Microcontroller Programming

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

Mikrocontroller-Technik


Source: eigenes Foto (CC0 1.0)

Die Mikrocontroller-Technik befasst sich damit, wie man einem programmierbaren Bauteil (“Mikrocontroller”) Leben einhaucht. Dabei wird die Software im Folgenden in der Programmiersprache C programmiert. Im Gegensatz zu der im 1. und 2. Semester im Kurs Informatik dargestellten Sprachkomponenten und Algorithmen wird hier Wert auf die Eigenheiten durch die Hardware-nähe gelegt.

Die Veranstaltung ist mit dem EST Labor kombiniert. Details zum Ablauf finden Sie beim EST Labor.

1. Hello Blinking World

1. Hello Blinking World

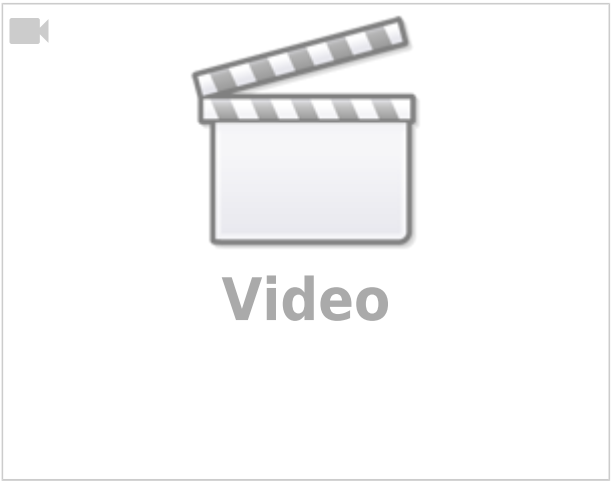
Ziele

Nach dieser Lektion sollten Sie:

1. x

Video

LED Blinken und Bit-Manipulation



2. Sound und Timer

3. Logische Funktionen

4. Up/Down Counter

5. Menüführung

6. Würfel und Zufall

7. Uhr und Zeitraster

8. Temperatur und Analog-Digital-Wandler

9. UART und Terminal

9. UART und Terminal

Bei der Programmierung wünscht man sich häufig die Möglichkeit Daten des Mikrocontrollers irgendwo darzustellen. Mit Hilfe des Freeware Programms PuTTY kann leicht ein Terminal für die Kommunikation mit dem PC geöffnet werden. Zusätzlich wird dann noch ein USB-to-serial Adapter benötigt.

1. Tutorial zu UART auf mikrocontroller.net
2. Datenpaket für RX/TX
3. Leitungslänge vs. Übertragungsrate

10. I2C Schnittstelle

10. I2C Schnittstelle

Ziele

Nach dieser Lektion sollten Sie:

1. wissen wie die Kommunikation zwischen I2C Master und Slave funktioniert

Dokumentation von Atmel

- Application Note: TWI Module as I2C Master
- Application Note: TWI Module as I2C Slave
- alternative Implementierung von Elia Ritterbusch

Beschreibung

Statemachine der I2C Kommunikation

Statemachine der I2C Kommunikation

<uml> keine_Kommunikation : - Datenausgabe1 : Datenbit = 1 Datenausgabe0 : Datenbit = 0

[*] --> keine_Kommunikation keine_Kommunikation -> keine_Kommunikation : !SCL || !SDA keine_Kommunikation --> Warten_auf_Clock_Lo : SCL && SDA Warten_auf_Clock_Lo -> Warten_auf_Clock_Lo : !SCL || SDA Warten_auf_Clock_Lo --> Bits_lesen_aktiv : SCL && !SDA Bits_lesen_aktiv -> Bits_lesen_aktiv : SCL Bits_lesen_aktiv --> Warten_auf_Clock_Hi : !SCL Warten_auf_Clock_Hi -> Warten_auf_Clock_Hi : !SCL Warten_auf_Clock_Hi --> Datenausgabe0 : SCL && !SDA Datenausgabe0 -> Datenausgabe0 : SCL && !SDA Datenausgabe0 --> Bits_lesen_aktiv : !SCL Datenausgabe0 --> keine_Kommunikation : SDA Warten_auf_Clock_Hi -> Datenausgabe1 : SCL && SDA Datenausgabe1 ---> Datenausgabe1 : SCL && SDA Datenausgabe1 --> Warten_auf_Clock_Hi : !SCL Datenausgabe1 --> Bits_lesen_aktiv : !SDA

</uml>

Zeitverlaufsdiagramm

11. SPI-Schnittstelle

11. SPI-Schnittstelle

Ziele

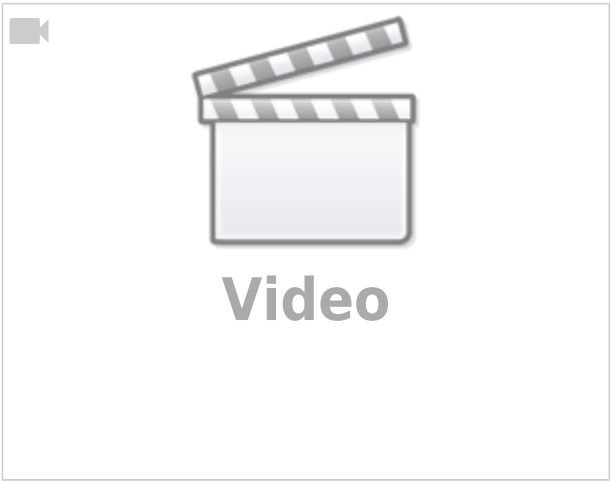
Nach dieser Lektion sollten Sie:

1. wissen, welche wie man theoretisch mehrere Slaves mit einem Master verbindet.
2. die Namen der 4 Leitungen und deren Funktionen kennen, welche jeweils an einem Slave enden.
3. die Abkürzungen SDI, SDO, MOSI, MISO, CS, SS, SCK kennen
4. die Vorteile von einer synchronen im Vergleich zu einer asynchronen Schnittstelle erklären können.

Video

Theorie zum SPI

Beispiel für SPI mittels Arduino



Tipps

- Nutzen Sie die Anzeige von Zeilennummern: Tools -> Options -> Text Editor -> All languages -> General -> Line numbers

Links

- avr_programmierung_fuer_dummies
- schönes Online Open Source Buch zum Erlernen der Programmiersprache C

- Eine schöne Einführung in die Embedded Softwareentwicklung ist im Buch Sensornetzwerke in Theorie und Praxis - Embedded Systems-Projekte erfolgreich realisieren von Kollegen Meroth und Sora zu finden. Dort wird der Einstieg in das Feld die (in Hardware) eingebettete Softwareentwicklung erklärt. Aus dem Hochschulnetz bzw. mit VPN können Sie dieses direkt bei Springer Link betrachten. Eine andere schöne Einführung findet sich auf Mikrocontroller.net.
- deutsche Übersetzung der ATmega88 Anleitung
- Tabelle der ASCII Zeichen
- Rechner für Interrupt Timer und PreScaler
- atmel_studio_tipps

- SimulIDE: kostenlose Simulation u.a. von ATMEL Chips. Diese sind auch programmierbar. Leider mit 2 Haken: 1. es kann nur ein Microcontroller simuliert werden, 2. die aktuelle Version (SimulIDE 0.3.12-SR3) scheint bei falscher Verlegung der Verbindungen leicht abzustürzen.

- Bauanleitung des Mexle AVR-Proggis