
GSPiC-Aufgabe #5: Ampel

(12 Punkte, Abgabe bis Mittwoch, 07.07.2010, um 18:00, keine Gruppen)

Entwerfen Sie eine Steuerung für die Ampelanlage an einem Fußgängerüberweg in einer Datei `ampel.c`. Hierbei wird die den Autos zugewandte Ampel durch die LEDs `REDO`, `YELLOWO` und `GREENO` dargestellt, die Fußgängerampel durch die LEDs `RED1` und `GREEN1` (kein Gelb). Durch das Drücken von `BUTTON0` können Fußgänger eine Umschaltung anfordern. Die LED `BLUE1` signalisiert den Fußgängern, dass eine Umschaltanforderung entgegengenommen wurde.

Die Steuerung soll im Detail wie folgt arbeiten:

- Im Ausgangszustand zeigt die Autoampel grün, die Fußgängerampel rot.
- Eine Umschaltanforderung wird durch Druck auf `BUTTON0` ausgelöst. Der Druck der Taste wird durch Aktivierung der LED `BLUE1` (= Signal kommt) bestätigt. Diese LED wird beim Start der Umschaltung wieder deaktiviert. Weitere Tastendrucke werden bis zum Ende der Fußgänger-Grünphase ignoriert (nicht aktiv).
- Nach erfolgter Umschaltanforderung zählt die Ampel über die Siebensegmentanzeige 8 Sekunden herunter, welche die Fußgänger noch warten müssen, bis ihre Ampel grün wird; in den übrigen Phasen bleibt die Siebensegmentanzeige aus. Von den 8 Sekunden bleibt die Autoampel noch 5 Sekunden grün, dann bleibt sie 1 Sekunde im Zustand gelb, bevor sie rot wird. Erst nach weiteren 2 Sekunden, während welcher beide Ampeln rot sind, schaltet die Fußgängerampel auf grün.
- Die Grünphase der Fußgängerampel soll exakt fünf Sekunden andauern, bevor sie wieder auf rot wechselt. Anschließend wechselt die Autofahrerampel für 1 Sekunde auf Gelb-Rot und wieder auf Grün in den Ausgangszustand.

Hinweise:

- Verwenden Sie die Module `LED` und `7SEG` der `libspicboard` für die Ausgabe, sowie das Modul `Timer` für die zeitkritischen Aktionen.
- Verwenden Sie das Modul `Button` der `libspicboard`, um eine Callbackfunktion zu registrieren. Diese Funktion wird bei Druck auf `BUTTON0` von der `libspicboard` ausgeführt.
- Beachten Sie: Sowohl die `Button-Callback-Funktion` als auch `Alarm-Callback-Funktionen` des `Timer-Moduls` werden im Kontext einer Unterbrechungsbehandlungsfunktion (`ISR`) ausgeführt! Achten Sie deswegen darauf, diese Funktionen sehr kurz zu halten. (In keinem Fall soll die komplette Ampelschaltungen in diesen `ISRs` realisiert werden!)
- Achten Sie darauf, dass der Mikrocontroller in Ruhephasen, während welcher keine Berechnungen durchgeführt werden, in den `Sleep-Modus` wechselt. Dies geschieht entweder implizit, z. B. in `sb_timer_delay()` (siehe Online-Dokumentation zu dieser Funktion), oder explizit durch die entsprechenden Funktionen in `avr/sleep.h`.
- Achten Sie auf die korrekte Verwendung des `volatile`-Schlüsselworts. Beschreiben Sie in einem Kommentar zu jeder verwendeten `volatile-Variable`, weshalb Sie dieses Schlüsselwort dort benötigen. Im Übrigen soll Ihr Programm mit der Optimierungsstufe `-Os` kompiliert korrekt funktionieren.