



Zielstellung

In diesem Schnelleinstieg werden Ihnen die ersten Schritte in der Programmierung mit dem myAVR Workpad erläutert. Das Ziel ist es, beim Start des Mikrocontrollers alle optischen Ausgabegeräte in Form der roten, gelben und grünen LED auf dem myAVR Board zum Leuchten zu bringen. Der Schnelleinstieg betrachtet zuerst die Programmiersprache Assembler und geht dann kurz auf C ein.

Als Mikrocontroller wird der ATmega8 verwendet.

Voraussetzungen

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

Software

- myAVR Workpad SE oder myAVR Workpad PLUS
- für die Programmiersprache C/C++ myAVR Workpad PLUS

Hardware

- Ein bestücktes myAVR Board MK2
- Programmierkabel (USB Kabel A-B)

Das Entwicklungswerkzeug myAVR Workpad

Das myAVR Workpad ist ein einfaches Entwicklungswerkzeug, mit dem man schnell und unkompliziert Mikrocontrollerlösungen programmieren kann. Beim Erstellen eines neuen Programms können Sie bereits aus vielfältigen Hilfen wählen, wie z.B.:

- CodeWizard: Sie können interaktiv Ihre Mikrocontroller-Lösung in der gewünschten Zielsprache erstellen
- leere Datei: in einem leeren Editorfenster können Sie den Quellcode erfassen und mit Code-Beispielen aus dem Assistenten ergänzen
- Grundgerüst: eine Datei mit dem Grundgerüst wird geöffnet
- Vorlage: eine Auswahl von vorgefertigten Beispielprogrammen wird Ihnen angeboten

Während der Arbeit mit dem myAVR Workpad stehen Ihnen spezifische Hilfen zur Verfügung, die über die Icons in der Menüleiste des Programms aktiviert werden können (Symbol „?“), z.B.:

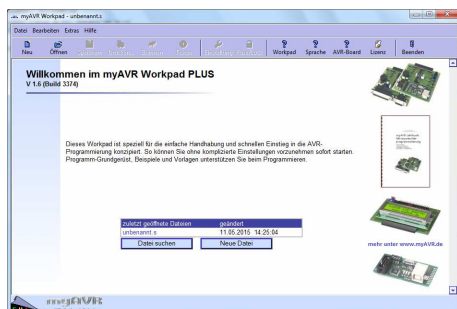
- die Handhabung von myAVR Workpad
- eine Übersicht zu AVR Assemblerbefehlen
- eine Übersicht zum myAVR Board



Zusätzliche Hilfe erhalten Sie während der Quelltexterfassung in Form von Tooltipp zum jeweils verwendeten Befehl.

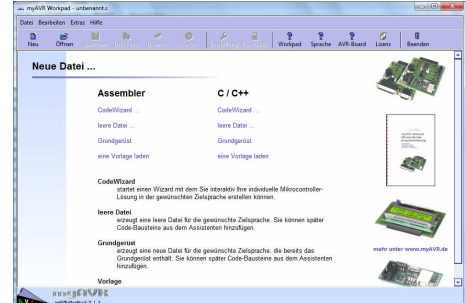
1. Eine neue Datei anlegen

Starten Sie myAVR Workpad und aktivieren Sie die Schaltfläche *Neue Datei*. Legen Sie eine *leere Datei* ... an und speichern Sie diese unter dem Namen „Alle_Lichter_an“.



2. Assembler-Programm erstellen

Wählen Sie im nächsten Fenster unter „Assembler“ die Vorlage „Grundgerüst“. Laden Sie die Vorlage für das Programmgerüst oder erstellen Sie den entsprechenden Quellcode.



3. Quellcode in Assembler bearbeiten

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

Ergänzen Sie die Programmvorlage, mit nachstehend aufgeführtem Quellcode.

```

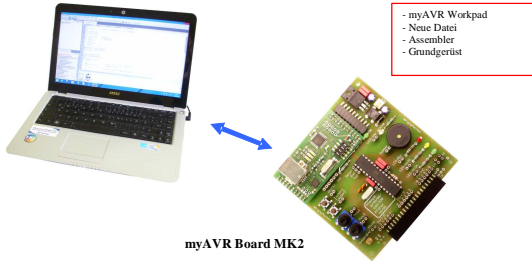
;+-----+
;| Titel           : Übung: „alle Lichter an“
;+-----+
;| Prozessor      : ATmega8 3,6864 MHz
;| Sprache        : Assembler
;| Datum         : 28.02.2006
;| Autor         : Dipl. Ing. Päd. Alexander Huwaldt
;+-----+
.include "AVR.H"
;+-----+
;Reset and Interrupt vector ;VNr. Beschreibung
rjmp 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. Hardware anschließen

Das myAVR Board MK2 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.

Verbinden Sie das Board mit Hilfe des Programmierkabels an den USB-Port Ihres Rechners.



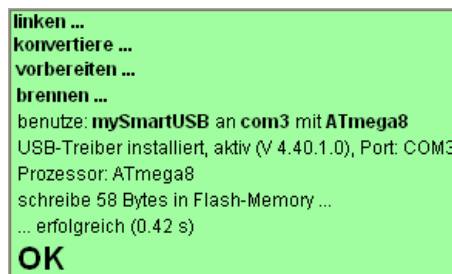
Aktivieren Sie den Menüpunkt *Extras/Einstellungen*. Im sich öffnenden Fenster suchen Sie Ihren Programmierer sowie den verwendeten Controller aus; testen Sie die Verbindung. Bestätigen Sie Ihre Auswahl zum Abschluss mit *Speichern*.



5. Kompilieren, Linken und Brennen

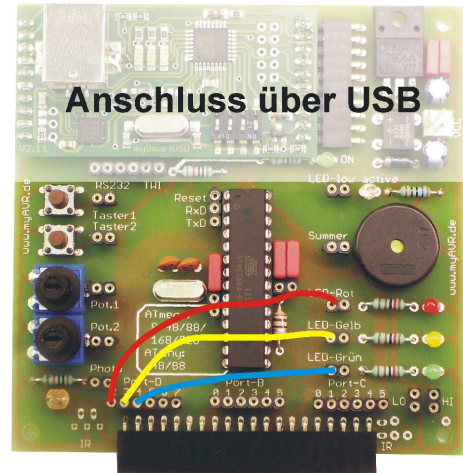
Der eingegebene Quellcode muss nun in Maschinencode für den AVR Prozessor übersetzt und auf den FLASH-Programmspeicher des Prozessors gebrannt werden. Wählen Sie dazu die Schaltfläche *Brennen*. Bei fehlerfreier Übersetzung liegt das Programm unter dem Namen „Alle_Lichter_an.hex“ vor und wird auf dem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

In Abhängigkeit Ihrer Konfiguration erhalten Sie im Ausgabefenster eine entsprechende Meldung:



6. Mikrocontrollerlösung testen

Zum Testen des Programmes ist es erforderlich, Port D mit den Ausgabegeräten zu verbinden. Ziehen Sie das Programmierkabel ab. Verbinden Sie die LEDs mit dem Prozessorport D, wie in der Abbildung dargestellt. Verwenden Sie Patchkabel!



Prüfen Sie die Verbindungen und schließen Sie das USB-Kabel wieder an. Das Programm startet automatisch und die LEDs auf Ihrem Board leuchten.

7. Quellcode in der Sprache C erstellen

Hinweis: Das Programmieren mit C wird nur von der Version myAVR Workpad PLUS unterstützt.

Auf die gleiche Funktionsweise soll nun in der Sprache C programmiert werden. Legen Sie dazu wiederum eine neue Datei an, wählen Sie den Namen „Alle_Lichter_an_C“ und geben Sie folgenden Quellcode in das Quelltextfenster ein:

```
//-----
// Titel   : Beispiel: „alle Lichter an_C“
//-----
// Datum  : 01.03.2006
// Version : 1.2
// Autor   : Dipl. Ing. Päd. Alexander Huwaldt
//-----
#include <avr/io.h>

#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
    {
        wdt_reset();
        PORTD=0x1C;   // auf Port D Bit 2,3 und 4
    }
}
//=====
```

8. Kompilieren, Linken, Brennen und Testen

Das geschriebene C-Programm muss auch in die Maschinensprache übersetzt und auf den Mikrocontroller gebrannt werden.

Wiederholen Sie für den Brennvorgang die oben beschriebenen Schritte 4 bis 6. Nach erforderlicher Verbindung mittels Patchkabeln und dem Anschluss von des Kabels leuchten alle LEDs auf dem Board.

Viel Spaß!