
WRTCLK - A timestamp distribution adapter

User Manual

Bastian Löher, GSI

2023-10-05



Contents

1	Introduction	1
1.1	Overview	1
1.2	Inputs/outputs	2
1.2.1	Auxiliary signals	2
1.3	Configuration	3
1.3.1	Rataclock duty cycle (RES1)	3
1.3.2	Clock source (RES2/RES3)	3
1.3.3	LED display (RES4)	4
1.3.4	Timestamp source (RES5/RES6)	5
1.4	Serial connection (RES7/RES8)	5
1.4.1	Serial connection properties	5
1.4.2	Serial protocol	6
1.4.3	Companion script	7
1.5	Firmware update	8
1.6	WRTCLK/EXPLODER adapter boards	8
1.6.1	EXPLODER HS2 JTAG	8
1.6.2	HDMI IDC	8
1.6.3	RES adapter for WRTCLK	8
1.6.4	KiCAD design files	8
1.7	Development	9
1.7.1	Compilation	9
1.7.2	PROM file	9
1.7.3	Connecting FPGA programmer	10
1.7.4	Flashing the FPGA bitstream (until next power cycle)	11
1.7.5	Testing WRTCLK	11

1 Introduction

This documentation is for WRTCLK firmware version 0.6. Older firmware versions are not supported.

The most recent PDF version of this document is available [here](#).

An online version of the WRTCLK documentation is [here](#)

Information about the different firmware releases can be found in the [builds.txt](#) file.

1.1 Overview

WRTCLK is an FPGA firmware that can translate between different timestamp and clock distribution protocols.

The current version of the firmware is built for the EXPLODER3 hardware from GSI (experiment electronics department).

The following protocols are supported in the firmware

Protocol	Base frequency	Input	Output
BuTiS 2-wire (Schakel)	200 MHz	yes	yes (internal)
Ratatime	100 MHz	(deactivated)	yes
Rataclock for PLL	25 MHz	yes	yes
Rataclock for any FPGA	6.25 MHz	no	yes

The optical White Rabbit protocol is **not** supported directly. To synchronize WRTCLK with White Rabbit, a certified receiver must be used to convert to the BuTiS 2-wire (Schakel) protocol. Such receivers are for example the GSI VETAR2A or PEXARIA5.

1.2 Inputs/outputs

The figure below shows all signals that are accepted and generated by the WRTCLK firmware on the EXPLORDER3 hardware.

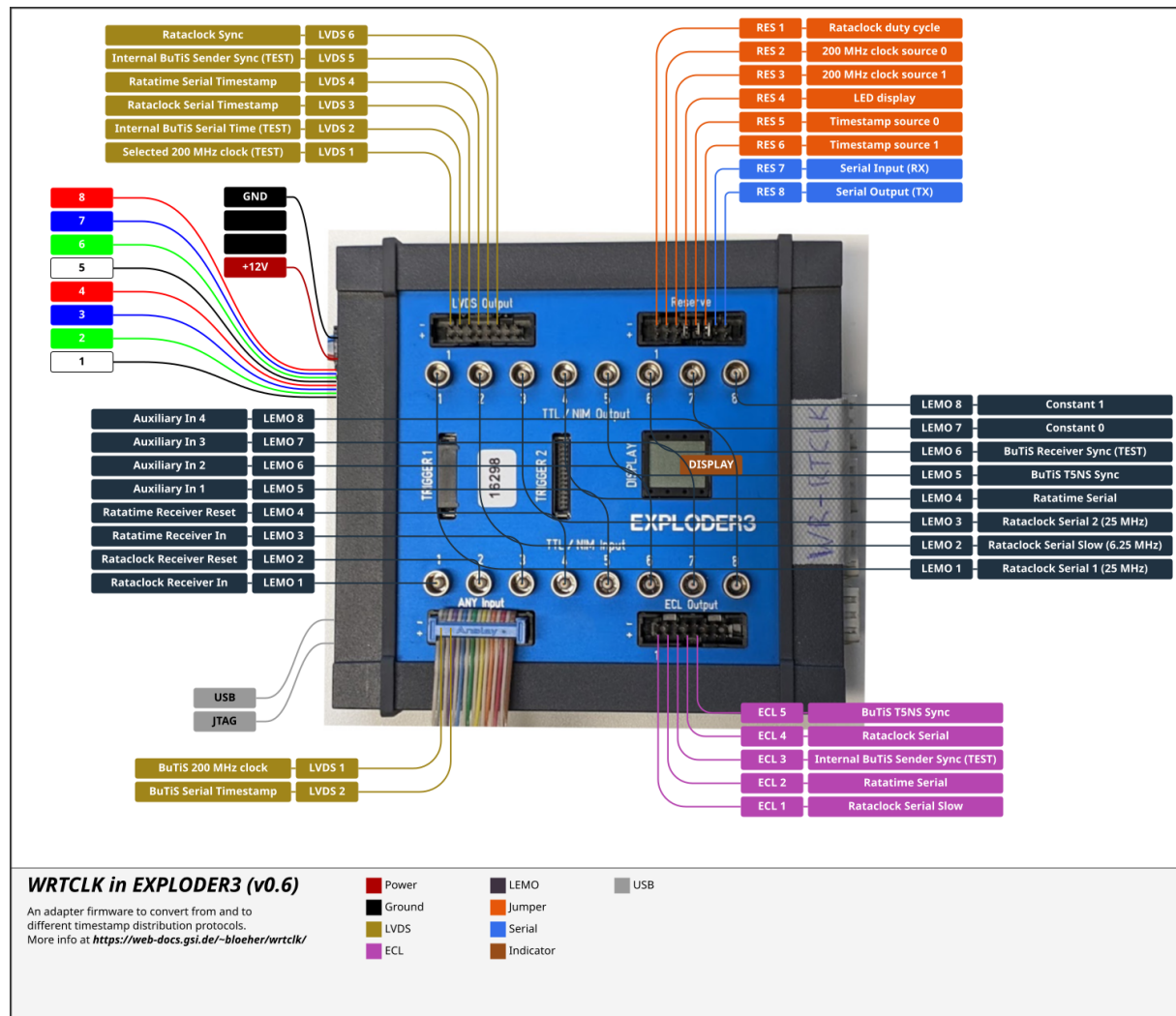


Figure 1.1: Overview of WRTCLK input/output signals on EXPLORDER3 hardware.

1.2.1 Auxiliary signals

LEMO inputs 5 to 8 can be used to send arbitrary (slowly changing) signals via the rataclock and ratime protocols. The current state of the inputs is sent as auxiliary signal bits [0..3]. All other auxiliary signal bits are set to 0.

1.3 Configuration

The configuration of WRTCLK can be changed using the 6 RES jumpers. Each jumper can be either in the **OFF / OPEN** or **ON / CLOSED** position. If the adapter board is used then the jumpers are replaced by small switches. In the following descriptions the **OFF / OPEN** position is marked as 0 and the **ON / CLOSED** position as 1.

1.3.1 Rataclock duty cycle (RES1)

The PLLs of rataclock receivers have somewhat strict requirements for the duty cycle of the received rataclock signal. However, the EXPLODER3 hardware differs slightly from one production run to the next, affecting the signal timing on the LEMO outputs. This variation in signal timing makes it impossible to produce a good rataclock output signal on all EXPLODER3 modules with a fixed duty cycle.

The RES1 jumper selects between two different settings for the rataclock output duty cycle. Check with a scope connected to the output, which setting produces the better signal.

A good rataclock signal is one, where the duty cycle of the short pulses is between 30% and 40%, and where the duty cycle of the long pulses is between 60% and 70%.

1.3.2 Clock source (RES2/RES3)

The main clock of WRTCLK runs at a frequency of 200 MHz. All other clocks are derived from this 200 MHz clock. The clock source switches (RES2/RES3) allow choosing between different sources for the main clock.

Setting	RES2	RES3	Clock source
0	0	0	Internal oscillator
1	1	0	Rataclock input (LEMO 1)
2	0	1	BuTiS input (Schakel) (LVDS 1)

Note: If a clock source other than the internal oscillator is selected, but no clock signal is applied at the respective input, then WRTCLK will not function properly.

1.3.3 LED display (RES4)

The 8 colored LEDs on the side of the EXPLODER3 case have different meaning depending on the RES4 switch setting.

RES4	LED mode
0	BuTiS status / Rataclock status
1	Timestamp counters

When RES4 is in **OFF/OPEN** position, information about the BuTiS input signals is displayed. When a BuTiS signal is connected, LED1 should toggle about 3 times per second. LED2 should light up green. LED3 should stay off. LED8 should toggle about once per second.

RES4	LED	Color	Meaning
0	1	white	BuTiS clock counter (bit 26)
	2	green	BuTiS receiver synced OK
	3	blue	BuTiS error
	4	red	Rataclock receiver status bit 0
	5	white	Rataclock receiver status bit 1
	6	green	Rataclock receiver sync lost bit 1
	7	blue	Rataclock receiver sync lost bit 2
	8	red	BuTiS receiver timestamp (bit 30)

When RES4 is in **ON/CLOSED** position, information about the internal timestamp counters is displayed. This can be used to check synchronicity between the different timestamps.

RES4	LED	Color	Meaning
1	1	white	internal timestamp bit 29
	2	green	internal timestamp bit 30
	3	blue	rataclock receiver timestamp bit 29
	4	red	rataclock receiver timestamp bit 30

RES4	LED	Color	Meaning
	5	white	BuTiS receiver timestamp bit 29
	6	green	BuTiS receiver timestamp bit 30
	7	blue	ratatime receiver timestamp bit 29
	8	red	ratatime receiver timestamp bit 30

1.3.4 Timestamp source (RES5/RES6)

The switches RES5 and RES6 allow to choose between several timestamp sources. This setting should normally follow the setting for the clock source, but it may be useful to make an independent selection.

Setting	RES5	RES6	Timestamp source
0	0	0	Internal rataclock sender
1	0	1	External rataclock receiver
2	1	0	External BuTiS receiver
3	1	1	External ratatime receiver

1.4 Serial connection (RES7/RES8)

WRTCLK implements a serial communication interface (UART) on the RES7 and RES8 pins. RES7+ is used as serial input (RX, receiver), and RES8+ as serial output (TX, transmitter). The RES7- and RES8- pins can be used as common ground.

Electrical signaling is similar to RS-232, so any standard USB-to-serial adapter can be attached to these pins and can then communicate with the WRTCLK firmware.

1.4.1 Serial connection properties

The UART uses these settings:

Property	Value
baud rate	115200 baud
stop bits	1
parity	none
byte size	8 bits

1.4.2 Serial protocol

WRTCLK continuously sends status updates in a fixed message format. Every message is 20 bytes long. The message format is described in the table below:

Byte	Meaning
0	“0”, fixed
1	“1”, fixed
2	bitmask of configuration jumpers (RES1-RES7)
3	firmware version, hexadecimal
4	bitmask of VHDL configuration bits
5	bits 7-2: BuTiS receiver timestamp bit 1: BuTiS synced state bit 0: BuTiS error
6	rataclock receiver status (3-5) / sync lost (0-2)
7	bits 0-3: PLL lock status
8	ratatime receiver status
9	LED status (BuTiS / rataclock)
10	received character 0
11	received character 1
12	received character 2
13	received character 3
14	“E”, fixed

Byte	Meaning
15	“x”, fixed
16	selected timestamp, bits 32-39
17	selected timestamp, bits 40-47
18	selected timestamp, bits 48-55
19	selected timestamp, bits 56-63

The VHDL configuration bits reflect which parts have been included in the WRTCLK firmware during build-time.

VHDL Config Bit	Meaning
0	Use ratatime sender
1	Use ratatime receiver
2	Use UART input
3	Use UART output
4	Use rataclock sender
5	Use BuTiS sender
6	Use BuTiS receiver
7	Use T5NS sender

The ‘received characters’ show the last 4 characters, that have been successfully received on the UART RX input.

1.4.3 Companion script

There exists a basic python script that allows to decode the data stream from WRTCLK. It is part of the full tarball mentioned below and can be found in the [test](#) directory.

```
1 ./test/receive.py
```

When executed, it connects to the serial port (one may need to adjust the port name in line 6 of the script), continuously decodes the received data and prints it on the screen.

1.5 Firmware update

The firmware may be updated either via the internal JTAG port (using a GSI PROMO-8 adapter) or the external JTAG port (micro-USB).

The easiest way is to use a Digilent HS2 programming adapter and a micro-USB cable to connect to the external JTAG port.

Use Xilinx Impact to connect with the adapter to JTAG and load the firmware into the on-board flash memory.

More details about the procedure are in Section [1.7.3](#).

1.6 WRTCLK/EXPLODER adapter boards

For convenience, some adapter PCBs have been designed.

1.6.1 EXPLODER HS2 JTAG

This is the adapter that allows plugging the Digilent HS2 adapter directly into the JTAG micro-USB port of the EXPLODER.

1.6.2 HDMI IDC

This adapter converts allows plugging the white rabbit / BuTiS signal from a PEXARIA or VETAR module directly into the ANY_IN input of the EXPLODER. It converts from an HDMI connector to the 16 pin IDC socket.

1.6.3 RES adapter for WRTCLK

The EXPLODER RES port is usually configured with jumpers that are placed / removed on the different pins. The RES adapter for WRTCLK provides 6 onboard switches to replace the jumpers and allows to connect a USB-UART serial adapter that connects to the RX/TX pins of the RES port. 3 sets of switches allow to set the pin mapping from the USB-UART adapter to the RES port.

1.6.4 KiCAD design files

Design files (KiCAD format) are available here for [download](#).

1.7 Development

WRTCLK is developed on Xilinx ISE Version 14.7. The FPGA part in use is not covered by the free Webpack edition of ISE, so a special licensed version is required to compile WRTCLK firmware.

A tarball of the most recent development state is available here for [download](#).

1.7.1 Compilation

Install and start ISE and then open the main project file `exploder_bl.xise`. Using the [Generate Programming File](#) button starts compilation and generation of the bitfile.

1.7.2 PROM file

In order to program also the flash memory in the EXPLODER it is necessary to prepare a PROM file (extension `.mcs`). This is done using the iMPACT program that is normally used to program the FPGA.

When iMPACT starts, choose to [Prepare a PROM file](#) from the initial menu.

In [Step 1](#) select [SPI Flash](#) -> [Configure Single FPGA](#) from the list and press the green arrow button.

In [Step 2](#) select [Storage Device \(bits\)](#) as 128M and press [Add Storage Device](#). Then press the next green arrow button.

In [Step 3](#) enter the [Output File Name](#) as `exploder_bl_prom.mcs` and extend the [Output File Location](#) field to point to the `build` directory.

Click the [OK](#) button.

In the message window click [OK](#).

In the file selection dialog choose the previously generated bitfile (`build/exploder_bl.bit`).

In the next message window (asking for another device file), select [No](#).

Confirm that device file entry is completed by selecting [OK](#).

In the process list ([iMPACT Processes](#)) on the left, double click on [Generate File...](#)

When the operation is successful, three files are generated with extensions `.mcs`, `.prm` and `.cfi`.

In the [iMPACT Flows](#) menu double click on [Boundary Scan](#). Now the PROM file can be flashed to the flash memory.

1.7.3 Connecting FPGA programmer

Several different programming cables can be used to program the EXPLODER. iMPACT supports at least the Xilinx Platform Cable 2 (red box) and the Digilent HS2. These have also been tested and are known to work well.

1.7.3.1 With PROMO adapter and internal port

When using the PROMO adapter, its dials must be set to these values: SELA1=1, SELZ1=8.

To use the internal port, the case must be opened by unscrewing the TX10 screws on the gray side-panels. Then the blue front panel and the two PCBs can be pushed out. This should be done very carefully to not break anything. Also, the exposed metal edges can be very sharp.

1.7.3.2 USB JTAG port

An adapter (cable) is needed from the programming adapter to the micro-USB port. Only a 5-wire USB cable may be used for this purpose (most cables only have 4 wires).

The pin mapping is shown in the table below:

Digilent HS2	USB wire	USB meaning	micro-USB pin
1 TMS	white	D-	2
2 TDO	red	VCC	1
3 TDI	green	D+	3
4 TCK	black	GND	5
5 GND	shield	-	-
6 VCC	yellow	ID	4

Using the EXPLODER HS2 JTAG adapter (see below) the Digilent HS2 programmer can be directly connected to the EXPLODER USB JTAG port.

1.7.4 Flashing the FPGA bitstream (until next power cycle)

1.7.4.1 With Xilinx iMPACT

Double click on [Boundary Scan](#) to show the programmable devices. The one with the part name [xc6slx150t](#) is the main FPGA of the EXPLODER.

Right-click on the device and click [Assign a new configuration file....](#)

Right-click again and choose [Program](#).

1.7.4.2 With openFPGAloader

Use this command:

```
1 openFPGAloader -c digilent_hs2 --index-chain 1 --fpga-part xc6slx150t
  exploder_bl.bit
```

1.7.4.3 Flashing the SPI memory (persistent)

The on-board flash memory is an SPI memory chip of type M25P128-VME6GB. The ID code as reported by iMPACT is 202018.

Programming the flash is currently only possible with iMPACT. There were attempts to implement flashing of the SPI memory with openFPGAloader, but this was unfortunately not successful. Contacts in this regard would be [Gwenhael Goavec](#) and [Uwe Bonnes](#).

Right click on the FPGA device and choose [Add SPI/BPI Flash...](#)

In the file dialog select the previously generated `.mcs` file.

In the next dialog select the M25P128 device. Leave the rest of the settings the same and click [OK](#).

Right click on the [FLASH](#) part attached to the FPGA and click on [Program](#).

Flashing the SPI memory takes considerably longer than flashing the FPGA only.

1.7.5 Testing WRTCLK

1.7.5.1 Standalone operation

Remove all jumpers (RES1-RES6) or set all switches to the [OPEN/OFF](#) position.

Now the internal oscillator is used as a time base.

With a scope, one can verify that the signals on the ECL/LVDS/LEMO outputs are the way they should be.

Using the serial adapter connected to RES7/RES8, one should be able to see the status updates emitted from WRTCLK.

1.7.5.2 Standalone loopback

Set RES4 to **OFF / OPEN**.

Connect “LVDS Output” 1 and 2 to “ANY Input” 1 and 2 to have a local loopback of the BuTiS time.

Observe: Leds 1 and 8 should start blinking, Led 2 should be constantly on.

With a scope check: - that the whiterabbit T0 pulses on LEMO output 6 and ECL output 3 are in sync. - LEMO output 5 shows the BuTiS T5NS / TZERO signal - LEMO output 1 shows the rataclock timestamp signal (from the sender with 25 MHz) - LEMO output 2 shows the rataclock timestamp signal (from the sender with 6.25 MHz)

Connect “LEMO output” 1 to “LEMO Input” 1 to have a local loop back for rataclock.

Observe: Leds 4 - 7 should turn on, display should turn green.

Connect “LEMO output” 8 to “LEMO Input” 2 for a short moment.

Observe: Leds 6 and 7 turn off, rataclock error flags are cleared.

1.7.5.3 Operation with upstream BuTiS

Set jumpers RES3 and RES6 or set these switches to the **CLOSED / ON** position.

Now the WRTCLK clock is driven by the external BuTiS (Schakel) signal.

The output signals should be the same as in standalone mode.

Also the status update from the serial monitor should work just the same.