

Prinzipien und Komponenten Eingebetteter System

Wintersemester 2013/2014

Christoph Steup	steup@ivs.cs.uni-magdeburg.de
André Dietrich	dietrich@ivs.cs.uni-magdeburg.de
Sebastian Zug	zug@ivs.cs.uni-magdeburg.de

Fertigung der Experimentierplatine für die 1. Praktische Aufgabe

Version 0.3

Allgemeine Hinweise

Bitte lesen Sie die folgenden Hinweise und die Aufgabenstellung aufmerksam durch. Sollten Ihnen einzelne Sachverhalte unklar bleiben, fragen Sie bitte beim Übungsleiter nach.

Die Fertigung erfolgt nur nach einer Terminvereinbarung und in Gegenwart eines Übungsleiters! Die Bestückung außerhalb des Labors der Arbeitsgruppe EOS muss ausdrücklich genehmigt werden.

Selbstständige Veränderungen an der Platine, die in der Aufgabenstellung nicht beschrieben sind, sind nicht erlaubt – gute Ideen zur Verbesserung des Designs der Übungsplatine aber immer gefragt.

Organisieren Sie in Ihrer Arbeitsgruppe die Aufgabenverteilung. Dies ist eine wichtige Voraussetzung für die systematische Abarbeitung der gestellten Aufgaben.

Beachten Sie das Gefahrenpotential beim elektrotechnischen Arbeiten!

- Machen Sie sich vor der Fertigung im Labor mit den Vorgängen beim manuellen Lötén vertraut.
- Gehen Sie beim Bestücken langsam und methodisch vor.
- Achten Sie auf die korrekte Handhabung des Lötkolbens.
- Stellen Sie eine Temperatur von 220 Grad an der Lötstation ein.
- Tragen Sie beim Schleifen, Bohren und Brechen von Bauteilen eine Schutzbrille!

Datenblätter

Komponente	Link zum Datenblatt
Arduino ADK	http://arduino.cc/en/Hacking/PinMapping2560
AtMega 2560	http://eos.cs.ovgu.de/wp-content/uploads/2013/09/ATmega2560.pdf
LTM-8522HR	http://eos.cs.ovgu.de/wp-content/uploads/2013/09/LTM8522HR.pdf
Proto-Shield	http://arduino.cc/de/Main/ArduinoProtoShield

Viel Spaß !

1. Löten Sie die Stiftleisten an das Proto Shield

Prüfen Sie die korrekte Lage des Proto-Shields gegenüber dem Arduino Core Board. Die 4 Bohrungen müssen übereinander liegen.

- Ordnen Sie die Stiftleisten entsprechend der Pinanzahl den Sockeleisten zu (6, 8, 8, 10 Pins).
- Stecken Sie die Leisten in das Arduino Board
- Setzen Sie das Proto-Shield auf die Stiftleisten auf
- Kontrollieren Sie den korrekten geraden Sitz!
- Verlöten Sie die Sockelleiste auf dem Proto-Shield
- Prüfen Sie die Verbindungen (visuell, Durchgangsprüfung)
- Testen Sie die Spannung am 3.3 und 5 V Pin des Experimental Shields



Abbildung 1 – Arbeitsschritt 1a - Arduino Board mit angepassten Stiftleisten

2. Schließen Sie den Logikteil des Displays an die Spannungsversorgung (5V) des Arduinio an

- Ziehen Sie das nun mit den Sockelleisten bestückte Proto-Shield von Arduino ab.
- Verbinden Sie den Pin 1 des Displays mit der GND Leiste mittels Draht auf der Rückseite des Boards.
- Verbinden den Pin 18 und 17 des Displays mit dem 5V Pin der Sockelleiste
- Prüfen Sie den Widerstand Ihrer Lötverbindung!

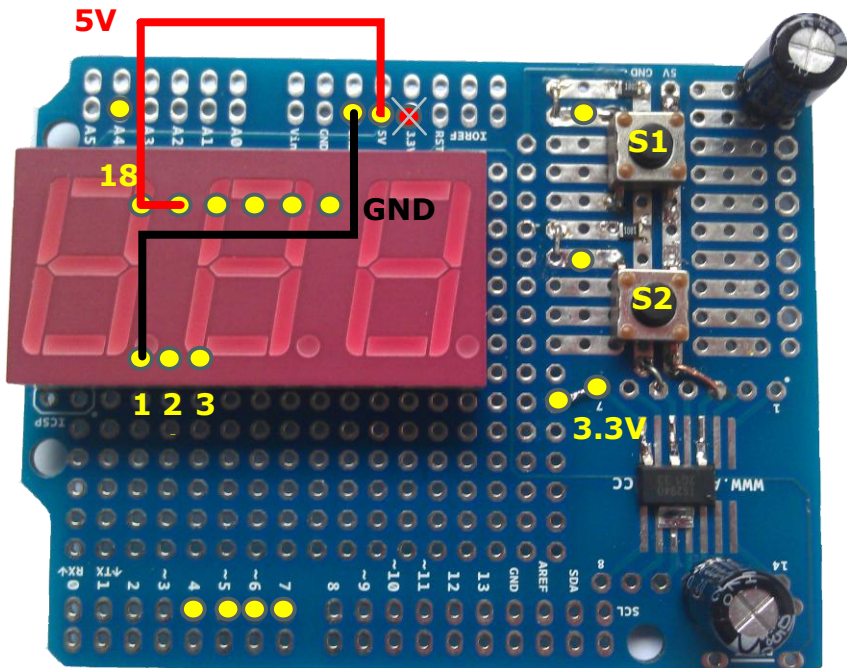


Abbildung 2 - Arbeitsschritt 2 - Spannungsversorgung des Displays

3. Verbinden Sie den Ausgang des Spannungswandler mit der LED Treiberstufe des Displays

- Prüfen Sie die korrekte Funktion des Wandlers mittels Multimeter (Spannungsbetrieb!). Messen Sie dazu die Spannung zwischen dem 3.3 Volt Ausgang des Wandlers und einem GND Kontakt.
- Verlöten Sie die Pins 2 und 3 des Displays
- Verbinden Sie diesen gemeinsamen Eingang mit dem 3.3 V Ausgang des Spannungswandlers (PIN)
- Prüfen Sie den Widerstand der Lötverbindung!

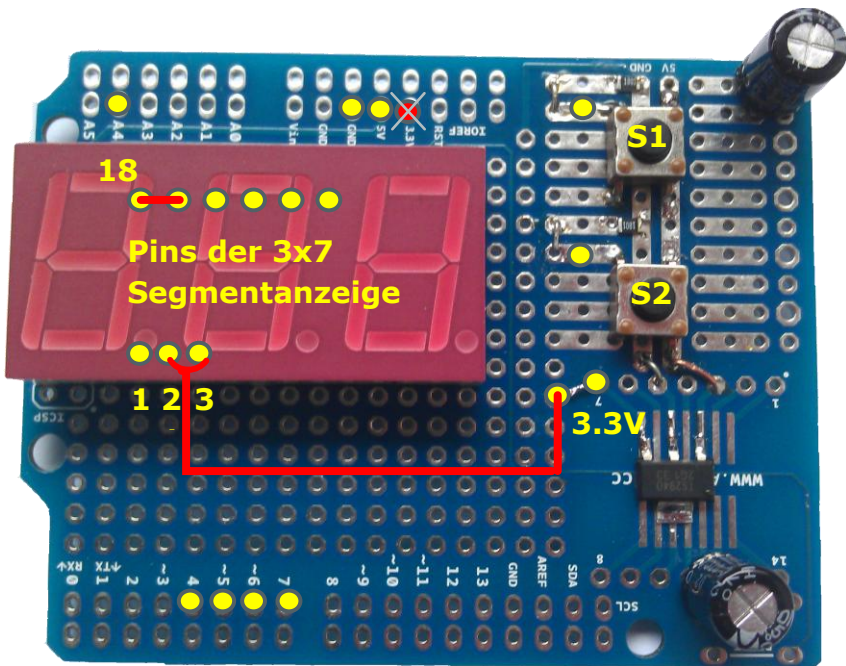


Abbildung 3 - Arbeitsschritt 3 - Spannungsversorgung der LED Treiberstufe

4. Verbinden Sie die seriellen Bus mit den entsprechenden Eingängen des Displays

- Verbinden Sie die Datenverbindung mit dem Display
 - Arduino Pin 7 - Display Pin 14 (Data Enable)
 - Pin 6 - Pin 15 (Data Input)
 - Pin 5 - Pin 16 (Clock)

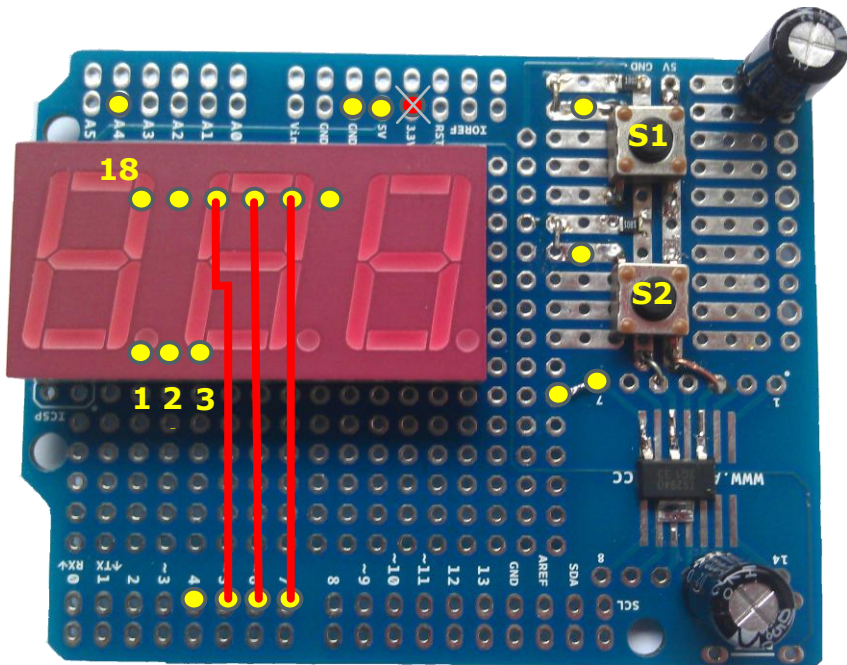


Abbildung 4 - Arbeitsschritt 5 - Serielle Busanbindung für das Display

5. binden der Buttons mit dem Mikrocontroller

- Verbinden Sie den Button S1 mit dem Arduino Pin A4 und den Button S2 mit dem Pin 4.
- Kontrollieren Sie die korrekte Funktionsweise, messen Sie dazu den Widerstand beim Drücken der Taster.

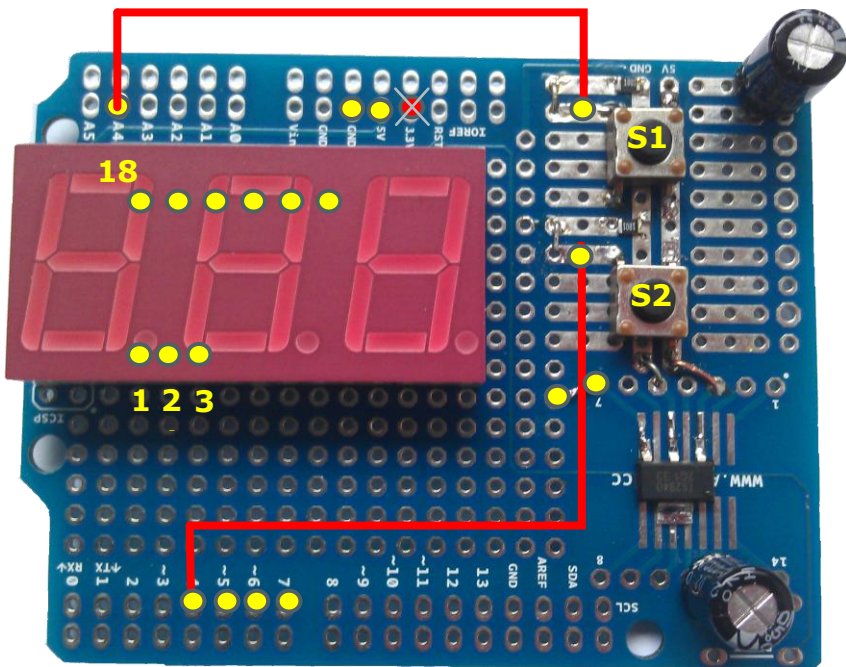


Abbildung 5 - Arbeitsschritt 6 – Verbinden der Buttons S1 und S2

6. Inbetriebnahme

1. Stecken Sie das Proto-Shield auf den Arduino
2. Verbinden Sie den Arduino mit Ihrem Rechner. Benutzen Sie die Buchse vom USB-Stecker Typ B.
3. Flashen Sie die Code-Vorlage der ersten praktischen Aufgabe.
4. Auf der Sieben-Segment-Anzeige sollten die Buchstaben „EOS“ erscheinen.
5. Starten Sie einen seriellen Ausgabeterminal und kontrollieren Sie die Ausgaben

Anhang

Installation der Entwicklungsumgebung (Eclipse):

Es wird empfohlen Eclipse als Entwicklungsumgebung mit dem dazugehörigen AVR-Plugin zu installieren. Folgen Sie den Installationshinweisen auf den jeweiligen Webseiten.

Eclipse: <http://www.eclipse.org/downloads>

AVR-Plugin: http://avr-eclipse.sourceforge.net/wiki/index.php/Plugin_Download

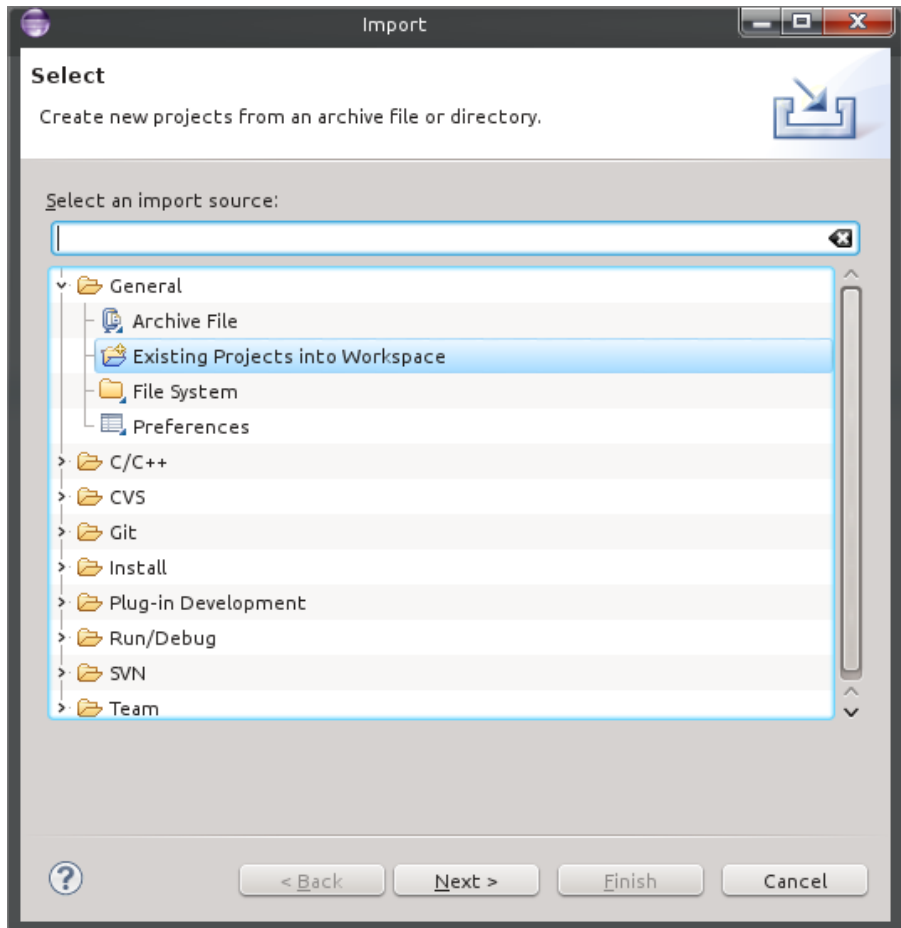
Import des Beispiel-Projektes in Eclipse:

Laden Sie das Beispiel-Projekt von der Webseite und entpacken Sie es in ein Verzeichnis Ihrer Wahl:

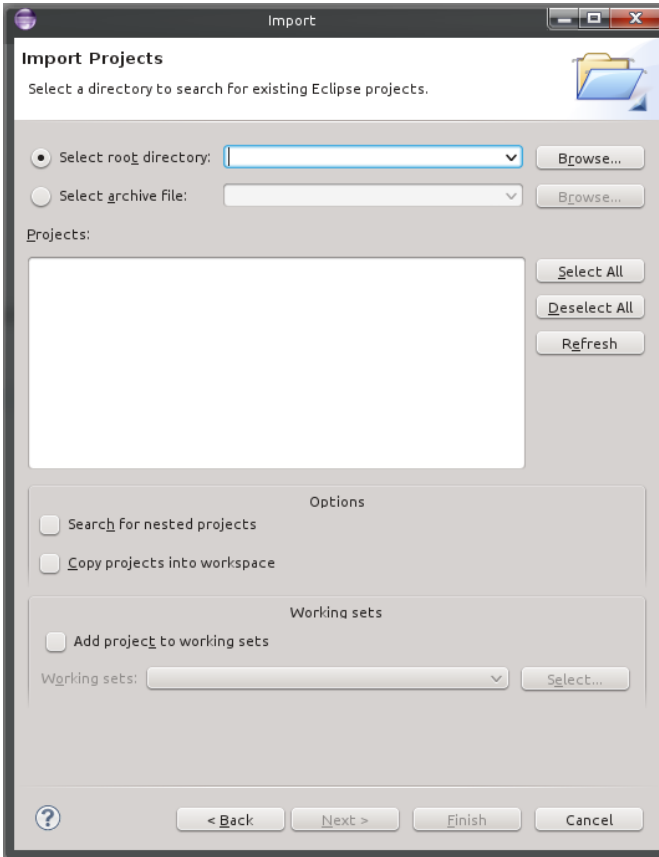
<http://eos.cs.ovgu.de/wp-content/uploads/2013/10/PKES1.zip>

Öffnen Sie hierfür Ihre Eclipseumgebung und klicken Sie auf „File“ > „Import“

Daraufhin erscheint der folgende Dialog, wählen Sie unter „General“ die Option „Existing Projects into Workspace“:

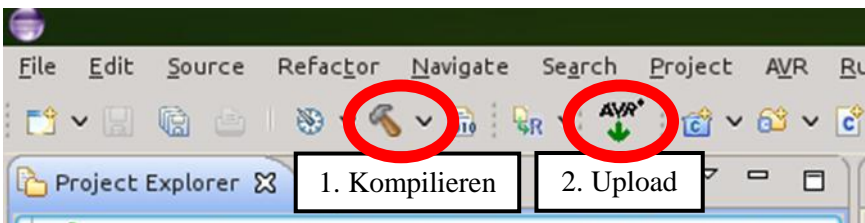


Klicken Sie auf den Button „Next“ und es erscheint der folgende Dialog. Wählen Sie unter dem Menüpunkt „Select root directory“ das entsprechende Verzeichnis des ersten Beispielprojektes und klicken Sie auf „Finish“.



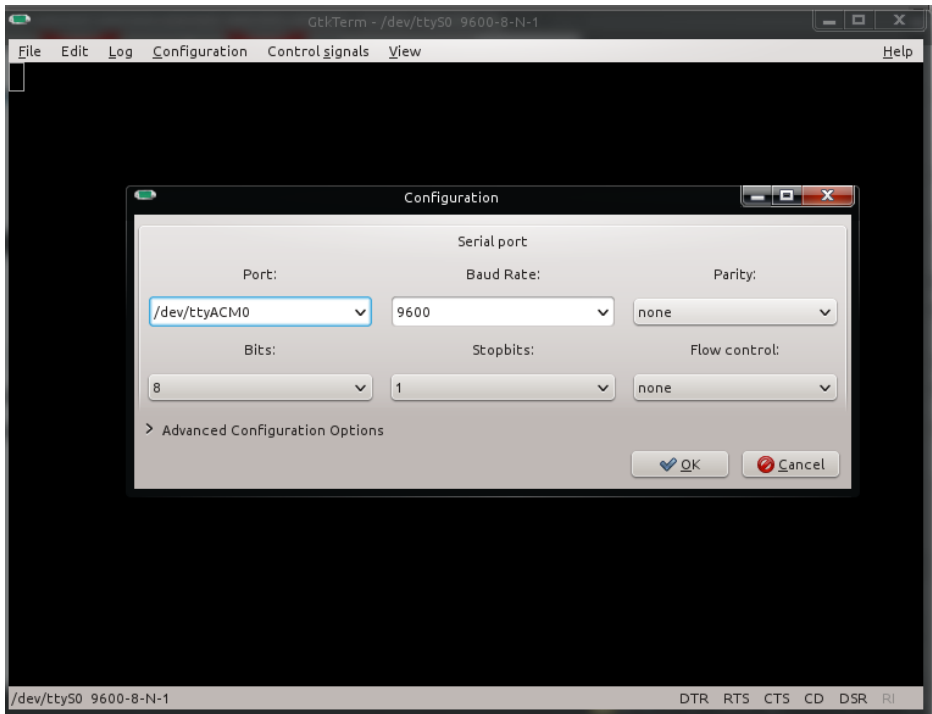
Flashen des Arduino-Boards:

Da bereits alle Parameter in den importierten Programm-Settings bezüglich der Eigenheiten des AVR's gesetzt sind, muss das Projekt PKES1 nur kompiliert und auf den AVR gebracht werden. Siehe hierfür folgende Abbildung:



Ausgabe über die Serielle Schnittstelle:

Hierfür eignet sich vor allem das Tool „gtkterm“, welches einfach über die Konsole aufgerufen werden kann. Über die Auswahl „Configuration“ > „Port“ gelangt man zu den folgenden Einstellungen, welche mit den unten abgebildeten Parametern befüllt werden sollte:



Dann einfach okay und das Auslesen beginnt... Man beachte, Flashen und gleichzeitiges Lesen über die Serielle-Schnittstelle sind nicht möglich, deshalb sollte „gtkterm“ immer beendet werden, bevor man ein neues Programm auf den flashed.

Für Hacker via „make“:

Zusätzlich ist noch ein Makefile im Projekt enthalten, welches folgende Optionen bereithält:

`make all` # kompilieren des Projektes

`make program` # `make all` und portieren via `avrdude`

`make asm` # erstellt Assemblercode separat

`make com` # `make program` & zusätzlicher Start von „`gtkterm`“