www.rexcontrols.com/rex

Getting started with REX and Raspberry Pi

User guide

REX Controls s.r.o.

Version 2.50.1
2016-11-07
Plzeň (Pilsen), Czech Republic

Contents

1 Introduction **3**
- 1.1 Features of the REX Control System . . . 3
- 1.2 Structure of the REX Control System . . . 4
- 1.3 Programming in the REX Control System . . . 4
- 1.4 Main components of the REX Control System . . . 5
  - 1.4.1 RexDraw – the development environment . . . 5
  - 1.4.2 RexHMI Designer – the user interface . . . 5
  - 1.4.3 RexComp – the compiler . . . 5
  - 1.4.4 RexCore – the runtime core . . . 5
  - 1.4.5 RexView – the diagnostic tool . . . 5

2 Installation of the development tools **6**
- 2.1 Windows Vista/7/8/10 . . . 6
  - 2.1.1 Installation procedure . . . 6
  - 2.1.2 Uninstall procedure . . . 7

3 Installing the runtime modules of the REX Control System on Raspberry Pi **8**
- 3.1 Installation of components on Raspberry Pi . . . 8
- 3.2 Installed files and folders . . . 9
- 3.3 Uninstall procedure . . . 9

4 Configuration, compilation and execution **10**
- 4.1 Creating a new project . . . 10
- 4.2 Compiling and running a project . . . 13
- 4.3 Adding a user interface (HMI) . . . 19
- 4.4 Ready for interaction with the outer world . . . 25

5 IO configuration for the Raspberry Pi platform **26**
- 5.1 Adding inputs and outputs to the project . . . 26
- 5.2 Working with GPIO pins . . . 28
  - 5.2.1 Modifications in the project main file . . . 28
  - 5.2.2 Modifications in the task . . . 30
- 5.3 Additional information . . . 34

5.3.1 Detailed description of the driver 34
5.3.2 Examples 34

6 Licensing of the REX Control System **35**
6.1 Obtaining a DEMO licence 35
6.2 Obtaining a permanent licence 37
6.2.1 Activation of the permanent licence 37

Bibliography **41**

Chapter 1

Introduction

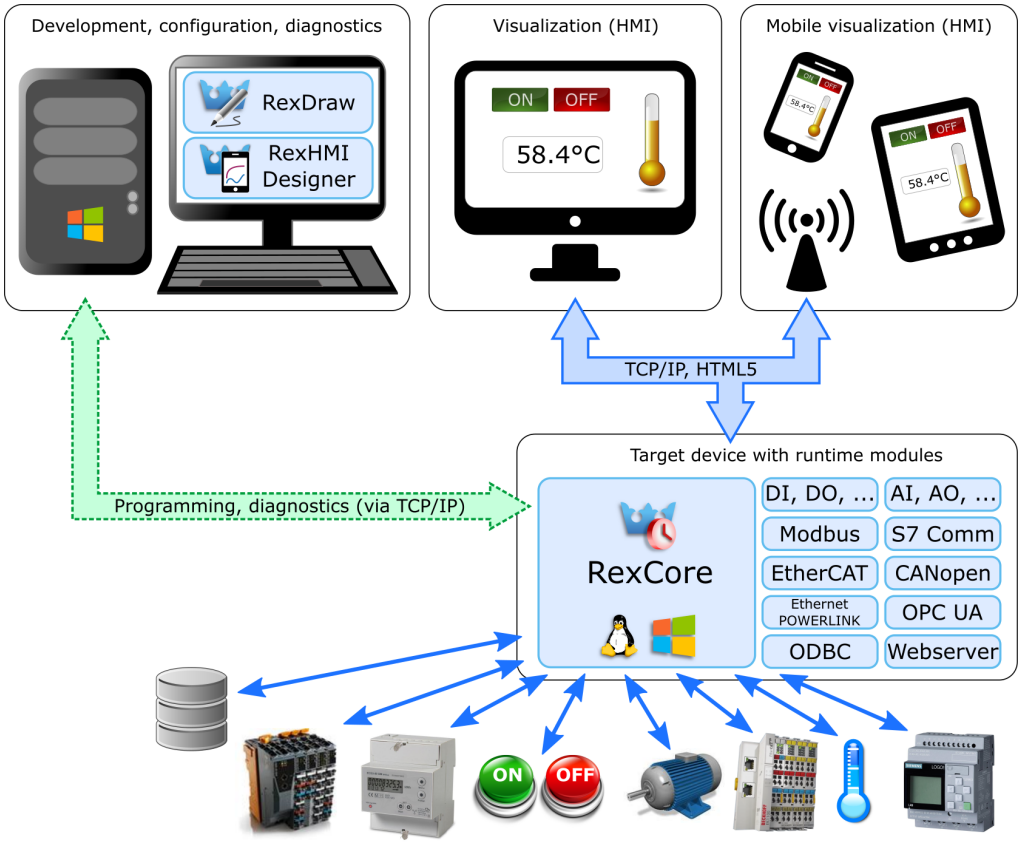
The REX Control System is a family of software products for automation projects. You can use it in all fields of automation, robotics, measurements and feedback control.

The runtime core of the REX Control System turns your Raspberry Pi into a programmable device which will run your algorithms.

1.1 Features of the REX Control System

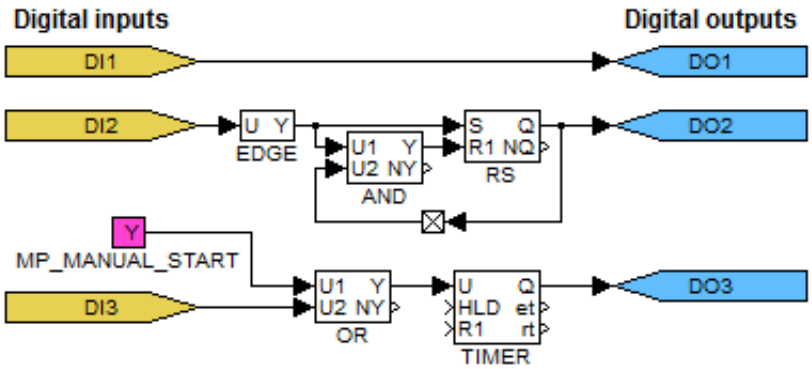
- Graphical programming without hand-coding
- Programming control units on a standard PC or laptop
- User interface for desktop, tablet and smartphone (HMI)
- Wide family of supported devices and input-output units (including Raspberry Pi)
- Industry-proven control algorithms
- Easy integration into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

1.2 Structure of the REX Control System



1.3 Programming in the REX Control System

The REX Control 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 REX Control System

1.4.1 RexDraw – the development environment

The RexDraw graphical environment is a developer's tool. You create the algorithms using the function block library[1] of the REX Control System [1]. The library contains simple comparators and timers as well as advanced blocks for signal processing and feedback control (PID controllers etc.). You can compile and run your algorithms on your Raspberry Pi.

Once running, you can watch your algorithm in real-time. Just select the signals and function blocks of your interest. You can connect via local network or over the Internet.

1.4.2 RexHMI Designer – the user interface

The RexHMI 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 Raspberry Pi along with the algorithm.

1.4.3 RexComp – the compiler

The RexComp compiler converts your algorithms into binary code of the REX Control System. The compiler is almost invisible for the user, it is called from the RexDraw development environment. The compiler detects and reports possible errors in your algorithms.

1.4.4 RexCore – the runtime core

The RexCore runtime core runs on the target device (Raspberry Pi). It handles timing and execution of your algorithms and provides various services. The individual tasks are prioritized and executed using preemptive multitasking.

RexCore further contains an integrated webserver providing user interface (HMI) and REST API.

1.4.5 RexView – the diagnostic tool

With RexView you can diagnose the runtime core and execution of your algorithm. It is an important tool for commissioning and diagnostics of control algorithms. You obtain detailed hierarchical information about the running control algorithm. You can connect via local network or over the Internet.

[1] The IEC 61131-3 standard defines Function Block Diagram (FBD) as one of the PLC programming techniques.

Chapter 2

Installation of the development tools

This chapter describes the steps to install and uninstall the development tools of the REX Control System on Windows Vista/7/8/10 operating systems.

2.1 Windows Vista/7/8/10

The installation package of the REX Control System development tools contains the RexDraw and RexHMI Designer development environments, the RexComp compiler and the RexView diagnostic tool. It also includes the RexCore runtime module for developing and testing purposes.

The installation package can be downloaded from
https://www.rexcontrols.com/software-download.

The installation process requires the administrator rights on your PC.

2.1.1 Installation procedure

1. Run the `REX-X.XX.XX.XXXX-PPP.exe` downloaded from https://www.rexcontrols.com/software-download.
2. Select the language and follow the installation wizard.
3. Select the target installation folder, the default is `C:\Program Files (x86)\REX Controls\REX X.XX.XX.XXXX`.
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.

There is no need to restart the system after installation.

2.1.2 Uninstall procedure

The common procedure can be used to uninstall the REX Control System – go to *Control panel* and choose *Install/Uninstall programs*.

Chapter 3

Installing the runtime modules of the REX Control System on Raspberry Pi

It is necessary to have a correctly configured Internet connection on your Raspberry Pi running the so-called Raspbian distribution of GNU/Linux[1] prior to using this guide. Visit www.raspberrypi.org for more information.

3.1 Installation of components on Raspberry Pi

1. On your Raspberry Pi, install GIT
   `sudo apt install git`
2. Go to your home directory
   `cd ~`
3. Download the latest revision of installation scripts
   `git clone https://github.com/rexcontrols/rex-install-rpi.git`
4. Change the working directory
   `cd rex-install-rpi`
5. Depending on your hardware, run **ONLY ONE** of the following installation scripts
   - For bare Raspberry Pi:
     `sudo bash install-rex.sh`
   - For Raspberry Pi with the Monarco HAT:
     `sudo bash install-rex-monarcohat.sh`
   - For Raspberry Pi with UniPi extension board:
     `sudo bash install-rex-unipi.sh`

[1] RexCore should also work on any other distribution based on Debian (e.g. Ubuntu).

- For Raspberry Pi with PiFace Digital extension board:
  `sudo bash install-rex-pifacedigital.sh`

6. Perform reboot if you are asked for it and you are done.

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 Raspberry Pi using the RexDraw and/or RexView programs. RexCore also automatically runs upon system (re)start.

3.2 Installed files and folders

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

3.3 Uninstall procedure

Use the `sudo apt-get remove rexcore` command to remove RexCore from the operating system.

Chapter 4

Configuration, compilation and execution

The process of creating the control algorithm will be demonstrated on a very simple example with two Boolean variables representing manual switches, which will be later replaced by physical inputs of the Raspberry Pi. A timer will be used for measuring the time when both variables are true (i.e. both switches are in the ON position). A Boolean signal will be used for signalling that the interval of predefined length has elapsed.

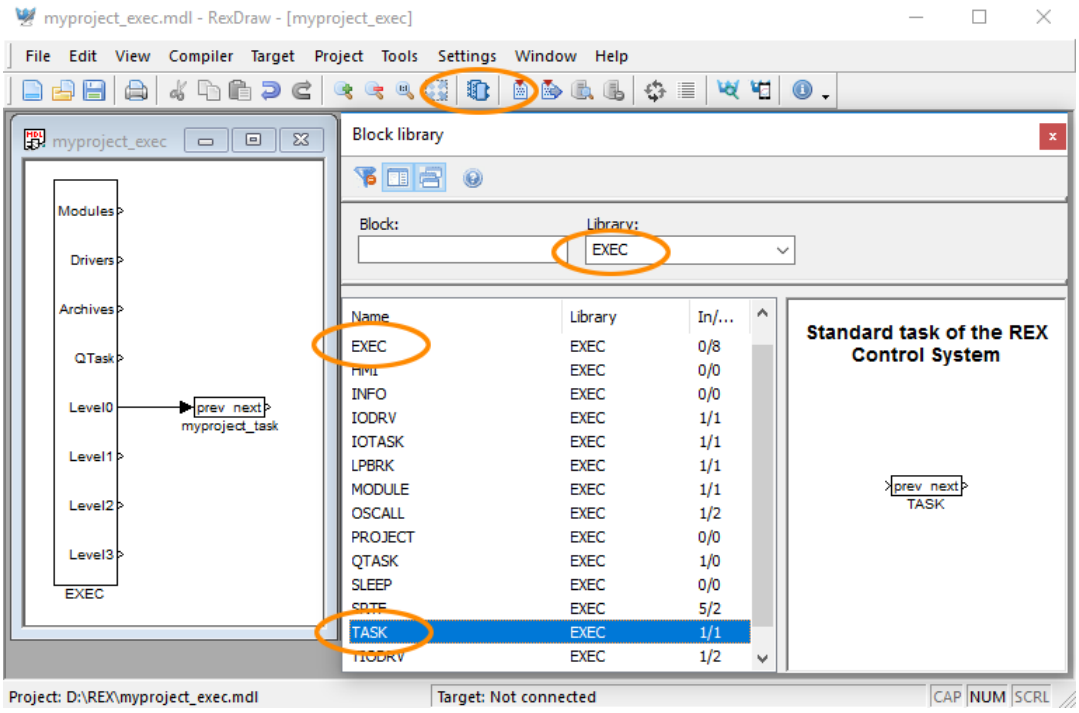
4.1 Creating a new project

The project configuration is created using the RexDraw 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).

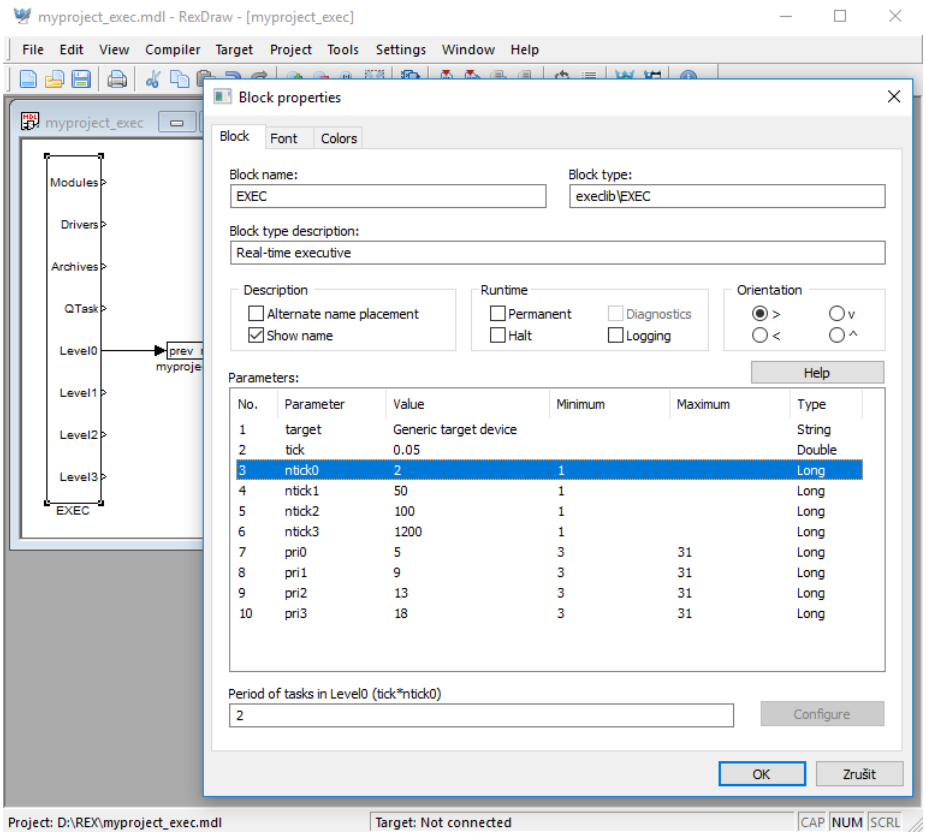
Standard approach:

1. Run the RexDraw program, start with an empty project and save the new file as e.g. `myproject_exec.mdl`.
2. Open `Block Library`, choose *View/Block Library* in the menu or use the icon from the toolbar.
3. Drag&drop the main block called `EXEC` and one `TASK` block to the `myproject_exec.mdl` file. Both of them are contained in the `EXEC` library. Further the location of blocks within the libraries will be denoted as `library/block`, e.g. `EXEC/TASK`.
4. Now connect the two blocks, namely the `Level0` output of the `EXEC` block and the `prev` input of the `TASK` block. To connect the blocks, drag the output arrow to the input arrow 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.



5. Each block can be configured by double-clicking on it. A block parameters and properties dialog appears. The parameters of all blocks of the REX Control System are described in the Help (press the F1 key) and in the Block reference manual [1].



6. Set `ntick0` = 2 in the block parameters dialog of the `EXEC` block, i.e. the task connected to the `Level0` output will be executed each `tick*ntick0`=0.1 seconds.

7. Rename the `TASK` block to `myproject_task` (the `Block name` field in the block parameters dialog). By this, we define the file associated with the `TASK` block. The `myproject_task.mdl` file will be created later.

8. Save the `myproject_exec.mdl` file. At this moment it is necessary to define the main file of the project. Pick *File/Set as main* in the menu. The window title changes to `myproject_exec - RexDraw [myproject_exec]`.

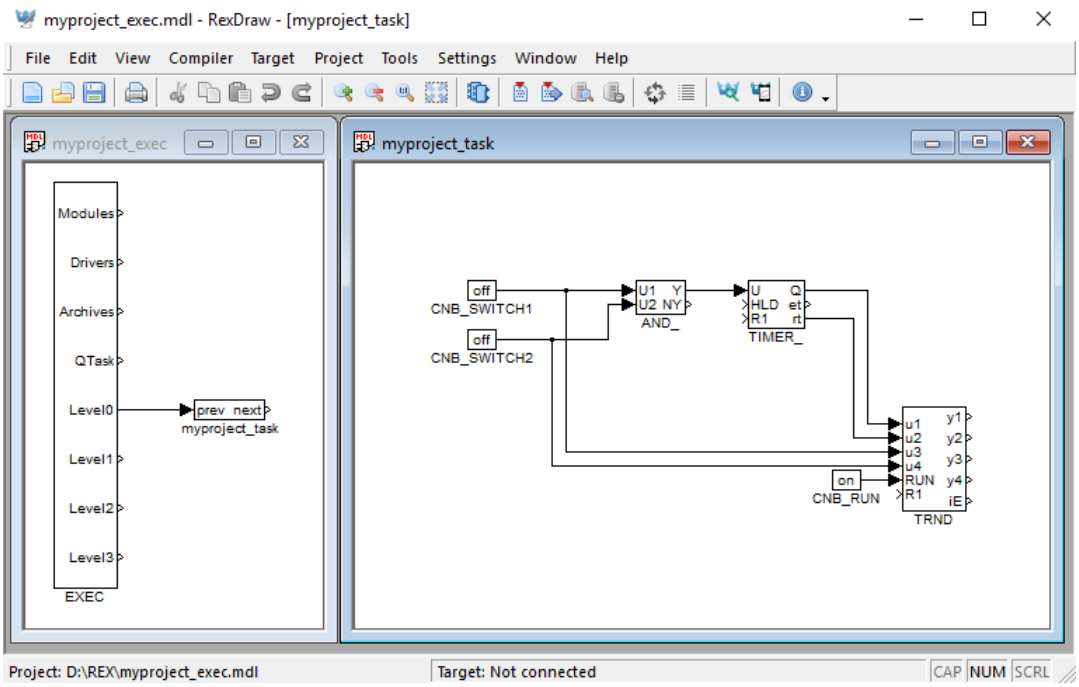
9. Open a new file and save it as `myproject_task`. Use the following function blocks from `Block Library`:

   - `MATH/CNB` – constant of type `Boolean`, change name to `CNB_SWITCH1`, set parameter `YCN` = `off`,
   - `MATH/CNB` – constant of type `Boolean`, change name to `CNB_SWITCH2`, set parameter `YCN` = `off`,
   - `LOGIC/AND_` – logical AND,
   - `MATH/TIMER_` – a timer block, set parameter `mode` = 2 : *DelayedON*, `pt` = 5,

- `ARC/TRND` – real-time recording, set parameters `l` = 500, `Title` = `My timer`, leave the default values otherwise
- `MATH/CNB` – constant of type `Boolean`, change name to `CNB_RUN`, set parameter `YCN` = `on`,

Connect the blocks as shown below.

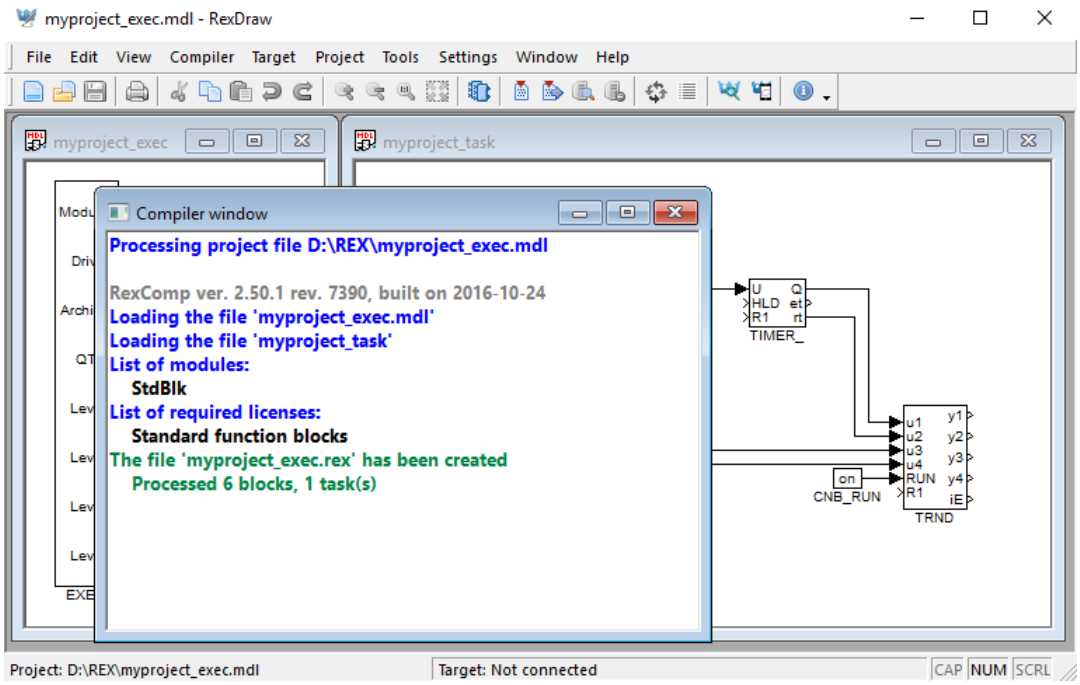
New branch of a line can be created by right mouse button dragging.



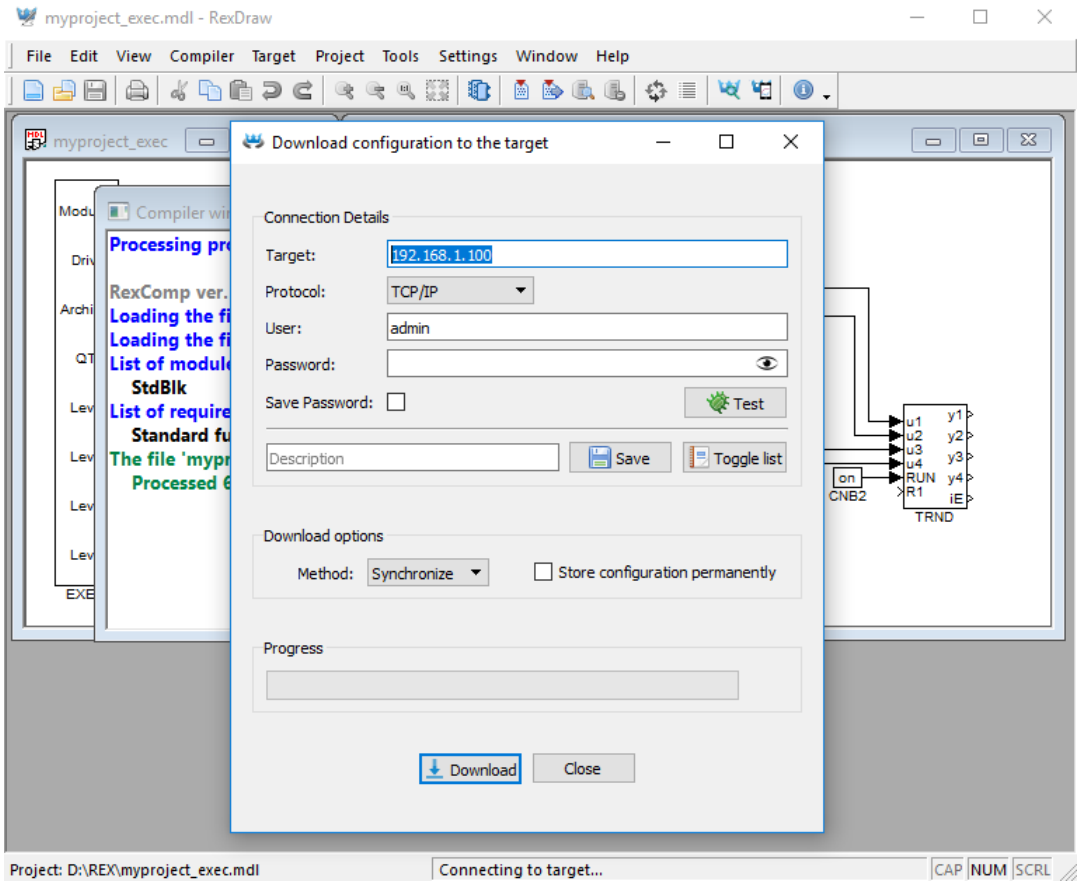
At this moment the executive `myproject_exec.mdl` and the corresponding file with the algorithm `myprojec_task.mdl` are ready for compilation.

4.2 Compiling and running a project

The developed algorithm must be compiled to binary form prior to deploying. Pick *Compiler/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.



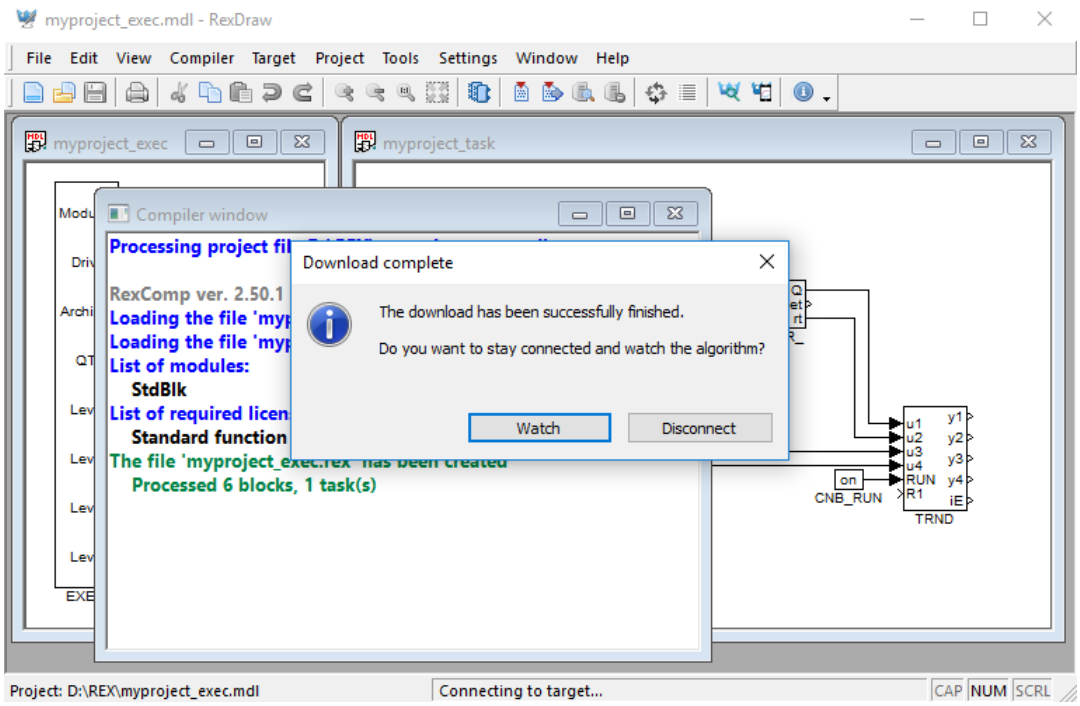
At this moment it is possible to deploy the control algorithm to the target platform. Use the `Compile and Download` icon for this purpose. A dialog for defining the target device appears upon successful compilation.



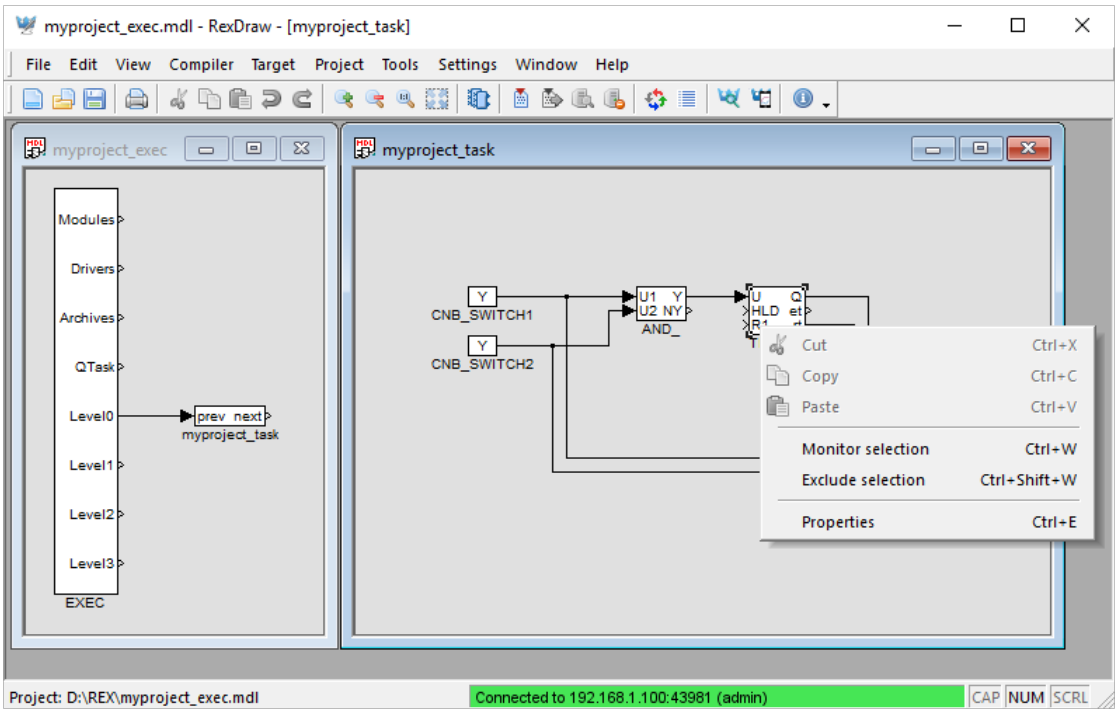
Enter the IP address of the Raspberry Pi in the `Target` field. The default user is admin and there is no password by default. Leave the other elements intact and click `Download`.

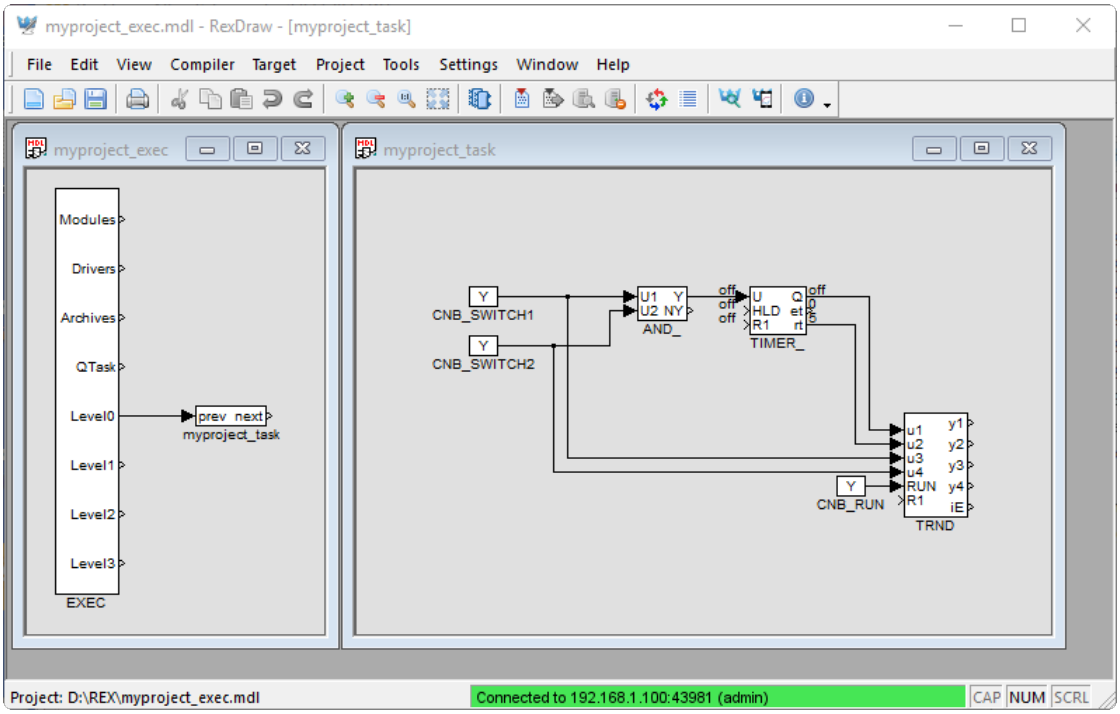
If there is no licence on your Raspberry Pi, you need to get one first. See chapter 6 for details and come back afterwards.

As soon as the download is complete it is possible to connect to the target device and watch the control algorithm in action – click `Watch`.



The RexDraw program now works in the so-called online mode. Right-click the `TIMER_` block and select `Monitor selection` in the menu to watch the inputs and outputs of the timer.

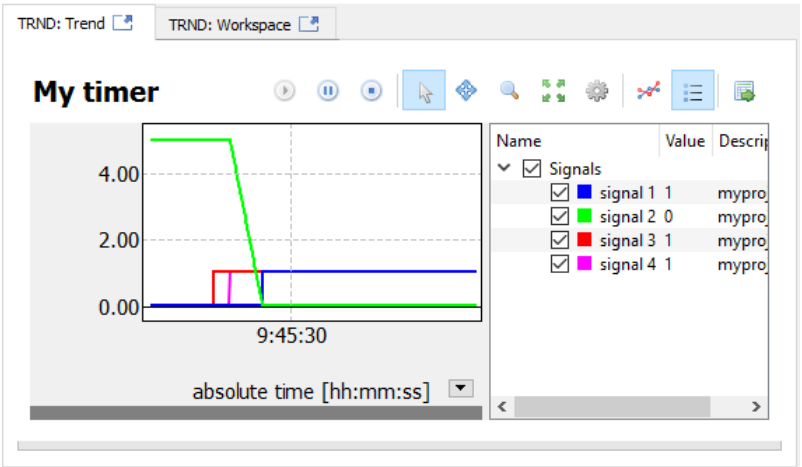




You can do the same with the `CNB` blocks or any other selection.

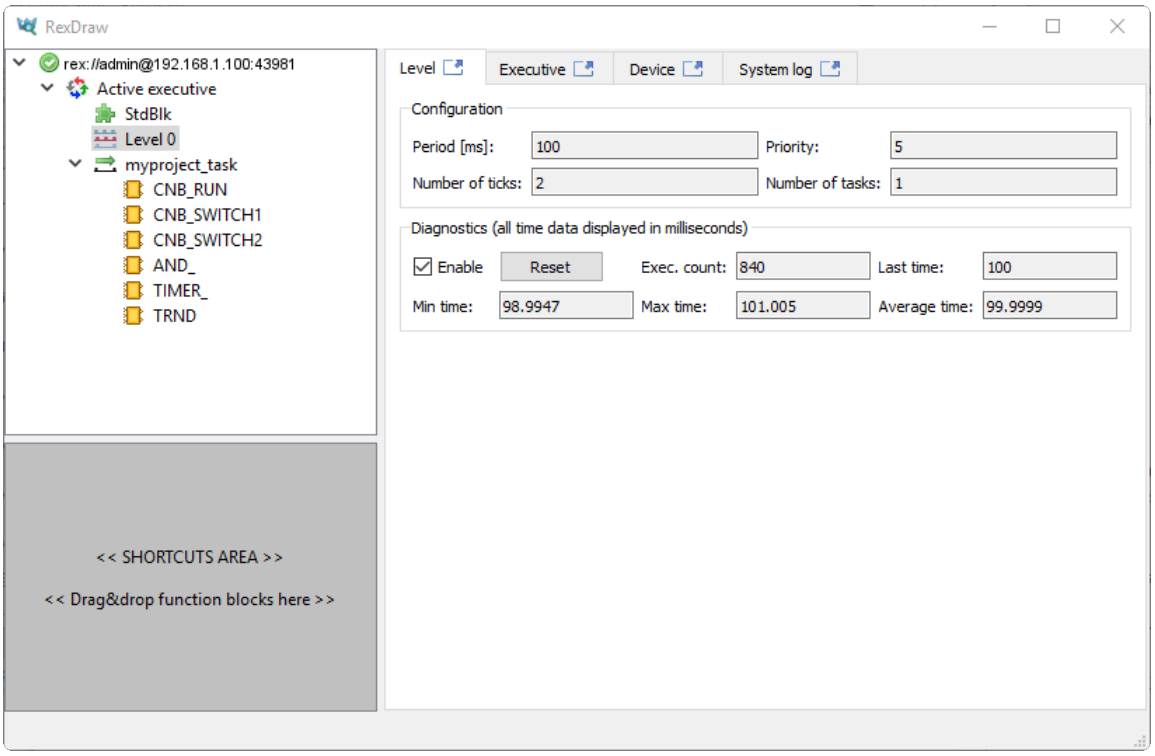
Now it is possible to double-click the `CNB_SWITCH1` block and change the Boolean variable to `YCN` = `on` (tick the checkbox and click OK). Once you do the same with the `CNB_SWITCH2` block, the `AND_` block output `Y` goes `on` and the `TIMER_` starts to count down (observe the `rt` output). Once the timer reaches zero, its output `Q` is set to `on` and it remains `on` as long as the `U` input is `on` (both the switches are `on`).

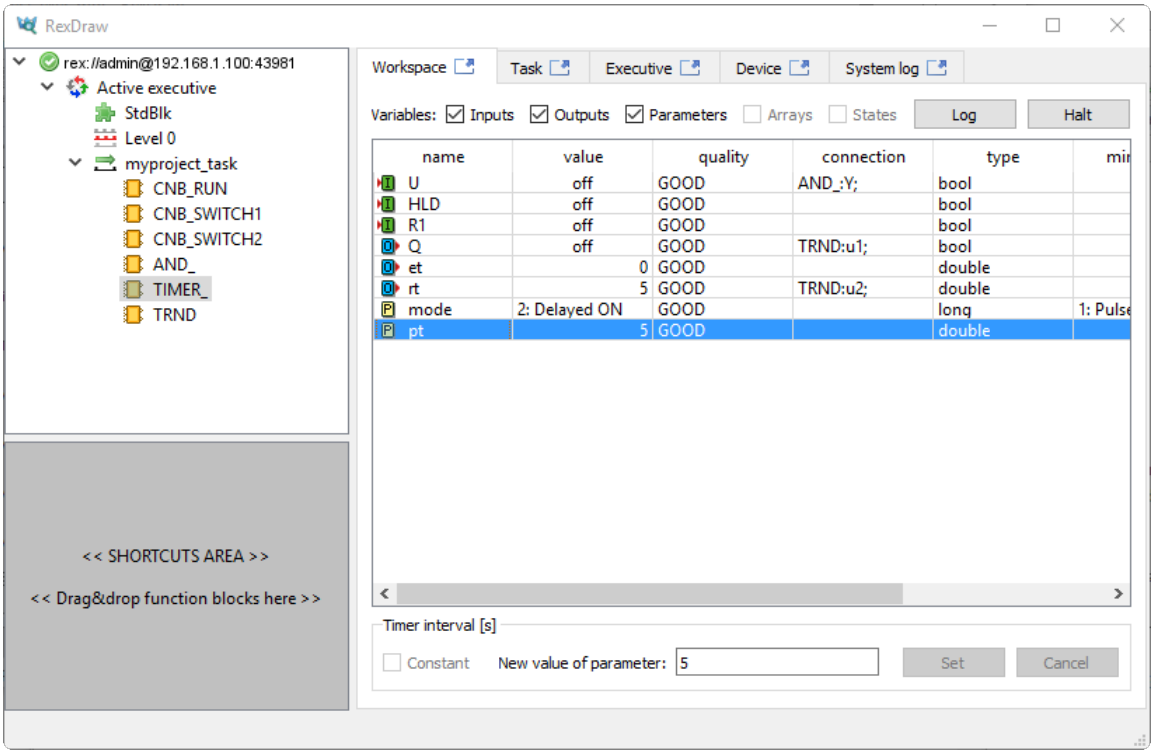
You can double-click the `TRND` block to see the signals in a real-time graph. Red line is the first switch, magenta line is the second switch, green line is the remaining time of the timer and blue line is the output of the timer.



Try turning the `CNB` blocks `off`, change the `pt` parameter of the `TIMER_` block, engage the switches 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. Click the and you will see the algorithm in a tree view which allows you to monitor the control algorithm in full detail. 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 separate diagnostics program called RexView. Click the RexView icon and confirm the IP address of the target platform.

Now you can disconnect RexDraw from the target device using the *Disconnect* icon. The RexDraw program offers synchronization of the changed parameters with the source file of the project, choose `No` at this moment.

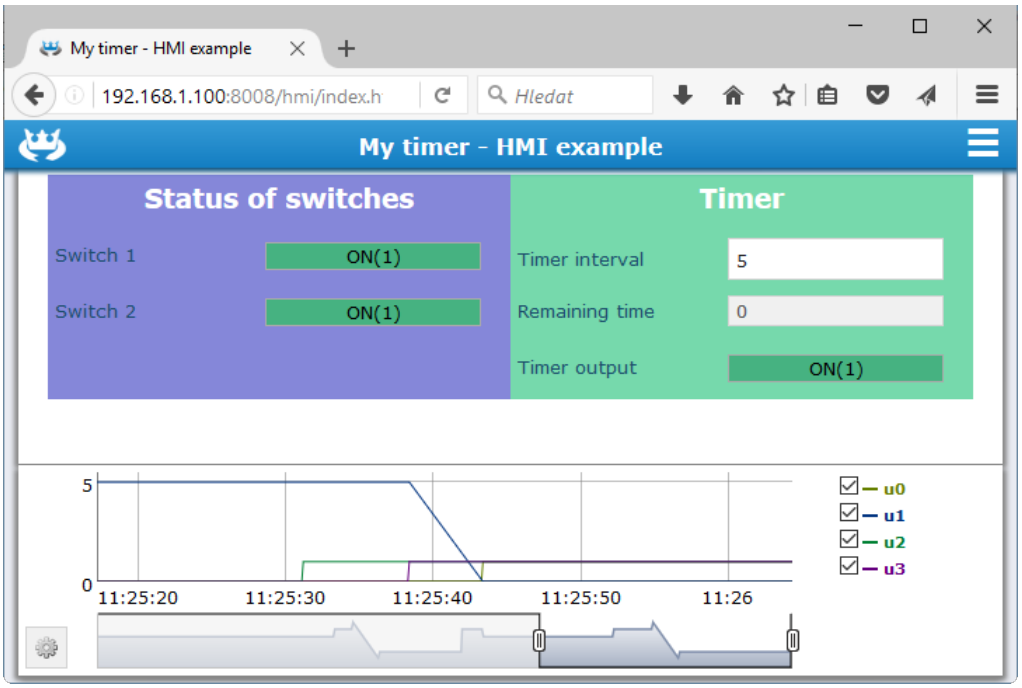
All changes made in online mode are not permanently stored in the target device (unless you decide so, see [3]). Upon restart of RexCore runtime module the algorithm will start with the parameters previously compiled and downloaded to the target device. To store the changes permanently, tick them when disconnecting and transfer the changes to the source files. Now *Compile and download* the project one more time and the changes become permanent.

4.3 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 who is unfamiliar with the REX Control System to interact with the algorithm. The HMI of the REX Control System relies on modern web-based technology and the HMI is thus accessible via web browser on desktop PC, tablet or smartphone.

In this tutorial, only a simple HMI called WebBuDi will be created. It provides very simple indicators and input fields to interact with the control algorithm via a web page. See [4] for information on creating more advanced graphical HMI screens.

This is how the HMI will look like and the steps to create it are below.

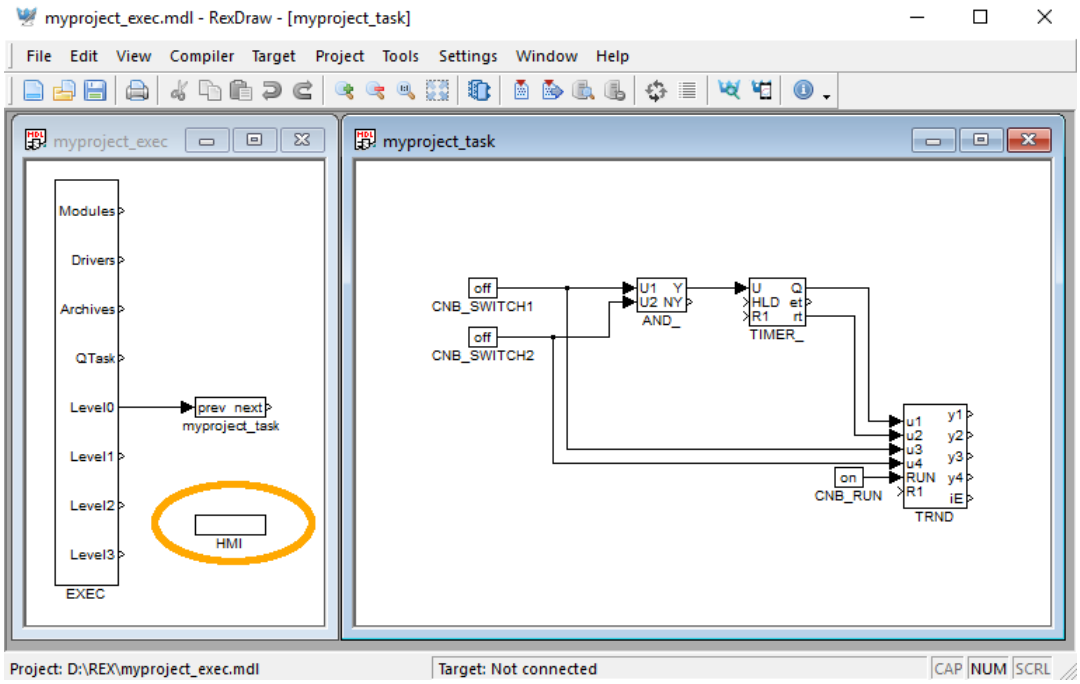


In the folder with the project files, create a `hmisrc` subfolder. Inside this folder, create a file named `index.html.js` and edit it with your favorite text editor. The content should be the following:

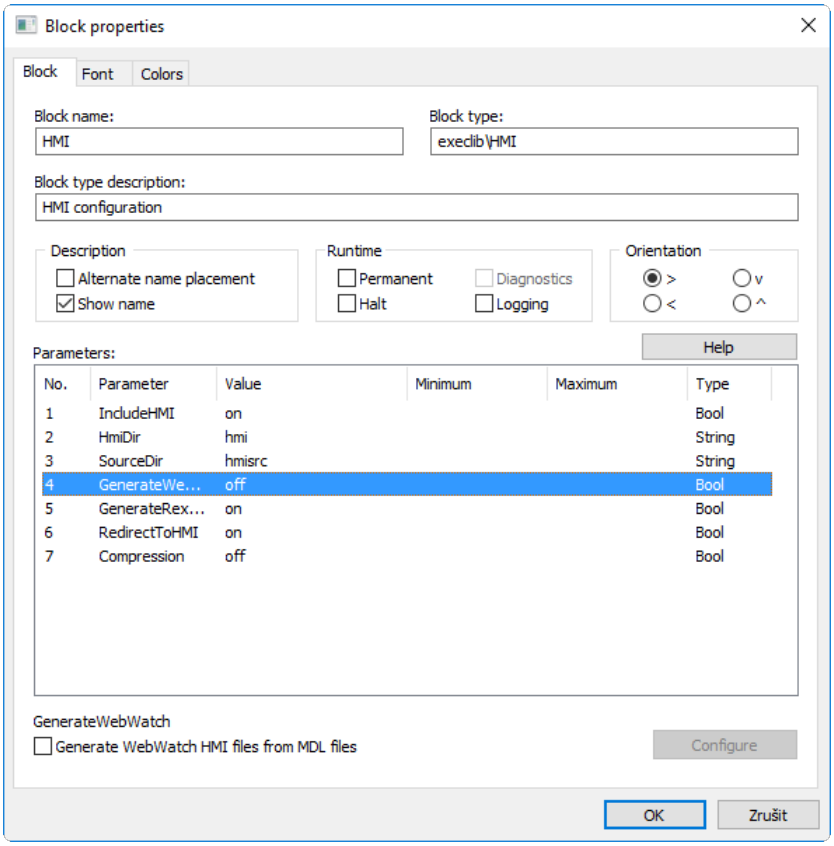
```
REX.HMI.init = function(){

//Indicators displaying the status of switches
  var switches = {
    column: 1,
    title: 'Status of switches',
    rows: [
      {alias: 'switch1', desc: 'Switch 1',
       cstring: 'myproject_task.AND_:U1', type: 'DR'},
      {alias: 'switch2', desc: 'Switch 2',
       cstring: 'myproject_task.AND_:U2', type: 'DR'},
      {type: 'ES'}
    ]
  };
  REX.WebBuDi.addSection(switches);
```

```
  //Timer settings and status
  var timer = {
    column: 2,
    title: 'Timer',
    rows: [
      {alias: 'interval', desc: 'Timer interval',
       cstring: 'myproject_task.TIMER_:pt', type: 'AW'},
      {alias: 'rt', desc: 'Remaining time',
       cstring: 'myproject_task.TIMER_:rt', type: 'AR'},
      {alias: 'timerQ', desc: 'Timer output',
       cstring: 'myproject_task.TIMER_:Q', type: 'DR'}
    ]
  };
  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');
}
```

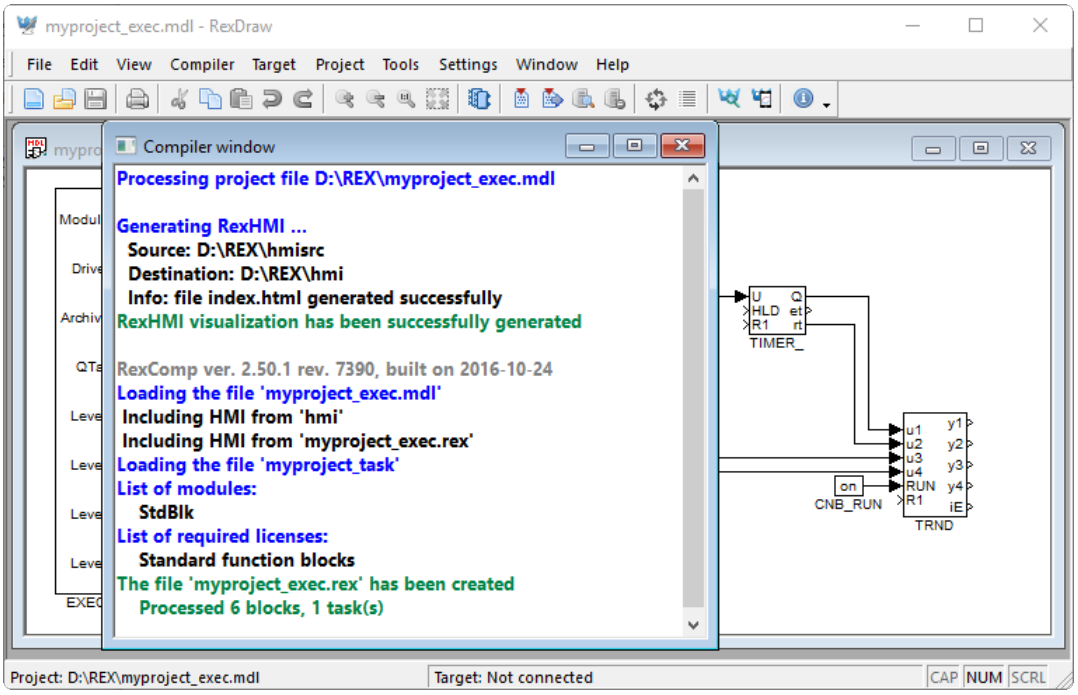
This file will be processed when compiling the project. However, it is necessary to add the `EXEC/HMI` block into the project main file first.



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 [4] for details.

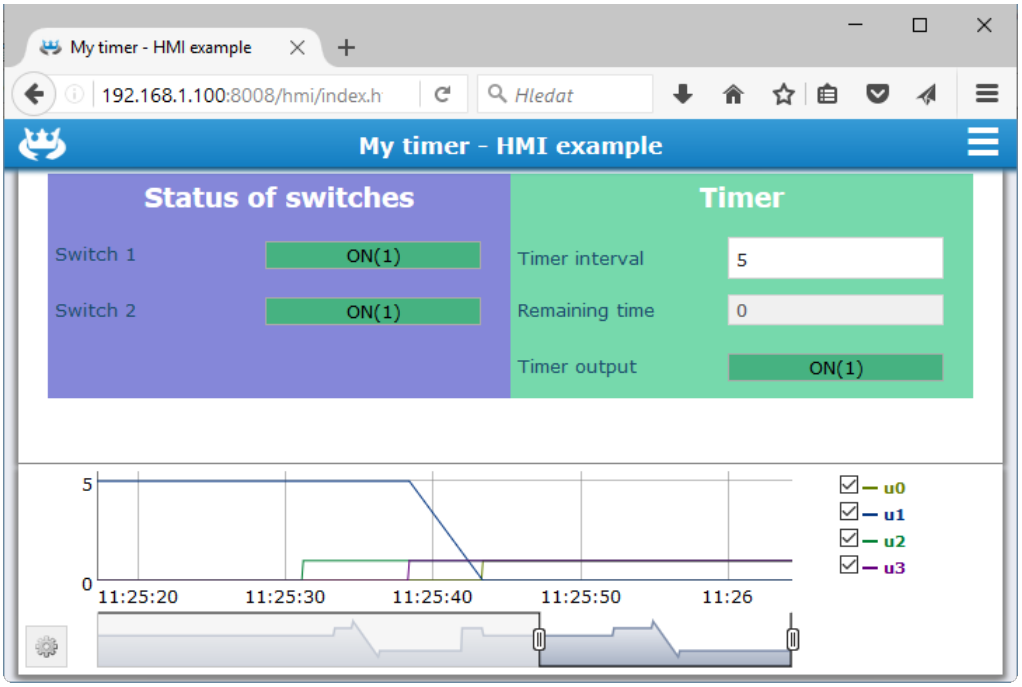


Once you compile the project again, you will see that the compile log contains more information. The HMI files were generated into the `hmi` subfolder and included in the resulting binary `myproject_exec.rex` file.



After you download the project to the target device, you can access the HMI via web browser. The address is `http://192.168.1.100:8008/hmi/index.html` (use the IP address of your Raspberry Pi). There are only indicators of the Boolean values (physical switches) in this simple example. Creating virtual buttons is also possible, see [4] for details.

The `index.html` file is generated from the source `index.hmi.js` file. The default port of the webserver (8008) can be changed in RexCore settings. See [2] for details.



See also the documentation of RexHMI Designer, which allows you to create graphical user interfaces

4.4 Ready for interaction with the outer world

You have learned the basic workflow for developing and running your algorithms using the REX Control 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.

Chapter 5

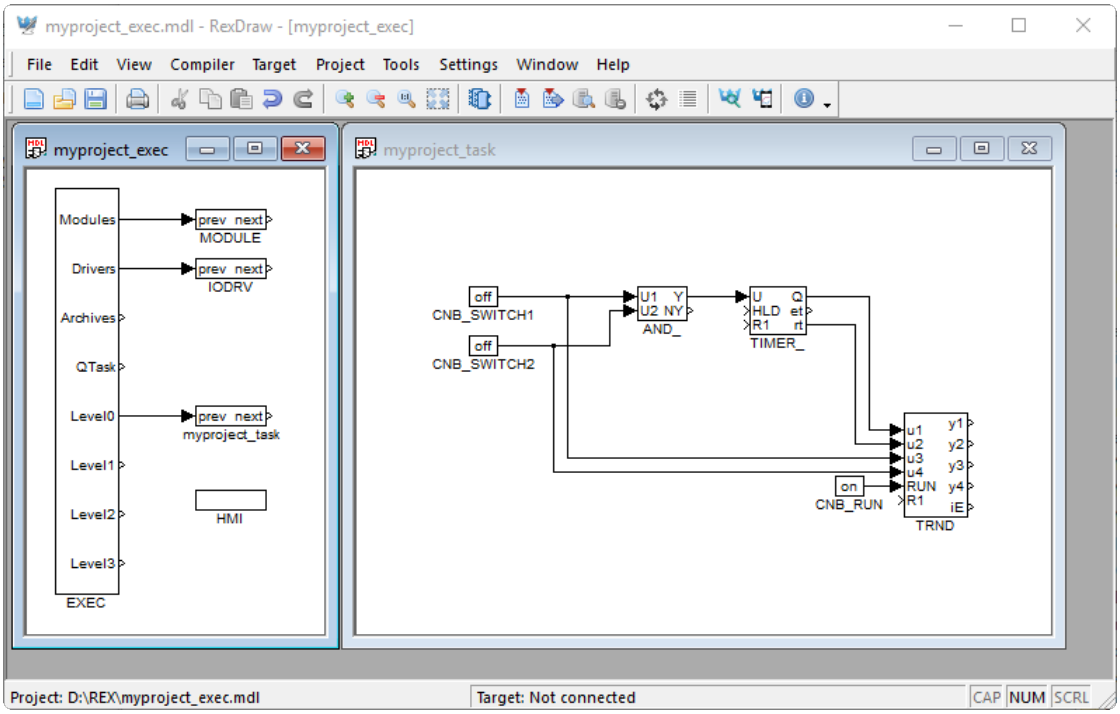
IO configuration for the Raspberry Pi platform

The previous chapter illustrated the process of creating a control algorithm in the REX Control 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).

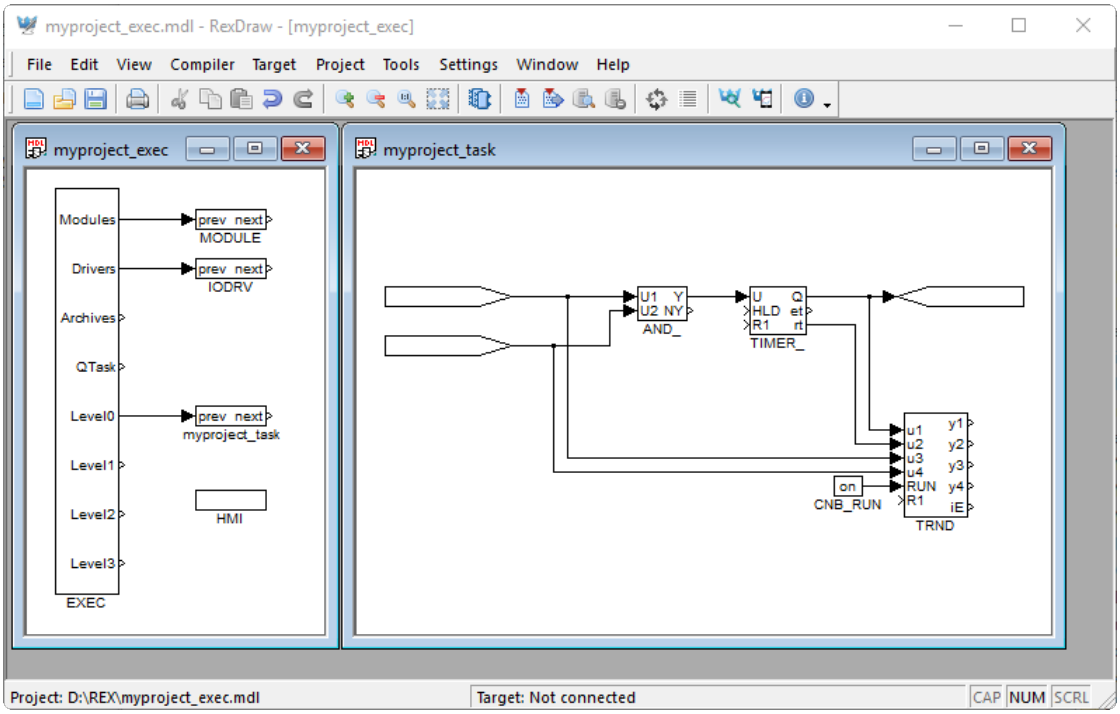
We will use the GPIO pins of the Raspberry Pi for interaction with the real world. The pins can be configured as inputs or outputs.

5.1 Adding inputs and outputs to the project

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_SWITCH2` 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?



5.2 Working with GPIO pins

5.2.1 Modifications in the project main file

Now we tell the compiler to use the Raspberry Pi I/O driver. This is what needs to be done:

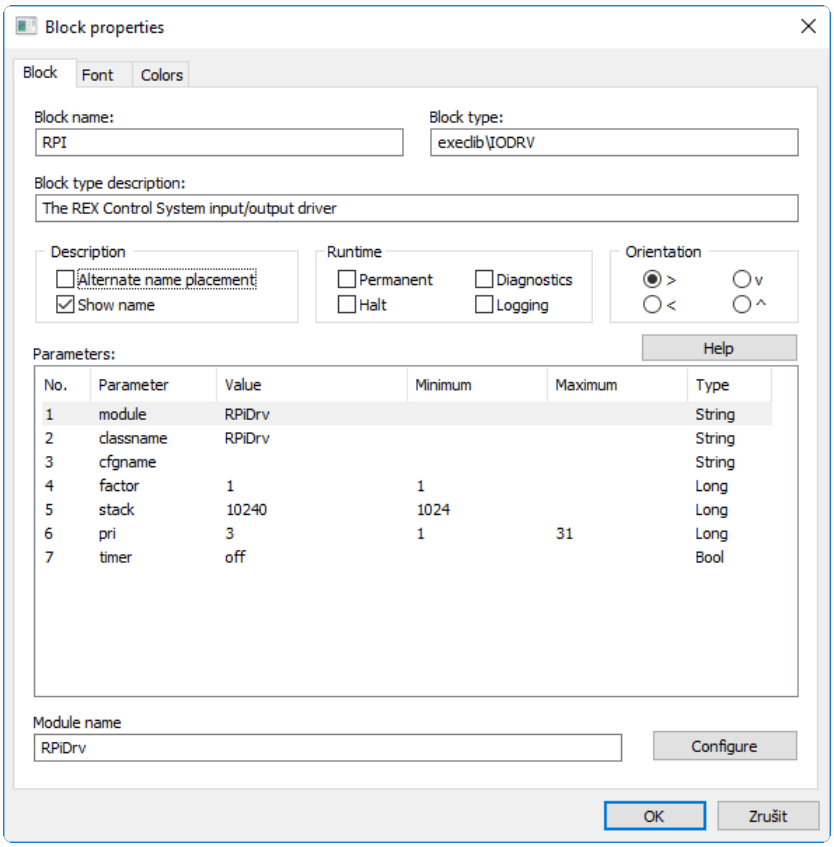
- Rename the `MODULE` block to `RPiDrv` – CASE SENSITIVE!

Further link the `IODRV` block with the `RPiDrv` module by setting

- `module` = `RPiDrv` – CASE SENSITIVE!

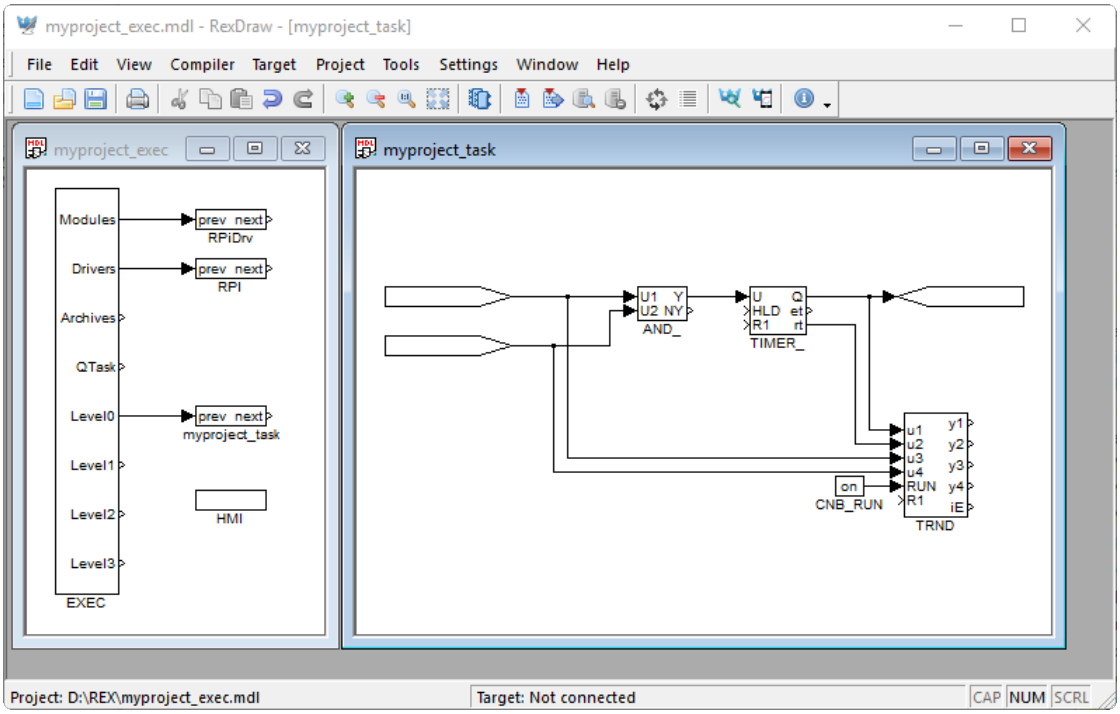
For bare Raspberry Pi set the following:

- `classname` = `RPiDrv` – CASE SENSITIVE!
- Leave the `cfgname` parameter blank.
- `factor` = `1`
- Leave the other parameters intact.



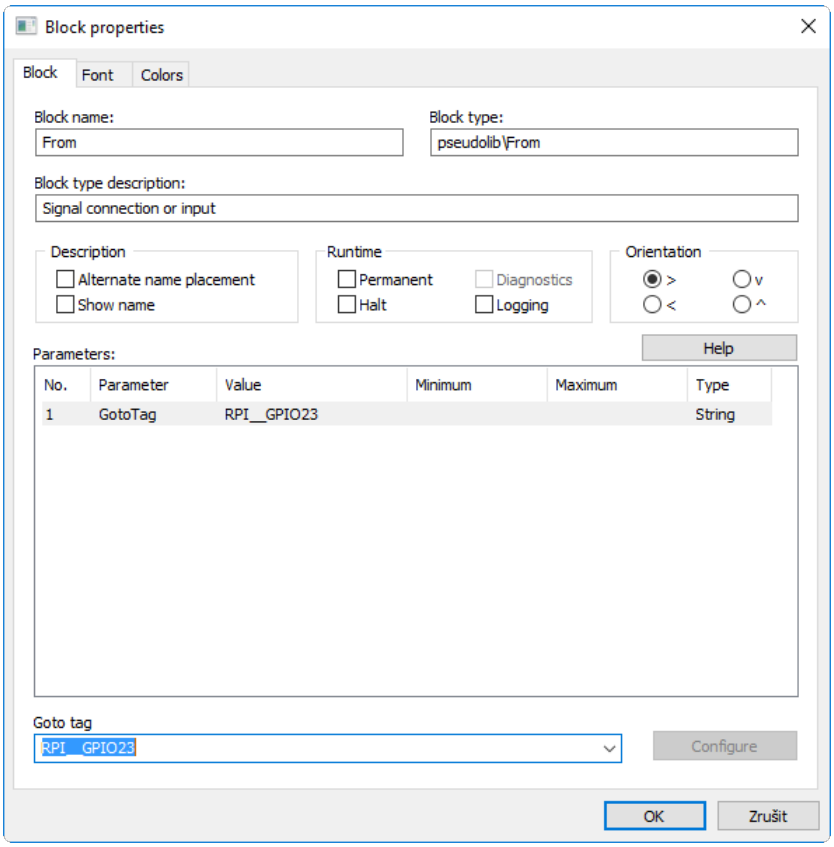
As the final step, rename the `IODRV` block to `RPI`, which will serve as a prefix for all I/O signals of this driver.

The executive of the REX Control System is configured, your project should look like this:



5.2.2 Modifications in the task

In the task double-click the input flag and set `GotoTag` = `RPI__GPIO23`. Note the RPI prefix and **two underscore characters**. The first physical switch will be connected to GPIO#23.



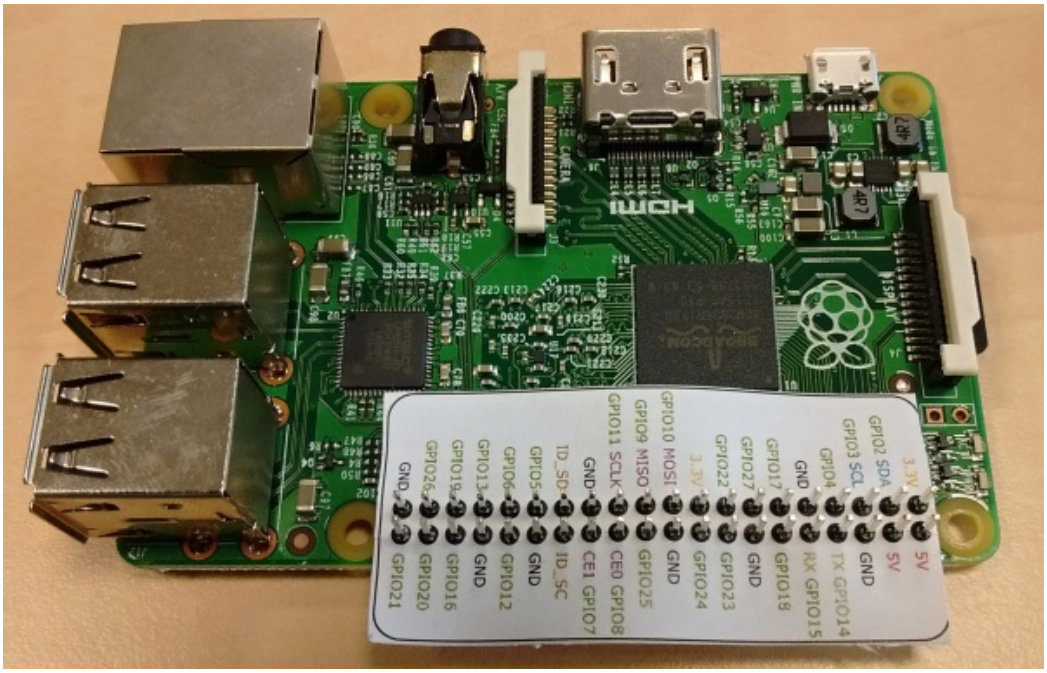
The second physical switch will be connected to GPIO#24 (`RPI__GPIO24`) and GPIO#25 will serve as the output signal (`RPI__GPIO25`).

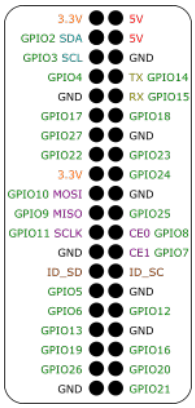
Similarly for other pins we could use the following flags:

- `Goto`, `RPI__GPIO22` – digital output 22
- `From`, `RPI__GPIO7U` – digital input 7 with internal pull-up resistor
- `From`, `RPI__GPIO8D` – digital input 8 with internal pull-down resistor
- `From`, `RPI__GPIO21` – digital input 21 without pull up/down resistor

A detailed description of the I/O driver for Raspberry Pi is available in a separate manual [5].

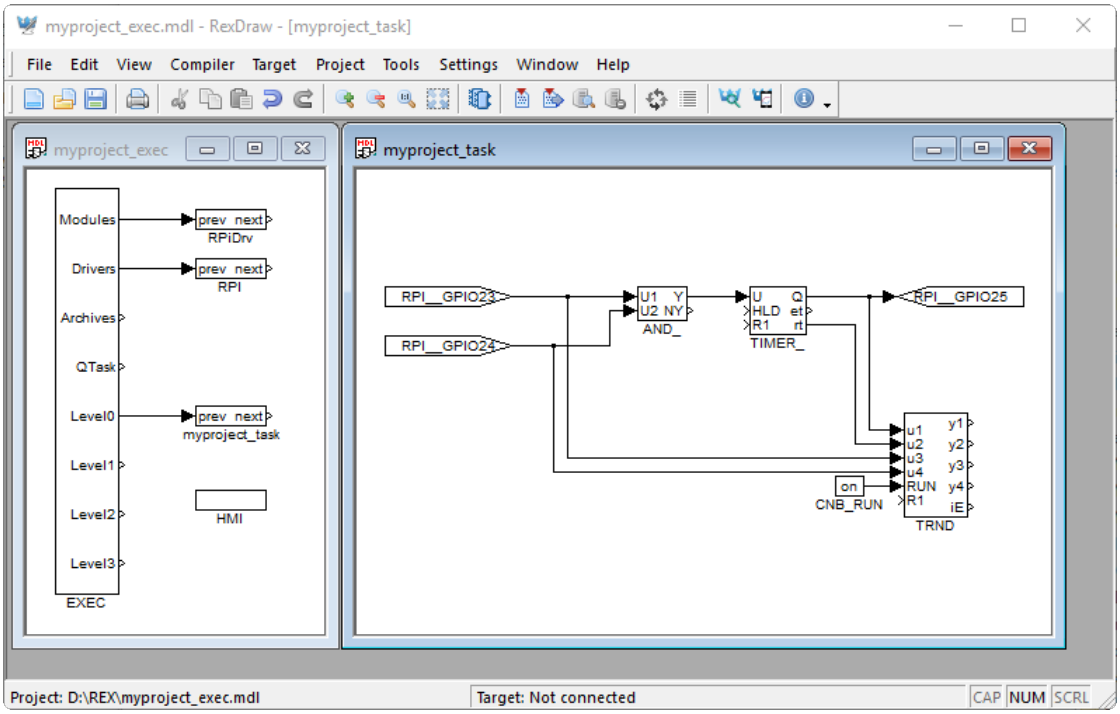
The pin numbering on Raspberry Pi B+, 2 and 3 is shown in the following pictures:



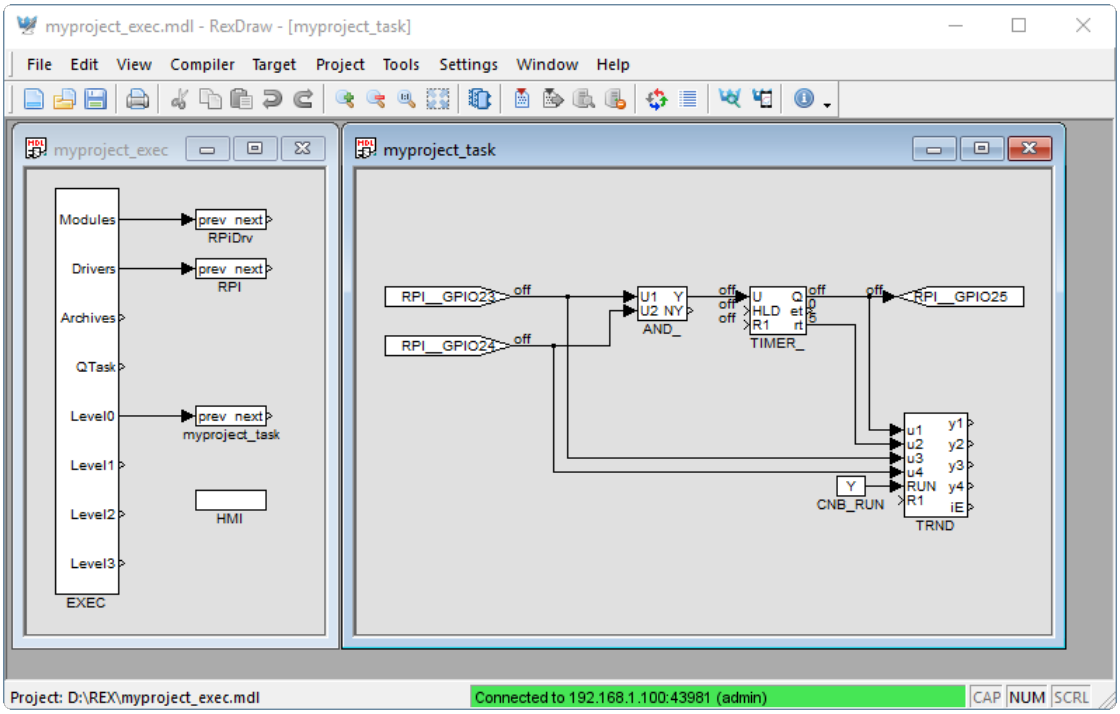


Visit the http://elinux.org/RPi_Low-level_peripherals webpage for detailed information about individual GPIO pins.

Your project should now look like this:



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



5.3 Additional information

5.3.1 Detailed description of the driver

A detailed description of the IO driver for Raspberry Pi is available in a separate manual [5].

5.3.2 Examples

Example projects and a set of all supported I/O flags are included in the installation package of the REX Control System development tools. In RexDraw, go to menu `Project` → `New project from template` and select one of the Raspberry Pi examples.

Chapter 6

Licensing of the REX Control System

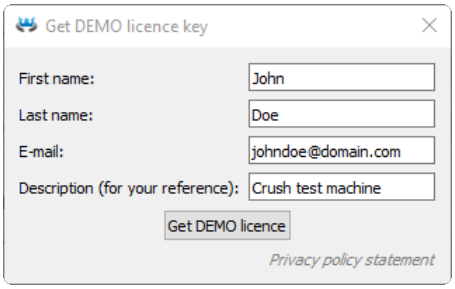
The licensing model of the REX Control System is quite simple:

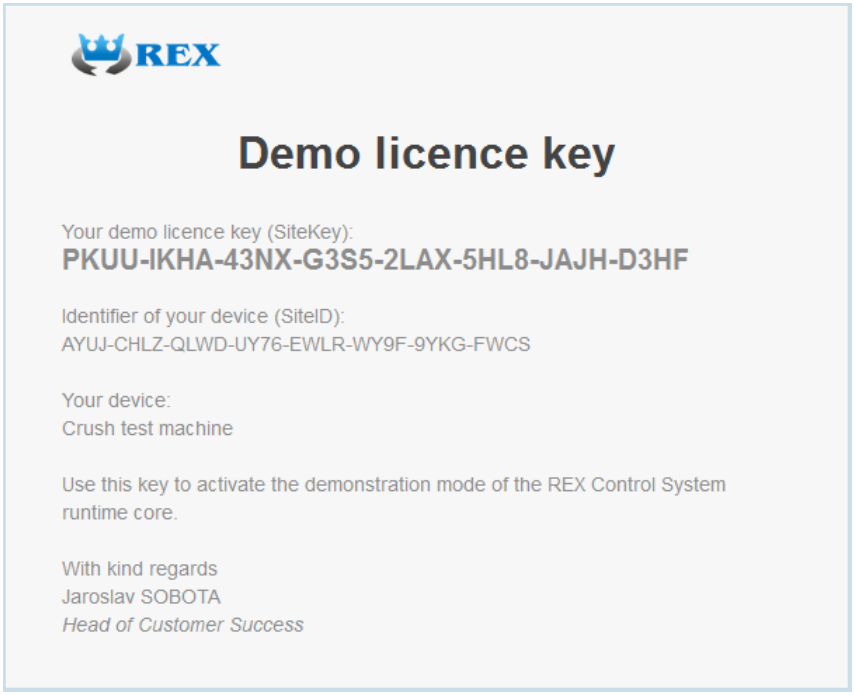
- The development tools are free to use, you can install it on as many computers as you want.
- The RexCore runtime module always needs a licence to run on your Raspberry Pi. There are DEMO licences available at no cost and there are permanent licences which you can purchase. Each Raspberry Pi needs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

6.1 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 licence, you are offered a chance to get a DEMO licence. Identify yourself and you'll receive a DEMO licence via e-mail (the so-called `SiteKey`).





REX

Demo licence key

Your demo licence key (SiteKey):
PKUU-IKHA-43NX-G3S5-2LAX-5HL8-JAJH-D3HF

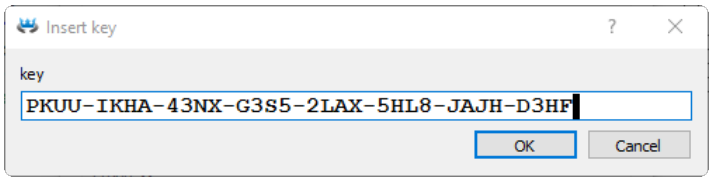
Identifier of your device (SiteID):
AYUJ-CHLZ-QLWD-UY76-EWLR-WY9F-9YKG-FWCS

Your device:
Crush test machine

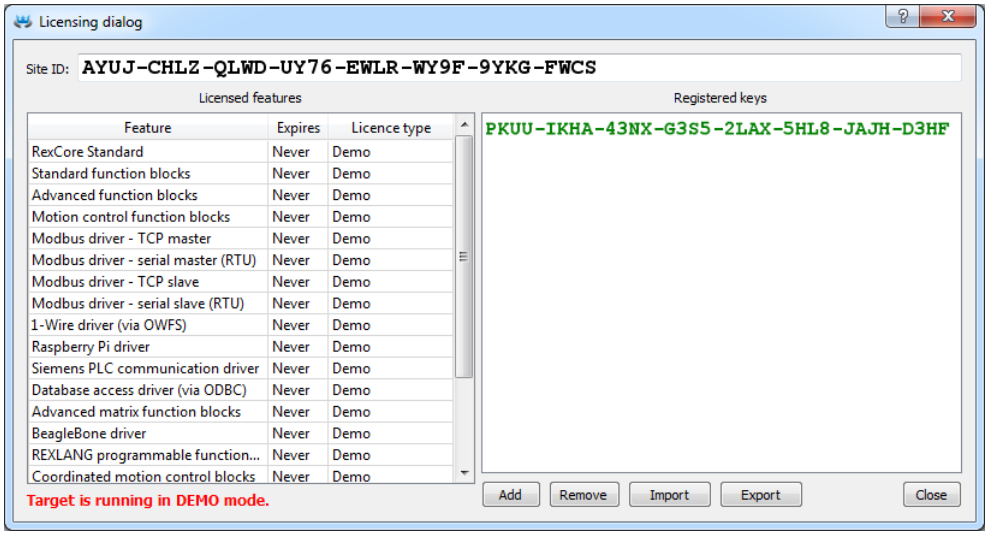
Use this key to activate the demonstration mode of the REX Control System runtime core.

With kind regards
Jaroslav SOBOTA
Head of Customer Success

Try downloading your algorithm once again and apply the DEMO licence (`SiteKey`).



Once applied, you will see all the modules.



Now you can run your algorithms on your Raspberry Pi.

Evaluation version of the RexCore runtime is functional for 2 hours. It is possible to run your algorithm on the Raspberry Pi but you cannot store it permanently, it resides only in the memory. 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.

6.2 Obtaining a permanent licence

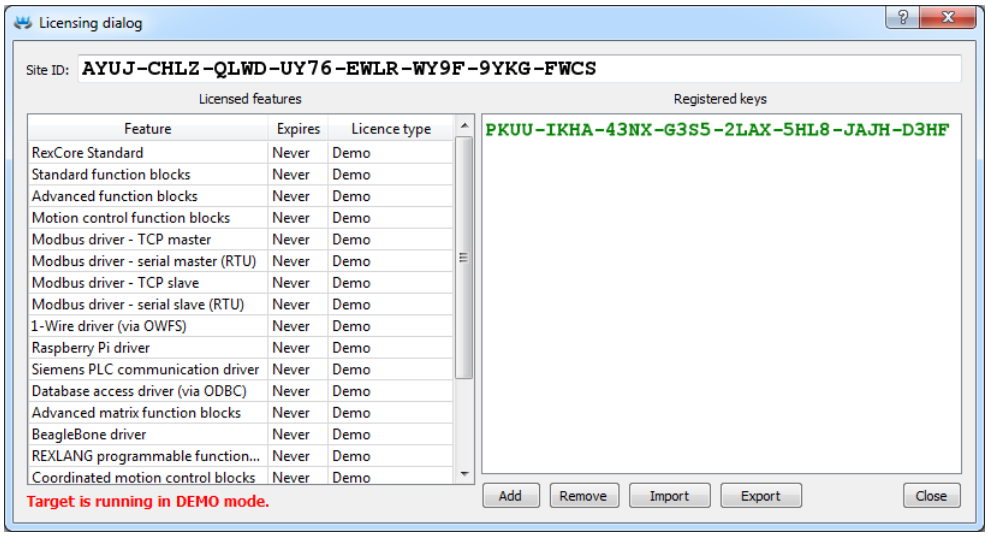
It is necessary to activate the RexCore runtime module and optional additional modules for permanent operation. This can be done using the licence, which you can obtain in an e-shop at

www.rexcontrols.com/e-shop

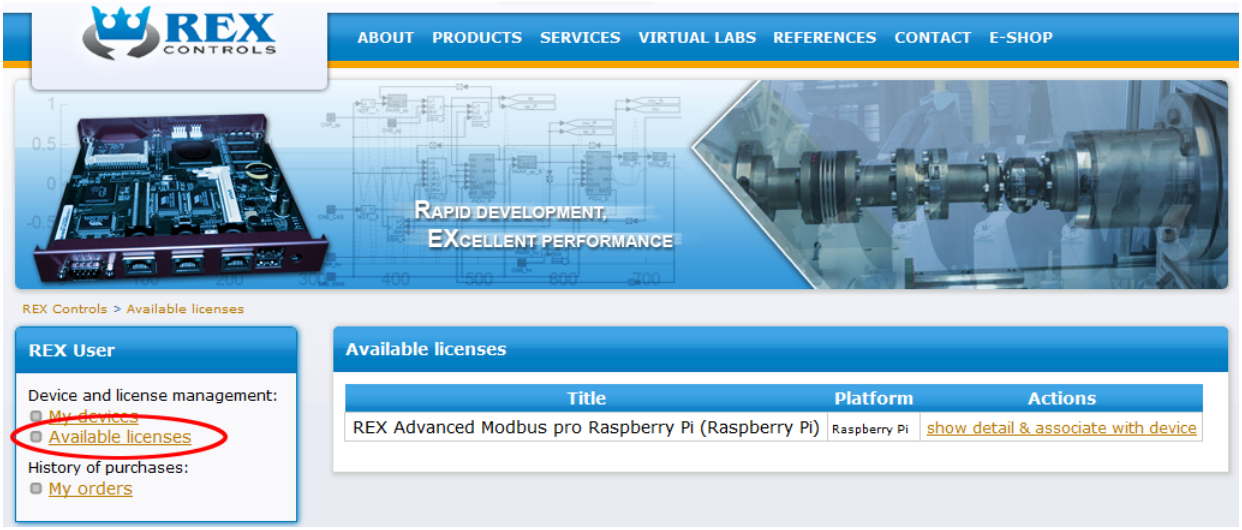
6.2.1 Activation of the permanent licence

The purchased licence must be associated with the hardware device.

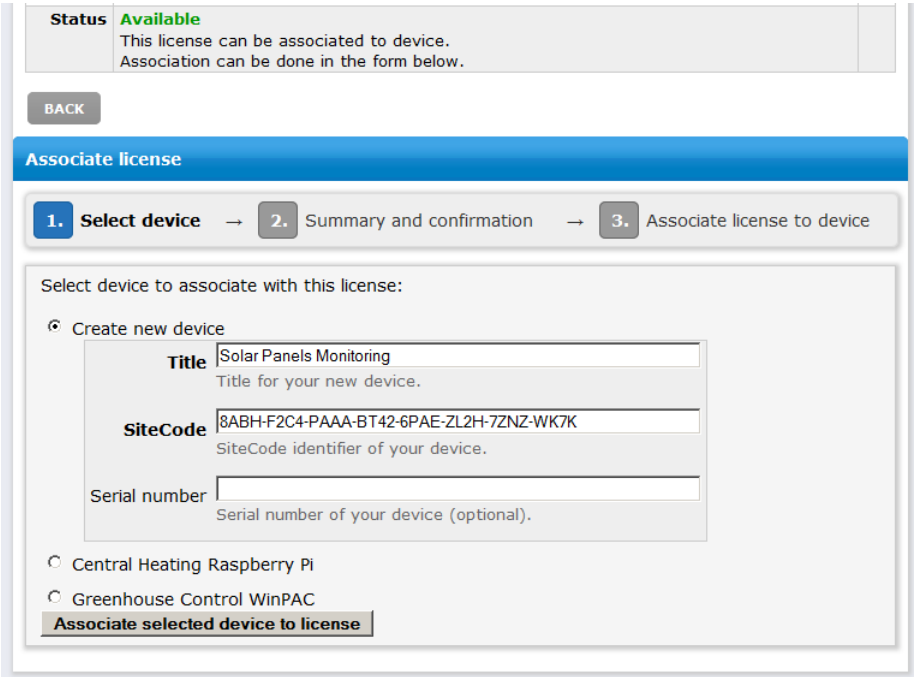
Each device running the RexCore runtime module is identified by the so-called `SiteID` tag. You can get it in RexDraw or RexView when connected to the device. After connecting to the target device, go to menu *Target* → *Licensing...*. A dialog pops up and you can copy the `SiteID` identifier.



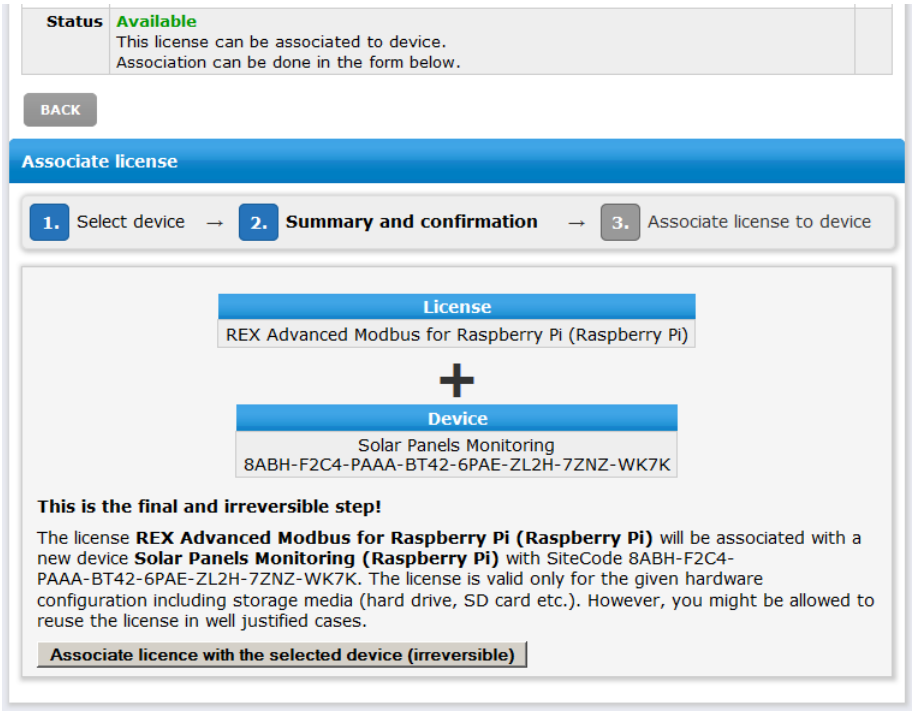
Afterwards, login to www.rexcontrols.com using your username and password and list the available licences which you possess.



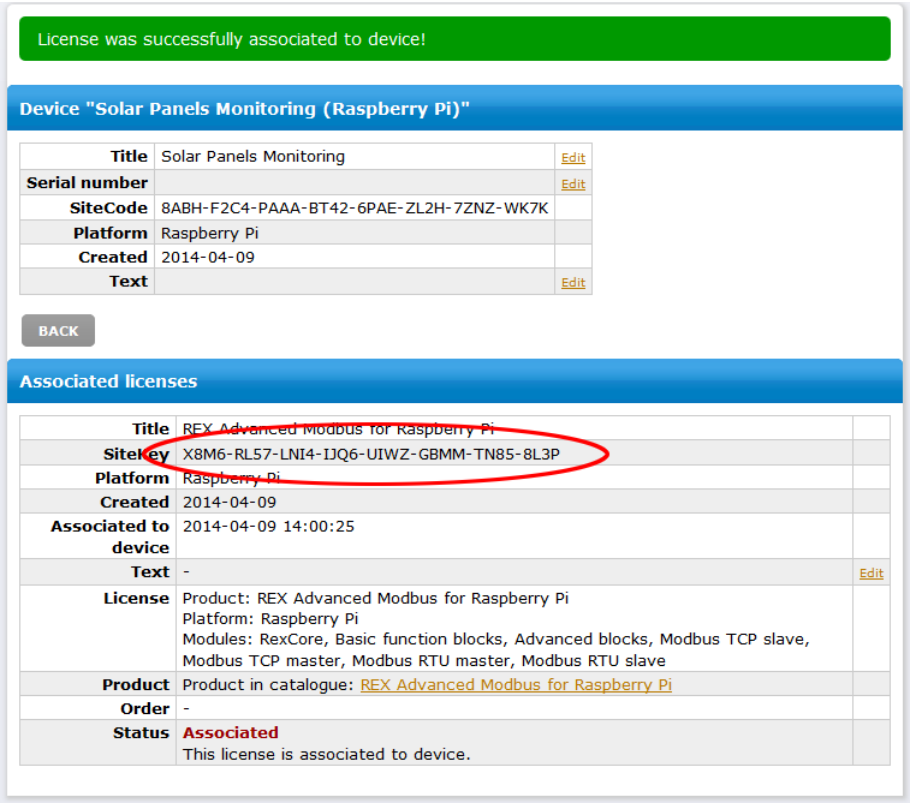
Use the `SiteID` identifier to associate the licence with the hardware device.



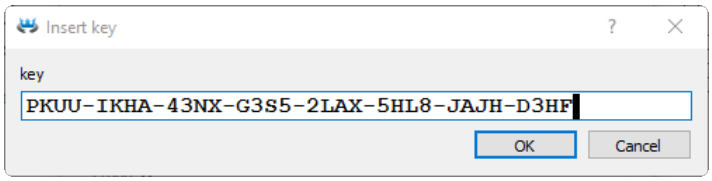
You are asked to confirm the association – this is the last and irreversible step.



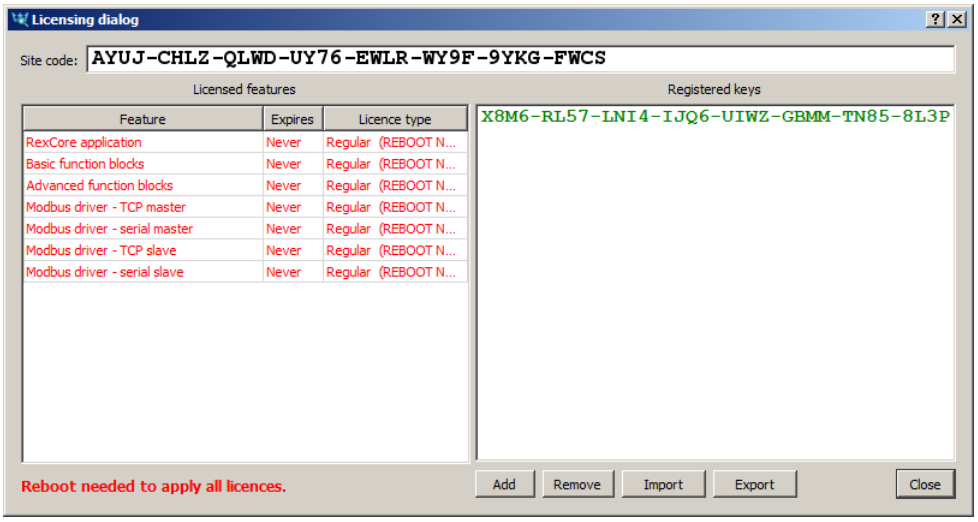
The so-called `SiteKey` activation key is generated upon associating the licence.



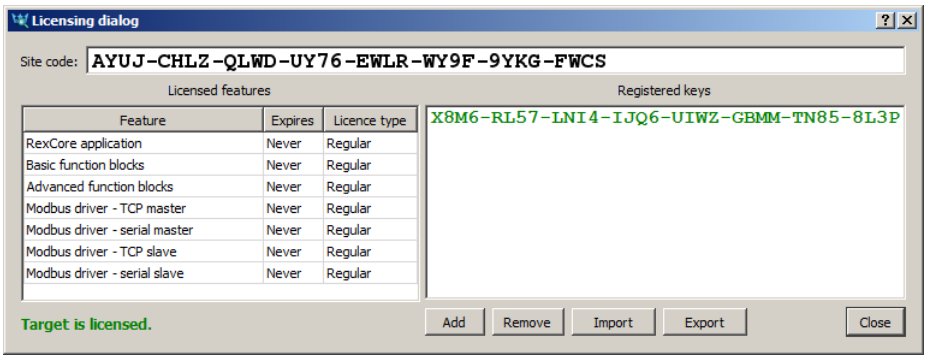
This key will allow permanent operation of the runtime core, but it must be applied to the target device. To do that, open the licensing dialog in RexDraw or RexView again and apply the `SiteKey` using the Add button.



If the key is valid, the activated modules appear. The Raspberry Pi must be restarted afterwards.



After reboot, check that the licence was applied correctly and that the RexCore runtime module no longer runs in demo mode.



Bibliography

[1] REX Controls s.r.o.. *Function blocks of the REX Control System – reference manual*, 2016.

[2] REX Controls s.r.o.. *RexCore – User manual*, 2016.

[3] REX Controls s.r.o.. *RexDraw – User manual*, 2016.

[4] REX Controls s.r.o.. *RexHMI – User manual*, 2016.

[5] REX Controls s.r.o.. *RPiDrv driver of the REX Control System for Raspberry Pi – user guide*, 2013.

Documentation reference number: 7567