Precision agriculture in vineyards using a low-power wide-area wireless sensor network

Bruno Casu

Content

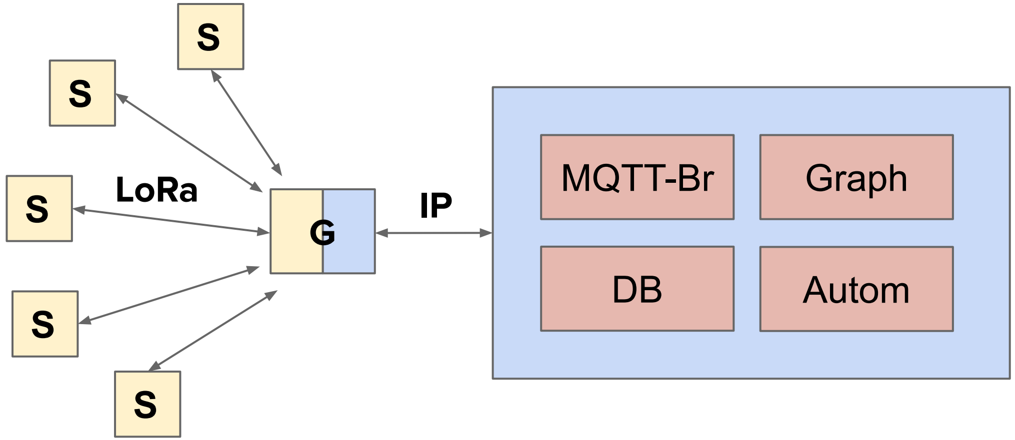
- Re-evaluation of WSN requirements
- Using LoRaWAN
- NUCLEO WL55JC1 Project
- Outlook

Re-evaluation of WSN requirements

Considerations about the WSN Configuration:

- Sensor nodes (LoRa End nodes) will attempt to reach the Gateway in one hop.
- The Gateway will remain in receiver mode, waiting for data frames from the End nodes (the Gateway will not be battery powered).
- End nodes will operate in non-beacon mode, transmitting data in a configurable interval.

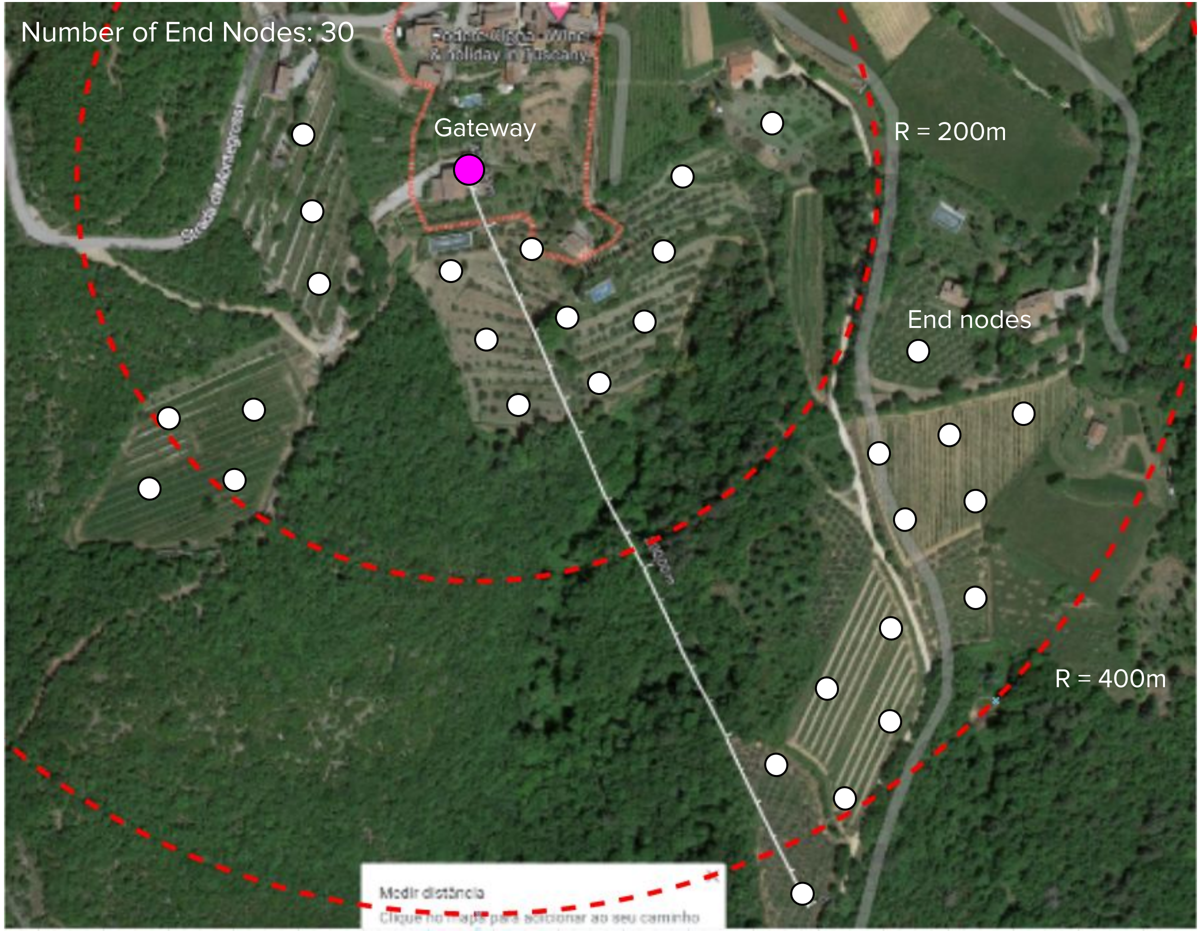
- **S**: LoRa End Node: sensore + MCU + transceiver LoRa + batteria
- **G**: LoRa Gateway: transceiver LoRa + SoC ARM/RiscV Linux + Ethernet PHY
- Back End: MQTT broker + database + grafici + automazione



Re-evaluation of WSN requirements - Environment

- Several hectares of vineyards (**from 5 ha up to 20 ha** - 20 ha is equivalent of a circular area with aprox. 500m in diameter). Sensor nodes will be distributed with an average of 30 meter between each node.
- Expected a number of sensors N in the network in the range of 20<N<100.
- Sensor nodes won't be guaranteed to be in line of sight with the Gateway.
- To ensure a high packet delivery ratio, the maximum distance between a sensor node and the gateway will be of aprox. 1000 meters (the maximum range is an estimate, the important parameter is the RSSI on the End Nodes - should be at the worst case -102dBm).
- To avoid Single point of failure (robust) an extra gateway may be used for redundancy.
- Sensor nodes will provide data with an interval of 30 minutes. After the Uplink frame, the End node will be able to receive a message from the central node application (configuration, action etc...)

Network Architecture re-evaluation - application scenario



Network Architecture re-evaluation - LoRa range evaluation

Field test using LoRa Radio with different Spread Factors configuration (Urban and Rural mixed area)

Table 2 LoRa Test Result

Point from Gateway	SF	RSSI (dBm) Mean	SNR (dB) Mean	PDR	PER
A (100 meter)	7	-68.8	>0	100%	0%
	8	-65.2	>0	100%	0%
	9	-66	>0	100%	0%
	10	-68	>0	100%	0%
	11	-70	>0	100%	0%
	12	-71	>0	100%	0%
B (240 meter)	7	-95.6	>0	97%	0%
	8	-96.3	>0	99%	0%
	9	-95.16	>0	100%	0%
	10	-96.51	>0	100%	1%
	11	-97.45	>0	100%	3%
	12	-98.33	>0	100%	3%
C (530 meter)	7	-103.091	-8.1	22%	50%
	8	-102.437	-7.73	71%	49.3%
	9	-102.222	-5.95	81%	11.22%
	10	-102.505	-7.097	93%	21.5%
	11	-102.4	-6.357	70%	8.56%
	12	-102.083	-8.3	72%	45.83%

Network Architecture re-evaluation - LoRa range evaluation

Comparative test with 802.15.4 Radio

Semi-rural Scenario

Distance	PDR
50m	100%
100m	75%
150m	70%
200m	40%
250m	60%
300m	0%

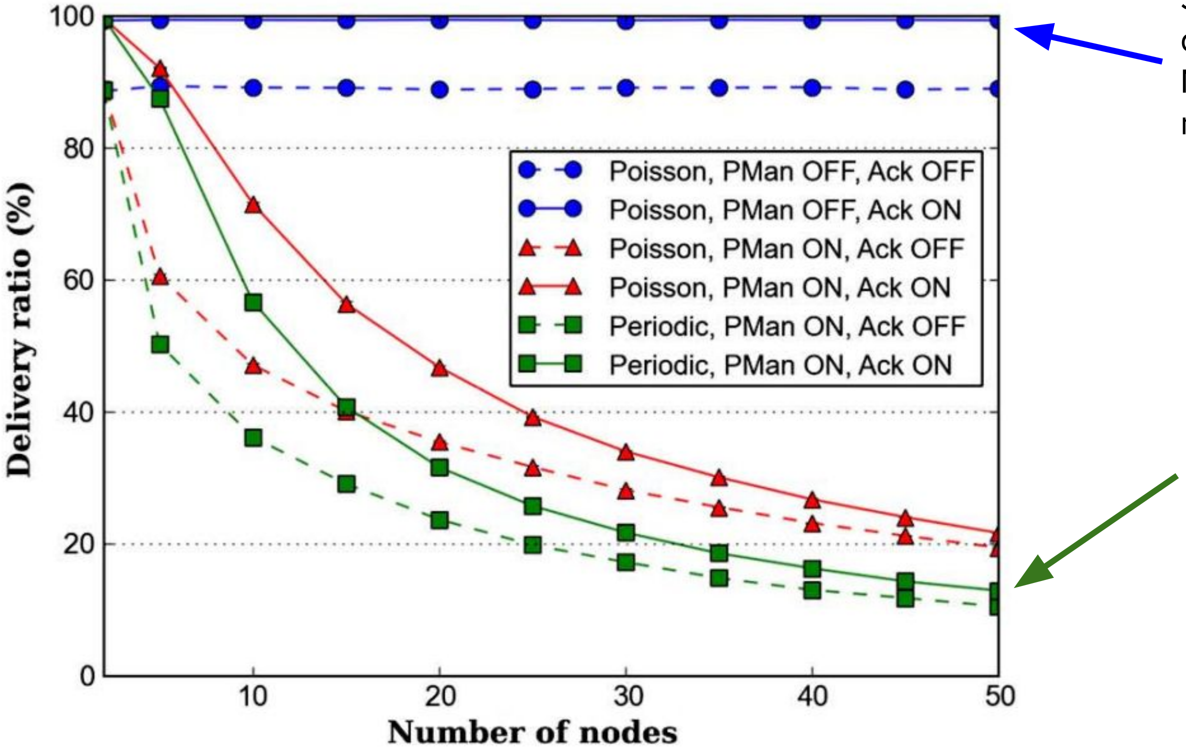
Test Config:

1. Zolertia re-mote sensor nodes equipped with omni-directional antennas with 2 dB gain.
2. Sensor nodes are based on the ARM Cortex-M3 system on chip (SoC), and runs the Contiki OS.
3. Texas Instruments CC1120 transceiver, default configuration for the IEEE 802.15.4g standard, operating at 868 MHz band, with 2- GFSK modulation.
4. Bit rate 50 Kbps, the receiver sensitivity is -109 dBm.

Network Architecture re-evaluation

From: A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks

End Nodes density and packet delivery ratio in IEE802.15.4 MAC (default config):



(a)

Similar Config to LoRaWAN Class A device setup (non beacon mode - Power Management OFF - Gateway always in Rx mode)

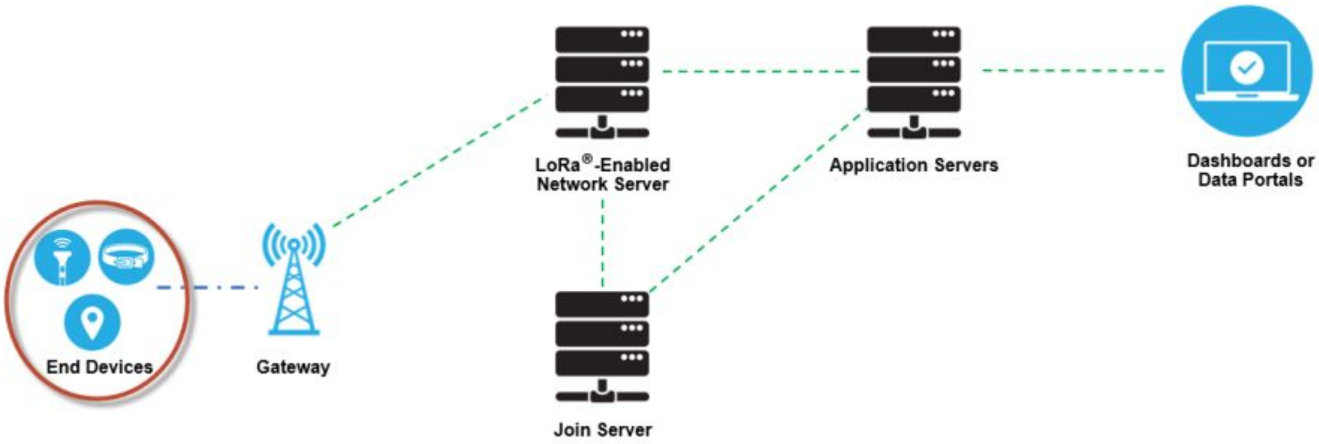
For a multihop implementation (hierarchical WSN), considering that the PAN coordinator is battery powered, beacon mode is the standard solution.

Configuration with Power Management ON (PAN coordinators are not in continuous Rx mode)

The low packet delivery ratio is due to the use of the Contention Access Period (CAP) with Limited retries (End nodes compete for channel access using a slotted CSMA-CA)

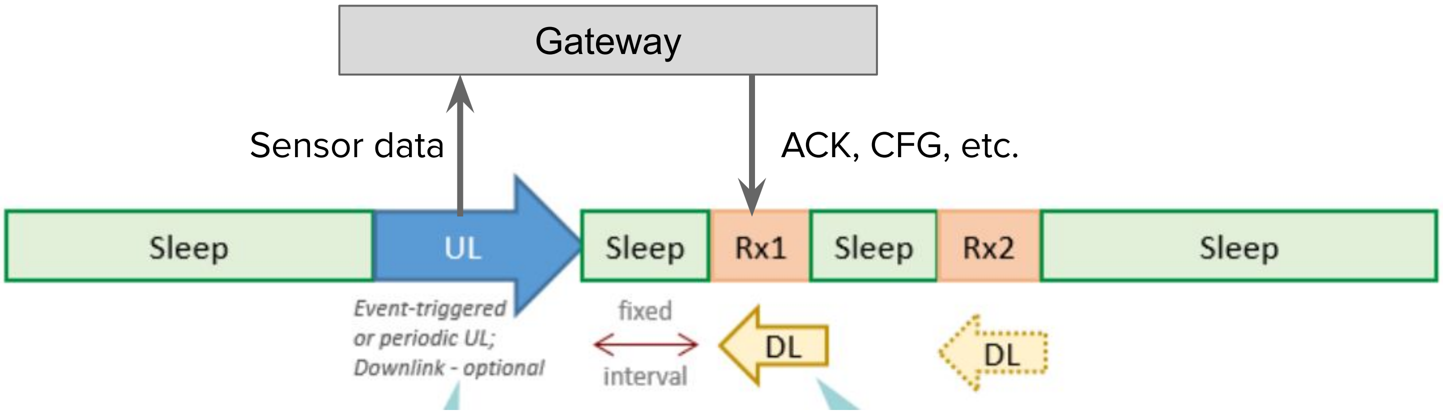
Using LoRaWAN

- Star topology network design (End nodes are connected to a gateway, and can reach it in one hop).
- Optimized for low power (duty cycling and access layer protocol are provided).
- High network capacity and minimal infrastructure (many sensor nodes can be handled by one gateway).
- Supports Firmware upgrade over-the-air (FUOTA) for applications.
- Provides secure encrypted communication.
- Software is under an Open Source license agreement.



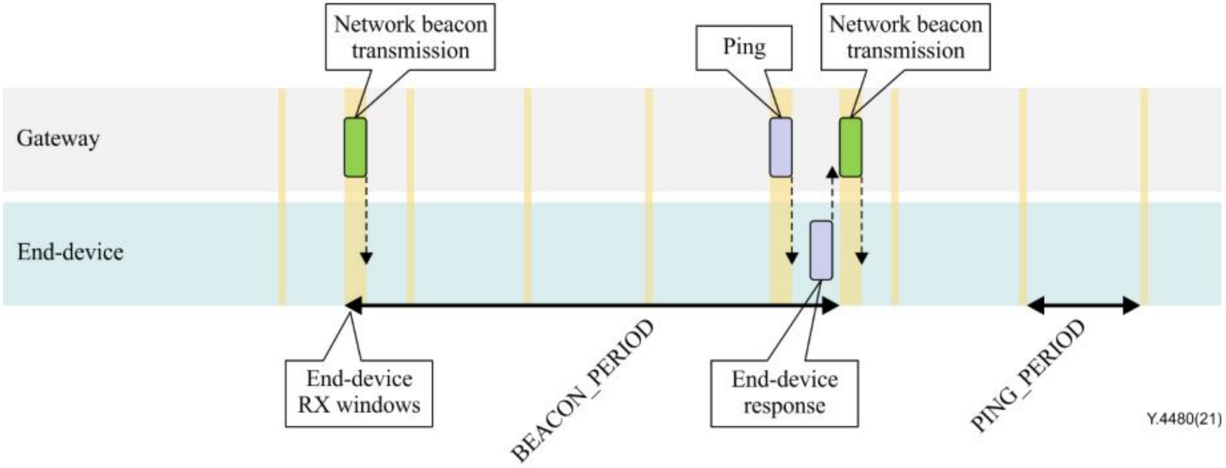
WSN architecture - LoRaWAN

- LoRaWAN End Node Classes:
  - **Class A** - Operate in Non Beacon Mode, sending data with a set interval. Most efficient implementation considering power consumption and duty cycling.



WSN architecture - LoRaWAN

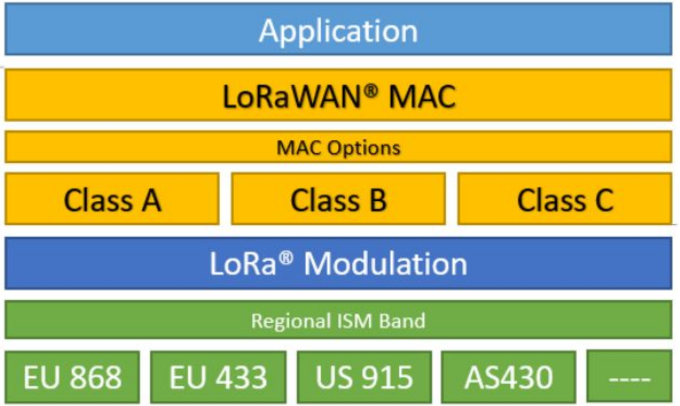
- LoRaWAN End Node Classes:
  - **Class B** - These devices operate in Beacon Enabled mode. This allows End nodes to synchronize a downlink slot (may improve system response time, if actions in the field are required with lower latency). Uplink works the same as Class A devices.



  - **Class C** - These devices must always remain active (**NOT suited for our application scenario**).

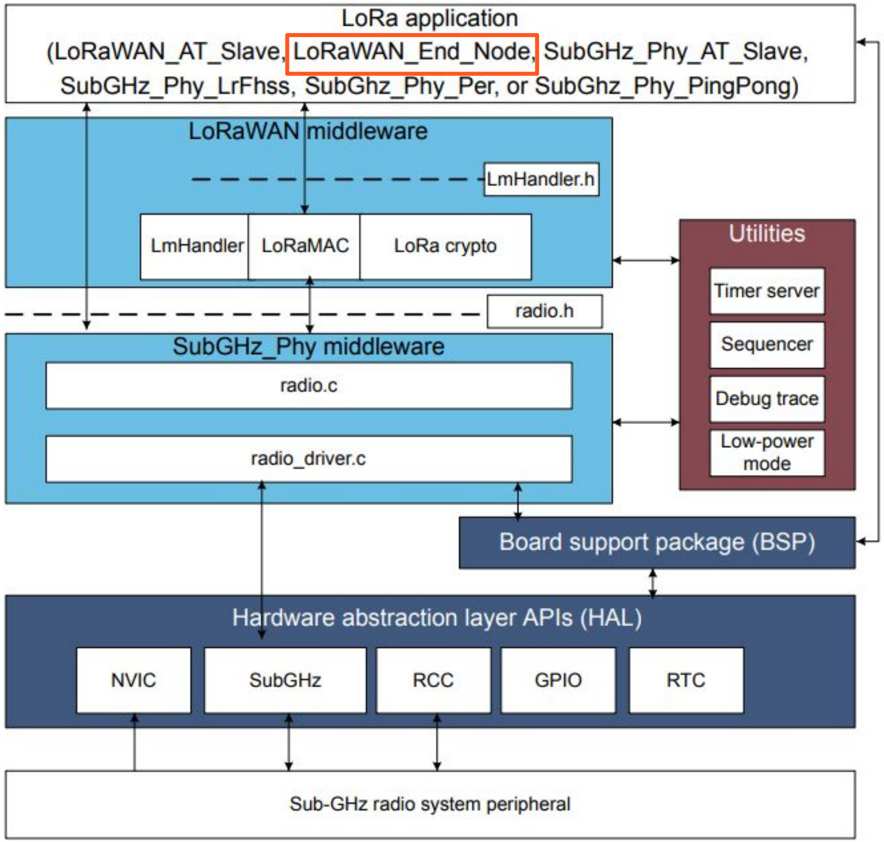
WSN architecture - LoRaWAN

LoRa protocol stack



STM32 Nucleo firmware

Figure 3. Static LoRa architecture



WSN architecture - Additional functionalities

Some additional functionalities could be added in the Application layer to handle possible adverse scenarios:

- When the Uplink connection is not established during the time frame (adverse environment conditions, or possible collisions with other nodes), the application layer should be able to retransmit sensor data. Possible implementations:
    - **Retention of latest sensor readings**: when the uplink connection fails, the current sensor data is stored in memory, and added to the next frame to be sent (no retransmission attempts). This can be recurred up to the limit of the frame size.
    - **Retransmission period**: The End node could automatically reduce its Standby period and retry the Uplink communication (set by a random value).
        - Retries outside the standard Uplink frame will increase power consumption, but can increase overall packet delivery ratio.

WSN architecture - Additional functionalities

- LoRaWAN Class A devices are described as “not suited for actuators”.
  - To add actuation capabilities in the sensor nodes, an application can be developed for the End Nodes to **process the readings locally** and evaluate the necessity countermeasures. The thresholds can be configured and set remotely allowing a central application to update the parameters.
  - Alternatively, the sensor nodes can be reconfigured once a central monitoring application detects possible risk scenarios. When reconfigured, the End nodes will **change its duty cycle to send data more frequently**, allowing the central application to issue an **Action Command** in the Rx slots available in the Class A transmission frame.

NUCLEO WL55JC1 Project

- Using **STM32Cube_FW_WL_V1.3.0** software package. Already contains LoRaWAN and FreeRTOS Middleware, BSP drivers, SX1276 Drivers and ready to use examples.
- Project Setup: DualCore mode with FreeRTOS

Some remarks:

- Core M4 project contains the lora app and configurations. The scheduler is initialized on Core M4, but the LoraWAN task runs on Core M0+. After boot, Core M4 can be set to Standby mode.
- Basic Lora app configs:
  - APP_TX_DUTYCYCLE 10000 // 10s
  - ACTIVE_REGION LORAMAC_REGION_EU868
  - LORAWAN_DEFAULT_CLASS CLASS_A
  - LORAWAN_APP_DATA_BUFFER_MAX_SIZE 242

NUCLEO WL55JC1 Project - basic operation

void **LoRaWAN_Init**(void)

```
Thd_LoraSendProcessId = osThreadNew(Thd_LoraSendProcess, NULL, &Thd_LoraSendProcess_attr);
UTIL_TIMER_Create(&TxTimer, TxPeriodicity, UTIL_TIMER_ONESHOT, OnTxTimerEvent, NULL);
UTIL_TIMER_Start(&TxTimer);
```

Scheduler start

Main thread

static void **Thd_LoraSendProcess**(void *argument)

```
osThreadFlagsWait(1, osFlagsWaitAny, osWaitForever);
SendTxData();
```

RTC Count = TxPeriodicity

callback

static void **OnTxTimerEvent**(void *context)

Set flag
Reset timer

static void **SendTxData**(void)

```
humidity    = (uint16_t)(sensor_data.humidity * 10);          /* in %*10     */
temperature = (int16_t)(sensor_data.temperature);
pressure = (uint16_t)(sensor_data.pressure * 100 / 10); /* in hPa / 10 */
AppData.Buffer = sensor_data…
status = LmHandlerSend(&AppData, LmHandlerParams.IsTxConfirmed, false);
```

NUCLEO WL55JC1 Project - Logs from the Serial interface

Board Initialization Log

CM0PLUS : Lora registration done
M4_APP_VERSION: V1.3.0
M0PLUS_APP_VERSION: V1.3.0
MW_LORAWAN_VERSION: V2.5.0
MW_RADIO_VERSION: V1.3.0
L2_SPEC_VERSION: V1.0.4
RP_SPEC_VERSION: V2-1.0.1
AppKey:
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
NwkKey:
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
AppSKey:
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
NwkSKey:
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
DevEUI: 00:80:E1:15:00:0A:90:39
AppEUI: 01:01:01:01:01:01:01:01
DevAddr: 00:0A:90:39
KMS ENABLED

First JOIN attempt

DutyCycle 10s

0s048:TX on freq 868300000 Hz at DR 0
1s534:MAC txDone
6s566:RX_1 on freq 868300000 Hz at DR 0 ***RX1 window***
6s764:IRQ_RX_TX_TIMEOUT
6s764:MAC rxTimeOut
7s566:RX_2 on freq 869525000 Hz at DR 0 ***RX2 window***
7s764:IRQ_RX_TX_TIMEOUT
7s764:MAC rxTimeOut

= JOIN FAILED
10s051:VDDA: 254 ***Battery level***
10s051:temp: 23 ***Temperature reading***
10s056:TX on freq 868500000 Hz at DR 0 ***New Transmission attempt***
11s541:MAC txDone
16s574:RX_1 on freq 868500000 Hz at DR 0
16s772:IRQ_RX_TX_TIMEOUT
16s772:MAC rxTimeOut
17s574:RX_2 on freq 869525000 Hz at DR 0
17s772:IRQ_RX_TX_TIMEOUT
17s772:MAC rxTimeOut

= JOIN FAILED

Outlook

- All the LoRaWAN network configurations are already set by the software package. The only necessary software for the NUCLEO End node is the application layer.
- The example is already running, next step is to setup and test the Gateway (Dragino Board).
- Choose a sensor to deploy with the NUCLEO End node - possibly BMP280 (temperature + pressure).
- For the Database to be implemented a document database is suited for sensor data applications (MongoDB + Grafana is a possible solution).

Backup

```
20s062:TX on freq 868500000 Hz at DR 0
21s546:MAC txDone
26s579:RX_1 on freq 868500000 Hz at DR 0
26s777:IRQ_RX_TX_TIMEOUT
26s777:MAC rxTimeOut
27s579:RX_2 on freq 869525000 Hz at DR 0
27s777:IRQ_RX_TX_TIMEOUT
27s777:MAC rxTimeOut

###### = JOIN FAILED
30s069:TX on freq 868300000 Hz at DR 0
31s553:MAC txDone
36s585:RX_1 on freq 868300000 Hz at DR 0
36s784:IRQ_RX_TX_TIMEOUT
36s784:MAC rxTimeOut
37s585:RX_2 on freq 869525000 Hz at DR 0
37s784:IRQ_RX_TX_TIMEOUT
37s784:MAC rxTimeOut

###### = JOIN FAILED
```

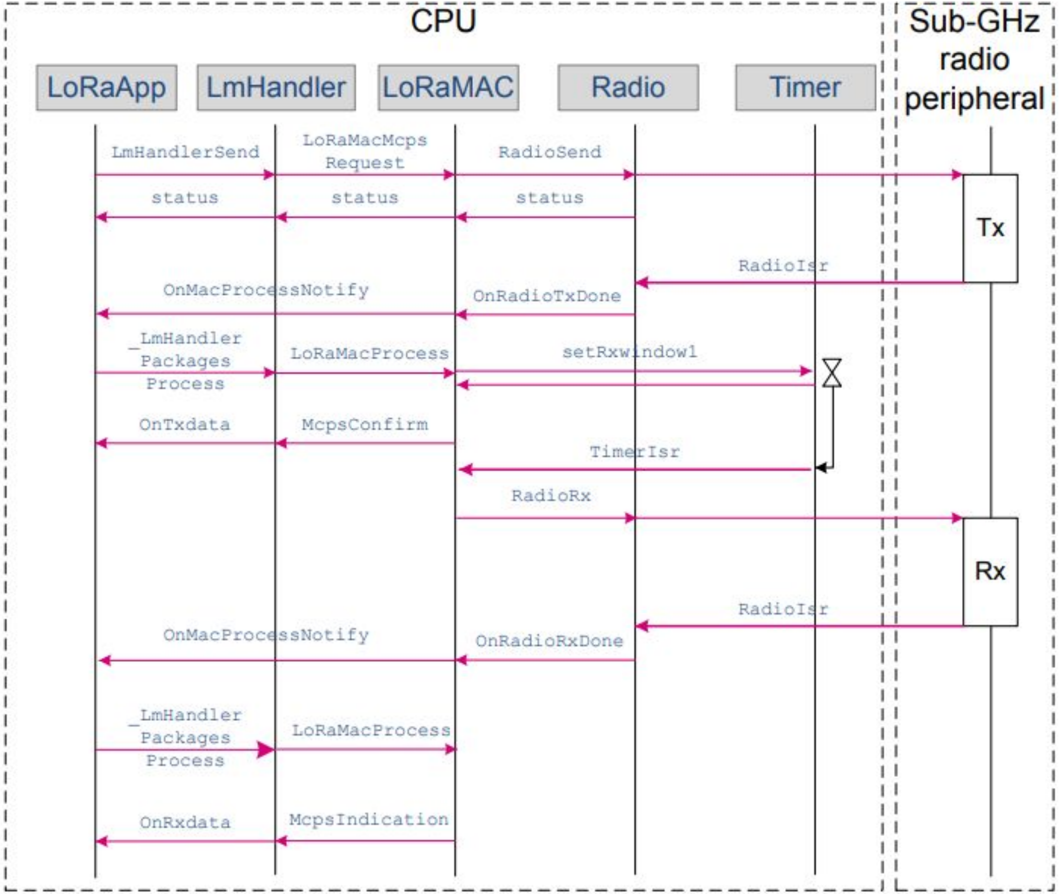
Duty Cycle reconfigured for 30s

```
120s076:TX on freq 868500000 Hz at DR 0
121s560:MAC txDone
126s592:RX_1 on freq 868500000 Hz at DR 0
126s791:IRQ_RX_TX_TIMEOUT
126s791:MAC rxTimeOut
127s592:RX_2 on freq 869525000 Hz at DR 0
127s791:IRQ_RX_TX_TIMEOUT
127s791:MAC rxTimeOut

###### = JOIN FAILED
150s083:TX on freq 868100000 Hz at DR 0
```

Backup

Figure 4. **Class A Tx and Rx processing MSC**



Backup

Very large open field scenario - The developed End nodes may not be able to communicate in such scenario.

Gateway could be moved to a central position.