Embedded Systems Labor

3. Laboratory

Autor: Roland Lezuo, Thomas Gadner
Last change: 14. Mai 2025

Eine Laboreinheit besteht aus mehreren Aufgaben. Für jede vollständig bearbeitete Aufgabe gibt es Punkte, die am Ende zur Gesamtnote beitragen. Für Fragen während des Labors stehen dir vier Betreuer zur Verfügung.

Task 1 A: Datenblatt lesen und Bauteil verstehen (10 Punkte)

Identifiziere den Beschleunigungsaufnehmer im Schaltplan des Beschleunigungsaufnehmers. Öffne das Datenblatt und verstehe, wie der Sensor entsprechend dem Beispielcode konfiguriert wurde. Untersuche dazu den Beispielcode und verweise zur Beantwortung der Frage auf die Beschreibung der Register im Datenblatt.

Task: 5 Punkte

Wie hoch ist die maximale Beschleunigung in der aktuellen Konfiguration?

Task: 5 Punkte

Welche Auflösung bietet der Sensor in der aktuellen Konfiguration?

Task 2 B: Umrechnung der Rohdaten in Beschleunigung (20 Punkte)

Um die Rohdaten für die weitere Verarbeitung nutzen zu können, müssen sie in die Beschleunigung in g umgerechnet werden. Ein kleiner Tipp vorweg. Wenn du das Datenblatt liest, wirst du feststellen, dass die Ergebnisse linksbündig im Ergebnisregister gespeichert werden. Beachte dies bei der Umrechnung.

Task: 10 Punkte

Finde im Datenblatt die Informationen zur Umrechnung der Rohdaten in die Beschleunigung. Stichwort Sensitivity und linksbündiges Register !

Task: 10 Punkte

Konvertiere die Rohdaten in deinem Programm und zeichne sie über die serielle Schnittstelle auf. Vorsicht bei der Verwendung von Teleplot: Die Plotfunktion selbst funktioniert nur mit ganzzahligen Werten, verwende daher eine textbasierte Ausgabe.

Task 3 C: Umrechnung der Daten in Neigungswinkel (20 Punkte)

Ein Beschleunigungssensor misst die Beschleunigung in drei Raumrichtungen: X, Y und Z. Im Ruhezustand registriert der Sensor nur die Erdbeschleunigung g, die je nach Neigung des Sensors unterschiedlich auf die drei Achsen projiziert wird. Aus den gemessenen Beschleunigungswerten kann mit Hilfe trigonometrischer Funktionen die Neigung eines Objektes im Raum bestimmt werden. Dabei werden die Roll- und Nickwinkel berechnet, die die Lage des Objekts relativ zur Erdoberfläche beschreiben. Die Neigungswinkel werden wie folgt berechnet:

$$Roll = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \tag{1}$$

$$Pitch = \arctan\left(-\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \tag{2}$$

Dabei sind A_x, A_y und A_z die vom Sensor gemessenen Beschleunigungen entlang der jeweiligen Achsen. Für mehr Informationen kannst du dir auch das APP-Note: **AN-1057: Using an Accelerometer for Inclination Sensing** durchlesen.

Task: 15 Punkte

Implementiere die Umrechnung der Beschleunigung in Roll und Pitch in deinem Programm. Die berechneten Winkel in Radiant müssen noch in Grad umgerechnet werden! Tipp: Verwende die <math.h> Library (https://de.wikipedia.org/wiki/Math.h).

Task: 5 Punkte

Plotte die Winkel unter verwendung von Teleplot im folgenden Format:

```
1         LOG(">Roll:%.4d\n", (int)Roll);
2
```

Task 4 Musikinstrument (50 Punkte)

In der folgenden Aufgabe soll ein einfaches elektronisches Instrument entwickelt werden, dessen Tonhöhe durch die Neigung des Mikrocontrollers gesteuert wird. Als Grundlage dient das Beispielprogramm für die PWM-Erzeugung. Deine Aufgabe besteht darin, ein Programm zu entwickeln, bei dem die Frequenz des ausgegebenen Tons - und damit die Tonhöhe - nicht fest vorgegeben ist, sondern sich durch das Kippen des Mikrocontrollers dynamisch ändert.

Task: 20 Punkte

Kombiniere dein Programm mit dem PWM-Beispiel und gib einen Ton mit konstanter Frequenz über den Lautsprecher aus. Wähle die Platinenbelegung so, dass du möglichst wenig Arbeit hast.

Task: 20 Punkte

Modifiziere das Programm nun so, dass sich die Frequenz des PWM-Signals (ARR-Register) mit dem Neigungswinkel ändert. Das ARR-Register könnte wie folgt berechnet werden

```
    uint32_t arr_from_freq(uint16_t freq){
        return SYSCLK/(freq*(TIM3->PSC+1)) - 1;
    }

```

Achtung! Um ein Rechtecksignal auszugeben, muss folgende Bedingung immer einghalten werden:

```
        TIMx->CCRx = TIMx->ARR/2;

```

Task: 10 Punkte

Das Instrument soll nun tatsächliche Noten spielen. Ändere dein Programm so ab dass dies Möglich ist.

```
    #define NOTE_E  659 // E5
    #define NOTE_B  987 // B5
    #define NOTE_C  1047 // C6
    #define NOTE_D  1175 // D6
    #define NOTE_F  1397 // F6
    #define NOTE_A  880 // A5
    #define NOTE_G  784 // G5

```

Task: Extraaufgabe (20 Punkte)

Die Änderung des Rollwinkels sollte nun die Note von A nach F ändern. Die Änderung des Pitch-Winkels soll die Oktave ändern.

Report

Wenn Sie eine oder mehrere Aufgaben abgeschlossen haben, rufen Sie einen der Betreuer im Labor zu dir. Wir werden die Aufgabe überprüfen.