TEB2000 CPLD

Table of contents

Overview

CPLD Device with designator U5: LCMX02-1200HC.

Feature Summary

- JTAG
- UART
- I2C
- Power
- Boot Mode
- Reset
- SD
- LED

Firmware Revision and supported PCB Revision

See Document Change History

Product Specification

Port Description

Name / opt. VHD Name	Direction	Pin	Pullup/Down	Bank Power	Description
UART_TXD	in	77	UP	3.3V	UART0 TX / Sends data to MIO13. MIO13 is connected to B2B JB1 Pin 98. (HSS console). In hardware is connected to FT230XQ (U12) chip pin TXD indirectly.
UART_RXD	out	84	UP	3.3V	UART0 RX / Recieves data from MIO12. MIO12 is connected to B2B JB1 Pin 100. (HSS console). In hardware is connected to FT230XQ (U12) chip pin RXD indirectly.

MIO12	in	100	NONE	3.3V	Sends data to UART_RX / In hardware is connected to B2B JB1-pin 100.
MIO13	out	99	NONE	3.3V	Read data from UART_TX / In hardware is connected to B2B JB1-pin 98.
FT_B_RX	out	138	NONE	3.3V	FTDI UART RX (UART1 RX) / In hardware is connected to FTDI chip U4 pin 33. (Linux console)
FT_B_TX / BDBUS0	in	139	UP	3.3V	FTDI UART TX (UART1 TX) / In hardware is connected to FTDI chip U4 pin 32. (Linux console)
MIO14	out	105	NONE	3.3V	Receives data from FT_B_TX of FTDI chip / In hardware is connected to B2B JB1-pin 91.
MIO15	in	95	NONE	3.3V	Sends data to FT_B_RX of FTDI chip / In hardware is connected to B2B JB1-pin 86.
CM0	in	76	UP	3.3V	DIP switch S2-2 / used as JTAG Selection/ If CM0 set to high (S2-2 OFF) Access to CPLD of module otherwise access to FPGA of module.
CM1	in	75	UP	3.3V	DIP switch S2-1 / Used to change PGOOD pin state /If CM1 set to high (S2-1 OFF) PGOOD = '1' otherwise '0'
EN1	out	81	UP	3.3V	B2B Power Enable / In firmware EN1 is connected to 3.3 V permanently.
FLED_1	inout	28	NONE	3.3V	LED (D3-red) / Shows the status of PGOOD or shows the FT_B_TX (UART1 RXD).

FLED_2	inout	27	NONE	3.3V	LED (D4-green) / Shows the status of PGOOD or shows the FT_B_RX (UART1 TXD).
ULED1	out	117	NONE	3.3V	LED (D1-red) / Shows the status of CM0 or shows the UART_RXD (UART0 RX).
ULED2	out	115	NONE	3.3V	LED (D2-green) / Shows the status of CM0 or shows the UART_TXD (UART0 TX).
PHY_LED1	out	86	DOWN	3.3V	Shows the status of NOSEQ and MIO0 signals.
PHY_LED1R	out	92	NONE	3.3V	Shows the status of NOSEQ and MIO0 signals.
PHY_LED2	out	85	NONE	3.3V	Shows the status of NOSEQ and MIO0 signals.
PHY_LED2R	out	91	NONE	3.3V	Shows the status of NOSEQ and MIO0 signals.
JTAGEN	---	120	---	3.3V	Enable JTAG access to CPLD for Firmware update (zero: normal IOs, one: CPLD JTAG access). Selectable over S2-3
M_TCK	in	131	NONE	3.3V	JTAG from/to FTDI
M_TDI	in	136	NONE	3.3V	JTAG from/to FTDI
M_TDO	out	137	NONE	3.3V	JTAG from/to FTDI
M_TMS	in	130	NONE	3.3V	JTAG from/to FTDI
TCK_B	out	1	NONE	3.3V	JTAG from/to Module
TDI_B	out	3	NONE	3.3V	JTAG from/to Module
TDO_B / C_TDO	in	2	UP	3.3V	JTAG from/to Module
TMS_B	out	4	NONE	3.3V	JTAG from/to Module

MIO0	in	94	UP	3.3V	This pin is connected to DIP swith S2-4 and B2B JB1 Pin 88. This signal is forwarded to MODE signal in firmware.
MIO9	out	96	NONE	3.3V	SD_CD signal is directed to this signal in firmware, if PGOOD = '1'. / In hardware is connected to B2B JB1 pin 92.
MODE	out	83	NONE	3.3V	Dip switch S2-4 (MIO0) is connected to MODE pin. This pin in hardware is connected to B2B JB1 pin 31.
NOSEQ	inout	78	UP	3.3V	NOSEQ can be set or reset by i2c interface in linux console, if an i2c interface is prepared already in linux.
PGOOD	inout	82	UP	3.3V	PGOOD can be set or reset via CM1 (dip switch S2-1). In hardware is connected to B2B JB1 pin 29.
PROGMODE	out	104	UP	3.3V	Enable B2B Module JTAG access to CPLD for Firmware update
RESIN	out	119	NONE	3.3V	Module Reset pin on B2B JB2 pin 17.
S1	in	114	UP	3.3V	Push Button / Used as module Reset. In hardware is connected to S1 reset pushbutton.
SDA / MIO11	inout	97	UP	3.3V	I2C Data
SCL /MIO10	in	98	UP	3.3V	I2C Clock
X0	inout	39	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.

X1	inout	38	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X2	inout	40	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X3	inout	41	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X4	inout	42	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X5	inout	43	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X6	inout	44	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X7	inout	45	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X8	inout	47	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X9	inout	48	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.

X10	inout	49	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X11	inout	50	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X12	inout	52	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X13	inout	54	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X14	inout	55	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X15	inout	56	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X16	inout	59	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
X17	inout	60	NONE	3.3V	This port can be read or written via GPIO_input register and GPIO_output register respectively.
BCBUS0	in	122	NONE	3.3V	currently_not_used
BCBUS1	in	121	NONE	3.3V	currently_not_used
BDBUS2	in	133	NONE	3.3V	currently_not_used
BDBUS3	in	132	NONE	3.3V	currently_not_used

BDBUS4	in	128	NONE	3.3V	currently_not_used
BDBUS5	in	127	NONE	3.3V	currently_not_used
BDBUS6	in	126	NONE	3.3V	currently_not_used
BDBUS7	in	125	NONE	3.3V	currently_not_used
ADBUS4	in	143	NONE	3.3V	currently_not_used
ADBUS7	in	142	NONE	3.3V	currently_not_used
ACBUS4	in	141	NONE	3.3V	currently_not_used
ACBUS5	in	140	NONE	3.3V	currently_not_used
USB_OC	in	73	NONE	3.3V	This port can be read via GPIO_input register.
E_SD_CMD	in	110	NONE	3.3V	currently_not_used
E_SD_SCLK	in	109	NONE	3.3V	currently_not_used
E_SD_DAT0	in	106	NONE	3.3V	currently_not_used
E_SD_DAT1	in	107	NONE	3.3V	currently_not_used
E_SD_DAT2	in	112	NONE	3.3V	currently_not_used
E_SD_DAT3	in	111	NONE	3.3V	currently_not_used
SD_SEL	out	113	NONE	3.3V	Set to GND / currently_not_used
SD_CD	in	93	UP	3.3V	This port is forwarded to PHY_LED2. If SD card is plugged , PHY_LED2 is on. This port forwarded to MIO9 too, if PGOOD = '1'.
DUMMY	out	74	NONE	3.3V	No Connect

Functional Description

Dip Switch

DIP Switch S2				
S2-1	**S2-2**	**S2-3**	**S2-4**	**Description**

CM1	CM0	JTAGEN	MIO0	JTAGEN set carrier board CPLD into the chain for firmware update.

JTAG

JTAG signals routed directly through the CPLD to module in B2B connector.

TEB2000 CPLD can be selected with JTAGEN (DIP switch S2-3). Access between CPLD of the TEB2000 board and the plugged module same as TEM0007 can be multiplexed via JTAGEN (S2-3). Logical one is for accessing to CPLD of TEB2000 (OFF) and logical zero is for the module (ON).

Access to FPGA of the plugged module or its CPLD can be switched with PROGMODE which is driven by CM0 (DIP switch S2-2). CM1 and CM0 are pulled up in CPLD internally.

S2-2	S2-3	CM0 (PROGMODE) (S2-2)	JTAGEN (S2-3)	Description
OFF	OFF	1	1	Access to TEB2000 CPLD
OFF	ON	1	0	Access to CPLD of B2B Module
ON	OFF	0	1	Access to TEB2000 CPLD
ON	ON	0	0	Access to FPGA of B2B Module

EN1

EN1 is set to one in firmware permanently.

NOSEQ

NOSEQ pulled up to 3.3V. NOSEQ can be set or reset by i2c interface in linux console, if an i2c interface is already prepared for the i2c interface in linux.

NOSEQ	Connected to	Related command in linux console	Description
'0'	GPIO_output[16]	`i2cset -y <related bus> 0x20 0x02 0x00`	`It is depends on the linux design. For example i2cset -y 0 0x20 0x02 0x00`
'1'	GPIO_output[16]	`i2cset -y <related bus> 0x20 0x02 0x01`	`It is depends on the linux design. For example i2cset -y 0 0x20 0x02 0x01`

To read the NOSEQ status, GPIO_input[16] must be read like the following instruction:

```
i2cget -y <related bus> 0x20 0x02

For example --> i2cget -y 0 0x20 0x02

Note that the bus number depends on the linux design and can be varied for
different designs.
```

PGOOD

PGOOD pulled up to 3.3V in CPLD internally. PGOOD pin can be set or reset by user. If CM1 set to high (S2-1 OFF) , PGOOD will be set to high otherwise PGOOD is set to low.

PGOOD	Condition	Description
'0'	CM1 = '0'	Dip switch S2-1 ON
'1'	CM1 = '1'	Dip switch S2-1 OFF

MODE

This pin can be controlled with dip switch S2-4 (MIO0).

MODE	Condition	Description
'0'	MIO0 = '0'	Dip switch S2-4 ON
'1'	MIO0 = '1'	Dip switch S2-4 OFF

Reset

There is two reset switchs on the board (S1 and S6). The S6 is a soft reset button (SRST) that is not directed to CPLD chip. The S1 signal is connected to CPLD chip and is used in firmware code to create a reset signal. RESIN signal is the output reset signal of the CPLD and is driven by S1 push button. The S1 push button is debounced.

Signal	Designator	Connected to	Active Level	Description
SRST	S6	B2B JB2 pin 56	Active low	This signal is not used in CPLD firmware.
S1	S1	CPLD chip pin 114	Active low	This signal is used in CPLD firmware.

Pin	CPLD Pin	Connected to	Description
RESIN	119	B2B JB2 Pin 17	Active-low

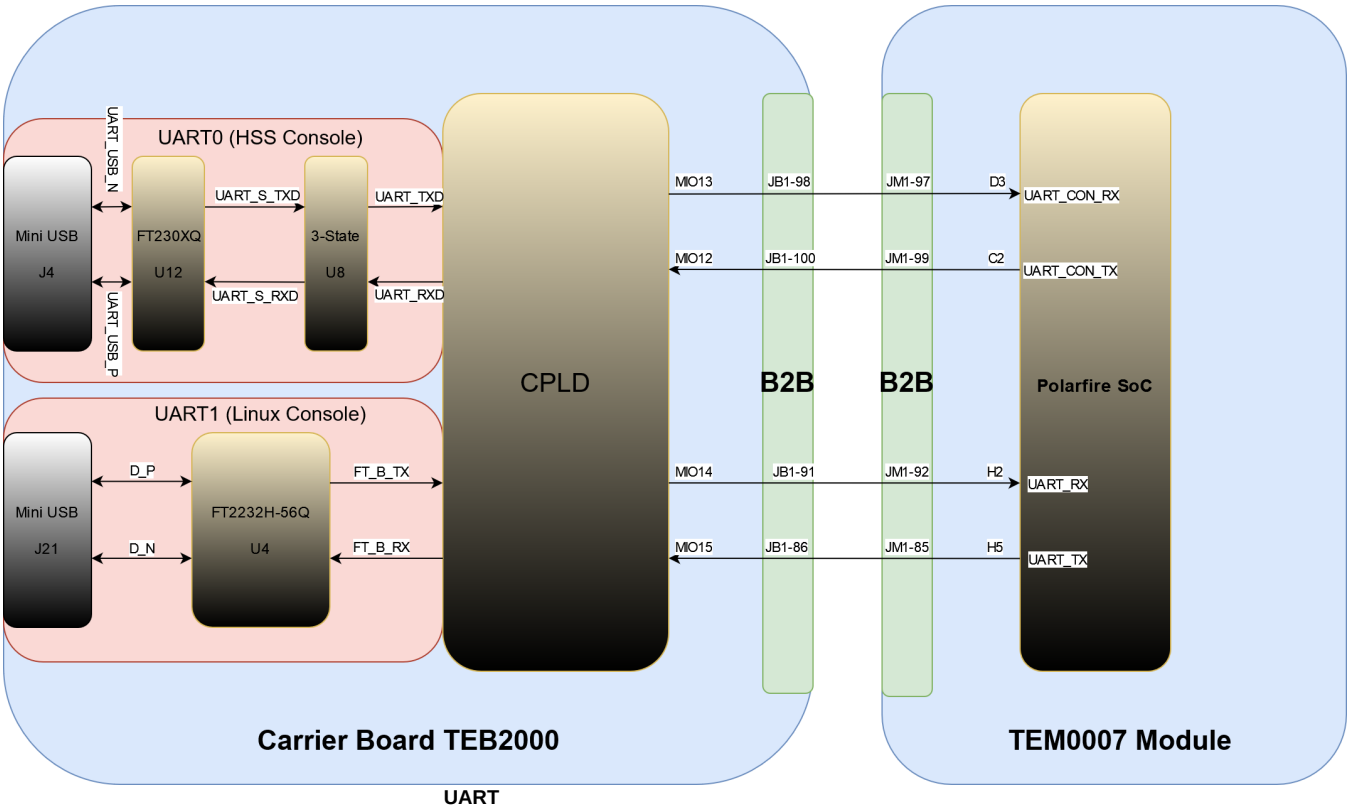
Boot Mode

For plugged Polarfire SoC module same as TEM0007 the boot mode can not be selected via MODE and PGOOD pins, because polarfire SoC boot mode can be selected only via HSS design for TEM0007.

UART

MIO14 is driven by BDBUS0 (FTDI RX). BDBUS1 (FTDI TX) is driven by MIO15 . MIO13 is driven by UART_TXD. UART_RXD is driven by MIO12.

UART 0 (HSS Console)						
CPLD UART Input Pin	**CPLD Pin**	**Connected to**	**CPLD UART Output Pin**	**CPLD Pin**	**Connected to**	**Description**
MIO12	100	B2B-JB1-100	**UART_RXD**	84	U8-Pin 11	
UART_TXD	77	U8-Pin 13	**MIO13**	99	B2B-JB1-98	
UART 1 (Linux Console)						
CPLD UART Input Pin	**CPLD Pin**	**Connected to**	**CPLD UART Output Pin**	**CPLD Pin**	**Connected to**	**Description**
FT_B_TX	139	FTDI Chip U4 Pin 32	**MIO14**	105	B2B-JB1-91	
MIO15	95	B2B-JB1-86	**FT_B_RX**	138	FTDI Chip U4 Pin 33	



UART

SD

SD selection (SD_SEL) is set to GND (SD Card slot). MIO9 is switched to SD_CD (Card Detect) and its status depends on SD_CD .

On-board LEDs

LED	Designator	LED Status	Condition	Description
ULED1 (Red)	D1	Fast blink red	Access to CPLD of module (CM0 = '1') S2-2 = OFF	
		Connected to UART1_RX	Otherwise	
ULED2 (Green)	D2	Fast blink green	Access to CPLD of module (CM0 = '1') S2-2 = OFF	
		Connected to UART1_TX	Otherwise	
FLED_1 (Red)	D3	Fast blink red	PGOOD = '1' (CM1 = '0') S2-1 = ON	
		Connected to UART0_RX	Otherwise	
FLED_2 (Green)	D4	Fast blink green	PGOOD = '1' (CM1 = '0') S2-1 = ON	
		Connected to UART0_TX	Otherwise	

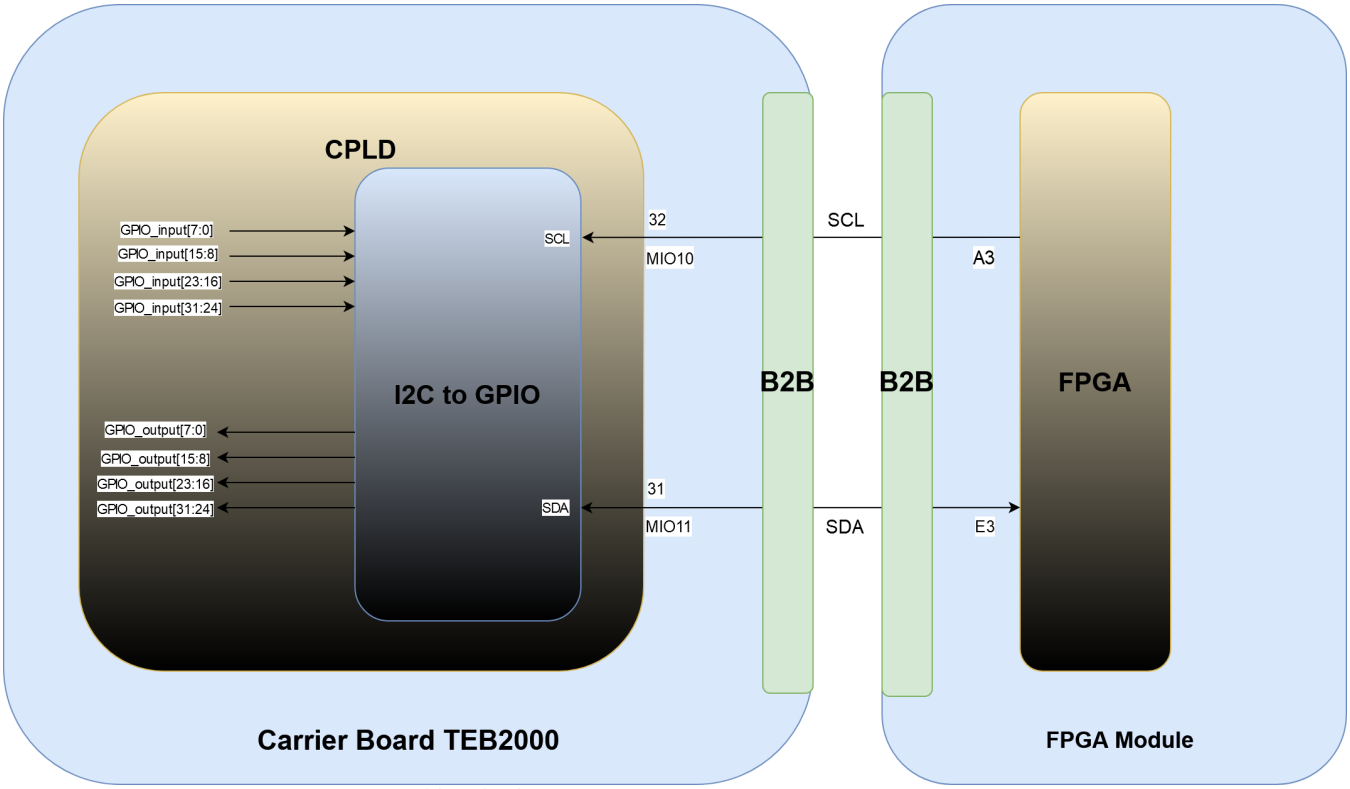
LED	Designator	LED Status	Condition	Description
PHY_LED1 (Green LED Anode, Yellow LED Cathode) / **PHY_LED1R (Green LED Cathode, Yellow LED Anode)**	J14B	Fast blink yellow	NOSEQ = '0' and MIO0 = '0'	`In Linux console enter: i2cset -y 0 0x20 0x02 0x00 and S2-4 = ON*`
		Slow blink yellow	NOSEQ = '0' and MIO0 = '1'	`In Linux console enter: i2cset -y 0 0x20 0x02 0x00 and S2-4 = OFF*`
		Fast blink yellow green	NOSEQ = '1' and MIO0 = '0'	`In Linux console enter: i2cset -y 0 0x20 0x02 0x01 and S2-4 = ON*`
		Slow blink yellow green	NOSEQ = '1' and MIO0 = '1'	`In Linux console enter: i2cset -y 0 0x20 0x02 0x01 and S2-4 = OFF*`
PHY_LED2 (Green LED Cathode, Yellow LED Anode) / **PHY_LED2R (Green LED Anode, Yellow LED Cathode)**	J14C	ON	SD card plugged (SD_CD = '0')	

		OFF	Otherwise	

*Note: The related bus number of I2C can be varied and it depends on the Linux design. For example the bus number is 0 here.

I2C to GPIO

I2C to GPIO is a subsystem in firmware of CPLD that provides an i2c interface that writes received data to GPIO_output 8 bit registers or reads 8 bit GPIO_input registers and send read data to i2c bus.



I2C to GPIO

I2C bus is connected to MIO10 (SCL signal) and MIO11 (SDA signal). MIO10 to MIO15 are direct connection between CPLD of TEB2000 and FPGA on the module (for example TEM0007) through B2B connector. If in FPGA design exists no i2c interface for MIO10 and MIO11, this block will be unused. More information about MIO10 to MIO15 are shown in the following table for TEM0007 modules and TEB2000 carrier board:

B2B Pin	B2B JB1-96	B2B JB1-94	B2B JB1-100	B2B JB1-98	B2B JB1-91	B2B JB1-86	
Carrier board	Label / Firmware function	Label / Firmware function	Label / Firmware function	Label / Firmware function	Label / Firmware function	Label / Firmware function	Description

TEB2000	MIO10 / I2C-SCL	MIO11 / I2C-SDA	MIO12 / UART0_RX	MIO13 / UART0_TX	MIO14 / UART1-TX	MIO15 / UART1-RX	MIO10 and MIO11 are used in CPLD firmware as I2C SCL and SDA respecti vely.
B2B Pin	**B2B JM1-95**	**B2B JM1-93**	**B2B JM1-99**	**B2B JM1-97**	**B2B JM1-92**	**B2B JM1-85**	
Module	**Label / Chip pin**	**Label / Chip pin**	**Label / Chip pin**	**Label / Chip pin**	**Label / Chip pin**	**Label / Chip pin**	**Description**
TEM0007	I2C_CON_SCL / A3	I2C_CON_SDA / E3	UART_CON_TX / C2	UART_CON_RX / D3	UART_RX / H2	UART_TX / H5	MIO10 and MIO11 are already set in test_design of TEM0007 as SCL and SDA respectively.

I2C to GPIO registers access methods

I2C to GPIO subsystem has 4 output and 4 input 8 bit registers. These registers can be written or read in linux as shown in the following tables:

GPIO input register	Address	Read command in Linux*	Description
GPIO_input[7:0]	0x00	`i2cget -y 0 0x20 0x00`	0x20 is device address. (I2C to GPIO address).
GPIO_input[15:8]	0x01	`i2cget -y 0 0x20 0x01`	0x20 is device address. (I2C to GPIO address).
GPIO_input[23:16]	0x02	`i2cget -y 0 0x20 0x02`	0x20 is device address. (I2C to GPIO address).
GPIO_input[31:24]	0x03	`i2cget -y 0 0x20 0x03`	0x20 is device address. (I2C to GPIO address).

GPIO output register	Address	Write command in Linux*	Description
GPIO_output[7:0]	0x00	`i2cset -y 0 0x20 0x00 <8bit data>`	0x20 is device address. (I2C to GPIO address). 0x00 is register address.
GPIO_output[15:8]	0x01	`i2cset -y 0 0x20 0x01 <8bit data>`	0x20 is device address. (I2C to GPIO address). 0x01 is register address.
GPIO_output[23:16]	0x02	`i2cset -y 0 0x20 0x02 <8bit data>`	0x20 is device address. (I2C to GPIO address). 0x02 is register address.
GPIO_output[31:24]	0x03	`i2cset -y 0 0x20 0x03 <8bit data>`	0x20 is device address. (I2C to GPIO address). 0x03 is register address.

*The related bus number can be varied with Linux design. For example the bus number is 0 here.

Note that resetting due to SRST push button or rebooting via reboot command in linux console will not delete the saved changed status of GPIO_output registers via i2c interface. However, resetting with the S1 reset button deletes all saved data in GPIO_output registers and sets these registers to the default values.

I2C to GPIO registers

GPIO input registers

The following port or signals can be read via GPIO_input[ ] registers.

GPIO_input[7:0] (address 0x00)		
GPIO input bit	**Port/Signal**	**Description**
0	CPLD_REVISION [0]*	
1	CPLD_REVISION [1]*	
2	CPLD_REVISION [2]*	
3	CPLD_REVISION [3]*	
4	CPLD_REVISION [4]*	
5	CPLD_REVISION [5]*	
6	CPLD_REVISION [6]*	
7	CPLD_REVISION [7]*	

*Note: CPLD_REVISION[7:0] = Firmware version

GPIO_input[15:8] (address 0x01)		
GPIO input bit	**Port/Signal**	**Description**
8	X0	
9	X1	
10	X2	
11	X3	
12	X4	
13	X5	
14	X6	
15	X7	

GPIO_input[23:16] (address 0x02)		
GPIO input bit	**Port/Signal**	**Description**

16	NOSEQ	
17	X8	
18	X9	
19	X10	
20	X11	
21	X12	
22	X13	
23	X14	

GPIO_input[31:24] (address 0x03)		
GPIO input bit	**Port/Signal**	**Description**
24	X15	
25	X16	
26	X17	
27	CM0	
28	CM1	
29	PGOOD	
30	MIO0 (MODE)	
31	USB_OC	

GPIO output registers

The following port or signals can be written via GPIO_output[ ] registers.

GPIO_output[7:0] (address 0x00)		
GPIO output bit	**Port/Signal**	**Description**
0	X0	
1	X1	
2	X2	
3	X3	
4	X4	
5	X5	

6	X6	
7	X7	

GPIO_output[15:8] (address 0x01)		
GPIO output bit	**Port/Signal**	**Description**
8	X8	
9	X9	
10	X10	
11	X11	
12	X12	
13	X13	
14	X14	
15	X15	

GPIO_output[23:16] (address 0x02)		
GPIO output bit	**Port/Signal**	**Description**
16	NOSEQ	
17	X16	
18	X17	
19	Reserved	
20	Reserved	
21	Reserved	
22	Reserved	
23	Reserved	

GPIO_output[31:24] (address 0x03)		
GPIO output bit	**Port/Signal**	**Description**
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	

28	Reserved	
29	Reserved	
30	Reserved	
31	Reserved	

Appx. A: Change History and Legal Notices

Revision Changes

- Changes REV01 to REV02
  - Bug fix: The process of GPIO_output read/write needed a reset signal to avoid i2c error by i2c reading or i2c writing first time after switching on.
  - The changed data in GPIO_output register will be reset to default value by resetting via S1 bush button. But rebooting via reboot command in linux or resetting via SRST button switch will not reset the GPIO_output register.
  - Unnecessary signal same as CR5 and CR4 has been deleted.
  - Internal oscillator frequency of CPLD is increased from 12.09MHz to 24.18MHz.
  - GPIO_input[30] is allocated to MIO0 (MODE) instead of S1.
  - GPIO_output register mapping changed.
  - FLED_1 and FLED_2 function changed.
- REV01
  - I2C port added.
  - I2C to GPIO component added.
  - NOSEQ can be changed via I2C port.
  - UART1 port added. This board has two UART ports. (UART0 and UART1)
  - New construction for UART0 and UART1 serial interfaces
  - Generic parameter CPLD_REVISION added.
  - JTAG timing correction
  - LED states and related ports/signals are changed. --> SD_CD, NOSEQ, MIO0, PGOOD, CM0 and CM1
  - New mapping for related ports/signals to GPIO_input and GPIO_output registers
  - New Ports are defined according to the schematic or revision 1

Document Change History

To get content of older revision got to "Change History" of this page and select older document revision number.

Date	Document Revision	CPLD Firmware Revision	Supported PCB Revision	Authors	Description

Error re Ambigu	**Error rendering macro 'page-info'** Ambigu	REV02	REV01	**Error rendering macro 'page-info'** Ambigu	• REV02 release • Bugfix : Correction on I2c read /write for first time after switching on • The frequency of internal oscillator of CPLD chip increased from 12.09 MHz to 24.18 MHz. • GPIO_output register mapping changed. • FLED_1 and FLED_2 function changed. • GPIO_input [30] allocation changed. • Firmware release (SC-PGM-TEB2000-001_CARRIER-00x02_20240429.zip)
2024-04-20	v.30	REV02	REV01	Mohsen Chamanbaz	• REV01 release • Firmware release (SC-PGM-TEB2000-001_CARRIER-00x01_20240418.zip)

Appx. B: Legal Notices

Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]