General Mechatronics
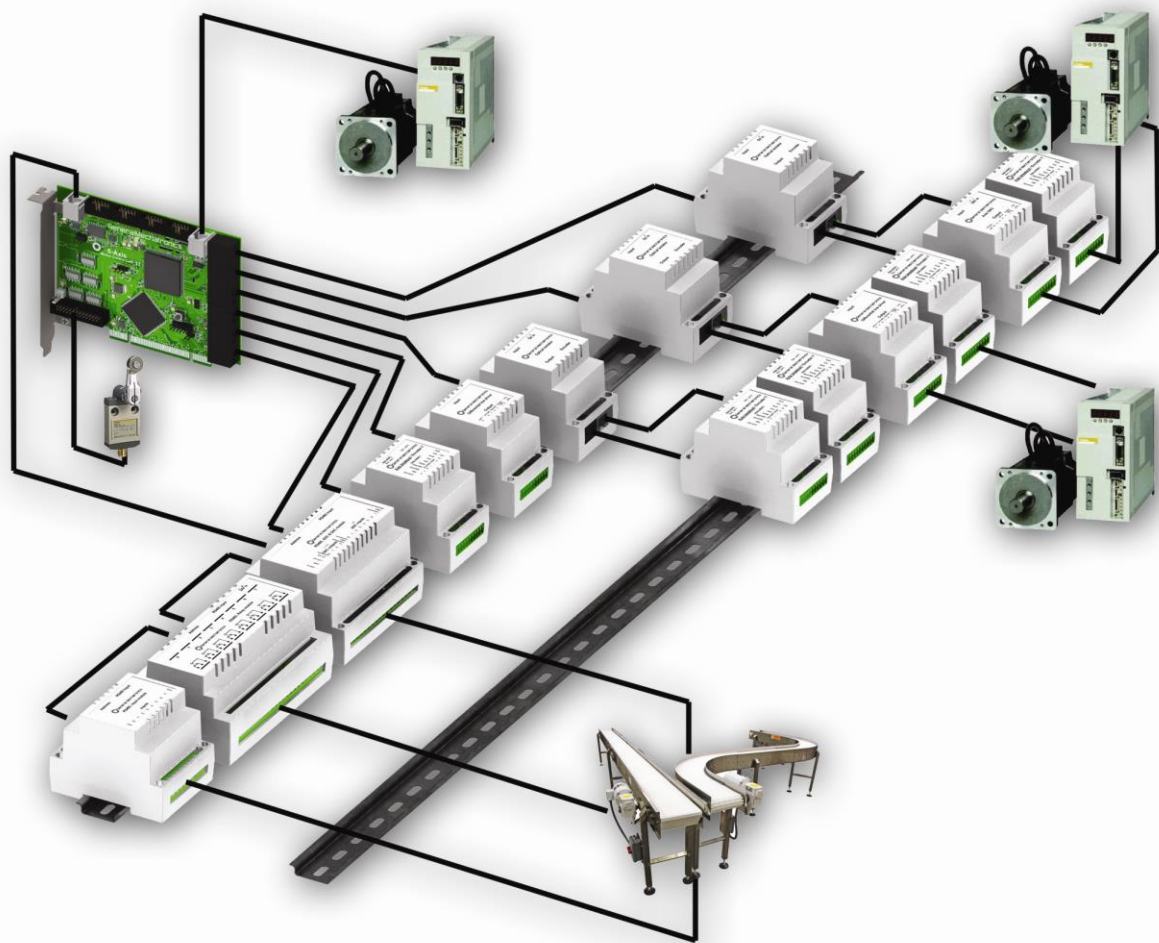
MOTION CONTROL SYSTEM
PCI version 1.2.1

SYSTEM INTEGRATION MANUAL

7th edition: 2015.01.26

TABLE OF CONTENTS

1. INTRODUCTION

This manual gives instructions and technical information for system integration of the General Mechatronics PCI motion control system. The system is suitable for many servo applications like robots, CNC machines, latches, plasma cutters etc. It can be easily integrated into LinuxCNC[1] for a complete machine control solution.

1.1. Features and interfaces

The PCI motion control card is based on an FPGA and a PCI bridge interface ASIC. There are connectors on the whole border of the card for interfacing with most possible system elements that can be found in an automated robotic cell.

- There are six RJ50 modular axis connectors at the inner edge of the card for driving incremental or classic analogue servos. Four different small DIN-rail mounted axis modules are available for making possible to drive six different servo configurations in a cost effective way. The interface can be made with an exact combination of the suitable axis modules for the actual servo configuration[2].

- A CAN bus interface can be seen at the top edge from the right side (RJ12 modular connector): this is for modern digital servo systems those communicate on CAN-bus.

- Then in the middle, four times eight general purpose I/O pins are placed on standard flat cable headers. These bare I/O pins can be configured in LinuxCNC for any custom purpose. In most cases an optical isolation is advisable here for preventing the PCI card and the PC from disturbances and unsafe voltage transients coming from the industrial environment.

- On the left side of the top edge an RS485 bus was designed for interfacing with compact DIN-rail mounted expander modules. An 8-channel digital input, an 8-channel relay output, an analogue I/O (4x ±10 Volts output and 8x ±5 Volts input) and a Teach Pendant modules are available now. Maximum 16 modules can be connected to the bus altogether.

- At the end, on the bottom, there is a 26 pin flat cable header with 20 optically isolated input pins. Six times three for the direct connection of two end switch and one homing sensor for each axis. And additionally, two optically isolated E-stop inputs.

With these functions a small automated manufacturing cell can be controlled with success, with a short time system integration procedure. The following figure demonstrating the typical connection of devices related to the control system.

---

[1] PC based motion control system. For more information, please visit the LinuxCNC official website: http://www.linuxcnc.org/

[2] See chapter 3.1.1. for the possible servo types and configurations
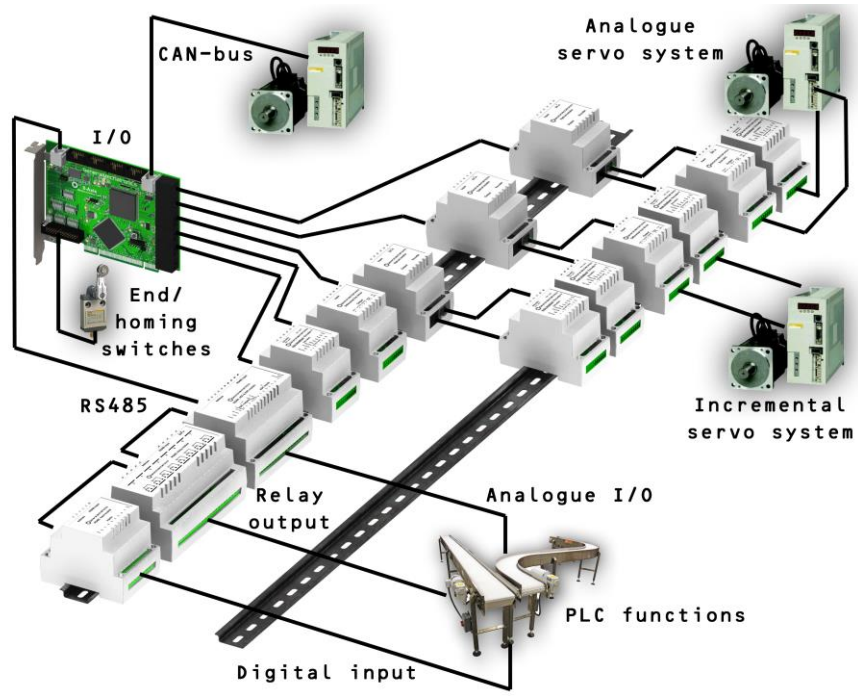
Connection layout of PCI card based motion control system

2. PCI CARD – SPECIFICATIONS

This chapter describes all connectors, pins and the electrical characteristics of the PCI card.

2.1. Pin-outs and electrical characteristics
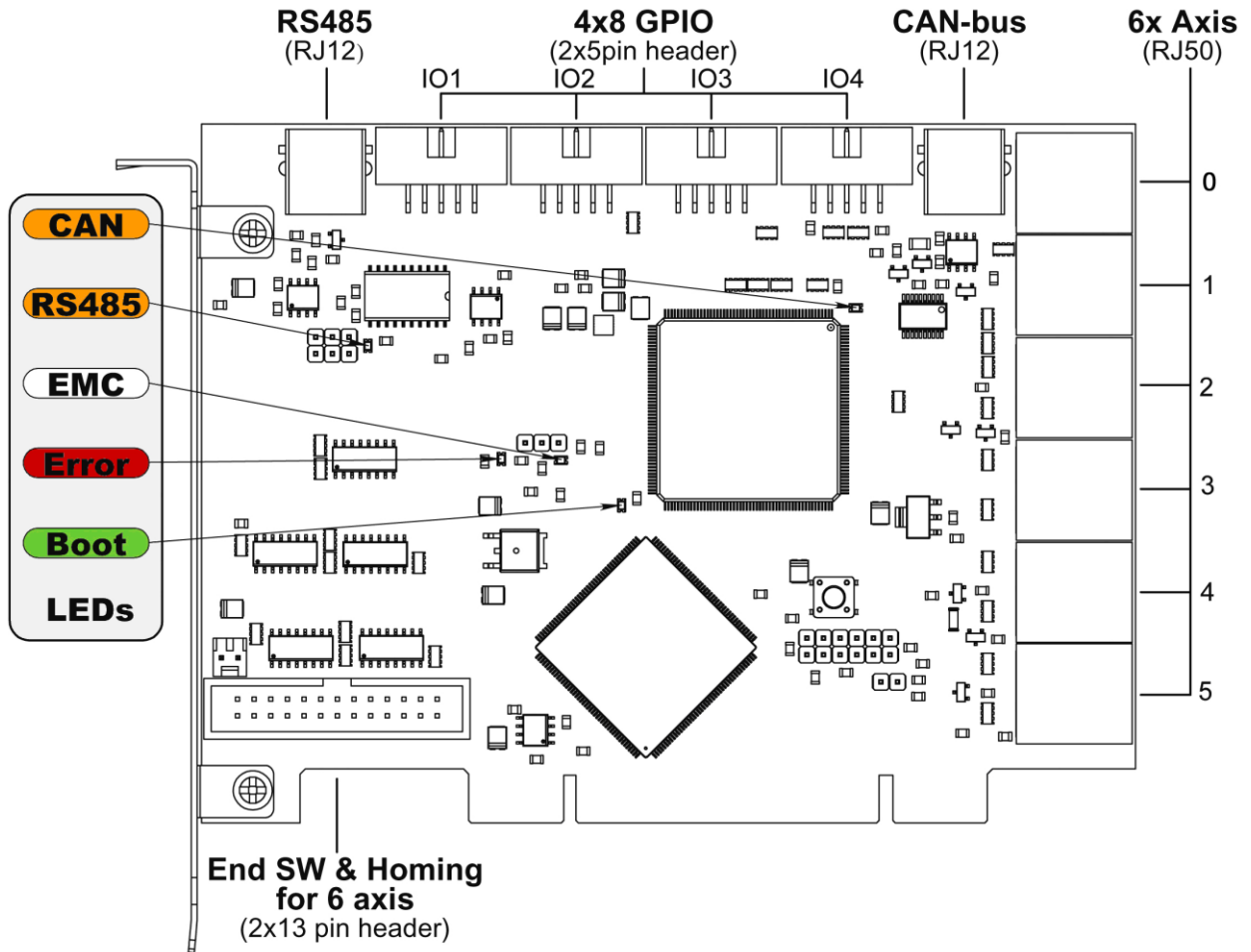
PCI card connectors and LEDs

2.1.1. RS485: Extension modules

1	Ground (PC)
2	Ground (PC)
3	RS485-A
4	RS485-B
5	+12 Volt (PC)
6	+12 Volt (PC)
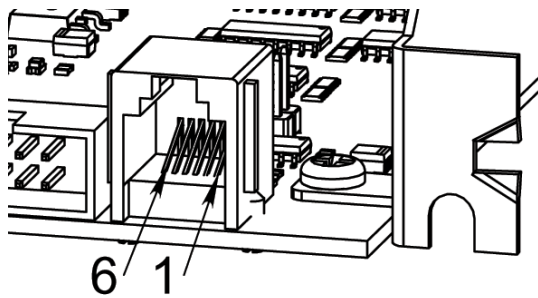
Pin numbering of RS485-bus connector   Pinout of RS485-bus connector

2.1.2. GPIO connectors

Each pin can be configured as digital input or output.

2.1.2.1. Pinout

9	7	5	3	1
IOx / 7	IOx / 5	IOx / 3	IOx / 1	VCC

10	8	6	4	2
GND	IOx / 6	IOx / 4	IOx / 2	IOx / 0
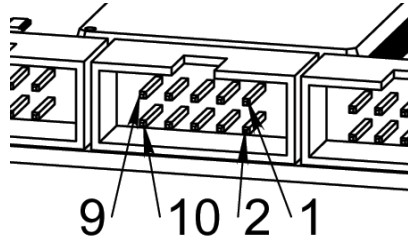
Pin numbering of GPIO connectors   Pinout of GPIO connectors

2.1.2.2. Input electrical characteristics

These are simple non-isolated I/O ports for general purpose usage. All voltage levels are referenced to the PC ground. It is not recommended to connect directly signals those can exceed the absolute maximum ratings.

Absolute maximum ratings

Maximum input voltage:	**3.3**	Volts
Minimum input voltage:	**0**	Volts.
Maximum current of the 4x8 output	**100**	mA

Logic levels

Minimum high-level input voltage:	**2**	Volts
Maximum low-level input voltage:	**0.8**	Volts

2.1.2.3. Output electrical characteristics

The totem pole outputs are short circuit protected by a series resistor, see figure:
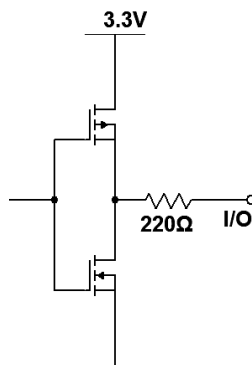
Equivalent circuit of an output pin[3]

Output logic levels (unloaded)

Minimum high-level output voltage:	**2.9**	Volts
Maximum low-level output voltage:	**0.4**	Volts

---

[3] direction of I/O pins depending on configuration see chapter 4.9

2.1.3. CAN-bus: Position reference for servo modules

1	Fault
2	Ground (PC)
3	CAN Low
4	CAN High
5	+5 Volt (PC)
6	Power Enabled
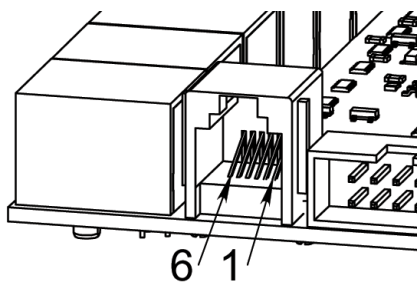
Pin numbering of CAN-bus connector   Pinout of CAN-bus connector

2.1.4. Axis connectors

All voltage levels on the axis connectors are referenced to the PC ground. It is recommended to use axis modules for connecting servo amplifier modules to the PCI card. See chapter 3.

1	Encoder A
2	+5 Volt (PC)
3	Encoder B
4	Encoder Index
5	Fault
6	Power Enabled
7	Step/CCW/B
8	Direction/CW/A
9	Ground (PC)
10	DAC serial line
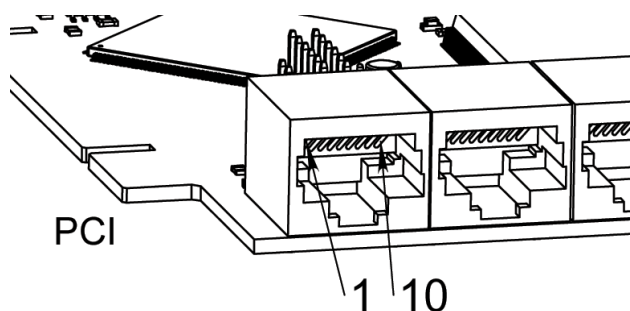
Pin numbering of axis connectors   Pinout of axis connectors

2.1.4.1. *Encoder* input characteristics

Absolute maximum ratings

Maximum input voltage:	**3.3**	Volts
Minimum input voltage:	**0**	Volts.

Logic levels

Minimum high-level input voltage:	**2**	Volts
Maximum low-level input voltage:	**0.8**	Volts

2.1.4.2. *Fault* inputs
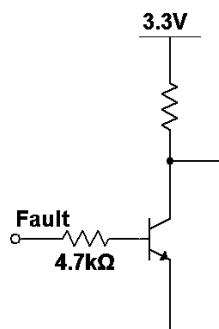
Equivalent circuit of fault input for an axis

Logic levels

Minimum high-level input voltage:	**1.5**	Volts
Maximum low-level input voltage:	**0.5**	Volts

2.1.4.3. *Enabled* outputs

The enable output is common for all axes.

    Maximum output source current: **100** mA
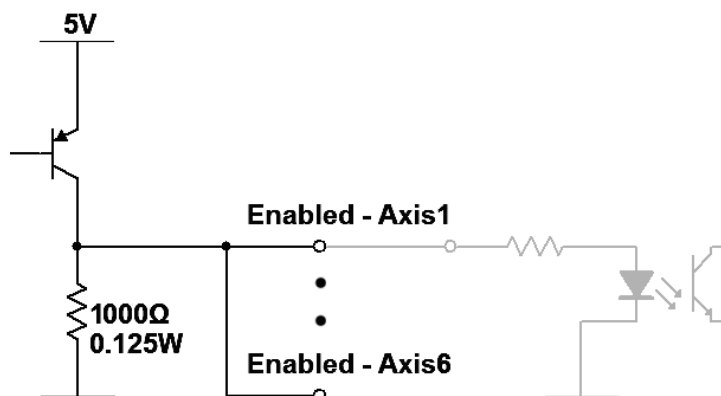
Equivalent circuit of enabled outputs

2.1.4.4. *Step*, *Direction* and *DAC serial line* output characteristics

The totem pole outputs are short circuit protected by a series resistor, see figure.
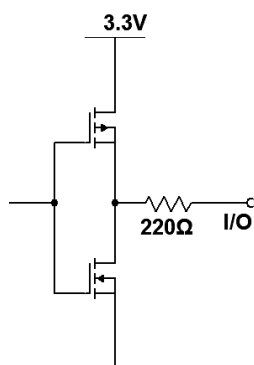
Equivalent circuit of output pins (Step, Direction, and DAC serial line) on axis connectors

Output logic levels (unloaded)

    Minimum high-level output voltage: **2.9** Volts

    Maximum low-level output voltage: **0.4** Volts

2.1.5. Homing & end switch connector

2.1.5.1. Pinout
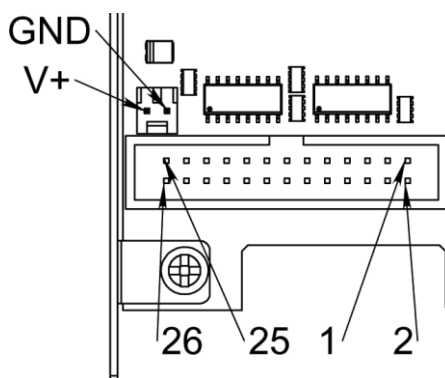
Pin numbering of homing & end switch connector

25	23	21	19	17	15	13	11	9	7	5	3	1
GND (External)	0/End-	1/End+	1/Homing	2/End-	3/End+	3/Homing	4/End-	5/End+	5/Homing	E-Stop 1	**V+** (Ext.)

26	24	22	20	18	16	14	12	10	8	6	4	2
GND (External)	0/End+	0/Homing	1/End-	2/End+	2/Homing	3/End-	4/End+	4/Homing	5/End-	E-Stop 0	**V+** (Ext.)

Pinout of homing & end switch connector

2.1.5.2. Input electrical characteristics
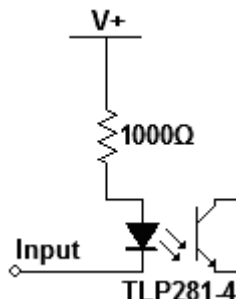
Equivalent circuit of input pins

Mechanical switch or open collector output can be connected between input pins and external ground (**GND**).

Logic levels referred to voltage drop between *V+* and *Input pins* and current consumption

Minimum high-level voltage drop:	**5** Volts	I_F **3.9** mA
Maximum recommended high-level voltage drop:	**12** Volts	I_F **11** mA
Maximum low-level voltage drop:	**1** Volts	

Absolute maximum ratings

Forward voltage drop between *V+* and *Input pins*: **14** Volts

Forward current along *V+* and *Input pins*: **13** mA

Reverse voltage drop between *V+* and *Input pins*: **5** Volts

2.1.6. LEDs

2.1.6.1. CAN

Color: Orange

- Blink, during data communication.

- On, when any of the buffers are full – communication error.

- Off, when no data communication.

2.1.6.2. RS485

Color: Orange

- Blink, during initialization of modules on the bus

- On, when the data communication is up between all initialized modules.

- Off, when any of the initialized modules dropped off because of an error.

2.1.6.3. EMC

Color: White

- Blink, when LinuxCNC is running.

- Otherwise off.

2.1.6.4. Boot

Color: Green

- On, when system booted successfully.

- Otherwise off.

2.1.6.5. Error

Color: Red

- Off, when there is no fault in the system.

- Blink, when PCI communication error.

- On, when *watchdog timer* overflowed.

2.2. Mechanical dimensions
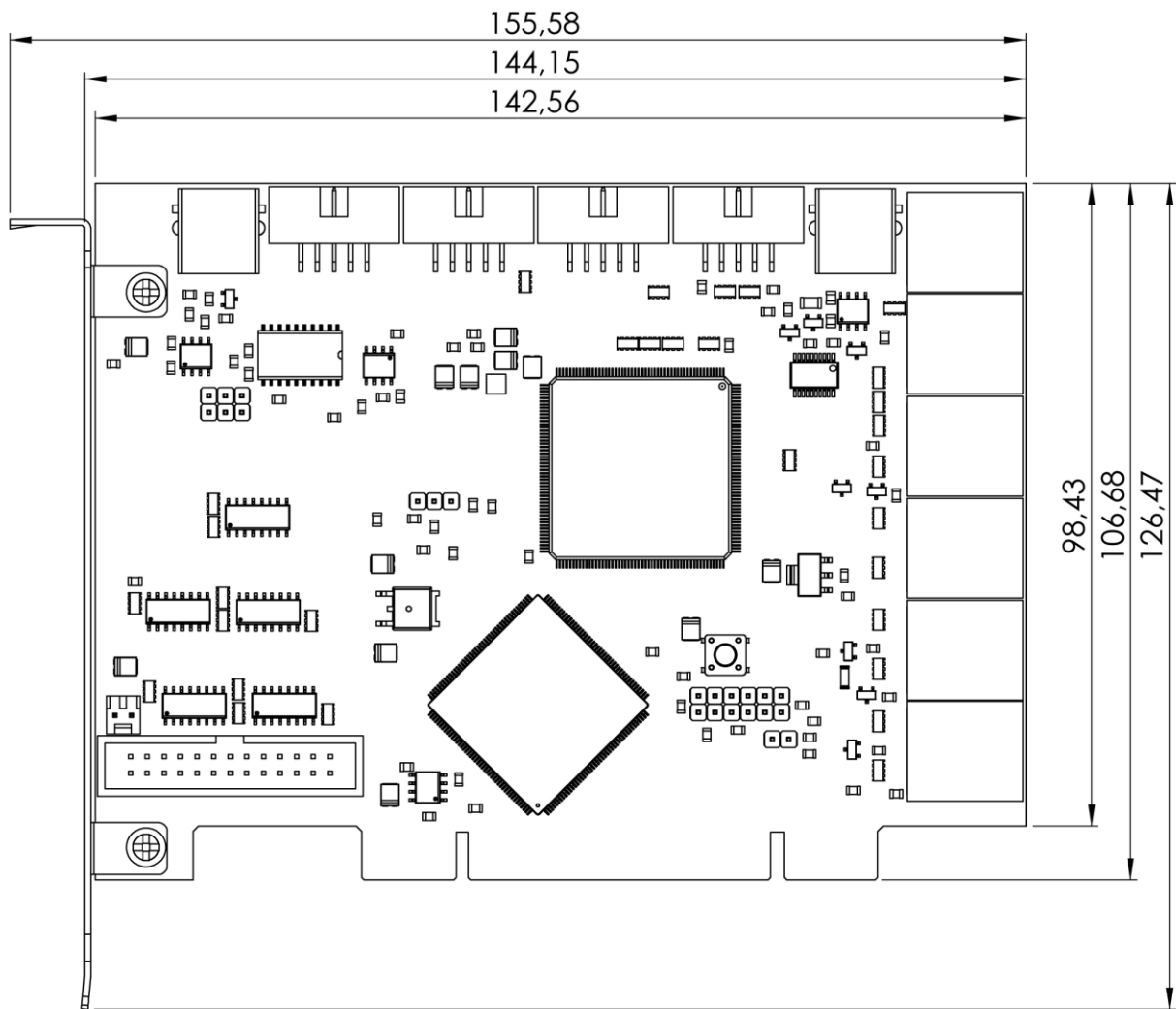
All dimensions are in mm. Not displayed dimensions are compliant with PCI specification and standard.

3. CONNECTING SERVO MODULES

This chapter is about how to connect servos to the PCI motion control card for driving a machine.

3.1. Axis interface modules

Small sized DIN rail mounted interface modules gives easy way of connecting different types of servo modules to the axis connectors. First in this chapter, seven different system configurations are presented that can be set up for evaluating typical applications. Then the detailed description of the four available interface modules is presented.

Axis interface modules: Differential line driver, Digital to analogue converter, Optical isolator, Encoder/ reference breakout

3.1.1. Typical servo configurations

For evaluating the appropriate servo-drive structure the modules have to be connected as the following block diagrams. All the modular cables (RJ50 and RJ45) have to be straight wired.

3.1.1.1. Analogue system with encoder feedback
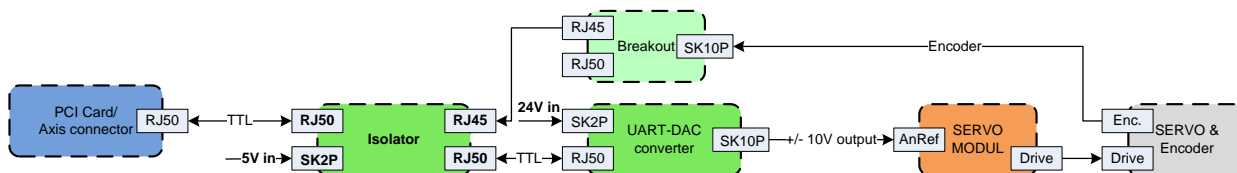
This configuration can be used for interfacing with the conventional analogue servo modules those have voltage level reference signal input.

3.1.1.2. Incremental digital system with encoder feedback and differential output
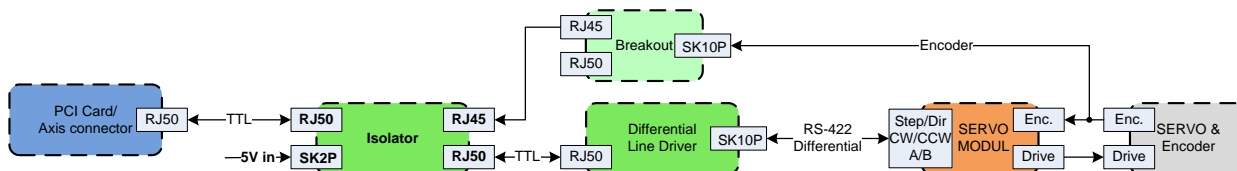
This configuration can be used for interfacing with three types of incremental servo modules with the following differential reference inputs: Step/Direction, Clockwise/Counter clockwise and Quadrant A/B. It is possible to read the servo position in the PC with the encoder feedback connection.

3.1.1.3. Incremental digital system with encoder feedback and TTL output
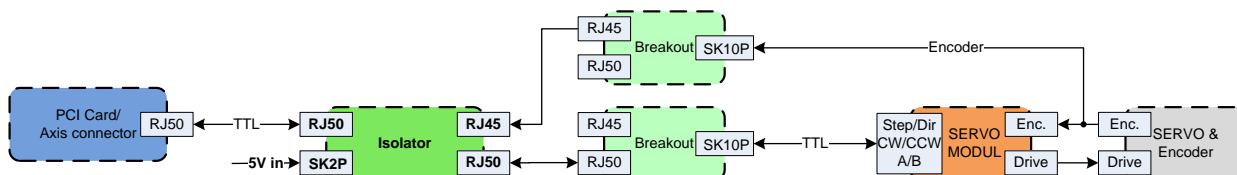
This configuration can be used for interfacing with three types of incremental servo modules with the following TTL level reference inputs: Step/Direction, Clockwise/Counter clockwise and Quadrant A/B. It is possible to read the servo position in the PC with the encoder feedback connection.

3.1.1.4. Incremental digital system with differential output
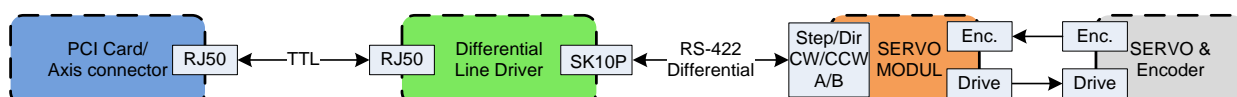
Same as configuration in chapter 3.1.1.2. but without isolation and encoder feedback. This configuration is recommended when the servo module has optically isolated inputs.

3.1.1.5. Incremental digital system with TTL output
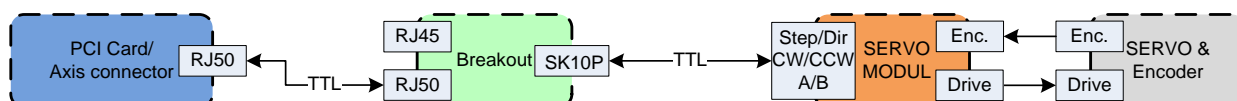
Same as configuration in chapter 3.1.1.3. but without isolation and encoder feedback.

3.1.1.6. Absolute digital (CAN based) system
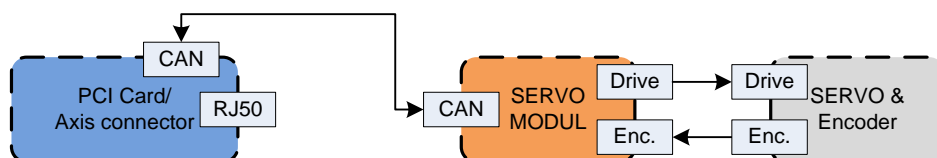
Servo modules connected along CAN bus interface.

3.1.1.7. Absolute digital (CAN based) system with conventional (A/B/I) encoder feedback
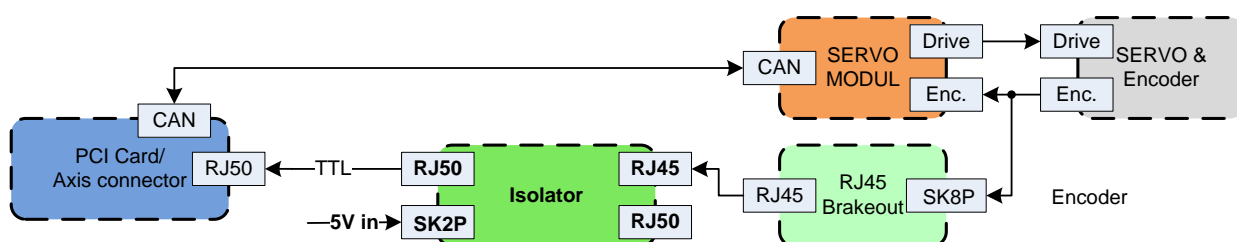
Servo modules connected along CAN bus interface, expanded with conventional encoder feedback along an optical isolator to the RJ50 axis connector.

3.1.2. AXIS – Optical Isolator

This module can be used for isolating all the signals referred to the PC ground, and interfacing with TTL or differential output encoders. The controller side of the module can be connected to the PCI card's Axis connector. The machine side reference output can be connected to an *AXIS – Breakout* module or to an *AXIS – Differential breakout* module, or can be hard wired directly to TTL inputs of a servo controller.
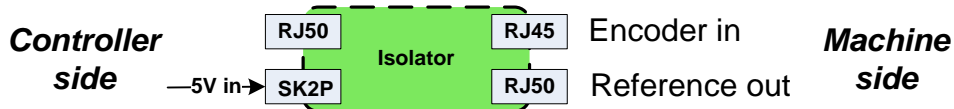
Block diagram of the optical isolator module connection

3.1.2.1. Pinout – Controller side
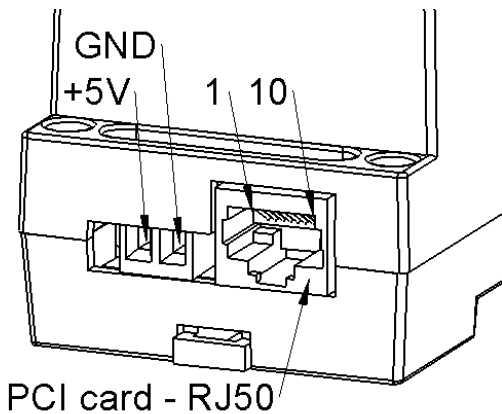
PIN	Controller side - RJ50
1	Encoder A
2	+5 Volt (PC)
3	Encoder B
4	Encoder Index
5	Fault
6	Power Enabled
7	Step/CCW/B
8	Direction/CW/A
9	Ground (PC)
10	DAC serial line

Pinout of PCI card (RJ50) connector and power input terminals

3.1.2.2. Pinout – Machine side
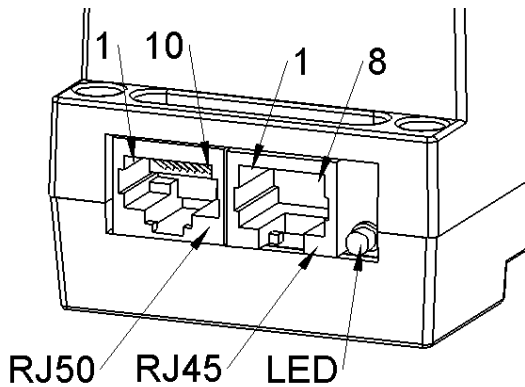
PIN	RJ50
1	not connected
2	+5 Volt (PC)
3	not connected
4	not connected
5	Fault
6	Power Enabled
7	Step/CCW/B
8	Direction/CW/A
9	Ground (PC)
10	DAC serial line

PIN	Encoder in - RJ45
1	GND
2	+5V
3	Encoder B-
4	Encoder I-
5	Encoder I+
6	Encoder B+
7	Encoder A-
8	Encoder A+

Pinout of reference output and encoder input connectors

3.1.2.3. Electrical characteristics

Output characteristics of pins on reference output (RJ50) connector:

All output pins are TTL compatible.

Maximum output high-level voltage:	**5**	Volts		no load	
Minimum output high-level voltage:	**2**	Volts	I_L	**7.5**	mA
Maximum output low-level voltage:	**0.1**	Volt			
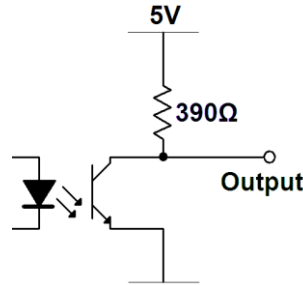
Equivalent circuit of output pins

Input characteristics of *fault input* pin on reference output (machine side RJ50) connector:

Minimum input high-level voltage:	**2**	Volts	I_F	**3.85**	mA
Maximum input low-level voltage:	**1**	Volts			
Absolute maximum input voltage:	**7**	Volts	I_F	**27**	mA
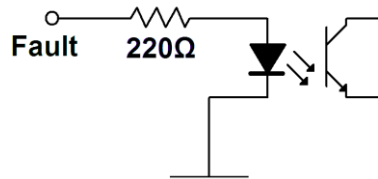
Fault signal input equivalent circuit

Input characteristics of pins on encoder input (RJ45) connector:

DC characteristics for **differential output encoder**:

Minimum high-level input voltage difference:	**0.2**	Volt
Minimum low-level input voltage difference:	**-0.2**	Volt
Maximum differential input voltage:	**±6**	Volts

DC characteristics for **TTL output** encoder: The negative differential inputs must be left open circuit. The outputs have to be connected to the positive inputs.

Minimum input high-level voltage:	**2**	Volts	I_H	**110**	µA
Maximum input low-level voltage:	**1.2**	Volts	I_L	**-110**	µA

Maximum frequency at symmetrical square signal:	**500**	kHz
Maximum load current of encoder power:	**500**	mA

Input DC characteristics of pins on *PCI card – RJ50* (controller side) connector:

Minimum input high-level voltage:	**2**	Volts	I_F	**3.85**	mA
Maximum input low-level voltage:	**1**	Volts			
Absolute maximum input voltage:	**7**	Volts	I_F	**27**	mA
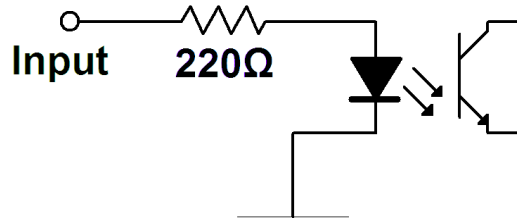
PCI card (RJ50) input equivalent circuit

Output DC characteristics of Fault output pin on *PCI card – RJ50* (controller side) connector:

Maximum output high-level voltage:	**5**	Volts		no load	
Minimum output high-level voltage:	**2**	Volts	I_L	**7.5**	mA
Maximum output low-level voltage:	**0.8**	Volt	I_L	**-360**	µA
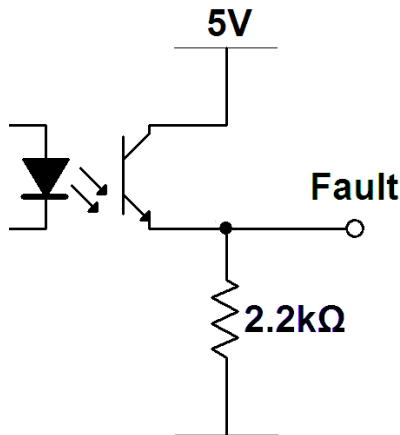
RJ50 to PCI card: Fault output equivalent circuit

Supply voltage:

Recommended supply voltage:	**5**	Volts
Absolute maximum supply voltage:	**5.5**	Volts
Absolute minimum supply voltage:	**4.5**	Volts
Maximum current consumption without encoder:	**120**	mA
Maximum current consumption with encoder:	**750**	mA

3.1.3. AXIS – DAC (Digital-to-Analogue Converter)

This module converts the digital signal of the PCI card to analogue output for driving a speed or current mode servo amplifier. The isolator module is recommended in most cases for using this module, see chapter 3.1.1.1. "Analogue system with encoder feedback".
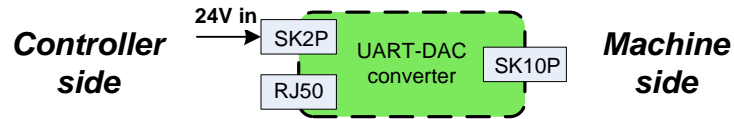
Block diagram of the digital to analogue converter module connection

The controller side RJ50 connector of this module has to be connected to the machine side of an **AXIS – Optical Isolator** module or directly to the PCI card (when servo amplifier is isolated). The machine side terminal output can be connected to an analogue type servo amplifier.

3.1.3.1. Controller side pinout
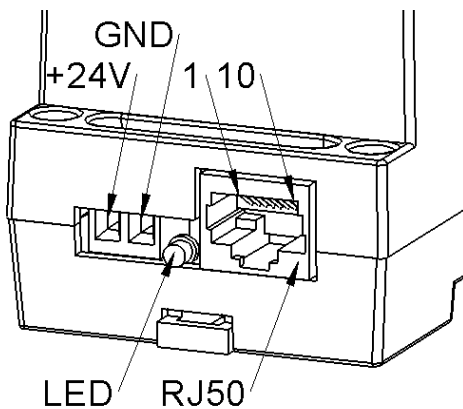
PIN	RJ50
1	DAC serial line −
2	+5 Volt (from isolator)
3	not connected
4	not connected
5	Fault
6	Power Enabled
7	not connected
8	not connected
9	GND
10	DAC serial line +

3.1.3.2. Machine side pinout
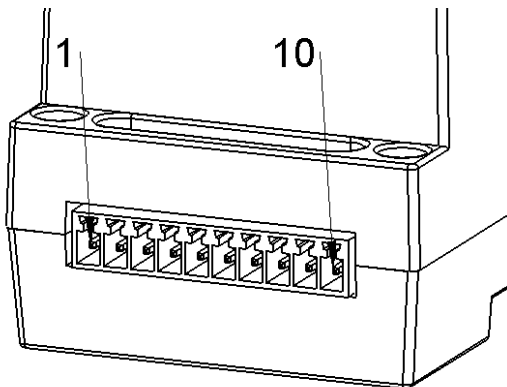
PIN	SK10P - Terminal
1	GND
2	Analogue output
3	not connected
4	not connected
5	24V out
6	GND
7	Enable - Collector
8	Enable - Emitter
9	Fault - Cathode
10	Fault - Anode

3.1.3.3. Electrical characteristics

24V external power supply input:

Minimum operating voltage:	**18**	Volts
Maximum operating voltage:	**26.5**	Volts

Controller side RJ50 connector:

Serial line in single line connection – only serial line+ connected (most cases):

Serial line+ minimum high level input voltage: **3** Volts

Serial line+ maximum low level input voltage: **2** Volts

Serial line in differential connection:

Minimum high level input voltage difference: **0.4** Volts

Maximum low level input voltage difference: **0.4** Volts

Fault signal – open emitter output signal:

Maximum high level output voltage: **5** Volts
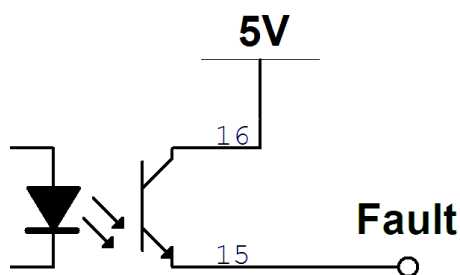
Equivalent circuit of fault signal output

Enabled input signal

Minimum high level input voltage: **2** Volts

Maximum low level input voltage: **0.8** Volts

Machine side terminal connector:

DAC output:

Maximum output voltage: **10** Volts

Minimum output voltage: **-10** Volts

Output voltage level accuracy: **±1** %

24V output:

Maximum output load current (fused): **200** mA

Enabled output signal:

Collector-emitter breakdown voltage:	**80**	Volts	I_C	**0.5**	mA
Emitter-collector breakdown voltage:	**7**	Volts	I_E	**0.1**	mA
Maximum collector dark current:	**0.1**	μA	V_{CE} **48**		Volts
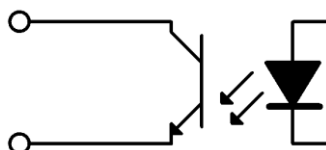
For more information please refer to Toshiba TLP291 optocoupler's datasheet.

Fault input signal - LED:

LED forward voltage: **1.15**   Volts   I_F   **10**   mA

Fault Anode
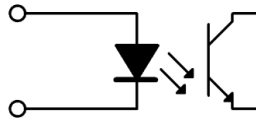
Fault Cathode

Equivalent circuit of fault input circuit

For more information please refer to Toshiba TLP291 optocoupler's datasheet.

3.1.3.4. Operating conditions

The following flowchart describes the internal logic and fault handling of the axis DAC module:

	Rootcause	**Workaround**
No 5V, CPU halt? → yes → LED: Off / Enable out: LOW / Output voltage: ?	There is no 5V from the PC side connector.	Check the RJ50 axis connection to the PC. Check the power supply of the optical isolator module is exists between the DAC module and the PCI card.
Watchdog reset ocurred? → yes → LED: Red & Green, Fast blink / Enable out: LOW / Output voltage: 0V	Unwonted CPU fault occurred.	Turn off enable signal, then turn on again. If the error still occurres, check 5V power supply or replace the DAC module.
24V low-voltage? → yes → LED: Red, Slow blink / Enable out: LOW / Output voltage: 0V	24V external supply voltage is under 15V.	The supply should be above 18V to recover the internal fault. Then turn off enable signal, then turn on again.
24V over-voltage? → yes → LED: Red, Fast blink / Enable out: LOW / Output voltage: 0V	24V external supply voltage is over 26.5V.	The supply should be under 26.5V to recover the internal fault. Then turn off enable signal, then turn on again.
UART error? → yes → LED: Red & Green, Slow blink / Enable out: LOW / Output voltage: 0V	No UART command frame received, or the frame is broken.	Check UART signal: Normally 22, but minimum 11 correct data frames have to be received within each millisecond. Then turn off enable signal, then turn on again to recover the fault state.
DAC voltage level error? → yes → LED: Green, Fast blink / Enable out: LOW / Output voltage: ?	DAC output voltage level feedback is out of the tolerance threshold.	Check that the output is not overloaded or shorcuted, also check for high capacitive loads if the error occurs during servo movement. Turn off enable signal, then turn on again. If the error still remains, replace the DAC module.
No internal DAC fault		
Servo fault signal? → yes → LED: Red, On / Enable out: LOW / Output voltage: 0V	The servo module sends fault signal.	Check the servo module.
No enabled signal? → yes → LED:Green, Slow blink / Enable out: LOW / Output voltage: 0V	The DAC is ready to operate.	Give enable signal for starting the motor.
LED: Green, On / Enable out: HIGH / Output voltage: commanded		

The following conditions have to be fulfilled in order for having the analogue voltage level and enabled signal at the output for driving the analogue servo module:

- The RJ50 connector is connected to the PC (PCI card or the **AXIS – Optical Isolator** module), and 5V supply is between 4.5...5.5 Volts.

- No watchdog error.

- External 24 Volt DC supply is connected and the voltage level is between 18...26.5 Volts.

- The valid digital reference signal comes on UART from the PC. (Normally 22, but minimum 11 correct data frames in each millisecond.)

- No DAC output voltage level fault

- No servo fault - The fault signal from the servo module is LOW

- Drive enabled - Enabled signal from the PC is HIGH

3.1.3.5. Fault conditions

- No 5V power supply from PC or from **AXIS – Optical Isolator** module:
    - CPU doesn't operate
    - LED is turned off
    - The fault signal from servo module is left LOW
    - The Enabled signal to the servo module is overdriven to LOW

- Watchdog reset occurred:
    - LED blinks fast, red and green (blinking period is 100 [ms])
    - The fault signal from servo module is overdriven to HIGH
    - The Enabled signal to the servo module is overdriven to LOW
    - The analogue output is equal to 0 Volt.
    - Reversed when enable signal from PC is turned off and then turned on again, or power reset occurred.
    - If the error still remains, replace the DAC module.

- 24V power supply is low (under 15 [V]):
    - LED blinks slow, red
    - The fault signal from servo module is overdriven to HIGH
    - The Enabled signal to the servo module is overdriven to LOW
    - The analogue output is equal to 0 Volt.
    - Reversed when 24V supply is over 18 Volts, and the enabled signal is turned off and then on.

- 24V power supply is low (over 26.5 [V]):
    - LED blinks fast, red
    - The fault signal from servo module is overdriven to HIGH
    - The Enabled signal to the servo module is overdriven to LOW

- The analogue output is equal to 0 Volt.

- Reversed when 24V supply is under 26.5 Volts, and the enabled signal is turned off and then on.

- No digital input or the UART frame is broken:

   - LED blinks slow, red and green

   - The fault signal from servo module is overdriven to HIGH

   - The Enabled signal to the servo module is overdriven to LOW

   - The analogue output is equal to 0 Volt.

   - Reversed when digital input signal on UART line becomes valid again and the enabled signal is turned off and then on.

- DAC output voltage level is not equal to the specified voltage (DAC error):

   - LED blinks fast, green

   - The fault signal from servo module is overdriven to HIGH

   - The Enabled signal to the servo module is overdriven to LOW

   - The analogue output can be anything due to a DAC fault.

   - Reversed when the DAC output voltage error is within the tolerance and the enabled signal is turned off and then on.

   - If the error still remains, replace the DAC module.

- Fault signal from servo module is HIGH:

   - LED on, red

   - The analogue output is equal to 0 Volt.

   - Reversed when fault signal becomes LOW again.

   - Does not affect the enabled signal.

- Enabled signal from PC is LOW:

   - LED blinks slow, green.

   - The analogue output is equal to 0 Volt.

   - Reversed when enabled signal becomes HIGH again.

3.1.4. AXIS – Differential breakout

This module can be used for interfacing with servo modules those have differential reference input.
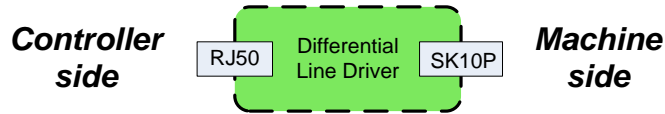
Block diagram of the differential line driver module connection

The controller side can be connected directly to the PCI card's axis connector if no need for optical isolation or encoder feedback to the controller. In any other cases the controller side should connected to the machine side RJ50 connector of an *AXIS – Optical Isolator* module.

3.1.4.1. Controller side pinout
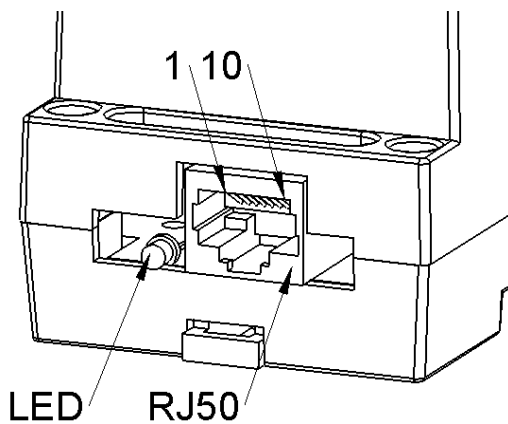
PIN	RJ50
1	nc.
2	+5 Volt (PC)
3	nc.
4	nc.
5	Fault
6	Power Enabled
7	Step/CCW/B
8	Direction/CW/A
9	Ground (PC)
10	DAC serial line

3.1.4.2. Machine side pinout
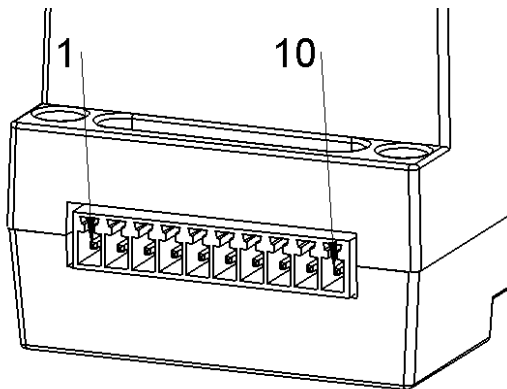
PIN	SK10P - Terminal
1	DAC serial line –
2	DAC serial line +
3	Direction/CW/A –
4	Direction/CW/A +
5	Step/CCW/B –
6	Step/CCW/B +
7	Enable - Collector
8	Enable - Emitter
9	Fault - Cathode
10	Fault - Anode

3.1.4.3. Electrical characteristics

Controller side RJ50 connector:

TTL inputs (pins: 6, 7, 8, 10):

Minimum high level input voltage: **2** Volts

Maximum low level input voltage: **0.8** Volts

Note: The Power Enabled signal (pin 6) is pulled down internally by 220 Ohm.

For more information please refer to AM26LS31 differential line driver's datasheet.

Machine side terminal connector:

Differential outputs:

Maximum low level output current:	**20**	mA			
Maximum high level output current:	**-20**	mA			

For more information please refer to AM26LS31 differential line driver's datasheet.

Enabled output signal:

Collector-emitter breakdown voltage:	**80**	Volts	I_C	**0.5**	mA
Emitter-collector breakdown voltage:	**7**	Volts	I_E	**0.1**	mA
Maximum collector dark current:	**0.1**	μA	V_{CE}	**48**	Volts
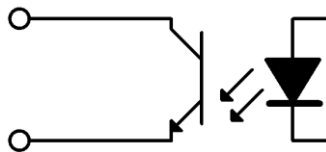
For more information please refer to Toshiba TLP291 optocoupler's datasheet.

Fault input signal - LED:

LED forward voltage:	**1.15**	Volts	I_F	**10**	mA
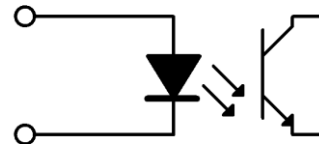
Equivalent circuit of fault input circuit

For more information please refer to Toshiba TLP291 optocoupler's datasheet.

3.1.5. AXIS – Breakout

This module can be used for wiring out the reference outputs or the encoder inputs modular contacts to a simple terminal connector. In case of encoder it can be useful for splitting the signals between servo module and encoder feedback to the PC.

3.1.5.1. Connectors
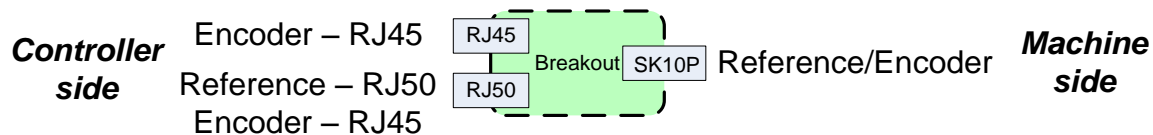
Block diagram of the breakout module connection

Controller side RJ50 & RJ45 modular connectors

The 10-pin RJ50 connector can host 8-pin RJ45 plug also. In case of splitting encoder signals, two RJ45 encoder cables can be used.
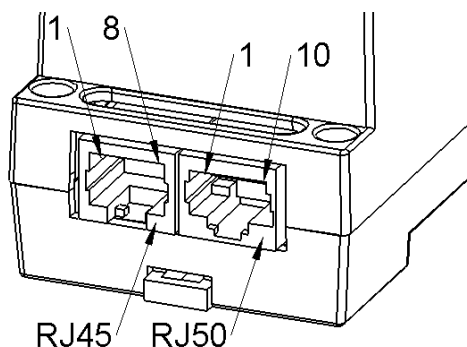
Pin numbering of RJ50 and RJ45 modular connectors

Controller side RJ50 – Reference or Encoder pinout:

PIN	Used for reference	Used for encoder
1	Encoder A	nc.
2	+5 Volt (PC)	GND
3	Encoder B	+5V
4	Encoder Index	Encoder B-
5	Fault	Encoder I-
6	Power Enabled	Encoder I+
7	Step/CCW/B	Encoder B+
8	Direction/CW/A	Encoder A-
9	Ground (PC)	Encoder A+
10	DAC serial line	nc.

Controller side RJ45 – Encoder pinout:

The power supply of the encoder is not connected on the secondary (RJ45) connector for ensuring that the encoder sensor gets the supply voltage from one source only.

PIN	Used for encoder
1	GND
2	nc.
3	Encoder B-
4	Encoder I-
5	Encoder I+
6	Encoder B+
7	Encoder A-
8	Encoder A+

Machine side SK10P - Terminal connector pinout:
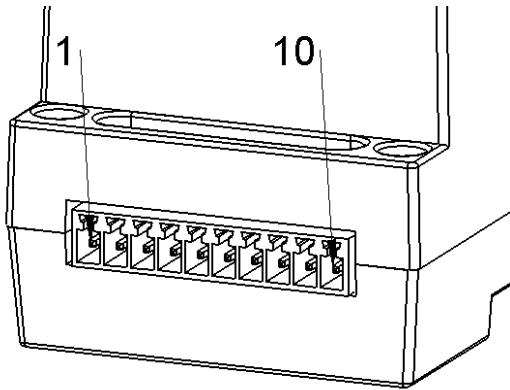
PIN	Used for reference	Used for encoder
1	DAC serial line	nc.
2	Ground (PC)	Encoder A+
3	Direction/CW/A	Encoder A-
4	Step/CCW/B	Encoder B+
5	Power Enabled	Encoder I+
6	Fault	Encoder I-
7	Encoder Index	Encoder B-
8	Encoder B	+5V
9	+5 Volt (PC)	GND
10	Encoder A	nc.

4. HALL SETTINGS

In this chapter, all the system relevant hall settings can be found, except for modules on the RS485 bus. Those can be found in the next chapter for each module.

All the pins and parameters of this chapter are updated by the following two functions:

```
gm.<nr. of card>.read
gm.<nr. of card>.write
```

Both should be added to servo thread in this given order (first read then write) for most applications.

4.1. Encoder

The motion control card has six encoder modules. Each encoder module has three channels:

- Channel-A
- Channel-B
- Channel-I (index).

It is able to count quadrature encoder signals or step/dir signals. Each encoder module is connected to the inputs of the corresponding RJ50 axis connector.

Every encoder pin and parameter name begins as follows:

```
gm.<nr. of card>.encoder.<nr of axis>
```
,where <nr of axis> is form 0 to 5.

For example: `gm.0.encoder.0.position` ► refers to the position of encoder module of axis 0.

The PCI card counts the encoder signal independently from LinuxCNC. Hal pins are updated by the `gm.<nr of card>.read` function.

4.1.1. Pins:

`.reset` (bit, In) ► When True, resets counts and position to zero.

`.rawcounts` (s32, Out) ► The raw count is the counts, but unaffected by reset or the index pulse.

`.counts` (s32, Out) ► Position in encoder counts.

`.position` (float, Out) ► Position in scaled units (=*.counts/.position-scale*).

`.index-enabled` (bit, IO) ► When True, *counts* and *position* are rounded or reset (depends on *index-mode*) on next rising edge of channel-I. Every time position is reset because of Index, index-enabled pin is set to 0 and remain 0 until connected hal pin does not set it.

`.velocity` (float, Out) ► Velocity in scaled units per second. GM encoder uses high frequency hardware timer to measure time between encoder pulses in order to calculate velocity. It greatly reduces quantization noise as compared to simply differentiating the position output. When the measured velocity is below *min-velocity-estimate*, the velocity output is 0.

4.1.2. Parameters:

`.counter-mode` (bit, R/W) ► When True, the counter counts each rising edge of the channel-A input to the direction determined by channel-B. This is useful for counting the output of a single channel (non-quadrature) or step/dir signal sensor. When false, it counts in quadrature mode.

`.index-mode` (bit, R/W) ► When True and *.index-enabled* is also true, *.counts* and *.position* are rounded (based on *.counts-per-rev*) at rising edge of channel-I. This is useful to correct few pulses error caused by noise. In round mode, it is essential to set *.counts-per-rev* parameter correctly. When `.index-mode` is False and `.index-enabled` is true, `.counts` and `.position` are reset at channel-I pulse.

`.counts-per-rev` (u32, R/W) ► Determine how many counts are between two index pulses. It is used only in round mode, so when both `.index-enabled` and `.index-mode` parameters are True. GM encoder process encoder signal in 4x mode, so for example in case of a 500 CPR encoder it should be set to 2000. This parameter can be easily measured by setting `.index-enabled` True and `.index-mode` False (so that `.counts` resets at channel-I pulse), than move axis by hand and see the maximum magnitude of `.counts` pin in halmeter.

`.index-invert` (bit, R/W) ► When True, channel-I event (reset or round) occur on falling edge of channel-I signal, otherwise on rising edge.

`.min-speed-estimate` (float, R/W) ► Determine the minimum measured velocity magnitude at which `.velocity` will be set as nonzero. Setting this parameter too low will cause it to take a long time for velocity to go to zero after encoder pulses have stopped arriving.

`.position-scale` (float, R/W) ► Scale in counts per length unit. `.position=.counts/.position-scale`. For example, if position-scale is 2000, then 1000 counts of the encoder will produce a position of 0.5 units.

4.1.3. HAL example

Setting encoder module of axis 0 to receive 500 CPR quadrature encoder signal and use reset to round position:

```
setp gm.0.encoder.0.counter-mode 0        # 0: quad, 1: stepDir

setp gm.0.encoder.0.index-mode 1          # 0: reset pos at index, 1:round pos at
                                                index

setp gm.0.encoder.0.counts-per-rev 2000   # GM process encoder in 4x mode,
                                                4x500=2000

setp gm.0.encoder.0.index-invert 0

setp gm.0.encoder.0.min-speed-estimate 0.1  #  in position unit/s

  setp gm.0.encoder.0.position-scale 20000   #  in case of 10 encoder rev
                                                cause the machine to move
                                                one position unit (10x2000)
```

Connect encoder position to LinuxCNC position feedback:

```
net Xpos-fb gm.0.encoder.0.position => axis.0.motor-pos-fb
```

4.2. Stepgen module

The motion control card has six stepgen modules, one for each axis. Each module has two output signals. It can produce Step/Direction, Up/Down or Quadrature (A/B) pulses. Each stepgen module is connected to the pins of the corresponding RJ50 axis connector.

Every stepgen pin and parameter name begins as follows:

`gm.<nr. of card>.stepgen.<nr of axis>` ,where nr of axis is form 0 to 5.

For example: `gm.0.stepgen.0.position-cmd`   ► refers to the position command of stepgen module of axis 0 on card 0.

The PCI card generates step pulses independently from LinuxCNC. Hal pins are updated by `gm.<nr of card>.write` function.

4.2.1. Pins:

`.enable`	(bit, In)	► Stepgen produces pulses only when this pin is true.
`.count-fb`	(s32, Out)	► Position feedback in counts unit.
`.position-fb`	(float, Out)	► Position feedback in position unit.
`.position-cmd`	(float, In)	► Commanded position in position units. Used in position mode only.
`.velocity-cmd`	(float, In)	► Commanded velocity in position units per second. Used in velocity mode only.

4.2.2. Parameters:

`.step-type`	(u32, R/W)	► When 0, module produces Step/Dir signal. When 1, it produces Up/Down step signals. And when it is 2, it produces quadrature output signals.
`.control-type`	(bit, R/W)	► When True, `.velocity-cmd` is used as reference and velocity control calculate pulse rate output. When False, `.position-cmd` is used as reference and position control calculate pulse rate output.
`.invert-step1`	(bit, R/W)	► Invert the output of channel 1 (Step signal in StepDir mode)
`.invert-step2`	(bit, R/W)	► Invert the output of channel 2 (Dir signal in StepDir mode)
`.maxvel`	(float, R/W)	► Maximum velocity in position units per second. If it is set to 0.0, `.maxvel` parameter is ignored.
`.maxaccel`	(float, R/W)	► Maximum acceleration in position units per second squared. If it is set to 0.0, `.maxaccel` parameter is ignored.
`.position-scale`	(float, R/W)	► Scale in steps per length unit.

`.position-fb=.count-fb/.position-scale`. For example, if position-scale is 1000, then 1 position unit command produces 1000 step pulses.

`.steplen`	(u32, R/W)	►	Length of step pulse in nano-seconds.
`.stepspace`	(u32, R/W)	►	Minimum time between two step pulses in nano-seconds.
`.dirdelay`	(u32, R/W)	►	Minimum time between step pulse and direction change in nano-seconds.

For evaluating the appropriate values see the timing diagrams below:
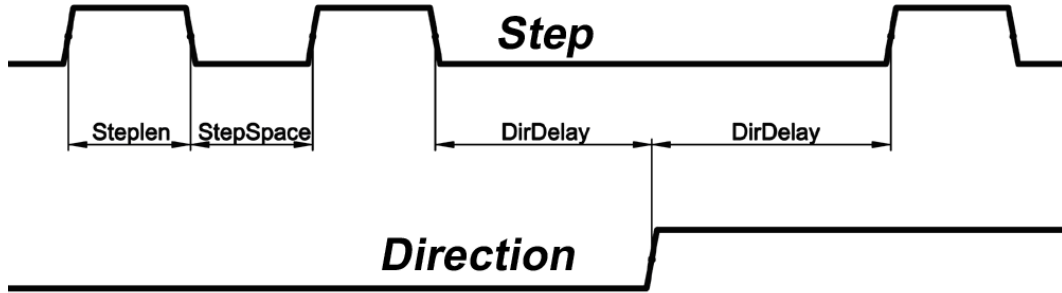
Step/Dir type reference
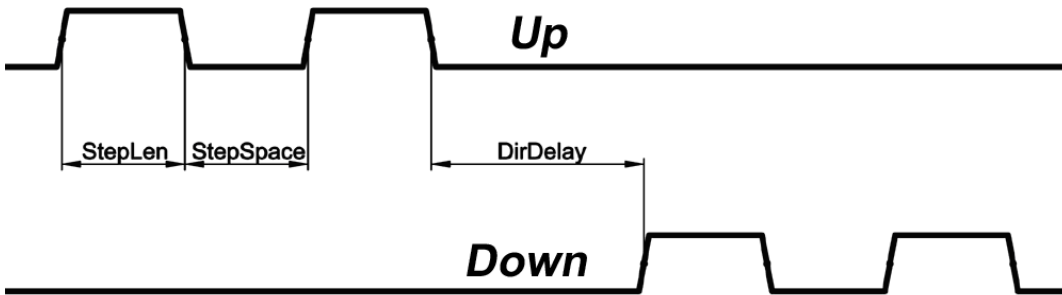
Up/Down count (CW/CCW) reference
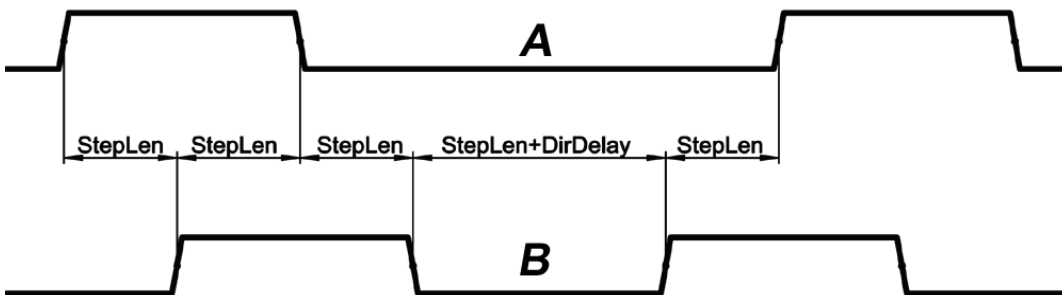
Quadrant (A/B) type reference

4.2.3. HAL example:

Setting stepgen module of axis 0 to generate 1000 step pulse per position unit:

```
setp gm.0.stepgen.0.step-type 0          # 0:stepDir,1:UpDown,2:Quad
setp gm.0.stepgen.0.control-type 0       # 0:Pos. control,1:Vel. Control
setp gm.0.stepgen.0.invert-step1 0
setp gm.0.stepgen.0.invert-step2 0
setp gm.0.stepgen.0.maxvel 0             # do not set maxvel for step
                                         generator, let interpolator control it.
setp gm.0.stepgen.0.maxaccel 0           # do not set max acceleration for step
                                         generator, let interpolator control it.
setp gm.0.stepgen.0.position-scale 1000  # 1000 step/position unit
setp gm.0.stepgen.0.steplen 1000         # 1000 ns = 1 us
setp gm.0.stepgen.0.stepspace1000        # 1000 ns = 1 us
setp gm.0.stepgen.0.dirdelay 2000        # 2000 ns = 2 us
```

Connect stepgen to axis 0 position reference and enable pins:

```
net Xpos-cmd axis.0.motor-pos-cmd => gm.0.stepgen.0.position-cmd
net Xen axis.0.amp-enable-out => gm.0.stepgen.0.enable
```

4.3. Axis DAC (digital-to-analogue converter)

The motion control card has six serial axis DAC driver modules, one for each axis. Each module is connected to the pin of the corresponding RJ50 axis connector.

Every axis DAC pin and parameter name begins as follows:

`gm.<nr. of card>.dac.<nr of axis>`   ,where nr of axis is form 0 to 5.

For example: `gm.0.dac.0.value`   ► refers to the output voltage of DAC module of axis 0.

Hal pins are updated by `gm.<nr of card>.write` function.

4.3.1. Pins:

`.enable`	(bit, In)	► Enable DAC output. When enable is false, DAC output is 0.0 V.
`.value`	(float, In)	► Value of DAC output in Volts.

4.3.2. Parameters:

`.offset`	(float, R/W)	► Offset is added to the value before the hardware is updated
`.high-limit`	(float, R/W)	► Maximum output voltage of the hardware in volts.

`.low-limit`   (float, R/W)   ► Minimum output voltage of the hardware in volts.

`.invert-serial`   (float, R/W)   ► PCI card is communicating with DAC hardware via fast serial communication to highly reduce time delay compared to PWM. DAC module is recommended to be isolated which is negating serial communication line. In case of isolation, leave this parameter to default (0), while in case of none-isolation, set this parameter to 1.

4.4. Enable and Fault signals

The PCI motion control card has one enable output and one fault input HAL pins, both are connected to each RJ50 axis connector and to the CAN connector.

Hal pins are updated by `gm.<nr of card>.read` function.

4.4.1. Pins:

`gm.<nr of card>.power-enable`   (bit, In)   ►   If this pin is True,

- and Watch Dog Timer is not expired

- and there is no power fault

Then power enable pins of axis- and CAN connectors are set to high, otherwise set to low.

`gm.<nr of card>.power-fault`   (bit, Out)   ► Power fault input

4.5. Watchdog timer

Watchdog timer resets at `gm.<nr of card>.read` function.

4.5.1. Pins:

`gm.<nr of card>.watchdog-expired`   (bit, Out)   ► Indicates that watchdog timer is expired.

Watchdog timer overrun causes the set of power-enable to low in hardware.

4.5.2. Parameters:

`gm.<nr of card>.watchdog-enable`   (bit, R/W)   ► Enable watchdog timer.

It is strongly recommended to enable watchdog timer, because it can disables all the servo amplifiers by pulling down all enable signal in case of PC error.

`gm.<nr of card>.watchdog-timeout-ns` (u32, R/W) ► Time interval in within the `gm.<nr of card>.read` function must be executed. The *gm.<nr of card>.read* is typically added to servo-thread, so watch timeout is typically set to 3 times of the servo period.

4.6. GM-CAN

The motion control card has CAN module to drive CAN servo amplifiers. Implementation of higher level protocols like CANopen is further development. Currently GM produced power amplifiers has upper level driver which export pins and parameters to HAL. They receive position reference and provide encoder feedback via CAN bus.

Every CAN pin and parameter name begins as follows:

`gm.<nr. of card>.can-gm.<nr of axis>`   ,where <nr of axis> is form 0 to 5.

For example: `gm.0.can-gm.0.position` ► refers to the output position of axis 0 in position units.

Hal pins are updated by `gm.<nr of card>.write` function.

4.6.1. Pins:

`.enable`	(bit, In)	► Enable sending position references
`.position-cmd`	(float, In)	► Commanded position in position units.
`.position-fb`	(float, Out)	► Feed back position in position units.

4.6.2. Parameters:

`.position-scale`	(float, R/W)	► Scale in per length unit.

`.position-fb=.encoder-counts/.position-scale`. For example, if position-scale is 1000, then 2000 encoder pulses on GM power amplifier produces 2 on `position-fb` pin.

4.7. Home and Limit switches

The PCI motion control card has two limit- and one homing switch input for each axis. All the names of these pins begin as follows:

`gm.<nr. of card>.axis.<nr of axis>` ,where nr of axis is form 0 to 5.

For example: `gm.0.axis.0.home-sw-in`   indicates the state of the axis 0 home switch.

Hal pins are updated by `gm.<nr of card>.read` function.

4.7.1. Pins:

`.home-sw-in`	(bit, Out)	► Home switch input
`.home-sw-in-not`	(bit, Out)	► Negated home switch input
`.neg-lim-sw-in`	(bit, Out)	► Negative limit switch input
`.neg-lim-sw-in-not`	(bit, Out)	► Negated negative limit switch input
`.pos-lim-sw-in`	(bit, Out)	► Positive limit switch input
`.pos-lim-sw-in-not`	(bit, Out)	► Negated positive limit switch input

4.8. Emergency stop input signals

In addition to home and limit switches, there are two emergency stop (E-Stop) inputs to detect if E-Stop button is pressed.

4.8.1. Pins:

`gm.0.estop.0.in`	(bit, Out)	► Estop 0 input
`gm.0.estop.0.in-not`	(bit, Out)	► Negated Estop 0 input
`gm.0.estop.1.in`	(bit, Out)	► Estop 1 input
`gm.0.estop.1.in-not`	(bit, Out)	► Negated Estop 1 input

4.9. General purpose I/O

GM 6 axis motion control card has 4 general purpose I/O (GPIO) connectors, with eight configurable I/O on each.

Every GPIO pin and parameter name begins as follows:

`gm.<nr. of card>.gpio.<nr of gpio con>` ,where `<nr of gpio con>` is form 0 to 3.

For example: `gm.0.gpio.0.in-0` indicates the state of the first pin of the first GPIO connector on the PCI card.

Hal pins are updated by `gm.<nr of card>.read` function.

4.9.1. Pins:

`.in-<0-7>`	(bit, Out)	► Input pin
`.in-not-<0-7>`	(bit, Out)	► Negated input pin
`.out-<0-7>`	(bit, In)	► Output pin. Used only when GPIO is set to output.

4.9.2. Parameters:

`.is-out-<0-7>`	(bit, R/W)	► When True, the corresponding GPIO is set to totem-pole output, other wise set to high impedance input.
`.invert-out-<0-7>`	(bit, R/W)	► When True, pin value will be inverted. Used when pin is configured as output.

5. RS485 MODULES

These modules were developed for expanding the I/O and function capability along an RS485 line of the PCI motion control card.

5.1. Available module types

8-channel relay output module

The relay output module gives eight NO-NC relay output on a three pole terminal connector for each channel.

8-channel relay output module

8-channel digital input module

The digital input module gives eight optical isolated digital input pins.

8-channel digital input module

8 channel ADC and 4-channel DAC module

The ADC and DAC module gives four digital-to-analogue converter outputs and eight analogue-to-digital inputs. This module is also optically isolated from the PCI card.

8 channel ADC and 4-channel DAC module

Teach Pendant module

The teach pendant module gives 8 digital input channel for pushbuttons, 6 ADC input channel for joystick or potentiometers and 1 encoder input for a handwheel.
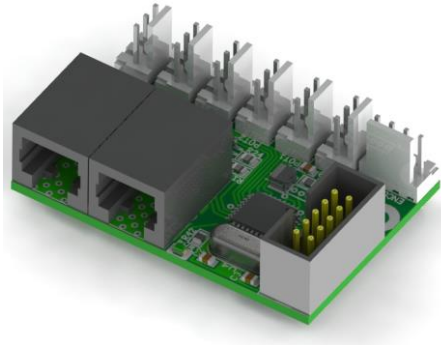
Teach Pendant module

5.2. Automatic node recognizing

Each node connected to the bus was recognized by the PCI card automatically. During starting LinuxCNC, the driver export pins and parameters of all available modules automatically.

5.3. Fault handling

If a module does not answer regularly the PCI card drops down the module.

If a module with output don't gets data with correct CRC regularly, the module switch to error sate (green LED blinking), and turns all outputs to error sate.

5.4. System description

5.4.1. Powering the nodes

Each module is electronically isolated from the bus, hence they have a bus powered side, and a field powered side.
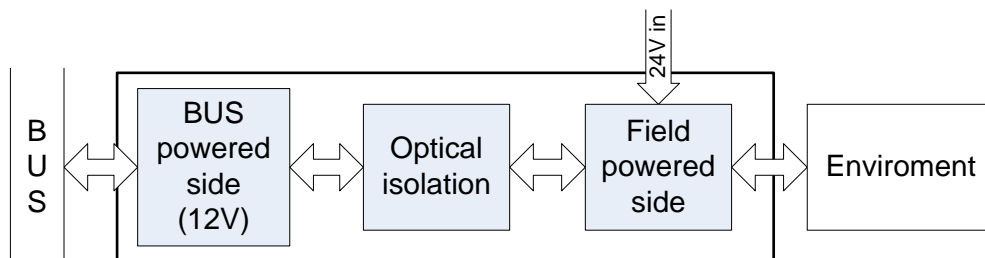
5.5. General bus settings
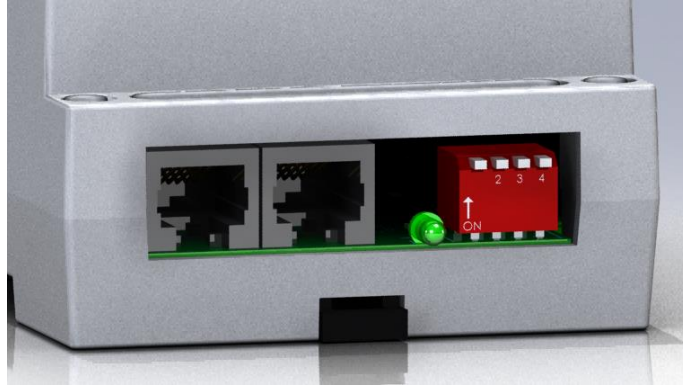
5.5.1. Connecting the nodes

The modules on the bus have to be connected in serial topology, with termination resistors on the end. The start of the topology is the PCI card, and the end is the last module.
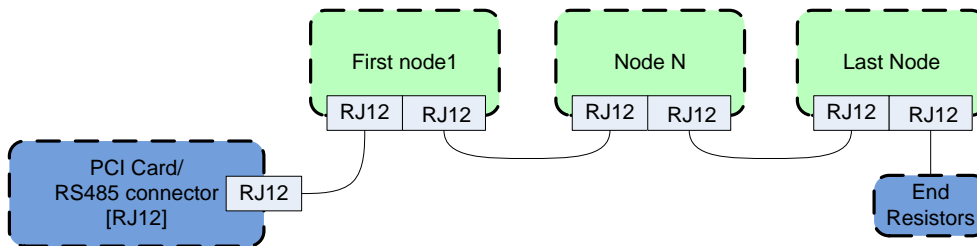
5.5.2. Addressing

Each node on the bus has a 4 bit unique address that can be set with the red DIP switch. Except for the Teach Pendant module, which has a pre programmed 00 ID. Upon request it is possible to order module with pre-programmed different ID.

Address	Switch setting 1234
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

5.5.3. Status LED

The green LED indicates the status of the module:

- Blink, when the module is only powered, but not jet identified, or when module is dropped down.

- Off, during identification (computer is on, but LinuxCNC not started)

- On, when it communicates continuously.

5.6. Modules

5.6.1. Relay output module

5.6.1.1. Block diagram
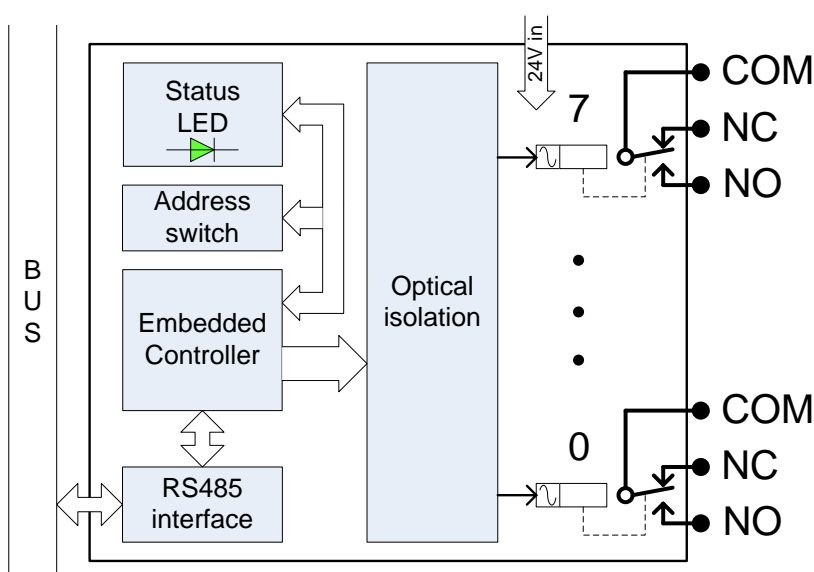
5.6.1.2. Electrical characteristics

Power:

- Bus voltage:.. 12 V

- Maximum bus power consumption:................................ 150 mA

- Field power voltage:... 24 V

- Maximum field power consumption (all relays on):........ 270 mA

Insulation:

- Optical isolation voltage: .... 2500 V_{RMS}

Relay characteristics:

- Maximum switching current: .... 10 A

- Maximum switching AC voltage: .... 250 V

- Maximum switching DC voltage: .... 30 V

- Dielectric strength: .... 5000 V

5.6.1.3. Connection
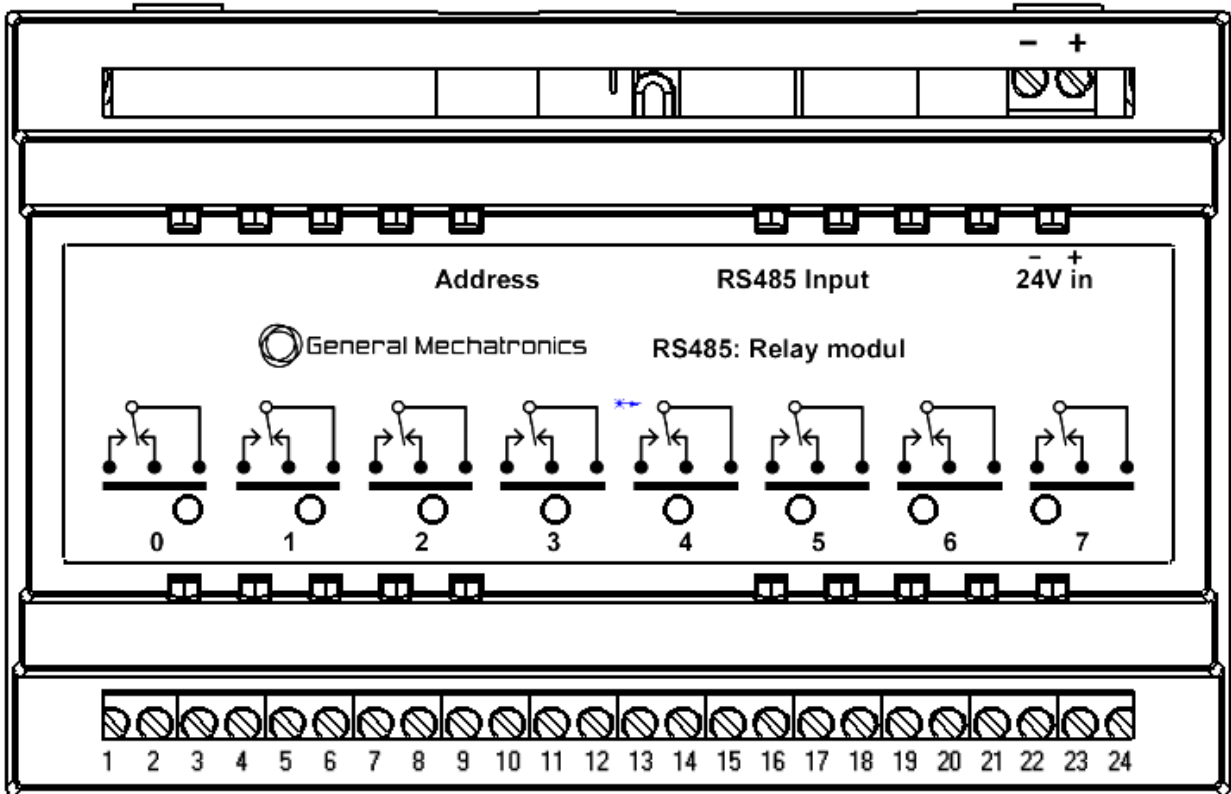
Numbering of output terminal connector and 24 input
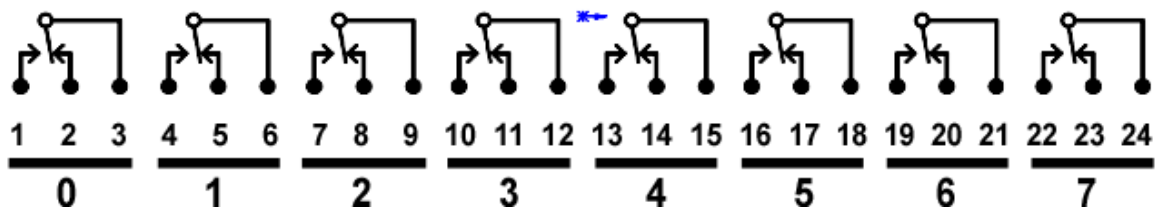
Output connection diagram

1	2	3	4	5	6
0-NO	0-NC	0-COM	1-NO	1-NC	1-COM

7	8	9	10	11	12
2-NO	2-NC	2-COM	3-NO	3-NC	3-COM

13	14	15	16	17	18
4-NO	4-NC	4-COM	5-NO	5-NC	5-COM

19	20	21	22	23	24
6-NO	6-NC	6-COM	7-NO	7-NC	7-COM

Pin assignment table: NO: Normally Open, NC: Normally Closed, COM: Common

5.6.1.4. Error state

If a bus error occurs, the module will switch to error state (green LED blinking). And turn off all output relays.

5.6.1.5. HAL configuration

All the pins and parameters are updated by the following function:

`gm.<nr. of card>.rs485`

It should be added to servo thread or other thread with larger period to avoid CPU overload.

Every RS485 module pin and parameter name begins as follows:

`gm.<nr. of card>.rs485.<modul ID>`   ,where <modul ID> is form 00 to 15.

Pins:

`.relay-<0-7>`   (bit, Out)   ► Output pin for relay

Parameters:

`.invert-relay-<0-7>`   (bit, R/W)   ► Negate relay output pin

HAL example:

`gm.0.rs485.0.relay-0` # First relay of the node.

# `gm.0`	Means the first PCI motion control card (PCI card address = 0)
# `.rs485.0`	Select node with address 0 on the RS485 bus
# `.relay-0`	Select the first relay

5.6.2. Digital input module

5.6.2.1. Block diagram
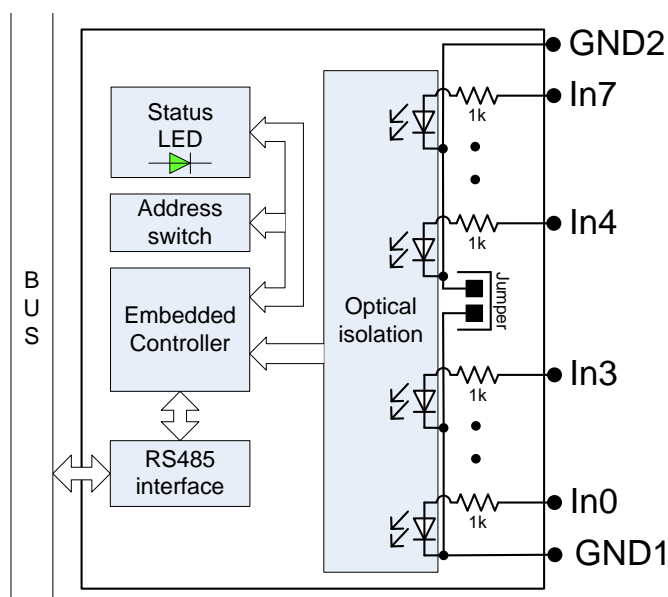
5.6.2.2. Electrical characteristics

Power:

- Bus voltage:.. 12 V
- Bus power consumption:.. 100 mA

Insulation:

- Optical isolation voltage: .... 2500 V_{RMS}

Input characteristics:
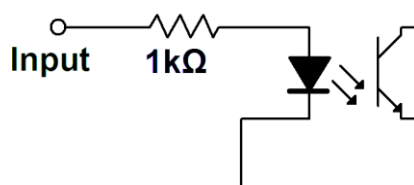
Equivalent circuit of digital input lines

Absolute maximum ratings

Maximum input voltage:	**30**	Volts
Minimum input voltage:	**-4**	Volts.
Maximum input current	**30**	mA

Logic levels

Minimum high-level input voltage:	5	Volts
Maximum low-level input voltage:	**0.6**	Volts

For more information please refer to Toshiba TLP291 optocoupler's datasheet.

5.6.2.3. Connection
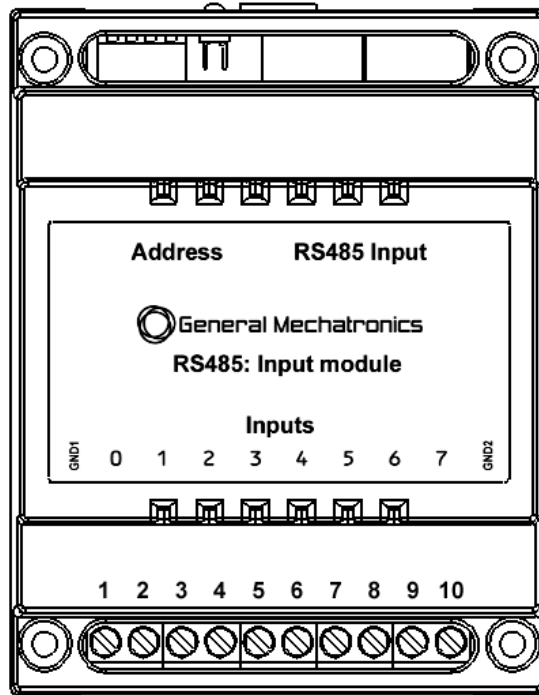
Numbering of input terminal connector

1	2	3	4	5	6	7	8	9	10
GND1	In0	In1	In2	In3	In4	In5	In6	In7	GND2

Pin assignment table

5.6.2.4. LinuxCNC HAL configuration

All the pins and parameters are updated by the following function:

`gm.<nr. of card>.rs485`

It should be added to servo thread or other thread with larger period to avoid CPU overload.

Every RS485 module pin and parameter name begins as follows:

`gm.<nr. of card>.rs485.<modul ID>`   ,where <modul ID> is form 00 to 15.

Pins:

`.in-<0-7>`	(bit, Out)	► Input pin
`.in-not-<0-7>`	(bit, Out)	► Negated input pin

HAL example:

`gm.0.rs485.0.in-0`   # First input of the node.

# `gm.0`	Means the first PCI motion control card (PCI card address = 0)
# `.rs485.0`	Select node with address 0 on the RS485 bus
# `.in-0`	Select the first digital input module

5.6.3. ADC & DAC module

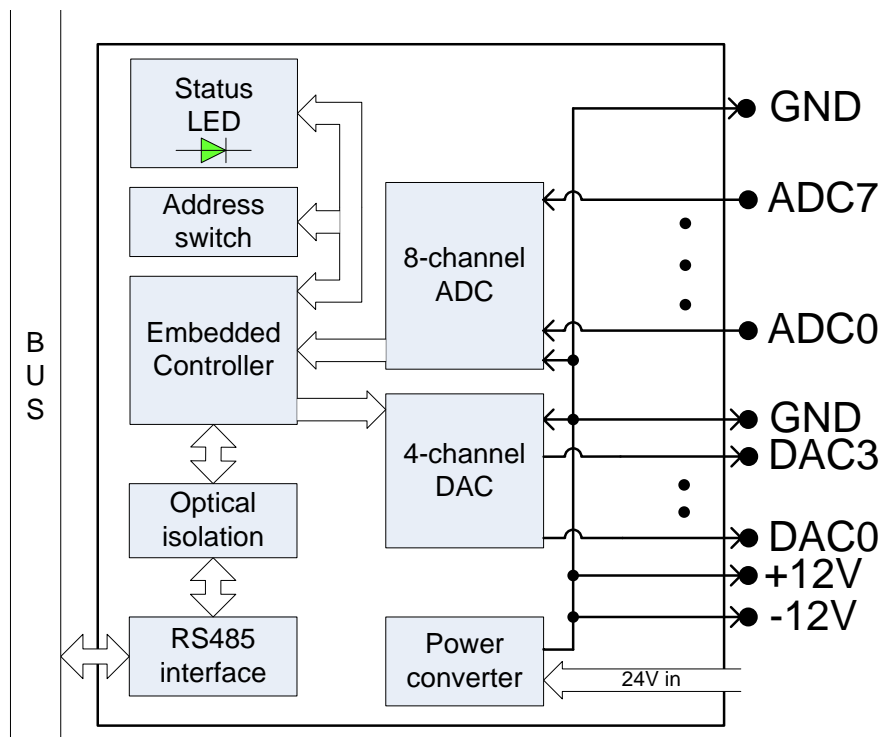### 5.6.3.1. Block diagram

5.6.3.2. Electrical characteristics

Power:

- Bus voltage:.. 12 V
- Bus power consumption:.. 100 mA
- Field power voltage:.. 24 V
- Maximum field power consumption: .... 500 mA

Insulation:

- Optical isolation voltage: .... 2500 V_{RMS}

AD converter:

- Input voltage range:...±5 V
- Input resistance:.. 820 kΩ
- Input capacitance:... 2 nF

DA converter:

- Output voltage range:...±10 V
- Maximum output current:... 20 mA

5.6.3.3. Connection
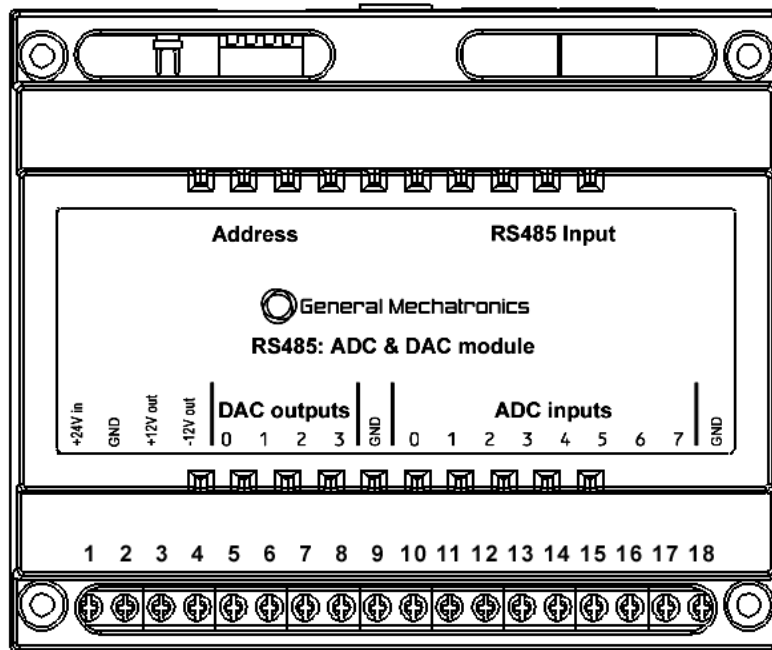
Numbering of the terminal connector

1	2	3	4	5	6	7	8	9
24V in	GND	-12V	+12V	DAC-0	DAC-1	DAC-2	DAC-3	GND

10	11	12	13	14	15	16	17	18
ADC-0	ADC-1	ADC-2	ADC-3	ADC-4	ADC-5	ADC-6	ADC-7	GND

Pin assignment table

5.6.3.4. LinuxCNC HAL configuration

All the pins and parameters are updated by the following function:

`gm.<nr. of card>.rs485`

It should be added to servo thread or other thread with larger period to avoid CPU overload.

Every RS485 module pin and parameter name begins as follows:

`gm.<nr. of card>.rs485.<modul ID>` ,where <modul ID> is form 00 to 15.

Pins:

|		
`.in-<0-7>`	(bit, Out)	► Input pin
`.in-not-<0-7>`	(bit, Out)	► Negated input pin
`.adc-<0-7>`	(float, Out)	► Value of ADC input in Volts.
`.dac-enable-<0-3>`	(bit, In)	► Enable DAC output. When *enable* is false DAC output is set to 0.0 V.
`.dac-<0-3>`	(float, In)	► Value of DAC output in Volts.

Parameters:

`.adc-scale-<0-7>`	(float, R/W)	► The input voltage will be multiplied by scale before being output to .adc- pin.
`.adc-offset-<0-7>`	(float, R/W)	► Offset is subtracted from the hardware input voltage after the scale multiplier has been applied.
`.dac-offset-<0-3>`	(float, R/W)	► Offset is added to the value before the hardware is updated.
`.dac-high-limit-<0-3>`	(float, R/W)	► Maximum output voltage of the hardware in volts.
`.dac-low-limit-<0-3>`	(float, R/W)	► Minimum output voltage of the hardware in volts.

HAL example:

`gm.0.rs485.0.adc-0`   # First analogue channel of the node.

  `#gm.0`   Means the first PCI motion control card (PCI card address = 0)

  `#.rs485.0`   Select node with address 0 on the RS485 bus

  `#.adc-0`   Select the first analogue input of the module

5.6.4. Teach pendant module
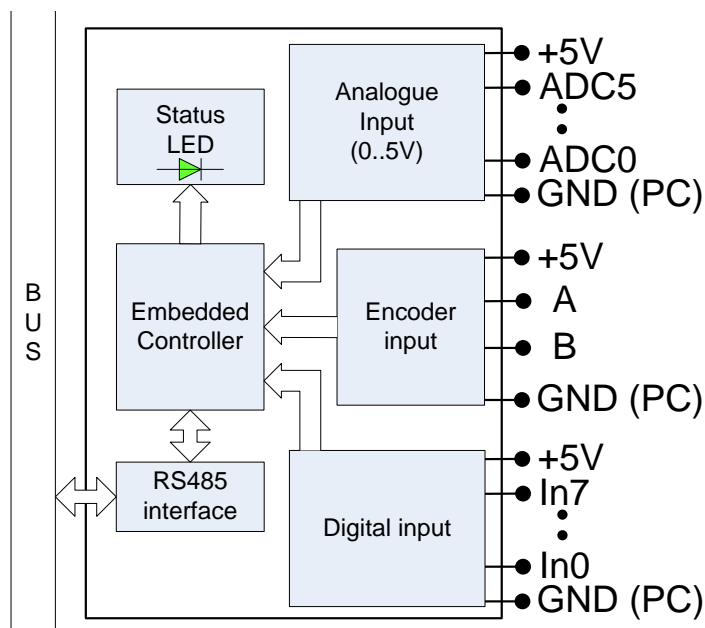
5.6.4.1. Electrical characteristics

Power:

- Bus voltage:..**12** V
- Bus power consumption:...**100** mA
- Maximum load of 5V outputs: ....**500** mA

AD converter:

- Input voltage range:...**0..5** V
- Input leakage current:..**50** nA
- Analogue input resistance: ....**100** MΩ

Input pin characteristics (Digital inputs and encoder inputs):

These are simple non-isolated I/O ports for general purpose usage. All voltage levels are referenced to the PC ground.

- Absolute minimum input voltage:...................................**-0.5** V
- Absolute maximum input voltage: ....**5.5** V
- Maximum low level input voltage: ....**0.3** V
- Minimum high level input voltage:.................................**0.6** V
- Input leakage current...**1** µA

5.6.4.2. Connection
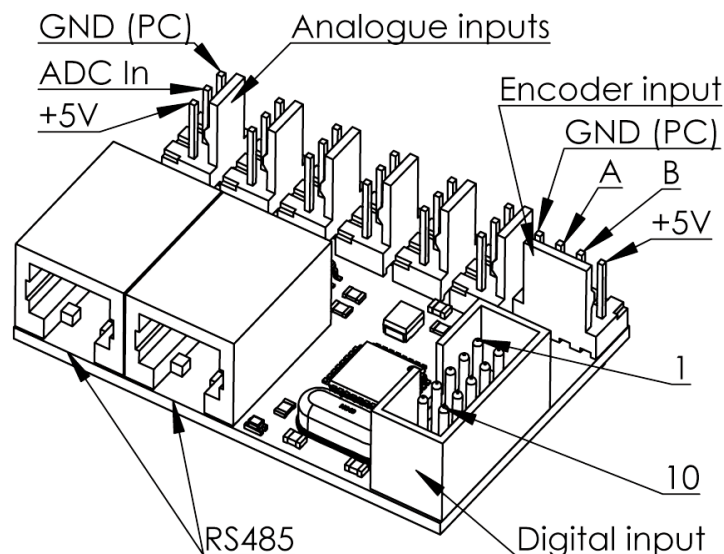
Connectors and pin numbering of the teach pendant module

9	7	5	3	1
In-0	In-2	In-4	In-6	GND (PC)

10	8	6	4	2
VCC (5V)	In-1	In-3	In-5	In-7

Pin assignment table of the digital input connector

5.6.4.3. LinuxCNC HAL configuration

All the pins and parameters are updated by the following function:

`gm.<nr. of card>.rs485`

It should be added to servo thread or other thread with larger period to avoid CPU overload.

Every RS485 module pin and parameter name begins as follows:

`gm.<nr. of card>.rs485.<modul ID>` ,where <modul ID> is 00. Upon request it is possible to order module with pre-programmed different ID.

Pins:

`.adc-<0-5>`	(float, Out)	► The input voltage will be multiplied by scale before being output to .adc- pin.
`.enc-reset`	(bit, In)	► When True, resets counts and position to zero.
`.enc-counts`	(s32, Out)	► Position in encoder counts.
`.enc-rawcounts`	(s32, Out)	► The raw count is the counts, but unaffected by reset.
`.enc-position`	(float, Out)	► Position in scaled units (=.enc-counts/.enc-position-scale).
`.in-<0-7>`	(bit, Out)	► State of input pin.
`.in-not-<0-7>`	(bit, Out)	► Negated state of input pin.

Parameters:

`.adc-scale-<0-5>` (float, R/W) ► The input voltage will be multiplied by scale before being output to .adc- pin.

`.adc-offset-<0-5>` (float, R/W) ► Offset is subtracted from the hardware input voltage after the scale multiplier has been applied.

`.enc-position-scale` (float, R/W) ► Scale in per length unit.

HAL example:

`gm.0.rs485.0.adc-0` # First analogue channel of the node.

    `# gm.0` Means the first PCI motion control card (PCI card address = 0)

    `# .rs485.0` Select node with address 0 on the RS485 bus

    `# .adc-0` Select the first analogue input of the module

5.7. Mechanical dimensions
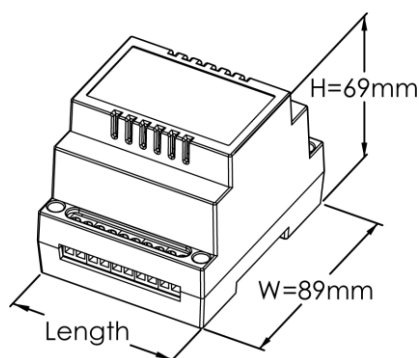
Lengths of each module:

- Relay output module: .... **139** mm
- Input module: .... **65** mm
- ADC & DAC module: .... **107** mm

6. ERRATA

6.1. GM6-PCI card Errata

The revision number in this section refers to the revision of the GM6-PCI card device.

6.1.1. Rev. **1.2**

1. Error:

The PCI card do not boot, when Axis 1. END B switch is active (low).

Found on November 16, 2013.

Reason:

This switch is connected to a boot setting pin of FPGA

Problem fix/workaround:

Use other switch pin, or connect only normally open switch to this switch input pin.

Disclaimer