# Wagon Controller

## 1 Hardware

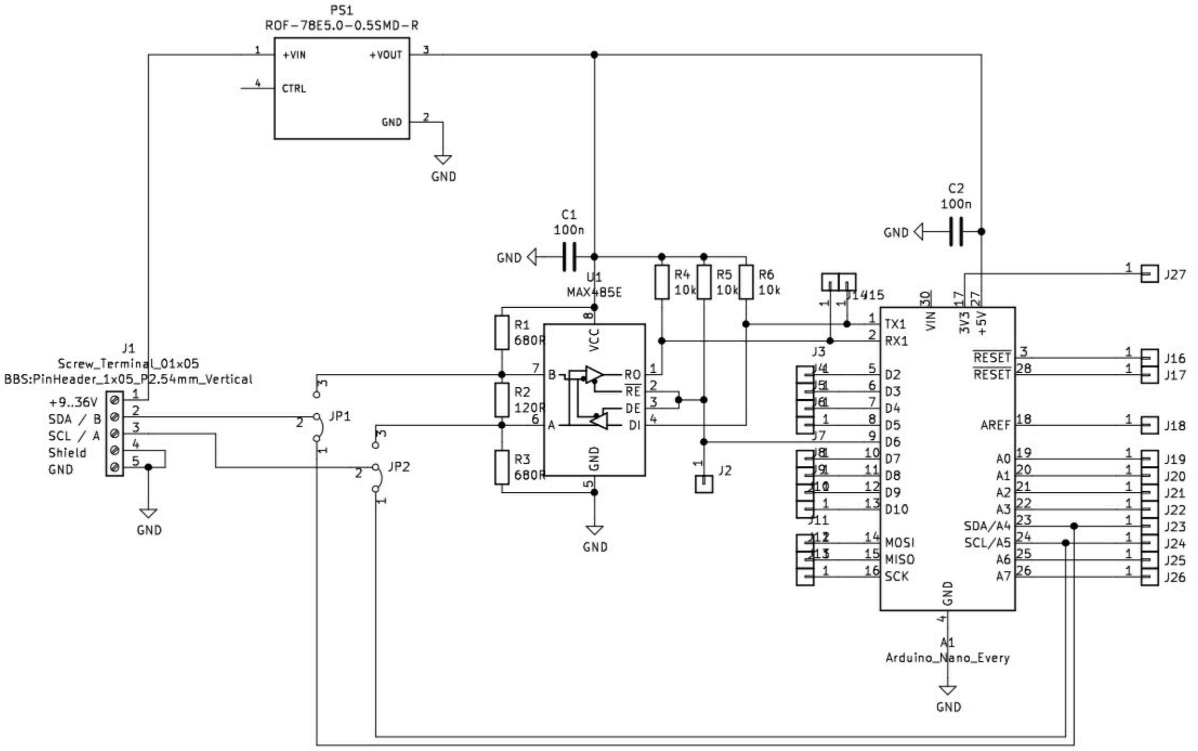
This small device can be used

a) as a replacement of the PCF8575 16 Bit I$^2$C bus expander. In this mode, the RS485-Transceiver will not be used. The socket for this chip can be left open, R1..R6 are not needed.

b) as a node on a RS485-Network. Based on this physical layer, a ModBus-Client will be placed. ModBus is a very famous protocol, widely used in industry. Libraries for Arduino are also available. It allows a much more sophisticated control of the wagon, than a simple bus expander can provide.

The following description covers both scenarios.

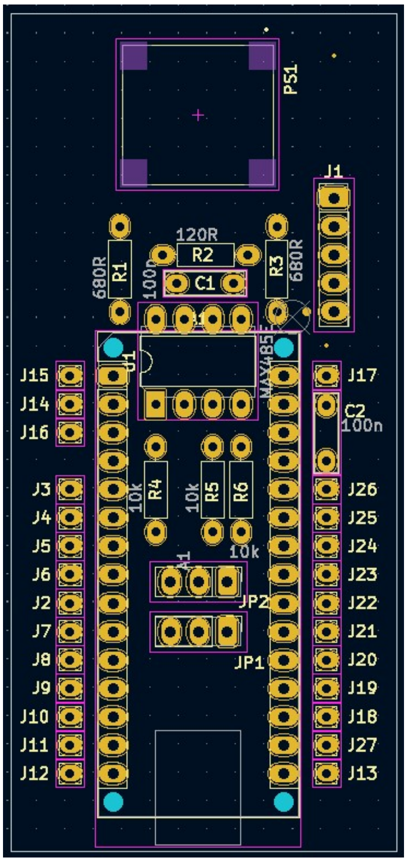
### 1.1 Schematic



## 1.2 Board

To allow a small footprint, most of the parts were placed underneath the controller board. We do not use SMD packages to keep soldering simple.

The dimensions of the pcb are:  35 x 75 mm



| | |
|---|---|
| FS1 | A Step-Down-Converter with a 5V / 500mA output and a wide input-range from 9V..36V.  Using a DC-DC-Converter per node will eliminate noise coming along the main power supply from the loco. One power supply will fit all needs. |
| JP1, JP2 | These two jumpers connects terminal 2/3 to either the RS485 device or the SDL/SCL lines of the Controller. The jumpers must only be changed, if the firmware at the node changes.<br>1-2 = I$^2$C mode<br>2-3 = RS485 mode |
| Address | To spare I/O-pins, the address of the device must be set via source code.<br>In RS485-mode, it will be possible to change the address via ModBus protocol. |
| R1,R3 | They provide a save potential at the lines A and B, if no sender is active. These resistors are only for safe operation and must not be placed several times. Place them at the first node in the chain. |
| R2 | Terminating resistor. Must correspond with the impedance of the wires. 120 Ohm will fit most needs.<br>Needed only at the beginning and the end of the bus. |

Schematic as well as the board were made with KiCAD, with is available for free.

KiCad EDA - Schematic Capture & PCB Design Software

# 2 Software

## 2.1 PCF8575 replacement

The PCF8575 is a 16Bit port expander for the $I^2C$-Bus. A maximum of 8 devices can be used per $I^2C$ bus. More devices requires the use of an $I^2C$ multiplexer.

Because of the lack of internal configuration registers (like MCP23016), the device is quite simple. After the initial write sequence transporting the device address, every subsequent byte written will be placed in the output latch A bevor B. The operation is module 2. Writing 6 bytes will have the same result then writing only the last 2 bytes.

The firmware to emulate the PCF8575 is based on the WIRE library, which comes with Arduino. The idea is to be able to replace a PCF8575 without making any changes to the $I^2C$-Master. Just setting the correct address at the slave. The address must be set in the source code and can not be changed by external settings like the PCF8575 can.

The controller board used here comes with real push-pull-outputs, where the PCF8575 uses only low-side switches.

The Controller used is an Arduino Nano Every. It is cheap, has a small footprint and supports 2 serial ports by Hardware, so one can be used for serial monitor and the other is used for RS485 device transceiver.

The Arduino-Pins are assigned to the bits of the PCF8575 and the PCB connectors as follows:

| PCF8575 Register A | Arduino Pin-Nr | Connector @ PCB | Connector @ PCB | Arduino Pin-Nr | PCF8575 Register B |
|---|---|---|---|---|---|
| Bit 0 | 2 | J3 (D2) | J10 (D10) | 10 | Bit 0 |
| Bit 1 | 3 | J4 (D3) | J11 (D11,MOSI) | 11 | Bit 1 |
| Bit 2 | 4 | J5 (D4) | J12 (D12,MISO) | 12 | Bit 2 |
| Bit 3 | 5 | J6 (D5) | J13 (D13,SCK)<br>BuiltIn-LED | 13 | Bit 3 |
| Bit 4 | 6 | J2 (D6) | J19 (A0) | A0 | Bit 4 |
| Bit 5 | 7 | J7 (D7) | J20 (A1) | A1 | Bit 5 |
| Bit 6 | 8 | J8 (D8) | J21( A2) | A2 | Bit 6 |
| Bit 7 | 9 | J9 (D9) | J22 (A3) | A3 | Bit 7 |
| | | | | | |
| | | J14 (RX1) | J23 (A4/SDA) | A4 | |
| | | J15 (TX1) | J24 (A5/SCL) | A5 | |
| | | | J25 (A6) | A6 | |
| | | | J26 (A7) | A7 | |
| | | | | | |
| | | | J18 (AREF) | | |
| | | | J27 (+3V3) | | |
| | | | J13 (SCK) | | |
| | | | | | |

The assignment is important for wiring the device. It can be easily changed in the firmware. The Pin numbers are collected in two arrays, named `portA` and `portB`

```
uint8_t portA[] = {2, 3, 4, 5, 6, 7, 8, 9};
uint8_t portB[] = {10, 11, 12, 13, A0, A1, A2, A3};
```

## 2.2 RS485-Node

RS485 is a differential 2-wire bus system widely used in industry. It is the physical layer for a protocol named ModBus.

tbc

# 3 Sources

PCF8575 datasheet
https://www.ti.com/lit/gpn/PCF8575

MCP 23016 datasheet
https://ww1.microchip.com/downloads/en/DeviceDoc/20090C.pdf

RS485 Bus transceiver
https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf

Arduino Nano Every
https://docs.arduino.cc/hardware/nano-every

Used library for PCF8575
https://github.com/xreef/PCF8575_library

Arduino-Client for MQTT
https://github.com/knolleary/pubsubclient

JSON-Library for Arduino
https://github.com/bblanchon/ArduinoJson