C++ Microcontroller

Maximilian Starke

23. Oktober 2017

1 Aufgaben

1.1 Übersicht

1.2 Inhalte

1.2.1 Schwierigkeit I

(001)

blinkende LED (Hello World)
Erstelle eine Schaltung und ein Programm, sodass eine LED blinkt. • Konfiguriere im Code einen Pin des Controllers als Ausgang. • Verbinde diesen Pin direkt oder in direkt mit einer LED (zulässigen maximalen Strom beachten!). • Lasse die LED in deinem Quellcode blinken.

Entprellen eines Tasters

(002)

Nutze einen Eingang des Microcontrollers für einen einfachen Taster und einen Ausgang für eine LED. Schreib zunächst eine Funktion, die Zahlen ausgibt, indem sie die LED entsprechend oft kurz hintereinander aufblinken lässt. Implementiere dann einen Algorithmus, der das Herunterdrücken und anschließende Loslassen eines Tasters korrekt erkennt. Schreibe auf dieser Basis nun einen Zähler so,

- dass der Startwert des Counters beim Starten des Controllers 0 ist,
- dass mit jedem ausreichend kurzem Tastendruck der Counter um eins erhöht wird,
- dass bei Gedrückthalten des Tasters für mind. 2 Sekunden der Counter auf 0 zurückgesetzt wird,
- dass jeweils beim Loslassen des Tasters die LED einmal den Counter ausgibt.

Für die Dauer der Ausgabe ist es akzeptabel, wenn neue Tastendrücke nicht erkannt werden. Optional kann dieser „Fehler“ auch noch behoben werden.

Eingabetaster zur Lichtsteuerung

(003)

Stell dir ein Haus mit n Zimmern vor. In jedem Zimmer gibt es Lampen und Taster. Immer wenn ein Taster in einem Zimmer gedrückt wird, gehen entweder alle Lampen im Zimmer an oder alle Lampen gehen aus. Konstruiere einen modellhaften Aufbau für dieses Szenario und schreibe ein entsprechendes Programm zur Steuerung. Implementiere folgende Erweiterungen:

1 Wenn durch Tastendruck in einem Zimmer das Licht angestellt wird, geht es in allen anderen Zimmern aus.

2 Verhalten (1) tritt nur ein, wenn man den Taster kurz betätigt, bei Betätigung über 1 Sekunde lang, geht kein Licht in anderen Zimmern aus.

3 Wenn ein Taster zweimal direkt hintereinander gedrückt wird, dann geht das Licht im Zimmer für eine bestimmte Zeit ($30s$) an.

(004)

binäre Stopp-Uhr

Realisiere eine binäre StoppUhr.

- Die StoppUhr soll mit einer Genauigkeit von $1s$ die Zeit messen.
- Es gibt zwei Taster: Mit dem `Start/Stop`-Taster wird die StoppUhr gestartet, angehalten bzw. nach dem Anhalten fortgesetzt. Wird der `Reset`-Taster gedrückt, so wird die StoppUhr angehalten und auf 0 zurückgesetzt.
- Der aktuelle Zählerstand der StoppUhr soll permanent binär über n LEDs ausgegeben werden. Wähle ein geeignetes n in deinem Aufbau und schreibe die Software universell für zumindest beliebige $n < 2^{32}$.

Als Zusatzaufgabe kann überlegt werden, welches gesondertes Verhalten bei einem Zählerüberlauf gewünscht ist

(005)

Skalierbare Ampel

Schreibe eine kleine Bibliothek für Ampelanlagen. Schreibe dazu unter anderem eine Klasse „Ampel“, die eine konkrete Ampel modelliert. Diese Sollte zwei Unterklassen haben, eine für Fußgängerampeln und eine für PKW / LKW. Schaffe eine Möglichkeit, solche Ampelobjekte miteinander zu verknüpfen, sodass eine ganze Ampelanlage entsteht und dabei keine “schlimmen“ Zustände entstehen. Z.B. sollten nicht gleichzeitig Fußgänger eine Straße bei grün überqueren, wenn gleichzeitig Autos dort entlang fahren. Gehe in folgenden Schritten vor.

- Erarbeite ein Design für die Klassenhierarchie und Beziehungen zwischen den Klassen.
- Implementiere dein Design
- Teste deine Implementierung, wenn es sich anbietet.
- Überlege dir Möglichkeiten, deine Implementation zu erweitern, ggf. zu verbessern.

1.2.2 Schwierigkeit I

2 Übersicht