STANDALONE MODE SAMPLE APPLICATION FOR ARDUINO ENVIRONMENT (Version 1.0.1)

Description

STANDALONE MODE SAMPLE APPLICATION FOR ARDUINO ENVIRONMENT (Arduino_standalone_mode_sample_app) is a sample application for LPWA transmission using Standalone mode on CXM150x in the Arduino development environment. The MCU to be used is STMicroElectronics STM32L073 or Espressif ESP32.

Communication control, power control and basic payload data generation are performed by the CXM150x itself according to the EEPROM setting of the CXM150x, and the host MCU performs only partial payload change and the operation mode setting at startup. As an example of attaching user data to a payload, this program replaces the end of the payload (bit 117 – 127) that CXM150x automatically generated with a tick (program run time).

At the time of the start-up, the callback function to be called when an error occurs is registered.

For details on each function and how to build the application, refer to the CXM150x HOST I/F Specification, CXM150x Configuration Manual, CXM150x Programmer's Manual and CXM150x Application Manual.

Hardware Preparation

When using the CXM150x DK-Board in conjunction with the STMicroElectronics NUCLEO-L073RZ evaluation board, install the CXM150x DK-Board on the NUCLEO-L073RZ Arduino connector (See DK Startup Manual).

Otherwise, wire the CXM150x according to the application circuit example in the data sheet, and connect it to the host MCU as follows. If the input/output voltage of the MCU is other than 1.8V - 2V, connect it to the CXM150x via the level converter as necessary.

CXM150x pin (DK board pin)	STM32L073 pin	ESP32 pin
ENABLE (D9)	PC 7	GPIO 19
WAKEUP (D12)	PA 6	GPIO 22
INT_OUT2 (D6)	PB 10	GPIO 4
RX (D8)	PA 9	GPIO 17
TX (D2)	PA 10	GPIO 16
CTS (D3)	PB 3	GPIO 15
RTS (D5)	PB 4	GPIO 14

Building Applications and Writing to MCU

The Arduino IDE is used to build and write applications.

Before generating the STM32L073 application, install the "STM32 MCU based board" board library using the board manager. Also enable '#define FOR_STM32_DRIVER' in the source file CXM150x_Port.cpp and comment out '#define FOR_ESP32_DRIVER'.

For ESP32, install the "ESP32" board library with the board manager beforehand. Also enable '#define FOR_ESP32_DRIVER' in the source file CXM150x_Port.cpp and comment out '#define FOR_STM32_DRIVER'.

To use the transmit duty limit function in the CXM1504GR, change the definition of the macro TX_DUTY_USE in main_standalone_mode_sample_app.cpp as follows:

```
#define TX_DUTY_USE                             (1)
```

To build and write:

・Unzip the distributed source file zip package and open standalone_mode_sample_app.ino from the Arduino IDE.

・In the Arduino IDE, select the board you want to use from the [Tools] menu and set the required settings.

・Build and write to MCU according to the operation of Arduino IDE

Additional Information

・Supported firmware version

System firmware version (GNSS firmware version)
FY0100_RA2400 (17166,3dac91c,122) or later

The following EEPROM settings are involved in the operation of this application.

EEPROM function	Description
INT_OUT2	Generates a signal to notify the CXM150x a specified time (in milliseconds) before it starts sending UARTs. Must be non-zero. A setting of 10 is recommended.
SM_TOUT	Specifies the time after which CXM150x automatically transitions to normal operation mode. Set to 0 (automatic transition disabled).
AUTOPLD_COLLECT	Set the data collection start time for the auto-payload data collection by the number of seconds before the data is sent. Must be non-zero. A setting of 5 is recommended.
WAKEUP_CTRL	Sets whether the UART interface circuit is automatically powered off. Set it to 0 (do not turn it off automatically).

AUTO_PERIODIC_SELECT PROFILE_SELECT MIN_DSLP_TIME DSLP_BUP	Set the value appropriate for the transmission profile to be used. See the CXM150x Application Manual and the CXM150x Configuration Manual for configuration.
AUTOPLD_SRC_SELECT AUTOPLD_LAT_BASE AUTOPLD_LON_BASE AUTOPLD_LAT_RANGE AUTOPLD_LON_RANGE AUTOPLD_LAT_RES AUTOPLD_LON_RES AUTOPLD_SRC6_BIT_WIDTH AUTOPLD_SRC7_BIT_WIDTH AUTOPLD_HEIGHT_OFFSET AUTOPLD_SRC1_BIT_POS AUTOPLD_SRC2_BIT_POS AUTOPLD_SRC3_BIT_POS AUTOPLD_SRC4_BIT_POS AUTOPLD_SRC5_BIT_WIDTH AUTOPLD_SRC5_BIT_POS AUTOPLD_SRC6_BIT_WIDTH AUTOPLD_SRC6_BIT_POS AUTOPLD_SRC7_BIT_WIDTH AUTOPLD_SRC7_BIT_POS AUTOPLD_SRC8_BIT_POS AUTOPLD_SRC9_BIT_WIDTH AUTOPLD_SRC9_BIT_POS	Specifies the data and placement to be set in the payload. See the CXM150x Application Manual and the CXM150x Configuration Manual for configuration.

This application is tested with the following versions of each software/library in the Arduino environment.

Software / Library	Version	Information URL
Arduino IDE	1.8.15	https://www.arduino.cc/software
STM32 MCU base board	2.0.0	https://github.com/stm32duino/Arduino_Core_STM32
ESP32	1.0.5	https://docs.espressif.com/projects/arduino-esp32/en/latest/