

# SiSy® Schnelleinstieg

## Mikrocontroller-Programmierung mit dem SiSy-AVR-Board

### Voraussetzungen:

Für die Bearbeitung der folgenden Aufgaben benötigen Sie folgende Software und Hardware.

Software:

- SiSy ab Version 2.16
- SiSy-Ausgabe AVR; Developer, Professional oder BS und *das installierte Add-On AVR*

Hardware:

- Ein bestücktes SiSy-AVR-Board
- LPT Verlängerungskabel
- Serielles Nullmodemkabel
- 9V Batterie oder Netzteil

Sie sollten mit SiSy und dem Add-On Simple Programmierung vertraut sein. Arbeiten Sie zum Beispiel die entsprechenden Schnelleinstiege ab.

### Zielstellung:

Der Schnelleinstieg Mikrocontroller-Programmierung soll Ihnen helfen, das Add-On AVR kennen zu lernen und erste Schritte in der hardwarenahen Programmierung mit SiSy zu gehen.

### 1. Ein neues Projekt anlegen

Starten Sie SiSy und wählen Sie *Assistent öffnen*. Wählen Sie den Menüpunkt *Neues Projekt anlegen*, vergeben Sie den Projektnamen „AVR\_Test1“ und bestätigen Sie mit *OK*. Wählen Sie das Vorgehensmodell „Programmierung“ und laden Sie keine Diagrammvorlage.

### 2. Kleines Assembler-Programm anlegen

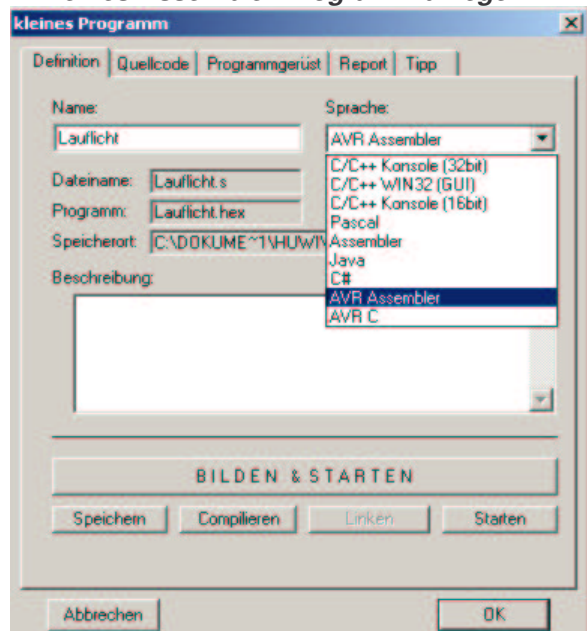


Abbildung: Dialog „kleines Programm“

Erstellen Sie ein Programm für den AVR Mikrocontroller, in dem Sie per Drag & Drop aus der Objektbibliothek ein „kleines Programm“ in das Diagrammfenster ziehen. In dem aufgeblendeten Dialogfenster vergeben Sie den Namen „Lauflicht“. Wählen Sie die Sprache „AVR Assembler“. Über die Registerkarte „Programmiergerüst“ kön-

nen Sie die Vorlage „Grundgerüst“ für ein AVR Assemblerprogramm laden.

```

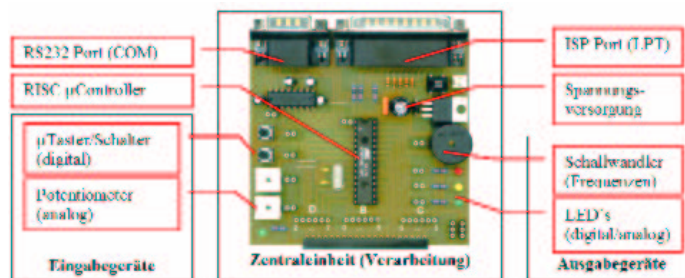
+-----+
| Title      : Programmkopf für SiSy-AVR-Board |
+-----+
| Prozessor  : ATmega8 3,6864 MHz             |
| Sprache    :                               |
| Datum     :                               |
| Autor     :                               |
+-----+
;include "AVR.H"
;-----+
; Reset and Interrupt 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 (I2C) Serial Interface
; reti ; 19 Store Program Memory Redy
;-----+
; Start, Power ON, Reset
main: ldi r16, lo8(RAMEND)
      out SPL, r16
      ldi r16, hi8(RAMEND)
      out SPH, r16
      ; Hier Init-Code eintragen.
;-----+
mainloop: wdr
          ; Hier den Quellcode eintragen.
          rjmp mainloop
;-----+

```

Laden Sie die Vorlage für das Programmgerüst oder erstellen Sie den entsprechenden Quellcode.

### 3. Quellcode in Assembler erstellen

Das SiSy-AVR-Board verfügt über folgende Komponenten:



Das Lauflicht soll über die Ausgabegeräte LED angezeigt und vom Prozessorport D gesteuert werden. Die Realisierung erfolgt über ein Bit im Register R18. Dieses wird mit dem Befehl Bit-Rotation nach rechts verschoben und an Port D des Prozessors ausgegeben. Auf Grund der Prozessorgeschwindigkeit muss die Ausgabe des Lauflichtes für unser Auge verzögert werden. Geben Sie folgenden Quellcode ein bzw. ergänzen Sie die Programmvorlage!

```

;+-----+
;| Title       :Lauflicht für SiSy-AVR-Board
;+-----+
;| Prozessor   :ATMEGA8 3,6864 MHz
;| Sprache     :Assembler
;| Datum       :13.06.2004
;| 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 (I²C) Serial Interface
reti    ; 19 Store Program Memory Redy
;+-----+
; Start, Power ON, Reset
main:
ldi     r16 , lo8(RAMEND)
out     SPL , r16 ;Init Stackpointer
ldi     r16 , hi8(RAMEND)
out     SPH , r16
ldi     r16 , 0b11111111 ;Port D auf Ausgang
out     DDRD , r16
ldi     r16 , 0b00000000 ;Alle BITS auf LOW
out     PORTD , r16
ldi     r17 , 0b00000000
ldi     r18 , 0b00000001 ;1 Lauflicht Bit
;+-----+
mainloop:
wdr
inc     r16 ; wait
brne   skip
inc     r17 ; wait
brne   skip
rcall  up1 ; Lauflicht
skip:
rjmp   mainloop
;+-----+
up1:
ror     r18 ; Bit-Rotation
out     PORTD , r18
ret
;+-----+

```

#### 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 als „Lauflicht.hex“ vor und kann auf den FLASH-Programmspeicher des Prozessors gebrannt werden.

#### 5. Hardware anschließen und brennen

Das SiSy-AVR-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 SiSy-AVR-Board direkt programmiert werden. Dazu schließen Sie das Programmierkabel an den LPT-Port Ihres Rechners und stecken Sie die Batterie / das Netzteil an die Stromversorgung des SiSy-AVR-Board. Zum Brennen wählen Sie die Schaltfläche „Brennen“. Bei erfolgreichem Brennvorang erhalten Sie im Ausgabefenster folgende Meldung:



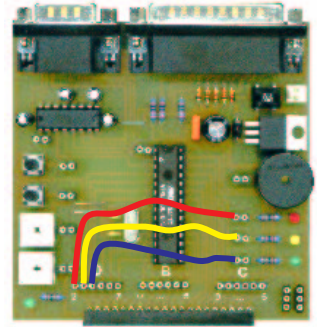
```

Formatierte Ausgabe | Ausgabe
avrduide: load data flash data from input file Lauflicht.hex:
avrduide: input file Lauflicht.hex contains 74 bytes
avrduide: reading on-chip flash data:
Reading |#####| 100% 0.03s
avrduide: verifying ...
avrduide: 74 bytes of flash verified
avrduide done. Thank you.
Ende.

```

#### 6. Mikrocontrollerlösung testen

Um das Programm zu testen ist es nötig den Port D mit den Ausgabegeräten LED's zu verbinden. Ziehen Sie die Batterie/das Netzteil und das Programmierkabel ab. Verbinden Sie die LED's mit dem Prozessorport D entsprechend dem folgenden Schema. Nutzen Sie den Schaltdraht!



Prüfen Sie die Verbindungen und schließen Sie die Batterie/das Netzteil wieder an. Gratulation! Das ist Ihre erste Mikrocontrollerlösung mit dem SiSy-AVR-Board.

#### 7. Quellcode in der Sprache C erstellen

Das SiSy-AVR-Board lässt sich auch mit der Sprache C programmieren. Legen Sie dazu wiederum ein kleines Programm an. Wählen Sie die Sprache AVR C und geben Sie folgenden Quellcode ein:

```

//-----+
// Title       :LAUFLICHT für ein AVR ATMEGA8L C-Programm
// Date        :28.04.2004
// Version     :1.0
// Autor       :Dipl. Ing. Päd. Alexander Huwaldt
//-----+
#include <io.h>
#include <avr/wdt.h>

void init(void)
{
    PORTD=0x04;
    DDRD=0x1C;
}
void delay ()
{
    int i=0;
    while (i<30000)
        i++;
}
main (void)
{
    init();
    while (true)
    {
        wdt_reset();
        delay();
        if (PORTD==0x10)
            PORTD=0x04;
        else
            PORTD+=PORTD;
    }
}
//-----+

```

#### 8. Kompilieren, linken, brennen und testen

Für das Übersetzen und Brennen des Programms wiederholen Sie die Schritte 4. bis 6. Viel Spaß!

#### Hinweis

Weitere Beispiele, Hinweise und interessante Lösungen bietet Ihnen der Assistent von SiSy.