Master Thesis Presentation

Design and Implementation of Wireless Sensor Networks Using LoRaWAN, MQTT and Cloud Computing

Creating long-range, low-power sensor network for applications in smart cities, industrial automation and smart agriculture.

Supervisor 1: Prof. Dr.-Ing. Roman Obermaisser
Supervisor 2 : Dr.-Ing. Daniel Onwuchekwa

Vishu Sharma | Matr Nr: 1523731

Contents

Introduction

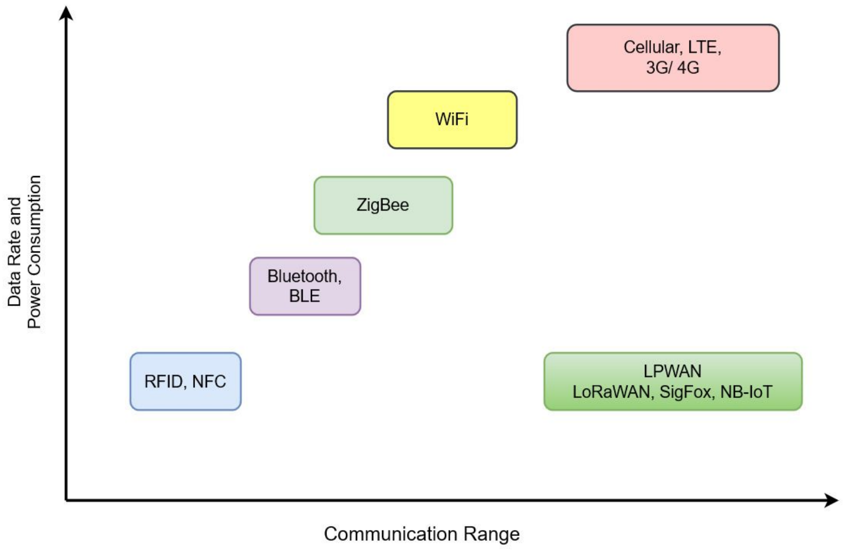
Motivation: To create the network layer of an IoT platform **Obodo** for application in smart cities. Small scale Wireless Sensor Networks (WSNs) were created as proof of concept for this IoT Platform.

Key requirements: Long range, low power, low cost sensor nodes
Key tasks: Selection of suitable technology stack - communication protocols, Network hardware, regulatory compliance, system reliability

Research: Thesis investigates the optimal system architecture of the WSNs based on LoRaWAN, MQTT and Cloud computing / on-premise servers.

Funding: Research was sponsored by **APS Tech Group GmbH** in form of network equipment, computing infrastructure, electronics and other lab resources.

Selection of technology stack - LoRaWAN



- Technology: Only LPWAN meets the requirement of long range and low power and suitable for sensor networks.
- Wifi and bluetooth offers high data transfer rate but at the cost of high power consumption and significantly low range.
- 3 main competing technologies in LPWAN space: Sigfox, NB-IoT, LoRaWAN

LoRaWAN was selected to be most appropriate communication technology:

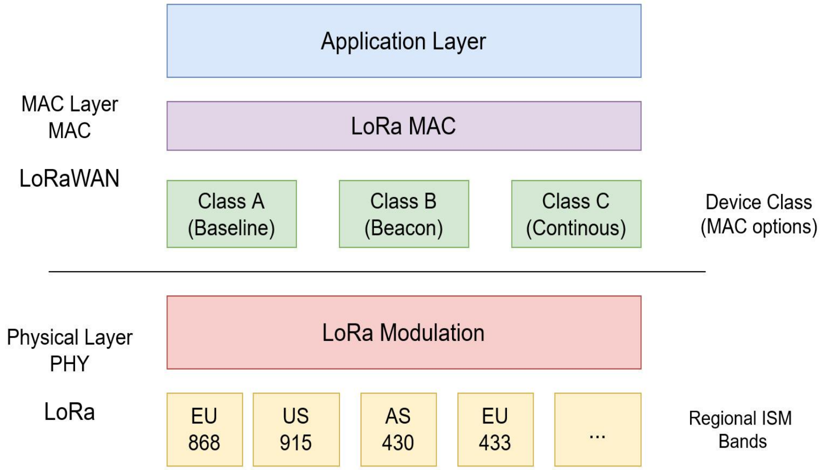
- Range and power consumption,
- Data rate and payload size
- *Cost and ease of network deployment*

LoRa and LoRaWAN Overview

LoRa - RF modulation based on Chirp Spread Spectrum

LoRaWAN - Protocol on top of LoRa Physical layer

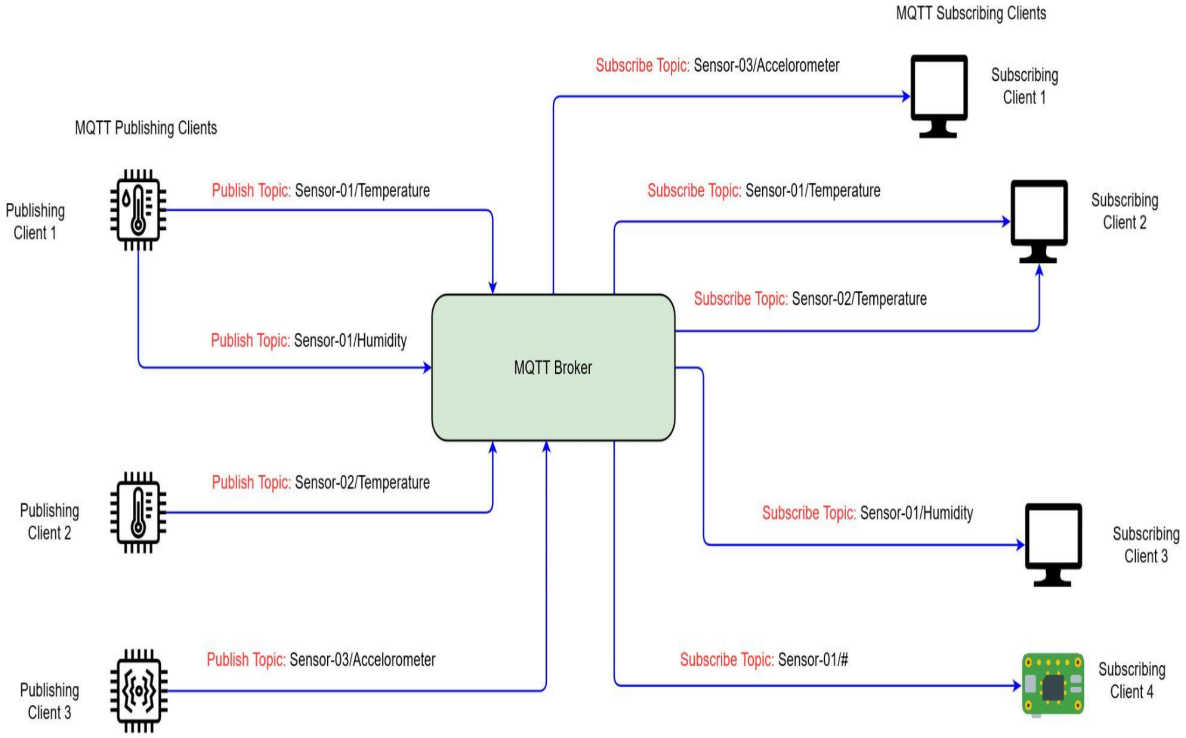
This is not based on TCP/IP stack.



Key Features

- Long Range wireless communication technology.
- Unlicensed Sub GHz ISM spectrum which varies from region to region.
- End devices run upto 10 years with 2- 10 Km range*
- Only suitable for sending small data packets (Max 243 bytes), at low bit-rate.
- LoRaWAN is an open standard (ITU-T Y.4480) maintained by LoRa Alliance.
- LoRa modulation is proprietary technology of Semtech Corporation

Selection of technology stack - MQTT



Key Features

- Lightweight pub-sub M2M message transport protocol.
- Application layer of TCP IP
- Decoupled relationship between message producers and message consumers
- Clients publish and subscribe to topics via broker.
- Works on range of devices- PCs, servers, SBCs, MCUs
- Various QoS
- Clients are easy to write, broker is difficult to implement.

Application areas for LPWAN based WSNs

Industrial automation:

- Temperature and humidity and indoor air quality monitoring in pharmaceutical and food processing industry
- Fluid level monitoring in tanks for industries like chemical, textile and petroleum

Smart cities:

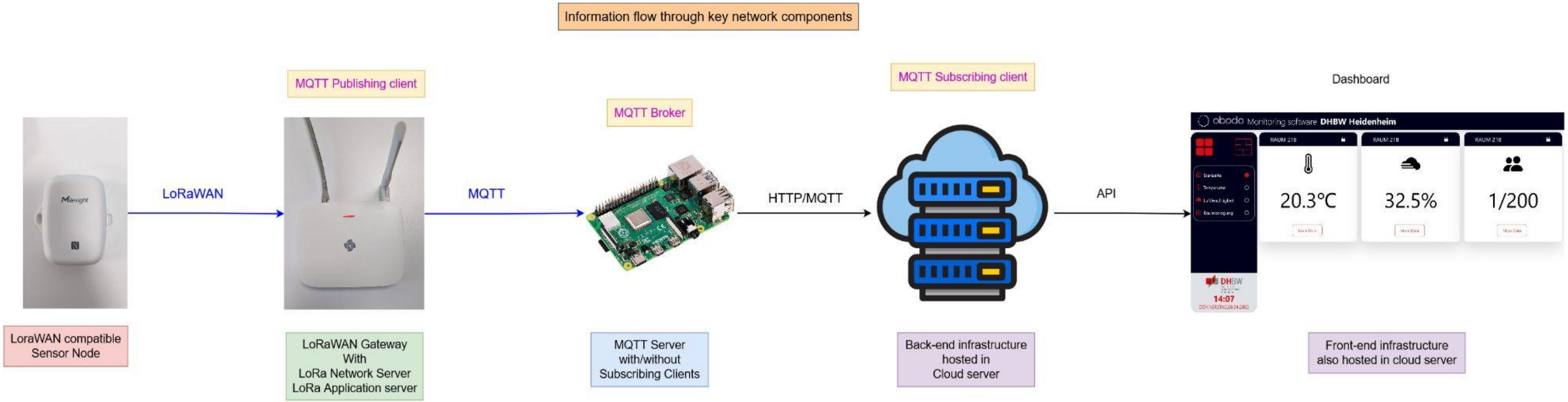
- Air quality index and particulate matter monitoring
- City waste management and garbage collection
- Parking occupancy by smart parking sensors
- Ambience monitoring in smart buildings

Smart agriculture:

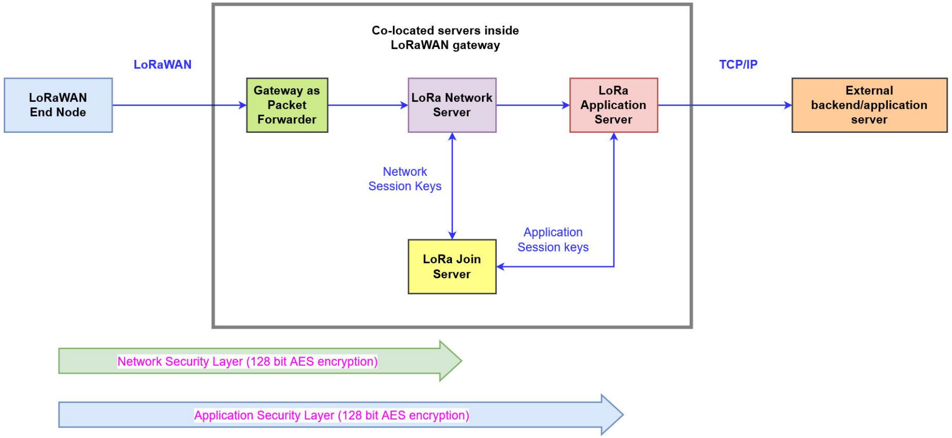
- Monitoring crop growing conditions like light intensity, soil moisture etc
- Toxic gas level monitoring in compost and livestock enclosures
- Outdoor weather stations, irrigation

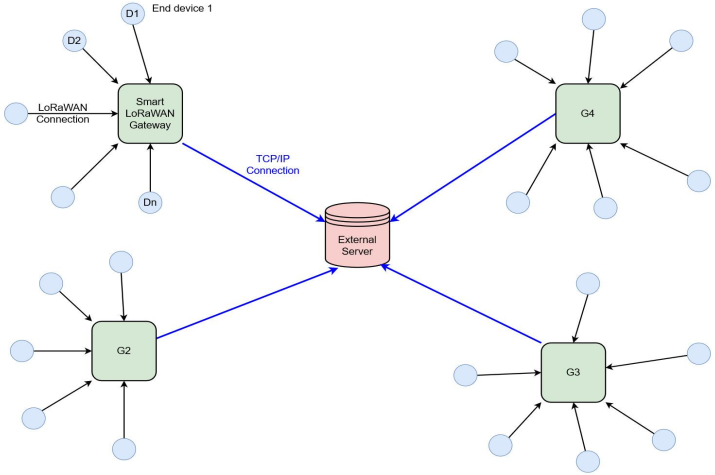
High level Information flow through Network Devices

Key Network devices - Sensor nodes, Gateways, Servers
Key communication protocols - LoRaWAN, MQTT, HTTP



LoRaWAN Network Topology

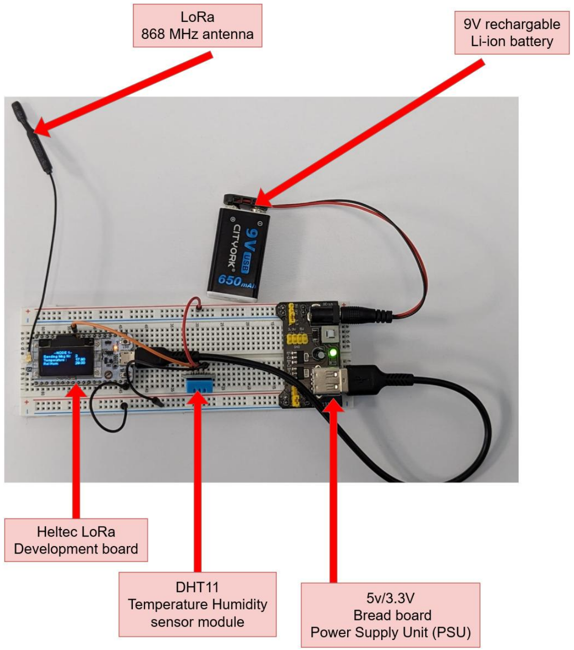


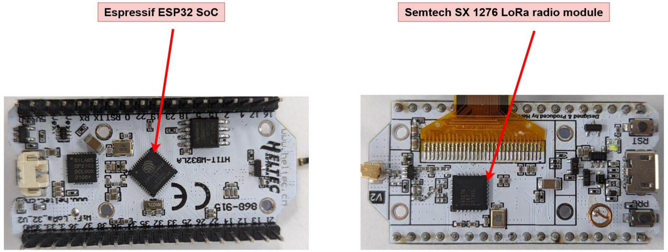


- LoRaWAN follows star of star network topology.
- Gateway acts a bridge between sensors and the internet / external servers.
- Key components in LoRaWAN network- End device, gateway, Join Server, LoRa Network Server, LoRa application server

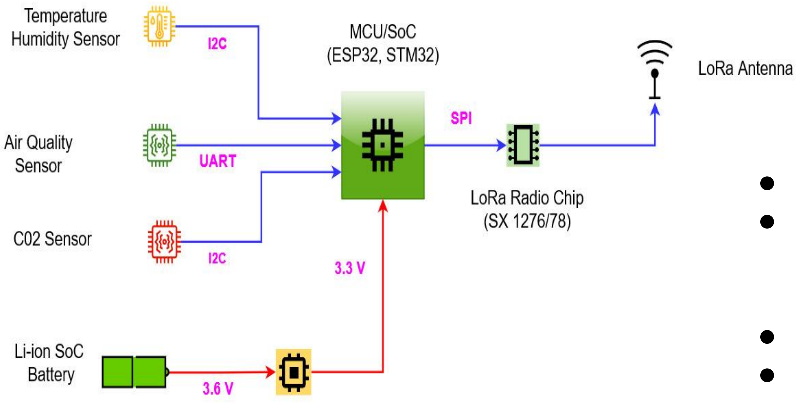
Sensor Nodes / End devices

Sensor node created with LoRa development board





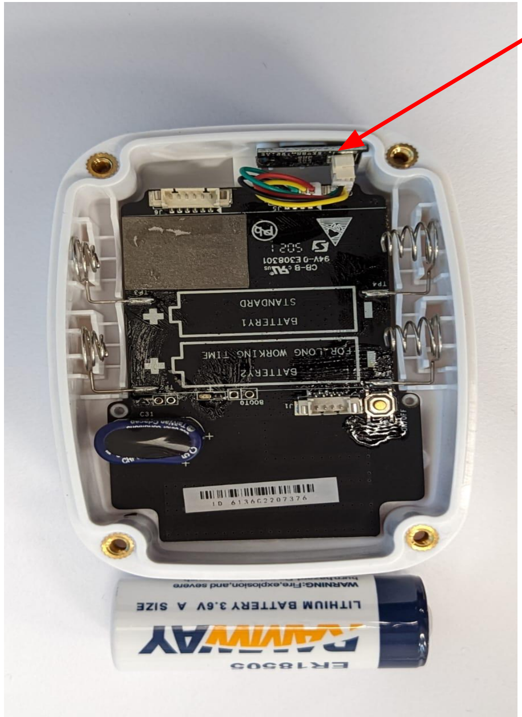
Internal architecture of LoRa sensor Node



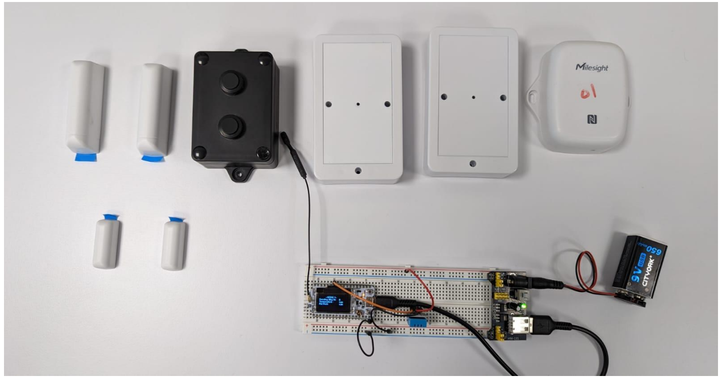
- System on Chip (SoC)
- LoRa Radio module (embedded/ stand alone)
- LoRa antenna
- Sensor Module
- Power supply with Li-ion batteries

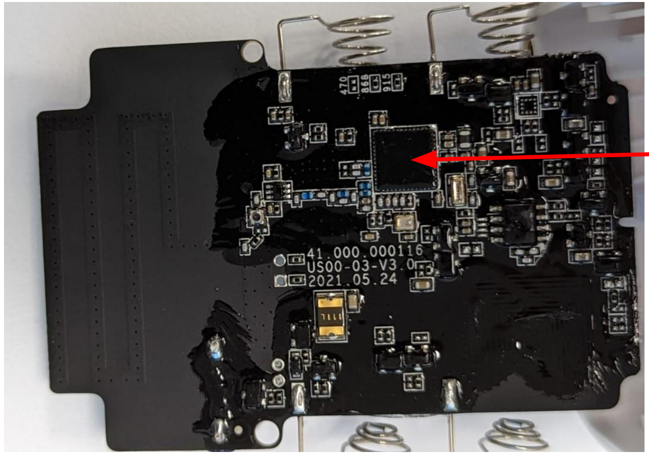
Commercial LoRaWAN compatible sensor nodes

Relative humidity and temperature sensor node from milesight



Sensor module connected via I2C

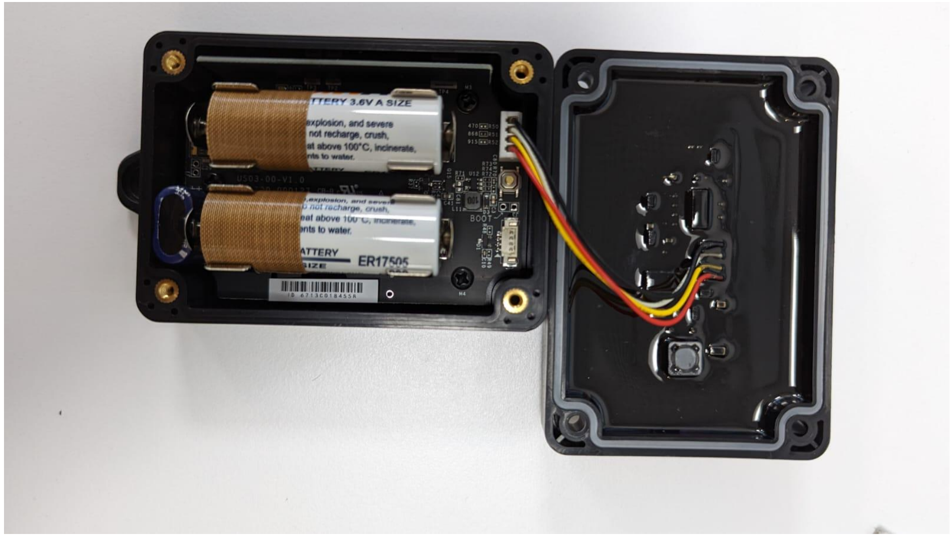


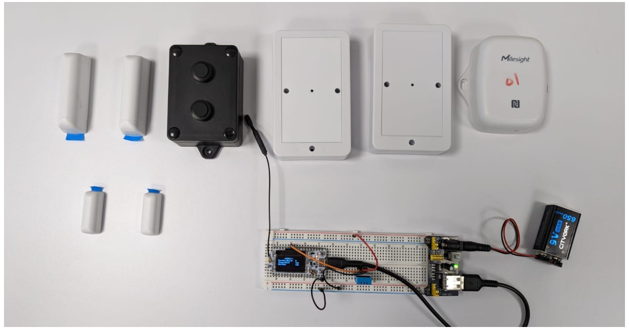


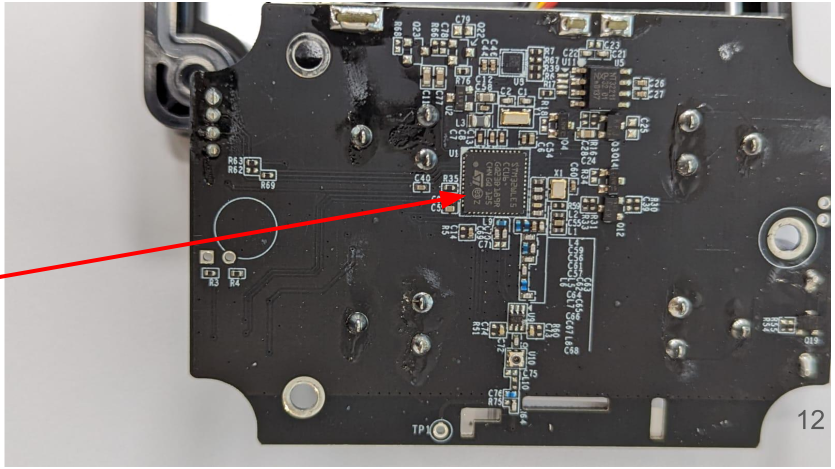
SoC and LoRa radio module in one chip

Commercial LoRaWAN compatible sensor nodes

Ultrasonic Distance Sensor from Milesight

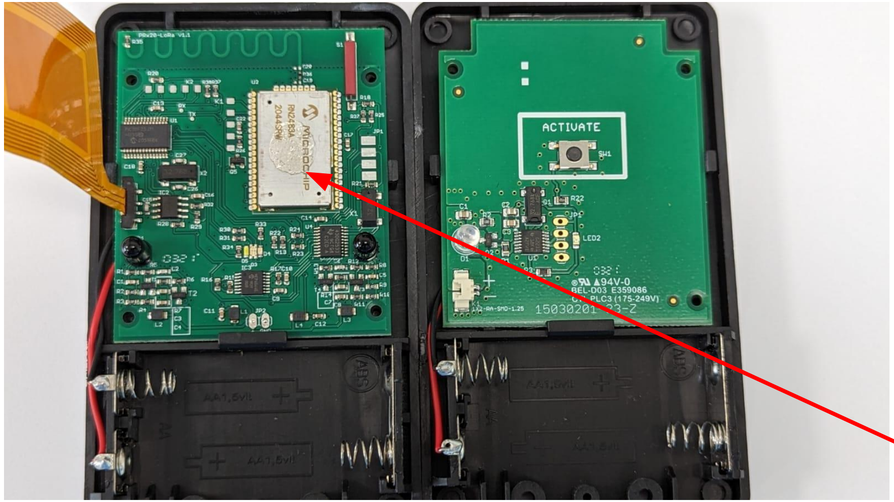


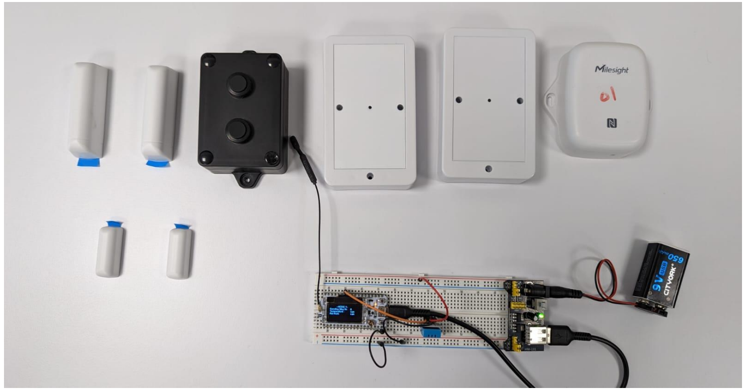




STM32 WLxx series chip

Commercial LoRaWAN compatible sensor nodes

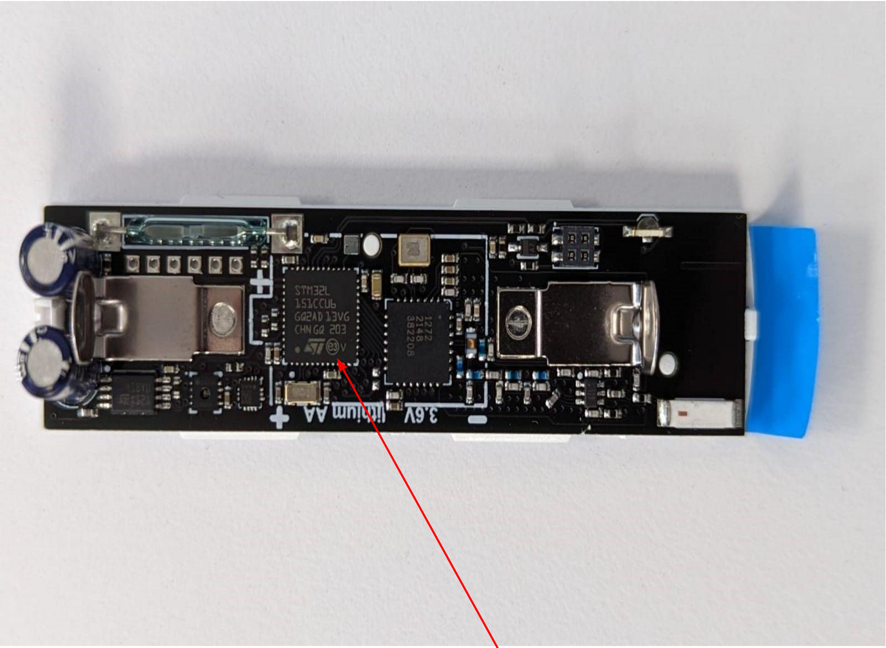


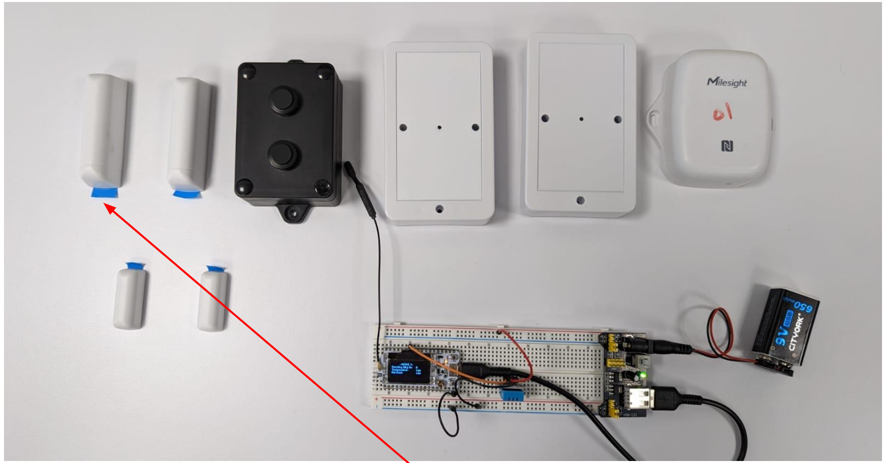


Microchip
RN2483 LoRa
stand alone
radio module

Infrared based
People Counter Sensor
from IM Buildings

Commercial LoRaWAN compatible sensor nodes



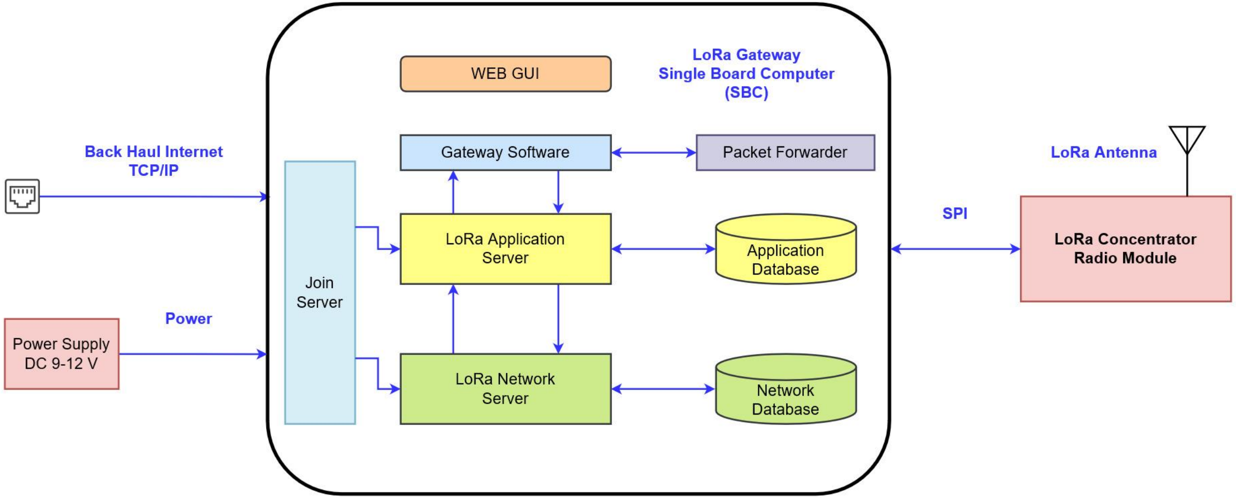


Magnetic contact switch door sensor by ELESYS

SoC and LoRa radio
module in one chip

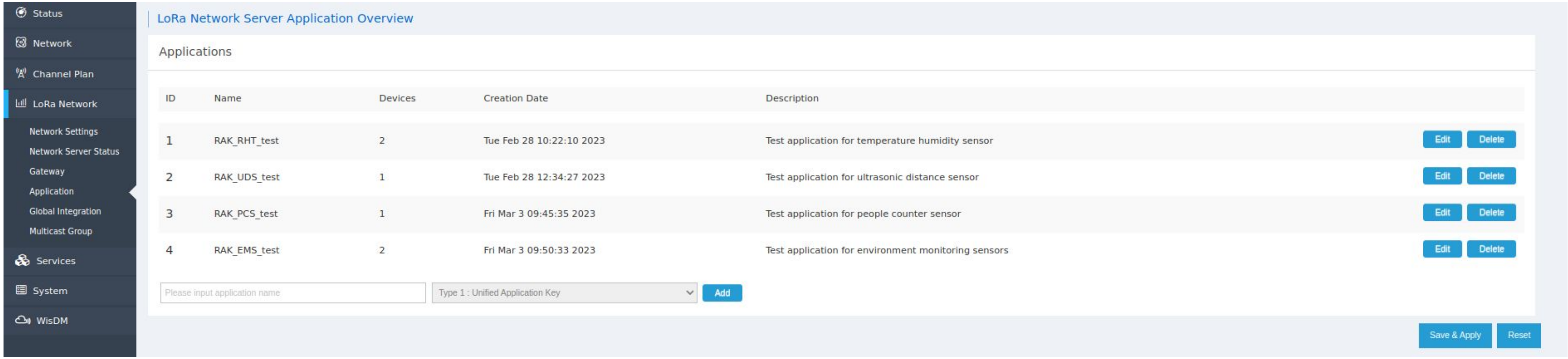
LoRaWAN Gateway internal Architecture

LoRaWAN Gateway (RAK wireless indoor 7268C)

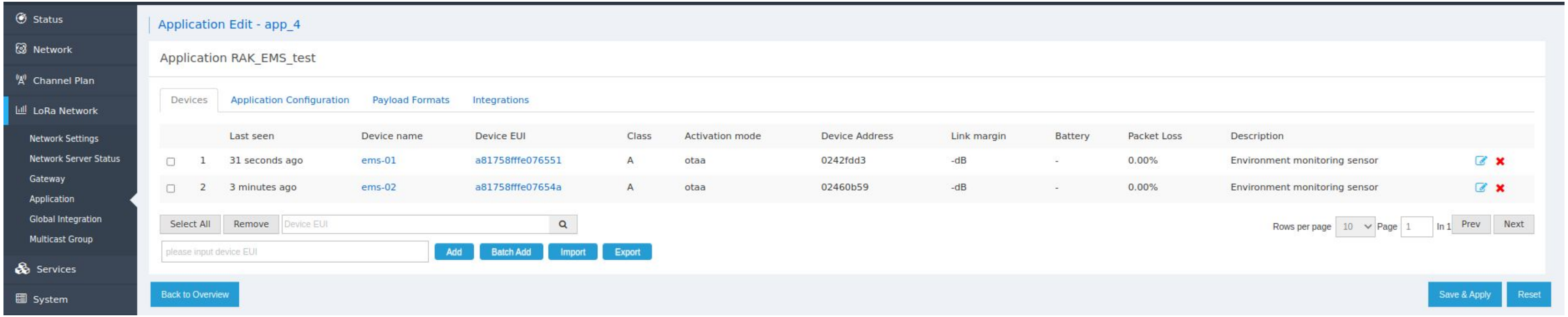


- Single Board Computer (Arm based multi-core processors)
- Multi-channel LoRa concentrator radio module/chip like iC880A or SX1302
- Usually 8,16, 64 channel gateway
- Chips capable of connecting to backhaul internet (WiFi/LTE Sim)
- Power Supply: DC adapter or Power over Ethernet (PoE).
- Separate Antennas for LoRa, WiFi and GPS
- Smart gateways have embedded LoRa Network Server and Application Server

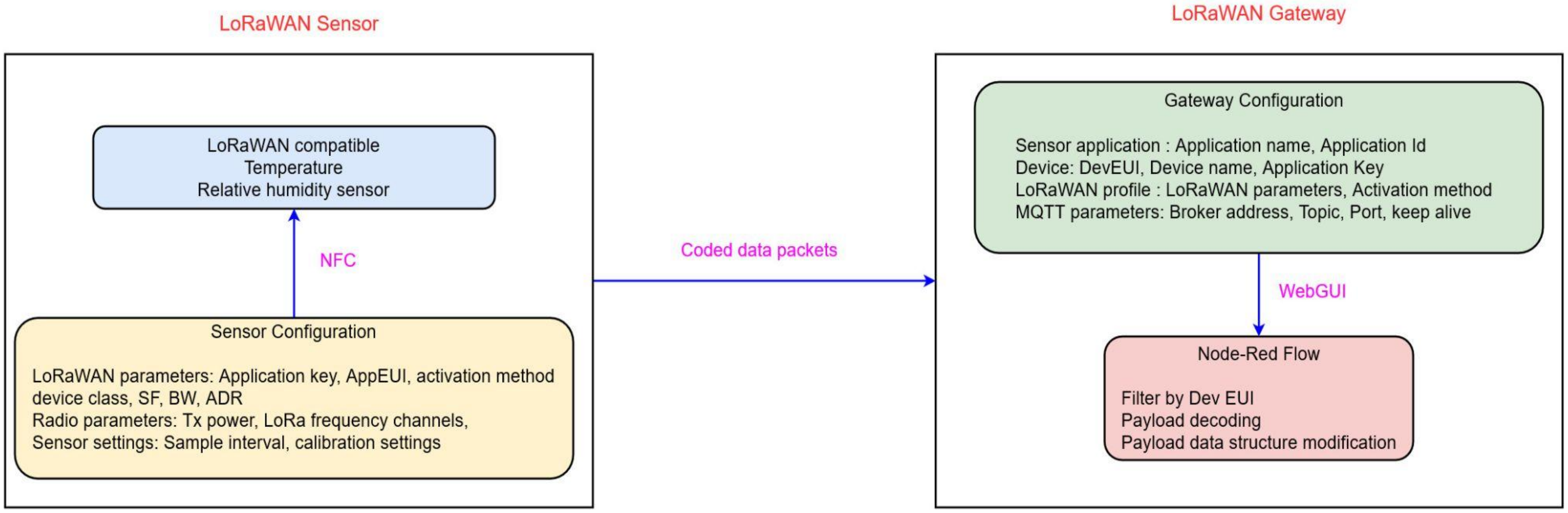
Web GUI of Gateway - Applications, Devices and LoRaWAN parameters



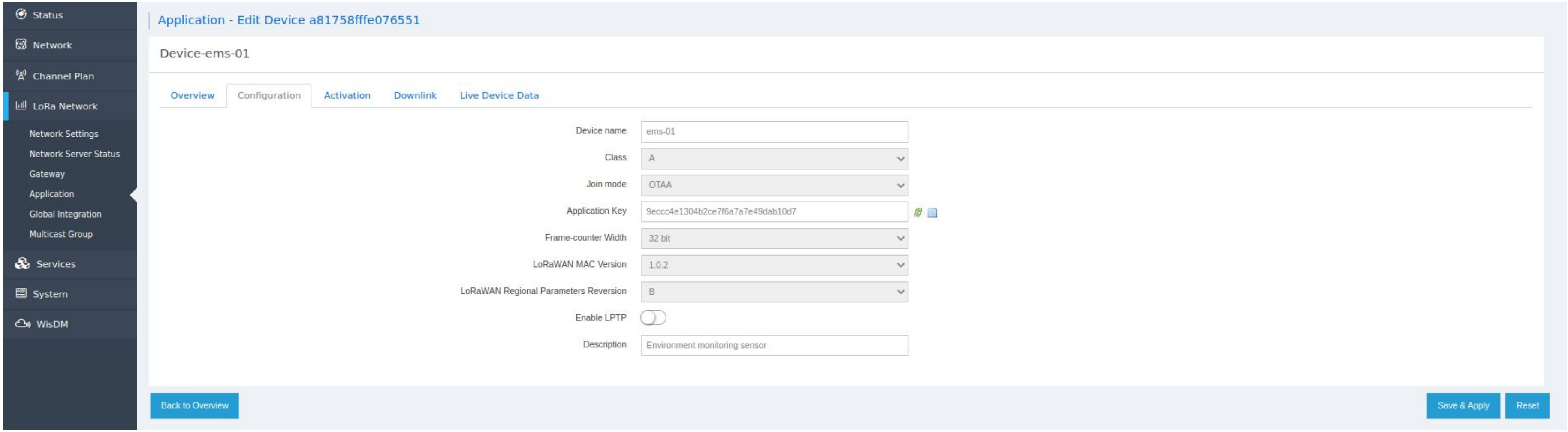
Device Management

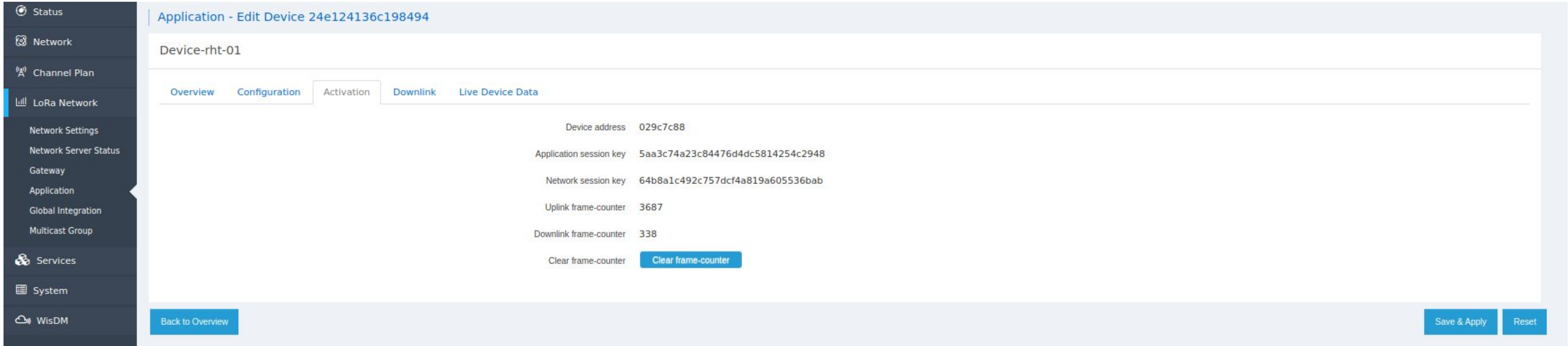


Configuring Commercial LoRaWAN Network devices.

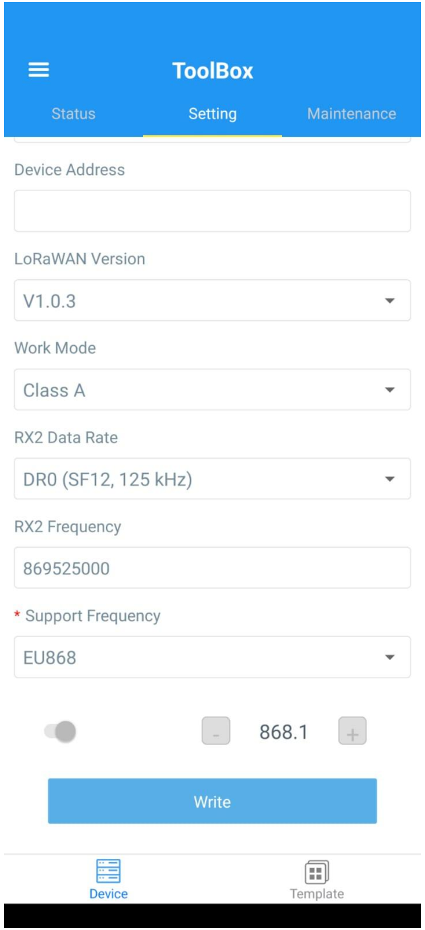


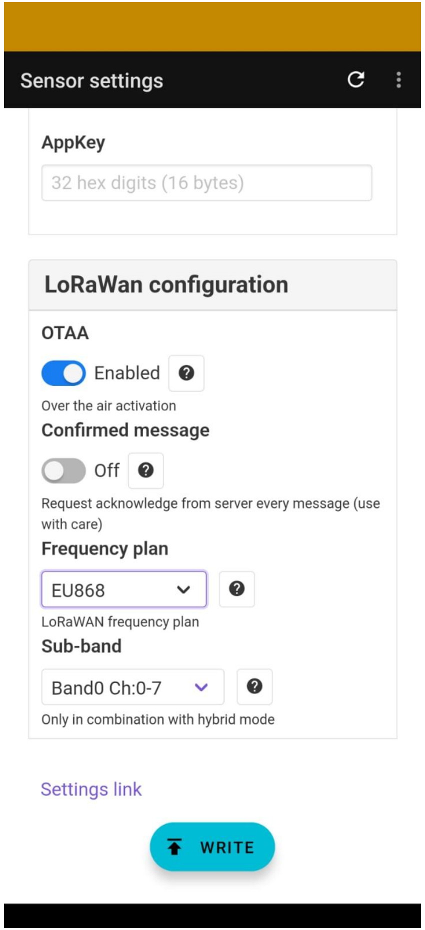
Web GUI of Gateway : Device configuration, security

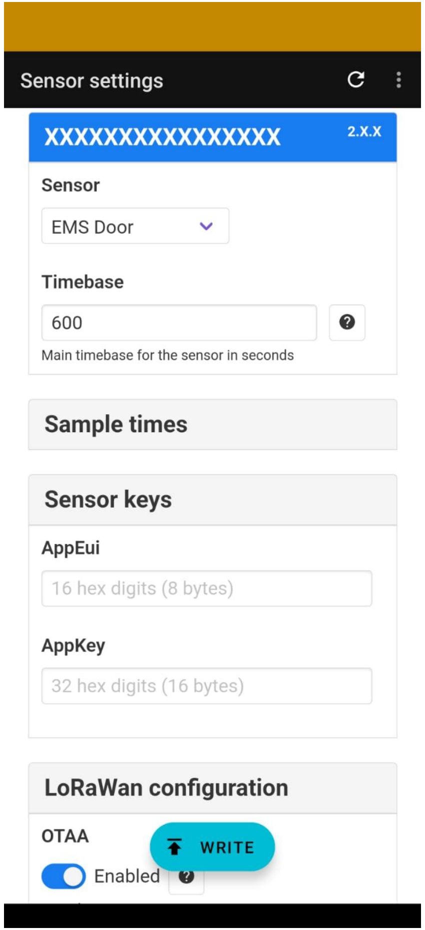




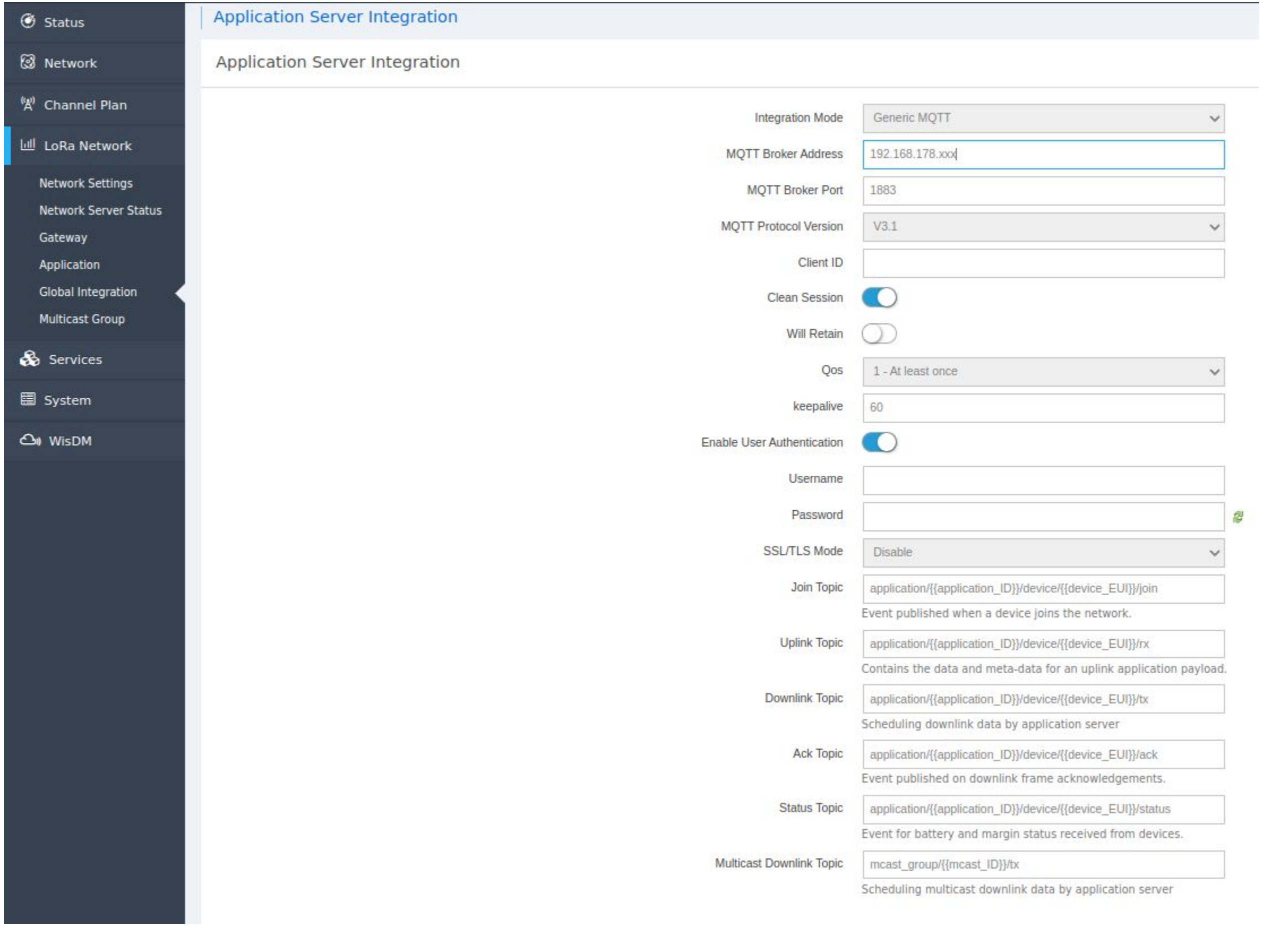
Sensor configuration application : Device configuration, security







Integration of Gateway with MQTT Server



Key Parameters:

Broker address
MQTT port
MQTT version
MQTT Topic

—-------------------

Security:

Username
Password
TLS certificate

—-----------------

Client Id
QoS
Acknowledgement
Keep Alive

MQTT Server and Clients

MQTT Client Devices used in this project ranged from PCs SBCs like RPi (RP3 and RP400) and a windows network server.

Mosquitto MQTT broker was configured on windows server and Raspberry Pi using CLI.

Gateway acted as a publishing client

PCs and another raspberry Pi was used as subscribing client to log the data.

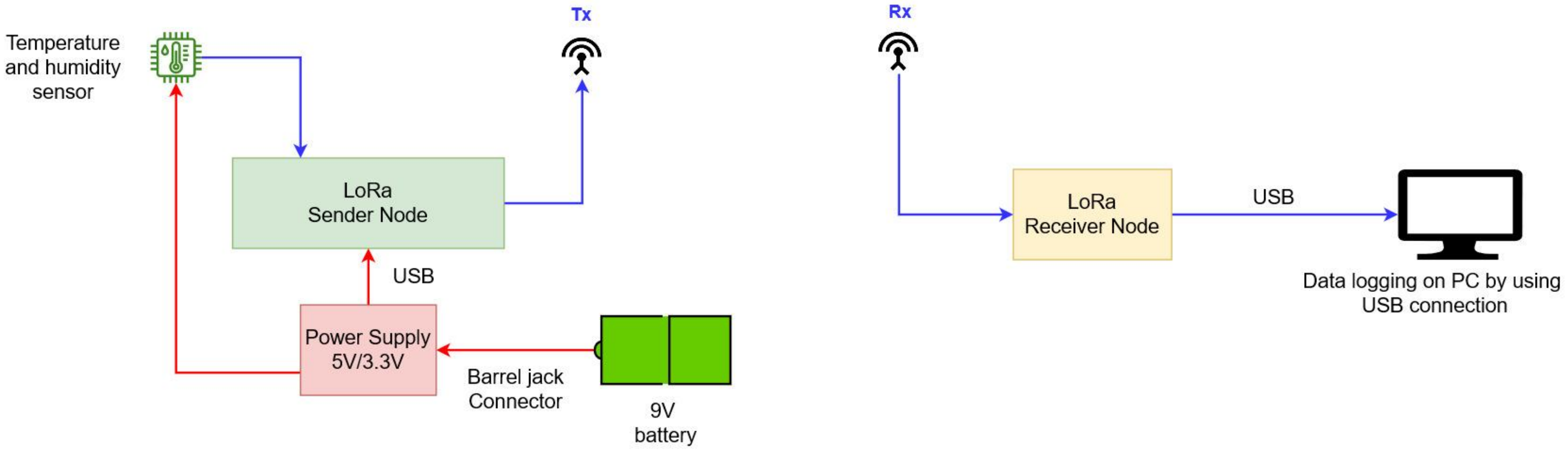
Node-Red flows and Python scripts were used subscribing clients and data logging.

MQTT Sample Client Script

```python
import paho.mqtt.client as paho
import time
import datetime

def Message_handler(client, userdata, msg):
    topic = msg.topic
    msg_decoded = msg.payload.decode("utf-8", "ignore")
    msg_JSON = topic + ":" + msg_decoded
    print(msg_decoded)

    #-----Logging messages --------


    #---------- Text File ----------

    with open('Filename_dd.mm.yyyy.txt', 'a', encoding = 'utf-8') as f:
        f.write(msg_decoded)
        f.write(",")
        f.write('\n')


# ------- MQTT Connection handling ----------

MQTT_broker = "192.xx.xx.xx"    # Enter the IP of your broker
MQTT_port = 1883
keep_alive = 60
MQTT_topic = "#"                # Enter the topics you want to subscribe
```

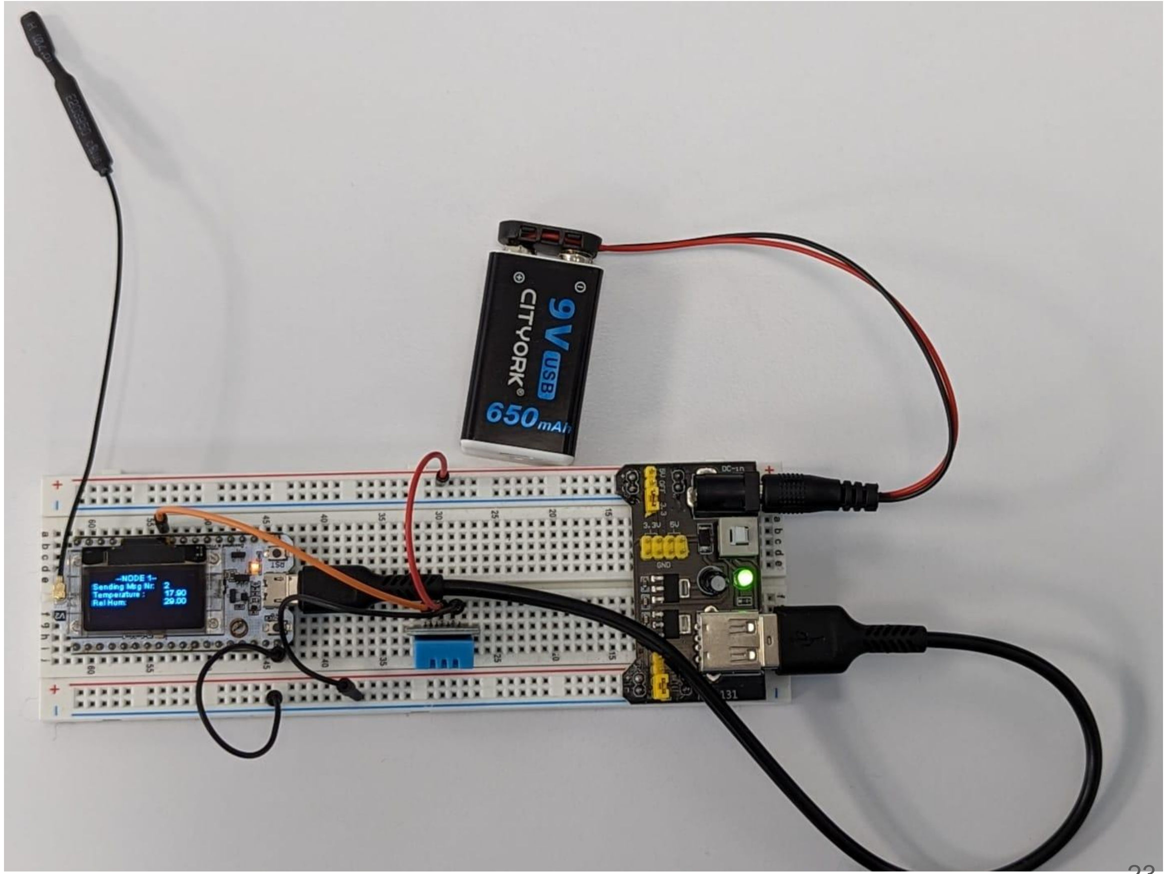
Initial experiments with LoRa PHY Layer

Schematic of System Setup for Node to Node LoRa Communication



Initial experiments with LoRa PHY Layer using development boards

Message_Number	Temp_Celsius	RH_Percent
1	27.3	45
2	27.3	46
3	27.3	46
4	27.3	45
5	27.2	45
6	27.2	45
7	27.2	45
8	27.2	45
9	27.1	45
10	27.1	45

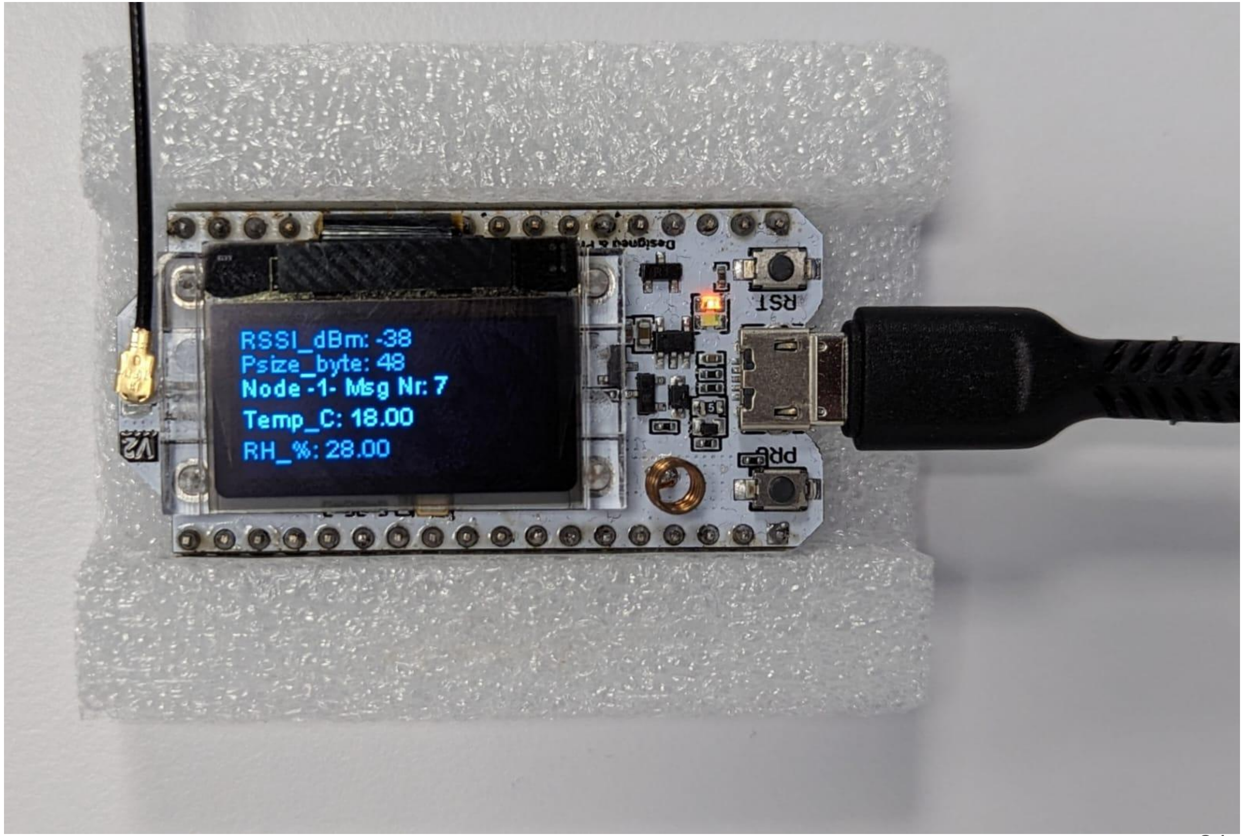


```
RSSI_dBm: -75
Psize_byte: 48
Node -1- Msg Nr: 1
Temp_C: 23.60
RH_%: 52.00
```

```
RSSI_dBm: -73
Psize_byte: 48
Node -1- Msg Nr: 2
Temp_C: 23.70
RH_%: 52.00
```

```
RSSI_dBm: -67
Psize_byte: 48
Node -1- Msg Nr: 3
Temp_C: 23.50
RH_%: 52.00
```

Initial experiments with LoRa PHY Layer



Software written for development boards

```
#include "heltec.h"        // from board manufactures for LoRa, OLED, ESP32
#include <DHT.h>           // From sensor manufacturer

#define Type DHT11         // Sensor type
#define BAND      868E6    //ISM band in EU is 868MHz for LoRa

// Pin assignment

int Pin_DHT11 = 23;     // Signal pin for DHT object attached to MCU

// variable assignment

unsigned int counter = 0;           // For message Number
String rssi = "RSSI --";            // Recieved signal strength
String packSize = "--";             // Data packet size
String packet ;                     // Data packet

int Node_num = 001  ;                // Sensor Node number
int baudrate = 115200;               // serial communication rate
float N1_Temp = 23;                  // Temperature in Celsius
float N1_RH = 55 ;                   // Relative Humidity in percent
int Time_pol = 29500;                // Sample Time for polling the sensor
bool label = true;                   // For labeling in serial communication

DHT DHT_Node_1(Pin_DHT11, DHT11);    // DHT object attached to Node_1


void setup()
{
   // OLED display initialization (Check Heltec documentation for more info)

  Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/);
  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);

  delay(2000);
  Heltec.display->clear();

  Heltec.display->drawString(0, 0, "Starting Node Nr: ");
  Heltec.display->drawString(90, 0, String(Node_num));
  Heltec.display->display();
  delay(3000);

  // Starting communication with sensor DHT11 on Node_1

  DHT_Node_1.begin();
  //Serial.begin(baudrate);  Do not use this statement!!!
}

void loop()
{

  // Reading Sensors:
  N1_Temp = DHT_Node_1.readTemperature();
  N1_RH = DHT_Node_1.readHumidity();
```

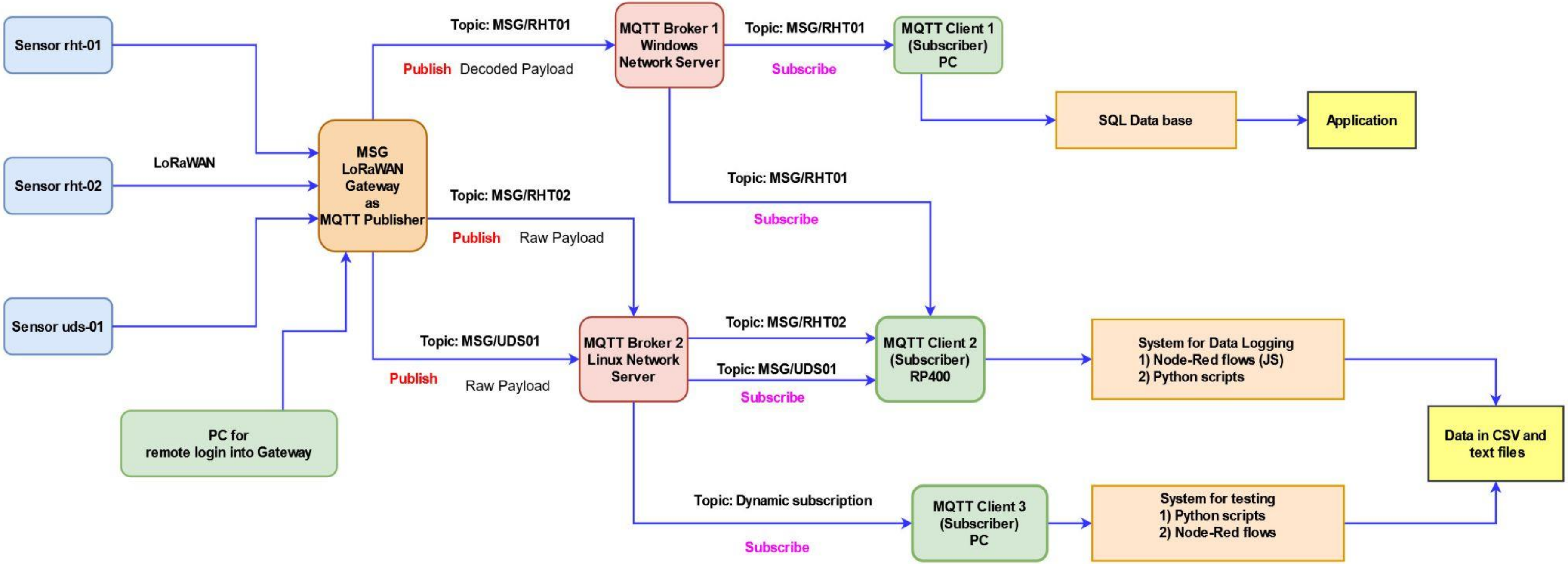
```
void loop()
{

  // Reading Sensors:
  N1_Temp = DHT_Node_1.readTemperature();
  N1_RH = DHT_Node_1.readHumidity();


  // Display info at sensor Node OLED display:

  Heltec.display->clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);

  Heltec.display->drawString(30,0, "--NODE 1-- ");
  Heltec.display->drawString(0, 10, "Sending Msg Nr: ");
  Heltec.display->drawString(90, 10, String(counter));
  Heltec.display->drawString(0, 20, "Temperature : ");
  Heltec.display->drawString(90, 20, String(N1_Temp));
  Heltec.display->drawString(0, 30, "Rel Hum: ");
  Heltec.display->drawString(90, 30, String(N1_RH));
  Heltec.display->display();


  // Sending data packet via LoRa modulated radio:

 /* Info from manufacturer of board:

 * LoRa.setTxPower(txPower,RFOUT_pin);
 * txPower -- 0 ~ 20  // 14 dBm is power efficient
 * RFOUT_pin could be RF_PACONFIG_PASELECT_PABOOST or RF_PACONFIG_PASELECT_RFO
 *   - RF_PACONFIG_PASELECT_PABOOST -- LoRa single output via PABOOST, maximum output 20dBm
 *   - RF_PACONFIG_PASELECT_RFO     -- LoRa single output via RFO_HF / RFO_LF, maximum output 14dBm
*/

  LoRa.beginPacket();

  LoRa.setTxPower(14,RF_PACONFIG_PASELECT_PABOOST); // Transmitting at 14dBm
  // This data packet will be displayed on the LoRa reciever

  LoRa.print("Node -1- Msg Nr: ");
  LoRa.println(counter);
  LoRa.print("Temp_C: ");
  LoRa.println(N1_Temp);
  LoRa.print("RH_%: ");
  LoRa.println(N1_RH);

  LoRa.endPacket();


  // Polling every 30 seconds

  digitalWrite(LED, HIGH);           // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for 2 second
  digitalWrite(LED, LOW);            // turn the LED off by making the voltage LOW
  delay(Time_pol);

  counter++;


  // Printing and Data logging via serial communication:
```

```
#include "heltec.h"

#define BAND    868E6      //ISM band in EU is 868MHz for LoRa
String rssi = "RSSI --";   // signal strength for reciever
String packSize = "--";    // Data size of packet in bytes
String packet ;            // Actual LoRa packet


// For Display on OLED screen on board

void LoRa_OLED(){
  Heltec.display->clear();

  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);
  Heltec.display->drawString(0, 0, rssi);
  Heltec.display->drawString(0 , 10 , "Packet size " + packSize + " bytes");
  Heltec.display->drawString(0 , 20, packet);

  Heltec.display->display();
}

// For reading and calculating packet size

void Packet_reading(int packetSize) {
  packet = "";
  packSize = String(packetSize,DEC);   // Packet size in decimal

  for (int i = 0; i < packetSize; i++) {
    packet += (char) LoRa.read();
    }
  rssi = "RSSI-- " + String(LoRa.packetRssi(), DEC) ;  // Gives the packet size
  Serial.println(packet);
  LoRa_OLED();
}


// Initial setup

void setup() {

  Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/);

  Heltec.display->init();
  Heltec.display->flipScreenVertically();
  Heltec.display->setFont(ArialMT_Plain_10);

  delay(2000);
  Heltec.display->clear();
  Heltec.display->drawString(0, 0, "Wait for incoming data...");
  Heltec.display->display();
  delay(2000);
  //LoRa.onReceive(cbk);
  LoRa.receive();
}
```

Limitations of initial tests with development boards

- Not suitable for commercial applications - IP67 enclosure, CE certification, long development time.
- Security: No encryption was implemented in the nodes.
- Not possible to create network topology without LoRaWAN protocol.
- High power consumption - 650 mAh battery lasted about only a day
- Low reliability - Packet drop rate is high at distances around 1 Km.
- Only suitable for lab tests, sensor integration and developing new sensor nodes.

Network Architecture with Local servers



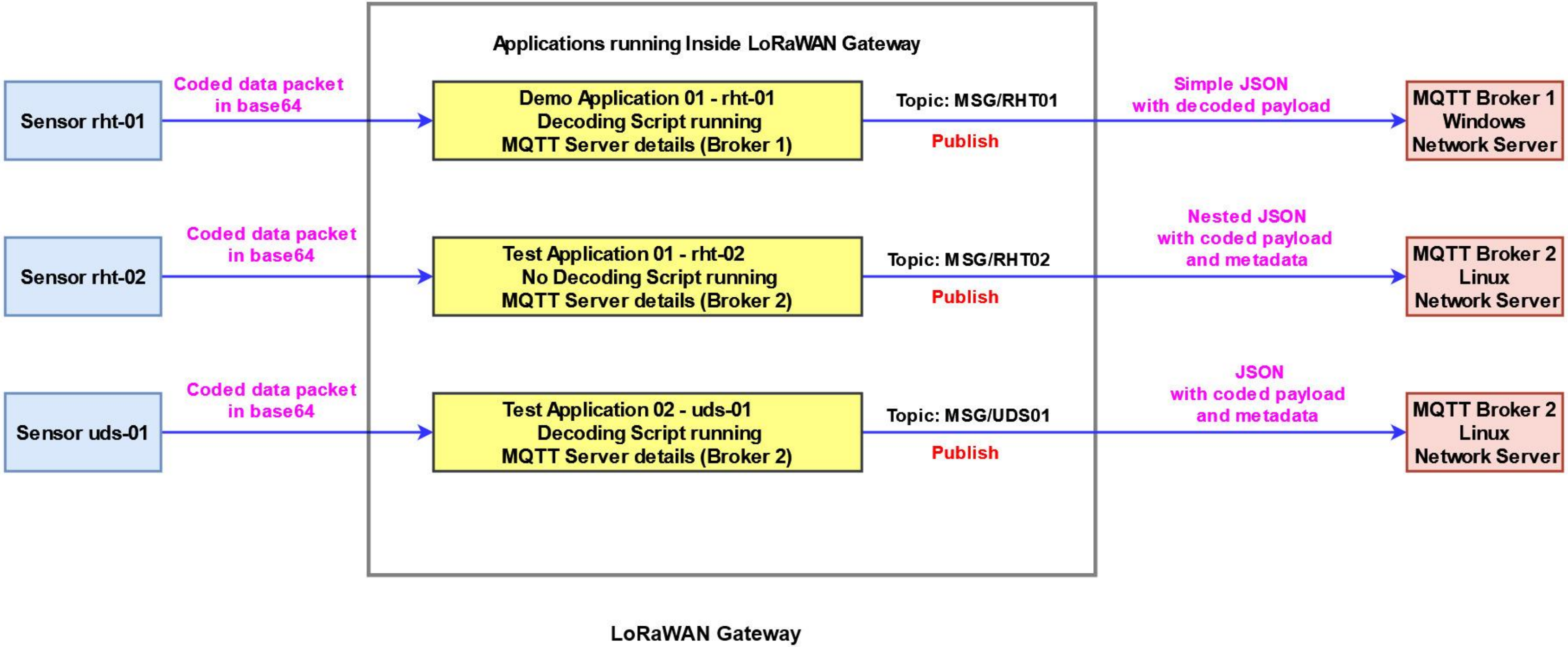
Applications created for the sensor nodes in LoRaWAN Gateway

Sensor Name	Description	Payload Information	Application Information
rht-01	Temperature and relative humidity sensor Manufacturer: Milesight	LoRaWAN data packet decoded in the gateway then transmitted to MQTT server as a JSON message	Demo Applitacion 01 Topic: MSG/RHT01 MQTT Server: MQTT Server 1 (Windows)
rht-02	Temperature and relative humidity sensor Manufacturer: Milesight	Raw data packets transmitted to the MQTT server without doing any modification in gateway	Test Applitacion 02 Topic: MSG/RHT02 MQTT Server: MQTT Server 2 (Linux)
uds-01	Ultrasonic distance sensor Manufacturer: Milesight	Raw data packets transmitted to the MQTT server without doing any modification in gateway	Test Applitacion 03 Topic: MSG/UDS01 MQTT Server: MQTT Server 2 (Linux)

Temperature Humidity Sensor

Ultrasonic Distance Sensor

Applications in Gateway



Sample JSON message from Gateway to MQTT Server

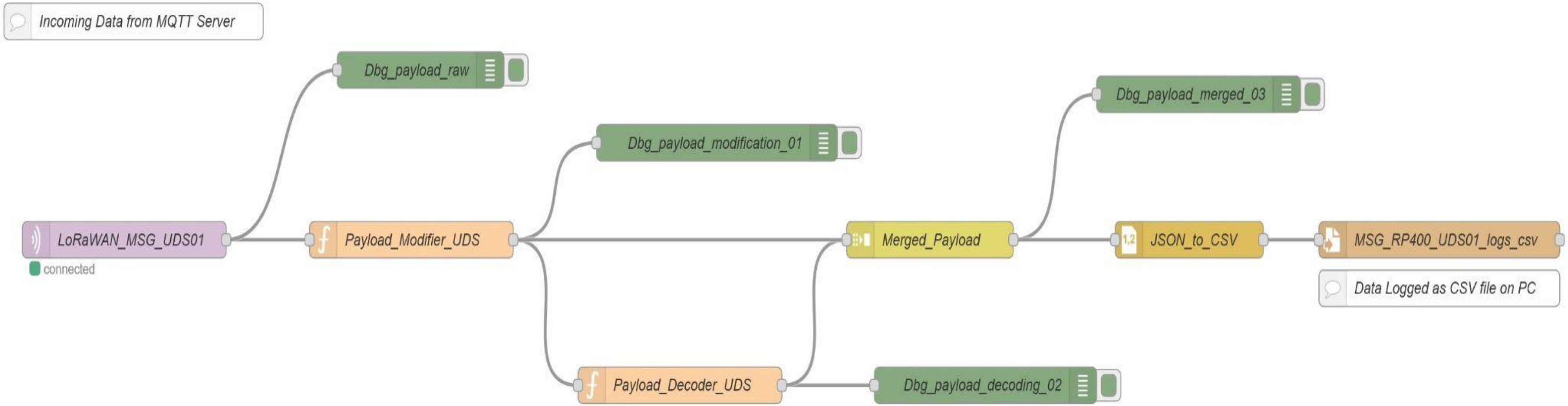

```json
{
    "topic": "MSG/UDS01",
    "payload": {
        "applicationID": "16",
        "applicationName": "Raspberry_Pi_400_UDS",
        "data": "AXVjA4J+CgQAAA==",
        "devEUI": "24e124713c018455",
        "deviceName": "uds-01",
        "fCnt": 2007,
        "fPort": 85,
        "rxInfo": [
            {
                "altitude": 0,
                "latitude": 0,
                "loRaSNR": 13.5,
                "longitude": 0,
                "mac": "24e124fffef57bd0",
                "name": "Local Gateway",
                "rssi": -55,
                "time": "2023-02-06T12:56:18.062038Z"
            }
        ],
        "time": "2023-02-06T12:56:18.062038Z",
        "txInfo": {
            "adr": true,
            "codeRate": "4/5",
            "dataRate": {
                "bandwidth": 125,
                "modulation": "LORA",
                "spreadFactor": 7
            },
            "frequency": 868300000
        }
    },
    "qos": 0,
    "retain": false,
    "_msgid": "a64d9e13d55be44a"
}
```


Raw Sensor payload and metadata from gateway to MQTT Server

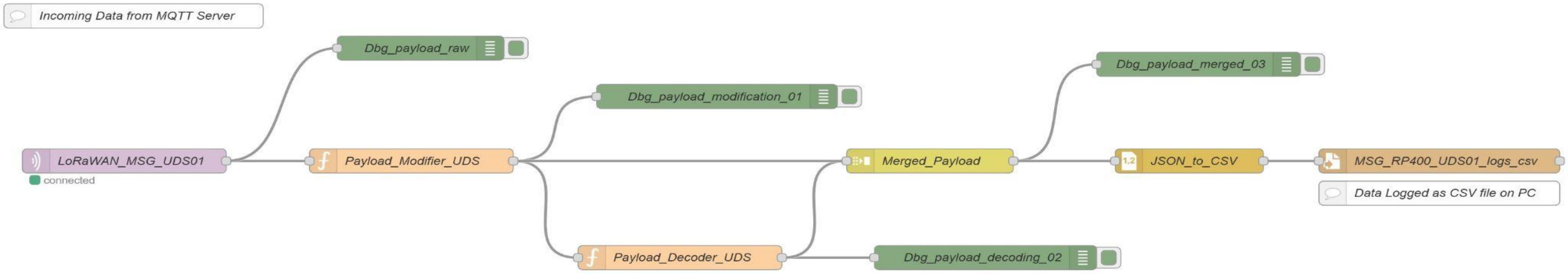

```
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4KNAQQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":3,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4KeBgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":4,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4KjBgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":5,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4KfBgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":6,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4J1CgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":7,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4J2CgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":8,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4J3CgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":9,"fPort":85,"rxInfo":[{"altitude":0,"
{"applicationID":"16","applicationName":"Raspberry_Pi_400_UDS","data":"AXVkA4J3CgQAAA==","devEUI":"24e124713c018455","deviceName":"uds-01","fCnt":10,"fPort":85,"rxInfo":[{"altitude":0,
```


Sensor payload stored as text files using Python Scripts on MQTT Client

Node-Red Flow for data logging and decoding running in subscribing client

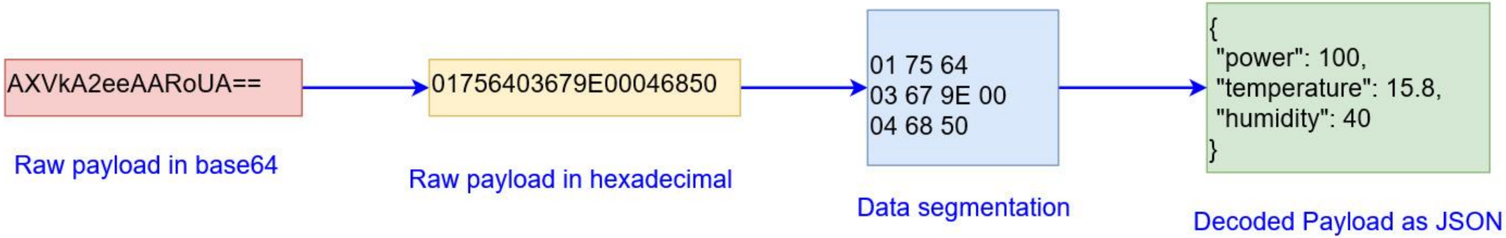


Node-Red Flow for data logging and decoding running in subscribing client



1	Topic	timestamp	devEUI	deviceName	Data_raw	battery	distance	position	gatewayEUI	rssi	loRaSNR	frequency
2	MSG/UDS01	2023-02-06T08:50:18.111428Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-56	10	868500000
3	MSG/UDS01	2023-02-06T08:52:18.109342Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-58	13.5	868100000
4	MSG/UDS01	2023-02-06T08:52:18.109342Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-58	13.5	868100000
5	MSG/UDS01	2023-02-06T08:54:18.116412Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1886	normal	24e124fffef57bd0	-59	13.5	868100000
6	MSG/UDS01	2023-02-06T08:54:18.116412Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1728	normal	24e124fffef57bd0	-59	13.5	868100000
7	MSG/UDS01	2023-02-06T08:56:18.11149Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1728	normal	24e124fffef57bd0	-59	12	868500000
8	MSG/UDS01	2023-02-06T08:56:18.11149Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-59	12	868500000
9	MSG/UDS01	2023-02-06T08:58:18.115359Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-62	14	868100000
10	MSG/UDS01	2023-02-06T08:58:18.115359Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-62	14	868100000

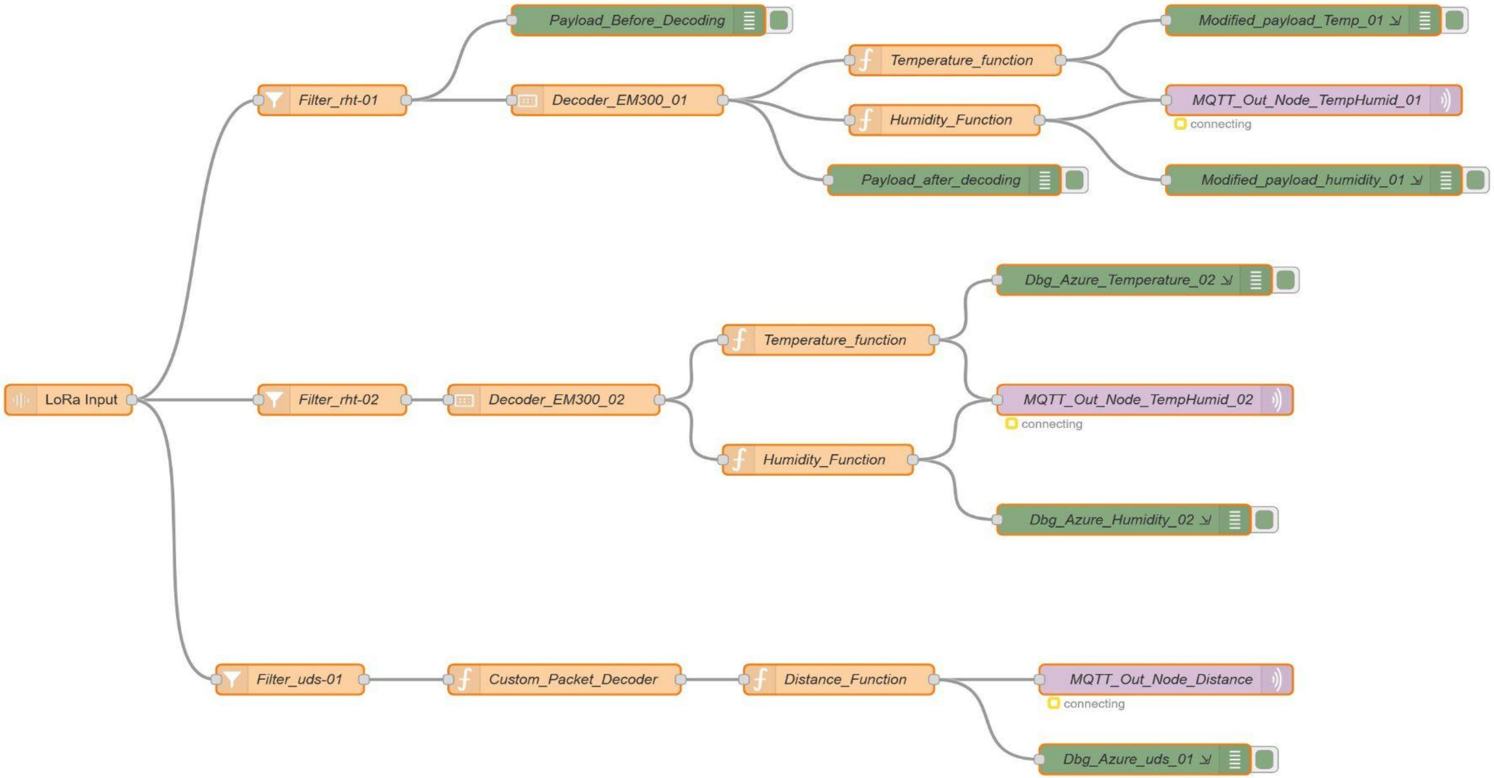
Sensor Payload decoding logic
(With an example of Relative Humidity-Temperature sensor)



Payload decoding logic for Milesight's EM310 Temperature humidity sensor

Channel Id	Channel type	Channel value	Value	Description	Unit
01	75	64	100	Battery level	%
03	67	9E 00	15.8	Temperature	°C
04	68	50	40	Relative Humidity	%

Node-Red Flow for data logging and decoding running in publishing client (Mile Sight UG65 indoor gateway)



Sensor payload decoding and data structure modification modification

Message Payload modification in Gateway using Node-RED

Raw Payload

12/19/2022, 1:49:06 PM node: Payload_Before_Decoding
msg : Object
▾object
▾dr: object
bandwidth: 125
modulation: "LORA"
spreadFactor: 7
adr: true
freq: 868100000
snr: 14.2
rssi: -43
time: "2022-12-19T12:49:05.894622Z"
fcnt: 753
fport: 85
deveui: "24e124136c198494"
payload: "AXVkA2eeAARoUA=="
_msgid: "24216bb7.08f644"

→ Decoding Script

Decoded Payload

12/19/2022, 1:49:06 PM node: Payload_after_decoding
msg : Object
▾object
▾dr: object
bandwidth: 125
modulation: "LORA"
spreadFactor: 7
adr: true
freq: 868100000
snr: 14.2
rssi: -43
time: "2022-12-19T12:49:05.894622Z"
fcnt: 753
fport: 85
deveui: "24e124136c198494"
▾payload: object
power: 100
temperature: 15.8
humidity: 40
_msgid: "24216bb7.08f644"

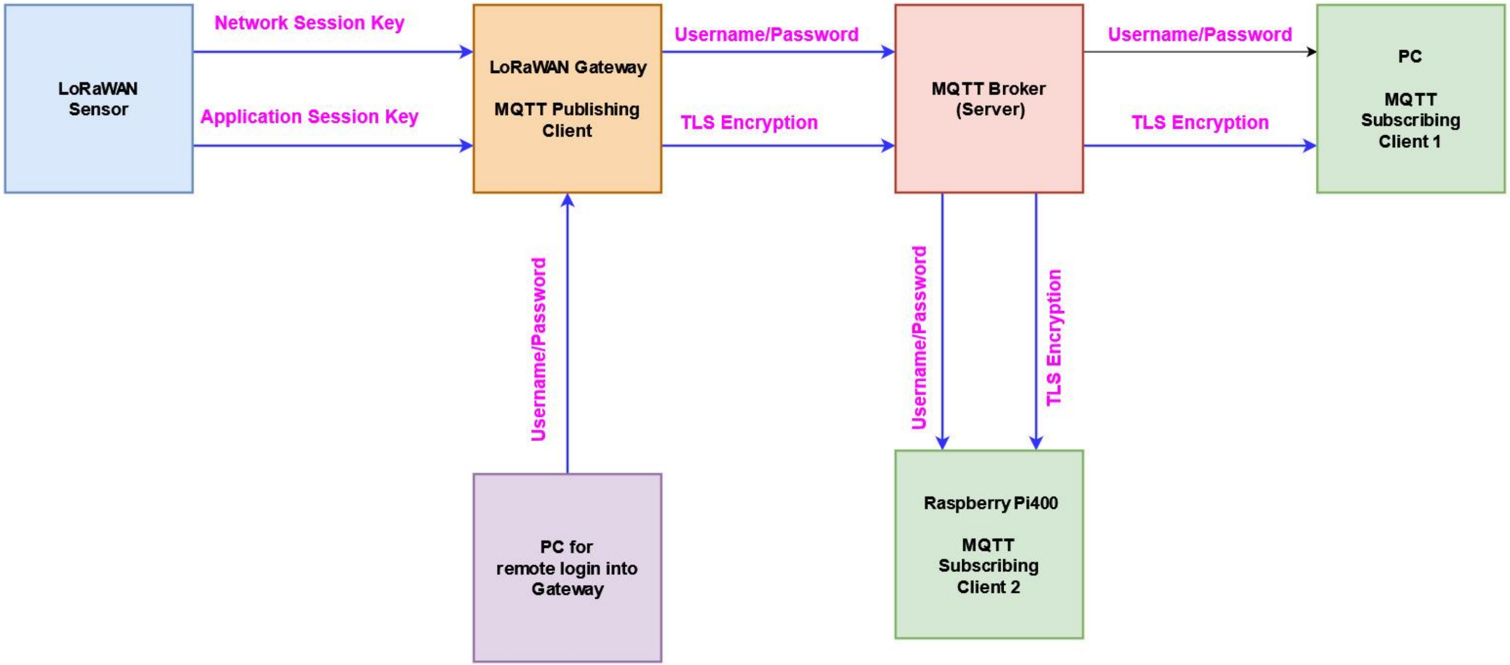
→ Custom JavaSript Function

Desired Payload

12/19/2022, 1:49:06 PM node: Modified_payload_Temp_01
msg : Object
▾object
▾payload: object
sensorId: "24E124136C198494"
timestamp: "2022-12-19T12:49:06.531Z"
temperature: 15.8
_msgid: "24216bb7.08f644"

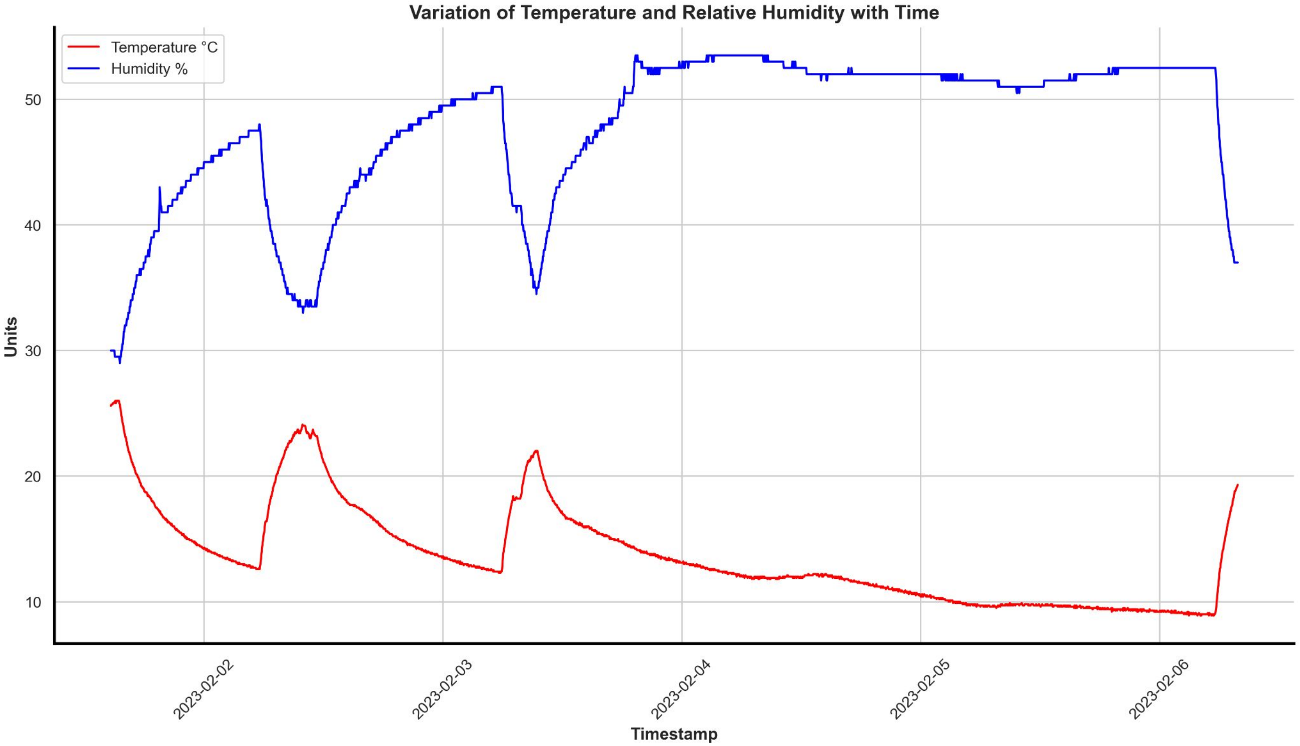
12/19/2022, 1:49:06 PM node: Modified_payload_humidity_01
msg : Object
▾object
▾payload: object
sensorId: "24E124136C198494"
timestamp: "2022-12-19T12:49:06.532Z"
humidity: 40
_msgid: "24216bb7.08f644"

Network Security Implementation



Results and Analytics from collected sensor data

Temperature humidity data collected from RHT sensors (rht -01)



Key parameters:

Test duration: 01- 06 Feb 2023

Recorded variables: Temp, RH, timestamp

Polling: 3 minutes

Data samples: 2264

Timestamp: ISO 8601 format

LoRa Frequency : 868 MHz band

Results and Analytics from collected sensor data

Temperature humidity data collected from RHT sensor (rht -02)

Topic	timestamp	Data	devEUI	deviceName	gatewayEUI	rssi	loRaSNR	frequency
MSG/RHT02	2023-01-31T15:27:47.584032Z	A2fEAARoVg==	24e124136c220737	rht-02	24e124fffef57bd0	-52	13.2	868300000
MSG/RHT02	2023-01-31T15:29:47.586139Z	A2fFAARoVg==	24e124136c220737	rht-02	24e124fffef57bd0	-53	14	868100000
MSG/RHT02	2023-01-31T15:31:47.586622Z	A2fEAARoVQ==	24e124136c220737	rht-02	24e124fffef57bd0	-49	11.5	868500000
MSG/RHT02	2023-01-31T15:33:47.593672Z	A2fFAARoVQ==	24e124136c220737	rht-02	24e124fffef57bd0	-52	13.8	868300000
MSG/RHT02	2023-01-31T15:35:47.582386Z	A2fFAARoVQ==	24e124136c220737	rht-02	24e124fffef57bd0	-46	13.5	868300000
MSG/RHT02	2023-01-31T15:37:47.587635Z	A2fFAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-45	10.2	868500000
MSG/RHT02	2023-01-31T15:39:47.578586Z	A2fEAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-47	10.2	868500000
MSG/RHT02	2023-01-31T15:41:47.582952Z	A2fFAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-56	13	868100000
MSG/RHT02	2023-01-31T15:43:47.586641Z	A2fFAARoVQ==	24e124136c220737	rht-02	24e124fffef57bd0	-50	13.5	868300000
MSG/RHT02	2023-01-31T15:45:47.580111Z	A2fEAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-50	9	868500000
MSG/RHT02	2023-01-31T15:47:47.584421Z	A2fEAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-47	13.8	868100000
MSG/RHT02	2023-01-31T15:49:47.583109Z	A2fGAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-46	13.5	868300000
MSG/RHT02	2023-01-31T15:51:47.580035Z	A2fFAARoVQ==	24e124136c220737	rht-02	24e124fffef57bd0	-50	13.5	868300000
MSG/RHT02	2023-01-31T15:53:47.575046Z	A2fFAARoVA==	24e124136c220737	rht-02	24e124fffef57bd0	-48	13	868100000

Key parameters:

Test duration:
31.01.2023 - 01.02.2023

Recorded variables:
Raw payload, timestamp, RSSI, SNR, LoRa frequency

Polling: 2 minutes

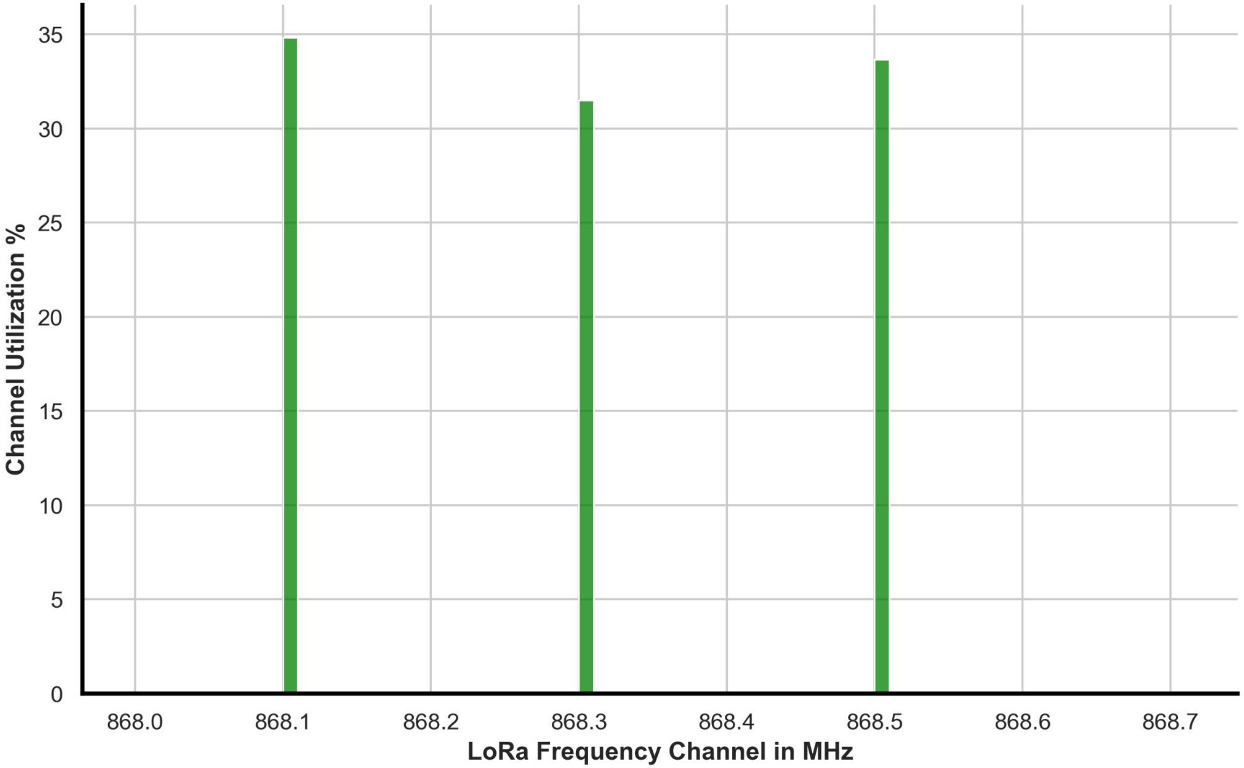
Data samples: 689

Timestamp: ISO 8601 format

LoRa Frequency : 868 MHz band

Results and Analytics from collected sensor data

LoRa Frequency channel usage (rht -02)



Key parameters:

Test duration:
31.01.2023 - 01.02.2023

Recorded variables: Raw payload, timestamp, RSSI, SNR, LoRa frequency

Polling: 2 minutes

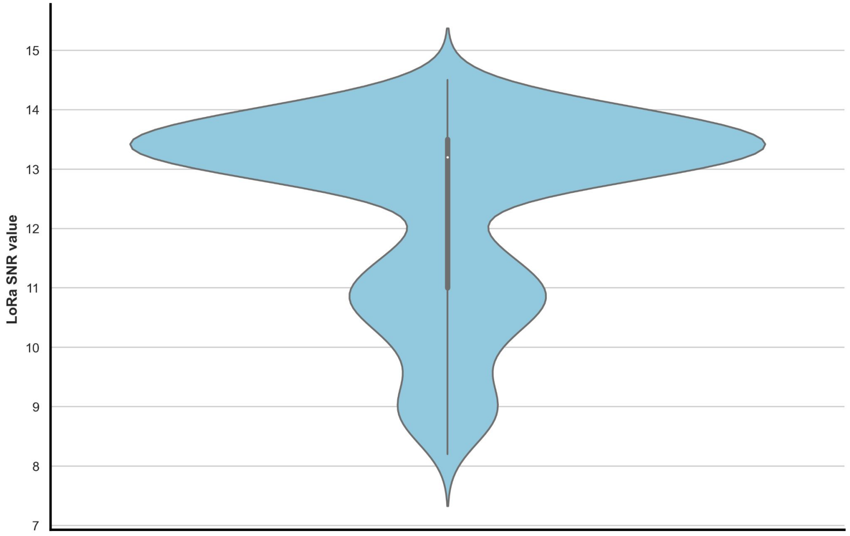
Data samples: 689

Timestamp: ISO 8601 format

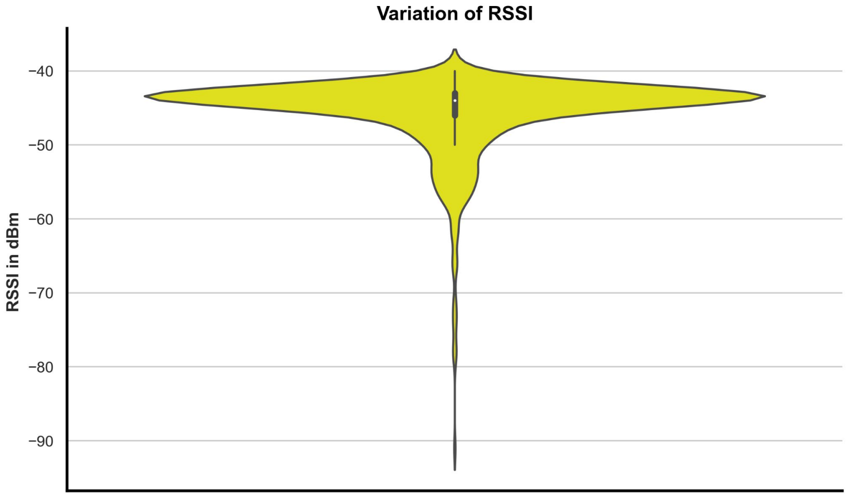
LoRa Frequency : 868 MHz band

RSSI and LoRa SNR variation from RHT sensor (rht -02)

LoRa SNR Variation



LoRa RSSI Variation



SNR varied from 8.2 to 14.5, Median SNR = 13.2

RSSI varied from -91 dBm to -40 dBm , Median RSSI = -44 dBm

Results and Analytics from collected sensor data

Data collected from Ultrasonic distance sensors (uds -01)

1	Topic	timestamp	devEUI	deviceName	Data_raw	battery	distance	position	gatewayEUI	rssi	loRaSNR	frequency
2	MSG/UDS01	2023-02-06T08:50:18.111428Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-56	10	868500000
3	MSG/UDS01	2023-02-06T08:52:18.109342Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-58	13.5	868100000
4	MSG/UDS01	2023-02-06T08:52:18.109342Z	24e124713c018455	uds-01	AXVjA4JeBwQAAA==	99	1886	normal	24e124fffef57bd0	-58	13.5	868100000
5	MSG/UDS01	2023-02-06T08:54:18.116412Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1886	normal	24e124fffef57bd0	-59	13.5	868100000
6	MSG/UDS01	2023-02-06T08:54:18.116412Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1728	normal	24e124fffef57bd0	-59	13.5	868100000
7	MSG/UDS01	2023-02-06T08:56:18.11149Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1728	normal	24e124fffef57bd0	-59	12	868500000
8	MSG/UDS01	2023-02-06T08:56:18.11149Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-59	12	868500000
9	MSG/UDS01	2023-02-06T08:58:18.115359Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-62	14	868100000
10	MSG/UDS01	2023-02-06T08:58:18.115359Z	24e124713c018455	uds-01	AXVjA4LCBgQAAA==	99	1730	normal	24e124fffef57bd0	-62	14	868100000
11	MSG/UDS01	2023-02-06T09:00:18.107491Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1730	normal	24e124fffef57bd0	-62	8.2	868500000
12	MSG/UDS01	2023-02-06T09:00:18.107491Z	24e124713c018455	uds-01	AXVjA4LABgQAAA==	99	1728	normal	24e124fffef57bd0	-62	8.2	868500000
13	MSG/UDS01	2023-02-06T09:02:18.106684Z	24e124713c018455	uds-01	AXVjA4LDBgQAAA==	99	1728	normal	24e124fffef57bd0	-64	13.5	868300000
14	MSG/UDS01	2023-02-06T09:02:18.106684Z	24e124713c018455	uds-01	AXVjA4LDBgQAAA==	99	1731	normal	24e124fffef57bd0	-64	13.5	868300000

Key parameters:

Test duration: 06 Feb 2023

Recorded variables: Distance, position, LoRa parameters

Polling: 2 minutes

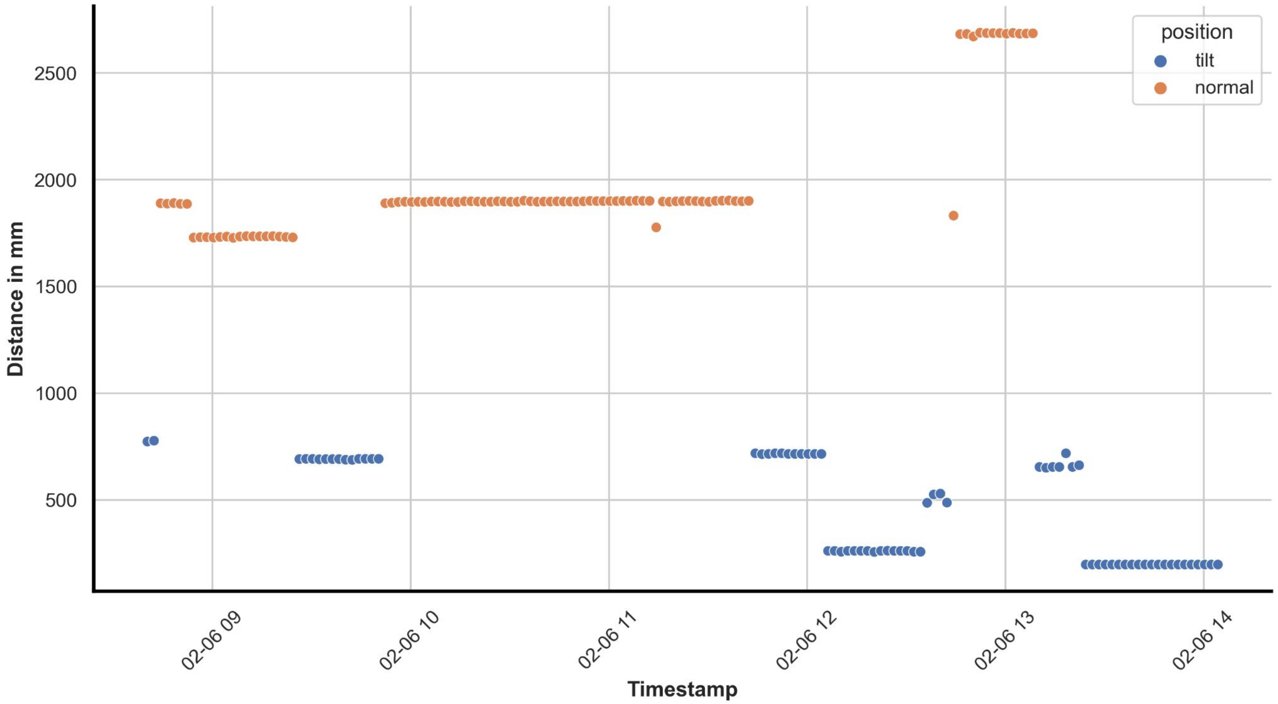
Data samples: 163

Timestamp: ISO 8601 format

LoRa Frequency : 868 MHz band

Results and Analytics from collected sensor data

Data collected from Ultrasonic distance sensors (uds -01)



Key parameters:

Test duration: 06 Feb 2023

Recorded variables: Distance, position, LoRa parameters

Polling: 2 minutes

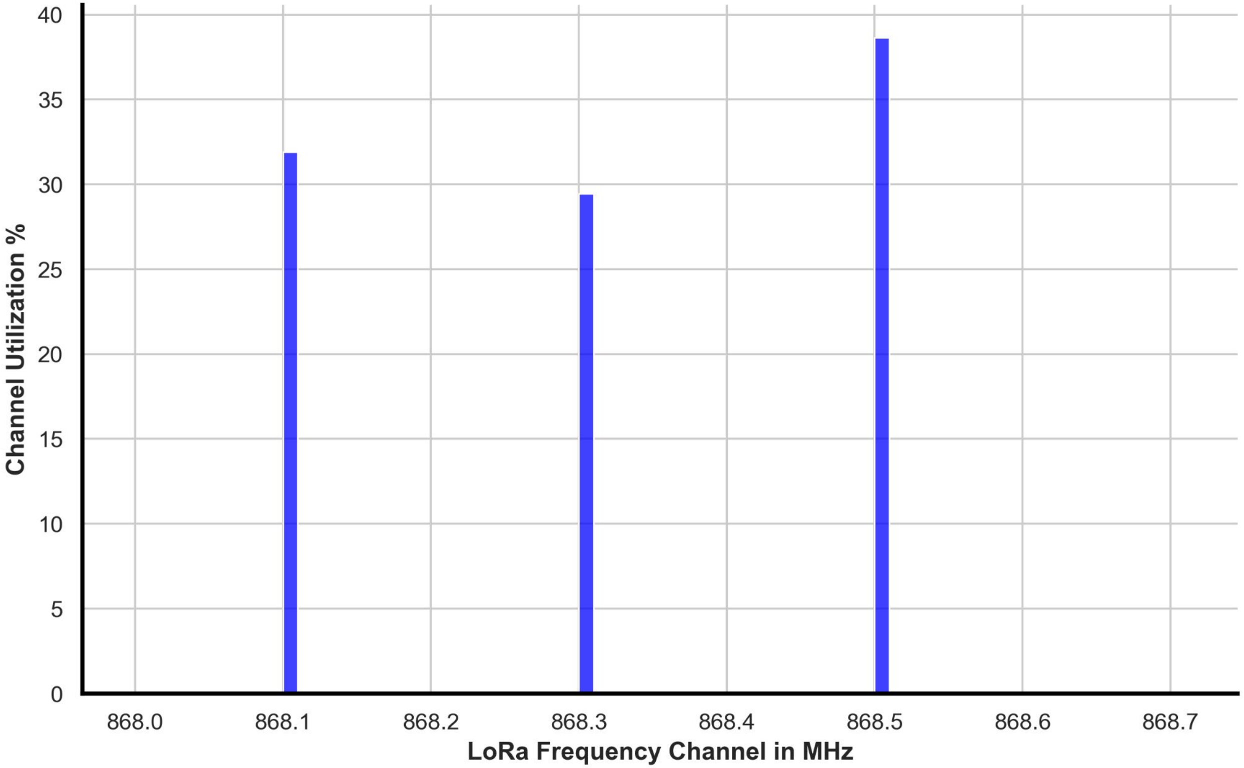
Data samples: 163

Timestamp: ISO 8601 format

LoRa Frequency : 868 MHz band

Results and Analytics from collected sensor data

LoRa Frequency channel usage (uds -01)



Key parameters:

Test duration: 06 Feb 2023

Recorded variables: Distance, position, LoRa parameters

Polling: 2 minutes

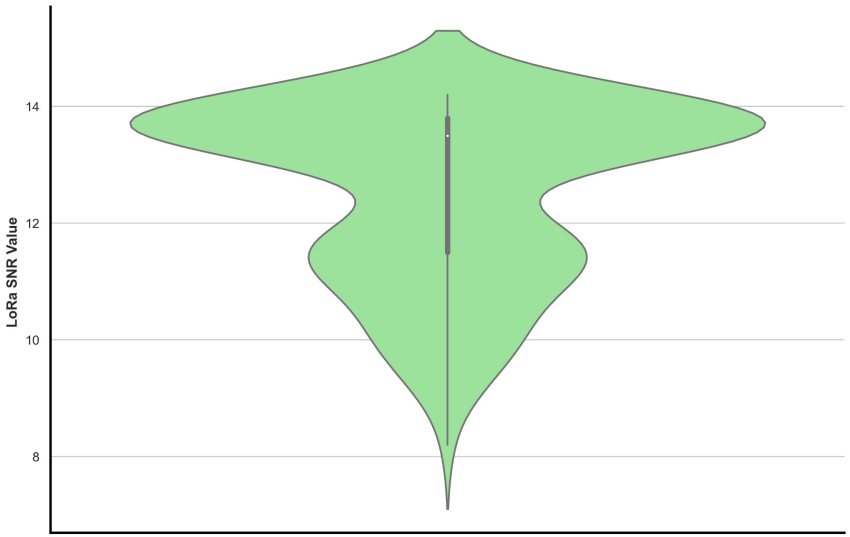
Data samples: 163

Timestamp: ISO 8601 format

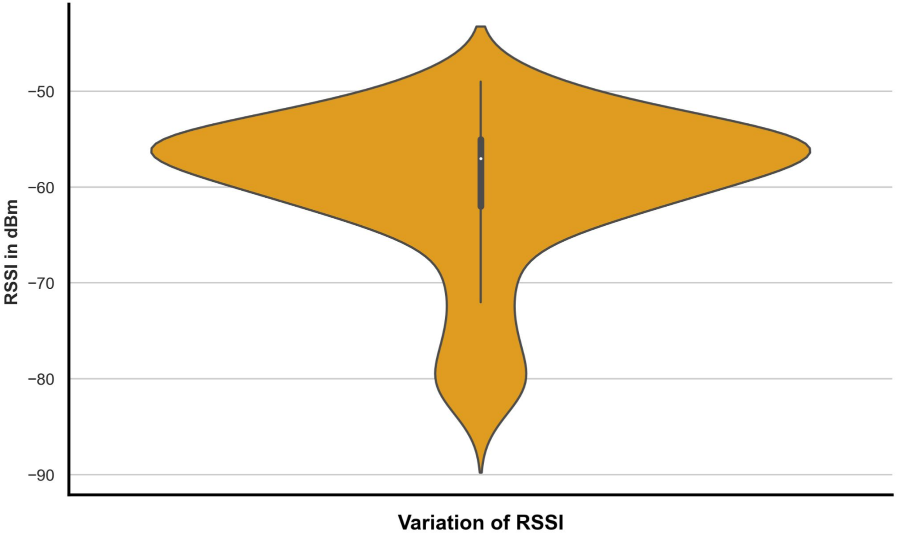
LoRa Frequency : 868 MHz band

RSSI and LoRa SNR variation from UDS sensor (uds-01)

LoRa SNR Variation



LoRa RSSI Variation



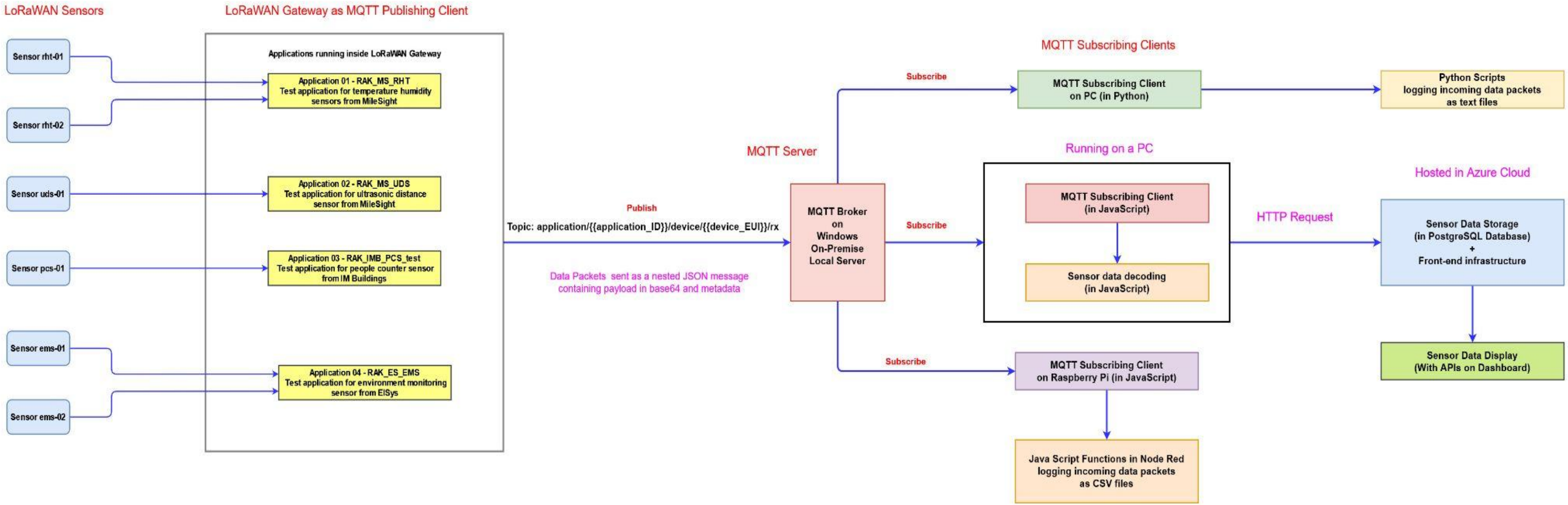
SNR varied from 8.2 to 14.2, Median SNR = 13.5

RSSI varied from -84 dBm to -49 dBm , Median RSSI = -57.6 dBm

Analysis on system architecture and test results

- It is possible to do the computational tasks of sensor payload decoding and data structure modification on either the publisher and the subscriber.
- Both methods works reliably. However not all gateways have the ability to run the decoding script. In this case only subscribing client can be used for these essential tasks.
- What happens when gateways from different manufacturers are used?
- What happens when there are a large number of gateways in the network?
- Gateway is already owned essential infrastructure, cloud / on-premise server requires additional costs and expertise. Example Azure IoT hub.
- Need of improvement in gateway software.

Network I - Test network in Campus



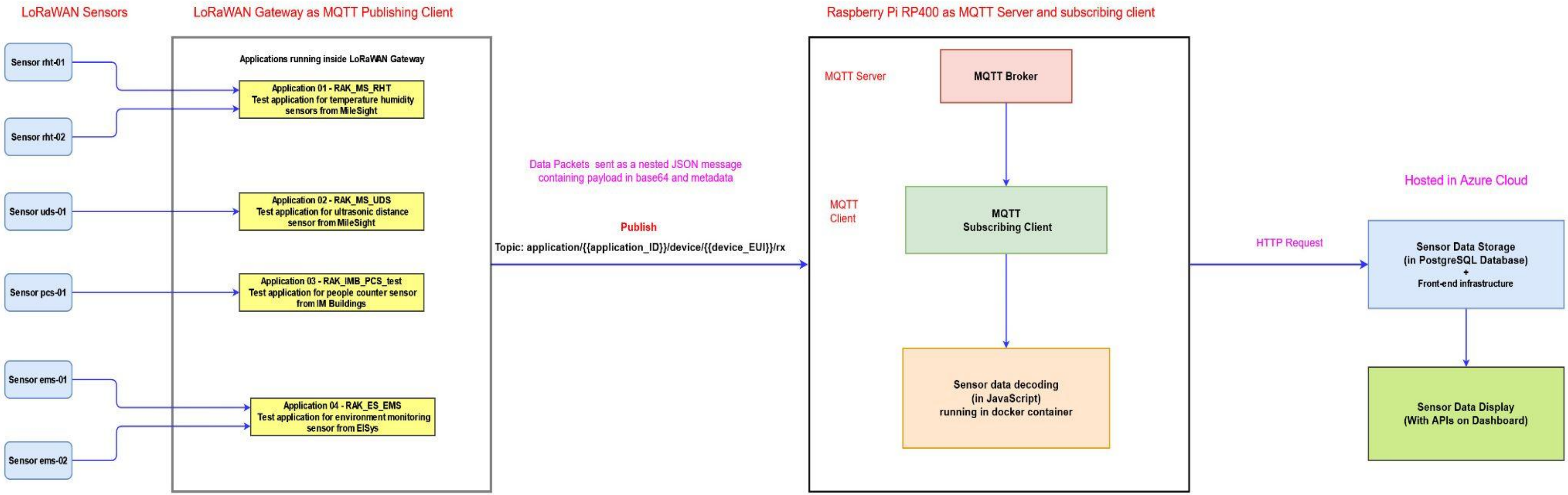
Key Features:

- Single point of integration for LoRaWAN gateway and MQTT Server.
- Improved application structure and no payload decoding in gateway.
- Simplified topic structure by application ID and Device EUI.
- Decoding scripts and MQTT subscribing clients were written mostly in JavaScript Mostly.

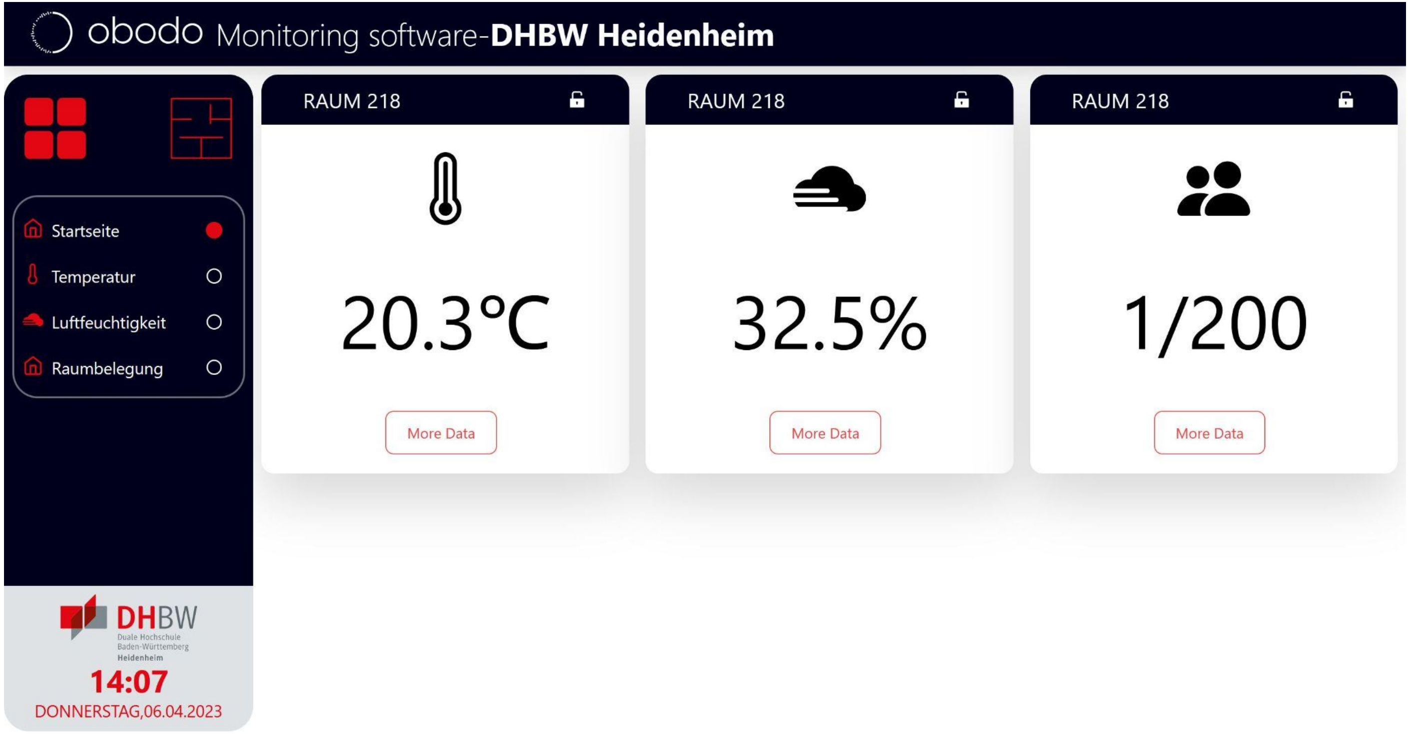
Network I - Collected Data

1	Topic	Time_stamp	Application_Name	Application_ID	Gateway_EUI	Device_EUI	Device_Name	Data	RSSI	LoRa_SNR	LoRa_Frequency	Data_Rate
2	application/4/device/a81758fffe07654a/rx	1678783701	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe07654a	ems-02	AQDSAiYDAQBABw4vCwAAAA4NAQ8AEgA=	-22	13.8	867300000	5
3	application/1/device/24e124136c198494/rx	1678783805	RAK_MS_RHT	1	ac1f09fffe070717	24e124136c198494	rht-01	A2fIAARoWQ==	-342	0	868800000	7
4	application/2/device/24e124713c018455/rx	1678783880	RAK_MS_UDS	2	ac1f09fffe070717	24e124713c018455	uds-01	AXVjA4JNBwQAAA==	-342	0	868800000	7
5	application/4/device/a81758fffe076551/rx	1678783887	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe076551	ems-01	AQDTAiYDAAA9Bw4lCwAAACcNAQ8AEgA=	-26	13.8	868300000	5
6	application/4/device/a81758fffe07654a/rx	1678784001	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe07654a	ems-02	AQDSAiYDAABABw4vCwAAAA4NAQ8AEgA=	-22	13.3	868300000	5
7	application/3/device/0004a30b00f7eac6/rx	1678784004	RAK_IMB_PCS	3	ac1f09fffe070717	0004a30b00f7eac6	pcs-01	AgYABKMLAPfqxgABJgAAAACAAAsACkQ=	-40	10.3	867100000	0
8	application/1/device/24e124136c198494/rx	1678784105	RAK_MS_RHT	1	ac1f09fffe070717	24e124136c198494	rht-01	A2fJAARoWA==	-342	0	868800000	7
9	application/2/device/24e124713c018455/rx	1678784180	RAK_MS_UDS	2	ac1f09fffe070717	24e124713c018455	uds-01	AXVjA4JZBwQAAA==	-342	0	868800000	7
10	application/4/device/a81758fffe076551/rx	1678784187	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe076551	ems-01	AQDSAiUDAP89Bw4lCwAAACcNAQ8AEgA=	-28	10	868500000	5
11	application/4/device/a81758fffe07654a/rx	1678784301	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe07654a	ems-02	AQDRAiYDAABABw4vCwAAAA4NAQ8CEgA=	-23	10	868500000	5
12	application/1/device/24e124136c198494/rx	1678784405	RAK_MS_RHT	1	ac1f09fffe070717	24e124136c198494	rht-01	A2fIAARoWA==	-342	0	868800000	7
13	application/2/device/24e124713c018455/rx	1678784480	RAK_MS_UDS	2	ac1f09fffe070717	24e124713c018455	uds-01	AXVjA4JaBwQAAA==	-342	0	868800000	7
14	application/4/device/a81758fffe076551/rx	1678784487	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe076551	ems-01	AQDSAiYDAAA+Bw4lCwAAACcNAQ8CEgA=	-27	14	867700000	5
15	application/4/device/a81758fffe07654a/rx	1678784601	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe07654a	ems-02	AQDQAiYDAP9ABw4vCwAAAA4NAQ8AEgA=	-22	13.8	867900000	5
16	application/1/device/24e124136c198494/rx	1678784705	RAK_MS_RHT	1	ac1f09fffe070717	24e124136c198494	rht-01	A2fIAARoWA==	-342	0	868800000	7
17	application/2/device/24e124713c018455/rx	1678784780	RAK_MS_UDS	2	ac1f09fffe070717	24e124713c018455	uds-01	AXVjA4JfBwQAAA==	-342	0	868800000	7
18	application/4/device/a81758fffe076551/rx	1678784787	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe076551	ems-01	AQDRAiYDAP8+Bw4lCwAAACcNAQ8AEgA=	-30	13.5	868100000	5
19	application/4/device/a81758fffe07654a/rx	1678784901	RAK_ES_EMS	4	ac1f09fffe070717	a81758fffe07654a	ems-02	AQDQAiYDAQBABw4vCwAAAA4NAQ8AEgA=	-23	13.3	867900000	5
20	application/3/device/0004a30b00f7eac6/rx	1678784904	RAK_IMB_PCS	3	ac1f09fffe070717	0004a30b00f7eac6	pcs-01	AgYABKMLAPfqxgABJgAAAACAAAsACkU=	-41	9.8	868300000	0

Network II - Network deployed for real world application



Dashboard designed for visualization of sensor data



Challenges and scope for future work

Challenges with LoRaWAN gateway:

- Output JSON not standardized- Effects database design
- MQTT integration with external server - (control level, application wise and global integration)
- MQTT client information control level varies from manufacturer to manufacturer
- Remote access to gateway - Difficult to access gateway from outside the network

Challenges with LoRaWAN sensor nodes:

- Sensor node configuration tools - Difficult to use, varying level of control for user
- Sensor payload decoding scripts - Only in JavaScript, No standardisation of logic
- Use of non standardized form factor in batteries - Vendor lock in

Challenges with MQTT and cloud integration

MQTT connection with cloud servers hs problems due to:
Security - TLS Certificate, MQTT protocol version, No connection on port 1883