

# **MATRIXBOX** MX2400 SERIES

## Programmer's Manual

Revision 2021.4.22

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Configuration</b>	<b>4</b>
<b>2.1</b>	Service terminal	<b>5</b>
<b>2.1.1</b>	Connection	<b>5</b>
<b>2.1.2</b>	IP address configuration	<b>7</b>
<b>3</b>	<b>Master</b>	<b>8</b>
<b>3.1</b>	Features	<b>8</b>
<b>3.2</b>	Commands	<b>8</b>
<b>3.2.1</b>	*idn? (Identification)	<b>8</b>
<b>3.2.2</b>	*detect (Card detection)	<b>8</b>
<b>3.2.3</b>	cards? (List of cards)	<b>8</b>
<b>3.2.4</b>	diag? (Diagnostics)	<b>9</b>
<b>4</b>	<b>IO interface</b>	<b>10</b>
<b>4.1</b>	Commands	<b>10</b>
<b>4.1.1</b>	*idn? (Identification)	<b>10</b>
<b>4.1.2</b>	s (Set outputs)	<b>11</b>
<b>4.1.3</b>	c (Clear outputs)	<b>11</b>
<b>4.1.4</b>	r (Read inputs)	<b>12</b>
<b>4.1.5</b>	d (Delay)	<b>12</b>
<b>5</b>	<b>Matrix</b>	<b>13</b>
<b>5.1</b>	Features	<b>13</b>
<b>5.2</b>	Routing	<b>13</b>
<b>5.2.1</b>	MX card routing	<b>13</b>
<b>5.2.2</b>	Master routing	<b>15</b>
<b>5.3</b>	Commands	<b>16</b>
<b>5.3.1</b>	clr set cset (Test-point control)	<b>17</b>
<b>5.3.2</b>	route croute (MX card routing)	<b>18</b>
<b>5.3.3</b>	route croute (Master routing)	<b>20</b>
<b>6</b>	<b>Multimeter</b>	<b>22</b>
<b>6.1</b>	Features	<b>22</b>
<b>6.2</b>	Routing	<b>23</b>
<b>6.3</b>	Commands	<b>23</b>
<b>6.3.1</b>	route (Multimeter routing)	<b>23</b>
<b>6.3.2</b>	meas? (Perform a single measurement)	<b>25</b>
<b>6.3.3</b>	conf (Measurement pre-configuration)	<b>27</b>
<b>6.3.4</b>	zero (Set zero offset)	<b>27</b>

---

6.3.5	*rst (Reset multimeter) . . . . .	28
<b>7</b>	<b>DIO cards</b>	<b>28</b>
7.1	Commands . . . . .	28
7.1.1	*idn? (Identification) . . . . .	28
7.1.2	r ra (Read inputs) . . . . .	29
7.1.3	s (Set outputs) . . . . .	29
7.1.4	c ca (Clear outputs) . . . . .	30
7.1.5	d (Delay) . . . . .	30
7.1.6	cm (Common signal configuration) . . . . .	30

# 1 Introduction

This document describes usage of MX2400 device plugin.

The MX2400 device is represented by a minimum of four virtual devices: [Master](#), [IO interface](#), [Matrix](#) and [Multimeter](#). Optionally there can be also [DIO cards](#) device. Each of them controls the specified section of MX2400.

## Plugin files:

- FpcMx2400.dll (main library)
- FpcMx2400.local.xml (localization)
- Lcs.Fastboot.dll (bootloader library)
- Lcs.Printing.dll (print library)
- PdfSharp.dll (print to PDF capability)

# 2 Configuration

Plugin requires to configure the target MX2400 device(s) in the **configuration dialog**:



In the list of configured devices, insert new row by **entering IP address** of target MX2400 to be connected. After dialog is closed, the new devices will appear in the detected device list, if connection was successful. When re-open the configuration dialog, you can see the MAC address of target host and status information.

**Default IP address** 192.168.100.250  
**Mask** 255.255.255.0  
**Default ports** 2400 and 2401 (see note below)

## Ports note:

Two ports are used - 2400 and 2401. First is for the control channel and second for the asynchronous event channel (i.e. for IO interface). Port numbers cannot be changed.

IP address can be [configured](#) using the [service terminal](#).

There is a web-based administration, accessible on port 8080 (enter "http://<your-mx-ip-address>:8080" to your web-browser). Actually it only shows summary information and list of detected cards. No configuration is possible to do here.

## 2.1 Service terminal

MX2400 supports the low-level configuration using the RS-232 service interface. The connection to computer is done using the RS-232 (DSUB-9) connector on the master card (see the Hardware manual for description). The service interface emulates **standard VT100 ANSI terminal**. It is recommended to use the [PuTTY](#) software or any other VT100-compatible ANSI terminal.

### 2.1.1 Connection

#### Parameters of communication:

115200 bps, 8 data bits, no parity, no flow control

#### Required cable wiring:

Computer RS-232			Master card RS-232/SYNC-IO	
Pin	Signal		Pin	Signal
2	RxD	↔	3	TxD
3	TxD	↔	2	RxD
5	GND	↔	5	GND

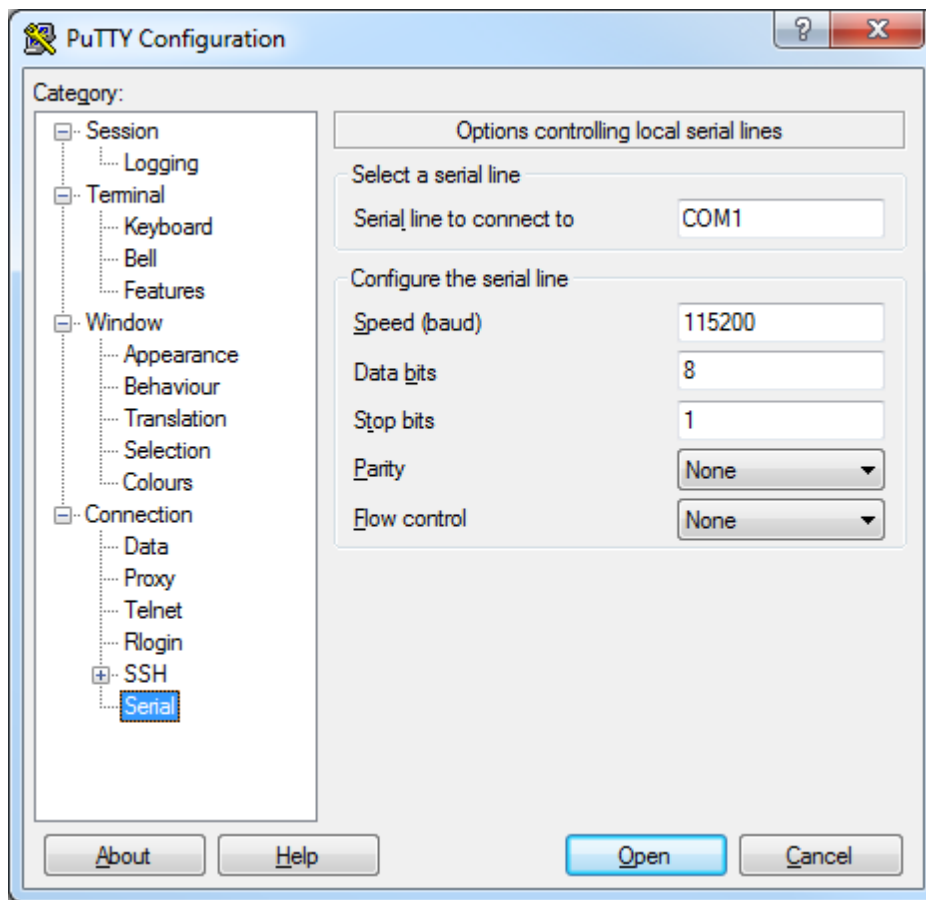
(CAN9 female to CAN9 female, cross-link)

The easiest way how to connect using the PuTTY is via command-line:

```
putty.exe -serial COM1 -sercfg 115200,8,n,1,N
```

Replace *COM1* argument for COMx that you really used.

You can of course connect using the PuTTY's GUI - do not forget to enter correct baudrate (115200 bps) and configure the serial communication:



Connecting via PuTTY

When you see the black terminal screen, on the keyboard press **Ctrl + S + T** (**Ctrl + Service + Terminal**). You should see similar to following on the screen:

```
MatrixBox MX2400 Service Terminal
2015-2-6 14:20:42

      F1      System
      F2      Network
      F3      Misc
```

If nothing is shown, check the connection between Master card's RS-232 and computer and/or the cable wiring.

#### Navigation in the terminal:

- F1 to F12: go to sub-menu
- Enter: confirm value
- Esc: discard value, go back to the previous screen

## 2.1.2 IP address configuration

Press F2 - enter Network:

```
MatrixBox MX2400 Service Terminal
2015-2-6 14:46:25

Network

      F1      Set IP address and mask
      F2      Restore default configuration

MAC address: 00:1E:C0:AF:AF:C3
Link: 100 Mb/s, full-duplex
Connected host: none

IP address: 192.168.1.253
Mask: 255.255.255.0
```

Press F1 to set IP address:

```
...

Enter IP address: 192.168.10.5
```

Press Enter to confirm:

```
...

Enter IP address: 192.168.10.5
Enter mask (optional):
```

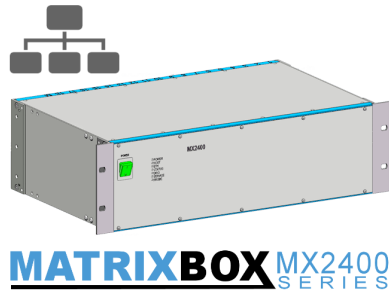
Leave the mask empty and press Enter:

```
...

IP address: 192.168.10.5
Mask: 255.255.255.0

IP address and mask successfully changed.
```

## 3 Master



Master device is the main device of the MX2400.

### 3.1 Features

- Connection control to MX2400 via ethernet
- List of detected MX cards, re-detecting cards
- Show state of MX cards
- Controls of special features of MX cards if available (LCMX card diagnostics for example)
- Updating firmware

### 3.2 Commands

#### 3.2.1 \*idn? (Identification)

```
*idn?
```

Return identification string.

##### Parameters

No parameters.

##### Return value

No return value.

#### 3.2.2 \*detect (Card detection)

```
*detect{:rst=[bool]}
```

Re-detect cards in the MX2400 system.

##### Parameters

<code>rst</code>	[bool]	Reset all cards before detection (via dedicated BUS signal). Optional, default: false
------------------	--------	--

##### Return value

No return value.

#### 3.2.3 cards? (List of cards)

```
cards?
```

Get list of all detected cards.



## Parameters

No parameters.

## Return value

$\langle pos_0 \rangle, \langle type_0 \rangle: \langle pos_1 \rangle, \langle type_1 \rangle: \dots: \langle pos_N \rangle, \langle type_N \rangle$

..where `pos` is the **zero-based position** of card in the system and `type` is the card type-ID:

- 139: MXC2-LCMX-32
- 144: MXC2-DEV-4/2/2
- 167: MXC2-HCMX-16
- 200: MXC2-DIO-64

## Examples

`cards?`

Returns `0, 139: 1, 139: 17, 144` for example, which means MXC2-LCMX-32 cards on positions 1 and 2 and MXC2-DEV-4/2/2 card on position 18.

## 3.2.4 diag? (Diagnostics)

`diag?: <location>`

Run diagnostics on selected card.

**Note:** currently supports **only LCMX and HCMX** cards.

### Parameters

<code>location</code>	[string]	Card position in the system: <ul style="list-style-type: none"> <li>• location in format <code># { box. } card</code>, where <code>box</code> is box number (1 or 2, only when two boxes are interconnected together) and <code>card</code> is card position, starting by 1 (same, like the marking on the top of the box)</li> <li>• direct address (zero-based)</li> </ul>
-----------------------	----------	--

### Return value

Depends on diagnosed card type.

**Generally:** when diagnostics was successful, return value is `PASS`

#### MXC2-LCMX MXC2-HCMX

- when OK, returns `PASS`
- when NG, returns `bLbH: t0. . . tN`, where "b" represents BUS A and C (\* = OK, S = NG - short) and "t" represents TP1..N (\* = OK, L = SHORT to L, H = SHORT to H, S = SHORT both)

### Examples

`diag?: 0`

Perform diagnostics of card with address 0. Returns i.e. `PASS` - card is OK.

`diag?: #6`

Perform diagnostics of card at location 6. Returns i.e. (for LCMX card)

`** : **L*****H`, which FAIL - in detail: buses are OK, but TP3 is shorted to L and TP32 is shorted to H.

```
diag?: #1. 13
```

Perform diagnostics of card in the box 1, location 13. Returns i.e. `PASS` - card is OK.

```
diag?: *
```

Run diagnostics of all cards.

**Note:** currently supports **only LCMX and HCMX** cards.

### Parameters

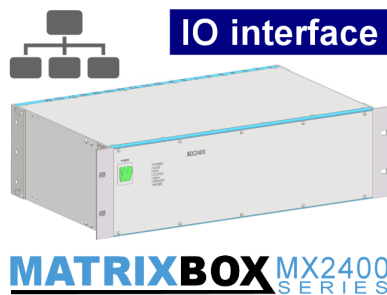
No parameters.

### Return value

Number of FAILs:

- 0: PASS
- >=1: FAIL

## 4 IO interface



Internal MX2400 IO interface, featuring 16 inputs and 16 outputs.

### 4.1 Commands

Commands to control IO interface. **Chaining of commands is supported.** This allows to pass more commands to IO interface at once.

#### Chaining example:

```
s: 0: d: 100: r: 1: d: 10: c: 0
```

Will do:

- Set output 0
- Delay for 100 ms
- Read state of input 1
- Delay for 10 ms
- Clear output 0

#### 4.1.1 \*idn? (Identification)

```
*idn?
```

Return identification string.

### Parameters

No paramters.

### Return value

No return value.

## 4.1.2 s (Set outputs)

```
s: <out0>{: <out1>: . . . : <outN>}
```

Set (switch ON) specified output(s).

### Parameters

out [int] Number of output signal in range of 0 to 15.

### Return value

No return value.

### Examples

```
s: 0: 5: 12  
Set outputs 0, 5 and 12 to ON.
```

## 4.1.3 c (Clear outputs)

```
c: <out0>{: <out1>: . . . : <outN>}
```

Clear (switch OFF) specified output(s).

### Parameters

out [int] Number of output signal in range of 0 to 15.

### Return value

No return value.

### Examples

```
c: 0: 5: 12  
Set outputs 0, 5 and 12 to OFF.
```

```
c: *  
c: all
```

Clear all output(s).

### Parameters

No parameters.

### Return value

No return value.

### 4.1.4 r (Read inputs)

```
r: <in0>{: <in1>: . . . : <inN>}
```

Read current state of specified input(s).

#### Parameters

`in` [int] Number of input signal in range of 0 to 15.

#### Return value

"0" when input is OFF

"1" when inputs is ON

If there is more than 1 input to read, the return value contains all input states (0 or 1), separated by a colon (":").

#### Examples

```
r: 7
```

Read state if input number 7, return value will be for example "1" (ON).

```
r: 2: 8: 11
```

Read state if inputs 2, 8 and 11, return value will be for example "1: 0: 1" (2 = ON, 8 = OFF, 11 = ON).

```
r: *
r: all
```

Read current state of all inputs.

#### Parameters

No parameters.

#### Return value

16 states ("0" or "1") in **ordered by input 15 to 0** (MSB first), separated by a colon (":").

### 4.1.5 d (Delay)

```
d: <delay>
```

Put a delay between sub-commands in the sequence.

#### Parameters

`delay` [int] Delay in milliseconds

#### Return value

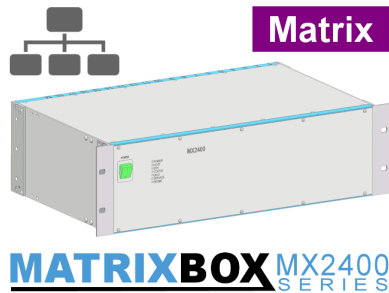
No return value.

#### Examples

```
s: 0: d: 250: c: 0
```

Set output 0 for 250 ms (make a pulse).

## 5 Matrix



Matrix and routing interface.

### 5.1 Features

- Test-point control (connecting any TP to LOW or HIGH bus)
- Routing control of each MX card
- Master routing control (Connection cards)
- Sequence measure control

### 5.2 Routing

Generally, routing is about **interconnection** of internal card's buses to main back-plane buses (BUS 1 - 4).

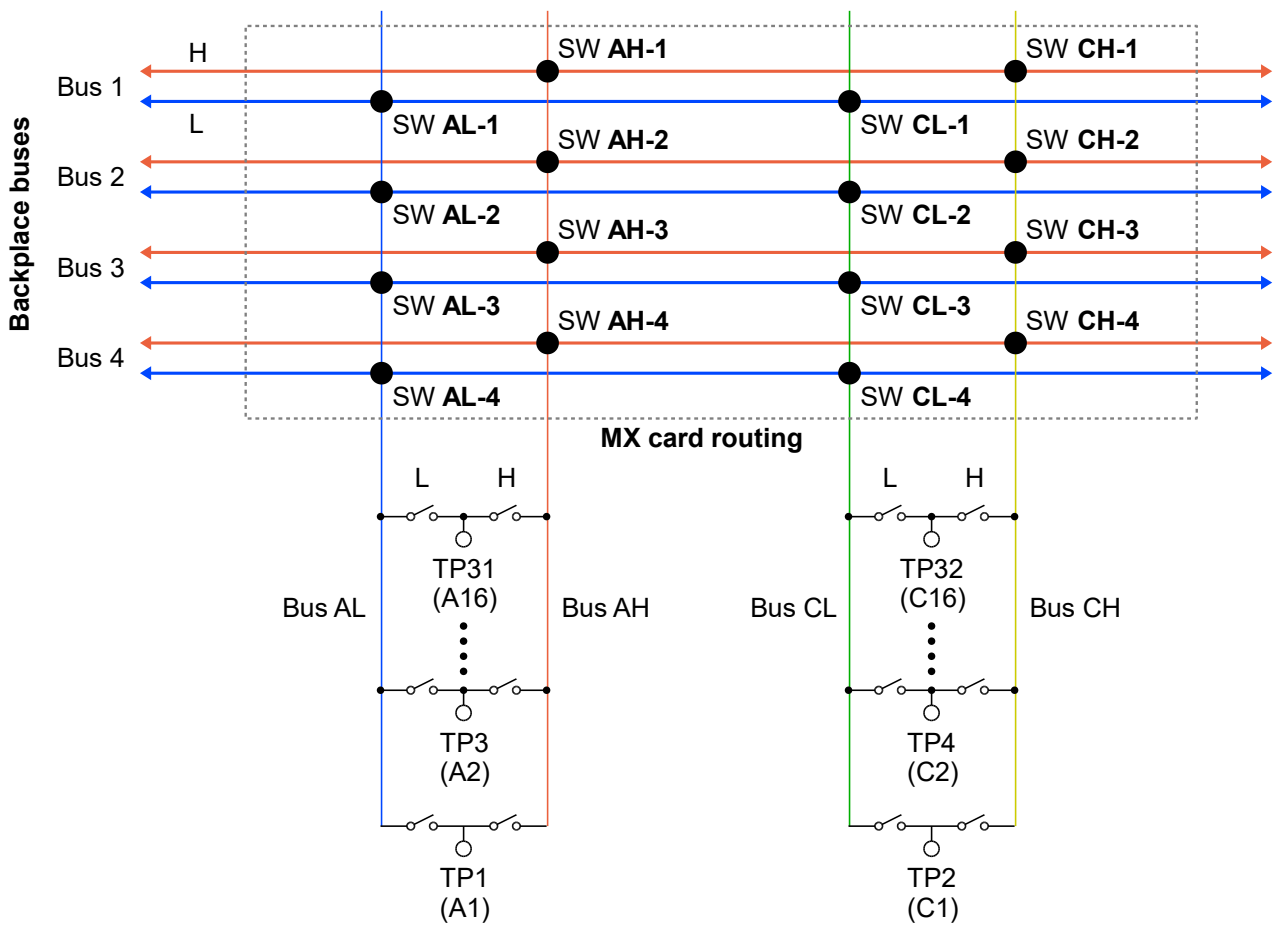
**MX2400 has two routings:**

- [Card routing](#) - define LCMX card interconnection of internal buses to main bus
- [Master routing](#) - define Connection card interconnection of inputs/outputs to main bus

This chapter describes a little bit the internal electrical solution and simplified schematics of routing for understanding what is controlled by [commands](#).

#### 5.2.1 MX card routing

MX card routing is about interconnection of LCMX card **internal A and C** buses to main back-plane **BUS 1 - 4**



Full diagram of MX card routing

LCMX card has internally two split buses: **A** and **C**. Each bus consists of **L** and **H** line.

A/C and L/H results to **4 signal lines** in total:

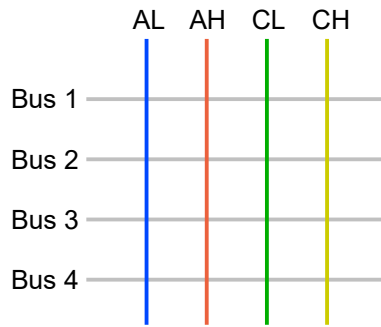
- **AL** - bus A, line LOW
- **AH** - bus A, line HIGH
- **CL** - bus C, line LOW
- **CH** - bus C, line HIGH

Routing is electrically controlled by set of routing switches **SW AL-\*** (for bus A, line LOW), **SW AH-\*** (A/HIGH), **SW CL-\*** (C/LOW) and **SW CH-\*** (C/HIGH).

Using routing switches above you can connect them to corresponding (L/H) main back-plane buses: AL to BUS 1-4 L, AH to BUS 1-4 H and so on.

**L line** of the card is possible to connect **only to L line** of main bus (and the same for H line). Routing does not allow to connect i.e. AL to BUS 1 H.

Because of impossibility to interconnect different lines (L to H), it is possible to simplify the routing schematic of every card to following matrix:

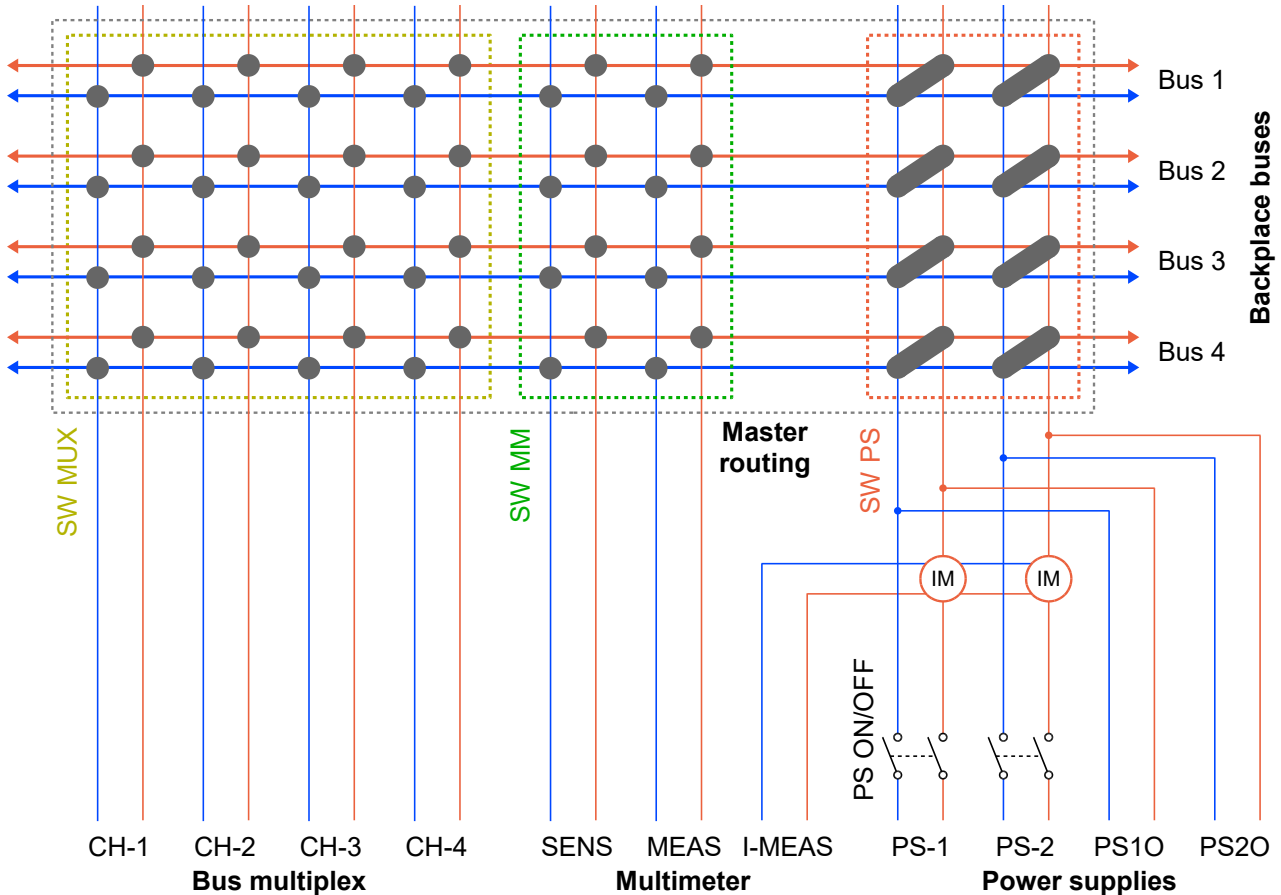


**Simplified MX card routing matrix**

Any cross-point of Bus 1-4 and AL/AH/CL/CH represents a corresponding switch. Cross-points of AL and Bus 1-4 means switches SW AL-\* and so on.

### 5.2.2 Master routing

Master routing is about interconnection of Connection card inputs/outputs to main back-plane buses.



**Full diagram of master routing/Connection card**

Connection card controls routing of following signals:

- **Bus multiplex** (CH-1 to CH-4) general inputs/outputs
- **Multimeter** (SENS/MEAS/I-MEAS) for full external measurement (2W/4W resistance, voltage, current)
- **Power supplies inputs** (PS-1/PS-2) for power supplies connection
- **Power supplies outputs** (PS10/PS20)

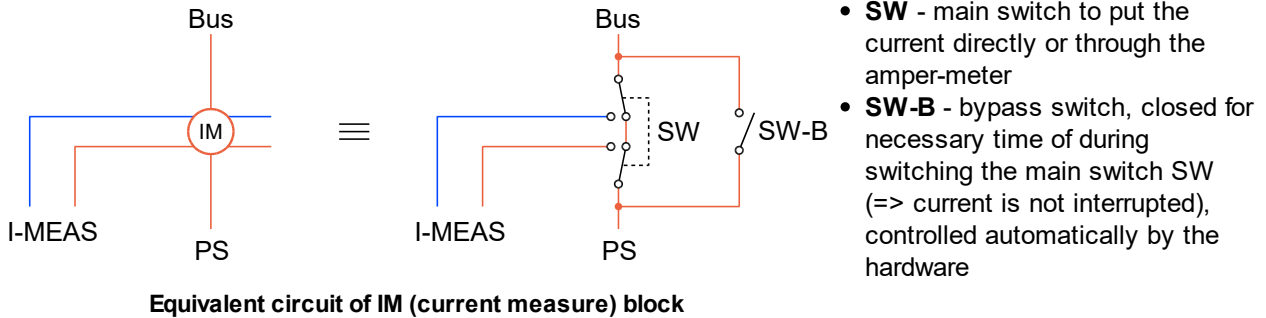
Each line L/H of these signals can be interconnected with corresponding line of any Bus 1 - 4. Except Power supplies, it is possible to connect L/H separately to different busses (for example CH-1 L to Bus 1 L and CH-1

H to Bus 3 H). Power supply L and H lines can be connected to Bus 1 - 4 only both at once.

Current measurement block:

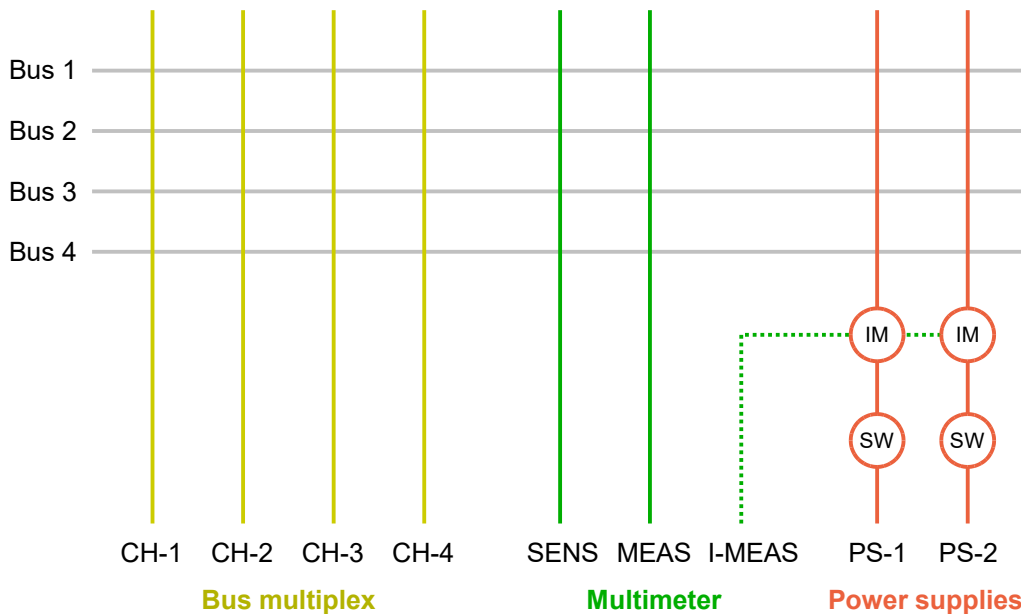
- **I-MEAS** (IM) to measure current of PS-1/2 using the external multimeter

IM block enables to route PS-1 or PS-2 current directly or through an amper-meter (I-MEAS lines). See equivalent schematic of IM block below:



The I-MEAS signal is not multiplexed to the main bus 1 - 4, it is connected always to both IM blocks (= equivalent switches of current measure).

Typically, both L and H of any signal are connected to the same bus and because power supplies can be connected only this way, we can simplify the schematic to following:



Any cross of any signal and bus represents a corresponding switch. For bus multiplex and multimeter signals it is possible to connect L or H only.

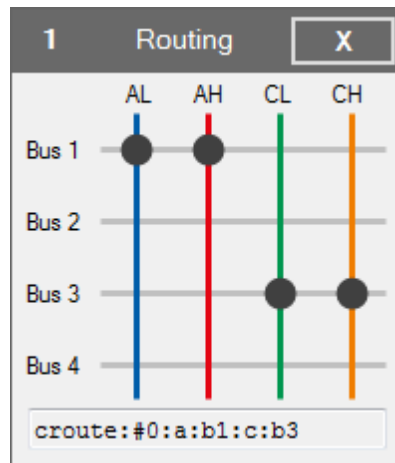
### 5.3 Commands

Commands to control test-points and routing of Matrix. **Chaining of commands is supported.**

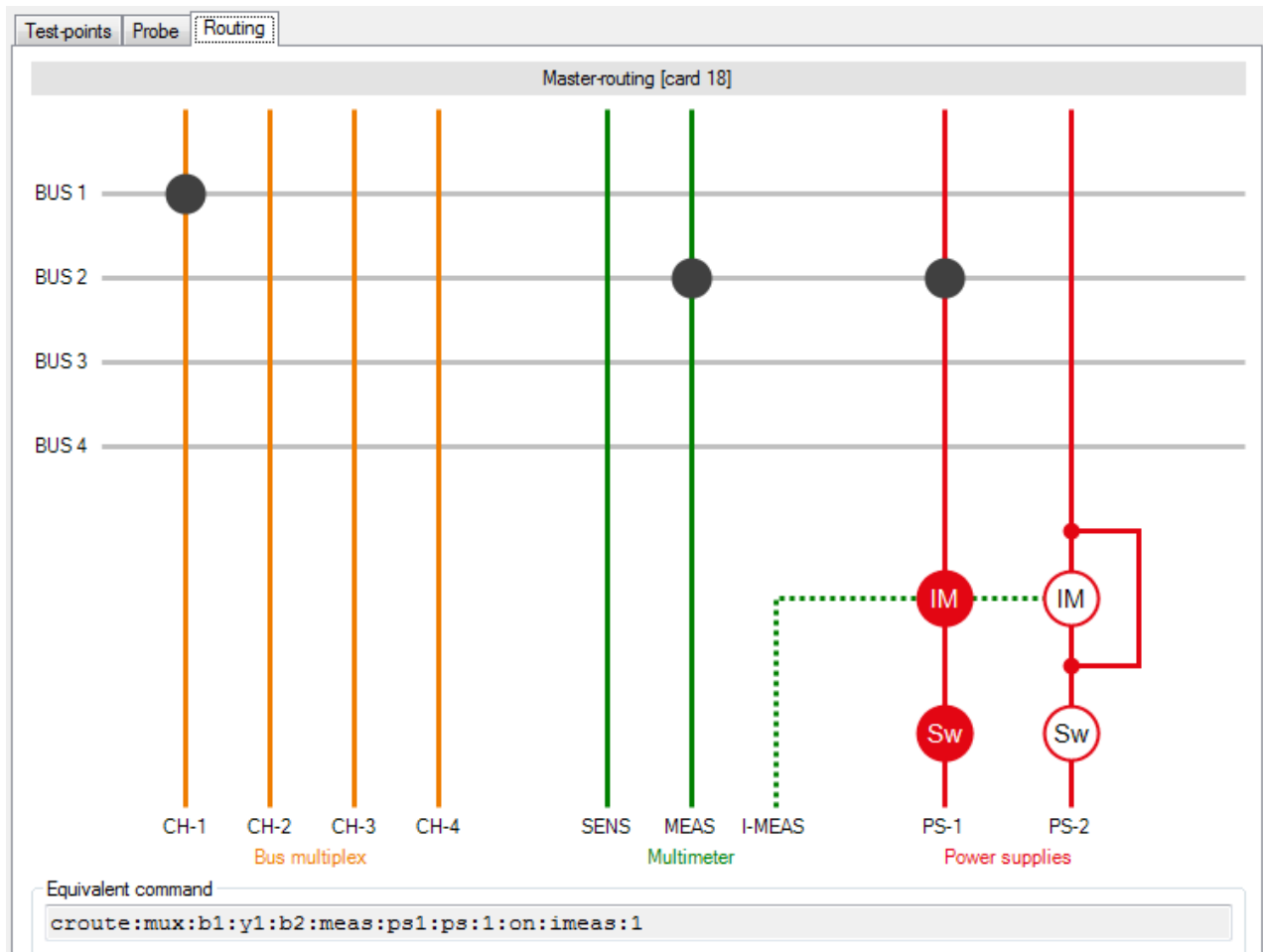
#### Tip

Routing commands can be copied from plugin's GUI control - both from MX card routing and also for Master routing. The command below the routing schematic for every MX and routing card will configure the routing like configured using the schematic.





MX card routing equivalent command



Master routing (Connection card) equivalent command

### 5.3.1 clr|set|cset (Test-point control)

```

clr:
set:
cset:
    <low>: <high>
    -or-
    L: <L1>{: <L2>: . . . : <LN>} : H: <H1>{: <H2>: . . . : <HN>}
    
```

Connect (`set`, `cset`) or disconnect (`clr`) a pair of TP to/from LOW/HIGH or a set of TP to/from LOW and set of TP to/from HIGH bus. The "cset" does almost the same like "set", but "cset" disconnects all previously connected TPs first.

### Parameters

<code>low</code> $L_N$	[int]	TP number (1 to N) to be connected to <b>LOW</b> bus.
<code>high</code> $H_N$	[int]	TP number (1 to N) to be connected to <b>HIGH</b> bus.

### Return value

No return value.

### Examples

```
cset: 10: 15
```

Disconnect all TPs and connect TP10 to LOW and TP15 to HIGH bus.

```
set: 10: 15
```

Leave currently connected TPs and connect moreover TP10 to LOW and TP15 to HIGH.

```
clr: 10: 15
```

Disconnect TP10 from LOW and TP15 from HIGH bus. Any other connected TP is leaved untouched.

```
cset: l: 10: 12: 14: h: 15: 17: 19
```

Disconnect all TPs and connect TPs 10, 12 and 14 to LOW and TPs 15, 17 and 19 to HIGH bus.

```
clr: l: 10: 12: h: 15: 17
```

Disconnect TPs 10 and 12 from LOW and TPs 15 and 17 from HIGH bus.

```
clr: l: 10: 12: set: l: 8: 9
```

Chaining example. Disconnect TPs 10 and 12 from LOW and connect TPs 8 and 9 to LOW. Any other TP is leaved untouched.

```
clr: *
clr: all
```

Disconnect all TPs from any bus.

### Parameters

No parameters.

### Return value

No return value.

## 5.3.2 route|croute (MX card routing)

```
route:
croute: <loc>{: <card-bus>}: <main-bus>
```

Interconnection of internal MX card's buses (or lines) to any main bus 1 - 4. The "croute" clears all previous routing before applying new (like cset/set of test-point control). The "route" clears only settings of passed main-bus.

The pair <card-bus>:<main-bus> can be several times in the one route|croute command - see examples.

**Parameters**

`loc`                                    `#[int]` or `*`                                    MX card location 1 to N. If number is passed, it must be prefixed by a sharp-character (#).  
To set routing of all MX cards, pass an asterisk (\*) as a location.

`card-bus`                                    `[enum]`                                    Card's internal bus or line:

- `a1` = A, low
- `c1` = C, low
- `ah` = A, high
- `ch` = C, high
- `l` = A + C, low (LOW lines of A and C)
- `h` = A + C, high (HIGH lines of A and C)
- `a` = A, low + high (both LOW + HIGH lines of A bus)
- `c` = C, low + high (both LOW + HIGH lines of C bus)
- `ac` = A + C, low + high (all card's buses)

If this parameter is not passed, the default is "ac" (all card buses).

See [MX card routing](#) description for details about lines/buses.

`main-bus`                                    `[enum]`                                    Main bus:

- `b1` = Bus 1
- `b2` = Bus 2
- `b3` = Bus 3
- `b4` = Bus 4

H/L of main bus is defined by card-bus parameter. L of card-bus is connect to L of main-bus (and the same for H).

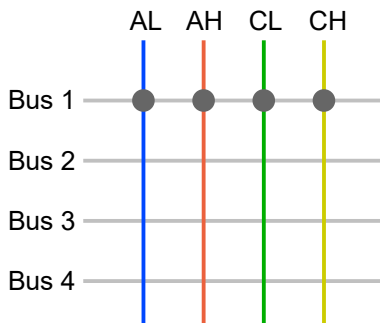
**Return value**

No return value.

**Examples**

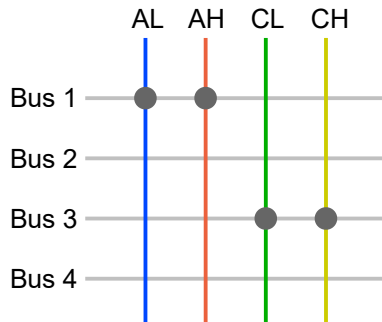
`croute: *: b1`

Clear all previous routing of all cards and connect both card's internal buses to main bus 1 for all cards. This is the most used command for most of the systems.



route: #3: a: b1: c: b3

Only for card at location 3. Clear previous routings of b1 and b3 and connect card's A bus (L + H) to main bus 1 and card's C bus (L + H) to main bus 3. Previous routing to b2 and b4 (if exists) is leaved untouched.



### 5.3.3 route|croute (Master routing)

```
route:
croute: mux{ : <main-bus>: <mux-ch0>{ : <mux-ch1>: . . . : <mux-chN> } }
```

Interconnection of main bus and Connection card input/output signals.

#### Parameters

main-bus [enum]

Main bus definition:

- b1 to b4, bus 1 - 4, low + high
- b1l to b4l, bus 1 - 4, low only
- b1h to b4h, bus 1 - 4, high only

mux-ch [enum]

Signal to be connected to the main-bus:

- ch1 to ch4 or y1 to y4 (master card with older firmware version accepts y1-y4 only), Bus Multiplex block (SW MUX switches)
- sens and meas, Multimeter block (SW MM switches)
- ps1 and ps2, represents PS-1 and PS-2 PSU inputs, Power supplies block (SW PS switches)

See [Master routing](#) description for details about blocks and signals.

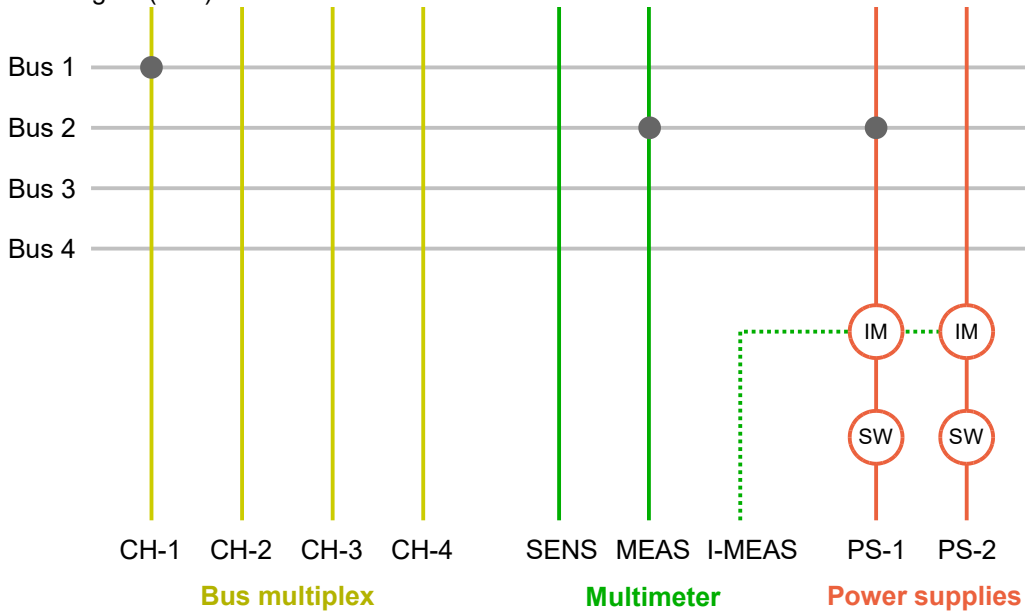
#### Return value

No return value.

### Examples

```
croute: mux: b1: ch1: b2: meas: ps1
```

Clear all previous routing and connect card's ch1 signal (both L+H) to main bus 1 and power supply 1 and sens signal (L+H) to main bus 2.



```
croute: mux
```

Clear all routing settings of "mux" block.

```
route:
croute: ps{ : <ps-nr>: <state>}
```

External power supplies ON/OFF control.

### Parameters

ps-nr	[int]	Power supply number, 1 or 2
state	[enum]	on or off

### Return value

No return value.

### Examples

```
route: ps: 1: on: 2: off
```

Turn PS-1 ON and PS-2 OFF.

```
croute: ps
```

Using croute to turn off both power supplies.

```
route:
croute: imeas{ : <imeas-nr>| off}
```

External current measurement control. During switching the measurement, the internal bypass realy (SW-B, see [equivalent IM schematic](#) on Master routing) is automatically closed to prevent supply interruption.

## Parameters

`imeas-nr` [int] or "off" Activate external current measurement (I-MEAS) for specified power supply 1 or 2.  
To deactivate currently selected measurement pass "off" instead of PSU number.

## Return value

No return value.

## Examples

```
route: imeas: 1
Activate external current measure of PS-1.
```

```
croute: imeas
route: imeas: off
Deactivate external current measurement.
```

```
croute: m
```

Clear all blocks of master routing (mux, ps, imeas). The "m" parameter is required to distinguish the "croute" command for MX card routing and master routing.

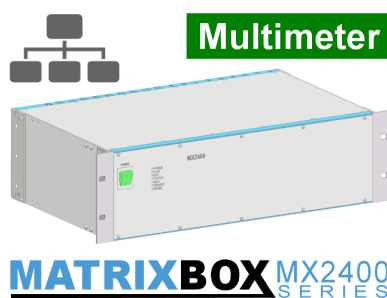
## Parameters

No parameters.

## Return value

No return value.

# 6 Multimeter



Internal MX2400 multimeter.

## 6.1 Features

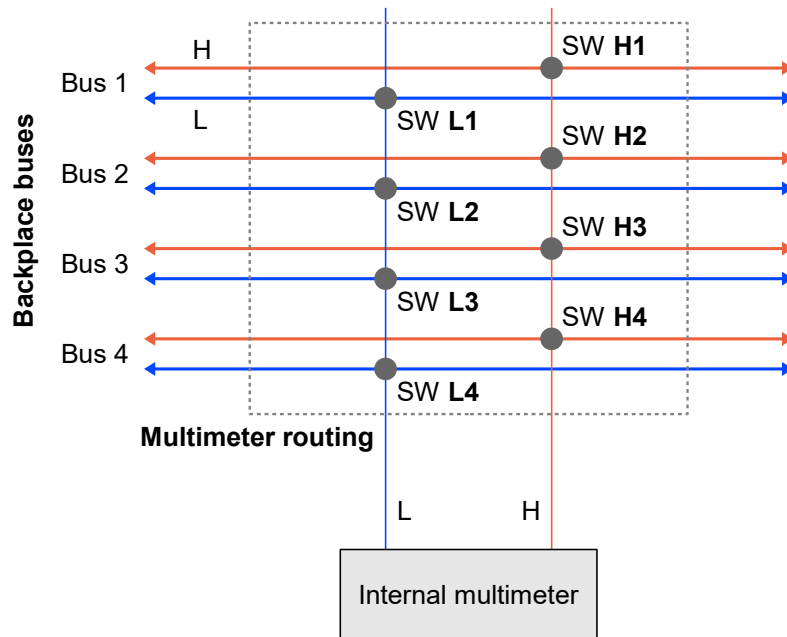
Measurement of:

- Voltage
- Resistance (2W)
- Capacity
- Inductance
- Continuity

See MX2400 datasheet to get DMM parameters.

## 6.2 Routing

Multimeter routing is about interconnection of internal MX2400 multimeter inputs to main back-plane buses.

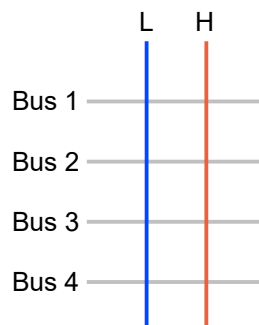


Internal multimeter routing

Multimeter are connected to back-plane bus 1 - 4 via **SW Lx** (L1 to L4) and **SW Hx** (H1 to H4) switches of the routing block.

The main L and H lines of the multimeter can be connected both to lines of the same bus number or different bus numbers. **L line** of multimeter is possible to connect **only to L line** of main bus (and the same for H line). Routing does not allow to connect i.e. multimeter's L to H Bus 2, etc.

Because of impossibility to interconnect different lines (L to H), it is possible to simplify the routing schematic to following matrix:



Simplified internal multimeter routing

Any cross-point Bus 1 to 4 and L or H line represents a corresponding switch (SW L1-4 and SW H1-4).

## 6.3 Commands

### 6.3.1 route (Multimeter routing)

```
route: { <mm-bus> } : <main-bus>
```

Clear and set the new multimeter's routing to main bus.

**Note:** this command will not set the routing switches (relays) immediately, it only configure the DMM to use

them. Switches are connected at the time of measuring for necessary time only (until measuring is finished).

**Tip:** it is possible to copy the equivalent routing command from the text-box of "routing" section in the multimeter's control form (GUI)

### Parameters

`mm-bus` [enum]

Multimeter's internal bus or line:

- `l` = multimeter's LOW line
- `h` = multimeter's HIGH line

If this parameter is not passed, both buses (L and H) are selected by default.

`main-bus` [enum]

See [multimeter routing](#) description for details about lines/buses.

Main bus:

- `b1` = Bus 1
- `b2` = Bus 2
- `b3` = Bus 3
- `b4` = Bus 4

H/L of main bus is defined by `mm-bus` parameter. L of `mm-bus` is connect to L of main-bus (and the same for H).

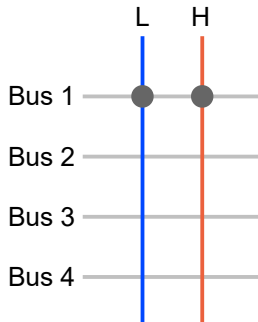
### Return value

No return value.

### Examples

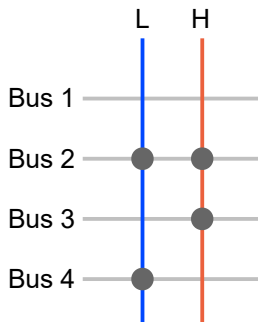
`route: b1`

Connect multimeter's both lines (L+H) to main bus 1.



`route: b2: l: b4: h: b3`

Connect multimeter's L line to main bus L 2 & 4 and H line to main bus H 2 & 3.





### 6.3.2 meas? (Perform a single measurement)

```
meas?: volt{ : <volt-range> }
```

Voltage measurement.

#### Parameters

<code>volt-range</code>	[enum]	<p>Voltage measurement range:</p> <ul style="list-style-type: none"> <li>• <code>auto</code> - auto-range functionality</li> <li>• <code>2v5</code> - 0 to 2,5V (~ 50µV resolution)</li> <li>• <code>25v</code> - 0 to 25V (~ 0,5mV resolution)</li> <li>• <code>250v</code> - 0 to 250V (~ 5mV resolution)</li> </ul>
-------------------------	--------	--

**Default:** `auto`

#### Return value

Voltage value in exponential format (0.0e+0), positive or negative, always with decimal point. If the value is above the selected range, the return value is "9.9e+37".

#### Examples

```
meas?: volt
Measure voltage using auto-range.
```

```
meas?: volt: 25v
Measure voltage using fixed range of 25V.
```

```
meas?: res{ : <res-range> }
```

Resistor measurement.

#### Parameters

<code>res-range</code>	[enum]	<p>Resistor measurement range:</p> <ul style="list-style-type: none"> <li>• <code>auto</code> - auto-range functionality</li> <li>• <code>1k</code> - 0 to 1kΩ (~ 15mΩ resolution)</li> <li>• <code>100k</code> - 0 to 100kΩ (~ 1,5Ω resolution)</li> <li>• <code>10m</code> - 0 to 10MΩ (~ 150Ω resolution)</li> </ul>
------------------------	--------	---

**Default:** `auto`

#### Return value

Resistance value in exponential format (0.0e+0), always with decimal point. If the value is above the selected range, the return value is "9.9e+37".

#### Examples

```
meas?: res
Measure resistor using auto-range.
```

```
meas?: res: 100k
Measure resistor using fixed range of 100k.
```

```
meas?: cont
```

Continuity test.

### Parameters

No parameters.

### Return value

Resistance value in exponential format (0.0e+0), always with decimal point. It does not measure the resistance value, but only compares with OPEN/SHORT threshold level (about 500Ω) using the comparator. Representing values:

- **OPEN**: 9.9e+37 (infinite resistance)
- **SHORT**: 0.0e+0 (zero resistance)

### Examples

```
meas?:cont  
Continuity test.
```

```
meas?:cap
```

Capacitor measurement.

### Parameters

No parameters.

### Return value

Capacitance value in exponential format (0.0e+0), always with decimal point. If the value is above the possible maximum range, the return value is "9.9e+37".

### Examples

```
meas?:cap  
Capacitor measurement.
```

```
meas?:ind
```

Inductor measurement.

### Parameters

No parameters.

### Return value

Inductance value in exponential format (0.0e+0), always with decimal point. If the value is above the possible maximum range, the return value is "9.9e+37".

### Examples

```
meas?:ind  
Inductor measurement.
```

```
meas?
```

Measurement using pre-configured function and range using the [conf](#) command.

## Parameters

No parameters.

## Return value

Inductance value in exponential format (0.0e+0), always with decimal point.  
Depends on pre-configured function.

### 6.3.3 conf (Measurement pre-configuration)

```
conf: <function>{: <range>}
```

Configure measurement to use the [meas?](#) command without arguments.

#### Parameters

<code>function</code>	[enum]	Measurement function to set and configure: <ul style="list-style-type: none"> <li>• <code>volt</code> - Voltage measurement</li> <li>• <code>res</code> - Resistor measurement</li> <li>• <code>cont</code> - Continuity test</li> <li>• <code>cap</code> - Capacitor measurement</li> <li>• <code>ind</code> - Inductor measurement</li> </ul>
<code>range</code>	[enum]	When voltage ( <code>volt</code> ) or resistance ( <code>res</code> ) is measured, it is possible to specify range. For details see the <code>volt-range</code> and <code>res-range</code> of <a href="#">meas?:volt</a> and <a href="#">meas?:res</a> commands.

#### Return value

No return value.

#### Examples

```
conf: volt
Configure to measure voltage using auto-range.
```

```
conf: res:100k
Configured to meas resistance using fixed 100k range.
```

### 6.3.4 zero (Set zero offset)

```
zero: <function>
```

Measure value of specified function and set it like an offset of this function. This offset is subtracted from every following measurement until multimeter reset or system reset.

**Note:** zero offset is measured using auto-ranging (if available)

#### Parameters

<code>function</code>	[enum]	Measurement function to measure and set offset: <ul style="list-style-type: none"> <li>• <code>volt</code> - Voltage measurement</li> <li>• <code>res</code> - Resistor measurement</li> <li>• <code>cap</code> - Capacitor measurement</li> <li>• <code>ind</code> - Inductor measurement</li> </ul>
-----------------------	--------	---

#### Return value

No return value.

```
zero: *rst
```

Reset all multimeter's offsets.

### Parameters

No parameters.

### Return value

No return value.

## 6.3.5 \*rst (Reset multimeter)

```
*rst
```

Reset following multimeter configuration:

- routing
- zero offsets
- configured function

### Parameters

No parameters.

### Return value

No return value.

## 7 DIO cards



DIO-64 card(s) representation. This device **merges all DIO cards** in the system into the one large DIO device with total number of inputs and outputs, given by a sum of IO of all cards.

This device is shown only when at least one MXC2-DIO-64 card is in the system.

## 7.1 Commands

### 7.1.1 \*idn? (Identification)

```
*idn?
```

Return identification string.

### Parameters

No paramters.

**Return value**

No return value.

**7.1.2 r|ra (Read inputs)**

```
r: <in0>{: <in1>: . . . : <inN>}
```

Read current state of specified input(s).

**Parameters**

<code>in</code>	[int]	Number of input signal in range of 0 to N - 1, where N is total number of inputs in the system.
-----------------	-------	---

**Return value**

"0" when input is OFF  
 "1" when inputs is ON

If there is more than 1 input to read, the return value contains all input states (0 or 1), separated by a colon (":").

**Examples**

r: 7  
 Read state if input number 7, return value will be for example "1" (ON).

r: 2: 8: 11  
 Read state if inputs 2, 8 and 11, return value will be for example "1: 0: 1" (2 = ON, 8 = OFF, 11 = ON).

```
ra
```

Read current state of all inputs.

**Parameters**

No parameters.

**Return value**

N states ("0" or "1") in **ordered by input (N - 1) to 0** (MSB first), separated by a colon (":").

**7.1.3 s (Set outputs)**

```
s: <out0>{: <out1>: . . . : <outN>}
```

Set (switch ON) specified output(s).

**Parameters**

<code>out</code>	[int]	Number of output signal in range of 0 to N - 1.
------------------	-------	---

**Return value**

No return value.

## Examples

```
s: 0: 5: 12
```

Set outputs 0, 5 and 12 to ON.

### 7.1.4 c|ca (Clear outputs)

```
c: <out0>{: <out1>: . . . : <outN>}
```

Clear (switch OFF) specified output(s).

#### Parameters

out	[int]	Number of output signal in range of 0 to N - 1.
-----	-------	---

#### Return value

No return value.

#### Examples

```
c: 0: 5: 12
```

Set outputs 0, 5 and 12 to OFF.

```
ca
```

Clear all output(s).

#### Parameters

No parameters.

#### Return value

No return value.

### 7.1.5 d (Delay)

```
d: <delay>
```

Put a delay between sub-commands in the sequence.

#### Parameters

delay	[int]	Delay in milliseconds
-------	-------	-----------------------

#### Return value

No return value.

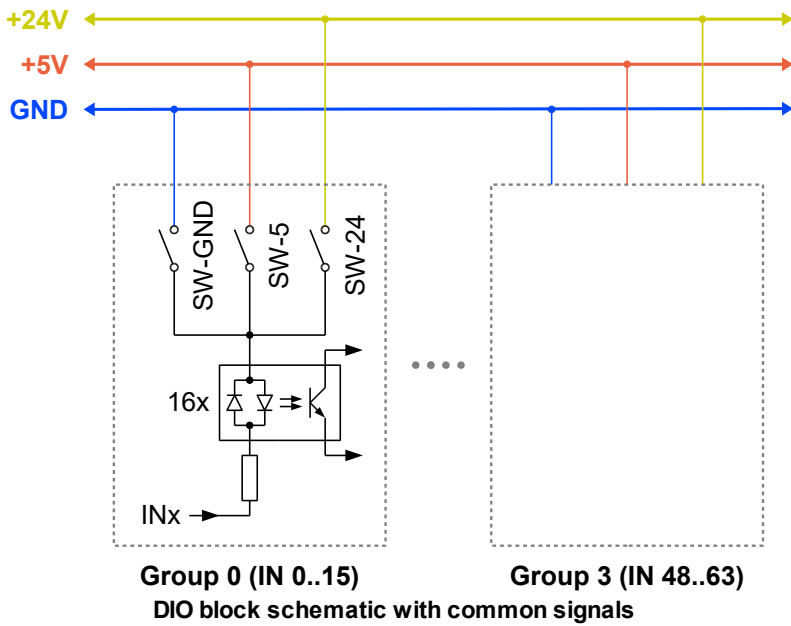
#### Examples

```
s: 0: d: 250: c: 0
```

Set output 0 for 250 ms (make a pulse).

### 7.1.6 cm (Common signal configuration)

Card has DIO channels divided into 4 groups by 16 DIO channel. IN stage requires a common signal to activate inputs. This signal is applied internally from internal power supplies.



SW-GND, SW-5 and SW-24 are onboard relays, which connects the common signal to specified IO group 0..3. This must be done, when any IO of any group is used like an input, otherwise it is not possible to activate the input.

```
cm: #<location>:<group>:<signal>
```

Set common signal of specified group of target DIO card.

**Parameters**

location	[int]	Target card location, equals to card position label on the box, first card has location 1. Asterisk (*) can be used to select all cards in the system.
group	[int]	DIO group, 0 to 3. Asterisk (*) can be used to select all groups of the card(s).
signal	[enum]	Common signal to connect: <ul style="list-style-type: none"> <li>• none - common signal disconnected</li> <li>• gnd - ground (0V) connected</li> <li>• v5 - +5V connected</li> <li>• v24 - +24V connected</li> </ul>

**Return value**

No return value.

**Examples**

```
cm: #3: *: v24
Set +24V common signal to all DIO groups of card at location 3.
```

```
cm: *: *: v24
Set +24V common signal to all DIO groups of all cards.
```