



### Goal

This quickstart will explain you the first step of programming with the myAVR Workpad.

The goal is to make all output devices on the myAVR Board, in form of red, yellow and green LED become glowing.

The quickstart consider first of programming language Assembler and then comes shortly of C.

We are using Atmega8 Softwarecontroller in our examples.

### Requirements:

To work through this quickstart, you will need this following software and hardware :

#### Software:

- myAVR Workpad SE or myAVR Workpad PLUS
- For the programming language C/C++ myAVR Workpad PLUS

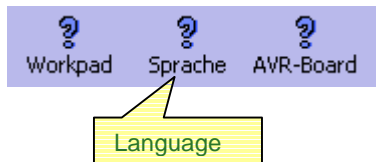
#### Hardware:

- A full equipped myAVR Board
- Programming cable (USB/LPT)
- An external power supply unit resp. a 9V battery

### Help in Workpad

During working with the myAVR Workpad you can activate the specific help by clicking the icon Symbol "?" in the menu bar of the program. These information are available :

- The handling from Workpad
- An overview to AVR Assembler commands
- An overview to myAVR Board



You can get the additional help during entering the source code in form of Tooltips to the actual command.

### 1. Create a new File

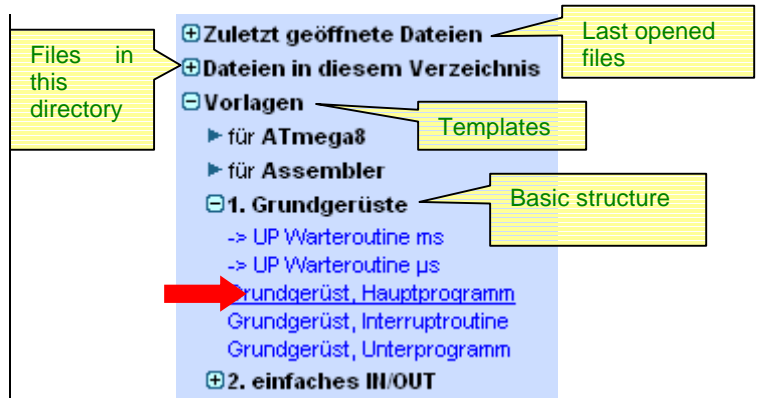
To create a new file, start myAVR Workpad and choose "Neue Datei" (new file), then you can open an empty file and save it under name "all\_lights\_on".

### 2. Create a new assembler program

You will see various examples and frameworks on the left window that you can refer to while doing programming with myAVR Workpad.

Please choose Vorlagen/Grundgerüste/Grundgerüst, Hauptprogramm (Template/basic structure/main program) through the navigation bar.

The basic structure is then loaded and available unlimited for further processing.



### 3. Editing the source code in Assembler

The output device (LEDs) should be switched over from Processor-port D.

Therefore we need to define a few pins of port D for output. You can follow this procedure (and use of register 16) while you completing the source code as shown on this below picture.

```

;-----+
;| 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
;-----+

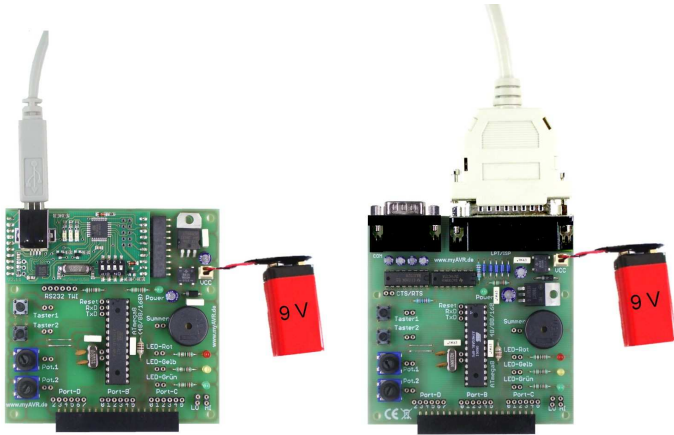
mainloop: wdr
          ldi r16,0b00011100 ; Bit 2-4 auf HIGH
          out PORTD,r16 ; Port D.2-4 auf HIGH
          rjmp mainloop
;-----+

```

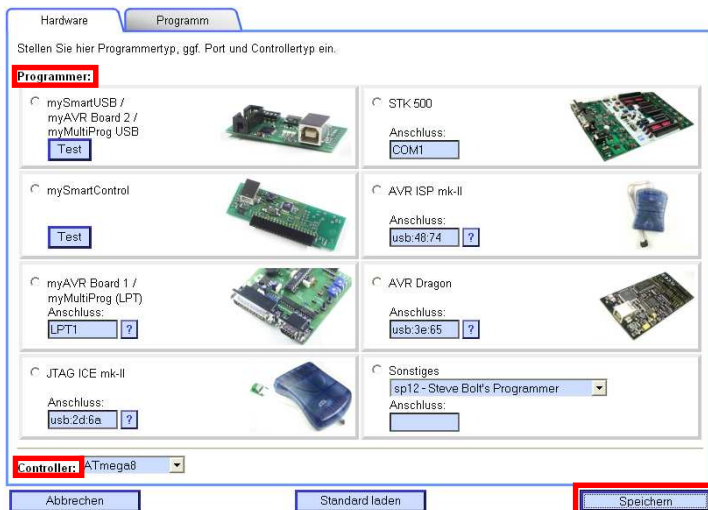
#### 4. Connect the hardware

The myAVR Board has an ISP (In System Programming) interface. That means you don't need to disconnect the microcontroller from the system if you want to program it.

You just need to connect the programming cable (LPT or USB – depending on your myAVR Board) to board and computer. An external power supply via PSU or battery might also necessary.



Please choose menu *Extras/Einstellungen* (Setting). From there you can search and choose the programmer and controller that you use, following by "Speichern" (save) to confirm your choice.



#### 5. Compile, link and burn

The assembler source code has to be translated into machine code first, in order to burn it on the FLASH-memory of the processor. To do so, push the button "Brennen" (Burn) on the toolbar. A HEX-file will be generated and burned on the Processor if the translation has been finished successfully.

You will see a message in the control window, not the same but similar like the message below, depend on your configuration.

```

linken ...
konvertiere ...
vorbereiten ...
brennen ...
benutze: mySmartUSB an com3 mit ATmega8
USB-Treiber installiert, aktiv (V 4.40.1.0), Port: COM3
Prozessor: ATmega8
schreibe 58 Bytes in Flash-Memory ...
... erfolgreich (0.42 s)
OK
  
```

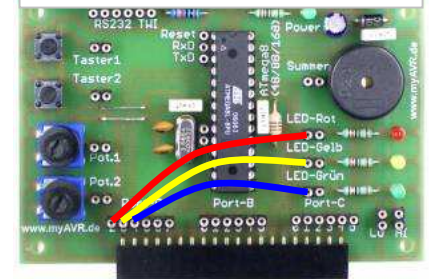
#### 6. Test your Microcontroller program

To test the program, please disconnect the myAVR Board from the external power supply and the programming cable if you join the different pins external. That is for your own safety and to protect the board from over voltage.

It is necessary to connect the Port D to the output device. Use the jumper wire to realize the connection.

Supply your board with power again and you should see glowing LCDs. Congratulations, you have had realized your first assembler program with your myAVR Board.

#### Anschluss über USB oder LPT/COM



#### 7. Create source code in C

Please note : You need at least myAVR PLUS if you want to make a program in C.

You need to do the same procedure to programming in C. You can see on this below picture, the task are the same like the one we made in Assembler.

Please open a new file in myAVR Workpad PLUS and named it "All\_lights\_on\_C" and type the code in the source code window.

```

//-----
// Titel   : Beispiel: „alle Lichter an“
//-----
// 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. Compile, link, burn and test

The C-Program must also be translated into Machine code and burned on the processor.

Please repeat the burn process as above from step 4 until step 6. All LEDs on your board will be lighted after you make a connection of the required Patch cable and the Battery's connection or the Power Supply unit.