

# **Prinzipien und Komponenten Eingebetteter System**

Wintersemester 2013/2014

|                 |  |
|-----------------|--|
| Christoph Steup | <a href="mailto:steup@ivs.cs.uni-magdeburg.de">steup@ivs.cs.uni-magdeburg.de</a>       |
| André Dietrich  | <a href="mailto:dietrich@ivs.cs.uni-magdeburg.de">dietrich@ivs.cs.uni-magdeburg.de</a> |
| Sebastian Zug   | <a href="mailto:zug@ivs.cs.uni-magdeburg.de">zug@ivs.cs.uni-magdeburg.de</a>           |

## **2. Praktische Aufgabe**

Version 0.1

## Anmerkungen

Machen Sie sich mit der Dateistruktur der Vorlage vertraut, diese umfasst insbesondere:

- diverse Arduino Bibliotheken (I2Cdev, HMC5883L, usw.)
- Flydurino.cpp, Flydurino.h
- vorlage.ino
- arduino.mk, Makefile

Sie können die Vorlage entweder in der Eclipse oder Arduino IDE bzw. als einfaches Makefile Projekt benutzen. Ihren Programmcode fügen Sie bitte in die Datei vorlage.ino ein.

Beide Teilaufgaben sollen in einen Programmcode umgesetzt werden. Für den Wechsel zwischen den Teilaufgaben sind die 2 Buttons zu benutzen. Erweitern Sie dazu die Funktion

```
int8_t checkButtons();
```

mit eigenem Programmcode.

## Teilaufgabe 1

Implementieren Sie einen Distanzsensord, der den Sharp GPD12x Sensor einbindet und einen Abstand in cm auf der 7-Segment-Anzeige ausgibt. Die Implementierung soll nicht mit den Analog-Funktionen der Arduino Bibliothek erfolgen sondern allein mit der avrlibc, da diese ein besseres Verständnis für die Vorgänge des AD-Wandlers gewährleistet.

Erweitern Sie in der Vorlage die Funktionen

```
int readADC(int8_t channel);  
  
int8_t linearizeDistance(int distanceRaw);  
  
void displayDistance (int8_t dist);
```

um die Aufgabe umzusetzen. Für die letzte Funktion sollten Sie Ihren Programmcode aus der praktischen Aufgabe 1 benutzen.

Verwenden Sie für die Linearisierungsfunktion ein Lineal, Gliedermaßstab oder ähnliches. Wie Sie die Kennlinie des Sensors auf einen Distanzwert abbilden können Sie selbst entscheiden. Die Ausgabe sollte allerdings im Bereich zwischen 10 und 30cm auf 2cm genau sein ...

## Teilaufgabe 2

Implementieren Sie eine rudimentäre Wasserwaage, die in 5 Stufen die Neigung des Boards entlang einer Achse anzeigt. Die Daten des Beschleunigungssensors werden in der Vorlage bereits über die serielle Schnittstelle ausgegeben. Experimentieren Sie zunächst ein wenig mit den Messwerten in verschiedenen Lagen des Arduino-Boards.

Leiten Sie den mathematischen Zusammenhang zwischen den Messdaten und dem Winkel des Boards gegenüber dem Horizont ab. Implementieren Sie diese in der Funktion

```
void displaySpiritLevel(int16_t acc_x, int16_t acc_y,  
int16_t acc_z);
```

Implementieren Sie gegebenenfalls ein Filter um die Ausgabe zu stabilisieren.

## Teilaufgabe 3 (optional)

Ersetzen sie die Wasserwaagen-Anwendung durch eine Kompasssoftware. Die Messdaten, die das magnetische Feld beschreiben, werden in der Vorlage bereits mit der Funktion

```
((Flydurino*)flydurinoPtr)->getOrientation(&ori_x,  
&ori_y, &ori_z);
```

ausgelesen. Wandeln Sie den Vektor in eine Richtungsangabe um und geben Sie die Winkelangabe in Grad auf dem Display aus. Vergleichen Sie die Ausgaben an verschiedenen Positionen im Raum. Was fällt auf?