

Aufgabe 8 (10 Punkte) - Arduino: Zustandsmaschine, Interrupt**Vorbereitung**

Lesen Sie das Tutorial über Timer Interrupts unter <http://www.engblaze.com/microcontroller-tutorial-avr-and-arduino-timer-interrupts/>. Lesen Sie das Beispiel unter <http://playground.arduino.cc/Learning/ArduinoSleepCode>, wie Sie ein Arduino Board in einen Energiesparmodus setzen und wieder aufwecken.

Beantworten Sie (kurz schriftlich auf einem Blatt Papier) die folgenden Fragen:

- Wie funktioniert ein Timer mit der Nutzung eines Interrupts?
- Wozu dienen der Prescaler und der Timer Count OCR1A? Wie wird der Timer Count benutzt?

Praktikum

Verwenden Sie die Schaltung gemäß dem Schaltbild A08. Die Schaltung entspricht einem Aufbau einer Fußgängerampel und einer Fahrzeugampel, wie man sie aus dem Straßenverkehr kennt.

Aufgabe 8.1 (1+3 Punkte)

Die Funktionalität der Schaltung ist mittels des Arduino Boards, wie folgt, umzusetzen:

- Im Default-Zustand ist die Fahrzeugampel grün, die Fußgängerampel rot.
- Mit einem Knopfdruck startet nach dem Ablauf einer ersten Zeitspanne (T_w) eine Umschaltsequenz der Ampeln. Diese Sequenz beginnt mit der Umschaltung der Fahrzeugampel von grün über gelb auf rot. Danach erfolgt eine Umschaltung der Fußgängerampel von rot auf grün. Jede Umschaltung dauert dabei eine zweite Zeitspanne (T_u). Nach einer dritten Zeitspanne (T_g) schalten die beiden Ampeln zurück. Diesmal erfolgt zuerst das Umschalten der Fußgängerampel von grün auf rot, dann der Fahrzeugampel von rot über gelb-rot auf grün. Jede Umschaltung der Ampeln dauert dabei wieder die zweite Zeitspanne (T_u).

Modellieren Sie das gewünschte Verhalten des Systems mittels eines SDL-Prozesses für Zustandsautomaten auf einem Blatt Papier. Eine Anzahl von ca. sieben Zuständen ist plausibel.

Implementieren Sie ausgehend von Ihrer Modellierung deren Funktionalität. Ihre Modellierung der Zustände muss in der Implementierung klar wieder zu finden sein. Die Verwendung von Timerinterrupts ist für diese Aufgabe nicht erforderlich.

Aufgabe 8.2 (2+4 Punkte)

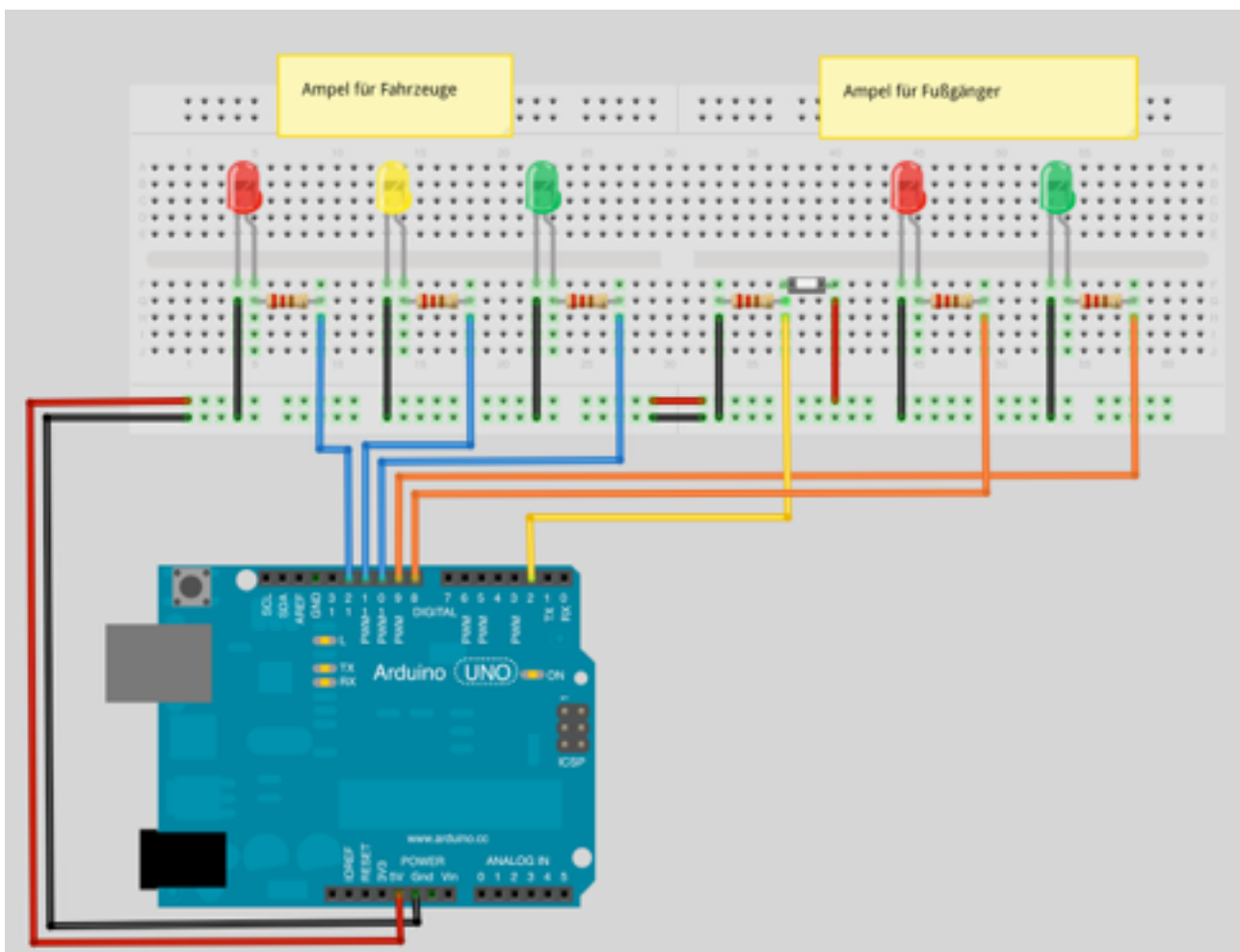
Fügen Sie dem Zustandsautomaten und dem Programm aus Aufgabe 8.1 die Eigenschaft hinzu, dass das Board (und die Ampeln) nach einer Zeit ohne Knopfdruck in einen

Energiesparmodus geht, aus welchem es bei Knopfdruck aufwacht. Der Energiesparmodus darf nur erreicht werden, wenn die Fahrzeugampel grün (und die Fußgängerampel rot) ist. Das Board muss nach dem Ablauf einer gegebenen vierten Zeitspanne T_e ohne Knopfdruck in den Energiesparmodus geschickt werden.

Verwenden Sie für diese Aufgabe einen Timerinterrupt. Schreiben Sie dazu eine Klasse `Timer`, die intern den Timerinterrupt benutzt. Die Klasse `Timer` muss u.a. die Methoden `setTimer(Zeitspanne)` und `resetTimer()` bereitstellen. Der Ablauf einer mit `setTimer(Zeitspanne)` eingestellten Zeitspanne muss bei der Ereignisverarbeitung benutzt werden. Implementieren Sie für die Klasse `Timer` das Singleton-Pattern, so dass nur eine Instanz der Klasse `Timer` existiert. Die Funktion `ISR` für den Timerinterrupt kann außerhalb der Klasse `Timer` sein. Die Funktion `ISR` kann über das Singleton-Pattern auf die `Timer`-Klasse zugreifen.

Das Programm darf nicht die Funktion `delay()` der Arduino-Bibliothek verwenden. Sämtliche Zeitabläufe sind über die Klasse `Timer` durchzuführen.

Für die Zeitspannen soll gelten: $T_e > T_w \geq T_g > T_u$. Die Zeitspannen sollen auf halbe Sekunden genau einstellbar sein.



Schaltbild A08