Introduction

You can integrate xxter with Modbus. The xxter controller can act as a Modbus client (sometimes also called “master”), by connecting to a Modus server (sometimes also called “slave”). This means you can visualize Modbus data in the xxter app and use xxter to write to Modbus coils or registers, but you can also bridge KNX to Modbus and vice versa.

Overview

This manual contains the following sections:

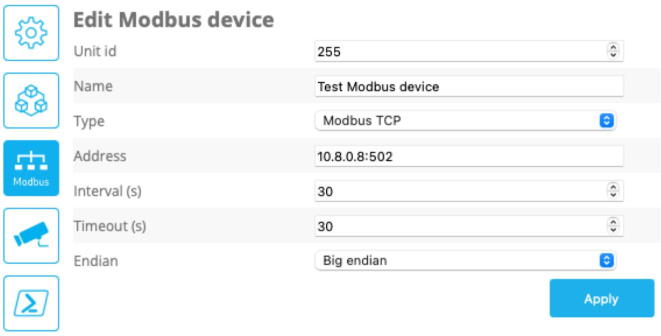
1) Setting up Modbus devices and objects
2) Adding Modbus objects to the xxter project
3) Enabling Modbus on the xxter controller

Setting up Modbus devices and objects

To let xxter connect to a Modbus server, a Modbus device needs to be added to the xxter project. Currently only Modbus TCP is supported.

Log in to *My xxter* as a professional, open your xxter project and select the Modbus option on the left, and add a device.

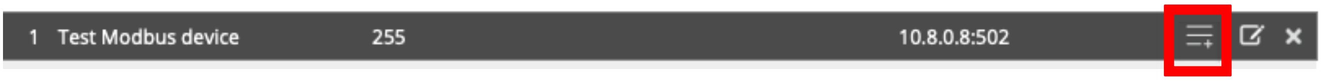
- Choose the device id (or unit id) of the Modbus device and provide a name for easy recognition
- Enter the IP address and port of the Modbus server
- Set up the interval at which the xxter device should request the latest values of the Modbus server. This must be a value between 30 and 3600 seconds (1 hour).
- Select the Endian that is used by your Modbus server. This is required to correctly address the register values.



Edit Modbus device

Unit id	255
Name	Test Modbus device
Type	Modbus TCP
Address	10.8.0.8:502
Interval (s)	30
Timeout (s)	30
Endian	Big endian

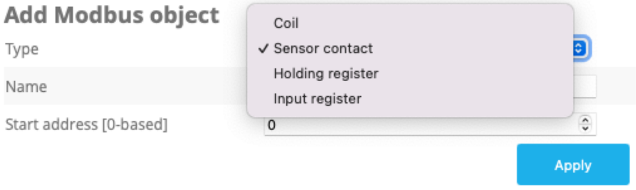
Once you have added the Modbus device, press the *add object* icon, to add the Modbus objects:



1 Test Modbus device 255 10.8.0.8:502

There are 4 types of objects:

Coil:	1 bit, allows read and write
Sensor contact:	1 bit, only read
Holding register:	number, allows read and write
Input register:	number, only read



Add Modbus object

Type: ✓ Sensor contact

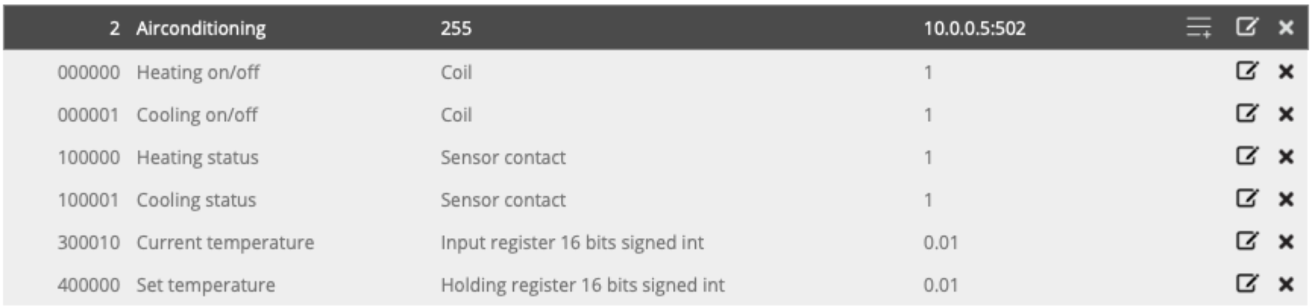
Name

Start address [0-based]: 0

For every object, the start address needs to be provided. This is the relative address for that type, which means xxter will automatically add the base address for the corresponding type, for instance 100000 for sensor contacts. The type and address details should be provided by the Modbus server’s specifications.

Input and holding registers also require the data format that is used, for instance “16 bit signed int” or “32 bit float”. Additionally, the resolution can be provided. For instance, “0.001” if you want xxter to use Kilowatts instead of Watts, depending on the Modbus server’s specifications.

Once you have added all the objects you intend to use, you will have an overview like this:

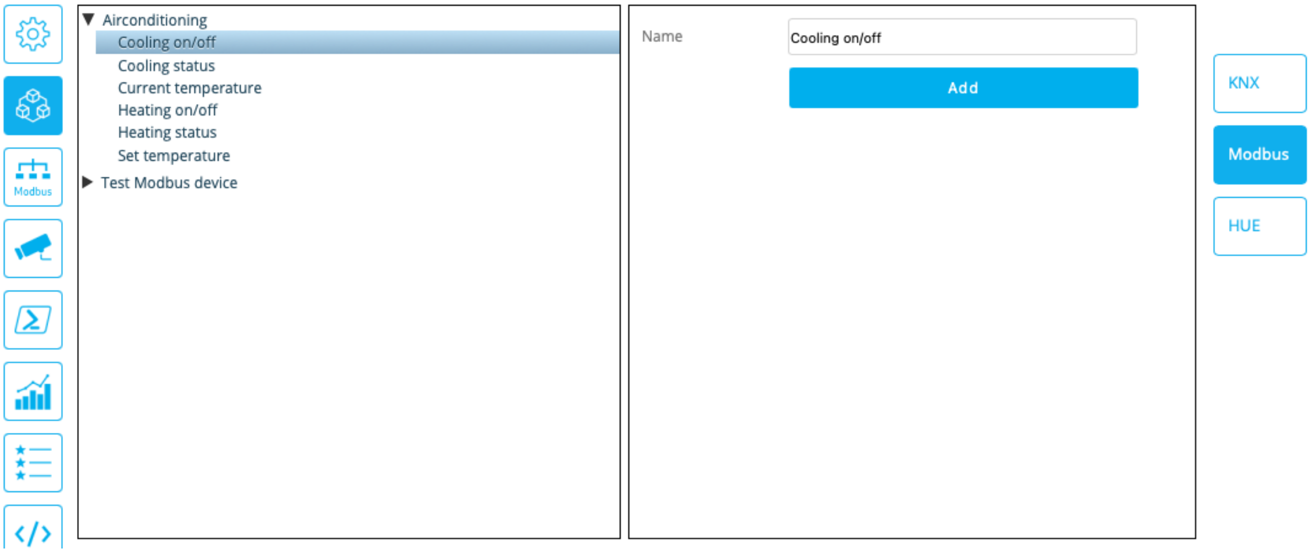


2	Airconditioning	255	10.0.0.5:502
000000	Heating on/off	Coil	1
000001	Cooling on/off	Coil	1
100000	Heating status	Sensor contact	1
100001	Cooling status	Sensor contact	1
300010	Current temperature	Input register 16 bits signed int	0.01
400000	Set temperature	Holding register 16 bits signed int	0.01

Adding Modbus objects to the xxter project

Now all the Modbus objects are defined, they can be added as components to the xxter project.

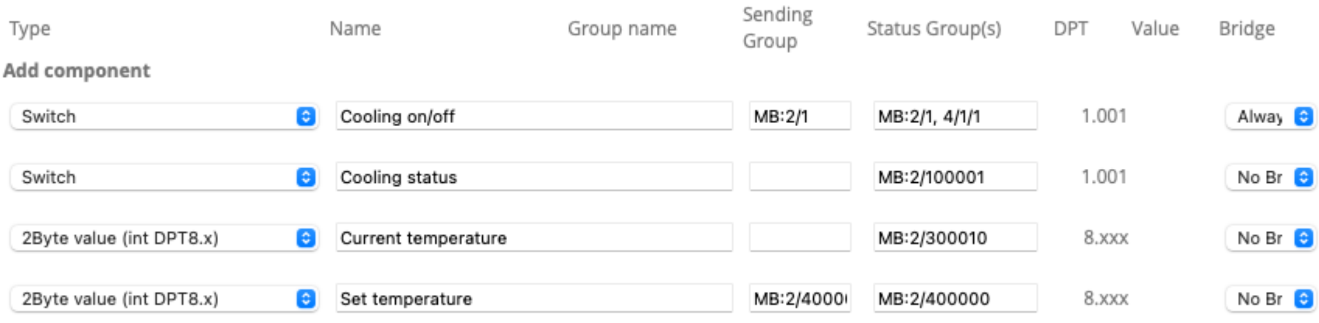
Select the components option, and press *Edit*. In the top window, click the Modbus button on the right.



- ▼ Airconditioning
  - Cooling on/off
  - Cooling status
  - Current temperature
  - Heating on/off
  - Heating status
  - Set temperature
- ▶ Test Modbus device

Name: Cooling on/off

You can add components, by selecting a Modbus object and press *Add*.



Type	Name	Group name	Sending Group	Status Group(s)	DPT	Value	Bridge
Add component							
Switch	Cooling on/off		MB:2/1	MB:2/1, 4/1/1	1.001		Alway
Switch	Cooling status			MB:2/100001	1.001		No Br
2Byte value (int DPT8.x)	Current temperature			MB:2/300010	8.xxx		No Br
2Byte value (int DPT8.x)	Set temperature		MB:2/4000	MB:2/400000	8.xxx		No Br

Please note that sensor contacts and input registers have no *sending group*, since they are read only values.

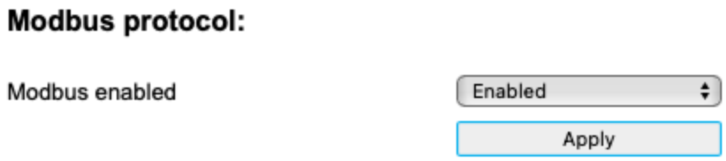
You can also add additional KNX status groups, or add a KNX sending group and enable a *bridge* option, to have xxter bridge the Modbus value to KNX or vice versa.

Once you have added all the Modbus objects as components to your xxter project, don't forget to press save.

Enabling Modbus on the xxter controller

Now the xxter project configuration is complete, all that remains is to enable Modbus on the device.

Log into the xxter controller, go to the *Protocols* menu and enable the Modbus protocol:



Modbus protocol:

Modbus enabled: Enabled

After this, don’t forget to press the *Load configuration* button, to load the latest changes on the device.

With Modbus enabled and the project loaded, the xxter controller will continually poll the values of the Modbus objects you have configured, at the interval you have set up. Of course, when you change the value of either a coil or a holding register, this is sent immediately to the Modbus server.