

Introduction	2
Physical layout.....	3
Hardware & software requirements	4
Software installation.....	4
Settings	4
Changing the multiLINK's IP address	5
Configuring the multiLINK's serial ports.....	5
Other configuration	5
Bridging and forwarding (Using the multiLINK as serial to TCP/UDP gateway).....	6
Modbus RTU to Modbus UDP	6
Modbus RTU to Modbus TCP	7
M-Bus over UDP	8
Schematic example; connecting 160 M-Bus water meters through a multiLINK	9
M-bus to Modbus TCP.....	12
Using the embedded webserver.....	13
Creating and uploading HTML pages	13
Using the HTML editor.....	13
Using the converter.....	14
Sending alarm messages.....	15
Prerequisites.....	15
Setting up alarms during programming.....	15
Configuring recipients on the multiLINK.....	16
Email settings.....	16
SMS settings.....	17
History	17
Restoring factory settings.....	18



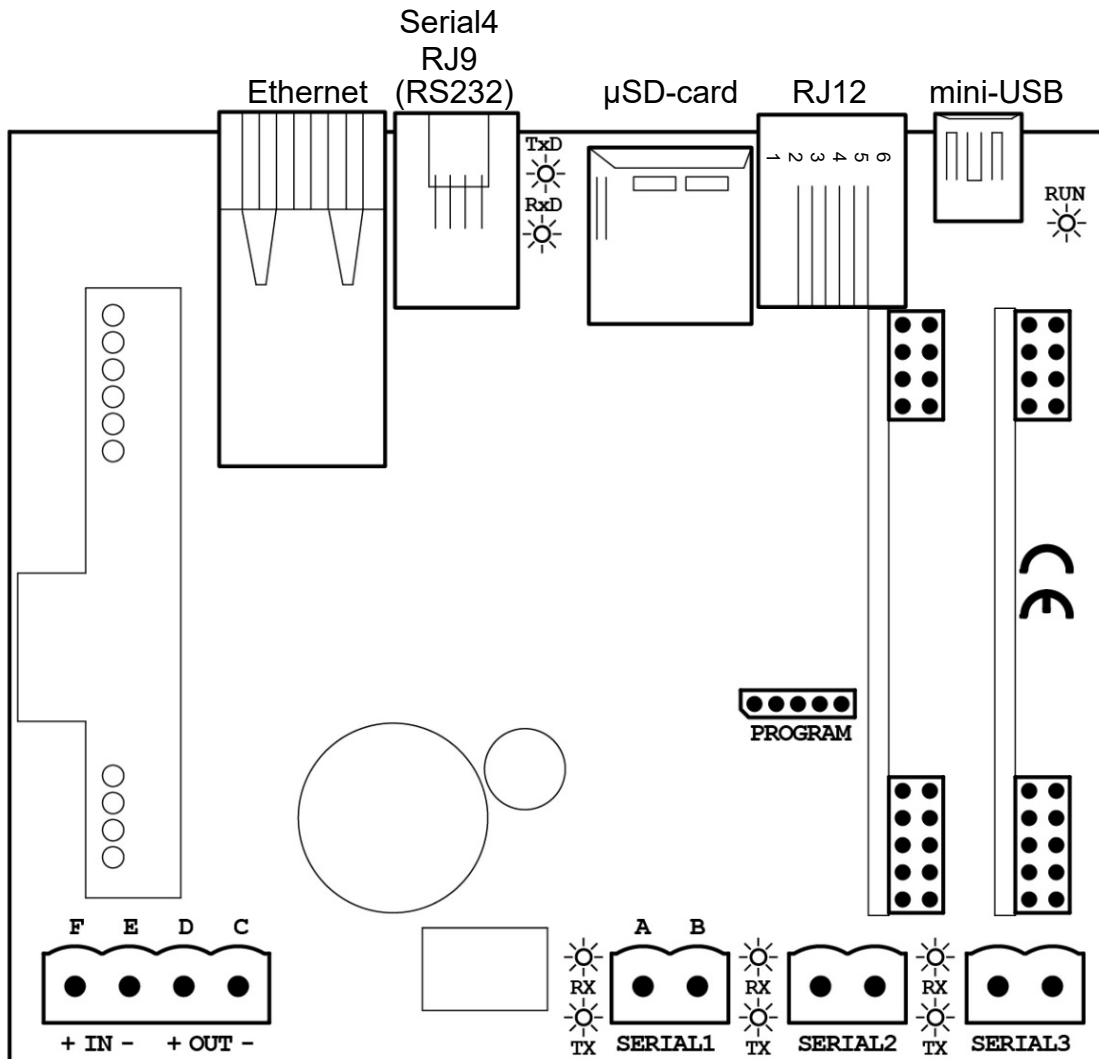
The Fidelix multiLINK is a freely programmable Modbus and M-Bus protocol converter with a free physical configuration. There are 24 different versions of the multiLINK available. The basic version is always configured with 1 LAN port (Ethernet) 1 Modbus RTU port (RS485). Upon this, two extra ports can be added.

Currently, we offer Modbus and M-bus ports to be added into the two extra slots of the multiLINK. You can either choose 1 or 2 additional Modbus ports, 1 or 2 additional M-bus ports or a combination of 1 additional Modbus and 1 additional M-bus port.

In addition to these two ports, the multiLINK can be equipped with a POE (power over Ethernet) LAN-port and/or a webserver.

The optionally integrated webserver can show information about the devices connected to the fieldbus. The pages to load to the multiLINK's webserver are made using the Fidelix Graphics Editor and can be viewed with a standard web browser.

Physical layout



The Ethernet connector is used to connect the multiLINK to the Internet or the local network. The orange signal LED on the left (closest to the other ports) is lit when a connection has been established, and blinks when data is transferred within the network. The green LED on the right (furthest away from the other ports) is lit when the connection speed is 100Mbps. It is not lit at 10 Mbps.

The RJ9 serial RS232 port (Serial port 4) is normally used as a debug connector, through which the multiLINK can be connected to a computer's serial port through telnet. It can also be used for connecting RS-232 level Modbus devices or an external alarm modem (for instance a Cinterion MC55iT). The signal LEDs next to the connector blink when sending and receiving data.

The μSD memory card reader can be used to extend the available memory capacity. (PLEASE NOTE: Not available in hardware version 1.0!)

The RJ12-port is internally connected in parallel with the first fieldbus, just like is the case for the multi24. This can for instance be used to connect a handheld multiDISPLAY. The cables are connected as following: 1 - Modbus B, 2 - unused, 3 - unused, 4 - 24V power supply, 5 - ground of the power supply, 6 - Modbus A.

The mini USB port is used for software or firmware updates. This can also be done through FTP.

The power connectors F and E are used to power the multiLINK. Operating voltage requirements are 24 VDC for a multiLINK with an M-bus port, and 10-36 VDC or 12-24 VAC for a multiLINK with only Modbus ports.

The power connectors D and C are used to power other devices. The voltage is the same as the voltage set on connectors F and E, or, when powered by PoE, 24 VDC. The power consumption may not exceed 0.8A when using an external power supply, or 0.3A when powered by PoE.

The Serial1 RS485 connector is the Modbus port that is on the basic version of every multiLINK. It is a galvanically isolated connector.

Serial ports 2 and 3 are the ports you can choose to equip with a Modbus or M-bus card.

Hardware & software requirements

To program a multiLINK, you need the following:

- An Ethernet cable
- A web browser

If you also want to use the webserver functionality of the multiLINK, you will need, in addition:

- A PC running Windows
- The Fidelix Graphics Editor (HTML Editor)
- The multiLINK graphics converter program

Both can be downloaded from the [Fidelix partner page](#).

Software installation

The installation and basic functionality of the Fidelix Graphics Editor (HTML Editor) is described in the getting started guide that is included in the installer zip file.

To install the multiLINK graphics converter program, simply download the folder, unblock it like described in the getting started guide of the graphics editor, and you are good to go!

Settings

To configure your multiLINK, you can connect to your multiLINK either through the Ethernet port, or using the RJ9 / RS232 port.

Through the Ethernet port you can connect with your browser or through telnet.

The factory settings IP address is 10.100.1.97, as is mentioned on the side of your multiLINK.

The multiLINK does not offer any form of security when connecting to it through a browser, when using the default settings. This means that your network administrator should provide the security on the network, or you can activate a user password through the administration menu using telnet.

To change settings from your browser, surf to 10.100.1.97/settings.htm to enter the configuration menu.

To change settings through telnet, connect to 10.100.1.97 using username "ADMIN" and password "multiWEB" to enter the configuration menu.

To enter the settings menu through the RJ9 debug port, use a terminal program (like Hyper Terminal or Tera Term) and connect with 115 200 bps. Once connected, you can open the menu by simply pressing "enter". The displayed menu is the same as the one you get when connecting through telnet.

Changing the multiLINK's IP address

From the browser, click "LAN Settings" or browse directly to 10.100.1.97/LanSettings.htm. Make all appropriate changes you want to and click "Save Changes".

With telnet through Ethernet or with a terminal connection from the debug port, enter the "Configure LAN interface" menu by entering "1" + "enter". Now enter "set_ip_add xxx.xxx.xxx.xxx" + "enter". Then, apply the new IP address by entering "save" + "enter".

The multiLINK will immediately change its IP address, but you have to connect to that new IP address to confirm the changed address. The multiLINK will keep the "temporary new IP address" for two minutes, during which you have to validate the new address. If this is not done within two minutes, the multiLINK will revert back to its old IP address. This to ensure the new settings actually work, and no mistakes have been made in the network security setup. Also, if somehow a mistake was made while entering the new IP settings, one only has to wait for two minutes to be able to connect to the old address again.

From the browser, browse to the new IP address and again enter the LAN settings and click "Validate settings". From telnet, enter the LAN configuration menu at the new IP address and enter "validate" + "enter", or from a terminal connection, simply re-enter the menu and validate.

NOTE: because configuration is more user friendly and comprehensibly presented on the settings webpages, we will not describe any other settings through the telnet / terminal menu in this manual. The menu is so clear that it is not necessary to explain it any more than it is inside the menu itself.

Configuring the multiLINK's serial ports

Click "Serial Settings" on the settings menu page, or browse directly to 10.100.1.97/SerialSettings.htm on your multiLINK to configure the serial ports.

"UDP to Serial" is used to bridge Modbus RTU or M-bus data to UDP over Ethernet.

"Modbus master" should be selected when using the multiLINK as stand-alone Modbus master (for instance when you set up the multiLINK with a webserver and only need to be able to consult the measurements from the multiLINK), or to bridge Modbus RTU data to Modbus TCP over Ethernet. Port 4 can be used as RS232 debug port, or to control an MC35iT or MC55iT GSM modem for sending SMS alarms.

The 120Ω termination should be enabled when the multiLINK is the first or last device in the bus.

The remote IP and port must be correctly configured only when using the multiLINK as bridge or gateway between the serial port and TCP or UDP.

Other configuration

All configuration can be done from your browser. For completeness, a list of all available pages:

- | | |
|----------------------|---|
| - LanSettings.htm | Changing network settings |
| - SerialSettings.htm | Changing serial port settings |
| - CommList.htm | A list of the data from and to the field busses |
| - LANStatistics.htm | Network traffic information |
| - Settings.htm | "Home" page of the settings |
| - Email.htm | Email and SMS settings |

Bridging and forwarding (Using the multiLINK as serial to TCP/UDP gateway)

The most common use for the multiLINK is to offer additional serial ports, and read them from the Ethernet side over TCP or UDP.

Modbus RTU to Modbus UDP

Set up your multiLINK's port as "UDP to serial".

The physical serial ports are bridged to UDP ports as follows: Serial port 1 → UDP port 10001, Serial port 2 → UDP port 10002, Serial port 3 → UDP port 10003 and Serial port 4 → UDP port 10004. This means that the external controller reads UDP ports 10001-10004 of the multiLINK when reading the Modbus registers of the connected devices.

On the multiLINK side, specify the UDP ports your controller will be using to read the Modbus registers. In the case of an FX-controller, the virtual ports 6-10 are connected to UDP ports 10006-10010.

Example with multiLINK at 10.10.11.144 and FX-controller at 10.10.11.48:

Mode

UDP to Serial

Serial speed

57600

Data bits

8 bits

7 bits

Parity

None

Even

Odd

Stop bits

1

1.5

2

120ohm terminating

enabled

disabled

Remote IP

10.10.11.48

IP of FX-controller

Remote Port

10006

Port number on FX-controller + 10 000

Alarms

Settings

Ports

Suomi (FI)

English (EN)

Swedish (SE)

Dutch (NL)

French (FR)

Russian (RU)

Change

Port settings

Port 0

MBUS

Port 1

Modem

9600

Port 3

Modbus

N82

57600

Send delay (ms)

0

Port 4

Free

N82

19200

Port 5

Free

N82

19200

Port 6

Modbus UDP

Send delay (ms)

0

Port 7

MBUS

Port 8

Modbus TCP

Send delay (ms)

0

Port 9

Free

Port 10

Free

The multiLINK's physical port "Serial1" (UDP port 10001) is bridged to "Port 6" (UDP port 10006) of the FX.

Alarms

Programming

Modbus devices

Add

Change

Delete

Tu 01.12.2015 14:27:29

Modbus devices

Update

Print

06.008

3=HOLDING

1

8

Port=6

Communication Ok

08.010

3=HOLDING

1000

60

Port=8

Communication Ok

Port number

6

Address of modbus device

8

Register type

3=HOLDING REGS (READ=3,WRITE=16)

TCP/IP module

☒

Start register

1

IP of multiLINK

IP address

10.10.11.144

Register count

8

Serial port number on multiLINK + 10 000

IP port

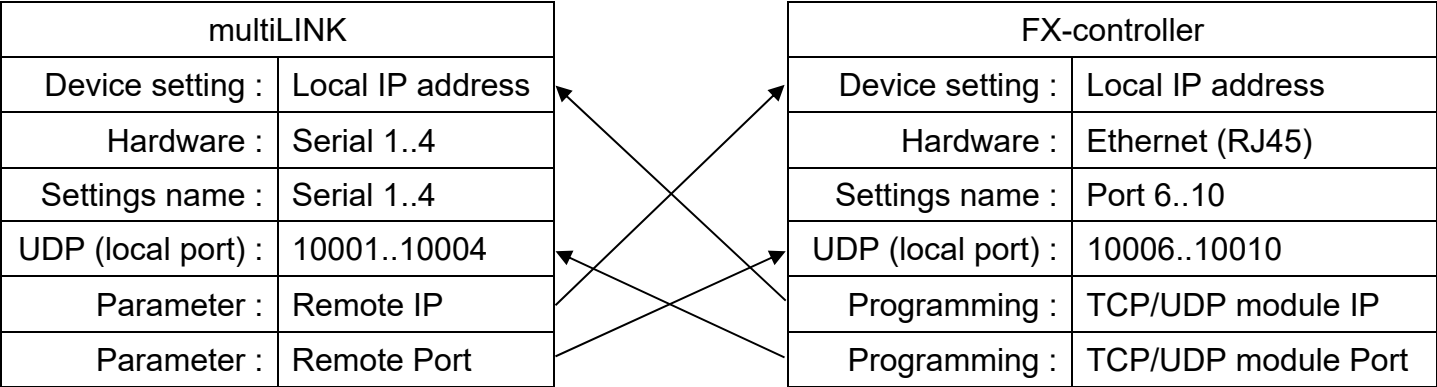
10001

Debug

☐

Send only

☐



Modbus RTU to Modbus TCP

To use this feature, make sure you have a multiLINK that has the web feature enabled. You can verify this by either checking the multiLINK's serial number; for a web server enabled multiLINK, this starts with "MW" (not "MC"), or by logging in through telnet / terminal where the same serial number is listed, and also the "Base Module Type", which for the web server enabled version is "multiWEB" (not "multiCOM").

Set up your multiLINK's port as "MODBUS master".

The physical serial ports are bridged to TCP ports as follows: Serial port 1 → TCP port 502, Serial port 2 → TCP port 503, Serial port 3 → TCP port 504 and Serial port 4 → TCP port 505. This means that the external controller reads TCP ports 502-505 of the multiLINK when reading the Modbus registers of the connected devices.

On the multiLINK side, no external port configuration needs to be done, because the multiLINK is Modbus master itself. Any IP or port values entered will be ignored.

Example with multiLINK at 10.10.11.144 and FX-controller at 10.10.11.48:

Mode

MODBUS master

Serial speed

57600

Data bits

8 bits

7 bits

Parity

None

Even

Odd

Stop bits

1

1.5

2

120ohm terminating

enabled

disabled

Remote IP

Remote Port

Alarms

Settings

Ports

Suomi (FI)

English (EN)

Swedish (SE)

Dutch (NL)

French (FR)

Russian (RU)

Change

Port settings

Port 0

MBUS

Port 1

Modem

9600

Port 3

Modbus

N82

57600

Send delay (ms) 0

Port 4

Free

N82

19200

Port 5

Free

N82

19200

Port 6

Modbus UDP

Send delay (ms) 0

Port 7

MBUS

Port 8

Modbus TCP

Send delay (ms) 0

Port 9

Free

Port 10

Free

Alarms

Programming

Modbus devices

Add

Change

Delete

Tu 01.12.2015 15:21:54

Modbus devices

Update

Print

06.008

3=HOLDING

1

8

Port=6

Communication Ok

08.010

3=HOLDING

1000

60

Port=8

Communication Ok

Port number 8

Address of modbus device 10

Register type

3=HOLDING REGS (READ=3,WRITE=16)

TCP/IP module

☒

Start register

1000

IP of multiLINK

IP address

10.10.11.144

Register count

60

Serial port number on multiLINK + 501

IP port

504

Debug

☐

Send only

☐

Note that with this setup, all connected registers can also be displayed locally, if a webserver has been configured in your multiLINK. There are however significant delays in displaying values, as they have to be processed on both ends (RTU and TCP). Also, any value changes done locally on the multiLINK will be overwritten by the external TCP master, so only use this to display values, not to change them locally.

multiLINK		FX-controller	
Device setting :	Local IP address	Device setting :	Local IP address
Hardware :	Serial 1..4	Hardware :	Ethernet (RJ45)
Settings name :	Serial 1..4	Settings name :	Port 6..10
TCP (local port) :	502..505	TCP (local port) :	N/A
Parameter :	Remote IP: N/A	Programming :	TCP/UDP module IP
Parameter :	Remote Port: N/A	Programming :	TCP/UDP module Port

M-Bus over UDP

Set up your multiLINK's port as "UDP to Serial".
The physical serial ports are bridged to UDP ports as follows: Serial port 2 → UDP port 10002, Serial port 3 → UDP port 10003 and Serial port 4 → UDP port 10004. This means that the external controller reads UDP ports 10002-10004 of the multiLINK when reading the M-bus registers (or indexes or records) of the connected devices.

On the multiLINK side, specify the UDP ports your controller will be using to read the M-bus registers (or indexes or records).

In the case of an FX-controller, the virtual ports 6-10 are all connected to UDP port 10002.

NOTE: This means that when several ports are configured to connect all M-bus devices, they will all be polled one by one. This also means that on the multiLINK, only one port will be communicating at a time, making the polling of for instance 124 water meters a slow process.

Example with multiLINK at 10.10.11.143 and FX-controller at 10.10.11.48

Mode	UDP to Serial		
Serial speed	2400		
Data bits	☒ 8 bits ☐ 7 bits		
Parity	☐ None ☒ Even ☐ Odd		
Stop bits	☒ 1 ☐ 1.5 ☐ 2		
120ohm terminating	☐ enabled ☒ disabled		
Remote IP	10.10.11.48 IP of FX-controller		
Remote Port	10002 Always port 10002		

The multiLINK's physical port "Serial3" (UDP port 10003) is bridged to "Port 7" (UDP port 10002, because it is configured as M-Bus port) of the FX.

Alarms

Settings

Ports

Suomi (FI)

English (EN)

Swedish (SE)

Dutch (NL)

French (FR)

Russian (RU)

Change

Port settings

Port 0	MBUS			
Port 1	Modem	9600		
Port 3	Modbus	N82	57600	Send delay (ms) 0
Port 4	Free	N82	19200	
Port 5	Free	N82	19200	
Port 6	Modbus UDP			Send delay (ms) 0
Port 7	MBUS			
Port 8	Modbus TCP			Send delay (ms) 0
Port 9	Free			
Port 10	Free			

Important here is to set the M-Bus device number (found on the device itself), tick the "Use device number" tick box, and set the receive timeout to a significant value. By default, the timeout is "0", which will cause the communication to fail of course.

Alarms

Programming

Modules

Add

Change

Delete

We 02.12.2015 14:50:07 I/O modules

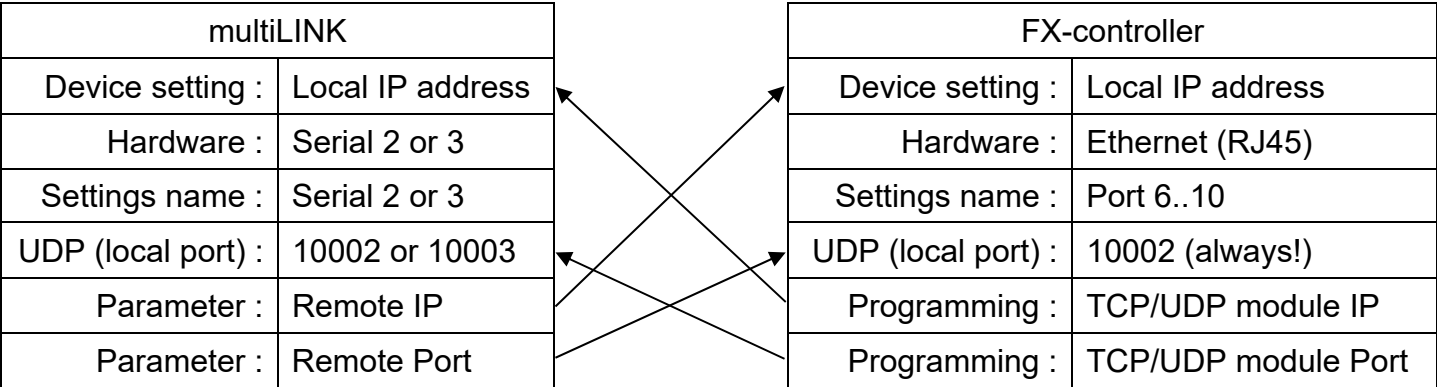
Update

Print

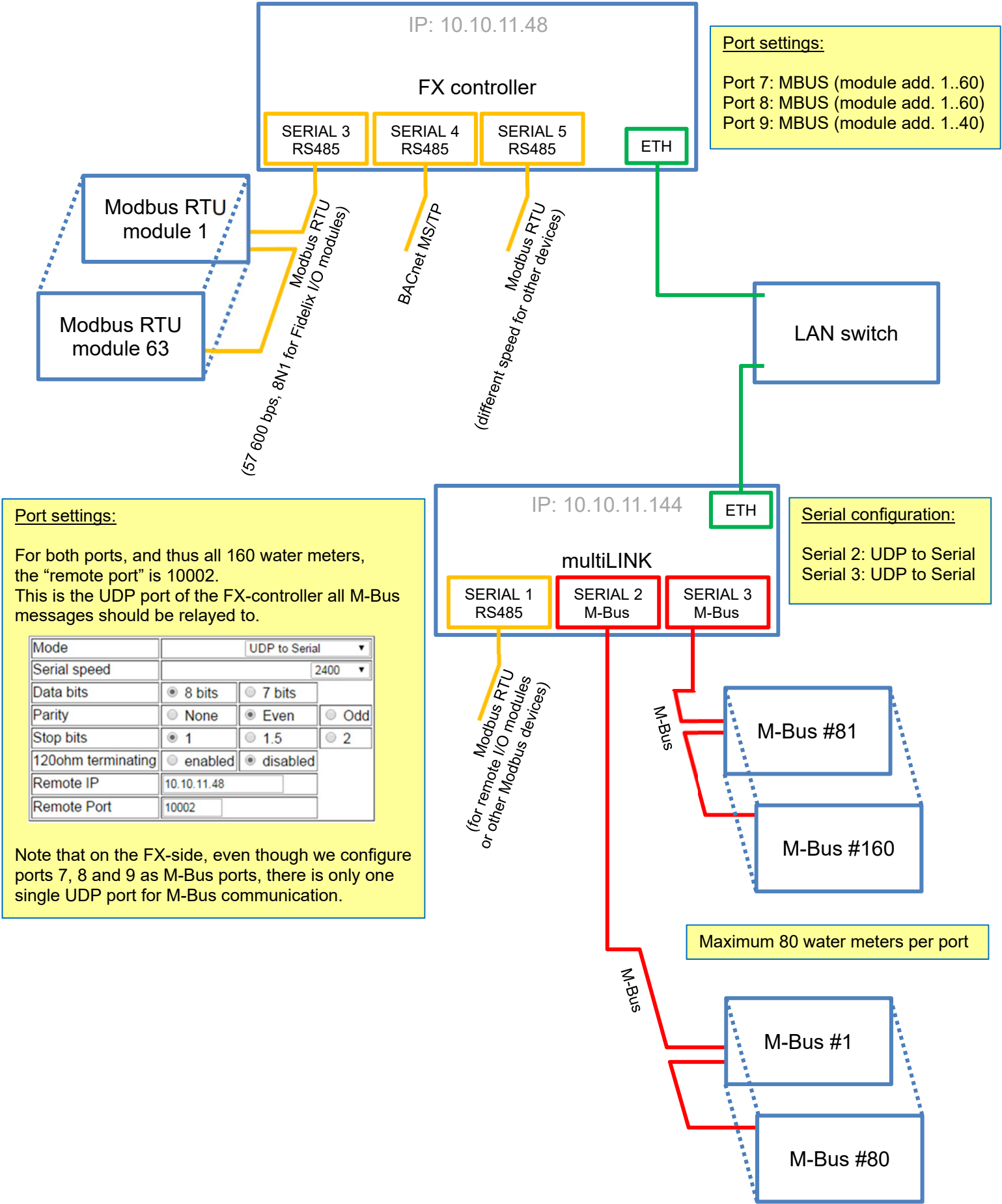
Port.Module 07.001 Type MBUS Version V0.00 Communication Ok Unknown Errors 46.73%

Port number 7 Module address 1 Type MBUS

Poll time (ms)	151	TCP/UDP module	☒	IP of multiLINK
Message count	1704	IP address	10.10.11.143	
Errors now	0	IP port	10003	
Errors max	72	Debug		
Errors total	1495			
Receive timeout (ms)	1000			
M-BUS device number	12003203	Use device number	☒	
M-BUS address	0	Send Reset command	☐	



Schematic example; connecting 160 M-Bus water meters through a multiLINK



Water meters 1..60 need to be configured as modules 1..60 on port 7 on the FX:

Th 17.08.2017 17:23:35 I/O modules Update

Port.Module	Type	Version	Communication	Status	Errors
Port.Module 07.001	MBUS	V0.00	Ok	Unknown	6.07%
Port.Module 07.002	MBUS	V0.00	Ok	Unknown	0.00%

Port number **7** Module address **2** Type MBUS

Poll time (ms) 0 TCP/UDP module ☒ Message count 209314 IP address 10.10.11.114 Errors now 0 IP port 10002 Errors max 1 Errors total 7 Debug ☐ Receive timeout (ms) 500

M-BUS device number **12009646** Use secondary address ☒ M-BUS address 0 Send Reset command ☐

Clear error counters Find modules

Virtual port 7 on the FX

Each water meter gets a virtual "Module address" that is used in the IEC code where the value is read from the device (GetMBUSValueF)

Module is connected via Ethernet to multiLINK IP

multiLINK serial port 2 = UDP port 10002

Using secondary address means we read the first M-Bus record to choose the right device.

Each M-Bus device always has 1 record containing its own serial number

Water meters 61..80 need to be configured as modules 1..20 on port 8 on the FX:

Th 17.08.2017 20:39:28 I/O modules Update

Port.Module	Type	Version	Communication	Status	Errors
Port.Module 07.001	MBUS	V0.00	Ok	Unknown	6.07%
Port.Module 08.016	MBUS	V0.00	Ok	Unknown	0.00%

Port number **8** Module address **16** Type MBUS

Poll time (ms) 0 TCP/UDP module ☒ Message count 209314 IP address 10.10.11.114 Errors now 0 IP port 10002 Errors max 1 Errors total 7 Debug ☐ Receive timeout (ms) 500

M-BUS device number **12009653** Use secondary address ☒ M-BUS address 0 Send Reset command ☐

Clear error counters Find modules

Virtual port 8 on the FX

Each water meter gets a virtual "Module address" that is used in the IEC code where the value is read from the device (GetMBUSValueF)

Module is connected via Ethernet to multiLINK IP

multiLINK serial port 2 = UDP port 10002

Using secondary address means we read the first M-Bus record to choose the right device.

Each M-Bus device always has 1 record containing its own serial number

At this point, all 80 water meters that are connected to serial port 2 on the multiLINK are configured. Let's move on to the 80 water meters connected to serial port 3 on the multiLINK;

Water meters 81..120 need to be configured as modules 21..60 on port 8 on the FX:

Th 17.08.2017 20:49:59 I/O modules Update

Port.Module 07.001	Type MBUS	Version V0.00	Communication Ok	Unknown	Errors 6.07%
Port.Module 08.058	Type MBUS	Version V0.00	Communication Ok	Unknown	Errors 0.00%

Port number **8** Module address **58** Type MBUS

Poll time (ms)	0	TCP/UDP module	☒
Message count	209314	IP address	10.10.11.114
Errors now	0	IP port	10003
Errors max	1		
Errors total	7	Debug	☐
Receive timeout (ms)	500		
M-BUS device number	12024587	Use secondary address	☒
M-BUS address	0	Send Reset command	☐

Clear error counters Find modules

Virtual port 8 on the FX

Each water meter gets a virtual "Module address" that is used in the IEC code where the value is read from the device (GetMBUSValueF)

Module is connected via Ethernet to multiLINK IP

multiLINK serial port 3 = UDP port 10003

Using secondary address means we read the first M-Bus record to choose the right device.

Each M-Bus device always has 1 record containing its own serial number

Water meters 121..160 need to be configured as modules 1..40 on port 9 on the FX:

Th 17.08.2017 20:49:59 I/O modules Update

Port.Module 07.001	Type MBUS	Version V0.00	Communication Ok	Unknown	Errors 6.07%
Port.Module 09.034	Type MBUS	Version V0.00	Communication Ok	Unknown	Errors 0.00%

Port number **9** Module address **34** Type MBUS

Poll time (ms)	0	TCP/UDP module	☒
Message count	209314	IP address	10.10.11.114
Errors now	0	IP port	10003
Errors max	1		
Errors total	7	Debug	☐
Receive timeout (ms)	500		
M-BUS device number	12028745	Use secondary address	☒
M-BUS address	0	Send Reset command	☐

Clear error counters Find modules

Virtual port 9 on the FX

Each water meter gets a virtual "Module address" that is used in the IEC code where the value is read from the device (GetMBUSValueF)

Module is connected via Ethernet to multiLINK IP

multiLINK serial port 3 = UDP port 10003

Using secondary address means we read the first M-Bus record to choose the right device.

Each M-Bus device always has 1 record containing its own serial number

M-bus to Modbus TCP

To use this feature, make sure you have a multiLINK that has the web feature enabled. You can verify this by either checking the multiLINK's serial number; for a web server enabled multiLINK, this starts with "MW" (not "MC"), or by logging in through telnet / terminal where the same serial number is listed, and also the "Base Module Type", which for the web server enabled version is "multiWEB" (not "multiCOM").

Set up your multiLINK's port as "M-Bus master".

The physical serial ports are bridged as follows to Modbus TCP ports: Serial port 2 → TCP port 503, Serial port 3 → TCP port 504.

The external Modbus master will read the values that are being passed into Modbus holding registers directly from the multiLINK. Therefore, also no external port configuration needs to be done (and because the multiLINK is the M-Bus master itself). Any IP or port values entered will be ignored.

At this point, all values to be read need to be configured with the help of the multiLINK / multiWEB converter program. Once this is done, all values read on the M-Bus side, will be offered as Modbus registers on the TCP side. The multiLINK can be polled on Modbus address 250. The values it reads from the connected M-Bus devices will be set to a plane of Modbus holding registers. The way to calculate the register numbers appropriate for each connected device is as follows:

Take the address of the M-Bus device (either primary or "virtual" when secondary addressing is used, see later in this document for more details), and apply a "Shift 9 bits to the left" operation on the binary value of the address.

E.g.: Address 4 is "0100" in binary, so after shifting 9 bits to the left: "1000 0000 0000" or 2048 in decimal.

Add the register number (or index or record number) which contains the measurement we are interested in, (in most cases "1"), in our example adding up to 2049.

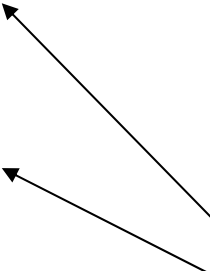
This will be the first Modbus register of 4 to read the measurement value of the M-Bus device. We need to use 4 Modbus registers, because M-Bus registers (or indexes or records) are stored as 64 bit values. This means that the first Modbus register contains the most significant bits, and the fourth contains the least significant bits of the 64 bit value.

For water meters, mostly, a 32 bit value is sufficient, so we will only need to read the last two registers.

A few more examples:

Address	Binary	Shift left by 9	Decimal	M-bus record #	Modbus register # range
45	0010 1101	0101 1010 0000 0000	23040	1	23041 - 23044
23	0001 0111	0010 1110 0000 0000	11776	1	11777 - 11780
17	0001 0001	0010 0010 0000 0000	8704	5	8709 - 8712
4	0000 0100	0000 1000 0000 0000	2048	5	2053 - 2056

multiLINK		FX-controller	
Device setting :	Local IP address	Device setting :	Local IP address
Hardware :	Serial 2 or 3	Hardware :	Ethernet (RJ45)
Settings name :	Serial 2 or 3	Settings name :	Port 6..10
TCP (local port) :	503 or 504	TCP (local port) :	N/A
Parameter :	Remote IP: N/A	Programming :	TCP/UDP module IP
Parameter :	Remote Port: N/A	Programming :	TCP/UDP module Port



Using the embedded webserver

Creating and uploading HTML pages

The multiLINK can be equipped with an on-board web server.

It can be used to display slave-bus values when either “Modbus master” or “M-Bus master” is selected as the operation mode for one or more serial ports.

It is important to realise that, while the multiLINK can be used as a stand-alone bus master with web server to show the measured values, this is NOT the intended purpose of the multiLINK. The web server is merely an extension, something that might be useful sometimes.

As long as the number of bus registers is not too high, this multiLINK will work fine to show bus registers on locally stored html pages, but we do not guarantee its continuous functionality when used with many values.

That is also why we didn't add any security to either the settings pages, or the web server pages you can upload to the multiLINK yourself; in its intended usage, the multiLINK is used as a protocol converter or as a physical gateway.

That being said, for smaller applications it can be very useful to be able to consult data directly from the multiLINK's web server pages. A possible example is a setup in a residential apartment building, where each apartment's water- and electricity meters are connected through M-Bus to the multiLINK, where residents can consult their (and their neighbour's!) consumption counters.

The BMS subsequently reads that data through Modbus TCP from the multiLINK, and saves historical data etc.

Using the HTML editor

To make html pages for the multiLINK, you need the Fidelix Graphics Editor (HTML Editor), which you can download from the partner webpage, or the Fidelix support website.

Use pointIDs with a 100% free syntax. So, unlike for the multiDISPLAY, the FX-controllers or webVision, the pointIDs can be anything you want, provided you don't use spaces or non-alphanumeric characters. The multiLINK converter detects pointIDs, and these can be linked to slave-bus registers. This also means that on different pages, the same PointID will always point to the same slave bus register. Time and date can be added in the same way they are displayed on the FX-controller, by using pointIDs TIME, TIME2 (without seconds) and DATE. Any element that does not have a pointID will become a part of the background image during conversion.

To make links between different pages, use PointID “LINK”, and then add the html page to link to, in the normal way in the “Link” field. Omitting the PointID will render the links unusable. The internal pages of the multiLINK (listed under [Other configuration](#)) can also be linked from the user defined pages.

The converter will only look inside the first 1024 x 768 pixel range of the pages you create (starting from the top left corner). This means that all your elements must be inside this area. An easy way to make sure, is to put symbols without pointID at following coordinates:

X	0	Y	768	Width	1024	Height	10	X	1024	Y	0	Width	10	Height	768
---	---	---	-----	-------	------	--------	----	---	------	---	---	-------	----	--------	-----

This will create a “working area” box that is easy on the eyes.

Using the converter

The HTML pages you make in the Fidelix Graphics Editor (HTML Editor), need to be converted to a format understandable by and suitable for the multiLINK. We use the multiLINK converter (multiWEB converter) for this, which you can download from our partner or support pages.

The converter will take screenshots of the pages you load into it, so make sure that

- 1) during the conversion process, you do not open any other windows on top of the converter
- 2) The converter window has at least 1024 x 768 pixels to work with, counting from its top left corner.

The conversion is initiated by opening the start page of your multiLINK project, which will be the first page displayed when a connection is formed to the multiLINK with a browser. The conversion program will automatically go through all the linked pages and convert them to the understandable format for the multiLINK.

At the same time, the conversion program will gather information on all the points it finds on the pages. Once conversion is completed, the converter pops up a list of all the detected points in the project on the screen.

When the list is compiled, there is an attempt to configure a unit and the amount of decimals for each point by using the defaults determined in the "DefaultUnits.txt" file, which is located in the converter folder. The syntax of the file (which you can freely edit), is as follows: `Suffix_of_the_pointname`, unit (can be empty), # of decimals, divider, and this as many times (on as many lines) as needed.

In this popup, you can edit the Point Type, Data Format and Point Unit to match the devices you have connected, where this wasn't yet correctly filled out by the converter.

In the "Decimals" column, you can enter the number of decimals like "2" or "3", or enter "2.2", meaning that a value of 2.5 will be shown as "02.50". With the same logic, "3.4" will display a value of 25.25 as "025.2500".

In the "Math Operation" column, all standard operators; / (divide), % (modulo), x or * (multiply), # (integer division), + (add), - (minus) can be used to show correct values on the webpages. The register values themselves stay unaltered.

The "Sample Interval" column is used to define the sampling time in seconds when saving points to the "brief history" of the multiLINK (see later in this document).

The "Extra Properties" column is used when reading M-Bus meters to enter the device address (the 8 decimals number).

Once all configurations have been done, click the "Generate Communication Table" button to generate the list of communication sockets the multiLINK will communicate over. The "registers per Request" value in the bottom of the screen can be altered if your connected device doesn't allow 100 registers to be polled at a time. The value is used to determine the size of the generated communication sockets.

At this point it is a good habit to save the point parameters as an xml file (You may have to click the "Edit Points" button to bring the popup back up), as each conversion of the pages will reset all changes that you made to the point parameters. During development it is thus very practical to have a backup of your settings, and load that Point lists after you have made changes to your html pages and ran them through the converter to get the newly added points.

Note that loading a saved xml-point list will overwrite all points that are currently in the displayed point list.

Use the same login credentials (ADMIN / multiWEB) as for administrator telnet / terminal access to load the pages, point configuration and communication sockets to the multiLINK.

You will see that an FTP connection is created and the files are loaded to the multiLINK.

Don't forget to restart your multiLINK (just cut the power supply for a second) to apply all changes appropriately.

Now you can start debugging your communications by checking the communication counters on the CommList page. The first thing you might want to change if you encounter any communication errors, is the timeout period, knowing this is global for all devices connected to that serial port.

Sending alarm messages

Prerequisites

Sending SMS messages requires connecting a MC55iT modem to the multiLINK's serial port 4 (RJ9). A "FX-SP-D9F" cable should be used for the connection, so that the modem is also powered by the multiLINK. The PIN code request of the SIM card should be disabled.

For sending email alarms, only a working network connection to the outgoing mail server is required. Though authentication can be used, you cannot use encryption, and only port 25 for sending emails from the multiLINK.

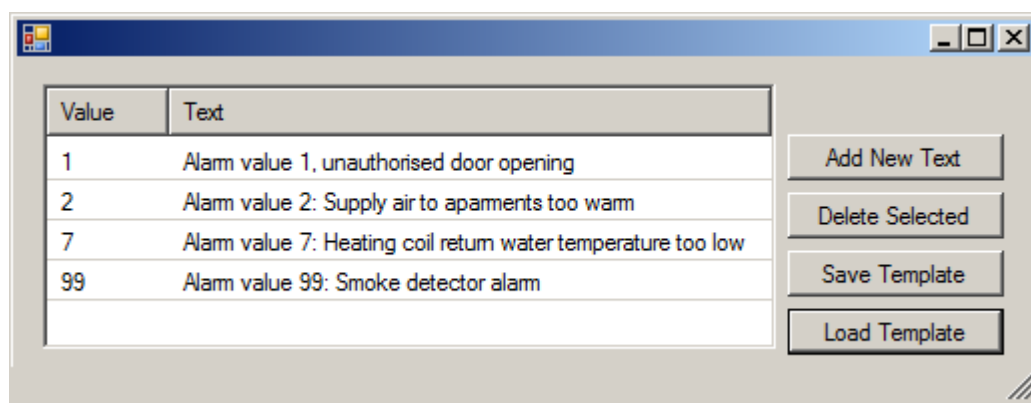
Setting up alarms during programming

The multiLINK can send alert messages via email and SMS.

Sending the alerts is based on using a Modbus "alarm register" on one of the connected devices, which the multiLINK reads to detect when it should send a pre-programmed message. This can be a register that is also otherwise already displayed on the webpages of the multiLINK, or a "hidden" register.

To add the alarm point, simply click "Add Empty Point" at the bottom of the "Edit Points" window. By default, this point will be set up as an alarm point (Point Type 16). Select the type of register (coil, holding or input), and the type of alarm you want to send (email or SMS), and enter the correct parameters for Serial Port, Module Address and Register Number. If you want to send out SMS and email messages for the same alarm point, create two separate new alarm points with the same settings.

Next, click the "Edit Alarm Templates" button, also at the bottom of the "Edit Points" window. By default, an empty alarm template will be opened. You can start editing it freely to match your project.



The "Value" column is the value read from the "alarm register". This means that from one device you can read many different alarms, and each value will represent a different problem in the connected system. The multiLINK will only send alarms for values in this list, and just once, when the new value is detected. In the example above, that means that when the register value changes from 0 to 4, nothing will happen. When it then changes to 2, the alarm for which this template has been selected will be sent once.

An individual template file can be created for each of the alarm points, or the same file can be used for several alarms.

Now, you can click the “Extra Properties” column in the “Edit Points” window to select the template you just made.

The screenshot shows the 'Form2' window with a table of points. The 'Extra Properties' dialog box is open, showing the 'Select Recipients' section with four recipients selected (Recipient 1, 2, 3, and 4). The 'Select Alarm Template' section shows 'alarms.dat' selected. The 'OK' button is highlighted.

HTML Page Name	Point Name	Point Type	Data Format	S_	M_	Point/_	Extra Properties
Manually Added Poi...	No Name 35758	16 ALARM	01 ALARM via email (holding reg)	1	1	1	
Manually Added Poi...	No Name 64448	16 ALARM	02 ALARM via SMS (holding reg)	2	10	1002	
Manually Added Poi...	No Name 933	16 ALARM	01 ALARM via email (holding reg)	2	10	1002	
water_meters.htm	DATE	23 Time / D_	03 Date (DATE)	0	0	0	
water_meters.htm	METER01	18 MBUS m_	01 Single Reading	3	1	5	
water_meters.htm	METER02	18 MBUS m_	01 Single Reading	3	2	5	
water_meters.htm	METER03	18 MBUS m_	01 Single Reading	3	3	5	
water_meters.htm	METER04	18 MBUS m_	01 Single Reading	3	4	1	
water_meters.htm	MULTI24_01_	19 Holding _	01 unsigned single register (uint1_	1	8	1	
water_meters.htm	MULTI24_02_	19 Holding _	01 unsigned single register (uint1_	1	8	2	
water_meters.htm	MULTI24_03_	19 Holding _	01 unsigned single register (uint1_	1	8	3	
water_meters.htm	MULTI24_04_	19 Holding _	01 unsigned single register (uint1_	1	8	4	
water_meters.htm	MULTI24_05_	19 Holding _	01 unsigned single register (uint1_	1	8	5	
water_meters.htm	MULTI24_06_	19 Holding _	01 unsigned single register (uint1_	1	8	6	

Below the table, there are fields for Point Type, Data Format, Serial Port, Module Address, Point Number, Point Unit, and Desi. There are also buttons for Save, Add Empty Point, Generate Communication Table, Load, Delete Selected Points, and Edit Alarm Templates. A 'Registers per Request' field is set to 100, and a 'point count is 38' label is present.

Here, you also select recipients to whom the alarm will be sent. You can select 1, 2, 3 or the maximum of 4 recipients for each alarm. The recipients are only identified by numbers 1 through to 4 during the configuration of the system on your computer. The actual email addresses or phone numbers to send the alarms to, will be configured on the multiLINK itself. Keep careful track of which alarms you forward to which recipients here, so you can set them up correctly on the multiLINK (= making sure “Recipient 1” is linked to the right email address or phone number).

Configuring recipients on the multiLINK

Go to “Email Settings” (Email.htm) to configure SMTP server parameter and recipient addresses and phone numbers.

Email settings

For sending emails, all that is needed is to enter the server address of the SMTP server and click “Save Settings”. Though authentication can be used, you cannot use encryption, and only port 25 for sending emails from the multiLINK.

Make sure that, even though you don’t use authentication, at least the “Domain” field has a (syntax wise) “valid looking” entry, otherwise the SMTP server will bounce the message back for not following the correct syntax.

Then, enter the recipients’ addresses one by one, and click the “Save and send a test email” button one by one.

You will see that when you navigate away from the Email.htm page and come back, the SMTP server IP address is now filled.

If there is still 0.0.0.0 for the SMTP server IP address, there is probably a problem with the DNS server configuration or the network access.

multiLINK - Email configuration

Sender's Email address	multiLINK@fidelix.net
SMTP server address	smtp.nblnetworks.fi
SMTP server IP address	83.145.220.69

SMTP server requires authentication		☐
Domain	my_domain.com	
User name		
User password		

Save settings

SMS settings

If your modem is connected correctly and “up and running”, you will see a value mentioned for the strength of the signal from 1 (very poor reception) to 31 (perfect connection).

Enter phone numbers in the format shown on the right and click “Save and send a test SMS” to make sure everything is set up correctly.

Enter your phone number here and send a test message. Test SMS messages sent: 1
Current signal strength: 29 (1..31, 0=Error, 99=No network)

Recipient 1 Phone number

Recipient 2 Phone number

Recipient 3 Phone number

Recipient 4 Phone number

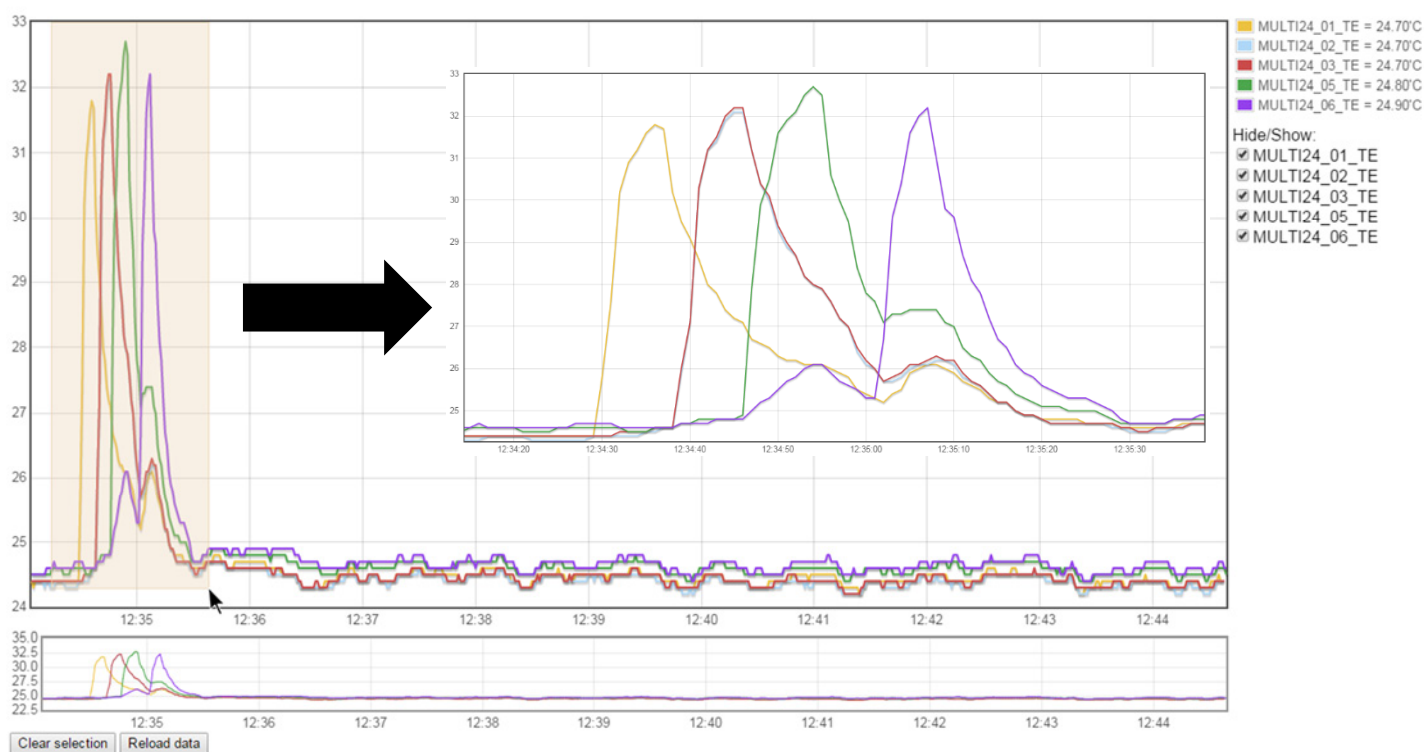
History

The history feature, though rudimentary, can be used to follow up on a system during installation, to fine-tune PID settings or other parameters that might affect the system.

Setting up is very easy; simply enter a value (in seconds) in the “Sample Interval” column. The data is recorded on the µSD card, so make sure you have one installed into your multiLINK when activating historic logging. The maximum amount of values stored per point, is +/- 7000, depending among others on the number of points.

The data can be consulted on the /trend.htm page. The view offers zooming by “click and drag” around the area you want to see enlarged.

multiLINK Trend

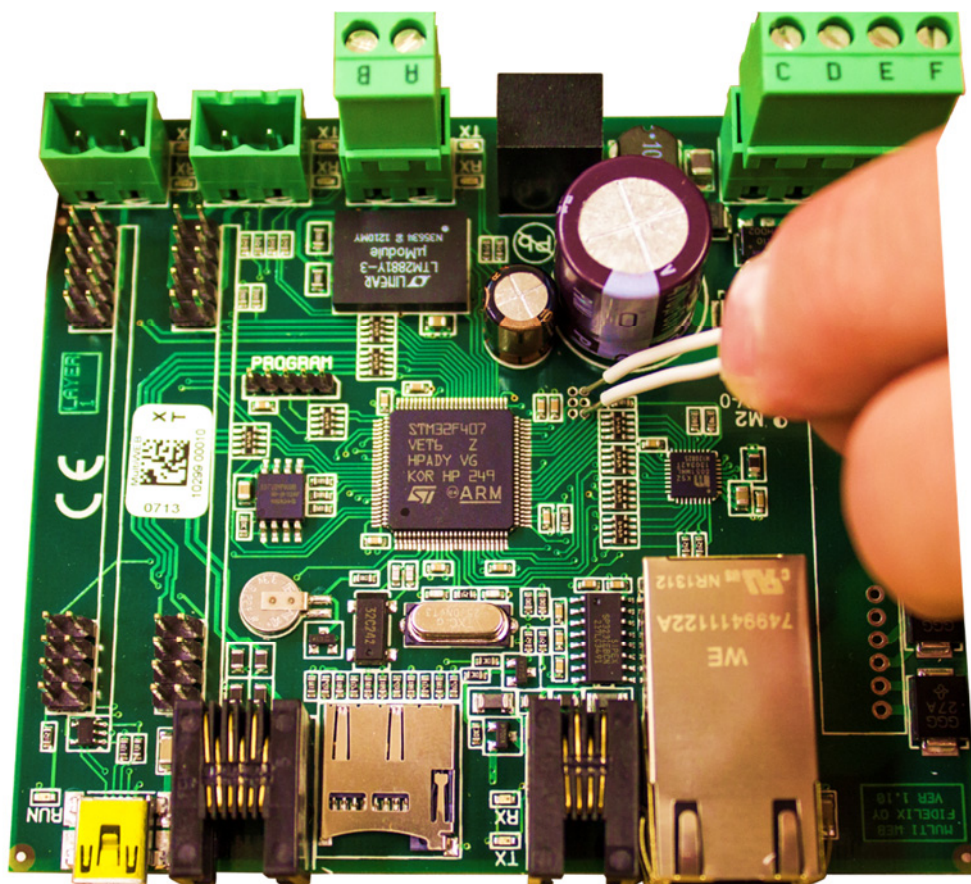


Clicking anywhere in the table, will display all values on that point in time on the right side.

Loading data might be a little slow sometimes, as priority is always given to the serial communication over any webserver request.

Restoring factory settings

If the IP address of the device is not known and the DEBUG serial port of the device is not available, a connection can still be established by restoring the factory settings of the device. After the factory settings have been restored, a connection to the default address (10.100.1.97) can be established. The factory settings are restored by connecting the two points on the circuit board, as shown on the following photo:



PLEASE NOTE! When the factory settings are restored, any HTML pages, serial communication parameters and any other settings created by the user will be deleted.