|  |
| --- |
| **MFE** |

**Software Requirements Specification (SRS)**

For the

**MFE Software**

Version: 1.9

|  |
| --- |
| **RESTRICTION OF USE, DUPLICATION, OR**  **DISCLOSURE OF PROPRIETARY INFORMATION**  This document contains proprietary information that is the sole property of DSIT. The document is submitted to the recipient for his use only. By receiving this document, the recipient undertakes not to duplicate the document or to disclose in part, or in whole any of the information contained herein to any third party without receiving beforehand written permission from DSIT. |

**Table of Contents**

[1 Changes 3](#_Toc489789705)

[2 Applicable Documents: 3](#_Toc489789706)

[3 Introduction 4](#_Toc489789712)

[3.1 The Unit 4](#_Toc489789713)

[3.2 MCU connection 5](#_Toc489789714)

[4 Objective 6](#_Toc489789716)

[5 Communication Channel 7](#_Toc489789717)

[5.1 Electrical Specification 7](#_Toc489789718)

[5.2 Discrete Lines 7](#_Toc489789719)

[5.3 Host Communication Protocol 8](#_Toc489789720)

[5.3.1 Packet Structure 9](#_Toc489789721)

[5.3.2 ACK Packet Structure 9](#_Toc489789722)

[5.4 Messages 9](#_Toc489789723)

[5.4.1 Control Message 10](#_Toc489789724)

[5.4.2 Raw Status Request 12](#_Toc489789725)

[5.4.3 Momentary Status Request 15](#_Toc489789726)

[5.4.4 Version Request 16](#_Toc489789728)

[5.4.5 Change Mode Request 17](#_Toc489789729)

[5.4.6 Set Calibration table 18](#_Toc489789730)

[5.4.7 Get Calibration table 21](#_Toc489789731)

[6 Software Update Session 22](#_Toc489789732)

[6.1 Software Update Messages 23](#_Toc489789733)

[6.1.1 Version Request 24](#_Toc489789734)

[6.1.2 Set Data Line 25](#_Toc489789735)

[6.1.3 Get Data Line 26](#_Toc489789736)

[6.1.4 Finish Update Process 27](#_Toc489789737)

[7 Hardware 28](#_Toc489789738)

[7.1 Schematic 28](#_Toc489789739)

[7.2 Microcontroller 29](#_Toc489789740)

[7.3 GPIO and Analog Pin Assignments 29](#_Toc489789741)

# Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Change | Version | Date |
| Tamir Blum | First Edition. | 1.0 | 10/12/2016 |
| Asaf Elias | Fixed after customer review. | 1.1 | 14/01/2017 |
| Asaf Elias | Changed the communication channel to RS422. | 1.2 | 24/01/2017 |
| Asaf Elias | Added a SN to the version request, added a Pre-Amp ADC input. | 1.3 | 01/02/2017 |
| Asaf Elias | Added the switch mode request, changed the calibration table, changed the bootloader commands, added a GPIO table and added some logic definitions in the text. | 1.4 | 23/04/2017 |
| Asaf Elias | Added PreAmp and ISense multipliers to the calibration table. | 1.5 | 01/05/2017 |
| Asaf Elias | Detailed the logic of the analog readings. Added frequency state to the status message. | 1.6 | 03/05/2017 |
| Asaf Elias | Removed the option of the software update to update itself. | 1.7 | 04/05/2017 |
| Asaf Elias | Change mode command now has payload, change the software update logic regarding PROG\_EN (removed the boot parameter) | 1.8 | 16/05/2017 |
| Asaf Elias | Split the status message to 2, Added a timing diagram to the control message and the raw status request. Added a control identifier variable, Changed the serial link speed. | 1.9 | 06/08/2017 |

# Applicable Documents:

|  |  |  |  |
| --- | --- | --- | --- |
| Document Name | File Name | Version | Date |
| MFE מול uNircom של ICD טיוטת | RAFDOCS-#14659290-v1-RAFDOCS-#14617414-v2-uNircom\_-\_MFE\_ICD | V2 | 27.09.2016 |
| Comments for SRS - 15292751 | Comments for SRS - 15292751 | V1 | 09.01.2017 |

# Introduction

This Document Describes the SW operation of the MFE unit. The unit is built from a PA TX Path and two receive Paths. The unit includes an on board MCU that monitors the device parameters and is responsible for communication with an external host. The MCU also controls the TX/RX switching and the diversity switch. The communication channel with the external host is RS422 UART based.

## The Unit



## MCU connection



# Objective

The MFE card is a command driven card that processes command from the remote host, change discrete lines value if needed and return the status of the system when requested.

The MFE controller has several main tasks, these tasks are:

* PA Gain - Control bias of both PA by controlling Gate and VDD Supply through DAC.
* Antenna Select - Select the antenna the signal is transmitting on.
* TX/RX - Select the unit operation mode.

The external host can control the following operations:

* TX \ RX Path select
* Go to Idle mode
* Set PA gain
* Select the TX antenna.

In addition, the external host can request the current value of the following data:

* FWD Power.
* REV Power.
* Input Power.
* Pre-Amp Power.
* Temperature.
* PA Current.
* The current PA Gain.
* The controller mode (as described in the list below).
* The Antenna state.

The MFE card has 4 modes of operations:

* Operational – commands are passed through the RS422 channel and Executed on the next rising edge of TTI\_SYNC LINE.
* Technician – commands are passed through the RS422 channel and executed asynchronously.
* Maintenance – configuration of calibration tables through the RS422 channel.
* Software Update– configuration and validation of the software stored in the flash.

On power-up, the MFE controller loads the Software Update component software. It in turn checks that the MFE operational software is stored correctly on the flash and if it does, loads it and jumps to it.

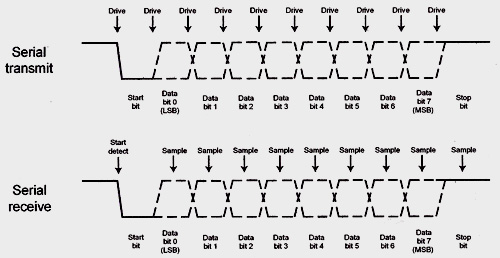
Upon starting from a successful software update session (as part of the power up sequence or after a strobe on the RST discrete line) the MFE card will start at the operational mode. The operational software will start accepting messages form the host TBD us after power up / RST discrete line falling edge.

# Communication Channel

## Electrical Specification

The chosen communication channel with the external host is an RS422 Half Duplex, 921Kb/s UART based communication channel.

A detailed waveform diagram of the UART channel can be seen below:



## Discrete Lines

In addition to the RS422 communication channel several discrete lines can be used to control the unit:

1. TTI\_SYNC – This line is used in operational mode to sync the changes inside the MFE. As it is a synchronization line it is connected as an interrupt to the MCU and will be handled with maximum priority. This line can oscillate at a maximum rate of 1 KHz.
2. PROG\_EN – Allows the Software Update component to start in Boot Mode.
3. RST – Reset the card.

A TTI Counter will count the number of times the TTI Sync discrete rising edge was detected by the MFE card and will be used to number the different status responses. On power-up the counter will be reset and the counting will start from 0.

## Host Communication Protocol

The communication protocol between host and microcontroller is a simple master/slave setup. The host always initiates communication by sending its command message. If the microcontroller receives a correct message (framing and CRC are correct), it sends back an ACK replay. If the received message is a request for data from the MFE, the returned data will should be treated as an ACK by the host.

If a packet is received with an error, the unit will not respond and the host should treat the sent packet as lost. The timeout for a lost packet is approx. 10ms.

The packets content is sent on the RS422 channel using an RFC1662 (PPP in HDLC-like Framing) inspired implementation and is protected by an 8bit CRC as required in the RFC.

Packet length is devised from the framing mechanism.

The RFC1662 specification guarantees that the framings char will not appear in the data payload by using an “escape” mechanism, each byte is in the message is compared to the framing character (0x7e) and to the escape character (0x7d) and if it is equal to one of them than the character is ‘escaped’ by inserting a 0x7d in-front of it in the message and then XORing the character with 0x20. As a byproduct of the escape method, the message length can increase by 100% + 2 framing bytes + 1 CRC byte.

The RFC1662 protocol defines the use of a CRC on the packet to ensure that a correct packet has been received. The CRC polynom used in the MFE is x^8 + x^2 + x + 1 and will be implemented via a lookup table (and a XOR calculation per byte).

The host can send up to 2000 messages per second to the MFE.

### Packet Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **OPCODE** |  | **Message Type** |
| **2** | **N Data Bytes** | N >= 0 | **Message Data** |
| **2 + n** | **CRC** |  | **CRC code.** |
| **3 + n** | **0x7E** |  | **Frame End Char** |

### ACK Packet Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x80** |  | **ACK message Opcode** |
| **2** | **CRC** |  | **CRC code.** |
| **3** | **0x7E** |  | **Frame End Char** |

## Messages

This section defines all messages transmitted from the external host and returned by the MFE.

All messages are transmitted as binary data.

Fields that are more than 1 byte long, will be passed in a little-endian format, meaning that the first byte is the least significant and last byte is most significant.

All fields in the messages below that have no explicit size defined, are 8 bit entities.

The following messages can be sent by the host:

* Control message.
* Status Request.
* Version Request.
* Change Mode Request.
* Set Calibration Table.
* Get Calibration Table.

The following messages can be returned by the MFE.

* Ack.
* Status Response.
* Version Response.
* Calibration Data Response.

### Control Message

The control message is the message that allows the external host to control the operation of the MFE.

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x0** |  | **Control Message OPCODE** |
| **2** | **Data** |  | **Control Message data.** |
| **3** | **Identifier** |  | **Control Message Identifier LSB** |
| **4** | **Identifier** |  | **Control Message Identifier MSB** |
| **3** | **CRC** |  | **CRC code.** |
| **4** | **0x7E** |  | **Frame End Char** |

A detailed description of Byte #2 can be seen below:

|  |  |  |
| --- | --- | --- |
| **Bit #** | **Name** | **Description** |
| **0** | **TxOn** | **Sets the MFE in TX mode (TxOn=1) or in RX Mode(TxOn=0)** |
| **1** | **PA Gain0** | **Sets the Gain of the RFFE (see table below)** |
| **2** | **PA Gain1** | **Sets the Gain of the RFFE (see table below)** |
| **3** | **PA Gain2** | **Sets the Gain of the RFFE (see table below)** |
| **4** | **TxAnt** | **Selects the TX antenna, Tx Ant 1 (TxAnt=0) or Tx Ant 2 (TxAnt=1)** |
| **5** | **High / Low** | **Select the operating frequency of the amplifier, (0 – Low, 1 – High)** |
| **6** | **Reset** | **Resets the MFE** |
| **7** | **Parity Bit** | **Binary XOR result of bits 0 to 6** |

PA Gain Values Table:

|  |  |  |
| --- | --- | --- |
| **Value** | **Gain Level** | **Gain (dBm)** |
| **0(000)** | PA Off |  |
| **1(001)** | PA On – Level 1 |  |
| **2(010)** | PA On – Level 2 |  |
| **3(011)** | PA On – Level 3 |  |
| **4(100)-7(111)** | PA On – Level 4 |  |

In operational mode command is synchronized to the next rising edge of the TTI Sync discrete input. Control message at time **t** is executed at time **t+1**.



The 16bit identifier should be stored and returned to the host in the status requests sent in the next time slice (**t+1**).

In technician mode, the command is executed immediately upon reception.

When changing to TX Mode (bit #0 = 1) the change into TX mode occurs after the time value saved in the TX\_ON\_TIMING\_USEC parameter. The return to RX occurs after the time value saved in the TX\_OFF\_TIMING\_USEC parameter.

The state of the TX/RX bit controls the value of both TX\_RX\_SEL and of the pair TX\_RX\_ENV and TX\_RX\_ENVM (which are reversed versions of each other).

When changing the Gain of the PA (the value in the PA-Gain bits are different than the last value used) the change in gain will occurs on the next TTI Sync.

When switching from PA OFF level to a PA ON level the change will occur after the time value saved in the PA\_ON\_TIMING\_USEC parameter. Shutting down the PA will occur after the time value saved in the PA\_OFF\_TIMING\_USEC parameter.

The PA gain bits controls the AD5312 DAC that controls the voltage of the PA. These values are converted using the data in tables 1 & 2 of the calibration tables (which table depends on the current High / Low state) from gain values to mVolts. The converted value is then sent to the AD5312.

When selecting an antenna, the change will occur after the time value saved in the ANT\_SEL\_TIMING\_USEC parameter.

The state of TxAnt controls the value of the pair ANT\_SEL and ANT\_SELN (which are reversed versions of each other).

As mention above, in operational mode the action of an incoming message will run after the next rising edge of the TTI\_SYNC signal (with additional delay as configured in the calibration table).

The maximum time for an action to trigger after the rising edge of TTI\_SYNC is TBS usec.

The MFE should respond to this message with an ACK response.

### Raw Status Request

The host can request the MFE to sample several analog values and return them to the host.

These analog values are sampled with accordance with incoming TTI\_SYNC signals.

The message for a status request is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x1** |  | **Raw Status Request** |
| **3** | **CRC** |  | **CRC code.** |
| **4** | **0x7E** |  | **Frame End Char** |

Below is the response form the MFE for a raw status request:

|  |  |  |  |
| --- | --- | --- | --- |
| Byte # | Value | Parameters | Details |
| 0 | **0x7E** |  | Frame Start Char |
| 1 | **0x81** |  | Status Response |
| 2 | **TTI Counter** |  | LSB of the TTI counter |
| 3 | **TTI Counter** |  | MSB of the TTI counter |
| 4 | **Control Identifier** |  | LSB of Control Identifier |
| 5 | **Control Identifier** |  | MSB of Control Identifier |
| 6 | **Forward Power** | 1st ADC sample | LSB of FWD Power |
| 7 | **Forward Power** |  | MSB of FWD Power |
| 8 | **Forward Power** | 2nd ADC sample | LSB of FWD Power |
| 9 | **Forward Power** |  | MSB of FWD Power |
| … | **…** | … | … |
|  | **Forward Power** | 4th ADC sample | LSB of FWD Power |
|  | **Forward Power** |  | MSB of FWD Power |
|  | **Reverse Power** | 1st ADC sample | LSB of REV Power |
|  | **Reverse Power** |  | MSB of REV Power |
|  | **Reverse Power** | 2nd ADC sample | LSB of REV Power |
|  | **Reverse Power** |  | MSB of REV Power |
| … | **…** | … | … |
|  | **Reverse Power** | 4th ADC sample | LSB of REV Power |
|  | **Reverse Power** |  | MSB of REV Power |
|  | **Input Power** | 1st ADC sample | LSB of In Power |
|  | **Input Power** |  | MSB of In Power |
|  | **Input Power** | 2nd ADC sample | LSB of In Power |
|  | **Input Power** |  | MSB of In Power |
| … | **…** | … | … |
|  | **Input Power** | 4th ADC sample | LSB of In Power |
|  | **Input Power** |  | MSB of In Power |
|  | **Pre-Amp Power** | 1st ADC sample | LSB of In Power |
|  | **Pre-Amp Power** |  | MSB of In Power |
|  | **Pre-Amp Power** | 2nd ADC sample | LSB of In Power |
|  | **Pre-Amp Power** |  | MSB of In Power |
| … | **…** | … | … |
|  | **Pre-Amp Power** | 4th ADC sample | LSB of In Power |
|  | **Pre-Amp Power** |  | MSB of In Power |
|  | **CRC** |  | CRC code. |
|  | 0x7E |  | Frame End Char |

The list of returned raw status ranges can be seen below:

|  |  |  |
| --- | --- | --- |
| **Status ID** | **Data Range** | **Description** |
| **Forward Power** | **40.0 – 20.0** | Unsigned 16 bit value |
| **Reverse Power** | **40.0 – 0.0** | Unsigned 16 bit value |
| **Input Power** | **-10.0 - +15.0** | Signed 16 bit value |
| **Pre-Amp Power** | **-40.0 – 20.0** | Signed 16 bit value |

All values are sent with a 1/100 precision with a direct conversion to the specific data type.

In operational mode, the ADC readings that are returned in the raw status message are sampled after receiving a TTI\_SYNC signal. Data sampled at time **t** is sent to the host in raw status request that arrive in **t+1**.



The control identifier of this specific time slice should be sent to back to the host in this response (in time **t+1** the value of the identifier sent in time **t**). If no new control messages arrived than the value should be of the last stored control message.

In Technician mode, the ADC reading are sampled and sent back to the host right after receiving the message from the host.

Each sample is sampled 4 times, where each sample is sampled after the time configured in the calibration table as described below:

|  |  |  |
| --- | --- | --- |
| **ADC Reading** | **Description** | **Parameter In Use** |
| Forward Power | The time to wait before sampling the FWD input after switching to TX and between samples. | FWD\_SAMP\_TIMING\_USEC |
| Reverse Power | The time to wait before sampling the REV input after switching to TX and between samples. | REV\_SAMP\_TIMING\_USEC |
| Input Power | The time to wait before sampling the INP\_PWR input after switching to TX and between samples. | INP\_PWR\_SAMP\_TIMING\_USEC |
| Pre-Amp Power | The time to wait before sampling the INP\_PWR input after switching to TX and between samples. | PRE\_AMP\_SAMP\_TIMING\_USEC |

The ADC samples are sent to the host as raw data and are not been processed by the MFE.

### Momentary Status Request

The host can request the MFE to sample several analog values and return them to the host.

The message for a status request is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x9** |  | **Momentary Status Request** |
| **3** | **CRC** |  | **CRC code.** |
| **4** | **0x7E** |  | **Frame End Char** |

Below is the response form the MFE for a Momentary status request:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x89** |  | **Status Response** |
| **2** | **TTI Counter** |  | **LSB of the TTI counter** |
| **3** | **TTI Counter** |  | **MSB of the TTI counter** |
| **4** | **Temperature** |  | **LSB of Temp** |
| **5** | **Temperature** |  | **MSB of Temp** |
| **6** | **Power Amplifier Current** |  | **LSB of PA Current (the addition of both PA1\_ISENSE and PA2\_ISENSE)** |
| **7** | **Power Amplifier Current** |  | **MSB of PA Current (the addition of both PA1\_ISENSE and PA2\_ISENSE)** |
| **8** | **Power Amplifier Gain** |  | **Current PA Gain** |
| **9** | **Mode** |  | **Current controller mode** |
| **10** | **Antenna** |  | **Current antenna state** |
| **11** | **High / Low** |  | **Current frequency state** |
| **12** | **CRC** |  | **CRC code.** |
| **13** | **0x7E** |  | **Frame End Char** |

The list of returned status ranges can be seen below:

|  |  |  |
| --- | --- | --- |
| **Status ID** | **Data Range** | **Description** |
| **Temperature** | **-40.0 – 100.0** | Signed 16 bit value |
| **Power Amplifier Current** | **0.10 – 0.80** | Unsigned 16 bit value |

All values are sent with a 1/100 precision with a direct conversion to the specific data type.

ADC samples in the momentary status response are sampled right after a momentary status request is received from the host (regardless of the current controller mode).

### Version Request

The host can request the MFE to send its version string and version date.

The message to request the version string is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x2** |  | **Version Request** |
| **2** | **CRC** |  | **CRC code.** |
| **3** | **0x7E** |  | **Frame End Char** |

Below is the response form the MFE for a version request:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x82** |  | **Version Response** |
| **2** | **Version Day** | **0 – 31** |  |
| **3** | **Version Month** | **0 – 12** |  |
| **4** | **Version Year LSB** |  | **LSB of the year** |
| **5** | **Version Year MSB** |  | **MSB of the year** |
| **6** | **Version Major** | **0 – 255** |  |
| **7** | **Version Minor** | **0 - 255** |  |
| **8** | **Serial Number LSB** | From calibration parameter | **LSB of the card serial number.** |
| **9** | **Serial Number MSB** | **MSB of the card serial number.** |
| **10** | **CRC** |  | **CRC code.** |
| **11** | **0x7E** |  | **Frame End Char** |

### Change Mode Request

The host can request the MFE to change its operating mode. Different modes of operation enables the MFE to ignore the TTI Sync discrete line and set the calibration table values.

The message for a change mode request is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Value** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x3** |  | **Change Mode Request** |
| **2** | **Mode** | 0-2 | **The Requested Mode** |
| **3** | **CRC** |  | **CRC code.** |
| **4** | **0x7E** |  | **Frame End Char** |

Below is the list of available operating modes:

|  |  |  |
| --- | --- | --- |
| **Mode** | **Mode Value** | **Description** |
| Operational | **0** | **In this mode all operations are synced to the rising edge of the TTI Sync discrete.** |
| Technician | **1** | **In this mode all operations are executed when received from host, without waiting for the next TTI Sync discrete strobe.** |
| Maintenance | **2** | **In this mode the host can set and read the calibration table.** |

The MFE should respond to this message with an ACK response.

### Set Calibration table

The MFE has several calibration parameters that are used to store calibration data and timing data for use in the MFE operation. Each parameter is 16 bits long and have a value for low band and for high band. Up to 31 parameters are available per calibration table.

The command to set the calibration table is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x4** |  | **Set Calibration Message** |
| **2** | **Table ID** | **0-2** | **Table ID, 0 – General, 1 – PA Low Band,  2 – PA High Band.** |
| **4** | **Parameter 0** |  | **MSB of Parameter #0** |
| **5** | **Parameter 0** |  | **LSB of Parameter #0** |
| **..** | **..** | **..** | **..** |
| **68** | **Parameter 31** |  | **MSB of Parameter #31** |
| **69** | **Parameter 31** |  | **LSB of Parameter #31** |
| **70** | **CRC** |  | **CRC code.** |
| **71** | **0x7E** |  | **Frame End Char** |

The response from the MFE for a Set calibration message is an ACK message.

The list of calibration parameters in Table 0 are listed below:

|  |  |  |
| --- | --- | --- |
| Parameter # | Parameter ID | Description |
| 0 | CAL\_TBL\_VERSION | The version of the calibration data saved, this allows the user to have several versions of the calibration data. |
| 1 | SERIAL\_NUM | The serial number of this specific MFE. |
| 2 | TEMP\_MULT | The returned TEMP value is multiplied by this constant and then divided by 1000. |
| 3 | FWD\_MULT | The returned TEMP value is multiplied by this constant and then divided by 1000. |
| 4 | REV\_MULT | The returned REV value is multiplied by this constant and then divided by 1000. |
| 5 | INP\_PWR\_MULT | The returned INP\_PWR value is multiplied by this constant and then divided by 1000. |
| 6 | PWR\_CURRENT\_MULT | The returned PWR\_CURRENT value is multiplied by this constant and then divided by 1000. |
| 7 | PRE\_AMP\_MULT | The returned Pre-Amp Power value is multiplied by this constant and then divided by 1000. |
| 8 | ISENSE\_PA1\_MULT | The returned PA1\_ISENSE value is multiplied by this constant and then divided by 1000. |
| 9 | ISENSE\_PA2\_MULT | The returned PA2\_ISENSE value is multiplied by this constant and then divided by 1000. |
| 10 | TX\_ON\_TIMING\_USEC | The time to wait before changing setting TX to on. |
| 11 | TX\_OFF\_TIMING\_USEC | The time to wait before changing setting TX to off (RX). |
| 12 | PA\_ON\_TIMING\_USEC | The time to wait before changing the PA gain value to something other than PA\_OFF. |
| 13 | PA\_OFF\_TIMING\_USEC | The time to wait before changing the PA gain value to PA\_OFF. |
| 14 | ANT\_SEL\_TIMING\_USEC | The time to wait before selecting another antenna. |
| 15 | FWD\_SAMP\_TIMING\_USEC | The time to wait before sampling the FWD input after switching to TX. |
| 16 | REV\_SAMP\_TIMING\_USEC | The time to wait before sampling the REV input after switching to TX. |
| 17 | INP\_PWR\_SAMP\_TIMING\_USEC | The time to wait before sampling the INP\_PWR input after switching to TX. |
| 18 | PRE\_AMP\_SAMP\_TIMING\_USEC | The time to wait before sampling the PRE\_AMP input after switching to TX. |

Note. Unused parameters are ignored by the MFE.

The list of calibration parameters in Table 1 and 2 are listed below:

|  |  |  |
| --- | --- | --- |
| Cell # | Parameter | Description |
| 0 | PA Gain Val 0 for DAC output A. | The value to set in DAC A when PA Gain equals 0 |
| 1 | PA Gain Val 1 for DAC output A. | The value to set in DAC A when PA Gain equals 1 |
| .. | **..** | .. |
| 15 | PA Gain Val 15 for DAC output A. | The value to set in DAC A when PA Gain equals 15 |
| 16 | PA Gain Val 0 for DAC output B. | The value to set in DAC B when PA Gain equals 0 |
| 17 | PA Gain Val 1 for DAC output B. | The value to set in DAC B when PA Gain equals 1 |
| .. |  |  |
| 31 | PA Gain Val 15 for DAC output B. | The value to set in DAC B when PA Gain equals 15 |

Note. Currently only 4 gain values are used per table.

### Get Calibration table

The MFE has several calibration parameters that are used to store calibration data and timing data for use in the MFE operation. Each parameter is 16 bits long and have a value for low band and for high band.

The command to get the calibration parameters is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x5** |  | **Set Calibration Message** |
| **2** | **Table ID** | **0-2** | **Table ID, 0 – General, 1 – PA Low Band, 2 – PA High Band.** |
| **3** | **CRC** |  | **CRC code.** |
| **4** | **0x7E** |  | **Frame End Char** |

The response from the MFE for a Set calibration message is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x85** |  | **Get Calibration Response** |
| **2** | **Table ID** | **0-2** | **Table ID, 0 – General, 1 – PA Low Band,  2 – PA High Band.** |
| **4** | **Parameter 0** |  | **MSB of Parameter #0** |
| **5** | **Parameter 0** |  | **LSB of Parameter #0** |
| **..** | **..** | **..** | **..** |
| **68** | **Parameter 31** |  | **MSB of Parameter #31** |
| **69** | **Parameter 31** |  | **LSB of Parameter #31** |
| **70** | **CRC** |  | **CRC code.** |
| **71** | **0x7E** |  | **Frame End Char** |

# Software Update Session

The MFE software update component is a piece of software that is responsible for checking the integrity of the operational software, loading it and running it.

The software update component is also responsible for getting a new software image from the host, verify its integrity and saving it to internal flash.

To start the software update process, a power up should be performed with the PROG\_EN discrete line set to ‘0’. When PROG\_EN equals ‘0’ the software update process does not load the operational software from flash but instead waits for configuration data from the host.

The software update process will sample the PROG\_EN discrete line for approx. TBD milliseconds. If while sampling the PROG\_EN discrete its value does not equal ‘0’ the software update process will load the operational software (if it exists) and jump to it.

As in the operational application, the communication channel is an RS422 Half Duplex, 1Mb/s communication channel as described in section 5.1.

Also, the software update component also uses the RFFC1662 framing as described in section 5.3 and the message / response packet structure is the same.

To verify that the software update component has indeed loaded correctly, the host should request the version string from the MFE. The Version string form the MFE software update component is in the form of B.x where B denotes that this is a software update component version.

When starting a new flash programming operation, the software update process will start by erasing the flash sections affected by the data received, this step is destructive and data saved on that sector will be lost.

The software update component cannot update itself, in case that the software update component needs to be updated a physical access to the MFE will be needed (a PICit3 or similar debugger will be needed to program the MFE controller via ICSP).

Calibration parameters are not programmed via the software update process, and will not be effected by a software upgrade.

## Software Update Messages

As stated above, the software update component uses the same communication channel, framing and message format as the operational software, this allows the host to use the same codebase for communicating with the MFE software update component .

The MFE software update component can receive the following messages:

* Version Request.
* Set Data Line.
* Get Data Line.
* Finished Update Process.

The MFE will respond with the following responses:

* Version Response.
* Data Line Status.
* Data Line Data.
* Ack.

### Version Request

The Version response of the software update component is identical to the response packet of the operational software, the returned version string is in the form of B.x where B denoted that this is a software update component version and x denotes the bootloader version.

The serial number component of the version response should be set to 0.

### Set Data Line

The host sends to the MFE a full sector of data to be saved in the flash memory. The message consist of 32 bits of destination address and 64 bytes of data.

The command to set a data line is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x6** |  | **Set Data Line Message** |
| **2** | **Address** |  | **Bits 0-8 of the destination address** |
| **3** | **Address** |  | **Bits 9-15 of the destination address** |
| **4** | **Address** |  | **Bits 16-23 of the destination address** |
| **5** | **Address** |  | **Bits 24-31 of the destination address** |
| **6** | **Data 0** |  |  |
| **..** | **..** | .. |  |
| **66** | **Data 63** |  |  |
| **67** | **Data 64** |  |  |
| **68** | **CRC** |  | **CRC code.** |
| **69** | **0x7E** |  | **Frame End Char** |

The data section has a fixed length of 64 bytes.

The response from the MFE for a Set Line message is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x86** |  | **Set Data Line Response** |
| **2** | **Address** |  | **Bits 0-8 of the destination address** |
| **3** | **Address** |  | **Bits 9-15 of the destination address** |
| **4** | **Address** |  | **Bits 16-23 of the destination address** |
| **5** | **Address** |  | **Bits 24-31 of the destination address** |
| **6** | **Status** |  | **Status of the programmed line, 0 for OK, 1 for Error in CS.** |
| **7** | **CRC** |  | **CRC code.** |
| **8** | **0x7E** |  | **Frame End Char** |

### Get Data Line

The host can request the MFE to return the data at a specific address in the flash address space, the request for a specific line is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x7** |  | **Set Data Line Message** |
| **2** | **Address** |  | **Bits 0-8 of the destination address** |
| **3** | **Address** |  | **Bits 9-15 of the destination address** |
| **4** | **Address** |  | **Bits 16-23 of the destination address** |
| **5** | **Address** |  | **Bits 24-31 of the destination address** |
| **6** | **CRC** |  | **CRC code.** |
| **7** | **0x7E** |  | **Frame End Char** |

The response from the MFE for a Get Data line is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x87** |  | **Get Data Line Response** |
| **2** | **Address** |  | **Bits 0-8 of the destination address** |
| **3** | **Address** |  | **Bits 9-15 of the destination address** |
| **4** | **Address** |  | **Bits 16-23 of the destination address** |
| **5** | **Address** |  | **Bits 24-31 of the destination address** |
| **6** | **Data 0** |  |  |
| **..** | **..** | .. |  |
| **66** | **Data 63** |  |  |
| **67** | **Data 64** |  |  |
| **68** | **CRC** |  | **CRC code.** |
| **69** | **0x7E** |  | **Frame End Char** |

The data section has a fixed length of 64 bytes.

### Finish Update Process

When the host completes the update process it sends the next request to the MFE.

This request signals the MFE that the process is complete and that it on next boot the updated software should run.

The command to finish the update process is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte #** | **Code** | **Parameters** | **Details** |
| **0** | **0x7E** |  | **Frame Start Char** |
| **1** | **0x8** |  | **Set Data Line Message** |
| **2** | **CRC** |  | **CRC code.** |
| **3** | **0x7E** |  | **Frame End Char** |

The MFE should respond to this message with an ACK response.

# Hardware

## Schematic



## Microcontroller

|  |  |
| --- | --- |
| **Recommended PIC** | **PIC18F45K22** |
| **Operating voltage** | **3.3V** |
| **Inputs (TTL / converter)** | **TBD** |
| **Outputs (TTL / converter)** | **TBD** |
| **POR** | **Available** |
| **Internal clock** | **8MHz and up to 64Mhz** |

* Flash Memory
  + The PIC microcontroller has 32Kbytes of internal flash memory, this memory will be used for both the software update component software and for the operational software.
* RAM Memory
  + The PIC microcontroller has 1536 bytes of internal RAM, the software will use this memory for its stack & heap.

## GPIO and Analog Pin Assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Name in document** | **PIN Name** | **Net Name** | **Type** |
| **TTI Counter** | **RB0** | **TTI\_EXT** | **Input Discrete** |
| **Forward Power** | **AN5** | **FFWR** | **Input Analog** |
| **Reverse Power** | **AN5** | **RREV** | **Input Analog** |
| **Temperature** | **AN6** | **TMP** | **Input Analog** |
| **Input Power** | **AN7** | **RF\_INDET** | **Input Analog** |
| **Power Amplifier Current** | **AN8 AN9** | **PA1\_ISENSE PA2\_ISENSE** | **Input Analog Input Analog** |
| **Pre-Amp Power** | **AN17** | **PreAmp\_det** | **Input Analog** |
| **Antenna Select** | **RD7 RD6** | **ANT\_SEL ANT\_SELN** | **Output Discrete Output Discrete** |
| **Program Enable** | **RB1** | **PROG\_EN** | **Input Discrete** |
| **Main Enable** | **RD4** | **MAIN\_EN** | **Output Discrete** |
| **5V Enable** | **RD5** | **5V\_EN** | **Output Discrete** |
| **DAC Data** | **RA4** | **DAC\_DATA** | **Output Discrete** |
| **DAC Sync** | **RA2** | **DAC\_SYNC** | **Output Discrete** |
| **DAC Clock** | **RA3** | **DAC\_CLOCK** | **Output Discrete** |
| **DAC Latch** | **RA1** | **DAC\_LDAC** | **Output Discrete** |
| **TX RX Select** | **RB4** | **TX\_RX\_SELECT** | **Output Discrete** |
| **TX\_RX\_ENV** | **RB5** | **TX\_RX\_ENV** | **Output Discrete** |
| **TX\_RX\_ENVM** | **RD0** | **TX\_RX\_ENVM** | **Output Discrete** |