ED-Nano-USB8PRMx and USB8PR V2

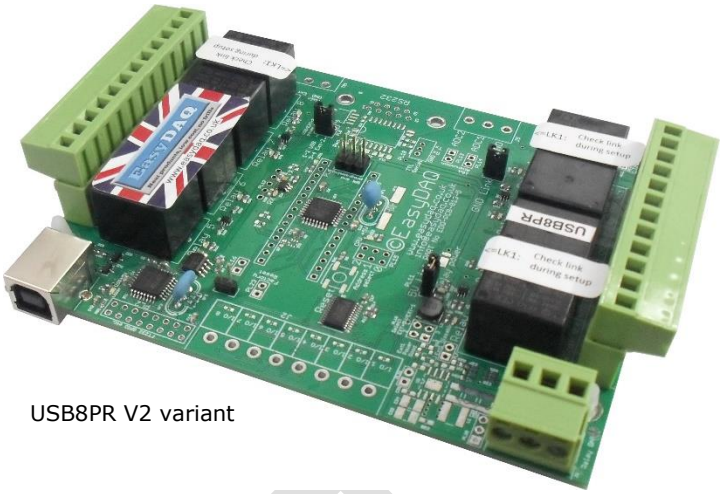
Low cost Data Acquisition & Control products

8 channel reprogrammable relay card with 2 ADCs

Product Datasheet 61

Features

- Re-programmable ATMega323P processor supplied with EasyDAQ compatible firmware.
- Low cost, high density, small profile.
- Available with 5V, 6V, 12V or 24V operating relays as required.
- 8 Power Relays are SPDT, Form C, changeover type, with N/O, COM and N/C contacts.
- Edge mounted screw terminal block access to N/O COM & N/C relay contacts of the 8 on board relays.
- USB control and programming as standard with Serial RS232, RS485, I2C and SPI options.
- LED status indicators for USB power/ connection, I/O and relay activation status.
- 8 protected 5V DIO ports with ESD protection to +/-30kV and over/under voltage protection to +17V and -25V.
- 2 protected Analog to Digital converter inputs with hardware selectable 10:1 input attenuation and software selectable Vcc or 1.1V reference voltages to enable measurement up to +50V.
- PCB tracking (& power relays) are designed to handle 10 amps @ 240V AC, or 8A @ 30V DC (switched or continuous)
- Supplied with nylon feet. Clear Perspex cover & base option available
- DIN rail base clip (with Perspex cover) mount option also available
- CE, RoHS



USB8PR V2 variant

Description

A relay card with 8, 10A, 240V relays plus 8 protected DIO ports and 2 protected Analog to Digital converter inputs.

The ED-Nano-USB8PRMx card has been designed around the ATMega328 processor so it can easily be re-programmed by a developer to be a standalone system or to add special functions.

The card has many power supply options including regulators and isolated earth planes between the communications interface and the relay functions.

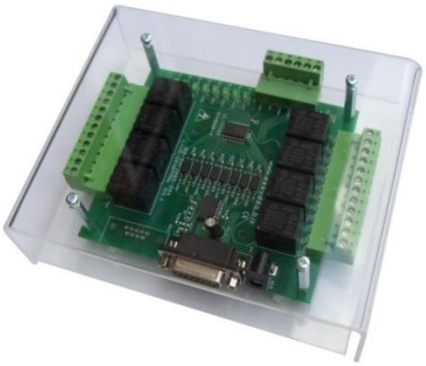
Relays are single pole changeover type, capable of switching 240VAC @ 10Amps. PCB tracking is designed to handle 10 Amps. Fitted with PSU & relay LED status indicators.

The 1- or 2-part screw terminal blocks give access to N/O, COM & N/C relay contacts, the channel input control signals and the 0V/5V DC supply terminals.

The processing and communication sections of the card can operate solely a USB port or from an external supply of either 5V DC or+7V to +15V DC via a on-board regulators. The relays can be powered from the processing and communications side or if full isolation is required then the relays can be powered via an external voltage or +7V to +15V DC via their own on-board regulator.

Both the on-board regulators can also be used to power external hardware.

A stripped-down variant of this card known as the "USB8PR V2" is designed to replace the USB8PR. See the USB8PR variant section below.



SEQ8PR2 + COVER8PRMx

Specifications

Operating temp range
0-70^{0}C

Power
See Power Options section.

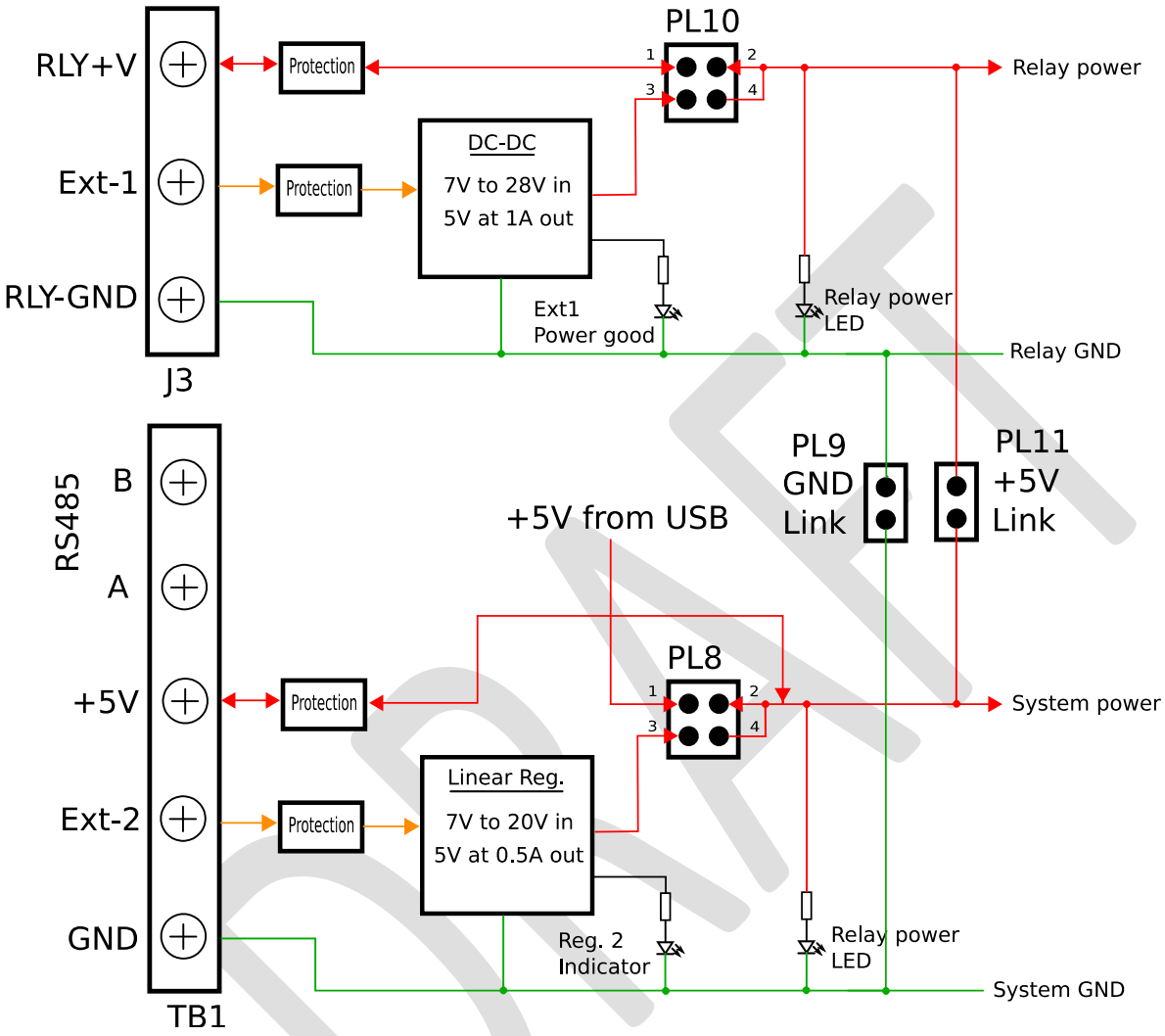
Relays
See: *Specifications: Relays* table

Dimensions
Dimensions 100mm (D) 130mm (W) 30mm (H) (inc. feet & 2 part

conns), Weight 120g..

Power Options:

The ED-Nano-USB8PRMx has many powering options. The design of the ED-Nano-USB8PRMx also allows for separation of the relay operating power from the processor and communications by means of 2 links separating the 0V and +5/(relay control voltage).



As Supplied, the ED-Nano-USB8PRMx can be driven directly from a USB port. However, if the relays are 5V operating and all switched on then the total current of the board will be up to 400mA. While most USB ports can supply sufficient current to drive all 8 relays it should be considered as a less than optimal long term solution.

For long term use it is recommended that the DC-DC converter or an external supply on EXT-1 is used to power the relays.

The Linear regulator is used when the board is not connected to power via the USB or is the ADC requires a more accurate supply.

The on-board power supplies can also be used to drive external circuitry as long as the overall current consumption of each supply is not exceeded.

Both power supplies use built in thermal overload protection.

It is recommended that careful attention be paid to the setup of the power supplies to ensure reliable long term operation.

Please contact EasyDAQ for help or more information if required.

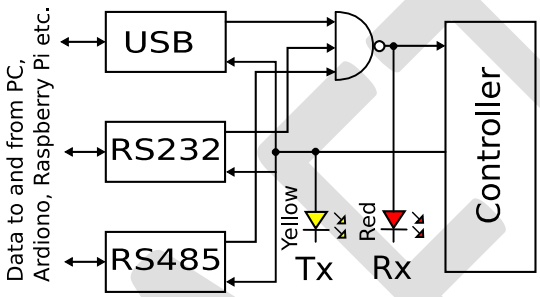
Communication ports:

The ED-Nano-USB8PRMx can be controlled though the USB, RS232 and RS485 ports.

The incoming data is "OR" wired so that all 3 channels can communicate with the board at any time. However, there is no flow control so it is possible that signals can become confused.

Outgoing data is paralleled so that all 3 channels receive the same data.

Representation of the signals circuit



All 3 ports are optional and can be omitted to reduce costs on volume orders.

The I2C bus is currently set up as a master so it cannot receive control signals. See "I2C Examples" below for control of external I2C peripherals.

Communications though the SPI port of the ATMega328 has not yet been implemented. Please contact EasyDAQ if this is a requirement.

Protected DIO port:

The DIO port on J2 is connected to Port 1 of a NEC PCAL9555A port expander. Each channel of the 8 bit port is configurable as input or output. Inputs can have pull up or pull down resistors enabled independently.

See commands Dx, Ex, Fx, Gx and Px.

The each channel of the DIO port is protected by a diode network and a 1k ohm resistor. This protection should make inputs safe up to +15 and -10V.

In output mode, each channel of the DIO is limited by the protection circuitry to +/- 5mA at 5V.

Analog to digital converters:

2 of the ADC channels of the ATMega328 are made available on J5. The ADCs are controlled through the use of commands "QT" for ADC 1 and "Qt" for ADC 2.

The ATMega328 ADC is 10bits wide so the value is returned as an ASCII string terminated by CR and LF characters.

The reference for the ADC is set to a default of the processors Vcc. This can be a noisy reference signal particularly when the board is powered through just the USB port.

There are 2 ways to improve the noise level on the reference voltage to the ADC.

1. Use an external power supply through the linear regulator connected to TB1. This will give a better 5V signal to the ADC via the processors Vcc.

2. Switch the ADC to the ATMega328's 1.1V internal reference. This is done using commands "SM" for Vcc reference voltage and "Sm" for the 1.1V reference. Note that the 1.1V reference will limit the measurable range to 0V to 1.1V full scale.

The ADC channels also each have a hardware selectable 10:1 attenuator. Linking PL14 for ADC1 and PL13 for ADC 2 will attenuate the input voltage by 10:1.

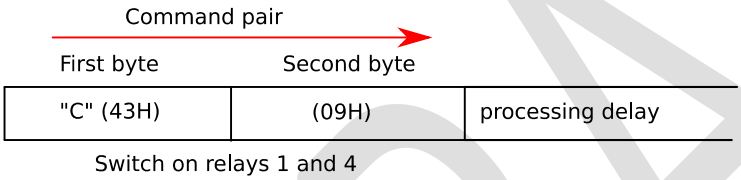
Voltage measurement ranges on the ADC wth respect to the reference and the attenuator:

Reference/Attenuator	1:1	10:1
5V (Vcc)	0V to 5V	0V to 50V
1.1V	0V to 1.1V	0V to 11V

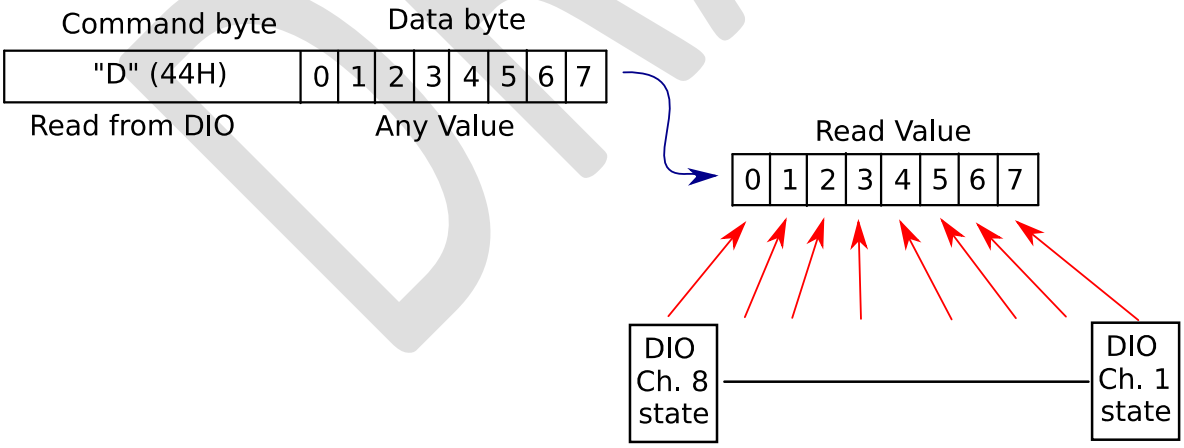
Standard firmware commands:

The card is commanded via simple single ASCII characters (+ status byte). I.e a 2 byte pair. These are commands that address each port of the PIC processor device (Hex equivalent shown in brackets). The card can be controlled using a Terminal emulator if connected via USB, RS232 or RS485 – see *Example 2.* below.

Example 1. Switch on relays 1 and 4:

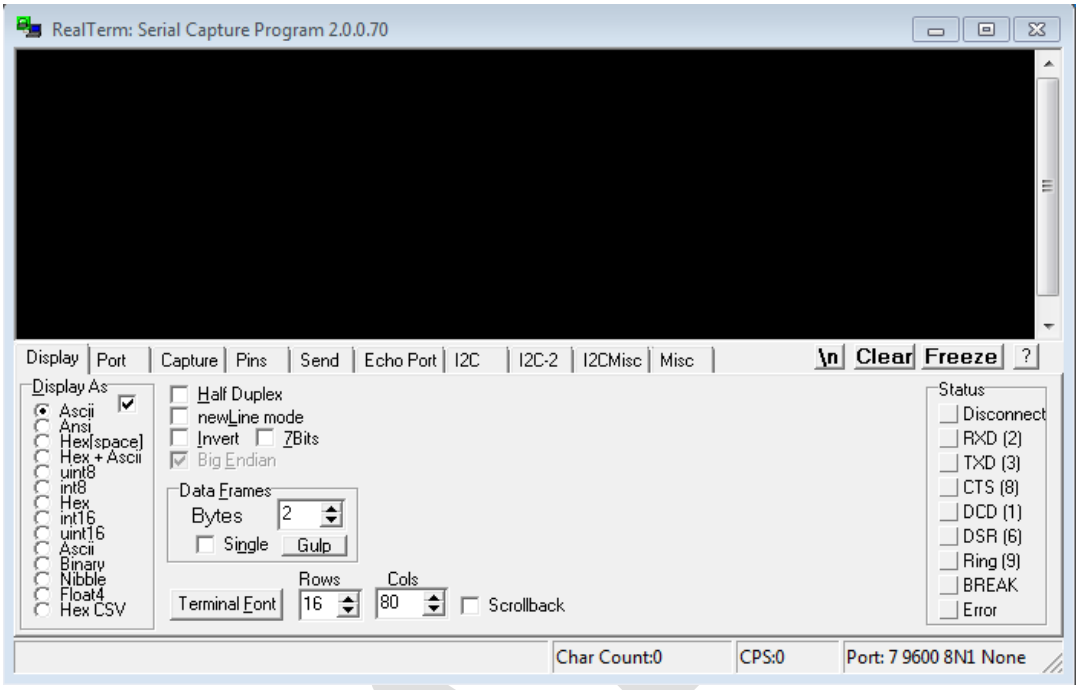


Example 2. Read data from the DIO:

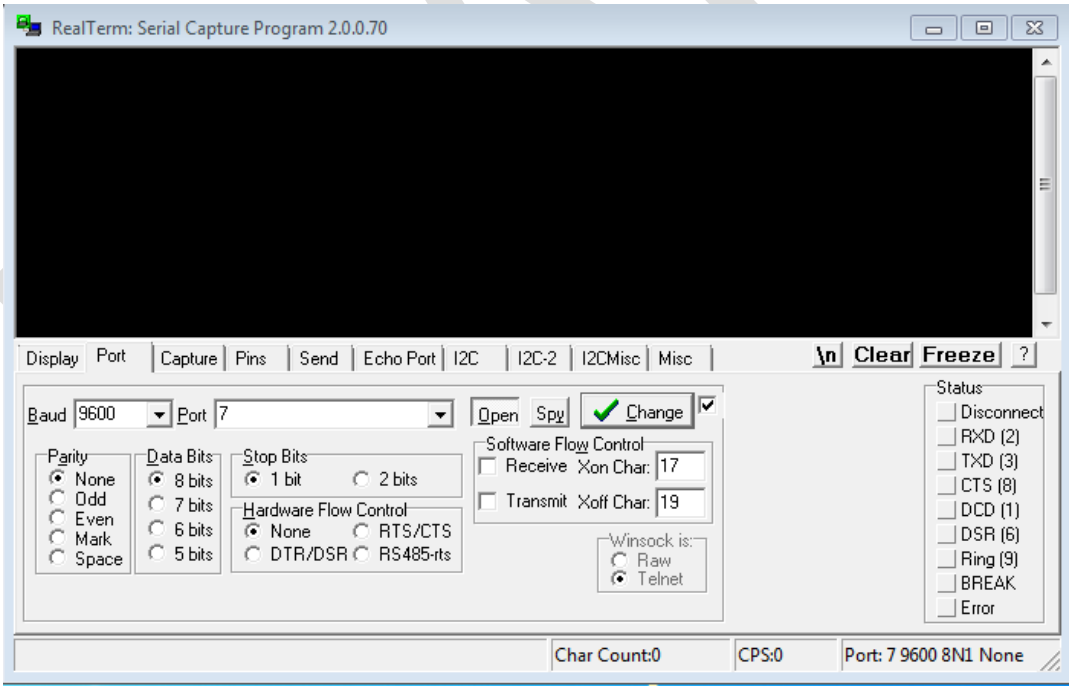


Data is return as a single byte except where specified (i.e. Read ADC commands return an ASCII string)...

Example 3. Using the "RealTerm" serial capture program on a Windows PC to switch on some relays. Plug your ED-Nano-USB8PRMx into a USB port and run up RealTerm.



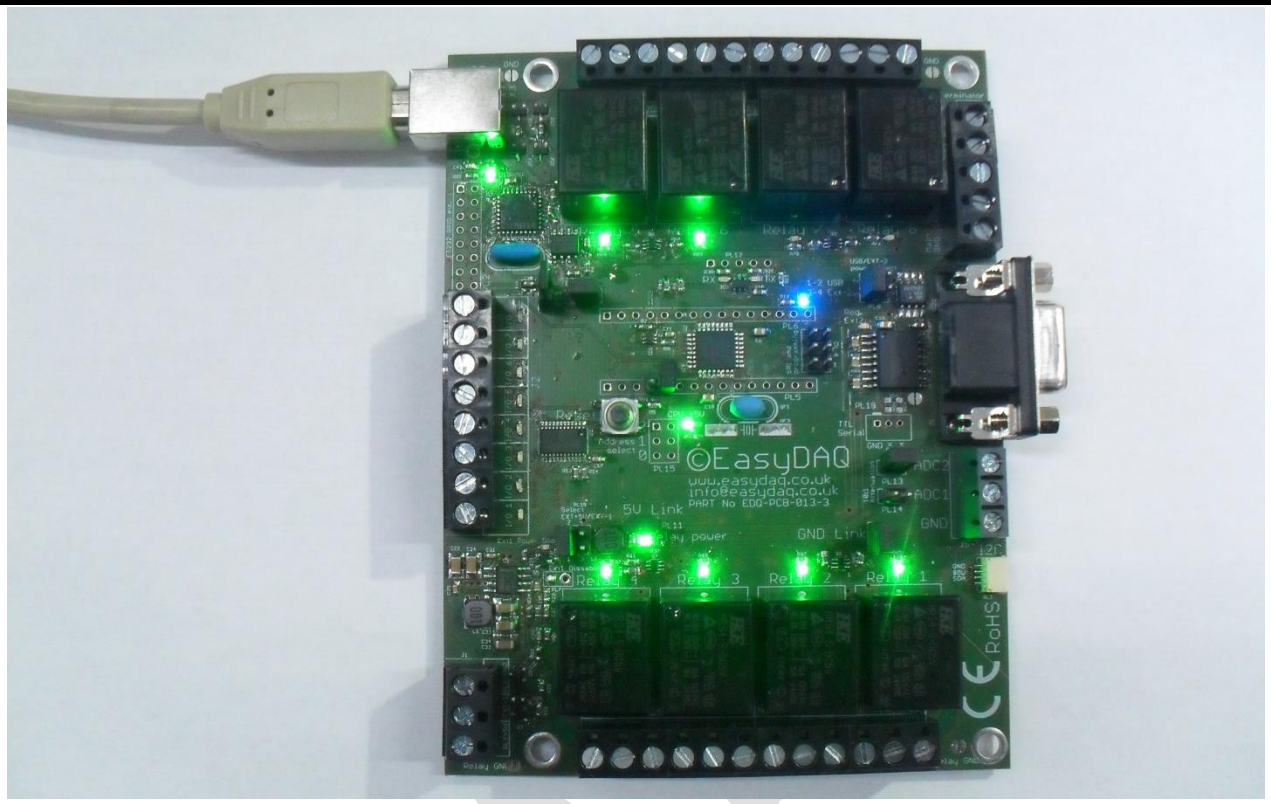
Select the port that your ED-Nano-USB8PRMx is plugged into using the "Port" tab:



Set the baud rate to 9600 (default for this board) and click on 

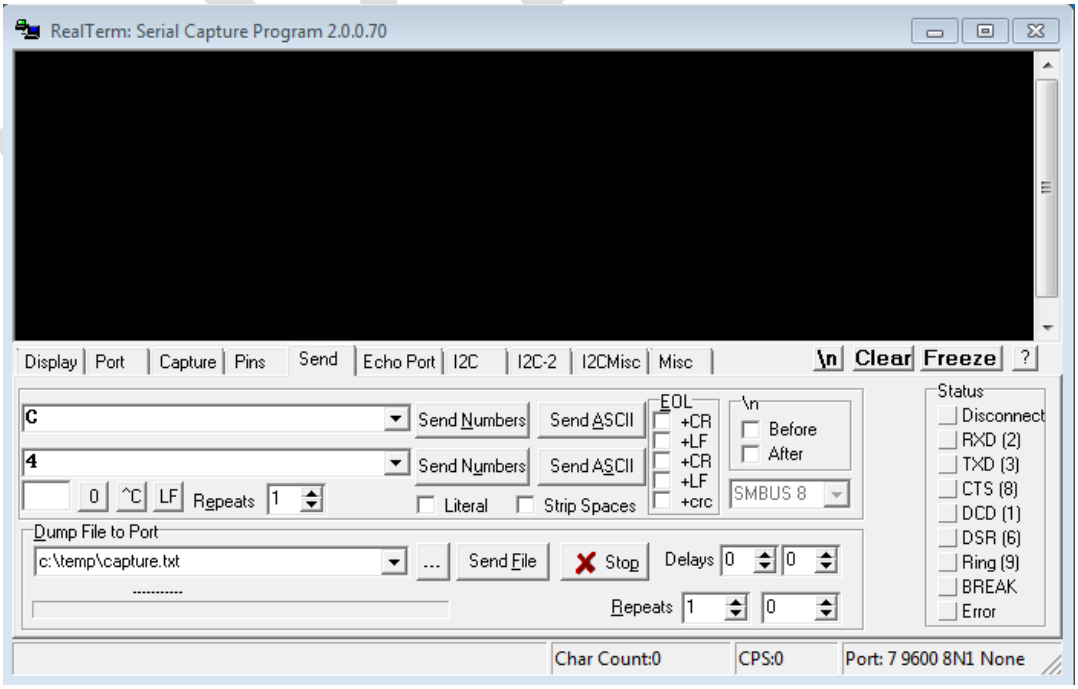
.

Click into the black display area and type "C?" without the quotes... Relays 1 to 6 should switch on.

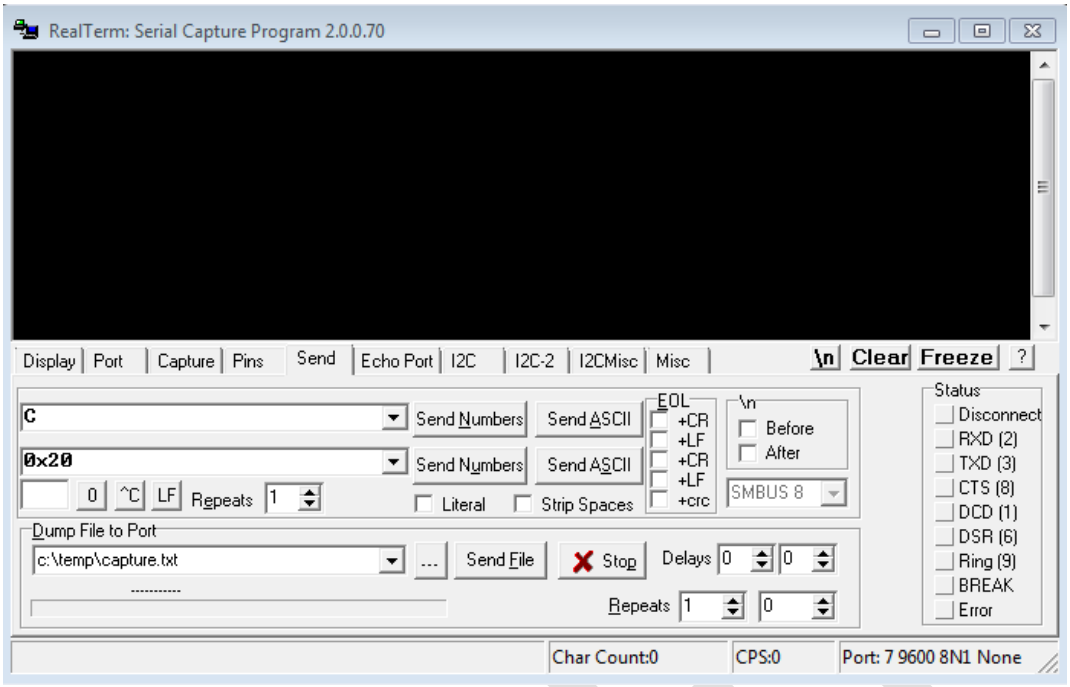


The range of pintable characters does not allow us to control each of the relays individually. For this we need to be able to send any value between 0 and 255 decimal.

Click on the "Send" tab in RealTerm. This tab allows us to send both ASCII characters and any value required. This program can handle ASCII character, decimal and Hex numbers. Try "C" and 4 as shown:



Now click on the higher Send ASCII button followed by the lower Send Numbers button. Relay number 3 should be the only one now switched on. This is because the binary value of 4 is 0b00000100. A value of 5 will switch on relays 1 and 3. In RealTerm is also possible to use Hexadecimal values as shown:



0x20 = 16 in decimal or 0b00100000 in binary so relay 6 will be switched on.

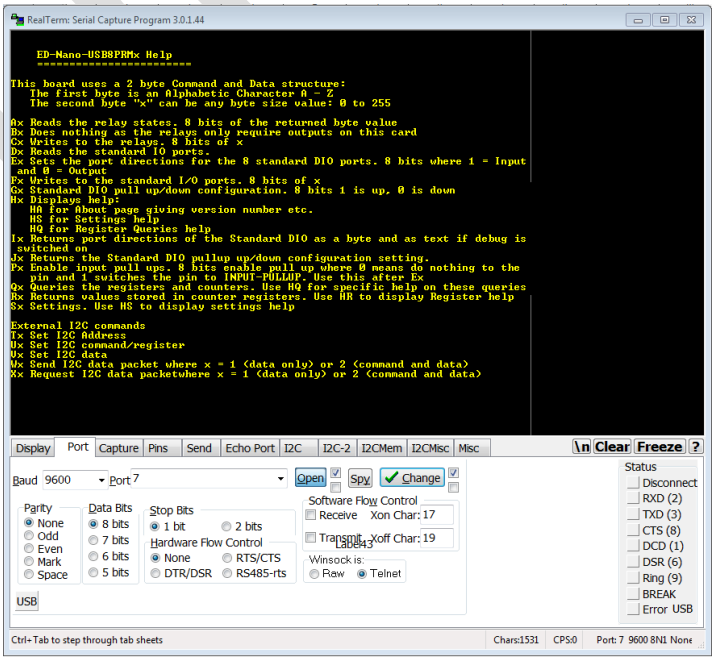
A list of values can also be sent so try 0x43 0x01 Send Numbers. 0x43 ASCII is the equivalent of "C" so in this case relay 1 will be switched on and all the others off.

Help pages:

Help pages are available as an output to the USB, RS232 and RS485 ports.

Connect your ED-Nano-USB8PRMx board to a PC and connect to it using a terminal emulator such as RealTerm.

If the serial port number and the baud rate (Default is 9600 Baud, 8 bits, No parity, 1 stop bit) are correct then sending the characters "HH" will cause the ED-Nano-USB8PRMx to return the main help page.



Commands:

The main commands are given in the following table:

Command	Description	
Ax	Reads the relay states. 8 bits of the returned byte value. A 1 in a bit position indicates that the associated relay is on.	
Bx	Does nothing as the relays only require outputs on this card	
Cx	Writes to the relays. 8 bits of x. A 1 in a bit position switches on the associated relay.	
Dx	Reads the standard IO ports. A 1 in a bit position indicates that the associated input is high.	
Ex	Sets the port directions for the 8 standard DIO ports. 8 bits where 1 = Input and 0 = Output	
Fx	Writes to the standard I/O ports. 8 bits of x. As each channel of the 8 bit DIO port is configurable as input or output then only output channels will be affected. A 1 in a bit position will take that output high if the channel is selected as an output.	
Gx	DIO pull up/down configuration. 8 bits. 1 is pull-up, 0 is pull-down	
Hx	Displays help pages:	
	HX	Displays the main help page (where X can be any character except for A, S or Q).
	HA	About page giving version number etc.
	HS	Settings help.
	HQ	Register Queries help.
Ix	Returns port directions of the Standard DIO as a byte and as text if debug is switched on	
Jx	Returns the Standard DIO pullup up/down configuration setting	
Px	Enable input pull ups. 8 bits enable pull up where 0 means do nothing to the pin and 1 switches the pin to INPUT-PULLUP. Use this after command Ex	
Qx	Queries the registers and counters. Use HQ for specific help on Queries.	
Rx	Returns values stored in counter registers. Use HR to display Register help.	
Sx	Settings. Use HS to display settings help.	
Tx	Set I2C Address of the device to be addressed.	
Ux	I2C Command and data stack. Up to 32 bytes can be stacked. The data counter will incremented for each byte stored.	
Vx	Data counter. Number of command + data bytes to be transmitted/received. X is a number (not ASCII) where x can be 0 to 32 bytes. Use V 0 to clear the counter.	
Wx	Send an I2C packet based on Tx, Ux and Vx. The data counter will be cleared (set to 0) when the data has been transmitted.	
Xx	Request data from the I2C. This will return a stream of Vx bytes. 'x' is the number of address or command bytes to be added to the I2C address. The information for the command is stored using Ux. The data counter will be cleared (set to 0) when the data has been transmitted.	

Query commands:

Command	Description	
Qn	returns Counter n value where n is 1 to 8	Query counters: Counter values are returned as an ASCII numeric string in the range 0 to 2^32-1 (4,294,967,295) and terminated by a carriage return character (ASCII 13, or \r) and a newline character (ASCII 10, or \n). The counters will wrap around (restart from zero) if the maximum count value is exceeded."));
Qa	returns Counter a value AND resets the counter to zero where "a" is ASCII "A" to "H" "A" returns and resets counter 1, "B" returns and resets counter 2 etc.	
QQ	Returns the software ID high byte	The software ID is to be used for identifying the version of software installed.
Qq	Returns the software ID low byte	

Command	Description	
QS	Returns the state of all the counters	Returns a byte where a 1 in a bit position indicates that the counter is disabled and a 0 indicates that it is enabled.
Qs	Returns the values of all the counters	
QT	Returns the value of ADC 1	The ADC value is returned as 2 bytes with a range of 0 to 1023.
Qt	Returns the value of ADC 2	

Settings commands:

Command	Description	
S0	Debug data off	Debug mode generates extra text describing the commands and results as they are used.
S1	Debug data on	
SA	Counter 1 on	Each of the 8 DIO channels on the Standard DIO port can have a counter enabled if that channel is set as an input. These counters will increase every time a rising or falling edge is seen on the enabled input pin. The maximum input frequency of the counters is yet to be determined. Contact us if more data is required.
Sa	Counter 1 off	
SB	Counter 2 on	
Sb	Counter 2 off	
SC	Counter 3 on	
Sc	Counter 3 off	
SD	Counter 4 on	
Sd	Counter 4 off	
SE	Counter 5 on	
Se	Counter 5 off	
SF	Counter 6 on	
Sf	Counter 6 off	
SG	Counter 7 on	
Sg	Counter 7 off	
SH	Counter 8 on	
Sh	Counter 8 off	
SI	All 8 counters on	
Si	All 8 counters off	
SJ	Frequency Display on	When Frequency Display is enabled a stream of data will be generated every second giving the difference in the count.
Sj	Frequency Display off	
SK	ADC1 Test program on	A simple test/demo program that takes the top 3 bits of the current ADC1 input value and uses this to select which relay to turn on.
Sk	ADC1 Test program off	
SL	Output Self-Test program on	A simple test/demo program that cycles though the relays and the Standard DIO as outputs.
Sl	Output Self-Test program off	
SM	ADC ref. = Vcc (Default)	The ADC reference voltage sets the top value of the Analog to digital converters. The internal 1.1V setting should be least noisy but this limits the maximum usable input voltage to 1.1V after which the result will always be maximum.
Sm	ADC ref. = internal 1.1V	
SR	Set Baud rate as 9600 (Default)	Changing the baud rate requires 2 commands in sequence. Your ED-Nano-USB8PRMx is set to 9600 baud by default to be compatible with our older hardware. Two additional baud rates are made available to improve overall system performance. Once a new baud rate has been saved the board will operate at that baud rate until changed. A hardware reset can be activated by linking the I2C interrupt line to 0V (GND) on power up or reset of the board. PL23 is provided for this function.
Sr	Save Baud rate as 9600. Must be preceded by SR to confirm	
SS	Set Baud rate as 115200	
Ss	Save Baud rate as 115200 Must be preceded by SS to confirm	
ST	Set Baud rate as 1000000	
Ss	Save Baud rate as 1000000 Must be preceded by ST to confirm	

NOTE an I2C only option (ED-I2C8PRMx) is available for this board which uses the on-board NXP PCAL9555A port expander device and no main processor. See https://www.nxp.com/products/interfaces/ic-spi-serial-interface-devices/ic-general-purpose-i-o/low-voltage-16-bit-ic-bus-gpio-with-agile-i-o-interrupt-and-weak-pull-up:PCAL9555A for communicating with this device.

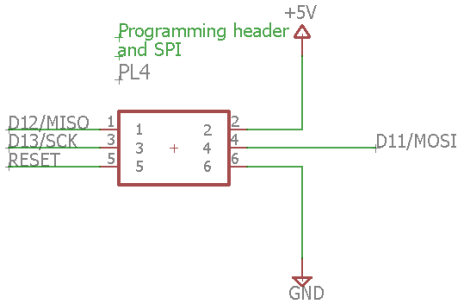
Processor re-programming:

The ED-Nano-USB8PRMx is essentially an Arduino™ Nano attached to an NPX PCAL9555A, 16 bit port expander. This allows the user to reprogram the board for their own purpose. However, reprogramming the board will erase the default software.

The NPX PCAL9555A is configured to its default address of 0x20 hex. There is a position for a 2x3 selector to enable the base address of the port expander to be shifted to addresses 0x21 – 0x27.

Port 0 of the port expander is connected to 8 relay drivers and power relays. Port 1 of the port expander is connected to the 8 protected DIO channels.

The ED-Nano-USB8PRMx has 2 rows of 0.1" pitch holes (PL5 and PL6) in the same configuration as an Arduino™ Nano and there is a 2X3 SPI/programming header (PL4) that can be used to reprogram the board.



The board can be reprogrammed using another Arduino™ product or USBASP ISP programmer.

If you have downloaded a boot loader into an Arduino™ Nano or similar then you have all you need to know to reprogram ED-Nano-USB8PRMx. If not then there are many good resources available on YouTube and instruction sites such as https://www.instructables.com/Burn-Bootloader-Into-Arduino-Nano-30-Clone-Board/

To reprogram the board via USB after a boot loader has been downloaded, it will be necessary to fit a link across PL1 to connect /DTR of the FT232BL to the reset pin of the ATMega328.

NOTE: The standard software is loaded without a boot loader and is locked.

The original or upgraded software can be reloaded and re-tested by EasyDAQ. This will incur a shipping charge and a small fee.

I2C port:

The ED-Nano-USB8PRMx uses the ATMega328 I2C port to control its on-board 16 bit port expander. This I2C channel is also connected to PL3, a JST SH 4 pin socket which is configured as a QWIIC interface and a 5 pin header which also includes a MORE HERE

I2C Examples:

1. Writing 1 byte to an EEPROM and to EEPROM then reading it back. In this example data was written to a CAT42C32 EEPROM. Values in blue brackets must be sent as numbers not ASCII.

Command and Responses	Notes

S1 = debug output on.	Turn on Debug mode so we can see what is going on. The 1 is in ASCII
TP	Set the address of the EEPROM. "P" = 80 decimal in ASCII
I2C address = 80	ASCII "P" = 80 decimal
U (0)	High byte of the Address in EEPROM
I2C data = 0	
U (0)	Low byte of the address in EEPROM
I2C data = 0, 0	
UH	Send character "H" as data
I2C data = 0, 0, 72	ACSII "H" = 72 decimal
V (3)	Confirm sending of 3 bytes
I2C number of bytes to send or receive = 3	
W3	"W" initiates the write. The second character can be anything.
Sending I2C packet: 80, 0, 0, 72	Data written to the EEPROM
U (0)	High byte of the Address in EEPROM
I2C data = 0	
U (0)	High byte of the Address in EEPROM
I2C data = 0, 0	
V (1)	1 byte to read
I2C number of bytes to send or receive = 1	
X (2)	2 bytes of EEPROM address data
Requested 1 bytes from 80, 0, 0	
Received bytes: 72	72 is ASCII for "H"

2. Write several bytes via I2C to EEPROM:

Command and Responses	Notes
S1 = debug output on.	Turn on Debug mode so we can see what is going on. The 1 is in ASCII
TP	Set the address of the EEPROM. "P" = 80 decimal in ASCII
I2C address = 80	ASCII "P" = 80 decimal
U (0)	High byte of the Address in EEPROM
I2C data = 0	
U (0)	Low byte of the address in EEPROM
I2C data = 0, 0	
UE	Add ASCII "E"
I2C data = 0, 0, 69	
Ua	Add ASCII "a"
I2C data = 0, 0, 69, 97	
Us	Add ASCII "s"
I2C data = 0, 0, 69, 97, 115	
Uy	Add ASCII "y"
I2C data = 0, 0, 69, 97, 115, 121	
UD	Add ASCII "D"
I2C data = 0, 0, 69, 97, 115, 121, 68	
UA	Add ASCII "A"
I2C data = 0, 0, 69, 97, 115, 121, 68, 65	
UQ	Add ASCII "Q"
I2C data = 0, 0, 69, 97, 115, 121, 68, 65, 81	
WW	Send the data to the EEPROM
Sending I2C packet: 80, 0, 0, 69, 97, 115,	

121, 68, 65, 81	

3. Read the written bytes:

Command and Responses	Notes
S1 = debug output on.	Turn on Debug mode so we can see what is going on. The 1 is in ASCII
TP	Set the address of the EEPROM. "P" = 80 decimal in ASCII
I2C address = 80	
U (0)	High byte of the Address in EEPROM
I2C data = 0	
U (0)	High byte of the Address in EEPROM
I2C data = 0, 0	
V (7)	Read 7 bytes
I2C number of bytes to send or receive = 7	
X (2)	2 bytes of EEPROM address data
Requested 7 bytes from 80, 0, 0	
Received bytes: 69, 97, 115, 121, 68, 65, 81	"EasyDAQ"

Power relays:

The 8 relays on the ED-Nano-USB8PRMx have optically isolated drivers to eliminate any unwanted voltage or current spikes.

The ED-Nano-USB8PRMx is fitted with 6V operating relays standard. The operating voltage of the relays can be selected to optimise power consumption of your system.

Specifications: Relays			
Parameter	**5V Power relays**	**6V Power relays**	**12V Power relays**
Rated voltage/current	5VDC/71mA each	6VDC/60mA each (50mA at 5V)	12VDC/30mA each
Must operate/release voltage	75%/10% of rated voltage		
Contact ratings	10A/240VAC or 8A 30VDC		
Contact resistance	100mΩ max		
Operate/release time	10mS/5mS		
Contact bounce period	0.6mS operate/ 7.2mS release		
Contact material	$AgSnO_2$		
Operational life (min)	Mechanical 10^7 / Electrical 10^5		
Contact arrangement	SPDT, Form C		

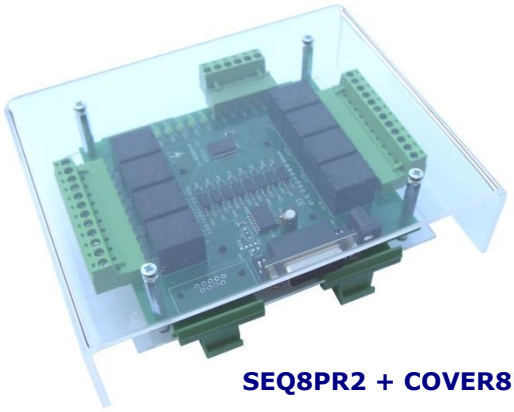
Describe using snubber on high voltage loads.

Board options:

The ED-Nano-USB8PRMx can be customised to your requirements. Some examples are given in the order codes and others can be generated on request. However there will be a minimum order quantity applied for some modifications.

The customisable elements are:

- **Processor code.** The code supplied can be modified to suit your purpose. Please contact us for details. Alternatively you can write your own software for the board.
- **Communications ports.** You can have just the communications ports required for your purpose fitted to reduce costs.
- **Regulators.** The board can be supplied without one or both of the regulators to reduce cost if they are not required.
- **Relay operating voltages.** The relays can be selected to suit your system if necessary.
- **1 or 2 part connectors.** The standard board is supplied with single part connectors as standard. If the system being built will need components changed easily then we recommend the 2 part horizontal or vertical connector options.



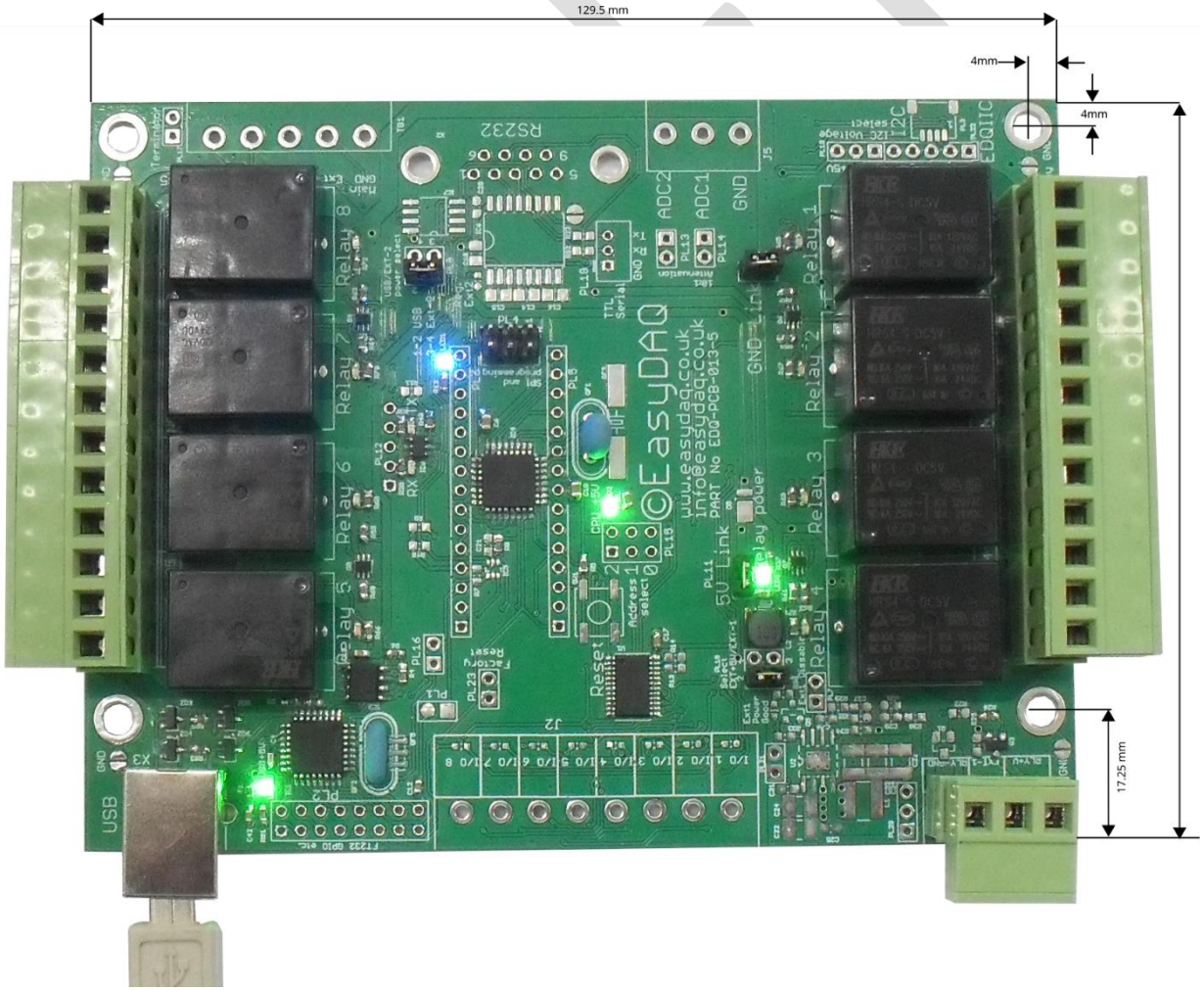
SEQ8PR2 + COVER8PRMxDIN

Order codes

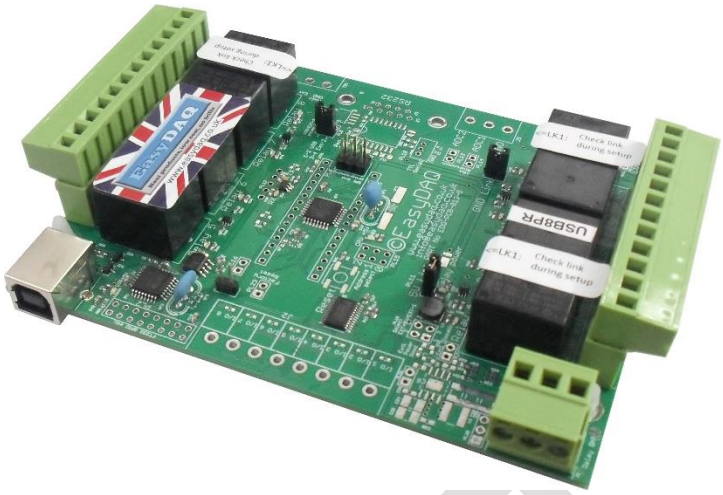
ED-Nano-USB-RS232-RS485- 8PRMx Full specification board with USB, RS232 and RS485 communications and 5V operating 10A, 240V power relays (10A), Single part connectors.
ED-Nano-USB8PRMx As above but just USB communications.
ED-Nano-USB8PRMx -5V = 5V relays
ED-Nano-USB8PRMx -12V = 12V relays
ED-Nano-USB8PRMx -24V = 24V relays
ED-Nano-USB8PRMx -2H = 2 part Horizontal connector
ED-Nano-USB8PRMx -2V = 2 part Vertical connector
ED-Nano-USB8PRMx-NR = USB only. No regulators
ED-I2C8PRMx NR = I2C only variant.
USB8PR -5V V2 = Mimicking the original USB8PR product
USB8PR -5V -2V V2 = Mimicking the original USB8PR -V2 product

Optional accessories

COVER8PRMx
COVER8PRMxDIN



USB8PR2 top view with dimensions



USB8PR variant

The USB8PR-V2 variant is a stripped-down, low-cost version of this card designed to replace the original EasyDAQ USB8PR. It uses 5V relays as standard and the same software as the ED-Nano-USB8PRMx.

The USB8PR V2 has the following features:

- 5V relays as standard
- 1 reverse polarity protected 5V input for external powering of the relays
- Optional 0V and 5V isolation through links
- Optional 6V, 12V and 24V relays
- Single and 2-part connector options

Unlike the original, the USB8PR V2 can also be queried to return the current relays states by using the "Ax" read port B command.

External Links:

ASCII: https://en.wikipedia.org/wiki/ASCII
RealTerm: https://sourceforge.net/projects/realterm/
Python test program: ToDo

Please contact us at EasyDAQ.co.uk if you need any help with our products including older versions of this data sheet.

Document versions:

Version number	Date	Notes
V1.0		Original. 100/DA/2005/002P based boards.
V2.0 (Draft)	27/04/22	Updated to describe new EDQ-PCB-013-x, ED-Nano versions of the board.
V2.1 (Draft)	07/06/22	Added dimensions on USB8PR2 top diagram

Product Datasheet 61

ToDo

Date	Description	Done	Notes
15/03/22	Describe I2C level shift		
15/03/22	Replace photos		
15/03/22	Overall system diagram		
15/03/22	I2C frames explanation		
15/03/22	Describe TTL Serial		
15/03/22	Describe SPI		
15/03/22	Describe power supply reverse voltage protection		
15/03/22	Describe earthing options on mounting holes		
15/03/22	Describe reset button		
15/03/22	Describe USB protection		
15/03/22	Complete EDQ-PCB-013-5 for photographing		
15/03/22	Describe RS485 including termination		
15/03/22	Describe RS232 voltages and options		
15/03/22	Describe using snubber on high voltage loads.		Relays
15/03/22	Describe USB I/O Port		