**Software Requirements Specification (SRS)**

For projects

**1533 (FAST LINK)**

Version: 1.0

|  |
| --- |
| **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](#_Toc506309771)

[2 *Project 1533*: 4](#_Toc506309772)

[2.1 Introduction: 4](#_Toc506309773)

[2.2 System block diagram 4](#_Toc506309774)

[2.3 Objective 5](#_Toc506309775)

[2.4 PLL configurations 7](#_Toc506309776)

[2.5 Errors indications and system failures treatment 8](#_Toc506309777)

[2.6 Synthesizer registers values 9](#_Toc506309778)

[3 Host PC serial control protocol 10](#_Toc506309779)

[3.1 Serial communication configuration 10](#_Toc506309780)

[3.2 Data / control directions 10](#_Toc506309781)

[3.3 Host commands 11](#_Toc506309782)

[4 Software Update Session 13](#_Toc506309783)

[4.1 Software Update Messages 14](#_Toc506309784)

[5 Hardware 15](#_Toc506309785)

[5.1 MCU Schematic 15](#_Toc506309786)

[5.2 Microcontroller 16](#_Toc506309787)

[5.3 GPIO and Analog Pin Assignments 17](#_Toc506309788)

# Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Change | Version | Date |
| Roee Zinoue | First Edition. | 1.0 | 25/01/18 |

# 

# Project 1533:

## Introduction:

This document describes the SW operation of the FAST LINK unit.

The system will consist two RF units:

Transmission unit: will output RF signal with unique pattern that generated by the on board MCU unit.

Receiving unit: Will receive RF signal, if the signal match by his pattern to what expected the on board MCU will blink a connected LED to indicate about it.

Via serial connection we will able to control state of the transmitting / receiving and transmitting pattern. The system also have connect to MCU LEDs to indicate about the state of the system operation.

## System block diagram



## Objective

Main object of the MCU is to transmit RF signals at particular formula and then capture, measure and collect data about the income RF signals states and export it to Host PC application.

Goals of the Tx channel:

1. The Transmission unit will be insert into BIT mode, at this mode the PIC MCU will configure on-board synthesizer unit to transmit number of RF signals according to ["PLL – configurations"](#_PLL_configurations) section and well-known RF particular formulas.
2. At the BIT mode the RF transmission signals will be transmitted for 2 sec time period.
3. RF signals data rate will be on 9Mbps.
4. When the transmission unit will be insert at BIT mode, ACK message will be send via serial communication to PC host application.
5. The transmission unit after system restart will be able to set at regular work mode (at regular work mode, the MCU work fully functional but no transmitting will occur).
6. If the Transmission unit will be shut down during BIT mode, after restart the unit will continue to work from regular work mode.
7. The transmission unit will also indicate the system user about:

* Advance power.
* If the signal latched by the on board synthesizer.
* Return power.

Goals of the Rx channel:

1. The receiving unit will capture RF signals according to ["PLL – configurations"](#_PLL_configurations) section at selected time window (that will be set via PC host).
2. The receiving will collect the number of captured RF signals and will transfer them after capture time window to PC host application.
3. Transfer Rx channel to stand-by mode (at this mode the system will not listen to any income RF signals).
4. The receiving unit will be able to measure the power of the input RF signal and receive that data to PC host Application (this data will be used to tune the power amplifiers by system user commands based on serial communication as a response).
5. Change working frequency mode of the frequency RF signal.
6. Turn on and off the RSSI sensor.

MCU goals:

1. The MCU will always check the serial communication state (even at BIT mode) and will indicate the system user about any failures related via on – board LEDs.
2. To communicate with PC host application using serial communication (Based on RS422), send and receive data.
3. The on board MCU unit also will start to collect samples data from on board ADC unit immediate after system initialize.
4. Those samples will be written to MCU internal flash and will output them to user via serial channel at user request.
5. To indicate the system user via on board LEDs about several system operation state.

## PLL configurations

The on board MCU need to configure on board PLL-Synthesizers to the following configurations:

**Tx signals**:

PLL parameters:

PLL type: ADF4113

RF reference to PLL = 10 MHz

PRESCALER: 64

Frequencies steps: 10 kHz

Registers values:

prog = 0b110010010000000010010011 = 0xC90093

prog = 0b000000000000111110100000 = 0x000FA0

|  |  |  |
| --- | --- | --- |
| Tx signals | | |
| Description | Range | Step size |
| – Low | 5.75 – 5.85 GHz | 1 MHz |
| – High | 5.15 – 5.25 GHz | 1 MHz |
|  | | |
| PLL – receive signal (Low) | 2.575-2.625 GHz | 0.5 MHz |
| PLL – receive signal (High) | 2.8625-2.9125 GHz | 0.5 MHz |

**Rx signals:**

PLL parameters:

PLL type: ADF4350

RF reference to PLL = 10 MHz

PRESCALER: 8

Frequencies steps: 10 kHz

IF FREQ=480MHz

Registers values:

R1 = 0x08008321 R2 = 0x18005EC2 R3 = 0x000004B3 R4 = 0x0085003C

R5= 0x 00580005

|  |  |  |
| --- | --- | --- |
| Rx signals | | |
| Description | Range | Step size |
| – Low | 5.15-5.25 GHz | 1 MHz |
| – High | 5.75-5.85 GHz | 1 MHz |
|  | | |
| PLL – receive signal (Low) | 2.335-2.385 GHz | 0.5 MHz |
| PLL – receive signal (High) | 2.6225-2.6725 GHz | 0.5 MHz |

## Errors indications and system failures treatment

The system have two connected LEDs that indicate about the following system states:

**Green led**:

This led indicate that certain system operation is set correctly. It will indicate about

1. System initialize: the system initialize ok and ready to operate at stable state if the following tests will be pass:

* Serial communication test: the on board MCU will send to host "start frame" and suppose to receive ACK.
* The Flash on the MCU free space not reach to quarter of his limit size.
* Connection to CPLD unit test: the on board MCU will send to the on board CPLD unit unique frame and suppose to receive ACK.

If all off the above tests will pass ok the green led will blink 3 times.

1. System operation – Advance power: Advanced power have received in the system transmitter. Green led will blink for 5 times.
2. System operation – synthesizer letch: the transmit synthesizer latched the configuration signals ok. Green led will blink for 7 times.
3. System operation – return power: the transmitter receive return power. The green led will blink for 9 times.
4. Rx signals latch – the Rx synthesizer receive RF signal that match the pattern of the sanded RF signal. Green led will blink for 11 times.

**Red led:**

This led indicate that certain system have failure on one of the requested operation. It will indicate about:

1. Serial communication failure: The MCU sent to host "start frame" and not receive ACK for 30 seconds. Red led will blink 2 times.
2. Flash memory failure: The Flash on the MCU free space reached to quarter or more of his limit size – at this case the system will not store data at the flash until user will select to empty the flash space throw host serial application. Red led will blink 4 times.
3. Connection to CPLD unit test failure: The MCU sent to CPLD unit "start frame" and not receive ACK for 30 seconds. Red led will blink 6 times.
4. Synthesizer latch failure: If the on board MCU don’t success to latch the selected frequency, the MCU will try to configure and send the digital words more 2 times. If after 3 times the frequency still not latched the red led will blink 8 times.

## Synthesizer registers values

Note: Please check Analog devices datasheet to learn about the way of frequency in the sensitizer unit.

In order to configure the registers of both ADF4113 and ADF4350 we use Analog devices simulator (Int N-PLL software).

# Host PC serial control protocol

The on board MCU unit will be controller by serial communication channel (based on RS-422) and via hyper terminal application. The following table will configure that commands that the host can send to the MCU.

## Serial communication configuration

* 1. Baud – rate: 115200 bps.
  2. 8 bit data size.
  3. 1 start bit, 1 stop bit.
  4. No CTS, RTS.

## Data / control directions

For both Rx and Tx the flow data / control direction can be describe as:



Frame description:

Each sending request command will begin with $ char delimiter and end with <cr> (carriage return).

On each request the MCU will send ACK response command that start with

"OK <cr>$" and end with carriage return.

## Host commands

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Command | Example |
| Tx commands | | | |
| 1 | Init TX system: init TX unit system, this command include carrier frequency that the system start to work with  (Frequency will set in steps of 1 MHz and in allowed range). | $TI XXXX <cr> Frequency in MHz XXXX | $TI 1230 <cr>  Init Tx unit and set frequancy of 1230 MHz |
| 2 | Frequency change: set frequency of the TX unit (in steps of 1 MHz and in allowed range). | $TF XXXX <cr>  XXXX = Frequency in MHz | $TF 1234 <cr>  Set TX unit frequancy at 1234 MHz. |
| 3 | Start / stop transmit: insert / stop TX unit to work at full operation mode. | $TX 1 / 0 < cr>  1: ON  0: OFF  Default: 0 | $TX 1 <cr>  Set TX unit on. |
| 4 | Get status: Get status from TX unit:   * Synth (PLL)t. Lock indication. * Input power status. * FREQ=XXXX (value in MHz). * REV POWER =±XX (value in dB). * FWD POWER =+XX (value in dB). * TEMP=XX.X (value in Celsius degree). * TRANSMIT= ON/OFF. * LIGHT=ON/OFF. * UNIT ID=XXX. * UNIT DC=YYXX. * SW VERSION=<string>. * Input power status. * UNIT TYPE=X (type T or R). * BIT = ON/OFF.   Each data value will be separate from each other with comma (","). | $TQ <cr> | When we receive:  1, 1, 4230, N32, 22, 38.2, 1, 1, 123, 1234, S4321, 0, 1, 1.  It will indicate that the system set on:   * Synth (PLL)= lock * input power status on * FREQ= 4230 MHZ * REV POWER=-32 dB. * FWD POWER=22 dBm * TEMP=38.2 c * TRANMSIT=ON * LIGHT=ON * UNIT ID=123 * UNIT DC=1234 * SW VERSION=S4321 * INPUT POWER=OFF * UNIT TYPE=T * BIT=ON |
| 5 | BIT: start / stop status bit | $TB 1 / 0 < cr>  1: ON  0: OFF  Default: 0 | $TB 1 <cr> |
| 6 | Power range: select between high power and low power. | $TP 1 / 0 <cr>  1: HIGH POWER  0: LOW POWER  Default: 0 | $TP 1 |
| # | Description | Command | Example |
| RX commands | | | |
| 1 | Init RX system: init RX unit system, this command include carrier frequency that the system start to work with.  (Frequency will set in steps of 1 MHz and in allowed range). | $RI XXXX <cr>  XXXX = Frequency in MHz | $RI 1234 <cr> |
| 2 | Start / stop receive: insert / stop RX unit to work at full operation mode. | $RX 1 / 0 < cr>  1: ON  0: OFF  Default: 0 | $RX 1 <cr> |
| 3 | Get status: Get status from TX unit:   * Synt (PLL). Lock indication. * Device temperature= XX.X (value in Celsius degree) * Input power status (ON, OFF) * FREQ=XXXX (value in MHz). * RSSI=± XX (value in dBm). * TEMP=XX.X (value in Celsius degree). * LIGHT=ON/OFF. * UNIT ID=XXX. * UNIT DC=YYXX. * SW VERSION=<string>. * UNIT TYPE=X (type T or R). * Compression point= XX (Value in dbm) | $RQ<cr> | When we receive:  1, 38.2, 1, 1230, N32, 45.2, 1, 123, 1234, 000A, 0, 23  It will indicate that the system set on:   * Synth (PLL): lock. * Device temp: 38.2 c. * Input power: on. * FREQ: 1230 (MHz). * RSSI: -32 dBm. * TEMP: 45.2 c. * LIGHT: IN * UNIT ID: 123 * UNIT DC: 1234 * SW VERSION: 10. * UNIT TYPE: R. * Compression point: 23 dBm. |
| 4 | Frequency: set the RX unit frequency (in steps of 1 MHz and in the RX frequencies allowed range). | $RF XXXX <cr>  XXXX = Frequency in MHz | $RF 1230 <cr> will set RX to work at 1230 MHz. |
| 5 | Compression point: The command will set the point in -30 dbm or 0dbm | $RC 1 / 0 <cr>  1: -30 dBm.  0: 0 dBm.  Default: 0 | $RC 1  Set RX Compression point to work at -30 dBm. |

# 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.

# Hardware

## MCU Schematic



## Microcontroller

|  |  |
| --- | --- |
| **Recommended PIC** | **PIC18F45K22 – 8 bit core** |
| **Operating voltage** | **3.3V** |
| **Inputs (TTL / converter)** | **TBD** |
| **Outputs (TTL / converter)** | **TBD** |
| **POR** | **Available** |
| **Internal clock** | **8MHz and up to 64Mhz** |
| **Pin count** | **40** |

* 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.
* Peripherals support:
  + Connectivity: 2-UART, 2-SPI, 2-I2C2-MSSP(SPI/I2C).
  + ADC: 28 ch, 10-bit.

## 

## GPIO and Analog Pin Assignments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MCU NET | I/O | SCHEMATIC NET | TX FUNCTION | RX FUNCTION | EXPLAIN |
| RC7/RX1 | I | RS422RX | RS422\_RX | RS422\_RX | RS422 Control RX |
| RD4/AN24/SDO2 |  |  | NC | NC |  |
| RD5/AN25 |  | RS422EN | NC | NC |  |
| RD6/AN26/TX2 | O | EN | POWER\_EN | POWER\_EN |  |
| RD7/AN27/RX2 | I | N16785247 | STBY\_IN | STBY\_IN | When HI unit goes into STBy ( EN=0) . If Low the according to Software word) |
| Vss1 |  | GND\_POWER |  |  |  |
| Vdd1 |  | MCU\_3p3V |  |  |  |
| RB0/INT0/AN12 | O | PWR\_CNTRL | PA\_ON |  | operating when negative power supply is ok and when software control used |
| RB1/INT1/AN10 |  | N16949169 | NC | NC |  |
| RB2/INT2/AN8 |  | N16920411 | NC | NC |  |
| RB3/CCP2/AN9 | O | LED\_LIGHT\_EN | MET\_EN | MET\_EN | when Hi enables EXTERNAL METER BACKLIGHT |
| NC1 |  |  |  |  |  |
| NC2 |  |  |  |  |  |
| RB4/AN11 | O | LED\_S2 | LED\_S2 | LED\_S2 | Blinks when not locked , Lights up when PLL locked |
| RB5/AN13/CCP3 | O | LED\_S1 | LED\_S1 | LED\_S1 | IN TX : on in full power . Blinks in low power mode , IN RX : Blinks when below RSSI THRESHHOLD, Lights above |
| RB6/PGC |  | PGC | PGC | PGC |  |
| RB7/PGD |  | PGD | PGD | PGD |  |
| M\C\L\R\/RE3 |  | MCLR | MCLR | MCLR |  |
| RA0/AN0 |  | N16785239 | NC | NC |  |
| RA1/AN1 | O | DAC\_LDAC | DAC\_LDAC | DAC\_LDAC | Controls DAC AD5312ARMZ for : |
| RA2/AN2 | O | DAC\_SYNC | DAC\_SYNC | DAC\_SYNC | TX: PORT A : analog voltage according to measured output power |
| RA3/AN3 | O | DAC\_CLK | DAC\_CLK | DAC\_CLK | TX: PORT B: Set negative voltage according to software control . HI /LO only |
| RA4/C1OUT | O | DAC\_DATA | DAC\_DATA | DAC\_DATA | RX: PORT A: analog voltage according to measured RSSI , outputs 1,11,111,1111, PORT B: Not used |
| RA5/AN4/C2OUT | IAN | RREV | RREV | RSSI | TX: Analog input detecting REV power for reading through software .RX : RSSI |
| RE0/AN5 | IAN | FFWR | FFWR | FFWR | TX : Analog input for FWD power to show on meter and in software.RX: not used |
| RE1/AN6 | IAN | TMP | TMP | TMP | Reads Temperature parameters , reads in status . |
| RE2/AN7 | IAN | N16785439 | VG\_MONITOR | NC | Monitors voltage , if Above ?Thereshhold , PA\_ON is off |
| Vdd2 |  | MCU\_3p3V |  |  |  |
| Vss2 |  | GND\_POWER |  |  |  |
| OSC1/RA7 |  | OSC1 | OSC1 | OSC1 | 8MHZ EXTERNAL CRYSTAL |
| OSC2/RA6 |  | OSC2 | OSC2 | OSC2 |  |
| RC0 | I | SYNTH\_LD | SYNTH\_LD | SYNTH\_LD | IN TX: ADF 4113 , IN RX : ADF4350 ( Need to decrease 480MHz IF) |
| NC3 |  |  |  |  |  |
| NC4 |  |  |  |  |  |
| RC1/CCP2 | O | SYNTH\_LE | SYNTH\_LE | SYNTH\_LE |  |
| RC2/CCP1/AN14 | O | SYNTH\_CLK | SYNTH\_CLK | SYNTH\_CLK |  |
| RC3/SCL/SCL1/SCK1 | O | SYNTH\_DATA | SYNTH\_DATA | SYNTH\_DATA |  |
| RD0/AN20/SCL2/SCK2 |  |  |  |  |  |
| RD1/AN21/CCP4/SDA2/SDI2 |  | N1678585842 |  |  |  |
| RD2/AN22 | I | N16785806 | HILO |  | TX ONLY, When "1" the tx power is set to low . When "0" Controlled by software |
| RD3/AN23 |  |  |  |  |  |
| RC4/SDA1/SDI1/AN16 |  |  |  |  |  |
| RC5/AN17/SDO1 |  | N1678585824 |  |  |  |
| RC6/TX1/AN18 | I | RS422TX | RS422\_TX | RS422\_TX | RS422 Control TX |