External BLE

Firmware design document v1.0

Revision History

REV	DESCRIPTION	DATE	AUTHOR
1.0	Initial Release	09/21/2021	Bijosh

Table of Contents

1. Introduction

Tracking the device when it’s turned OFF has always been a challenge for Zebra’s customers. To address this problem, we have come up with a solution by having a separate low power BLE chip either in the removable battery or in the terminal.

This document covers firmware design for the BLE chip in detail.

1.1 Intended Audience

Zebra SW/EE engineering/integration team

1.2 Reference Materials

https://infocenter.nordicsemi.com/pdf/nRF52810_PS_v1.0.pdf

https://infocenter.nordicsemi.com/index.jsp

1.3 Acronyms

GPIO	General purpose IO
FW	Firmware
BLE	Bluetooth Low Energy
BT	Bluetooth
SD	Soft device
MAC	Machine Access Control
I2C	Inter Integrated Circuit

2. NRF52810 Specifications

	nrf52810
Processor	32-bit ARM Cortex-M4 Processor
RAM	24 KB
Flash	192 KB
NFC	
Package	
Operating channel	
TX power (Max)	+4dBm
RX sensitivity	-96dBm
Support data rate	2Mbps/1Mbps
TX current	
RX current	
Sleep current	
Serial interface	1 x Master/Slave SPI
	1 x Two-wire interface (I^2C)
	UART (RTS/CTS)

Supply voltage	1.7 -3.6 V

3. Modes of firmware

At a given time, the Ble firmware will be operating in one of the below modes. The transitions from one mode to other is triggered by the state of battery thermal and voltage across I2C line. One state to other transition is done by a soft reset.

3.1 Active Mode

In this mode the terminal is active and running. In firmware:

1. I2C enabled
2. Thermal line polling is enabled
3. I2C voltage detection is enabled
4. BLE is disabled

Thermal line pulsing/high in this state.

3.2 Beaconing Mode

If the terminal is turned off and the chip is configured by the EMMs to beacon, the device enters into beaconing mode. In this mode:

1. I2C disabled
2. Thermal polling active
3. I2C voltage detection enabled
4. BLE advertising enabled

NB: Beaconing as to continue for 7days in case of low battery shutdown

Beaconing has to be stopped when the I2C clk voltage reaches 3V

3.3 Low power mode

Device enters this state when:

1. The device is turned off and Ble is not configured to beacon
2. When the battery voltage hits 3V (from beaconing mode)
3. On battery removal (I2C CLK voltage < 1.2)

In this state:

1. I2C disabled
2. Thermal polling enabled
3. I2C voltage detection is enabled
4. BLE disabled

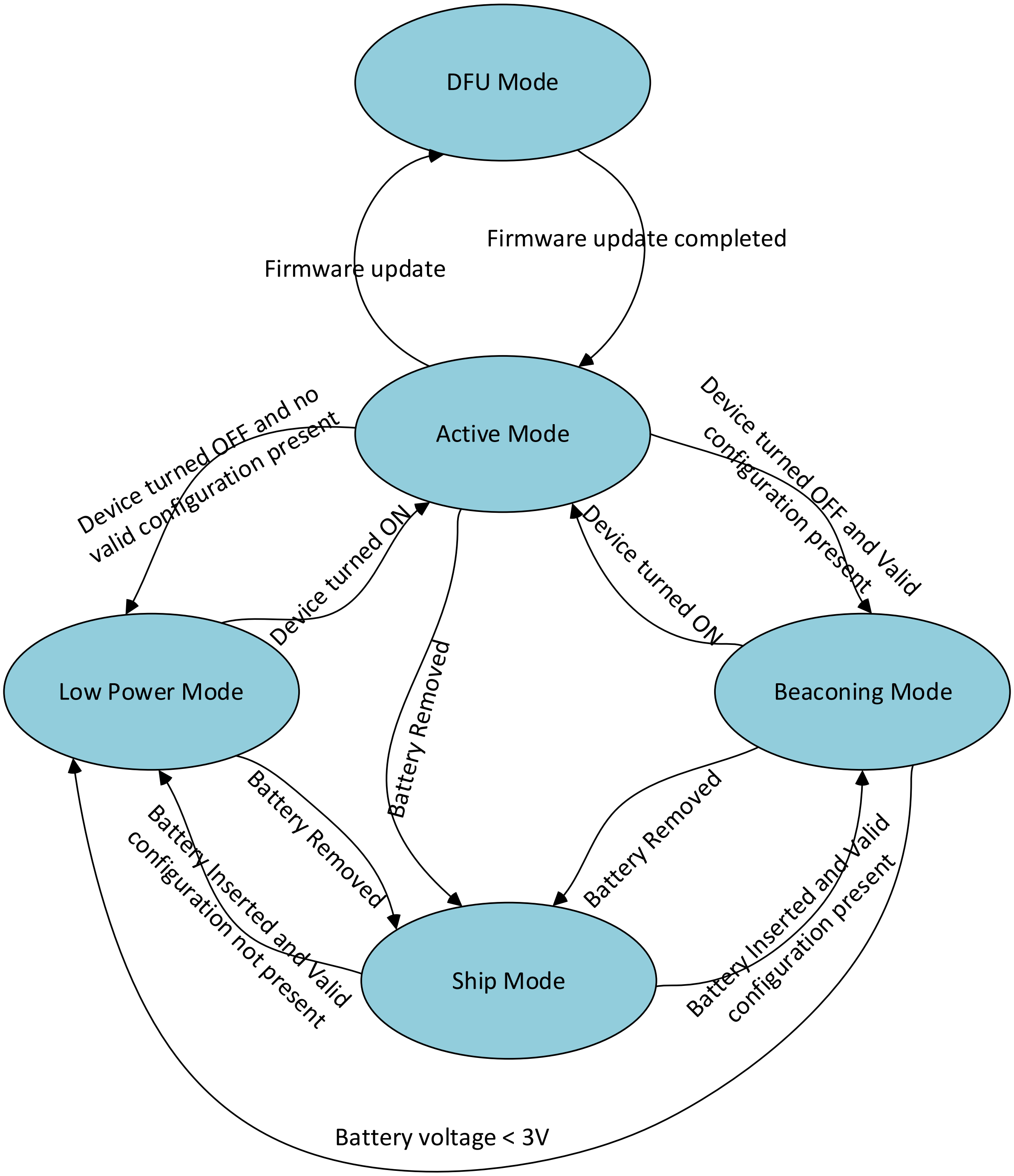
3.4 DFU Mode

Firmware issues a soft reset to put the device in to this mode in order to update firmware/bootloader.

3.5 Shipping mode

Supported only on the devices that does not support removable battery. This is the lowest possible power state that a BLE can get into and is currently supported only on Simba

3.6 State machine


DFU Mode
Firmware update
Firmware update completed
Active Mode
Device turned OFF and no valid configuration present
Device turned OFF and Valid configuration present
Device turned ON
Device turned ON
Battery Removed
Low Power Mode
Beaconing Mode
Battery Removed
Battery Removed
Battery Inserted and Valid configuration not present
Battery Inserted and Valid configuration present
Ship Mode
Battery voltage < 3V

4. The protocol

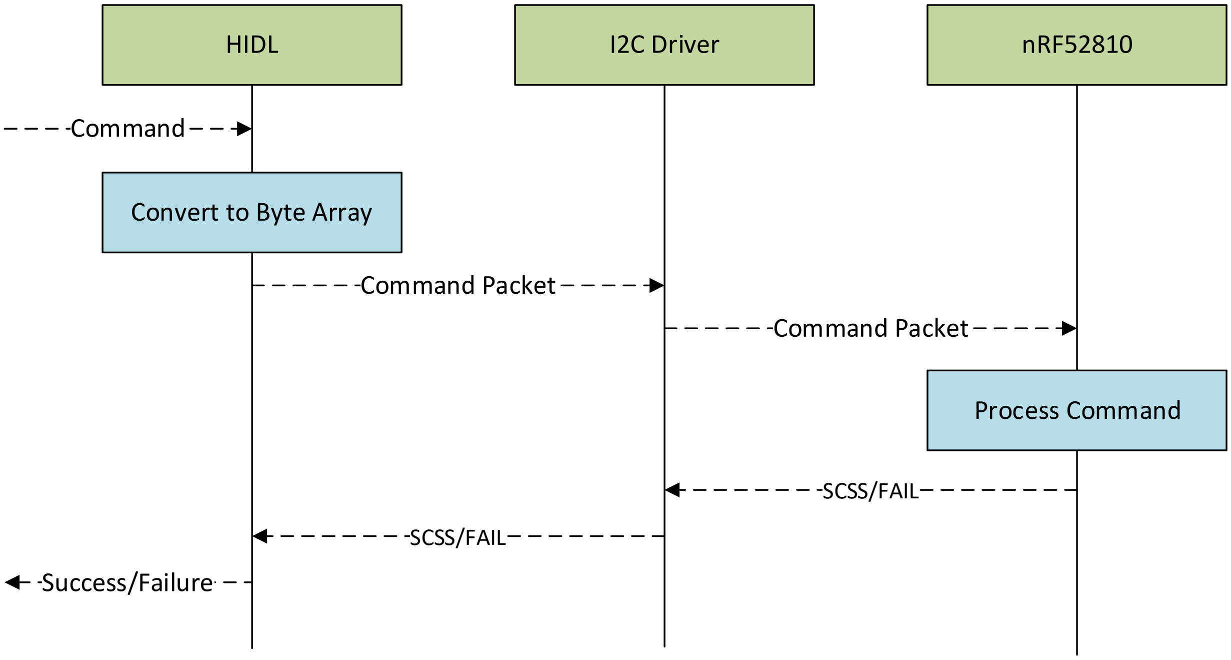
4.1 Protocol

The protocol architecture uses simple master slave communication approach where the terminal always acts as master and the BLE chip as slave. All the commands are initiated by the master.

The slave responds to the commands through success/failure (SCSS/FAIL) packets. The SCSS packets may contain data requested by the master if it is a read request

In case of failure the slave responds with a failure (FAIL) packet. FAIL packets will have an error code corresponding to the failure. The error code can be one among the following:

- ERR_CRC = -1
- ERR_INVALID_CMD = -2
- ERR_INVALID_DATA = -3
- ERR_INVALID_PKT = -4
- ERR_TIMEOUT = -5
- ERR_EXECUTION = -6
- ERR_MEMORY_ALLOC = -7



4.2 External BLE commands & packet structure

All the I2C commands to the chip follow below packet structure:

STX	LEN	CMD	DATA	CRC	ETX

STX => Start byte. This is fixed to 0x0A

LEN => Length of packet excluding STX and ETX bytes

CMD => The command to BLE chip

DATA => The data (if any) associated with the command

CRC => Checksum on the packet (described in detail in section 4.5)

ETX => End byte. This is fixed to 0x0D

Response to each command from the will be through an SCSS/FAIL command. These packets are described in detail at sections 4.3 and 4.4

The bellow sections will be using following legends:

This field is a variable

This field is not required for the command

These fields are fixed

4.2.1 CMD_PING_BLE

Command to read the BLE firmware version from the chip

Command	Read	Write
CMD_PING_BLE	0x01	N.A.

Packet structure:

0x0A	0x02	0x01	DATA	CRC	0x0D

The response (SCSS) to this command will have the current BLE firmware version. FAIL will be sent in case of failure.

4.2.2 CMD_TX_PWR

Command to set the Tx power

Command	Read	Write
CMD_TX_PWR	N.A.	0x82

Packet structure:

0x0A	0x03	0x82	DATA	CRC	0x0D

The data field can have following values:

Power	Value
1db	0x01
-7db	0xF9
-15db	0xF1
-21db	0xEB

The chip responds back with an SCSS in case of success and FAIL in case of failure.

4.2.3 CMD_TX_RATE

Command to set the Tx Rate

Command	Read	Write
CMD_TX_RATE	N.A.	0x83

Packet Structure:

0x0A	0x04	0x83	DATA (2bytes)	CRC	0x0D

Data field can have following values:

Tx Rate	Decimal Value	Byte -1	Byte -2
100ms	160	0x00	0xA0
250ms	400	0x01	0x90
1000ms	1600	0x06	0x40

The chip responds back with an SCSS in case of success and FAIL in case of failure.

4.2.4 CMD_EXT_BEACON

Command to enable/disable the beaconing while the device it turned off.

Command	Read	Write
CMD_EXT_BEACON	N.A.	0x84

Packet Structure:

0x0A	0x03	0x84	DATA (1byte)	CRC	0x0D

The data field can have following values:

Data	Description
0x01	The chip will beacon on turning off the device if a valid beacon data is present in the chip.
0x00	The beaconing functionality is disabled. Device will not beacon even if there is a valid configuration present in the chip.

The chip responds back with an SCSS in case of success and FAIL in case of failure.

4.2.5 CMD_SHIP_MODE

This command is used to put the BLE chip into ship mode. This command is applicable only for the devices with BLE chip in the terminal.

Command	Read	Write
CMD_SHIP_MODE	N.A.	0x85

Packet Structure:

0x0A	0x03	0x85	DATA (1byte)	CRC	0x0D

The data field can have following values:

Data	Description
0x01	Enable ship mode. This will put the chip into system OFF mode. Waking up from system off mode is controlled by HW. Hence there is no separate command to disable the ship mode.

The chip responds back to this command with an SCSS in case of success and FAIL in case of failure.

4.2.6 CMD_BEACON_DATA

This command is used to configure the BLE chip with beacon data.

Command	Read	Write
CMD_EXT_BEACON	N.A.	0x86

Packet Structure:

0x0A	0x1F	0x85	DATA (28 byte)	CRC	0x0D

The data field is split as:

AD Length (1 - Byte)	AD Type (1 Byte)	MFG ID (2 Bytes)	Beacon Code (2 Bytes)	Beacon ID (16 Bytes)	Major No (2 Bytes)	Minor No (2 Bytes)	Rfc RSSI (1 Byte)	Mfg Rsvd (1 Byte)
0x1B	0xFF	0x01F1	0xBEAC	configurable	configurable	configurable	configurable	configurable

Please refer AltBeacn specification for more details.

The chip responds back to this command with an SCSS in case of success and FAIL in case of failure.

4.2.7 CMD_DTM_INIT

DTM test command

4.2.8 CMD_DTM_INS

DTM test command

4.2.9 CMD_DTM_EXIT

DTM test command

4.2.10 CMD_DTM_RESULT

DTM test command

4.2.11 CMD_DFU

Command to put the device chip into firmware update mode.

Command	Read	Write
CMD_DFU	N.A.	0x8B

Packet Structure:

0x0A	0x02	0x8B	DATA	CRC	0x0D

This command will reboot the chip to bootloader and move the state to firmware update mode. The chip will be broadcasting BLE backets with battery part number to uniquely identify the chip.

The chip responds back to this command with an SCSS in case of success and FAIL in case of failure.

4.2.12 CMD_MAC

Command to fetch the MAD address of the BLE chip.

Command	Read	Write
CMD_MAC	0x0C	N.A.

Packet structure:

0x0A	0x02	0x0C	DATA	CRC	0x0D

The response (SCSS) to this command will contain the BLE MAC address. FAIL will be sent in case of failure.

4.2.13 CMD_BEACON_CRC

Beacon data maintains a separate CRC. This is based on this this field the terminal decides whether to update the BLE chip configuration or not.

Command	Read	Write
CMD_BEACON_CRC	0x0D	N.A.

Packet structure:

0x0A	0x02	0x0D	DATA	CRC	0x0D

The response (SCSS) to this command will contain the beacon CRC. FAIL will be sent in case of failure.

4.2.14 CMD_BOOTLOADER

This command fetches the bootloader version from the chip.

Command	Read	Write
CMD_BOOTLOADER	0x0E	N.A.

Packet Structure:

0x0A	0x02	0x0E	DATA	CRC	0x0D

The response (SCSS) to this command will contain the bootloader version. FAIL will be sent in case of failure.

4.2.15 CMD_BEACON_MODE

Command to define the beaconing behavior on battery reinsertion to a turned off device

Command	Read	Write
CMD_BEACON_MODE	N.A.	0x8F

Packet Structure:

0x0A	0x03	0x8F	DATA (1 Byte)	CRC	0x0D

The data field can have following values:

Functionality	Value
Enable beaconing on reinsert	0x00 (default)
Disable beaconing on reinsert	0x01

The chip responds back to this command with an SCSS/FAIL packets in case of success/failure.

1.1.1 CMD_BATTERY_ID

Command to set the battery ID in the chip’s config area. The chip broadcasts this ID when in the DFU mode to uniquely identify the battery.

Command	Read	Write
CMD_BATTERY_ID	N.A.	0x9A

Packet Structure:

0x0A	0x16	0x9A	DATA (20 Bytes)	CRC	0x0D

Data field contains the 20 bytes battery ID.

1.2 SCSS packet structure

The BLE firmware responds with and SCSS (success) packet if the processing of the incoming I2C command from the host terminal is successful. The packet will have following structure:

STX	LEN	SCSS	CMD	DATA (if any)	CRC	ETX

Packet field	Description	Value
STX	Start of packet	0x0A
LEN	Length of the packet excluding STX and ETX	
SCSS	Success command	0x71
CMD	The command that is acknowledged	
DATA (if any)	Data if any	
CRC	Calculated CRC	
ETX	End of packet	0x0D

1.3 FAIL packet structure

In case of failure in processing an incoming I2C command the BLE chip responds with a FAIL (failure) packet. This packet has following structure:

STX	LEN	FAIL	CMD	ERROR	CRC	ETX

Packet field	Description	Value
STX	Start of packet	0x0A
LEN	Length of the packet excluding STX and ETX	
FAIL	Failure command	0x72
CMD	The command that is failed	
ERROR	Error code	
CRC	Calculated CRC	
ETX	End of packet	0x0D

The ERRPR code can be one of the following:

- ERR_CRC = -1
- ERR_INVALID_CMD = -2
- ERR_INVALID_DATA = -3
- ERR_INVALID_PKT = -4
- ERR_TIMEOUT = -5
- ERR_EXECUTION = -6
- ERR_MEMORY_ALLOC = -7

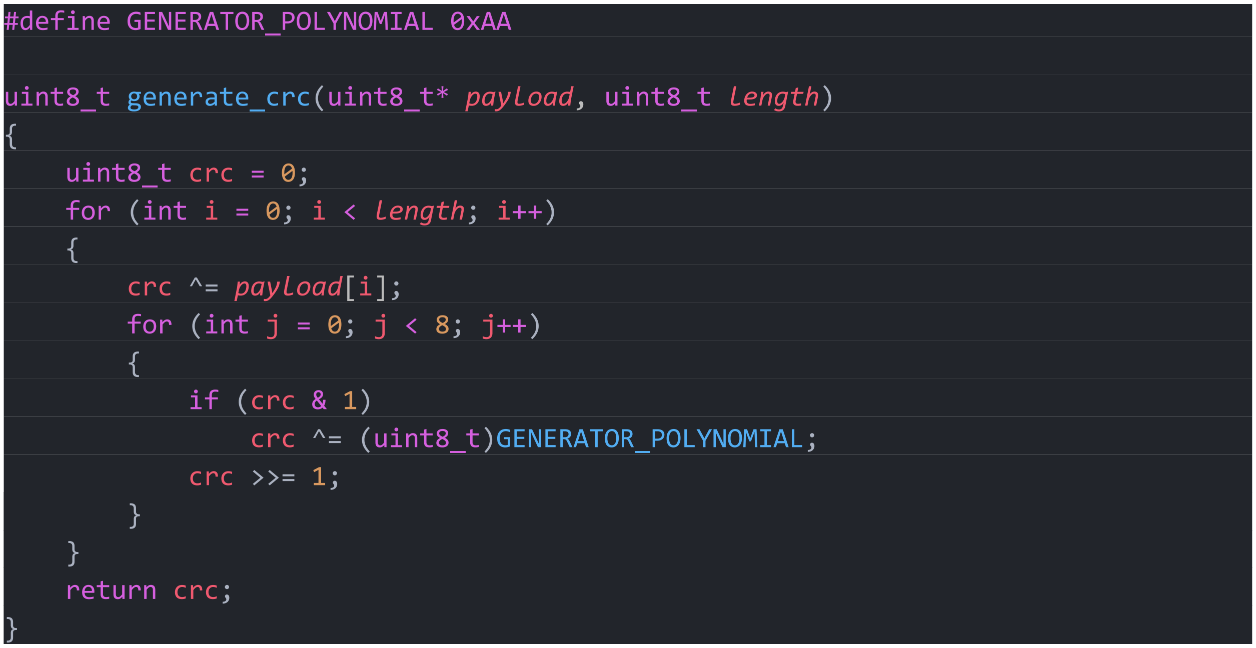
1.4 CRC Calculation

To make sure the integrity of the packet, each packet that is transmitted and received by the BLE chip has one byte CRC.

The firmware is using polynomial division method to calculate the CRC of the payload. STX and ETX bytes are excluded from the calculation.

The generator polynomial used in the firmware (hence at the terminal) is: $x^7 + x^5 + x^3 + x$. The corresponding hex notation is 0xAA.

Below code snippet does the CRC calculation:



```c
#define GENERATOR_POLYNOMIAL 0xAA

uint8_t generate_crc(uint8_t* payload, uint8_t length)
{
    uint8_t crc = 0;
    for (int i = 0; i < length; i++)
    {
        crc ^= payload[i];
        for (int j = 0; j < 8; j++)
        {
            if (crc & 1)
                crc ^= (uint8_t)GENERATOR_POLYNOMIAL;
            crc >>= 1;
        }
    }
    return crc;
}
```