|  |
| --- |
| **1410TX** |

**Software Requirements Specification (SRS)**

For the

**Software**

Of

SOQPSK TRANSMITTER

Version: 1.2

**Changes Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Action / Function | Name | Signature | Date |
| Prepared by | Asaf Elias |  | 30/12/2014 |
| Approved by | Tamir Blum |  | 30/12/2014 |
| SQA |  |  |  |
| Customer – approved by |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Changes** | **Version** | **Date** |
| All | 1st version | 1.0 | 30/12/2014 |
| 5.3 | Minor updates | 1.1 | 30/12/2014 |
| 5.3, 5.5 | Added a binary status. Added the SV command to save a specific setup version. | 1.2 | 18/05/2017 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of contents**

[1 Scope 5](#_Toc407697137)

[2 Referenced Documents 5](#_Toc407697138)

[2.1 Standards Documents 5](#_Toc407697139)

[2.2 Project documents 5](#_Toc407697140)

[3 Hardware Description 6](#_Toc407697141)

[3.1 Full Unit 6](#_Toc407697142)

[3.2 Setup Data 6](#_Toc407697143)

[4 Requirements 6](#_Toc407697144)

[5 Communications 7](#_Toc407697145)

[5.1 Serial Channel Definition 7](#_Toc407697146)

[5.2 Host Command Structure 8](#_Toc407697147)

[5.3 TTCCP Commands 9](#_Toc407697148)

[5.3.1 Set Frequency (FR) 10](#_Toc407697149)

[5.3.2 Set Data Source (DS) 10](#_Toc407697150)

[5.3.3 Set Data Polarity (DP) 10](#_Toc407697151)

[5.3.4 Set up SOQPSK (DE) 10](#_Toc407697152)

[5.3.5 Power High/Low (RP) 10](#_Toc407697153)

[5.3.6 RF Power (RF) 11](#_Toc407697154)

[5.3.7 Randomizer On/Off (RA) 11](#_Toc407697155)

[5.3.8 Operation Mode (MO) 11](#_Toc407697156)

[5.3.9 Bit Rate (IC) 11](#_Toc407697157)

[5.3.10 Internal Pattern (ID) 12](#_Toc407697158)

[5.3.11 High power Level (VP) 12](#_Toc407697159)

[5.3.12 Low power Level (VL) 12](#_Toc407697160)

[5.3.13 Negative Power Level (VM) 12](#_Toc407697161)

[5.3.14 Positive Power Level (VC) 13](#_Toc407697162)

[5.3.15 Clock Polarity (CP) 13](#_Toc407697163)

[5.3.16 Clock Source (CS) 13](#_Toc407697164)

[5.3.17 Help (H) 13](#_Toc407697165)

[5.3.18 Transmit On/Off (RB) 13](#_Toc407697166)

[5.3.19 Transmitter High on/off (RC) 14](#_Toc407697167)

[5.3.20 Return Firmware Version (VE) 14](#_Toc407697168)

[5.3.21 Request Current Parameters (Q) 14](#_Toc407697169)

[5.3.22 Save Parameters (SV) 14](#_Toc407697170)

[5.3.23 Reset to Factory Defaults (RE) 14](#_Toc407697171)

[5.4 1410TX Native Commands 15](#_Toc407697172)

[6 System Setup and Operation 16](#_Toc407697173)

[7 Hardware 16](#_Toc407697174)

[7.1 Microcontroller 16](#_Toc407697175)

[7.2 MCU Pin Assignments 17](#_Toc407697176)

[7.3 ARTM FPGA 18](#_Toc407697177)

[7.3.1 FPGA Write Registers 19](#_Toc407697178)

[7.3.2 Read Registers 20](#_Toc407697179)

[7.3.3 Bit Rate Register Calculation 21](#_Toc407697180)

[7.3.4 FPGA Initialization 21](#_Toc407697181)

[7.3.5 FPGA Register Modification 21](#_Toc407697182)

[7.4 AD5312 Dual DAC 22](#_Toc407697183)

[7.5 ADF4351 Synthesizer 24](#_Toc407697184)

[7.6 AD9746 High Speed DAC 26](#_Toc407697185)

[7.7 RS-485 channel 26](#_Toc407697186)

[7.8 Discrete I/O 27](#_Toc407697187)

**List of Figures**

[Figure 1 – Full Unit Block Diagram 6](#_Toc407697188)

[Figure 2 - Communications 7](#_Toc407697189)

[Figure 3 - MCU Hardware Schematic 16](#_Toc407697190)

[Figure 4 - FPGA, DAC and RF Section 18](#_Toc407697191)

[Figure 5 - ARTM FPGA SPI Interface 19](#_Toc407697192)

**List of Tables**

[Table 1 – List of TTCCP Commands 9](#_Toc407697193)

[Table 2 – List of $ Commands 15](#_Toc407697194)

[Table 3 - MCU Pin Assignments 17](#_Toc407697195)

[Table 4 - FPGA Write Registers 19](#_Toc407697196)

[Table 5 - FPGA Read Registers 20](#_Toc407697197)

[Table 6 - ADF4351 Registers 24](#_Toc407697198)

# Scope

The BA-SB-DTX10-V1 is a TELEMETRY TRANSMITTER.

It is an FCM/FM and SOQPSK Transmitter, designed to support a general building block to multiple projects. This product allows the selection of almost every basic transmitter parameter and fits to all modulation rates and bandwidths.

The product has two connectors:

1. RS485
2. RF out channel.

The Transmitter receives setup commands through the RS485 channel, and saves the setup in on-board non-volatile memory.

The DATA content a setup mode for the unit according to a code programming on the microchip's controllers and RS485 protocol, and one Frequency channel with SMA adaptor.

The unit is operated by two main devices:

* A Microchip PIC18 8-bit MCU.
* An Altera FPGA device.

# Referenced Documents

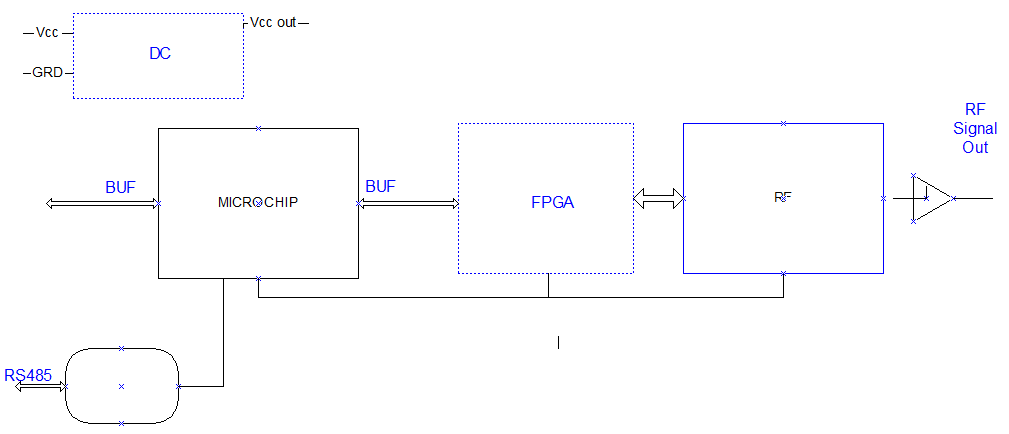
## Standards Documents

## Project documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **identification** | **Name/Description** | **Publishing Agency** | **Revision/Date** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

Customer Document

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **identification** | **Name/Description** | **Publishing Agency** | **Revision/Date** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |



# Hardware Description

## Full Unit

Figure 1 – Full Unit Block Diagram

## Setup Data

# Requirements

The transmitter has a variety of controllable definitions which the user can change according the need of his setup as mentioned in System Structure.

# Communications

The unit MCU has a few communication channels:

* RS-485 channel from host to unit MCU, allows sending commands to unit and changing its operational modes.
* SPI channel of MCU lets the MCU set up and operate the FPGA and DAC.
* Synchronous serial interface to control the VCO, which is used to select operating frequency.



Figure 2 - Communications

## Serial Channel Definition

To make the transmitter and receiver UART work together, they must agree on the same values of four parameters:

* Number of bits per character must be defined.
* Bit rate (bits per second) has to be the same for both host and MCU.
* If parity is used, both sides must agree on using odd or even parity.
* The number of stop bits must be the same for host and MCU.

The values selected for this system are:

|  |  |
| --- | --- |
| Data bits | 8 |
| Bit rate | 115.2 Kbps |
| Parity | none |
| Stop bit(s) | 1 |

## Host Command Structure

Commands from the host to MCU are divided into two groups:

1. TTCCP standard commands
2. B.A.Microwaves native $ commands

The $ commands are in the following format:

$<command> [optional parameter(s)]<cr>

Where:

<command> is a one or two letter designator.

optional parameter(s) are numbers or letters, which are predefined for each command.

<cr> is the binary character 13, for end of line.

The TTCCP command are in the following formats:

<command><cr> Request data set for this parameter

<command><data><cr> Set parameter to given data

\* If the transmitter is in Mode 1 (SOQPSK) then it will make a differential encoding

## TTCCP Commands

All TTCCP commands start with two capital letters, followed by a value for setting a parameter, or no value for requesting a reply from MCU with currently set value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Name** | **Parameter**  **values** | **Description** | **Paragraph** |
| Set Frequency | FR | 2158.0 - 2400.0 MHz | Steps of 100KHz |  |
| Data Source | DS | 0 or 1 | 1-Internal, 0-External |  |
| Data Polarity | DP | 0 or 1 | 0-Normal, 1-Inverted |  |
| Setup SOQPSK | DE | 0 or 1 | 0-off, 1-on\* |  |
| Power High/Low | RP | 0 or 1 | 0-Low, 1-High |  |
| RF Power | RF | 0 or 1 | 0-Off, 1-On |  |
| Randomizer | RA | 0 or 1 | 0-Off, 1-On |  |
| Operation Mode | MO | 0 to 3 | 0-PCM/FM  1-SOQPSK  2-Carrier Only  3-IQ imbalance |  |
| Bit Rate | IC | 1.00-30.00 Mbps | In 10KHz steps.  In the form of MM.KK |  |
| Internal Pattern | ID | 0 to 3 | 0-always 0  1-always 1  2-PTN15  3-PTN17 |  |
| High Power Level | VP | 20 to 40 | Value in dbm |  |
| Low Power Level | VL | 20 to 40 | Value in dbm |  |
| Negative Power Level | VM | 0 or 1 |  |  |
| Positive Power Level | VC | 0 or 1 |  |  |
| Clock Phase Shift | CS | 0 or 1 | 0 > No phase shift  1 > 180 Deg phase shift |  |
| Help | H |  | List available commands |  |
| Transmit On/Off | RB | 0 or 1 | 0 > "AIR"=RF1, "GND"=RF0  1 > "AIR"=RF0 "GND"=RF1 |  |
| Transmit High On/Off | RC | 0 or 1 | 0 > "AIR"=sv "+3V"=VL  1 > "AIR"=VL "+3V"=sv |  |
| Return Version | VE |  | Return Version and model |  |
| Return State | Q |  | Return all parameters |  |
| Save Parameters | SV |  | Save parameters in  non-volatile memory |  |
| Reset to Factory Defaults | RE |  | Factory default State:  Frequency 2185.0  Mode PCM/FM RF0 |  |
| Setup Version | VS | major.minor | Set the setup file version |  |

Table 1 – List of TTCCP Commands

### Set Frequency (FR)

Host instructs BA1410TX to set operating frequency

Format:

FR <frequency><cr>

Where:

<frequency> is defined in 100KHz steps, in range of 2158.0MHz to 2400.0 MHz

Example:

FR 2201.1<cr> Set frequency to 2201.1MHz

### Set Data Source (DS)

Host instructs BA1410TX to select source of data

Format:

DS <source><cr>

Where:

<source> is 1 for internal source or 0 for external source

Example:

DS 1<cr> Select external data source

### Set Data Polarity (DP)

Host instructs BA1410TX to set data polarity

Format:

DP <polarity><cr>

Where:

<polarity> is 0 for normal polarity or 1 for inverted polarity

Example:

DP 0<cr> Select normal data polarity

### Set up SOQPSK (DE)

Host instructs BA1410TX to select SOQPSK mode

Format:

DE <mode><cr>

Where:

<mode> 0 is off, 1 is on

Example:

DE 1<cr> Set SOQPSK to on

### Power High/Low (RP)

Host instructs BA1410TX to select High or Low power level

Format:

RP <level><cr>

Where:

<level> 0 is low, 1 is high

Example:

RP 1<cr> Select high power level

### RF Power (RF)

Host instructs BA1410TX to turn RF power on/off

Format:

RF <power><cr>

Where:

<power> 0 is power off, 1 is power on

Example:

RF 1<cr> Set RF power on

### Randomizer On/Off (RA)

Host instructs BA1410TX to turn randomizer on/off

Format:

RA <rand><cr>

Where:

<rand> 0 is randomizer off, 1 is randomizer on

Example:

RA 1<cr> Set randomizer on

### Operation Mode (MO)

Host instructs BA1410TX to select one of four operation modes

Format:

MO <mode><cr>

Where:

<mode> 0-PCM/FM  
1-SOQPSK  
2-Carrier Only  
3-IQ imbalance

Example:

MO 3<cr> Set mode to IQ imbalance

### Bit Rate (IC)

Host instructs BA1410TX to set data bit rate

Format:

IC <bitrate><cr>

Where:

<bitrate> 1.00 to 30.00 in 10KHz steps

Example:

IC 2.15<cr> Set bit rate to 2.15MHz

### Internal Pattern (ID)

Host instructs BA1410TX to set internal pattern

Format:

ID <pattern><cr>

Where:

<pattern> 0-always 0  
1-always 1  
2-PTN15   
3-PTN17

Example:

ID 2<cr> Set internal pattern to PTN15

### High power Level (VP)

Host instructs BA1410TX to set power level in dBm for high state

Format:

VP <level><cr>

Where:

<level> 20 to 40 dBm

Example:

VP 25<cr> Set power level to 25dBm in high state (RP 1)

### Low power Level (VL)

Host instructs BA1410TX to set power level in dBm for low state

Format:

VL <level><cr>

Where:

<level> 20 to 40 dBm

Example:

VL 29<cr> Set power level to 29dBm in low state (RP 0)

### Negative Power Level (VM)

Host instructs BA1410TX to set negative power level

Format:

VM <level><cr>

Where:

<level> 0 for low level, 1 for high level

Example:

VM 0<cr> Set negative power level to low

### Positive Power Level (VC)

Host instructs BA1410TX to set Positive power level

Format:

VC <level><cr>

Where:

<level> 0 for low level, 1 for high level

Example:

VC 0<cr> Set positive power level to low

### Clock Polarity (CP)

Host instructs BA1410TX to set clock polarity

Format:

CS <polarity><cr>

Where:

<polarity> 0 for no phase shift, 1 for 180o phase shift

Example:

CS 0<cr> No phase shift

### Clock Source (CS)

Host instructs BA1410TX to set clock source

Format:

CS <source><cr>

Where:

<source> 0 for external source, 1 for internal source

Example:

CS 0<cr> No phase shift

### Help (H)

Host requests a listing of available commands

Format:

H<cr>

Example:

H<cr>

### Transmit On/Off (RB)

Host instructs BA1410TX to set transmitter power on/off.

Format:

RB <power><cr>

Where:

<power> 0 is off ("AIR"=RF1, "GND"=RF0)  
1 is on ("AIR"=RF0, "GND"=RF1)

Example:

RB 1<cr> Set transmitter power on

### Transmitter High on/off (RC)

Host instructs BA1410TX to set transmitter power to high/low.

Format:

RC <level><cr>

Where:

<level> 0 is low ("AIR"=sv, "+3V"=VL)  
1 is high ("AIR"=VL, "+3V"=sv)

Example:

RC 1<cr> Set transmitter power to high

### Return Firmware Version (VE)

Host firmware version

Format:

VE<cr>

Example:

VE<cr>

### Request Current Parameters (Q)

Host requests dump of current values of all parameters.

Format:

Q<cr>

Example:

Q<cr>

### Save Parameters (SV)

Host instructs BA1410TX to save all parameters in no-volatile memory.

Format:

SV<cr>

Example:

SV<cr>

### Reset to Factory Defaults (RE)

Host instructs BA1410TX to set parameters to default values.

Factory default State:

* Frequency 2185.0
* Mode PCM/FM RF0

Format:

RE<cr>

Example:

RE<cr>

### Setup Version (VS)

Host instructs BA1410TX to set the setup file version

Format:

VS <major>.<minor> <cr>

Where:

< major > is a number between 0 to 255

< minor > is a number between 0 to 255

Example:

VS 3.1 <cr> Set setup file version to 3.1

## 1410TX Native Commands

The 1410TX native commands are mainly used for testing and basic setup, and are not used by the host computer for normal operation. These commands are listed here for reference only.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Command**  **Letters** | **Parameter**  **values** | **Description** |
| List commands | $H |  |  |
| Return analog value | $AI | 1 to 3 | Return analog value of ch 1,2 or 3 |
| Set frequency | $CF | 2158.0 to 2400.0 | In MHz |
| Set clock polarity | $CP | 0 or 1 |  |
| Reset DAC | $RD |  | Send a reset pulse to DAC |
| Set Bit Rate | $BR | 1.00 to 30.00 in 10KHz steps |  |
| Get FPGA register | $GF |  |  |
| Get D2A register | $GD |  |  |
| Set FPGA register | $SF |  |  |
| Set DAC register | $SD |  |  |
| Save setup | $SV |  |  |
| Return to TTCCP | $TTCCP |  |  |

Table 2 – List of $ Commands

## Binary Status Structure

The MCU can also be programmed to transmit a binary status to the host, this binary stracture is defined below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte** | **Name** | **Values** | **Description** |
| 0 | Sync 1 | 0xFB | Synchronization Byte 1 |
| 1 | Sync 2 | 0x60 | Synchronization Byte 2 |
| 2 | TX Counter |  | The counter increments by one each time an update is sent mod 255 |
| 3 | Setup Frequency |  | MSB of TX frequency |
| 4 | Setup Frequency |  | LSB of TX frequency |
| 5 | Setup Mode |  | State of modulation state |
| 6 | Bits Status 1 |  | 6.0 – Setup Data Source  6.1 – Setup Data Polarity  6.2 – Setup Randomizer  6.3 – Setup Clock Source  6.4 – Setup Clock Polarity  6.5 – Setup SOQPSK  6.6 – 0  6.7 – 0 |
| 7 | Temperature |  | MSB of the current temperature, 0x7F if power enable is not set. |
| 8 | Temperature |  | LSB of the current temperature, 0xFF if power enable is not set. |
| 9 | Bitrate |  | MSB of the Bitrate |
| 10 | Bitrate |  | LSB of the Bitrate |
| 11 | Current Power |  |  |
| 12 | Selected Internal Pattern |  | Data type while using internal data |
| 13 | Bits Status 2 |  | 6.0 – Setup Cot  6.1 – Setup RC  6.2 – Current Power Enable Value  6.3 – ‘1’ if Power enable equals the requested power level, ‘0’ if not.  6.4 – Setup Uart Status  6.5 – Reverse Power Status  6.6 – 0  6.7 – 0 |
| 14 | TX Low Power (VL) |  | TX low power in dBm (VL) |
| 15 | TX Low Power (VP) |  | TX low power in dBm (VP) |
| 16 | Reverse Power |  | Reverse power divided by 2 |
| 17 | Version |  | Version |
| 18 | Serial Number |  | MSB of Serial number |
| 19 | Serial Number |  | LSB of Serial number |
| 20 | Configuration Version |  | MSB of the Configuration version |
| 21 | Configuration Version |  | LSB of the Configuration version |
| 22 | Spare | 0xAA | Spare |
| 23 | Spare | 0xAA | Spare |
| 24 | Checksum |  | LSB of Checksum – addition of all the bytes in the message mod 255 (except the checksum bytes) |
| 25 | Checksum |  | MSB of Checksum – addition of all the bytes in the message mod 255 (except the checksum bytes) |

# System Setup and Operation

**Control Unit/ Microcontroller**

The microcontroller (the control unit of the Transmitter) will do the following:

The microcontroller will initialize the system, output level, setup, PLL, A2D and Altera FPGA unit.

control the Synthesizer. The synthesizer should get 4 data-lines from the MCU: Clock, Data, LE, LD.

The Synthesizer gets constant setting in power up.

\* Each channel has a shutdown mode. This is controlled by one bit per channel.

Low – the channel is off, High – the channel is on.

\* Unit setup Mode:

\* Read the temperature status of each channel

# Hardware

## Microcontroller

The following diagram is a partial schematic of the MCU surroundings.

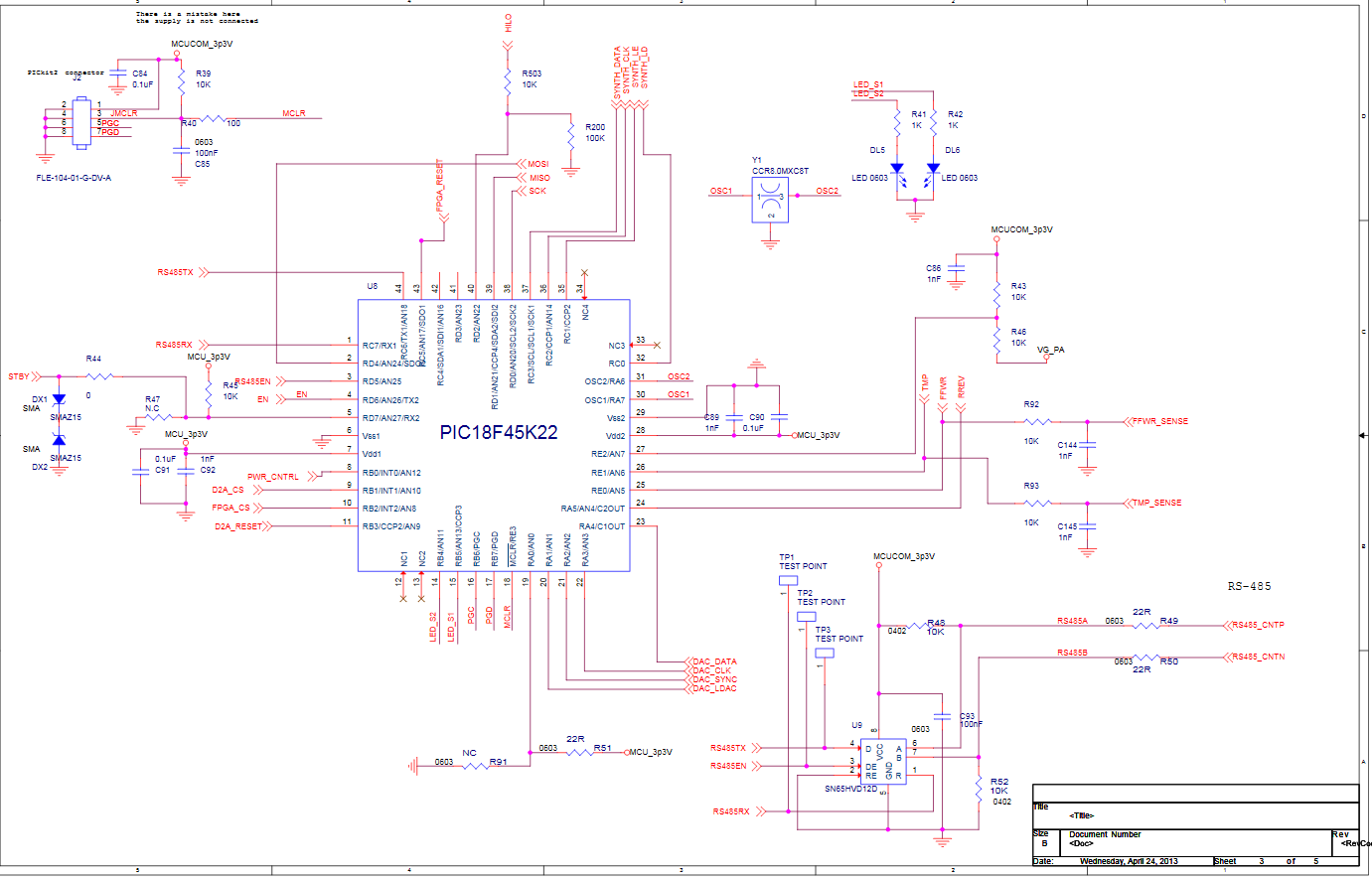


Figure 3 - MCU Hardware Schematic

The MCU is this unit is an 8-bit Microchip processor.

|  |  |
| --- | --- |
| Selected MCU | PIC18F45K22-I/PT |
| Operating voltage | 3.3V |
| Clock | External 8MHz crystal CCR8.0-MXC |
| Operating speed | 32MHz (using internal PLL multiplier) |

## MCU Pin Assignments

The PIC18F45K22 is a 44 pin device. The following table describes the usage of the various pins.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MCU Pin | Name | # pins | Direction | Description |
| RC1 | PLL\_LAT | 1 | O | Broadcast frequency PLL synchronous communication lines |
| RC2 | PLL\_CLK | 1 | O |
| RC3 | PLL\_DAT | 1 | O |
| RC0 | PLL\_LD | 1 | O |
| RB2 | CSN | 1 | O | FPGA Control |
| RD5 | RS485\_EN | 1 | O | RS485 Transmit enable |
| RB4 | LED1 | 1 | O | LED output |
| RB5 | LED2 | 1 | O | LED output |
| RB3 | D2A\_RESET | 1 | O | Main DAC control lines |
| RB1 | D2A\_CSB | 1 | O |
| RD4 | MOSI | 1 | O | SPI lines common to FPGA and main DAC devices |
| RD1 | MISO | 1 | I |
| RD0 | SCLK | 1 | O |
| RD6 | POWER\_EN | 1 | O | Set the voltage controllers |
| RD2 | HILO | 1 | O |  |
| RD7 | STANDBY | 1 | O |  |
| RA3 | DA2\_CLK | 1 | O | AD5312 control lines |
| RA4 | DA2\_DAT | 1 | O |
| RA2 | DA2\_SYN | 1 | O |
| RA1 | DA2\_LAT | 1 | O |
| RC6 | xmit | 1 | O | RS485 UART TX |
| RC7 | rcv | 1 | I | RS485 UART RX |
| RA6 | OSC2 | 1 | I | External Crystal lines |
| RA7 | OSC1 | 1 | O |
| Vss | Vss | 1 | I | MCU signal ground |
| Vdd | Vdd | 1 | I | MCU 3.3V power |
| RA5 | RREV | 1 | Analog | Reverse power input |
| RE1 | TMP | 1 | Analog | Temperature reading |
| RE0 | FFWR | 1 | Analog | Forward power reading |
| RA4 | DAC\_DATA | 1 | O | Control the gate power voltage |
| RA3 | DAC\_CLK | 1 | O |
| RA2 | DAC\_SYNC | 1 | O |
| RA1 | DAC\_LDAC | 1 | O |
| RE3 | MCLR | 1 | I | Microcontroller program/debug pins |
| RB7 | PGD | 1 | I |
| RB6 | PGC | 1 | I |
| RC1 | PLL\_LAT | 1 | O |  |
| RC2 | PLL\_CLK | 1 | O | ADF4351 Synthesizer |
| RC3 | PLL\_DAT | 1 | O |  |
| RC0 | PLL\_LD | 1 | O |  |

Table 3 - MCU Pin Assignments

## ARTM FPGA

The Altera FPGA is used for differential encoding of transmitted data. It sends digital data to the DAC device, for converting to analog stream.



Figure 4 - FPGA, DAC and RF Section

The MCU communicates with the FPGA using the following lines:

|  |  |  |
| --- | --- | --- |
| **MCU pin** | **Function** | **Description** |
| RB2 | FPGA\_CSN | FPGA chip select, active low |
| RD4 | MOSI | Data output from MCU |
| RD1 | MISO | Data output from FPGA |
| RD0 | SCLK | Clock from MCU |



Figure 5 - ARTM FPGA SPI Interface

### FPGA Write Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Address** | **Bits** | **Field** | **Modes** |
| 0 | 3:0  4  5  6 | Mode  Clk polarity  Data polarity  Randomizer | 0000 PCM-FM  0001 SOQPSK  0011 Offset Carrier  0110 Carrier Only  0= normal  1= inverted  0= normal  1= inverted  0= disabled  1= enabled |
| 1 | 7:0 | Phase offset | 2's complement |
| 2 | 7:0 | Bit rate | Bits 0 to 7 (LSB) |
| 3 | 7:0 | Bit rate | Bits 8 to 15 |
| 4 | 7:0 | Bit rate | Bits 16 to 23 |
| 5 | 7:0 | Bit rate | Bits 24 to 31 |
| 6 | 0  1  5:2 | Clock source  Data Source  Internal pattern | 0= external  1= internal  0= external  1= internal  0000 always 0  0001 always 1  0010 PN 15  0011 PN 17 |

Table 4 - FPGA Write Registers

The first eight bits of the control word are always the address, ranging from 0 to 31. The seven byte transfer can be accomplished by seven 16-bit transfers, with each transfer consisting of one address byte and one data byte.

A second method of transferring the seven data bytes is to send an eight byte transfer with the first word set to zero, corresponding to starting address 0, followed by the seven configuration bytes.

The CSN signal must remain low for the entire eight byte transfer.

### Read Registers

|  |  |  |
| --- | --- | --- |
| **Address** | **Bits** | **Function** |
| 16 | 0 | PLL Lock |
| 16 | 5:1 | FPGA Version |
| 17 | 7:0 | PLL Bit Rate [7:0] |
| 18 | 7:0 | PLL Bit Rate [15:8] |
| 19 | 7:0 | PLL Bit Rate [23:15] |
| 20 | 7:0 | PLL Bit Rate [31:24] |

Table 5 - FPGA Read Registers

The write bit rate establishes the modulator bit rate when no input data clock is present. If data is present with no data clock, the nominal bit rate must be close to this value.

When offset carrier mode is selected, the write bit rate establishes the frequency of the offset.

Consider the example of setting up the demodulator for 1 Mbps, given a 240 MHz sampling clock. The 32 bit control word (BR) is 01111111 h (17895697 decimal). If the configuration is performed with four transfers, the data for these transfers would be:

02h 01h

03h 11h

04h 11h

05h 11h

If the configuration is done with one transfer, the data would be:

02h 01h 11h 11h 11h

The sequence of bits on the SDI line for this case is:

00000010 00000001 00010001 00010001 00010001

The 32 bit control word which defines the modulator internal bit rate may be read on the SDO line. If the input data clock is active, this value corresponds to the frequency of that clock. The addresses associated with the read function are 17 through 20.

SDO 00010001 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

SDI zzzzzzzz aaaaaaaa bbbbbbbb cccccccc dddddddd

**ls byte** **ms byte**

When the modulator is tracking the input clock, the internal bit rate is constantly changing around the nominal bit rate.

### Bit Rate Register Calculation

The bit rate register contents is calculated according to currently selected bit rate:

float temp, bitspersec;

bitspersec = (float)bitrate;

temp = bitspersec / 240000000.0;

temp \*= 65536.0;

temp \*= 65536.0;

BitrateRegister = (int32)temp - 1;

### FPGA Initialization

All registers of the FPGA are written to on power up.

Before writing to the FPGA, the 32-bit bit rate value is computed as described in previous paragraph, and then split to registers 2 through 5, with least significant byte in register 2.

The other registers' values are concatenation of bits from fields in the setup structure.

### FPGA Register Modification

All FPGA registers may be written to or read from externally, using the $SF and $GF commands, respectively.

## AD5312 Dual DAC

The AD5312 dual DAC is used as follows:

* Channel A controls the pre-amplifier
* Channel B controls the amplifier

Channel A is continously modified, according to current analog reading of Forward Power.

The current level reference is selected as follows:

if (setup.rc == 1)

{

if (input(HILO) && setup.rp == 1)

level = power\_level; //SV

else if (input(HILO) && setup.rp == 0)

level = low\_power\_level; //VL

else if (!input(HILO) && setup.rp == 0)

level = power\_level; //SV

else

level = low\_power\_level; // VL

}

else

{

if (input(HILO) && setup.rp == 0)

level = power\_level; //SV

else if (!input(HILO) && setup.rp == 1)

level = power\_level;

else if (input(HILO) && setup.rp == 1)

level = low\_power\_level;

else

level = low\_power\_level; // VL

}

power = read\_analog(FORWARD\_POWER);

if (power > level + DEADBAND || power < level - DEADBAND)

{

if (power < level)

{

if (power\_control >= 150)

power\_control -= DEADBAND / 2;

}

else if (power\_control <= 1010)

{

power\_control += DEADBAND / 2;

}

set\_AD5312(DAC\_POS\_VOLT, power\_control);

}

Channel B is set up according to selected frequency, using preprogrammed values:

if (setup.frequency < FREQ\_LOW\_THRESH)

set\_AD5312(DAC\_NEG\_VOLT, setup.negative\_voltage[0]);

else if (setup.frequency < FREQ\_HIGH\_THRESH)

set\_AD5312(DAC\_NEG\_VOLT, setup.negative\_voltage[1]);

else

set\_AD5312(DAC\_NEG\_VOLT, setup.negative\_voltage[2]);

## ADF4351 Synthesizer

The ADF4351 synthesizer is controlled by the MCU, using three control lines:

|  |  |  |
| --- | --- | --- |
| **MCU pin** | **Function** | **Description** |
| RC1 | PLL\_LAT | Latch data internally |
| RC2 | PLL\_CLK | Synchronous clock |
| RC3 | PLL\_DAT | Serial data |

The ADF4351 is used to generate the transmission frequency of the transmitter.

Some of the synthesizer registers are set up with constant data, while registers 0 and 1 are set up according to selected frequency.

Table 6 - ADF4351 Registers

The ADF4351 registers are initialized as follows:

|  |  |
| --- | --- |
| **Register** | **Value** |
| 0 | 0x00DC0000L |
| 1 | 0x08008191L |
| 2 | 0x18005EC2L |
| 3 | 0x000004B3L |
| 4 | 0x0095003CL |
| 5 | 0x00400000L |

After initialization, registers 0,1 and 4 are modified to reflect the selected frequency.

The algorithm for setting up the values for registers 0,1 and 4 is:

FREQ\_STEP = 100; // 100KHz

FREQ\_OSC = 10000; // 10MHz

RFdiv = 2;

PLL\_RFdiv = 0x0095003CL;

if (Frequency >= 2200.0 MHz)

{

RFdiv = 1;

PLL\_RFdiv = 0x0085003CL;

}

D = D \* RFdiv \* FREQ\_STEP;

PLL\_int = D / FREQ\_OSC;

PLL\_frac = (D - (PLL\_int \* FREQ\_OSC)) / (FREQ\_STEP \* RFdiv);

FREQ\_MOD = (FREQ\_OSC / FREQ\_STEP) / RFdiv;

PLL\_mod = 0x08000000L + (FREQ\_MOD << 3) + 1L;

Register0 = (PLL\_int << 15) + (PLL\_frac << 3); // N counter latch

Register1 = PLL\_mod;

Register4 = PLL\_RFdiv;

## AD9746 High Speed DAC

The AD9746 DAC is set up by the MCU, and is fed with data by the FPGA, during transmission.

The MCU uses the same SPI lines to set up the DAC, with a separate chip select line.

|  |  |  |
| --- | --- | --- |
| **MCU pin** | **Function** | **Description** |
| RB3 | D2A\_RESET | A positive pulse causes a reset |
| RB1 | D2A\_CSB | DAC chip select, active low |
| RD4 | MOSI | Data output from MCU |
| RD1 | MISO | Data output from DAC |
| RD0 | SCLK | Clock from MCU |

The AD9746 DAC is not initialized automatically by the MCU. It is set up only if the host sends $SD commands to the MCU.

The DAC is reset upon power up.

All DAC registers may be externally written to or read from, using the $SD or $GD commands, respectively.

## RS-485 channel

The RS-485 chip is connected to USART1 of the MCU:

|  |  |  |
| --- | --- | --- |
| **MCU pin** | **Function** | **Description** |
| RC6 | TX | Asynchronous data out |
| RC7 | RX | Asynchronous data in |
| RD5 | EN | RS-485 transmit buffer enable |

The EN pin is used to enable transmission. It is asserted high just before transmission, and returned to low state after last character is completely sent out.

The EN is not allowed to stay high when not transmitting, since the RS-485 channel is half duplex, meaning only one device may transmit at the same time.

## Discrete I/O

The unit contains a few discrete I/O pins.

|  |  |  |  |
| --- | --- | --- | --- |
| MCU pin | Function | Direction | Description |
| RD6 | POWER\_EN | Output | The POWER\_EN is asserted high on power up. It is turned off during standby mode. |
| RD2 | HILO | Input | HILO selects high/low power level operation. It read by power\_output() function, and together with setup.rp, causes the function to select reference power level value. |
| RD7 | STANDBY | Input | External STANDBY mode selection. |