AND9478/D

AX-SIGFOX-API SIGFOX Compliant Software Stack

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

Introduction

This application note describes the ON Semiconductor SIGFOX API, a SIGFOX compliant protocol stack for use on AX−SIGFOX−API−1 chips. The software stack can easily be incorporated in a customer written application. AX−SIGFOX−API−1 chips are supplied with a preprogrammed SIGFOX identity.

This application note is accompanied with a simple example firmware.

Prerequisities

This chapter discusses the requirements needed to successfully setup a software project with the ON Semiconductor SIGFOX protocol stack.

Software Tools

In order to build a firmware project, the following tools need to be installed:

- AX8052−IDE, version 1.17 or later
- IAR Embedded Workbench for 8051 version 9.30 or later

Project Setup

The easiest way to setup the firmware project is to copy the supplied example project and start modifying it.

If it is desired to setup a project from scratch, the following steps should be used.

The AxCodeBlocks new project wizard should be used to setup the project structure. The defaults should be used.

Include and library directories need to be added to the compiler and linker search paths. In the example project, the directories are “api/include” and “api/lib”.

The following libraries need to be linked in: “libmf−pli−nlpc−1e16x01.r51” and “libmfcrypto−pli−nlpc−1e16x01.r51” (installed by AX8052−IDE), and “libaxsigfox.r51” and “libsfx_v187.r51” (supplied with this kit).

Code Requirements

All files that use SIGFOX protocol stack functions should include “sigfox_ax.h”. Furthermore, the file that contains the main function (usually main.c) also needs to include “sigfox_ax.h”, to ensure that the radio interrupt handler is placed into the interrupt vector table.

The SIGFOX protocol stack uses the wakeup timer (wtimer) facilities of libmf. To initialize it, the following functions should be called early during __low_level_init():

- wtimer0_setclksrc(CLKSRC_LPOSC , 1);// LPOSC (640Hz) x1
- wtimer1_setclksrc(CLKSRC_FRCOSC, 7); // FRCOSC (20MHz) / 64

Furthermore, later during the startup process, but before using any SIGFOX protocol stack functions, wtimer_init() must be called.

API

This chapter documents the API functions of the Axsem SIGFOX protocol stack.

To send or receive SIGFOX messages, use **ONSEMI_send_frame()**. This command opens the Sigfox library and closes it after the frame transmission. For more details about the SIGFOX API see the ICD_Sigfox_Protocol document by SIGFOX. The only difference is that you have to call **ONSEMI_send_frame()** instead of **SIGFOX_API_open()** followed by **SIGFOX_API_send_frame()** and **SIGFOX_API_close()**.

- SFX_ERROR_T ONSEMI_INITIALIZE(UINT8_T FLAGS)

This function ensures everything is properly initalized and the radio turned off. The radio will then automatically be turned on when you call **ONSEMI_send_frame()**. Call this function when the module has been reset or woken from deepsleep. This function does not call **SIGFOX_API_open()**. If this function is not called, the module will not work properly.

flags is a bitfield and can be used to calibrate the microcontrollers oscillator from the radio’s temperature compensated crystal oscillator (TCXO). Possible flags are **ONSEMI_INIT_CALIBRATE_FRCOSC**, **ONSEMI_INIT_CALIBRATE_LPOSC** and **ONSEMI_INIT_DONT_LOAD_SIGFOX_SETTINGS** The last one indicates that the current settings in RAM should be kept and not be overwritten by the values in flash.

- UINT32_T ONSEMI_GET_ID(VOID)
  UINT32_T ONSEMI_GET_INITIAL_PAC_LO(VOID)
  UINT32_T ONSEMI_GET_INITIAL_PAC_HI(VOID)

These functions return the 32 bit ID and the 64 bit PAC. These two numbers are required in order to register the

modem with the SIGFOX network. Both numbers are different for each AX−SIGFOX−API−1 chip. It is up to the application developer to ensure that the customer is able to retrieve these numbers. The example application simply prints them on the DebugLink.

- SFX_ERROR_T ONSEMI_SEND_FRAME(U8 *CUSTOMER_DATA, U8 CUSTOMER_DATA_LENGTH, U8 *CUSTOMER_RESPONSE, BOOL INITIATE_DOWNLINK_FLAG) __REENTRANT

This function sends a SIGFOX datagram to the network. The user supplies the datagram contents using the **customer_data** pointer and the **customer_data_length** variable. Up to 12 bytes may be transmitter (SIGFOX network limitation). Setting **initiate_downlink_flag** to true requests the network to reply with a downlink datagram, whose contents are placed into memory pointed to by **customer_response**. Otherwise, **customer_response** may be 0 or NULL.

- SFX_ERROR_T ONSEMI_SEND_BIT(BOOL BIT_VALUE, U8 *CUSTOMER_RESPONSE, BOOL INITIATE_DOWNLINK_FLAG) __REENTRANT

This function sends a single byte SIGFOX datagram to the network. The contents are either 00 or 01, depending on the value of **bit_value**. Otherwise, the function behaves exactly the same as **ONSEMI_send_frame**.

- SFX_ERROR_T ONSEMI_SEND_OUTOFBAND(VOID) __REENTRANT

This function transmits an out−of−band datagram to the SIGFOX network. The out−of−band datagram contains temperature and battery voltage.

- SFX_STATE_T ONSEMI_GET_TX_STATE(VOID) __REENTRANT

This function returns the state of the transmitter. It is not normally used.

- U8* ONSEMI_GET_SIGFOX_VERSION(VOID) __REENTRANT

This unction returns a pointer to a string describing the SIGFOX protocol stack version.

- VOID ONSEMI_TEST_MODE(SFX_TEST_MODE_T TEST_MODE, SFX_U8 CONFIG) __REENTRANT

This function is a wrapper for the sigfox tests. There are 5 tests that can be executed. It tests up− as well as down−link. For the exact behaviour of the tests, please see the doxygen documentation in the *docu*/*html* directory.

- UINT8_T ONSEMI_SAVE_USER_CONFIG() UINT8_T ONSEMI_LOAD_USER_CONFIG()

These functions save and load the user config (e.g. the center frequency, tx_repeat) to/from flash, making it persistent in case of power loss.

- VOID ONSEMI_SET_TX_CENTER_FREQ(UINT32_T F) VOID ONSEMI_SET_RX_CENTER_FREQ(UINT32_T F) VOID ONSEMI_SET_TX_POWER(UINT8_T DBM) VOID ONSEMI_SET_TX_MODE(UINT8_T MODE) SFX_ERROR_T ONSEMI_TEST_MODE(SFX_TEST_MODE_T TEST_MODE, SFX_U8 CONFIG) ONSEMI_SET_AES_KEY_TYPE(UINT8_T KEY_TYPE)

These low level functions are used for compliance testing, and enable the software stack to transmit a carrier wave only (CW) signal. **ONSEMI_set_tx_center_freq** sets the frequency to transmit, **ONSEMI_set_tx_power** sets the transmit power, and **ONSEMI_test_mode(SFX_TEST_MODE_TX_CW, 0)** and **ONSEMI_test_mode(SFX_TEST_MODE_TX_OFF, 0)** enable and disable transmission of the CW signal.

For further documentation of the ONSEMI−layer in this API, please refer to the doxygen docu in the *docu*/*html* directory.

Callbacks

These callback functions need to be implemented by the application code and are called by the SIGFOX software stack.

- VOID ONSEMI_REPORT_TEST_RESULT(SFX_BOOL STATUS, SFX_U8 __XDATA *FRAME, SFX_S8 RSSI) __REENTRANT

This function gets called in receive test mode whenever a valid down−link datagram is received. Or after timeout. The status indicates whether the frame was received successfully or not.

- VOID SFX_SET_SFLIB_LED(UINT8_T ENABLED) VOID SFX_SET_RADIO_LED(UINT8_T ENABLED) VOID SFX_SET_RX_LED(UINT8_T ENABLED) VOID SFX_SET_TX_LED(UINT8_T ENABLED)

These functions may be used to implement status LEDs. This functions indicate activity of the protocol stack, of the radio, or receive or transmit activity.

Power Management

Normal Startup

During normal startup, the following functions should be called:

- wtimer_init();
  ONSEMI_initialize(CALIBRATE_LPOSC | CALIBRATE_FRCOSC);

Wakeup from Sleep

During wakeup from sleep, the following functions should be called:

- wtimer_init();
  ONSEMI_initialize(DONT_LOAD_SIGFOX_SETTINGS | CALIBRATE_LPOSC | CALIBRATE_FRCOSC);

Specifying DONT_LOAD_SIGFOX_SETTINGS ensures that SIGFOX parameters are not reset to defaults.

Wakeup from Deep Sleep

During wakeup from deepsleep, the following functions should be called:

- wtimer_init_deepsleep();
  ONSEMI_initialize(CALIBRATE_LPOSC | CALIBRATE_FRCOSC);

Example Application

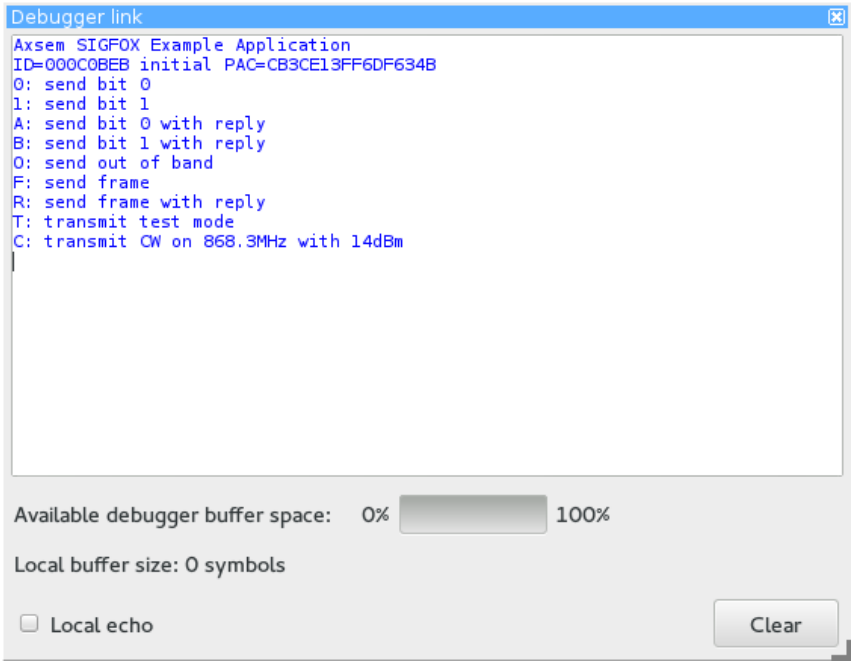

Figure 1.

The example SIGFOX application can be opened by clicking on sigfoxapi.cbp. This is a simple example on how to use the SIGFOX API. It outputs results on the DebugLink channel and expects commands via DebugLink. The following keypresses are recognized:

Table 1.

Key	Action
0	Send Bit 0
1	Send Bit 1
A	Send Bit 0 with reply
B	Send Bit 1 with reply
O	Send Out−of−Band data (temperature and battery voltage)
F	Send Frame
R	Send Frame, with reply
T	Transmit Test mode (sends 10 frames)
C	CW Transmit mode (sends constant carrier wave at 868.3 MHz with 14 dBm)

Registering a Device with the SIGFOX Network

In order to register the Modem with the SIGFOX Network, open an Internet Browser and enter http://backend.sigfox.com. Click on “DEVICE”, then “NEW”. The ID can be obtained using ONSEMI_get_id(), and the initial PAC can be obtained by calling ONSEMI_get_initial_pac_lo() and ONSEMI_get_initial_pac_hi().

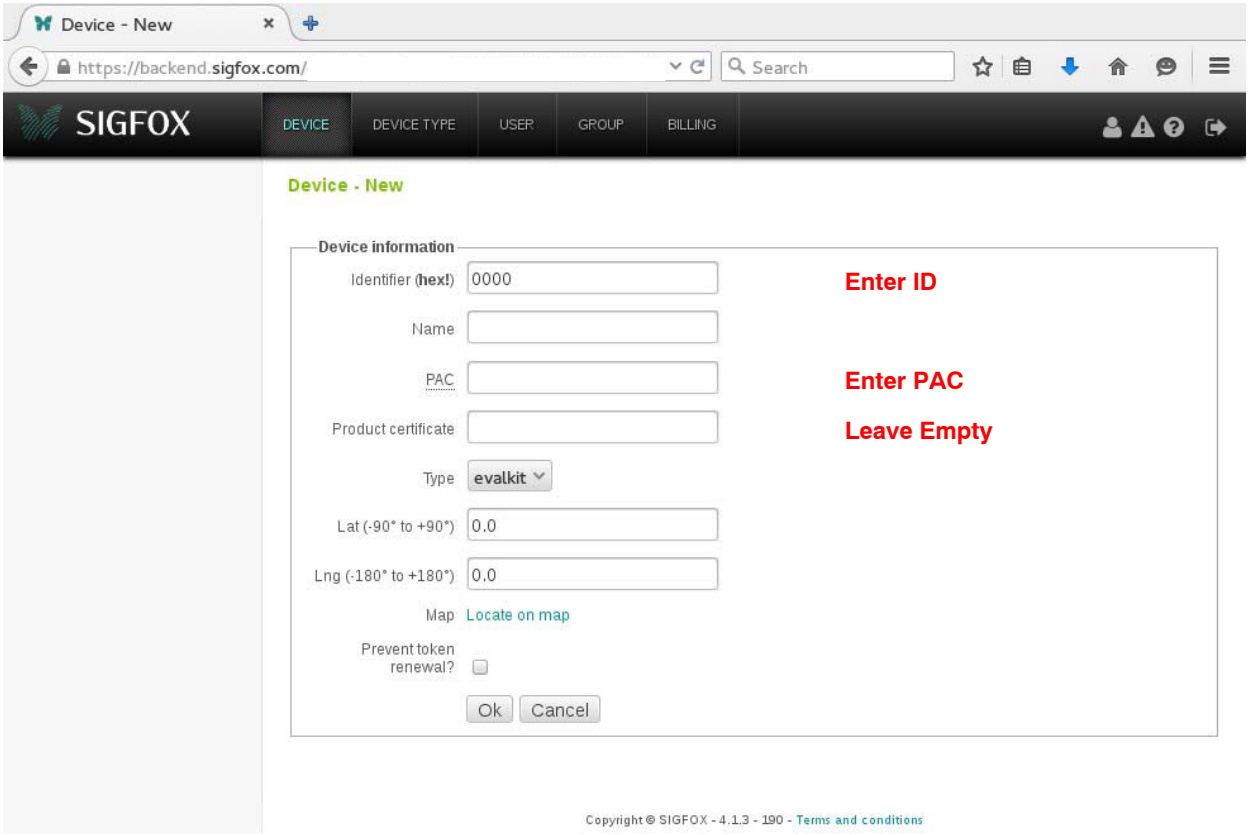

Figure 2.

Clicking again on “DEVICE”, the modem ID should now be listed in the device list.

You are now ready to send your first packet through the SIGFOX network.

Calling:

- static const u8 __code testpacket[] = { 0x00, 0x11, 0x22, 0x33, 0x44 };
  ONSEMI_send_frame(testpacket, sizeof(testpacket), 0, false);

If you now click on the modem's ID, and then "Messages", you should see the packet just transmitted:

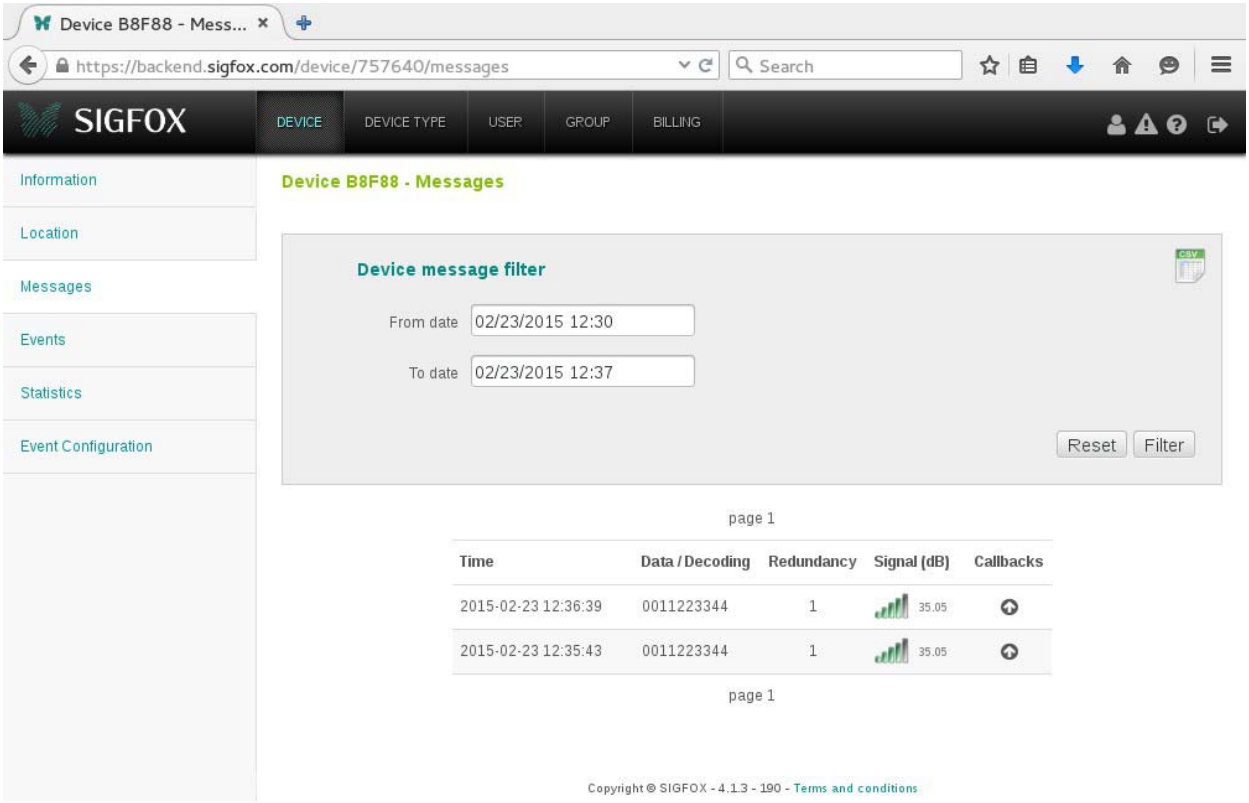

Figure 3.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81−3−5817−1050

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative