AT Commands for HT32SX

1 ABOUT THIS APPLICATION

The AT Commands for HT32SX is an **EXAMPLE** application developed for a master-slave protocol, where an external module or microcontroller is used to control the entire system.

This is an open-source code. Users are responsible to change and adapt it to their specific context.

Table 1. Available Commands.

Command	**Arguments**	**Description**
AT+SEND	**DOWNLINK_FLAG:** Downlink flag is set to 1 to wait for a downlink. **PAYLOAD:** Payload that will be sent to the SigFox Network. It must be less or equal than 12 bytes.	Send a payload to the SigFox Network. Before calling this function, it is necessary to call the ***AT+CFGRCZ*** first. **Example 1:** *AT+SEND=1:AAAAAAAA;* (Wait for a downlink). **Example 2:** *AT+SEND=0:AAAAAAAA;* (Do not wait for a downlink). It is recommended to close Sigfox Library after using this command (AT+CLOSE;).
AT+CFGRCZ	**RCZ:** RC value corresponding to the SigFox region where the device is going to operate in.	Open SigFox library according to the region. Returns 0 if ok. **Example 1:** *AT+CFGRCZ=2;*
AT+MONARCH	**RCZ:** RC beacon expected (in order to scan every region available, the RCZ value should be 127). **Minutes:** Timeout. (It is recommended to use at least 6 minutes).	Scan a Monarch Beacon and returns by a serial terminal, the region founded. The library must be closed before use this command (command AT+CLOSE). **Example 1:** *AT+MONARCH=2:6; (*scan only RC2 beacons). **Example 2:** *AT+MONARCH=127:6;*
AT+STPMONARCH	None	Stops an already running Monarch Scan. Returns 0 if ok. **Example:** *AT+STPMONARCH;*
AT+CLOSE	None	This command closes the SigFox library (Free the allocated memory of SIGFOX_API_open and close RF). Returns 0 if success. **Example:** AT+CLOSE;

AT+RESET	None	Resets the MCU. **Example:** *AT+RESET;*
AT+STOP	None	Makes the MCU enters in stop mode. **Example:** *AT+STOP;*
AT+DEEPSLEEP	None	Makes the MCU enters in deep sleep mode (calls stop mode and turn off most of peripherals, keeping on only the USART). **Example:** *AT+DEEPSLEEP;*
AT+WKP	None	Wake up the MCU after a stop or deep sleep mode. **Example:** *AT+WKP;*

AT+SWITCHPA	**PA_Flag:** 1 to enable and 0 to disable.	Turn on or turn off PA.
		Example 1 (turning on): *AT+SWITCHPA=1;*
		Example 2 (turning off): *AT+SWITCHPA=0;*

AT+SWITCHBOOST	**Boost_Flag:** 1 to enable boost mode and 0 to disable.	Turn on or turn off S2LP boots mode.
		Example 1 (turning on): *AT+SWITCHBOOST=1;*
		Example 2 (turning off): *AT+SWITCHBOOST=0;*

AT+REDUCEPOWER	**Power_value:** Reduce factor	Reduces the output power of the transmitted signal. (reduction*0.5dB against the actual value). **Example:** *AT+REDUCEPOWER=-27;*
AT+FREQOFFSET	**Offset:** Frequency offset in Hz as an integer.	Sets the RF frequency offset in Hertz. **Example:** *AT+FREQOFFSET=1200;*
AT+RSSIOFFSET	**Offset:** RSSI offset as an integer in dB.	Sets a RSSI offset. **Example:** *AT+RSSIOFFSET=-17;*
AT+LBTOFFSET	**Offset:** Integer representing LBT offset in dB.	Set an offset (dB) for turning the LBT mechanism. **Example:** *AT+LBTOFFSET=0;*

- **Payload data type**:
  The default payload data consists of hexadecimal values.

  Payload example with hexadecimal values:

  AT+SEND=0:0123456789ABCDEF01234567;

  If there is a need to send ASCII data, the user should set a new define symbol before compile the code again:

1. Click “*Project -> Properties*”:

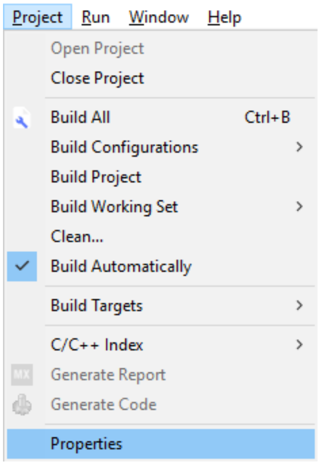
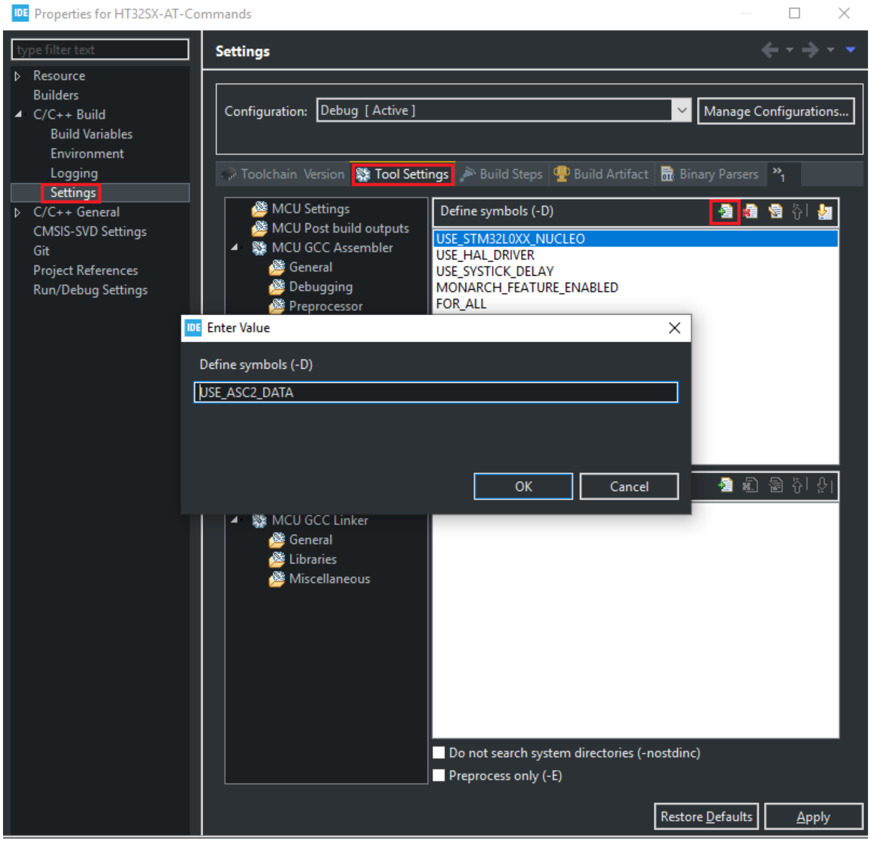


Image 1 – *Project* tab.

2. Click “*C/C++ Build -> Settings*”, select “*Tool Settings*” tab and click on the icon to add a new define symbol:



Configuration: Debug [Active]

Define symbols (-D)

USE_STM32L0XX_NUCLEO
USE_HAL_DRIVER
USE_SYSTICK_DELAY
MONARCH_FEATURE_ENABLED
FOR_ALL

Define symbols (-D)

USE_ASC2_DATA

□ Do not search system directories (-nostdinc)

□ Preprocess only (-E)

Image 2 – Project properties.

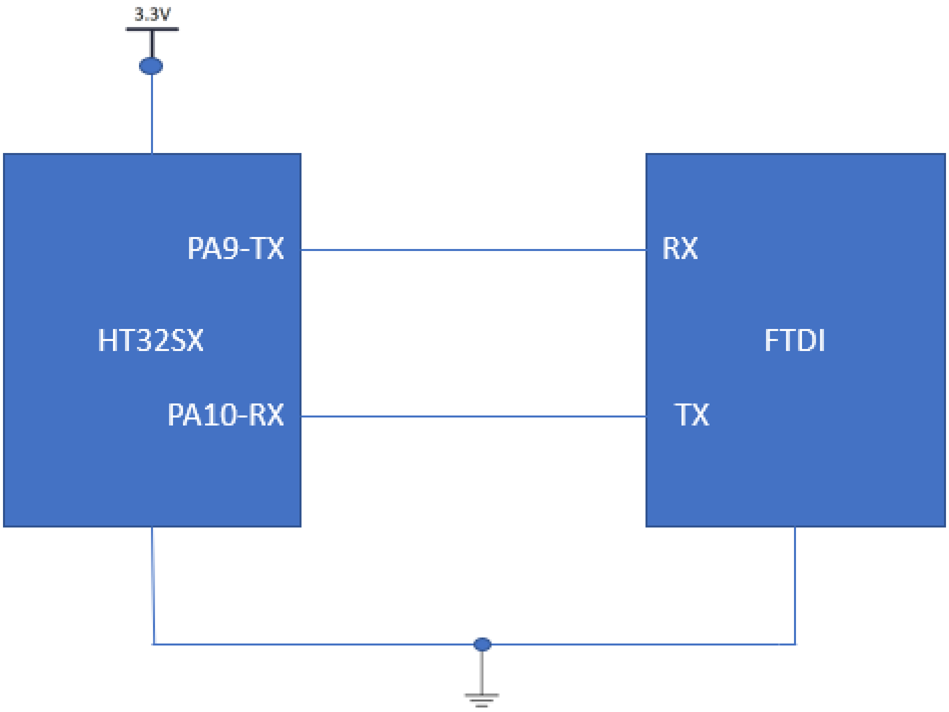
3. Write *USE_ASC2_DATA* and click on *Apply* to save all changes.

2 DEVELOPMENT SETUP

- GIT (for Windows, *git-scm.com* is recommended).

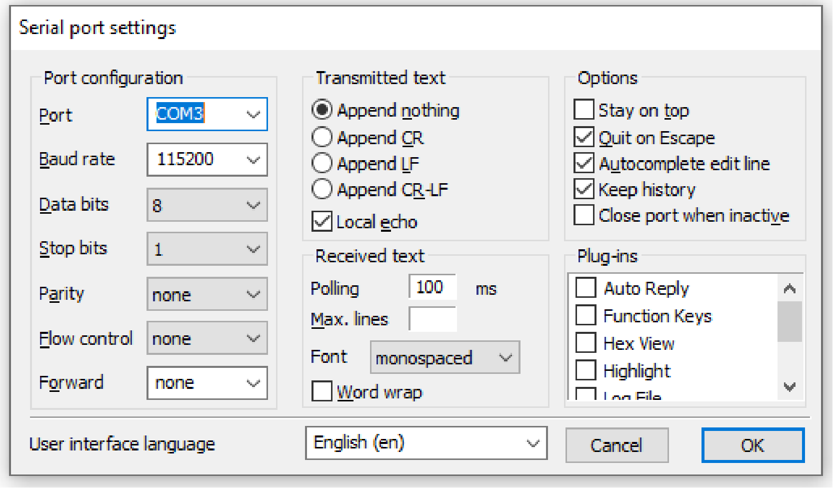
- ST-LINK Utility or STM32CubeProgrammer.
- STM32CubeIDE.
- RS232 terminal (Termite is recommended).
- HT32SX chip.
- An antenna for your respective Sigfox RC Zone or a multiband antenna.
- A FTDI (USB-Serial converter).

Block Diagram:



3 EXECUTING WITH A SERIAL TERMINAL

Termite is recommended.



Serial port settings

Port configuration
Port COM3
Baud rate 115200
Data bits 8
Stop bits 1
Parity none
Flow control none
Forward none

Transmitted text
● Append nothing
○ Append CR
○ Append LF
○ Append CR-LF
☑ Local echo

Received text
Polling 100 ms
Max. lines
Font monospaced
□ Word wrap

Options
□ Stay on top
☑ Quit on Escape
☑ Autocomplete edit line
☑ Keep history
□ Close port when inactive

Plug-ins
□ Auto Reply
□ Function Keys
□ Hex View
□ Highlight
□ Log File

User interface language English (en)

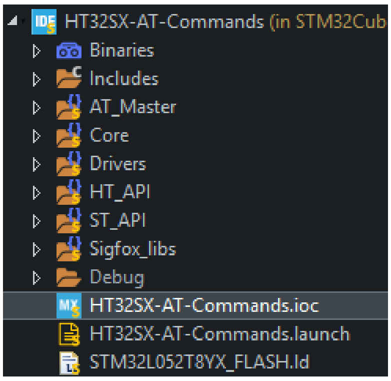
Image 3 – Termite configuration.

4 SLEEP AND WAKE-UP MECHANISM

The AT Commands for HT32SX was made to wake-up from deep sleep mode using only the USART1 RX. It means that after an *AT+DEEPSLEEP;* command, a simple *AT+WKP;* should be enough to awake it again. Even though it seems convenient, keeping USART awake during the whole deep sleep process will increase the current consumption to around 90 µA. In cases where the power consumption is critical, it is recommended to change the deep sleep routine to use the Standby Mode instead of Stop Mode, awaking your device through Wake-up Pin (PA0). These changes will decrease the current consumption to about 3 µA.

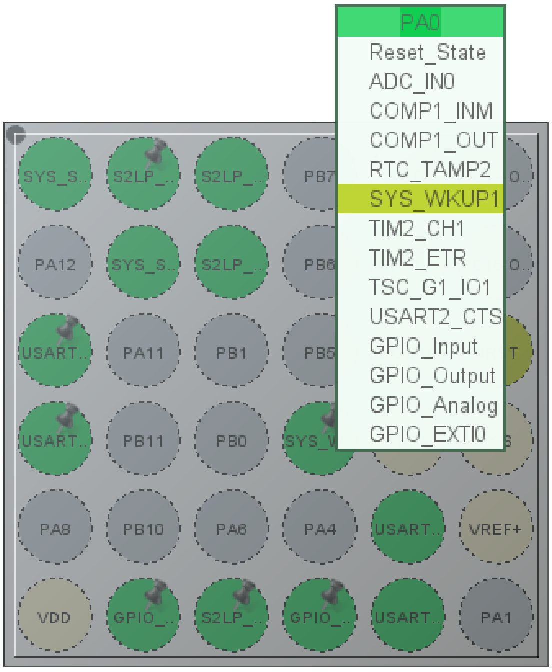
Changing the AT Commands to wake-up using WKP_PIN:

1. Open the AT Commands project in STM2CubeIDE and double click the .ioc file to open the GUI tool.

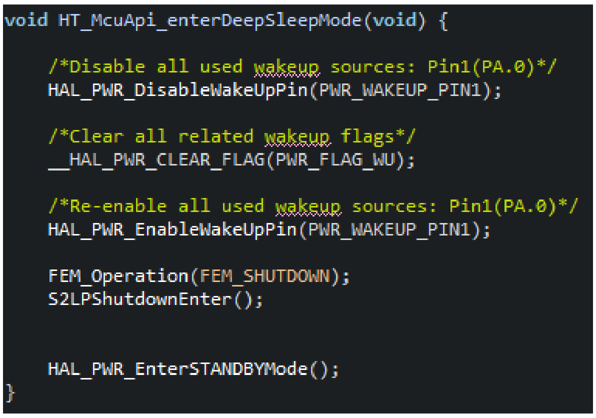


```
HT32SX-AT-Commands (in STM32Cub
  Binaries
  Includes
  AT_Master
  Core
  Drivers
  HT_API
  ST_API
  Sigfox_libs
  Debug
  HT32SX-AT-Commands.ioc
  HT32SX-AT-Commands.launch
  STM32L052T8YX_FLASH.ld
```

2. Find the PA0 pin, click on it and select SYS_WKUP1 option:



3. Go to *Project -> Generate Code* to salve all changes and generate the related code.
4. Then go to *HT_API -> Src -> HT_mcu_api.c* and change the *HT_McuApi_enterDeepSleepMode* function as shown in the example below:

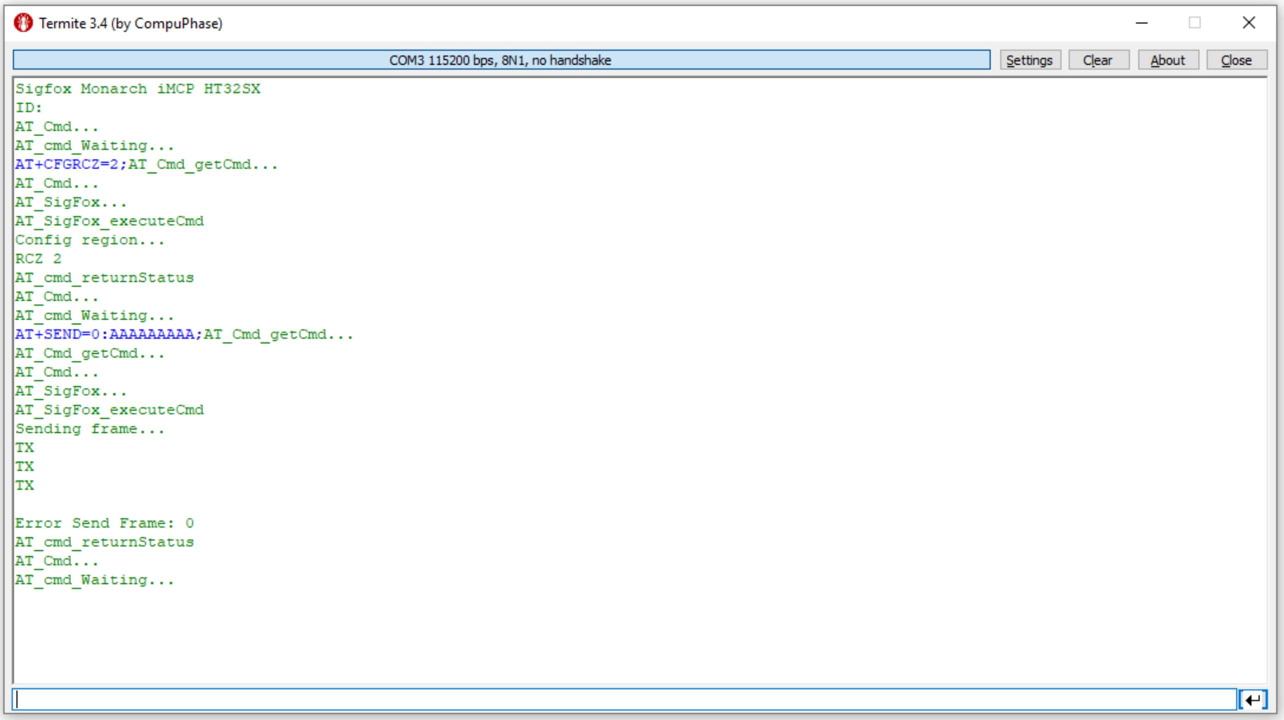


```
void HT_McuApi_enterDeepSleepMode(void) {

    /*Disable all used wakeup sources: Pin1(PA.0)*/
    HAL_PWR_DisableWakeUpPin(PWR_WAKEUP_PIN1);

    /*Clear all related wakeup flags*/
    __HAL_PWR_CLEAR_FLAG(PWR_FLAG_WU);

    /*Re-enable all used wakeup sources: Pin1(PA.0)*/
    HAL_PWR_EnableWakeUpPin(PWR_WAKEUP_PIN1);

    FEM_Operation(FEM_SHUTDOWN);
    S2LPShutdownEnter();


    HAL_PWR_EnterSTANDBYMode();
}
```

5 RESULTS

Some results are expected running this application. It is recommended to use a serial terminal to check if the commands are being executed (Termite was used in these examples).

1. Sending a frame to the Sigfox Network:

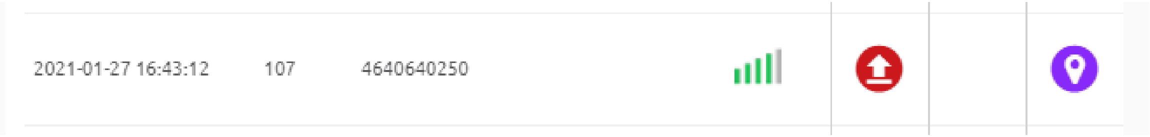


COM3 115200 bps, 8N1, no handshake

```
Sigfox Monarch iMCP HT32SX
ID:
AT_Cmd...
AT_cmd_Waiting...
AT+CFGRCZ=2;AT_Cmd_getCmd...
AT_Cmd...
AT_SigFox...
AT_SigFox_executeCmd
Config region...
RCZ 2
AT_cmd_returnStatus
AT_Cmd...
AT_cmd_Waiting...
AT+SEND=0:AAAAAAAAA;AT_Cmd_getCmd...
AT_Cmd_getCmd...
AT_Cmd...
AT_SigFox...
AT_SigFox_executeCmd
Sending frame...
TX
TX
TX

Error Send Frame: 0
AT_cmd_returnStatus
AT_Cmd...
AT_cmd_Waiting...
```

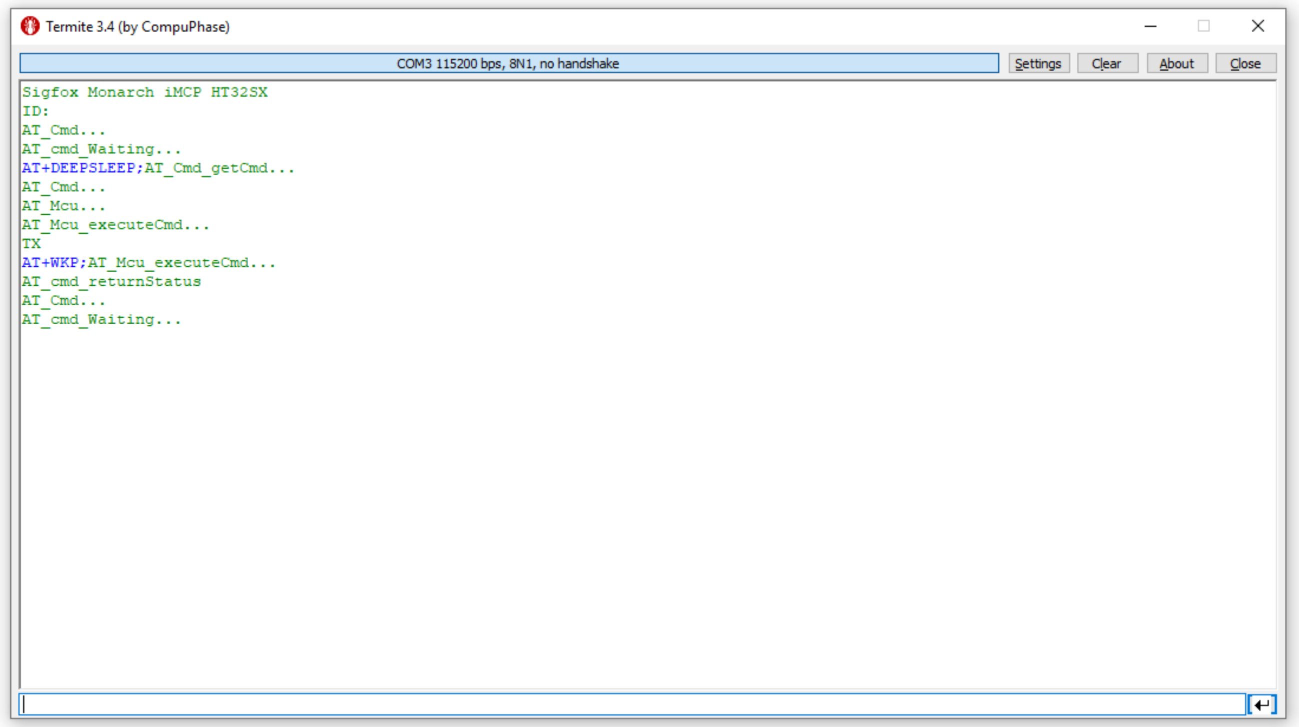
Image 7: RS232 terminal with the expected results after the send frame command.

Check if your message was received in our backend. In case you still do not have an account on the Sigfox Backend, please check our FAQ (Q1).



2021-01-27 16:43:12 107 4640640250

2. Sleep mode:


Termite 3.4 (by CompuPhase)
COM3 115200 bps, 8N1, no handshake
Settings
Clear
About
Close

```
Sigfox Monarch iMCP HT32SX
ID:
AT_Cmd...
AT_cmd_Waiting...
AT+DEEPSLEEP;AT_Cmd_getCmd...
AT_Cmd...
AT_Mcu...
AT_Mcu_executeCmd...
TX
AT+WKP;AT_Mcu_executeCmd...
AT_cmd_returnStatus
AT_Cmd...
AT_cmd_Waiting...
```

Image 8: RS232 terminal with the expected results after the deep sleep command.

6 EXTRA DOCUMENTATION

Datasheets and other firmware examples can be found at the HT32SX Repository.

Basic Firmware Examples.

Please access our FAQ to learn extra useful information about the Sigfox Network, such as registration of a new device and a new user on Sigfox Backed.

7 REFERENCES

For additional information about SigFox libraries designed by ST Microelectronics, please refer to the UM2173 document (note: the function names still the same, but the code was adapted to use in the iMCP design, so it is different from the one distributed by ST).

8 CONTACT INFORMATION

Head Office – São Leopoldo, RS
HT Micron Semiconductors
Unisinos Avenue, 1550
São Leopoldo - RS
ZIP 93022-750
Brazil
Tel: +55 51 3081-8650
E-mail (Support): support_iot@htmicron.com.br
E-mail (General Enquiries): htmicron@htmicron.com.br

Version Control:

1.0 – Aimed for the engineering samples of iMCP