SRM100A EVB User Manual

Rev.03

January. 16, 2020

Contents

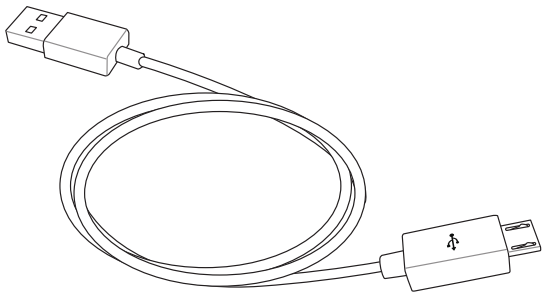
Model	F/W
SRM100A_EVB	-

Hard Ware

Evaluation Kit Component

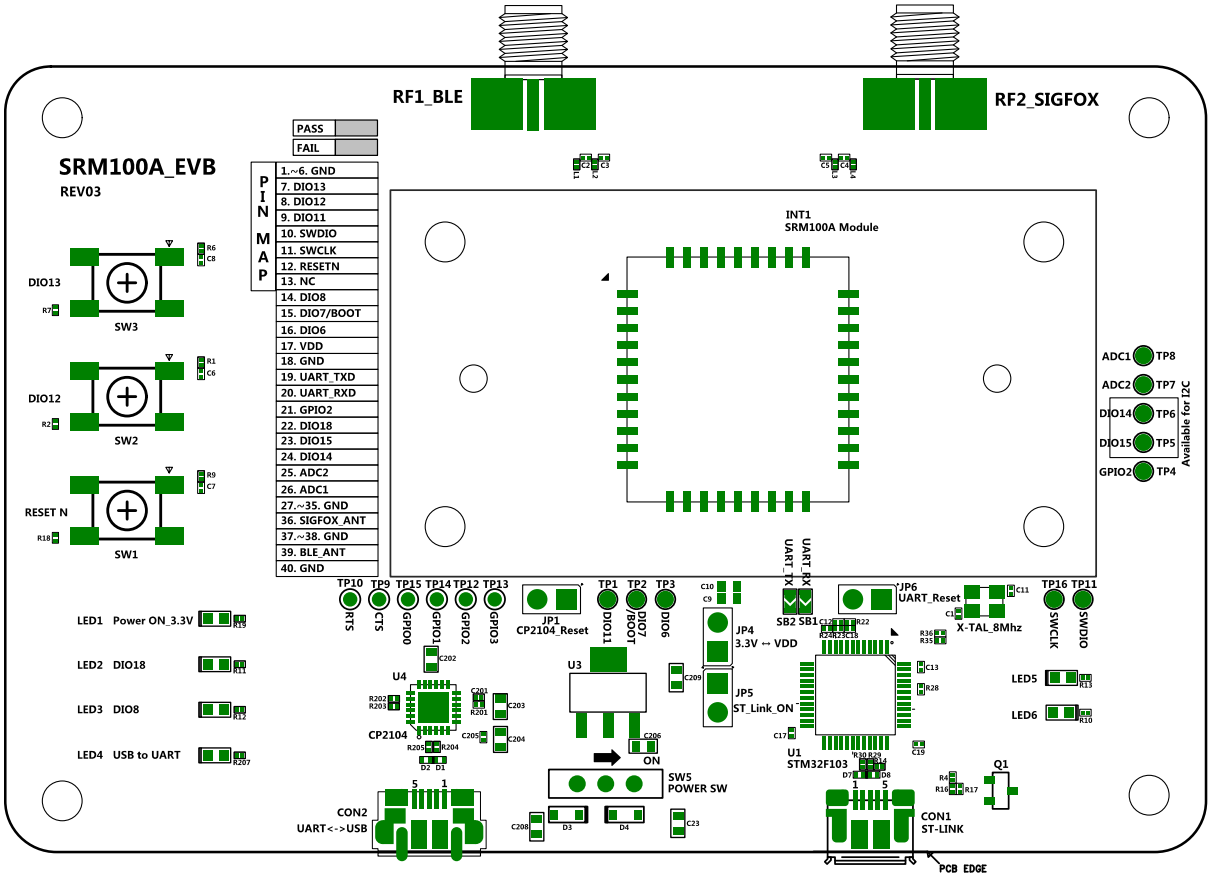
SRM100A_EVB Evaluation Kit Component

1) SRM100A_EVB(Rev.4): 1EA
2) SMA Antenna: 1EA
3) Micro USB cable: 1EA



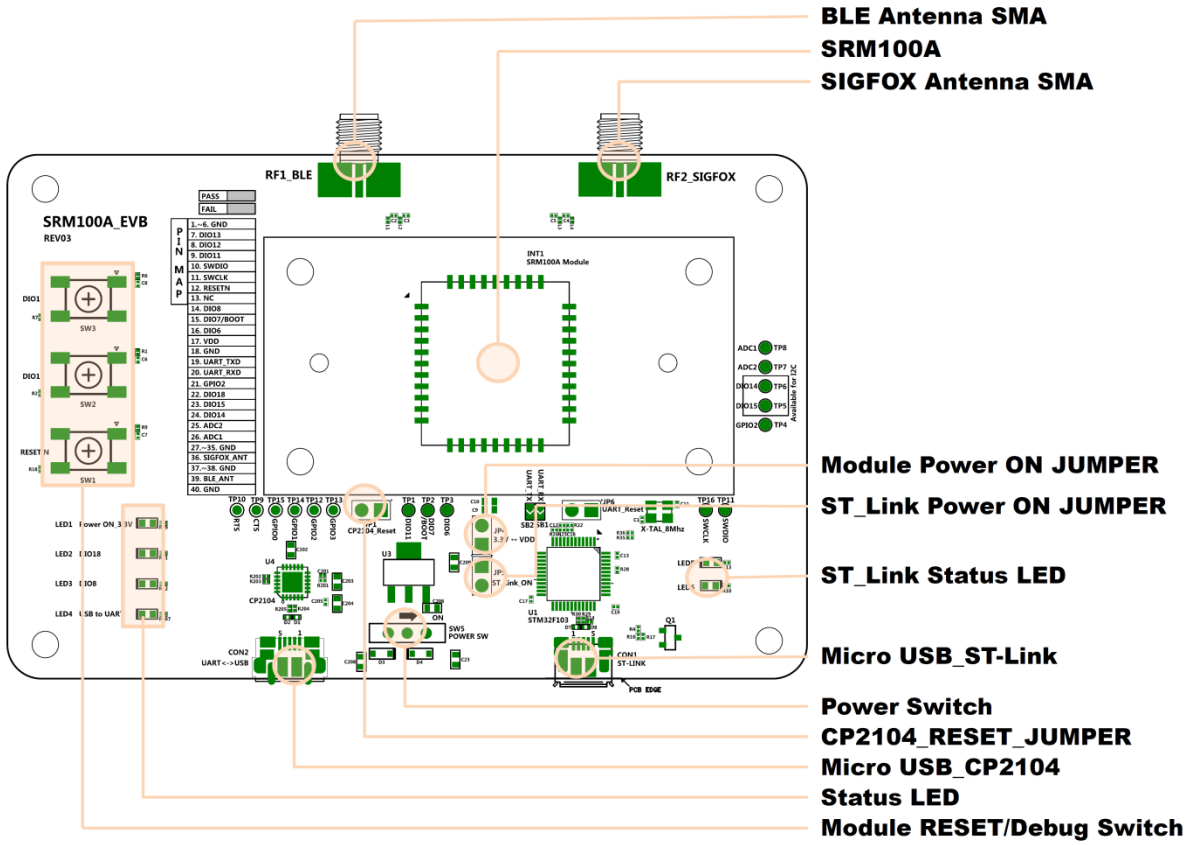
Micro USB cable

Antenna



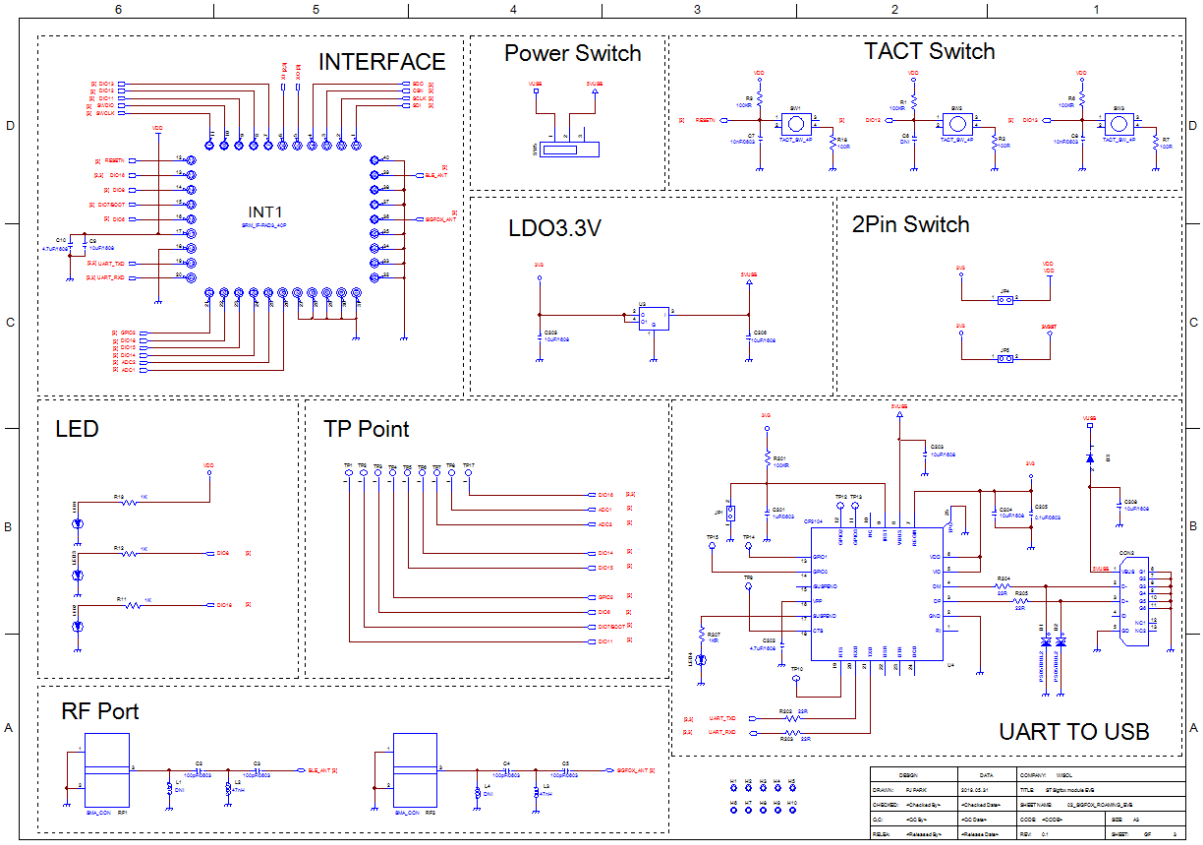
SRM100A_EVB(REV04)

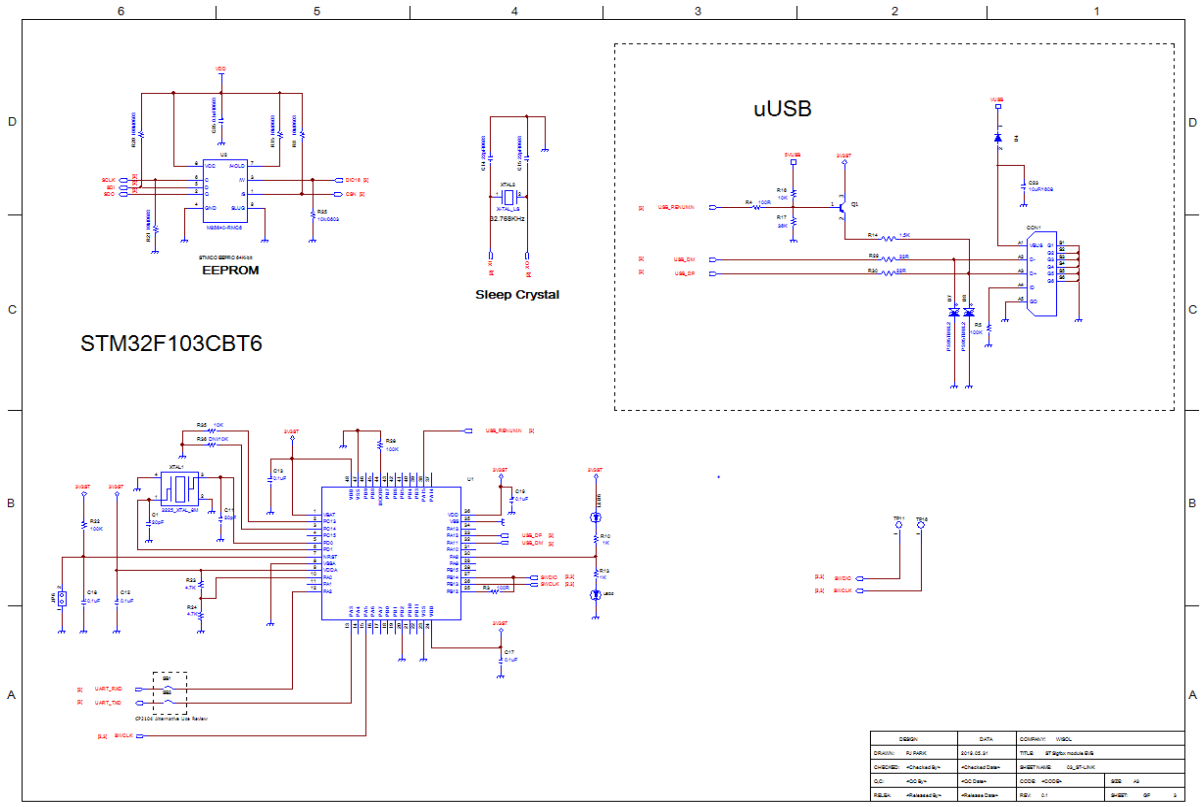
SRM100A_EVB Board



- **BLE Antenna SMA :** BLE connector for Antenna
- **SRM100A :** Sigfox Roaming module
- **WIFI Antenna SMA :** WIFI connector for Antenna
- **Module power Jumper :** SRM100A power supply jumper PIN
- **ST_Link Power on Jumper :** When downloading F/W using ST_Link
- **ST_Link Status LED :** ST_Link operation status LED
- **Micro USB_ST_Link :** Micro USB Connector
- **Power Switch :** EVB Power On/OFF Switch
- **CP2104_RESET_JUMPER** : CP2104 reset
- **Micro USB_CP2104** : Micro USB Connector
- **Staus LED** : Power On, USB to UART, I/O operation status check LED
- **Module RESET/Debug Switch** : RESET Tact Switch

Schematic


INTERFACE
INT1
Power Switch
TACT Switch
LDO3.3V
2Pin Switch
LED
TP Point
RF Port
UART TO USB


EEPROM
Sleep Crystal
uUSB
STM32F103CBT6

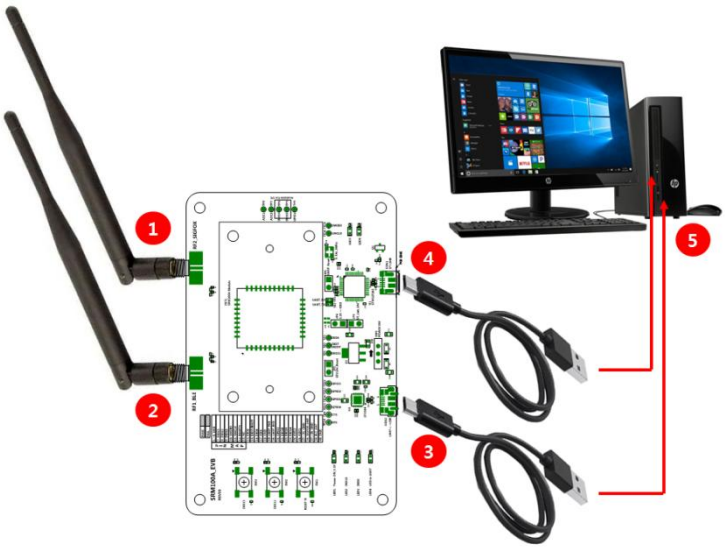
Connector PIN Description

Pin No.	Pin name	Type	Descript	Remark
1~6	GND	GND	Common ground	
7	DIO13	I/O	General purpose digital I/O	
8	DIO12	I/O	General purpose digital I/O	
9	DIO11	I/O	General purpose digital I/O	
10	SWDIO	I	Serial wire clock	
11	SWCLK	I/O	Serial wire debug data in/output	
12	RESETN	I	System reset	
13	N.C	N.C	not connected.	
14	DIO8	I/O	General purpose digital I/O	
15	DIO7/BOOT	I/O	Bootloader pin, General purpose digital I/O	
16	DIO6	I/O	General purpose digital I/O	
17	VDD	VDD	Supply voltage input, +3.3Vdc typ.	
18	GND	GND	Common ground	
19	UART_TXD	O	Uart tx data	
20	UART_RXD	I	Uart Rx data	
21	GPIO2	O	Signal monitor pin	
22	DIO18	I/O	General purpose digital I/O	
23	DIO15	I/O	General purpose digital I/O, I2C1_DAT	
24	DIO14	I/O	General purpose digital I/O, I2C1_CLK	
25	ADC2	I	ADC input 1	
26	ADC1	I	ADC input 2	
27~35	GND	GND	Common ground	
36	SIGFOX_ANT	RF I/O	Sigfox RF in/out put	
37,38	GND	GND	Common ground	
39	BLE_ANT	RF I/O	BLE RF in/out put	
40	GND	GND	Common ground	

Test Program

Evaluation board Connection

1. SRM100A_EVB connect to Window PC by USB cable.

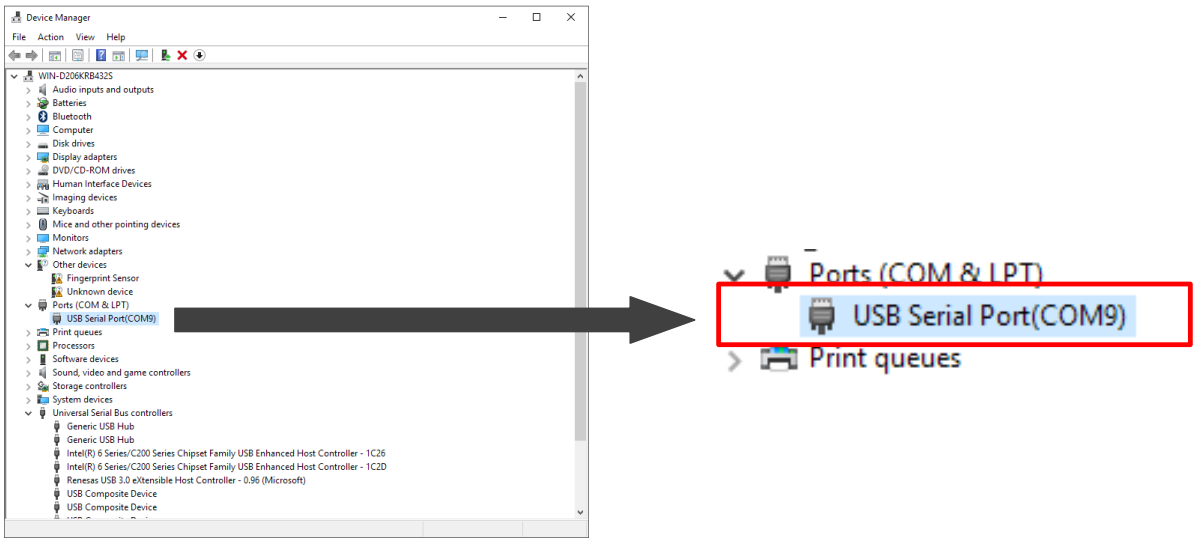


(1) SIGFOX Multi-Band Antenna
(2) BLE Antenna
(3) Micro USB cable_CP2104
(4) Micro USB Cable_ST_Link
(5) PC

Program execution

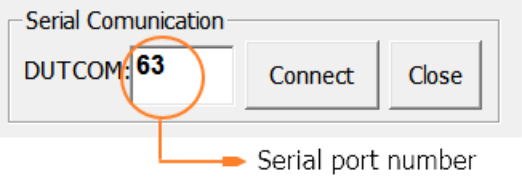
1. SRM100A_EVB connected serial-port in Windows PC, and then check the COM-port number in device manager.

- USB Serial Port(Com□□)



[Fig. SRM100A_EVB serial port]

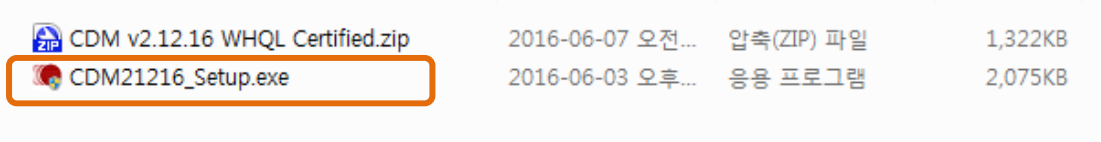
2. Run serial communication program "SRM100A_AT_TEST.exe"
3. Write serial port Number in 'DUTCOM' BOX, and then 'connect' click.



[Fig. SRM100A_EVB serial port number]

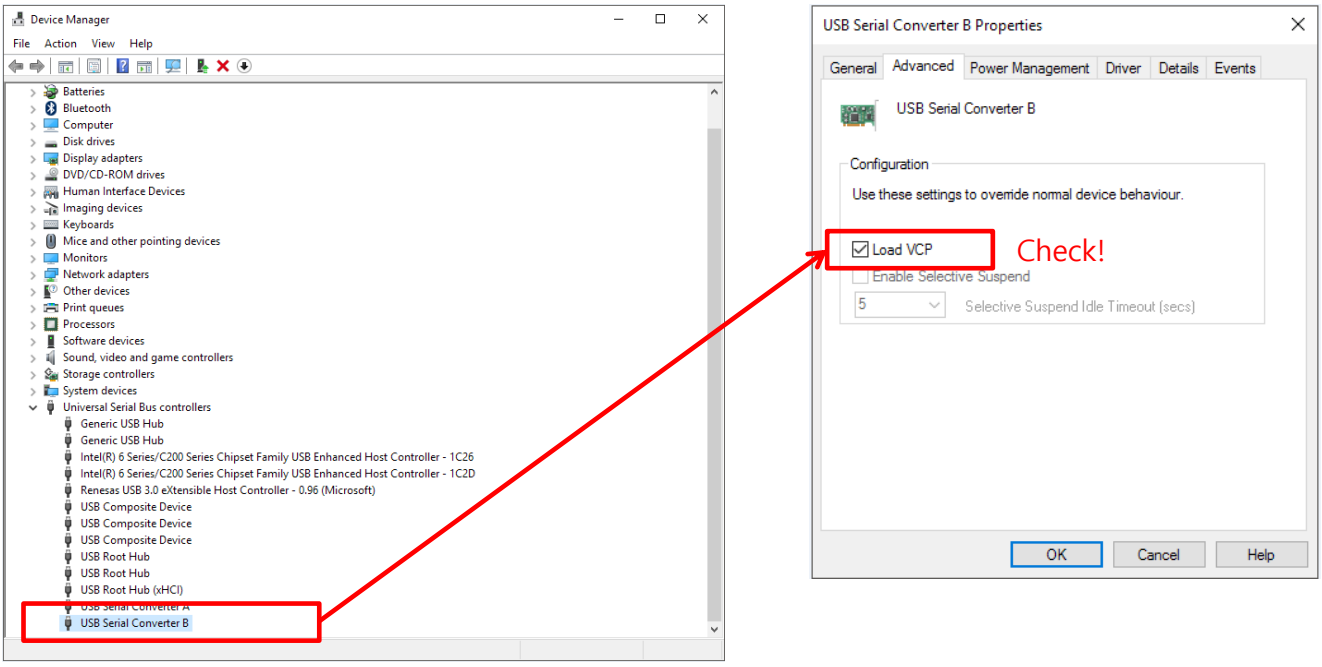
Install USB driver

1. Execute "CDM21216_Setup.exe" file.



[Fig. USB driver set-up file]

2. Setting device manager in Windows.



[Fig. Setting device manager]

Test Program

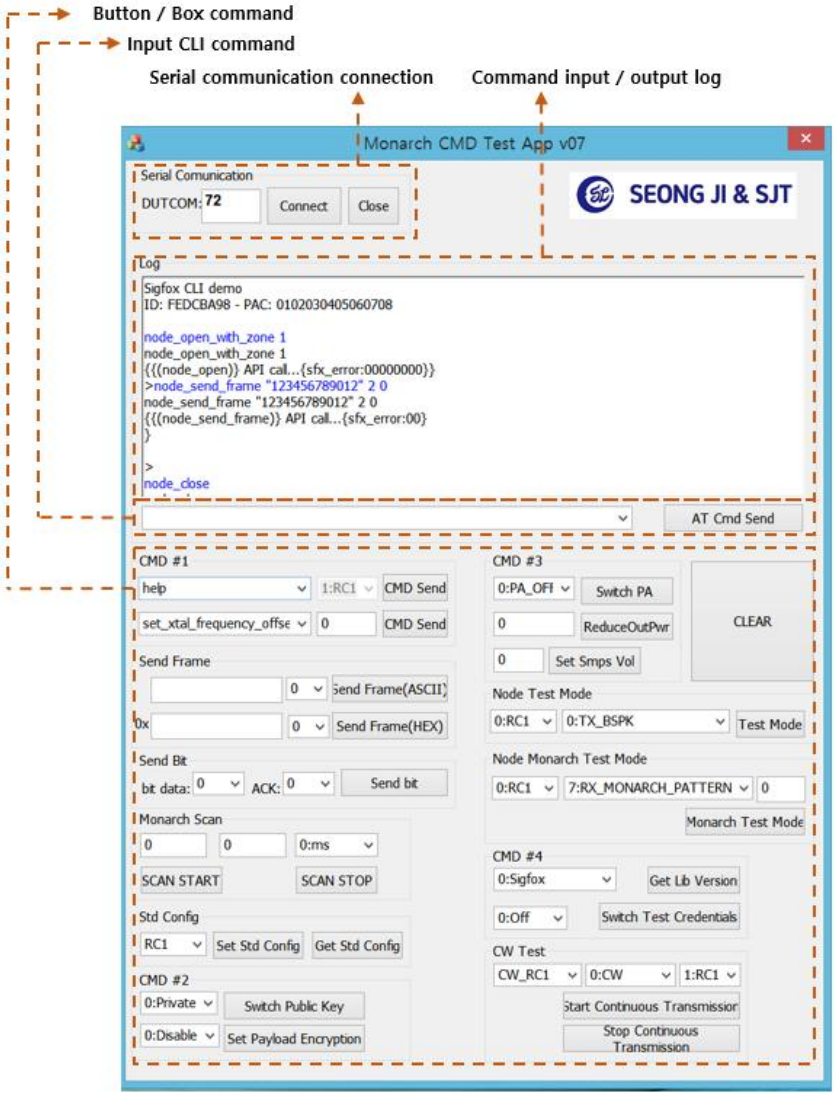
Test program Description (MONARCH_CMD_Test_App_vxx.exe)

This program is for controlling and debugging the SRM100A

You may use another serial communication terminal. However, this program is designed for the SRM100A so that you can enter commands easily.

Serial communication

- Baud Rate : 115200 bps
- Data bits: 8
- Stop bits: 1
- Parity: None



[Fig. Monarch Command Tool program**(v07)**]

CLI command set

A typical serial terminal emulator can also be used to control the EVK instead of the proposed test SW. In that case the following parameters should be used:

- Speed : 115200 bauds
- Data bits: 8
- Stop bits: 1
- Parity: None
- String that specifies the number and types of arguments the command accepts.
- The argument specifiers are:
  * - u: one-byte unsigned integer.
  * - v: two-byte unsigned integer
  * - w: four-byte unsigned integer
  * - s: one-byte signed integer
  * - b: string. The argument can be entered in ascii by using quotes, for example: "foo". Or it may be entered in hex by using curly braces, for example: { 08 A1 f2 }. There must be an even number of hex digits, and spaces are ignored.
  * - n: indicates this is a 'n'ested command. The action points to a table of subcommands. If used, this must be the only specifier. It also adds one to the total argument count of the complete command.
  * - l eight-byte unsigned integer
  * Integer arguments can be either decimal or hexidecimal.
  * A 0x prefix indicates a hexidecimal integer. Example: 0x3ed.

Table 1 APIs

Name	Arg	Arguments description	Description
node_close		None	Closes the Sigfox library, resetting its state
node_open	u	**rc**: pointer to sfx_rc_t type representing the RC number (RC1=1, RC2=2, RC3C=3, RC4=4, RC5=5 or RC6=6).	
Node_open_with_zone	u		
node_get_info		None	
node_get_version	u	**type**: The type of version (0=Sigfox, 1=MCU, 2=RF, 5=Monarch, 6=Device)	
node_send_bit	uuu	**bit_value**: bit value to send (0 or 1) **tx_repeat**: tx repeat value (default : 2) **initiate_downlink_flag**: wait for a response after transmitting. (0 or 1)	This function is used to send a single bit. It is mainly used when the node seeks downlink data (and not to transmit).
node_send_frame	buu	**cust_data**: pointer to the data to transmit ex) ASCII : “12345678” Hexa : {0102030405060708} **tx_repeat**: tx repeat value (default : 2) **initiate_downlink_flag**: wait for a	DM00365435.pdf Please refer to page 9 of the “DM00365435.pdf” file

		response after transmitting. (0 or 1)	
Node_execute_monarch_scan	uuu	**rc_capability**: rc 6 5 4 3 2 1 bit 5 4 3 2 1 0 **time**: scan time **time_unit**: 0: ms, 1:sec, 2:min, 3:hour	Execute Monarch scan. rc_capability, time, unit
Node_stop_monarch_scan		None	This function stops any ongoing RC scan
node_set_std_config	wwwv	**config_word1**: ch1 ~ 32 for RC2,4 **config_word2:** ch33~64 for RC2,4 **config_word3:** ch65~86 for RC2,4 **timer_enable :** (0,1) for RC2,4	DM00365435.pdf Please refer to page 10 of the "DM00365435.pdf" file
Node_get_std_config		none	Get std_config value.
start_continuous_transmission	wu	**frequency**: Frequency at which the signal has to be generated **type**: Type of modulation to use in continuous mode (SFX_NO_MODULATION=0 SFX_DBPSK_100BPS=1 SFX_DBPSK_600BPS=2)	Executes a continuous wave or modulation depending on the parameter type
stop_continuous_transmission		None	Stop the current continuous transmission
node_test_mode	uu	**rc** : pointer to sfx_rc_t type representing the RC number (0, 1, 2, 3, 4, 5 or 6). **test_mode** : (SFX_TEST_MODE_TX_BPSK =0 SFX_TEST_MODE_TX_PROTOCOL =1 SFX_TEST_MODE_RX_PROTOCOL =2 SFX_TEST_MODE_RX_GFSK =3 SFX_TEST_MODE_RX_SENSI =4 SFX_TEST_MODE_TX_SYNTH =5 SFX_TEST_MODE_TX_FREQ_DISTRIBUTION =6 SFX_TEST_MODE_TX_BIT=11 SFX_TEST_MODE_PUBLIC_KEY=12 SFX_TEST_MODE_NVM=13)	Sigfox test mode rc : 0 : RC1 1 : RC2 2 : RC3A 3 : RC3C 4 : RC4 5 : RC5 6 : RC6
node_monarch_test_mode	uuu	**rc** : pointer to sfx_rc_t type representing the RC number (0, 1, 2, 3, 4, 5 or 6). **test_mode** : (SFX_TEST_MODE_RX_MONARCH_PATTERN_LISTENING_SWEEP=7 SFX_TEST_MODE_RX_MONARCH_PATTERN_LISTENING_WINDOW=8 SFX_TEST_MODE_RX_MONARCH_BEACON=9 SFX_TEST_MODE_RX_MONARCH_SENSI=10) **rc_capability**: rc 6 5 4 3 2 1 bit 5 4 3 2 1 0	Sigfox monarch test mode rc : 0 : RC1 1 : RC2 2 : RC3A 3 : RC3C 4 : RC4 5 : RC5 6 : RC6
switch_public_key	u	**key** : private=0, public=1	Switch device on public or private key.
Switch_test_credentials	u	**credentials :** 1 : test ID,PAC 0 : module ID, PAC	Set test credentials 1=On, 0=Off
set_payload_encryption	u	**enc** : encryption enable : 1 disable : 0	Payload encription
switch_pa	u	**pa** : set external power amplifiler (1 if a PA, 0 if not.).	Instructs the library to configure the S2-LP for a external PA (Power Amplifier).
set_smps_voltage	u	**smps**: smps voltage of the device (1.2V=1…. 1.8V=7) The default is to use the S2-LP at 1.8V	Instructs the library to configure the S2-LP with a user defined smps frequency
get_smps_voltage		None	Get SMPS voltage
set_rssi_offset	u	**rssi_value** : Rssi offset value in dB	Set an RSSI offset for the RSSI. Very useful if the RF frontend has an LNA or to calibrate the RSSI measurement.
get_rssi_offset		None	Get the RSSI offset for the RSSI
set_frequency_offset	w	**xtal** : xtal value in Hz	Sets the Vender frequency of the S2-LP in Hertz (default is 50MHz).
get_frequency_offset		None	Get Vender frequency
set_xtal_frequency_offset	w	**freq_offset**: RF offset value in Hz	Sets the RF frequency offset in Hertz (default is 0 Hz).

get_xtal_frequency		None	Get xtal frequency
reduce_output_power	v	**o_pwr**: power reduction in half dB	Reduces the output power of the transmitted signal by a facor (reduction*0.5dB against the actual value)
get_reduce_output_power		None	Get reduce output power
set_lbt_thr_offset	u	**lbt** : LBT threshold offset	Set LBT threshold offset
get_lbt_thr_offset		None	Get LBT threshold offset
get_id		None	ID stored in the current node
get_pac		None	PAC stored in the current node
get_rcz		None	RCZ stored in the current node
get_lib_version	u	**lib_ver** : 0 : Sigfox, 1 : MCU_API 2 : RF_API, 5 : MONARCH_API 6 : DEVICE_CONFIG_API	Get version of specified module
_set_rcz	u	**rc**: pointer to sfx_rc_t type representing the RC number (RC1=1, RC2=2, RC3C=3, RC4=4, RC5=5 or RC6=6).	Set rc
get_swver		None	Get software version
sleep		None	Sleep mode(Wake up toggle GPIO13)
enter_service_manager		None	Start BLE OTA Manager (*The flash is erased.)
ble_get_mac		None	Return MAC address
ble_set_beacon_data	b	**advertising_data :** Max 21byte	Set the advertising data. hex: *ble_set_beacon_data {000102030405060708090a0b0c0d0e0f1011121314}* *string:* *ble_set_beacon_data "123456789012345678901"*
ble_send_noti_Character	b	**notification_data :** Max 14byte	Set the notification data. hex: *ble_send_noti_Character {000102030405060708090a0b0c0d}* *string:* *ble_send_noti_Character "12345678901234"*
ble_set_read_Character	b	**read_data :** Max 14byte	Set the read data. (Same as notification data) hex: *ble_set_read_Character {000102030405060708090a0b0c0d}* *string:* *ble_set_read_Character "12345678901234"*
ble_start	uvv	**adv_type**: 0: Connectable undirected advertising 1: Connectable directed advertising 2: Scannable undirected advertising 3: Non connectable undirected advertising **Advertising_Interval_Max**: 32(20.000 ms)~ 16384(10240.000 ms) **Advertising_Interval_Min**: 32(20.000 ms)~ 16384(10240.000 ms)	Start ble for the option. In the connected mode, the write value is output to Serial.. *modified_event : 0x00 0x00…(8byte)*
ble_set_tx_power_lvl	uu	**high_power** : 0-disalbe 1-enable **level** : 0: -14 dBm (High Power) 1: -11 dBm (High Power) 2: -8 dBm (High Power) 3: -5 dBm (High Power) 4: -2 dBm (High Power) 5: 2 dBm (High Power) 6: 4 dBm (High Power) 7: 8 dBm (High Power)	Set the power of tx
ble_test_tx	uuu	**Frequency**: 0(2042MHz)~39(2480MHz) **Length**:0-255 **Payload**: 0: Pseudo-Random bit sequence 9 1: Pattern of alternating bits '11110000' 2: Pattern of alternating bits '10101010' 3: Pseudo-Random bit sequence 15 4: Pattern of All '1' bits 5: Pattern of All '0' bits 6: Pattern of alternating bits '00001111' 7: Pattern of alternating bits '0101'	Start ble tx test.
ble_test_rx	u	**Frequency**: 0(2042MHz)~39(2480MHz)	Start ble rx test.
ble_test_stop		None	Stop ble test. Returns the number of

			received packets.
ble_tone_start	u	**Frequency**: 0(2042MHz)~39(2480MHz)	Start the ble tone test.
ble_tone_stop		None	Stop the ble tone test.
ble_reset		None	Reset the ble.

Getting started

The module requires Device ID and Pac code.

You can get them from SIGFOX.

If you do not have them, enter the test device ID and test pac code for the test

Test device ID : 0xFEDCBA98

Test Pac code : 0x0102030405060708

- Be sure to enter the device ID and then enter Pac code.

Use the Monarch CMD Tool program.

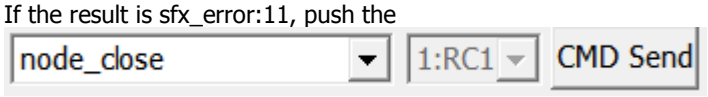
Do not connect the USB cable to con1. Because, when power on, S2-LP and EEPROM do not work normally. However, pressing the Reset button will work normally.

RCZ1

1. Open with Zone

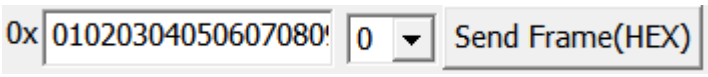
Input : node_open_with_zone 1<CR><LF>
Output : node_open_with_zone 1<LF><CR>
Result : {{(node_open)} API call...{sfx_error:00000000}}<LF><CR>

If the result is sfx_error:11, push the



Input : node_close<CR><LF>
Output : node_close<LF><CR>
Result : {{(node_close)} API call...{sfx_error:00}}<LF><CR>

2. Send Frame(HEXA data)



Input : node_send_frame {0102030405060809} 2 0<CR><LF>
Output : node_send_frame {0102030405060809} 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:30:49	11	0102030405060809		

If the result is sfx_error:60, push the

If result is sfx_error:00 and do not find the message from Sigfox network,

A. node_close -> node_open_with_zone -> Send Frame(ASCII)

B. Reset N(SW1) -> node_open_with_zone -> Send Frame(ASCII)

3. Send Frame(ASCII data)

Input : node_send_frame "1234567890ab" 2 0<CR><LF>
Output : node_send_frame "1234567890ab" 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:52:48	38	3132333435363738393041 42		

If the result is sfx_error:60, push the

If result is sfx_error:00 and do not find the message from Sigfox network,

C. node_close -> node_open_with_zone -> Send Frame(HEX)

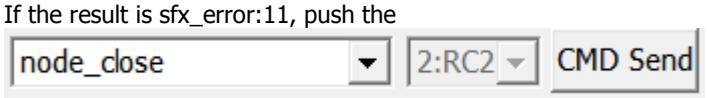
D. Reset N(SW1) -> node_open_with_zone -> Send Frame(HEX)

RCZ2

1. Open with Zone

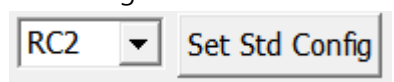
Input : node_open_with_zone 2<CR><LF>
Output : node_open_with_zone 2<LF><CR>
Result : {{(node_open)} API call...{sfx_error:00000000}}<LF><CR>

If the result is sfx_error:11, push the



Input : node_close<CR><LF>
Output : node_close<LF><CR>
Result : {{(node_close)} API call...{sfx_error:00}}<LF><CR>

2. Set Configuration

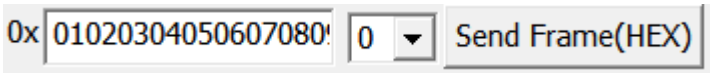


Input : node_set_std_config 1 0 0 1<CR><LF>
Output : node_set_std_config 1 0 0 1<LF><CR>
Result : {{(node_set_std_config)} API call...{sfx_error:00}}<LF><CR>

3. Set PA

Input : switch_pa 1<CR><LF>
Output : switch_pa 1<LF><CR>
Result : {{(switch_pa)} API call...}<LF><CR>

4. Send frame(HEXA data)



Input : node_send_frame {010203040506070809} 2 0<CR><LF>
Output : node_send_frame {010203040506070809} 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:30:49	11	010203040506070809		

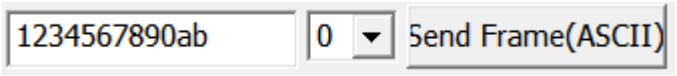
If the result is sfx_error:60, push the

and

If result is sfx_error:00 and do not find the message from Sigfox network,

A. node_close -> node_open_with_zone -> Send Frame(ASCII)
B. Reset N(SW1) -> node_open_with_zone -> Send Frame(ASCII)

5. Send frame(ASCII data)

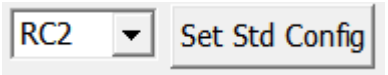


Input : node_send_frame "1234567890ab" 2 0<CR><LF>
Output : node_send_frame "1234567890ab" 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:52:48	38	3132333435363738393041 42		

If the result is sfx_error:60, push the

and



If result is sfx_error:00 and do not find the message from Sigfox network,

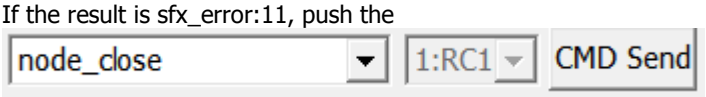
C. node_close -> node_open_with_zone -> Send Frame(HEX)
D. Reset N(SW1) -> node_open_with_zone -> Send Frame(HEX)

RCZ3

1. Open Zone

Input : node_open_with_zone 3<CR><LF>
Output : node_open_with_zone 3<LF><CR>
Result : {{(node_open)} API call...{sfx_error:00000000}}

If the result is sfx_error:11, push the

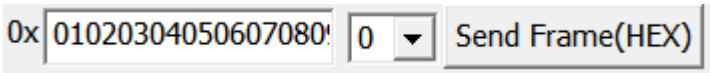


Input : node_close<CR><LF>
Output : node_close<LF><CR>
Result : {{(node_close)} API call...{sfx_error:00}}<LF><CR>

2. Set Configuration

Input : node_set_std_config 3 0x1388 0x9B 0<CR><LF>
Output : node_set_std_config 3 0x1388 0x9B 0<LF><CR>
Result : {{(node_set_std_config)} API call...{sfx_error:00}}<LF><CR>

3. Send frame(HEXA data)



Input : node_send_frame {0102030405060709} 2 0<CR><LF>
Output : node_send_frame {0102030405060709} 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:30:49	11	0102030405060708		

If the result is sfx_error:60, push the

and

If result is sfx_error:00 and do not find the message from Sigfox network,

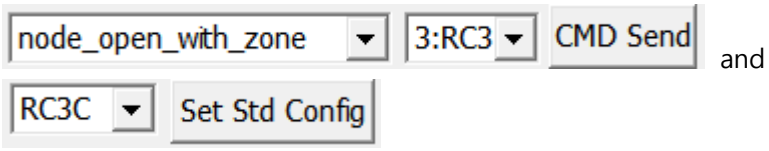
A. node_close -> node_open_with_zone -> Send Frame(ASCII)
B. Reset N(SW1) -> node_open_with_zone -> Send Frame(ASCII)

4. Send frame(ASCII data)

Input : node_send_frame "1234567890ab" 2 0<CR><LF>
Output : node_send_frame "1234567890ab" 2 0<LF><CR>
Result : {{(node_send_frame)} API call...{sfx_error:00}<LF><CR>
Sigfox Network :

Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2019년 4월 23일 오후 2:52:48	38	3132333435363738393041 42		

If the result is sfx_error:60, push the



and

If result is sfx_error:00 and do not find the message from Sigfox network,

C. node_close -> node_open_with_zone -> Send Frame(HEX)

D. Reset N(SW1) -> node_open_with_zone -> Send Frame(HEX)

RCZ4

- Same as RC2 flow

RCZ5

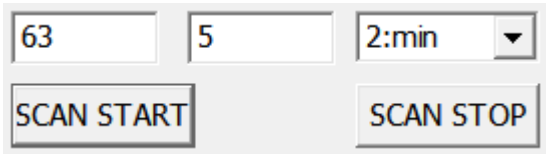
- Same as RC3 flow

RCZ6

- Same as RC1 flow

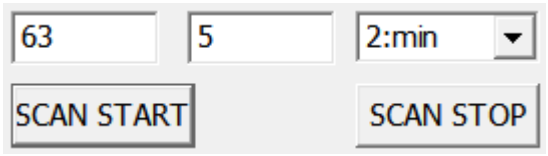
Scan RC Zone

1. Start scan RC zone (RC1 ~ RC6, for 5minutes)



Input : node_execute_monarch_scan 63 5 2<CR><LF>
Output : node_execute_monarch_scan 63 5 2<LF><CR>
Result : {{(SIGFOX_MONARCH_API_execute_rc_scan)} API call...{sfx_error:00}}<LF><CR>
Return : return rc_bit_mask 2
Return rssi -97
Detected RC2!!!:

2. Stop scan RC zone



Input : node_stop_monarch_scan<CR><LF>
Output : node_stop_monarch_scan<LF><CR>
Result : {{(SIGFOX_MONARCH_API_stop_rc_scan_Action)} API call...{sfx_error:00}}<LF><CR>