IRS cRIO OPIF 10MBit/s

Optical interface 3-fold Tx/Rx 10MBit/s for Compact RIO

Manual

Contents

1. General

1.1 Change History

Date	Version	Changes
05.06.2012	1.0	First draft

Table 1: Change History

1.2 Purpose of the document

This document describes the optical interface module for Compact-RIO, which contains

- 3 optical Transmitters
- 3 optical Receivers

Both transmitters and receivers are suitable for “Versatile Link” plastic optical fibers at speeds up to 10MBit/s respectively. In the following chapters the hardware and the basic software drivers are highlighted.

1.3 Applicable documents

Nr.	Document	Date	Remark
1	cRIO Optical Interface 10MBits_V1.0_Schem.pdf	25.04.2012	Schematics
2	AVAGO_HFBR_x528.pdf	07.12.2007	Optical transceiver data sheet

Table 2: Applicable documents

1.4 Abbreviations

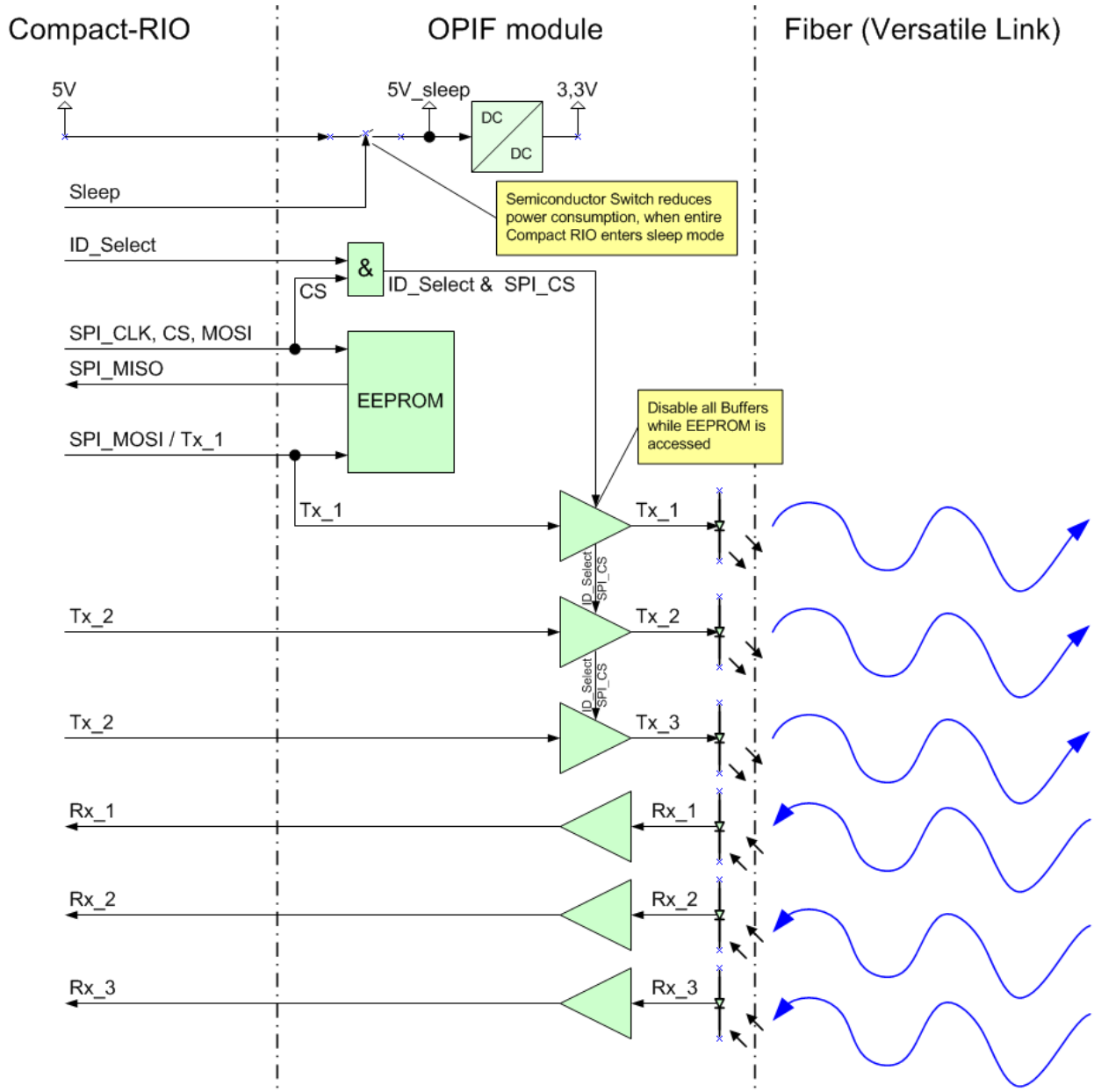
Abbreviation	Description	Remarks
cRIO	Compact RIO	Measurement device platform of National Instruments
OPIF	Optical Interface	

Table 3: Abbreviations

1. Hardware

1.1 Overview

Following block diagram shows the hardware of the module:



During **start-up or reset** of the Compact-RIO system the IOs between the optical interface module and the Compact-RIO are floating. The module is designed in this way, that the transmitter outputs do not emit light. I.e. **NO LIGHT** should be regarded as the **SAVE STATE**.

If the **EEPROM is accessed** from the Compact-RIO, all transmit Buffers are disabled. **No light is emitted** by the transmitters.

For detailed description of the circuits, refer to the schematics (applicable document Nr. 1)

1.2 Technical data

Following table shows the technical data of the module.

Item	Min	Typical	Max	Unit
Ambient Temperature	0	25	60	°C
Supply Voltage (supplied by Compact-RIO)	4.75	5.0	5.25	V_{DC}
Supply current (all transmitters off)	40	50	60	mA
Baudrate	0	10	15	MBit/s
Forward current per transceiver (Light On, Logic level Low)		48		mA
Driver buffer propagation delay (between Compact-RIO and transmitter LED) additional to HFBR1528		3	5	ns
Receiver buffer propagation delay (between Receiver and Compact RIO) additional to HFBR2528		2	5	ns

For detailed technical data of transmitters and receivers, refer to HFBR1528/2528 datasheet (document Nr. 2)

2. Software overview

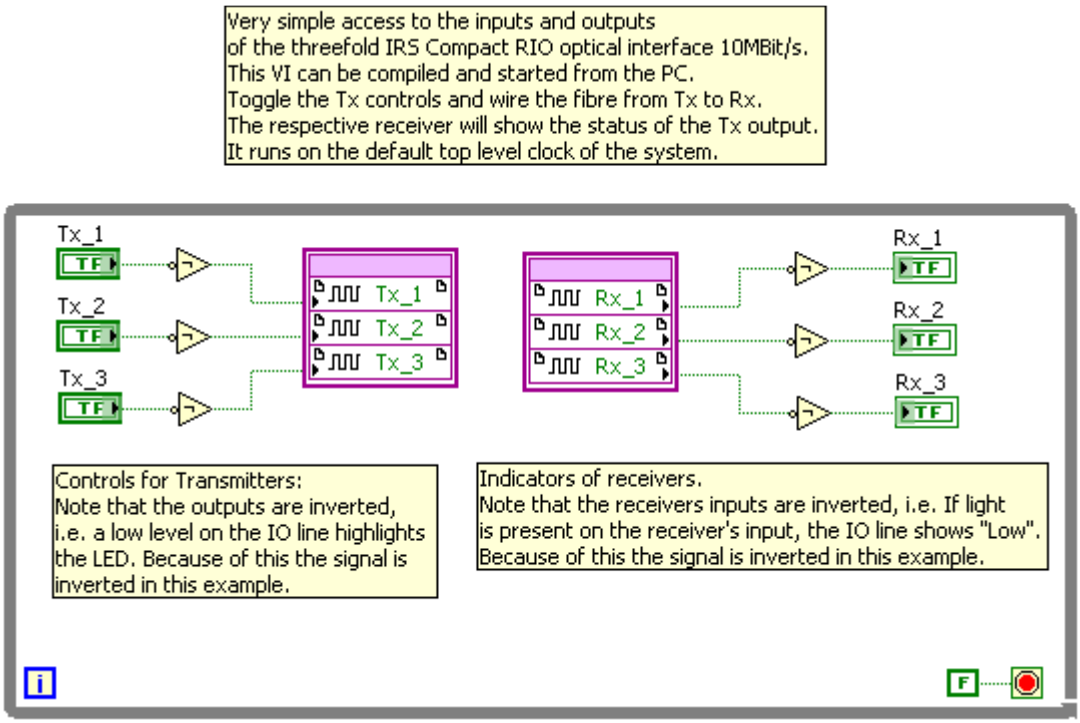
In the following chapter the example projects are described. In the first example, the user is introduced, how to include the module into the user's application project.

2.1 Simple access example

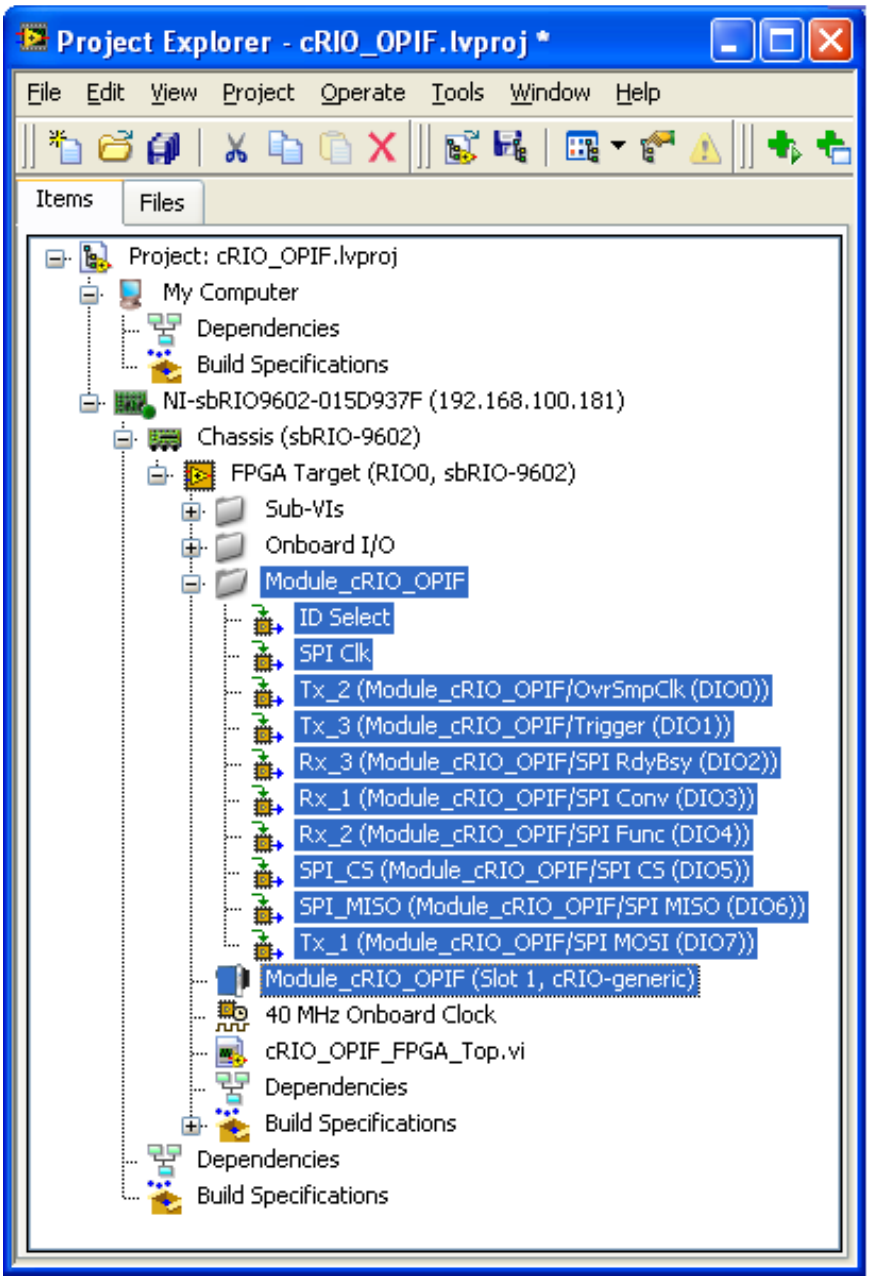
The simple access example can run only on the Compact-RIO FPGA and may be started directly from the host-PC. The example is included in the documentation CD: "**cRIO_OPIF_Basic_Access**"

The example shows three buttons to control the Tx outputs of the module and three indicators, which show the status of the receiver inputs. The following figure shows the front panel, where the receiver inputs are directly connected to the transmitter outputs via plastic fibre.

The following block diagram illustrates, how the IO lines of the module can be directly accessed in the example, using read/write controls to the user and FPGA IO nodes to the module's hardware.



The FPGA IO nodes have to be specified in the project as shown in the following figure:

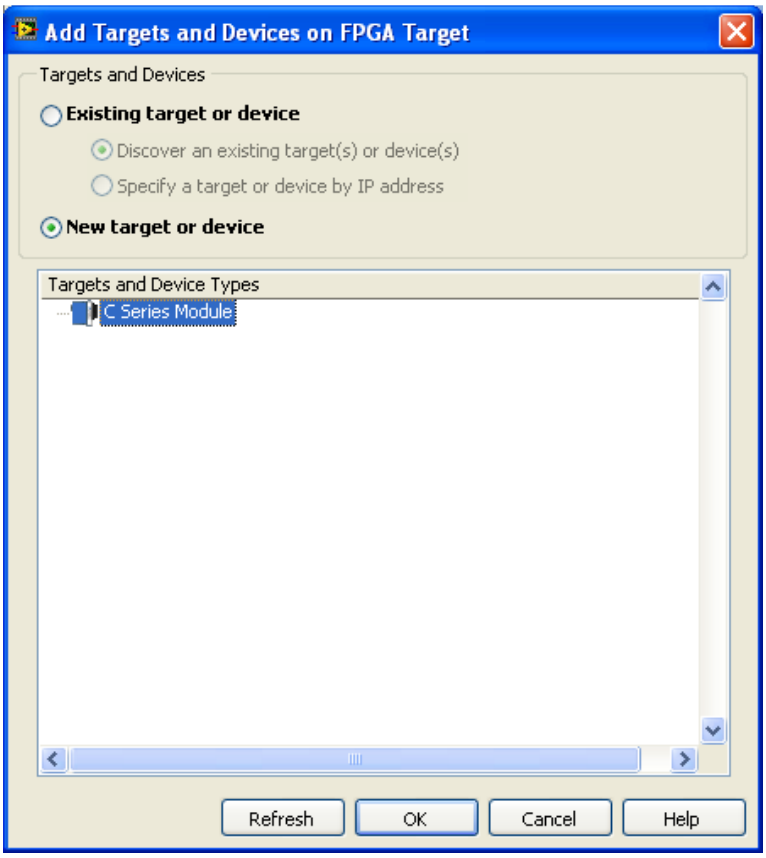


2.2 Including the module in the user application

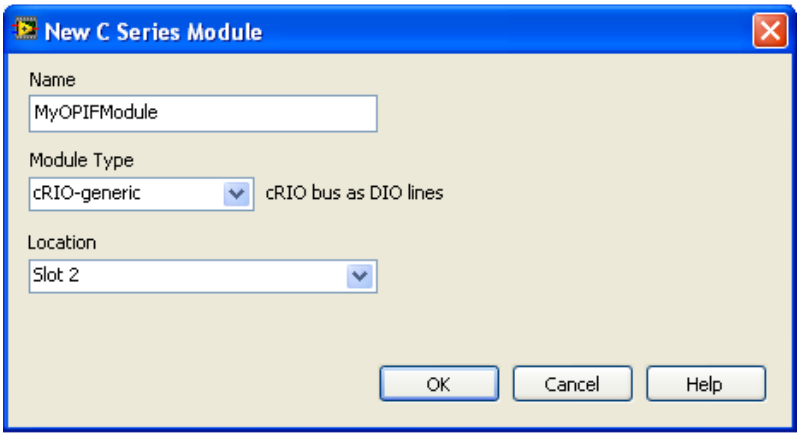
To include the module into the user application the blue marked items of the previous figure have to be copied into the LabVIEW project of the user. There are some steps to perform after you have setup your LabVIEW project according to your applied Compact-RIO system:

First include a new cRIO device:

- **Right-click** on "**FPGA Target**"
- Select from Popup- Menu "**New->C Series modules…**"
- Select the checkbox "**New Target or Device**" in the following window:
- Select C Series Module and press "**OK**"



The following window will appear.



Select

- an appropriate name for the module
- "**cRIO-generic**"
- the **Slot** of the Compact-RIO chassis, where the module should be applied.

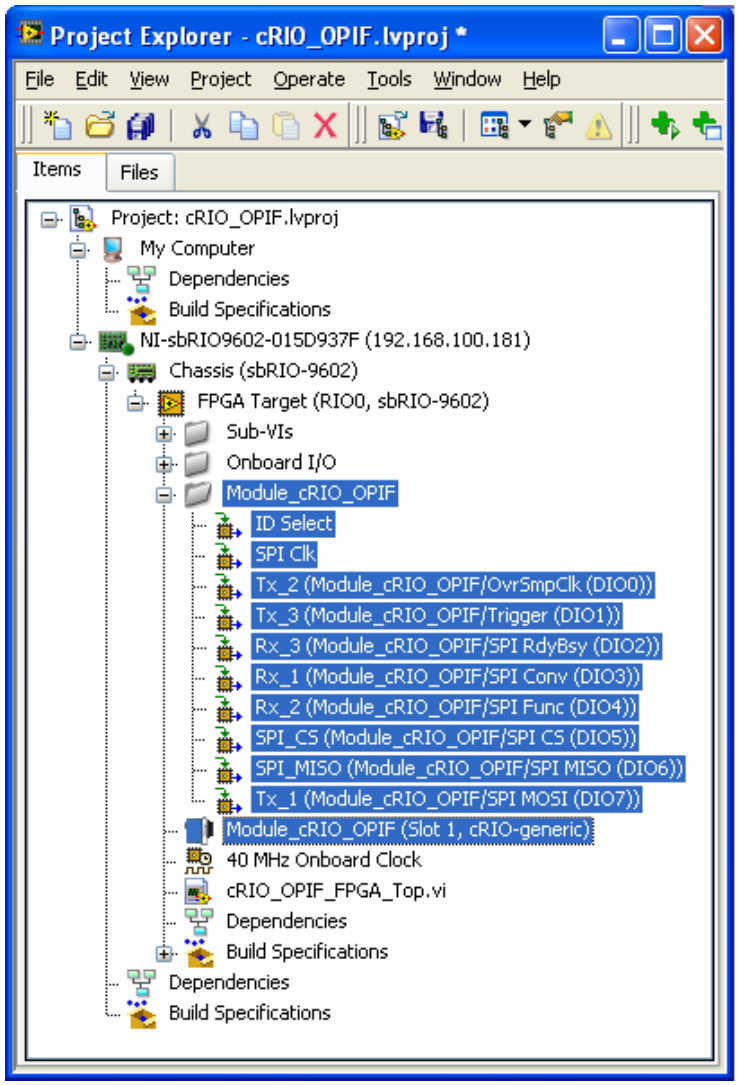
Note: If the cRIO-generic module can not be selected, please include the following line into your "LabVIEW.ini" file, while LabVIEW is closed.

```
cRIO_FavoriteBrand=generic
```

Rename the IOs of the module according to the following table, where the respective Rx (Rx_1, Rx_2, Rx_3) and Tx channel (Tx_1, Tx_2, Tx_3) must fit to the respective DIO-line (DIO0…DIO7), which is shown automatically after the C-series module has been included

OPIF Signal	C-Series Module DIO line
SPI_Clk	SPI_Clk
ID_Select	ID_Select
Tx_2	DIO0
Tx_3	DIO1
Rx_3	DIO2
Rx_1	DIO3
Rx_2	DIO4
SPI_CS	DIO5
SPI_MISO	DIO6
Tx_1	DIO7

The Project window will appear similar to the figure on the right:



Note, that you may also use different names of the signals.

If you want to apply several optical interface modules, you have to repeat the process as often as there are modules. In this case, the signal names of all modules must be unique in the whole project.

2.3 Streaming example with DMA

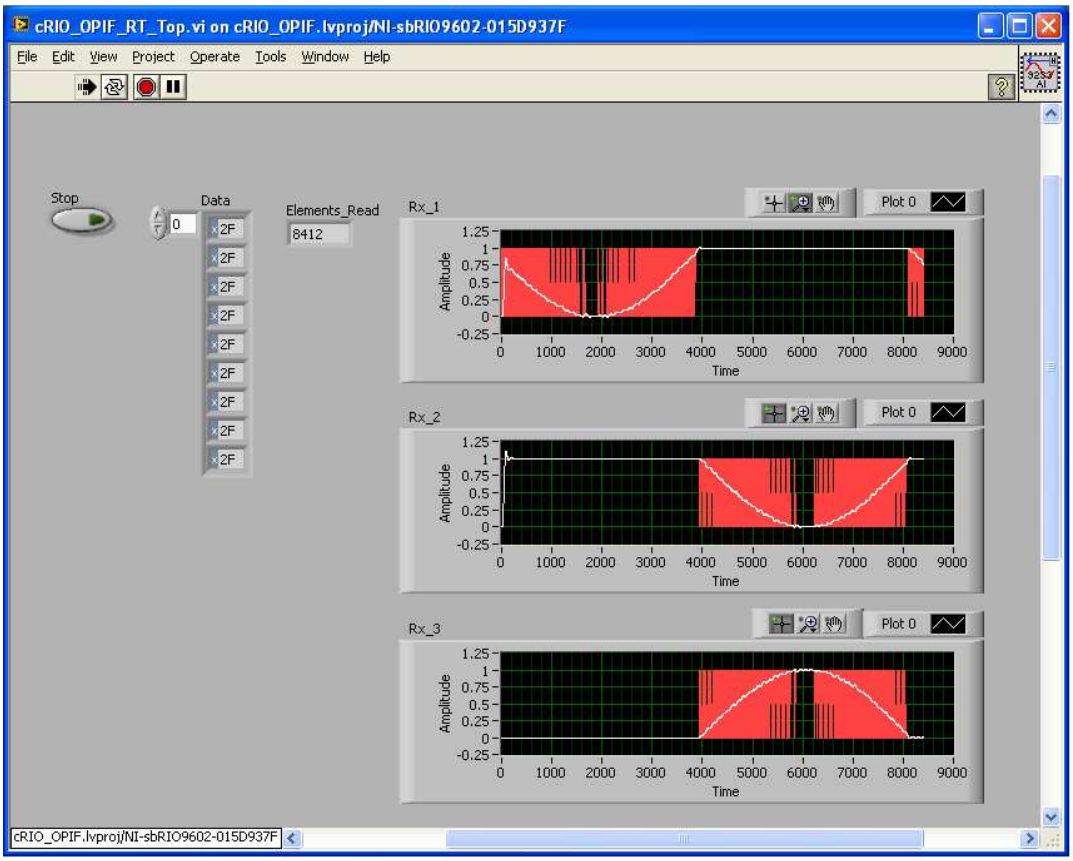
The following example generates a bit-stream in the FPGA, transmits the bit-stream on the optical transmitters, receives the data via the optical interconnected receivers and passes received data via DMA to the RT host software. The RT host software visualizes the received data.

The example is included in the documentation CD: "**cRIO_OPIF_Example_with_DMA**"

In detail the example performs the following:

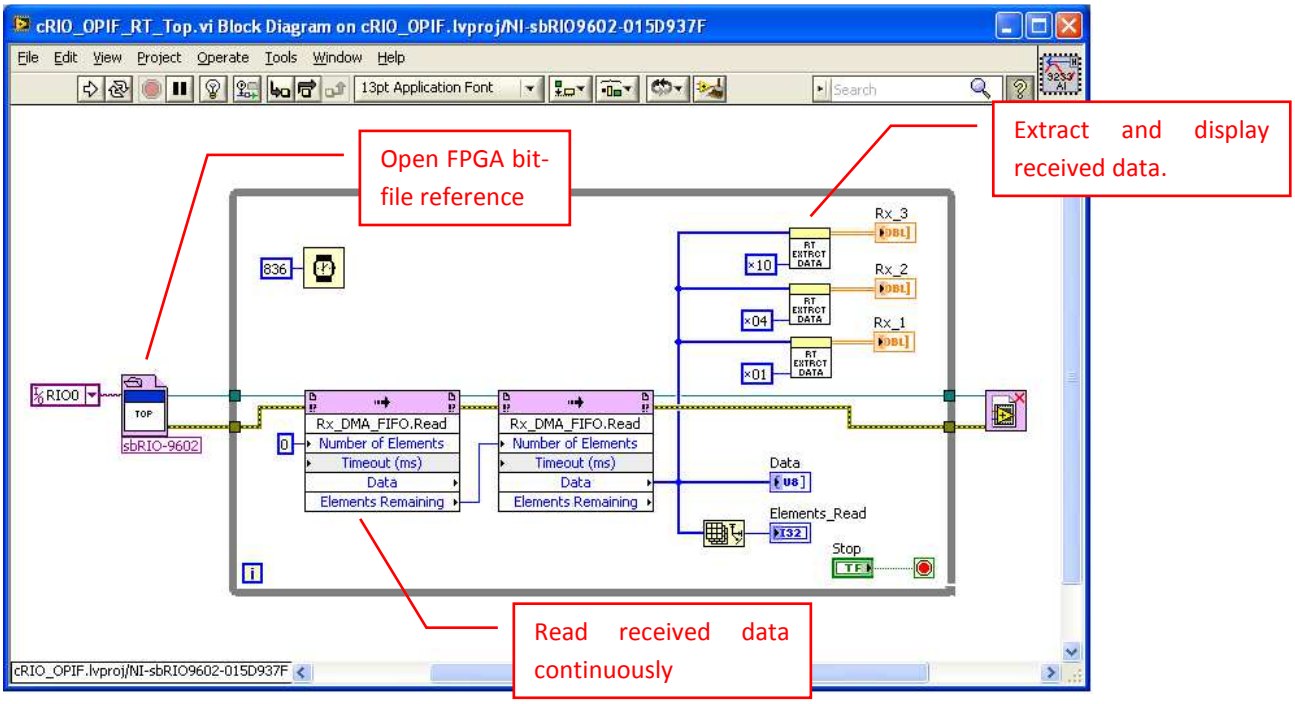
- The FPGA generates bit-streams for the transmitters continuously
- The bit-streams are three PWM-signals
    - The pulse width represents the elongation of a sine wave, resolved in 256 samples.
    - Every sample of the sine wave is represented by a single pulse - a single PWM-cycle.
    - The period of the PWM is 3.2768ms with a resolution of 100ns.
    - After every PWM cycle the next sample of the sine wave is valid.
    - The period of the sine wave is 32768*256*100ns -> approx. 1,2Hz sine wave frequency.
- The three bit-streams are passed to the module's transmitters
- Transmitters should be optically connected to the receivers.
- The received data is over-sampled at 40Mhz, where every fourth sample is valid. I.e. the real data sampling frequency is 10MHz, as it is the same as for the transmitters.
- The received data is passed to the host RT system via DMA. Actually only every 1000 samples one sample is transferred to reduce amount of data.
- In the RT software,

The following figure shows the example front panel of the RT software, which visualizes the received data as raw data, digital waveforms and an analogue representation after a filter:



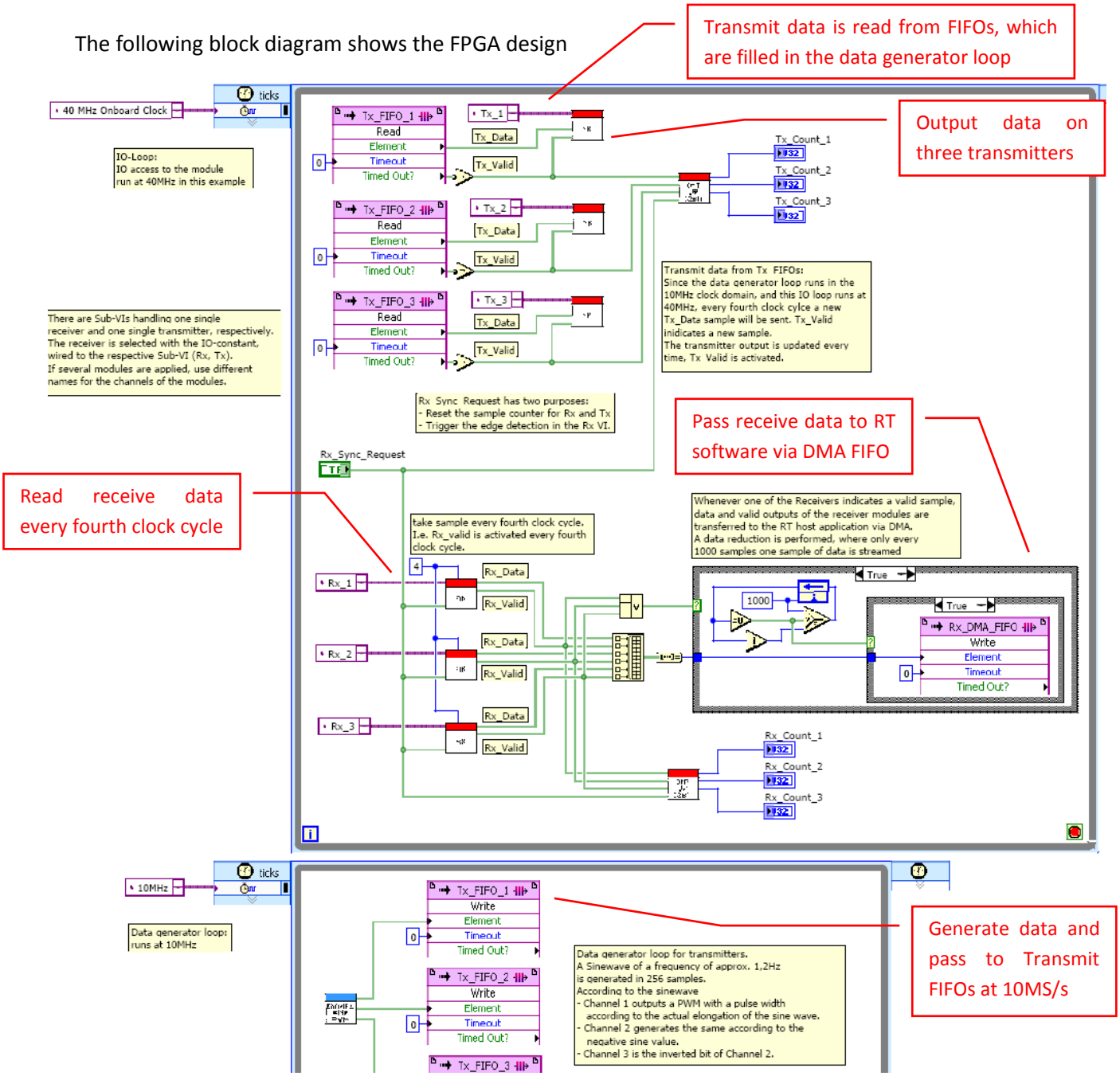
After the VI has started, both RT and the FPGA will automatically run and will generate data. If the optical transmitters and receivers are interconnected, the received data will be displayed as in the figure above.

Following block diagram shows the RT software:



In the RT software the reference to the FPGA-bit-file is opened. With a period of about 836ms the entire received data of approx. one sine wave period is read from the DMA FIFO. Received data bytes contain the bit-streams of all three receiver channels, which are extracted by the bitmasks (“0x01”, “0x04”, “0x10”). Take care in the RT software that the reference to the FPGA bit-file is a valid path to your bit-file on hard disk.

The following block diagram shows the FPGA design



The Tx and Rx Sub-Vis may be used by any user application for reading and transmitting data at different data rates. Note that this is just an example, where FIFO overflow or underflow is not handled. The example has to be adapted to the user’s requirements.

A detailed description of the software is included in the Vis.