Communication Protocol Document

IF Receiver and Synthesizer (Ku-mIFRS)

Customer P/N: L3006-2300-500

Per Customer Spec.: SP-L3006-2300

Engineering Approval	**Date**
Y. Gashkalimy	30.10.2022
Customer Approval	**Date**

KRATOS | GMI EYAL
Microwave Electronics Division

Revision History

Revision	ECN	Date	Prepared by	Description
A	-----	25.11.2021	A. Miller	Created
B	-----	08.09.2022	A. Miller	Updated according to customer requests
C	-----	30.10.2022	Y. Gashkalimy	Updated communication protocol.

Table of Contents

List of Figures

1 Scope

The purpose of this document is to detail the serial communication protocol used for controlling the **Ku-mIFRS System** from an Electronic Assembly (EA), including the various extensions used for simulator and system control.

2 Relevant Documents

- Customer Specification Documents: SP-L3006-2300
- Customer ICD Document: 9750303600-18
- Outline drawing: 9750303600-12

3 Communication Protocol

3.1 Introduction

The host Electronic Assembly (EA) communicates with the Ku-mIFRS system through a Serial Peripheral Interface (SPI) protocol.

3.2 Hardware Definitions

The communication protocol between Electronic Assembly (EA) system and mIFRS system is a standard 4-wire SPI protocol, mode 0: SPI is a synchronous, full duplex master-slave-based interface.
The mIFRS system is a slave and the EA system is the master.

- SPI Clock frequency: 10MHz.
- SPI Mode 0: Clock polarity (CPOL) = 0, clock phase (CPHA) = 0: CLK idle state = low, data sampled on rising edge and shifted on falling edge.
- Dummy clock: 8 SPI Ticks.
    - SPI Slave (mIFRS) will return ACK/NACK message after 8 SPI dummy clocks.

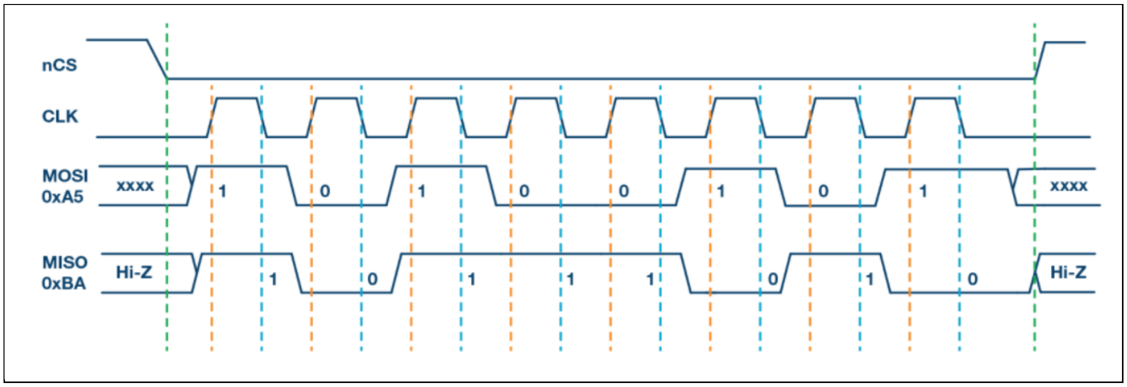


Figure 1: Example for SPI Mode 0

All communications will be initiated by the EA.
Communication interface description:

- Input SPI Clock (SCLK): *SPI_SCLK* (LVCMOS 1.8V)
- Input SPI Chip Select (CS): *SPI_CS* (LVCMOS 1.8V)
- Input SPI Master Out, Slave In (MOSI): *SPI_ MOSI* (LVCMOS 1.8V)
- Output SPI Master In, Slave Out (MISO): *SPI_ MISO* (LVCMOS 1.8V)
- Input discrete: *RF_UPDATE* (LVDS 2.5V)
- Input discrete: *TX_PRECMD* (LVDS 2.5V)
- Input discrete: *RCV_TRIG* (LVDS 2.5V)

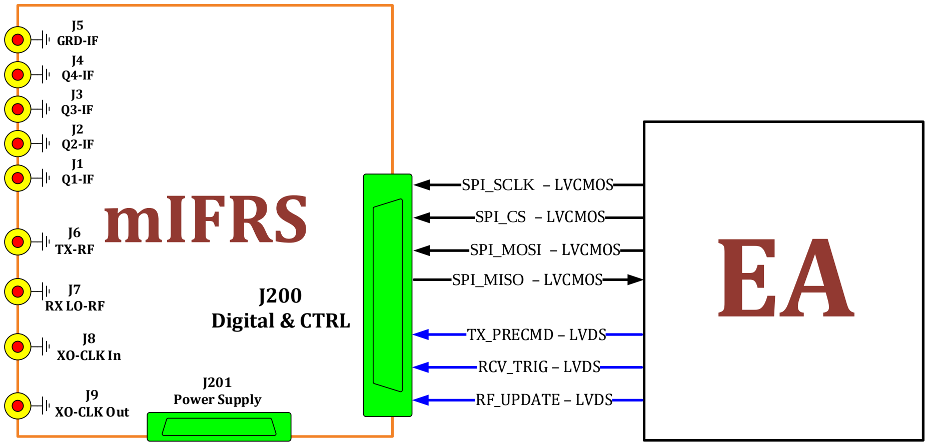


Figure 2: Serial Communication Diagram

3.3 Frame Structure – full parameter command

The frame is constructed from 16 header bytes then the data payload and completed with a checksum word (4 bytes).

- **Byte 0**: Preamble byte, which is Tx: 0x82, Rx: 0x83.
- **Byte 1**: Command opcode byte, which described in the following tables.
- **Byte 2÷3**: Message counter. Tx counter and Rx counter are independent, ascending number, LSB first. RESET value is '0'
- **Byte 4÷7**: Length of the entire command (including the header bytes), LSB first.
- **Byte 8**: Command activation. all commands are triggered to one of the discrete below:
  1. 'RF_UPDATE' Triggered commands: The response to this type of message shall be at the next start of 'RF_UPDATE' discrete. Upon assertion of 'RF_UPDATE' signal, the OGT shall start the process of changing RF control parameters as required by EA command. The reply to this message shall start right after the receiving command – see Figure 3.
     CMD Value: 0x01
  2. 'TX_PRECEDING_CMD' Triggered commands: The response and reply to this type of message shall be at the next start of 'TX_PRECEDING_CMD' discrete (RF transmission pulse) – see Figure 3.
     CMD Value: 0x02
  3. 'RCV_TRIGGER' Triggered commands: The response and reply to this type of message shall be at the next start of 'RCV_TRIGGER' discrete – see Figure 3.
     CMD Value: 0x04
- **Byte 9÷11**: Spare
- **Byte 12÷15**: Message time tag:
  - Tx Message Time: mSEM time tag
  - Rx Message Time: reply the mSEM time tag
- **Data Bytes**: Payload Data,
  - Tx Data: 20 bytes long. Bytes #: 16÷35
  - Rx Data: 16 bytes long followed by 44 bytes status. Bytes #: 16÷75

  When Error – returns checksum value, preamble and opcode values.

  ‘Status Response' shall be sent at the end of each Rx message.
- **Checksum Bytes:** Checksum words (4 bytes),
  Which is the result of the sum of all words (4 bytes) in the frame (unsigned).
  - Tx Bytes #: 36÷39 , Rx Bytes #: 76÷79

Tx Frame (EA → mIFRS):

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8	9÷11	12÷15 (LSB 1st)
Byte Description	0x82	CMD	Count	Length	Activation	0x00	Time

Byte Name	Tx Data 20 Bytes			Check Sum
Byte Number	16	…	35	36÷39
Byte Description	TD0	…	TD19	CHK

Total of 40 Bytes

Rx Frame (mIFRS → EA):

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8	9÷11	12÷15 (LSB 1st)
Byte Description	0x83	CMD	Count	Length	Activation	0x00	Time

Byte Name	Rx Data – zero padding not in use. 16 Bytes		
Byte Number	16	. . .	31
Byte Description	0x00	. . .	0x00

Byte Name	Status Response (*)	Check Sum
Byte Number	32÷75 44 Bytes	76÷79
Byte Description	System status response	CHK

Total of 80 Bytes

(*) 'Status Response' shall be sent at the end of each Rx message
(**) Between two following Rx messages there will be time of at least 100 clocks without traffic on the SPI BUS

Error message format:

Rx Frame (mIFRS → EA):

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8	9÷11	12÷15 (LSB 1st)
Byte Description	0x83	Error code	Count	Length	Activation	0x00	Time

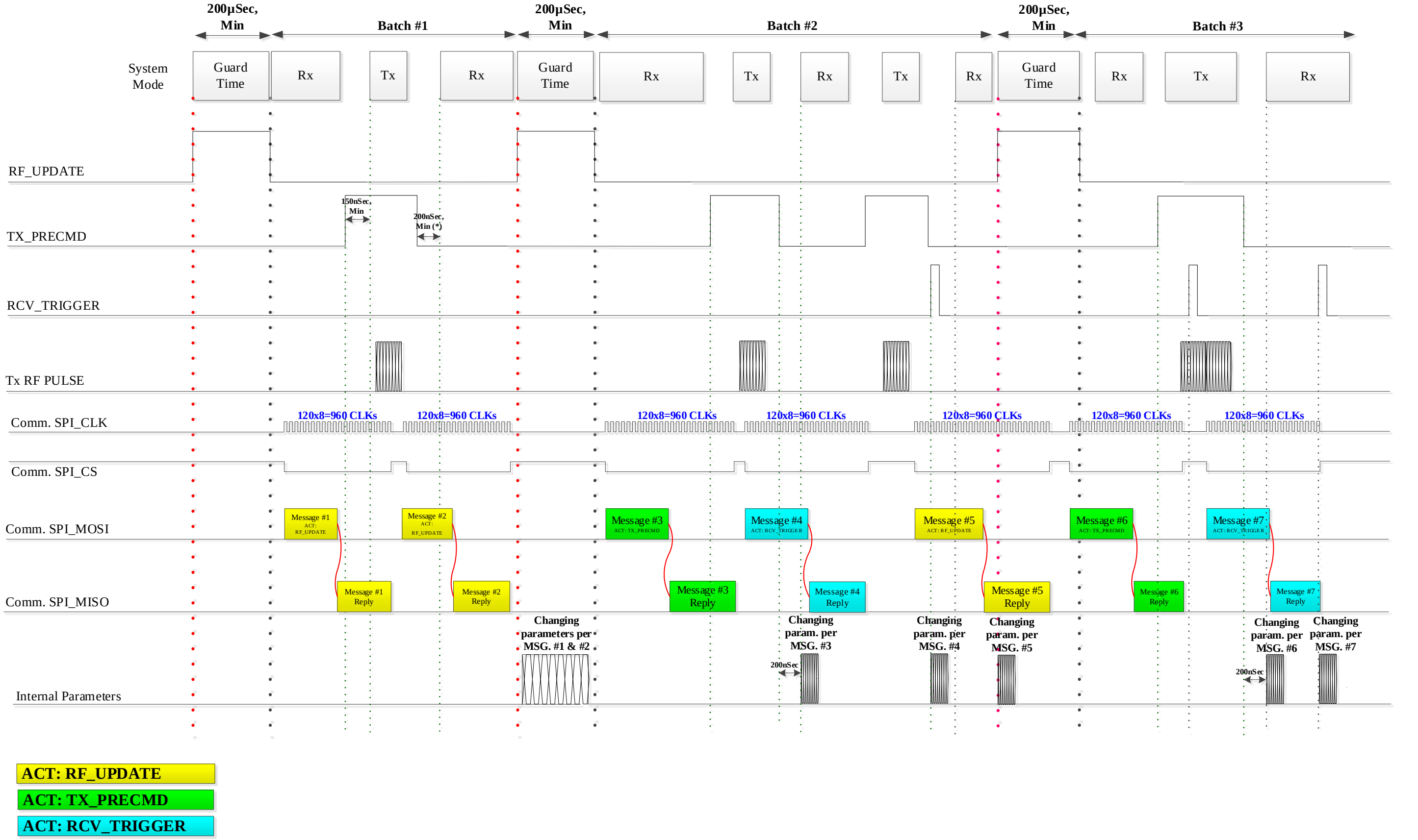
Byte Name	Read checksum	Calculated checksum	Received preamble	Received opcode	Spare –zero padding
Byte Number	16÷19 (LSB 1st)	20÷23 (LSB 1st)	24	25	26÷31 (LSB 1st)
Byte Description	Received checksum	Calculated checksum	Preamble value	Opcode value	0x000000000000

Byte Name	Status Response (*)	Check Sum
Byte Number	32÷75 44 Bytes	76÷79
Byte Description	System status response	CHK

Total of 80 Bytes

(*) 'Status Response' shall be sent at the end of each Rx message
(**) Between two following Rx messages there will be time of at least 100 clocks without traffic on the SPI BUS

3.4 Timing Diagram



(*) Rx state start 200nSec after *TX_PRECMD* falling edge, Tx parameters can be updated at Guard time or at Rx state only

Figure 3: Timing Diagram

- Minimum 'RF_UPDATE' pulse duration is 200µSec.
- Batch is define by the time between two consecutive 'RF_UPDATE' Pulses.
- If during RF update process, comes a requirement for generating a new Tx pulse (by TX_PRECMD) - this RF pulse shall be generated, but with no guaranty for its RF substance

3.5 Frame Structure – FPGA Set & Get registers commands

The frame is constructed from 16 bytes frame.

- **Byte 0**: Preamble byte, which is Tx: 0x82, Rx: 0x83.
- **Byte 1**: Command opcode byte, which described in the following tables.

  0x21 – set register (write command).

  0x22 – get register (read command).
- **Byte 2÷3**: Message counter. Ascending number, LSB first. RESET value is '0'.
- **Byte 4÷7**: FPGA internal registers address field, LSB first.
- **Byte 8÷11**: FPGA internal registers data field, LSB first.

  Write command: write data value - 4 bytes, and FPGA returns 0xFFFFFFEE.

  Read command: write data value – 0x00000000 and FPGA returns the register value – 32 bit.
- **Byte 12÷35**: Zero padding, spare bytes.
- **Data Bytes**: Payload Data,
  - Tx Data: 20 bytes long. Bytes #: 16÷35 – NA.
  - Rx Data: 16 bytes long followed by 44 bytes status. Bytes #: 16÷75

    When Error – returns checksum value, preamble and opcode values.

  ‘Status Response' shall be sent at the end of each Rx message.
- **Checksum Bytes:** Checksum words (4 bytes),
  Which is the result of the sum of all words (4 bytes) in the frame (unsigned).
  - Tx Bytes #: 36÷39
  - Rx Bytes #: 76÷79

Write Command:

Tx Frame (Host → mIFRS):

Byte Name	Preamble	CMD Opcode	Message Counter	Register Address	Write Data	Zero padding
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8÷11 (LSB 1st)	12÷15 (LSB 1st)
Byte Description	0x82	0x21	Counter	Address – 32 bit	Data – 32 bit	0x00000000

Byte Name	Tx Data – zero padding 20 Bytes			Check Sum
Byte Number	16	…	35	36÷39
Byte Description	0x00	…	0x00	CHK

Total of 40 Bytes

Rx Frame (mIFRS → Host):

Byte Name	Preamble	CMD Opcode	Message Counter	Register Address	Read Data	Zero padding
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8÷11 (LSB 1st)	12÷31 (LSB 1st)
Byte Description	0x83	0x21	Counter	Address – 32 bit	Return value 0xFFFFFFEE	0x00000000

Byte Name	Status Response (*)	Check Sum
Byte Number	32÷75 44 Bytes	76÷79
Byte Description	System status response	CHK

Total of 80 Bytes

Read Command:

Tx Frame (Host → mIFRS):

Byte Name	Preamble	CMD Opcode	Message Counter	Register Address	Write Data - NA	Zero padding
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8÷11 (LSB 1st)	12÷15 (LSB 1st)
Byte Description	0x82	0x22	Counter	Address – 32 bit	0x00000000	0x00000000

Byte Name	Tx Data – zero padding 20 Bytes			Check Sum
Byte Number	16	…	35	36÷39
Byte Description	0x00	…	0x00	CHK

Total of 40 Bytes

Rx Frame (mIFRS → Host):

Byte Name	Preamble	CMD Opcode	Message Counter	Register Address	Read Data	Zero padding
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8÷11 (LSB 1st)	12÷15 (LSB 1st)
Byte Description	0x83	0x22	Counter	Address – 32 bit	Return register value – 32 bit	0x00000000

Byte Name	Status Response (*)	Check Sum
Byte Number	32÷75 44 Bytes	76÷79
Byte Description	System status response	CHK

Total of 80 Bytes

Error message format:

When error message occurs FPGA will return Nack message with error code.

Rx Frame (mIFRS → EA):

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte Number	0	1	2÷3 (LSB 1st)	4÷7 (LSB 1st)	8	9÷11	12÷15 (LSB 1st)
Byte Description	0x83	Error code	Count	Length	Activation	0x00	Time

Byte Name	Read checksum	Calculated checksum	Received preamble	Received opcode	Spare –zero padding
Byte Number	16÷19 (LSB 1st)	20÷23 (LSB 1st)	24	25	26÷31 (LSB 1st)
Byte Description	Received checksum	Calculated checksum	Preamble value	Opcode value	0x000000000000

Byte Name	Status Response (*)	Check Sum
Byte Number	32÷75 44 Bytes	76÷79
Byte Description	System status response	CHK

Total of 80 Bytes

(*) Ack/Nack message shall be sent at the end of each Tx frame.
(**) Between two following Rx messages there will be time of at least 100 clocks without traffic on the SPI BUS.
(***) Error code shall be sent in Nack messages according to chapter 3.4.
(****) Error message bytes: 8, 11 to 15 holds invalid value and must be ignore

3.6 Error Codes

The target will return an error code in the CMD field when an error occurs.

No action will be taken if there is an error.

The following error codes supported:

Command	**Code**	**Remarks**
Header error	0xF0	Will return one parameter with the unidentified header
Command error	0xF1	Will return one parameter with the unidentified command
Checksum error	0xF2	Will return two parameters with the received checksum and the calculated checksum
Data error	0xF3	Not implemented
Execution Error	0xF4	Not implemented
Time-out Error type 1	0xF5	FPGA, Communication timeout event.
Time-out Error type 2	0xF6	
Time-out Error type 3	0xF7	
Reserved	0xF8	
Reserved	0xF9	
Reserved	0xFA	
Reserved	0xFB	
Reserved	0xFC	
Reserved	0xFD	
Reserved	0xFE	
Reserved	0xFF	

Table 1: Error Codes Table

4 Commands

4.1 Commands List

Number	Command	Function	Direction	Remarks
Operational Commands				
1.	0x51	Full Parameters Command	Write	
Error Codes				
2.	0xF0-0xFF	See Table 1		
Debug Commands				
3.	0x21	Set FPGA Register Value	Write	
4.	0x22	Get FPGA Register Value	Read	

Table 2: Commands List

4.2 Details Operational Command Description

4.2.1 Full Parameters Command

Description: Set all mIFRS main parameters.

Command OpCode: `0x 51`

4.2.1.1 Full Parameters Command Tx Data

Length: `0x 00 28` `// (40Dec)`
Tx Data: `See table below`

Tx Header – 16 Bytes

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte #	**0**	**1**	**2÷3**	**4÷7**	**8**	**9÷11**	**12÷15**
Byte Description	0x82	0x51	Count	0x28	Activation	0x00	mSEM Time Tag

Tx Data – 20 Bytes

Byte #	**16**	**17**	**18**	**19**
Description	Mask Byte #1 See Paragraph: 4.2.1.2	Mask Byte #2 See Paragraph: 4.2.1.2	Spare	System Mode

Byte #	**20**	**21**	**22**	**23**
Description	Reset Alarm Counter	Spare	Tx Frequency	Tx Input Power

Byte #	**24**	**25**	**26**	**27**
Description	Tx Duty Cycle	Spare	Spare	Rx Frequency

Byte #	**28**	**29**	**30**	**31**
Description	Rx CH#1 IF Attenuator	Rx CH#2 IF Attenuator	Rx CH#3 IF Attenuator	Rx CH#4 IF Attenuator

Byte #	**32**	**33**	**34**	**35**
Description	Rx GRD CH IF Attenuator	Spare	Spare	Spare

Tx Check Sum – 4 Bytes

Byte #	**36**	**37**	**38**	**39**
Description	Check Sum			

Table 3: Full Parameters Command Tx Data

4.2.1.2 Masking Bytes Description

Item	Command Name	Full Parameters Command (0x01) Masking Bytes Data		Message Function Link	Command Byte #
		Message Byte Number	Bit Number		
1.	System Mode	Byte #16	Bit #7	*System Mode Byte*	19
2.	Reset Alarm Counter		Bit #6	*Reset Alarm Counter Byte*	20
3.	Spare		Bit #5	*Spare*	21
4.	Tx Frequency		Bit #4	*Tx Frequency Index Byte*	22
5.	Tx Input Power		Bit #3	*Tx Power Index Byte*	23
6.	Tx Duty Cycle		Bit #2	*Tx Duty Cycle Byte*	24
7.	Spare		Bit #1	*Spare*	25
8.	Spare		Bit #0	*Spare*	26
9.	Rx Frequency	Byte #17	Bit #7	*Rx Frequency Index Byte*	27
10.	Rx CH#1 IF Attenuator		Bit #6	*Rx Attenuators Bytes*	28
11.	Rx CH#2 IF Attenuator		Bit #5	*Rx Attenuators Bytes*	29
12.	Rx CH#3 IF Attenuator		Bit #4	*Rx Attenuators Bytes*	30
13.	Rx CH#4 IF Attenuator		Bit #3	*Rx Attenuators Bytes*	31
14.	Rx GRD CH IF Attenuator		Bit #2	*Rx Attenuators Bytes*	32
15.	Spare		Bit #1	*Spare*	33
16.	Spare		Bit #0	*Spare*	34

Table 4: Masking Bytes Description

** Mask bit logic: '0' – Disable, '1' – Enable

4.2.1.3 System Mode Byte

Description: Set mIFRS system mode.

Tx Data:

```
Operational  = 0    // TDD, Reset Value
Simultaneous = 1    // Tx and Rx are simultaneous
BIT          = 2
Standby      = 3    // TBD
```

4.2.1.4 Reset Alarm Counter Byte

Description: Reset all mIFRS alarm counters.

Tx Data:

```
0: Reset all alarm counters             // Reset Value
1: Not reset
```

4.2.1.5 Tx Frequency Index Byte

Description: Set transmitter frequency index.

Tx Data:

```
Index  = 0÷80       // FLOW=0, FHIGH=80, Reset Value: 40
```

4.2.1.6 Tx Power Index Byte

Description: Set transmitter input power index.

Tx Data:

```
Index  = 0÷3, Resolution=TBD     // HIGH=0, LOW=3, Reset Value: 0
```

4.2.1.7 Tx Duty Cycle Byte

Description: Set transmitter duty cycle.

Tx Data:

```
Duty Cycle = 0÷35, Resolution=1%       // Reset Value: 30%
```

4.2.1.8 Rx Frequency Index Byte

Description: Set receiver frequency index.

Tx Data:

```
Index  = 0÷80       // FLOW=0, FHIGH=80, Reset Value: 40
```

4.2.1.9 Rx Attenuators Bytes

Description: Set all receivers attenuators value.

Tx Data:

```
Byte #28 – Channel #1 IF Attenuator
   • Attenuation Value = 0÷20, Resolution=1dB // Reset Value = 0

Byte #29 – Channel #2 IF Attenuator    // the same as Byte #28
Byte #30 – Channel #3 IF Attenuator    // the same as Byte #28
Byte #31 – Channel #4 IF Attenuator    // the same as Byte #28
Byte #32 – GRD Channel IF Attenuator   // the same as Byte #28
```

4.2.1.10 Full Parameters Command Rx Data

Length: `0x 00 50` `// (80Dec)`

Rx Data: *See table below*

Rx Header – 16 Bytes

Byte Name	Preamble	CMD Opcode	Message Counter	Message Length	CMD Activation	Spare	Message Time
Byte #	**0**	**1**	**2÷3**	**4÷7**	**8**	**9÷11**	**12÷15**
Byte Description	0x83	0x51	Count	0x50	Activation	0x00	Reply mSEM Time Tag

Rx Data – 16 Bytes

Byte #	**16**	**…….**	**…….**	**31**
Description	NA - Zero padding			

Status Response – 44 Bytes

Byte #	**32**	**…….**	**…….**	**75**
Description	See Paragraph 4.2.1.11: Get Status			

Rx Check Sum – 4 Bytes

Byte #	**76**	**77**	**78**	**79**
Description	Check Sum			

Table 5: Full Parameters Command Rx Data

4.2.1.11 Get Status

Data	Byte	Bit	Description
mIFRS Actual System Mode	32	0	1 = Operational 2 = Simultaneous 3 = BIT 4 = Standby
		1	
Spare		2	'00000'
		3	
		4	
		5	
		6	
mIFRS Fail		7	‘0’ - No Fail ‘1’ – Fail (Fail = **OR** of all the failure bits)
Serial Number	33 ÷ 34		16 bits resolution LSB=1; FS=65535
FPGA Firmware Version	35 ÷ 36		Version Low: 0xLL (byte 35) Version High: 0xHH (byte 36) Full version: HH.LL
FPGA Software Version	37 ÷ 38		Version Low: 0xLL (byte 37) Version High: 0xHH (byte 38) Full version: HH.LL
Reserved	39		0x00
INPUT_VOLTAGE	40	[7:0]	INPUT_VOLTAGE [V] = TBD
INPUT_CURRENT	41	[7:0]	INPUT_CURRENT [mA] = TBD
DIGITAL_3V3	42	[7:0]	DIGITAL_3V3 [V] = TBD
RF_3V8	43	[7:0]	RF_3V8 [V] = TBD
FPGA_CORE_1V	44	[7:0]	FPGA_CORE_1V [V] = TBD
FPGA_DDR_1V35	45	[7:0]	FPGA_DDR_1V35 [V] = TBD
FPGA_AUX_1V8	46	[7:0]	FPGA_AUX_1V8 [V] = TBD
AFE79XX_0V95	47	[7:0]	AFE79XX_0V95 [V] = TBD
AFE79XX_1V2	48	[7:0]	AFE79XX_1V2 [V] = TBD
AFE79XX_1V8	49	[7:0]	AFE79XX_1V8 [V] = TBD
FPGA_TEMP	50	[7:0]	FPGA_TEMP [ºC] = TBD
ADC_TEMP	51	[7:0]	ADC_TEMP [ºC] = TBD

Data	Byte	Bit	Description
INPUT_VOLTAGE_OV_COUNTER	52	[7:0]	8 bits
INPUT_VOLTAGE_UV_COUNTER	53	[7:0]	8 bits
INPUT_VOLTAGE_OI_COUNTER	54	[7:0]	8 bits
INPUT_VOLTAGE_UI_COUNTER	55	[7:0]	8 bits
DIGITAL_3V3_OV_COUNTER	56	[7:0]	8 bits
DIGITAL_3V3_UV_COUNTER	57	[7:0]	8 bits
RF_3V8_OV_COUNTER	58	[7:0]	8 bits
RF_3V8_UV_COUNTER	59	[7:0]	8 bits
FPGA_CORE_OV_COUNTER	60	[7:0]	8 bits
FPGA_CORE_UV_COUNTER	61	[7:0]	8 bits
FPGA_DDR_OV_COUNTER	62	[7:0]	8 bits
FPGA_DDR_UV_COUNTER	63	[7:0]	8 bits
FPGA_AUX_OV_COUNTER	64	[7:0]	8 bits
FPGA_AUX_UV_COUNTER	65	[7:0]	8 bits
AFE79XX_0V95_OV_COUNTER	66	[7:0]	8 bits
AFE79XX_0V95_UV_COUNTER	67	[7:0]	8 bits
AFE79XX_1V2_OV_COUNTER	68	[7:0]	8 bits
AFE79XX_1V2_UV_COUNTER	69	[7:0]	8 bits
AFE79XX_1V8_OV_COUNTER	70	[7:0]	8 bits
AFE79XX_1V8_UV_COUNTER	71	[7:0]	8 bits
Reserved	72	[7:0]	0x00
SUM_COUNTER_ALARMS	73 ÷ 74	[15:0]	Sum of all alarm counters, 15 bits
Reserved	75	[7:0]	0x00

Table 6: Get Status Response

Total Status Length: 44 bytes

4.3 Details Debug Command Description

4.3.1 Hardware Definitions

The communication protocol between PC and simulator system is a standard UART. All communications will be initiated by the master.

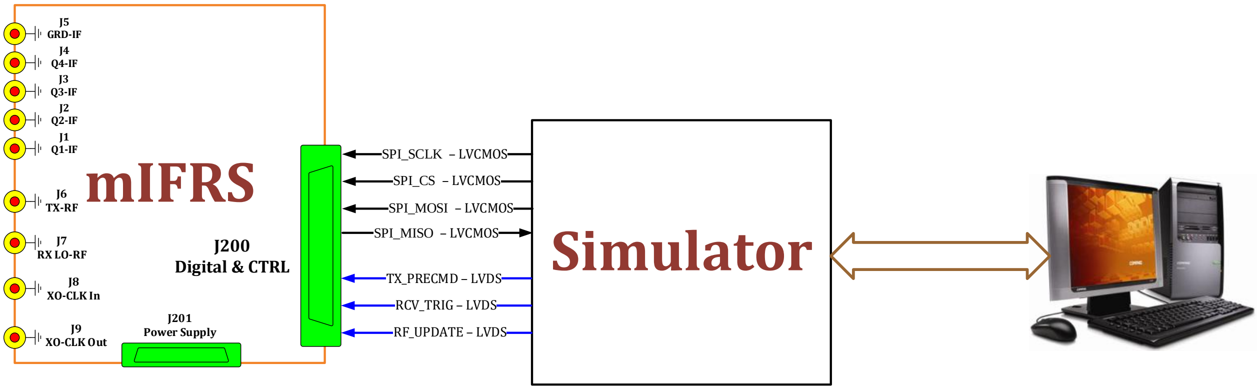
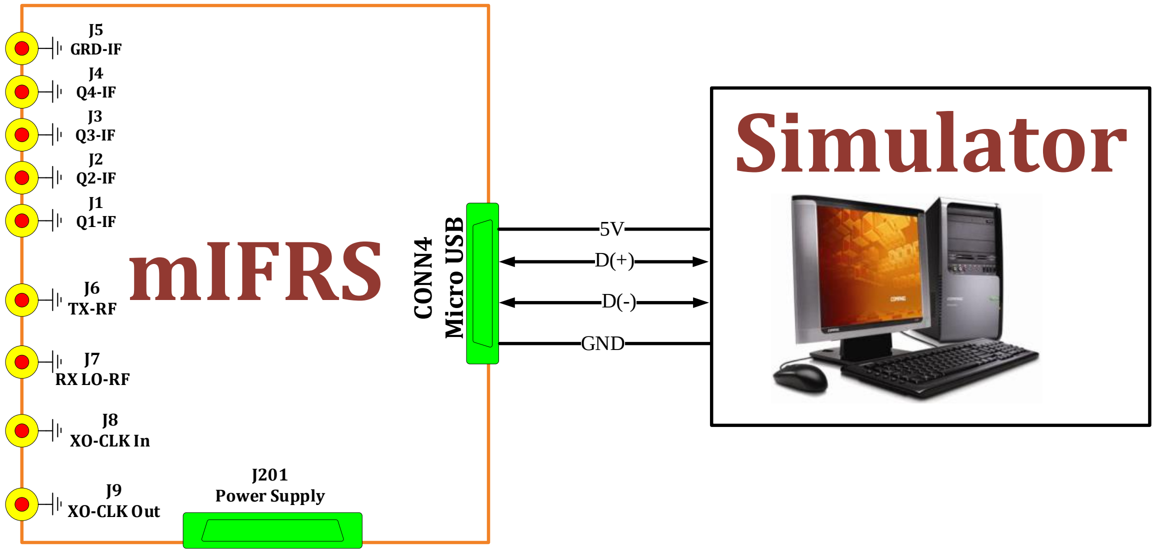
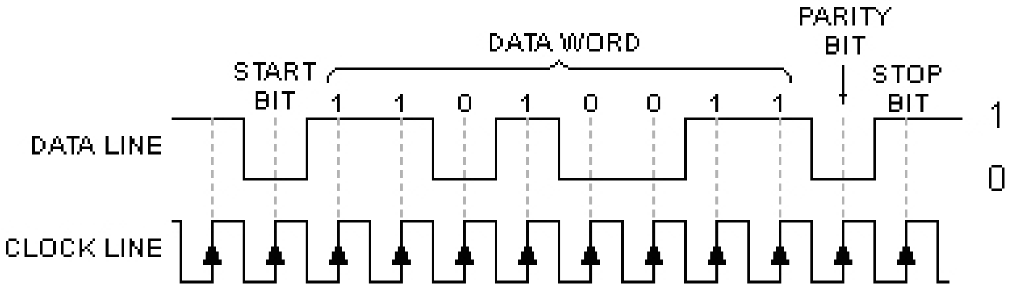


Figure 4: SPI Serial interconnection diagram

4.3.2 Serial Frame Structure

UART Over micro USB





- Start bit - 1
- Data bits – 8 (lsb sent first).
- Parity bit – 1 even parity bit.
- Stop bit – 1 bit.
- Speed – 115200 bps.