UM2786

User manual

STM32CubeWL Nucleo demonstration firmware

Introduction

STM32Cube is an STMicroelectronics original initiative to significantly improve designer's productivity by reducing development effort, time, and cost. STM32Cube covers the whole STM32 portfolio.

STM32Cube includes:

- A set of user-friendly software development tools to cover project development from conception to realization, among which are:
  - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
  - STM32CubeIDE, an all-in-one development tool with peripheral configuration, code generation, code compilation, and debug features
  - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
  - STM32CubeMonitor-Power (STM32CubeMonPwr), a monitoring tool to measure and help in the optimization of the power consumption of the MCU
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeWL for the STM32WL Series), which include:
  - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
  - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over the HW
  - A consistent set of middleware components such as FAT file system, RTOS, mbed Crypto, sub-GHz physical layer, STM32 Secure Engine, and STM32 key management services
  - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with:
  - Middleware extensions and applicative layers
  - Examples running on some specific STMicroelectronics development boards

The STM32CubeWL Nucleo demonstration firmware is built around the STM32Cube hardware abstraction layer (HAL) and low-layer (LL) APIs, and board support package (BSP) components.

The STM32CubeWL Nucleo demonstration firmware features a local network using LoRa® communication settings including one concentrator and up to 14 sensors that can connect and send data to the concentrator.

The STM32WL Nucleo-64 board order codes compatible with the STM32CubeWL Nucleo demonstration firmware are listed in the NUCLEO-WL55JC ordering information table.

A concentrator sends one beacon frame and one sync frame every 16 seconds to administrate a network of up to 14 sensors and receives each connected sensor data. The concentrator can be connected to STM32CubeMonitor to configure the geographical area and display the list of sensors detected and connected sensor data.

Both concentrator and sensor are STM32WL Nucleo-64 boards with corresponding embedded software. Refer to section 2.1.1 Demonstration set-up overview of the application note *RF certification process for NUCLEO-WL55JC* (AN5566) for further details.

1 Acronyms

Table 1. Acronym definitions

Acronym	Definition
CAD	channel activity detection
CRC	cyclic redundancy check
DR	data rate
EUI	extended unique identifier
FH	frequency hopping
FSK	frequency-shift keying
LBT	listen before talk
MLCG	Mersenne linear congruential generator

2 Local network demonstration overview

2.1 Setup

In the local network, there is one concentrator and up to 14 sensors. The concentrator is silent until it gets a command from the connected PC to start sending a beacon on one of the beacon frequencies. The frequency band is selected depending on the region.

At sensor reset, the sensor continuously scans for beacon sweeping on all supported beacon frequencies until it finds one valid beacon. Once it finds a beacon, it listens for the sync to know which time slots are free. The sensor chooses one of the free time slots to respond to both control information and its sensor readouts.

For a possibility of future development, it is decided that the beacon is composed of two parts. The first part, named beacon, contains only a preamble, a subregion number, and a frequency seed. Then, there is a second part, named sync, with a standard preamble and a lot more data for the management of the local network.

STM32CubeMonitor may be used as a user interface. All radio management is done in the concentrator.

Local network projects are available in the STM32CubeWL MCU Package. They are located under `Projects\NUCLEO-WL55JC\Demonstrations\LocalNetwork`.

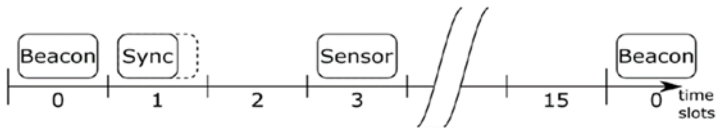
2.2 Time slots

There are 16 slots, each 1 second long. The 16 windows second window is called a superframe.

- slot 0 is concentrator beacon,
- slot 1 is for concentrator sync, the rest is for sensors.
- slots 2 through 15 are used to receive sensor data.

Depending on the region, slots 1 through 15 may use frequency hopping. Sync packet is usually shorter than the full slot.

Figure 1. Time slots



2.3 Beacon

Since the sensor is scanning multiple region beacon frequency, the preamble must last the longest time possible to maximize the synchronization probability. The beacon is mostly a preamble for the sensors to find the network. The beacon modulation needs to be specific for the USA and the rest of the world.

2.3.1 Frequency hopping region

The beacon needs to fit inside US 0.4 s dwell time. For that reason, the frequency hopping (FH) region-specific settings for LoRa® do not allow for the slowest LoRa® communication. Since the beacon cannot hop (to make sure that the sensor can find it), signal bandwidth must be greater or equal to 500 kHz. Still, the modulation must be the most robust possible to ensure that any sensor receives it. LoRa® SF12 with 500 kHz is chosen.

For SF12 at 500 kHz:

$$0.4 \div \left(2^{12} \div 500000\right) > 48$$

48 symbols are available. Using packet calculator, optimum settings are:

Table 2. US optimum settings

Parameter	Value
Payload size	4 bytes
Spread factor	SF12
CRC used	no
Explicit header	no
Low DR optimize	no
Coding rate	4 / 5
Preamble symbols	36
Bandwidth	500 kHz

The packet length is 395.3 ms which remains below 400 ms.

2.3.2 Duty cycle and listen before talk region

In the EU, the 10% duty cycle frequency band does not fit 250 kHz modulation. Some other regions require bandwidth to be 200 kHz or less. Therefore, 125 kHz LoRa® bandwidth is chosen. Using packet calculator:

Table 3. EU optimum settings

Parameter	Value
Payload size	4 bytes
Spread factor	SF11
CRC used	no
Explicit header	no
Low DR optimize	yes
Coding rate	4 / 5
Preamble symbols	36
Bandwidth	125 kHz

The packet length is 790.5 ms.

2.3.3 Beacon data structure

There is a 36-symbol preamble followed by 4 bytes for data.

- There is subregion information that informs the sensor about details of the communication modulation.
- There is a seed used in some regions for hopping. The seed is regenerated for every superframe.
- There is a behavioral major version to prevent incompatible devices to connect. Concentrator behavioral version is 0 (other values are reserved).
- There is an offset number to calibrate the time frame in sensors.
- Lastly, there is a basic checksum as the beacon packet doesn't use the header and LoRa® CRC.

2.4 Sync

The purpose of sync is to provide more information about the network, to verify beacon information in CRCed packet and to publish coding changes (changes of used coding rate and modulation parameters) for sensor uplink packets. The sync uses LoRa® SF11 / 250 kHz modulation, but the packet format is different. The preamble length is the default, there is an explicit header and there is a CRC.

Note: Private network sync word is used.

2.4.1 Frequency hopping region sync duration

Table 4. **Optimum settings**

Parameter	Value
Payload size	27 bytes
Spread factor	SF11
CRC used	yes
Explicit header	yes
Low DR optimize	no
Coding rate	4 / 5
Preamble symbols	8
Bandwidth	250 kHz

The packet length is 370.7 ms.

2.4.2 Duty cycle and listen before talk region sync duration

Table 5. **Optimum settings**

Parameter	Value
Payload size	27 bytes
Spread factor	SF11
CRC used	yes
Explicit header	yes
Low DR optimize	yes
Coding rate	4 / 5
Preamble symbols	8
Bandwidth	125 kHz

The packet length is 823.3 ms.

2.4.3 Sync data structure

The calculated payload is usable data. The packet structure already includes the header and CRC.

The sync packet doesn’t contain the time offset. The time needs to be calibrated by the beacon.

The sync contains a mask of occupied slots. This mask shows which slots are used and which are free to be used by a new sensor.

The sync also contains an array which may be filled with coding and modulation change request per slot-EUI relations for the management of the sensor. Each sensor management structure has its header and differently sized payload.

The EUI refers in the document is a lower 4 bytes of a full EUI which is programmed (OTP) in the STM32WL Series microcontroller and printed on the kits as a sticker.

2.5 Concentrator behavior

At reset the concentrator toggling all LEDs, one after one.

The concentrator is silent and does nothing until a command from an external host. Then a green LED flashes on each time slot of the network. When the concentrator transmits the beacon and sync packet, it lights up the red LED for the duration of the transmit. When the concentrator receives something, a blue LED flashes at the end of the received packet.

2.6 Sensor behavior

2.6.1 Sensor packet

Sensor packets are sent between sync and the next beacon, meaning in slot 2 to 15. The sensor can transmit only if it successfully received the beacon and sync of the current period. The default coding is the same as the concentrator sync.

The packet needs to fit inside one slot and comply with the regional regulations. The payload must contain at least a physical address, the counter of sent packets, and the version. After that, there are sensor data which can be compressed to shorten the packet length.

2.6.2 Preamble scanning

The sensor scans for a beacon preamble on all possible beacon frequencies. When it finds one, it locks to the frequency band of that region until reset.

Channel Activity Detection (CAD) can catch preamble from 2 symbols.

We have 6 different regions. Region 1 uses a shorter symbol time and has a 329.7 ms preamble. In this time, the CAD needs to sniff all the other regions with longer symbols and region 1.

$$T_{scan} = T_{S1} \times 2 \times 5 + T_{S2} \times 2 \sim 180\ ms$$

where

$$T_{S1}\left(SF11/125\ kHz\right) = 2^{11}/125000 \sim 16\ ms$$

is the longer symbol time and

$$T_{S2}\left(SF12/500\ kHz\right) = 2^{12}/500000 \sim 8\ ms$$

is the Region 1 symbol time. This number doesn't account for communication between MCU and the radio nor switching states and frequencies.

On STM32CubeWL, measurement shows that CAD set to two valid symbols takes about 270 ms to scan 5 + 1 regions. There is a reasonable margin for robust operation since the preamble lasts at least 294 ms. The CAD can be set to one valid symbol to make the scanning even faster, but it may be less robust.

2.6.3 Sensor state diagram

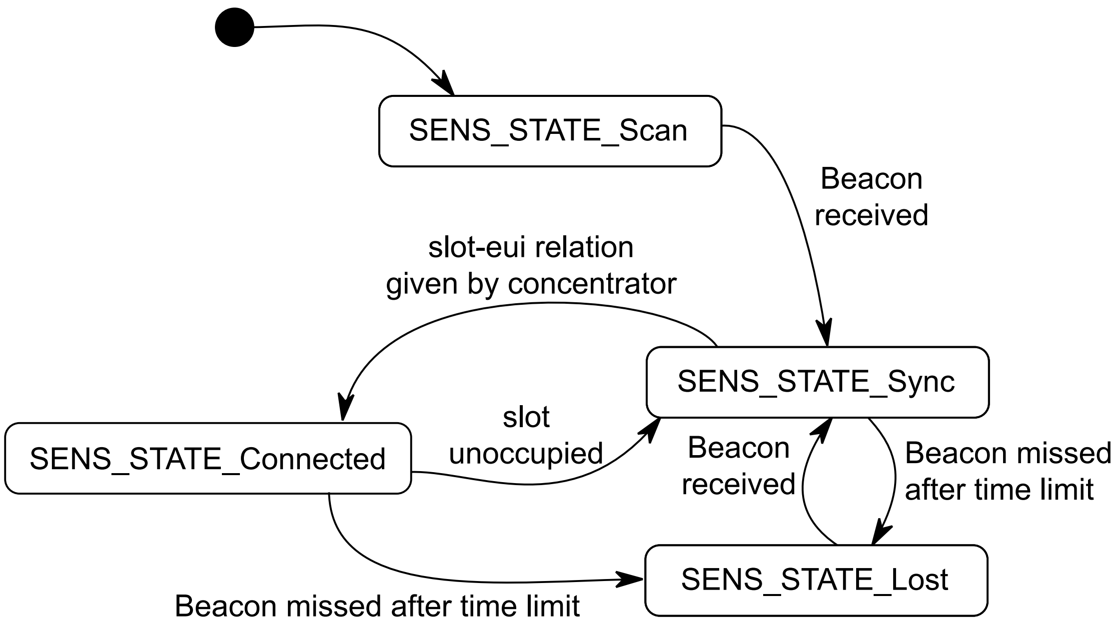
The sensor starts in the **SENS_STATE_Scan** state. In this state, the sensor flashes the red LED rapidly. Once it finds one beacon, it can never return to scan state except for reset.

In the **SENS_STATE_Sync** state, the sensor tracks beacons but is not connected. It can transmit in a random empty slot. In this state, any activity Rx or Tx makes the blue LED flash.

In the **SENS_STATE_Lost** state, the sensor is unable to track beacons. It periodically switches between long sleep periods and long receive, trying to find a beacon while saving battery. In this mode, the activity is reported by the red LED.

In the **SENS_STATE_Connected** state, the sensor is fully connected and transmits measured data in one constant slot. In this state, any activity flashes the green LED. If the concentrator fails to receive consecutive sensor packets, the concentrator sets the sensor slot to un-occupied and the sensor transits to **SENS_STATE_Sync** state.

Figure 2. **Sensor state diagram**



2.6.4 Connecting the concentrator

Once a sensor is synchronized to the concentrator, it randomly selects one unused slot from the mask of occupied slots and transmit sensor packet.

If the process is successful, the next concentrator sync appends a slot-EUI relation and mark the slot as occupied in the mask.

If the process is not successful, the next concentrator sync does not append the slot-EUI relation nor mark the bit in the mask and sensor must repeat the random selection.

If the sensor misses the next beacon or sync, it must wait for another beacon or sync. The occupied slot and the slot-EUI relation is kept in sync until a packet is received or until the sensor is considered lost.

The random selection can lead to collisions. The slot-EUI relation is added to prevent collisions that may persist forever. Demonstration implementation shows that the concentrator tends to receive one of the two colliding packets quite well. Using only the occupied mask resulted in a collision where both sensors saw the occupied bit being set, but only one sensor is periodically received by the concentrator. With the slot-EUI relation, the collision cannot persist and the concentrator has an additional option to shuffle sensors to any given slot.

2.6.5 Losing a sensor

The sensor is considered lost when it does not use its slot 5 consecutive times (the limit is multiplied by transmitting period+1). The bit in the occupied mask is cleared and the sensor needs to use the connection process with the default coding. For simplification, if the sensor does not hear either beacon or sync, it cannot transmit until the next beacon.

2.6.6 Measuring the meters

The `meters` module is responsible to measure temperature and battery voltage inside each sensor. They are measured right before transmission. The meters module is synchronized with the network module each time the data are read from the meters module. The meters are enabled 160 ms before the next transmission is planned. 10 ms before the transmission, the data are read from the meters and meters are disabled.

2.7 Changing sensor packet coding

For purposes of testing different coding and modulation, the modulation of sensor messages can be changed. The change is initialized by the concentrator adding one of the coding change structures into a sync packet. The coding is changed as soon as the change is published in the sync. The change is kept in the following sync packets until a valid packet is received from the sensor or until the sensor is lost.

2.7.1 Allowed coding combinations

There are various limits for the duty cycle, maximal bandwidth, and the maximal packet length. The packet length is maximally 0.9 s, even if the regulations may allow more. The remaining time of a slot is reserved for timing imprecision, listen before talk, and concentrator radio configuration.

The duty cycle can be limited to period settings. The period indicates how many beacons are skipped between two sensor transmissions. The concentrator automatically increases the period to fulfill the duty cycle.

The sensor packet length can be limited by data_lim. It tells how long the sensor payload can maximally be. The concentrator automatically lowers the limit to fit the packet in the time slot. If the packet length may not allow sending at least sensor address, packet counter, and version (payload of 6 bytes), then the modulation is forbidden.

For subregions without frequency hopping, the concentrator tries all defined channels when calculating the limits. The sensor packet is put on the first channel which fits the bandwidth and other parameters. The example is 250 kHz LoRa® bandwidth in Region 0, where a channel with a lower duty cycle must be used, which results in an increased period.

2.7.2 Equations to calculate LoRa® packet length

Formulas to calculate LoRa® packet length are detailed here:

$$T_S = \frac{2^{SF}}{BW}$$

$$T_{preamble} = \left(n_{preamble} + 4.25\right) \times T_S$$

$$n_{payload} = 8 + \max\left(ceil\left(\frac{8PL - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)}\right) \times \left(CR + 4\right), 0\right)$$

$$T_{payload} = n_{payload} \times T_S$$

$$T_{packet} = T_{preamble} + T_{payload}$$

Where:

- T_S is the symbol duration.
- *BW* is the modulation bandwidth.
- $N_{preamble}$ is the number of preamble symbols
- *PL* is the number of Payload bytes (1 to 255)
- *SF* is the spreading factor (6 to 12)
- *CRC=1* is 1 when CRC is used (CRC is always enabled)
- *IH=0* is 1 when the header is not present (header is always enabled)
- *DE* is 1 when LowDataRateOptimize is used
- **CR** is the coding rate (1 corresponding to 4/5, 4 to 4/8)

2.7.3 Equations to calculate FSK packet length

This simple FSK has one bit per symbol and symbol duration is the direct inverse of bitrate. There is a 4-byte preamble, a 4-byte sync word, a 1-byte length, payload, and 2 bytes of CRC.

$$T_{packet} = \frac{(4 + 4 + PL + 1 + 2) \times 8}{BR}$$

Where *PL* is payload in bytes and BR is bitrate.

2.7.4 Bandwidth equations

There are also limitations on the maximum bandwidth used. For FSK, approximation based on Carson's bandwidth rule is

$$BW = 2 \times (BT \times sr + f_{dev})$$

Where:

- sr is the symbol rate,
- BT is the Gaussian shaping (lower *BT* means bigger effect),
- f_{dev} is the frequency deviation.

For LoRa® the approximation is very rough

$$BW = BW_{set} \times 1.6$$

Where:

- BW_{set} is the bandwidth setting,
- BW is the actual bandwidth including sidelobes.

The sources for this are LoRaWAN® regional parameters where 200 kHz is used to fit 125 kHz LoRa®.

2.8 Regions specificities

2.8.1 Frequency hopping

Frequency hopping is used only in subregions where necessary due to regulations.

Beacon is not part of the hopping. Sync and sensors data slot use as many channels as available in the channel list.

Sync channel is controlled entirely by beacon seed. The seed is generated from a PRBS9 generator (x9+x5+1) valid for one superframe. The seed is the 8-bit LSBs PRBS9 output. The sync frequency channel is the beacon seed modulo the number of available frequencies (45 in this demo ranging from 902.5 MHz to 924.5 MHz with 500 kHz steps**)**

The slot channel relations for sensor packets are given entirely by a Mersenne MLCG generator initialized by the beacon seed every superframe.

$$x_{n+1} = x_n \times 6208991 \bmod \left(2^{31} - 1\right)$$

Note: The generator cannot handle 0. If the seed is 0, it is replaced by `0x32b87`*.*

2.8.2 Listen before talk (LBT)

Listen before talk is used only in subregions where necessary due to regulations.

All communication starts with receiving. The device reads the received signal strength indicator (RSSI) and measures the time when the indicator is below the required value. If the channel is busy the measured time is restarted. Only when the channel is free for the necessary time interval, the transmission can begin.

The packet can be delayed for only as long as there is space in the slot. This means that the packet must end before 940 ms after slot start. The remainder of the slot is a delay required by regulations and time to reset the radio.

3 Channel maps and hopping

Regions, subregions, and channels are internally held in a tree of linked structures. Changing system behavior means only modifying few constants in the Flash memory.

3.1 Region 0, Europen Union

Beacon frequency is 869.525 MHz. Sensor frequency is the same for modulations narrower than 250 kHz. This channel has a 10% duty cycle. Maximum power of 22 dBm EIRP can be used.

Modulations with bandwidth between 250 kHz and 600 kHz (including LoRa® 250 kHz) are switched to an alternative frequency 868.3 MHz. This channel has a 1% duty cycle and 16.15 dBm EIRP power.

3.2 Region 1 frequency hopping

Beacon is sent at 925 MHz.

3.2.1 Subregion 1.0, United States of America

Maximum power of 21 dBm EIRP is compliant for all modulations.

The frequency hopping uses 44 channels for sync and sensors. Channels are separated by 500 kHz, starting on 924.5 MHz (channel 0) and going down to 902.5 MHz (channel 44).

Bandwidth is limited to 500 kHz.

3.2.2 Subregion 1.1, Australia

Maximum power of 21 dBm EIRP is compliant for all modulations.

The frequency hopping uses 19 channels for sync and sensors. Channels are separated by 500 kHz, starting on 924.5 MHz (channel 0) and going down to 915.5 MHz (channel 44).

Bandwidth is limited to 500 kHz.

3.3 Region 2, Asia

Beacon is transmitted at 923.4 MHz.

3.3.1 Subregion 2.0, Korea

Power is 12.15 dBm EIRP.

There is an LBT of 5 ms with a level of -80 dBm.

All communications use the beacon frequency.

The bandwidth is limited to 200 kHz.

3.3.2 Subregion 2.1, Japan

Power is 15.15 dBm EIRP.

There is an LBT of 5 ms with a level of -80 dBm.

Channel 0 is equal to the beacon frequency and is limited to 200 kHz. Alternative channels are shifted to fit inside given bounds.

Table 6. **Japan optimum settings**

Channel	Frequency	Bandwidth
0	923.4 MHz	200 kHz
1	923.3 MHz	400 kHz
2	923.2 MHz	600 kHz
3	923.1 MHz	800 kHz
4	923.0 MHz	1000 kHz

3.3.3 Subregion 1.2, Generic Asia

Power is limited to 16 dBm EIRP.

All communications use the beacon frequency.

The bandwidth is limited to 200 kHz.

3.4 Region 3, China

Power is 19.15 dBm EIRP.

All communications use 470.3 MHz.

Bandwidth is limited to 200 kHz.

3.5 Region 4, India

Maximum power of 22 dBm EIRP is compliant for all modulations.

All communications use 865.1 MHz.

Bandwidth is limited to 200 kHz.

3.6 Region 5, Russia

Power is 16.25 dBm EIRP.

All communications use 868.95 MHz.

Bandwidth is limited to 500 kHz.

4 Example of concentrator operation

The connection between PC software and concentrator is Virtual COM port provided by ST-LINK. The settings are 9600 bauds, 8-bit, no parity, and 1 stop bit. The concentrator listens for specific AT commands and responds using AT prefix or OK. Anything without AT prefix (except OK) is debugging information and can be ignored.

4.1 STM32CubeMonitor

The STM32CubeMonitor provides a user GUI for easy region selection and sensor data presentation.

The flow (WLR_demo.json) must be loaded in STM32CubeMonitor located in the STM32CubeWL Firmware Package under `\Projects\NUCLEO-WL55JC\Demonstrations\LocalNetwork\LocalNetwork_Concentrator\STM32CubeMonitor`.

STM32CubeMonitor connects to the board running the `Demo_Concentrator` binary, configures the geographical area and displays the list of sensors detected.

More information to install and use the dashboard are available in wiki at `https://wiki.st.com/stm32mcu/wiki/STM32CubeMonitor:How_to_start_with_STM32_Wireless_Long_Range_demo`

4.2 Example of concentrator operation with terminal

4.2.1 Initialization

After the version check, the first three commands must set region, subregion, and start the beacon. For European Union it is **AT+REGION=0**, **AT+SUBREGION=0**, and **AT+BEACON_ON**. The first two commands select the format of the transmitted beacon. The third command starts sending the beacon. For a list of available regions, refer to Section 3 or run **AT+LIST_REGIONS**.

Note: The first beacon is sent after a random time at most 16 seconds after the command.

4.2.2 Sensor data received

For each received sensor packet, there is **AT+RCV** printed information. The format is explained in Section 4.3.1 .

4.2.3 Changing the coding to LoRa®

The first step is to put the coding parameters in the corresponding registers. To get a LoRa® 250 kHz with spreading factor SF9 and code rate 4/5 it is **AT+DE=0**, **AT+CR=1**, **AT+SF=9**, and **AT+BW=8**.

The second step is to apply the new modulation. To change sensor 0x12345678, the command is **AT+MOD_LORA=0x12345678**. The register values are persistent and can be used to set multiple sensors to the same modulation.

The modulation is acknowledged by **OK**, but the actual change happens at best in the following period. When the modulation is changed, the concentrator sends a notification **AT+MOD_OK=0x12345678**.

4.3 Concentrator AT interface

The full list of AT interfaces is described below.

4.3.1 Commands

- Get means AT+COMMAND=?<cr><lf>
- Set means AT+COMMAND=param<cr><lf>
- Run means AT+COMMAND<cr><lf>

All parameters are decimal numbers unless stated otherwise.

Table 7. Commands

<table>
<tr><th>AT command</th><th>Actions</th><th colspan="2">Comment</th></tr>
<tr><td>Z</td><td>Run</td><td colspan="2">Reset</td></tr>
<tr><td>+VER=?</td><td>Get</td><td colspan="2">Get the firmware version in vX:Y format, where X is the major version and Y the minor one.
The major version is incompatible with others, regions and subregions may be changed.
The minor version may change for instance the sensor data format.</td></tr>
<tr><td>+VL=level</td><td>Set</td><td colspan="2">Set verbose level
• level 0, only AT responses
• level 1, additional error information
• level 2, additional occupation state every period</td></tr>
<tr><td>+LIST_REGIONS</td><td>Run</td><td colspan="2">Print list of all available regions and subregions</td></tr>
<tr><td>+DE=de</td><td>Get, Set</td><td colspan="2">Register for changing LowDataRateOptimize [0 .. 1], recommended when symbol length is greater than 16 ms</td></tr>
<tr><td>+CR=cr</td><td>Get, Set</td><td colspan="2">Register for changing the LoRa® code rate [1 .. 4] = [4/5 .. 4/8]</td></tr>
<tr><td>+SF=sf</td><td>Get, Set</td><td colspan="2">Register for changing the LoRa® spreading factor [6 .. 12]</td></tr>
<tr><td>+BW=bw</td><td>Get, Set</td><td>Register for changing the LoRa® bandwidth [0 .. 4]
• 0 - 7.81 kHz
• 1 - 10.42 kHz
• 2 - 15.63 kHz
• 3 - 20.83 kHz
• 4 - 31.25 kHz</td><td>Register for changing the LoRa® bandwidth [5 .. 9]
• 5 - 41.67 kHz
• 6 - 62.5 kHz
• 7 - 125 kHz
• 8 - 250 kHz
• 9 - 500 kHz</td></tr>
<tr><td>+RISE=rise</td><td>Get, Set</td><td>Register for changing FSK power ramp [0 .. 3]
• 0 - 10 µs
• 1 - 20 µs
• 2 - 40 µs
• 3 - 80 µs</td><td>Register for changing FSK power ramp [4 .. 7]
• 4 - 200 µs
• 5 - 800 µs
• 6 - 1700 µs
• 7 - 3400 µs</td></tr>
<tr><td>+BR=br</td><td>Get, Set</td><td colspan="2">Register for changing FSK bitrate [600 .. 300000 bits/s]</td></tr>
<tr><td>+FDEV=fdev</td><td>Get, Set</td><td colspan="2">Register for changing FSK frequency deviation [Hz]</td></tr>
<tr><td>+BT=bt</td><td>Get, Set</td><td colspan="2">Register for changing FSK Gaussian symbol shaping BT [0 .. 4]
• 0 - off, no shaping
• 1 - 0.3, the biggest effect
• 2 - 0.5
• 3 - 0.7
• 4 - 1, the smallest effect</td></tr>
<tr><td>+MOD_LORA=EUI</td><td>Set</td><td colspan="2" rowspan="2">Set modulation for the sensor to LoRa®/FSK. Parameter EUI is hexadecimal `0xabcd1234`.
Modulation parameters are taken from DE, CR, SF, and BW for LoRa® and RISE, BR, FDEV, and BT for FSK. Some combinations are not allowed.
May return AT+MOD_LIM=period,data_lim if the modulation limits the period or length of received data.</td></tr>
<tr><td>+MOD_FSK=EUI</td><td>Set</td></tr>
<tr><td>+MOD_TEST_LORA</td><td>Run</td><td colspan="2" rowspan="2">Same as AT+MOD_XXX=EUI, but test only.
Returns AT+MOD_LIM=period,data_lim every time.</td></tr>
<tr><td>+MOD_TEST_FSK</td><td>Run</td></tr>
<tr><td>+REGION=r</td><td>Get, Set</td><td colspan="2" rowspan="2">Set region and subregion.
Is effective only after next AT+BEACON_ON.</td></tr>
<tr><td>+SUBREGION=s</td><td>Get, Set</td></tr>
<tr><td>+BEACON_ON</td><td>Get, Set, Run</td><td colspan="2">Turn Beacon on and off (Run equals to Set=1)</td></tr>
</table>

4.3.2 Responses

Table 8. Responses

Response	Description
OK	Command processed successfully
AT_ERROR	Error while processing command
AT_PARAM_ERROR	Wrong parameters for the command
AT_TEST_PARAM_OVERFLOW	Parameters are too long.
AT_RX_ERROR	UART receives an error.
AT_MOD_NOT_ALLOWED	Selected modulation is not allowed.
AT_EUI_NOT_CONNECTED	The requested sensor is not available.

4.3.3 Spontaneous messages

Table 9. Spontaneous messages

Message	Comment
AT+RCV=EUI, counter, version_major:version_minor, RSSI, SNR, data, data...	Received sensor packet, data are described in Section 4.4 . The counter is incremented in each transmitted packet, RSSI is the signal strength [dBm], SNR is signal to noise ratio [dB].
AT+MOD_LIM=period,data_lim	The modulation is accepted (OK is returned as usual), but period or data are limited. • period, how many periods are skipped, 0 - transmit once in 16 s, 1 - transmit once in 32 s • data_lim, how many bytes (including 6 bytes of EUI, counter, and version) is sent, 26 bytes is maximum
AT+MOD_OK=EUI	Change of modulation is successful. This can appear any time if the sensor loses synchronization, connects again and the modulation is changed again.
AT+LOST=EUI	When a sensor is lost

4.4 Data and packet formats

Below is a description of data measured, sent, and printed in AT+RCV. A letter **E** can be printed instead of value if the data are not available due to payload data limit or error in measurement.

Table 10. Data measurement results

Data	Description	On-air format	Format in AT+RCV	Unit in AT+RCV
Temperature	MCU temperature measurement	`int16_t [0.01 `C]`	+22.41	°C
Voltage	Voltage on MCU Added in v0:6, printed at the end	`uint8_t [0.05 V]` (sent before distance, where activity is in v0:5)	3.15	V

Examples:

AT+RCV=EUI,counter,version_major:version_minor,rssi,snr,data…

AT+RCV=0x12126741,0x69,1:0,-98,6,+27.04,3.3

AT+RCV=0x12126741,0x6a,1:0,-98,6,+27.08,3.3

AT+RCV=0x12126741,0x6b,1:0,-95,7,+27.09,3.3

AT+RCV=0x12126741,0x6c,1:0,-98,6,+27.11,3.3

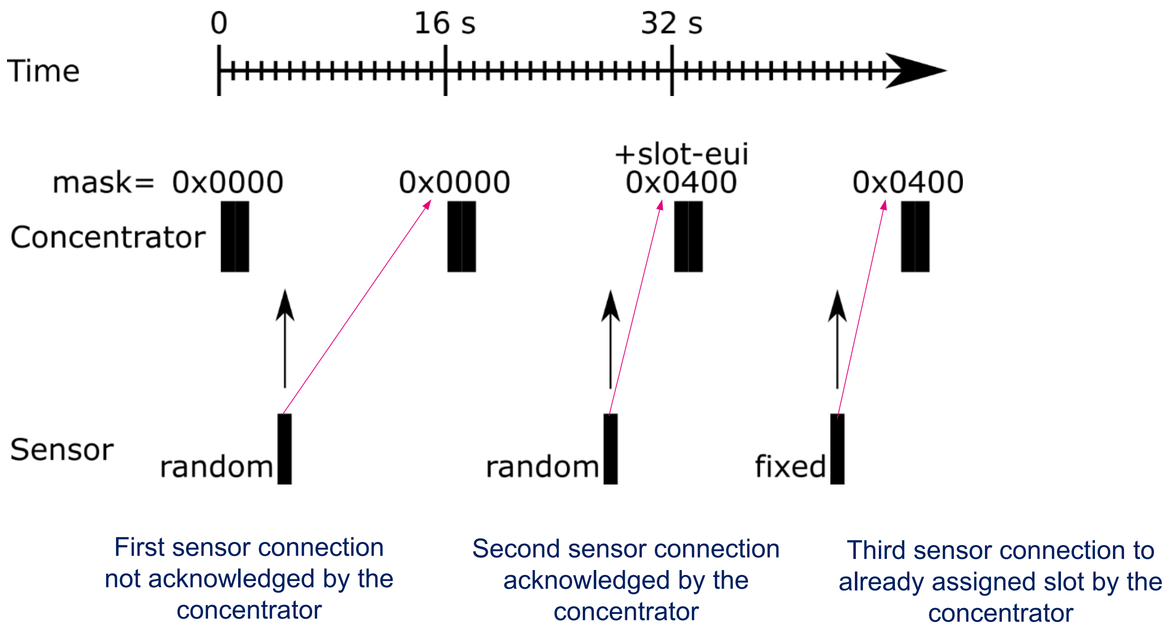
5 Radio behavior examples

5.1 Connecting to the concentrator

The sensor chooses a random available slot and transmits. Concentrator acknowledges the connection by appending slot-EUI relation and setting the correct bit in the mask of occupied channels.

If the sensor is already known, the slot-EUI can switch the sensor to its previous slot. This can happen when the sensor is power reset and not yet considered lost.

Figure 3. Connection process



5.2 Frequency hopping

Each channel must be used equally. The hopping is controlled by a seed number generated by the concentrator and published in a beacon. From this seed, channels are randomly generated.

Table 11 shows the first ten superframe frequencies per slot.

Table 11. US frequency hopping per slot and superframe in MHz

Superframe	Hexa. seed	Sync	Slot1	Slot2	Slot3	Slot4	Slot5	Slot6	Slot7	Slot8	Slot9	Slot10	Slot11	Slot12	Slot13	Slot14
superframe1	AB	906.5	913	913	908	907.5	920	917	905.5	904	919	903	919	923	923	904.5
superframe2	56	904	921	921.5	911	915.5	916	912.5	923.5	907	910	907.5	917	907.5	911	907
superframe3	AC	906	917.5	914.5	916	906.5	907.5	923	922.5	912	918	909	909.5	909	916	912
superframe4	58	903	907.5	920.5	908.5	913.5	913.5	906	916.5	923	908	919.5	920.5	920.5	911.5	903.5
superframe5	B0	904	909	912.5	911	902.5	902.5	910	904.5	921.5	914	910.5	912.5	916.5	921	923.5
superframe6	61	921	903	918	910.5	908.5	909.5	912.5	924	905	921.5	910	917	907	914.5	911.5
superframe7	C3	917	904.5	913	923	914	904.5	910.5	918	916	917.5	924	922.5	916.5	912	924.5
superframe8	86	902.5	908	917.5	918.5	910	903.5	903.5	903.5	904.5	907	904.5	920	913	907.5	906
superframe9	0D	918	915.5	924	921.5	919.5	915.5	922	918	916	915.5	920.5	905.5	918.5	910	921.5
superframe10	1B	911	911	902.5	904	909.5	916.5	903	924.5	915.5	924	922.5	918	921	907	907.5

5.3 EU duty cycle control

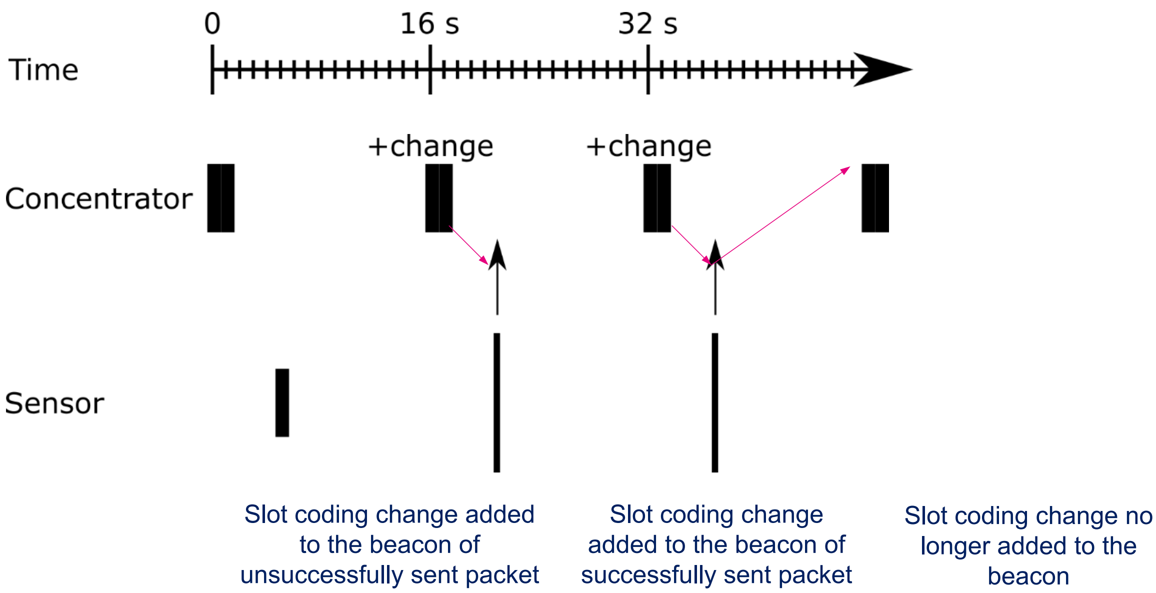
Sensors in the EU may use the h1.4 frequency band, which requires a 1% duty cycle. If the packet is longer than 0.16 seconds, the channel coding needs to set a communication period. Period 0 means that transmission happens after each beacon. Period 1 skips even beacons for a packet between 0.16 s and 0.32 s long. The maximal packet length is 0.9 s.

Figure 4. EU duty cycle limiting for period = 1

5.4 Change the slot coding

The slot coding change is added to the beacon until a first successful packet from the sensor is received.

Figure 5. Changing the slot coding



Revision history

Table 12. **Document revision history**

Date	Revision	Changes
7-Dec-2020	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved