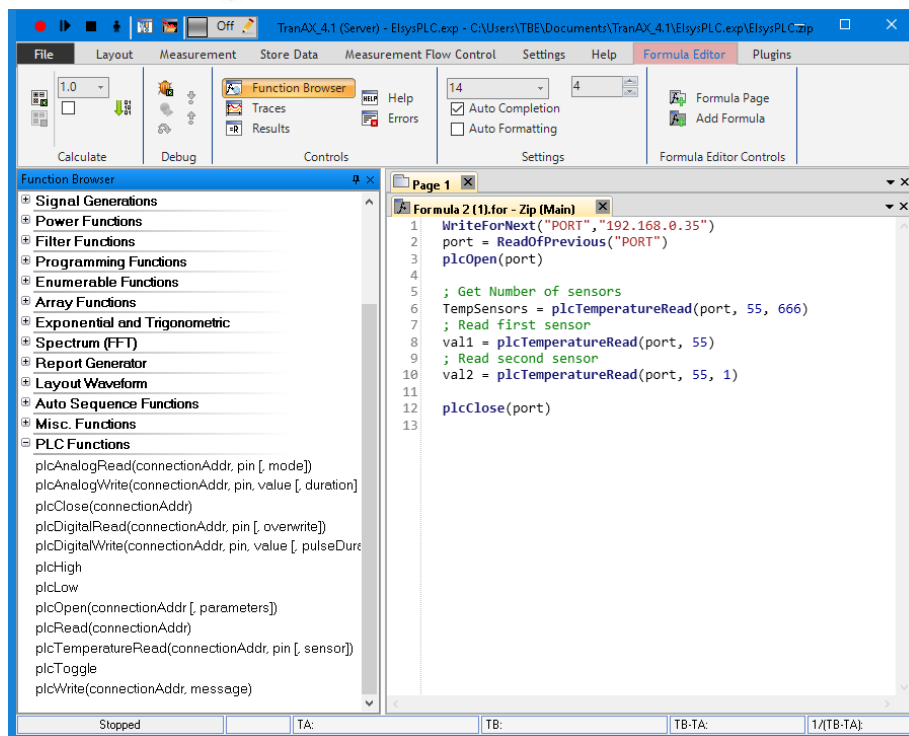


Elsys PLC

Programmable Logic Controller



User manual



Elsys AG
Mellingerstrasse 12
CH-5443 Niederrohrdorf

+41 56 496 01 55
www.elsys-instruments.com

Table of contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 4 |
| 1.1 | Preamble | 4 |
| 1.2 | Necessary Tools and Software | 4 |
| 1.2.1 | Support of multiple devices..... | 4 |
| 2 | Hardware setup and wiring..... | 5 |
| 2.1 | Power supply..... | 5 |
| 2.2 | USB connection..... | 5 |
| 2.3 | Ethernet connection | 5 |
| 3 | Formula Editor functions and commands | 6 |
| 3.1 | Device handling..... | 6 |
| 3.1.1 | plcOpen, plcClose | 6 |
| 3.2 | Pin handling | 8 |
| 3.2.1 | plcDigitalWrite | 8 |
| 3.2.2 | Digital Read..... | 9 |
| 3.2.3 | plcAnalogWrite | 10 |
| 3.2.4 | plcAnalogRead | 11 |
| 3.2.5 | plcTemperatureRead | 12 |
| 3.3 | Special functions | 13 |
| 3.3.1 | plcWrite | 13 |
| 3.3.2 | plcRead | 13 |
| 4 | Additional Information | 14 |
| 4.1 | Lowlevel Access | 14 |
| 4.2 | Factory Reset | 14 |
| 5 | Trouble shooting | 15 |
| 5.1 | Timeout while writing to device | 15 |
| 5.2 | PLC commands not available | 15 |
| 5.3 | Port is busy/occupied | 15 |
| 5.4 | Wrong serial port..... | 15 |
| 5.5 | Wrong IP Address or Subnet mask | 15 |
| 5.6 | Serial configuration..... | 15 |
| 5.7 | Additional commands..... | 15 |
| 6 | Pin definitions | 16 |
| 6.1 | Pin definition Outputs..... | 16 |
| 6.2 | Pin definition Inputs..... | 17 |

1 Introduction

1.1 Preamble

The Elsys PLC is a hardware extension, based on a Controllino Maxi Automation device (Atmel ATmega2560 microcontroller) to control external actuators, read external sensors, read simple low speed analogue signals like a potentiometer or a position sensor, write analogue signals for low speed applications, read digital temperature sensors.

It can be used and controlled by the Elsys's TranAX Software, with the built in Formula Editor, simple to use and straight forward programming.

1.2 Necessary Tools and Software

To control the Elsys PLC with TranAX, the following tools and software version are required

| Software | Version | Description |
|--------------------|------------|--------------------------------|
| TranAX | 4.1.0.1900 | Elsys Data Analysis Software |
| MathFunc_PLC.dll | 4.1.0.1929 | Copy into the TranAX directory |
| Firmware Elsys PLC | 180308a | Installed on PLC hardware |

1.2.1 Support of multiple devices

There can be used multiple Elsys PLC devices in the same system, depending on its usage. Elsys PLC can be connected per USB, which is a common Serial port connection. For more sophisticated application, a network connection will be recommended.

2 Hardware setup and wiring

2.1 Power supply

Elsys PLC can be powered either with 12Vdc or 24Vdc power supply. It is recommended to use at least a 12W power supply for operation only.



Digital outputs can drain up to 2A, please consider this for the evaluation of a suitable power supply. Elsys will provide a 24V, 50W power supply with each device.

2.2 USB connection

The Elsys PLC can be used with a USB connection, the cable length should not exceed more than 2m. The device will be recognized and listed as a Serial port (Com port).



Arduino Mega 2560 (COM12)



Elsys PLC can be used as a simple Comport device, according to the parameter specification, any other application can be used to control the device

After start up, the device prints its settings to the comport.

```
-----  
ID: Elsys PLC  
HW: Controllino MAXI  
SW: 180308a  
-----  
DHCP: off  
IP: 192.168.0.35  
MAC: 90:A2:DA:0D:47:61  
-----
```

2.3 Ethernet connection

A connection per Ethernet extends the cable length. As long as it is in the same IP range as the TranAX computer, the length of the cables doesn't matter, according to **given standard limitation of network equipment**. For Ethernet connections, CAT5 or better cables will work. Connection speed is limited to 100Mbps



Standard IP Address will be set to **192.168.0.35 before shipment**. This address can be changed, per serial terminal. This address will be set back to default (192.168.0.35) after a factory reset.

3 Formula Editor functions and commands

3.1 Device handling

As every serial connection or Ethernet connection, a port has to be defined to be opened. After successfully communication (data transmitted and received) with the device, this port has to be closed.

3.1.1 plcOpen, plcClose

| Function | Return |
|---|--------|
| plcOpen (connectionAddr, [, parameters]) | Status |

Opens a serial or ethernet (port 80) connection to an Elsys PLC device. It is possible to combine multiple devices. Therefore, each PLC function and command needs the parameter **connectionAddr**.

It is recommended to save the connection Address as a variable. In case of changing the IP Address or the serial port number, changes has just to be done once in the code. In these examples, **connectionAddr** will be saved with the function **WriteForNext()** and read with **ReadOfPrevious()**. The benefit of these two functions is to use the **connectionAddr** in code segments of the **Measurement Flow Control (MFC)** and in functions inside the Formula Editor code.



In case of a still opened connection at start of a Formula, **plcOpen** will automatically try to reuse the Serial Port or connection from the previous calculation. In case a serial device is already opened by another application, this application must be closed first.

| Function | Return |
|----------------------------------|--------|
| plcClose (connectionAddr) | n.a. |

Closes a Serial Port or Ethernet connection to an Elsys PLC device.

| Parameter | Description |
|-----------------------|---|
| connectionAddr | Comport or IP-Address as string, e.g. "COM12" for a serial port or "192.168.0.35" for an IP-Address |
| parameters | Generally, there are no additional parameters necessary. In case of redefining serial port parameters, a dictionary with the following parameters are allowed: baudrate, dataBits, stopBits, readTimeout, writeTimeout, newline and dtrEnable |

Based on the examples below, all other code examples can be implemented.

Examples 1, default Ethernet connection:

```

; open a Ethernet Connection
WriteForNext("PORT","192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
; your code here
plcClose(port)
    
```

Examples 2, default serial port connection:

```
; open a Serial Port Connection
WriteForNext("PORT","COM15")
port = ReadOfPrevious("PORT")
plcOpen(port)
; your code here
plcClose(port)
```

Examples 3, custom serial port connection:

```
dictParams = Dictionary()
dictParams("baudrate") = 115200
dictParams("readTimeout") = 2000
dictParams("writeTimeout") = 2000
dictParams("newline") = "\n"
dictParams("dtrEnable") = False
```

```
WriteForNext("PORT","COM15")
port = ReadOfPrevious("PORT")
plcOpen(port, dictParams)
; your code here
plcClose(port)
```



Default parameters for a serial connection are:
115200 Baud, 8 data bits, 1 stop bit, parity none



For **Ethernet connection, port 80 (http)** has to be opened for TranAX application in the firewall.

3.2 Pin handling

The following instruction set is currently implemented.

3.2.1 plcDigitalWrite

| Function | TP | Return |
|---|----|--------|
| plcDigitalWrite (connectionAddr, pin, value [, pulseDuration]) | 0 | Status |

Defines a pin as a digital output, sets its status according to value.

| Parameter | Description |
|-----------------------|--|
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information |
| pin | Arduino Pin according pin out list, see appendix for more details. |
| value | Valid parameter for value are: <ul style="list-style-type: none"> • plcLow, equal to integer value 0, output is OFF • plcHigh, equal to integer value 1, output is ON • plcToggle, equal to integer value 2, inverts the current output (high switches to low and vice versa) All three parameters are predefined keywords. |
| pulseDuration | The optional parameter pulseDuration is used to define a Monoflop function, the output will set to value for the defined duration in seconds (integer or float). |

Example 1: Two examples of a Monoflop Output. The second solution with additional parameter pulseDuration doesn't block the code execution

```
WriteForNext("PORT", "192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
```

```
; 1. Monoflop
plcDigitalWrite(port, 22, plcHigh)
Delay(1.5)
plcDigitalWrite(port, 22, plcLow)
```

```
; 2. Monoflop
plcDigitalWrite(port, 22, plcHigh, 1.5)
```

```
plcClose(port)
```

Example 2: Toggle output from ON to OFF and vice versa.

```
WriteForNext("PORT", "192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
```

```
plcDigitalWrite(port, 22, plcToggle)
```

```
plcClose(port)
```


3.2.2 Digital Read

| Function | TP | Return |
|--|----|--------|
| plcDigitalRead(connectionAddr, pin [, overwrite]) | 1 | 0, 1 |

Defines a pin as a digital input. Return value would be 0 if input signal is low, and 1 if input signal is high.

| Parameter | Description |
|-----------------------|--|
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information |
| pin | Arduino Pin according pin out list, see appendix for more details. |
| overwrite | The optional parameter overwrite is used to redefine a pin function. If a pin is defined as an output, plcDigitalRead returns its value. With the optional parameter overwrite = true, the pin will be reconfigured as a digital input. |



The redefinition of a digital output to a digital input is not recommended in combination with a Elsys PLC device.

Example 1, Input AIO is connection with a push button to Vin of the Elsys PLC.

```
WriteForNext("PORT", "192.168.0.35")
```

```
port = ReadOfPrevious("PORT")
```

```
plcOpen(port)
```

```
; Pushbutotn pressed InputVal= 1, else = 0
```

```
InputVal = plcDigitalRead(port, 54)
```

```
plcClose(port)
```

3.2.3 plcAnalogWrite

| Function | TP | Return |
|---|----|--------|
| <code>plcAnalogWrite(connectionAddr ,pin, value [, duration] [, mode])</code> | 2 | Status |

Defines a pin as an analogue output. Please note **not every pin** can be configured as an analogue output, just pins which **supports PWM**.

| Parameter | Description |
|-----------------------|--|
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information |
| pin | Arduino Pin according pin out list, see appendix for more details. |
| value | Value defines the output signal in thousandth (0.1%) interval. 0 means 0V, 1000 means max Output signal, 10V for Elsys PLC devices. |
| duration | The optional parameter duration is used to generate a ramp as an output signal. It will reach the analogue output value after the time duration in seconds. |
| mode | The optional parameter mode is used to generate an exponential output signal, which allows linear dimming connected LEDs. <ul style="list-style-type: none"> • Not defined, False or 0 means Linear output • True or 1 means exponential output |

Example 1, Ouput AO0 used as an analogue output.

```
WriteForNext("PORT","192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
```

```
; Set output to 0V
plcAnalogWrite(port, 12, 0)
; Wait 1 second
delay(1)
; Set output to 10V, ramp for 2 seconds
plcAnalogWrite(port, 12, 1000, 2)
```

```
plcClose(port)
```

Example 2, Ouput AO0 used as an analogue output for LEDs.

```
WriteForNext("PORT","192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
```

```
; Set output to 0V
plcAnalogWrite(port, 12, 0)
; Wait 1 second
delay(1)
; Set output to 10V, ramp for 2 seconds
plcAnalogWrite(port, 12, 1000, 2, True)
```

```
plcClose(port)
```

3.2.4 plcAnalogRead

| Function | TP | Return |
|--|----|-----------|
| plcAnalogRead(connectionAddr, pin [, mode]) | 3 | ADC Value |

Defines a pin as an analogue input. Please note not every pin can be configured as an analogue input. Return value is the ADC value itself, for Elsys PLC devices a value between 0 and 1023 (10bit ADC).

| Parameter | Description | | | | | | | | | | | | | | |
|---------------------------|---|------|----------|---------------------------|-------------------|----------|---------------|----------|---------------|----------|------------|----------|-----------|------------|---|
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information | | | | | | | | | | | | | | |
| pin | Arduino Pin according pin out list, see appendix for more details. | | | | | | | | | | | | | | |
| mode | The optional parameter mode is used to read statistic data from the analogue input. Update interval is set to 100ms. <table border="1" data-bbox="523 660 1469 940"> <thead> <tr> <th>Mode</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0 (or not defined)</td> <td>current ADC value</td> </tr> <tr> <td>1</td> <td>Minimum value</td> </tr> <tr> <td>2</td> <td>Maximum value</td> </tr> <tr> <td>3</td> <td>Mean value</td> </tr> <tr> <td>4</td> <td>RMS value</td> </tr> <tr> <td>666</td> <td>Restart statistic data for mode 1-4 above</td> </tr> </tbody> </table> | Mode | Function | 0 (or not defined) | current ADC value | 1 | Minimum value | 2 | Maximum value | 3 | Mean value | 4 | RMS value | 666 | Restart statistic data for mode 1-4 above |
| Mode | Function | | | | | | | | | | | | | | |
| 0 (or not defined) | current ADC value | | | | | | | | | | | | | | |
| 1 | Minimum value | | | | | | | | | | | | | | |
| 2 | Maximum value | | | | | | | | | | | | | | |
| 3 | Mean value | | | | | | | | | | | | | | |
| 4 | RMS value | | | | | | | | | | | | | | |
| 666 | Restart statistic data for mode 1-4 above | | | | | | | | | | | | | | |

Example 1, Input AI0 used as an analogue input.

```
WriteForNext("PORT", "192.168.0.35")
```

```
port = ReadOfPrevious("PORT")
```

```
plcOpen(port)
```

```
; Clear Analogue Statistic
```

```
plcAnalogRead(port, 54, 666)
```

```
adcval = plcAnalogRead(port, 54) ; equal to plcAnalogRead(port, 54, 0)
```

```
minval = plcAnalogRead(port, 54, 1)
```

```
maxval = plcAnalogRead(port, 54, 2)
```

```
meanval = plcAnalogRead(port, 54, 3)
```

```
rmsval = plcAnalogRead(port, 54, 4)
```

```
plcClose(port)
```

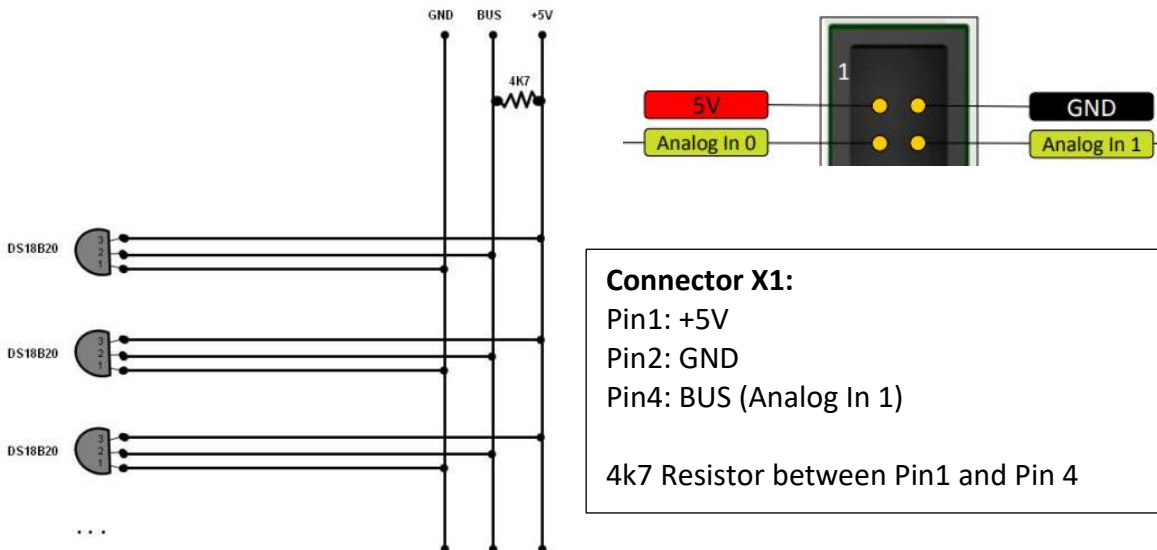
3.2.5 plcTemperatureRead

| Function | TP | Return |
|--|----|---------------|
| plcTemperatureRead (connectionAddr, pin [, sensor]) | 4 | Nr of sensors |

Defines a pin as a OneWire pin for DS1820 temperature sensors.

| Parameter | Description | | | | | | | | |
|-----------------------|---|------|----------|--------------------|-----------------------|------------------|---------------|------------|--|
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information. | | | | | | | | |
| pin | Arduino Pin according pin out list, see appendix for more details. | | | | | | | | |
| sensor | The optional parameter sensor is used to define the connected sensor. OneWire support up to 127 sensors. | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Mode</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>not defined</td> <td>First sensor in chain</td> </tr> <tr> <td>0 ... 126</td> <td>Sensor number</td> </tr> <tr> <td>666</td> <td>Get number of devices in OneWire chain</td> </tr> </tbody> </table> | Mode | Function | not defined | First sensor in chain | 0 ... 126 | Sensor number | 666 | Get number of devices in OneWire chain |
| Mode | Function | | | | | | | | |
| not defined | First sensor in chain | | | | | | | | |
| 0 ... 126 | Sensor number | | | | | | | | |
| 666 | Get number of devices in OneWire chain | | | | | | | | |

Example to connect one or multiple OneWire DS18B20 sensors:



Example 1, OneWire on Pin A1, 55, with two DS18B20 sensors, 4k7 Ohm Resistor Pullup

```

WriteForNext("PORT", "192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)

; Get Number of sensors
TempSensors = plcTemperatureRead(port, 55, 666)
; Read first sensor
val1 = plcTemperatureRead(port, 55)
; Read second sensor
val2 = plcTemperatureRead(port, 55, 1)

plcClose(port)
    
```

3.3 Special functions

In case of new beta firmware or tests with other hardware and devices, the follow functions can be used:

3.3.1 plcWrite

| Function | Return |
|--|---|
| plcWrite (connectionAddr, message [, delayMs [, collectResults]]) | NotDefined, True, List |
| Write a message, based on a string to a connectionAddr. If a timeout occurs the return value will be NotDefined. | |
| Parameter | Description |
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information |
| message | Elsys PLC specific String, for other devise, according its specification. |
| delayMs | Defines a delay in milliseconds after writing the data. This can be used if the next read or write command must wait a specific time. |
| collectResults | Set collectResults to true if you want to automatically capture all the responses. You will receive a list with values. |

3.3.2 plcRead

| Function | Return |
|--|---|
| plcRead (connectionAddr [, delayMs [, collectResults]]) | String, Double, List, NotDefined |
| Read a parameter from connectionAddr. Returns a string or a double value depending on the response. If a timeout occurs the return value will be NotDefined. | |
| Parameter | Description |
| connectionAddr | Comport or IP-Address please see section plcOpen() and plcClose() for more detailed information |
| delayMs | Defines a delay in milliseconds after reading the data. This can be used if the next read or write command must wait a specific time. |
| collectResults | If you want to do multiple reads set this to true. You will receive a list with all the return values. |

Example 1, Read an analogue input

```
WriteForNext("PORT","192.168.0.35")
port = ReadOfPrevious("PORT")
plcOpen(port)
```

```
; read analog input A0 RMS value
val = plcWrite(port, "PN:54;TP:3;P0:4;")
```

```
plcClose(port)
```

Examples:

4 Additional Information

4.1 Lowlevel Access

Return value of a write function will be a float value: 0 means no errors, every other number means a failure duration operation or setup. Read functions will return its status or return value. The protocol is based on simple combined string parameters.

Write command:

PN:nn;TP:xy;P0:aaa;P1:bbb;P2:ccc;...;Pn:nnn;

Read command:

PN:nn;[P0;]

| Parameter | Description |
|-----------|--|
| PN | Defines the Pin number according the pin layout, see section 6 for more detailed information. |
| TP | Defines the Type of function: 0: Digital output, 1: Digital input, 2: Analog output, 3: Analog input, 4. One-Wire bus for temperature sensors |
| P0 ... Pn | Additional parameters, depending on the pin type definition |

Example 1, switch on R0 for 2 seconds over serial command

PN:22;TP:0;P0:1;P1:1;

Example 2, switch on R0 for 2 seconds per html request over ethernet

http://192.168.0.35/plc?PN=22&TP=0&P0=1&P1=2

4.2 Factory Reset

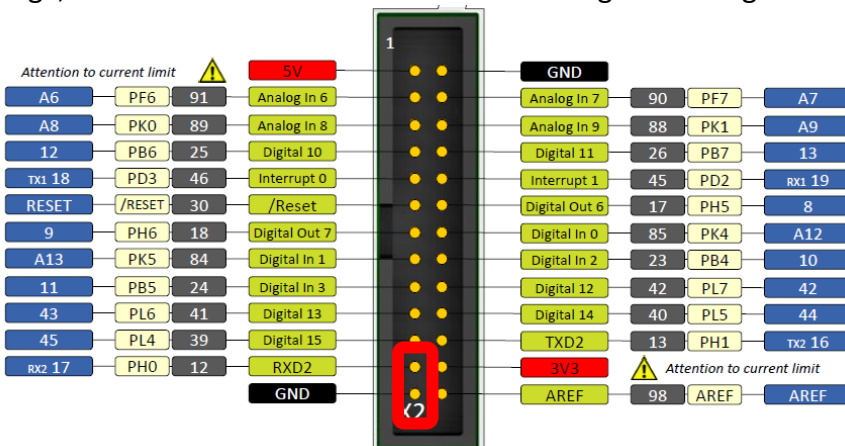
A Factory Reset can be done per hardware manipulation or per USB connection.

Software Reset:

Open a Terminal application (like HTerm), enter **FRESET**; and press enter. This will reset the IP settings back to 192.168.0.35, DHCP disabled.

Hardware Reset:

Power off ELSYS PLC, make a connection between **Pin 23 and 25 on the connector X2** and power up the device, wait a second and remove the connection. This will reset the Network settings and restart the PLC. This connection is equal to connecting Pin PH0 (Pin 17 on the Arduino Mega, or Pin12 from the microcontroller ATmega 2560 to ground.



5 Trouble shooting

5.1 Timeout while writing to device

Check the connection of the cables, either USB connection or Ethernet connection. Make sure the Elsys PLC is powered up and any additional devices like network switches are connected and switched on.

Also Check the port configuration: For USB connection the Comport settings, please note that the serial port number can change if the USB connector is connected to another port.

IP-addresses must fit into the same IP range of the system on which TranAX is running. Default IP settings of Elsys PLC is 192.168.0.35.

5.2 PLC commands not available

Elsys PLC commands are not per default implemented into TranAX at the moment. You will have to copy the latest version of MathFunc_PLC.dll into the TranAX installation directory. Please contact info@elsys-instruments.com for more information or any support.

5.3 Port is busy/occupied

Make sure no other software is using the specified serial port. Otherwise TranAX will not be able to communicate with the connected Elsys PLC device.

5.4 Wrong serial port

Switching USB port or running an Experiment on another computer system will change the number of the serial port. Make sure you are either using a fixed Serial port number or change the settings in your Formula Editor.

5.5 Wrong IP Address or Subnet mask

Computer and Elsys PLC must be in the same IP range. Default IP settings of Elsys PLC is 192.168.0.35 with Subnet mask 255.255.255.0. Changes can be done on its built in Web interface or per serial command.

5.6 Serial configuration

The following commands can be used to setup a ethernet connection via serial port:

| Command | Description |
|---------------------|--|
| DHCP:1; and DHCP:0; | Enable or disable DHCP |
| IP:[Address] | Set an IP-Address, please check and configure DHCP settings! |
| FRESET; | Factory Reset, DHCP off, IP 192.168.0.35 |

5.7 Additional commands

The following commands can be used to test the Elsys PLC via serial port:

| Command | Description |
|----------|--|
| GETINFO; | Returns the Elsys PLC settings and configuration |
| REBOOT; | Restart the PLC, like power off and on again |
| FRESET; | Factory Reset, DHCP off, static IP 192.168.0.35 |

6 Pin definitions

For controlling inputs and outputs, the **Arduino Mega pin numbers** are used. To control the Relays "R0", the pin 22 must be written and used.

Digital Outputs DO0 until DO7 can also be used as PWM outputs, means used with plcAnalogWrite() command. DO0 until DO7 are high side switches, the load has to be connected between this pin and GND.

6.1 Pin definition Outputs

| PLC name | Pin number | Description, level, values |
|----------|------------|---------------------------------------|
| R0 | 22 | Relay N.O. |
| R1 | 23 | Relay N.O. |
| R2 | 24 | Relay N.O. |
| R3 | 25 | Relay N.O. |
| R4 | 26 | Relay N.O. |
| R5 | 27 | Relay N.O. |
| R6 | 28 | Relay N.O. COMM. for R6 and R7 |
| R7 | 29 | Relay N.O. COMM. for R6 and R7 |
| R8 | 30 | Relay N.O. |
| R9 | 31 | Relay N.O. |
| DO0 | 2 | High side switch, 2A max. |
| DO1 | 3 | High side switch, 2A max. |
| DO2 | 4 | High side switch, 2A max. |
| DO3 | 5 | High side switch, 2A max. |
| DO4 | 6 | High side switch, 2A max. |
| DO5 | 7 | High side switch, 2A max. |
| DO6 | 8 | High side switch, 2A max. |
| DO7 | 9 | High side switch, 2A max. |
| A00 | 12 | 0 ... 10V |
| A01 | 13 | 0 ... 10V |

6.2 Pin definition Inputs

Analog Inputs depends on the power supply voltage.

| Powering voltage | ADC resolution | ADC |
|------------------|----------------|-------------------|
| 12V | 15mV / bit | 10bit, 0 ... 1023 |
| 24V | 30mV / bit | 10bit, 0 ... 1023 |

Digital inputs have the following levels:

| Powering voltage | Logic value | Level |
|------------------|-------------|--------------|
| 12V | 0 | 0 ... 3.6V |
| 12V | 1 | 9 ... 13.2V |
| 24V | 0 | 0 ... 7.2V |
| 24V | 1 | 18 ... 26.4V |

| PLC name | Number | Description, level, values |
|----------|--------|---|
| AI0 | 54 | analog 0 ... 1023, digital 0, 1 |
| AI1 | 55 | analog 0 ... 1023, digital 0, 1 |
| AI2 | 56 | analog 0 ... 1023, digital 0, 1 |
| AI3 | 57 | analog 0 ... 1023, digital 0, 1 |
| AI4 | 58 | analog 0 ... 1023, digital 0, 1 |
| AI5 | 59 | analog 0 ... 1023, digital 0, 1 |
| AI6 | 60 | analog 0 ... 1023, digital 0, 1 |
| AI7 | 61 | analog 0 ... 1023, digital 0, 1 |
| AI8 | 62 | analog 0 ... 1023, digital 0, 1 |
| AI9 | 63 | analog 0 ... 1023, digital 0, 1 |
| AI10 | 64 | analog 0 ... 1023, digital 0, 1 |
| AI11 | 65 | analog 0 ... 1023, digital 0, 1 |
| AI12 | 68 | analog 0 ... 1023, digital 0 or 1, 0... 10V |
| AI13 | 69 | analog 0 ... 1023, digital 0 or 1, 0... 10V |
| DI0 | 66 | digital 0, 1 |
| DI1 | 67 | digital 0, 1 |
| DI2 | 10 | digital 0, 1 |
| DI3 | 11 | digital 0, 1 |
| IN0 | 18 | digital 0, 1 |
| IN1 | 19 | digital 0, 1 |

Document information

Filename: Elsys PLC Manual EN.docx
Status: Released

Last modification

Date: 28.03.2018
Author: Thomas Berger