Reading the EM340 with USB Converter and Python

Converter

Any USB to RS485 converter is suitable for the readout. This document describes the OEM USB dongle based on the CH340G chip.

Chip & Driver

The converter is based on the CH340G from WCH (Jiangsu Qinheng Co., Ltd). It's like FTDI but another manufacturer and therefore needs different drivers.

The actual driver is platform (OS) dependent: See https://kig.re/2014/12/31/how-to-use-arduino-nano-mini-pro-with-CH340G-on-mac-osx-yosemite.html

Wiring

> Do not touch or modify any wiring on the EM340 without ensuring that device is completely unplugged from the power outlet! Do not touch any terminals or wires while the power is connected. The voltages of 230V AC can result in death or serious injuries by electrical shock.

The A & B notation in RS485 is somewhat confusing since different manufacturers use different schemes for negative and positive. Generally, only the polarities have to match, therefore connect "plus to plus" and "minus to minus"

Converter	EM340
B (D-)	A- (9)
A (D+)	B+ (8)

In Addition, a terminator resistance is required on either end of the signal line. This can be achieved by connecting a wire between terminal 8 and 7 on the EM340.

Communication

The communication with the EM340 is done with the Modbus protocol. There are sufficient libraries available to use Modbus without having to understand the protocol in depht.

Two possible options are:

- Python with the minimalmodbus library
- NodeRed with the Nodred Modbus Extension

While there are only readouts for specific values in this document, the principle remains the same and all other values can be read by changing the address of the register.

Python

Preconditions: Installed Python & PIP installed Minimalmodbus

For an example Python script (tested with python 3.6)

This script can be found in attachment **em340.py**

em340.py

```
#!/usr/bin/env python
import minimalmodbus
import serial


#Adapt this device to your OS and actual device path
em340 = minimalmodbus.Instrument('/dev/cu.wchusbserial1410', 1) # port
name, slave address (in decimal)

em340.serial.port          # this is the serial port name
em340.serial.baudrate = 9600   # Baud
em340.serial.bytesize = 8
em340.serial.parity   = serial.PARITY_NONE
em340.serial.stopbits = 1
em340.serial.timeout  = 0.05   # seconds
em340.mode = minimalmodbus.MODE_RTU   # rtu or ascii mode

## Read value from EM340
temperature = em340.read_register(288, 10) # Registernumber, number of
decimals
print(temperature)

## Write a config to the EM340
#SOME_VALUE = 95
#em340.write_register(24, SOME_VALUE, 1) # Registernumber, value, number
of decimals for storage
```

NodeRed

Preconditions: Installed NodeRed, Installed Modbus extension, wired converter

The following NodeRed Flow reads the basic values (Current, Voltage, Frequence, Power) and parses it in a JSON (tested with Raspberry Pi 3)

See attachment **EM340NoderRed.json**