



Zielstellung

Der Schnelleinstieg zur Mikrocontroller-Programmierung soll Ihnen helfen, SiSy AVR bzw. das Add-On AVR kennen zu lernen und ein erstes Programm in der hardwarenahen Programmierung mit SiSy AVR und dem myAVR Board MK2 zu erstellen.

Das Ziel ist es, beim Start des Mikrocontrollers alle optischen Ausgabegeräte (rote, gelbe und grüne LED) auf dem myAVR Board zum Leuchten zu bringen. Als Mikrocontroller wird der ATmega8 verwendet.

Voraussetzungen

Für die Bearbeitung der Aufgaben benötigen Sie folgende Software und Hardware:

Software

- SiSy-Ausgabe AVR, Microcontroller++, Professional oder Developer ab der Version 3.4
- Ab Windows XP bis Windows 10
- Installierter USB-Treiber

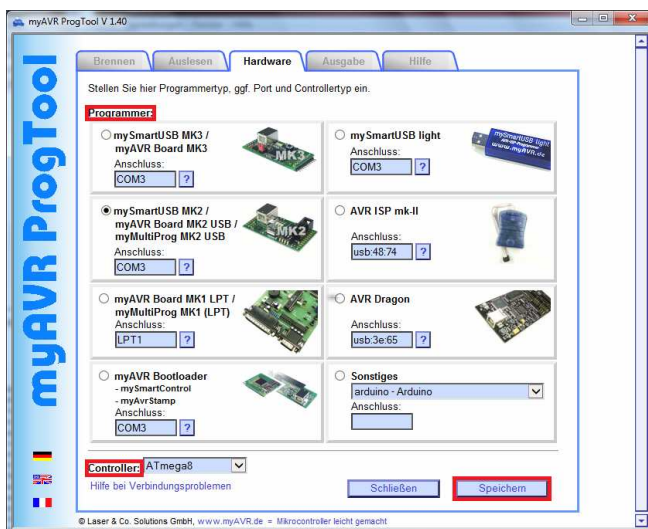
Hardware

- ein myAVR Board MK2 oder myAVR Board light
- Programmierkabel (USB)
- 9 V Batterie oder Netzteil (optional)

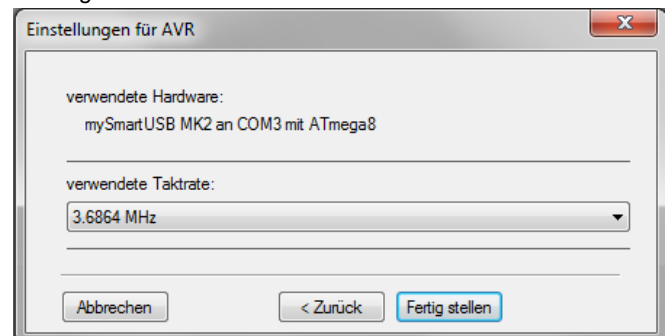
Im SiSy LibStore finden Sie Beispielprogramme und Programmvorlagen zum Download, die kontinuierlich aktualisiert werden. Eine ausführliche Beschreibung zum SiSy LibStore und der Hilfsfunktionen, z.B. Syntax zu Befehlen oder Druckmöglichkeiten, finden Sie im Benutzerhandbuch von SiSy.

1. Ein neues Projekt anlegen

Starten Sie SiSy, wählen Sie „Neues Projekt erstellen“ und vergeben Sie einen Programmnamen, z.B. „Alle Lichter an“. Aktivieren Sie das Vorgehensmodell „Programmierung“ und bestätigen Sie mit „OK“. Sie erhalten das Fenster vom „myAVR ProgTool“. Wählen Sie Ihren Programmierer sowie den Controller aus; „Speichern“ Sie.



In dem folgenden Dialogfenster werden Ihnen diese Angaben zur Kontrolle angezeigt. Die empfohlene Taktrate ist voreingestellt. Sie können diese ändern.

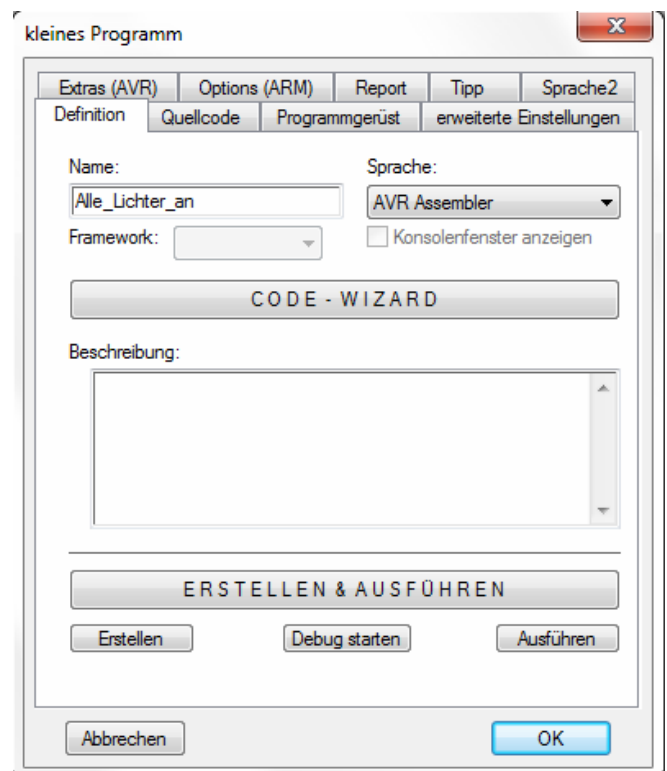


Im nächsten Dialogfeld markieren Sie „leeres Diagramm“ und klicken auf „weiter“.

2. Kleines Assembler-Programm anlegen

Erstellen Sie ein Programm für den AVR Mikrocontroller, indem Sie per Drag & Drop aus der Objektbibliothek ein Objekt „kleines Programm“ in das Diagrammfenster ziehen.

In dem aufgeblendeten Dialogfenster vergeben Sie den Namen „Alle_Lichter_an“. Der Datei- und Programmname wird dabei automatisch vergeben. Wählen Sie die Sprache „AVR Assembler“.



Über das Dialogfenster „Programmgerüst“ können Sie die Vorlage „Grundgerüst“ für ein AVR Assemblerprogramm laden. Aktivieren Sie dafür die Schaltfläche „Struktur laden“. Das geladene Programmgerüst steht Ihnen nun uneingeschränkt für die Weiterverarbeitung zur Verfügung.

3. Quellcode in Assembler erstellen

Die Ausgabegeräte (LEDs) sollen vom Prozessorport D gesteuert werden. Die Realisierung erfolgt über Bits im Register R16.

Ergänzen Sie die Programmvorlage mit nachfolgend aufgeführtem Quellcode.

```

;+-----+
; | Titel      : Übung: „alle Lichter an“
;+-----+
; | Prozessor  : ATmega8 3,6864 MHz
; | Sprache   : Assembler
; | Datum     : 28.02.2006
; | Autor     : Dipl. Ing. Päd. Alexander Huwaldt
;+-----+
#include "AVR.H"
;+-----+
;Reset and Interrupt vector ;VNr. Beschreibung
rjmp main ;1 POWER ON RESET
reti ;2 Int0-Interrupt
reti ;3 Int1-Interrupt
reti ;4 TC2 Compare Match
reti ;5 TC2 Overflow
reti ;6 TC1 Capture
reti ;7 TC1 Compare Match A
reti ;8 TC1 Compare Match B
reti ;9 TC1 Overflow
reti ;10 TC0 Overflow
reti ;11 SPI, STC Serial Transfer Complete
reti ;12 UART Rx Complete
reti ;13 UART Data Register Empty
reti ;14 UART Tx Complete
reti ;15 ADC Conversion Complete
reti ;16 EEPROM Ready
reti ;17 Analog Comparator
reti ;18 TWI (IC) Serial Interface
reti ;19 Store Program Memory Ready
;+-----+
;Start, Power ON, Reset
main: ldi r16,lo8(RAMEND)
      out SPL,r16
      ldi r16,hi8(RAMEND)
      out SPH,r16
      ldi r16,0b00011100 ; Bit 2-4 auf HIGH
      out DDRD,r16 ; Port D.2-4 auf OUT
;+-----+
mainloop: wdr
          ldi r16, 0b00011100 ; Bit 2-4 auf HIGH
          out PORTD,r16 ; Port D.2-4 auf HIGH
          rjmp mainloop
;+-----+

```

4. Kompilieren und Linken

Der eingegebene Quellcode muss nun in Maschinencode für den AVR Prozessor übersetzt werden. Wählen Sie dazu die Schaltflächen „Kompilieren“ und „Linken“. Bei fehlerfreier Übersetzung liegt das Programm unter dem Namen „Alle_Lichter_an.hex“ vor und kann auf den FLASH-Programmspeicher des Prozessors gebrannt werden.

5. Hardware anschließen und brennen

Das myAVR Board verfügt über eine ISP (In System Programming) Schnittstelle. Der Prozessor muss also nicht für die Programmierung aus dem System entfernt werden, um ihn in einem gesonderten Programmiergerät zu brennen, sondern kann im myAVR Board direkt programmiert werden.

Dazu verbinden Sie das myAVR Board über das Programmierkabel mit dem USB-Port Ihres Rechners. Zum Brennen wählen Sie die Schaltfläche „Brennen“. In Abhängigkeit Ihrer Konfiguration erhalten Sie im Ausgabefenster eine entsprechende Meldung:

vorbereiten ...
brennen ...

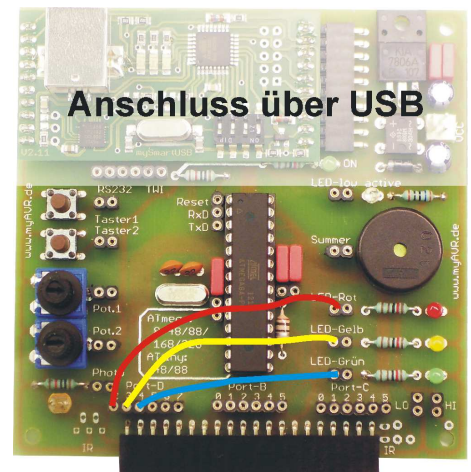
benutze: mySmartUSB an COM3 mit ATmega8
USB-Treiber installiert, aktiv (V 5.0.0.0), Port: COM3
Prozessor: ATmega8
schreibe 1 Bytes in Flash-Memory ...
... erfolgreich (0.39 s)
OK

6. Mikrocontrollerlösung testen

Zum Testen des Programms ist es erforderlich, Port D mit den Ausgabegeräten zu verbinden. Ziehen Sie das Programmierkabel ab.

Verbinden Sie die LEDs mit dem Prozessorport D entsprechend der Abbildung. Verwenden Sie Patchkabel!

Prüfen Sie die Verbindungen und schließen Sie das Programmierkabel an. Das Programm startet automatisch und die LEDs auf Ihrem Board leuchten.



7. Quellcode in der Sprache C erstellen

Das myAVR Board lässt sich auch mit der Sprache C programmieren. Legen Sie dazu wiederum ein kleines Programm mit dem Namen „Alle_Lichter_an_C“ an. Wählen Sie die Sprache AVR C und geben Sie folgenden Quellcode ein:

```

//-----
// Titel      : Beispiel: „alle Lichter an_C“
//-----
// Datum     : 01.03.2006
// Version    : 1.2
// Autor      : Dipl. Ing. Päd. Alexander Huwaldt
//-----
#define F_CPU 3686400

#include <avr/wdt.h>

void init (void)
{
  DDRD=0x1C; // Set Bits 2, 3 und 4 von
             // PORTD in 1 - "Ausgang"
}

//=====
main (void)
{
  init(); // Initialize Peripherals
  while (1) // Mainloop
  {
    PORTD=0x1C; // auf Port D Bit 2,3 und 4
  }
}
//=====

```

8. Kompilieren, Linken, Brennen und Testen

Für die Fertigstellung des Programms wiederholen Sie bitte die Schritte 4. bis 6.

Viel Spaß!