



Series of devices

IMOD

iMod + PLC Instructions



Table of Contents:

[Chapter 1 Introduction](#)

[Chapter 2 Prepare the work environment](#)

[Linux on any system \(virtual-box + image\)](#)

[application Ladder Design](#)

[Configuration](#)

[IMod](#)

[Chapter Sample configuration](#)

[First modification of the status LEDs counting USER_LED](#)

[LadderClassic Structure and configuration](#)

[Program Description](#)

[Configuration](#)

[Construction and ConfigurationiMod](#)

[Definition channels](#)

[Definition of parameters](#)

[Verification example.in](#)

[Establishing a connection](#)

[verification of the example](#)

[Second e-mail notification at 10 times the change of status LEDs USER_LED](#)

[Structure and configuration Classic Ladder](#)

[Program Description](#)

[Configuration](#)

[Building configurationiMod](#)

[Definition](#)

[The definition of parameters](#)

[Verification example.in](#)

[Establishing a connection](#)

[verification of the example](#)

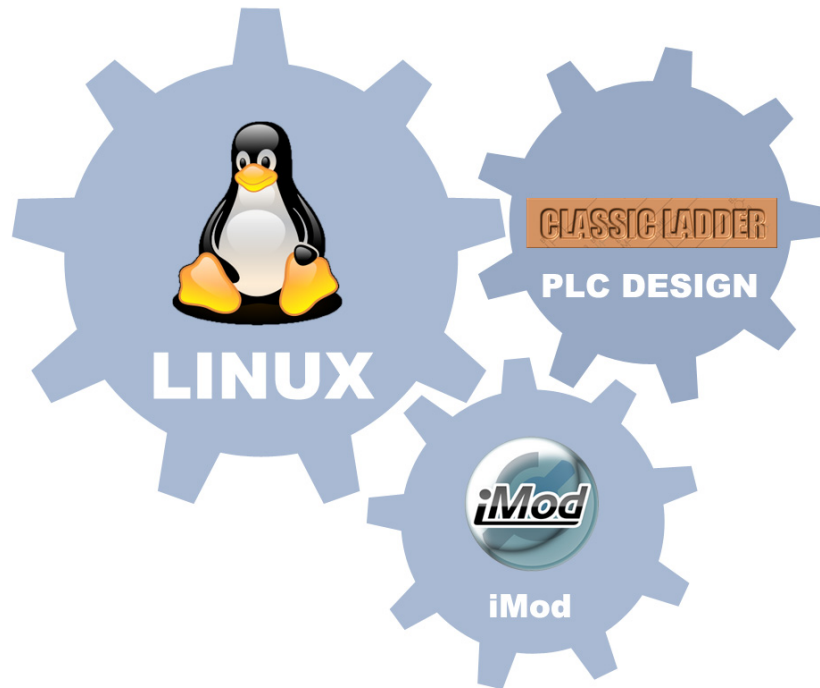
[Appendix A: Description of the function blocks the Classic Ladder](#)

Chapter 1 Introduction

iMod NPE running on the platform has the ability to work as a PLC. To this end, beyond the internal configuration file, use the application ClassicLadder used to convert the ladder language LD to Clanguage.



Chapter 2 Prepare the work environment



to achieve the functionality of the PLC on iModzie need:

1. Linux or program for virtual machines, such as [Oracle VirtualBox](#) with the image of the [Linux](#)'a
2. software [ClassicLadderDesign](#)
3. latest version of iMod ENGIN NPE platform (more on this in the chapter devoted to the preparationapplication) [iMod](#)

project files used in this manual are on the server

ftp:adres: [ftp.a2s.com](ftp://ftp.a2s.com)

login: [npe_imod@ftp.a2s.pl](ftp://npe_imod@ftp.a2s.pl):

password npe_1m0d

Linux on any system (virtual-box + image)

To build an application PLC system is needed to support Linux GTK + desktop environment.

Linux can be run directly from Windows using a virtual machine.

VirtualBox can be downloaded from:

<https://www.virtualbox.org/>

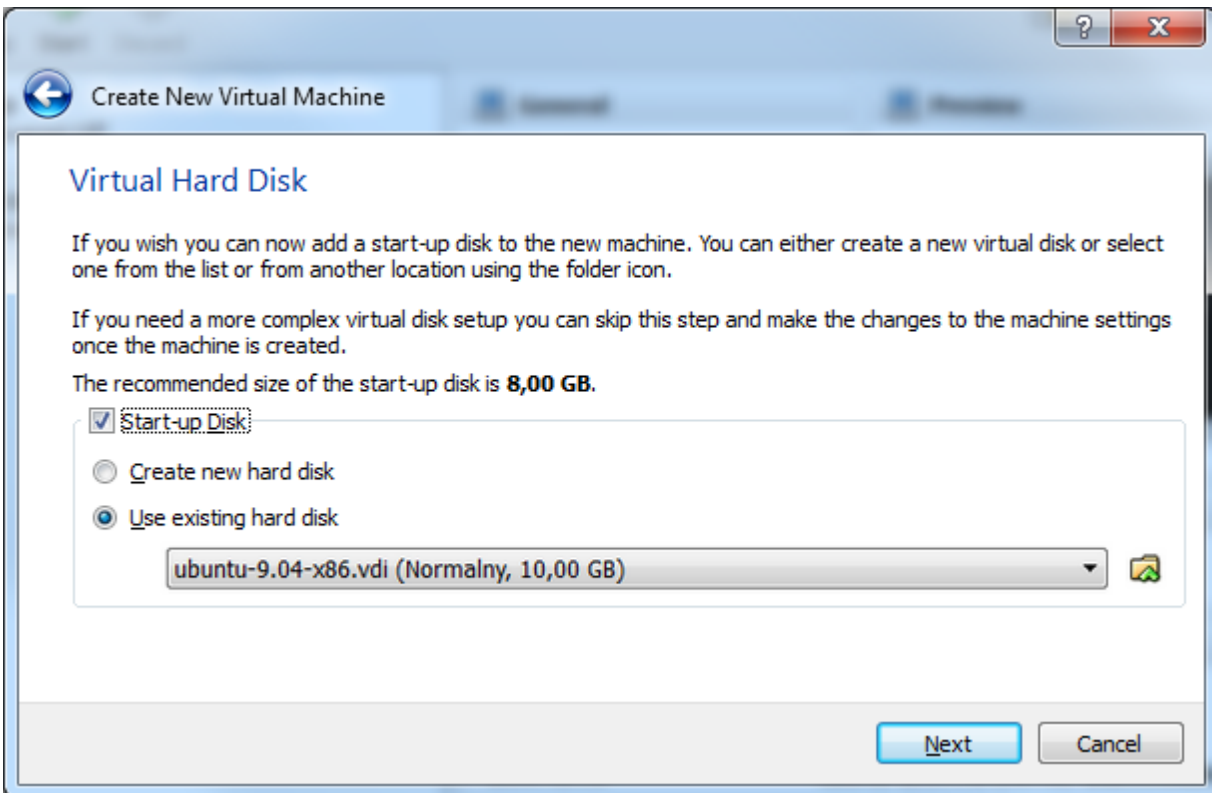
Linux image can be downloaded from:

<http://virtualboxes.org/images/ubuntu/>

1 Start VirtualBox



second Stsackcloth new virtual machine by clicking the new
third Name the virtual machine, operating system, Linux, and select the appropriate version of the
fourth Assign min. amount of RAM for the virtual machine (256 MB)
5 Select 'Existing Hard Drive' and select the image file of Ubuntu

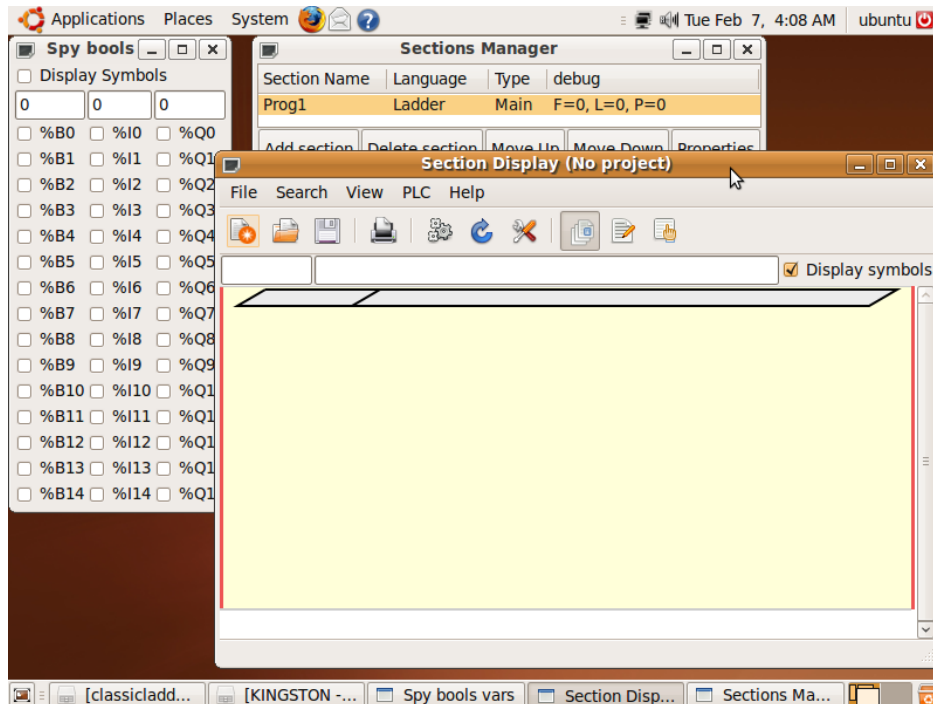


6th After the correct image is loaded, a dialog with the parameters of virtual machines. Right click on the added virtual machine and select the "Start".



Password and login to Linux should be used in place of download.
For Ubuntu Linux version
**9.04: login:ubuntu:
password reverse**

application Ladder Design



For purposes of this instruction was used ClassicLadder program.

Classic Ladder software can be downloaded from:

<http://sourceforge.net/projects/classicladder/>

or from ftp.a2s.pl

to add new elements in the ladder in the application, run the editor window:

View-> Editor Window -> " Modify

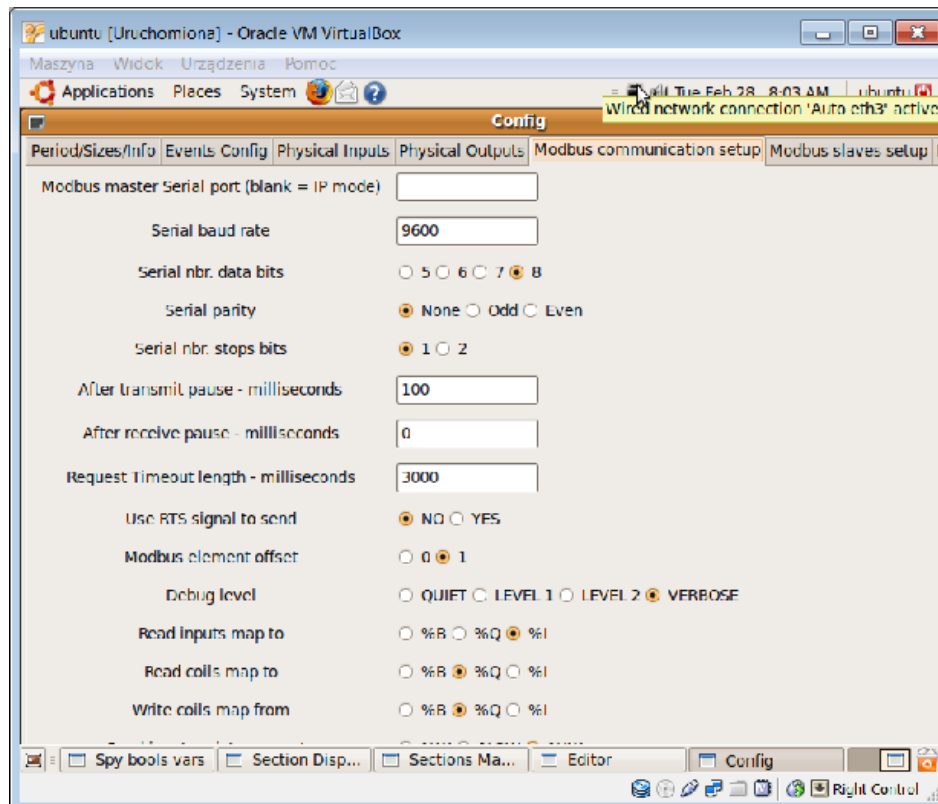
"inthefwindow, *Properties* you can give the address of the variable and possibly other parameters of the function block.



Please note that before writing your project, thewindow, *Editor* exit the edit mode by clicking **OK**.

Configuration

To enter the configuration settings you must go to the PLC → Configuration



tab *Modbus communication setup* set the communication parameters of the device. At the time of leaving an empty field *Serial Port Modbusmaster*, the communication will take place through an Ethernet interface.



To save changes to configuration parameters, simply close the *Config*. The changes are saved automatically.

IMod

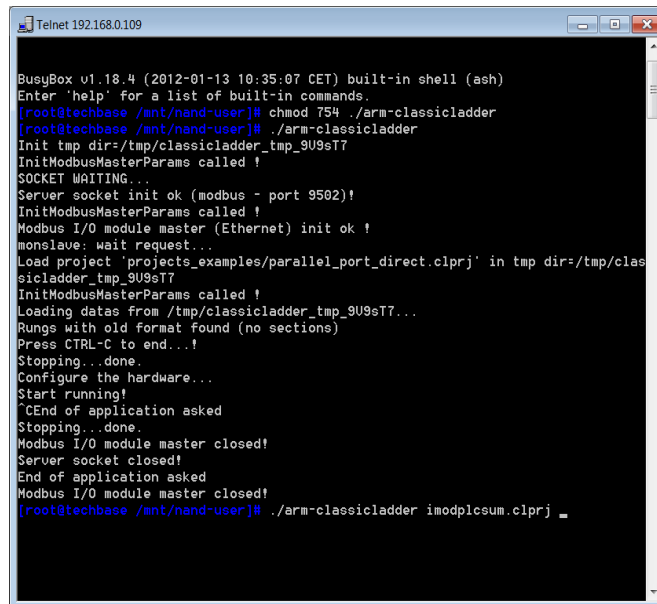
Before performing the following steps, make an application using *softmgr* have the latest software.



Free technical support is always applies the latest stable packages. You can update the iMod engine with the command:

update softmgr iMod

In order to move the PLC mode, run the project iModzie ClassicLadder compiled application through the cross arm-classicladder available on the server. ftpA2s.Com (login and password at the beginning of Chapter 2)



```
Telnet 192.168.0.109
BusyBox v1.18.4 (2012-01-13 10:35:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
[root@techbase /mnt/nand-user]# chmod 754 ./arm-classicladder
[root@techbase /mnt/nand-user]# ./arm-classicladder
Init tmp dir=/tmp/classicladder_tmp_9U9sT7
InitModbusMasterParams called !
SOCKET WAITING...
Server socket init ok (modbus - port 9502)!
InitModbusMasterParams called !
Modbus I/O module master (Ethernet) init ok !
monslave: wait request...
Load project 'projects_examples/parallel_port_direct.clprj' in tmp dir=/tmp/classicladder_tmp_9U9sT7
InitModbusMasterParams called !
Loading datas from /tmp/classicladder_tmp_9U9sT7...
Rungs with old format found (no sections)
Press CTRL-C to end...!
Stopping...done.
Configure the hardware...
Start running!
^C End of application asked
Stopping...done.
Modbus I/O module master closed!
Server socket closed!
End of application asked
Modbus I/O module master closed!
[root@techbase /mnt/nand-user]# ./arm-classicladder imodplcsum.clprj _
```

1. Copy files in binary *arm-classicladder* mode, and *projekt.clprj* (project previously created application, PLC) to the directory:
/ mnt / nand-user /

2. Assign the appropriate permissions for the file `arm-classicladder` by the command
`chmod 754 /mnt/nand-user/arm-classicladder`
3. Run the application `arm-classicladder`
`/mnt/nand-user/arm-classicladder`
4. Run the project through the application `arm-classicladder`
`/mnt/nand-user/arm-classicladder project.clprj`

PLC application termination is done by breaking the program, the key combination `Ctrl + C`, or when you restart iMod.



Agency application restart iMod PLC is interrupted, so it is advisable to add a startup script to the `/mnt/mtd/rcs` which will auto-start script after reboot the machine.



To run the application locally, go to the menu `PLC -> RUN`.
With the window `var bools Spy` (View -> `var Boolswindow`) and the window `free vars Spy` (View -> `Free varwindow`), you can control the states and the parameter values.

Chapter Sample configuration

The sample configuration iMod telemetry module in operation as the PLC will be presented two examples:

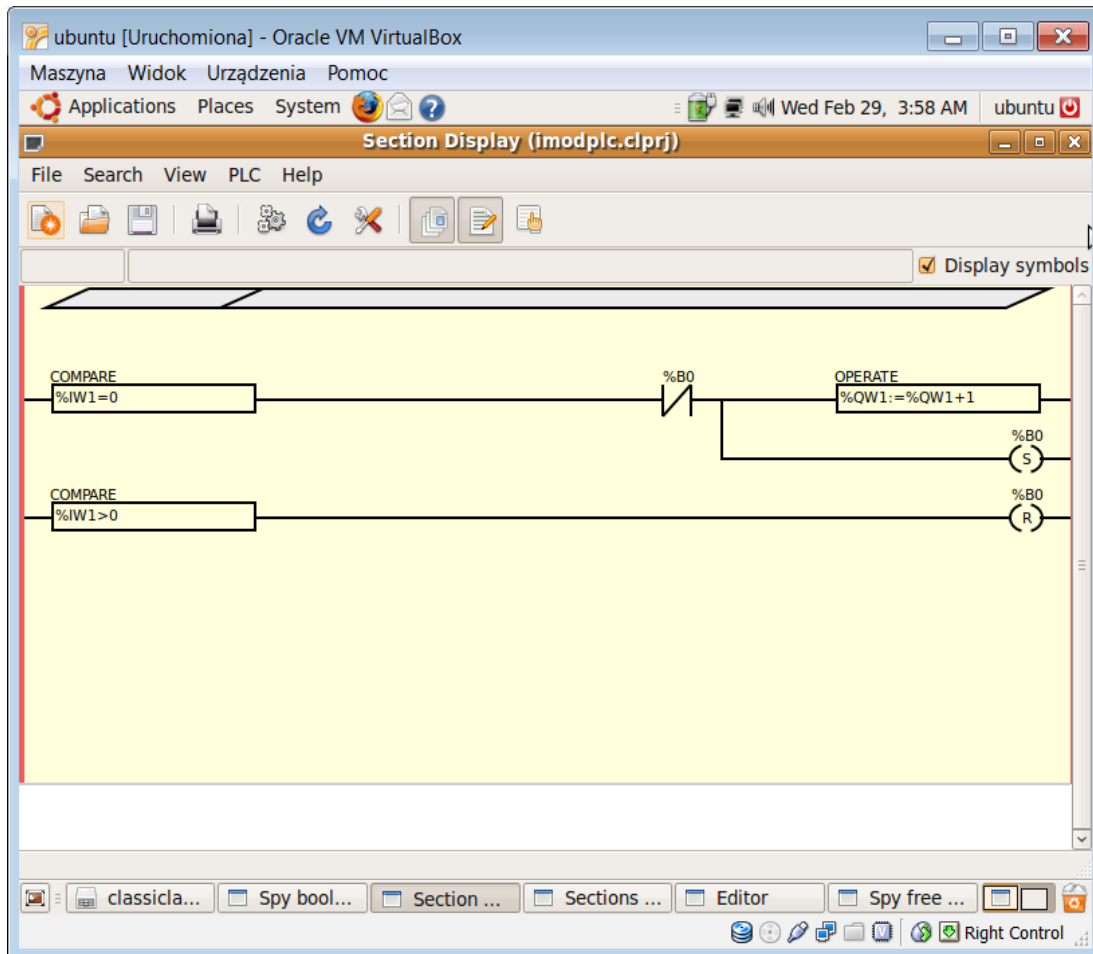
1. the count changes in the status LEDs on the state USER_LED high
2. for the number 10 diosy USER_LED status changes will be sent an e-mail.

First modification of the status LEDs counting USER_LED



In this example, , the application will calculate the amount of state changes and save the diode USER_LED this number to register iModzie.

LadderClassic Structure and configuration



used in this application are two main variables:

- IW1 - LED read-out of USER_LED
- QW1 - save the changesnumber of

and auxiliary variables:

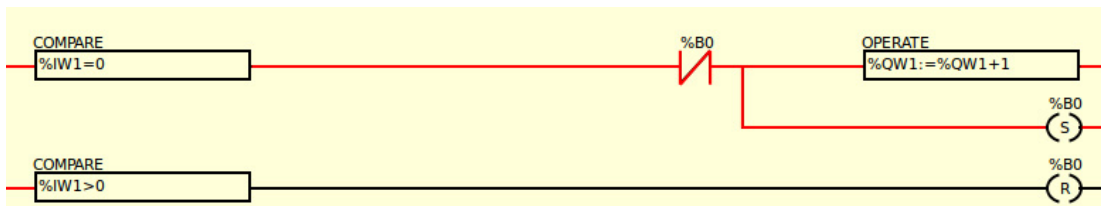
- B0

are used the function blocks:

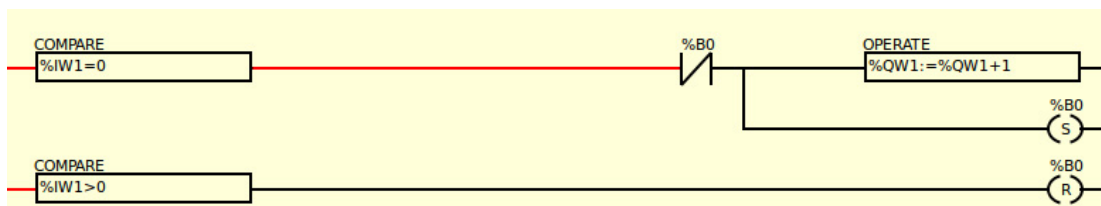
- Comparison Variable
- Variable Assignment
- Set Output
- ResetOutput
- InputNC

Program Description

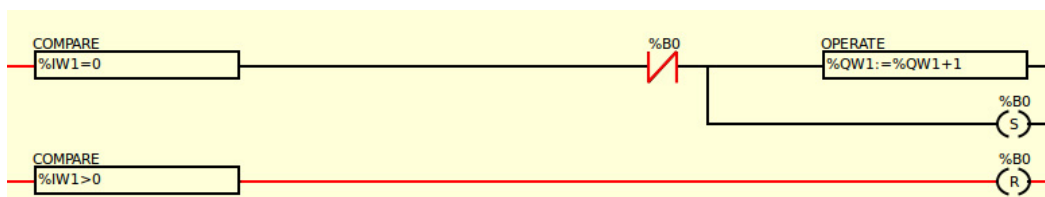
iMod defaults to storing state I / O because the application was set to count changes in USER_LED diode when the diode is taken at 0 (LED off).



- In the time evolution of LEDs USER_LED to 0, the signal will be released with the help of the function block COMPARE% IW1 = 0 through the normally closed contact with the additional variable B0 and reach first the function OPERATE% QW1: =% QW1 1 and immediately afterwards to the coil SET variableB0.



- With OPERATE function key block, variable QW1 will increase by one. SET coil results in a change of variable B0, which will disconnect the coil normally closed to the variable, so that the signal will not reach to the function block OPERATE. This is a security that QW1 parameter was increased by only onepoint.



- If you change the LED status USER_LED to 1, the variable B0 is reset by means of a function block% IW1> 0, so that the normally closed solenoid is closed again and the next amendment to the state of LEDs USER_LED to 0 will pass the signal.

Configuration

Configuration modbusowej communication

(PLC → Configuration → Modbus communication setup):

For this example we will connect with the device through the Ethernet port, so Modbus Master Serial Port leave blank, the rest is left to the default settings.

modbusowej addressing configuration

(→ PLC Configuration → Modbus slaves setup):

Slave N°	Slave Address	TCP/UDP mode	Module Informations
0 :	0.0.0.0:502	<input type="checkbox"/> UDP instead of TCP	iMod9300

Configuration parameters modbusowych

(→ Configuration → PLC Modbus I / O setup):

Slave N°	Request Type	1st Modbus Ele.	Nbr of Ele	Logic	1st I/Q/IW/QW mapped
0: 0.0.0.0:502	ReadInputs (to %I)	101	1	<input type="checkbox"/> Inverted	1
0: 0.0.0.0:502	ReadInputs (to %I)	102	1	<input type="checkbox"/> Inverted	1



Pay attention to addressing modbusową. It often happens that the address is shifted by 1 up or down. In this case, addressing ClassicLadder parameters should be postponed for one up off the iMod.

Construction and Configuration iMod

Configuration iMode exclusively through MainConfig.xml file. First you have to configure the access channels and addressing parameters.

Definition channels

of communication as configured in the program modbusowej ClassicLadder, access channel must be created at modbusowym device 0, via the Ethernet port on the 502nd The source parameters of this device iMod.

```
<? Xml version = "1.0" encoding = "UTF-8"?>
<imod Version="1.1.0">
  <group name="Definicje kanalow">
    <access-channel name="Modbus_S1">
      <protocol name="MODBUS" />
      <port> "ET-502-TCP" </ port>
      <property name="device-id" value="0"/>
    </ Access-channel>
    <source-channel name="NPE_io">
      <protocol name="HARDWARE"/>
      <gap> 0 </ gap>
      <cycle> 2 </ cycle>
    </ Source-channel>
  </ group>
</ iMod>
```

Definition of parameters

create two parameters, one responsible for the status LED read-out values USER_LED (parameter d 100), one in which you saved the amount of change of states (the id parameter 101).

```
<IMod version = "1.1.0">
  <group name="Definicje kanalow">

    <parameter>
      <id> "100" </ id>
      <description> "USER_LED" </ description>
      <source-channel channel-name="NPE_io" parameter-id="USER_LED"/>
      <access-channel channel-name="Modbus_S1" parameter-id="100">
        <property name="varspace" value="DISCRETE"/>
      </ Access-channel>
    </ Parameter>

    <parameter>
      <id> "counter" </ id>
      <access-channel channel-name="Modbus_S1" parameter-id="101"/>
    </ Parameter>

  </ group>

</ iMod>
```

Verification example,in

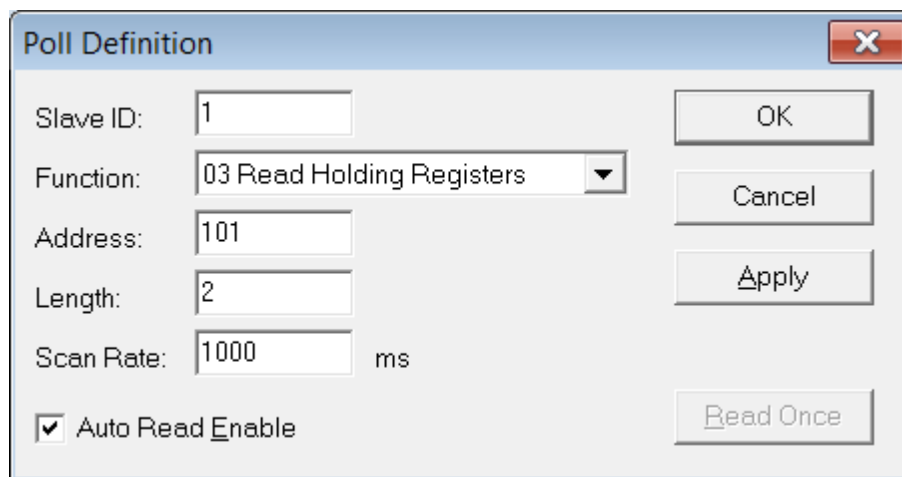
For verifying the correct application example will use a master Modbus - Modbus Poll

Modbus Poll application is available online at:

http://www.modbustools.com/modbus_poll.asp

In addition,upload the configuration file discussed earlier, and run the application *arm-classicladder* together with the draft generated by ClassicLadder.

Establishing a connection

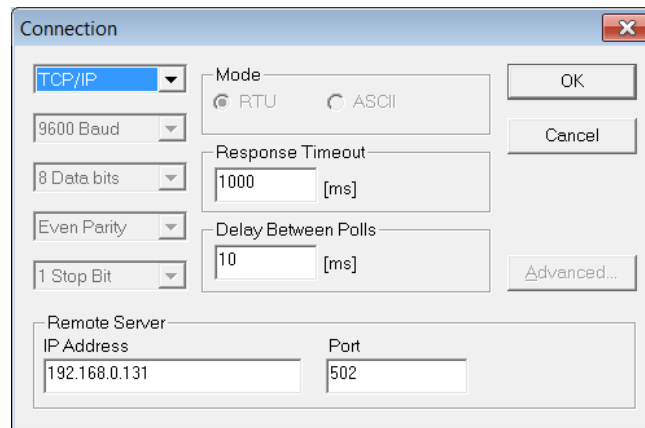


The screenshot shows a dialog box titled "Poll Definition" with the following fields and controls:

- Slave ID: 1
- Function: 03 Read Holding Registers
- Address: 101
- Length: 2
- Scan Rate: 1000 ms
- Auto Read Enable
- Buttons: OK, Cancel, Apply, Read Once

must first be set parameters which will be by application Modbus Poll. To do this, go to the Poll Window Definition(*Setup-> PollDefinition*).

In this example, read the slave device that is iMod. His address modbusowy is 1, we use function code modbusowej 03, reads the address registers 100, 101 and 102 because we start scanning for 101 and three addresses (modbusowe shift occurs here.)



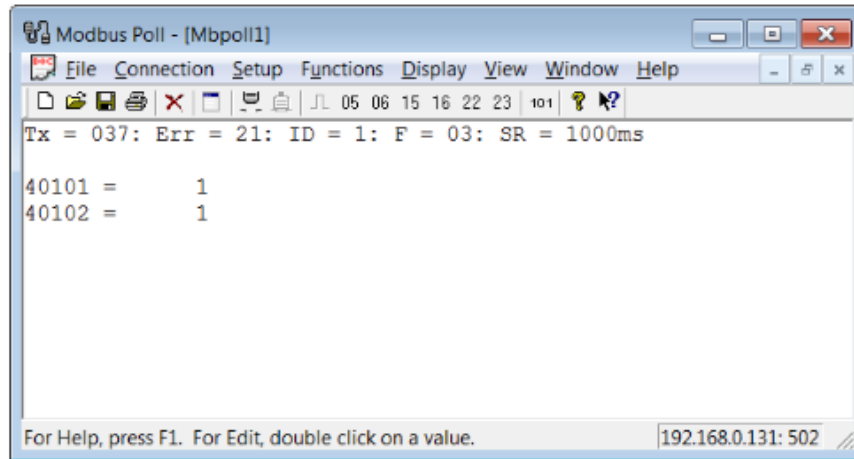
The next step is to connect to iMod'em (Connection -> Connect .)

According to the configuration file, we connect via TCP / IP and port 502nd



IP default iMod telemetry module are:
192.168.0.101

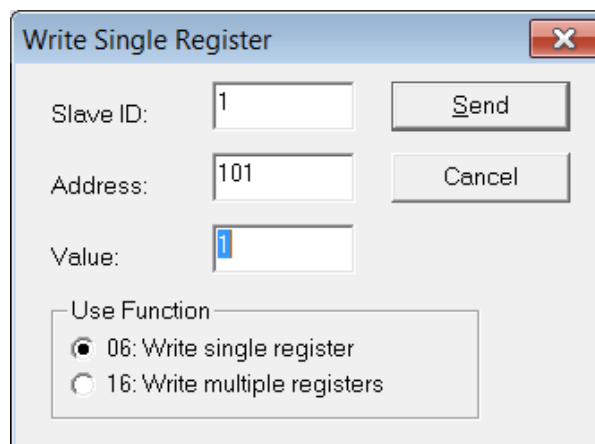
verification of the example



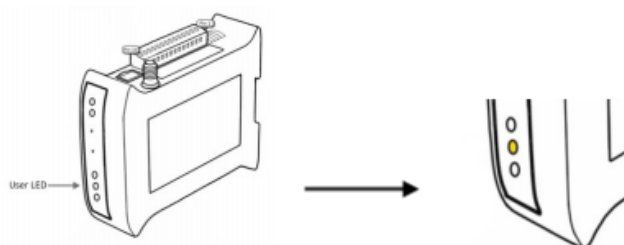
After successful combination should display a window like this (depending on the version of Modbus Polla window may vary) .

description of the parameters in the Modbus Poll:

Address	Description
101	LED USER_LED
102	Counter - indicates the number of changes the status LEDs

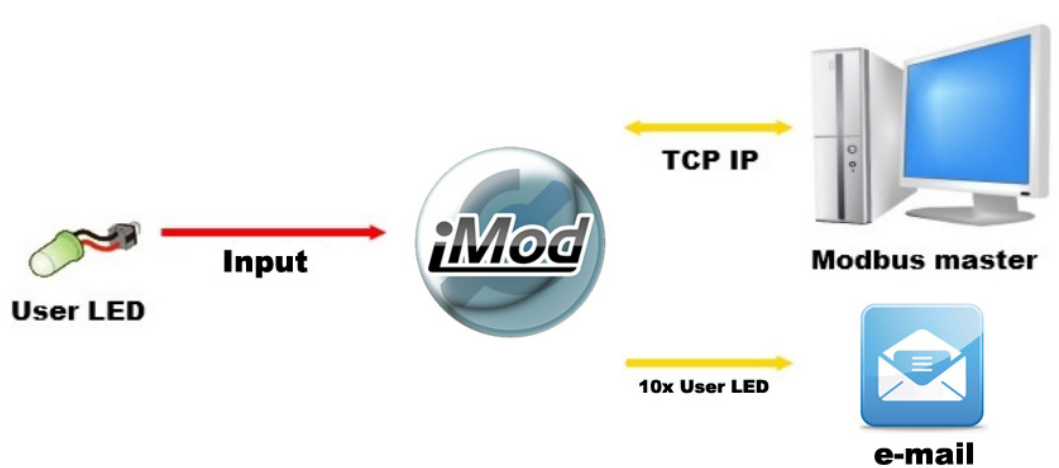


to change the selected parameter, double-click on the address or the displayed value.



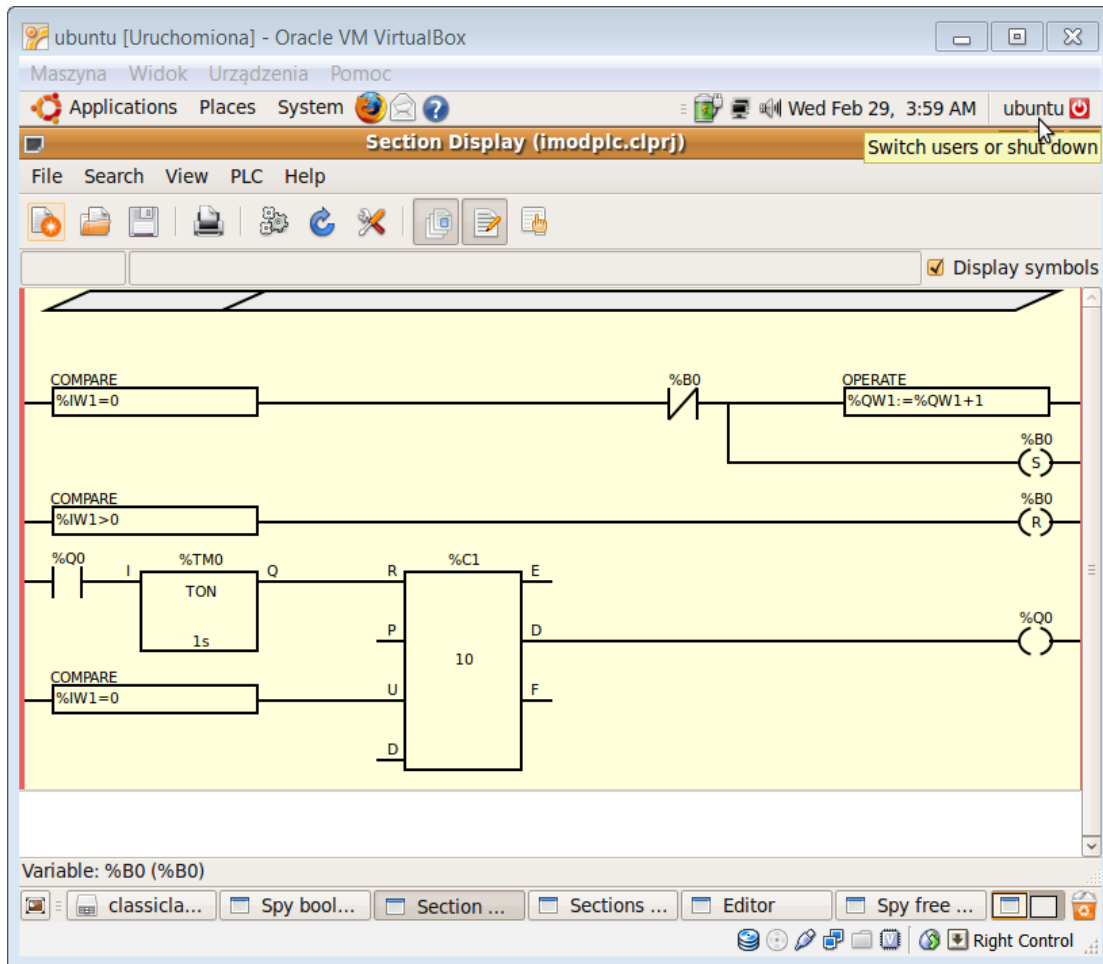
To change a parameter value, enter it in the Value window send to device using the Send button. Status LEDs on the device USER_LED should change, and the parameter value 102 should increase by one point.

Second e-mail notification at 10 times the change of status LEDs USER_LED



application will send an e-mail to the address indicated in the configuration file at a time when the value changes LED status exceeds the 10th USER_LED. When sending e-mail, the number of changes in diode is reset and the counting is done from the beginning.

Structure and configuration Classic Ladder

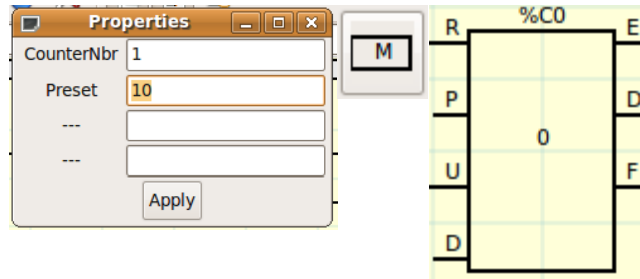


This application is an enhanced version of the application from the previous example, because all parameters are the same. In addition to the variables which are presented in the previous example, there is an additional variable:

- `Q0` - the variable is responsible for sending an email

Program Description

This program counts the number of status changes USER_LED diode, which diode 10 changes state to state, followed by e-mail notification of predefined recipients .



It is used here for the function block counter (Counter Block) and timer (Timer Block IEC). The block counter is set to run after and send a signal to the coil% Q0 when the number of counts is greater than



1. 10when USER_LED status is zero, the signal comes to the counter to U-joints, raising the number 1 (to be paid note that this is a one-time change and security need not be applied as before).
2. At the time when the number of changes will be 10, the output signal D will be announced at launch coil% Q0. The parameter Q0 is responsible for sending mail.
3. Commissioning of high resets the counter to be able to start counting from the beginning. However, before the counter is reset, the state of the high passes to the timer, which is set to one second.
4. Only after one second timer is reset. Please use it too fast because the counter reset, it may not have time to read iMod parameter Q0 and thus send a message.

Configuration

and communication configuration modbusowej modbusowej addressing configuration remains unchanged.

For modbusowych parameter configuration, add a new variable (PLC → Configuration → Modbus I / Osetup)

Slave N°	Request Type	1st Modbus Ele.	Nbr of Ele	Logic	1st I/Q/IW/QW mapped
0: 0.0.0.0:502	ReadInputRegs (to %IW)	101	1	<input type="checkbox"/> Inverted	1
0: 0.0.0.0:502	WriteHoldRegs (from %QW)	102	1	<input type="checkbox"/> Inverted	1
0: 0.0.0.0:502	WriteCoils (from %Q)	103	1	<input type="checkbox"/> Inverted	0



Pay attention to the 1st column mapped - in this case is set to zero so that the variable Q0 begins with the number 0 and not as before, the variables start from 1

Building configuration iMod

iMod configuration file is a file extension the previous example. Channels and parameters, add the appropriate lines of code.

Definition

Channel definition channels to add a notification channel, which will be in the configuration of mail and email message content channel.

```
<message-channel name="Email_sender">
  <protocol name="EMAIL">
    <property name="user" value="testnpe"/>
    <property name="password" value="123npe"/>
  </ Protocol>
  <port> "poczta.o2.pl" </ port>
  <recipient> "info@a2s.pl" </ recipient>
</ Message-channel>

<message id="Mess_1">
  <! [CDATA [
    "USER_LED change state 10 times"
```

```
    ]]>  
</ Message>
```

The definition of parameters

to get the functionality to send e-mail after 10 times you send an e-mail, add the parameter responsible for sending e-mail after the change of state.

```
<parameter>  
  <id> "10" </ id>  
  <access-channel channel-name="Modbus_S1" parameter-id="102">  
    <property name="varspace" value="COIL"/>  
  </ Access-channel>  
  <Event type="OnChange">  
    <message-channel channel-name="Email_sender"/>  
    <message-id> "Mess_1" </ message-id>  
  </ Event>  
</ Parameter>
```

Verification example,in

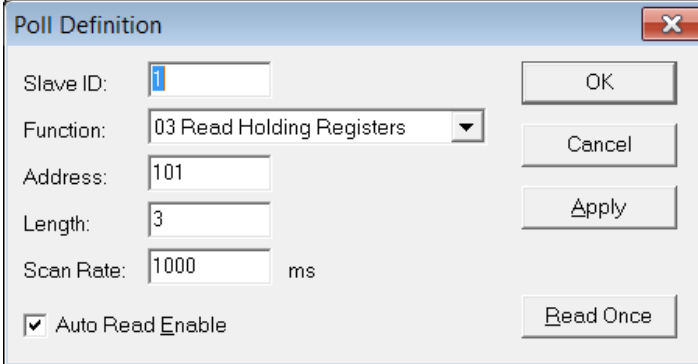
For verifying the correct application example will use a master Modbus - Modbus Poll

Modbus Poll application is available online at:

http://www.modbustools.com/modbus_poll.asp

Additionally, you should upload the configuration file discussed earlier and run the application arm-classicladder together with the draft generated by ClassicLadder.

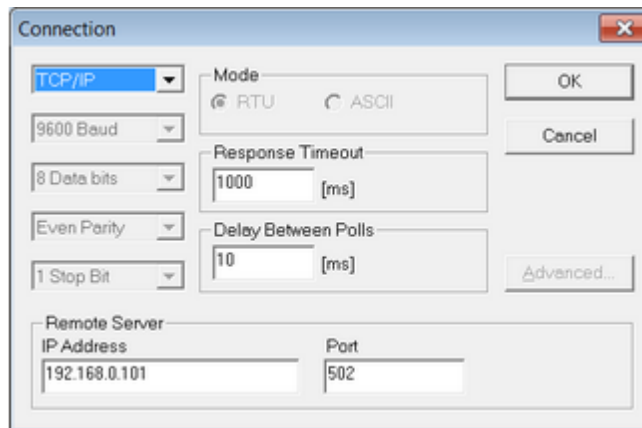
Establishing a connection



The screenshot shows the 'Poll Definition' dialog box. The fields are: Slave ID: 1, Function: 03 Read Holding Registers, Address: 101, Length: 3, Scan Rate: 1000 ms. The 'Auto Read Enable' checkbox is checked. Buttons include OK, Cancel, Apply, and Read Once.

must first be set parameters which will be by application Modbus Poll. To do this, go to the Poll Window Definition(*Setup-> PollDefinition*).

In this example, read the slave device that is iMod. His address is 0 modbusowy We use function code 03 modbusowej Addresses for the reading to the address registers 100, 101 and 102 because zaczynij scanning for 101 and three addresses (modbusowe shift occurs here.)



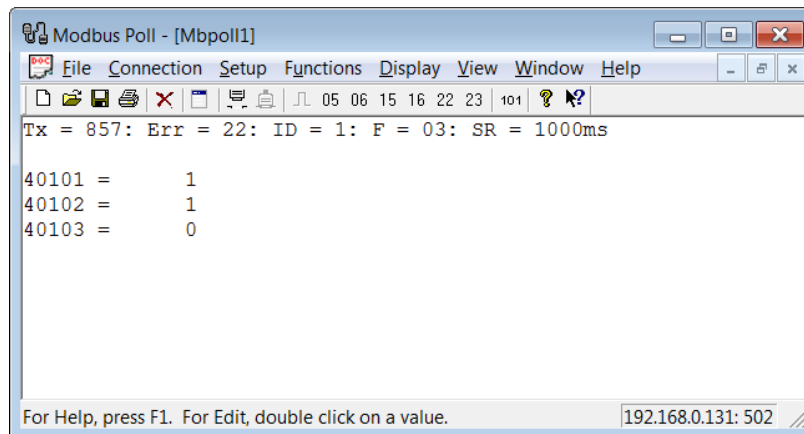
The next step is to connect to iMod'em (Connection -> Connect).

According to the configuration file, connect to TCP / IP and port 502nd



IP default iMod telemetry module are:
192.168.0.101

verification of the example



After successful combination should display a window like this (depending on the version of Modbus Polla window may be different).

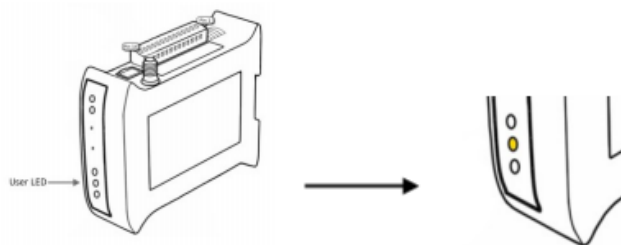
description of the parameters in the Modbus Poll:

Address	Description
101	LED USER_LED
102	Counter - indicates the number of changes the status LED
103	Counter - when this parameter reaches a value of 10 will send a notification email and the value is reset

The screenshot shows a dialog box titled "Write Single Register" with a close button (X) in the top right corner. It contains the following fields and options:

- Slave ID: 1
- Address: 101
- Value: 1
- Buttons: Send, Cancel
- Use Function:
 - 06: Write single register
 - 16: Write multiple registers

to change the selected parameter, double-click on the address or the displayed value.



To change a parameter value, enter it in the Value box then send to the device through the Send button. Status LEDs on the device USER_LED should change, and the value of the parameters 102 and 103 should increase by one point.

At 10 times the change of status LEDs USER_LED iMod application sends an e-mail to the address defined.























Email_sender: 101

Od: testnpe@o2.pl ([więcej](#)) dnia: 13 marca 2012 13:36

USER_LED change state 10 times

Appendix A: Description of the function blocks the Classic Ladder

Symbol	Description	Function
	Normally open	signal leads the (closes contacts), the value of the variable is assigned a logical "1"
	Normally closed	Normally closed. Leads the signal (contact closes) when the value of the variable is assigned a logical "0"
	contact with the rising edge of	this contact sends a pulse when the status changes from open to close
	contact with the falling edge of	this contact sends a pulse when the state changes from a closed to an open
	line connecting horizontal	The horizontal line connecting the elements of the
	line connecting the vertical	vertical line connecting the elements of
	a horizontal line connecting the end	creates a horizontal line to the end of
	the timer function block	Timer, selectable timer name, value, fixed, and mode: <ul style="list-style-type: none"> • ON • OFF
	function block	Counter R-counter reset counter P-sets to the reference value to output was 1 U-counting up the D on the left - counting down the D on the right - the output when the counted value is equal to or greater than the set point is the state of a
	comparison function block	using the block You can use the mathematical comparison of

	the timer function block	Timer counts down the time specified.
	monostable block	This block sets the output to "1" for a certain time and then obtains the output state "0"
	normally open coil	relaysets the value assigned to the variable to "1", when you are given a signal. This is a relay with the contacts open.
	Normally closed coil	relaysets the value assigned to the variable to "0" when you are given a signal. It is a compact relay with contacts.
	coil setting	Adjustablerelay"SET". The relay is set to "1" variable.
	Reset coil	relayadjustable "RESET" designed work towith the relay, "SET". When the relay is given the signal value of the variable assigned to it will be set to "0".
	Coil jump	Jump to the selected line
	coil call	call the selected line
	operation	The ability to perform mathematical operations
	Gross	Net