

www.rexygen.com

Getting started with $\mathsf{REXYGEN}$ on UniPi Neuron

User guide

REX Controls s.r.o.

Version 2.50.12 2022-11-22 Plzeň (Pilsen), Czech Republic

Contents

1	Intr	roduction	3
	1.1	Features of the REXYGEN system	3
	1.2	Structure of the REXYGEN system	4
	1.3	Programming in the REXYGEN system	4
	1.4	Main components of the REXYGEN system	4
		1.4.1 REXYGEN Studio – the development environment	4
		1.4.2 REXYGEN HMI Designer	5
		1.4.3 REXYGEN Compiler	5
		1.4.4 RexCore	5
		1.4.5 REXYGEN Diagnostics	5
2	Inst	tallation of REXYGEN development tools	6
	2.1	Windows 7/8/10	6
		2.1.1 Installation procedure	6
		2.1.2 Uninstall procedure	7
	2.2	GNU/Linux	7
3	Inst	talling the runtime modules of the REXYGEN system on UniPi Neu-	
3	Inst ron	talling the runtime modules of the REXYGEN system on UniPi Neu-	8
3		talling the runtime modules of the REXYGEN system on UniPi Neu- Install using operating system image with REXYGEN	8 8
3	ron	Install using operating system image with REXYGEN	8
3	ron 3.1	Install using operating system image with REXYGEN	8 8
3	ron 3.1 3.2	Install using operating system image with REXYGEN	8 8 8
3	ron 3.1 3.2 3.3 3.4	Install using operating system image with REXYGEN	8 8 8 9
	ron 3.1 3.2 3.3 3.4	Install using operating system image with REXYGEN Manual installation on running UniPi Neuron Starting the runtime Installed files and folders	8 8 9 9
	ron 3.1 3.2 3.3 3.4 Lice	Install using operating system image with REXYGEN	8 8 9 9 9
	ron 3.1 3.2 3.3 3.4 Lice 4.1	Install using operating system image with REXYGEN Manual installation on running UniPi Neuron Starting the runtime Installed files and folders Installed files and folders ensing REXYGEN using REXYGEN DWM on UniPi Neuron Obtaining a DEMO licence	8 8 9 9 9 10 11
	ron 3.1 3.2 3.3 3.4 Lice 4.1 4.2	Install using operating system image with REXYGEN Manual installation on running UniPi Neuron Starting the runtime Installed files and folders Installed files and folders ensing REXYGEN using REXYGEN DWM on UniPi Neuron Obtaining a DEMO licence Obtaining a permanent licence	8 8 9 9 9 10 11 14
4	ron 3.1 3.2 3.3 3.4 Lice 4.1 4.2	Install using operating system image with REXYGEN	8 8 9 9 10 11 14 14 14
4	ron 3.1 3.2 3.3 3.4 Lice 4.1 4.2	Install using operating system image with REXYGEN Manual installation on running UniPi Neuron Starting the runtime Installed files and folders Installed files and folders ensing REXYGEN using REXYGEN DWM on UniPi Neuron Obtaining a DEMO licence Obtaining a permanent licence 4.2.1 Activation of the permanent licence Sault REXYGEN project on UniPi Neuron	8 8 9 9 10 11 14 14 14 16

Con	figuration, compilation and execution	23
6.1	Creating a new project	23
6.2		28
6.3		34
6.4	Ready for interaction with the outer world	40
I/O	configuration for the UniPi Neuron	41
7.1	Physical connections	41
7.2	Adding inputs and outputs to the project	41
7.3	Working with I/O signals	42
	7.3.1 Modifications in the project main file	42
	7.3.2 Modifications in the task	44
7.4	Updating the HMI	46
7.5	Additional information	47
	7.5.1 Detailed description of the driver	47
	7.5.2 Examples	47
Sun	nmary	48
Gra	phical HMI with REXYGEN HMI Designer	49
A.1	· · · · · · · · · · · · · · · · · · ·	
A.2		
A.3		53
A.4		58
A.5		59
A.6	The final steps	60
Lice	unsing of the REXYGEN system	63
	0	63
		66
<u></u>		
	1	
	6.1 6.2 6.3 6.4 I/O 7.1 7.2 7.3 7.4 7.5 Sum Gra A.1 A.2 A.3 A.4 A.5 A.6 Lice	 6.2 Compiling and running a project 6.3 Adding a user interface (HMI) 6.4 Ready for interaction with the outer world I/O configuration for the UniPi Neuron 7.1 Physical connections 7.2 Adding inputs and outputs to the project 7.3 Working with I/O signals 7.3.1 Modifications in the project main file 7.3.2 Modifications in the project main file 7.3.2 Modifications in the task 7.4 Updating the HMI 7.5 Additional information 7.5.1 Detailed description of the driver 7.5.2 Examples Summary Graphical HMI with REXYGEN HMI Designer A.1 Initializing the HMI design A.2 Adding the first HMI components A.3 Linking HMI components with the running algorithm A.4 Adding more HMI components A.5 Additional links to the running algorithm A.6 The final steps

Introduction

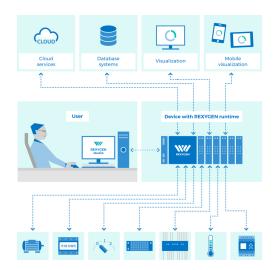
The **REXYGEN** system is a family of software products for automation projects. You can use it in all fields of automation, robotics, measurements and feedback control.

The runtime core of the **REXYGEN** system turns your UniPi Neuron into a programmable device which will run your algorithms.

1.1 Features of the **REXYGEN** system

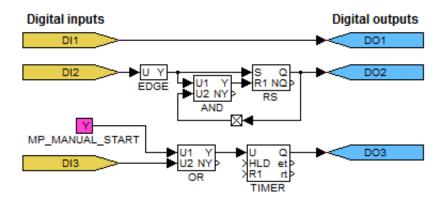
- Graphical programming without hand-coding
- Programming control units on a standard PC or laptop
- User interface for desktop, tablet and smartphone (HMI)
- Wide family of supported devices and input-output units (including UniPi Neuron)
- Industry-proven control algorithms
- Easy integration into business IT infrastructure (ERP/BMS)
- REST API for seamless integration into Industry 4.0 and (I)IoT solutions

1.2 Structure of the **REXYGEN** system



1.3 Programming in the **REXYGEN** system

The REXYGEN system offers a graphical development environment for programming the algorithms. You can use standard desktop or laptop PC. You create the algorithms from the so-called function blocks. The library includes countless items (timers, comparators, filters, PID controllers and many more).



1.4 Main components of the **REXYGEN** system

1.4.1 **REXYGEN Studio** – the development environment

REXYGEN Studio is a developer's tool which runs on a standard PC with Windows. You create the algorithms using the function block library¹ of the **REXYGEN** system [1]. The

 $^{^1{\}rm The}$ IEC 61131-3 standard defines Function Block Diagram (FBD) as one of the PLC programming techniques.

library contains simple comparators and timers as well as advanced blocks for signal processing and feedback control (PID controllers etc.). You compile your project on your PC and run it on your UniPi Neuron.

Once running, you can watch your algorithm in real-time. Just select the signals and function blocks of your interest. You can also obtain detailed hierarchical information about the running algorithm and diagnose the runtime core and execution of your algorithm. You can connect via local network or over the Internet.

1.4.2 **REXYGEN HMI Designer**

The REXYGEN HMI Designer is another developer's tool intended for designing graphical user interface (or HMI, Human Machine Interface, if you prefer) for your algorithms. The user interface is included in the project and it is copied to your UniPi Neuron along with the algorithm.

1.4.3 **REXYGEN** Compiler

The REXYGEN Compiler converts your algorithms into binary code of the REXYGEN system. The compiler is almost invisible for the user, it is invoked from the REXYGEN Studio development environment. The compiler detects and reports possible errors in your algorithms.

1.4.4 RexCore

The RexCore (REXYGEN runtime core) service runs on the target device (UniPi Neuron). Once installed, it runs in the background and therefore it is almost invisible for the user. It handles timing and execution of your algorithms and provides various services. The individual tasks are prioritized and executed using preemptive multitasking. The RexCore service is automatically started at system startup.

RexCore further contains an integrated webserver providing user interface (HMI) and REST API. Chapter 3 deals with installing all the necessary modules on UniPi Neuron.

1.4.5 **REXYGEN** Diagnostics

With REXYGEN Diagnostics you can diagnose the runtime core and execution of your algorithm. It is an alternative standalone tool for commissioning and diagnostics of control algorithms. All the diagnostic functions are included in REXYGEN Studio, therefore REXYGEN Diagnostics can be considered obsolete. You can connect via local network or over the Internet.

Installation of REXYGEN development tools

This chapter describes the steps to install and uninstall the development tools of the REXYGEN system on Windows 7/8/10 operating systems. The tools also run on various GNU/Linux distributions using Wine¹.

2.1 Windows 7/8/10

The installation package of REXYGEN development tools contains the REXYGEN Studio and REXYGEN HMI Designer development environments, the REXYGEN Compiler compiler and the REXYGEN Diagnostics tool. It also includes the RexCore (REXYGEN runtime core) module for developing and testing purposes.

The installation package can be downloaded from

https://www.rexygen.com/software-download.

The installation process requires the administrator rights on your PC.

2.1.1 Installation procedure

- Run the REXYGEN-X.XX.XXX.PPP.exe downloaded from https://www.rexygen.com/software-download.
- 2. Select the language and follow the installation wizard.
- Select the target installation folder, the default is C:\Program Files\REX Controls\REXYGEN X.XX.XX.XXX.
- 4. Afterwards you can select the components to install. The requirements are quite low (approx 400 MB disk space) therefore the Full install option is recommended.
- 5. The following steps are standard and do not require further explanation.

¹http://www.winehq.org

There is no need to restart the system after installation.

2.1.2 Uninstall procedure

The common procedure can be used to uninstall REXYGEN development tools – go to *Control panel* and choose *Install/Uninstall programs*.

2.2 GNU/Linux

On various GNU/Linux distributions, you may install the development tools using Wine² 2.0 or later. Although we test the development tools in Wine regularly, please note that this is meant for experienced users and limited support is provided.

Follow the instructions for installation under Windows 7/8/10 given in section 2.1. The installation is started with wine REXYGEN-X.XX.XX.XXX-PPP.exe command entered at the console prompt. Some distributions (e.g. Ubuntu) allow running the installator by clicking the REXYGEN-X.XX.XXX.PPP.exe file.

 $^{^{2}}$ http://www.winehq.org

Installing the runtime modules of the REXYGEN system on UniPi Neuron

3.1 Install using operating system image with REXYGEN

IMPORTANT: This procedure will delete all old data and settings. Make sure you have all the necessary files stored elsewhere!

- It is highly recommended to use the official UniPi Neuron OS image with REXYGEN which you can download from http://rexygen.com/downloads/neuron-rexygen_image.zip.
- After you download the file, write it to a SD card with a capacity of 2 GB or more. You can use a Win32 Disk Imager for example: https://sourceforge.net/projects/win32diskimager/.
- 3. Put the SD card in and power up your UniPi Neuron.

3.2 Manual installation on running UniPi Neuron

It is also possible to install the RexCore (REXYGEN runtime core) on running UniPi Neuron manually. Following instructions guide you through installation of REXYGEN runtime on plain UniPi Neuron OpenSource OS. See documentation for other platforms for specific installation steps.

1. Install some necessary packages:

```
sudo apt update
sudo apt install wget gnupg2
```

2. Add **REXYGEN** public key:

wget -nc https://download.rexcontrols.com/files/key/rex-debian-key.pub sudo apt-key add rex-debian-key.pub

3. Add the repository to /etc/apt/sources.list or create a new *.list under /etc/apt/sources.list.d/ with the following content:

deb http://download.rexcontrols.com/repositories/debian-rex-2.50/ bullseye non-free deb http://download.rexcontrols.com/repositories/rex-dwm/debian bullseye non-free

Replace bullseye with buster or stretch if you are still running the older Debian distribution. Replace debian-rex-2.50 with debian-rex-2.50-testing or debian-rex-2.50-next to install development branches. The "testing" branch contains a REXYGEN version that is to be released soon, but still under testing. The "next" branch contains the current state of a next REXYGEN version under development. Use these options carefully.

4. Update package list:

sudo apt update

5. Install **REXYGEN** runtime and DWM:

```
sudo apt install -y --install-recommends rexygen-runtime
sudo apt install rex-dwm
```

3.3 Starting the runtime

Right after the installation, the RexCore runtime module is started automatically in the background as a daemon and it is possible to establish connection between the host PC and the UniPi Neuron using the REXYGEN Studio program. RexCore also automatically runs upon system (re)start.

3.4 Installed files and folders

Please refer to the RexCore user guide [2] for detailed information about the files and folders of the REXYGEN system runtime modules.

Licensing REXYGEN using REXYGEN DWM on UniPi Neuron

The first thing you will need is to get the license. The fastest way is to use the REXYGEN DWM tool. Open your web-browser and input IP address of UniPi Neuron. You should see the **Device is not licensed!** message on the bottom of the screen. Please, click on the Licensing.

	\\\/	Resources			
ē	Resources	Uptime	CI	טי	
0	System Info	Current time: 2020-09-16 14:39:09	Cores:	4	
÷	Network	Startup time: 2020-09-16 14:38:04	Model:	ARMv7 Processor rev 4 (v7I)	
:0:	Settings 🗸 🗸	Uptime: a minute	Hardware:	BCM2835	
	5		Serial:	0000000f740f3f0	
2	Service	RAM			
•	Security		Load average over 1 minute:	1.22	
-		Total: 905.06 MB	Load average over 5 minutes:	0.42	
	File Browser	Free: 777.11 MB	Load average over 15 minutes:	0.15	
Ē	Logs	Available: 785.09 MB	Processes (running/total):	1/112	
	Password				
462			Disks		
Ð	Archives	🗹 ext4 🗹 vfat	🗖 tmpfs	devtmpfs	
07	Licensing				
		Device: Device is not licensed! Label: Turce	mcblk0p2 acquire a valid licence.		

The licensing model of the REXYGEN system is quite simple:

• The development tools are free to use, you can install it on as many computers as you want.

• The RexCore runtime module always needs a licence to run on your UniPi Neuron. There are DEMO licences available at no cost and there are permanent licences which you can purchase. Each UniPi Neuron needs an individual licence.

4.1 Obtaining a DEMO licence

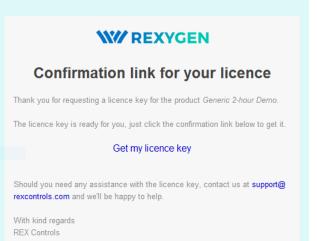
The DEMO licence is intended for evaluating, testing and educational purposes. Feel free to experiment with the DEMO licence as long as you need. Commercial use of the DEMO licence is not allowed.

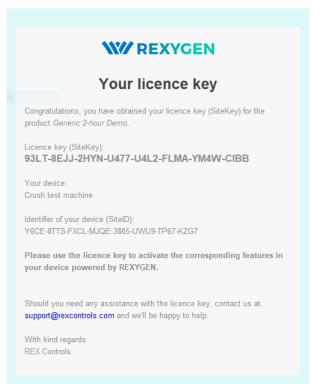
When you try to run your algorithm on a device which does not have a valid licence, you are offered a few options. Unless you already have a licence key or a licence voucher, you'll need to get a DEMO licence.

=	***	Licensing
	Resources	Active Licence
0	System Info	Site ID Y6CE-8TTS-FXCL-MJ0E-3865-UWU9-TP67-KZG7
(:-	Network	
:0:	Settings	, No active site keys available <u>Get demo licence</u>
٩	Service	
•	Security	Configuration
	File Browser	Apply Voucher Apply Licence Keys
Ē	Logs	Voucher
ð	Password	
Ð	Archives	Description for your reference
04	Licensing	APPLY VOUCHER

Click on <u>Get demo licence</u> and you will be redirected to the registration form. Identify yourself and you'll receive a DEMO licence key via e-mail. The licence key is called a **SiteKey**.

=	***	REXYGEN - Licensing	ĉ
		Demo licence	
	Use Voucher	If you have an account already, try to log in first.	
	Use Voucher DEMO Licence	Get demo licence SitelD (hardware identifier of your device)* Y6CE-8TTS-FXCL-MJQE-3865-UWU9-TP67-KZG7 Description (for your reference) crush test machine E-mail* johndoe@domain.com First name* John Last name*	-
		Doe	-
		V I'm not a robot	
			-





Once you have your DEMO licence key (SiteKey), apply it in REXYGEN DWMby selecting Apply Licence Keys and clicking on Apply keys.

=	\\	
Ę	Resources	Active Licence
0	System Info	Site ID BNI8-W33Y-64X6-93XJ-WDHQ-RMXG-NBK4-4J7G
((+	Network	DUT0-11-04Y0-2Y1-AUMU-04Y0-12Y1-04Y0-04Y0-04Y0-04Y0-04Y0-04Y0-04Y0-04
0	Settings ~	No active site keys available <u>Get demo licence</u>
٩	Service	
Ø	Security	Configuration
	File Browser	Apply Voucher Apply Licence Keys
Ē	Logs	Enter site keys 93LT - 8EJJ - 2HYN - U477 - U4L2 - FLMA - YM4W - CIBB
ô	Password	
Ð	Archives	
07	Licensing	
		<u>#</u>
		APPLY KEYS

Evaluation version of RexCore (REXYGEN runtime core) is functional for 2 hours. It is possible to run your algorithm on the UniPi Neuron but you cannot store it permanently.

The algorithm resides only in the RAM memory therefore it will not run after a restart or power-up. You can use almost all function blocks, see [1]. The RexCore runtime core on the target device is terminated after 2 hours of operation in demo mode without any warning. After a restart, you have another 2 hours for your experiments.

4.2 Obtaining a permanent licence

It is necessary to activate the RexCore runtime module and optional additional modules for permanent operation. This can be done using the licence, which you can obtain at

```
www.rexygen.com/pricing
```

4.2.1 Activation of the permanent licence

Each device running the RexCore runtime module is identified by the so-called SiteID tag. The purchased licence is associated with the hardware device, i.e. with the SiteID tag.

1. Open your favorite web-browser on the desktop PC and fill in the IP address of UniPi Neuron. Once the DWM appears, select *Licensing* \rightarrow *Apply Voucher* and enter your licence voucher code.

=	***	Licensing
Ę	Resources	Active Licence
0	System Info	Site ID Y6CE-8TTS-FXCL-MJQE-3865-UWU9-TP67-KZG7
(0	Network	
٠	Settings	, Site Keys
٩	Service	93LT-8EJJ-2HYN-U477-U4L2-FLMA-YM4W-CIBB
0	Security	Target is running in DEMO mode.
	File Browser	
Ē	Logs	Configuration
8	Password	Apply Voucher Apply Licence Keys
0	Archives	Voccher V2Q-ZE1-2UY
07	Licensing	Description crush test machine
		for your reference APPLY VOUCHER

2. If other licenses are already present, a new window will appear where you can specify what should happen to the old licenses. Use the *Overwrite* option if the UniPi Neuron is running in DEMO mode. This is the last and irreversible step.

≡	W	Licensing 🔚
	Resources System Info Network Settings Service Security File Browser Logs Password	Active Licence Site ID Y6CE-8TTS-FXCL-MJQE-3865-UWU9-TP67-KZG7
-5) 07	Archives	Description crush test machine
	g	for your reference APPLY VOUCHER

3. The so-called SiteKey licence key is generated from your licence voucher and it is stored in your device, which is shown in the *Active License* panel. The licence key allows permanent operation of the runtime core. It is highly recommended to check the licence again after a reboot of the device.

≡	***	Licens	ing ∷≡
	Resources		Active Licence
0	System Info		Site ID
(0	Network	Y6CE-81	TS-FXCL-MJQE-3865-UWU9-TP67-KZG7
:0:	Settings	~	Site Keys
٩	Service	FYLN-P5	39-6F4Z-R6UI-7YI2-GSJE-5TT9-DDEN
Ø	Security		\65-HWC3-TAKY-K2CH-C7ZR-CULU-865T \Y3-N5ZS-GBE6-9PPF-AZ3P-PZ2P-SHE7
-	File Browser		IB3-CF5R-HGUW-8GMS-P5LR-KAGW-329C
Ē	Logs		DQ-PDGD-MHBK-F6N6-MEP5-4DZK-UNXZ 5JD-WNMN-8QGQ-GMYZ-BJLJ-NS9N-5EWU
8	Password		
Ð	Archives		Configuration
04	Licensing	Apply Voucher Apply Licence Keys	
		Voucher V2Q-ZE1-2UY	
		Description crush test machine	
		for your reference	APPLY VOUCHER

Default REXYGEN project on UniPi Neuron

In order to help you speed up a start with REXYGEN and UniPi Neuron we have create an example project for all UniPi Neuron which is already preinstalled on the image with REXYGEN (installation is described in Section 3.1). The project serves as a basic validation of the installation process and as a starting point for more complex applications. One can control digital output(s) DO, analog output(s) AO, observe the state of digital inputs(s) DI and analog input(s) AI. These general inputs and outputs are present on all UniPi Neuron models. The steps to create and develop a new project from scratch is described in Chapter 6 in detail.

IMPORTANT: If you have valid permanent licence stored on your UniPi Neuron, this project will start automatically once RexCore (REXYGEN runtime core) starts. Otherwise, if you have only a Demo license, you must compile and download the project from the REXYGEN Studio on the UniPi Neuron first. The project can be opened in the REXYGEN Studio using either the procedure described in Section 5.3, or the project can also be found in the Examples library as 0126-02_UniPi_Neuron_Generic¹. The process of compiling and running the project is described in Section 6.2.

5.1 Control UniPi Neuron from HMI

The project contains graphical user interface (so-called Human-Machine-Interface) for easy control and observing of general outputs and inputs. The HMI can be found at URL address http://192.168.1.100:8008/hmi/index.html (replace 192.168.1.100 with the IP address of your UniPi Neuron).

¹All examples which are part of the installation are marked by an ExampleID. The most up-to-date examples are available with the latest installation of development tools or at https://www.rexygen.com/example-projects/



5.2 Control UniPi Neuron through OPC UA

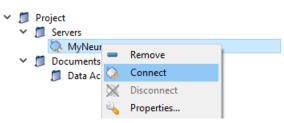
You can use the OPC UA interface to control your UniPi Neuron too. Install any OPC UA Client and connect to the OPC UA Server, which is running right now on your UniPi Neuron. There is the *UAExport* software used in this tutorial:

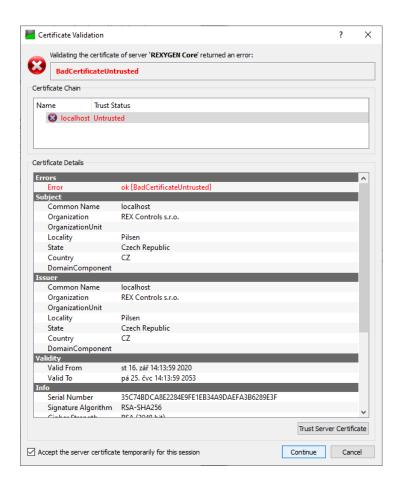
https://www.unified-automation.com/downloads/opc-ua-clients.html.

Click on Server in the Menu $\rightarrow Add \rightarrow Advanced mode$ and fill the Configuration Name and the Endpoint URL (replace 192.168.1.100 with the IP address of your UniPi Neuron).

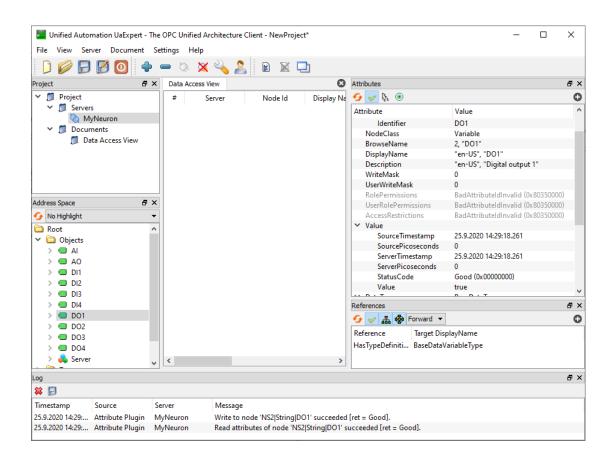
Add Server	P
Discovery Advanced	.
Server Information	
Endpoint Url	opc.tcp://192.168.1.100:4840/
Reverse Connect	
Security Settings	
Security Policy	None 🔻
Message Security Mode	None 👻
Authentication Settings	
Anonymous	
Username	Store
Password	
Certificate	
O Private Key	
Session Settings	
Session Name	
Connect Automatically	

After confirmation (OK), right-click on the new server name in the *Project* panel and choose *Connect*.





Congratulations, you are now connected to your UniPi Neuron! You can now set the DO1 to value *true* for example. Navigate to *Address Space* panel and select DO1. In *Attributes* panel search *Value* \rightarrow *Value*, double-click on it, check the radio box and hit Enter. The first Digital outputs LED diod on UniPi Neuron should glow. Detailed tutorial on using OPC UA with REXYGEN is there [3].



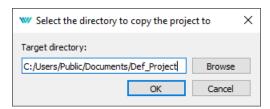
5.3 Loading project from UniPi Neuron to REXYGEN Studio

After installation of REXYGEN development tools (described in Chapter 2), you can open the REXYGEN Studio on your PC and download algorithm of this default project from your UniPi Neuron.

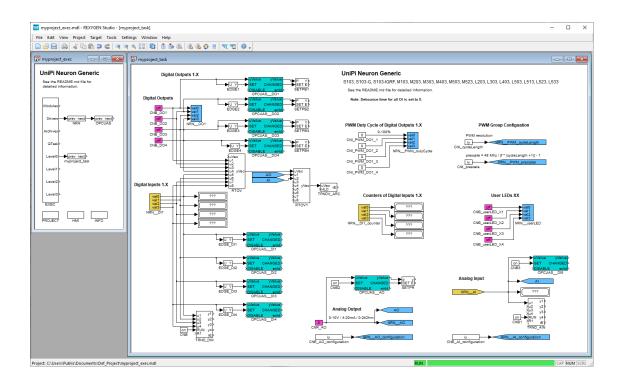
Select Load Project from Target Device from starting dialog or from $File \rightarrow Open$ Project from Target Device and connect to your UniPi Neuron (replace 192.168.1.100 with the IP address of your UniPi Neuron).

	W How would you like to start?					×	
		Thank yo	ou for	choosing REXYGE	EN!		
	REXYGEN			Start with a Pla	in Proje	ect	
			Start from an Example Project Open an Existing Project				
			Lo	ad Project from 1	Farget D	Device	
	Don't s	how this dia	log at	start	Clos	se]
W C	onnect to the	target			_		×
C	onnection Detai	s					
Та	arget:	192.168.1	. 100:4	43981			
Pr	rotocol:	TCP/IP		•			
U	ser:	admin					
Pa	assword:						۲
Si	ave Password:					T 🐝 🗌	est
	escription			Add station	to list	🛄 Tog	gle list

Choose the directory where will be the project stored on your PC and hit OK.



If everything went well, the project algorithm will appear.



Configuration, compilation and execution

The process of creating a control algorithm will be demonstrated on a very simple example with four Boolean variables representing manual switches. In Chapter 7.3, two of them will be replaced by physical inputs of the UniPi Neuron. A software timer will be used for measuring the time when the variables are true (i.e. the switches are in the ON position). A Boolean signal will indicate that the interval of predefined length has elapsed.

6.1 Creating a new project

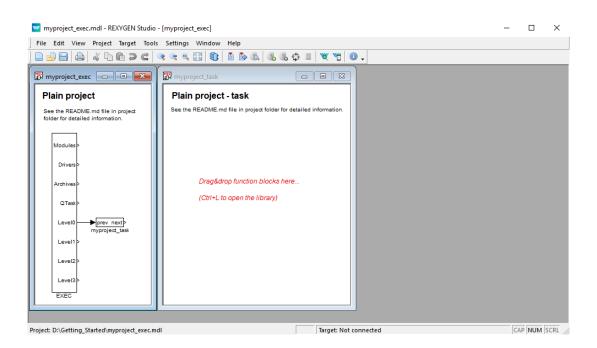
The project configuration is created using the REXYGEN Studio program. Each project consists of at least two .mdl files. The first file is the main file of the project, which is used for configuration of tasks, drivers, priorities and timing. The other file(s) contain the individual control algorithms (tasks).

First we'll create the example 0101-01 from scratch¹. Standard approach:

- 1. Run the REXYGEN Studio program. Start with a plain project and select a folder to save the project files in (e.g. D:\GettingStarted).
- 2. The folder will contain two important files:
 - myproject_exec.mdl
 - myproject_task.mdl

¹All examples which are part of the installation are marked by an ExampleID. The most up-to-date examples are available with the latest installation of development tools or at https://www.rexygen.com/example-projects/

3. The myproject_exec.mdl is the project main file. It contains one EXEC block from the EXEC library. The other block is the TASK block from the same library and it is renamed to myproject_task to reference the second file of the project (myproject_task.mdl), which will contain the algorithm (the so-called task).



- 4. The task is connected to the LevelO output of the EXEC block and therefore its timing is defined by tick and ntickO parameters of the EXEC block.
- 5. The EXEC block (and any other block) can be configured by double-clicking on it. A block parameters and properties dialog appears. The parameters of all blocks of the REXYGEN system are described in the Help (press the F1 key) and in the Block reference manual [1].

	name:		Block type:	
EC			execlib \EXEC	
ck t	ype description:			
eal-t	time executive			
en b	olock documentation	<u></u>		Toggle quick reference
	eters Runtime	Style		
		Style		
cala	ar parameters			
	Parameter	Value	Туре	Description
1	target	Generic target device 💌	String	Target device
2	tick	0.05	Double	Base tick (period) of the r
3	ntick0	2	Long	Period of tasks in Level0 (
4	ntick1	50	Long	Period of tasks in Level1 (
5	ntick2	100	Long	Period of tasks in Level2 (
6	ntick3	1200	Long	Period of tasks in Level3 (
7	pri0	5	Long	Priority of tasks in Level0
	pri1	9	Long	Priority of tasks in Level1
8		13	Long	Priority of tasks in Level2
8 9	pri2	15		

- 6. Note that tick=0.05 and ntick0=2, therefore the task will run each 100 milliseconds $(0.05 \cdot 2 = 0.1s)$. There is no need to change any parameter at the moment. Close the dialog.
- 7. You can delete all the descriptive texts in project files. These have no effect on the functionality and can be considered programmer's comments.
- 8. Open Block Library, choose *View/Block Library* in the menu or use the **1** icon from the toolbar.
- 9. By default, the library is in *Tree view* mode where the blocks are organized in sublibraries. Their location is always denoted as sublibrary/block, e.g. LOGIC/AND for the logical AND block in the LOGIC sublibrary. Inside a sublibrary, the blocks are ordered in alphabetical order.

ile Edit View Project Target Tools				
Y	 Market and the second se			- • ×
See the README.md file in project folder for detailed information.	See the REA		Quick refe	rence
Modules>	Tree view	O List view		Boolean (logic) constant
Drivers >	Block name > GEN > INOUT > LOGIC	Description		on≯ bool CNB
QTask>		Absolute value Addition of two signals Addition of sixteen signals	Outp	Logical output of the block
myproject_task Level1> Level2>		Addition of eight signals Addition of four signals Boolean (logic) constant Enumeration constant Integer constant Real constant	Para YCN	ameters Boolean constant
EXEC	DIF_ DIV	Difference Division of two signals	•	Open block documentation

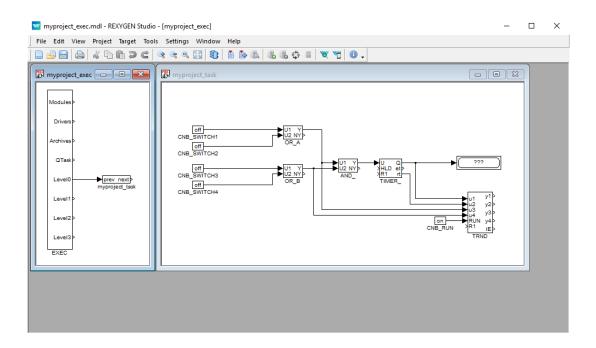
- 10. You can also switch the library to *List view* mode, where all the blocks are sorted alphabetically, regardless of the sublibrary they belong to.
- 11. Locate the following function blocks in the Block Library and drag them to the task file:
 - MATH/CNB constant of type Boolean. Once dropped, double-click its name and change it to CNB_SWITCH1. Double-click the block and set parameter YCN = off.
 - LOGIC/OR_ logical OR. Rename it to OR_A.
 - LOGIC/AND_ logical AND.
 - LOGIC/TIMER_ a timer block. Set parameter mode = 2: Delayed ON, pt = 5.0.
 - INOUT/Display a display to show values in real-time.
 - ARC/TRND real-time recording. Set parameters l = 2000, Title = My timer, leave the default values otherwise.
 - MATH/CNB constant of type Boolean, change name to CNB_RUN, set parameter YCN = on.

111	myproject_exec.r	ndl - REXYGEN	Studio	- [myproje	ect_exec]						-	×
-	ile Edit View											
		4 6 6 2	C	e e e		ă 🗟 🗟	₲₲\$ =	V VI 🔍 🗸				
	🔋 myproject_exec		×	😨 mypi	oject_task						• X	
	Modules >											
	Drivers >			CN	off) S_SWITCH1		XU1 Y> XU2 NY>					
	Archives>						OR_A					
	QTask	•						AND	XU QA XHLD eta XR1 rta	>(???		
Ш	Level0	prev_next myproject_task						_	TIMER_	yu1 y1	<i>,</i>	
	Level2>									u2 y2 u3 y3	>	
	Level3 >									CNB_RUN X4 CNB_RUN XR1 iE TRND	>	
	EXEC									Hub		

- 12. Duplicate the CNB_SWITCH1 block with right mouse button dragging. Or simply Copy&Paste the block.
- 13. Duplicate the block 2 more times.
- 14. Duplicate also the OR_A block. Rename the duplicate to OR_B.

w myproject_exec.mdl - REXYGEN Studio	o - [myproject_exec]			-	×
File Edit View Project Target Too		⊾ ⊪ ⊪ ⇔ ≡ ∖	¥ \\ 10 _		
Image: Transmission Image: Transmission Imag				277 1 91 2 92 1 91 2 92 1 93 1 12 TRND	

15. Connect the blocks as shown below. To connect the blocks, drag the output arrow of one block to the input arrow of the other block using the left mouse button. The connection will be established when the line goes bold and green. After releasing the mouse button you can recognize a successfully connected line by its style. A full line terminated by a full arrow at the input of the connected block indicates a valid connection. A dashed red line ending with a thin arrow indicates an unconnected line. New branch of an existing line can be created by dragging an existing line with the right mouse button.

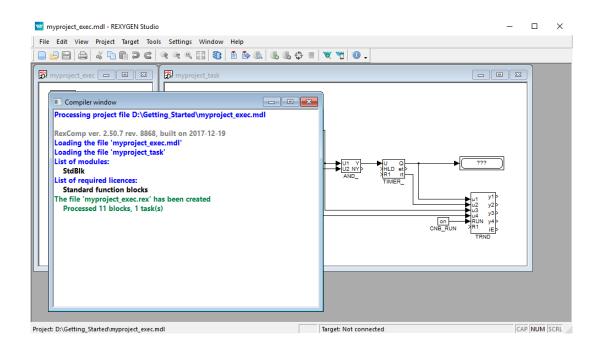


At this moment the executive configuration myproject_exec.mdl and the corresponding myproject_task.mdl file with the algorithm are ready. The algorithm will be evaluated in the direction of the arrows, starting from the source CNB blocks, passing through the OR_, AND_ and TIMER_ blocks and finishing at the Display and TRND blocks.

Congratulations, your first project is ready for compilation!

6.2 Compiling and running a project

The developed algorithm must be compiled to binary form prior to deploying. Pick *Project/Compile* from the menu or use the icon from the toolbar. The compiler output is displayed in the Compiler window. If no error is found, the myproject_exec.rex file is created.



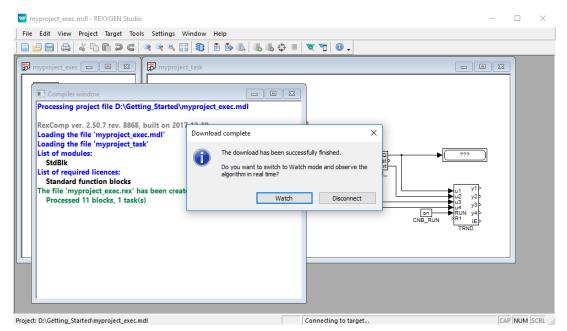
At this moment it is possible to deploy the control algorithm to the target platform. Use *Project/Compile and Download* in the menu or click the Compile and Download icon for this purpose. A dialog for defining the target device appears upon successful compilation.

Enter the IP address of your UniPi Neuron in the Target field.

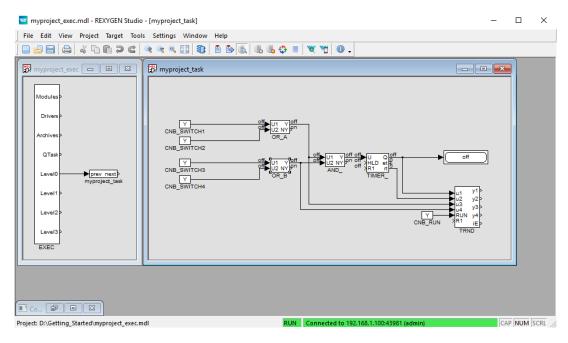
Enter REXYGEN login and password. The default user is admin and there is no password by default. Leave the other elements intact and click Download.

If there is no licence on your UniPi Neuron, you need to get one first. You can get a DEMO licence for free. See Appendix B for details and come back afterwards.

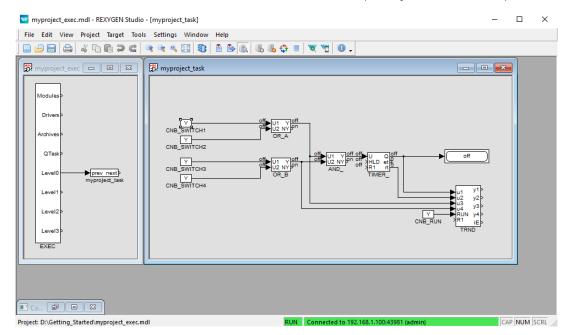
As soon as the download is complete it is possible to switch the REXYGEN Studio application to the so-called *Watch mode* and watch the control algorithm in real-time – click Watch.



In the *Watch mode*, the background of all files goes gray and you cannot move or delete any blocks or connections. Right-click the TIMER_ block and select *Watch selection* in the menu to watch the inputs and outputs of the timer.



You can do the same with the OR_ and AND_ blocks (or any other selection).

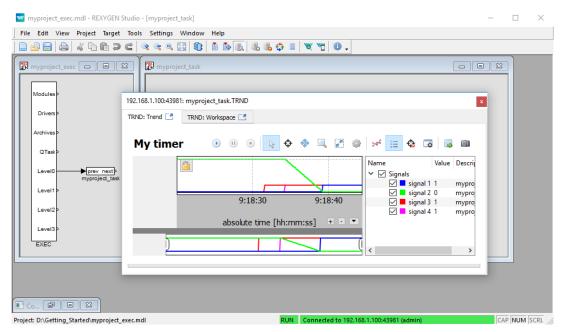


Now it is possible to double-click the CNB_SWITCH1 block and change the Boolean variable to YCN = on (tick the checkbox and click OK). Once you do the same with the CNB_SWITCH3 block, the outputs of both OR blocks are on and the Y output of the AND_ block goes on and the TIMER_ starts to count down. Observe the rt output.² Once the

 $^{^{2}}$ Do not get confused by the default 1 second refresh rate of the *Watch mode*. The algorithm on the target device runs each 100 milliseconds as mentioned earlier.

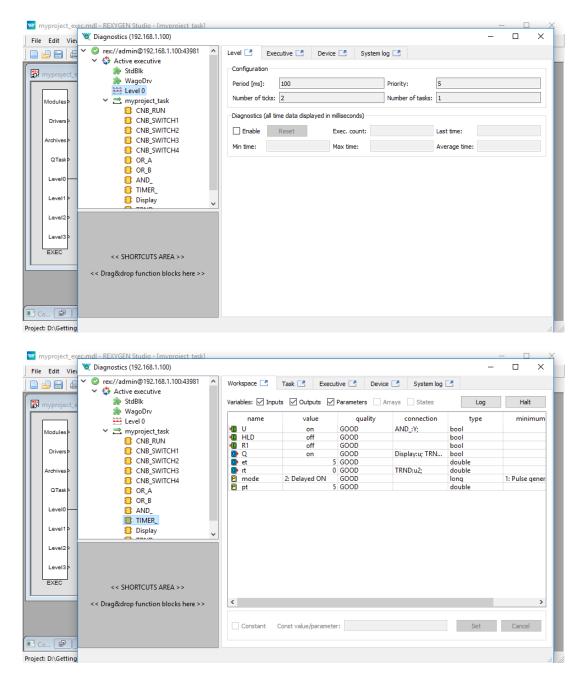
timer reaches zero, its output Q is set to on and it remains on as long as the U input is on.

You can double-click the TRND block to see the signals in a real-time graph. The red line is the output of the OR_A block, the magenta line is the output of the OR_B block, the green line is the remaining time of the timer and the blue line is the Boolean output of the timer.



Try turning the CNB blocks off and change the pt parameter of the TIMER_ block. Afterwards turn the CNB blocks on again and observe the signals in the TRND block again. As you can see, you can modify any parameter in real-time, which allows you to fine tune your algorithm.

It is also possible to open a Diagnostics view of the algorithm. Pick Target/Diagnostics from the menu or click the 💝 icon and you will see the algorithm in a tree view which allows you to monitor the control algorithm in full detail. You can verify that the sampling rate of your algorithm is indeed 100 milliseconds. You can also adjust parameters of individual function blocks, which has the same effect as modifying them directly in the *Block properties* dialog.



Note: There is also a standalone diagnostics program called REXYGEN Diagnostics. Click the REXYGEN Diagnostics icon ^{WC} and confirm the IP address of the target platform.

Now you can switch REXYGEN Studio back to the *Development mode*. You can do so by deactivating the *Watch mode* (use the \blacksquare icon). You are offered synchronization of the changed parameters with the source files of the project, choose No at this moment.

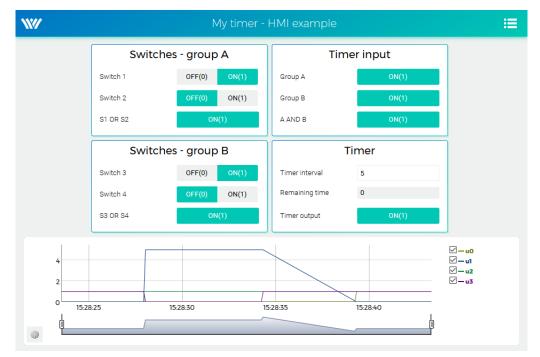
All changes made while in the *Watch mode* are not permanently stored in the target device (unless you decide so, see [4]). Upon restarting the RexCore runtime module the algorithm will start with the parameters defined in the project source files, which were valid when compiling and downloading the algorithm to the target device. To apply the changes permanently, you have to transfer the changes to the source files and *Compile and download* the project one more time which defines new startup values.

6.3 Adding a user interface (HMI)

The next step in developing a control algorithm is its user interface, or HMI, Human-Machine-Interface. It allows anyone (even those who are not familiar with the REXYGEN system) to interact with the algorithm. The HMI of the REXYGEN system relies on modern web-based technology and the HMI is therefore accessible via web browser on desktop PC, tablet or smartphone.

In this tutorial, a simple HMI will be created using the so-called WebBuDi technology. It provides very simple indicators and input elements to interact with the control algorithm via a web page (Web Buttons and Displays).

The steps to create the HMI are described below. This is how the HMI will look like in the end:

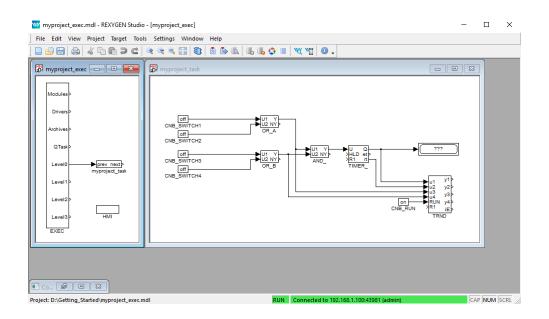


1. In the folder with the project files, create a hmisrc subfolder. Inside this folder, create a file named index.hmi.js and edit it with your favorite text editor. The content should be the following:

```
REX.HMI.init = function(){
//Indicators and virtual switches - group A
  var switchesA = {
    column: 1,
    title: 'Switches - group A',
    rows: [
      {type: 'DW', alias: 'switch1', desc: 'Switch 1',
        cstring: 'myproject_task.CNB_SWITCH1:YCN'},
      {type: 'DW', alias: 'switch2', desc: 'Switch 2',
        cstring: 'myproject_task.CNB_SWITCH2:YCN'},
      {type: 'DR', alias: 'S1orS2', desc: 'S1 OR S2',
        cstring: 'myproject_task.OR_A:Y'},
    ]
  };
  REX.WebBuDi.addSection(switchesA);
//Indicators and virtual switches - group A
  var switchesB = {
    column: 1,
    title: 'Switches - group B',
    rows: [
      {type: 'DW', alias: 'switch3', desc: 'Switch 3',
        cstring: 'myproject_task.CNB_SWITCH3:YCN'},
      {type: 'DW', alias: 'switch4', desc: 'Switch 4',
        cstring: 'myproject_task.CNB_SWITCH4:YCN'},
      {type: 'DR', alias: 'S3orS4', desc: 'S3 OR S4',
        cstring: 'myproject_task.OR_B:Y'},
    ]
  };
  REX.WebBuDi.addSection(switchesB);
//Timer input
  var timerInput = {
    column: 2,
    title: 'Timer input',
    rows: [
      {type: 'DR', alias: 'inputA', desc: 'Group A',
        cstring: 'myproject_task.AND_:U1'},
      {type: 'DR', alias: 'inputB', desc: 'Group B',
        cstring: 'myproject_task.AND_:U2'},
      {type: 'DR', alias: 'AandB', desc: 'A AND B',
        cstring: 'myproject_task.AND_:Y'},
    ]
```

```
};
  REX.WebBuDi.addSection(timerInput);
//Timer settings and status
  var timer = {
    column: 2,
    title: 'Timer',
    rows: [
      {type: 'AW', alias: 'interval', desc: 'Timer interval',
        cstring: 'myproject_task.TIMER_:pt'},
      {type: 'AR', alias: 'rt', desc: 'Remaining time',
        cstring: 'myproject_task.TIMER_:rt'},
      {type: 'DR', alias: 'timerQ', desc: 'Timer output',
        cstring: 'myproject_task.TIMER_:Q'},
    ]
  };
  REX.WebBuDi.addSection(timer);
  //Add real-time trend
  REX.HMI.Graph.addTrend({cstring: 'myproject_task.TRND'});
  REX.HMI.Graph.setMaxBufferSize(200);
  // Change title of the page
  REX.HMI.setTitle('My timer - HMI example');
}
```

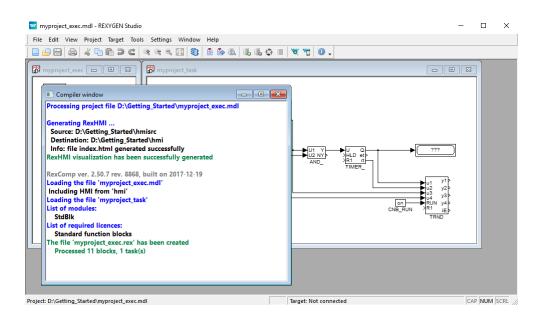
2. This file will be processed when compiling the project. However, it is necessary to add the EXEC/HMI block into the project main file first.



3. Double-click the HMI block to edit its parameters. Set GenerateWebWatch = off and confirm. WebWatch is another type of HMI, which you do not need at the moment. See [5] for details, it is a very interesting tool for developers and technicians.

k	name:		Block type:	Block type:		
I			execlib\HMI			
k	type description:					
11	configuration					
en	block documentation	<u>1</u>	<u> </u>	Toggle quick reference		
	meters Runtime					
		Style				
ca	lar parameters					
	Parameter	Value	Туре	Description		
1	IncludeHMI	🗹 on	Bool	Include HMI files in the pr		
2	HmiDir	hmi	String	Output folder for HMI files		
3	SourceDir	hmisrc	String	Source directory		
4	GenerateWebW	off	Bool	Generate WebWatch HMI		
5	GenerateRexHMI	🗹 on	Bool	Build HMI from SVG and J		
6	RedirectToHMI	🗹 on	Bool	Webserver will automatica		
7	Compression	off	Bool	Enable data compression		

4. The HMI is now an integral part of your project. Compile the project again and you will see that the compile log contains more information. The index.html file is generated from the source index.hmi.js file. All the HMI files are generated into the hmi subfolder (HTML, JS and CSS files) and included in the resulting binary myproject_exec.rex file.



5. After you download the project to the target device, you can access the HMI via web browser. Go to menu *Target/Web Interface* which will open the webpage. Remember the default login credentials: admin with no password.

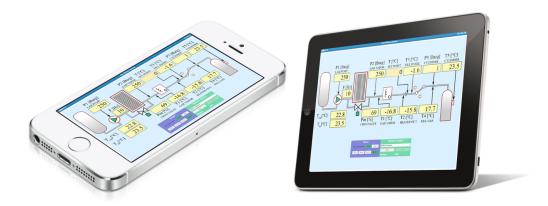
	<u>ଜ</u> ୁ ପ୍ର	2.168.1.100:8008/hmi/index.htm			⊻ ॥/ ₪
*//		My timer	- HMI example		
	Swite	ches - group A	Tim	ner input	
	Switch 1	OFF(0) ON(1)	Group A	ON(1)	
	Switch 2	OFF(0) ON(1)	Group B	ON(1)	
	S1 OR S2	ON(1)	A AND B	ON(1)	
	Swite	ches - group B	-	Timer	
	Switch 3	OFF(0) ON(1)	Timer interval	5	
	Switch 4	OFF(0) ON(1)	Remaining time	0	
	S3 OR S4	ON(1)	Timer output	ON(1)	
					□ – u0
4					⊻ — u1 ⊻ — u2
2					∠ – u3
0	15:22:55	15:23	15:23:05	15:23:10	

6. You can toggle the switches and observe the results. The virtual switches are linked

to the individual CNB function blocks therefore the effect is the same as toggling the values directly in REXYGEN Studio.

- 7. You can also change the timer setting and shorten or lengthen the interval.
- 8. See [5] for detailed information about WebBuDi elements and possible customization (colors, backgrounds etc.).
- The default URL address is http://192.168.1.100:8008/hmi/index.html (replace 192.168.1.100 with the IP address of your UniPi Neuron).
- 10. The default port of the webserver (8008) can be changed in RexCore settings. See [2] for details.

Please note that there is also a program called REXYGEN HMI Designer, which allows you to create graphical user interfaces. Appendix A of this guide shows the steps to create a basic graphical interface. A more complex HMI is shown below solely for inspiration purposes.



6.4 Ready for interaction with the outer world

Well done! You have created the example 0101-01 from scratch³. You have learned the basic workflow for developing and running your algorithms using the REXYGEN system, which is the same for all platforms. Now it's time to add the so-called *input-output drivers* so that the algorithm can interact with sensors, actuators and external data.

³All examples which are part of the installation are marked by an ExampleID. The most up-to-date examples are available with the latest installation of development tools or at https://www.rexygen.com/example-projects/

Chapter 7

I/O configuration for the UniPi Neuron

The previous chapter illustrated the process of creating a control algorithm in the REXY-GEN system and deploying it to the target device. But so far, the algorithm does not interact with the outer world, it is not connected to any physical signal (or external data).

We will use the input and output signals of the UniPi Neuron for interaction with the real world.

As mentioned in Chapter 6, two physical switches will be connected as inputs and a software timer will control one output signal. In other words, this chapter is devoted to advancing from the generic example 0101-01 to example 0126-01 for the UniPi Neuron.

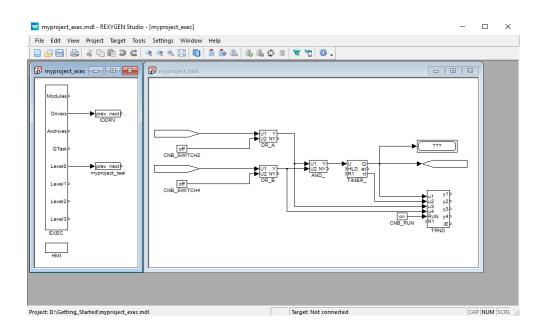
7.1 Physical connections

Connect the switches as shown below.

7.2 Adding inputs and outputs to the project

Now when we have everything wired up, it's time to include the physical signals into the algorithm. You need to expand your project main file with 2 additional function blocks to access the inputs and outputs from the control algorithm in your project. Insert the EXEC/MODULE and EXEC/IODRV blocks from the *Block library* and attach them to the EXEC block as shown below.

In the task file, delete the CNB_SWITCH1 and CNB_SWITCH3 blocks and replace them with INOUT/From blocks. These will be the input signals. Also add one INOUT/Goto block, which will serve as an output and which will be controlled by the timer. You already know that a new branch of a line is created by right-button dragging, don't you?



7.3 Working with I/O signals

7.3.1 Modifications in the project main file

Now we tell the compiler to use the UniPi Neuron I/O driver. To do this, edit the IODRV block parameters. Communication will be mediated using Modbus TCP. Link the IODRV block with the MBDrv (Modbus) module by setting

• module = MBDrv (beware, it's CASE SENSITIVE!)

Continue editing the IODRV block parameters. For UniPi Neuron set the following:

- classname = MtmDrv (beware, it's CASE SENSITIVE!)
- cfgname = neuron_cfg.rio
- factor = 1
- Leave the other parameters intact.

	name:		Block t	ype:
NRN			exedi	b\IODRV
	type description:			
REX	GEN input/output	driver		
ipen	block documentation	<u>on</u>		Doggle quick reference
Sca	lar parameters	Value	Туре	Description
1	module	MbDry	string	Module name
2		MtmDrv	string	I/O driver class name
	cfgname	neuron_cfg.rio	string	Configuration file name
4	factor	1	long	Execution factor
5	stack	10240	long	Stack size [bytes]
6	pri	3	long	Driver thread logical priority
7	cpu	-1	long	CPU core assigned to driver
8	timer	off	bool	Driver is a source of time

As the final step, rename the $\tt IODRV$ block to NRN, which will serve as a prefix for all I/O signals of this driver.

You have to create the $neuron_cfg.rio$ file. Click on *Configure* button and fill the form as shown bellow. After that, click on OK and the file will be stored in the project folder.

	ponse tin ry time [s	meout [s]: s]:	1.00 ÷				(×	lod	ous	
lav	es										
	N	lame		A	ddress			Port	Max. requests	Subaddress	
1	slav	veMB		12	7.0.0.1			502	16	0	4
											1
											J
											1 J
-		xa (for Iter Slave	n Addresse: Item Address	s and In Type)		de	Funct	ion code	
	Show He Name		ltem					de		ion code single coil	_
1	Show He Name DO1_1	Slave	ltem Address	Туре	Count		Flags	de	5 - write		
1	Name DO1_1 DI1_1	Slave slaveMB slaveMB	ltem Address 0	Type COIL	Count 1		Flags W	de	5 - write 1 - re	single coil	
1	Name DO1_1 DI1_1	Slave slaveMB slaveMB	ltem Address 0 4	Type COIL COIL	Count 1 1		Flags W R	de	5 - write 1 - re	single coil ad coils	

The executive of the $\mathsf{REXYGEN}$ system is configured, your project should look like this:

myproject_exec	• 💌
Getting started with REX	GEN
See the README.md file for detailed information.	
Modules >	
Drivers prev next	
Archives >	
QTask>	
Level0 prev_next myproject_task	
Level1 >	
Level2>	
Level3>	
EXEC	
HMI	

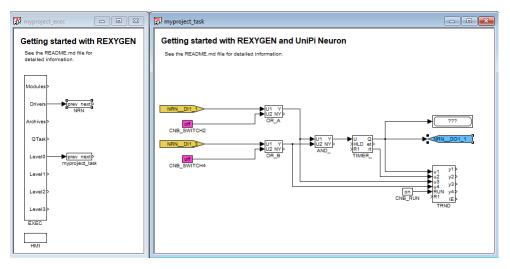
7.3.2 Modifications in the task

In the task double-click the input flag and set $GotoTag = NRN_DI1_1$. Note the NRN prefix and two underscore characters. The first physical switch (SW1) is connected to first group of Digital Inputs - input 1.

Block				
Block name:		Block type	:	
From		From		
Block type description:				
Signal connection or input				
Open block documentation			Toggle quick reference	
Parameters Options	Style			
Scalar parameters				
Parameter	Value	Туре	Description	 ٦
1 GotoTag	NRN_DI1_1 🧔	string	Goto tag	

The other physical switch (SW3) is connected to Digital Input 1.3 (NRN_DI1_3). The last (Digital Output) flag will be set as GotoTag = NRN_DO1_1 Your preject should now look like this:

Your project should now look like this:



After compiling the project and downloading it to the UniPi Neuron the control algorithm interacts with the physical world. Again it is possible to switch to Watch mode and observe the signals in real-time or analyze the trends of signals. Flip the physical switches and watch the signals.

7.4 Updating the HMI

It is also necessary to update the HMI. The CNB_SWITCH1 and CNB_SWITCH3 blocks are no longer present in the algorithm. Moreover, we need to replace virtual input elements (DW) with indicators (DR). Therefore open the index.hmi.js file and replace

```
{type: 'DW', alias: 'switch1', desc: 'Switch 1',
    cstring: 'myproject_task.CNB_SWITCH1:YCN'},
```

with

```
{type: 'DR', alias: 'switch1', desc: 'Switch 1',
    cstring: 'myproject_task.OR_A:U1'},
Similarly for switch no. 3, replace
```

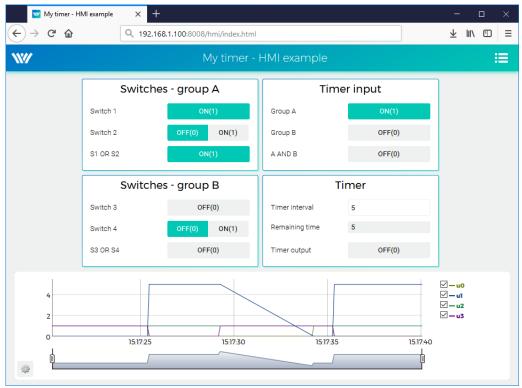
{type: 'DW', alias: 'switch3', desc: 'Switch 3',

```
cstring: 'myproject_task.CNB_SWITCH3:YCN'},
```

with

```
{type: 'DR', alias: 'switch3', desc: 'Switch 3',
    cstring: 'myproject_task.OR_B:U1'},
```

Save the file, compile and download the project again and open the web interface. Push the two physical switches and wait until the timer triggers the output. Alternatively, you can still use the virtual switches. This demonstrates that you can combine physical and virtual input elements.



7.5 Additional information

7.5.1 Detailed description of the driver

A detailed description of the Modbus TCP IO driver which serves as communication platform for UniPi Neuron is available in a separate manual [6].

7.5.2 Examples

Example projects and a set of all supported I/O flags are included in the installation package of the REXYGEN system development tools. In REXYGEN Studio, go to menu File \rightarrow Start from an Example Project and select one of the UniPi Neuron examples. As mentioned earlier, the most up-to-date information about examples are available at

https://www.rexygen.com/example-projects/

Chapter 8

Summary

Congratulations, you have created the example from scratch! You have learned how to develop, compile and run your algorithms on the UniPi Neuron. The interaction with sensors and actuators is provided via input-output driver of the REXYGEN system, which you have learned to configure and use.

You have achieved quite a lot in a relatively short time, haven't you? The purpose of this guide was to quickly show you the basic steps and tools for developing a project.

Now it's time to focus on your own project and keep learning on the go. There are function blocks which are much more powerful than the ones mentioned in this guide, there are many inspiring example projects, there are additional I/O drivers which you can use to expand the scope of your project, there are many ways to exchange data with external systems and devices, etc.

Remember that whenever you have some achievement to share, we will be happy to hear from you. And whenever you encounter any difficulties, we will be happy to help. You can reach us at support@rexygen.com.

Appendix A

Graphical HMI with REXYGEN HMI Designer

Chapter 6 describes the creation of a simple WebBuDi user interface. The REXYGEN development tools also contain the REXYGEN HMI Designer program which is a tool for designing custom graphical visualizations from predefined components¹. The REXYGEN HMI Designer is based on the well-known open-source vector editor InkscapeTM https://inkscape.org/en/.

In this chapter we'll develop an alternative HMI for the example 0101-01. Just like in the case of the WebBuDi user interface, the SVG file created with REXYGEN HMI Designer will serve as a source file which will become a part of your REXYGEN project. During compilation of the project the SVG file will be processed and converted to HTML, JS and CSS files.

A.1 Initializing the HMI design

After launching the REXYGEN HMI Designer from *Start Menu* you will find a clean page. The first thing to do is to initialize the new visualization. The HMI is configured via the *REXYGEN HMI* extensions. Navigate to *Extensions* \rightarrow *REXYGEN HMI* \rightarrow *Edit HMI Config* in the menu.

¹Definition of custom components is also possible but it requires a bit of Javascript coding.

New document 1 - RexHMI Designer		
<u>File Edit View Layer Object Path</u>	<u>T</u> ext Filters Extensions <u>H</u> elp	
	Previous Extension Previous Extension Settings	Alt+Q . Shift+Alt+Q . Shift+Alt+Q
N	REXYGEN HMI	Browse Components Library Ctrl+L
F	Arrange	 Edit Component Ctrl+E
R WW RI	EXYGEN Colour	 Edit HMI Config
FINI	Document	Export HMI Ctrl+H
	Export	 Group Animation Ctrl+.
E	Gcodetools	Update Components
	Generate from Path	Version Info
	Images	
	JessyInk Modify Path	
	Raster	
\bigcirc	Render	
€ <u>}</u>	Stylesheet	•
	Text	
0	Typography	
20	Visualise Path	
	Web	>
<u>T</u>		
V		
>>		» »
<		4
Fill: N/A O: 0 ↓ ₩ ⓓ •L Stroke: N/A	Layer 1 🔻	X: 388,57 Y: 1042,86 Z: 35%

This extension adds a special component which contains general settings of the HMI. Close it with the OK button for now.

		cument			-	N.C	REX HMI Configuration				×
<u>F</u> ile	Edit	View	Layer	<u>O</u> bje	ct <u>P</u> at		Connections Options HTML export)			
	::=# ***]	0 20		<i>4</i> 10. "	• *		Alias	Connection String	Туре		
					- [
	•										^i
											÷ 🔊
											- μ
0)										<u>ک</u>
0 20 7											□ <u>*</u> *
28	'										
							+ - Browse	Save	.csv Load .csv		9
>>> >>			_								» »
		"						ОК	Cancel		Þ
Fill: Strok	e:	N/A N/A	O: ()		3	a 1 Curi. Click to select in groups, t	nag to move nor/vert	Y: 5f	,71 50,00 Z:	35% 韋

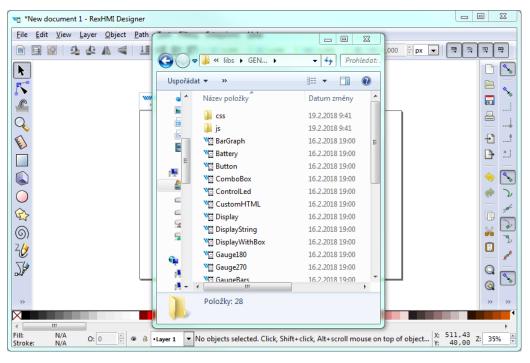
In order to include the HMI during the $\mathsf{REXYGEN}$ project compilation, the file name

has to end with .hmi.svg. Save the file as e.g. designer.hmi.svg to the hmisrc subfolder of your project. Use the standard $File \rightarrow Save as$ menu.

Note: If you want to replace the WebBuDi interface with the REXYGEN HMI Designer interface just delete the index.hmi.js and save the HMI as index.hmi.svg.

A.2 Adding the first HMI components

Now we will add some displays and inputs. The REXYGEN HMI Designer contains a library of components which you can use to build your HMI. The library is available through *Browse Components Library* extensions. Navigate to *Extensions* \rightarrow *REXYGEN HMI* \rightarrow *Browse Components Library* (Ctrl+L). It will open the explorer window with several folders. Open the GENERAL folder and drag&drop the Display to the drawing. The display will be used as an indicator of the remaining time in the TIMER function block.



For configuration of the display settings select the display by mouse click on top of it and use the *Edit Component* extension from *Extensions* \rightarrow *REXYGEN HMI* \rightarrow *Edit Component* (Ctrl+E). When the configuration dialog is opened you can change the *Title* to *Display_remaining*. The Edit Component dialog has two tabs: *Data points* and *Options*.

The *Data points* tab contains three items defining the behavior and animations of the component. Each data point contains an alias, which is in fact a connection to live data from the REXYGEN algorithm.

• value – The value to display.

- disable_by If true the display is disabled and data are no longer updated.
- hide_by If true the display is hidden.

The value property contains T_value . The T will be later automatically substituted by the *Title* of the component, resulting in the Display_remaining_value alias. The disable_by and hide_by data points are optional. Leave them blank at the moment.

The *Options* tab contains several properties which are specific for the Display component. You can find the description of each component and property in [5]. Leave the default values for now and press OK.

Note: Each component is in fact one SVG group with unique content. You can copy the components all over the screen using copy (Ctrl+C) and paste (Ctrl+V) approach.

Edit HMI Component		Edit HMI Component	
Title (\$T): Display_remaining Data points Options	ID: Display Module: Display Version: 10	le (\$T): Display_remaining	ID: Display Module: Display Version: 10
value \$T	_value	format	number 🔻
disable_by		text_format	
hide_by		scale	1
		offset	0
		decimals	2
		color	#000000
		colorAbove	#df2626
		colorBelow	#0074c5
		rangeMax	10
		rangeMin	0
Browse Aliases		units	
	OK Cancel		OK Cancel

Now we will add controls for all the switches (CNB blocks). All of them will be controlled using the *PushOnOff* components. Add them from the library. Select the first *PushOnOff* and open the editor dialog *Extensions* \rightarrow *REXYGEN HMI* \rightarrow *Edit Component* (*Ctrl+E*). Change the title to *PushOnOff_SW_1* and select the *Options* tab. Select the **ToggleButton** item in the **type** property list. Close the *Edit Component* dialog using OK button. Copy-paste the button three times and remember to change the titles to *PushOnOff_SW_2*, *PushOnOff_SW_3*, *PushOnOff_SW_4*).

Note: Throughout the REXYGEN HMI Designer you can use double click in Options tab to open the corresponding configuration dialog (color picker, number input, etc.).

Edit HMI Component		
Title (\$T): PushOnOff_SW_2		g2715 PushOnOff
	Version:	
Data points Options	version:	12
type	ToggleButtor	
reverseMeaning	PushButton ManualPulse	
colorFalse	ManualPulsel ToggleButtor	
colorTrue	#00C8B4	
writePerm	0	
	C	OK Cancel

Now we have four PushOnOff buttons and one Display and we want to link all components with live data from the target device.

A.3 Linking HMI components with the running algorithm

Open the HMI Configuration dialog either using Extensions $\rightarrow REXYGEN HMI \rightarrow Edit$ HMI Config or just unselect all components in the drawing (click outside any component) and press Ctrl+E. The configurator parses all components and creates a list of used Aliases. Each of them should be linked with one signal in the running algorithm. You can either fill in each connection string manually or you can use the Browse function.

Alias	Connection String	Туре
Display_remaining_value		R 🔻
PushOnOff_SW_1		w -
PushOnOff_SW_2		w -
PushOnOff_SW_3		w -
PushOnOff_SW_4		w -

The *Browser* requires a target device with running algorithm. Make sure the algorithm is running, see Section 6.2. Also the target URL must be set. Press *Browse* button. The login dialog will be opened. Change Target to 192.168.1.100:8008 (replace 192.168.1.100 with the IP address of your UniPi Neuron). Unless you changed the login credentials, use the default username admin with an empty password. After a successful login the connection tab is expanded with a tree-view of the running algorithm (you have already seen this tree-view in algorithm diagnostics).

Alias		Connection	String	Туре
Display_remaining_	value			R 🔻
PushOnOff_SW_1				w -
PushOnOff_SW_2				w -
PushOnOff_SW_3				w -
PushOnOff_SW_4				w -
	Target: Usernan Passwor Save pa	d:		
+ - Brow	100		Save .csv	Load .csv

Select *Connection String* field of the Display_remaining_value item and afterwards browse the tree to the TIMER_ block and double-click the rt parameter. The connection string of the parameter is copied to the Display_remaining_value alias, which is shown in the next figure.

Alias	Connection String	Туре	myproject_task
Display_remaining_value	myproject_task.TIMER_:rt	R 🔻	CNB_RUN CNB_SWITCH1
PushOnOff_SW_1		w -	
PushOnOff SW 2		w -	CNB_SWITCH3 CNB_SWITCH4
PushOnOff_SW_3		w -	OR_A
PushOnOff_SW_4		W 🔻	AND_ IND_ INDER
			• <u>•</u> • • • • • • • • • • • • • • • • •
			HLD
			MI R1
			P mode
			P pt
			🕑 Q
			🕒 et
			🕒 rt
			Display
			TRND

Once the display is linked we will also link the switches. Just browse the tree to CNB_SWITCH1, select the *Connection String* field of the *PushOnOff_SW_1* alias and double-click the YCN parameter. Repeat this for the remaining connection strings. Afterwards press OK to save the settings and close the dialog.

REX HMI Configuration		
Connections Options	HTML export \	
Alias	Connection String	Туре
Display_remaining_value	myproject_task.TIMER_:rt	R 🔻
PushOnOff_SW_1	myproject_task.CNB_SWITCH1:YCN	w -
PushOnOff_SW_2	myproject_task.CNB_SWITCH2:YCN	w -
PushOnOff_SW_3	myproject_task.CNB_SWITCH3:YCN	w -
PushOnOff_SW_4	myproject_task.CNB_SWITCH4:YCN	w -
+ - Browse	Save	.csv
	ОК	Cancel

The interactive components in REXYGEN HMI Designer are just parts of the drawing. The user can position the components arbitrarily and add as many decorative static components to as needed. We will add some text descriptions to distinguish individual buttons. Use the *Text tool* (F8), click anywhere in an empty space and start typing. Pick the *Select and Transform tool* (F1) afterwards and move the texts and buttons.

Note: More information about custom drawing can be found in Inkscape tutorials (See the Help \rightarrow Tutorials \rightarrow Inkscape: Basic)

 ayer <u>O</u> bject <u>P</u> ath <u>T</u> ext Fil	tter <u>s</u> Exte <u>n</u> sions <u>H</u> elp ■ ↑ X: 12,000 v Y: 602,0	00 x W: 386,000 x & H: 186,000 x	
WREXYCEN HMI Config switch 1 switch 2 switch 3 switch 4	Timer input S1 OR S2 S3 OR S4 A AND B	Timer timer interval remaining X.XX time output	
C: 0 ⊻ & ∂ •Layer 1 •	Click to Select and Transform	n objects, Drag to select many objects.	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

A.4 Adding more HMI components

Next we will add more components to control the timer and show the status of the OR and AND blocks. Open the Elements library (Ctrl+L) and add one *Input* and four *Led* components. The LEDs will show the status of Boolean values and the *Input* will be used for changing the default timer interval value.

In the *Led* components just edit the Title via the *Edit Component* extension (select the component by single click and press Ctrl+E). The titles should be Led_OR_A, Led_OR_B, Led_AND, and Led_TIMER_OUT respectively. Finally edit the *Input* component by changing the Title to Input_interval.

The last component we need is a graph showing the time-plot of data from the TRND block. Use the components library (Ctrl+L) and add a TRND component. You can adjust its size to fit the desired position. Edit the component (Ctrl+E) and change the Title to TRND and switch to the *Options* tab. Double-click the signals property. Add the following labels using the plus (+) button: timer out, rem. time, OR A, OR B. These labels will be shown in the legend of the graph.

W Edit	HMI Component			8
Title (\$1	T): TRND	ID:	TRND	
		Module:	TRND	
		Version:	10	
	points Options		8 X	
ť				
s		label		
Ł	timer out			
r	rem. time			
н	OR A			
	OR B			
	+ •	ОК	Cancel	
			ОК Са	ncel

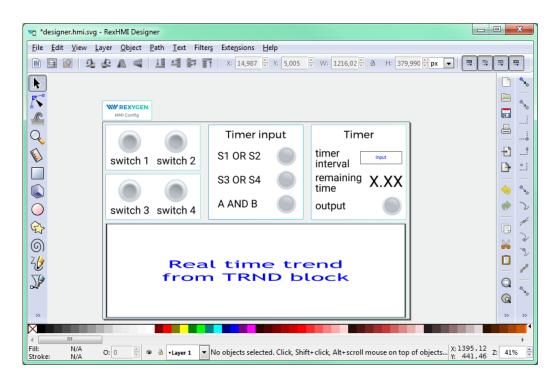
A.5 Additional links to the running algorithm

When all the components are in place we link them to the running algorithm again. Just repeat the procedure described in Section A.3, open the HMI Configuration dialog (*Extensions* \rightarrow *REXYGEN HMI* \rightarrow *Edit HMI Config* and browse the running algorithm to pair the remaining aliases with corresponding connection strings. The list is shown in the following image.

Alias	Connection String	Туре
Display_remaining_value	myproject_task.TIMER_:rt	R -
Input_interval_W	myproject_task.TIMER_:pt	w -
Led_AND	myproject_task.AND_:Y	R -
Led_OR_A	myproject_task.OR_A:Y	R 🔻
Led_OR_B	myproject_task.OR_B:Y	R 🔻
Led_TIMER_OUT	myproject_task.TIMER_:Q	R 🔻
PushOnOff_SW_1	myproject_task.CNB_SWITCH1:YCN	w -
PushOnOff_SW_2	myproject_task.CNB_SWITCH2:YCN	w -
PushOnOff_SW_3	myproject_task.CNB_SWITCH3:YCN	w -
PushOnOff_SW_4	myproject_task.CNB_SWITCH4:YCN	w -
TRND	myproject_task.TRND:u1	R 🔻

A.6 The final steps

Congratulations, your first graphical HMI is almost ready! Add a few rectangles which will visually divide the HMI into individual sections. Use the *Squares and rectangles tool* (F4), draw the rectangle, pick a color from the palette and send the rectangle to the background using the End key. Do not forget to save the drawing.



As mentioned earlier, the *.hmi.svg files in the hmisrc folder are automatically processed while the project is compiled in the REXYGEN Studio application. The project main file must contain the HMI block with GenerateRexHMI parameter enabled. This was already covered in Section 6.3 so you should have everything ready.

Once you compile the project again and download it to your UniPi Neuron, the HMI will be accessible via a web browser. Navigate to

http://192.168.1.100:8008/hmi/designer.html (replace 192.168.1.100 with the IP address of your UniPi Neuron). You will see your HMI with live data.



This tutorial covers only the very basic components. If you want to get more information about additional components, see [5].

Appendix B

Licensing of the REXYGEN system

The licensing model of the **REXYGEN** system is quite simple:

- The development tools are free to use, you can install it on as many computers as you want.
- The RexCore runtime module always needs a licence to run on your UniPi Neuron. There are DEMO licences available at no cost and there are permanent licences which you can purchase. Each UniPi Neuron needs an individual licence.

B.1 Obtaining a DEMO licence

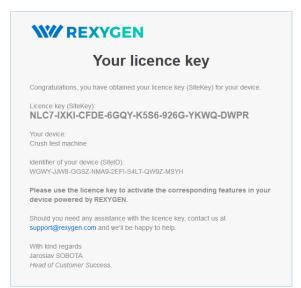
The DEMO licence is intended for evaluating, testing and educational purposes. Feel free to experiment with the DEMO licence as long as you need. Commercial use of the DEMO licence is not allowed.

When you try to run your algorithm on a device which does not have a valid licence, you are offered a few options. Unless you already have a licence key or a licence voucher, you'll need to get a DEMO licence.

W No licence on target device	×
No valid licence key has been found on the target device.	
Request a free DEMO LICENCE	
Try REXYGEN at no cost	
I have a LICENCE VOUCHER	
Apply your licence voucher and get a licence key	
I have a LICENCE KEY	
Enter your licence key and you are ready to go	
Connect anyways	
Target device identifier (Site ID):	
WGWY-JJW8-GGSZ-NMA9-2EFI-S4LT-QW9Z-MSYH Version:	
REXYGEN runtime core version 2.50.7.9077	

Identify yourself and you'll receive a DEMO licence key via e-mail. The licence key is called a SiteKey.

₩ Get DEN	10 licence key X
First name:	John
Last name:	Doe
Email:	johndoe@domain.com
Description (for your reference):
Crush test r	nachine
	Get DEMO licence
Privacy polic	y statement Registered REXYGEN user?



Once you have your DEMO licence key (SiteKey), apply it.

WW Ne lines on terrat daying	×	
W No licence on target device	~	•
No valid licence key has been found on the target devic	e.	
Request a free DEMO LICENCE Try REXYGEN at no cost		
I have a LICENCE VOUCHER Apply your licence voucher and get a licence key		
I have a LICENCE KEY Enter your licence key and you are ready to go		
Connect anyways		
Target device identifier (Site ID): WGWY-JJW8-GGSZ-NMA9-2EFI-S4LT-QW9Z-MSYH Version: REXYGEN runtime core version 2.50.7.9077		
W Apply licence key	?	×
Enter your licence key:		
NLC7-IXKI-CFDE-6GQY-K5S6-926G-YKW	Q-D	WPR
Apply licence key		



Once successfully applied, your algorithm will download to your UniPi Neuron. You can review your licence keys in REXYGEN Studio. Go to menu $Target \rightarrow Li$ censing to list the licence keys and activated modules.

Site ID: WGWY-JJW8-GGSZ-	MMA9-2	2EFI-S4LT-QW	19z-msyh
Licensed fe	atures		Registered keys
Feature	Expires	Licence type	NLC7-IXKI-CFDE-6GQY-K5S6-926G-YKWQ-DWP
RexCore Plus	Never	Demo	
Standard function blocks	Never	Demo	
Advanced function blocks	Never	Demo	
Motion control blocks (single- and	Never	Demo	
Modbus driver - TCP master	Never	Demo	
Modbus driver - serial master (RTU)	Never	Demo	
Modbus driver - TCP slave	Never	Demo	
Modbus driver - serial slave (RTU)	Never	Demo	
EtherCAT driver - master node	Never	Demo	
CAN/CANopen driver	Never	Demo	
1-Wire driver (via OWFS)	Never	Demo	
Siemens PLC communication driver	Never	Demo	
Database access driver (via ODBC)	Never	Demo	
Advanced matrix function blocks	Never	Demo	
REXLANG programmable function	Never	Demo	
Motion control blocks (coordinate	Never	Demo	
Omron PLC communication driver	Never	Demo	

Evaluation version of RexCore (REXYGEN runtime core) is functional for 2 hours. It is possible to run your algorithm on the UniPi Neuron but you cannot store it permanently. The algorithm resides only in the RAM memory therefore it will not run after a restart or power-up. You can use almost all function blocks, see [1]. The RexCore runtime core on the target device is terminated after 2 hours of operation in demo mode without any warning. After a restart, you have another 2 hours for your experiments.

B.2 Obtaining a permanent licence

It is necessary to activate the RexCore runtime module and optional additional modules for permanent operation. This can be done using the licence, which you can obtain at

www.rexygen.com/pricing

B.2.1 Activation of the permanent licence

Each device running the RexCore runtime module is identified by the so-called SiteID tag. The purchased licence must be associated with the hardware device, i.e. with the SiteID tag.

1. In REXYGEN Studio, connect to the device and go to menu $Target \rightarrow Licensing$. A dialog pops up and you will see the SiteID and your licences.

ite ID: WGWY-JJW8-GGSZ-I	MMA9-	2EFI-S4LT-QW	9Z-MSYH
Licensed fe	atures		Registered keys
Feature	Expires	Licence type	NLC7-IXKI-CFDE-6GQY-K5S6-926G-YKWQ-DWP
RexCore Plus	Never	Demo	
Standard function blocks	Never	Demo	
Advanced function blocks	Never	Demo	
Motion control blocks (single- and	Never	Demo	
Modbus driver - TCP master	Never	Demo	
Modbus driver - serial master (RTU)	Never	Demo	
Modbus driver - TCP slave	Never	Demo	
Modbus driver - serial slave (RTU)	Never	Demo	
EtherCAT driver - master node	Never	Demo	
CAN/CANopen driver	Never	Demo	
1-Wire driver (via OWFS)	Never	Demo	
Siemens PLC communication driver	Never	Demo	
Database access driver (via ODBC)	Never	Demo	
Advanced matrix function blocks	Never	Demo	
REXLANG programmable function	Never	Demo	
Motion control blocks (coordinate	Never	Demo	
Omron PLC communication driver	Never	Demo	

2. Delete any DEMO licence keys if present.

3. Afterwards, click the *Apply licence voucher* button and enter your licence voucher code.

W Apply licence voucher X
Licence voucher:
VZI-QMQ-61U
Description (for your reference):
Crush test machine
Apply licence voucher
Privacy policy statement Registered REXYGEN user?
Target device identifier (Site ID):
WGWY-JJW8-GGSZ-NMA9-2EFI-S4LT-QW9Z-MSYH
Version:
REXYGEN runtime core version 2.50.7.9077

4. You are asked to confirm the association – this is the last and irreversible step.

W Apply licence voucher permanently?						
This is irreversible operation. Once the voucher is applied, it is permanently tied to the hardware device.						
Are you sure you wish to proceed?						
	YES					
A licence key will be generated						
NO						

5. The so-called SiteKey licence key is generated from your licence voucher and it is stored in your device. The licence key allows permanent operation of the runtime core.



6. To check the licence, make sure the REXYGEN Studio is connected to the device and open the licensing dialog again ($Target \rightarrow Licensing$). Check that the licence was applied correctly and that the RexCore runtime module no longer runs in demo mode.

ite ID: WGWY-JJW8-GGSZ-I	MMA9-2	2EFI-S4LT-QW	9z-msyh
Licensed features			Registered keys
Feature	Expires	Licence type	GENM-MU3J-HYHC-PMD5-IS92-BWTW-7U6M-5RP
Standard function blocks	Never	Regular	
Advanced function blocks	Never	Regular	
Modbus driver - TCP master	Never	Regular	
Modbus driver - serial master (RTU)	Never	Regular	
Modbus driver - TCP slave	Never	Regular	
Modbus driver - serial slave (RTU)	Never	Regular	
CAN/CANopen driver	Never	Regular	
1-Wire driver (via OWFS)	Never	Regular	
Siemens PLC communication driver	Never	Regular	
Database access driver (via ODBC)	Never	Regular	
REXLANG programmable function	Never	Regular	
Autotuning controllers	Never	Regular	
RexCore Professional	Never	Regular	
Omron PLC communication driver	Never	Regular	

7. It is highly recommended to check the licence again after a reboot of the device.

Bibliography

- [1] REX Controls s.r.o.. Function blocks of REXYGEN reference manual, 2020. \rightarrow .
- [2] REX Controls s.r.o.. RexCore User manual, 2020. \rightarrow .
- [3] REX Controls s.r.o.. OpcUaDrv driver of REXYGEN user guide, 2020. \rightarrow .
- [4] REX Controls s.r.o.. REXYGEN Studio User manual, 2020. \rightarrow .
- [5] REX Controls s.r.o.. REXYGEN HMI User manual, 2020. \rightarrow .
- [6] REX Controls s.r.o., Modbus driver of REXYGEN User guide, 2020. \rightarrow .

Documentation reference number: 14711