# Leaf Controller

We call Leaf Controller to the Leaf-Linux (BananaPi-R2 or ExpressoBin-Ultra or ModBerry) plus the Leaf-ControllinoMega:



ModBerry

A picture containing electronics, circuit

Description automatically generated

EspressoBin-Ultra

A close-up of a computer chip

Description automatically generated with low confidence

BananaPi-R2

A close-up of a computer chip

Description automatically generated with medium confidence

Leaf-ControllinoMega

Leaf-Linux

**Leaf-Linux**

BananaPi-R2

EspressoBin-Ultra

ModBerry

Leaf-ControllinoMega

Ethernet

USB

# Development Environment

For the Leaf-Linux Development Environment, we’ll use PyCharm running on a Windows laptop.

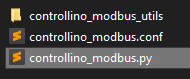
The project is found in **\FEleafcontroller\Leaf-Linux**.

Text

Description automatically generated

The modules pandas, pymodbus, twisted need to be installed into PyCharm.

The Linux System runs a Python program, whose files are:



Text

Description automatically generated

Controllino\_modbus.py contains the main, controllino\_modbus.conf is the configuration file, and the controllino\_modbus\_utils directory contains device specific definitions and code.

For the Leaf-Controllino Development Environment, we’ll use Atmel Studio.

The Atmel Studio project solution can be found in **\FEleafcontroller\Leaf-Controllino\Controllino**.

A screenshot of a computer

Description automatically generated with medium confidence

# qModMaster communication with Controllino

qModMaster (unzip qModMaster-Win64-exe-0.5.3-beta.zip, executable qModMaster.exe, https://sourceforge.net/projects/qmodmaster/) can be used to dialog to the Controllino through the Ethernet interface.

1. Configure the Windows laptop to have address 192.168.2.10

Graphical user interface, application

Description automatically generated

1. Connect a direct Ethernet cable between the Windows laptop and the Controllino
2. The Controllino has address 192.168.2.3 port 502, so in QModMaster select “Options | Modbus TCP…” and set

Graphical user interface, application

Description automatically generated

1. Use qModMaster to write Outputs and read Inputs, using Unit ID 42, and the proper Modbus Function Code.

qModMaster Start address is 1, whereas in the LeafController Linux-Python is 0!

Graphical user interface, text, application, email

Description automatically generated

# qModMaster communication with the Leaf-Linux

In order to communicate directly with the Leaf-Linux:

1. Configure the Windows laptop to have address 192.168.2.10

Graphical user interface, application

Description automatically generated

1. Graphical user interface, application

   Description automatically generatedPerform a loopback connection to the Windows laptop on address 192.168.2.10 port 1502, so in QModMaster select “Options | Modbus TCP…” and set
2. Use qModMaster to read Input Registers, using Unit ID 42, and the proper Modbus Function Code.

qModMaster Start address is 1, whereas in the LeafController Linux-Python is 0!

So, to read the variables in controllino\_controllino.py:

['aggTripAlarmCore', 0x182A, 0xAB, 'ro', 'v'], # address 6186

Graphical user interface, text, application, email

Description automatically generated ['aggTripAlarmCube', 0x182B, 0xBA, 'ro', 'v'] # address 6187

# WireShark

WireShark can be used to capture the traffic between the Leaf-Linux and the Leaf-Controllino.

The filter ‘**ip.addr == 192.168.2.3 and modbus’** is useful to catch only the Modbus traffic that flows through the Controllino, IP address 192.168.2.3.

Table

Description automatically generated

Allowing to inspect the Ethernet / Modbus TCP:

A picture containing application

Description automatically generated

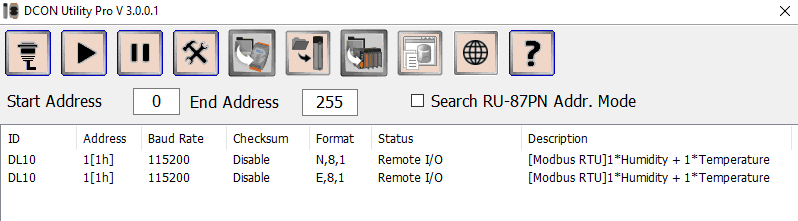
# Connecting the DL-10 Temperature and Humidity Sensor

Connect the DL-10 to the Windows laptop using a RS485 to USB converter. It’s necessary to connect the power of DL-10 to a power supply of 12VDC and the RS485 between the DL-10 the RS485 to USB converter.



Use the DL-10 software, **DCON\_Utility\_Pro\_PC DCON\_Utility\_Pro.exe**, to scan for the DL-10.

Be sure that you are using the right COM port by pressing the “COM Port” toolbar button, and then initiate the scan by pressing the “Start Search”.



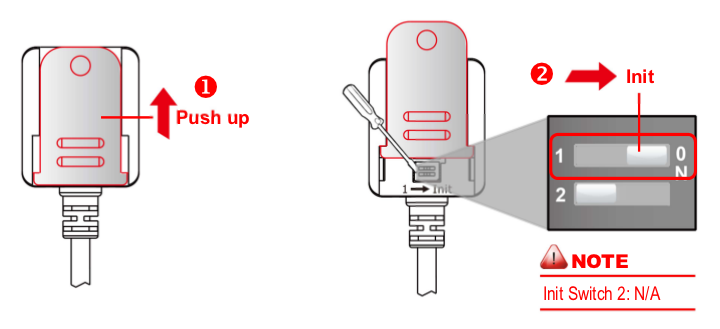
This proofs that the DL-10 is working.

## Changing DL-10 configuration

Now the DL-10 must have its configurations changed to be integrated

The Controllino uses 19200,N,8,1 and the DL-10 should have the Address 5. So, these need to be changed.

1. Power off the DL-10.
2. Change the switch on the back of the DL-10 to the Init position which allows to change the configuration.



1. Power the DL-10.
2. Perform a new scan.

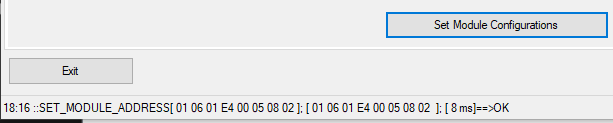
Graphical user interface, application, Word

Description automatically generatedThe module should be found again:

1. Graphical user interface, text, application

   Description automatically generatedJust double-click on the line that contains the DL10 N,8,1 and change the configuration to:

Once “Set Module Configurations” is pressed a message at the bottom of the window should appeared:



Text

Description automatically generatedFollowed by another window:

1. Disconnect the DL-10 power.
2. Change the switch to position 1 (to the left): Diagram

   Description automatically generated
3. Change the com port to scan only on 19200, by default only 115200 and 9600 are used.

Graphical user interface, application

Description automatically generated

1. Repower the module and rescan.

Graphical user interface, text, application

Description automatically generatedNow the module should be found at address 5 with the communications parameters of 19200,N,8,1.

Double-click on the module and check if the temperature and humidity are read and make sense.

Graphical user interface, text

Description automatically generated

1. Exit the DCON Utility and as a second proof use qModMaster to verify these readings.

qModMaster needs to be used with the following parameters:

Graphical user interface, text, application, email

Description automatically generated

The 2 values read are by order, the Humidity, and the Temperature, with a factor of x100.

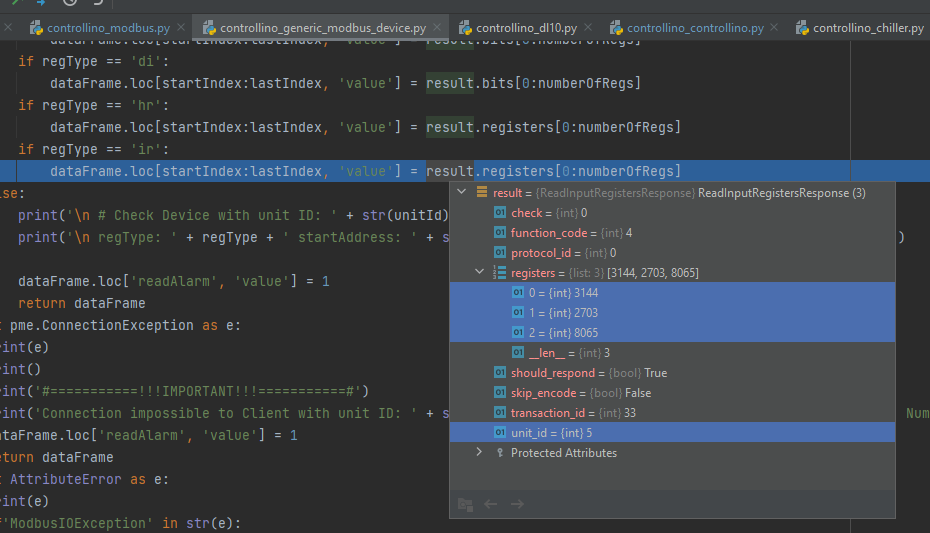
As above, 40.61% of Humidity and 24.88C of Temperature.

1. Connect the DL-10 to the Controllino RS485 Modbus network.

To verify that this is working, open the Linux-Python project and place a breakpoint in file controllino\_generic\_modbus.py on the line shown below in blue.

When the program hits the breakpoint verify that is a read of the DL-10, by checking variable results, field unit\_id == 5 and then inspect the registers field.

0 is the Humidity, 1 is the Temperature in C, 2 is the Temperature in F.

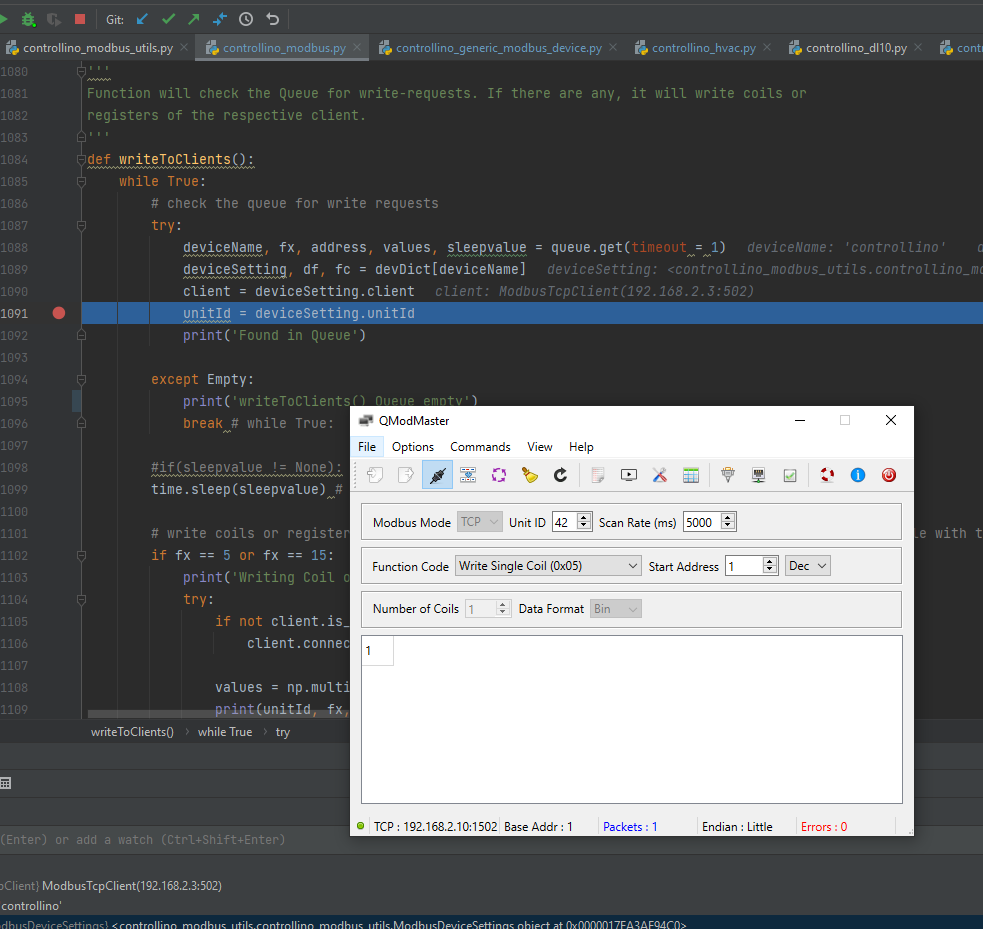


## Writing a Controllino digital Output

The qModMaster can be used to write an output of the Controllino (coil)

Run the Controllino\_modbus.py and establish a connection to 192.168.2.10 (that’s the laptop where the Python and qModMaster are running, so a loopback) and using the parameters shown below execute the write.

This will hit the python function writeToClients().



## Writing multiple Controllino Outputs

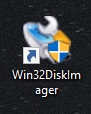
Multiple outputs can be written as well, up to 16. Function writeToClients is also used.

Graphical user interface, text, application, email

Description automatically generated

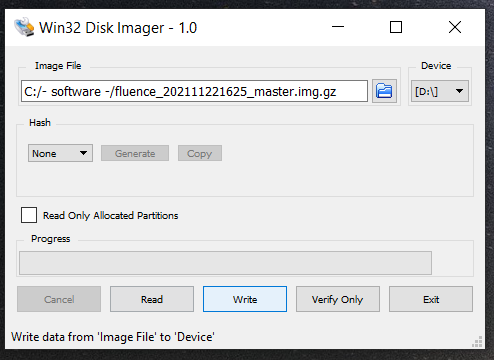
# Creating a Linux SDcard for BananaPi R2

For this purpose, Win32 Disk Imager software can be used, <https://sourceforge.net/projects/win32diskimager/>



The file to be used is **fluence\_202111221625\_master.img.gz**.

Use the Write command to perform a total initialisation of the SD card with the Linux system:



Perform then a Verify.

