

Übungen zu Systemnahe Programmierung in C

Abschnitt 8.4: Aufgabe (ampel)

15.06.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg

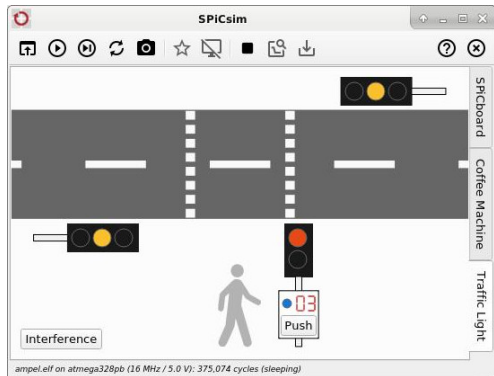


Lehrstuhl für Verteilte Systeme
und Betriebssysteme



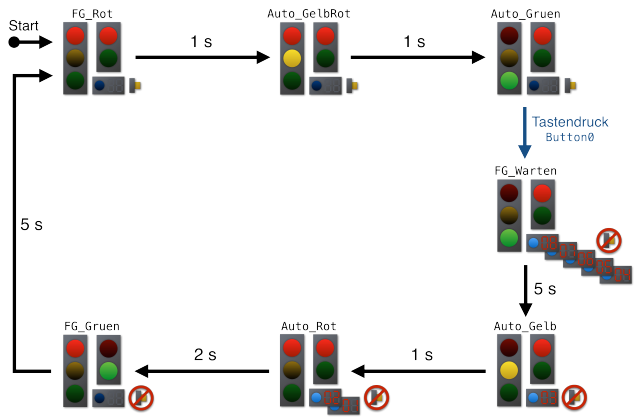
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

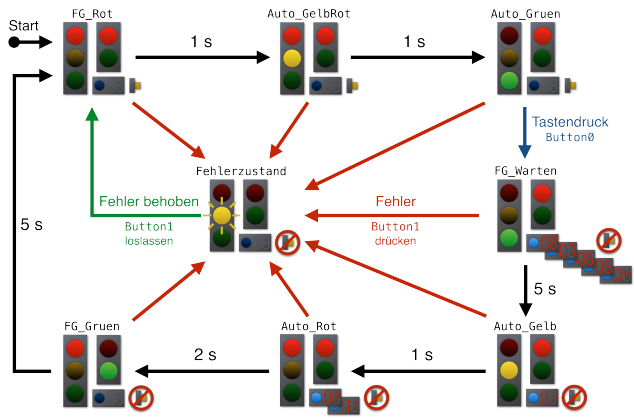


- Implementierung einer (Fußgänger-)Ampel mit Wartezeitanzeige

Ampel als Zustandsautomat ↷ 8.1



Ampel als Zustandsautomat ↷ 8.1





- Alle Transitionen werden durch Interrupts ausgelöst
 - BUTTON0 und BUTTON1 für Interrupts konfigurieren
 - ⇒ Welche Flanke soll Interrupts auslösen?
 - TIMER0 konfigurieren (Einheit: 1 Sekunde) *→ 8.2*
- Keine Verwendung des Timer Moduls der libspicboard für die Abgabe
 - ⇒ Zum debuggen aber u.U. praktisch



■ Hinweise

- Ablauf (exakt) nach Aufgabenbeschreibung \leadsto Aufgabenblatt
(Referenzimplementierung verfügbar)
- Tastendrucke und Alarmer als Ereignisse
- Passiv Warten auf die jeweiligen Interrupts
- Deaktivieren des Tasters durch Ignorieren des entsprechenden Interrupts
(Änderung der Interrupt-Konfiguration ist nicht notwendig)
- Abbildung auf Zustandsautomaten sinnvoll \leadsto 8.1
- Verwendung von volatile begründen

\leadsto 8.3