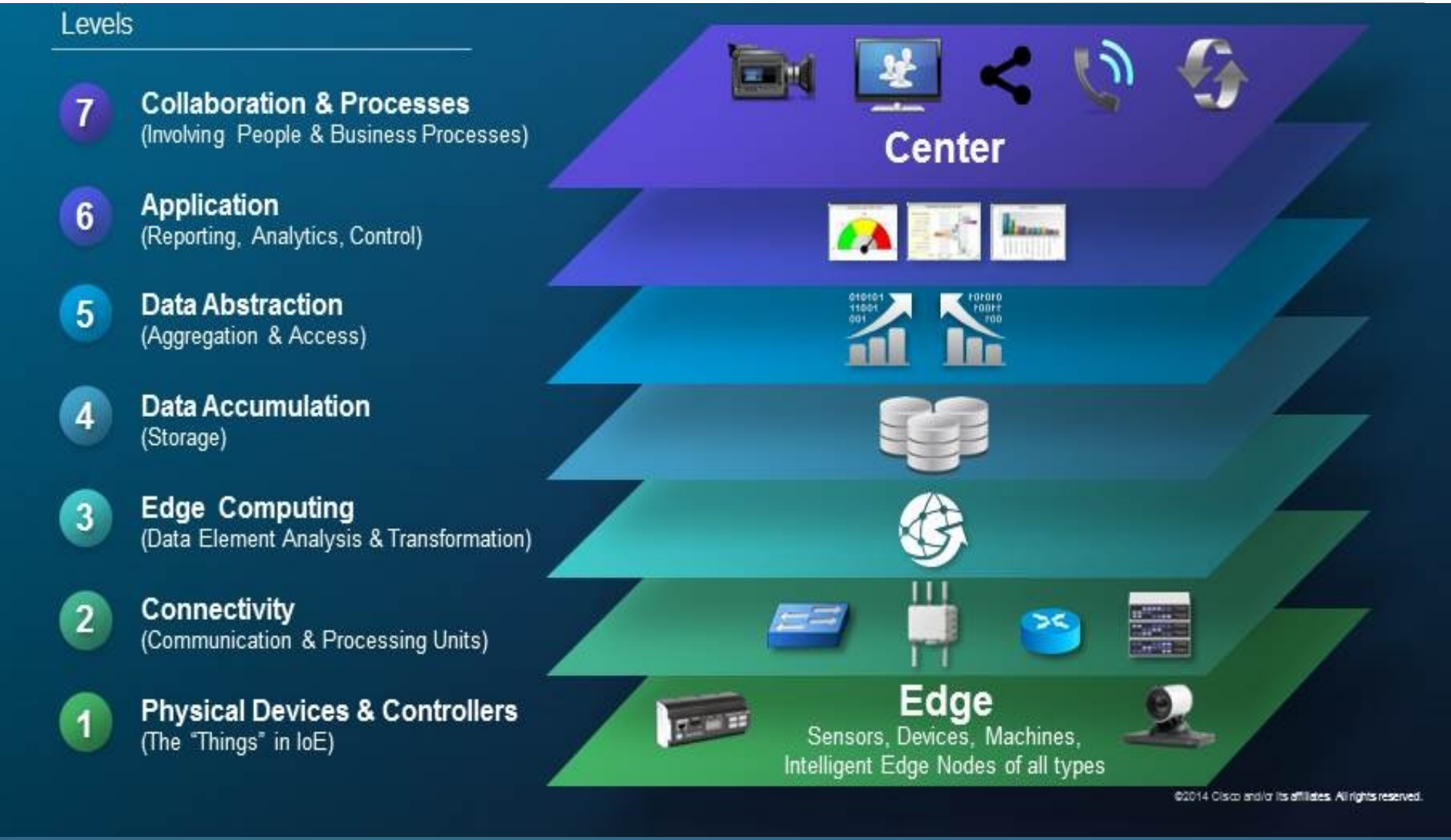
- Targeted applications
  - Emergency responses
  - Hazardous area monitor

Internet of Things (IoT)



http://www.opinno.com/en/content/internet-things-0

IoT Reference Model



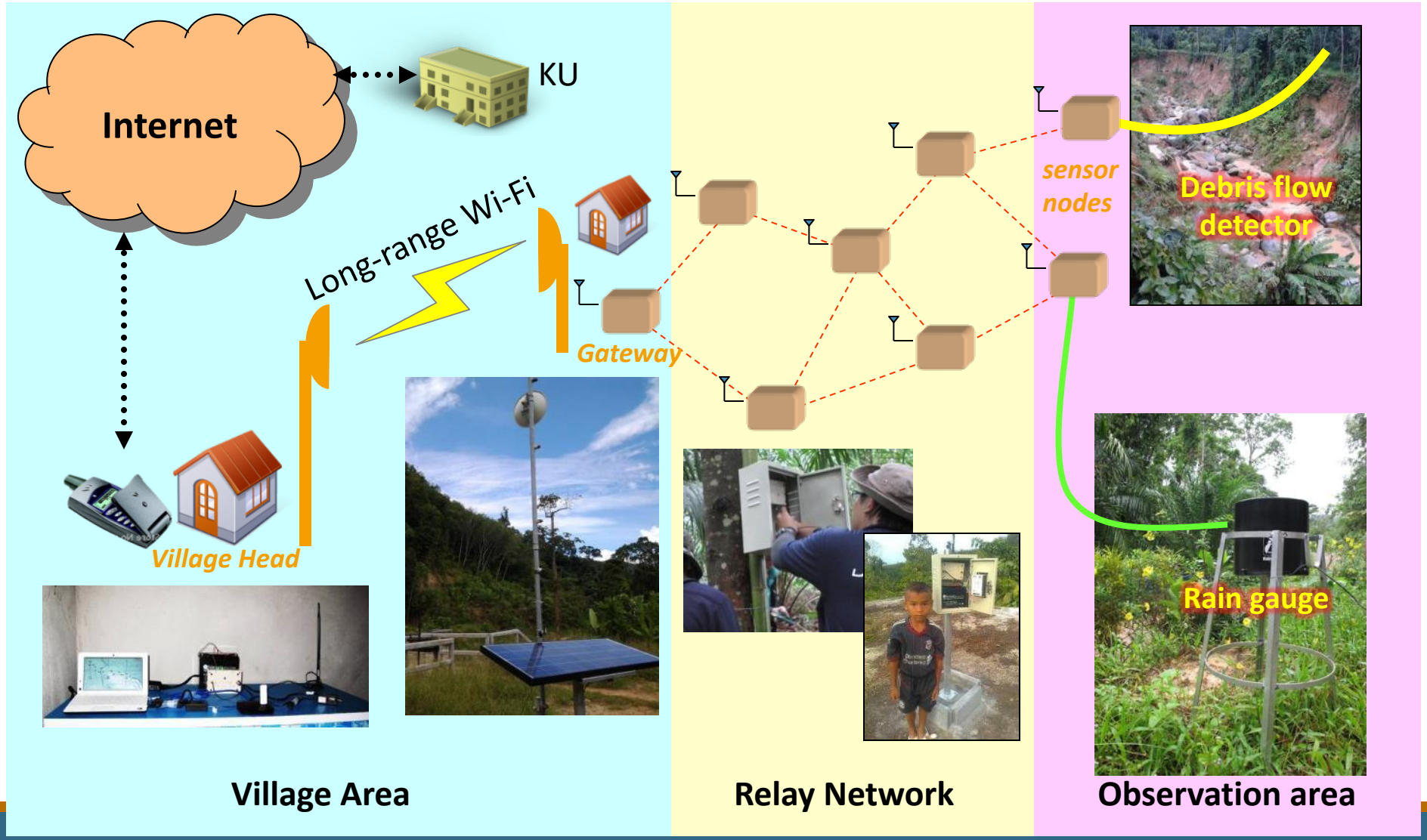
https://insightaas.com/new-iot-reference-model-creates-common-tongue/

Project: Landslide Monitor

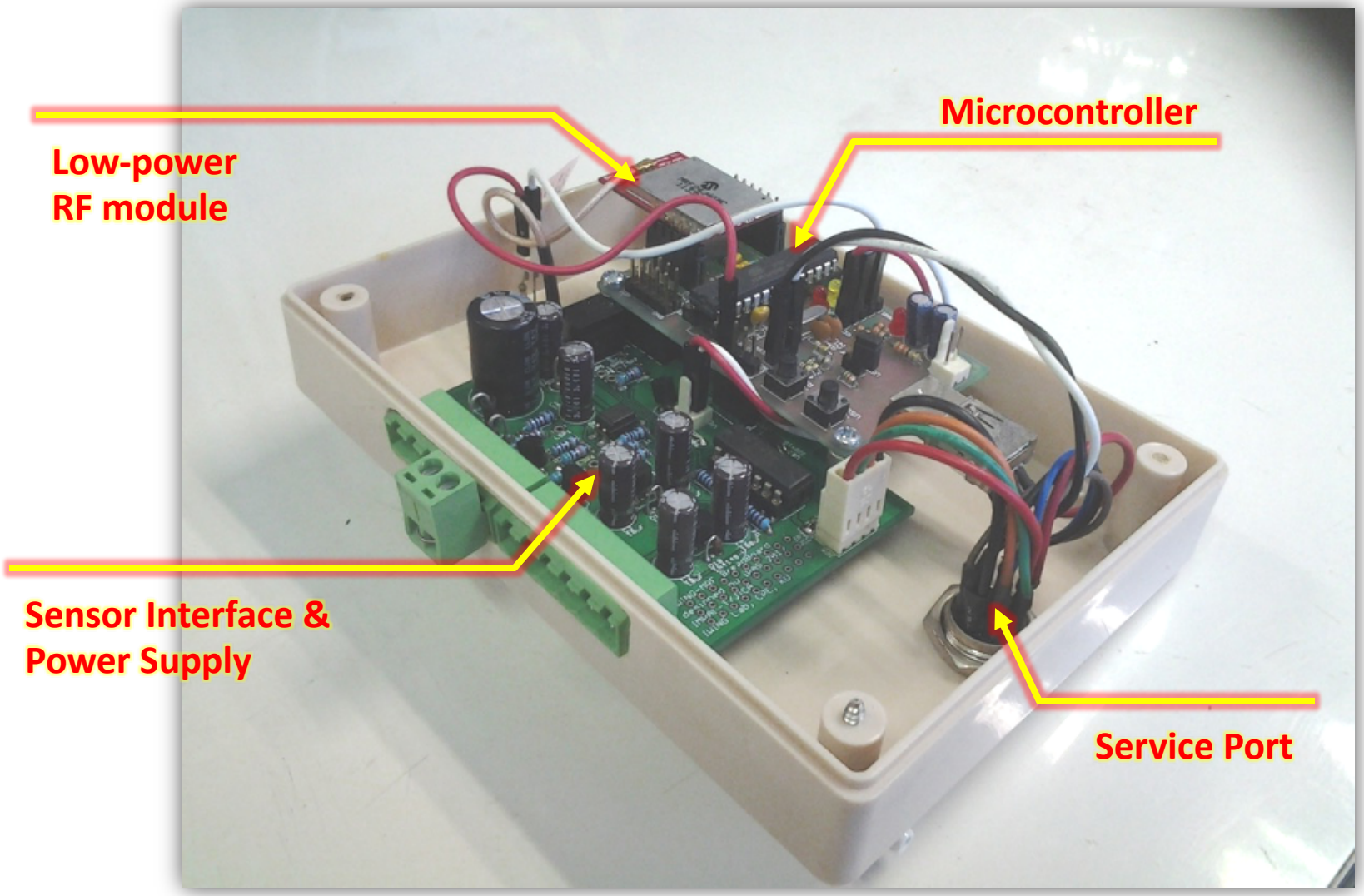
- Location: Khao Panom, Krabi
- Period: 2011 – 2013
- Communication technologies
  - IEEE 802.15.4
  - Ubiquiti's airMAX® TDMA
  - 2G/3G
- Purpose
  - Measure rainfalls and detect landslide on the mountain

Khao Phanom, Krabi Province, Southern Thailand

Landslide Monitor: System Overview



Sensor Nodes: *Internals*



Landslide Monitor: Observation Areas

- Rain gauge installation



- Debris flow detector installation


Landslide Monitor: Relay Nodes

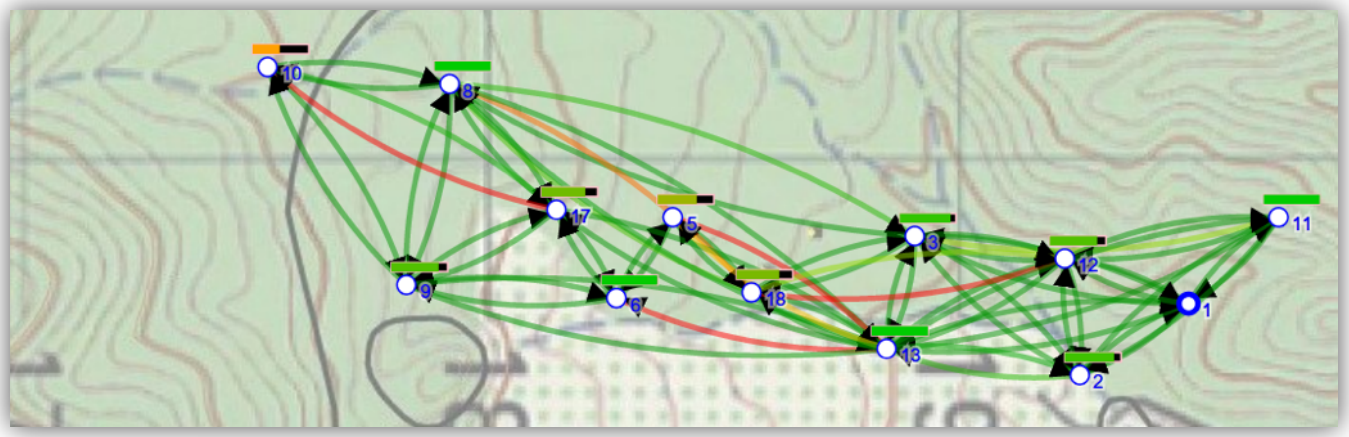
Landslide Monitor: Deployment






Network Status Monitor

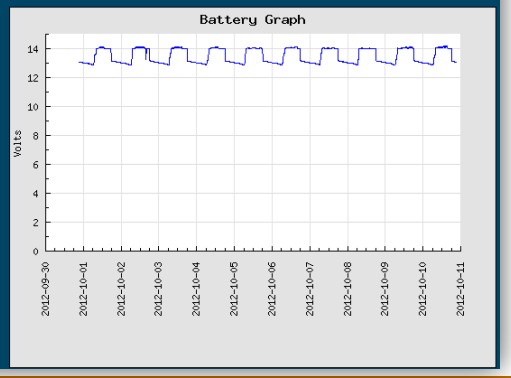
- User-friendly interface for the villagers



PAN: Huay Nam Kaew, Krabi (1282)
Node ID: 8

Battery: 13.06 V
Last updated: 2012-10-10 21:00:32

hear_from	RSSI	LR	datetime
18	2.75	30.2	2012-10-10 21:00:32
9	47.84	97.65	2012-10-10 21:00:31
3	0.78	97.65	2012-10-10 21:00:30
10	47.45	97.65	2012-10-10 21:00:30
13	10.98	97.65	2012-10-10 21:00:30
17	21.18	97.65	2012-10-10 21:00:30
6	0	81.57	2012-10-10 20:53:29



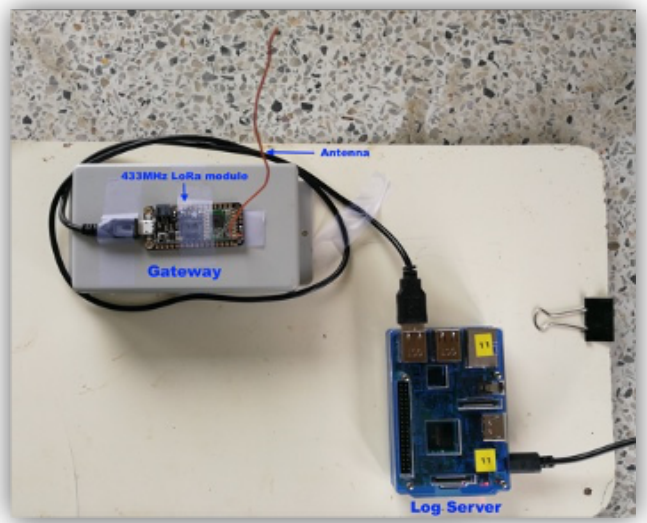
Project: Cattle/Dog Tracking

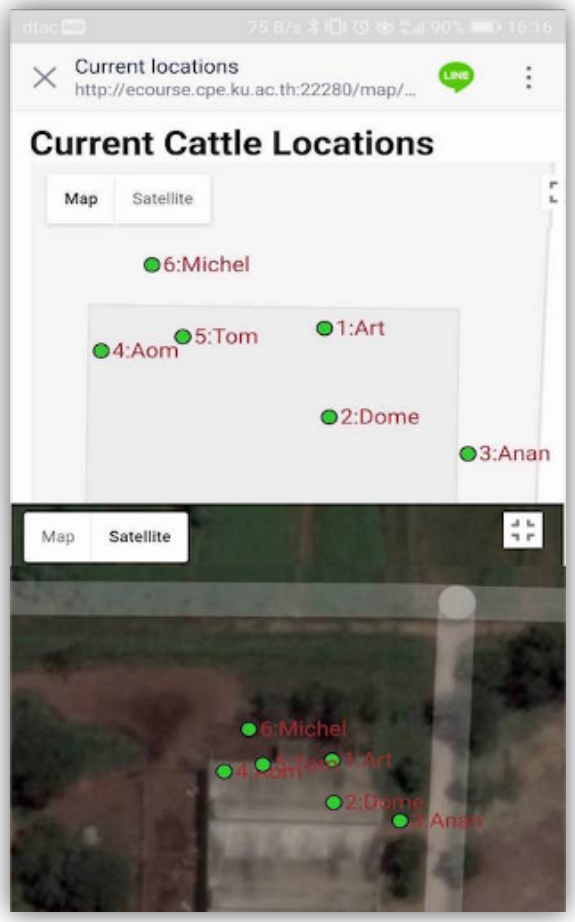
- Locations: Kanchanaburi, Nan, and Vietnam
- Period: 2018 – 2019
- Communication technologies
  - LoRa
  - 3G/4G
- Purpose
  - Track animals and their interaction in the field for up to 1-3 months

cirad

Cattle/Dog Tracking

- Tracks and records cattle movements with GPS receiver





18

Power Management

https://www.cpe.ku.ac.th/~cpj/cattrack/kps.2018-06-11.html

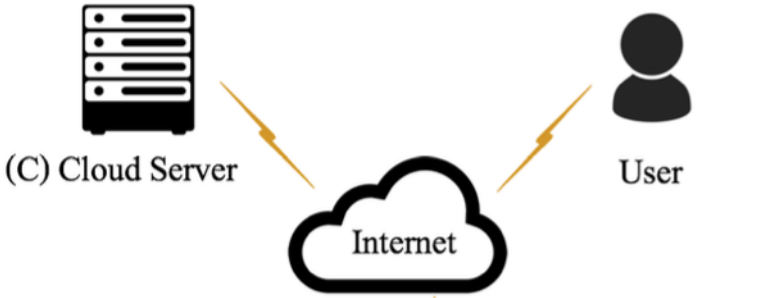
19

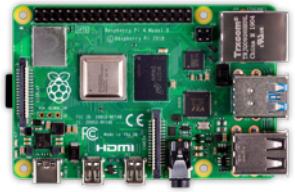
Growth Chamber Control

- Locations: Department of Biochemistry, Faculty of Science, KU
- Period: 2018 – 2020
- Communication technologies
  - Wi-Fi
- Purpose
  - To monitor and control environment inside a growth chamber

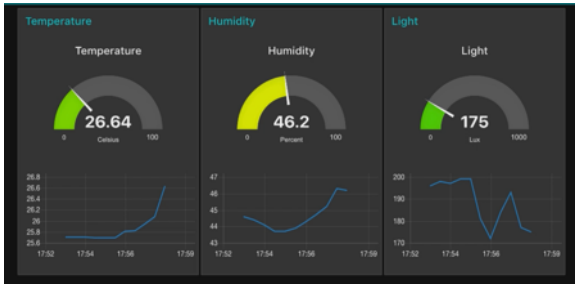
Growth Chamber Monitoring

- Monitors and controls light, temperature and humidity in a chamber for plant growth and tissue culture





Growth Chamber Monitoring



A

Humidity & Temperature Sensor

Light Sensor

GPIO

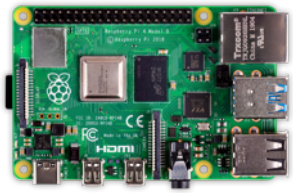
GPIO

Microcontroller

GPIO

MQTT Client

Wi-Fi



B

Local Server

Database

MQTT Broker

Wi-Fi

Ethernet

Internet

C

Cloud Server

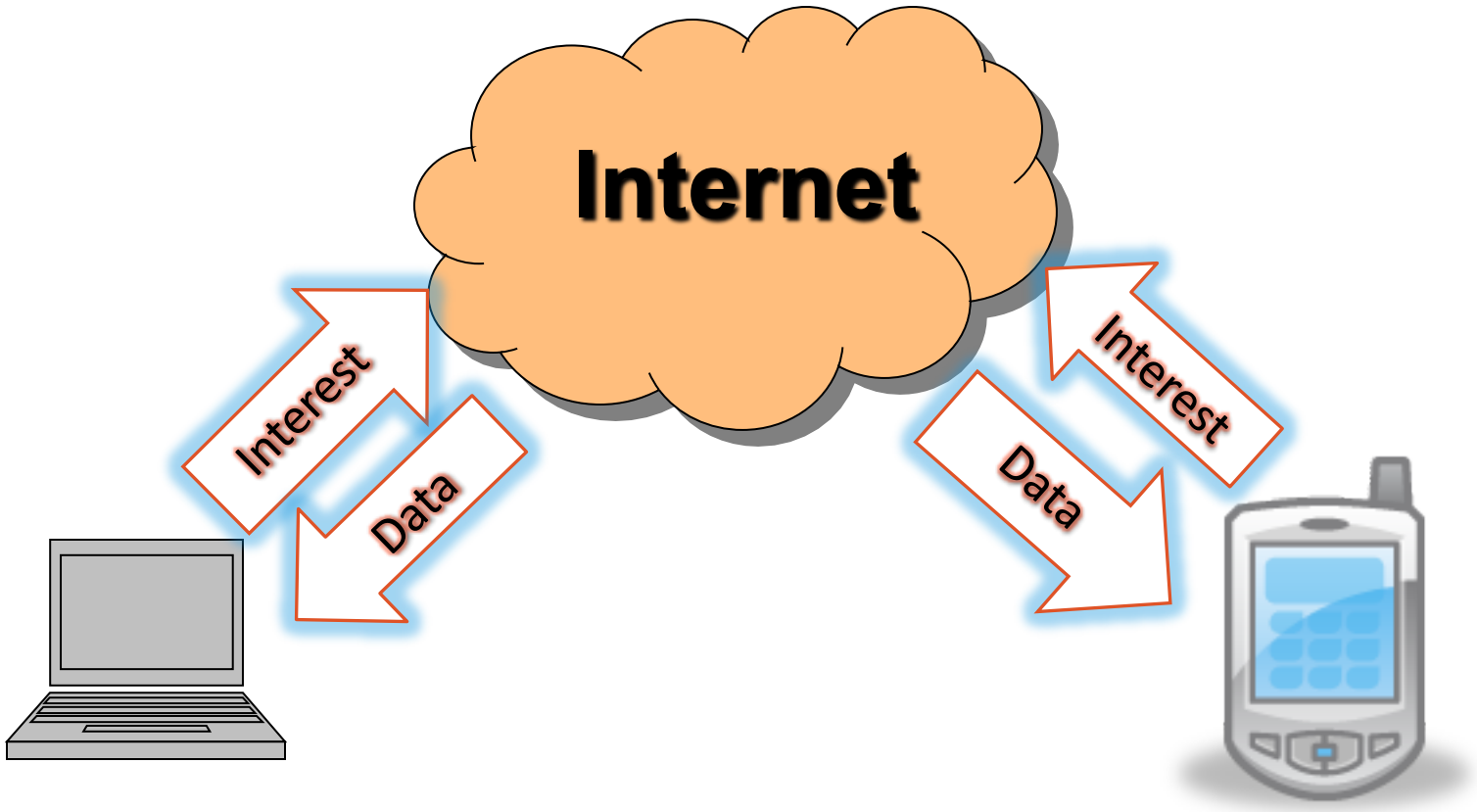
Database

Server

MQTT Client

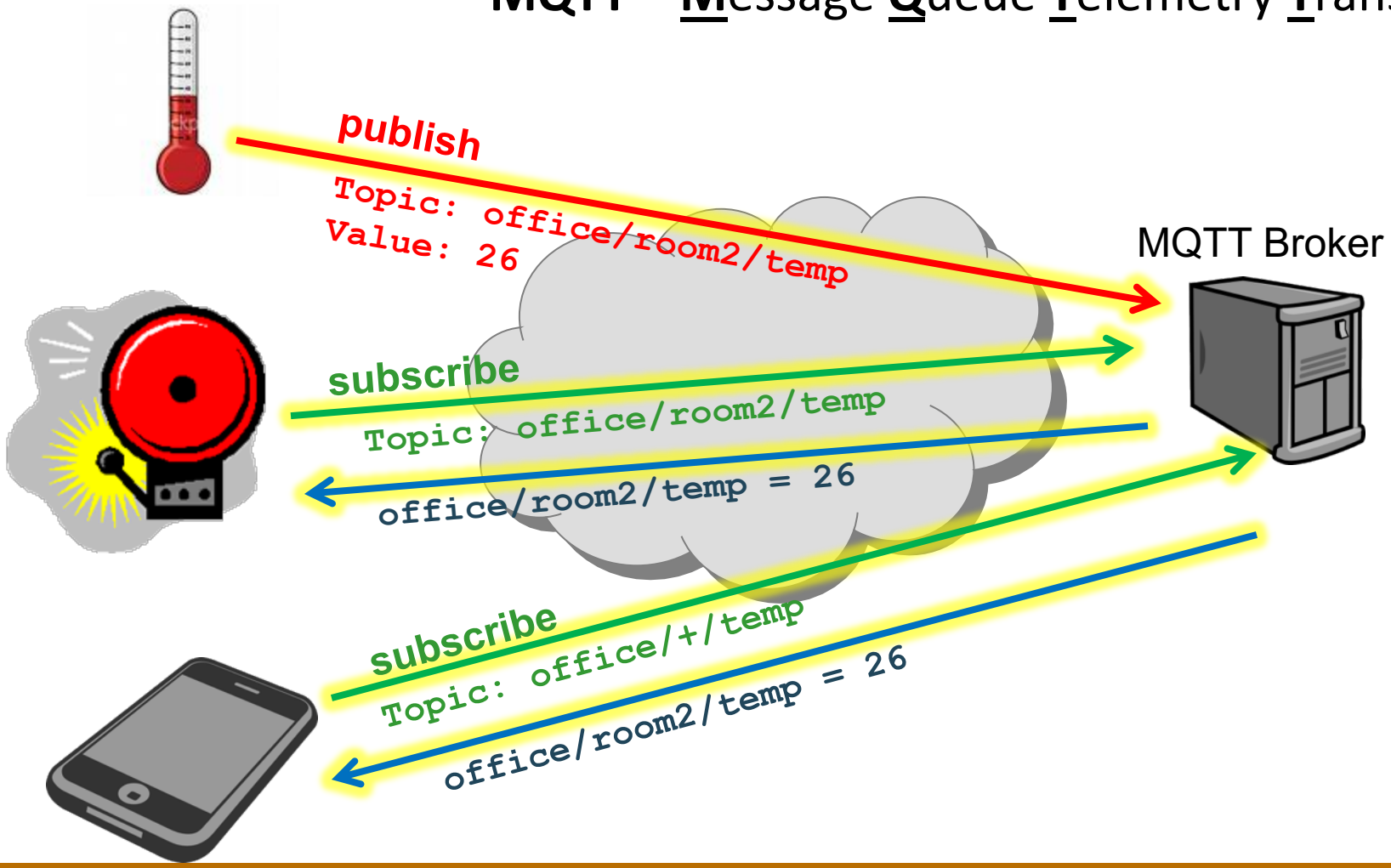
Ethernet

Content-Centric Networking Model



MQTT Protocol

MQTT – **M**essage **Q**ueue **T**elemetry **T**ransport

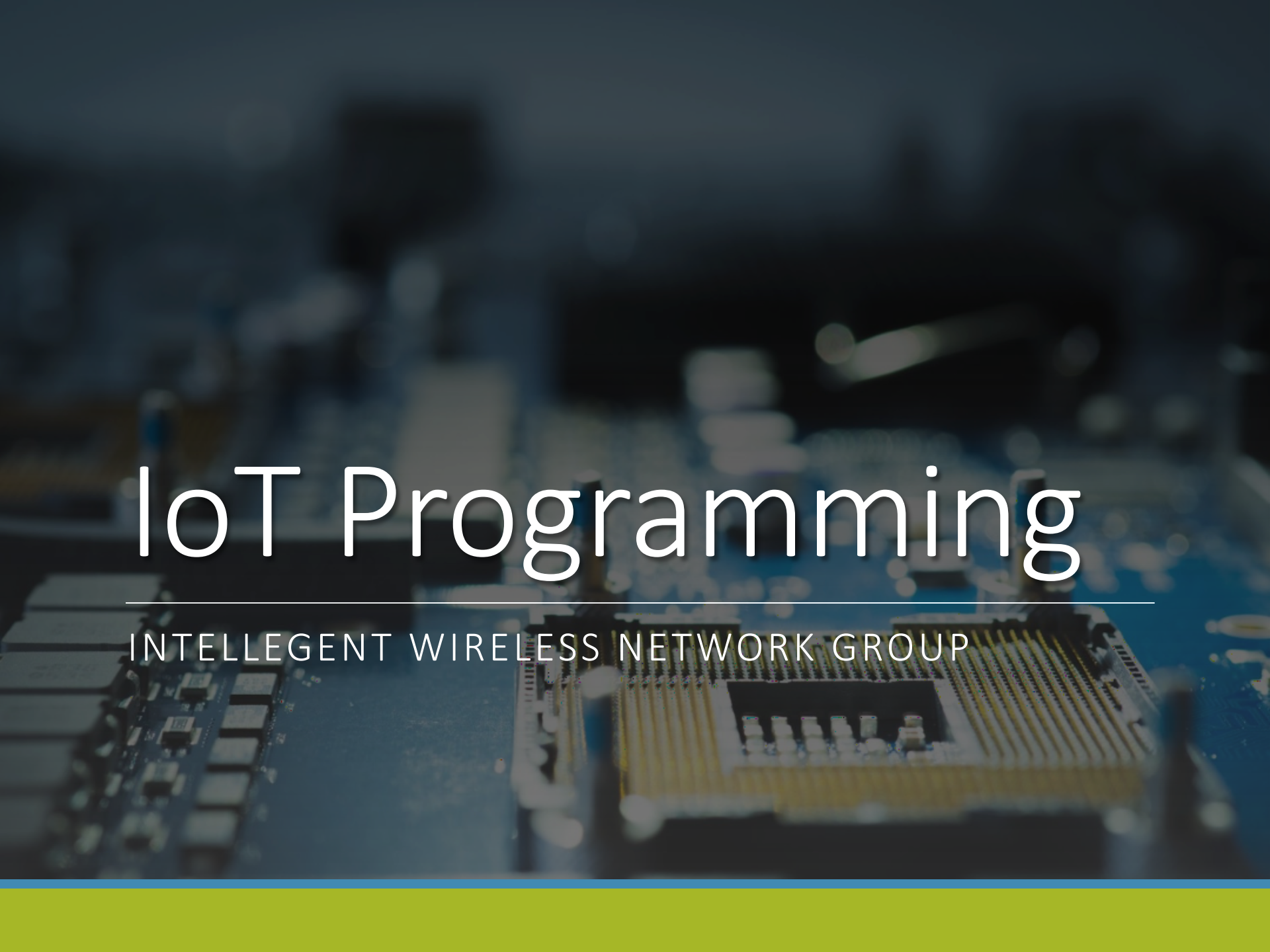


Demonstration: MQTT

- Install MQTT software on your mobile devices

MyMQTT (for Android)

MQTTool (for iOS)



IoT Programming

INTELLEGENT WIRELESS NETWORK GROUP

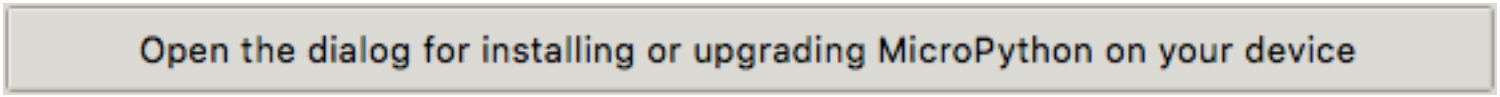
MicroPython IDE Installation

- Download and install **Thonny** from https://thonny.org
- Update Thonny and install ESPTool
  - Go to Tools → Open system shell…
  - At the shell/command prompt, enter

```
pip3 install --upgrade esptool thonny
```

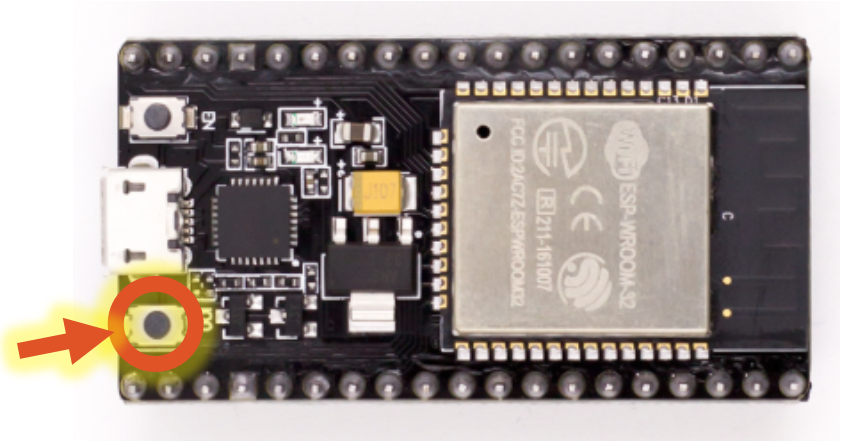
  - Exit the shell and restart Thonny

Flashing MicroPython Firmware

- Download MicroPython firmware from https://ecourse.cpe.ku.ac.th/intro-cpe/iwing/
- Plug in an NodeMCU-32S/ESP32 module
- In Thonny, go to Run → Select interpreter…
- Select **MicroPython (ESP32)** as the interpreter
- Choose the port the ESP32 module is attached to
- Click 

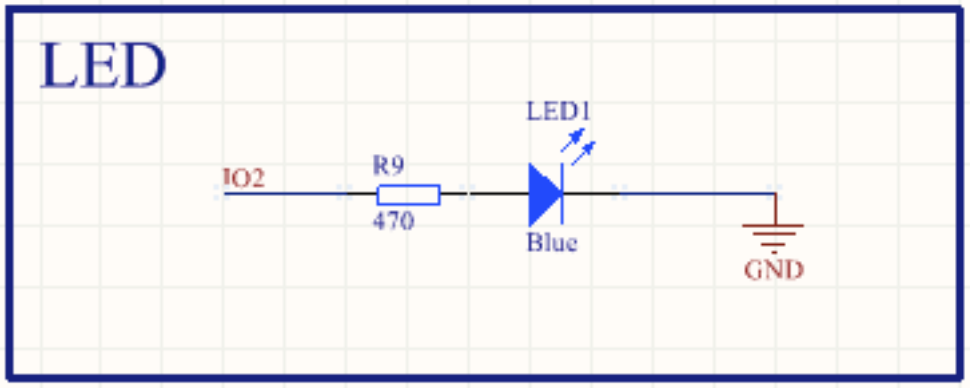
Flashing MicroPython Firmware

- Choose the firmware micropython-1.12.cpe-ku.bin
- Make sure **Erase flash before installing** is checked
- Hold down the IO0 button to force device into boot loader mode after reset, then click Install



Setting Up Built-in LED

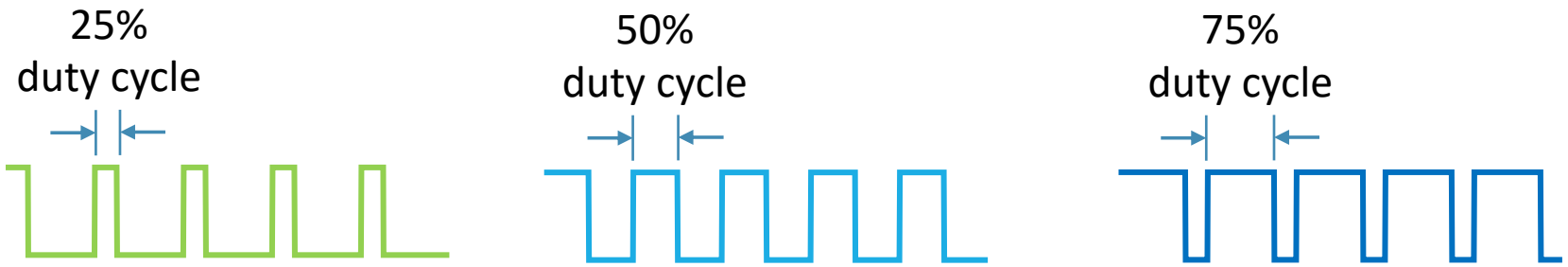
- On-board LED attached to GPIO2




```
from machine import Pin
led = Pin(2, Pin.OUT)
led.value(1)  # turn LED on
led.value(0)  # turn LED off
```

Analog Control for LED

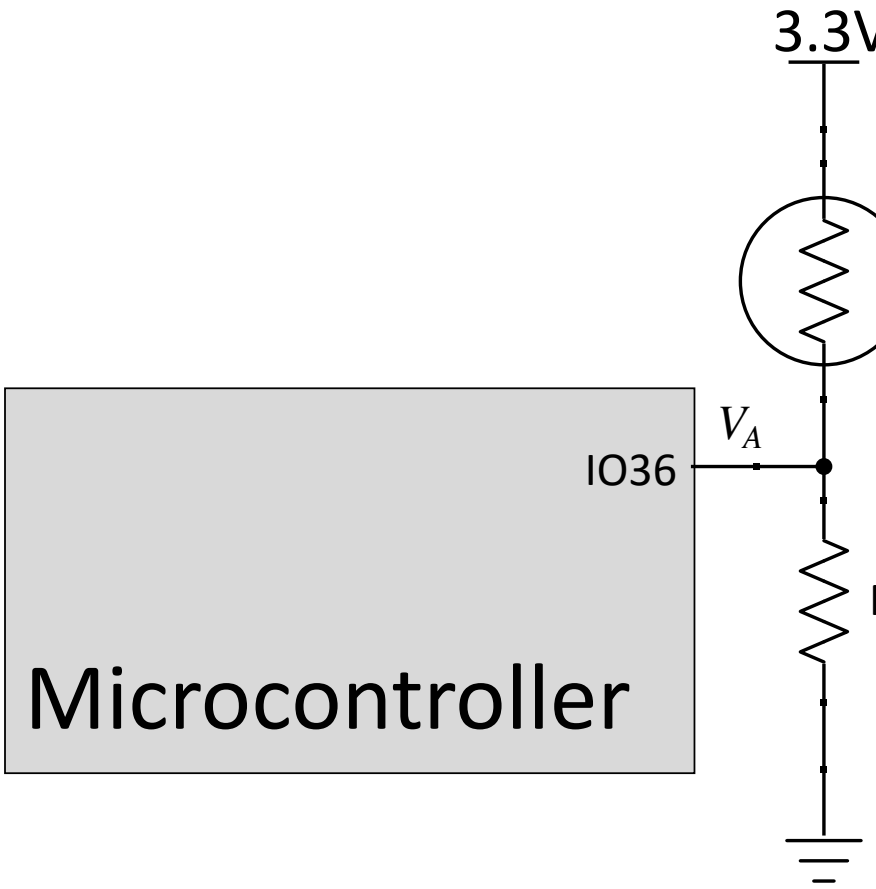
- LED can be used as an analog output using PWM (pulse-width modulation)



- Examples:

```
from machine import PWM
pwm = PWM(led)
pwm.freq(1000)

pwm.duty(0)     # LED off
pwm.duty(1023)  # LED on with full brightness
pwm.duty(512)   # LED on with 50% brightness
```

Light Sensing Circuit



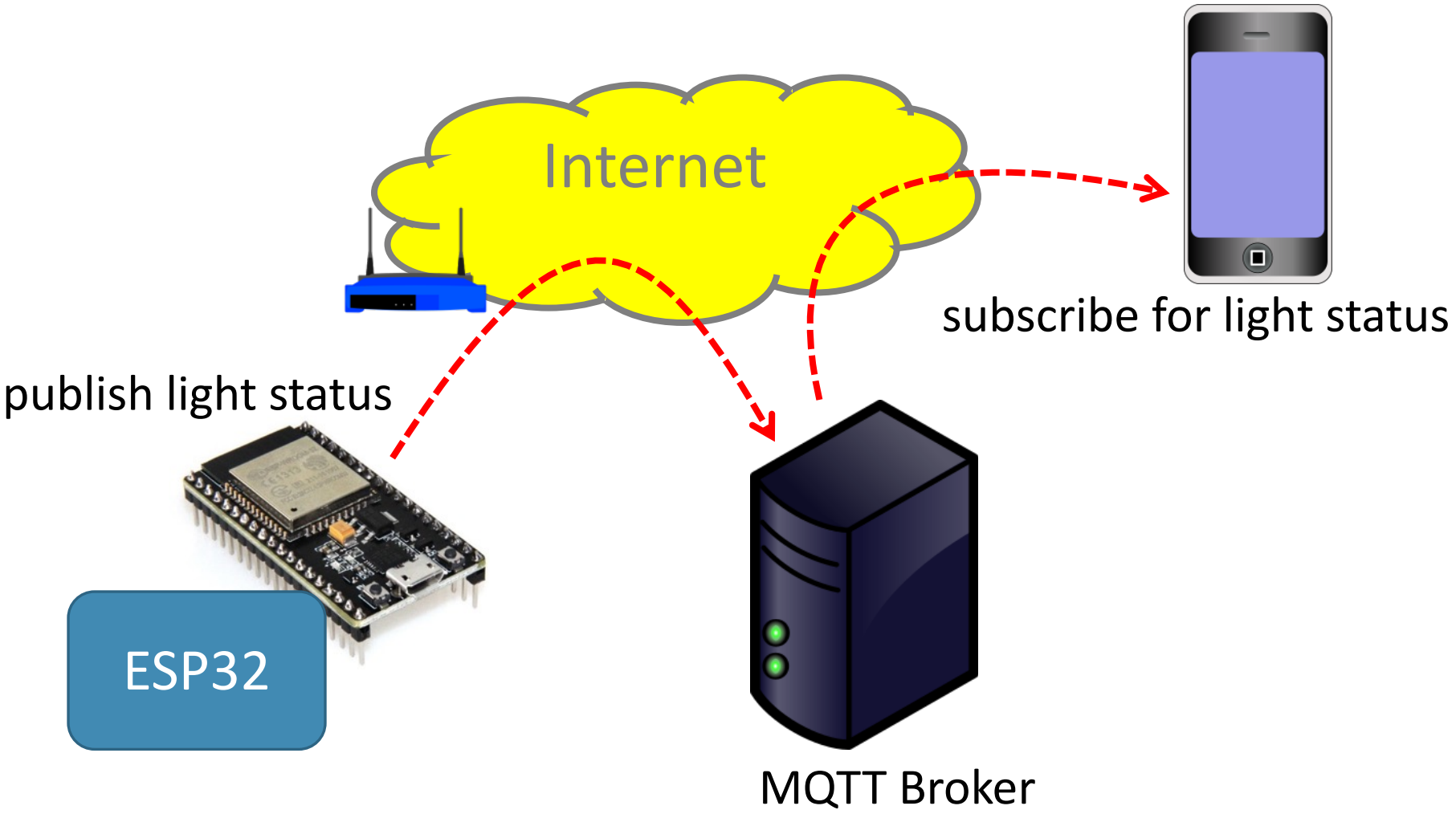
By Ohm's Law,

$$V_A = 3.3 \times \frac{R_2}{R_2 + R_{LDR}}$$

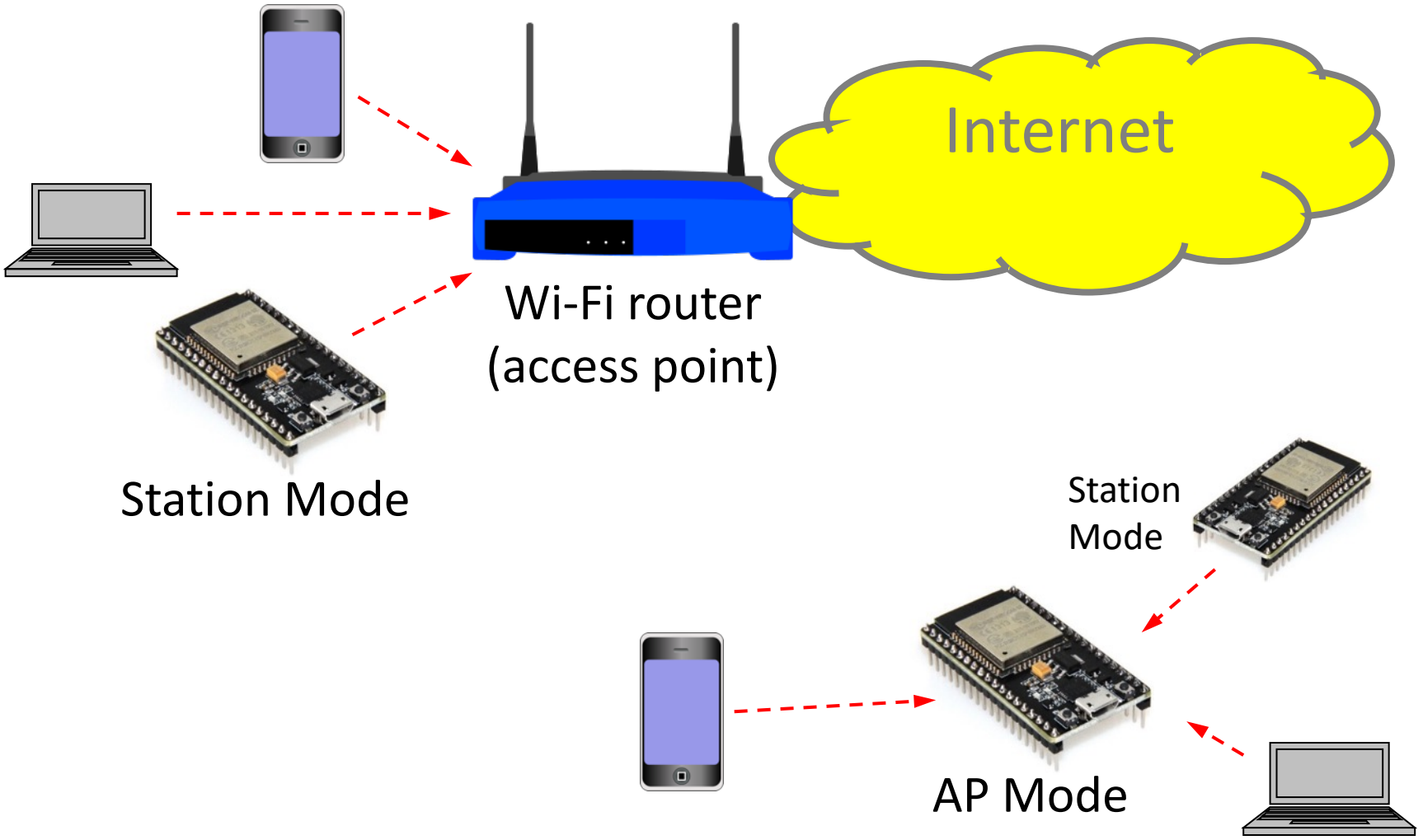
LDR Setup

```
from machine import Pin, ADC

# attach LDR to Pin 36
ldr = ADC(Pin(36))

# set measurement range of 150-2450 mV
ldr.atten(ADC.ATTN_11DB)

print(ldr.read())
```

Scenario 1 – Publishing Data



ESP32 Networking Modes



Connecting to WiFi

```
import time
import network

wifi = network.WLAN(network.STA_IF)
wifi.active(True)
wifi.connect("SSID","PASS")
while not wifi.isconnected():
    time.sleep(0.5)
print('WiFi connected')
```

Connecting to MQTT Broker

```
from umqtt.simple import MQTTClient

mqtt = MQTTClient("UNIQUE-ID","BROKER")
mqtt.connect()
print("MQTT Connected")
```

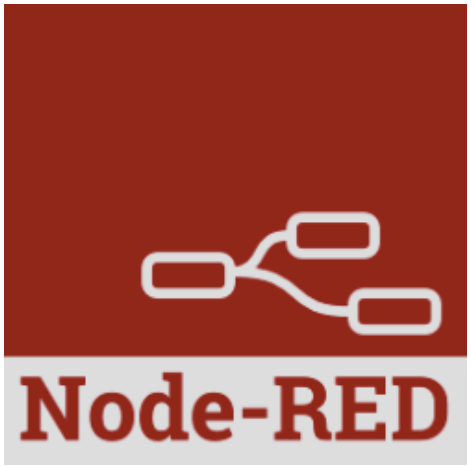
Publishing Data

- The following publishes the knob's value to the topic `ku/cpe/XXXXXX/light` every 5 seconds

```
while True:
   value = ldr.read()
   mqtt.publish("ku/cpe/XXXXXX/light",str(value))
   time.sleep(2)
```

Node-RED

- Flow-based programming tool for wiring together:
  - Hardware (e.g., IoT) devices
  - Files/databases connectivity
  - APIs
  - Online services
  - UI dashboard
- Originally developed by IBM
- Built on Node.js
- Additional nodes can be installed, created, and shared with other developers

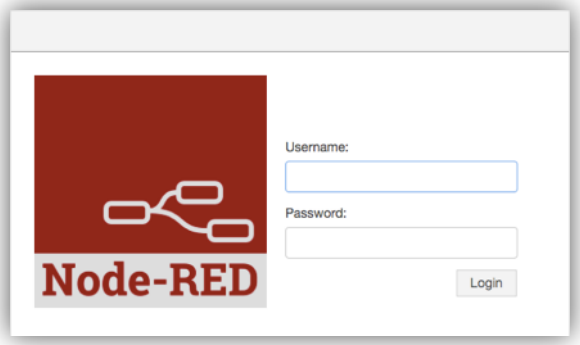


Accessing Your Node-RED Instance

- Each of you has access to a personal Node-RED instance running inside a Docker container
- Point your browser to

  https://158.108.34.31/red/*<ku-account>*

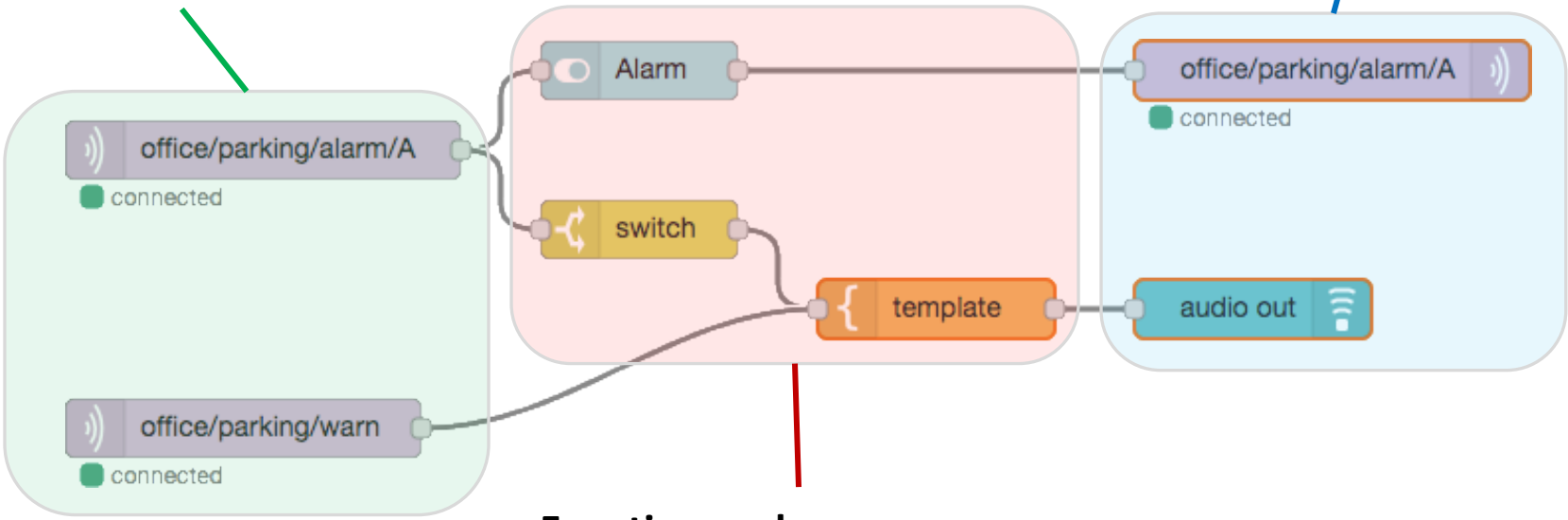
- The site's security certificate is self-signed, so please accept the warning and continue
- The password is the same as your KU account's password



Typical Node-RED Flows

Input nodes
accept input from outside and generate messages

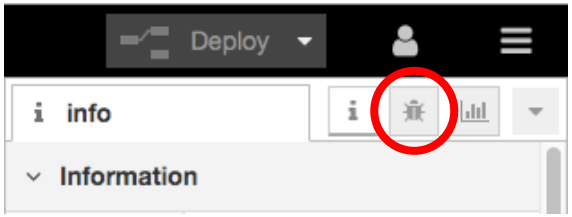
Output nodes
send messages' data to outside



Function nodes
process/manipulate messages

Your First Flow: Hello, World!

- Add an **inject** node and a **debug** node
- Wire the Inject node's output to Debug node's input
- Double-click the Inject node to change the payload type and content
- Save and run the flow by clicking 

- Switch to the Debug tab 

- Click the Inject node's button and observe the result

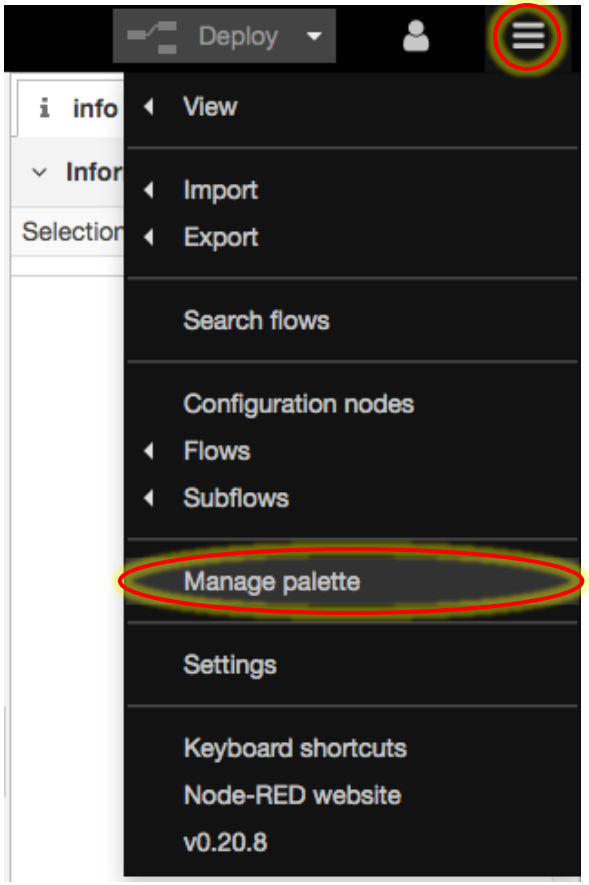
MQTT Interaction

- Node-RED comes with two built-in MQTT nodes
  - MQTT subscribe node (input)

  - MQTT publish node (output)

Installing More Nodes

- Use the menu Manage palette to manage and install your nodes



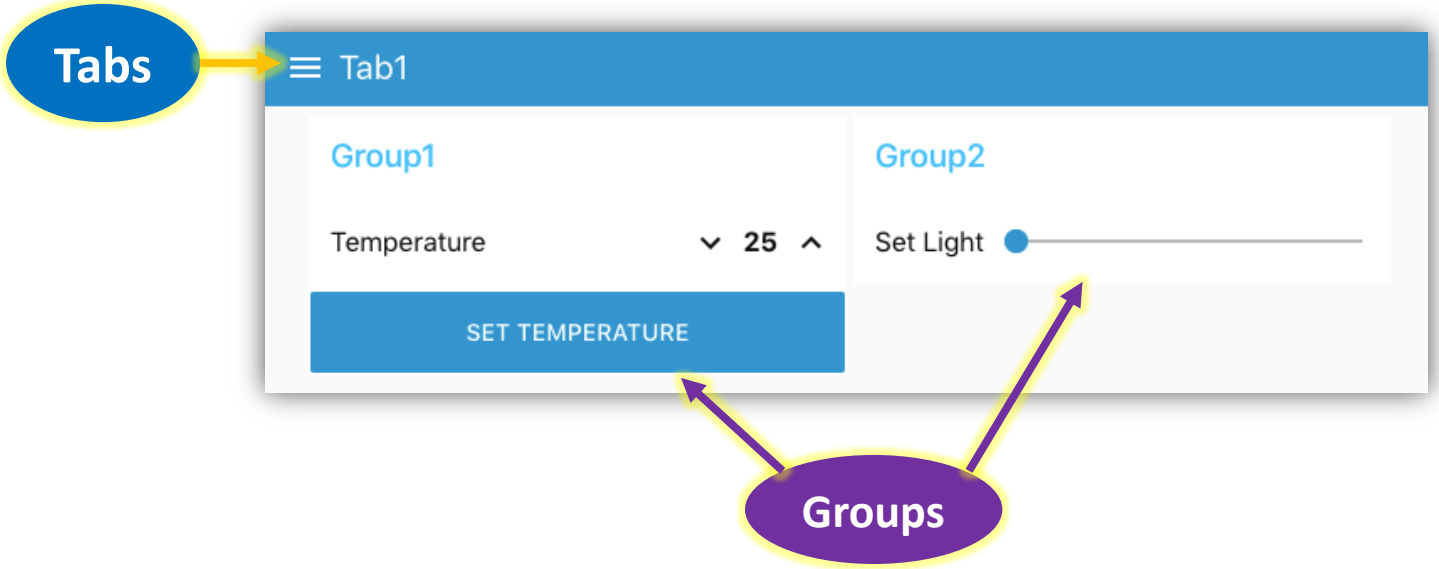
Node-RED Dashboard

- Installed with **node-red-dashboard** package
- Allows simple, yet highly customizable UI based on Angular and Material Design

https://flows.nodered.org/node/node-red-dashboard

UI Groups and Tabs

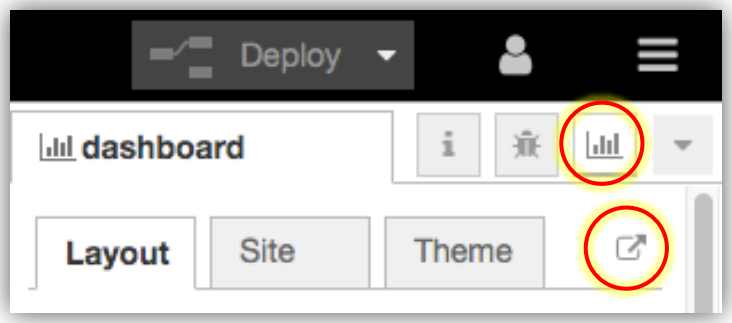
- UI dashboard consists of one or more tabs
- Each dashboard tab consists of one or more groups



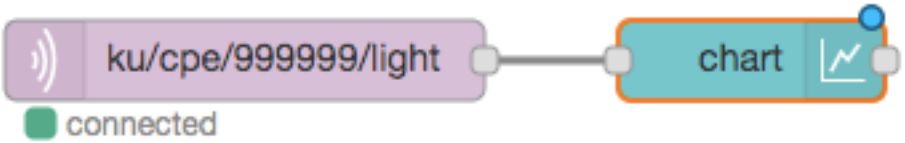
- When adding the very first UI widget, you will need to create a new UI group and tab for it

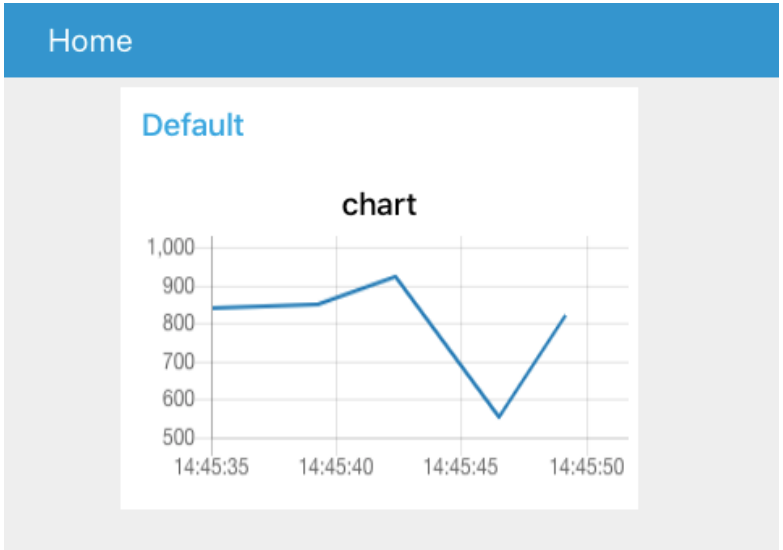
Dashboard Page

- To open the dashboard page, select the dashboard tab and click the link icon

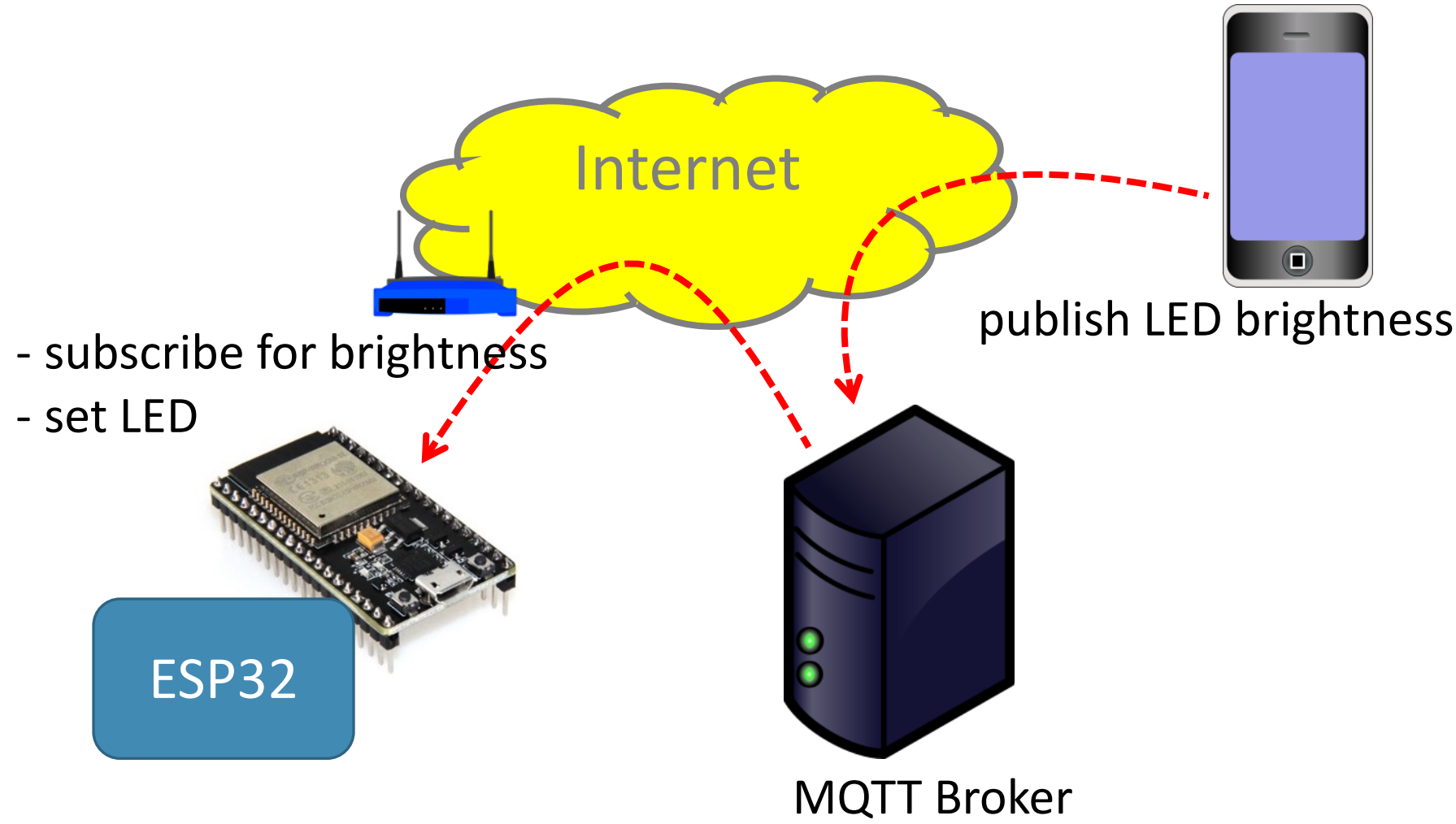


Example: Light Chart





Scenario 2 – Subscribing



Subscribing to a Topic

```
def sub_cb(topic,msg):
   if topic == b"ku/cpe/XXXXXX/led":
      value = int(msg)
      pwm.duty(value)

mqtt.set_callback(sub_cb)
mqtt.subscribe("ku/cpe/XXXXXX/led")
while True:
      mqtt.check_msg()
      time.sleep(0.2)
```