Software Requirements Specification (SRS)

For projects

1531 (KU FE Module)

Version: 1.0

B.A. Microwaves LTD.

10 Marconi St. P.O.B. 25482, Haifa 3125401

Tel. +972-4-8202715 Fax. +972-4-8202716

bamw@bamicrowaves.co.il www.bamicrowaves.co.il

Table of Contents

1 Changes

Name	Change	Version	Date
Roee Zinoue	First Edition.	1.0	07/01/18

2 Project 1531

2.1 Introduction

This document describes the SW operation of the KU FE module.

This module will output range of frequencies according to digital data that given on a serial communication channel (based on UART-422).

The on board MCU will also configure an on board synthesizer (ADF-5355) unit to output RF signals at range of:

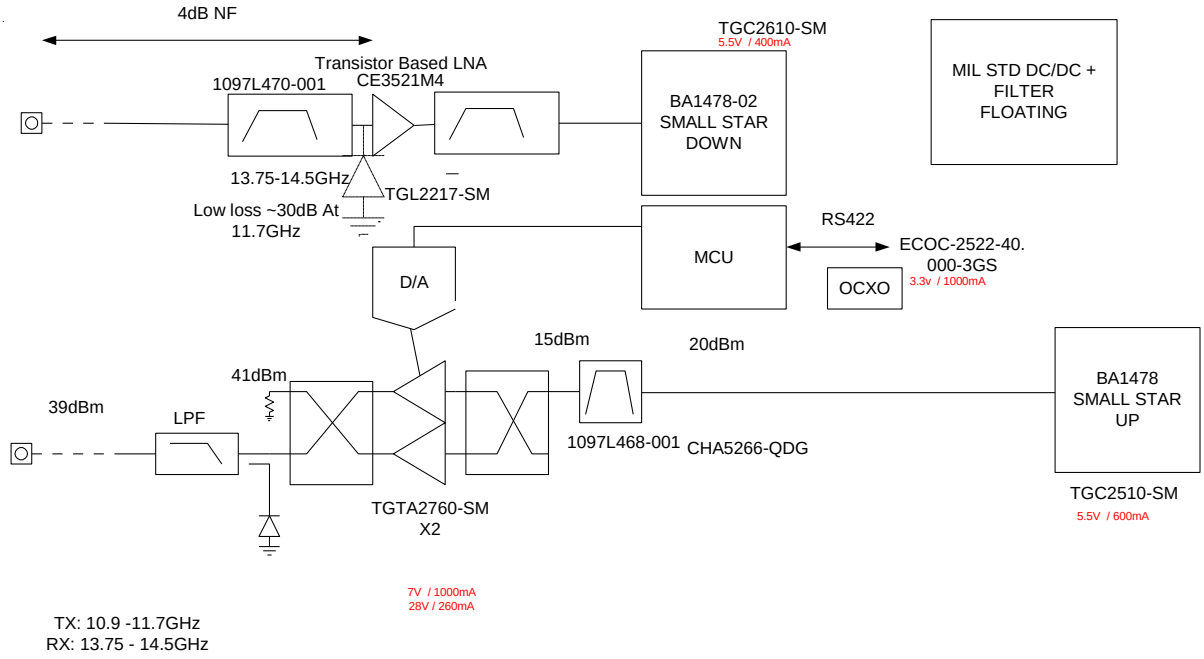
a. From 950 – 2620 MHz to 10.95 – 11.7 GHz. (Tx RF signal).
b. From 13.75 – 14.5 GHz to 950 – 2620 MHz. (Rx RF signal).

The MCU will also control the on board synthesizer by calibrated clock. The calibration will be made via on board CPLD unit and external DAC (PLL circuit).

The module will also be able to sample 6 analog inputs: Temperature, reverse / forward / real voltage and current and stored the samples info on internal FLASH memory section at the MCU (PIC18F45K22) unit.

At a user request those ADC samples (that stored on the internal MCU flash memory) can be read via serial communication at external PC application (among other parameters like: SW version, SW state, synthesizer registers values, RF output signal, etc...).

2.2 System block diagram



2.3 Objective

Main object of the on board MCU (PIC18F45K22):

a. To configure the on board synthesizer (ADF-5355) to exit RF signals at the range of:
    1. Tx RF signal: From 950 – 2620 MHz to 10.95 – 11.7 GHz.
    2. Rx RF signal: From 13.75 – 14.5 GHz to 950 – 2620 MHz.
b. To sample 6 analog inputs signal like: Temperature, reverse / forward / real voltage and current.
c. To store ADC samples at internal flash memory section. And flush them at user selection (this request will be given on the serial channel).
d. To communication via serial communication (RS-422) with external host station (PC) and send / receive data.
e. To indicate the system user on system state / failures via external LEDs.

Main object of the on board CPLD unit:

a. To calibrate digital clock that is an input to the on board synthesizer unit (ADF-5355) via VCO circuit using external DAC.

2.4 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

a. 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.

Red led:

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

a. Serial communication failure: The MCU sent to host "start frame" and not receive ACK for 30 seconds. Red led will blink 2 times.
b. 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.
c. 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.
d. 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.

2.5 Synthesizer registers values

The registers values that configure the RF output signals are calculated on the on board MCU at each system run. Please refer to document appendix A to see how they are calculated.

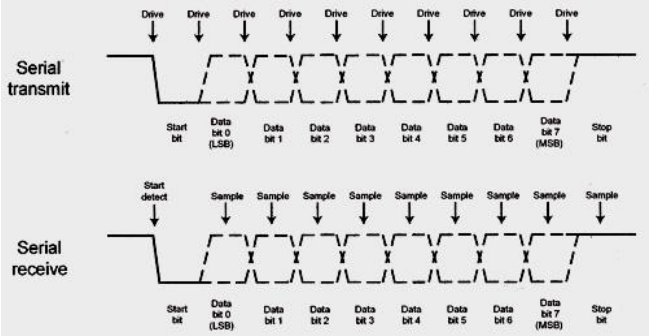
3 EUSART

3.1 Electrical Specification

The chosen communication channel with the external host is an RS422 with configuration:

a. Half Duplex – Tx / Rx channels.
b. Baud-rate of: 115200 Kb/s.
c. 8 bits, 1 start bit, 2 stop bit.
d. Hardware control, No CTS, RTS.
e. Voltage range: -7 to +7 VDC.

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



3.2 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 a data from the MCU unit, 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 and 10 continues lost packets.

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 MCU unit.

3.2.1 Packet Structure

Description about packet structure: Each request from host to MCU will get feedback (ACK) message response.

3.2.1.1 From host to MCU – request packet

Byte #	Value	Details
0 – 1	0xA5 0x5A	Frame Start Chars.
2	MSG_GROUP	Message group.
3	MSG_REQ	Message request type.
N >= 0		Number of data bytes.
D = N		Data.
4 – 5	CRC	CRC-16 code.

Each request consist 6 fixed bytes + N bytes of data in single request frame.

3.2.1.2 From MCU to host – response packet

Byte #	Value	Details
0 – 1	0xA5 0x5A	Frame Start Chars.
2	MSG_GROUP	Message group.
3	MSG_RESP	Message response type.
N >= 0		Number of data bytes.
D = N		Data.
4 – 5	CRC	CRC-16 code.

Each response consist 6 fixed bytes + N bytes of data in single request frame.

3.3 Messages

This section defines all messages groups transmitted from the external host to the MCU.
All messages are transmitted as hexadecimal data bytes.
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.

3.3.1 Messages request groups – from host to on board MCU

Group ID	Group name	Short group information
0x01	Control	Host set the MCU operation parameters
0x02	Status and version	Host get the MCU operation status and run-time statistics.
0x03	ADC configuration	Host set the MCU ADC configuration parameters.
0x04	Synthesizer (Up / Down) configuration	Host get or set the Synthesizer up registers configuration parameters.
0x05	Flash memory configuration	Host get or set the internal MCU flash memory.
0x06	DAC control	Host set the external CPLD DAC configuration.

3.3.2 Control Message

Group ID: 0x01

The control message group allows the external host to control the operation of the MCU.
System obligations:

a. MCU to host baud rate speed always set at 57600 bps.

Request packet info (MSG_REQ):

Byte MSG_REQ value	Name	Description
0x01	MCU reset	Reset the MCU.
0x02	CPLD reset	Reset the CPLD.
0x03	P.A 1 Enable	Enable the P.A 1.
0x04	P.A 1 set power	Host set power of the P.A 1 (range: 1 – 15).
0x05	P.A 2 Enable	Enable the P.A 2.
0x06	P.A 2 set power	Host set power of the P.A 2 (range: 1 – 15).
0x07	Test LEDs	Insert the 2 connected Leds to blink mode test (the Leds will blink for 10 sec at 1 sec toggle time interval).

Response packet info (MSG_RESP):

Byte MSG_RESP value	Name	DATA size	Description
0x03	P.A 1 Enable OK	0	ACK packet response bit to indicate that the MCU set this option
0x04	P.A 1 set power OK	0	ACK packet response bit to indicate that the MCU set this option
0x05	P.A 2 Enable OK	0	ACK packet response bit to indicate that the MCU set this option
0x06	P.A 2 set power OK	0	ACK packet response bit to indicate that the MCU set this option
0x07	Test LEDs ok	0	Leds test start indication.

3.3.3 Status Message

Group ID: 0x02

The status message group allows the external host to read statistic from the on board mcu such as run time and the on board MCU FW version.

Request packet info (MSG_REQ):

Byte MSG_REQ value	Name	Description
0x01	Get run time (mins).	Request from the on board MCU the run time (power on) in time format (HH:MM:SS).
0x02	Get MCU FW version	Request from the on board MCU FW version.
0x03	Get CPLD FW version	Request from the on board CPLD FW version.

Response packet info (MSG_RESP):

Byte MSG_RESP value	Name	DATA size (Bytes)	Description
0x01	MCU run time value.	6	MCU run time in format: HH:MM:SS.
0x02	Get MCU FW version	6	On board MCU Firmware version
0x03	Get CPLD FW version	0	On board CPLD Firmware version

3.3.4 ADC configuration Message

Group ID: 0x03

The ADC configuration Message group allows the external host to set the internal on board MCU ADC unit registers configuration values.
Some obligations:

a. All samples rate are equal: Time interval between samples = 30 seconds.
b. The system will start to samples all the connected analog channels (channel 1 to 6) immediately after system boot and initialize.
c. All samples will save to internal MCU flash memories until memory section is full.
d. During the synthesizer receiving only the Pre-Amp Temperature will be sampled.
e. During operation mode we will use default settings for:$V_{referance}$, and in Right justified result format. In technician mode we can select the desire analog channel to be sampled and result format mode.
f. Binary word after conversion of the ADC consist 10 bit.

3.3.4.1 Request packet info (MSG_REQ)

Byte MSG_REQ value	Name	Description
0x01	ADC enable	Enable / disable the internal MCU ADC unit operation (technician mode).
0x02	ADC technician mode	Enable / disable the internal ADC (technician mode).
0x03	Analog Channel sample Select (in technician mode only).	Set ADC to sample single channel (CH: 0 -27) or range of channels (set by host application) in circular mode. (In technician mode).
0x04	Conversion Result Format (in technician mode only).	Select if the ADC conversion results are Right justified or Left justified.

ADC channel selection:

Data Byte value	Channel name
0x01	Temperature.
0x02	Reverse analog voltage.
0x03	Forward analog voltage.
0x04	Input to system analog voltage.
0x05	Current analog voltage.

3.3.4.2 Response packet info (MSG_RESP)

Byte MSG_RESP value	Name	DATA size (Bytes)	Description
0x01	Set ADC enable OK	0	ACK packet response bit to indicate that the MCU set this option.
0x02	Set Conversion Result Format OK	0	ACK packet response bit to indicate that the MCU set this option.

3.3.5 Synthesizer (Up / Down) configuration message

Group ID: 0x04

The Synthesizer (up / down) configuration message group allows the external host to set the configuration registers values of synthesizer (ADF-5355) up / down (Rx / Tx) units.
Please refer to Appendix B at this document to learn how internal frequencies made in synthesizer ADF-5355.

Some obligations:

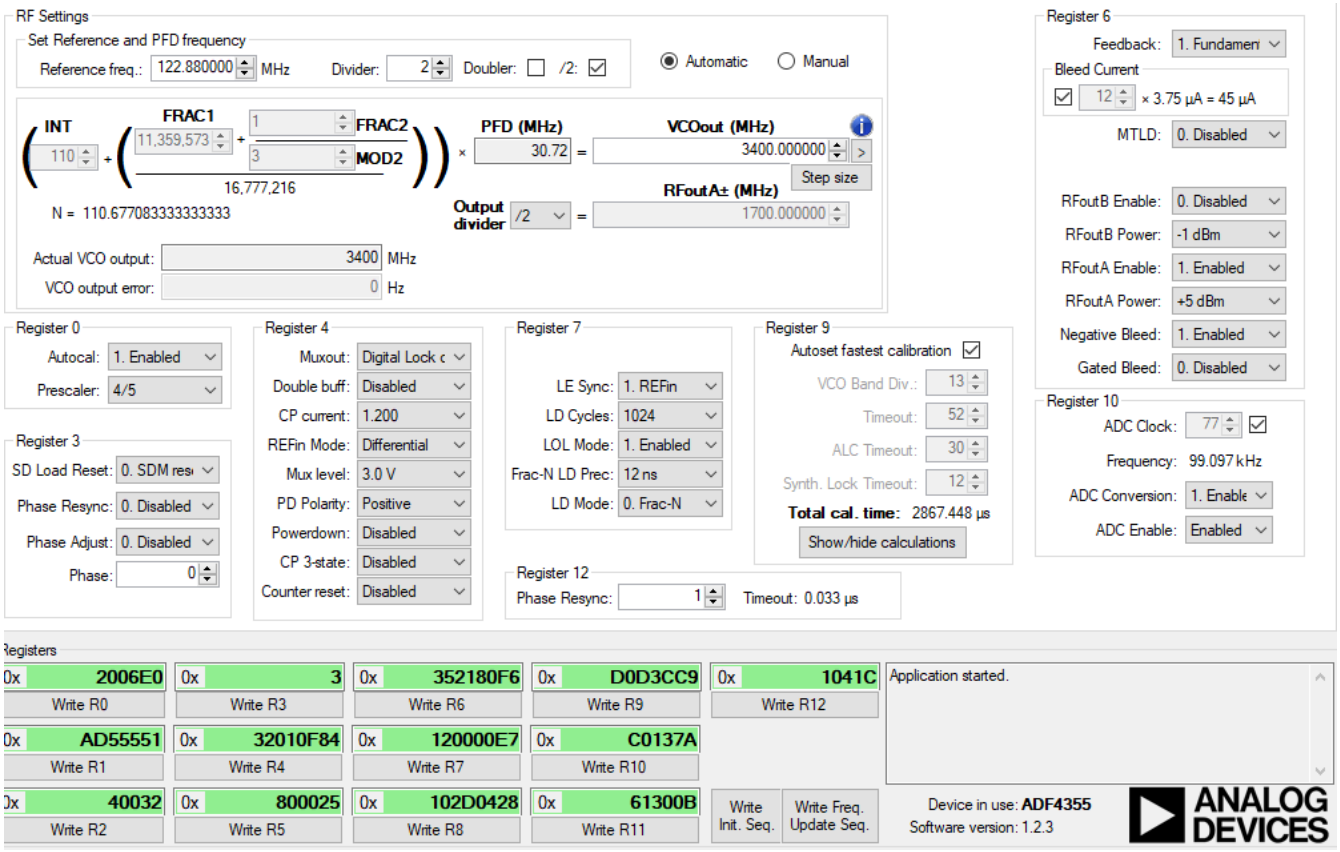
a. The user from host will set RF_{out} and RF_{in}signals parameters, all other synthesizer configuration are on default value (as appear on the picture below).
b. At system boot up, the RF_{out} and RF_{in} signals values will be: $RF_{out} = 1.2\ GHz\ RF_{in} = 1.2\ GHz$.
c. Resolution (steps size) frequency configuration set at 10 kHz.

3.3.5.1 Request packet info (MSG_REQ):

Byte MSG_REQ value	Name	Description
synthesizer down		
0x01	Set synthesizer down RF_{out} frequency value	Host set the $\boldsymbol{RF_{out}}$ value of the down synthesizer down.
0x02	Set synthesizer down F_{if} frequency value	Host set the F_{if} value of the down synthesizer down.
Synthesizer up		
0x01	Set synthesizer down RF_{out} frequency value	Host set the $\boldsymbol{RF_{out}}$ value of the down synthesizer down.
0x02	Set synthesizer down F_{if} frequency value	Host set the F_{if} value of the down synthesizer down.

3.3.5.2 Example of configuration:

$RF_{out} = 1.7\ GHz$
$VCO_{out} = 3.4\ GHz$
$CP\ current = 1.2$



3.3.5.3 Response packet info (MSG_RESP)

Response bit indicate all the request bits at their location.
If they Set: the MCU set the option on ok.
Preset: the MCU fail to set this option.

3.3.6 Flash memory configuration control message

Group ID: 0x05

The Flash memory configuration control message group allows the external host to control operation related to internal flash memory on the on board MCU unit.

3.3.6.1 Request packet info (MSG_REQ)

Byte MSG_REQ value	Name	Description
0x01	Erase flash memory	Host will erase all flash memory pages.
0x02	Check memory conditions	Host will get percentage indication of the free memory in the flash.
0x03	Read raw data from flash	Host will get 4 bytes of raw data at selected address from flash memory.

3.3.6.2 Response packet info (MSG_RESP)

Byte MSG_RESP value	Name	DATA size (Bytes)	Description
0x01	Erase flash memory OK	0	ACK packet response bit to indicate that the MCU set this option.
0x02	Get memory state of the flash	0	Host will get percentage (in steps of 10 to indicate how memory left in the internal MCU flash.
0x03	Get raw data packets from the flash	4	Host Get raw data packets from the flash (MSB byte to LSB byte).

3.3.7 DAC control message

Group ID: 0x06

The DAC control message group allows the external host to set the connected to MCU DAC registers values.

3.3.7.1 Request packet info (MSG_REQ)

Byte MSG_REQ value	Name	Description
0x1	Set ADC in Normal Operation mode	Host set the DAC to operate in normal voltage mode.
0x2	**Set digital value of ADC**	Host set the DAC digital value of the ADC on board unit. (Value size = 10 bits).

3.3.7.2 Response packet info (MSG_RESP)

Byte MSG_RESP value	Name	DATA size (Bytes)	Description
0x1	Set ADC in Normal Operation mode OK	0	ACK packet response bit to indicate that the MCU set this option.
0x2	**Set DAC digital value OK**	**0**	ACK packet response bit to indicate that the MCU set this option.

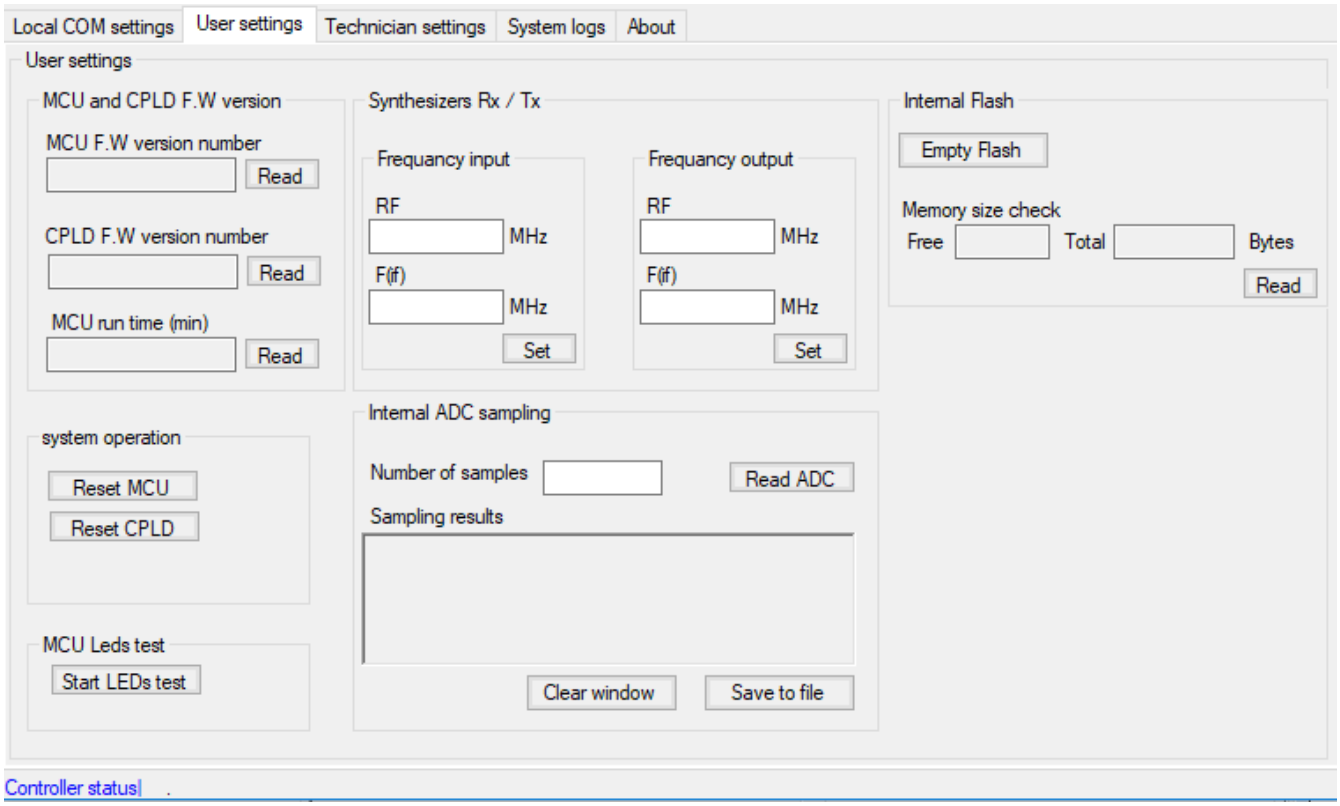
4 Host PC serial application

The user have ability to configure options of the operation mode of the on board MCU and CPLD unit. This doing by two screen: user mode and technician mode (to enter you must set first user name and password).

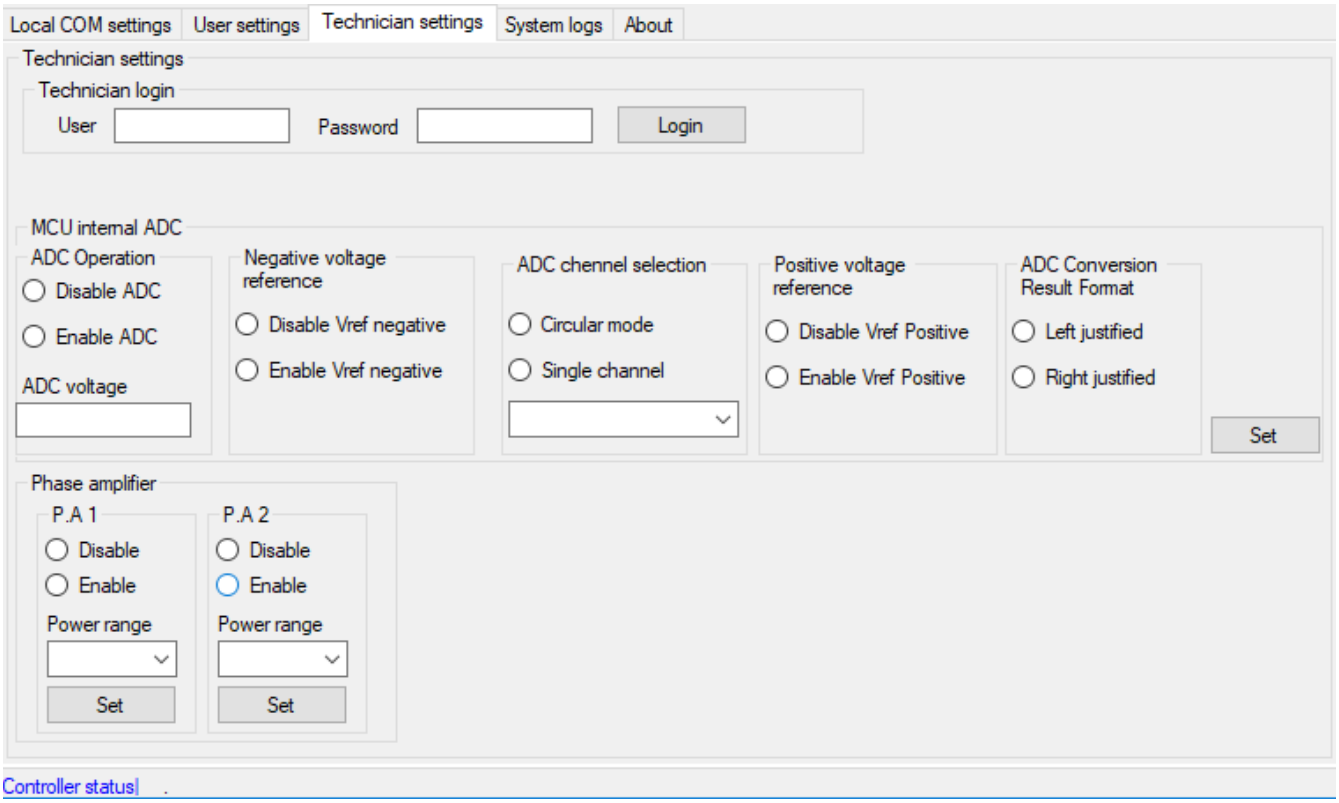
Besides that, the application will:

a. Log application activity to external text file.
b. Load default values or last configuration values when application start.
c. Have status strip to indicate if the request receive and treat by the on board MCU.

4.1 User mode



4.2 Technician mode



5 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.

5.1 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.

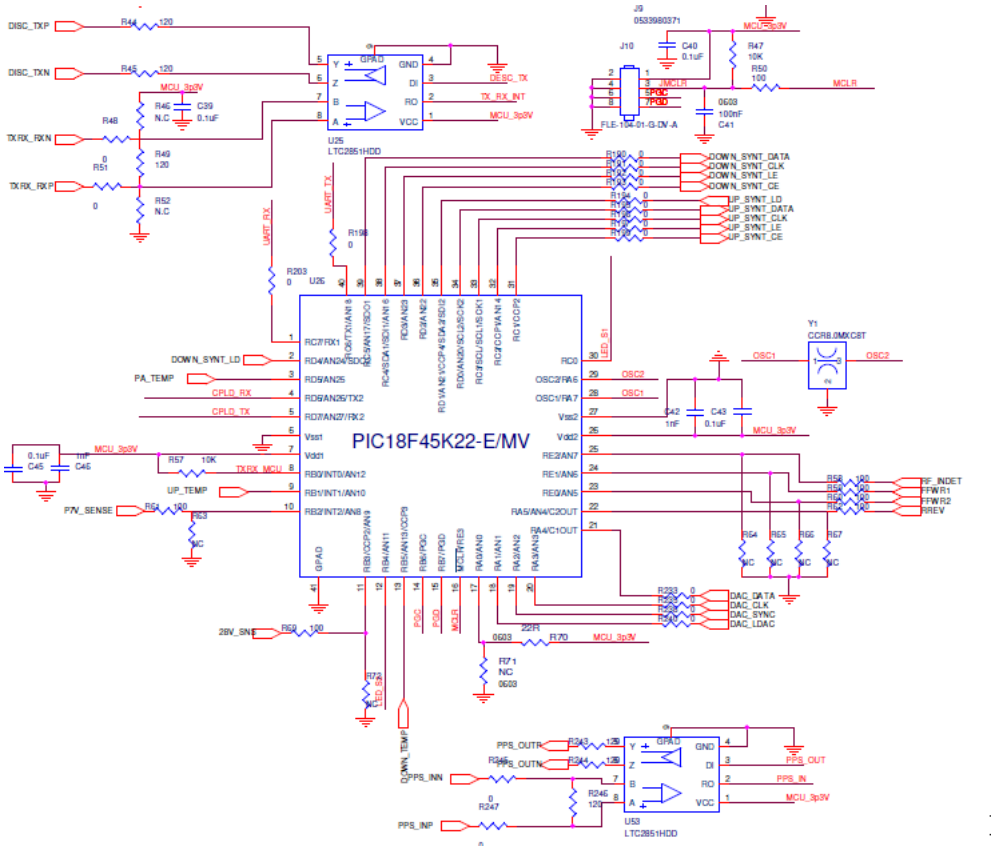
The MFE will respond with the following responses:

- Version Response.
- Data Line Status.
- Data Line Data.
- Ack.

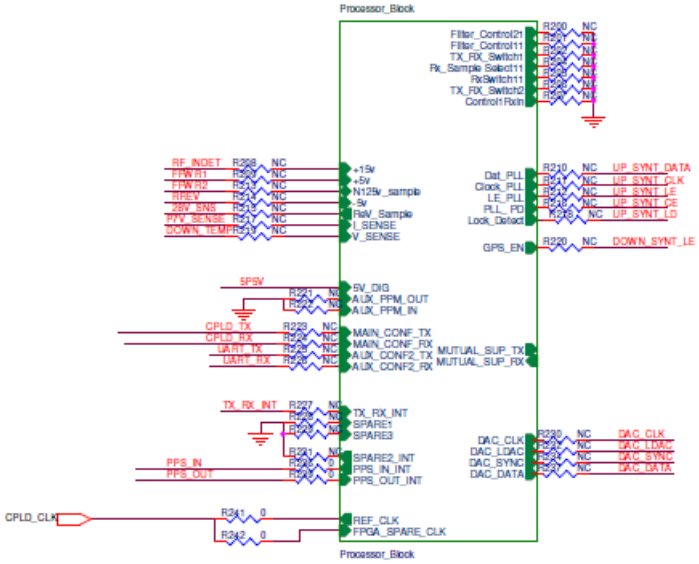
6 Hardware

6.1 Schematic

6.1.1 MCU unit



6.1.2 CPLD unit


Processor_Block
Filter_Control21
Filter_Control11
TX_RX_Switch1
Rx_Sample Select11
RxSwitch11
TX_RX_Switch2
Control1Rxin
R200 NC
RF_INDET R208 NC
FPWR1 R209 NC
FPWR2 R211 NC
RREV R212 NC
28V_SNS R213 NC
P7V_SENSE R214 NC
DOWN_TEMPR R215 NC
+15v
+5v
N125v_sample
-5v
RelV_Sample
I_SENSE
V_SENSE
Dat_PLL
Clock_PLL
LE_PLL
PLL_PD
Lock_Detect
R210 NC UP_SYNT_DATA
R211 NC UP_SYNT_CLK
R212 NC UP_SYNT_LE
R213 NC UP_SYNT_CE
R216 NC UP_SYNT_LD
GPS_EN
R220 NC DOWN_SYNT_LE
5P5V
R221 NC
R224 NC
5V_DIG
AUX_PPM_OUT
AUX_PPM_IN
CPLD_TX R223 NC
CPLD_RX R225 NC
UART_TX R226 NC
UART_RX R228 NC
MAIN_CONF_TX
MAIN_CONF_RX
AUX_CONF2_TX
AUX_CONF2_RX
MUTUAL_SUP_TX
MUTUAL_SUP_RX
TX_RX_INT R227 NC
R229 NC
R231 NC
R234 0
TX_RX_INT
SPARE1
SPARE3
SPARE2_INT
PPS_IN_INT
PPS_OUT_INT
PPS_IN
PPS_OUT
DAC_CLK
DAC_LDAC
DAC_SYNC
DAC_DATA
R230 NC DAC_CLK
R232 NC DAC_LDAC
R233 NC DAC_SYNC
R235 NC DAC_DATA
CPLD_CLK
R241 0
R242 0
REF_CLK
FPGA_SPARE_CLK
Processor_Block

6.2 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.

6.3 GPIO and Analog Pin Assignments

Name in document	PIN Name	Net Name	Type
MCU power, clocks and programing pins:			
3.3 VDC Enable	VDD1	VDD1	Input Analog
MCU_3p3V	RA0/AN0	MCU_3p3V	Input Analog
Digital ground	VSS1	VSS1	Input Analog
Program data Enable	RB7/PGD	PGD	Bi-directional Discrete
Program clock Enable	RB6/PGC	PGC	Input Discrete
System reset	MCLR/RE3	MCLRM	Input Discrete
OSC1	OSC1/RA7	OSC1	Input Analog
OSC2	OSC2/RA6	OSC2	Input Analog
RS-422 serial communication			
UART 422 RX	RC7/RX1	UART_RX	Input Discrete
UART 422 TX	RC6/TX1/AN18	UART_TX	Output Discrete
TXRX_MCU	RB0/INT0/AN12	TXRX_MCU	Input Discrete
ADC analog inputs			
RF_INDET	RE2/AN7	RF_INDET	Input Analog
FFWR1	RE1/AN6	FFWR1	Input Analog
FFWR2	RE0/AN5	FFWR2	Input Analog
RREV	RA5/AN4/C2OUT	RREV	Input Analog

PA_TEMP	RD5/AN25	PA_TEMP	Input Analog
UP_TEMP	RB1/INT1/AN10	UP_TEMP	Input Analog
DOWN_TEMP	RB5/AN13/CCP3	DOWN_TEMP	Input Analog
System status indication			
LED_S1	RC0	LED_S1	Output Discrete
LED_S2	RB4/AN11	LED_S2	Output Discrete
Synthesizer RX down			
DOWN_SYNT_LD	RD4/AN24/SDO2	DOWN_SYNT_LD	Input Discrete
DOWN_SYNT_DATA	C5/AN17/SDO1	DOWN_SYNT_DATA	Output Discrete
DOWN_SYNT_CLK	RC4/SDA1/SDI1/AN16	DOWN_SYNT_CLK	Output Discrete
DOWN_SYNT_LE	RD3/AN23	DOWN_SYNT_LE	Output Discrete
DOWN_SYNT_CE	RD2/AN22	DOWN_SYNT_CE	Output Discrete
Synthesizer TX up			
UP_SYNT_LD	RD1/AN21/CCP4/SDA2/SDI2	UP_SYNT_LD	Input Discrete
UP_SYNT_DATA	RD0/AN20/SCL2/SCK2	UP_SYNT_DATA	Output Discrete
UP_SYNT_CLK	RC3/SCL/SCL1/SCK1	UP_SYNT_CLK	Output Discrete
UP_SYNT_LE	RC2/CCP1/AN1	UP_SYNT_LE	Output Discrete
UP_SYNT_CE	RC1/CCP2	UP_SYNT_CE	Output Discrete
DAC inputs – Need change connections to CPLD			
DAC_DATA	RA4/C1OUT	DAC_DATA	Input Discrete
DAC_CLK	RA3/AN3	DAC_CLK	Input Discrete
DAC_SYNC	RA2/AN2	DAC_SYNC	Input Discrete
AC_LDAC	RA1/AN1	AC_LDAC	Input Discrete
Sensors			
28V_SNS	RB3/CCP2/AN9	28V_SNS	Input Discrete
P7V_SENSE	P7V_SENSE	P7V_SENSE	Input Discrete
CPLD connection			
CPLD_RX	RD6/AN26/TX2	CPLD_RX	Input Analog
CPLD_TX	RD7/AN27/RX2	CPLD_TX	Output Discrete

7 Appendix A – How to set the registers values of sensitizer ADF-5355

** Please refer to unit datasheet for full details.

The Sensitizer ADF-5355 is software programmable unit which mean that each of the unit registers data is given and controlled by 12 registers that each one have control buffer of 32-bit. This registers configuration values will output from the on board MCU unit.

The main equation to calculate the output RF signal from this synthesizer is:

$$RF_{OUT} = INT + \frac{FRAC1 + \frac{FRAC2}{MOD2}}{MOD1} \times (f_{PFD}) / RF\ Divider$$

Where:

RFOUT is the RF output frequency.
INT is the integer division factor.
FRAC1 is the fractionality.
FRAC2 is the auxiliary fractionality.
MOD1 is the fixed 24-bit modulus.
MOD2 is the auxiliary modulus.
RF Divider is the output divider that divides down the VCO frequency.

And

$$f_{PFD} = REF_{IN} \times ((1 + D) / (R \times (1 + T)))$$

Where:
REFIN is the reference frequency input.
D is the REFIN doubler bit.
R is the REF reference division factor.
T is the reference divide by 2 bit (0 or 1).

All this registers values will set from the system host via serial communication channel.