Monarch Move-ME with HT32SX

1 INTRODUCTION

The HT32SX is a System-in-Package device build for the Internet of Things providing a **ready-to-use** connectivity solution. The system provides an ARM Cortex M0+ 32bit, the STM S2-LP low power transceiver and an RF Front-End module combining all the advantages, integration, energy efficiency and convenience of advanced semiconductor packaging technology into a single chip with a 50Ω RF TX/RX interface. As a SigFox™ Monarch enabled device, it can operate in all regions covered by SigFox™ Network without need of reconfiguration. This is possible because the device can detect the region of operation and rearrange its setup automatically.

1.1 Application description

This application explain how the SigFox™ network can be accessed using the HT32SX . It searches the available SigFox™ network region via Monarch feature and publishes the found network region after a motion detection. The motion detection is developed with the help of the ST Microelectronics X-NUCLEO-IKS01A2 board (https://www.st.com/en/ecosystems/x-nucleo-iks01a2.html) using an accelerometer. The device operates in two modes:

- Monarch mode: the device scans for available SigFox™ networks through the Monarch feature and stores the network configuration it finds for later use.
- Move-me mode: the device publishes found SigFox™ network configuration after detecting a valid motion.

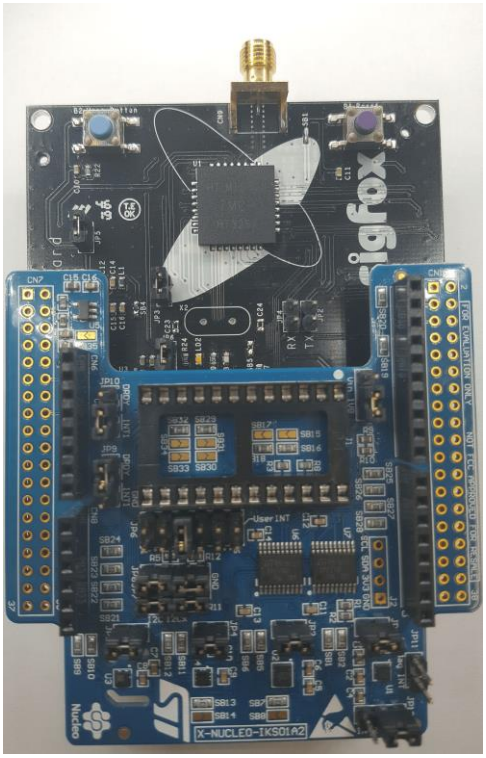
The user may press USER BUTTON (blue button) to change the device mode (Monarch mode or POC mode) as many times as desired. Press RESET BUTTON (purple button) to revert the device to initial condition. To detect the SigFox™ network configuration via Monarch feature, press the RESET button (purple button) on the evaluation board and press the USER BOTTON (blue button) once to switch to Monarch mode. After detecting a valid SigFox™ network Monarch Beacon (https://build.sigfox.com/monarch), the device will show the SigFox™ network region found as a message on the terminal.

Press the USER button (blue button) again to switch to POC mode. In this mode, the device posts a message to the SigFox™ network after a motion detection, which is done by an accelerometer built into the shield mounted on top of the device. The device will publish a message on the SigFox™ network containing the found region after a valid motion detection. The USER LED (white LED) indicates the current state of the device (OFF: IDLE or ON: BUSY) and will remain ON while the device sends the SigFox™ message indicating that the device is in the BUSY state. The user can send a new SigFox™ message after the USER LED (white LED) turn OFF, indicating the IDLE state. While the device s in BUSY state (USER LED ON), the user may not change the operating mode.

2 HARDWARE SETUP

The pin connections of the HT32SX_DEV_KiT can be found in the "*Getting Started with HT32SX"* document at https://github.com/htmicron/ht32sx. The required hardware to perform the steps described in this document consists of:

- HT Micron iMPC Evaluation Board
- X-NUCLEO-IKS01A2
- 868MHz – 928MHz standard Antenna
- Micro-USB cable for power supply



3 SOFTWARE SETUP

This section describes all necessary steps needed to use the SigFox™ with the **HT32SX**.

Also, these programs are recommended:

- GIT (for Windows, *git-scm.com* is recommended)
- HT Credential Generator (github.com/htmicron/imcp-credential)
- STM32 ST-LINK (www.st.com/en/development-tools/stsw-link004.html)
- ARM KEIL™ for STM32 (www2.keil.com/stmicroelectronics-stm32)
- Optionally, STM32CubeMX for personalisation of the program using a graphical interface (*www.st.com/en/development-tools/stm32cubemx.html*)

4 APPLICATION CODE

4.1 Code repository

The code is available at the GitHub™ (https://github.com/htmicron/ht32sx)

Additional to the code, we also distribute the compiled program in the *hex* format at the same repository in the GitHub.

4.2 Functions description

The general overview of the program is explained in this chapter. Some key functions used are explored in the code chapter.

This program was developed using STM32CubeMX to initialise the required peripherals and features of the MCU (microcontroller unit) prior to the coding. The Cube file is included. To develop the code ARM Keil™ used. Libraries to Sigfox™ are also included.

The power mode STOP is explored: this mode saves a lot of power, but it keeps the RAM powered on, and some peripheral required for the properly functioning of the Radio and the Power Amplifier.

This code has a software encoded FSM (Finite State Machine) that encompass the full execution of the example. There are two different operations encoded in the several states: the MCU go to the STOP mode; after and interruption event (an external one: a button pressed), the CPU resume execution – reinitialising the peripherals –, and send a message in the SigFox™ network. After, the CPU goes back to the STOP mode.

Some customisations might be done by setting the proper #DEFINE on the main.c file. They are:

***#define ENABLE_MONARCH*:** to enable the Monarch scanning feature. It detects in which region the device is, and the message is sent using the corresponding parameters.

#define ENABLE_ONLY_RCn where *n* is the region number, from 1 up to 6. In the case where the Monarch feature is not enabled, one region must be set using this define *format*.

The key functions that allows for the functionality are:

***void ht_mcu_config(void)*:** this function initialises the MCU customisations as well as the external-to-MCU peripherals included on the SiP (namely, the radio transceiver and the power amplifier).

***void ht_monarch_scan(void)*:** this function scans for the region in which the device is. Additionally, this function will call *ht_callback*, and this will set the global variable called *ht_region_monarch*, with a number from 1 to 6 with the region.

***void ht_send_message(void)*:** this function setups the device to send a message, disabling the Monarch scan feature, and calls *ht_send_frame* to properly sends the message.

5 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).

6 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