Rev. 5 - October 9th, 2024

DT5495

Desktop Programmable Logic Unit

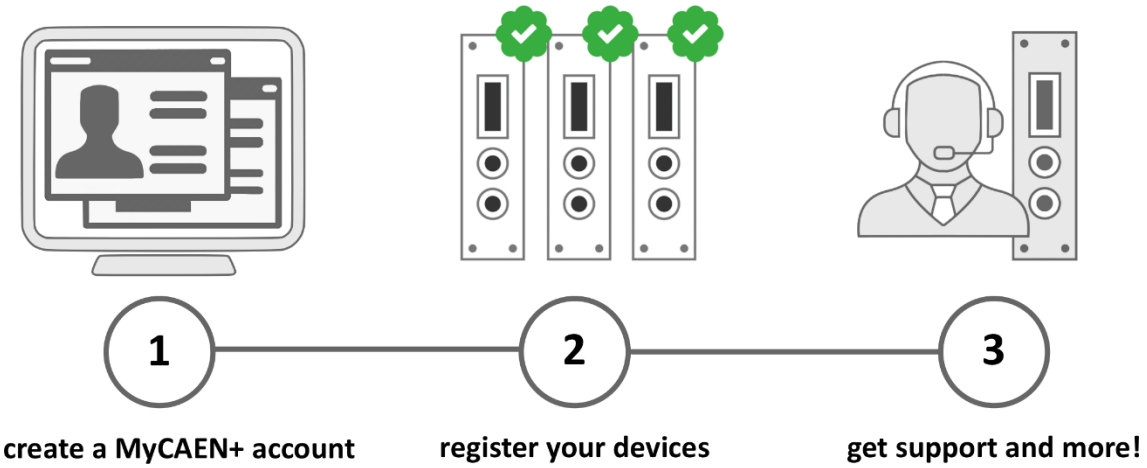
CAEN
Tools for Discovery

User Manual
UM6508

Register your device

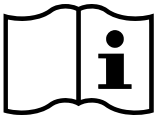
Register your device to your **MyCAEN+** account and get access to our customer services, such as notification for new firmware or software upgrade, tracking service procedures or open a ticket for assistance. **MyCAEN+** accounts have a dedicated support service for their registered products. A set of basic information can be shared with the operator, speeding up the troubleshooting process and improving the efficiency of the support interactions.

MyCAEN+ dashboard is designed to offer you a direct access to all our after sales services. Registration is totally free, to create an account go to https://www.caen.it/become-mycaenplus-user and fill the registration form with your data.



https://www.caen.it/become-mycaenplus-user/

Purpose of this Manual



This document contains the full hardware and software description of the DT5495 Desktop Programmable Logic Unit.

Change Document Record

Date	Revision	Changes
August 2nd, 2018	00	Initial release
May 30th, 2019	01	Added Sec. **PLUscaler_daq** and Chap. **14**. Updated Sec. **Direct USB Driver** and Sec.14.2.
August 29th, 2019	02	Added "DT5495 Mezzanine Installation Notes" in Sec. **Reference Documents**. Removed "Product Code" column from **Tab. 1.2**. Updated **Tab. 5.5**, Sec. **Expansion Mezzanine**, and Sec. **User FPGA I/O ports** with impedance information and instructions on the A395D channel impedance selection. Removed Sec. 14.1 and Sec. 14.2 and renamed Chap. **16**.
October 25th, 2019	03	Removed "Preliminary". Updated Sec. **Expansion Mezzanine**.
March 31st, 2023	04	Added Sec. **Safety Notice**, and Sec. **12.3.1**
October 9th, 2024	05	Adde Chap. **7** and **8**. Updated Sec. **9.2**. Replaced CAEN Upgrader description (obsolete) with CAEN Toolbox in Sec. **10.4**. Updated PID description in Sec. **11.1.3**. Updated LAD signal width in **Tab. 12.8**. Updated Chap **16**.

Symbols, Abbreviated Terms, and Notations

AM	Address Modifier
ETH	Ethernet Interface
FPGA	Field Programmable Gate Array
GDG	Gate and Delay Generator
LB	Local Bus
LBM	Local Bus Master
LBS	Local Bus Slave
LED	Light Emitting Diode
MFPGA	Main FPGA
SPI	Serial Peripheral Interface
UFPGA	User FPGA
USB	Universal Serial Bus

Reference Documents

[RD1] UM11111 – CAEN Toolbox user Manual

[RD2] A967 Adapter for P50E – 068S 68 Pin Connectors

[RD3] UM6510 – CAEN PLULib Library User Manual

[RD4] UM5175 – FW2495SC - DT5495 V2495 Scaler Firmware User Manual

[RD5] DT5495 Mezzanine Installation Notes

All documents can be downloaded at: https://www.caen.it/support-services/documentation-area/

Manufacturer Contacts

CAEN S.p.A.
Via Vetraia, 11 55049 Viareggio (LU) - ITALY
Tel. +39.0584.388.398 Fax +39.0584.388.959
www.caen.it | info@caen.it

Limitation of Responsibility

If the warnings contained in this manual are not followed, CAEN will not be responsible for damage caused by improper use of the device. The manufacturer declines all responsibility for damage resulting from failure to comply with the instructions for use of the product. The equipment must be used as described in the user manual, with particular regard to the intended use, using only accessories as specified by the manufacturer. No modification or repair can be performed.

Disclaimer

The information contained herein has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. CAEN spa reserves the right to modify its products specifications without giving any notice; for up to date information please visit www.caen.it.

Made in Italy

We remark that all our boards have been designed and assembled in Italy. In a challenging environment where a competitive edge is often obtained at the cost of lower wages and declining working conditions, we proudly acknowledge that all those who participated in the production and distribution process of our devices were reasonably paid and worked in a safe environment (this is true for the boards marked "MADE IN ITALY", while we cannot guarantee for third-party manufactures).

Index

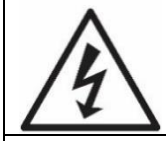
List of Figures

List of Tables

Safety Notices

N.B. Read carefully the "Precautions for Handling, Storage and Installation" document provided with the product before starting any operation.

The following HAZARD SYMBOLS may be reported on the unit:

	Caution, refer to the product manual
	Caution, risk of electrical shock
	Protective conductor terminal
	Earth (Ground) Terminal
	Alternating Current
	Three-Phase Alternating Current

The following symbol may be reported in the present manual:

 General warning statement

The symbol could be followed by the following terms:

- **DANGER**: Indicates a hazardous situation that, if not avoided, will result in serious injury or death.
- **WARNING**: Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
- **CAUTION**: Indicates a situation or condition that, if not avoided, could cause physical injury, or damage the product and/or its environment.

CAUTION: Avoid potential hazards.

USE THE PRODUCT ONLY AS SPECIFIED.

ONLY QUALIFIED PERSONNEL SHOULD PERFORM SERVICE PROCEDURES

CAUTION: Avoid Electric Overload.

TO AVOID ELECTRIC SHOCK OR FIRE HAZARD, DO NOT POWER A LOAD OUTSIDE OF ITS SPECIFIED RANGE

CAUTION: Avoid Electric Shock.

TO AVOID INJURY OR LOSS OF LIFE, DO NOT CONNECT OR DISCONNECT CABLES WHILE THEY ARE CONNECTED TO A VOLTAGE SOURCE

CAUTION: Do Not Operate in Wet/Damp Conditions.

TO AVOID ELECTRIC SHOCK, DO NOT OPERATE THIS PRODUCT IN WET OR DAMP CONDITIONS

CAUTION: Do Not Operate in an Explosive Atmosphere.

TO AVOID INJURY OR FIRE HAZARD, DO NOT OPERATE THIS

PRODUCT IN AN EXPLOSIVE ATMOSPHERE

THIS DEVICE SHOULD BE INSTALLED AND USED BY A SKILLED TECHNICIAN ONLY OR UNDER HIS SUPERVISION

DO NOT OPERATE WITH SUSPECTED FAILURES.
IF YOU SUSPECT THIS PRODUCT TO BE DAMAGED, PLEASE CONTACT THE TECHNICAL SUPPORT

1 Introduction

The DT5495 is a general-purpose programmable FPGA and I/O unit in a compact desktop form factor. The board is a suitable solution for the implementation of digital functions such as Coincidence, Trigger Logic, Gate and Delay Generator, Input/Output Register and more.

The programmable architecture is based on the User FPGA (hereafter UFPGA). The UFPGA is directly interfaced to the front panel I/Os and to an onboard Gate and Delay Generator, that allows to delay and gate up to 32 signals. A second FPGA, the Main FPGA (hereafter MFPGA), is responsible for USB and Ethernet interface management. The MFPGA communicates with the UFPGA through an internal local bus.

The presence of three expansion slots interfaced to the UFPGA allows to extend the channel interface of the DT5495 by adding up to three independent mezzanine boards. Five mezzanine board types are available: A395A, A395B, A395C, A395D, A395E (see **Tab. 1.2**). The DT5495 can reach a maximum of 194 I/O channels.

The board can be controlled and programmed through either the Ethernet or the USB interface. The custom firmware can be developed either with VHDL tools or the new SCI-Compiler application (Chap. **12**). The firmware is then loaded on the User FPGA by a dedicated JTAG connector on the rear panel, allowing for in-system JTAG configuration and debugging (e.g. using Altera SignalTap).

Simple operations, like retrieving the board information or changing the ethernet settings, can be performed simply via the Internet or a Local Area Network (LAN) through the web inteface, without need of any control software but just opening a web browser.

The DT5495 is also available in VME form factor, called V2495, which represents an evolution of CAEN V1495 board. Thanks to several analogies with the V1495 (e.g. the front panel connectors have the same function and nomenclature with respect to the V1495), users who want to upgrade their system with the DT5495 can port their firmware already developed for the V1495 just considering few hints (Sec. **12.6**).

The following **Tab. 1.1** is a comparison table between the two modules.

	V1495	DT5495
User FPGA	Altera Cyclone EPI1C20, 20k LEs	Altera Cyclone V GX, 50k LEs
Front Panel I/O	Sections A and B (2 x 68-pin Robinson Nugent connector; 32 input channel each) Section C (1 x 68-pin Robinson Nugent connector; 32 output channels) Section G (2 X LEMO connector; input/output channels)	
On-board Delay Generator	4 Delay lines	32 Gate and Delay lines
Local Bus	16-bit 50 MHz parallel interface	
VME Bus	Addressing spaces: A24, A32 Data transfer modes: D16, D32, BLT32	Not Supported
USB Interface	Not Supported	USB2.0
ETH Interface	Not Supported	ETHERNET 10/100T
Mezzanine Boards	A395A, A395B, A395C, A395D, A395E	

Tab. 1.1: Comparison table between V1495 and DT5495

Tab. 1.2 lists the board models, the firmware and hardware related products as well as the ordering option information.

Board Models	Description
DT5495	Desktop Programmable Logic Unit
5495KITPRO	DT5495 + SW555PRO - Programmable Logic Unit & Sci-Compiler PRO Bundle Kit
User Firmware	**Description**
FW2495SC(*)	Channel Latching Scaler for V2495 and DT5495
Accessories	**Description**
A395A	32 LVDS/ECL/PECL input channels
A395B	32 LVDS output channels
A395C	32 ECL output channels
A395D	8 NIM/TTL input/output channels
A395E	8 Channel 16Bit ±5V DAC
A967	32 Channel Cable Adapter (1x32 to 2x16)

Tab. 1.2: Table of models and related items

(*) Pay firmware. Unlocking license needed. Please, refer to the FW2495SC web page for details.

2 Block Diagram

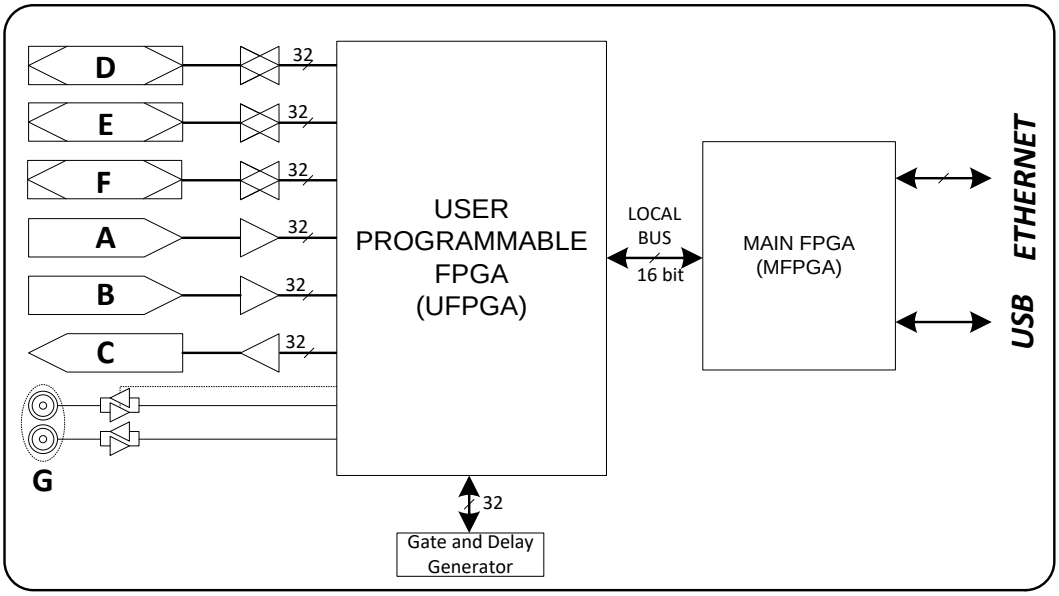


Fig. 2.1: Block diagram

3 Main Components and Interconnections

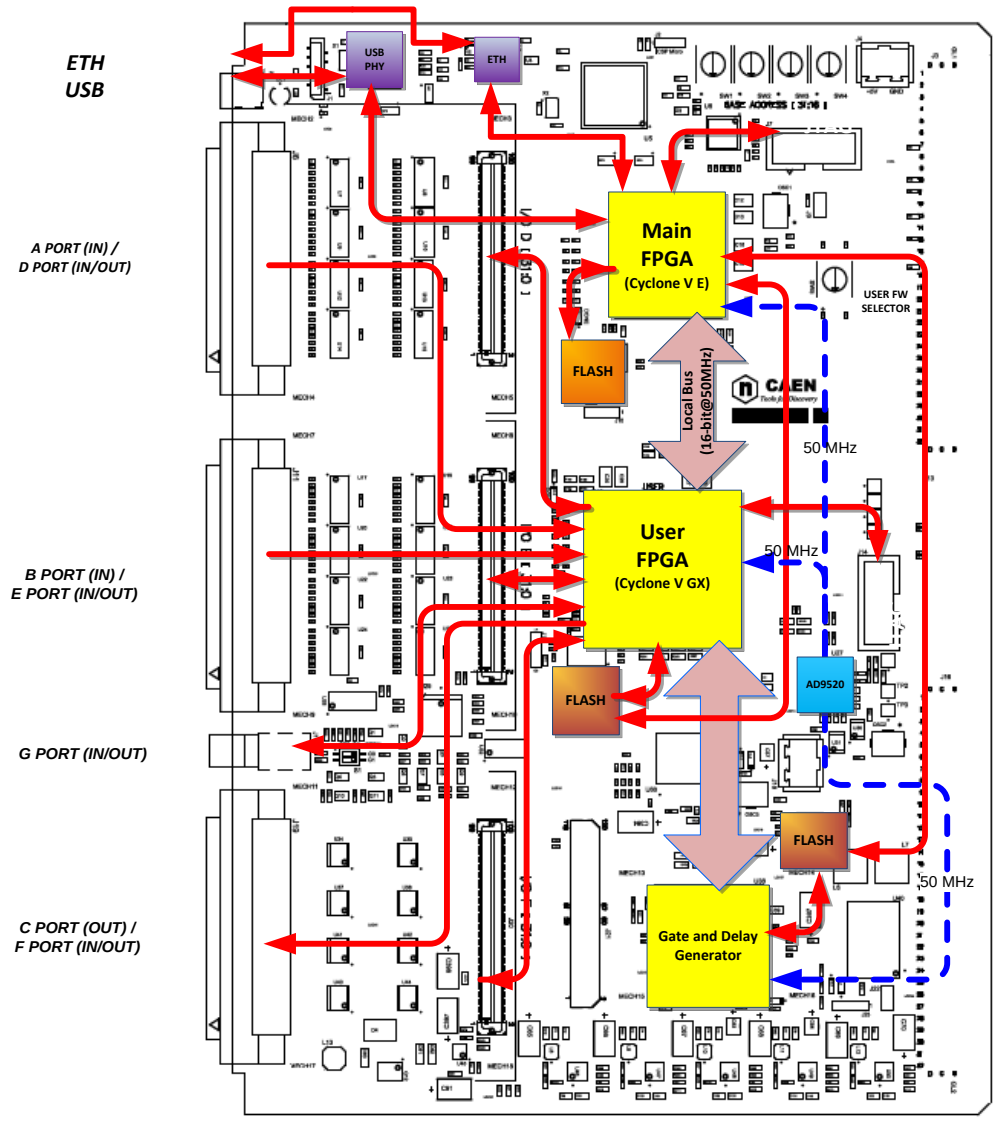


Fig. 3.1: Main components and interconnections

3.1 USB Interface

The DT5495 is equipped with a USB2.0 interface. The USB physical layer is managed by a high-speed transceiver controlled by the Main FPGA.

3.2 Ethernet Interface

In addition to the USB, the DT5495 provides a 10/100T Ethernet interface controlled by the Main FPGA.

3.3 Main FPGA

The MFPGA (Altera Cyclone V E) manages the Ethernet and USB interfaces and the connection with the UFPGA through a proprietary 16-bit@50 MHz local bus. The MFPGA has a dedicated external flash memory for configuration purposes. It also pilots the flash memories dedicated to loading the firmware on the UFPGA and on the GDG.

3.4 User FPGA

The User FPGA (Altera Cyclone V GX) manages the I/O peripherals (A/D, B/E, G, C/F ports) and communicates with the GDG. A dedicated external flash memory can store a set of firmware images to be loaded on the User FPGA. A dedicated JTAG connector allows to program the UFPGA “on-the-fly” for fast firmware prototyping and debugging.

3.5 Gate and Delay Generator

The DT5495 hosts a Gate and Delay Generator (see **Fig. 3.2)** able to provide up to 32 gated and delayed signals (“delayed signals”) triggered by 32 inputs (“start signals”). The gate width and delay value are user programmable. The GDG is an external component implemented in a Xilinx Spartan-6 FPGA. It is connected through a serial bus (SPI) to the User FPGA for gate and delay register programming (refer to Sec. **3.5** for detailed information). The GDG configuration is stored in a dedicated flash memory. The GDG firmware cannot be modified by the user.

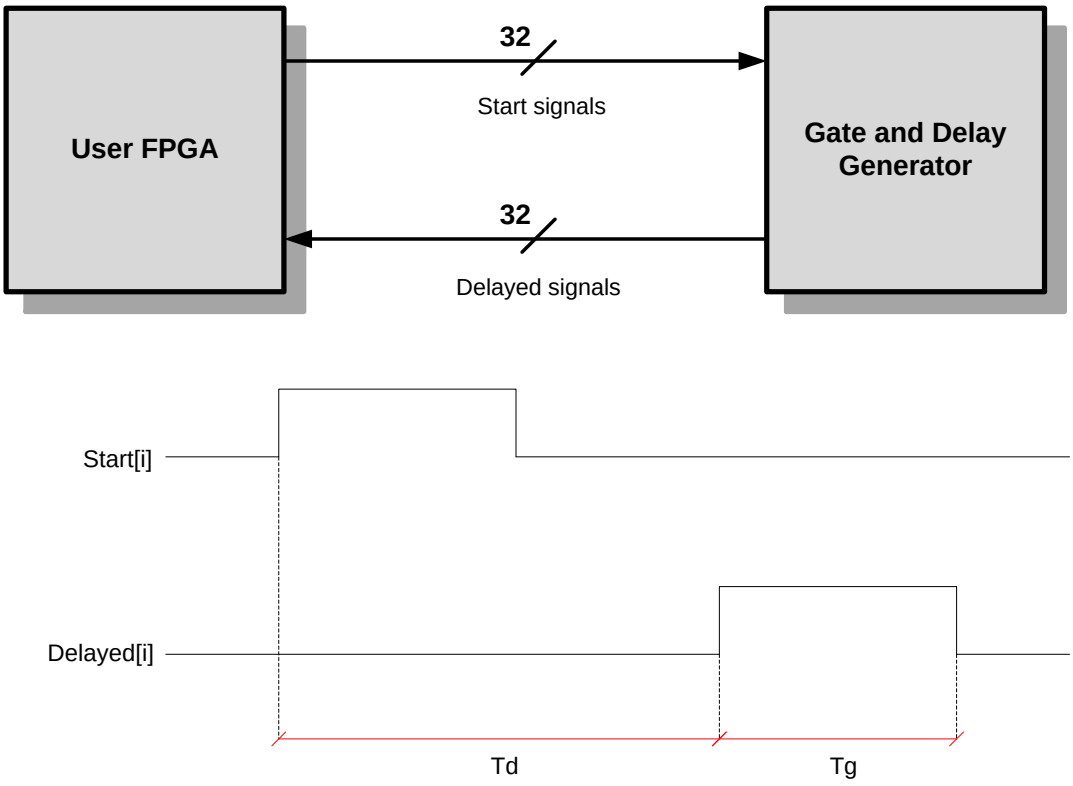
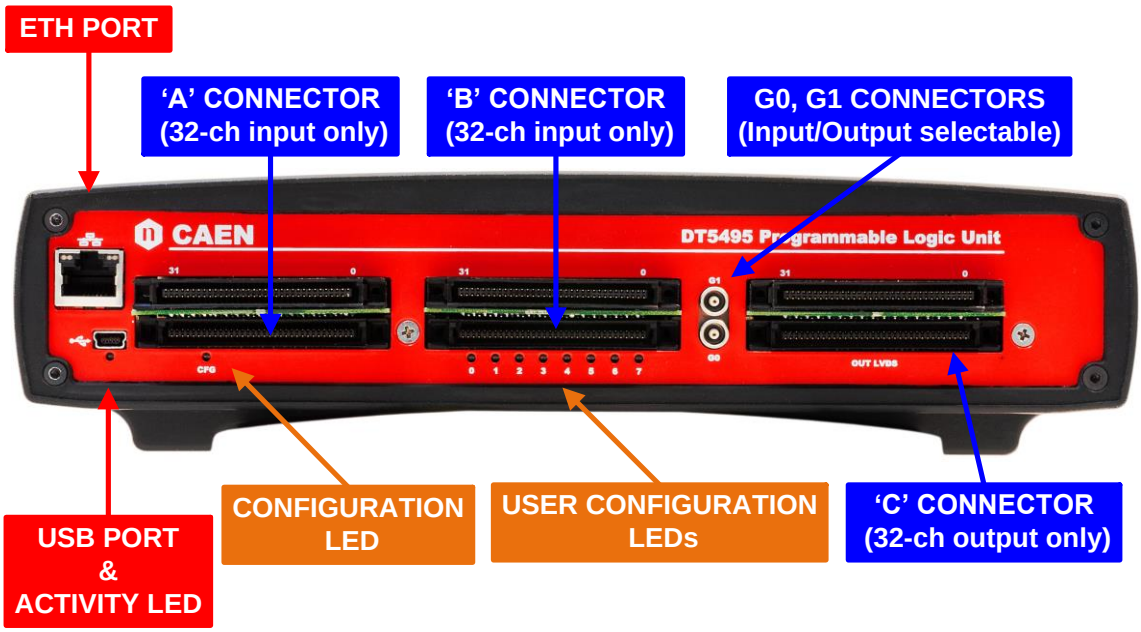


Fig. 3.2: The UFPGA and GDG interface.

3.6 Clock Distribution

Each FPGA receives the same 50-MHz system clock generated by a common on-board oscillator.

4 Front Panel Connectors, LEDs and Labels



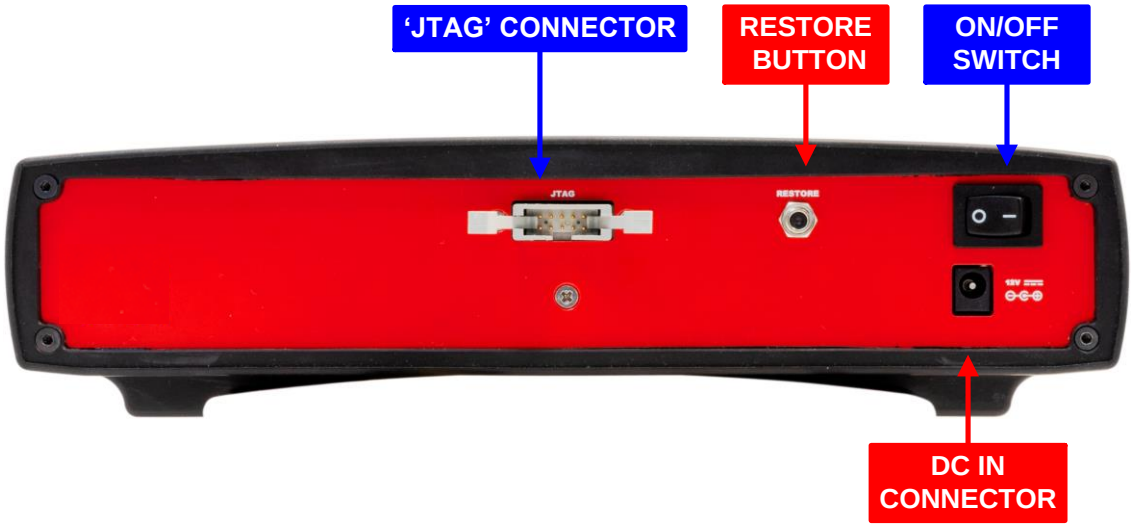


Fig. 4.1: front (top) and rear (down) panel view

4.1 Front Panel

USB PORT		
	**FUNCTION** MiniUSB connector to communicate with the DT5495 by USB link for board configuration and firmware upgrade. **ELECTRICAL SPECS** *N.A.*	**MECHANICAL SPECS** Series: miniUSB connectors. Type: SD-54819-026 (B-type). Manufacturer: Molex Inc. **CABLES** A 1-m USB standard cable is included with the DT5495 (USB A-type to miniUSB B-type).

USB LED (GREEN): driven by the Main PFGA, this LED lights up if a USB read/write access to the board is performed.

ETHERNET PORT		
	**FUNCTION** Ethernet connector to communicate with the DT5495 for board configuration and firmware upgrade. **ELECTRICAL SPECS** 10/100T ETHERNET	**MECHANICAL SPECS** Series: RJ45 connectors. Type: GIGABIT MIC3801D-5166. Manufacturer: Wurt. **CABLES** A 2-m Ethernet cable is included with the DT5495 (type ASSMANN: A-MCSSP60020/R).

A, B, C CONNECTORS	
	
FUNCTION Motherboard I/O 34+34 pin connectors: - A/B are inputs - C is an output **ELECTRICAL SPECS** See **Tab. 5.1**.	**MECHANICAL SPECS** Series: 80-0009-0666-1. Type: P50E-068-P1-SR1-TG. Manufacturer: 3M.

ETHERNET PORT		
	**FUNCTION** This LED is constantly on when all the on-board FPGAs are configured by the application firmware on their relevant flash memory. The LED blinks if the MFPGA is in Factory mode.	**COLOR** Green.

G0, G1 CONNECTORS		
	**FUNCTION** Bidirectional single-ended connectors. Factory direction is Output. (high impedance state). External 50Ω termination required when using the connector as input. **ELECTRICAL SPECS** See **Tab. 5.1**.	**MECHANICAL SPECS** Series: 00 LEMO Connectors. Type: EPY.00.250.NTN. Manufacturer: LEMO

0, 1, 2, …, 7 LEDs	
	
FUNCTION The status of these LEDs is user-programmable NOTE: when the UFPGA is in Factory mode, odd and even LEDs alternatively blink.	**COLOR** Green.

4.2 Rear Panel

JTAG CONNECTOR		
	**FUNCTION** This connector allows for on-the-fly FPGA programming. **ELECTRICAL SPECS** *N.A.*	**MECHANICAL SPECS** Series: 4600 - panel rectangular connectors. Type: 4610-6350. Manufacturer: 3M.

RESTORE BUTTON		
	**FUNCTION** Restores the default IP address and forces the module in factory state (see Sec. **9.3.1**). **ELECTRICAL SPECS** *N.A.*	**MECHANICAL SPECS** Series: 9000 - momentary pushbutton switches. Type: APR9433NA-O. Manufacturer: Apem.

JTAG CONNECTOR		
	**FUNCTION** Input connector for the DT5495 main power supply from the external AC/DC adaptor. **ELECTRICAL SPECS** Input voltage: +12 VDC (typ.).	**MECHANICAL SPECS** Series: DC power Jacks. Type: KLDX-0202-A-LT. Manufacturer: KYCON.

ON/OFF SWITCH		
	**FUNCTION** Power switch of the module: on (I) / off (O) **ELECTRICAL SPECS** *N.A.*	**MECHANICAL SPECS** Series: Power Switches. Type: Rocker SwitchP (A11131121000) Manufacturer: Molveno.

5 Technical Specifications

DT5495 Motherboard				
FORM FACTOR	Desktop module			
I/O SECTIONS A and B	**Nr. of Channels** 32 **Direction** Input	**Logic** Direct **Signal** Differential LVDS/ECL/PECL (single ended TTL optional) Z_{diff}: 100 Ω Extended Common Mode Input range: -4V to +5V Fail Safe input feature	**Bandwidth** 200 MHz **Front Panel Connector** Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins	
I/O SECTION C	**Nr. of Channels** 32 channels **Direction** Output	**Logic** Direct **Signal** Differential LVDS Require 100 Ω termination	**Bandwidth** 250 MHz **Front Panel Connector** Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins	
I/O SECTION G	**Nr. of Channels** 2 **Direction** I/O (default is Ouput) External 50 Ω termination required when used as input	**Logic** TTL IN = Direct TTL OUT = Direct NIM IN = Invert NIM OUT = Direct **Signal** Single ended NIM/TTL selectable	**Bandwidth** 250 MHz **Front Panel Connector** LEMO 00	
GATE and DELAY GENERATOR	**Minimum Delay/Gate**	**Min.**	**Typ.**	**Max.**
		9.6 ns	10.7 ns	11.8 ns
	Maximum Delay/Gate	631 µs	701.2 µs	771.5 µs
	Maximum channel-to-channel spread: 20%			
COMMUNICATION INTERFACE	**ETHERNET** 10/100T		**USB** USB 2.0 compliant	
POWER REQUIREMENTS	750 mA (max.) @ +12VDC AC-DC 12V-45W power unit included			

Tab. 5.1: DT5495 specifications table

A395A Mezzanine Board			
I/O SECTION	**Nr. Of Channels** 32 **Direction** Input	**Logic** Direct **Signal** Differential LVDS/ECL/PECL (single ended TTL optional) Z_{diff}: 100 Ω Extended Common Mode input range: -4V to +5V Fail Safe input feature	**Bandwidth** 200 MHz **Front Panel Connector** Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins
POWER CONSUMPTIONS	0.1 A (max) @ +5V internal rail +12V and -12V internal rails are not used		

Tab. 5.2: A395A Mezzanine specifications table

A395B Mezzanine Board			
I/O SECTION	**Nr. of Channels** 32 **Direction** Output	**Logic** Direct **Signal** Differential LVDS Requires 100 Ω termination	**Bandwidth** 250 MHz **Front Panel Connector** Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins
POWER CONSUMPTIONS	0.1 A (max) @ +5V internal rail +12V and -12V internal rails are not used		

Tab. 5.3: A395B Mezzanine specifications table

A395C Mezzanine Board			
I/O SECTION	**Nr. of Channels** 32 **Direction** Output	**Logic** Direct **Signal** Differential ECL	**Bandwidth** 300 MHz **Front Panel Connector** Robinson Nugent P50E-068-P1-SR1-TG type, (34+34) pins
POWER CONSUMPTIONS	1.4 A (max) @ +5V internal rail +12V and -12V internal rails are not used		

Tab. 5.4: A395C Mezzanine specifications table

A395D Mezzanine Board			
I/O SECTION	**Nr. of Channels** 8 **Direction** I/O selectable **Impedance** Selectable 50Ω/high-Z by on-board jumper	**Logic** Direct **Signal** selectable TTL/NIM: TTL IN = Direct TTL OUT = Direct NIM IN = Invert NIM OUT = Direct	**Bandwidth** 250 MHz **Front Panel Connector** LEMO 00
POWER CONSUMPTIONS	1.1 A (max) @ +5V internal rail +12V and -12V internal rails are not used		

Tab. 5.5: A395D Mezzanine specifications table

A395E Mezzanine Board			
I/O SECTION	**Nr. of Channels** 8 **Direction** Output	**Logic** Analog **Signal** 16-bit resolution ±5V @10kΩ RL ±4V @200Ω RL	**Bandwidth** n.a. **Front Panel Connector** LEMO 00
POWER CONSUMPTIONS	0.3 A (max) @ +5V internal rail +12V and -12V internal rails are not used		

Tab. 5.6: A395E Mezzanine specifications table

6 Power Requirements

The DT5495 is powered by the external 45W 12V AC/DC stabilized power supply unit included in the delivered kit (see Chap. Errore. L'origine riferimento non è stata trovata.). The maximum required current is typically 500mA (without any mezzanine extension).

Switchbox FRA030/045/050 Series

30-50W SINGLE OUTPUT AC/DC DESKTOP ADAPTOR

Features

- Universal input
- IEC320 receptacle 2P or 3P
- Optional output connector
- OVP, OCP, OPP, auto recovery
- CEC compliance

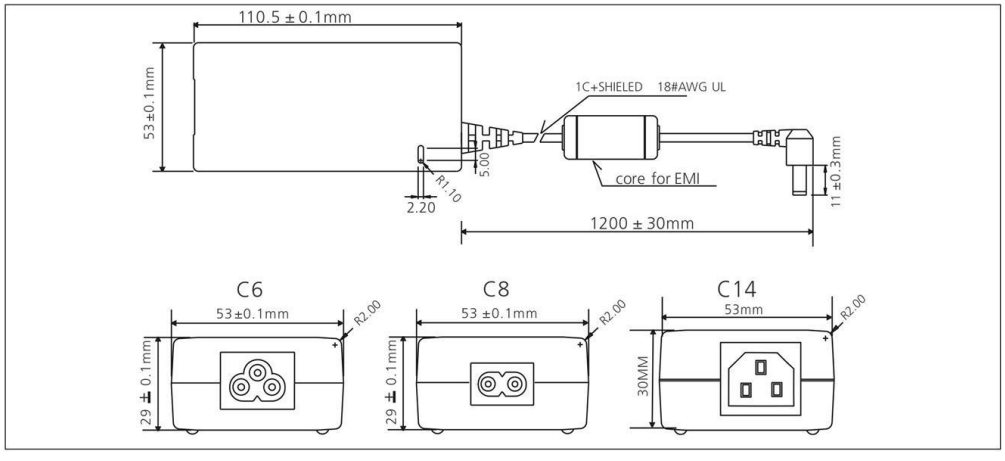
Specifications

INPUT	
Voltage range	100-240VAC.
Inrush current	40A at 115VAC / 80A at 230VAC max.
Dielectric withstand	Input/output 3,000VDC.
OUTPUT	
Output voltage	5-48V.
Ripple and noise	2% p-p max.
Load regulation	±5% max.
No load stand by power	<0.5W @ 230VAC.
Efficiency	>=85% for CEC requirement.
Hold up time	10mS at nominal line.
Protections	OCP, OVP, over power & short circuit.
GENERAL	
Std output connector	Dc barrel jack.
Std output cable/length	UL1185, #18AWG / 5 ft.
ENVIRONMENTAL	
Operating temperature	0°C to +40°C.
Storage temperature	–20°C to +85°C.
STANDARDS	
Safety standards	IEC/UL/EN60950-1, CE, CB.
EMC	EN55022 (CISPR 22) class B, FCC class B.

MODEL NUMBER	OUTPUT VOLTAGE	OUTPUT CURRENT	MAX WATTS	CEC*
FRA030-S05-X	5~7 V	6.00~4.30 A	30 W	
FRA045-S09-X	7~9 V	6.00~5.00 A	45 W	IV
FRA045-S12-X	12~15 V	3.75~3.00 A	45 W	IV
FRA045-S15-X	15~18 V	3.00~2.50 A	45 W	IV
FRA045-S24-X	18~24 V	2.50~1.88 A	45 W	IV
FRA050-S12-X	12~15 V	4.17~3.33 A	50 W	IV
FRA050-S15-X	15~18 V	3.33~2.87 A	50 W	IV
FRA050-S24-X	18~24 V	2.78~2.08 A	50 W	IV
FRA050-S36-X	30~36 V	1.67~1.38 A	50 W	IV
FRA050-S48-X	40~48 V	1.25~1.04 A	50 W	IV

**CEC compliance model provide under customer's request.*
**CEC compliance model standby power (@ no load) <0.5W.*

Note:
X = Inlet type code
X = 4, IEC320 C14 *X = 6, IEC320 C6* *X = 8, IEC320 C8*



powerbox
www.powerbox.info

20081029

7 Packaging and Compliancy

The DT5495 board is available in a desktop form factor.

The device is inspected by CAEN before shipment, and it is guaranteed to leave the factory free of mechanical or electrical defects.

The content of the delivered package standardly consists of the part list shown in the table below.

	Part	Description	Qt
	DT5495 (with or without mezzanine board extension[1])	**Programmable Logic Unit**	x1
	USB cable	**USB-2.0 cable, type A to mini-USB, L=1mt**	x1
	Ethernet cable	**Cat6 SSTP Ethernet cable assembly RJ45**	x1
	Documentation	**UM5175 - V2495 PLU User Manual**	-

Tab. 7.1: Delivered kit content

[1] Mezzanine boards are available by ordering option (refer to **Tab. 1.2**).

CAUTION: to manage the product, consult the operating instructions provided.

When receiving the unit, the user is strictly recommended to:

- Inspect containers for damage during shipment. Report any damage to the freight carrier for possible insurance claims.
- Check that all the components received match those listed on the enclosed packing list as in **Tab. 7.1**. (CAEN cannot accept responsibility for missing items unless we are notified promptly of any discrepancies.)
- Open shipping containers; be careful not to damage contents.
- Inspect contents and report any damage. The inspection should confirm that there is no exterior damage to the unit such as broken knobs or connectors and that the front panel and display face are not scratched or cracked. Keep all packing material until the inspection has been completed.
- If damage is detected, file a claim with the carrier immediately and notify CAEN service.
- If equipment must be returned for any reason, carefully repack equipment in the original shipping container with original packing materials if possible. Please, contact CAEN service.
- If equipment is to be installed later, place equipment in the original shipping container and store it in a safe place until ready to install.

DO NOT SUBJECT THE ITEM TO UNDUE SHOCK OF VIBRATIONS

DO NOT BUMP, DROP OR SLIDE SHIPPING CONTAINERS

DO NOT LEAVE ITEMS OR SHIPPING CONTAINERS UNSUPERVISED IN AREAS WHERE PERSONNEL UNTRAINED MAY MISHANDLE THE ITEMS

USE ONLY ACCESSORIES WHICH MEET THE MANUFACTURER'S SPECIFICATIONS

8 PID (Product Identifier)

PID is the CAEN product identifier, an incremental number greater than 10000 that is unique for each product. The PID is on a label affixed to the product (**Fig. 8.1**), it is even stored in an on-board non-volatile memory (Sec. **11.1.3**), and readable by CAEN toolbox software (Sec. **10.4.1**).

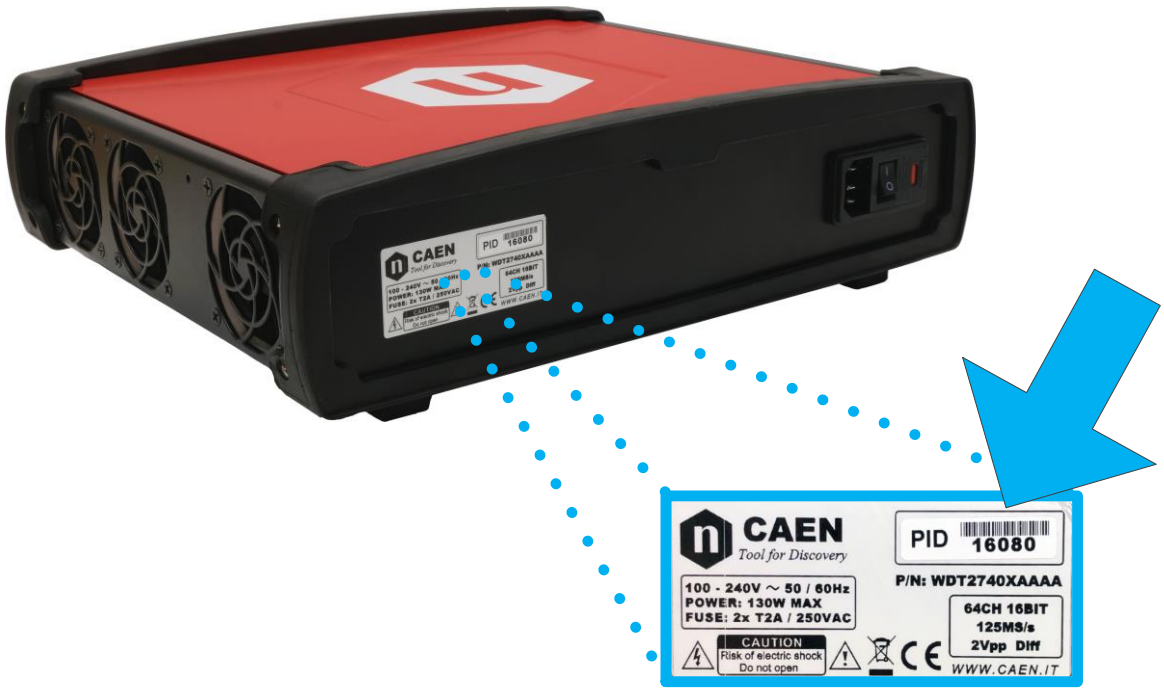


Fig. 8.1:PID location on the desktop device (the number in the picture is purely indicative)

9 Getting Started with DT5495

9. Expansion Mezzanines

Four different type of mezzanine boards are available by ordering option to extend the DT5495 functions (**Tab. 1.2**).

Fig. 9.1: DT5495 motherboard with mezzanine boards

9.1 Mezzanine Installation

Users who need to install one or more A395x-series mezzanine board on the DT5495 can follow the instructions in the dedicated Guide **[RD5]** or look at the tutorial video on the product web page.

9.2 Front Panel Connector Cabling

Motherboard I/O sections A, B, C and A395A, A395B and A395C mezzanine boards feature the Robinson Nugent P50E-068-P1-SR1-TG multi-pin connector (see Chap. **4**), whose pin layout is shown in **Fig. 9.2**.

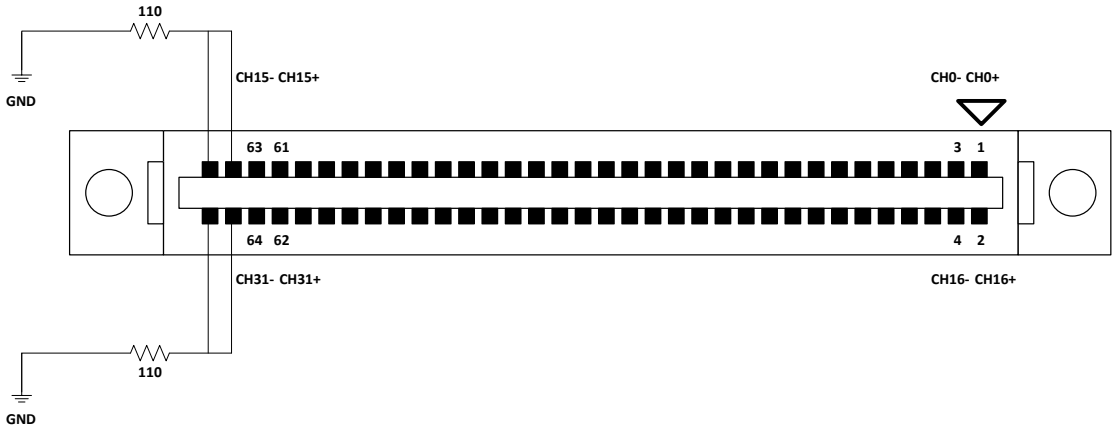


Fig. 9.2: Multi-pin connector pin assignment

The CAEN A967 Cable Adapter (**Fig. 9.3**) allows to adapt each Robinson Nugent multi-pin connector into two 1” 17+17-pin Header-type male connectors (3M, 4634-7301) with locks through two 25 cm long flat cables. Refer to the cable datasheet **[RD2]** for specifications and to **Tab. 1.2** for the ordering option.

Fig. 9.3: CAEN A967 Cable Adapter

9.3 Power-on Configuration Sequence

When the board is powered on, the on-board programmable devices are configured with the firmware images stored in their dedicated flash memories. If all programmable devices (MFPGA, UFPGA and GDG) are correctly configured, the CFG LED is lit on. In case an anomaly occurs, the CFG LED status can be:

1) BLINKING: the FPGAs are configured, but the Main FPGA remained in Factory mode.

The MFPGA can be configured by two kind of firmware images which are stored on the dedicated flash memory: the Factory firmware and the Application firmware. The Factory firmware is a basic firmware specifically intended for boot and recovery functions. The Application firmware is the standard operating firmware and can be upgraded by the user through the Ethernet or the USB interface.

At power-on, the MFPGA configures itself with the Factory firmware image and subsequently attempts to load the Application firmware. If the application firmware is missing or corrupted, the MFPGA will remain in the factory mode. The Factory firmware guarantees the communication with the board and allows the MFPGA Application firmware upgrade.

2) OFF: one or more FPGAs may not have been configured.

In case of a MFPGA configuration issue, an attempt to force the MFPGA in Factory mode can be performed by the Restore button (see next paragraph). If the CFG LED starts blinking, an application firmware upgrade can be attempted.

9.3.1 Restore Function

The Restore button on the DT5495 rear panel (see **Fig. 4.1**) allows to:

- Reset the board to the default IP address: 192.168.0.90. The IP can anytime be changed by the Web Interface (see Sec. **10.3**)
- Force the board in the Factory firmware of the Main FPGA.

Use the button in two alternative modes:

1. Hold the button down (with a pen tip, for example).

   2a. Power cycle the DT5495, then wait for less than 5 seconds; this mode only forces the boot from the factory Main FPGA firmware, while the ethernet IP address of the board is not affected.

   2b. Power cycle the DT5495, then wait for at least 15 seconds; this way, both the IP address is restored to the default value and the board is forced to boot from the factory Main FPGA firmware.

3. Release the button.

4. Power cycle the DT5495 again.

In factory mode, the CFG LED starts blinking (see the CFG description in Chap. **4**). Use the Web Interface with ethernet connection to 192.168.0.90 for check in case of IP address change (Sec. **10.3**).

10 Driver and Software Installation

10.1 Drivers

To communicate with the board using one of the available communication buses, specific drivers should be installed on your host device.

Note: to ensure the best system performance, it is recommended to have always the latest drivers release installed.

10.1.1 Direct USB Driver

In order to communicate with the DT5495 through the front panel USB interface (see Chap. **4**), the related driver is available for free download at the DT5495 *Downloads* page on CAEN web site (**login required**).

- **WINDOWS OS**

A step-by-step driver installation procedure is given next.

Note: The procedure is based on a Windows 10 system. It may be slightly different for other Windows OSs.

1. Download and unpack the driver installation package on the host PC.
2. Plug the USB cable on the DT5495 and on the PC. Windows should detect the new hardware.
3. Go to the *Device Manager* area; the new hardware is listed in *Other devices* as "DT5495" (see **Fig. 2.1**).

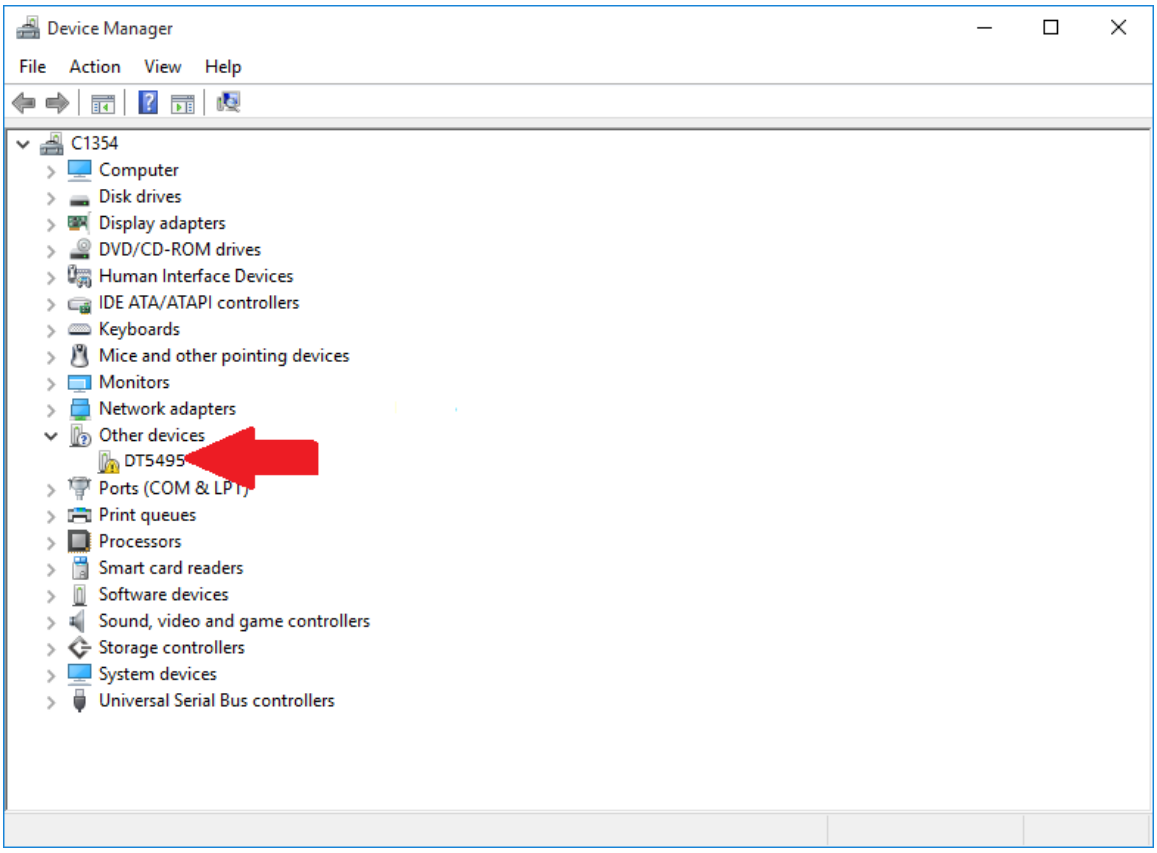


Fig. 10.1: DT5495 hardware detection

4. Right click on “DT5495” item and select *Update Driver Software* option in the slide menu.

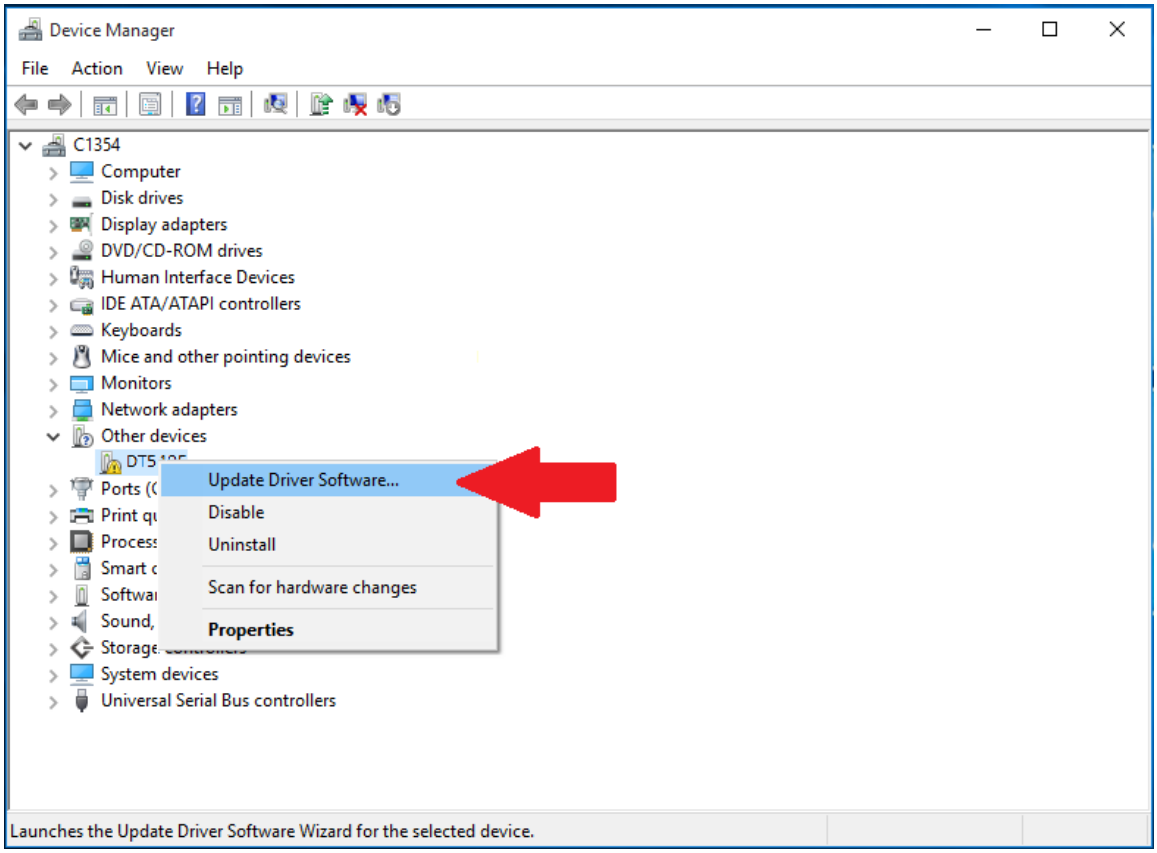


Fig. 10.2: USB driver manual installation: Step1

5. Select *Browse my computer for driver software* as in **Fig. 10.3**.

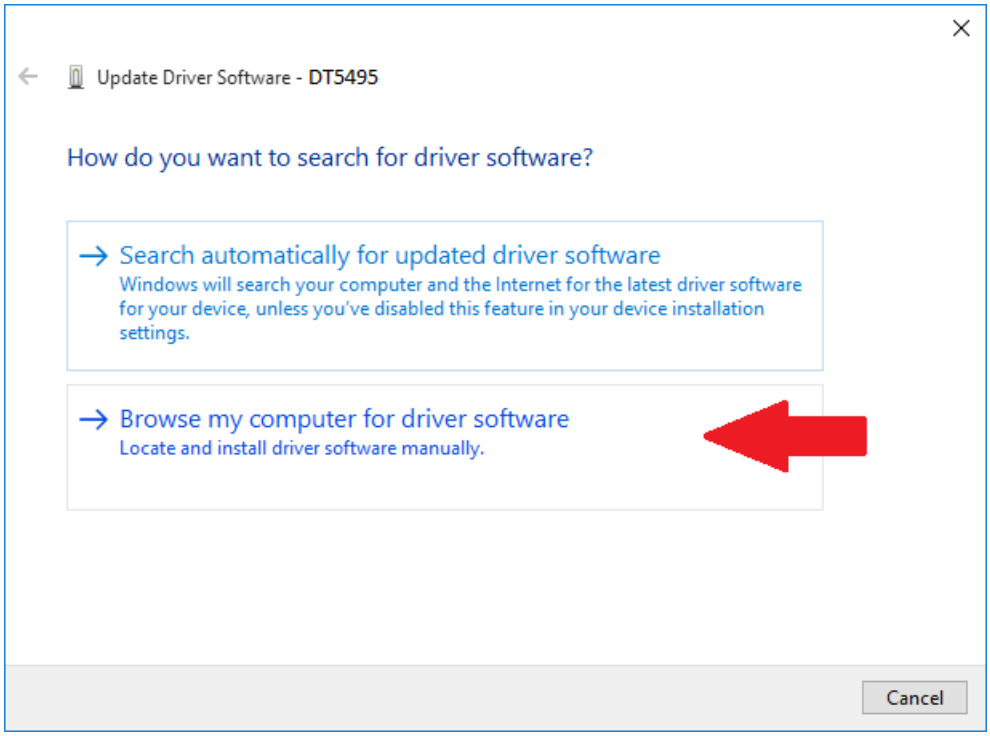


Fig. 10.3: USB driver manual installation: Step2

6. Use the [Browse] button to point to the driver folder in the destination path on the host PC, then click the [Next] button.

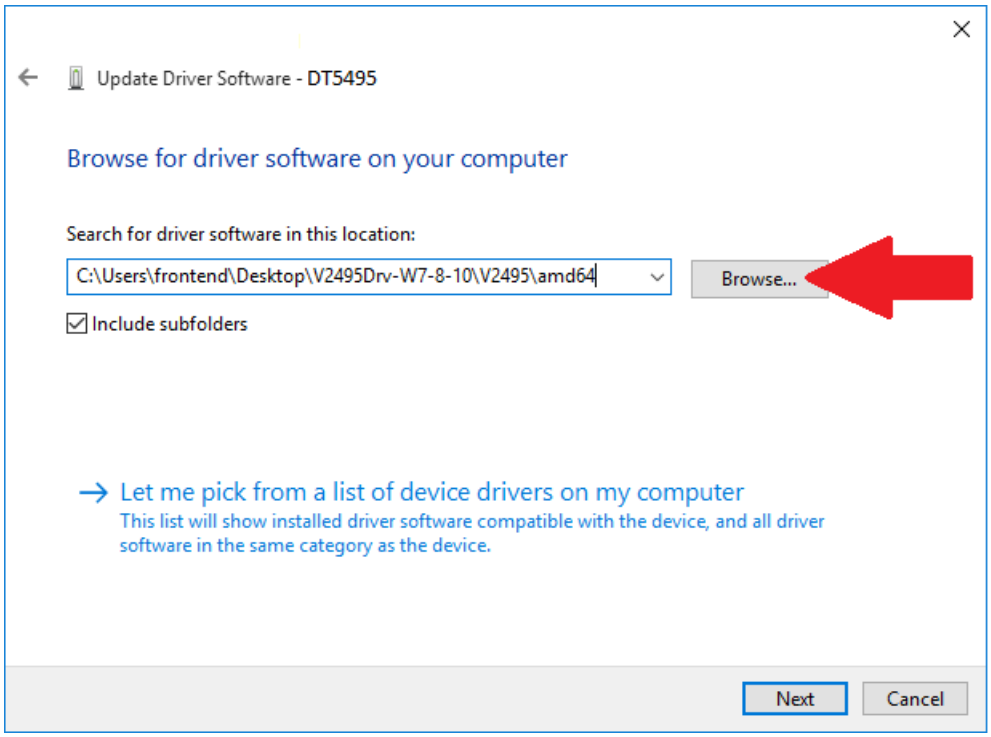


Fig. 10.4: USB driver manual installation: Step3

7. Click the [Close] button at the end of the installation process.

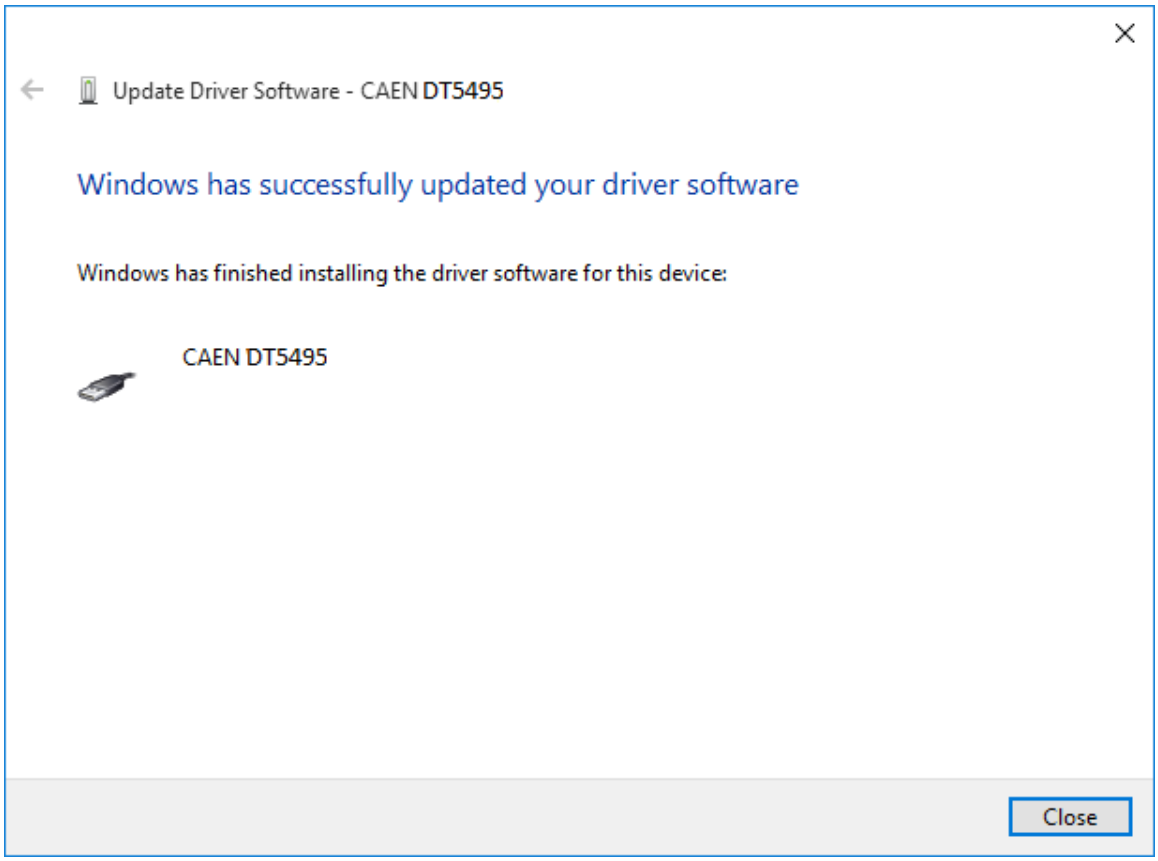


Fig. 10.5: USB driver manual installation: Step4

8. Check that the "CAEN DT5495" item appears in the *Universal Serial Bus controllers* list.

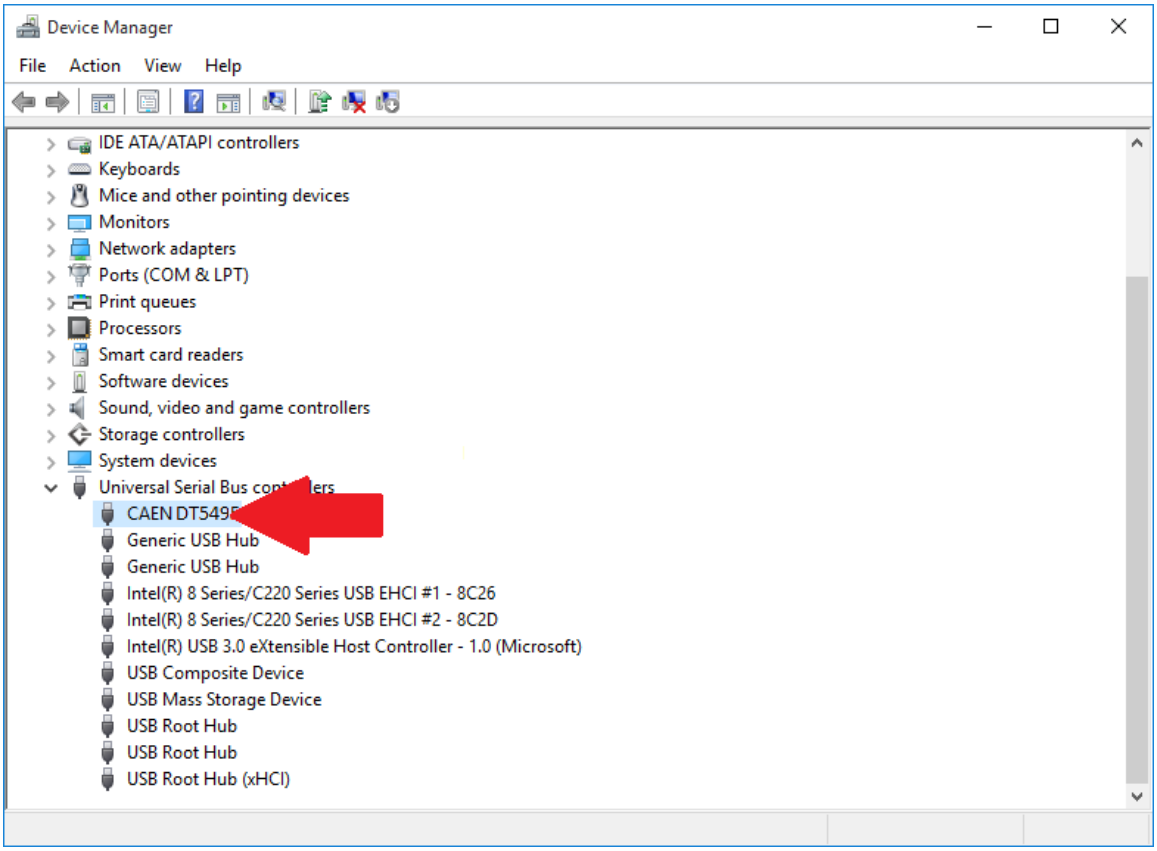


Fig. 10.6: USB driver manual installation: Step5

➢ **Linux OS**

DT5495 is fully supported by Linux kernels from kernel version 3.13 on. This means that such kernel versions should be able to recognize the hardware automatically, without requiring the user to install any driver.

Note: The in-Linux drivers don't automatically give to the user the rights to directly connect to the board. It is then recommended to authenticate as root or to enable the low-level user rights before to try any communication by software.

10.2 Ethernet Configuration

1. Connect the Ethernet cable between Hexagon and the PC.
2. Configure the Ethernet network of your PC in the following steps.
   a. Open the path: *Control Panel - Network and Internet - Network and Sharing Center* (Errore. L'origine riferimento non è stata trovata.).

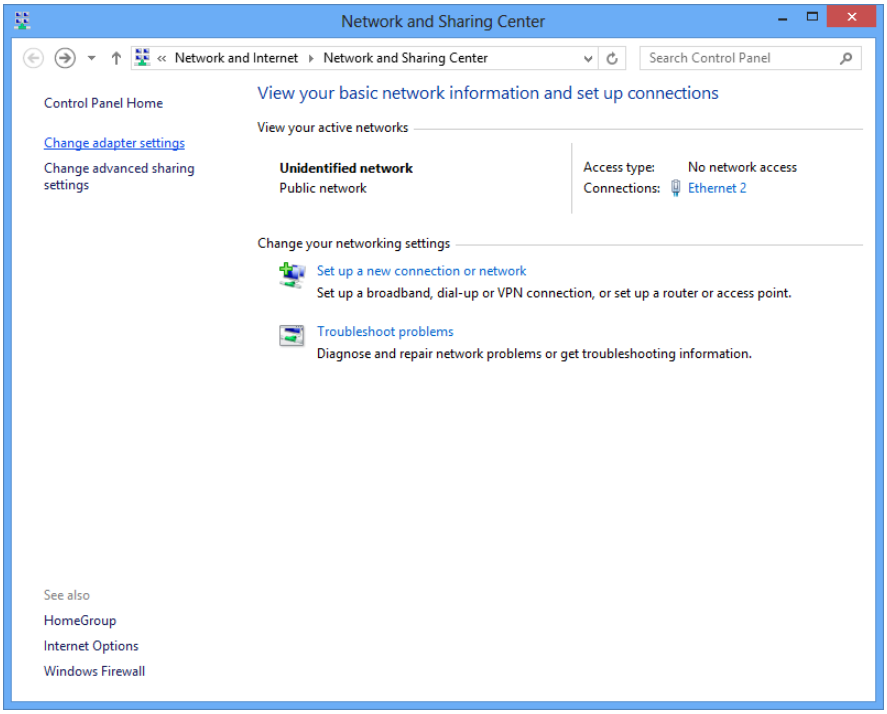


Fig. 10.7: The Network and Sharing Center window

   b. Click on ”Change adapter settings”.
   c. Right click on the Ethernet icon and select ”Properties” from the slide menu (**Fig. 10.8**).

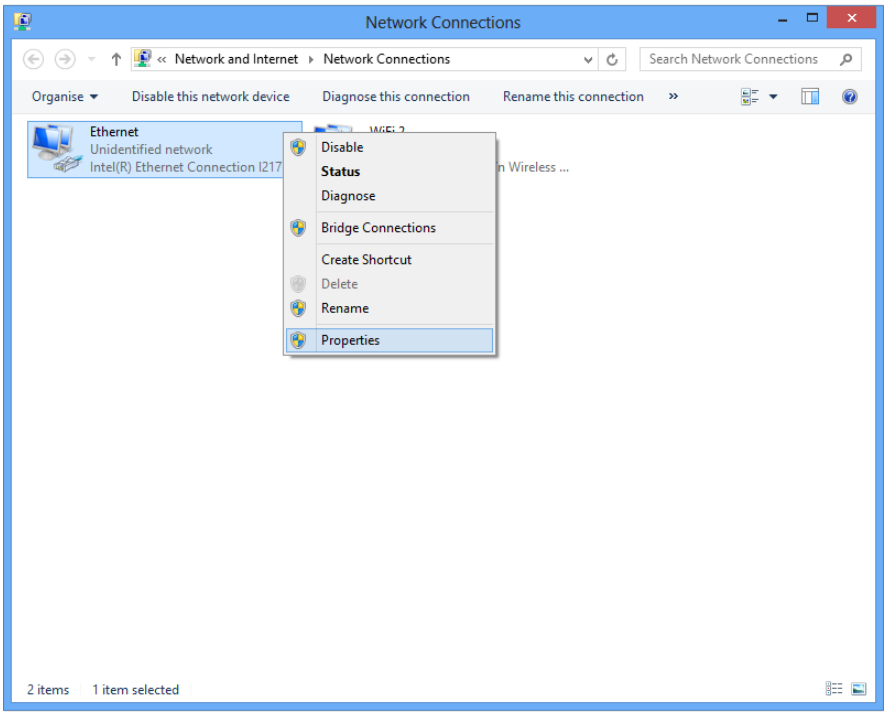


Fig. 10.8: Properties window of the Ethernet network

d. Click on ”Internet Protocol Version (TPC/IPv4)”, and select ”Properties” (**Fig. 10.9**).

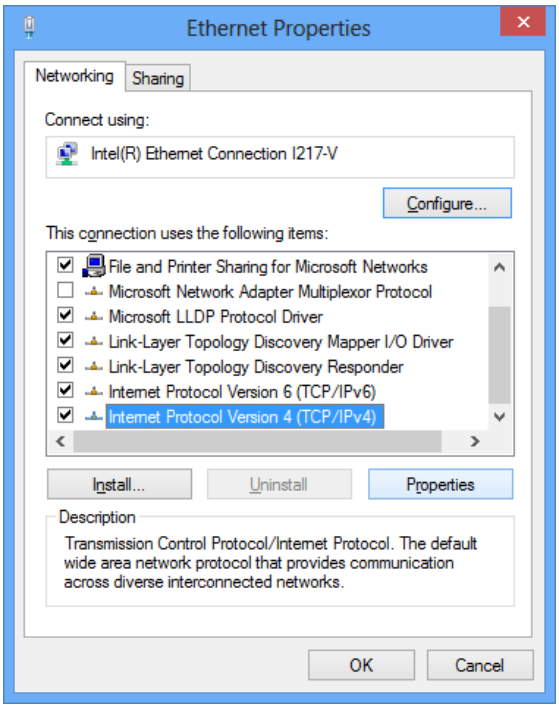


Fig. 10.9: Ethernet Properties window”

e. Copy the configuration as in **Fig. 10.10** on the ”Internet Protocol Version (TPC/IPv4) Properties” window.

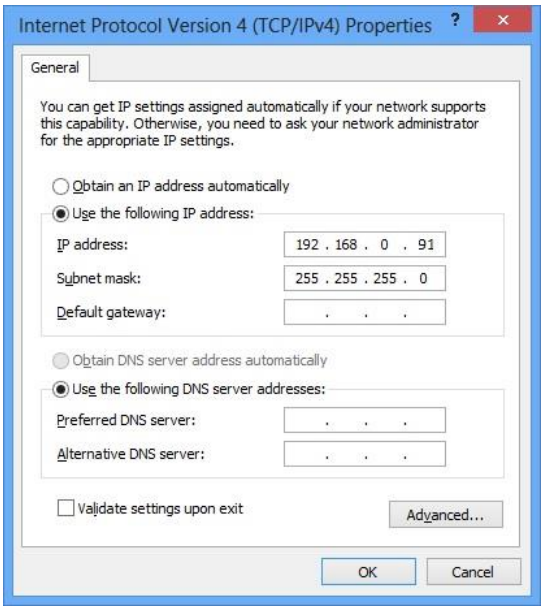


Fig. 10.10: Properties window of the ”Internet Protocol Version (TPC/IPv4)”

Note: Considering the factory default IP address of Hexagon is 192.168.0.90, use for the Network settings of the PC the same first three IP Address numbers of the Hexagon IP, while make sure to set the fourth decimal number to a value different from the Hexagon one to avoid IP conflicts over the net.

It is finally possible to connect to Hexagon by the Web Interface described in the next section (Sec. **10.3**).

10.3 Web Interface

Note: Check that the DT5495 is powered on, the Ethernet cable is connected, and the Network is set according to Sec. **10.2**.

Open the web browser and type the DT5495 IP Address as web address. In the default case, the IP Address is 192.168.0.90:

http://192.168.0.90

The Web Interface will enter the "Instrument Information" page (**Fig. 10.11**).

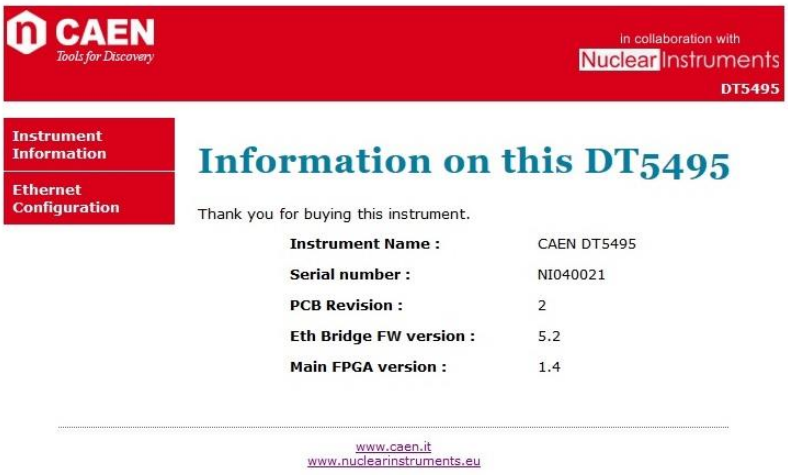


Fig. 10.11: Instrument Information page of the DT5495 Web Interface

Information includes the name, the serial number, the PCB revision, the revision of the Ethernet Bridge and the Main FPGA firmware version.

Selecting "Ethernet Configuration", the Web Interface shows the page where the ethernet settings can be changed by the user.

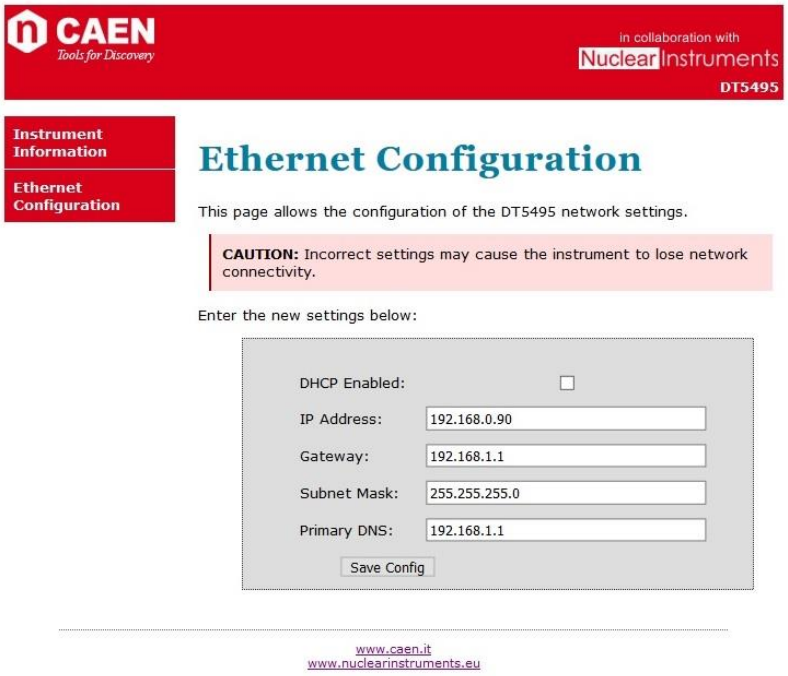


Fig. 10.12: Instrument Information page of the DT5495 Web Interface

Press the "Save Config" button after modifications and remind to power cycle the board to make the new settings effective.

10.4 Software Tools

10.4.1 CAEN Toolbox

CAEN Toolbox is the comprehensive software suite designed for CAEN Front-End boards.

With V2495, CAEN Toolbox simplifies various tasks into a few easy steps, including:

- Upgrade the MFPGA firmware
- Load the User firmware on the UFPGA
- Verify the Main and User FPGA firmware
- Read the release number of the MFPGA firmware
- Store and get the Product Unlock Code in case of paid firmware
- Get the Board Info file (useful in case of support requests)

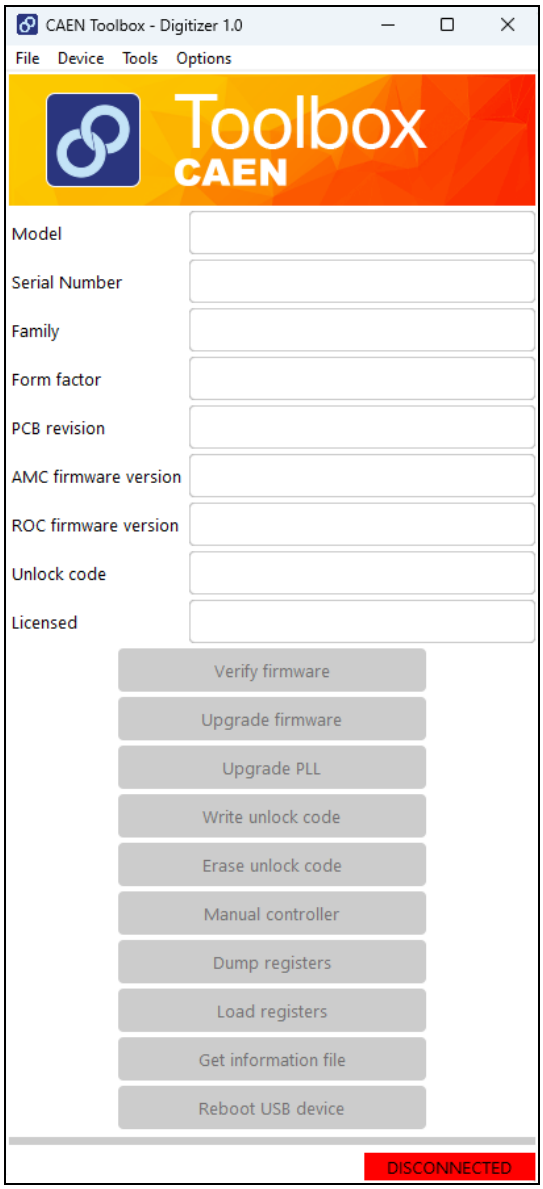


Fig. 10.13: CAEN Toolbox Graphical User Interface

CAEN Toolbox runs on Windows® and Linux® platforms. The software is stand-alone on both the available versions.

Refer to the CAEN Toolbox documentation for installation instructions and a detailed description **[RD1]**

10.4.2 PLULib Library

CAEN PLULib library is a library of C functions developed to control V2495 and DT5495 programmable logic units through direct USB and Ethernet communication interfaces, but also through the VMEbus connection by using CAEN Bridges, if supported.

The library includes a simple demo application, which is not intended for the board readout, but rather to automatically test the library functions. The user can then inspect the code as reference for his customized software development based on the PLU library.

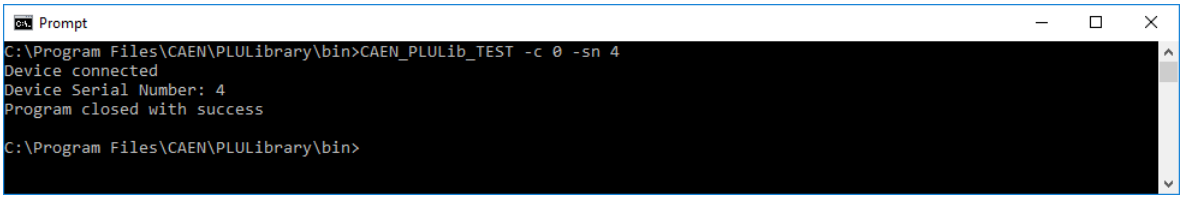


Fig. 10.14: CAEN PLULib Demo application prompt

CAEN PLULib library can operate on Windows® and Linux®, 32 and 64-bit, and requires the CAENComm library to be isntalled. The documentation **[RD3]** and installation packages are available for free download at the PLULib *Downloads* page on CAEN web site (**login required**).

10.4.3 PLUscaler_daq

PLUscaler_daq is a CAEN software dedicated to manage data acquisition from DT5495 and V2495 Programmable Logic Unit with the FW2495SC firmware (see Chap. **14**). It is a C-based simple application without a graphical user interface which relies on an external configuration text file.

The source code included in the package gives to the user a useful starting point to develop customized software applications.

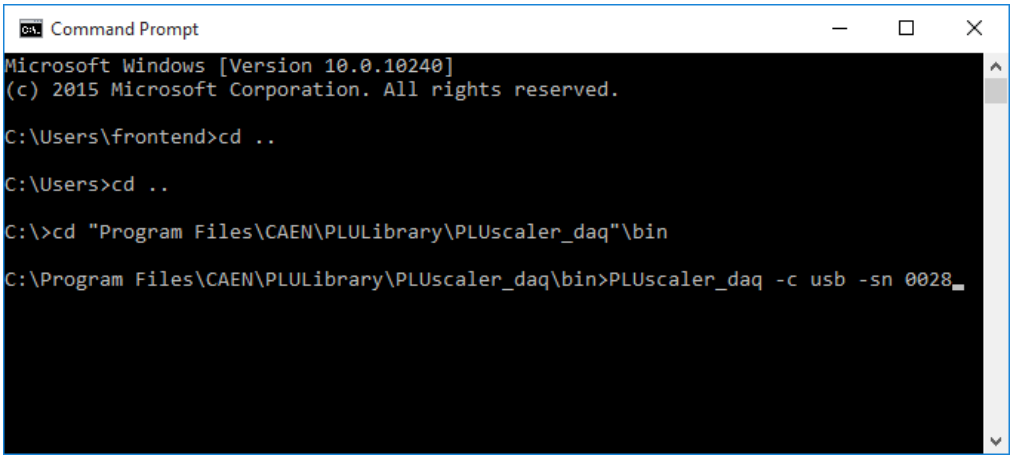


Fig. 10.15: PLUscaler_daq prompt

PLUscaler_daq can operate on Windows® and Linux®, 32 and 64-bit, and requires the PLULib library to be installed. The documentation **[RD4]** and installation packages packages are available for free download at the FW2495SC *Downloads* page on CAEN web site (**login required**).

11 Communication Interfaces

The DT5495 board can be accessed via both Ethernet and USB interface. All registers are accessible from both interfaces.

11.1 Address Map

The following table illustrates the DT5495 address map:

Address	Description
0x0000-0x0FFF	UFPGA data access
0x1000-0x7FFF	UFPGA register access
0x8000-0x80FF	*reserved*
0x8100-0x81FF	Configuration ROM
0x8200-0x83FF	Configuration and Status Registers
0x8400-0x84FF	Reserved
0x8500-0x86FF	MFPGA flash configuration
0x8700-0x88FF	UFPGA flash configuration
0x8900-0x8AFF	GDG flash configuration
0x8B00-x8BFF	Reserved
0x8C00-0x8FFF	Internal scratch RAM
0x9000-0xFFFF	reserved

Tab. 11.1: DT5495 register address map

11.1.1 User FPGA Data Access (0x0000-0x0FFF)

The UFPGA data access space is allocated specifically for the readout of data produced in the UFPGA logic which need to exploit the maximum readout throughput available. This space can be accessed by using the available block transfer mechanism over the Ethernet or USB communication interfaces.

Block data transfer allows to implement a faster readout. Block transfers requests over Ethernet or USB do not trigger any transfer over the internal local bus between the MFPGA and UFPGA: a data prefetch mechanism, implemented in the MFPGA, stores User data into a local data queue (prefetch data queue). Please, refer to the prefetch mechanism description in Sect **Local Bus Interface**. Consequently, data from the UFPGA data access space are read from the prefetch data queue in the MFPGA.

11.1.2 User FPGA Register Access (0x1000-0x7FFF)

UFPGA register address range is allocated for User register read/write.

A read or write transfer (single cycle) can be triggered on the local bus by performing a corresponding read transfer in the 0x1000-0x7FFF address interval.

The local bus master in the MFPGA acts as a transparent bridge between the communication interface (ETH/USB) and the local bus in this interval. A couple of examples are given to illustrate the transparent bridge behaviour.

11.1.3 Configuration ROM (0x8100-0x81FF)

The following registers contain module configuration information according to **Tab. 11.2**.

Only the 8 LSBs of each location of the Configuration ROM are significant.

Description	Address	Content
checksum	0x8100	An eight bit 2's complement binary checksum. The sum of the bytes starts from offset 0x8104 for the number of bytes specified in the length field (inclusive)
checksum_length2	0x8104	0x00
checksum_length1	0x8108	0x00
checksum_length0	0x810C	0x20
constant2	0x8110	0x84
constant1	0x8114	0x84
constant0	0x8118	0x01
c_code	0x811C	0x43
r_code	0x8120	0x52
oui2	0x8124	0x00
oui1	0x8128	0x40
oui0	0x812C	0xE6
version	0x8130	0x00
board2	0x8134	0x00
board1	0x8138	0x09
board0	0x813C	0xBF
revis3	0x8140	0x00
revis2	0x8144	0x00
revis1	0x8148	0x00
revis0	0x814C	PCB revision
sernum3_32	0x8170	PID 32 bit (LSB) / *reserved*
sernum2_32	0x8174	PID 32 bit / *reserved*
sernum1_32	0x8178	PID 32-bit / *reserved*
sernum0_32	0x817C	PID 32 bit (MSB) / *reserved*
sernum1_16	0x8180	Serial Number (MSB) / PID 16 bit (MSB) / 0xFF
sernum0_16	0x8184	Serial Number (LSB) / PID 16 bit (LSB) / 0xFF

Tab. 11.2: ROM Address Map of the DT5495

Note: The oui0/1 fields represent CAEN Manufacturer identifier (IEEE OUI), which is equal to 0x40E6.

Note: Old boards are identified by the Serial Number. It is a 16-bit value (decimal number between 0 and 9999) readable by accessing at 0x8180 (MSB) and 0x8184 (LSB) addresses: if the serial number on the module front panel is 1245 (hex 0x4DD), for instance, the Serial Number's MSB (0x8180) will be 0x04, while the Serial Number's LSB (0x8184) will be 0xDD.

On new boards, the Serial Number has then been replaced by the PID. It initially was a 16-bit value (decimal number between 10000 and 65535) readable by accessing 0x8180 (MSB) and 0x8184 (LSB) addresses.

On very new boards, the PID was finally extended to 32 bits and readable by accessing 0x817C (MSB), 0x8178, 0x8174, and 8170 (LSB) addresses, while registers 0x810 and 0x8184 are fixed to 0xFF.

11.1.4 Configuration and Status Registers (0x8200-0x83FF)

Address	Register/Content	Read/Write
0x8200	MFPGA firmware revision	R
0x8204 ÷ 0x8214	*reserved*	
0x8218	Software reset	W
0x8220	Scratch register	R/W
0x8224 ÷ 0x83FC	*reserved*	-

Tab. 11.3: CSR registers

11.1.5 MFPGA Firmware Revision Register

Bit	Description
[31:16]	reserved
[15:8]	Major revision number
[7:0]	Minor revision number

11.1.6 Software Reset Register

Bit	Description
[31:0]	The value written into this register will determine the kind of software reset: 1 = generate a local bus reset only (nLBRES local bus signal) Others = reserved for future options

11.1.7 Scratch Register

Bit	Description
[31:0]	This register allows to perform 32-bit accesses for test purposes. Default value is 0xAAAAAAAA

11.1.8 Flash Configuration (0x8500-0x8AFF)

This address range is reserved to flash remote programming.

Note: access to the FLASH is through the provided functions of the PLULib library (see Sec. **10.4**).

11.1.9 Internal Scratch SRAM(0x8C00-0x8FFF)

This area is available either for test access or for volatile data storage. Any address in this interval is implemented by an internal RAM location.

12 Firmware Development

12.1 Introduction

The goal of this chapter is to let the User familiarize with the information and tools needed to develop proprietary firmware for the DT5495 board.

A thorough description of the main firmware functionalities will be provided; the features of essential components like the local bus interface and the gate and delay generator will also be discussed.

Basically, two ways of making firmware can be used:

- VHDL tools on the market. This method is intended for experts in FPGA programming.
- The SCI-Compiler, Graphical Programming Language for CAEN Open FPGA Boards. This new method is available for all level users, but principally for those who are not familiar with VHDL or other FPGA programming languages.

Next section briefly introduces SCI-Compiler and gives references to find all the detailed information needed.

Following, the focus is on the VHDL method, describing also four demos provided in the UFPGA flash memory. A step-by-step guide will be given to run the examples. Their goal is to provide an extensive overview of the main functionalities of the DT5495 board and of its mezzanine cards. The code of these demos, together with a firmware template, can be downloaded from the CAEN website. When developing custom projects, it is recommended to start with the included template firmware, as it includes the correct FPGA pinning and constraints. A short description of the user top-level I/O signals will also be given.

The DT5495 board is the desktop format of an upgraded version of the V1495 board. The two boards have the same I/O connectors and mezzanine card support. On the firmware development side, there are differences, mainly due to the new Gate and Delay Generator and to the newly introduced USB connection and Ethernet one. Please refer to Sec. **12.6** for the details to port the V1495 projects on the DT5495 board.

Note: Due to the different I/O pin mapping and FPGA devices, the FPGA bit-streams and the projects previously developed for the V1495 board are not compatible with the DT5495 board.

12.1.1 Safety Notice

When working with mezzanine boards featuring input channels, particular attention must be paid during the firmware development.

The signals from such channels, after electrical conditioning on the mezzanine, get into the USER FPGA inputs. At power-on, if the firmware loaded on the FPGA drives those inputs without the appropriate precautions, severe conflicts may be generated, and hardware components could be damaged if the current requirements exceed the current limit provided by the motherboard on the 3V3 supply voltage.

CAUTION: Firmware development requires special attention.

WITHOUT PROPER ATTENTION, LOADING ON THE USER FPGA A FIRMWARE DRIVING THE INPUTS OF MEZZANINE BOARDS MAY GENERATE CONFLICTS LEADING TO SEVERE DAMAGE OF HARDWARE COMPONENTS.

Some useful methods for coding the firmware to avoid conflicts are described in Sec. **12.3.1**.

12.2 Sci-Compiler

Sci-Compiler represents an innovative approach to simplify firmware development on CAEN Open FPGA boards. This software utilizes a block-diagram-based approach that incorporates a range of high-level functionalities (blocks) to conceal the real firmware coding. As a result, it becomes possible to create and compile custom firmware from a basic block diagram structure.

The power of Sci-Compiler is to make it possible to program a FPGA without strictly requiring knowledge of FPGA programming languages.

Please, find all the information and material about Sci-Compiler, and how to use it with your CAEN Programmable Logic Unit, on the dedicated website:

https://www.sci-compiler.com/

12.3 User FPGA I/O ports

This section illustrates the I/O ports of the UFPGA. Port names are the same of the VHDL entity top-level ports used in the template and demo firmware.

- **Clock:** The user FPGA receives the board system clock

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
CLK	1	Input	50 MHz system clock

Tab. 12.1: Clock ports description table

- **Mainboard Robinson Nugent connector ports:** the ports A and B are input only. They feature 32 input channels that reach the UFPGA as 32 single-ended LVTTL lines (3.3 V).

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
A	32	Input	A-port values
B	32	Input	B-port values
C	32	Output	C-port value

Tab. 12.2: Mainboard Robinson-Nugent connector description table

- **LEMO ports:** each of the LEMOs on the front panel can be used as input or output. The signals used are the following:

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
GIN	2	Input	Input values for G0, G1
GOUT	2	Output	Output values for G0, G1
SELG		Output	NIM/TTL selector 0 = NIM 1 = TTL
nOEG		Output	Output enable (active low) (0=output, 1=input)

Tab. 12.3: LEMO G ports description table

- **Expansion I/O ports**: for each mezzanine, **Tab. 12.4** shows the supported signals (X = D, E, F).

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
IDx	3	Input	Card ID “000” = A395A “001” = A395B “010” = A395C “011” = A395D “100” = A395E “111” = No mezzanine present
SELx	1	Output	NIM/TTL selector 0 = NIM 1 = TTL
nOEx	1	Output	Output enable (active low) 0 = output 1 = input
X	32	Input/Output	Data bus to/from the mezzanines

Tab. 12.4: Expansion I/O ports description table

Please note that:

- The nOEx signal is meaningful for the A395B, A395C and A395D mezzanines, while SELx is meaningful only for the A395D mezzanine card. If the A395D is not used, the nOEs and SELx signals can be left undriven. When the A395B or A395C are used, the nOEx signal must be set low (nOEx = 0).
- Mezzanines with 32 I/O signals have a one-to-one correspondence between physical signals and I/O pins (i.e. X[i] corresponds to the i-th channel of the mezzanine).

The A395D mezzanine card also allows its 8 channels to be individually used as inputs.

The steps to do are:

- set nOEx to 0 (all channels are outputs);
- terminate to 50 Ohm the channel(s) intended to be used as input.

The second step is possible by moving to 1-2 the relevant jumper on the mezzanine board which relies next to each channel, as in **Fig. 12.1**.

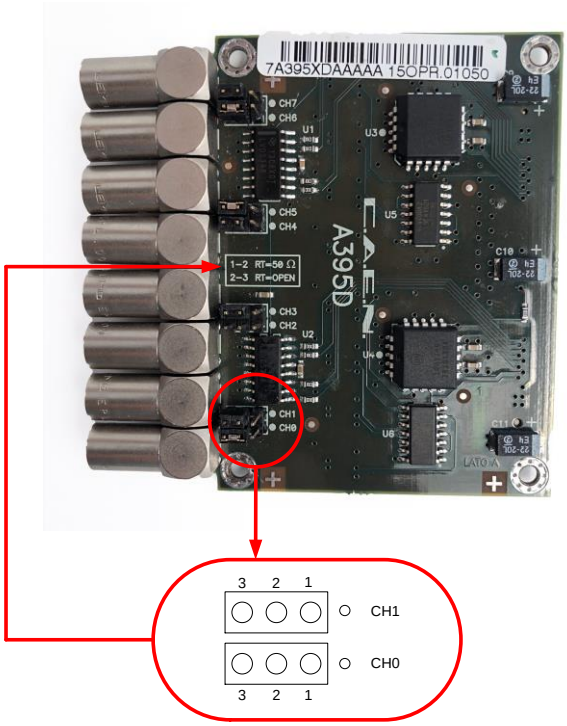


Fig. 12.1: A395 jumpers for the channel impedance selection

In case the A395D mezzanine is used, the mapping between the external channels and the X bus is shown in **Tab. 12.5**. A schematic view of the A395D board is shown in **Fig. 12.2**.

CHANNEL	INPUT SIGNALS	OUTPUT SIGNALS
0	X[2]	X[0]
1	X[18]	X[16]
2	X[3]	X[1]
3	X[19]	X[17]
4	X[14]	X[12]
5	X[30]	X[28]
6	X[15]	X[13]
7	X[31]	X[29]

Tab. 12.5: A395D mapping

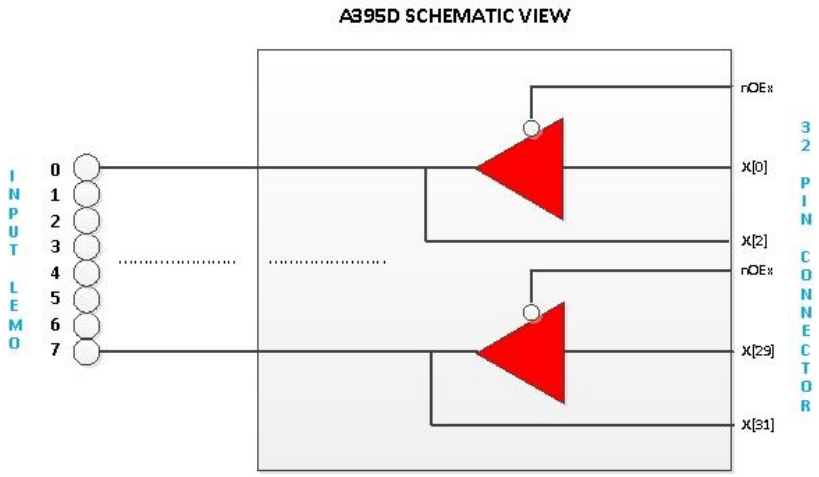


Fig. 12.2: CAENComm Demo Java and LabVIEW graphical interface

For example:

- A395D channel 0 can be driven with signal X[0] and its status can be read from X[2].
- A395D channel 1 can be driven with signal X[16] and its status can be read from X[18].

➢ **User LEDs:** Refer to the eight User LEDs on the front panel (see Chap. **4**).

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
LED	8	Output	LED drivers

Tab. 12.6: LED ports description table

In order to switch on one of the eight LEDs on the front panel, the corresponding signal (LED[7:0]) should be set to 1.

➢ **Gate & Delay Generator interface ports:** The GDG is connected to the UFPGA via the following signals:

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
GD_START	32	Output	GDG start signals
GD_DELAYED	32	Input	GDG output signals
SPI_MISO	1	Input	SPI MISO (Master Input Slave Output)
SPI_SCLK	1	Output	Serial clock
SPI_CS	1	Output	SPI Chip Select
SPI_MOSI	1	Output	SPI MOSI (Master Output Slave Input)

Tab. 12.7: Gate and Delay Generator ports description table

➢ **Local Bus ports:** The Local Bus signals are shown in **Tab. 12.8**.

SIGNAL NAME	WIDTH	TYPE	DESCRIPTION
nLBRES	1	Input	Bus reset (active low)
nBLAST	1	Input	Last cycle (active low)
WnR	1	Input	Write/Read cycle 0 = read 1 = write
nADS	1	Input	Address strobe (active low)
nREADY	1	Output	Slave ready/Prefetch request
LAD	16	Input/Output	Data/address bus

Tab. 12.8: Local Bus ports description table

12.3.1 Preventing User FPGA Input Conflicts

In this section, some technics are described for the firmware development to avoid conflicts when loading custom firmware on the USER FPGA which drives the inputs of an A395x mezzanine (see p. **8**).

- Put the following constant declarations in a package or in the architecture declarative region:

```
constant A395A_ID_CODE : std_logic_vector(2 downto 0) := "000" ;
constant A395B_ID_CODE : std_logic_vector(2 downto 0) := "001" ;
constant A395C_ID_CODE : std_logic_vector(2 downto 0) := "010" ;
constant A395D_ID_CODE : std_logic_vector(2 downto 0) := "011" ;
constant A395E_ID_CODE : std_logic_vector(2 downto 0) := "100" ;
```

- Put the code below into your top-level architecture, where nOED, nOEE and nOEF are the top-level ports to control mezzanine I/O direction for port D, E and F respectively; IDD, IDE and IDF are the top-level ports to sense the mezzanine ID code for auto-detection; d_oe_n_reg_i, e_oe_n_reg_i and f_oe_n_reg_i are the user internal signals to drive mezzanine direction.

  If the mezzanine being detected on a slot has inputs, this code avoids driving them.

```
nOED <= d_oe_n_reg_i and d_oe_n_auto;
nOEE <= e_oe_n_reg_i and e_oe_n_auto;
nOEF <= f_oe_n_reg_i and f_oe_n_auto;

d_oe_n_auto <= '0' when ((IDD = A395B_ID_CODE) or (IDD = A395C_ID_CODE) or
(IDD = A395E_ID_CODE)) else
               '1';

e_oe_n_auto <= '0' when ((IDE = A395B_ID_CODE) or (IDE = A395C_ID_CODE) or
(IDE = A395E_ID_CODE)) else
               '1';

f_oe_n_auto <= '0' when ((IDF = A395B_ID_CODE) or (IDF = A395C_ID_CODE) or
(IDF = A395E_ID_CODE)) else
               '1';
```

12.4 Local Bus Interface

The Local Bus (LB) is the communication interface between the MFPGA and the UFPGA.

It is made of a Master (LBM), implemented inside the MFPGA, and of a Slave (LBS), implemented inside the UFPGA. The communication protocol between the LBM and the LBS is based on the following signals. In what follows OUT(IN) means that the signal goes from the LBM(LBS) to the LBS(LBM) and AL stands for active-low:

- nADS (OUT, AL): it is set by the LBM when a r/w operation is performed. The LBS samples the register address from LAD when nADS is set.
- WnR (OUT): it notifies the LBS whether the access is in write- (WnR=1) or read- (WnR=0) mode. The value of WnR is relevant only when nADS is set.
- nREADY (IN, AL): It is a dual-purpose signal. it is set by the LBS to notify the LBM that the LBS is ready for r/w data transfer. It can also be used for the data prefetch mechanism (see next).
- nBLAST (OUT, AL): it signals the last cycle of a data transfer. It is set when either the LBS is not ready, or the LBM, cannot accept data (e.g. because the FIFO used for data prefetch is full).
- LAD (IN/OUT): the 16-bit bus used to read/write both address and data.

LBM can generate two kind of transfers over the local bus:

- Single word (32-bit) read/write transfer
- Multi word prefetch transfer

Single word transfers are initiated by the LBM for register access. It it composed by a single clock address phase and two 16-bit data phases. A data phase is completed if the LBS asserts nREADY low. The transfer ends on the second data phase when the LBM asserts the nBLAST signal low.

Examples of the signal logic taken using Altera™ Signal Tap are shown in **Fig. 12.3** and **Fig. 12.4**. For a read and a write access on the LBS.

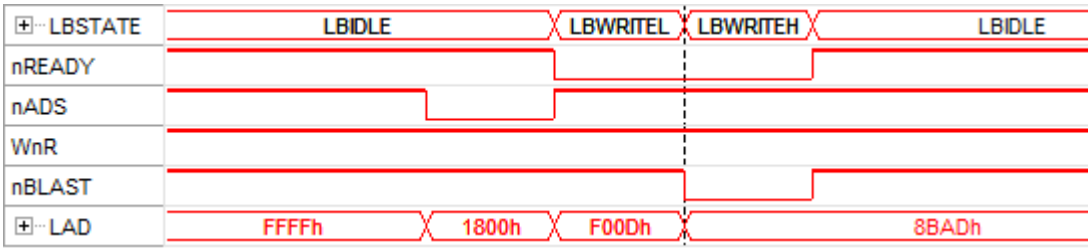


Fig. 12.3: Local bus signals at a write access (x8BADF00D is written on register x1800)

Fig. 12.4: Local bus signal at a read access (x8BADF00D is read from register x1800)

A data prefetch mechanism, implemented in the MFPGA, stores User data into a MFPGA local data queue (prefetch data queue). The MFPGA prefetch queue data is available for readout over the Ethernet or USB interface (see Sec. **11.1**).
The prefetch mechanism is illustrated by **Fig. 12.5**.

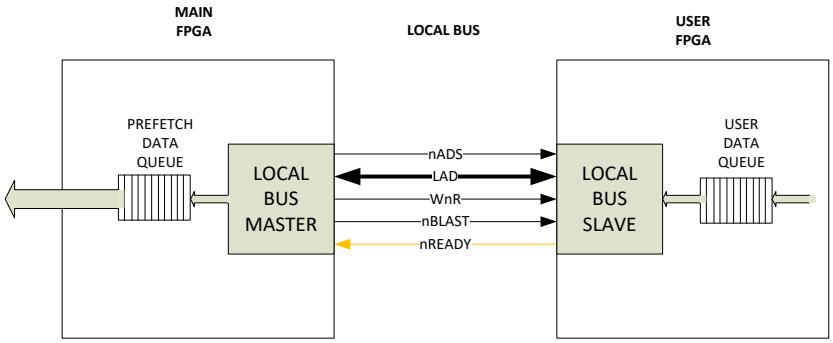


Fig. 12.5: Prefetch mechanism

Prefetch read cycles are initiated by the UFPGA logic: nREADY local bus signal must be used by the User logic to request a new data prefetch cycle by the local bus master (MFPGA): The request is active when the nREADY is low. The local bus master should have an internal prefetch queue (FIFO component). If a user request is pending and the local bus master has available space into the prefetch queue, it will start an internal block transfer from local bus address 0x0000 (fixed). The User logic is required to respond only to address 0x0000: data from its data queue are made available to the DT5495 local bus. If a new register access is requested to the local bus master while a prefetch cycle is ongoing, the prefetch cycle will be interrupted by asserting the nBLAST signal and the register access will be served with priority. At the end of a register access, the user logic can keep the nREADY signal active to request a new prefetch cycle to the LBM, which will fetch new data in its queue.

Please refer to the lb_int.vhd source code in the User demo firmware projects.: it is a Local Bus slave implementation provided by CAEN as a reference.
The lb_int.vhd found in the Pattern Recorder demo firmware (see Sec. **13.5**) is a demonstration of user logic implementation of the data prefetch mechanism.

12.5 Gate and Delay Controller

12.5.1 General Description

The GDG can be used by the user logic to gate and delay up to 32 digital signals. The delay and gate values can be set using a dedicated serial bus (SPI bus). The signals are sent to the GDG, where they are gated and delayed according to the user settings and then returned to the UFPGA. The width of the delayed signal only depends on the gate value set and is not related to the width of the incoming signal.

The gate and delay values are 16-bit wide and **their sum cannot exceed 65535 (0xFFFF).** They are both generated by using an internal clock, which computes the number of clock cycles between the incoming signal and the leading and trailing edge of the gate signal. Referring to **Fig. 12.6**, the times of occurrence of the leading (T_{lead}) and trailing (T_{trail}) can be obtained using the following formulas, where N_g(N_d) is the gate(delay) value set by register and T_0, T_1 are the time values reported in **Tab. 12.9**:

T_{lead} = 0 if N_d=0

T_0+T_1(Nd-1) if N_d>0

T_{trail} = 0 if N_d=N_g=0

T_0+T_1(N_d+N_g-1) if (N_d+N_g)≥1

The relationship between T_{lead}, T_{trail} and the delay and gate duration is:

T_{delay}=T_{lead}

T_{gate}=T_{trail}-T_{lead}

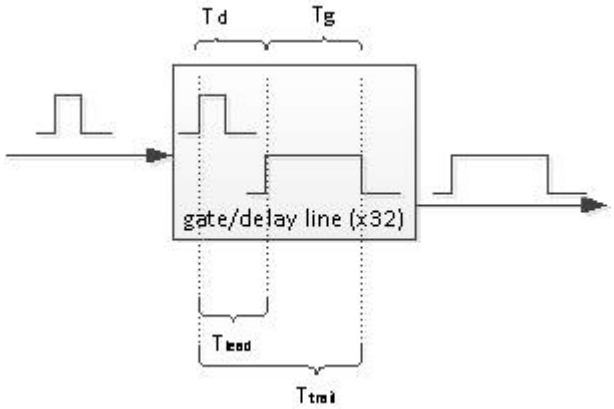


Fig. 12.6: Gate and Delay parameters representation

T_1 represents the minimum increment of either the gate or the delay value. The presence of T_0 is due to a slightly longer duration of the minimum delay time increment (or gate increment, if the N_g=0) with respect to the standard T_1 increment. It should be noted that both T_1 and T_0 can vary from channel to channel, with an expected variation interval of ±10% for both. Please also consider that, when setting N_d=0, the observed delay of the outgoing signal with respect to the incoming one is due to the paths on the PCB and inside the FPGAs, which are delay channel-dependent.

Example 1: If the delay value is set to 0x0 and the gate value is set to 0x5, the gate will last 12.2+10.7*4=55 ns.

Example 2: If the delay value is set to 0x8 (typ. 12+7*10.7=86.9 ns), the maximum possible gate value is 0xFFF8 (typ. 65527*10.7 ≈ 701 µs).

Example 3: If the delay value is set to 0xFFF8 (≈ 701 µs), the ma maximum possible gate value is 0x8 (typ. 86.9 ns).

PARAMETER	VALUE
T_0	12 ns (typ.) ± 10%
T_1	10.7 ns (typ.) ± 10%
T_{max}	T_0 + T_1 (2^{16}-1)

Tab. 12.9: Main parameters of the Gate and Delay Generator

The UFPGA can set the relevant configuration registers and read data from the generator through the SPI protocol. CAEN does not provide a detailed description of the communication protocol over the SPI bus for GDG configuration. A VHDL component is provided in the gd_control.vhd source code (GD_CONTROL), included in all the user demo firmware.

GD_CONTROL implements the following registers that can be accessed from the local bus. The LB_INT component reserves addresses starting from 0x7F00 to GD_CONTROL access. LB_INT and GD_CONTROL are connected by a dedicated interface for register write and read.

A set of LB registers are used to configure the GDG, as described in **Tab. 12.10**:

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x7F00	Data Write	A24/A32	D16/D32	R/W
0x7F04	Command register	A24/A32	D16/D32	R/W
0x7F08	Control register	A24/A32	D16/D32	R/W
0x7F0C	Data read	A24/A32	D16/D32	R
0x7F10	Status register	A24/A32	D16/D32	R

Tab. 12.10: Local Bus registers description for the Gate and Delay Generator configuration

12.5.2 Register Description

- **DATA WRITE:** the datum (gate or delay) must be written here before it is transmitted through SPI to the GDG.

  Address: 0x7F00.

  Mode: Read only.

Bit	Description
[31:0]	Gate or delay datum to transmit to the GDG

- **COMMAND register:** it is where the action to be performed is set. The action depends on the value of the 2 LSBs of the Control register (read or write mode): one of these bits should be asserted to execute the related action. The channel interested by the action is set through the 6 LSBs of the register (relevant when [15:0] = 0x100 in write mode and [15:0] = 0x2000, 0x3000 in read mode).

  Address: 0x7F04

  Mode: Read only

Write Mode

Bit	Description
[31:16]	*reserved*
[15:0]	0x2000 = the internal gate value is modified with the content of the data write register (but not sent to the DFPGA) 0x3000 = the internal delay value is modified with the content of the data write register (but not sent to the DFPGA) 0x0100(+channel) = the gate AND delay values of the selected channel are modified with the internal values, sent to the DFPGA by the UFPGA through SPI 0x0300 = the delay of all channels is reset 0x0400 = the delay of all channels is calibrated 0x0500 = the internal gate and delay values are broadcast to all 32 channels

Read Mode

Bit	Description
[31:16]	*reserved*
[15:0]	0x2000(+channel) = the gate value of the specified channel is read through SPI 0x3000(+channel) = the delay value of the specified channel is read through SPI

- **CONTROL register:** bit[0] sets a write or read action.

  Address: 0x7F08.

  Mode: Read only.

Bit	Description
[31:2]	*reserved*
[1]	1 = a read action is performed
[0]	1 = a write action is performed

- **DATA READ:** it is where the datum (gate or delay) can be read after an SPI read access to the GDG has been performed (see the Read Mode of the COMMAND register).

  Address: 0x7F0C.

  Mode: Read only.

Bit	Description
[31:0]	Gate or delay datum resulting from READ MODE access to GDG

- **STATUS register**

  Address: 0x7F10.

  Mode: Read only.

Bit	Description
[31:1]	*reserved*
[0]	Gate and delay ready signal (when 0, the Gate and Delay controller is busy. A new command to the Gate and Delay controller should not be issued before this bit sets back to 1)

12.5.3 Example Procedures

- **MODIFY THE GATE OR THE DELAY VALUE OF A CHANNEL**

The gate or the delay of a signal from the UFPGA can be modified by following this procedure (the registers quoted are accessible via VME):

- Set a delay (gate) value in the DATA WRITE register
- Set the COMMAND register to 0x2000 (0x3000) to modify the delay (gate) internal value
- Set the control register to 0x1/0x0
- Set the COMMAND register to 0x100 + CHANNEL (6 bits) to write the gate and delay values on the FPGA through the SPI bus, for the channel specified by the six LSBs. Note that the gate and delay values are written in the same action
- Set the CONTROL register to 0x1/0x0

- **READ THE GATE OR THE DELAY VALUE OF A CHANNEL**

- Set the COMMAND register to 0x2000 (0x3000) + CHANNEL (6 bits) to read the delay (gate) value of a specific channel
- Set the CONTROL register to 0x2/0x0
- Read the required value in the DATA READ register

- **BROADCAST A GATE OR DELAY VALUE OF A CHANNEL**

- Set the desired values of the delay and gate values (see the first example, first two steps)
- Set the COMMAND register to 0x500
- Set the CONTROL register to 0x1/0x0

Note that in the case of a broadcast command the delay and gate values are serialized to the GDG at the same time. The bits related to the channel index (six LSBs of the CONTROL register) are irrelevant in this case.

12.6 Porting V1495 to DT5495

Firmware designs targeted to the V1495 User FPGA can be ported to DT5495 by keeping in mind the following differences between the two modules:

- The DT5495 FPGA device is different (Cyclone V versus Cyclone I) and with larger logic resources. While there will be no resource constrains, the FPGA configuration scheme and output binary files are different for the two boards.
- A design that used on-board delay lines with V1495 must be redesigned to consider the new features and characteristics of the DT5495 GDG.
- The address map of the V1495 VME FPGA and DT5495 MAIN FPGA have some differences to consider:
    - Registers in V1495 are mapped at 16-bit aligned addresses, while DT5495 features a 32-bit aligned address map. Register access should always be in 32-bit mode.
    - Common registers with different address:
        - Firmware revision:
            - 0x800C in V1495
            - 0x8200 in DT495
- The local bus map is the same for V1495 and DT5495. Block transfers are allowed only in the 0x0000-0x0FFF address interval. The same user data prefetch mechanism is implemented. Local bus register accesses are mapped to the same 0x1000-0x7FFF address interval.

Note: The DT5495 does not require the 0x100C local bus address to be reserved for user firmware revision register with special constraints due to the firmware licensing mechanism. In DT5495, all user registers can be implemented in the allowed address range without any constraints of functionality or content

- The flash access register map and protocol are different between the two boards: the specific setting and tools must be used for each module.

13 Demo Firmware

13.1 Introduction

This chapter describes a set of UFPGA demos that CAEN provides as demonstration of the DT5495 functionalities and as a guide for developing customized user application firmware.

The wanted demo can be stored only on the application 1 page of the UFPGA FLASH memory.

To run a firmware "as it is", it is required to use the functions of the CAEN PLULib library to access the proper registers (refer to Sec. **10.4**). To modify demos and upload the new firmware on the FPGA, a basic knowledge of VHDL language and Altera Quartus II software are required.

In what follows, the features of the four demos loaded in the UFPGA's flash memory will be described.

By default, each demo lights front panel user LED with the following configurations:

Demo	Description
Demo1	LED 0 ON; others OFF
Demo2	LED 1 ON; others OFF
Demo3	LED 1 and LED 0 ON; others OFF
Demo4	LED 2 ON; others OFF

13.2 Demo Structure

All demo projects feature a common file structure:

- A configuration file (**V2495_package.vhd**), allowing to set the number of CONFIG and MONITOR registers, their respective base address, the demo ID and the firmware revision. The CONFIG registers are mapped between 0x1000 and 0x17FC, while the MONITOR registers are mapped between 0x1800 and 0x1FFC.
- A gate and delay block (**gd_control.vhd**), acting as a controller for the GDG (the example in Sec. **13.7** specifically focuses on the configuration of GDG delays);
- The local bus slave (**lb_int.vhd**), allowing to read/write the registers needed in the demo operations;
- The SPI master (**spi_master.vhd**) and interface (**spi_interface.vhd**), allowing the communication to/from the GDG to be serialized;
- A demo-specific part, containing core demonstration logic.

13.3 Demo Setup

To run the demo firmware just stored in the application 1 page of the UFPGA FLASH memory, the following instructions should be followed:

- Connect to the DT5495 board to read/write registers
- Set the required configuration through the control register(s).

13.4 Gate Pattern Demo Description

13.4.1 Introduction

In this example, a logic combination of the input port signals A and B is sent to the C connector (32 lines). The user can choose the output among 10 available options by setting the 4 LSBs of the control register (see the register list). Apart from the value of A, B, A AND B and A OR B, the aforementioned signals in “capture” mode (a bit is set whenever a signal shows up) can be selected.

13.4.2 Register Map

In this example, 9 registers are used, 5 of which (MONITOR registers) can only be read and 4 (CONTROL registers) can be both read and written. All registers are 32 bit wide and can be accessed in single access mode.

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x1000	Firmware version	A24/A32	D16/D32	R
0x1004	Value of port A (input)	A24/A32	D16/D32	R
0x1008	Value of port B (input)	A24/A32	D16/D32	R
0x100C	Value of port C (output)	A24/A32	D16/D32	R
0x1010	Status register	A24/A32	D16/D32	R
0x1800	Mask of input port A	A24/A32	D16/D32	R/W
0x1804	Mask of input port B	A24/A32	D16/D32	R/W
0x1808	Control register	A24/A32	D16/D32	R/W
0x180C	User Value of port C	A24/A32	D16/D32	R/W

13.4.3 Register Description

- **FIRMWARE REVISION register:** Contains the firmware revision number.

  Address: 0x1000.

  Mode: Read only.

Bit	Description
[31:0]	Firmware revision progressive number

- **A-PORT VALUE register:** Contains the 32 input bits of port A.

  Address: 0x1004.

  Mode: Read only.

Bit	Description
[31:0]	Input value of Port A

- **B-PORT VALUE register:** Contains the 32 input bits of port B.

  Address: 0x1008.

  Mode: Read only.

Bit	Description
[31:0]	Input value of Port B

- **C-PORT VALUE register:** Contains the 32 output bits of port C (selectable by the user in this demo).

  Address: 0x100C.

  Mode: Read only.

Bit	Description
[31:0]	Output value of port C

- **STATUS register:** Contains status information.

  Address: 0x1010.

  Mode: Read only.

Bit	Description
[31:5]	reserved
[4]	Register clear signal
[3]	Register gate signal
[2]	External clear signal
[1]	External gate signal
[0]	Data ready

- **A-PORT MASK register:** Masks the channels of port A.

  Address: 0x1800.

  Mode: Read and Write.

Bit	Description
[31:0]	The n-th bit set to 1 means that the corresponding channel of port A is masked

- **B-PORT MASK register:** Masks the channels of port B.

  Address: 0x1804.

  Mode: Read and Write.

Bit	Description
[31:0]	The n-th bit set to 1 means that the corresponding channel of port B is masked

- **CONTROL register:** Allows to set the demo configuration.

  Address: 0x1808.

  Mode: Read and Write.

Bit	Description
[31:9]	reserved
[8]	Allows to select the signal type on ports G0, G1: 0 = NIM level, 1 = TTL level
[7:6]	Reserved
[5]	The gate of input port C ends when this bit is set
[4]	The gate of input port C starts when this bit is set
[3:0]	0x0 = the port A (possibly masked) is sent to C when a gate signal is active. 0x1 = the port B (possibly masked) is sent to C when a gate signal is active. 0x2 = the AND of A and B (possibly masked) is sent t to C when a gate signal is active. 0x3 = the OR of A and B (possibly masked) is sent t to C when a gate signal is active. 0x4 = the port A (possibly masked) is captured and sent to C when a gate signal is active 0x5 = the port B (possibly masked) is captured and sent to C when a gate signal is active 0x6 = the AND of ports A and B (possibly masked) is captured and sent to C when a gate signal is active 0x7 = the OR of ports A and B (possibly masked) is captured and sent to C when a gate signal is active 0x8 = a clock counter (@ 50 MHz) is sent to C 0x9/0xF = the value of a register is sent to C

- **C-PORT USER VALUE register:** Allows to set the C port value when Control register bits[3:0] are set to 0x9/0xF.

  Address: 0x180C.

  Mode: Read and Write.

Bit	Description
[31:0]	User value of Port C

13.5 Pattern Recorder Demo Description

13.5.1 Introduction

In this demo, a digital pattern recorder is implemented. When a start signal is received, the digital input specified through the control register is sampled at a user-configurable sampling rate, and a user-configurable number of samples are collected in a FIFO. Once all samples have been stored, they can be readout through read accesses.

13.5.2 Register Map

In this example, 9 registers are used, 5 of which (MONITOR registers) can be only read and 4 (CONTROL registers) can be both read and written.

The sampling clock can be either internal (100 MHz, generated from the 50 MHz input clock through a PLL) or external, through the G0 connector: in this case a signal compatible with one of the supported levels (NIM or TTL) should be provided.

All registers are 32 bit wide and can be accessed in single access mode.

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x0000	FIFO content readout	A24/A32	D16/D32	R
0x1000	Firmware version	A24/A32	D16/D32	R
0x1004	Value of port A (input)	A24/A32	D16/D32	R
0x1008	Value of port B (input)	A24/A32	D16/D32	R
0x100C	Value of port C (output)	A24/A32	D16/D32	R
0x1800	Mask of input port A	A24/A32	D16/D32	R/W
0x1804	Mask of input port B	A24/A32	D16/D32	R/W
0x1808	Control register	A24/A32	D16/D32	R/W
0x180C	User Value of port C	A24/A32	D16/D32	R/W

13.5.3 Register Description

- **FIRMWARE REVISION register:** Contains the firmware revision number

  Address: 0x1000.

  Mode: Read only.

Bit	Description
[31:0]	Firmware revision progressive number

- **A PORT VALUE:** the 32 input bits of port A

  Address: 0x1004.

  Mode: Read only.

Bit	Description
[31:0]	Input on port A

- **B PORT VALUE:** the 32 input bits of port B

  Address: 0x1008.

  Mode: Read only.

Bit	Description
[31:0]	Input on port B

- **C PORT VALUE**: the 32 output bits of port C (selectable by the user in this demo)

  Address: 0x100C.

  Mode: Read only.

Bit	Description
[31:0]	Output to port C

- **A-PORT MASK register:** Masks the channels of port A.

  Address: 0x1800.

  Mode: Read and Write.

Bit	Description
[31:0]	The n-th bit set to 1 means that the corresponding channel of port A is masked

- **B-PORT MASK register:** Masks the channels of port B.

  Address: 0x1804.

  Mode: Read and Write.

Bit	Description
[31:0]	The n-th bit set to 1 means that the corresponding channel of port B is masked

- **CONTROL register:** allows to set the demo configuration.

  Address: 0x1808.

  Mode: Read and Write.

Bit	Description
[31:20]	Number of stored samples in the FIFO
[19:16]	Downsampling factor (frequency divider)
[15:10]	*reserved*
[9]	Software acquisition start
[8]	Software acquisition stop
[7:6]	*reserved*
[5:4]	Output selection (port C) 00 = Port A 01 = Port B 10 = clock counter 11 = user output (register 0x180C)
[3:2]	*reserved*
[1]	The selected clock source 0 = internal clock source 1 = external clock source (G0 port)
[0]	I/O port level 0 = NIM level 1 = TTL level

- **C PORT VALUE register:** allows to set an arbitrary value to port C.

  Address: 0x180C.

  Mode: Read and Write.

Bit	Description
[31:0]	User output value for port C

13.6 DAC Demo Description

13.6.1 Introduction

In this demo, the use of a A395E mezzanine card to configure a DCC output value on its 8 output channels will be explained. The DAC registers are 16 bits wide, and the dynamic range of the output voltage is +/- 5 V (refer to **Tab. 5.6** for detailed specifications).

13.6.2 Register Map

In this example, 4 registers are used, 2 of which (MONITOR registers) can only be read and 2 (CONTROL registers) can be both read and written. All registers are 32-bit wide and can be accessed in single access mode.

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x1000	Firmware Version	A24/A32	D16/D32	R
0x1004	Mezzanine ID number	A24/A32	D16/D32	R
0x1800	Control register	A24/A32	D16/D32	R/W
0x1804	DAC register	A24/A32	D16/D32	R/W

13.6.3 Register Description

- **FIRMWARE VERSION register:** Stores the revision number of the firmware.

  Address: 0x1000.

  Mode: Read only.

Bit	Description
[31:0]	Revision Number

- **MEZZANINE ID NUMBER register:** Contains the Mezzanine ID values (see **Tab. 12.4**).

  Address: 0x1004.

  Mode: Read only.

Bit	Description
[31:12]	*reserved*
[11:8]	Mezzanine ID, port F
[7:4]	Mezzanine ID, port E
[3:0]	Mezzanine ID, port D

- **CONTROL register**

  Address: 0x1800.

  Mode: Read and Write.

Bit	Description
[31:29]	*reserved*
[28]	Activation of test mode (sawtooth signal on all channels)
[27:25]	Reserved
[24]	1 = the DAC value is written for the specified mezzanine and channel 0 = the DAC value is read for the specified mezzanine and channel (the result is put in 0x1804)
[23:22]	*reserved*
[21:20]	Selects the mezzanine for either a write or read access “00” = F “01” = E “10” = D
[19:16]	The 3 LSBs [18:16] select the channel for either a write or read access. If a write access is performed and [19] = 1 all channels are selected for writing; if a read access is performed [19] is ignored
[15:0]	DAC value

- **DAC register:** Contains the DAC value of the channel specified by bits[21:16] of the register 0x1800.

  Address: 0x1804.

  Mode: Read and Write.

Bit	Description
[31:0]	DAC value

Please, note that the DAC value obtained from register 0x1804 is not read from the chip, but it is the last set value for that specific channel.

13.7 Gate and Delay Demo Description

13.7.1 Introduction

In this demo the functionalities and use of the Gate and Delay Generator will be shown. The signal to be delayed can be selected between an internally generated clock of configurable frequency (submultiples of 50 MHz) and the input from the G0 connector (please refer to Sec. **3.5** for the G0 and G1 port configuration required in this case). An output signal of configurable width and delay with respect to the input signal's leading edge will be available on G1.

13.7.2 Register Map

In this example, 3 registers are used, 1 of which (MONITOR register) can be only read and 2 (CONTROL registers) can be both read and written. All registers are 32-bit wide and can be accessed in single access mode.

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x1000	Firmware Version	A24/A32	D16/D32	R
0x1800	Control register	A24/A32	D16/D32	R/W
0x1804	Clock frequency register	A24/A32	D16/D32	R/W

The registers of the Gate and Delay controller are used to configure the GDG parameters.

ADDRESS	REGISTER/CONTENT	ACCESS MODE		Read/Write
0x7F00	Data write register	A24/A32	D16/D32	R/W
0x7F04	Command register	A24/A32	D16/D32	R/W
0x7F08	Control register	A24/A32	D16/D32	R/W
0x7F0C	Data read register	A24/A32	D16/D32	R
0x7F10	Status register	A24/A32	D16/D32	R

13.7.3 Register Description

- **FIRMWARE VERSION register:** Stores the revision number of the firmware.

  Address: 0x1000.

  Mode: Read only.

Bit	Description
[31:0]	Revision Number

- **CONTROL register:**

  Address: 0x1800.

  Mode: Read and Write.

Bit	Description
[8:4]	The DFPGA channel used to delay the input signal
[1]	Input signal selection: 0 = clock signal (frequency set by bits [8:4] 1 = G0 input
[0]	I/O port level selection: 0 = NIM level 1 = TTL level

WARNING: When the G0 connector is used as an input, the user should follow the procedure described in Sec. **12.3**.

- **CLOCK FREQUENCY register:** Sets the value of the frequency divider factor of the clock selectable as input (0x0 = 50 MHz, 0x1= 25 MHz, ….).

  Address: 0x1804.

  Mode: Read and Write.

Bit	Description
[31:0]	Frequency divider factor 0x0 = Clock frequency is 50 MHz 0x1 = Clock frequency is 25 MHz …

- **DATA WRITE register:** The information written in this register is sent to the GDG.

  Address: 0x7F00.

  Mode: Read and Write.

Bit	Description
[31:0]	Data

- **COMMAND register**

  Address: 0x7F04.

  Mode: Read and Write.

Bit	Description
[31:16]	reserved
[15:12]	**Control register[0]=1:** 0x2 = The content of the Data write register is written in the gate internal buffer 0x3 = The content of the Data write register is written in the delay internal buffer **Control register[1]=1:** 0x2 = The gate value of the channel specified by [7:0] is read 0x3 = The delay value of the channel specified by [7:0] is read
[11:8]	0x1 = The values of gate and delay previously buffered using [15:12] are sent to the DFPGA for the channel specified by [7:0] 0x3 = The delays are reset 0x4 = The delays are calibrated 0x5 = The values of gate and delay previously buffered using [15:12] are broadcast to all channels
[7:0]	Delay channel selection (the 3 MSBs should be set to 0)

- **CONTROL register**

  Address: 0x7F08.

  Mode: Read and Write.

Bit	Description
[30:2]	*reserved*
[1]	1 = Start of read operation
[0]	1 = Start of write operation

- **DATA READ register:** The information from the DFPGA can be read in this register.

  Address: 0x7F0C.

  Mode: Read only.

Bit	Description
[31:0]	Data

- **STATUS register**

  Address: 0x7F10.

  Mode: Read only.

Bit	Description
[31:1]	*reserved*
[0]	Gate and delay ready signal (when 0, the Gate and Delay controller is busy. A new command to the Gate and Delay controller should not be issued before this bit sets back to 1)

14 FW2495SC Pay Firmware

CAEN provides the FW2495SC pay firmware for the DT5495 and V2495 boards. By loading this firmware on the USER FPGA of the board, it can be used as a Multievent latching scaler housing up to 160 independent counting channels (reachable thanks to the channel extension given by the x395 mezzanines).

The firmware is downloadable as free trial from CAEN website at the FW2495SC page and will work fully functional for 30 minutes; rebooting the board gives another 30-minute session. The user must then purchase a license to unlock the firmware and use it without time limitaiton.

In order to demonstrate the FW2495SC firmware, the PLUscaler_daq is web available, including the source code as a base for the user development of customized applications (see Sec. **10.4**).

For a detailed description of the firmware and related software, please refer to the FW2495SC User Manual **[RD4]**.

15 Software Development

Software applications can be developed for the DT5495 by using the functions and features provided by the CAEN PLULib library **[RD3]**.

CAEN PLULib library requires the Main FPGA Application Firmware revision 1.4 or higher!

16 Technical Support

To contact CAEN specialists for requests on the software, hardware, and board return and repair, it is necessary a MyCAEN+ account on www.caen.it:

https://www.caen.it/support-services/getting-started-with-mycaen-portal/

All the instructions for use the Support platform are in the document:

A paper copy of the document is delivered with CAEN boards.

The document is downloadable for free in PDF digital format at:

https://www.caen.it/safety-information-product-support

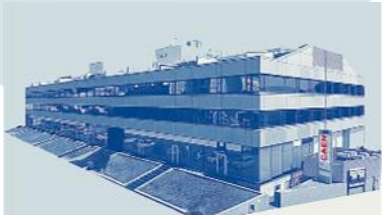
User Manual DT5495
Desktop Programmable Logic Unit

CAEN GmbH
Brunnenweg 9
64331 Weiterstadt
Phone +49 (0)212 254 4077
Mobile +49 (0)151 16 548 484
info@caen-de.com
www.caen-de.com

CAEN Technologies, Inc.
1 Edgewater Street - Suite 101
Staten Island, NY 10305
USA
Phone: +1 (718) 981-0401
Fax: +1 (718) 556-9185
info@caentechnologies.com
www.caentechnologies.com

CAEN S.p.A.
Via Vetraia 11
55049 - Viareggio
Italy
Phone +39 0584 388 398
Fax +39 0584 388 959
info@caen.it
www.caen.it

CAENspa INDIA Private Limited
B205, BLDG42, B Wing,
Azad Nagar Sangam CHS,
Mhada Layout, Azad Nagar, Andheri (W)
Mumbai, Mumbai City,
Maharashtra, India, 400053
info@caen-india.in
www.caen-india.in



UM6508 - DT5495 User Manual rev. 5 - October 9th, 2024 00106/17:DT5495.MUTX/05