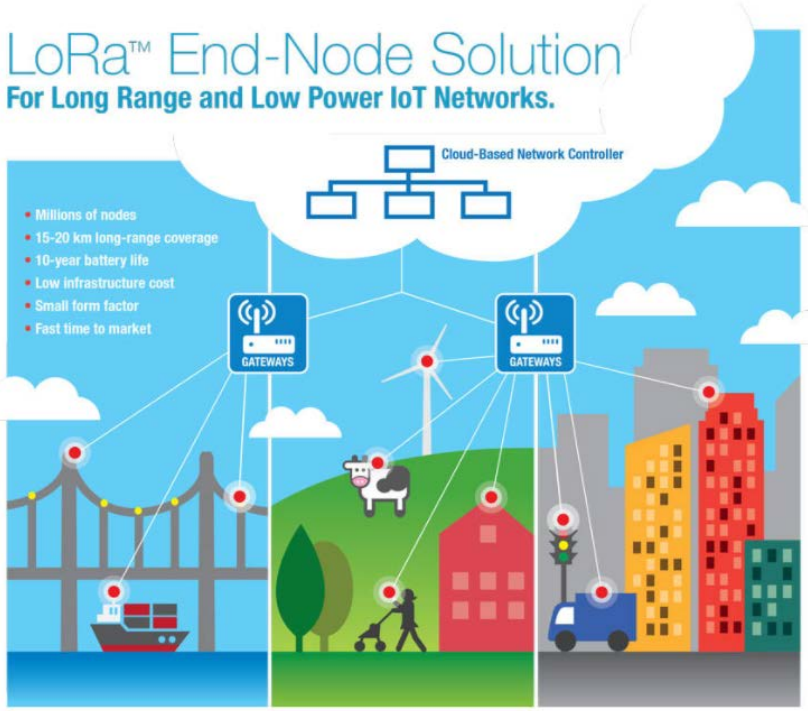
# RN2903
# Microchip LORA module
# REV 1.0





**Mr. Natapol J. /SR. FAE**

*ELECTRONICS SOURCE CO., LTD www.es.co.th*



***Email: natapol@es.co.th***

# Agenda

1) Background information about LoRaWAN
2) LoRaWAN Class A,B,C
3) LoRaWAN
4) RN2903 Module
5) Microchip LoRa Mote board /BASIC and Overview
6) Mote Menu operating
7) RN2903 Command via UART
8) HOW to connect between Mote and Gateway
9) RN2903A Command detail
10) GPIO Control via serial command
11) FSK and LoRa mode (P2P)
12) Reference website link

# Background information about LoRaWAN

# Terminology

- **End Device, Node, Mote** - an object with an embedded low-power communication device.

- **Gateway** - antennas that receive broadcasts from End Devices and send data back to End Devices.

- **Network Server** - servers that route messages from End Devices to the right Application, and back.

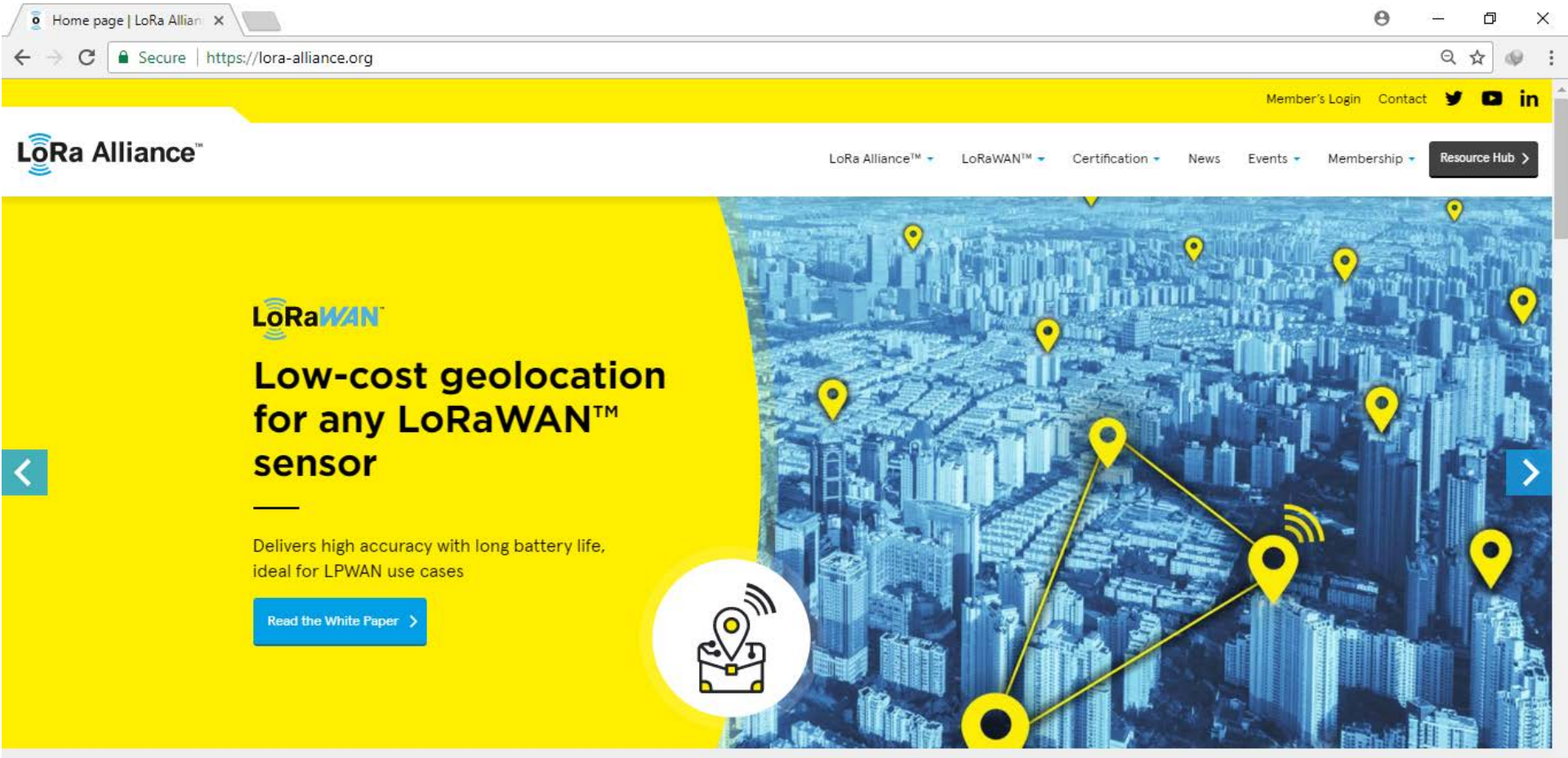
# Agenda

## Terminology

- **Application** - a piece of software, running on a server.
- **Uplink Message** - a message from a Device to an Application.
- **Downlink Message** - a message from an Application to a Device.

# LoRa™ Alliance

## Wide Area Networks for IoT

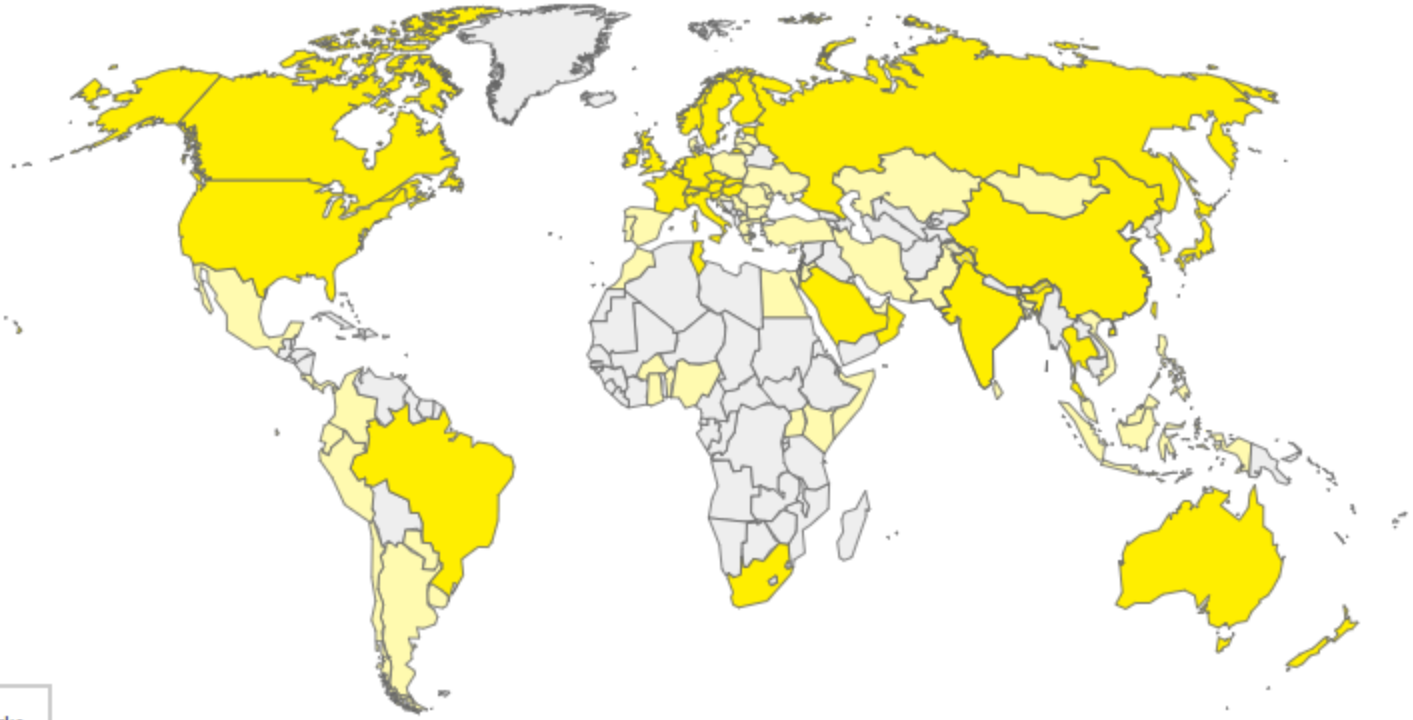

Home page | LoRa Allian
Secure
https://lora-alliance.org
Member's Login
Contact
LoRa Alliance™
LoRa Alliance™
LoRaWAN™
Certification
News
Events
Membership
Resource Hub
LoRaWAN
Low-cost geolocation for any LoRaWAN™ sensor
Delivers high accuracy with long battery life, ideal for LPWAN use cases
Read the White Paper

**83**
Network Operators

**57**
Alliance Member Operators

**49**
Countries operating in

**95**
Countries with LoRaWAN Deployments



# The LoRa® Alliance

**“ENABLING THINGS TO HAVE A GLOBAL VOICE”**

- The LoRa® Alliance (http://lora-alliance.org/) is an **open**, non-profit association of members.
- **Mission**: to standardize Low Power Wide Area Networks
- Alliance members will collaborate to drive the global success of the LoRaWAN™ protocol

**Strategy Committee**
Roadmap & security

**Technical Committee**
Specification & feature updates


**Marketing Committee**
Brand, media, trade-shows, open house

**Certification Committee**
Test specs & accreditation

# LoRa® Alliance Ecosystem

500+ members

# Alliance Member Open Network Operators

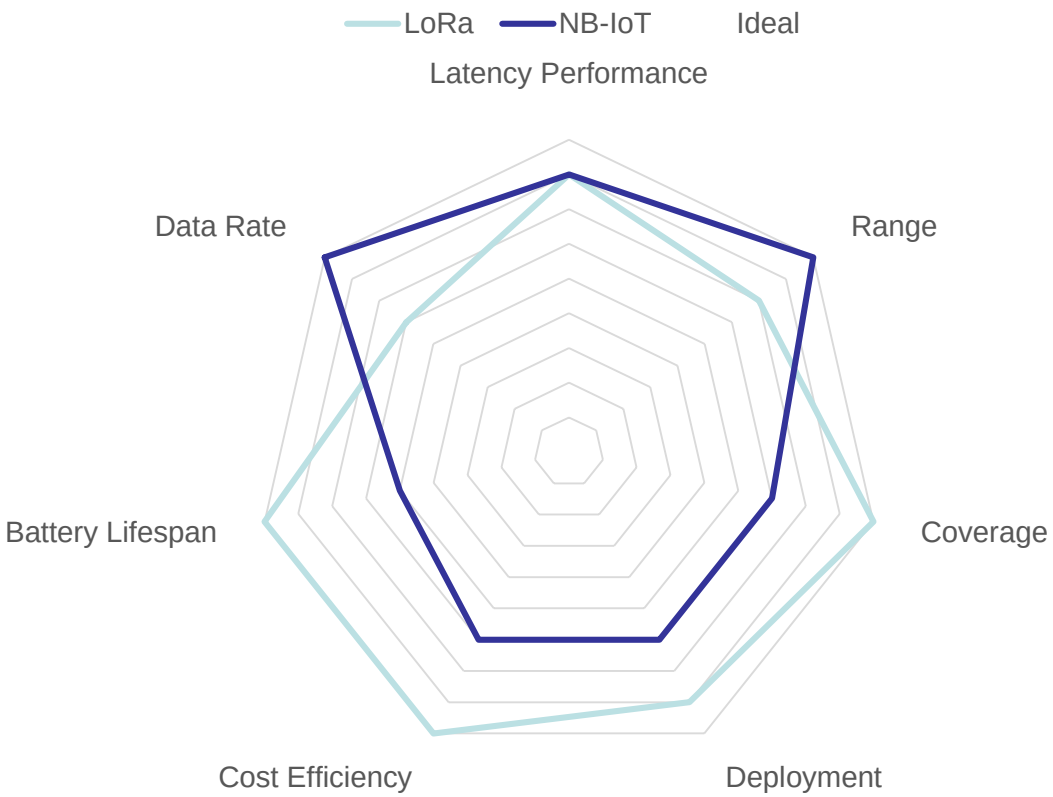

All information contained herein is curr .oRa Alliance is . of information presented

March 2018

# LPWAN Comparison

| Parameters | | LoRa | NB-IoT | Remarks |
|---|---|---|---|---|
| Spectrum | | Unlicensed | Licensed LTE bandwidth | Heavy Investment for Licence of Spectrum |
| Modulation | | CSS | QPSK | |
| Bandwidth | | 500 kHz–125 KHz | 180 KHz | |
| Peak Data Rate | | 290 bps-50 Kbps (DL/UL) | DL:234.7 kbps; UL:204.8 kbps | LoRa Targeting at low Data application |
| Link Budget | | 154 dB | 150 dB | |
| Max. # message/day | | Unlimited | Unlimited | |
| Power efficiency | | Very High | Medium High | |
| Mobility | | Better than NB-IoT | No connected mobility (only idle mode reselection) | |
| Battery Lifespan | | >10 years battery life of devices | >10 years battery life of devices | More Information needs for this cliam |
| Spectrum Efficiency | | Chirp SS CDMA better than FSK | Improved by standalone, in-band, guard band operation | |
| Area Traffic Capacity | | Depends on gateway type | 40 devices per household, 55k devices per cell | |
| Interference immunity | | Very High | Low | Noise Interference |
| Power Efficiency | Typ Tx Current | 124mA (20 dBm) | 220 mA (23 dBm) | 2x Tx Current |
| | Transmission Time | 0.93 s (16 Bytes payload) | 2 sec (for Tx) and 10 secs (for Rx) | 5x Time |
| | Active Mode Current (Rx) | 38 mA | 46mA | |
| | Sleep Current | 1.8 uA | 3 uA | 3x Sleep |
| | Device Class | Class A : Sleep except when Tx | PSM (Power Saving Mode) | |
| | | Class B : Periodically wakes up to listen for | eDRX (Extended Discontinuous | |
| | | Class C : In Rx mode most of the time | | |

# LoRa vs NB-IoT



# LoRaWAN
## Global Standards for LPWAN IoT Networks

**Long Range**
Provides Long range up to 15km communications.


**Licensed-Free Band**
Unlicensed spectrum provides low-cost deployment for LoRaWAN ; licensed band results in high investment.

**Bidirectional**
Provides Uplink and Downlink communication to enables wide variety of uses case.

**High Penetration & Low Noise**
The LoRaWAN modulation provide high penetration over walls and provide the ability to communicate with indoor monitoring sensors. Innovative & cost sensitive implementation providing Demod signals below the noise floor, improving sensitivity by ~20dB

**Open Source Standard**
LoRaWAN standard is based on an open protocol approach managed by a strong team of Alliance members.


**Power Consumption**
LoRaWAN data transmission and reception requires ultra low power and providing battery operation an excellent feature for long lifespan.

**Cost**
LoRaWAN open standard combined with cost-free operations frequencies and cost-effective base-stations allowing public networks and private networks to be roll out in a very short timeframe. Provides a Opex business model over Capex


**Unique Value Proposition**
Unlike NB-IoT where it is limited to locations where 4G/LTE base stations can be set up. The number of base stations is about 7 – 10 times less compared to NB-IoT.

# Application

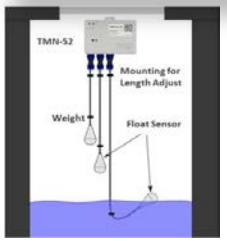
## Covering diverse applications with a common air interface and network architecture

Huge amount of devices with small data and delay tolerant traffic

**Smart Metering**
- Small amount of uplink only
- Every 15 min
- Rare handover

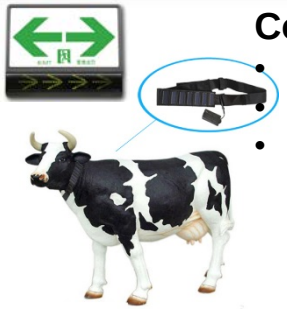

**Connected Trash Cans**
- Small amount of uplink only
- Spontaneous communication
- Rare handover

Devices running on Battery only – sometimes in difficult environment




**Smart Metering (Flood/Water/…)**
- 10 year battery life time
- Located in the basement
- Rare handover


LoRa



**Connected Herd**
- Small amount of uplink only
- Spontaneous & Periodic communication
- Rare handover (coverage)

# Deployment – Nationwide
## Australia & New Zealand

*New Zealand telecommunications carrier Spark has partnered with Activity to build a LoRa® IoT network across the country by 2018.*

*NNNCo announces rollout of LoRaWAN™ network comes six months after the trial of the LoRaWAN™ technology.*

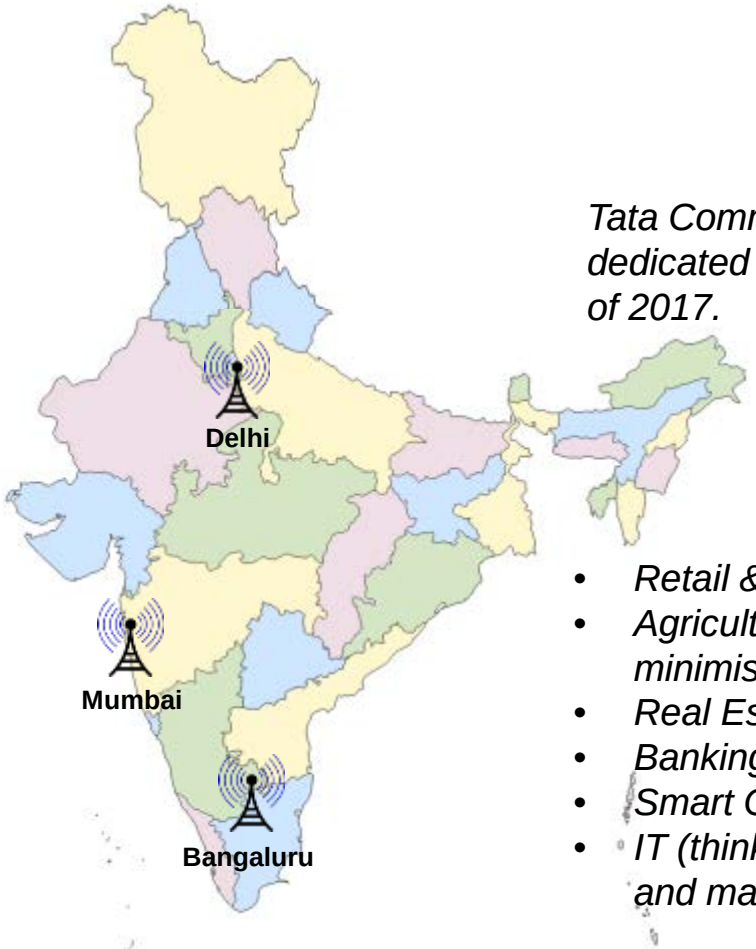
*KotahiNet aims to expand their LoRa® deployment to Auckland, Hamilton, Taranaki, Christchurch, and rural Canterbury over the next year, and go nationwide after that.*

# Deployment – Private & Public
## West Malaysia & Brunei


TM


*Telekom Brunei Berhad (TelBru) is moving forward to facilitate alternative resources of technological assurances with the introduction of long-range, low-powered, low-bandwidth and low-cost sensors.*

# Deployment – Nationwide
## India

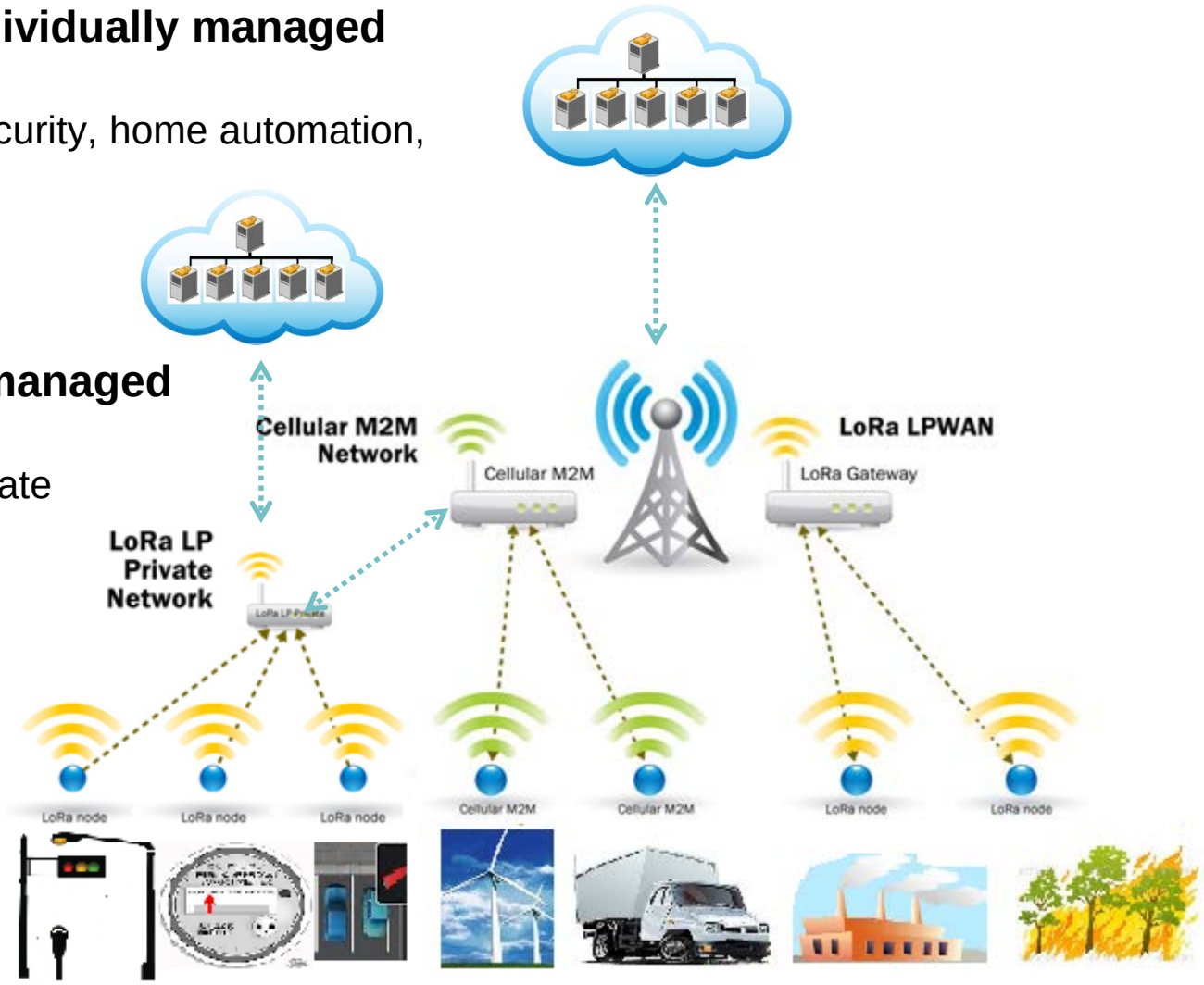



*Tata Communications plans to expand its LoRaWAN™ network dedicated solely for the Internet of Things to 60 cities by the end of 2017.*

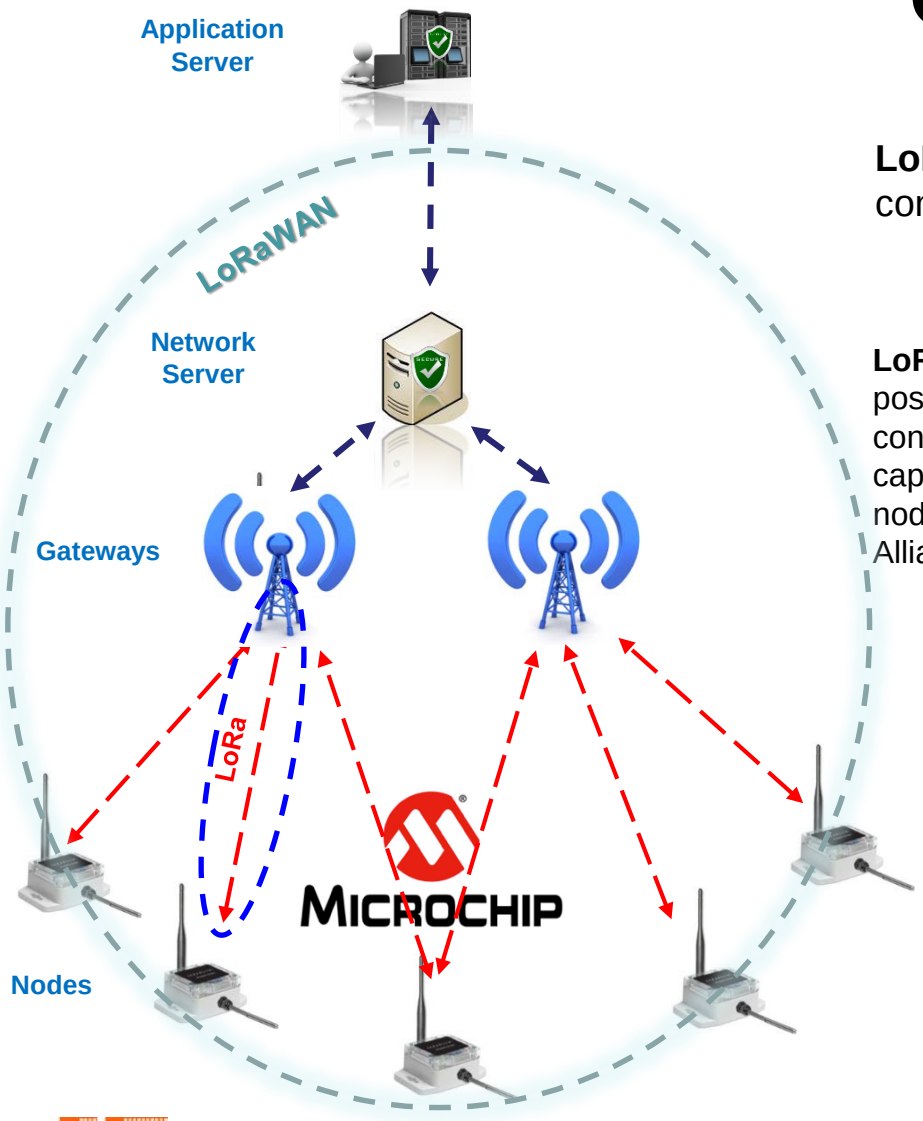
- *Retail & Wholesale Trade (think Logistics, Asset Tracking)*
- *Agriculture (Smart Farming to improve crop yields and minimise pest infestation)*
- *Real Estate (Smart Building and Employee Safety)*
- *Banking & Insurance (think Asset Tracking)*
- *Smart Cities (Street Lightings, Traffic Light)*
- *IT (think the IoT ecosystem to support the design, deployment and management of IoT solutions)*

# LoRaWAN™ Network Deployment

- **Private network- individually managed networks**
  - Smart city, Metering, security, home automation, industrial control
  - Local network
- **Public network- Telecom/operator managed networks**
  - Includes traditional private networks
  - Permits numerous new applications
  - A much more scalable model
  - Nationwide network



# LoRa & LoRaWAN™ Overview




**LoRa** contains only the link layer protocol used in P2P communications between nodes.

**LoRaWAN™** includes the LoRa & network layer together so it is possible to send the information to any Base Station already connected to a Cloud platform. It is the MAC protocol for high capacity, long range, low power, Internet of Things network of LoRa nodes. It is an open LPWAN standard maintained by the LoRa Alliance.

- Star topology in which gateways is a transparent bridge relaying messages between end-devices and a central network server in the backend.
- Gateways are connected to the network server via standard IP connections while end-devices use single-hop wireless communication to one or many gateways.

# LoRaWAN™ Network Protocol
## LoRa® Technology Modulation

**LoRa Modulation is set by the following settings :**

- **Spreading Factor (SF) ;** logarithm in base 2
  - Programmable SF:

    7, 8, 9, 10, 11, 12
  - The higher the SF the more information transmitted per bit; therefore higher processing gain

- **Bandwidth (BW)**
  - Programmable signal BW settings:

    125 kHz, 250 kHz, 500 kHz
  - For a given SF, a narrower BW = increased receive sensitivity; however, increased time on air
- **Forward Error Correction (FEC) Code Rate (CR)**
  - Additional coding rate provides more redundancy to detect errors and correct them

# LoRaWAN™ Network Protocol
# Modulation Settings AS923

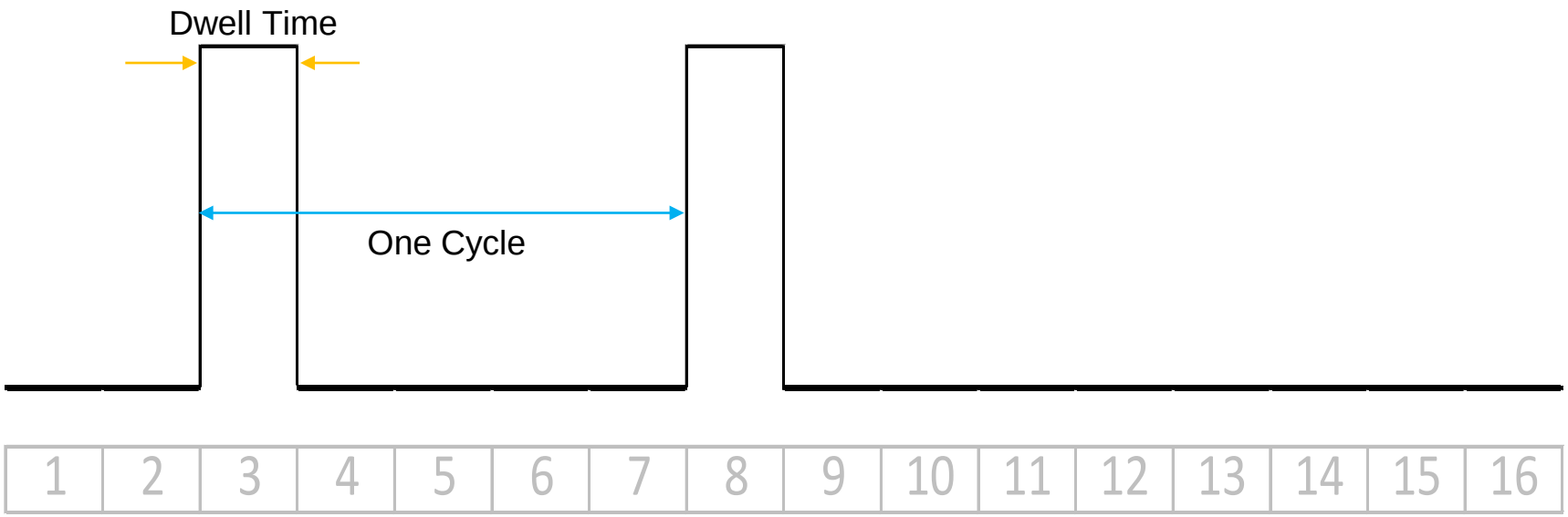
**Longest Distance on LoRa® Modulation**

- **Data Rate (DR) = 0**
  - LoRa® modulation
  - Spreading Factor (SF) = SF10
  - Bandwidth (BW) = 125 kHz
  - Coding Rate (CR) = 4/5

- **Bit Rate = 976 bps**

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR$$

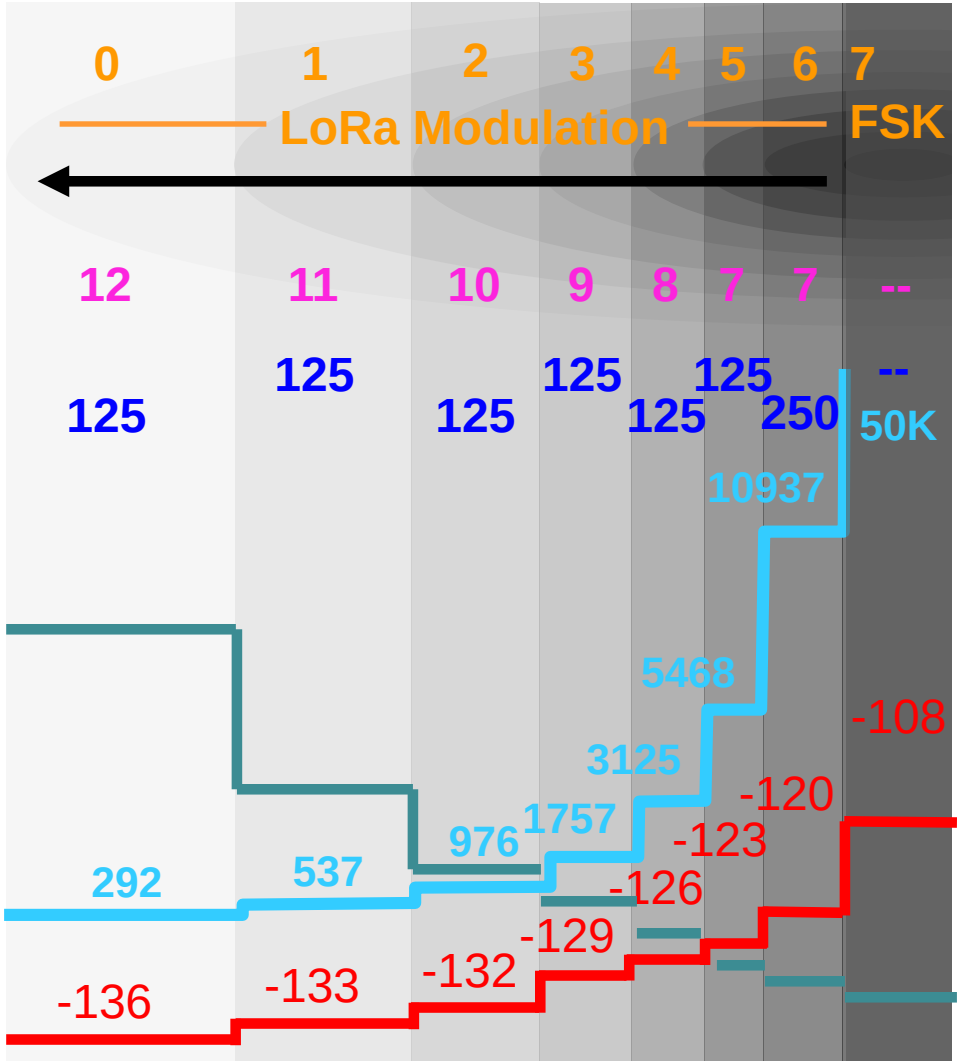
# Duty Cycle & Dwell Time




Duty cycle is the ratio of transmission time to the time of the next transmission.
AS923 Duty Cycle is at 1%.
For every 400ms, the device cannot Tx on the same channel until 40 secs later.

Dwell Time is the time take to transmit Tx. Also known as Time on Air.
AS923 Limits Dwell Time to 400ms

# LoRaWAN™ Network Protocol
## Modulation Settings AS923




Data Rate (DR)

Range

Spreading Factor (SF)

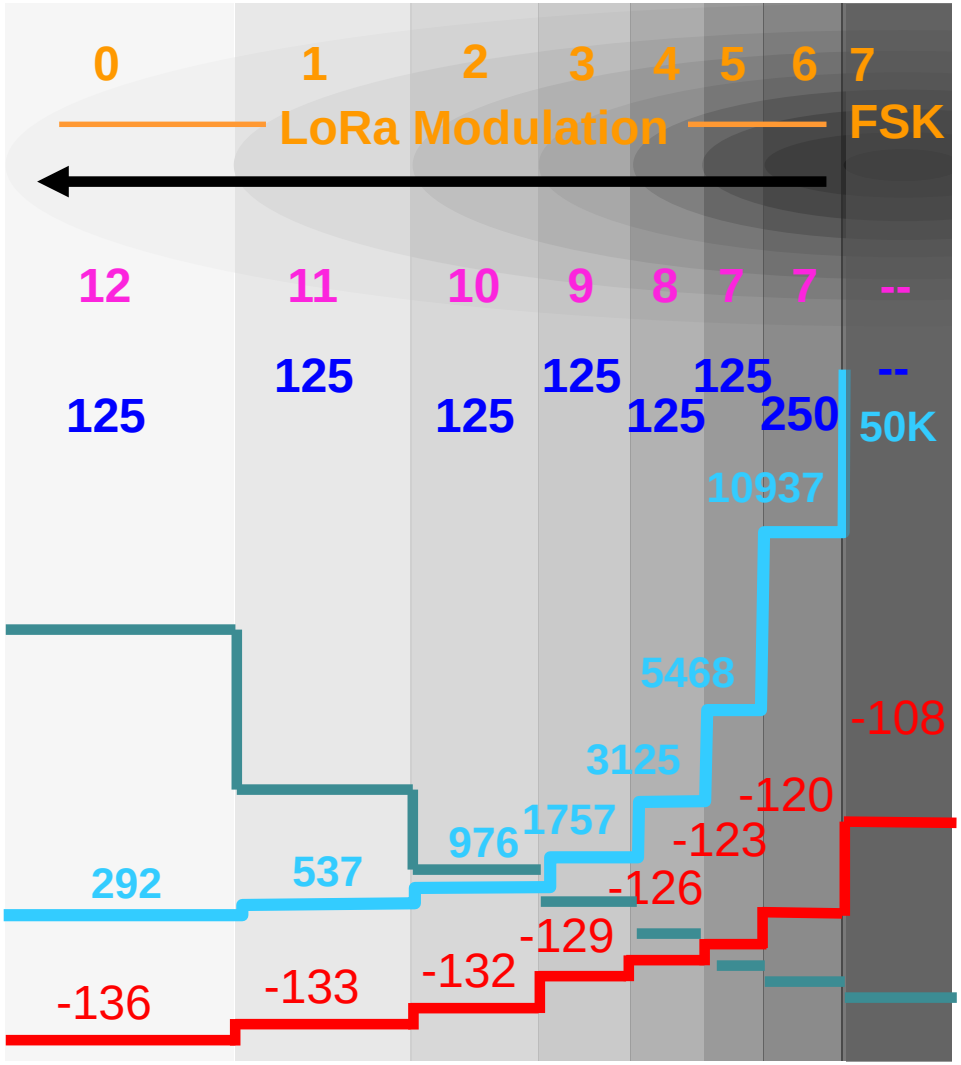
Bandwidth (BW) (kHz)

Bitrate (BR) (bps)

Receive Sensitivity (dBm)

Time-on-air & consumption

# LoRaWAN™ Network Protocol
## Modulation Settings AS923




| DataRate | Configuration | Indicative physical bit rate [bit/s] |
|---|---|---|
| 0 | LoRa: SF12 / 125 kHz | 250 |
| 1 | LoRa: SF11 / 125 kHz | 440 |
| 2 | LoRa: SF10 / 125 kHz | 980 |
| 3 | LoRa: SF9 / 125 kHz | 1760 |
| 4 | LoRa: SF8 / 125 kHz | 3125 |
| 5 | LoRa: SF7 / 125 kHz | 5470 |
| 6 | LoRa: SF7 / 250 kHz | 11000 |
| 7 | FSK: 50 kbps | 50000 |
| 8..15 | RFU | |

# Payload (AS923)

| DataRate | *Uplink MAC Payload Size (M)* | | *Downlink MAC Payload Size (M)* | |
|---|---|---|---|---|
| | UplinkDwellTime = 0 | UplinkDwellTime = 1 | DownlinkDwellTime = 0 | DownlinkDwellTime = 1 |
| 0 | 59 | N/A | 59 | N/A |
| 1 | 59 | N/A | 59 | N/A |
| 2 | 59 | 19 | 59 | 19 |
| 3 | 123 | 61 | 123 | 61 |
| 4 | 230 | 133 | 230 | 133 |
| 5 | 230 | 250 | 230 | 250 |
| 6 | 230 | 250 | 230 | 250 |
| 7 | 230 | 250 | 230 | 250 |
| 8:15 | RFU | | RFU | |

Table 53: AS923 maximum payload size

# Time on Air (AS923)

| Payload (Bytes) | PHY (Bytes) | PHY (Bits) | SF12 | SF11 | SF10 | SF9 | SF8 | SF7 | Spreading Factor |
|---|---|---|---|---|---|---|---|---|---|
| | | | 250 | 440 | 980 | 1760 | 3125 | 5470 | bps |
| **4** | 17 | 136 | 0.544 | 0.309 | 0.139 | 0.077 | 0.044 | 0.025 | Time (sec) |
| **8** | 21 | 168 | 0.672 | 0.382 | 0.171 | 0.095 | 0.054 | 0.031 | |
| **16** | 29 | 232 | 0.928 | 0.527 | 0.237 | 0.132 | 0.074 | 0.042 | |
| **24** | 37 | 296 | 1.184 | 0.673 | 0.302 | 0.168 | 0.095 | 0.054 | |

https://docs.google.com/spreadsheets/d/1QvcKsGeTTPpr9icj4XkKXq4r2zTc2j0gsHLrnplzM3I/edit#gid=0

# LoRaWAN™ Network Protocol
## LoRaWAN Channels

- **License free Sub-GHz Frequencies**
  - Asia AS923 (LoRaWAN102)
  - Network channels can be freely attributed by the network operator
  - 2 mandatory channels that all Network Gateways should constantly receive:

| Modulation | Bandwidth [kHz] | Channel Frequency [MHz] | FSK Bitrate or LoRa DR / Bitrate | Nb Channels | Duty cycle |
|---|---|---|---|---|---|
| LoRa | 125 | 923.2<br>923.4 | DR0 to DR5<br>/ 0.3-5 kbps | 2 | < 1% |

  - End-devices must be capable of at least 16 channels
  - The default JoinReq Data Rate is DR2 (SF10/125KHz), this setting ensures that end-devices are compatible with the 400ms dwell time limitation until the actual dwell time limit is notified to the end-device by the network server via the MAC command “TxParamSetupReq”. (Used by the network server to set the maximum allowed dwell time and Max EIRP of end-device, based on local regulations)

| Modulation | Bandwidth [kHz] | Channel Frequency [MHz] | FSK Bitrate or LoRa DR / Bitrate | Nb Channels | Duty cycle |
|---|---|---|---|---|---|
| LoRa | 125 | 923.2<br>923.4 | DR2 | 2 | < 1% |

# AS923 ISM Band Channel Frequencies

This section applies to regions where the frequencies [923…923.5MHz] are comprised in the ISM band, which is the case for the following countries:

- Brunei [923-925 MHz]
- Cambodia [923-925 MHz]
- Hong Kong [920-925 MHz]
- Indonesia [923-925 MHz]
- Japan [920-928 MHz]
- Laos [923-925 MHz]
- New Zealand [915-928 MHz]
- Singapore [920-925 MHz]
- Taiwan [922-928 MHz]
- Thailand [920-925 MHz]
- Vietnam [920-925 MHz]

# LoRaWAN

- CLASS A
- CLASS B
- CLASS C

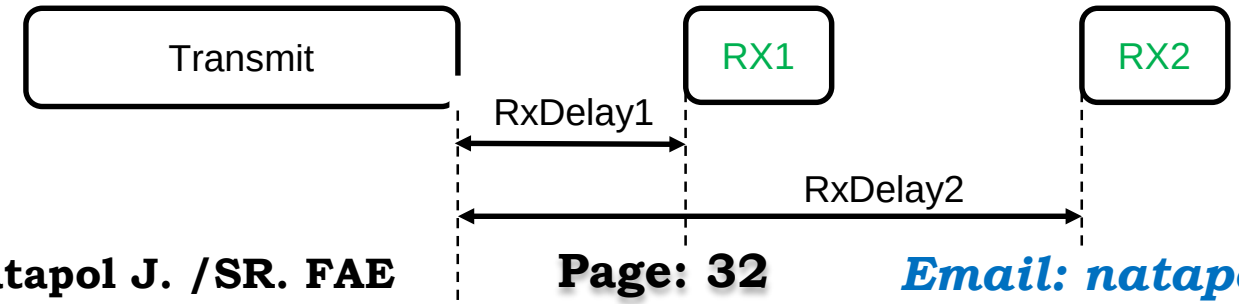
# LoRaWAN™ Network Protocol
## End-Device Classes

- **Each end-device class has different behavior depending on the choice of optimization:**
  - Battery Powered – Class A
  - Low Latency – Class B
  - No Latency – Class C

# LoRaWAN™ Network Protocol

## End-Device Classes

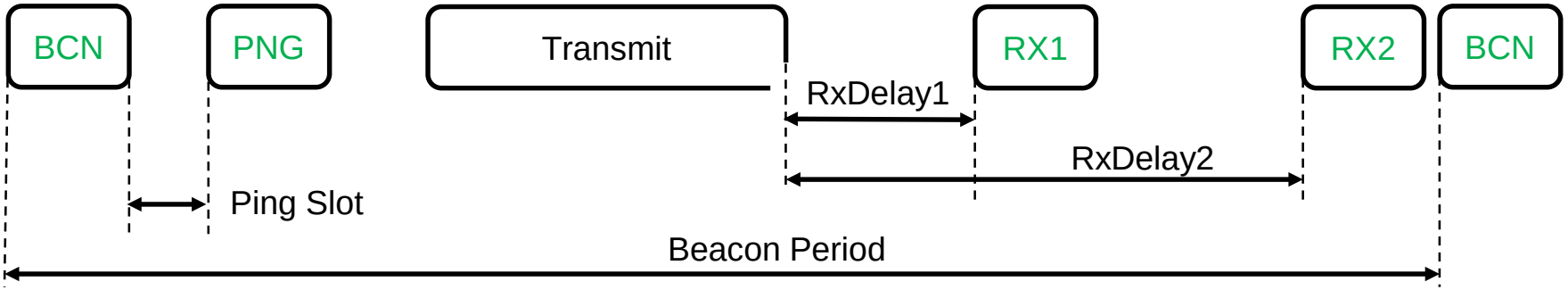
- **Battery Powered – Class A**
  - Bidirectional communications
  - Unicast messages
  - Small payloads
  - Long intervals
  - End-device initiates communication (uplink)
  - Server communicates with end-device (downlink) during predetermined response windows:



# LoRaWAN™ Network Protocol

## End-Device Classes

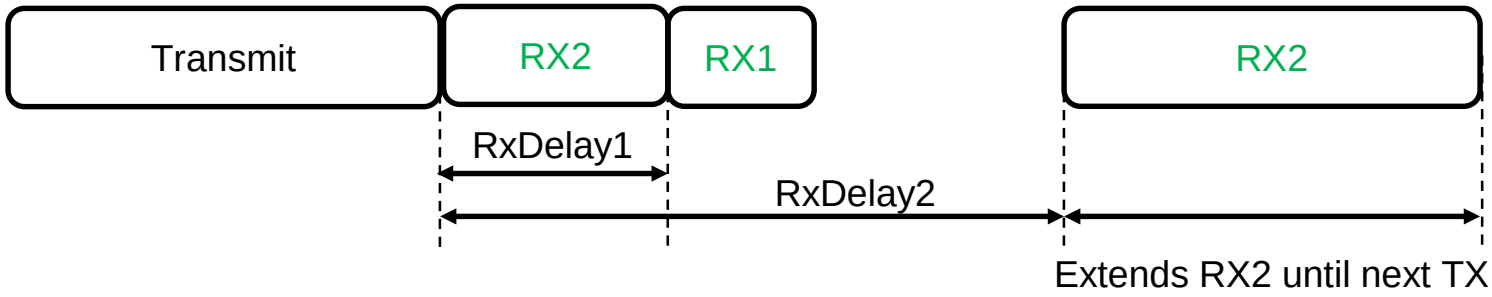
- **Low Latency – Class B**
  - Bidirectional with scheduled receive slots
  - Unicast and Multicast messages
  - Small payloads
  - Long intervals
  - Periodic beacon from gateway
  - Extra receive window (ping slot)
  - Server can initiate transmission at fixed intervals



# LoRaWAN™ Overview

## End-Device Classes

- **No Latency – Class C**
  - Bidirectional communications
  - Unicast and Multicast messages
  - Small payloads
  - Server can initiate transmission at any time
  - End-device is constantly receiving



# LoRaWAN™ Network Protocol

## End-Device Classes

| Device Class | LoRa Class A | LoRa Class B | LoRa Class C |
|---|---|---|---|
| Latency | High Latency | Low Latency | No Latency |
| Communication Direction | Bidirectional communications | Bidirectional with scheduled receive slots | Bidirectional communications |
| Packet Delivery | Unicast messages | Unicast and Multicast messages | Unicast and Multicast messages |
| Payload Size | Small payloads, long intervals | Small payloads, long intervals, Periodic beacon from gateway | Small payloads |
| Communication Initiative | End-device initiates communication (uplink) | Extra receive window (ping slot) | Server can initiate transmission at any time |
| | Server communicates with end-device (downlink) during predetermined response windows | Server can initiate transmission at fixed intervals | End-device is constantly receiving |

# LoRaWAN

* PC Not Included

# R&D Evolution
## Transition to LoRaWAN Ecosystem

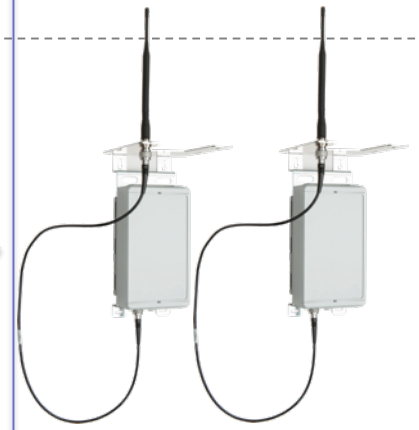
Evaluation

Deployment

Network Server

*

Gateway



kerlink
communication is everything

Devices

Your Microchip-based Sensors

* PC Not Included

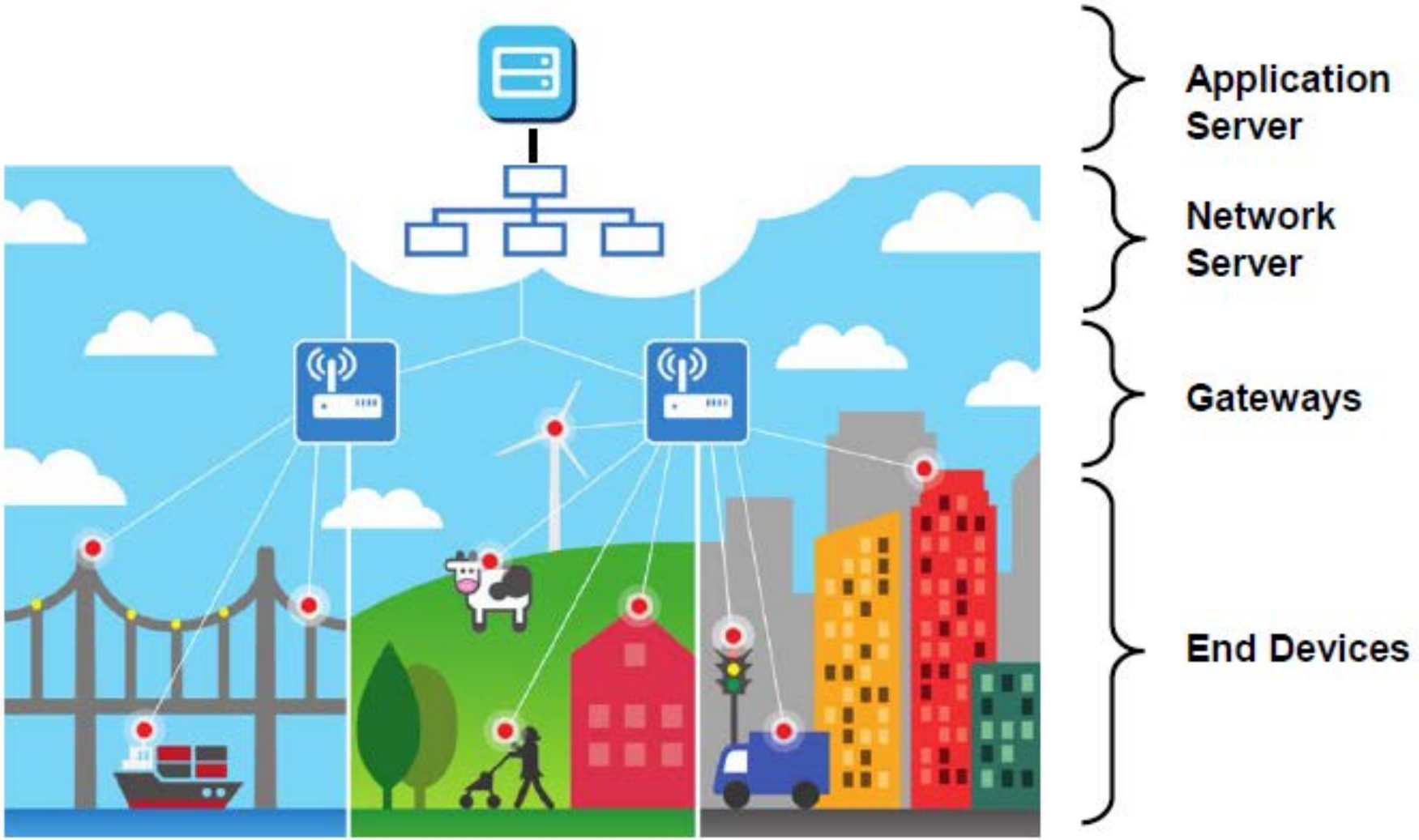
# Specification Status

- **1R0 - Original Release, mid 2015**
  - EU, NA, China bands
  - Certification released Nov 2015
- **v1.0.1 – Minor editorial updates, April 2016**
  - Adds Korea & Australia
  - EU Certification released mid 2016
  - US Certification released Jan 2017
- **v1.0.2 – Separated Regional Parameters, Oct 2016**
  - Accelerates regional releases outside of 60day IPR review
  - Adds multiple regional channel plans
  - Certification due soon, inc class C
- **V1.1 – Major Updates (in IPR review now)**
  - Standardises back-end interfaces for roaming & join servers
  - Formalises class B

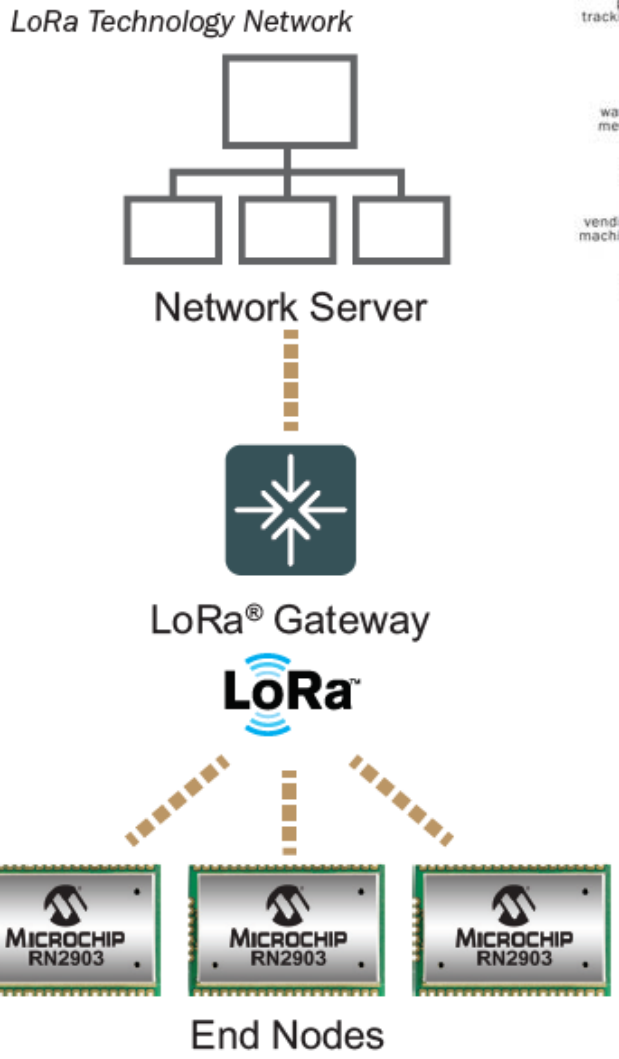
# Other Ecosystem Developments

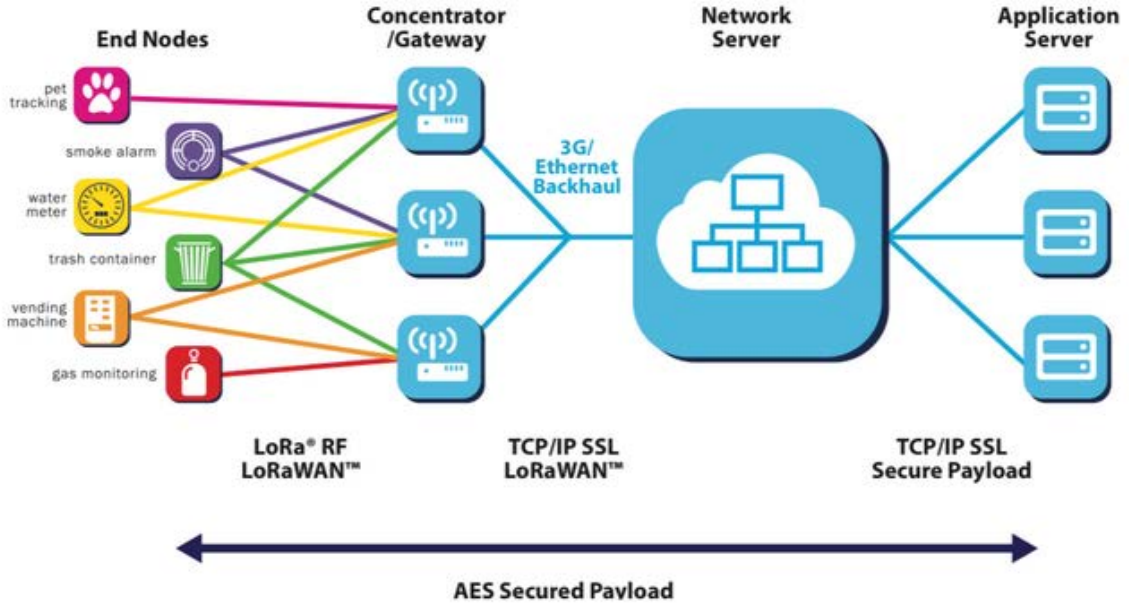
- Geo-location feature is becoming available
- New players for join servers & data brokers
- Multiple demos of firmware OTA updates
  - Relies on spec v1.0.1 with class C & multi-cast
- Additional PHY layers being discussed
- Most collaborations are Cortex-focused
- PoCs & trails are moving to scale

# LORA Technology Network



# LORA Technology Network




LoRaWAN ขอให้แบ่งส่วนประกอบในการเชื่อมต่อออกเป็น 4 ส่วนได้แก่

1. End-Devices
2. Concentrator/Gateway
3. Network Server
4. Application Server

# RN2903 Module

# Microchip LORA module

LoRa™ Long-Range Sub-GHz Module
(Part # RN2903)

## Parametrics

| Name | Value |
|---|---|
| Type | Sub-GHz |
| Output Power (dBm) | 18.50 (70.79 mWatts) |
| Host Interface | UART |
| Pin Count | 47 |
| Package | Surface mount module |
| IO Pins | 14 |
| RF Module | Yes |
| RF Transceiver | Yes |
| Operating Temperature Range | -40 to 85 |
| Frequency Range | 915 MHz , support 923MHz |
| Input Sensitivity (mVpp) | -146 |
| Rx Input Sensitivity (dB) | -146 |
| TX Current Consumption | 124 mA (max) |
| RX Current Consumption | 13.5 mA |

# Sub-GHz 915 MHz long range LoRa Module Support in Thailand 923MHz

LoRa™ Long-Range Sub-GHz Module
(Part # RN2903)

**NBTC (กสทช): 920-925MHz.**

สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์และกิจการโทรคมนาคมแห่งชาติ (กสทช)

## Features:

- On-board LoRaWAN™ Class A protocol stack
- ASCII command interface over UART
- Compact form factor 17.8 x 26.7 x 3 mm
- Castellated SMT pads for easy and reliable PCB mounting
- Device Firmware Upgrade (DFU) over UART
- 14 GPIO for control, status, and ADC

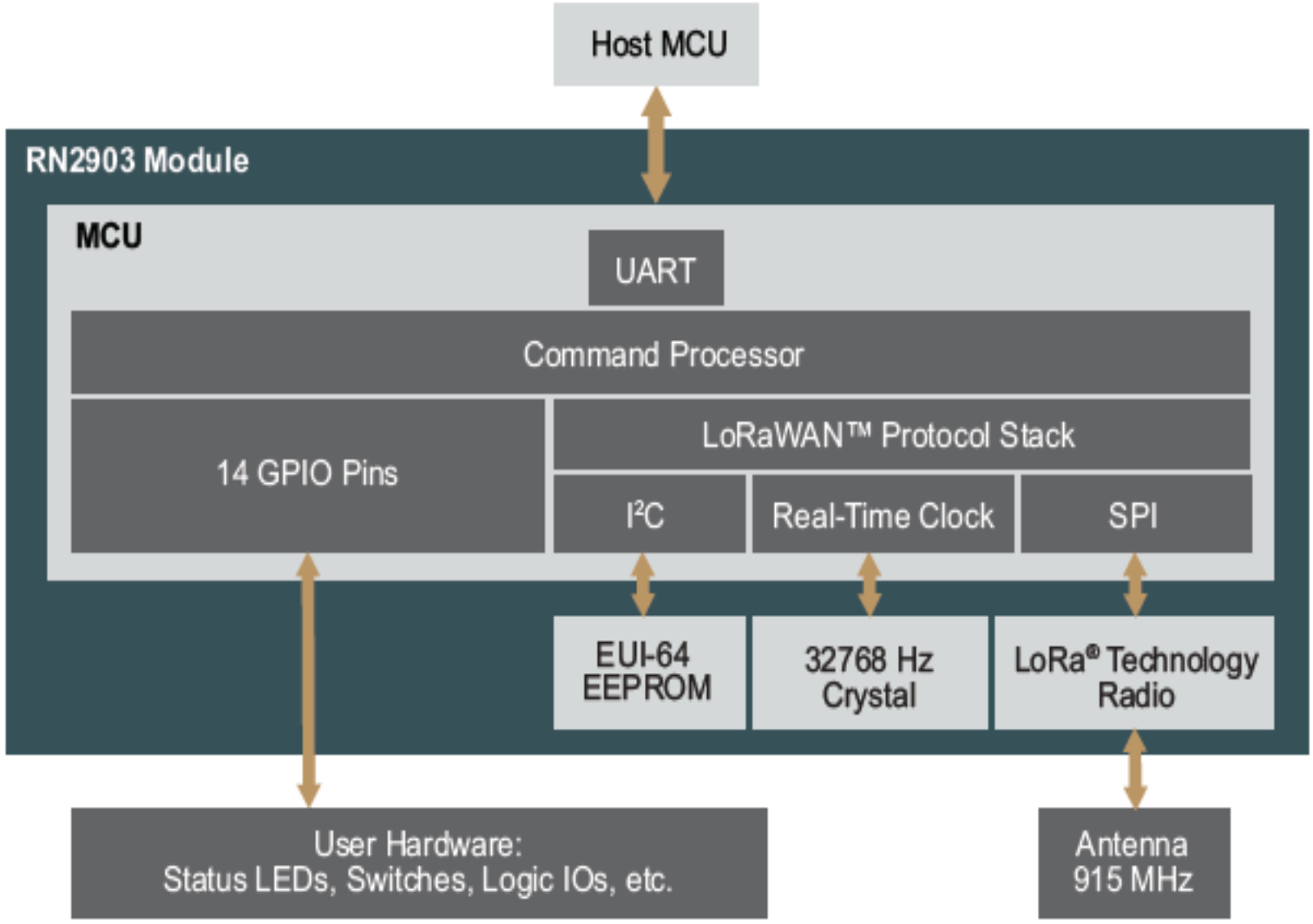
# Microchip LORA module


LoRa™ Long-Range Sub-GHz Module
(Part # RN2903)

## Additional Features

- On-board LoRaWAN™ Class A protocol stack
- ASCII command interface over UART
- Compact form factor 17.8 x 26.7 x 3 mm
- Castellated SMT pads for easy and reliable PCB mounting
- Device Firmware Upgrade (DFU) over UART
- 14 GPIO for control, status, and ADC
- Highly integrated module with MCU, crystal, EUI-64 Node Identity Serial EEPROM, Radio transceiver with analog front end, and matching circuitry
- Environmentally friendly, RoHS compliant
- FCC and IC Certified

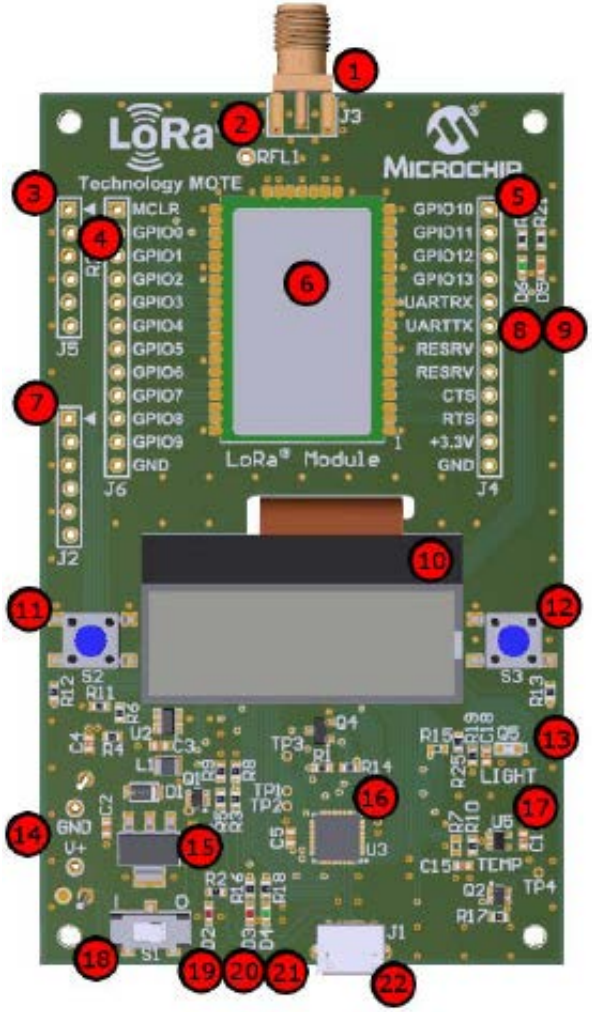
# Block Diagram



Features | Package Contents

# ES P/N: 0035-2040-3
# Product P/N: DM164139

- 915 MHz High-Frequency SMA Connector
- USB Mini-B Connector
- PIC18LF45K50 8-bit MCU
- Mote ICSP Programming
- LCD Display
- S1 & S2 Switches (for Menu Navigation)
- Ambient Light Sensor
- Linear Active Thermistor (MCP9700T)
- LDO Regulator (MCP1825S)
- Descriptive LEDs, (2) Controlled by PIC18, (2) Controlled by Module
- (2) AAA Battery Pack
- Battery Power Switch
- Alternative Power Supply Through Hole Connectors



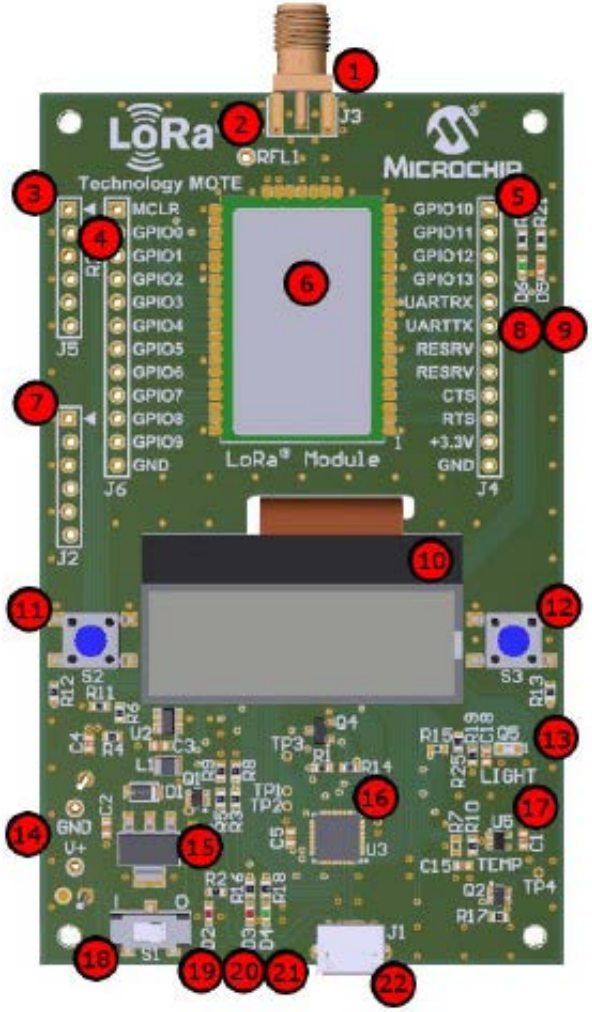

Top

Support AS923

## FEATURES

The LoRa® Mote has the following features, as represented in Figure

1. 868/915 MHz High-Frequency SMA Connector
2. 433 MHz Low-Frequency Antenna Point
3. RN Module ICSP™ Programming Through Holes
4. Module Breakout Header 1
5. Module Breakout Header 2
6. Microchip LoRa® Module
7. Mote ICSP Programming Through Holes
8. Green LED controlled by RN Module GPIO10

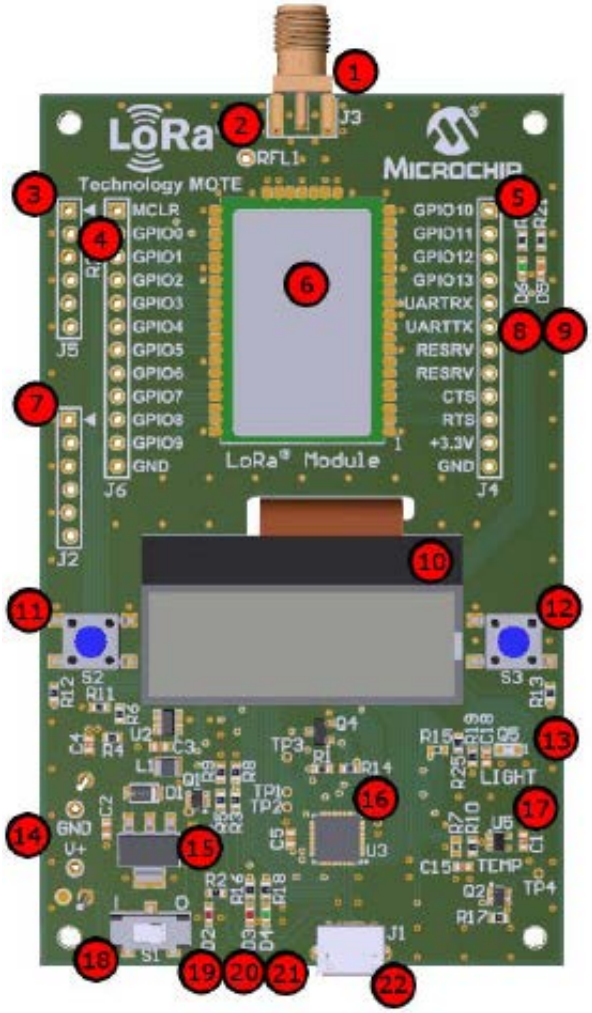



Top

Support AS923

## FEATURES

9. Orange LED controlled by RN Module GPIO11
10. Backlight LCD Display; 1.2 Inch 128 X 32 Dot Matrix
11. S1 Switch (Navigation)
12. S2 Switch (Selection)
13. Everlight (ALS-PT19-315C) Ambient Light Sensor
14. Alternative Power Supply Through Hole Connectors
15. MCP1825S – LDO Regulator

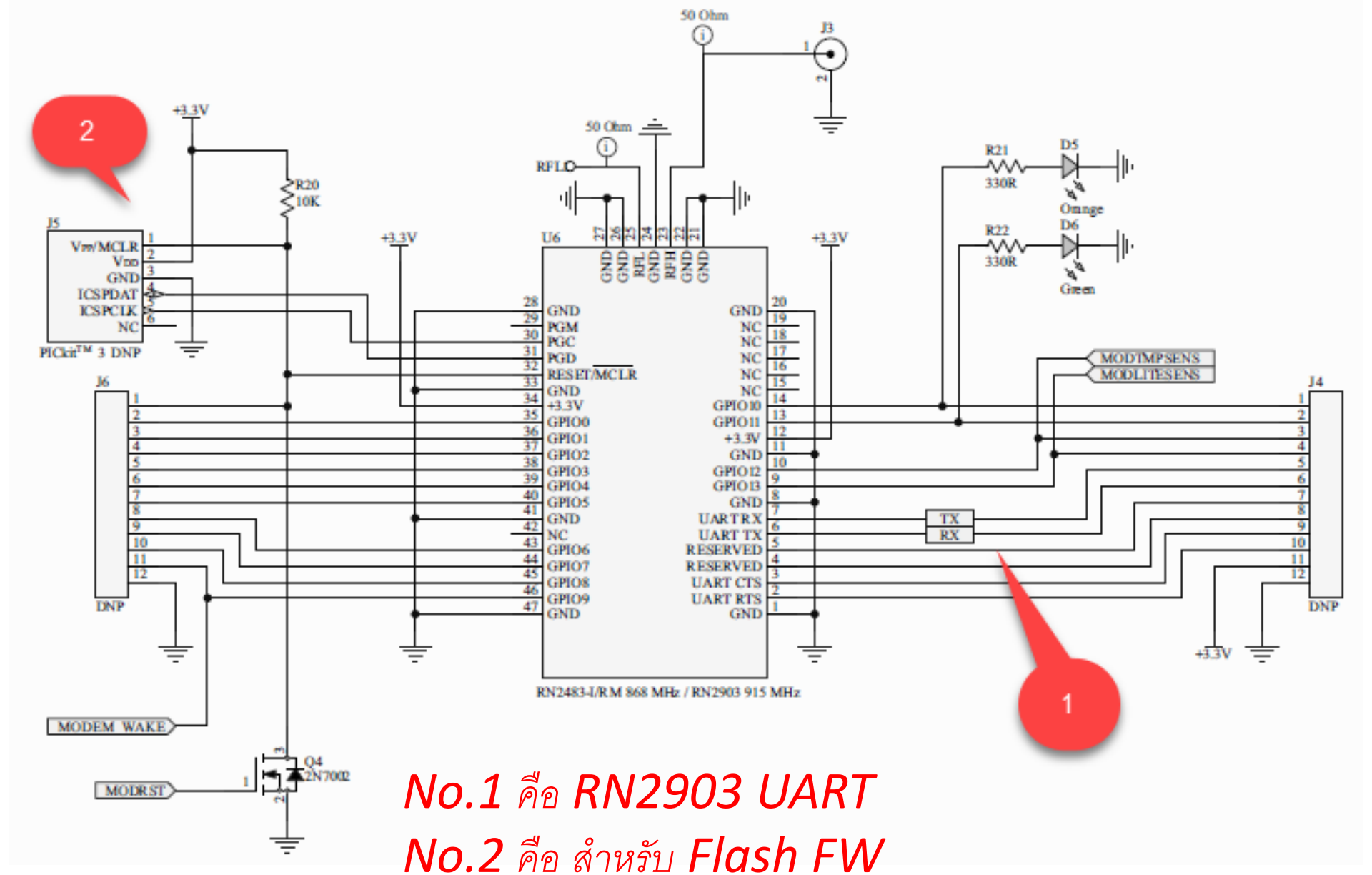



Top

Support AS923

## FEATURES

16. PIC18LF45K50 8-bit USB XLP Flash 32Kb MCU
17. MCP9700 – Linear Active Thermistor
18. Battery Power Switch
19. Mote Power Indicator Red LED
20. Red LED controlled by MCU pin RA7 (28)
21. Green LED controlled by MCU pin RA6 (29)
22. USB Micro-B Connector
23. LCD Ribbon Connector
24. LCD Backlight Power Connector
25. (2) AAA Battery Pack

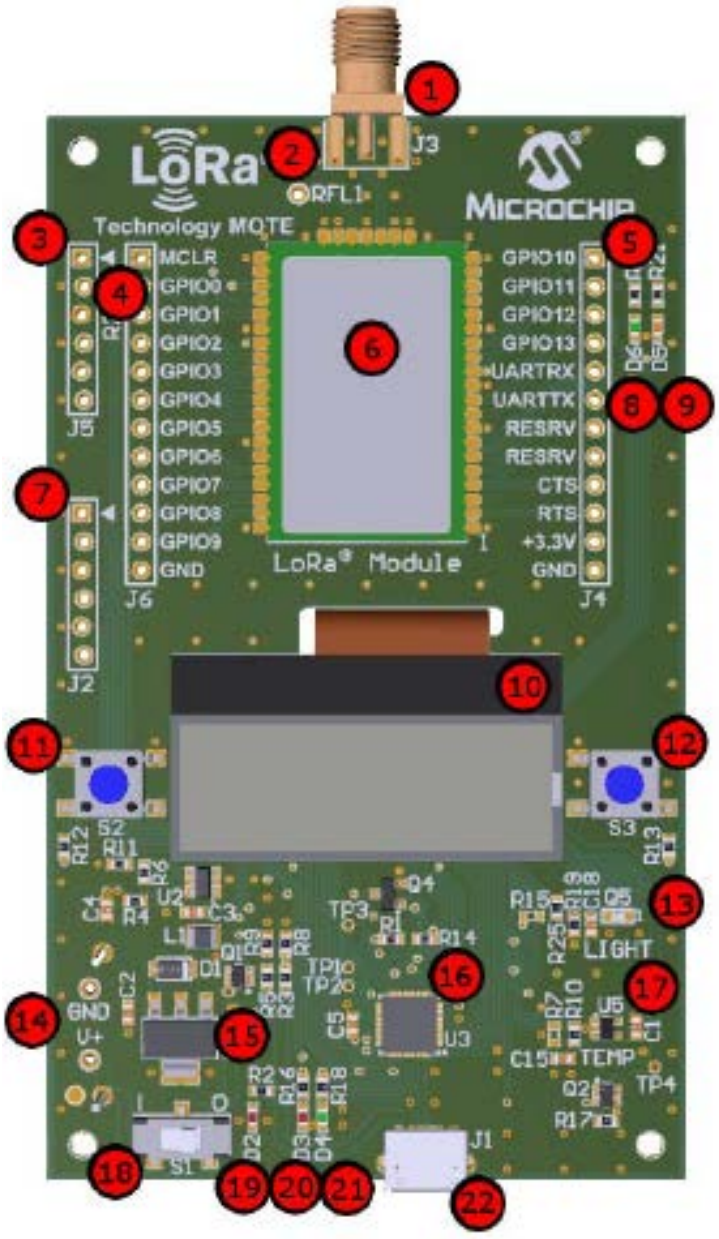



*No.1 คือ RN2903 UART*

*No.2 คือ สำหรับ Flash FW*

TABLE 2-3: MODULE JUMPER CONNECTIONS

| Signal Name | Description | Module Pin Connection | Mote Function |
|---|---|---|---|
| $\overline{MCLR}$ | Reset | 32 | ICSP™ Programmer; Connected to MODRST (RB5, 13) |
| GPIO0 | General Purpose I/O | 35 | Unused |
| GPIO1 | General Purpose I/O | 36 | Unused |
| GPIO2 | General Purpose I/O | 37 | Unused |
| GPIO3 | General Purpose I/O | 38 | Unused |
| GPIO4 | General Purpose I/O | 39 | Unused |
| GPIO5 | General Purpose I/O | 40 | Unused |
| GPIO6 | General Purpose I/O | 43 | Unused |
| GPIO7 | General Purpose I/O | 44 | Unused |
| GPIO8 | General Purpose I/O | 45 | Unused |
| GPIO9 | General Purpose I/O | 46 | MODEM_WAKE (RC2, 32) |
| GPIO10 | General Purpose I/O | 14 | D5 – Orange LED |
| GPIO11 | General Purpose I/O | 13 | D6 – Green LED |
| GPIO12 | General Purpose I/O | 10 | Connected to Temperature Sensor; MODTMPSENS (Analog, ADC) |
| GPIO13 | General Purpose I/O | 9 | Connected to Ambient Light Sensor; MODLITESENS (Analog, ADC) |
| UARTRX | Module Communication | 7 | PIC® MCU TX (RC6, Pin 40) |
| UARTTX | Module Communication | 6 | PIC® MCU RX (RC7, Pin 1) |
| CTS | Module Communication | 3 | Unused |
| RTS | Module Communication | 2 | Unused |
| +3.3V | Power Source | 34, 12 | +3.3V Rail |
| GND | Ground Reference | 1,8,11,20,21,22 ,24,26,27, 28,33,41,47 | Ground |



# ค่าย่านความถี่แต่ละประเทศ / Frequency on country.

### 2.7.2 AS923 ISM Band channel frequencies

This section applies to regions where the frequencies [923…923.5MHz] are comprised in the ISM band, which is the case for the following countries:

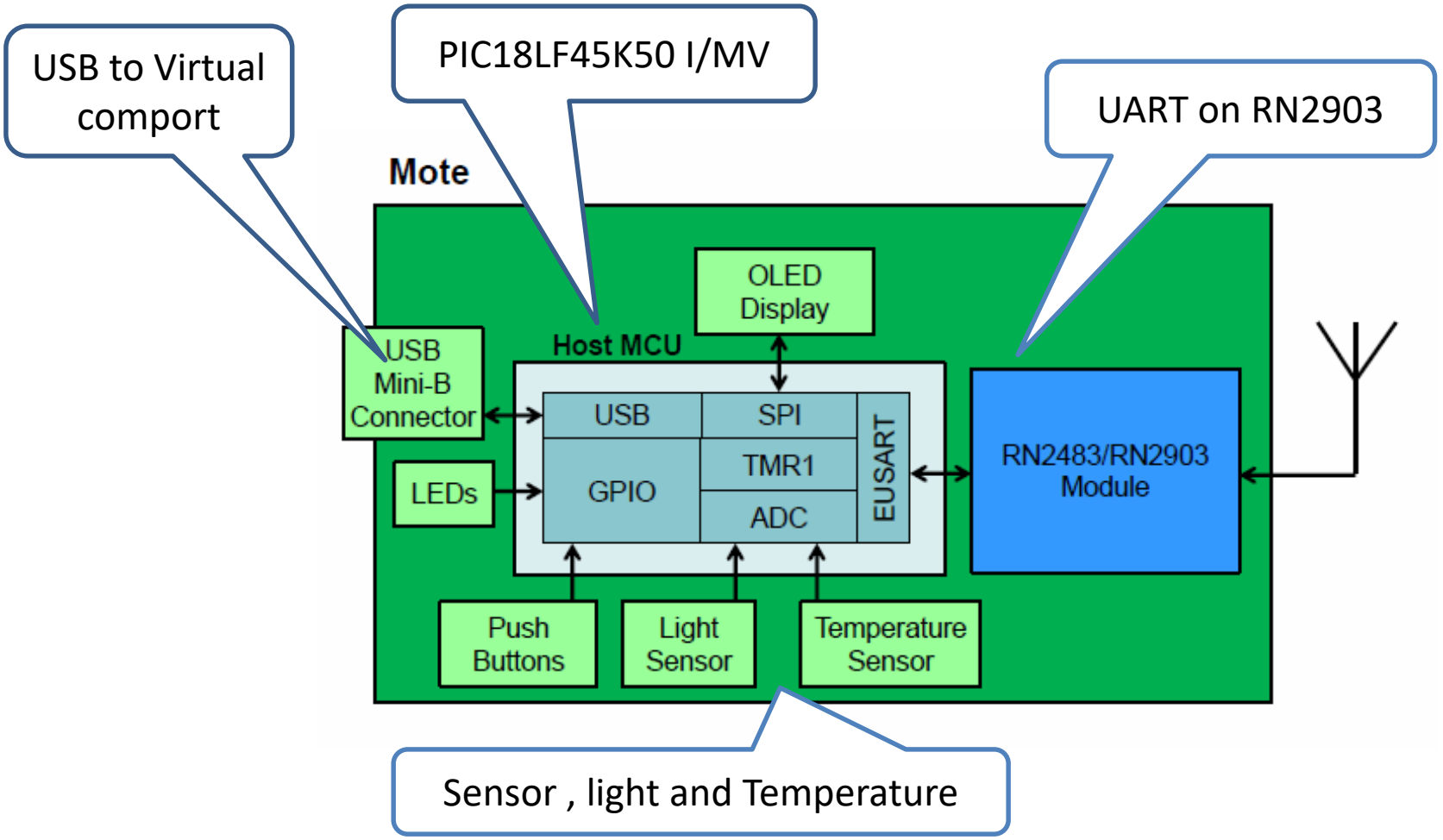
- Brunei [923-925 MHz]
- Cambodia [923-925 MHz]
- Indonesia [923-925 MHz]
- Japan [920-928 MHz]
- Laos [923-925 MHz]
- New Zealand [915-928 MHz]
- Singapore [920-925 MHz]
- Taiwan [922-928 MHz]
- Thailand [920-925 MHz]
- Vietnam [920-925 MHz]

| Modulation | Bandwidth [kHz] | Channel Frequency [MHz] | FSK Bitrate or LoRa DR / Bitrate | Nb Channels | Duty cycle |
|---|---|---|---|---|---|
| LoRa | 125 | 923.20<br>923.40 | DR0 to DR5 / 0.3-5 kbps | 2 | < 1% |

Table 48: AS923 default channels

# Microchip LoRa Mote board BASIC and Overview

# Microchip LoRa Mote board



# Command Syntax

- Key word(s) issued, followed by optional parameter(s)
- Separated by space Character
- Key Word(s) Case Sensitive
- Parameter(s) Case Insensitive
- CR+LF Command Delimiter

- **Command Request example:**

< mac set devaddr 048E436e\r\n

- **Command Reply example:**

> ok\r\n

# UART INTERFACE

All of the RN2903 module's settings and commands are transmitted over UART using
the ASCII interface.

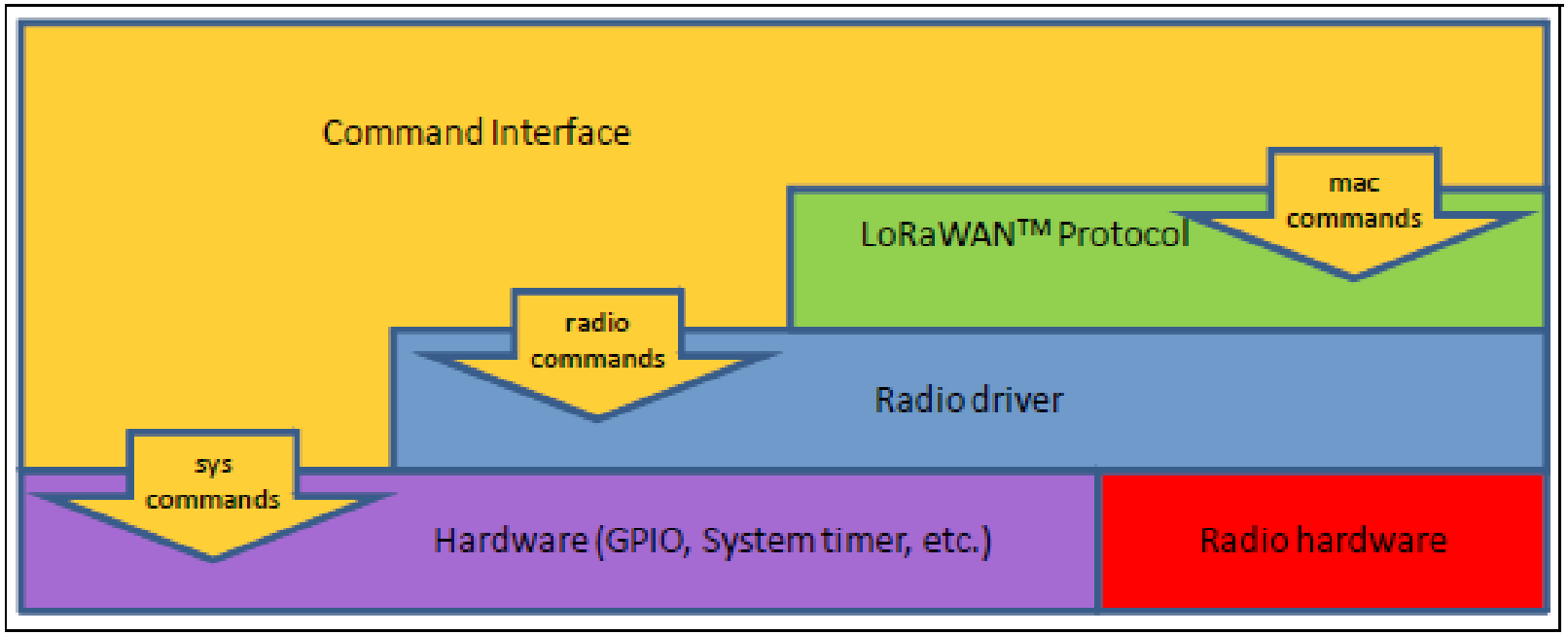
All commands need to be terminated with <CR><LF> and any replies they generate willalso be terminated by the same sequence.

**The default settings for the UART interface are 57600 bps, 8 bits, no parity, 1 Stop bit, no flow control.**

MICROCHIP

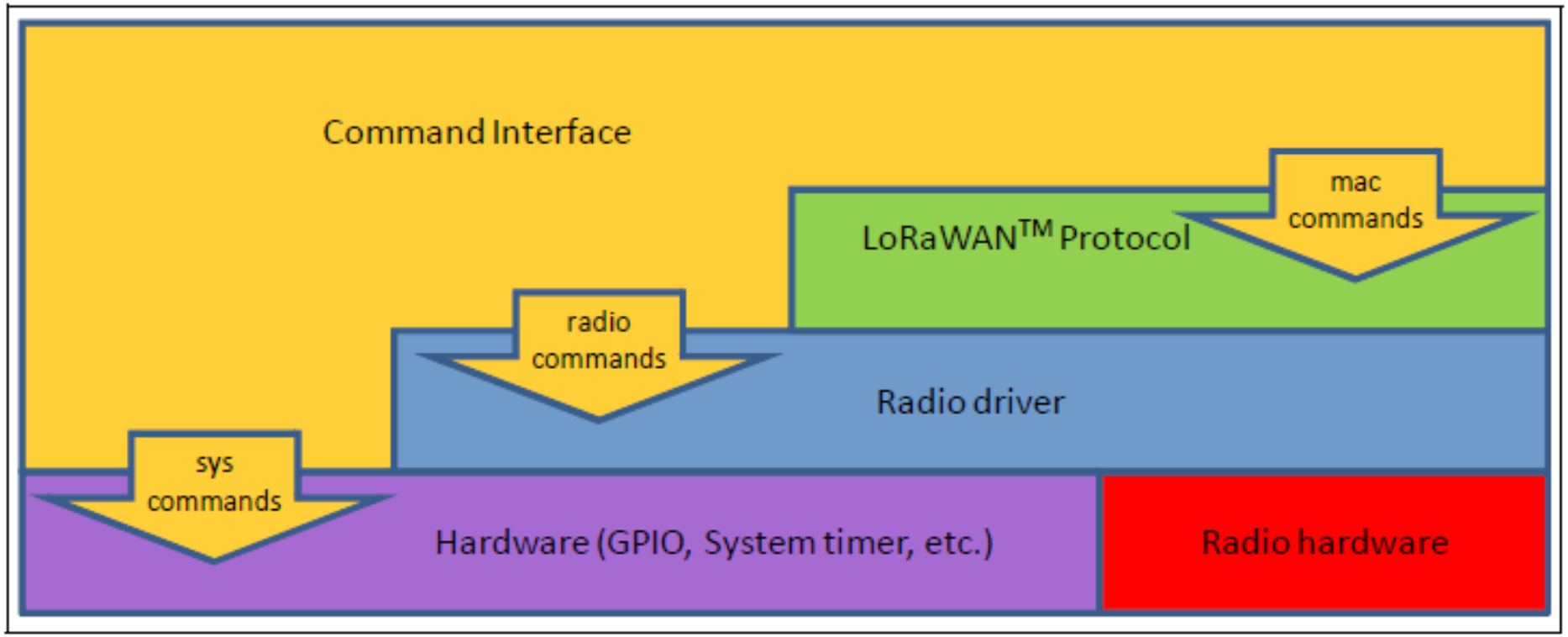
# RN2903 LoRa™ Technology Module Command Reference User's Guide

## RN2903 COMMAND INTERFACE (YELLOW) AND ITS RELATIONSHIP TO THE MODULE'S INTERNAL COMPONENTS



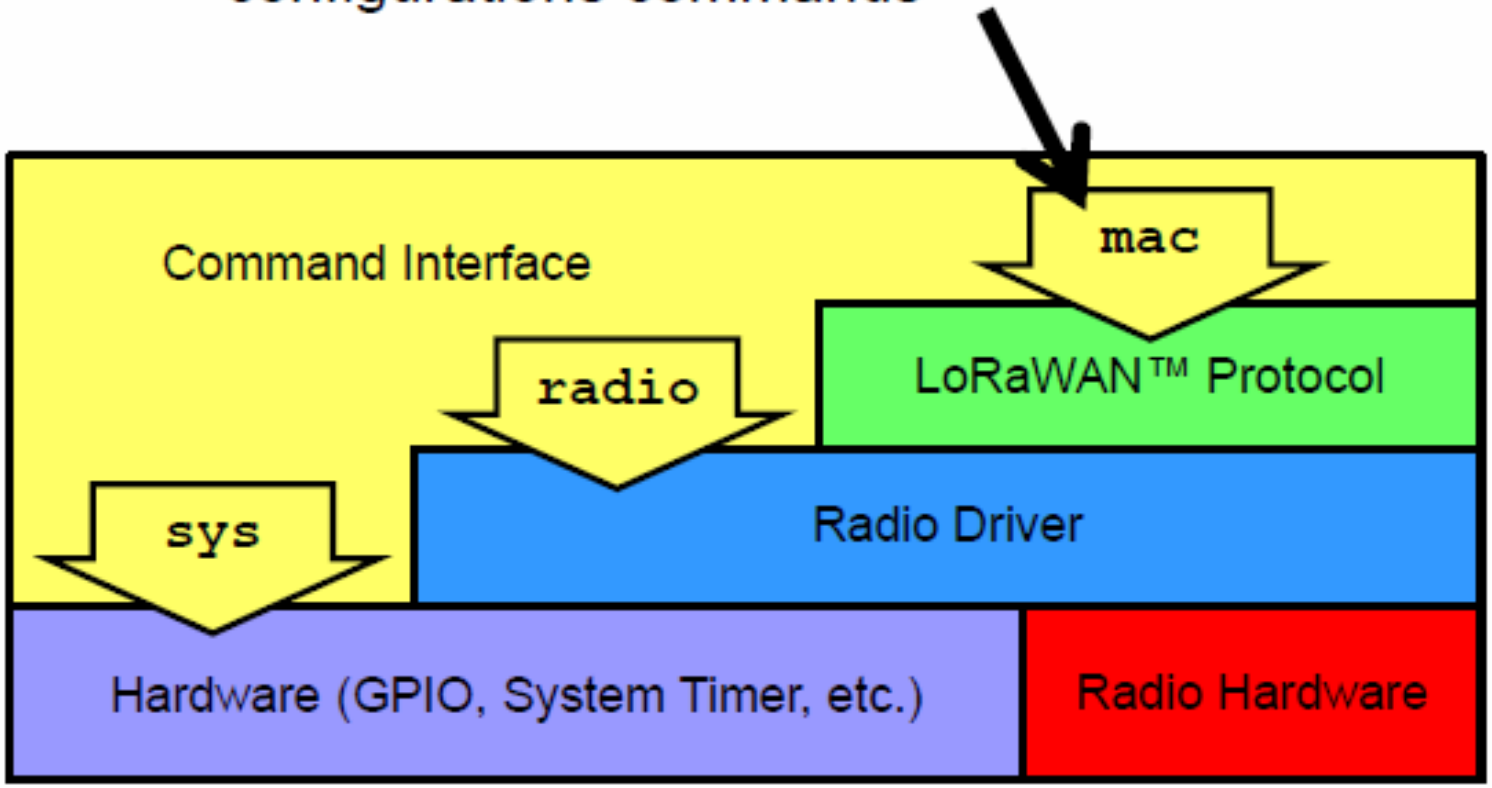
- LoRaWAN configuration and control, using the `mac` group of commands
- Radio configuration and control, using the `radio` group of commands
- Other module functions, using the `sys` group of commands

**FIGURE 1-2: RN2903 COMMAND INTERFACE (YELLOW) AND ITS RELATIONSHIP TO THE MODULE'S INTERNAL COMPONENTS**



# Command Interface

`mac` : Issues LoRaWAN™ Class A protocol network communication behaviors, actions and configurations commands



`mac` : Issues LoRaWAN™ Class A protocol network communication behaviors, actions and configurations commands

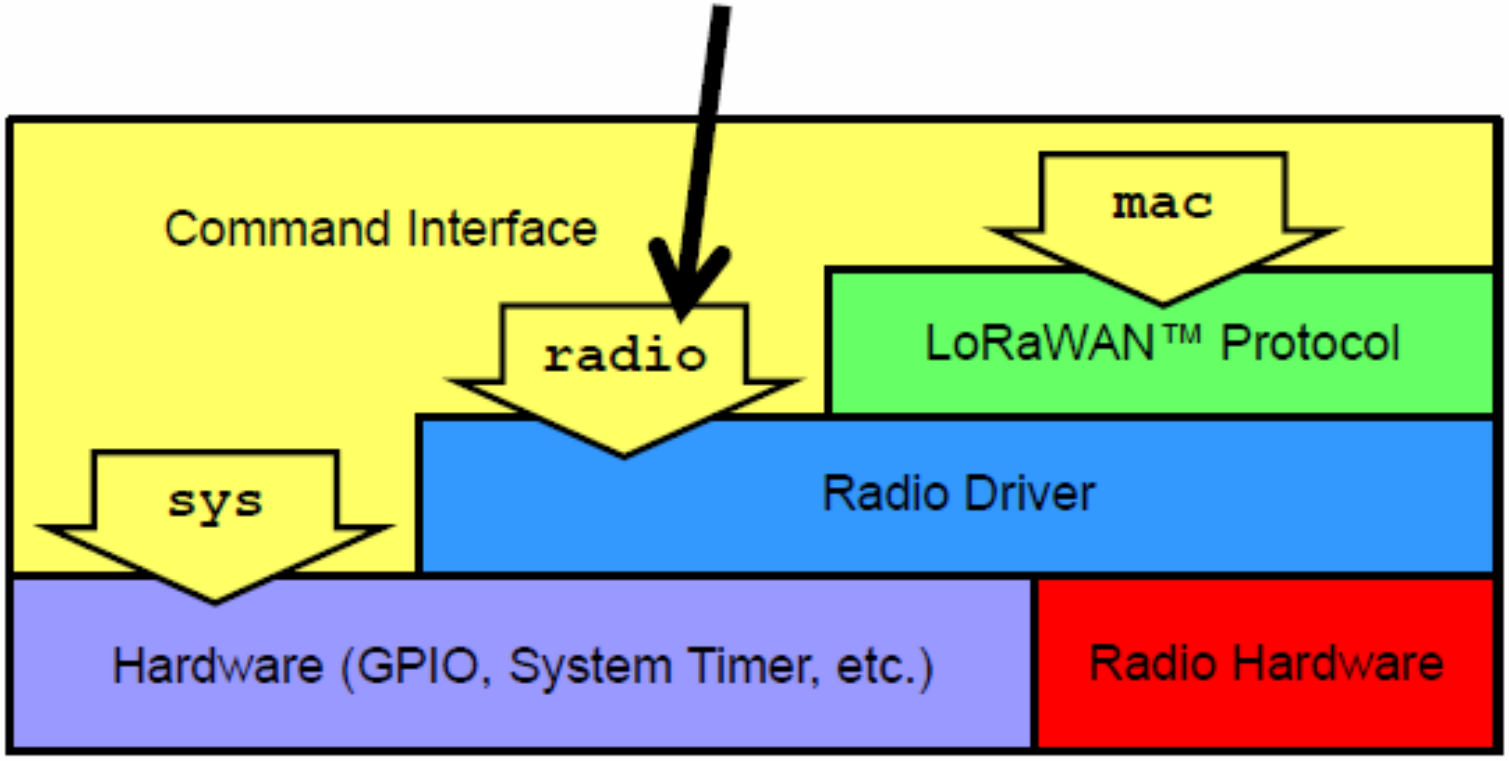
| Parameter | Description |
|---|---|
| `reset` | Resets the RN2483 module to a specific frequency band. |
| `tx` | Sends the data string on a specified port number and sets default values for most of the LoRaWAN parameters. |
| `join` | Informs the RN2483 module to join the configured network. |
| `save` | Saves LoRaWAN Class A configuration parameters to the user EEPROM. |
| `forceENABLE` | Enables the RN2483 module after the LoRaWAN network server commanded the end device to become silent immediately. |
| `pause` | Pauses LoRaWAN stack functionality to allow transceiver (radio) configuration. |
| `resume` | Restores the LoRaWAN stack functionality. |
| `set` | Accesses and modifies specific MAC related parameters. |
| `get` | Reads back current MAC related parameters from the module. |

```
< mac set devaddr 048E436E
> ok


< mac join abp
> ok
> accepted
```

# Command Interface

`radio` : Issues radio specific configurations, directly accessing and updating the transceiver setup



## `radio` : Issues radio specific configurations, directly accessing and updating the transceiver setup

| Parameter | Description |
|---|---|
| `rx` | This command configures the radio to receive simple radio packets according to prior configuration settings. |
| `tx` | This command configures a simple radio packet transmission according to prior configuration settings. |
| `cw` | This command will put the module into a Continuous Wave (cw) Transmission for system tuning or certification use. |
| `set` | This command allows modification to the radio setting directly. This command allows for the user to change the method of radio operation within module type band limits. |
| `get` | This command grants the ability to read out radio settings as they are currently configured. |

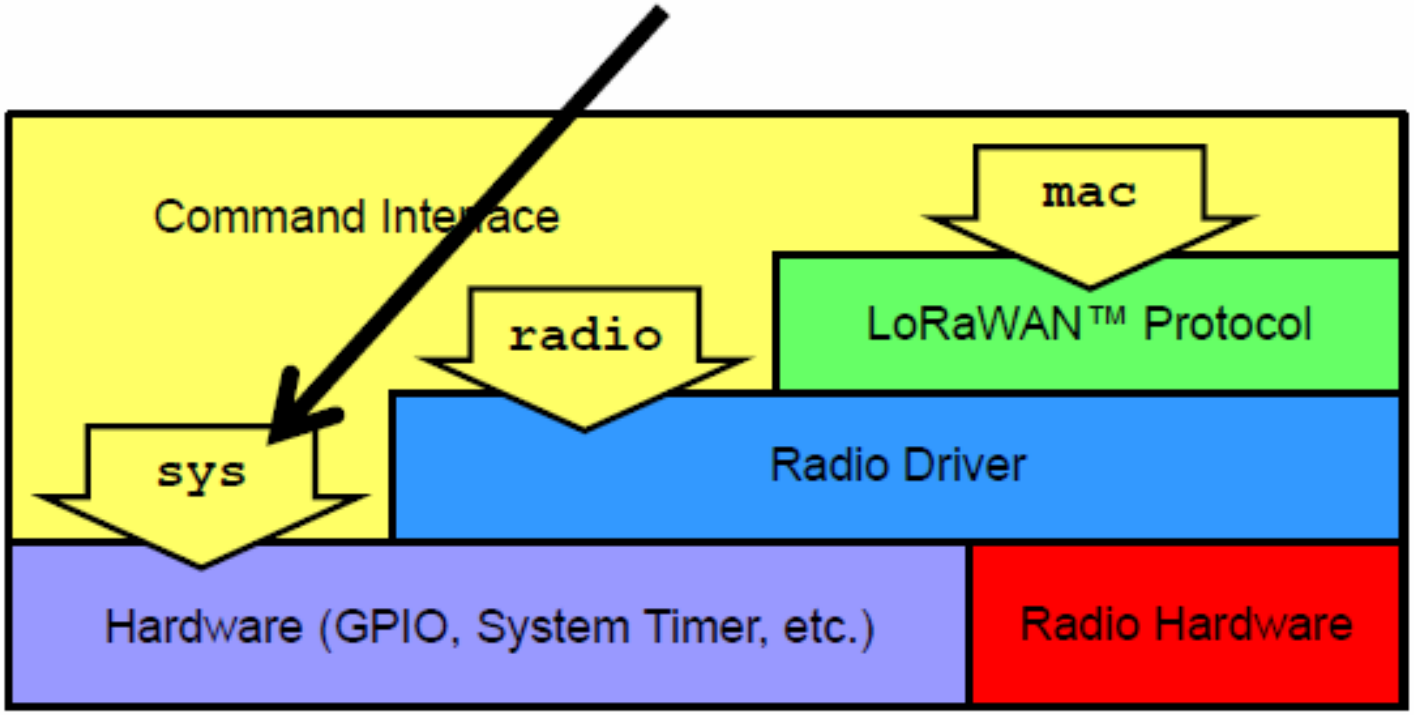
**Note 1:** The `mac pause` command must be called before any radio transmission or reception, even if no MAC operations have been initiated before.

```
< radio cw on
> ok

< radio get mod
> lora
```

# Command Interface

**sys :** Issues system level behavior actions, gathers status information on the firmware and hardware version, or accesses the module user EEPROM memory

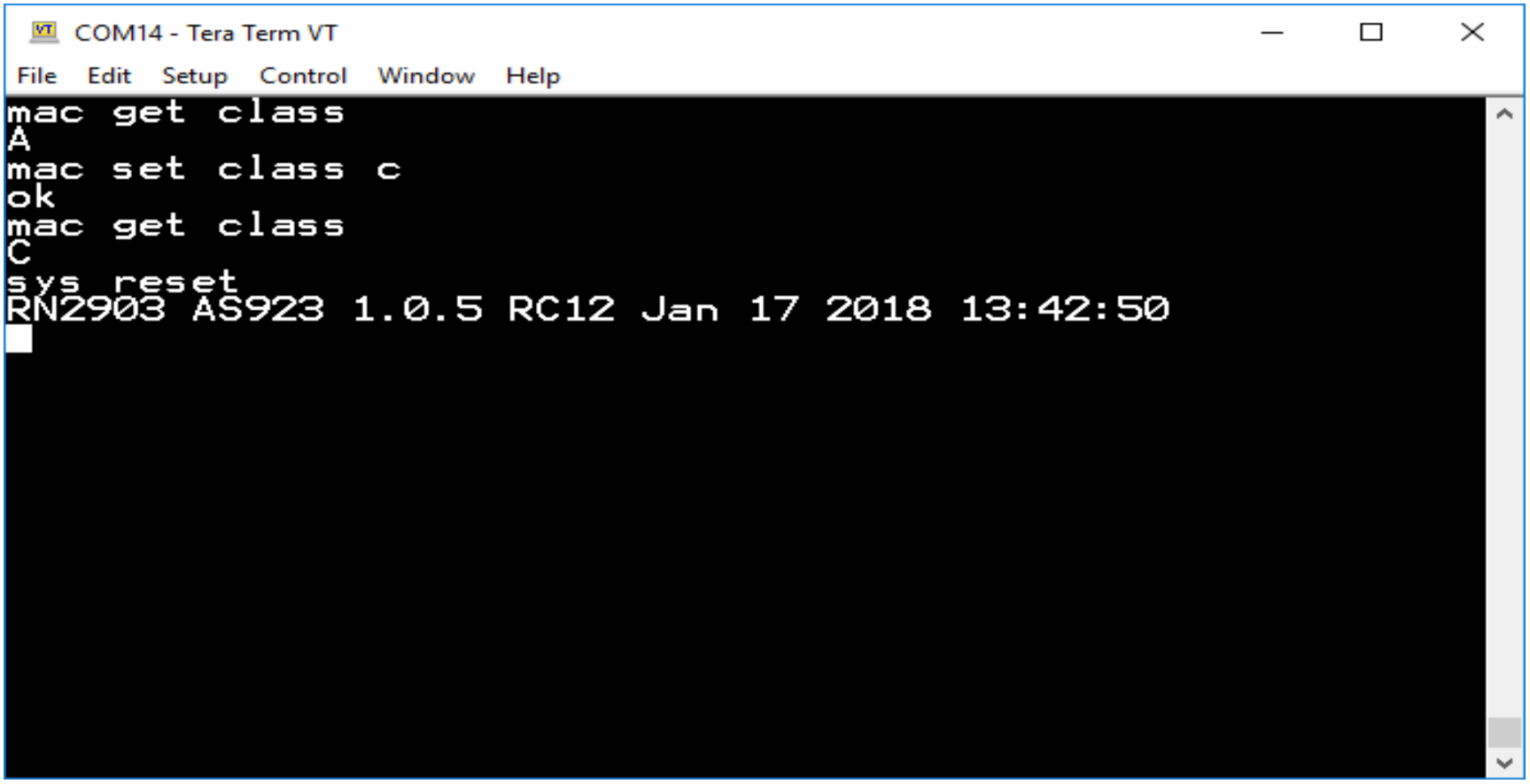


**sys** : Issues system level behavior actions, gathers status information on the firmware and hardware version, or accesses the module user EEPROM memory

| Parameter | Description |
|---|---|
| sleep | Puts the system in Sleep for a finite number of milliseconds. |
| reset | Resets and restarts the RN2483 module. |
| eraseFW | Deletes the current RN2483 module application firmware and prepares it for firmware upgrade. The RN2483 module bootloader is ready to receive new firmware. |
| factoryRESET | Resets the RN2483 module's configuration data and user EEPPROM to factory default values and restarts the RN2483 module. |
| set(1) | Sets specified system parameter values. |
| get(1) | Gets specified system parameter values. |

```
< sys sleep 5000
> ok

< sys reset
> RN2483 0.9.5 Mar 24 2015 14:17:03
```

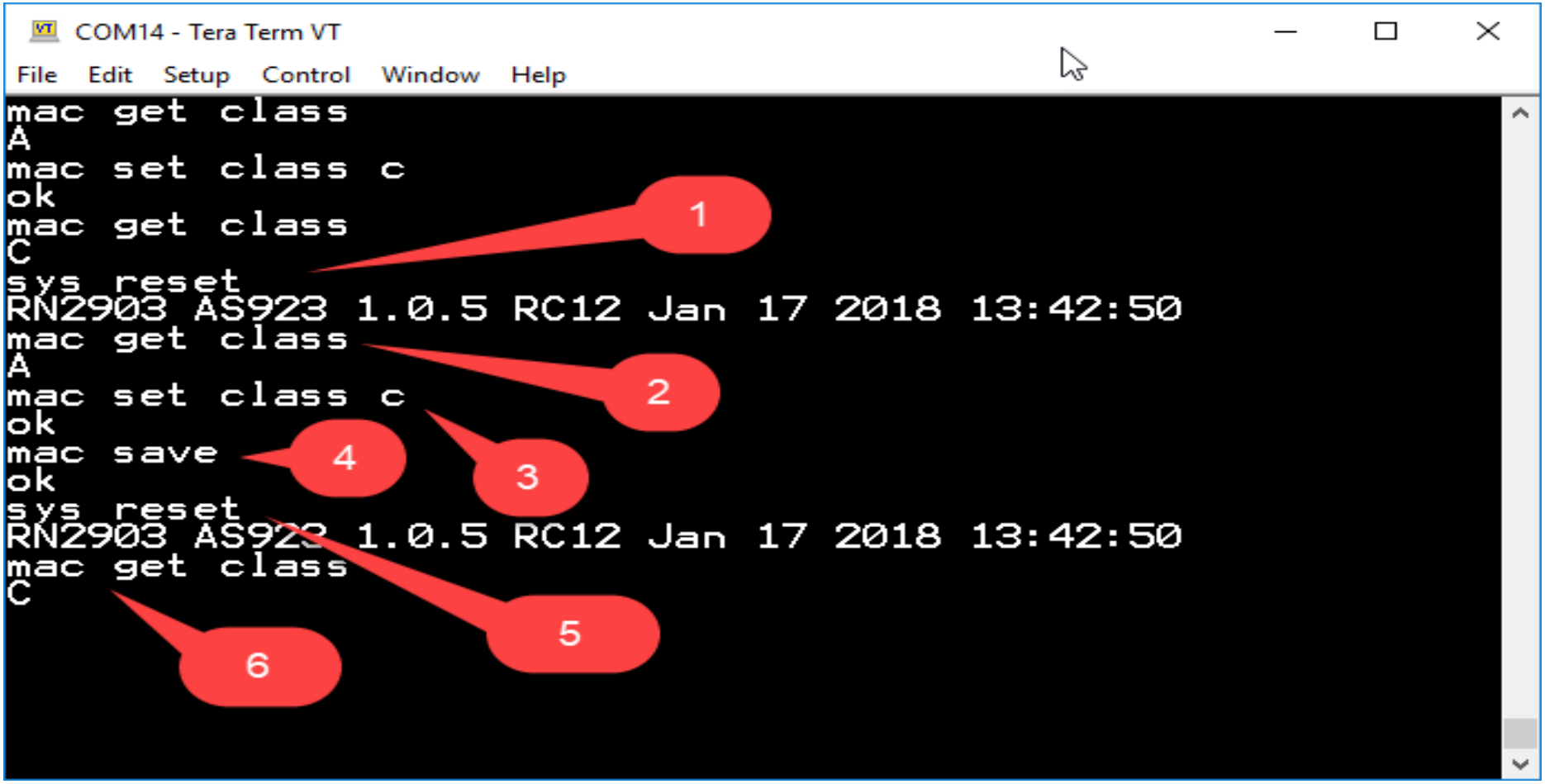



**ตัวอย่างคำสั่งเบื้องต้นของ RN2903**

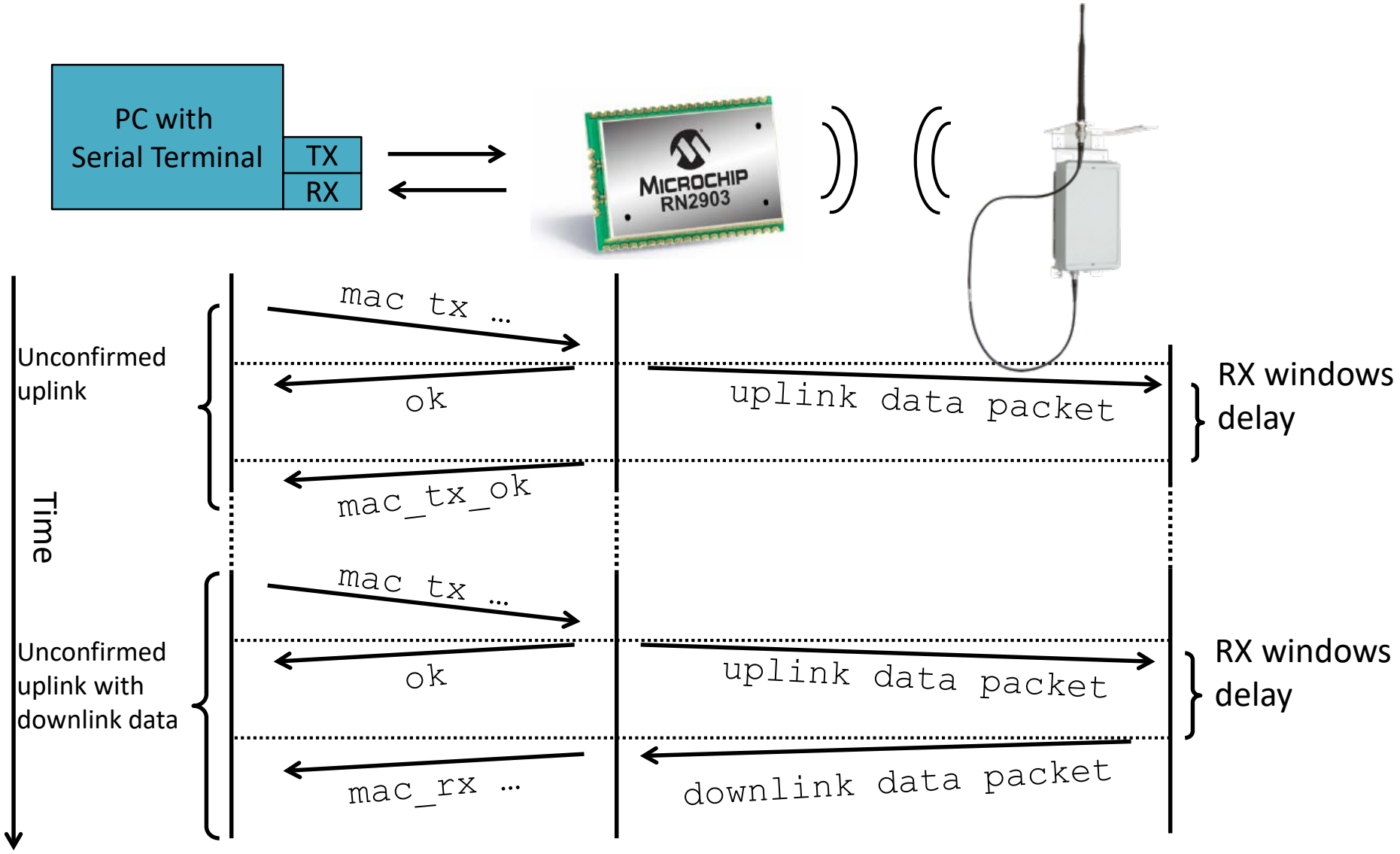
-mac get class // คำสั่งในการตรวจสอบ class

-mac set class c // คำสั่งในการตั้งค่า class c

-sys reset // คำสั่ง reset

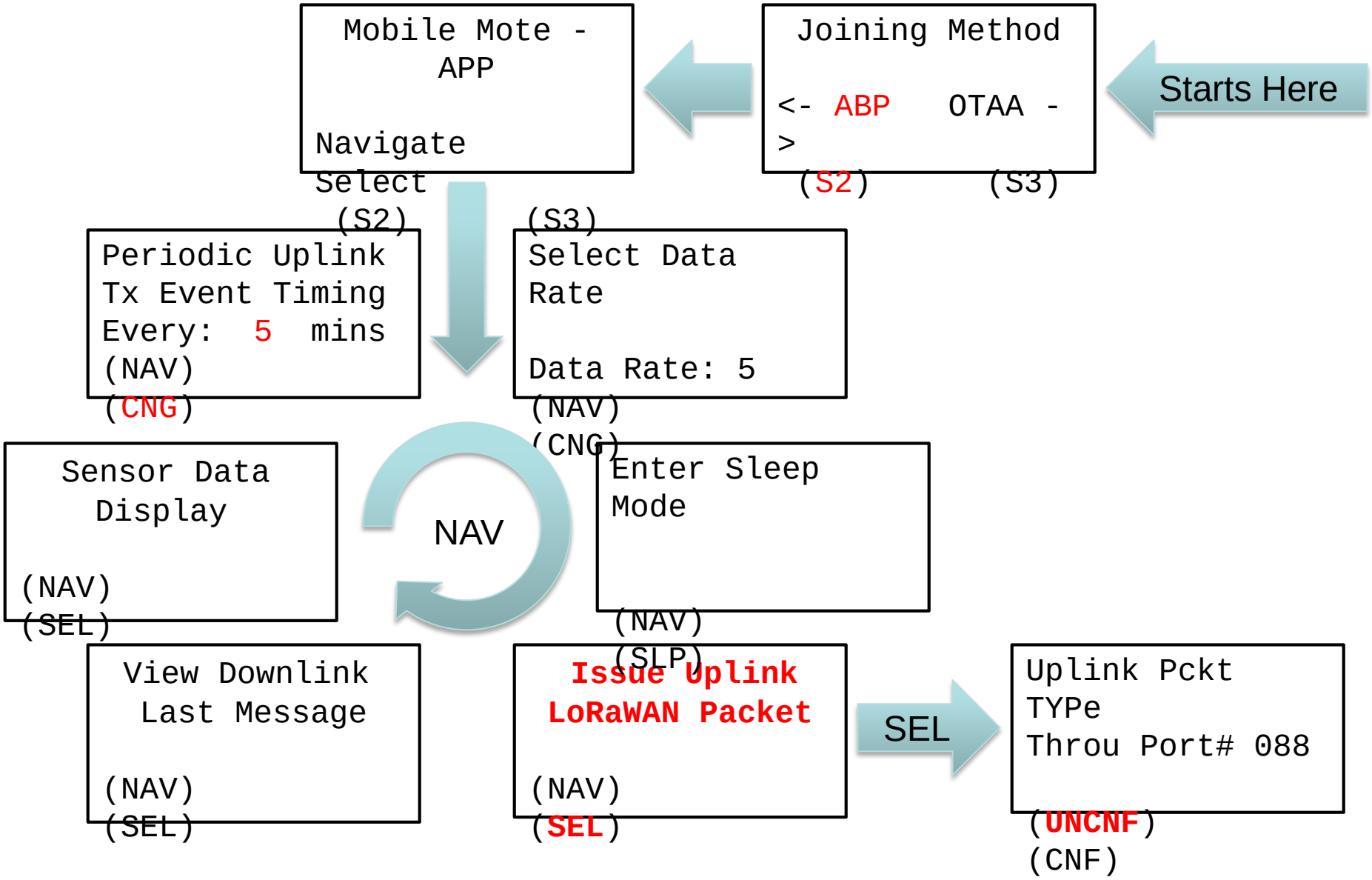



Note: ข้อควรระวัง สังเกตถ้าเราไม่ใช้คำสั่ง mac save จะไม่มีการบันทึกการตั้งค่าใน EEPROM ของ RN2903


PC with
Serial Terminal
TX
RX
MICROCHIP
RN2903
Unconfirmed
uplink
mac tx ...
ok
uplink data packet
RX windows
delay
mac_tx_ok
Time
mac tx ...
Unconfirmed
uplink with
downlink data
ok
uplink data packet
RX windows
delay
mac_rx ...
downlink data packet

# Mote Menu operating

# Mote Menus


Mobile Mote - APP
Navigate
Select
(S2)
Joining Method
<- ABP OTAA ->
(S2) (S3)
Starts Here
(S3)
Periodic Uplink
Tx Event Timing
Every: 5 mins
(NAV)
(CNG)
Select Data Rate
Data Rate: 5
(NAV)
(CNG)
Sensor Data Display
(NAV)
(SEL)
NAV
Enter Sleep Mode
(NAV)
(SLP)
View Downlink
Last Message
(NAV)
(SEL)
Issue Uplink
LoRaWAN Packet
(NAV)
(SEL)
SEL
Uplink Pckt TYPe
Throu Port# 088
(UNCNF)
(CNF)

# Mote Menus on testing.

## Mote operate on Battery.



**Joining Method**

-ABP

-OTAA

Select OTAA (S3)


Select OTAA (S3)


Select SEL (S3)


Select (CNF) ->S3

CNF= Confirm.
Uplink package port: 083


# Mote Menus on testing.

## Mote operate on USB CABLE


**NOTE: For MOTE is connected by USB cable.**

1. Push S2
2. Push S1

If push S1 at first time, will go to Testing mode.



## Select OTAA to test



Select (S3) and (SEL) S3



**SELECT CNF and Wait (DATA Transmitted)**

# PAYLOAD of MOTE

Received data

DOWNLOAD

| Id | Endpoint ID | Received time | Sequence number | Port | Radio ID | Channel | SNR | RSSI | Frequency | Modulation | Data Rate | Coding rate | Payload HEX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 42 | 4A30B00000003 | 10/26/2017 06:56:29 PM | 5 | 124 | 0 | 0 | 10 dB | -5 dBm | 923.200 MHz | LoRa | SF12BW125 | 4/5 | 3935312030323800 |
| 41 | 4A30B00000003 | 10/26/2017 06:56:20 PM | 4 | 50 | 0 | 2 | 10.5 dB | -8 dBm | 923.000 MHz | LoRa | SF12BW125 | 4/5 | 3935392030323700 |
| 40 | 4A30B00000003 | 10/26/2017 06:55:54 PM | 3 | 117 | 0 | 5 | 9.2 dB | -15 dBm | 921.800 MHz | LoRa | SF12BW125 | 4/5 | 3237203032370000 |
| 39 | 4A30B00000003 | 10/26/2017 06:55:20 PM | 2 | 151 | 0 | 1 | 12.2 dB | -33 dBm | 923.400 MHz | LoRa | SF12BW125 | 4/5 | 3931203032370000 |
| 38 | 4A30B00000003 | 10/26/2017 06:54:36 PM | 1 | 1 | 0 | 3 | 9.8 dB | -25 dBm | 922.800 MHz | LoRa | SF12BW125 | 4/5 | 3231203032380000 |
| 37 | 4A30B00000003 | 10/26/2017 06:54:05 PM | 0 | 78 | 0 | 6 | 8.5 dB | -34 dBm | 922.000 MHz | LoRa | SF12BW125 | 4/5 | 3520003032370000 |
| 36 | 4A30B00000003 | 10/26/2017 05:59:43 PM | 4 | 132 | 0 | 3 | 10.2 dB | 1 dBm | 922.800 MHz | LoRa | SF12BW125 | 4/5 | 3534203231360000 |
| 35 | 4A30B00000003 | 10/26/2017 05:59:15 PM | 3 | 125 | 0 | 7 | 8 dB | 2 dBm | 922.200 MHz | LoRa | SF12BW125 | 4/5 | 3536203032360000 |
| 34 | 4A30B00000003 | 10/26/2017 05:58:34 PM | 2 | 32 | 0 | 6 | 9.8 dB | 1 dBm | 922.000 MHz | LoRa | SF12BW125 | 4/5 | 3931382030323700 |
| 33 | 4A30B00000003 | 10/26/2017 05:58:16 PM | 1 | 43 | 0 | 5 | 10 dB | 1 dBm | 921.800 MHz | LoRa | SF12BW125 | 4/5 | 3238332030323500 |

| Payload | HEX |
|---|---|
| 3935312030323800 | |
| 3935392030323700 | |
| 3237203032370000 | |
| 3931203032370000 | |
| 3231203032380000 | |
| 3520003032370000 | |
| 3534203231360000 | |
| 3536203032360000 | |
| 3931382030323700 | |
| 3238332030323500 | |

39 35 31 20 30 32 38 00 -> 951 0280

951 = light
0280 = temp

**Note:** *ASCII Table*

MICROCHIP

# ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | [NULL] | 32 | 20 | [SPACE] | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | [START OF HEADING] | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | [START OF TEXT] | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | [END OF TEXT] | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | [END OF TRANSMISSION] | 36 | 24 | $ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | [ENQUIRY] | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | [ACKNOWLEDGE] | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | [BELL] | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | [BACKSPACE] | 40 | 28 | ( | 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | [HORIZONTAL TAB] | 41 | 29 | ) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | [LINE FEED] | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | [VERTICAL TAB] | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | [FORM FEED] | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | [CARRIAGE RETURN] | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | [SHIFT OUT] | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | [SHIFT IN] | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | [DATA LINK ESCAPE] | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | [DEVICE CONTROL 1] | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | [DEVICE CONTROL 2] | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | [DEVICE CONTROL 3] | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | [DEVICE CONTROL 4] | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | [NEGATIVE ACKNOWLEDGE] | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | [SYNCHRONOUS IDLE] | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | [ENG OF TRANS. BLOCK] | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | [CANCEL] | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | [END OF MEDIUM] | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | [SUBSTITUTE] | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | [ESCAPE] | 59 | 3B | ; | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | [FILE SEPARATOR] | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | \| |
| 29 | 1D | [GROUP SEPARATOR] | 61 | 3D | = | 93 | 5D | ] | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | [DEL] |

| Payload HEX |
| --- |
| 3935312030323800 |
| 3935392030323700 |
| 3237203032370000 |
| 3931203032370000 |
| 3231203032380000 |
| 3520003032370000 |
| 3534203231360000 |
| 3536203032360000 |
| 3931382030323700 |
| 3238332030323500 |

High light testing

High heat testing

32 31 20 30 32 38 00 00
-> 21 02800

Normal environment

32 38 33 20 30 32 35 00
-> 283 0250

# BREAK TIME

# RN2903 Command via UART

# Software:
# Teraterm or any Serial terminal program.

Baudrate: 57600 8N1

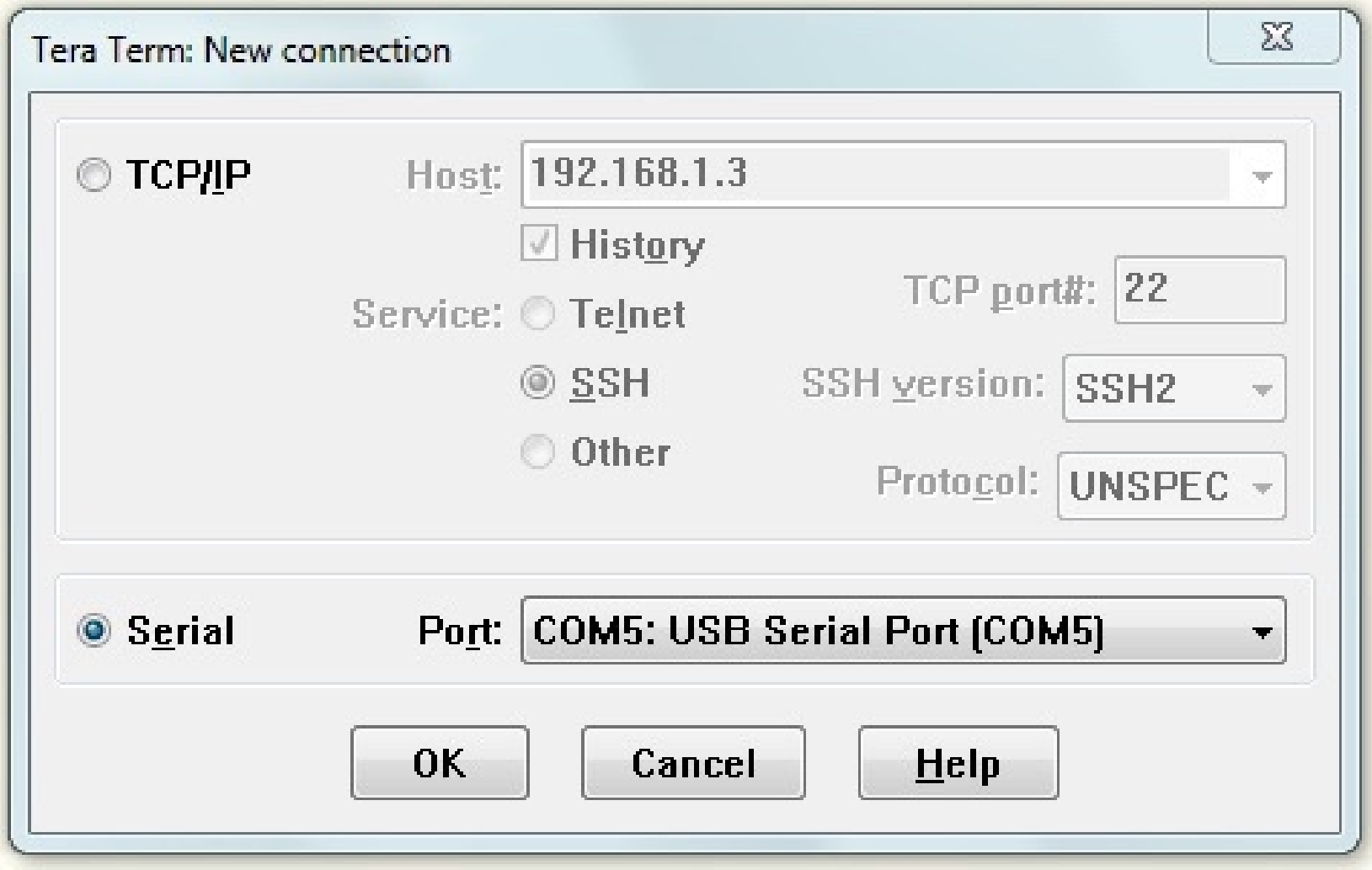

## Reference manual:

RN2903 LoRa™ Technology Module Command Reference User's Guide (DS40001811A)

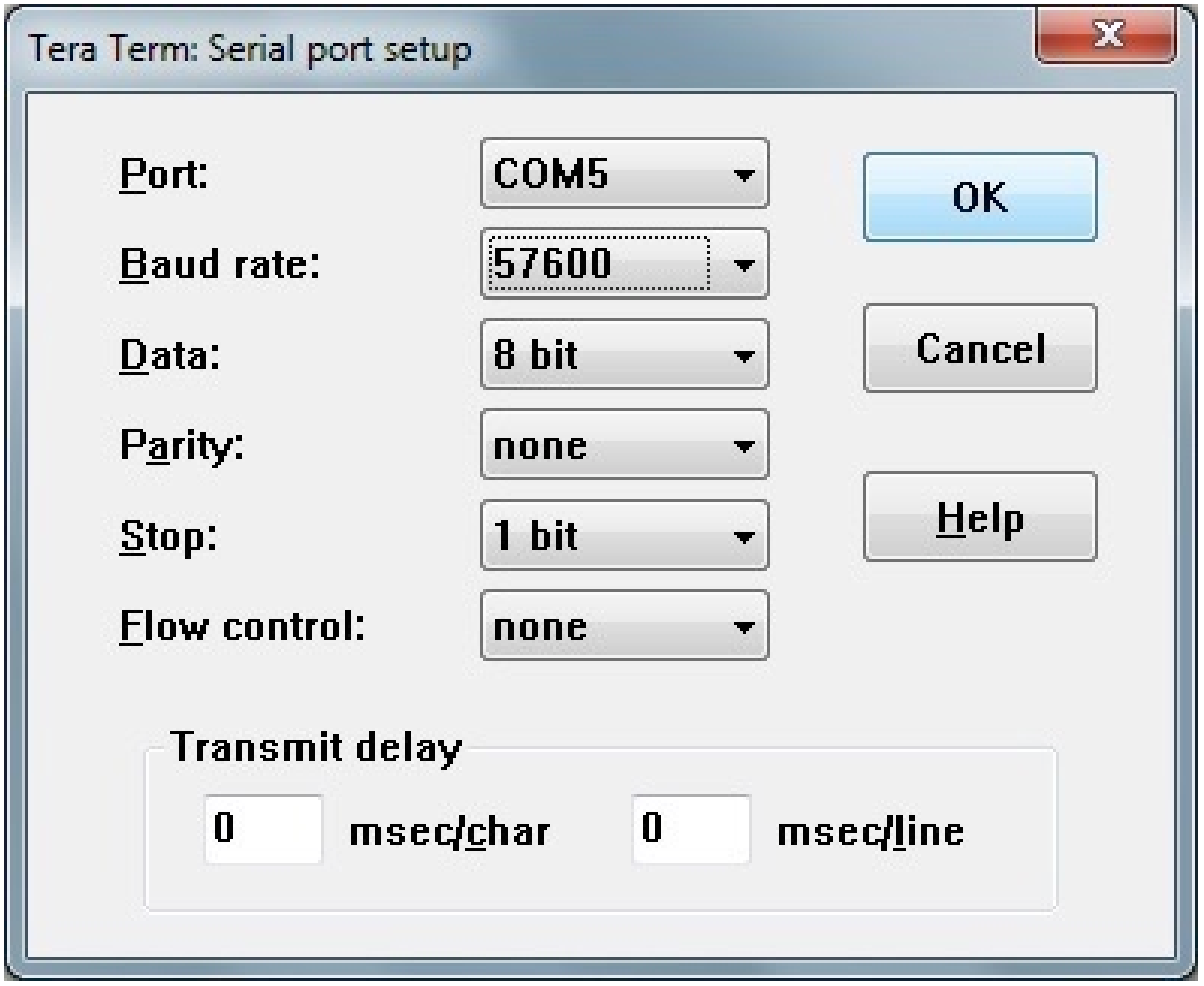
RN2483 LoRa® Technology Module Command Reference User's Guide (DS40001784F)

LoRa Mote User's Guide (DS40001808B)

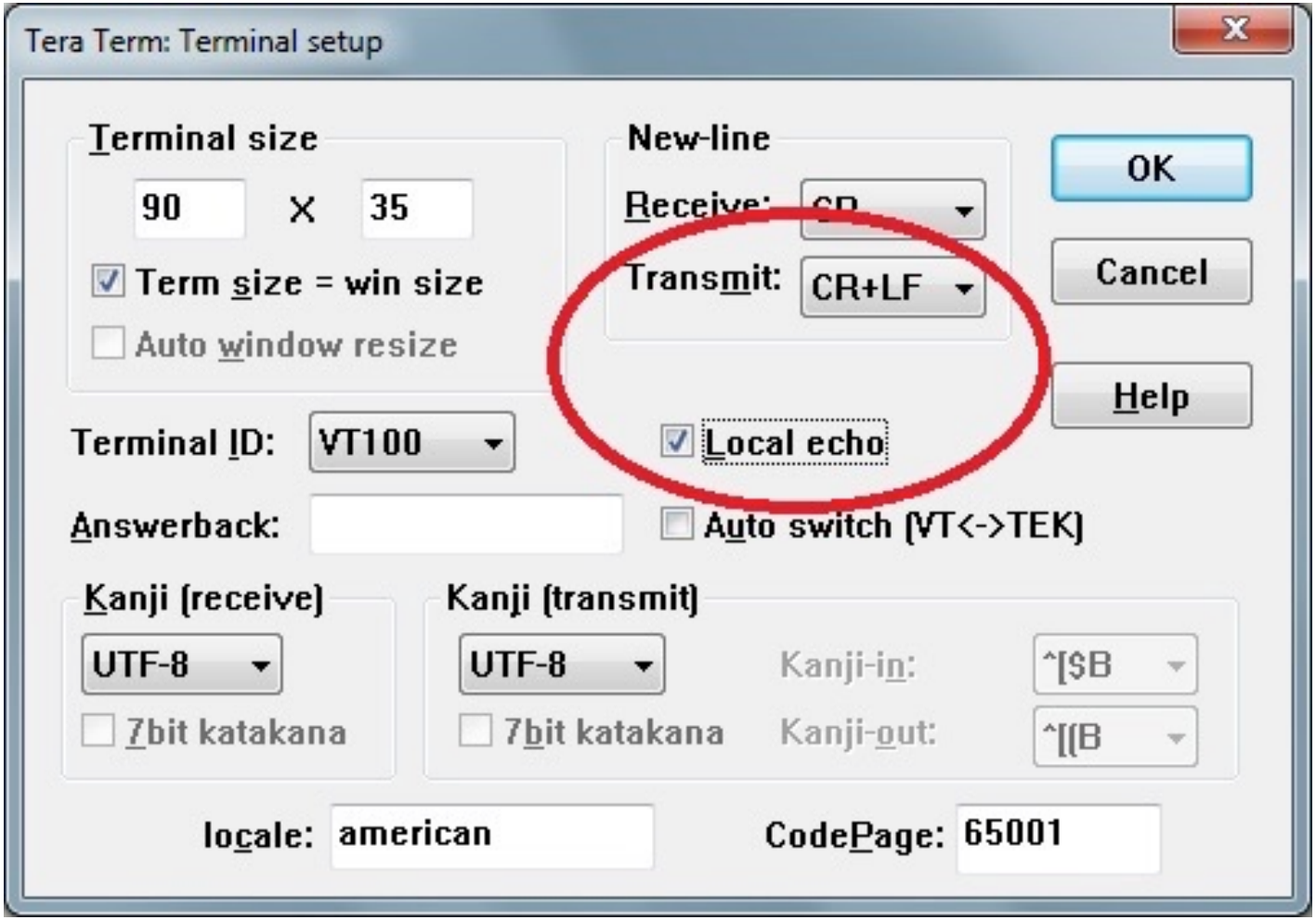
# COM Port Selection



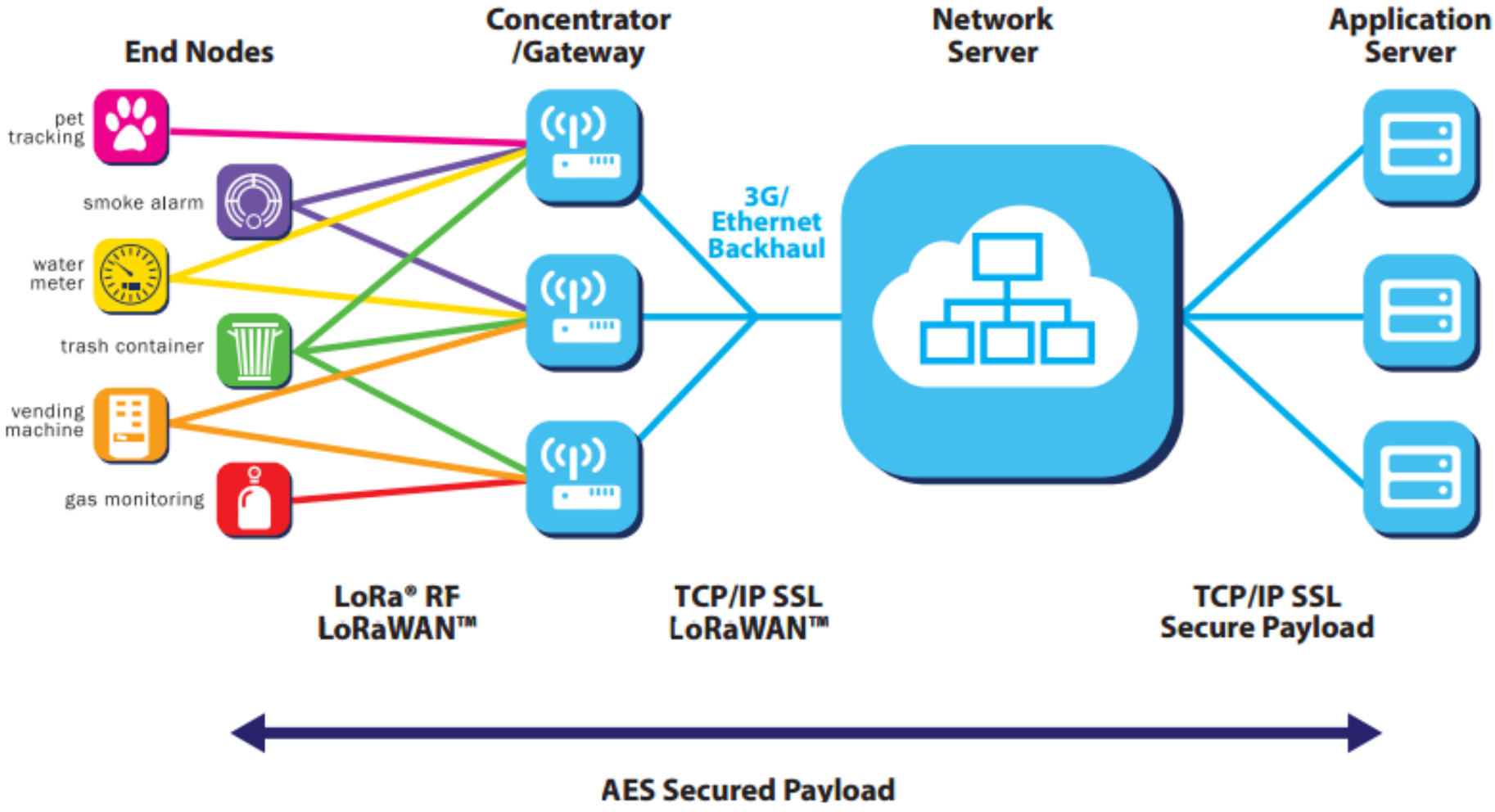
# Serial Settings



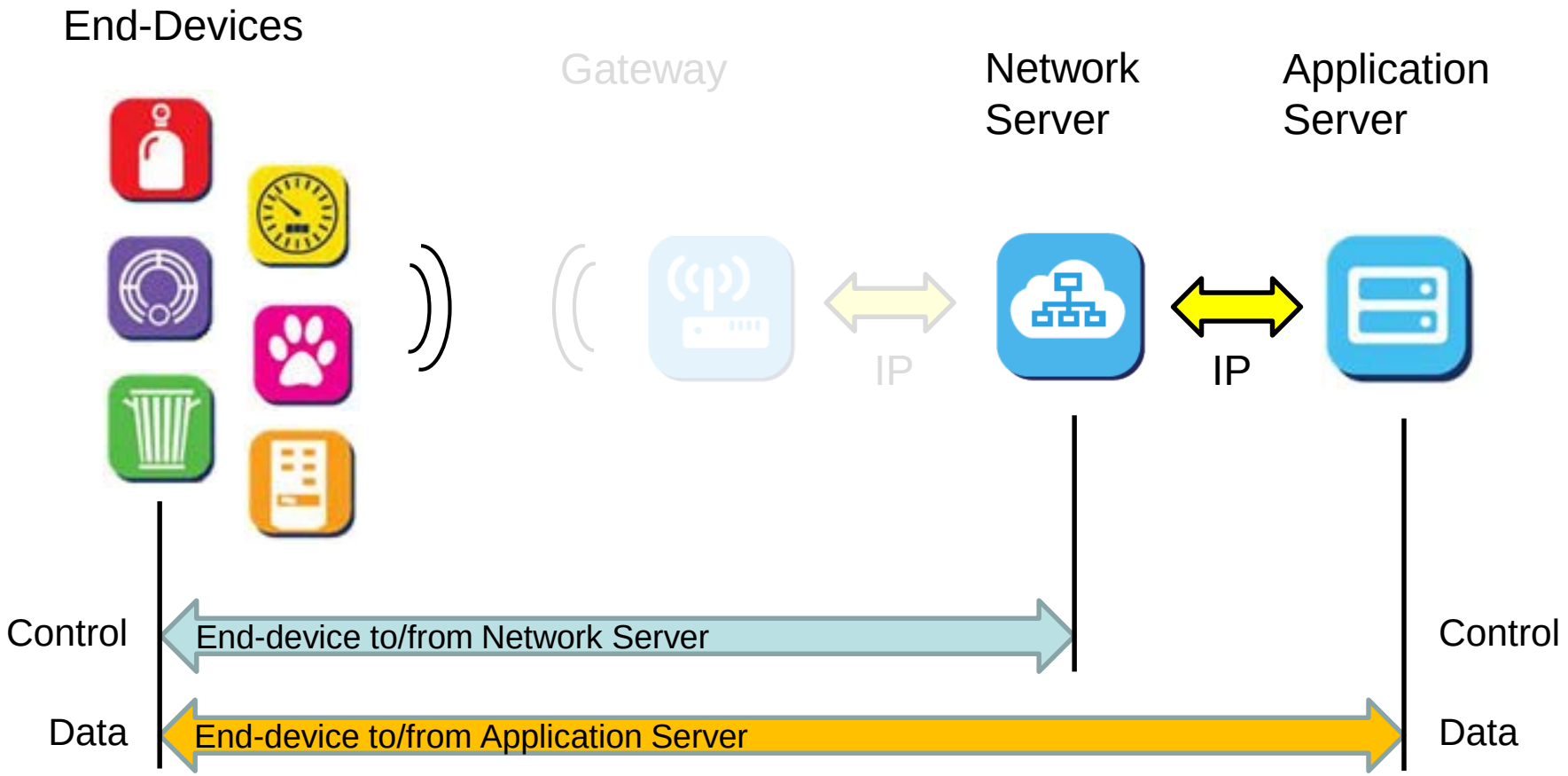
# Terminal Settings



# HOW to connect between Mote and Gateway


End Nodes
Concentrator /Gateway
Network Server
Application Server
pet tracking
smoke alarm
water meter
trash container
vending machine
gas monitoring
3G/ Ethernet Backhaul
LoRa® RF LoRaWAN™
TCP/IP SSL LoRaWAN™
TCP/IP SSL Secure Payload
AES Secured Payload

# Logical Data Flow (Programmer's Model)



- **To exchange this information, two activation methods are available:**

| Over-the-Air Activation (OTAA) | Activation By Personalization (ABP) |
| --- | --- |
| • Based on Globally Unique Identifier<br>• Over the air message handshaking | • Shared keys stored at production time<br>• Locked to a specific network |
|  |  |

# End-Device Activation (Joining)

## Over-the-Air-Activation (OTAA)

- End-device transmits Join Request to application server containing:
  - Globally unique end-device identifier (DevEUI)
  - Application identifier (AppEUI)
  - Authentication with Application key (AppKey)
- End-device receives Join Accept from application server

# End-Device Activation (Joining)

## Over-the-Air-Activation (OTAA)

- End-device authenticates Join Accept
- End-device decrypts Join Accept
- End-device extracts and stores Device Address (DevAddr)
- End-device derives:
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)

} **Security Keys**

# End-Device Activation (Joining)

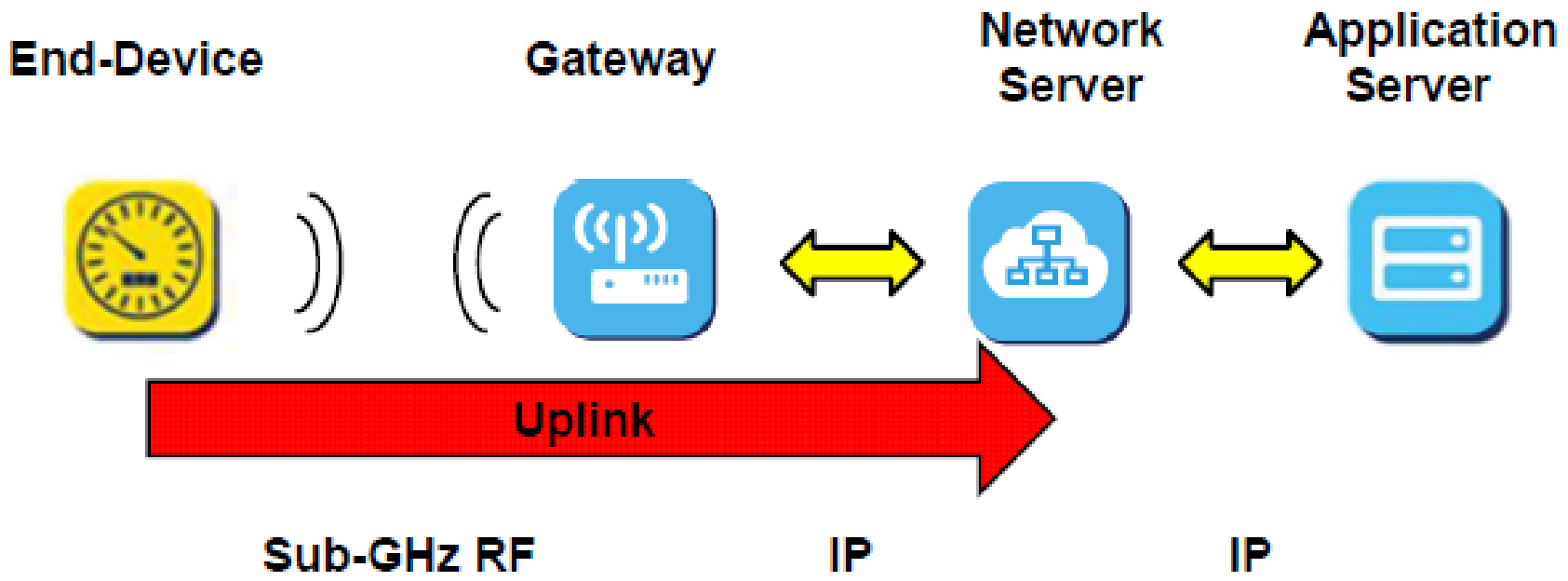

## Activation By Personalization (ABP)

- The following information is configured at production time:
  - Device Address (DevAddr)
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)
- No over the air handshaking
- Device is ready to communicate on the network without any additional procedure.
- Note that the end result is the same, the DevAddr and security keys are now known to the end-device

- **Over The Air Association (OTAA)**
  - Using AppEUI, DevEUI and AppKey

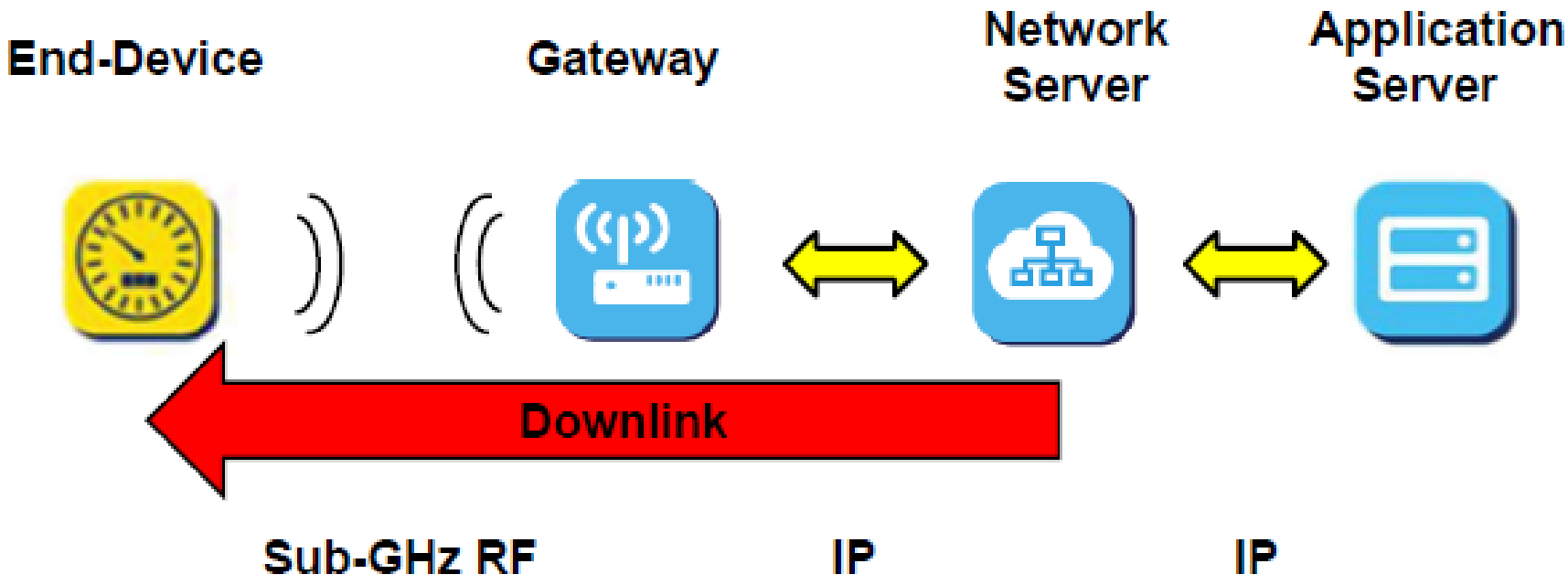

- **Association By Personalization (ABP)**
  - Requires DevAdr, NetSKey, AppSKey

- **Uplink Message**
  - End-Device to Network Server relayed by one or many Gateways



# Downlink Message

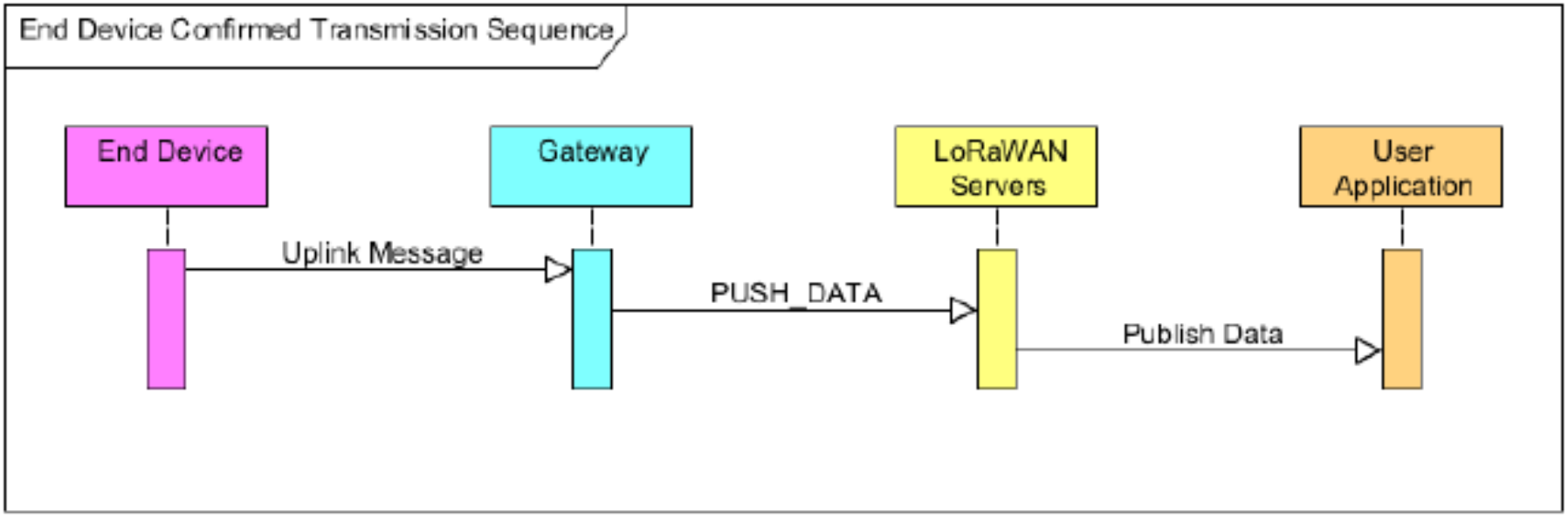
- Sent by the Network Server to only one End-Device and is relayed by a single Gateway



# Unconfirmed-Data Message

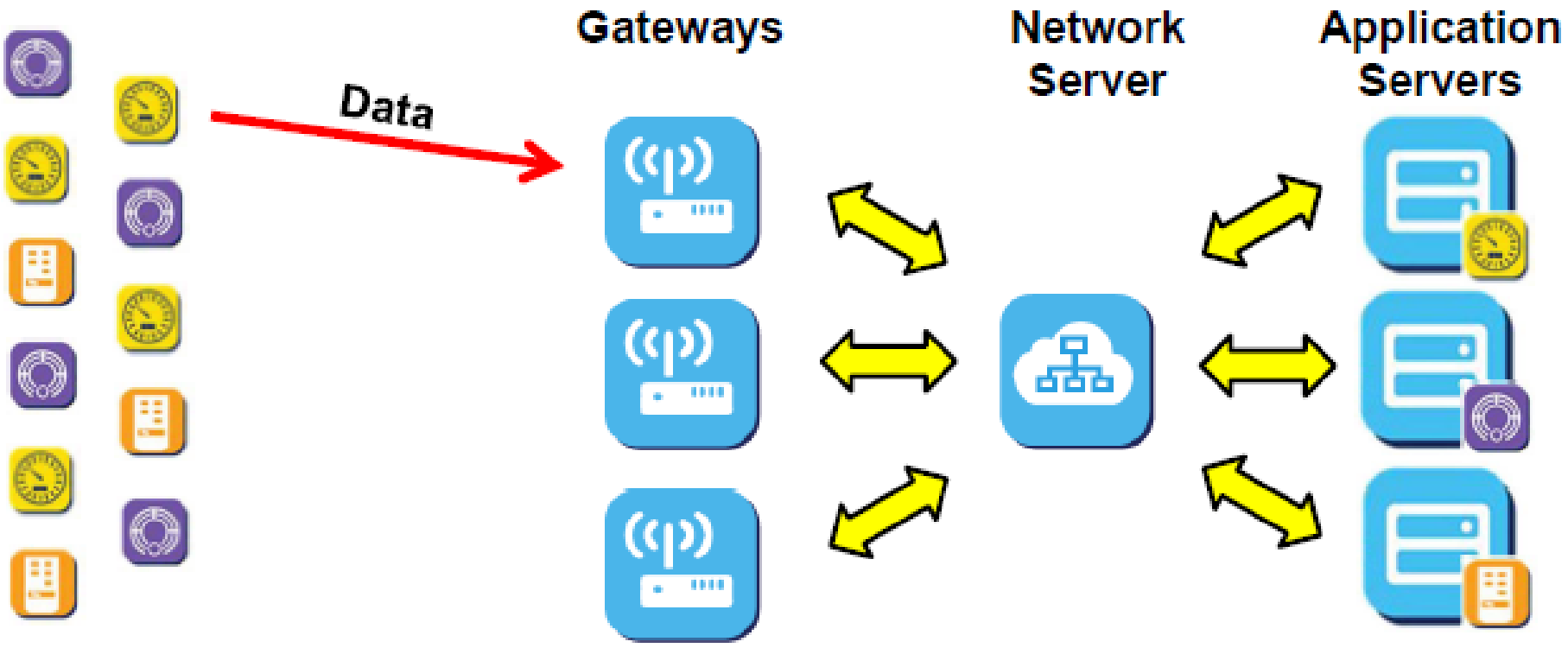
## End-Device Data Message does not require an acknowledgement

# Unconfirmed transmission



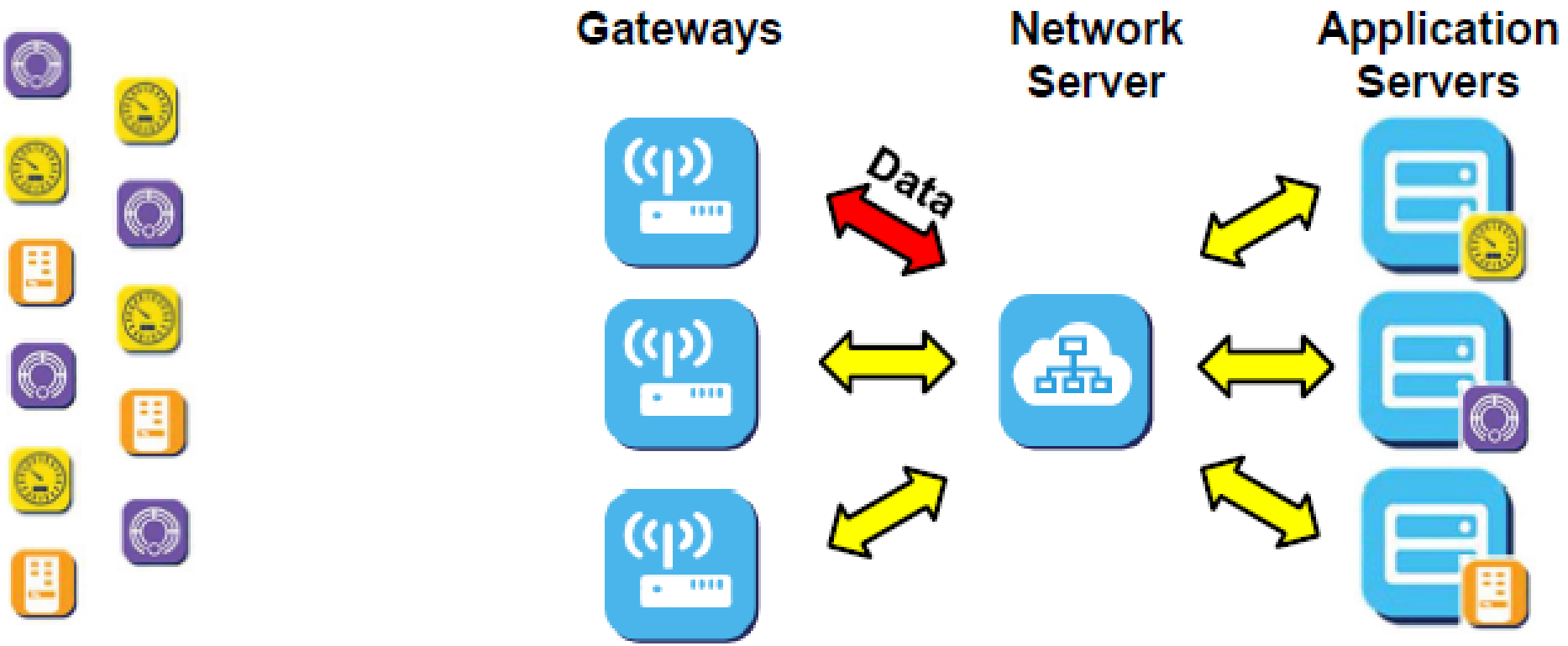

- Unidirectional data flow
- Uplink payload encrypted using AppSKey
- Uplink message integrity code (MIC) uses NwkSKey
- Message sequence number incremented (tracked by network servers)

# Unconfirmed-Data Message



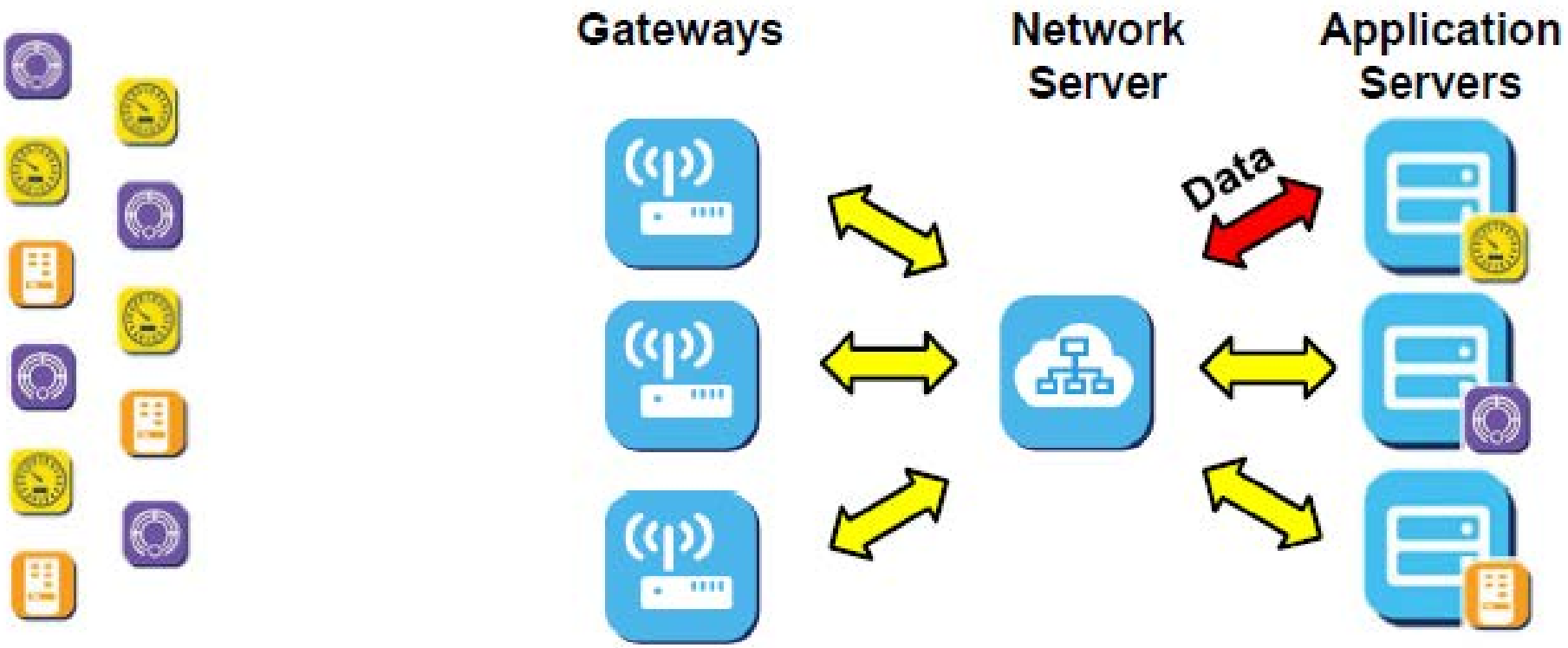

1. Electric meter transmits data

# Unconfirmed-Data Message



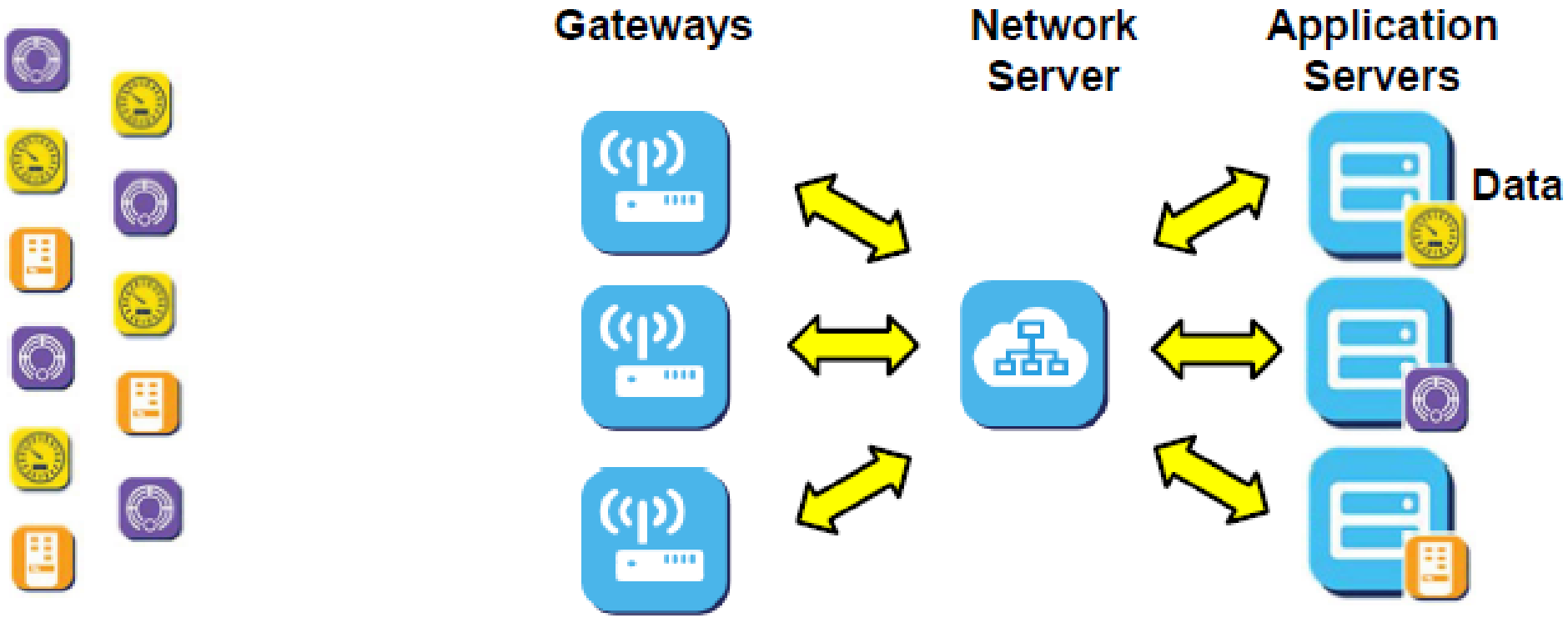

2. Gateway receives data and passes to Network Server

# Unconfirmed-Data Message




3. The Network Server authenticates data and passes it to Electric Meter Application Server

# Unconfirmed-Data Message

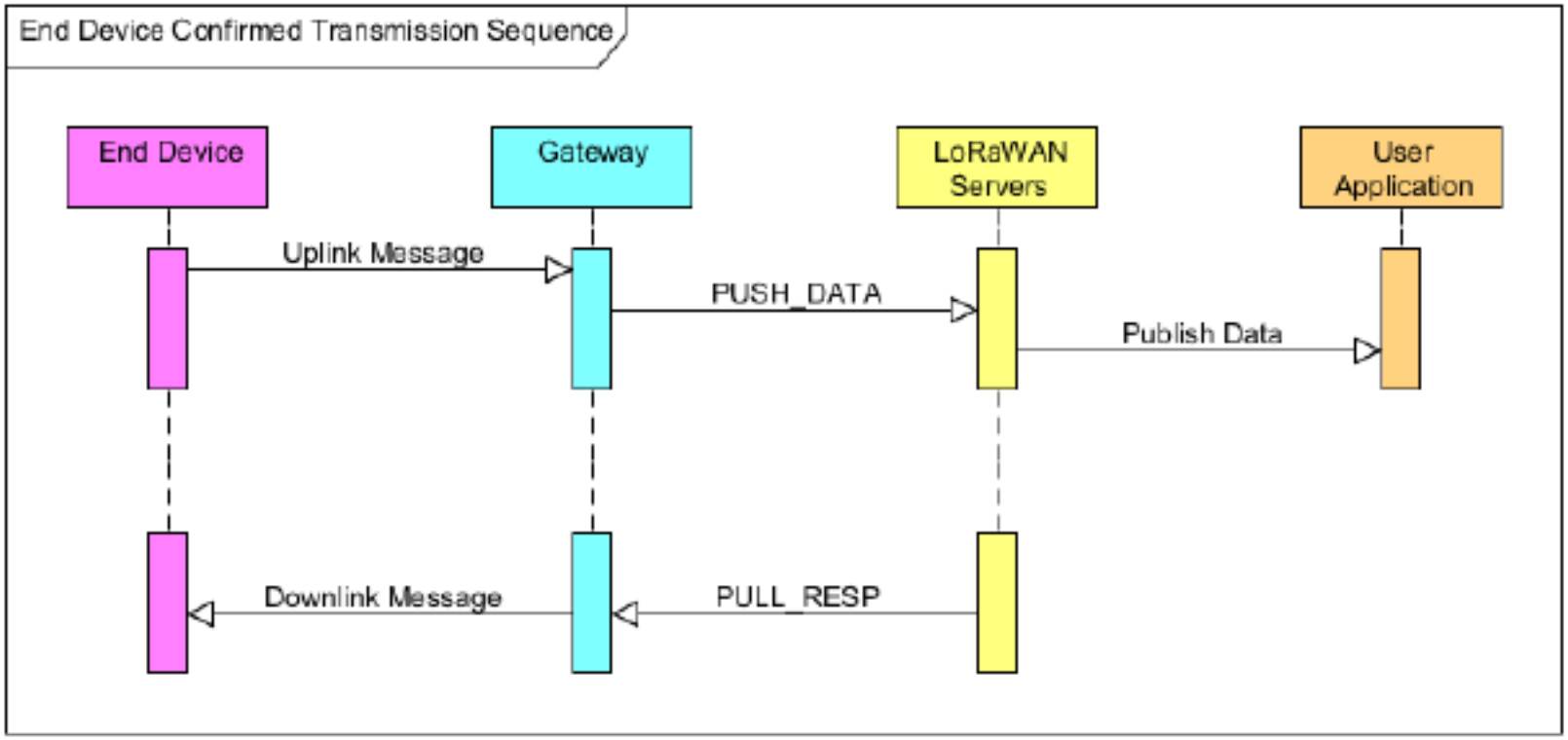



4. Electric Meter Application Server decrypts data

# Confirmed-Data Message

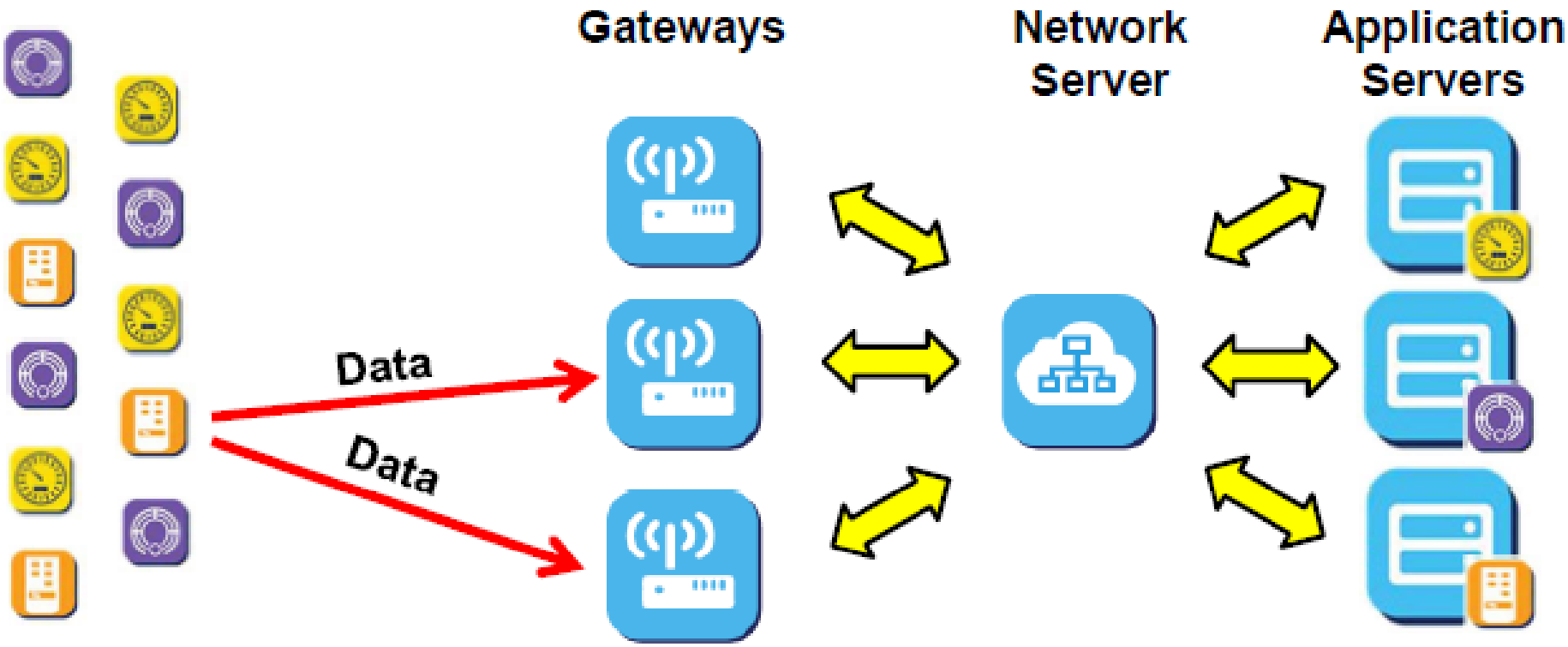
End-Device Data Message has to be acknowledged by the receiver

# Confirmed transmission



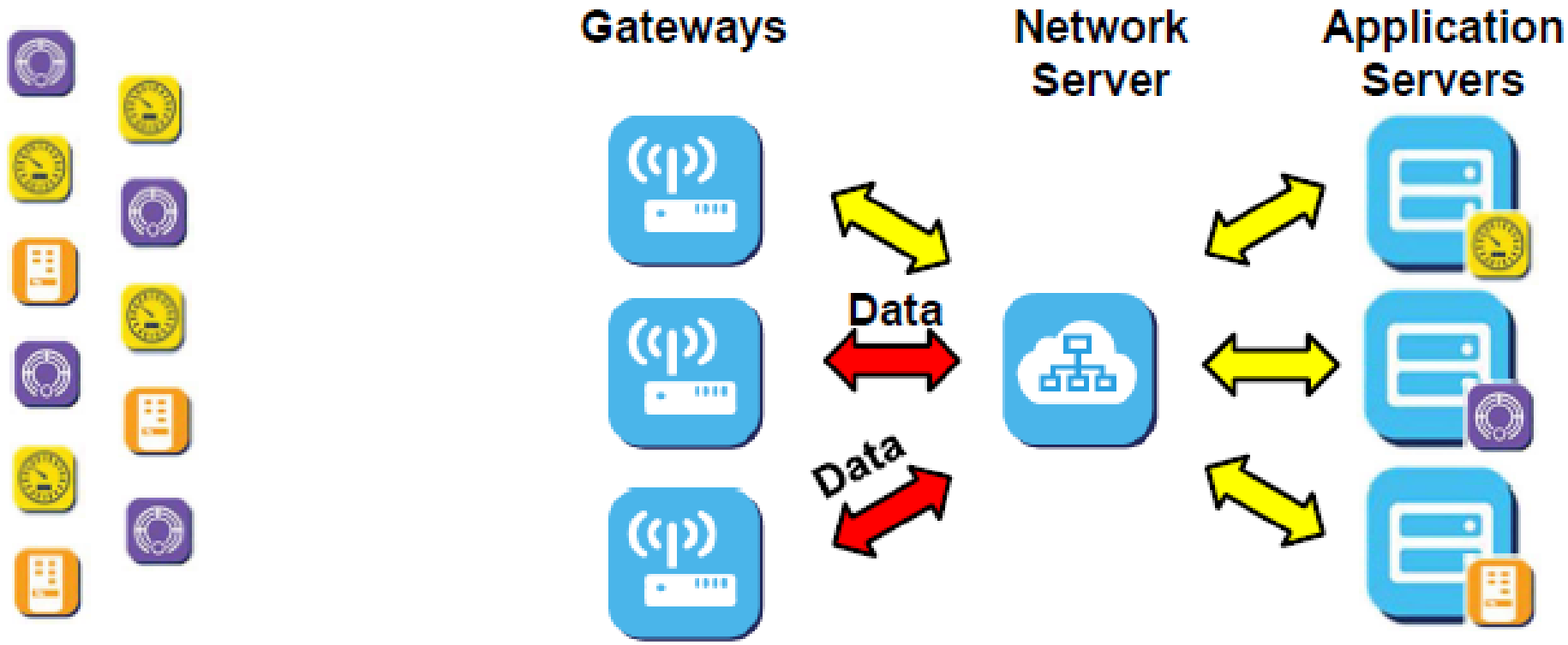

- Bidirectional data flow.
- Uplink payload encrypted using AppSKey
- Uplink message integrity code (MIC) uses NwkSKey
- Uplink sequence number returned in downlink message
- Bad / missing Ack results in uplink message retry

# Confirmed-Data Message



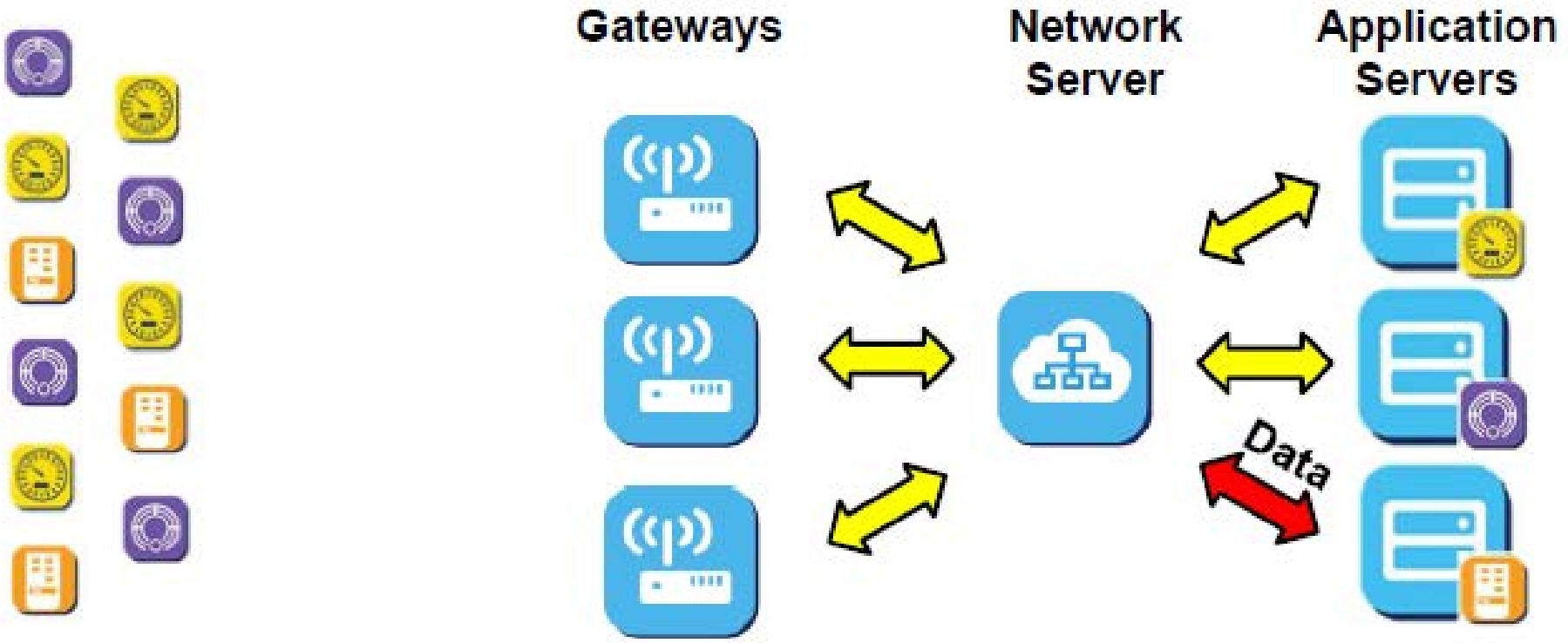

1. Vending Machine transmits data.
It is received by two Gateways.

# Confirmed-Data Message



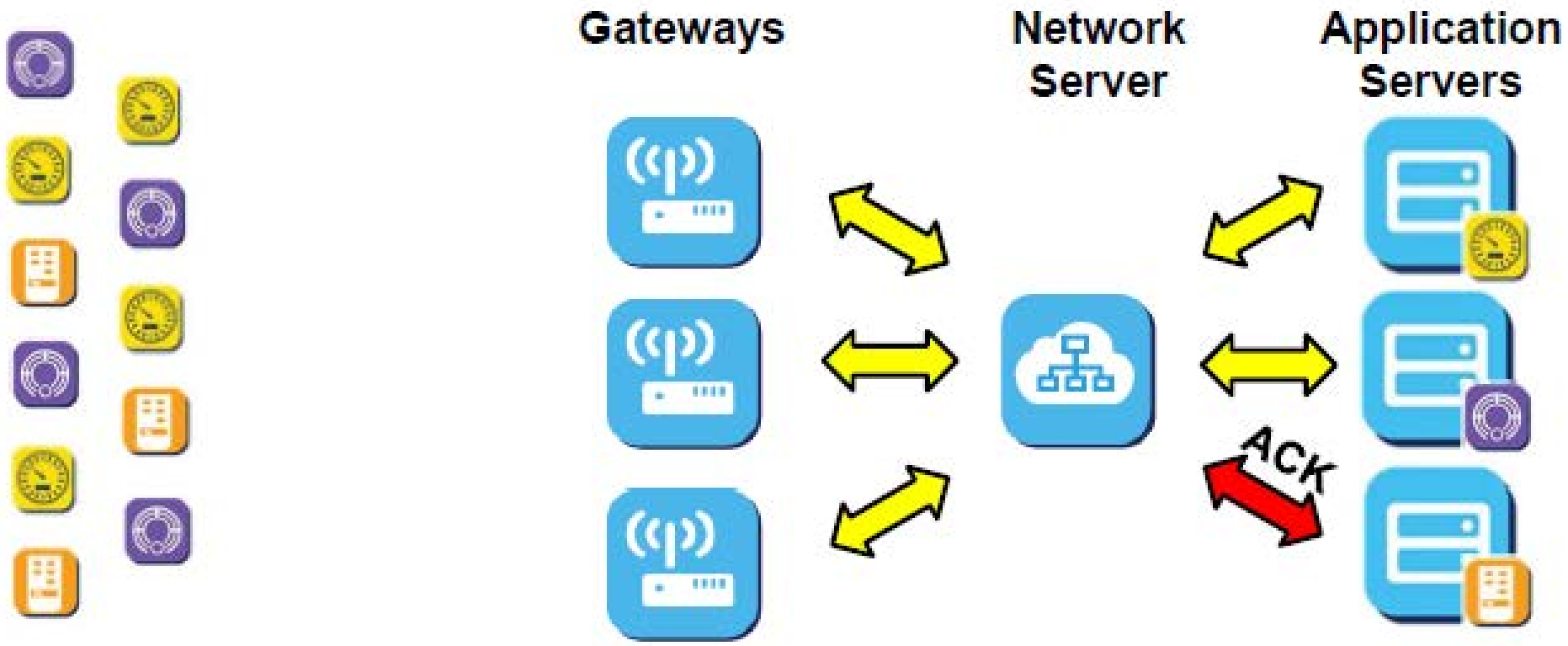

2. Both gateways "pass through" the data to the Network Server.

# Confirmed-Data Message



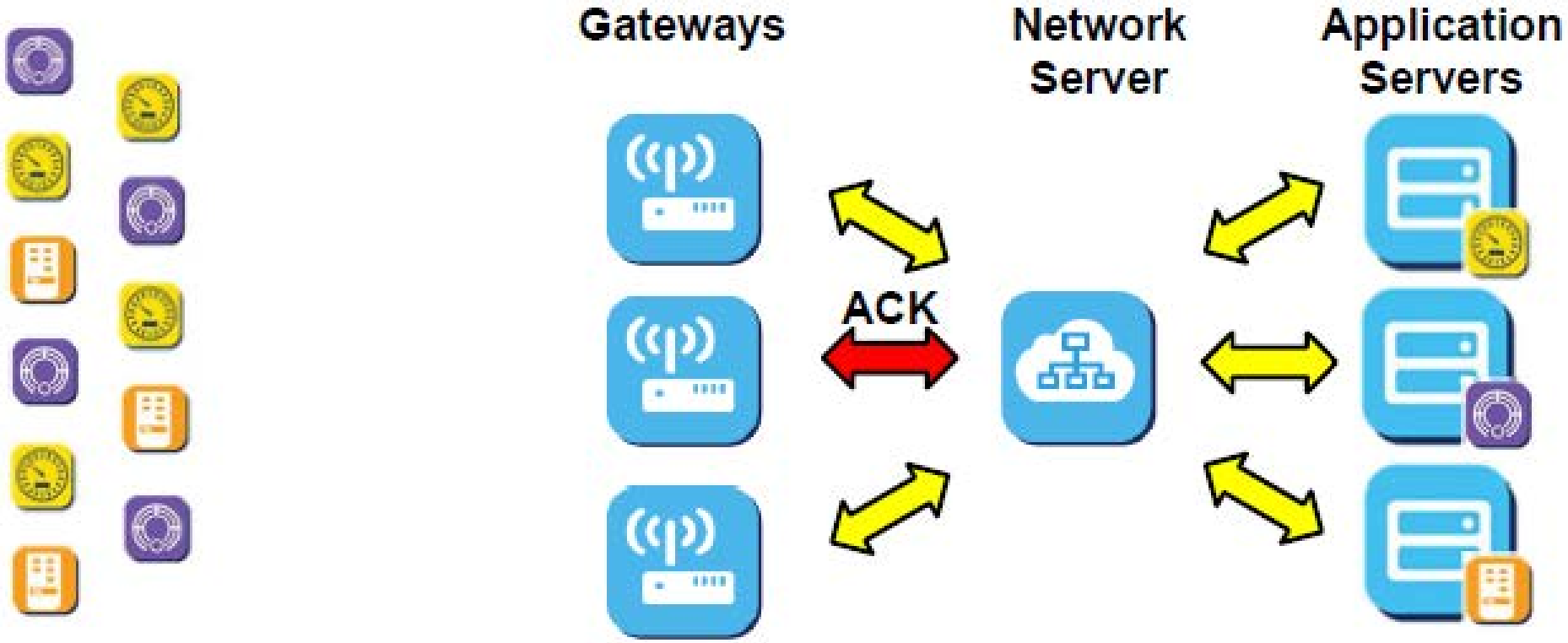

3. The Network Server forwards the data to the Vending Machine Applications Server

# Confirmed-Data Message



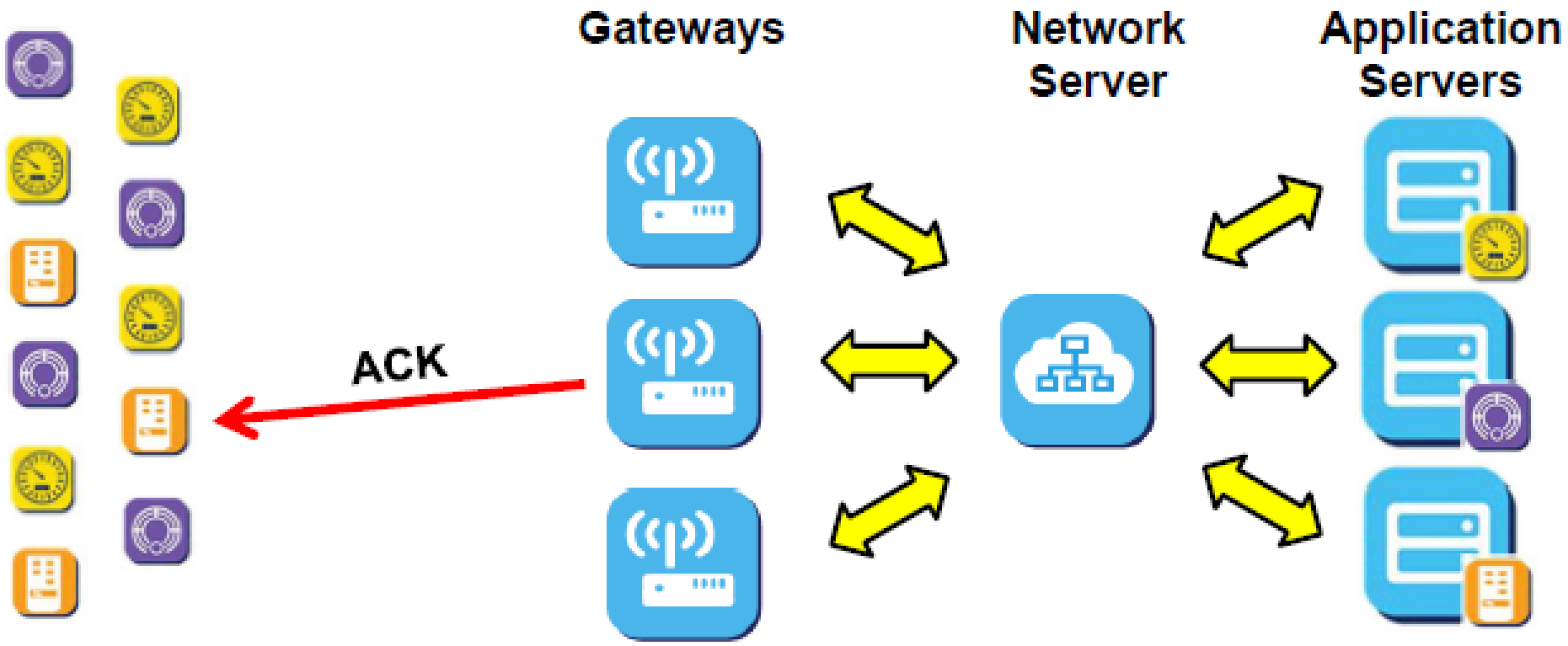

4. The Vending Machine Applications Server sends an acknowledgement

# Confirmed-Data Message




5. The Network Server selects the best path (gateway) to transmit the acknowledgement to the end-device.

# Confirmed-Data Message




6. The Gateway transmits the acknowledgement to the end-device

# Frequency AS923 of each CHANNEL

| Channel | Ch Freq | Command |
|---|---|---|
| 0 | 923.2MHz | 923200000 |
| 1 | 923.4MHz | 923400000 |
| 2 | 923.0MHz | 923000000 |
| 3 | 922.8MHz | 922800000 |
| 4 | 921.6MHz | 921600000 |
| 5 | 921.8MHz | 921800000 |
| 6 | 922.0MHz | 922000000 |
| 7 | 922.2MHz | 922200000 |

# Set Channel of mote

**Mote board**

**Ex set channel 2**

**> mac set ch freq 2 923000000 // Sets ch2 to 923.000MHz (Same channels set in the gateway)**

**> mac set ch dcycle 2 999 // Sets the duty cycle to 0.1% (= 99.9% off)**

**> mac set ch drrange 2 0 5 // Allows data rates 0 to 5 on this channel**

**> mac set ch status 2 on // Enables the channel**

# OTAA Joining

========== OTAA ==========

1. sys reset // คำสั่ง reset
2. sys get hweui // อ่านค่า hweui
3. mac set deveui 0004A30B00000004
4. mac set appeui 10203040A0B0C0D0
5. mac set appkey 10002000300040005000600070008000
6. mac save

7. mac join otaa //คำสั่งการติดต่อแบบ otaa
8. mac tx uncnf 4 003E8 //คำสั่งส่งแบบ uncnf

# APB Joining

========== APB ==========

sys reset

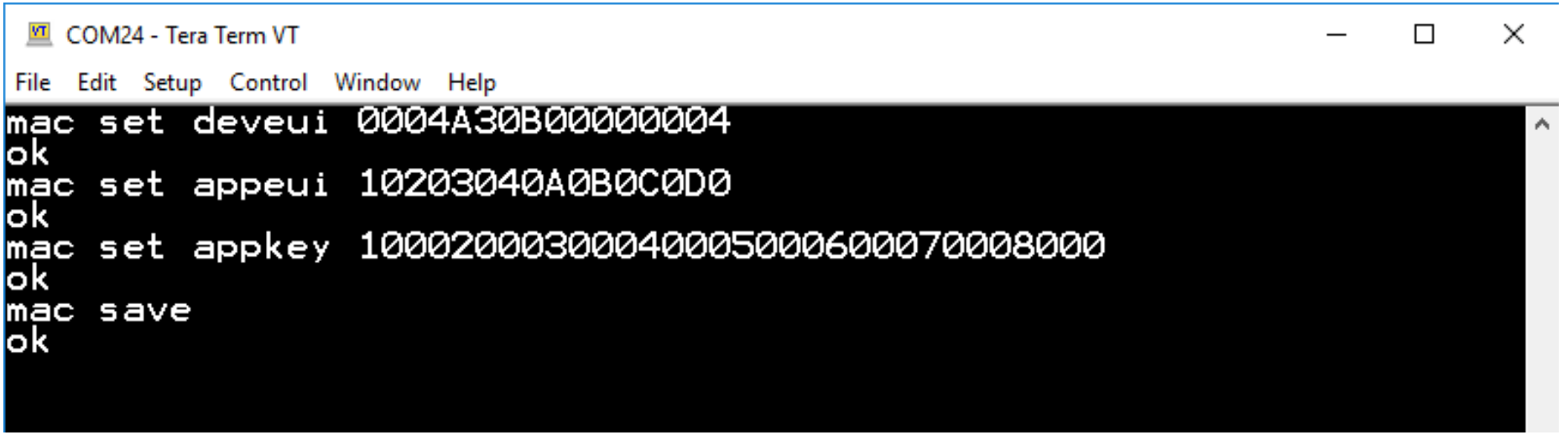
1. sys get hweui
2. mac set devaddr ABCD0004
3. mac set nwkskey 10002000300040005000600070008000
4. mac set appskey 10002000300040005000600070008000
5. **mac save**
6. mac join abp

7. mac tx uncnf 4 003e8

# JOIN OTAA

# Command mote board via USB cable.

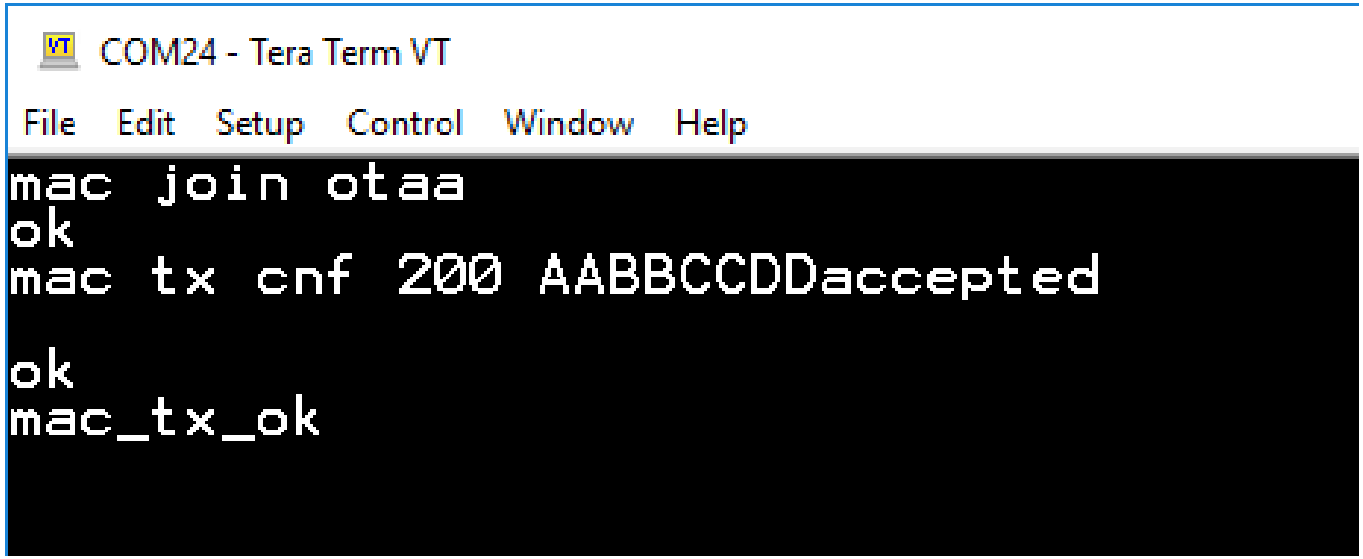
## OTAA




## Resister Mote to Gateway

- deveui
- appeui
- appkey

# Command mote board via USB cable.

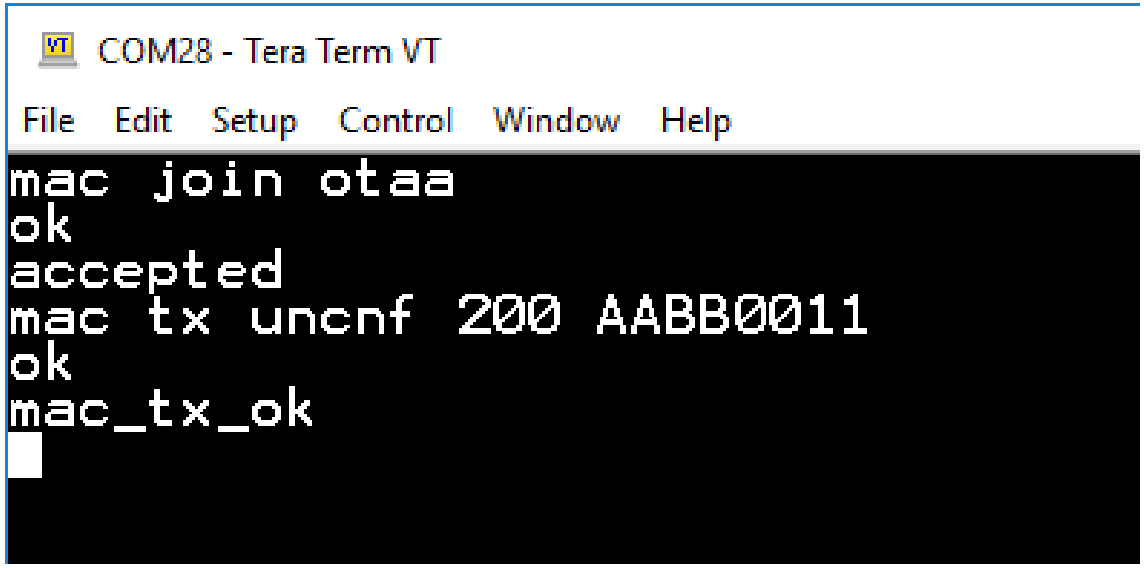



## OTAA Joining

- Mac join otaa
- Mac tx cnf port xxxxxx //Confirm

Note: xxxxxx is message.

# Command mote board via USB cable.




## OTAA Joining

- Mac join otaa
- Mac tx uncnf port xxxxxx //Unconfirm

Note: xxxxxx is message.

# Command mote board via USB cable.

Received data

DOWNLOAD

| Id | Endpoint ID | Received time | Sequence number | Port | Radio ID | Channel | SNR | RSSI | Frequency | Modulation | Data Rate | Coding rate | Payload HEX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 39 | 4A30B00000004 | 10/26/2017 07:17:43 PM | 0 | 200 | 0 | 6 | 10.5 dB | -24 dBm | 922.000 MHz | LoRa | SF12BW125 | 4/5 | AABB0011 |
| 38 | 4A30B00000004 | 10/26/2017 07:16:55 PM | 1 | 200 | 0 | 7 | 9.2 dB | -25 dBm | 922.200 MHz | LoRa | SF12BW125 | 4/5 | AABB0011 |
| 37 | 4A30B00000004 | 10/26/2017 07:16:49 PM | 0 | 200 | 0 | 2 | 10 dB | -26 dBm | 923.000 MHz | LoRa | SF12BW125 | 4/5 | AABBCCDD |
| 36 | 4A30B00000004 | 10/26/2017 07:13:57 PM | 0 | 200 | 0 | 3 | 9.5 dB | -28 dBm | 922.800 MHz | LoRa | SF12BW125 | 4/5 | AABBCCDD |

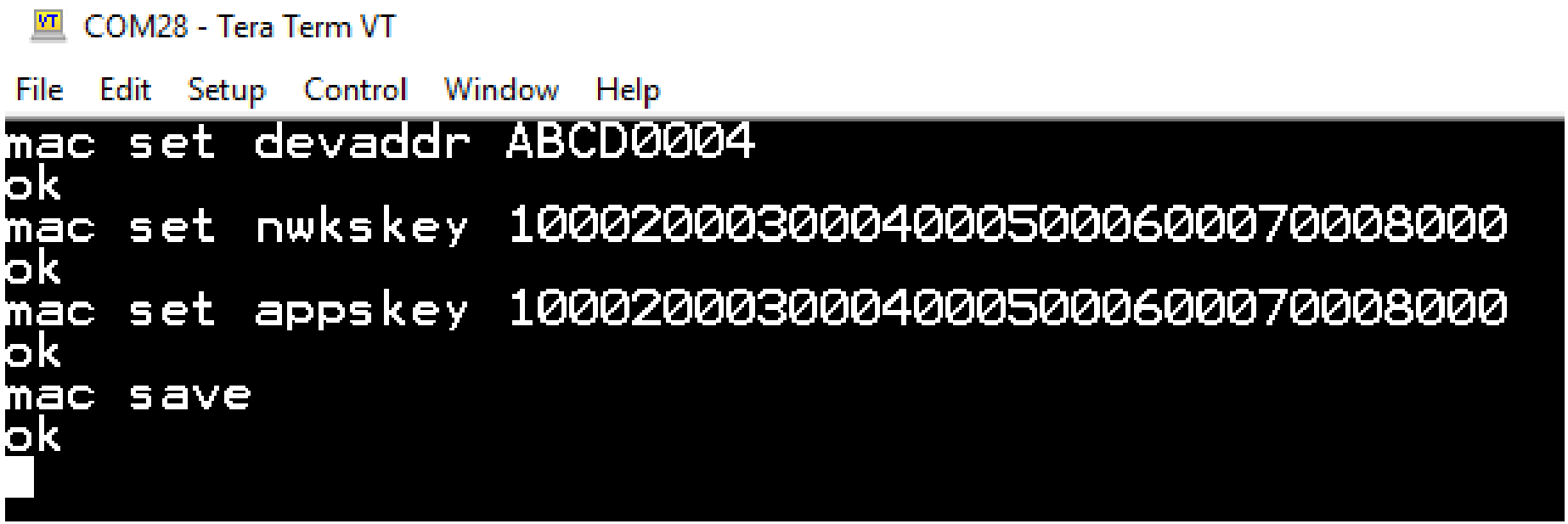
**Note:** Result of Tx confirm and unconfirm.

**Payload :AABBCCDD is confirmed.**
**Payload :AABB0011 in unconfirmed.**

# JOIN ABP

# Command mote board via USB cable.

## ABP




**Note: ABP**

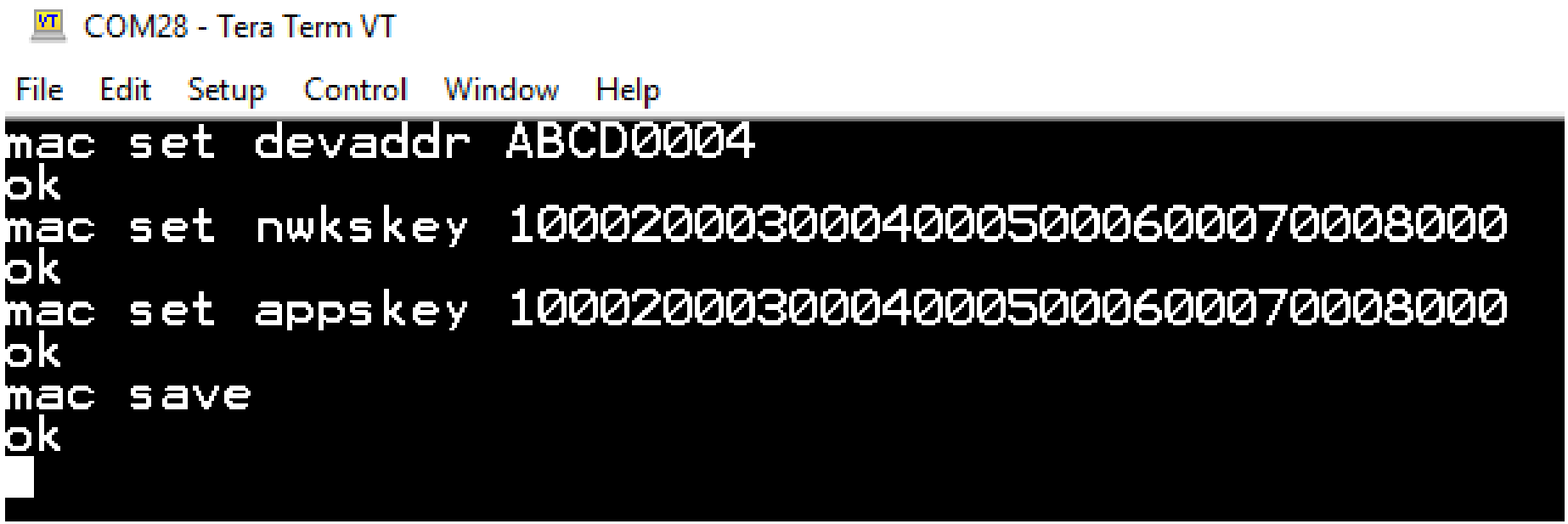
Set devaddr

Set nwkskey

Set appskey

# Command mote board via USB cable.

## ABP



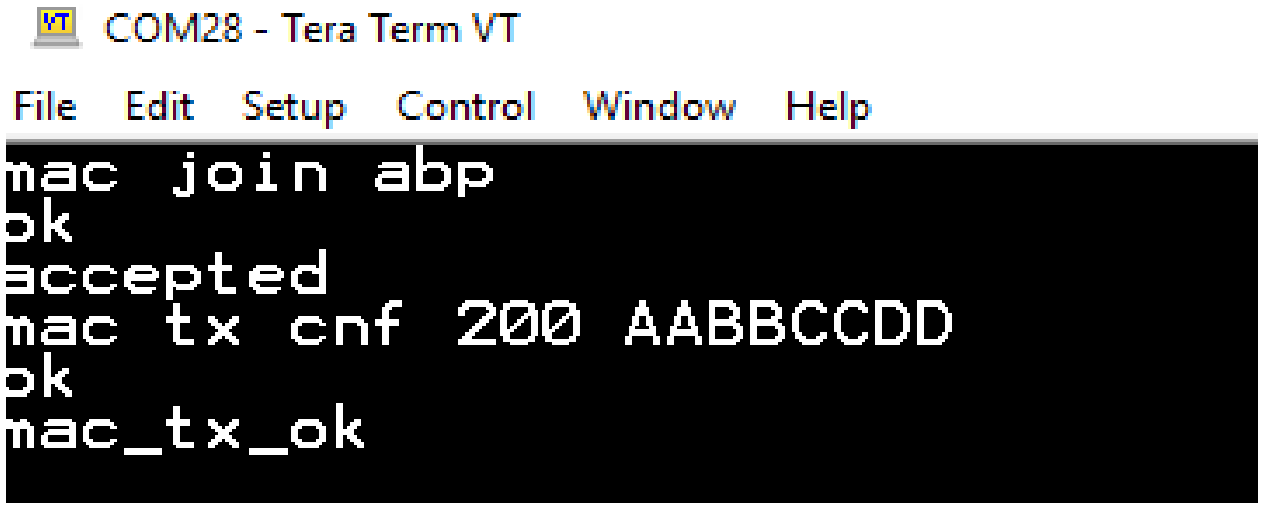

**Note: ABP**

Set devaddr

Set nwkskey

Set appskey

# Command mote board via USB cable.

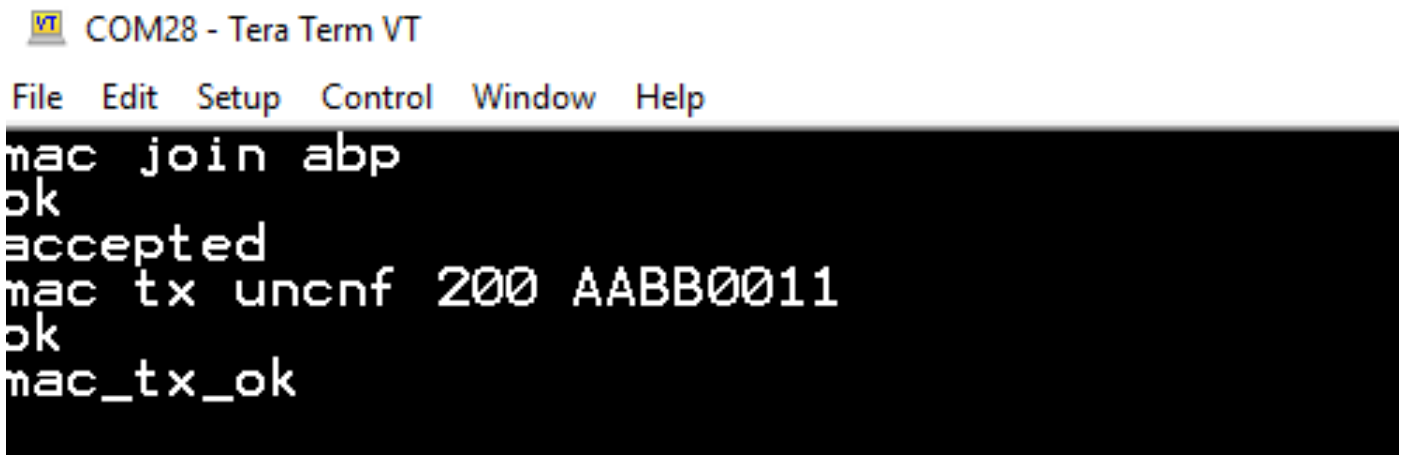



## ABP Joining

- Mac join abp
- Mac tx cnf port xxxxxx //Confirm

Note: xxxxxx is message. (payload)

# Command mote board via USB cable.




## ABP Joining

- Mac join abp
- Mac tx uncnf port xxxxxx //Unconfirm

Note: xxxxxx is message. (payload)

# Command mote board via USB cable.

Received data

DOWNLOAD

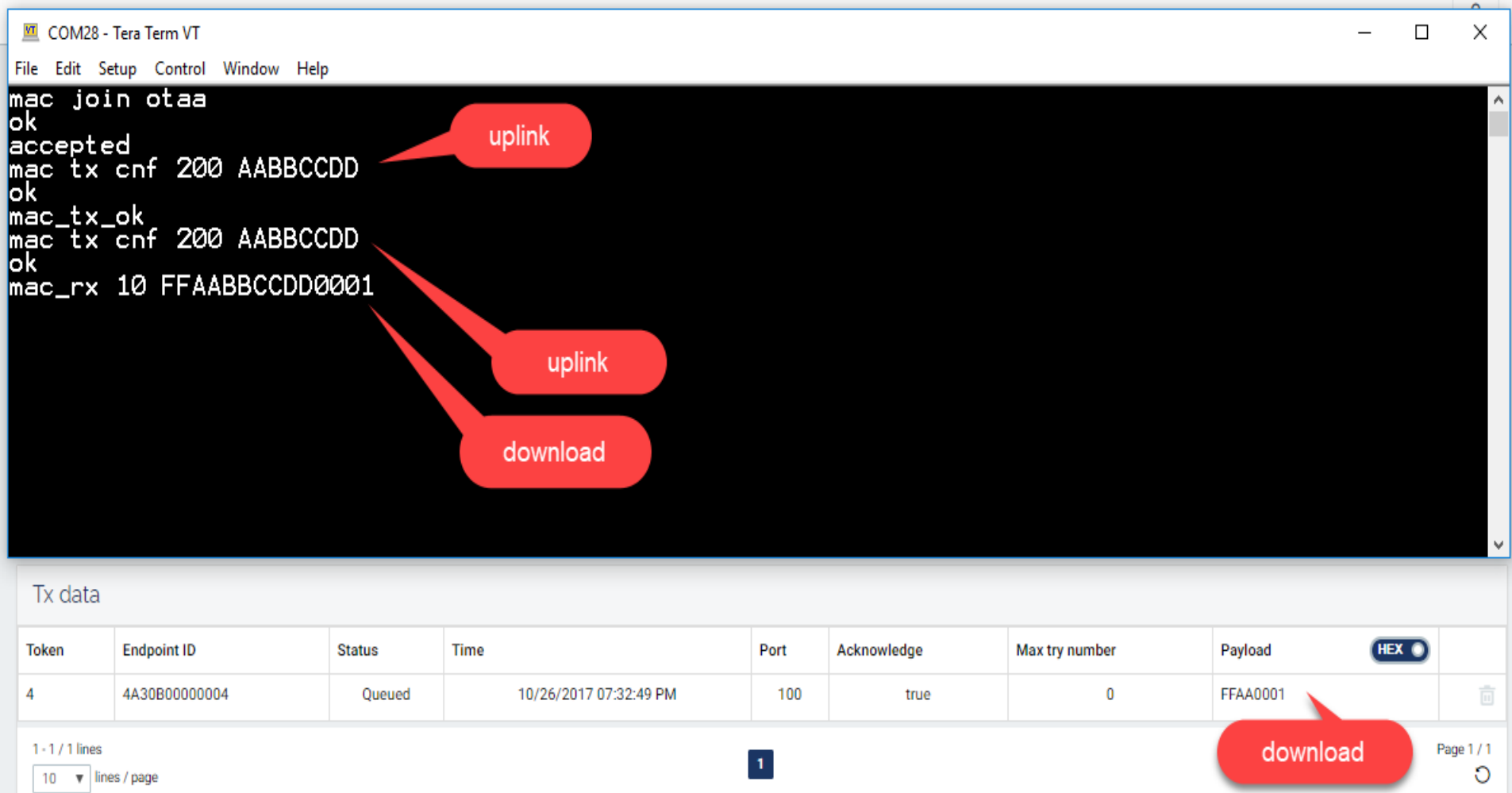
| Id | Endpoint ID | Received time | Sequence number | Port | Radio ID | Channel | SNR | RSSI | Frequency | Modulation | Data Rate | Coding rate | Payload HEX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 41 | ABCD0004 | 10/26/2017 07:27:41 PM | 2 | 200 | 0 | 4 | 9.5 dB | -27 dBm | 921.600 MHz | LoRa | SF12BW125 | 4/5 | AABB0011 |
| 40 | ABCD0004 | 10/26/2017 07:26:01 PM | 1 | 200 | 0 | 0 | 8.5 dB | -26 dBm | 923.200 MHz | LoRa | SF12BW125 | 4/5 | AABBCCDD |
| 39 | 4A30B00000004 | 10/26/2017 07:17:43 PM | 0 | 200 | 0 | 6 | 10.5 dB | -24 dBm | 922.000 MHz | LoRa | SF12BW125 | 4/5 | AABB0011 |
| 38 | 4A30B00000004 | 10/26/2017 07:16:55 PM | 1 | 200 | 0 | 7 | 9.2 dB | -25 dBm | 922.200 MHz | LoRa | SF12BW125 | 4/5 | AABB0011 |
| 37 | 4A30B00000004 | 10/26/2017 07:16:49 PM | 0 | 200 | 0 | 2 | 10 dB | -26 dBm | 923.000 MHz | LoRa | SF12BW125 | 4/5 | AABBCCDD |
| 36 | 4A30B00000004 | 10/26/2017 07:13:57 PM | 0 | 200 | 0 | 3 | 9.5 dB | -28 dBm | 922.800 MHz | LoRa | SF12BW125 | 4/5 | AABBCCDD |

**Note:** Result of Tx confirm and unconfirm. (ABP)

**Payload :AABBCCDD is confirmed.**
**Payload :AABB0011 in unconfirmed.**

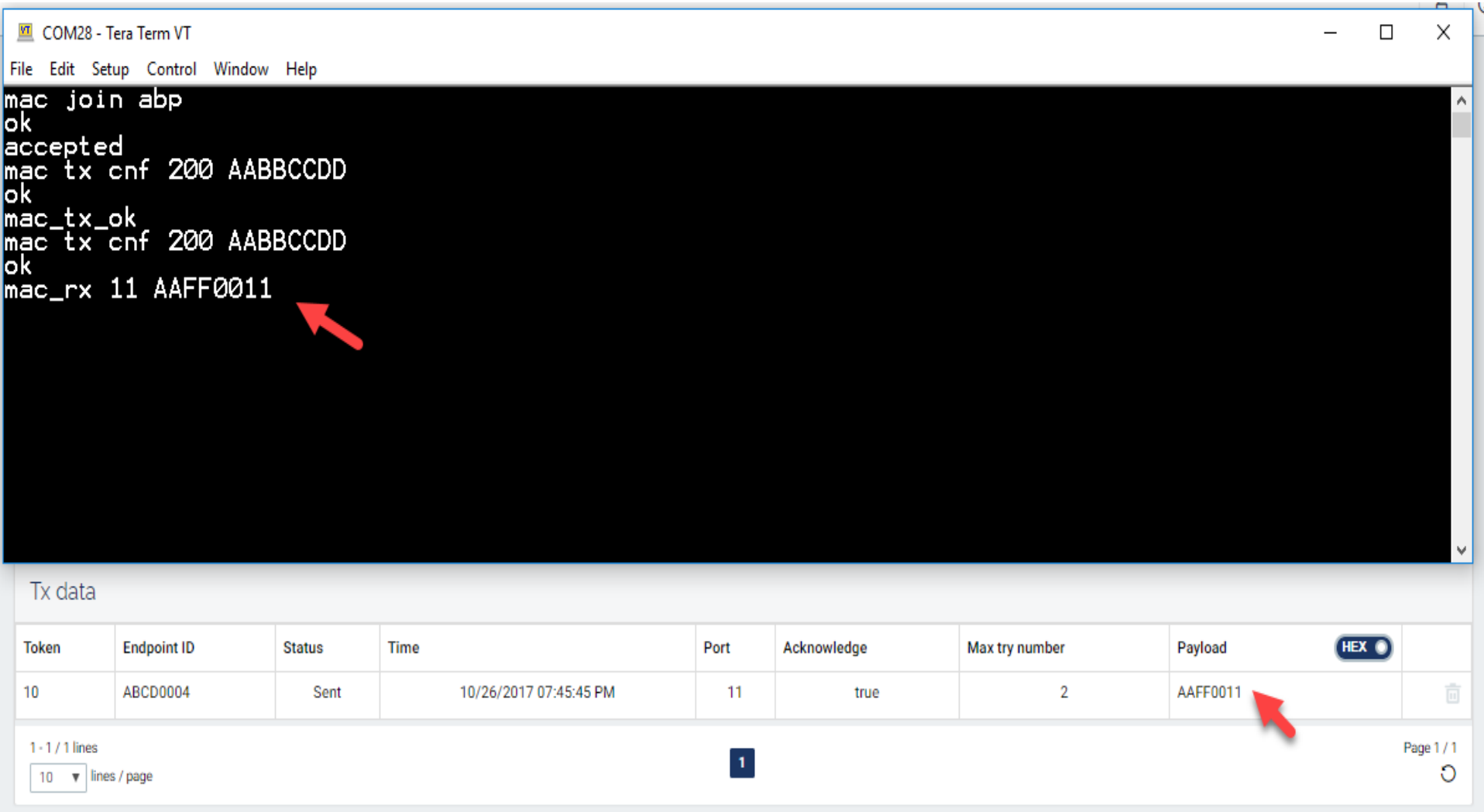
# Command mote board via USB cable.

## Uplink and downlink by OTAA



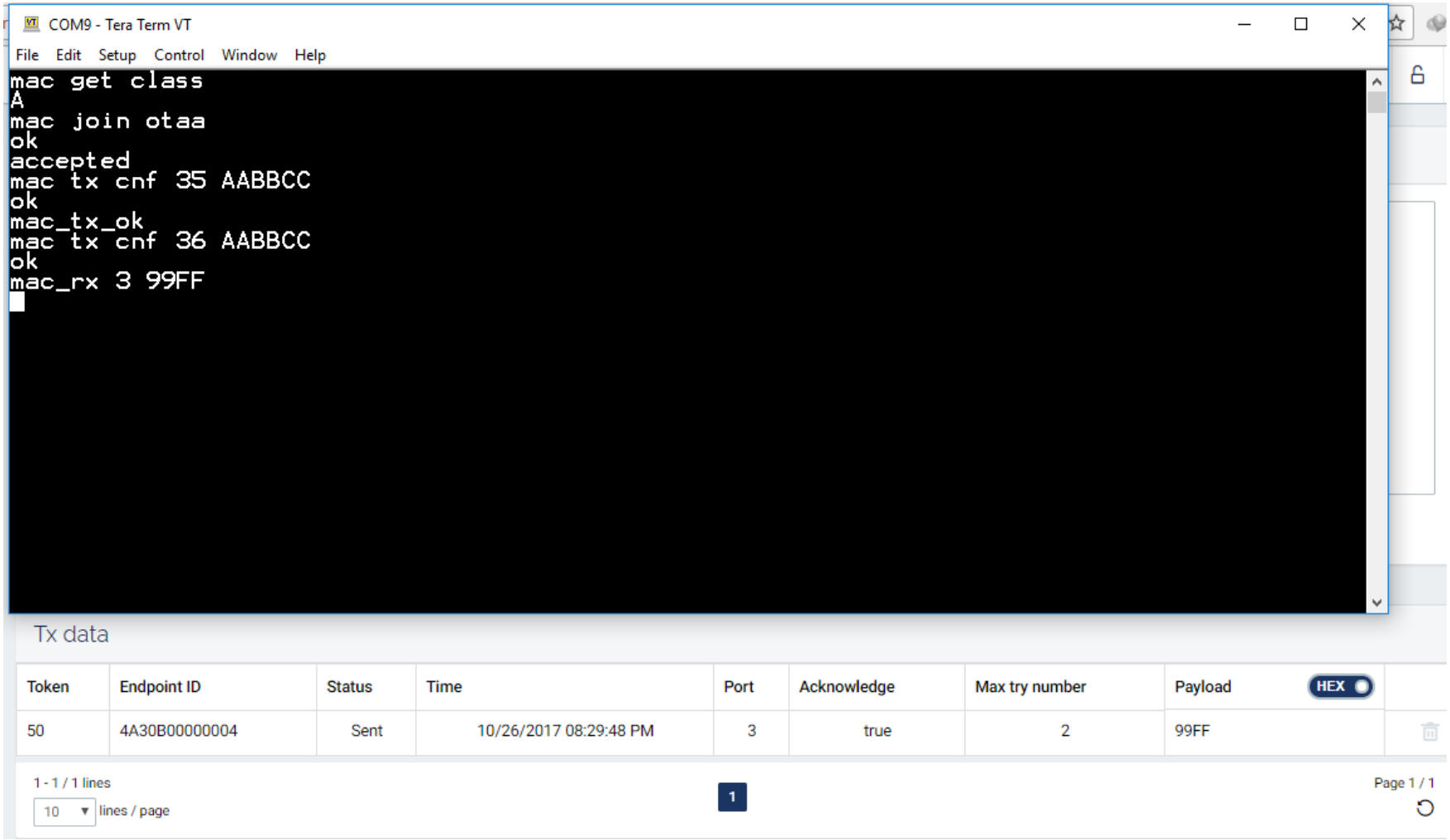
# Command mote board via USB cable.

## Uplink and downlink by ABP



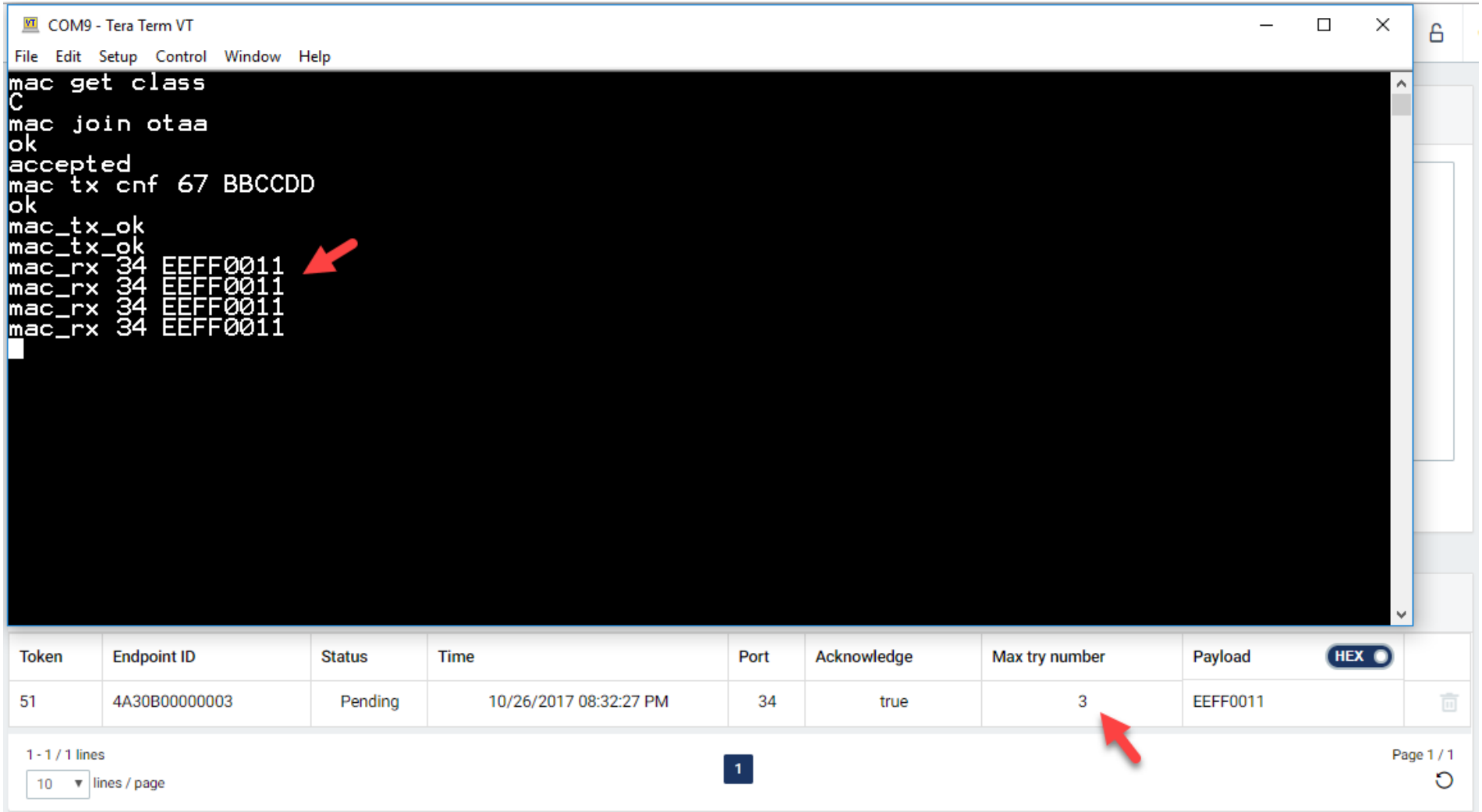
# Command mote board via USB cable.

## Tx and Rx on Class A



# Command mote board via USB cable.

## Tx and Rx on Class C



# RN2903A Command detail

# Reference manual:

1. RN2903 LoRa Technology Module Command Reference
2. RN2483 LoRa Technology Module Command Reference
3. LoRa Technology Mote User's Guide

*Please open reference manual !!!*

# Command mote board via USB cable.

## mac tx <type> <portno> <data>

**<type>:** string representing the uplink payload type, either cnf or uncnf (cnf – confirmed, uncnf – unconfirmed)

**<portno>:** decimal number representing the port number, from 1 to 223

**<data>:** hexadecimal value. The length of <data> bytes capable of being transmitted are dependent upon the set data rate (please refer to the *LoRaWAN™ Specification* for further details).

# Command mote board via USB cable.

## Response after entering the command:

- ok – if parameters and configurations are valid and the packet was forwarded to the radio transceiver for transmission
- invalid_param – if parameters (<type> <portno> <data>) are not valid
- not_joined – if the network is not joined
- **no_free_ch** – if all channels are busy

- silent – if the module is in a Silent Immediately state
- frame_counter_err_rejoin_needed – if the frame counter rolled over

- **busy** – if MAC state is not in an Idle state
- mac_paused – if MAC was paused and not resumed back
- **invalid_data_len** if application payload length is greater than the maximum application payload length corresponding to the current data rate

# Command mote board via USB cable.

## Response after the uplink transmission:

- mac_tx_ok if uplink transmission was successful and no downlink data was received back from the server;
- mac_rx <portno> <data> if transmission was successful, <portno>: port number, from 1 to 223; <data>: hexadecimal value that was received from the server;

- mac_err if transmission was unsuccessful, ACK not received back from the server
- invalid_data_len if application payload length is greater than the maximum application payload length corresponding to the current data rate

# Command mote board via USB cable.

## mac join <mode>

**Response after entering the command:**

- ok – if parameters and configurations are valid and the join request packet was forwarded to the radio transceiver for transmission

- invalid_param – if <mode> is not valid
- keys_not_init – if the keys corresponding to the Join mode (otaa or abp) were not configured

- no_free_ch – if all channels are busy
- silent – if the device is in a Silent Immediately state
- busy – if MAC state is not in an Idle state
- mac_paused – if MAC was paused and not resumed back

# Command mote board via USB cable.

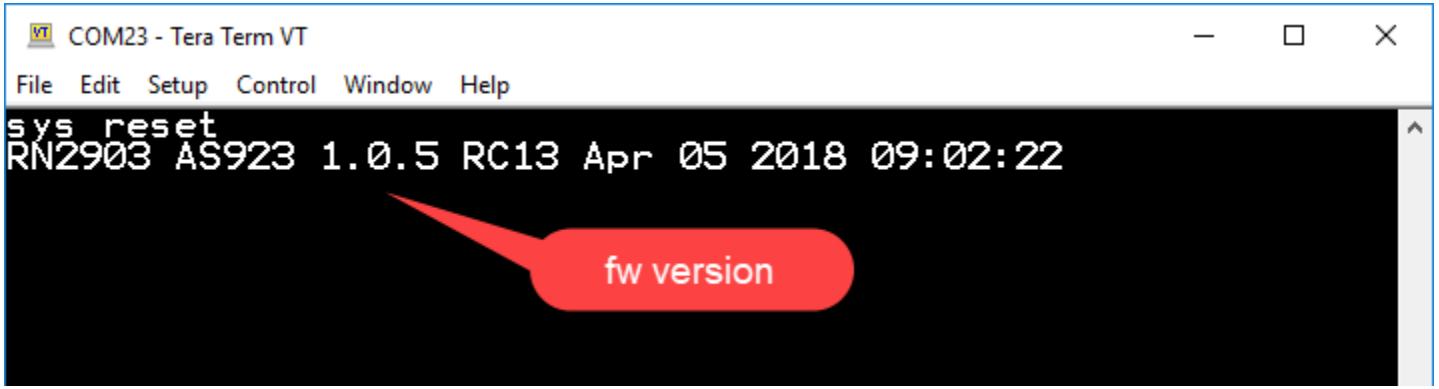
Response after the join procedure:

- denied if the join procedure was unsuccessful (the module attempted to join the network, but was rejected);

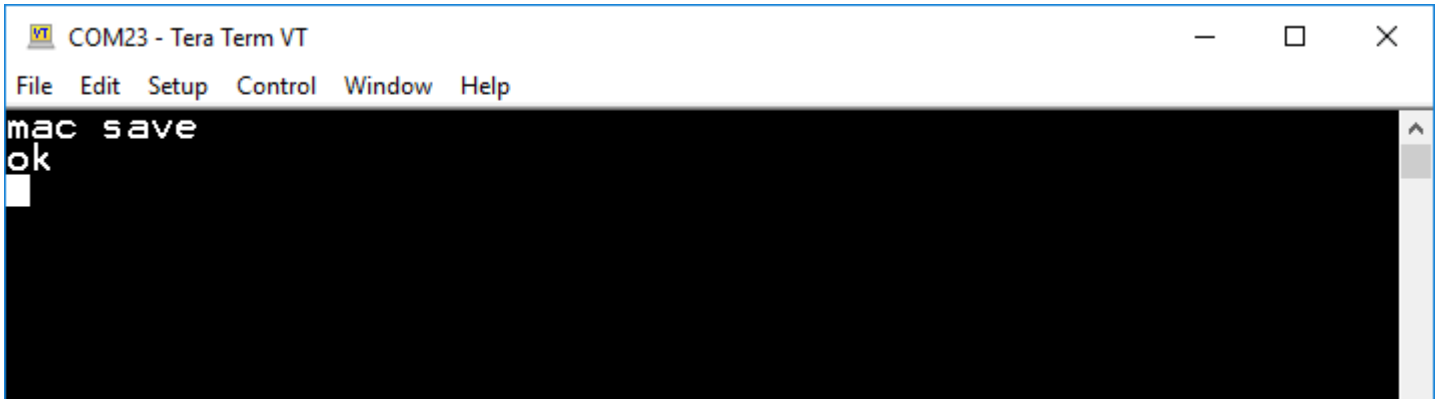
- accepted if the join procedure was successful;

# BASIC GET /Check information command

## sys reset




## mac save // save configuration



# BASIC GET /Check information command

**mac save**

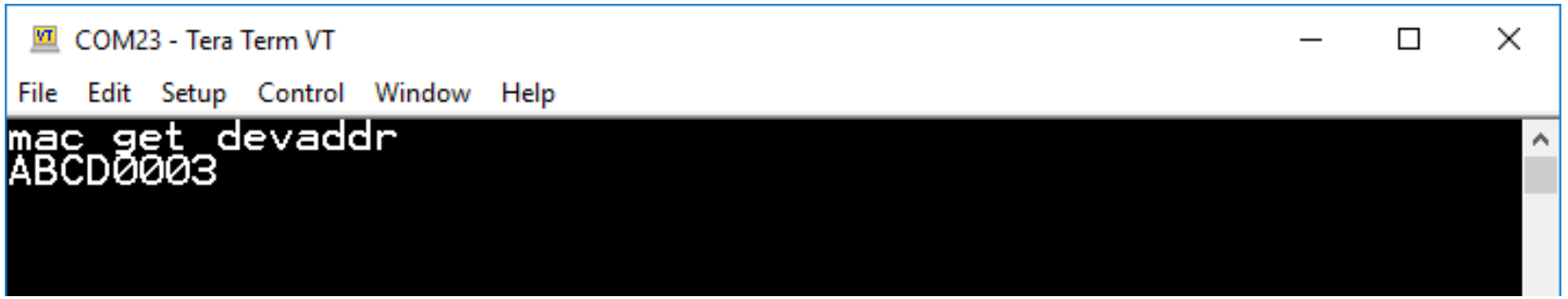
The LoRaWAN Class A protocol configuration savable parameters are:

- `band`: Band
- `fcntup`: Uplink Frame Counter
- `fcntdown`: Downlink Frame Counter
- `dr`: Data Rate
- `rx2dr`: Data Rate parameter for the second receive window
- `rx2freq`: Frequency parameter for the second receive window
- `adr`: Adaptive Data Rate state
- `deveui`: End-Device Identifier
- `appeui`: Application Identifier
- `appkey`: Application Key
- `nwkskey`: Network Session Key
- `appskey`: Application Session Key
- `devaddr`: End Device Address
- `ch`: All Channel Parameter
  - `freq`: Frequency
  - `dcycle`: Duty Cycle
  - `drrange`: Data Rate Range
  - `status`: Status

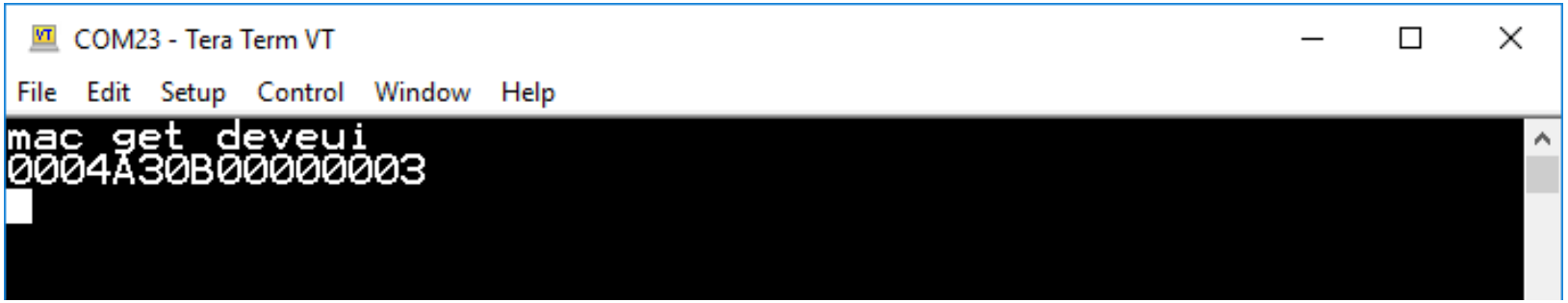
# Command mote board via USB cable.

## BASIC GET /Check information command

### mac get devaddr

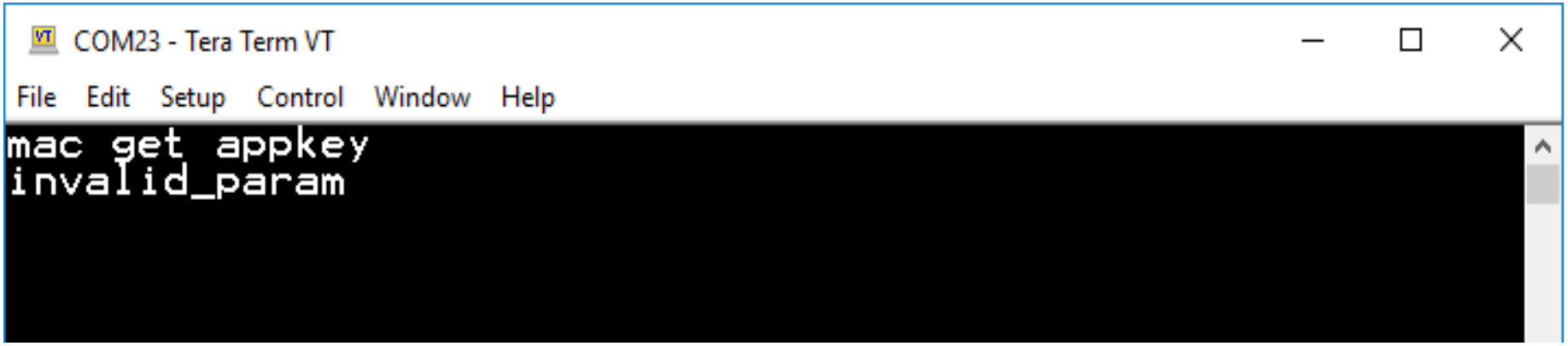


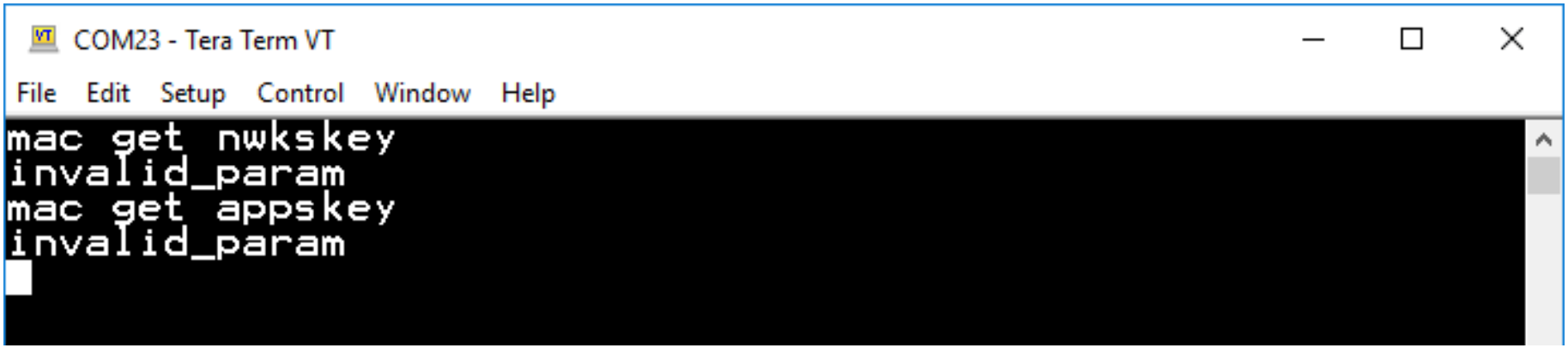

### mac get deveui



# BASIC GET /Check information command

## mac get appkey // Cannot read appkey !!!




**Note:** We cannot ready any Security Key !!!

# Command mote board via USB cable.

## MAC GET CHANNEL COMMANDS

**TABLE 2-9: MAC GET CHANNEL COMMANDS**

| Parameter | Description |
|---|---|
| `freq` | Gets the module operation frequency for the specified channel ID. |
| `dcycle` | Gets the module duty cycle used for transmission on the specified channel ID. |
| `drrange` | Gets the valid data rate range (min. to max.) allowed for the module on the specified channel ID |
| `status` | Gets the status for the specified channel ID to indicate if it is enabled for use. |

# COMMAND ORGANIZATION

**TABLE 2-1: COMMAND TYPES**

| Command Type | Keyword | Description |
|---|---|---|
| System | <sys> | Issues system level behavior actions, gathers status information on the firmware and hardware version, or accesses the module user EEPROM memory. |
| LoRaWAN™ Protocol | <mac> | Issues LoRaWAN protocol network communication behaviors, actions and configurations commands. |
| Transceiver commands | <radio> | Issues radio specific configurations, directly accessing and updating the transceiver setup. |

**Note:** Upon successful reception of commands, the module will respond with one of the following:

- ok
- invalid_param
- Requested Information
- Descriptive Error Message

## 2.4.8.19.1 `mac set ch freq <channelID> <frequency>`

`<channelId>`: decimal number representing the channel number, from 3 to 15.

`<frequency>`: decimal number representing the frequency, from 863000000 to 870000000 or from 433050000 to 434790000, in Hz.

Response: `ok` if parameters are valid

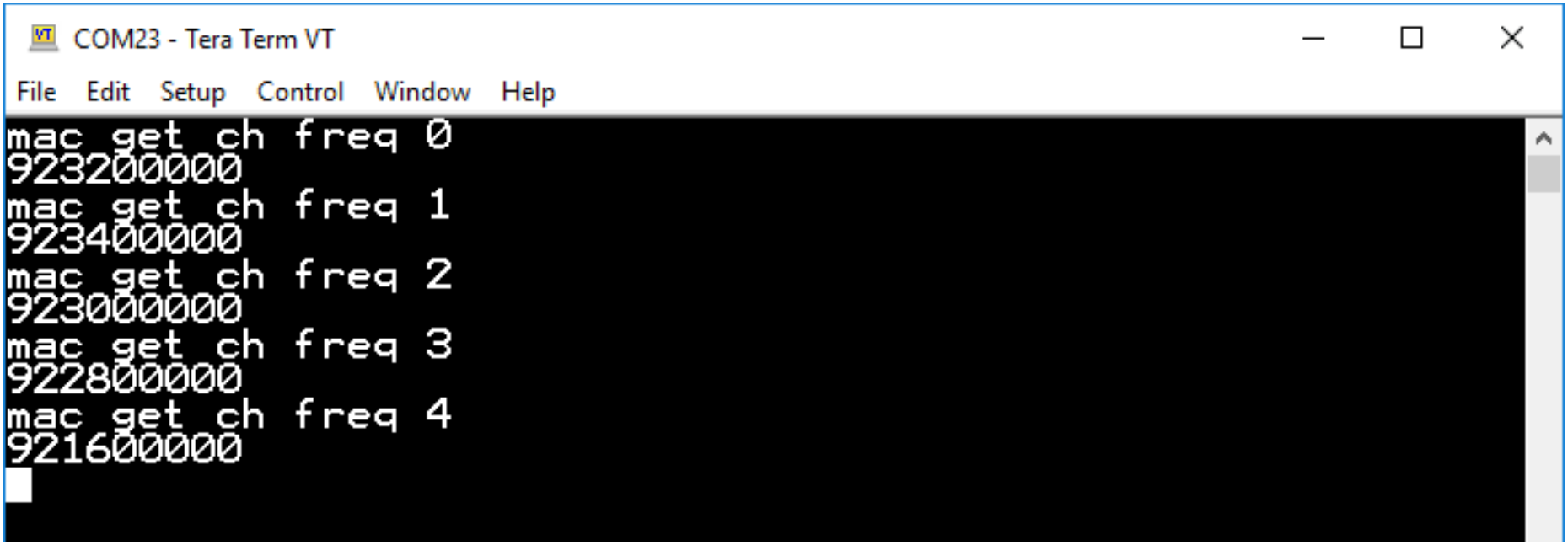
`invalid_param` if parameters are not valid

This command sets the operational frequency on the given channel ID. The default channels (0-2) cannot be modified in terms of frequency.

Example: `mac set ch freq 13 864000000` // Define frequency for channel 13 to be 864 MHz.

# BASIC GET /Check information command

- mac get ch freq 0
- mac get ch freq 1
- mac get ch freq 2
- mac get ch freq 3
- mac get ch freq 4



2.4.8.19.2 `mac set ch dcycle <channelID> <dutyCycle>`

`<channelId>`: decimal number representing the channel number, from 0 to 15.

`<dutyCycle>`: decimal number representing the duty cycle, from 0 to 65535.

Response: `ok` if parameters are valid

`invalid_param` if parameters are not valid

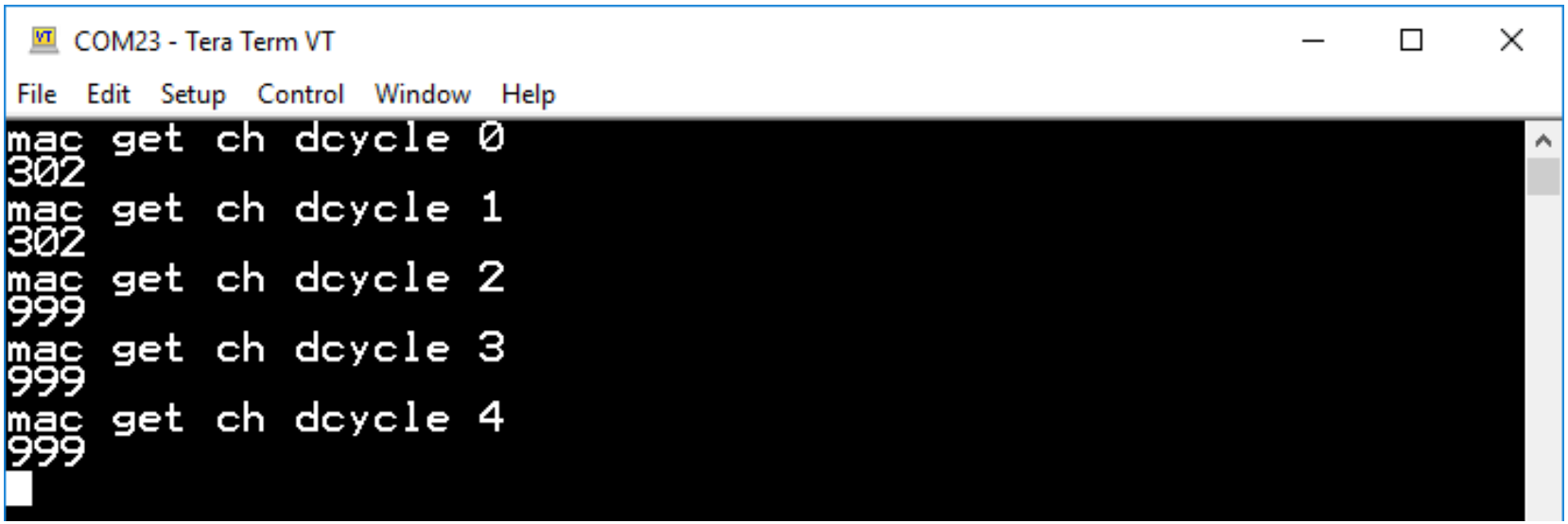
This command sets the duty cycle used on the given channel ID on the module. The `<dutyCycle>` value that needs to be configured can be obtained from the actual duty cycle X (in percentage) using the following formula: `<dutyCycle>` = (100/X) – 1. The default settings consider only the three default channels (0-2), and their default duty cycle is 0.33%. If a new channel is created either by the server or by the user, all the channels (including the default ones) must be updated by the user in terms of duty cycle to comply with the ETSI regulations.

Example: **`mac set ch dcycle 13 9`** // Defines duty cycle for channel 13 to be 10%. Since (100/10) – 1 = 9, the parameter that gets configured is 9.

**Note:** If this parameter was previously saved to user EEPROM by issuing the `mac save` command, after modifying its value, the `mac save` command should be called again.

# BASIC GET /Check information command

mac get ch dcycle <channelId>




**Note:**

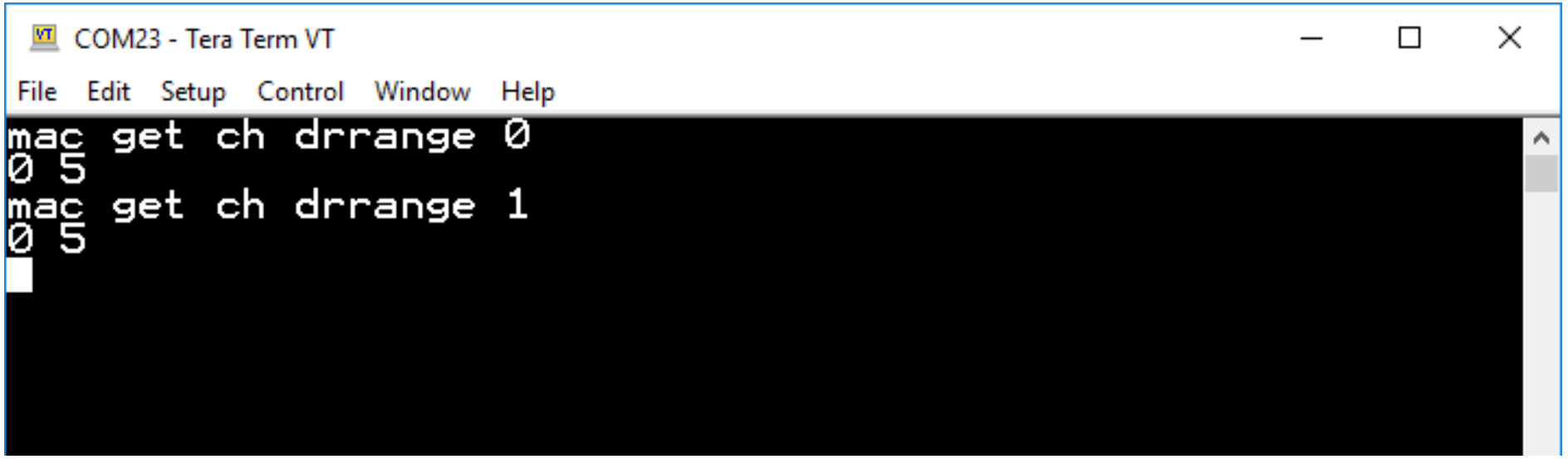
percentage is 100/(99 + 1) = 1 %

percentage is 100/(999 + 1) = 0.1%

percentage is 100/(303 + 1) = 0.33%

# BASIC GET /Check information command

mac get ch drrange <channelId>

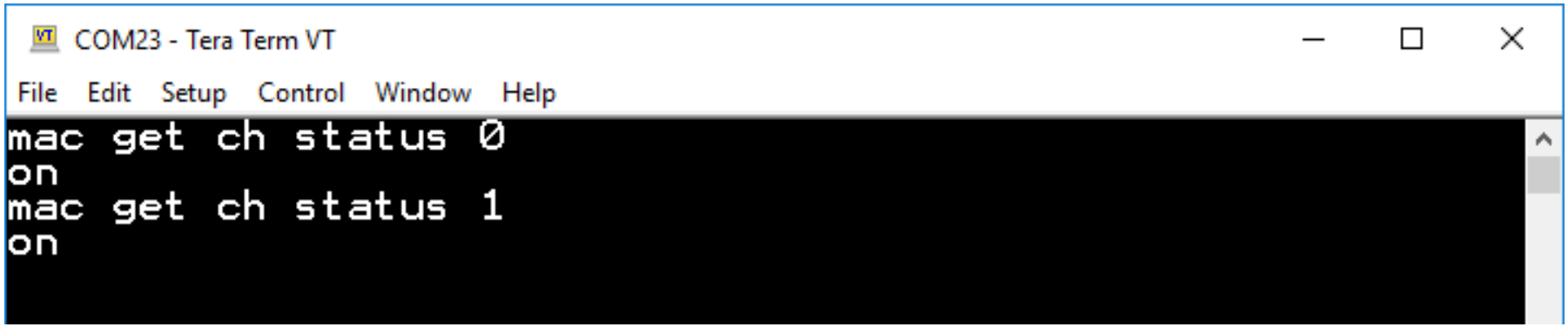



**Note:**

Please refer to the *LoRaWAN™ Specification* for the description of data rates and the corresponding spreading factors.

# BASIC GET /Check information command

mac get ch status <channelId>



# BASIC GET /Check information command

**sys set pinmode <pinname> <pinFunc>**

**<pinname>:** string representing the pin. Parameters can be: GPIO0 - GPIO13, UART_CTS, UART_RTS, TEST0, TEST1

**<pinFunc>:** string representing the function of the pin. Parameters can be: digout, digin or ana.

**Response:** ok if the parameters are valid
invalid_param if the parameters are not valid

**Example**:

sys set pinmode GPIO0 ana  // PORT GPIO0 analog
sys set pinmode GPIO0 digout  // PORT GPIO0 digital out

# BASIC GET /Check information command

**sys set pindig <pinname> <pinstate>**

<pinname>: string representing the pin. Parameter values can be: GPIO0 - GPIO13, UART_CTS, UART_RTS, TEST0, TEST1

<pinstate>: decimal number representing the state. Parameter values can be: 0 or 1

**Response:** ok if the parameters (<pinname>, <pinstate>) are valid
invalid_param if the parameters (<pinname>, <pinstate>) are not valid

**Example: sys set pindig GPIO5 1 // Drives GPIO5 high 1, VDD.**

# Command mote board via USB cable.

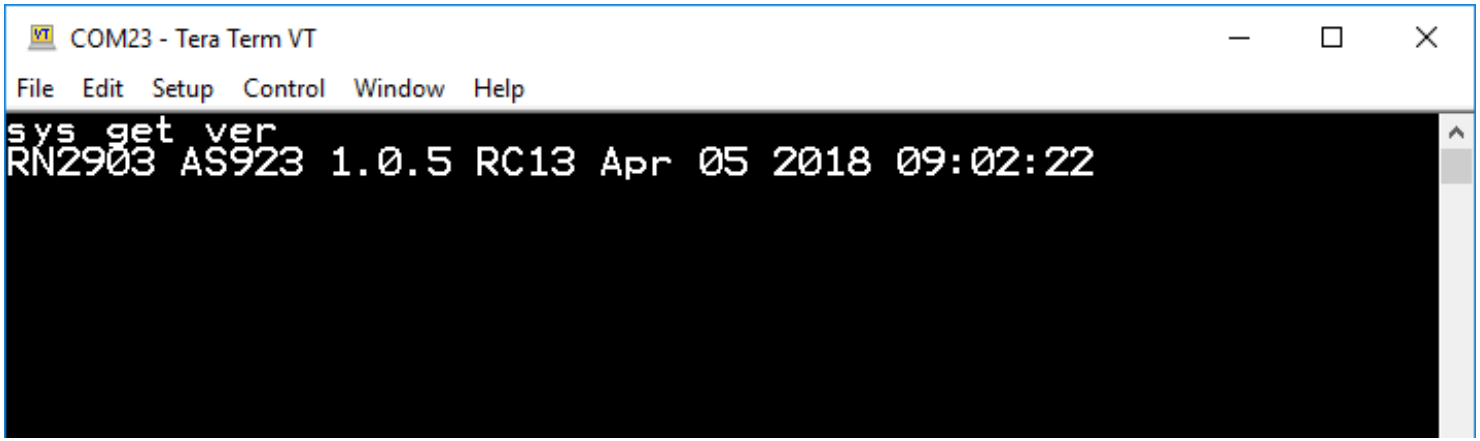
## 2.3.6 System Get Commands

**TABLE 2-4: SYSTEM GET COMMANDS**

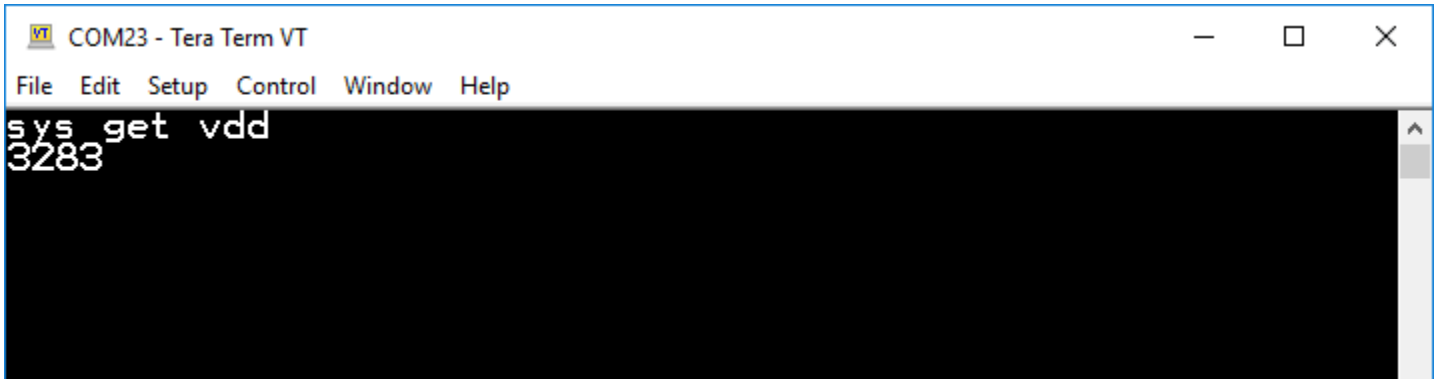
| Parameter | Description |
|---|---|
| ver | Returns the information on hardware platform, firmware version, release date. |
| nvm | Returns data from the requested user EEPROM <address>. |
| vdd | Returns measured voltage in mV. |
| hweui | Returns the preprogrammed EUI node address. |
| pindig | Returns the state of a digital input. |
| pinana | Returns the state of an analog input. |

# Command mote board via USB cable.

## sys get ver

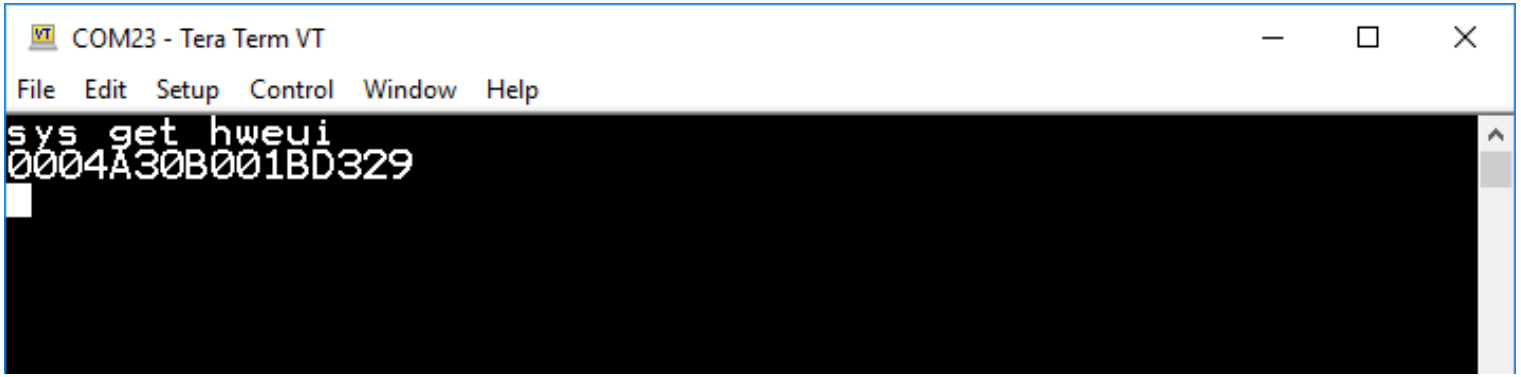



## sys get vdd



# Command mote board via USB cable.

sys get hweui




2.3.6.4 `sys get hweui`

Response: hexadecimal number representing the preprogrammed EUI node address

This command reads the preprogrammed EUI node address from the RN2903 module. The value returned by this command is a globally unique number provided by Microchip.

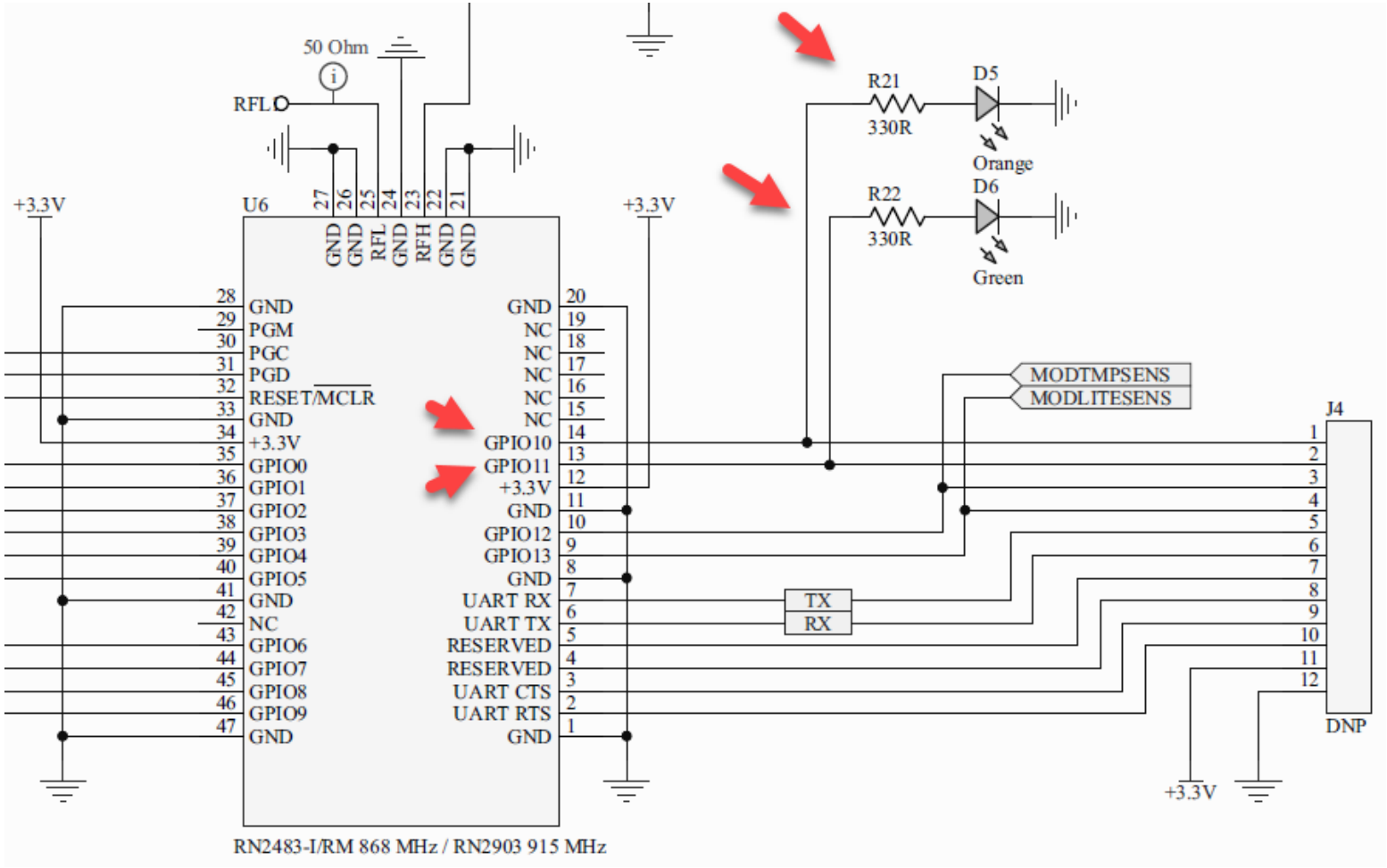
Example: **`sys get hweui`** // Reads the preprogrammed EUI node address.

**Note:** The preprogrammed EUI node address is a read-only value and cannot be changed or erased. This value can be used to configure the device EUI using the `mac set deveui` command (see Section 2.4.8.2).

# GPIO Control via serial command

# BASIC GET /Check information command

## GPIO Control



# Command mote board via USB cable.

## sys get pindig <pinname>

<pinname>: string representing the pin. Parameters can be: GPIO0 - GPIO13, UART_CTS, UART_RTS, TEST0, TEST1

Response: decimal number representing the state (either 0 or 1). This command allows the user to read the state of a digital input. To be used as a digital input, a pin needs to be configured using the sys set pinmode command.

**Example:** sys get pindig GPIO0 //Reads the state of the GPIO0 digital input

# Command mote board via USB cable.

## sys get pinana <pinname>

<pinname>: string representing the pin. Parameters can be: GPIO0 - GPIO3, GPIO5 - GPIO13

Response: decimal number representing the result of the conversion, from 0 to 1023,
where 0 represents 0V and 1023 is VDD, the supply voltage of the module.

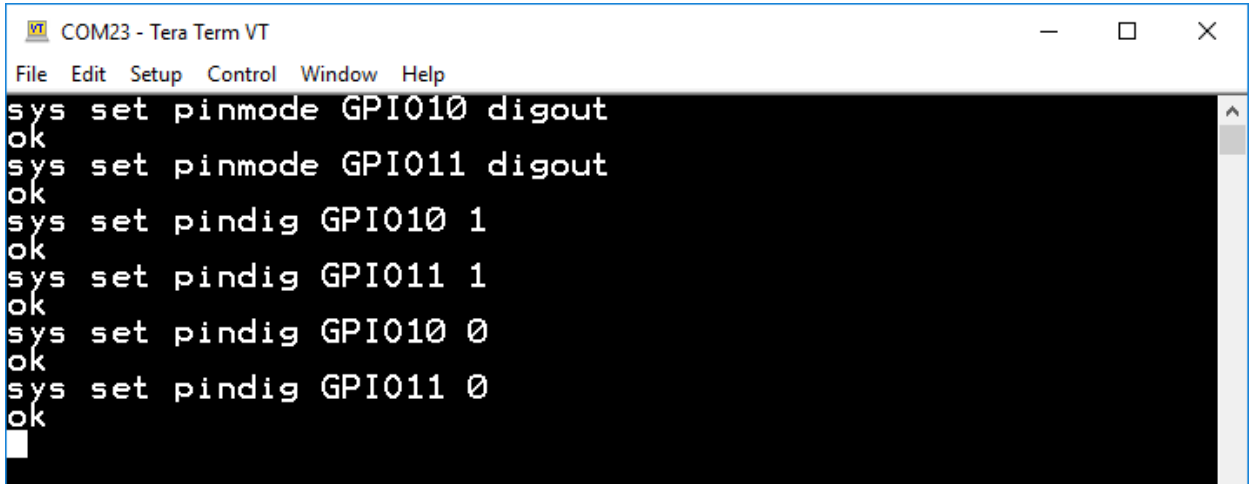
Example: sys get pinana GPIO0 //Reads the state of the GPIO0 analog input

# Command mote board via USB cable.

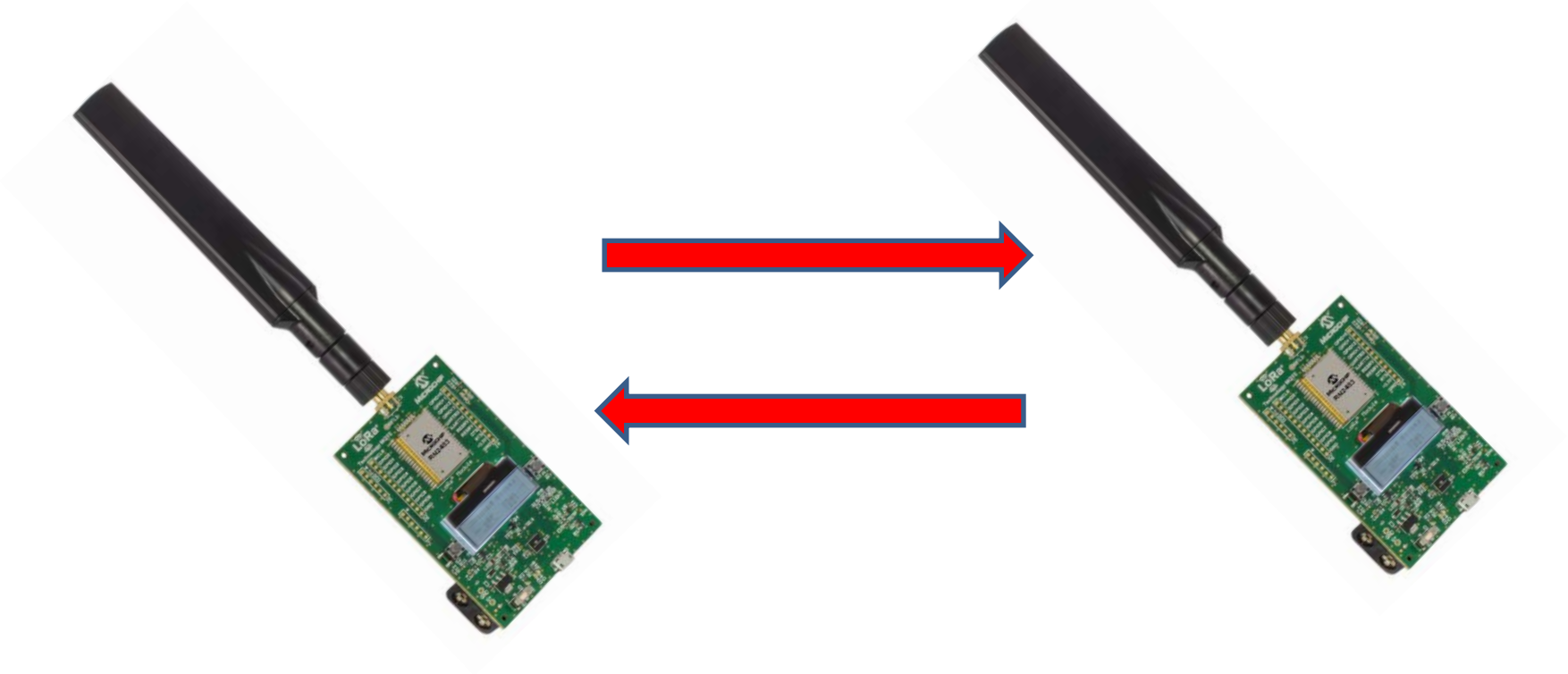
- sys set pinmode GPIO10 digout
- sys set pinmode GPIO11 digout

- sys set pindig GPIO10 1  // GPIO 10 = “1”
- sys set pindig GPIO11 1

- sys set pindig GPIO10 0 // GPIO 10 = “0”
- sys set pindig GPIO11 0



# FSK and LoRa mode (P2P)



**HARD WARE**

-RN2903A MOTE x 2 sets

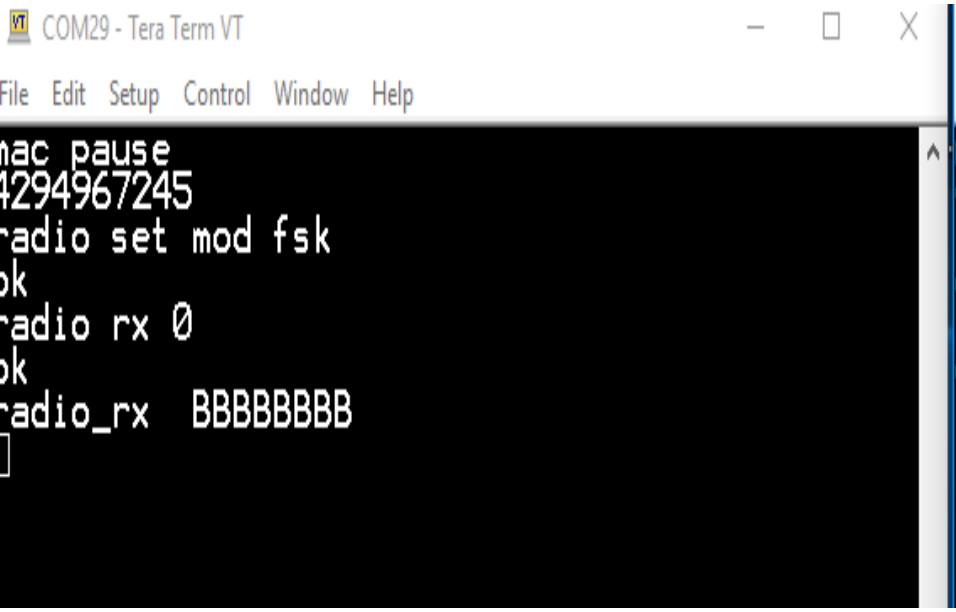
**Software**

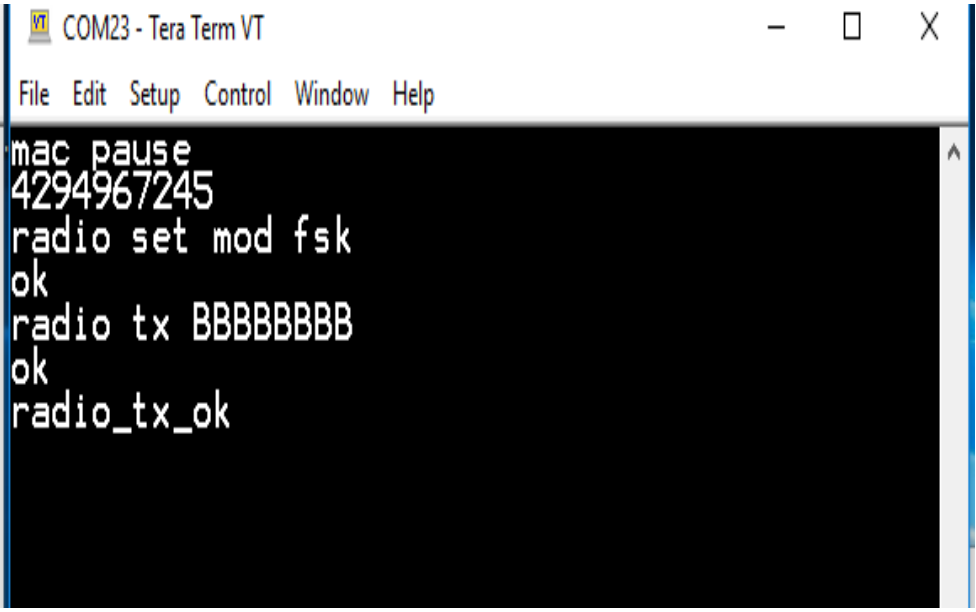
-Teraterm

# Command list:

1.mac pause
2.radio set mod fsk

- Transmitter : radio tx BBBBBBBB
- Receiver: radio rx

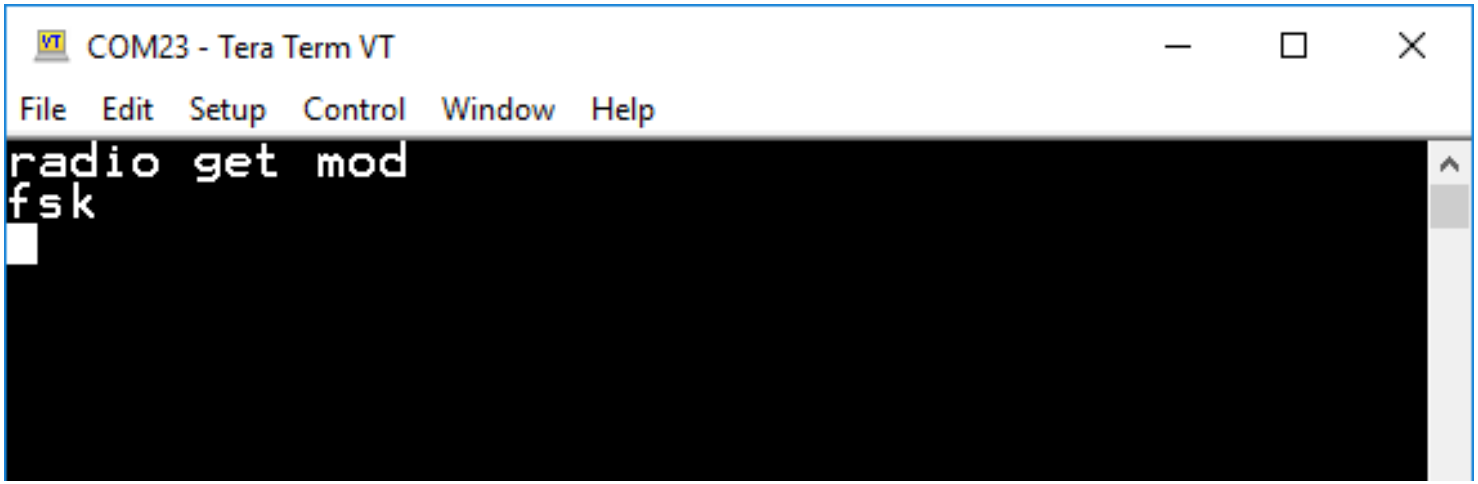


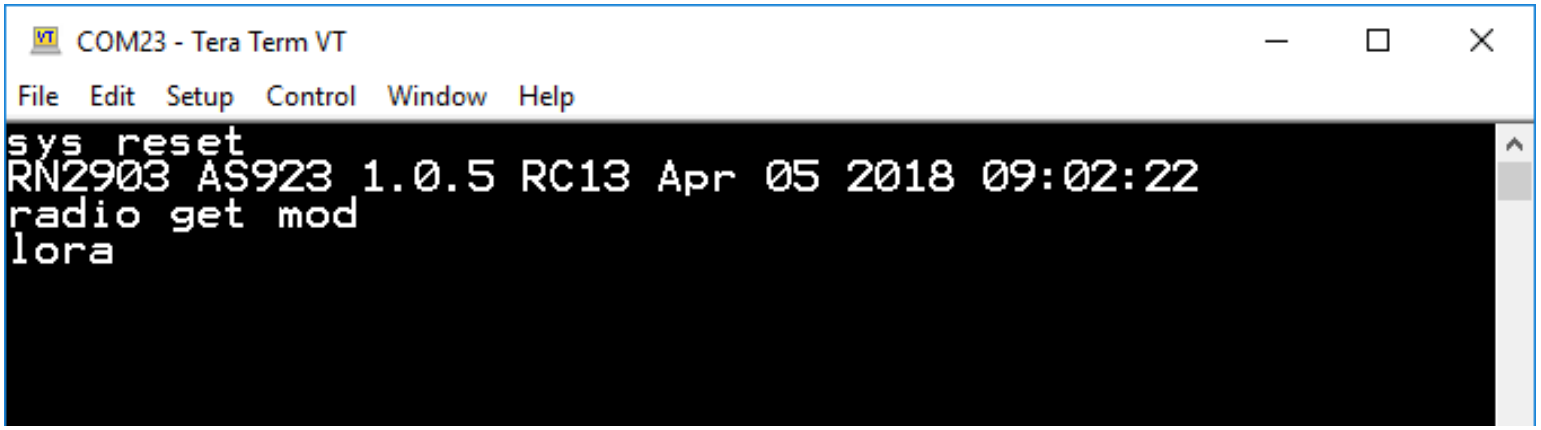




**Note:**

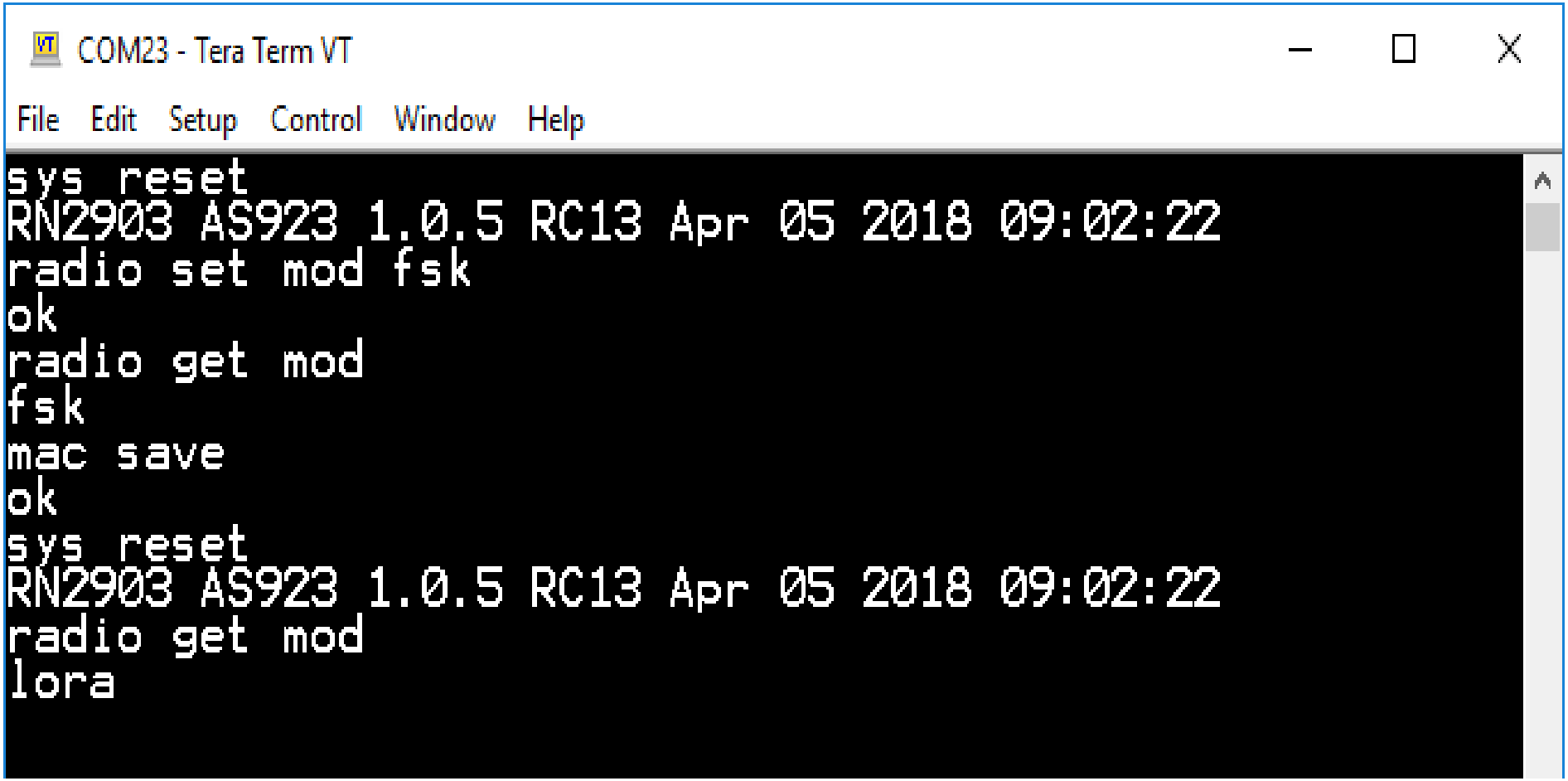
COM29 is receiver and COM23 is transmitter.

# radio get mod



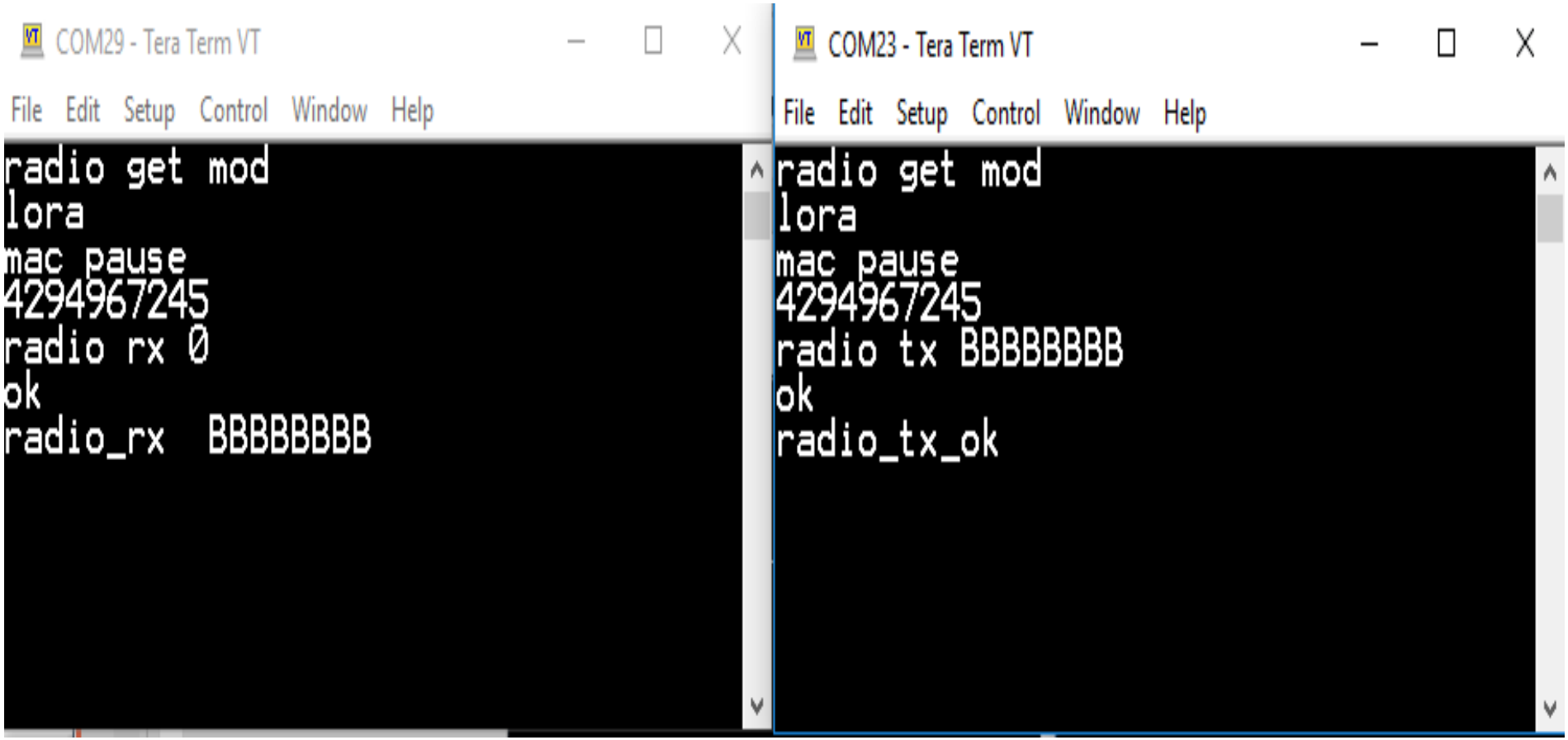




**Note**: After sys reset or power on reset, radio mode will be lora (default).




**Note**: After sys reset or power on reset, radio mode will be lora (default).

# Lora on P2P Testing.




Note: COM29 is Receiver and COM23 is transmitter.

# Homework Preparation

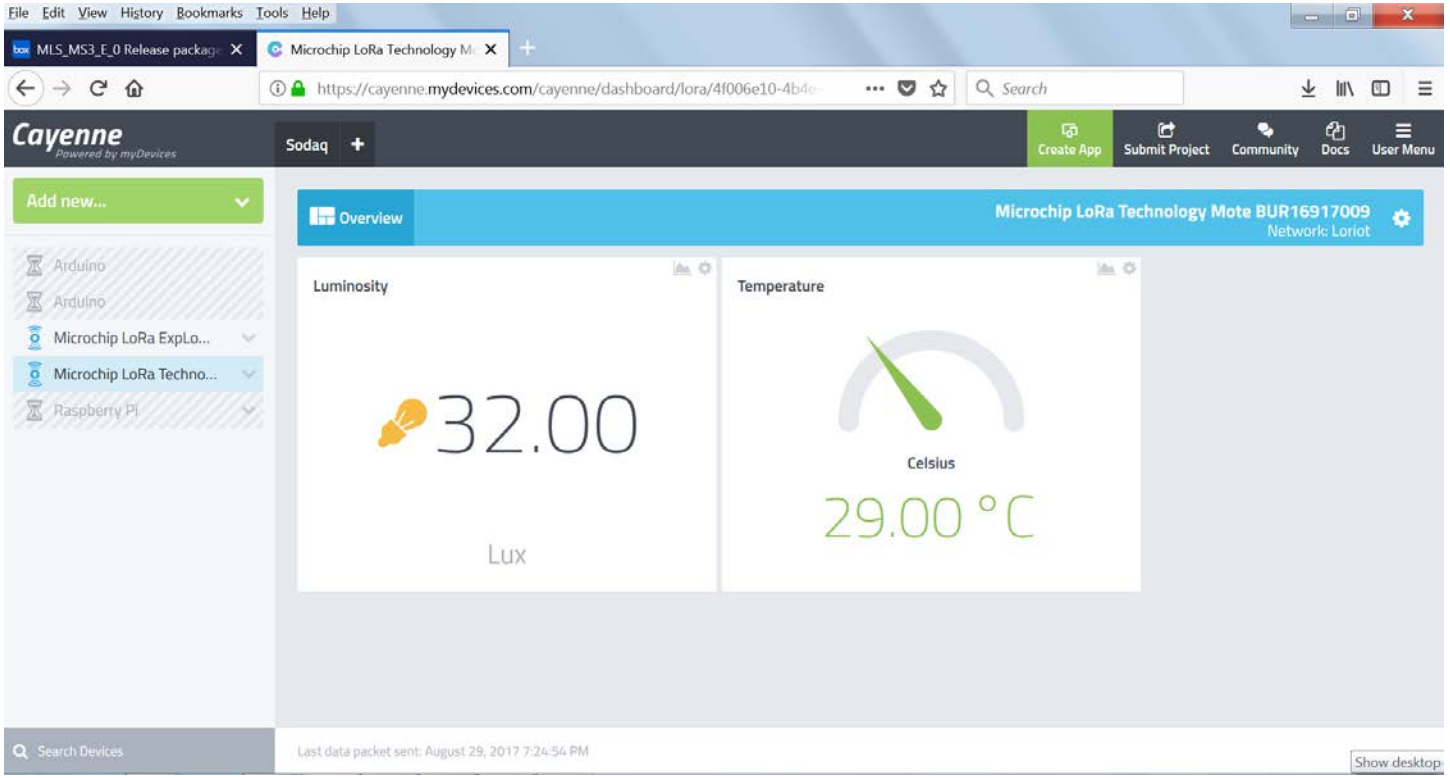
- **You will need:**
  - A Laptop with USB port
  - Microchip USB driver (can auto-install, but slow)
    - www.Microchip.com/MCP2200 (under documentation tab)
  - Any generic “Terminal” app (*but not PuTTY*)
    - Termite (http://termite.soft112.com/ )
    - **TeraTerm (http://teraterm.software.informer.com/ ) - Preferred**
    - Coolterm (www.macupdate.com/app/mac/31352/coolterm )
    - Etc
  - **Settings: 57600bps, 8n1, no flow control, echo on, set options to include CR+LF**

# MyDevice Dashboard

- **Create a login ID account.**

**https://cayenne.mydevices.com/cayenne/login**

- **You can add 10 devices onto your account free.**



# Additional Resources

- http://lora-alliance.org/

- http://www.microchip.com/RN2483
- RN2483 LoRa® Transceiver Module Datasheet
- RN2483 LoRa Command Reference User's Guide

- http://www.microchip.com/RN2903
- RN2903 LoRa Transceiver Module Datasheet
- RN2903 LoRa Command Reference User's Guide

# THANK YOU

**Email: Natapol@es.co.th**

***ELECTRONICS SOURCE CO., LTD*** ***www.es.co.th***