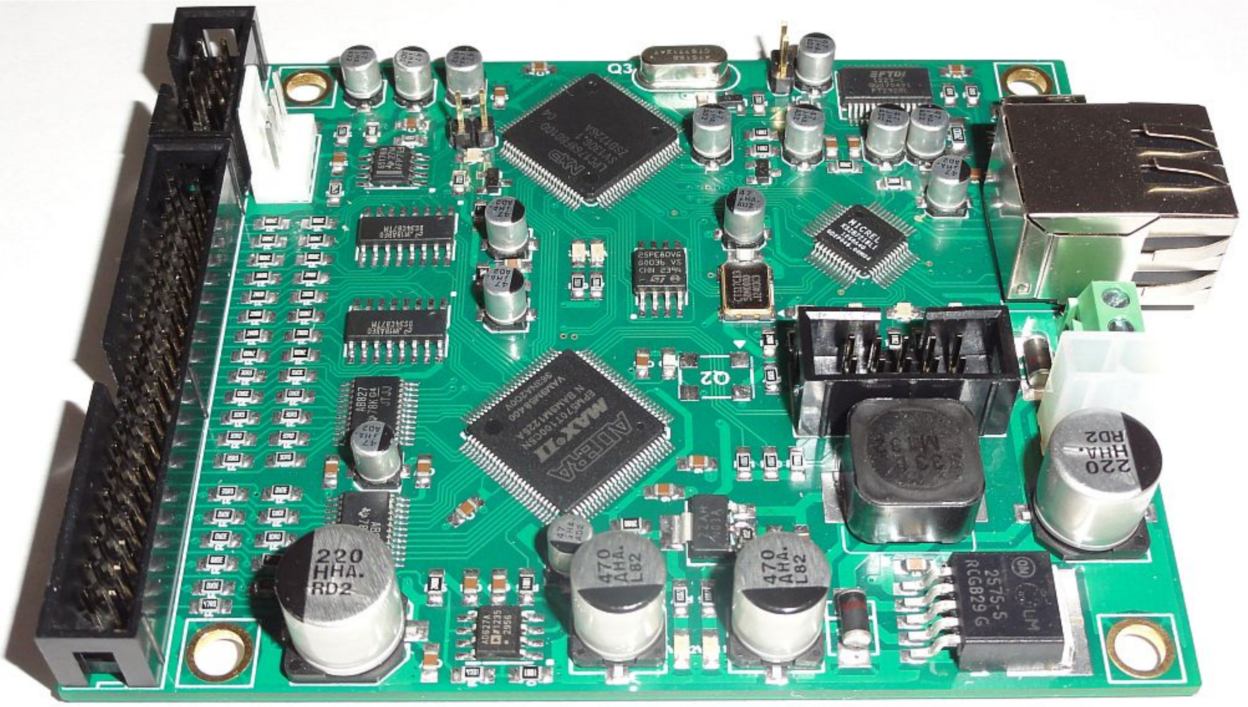
myCNC control & software.

MyCNC-ET3 CNC controller board

User manual (rev 0.01 Preliminary)

last edited 2013-05-20



myCNC
http://www.bevelcutting.com
email: sk@bevelcutting.com
4048 Carling Ave PO Box 72074 Kanata
North, Kanata ON K2K2P4, Canada

Using This Manual

This user manual provides information for proper installation of the myCNC-ET3 controller and operation of machining centers based on myCNC control software.

WARNING: Machinery in motion can be dangerous! It is the responsibility of the user to design effective error handling and safety protection as part of the machinery. myCNC shall not be liable or responsible for any incidental or consequential damages

Table of content

1 Introduction

The myCNC-ET3 is "all-in-one" 4 axes motion controller and PLC controller. The controller communicates with HMI software via the LAN/Ethernet. Performance capability of the controller includes:

- 4 channels STEP/DIR outputs with maximum pulse frequency is 3MHz;
- Motion controller processing time is 82us ;
- Virtual machine processing time for integrated PLC controller and peripheral units (5V TTL outputs, 5V TTL inputs, 5V TTL PWM outputs, 0...10V DAC output) is 1ms;
- 1 Mbytes flash memory for PLC program, motion program and parameters storage for maximum flexibility (more than 100 000 write cycles).

myCNC-ET3 controller drives up tp 4 motor drivers (stepper or servo which accept step/dir signals as inputs). Modes of motion include jogging, point-to-point positioning and contouring. Stand-alone working mode (offline working without computer) is available. Electronic gearing is available through myCNC control software. Several motion parameters can be specified including acceleration and deceleration rates.

For synchronization with outside events, the myCNC-ET3 control board provides peripherals:

- 10x 5V TTL inputs;
- 7x 5V TTL outputs;
- 3x 5V TTL PWM outputs;
- 1 DAC output (0...10V);
- 1x RS485;

Committed digital inputs can be configured as abort, jog, start and home events or can be flexibly used while machining process via PLC controller.

For communication between controller and HMI software specially designed binary full-duplex protocol is used implemented through TCP-IP sockets or UDP datagrams.

To prevent system damage during machine operation, the myCNC software programming interface and myCNC-ET3 controller provide many error handling features. These include software and hardware limits, automatic shut-off on excessive error, user-definable abort input.

2 MyCNC-ET3 Functional units

The myCNC-ET3 circuit can be divided into the following functional elements:
CPU unit – ARM Cortex M3 based microcomputer (512k Flash, 64k RAM) 100MHz ;

- Main processing unit - All-in-one Motion Controller (MC) and Programmable Logic Controller (PLC);
- Motor interface based on Altera FPGA programmable logic;
- 4 channel Stepper motor drivers;
- 3 channel Power PWM, programmable via PLC;
- 10 digital inputs, programmable via PLC;
- 7 digital outputs, programmable via PLC;

- USB slave with integrated USB-to-serial converter for diagnostic/ programming/ configuring;
- LAN/Ethernet controller for HMI connection;

2.1. Main processing unit section

The main processing unit of the myCNC-ET3 is a 32-bit NXP ARM Cortex-M3 series microcontrollers with 512 kbytes internal Flash memory, 1Mbytes external flash memory and 64 kbytes RAM memory. The RAM provides memory for variable storage and application programs. The external flash memory provides non-volatile storage of variables, motion and PLC programs, and arrays. Internal flash memory contains the myCNC-ET3 firmware. On the project official website there is available up-to-date version of firmware to reflash the board.

2.2. Motor Interface

Altera FPGA programmable logic installed on myCNC-ET3 controllers provides high speed motor STEP/DIR motor interface, PWM control, relays and binary inputs control with integrated digital filtering.

2.3. Communication

For communication myCNC-ET3 controllers with HMI (myCNC software) is used LAN/Ethernet interface (10base-T Ethernet port).

For communication myCNC-ET3 controller with the Torch Height Controller myTHC-RU01 is used integrated RS485 bus port (communication speed is 115kbaud).

For programming and configuring the device is used USB slave interface with integrated USB-to-serial converter. Communication parameters for connection is 115200-N-8-1

2.4. myCNC peripherals.

myCNC-ET3 controller board contains number of peripherals (PWM, binary inputs/outputs, DAC output). There are available operations that synchronized with motion and asynchronous operations that run immediately.

All peripherals are available via PLC reserved variables or functions.
In a table below there are PLC variable names and peripherals that assigned to it.

PWM, DAC are unsigned 12-bit values. Complete range for these variables is 0…4095 (in hex 0…0xfff)

Table 1. PLC variables for access to DAC, PWM

PLC variable	Peripherals
pwm01	PWM1
pwm02	PWM2
pwm03	PWM3
pwm04	Reserved for PWM4
dac01	DAC

Digital inputs available via PLC function:

```
getport(int port_number)
//read digital input nr. (port_number+1); return value is 0 if
input pin open, 1 if closed;
```

Digital outputs available via PLC functions:

```
portset(int port_number); //turn on relay nr. (port_number+1)
portclr(int port_number); //turn off relay nr. (port_number+1)
```

Samples:

1. Turn on relay nr.3 and turn off relay nr.1
   ```
   portset(2); // turn on relay output nr.3
   portclr(0); // turn off relay output nr.1
   ```
2. if input pin nr.5 is open turn off PWM channel nr. 1, otherwise turn on it on a half of maximum;
   ```
   if (portget(4)==0)
   pwm=0;
   else
   pwm01=2048;
   ```
3. PWM channel nr. 2 repeats value on ADC channel nr.1
   ```
   pwm02=dac01;
   ```

2.5. System Elements

The myCNC-ET3 is part of a motion control system which includes Computer control (PC compatible, Android tablet or Embedded Linux system) with installed myCNC software, servo or stepper motor drivers, motors and power supply.

3 myCNC-ET3 connection.

3.1. MyCNC-ET3 outline.

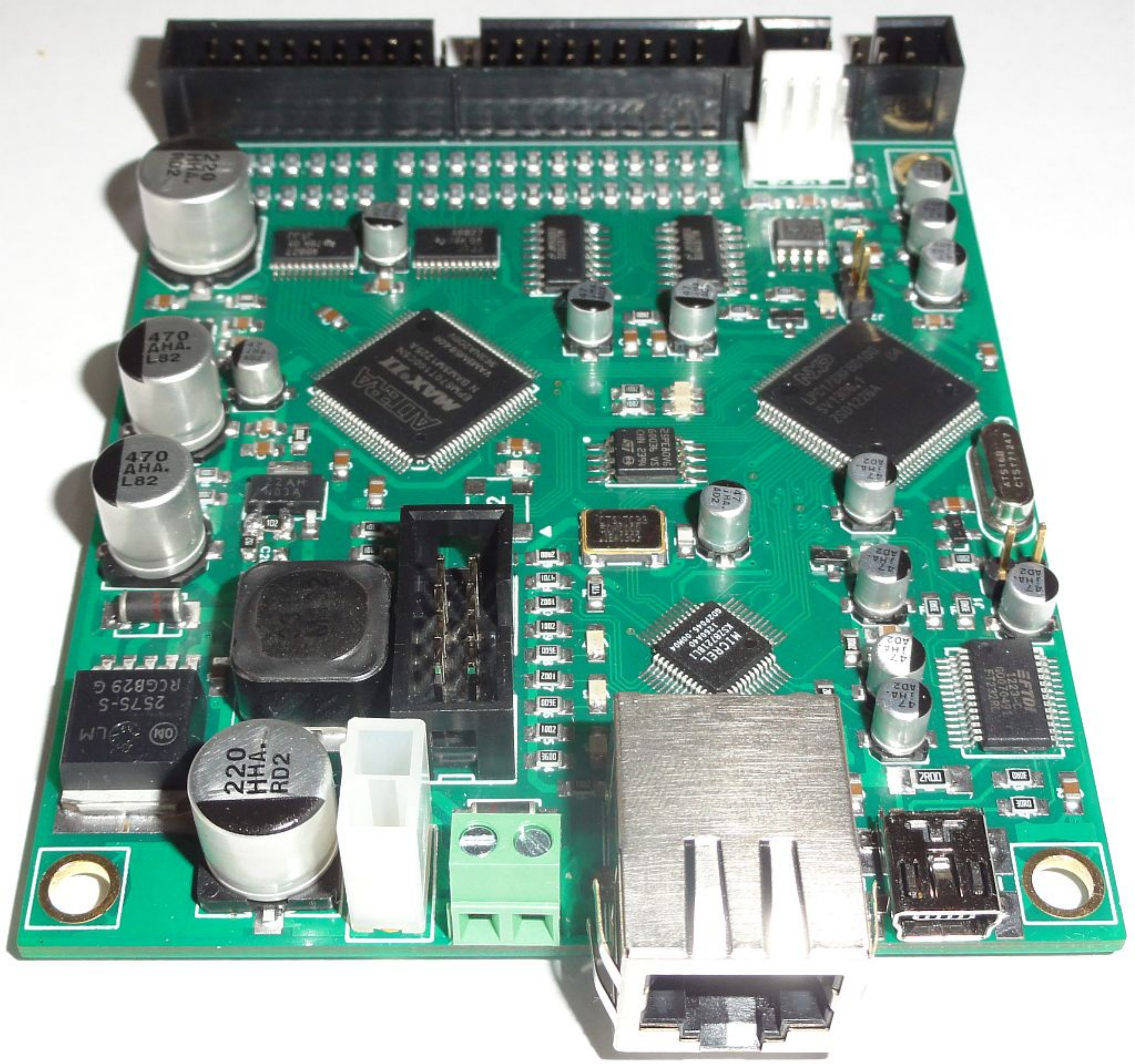
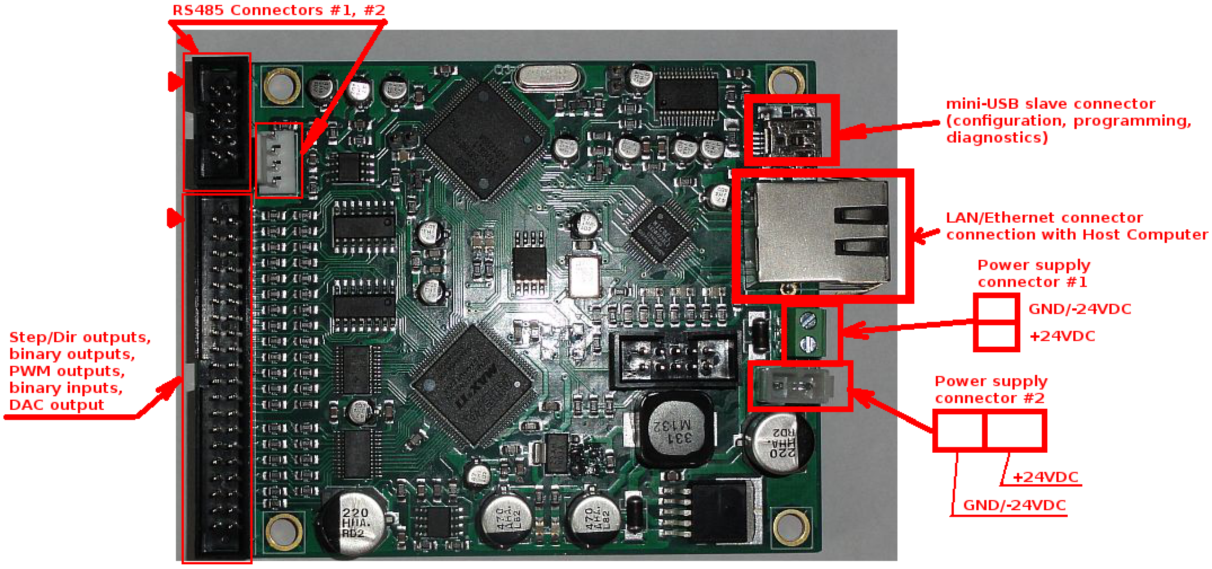
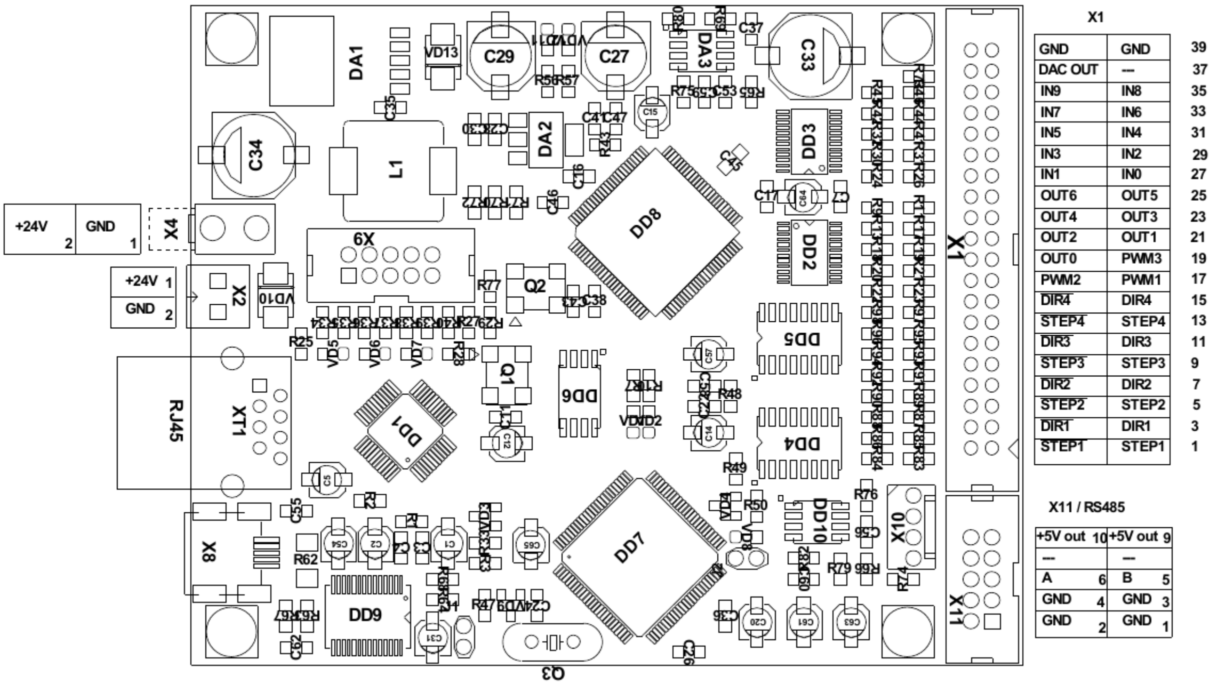


Figure 1. myCNC-ET3 board (revision 1).



RS485 Connectors #1, #2
mini-USB slave connector (configuration, programming, diagnostics)
LAN/Ethernet connector connection with Host Computer
Power supply connector #1
GND/-24VDC
+24VDC
Step/Dir outputs, binary outputs, PWM outputs, binary inputs, DAC output
Power supply connector #2
+24VDC
GND/-24VDC

Figure 2. myCNC-ET3 (rev 1) board outline



X1

GND	GND	39
DAC OUT	---	37
IN9	IN8	35
IN7	IN6	33
IN5	IN4	31
IN3	IN2	29
IN1	IN0	27
OUT6	OUT5	25
OUT4	OUT3	23
OUT2	OUT1	21
OUT0	PWM3	19
PWM2	PWM1	17
$\overline{DIR4}$	DIR4	15
$\overline{STEP4}$	STEP4	13
$\overline{DIR3}$	DIR3	11
$\overline{STEP3}$	STEP3	9
$\overline{DIR2}$	DIR2	7
$\overline{STEP2}$	STEP2	5
$\overline{DIR1}$	DIR1	3
$\overline{STEP1}$	STEP1	1

X11 / RS485

+5V out 10	+5V out 9
---	---
A 6	B 5
GND 4	GND 3
GND 2	GND 1

Figure 3. myCNC-ET3 board outline/pin description.

Table 2. myCNC-ET3 peripherals. Address-to-pin assosiations.

Software name	Connector	Pin numbers	Comments
Binary outputs (5V TTL)			
OUT 0	X1	20	
OUT 1	X1	21	
OUT 2	X1	22	

OUT 3	X1	23	
OUT 4	X1	24	
OUT 5	X1	25	
OUT 6	X1	26	
Binary inputs (5V TTL)			
IN 0	X1	27	
IN 1	X1	28	
IN 2	X1	29	
IN 3	X1	30	
IN 4	X1	31	
IN 5	X1	32	
IN 6	X1	33	
IN 7	X1	34	
IN 8	X1	35	
IN 9	X1	36	
PWM outputs			
PWM 1	X1	17	
PWM 2	X1	18	
PWM 3	X1	19	
Motor channels			
Motor 1 (Axis 1)			
STEP 1	X1	1	
~STEP 1	X1	2	
DIR 1	X1	3	
~DIR 1	X1	4	
Motor 2 (Axis 2)			
STEP 2	X1	5	
~STEP 2	X1	6	
DIR 2	X1	7	
~DIR 2	X1	8	
Motor 3 (Axis 3)			
STEP 1	X1	9	
~STEP 1	X1	10	
DIR 1	X1	11	
~DIR 2	X1	12	
Motor 4 (Axis 4)			
STEP 1	X1	13	
~STEP 1	X1	14	
DIR 1	X1	15	
~DIR 2	X1	16	

3.2. Power supply.

The board contains two power connectors for 24V DC power supply X2, X4. Any of connector can be used (it's parallel connected in the board).

Table 3. myCNC-ET3. Power supply. X2 pin description.

Pin nr.	Description
1	24V DC
2	GND / COMMON

Table 4. myCNC-ET3. Power supply. X4 pin description.

Pin nr.	Description
1	GND / COMMON
2	24V DC

3.3. Motor Interface.

myCNC-ET3 board contains 4 pulse/dir channels to control motor drivers. All motor channels wires are connected to X1 connector. Motor channel pins X1 pins are described on a table below.

Step/Dir motor interface is made with 5V linear driver chip DS34C87 (RS422 compatible).

Table 5. myCNC-ET2 step/dir pins connection.

Pin name	Connector	Pin number	Comments
Motor channels			
Motor 1 (Axis 1)			
STEP 1	X1	1	
~STEP 1	X1	2	
DIR 1	X1	3	
~DIR 1	X1	4	
Motor 2 (Axis 2)			
STEP 2	X1	5	
~STEP 2	X1	6	
DIR 2	X1	7	
~DIR 2	X1	8	
Motor 3 (Axis 3)			
STEP 1	X1	9	
~STEP 1	X1	10	
DIR 1	X1	11	
~DIR 2	X1	12	
Motor 4 (Axis 4)			
STEP 1	X1	13	
~STEP 1	X1	14	
DIR 1	X1	15	
~DIR 2	X1	16	

Schematic design of step-dir motor intefrace is shown in a picture below.

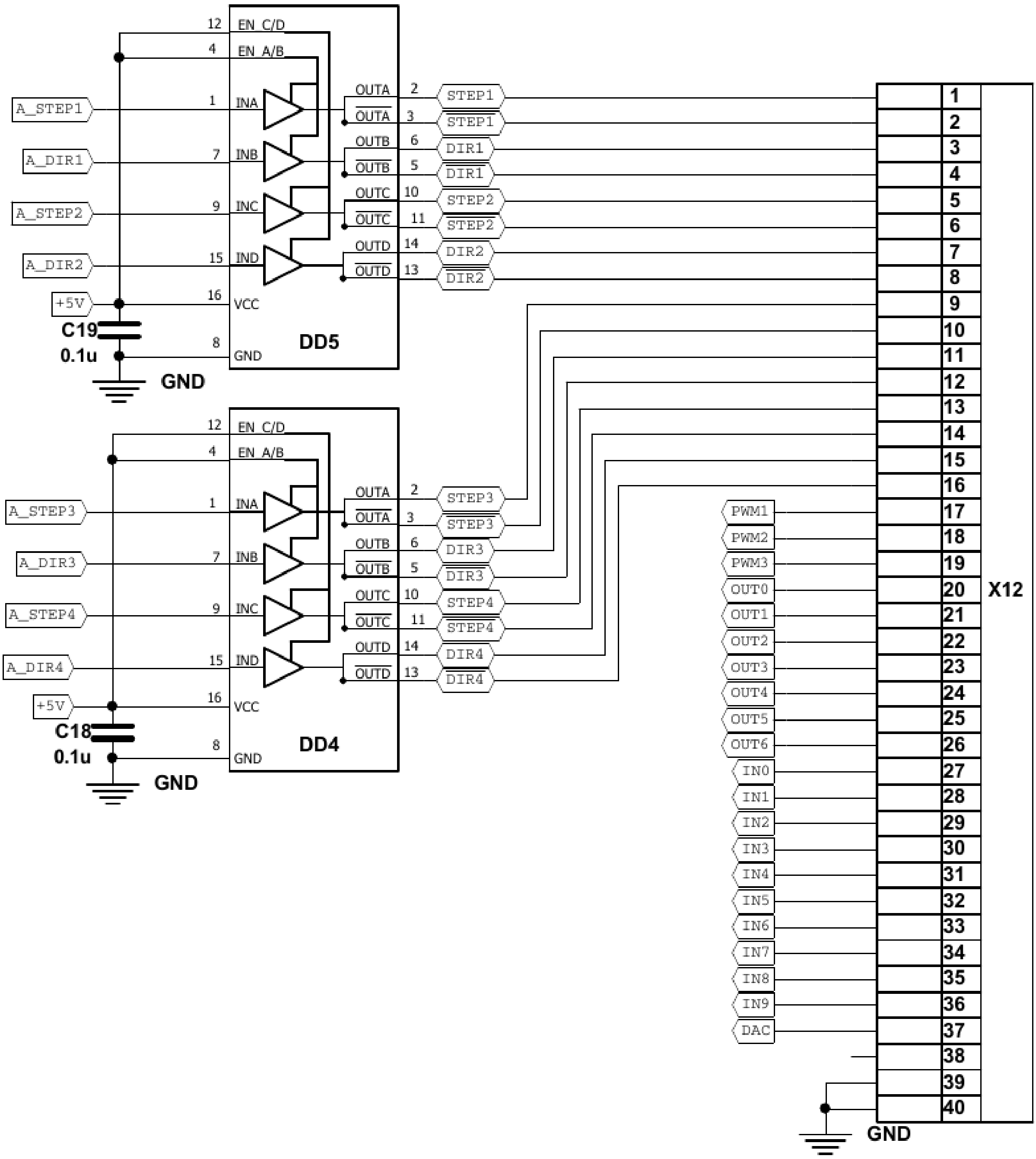


Figure 4. STEP/DIR interface schematic design for myCNC-ET3 control board.

3.4. Binary inputs.

The myCNC-ET2 contains 10 binary inputs (TTL 5V compatible) for connection any type of sensors, keys, switchers.

Inputs connected to X2 connectors. X2 connector pin description is shown on a Table 2.

3.5. PWM outputs, DAC (Spindle speed control) output.

The myCNC-ET3 board contains 3 channels of PWM outputs (5V TTL compatible) and 1 channel 0...10V DAC output. The outputs connected to X2 connector. X2 pin description is shown on a table 2.

3.6. RS485 connectors.

ET3 control board contains RS485 bus connector. RS485 might be configured as Modbus-ASCII server, IPG fiber laser interface, THC board interface.

Connector X11 (X10) is used for RS485. X11,X10 connectors schematic design is shown on a picture below.

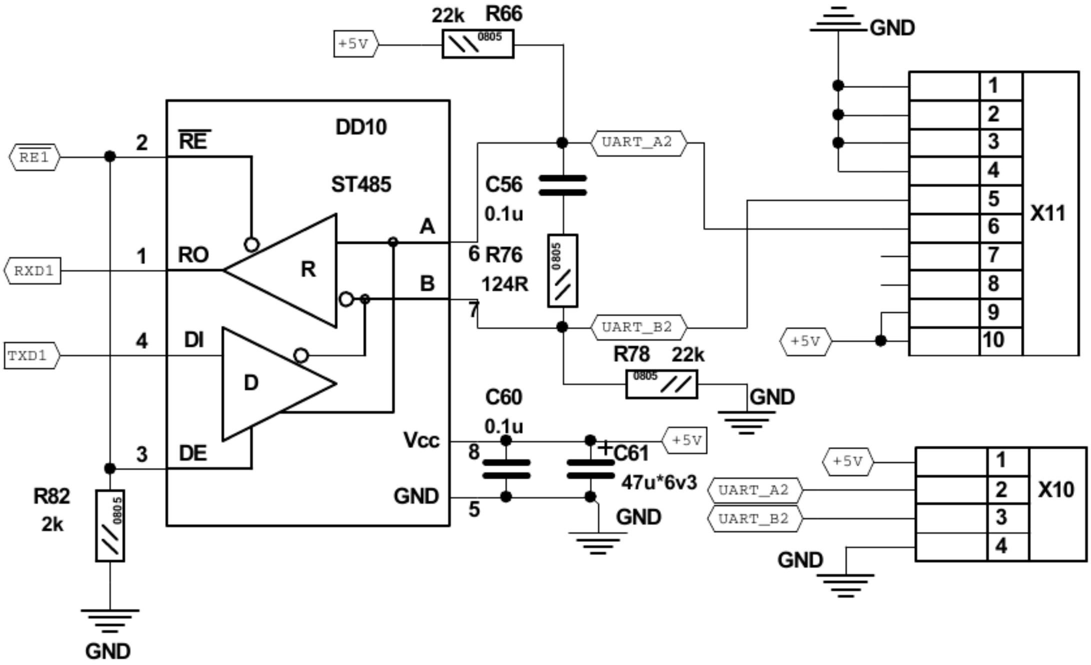


Figure 5. RS485 connectors (X11, X10) schematic design.

Pin description for X11 connector is shown below.

Table 6. ET3-RS485 X31, X32 pin description.

Pin nr.	Description
1, 2, 3, 4	GND/COMMON
5	RS485 - B
6	RS485 - A
7, 8	Not used
9, 10	+5V DC output

3.7. Connection to the control PC.

MyCNC-ET3 control board is connected to PC HMI through LAN/Ethernet connector XT1. Standard Ethernet cable can be used for connection. The control board may be connected either directly to the control PC or through Ethernet Switch/HUB to Local Network.

For reflashing, configuration and diagnistics is used USB slave connector X8.

3.8. MyCNC-ET1 (ET3) connection examples.

3.8.1 Motor interface connection examples.

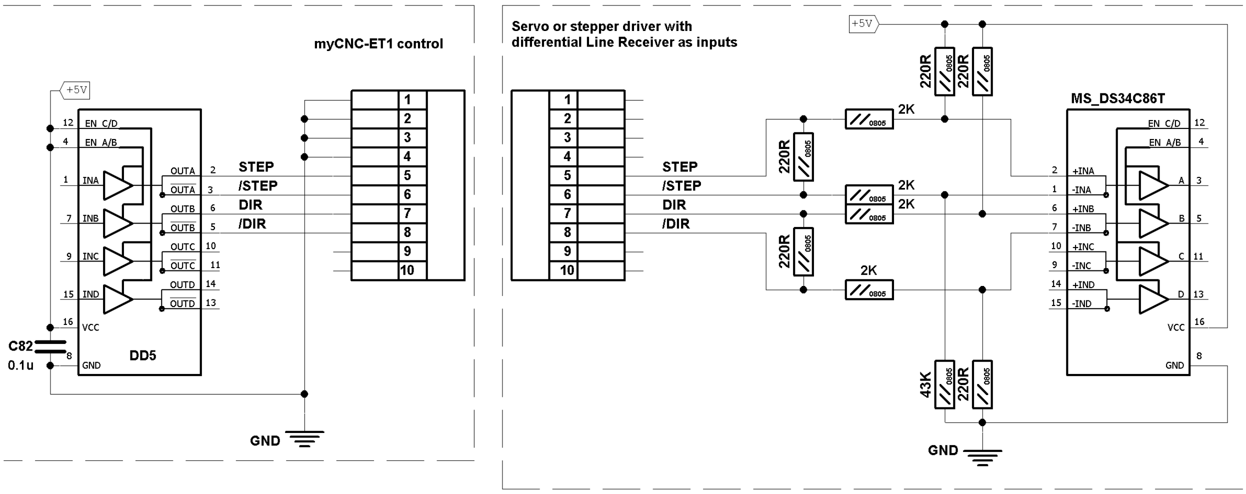


Figure 6. Connection servo or stepper driver with differential line driver as inputs.

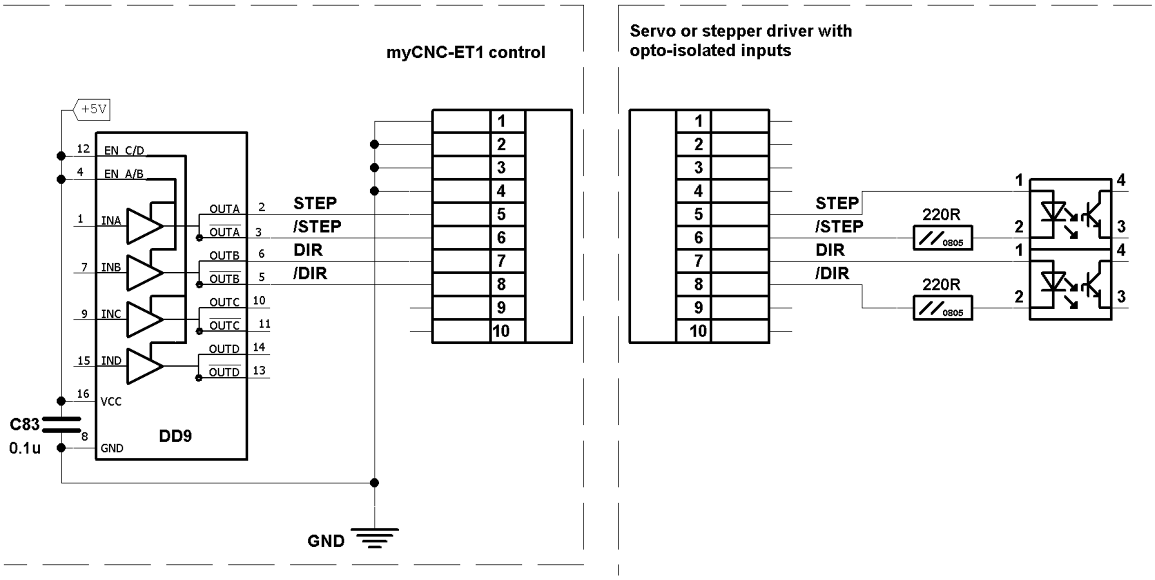


Figure 7. Connection servo or stepper driver with opto-isolated inputs.

4 Configuration and diagnostic.

4.1. Connection the myCNC-ET3 controller to diagnostic channel.

For programming, reflashing of configuring the ET3 controller should be connected to PC through USB-slave connector.

USB-to-serial converter chip FT232RL is installed on the ET3 controller for serial communication with Host PC. Drivers for this converter should be installed on the Host. Driver for Linux OS is called «ftdi_sio» and included into the Linux Kernel.

For Windows machines the drivers can be installed from myCNC folder:
myCNC/driver.usb-to-serial

or directly from the manufacturers website:
http://www.http://www.ftdichip.com/Drivers/D2XX.htm

Terminal software like «Hyperterminal» (MS Windows), «TeraTerm» (MS Windows), «minicom» (Linux) can be used to communicate with myCNC-ET3 control board.

Serial connection is used with parameters 115200 8N1.

4.2. PC to driver command format.

Command to the driver is text line that begins with symbol "#" and ends with <CR/LF>. Text line is case sensitive. There are available Set, Print and Debug commands.

4.3. Set commands.

Table 7. Set LAN/Ethernet commands (myCNC-ET1/ET3 controllers).

Command format.	Command description.
#SLE {A}<CR/LF> Possible value for {A} is 0,1	[S]et [L]AN [E]nable – turn on/off LAN/Ethernet interface. Automatically save settings in flash memory. Change will take effect after restart the driver. {A} – "1" LAN enable {A} – "0" LAN disable Examples: #SLE 0 turns LAN interface off; #SLE 1 turns LAN interface on
#SLA {A} {B} {C} {D}<CR/LF> Possible value for {ABCD} are 0...255	[S]et [L]AN [A]ddress – set LAN/Ethernet address of the board as given values ABCD: IP Addr ={A}.{B}.{C}.{D} Automatically save settings in flash memory. Change will take effect after restart the driver.

	Example: #SLA 192 168 4 78<CR/LF> sets IP address to 192.168.4.78
#SLG {A} {B} {C} {D}<CR/LF> Possible value for {ABCD} are 0...255	[S]et [L]AN [G]ateway – set LAN/Ethernet gateway as given values ABCD: GW Addr ={A}.{B}.{C}.{D} Automatically save settings in flash memory. Change will take effect after restart the driver. Example: #SLG 192 168 4 1<CR/LF> sets GW address to 192.168.4.1

Table 8. Print command (myCNC-ET1/ET3 controller).

Command format.	Command description.
#PP <CR/LF>	[P]rint [P]ID regulator data. Print current PID coefficients Example: #PP

Table 9. Debug commands (myCNC-ET1/ET3 controller).

Command format.	Command description.
#DP {A} <CR/LF> Possible value range for {A} is 0...3	[D]ebug [P]ID regulator level. Turn on PID control diagnostic with given level or turn off it Give value- 0 – turn off diagnostic; 1 – print position error, speed; 2 – print position error, motor voltage, speed; 3 – print position error, speed error, motor voltage, speed; Examples: #DP0 turn off diagnostic #DP1 turn on PID diagnostic with level 1

5 Updating firmware for myCNC-ET3 control board.

At the moment reflashing utility is available for Linux and MS Windows operationg systems.

For Linux: ftdi_sio module should be active on the system for USB-to-serial converter FT232, installed on the board.

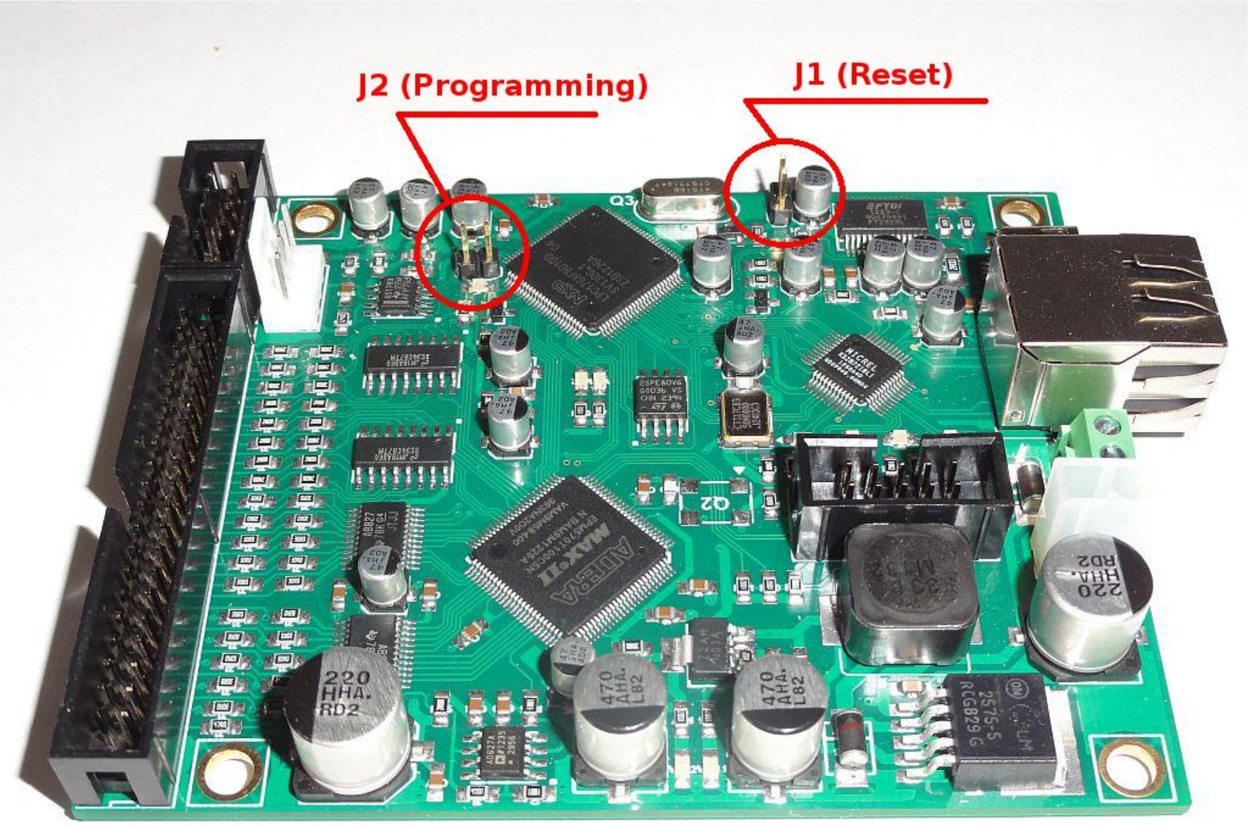
For Windows: FT232RL USB to serial converter drivers should be installed on the system. The driver can be found either in myCNC software ditributive package or downloaded directly from FTDI web site (try to Google "ftdi chip ft232"). For Windows 7 the driver will be downloaded and installed automatically by Windows just after the board is connected.

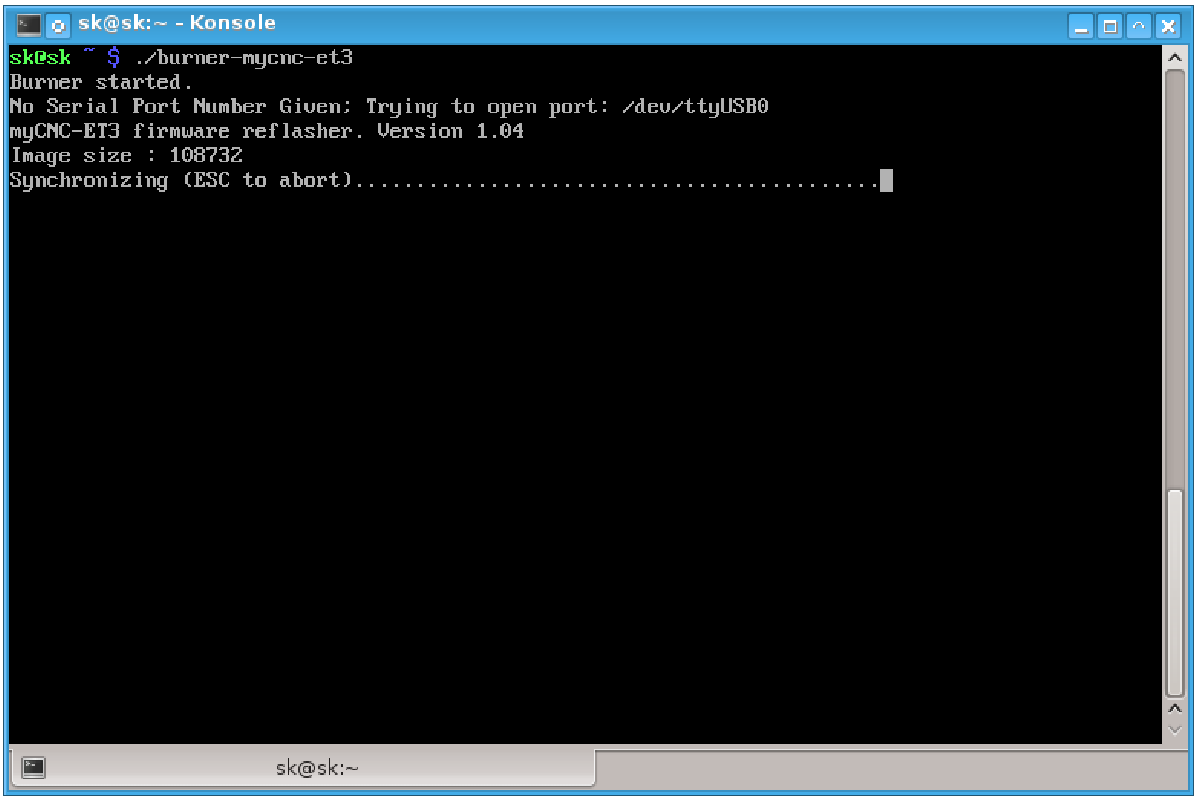
To update firmware:

1. Host PC should be connected to the Internet.
2. Download firmware burner utility:
   For Linux: **burner-mycnc-et3.tar.gz**
   For Windows: **burner-mycnc-et3-win.zip**
3. Extract file from archive -

   for Linux — downloaded file is tar.gz archive. To extract files from archive use command -
   tar zxf burner-mycnc-et3.tar.gz
   User-friendly Linux distributives (like Ubuntu) may help to extract files from archive with 1-2 mouse click.

   for Windows — downloaded file is zip archive. Use internal Windows tools (assigned with right mouse button) to extract the files.
4. Power up the board (+24V DC power supply) and connect with Host PC via USB.
5. Close Jumpers J1 (Reset) & J2(Programming) on the board.





```
sk@sk ~ $ ./burner-mycnc-et3
Burner started.
No Serial Port Number Given; Trying to open port: /dev/ttyUSB0
myCNC-ET3 firmware reflasher. Version 1.04
Image size : 108732
Synchronizing (ESC to abort)..........................................
```

6. Run the utility.

   burner-mycnc-et3 utility is recommended for UDP comminication interface (MS Windows, Mac OS, Linux, Embedded Linux, Android operating systems);

If you see message "Synchronizing (ESC to abort)......", then you have about 5 seconds to open J1 jumper.
Reflash process should start.
After the process is finished, jumper J2 jumper should be opened and the board restarted (power off, wait 2 seconds, power on).

6 RS485 Modbus interface.

6.1. Introduction.

MyCNC-ET3 control board provide RS485 Modbus interface for access to Peripherals and Motion controller features of the controller.

Modbus ASCII protocol through RS485 is implementer in myCNC-ET3 control board. Physical specification of connection is 115200n81

Modbus server service is implementer in ET3 by default. Optional client features can be activated through Ethernet connection with myCNC control software.

Implemented commands listed in tables below.

Table 10. RS485 Modbus IO access commands (myCNC-ET1/ET3).

	Description	Value, range	Address
20	Read binary inputs.	16-bit value	0x800
21	Read/Write binary outputs (relay & open collector outputs)	16-bit value	0x804
22	Read ADC values (ADC1... ADC4).	16-bit value (low 12 bit is actual ADC value)	0x808, 0x809, 0x80A, 0x80B
23	Read/Write DAC outputs value (DAC1, DAC2).	16-bit value (low 12 bit is actual avlue)	0x80C, 0x80D
24	Read/Write PWM outputs value(PWM1... PWM4).	16-bit value (low 12 bit is actual avlue)	0x810, 0x811, 0x812, 0x813

Table 11. RS485 Modbus Positioning commands — Motion control access through Modbus (myCNC-ET1/ET3).

	Description	Value, range	Address
1	Motion speed set (pulses per sec), 32-bit format : 24.8	32-bit value, (0x814 — hi word, 0x815 — lo word)	0x814, 0x815
2	Acceleration set for motion command (pulses per sec^2), integer value	32-bit value, (0x816 — hi word, 0x817 — lo word)	0x816, 0x817
3	Increment for axis X	32-bit value, (0x820 — hi word, 0x821 — lo word)	0x820, 0x821
4	Increment for axis Y	32-bit value, (0x822 — hi word, 0x823 — lo word)	0x822, 0x823
5	Increment for axis Z	32-bit value, (0x824 — hi word, 0x825 — lo word)	0x824, 0x825
6	Increment for axis A	32-bit value, (0x826 — hi word, 0x827 — lo word)	0x826, 0x827
7	Increment for axis B	32-bit value, (0x828 — hi word, 0x829 — lo word)	0x828, 0x829
8	Increment for axis C	32-bit value, (0x82A — hi word, 0x82B — lo word)	0x82A, 0x82B
9	Start motion in given axes.	16-bit value (6 bit binary fields). «1» in the fiels means moving in the given axis, «0» - ignore increment, no motion. Bit 0 — axis X Bit 1 — axis Y Bit 2 — axis Z Bit 3 — axis A Bit 4 — axis B Bit 5 — axis C	0x830