

Achtung: optionale Aufgaben 9 oder 9*

Die beiden folgenden Aufgaben stehen zur Auswahl zur Bearbeitung dieses Aufgabenblatts. Sie können nur eine (exklusive oder) Aufgabe abgeben, um die maximale Punktzahl von 10 zu erreichen. Abgabe in der 25 KW (ca. zwei Wochen Bearbeitungszeit).

Option a) Aufgabe 9 (10 Punkte) - Arduino: Watchdog, Input/Output**Vorbereitung (2 Punkte)**

Die folgenden Fragen sind vor dem Praktikum schriftlich zu beantworten.

a) In der Vorlesung wurde die Funktion eines sogenannten Watchdogs besprochen. Die Funktion eines Watchdog Timers kann auch dazu verwendet werden, ein System in regelmäßigen Abständen aus einer Energiesparfunktion zu wecken. Lesen Sie in der Dokumentation der ATmega Microcontroller (<http://www.atmel.com/Images/doc8161.pdf>, Kapitel 10.8 und 10.9) nach, wie Sie die Watchdog Funktion des ATmega328P anwenden.

Welche Reihenfolge des Setzen von Bits muss eingehalten werden, um Einstellungen des Watchdogs ändern zu können? Wie lautet die Anweisung (in C/C++) zur Einstellung einer Zeitspanne von 4 Sekunden für den Watchdog Timer?

b) Verstehen Sie die Benutzung von Analog I/O mit dem Arduino (<http://arduino.cc/en/Reference/HomePage>).

Welche Werte kann die Funktion `analogRead` liefern und wie sind diese zu interpretieren? Welchen Wert liefert die Funktion bei einer Eingangsspannung von 5V?

c) Sehen Sie sich das Datenblatt (Seite 1) des Temperatursensors LM35 (<http://www.ti.com/lit/ds/symlink/lm35.pdf>) an.

Welche Spannung sollte an dem Sensor bei einer Umgebungstemperatur von 10° Celsius anliegen?

Praktikum

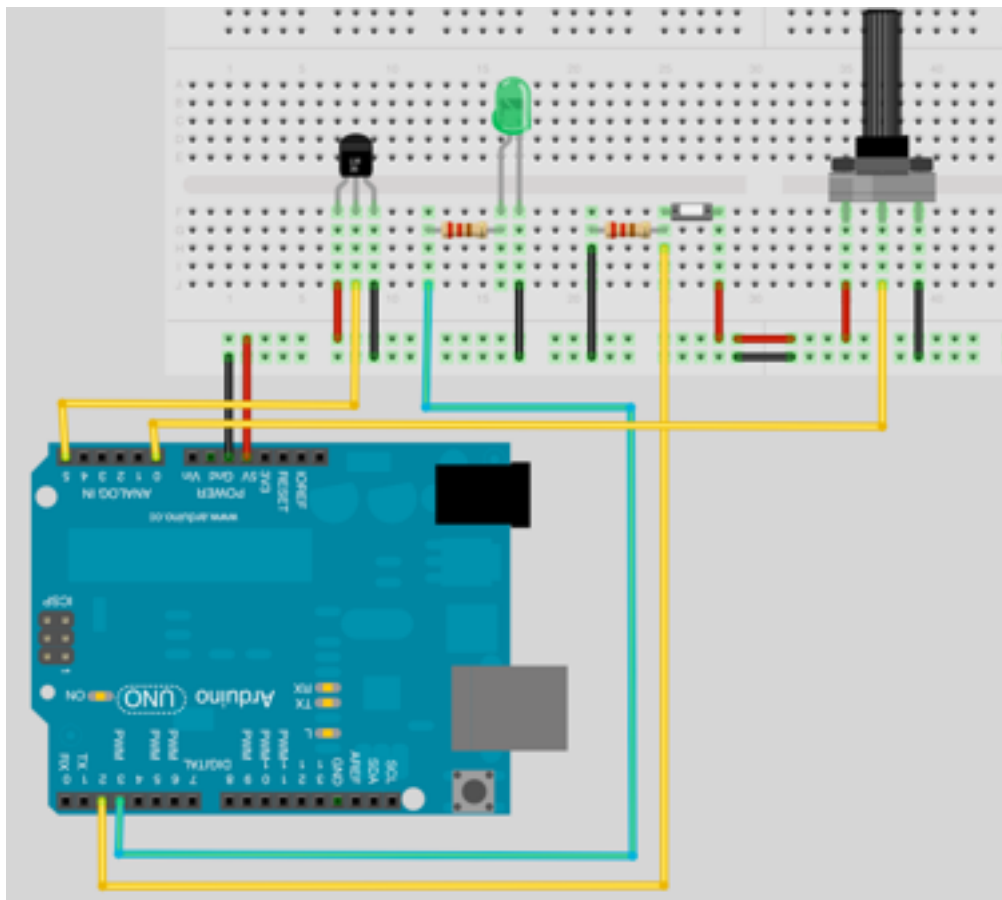
Nutzen Sie die Schaltung gemäß dem unten dargestellten Schaltbild. Die Schaltung umfasst einen Temperatursensor LM 35 (Wichtig: Nicht die Pole vertauschen!), einen Potentiometer/Drehknopf, eine Led, zwei Widerstände (220 Ohm) und das Arduino Board.

Aufgabe 9 (2+2+2+2 Punkte)

Das Arduino Board soll die folgenden Funktionen in der Schaltung übernehmen:

- In einem ersten Zustand wird ein Zeitintervall mittels des Potentiometers eingestellt. Das Zeitintervall wird zwischen 1, 2, 4 und 8 Sekunden gewählt. Das gewählte Zeitintervall wird über die Helligkeit der LED derart angezeigt, dass diese für das größte Intervall (8s) am dunkelsten und für das kürzeste Intervall (1s) am hellsten leuchtet (Stichwort: Pulse Width Modulation).
- In einem zweiten Zustand wird die Temperatur des Temperatursensors ausgelesen. Die Temperatur wird dabei aus 8 Messpunkten gemittelt, die gleichmäßig über 250ms verteilt sind. Das Messergebnis wird seriell an den angeschlossenen Rechner übertragen.
- Im zweiten Zustand wird nach der Messung (Messreihe mit 8 Messpunkten) die Schaltung in einen Energiesparzustand versetzt. Die Schaltung wacht regelmäßig aus diesem Zustand auf und führt dann wieder die Messung durch. Die Schaltung ist für genau die Zeit im Energiesparmodus, die mittels des Drehknopf eingestellt wurde. Im Energiesparmodus ist die LED aus.
- Der erste Zustand ist der Startzustand. Ein Betätigen des Knopfes schaltet zwischen den Zuständen um, ggf. ist der Energiesparzustand hierzu zu verlassen.

Implementieren Sie die beschriebenen Funktionen auf dem Arduino Board. Verwenden Sie für das regelmäßige Aufwecken einen Watchdog Timer. Tipp: Die Dateien `avr/sleep.h` und `avr/wdt.h` sollten eingebunden werden.



Schaltbild für Aufgabe 9

Option b) Aufgabe 9* (10 Punkte) - MDD (Model-Driven-Development)

Programmieren Sie gemäß dem Model-Driven-Development mit der Rhapsody Software entweder die oben in Aufgabe 9 genannte Funktionalität mit dem TI MSP Launchpad oder eine eigene Anwendung ähnlicher Komplexität (mehrere Zustände, Hardwarezugriff, I/O) mit den zur Verfügung stehenden Bauteilen.

Sie finden Informationen zu Rhapsody z.B. unter <http://www.willert.de/produkte/model-driven-sw-engineering/ibm-rational-rhapsody-/> und <http://www.willert.de/service/getting-starteds/> sowie zum verwendeten Mikrocontroller der MSP430-Reihe unter: <http://www.ti.com/product/MSP430G2553/technicaldocuments>.

Zur Bewertung Ihrer Lösung sollten Sie die einzelnen Schritte, die Sie im Model-Driven-Development durchgeführt haben, und Ihr Model erklären.