NMEA sentences recording

Purpose

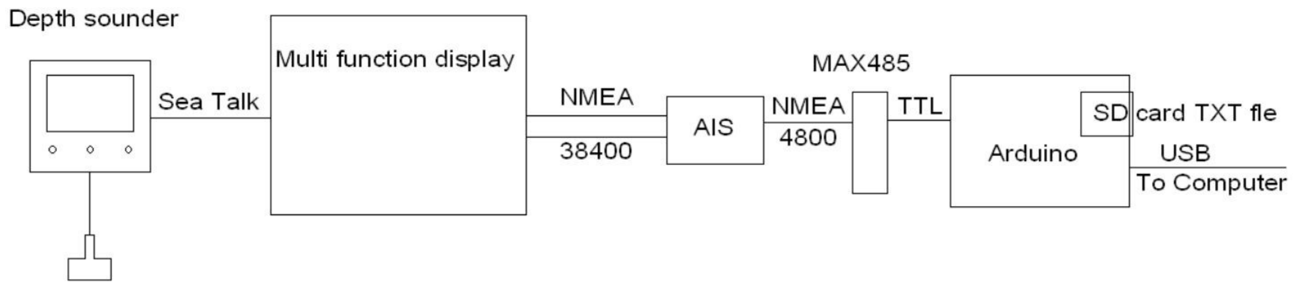
Storing NMEA sentences in a TXT file. For example for a water depth survey the depth ($SDDBT sentence) and GPS position ($GPRMC sentence) are stored together in order to plot the depth in the right position.

Data sources used

For the position:	The GPS position from or via an Em-trak B-100 AIS (A Raymarine GPS with Raymarine AIS receiver also works)
For water depth:	A Raymarine depth sounder coupled to a Multi Function Display (MFD).

Interfacing

The Em-trak AIS communicates with the MFD at a baud rate of 38400 baud.
The depth sounder is linked to the MFD via the Seatalk link.



When the MFD and AIS function correctly, the depth information from the sounder is delivered to the MFD via the SeaTalk connection and from the MFD via NMEA 38400 baud to the AIS.
The AIS will combine the data with ts own NMEA data into one NMEA data-stream. With the other NMEA sentences of the AIS, this depth of information is recorded in one NMEA data stream. This is accessible via the 38400 baud and the 4800 baud port of the AIS. The outgoing NMEA 4800 data stream is used by the recorder.
Apart from the position data, the AIS in this case is not used for its specific AIS data but mainly for merging the two NMEA data streams into one NMEA data stream.

Data recording

The data recorder reads the NMEA sentences and stores them on an SD card. Only the date, position and depth are important for our purpose, so only the necessary sentences are kept. (GPRMC and SDDBT) Writing on the card takes time and space, so the fewer unused sentences are written down, the more time there is for the used strings. When processing the data, at a later time, the position and water depth are removed from the strings.

File format and name

The data is automatically saved in a text file with the file name MMDDUUmm.txt in which:
MM month of the year
DD day of MM
UU hours of UTC
mm minutes from UU (The file name is limited to 8 characters)
The data recorder doesn't have its own clock or memory. Date and time are therefore extracted from a received GPRMC sentence at the start of the program in order to create the file name. It is therefore important that the data flows correctly when the data recorder is switched on.

Example of a recording

10231110.TXT (start of recording October 23 at 11:10 UTC)

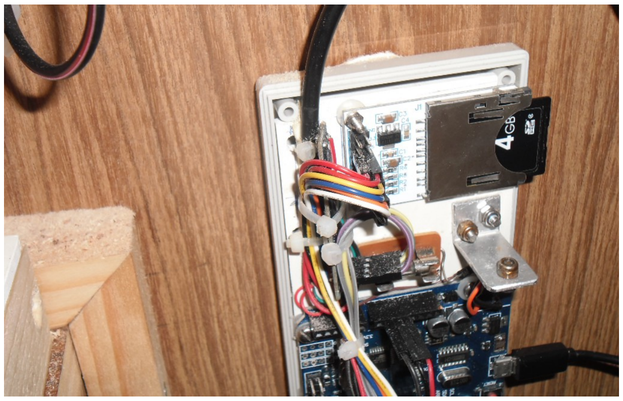
```
$GPRMC,111018.00,A,5253.79623,N,00538.73222,E,0.030,,231021,,,A*78

$GPRMC,111019.00,A,5253.79624,N,00538.73222,E,0.022,,231021,,,A*7D

$SDDBT,5.8,f,1.7,M,1.0,F*0C

$GPRMC,111020.00,A,5253.79624,N,00538.73220,E,0.018,,231021,,,A*7C
```

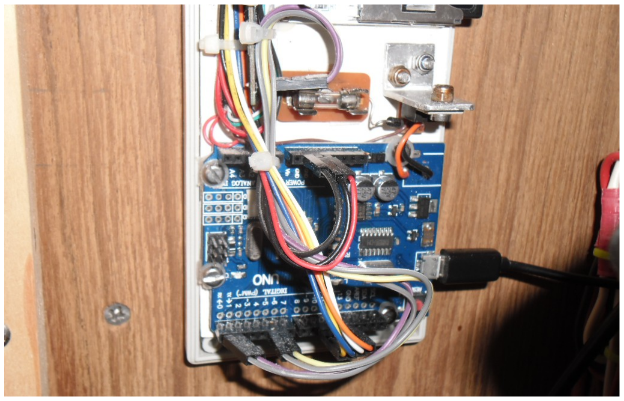
...... and so on

The automatically generated filenames don't have a year designation. That is why I occasionally manually put the recordings in a file folder with the relevant year. In the example above, the date is 231021 (see the GPRMC string) and the recording has been moved to the file folder 2021.

BUG: We have corrected our Raymarine sounder for the depth of the transducer (offset) so that the screen indicates the depth measured from the surface of the water. The sentence identifier DBT in the previous example, in which the water depth is 1.7 m from the water surface, should now actually be called DBS (Depth Below Surface) instead of DBT (Depth below Transducer) because of the correction.
Whether other Raymarine software versions also don't rename the identifier after correction is not known to me. It is therefore wise, before using the depth information, to make a manual sounding to see what value is actually given and included.

NOTE: The values of the depths in the SDDBT sentence have only one decimal in the official NMEA format. When processing the data at a later date, I therefore use the depth in feet and convert it into centimeters myself. This gives me greater accuracy. (1/10 feet is about 3 cm and 1/10 meter is 10 cm)





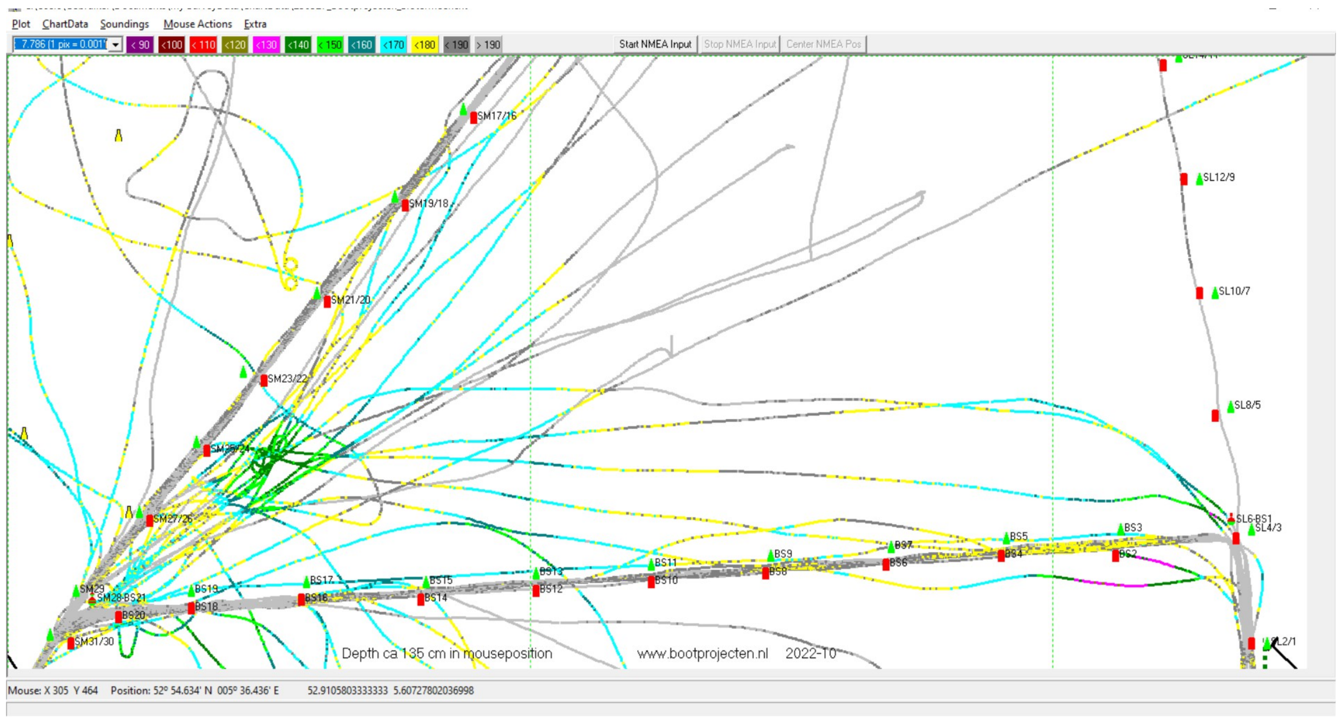
The data recorder
As a data recorder I use an Arduino Uno type micro controller board, an SD card module and a MAX485 module. (in the photo vertical, top left, slightly hidden between the cables)
The NMEA 4800 baud data stream is a differential system with two data lines A and B. They are not grounded 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 this NMEA data stream into TTL with 5 volts relative to ground.
The received NMEA data stream therefore runs via the MAX485 to the digital data port D1 of the

Arduino. The NMEA sentences that I want to use later are written on the SD card.
The data stream can be followed simultaneously via the USB port with a Serial Monitor program on the computer. Other computer programs, which can use NMEA input via the USB port, could also use it. (Make sure to choose the correct COM port and baud rate for the computer).
The program with which the micro controller board is controlled is written in the Arduino IDE on a windows computer and is loaded onto the micro controller board via a USB connection. The program is stored there and starts running when voltage is put on the Arduino. The power supply runs via a 7808 stabilizer to prevent over voltage during battery charging.

<u>The program</u>

```
/*
The MAX 485 module is connected to the Arduino as follows
        RO      Receive Out     to      D1      purple (Hardware Serial connection)
        DI      Data In         to      D2      grey
        RE      Receive Enable          D5      grey
        DE      Data Enable             D6      yellow
        5V                              5V      red
        GND                             GND     black

The SDCard connection to the Arduino:
        CS      yellow          to      D10     sdPin
        MOSI    blue            to      D11
        MISO    white           to      D12
        SCK     orange          to      D13
        5V      red             to      5V
        GND     black           to      GND
The MAX485 Module
        A               NMEA+           green
        B               NMEA-           white
*/
#include <SPI.h>
#include <SD.h>
#define REPin 5         // Receive Enable
#define DEPin 6         // Data Enable
#define sdPin 10        // SDCard
File myFile;
String  inputString = "";
char charReceived = "A";
bool txtnameExcist = false;
char txtname[15];       // filename
void setup()
{
  pinMode(sdPin, OUTPUT);
  pinMode(REPin, OUTPUT);
  pinMode(DEPin, OUTPUT);
  digitalWrite(REPin, LOW);  // receive mode
  digitalWrite(DEPin, LOW);
  inputString = "";
  inputString.reserve(200);
  Serial.begin(4800);             // baudrate
  if (!SD.begin(sdPin))
  {
    Serial.println("startup failed");
    return;
  }
  strcpy(txtname, "MoDDHrMi.TXT");    // Month, Day, GMT Hour, Minutes.TXT
}
void loop()
{
  if(Serial.available())
  {
    charReceived = (char)Serial.read();
    inputString += charReceived;
    if(charReceived == '\n')      // end of NMEA sentence - start checking it out
    {
      if((inputString.substring(0,6) == String("$GPRMC")) && (txtnameExcist == false))
          // start with extracting the day and time if this is the RMC string and create the filename
      {
        txtname[0] = inputString.charAt(55);
        txtname[1] = inputString.charAt(56);
        txtname[2] = inputString.charAt(53);
        txtname[3] = inputString.charAt(54);
        txtname[4] = inputString.charAt(7);
        txtname[5] = inputString.charAt(8);
        txtname[6] = inputString.charAt(9);
        txtname[7] = inputString.charAt(10);
        myFile = SD.open(txtname, FILE_WRITE);
        if( ! myFile )
        {
          Serial.print("Couldnt create ");
          Serial.println(txtname);
        }
        myFile.println(txtname);           // print the filename in the file
        myFile.println("");                // print a bank line
        myFile.flush();                    // print to the SD card
        txtnameExcist = true;              // skip filename creating and start recording
      }
     if(    (inputString.substring(0,6) == String("$GPRMC")) ||
          (inputString.substring(0,6) == String("$SDDBT")))// avoid unneeded strings
      {
        if(txtnameExcist == true)
        {
          myFile.println(inputString);  // print the inputstring in the file
          myFile.println("");           // print a blank line
          myFile.flush();               // print to the SD card
          Serial.println(inputString);  // print the sentence to the USB exit
        }
      }
      inputString = "";                          // clear the inputstring
    }
  }
}
```

From the recorded strings, the position and depth can be plotted at a later stage.
For example:



```
Mouse: X 305  Y 464     Position: 52° 54.634' N  005° 36.436' E        52.9105803333333  5.60727802036998
```

The lateral buoys indicates the lateral boundary of the waterway. The depth in these waterways should be at least 1.7 m (yellow or grey measuring points)

Note: In the presence of other vessels, we follow, for the sake of clarity, the SB side of the waterway. This also avoids the obligation to give way to exercising sailing school boats.

Jeroen Droogh   bootprojecten@gmail.com