cosylab

M-Box with PDC Motion Control

User manual

Revision:	4.11
Status:	Released
Repository:	/svn/acc/microIOC/microIOC-M-Box/
Project:	ACC-FAIR-uIOC-and-M-Box
Folder:	trunk/user_guides
Document ID:	CSL-DOC-13-83591
File:	DOC-microIOC-MBOX-PDC-UserManual.doc
Owner:	mlevicnik
Last modification:	May 28, 2020
Created:	February 25, 2013

Document History

Revision	Date	Changed/ reviewed	Section(s)	Modification
1.0	2007-7-12	jdedic/ gpajor	All	Created.
1.1	2007-7-16	jdedic	2.1.1 various 4	PLIM/MLIM and AXINT usage. modified Table 2, 5, 7, 12, and Figure 12 created/added
2.0	2007-10-25	jdedic	All	See below text for explanation
2.0.1	2007-11-02	jdedic	2.2.2 3.1	inserted minor changes
2.1	2007-11-08	jdedic	2.1.2 Table 17 all	modified **DevMov** register (0x0b) not used anymore inserted renamed: mEB → PDC
3.0	2011-01-27	mlevicnik	All	modified
3.1	2011-01-28	gjansa	All 3	Review Updated
3.2	2012-06-15	mlevicnik	2.1.2	**DevMov** description added
4.0	2013-03-15	mlevicnik	All	updated
4.1	2013-04-03	mlevicnik	All	Updated/changed information of PDC analog input module
4.2	2013-07-16	gjansa	3	Updated
4.3	2013-07-18	gjansa	3	Updated figure
4.4	2013-08-30	mlevicnik	2.1.8 4	XPORT description moved to microIOC User's manual Specifications with operating conditions added
4.5	2014-03-11	mlevicnik	2.1.2, 2.1.3, 2.1.4, 2.2.4, 2.2.6	Updated
4.6	2014-03-31	gjansa	3	Updated shared memory monitor description
4.7	2015-01-12	mlevicnik	2.4	Info about cable length added
4.8	2015-02-18	mlevicnik	2.2.10	Info about current adjustment on PDC
4.9	2016-05-23	mlevicnik	All	Minor updates (text and figures)
4.10	2020-05-22	mmarn	2.2.7, 3, 3.3	Updated SSI communication interface in 2.2.7. There is now only one FESA class (3). New LED behaviour (3.3).
4.11	2020-05-28	mlevicnik	2.1.7, 2.1.9	References updated. CPLD flashing procedure added. Insertion description of PMC FTRN added.

Confidentiality

This document is classified as a **`public document`**. As such, it or parts thereof are openly accessible to anyone listed in the Audience section, either in electronic or in any other form.

Scope

This document covers the system specifications, requirements and implementation of a motion control system for GSI.

Audience

All Cosylab and GSI employees.

Typography

This document uses the following styles:

A box like this contains important information.

Warning!
A box like this provides information, which should not be disregarded!

Glossary of Terms

API.....Application Program Interface
CPLD .....Complex Programmable Logic Device
GUI.....Graphical User Interface
HW.....Hardware
NTC.....Negative Temperature Coefficient resistor
PDC.....Power Drive Case
PMAC.....Programmable Multi-Axis Controller
SSI .....Synchronous Serial Interface
SW.....Software

References

[1] GSI Motion Control - System specifications, revision 0.7

[2] www.microioc.com/download/microIOC-PD-baseline.pdf

[3] DeltaTau PMAC2A PC/104:
www.deltatau.com/Common/products/pc104.asp?node=id110&connectionstr=release

[4] Xilinx, CoolRunner XPLA3 series CPLD:
www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/coolrunner_xpla3_cplds/index.htm

[5] Mean Well, PSP-600-24 (24V, 25A, 600W):
https://www.meanwell-web.com/en/product-info/ac-dc-power-supply/enclosed-type/401-600-w/psp-600/product/PSP-600-24

[6] 8-channel Analog Input Module to RS-485, 16-bit differential:
http://www.advantech.com/products/ADAM-4117/mod_9161079D-563C-428C-8DF4-AA29FFFA995D.aspx

[7] XOG008-01, Xtreme/104-Plus Opto Serial Communications Card
http://www.connecttech.com/sub/Products/PC104PlusProducts_Xtreme104PlusOpto.asp?l1=pc104plus&l2=xt104plusopto

[8] Lantronix Device Installer:

http://www.lantronix.com/device-networking/utilities-tools/device-installer.html

[9] Lantronix XPORT Pro:
http://www.lantronix.com/support/downloads/?p=XPORTPRO

[10] PDC schematics and wiring documentation

[11] G. Jansa, Stepper Motor Control – System Design, 1.4

[12] G. Gaspersic, M.Marn, Stepper Motor Control – Local control, 1.11

[13] V. Juvan et al., Stepper Motor Control – LCD, 1.2

[14] G. Jansa, Stepper Motor Control – Installation and Configuration, 1.3

[15] M. Levicnik, microIOC - Hardware installation and user's manual, 1.9

Table of Contents

1. System overview **7**
1.1. microIOC M-Box 7
1.2. PDC 7

2. Signal descriptions **9**
2.1. microIOC-M-Box-PMAC 9
2.1.1. Axes inputs/outputs 10
2.1.2. Device moving outputs 10
2.1.3. Pulse outputs 11
2.1.4. Interlock outputs 12
2.1.5. RS-485 serial connectors 12
2.1.6. PMAC serial connector 13
2.1.7. JTAG connector 13
2.1.8. XPORT Pro connection 14
2.1.9. Insertion of PMC FTRN card 15
2.2. PDC 18
2.2.1. Axes inputs/outputs 19
2.2.2. RS-485 serial connectors 19
2.2.3. Axis-interlock inputs 19
2.2.4. Out-position signal outputs 20
2.2.5. Potentiometer and limit switches connector 20
2.2.6. Stepper motor connector 22
2.2.7. SSI module and encoder connectors 22
2.2.8. Serial module and analog input signals 23
2.2.9. Middle switch configuration 23
2.2.10. Current Cutback Release connector 24
2.3. PDC wiring scheme 26
2.4. Cabling between microIOC M-Box-PMAC and PDC 26
2.4.1. Axis signal connection cable 26
2.4.2. PDC communication cable 26

3. Software **28**
3.1. Roles of software components 29
3.2. PMAC software 30
3.2.1. PMAC-CPLD memory access 30
3.3. PMAC interface board and CPLD configuration 32

4. Specifications **35**
4.1. Electrical 35
4.2. Operating Environment 35
4.3. Storage Environment 35

Figures

Figure 1: Control system description 7
Figure 2: microIOC M-Box block diagram 8
Figure 3: PDC block diagram 8

Figure 4: Back panel (right) connections of microIOC M-Box 9
Figure 5: Back panel (left) connections of microIOC M-Box 9
Figure 6: axis connector (DB-25F) 10
Figure 7: Device Moving output board 11
Figure 8: Interlock output board 12
Figure 9: Xilinx programming tool connected to cable adapter 14
Figure 10: Removing screws for removing M-Box top cover 15
Figure 11: Removal of the top cover and serial PCI-104 card 16
Figure 12: Insertion of PMC FTRN 16
Figure 13: Alignment of PMC connectors 17
Figure 14: Fixation of PMC FTRN using two M2.5 x 6 mm screws 17
Figure 15: Mounting and fixating PCI-104 serial card 18
Figure 16: PDC front panel 18
Figure 17: PDC back panel 19
Figure 18: potentiometer and limit switch connector (DB9F) 20
Figure 19: PDC middle switch configuration switches 24
Figure 20: Principle wiring scheme within PDC 26
Figure 21: system overview 29
Figure 22: PMAC interface board overview 32

Tables

Table 1: Device moving outputs pinout (applies to DevMov 1 – DevMov 8) 11
Table 2: Relay specifications 11
Table 3: TTL pulse outputs pinout (applies to Pulse 1 – Pulse 8) 12
Table 4: Interlock outputs pinout (applies to OutEnd 1 – OutEnd 8) 12
Table 5: RS-485 serial connector pinout, DB9M, (applies to RS485/1 to RS485/4) 13
Table 6: RS232 PMAC connector pinout, DB9F 13
Table 7: CPLD JTAG connector pinout, DB9F 14
Table 8: serial connections on PDC unit 19
Table 9: axis interlock input connector, LEMO socket type EPL.0S.302.HLN 19
Table 10: out-position signal output connector, LEMO socket, EPL.0S.302.HLN 20
Table 11: Potentiometer/feedback connector pinout (DB9F) 20
Table 12: stepper motor connector (UTG01412S, SOURIAU) and connection to stepper driver 22
Table 13: SSI encoder connector pinout (DB9F) 23
Table 14: serial module input output signals 23
Table 15: PDC middle switch jumper settings 24
Table 16: Connection/pinout of current cutback release connector 25
Table 17: CPLD registers' descriptions 33
Table 18: CPLD LED registers 33
Table 19: LED states 33

1. System overview

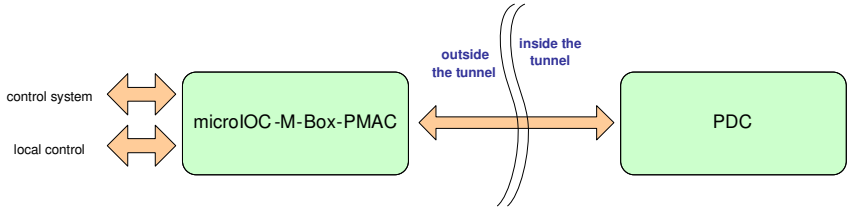

Figure 1: Control system description

This document describes the system for controlling up to 8 stepper motors. System requirements specifications are provided in [1]. Figure 1 gives a system overview of the system. The main components of the system are the microIOC M-Box and the PDC (power drive case).

1.1. microIOC M-Box

The microIOC M-Box shown in Figure 2 is a microIOC-based product, offering an advanced motion control functionality and control of general-purpose signals. microIOC is a very flexible software (SW) platform, which provides control of all modules and provides a hosting environment for the control-system application (product benefits and features are described in [2]). Motion control is implemented by a PMAC (programmable multi-axis controller) [3] that handles the motion control of up to eight axes. Additionally, a PMAC interface board was developed that provides voltage translation and optical isolation of signals (addressing large distances and possible noisy environments). This interface board is highly programmable (using a Xilinx CPLD [4]) by the means of mapping signals that pass through the board (signals can be inverted, logically combined, etc). The PMAC interface board also extends the available PMAC's per-axis input/output signals to provide control of additional functionality (such as brake control, interlocks, etc.). To provide communication with remote analog module (for position feedback) an RS-485 communication card with optically isolated outputs (addressing large distances and possible noisy environments) was added.

1.2. PDC

The PDC (Figure 3) contains the power supply [5] for 4 stepper drives (Oriental motor CRD5128PB), an RS-485 analogue input module [6], and a simple PDC interface circuit (signal adaptation).

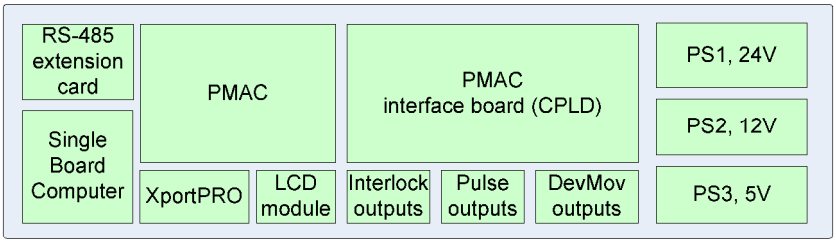

Figure 2: microIOC M-Box block diagram

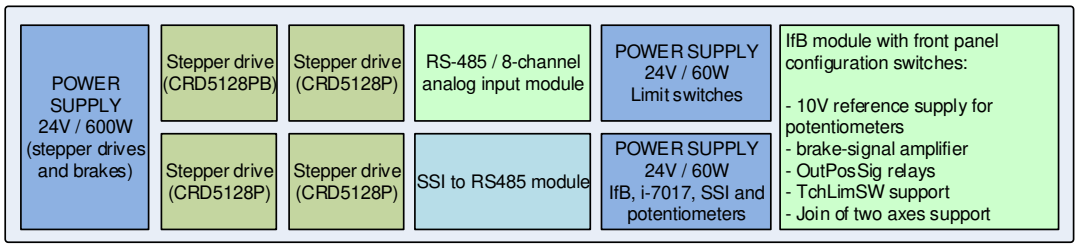

Figure 3: PDC block diagram

According to the specification requirements, the system has to be fully functional even with distances of 300 m between microIOC M-Box and PDC (see Figure 1). To achieve this, all the analog signals (i.e. potentiometer position feedback) and power signals (to stepper motors) are handled inside the PDC, closer to the controlled motors. Two types of communication exists between the two boxes; RS-485 (optically isolated at the microIOC M-Box side) and communication with the PMAC interface board (optically isolated, 24V signals, 10-20mA). This scheme provides a stable and noise-immune communication over the 300 m distance.

2. Signal descriptions

2.1. microIOC-M-Box-PMAC

The back panel connections for the microIOC M-Box are shown in Figure 4 and Figure 5.

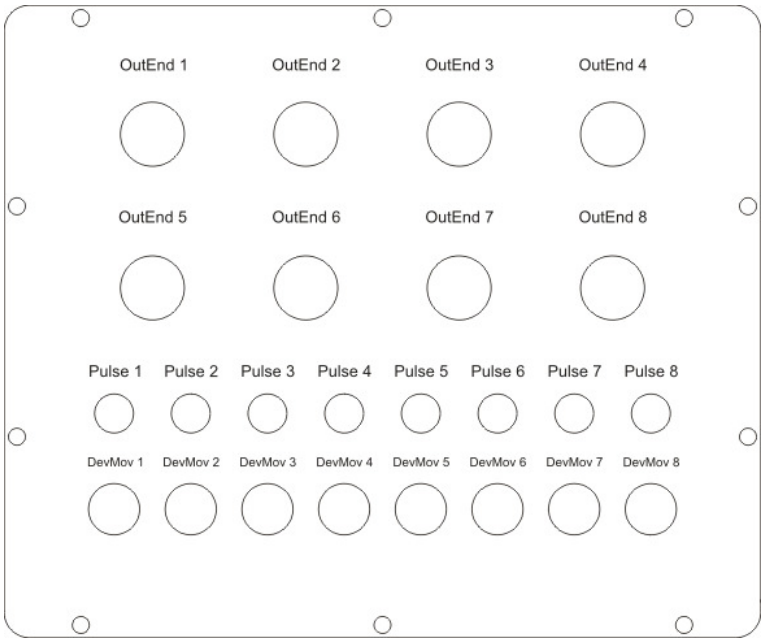

Figure 4: Back panel (right) connections of microIOC M-Box

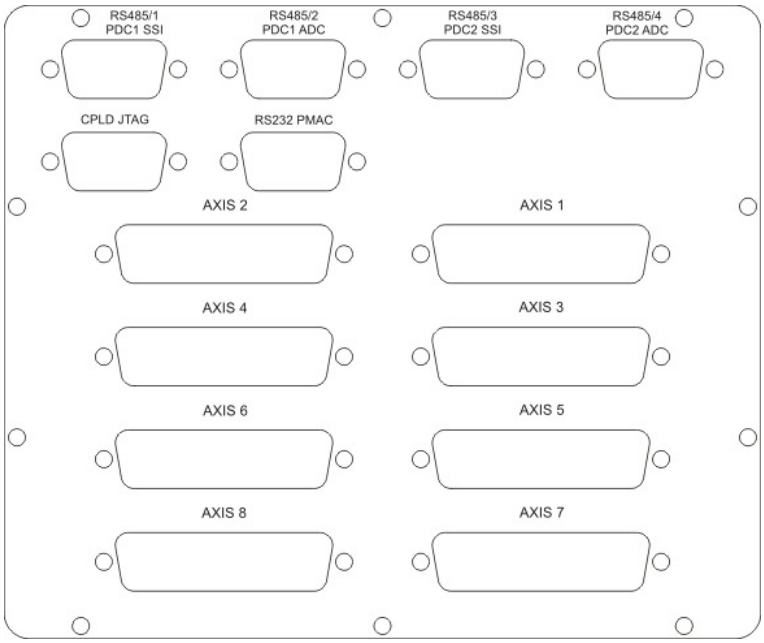

Figure 5: Back panel (left) connections of microIOC M-Box

2.1.1. Axes inputs/outputs

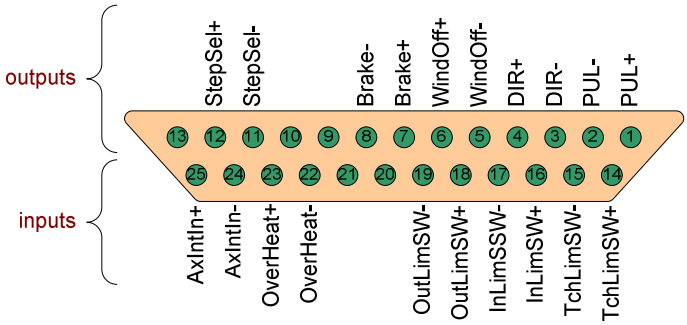

Figure 6: axis connector (DB-25F)

Each controlled axis has a dedicated DB-25 connector, so there are 8 connectors in total. Only low-power (24V, ~15mA) signals are transmitted to the PDC. The following signals are available per connector:

- **`PUL`**, **`DIR`**: signals to control the stepper drive. These are PMAC's signals and therefore handled by its motion control programs. If required, **`PUL`** signal can be conditionally gated by CPLD (e.g. to provide an additional safety level). Outputs of these signals are prepared for direct connection to optocoupler inputs of the stepper drive.
- **`WindOff`**, **`StepSel`**: signals to control the stepper drive. **`WindOff`** is handled automatically by PMAC's motion control programs. **`StepSel`** is controlled through a user accessible register; see Table 17 for a description of the CPLD registers. Outputs of these signals are prepared for direct connection to optocoupler inputs of the stepper drive.
- **`Brake`**: signal to control the brake amplifier circuitry within the PDC interface board. The signal is handled automatically by PMAC's motion control programs and is not intended to be controlled by higher level SW.
- **`TchLimSW`**, **`InLimSW`**, **`OutLimSW`**: input signals for detecting limit positions. PMAC's motion control programs provide responses to them. The CPLD also reads these inputs and can provide any actions regarding the states of these inputs. Signal **`InLimSW`** is used to detect the movement in the direction of the beam. Signal **`OutLimSW`** is used to detect the movement out of the beam. **`InLimSW`** and **`OutLimSW`** switches must be of the type that is normally-closed and floating. See Section 2.2.5 for connection details.
- **`OverHeat`**: input signal from stepper drive for detecting overheat condition. This signal is accessible through CPLD register, Table 17: CPLD registers' descriptions.
- **`AxIntIn`**: input signal from PDC (LEMO connector). Signal is routed to PMAC and if any of the input signals is not short circuited, PMAC prohibits any movement.

2.1.2. Device moving outputs

microIOC M-Box provides eight device moving output signals – one per axis. Two pin LEMO connectors are used (type EPL.0S.302.HLN). **`DevMov`** outputs are implemented by relays. These

signals can be used, for example, to alert other devices about movement in progress and are directly controlled with the **`WindOff`** signal for the stepper driver. Implementation is fail-safe:

- axis is moving or device has no power – relay contacts are open (as if the cable is disconnected)
- axis is stopped – relay contacts are closed

State of relay output can be changed with jumpers. NO or NC contacts can be selected. Default settings of device moving outputs are NO, as seen on Figure 7. Relay ratings specified in Table 2 should not be exceed.

Contact	Signal
1	**`DevMov`** (relay contact 1)
2	**`DevMov`** (relay contact 2)

Table 1: Device moving outputs pinout (applies to DevMov 1 – DevMov 8)

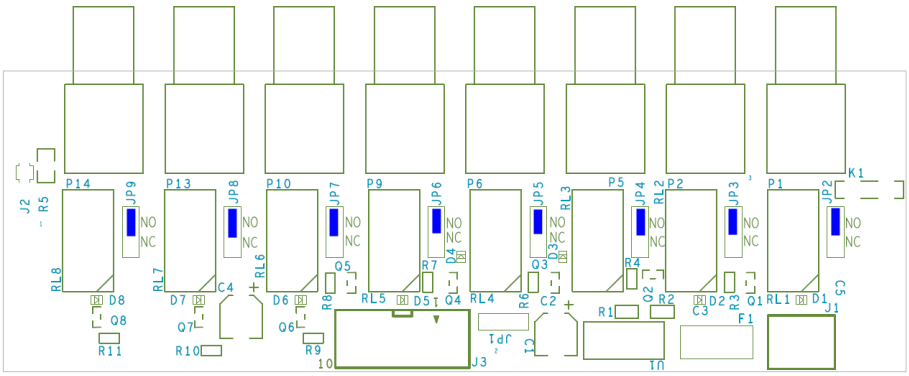

Figure 7: Device Moving output board

Item	Specifications
Nominal switching capacity	2 A 30 V DC
Max. switching power	60 W (DC)
Max. switching voltage	220 V DC
Max. switching current	2 A
Min. switching capacity	10µA 10mV DC
Insulation resistance	Between open contacts 1,000 Vrms Between contact and coil 2,000 Vrms

Table 2: Relay specifications

2.1.3. Pulse outputs

microIOC M-Box provides eight TTL pulse output signals – one per axis. One pin LEMO connectors are used (type EPL.00.250.NTN). The TTL pulse output signal is the same signal as used for driving stepper motor. Load should not exceed ±50 mA.

Contact	Signal
1	**Pulse**
shield	**GND**

Table 3: TTL pulse outputs pinout (applies to Pulse 1 – Pulse 8)

2.1.4. Interlock outputs

microIOC M-Box provides eight interlock output signals – one per axis. Five pole female M12 connectors are used (type CONEC M12 43-01199, A coding). Interlock outputs are implemented using relays.

- axis interlock or device has no power – relay contacts are open (as if the cable is disconnected)
- device is powered on, ready and in operation – relay contacts are closed

Type of outputs (NO or NC relay contacts) can be selected by using jumpers. Default settings can be seen on Figure 8. If inner limit is not needed to be present on output connector, in limit jumpers should be removed. Relay ratings specified in Table 2 should not be exceed.

Contact	Signal
1	**Common (In and Out limit)**
2	**n.c.**
3	**Dry contact – Outer limit**
4	**n.c.**
5	**Dry contact – Inner limit**

Table 4: Interlock outputs pinout (applies to OutEnd 1 – OutEnd 8)

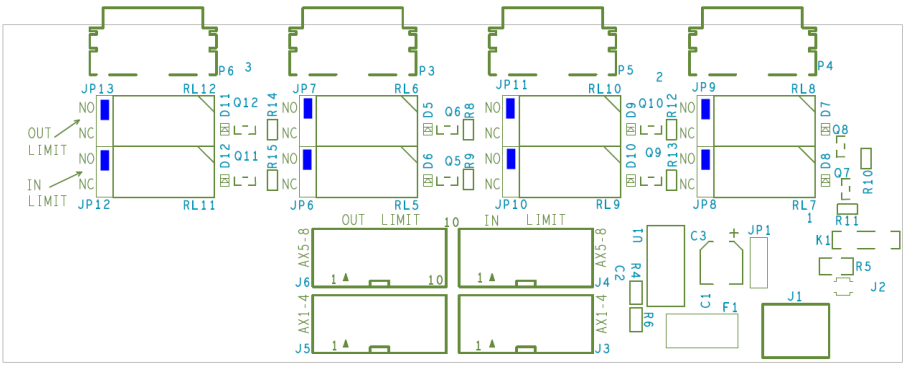

Figure 8: Interlock output board

2.1.5. RS-485 serial connectors

microIOC M-Box is equipped with a four port optically isolated RS-485 serial card [7]. Four serial ports are used for communication with RS-485 analogue modules and SSI modules inside PDC units.

Pin # DB-9	RS-422/485 Signal	Direction
1	RxD+	Input
2	RxD-	Input
3	TxD+	Output
4	TxD-	Output
5	SR	Signal Reference
6	CTS-	Input
7	RTS+	Output
8	CTS+	Input
9	RTS-	Output

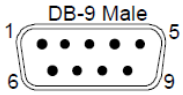

Table 5: RS-485 serial connector pinout, DB9M, (applies to RS485/1 to RS485/4)

2.1.6. PMAC serial connector

This is a standard RS-232 serial connector (DB9F) provided for a direct connection to PMAC. It is used for low-level debugging and programming of PMAC. Special development SW from DeltaTau is required.

Contact	Signal
1	PHASE – optional, insert E8 jumper on PMAC CPU board
2	TXD
3	RXD
4	DSR – internally connected to DTR
5	GND
6	DTR – internally connected to DSR
7	CTS
8	RTS
9	SERVO – optional, insert E9 jumper on PMAC CPU board

Table 6: RS232 PMAC connector pinout, DB9F

2.1.7. JTAG connector

This connector is used for updating firmware on CPLD device. The pinout is shown in Table 7.

Contact	Signal
1	VCC
2	GND
3	TCK
4	TDO
5	TDI
6	TMS

Table 7: CPLD JTAG connector pinout, DB9F

A special cable adapter is needed to connect to the Xilinx programming tool which uses a 14-pin IDC connector with 2 mm raster.

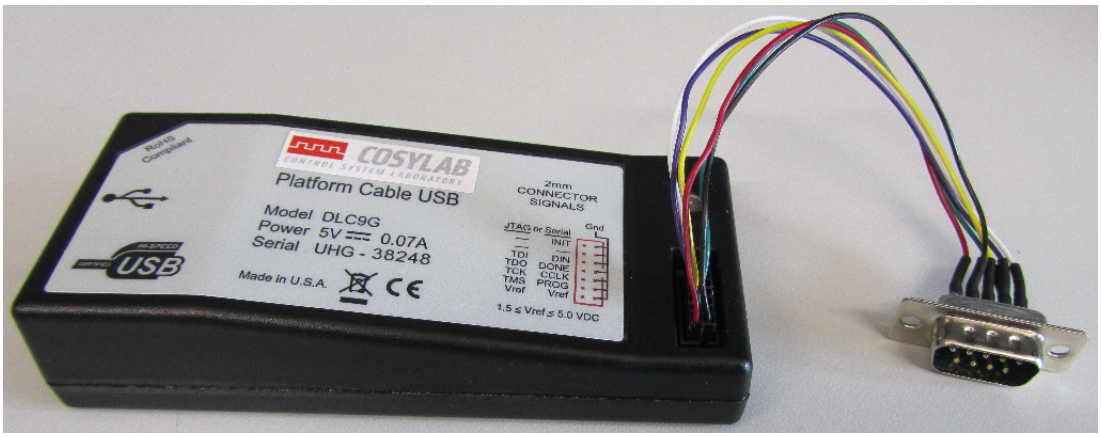

Figure 9: Xilinx programming tool connected to cable adapter

To flash CPLD download and install Xilinx ISE 14.7:

https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html

1. Connect Xilinx JTAG cable to the back panel JTAG connector (DB9F connector)
2. Open Xilinx iMPACT and create a new project.
3. Check that cable is recognized by iMPACT.
4. Initialize JTAG chain (File > Initialize Chain).
5. Readback CPLD IDs.
6. Select .jed file for CPLD.
7. Program CPLD.

2.1.8. XPORT Pro connection

XPORT Pro connects the microIOC M-Box serial port to Ethernet. It allows the user to control microIOC M-Box remotely including access to BIOS, Grub and Linux terminal. Another feature presents the option of resetting the microIOC M-Box (hard reset) by setting RESET switch to 1 for more than 3 seconds. XPORT resets SBC and PMAC.

To configure or use XPORT, the user can choose to use a web interface or connect to the device via a terminal (telnet, ssh,…). All that is needed is its IP address which can be obtained from your network administrator or by using a program named DeviceInstaller [8]. This will show all XPORTs on the local network. They can be differentiated by their MAC address.

More information on the device and how to use it can be found in microIOC Hardware installation and user's manual [15] and on Lantronix XPORT Pro web site [9].

2.1.9. Insertion of PMC FTRN card

PMC FTRN card can be additionally inserted into the M-Box on the special pre-prepared PCI-104 to PMC adapter board which is already mounted inside.

Before opening and removing the top cover of M-Box unit turn off mains switch and remove the power cord from power inlet!

Unscrew four M4 x 10mm bolts located on the top corners and two M4 x 8 mm located in the middle top of the enclosure front and back panel as depicted in Figure 10.

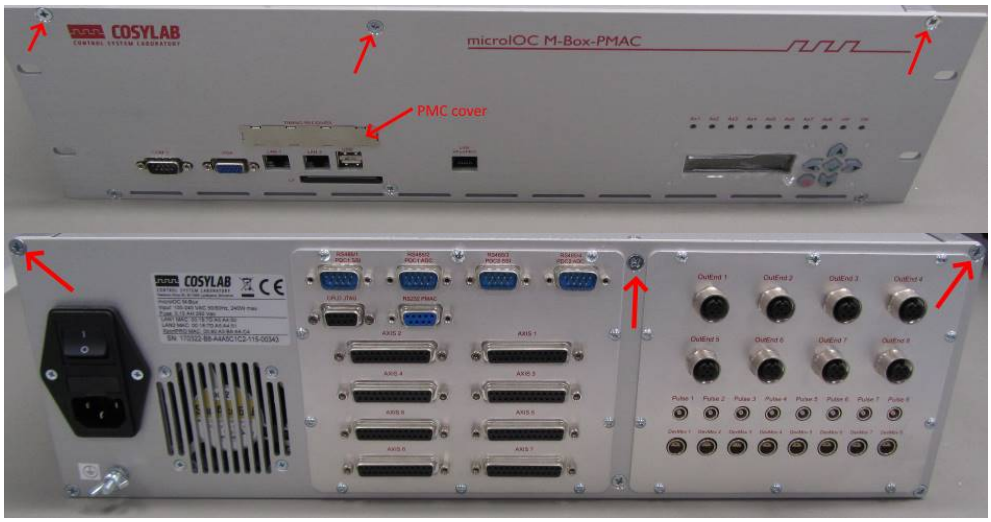

Figure 10: Removing screws for removing M-Box top cover

Now remove the top cover and place it behind the enclosure as seen in Figure 11. Unscrew four M3 x 6 mm bolts that are holding serial PCI-104 card. Carefully remove the serial card by pulling it upwards. Remove PMC metal cover from the front panel of the M-Box unit.

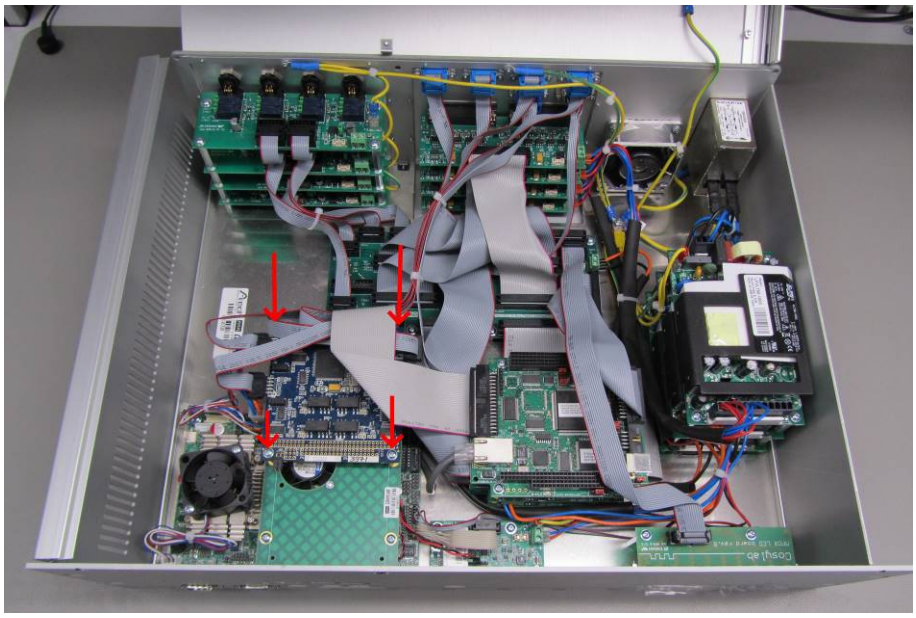

Figure 11: Removal of the top cover and serial PCI-104 card

Take PMC FTRN card and first, insert its front side through PMC hole located on the front panel of M-Box. Insert it carefully as seen in Figure 12. **Before mating it with PMC adapter board carefully align PMC connectors of PMC adapter board with PMC connectors of the FTRN as seen in Figure 13.**

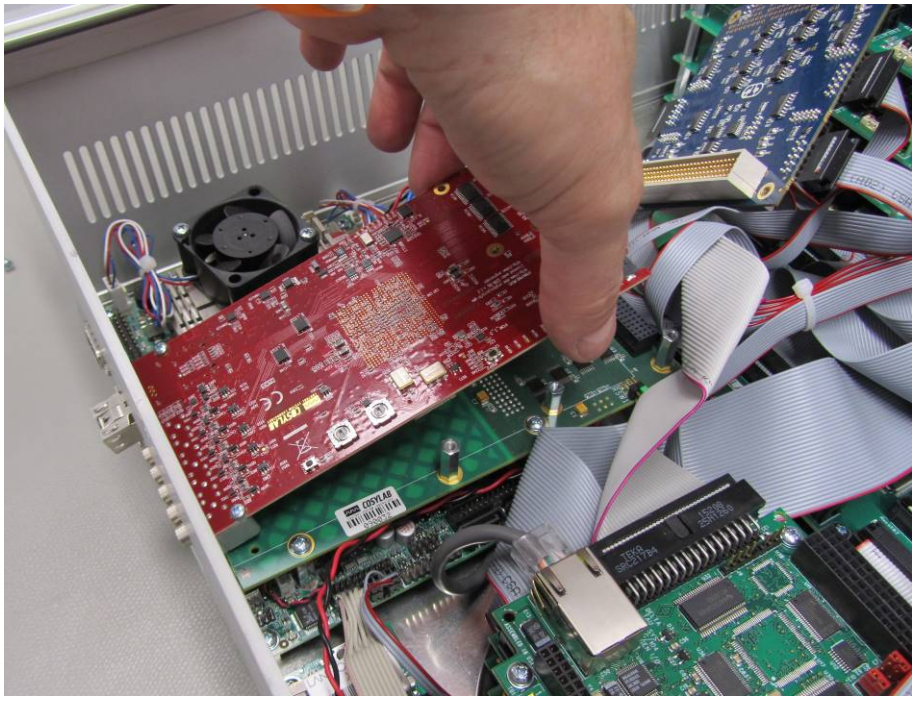

Figure 12: Insertion of PMC FTRN

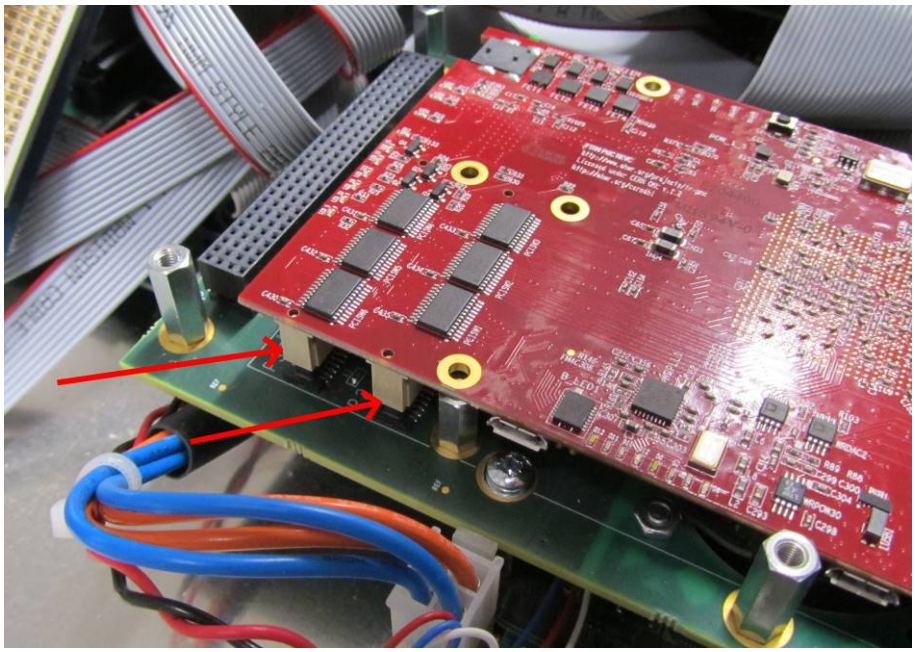

Figure 13: Alignment of PMC connectors

Fixate PMC FTRN card using two M2.5 x 6 mm screws as seen in Figure 14.

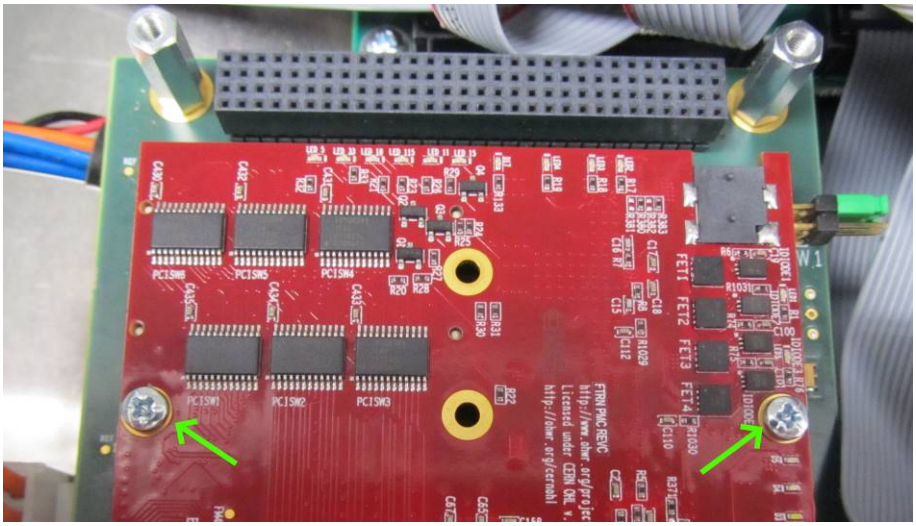

Figure 14: Fixation of PMC FTRN using two M2.5 x 6 mm screws

Take previously removed PCI-104 serial card and put it back on the top of the PMC adapter board. Carefully align PCI-104 connectors and fixate the card using four M3 x 6 mm screws as seen in Figure 15. Take the top cover of M-Box unit, mount it on its place and fixate it using the same six M4 screws that were previously removed (Figure 10).

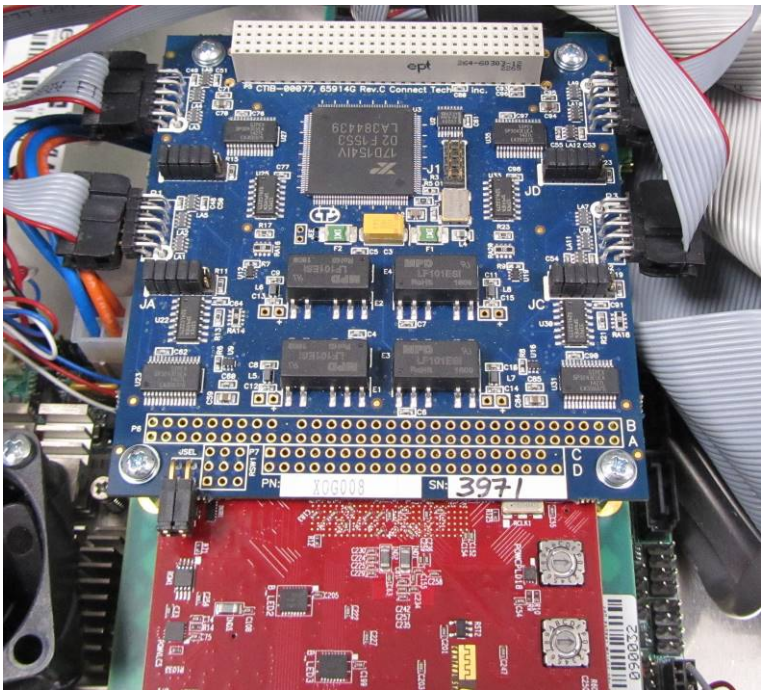

Figure 15: Mounting and fixating PCI-104 serial card

2.2. PDC

PDC extends microIOC M-Box with power driving capability for direct connection of 5-phase stepper motors and position switches. It also remotely handles voltage acquisition from feedback-potentiometers and provides it via serial RS-485 communication. The PDC block diagram is shown in Figure 3 and back panel in Figure 17. PDC is capable of serving four axes.

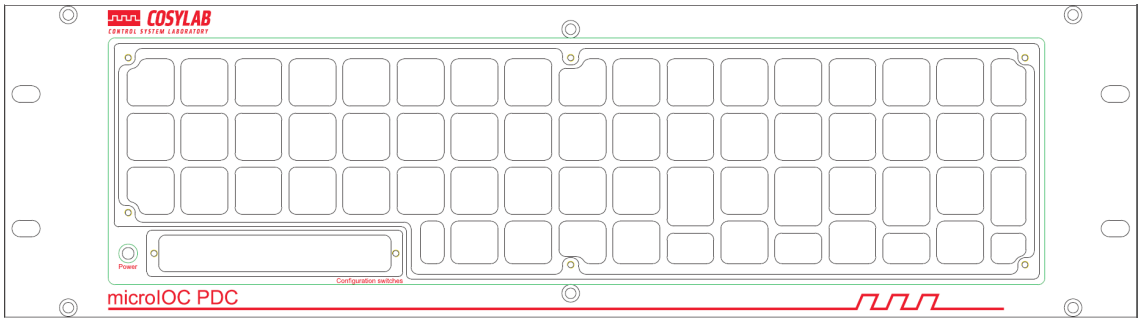

Figure 16: PDC front panel

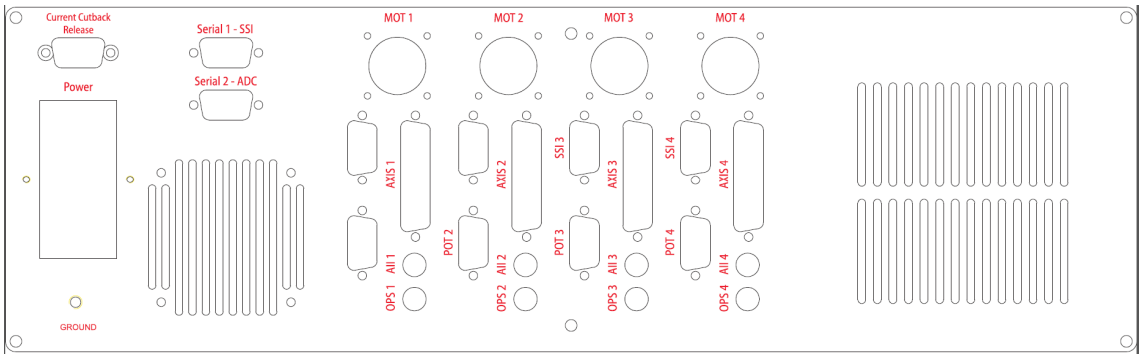

Figure 17: PDC back panel

2.2.1. Axes inputs/outputs

Axes input/output connectors (Figure 17, AXIS 1 – AXIS 4) are prepared for a direct 1-to-1 connection to microIOC M-Box-PMAC. Pinout is identical to the one shown in Figure 6. Connectors AXIS 1 – AXIS 4 are of type DB-25 female.

2.2.2. RS-485 serial connectors

Serial connector Serial 1-SSI is used for communication to the SSI module located inside PDC. Serial connector Serial 2-ADC is used for communication to the RS-485 analogue module inside PDC. Connectors are of type DB-9 female and pinouts are given in Table 8.

Contact	Signal
1, 3	RxD+, TxD+
2, 4	RxD-, TxD-
other	n.c.

Table 8: serial connections on PDC unit

2.2.3. Axis-interlock inputs

Four axis-interlock inputs (Figure 17, AII 1 – AII 4) are wired to microIOC-M-Box-PMAC.

Contact	Signal
1	Axis interlock in +
2	Axis interlock in -

Table 9: axis interlock input connector, LEMO socket type EPL.0S.302.HLN

To allow any movement, the axis interlock inputs must be short circuited.
A floating short circuit must be applied.

2.2.4. Out-position signal outputs

PDC provides four **`OutPosSig`** (out-position signal, Figure 17, ops1-ops4) outputs that are controlled directly by **`OutLimSW`** inputs. **`OutPosSig`** outputs are implemented by relays. Output of the relay is closed only when PDC is connected to a power supply and the out limit switch is hit (i.e. out limit switch is opened). Relay ratings specified in Table 2 should not be exceed.

Contact	Signal
1	out-position signal (relay contact 1)
2	out-position signal (relay contact 2)

Table 10: out-position signal output connector, LEMO socket, EPL.0S.302.HLN

2.2.5. Potentiometer and limit switches connector

Connector wiring is shown in Table 11. PDC interface circuit provides a 10V reference voltage that is applied to potentiometer end contacts. The potentiometer middle tap voltage is differentially measured using RS-485 analogue acquisition module. **Maximum output current on pin 8 – Potentiometer reference voltage, should not exceed 30mA.**

contact	signal	connection
1	**`OutLimSW`** (out limit switch)	PDC interface bord ⇨ microIOC-M-Box-PMAC
2	**`TchLim`** (touching limit switch)	microIOC-M-Box-PMAC
3	common contact for limit switches	24V power supply
4	**`InLimSW`** (in limit switch)	PDC interface bord ⇨ microIOC-M-Box-PMAC
5	N.C.	
6	Potentiometer reference voltage GND	Reference voltage GND RS-485 analogue_in_-
7	Potentiometer middle tap	RS-485 analogue_in_+
8	Potentiometer reference voltage +10V	Reference voltage (+10V) RS-485 analogue_in_+
9	N.C.	

Table 11: Potentiometer/feedback connector pinout (DB9F)

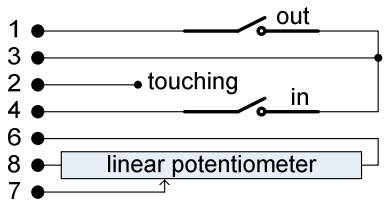

Figure 18: potentiometer and limit switch connector (DB9F)

Output signal **`OutPos`** (out-position signal, Section 2.2.4) is controlled by **`OutLimSW`** (out limit switch). **`OutLimSW`** connection position is thus important.

2.2.6. Stepper motor connector

Connector wiring is shown in Table 12. Maximum output current 2,8A/phase should not be exceeded.

Contact	Signal	connection	
A	Phase W1	1	Stepper motor drive
B	Phase W2	4	
C	Phase W3	2	
D	Phase W4	5	
E	Phase W5	3	
F	Phase W1/	2	
G	Phase W2/	5	
H	Phase W3/	3	
J	Phase W4/	1	
K	Phase W5/	4	
L	Brake +	Brake + 24 V	Brake circuitry
M	Brake GND	GND	

Table 12: stepper motor connector (UTG01412S, SOURIAU) and connection to stepper driver

2.2.7. SSI module and encoder connectors

The SSI module reads data from encoders. Position of the encoders is readable through two wire RS-485 communication. Communication speed is set to 115200 baud.

Short summary of the implemented commands:

g[N].[C]\n Get position of N-th sensor with sense pin set to C, where N can be in range from 0-3. C can be 0 or 1.

gv\n Get firmware version.

gt\n Get temperature reading as voltage from NTC

In case of an invalid command, "E" is returned signaling that the input from user could not be interpreted. Connector wiring is shown in Table 13. The SSI module also provides 24V power supply for SSI encoders.

Pin on DB9F	Signal name
1	SSI_CLK +
2	SSI_CLK -
3	SSI_DATA +
4	SSI_DATA -
5	Enable
6	+24V
7	+24V
8	GND
9	GND

Table 13: SSI encoder connector pinout (DB9F)

2.2.8. Serial module and analog input signals

Serial module ADAM-4117 [6] provides eight 16-bit analogue differential channels. Module is configured for input voltage range of 0-10V to allow potentiometer position read-back. Communication speed is set to 115200 baud. Potentiometer and reference voltages can be read using commands:

#02\n Get value of all eight channels

#02[N]\n Get value of N-th channel, where N can be in range from 0-7

Input channel	description
0	voltage from potentiometer 1
1	reference voltage +10V
2	voltage from potentiometer 2
3	reference voltage +10V
4	voltage from potentiometer 3
5	reference voltage +10V
6	voltage from potentiometer 4
7	reference voltage +10V

Table 14: serial module input output signals

2.2.9. Middle switch configuration

Configurations switches and jumpers for middle switch are located on the PDC front panel under the cover.

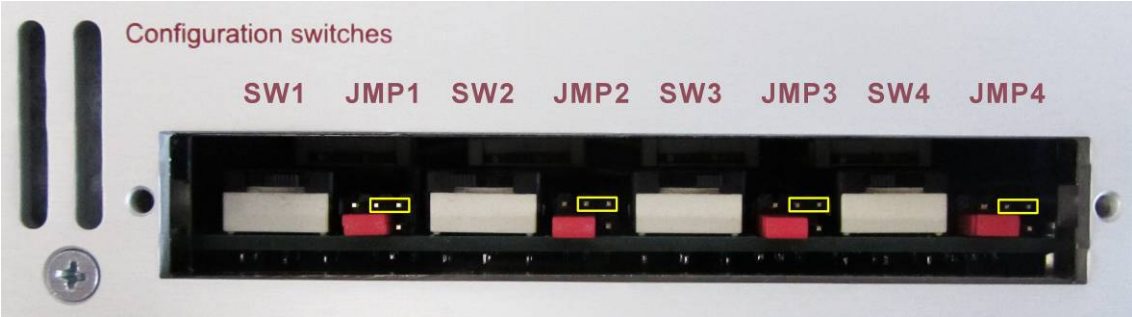

Figure 19: PDC middle switch configuration switches

Double row jumpers are used; the bottom row is suffixed with "a" and the top row is suffixed with"b". The top row **jumpers (JMP1b to JMP4b), is used to enable/disable middle switch functionality.** To disable middle switch functionality insert jumper between pins 2-3 (right position, Figure 19 – yellow markings).

Settings for middle switch support are detailed in Table 15.

Item	Setting for N.O. middle switch	Setting for N.C. middle switch
SW1	Right	Left
SW2	Left	Right
SW3	Right	Left
SW4	Left	Right
JMP1a	1-2	2-3
JMP2a	1-2	1-2
JMP3a	1-2	2-3
JMP4a	1-2	1-2
JMP1b to JMP4b	Not inserted	Not inserted

Table 15: PDC middle switch jumper settings

2.2.10. Current Cutback Release connector

Current Cutback Release connector is used to set motor operating and motor standstill current.

Adjusting the Motor Operating Current:

Set "Automatic Current Cutback Release" (C.D.INH) signal to the "photocoupler ON" state when adjusting the RUN current. Adjustment range is from 1.0A/phase to 2.8A/phase. Current can be readjusted using the RUN potentiometer. The operating current can be lowered to suppress temperature rise in the motor/driver, or lower operating current in order to allow a margin for motor torque or to reduce vibration. The motor RUN current should be less than the motor rated current.

Adjusting the Current at Motor Standstill:

Set "Automatic Current Cutback Release" (C.D.INH) signal at in the "photocoupler OFF" state when adjusting the current while the motor is stopped. Adjustment range is from 0.7A to 2.3A/phase. The STOP potentiometer can be used to readjust the current at motor standstill to the current value required to produce enough holding torque.

Axis	Driver pin	DB9 connector pin
1	Pin 9 Pin 10	Pin 1 Pin 6
2	Pin 9 Pin 10	Pin 2 Pin 7
3	Pin 9 Pin 10	Pin 3 Pin 8
4	Pin 9 Pin 10	Pin 4 Pin 9

Table 16: Connection/pinout of current cutback release connector

2.3. PDC WIRING SCHEME

Detailed wiring scheme and schematics of the PDC adaptation circuit is given in [9].

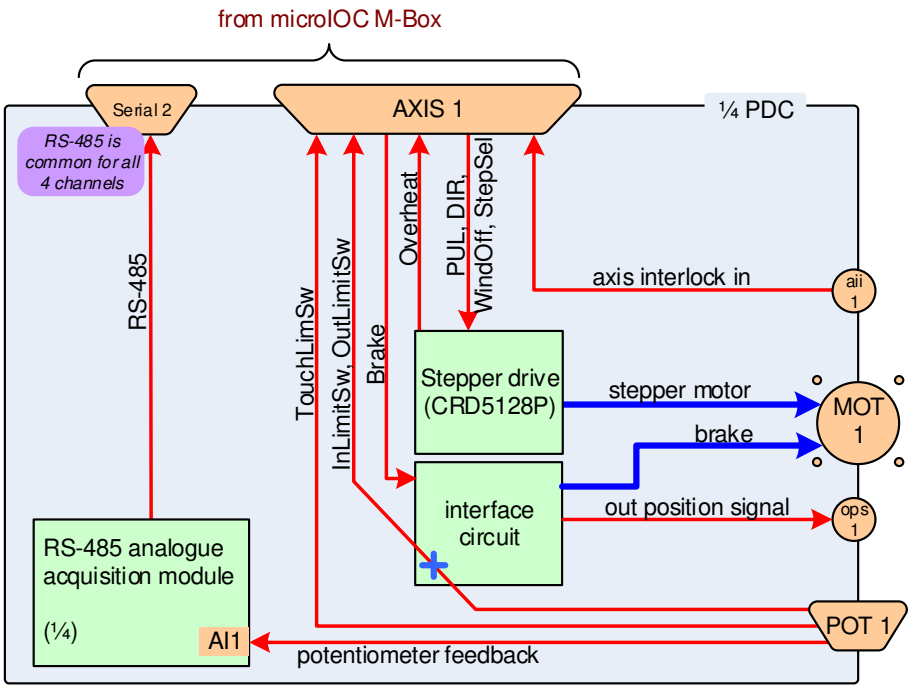

Figure 20: Principle wiring scheme within PDC

2.4. CABLING BETWEEN MICROIOC M-BOX-PMAC AND PDC

Maximal cable length between M-Box and PDC unit is 250 meters (AXIS and serial communication cables). Maximal distance between PDC and motor is 60 meters (MOT and POT cables).

2.4.1. Axis signal connection cable

Each axis requires one signal connection cable. Both sides should have identical DB-25 male connectors with one-to-one connection. This connection transmits only low-power signals: 24V / 10~15mA. Twisted-pair wiring should be used on the following pins:

- 1-2, 3-4, 5-6, 7-8, 11-12, 14-15, 16-17, 18-19, 22-23, 24-25

Pins 9, 10, 13, 20, 21 are not used. The extension cable must be shielded and shield connected to both connectors' casings.

2.4.2. PDC communication cable

PDC requires two communication cables. Straight 1:1 DB-9 serial cable with male connector on one and female connector on the other side should be used. This cable transmits only low-power

RS-485 signals. Twisted-pair wiring should be used for the pins 1, 3 and 2, 4. The other pins are not used. Shielding must be connected to the connectors' casings.

3. SOFTWARE

As show on figure below main components of the software are:

- FESA device class
  - FESA class which models motors and/or pairs of motors (slit)
- Local control server
  - Local control server used for motor configuration
- Local control GUI
  - Local control GUI for motor configuration
- System driver
  - C++ system driver that communicates with PMAC via TCP/IP
  - Communicates with any other additional device (e.g. encoders) via serial connection
  - Transforms requests from upper level software to PMAC commands
  - Handles system mode (local control/remote/LCD control)
- Access control monitor
  - Used to implement access control fall-back and shared memory cleanup in case any of the application crashes unexpectedly.
  - Used to control the LEDs on microIOC M-Box front panel
  - Used to monitor minimum distance for pair of motors in case hardware limit switch is disabled
- LCD control
  - Used for controlling motors locally from the LCD on the front panel of M-Box.
- PMAC software
  - Low level software used to control single or pair of motors which resides on PMAC controller inside M-Box.
- VHDL code
  - Used for signal configuration

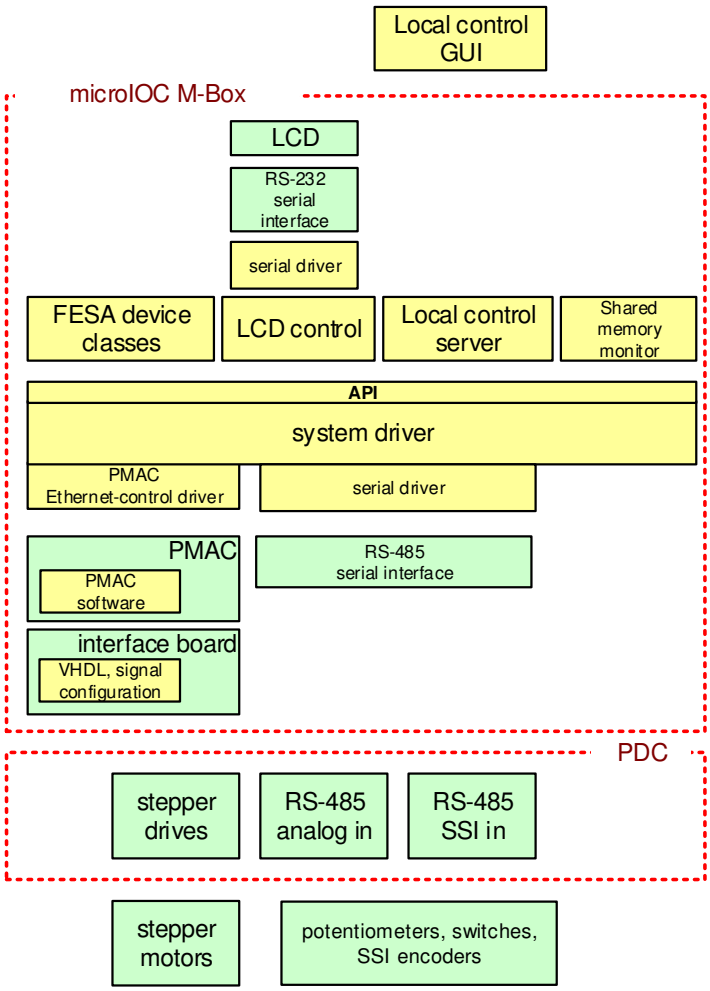

Figure 21: system overview

Details of the software can be found in [11], [12], [13] and [14].

3.1. Roles of software components

Current arrangement of the position-feedback system used in GSI (reading voltage of the potentiometer middle tap, moving in parallel with axis of the motor) does not provide the relation, for which unique calculation could be defined for different axes. In fact, each axis has to be visually (or by means of end switches) inspected for both end positions, where voltages from position potentiometers should be captured and stored. The next parameter to be captured is the number of steps of the stepper motor for the entire allowed range of movement. Using these parameters, the required transformation steps-to-actual-movement-in-mm can be calculated. To simplify this procedure local control GUI and server were developed which provides functionality for commissioning of the motors without the need for FESA device class software. Local control is explained in detail in [12].

Once the required parameters are known, the PMAC's motor positions should be reset. The system driver API provides the means to move a single motor or a pair of motors to an absolute position. The PMAC provides all the required assistance for moving a pair of motors, but it has no way to

check if the movements have actually taken place. The PMAC can only issue a sequence of steps, for which it expects the motors will move accordingly. It is the role of the application to request a move (from a PMAC) and check if the axes have moved accordingly. This can be done by reading the feedback potentiometer voltages and using the above mentioned relation for calculation.

The features that are in the domain of the PMAC can be easily set using an API: maximum motor velocity, acceleration (deceleration) time (PMAC handles acceleration and deceleration automatically using these parameters), internal position counters (steps sent to stepper driver), required position and others. See [12] and [14] for details on what can be configured. There are also some parts of the motion control that PMAC can handle automatically based on the data it reads, for example, not allowing movement when the axis interlock input signal is active – when this happens, the control system calls cannot override this, but it can read the status and figure out why the movement is not happening.

3.2. PMAC SOFTWARE

PMAC is a very flexible and high-performance motion control system. The system runs motion control programs that are specific to the application being controlled. If a modified motion control scheme is required, a PMAC can be easily reconfigured and/or its API extended. For details see [3] and [12].

3.2.1. PMAC-CPLD memory access

To enable flexible reconfiguration of the PMAC interface board, it is equipped with a CPLD device (Section 3.3). In a CPLD there are configuration and status registers that are memory mapped to PMAC memory space via JOPT and JTHW ports. This enables 8-bit memory accesses to registers inside CPLD. PMAC software can read (or write) from (or to) CPLD registers with use of M-variables which are pointing to JOPT and JTHW ports. Variable mapping is as follows:

```
M48->Y:$078402,8,8,U ; SEL0-7 lines treated as a byte (ADDRESS)
M58->Y:$078402,0,8,U ; DAT0-7 lines treated as a byte (DATA_IN)
M64->Y:$078400,0,8,U ; I/O00-7 lines treated as a byte (DATA_OUT)
```

3.2.1.1. Read process

PMAC can read from a given CPLD register by issuing the following procedure:

```
M48=ADDRESS  ;set the CPLD ADDRESS bits from which we are reading
;timer (wait for e.g. 1ms)
M58 ;read 8bit DATA that is accessible on M58 variable
```

3.2.1.2. Write process

PMAC can write to a given CPLD register by issuing the following procedure:

```
M48=ADRESS ;set the CPLD ADDRESS bits which user wants to write to
M64 = DATA ; set DATA (JOPT bits) which user wants to write
```

```
;timer (wait for e.g. 1ms)
M48=$D4 ;set the ADDRESS bits to 0xD4 which signals 'write command'
to CPLD logic
```

3.3. PMAC INTERFACE BOARD AND CPLD CONFIGURATION

Outputs of the PMAC are not directly routed to axes connectors. To provide noise immune communication, all the signals are optically isolated and raised to 24V voltage level. For flexibility and future adaptations almost all signals are either monitored by a CPLD or the signals are passed through it. CPLD is a programmable integrated circuit with multiple general-purpose inputs/outputs, which can have user defined relation among them. For example, logical functions between signals or state machine processing can be applied.

Given that additional signals—besides PMAC natively supported signals—are required, these signals are implemented using CPLD; i.e. **`Brake`**, **`StepSel`**, **`OverHeat`**, etc. A CPLD implements PMAC memory mapped registers that control these additional signals. The access to registers is implemented via PMAC expansion ports (see 3.2.1).

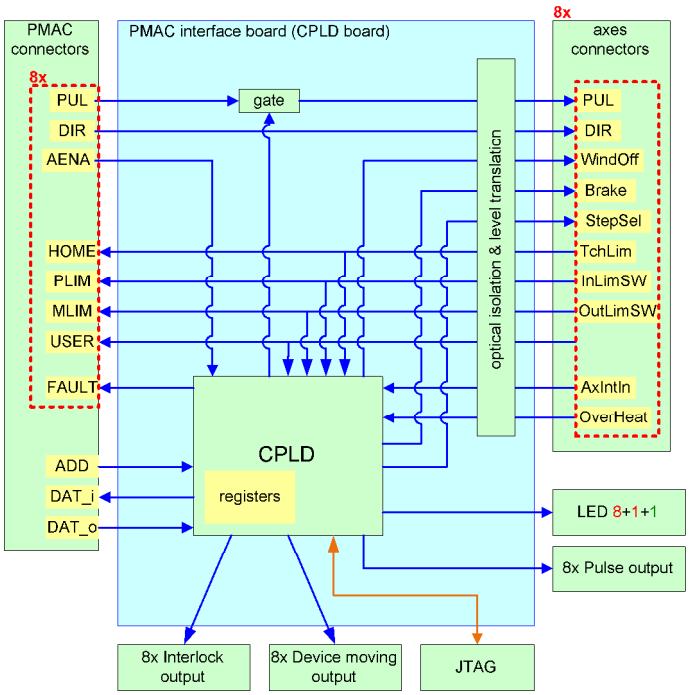

Figure 22: PMAC interface board overview

address	value	R/W	
0x01	**InLimSW** bit3: axis-4 → bit0: axis-1	R	In limit switch, 1 = opened, 0 = closed
0x02	**OutLimSW** bit3: axis-4 → bit0: axis-1	R	Out limit switch, 1 = opened, 0 = closed
0x03	**TchLimSW** bit3: axis-4 → bit0: axis-1	R	touch limit switch
0x04	**USER** bit3: axis-4 → bit0: axis-1	R	not used input on DB25(20,21) connector, will read 1
0x05	**OverHeat** bit3: axis-4 → bit0: axis-1	R	overheat signal from stepper drive (1 when ok, 0 when overheat)
0x06	**AxIntIn** Bit7: axis-8 → bit0: axis-1	R	axis interlock inputs (for movement to be allowed, inputs must be 0)
0x07	**AxesLEDs1** bit7: axis-8 → bit0: axis-1	R/W	Combination of this register and AxisLEDs2 register determine the LED state.
0x08	**MasterLEDs**	R/W	b1 and b0 determine the state. (OFF, Blink SLOW, Blink FAST, ON)
0x09	**VHDLversion**	R	read version of the CPLD configuration
0x0A	**Brake** bit3: axis-4 → bit0: axis-1	R/W	BRAKE output: 1 = brake is off, 0 = brake is on this signal should only be controlled by PMAC
0x0B	**DeviceMove**	R/W	Device moving outputs
0x0C	**AxisLEDs2** bit7: axis-8 → bit0: axis-1	R/W	Combination of this register and AxisLEDs1 register determine the LED state.
0x0D	**StepSel** bit3: axis-4 → bit0: axis-1	R/W	step select signal for stepper drive 1 = half step, 0 = full step

Table 17: CPLD registers' descriptions

Register \ LED	7	6	5	4	3	2	1	0
MasterLED1 (A)	Ab7	Ab6	Ab5	Ab4	Ab3	Ab2	Ab1	Ab0
MasterLED2 (B)	Bb7	Bb6	Bb5	Bb4	Bb3	Bb2	Bb1	Bb0

Table 18: CPLD LED registers

There are 4 possible LED states for each LED.

State \ Register	A	B
OFF	0	0
Blink SLOW (1Hz)	0	1
Blink FAST (4Hz)	1	0
ON	1	1

Table 19: LED states

Each of the axis LEDs is coded with two bits from corresponding register as shown in Table 18. The two bits determine the state of LED as shown in Table 19. The master LED (General Fault) is coded in the same way with the b1 and b0 of the MasterLEDs register.

By using the provided command line application (and API), reading and writing to CPLD registers can be done by issuing commands **`writecpld`** and **`readcpld`**. All registers are 8-bit in size.

4. Specifications

This chapter contains specifications for the M-Box and PDC unit.

4.1. Electrical

Input voltage range	100 to 240 V AC
Operating voltage range	90 to 264 V AC
Input frequency	50/60 Hz
Operating frequency range	47 to 63 Hz
Power rating	microIOC M-Box, 240 W max. PDC, 720 W max.
Over current protection	M-Box, 2x T3.15A H 250 VAC PDC, 2x T6.3A H 250 VAC
Efficiency	>80% typical

4.2. Operating Environment

Maximum altitude	2000 m
Pollution degree	2
Ambient temperature range	5 to 40°C
Relative humidity range	10 to 90%, noncondensing

For indoor use only.

4.3. Storage Environment

Ambient temperature range	-20 to 60°C
Relative humidity range	10 to 95%, noncondensing