GPS Monitor für Geo-Cacher

Hallo und Willkommen einer neuen Projektreihe. Diesmal wollen wir uns mit dem spannenden Gebiet des Global Positioning Systems (GPS) befassen. Im ersten Teil der mehrteiligen Reihe bauen wir eine kostengünstige und mobile LCD-GPS Anzeige, die Geo-Cachern eine tolle Hilfe bei dem Aufstöbern von Geocachen ist. Der Aufbau besteht ausfolgenden 3 Hauptkomponenten :

Anzahl	Bauteil	Anmerkungen
1	HD44780 2004 LCD Display Bundle 4x20 Zeichen mit I2C Schnittstelle für Arduino	
1	NEO-6M GPS Modul	
1	Nano V3.0 CH340 Chip fertig verlötet	

Es werden folgende Ort und Zeit Informationen auf dem Display angezeigt.

Kürzel	Bezeichnung	Ergänzende Hinweise
Alt	Altitude	Höhe über Wasser
S	Sattelite	Satelliten in Benutzung
C	Course	Kurs (Grad)
LAT	Geographische Breite	
LON	Geographische Länge	
GS	Ground Speed	Geschwindigkeit
20.01.2010 19:49	Datum / Uhrzeit	

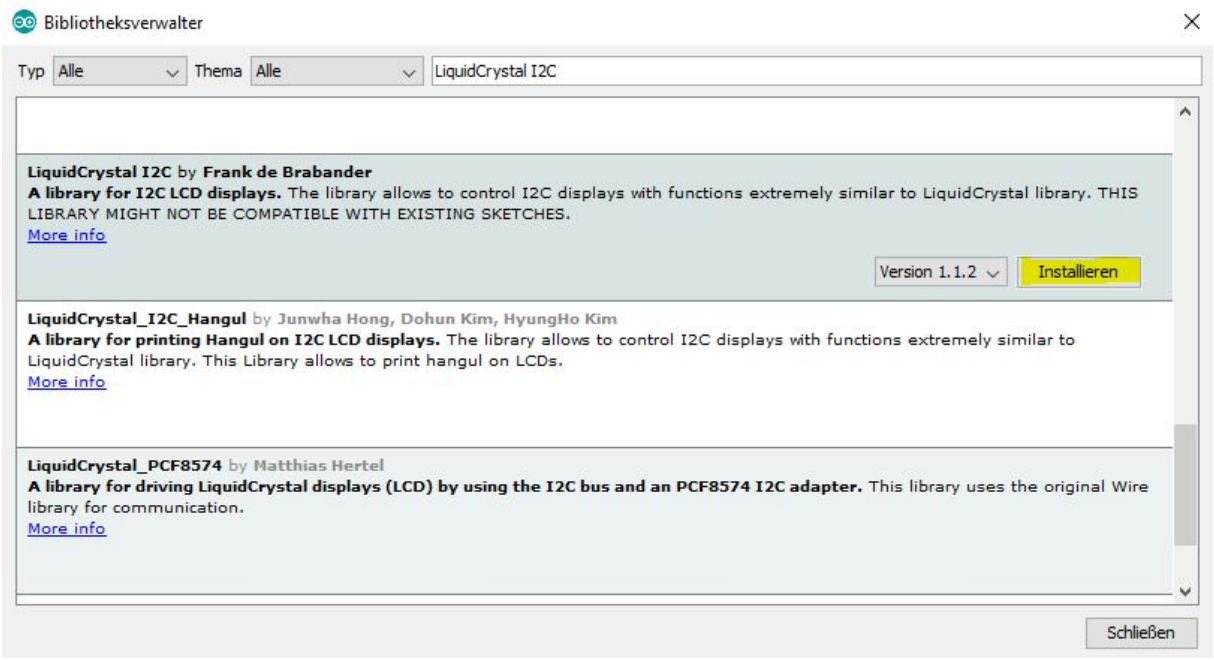
(Ansicht LCD-Display in Betrieb (fiktive Daten(Daten geändert))

Bevor der nachfolgende Code kompiliert und ausgeführt werden kann, brauchen wir zwei neue Bibliotheken.

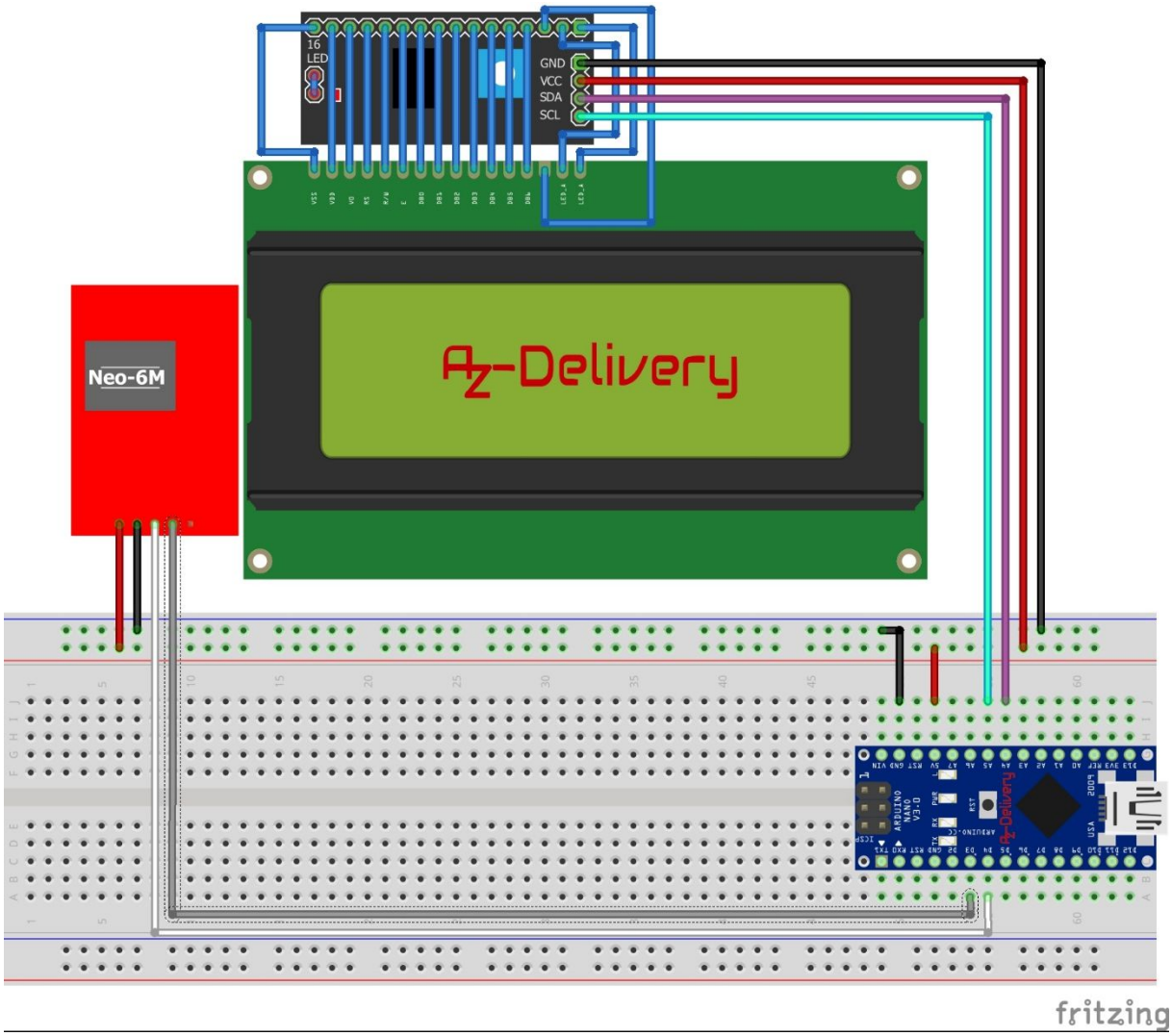
Die erste Bibliothek, TinyGPS ++ ist eine Arduino-Bibliothek zum Parsen von NMEA-Datenströmen, die von GPS-Modulen bereitgestellt werden. Wie bereits der Vorgänger der Bibliothek, TinyGPS bietet diese Bibliothek viele einfach zu nutzende Methoden zum Extrahieren von Position, Datum, Uhrzeit, Höhe, Geschwindigkeit und Kurs.

Die Bibliothek TinyGPS++ kann auf GIT Hub heruntergeladen werden: https://github.com/mikalhart/TinyGPSPlus/releases

Als weitere Bibliothek wird zur Ansteuerung des IC2 LCD Displays die Bibliothek LiquidCrystal I2C von Frank de Brabander benötigt. Diese kann direkt über den Bibiotheksverwalter unter dem Stichwort „LiquidCrystal I2C“ gefunden und installiert werden:



Nach den Vorbereitungen der IDE kann die Hardware verkabelt werden:



Danach kann nachfolgender Code hochgeladen werden:

```
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
LiquidCrystal_I2C lcd(0x27,20,4);  // set the LCD address to 0x27 for a 20 chars and 4 line display

TinyGPSPlus gps;
SoftwareSerial ss(4, 3);

void setup()
{
  Serial.begin(9600);
  lcd.init();                  // Initalisiere LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.clear();
  ss.begin(9600);
```

```
  smartdelay(1);
}

void loop()
{
if (gps.altitude.isValid())
   {
   if (gps.altitude.isUpdated()||gps.satellites.isUpdated()||gps.course.isUpdated() )
    {
    lcd.setCursor(0,0);
    lcd.print("Alt:");
    lcd.print(gps.altitude.meters()); // Altitude in meters (double)
    lcd.print("m");
    lcd.print(" S:");
    lcd.print(gps.satellites.value()); // Number of satellites in use (u32)
    lcd.print(" C:");
    lcd.print(gps.course.deg()); // Course in degrees (double)
    }
   } else
   {
   lcd.setCursor(0,0);
   lcd.print("No valid Data      ");
   }
if (gps.location.isValid())
   {
   if (gps.location.isUpdated())
    {
    lcd.setCursor(0,1);
    lcd.print("LAT:");
    lcd.print(gps.location.lat(), 4); // Raw latitude
    lcd.print("LNG:");
    lcd.print(gps.location.lng(), 4); // Raw longitude
    }
   } else
   {
   lcd.setCursor(0,1);
   lcd.print("No valid location   ");
   }
if (gps.date.isValid() & gps.date.isValid())
   {
   lcd.setCursor(0,2);
   lcd.print(gps.date.day());
   lcd.print(".");
   lcd.print(gps.date.month());
   lcd.print(".");
   lcd.print(gps.date.year());
   lcd.print(" ");
   lcd.print(gps.time.hour()+1);
   lcd.print(":");
   lcd.print(gps.time.minute());
   lcd.print("    ");
   } else
   {
```

```
   lcd.setCursor(0,2);
   lcd.print("No valid Timestamp  ");
   }
if (gps.speed.isValid())
   {
   lcd.setCursor(0,3);
   lcd.print("GS:");
   lcd.print(gps.speed.kmph());
   lcd.print("km/h ");
   lcd.print(gps.speed.knots()); // Speed in knots (double)
   lcd.print("kt ");
   } else
   {
   lcd.setCursor(0,3);
   lcd.print("No valid Speeddata  ");
   }

smartdelay(5000);
}

static void smartdelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (ss.available())
      gps.encode(ss.read());
  } while (millis() - start < ms);
}
```

Nach dem Einschalten wird ein Initialisierungsbildschirm angezeigt:

Nach empfang eines validen GPS-Signals (blaue LED des GPS Moduls blinkt) werden die Geoinformationen und Zeitdaten angezeigt:

(fiktive Daten / Daten geändert)

Ich wünsche viel Spaß beim Nachbau. Wie immer findet Ihr alle Projekte unter der GitHub Seite https://github.com/kuchto