CONV-TTL-BLO
User Guide

Theodor-Adrian Stana
Carlos Gil Soriano
BE-CO-HT

February 26, 2013

Abstract

This document describes the CONV-TTL-BLO board, a Blocking pulse repeater 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

List of abbreviations

- RTM – Rear Transition Module
- RTMM – RTM Motherboard
- RTMP – RTM Piggiback

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.

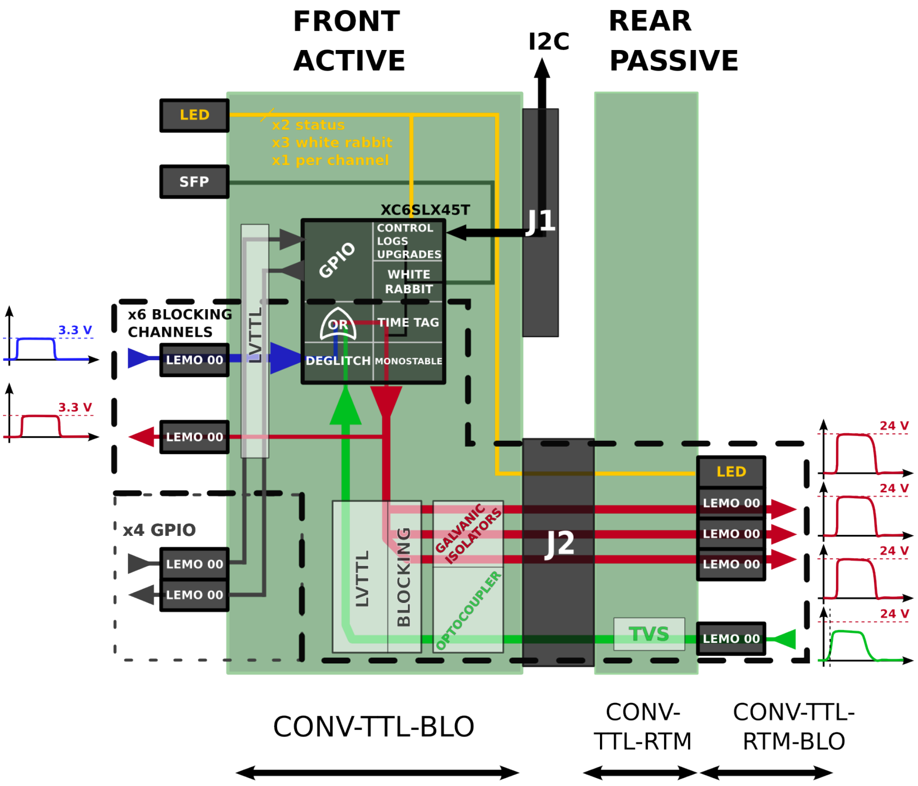

Figure 1: Pulse Repetition system

CONV-TTL-BLO contains all the active circuitry and it is connected as a Front Module to a VME64 backplane. 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 (RTM) 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 contains 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

LED	Description
PW	Power LED. Lights *green* when a valid CONV-TTL-BLO firmware is loaded to the FPGA.
ERR	Error LED. Lights *red* when no rear transition module board is present.
TTL_N	Negated-TTL status LED. Lights *green* when negated TTL logic is selected via the 8^{th} position of the on-board selection switch.
I2C	I^2C status LED. Lights *red* until an I^2C transfer has taken place. Once either a read or a write is successfully completed, the I^2C status LED lights *green* to signal the communication is up.

Table 2: Status LEDs on CONV-TTL-BLO front panels

2 Main Board Front Panel

The front panel of CONV-TTL-BLO boards is shown in Figure 2. It consists of status LEDs and several ports, divided in three sections from top to bottom:

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

2.1 Status LEDs

In the current version of the CONV-TTL-BLO boards, only several of the status LEDs present on the board are used, due to limited firmware support in the FPGA. The implemented status LEDs are presented in Table 2. Unimplemented status LEDs are off by default.

2.2 SFP connector

This connector is used to add White Rabbit support to the CONV-TTL-BLO boards. If an optic fibre cable is connected to this socket, White Rabbit precise time-stamping can be added to CONV-TTL-BLO. Three LEDs above the connector are provisioned to show the status of the White Rabbit link.

White Rabbit is currently not supported in the CONV-TTL-BLO firmware.

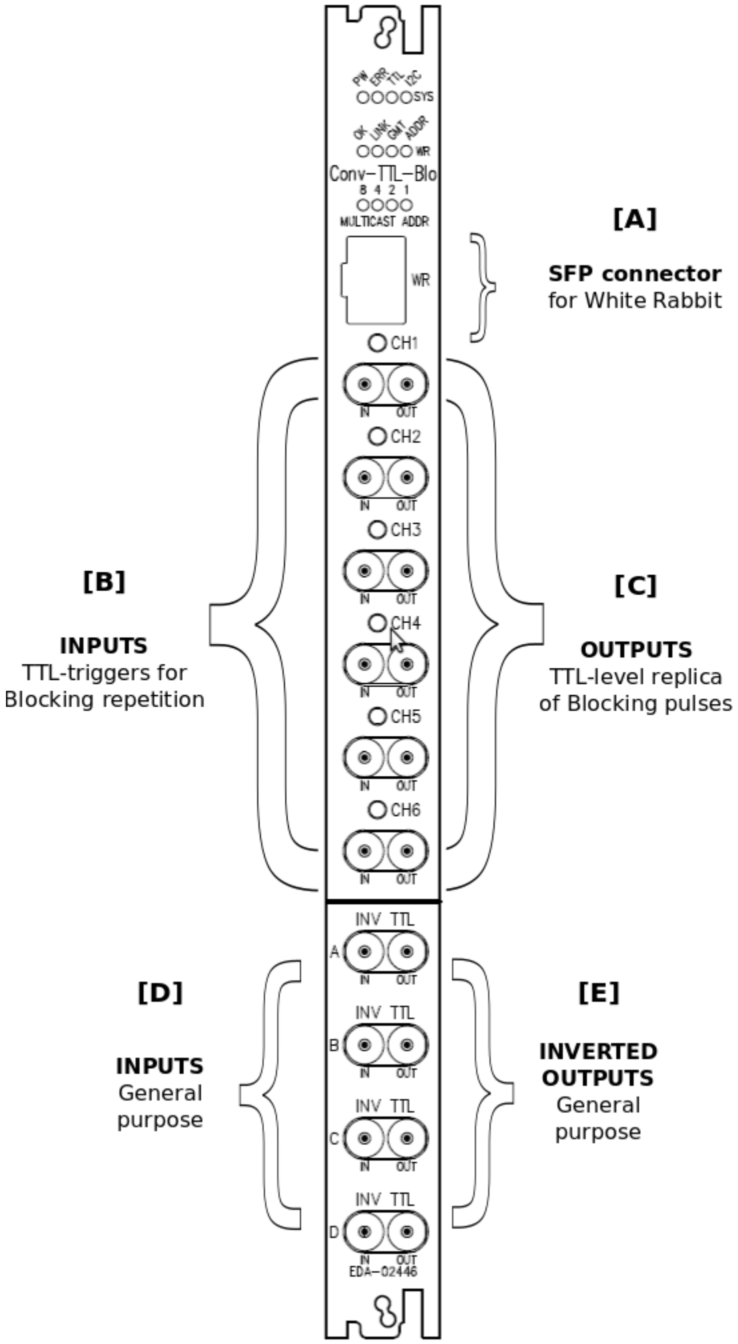

Figure 2: CONV-TTL-BLO Front Panel

2.3 TTL triggers

The TTL triggers correspond to block [**B**] in Figure 2. The connectors are LEMO 00 (type EPY). By connecting an external trigger source to one of the connectors a pulse is replicated in a Blocking Pulse level in the Rear Panel and in a TTL panel in the Front Panel. All input channels are 50Ω-terminated.

2.4 Repeated TTL pulses

These correspond to block [**C**] in Figure 2. The connectors are LEMO 00 (type EPA). 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; the rise time and top pulse level are however different from the Blocking output.

When the pulse is output, the LED of the corresponding channel blinks for 125 ms.

The TTL output lines are not internally terminated.

2.5 General pupose

Four dedicated inverters can be found in the lower part of the Front Panel ([**D**] and [**E**] in Figure 2). The output is a TTL inverted version of the TTL input. The inverted-TTL outputs are not internally terminated.

3 Getting Started

This section provides a reference to testing the CONV-TTL-BLO boards for basic functionality. The steps listed below were run on a Linux Ubuntu 12.04 unit connected to the wired Ethernet interface on the CERN network. The steps to follow should be similar on any Linux or Windows machine; some details such as ELMA crate IP or TELNET client escape characters may differ in the reader's case.

The following steps should be performed for testing board functionality:

1. Plug in a front module card to the ELMA crate. Turn on power to the crate and program the Spartan-6 FPGA.

2. Check that the *PW* LED lights *green* and the *ERR* and *I2C* LEDs light *red*. The *TTL_N* LED may also be lit. If it is and the LED is *green*, this is not an issue.

3. Connect to an ELMA crate via TELNET, using the following command on the command line:

   ```
   $ telnet 137.138.192.90
   Trying 137.138.192.90...
   Connected to 137.138.192.90.
   Escape character is ’^]’.
   login:user
   password USER
   %>
   ```

4. If the TELNET access is successful, the user should now be presented with a command line to the SysMon board. To test basic SysMon functionality, run a `voltage` command on the SysMon, which should output the following:

   ```
   %>voltage


   ----------------------------Sensor List---------------------------

   --no--Name-------------Type----Value--Unit---State------------------

   *   2 +3.3V            Thr     3.31   V      Ok
   *   3 +5V              Thr     5.01   V      Ok
   *   4 +12V             Thr     12.09  V      Ok
   *   5 -12V             Thr     -12.38 V      Ok
   ```

5. It can now be proceeded to reading a register from the CONV-TTL-BLO boards. The `readreg` command can be used for this purpose. Assuming a CONV-TTL-BLO board in VME slot 1, reading register I2C_CTR0 at address 0x40 (see Sec. 5) is done as follows:

```
%>readreg 1 17
 Read Data: 004042BC
```

6. The output of the command should be as above, yielding the default value of the I2C_CTR0 register. The *I2C* LED on the front panel should also be lit *green* now. It can now be proceeded to checking the pulse repetition mechanisms.

7. First, connect one end of a cable with LEMO 00 connectors at both ends to the front module front panel TTL input ports.

8. Configure a pulse generator to output TTL level pulses (*max. 5V*) at a frequency of about 3 Hz with a pulse length of ≥1 μs.

9. Connect the other end of the cable to the pulse generator. The LED of the corresponding channel should light for 125 ms when a pulse arrives.

10. Connect the corresponding channel output port to an oscilloscope and measure the signal level of the pulse and that the pulse width is 1 μs long.

11. Repeat the operation with the other front channels.

12. Connect a rear transition module (RTM) board to the back-plane of the ELMA, on the same VME slot as the front module. The *ERR* LED should turn off.

13. Connect the LEMO cable to each channel on the front module front panel in turn to each of the channels. As with the front module front panel, the LED corresponding to the connected channel on the rear transition module piggyback (RTMP) front panel should be lit for about 125 ms when a pulse arrives. Measure that the output pulse on the blocking channels is 1 μs long and 24 V in amplitude.

14. Disconnect the LEMO cable from the front module front panel and configure the pulse generator for 15 V pulse amplitude.

15. Connect the LEMO cable to channel 1 the front panel of the RTMP board. Measure the output pulse on the corresponding channel of the RTMP front panel; it should be 1 μs long and 24 V in amplitude.

16. Finally, measure on the front panel of the front module that on channel 1 the output pulse is 1 μs long and 3.3 V.

17. Repeat for all other channels.

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 in the rear part of the crate. The system formed by these three boards offers three independent channels for outputting Blocking Pulses. Refer to Figure 1 for a visual, whole-system, description.

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 3: Trigger sources

All the control logic of the system is implemented in a Xilinx Spartan-6 FPGA. Apart from pulse repetition, an I^{2}C to Wishbone bridge is also implemented. This module translates I^{2}C accesses as presented in accesses to internal memory-mapped registers. The status of the various components in the system can in this way be checked and controlled.

5 Accessing internal registers

5.1 ELMA crates

CONV-TTL-BLO boards have been designed to operate in ELMA crates. These crates provide a back plane with VME64x connectors which boards can be plugged into. A dedicated board inside the ELMA crates called the SysMon (System Monitor) monitors overall system status and provides access to boards plugged into the VME back plane.

The user can connect to SysMon boards either through a simple RS-232 interface, or through Telnet. In order to send commands to a board plugged into an ELMA crate, the user would connect to the SysMon over one of these two interfaces and send `readreg` and `writereg` commands to board-specific registers, as exemplified in Section 3 to control their functioning.

5.2 Board Addressing

Communication with the CONV-TTL-BLO FPGA is done via I^2C interface through the SERA and SERB pins in P1 VME64x connectors. In order to access a CONV-TTL-BLO board, it is necessary to send:

- The board's 7-bit I^2C address. Every CONV-TTL-BLO has an address that prepends two bits of value *10* to the Geographical Address of the slot according to VME64x specifications.
- An internal CONV-TTL-BLO register address. It is a 16-bit integer in *little endian* format (most significant byte is sent first).

After this, four bytes of data are read/written from/to the internal CONV-TTL-BLO register. These four bytes of data are written in *big endian* format (least significant byte is sent first).

The addressing protocol is thoroughly described in [2]. The SysMon acts as a master on the I^2C interface and has the protocol implemented as software on the on-board processor. The FPGAs on CONV-TTL-BLO boards implement an I^2C slave and decode the data streams sent by the SysMon.

5.3 CONV-TTL-BLO memory map

Table 4 summarizes the registers mapping in the current version of the CONV-TTL-BLO firmware. The first column represents the Wishbone address internal to the FPGA firmware and the second column represents the ELMA register address (in `readreg` and `writereg` commands).

Address	ELMA	Name	Access	Description
0x00	1	STAT_L	R	Lower 32 bits of system status register
0x04	2	STAT_H	R	Upper 32 bits of system status register
0x40	17	I2C_CTR0	R	I^2C control register
0x44	18	I2C_LT	R	I^2C line timing register, provides the current I^2C line speed
0x48	19	I2C_DTX	R/W	Data to transmit through the I^2C interface
0x4C	20	I2C_DRXA	R	Lower 32 bits of data received through the I^2C interface
0x50	21	I2C_DRXB	R	Upper 32 bits of data received through the I^2C interface

Table 4: Memory map of the CONV-TTL-BLO design

5.4 Register description

5.4.1 STAT_L

Bit	Field	Reset	Description
31..0	IDENT_L	0x00000000	Lower 32 bits of board identity, as provided by Maxim DS18B20U+ thermometer

5.4.2 STAT_H

Bit	Field	Reset	Description
15..0	IDENT_H	0x0000	Upper 32 bits of board identity, as provided by Maxim DS18B20U+ thermometer
18..16	RTMM	”000”	Rear transition module mainboard (RTMM) identification [3]
21..19	RTMP	”000”	Rear transition module piggyback (RTMP) identification [3]
31..22	*unimplemented*	–	Unimplemented bits, read undefined, write as ’0’

5.4.3 I2C_CTR0

Bit	Field	Default	Description
0	I2C_OP	'0'	I^{2}C operation, unused bit
7..1	I2C_ADDR	see Sec. 5.2	I^{2}C address of the CONV-TTL-BLO board
11..8	BIA	0x2	Bytes of Indirect Addressing
19..12	BRD	0x4	Bytes to be read from FPGA
27..20	BWR	0x4	Bytes to be written to FPGA
31..28	*unimplemented*	–	Unimplemented bits, read undefined, write as '0'

5.4.4 I2C_LT

Bit	Field	Default	Description
7..0	WBCP	20	WishBone Clock Period, in ns
31..8	SCLP	0x000000	SCL Period

5.4.5 I2C_DTX

Bit	Field	Default	Description
31..0	DTX	0x00000000	DTX register bits

5.4.6 I2C_DRXA

Bit	Field	Default	Description
31..0	DRXA	0x00000000	DRXA register bits

5.4.7 I2C_DRXB

Bit	Field	Default	Description
31..0	DRXB	0x00000000	DRXB register bits

References

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

[2] ELMA, "Access to board data using SNMP and I2C." `http://www.ohwr.org/documents/227`.

[3] "RTM detection." `http://www.ohwr.org/projects/conv-ttl-blo/wiki/RTM_board_detection`.