

Hardware Test API Baliz

API List for Firmware Hardware Test

22/06/2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Written by** | **Validated by** | **Confidential** |
| *19/04/2019* | *Yoann Rebischung* |  | *EXOTIC SYSTEMS* |
| **Date** | **Revisions** | | |
| *19/04/2019* | *A* |  | *File Creation* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Table of Contents 2](#_Toc6571424)

[Termes et abbreviations 3](#_Toc6571425)

[Reference documents 3](#_Toc6571426)

[1. Introduction 4](#_Toc6571427)

[2. Frame Format 5](#_Toc6571428)

[3. Command list 6](#_Toc6571429)

[**Command Type MAC** 6](#_Toc6571430)

[**Command Type I2C** 6](#_Toc6571431)

[**Command Type SPI** 6](#_Toc6571432)

[**Command Type ISS** 6](#_Toc6571433)

[**Command Type RSS** 6](#_Toc6571434)

[**Command Type LED** 7](#_Toc6571435)

[**Command Type BUZ** 7](#_Toc6571436)

[**Command Type SFC** 7](#_Toc6571437)

[**Command Type GPS** 8](#_Toc6571438)

[**Command Type BLE** 8](#_Toc6571439)

[**Command Type INT** 8](#_Toc6571440)

[**Command Type LPM** 8](#_Toc6571441)

# Termes et abbreviations

|  |  |
| --- | --- |
| Abbreviations | Significations |
| *BLE* | *Bluetooth Low Energy* |
| *MCU* | *Microcontroller* |
| *CRC* | *Cyclic Redundancy Check* |
| *WDT* | *Watch Dog Timer* |
| *EEPROM* | *Electrically-Erasable Programmable Read-Only Memory* |
| *MSB* | *Most Significant Bit* |
| *LSB* | *Less Significant Bit* |
| *nRF* | *nRF52* | |
| *MSM* | *Main State Machine* | |
| *SSM* | *Sub State Machine* | |
| *RTC* | *Real Time Clock/Counter* | |
|  |  |

# Reference documents

|  |  |  |
| --- | --- | --- |
| Référence | Description | Nom |
| *[ES001]* |  |  |
|  |  |  |

# Introduction

The aim of this document is to describe how to use the specific firmware Hardware Test for Baliz Board V2 over the Debugger RTT J-Link.

# Frame Format

Frame is restricted to a maximum of 32 characters.

**Each part of the command is separated by a comma “,” to help decoder process.**

Command format:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Preamble | Cmd Type | 1st Argument | (2nd Argument) | (n Arg) | CR | LF |

|  |  |  |
| --- | --- | --- |
| Field | Length | Description |
| Preamble | 1 character | “$” |
| Cmd Type | 3 characters | “CHK”, “CFG”, … |
| 1st Argument | 3 characters | “I2C”, ”SPI”, ”ISS”, … |
| 2nd / n Arguments | N characters | “BME”, “MAX”, … |
| CR, LF | 2 characters | “\r\n” |

Acknowledge format:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Preamble | Ack Type | Previous Cmd | Delimiter | Result | Help | CR | LF |

|  |  |  |
| --- | --- | --- |
| Field | Length | Description |
| Preamble | 1 character | “$” |
| Ack Type | 3 characters | “ACK” |
| Previous Cmd | 3 to 6 characters + 1 for comma | “I2C”, ”ISS,BME”, … |
| Delimiter | 1 character | “+” |
| Result | 1 character | “0”, “1” |
| Help(optional) | 3 characters | For example, if RSS is called before an ISS, “ISS” is added to the result. |
| CR, LF | 2 characters | “\r\n” |

Result format:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Preamble | Rslt Type | Previous Cmd | Delimiter | Result | Data Field | CR | LF |

|  |  |  |
| --- | --- | --- |
| Field | Length | Description |
| Preamble | 1 character | “$” |
| Rslt Type | 3 characters | “RSL” |
| Previous Cmd | 3 to 6 characters + 1 for comma | “I2C”, ”ISS,BME”, … |
| Delimiter | 1 character | “+” |
| Result | 1 character | “0”, “1”, “?” |
| Data Field(optional) | n characters | Depends on command. |
| CR, LF | 2 characters | “\r\n” |

No checksum is added to any of the frame to simplify the process.

# Command list

### **Command Type MAC**

Command to get Bluetooth MAC Address of the device.

Example:

*$CHK,MAC<CR><LF>*

*$ACK,MAC+1<CR><LF>*

*$RSL,MAC+1,D5:20:AB:49:EF:47<CR><LF>*

### **Command Type I2C**

Command to initialize the hardware I2C layer of the device.

Example:

*$CHK,I2C<CR><LF>*

*$ACK,I2C+1<CR><LF>*

*$RSL,I2C+1<CR><LF>*

### **Command Type SPI**

Command to initialize the hardware SPI layer of the device.

Example:

*$CHK,SPI<CR><LF>*

*$ACK,SPI+1<CR><LF>*

*$RSL,SPI+1<CR><LF>*

### **Command Type ISS**

Command to initialize the Sensors on the device.

Possible Arguments:

* BME for Bosch BME280 sensor.
* MAX for Maxim MAX44009 sensor.
* LIS for ST LIS2MDL sensor.
* LSM for ST LSM6DSL sensor.
* LTC for Linear LTC2943 sensor.
* ADX for analog ADXL632 sensor.

Example:

*$CHK,ISS,BME<CR><LF>*

*$ACK,ISS,BME+1<CR><LF>*

*$RSL,ISS,BME+1<CR><LF>*

Or

*$RSL,ISS,BME+0,I2C<CR><LF>*

### **Command Type RSS**

Command to read the Sensors of the device.

Possible arguments are the same as ISS command Type.

Example:

*$CHK,RSS,BME<CR><LF>*

*$ACK,RSS,BME+1<CR><LF>*

*$RSL,RSS,BME+1,25.50,983.45,63.23<CR><LF>*

Or

*$RSL,RSS,BME+0,ISS<CR><LF>*

### **Command Type LED**

Command to process a blink of all LED of the device.

Since the device cannot confirm the good execution of the command the result is always “?”.

Example:

*$CHK,LED<CR><LF>*

*$ACK, LED+1<CR><LF>*

*$RSL,LED+?<CR><LF>*

### **Command Type BUZ**

Command to process a buzzer.

As LED Command, since the device cannot confirm the good execution of the command the result is always “?”.

Example:

*$CHK,BUZ<CR><LF>*

*$ACK,BUZ+1<CR><LF>*

*$RSL,BUZ+?<CR><LF>*

**Command Type CSF**

Command to initiate a communication over UART with the SigFox component on the device.

In case of success, the result is always Device PAC then Device ID.

Example:

*$CHK,CSF<CR><LF>*

*$ACK,CSF+1<CR><LF>*

*$RSL,CSF+1,FDE07CC5F3CC56A7,0030D2F4<CR><LF>*

### **Command Type SFC**

Command to generate an RF clock wave corresponding to the SigFox Frequency 868.0MHz.

This command is a toggle.

Example:

*$CHK,CSF<CR><LF>*

*$ACK,CSF+1<CR><LF>*

Result on start:

*$RSL,CSF+?<CR><LF>*

Result on stop

*$RSL,CSF+1<CR><LF>*

### **Command Type GPS**

Command to initiate a communication over UART with the GPS component on the device.

The result of this command is the output of the GPS so all the NMEA frames.

Example:

*$CHK,GPS<CR><LF>*

*$ACK,GPS+1<CR><LF>*

*$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,\*47<CR><LF>*

### **Command Type BLE**

Command to generate an RF clock wave corresponding to the Bluetooth Frequency 2.4GHz.

This command is a toggle.

Example:

*$CHK,BLE<CR><LF>*

*$ACK,BLE+1<CR><LF>*

Result on start:

*$RSL,BLE+?<CR><LF>*

Result on stop

$RSL,BLE+1<CR><LF>

### **Command Type INT**

Command to check interruption electronic line of the different sensors on the device.

### **Command Type LPM**

Command to put the device in the deepest power consumption mode.

To test this mode properly, send the command and close the RTT Viewer and disconnect the JLINK.

Example:

*$CHK,LPM<CR><LF>*

*$ACK,LPM+1<CR><LF>*

*$RSL,LPM+1<CR><LF>*

### **Command Type CFG**

Command to configure sensors and get data.

* **MAX17205 (gauge)**:

$CHK,I2C*<CR><LF>* /\* Init I2C first \*/

$ACK,I2C+1*<CR><LF>*

$RSL,I2C+1*<CR><LF>*

$CHK,CFG,MAX,c*<CR><LF>* /\* Set context \*/

$ACK,CFG+1*<CR><LF>*

$RSL,CFG,MAX+1*<CR><LF>*

$CHK,CFG,MAX,I*<CR><LF>* /\* Init max17205 \*/

$ACK,CFG+1*<CR><LF>*

$RSL,CFG,MAX+1*<CR><LF>*

$CHK,CFG,MAX,V*<CR><LF>* /\* Get voltage in mV (cell1,cell2,cell3,cell4,cellX,Vbatt \*/

$ACK,CFG+1*<CR><LF>*

$RSL,CFG,MAX+1,3523,3571,1320,0,0,8380*<CR><LF>*

$RSL,CFG,MAX+1*<CR><LF>*

$CHK,CFG,MAX,I*<CR><LF>* /\* Get current in µA \*/

$ACK,CFG+1*<CR><LF>*

$RSL,CFG,MAX+1,1633*<CR><LF>*

$RSL,CFG,MAX+1*<CR><LF>*

### **Command Type NFC**

Command to test the NFC (ST25DV sensor. The user has 10 seconds to approach an NFC reader close to the tag and generate the interrupt. The first 0/1 corresponds to the ST25DV test result and the second to the interruption test.

Example:

$CHK,I2C  
$ACK,I2C+1  
$RSL,I2C+1  
$CHK,NFC*<CR><LF>*  
$ACK,NFC+1*<CR><LF>*  
$RSL,NFC+1,1*<CR><LF>*