eShark
HD-PLC LON
EVK
User Manual

This manual refers to the eShark HD-PLC LON EVK

Gesytec GmbH
Pascalstr. 6
52076 Aachen, Germany

Tel. + (49) 24 08 / 9 44-0
Fax + (49) 24 08 / 94 4-100
email: info@gesytec.de
www.gesytec.com

Doc. ID: HD-PLC/UserDoc/eShark_HD-PLC_LON_manual_en_V0201.docx,
Version v2.1, 09/23/2020

Gesytec

This Manual …

...describes set-up of and working with the eShark HD-PLC LON EVK. This manual does not go into details related to theoretical backgrounds of the related technologies. Please inquire the corresponding literature for further information. More information about HD-PLC can be found on the website www.hd-plc.org. Backgrounds about LONWORKS® can be found on the website www.eche-lon.com.

HD-PLC is a well-established standard to distribute Ethernet and RS485 using power line cabling or others like twisted pair, coax and more. It is based on IEEE 1901 standard, extended by ITU G.9905 for multi hopping.

ISO/IEC 14908-8 defines the LON protocol using HD-PLC and makes this type of LON communication open and vendor independent.

Because there is need to download and install a couple of software to start with the EVK, please reserve at least 4 hours to run through this very first step.

MegaChips is a provider for HD-PLC compliant silicon. Gesytec provides since a long time a full LON compliant software stack. Bothe technologies combined make a new hardware and software solution for LON, independent of other technology providers.

Contents

1 About HD-PLC LON EVK 4
1.1 Content of the EVK 4
1.2 Hardware 4
1.3 Software provided 7
1.4 3rd party software needed 8
1.5 PC requirements 8
1.6 Important Notes 9

2 Basics 10
2.1 HD-PLC 10
2.2 LON 11
2.3 HD-PLC LON 11
2.4 IP addressing 11

3 Install Software and Settings 13
3.1 IP address for the PC 13
3.2 STM32CubeIDE 13
3.3 7ST-LINK/V2 13
3.4 Easylon IP Interface+ 13
3.5 Easylon HD-PLC gateway software 17
3.6 Easylon OPC Server M 17
3.7 EPOS XIF Editor code generator 18
3.8 Easylon Analyzer 18
3.9 Gesyline Manager 18
3.10 Echelon IzoT CT 18
3.10.1 Microsoft Visio 2016 18
3.10.2 Echelon IzoT CT 18

4 Overview 19

5 Setup the hardware 20
5.1 First steps 20
5.1.1 Easylon HD-PLC Bridge 20
5.1.2 eShark Box 20
5.1.3 Gesyline Manager 21
5.1.4 HD-PLC LON Master Setting 23

5.1.5 EasyCheck....24
5.2 Setup for development....26
5.2.1 Breakout board....26
5.2.2 Debugger....27
5.2.3 USB terminal connection....27
5.2.4 Check with preloaded application....28

6 Developing a LON application....29
6.1 Software Structure....30
6.2 Structure of the LON application on the STM32F091....31
6.2.1 Header files....31
6.2.2 Source Code....31
6.3 Using the XIF editor....32
6.3.1 Define network interface....32
6.3.2 Generate ANSI-C application frame work....34
6.4 Drop in custom code....35
6.4.1 Remarks....36
6.5 Download....36
6.6 Commission in IzoT CT....36

7 EPOS API for STM32....45
7.1 Diagnostics....45

8 Hardware details....46
8.1 Peripherals....46
8.2 STM32CubeIDE....47
8.3 STM32F091 pin out....48

1 About HD-PLC LON EVK

1.1 Content of the EVK

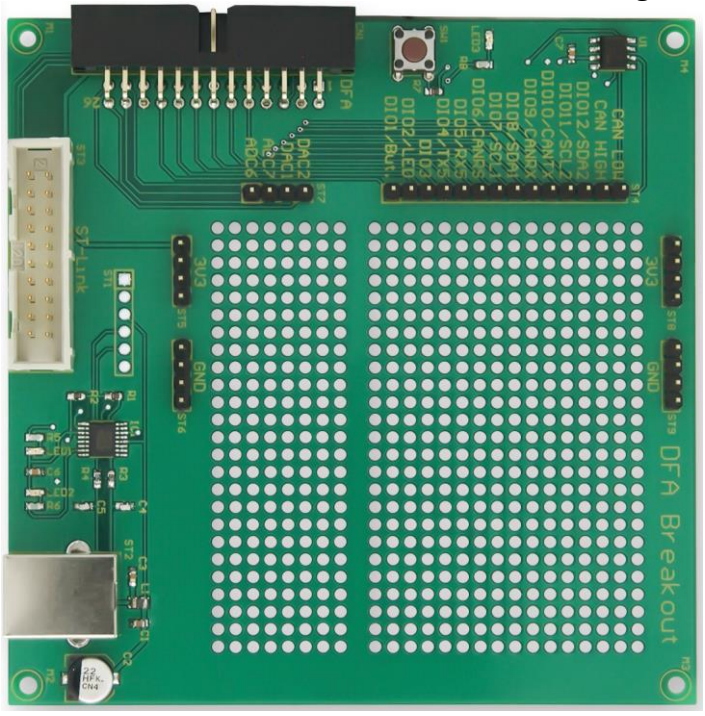
The LON HD-PLC evaluation kit comes with all the necessary hard- and software to make your first steps in developing LON applications for HD-PLC.

1.2 Hardware

To allow development of HD-PLC LON applications, the EVK consists of this hardware:

- **2 eShark boxes**
  These boxes are connected to main power. They contain the HD-PLC CPU, running the LON protocol stack, and the STM32F091CB CPU, running the end users application.
  Both boxes are preloaded with an application, driving all the peripherals being available on the breakout boards.
  Additionally power cables for EU and US are provided.

- **2 breakout boards**
  The breakout board allows testing own hardware. It is connected using a ribbon cable with the eShark box. A breakout board additionally adapts to the ST-LINK/V2 debugger. Furthermore, the diagnostics UART of the eShark is available with a FTDI UART/USB bridge.



- **1 ST-Link/V2 USB debugger module**
  This is the debugger to be used for the end user application, running on the STM32F01CB CPU.

- **Easylon HD-PLC Bridge device**
  To implement a simple solution for the HD-PLC master device an off the shelf HD-PLC Ethernet converter is used..

1.3 Software provided

Together with the eShark HD-PLC LON EVK you receive credentials to access the FTP area for the EVK at https://ftp.gesytec.de/pub/Gesyline/eShark

Here you will get:

- **Source code of the demo application**
  The boxes come with a preloaded application, operating all I/Os of the breakout boards.

- **Easylon IP Interface**
  All Easylon drivers for a single hardware interface implement multiple Windows driver instances in MIP and VNI mode for a single interface. The HD-PLC LON EVK makes use of the Easylon IP driver to give multiple PC applications access to the HD-PL CLON network simultaneously. The kit includes a USB dongle to have a unique neuron IDs for the PC.

- **Easylon HD-PLC gateway software for Windows**
  Because the eShark box allows connecting the HD-PLC using Ethernet, the simplest LON connection is making a software bridge from the HD-PLC LON protocol to a standard Windows LON driver.

- **Easylon OPC Server**
  For easy monitoring a license of the Easylon OPC Server M is provided. This software comes “as is” and is not intended to be used in productive environments. Furthermore it is necessary to run the XIF Editor code generator.

- **Gesytec XIF Editor Code generator**
  Using the XIF Editor code generator allows defining a XIF and creating an ANSI-C application framework from the defined LON network template.

- **Easylon Analyzer setup**
  A license for the Easylon analyzer is provided with the EVK. This software comes “as is”.

- **Gesyline Manager**
  Using this software allows updating the firmware of the BlueChip and other diagnostics of the HD-PCL network.

1.4 3rd party software needed

For full working with the EVK, some 3rd party software is necessary:

- **STM32CubeIDE**
  The STM32CubeIDE is the preferred solution to define the peripheral pin definition of the STM32F091CB. It is also designated ANSI-C IDE for the STM32F091CB application CPU of the eShark box. You can get it here:
  https://www.st.com/en/development-tools/stm32cubeide.html
  Please download the standalone version

- **Echelon IzoT Commissioning Tool (CT)**
  Previously known as LonMaker the IzoT CT is necessary to create connections between two devices. Izot CT needs to be purchased from Echelon directly or resellers. A limited demo version can be downloaded from the Echelon website but needs to have a Microsoft Visio 2016 installed additionally.

- **UART Terminal application**
  To pick up the diagnostic output of the STM32F091CB available on UART4, the breakout board provides a FTDI UART/USB bridge. A Terminal application is required e.g. TeraTerm.
  https://osdn.net/projects/ttssh2/releases/

1.5 PC requirements

All software is running on Windows operating system from Windows 7 to Windows 10. The PC should have

- 64 GByte free disk space
- 8 Gbyte RAM
- A second Ethernet interface is recommended

1.6 Important Notes

The designated use of the eShark HD-PLC LON EVK is evaluating the HD-PLC LON technology and preparing own hardware solutions. It must be used in safe laboratory environments only.

Due to the powerline technology, the eShark HD-PLC LON is connected to main power. It must be used only by users, having an appropriate education. Main power can cause severe harm.
The eShark PLC LON EVK is not designed to run in end users applications. It is for development and lab usage only. Especially take care on the high voltage connection. Don't open the enclosure, when it is under operation.

2 Basics

2.1 HD-PLC

This powerline technology, standardized by IEEE 1901 and ITU G.9905 is a highly efficient Ethernet transmission on a couple of cable types, with or without power. It is using a spread spectrum transmission in the frequency range of 2-28 MHz. The following table lists key performance features of HD-PLC.

Standard	IEEE1901 HD-PLC
Frequency band	2 -28 MHz
Modulation	Wavelet OFDM
Transmission PHY Rate	240Mbps
Access Method	CSMA/CA, Dynamic Virtual Token Passing
Security	AES 128-bit Encryption
Error Correction	Reed-Solomon/LDPC-CC
Coexistence	ISP (Inter-System Protocol)
Routing	CMSR (ITU-T G.9905)
IP Support	IPv4/IPv6 (IETF)

There are some special terms with HD-PLC which will be used in this documentation.

- **Master**
  This is a dedicated device, managing the routes between the individual HD-PLC devices. Especially in case of a Multi-Hop network, the master device permanently checks the connections and calculates the most efficient routes between the devices. Usually an HD-PLC device offers a switch to enable the Master function.
  The HD-PLC EVK comes with a dedicated hardware device, acting as the HD-PLC master.

- **Terminal**
  A terminal device is a usual device in an HD-PLC network. The eShark boxes act as terminal device.

- **Multi-Hop**
  In case of large networks, a terminal device can act as a hop. A hop works like a repeater and extends the HD-PLC networks physical dimensions. In this case the HD-PLC network is a Multi-Hop network. Multi-hopping is defined with ITU-T G.9905.

2.2 LON

LON is a control network defined by ISO/IEC 14908. It can use different transport layers by implementing the identic upper layers of addressing, transaction processing and representation of data. This is a unique feature for a control network. Because HD-PLC is implementing an Ethernet-like transport, LON can be adapted to use HD-PLC easily.

- **Neuron-ID**
  This is the physical address of a LON node. It is implemented by a 6 byte array, comparable with the MAC address in Ethernet.
  The neuron ID of a eShark box is derived from its Ethernet MAC address, by replacing the leading 0x00 byte with a 0xFE, which defines a virtual Neuron-ID. The LON stack for the BlueChip dedicated for production use comes with an individual and unique neuron ID.

- **MIP**
  MIP means Microprocessor Interface Program and was a firmware for a LON neuron chip to build a LON PC interface. Nowadays it stands more for a LON interface on LON layer 5. So a MIP interface already addressing and transaction handling.

- **VNI**
  VNI stands for Virtual Network Interface. A VNI allows accessing a LON network on layer, so all LON messages can be received and sent. It is the most flexible solution and offering best performance. Echelon's IzoT is using a VNI interface to access the network.

2.3 HD-PLC LON

With ANSI/CTA 709.8 HD-PLC LON is already standardized in the U.S and Europe will follow with EN 14908-8.

- **HD-PLC LON master functionality**
  Finally this is implemented in the EPOS firmware. The device which is the HD-PLC Master provides the HD-PLC LON master functionality automatically. In HD-PLC LON terminal devices will send the HD-PLC LON message using unicast addressing to the master, which will multicast the message back into the network.

2.4 IP addressing

To simplify the setup of the EVK, it uses fixed IP addresses in the 192.168.0.x network, default IP address is 192.168.0.249. The PC used for the work with the EVK needs to have 192.168.0.1 as IP address.

The IP address of the eShark Boxes can be set, using the Gesyline Manager.

3 Install Software and Settings

3.1 IP address for the PC

The EVK comes with static addresses in the 192.168.0.x subnet, default address is 192.168.0.249. The Ethernet interface of the PC to be connected with the EVK network needs to be setup to **192.168.0.1**

It is recommended to use a dedicated Ethernet interface for the communication with the HD-PLC system to have a clean environment.

3.2 STM32CubeIDE

After downloading the software just execute the setup according to the setup guidelines.

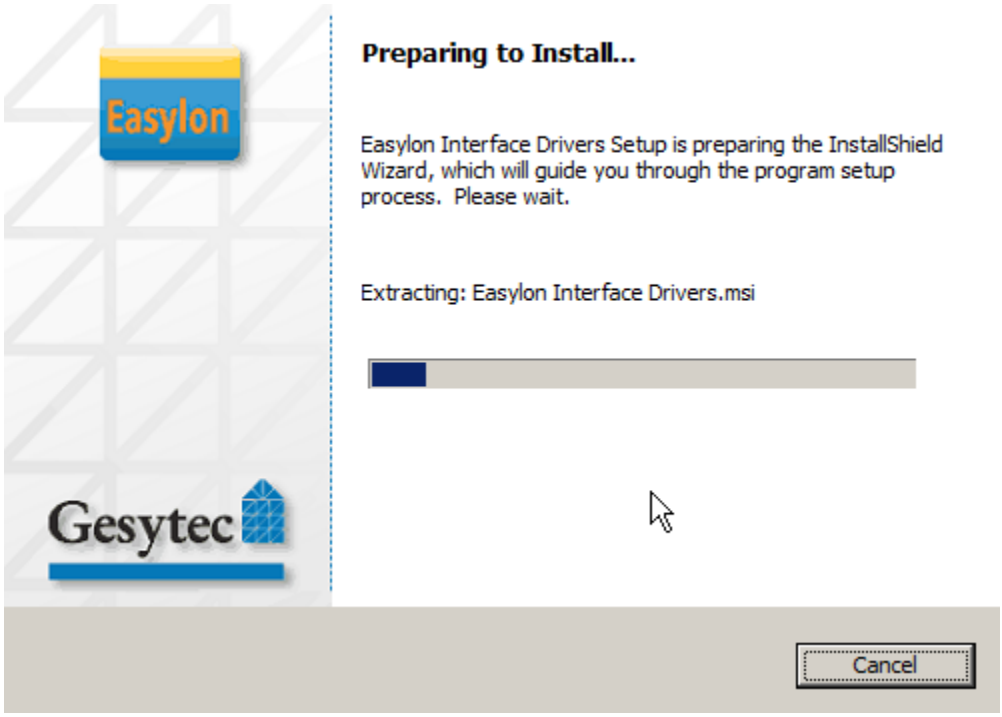
Please check after installing, that the SMT32F0 package is installed, by calling Help->Manage Embedded Software Packages menu. Usually an STM32F0 is already installed and can be maybe already updated.

3.3 7ST-LINK/V2

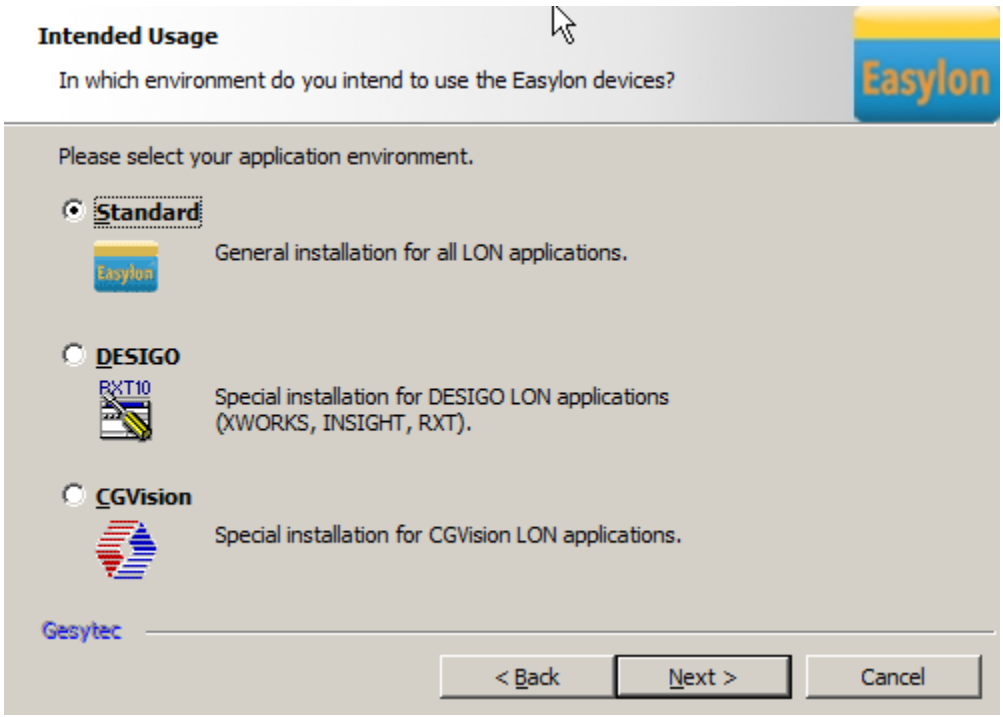
Just connect the ST-LINK/V2 debugger to your PC. The driver software will install automatically.

3.4 Easylon IP Interface+

Please run EasylonDriver-7.22.2177.0.exe from the downloaded FTP archive in **HDPLC\EasylonDriver**



Run the “Standard” setup and select “Complete” in the following dialog.



The setup for the Easylon Interface drivers installs a management tool named “Easylon Interface Management Center”, short EIMC to control the drivers.

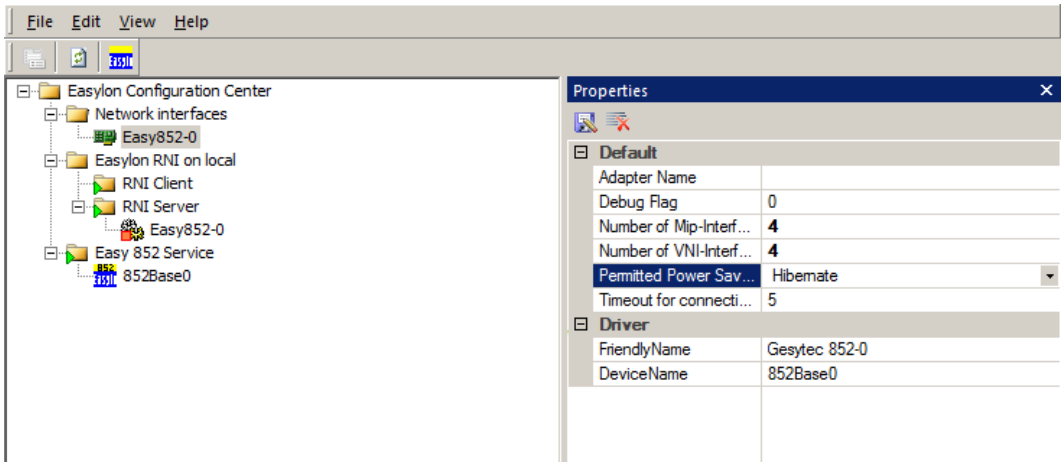
Now connect the USB dongle to the PC.

Please start the EIMC and check if the configuration tree looks like the next picture – the Easy852-0 network interface present, the Easy 852 Service with one

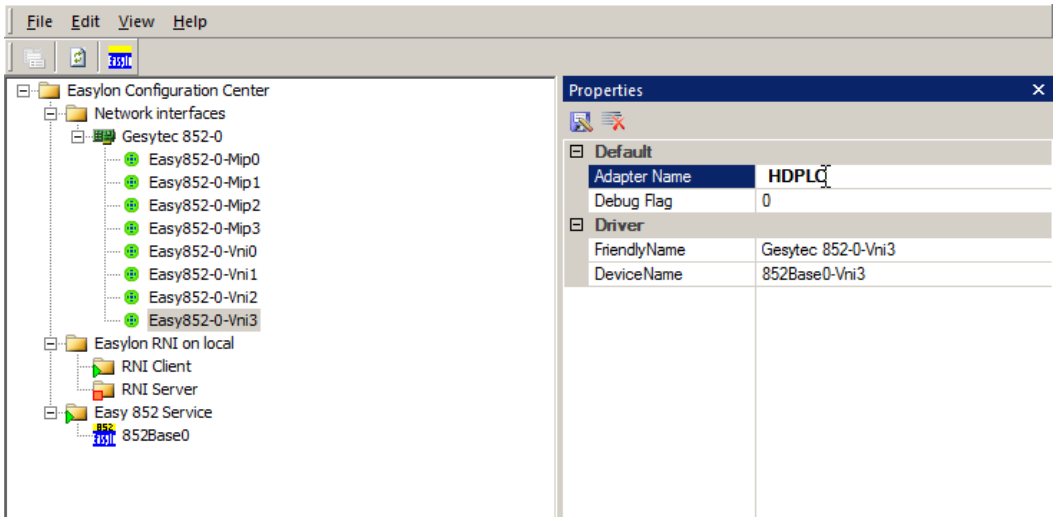
852 server instance (852Base0) running. If not do right-click on the ‘Easy 852 Service’ and select ‘New 852 server’ in the context menu. New 852 server instance will be created.

Select the Easy852-0 interface and change the number of VNI and MIP interfaces to 4.

Please note: All the time, if there are entries in bold in the properties pane, you need to press the disk symbol, to make the changes become active.



In a next step please change the names of the instances below Easy852-0 to the shown names.



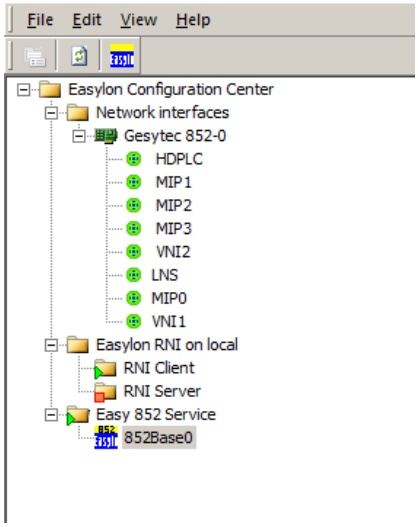
Take care: sometimes a blank is added as first character of the Adapter Name, please delete that blank.

Rename like this:

xxx-VNI0	HDPLC
xxx-VNI1	LNS
xxx-VNI2	VNI1

xxx-VNI3	VNI2
xxx-MIP0	MIP0
xxx-MIP1	MIP1
xxx-MIP2	MIP2
xxx-MIP3	MIP3

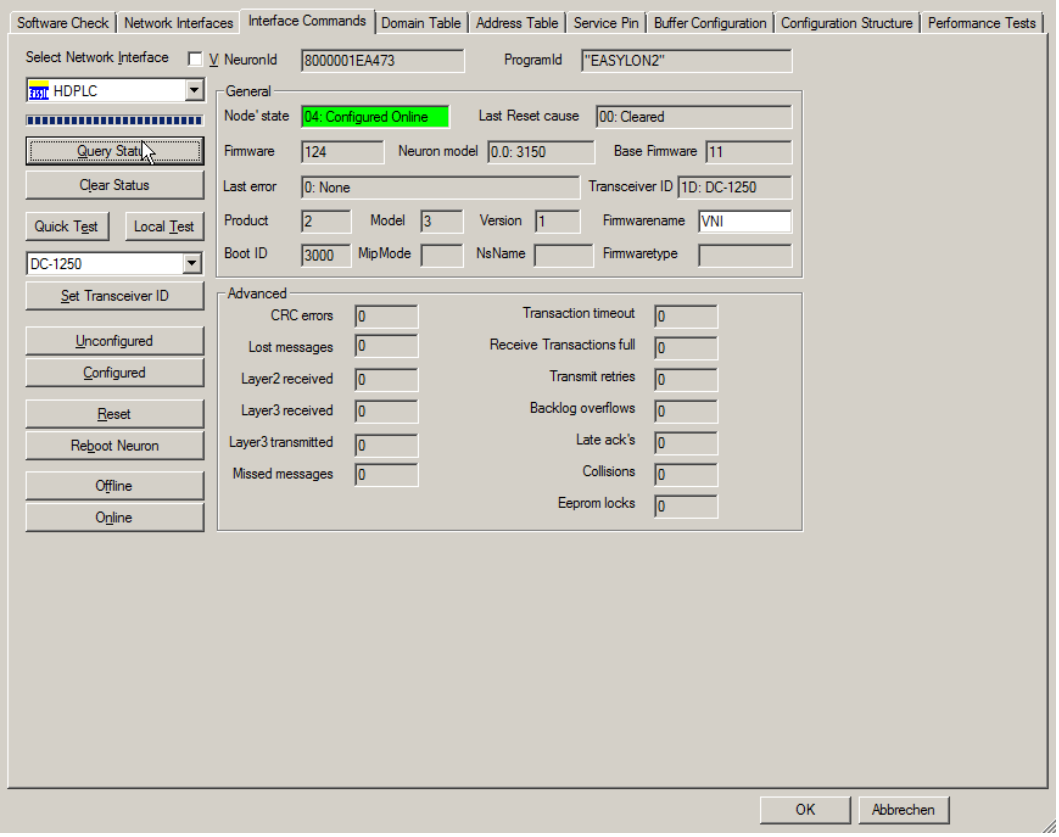
Using this logical name prevents us from getting lost in the different applications with the LON driver names.



HDPLC will be used for the Easylon HD-PLC gateway software, LNS will be used for Echelon IzoT CT. MIP0-MIP3 will be used for any other LON software, like the Easylon Analyzer, OPC Server, EasyCheck or 3rd party application.

Each hook can only be used by a single application, e.g. the Easylon Analyzer and the Easylon OPC Server can't share an interface at the same time. Please track which MIP hook is used for which application. You may rename MIP0-MIP3 according to your favor to realize an easy differentiation of use.

Finally run EasyCheck from the start menu in the Easylon category. Select the Interface Commands tab and HDPLC from the Select Network Interface combo box. Press the Query Status button and you should see a green field with the node status:



3.5 Easylon HD-PLC gateway software

To connect with the HD-PLC system on the PC with standard LON drivers, Gesytec implemented a gateway software which is mapping the HD-PLC LON messages into standard Windows LON drivers. This is the reason, why an installation of the Easylon IP drivers is necessary. The HD-PLC gateway application is connected to one of the VNI hooks of the Easylon driver and the Easylon driver makes the LON messages available on all other driver hooks. That's the reason why one of the hooks was renamed to HDPLC to make the setup easy.

Just start the hdplc.exe, found in **HDPLC\EasylonDriver** with this command line:

```
hdplc -d HDPLC -L 192.168.0.1
```

Most likely Windows will ask now to enable the application working through the firewall.

3.6 Easylon OPC Server M

Since a long time, Gesytec is having an OPC (OLE for Process Control) Server for LON. Due to the vanishing role of OPC DA in the industry, the software is no longer actively marketed, but still available. It gives easy access to all values

of a LON network. The user interface can be used to have an overview on the network, if LonMaker is not available. Please refer to the manual of the Easylon OPC server.

The setup is located in **HDPLC\EasylonOPC**.

If used, the Easylon OPC Server M needs to be connected to one of the MIP0-MIP3 interfaces. You find a license for the Easylon OPC Server M on the license sheet, provided with the EVK.

The setup of the Easylon OPC Server M in mandatory to install the environment for the Easylon XIF Editor.

3.7 EPOS XIF Editor code generator

This is an update of the XIF editor, coming with the previous installed Easylon OPC Server. The software is located in **HDPLC\XIF2Code**. The exe-file can be started without any installation. Please copy the content of the directory, together with the .c.xtf and .h.xtf files to a directory of your favour.

3.8 Easylon Analyzer

The Easylon Analyzer needs to be connected to one of the MIP0-MIP3 interfaces. You find a license for the Easylon Analyzer on the license sheet, provided with the EVK. The setup is located **in HDPLC\EasylonAnalyzer**.

3.9 Gesyline Manager

The Gesyline Manager Setup is in the FTP area for the EVK at https://ftp.gesytec.de/pub/Gesyline/eShark.

3.10 Echelon IzoT CT

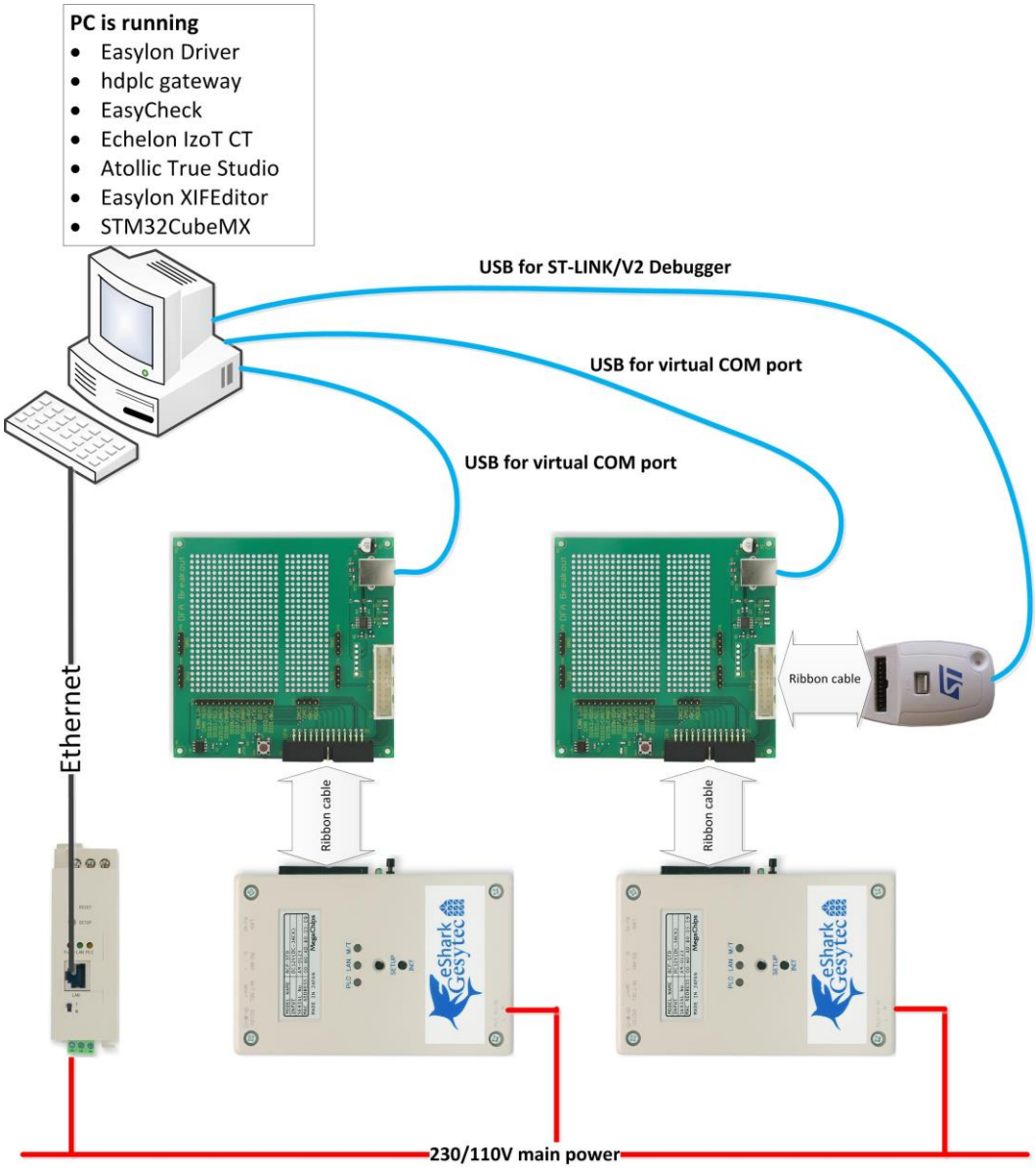
3.10.1 Microsoft Visio 2016

Echelon IzoT CT requires Microsoft Visio 2016 as user interface. If you are using the trial version of Echelon IzoT CT you need to get Microsoft Visio 2016 from another source and this needs to be installed before running the Echelon IzoT CT trial setup. A purchased version of Echelon IzoT CT includes Microsoft Visio 2016.

3.10.2 Echelon IzoT CT

Please run the setup according to Echelon’s installation guidelines.

4 Overview



The drawing shows the full setup of the EVK. In the next chapter you will guided step by step through doing the physical connections and how to verify a proper setup.

5 Setup the hardware

It is recommended using a power strip with power switch to turn on and off the whole system easily.

5.1 First steps

Select one device to be the HD-PLC Master.

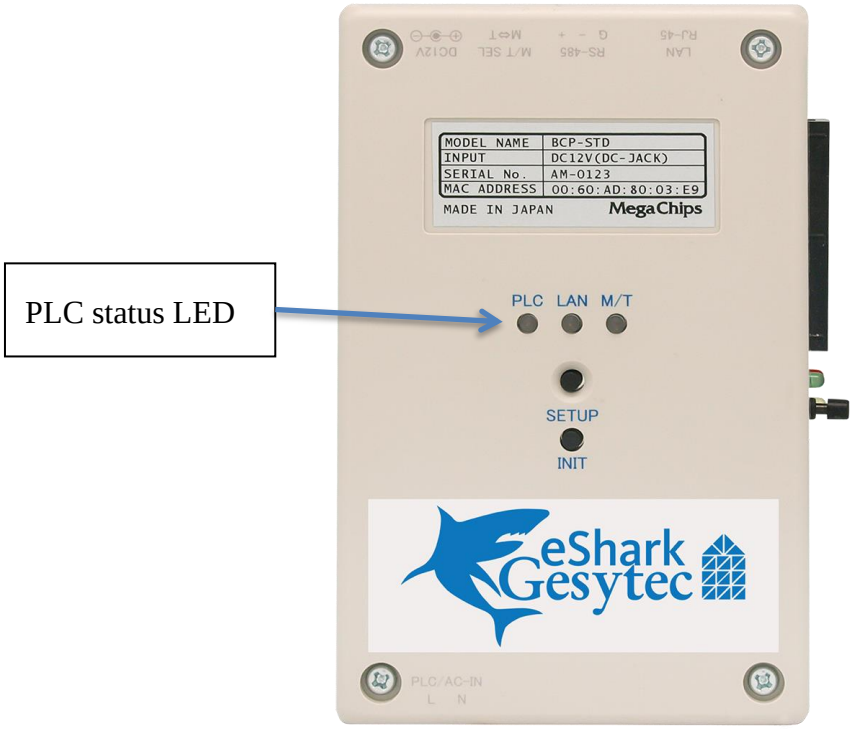
5.1.1 Easylon HD-PLC Bridge

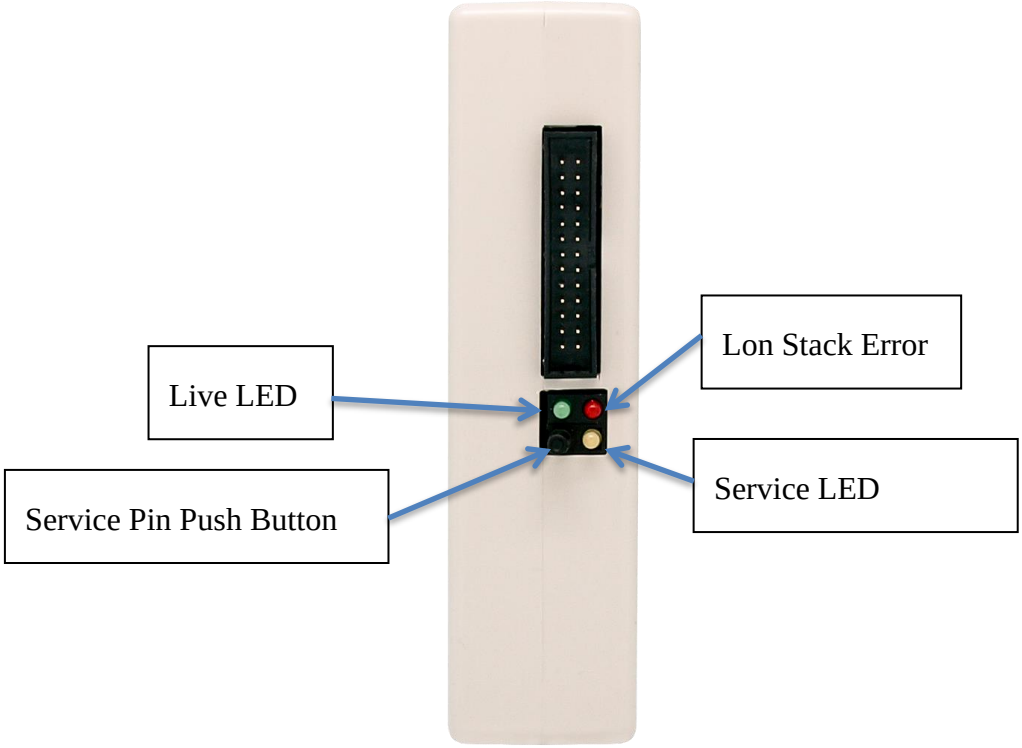
Connect a power cable to the eShark box. Verify the position of the M/T switch, it must be set to M. Connect the master HD-PLC device to the PC with an Ethernet cable. You may use an Ethernet switch.

5.1.2 eShark Box

Connect a power cable to the eShark boxes. Check the position of the M/T switch. It must be set to terminal.

Now connect all devices to main power. After a short time, the yellow PLC LED on the master device must start blinking. The green PLC LED on the eShark boxes will turn on.

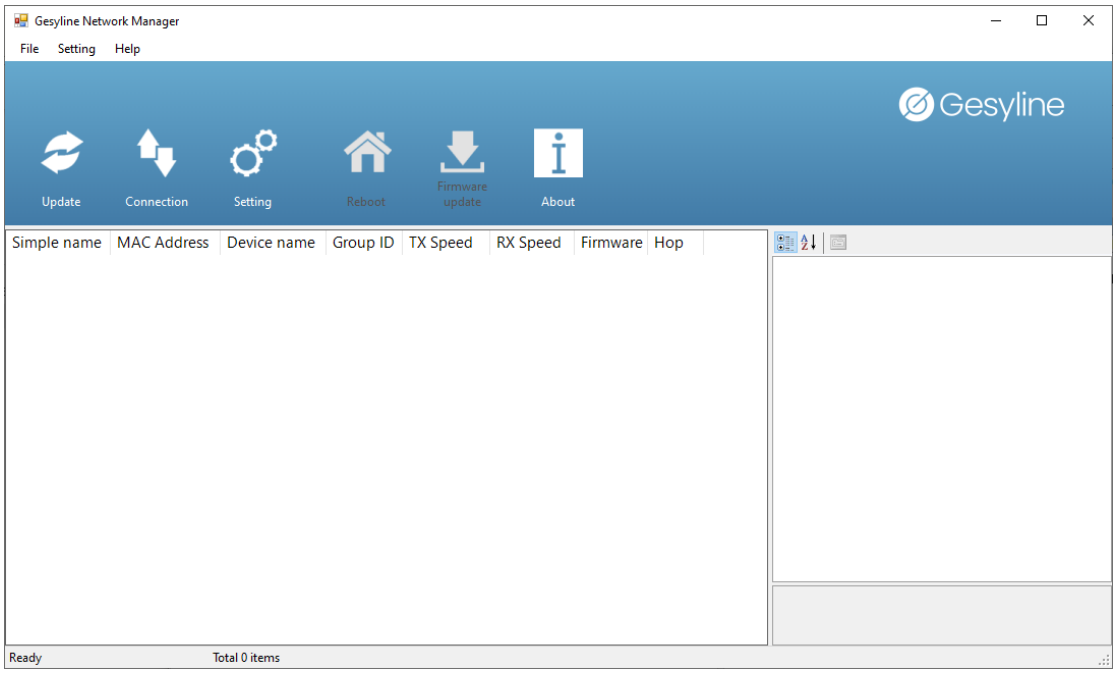




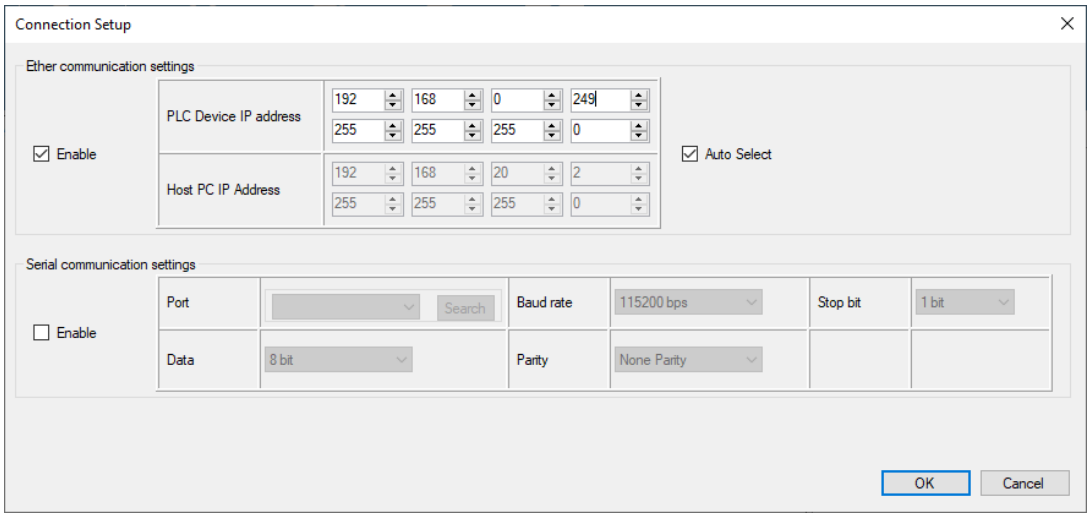
5.1.3 Gesyline Manager

Start the Gesyline Manager.

Note: The Gesyline Manager MUST be launched with administrators' rights!

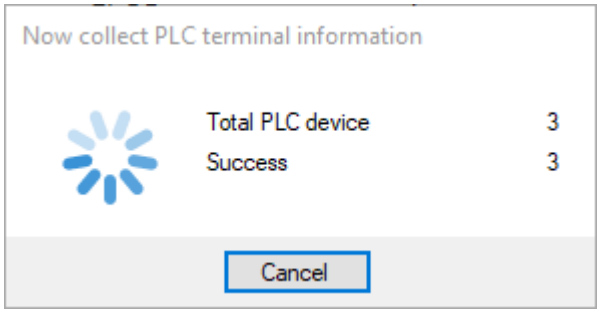


Verify the IP address setup by selecting the Connection setup dialog:

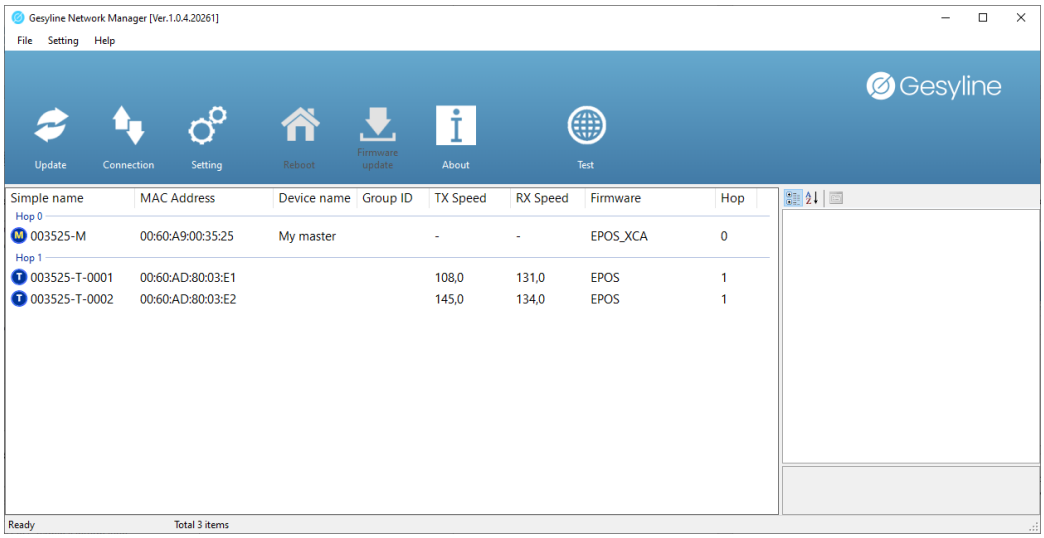


Verify the PLC Device IP address and the Host PC IP address.

Now close the dialog and press the Update button on the left side. The software now scans the HD-PLC network for devices:

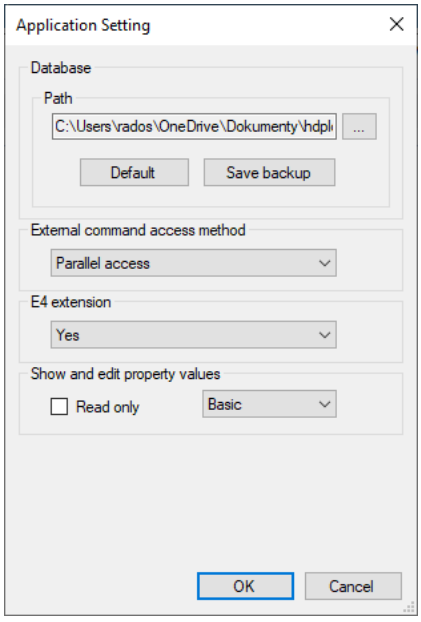


Because we have three devices in the system, three devices will be found.



Now the Gesyline manager is operational and may be used to update the firmware of a module. Select device or devices and press Firmware update button. Then reboot.

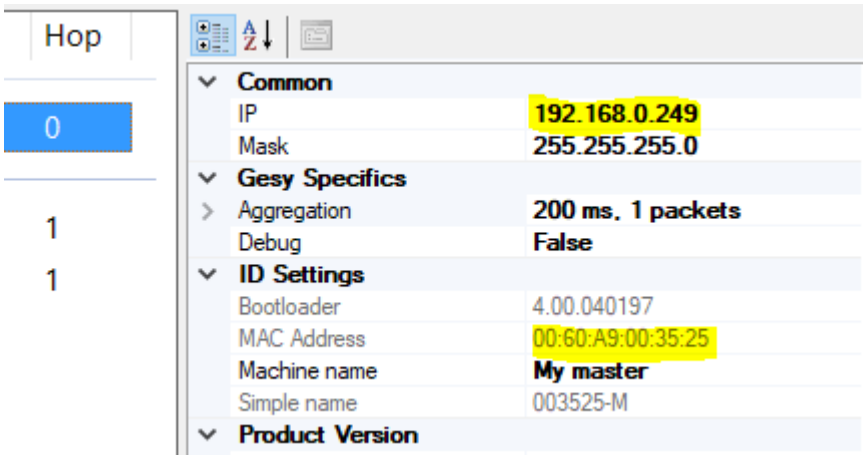
If you want to change a device setting you need to uncheck the Read only flag in the dialog Setting.



5.1.4 HD-PLC LON Master Setting

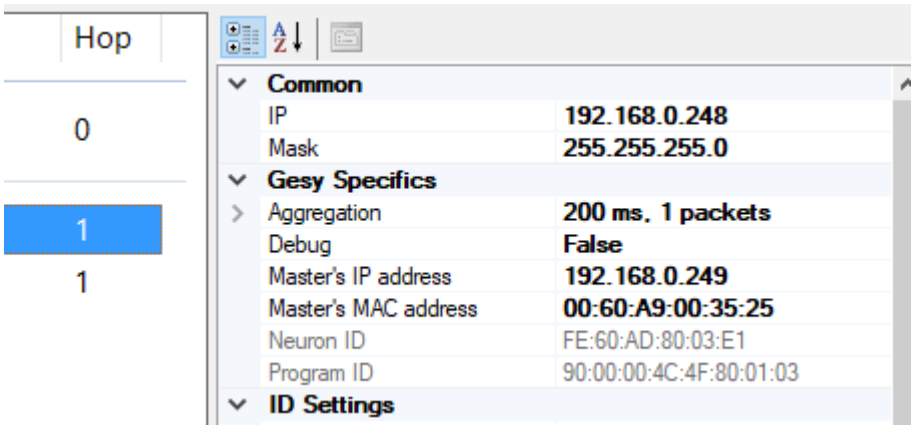
The Easylon HD-PLC Bridge (firmware name EPOS_XCA) that is the HD-PLC Master provides the HD-PLC LON Master functionality. It receives unicast messages from other eShark boxes (terminals) collects them according to the aggregation algorithm and send them further as a multicast message.

All terminal eShark boxes receive multicast and send unicast messages to the master. So, they need to know its IP and MAC address.



This is the master's setting:
IP address 192.168.0.249, MAC address 00:60:A9:00:35:25. (The MAC address of your devices is different.)

The data must be set to the paramers "Master's IP address" and "Master's MAC address" for each eShark terminal.



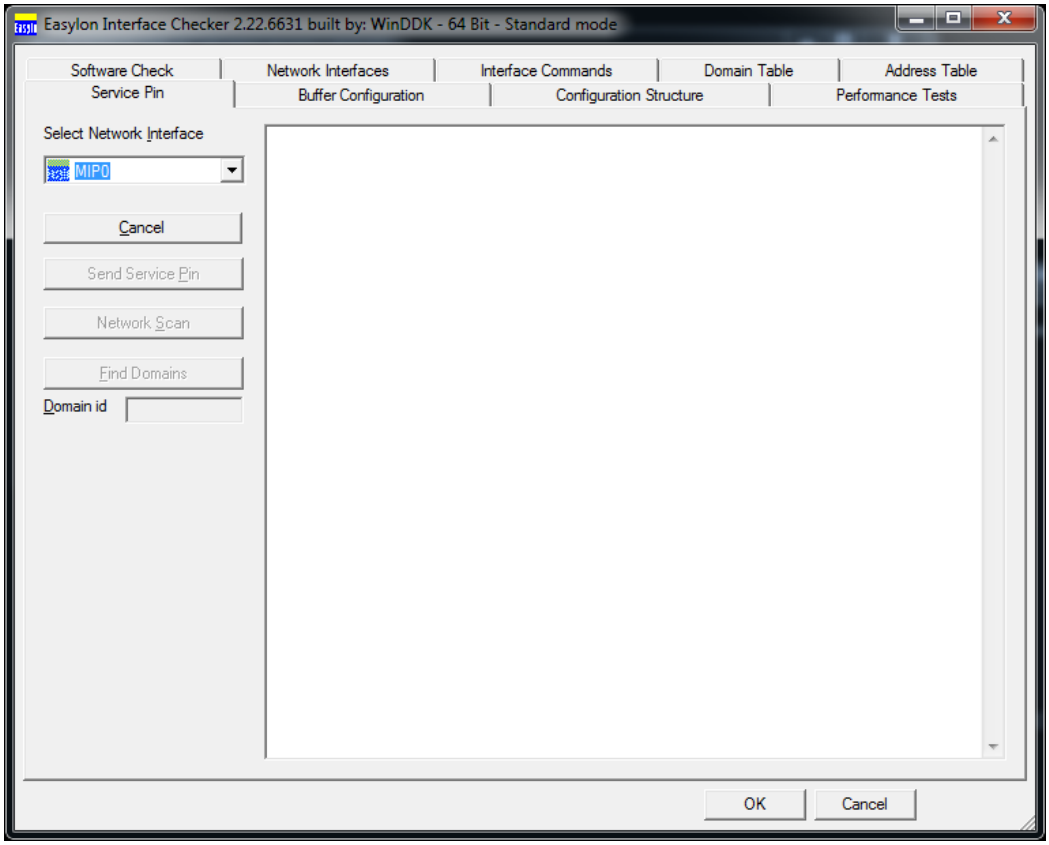
Note: The master's IP address MUST be unique. Change the IP address of the terminals to another value, for example 192.168.0.248.

5.1.5 EasyCheck

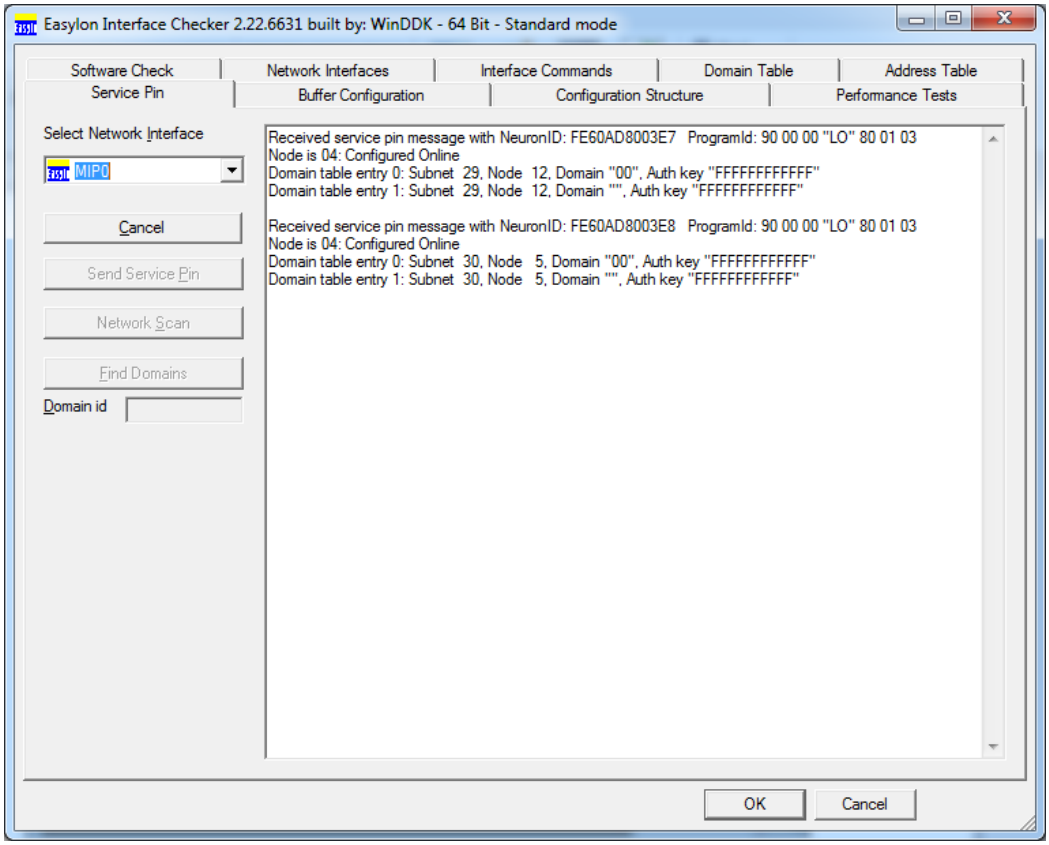
Please check that hdplc.exe is running as described above. We will do now the first HD-PLC LON communication test, using EasyCheck, the service utility for the Easylon Interfaces.

Pressing the button close to the ribbon cable connector forces the LON stack in the eShark box to send a "Service Pin Messages" which is including the NeuronID and ProgramID of the device. This message can be picked up using EasyCheck.

Start EasyCheck and select the Service Pin tab. Select in the network interface combo box MIP0 and then press the button “Wait for Service Pin”.



Now press the Service Pin button on the eShark boxes and you will have this output in EasyCheck.



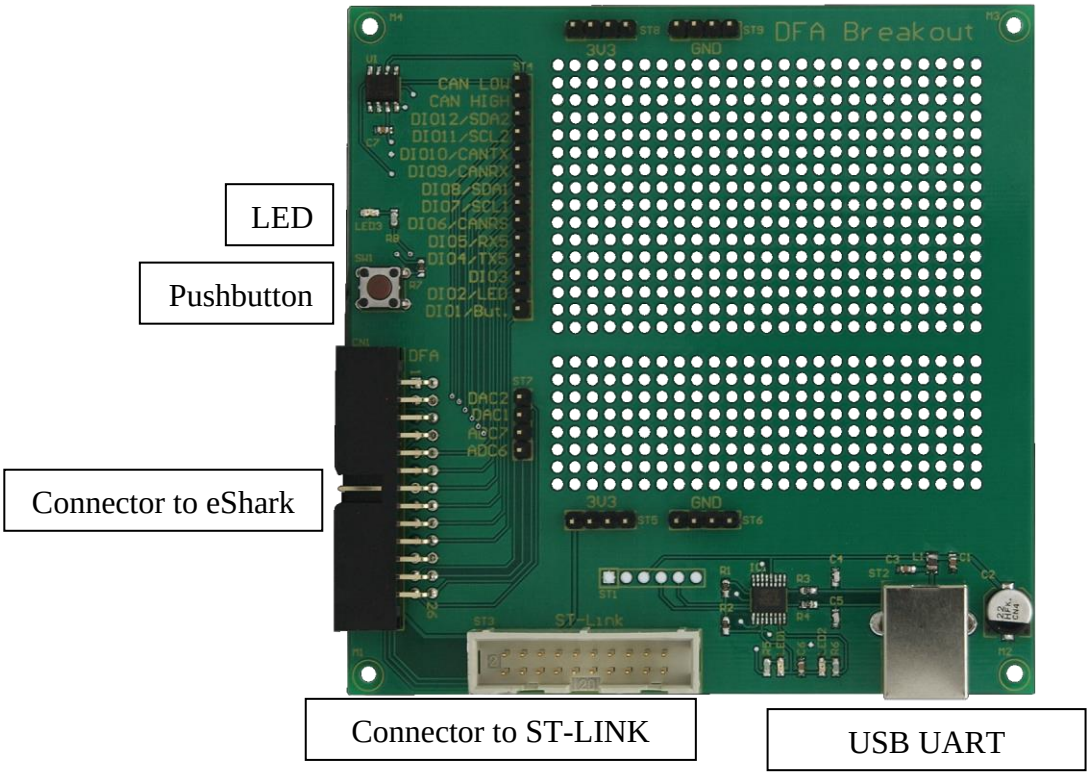
After verification that the communication is working and we are ready to go into detail.

5.2 Setup for development

Take care to turn of power for the EVK system.

5.2.1 Breakout board

Connect the breakout boards using the ribbon cable



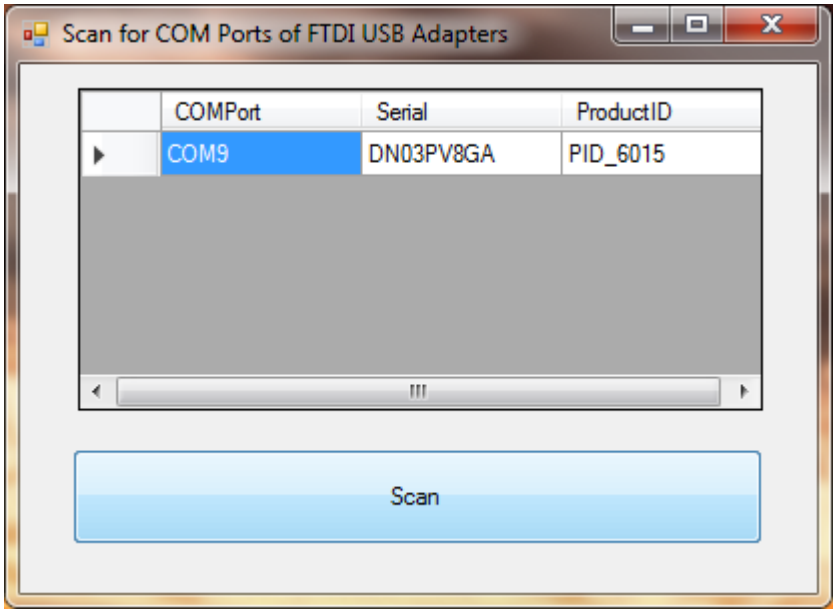
5.2.2 Debugger

Now connect the ST-LINK/V2 with a ribbon cable, the plug is marked on the silk screen printing of the breakout board.

5.2.3 USB terminal connection

Finally connect a USB cable to the USB port of the breakout board. This will install the FTDI USB drivers on the PC and generate a virtual COM port.

In **HDPLC/utilities** you will find FTDI-Ports.exe. This application scans for FTDU USB Uarts and list them together with their serial number. The serial number of the FTDI USB uart chip is printed on the breakout board.



The demo application is using 115200,n,8,1 as parameters.

5.2.4 Check with preloaded application

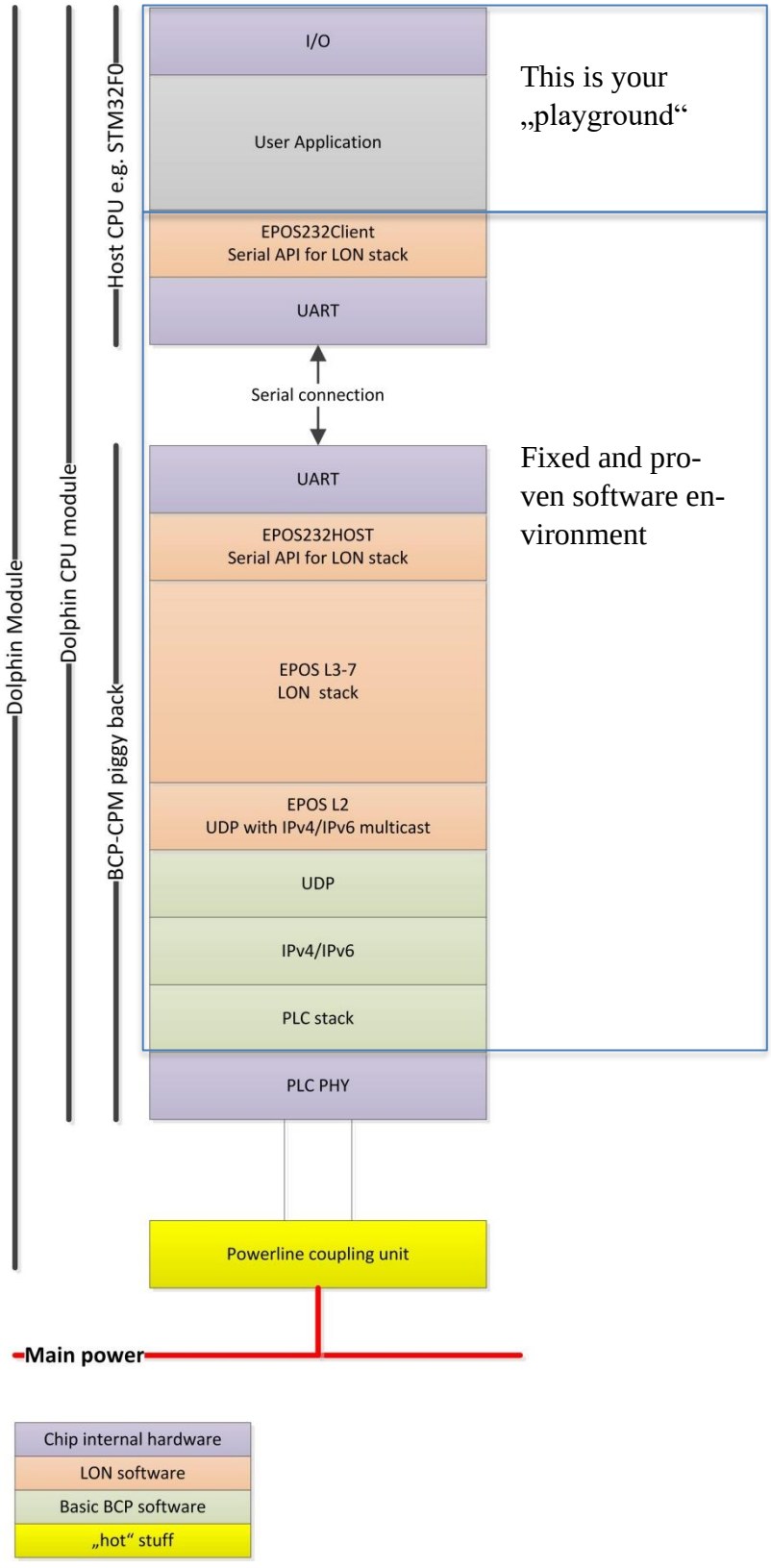
Please turn on now main power. The red LEDs on the breakout boards will turn on. Pressing the pushbutton on a breakout board will turn off the red LED of the other one.

6 Developing a LON application

In general, the development of a LON application starts with the defining of the network variables (NVs), the device shall implement, and the LonMark objects the NVs are grouped in. The EVK software helps the developer by generating an ANSI-C framework, including the definitions for the NVs and handling updates between the BlueChip and the SMT32F091CB microcontroller.

This chapter uses the provided source code of the demo application to explain the development workflow. The demo application is handling the digital and analog inputs and outputs together with the integrated temperature sensor of the STM32F091CB.

6.1 Software Structure



6.2 Structure of the LON application on the STM32F091

You will find the source code in the FTP archive, located in **HDPLC/sourcecode/DFA-EPOS**.

The demo application is using the HAL, provided by ST, which is located in the **drv** subfolder of the project. As a general rule all files provided by ST itself contain STM32 in the filename. This relates especially to the header files located in the **inc** subfolder.

Files, beginning with Epos are related to the LON application. Files, beginning with HD_BDG are containing the serial driver to the BlueChip. The files for the serial driver must remain unchanged and are not further explained.

6.2.1 Header files

EposCustomApp.h	Defines for the LON application, e.g. including the index definitions for the network variables. **It is generated by the XIF editor utility.**
EposDebug.h	The LON API generates debug prints, which are sent to UART4 of the STM32, being available with the FTDI USB UART. The print can be deactivated in this file.
EposHostApp.h	This file remains unchanged, it is for internal use by the LON API.
EposSerialComm.h	This file remains unchanged, it is for internal use by the LON API.
EposSnvtFmt.h	The demo application includes a tiny formatter to interpret network variables (SNVT). EposSnvtFmt.h belongs to EposSnvtFmt.c and might be extended for additional SNVTs.

6.2.2 Source Code

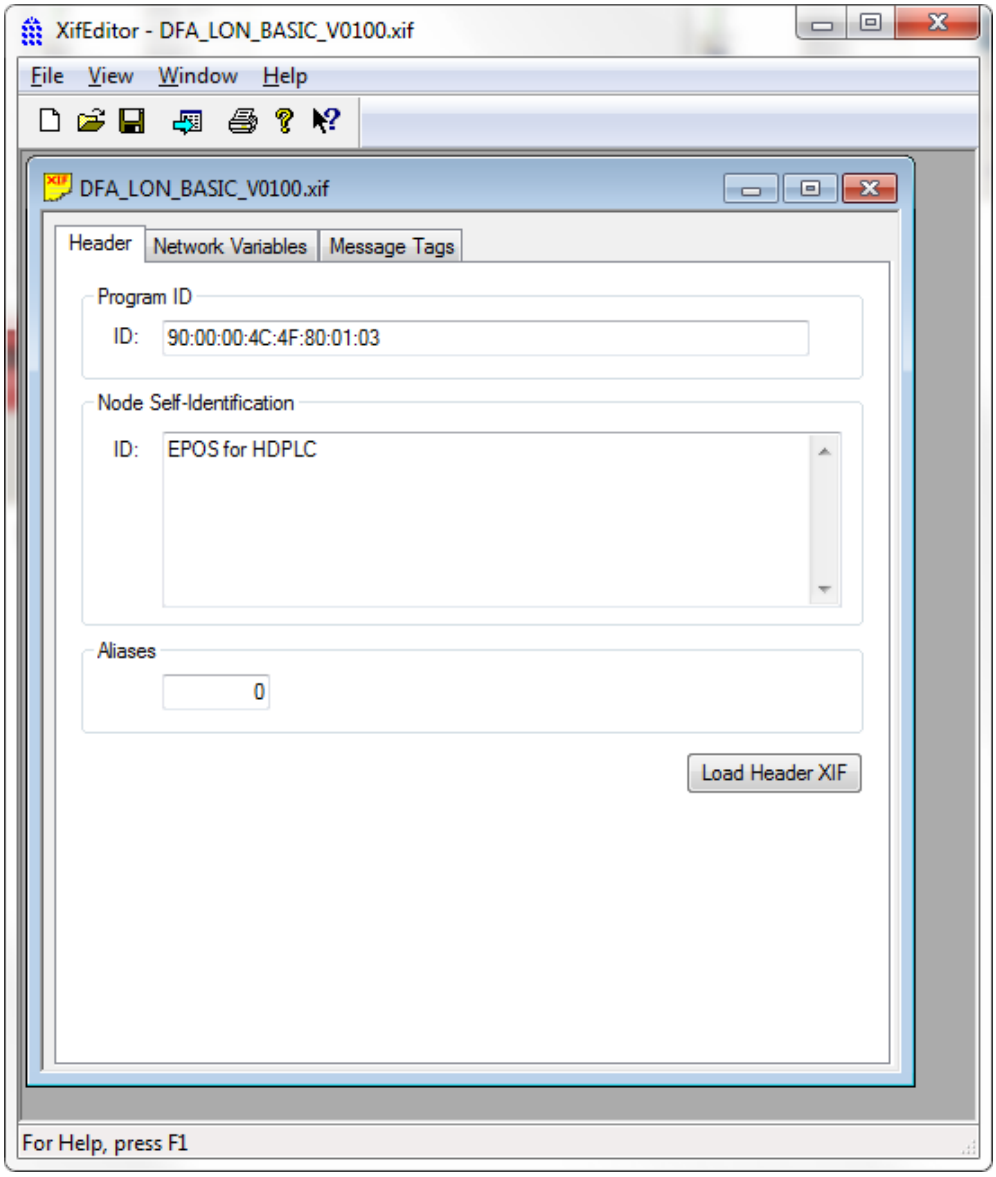
DFAPeripherals.c	In this file the handling of the peripherals is encapsulated.
EposCustomApp.c	This is the part of the application, handling the creation of network variables and updates of network variable values. **It is generated by the XIF editor utility**.
EposDebug.c	This file implements the debug print function.

EposHostApp.c	Here is the state machine of the LON api implemented. Especially it contains the setup and merry go round function of the LON api.
EposSerialComm.c	In this file the interface between the LON api and the serial interface is located. **This file remains unchanged.**
EposSnvtFmt.c	This file implements conversion functions for the interpretation of SNVTs. It might be extended for further SNVTs.
HD_BDG_*.c SerAppUtil.c serrecv.c	All this files implement the serial driver for the communication between the STM32F091CB and the BlueChip. **These files remain unchanged.**
Main.c	It includes the startup of the application and the main while(1) loop, calling the merry go round functions for the LON api and the peripheral handling. **It is generated by the STM32CubeMx utility**. User code must be placed between the /* USER CODE BEGIN */ and /* USER CODE ENDS */ comments, otherwise changes are lost after running STM32CubeMX

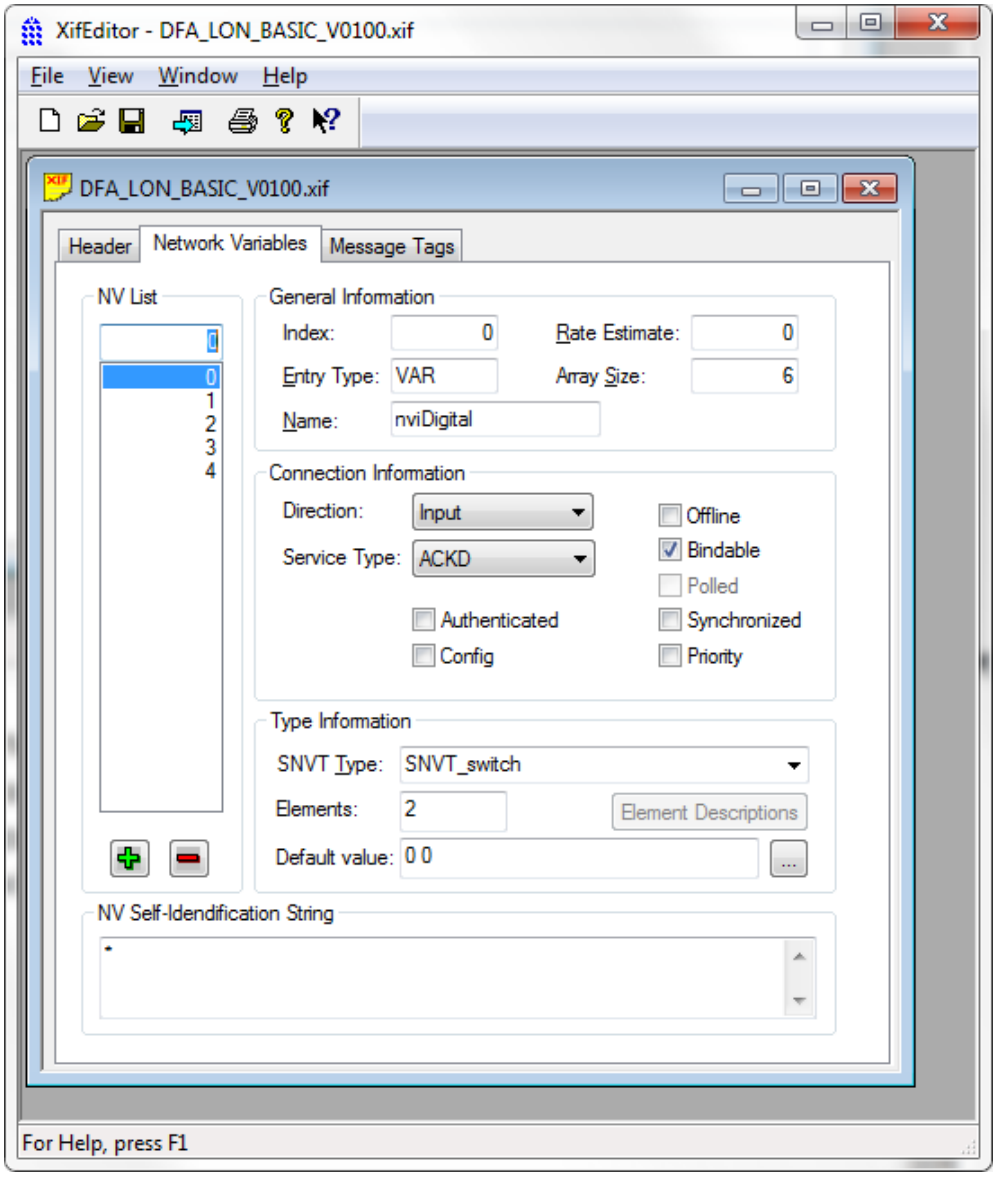
6.3 Using the XIF editor

6.3.1 Define network interface

Just start the XIF Editor, provided with the software archive for the EVK. Open the XIF file which comes with the source code of the demo application.



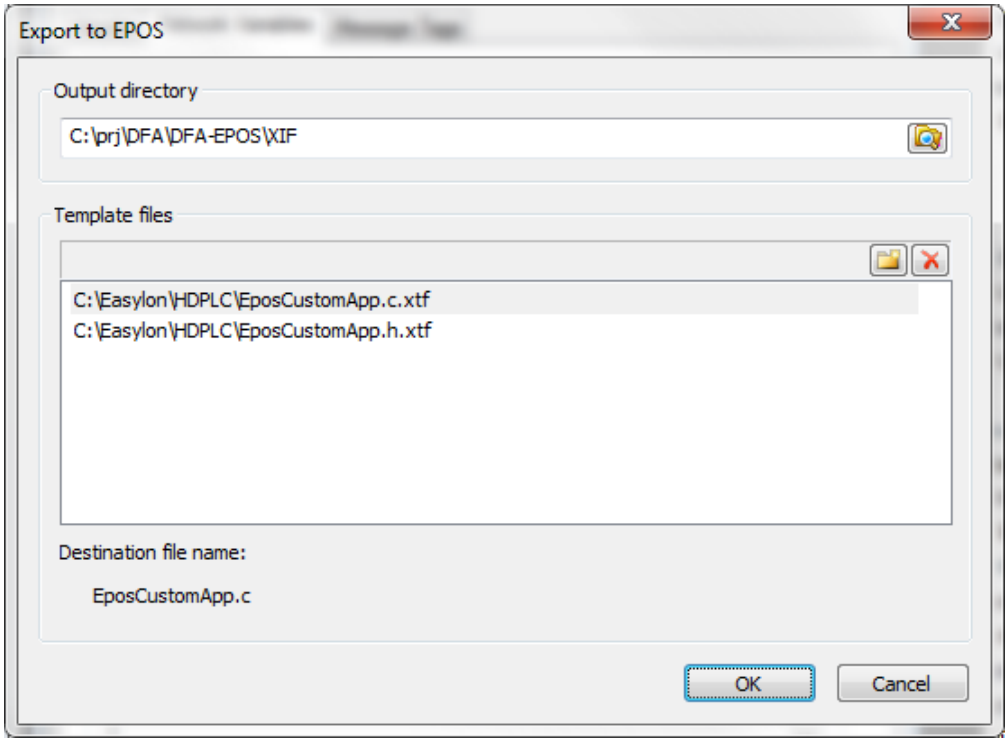
Now you are able to add or remove network variables. Adding network variables requires selecting a SNVT type. If the number or types of network variables are changed, the program id MUST be changed. Otherwise IzoT CT will complain an inconsistency with the previous definition and stop commissioning the device.



The demo application is using arrays for the GPIO related network variables, so they do have an Array Size according to the number of inputs and outputs. The demo application is configured to use 6 digital inputs and 6 digital outputs.

6.3.2 Generate ANSI-C application frame work

After defining the network variables, the C framework needs to be generated. Select the menu entry "**File->Export for EPOS**".



At first select the location of the template files with the **xtf** extension. Please refer to the installation chapter. Afterwards select the output directory, which sould be a good place in your project.

The XIFEditor generates now a new EposCustomApp.c and EposCustomApp.h. So it is wise to generate not directly into the source code tree but in a separate subfolder. Otherwise your changes in these files will be destroyed. Sync both source codes afterwards.

6.4 Drop in custom code

Now custom code needs to be dropped into the generated framework. For incoming updates, here the updates for the digital output network variable (nviDigitalx)

```
switch (msg->nv_index) {
	case NvIdx_nviDigital_0:  // 0; nviDigital[0]
		HAL_GPIO_WritePin(DIO2_GPIO_Port, DIO2_Pin,
			msg->value[0] ? GPIO_PIN_SET : GPIO_PIN_RESET);
		break;
	case NvIdx_nviDigital_1:  // 1; nviDigital[1]
		HAL_GPIO_WritePin(DIO4_GPIO_Port, DIO4_Pin,
			msg->value[0] ? GPIO_PIN_SET : GPIO_PIN_RESET);

		break;
```

In dependency of the incoming value, directly the STM32 HAL functions to set the digital output are called. Same happens for the analog outputs.

6.4.1 Remarks

The analog inputs are read using IRQ in **DFA_TIM14_IRQHandler()**. This is implemented in DFAPeripherals.c. Furthermore the timer handler toggles the green life LED of the eShark box. Because you can't call EPOS functions in an interrupt service routine, a merry go round function **DFA_HW_DoIt()** is implemented which is called from the main loop. **DFA_HW_DoIt()** is checking for changed peripheral status and calls the appropriate change function for EPOS.

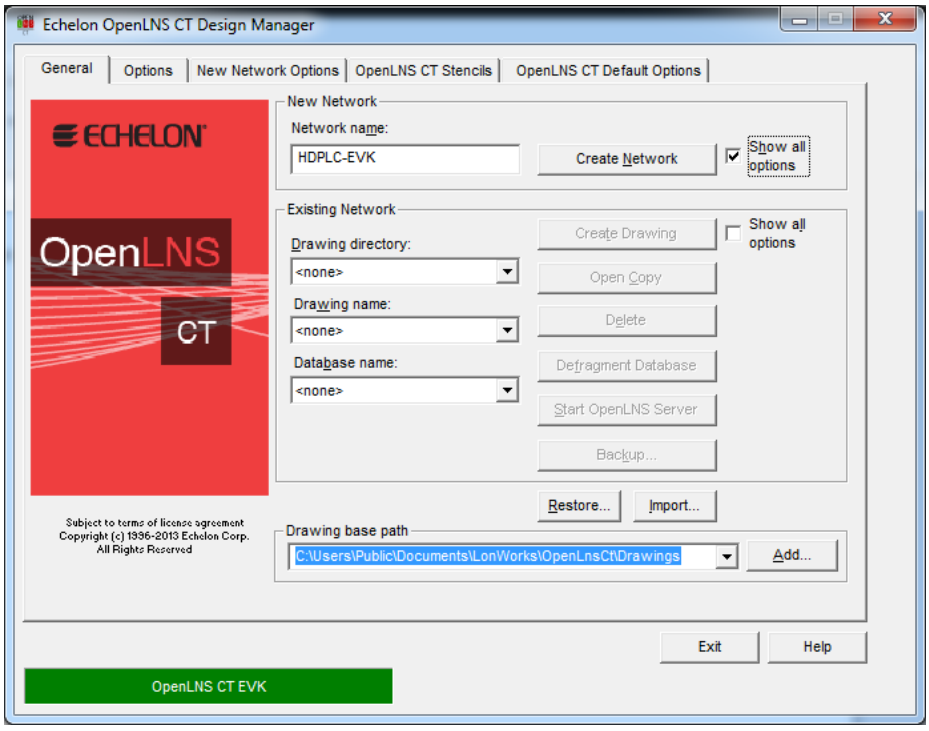
6.5 Download

After compiling the code, it can be downloaded using a connected ST-LINK/V2 debugger. You may use breakpoints to pick up the events. Additionally you may use TeraTerm to receive the debug prints of the application.

6.6 Commission in IzoT CT

When the system is running you are able now to generate a connection between input and output network variables. In the following sample, the connection between the pushbutton and the red LED is demonstrated.

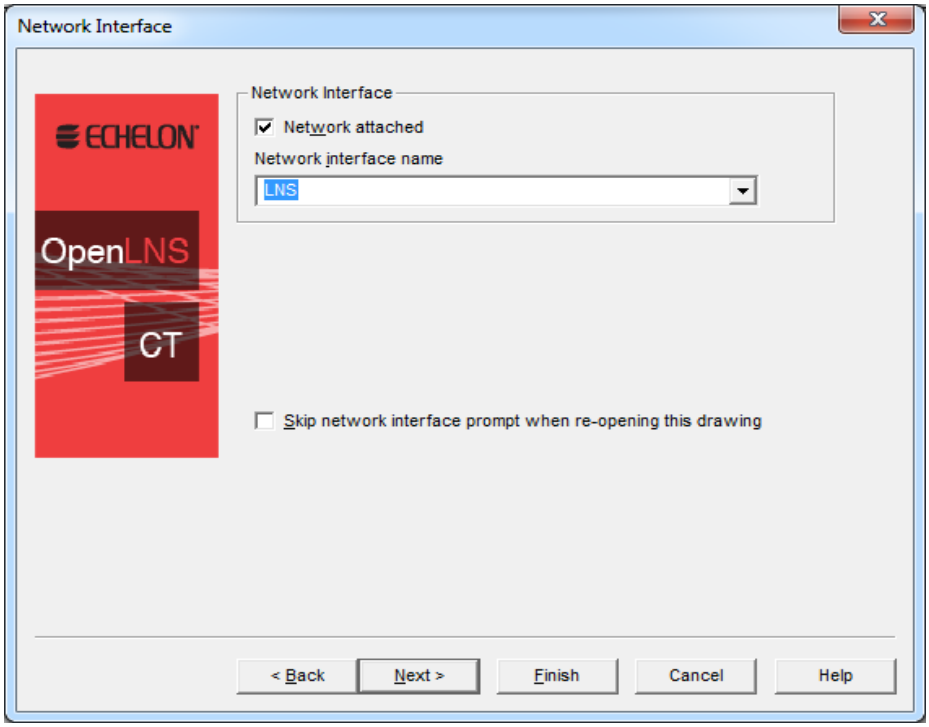
First start IzoT CT



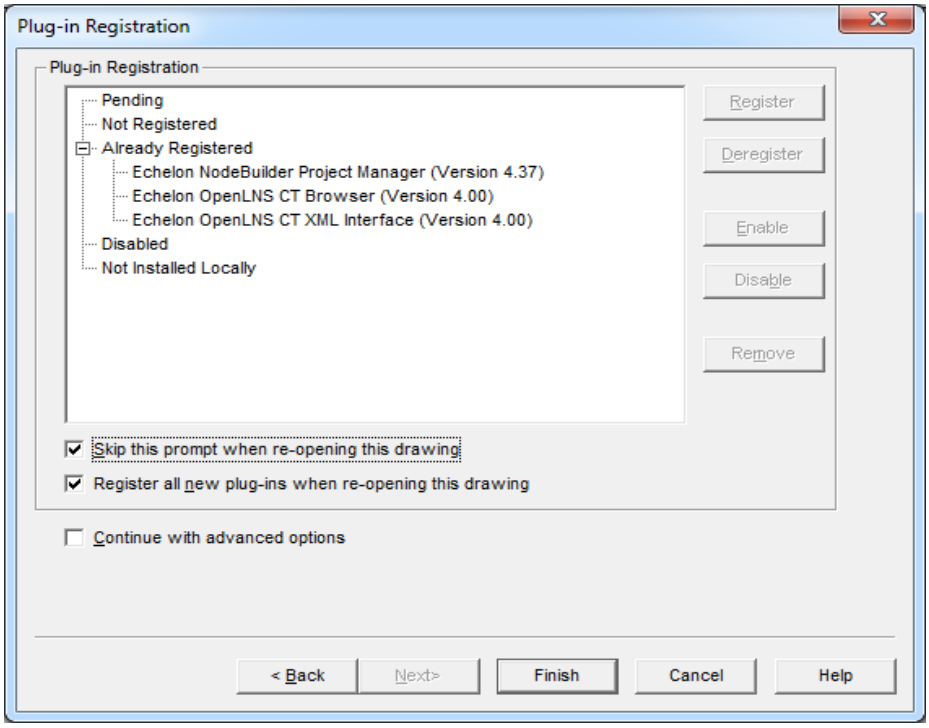
Define a name in the New Network frame, tick the "Show all options" checkbox and press the **Create Network** button.

Microsoft Visio is starting now and will act as graphical user interface for IzoT CT.

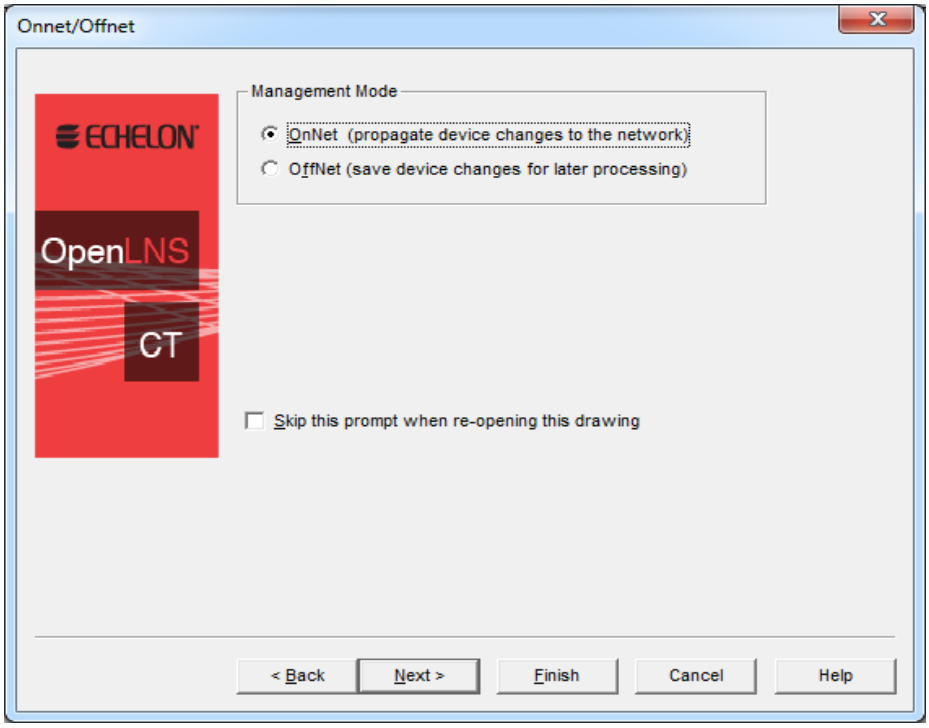
The next dialog asks for the Network interface name, select Network attached and LNS as Network interface name:



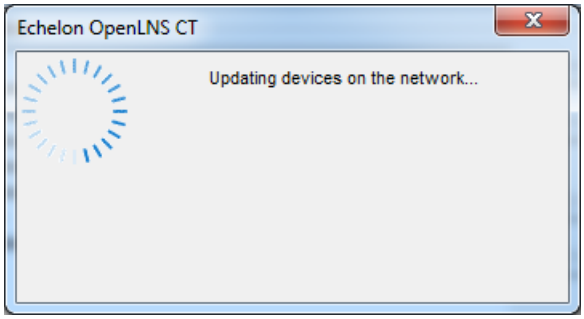
IzoT CT asks now for Plug-In registration. Please tick the checkboxes as shown below and press Finish.



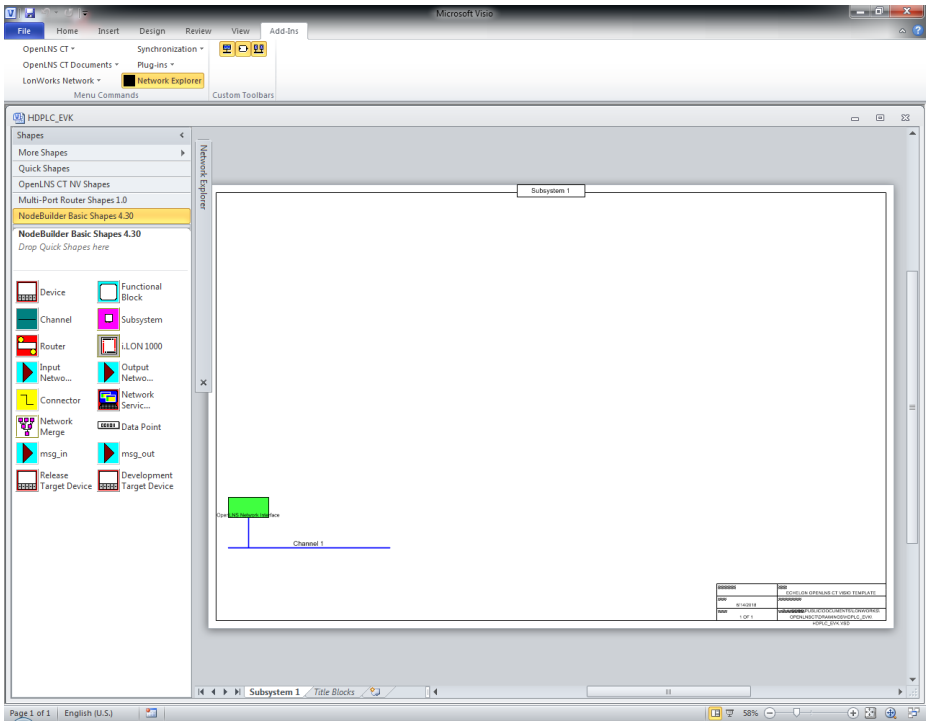
Select the Management Mode OnNet in the next formular:



Now IzoT CT is preparing the network:

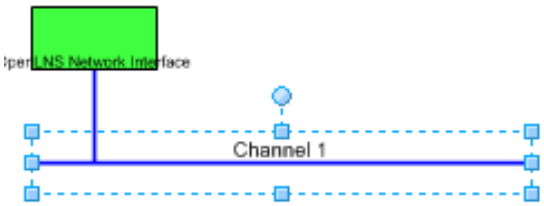


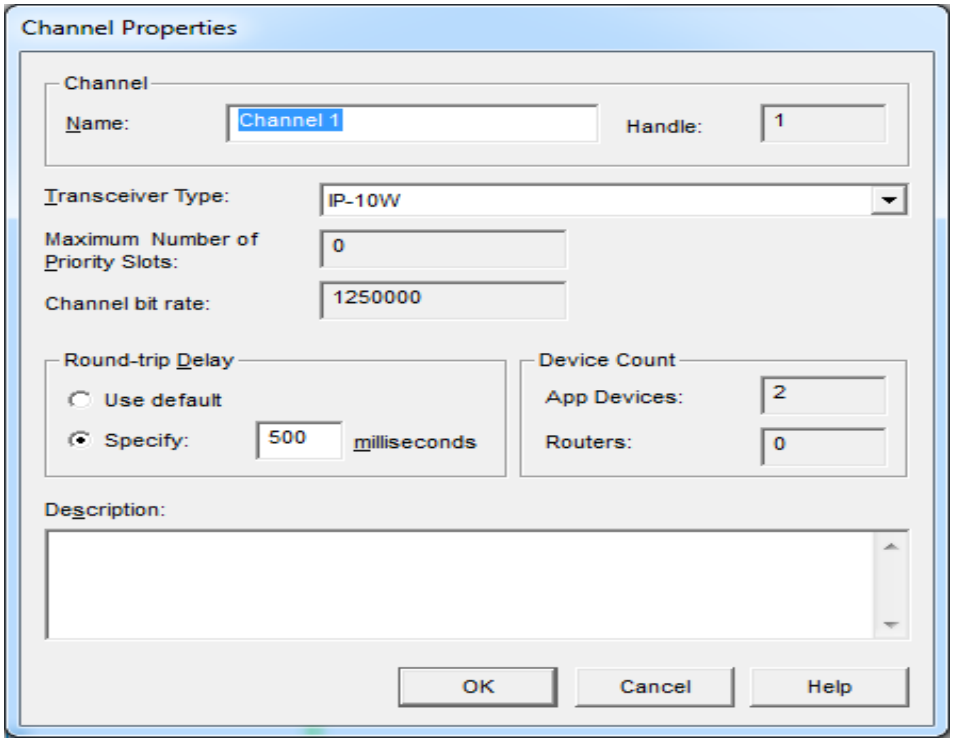
Finally you will have this view in Visio:



Now we need to adapt the channel settings, right now IZot CT does not provide settings for an HD-PLC LON channel.

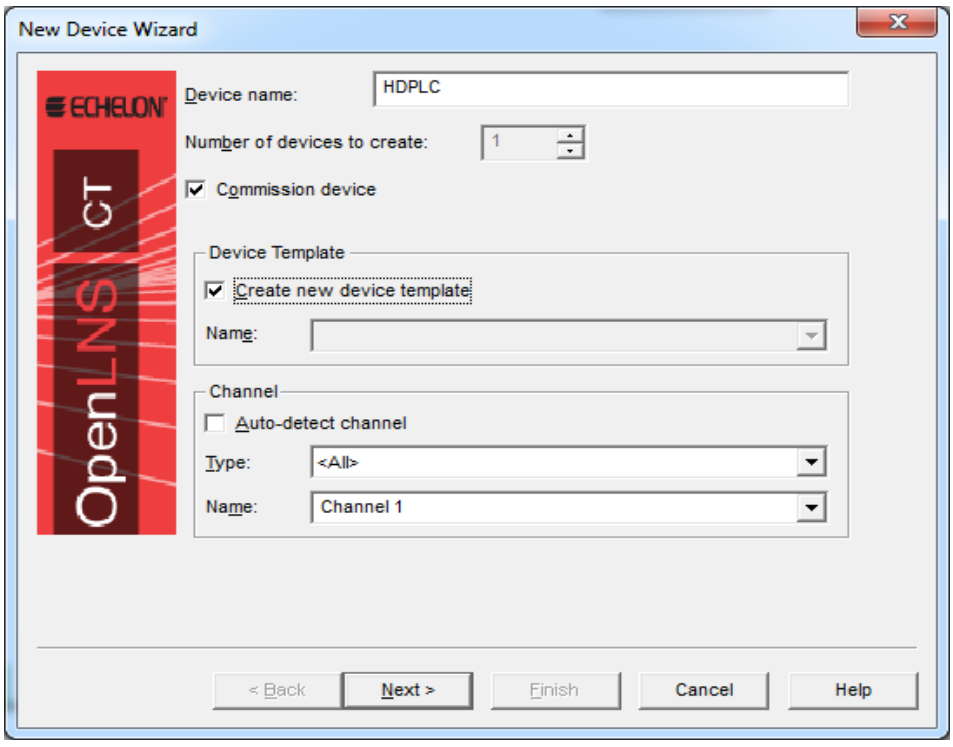
Right click on “Channel 1” and select Properties:



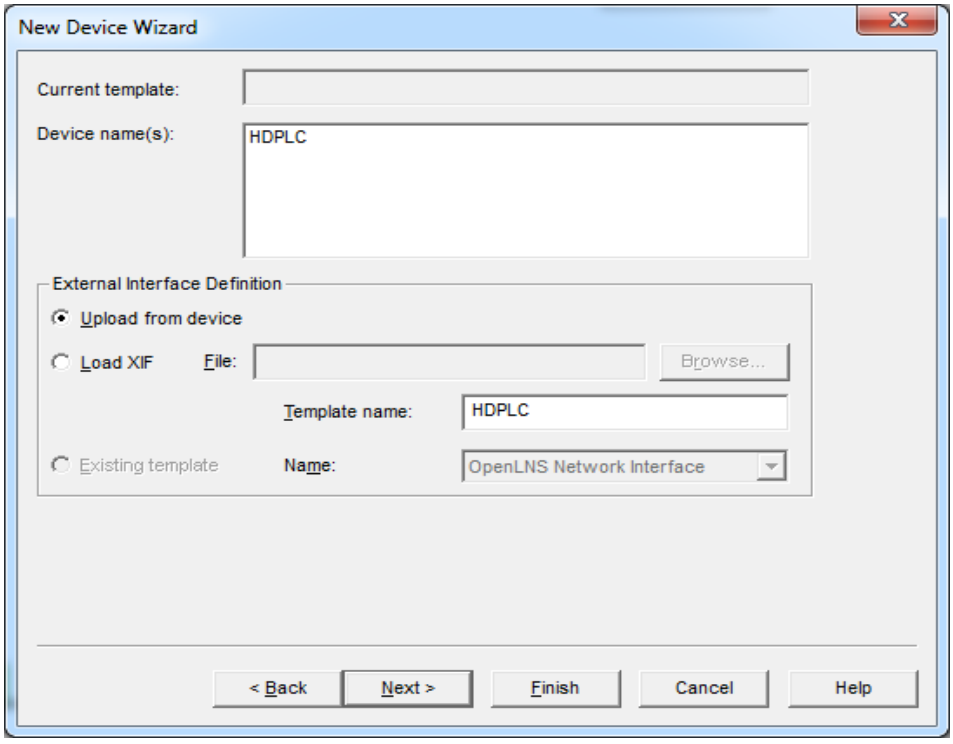


Select Transceiver Type **IP-10W** which sets similar timing compared with HD-PLC LON.

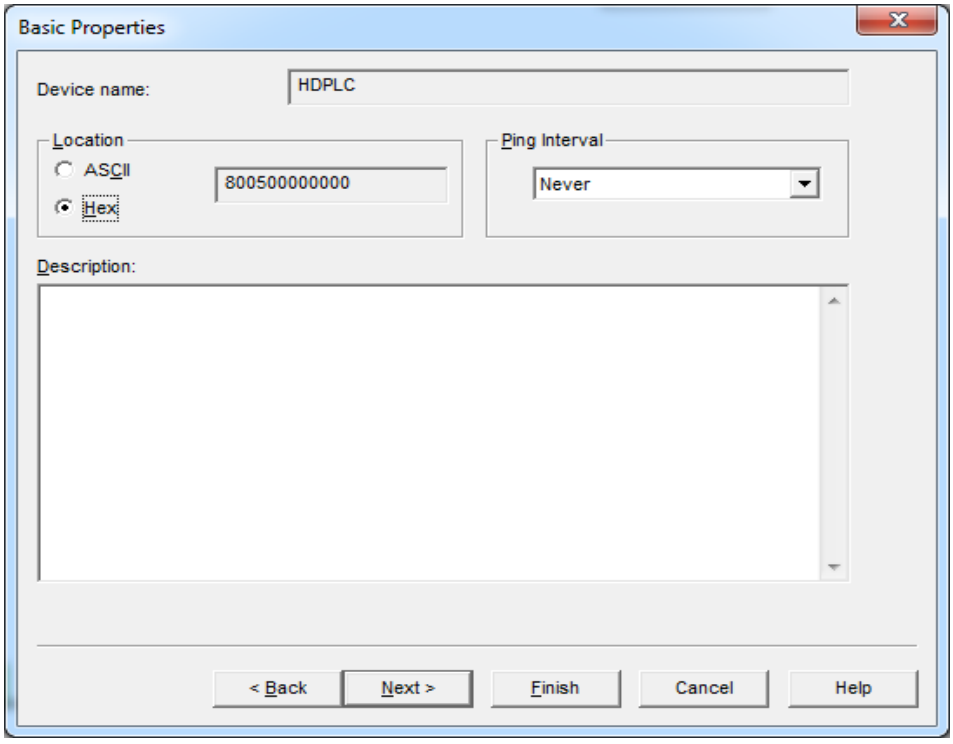
Now drag a device shape from the LON shapes to the working sheet:



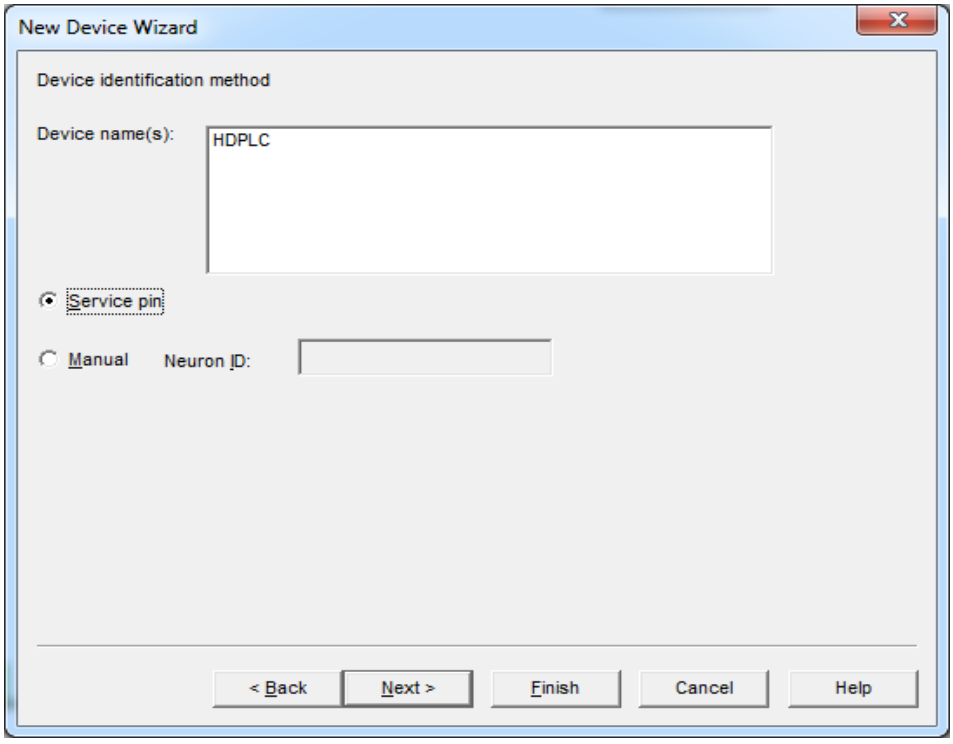
Check the checkboxes as shown and press Next.



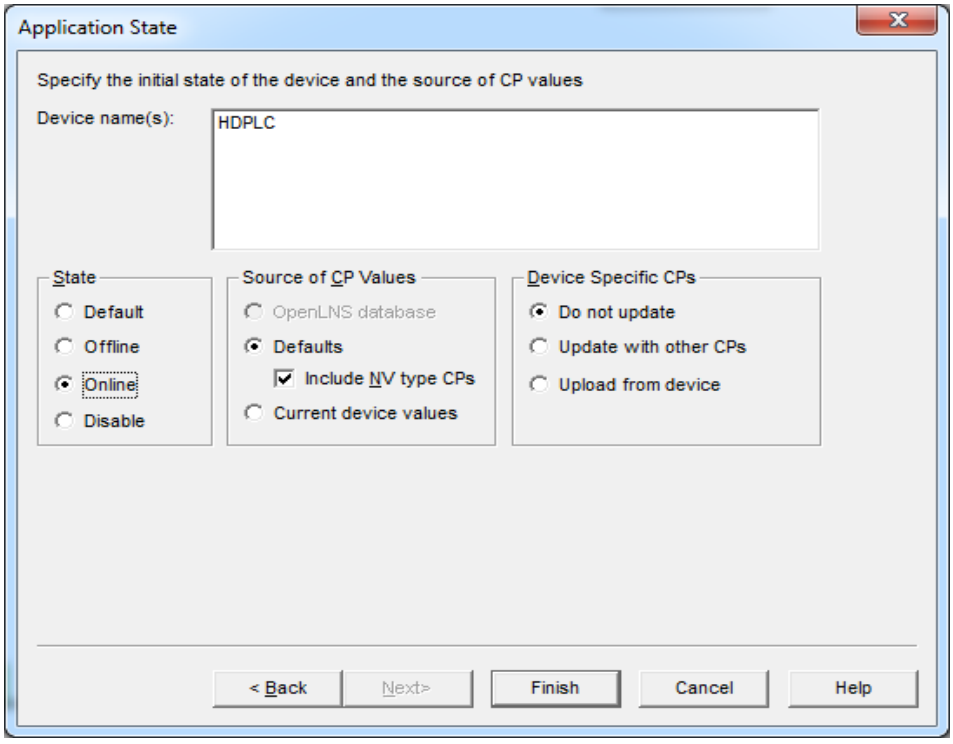
Please select here „Upload from Device“ and press Next.



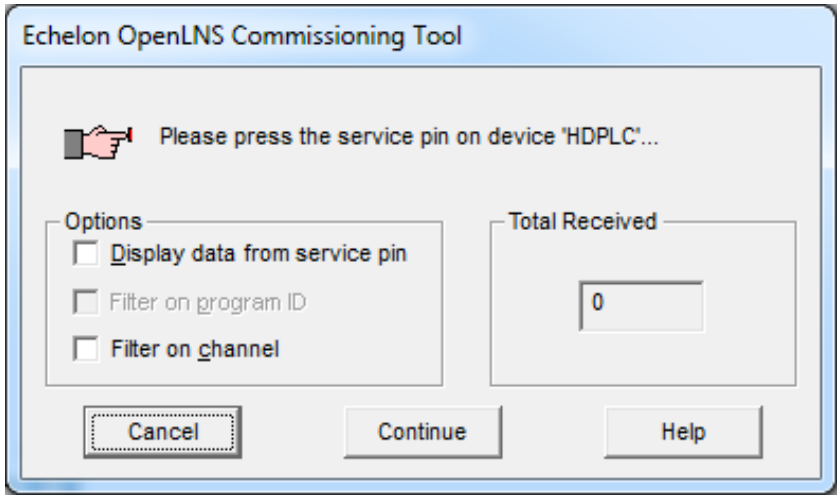
No changes are needed here, just press again Next.



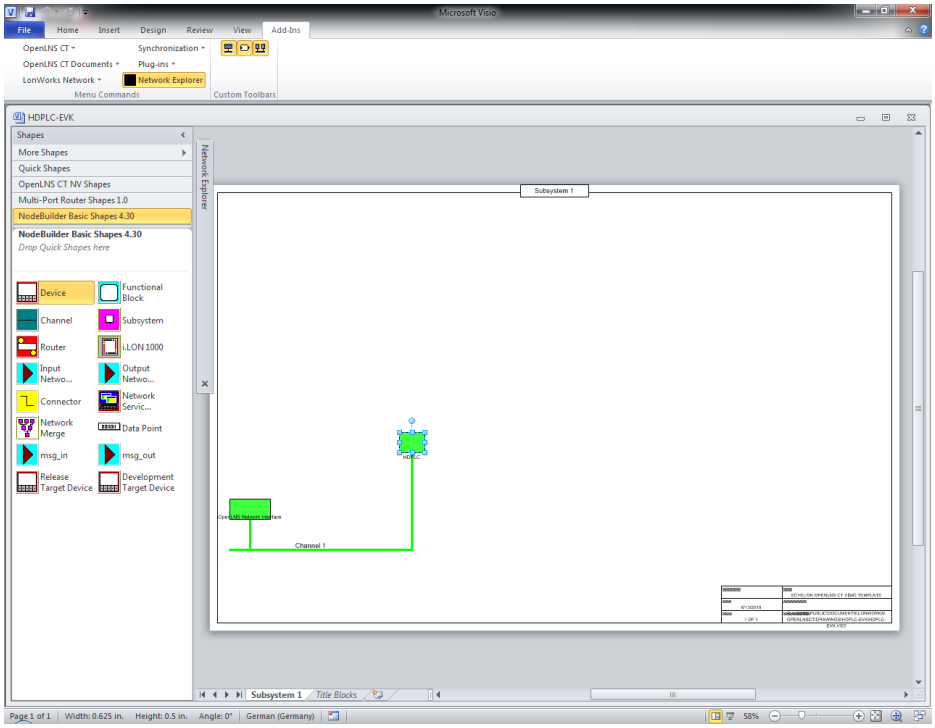
Select Service Pin option here and press Next.



In State you need to set Online, the press Next.

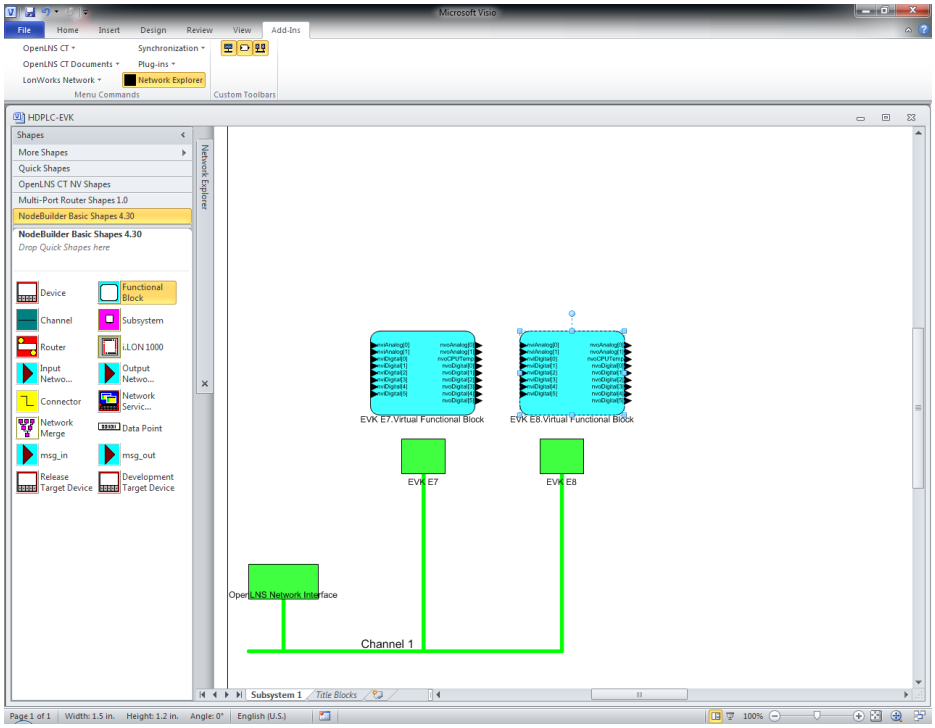


Press nothing in this dialog! But press the service pin push button of the first eShark box, IzoT CT starts now commissioning the box and after a short time it appears on the working sheet.

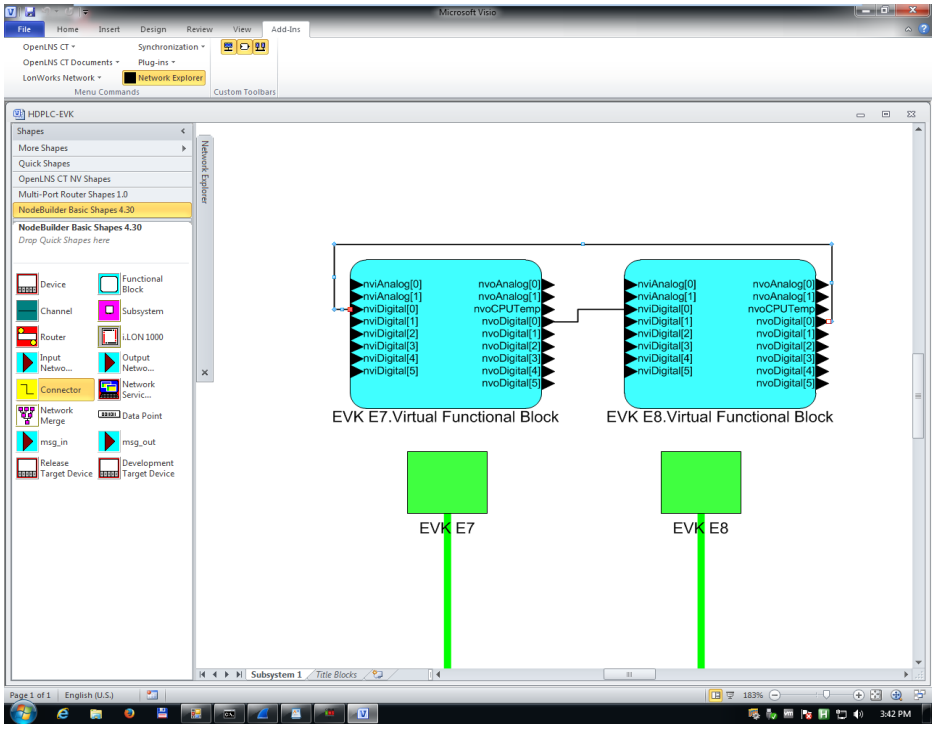


Please repeat the procedure with the second box.

Now drag a functional block shape on the sheet and tick create for all network variables and repeat this for the second device.



Finally drag a Connector on the sheet and connect nvoDigital[0] of one box with nviDigital[0] of the other box. This maps the status of the pushbutton on the red LED of the other board. Repeat the procedure vice versa for the second box.



Congratulations, now you created your first HD-PLC LON system.

7 EPOS API for STM32

7.1 Diagnostics

DPRINTF
This macro effects a debug printf on UAR4 on the breakout board. It is used like this:
```
DPRINTF(("EPOS AI %d %d", nvIndex, value));
```
take care on the double parenthesis

DUMPBUFFER
generates a hexdump of a byte buffer:
```
DUMPBUFFER((">> ", msg->value, msg->length));
```
it starts with a text, followed by the address of the byte buffer and the length.

8 Hardware details

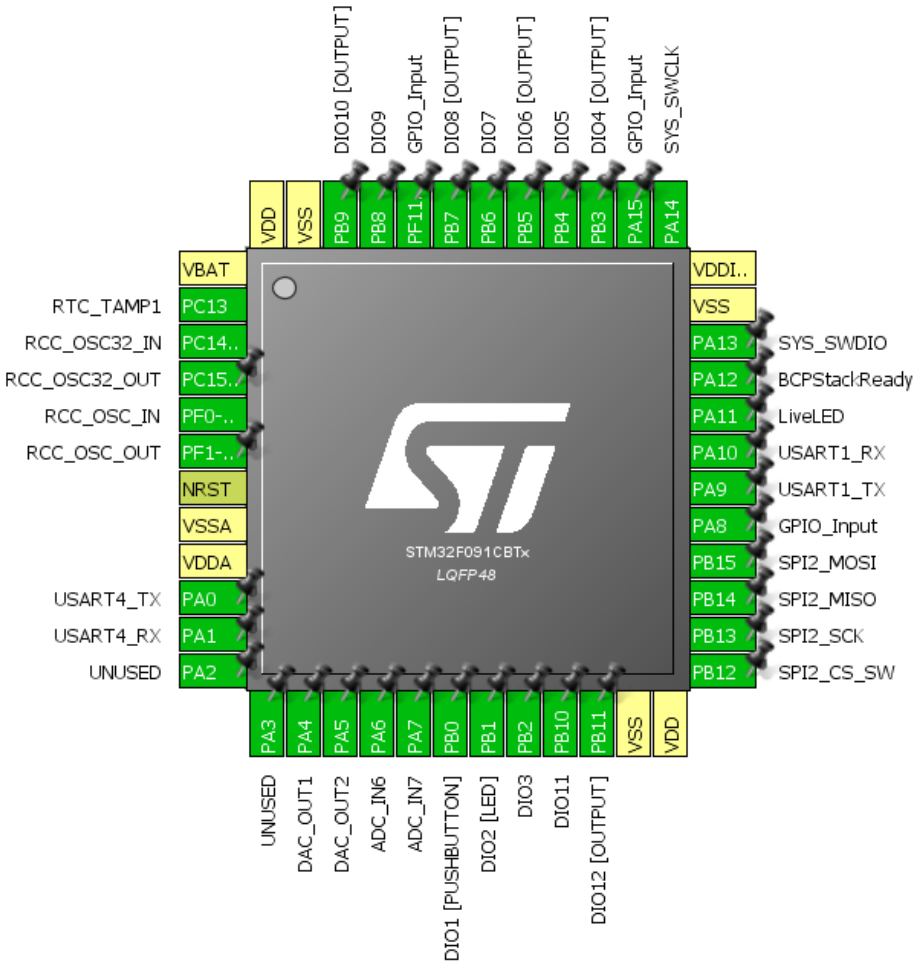
8.1 Peripherals

eShark box header:

	Row-A	Row-B	Comment
1	DAC_OUT1	DAC_OUT2	fixed
2	ADC_IN6	ADC_IN7	fixed
3	GND-A	VCC-A	
4	PB0	PB1	programmable digital
5	PB2	PB3	programmable digital
6	PB4	PB5	programmable digital
7	PB6	PB7	programmable digital
8	PB8	PB9	programmable digital
9	PB10	PB11	programmable digital
10	GND	VCC	
11	SWDIO	SWCLK	ARM Debug
12	GND	Reset	ARM Debug
13	UART4TX	UART4_RX	Diagnostics

8.2 STM32CubeIDE

The picture shows the GPIO setting, used by the demo application. For changes, just PB0-PB11 can be changed. The other functions are fixed. The demo application is using PB0-PB11 as digital inputs and outputs.



8.3 STM32F091 pin out

Pin Name	Con-nector	Usage	
PA0	y	Debug UART	**USART4_TX**
PA1	y	Debug UART	**USART4_RX**
PA2		Unused	
PA3		Unused	
PA4	y	User	**DAC_OUT1**
PA5	y	User	**DAC_OUT2**
PA6	y	User	**ADC_IN6**
PA7	y	User	**ADC_IN7**
PA8		Unused	
PA9		BCP UART	USART1_TX
PA10		BCP UART	USART1_RX
PA11		Live LED	PA11
PA12		BCP Stack Ready	PA12
PA13	y	Debug	SWDIO
PA14	y	Debug	SWCLK
PA15		Unused	
Pin name			
PB0	y	User	flexible
PB1	y	User	flexible
PB2	y	User	flexible
PB3	y	User	flexible
PB4	y	User	flexible
PB5	y	User	flexible
PB6	y	User	flexible
PB7	y	User	flexible
PB8	y	User	flexible
PB9	y	User	flexible
PB10	y	User	flexible
PB11	y	User	flexible
PB12		**SPI2_NSS**	SPI1
PB13		**SPI2_SCK**	SPI1
PB14		**SPI2_MISO**	SPI1
PB15		**SPI2_MOSI**	SPI1