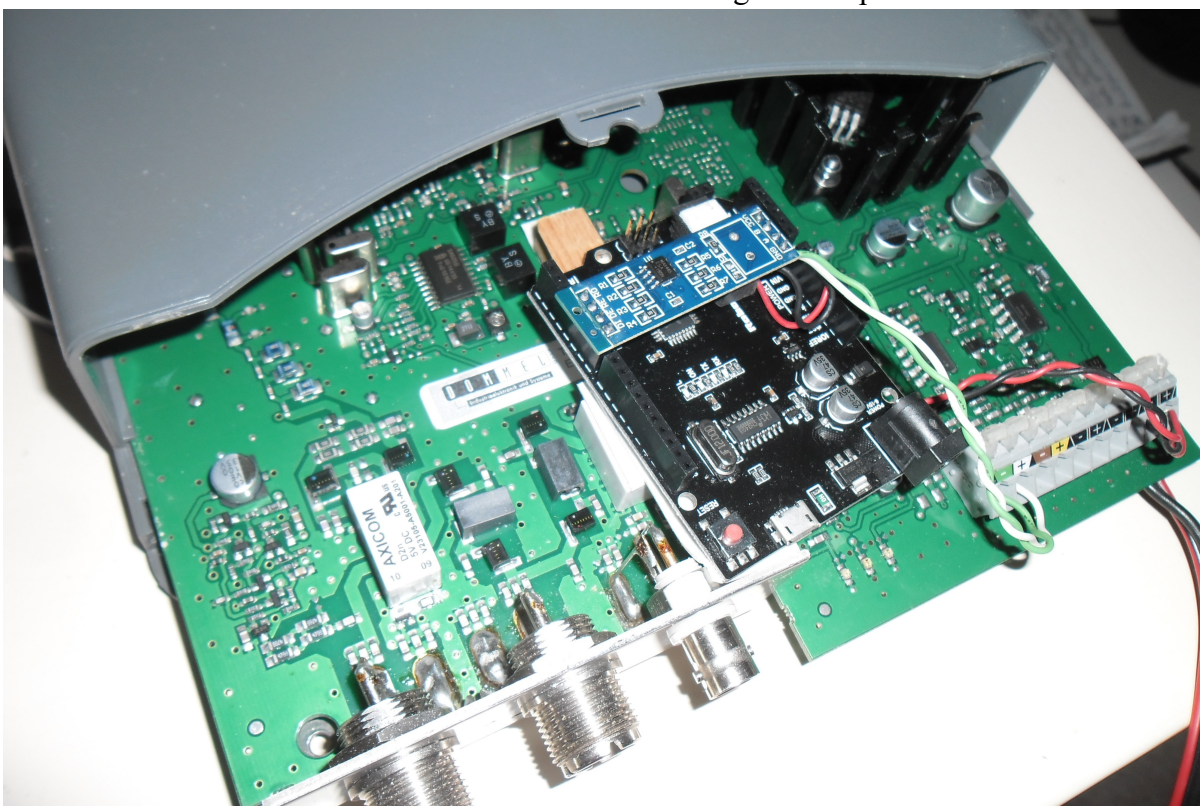
Using NMEA data via USB on Computer

Purpose
In my case for using a Raymarine AIS receiver as a data source for the OpenCPN navigation/plotter program.
Also for inspection of a NMEA data stream with a Serial Monitor program. (Standard program with the Arduino IDE) this connection can be useful.

Used hardware
There are other possibilities but because I still had the parts lying around I used an Arduino Uno type microcontroller with a MAX485 module.

The implementation
The Arduino Uno has a USB connection to communicate with a computer: This connection is used both to program the Arduino and to be able to use or inspect data while running the program.
The NMEA data stream is a differential system with two data lines A and B. They are not grounded to be more resistant to interference. If line A is 5 volts, line B is 0 volts and vice versa. The MAX 485 module converts the NMEA data stream into TTL with 5 volts relative to ground. The received NMEA AIS data stream therefore runs via the MAX485 to the digital data ports of the Arduino:



The Raymarine AIS receiver with the (black) clone of an Arduino-Uno. The MAX 485 module is connected with 4 header pins to the digital inputs D4 to D7. The NMEA data stream runs via the twisted white green connection of the AIS receiver to the A and B input of the MAX485 module. The 5V for the MAX is drained from the Arduino. The USB port for the connection to the laptop is located at the bottom left of the black Arduino printed circuit board just to the right of the red reset button. The prints are fixed with double-sided foam tape.

The Program

```
// Connections
// Arduino  to  MAX 485
//        6              DE  Data Enable       (SSerialTXControl)
//        5              RE  Receive Enable    (LedPin)
//        4              RO  Receive Out       (SSerialRX)
//        7              DI  Data In           (SSerialTX)
//        5V             5V                    red
//        GND            GND                   black
// AIS            CABLE         MAX485
// NMEA -         white         A
// NMEA +         green         B
// 12 V +         red           removed socket
// 12 V -         black         removed socket

#include <SoftwareSerial.h>
#define SSerialRX 4            // Receive Out
#define REPin 5                // Receive Enable
#define SSerialTXControl 6  // Data Enable
#define SSerialTX 7            // Data In
#define RS485Receive LOW
#define RS485Transmit HIGH
String  inputString = "";
SoftwareSerial RS485Serial(SSerialRX, SSerialTX);

void setup()
{
  inputString = "";                    // start with empty line
  inputString.reserve(200);
  Serial.begin(38400);                 // data rate for communication with PC
  pinMode(REPin, OUTPUT);
  pinMode(SSerialTXControl, OUTPUT);
  digitalWrite(SSerialTXControl, RS485Receive);
  RS485Serial.begin(38400);            // set data rate for communication MAX485 with AIS
}
void loop()
{
  if(RS485Serial.available())
  {
    char charReceived = (char)RS485Serial.read();
    inputString += charReceived;
    if(charReceived == '\n')           // end of line
    {
      Serial.println(inputString);     // print line to USB/PC
      Serial.println("");              // blanco line
      inputString = "";                // clear inputstring
    }
  }
}
```

In the example above Software Serial is used. It is also possible to use the first two hardware data ports D0 and D1 which are connected to the USB by default. The MAX 485 module connection in this case:

RO	*Receive Out*	*to*	*D0 (Hardware Serial connection)*
DI	*Data In*	*to*	*D1*
RE	*Receive Enable*		*LOW in receive mode to Signal Ground*
DE	*Data Enable*		*LOW in receive mode to Signal Ground*
5V			*5V*
GND			*GND*

The MAX485 Module

A	*NMEA+*
B	*NMEA-*

The program: Just an empty loop() works

To use the USB data stream on the computer, the correct COM port and baud rate must be set.
Within the OpenCPN program, this can be done via the menu Tools > Options > Connections

The OpenCPN program can receive data via multiple USB inputs. For example, a GPS mouse position via one USB port and the AIS data stream via another USB input. In this case, it is therefore not necessary to first merge two NMEA data streams into one data stream. (as is done with the NMEA data recorder)

Jeroen Droogh bootprojecten@gmail.com