CONV-TTL-BLO
User Guide

Carlos Gil Soriano
BE-CO-HT
carlos.gil.soriano@cern.ch

August 21, 2012

Abstract

This document describes the CONV-TTL-BLO board, a Blocking pulse repetitor board in double height VME format. It replaces all the following boards:

- 8 channel repeater
- 16 channel repeater
- CTDAC
- LA-BLO-TTL
- LAF-BLO-TTL
- LASB-TTL-BLO
- LA-GATE
- LA-TTL-BLO
- LAPF-TTL-BLO[1]

[1]For replacing this board a pulse width of 4us must be set.

Contents

List of Figures

List of Tables

1 Introduction

CONV-TTL-BLO is a board intended for replicating Blocking Pulses, offering six totally independent replication channels. The shape of the pulses is defined in [1]. CONV-TTL-BLO works together with two more boards: CONV-TTL-RTM and CONV-TTL-RTM-BLO.

CONV-TTL-BLO holds all the active circuitry and it is connected as a Front Module to a VME64 backplane. The ports offered in the Front Panel are explained in the next section.

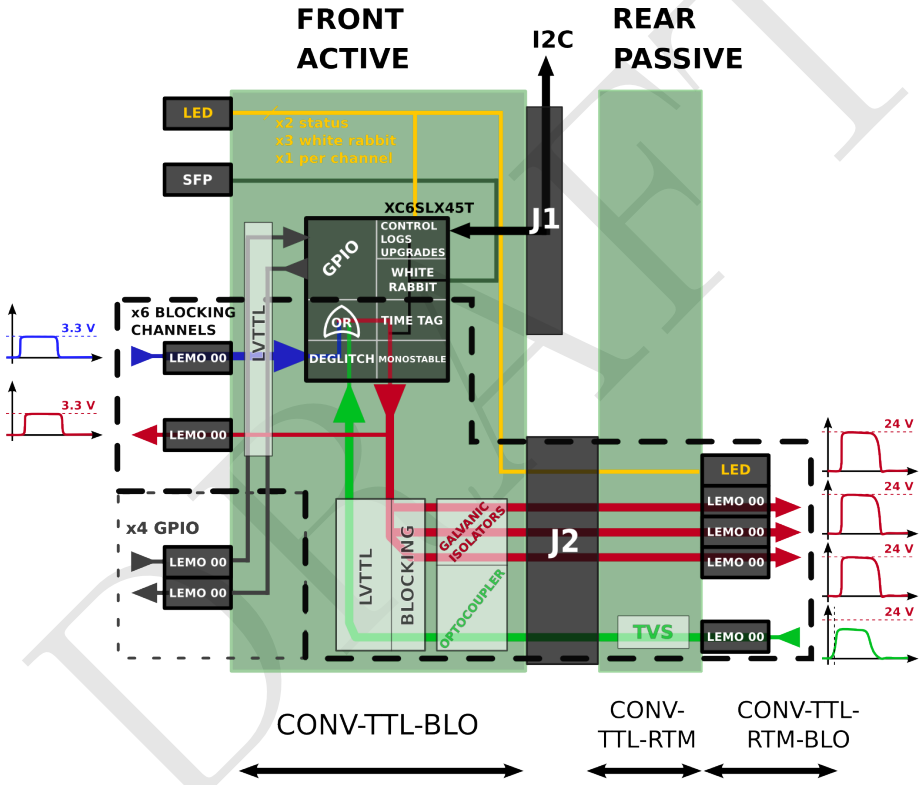

Figure 1: Pulse Repetition system

CONV-TTL-RTM and CONV-TTL-RTM-BLO are both connected to the rear part of the crate and provide, in the rear panel, the connectivity of the I/O Blocking lines. Every channel offers, in the Rear Panel, three Blocking Pulse outputs and one Blocking Pulse input.

CONV-TTL-RTM is a motherboard attached to the Rear Transition Module of the P2 VME64 connector. It connects CONV-TTL-BLO to

CONV-TTL-RTM-BLO and provides overvoltage protection for all the I/Os of all the channels.

CONV-TTL-RTM-BLO is a piggyback board mounted on CONV-TTL-RTM. It holds all the LEMO 00 connectors and channel LEDs that are offered in the Rear Panel.

Board	**Connection**	**Ports**
CONV-TTL-BLO	Front	SFP TTL Blocking triggers TTL Blocking output replica inverters
CONV-TTL-RTM	Back	-
CONV-TTL-RTM-BLO	Back	Blocking Pulse input Blocking Pulse outputs

Table 1: Boards for Blocking repetition

2 Board Ports

CONV-TTL-BLO consist of several ports, divided in three sections (from top to bottom as shown in Figure 2):

- SFP connector: [**A**].
- Blocking connectors: [**B**] and [**C**].
- General Purpose connectors: [**D**] and [**E**].

2.1 TTL-triggers

They correspond to the block [**B**] in Figure 2. The connectors are LEMO 00. By connecting an external trigger source to one of the connectors a pulse can be replicated, either in a Blocking Pulse level in the Rear Panel or in a TTL panel in the Front Panel.

Name	**Type**	**Level**	**Internal Termination**	**Location**
CH1 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]
CH2 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]
CH3 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]
CH4 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]
CH5 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]
CH6 IN	Pulsed	TTL or $\overline{TTL}$	50Ω Parallel	[B]

Table 2: TTL-triggers Front Panel inputs

2.2 TTL-level Blocking pulses replica

They correspond to the block [**C**] in Figure 2. The connectors are LEMO 00. From these connectors a TTL-level Blocking Pulse replica of the Rear Panel outputs if offered to the Front Panel. The pulse width of this output is similar to the pulse outputted in the Rear Panel; however, the rise time and top pulse level are obviously different from the Blocking output.
Once pulse is outputted, the LED of the corresponding channel blinks for 500 ms.
A summary of all the connector of this kind is found in Figure 3.

2.3 SFP connector

It is marked in Figure 2 as [**A**]. If the cable is plugged-in the socket, *White Rabbit* precise time-stamping can be obtained in CONV-TTL-BLO. Three LEDs above the connector show the status of the *White Rabbit* link.

Name	Type	Level	Internal Termination	Location
CH1 OUT	Pulsed	TTL	No	[C]
CH2 OUT	Pulsed	TTL	No	[C]
CH3 OUT	Pulsed	TTL	No	[C]
CH4 OUT	Pulsed	TTL	No	[C]
CH5 OUT	Pulsed	TTL	No	[C]
CH6 OUT	Pulsed	TTL	No	[C]

Table 3: TTL-level Blocking pulses Front Panel output

2.4 General pupose

In the lower part of the Front Panel ([**D**] and [**E**] in Figure 2) can be found four dedicated inverters. The output is a TTL inverted version of the TTL input.

Name	Type	Level	Internal Termination	Location
General Purpose Inputs				
INV A IN	Pulsed	TTL or $\overline{TTL}$	No	[D]
INV B IN	Pulsed	TTL or $\overline{TTL}$	No	[D]
INV C IN	Pulsed	TTL or $\overline{TTL}$	No	[D]
INV D IN	Pulsed	TTL or $\overline{TTL}$	No	[D]
General Purpose Inverted Outputs				
INV A OUT	Pulsed	$\overline{TTL}$ or TTL	No	[E]
INV B OUT	Pulsed	$\overline{TTL}$ or TTL	No	[E]
INV C OUT	Pulsed	$\overline{TTL}$ or TTL	No	[E]
INV D OUT	Pulsed	$\overline{TTL}$ or TTL	No	[E]

Table 4: General pupose Front Panel IOs

2.5 Front Panel

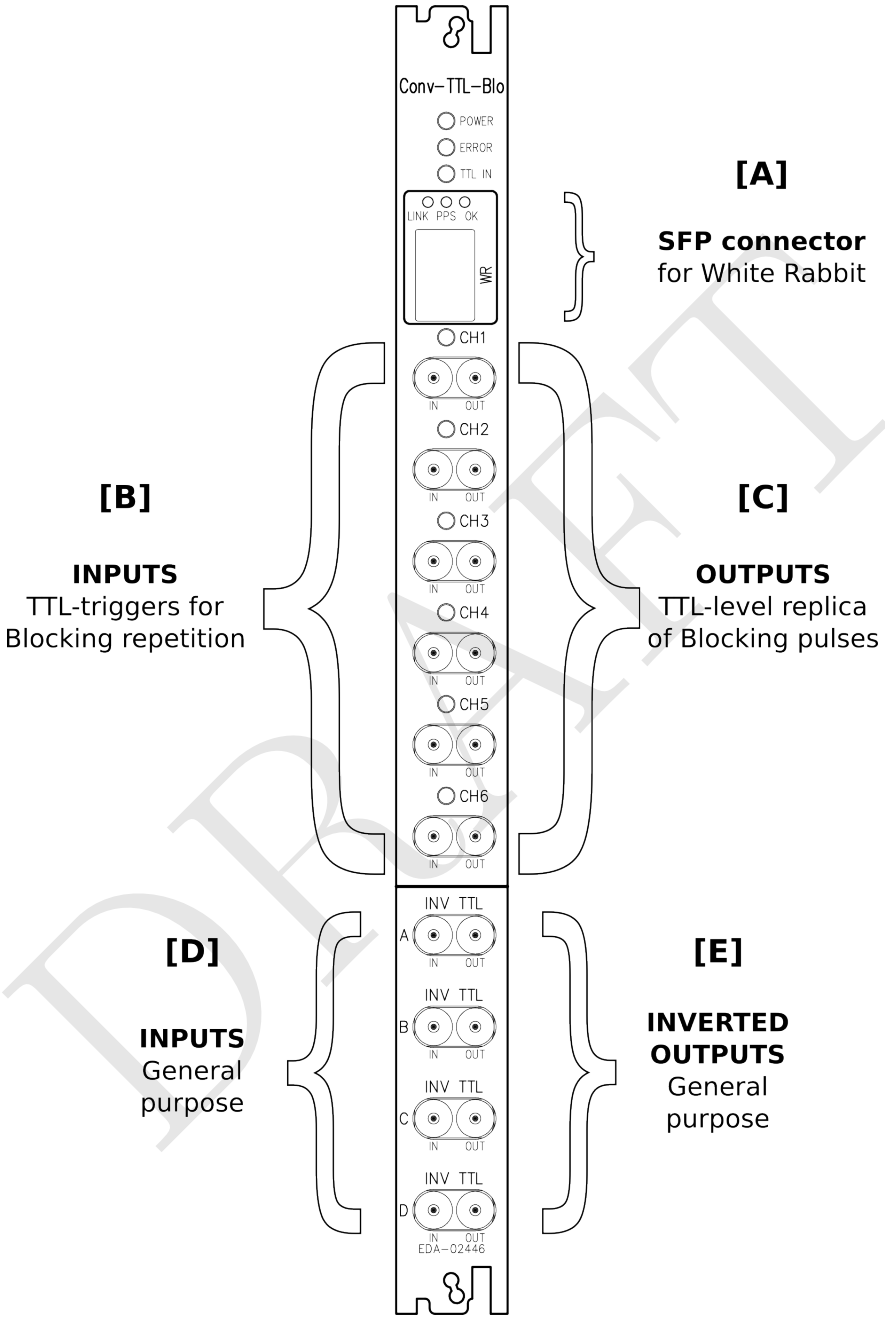

Figure 2: CONV-TTL-BLO Front Panel

3 Board Addressing

Communication with the FPGA is done via I2C interface through SERA and SERB pins in P1 VME64x connectors. In order to access one card inside a certain crate, it is necessary to set:

- An **I2C address**. Every CONV-TTL-BLO has an I2C address that equals the Geographical Address of the slot according to VME64x specifications. It is 7 bits width.
- An **internal CONV-TTL-BLO address**. It is a short integer (16 bits) in big endian format.

4 Functional Description

The task of CONV-TTL-BLO is to output a Blocking Pulse upon a reception of a trigger is received.

As stated before, CONV-TTL-BLO works together with CONV-TTL-RTM and CONV-TTL-RTM-BLO which are both in the rear part of the crate. The system formed by this three boards offers three independent channels for outputting Blocking Pulses. Refer to Figure 1 for a visual, whole-system, description.

CONV-TTL-BLO allows the user to set propierties for the pulse replication and receive information above the latest events logged in the board.

4.1 Trigger sources

There are two sources for triggering in every channel: one coming from the Front Panel (CONV-TTL-BLO) and other coming from the Rear Panel (CONV-TTL-RTM through CONV-TTL-RTM-BLO).

The trigger policy is that *a Blocking pulse will be outputted whenever either a trigger in the Front Panel or the Rear Panel is detected*:

Trigger	**Board**	**Connection**
TTL	CONV-TTL-BLO	Front Panel
Blocking	CONV-TTL-RTM-BLO	Rear Panel

Global trigger is OR function of the two sources above

Table 5: Trigger sources

5 Memory Map

Channel	Address	Data type	Name	Comment
Access	0x0101	Bit pattern	CTR0_ACCESS	Control device access 0
	0x0102	Bit pattern	ACTR1	Control device access 1
I2C	0x0202	Bit pattern	CTR0_I2C	I2C control 0
	0x0203	Bit pattern	I2C_CTR1	I2C control 1
Ch 1	0x0301	Bit pattern	CTR0_CH1	Channel 1 control
	0x0302	Integers	CTR1_CH1	Channel 1 min/max Pulse length
	0x0305	Bit pattern	RAM0_CH1	RAM control Channel 1
	0x0306	Bit pattern	RAM1_CH1	Current read/write Address pointers
	0x0307	Bit pattern	RAM2_CH1	Start/end read address
Ch 2	0x0401	Bit pattern	CTR0_CH2	Channel control
	0x0402	Integers	CTR1_CH2	Channel 2 min/max Pulse length
	0x0405	Bit pattern	RAM0_CH2	RAM control Channel 2
	0x0406	Bit pattern	RAM1_CH2	Current read/write Address pointers
	0x0407	Bit pattern	RAM2_CH2	Start/end read address
Ch 3	0x0501	Bit pattern	CTR0_CH3	Channel 3 control
	0x0502	Integers	CTR1_CH3	Channel 3 min/max Pulse length
	0x0505	Bit pattern	RAM0_CH3	RAM control Channel 3
	0x0506	Bit pattern	RAM1_CH3	Current read/write Address pointers
	0x0507	Bit pattern	RAM2_CH3	Start/end read address

Table 6: CONV-TTL-BLO Memory map I/II

Channel	Address	Data type	Name	Comment
Ch 4	0x0601	Bit pattern	CTR0_CH4	Channel 4 control
	0x0602	Integers	CTR1_CH4	Channel 4 min/max Pulse length
	0x0605	Bit pattern	RAM0_CH4	RAM control Channel 4
	0x0606	Bit pattern	RAM1_CH4	Current read/write Address pointers
	0x0607	Bit pattern	RAM2_CH4	Start/end read address
Ch 5	0x0701	Bit pattern	CTR0_CH5	Channel 5 control
	0x0702	Integers	CTR1_CH5	Channel 5 min/max Pulse length
	0x0705	Bit pattern	RAM0_CH5	RAM control Channel 5
	0x0706	Bit pattern	RAM1_CH5	Current read/write Address pointers
	0x0707	Bit pattern	RAM2_CH5	Start/end read address
Ch 6	0x0801	Bit pattern	CTR0_CH6	Channel 6 control
	0x0802	Integers	CTR1_CH6	Channel 6 min/max Pulse length
	0x0805	Bit pattern	RAM0_CH6	RAM control Channel 6
	0x0806	Bit pattern	RAM1_CH6	Current read/write Address pointers
	0x0807	Bit pattern	RAM2_CH6	Start/end read address
Multiboot	0x0901	Bit pattern	CTRL0	Multiboot operation control
	0x0904	Integer	GENERAL1	Lower bits 1^{st} bitstream address
	0x0905	Bit pattern	GENERAL2	Lower bits 1^{st} bitstream address
	0x0906	Integer	GENERAL3	Higher bits 2^{nd} bitstream address
	0x0907	Bit pattern	GENERAL4	Higher bits 2^{nd} bitstream address
EEPROM				
White Rabbit				

Table 7: CONV-TTL-BLO Memory map II/II

6 Register Description

6.1 Access Registers

6.1.1 CTR0_ACCESS

The *CTR0* register is a read-write register that enable and disable access to Wishbone devices in the memory map.

CTR0_ACCESS		
Bits	**Field**	**Meaning**
0	EN0	ENable access to device 0 in memory map
1	EN1	ENable access to device 1 in memory map
2	EN2	ENable access to device 2 in memory map
3	EN3	ENable access to device 3 in memory map
4	EN4	ENable access to device 4 in memory map
5	EN5	ENable access to device 5 in memory map
6	EN6	ENable access to device 6 in memory map
7	EN7	ENable access to device 7 in memory map
8	EN8	ENable access to device 8 in memory map
9	EN9	ENable access to device 9 in memory map
10	EN10	ENable access to device 10 in memory map
11	EN11	ENable access to device 11 in memory map
31-13	x	Reserved

Table 8: CTR0_ACCESS Register

6.1.2 CTR1_ACCESS

The *CTR1* register is a read-write register that enable and disable read and write permissions to Wishbone devices in the memory map.

CTR1_ACCESS		
Bits	**Field**	**Meaning**
0	GWR	General write permissions
1	GRD	General read permissions
2	WR0	WRite permissions in memory map 0
3	RD0	ReaD permissions in memory map 0
4	WR1	WRite permissions in memory map 1
5	RD1	ReaD permissions in memory map 1
6	WR2	WRite permissions in memory map 2
7	RD2	ReaD permissions in memory map 2
8	WR3	WRite permissions in memory map 3
9	RD3	ReaD permissions in memory map 3
10	WR4	WRite permissions in memory map 4
11	RD4	ReaD permissions in memory map 4
12	WR5	WRite permissions in memory map 5
13	RD5	ReaD permissions in memory map 5
14	WR6	WRite permissions in memory map 6
15	RD6	ReaD permissions in memory map 6
16	WR7	WRite permissions in memory map 7
17	RD7	ReaD permissions in memory map 7
18	WR8	WRite permissions in memory map 8
19	RD8	ReaD permissions in memory map 8
20	WR9	WRite permissions in memory map 9
21	RD9	ReaD permissions in memory map 9
22	WR10	WRite permissions in memory map 10
23	RD10	ReaD permissions in memory map 10
24	WR11	WRite permissions in memory map 11
25	RD11	ReaD permissions in memory map 11
31-26	x	Reserved

Table 9: CTR1_ACCESS Register

6.2 I2C Registers

6.2.1 CTR0_I2C

The CTR0 register is a write-read register. It controls the indirect addres and holds the I2C address (which in the case of *CONV-TTL-BLO* will be connected to VME64x geographical address pins).

CTR0_I2C		
Bits	**Field**	**Meaning**
0	EN	general ENable
1	RST	general ReSeT
2	PEN	Prescaler ENable
5-3	x	Reserved
7-6	BIA	Bytes Indirect Addressing
14-8	A[6:0]	I2C address
15	x	Reserved
31-16	-	Not used

Table 10: CTR0_I2C Register

6.2.2 CTR1_I2C

The CTR1 register is a write-read register. It shows the fsm of the separate *read* and *write* fsms.

CTR1_I2C		
Bits	**Field**	**Meaning**
7-0	RDS	fsm status: ReaD Status
15-8	WDS	fsm status: WRite Status
31-16	-	Not used

Table 11: CTR1_I2C Register

6.3 Channel Registers

6.3.1 CTR0_CH[x]

The CTR0_CH[x] register is a read-write register. It allows setting up the basic configuration of the trigger HDL core, its RAM blocks and both the deglitching mask and output pulse length to be used.

CTR0_CH[x]		
Bits	**Field**	**Meaning**
0	EN	General ENable
1	CLR	General CLeaR
3-2	x	Reserved
4	EN_TT	Enable Time-Tagging
5	CLR_TT	Clear Time-Tagging
7-6	RDM	time-tagging ReaD Mode
15-8	CGM	Current Glitch Mask
31-16	CPL	Current Pulse Length

Table 12: CTR0_CH[x] Register

6.3.2 CTR1_CH[x]

The CTR1_CH[x] register is a read-write register. It allows setting up the boundaries that override invalid values of CPL field in CTR0.

CTR1_CH[x]		
Bits	**Field**	**Meaning**
15-0	MinPL	Minimum Pulse Length
31-16	MaxPL	Maximum Pulse Length

Table 13: CTR1_CH[x] Register

6.3.3 RAM0_CH[x]

The RAM0 register is a read-write register. It allows setting up the RAM and the read request to it.

RAM0_CH[x]		
Bits	**Field**	**Meaning**
0	EN_TT	Enable Time-Tagging
1	CLR_TT	Clear Time-Tagging
3-2	RDM	time-tagging ReaD mode
4	EMPTY	RAM empty
5	FULL	RAM full
6	WA	RAM Wrapped Around
7	RQT	ReQuesT read
31-8	x	Reserved

Table 14: RAM0_CH[x] Register

6.3.4 RAM1_CH[x]

The RAM1 register is a read-only register. It shows the current read and write address configured to be accessed.

RAM1_CH[x]		
Bits	**Field**	**Meaning**
15-0	CRA	Current Read Address
31-16	CWA	Current Write Address

Table 15: RAM1_CH[x] Register

6.3.5 RAM2_CH[x]

The RAM2 register is a read-write register. It allows setting up the RAM adress range to be read.

RAM2_CH[x]		
Bits	**Field**	**Meaning**
15-0	SA	Starting read Address
31-16	EA	Ending read Address

Table 16: RAM2_CH[x] Register

6.4 Multiboot Registers

6.5 White Rabbit Registers

7 Installation

References

[1] C. Gil Soriano. Standard Blocking Output Signal Definition for CTDAH board, September 2011. http://www.ohwr.org/documents/109.