

Benutzerhandbuch

SiSy[®]
AVR

Add-On Version: 1.4
Ab SiSy Version: 2.17

Die in diesem Handbuch enthaltenen Angaben können im Zuge einer Weiterentwicklung ohne vorherige Ankündigung geändert werden. Laser & Co. Solutions GmbH geht damit keinerlei Verpflichtungen ein.

Die im Handbuch beschriebene Software wird auf der Basis der im Programm-Setup angezeigten Lizenzvereinbarung geliefert und darf nur in Übereinstimmung mit den dort genannten Bedingungen verwendet, kopiert oder weitergegeben werden.

Der Käufer darf nur zu Sicherungszwecken eine Kopie des Programms anfertigen.

Ohne vorherige schriftliche Genehmigung von Laser & Co. Solutions GmbH darf weder das Handbuch noch Teile davon mit elektronischen oder mechanischen Mitteln, durch Fotokopieren oder andere Aufzeichnungsverfahren oder auf irgendeine andere Weise vervielfältigt oder übertragen werden.

© 2005 Laser & Co. Solutions GmbH

	Name	Datum
erstellt/bearbeitet	Frau Mark	20.06.05
geprüft	Herr Riedel	27.06.05
freigegeben	Herr Huwaldt	28.06.05

Inhalt

1	Einleitung	7
2	Vorbereitung.....	8
2.1	Beschaffung und Installation der Software	8
2.1.1	Voraussetzungen.....	8
2.1.2	Setup von der SiSy-CD	8
2.2	Beschaffen bzw. Herstellen der Hardware	10
3	Arbeiten mit der Entwicklungsumgebung von SiSy	11
3.1	Projektarbeit.....	11
3.1.1	Was ist ein Projekt?.....	11
3.1.2	Neues Projekt erstellen	11
3.1.3	Vorhandenes Projekt öffnen.....	11
3.1.4	Projekt archivieren.....	11
3.1.5	Projektarchiv einlesen	12
3.2	Die Modellierungselemente von SiSy.....	12
3.3	Die Fenster für die Modellierung.....	13
4	Mikrocontroller-Entwicklungsumgebung SiSy.....	15
4.1	Voraussetzungen	15
4.2	Zielstellung.....	15
4.3	Vorgehen	15
5	Entwicklung eines Programmablaufplans mit SiSy.....	21
5.1	Einleitung	21
5.2	Vorbereitung	21
5.3	Aufgabenstellung	23
5.4	Grundstruktur laden	23
5.5	Logik entwerfen.....	23
5.6	Befehle eingeben	25
5.7	Übersetzen, brennen und Test	28
6	Dokumentengenerierung „QuickDok“	30
6.1	Zweck.....	30
6.2	Einstellungen in Word.....	30
6.3	Die Benutzung von QuickDok.....	30
	Anhang: Tastaturbelegung.....	31
	Anhang: Mausoperationen.....	33

1 Einleitung

Sie haben eine Ausgabe des Modellierungswerkzeuges Simple System, kurz SiSy, erworben. Bevor auf die verschiedenen Funktionen des Programms eingegangen wird, noch einige Worte zum vorliegenden Handbuch. Mit Hilfe des Handbuchs werden dem Nutzer die Grundlagen der Bedienung von SiSy erläutert. Der Inhalt, die Gestalt und die Regeln der Modelle werden nur am Rand betrachtet. Das genaue Vorgehen für die Programmierung eines Mikroprozessors wird an einem Beispiel ausführlich beschrieben. Auf die Grundlagen der Mikroprozessorprogrammierung wird im Rahmen dieses Handbuches nicht eingegangen.

Dem Nutzer werden der Einstieg in das Programm erleichtert und die umfangreichen Funktionen von SiSy kurz und verständlich beschrieben. Bei der Arbeit mit SiSy erstellt der Anwender Modelle in Form von Diagrammen und in ihnen enthaltene Symbole. Die Grundlagen der Entstehung und Bearbeitung solcher Diagramme sind Gegenstand der Betrachtung dieses Handbuchs.

Folgende Darstellungs- und Gestaltungsmittel sind für den Nutzer bei der Arbeit mit SiSy besonders wichtig:

- die Diagramme als Fenster zur Ansicht und Bearbeitung von Modellen;
- der Navigator als Fenster zur Steuerung und Bewegung in Modellen;
- der Assistent mit Hilfestellungen zum jeweils geöffneten Diagramm und mit Diagrammvorlagen (wenn vorhanden);
- die Menüs und Schalter für Befehle an Navigator, Diagramm und Objekt im Kontext mit der Modellierung.

Zu den Bezeichnungen im Text:

- Falls ein Menübefehl nur über Untermenüs zu erreichen ist, werden die einzelnen Menübezeichnungen kursiv geschrieben und durch Schrägstriche voneinander getrennt.
Beispiel: Menü *Hilfe/über SiSy*
- Titel von Dialogboxen, Schaltern und Menüpunkten werden in Anführungszeichen gesetzt.
Beispiel: Dialogbox „Definition“, Schalter „OK“

2 Vorbereitung

In diesem Kapitel werden Sie über notwendige Schritte zur Beschaffung, Installation, Konfiguration und Aufbau einer funktionsfähigen Entwicklungsumgebung informiert.

2.1 Beschaffung und Installation der Software

Für die Bearbeitung der Übungen und Aufgaben steht Ihnen die AVR-Entwicklungsumgebung SiSy-AVR zur Verfügung. Sollten Sie SiSy-AVR bereits installiert haben, können Sie dieses Kapitel überspringen.

2.1.1 Voraussetzungen

Für die Installation benötigen Sie einen Freischaltcode (Lizenzangaben). Falls Sie diese Angaben nicht mit der Software erhalten haben, können Sie diese online abrufen von

www.myAVR.de → **Service**

oder fordern Sie diese beim Hersteller an:

Tel: 03585-470222

Fax: 03585-470233

e-Mail: hotline@myAVR.de.

Außerdem sollten Sie prüfen, ob die Systemvoraussetzungen für die Installation und die Arbeit mit SiSy-AVR gewährleistet sind.

- PC-Arbeitsplatz oder Notebook mit LPT-Port und mindestens einem COM-Port
- Prozessor ab 800 MHz
- 300 MB freier Speicherplatz auf der Festplatte
- Windows 2000, XP und 2003
- mindestens 64 MB RAM, empfohlen 256 MB RAM
- Microsoft Internet-Explorer ab Version 5.0
- Maus, oder ähnliches Zeigegerät
- SiSy ab Version 2.16
- Assembler Entwicklungsumgebung (in SiSy bereits integriert)
- myAVR-Board
- LPT-Verlängerung
- Null-Modemkabel
- Bei Bedarf (z.B. autonomer Einsatz des AVR-Boards) geeignete Spannungsversorgung z.B. 9 V Batterie

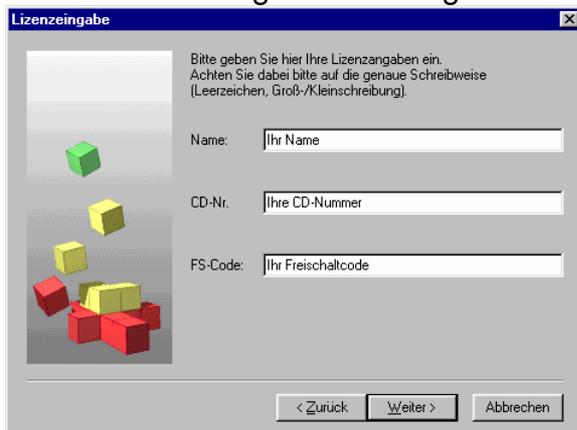
Desweiteren sollten Sie Grundkenntnisse in einer beliebigen Programmiersprache besitzen. Die Installation und der erste Start müssen mit Administratorrechten durchgeführt werden.

2.1.2 Setup von der SiSy-CD

Legen Sie die CD „SiSy“ in Ihr CD-ROM-Laufwerk ein. Falls die CD nicht automatisch startet, wählen Sie bitte im Explorer das CD-ROM-Laufwerk und starten „Cd.exe“ aus der Wurzel des Laufwerks. Das Programm bietet Ihnen vielseitige Informationsmöglichkeiten über SiSy in Form von Präsentationen, Dokumentationen oder Schnelleinstiegen. Betätigen Sie die Schaltfläche „SiSy installieren“. Die Hinweise zur Lizenz sollten Sie lesen, da sie Einfluss auf die Installation bzw. das Starten von SiSy haben. Betätigen Sie danach „Setup starten“. In Abhängigkeit Ihrer Rechnerkonfiguration kann der Start des Setup-Programms einige Sekunden dauern. Das gestartete Setup-Programm wird Sie durch die weitere Installation führen.

Beginn der Installation

Betätigen Sie im Setup-Programm die Schaltfläche „Weiter“. Sie erhalten die Lizenzbestimmungen. Bitte lesen Sie diese sorgfältig durch. Wenn Sie sich mit diesen Bestimmungen einverstanden erklären, bestätigen Sie die Lizenzbestimmungen. Sie werden im folgenden Dialog dazu aufgefordert, Ihre Lizenzangaben einzugeben.



Danach erscheint folgende Dialogbox. Im unteren Teil wird der Pfad verwaltet, in dem SiSy zu installieren ist. Wenn ein anderer Pfad (bzw. ein anderes Laufwerk) gewünscht wird, ist die Schaltfläche „Durchsuchen“ zu betätigen. Eine Dialogbox erscheint; in den Feldern „Laufwerke“ und „Verzeichnisse“ können Laufwerk und Verzeichnis festgelegt werden.

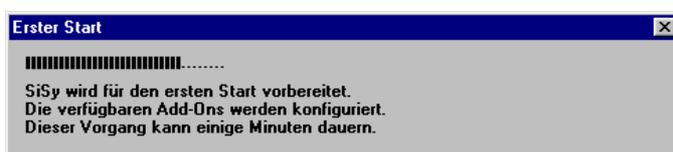


Wählen Sie danach den Programmordner, in dem die Verknüpfungen von SiSy eingefügt werden. Sie können den Zielordner ändern.

Setzen Sie die Installation mit „Weiter“ fort. Der für die Installation erforderliche Speicherplatz wird angezeigt und mit dem auf dem Laufwerk freien Speicherplatz verglichen. Wenn der freie Speicherplatz nicht ausreichend ist, erfolgt eine Warnung und Sie sollten den erforderlichen Speicherplatz zur Verfügung stellen.

Abschluss der Installation

Die Installation ist erst nach dem ersten Start von SiSy abgeschlossen. Dieser sollte im Anschluss an das Setup ausgeführt werden. Der Vorgang kann einige Minuten dauern.



Mit Beendigung dieses Vorgangs erscheint auf Ihrem Bildschirm der Dialog „Willkommen in SiSy“. Folgen Sie dann den Hinweisen des Assistenten.



2.2 Beschaffen bzw. Herstellen der Hardware

Alle Ausführungen, Übungen und Aufgabenstellungen beziehen sich auf das myAVR-Board als Referenzhardware. Wenn Sie Spaß an Elektronik haben, können Sie die Hardware auch selbst fertigen. Die Komponenten erhalten Sie als Bausatz oder fertig bestückt inklusive Schaltplan etc. unter www.myAVR.de.

Ausführliche Beschreibungen zur Programmierung mit Assembler, C/C++, BASCOM sind nicht Inhalt dieses Handbuches. Weiterführende Erklärungen dazu sind im „Lehrmateriel AVR-Assembler“ enthalten.

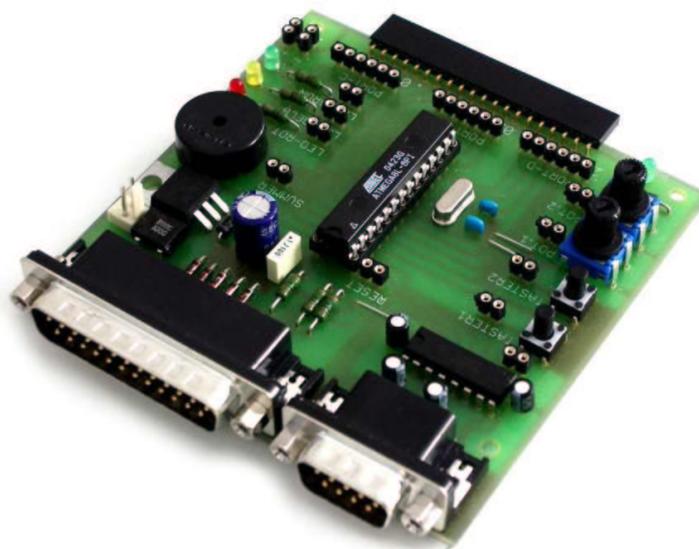


Abbildung 1: myAVR-Board

3 Arbeiten mit der Entwicklungsumgebung von SiSy

Um dem Nutzer zu erläutern, wie er in SiSy modellieren kann, werden zweckentsprechende Definitionen der Begriffe gegeben.

3.1 Projektarbeit

3.1.1 Was ist ein Projekt?

Ein Projekt ist eine abgegrenzte Menge von verknüpften Elementen für ein zu bearbeitendes Problem. Alle aktuellen Daten sind in einer Projektdatenbank gespeichert. Die Projektdatenbank besteht aus einer Anzahl von Dateien im Projektverzeichnis, wobei jedes Projekt sein eigenes Verzeichnis hat. Durch das Anlegen eines Projektarchivs können diese in einer Datei komprimiert werden.

3.1.2 Neues Projekt erstellen

Ein neues Projekt wird eingerichtet und für die Bearbeitung bereitgestellt. Die Definition (Erstellung) eines neuen Projektes erfolgt durch Vergabe eines Projektdateinamens und/oder durch Überschreiben eines alten Projektes gleichen Namens. Über die Schaltfläche „Verzeichnis“ kann die vorgeschlagene Pfadangabe geändert werden.

Hinweis:

Falls ein bereits vorhandener Projekt- oder Verzeichnisname gewählt wird, erfolgt eine Überschreibwarnung.

Nach der Vergabe eines Projektnamens kann im nachfolgenden Fenster ein Vorgehensmodell ausgewählt werden.

3.1.3 Vorhandenes Projekt öffnen

Ein vorhandenes Projekt wird geöffnet und die abgespeicherten Diagramme und Objekte sowie alle Möglichkeiten für die weitere Bearbeitung werden verfügbar gemacht. Die Wahl des Projektes erfolgt durch Klicken auf den entsprechenden Namen oder über die Schaltfläche „Suchen“.

Beim Aktivieren der Checkbox „Beschreibung anzeigen“ werden zum markierten Projekt Informationen angezeigt. Dazu gehören außer der Beschreibung auch Informationen zum Bearbeitungsstand.

3.1.4 Projekt archivieren

Menü *Projekt/Archiv/Anlegen*.

Es kann ein komprimiertes Archiv des Projektes erzeugt werden. Dies ist besonders aus Gründen der Datensicherheit sinnvoll. Zielverzeichnis und Dateiname für die Archivdatei werden vorgeschlagen und können korrigiert werden. Der Umfang der Archivierung ist festzulegen und die Entscheidung für eine Komprimierung zu treffen, wobei diese empfohlen wird. Wenn ein Projekt unter einem bereits vorhandenen Archivnamen angelegt werden soll, wird eine Überschreibwarnung angezeigt. Bei Auswahl von „Nein“ wird die Erstellung des Archivs abgebrochen, bei „Ja“ wird das Projekt archiviert.

Hinweis: SiSy bietet die Möglichkeit des regelmäßigen Abspeicherns verschiedener Arbeitsstände, d.h. ein archiviertes Projekt wird nicht überschrieben. Ein Projektstand kann in einer neuen Archivdatei abgelegt werden

3.1.5 Projektarchiv einlesen

Menü Projekt/Archiv/Einlesen.

Hierunter versteht man das Einlesen eines Archivs zum Zweck der Rekonstruktion des Projektes.

Einlesen bedeutet Entpacken eines archivierten Projektes. Dazu sind der Archivpfad und der Dateiname des Archivs sowie das Zielverzeichnis anzugeben.

Hinweis:

Wenn im Zielpfad des Entpackens bereits ein Projekt existiert, erscheint eine Überschreibwarnung.

3.2 Die Modellierungselemente von SiSy

Werkzeug

SiSy stellt für die Bearbeitung der Modelle und Teilmodelle Werkzeuge der entsprechenden Methodik bereit. Werkzeuge sind Editoren, mit denen in einem Fenster die grafische Darstellung (Diagramme) der Modelle bearbeitet werden kann.

Diagramme

Diagramme sind grafische Darstellungen von Modellen oder Teilmodellen, die mit einem bestimmten Werkzeug erstellt werden. Die Modellierungselemente werden als Objekte in den Diagrammen unter Einhaltung von Regeln zusammengestellt.

Objekte

Objekte sind mögliche Modellelemente in Diagrammen, z.B. „kleines Programm“ in der „Programmierung“. Objekttypen sind konkrete Ausprägungen von Objekten, die in einem Diagramm angelegt wurden, z.B. Objekttyp „Lauflicht“ vom Objekt „kleines Programm“..

Referenzen

Die Objekte eines Diagramms können in anderen Diagrammen wiederverwendet werden. Durch das Hineinziehen aus dem Navigator oder aus einem offenen Diagramm wird eine Referenz vom Originalobjekt erzeugt. Die Referenz ist nur ein Verweis auf das Original, alle angezeigten Informationen wie Kurzname, Langname und Beschreibung werden vom Original bezogen. Somit sind die Informationen in allen Referenzen eines Objektes identisch mit dem Original. Dadurch werden Änderungen dieser Informationen automatisch auf alle Referenzen übertragen. Weiterhin ist es möglich, diese Referenzierung über einen sogenannten Report auszuwerten.

Kanten

Kanten sind Verbindungselemente zwischen Objekten. Eine Verbindung wird durch Ziehen mit der Maus (linke Maustaste) vom Verteiler des selektierten Objektes auf das gewünschte Objekt erreicht. Nach Loslassen der Maustaste und Prüfung der Verbindungszulässigkeit durch SiSy erscheint ein Kanten-Dialog, in dem das Element definiert und individuelle Einstellungen getroffen werden können.

Hinweis:

Bei Verbindung mit gehaltener STRG-Taste wird die Prüfung vernachlässigt und eine Zwangsverbindung erreicht.

Rahmen

Ein Rahmen fasst ausgewählte Objekte des Diagramms optisch zusammen. Er besitzt einen Kurz- sowie Langnamen und eine Objektbeschreibung, kann also als Objekt aufgefasst werden.

Hinweis:

Inhalte von Rahmen sind in Reports oder einer Dokumentengenerierung auswertbar.

3.3 Die Fenster für die Modellierung

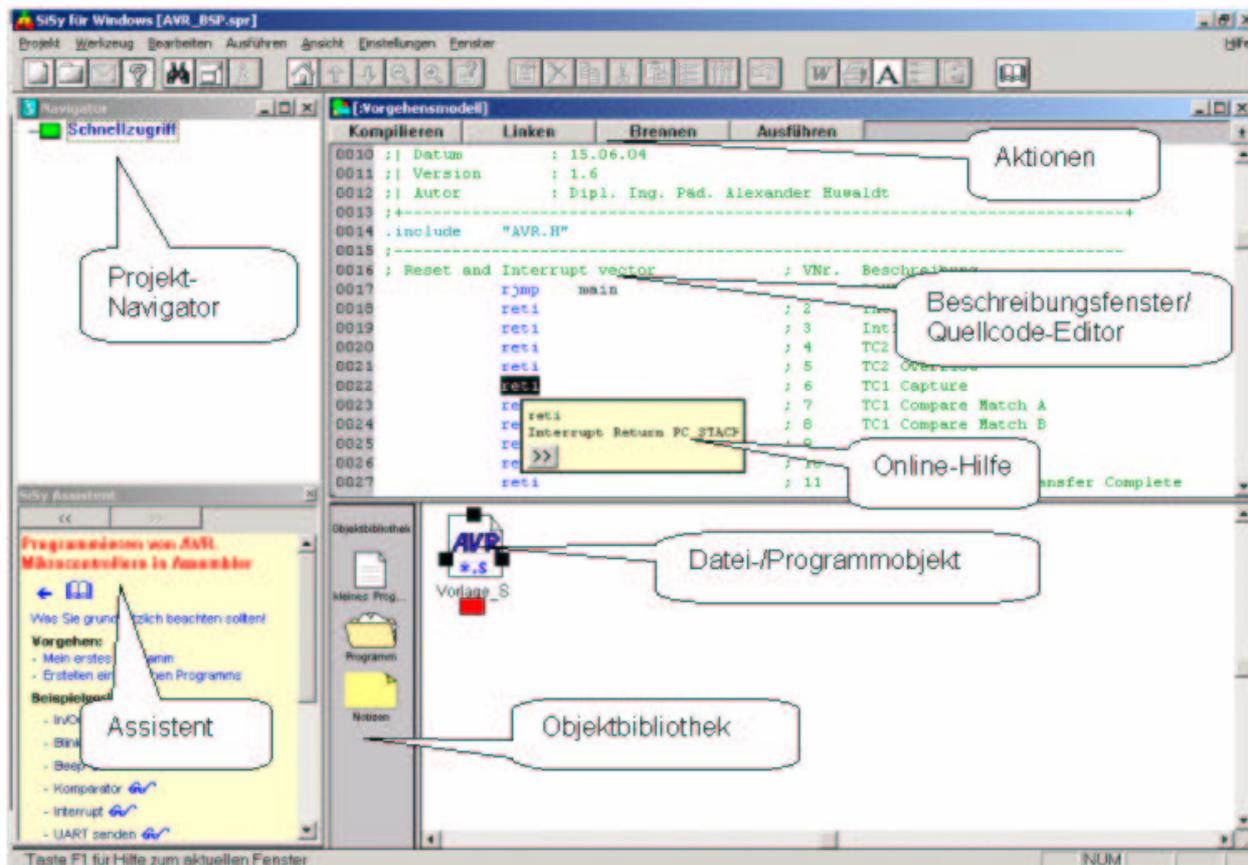


Abbildung 2

Navigator

Dieser befindet sich im linken, oberen Teil des Fensters. Er ermöglicht dem Anwender nach dem Start von SiSy neben dem Vorgehensmodell eine Reihe leicht zu handhabender Schaltflächen, Menüs und weitere Hilfsmittel angezeigt.

SiSy-Assistent

Am linken unteren Bildschirmrand befindet sich diese Nutzerunterstützung.

- Er begleitet den Anwender durch das gesamte Projekt und hält immer passende Informationen zur aktuellen Sicht parat
- Er ist auf die jeweilige Ausgabe von SiSy bezogen.
- Oft können Beispiele als Vorlagen vom Assistenten geladen werden.

Diagrammfenster

Am oberen Bildrand befinden sich das **Hauptmenü**, das hilfreiche Funktionen zu SiSy bereithält, und eine **Werkzeugleiste**, mit deren Hilfe schnell auf nützliche Anwendungen zugegriffen werden kann. Das Diagrammfenster nimmt die rechte Bildschirmseite ein und ist der Raum, in dem der Nutzer modelliert. Es enthält:

- das ausgewählte Vorgehensmodell,
- die Objektbibliothek mit den möglichen Objekten des aktuellen Diagramms sowie
- ein Fenster zur Beschreibung des markierten Objekts, in diesem Fall zum Editieren des Quelltextes.

Die Bedienelemente/Objektbibliothek

SiSy bietet, wie bei Windows-Anwendungen üblich, die Steuerung von Befehlen über das Hauptmenü über die Werkzeugleisten, die Tastatur oder die Objektbibliothek an. Darüber hinaus enthalten das Kontextmenü und der Navigator Steuerfunktionen.

Die Anzahl der möglichen Befehle in der Menüleiste ist abhängig davon, ob ein Projekt geöffnet ist. Ist das nicht der Fall, erscheint ein Menü mit wenigen Befehlen. Bei einem geöffneten Projekt hält SiSy umfangreichere Menüs bereit. Die wichtigsten Menübefehle befinden sich auch als grafische Schalter in der Werkzeugleiste, die eine schnelle und effiziente Bedienung des Programms ermöglicht. Die Toolbox-Darstellung bietet dem Anwender wichtige Programmfunktionen als direkten Link an.

Ein weiteres Bedienelement ist die Objektbibliothek. Sie unterstützt das Anlegen neuer Objekte. Dazu kommt noch der Navigator, welcher in Kapitel ausführlich behandelt wird.

4 Mikrocontroller-Entwicklungsumgebung SiSy

Schauen wir uns als nächstes kurz in der Entwicklungsumgebung SiSy um. Für die Entwicklung von Mikrocontrollerlösungen bietet sich die einfache Programmierung an.

4.1 Voraussetzungen

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

Software:

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

Hardware:

- Ein bestücktes myAVR-Board
- LPT-Verlängerungskabel
- 9 V Batterie bei Bedarf (z.B.: autonomer Einsatz)
- Nullmodemkabel für die COM Schnittstelle
- Patchkabel

4.2 Zielstellung

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

Zielstellung für das erste Beispielprogramm: Die drei LED's auf dem myAVR-Board sollen nacheinander aufleuchten und damit ein „Lauflicht“ erzeugen.

4.3 Vorgehen

Ein neues Projekt anlegen

Starten Sie SiSy, wählen Sie Assistent öffnen und bestätigen Sie mit OK. Wählen Sie den Menüpunkt *Neues Projekt anlegen*, vergeben Sie den Projektnamen „Lauflicht“. Wählen Sie das Vorgehensmodell „Programmierung“. Wählen Sie aus dem Dialog der Diagrammvorlagen „keine Vorlage verwenden“ oder brechen Sie den Dialog ab.

Hinweis:

In SiSy legen Sie stets ein Projekt an. In dieses Projekt integrieren Sie Ihr Programm bzw. mehrere Programme. Unabhängig vom Programmnamen benötigt jedes Projekt einen Namen.

Kleines Assembler-Programm anlegen

Erstellen Sie ein Programm für den AVR-Mikrocontroller, indem Sie per Drag & Drop aus der Objektbibliothek ein „kleines Programm“ in das Diagrammfenster ziehen. Markieren Sie das Objekt und wählen Sie aus dem Kontextmenü (rechte Maustaste) *Definieren*.

Auf der Registerkarte „Definition“ tragen Sie den Programmnamen ein (im Beispiel „Lauflicht“) und wählen die Programmiersprache aus, hier „AVR Assembler“ siehe Abbildung 3.

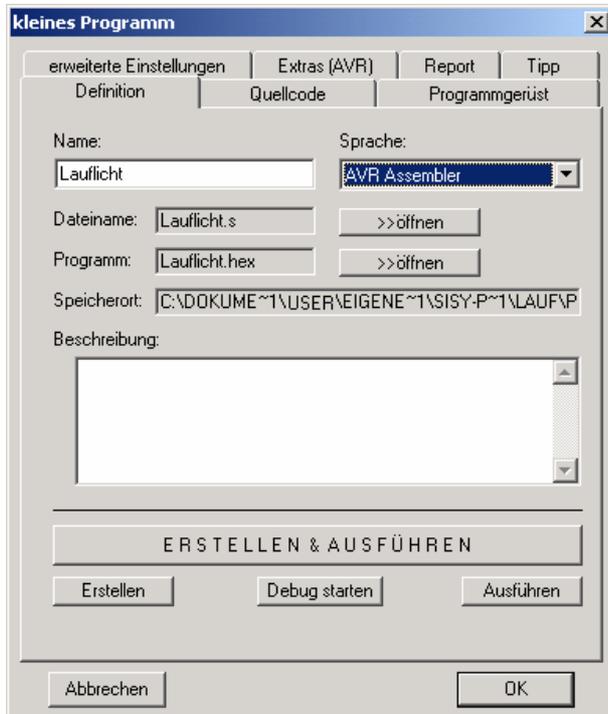


Abbildung 3:

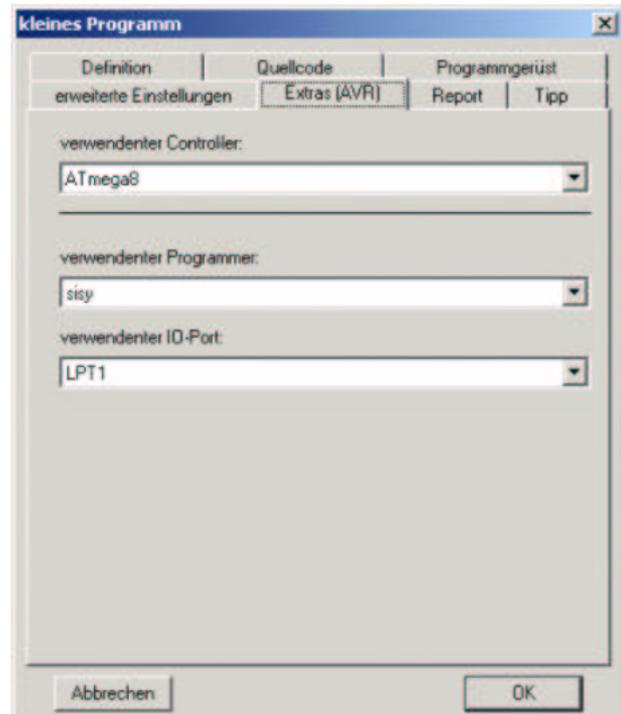


Abbildung 4

Kontrollieren Sie auf der Registerkarte „Extras (AVR)“ den ausgewählten Controller, siehe Abbildung 4

Über die Registerkarte „Programmgerüst“ können Sie das folgende Grundgerüst für ein AVR Assemblerprogramm laden; in der Registerkarte „Quellcode“ können Sie den Quellcode eigenständig eintragen.

Hinweis: Den Quellcode können Sie auch im Beschreibungsfenster/Editorfenster der SiSy-Benutzeroberfläche eintragen bzw. korrigieren.

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

```

;-----
;* Titel           :Programmvorlage für myAVR-Board
;* Prozessor      :ATmega8 mit 3,6864 MHz
;* Schaltung       :...
;* Datum          :15.03.2004
;* Autor          :Dipl. Ing. Päd. Alexander Huwaldt
;-----
.include "avr.h"
;-----
; Reset and Interruptvektoren ; VNr. Beschreibung
begin:      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 LO
         ldi     r16 , hi8(RAMEND)
         out     SPH , r16
         ; Hier den Init-Code eintragen
;-----
mainloop: wdr
          ; Hier den Quellcode eintragen
          rjmp  mainloop
;-----

```

Abbildung 5: Grundgerüst AVR-Assembler Programm

Quellcode in Assembler erstellen

Das Lauflicht soll über die Ausgabegeräte LED angezeigt und von dem Prozessorport D gesteuert werden. Die Realisierung erfolgt über je 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!

```

;-----
;* Titel           :Lauflicht für myAVR-Board
;* Prozessor       :ATmega8 mit 3,6864 MHz
;* Schaltung       :PORT D.2 bis PORT D.4 an LED 1 bis 3
;* Datum          :15.03.2004
;* Autor          :Dipl. Ing. Päd. Alexander Huwaldt
;-----
#include           "avr.h"
;-----

; Reset and Interruptvectoren ; VNr. Beschreibung
begin:    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 LO
           ldi    r16 , hi8(RAMEND)
           out    SPH , r16                ; Init Stackpointer HI
           ldi    r16 , 0b11111111
           out    DDRD , r16               ; PORT D auf Ausgang
           ldi    r16 , 0b00000000
           out    PORTD , r16              ; Alle Bits auf LOW
           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:      rol    r18                      ; Bit-Rotation
           out    PORTD , r18
           ret
;-----

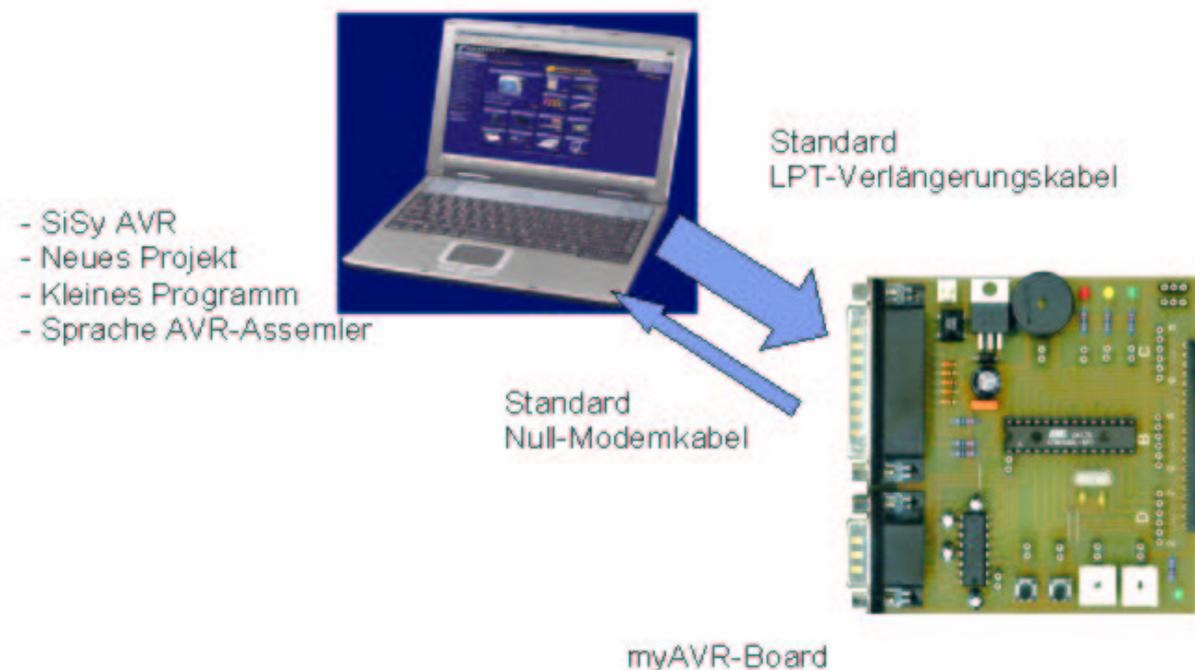
```

Compilieren und Linken

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

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 schließen Sie das Programmierkabel an den LPT-Port Ihres Rechners.



Zum Brennen wählen Sie die Schaltfläche „Brennen“. Bei erfolgreichem Brennvorgang erhalten Sie im Ausgabefenster folgende Meldung:



Abbildung 6: Ausgabefenster

Mikrocontrollerlösung testen

Um das Programm zu testen ist es nötig, den Port D mit den Ausgabegeräten LED zu verbinden.

- Wenn vorhanden ziehen Sie die Batterie/das Netzteil und das Programmierkabel ab.
- Verbinden Sie die LEDs mit dem Prozessorport D entsprechend dem folgenden Schema. Nutzen Sie Patchkabel!
- Prüfen Sie die Verbindungen und schließen Sie die Batterie/das Netzteil oder das Programmierkabel wieder an und nehmen die Mikrocontrollerlösung in Betrieb.
- Es ist jetzt an den LED's ein Lauflicht zu sehen.
- Gratulation!
Das ist Ihre erste Mikrocontrollerlösung mit dem myAVR-Board.

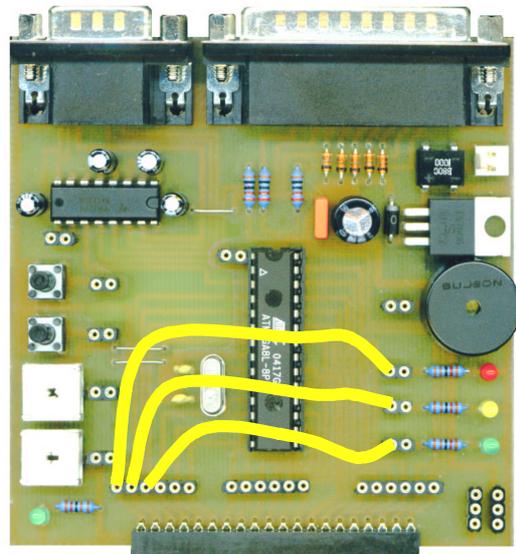


Abbildung 7

Beim Kompilieren, Linken und Brennen des Schnellstart-Beispiels öffnete sich ein Ausgabefenster und zeigte Protokollausgaben der Aktionen an. Wenn die Hardware ordnungsgemäß angeschlossen, von der Software erkannt wurde und das Programm erfolgreich auf den Programmspeicher des Mikrocontrollers übertragen wurde, muss die letzte Ausschrift folgenden Inhalt haben:

```
avrdude: xxx bytes of flash written
...
avrdude: verifying ...
avrdude: xxx bytes of flash verified
avrdude done: Thank you
Ende.
```

Abbildung 8: Ausgabefenster mit "Brenn"-Protokoll

5 Entwicklung eines Programmablaufplans mit SiSy

5.1 Einleitung

Für die Entwicklung eines Programmablaufplans (PAP) sind konkrete Vorstellungen über die Systemlösung und Kenntnis der Hardware nötig. Ein Programmablaufplan kann aus einer genauen Aufgabenstellung abgeleitet werden.

Beispielaufgabe:

Entwickeln Sie eine Mikrocontrollerlösung, bei der ein Taster eine LED schaltet. Der Controller ist so zu initialisieren, dass an Port B.0 der Taster und an Port B.1 die LED angeschlossen ist. Danach ist fortlaufend der Taster abzufragen. Wenn der Taster gedrückt ist, wird die LED eingeschaltet, sonst bleibt die LED aus.

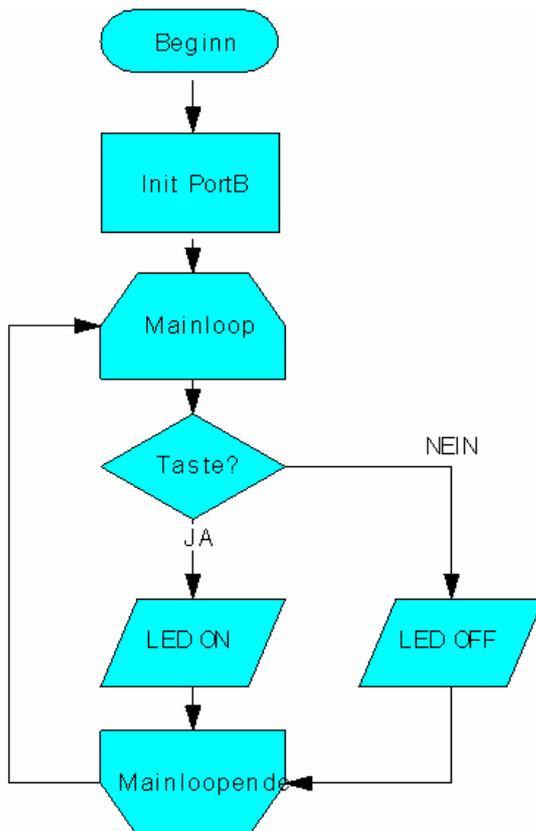
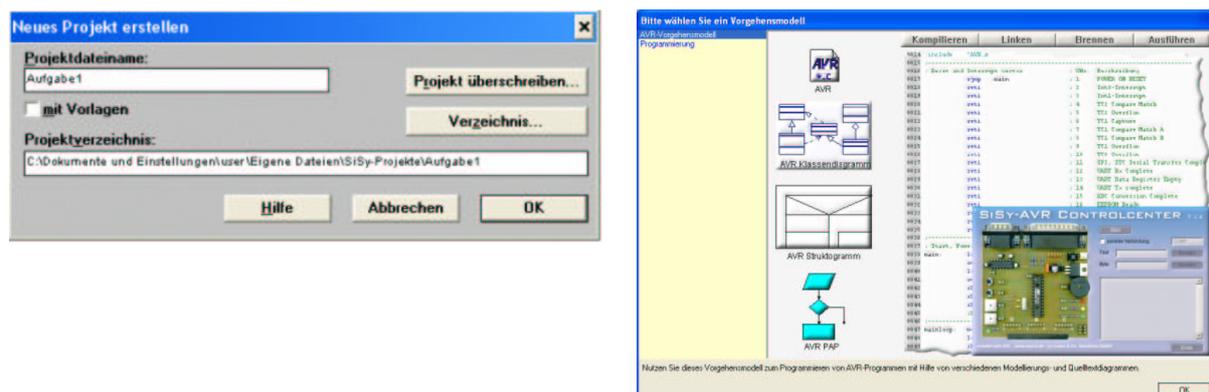


Abbildung 11: PAP zur Beispielaufgabe

5.2 Vorbereitung

Starten Sie SiSy und legen ein neues Projekt an. Wählen Sie das AVR-Vorgehensmodell.



Danach öffnet sich die typische Benutzeroberfläche von SiSy mit einem leeren Vorgehensmodell und Sie können mit der Arbeit beginnen. Falls Sie noch die Option „Menü bei Doppelklick“ und „Direkthilfe“ eingeschaltet haben, können Sie diese über den Menüpunkt „Einstellungen“ im Hauptmenü abschalten.

Ziehen Sie als nächstes aus der Objektbibliothek ein Objekt vom Typ *PAP* in das leere Diagramm. Benennen Sie den PAP mit „Aufgabe 1“. Beachten Sie die Einstellungen zum Controllertyp und Programmieradapter unter „Extras (AVR)“; vgl. Abbildung 12. In den Dialog „Definieren“ kommen Sie über das Rechte-Maustasten-Menü auf dem Objekt.

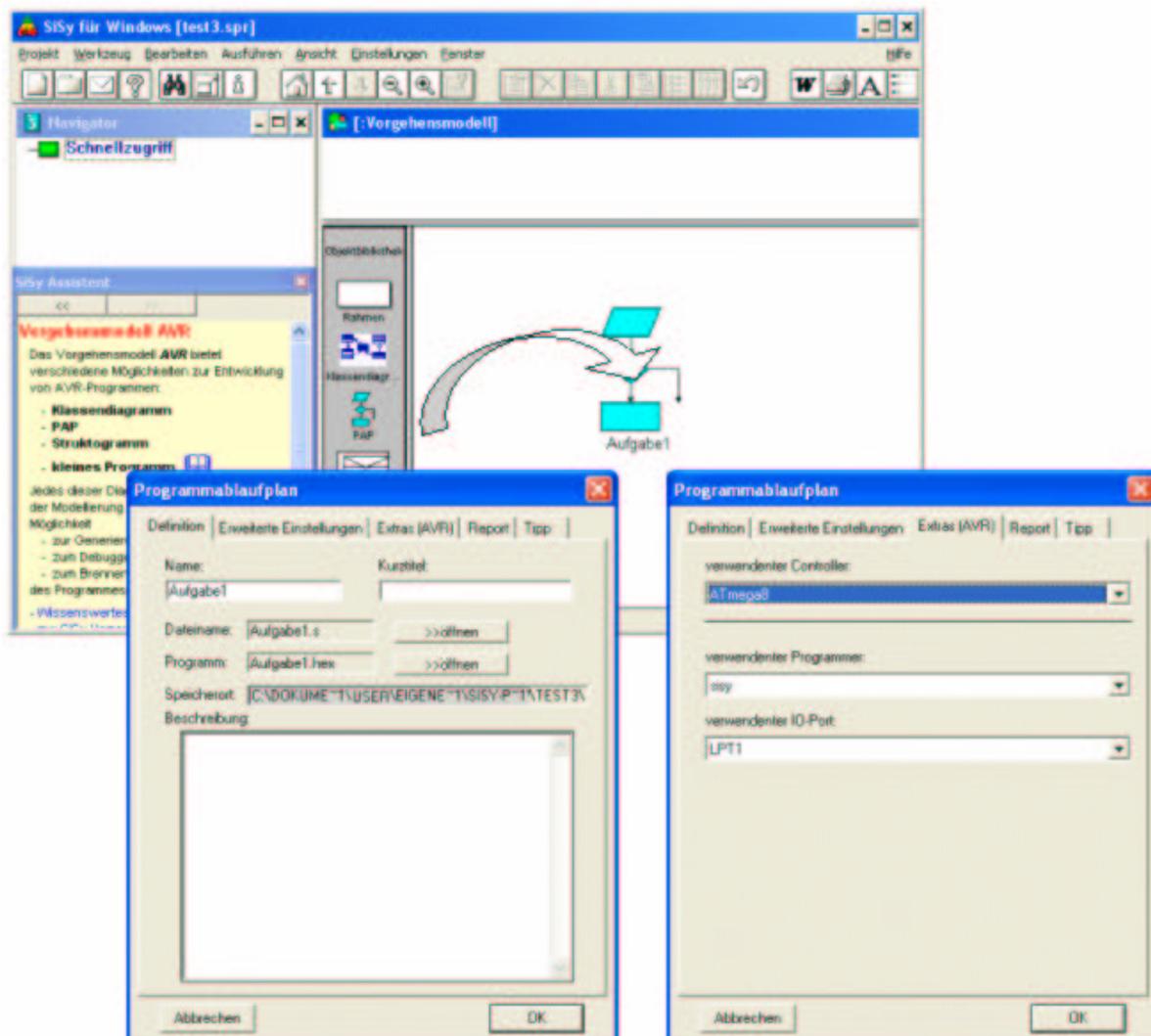
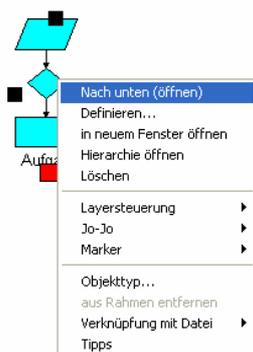


Abbildung 12



Der nächste Schritt ist das Aufstellen des Programmablaufplanes. Dazu muss das Diagramm unter dem Symbol geöffnet werden. Wählen Sie rechte Maustaste „nach unten (öffnen)“ um in dieses Diagramm zu gelangen

Abbildung 13

5.3 Aufgabenstellung

Es ist im ersten Schritt eine einfache Mikrocontroller-Anwendung mit der Technik des Programmablaufplanes zu entwerfen und in der Sprache Assembler zu realisieren.

Aufgabe:

Entwickeln Sie eine Mikrocontrollerlösung, bei der ein Taster eine LED schaltet.

Schaltung:

Port B.0 = Taster 1

Port B.1 = LED

5.4 Grundstruktur laden

Wenn ein Diagramm leer ist, bietet SiSy typische Vorlagen zum Importieren an, siehe Abbildung 15. Diese können dann weiterbearbeitet werden. Wählen Sie die Diagrammvorlage „Mainprogramm-Grundgerüst“. Abbildung 14 zeigt den PAP zu diesem Grundgerüst.

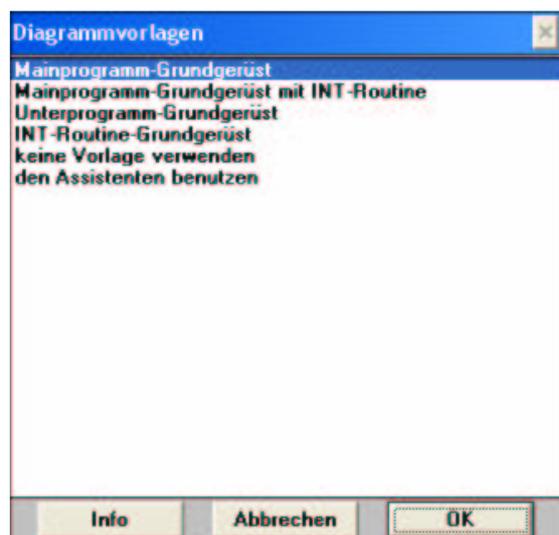


Abbildung 15: Diagrammvorlagen

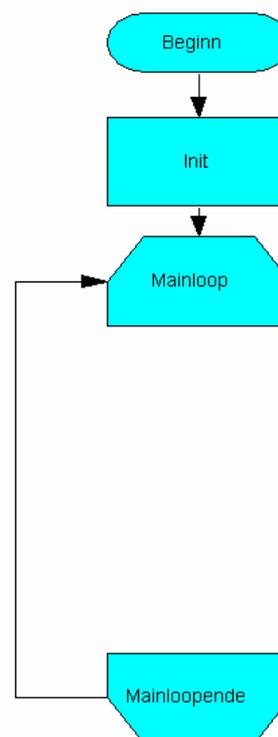


Abbildung 14

5.5 Logik entwerfen

Um die Programmlogik im PAP abzubilden, muss die Vorlage um die fehlenden Elemente ergänzt werden. Des Weiteren sind die Elemente durch gerichtete Verbindungen (Kanten) in der Reihenfolge ihrer Abarbeitung zu verbinden.

Um ein Objekt im Diagramm zu ergänzen, wird der entsprechende Objekttyp in der Objektbibliothek mit der Maus ausgewählt und per Drag & Drop an die entsprechende Position im Diagramm gezogen. Objekte können mit dem Mauscursor per Click selektiert und mittels Drag & Drop auch verschoben werden. Selektierte Objekte lassen sich mit der Taste „Entf“ löschen. Verbindungen zwischen den Objekten können über den rot markierten „Verteiler“ von selektierten Objekten hergestellt werden. Dazu ist das Ausgangsobjekt zu selektieren mit dem Mauscursor auf den roten Verteiler zu klicken und bei gedrückter linker Maustaste eine Verbindung zum Zielobjekt zu ziehen. Um Objekte zu benennen, können Sie einen Doppelklick auf dem betreffenden Objekt durchführen oder über rechte Maustaste in den Dialog *Definieren* gelangen.

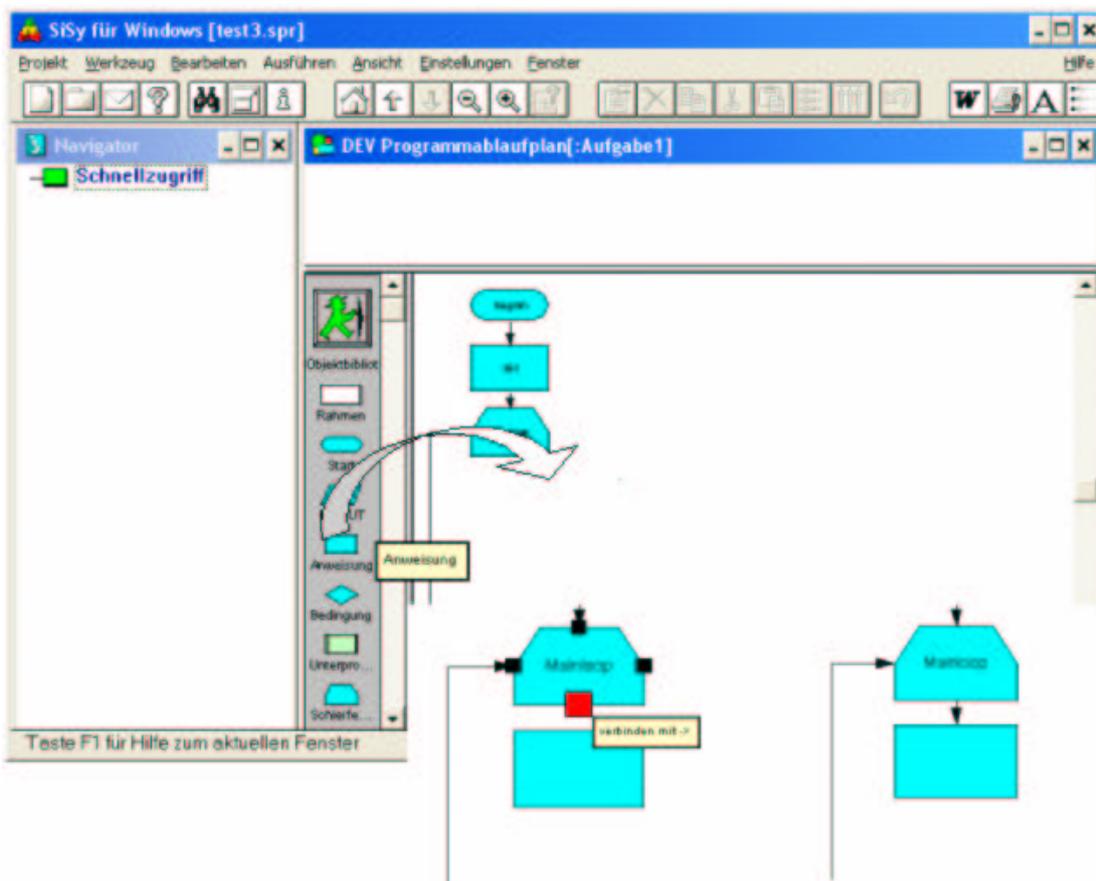


Abbildung 16

Zeichnen Sie den folgenden Programmablaufplan (vgl. Abbildung 17):

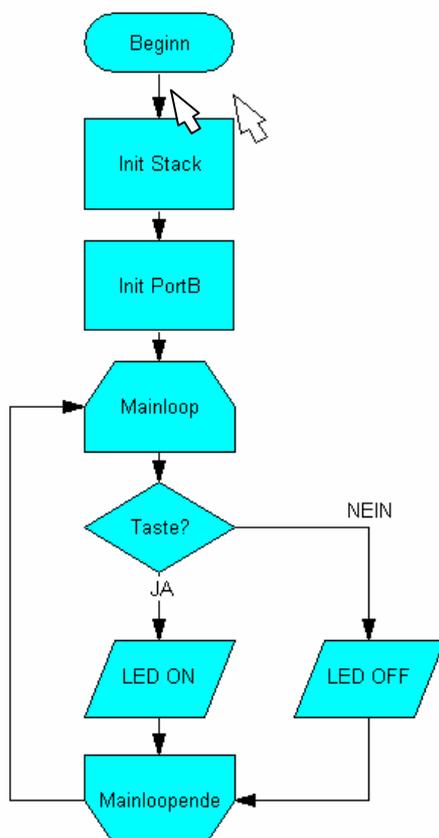


Abbildung 17

5.6 Befehle eingeben

Nachfolgend soll aus dem Programmablaufplan Assembler Quellcode generiert werden. Dazu ist es nötig, die einzelnen Elemente des PAP mit den entsprechenden Assembleranweisungen zu versehen. Dafür gibt es mehrere Möglichkeiten.

Zum einen bietet SiSy beim ersten Öffnen eines jeden Elementes typische Code-Vorlagen an, die über die Schaltfläche „Laden“ dem Element zugewiesen werden können. Wird der Definieren-Dialog mit der Schaltfläche „OK“ beendet, so wird die Auswahl im Objekt gespeichert und beim nächsten Aufruf des Dialoges „Definieren“ erscheinen die Code-Vorlagen nicht mehr, sondern das Element kann ganz normal bearbeitet werden. In Abbildung 18 und Abbildung 19 sind beide Varianten des Dialoges „Definieren“ zu sehen“.

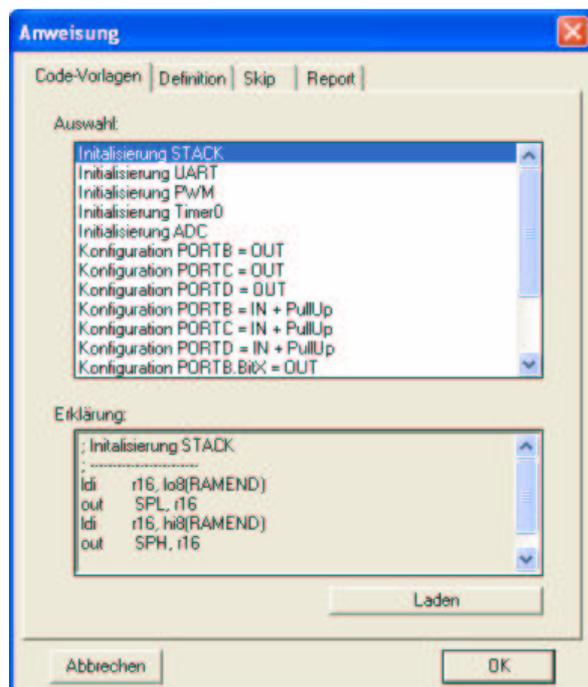


Abbildung 18

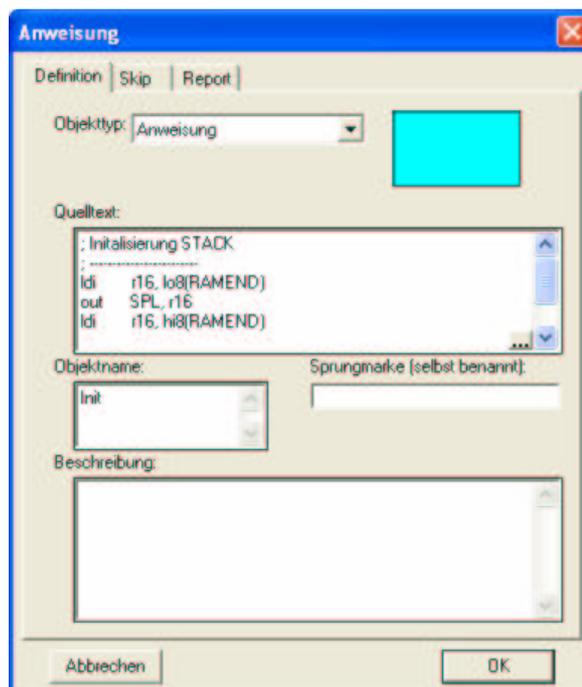


Abbildung 19

Die zweite Möglichkeit besteht beim Selektieren von Elementen über den Quellcodeeditor oberhalb des Diagrammfensters, vgl. Abbildung 20.

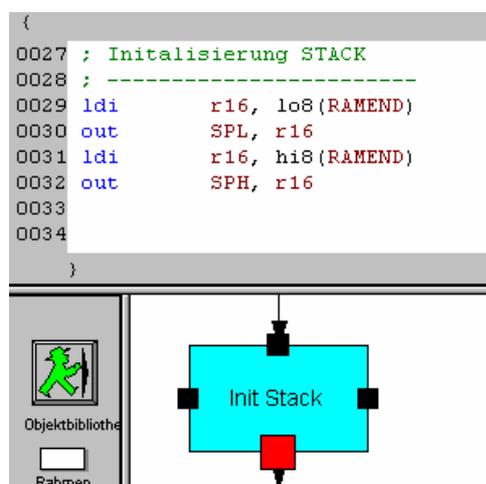


Abbildung 20

Geben Sie die gezeigten Quellcodes in die Objekte ein!

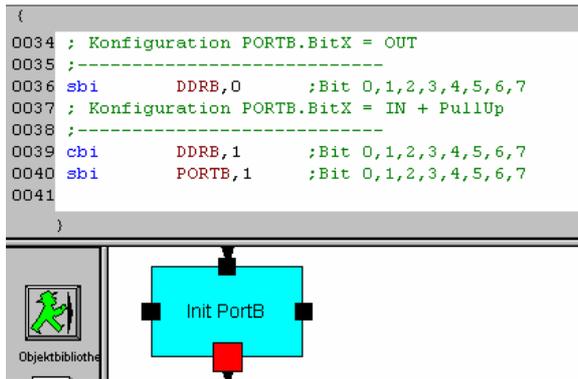


Abbildung 21

Bedingungen haben spezielle Vorlagen, die eine Codegenerierung bei übersichtlichem Programmablaufplan vereinfachen.

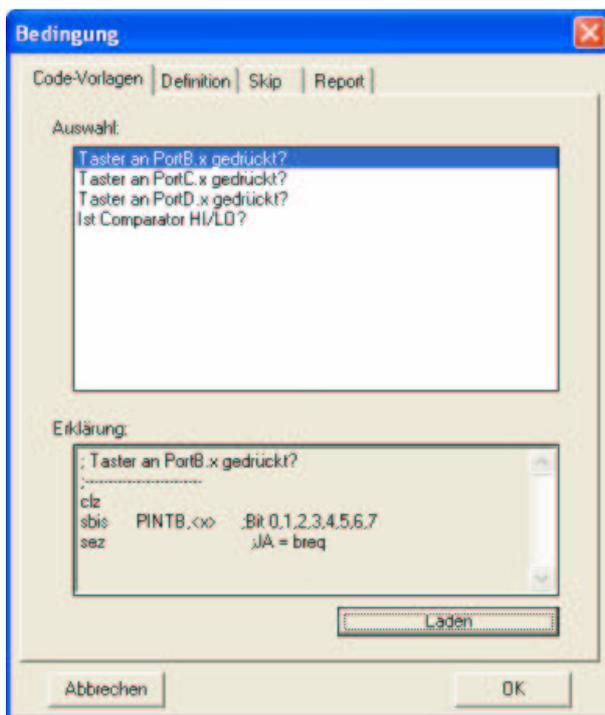


Abbildung 22

Jede Bedingungs-vorlage ist so konstruiert, dass eine JA/NEIN Entscheidung erzeugt werden kann. Findet der Codegenerator das Schlüsselwort JA an einer der folgenden Verbindungen, setzt er diese in eine Sprunganweisung „breq“ um. Das Schlüsselwort NEIN wird in „brne“ umgewandelt. Alternativ können statt dieser Schlüsselworte auch der Sprungbefehl selber an eine der Kanten geschrieben werden (breq, brne, brge, brlo, usw.)

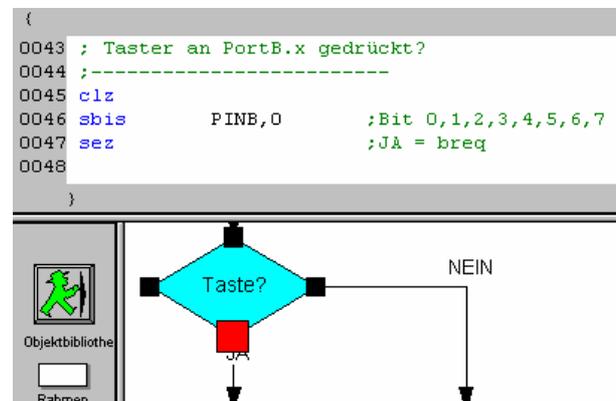


Abbildung 23

Auch das Element „IN/OUT“ verfügt über spezifische Vorlagen. Diese sind gegebenenfalls mit zu ergänzen. Dazu sind spitze Klammern als Platzhalter in den Vorlagen eingefügt.

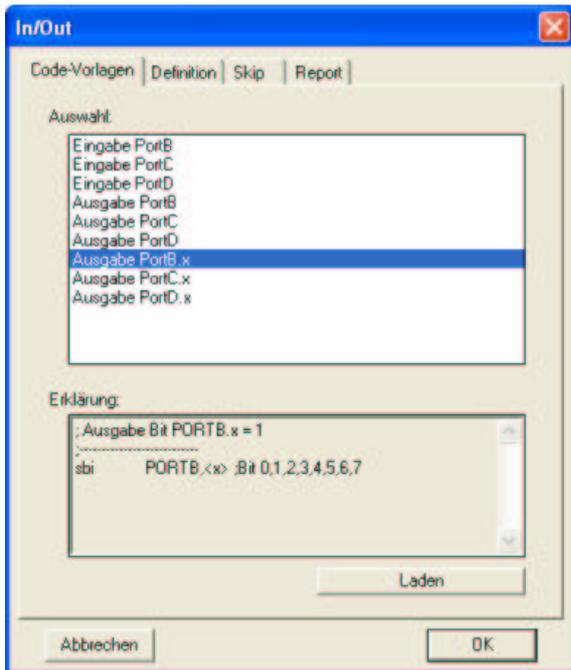
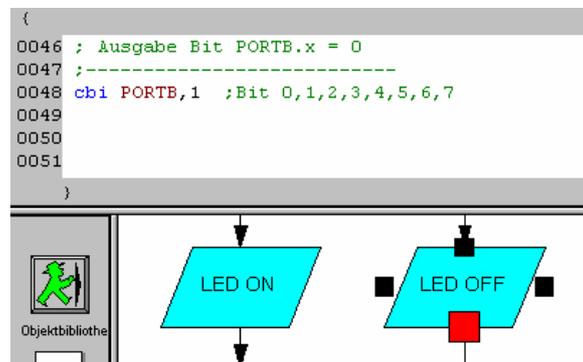
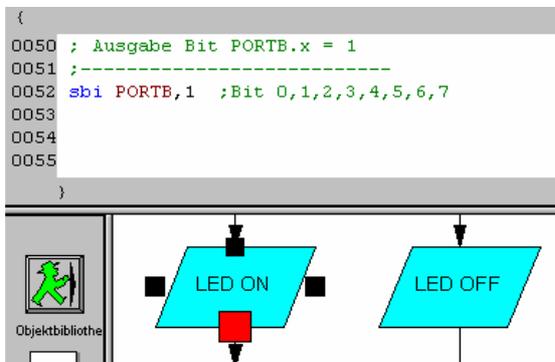


Abbildung 24

Ergänzen Sie die gezeigten Elemente!



5.7 Übersetzen, brennen und Test

Sind alle betreffenden Elemente mit Quellcode hinterlegt kann aus dem Programmablaufplan der komplette Quellcode generiert, kompiliert, gelinkt und auf den Mikrocontroller übertragen werden. Die gewünschte Funktion kann aus dem Aktionsmenü ausgewählt werden (Abbildung 25).

Hinweis:

Beachten Sie, dass für das Brennen des Controllers das Programmierkabel angeschlossen sein muss und bei Bedarf eine geeignete Spannungsquelle anzuschließen ist.



Abbildung 25

Funktionalitäten des Aktionsmenüs:

- | | |
|--------------------------|--|
| - Alles Ausführen | Quellcode generieren, kompilieren, linken, brennen |
| - Nur Quelltet erstellen | Quellcode generieren mit allen Marken |
| - Kompilieren | Quellcode zu Objektdatei übersetzen |
| - Linken | Objektdatei zu HEX-Datei binden |
| - Brennen | HEX-Datei an den Controller übertragen |
| - Testen | myAVR Controlcenter öffnen |
| - Quelltextdatei öffnen | Quellcodedatei öffnen |
| - Quelltext bereinigen | Quellcode von überflüssigen Marken bereinigen |

Im Ausgabefenster (siehe Abbildung 26) werden die jeweiligen Aktionen angezeigt.

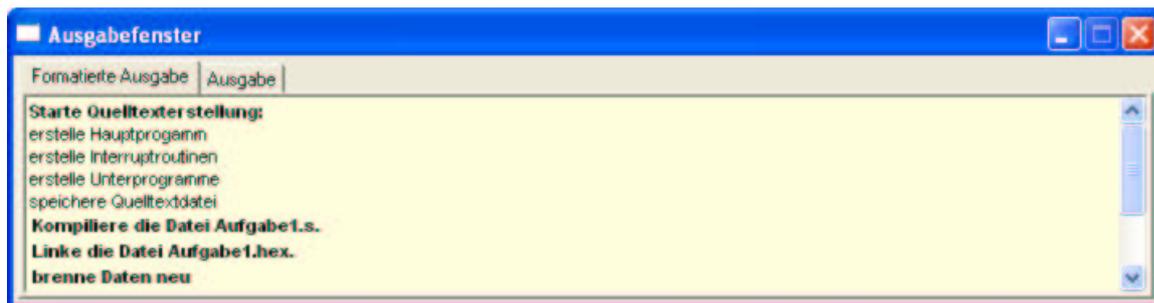


Abbildung 26: Ausgabefenster

Bei Compilerfehlern werden diese ebenfalls im Ausgabefenster mit der entsprechenden Zeilennummer angezeigt. Um zu dem Fehler zu gelangen, genügt meist ein Klick auf die Fehlermeldung. Das betreffende Objekt wird selektiert und die Zeile hervorgehoben.

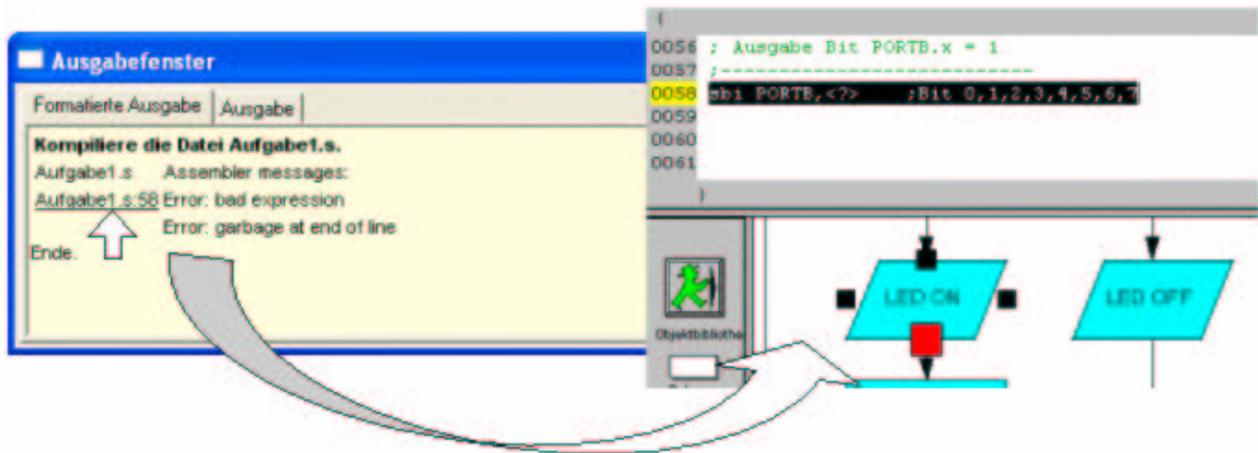
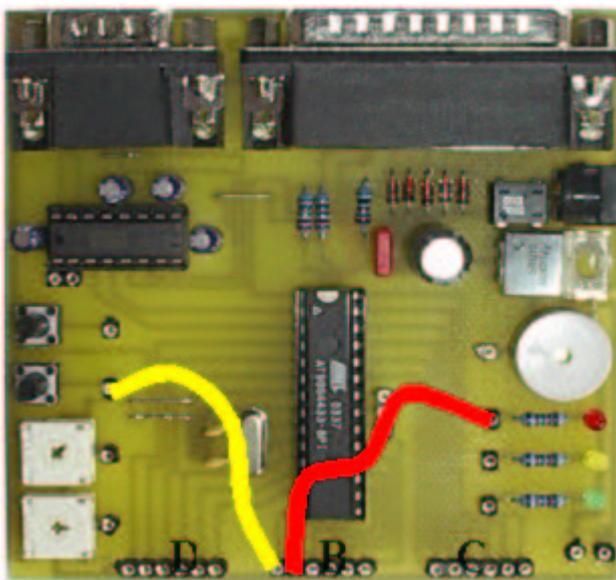


Abbildung 27: Fehlerbehandlung



Nachdem das Programm erfolgreich übersetzt und auf den Controller übertragen wurde, kann die Anwendung getestet werden. Dazu ist das myAVR-Controlcenter zu öffnen und die vorgegebenen Verbindungen auf dem Board zu stecken. Über die Schaltfläche „Start/Stopp“ kann das myAVR-Board ein und ausgeschaltet werden.

Abbildung 28



Abbildung 29

6 Dokumentengenerierung „QuickDok“

6.1 Zweck

Die Funktionalität „QuickDok“ beinhaltet die Generierung eines kurzen Word-Dokumentes. Dieses Dokument wird aus einer QuickDok-Vorlage erzeugt und beinhaltet die Beschreibung und Darstellung des aktiven Diagramms. Mit Hilfe der Funktionalität „QuickDok“ kann der Quelltext eines Programmes ausgedruckt werden.

6.2 Einstellungen in Word

Für die Dokumentengenerierung muß auf dem PC MS Word installiert sein. Unterstützt werden die Versionen Word 2000/XP/2003.

Die Dokumentengenerierung basiert auf Makros; deshalb ist in Word der Makrovirenschutz auf *mittel* zu stellen.

Vorgehensweise in Word:

- Wählen Sie die Menüfolge *Extras/Makro/Sicherheit*
- Aktivieren Sie auf der Registerkarte „Sicherheitsstufe“ *mittel*
- Auf der Registerkarte „Vertrauenswürdige Quellen“ setzen Sie die Häkchen bei „Allen Vorlagen und Add-Ins vertrauen“ und wenn vorhanden bei „Zugriff auf Visual Basic Projekte vertrauen“

6.3 Die Benutzung von QuickDok

Zum Start der Generierung betätigen Sie den Schalter  in der Menüleiste. Falls zu einem Dokument mehrere Vorlagen existieren, öffnet eine Auswahlliste.

Im folgendem Sichtfenster ist unter der Registerkarte „Definition“ ein änderbarer Vorschlag für die Dokument-Eigenschaften in Bezug auf Beschreibung, Titel, Kurztitel und Dateiname. Auf der Registerkarte „weitere Einstellungen“ kann eine eigene Vorlage für das Layout hinterlegt werden. In Abhängigkeit der gewählten Vorlage können weitere Einstellungen möglich sein.

Nach der Bestätigung mit OK wird Word gestartet. In der nachfolgenden Warnung betätigen Sie den Schalter „Makros aktivieren“; das Dokument wird generiert.

Anhang: Tastaturbelegung

Die Tastenbelegung ist abhängig vom jeweiligen Diagramm und der verwendeten Ausgabe:

F1	Hilfe wird geöffnet
F2	Zoomen
F3	Objekt wird dem Layer zugefügt
F4	Objekt wird vom Layer entfernt
F5	Farbe bei Rahmen wird geändert; Form am Anfang einer Verbindung ändert sich
F6	Bei Rahmen und Verbindungen ändert sich die Form
F7	Bei Rahmen und Verbindungen ändert sich der Linientyp
F8	Form am Ende einer Verbindung ändert sich
F9	Bewirkt, dass der Mittelpunkt einer Kante auf Null gesetzt wird
ESC	Im Diagramm: Nach oben Im Text-Editormodus: Beendet ihn mit Rückspeichern Im Dialog: Bricht ihn ab ohne zurückzuspeichern
Tab	In der Reihenfolge, in der die Objekte erstellt wurden, werden sie markiert
Umschalttaste + Enter	Objektbeschreibung wird geöffnet
Leertaste	Objektbeschreibung wird geöffnet
Alt + Enter	Dialog Definieren
Strg + Enter	Diagramm nach unten
Strg + `R	Report für selektiertes Objekt
Strg + `A`	Executebefehl ausführen (nur in bestimmten Ausgaben)
Strg + `D`	Diagrammreport wird aufgerufen
Strg + `I`	Import von Diagrammen
Strg + `T`	Tokensteuerung starten/beenden
Strg + `X`	Export von Diagrammen
Strg + `+`	Selektiertes Objekt wird vergrößert (nur in bestimmten Ausgaben)

Strg + ` -`	Selektiertes Objekt wird verkleinert (nur in bestimmten Ausgaben)
Strg + ` *`	Ursprüngliche Objektgröße wird wiederhergestellt
Strg + Maustaste	Selektiertes Objekt wird innerhalb des Diagramms kopiert
Strg + Kursortasten:	
- Cursor nach links:	Selektiertes Objekt wird in X-Richtung verkleinert
- Cursor nach rechts:	Selektiertes Objekt wird in X-Richtung vergrößert
- Cursor nach oben:	Selektiertes Objekt wird in Y-Richtung vergrößert
- Cursor nach unten:	Selektiertes Objekt wird in Y-Richtung verkleinert
Enter	Editormodus zum Definieren der Objekte
Entf	Löschen
Kursortasten	Selektiertes Objekt wird verschoben (in Verbindung mit der Umschalttaste sind größere Schritte möglich)
` +`	Diagramm vergrößern
` -`	Diagramm verkleinern
` *`	Einpassen des Diagramms

Anhang: Mausoperationen

Die Maus hat in SiSy eine Anzahl von nützlichen Funktionen, welche die Arbeit in Projekten erleichtern.

Selektion	<i>Klick auf Objekt</i> Objekt ist markiert und kann separat weiterbearbeitet werden.
Selektion aufheben	<i>Klick auf Fensterhintergrund</i> Aufhebung der Objektmarkierung.
Mehrfachselektion	<i>Umschalttaste + Klick auf Objekt</i> Selektion/Markierung von mehreren Objekten zur Weiterbearbeitung. <i>Markise</i> Mit gedrückter linker Maustaste auf Fensterhintergrund und Ziehen eines Rechtecks über zu markierende Objekte.
Verschieben	<i>Drag & Drop im Diagramm</i> Objekt mit linker Maustaste anfassen und verschieben. Objekte werden am Raster verschoben. <i>Umschalttaste + Drag & Drop im Diagramm</i> Verschieben von Objekten ohne Raster.
Fensterinhalt schieben	<i>Linke und rechte Maustaste drücken + Verschieben der Maus im Diagramm</i> Der komplette Diagramminhalt wird geschoben.
Objekt kopieren	<i>STRG + Drag & Drop</i> Maustaste gedrückt halten und Mauszeiger vom Objekt auf den Fensterhintergrund führen. Eine Originalkopie des Objektes wird im aktuellen oder in einem anderen Diagramm erzeugt.
Referenz erzeugen	<i>Drag & Drop aus Navigator</i> Ziehen des gewünschten Objektes aus dem Navigator in das Diagramm. Es wird eine Referenz des gewählten Objektes erzeugt. <i>Drag & Drop aus Objektbibliothek</i> Rot beschriftete Objekte können nur als Referenz erzeugt werden. Eine Liste zur Auswahl des gewünschten Typs erscheint. <i>Strg + Drag & Drop aus Objektbibliothek</i> Eine Liste zur Auswahl der gewünschten Referenz des Originalobjektes erscheint.

	<p><i>Drag & Drop aus anderem Diagramm</i> Ziehen des gewünschten Objektes aus dem Quelldiagramm in das Zieldiagramm. Es wird eine Referenz des gewählten Objektes erzeugt.</p>
Objekt anlegen	<p><i>Drag & Drop aus Objektbibliothek</i> Ein Objekt aus der Objektbibliothek wird im Diagramm angelegt und steht zur Verfeinerung bereit.</p>
Objekt anhängen	<p><i>Drag & Drop Verteiler auf Fensterhintergrund</i> Durch Ziehen einer Kante vom Verteiler auf den Fensterhintergrund wird ein neues Objekt erzeugt. Nach Auswahl des Objekttyps sind die Objekte miteinander verbunden.</p>
Objekte verbinden	<p><i>Drag & Drop Verteiler zu Objekt</i> Klick auf den Verteiler des zu verbindenden Objektes. Bei gedrückter linker Maustaste auf das gewählte Objekt ziehen.</p> <p><i>Verbindung aus Objektbibliothek (in der UML)</i> Hierbei wird erst die gewünschte Verbindung in der Objektbibliothek angeklickt und danach die beiden zu verbindenden Objekte im Diagramm nacheinander.</p>
Verbindung anordnen	<p><i>Drag & Drop Mittelpunkt</i> Beliebige Gestaltung der Verbindung durch Ziehen mit der Maus.</p>
Verbindung ändern	<p><i>Drag & Drop Anfangs-/Endpunkt einer Kante</i> Für die Verbindung wird ein neues Zielobjekt gewählt.</p>
Objekt definieren	<p><i>Doppelklick auf Objekt</i> Durch Doppelklick auf Objekte öffnet sich das Kontextmenü. Bei Abschalten des Menüs unter <i>Einstellungen/Menü bei Doppelklick</i> erscheint eine Zeile zur Namensgebung. Mit ESC wird die Eingabe bestätigt.</p> <p><i>Doppelklick auf Verteiler</i> Es wird der Definieren-Dialog aufgerufen, in dem das Objekt benannt und beschrieben werden kann.</p>
Kontextmenü öffnen	<p><i>Klick mit rechter Maustaste auf Objekt</i></p>
Fenster neu zeichnen	<p><i>Doppelklick auf Fensterhintergrund</i></p> <p><i>Hinweis:</i> Doppelklick mit linker Maustaste wirkt wie Enter.</p>
Fenster aktualisieren	<p><i>Strg + Doppelklick auf Fensterhintergrund</i> Die vom Programm ausgeführten aber noch nicht sichtbargemachten Befehle werden im Fenster erstellt. Das Fenster wird aktualisiert.</p>