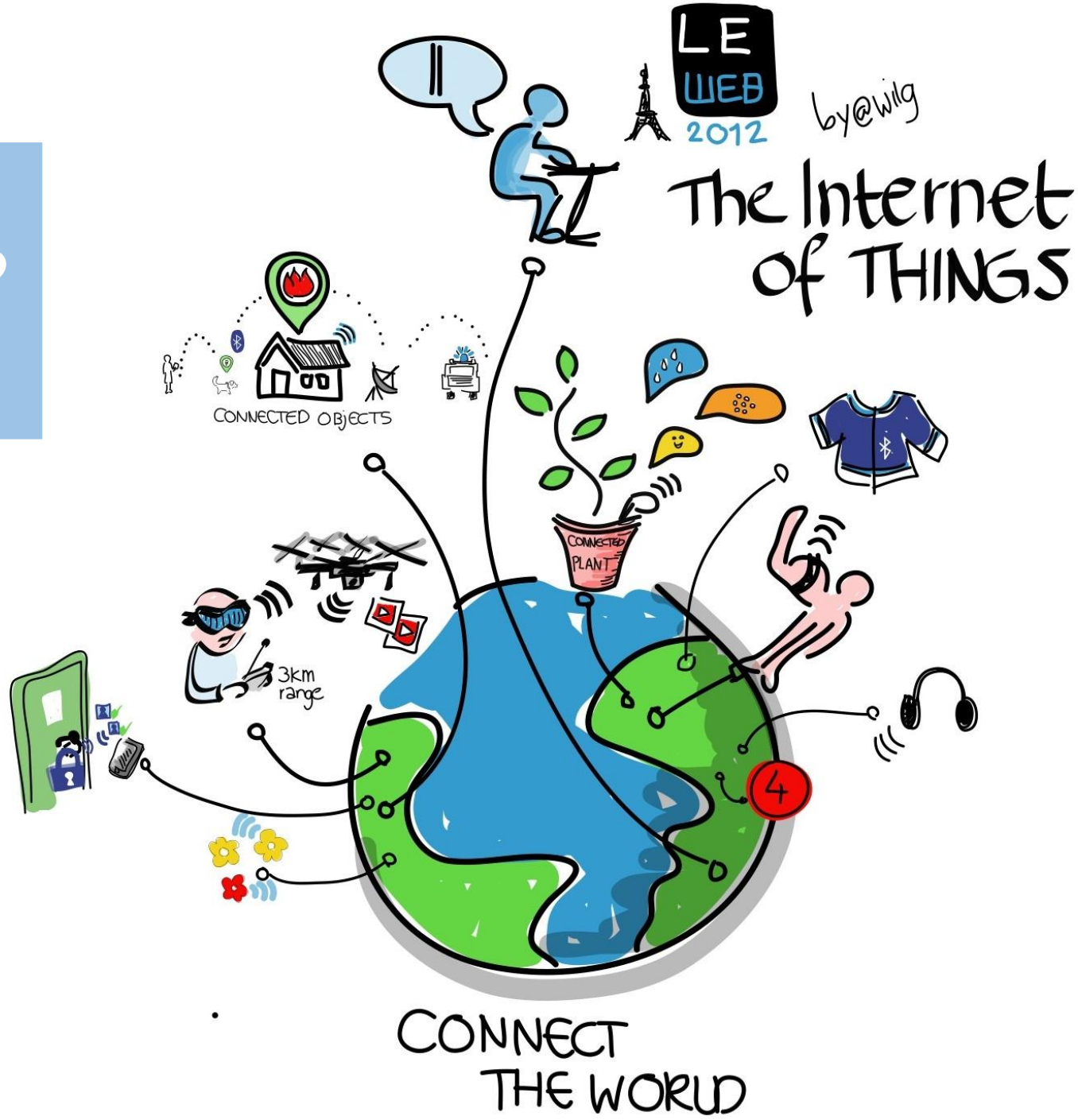
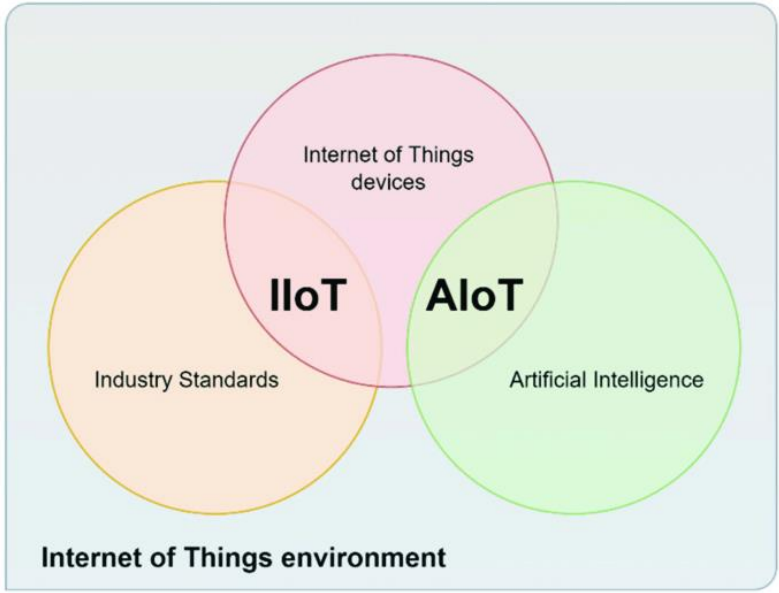
Environment Sensors and IOT for Solar Farm

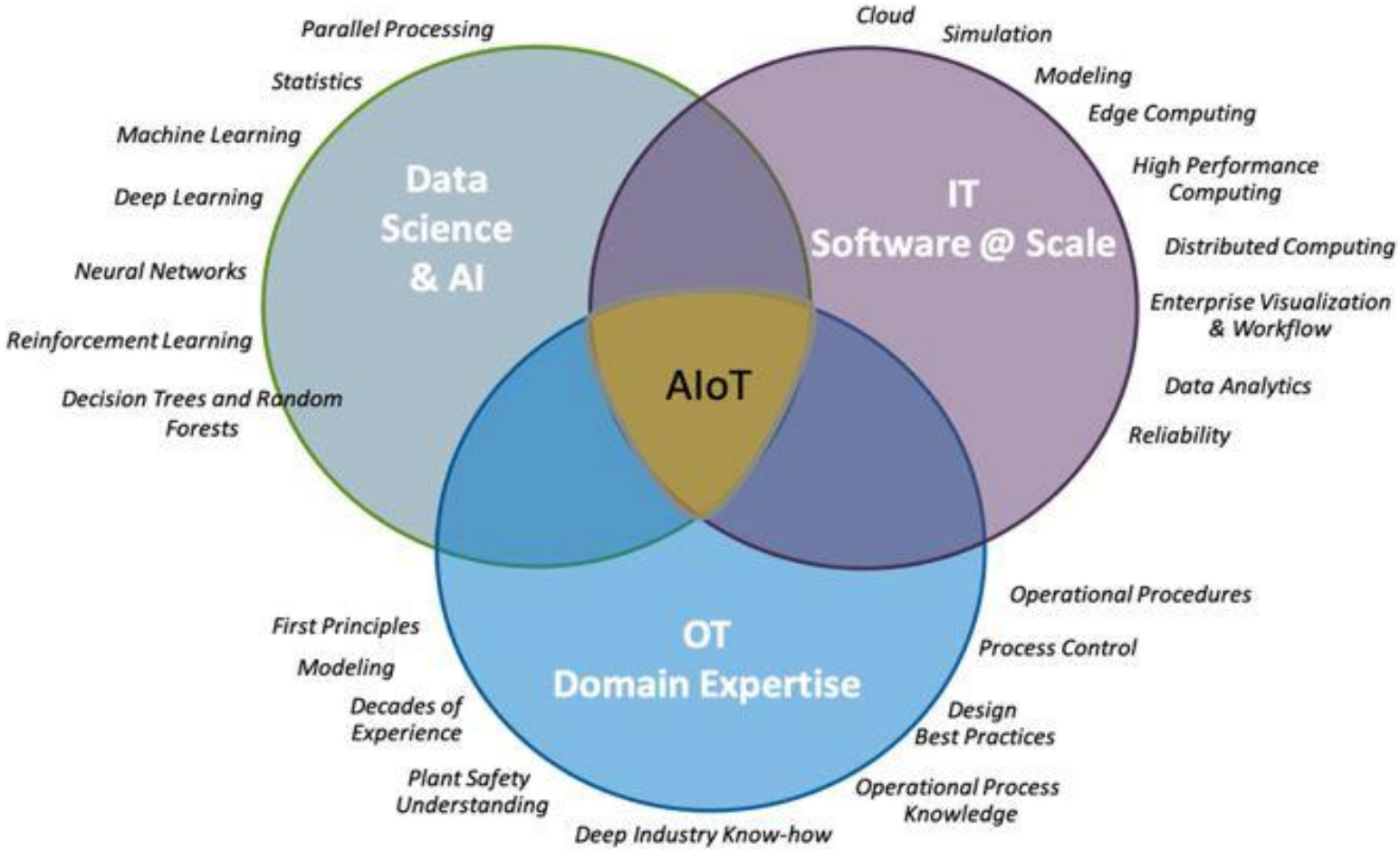
Internet of Things ?



Internet of Things ?

Environment Sensors and IOT for Solar Farm





The Convergence of IT-OT-AI: Democratizing and Embedding AI

Internet of Things : Timeline

Environment Sensors and IOT for Solar Farm

1982 — Carnegie Mellon researchers connect a vending machine to Internet to remotely check for cold sodas

1990 — 1990 the first toaster was operated over the net

1999 — Term "The Internet of Things" (IoT) was coined by Kevin Ashton in his presentation to Proctor & Gamble

2000 — LG introduce first smart fridge

2004 — First smart watch introduced

2007 — Apple smart iPhone and wearable Fitbit released

2009 — Google starts testing self-driven cars

2011 — Smart TV is introduced

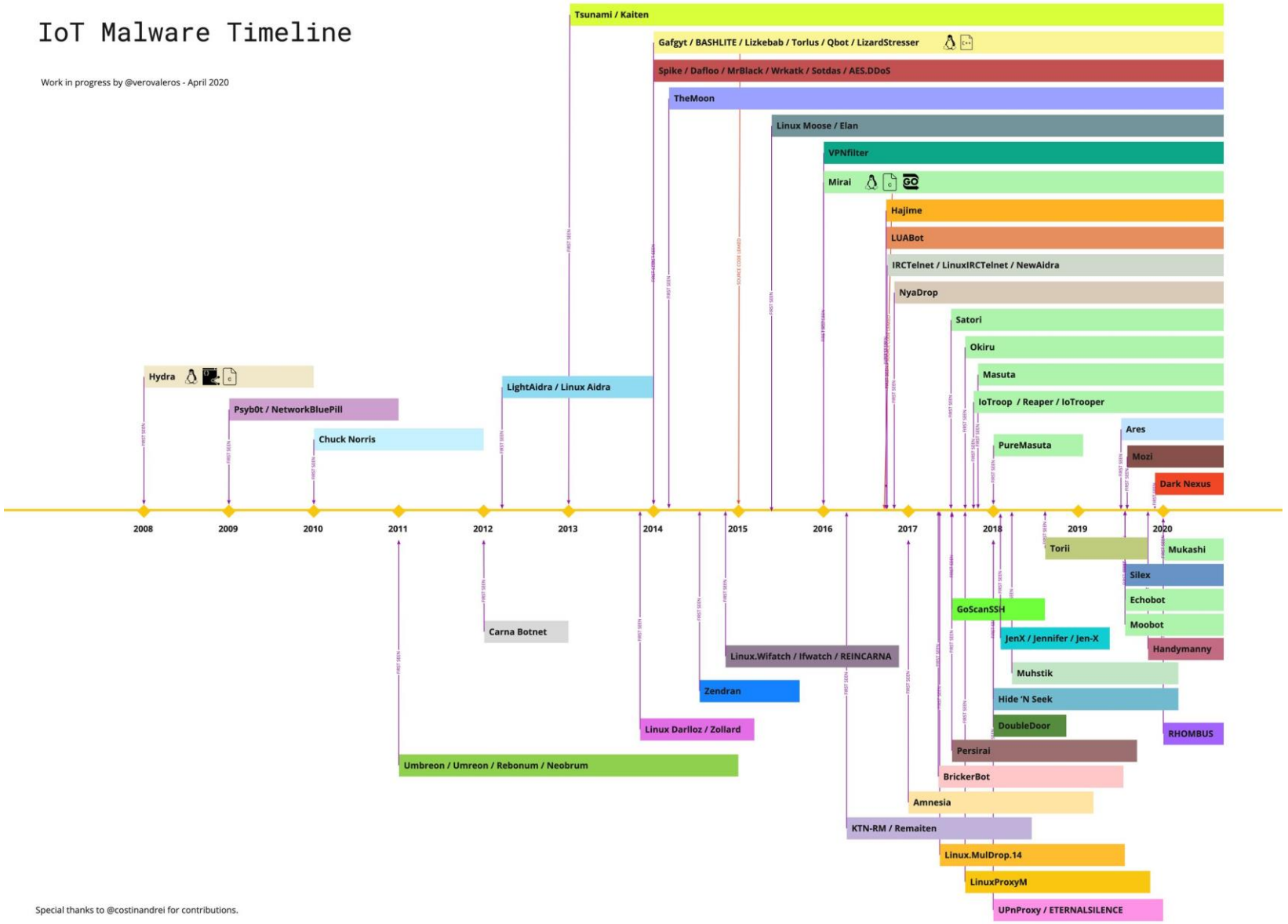
2013 — Google Lens is released

2014 — Echo causes a surge in Smart Home Market

2015 — Tesla comes out with Auto Pilot drive for their cars

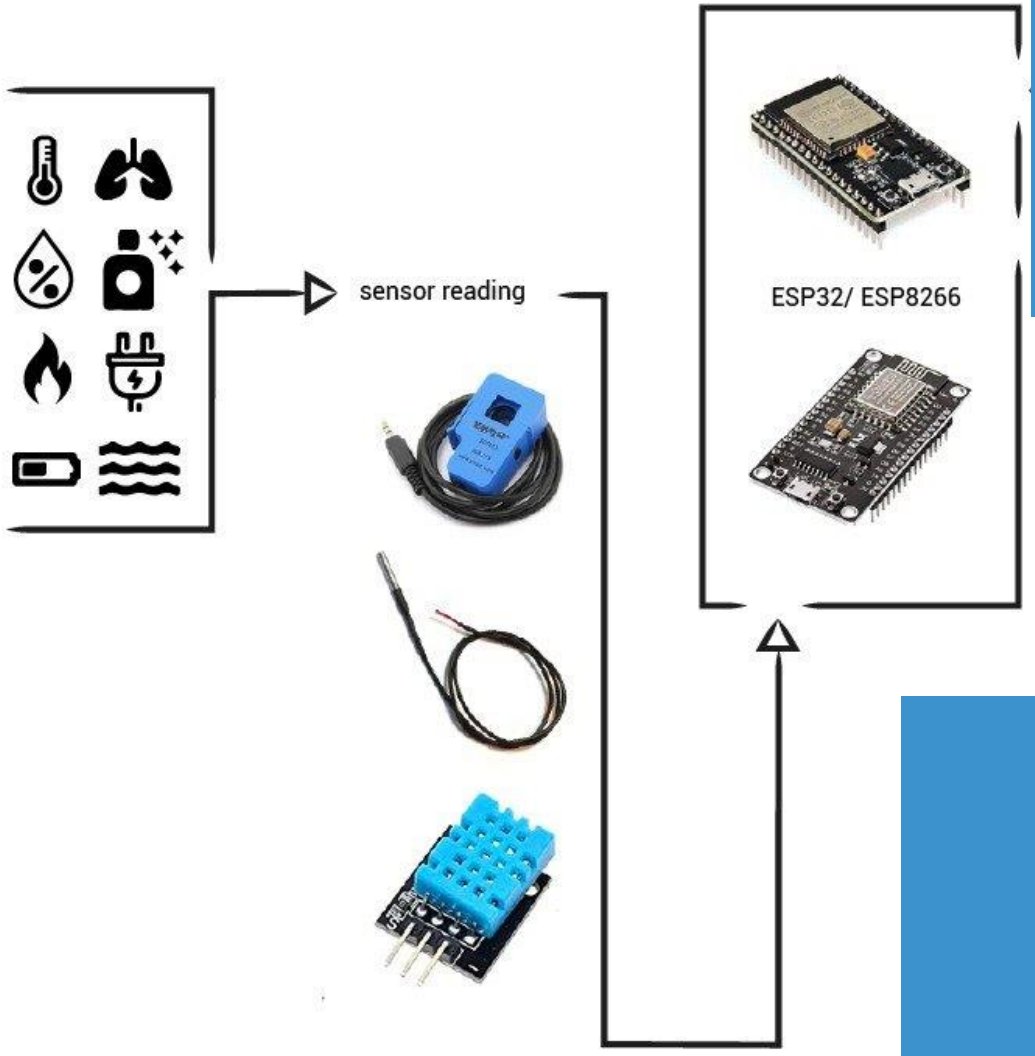
Internet of Things : Malware Timeline

Environment Sensors and IOT for Solar Farm



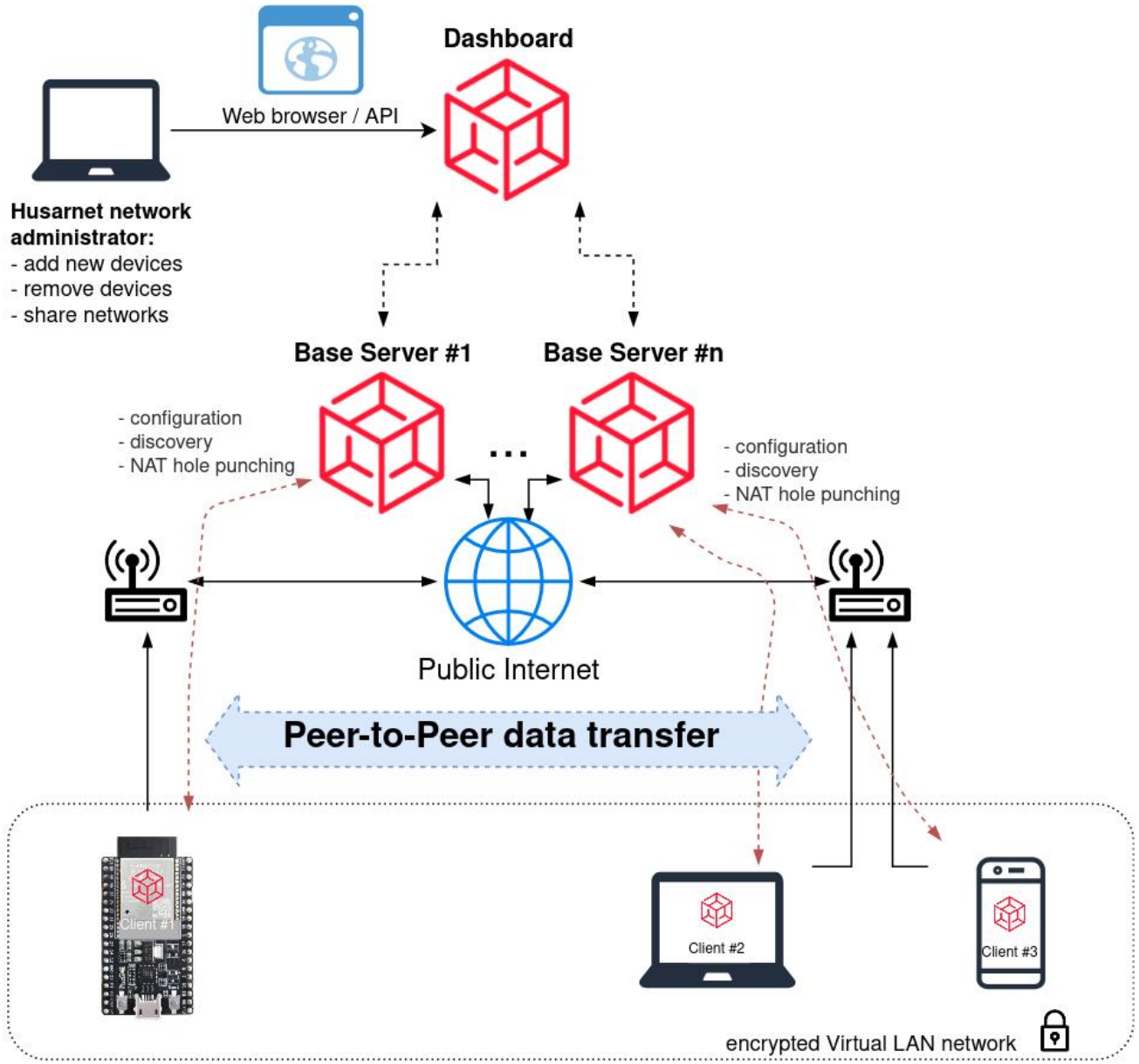
Internet of Things ?

Environment Sensors and IOT for Solar Farm



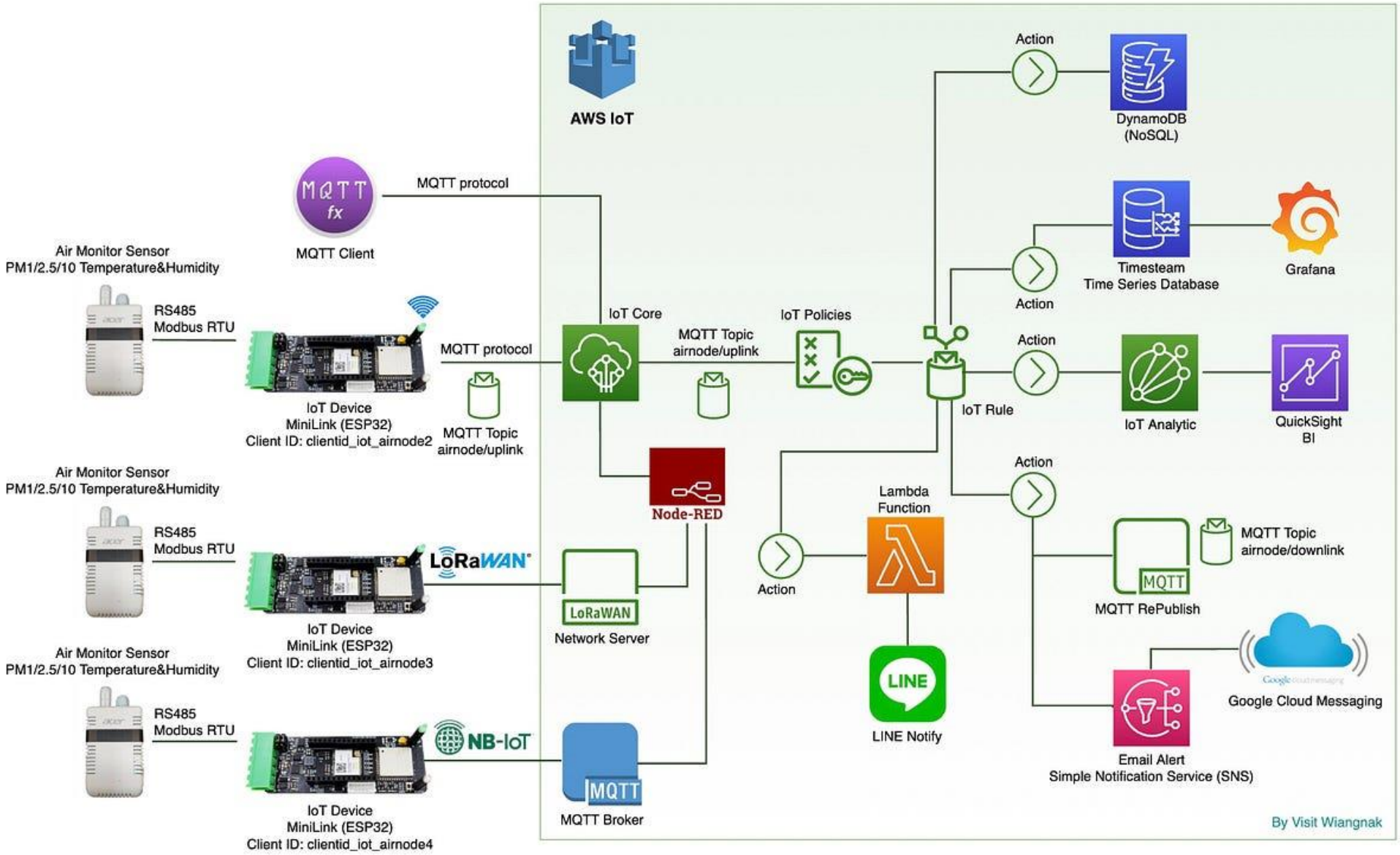
Internet of Things : Architecture

Environment Sensors and IOT for Solar Farm



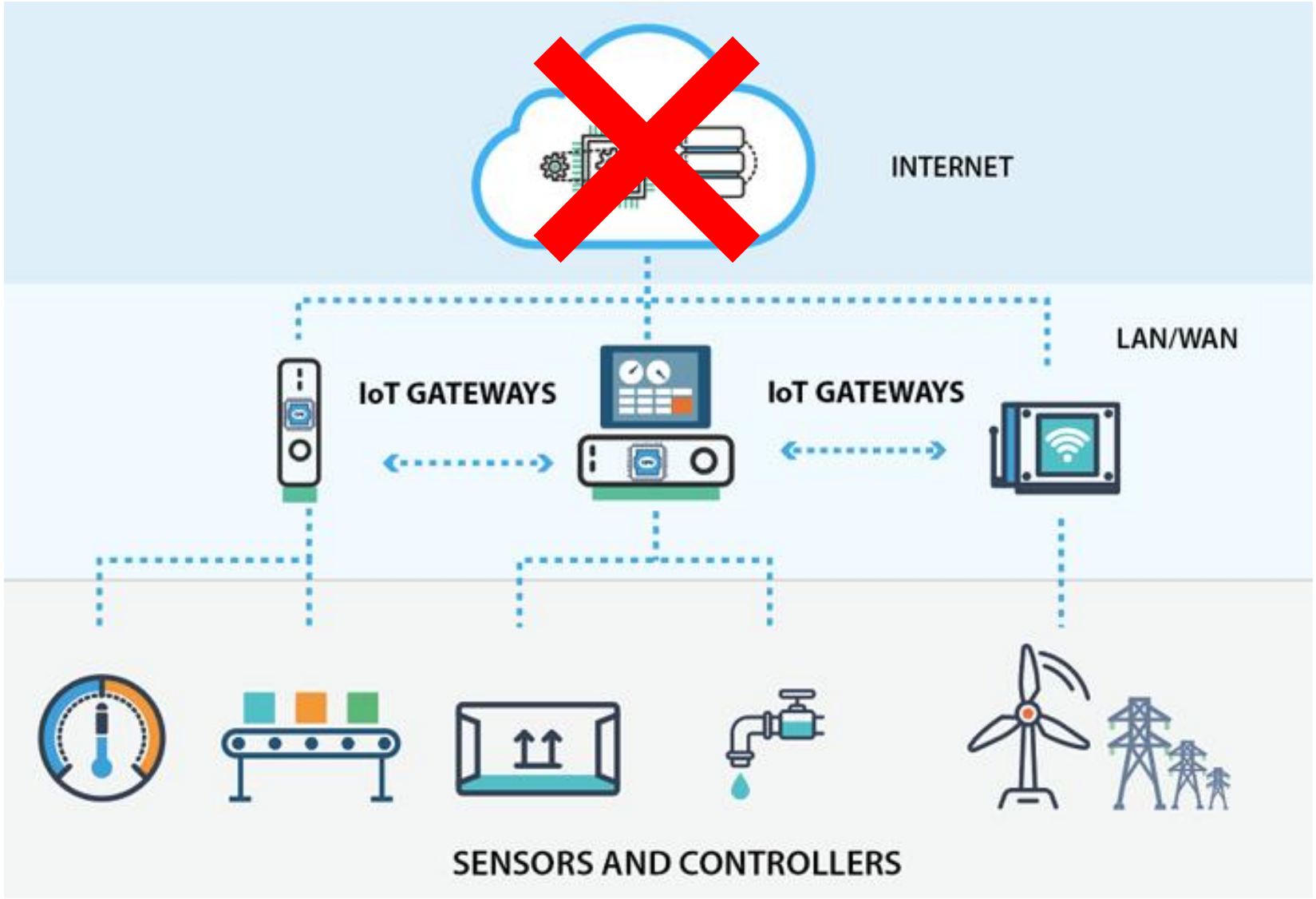
Internet of Things : Architecture

Environment Sensors and IOT for Solar Farm



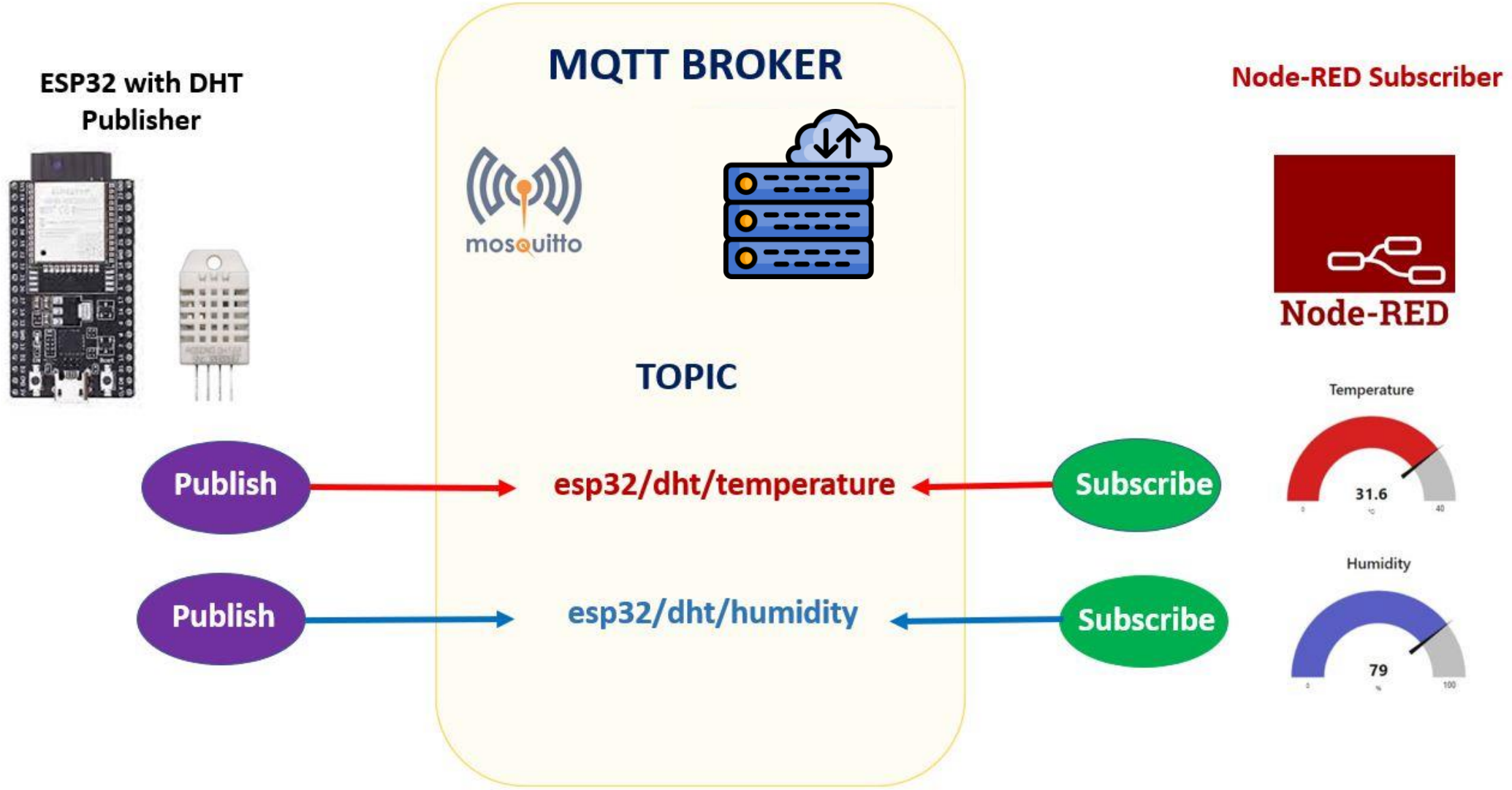
Internet of Things : Architecture

Environment Sensors and IOT for Solar Farm



Internet of Things : Architecture

Environment Sensors and IOT for Solar Farm



Environment Sensors and IOT for Solar Farm

Solar Farm

AJ. Ton

รูปแบบการผลิตกระแสไฟฟ้า

- Solar Rooftop
- Solar Farm
- Solar Floating

ในปัจจุบันการผลิตไฟฟ้าในลักษณะที่ติดตั้งบนหลังคา หรือแสงอาทิตย์รายเล็กมาก (VSPP) ติดตั้งทั้งสิ้น 994 ราย กำลังผลิตติดตั้ง 5.49 เมกะวัตต์ และมีแนวโน้มจะเพิ่มขึ้นเรื่อย ๆ เนื่องจากการส่งเสริมแบบ FIT ใน พ.ศ. 2560 และราคาระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์นั้นไม่สูงมาก ส่งผลให้ภาคประชาชนเริ่มหันมาสนใจติดตั้งแผงเซลล์แสงอาทิตย์บนหลังคามากขึ้น

เซลล์แสงอาทิตย์บนหลังคา หรือ Solar Rooftop
(ที่มา : https://www.egat.co.th/egattoday/)

โรงไฟฟ้าแสงอาทิตย์ หรือ Solar Farm

(ที่มา : http://www.sunpro-solar.net/article/10/ระบบโซลาร์ฟาร์ม-solar-farm)

Solar Farm

มีด้วยกัน 2 แบบหลักคือ

- แบบยึดติดอยู่กับที่ (Fixed System)
- แบบหมุนตามดวงอาทิตย์ (Tracking System)
  - สามารถรับแสงอาทิตย์ได้ดีกว่าแบบอยู่กับที่ถึง 20%

Solar Floating หรือการติดตั้งแผงเซลล์แสงอาทิตย์บนผืนน้ำ

เนื่องจากมากกว่า 30% ของพื้นที่ประเทศไทย เป็นพื้นที่น้ำและพื้นที่ชุ่มน้ำ เพิ่มประสิทธิภาพกำลังการผลิตได้ถึง 5% – 20% จากความเย็นของน้ำ

มีแนวโน้มการใช้งานเพิ่มมากขึ้น เนื่องด้วยความได้เปรียบในด้านประสิทธิภาพที่เพิ่มขึ้นประมาณ 11%

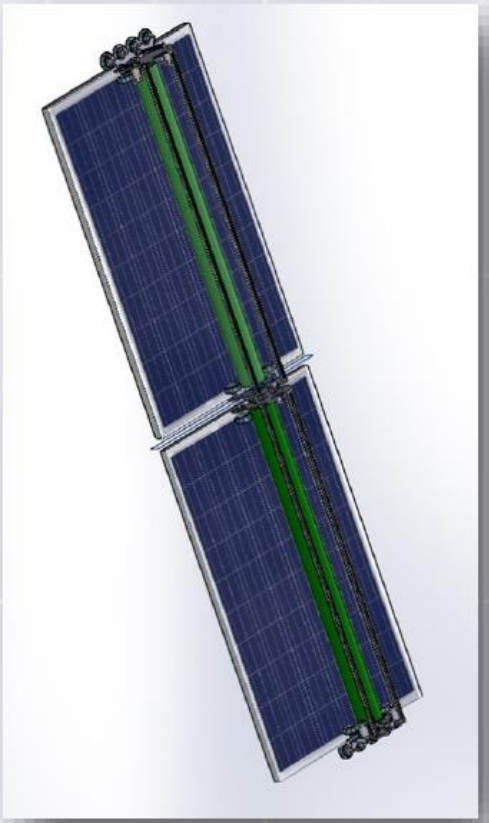
แผงเซลล์แสงอาทิตย์บนทุ่นลอยน้ำ

(ที่มา : https://www.scg.com/innovation/floating-solar-farm/)

Different Designs for Suitable Jobs

Total Analysis for Solar Panel Cleaning Robot

Model No. 1 | Solar Farm Cleaner | Details



Total Analysis for Solar Panel Cleaning Robot

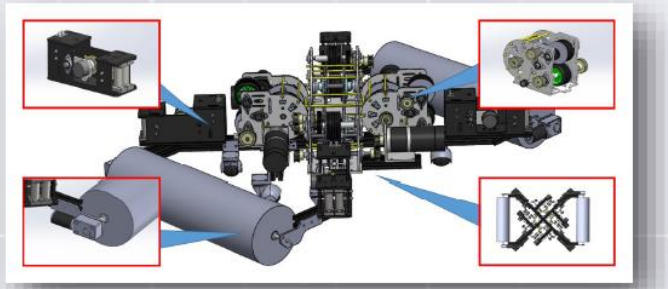
On–Sites Test Results

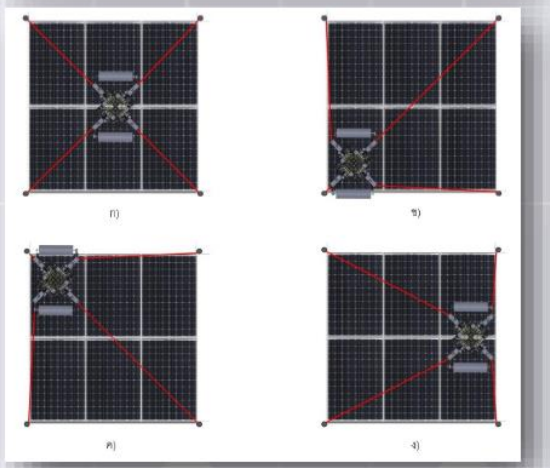
Model No. 1

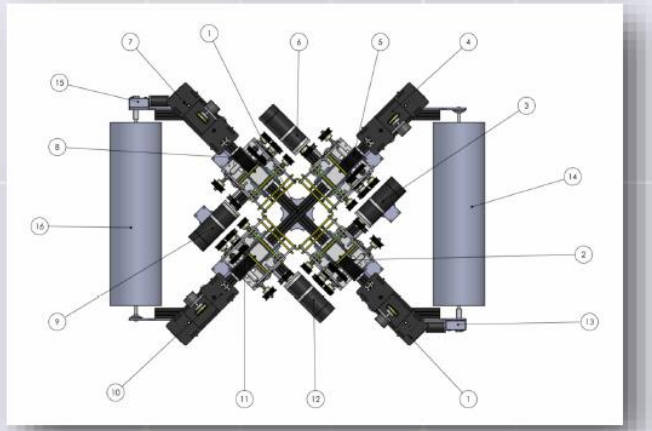
Before & After

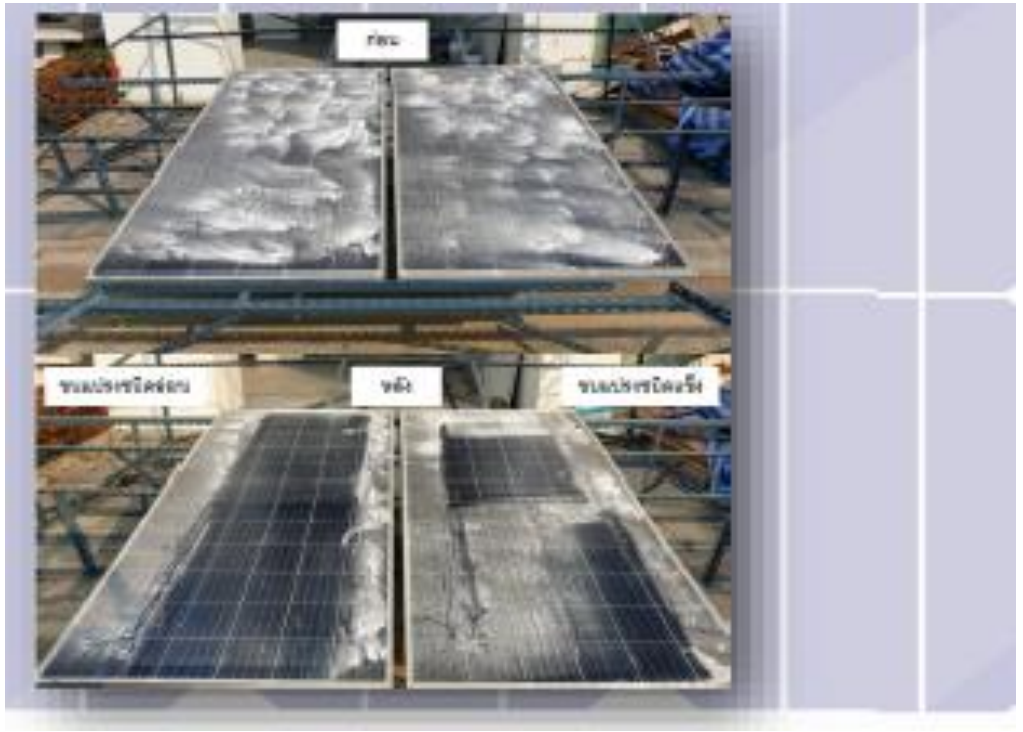
Total Analysis for Solar Panel Cleaning Robot

Model No. 2 | Workspace Adjustable Cleaner | Details









Environmental factors that affect solar panel efficiency

Introduction

Solar energy production is influenced by many factors. This factors will reduce or hinder the production of electricity for solar panels. [Nawaf, 2014]

1.Shadowing

[Sullivan,2011]

2. High environmental temperature

[Hill, 2013]

3. Dust, dirt, bird droppings

[Kasim, 2010]

Problems

Dust, dirt, bird droppings

Solar energy production will be influenced by dust or dirt that can reduce the electricity generation efficiency for solar panels.

According to Natthakan's research, the dust accumulation will affect the performance of solar power plants and production efficiency ratio. In that investigation, the efficiency of the solar panel is decreased by 1.6-3% for dust accumulation in 1 month and 6-8% for dust accumulation in 2 months. However, after cleaning the dust on solar panels, the efficiency ratio increased by 10%

Cleaning helps reduce dust problems and maintain electricity production efficiency.

Problems

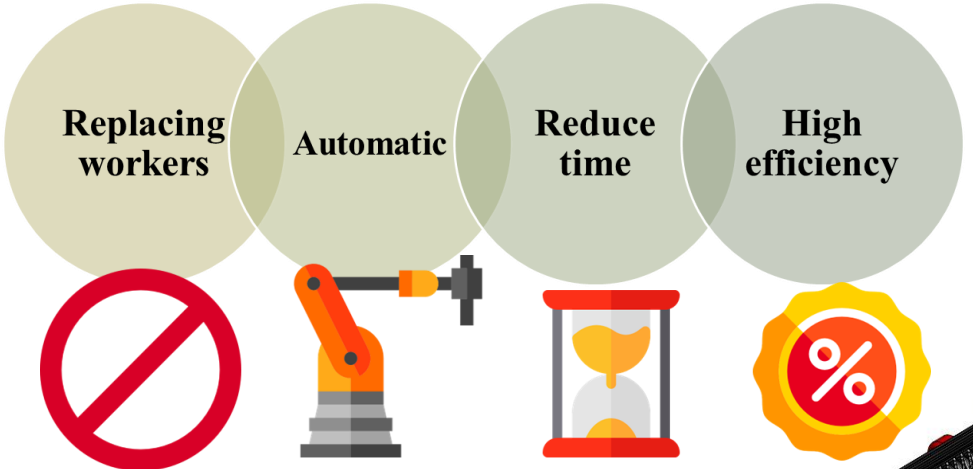
In Thailand, most of solar panel cleaning is manual nowadays.

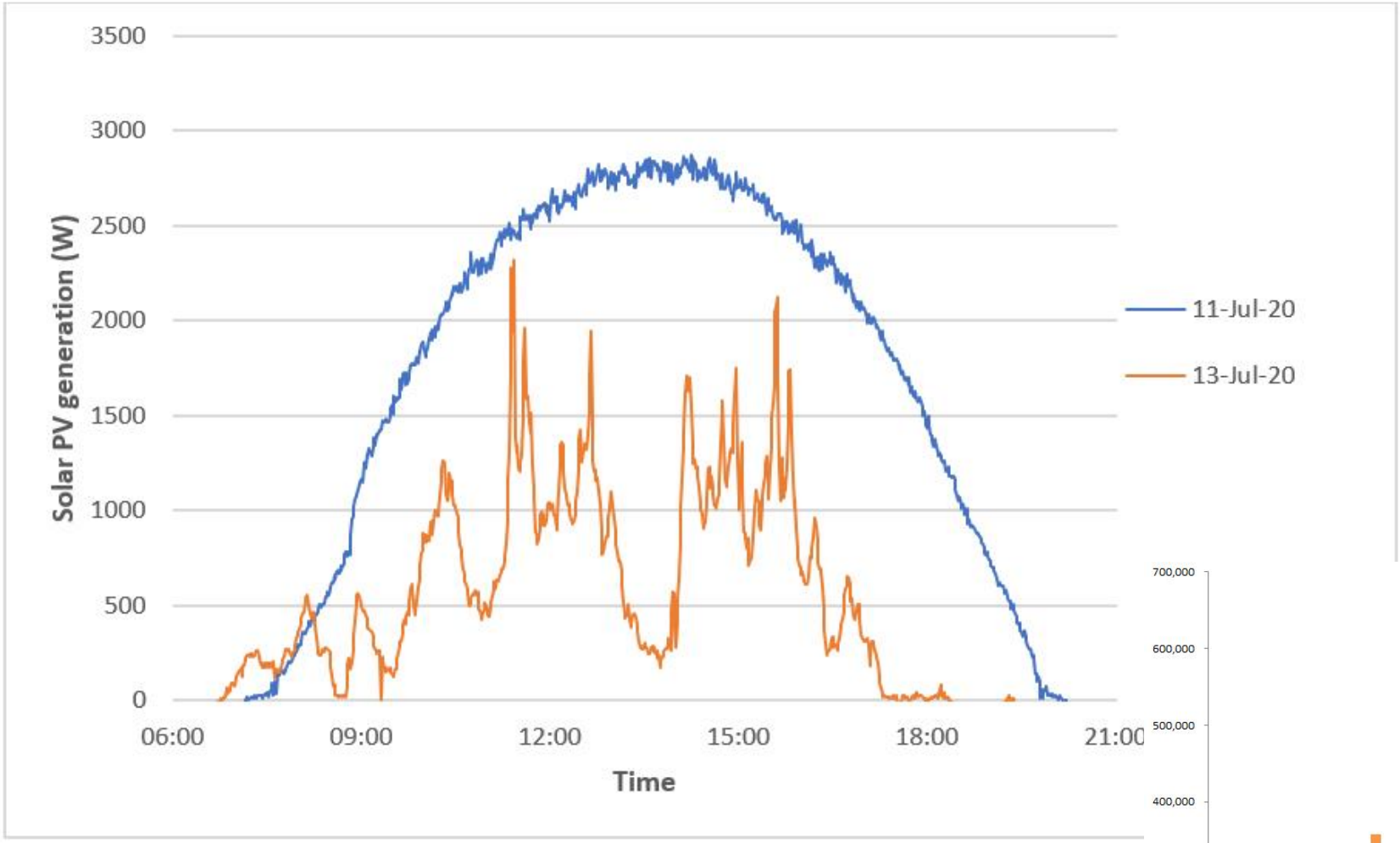
- **Hard work.**
- **Labor shortage.**
- **Expensive wages.**
- **Low efficiency.**
- **Inconsistent quality.**

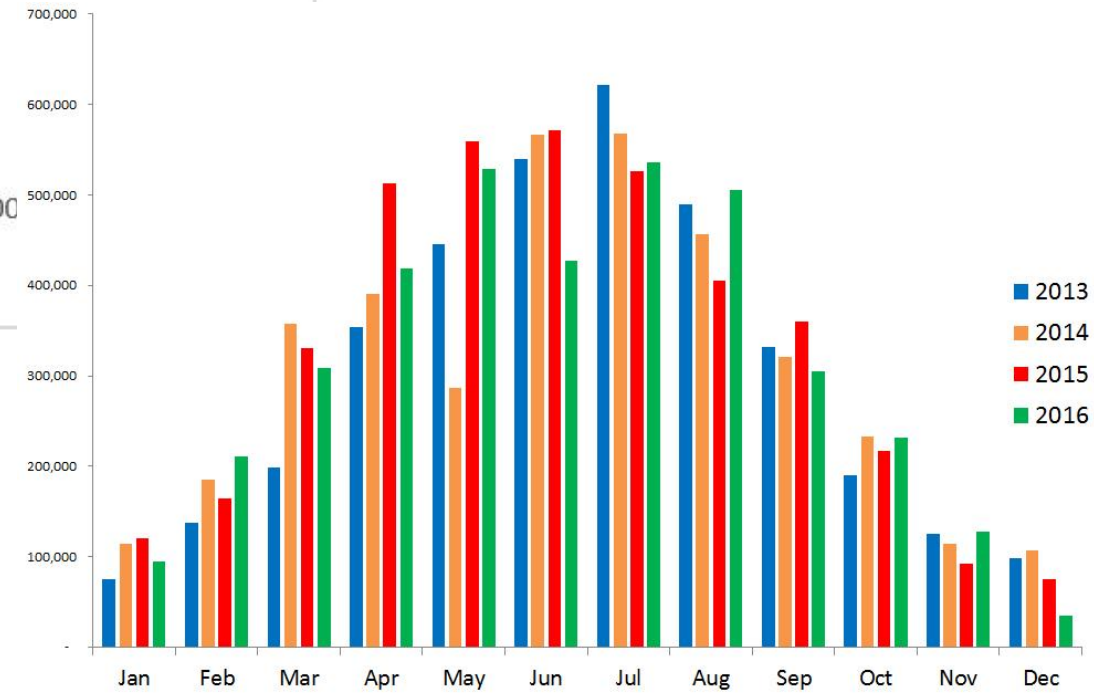
VS

Solutions

Development of a Modular Robot for Cleaning Solar Cell Panels.



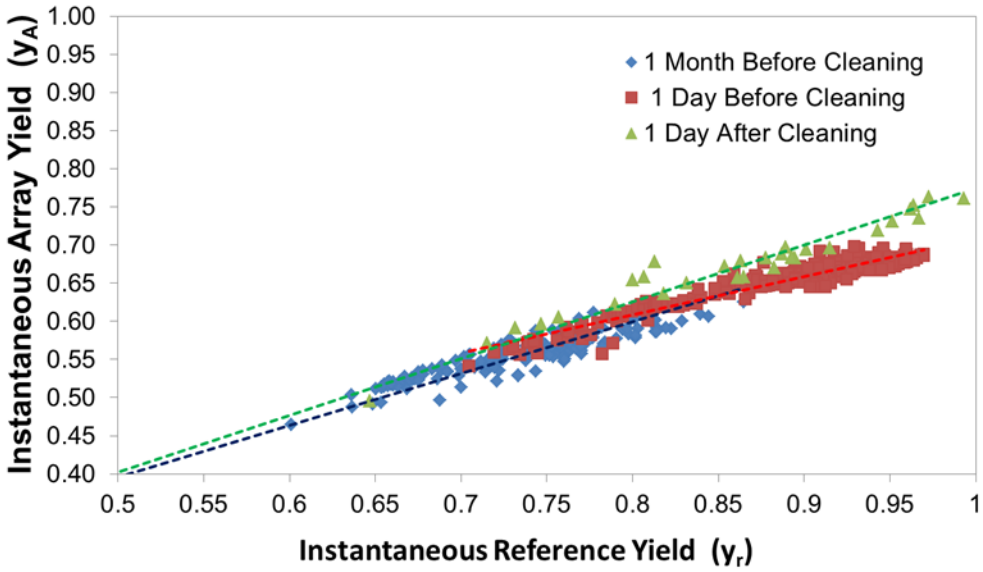

3500
3000
2500
2000
1500
1000
500
0
Solar PV generation (W)
06:00
09:00
12:00
15:00
18:00
21:00
Time
11-Jul-20
13-Jul-20


700,000
600,000
500,000
400,000
300,000
200,000
100,000
-
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2013
2014
2015
2016

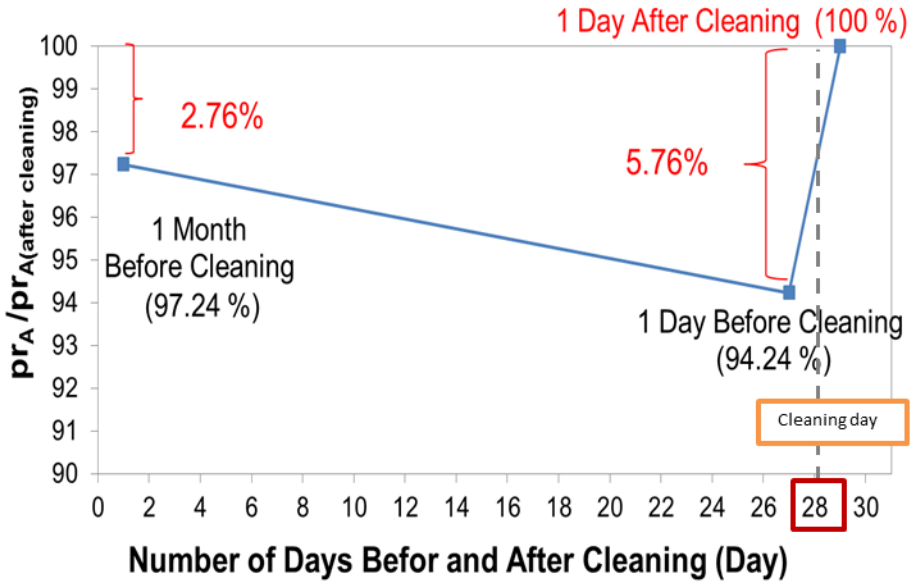
การวิเคราะห์ข้อมูลพลังงานและ Performance ratio

ระบบเซลล์แสงอาทิตย์ที่ติดตั้งบนหลังคา (ระบบที่ 1)

ความสัมพันธ์ระหว่างค่า instantaneous array yield และค่า instantaneous reference yield



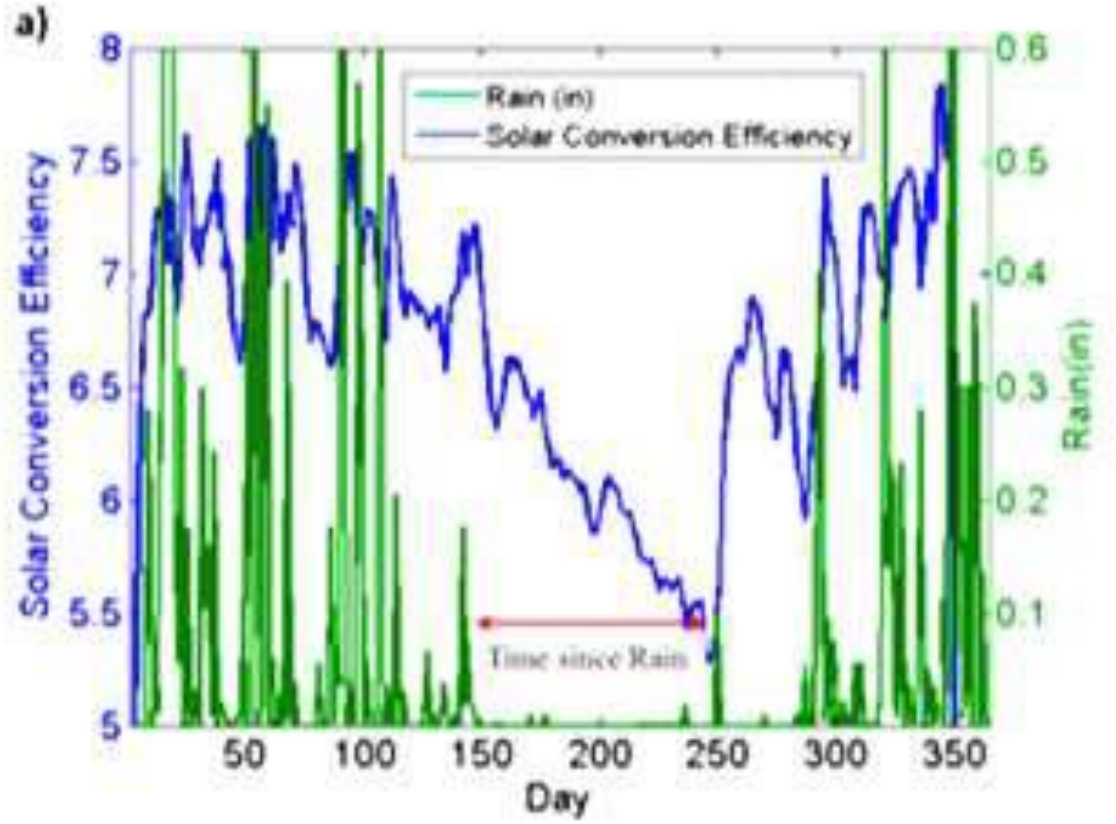
ความสัมพันธ์ระหว่างค่า instantaneous array performance ratio (%) ที่ลดลง และจำนวนของวันก่อนทำความสะอาดและหลังทำความสะอาด



- ข้อมูลในช่วงเดือน กุมภาพันธ์ - มีนาคม ปี 2013

1. N. Sakarapunthip, D. Chenvidhya, S. Chuangchote, K. Kirtikara, T. Chenvidhya and W. Onreabroy, “Effects of dust accumulation and module cleaning on performance ratio of solar rooftop system and solar power plants”, Japanese J. Phys, 56, 2017.

2. B. Liamphrachan, Ph. Thamkaew, K. Buayai and K. Kerdchuen, “The Study for Performance Ratio Increasing of Soiling Solar PV Systems”, EENET J., vol 1. 2017.



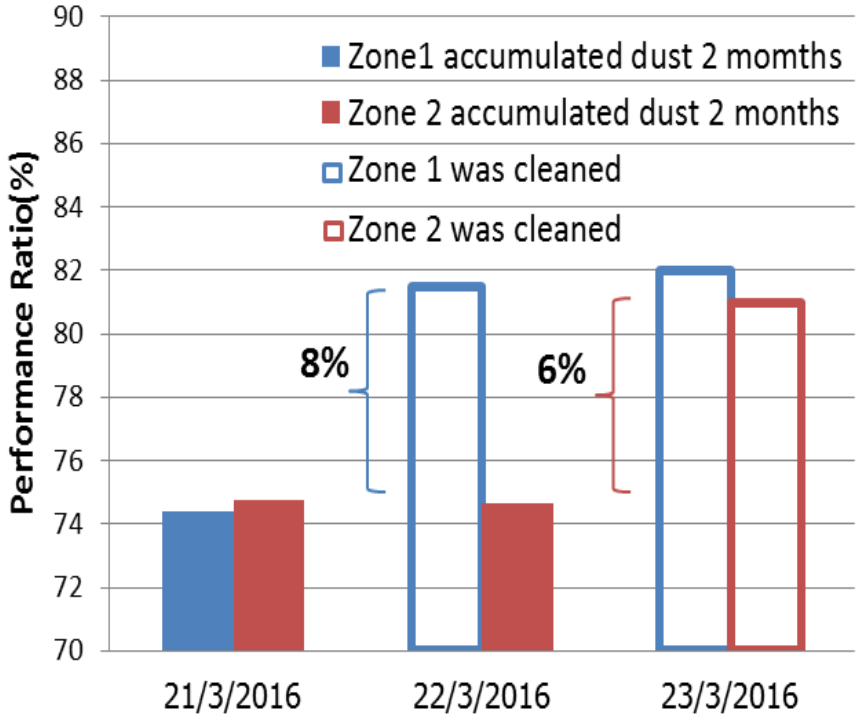
กราฟความสัมพันธ์ระหว่างวันและประสิทธิภาพของเซลล์แสงอาทิตย์และปริมาณฝน

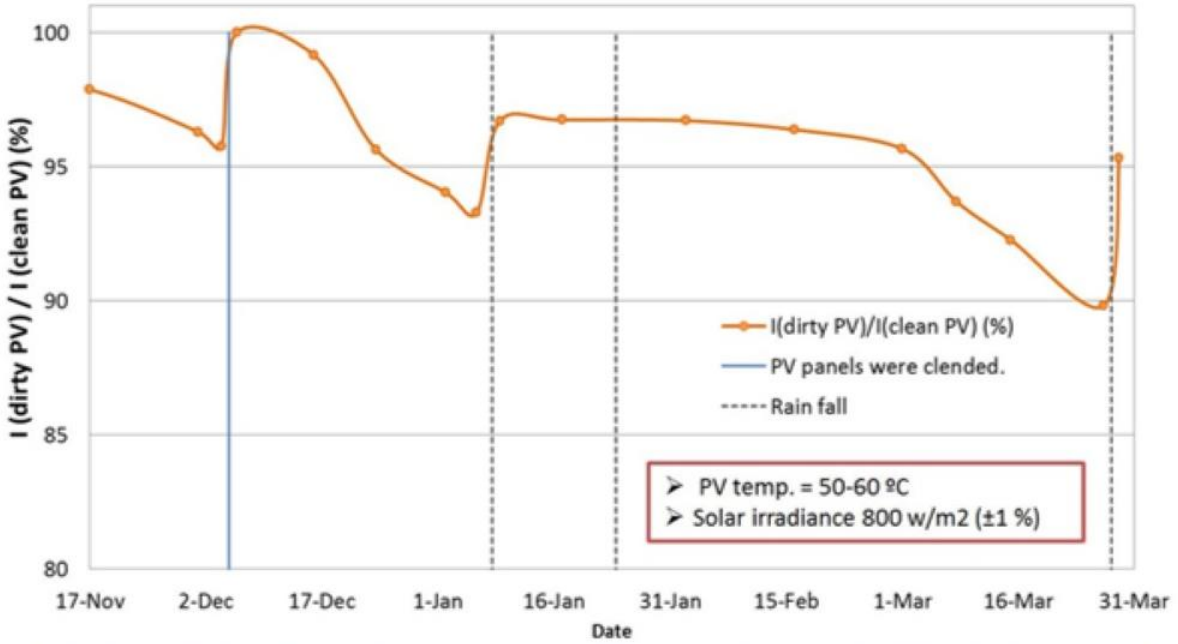
ประสิทธิภาพของเซลล์มีแนวโน้มลดลงอย่างชัดเจนในช่วงที่ไม่มีฝน ทั้งนี้กลุ่มวิจัยของ S. Mekhilef และคณะ ทำการศึกษาผลกระทบของฝุ่น ความชื้น และความเร็วลมที่มีผลต่อประสิทธิภาพของเซลล์แสงอาทิตย์ พบว่า ผลกระทบจากฝุ่นที่สะสมที่มีต่อการลดลงของประสิทธิภาพของแผงเซลล์จะอยู่ในช่วงที่กว้าง

ประสิทธิภาพของการทำความสะอาด

ค่าพลังงานที่สูญเสียเนื่องจากการสะสมฝุ่น
2 เดือน ประมาณ 426 kWh/MWp
คิดเป็น 6-8%

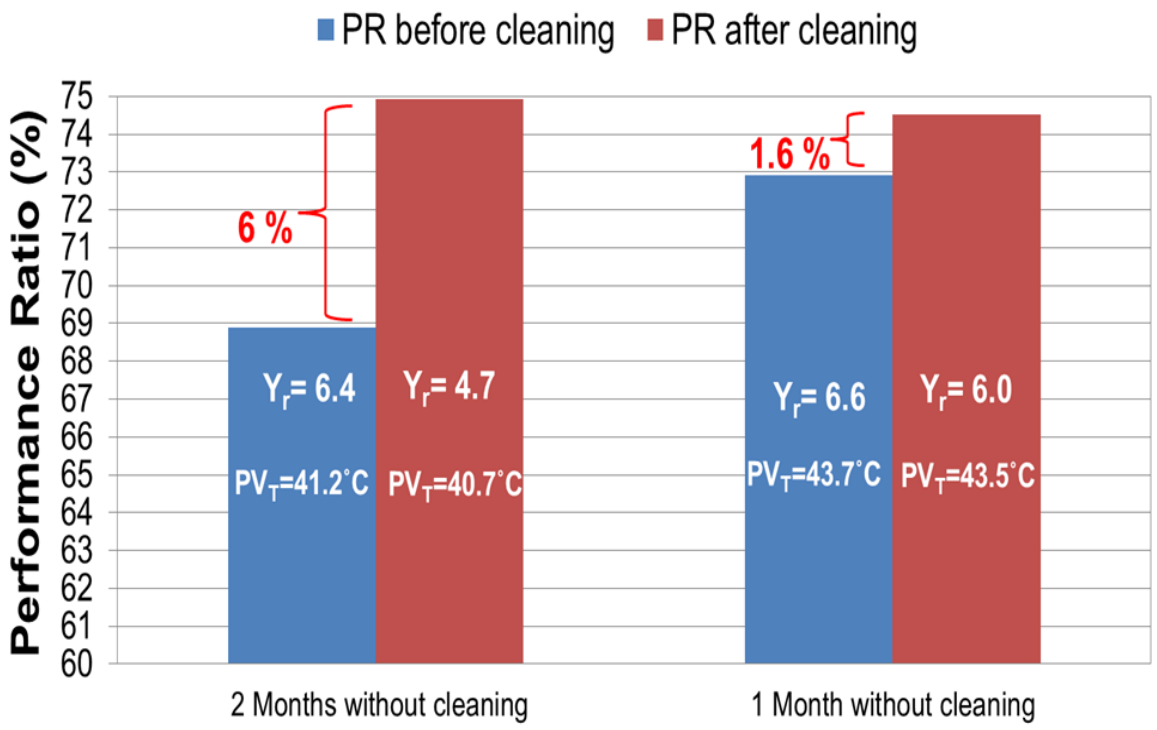
โรงไฟฟ้าในจังหวัดนครปฐม



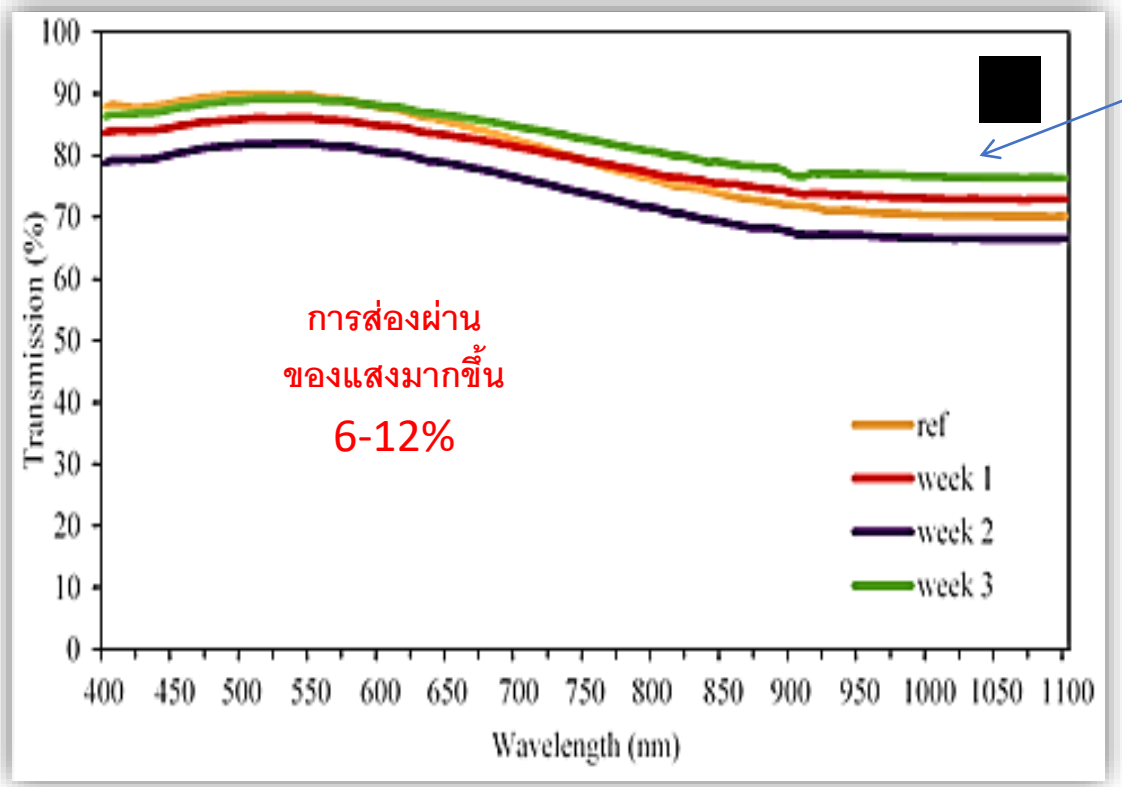


During 4 months, PV current reduced approximately 10 %

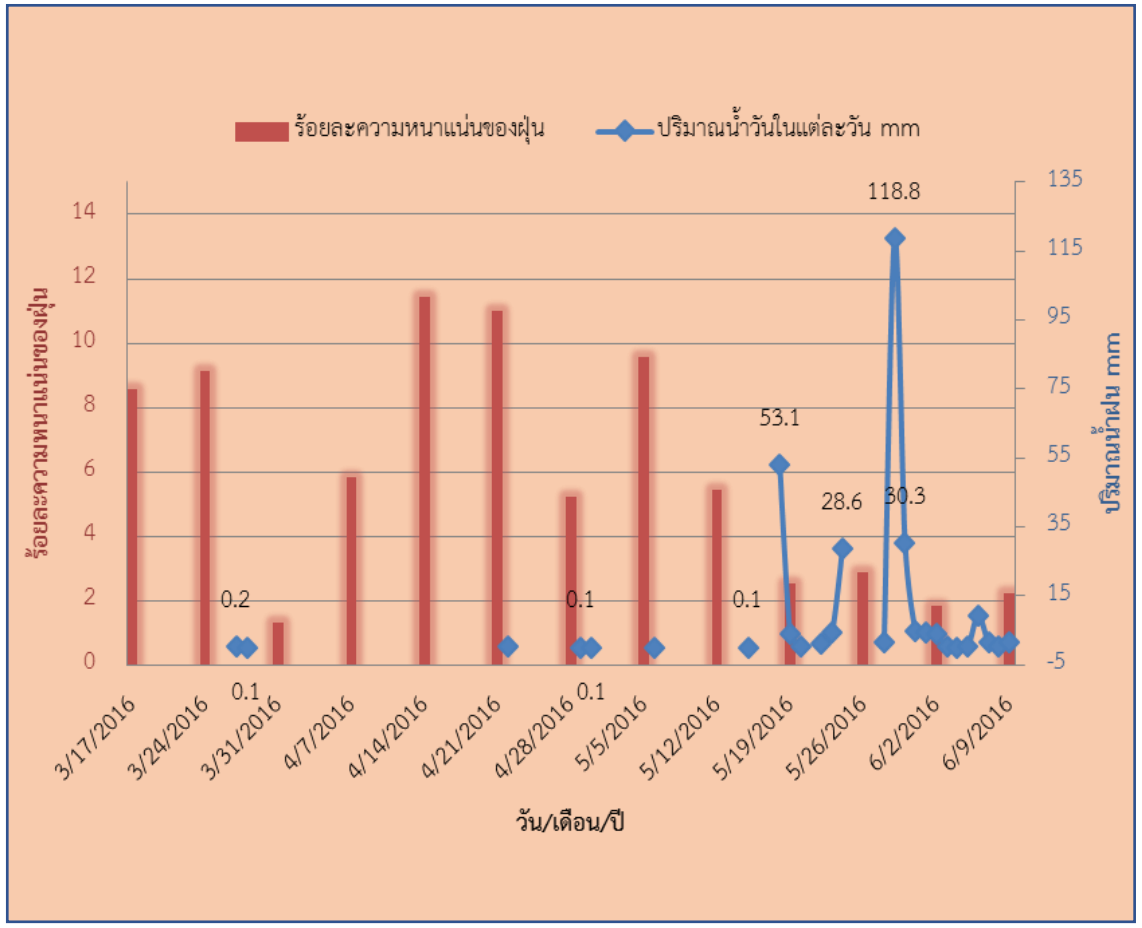
ผลการศึกษาวิจัยเบื้องต้น

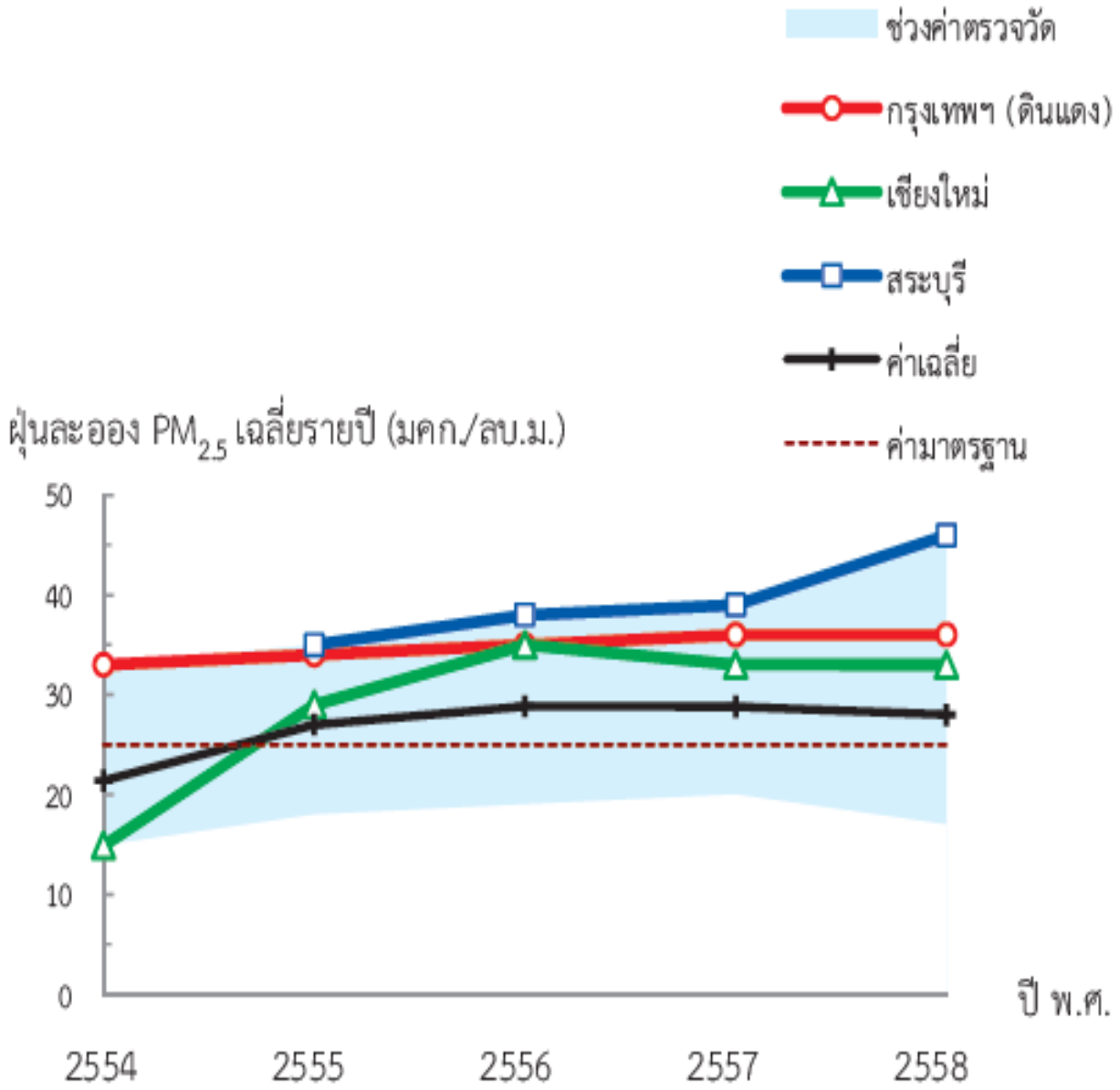


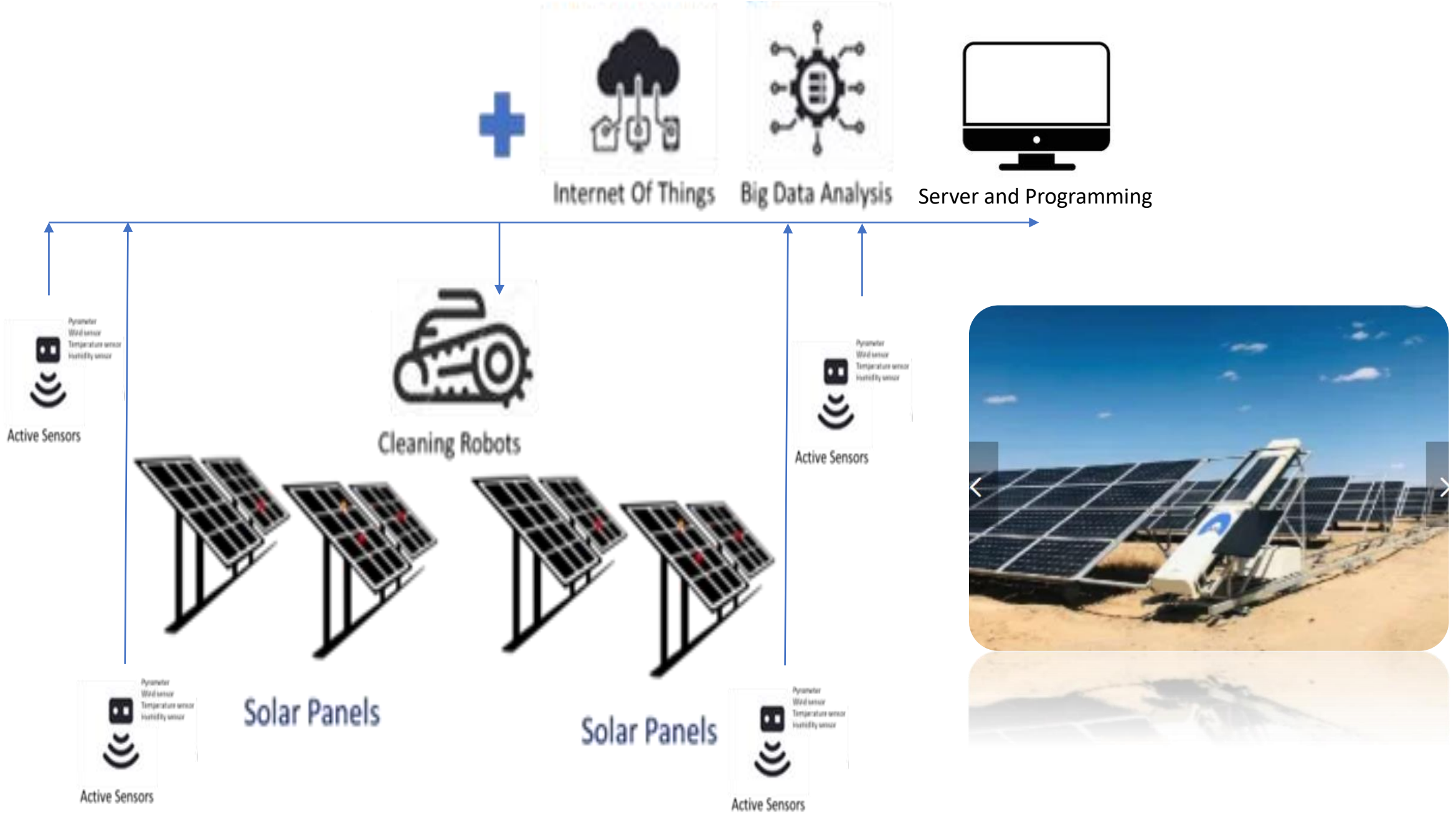
ค่าการส่องผ่านแสงของฝุ่นที่สะสมในแต่ละสัปดาห์



การติดยึดของฝุ่น

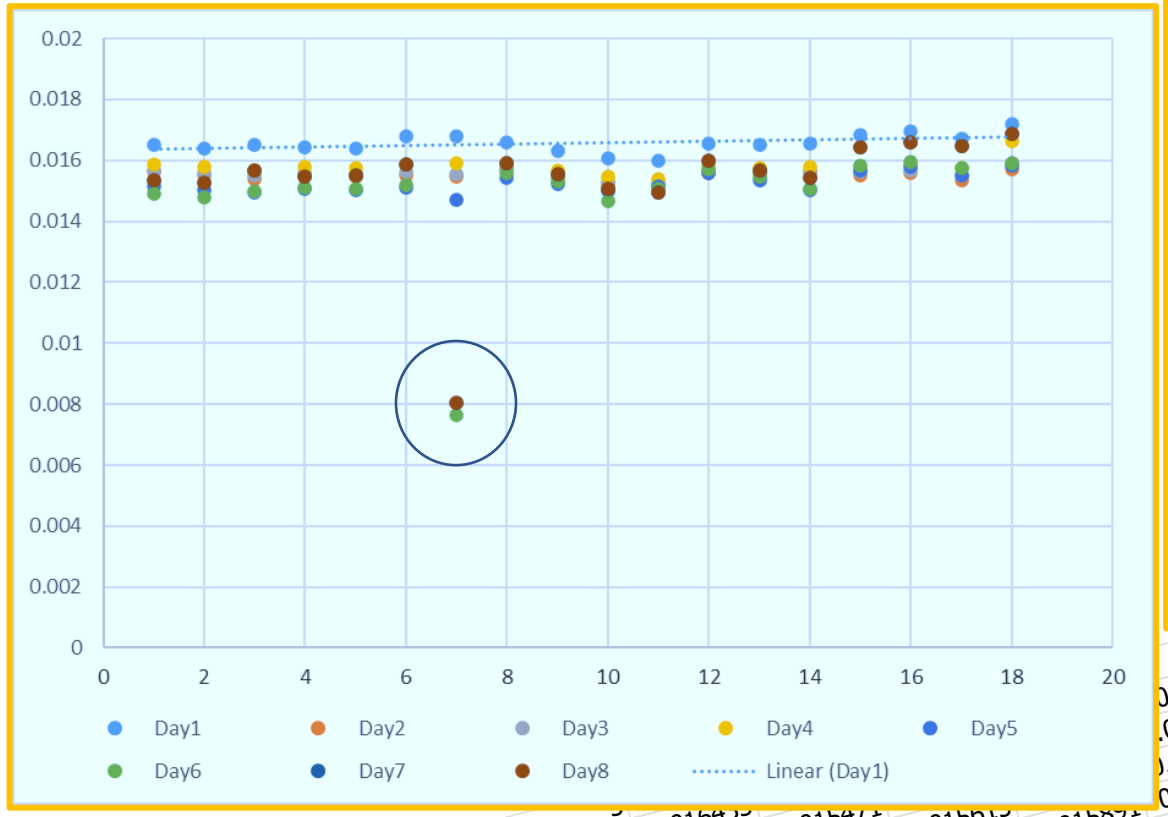


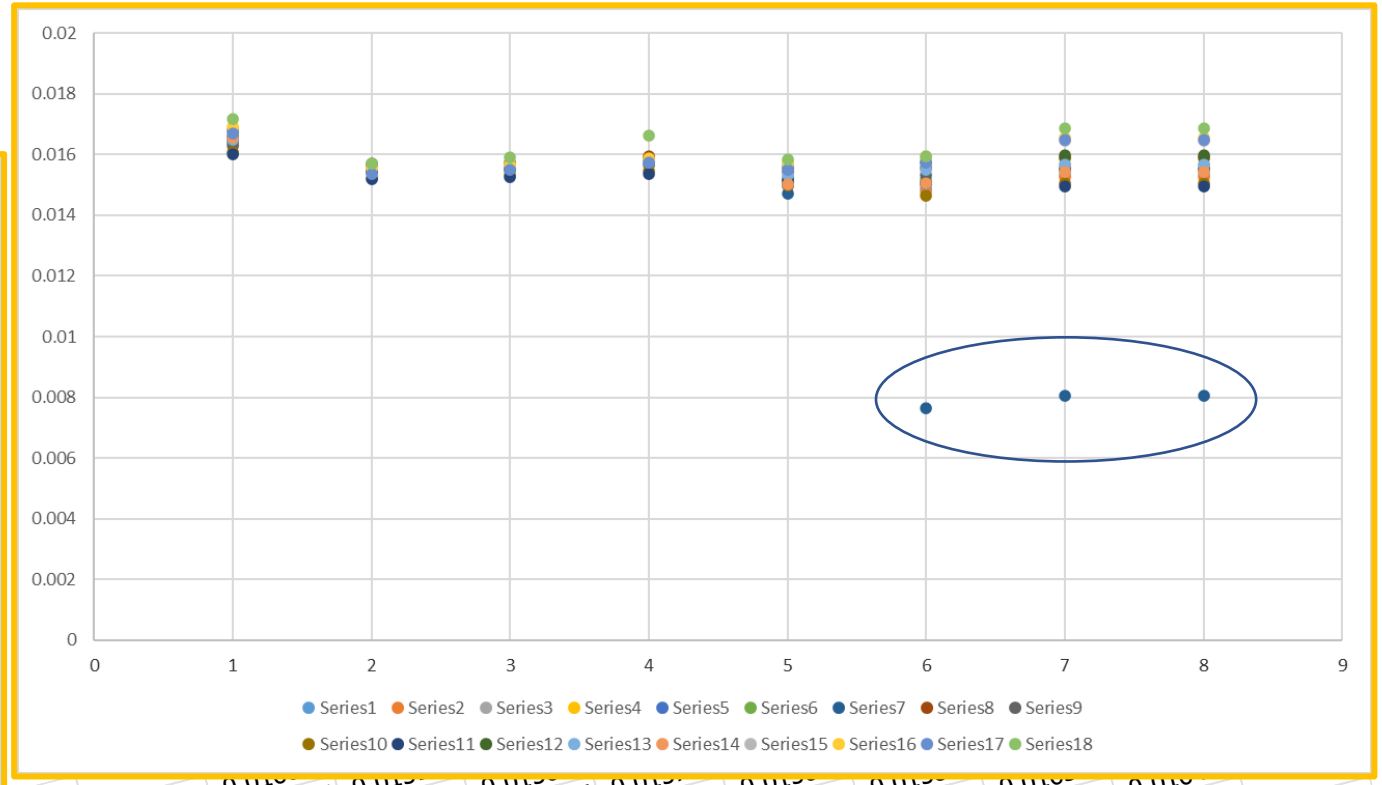


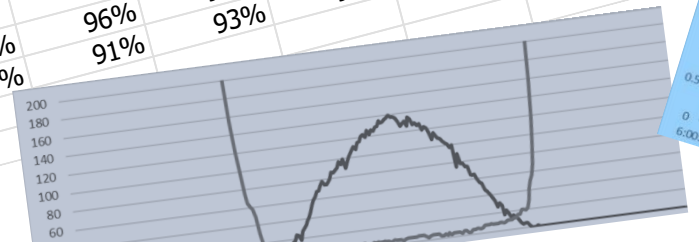

Internet Of Things
Big Data Analysis
Server and Programming
Active Sensors
Cleaning Robots
Active Sensors
Solar Panels
Solar Panels
Active Sensors
Active Sensors

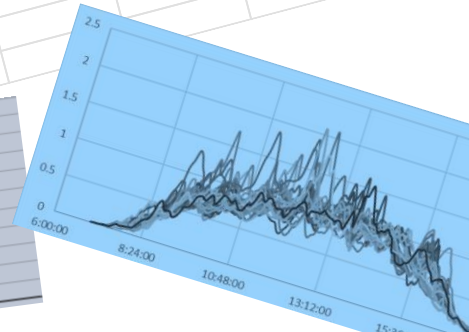
Efficiency by data analysis

By environmental sensors



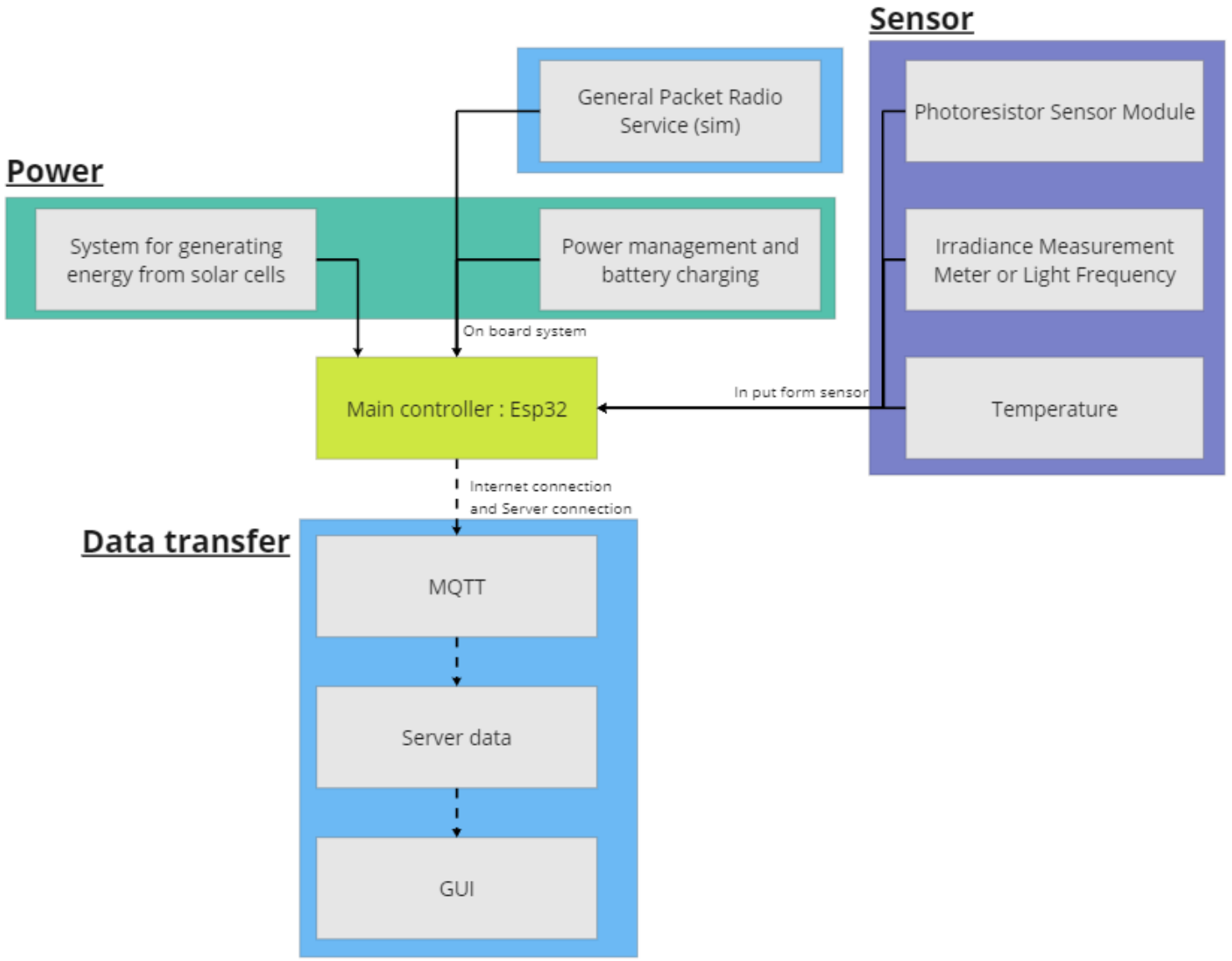


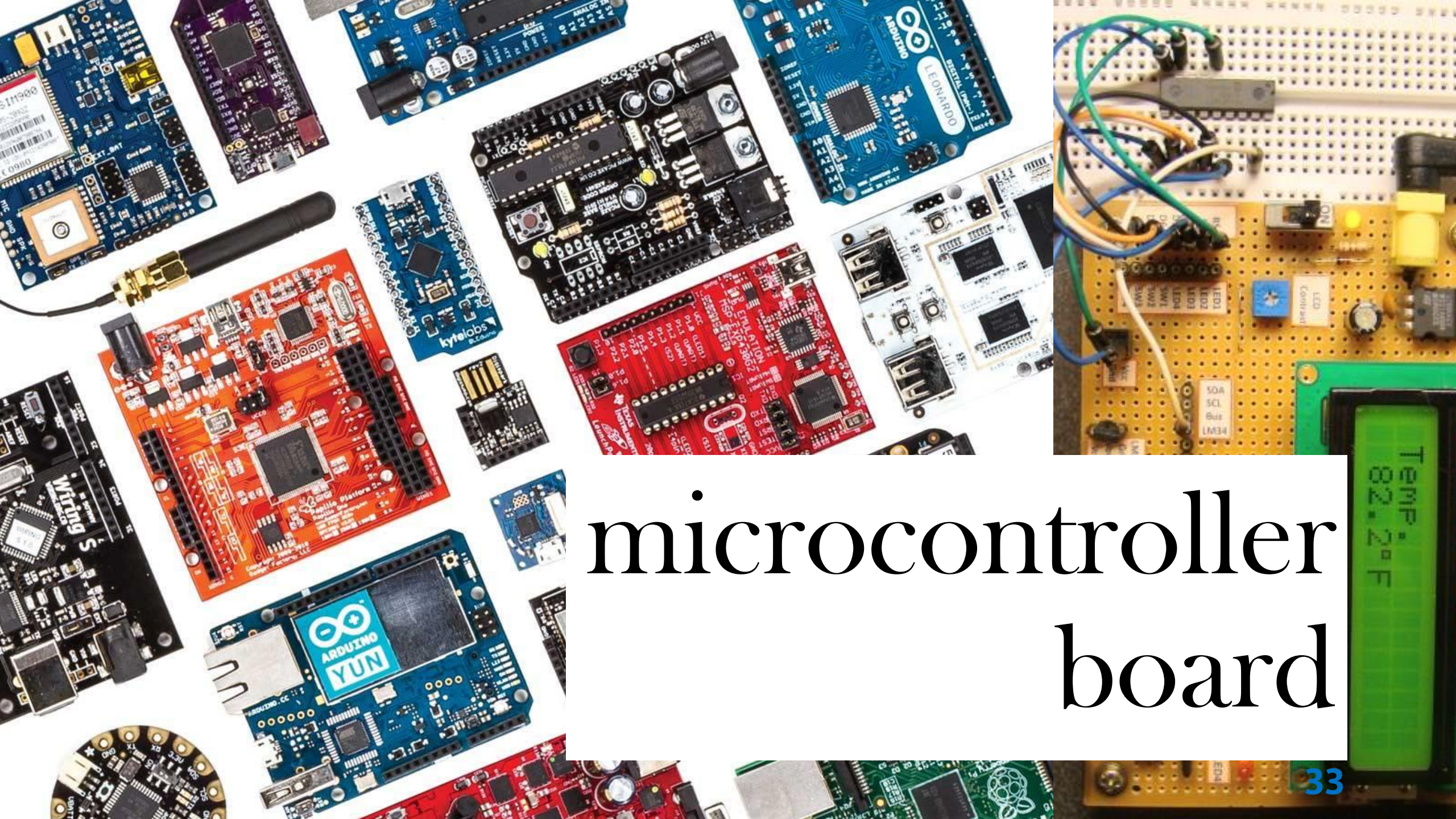




Internet of Things : Architecture Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm

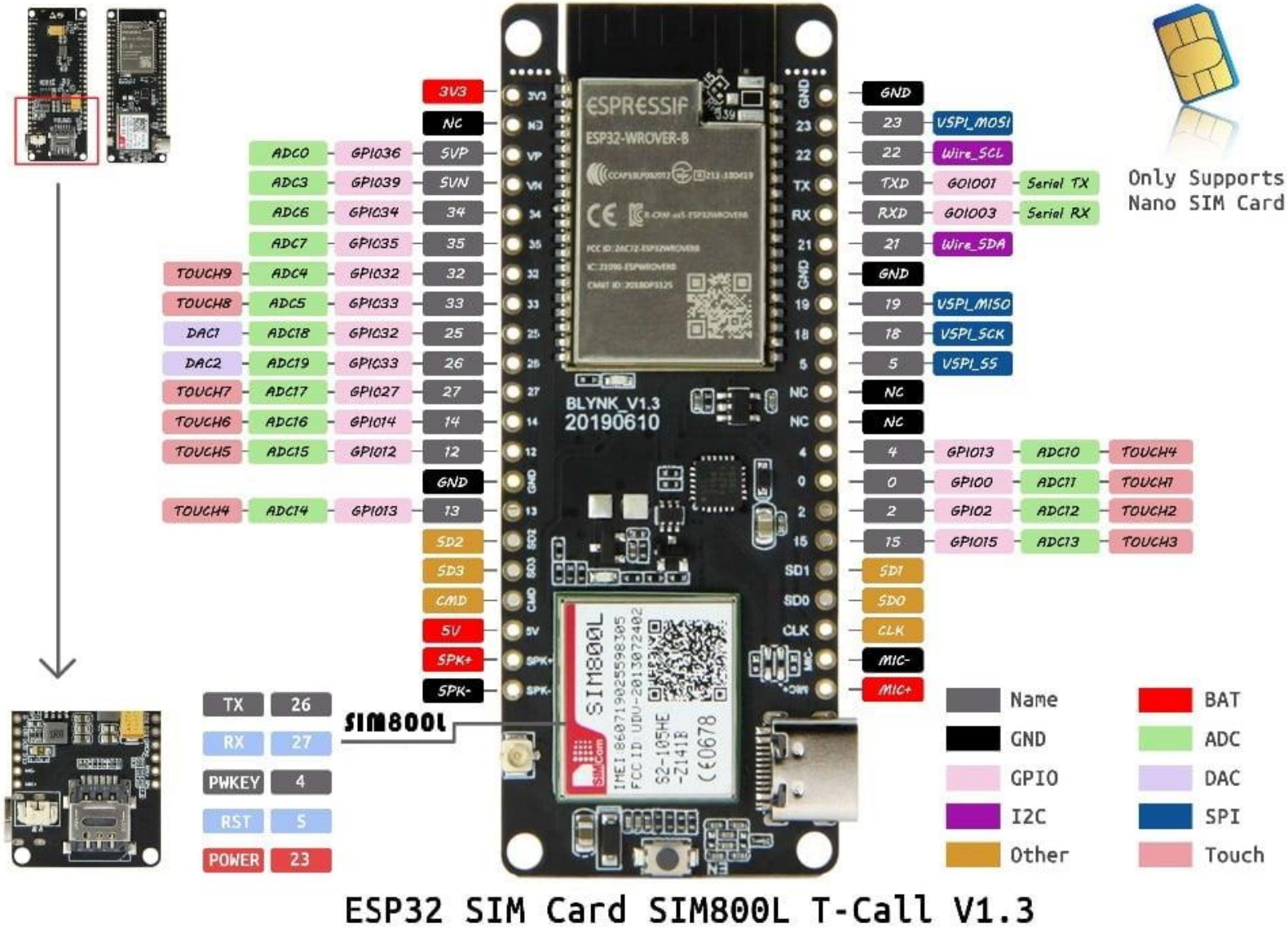




microcontroller board

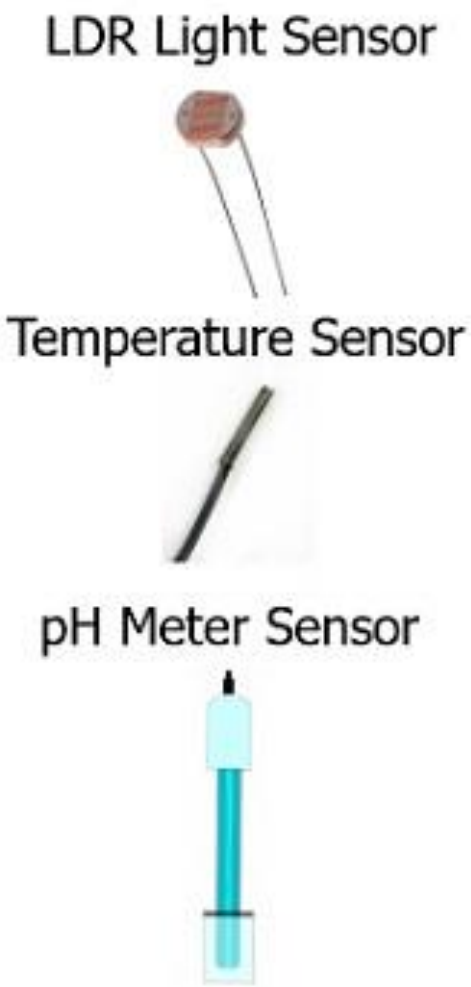
Microcontroller : Esp32 Pinout

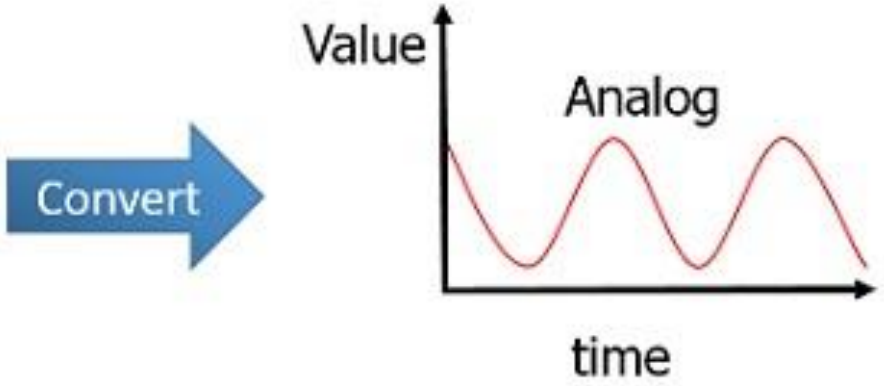
Environment Sensors and IOT for Solar Farm



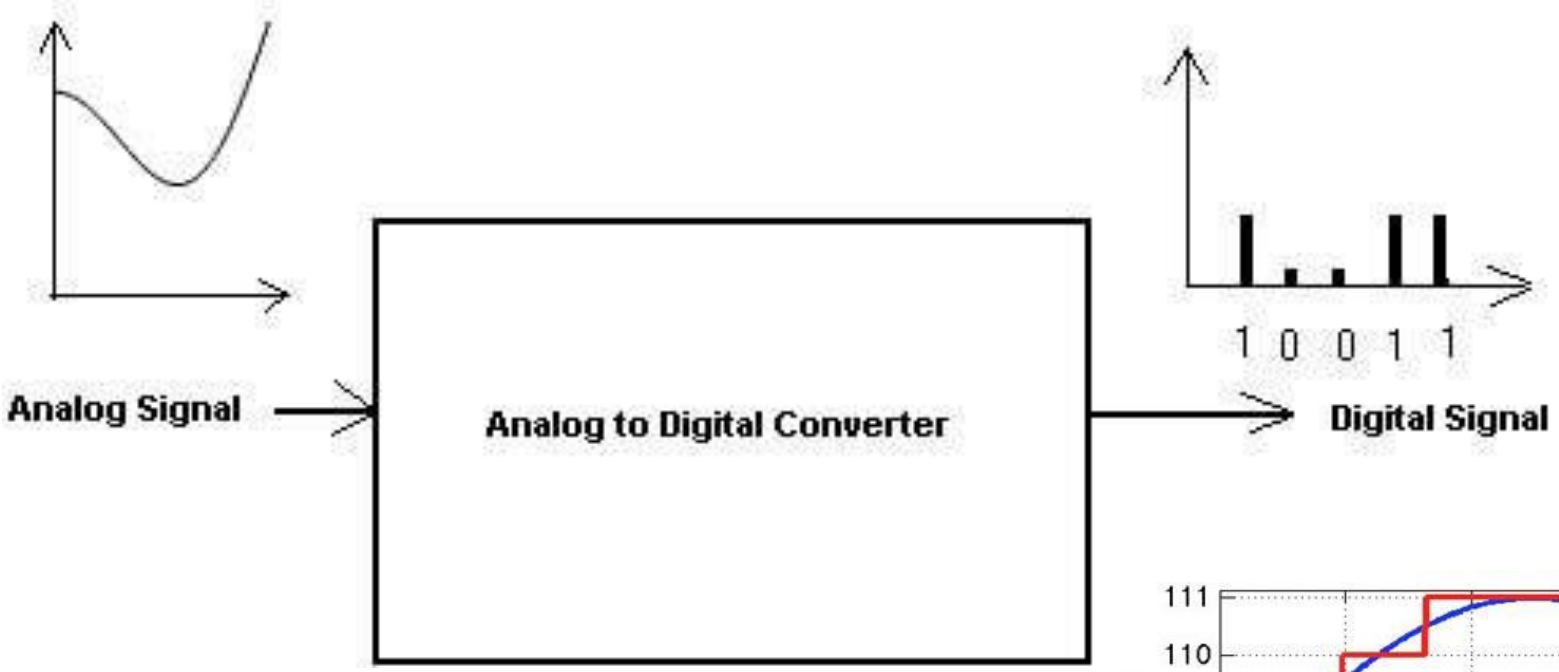
ESP32 SIM Card SIM800L T-Call V1.3

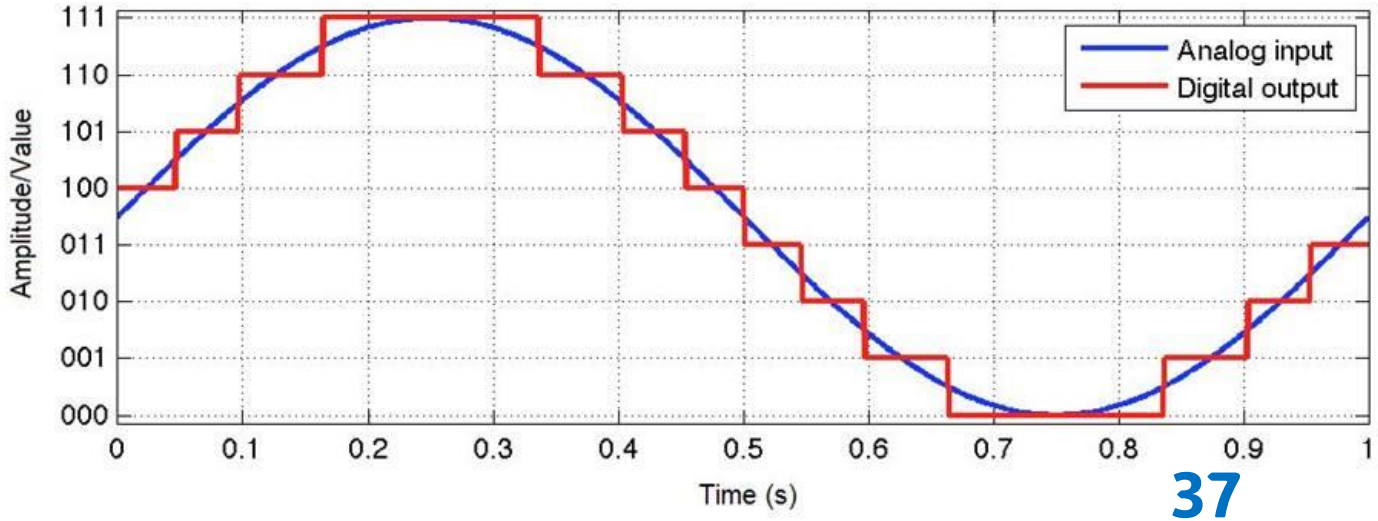
Microcontroller : Esp32 Analog Input





Microcontroller : Analog to Digital Converter (ADC)





Microcontroller : Digital I/O

PIR Motion Sensor

Button

ON-OFF Switch

Proximity Switch

INPUT

OUTPUT

Relay

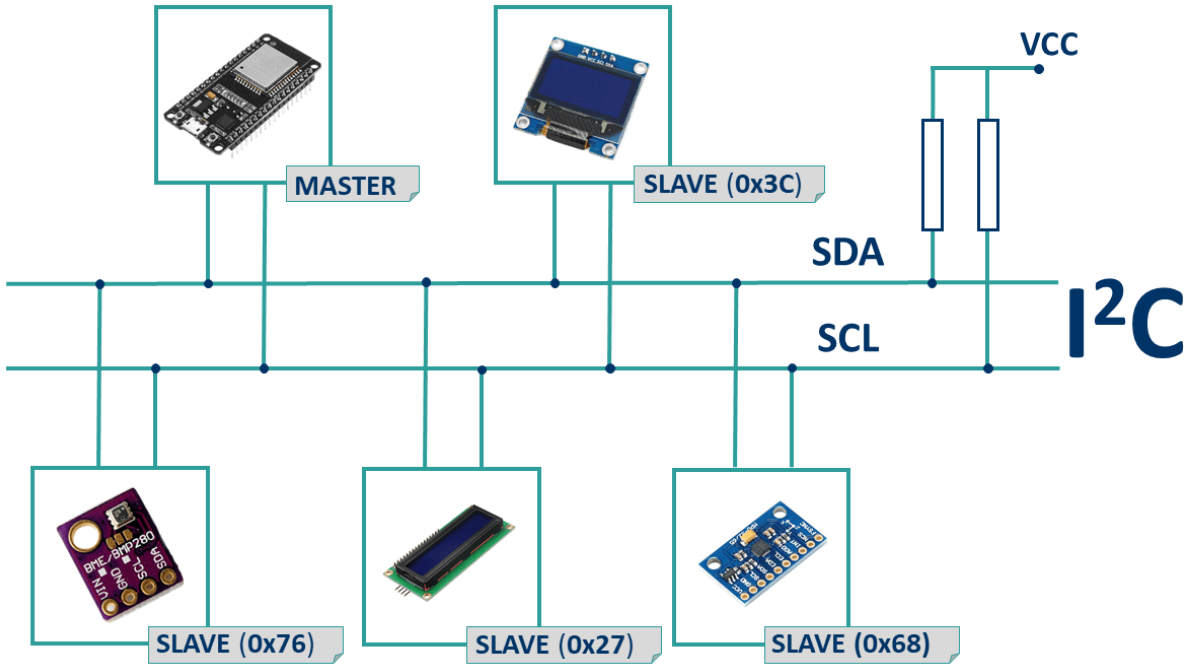
LED Bar Graph

Digital Input จากผู้ใช้งานหรือจากเซ็นเซอร์

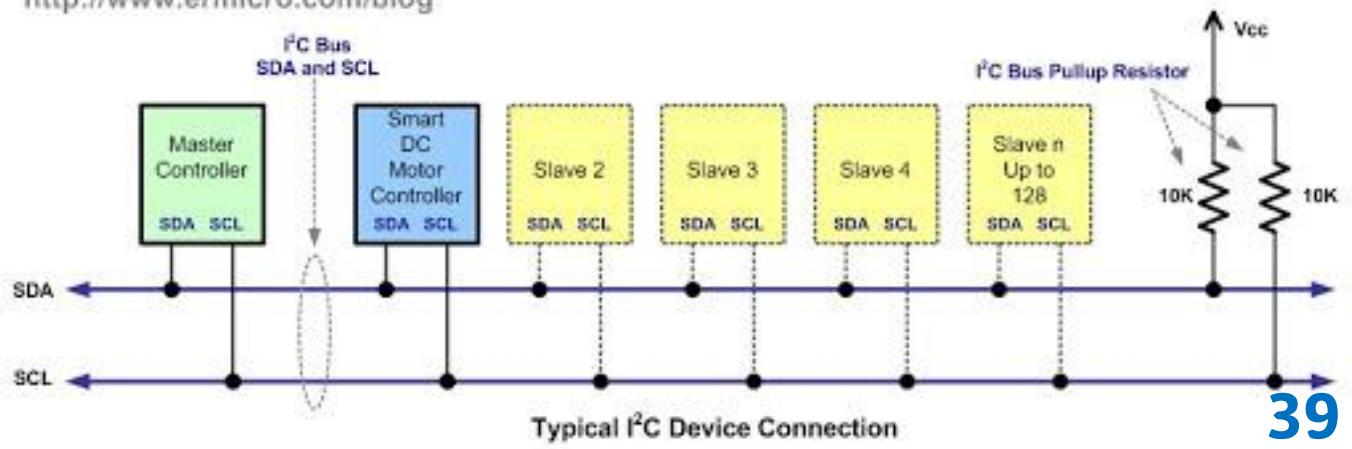
Digital Output จากไมโครคอนโทรเลอร์

Microcontroller : I2C I/O

Environment Sensors and IOT for Solar Farm



http://www.ermicro.com/blog



Typical I²C Device Connection

Microcontroller : SPI I/O

Send - Receive

ADC Module

Display Module

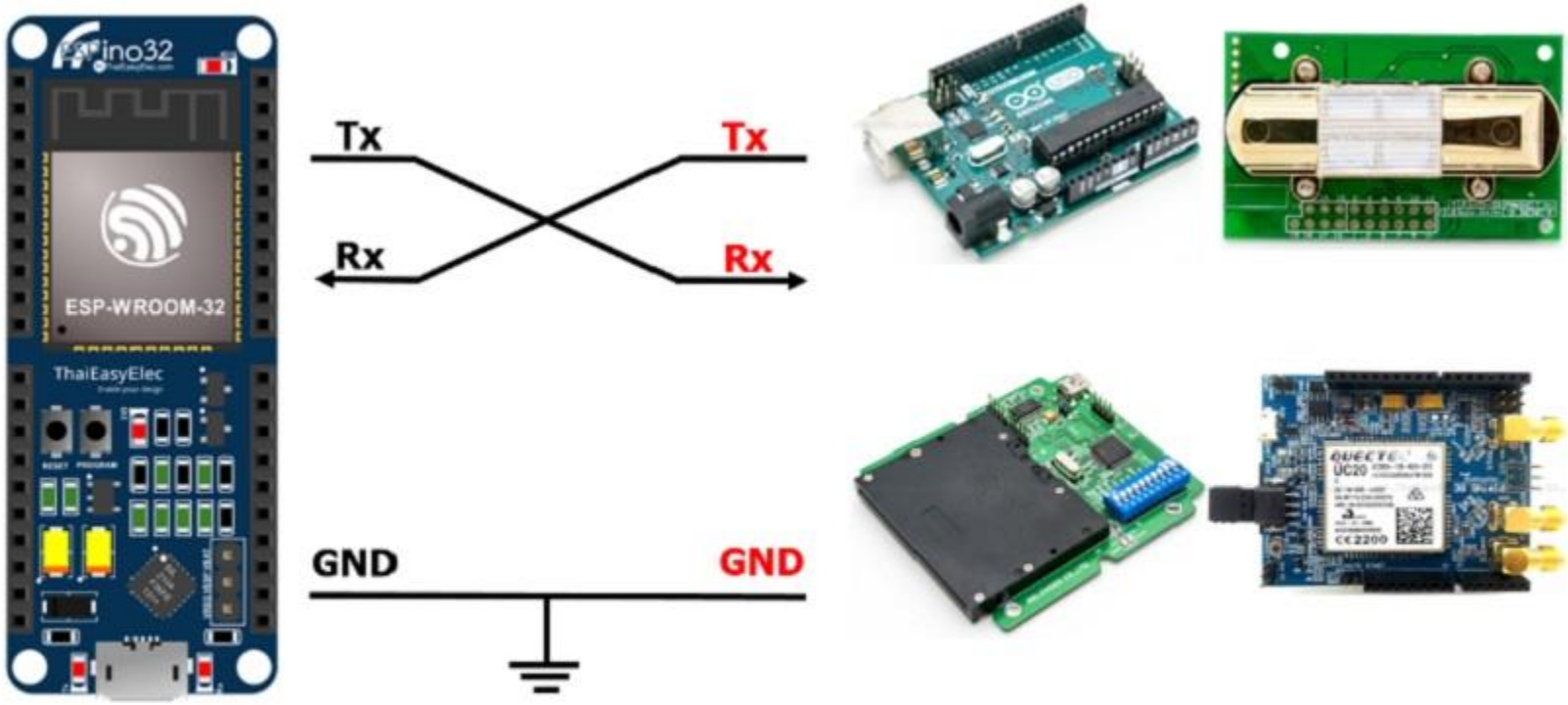
SD Card Module

RFID Module

ESPino32

SPI Device

Microcontroller : UART I/O

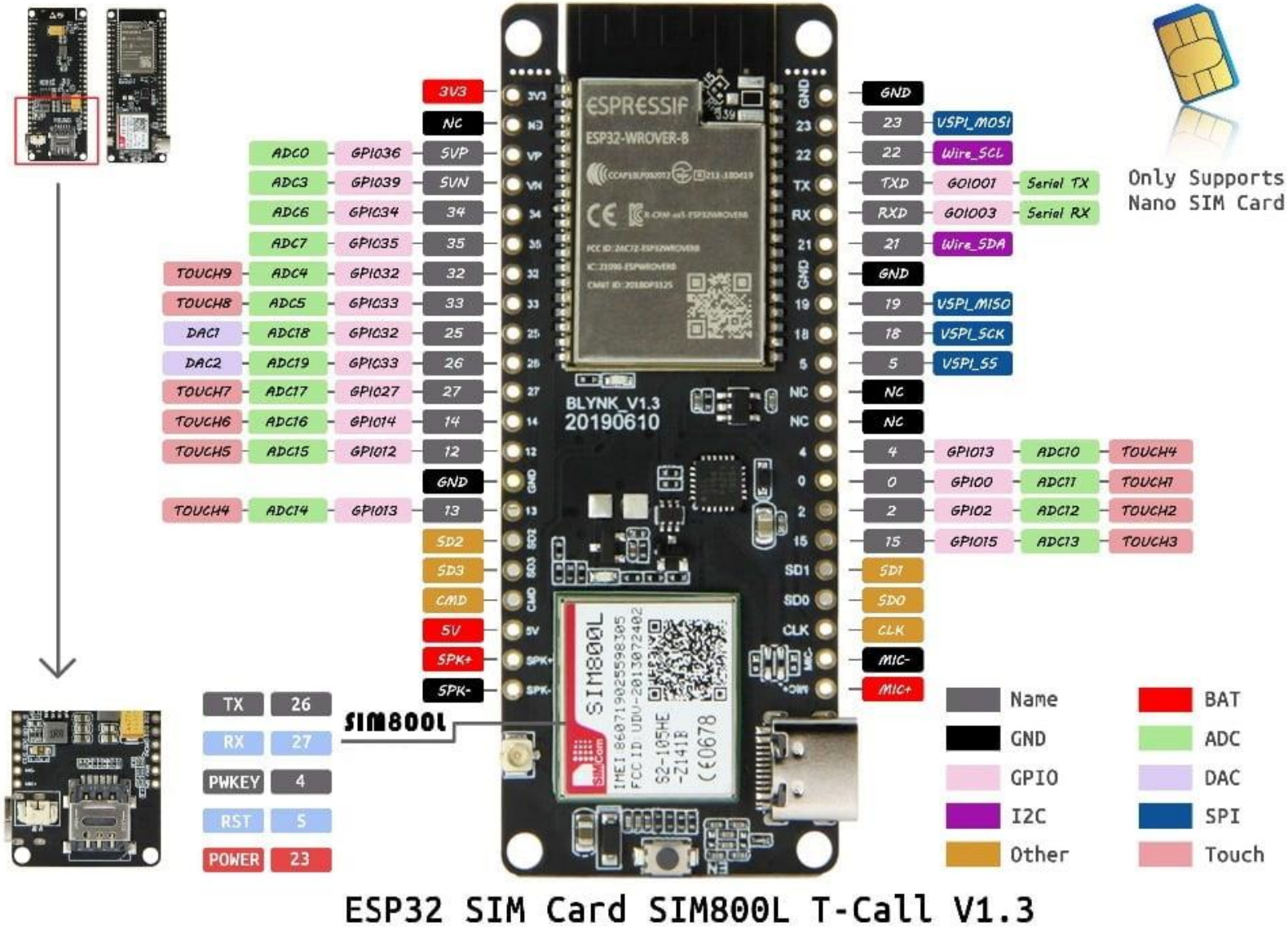


ESPino32

UART Interface

Microcontroller : Esp32 Pinout

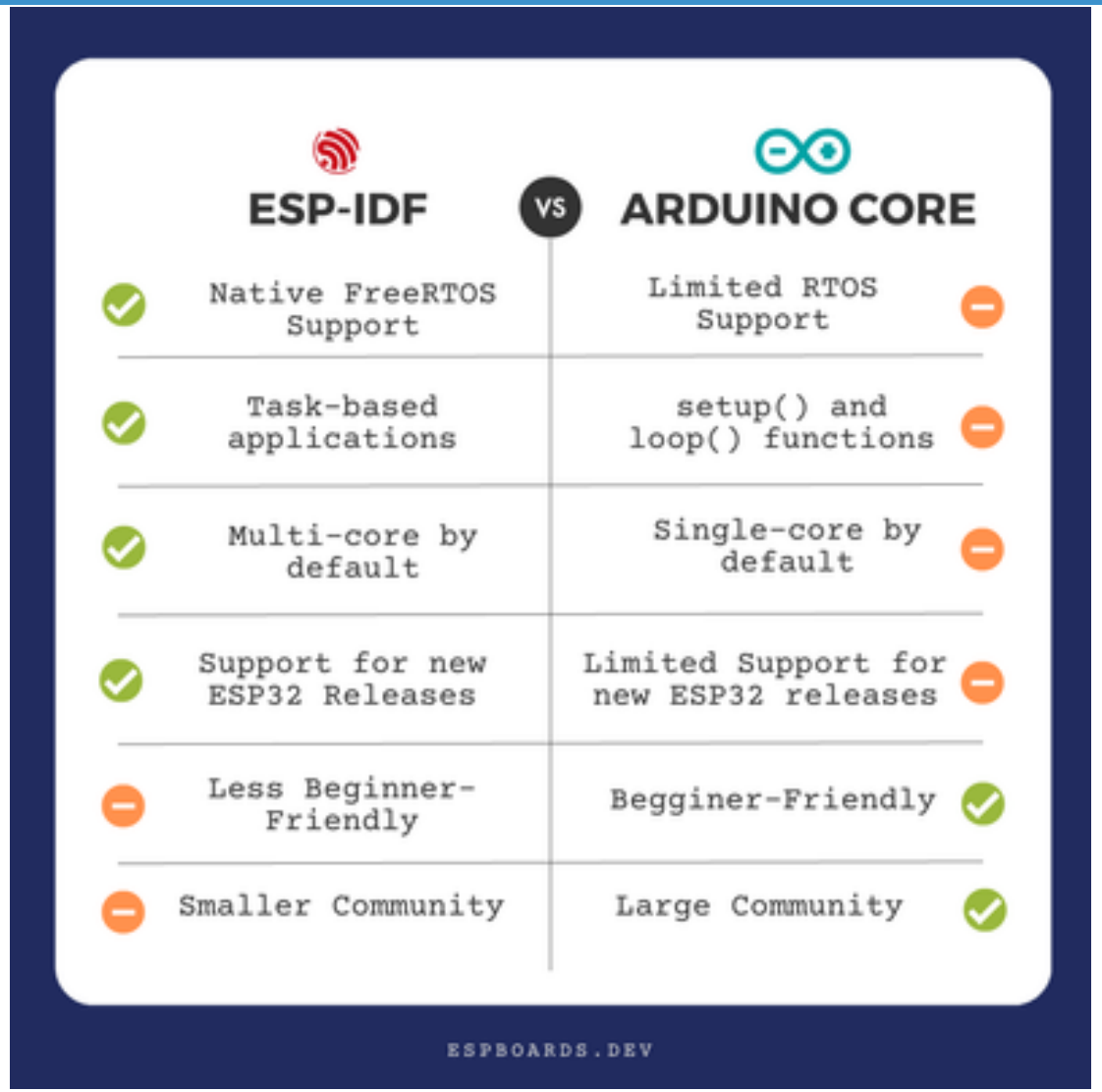
Environment Sensors and IOT for Solar Farm



Microcontroller : Esp32

Environment Sensors and IOT for Solar Farm

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

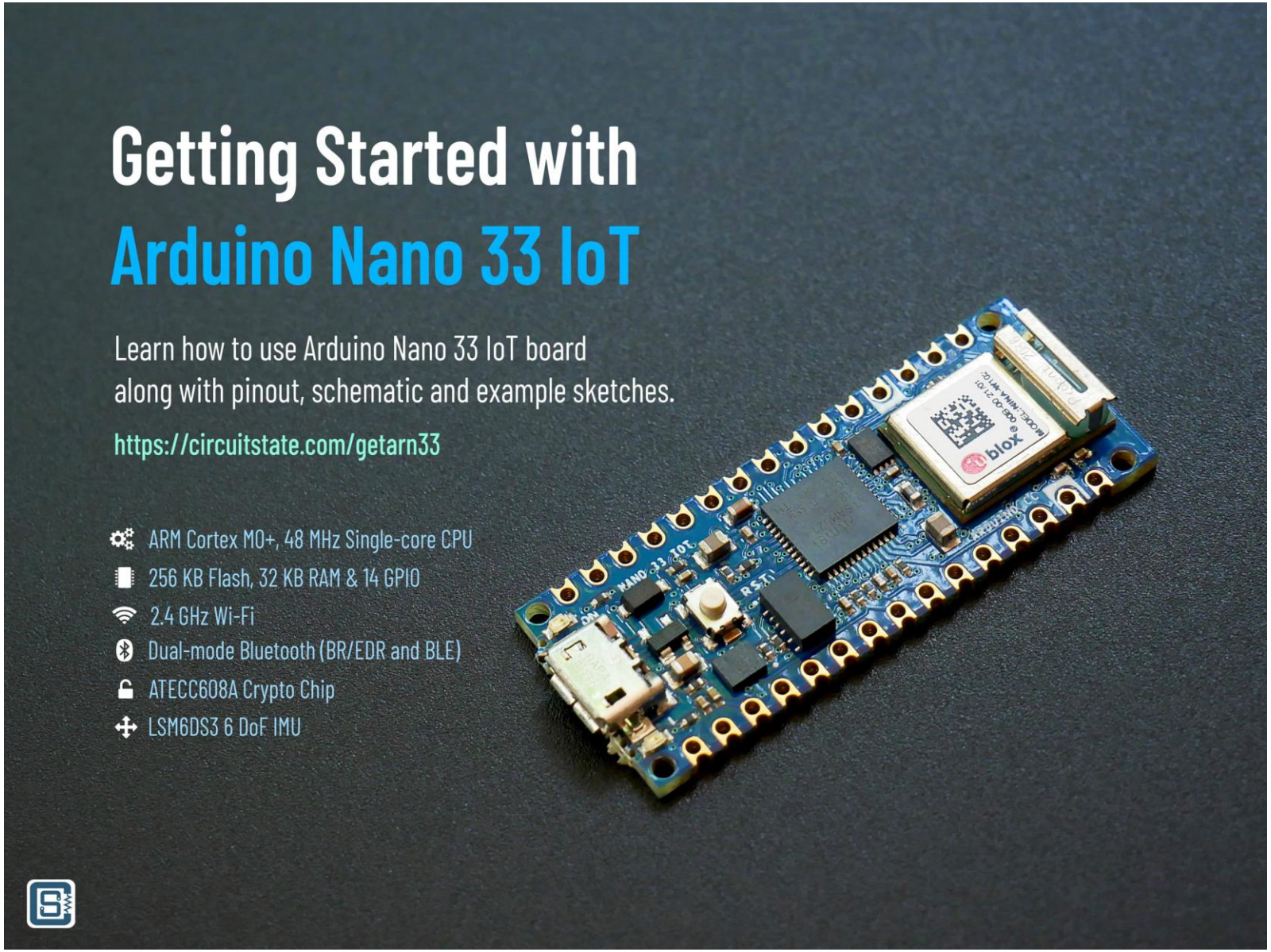


Microcontroller : Esp32 for IOT

Environment Sensors and IOT for Solar Farm

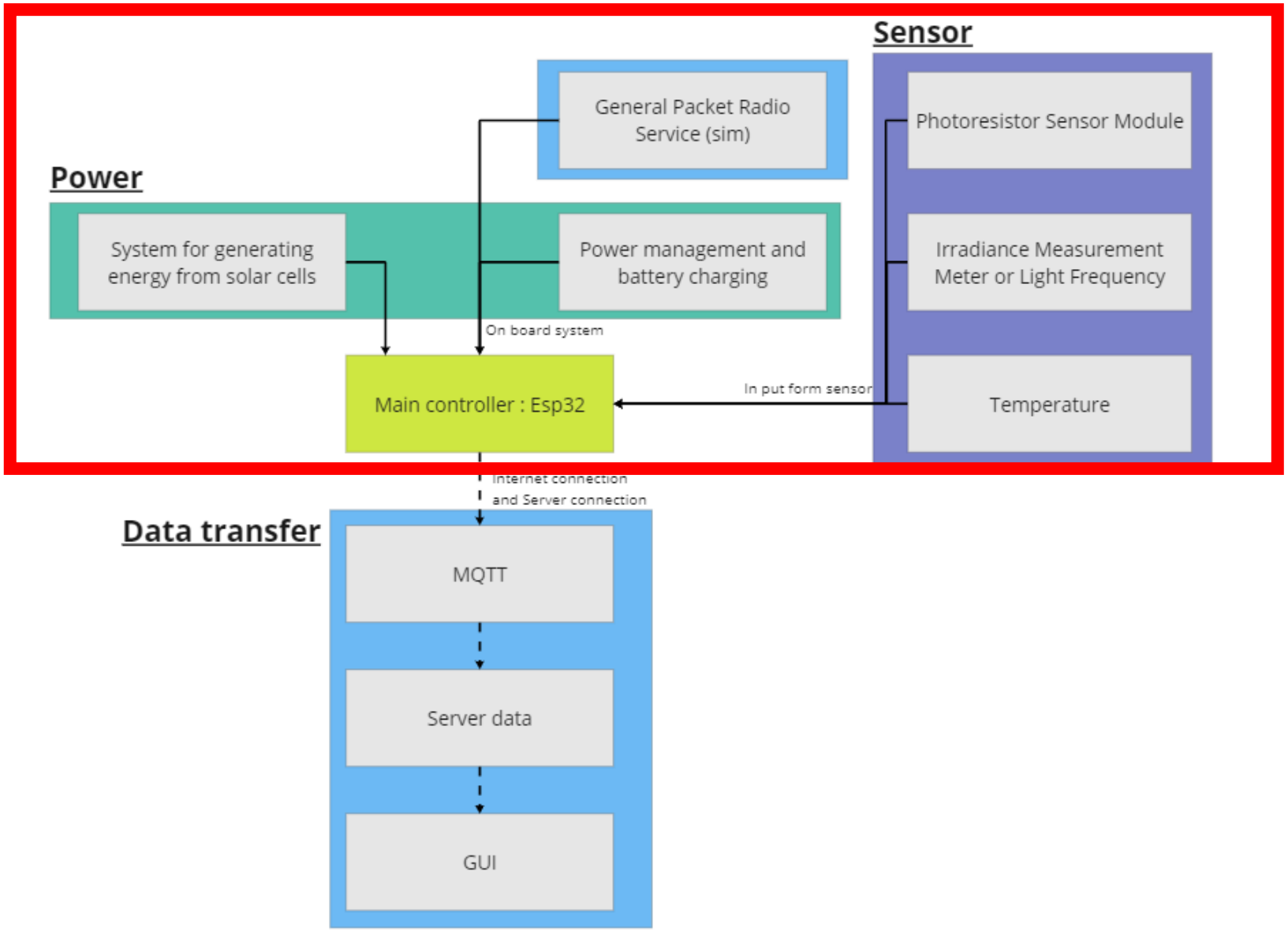
Microcontroller : Nano for IOT

Environment Sensors and IOT for Solar Farm



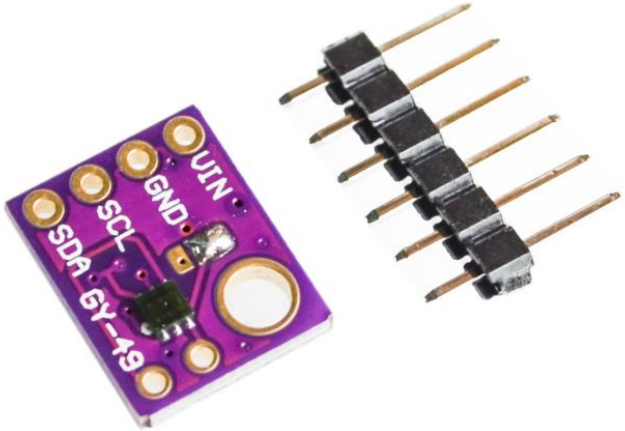
Internet of Things : Architecture Sensors for Solar Farm

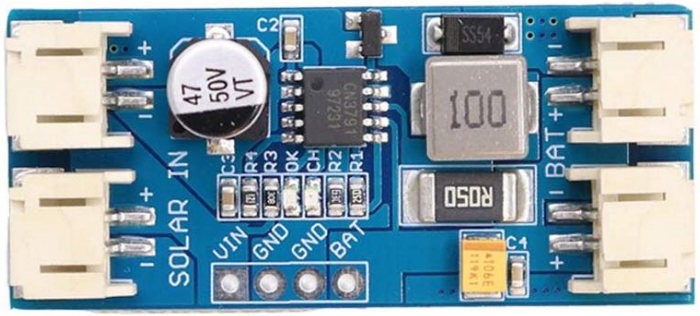
Environment Sensors and IOT for Solar Farm

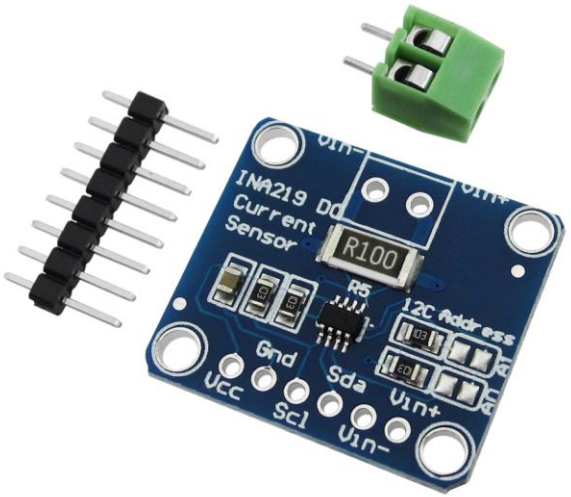


Sensor : Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm

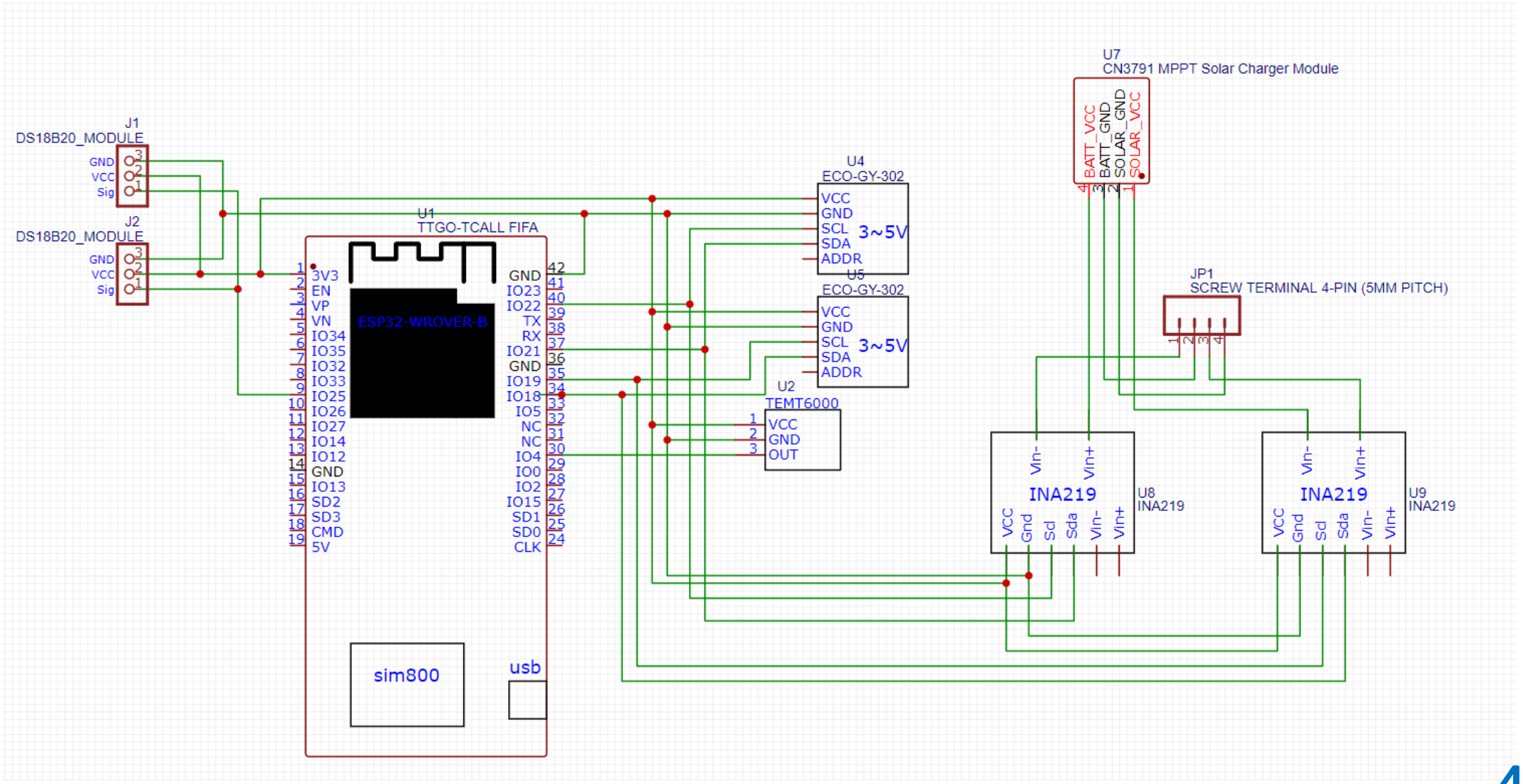






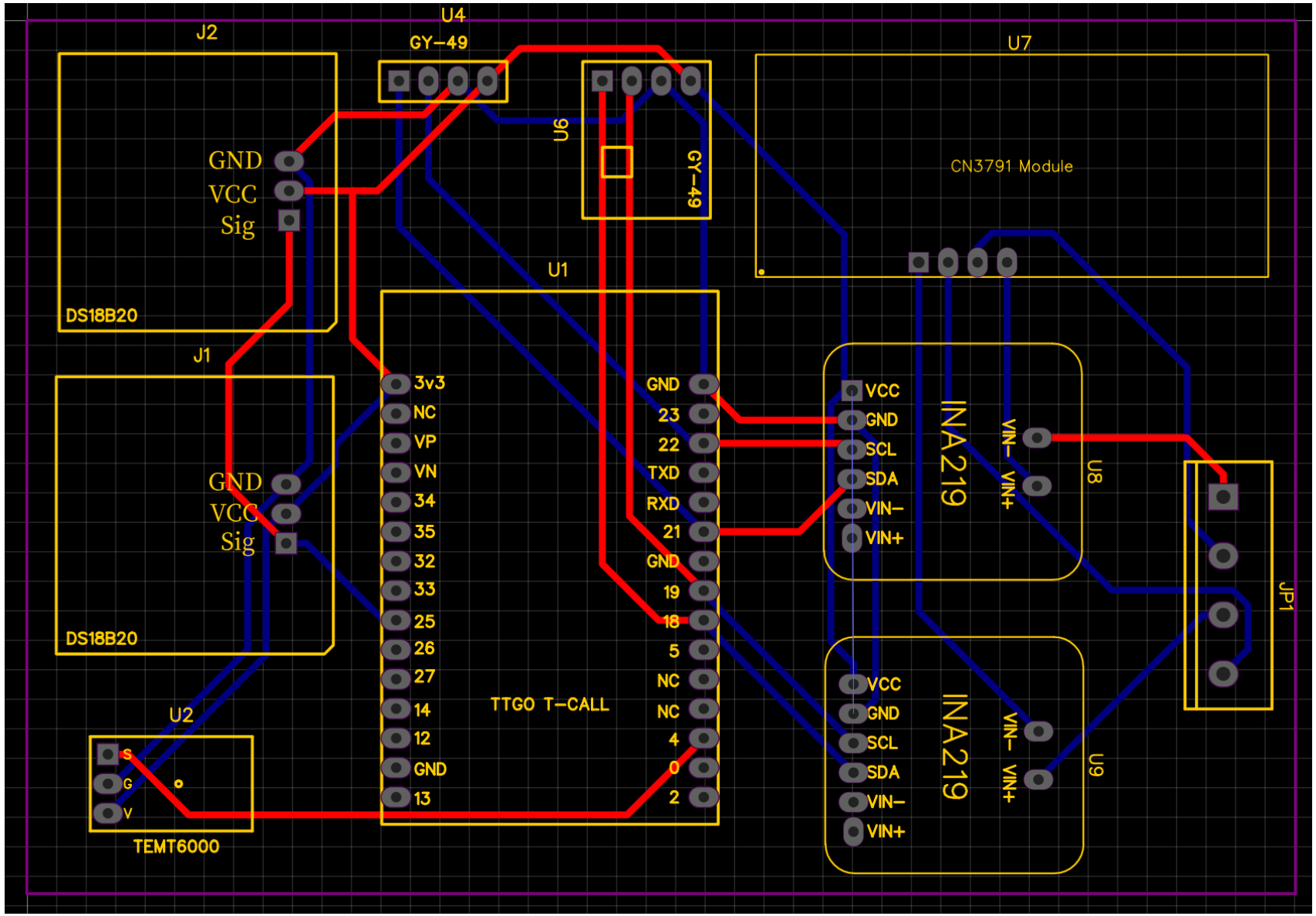
Sensor : Schematic Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm



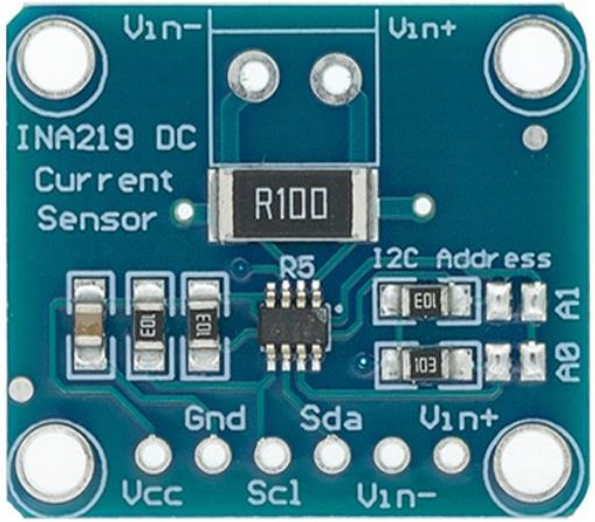
Sensor : PCB Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm



Sensor : INA219

Environment Sensors and IOT for Solar Farm

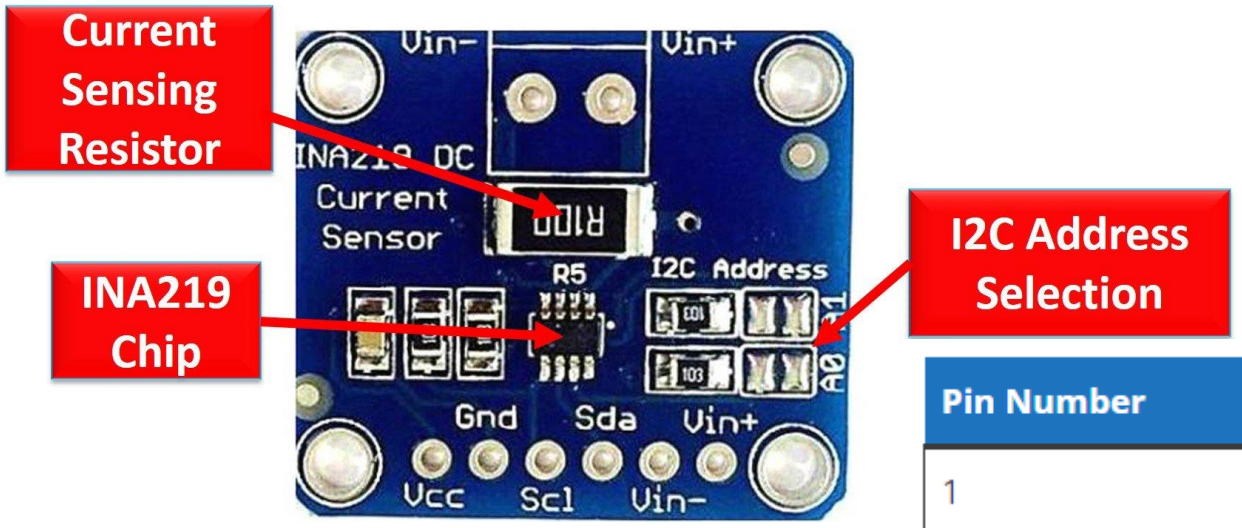


Features and Specifications

- Operational Voltage: 3 – 5.5 Volts
- Operating Temperature: -400C – 1250C
- Maximum Voltage: 6 Volts
- Bus Voltage Range: 0 – 26 Volts
- Current sensing Range: ±3.2A with ±0.8mA resolution
- 0.1 ohm 1% 2W current sense resistor

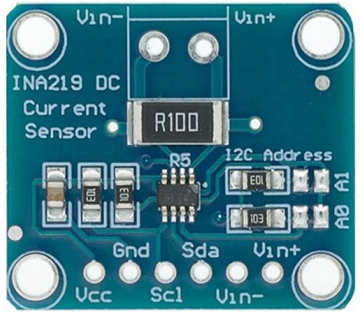
Sensor : INA219

Environment Sensors and IOT for Solar Farm



Pin Number	Pin Name	Function
1	A1	Address1 pin
2	A0	Address0 pin
3	SDA	Serial Data pin
4	SCL	Serial Clock pin
5	VS	Power Supply pin
6	GND	Ground pin
7	IN–	Positive Analog Input pin
8	IN+	Negative Analog Input pin

Sensor : INA219



ก่อนเริ่มใช้งาน ต้อง download library ของ INA219 2 ตัว คือ

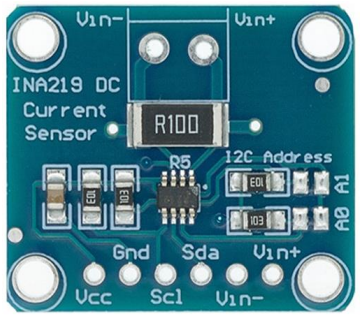
- Adafruit_INA219.h
- Adafruit_BusIO_Register.h

จะได้ไฟล์มาในรูปแบบนามสกุล .zip ให้กด Add .ZIP Library ใน Arduino IDE จึงจะพร้อมใช้งาน

หรือโหลดจาก lib ภายใน Arduino ide

Sensor : INA219

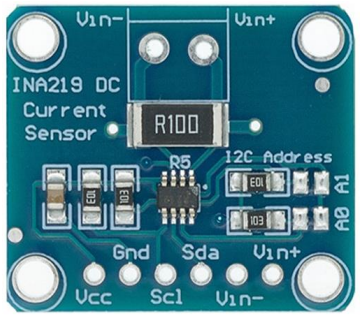
Environment Sensors and IOT for Solar Farm




```ino
#include "Wire.h"
#include "Adafruit_INA219.h"

Adafruit_INA219 ina219;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  if (! ina219.begin()) {
    Serial.println("Failed to find INA219 chip");
    while (1) { delay(10); }
  }

  Serial.print("BV"); Serial.print("\t"); // Bus Voltage
  Serial.print("SV"); Serial.print("\t"); // Shunt Voltage
  Serial.print("LV"); Serial.print("\t"); // Load Voltage
  Serial.print("C"); Serial.print("\t");  // Current
  Serial.println("P");  // Power
}
```

Sensor : INA219

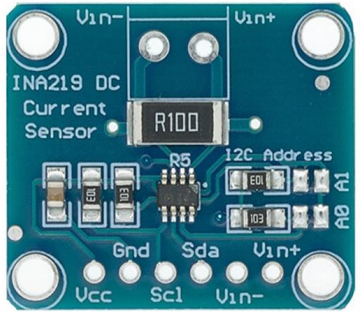
Environment Sensors and IOT for Solar Farm




```
void loop() {
  float shuntvoltage = 0;
  float busvoltage = 0;
  float current_mA = 0;
  float loadvoltage = 0;
  float power_mW = 0;

  shuntvoltage = ina219.getShuntVoltage_mV();
  busvoltage = ina219.getBusVoltage_V();
  current_mA = ina219.getCurrent_mA();
  power_mW = ina219.getPower_mW();
  loadvoltage = busvoltage + (shuntvoltage / 1000);

  Serial.print(busvoltage); Serial.print("\t");
  Serial.print(shuntvoltage); Serial.print("\t");
  Serial.print(loadvoltage); Serial.print("\t");
  Serial.print(current_mA); Serial.print("\t");
  Serial.println(power_mW);

  delay(1000);
}
```

Sensor : INA219

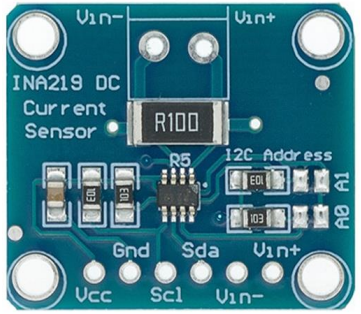
Environment Sensors and IOT for Solar Farm




```
1  #include "Wire.h"
2  #include "Adafruit_INA219.h"
3
4  Adafruit_INA219 ina219_a;
5  Adafruit_INA219 ina219_b;
6
7  void setup() {
8    Serial.begin(115200);
9    Wire.begin(18, 19);
10   Wire1.begin(21, 22);
11   ina219_a.begin(&Wire);
12   ina219_b.begin(&Wire1);
13
```

Sensor : INA219

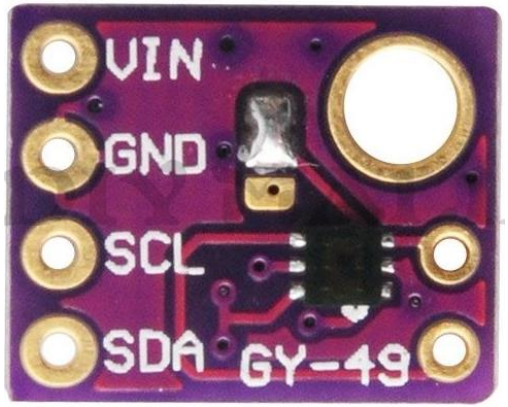
Environment Sensors and IOT for Solar Farm




```
shuntvoltage = ina219_b.getShuntVoltage_mV();
busvoltage = ina219_b.getBusVoltage_V();
current_mA = ina219_b.getCurrent_mA();
power_mW = ina219_b.getPower_mW();
loadvoltage = busvoltage + (shuntvoltage / 1000);

Serial.print(busvoltage);
Serial.print("\t");
Serial.print(shuntvoltage);
Serial.print("\t");
Serial.print(loadvoltage);
Serial.print("\t");
Serial.print(current_mA);
Serial.print("\t");
Serial.println(power_mW);
delay(1000);
```

Sensor : GY-49 MAX44009

Environment Sensors and IOT for Solar Farm



Introduction of Pins	
VIN	Anode of the power supply
GND	ground
SCL	I2C clock
SDA	I2C data

- MAX44009 ambient light sensor has I2C digital output
- has 22-bit ultra-wide dynamic range (0.045lm~188,000lm)

Features

- Wide Detection Range: 0.045lm~188,000lm
- Small size, 2mm x 2mm x 0.6mm UTDFN-Opto package
- VCC = 1.7V~3.6V
- Working Current ICC = 0.65μA
- Working Temperature: -40°C~+85°C
- Device Optional Address
- 1001 010x and 1001 011x

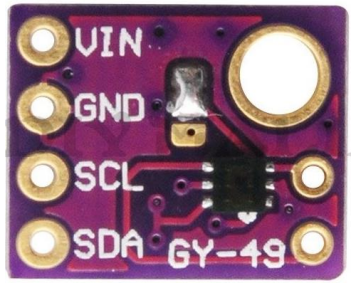
Sensor : GY-49 MAX44009

Environment Sensors and IOT for Solar Farm


```
#include "Wire.h"

#define Addr 0x4A

void setup() {

  // Initialise serial communication
  Serial.begin(115200);
  Wire.begin(21, 22);
  Wire.beginTransmission(Addr);
  Wire.write(0x02);
  Wire.write(0x40);
  Wire.endTransmission();

}
```

Sensor : GY-49 MAX44009

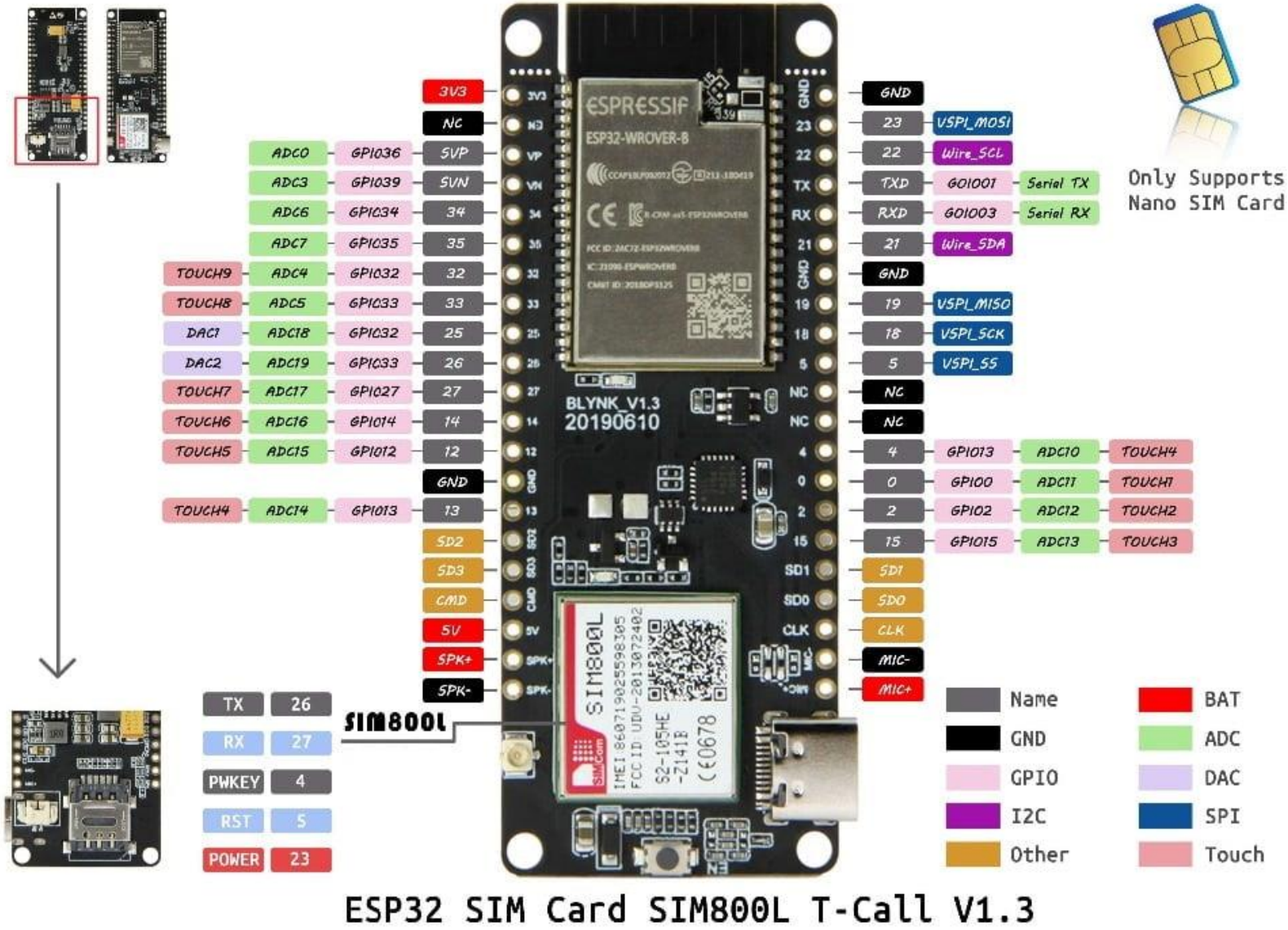
Environment Sensors and IOT for Solar Farm




```
void loop() {
  unsigned int dataone[2];
  Wire.beginTransmission(Addr);
  Wire.write(0x03);
  Wire.endTransmission();

  // Request 2 bytes of data
  Wire.requestFrom(Addr, 2);

  // Read 2 bytes of data luminance msb, luminance lsb
  if (Wire.available() == 2) {
    dataone[0] = Wire.read();
    dataone[1] = Wire.read();
  }

  // Convert the data to lux
  int exponent = (dataone[0] & 0xF0) >> 4;
  int mantissa = ((dataone[0] & 0x0F) << 4) | (dataone[1] & 0x0F);
  float luminance = pow(2, exponent) * mantissa * 0.045;

  Serial.print("Ambient Light luminance 1 :");
  Serial.print(luminance);
  Serial.println(" lux");

  delay(500);
}
```

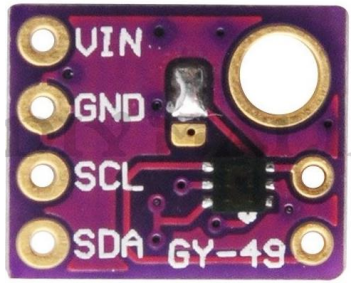
Microcontroller : Esp32 Pinout

Environment Sensors and IOT for Solar Farm



Sensor : GY-49 MAX44009

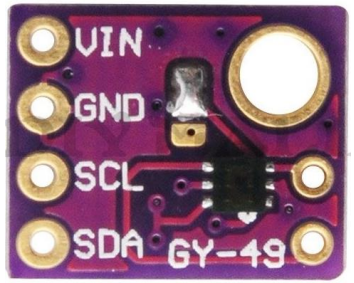
Environment Sensors and IOT for Solar Farm




```
#include "Wire.h"

#define Addr 0x4A

void setup() {

  // Initialise serial communication
  Serial.begin(115200);
  Wire.begin(18, 19);
  Wire.beginTransmission(Addr);
  Wire.write(0x02);
  Wire.write(0x40);
  Wire.endTransmission();

  Wire1.begin(21, 22);
  Wire1.beginTransmission(Addr);
  Wire1.write(0x02);
  Wire1.write(0x40);
  Wire1.endTransmission();
  delay(300);
}
```

Sensor : GY-49 MAX44009

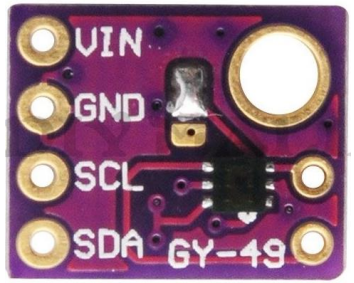
Environment Sensors and IOT for Solar Farm




```
23  void loop() {
24    unsigned int dataone[2];
25    Wire.beginTransmission(Addr);
26    Wire.write(0x03);
27    Wire.endTransmission();
28
29    // Request 2 bytes of data
30    Wire.requestFrom(Addr, 2);
31
32    // Read 2 bytes of data luminance msb, luminance lsb
33    if (Wire.available() == 2) {
34      dataone[0] = Wire.read();
35      dataone[1] = Wire.read();
36    }
37
38    // Convert the data to lux
39    int exponent = (dataone[0] & 0xF0) >> 4;
40    int mantissa = ((dataone[0] & 0x0F) << 4) | (dataone[1] & 0x0F);
41    float luminance = pow(2, exponent) * mantissa * 0.045;
42
43    Serial.print("Ambient Light luminance 1 :");
44    Serial.print(luminance);
45    Serial.println(" lux");
```

Sensor : GY-49 MAX44009

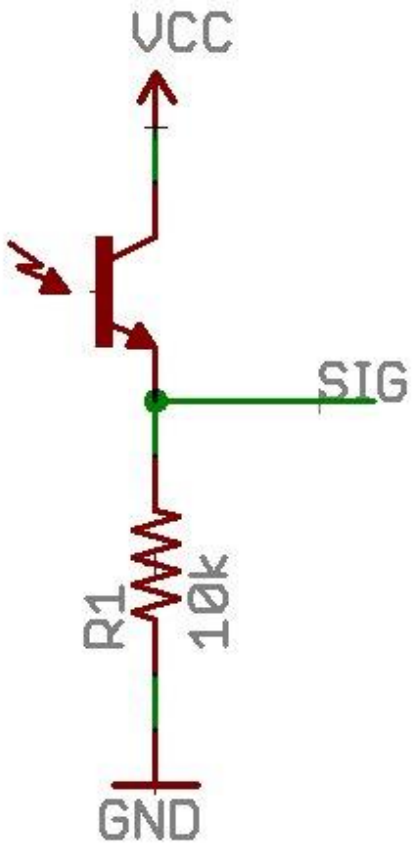
Environment Sensors and IOT for Solar Farm

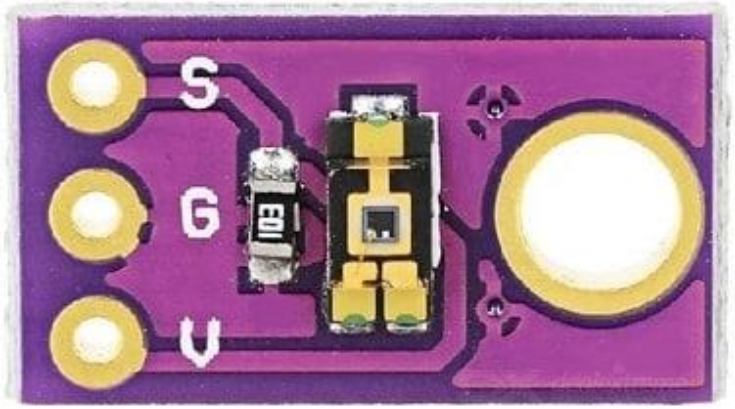



```
unsigned int datatwo[2];
Wire1.beginTransmission(Addr);
Wire1.write(0x03);
Wire1.endTransmission();

// Request 2 bytes of data
Wire1.requestFrom(Addr, 2);

// Read 2 bytes of data luminance msb, luminance lsb
if (Wire1.available() == 2) {
  datatwo[0] = Wire1.read();
  datatwo[1] = Wire1.read();
}

// Convert the data to lux
int exponenttwo = (datatwo[0] & 0xF0) >> 4;
int mantissatwo = ((datatwo[0] & 0x0F) << 4) | (datatwo[1] & 0x0F);
float luminancetwo = pow(2, exponenttwo) * mantissatwo * 0.045;

Serial.print("Ambient Light luminance 2 :");
Serial.print(luminancetwo);
Serial.println(" lux");
delay(500);
```

Sensor : TEMT6000

Environment Sensors and IOT for Solar Farm

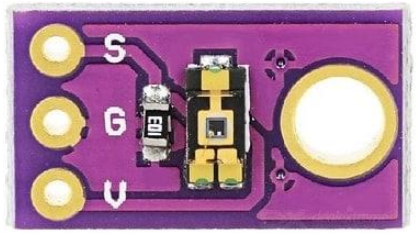




TEMT6000 Pin	Arduino Pin
SIG	A0
GND	GND
VCC	5V

Sensor : TEMT6000

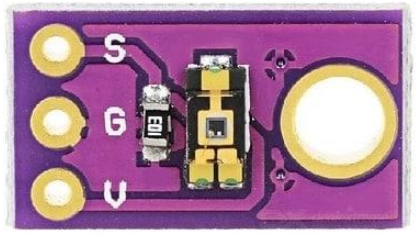
Environment Sensors and IOT for Solar Farm



Examples	
Illuminance	**Surfaces illuminated by:**
0.0001 lux	Moonless, overcast night sky (starlight)[3]
0.002 lux	Moonless clear night sky with airglow[3]
0.27–1.0 lux	Full moon on a clear night[3][4]
3.4 lux	Dark limit of civil twilight under a clear sky[5]
50 lux	Family living room lights (Australia, 1998)[6]
80 lux	Office building hallway/toilet lighting[7][8]
100 lux	Very dark overcast day[3]
320–500 lux	Office lighting[6][9][10][11]
400 lux	Sunrise or sunset on a clear day.
1000 lux	Overcast day;[3] typical TV studio lighting
10 000–25 000 lux	Full daylight (not direct sun)[3]
32 000–100 000 lux	Direct sunlight

Sensor : TEMT6000

Environment Sensors and IOT for Solar Farm



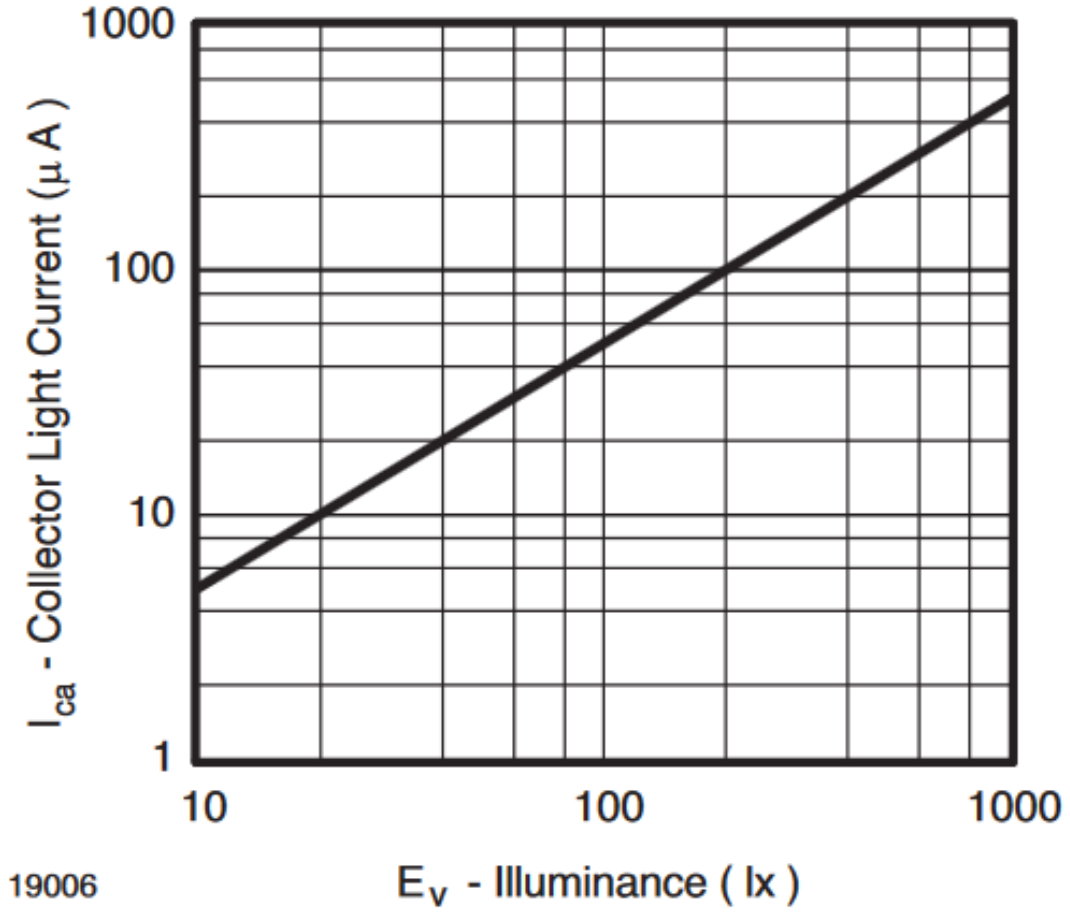
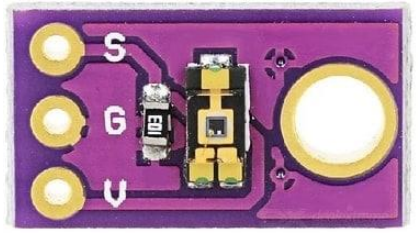


Figure 1. Collector Light Current vs. Illuminance

Found in the TEMT6000 datasheet.

Sensor : TEMT6000

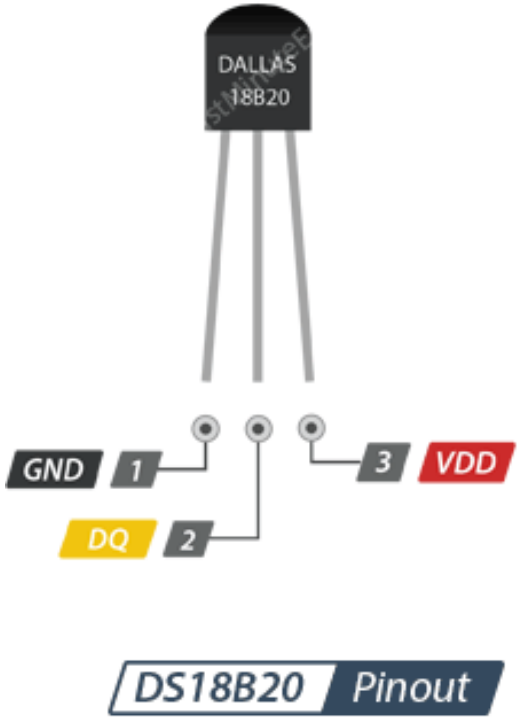
Environment Sensors and IOT for Solar Farm

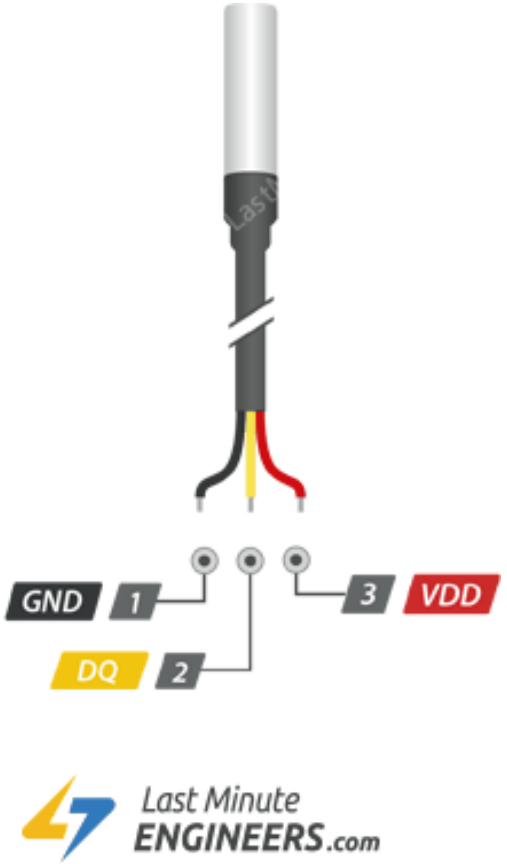



```
1   int sensor = 26;
2
3   void setup() {
4   Serial.begin(115200);
5   }
6
7   void loop() {
8   float lux = analogRead(sensor) * 0.64453125;
9   Serial.println(lux);
10
11  delay(100);
12  }
```

Sensor : DS18B20

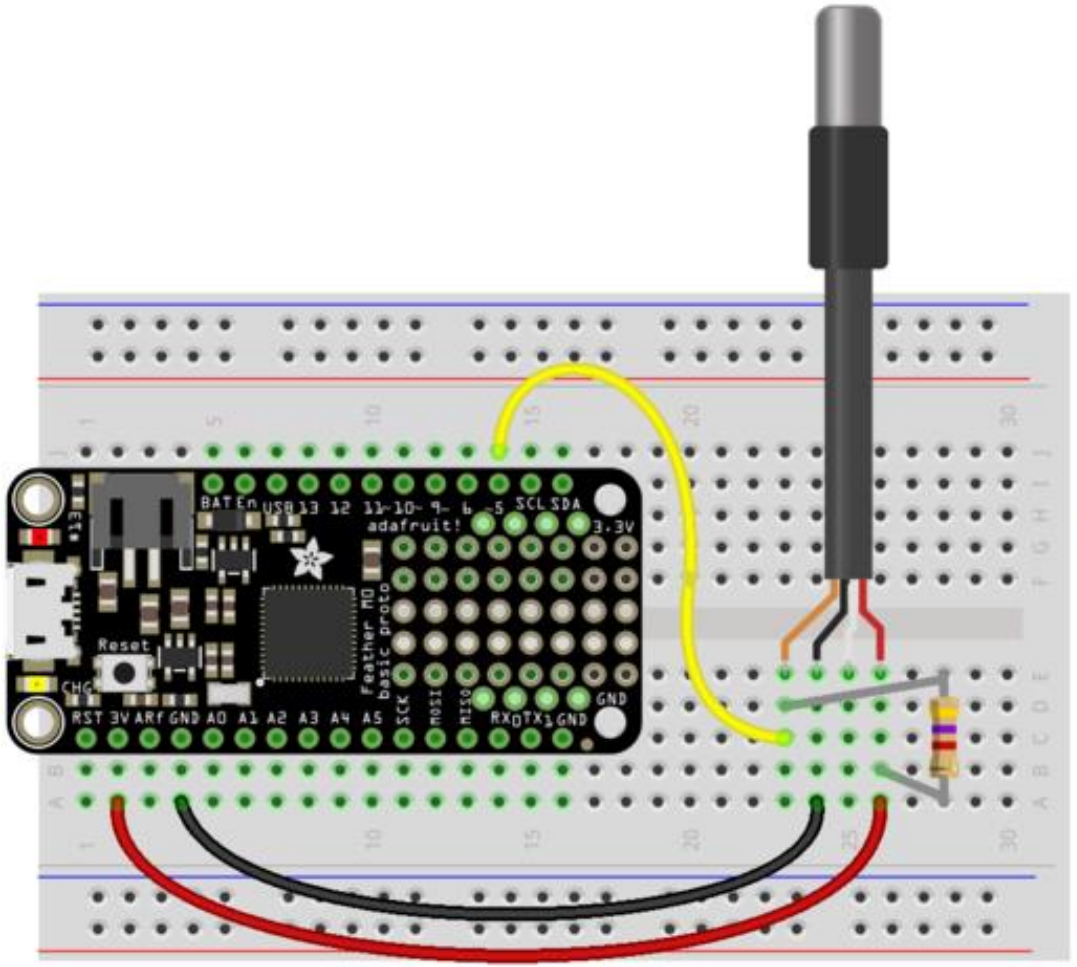
Environment Sensors and IOT for Solar Farm





Sensor : DS18B20

Environment Sensors and IOT for Solar Farm



For high temperature probe (white insulation) connect:
- White w/ orange stripe = 3.3-5V
- White w/ blue stripe = signal (D5)
- White (solid) = ground

Connect 4.7k resistor from white w/ orange to white w/ blue (signal to 3.3/5V).

fritzing

Sensor : DS18B20

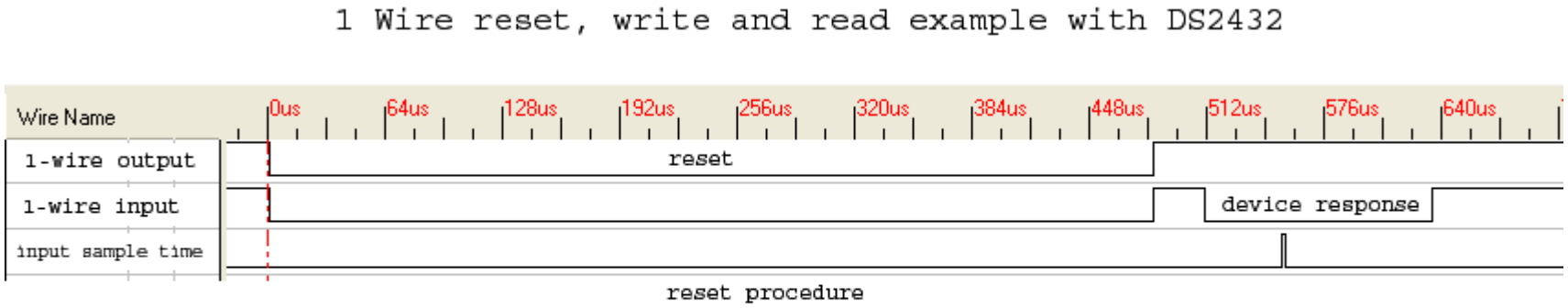
Environment Sensors and IOT for Solar Farm

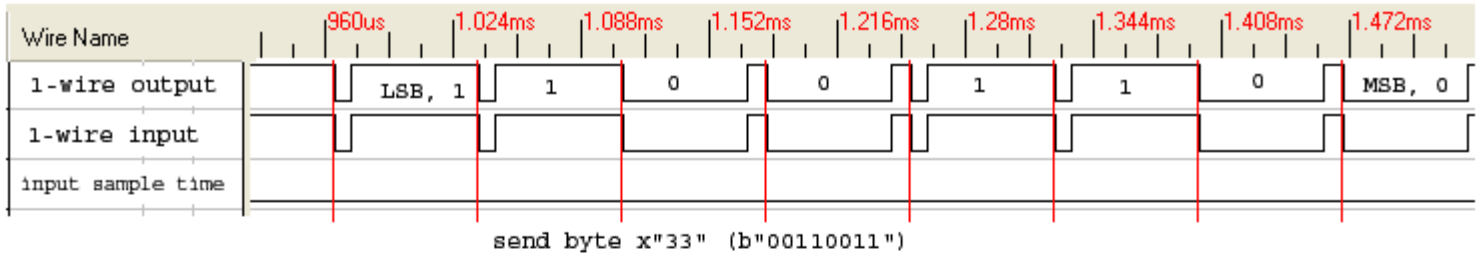
Features

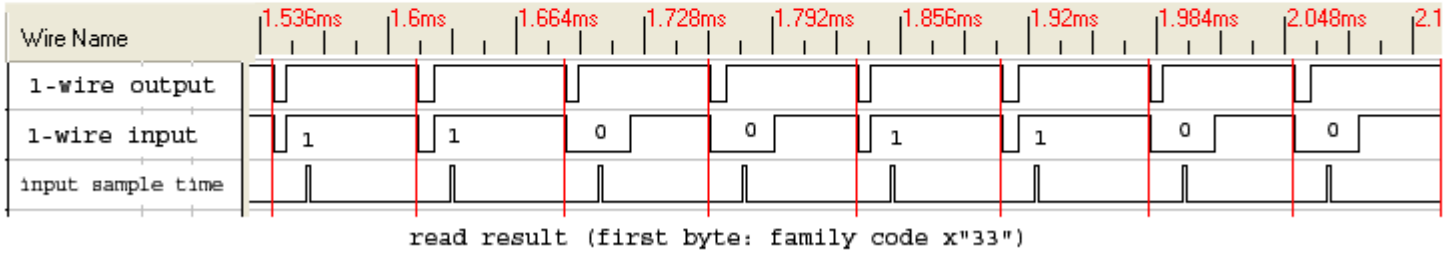
- Temperature sensor chip
- Stainless steel tube 6mm diameter by 30mm long
- Cable is 36" long / 91cm, 4mm diameter
- Uses 1-Wire interface- requires only one digital pin for communication
- Unique 64 bit ID burned into chip
- Multiple sensors can share one pin

Sensor : DS18B20

Environment Sensors and IOT for Solar Farm







Sensor : DS18B20

Environment Sensors and IOT for Solar Farm

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged TO GPIO 4
#define ONE_WIRE_BUS 25

// Setup a oneWire instance to communicate with any OneWire devices (
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

// Number of temperature devices found
int numberOfDevices;

// We'll use this variable to store a found device address
DeviceAddress tempDeviceAddress;
```

Sensor : DS18B20

Environment Sensors and IOT for Solar Farm

```
void setup(){
  // start serial port
  Serial.begin(115200);

  // Start up the library
  sensors.begin();

  // Grab a count of devices on the wire
  numberOfDevices = sensors.getDeviceCount();

  // locate devices on the bus
  Serial.print("Locating devices...");
  Serial.print("Found ");
  Serial.print(numberOfDevices, DEC);
  Serial.println(" devices.");

  // Loop through each device, print out address
  for(int i=0;i<numberOfDevices; i++){
    // Search the wire for address
    if(sensors.getAddress(tempDeviceAddress, i)){
      Serial.print("Found device ");
      Serial.print(i, DEC);
      Serial.print(" with address: ");
      printAddress(tempDeviceAddress);
      Serial.println();
    } else {
      Serial.print("Found ghost device at ");
      Serial.print(i, DEC);
      Serial.print(" but could not detect address. Check power and cabling");
    }
  }
}
```

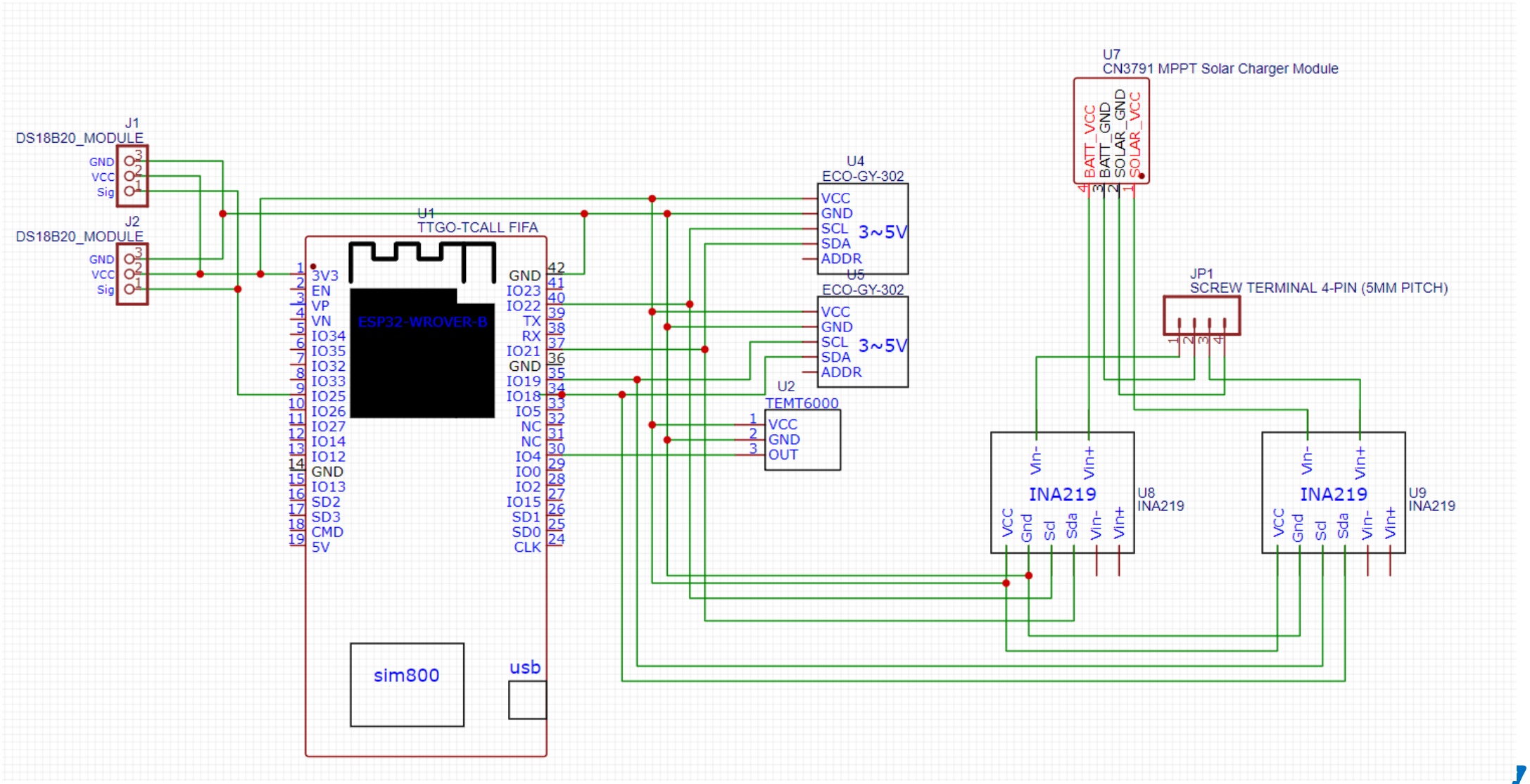
Sensor : DS18B20

Environment Sensors and IOT for Solar Farm

```
void loop(){
  sensors.requestTemperatures(); // Send the command to get temperatures

  // Loop through each device, print out temperature data
  for(int i=0;i<numberOfDevices; i++){
    // Search the wire for address
    if(sensors.getAddress(tempDeviceAddress, i)){
      // Output the device ID
      Serial.print("Temperature for device: ");
      Serial.println(i,DEC);
      // Print the data
      float tempC = sensors.getTempC(tempDeviceAddress);
      Serial.print("Temp C: ");
      Serial.print(tempC);
      Serial.print(" Temp F: ");
      Serial.println(DallasTemperature::toFahrenheit(tempC)); // Converts tempC t
    }
  }
  delay(5000);
}

// function to print a device address
void printAddress(DeviceAddress deviceAddress) {
  for (uint8_t i = 0; i < 8; i++){
    if (deviceAddress[i] < 16) Serial.print("0");
      Serial.print(deviceAddress[i], HEX);
  }
}
```

Sensor : Integrated

Environment Sensors and IOT for Solar Farm



Sensor : Integrated

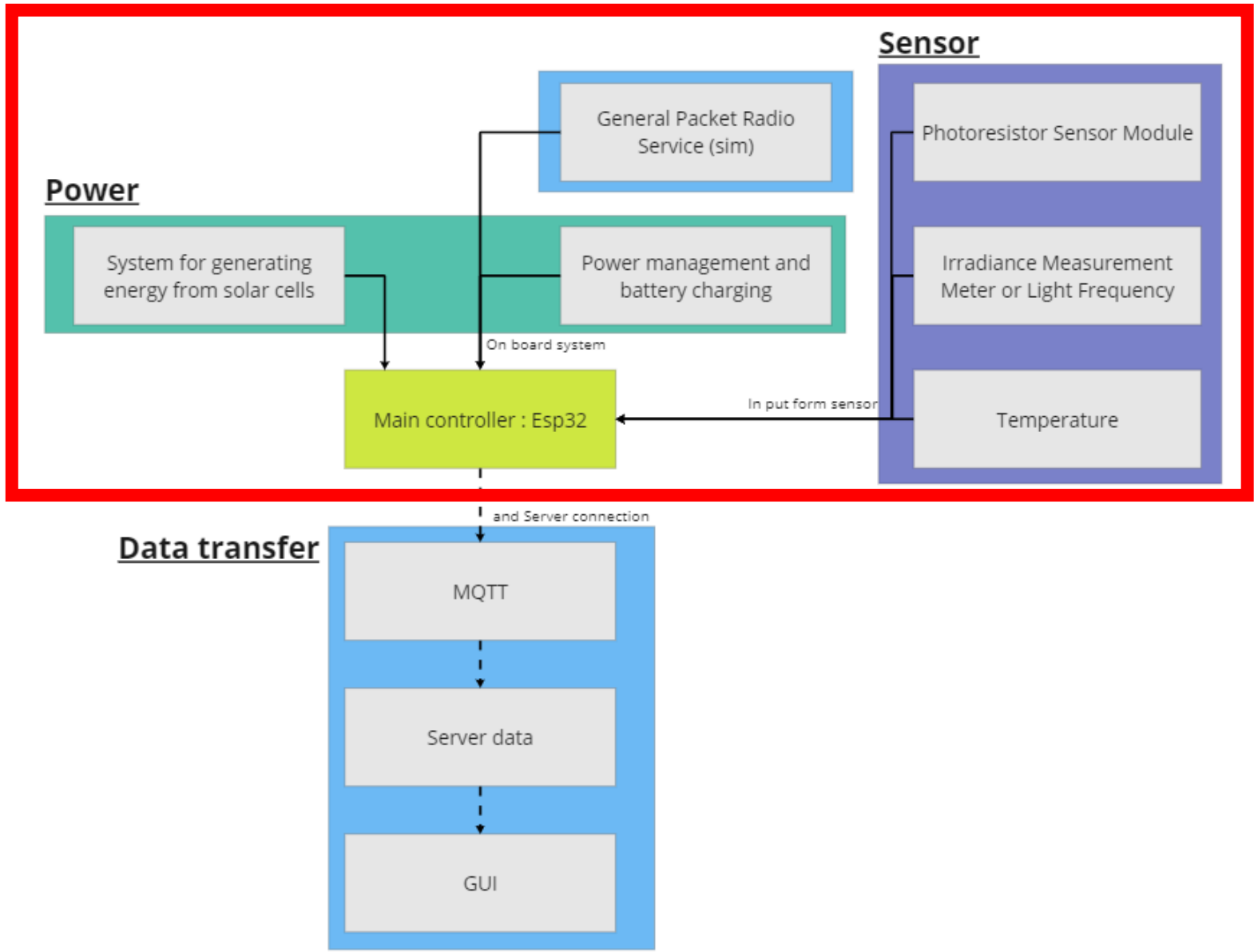
Environment Sensors and IOT for Solar Farm

```
298
299     //payload += lux;
300     //payload += ",";
301     payload += light_level_a;
302     payload += ",";
303     payload += light_level_b;
304     payload += ",";
305     payload += busvoltage_a;
306     payload += ",";
307     payload += busvoltage_b;
308     payload += ",";
309     payload += tempC1;
310     payload += ",";
311     payload += tempC2;
312     payload +=",";
313     payload += loadvoltage_b;
314     payload +=",";
315     payload += power_mW_b;
316     payload +=",";
317     payload += current_mA_b;
318     payload +=",";
319     payload += shuntvoltage_b;
320     payload +=",";
321     payload += loadvoltage_a;
322     payload +=",";
323     payload += power_mW_a;
324     payload +=",";
325     payload += current_mA_a;
326     payload +=",";
327     payload += shuntvoltage_a;
```

```
328
329     int payload_len = payload.length() + 1;
330     char char_array[payload_len];
331     payload.toCharArray(char_array, payload_len);
332     SerialMon.println(char_array);
333     mqtt.publish("SolarBox1", char_array);
334     payload = "";
335 }
```

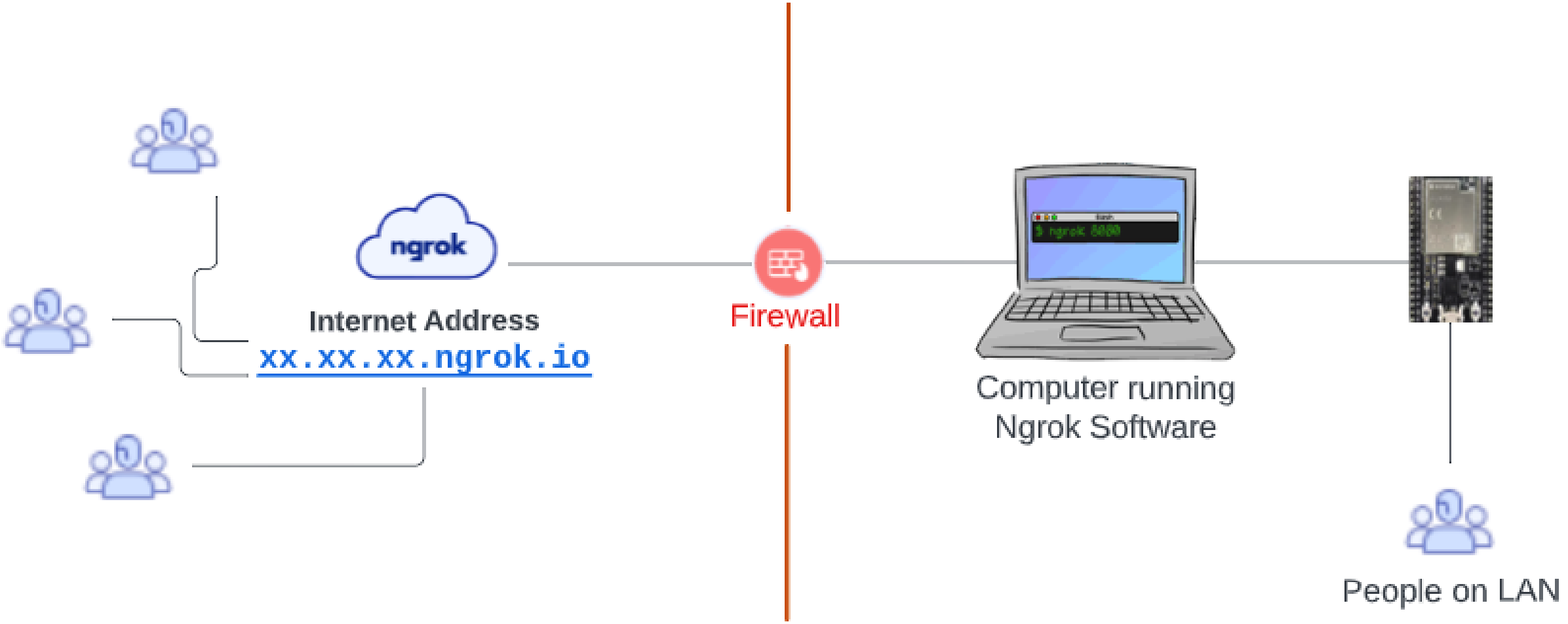
Internet of Things : Architecture Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm



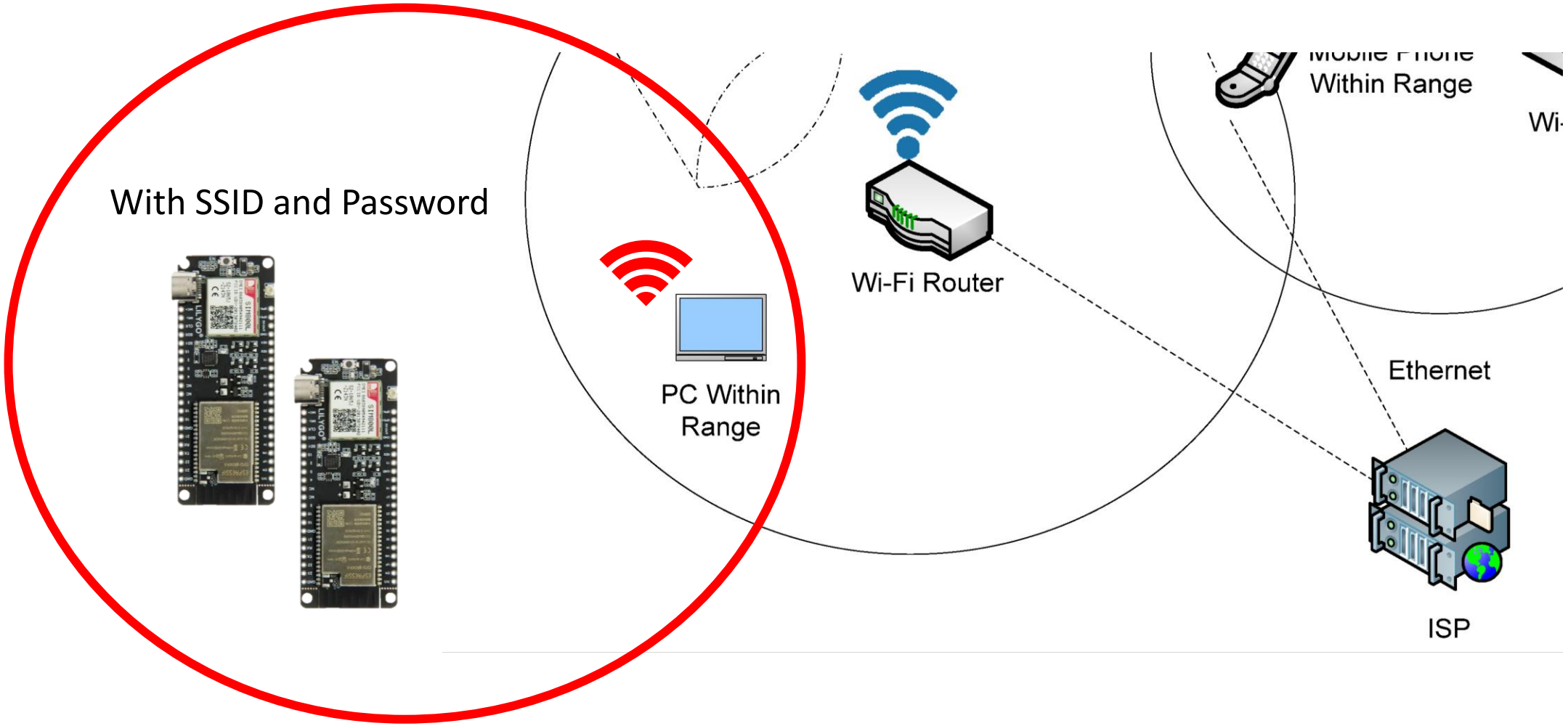
Esp32 : Internet

Environment Sensors and IOT for Solar Farm



Esp32 : Internet

Environment Sensors and IOT for Solar Farm

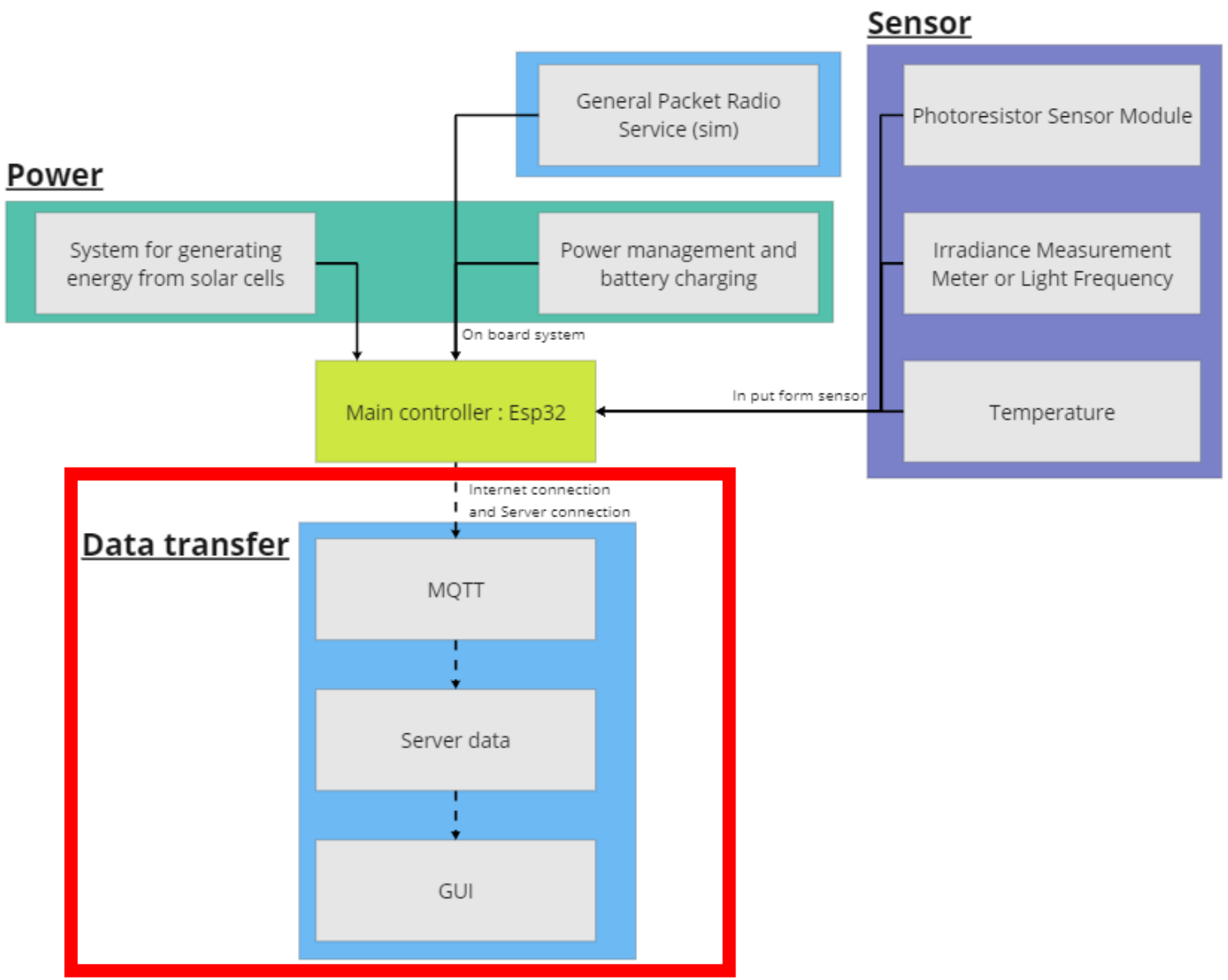


Esp32 : Internet AdvancedWebServer Test


```
30
31  #include <WiFi.h>
32  #include <WiFiClient.h>
33  #include <WebServer.h>
34  #include <ESPmDNS.h>
35
36  const char *ssid = "YourSSIDHere";
37  const char *password = "YourPSKHere";
38
39  WebServer server(80);
40
41  const int led = 13;
42
43  void handleRoot() {
44    digitalWrite(led, 1);
45    char temp[400];
46    int sec = millis() / 1000;
47    int min = sec / 60;
48    int hr = min / 60;
49
50    snprintf(temp, 400,
```

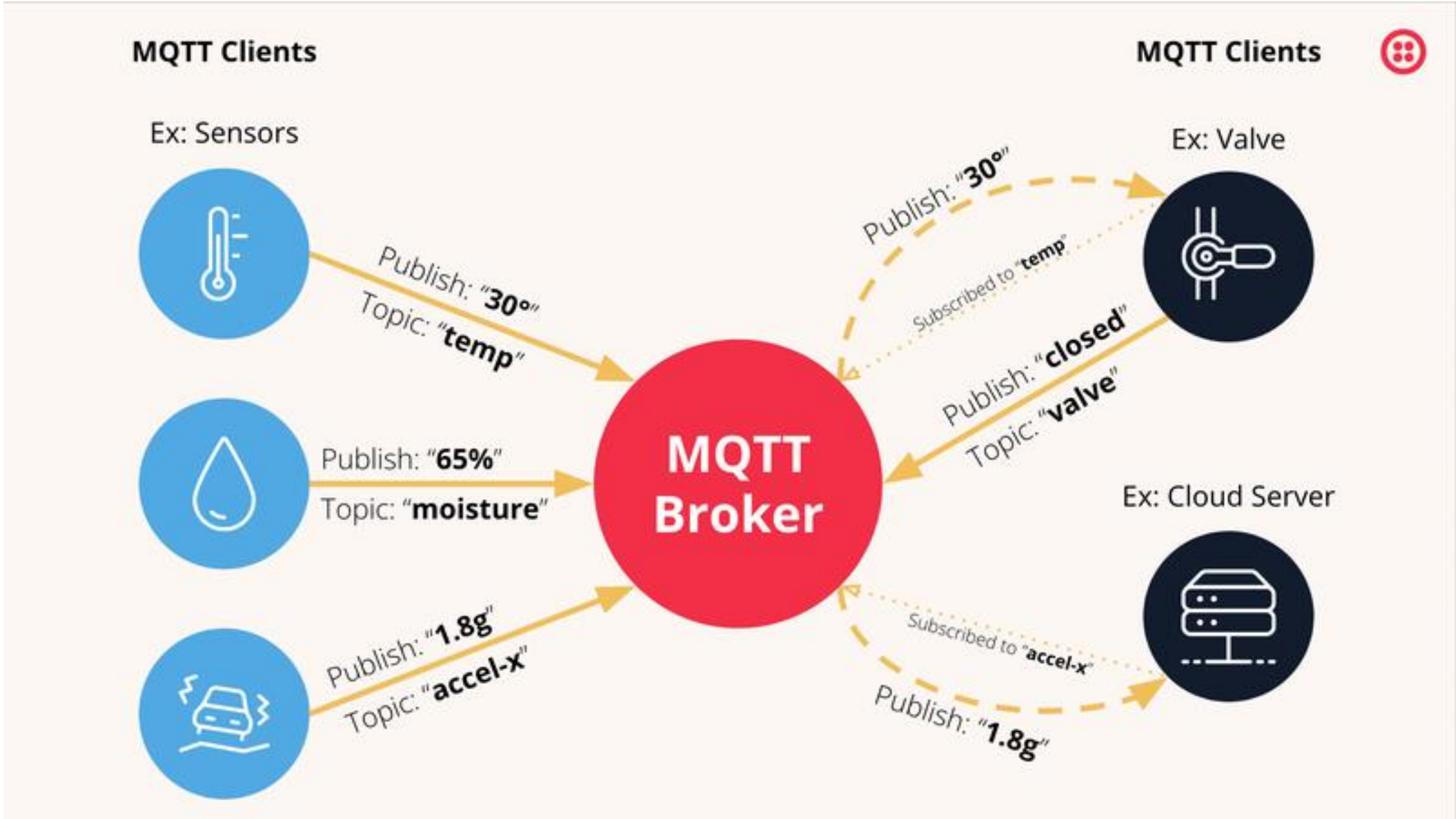
Internet of Things : Architecture Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm



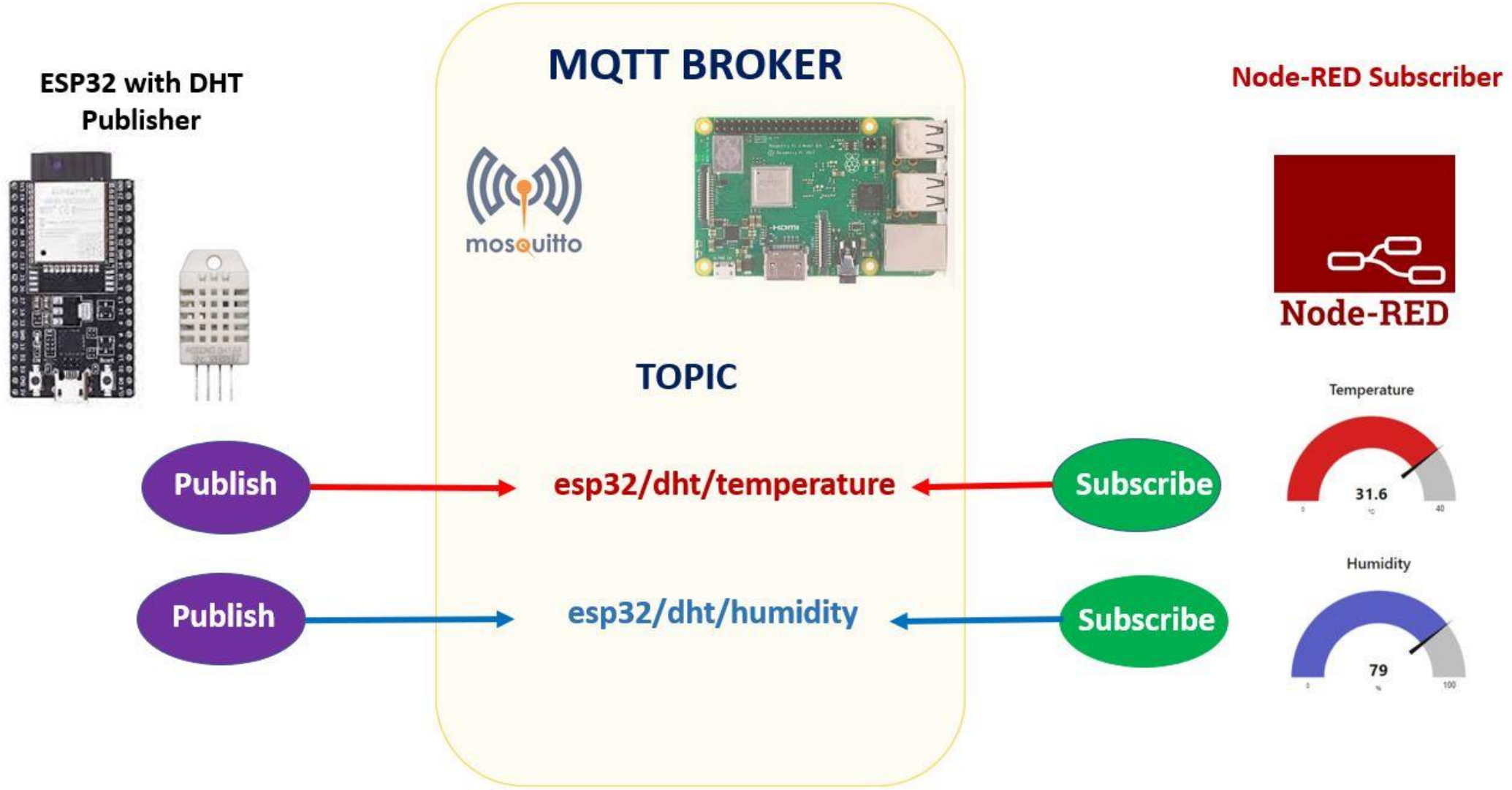
MQTT : Introduction

Environment Sensors and IOT for Solar Farm



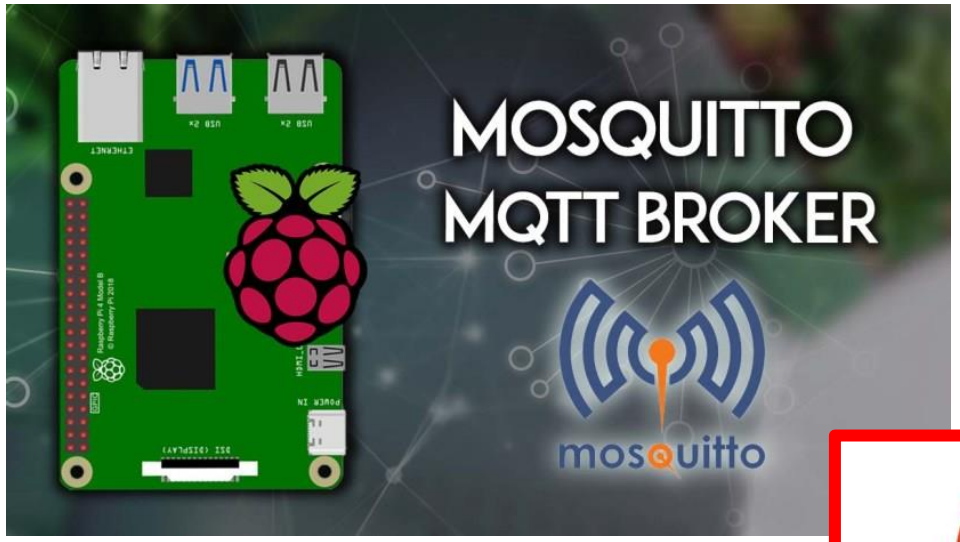
Internet of Things : Architecture

Environment Sensors and IOT for Solar Farm

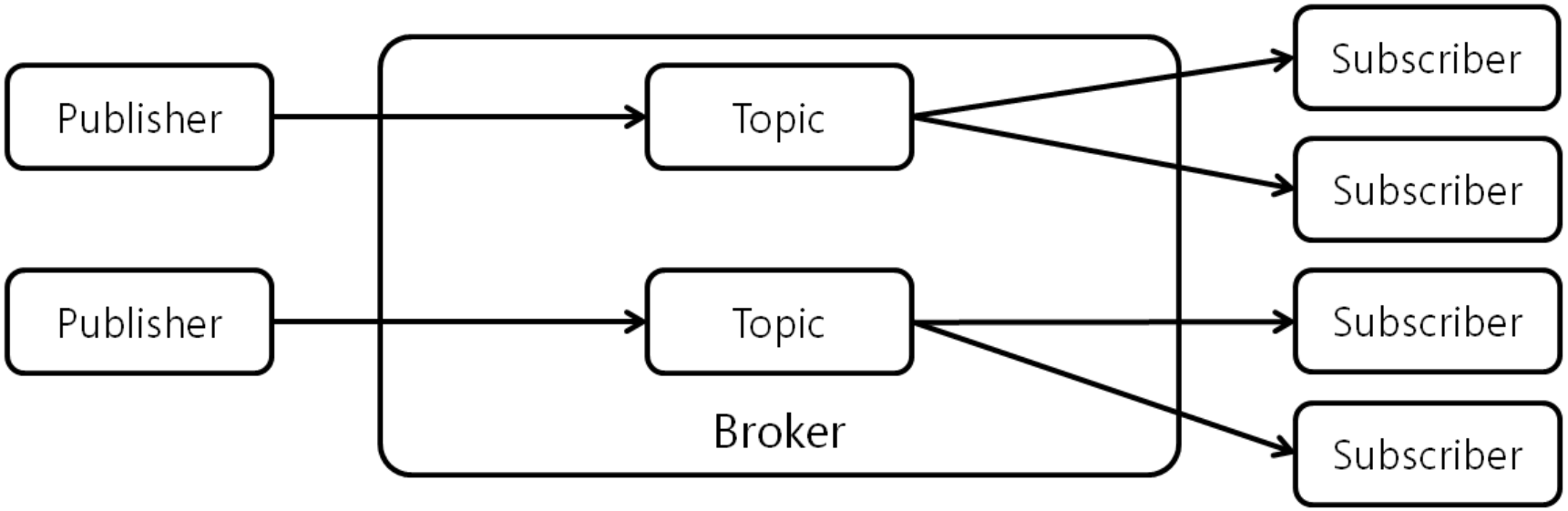


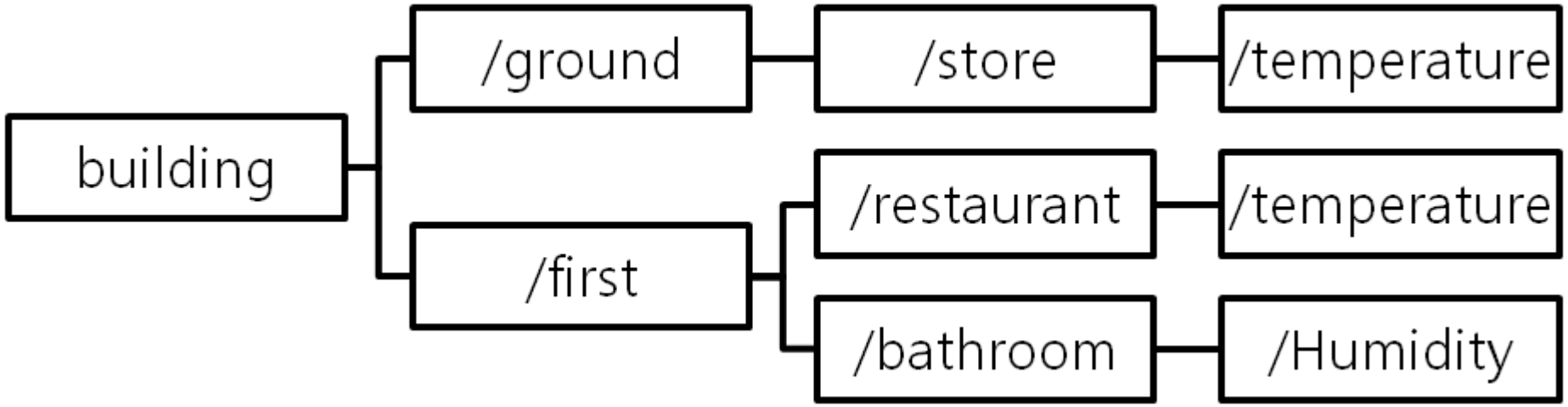
MQTT : Setup

Environment Sensors and IOT for Solar Farm



Esp32 : MQTT Topic Subscribe

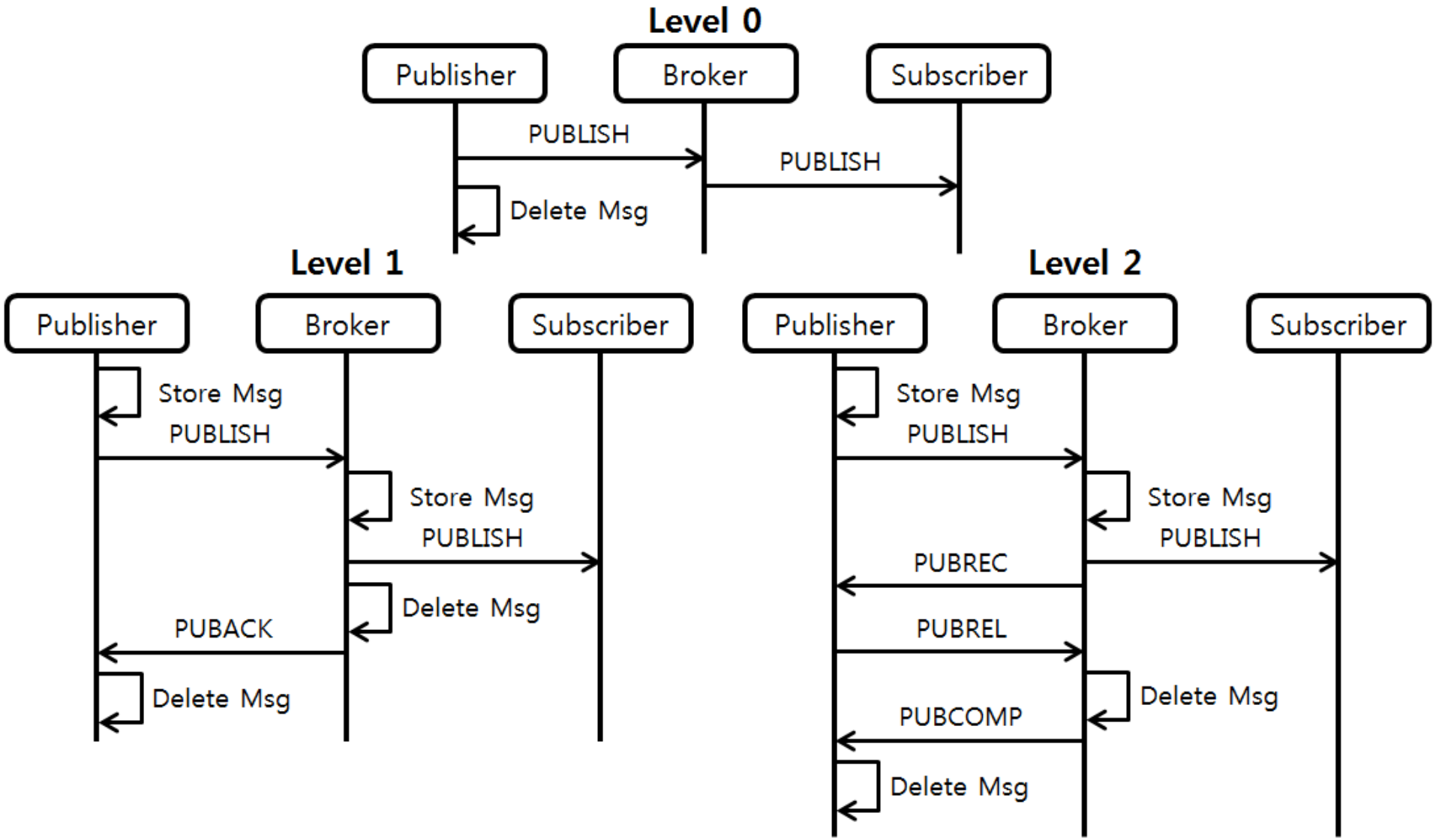




Esp32 : MQTT Topic Subscribe Code test

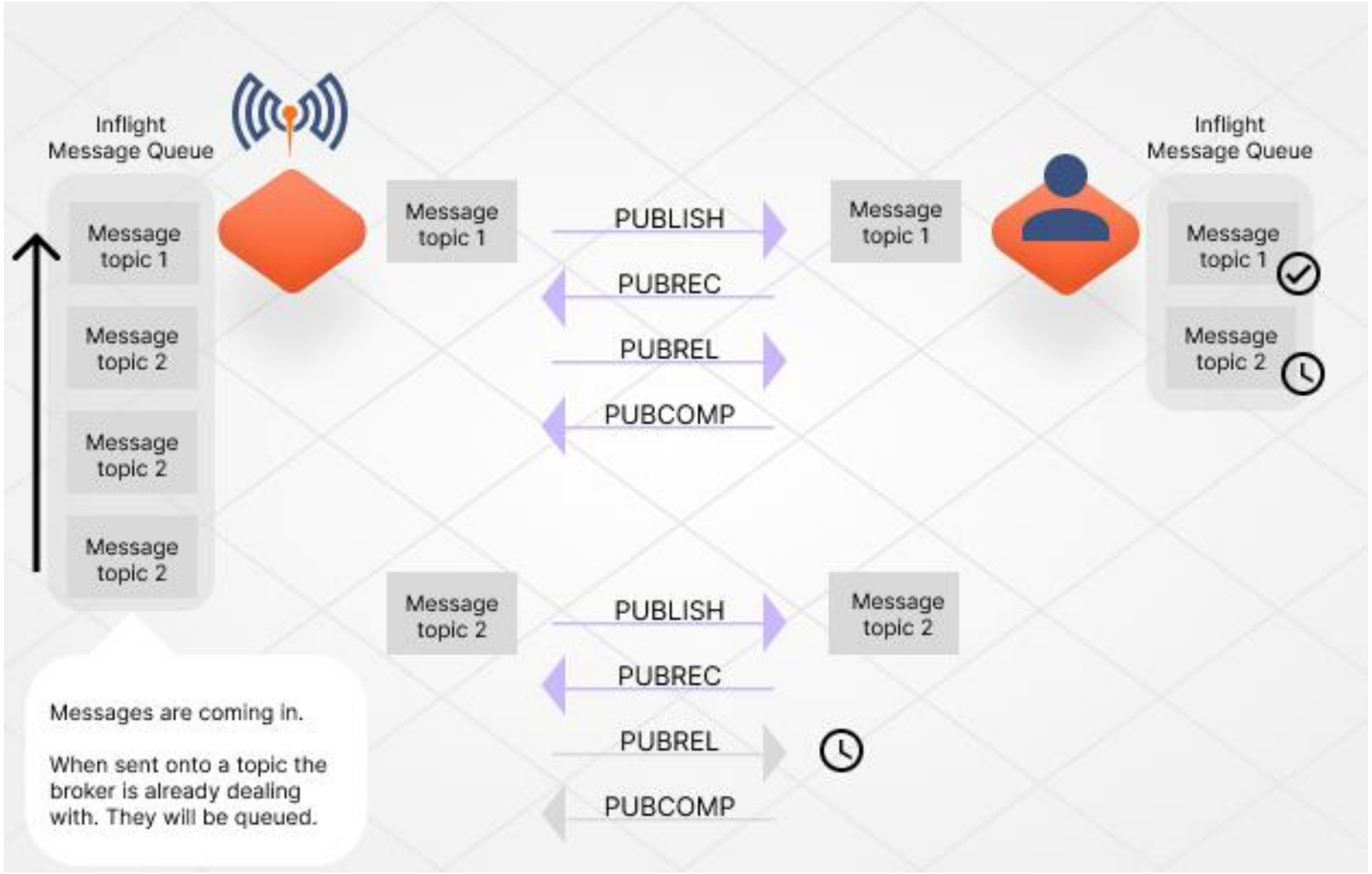
Esp32 : MQTT Topic Subscribe

Environment Sensors and IOT for Solar Farm



Esp32 : MQTT Topic Subscribe

Environment Sensors and IOT for Solar Farm



Esp32 : MQTT Group chat

Environment Sensors and IOT for Solar Farm

```
10  EspMQTTClient client(
11    "WifiSSID",
12    "WifiPassword",
13    "192.168.1.100",   //
14    "MQTTUsername",    //
15    "MQTTPassword",    //
16    "TestClient",      //
17    1883               //
18  );
```

```
19
20  void setup()
21  {
22    Serial.begin(115200);
23
24    // Optional functionalities of EspMQTTClient
25    client.enableDebuggingMessages(); // Enable debugging messages sent to serial output
26    client.enableHTTPWebUpdater(); // Enable the web updater. User and password default to values of MQTTUserr
27    client.enableOTA(); // Enable OTA (Over The Air) updates. Password defaults to MQTTPassword. Port is the d
28    client.enableLastWillMessage("TestClient/lastwill", "I am going offline");  // You can activate the retair
29  }
30
31  // This function is called once everything is connected (Wifi and MQTT)
32  // WARNING : YOU MUST IMPLEMENT IT IF YOU USE EspMQTTClient
33  void onConnectionEstablished()
34  {
35    // Subscribe to "mytopic/test" and display received message to Serial
36    client.subscribe("mytopic/test", [](const String & payload) {
37      Serial.println(payload);
38    });
39
40    // Subscribe to "mytopic/wildcardtest/#" and display received message to Serial
41    client.subscribe("mytopic/wildcardtest/#", [](const String & topic, const String & payload) {
42      Serial.println("(From wildcard) topic: " + topic + ", payload: " + payload);
43    });
44
45    // Publish a message to "mytopic/test"
46    client.publish("mytopic/test", "This is a message"); // You can activate the retain flag by setting the th
47
48    // Execute delayed instructions
49    client.executeDelayed(5 * 1000, []() {
50      client.publish("mytopic/wildcardtest/test123", "This is a message sent 5 seconds later");
51    });
52  }
53
54  void loop()
55  {
56    client.loop();
57  }
```

Esp32 : MQTT Topic Subscribe

Environment Sensors and IOT for Solar Farm

```
#include <WiFi.h>
extern "C" {
  #include "freertos/FreeRTOS.h"
  #include "freertos/timers.h"
}
#include <AsyncMqttClient.h>

#define WIFI_SSID "FIFA7799"
#define WIFI_PASSWORD "eieieieiei"

// Digital Ocean MQTT Mosquitto Broker
#define MQTT_HOST IPAddress(10,61,200,42)
// For a cloud MQTT broker, type the domain name
//#define MQTT_HOST "example.com"
#define MQTT_PORT 1883

#define MQTT_USERNAME "tonton"
#define MQTT_PASSWORD "fiboxx"

// Test MQTT Topic
#define MQTT_PUB_TEST "test"

AsyncMqttClient mqttClient;
TimerHandle_t mqttReconnectTimer;
TimerHandle_t wifiReconnectTimer;

unsigned long previousMillis = 0;   // Stores last time temperature
const long interval = 1000;         // Interval at which to publish
```

```
void connectToWifi() {
  Serial.println("Connecting to Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}

void connectToMqtt() {
  Serial.println("Connecting to MQTT...");
  mqttClient.connect();
}

void WiFiEvent(WiFiEvent_t event) {
  Serial.printf("[WiFi-event] event: %d\n", event);
  switch(event) {
    case SYSTEM_EVENT_STA_GOT_IP:
      Serial.println("WiFi connected");
      Serial.println("IP address: ");
      Serial.println(WiFi.localIP());
      connectToMqtt();
      break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
      Serial.println("WiFi lost connection");
      xTimerStop(mqttReconnectTimer, 0); // ensure we don't
      xTimerStart(wifiReconnectTimer, 0);
      break;
  }
}

void onMqttConnect(bool sessionPresent) {
  Serial.println("Connected to MQTT.");
  Serial.print("Session present: ");
  Serial.println(sessionPresent);
}
```

Esp32 : MQTT Topic Subscribe

Environment Sensors and IOT for Solar Farm

```
76  void onMqttDisconnect(AsyncMqttClientDisconnectReason reason) {
77    Serial.println("Disconnected from MQTT.");
78    if (WiFi.isConnected()) {
79      xTimerStart(mqttReconnectTimer, 0);
80    }
81  }
82
83  /*void onMqttSubscribe(uint16_t packetId, uint8_t qos) {
84    Serial.println("Subscribe acknowledged.");
85    Serial.print("  packetId: ");
86    Serial.println(packetId);
87    Serial.print("  qos: ");
88    Serial.println(qos);
89  }
90  void onMqttUnsubscribe(uint16_t packetId) {
91    Serial.println("Unsubscribe acknowledged.");
92    Serial.print("  packetId: ");
93    Serial.println(packetId);
94  }*/
95
96  void onMqttPublish(uint16_t packetId) {
97    Serial.print("Publish acknowledged.");
98    Serial.print("  packetId: ");
99    Serial.println(packetId);
100 }
```

Esp32 : MQTT Topic Subscribe

Environment Sensors and IOT for Solar Farm

```
102  void setup() {
103    Serial.begin(115200);
104    Serial.println();
105
106    mqttReconnectTimer = xTimerCreate("mqttTimer", pdMS_TO_TICKS(2000), pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToMqtt));
107    wifiReconnectTimer = xTimerCreate("wifiTimer", pdMS_TO_TICKS(2000), pdFALSE, (void*)0, reinterpret_cast<TimerCallbackFunction_t>(connectToWifi));
108
109    WiFi.onEvent(WiFiEvent);
110
111    mqttClient.onConnect(onMqttConnect);
112    mqttClient.onDisconnect(onMqttDisconnect);
113    /*mqttClient.onSubscribe(onMqttSubscribe);
114    mqttClient.onUnsubscribe(onMqttUnsubscribe);*/
115    mqttClient.onPublish(onMqttPublish);
116    mqttClient.setServer(MQTT_HOST, MQTT_PORT);
117    // If your broker requires authentication (username and password), set them below
118    mqttClient.setCredentials(MQTT_USERNAME, MQTT_PASSWORD);
119    connectToWifi();
120  }
```

Esp32 : MQTT Topic Subscribe

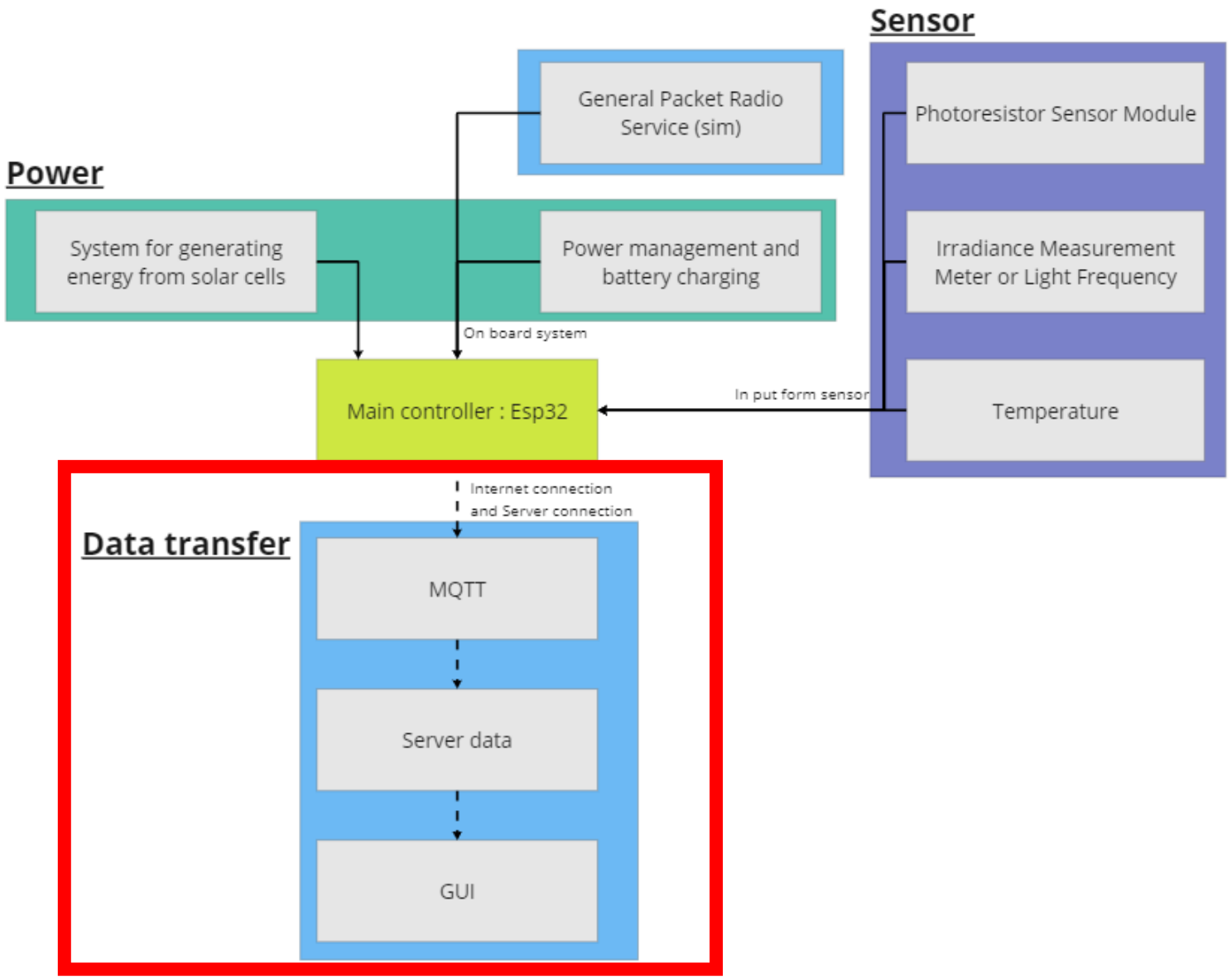

```
122 void loop() {
123   unsigned long currentMillis = millis();
124   // Every X number of seconds (interval = 5 seconds)
125   // it publishes a new MQTT message
126   if (currentMillis - previousMillis >= interval) {
127     // Save the last time a new reading was published
128     previousMillis = currentMillis;
129 
130     String testString = "Hello, world! #" + String(i);
131     // Publish an MQTT message on topic test
132     uint16_t packetIdPub1 = mqttClient.publish(MQTT_PUB_TEST, 1, true, String(testString).c_str());
133     Serial.printf("Publishing on topic %s at QoS 1, packetId: %i", MQTT_PUB_TEST, packetIdPub1);
134     Serial.printf(" Message: %.2f \n", testString);
135     i++;
136   }
137 }
```

Esp32 : MQTT Topic Subscribe


```
328
329     int payload_len = payload.length() + 1;
330     char char_array[payload_len];
331     payload.toCharArray(char_array, payload_len);
332     SerialMon.println(char_array);
333     mqtt.publish("SolarBox1", char_array);
334     payload = "";
335   }
```

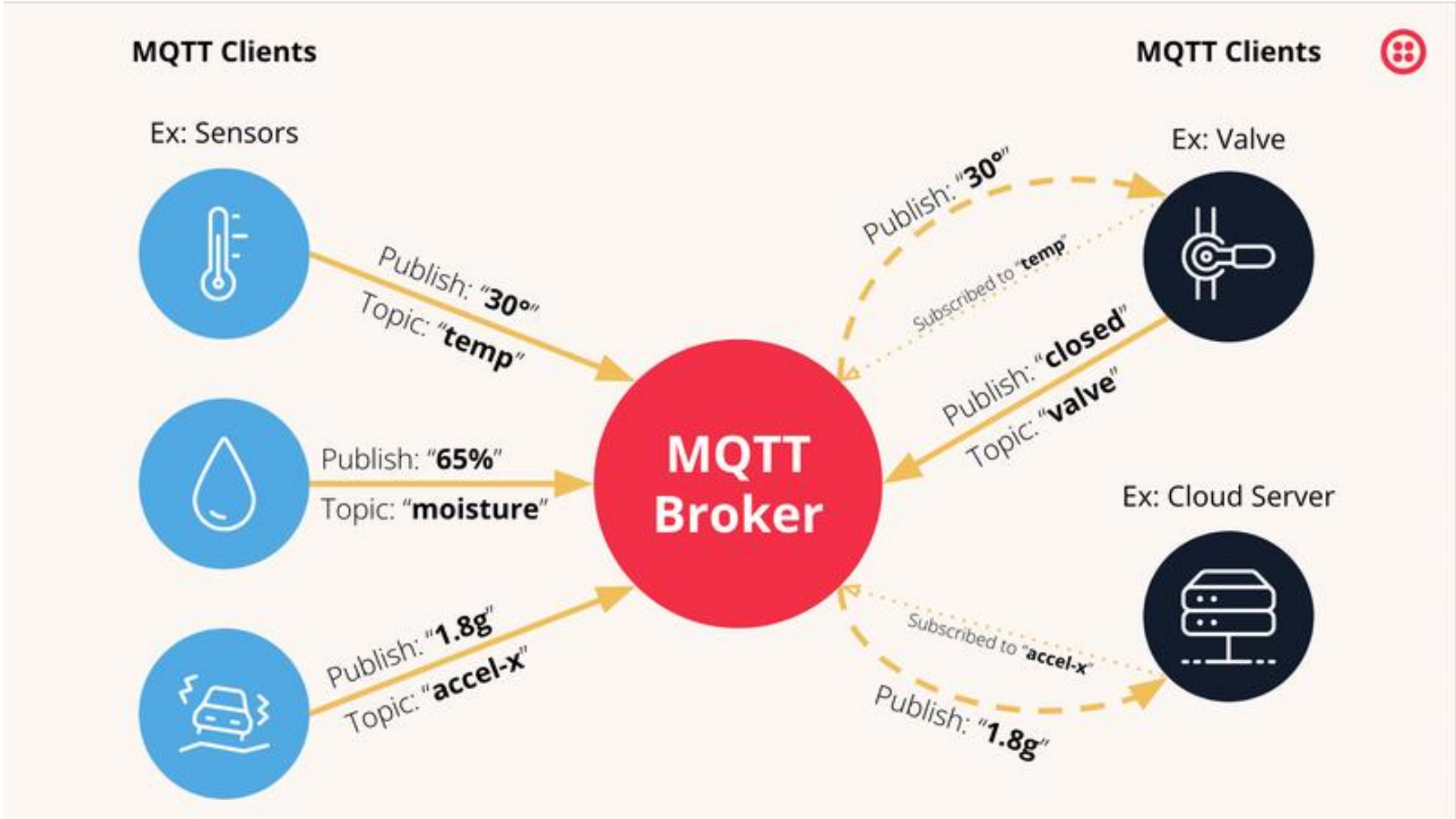
Internet of Things : Architecture Sensors for Solar Farm

Environment Sensors and IOT for Solar Farm



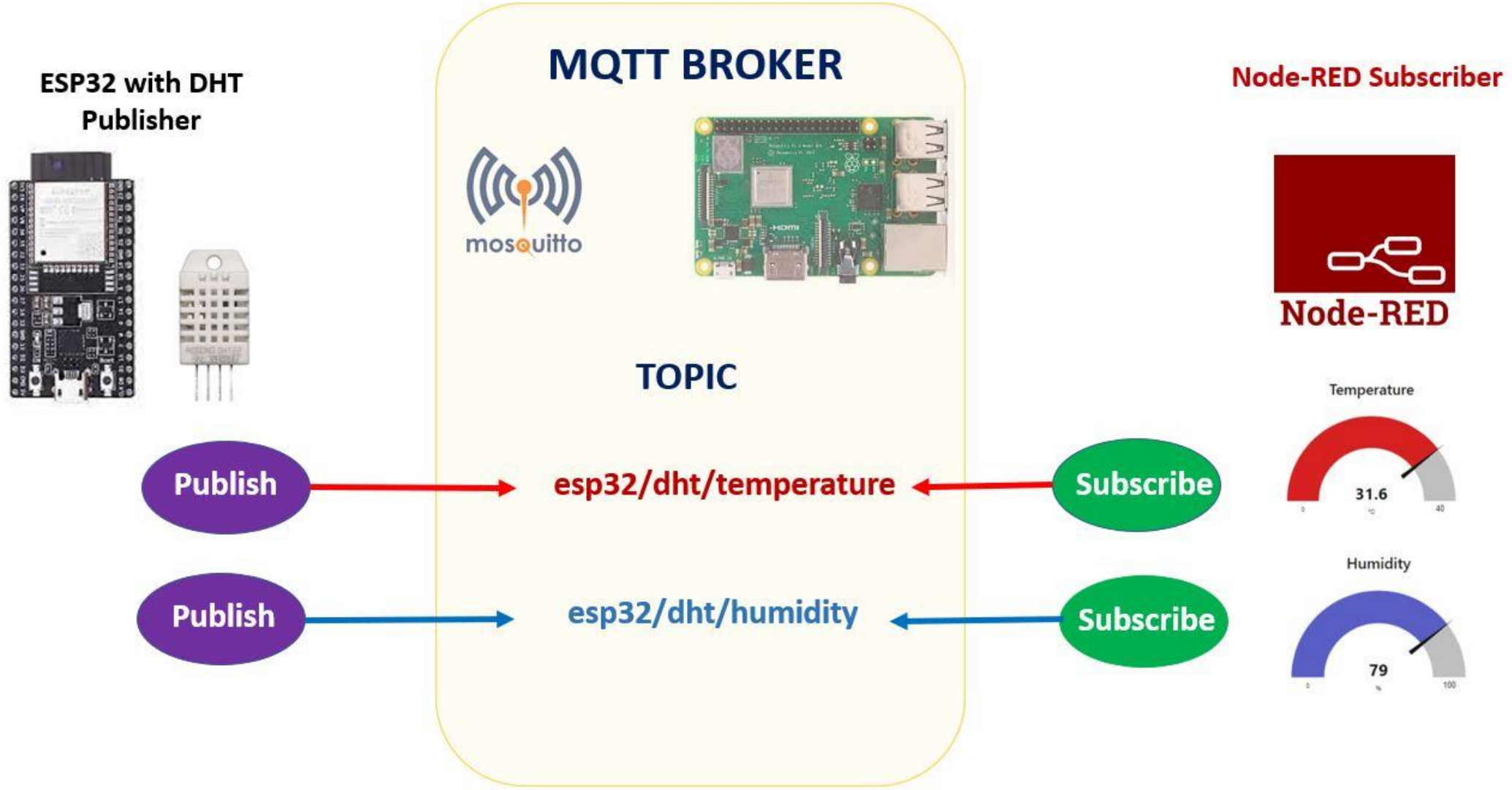
MQTT : Introduction

Environment Sensors and IOT for Solar Farm



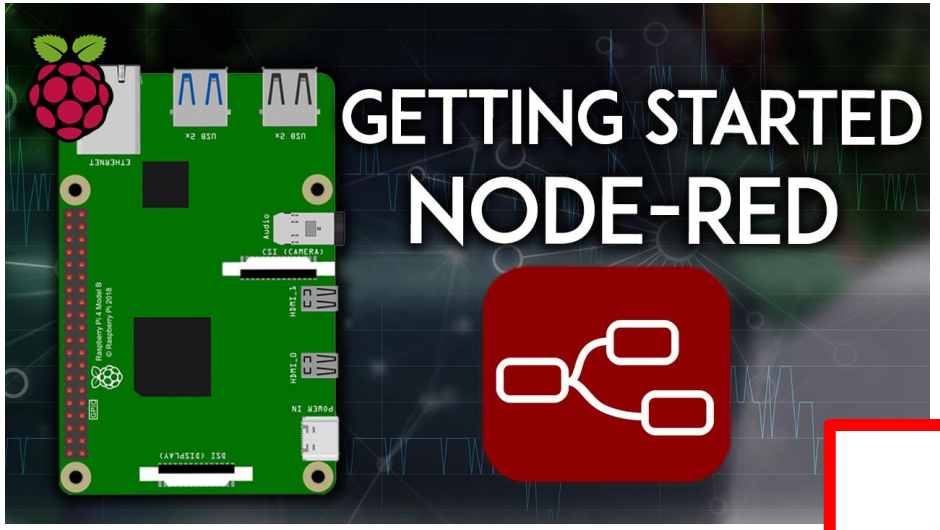
Internet of Things : Architecture

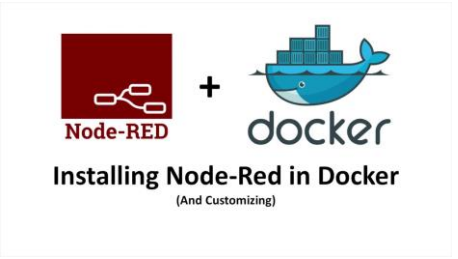
Environment Sensors and IOT for Solar Farm



NODE-RED : Setup

Environment Sensors and IOT for Solar Farm





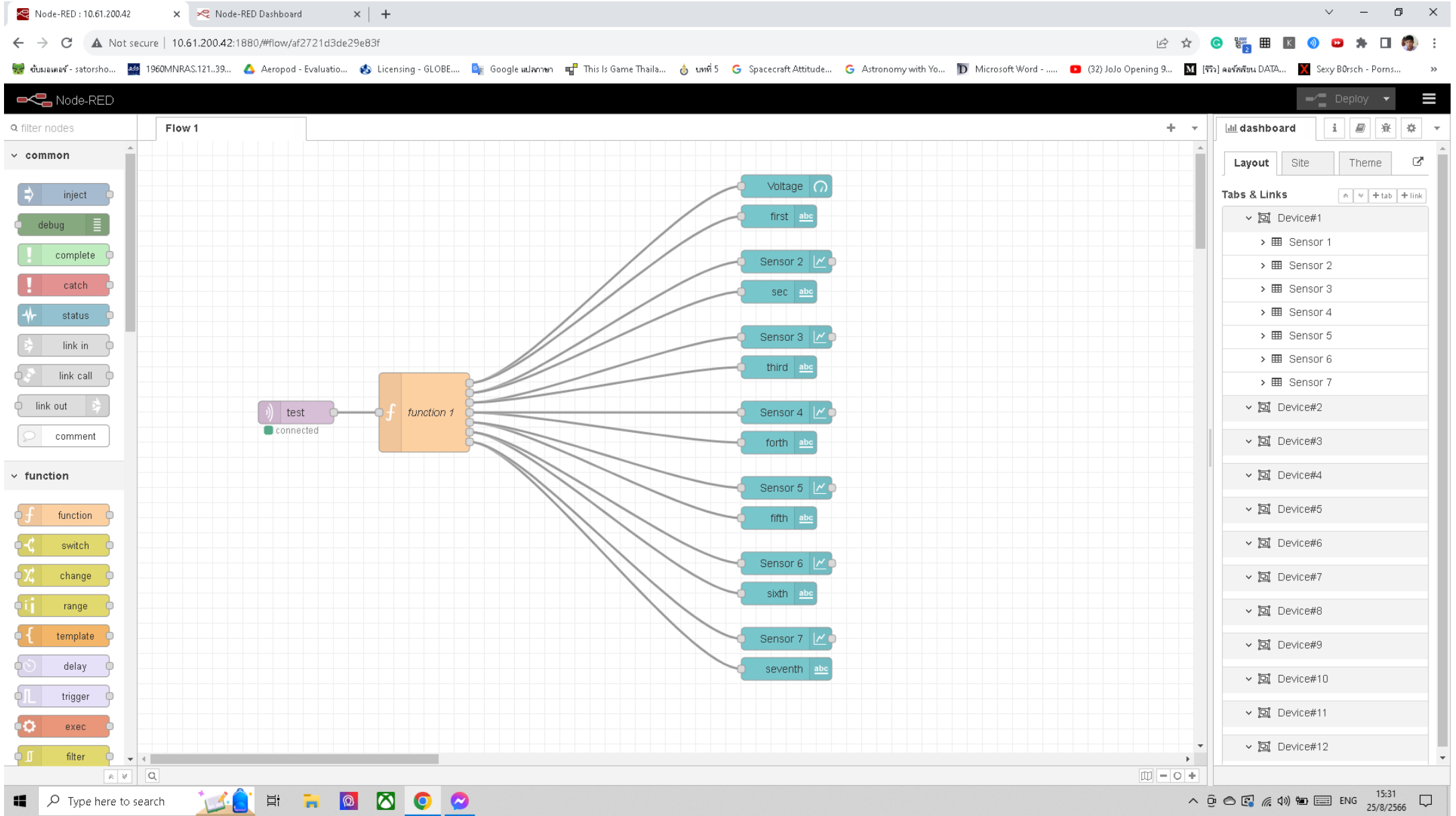
NODE-RED : Application

http://10.61.200.42:1880/#flow/af2721d3de29e83f

ต้องเชื่อมต่ออินเตอร์เน็ต KMUTT-Secure ถึงจะเข้าใช้งานได้

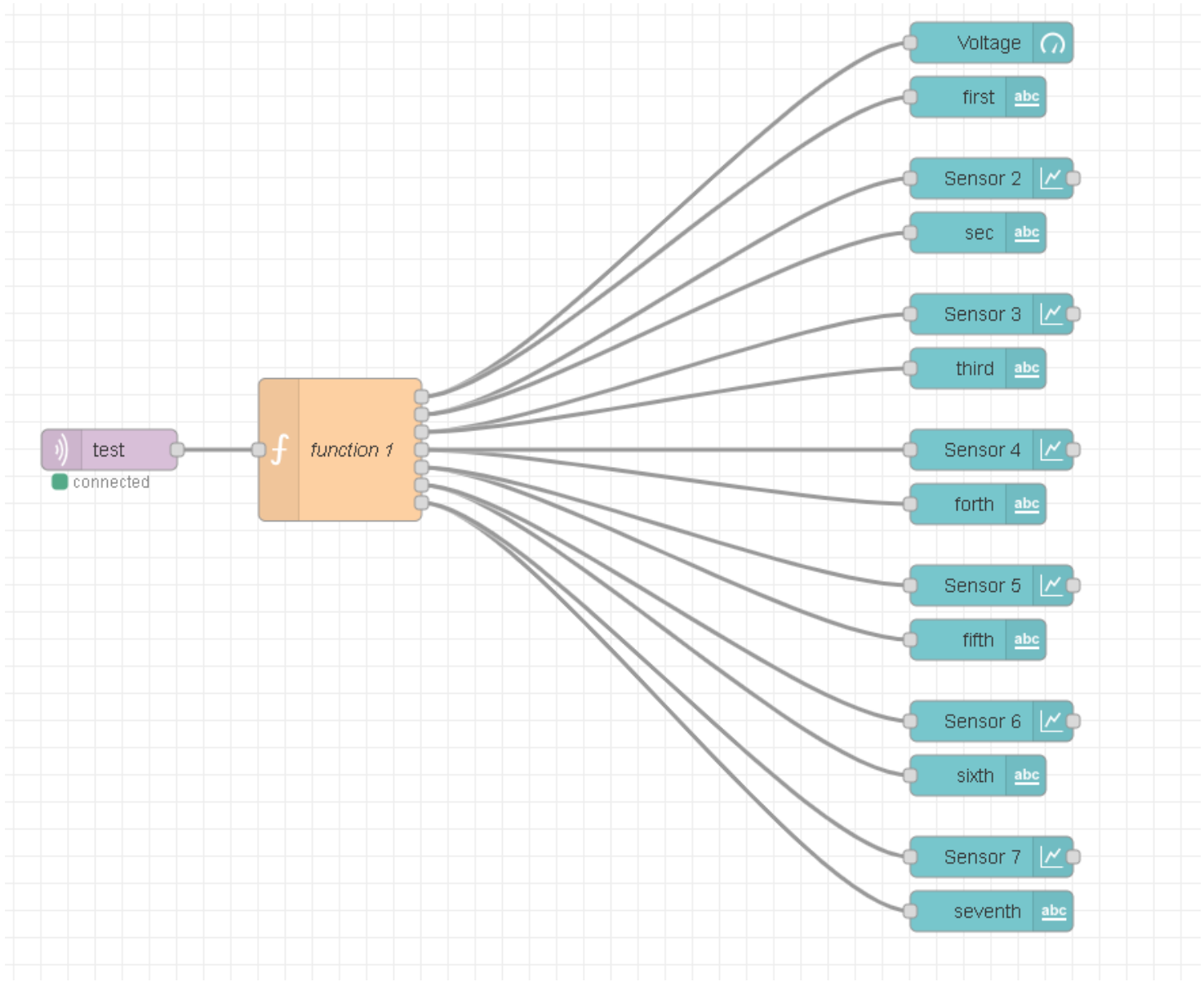
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



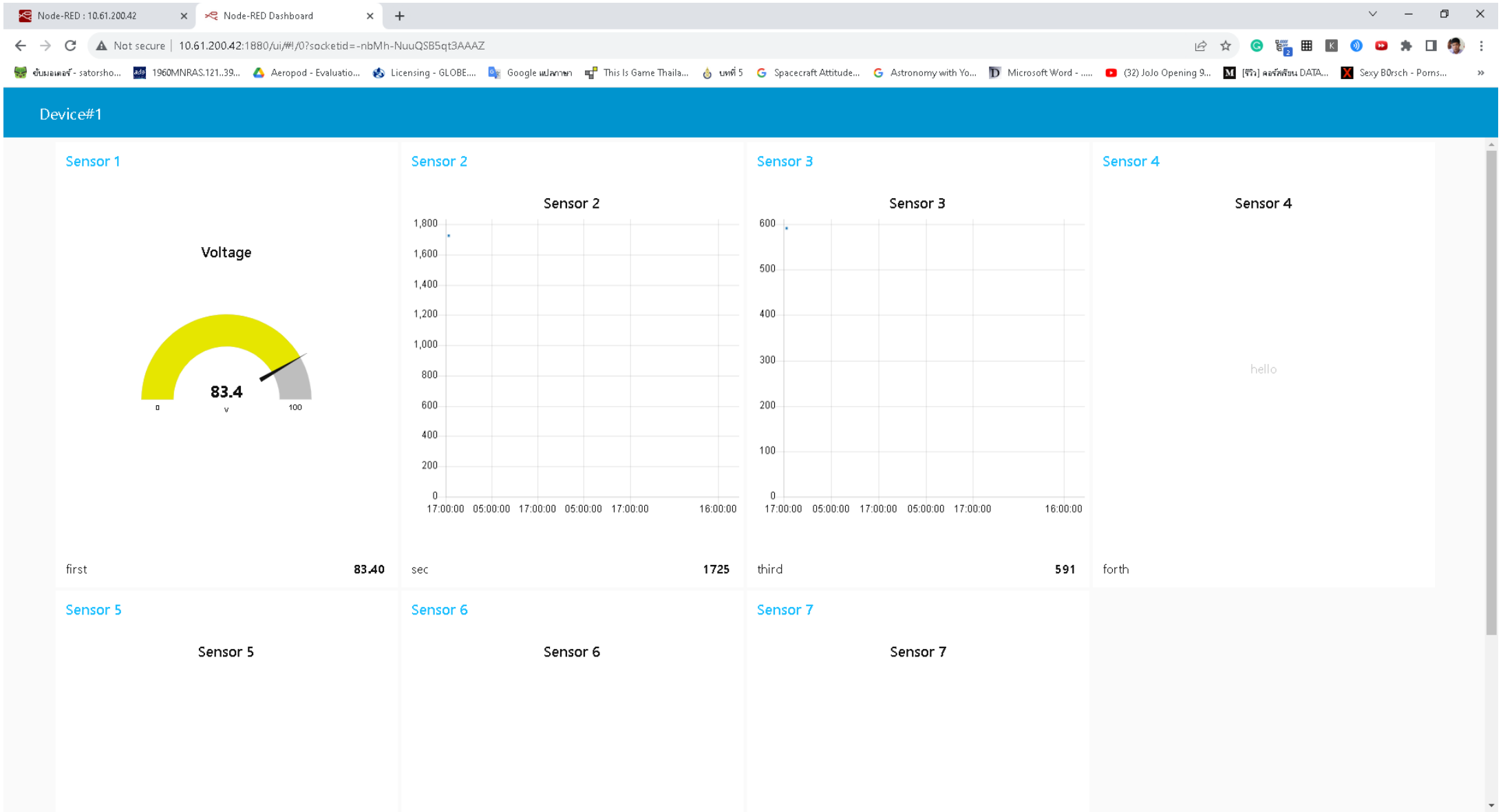
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



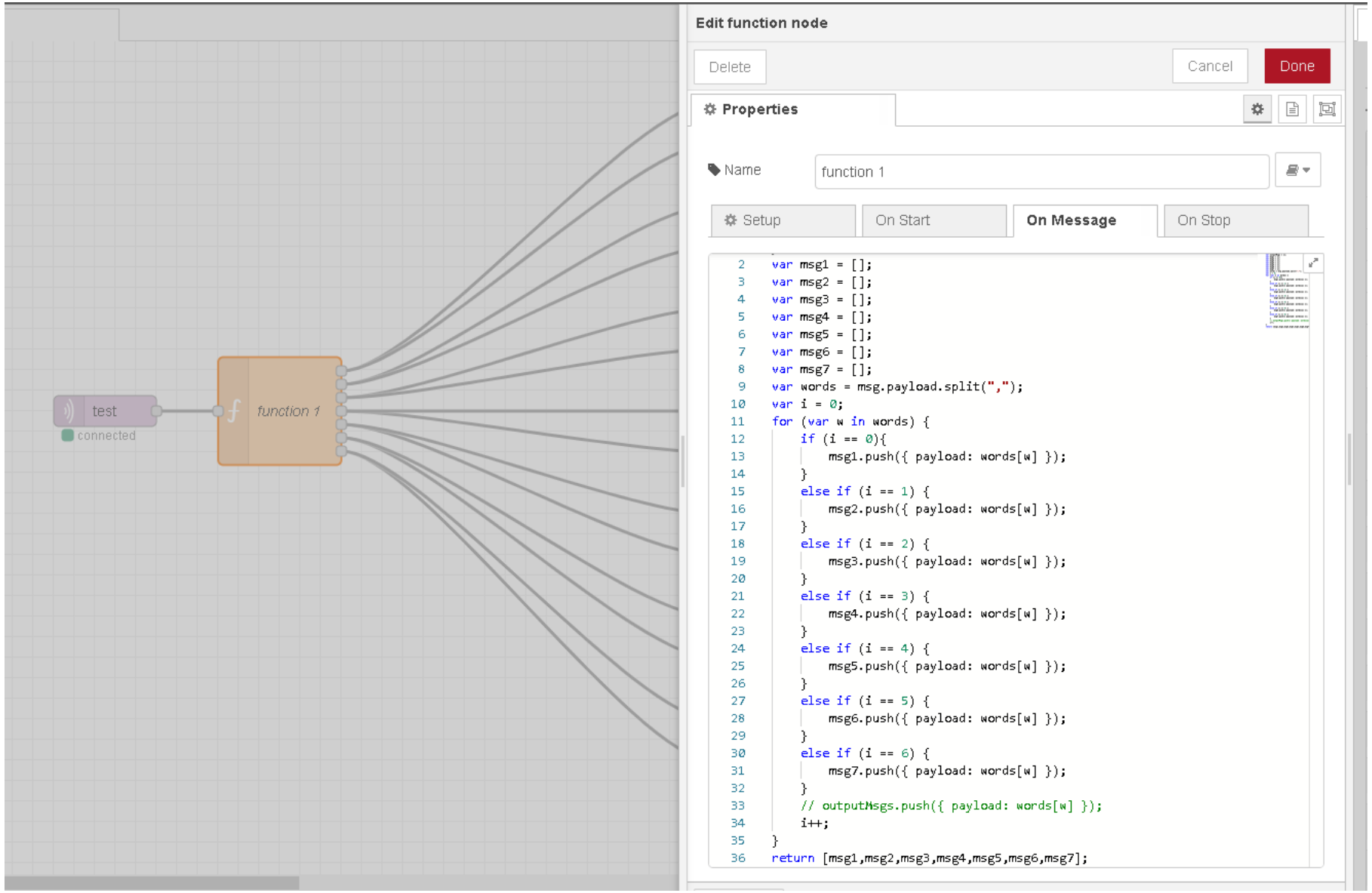
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



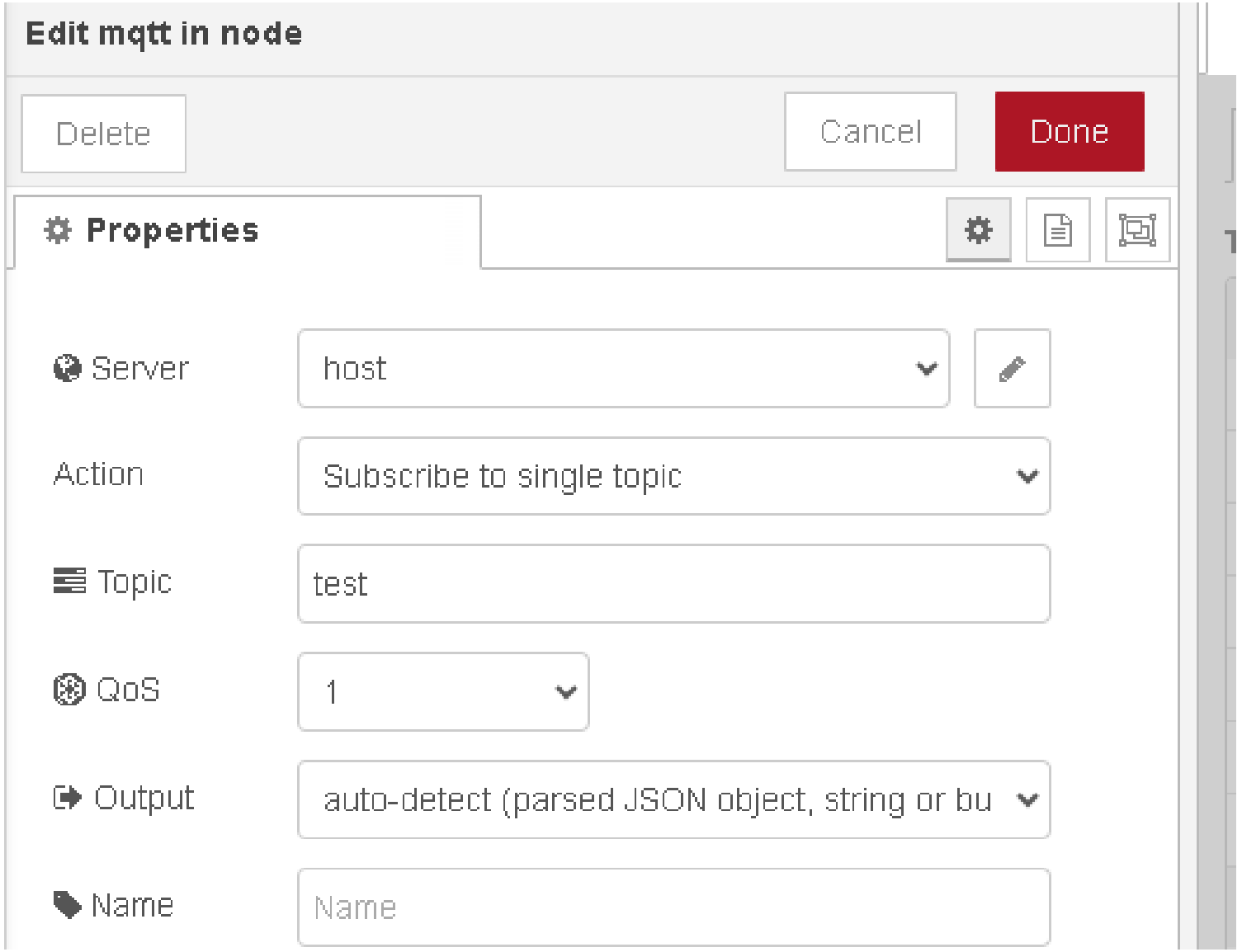
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



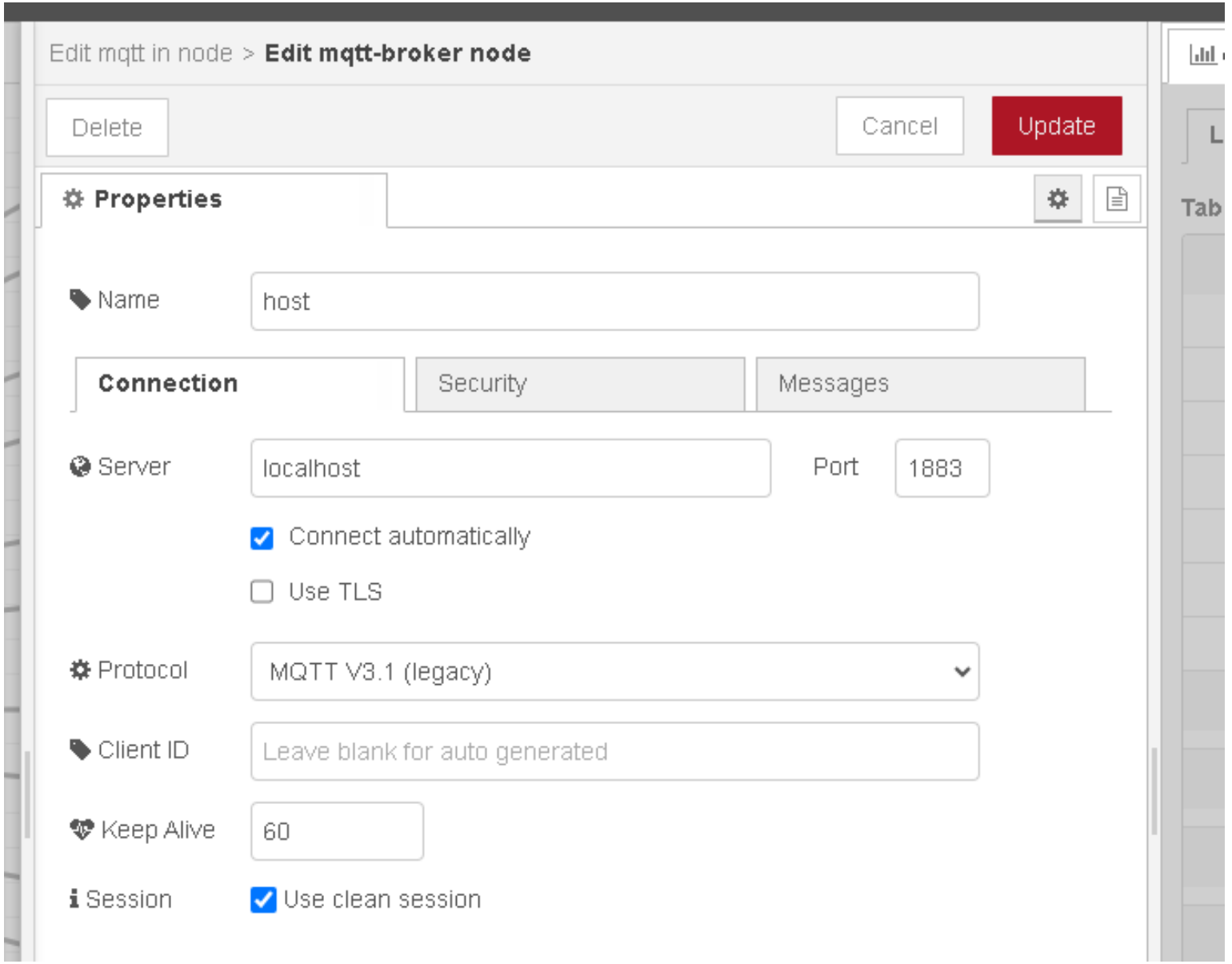
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



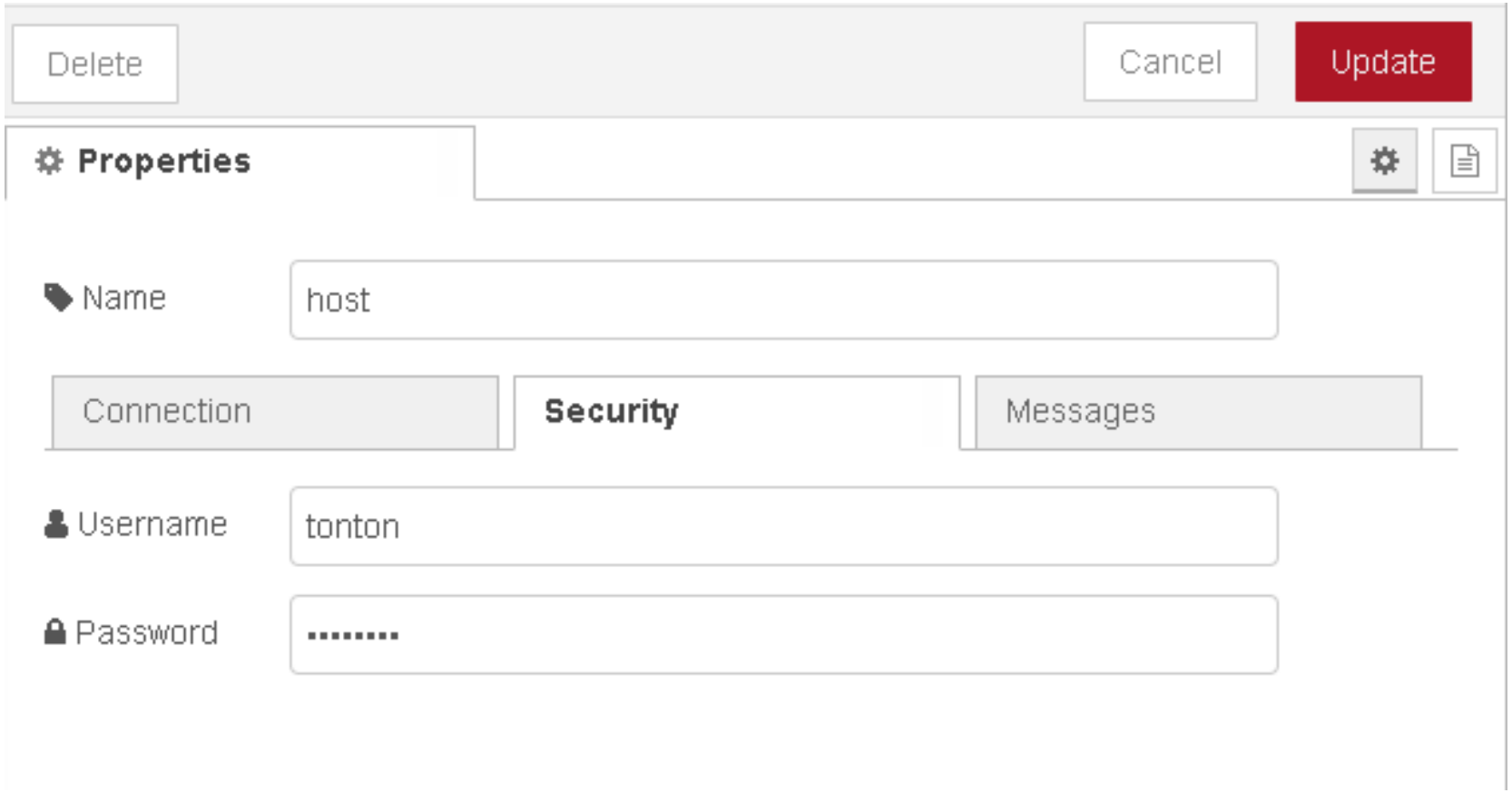
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



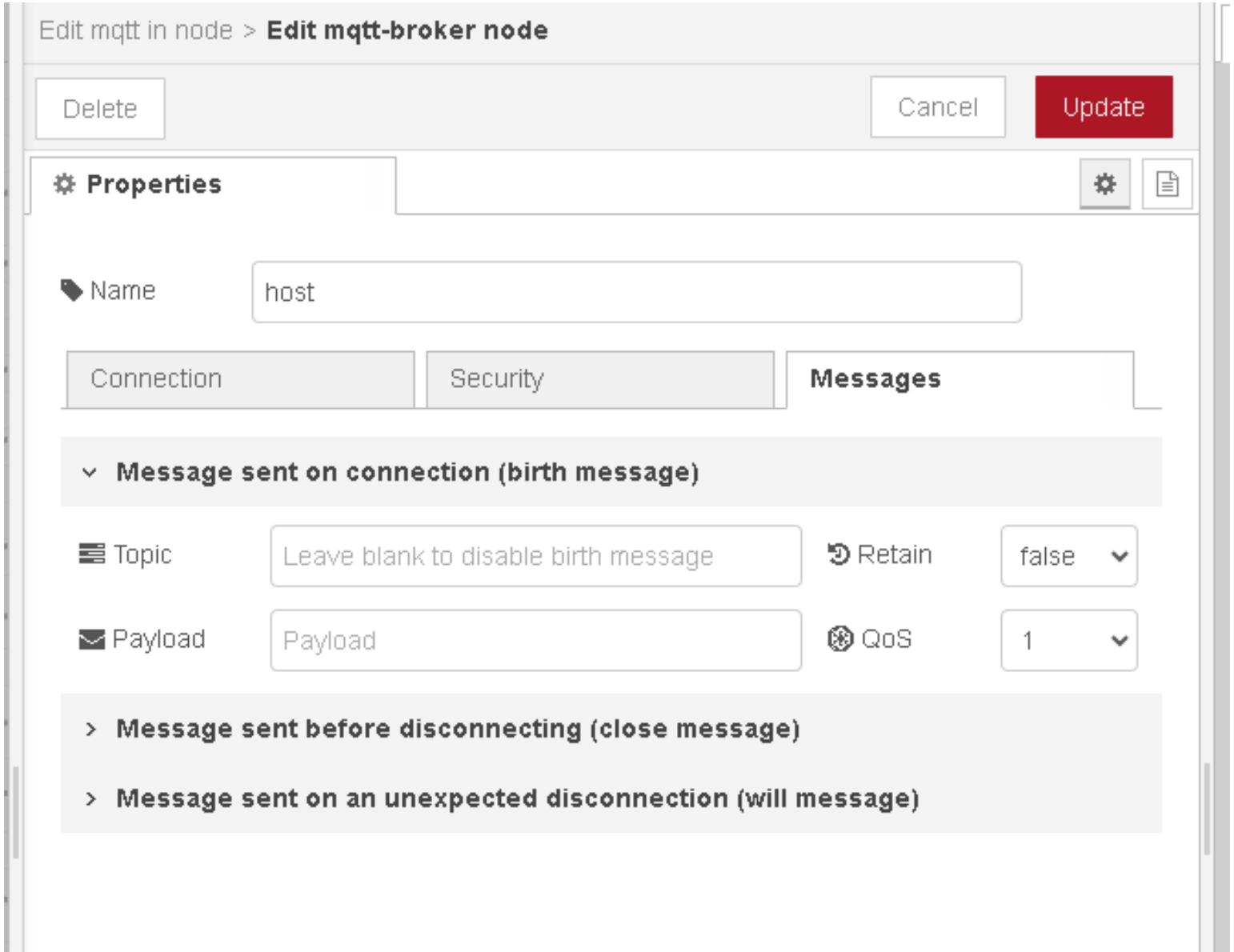
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



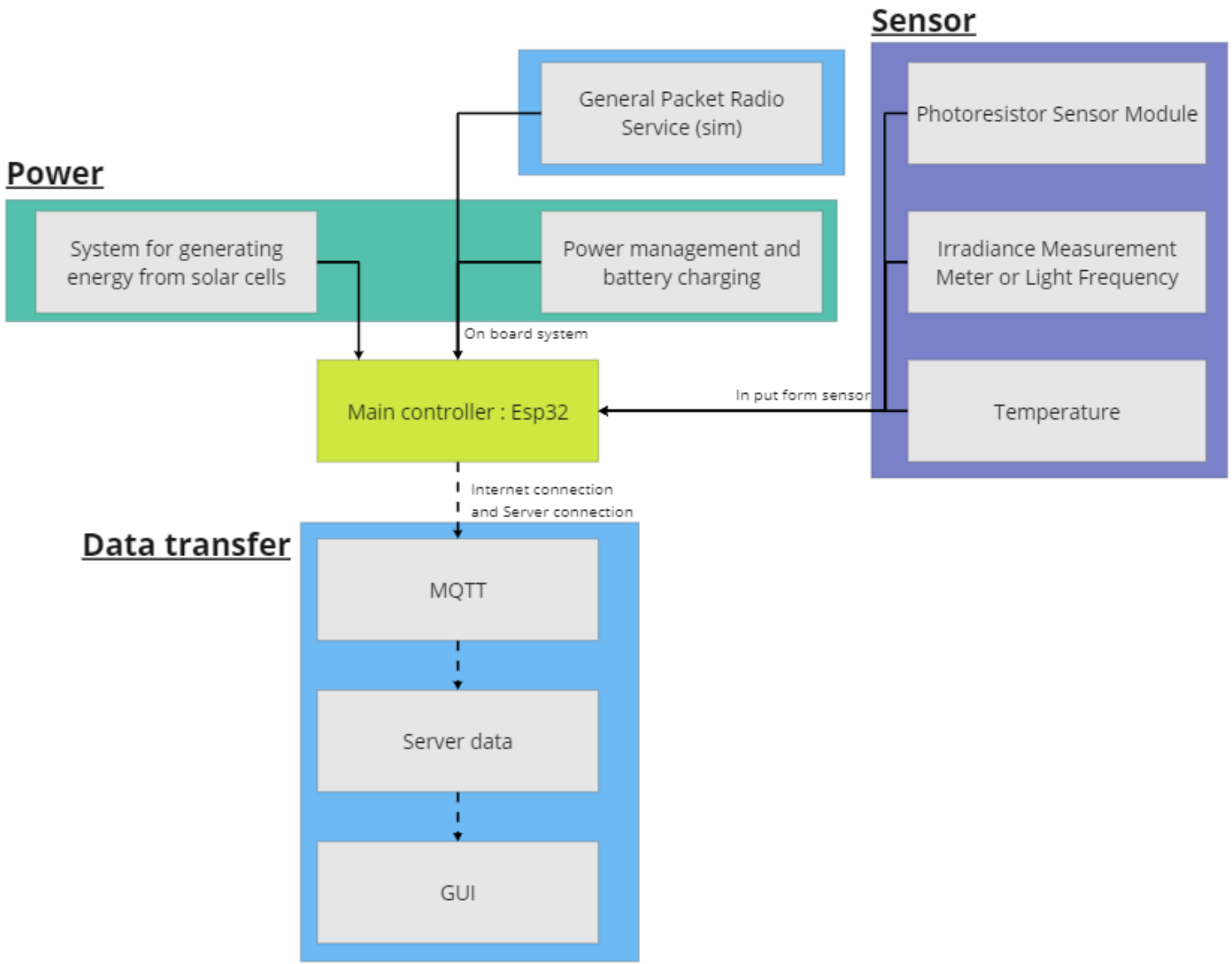
NODE-RED : Application

Environment Sensors and IOT for Solar Farm



Internet of Things : Architecture Sensors for Solar Farm

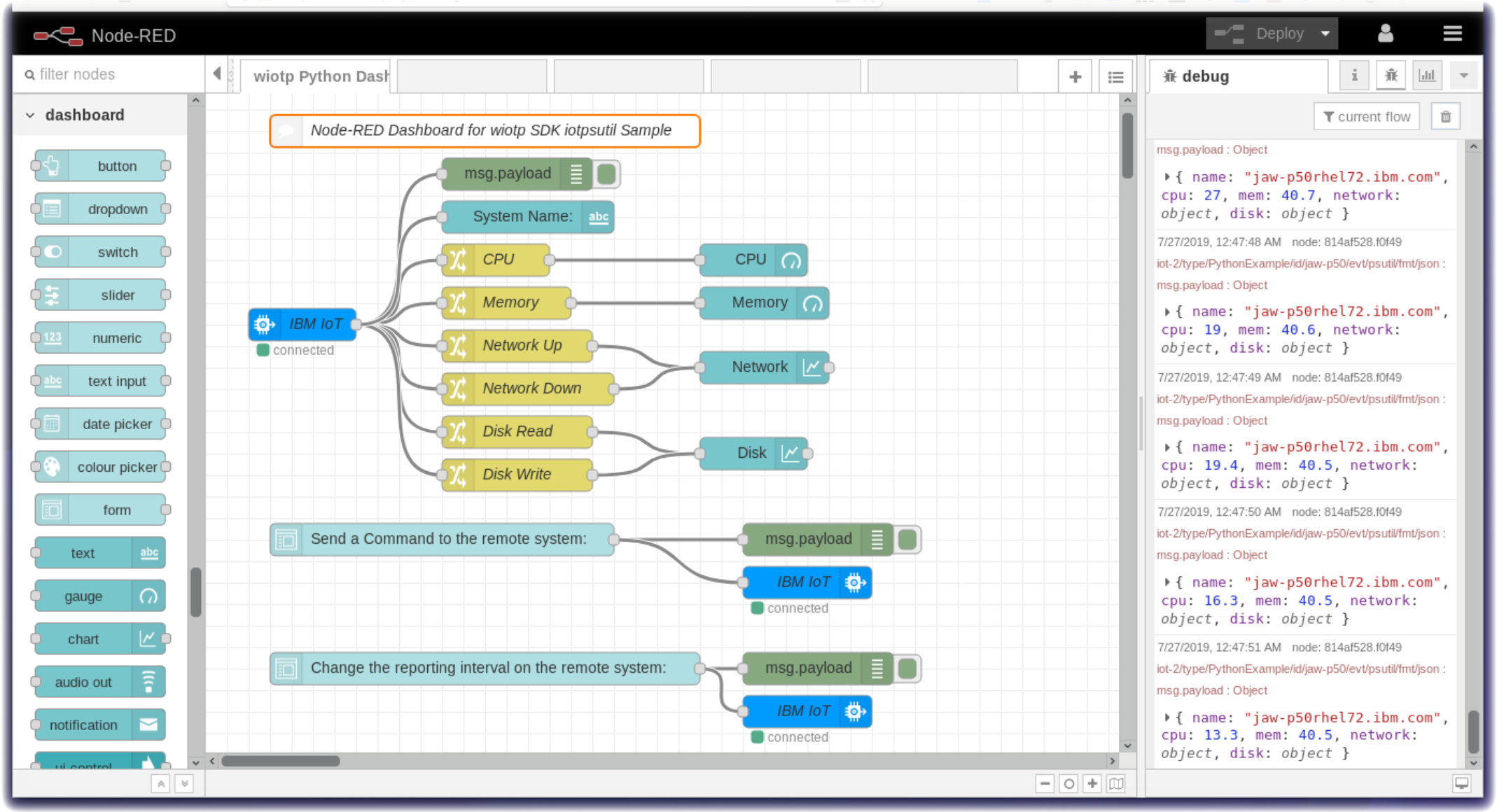
Environment Sensors and IOT for Solar Farm



Environment Sensors and IOT for Solar Farm

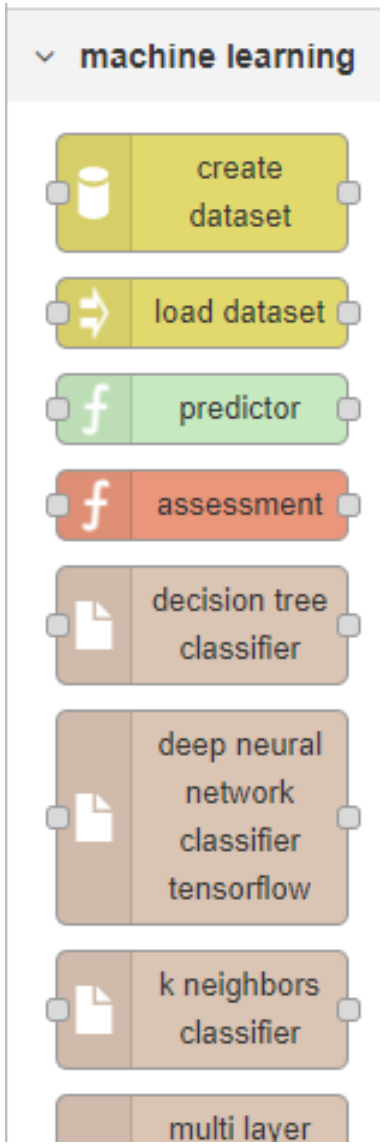
NODE-RED : Python

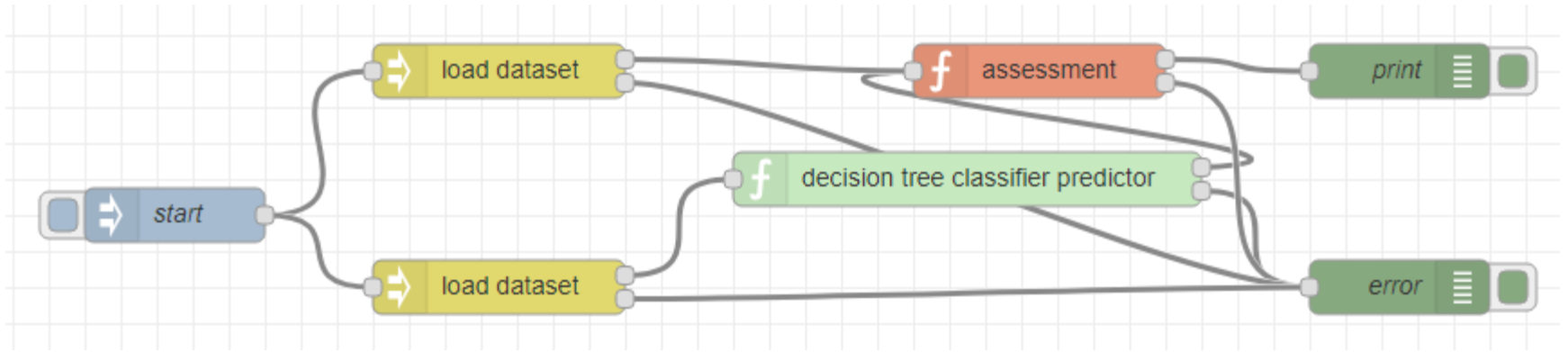
Environment Sensors and IOT for Solar Farm

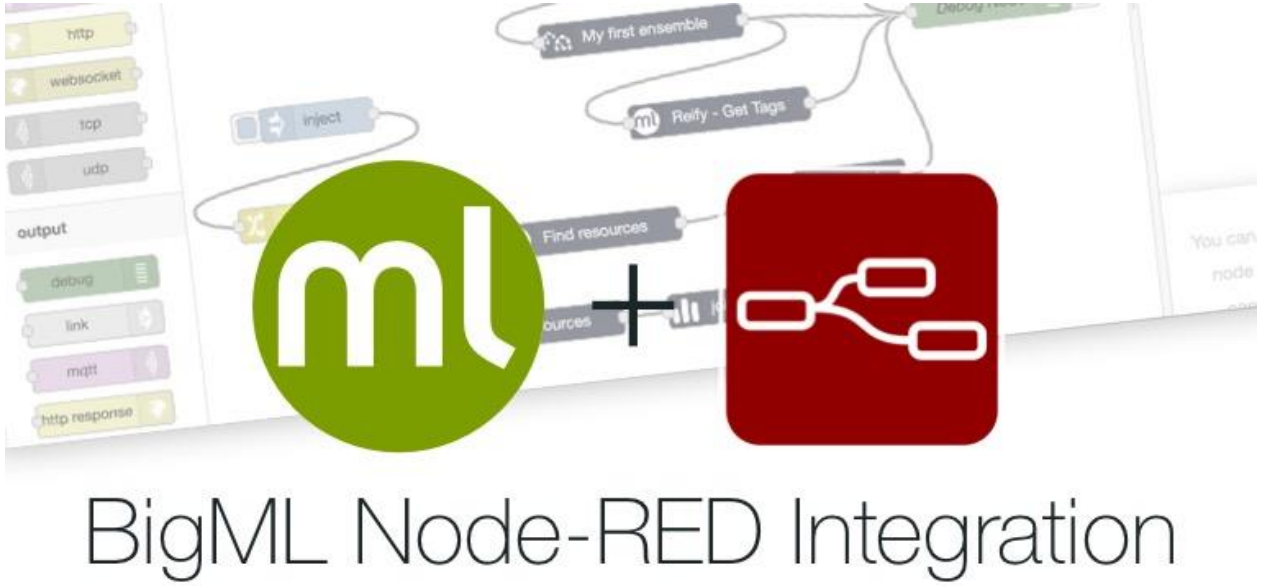


NODE-RED : ML

Environment Sensors and IOT for Solar Farm







Artificial Intelligence of Things : Architecture

Environment Sensors and IOT for Solar Farm