The Example of Sigfox

WISOL

March 02, 2017

Index of Contents

Introduction	1
Create	3
Initialize	4
UPLINK/DOWNLINK	6
Close	
Example	

Introduction

Device Model

The example of Sigfox documentation includes descriptions to help you understand and develop sifox in the module. It is provided for development purposes only and should always be tested with your design.

WISOL CONFIDENTIAL 2 / 6

Create

void cfg_sigfox_timer_create(void)

function for creating sigfox module timer instance.

Initialize

```
void sigfox_set_state(sigfox_state_s m_state);
```

```
function for setting sigfox state in sigfox scheduler parameter[in] sigfox_state_s m_state : enum value in sigfox_state_s
```

```
typedef enum

{

NONE_S,

SETUP_S, // initialize IO

INIT_S, // send at command of IO test

INIT_R, // receive result

:

TRANSMIT_FRAME_S, // send message without DOWNLINK

TRANSMIT_FRAME_R, // receive result

TRANSMIT_FRAME_DOWNLINK_S, // send message with DOWNLINK

TRANSMIT_FRAME_DOWNLINK_R, // receive result

:

EXIT // unitialize IO and power off

} sigfox_state_s;
```

The sigfox_state_s is the internal state defined in Sigfox scheduler SETUP_S must be set in initial state.

A sigfox module is powered and Uart is initialized at 9600 bps in SETUP_S.

*It is only used for calling sigfox module

UPLINK/DOWNLINK

WISOL CONFIDENTIAL 3 / 6

void sigfox_set_rcz(sigfox_rcz which_rcz) function for setting RCZ 1-4 param[in] sigfox_rcz which_rcz : set RCZ_1, RCZ_2, RCZ_3, or RCZ_4 void sigfox_send_payload(uint8_t * send_data, uint8_t * received_data) function for sending data param[in] uint8_t* send_data : pointer to user data parameter[out] uint8_t *received_data : pointer to downlink buffer *It is only used for calling API bool cfg_sigfox_set_powerlevel(int level); function for setting the power level param[in] int level : power level (0-15 dbm)

Example

How to code

retval true in success

```
1. API mode. : We have to wait for result of sending.
void cfg_examples_sigfox(void)
{
// user data
    uint8_t test_data[SIGFOX_SEND_PAYLOAD_SIZE];
    uint8_t *p_down_link_data;

    //timer Initialize
    APP_TIMER_INIT(APP_TIMER_PRESCALER, APP_TIMER_OP_QUEUE_SIZE, false);

    //sd init
    ble_stack_init_minimal();
```

WISOL CONFIDENTIAL 4 / 6

```
//snek mode enable
      m_module_parameter.sigfox_snek_testmode_enable = 1;
//
    //set test send data
    sprintf((char*)test_data,"AABBCCDDEEFF");
    // create sigfox timer instance
    cfg_sigfox_timer_create();
   //Set the power level
   if(!cfg_sigfox_set_powerlevel(14))
    {
        cPrintLog(CDBG_MAIN_LOG, "ERROR SET POWER LEVEL");
    }
    //set RCZ
    sigfox_set_rcz(RCZ_1);
    sigfox_send_payload(test_data,p_down_link_data);
    cPrintLog(CDBG_MAIN_LOG, "%s %d SIGFOX downlink data:%s, size:%d₩n", __func__, __LINE__,
p_down_link_data, strlen(p_down_link_data));
    while(1)
    {
        sd_app_evt_wait();
    } }
2. Module mode.: module automatically sends and we don't have to wait for result of sending.
    cfg_sigfox_timer_create();
    // initialize and set initial state ins sigfox state
    sigfox_set_state(SETUP_S);
    //set RCZ
    sigfox_set_rcz(RCZ_1);
```

WISOL CONFIDENTIAL 5 / 6

// set flag to decide to receive downlink

cfg_sigfox_downlink_on_off(true);

```
// copy user data to sending buffer
cfg_bin_2_hexadecimal(test_data, SIGFOX_SEND_PAYLOAD_SIZE, (char *)frame_data);

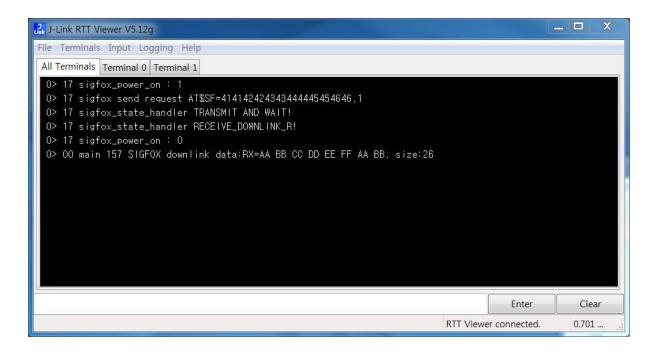
// start sigfox module timer
cfg_sigfox_timers_start();

// check whether sigfox module finishes
while(!sigfox_check_exit_excute());

// stop sigfox module timer
cfg_sigfox_timers_stop();

p_down_link_data = cfg_sigfox_get_downlink_ptr(&down_link_data_size);
```

Result



WISOL CONFIDENTIAL 6 / 6