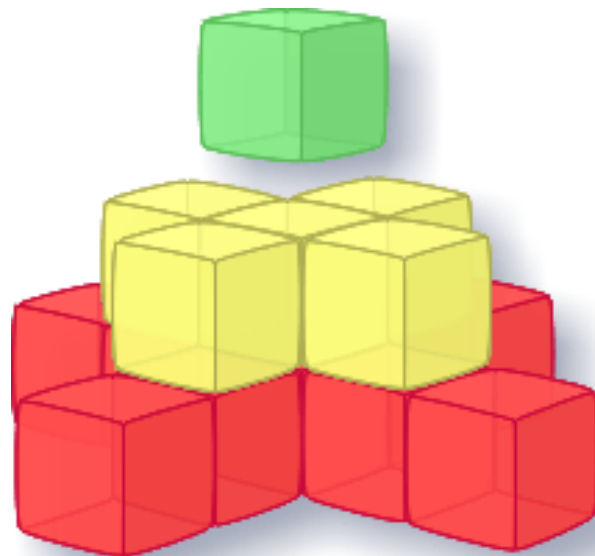


Dipl. Ing. Toralf Riedel  
Dipl. Ing. Päd. Alexander Huwaldt

# Benutzerhandbuch SiSy<sup>®</sup>

gültig ab SiSy-Version 3.6



---

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.  
Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.  
Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.  
Trotzdem können Fehler nicht vollständig ausgeschlossen werden.  
Die Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.  
Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.  
Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Dokument erwähnt werden, sind gleichzeitig auch eingetragene  
Warenzeichen und sollten als solche betrachtet werden.

Auflage 10-2: November 2016

© Laser & Co. Solutions GmbH  
[www.laser-co.de](http://www.laser-co.de)  
[www.myMCU.de](http://www.myMCU.de)  
[service@mymcu.de](mailto:service@mymcu.de)  
Tel: ++49 (0) 3585 470 222  
Fax: ++49 (0) 3585 470 233

# Inhalt

1	Einleitung .....	5
2	Vorbereitung .....	6
2.1	Installation der Software .....	6
2.2	Hardware für die Mikrocontrollerprogrammierung .....	8
3	Arbeiten mit SiSy, allgemein .....	9
3.1	Projektarbeit .....	9
3.2	Die Modellierungselemente von SiSy .....	11
3.3	Die Fenster für die Modellierung .....	13
3.4	Modelle bearbeiten .....	16
3.4.1	Modellebenen .....	16
3.4.2	Bewegen in Modellen .....	17
3.4.3	Modellelemente anlegen .....	18
3.4.4	Modellelemente auswählen .....	18
3.4.5	Modellelemente referenzieren .....	18
3.4.6	Modellelemente verschieben .....	19
3.4.7	Modellelemente löschen .....	19
3.4.8	Modellelemente verbinden .....	19
3.4.9	Die Größe von Modellelementen verändern .....	19
3.4.10	Modellelemente untersuchen (Report) .....	20
3.4.11	Modellelemente kopieren .....	20
3.4.12	Modellelemente ausschneiden .....	20
3.4.13	Diagramme kopieren (flache Kopie) .....	20
3.4.14	Modelle kopieren (tiefe Kopie) .....	21
3.4.15	Die Modellhierarchie bearbeiten (Jo-Jo) .....	21
3.5	Die Ansicht verändern .....	21
3.5.1	Fensterinhalt verschieben .....	21
3.5.2	Fensterinhalt vergrößern oder verkleinern .....	21
3.5.3	Fensterinhalt farbig oder als Kontur darstellen .....	22
3.5.4	Die Schriftart ändern .....	22
3.6	Druckfunktionen in SiSy .....	22
3.6.1	Diagramme drucken .....	23
3.6.2	Grafiken und Inhalte drucken .....	24
3.6.3	Nur Quellcodes drucken .....	25
3.6.4	Nutzen der Zwischenablage .....	25
3.6.5	Dokumentengenerierung .....	26
3.7	SiSy Add-Ons .....	27
3.8	SiSy LibStore .....	30
3.8.1	ein Projekt mit SiSy LibStore anlegen .....	30
3.8.2	SiSy LibStore in einem Projekt nutzen .....	30
3.9	Die Hilfsfunktionen in SiSy .....	31
3.9.1	SiSy Online-Tutorials .....	31
3.9.2	Die allgemeine Hilfe .....	31
3.9.3	Der Assistent .....	32
3.9.4	spezielle Hilfen zur Programmierung .....	33
4	Ausgewählte Methoden/Profile zur Programmierung .....	36
4.1	Programmierung im Klassendiagramm .....	36
4.1.1	PC-Programm .....	37
4.1.2	Mikrocontrollerprogrammierung in C++ .....	40
4.2	Programmentwicklung aus einem Struktogramm .....	44
4.2.1	Grundlagen .....	44
4.2.2	Konsolenprogramme .....	45
4.2.3	AVR Mikrocontroller-Programme .....	46
4.3	Entwicklung Programmablaufplan für AVR Programme .....	48

---

4.3.1	Einfache Programmentwicklung aus einem PAP.....	48
4.3.2	Unterprogrammtechnik im PAP.....	51
4.3.3	Interrupt-Service-Routinen (ISR) im PAP .....	53
4.3.4	Daten im PAP.....	54
4.4	Quellcode-Programmierung ohne Modellierung.....	55
4.4.1	Kleines Programm.....	55
4.4.2	Programm mit mehreren Dateien.....	60
5	Weitere Methoden .....	64
5.1	Objektorientierte Techniken mit der UML.....	64
5.2	Prozessmodellierung.....	67
5.3	Datenmodellierung (ERD/Klassendiagramm) .....	70
6	Zusätzliche Werkzeuge .....	71
6.1	Einführung .....	71
6.2	Das SiSy ControlCenter .....	71
6.2.1	Kommunikation mit dem Controller.....	72
6.2.2	Daten empfangen vom Controller.....	74
6.2.3	Empfangene Daten speichern.....	76
6.2.4	Daten an den Controller senden .....	77
6.3	Der myAVR Code-Wizard.....	79
6.4	Das STM32 ST-Link Utility .....	83
6.5	Das myAVR ProgTool .....	89
6.5.1	Übersicht zum myAVR ProgTool .....	89
6.5.2	Einstellungen Fuse- und Lock-Bits für AVR Produkte .....	94
6.6	Der Debugger .....	98
6.6.1	Debuggen von SVL- und ARM-Programmen.....	98
6.6.2	Der SVL Debug-Monitor.....	100
6.6.3	Das SVL-Werkzeug RegExp.....	101
6.7	Weitere Werkzeuge .....	102
6.7.1	mySmartUSB Terminal .....	102
6.7.2	mySmartUSB light - SupportBox.....	102
7	Informationen zu SiSy-Ausgaben .....	103
Anhang: Tastaturbelegung, allgemein.....		104
Anhang: Mausoperationen .....		106
Index .....		108

# 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 eine bestimmte Methode, z.B. Prozessmodellierung oder die Möglichkeiten der Programmierung eines Mikroprozessors, wird an ausgewählten Beispielen beschrieben. Auf die Grundlagen der Mikroprozessorprogrammierung wird im Rahmen dieses Handbuchs nicht eingegangen. Dazu dienen die verschiedenen myMCU Lehrbücher und Tutorials. Wichtige Funktionen des Programms sind in der Hilfe-Funktion von SiSy abrufbar und ergänzen die Ausführungen dieses Buches.

Dem Nutzer wird in diesem Handbuch 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 Explorer zur gezielten Suche enthaltener Objekte oder Diagramme im Projekt;
- der SiSy LibStore mit Vorlagen, Beispielen, Mustern und Bibliotheken;
- 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, Schaltflächen und Menüpunkten werden in Anführungszeichen gesetzt.

Beispiel: Dialogbox „Definition“, Schaltfläche „OK“

## **Zu den beschriebenen SiSy Ausgaben**

Für die unterschiedlichen Einsatzmöglichkeiten von SiSy gibt es spezifische Ausgaben, die auf die jeweiligen Aufgabenstellungen zugeschnitten sind. Je nach Einsatzgebiet der Entwicklungsumgebung gibt es zu jeder Ausgabe mehrere Lizenzmodelle. Eine Auflistung der Ausgaben und Lizenzmodelle sind unter [www.sisy.de](http://www.sisy.de) zusammen gestellt.

Die Beschreibungen in diesem Handbuch sind weitgehend allgemeingültig und beziehen sich auf keine explizite SiSy Ausgabe. Auf beschriebene Funktionalitäten, die nicht in allen SiSy Ausgaben verfügbar sind, wird separat hingewiesen.

## 2 Vorbereitung

### 2.1 Installation der Software

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

Sollten Sie SiSy bereits installiert haben, können Sie dieses Kapitel überspringen.

Die Installation und der erste Start  
müssen mit Administratorrechten ausgeführt werden

#### **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.sisy.de](http://www.sisy.de) → Meine SiSy-Lizenz      oder  
[www.mymcu.de](http://www.mymcu.de) → Online-Shop → Kontakt/Service  
oder fordern Sie diese beim Hersteller an:  
Tel: 03585-470222  
Fax: 03585-470233  
e-Mail: [support@mymcu.de](mailto:support@mymcu.de)

Bitte prüfen Sie die Systemvoraussetzungen für die Installation und die Arbeit mit SiSy:

- PC-Arbeitsplatz bzw. Notebook (mit USB-Anschluss für Mikrocontrollerprogrammierung)
- PC mit Windows XP ... 10
- Maus oder ähnliches Zeigegerät
- Zusätzlich für die Mikrocontroller-Programmierung
  - Assembler bzw. C/C++ Entwicklungsumgebung (in SiSy bereits integriert)
  - Hardware entsprechend Einsatzgebiet
  - Programmierkabel: USB Kabel bzw. Mini USB-Kabel (abhängig von Hardware)
  - Bei Bedarf externe Spannungsversorgung z.B. stabilisiertes 9 V Netzteil

Sie sollten Grundkenntnisse in einer beliebigen Programmiersprache besitzen.

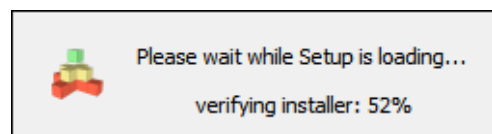
#### **Installation**

Legen Sie den Datenträger CD „SiSy“ in Ihren PC ein. Falls der Start nicht automatisch erfolgt, wählen Sie bitte im Windows-Explorer das Laufwerk mit dem Datenträger und starten die „setup.exe“ aus dem Pfad *Datenträger-Laufwerk:\Ausgabe\SiSy*.

Auf dem Startbildschirm finden Sie Schaltflächen zur Installation der Software und zum Öffnen von Begleitdokumenten.

#### **Auswahl für die Installation**

Für die Installation der Software betätigen Sie die entsprechende Schaltfläche. In Abhängigkeit Ihrer Rechnerkonfiguration kann der Start des Setup-Programms einige Sekunden dauern. Das gestartete Setup-Programm wird Sie durch die weitere Installation führen.



Im ersten Fenster nach dem Start der Installation werden Sie darauf hingewiesen, alle anderen Programme zu schließen. Folgen Sie diesem Hinweis.

Die Lizenzbestimmungen werden Ihnen angezeigt. Bitte lesen Sie diese sorgfältig durch. Wenn Sie sich damit einverstanden erklären, bestätigen Sie diese.

Im folgenden Dialog werden Sie aufgefordert, Ihre Lizenzangaben einzugeben. Beachten Sie dabei bitte die korrekte Schreibweise, so wie Sie die Angaben erhalten haben.

Im weiteren Verlauf der Installation werden in Dialogfenstern jeweils die Standard-Einstellungen angezeigt, die Sie individuell ändern können. Das betrifft z.B. die Auswahl der zu installierenden Komponenten, den Installationspfad sowie den Startmenü-Ordner.

### *Beginn der Installation*

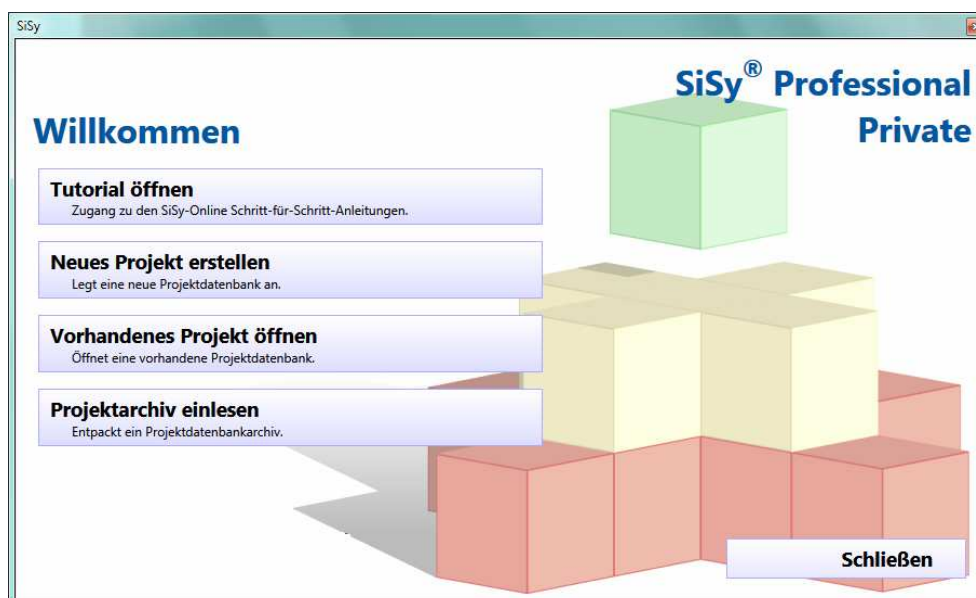
Beginnen Sie die Installation mit Betätigen der Schaltfläche „Installieren“. Während der Installation wird Ihnen der Fortschritt angezeigt; ggf. erhalten Sie Hinweise, dass verschiedene Treiber installiert bzw. aktualisiert werden sollten. Zu diesen Hinweisen wird stets eine Eingabe erwartet. In Abhängigkeit Ihrer Rechnerkonfiguration kann die komplette Installation einige Minuten in Anspruch nehmen.



Zum Abschluß der Installation wird ein entsprechendes Dialogfeld angezeigt. Mit dem Aktivieren der Schaltfläche „Fertig stellen“ ist die Installation abgeschlossen.

### *Start der installierten Ausgabe*

Für den Start von SiSy ist auf dem Desktop eine Verknüpfung angelegt; haben Sie keine Verknüpfung angelegt, starten Sie aus dem SiSy-Installationsverzeichnis die Datei „sisy.exe“. Auf Ihrem Bildschirm erscheint für die weitere Auswahl der Dialog „Willkommen“ (Willkommenbildschirm). Mit den zur Auswahl stehenden Schaltflächen können Sie Ihre Arbeit fortsetzen.



**Abbildung 1: Willkommensbildschirm in SiSy**

### Hinweis:

SiSy beinhaltet 2 Dateien mit Makros („handbuch.doc“, „multi.doc“). Von einigen Virenschannern werden diese Makros als „Virus“ erkannt und entsprechend behandelt. In den Heuristik-Einstellungen des Virenschanners kann diese Behandlung unterdrückt werden.

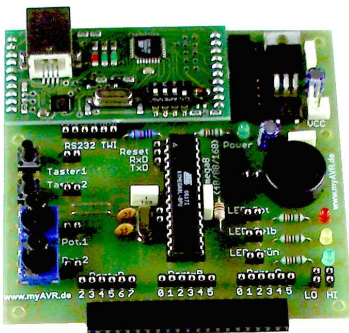
## 2.2 Hardware für die Mikrocontrollerprogrammierung

Dieses Kapitel bezieht sich nur auf Programmierung von Mikrocontrollern.

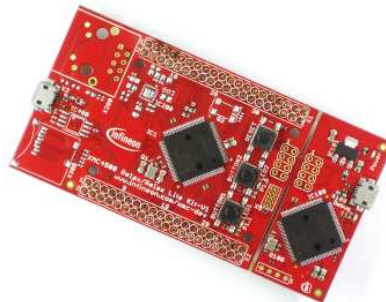
Alle Ausführungen und Beispiele für die Mikrocontrollerprogrammierung in diesem Handbuch beziehen sich auf folgende Referenzhardware, die bei [www.myMCU.de](http://www.myMCU.de) erhältlich ist:

- myAVR Board MK2 für die AVR-Programmierung
- STM32F4-Discovery für die Programmierung mit einem STM32-Controller
- XMC4500 Relax (Lite) Kit für die Programmierung mit einem XMC-Controller

Dokumente sowie Anwendungsbeispiele zu myAVR Boards stehen zum Download bereit unter [www.myAVR.de](http://www.myAVR.de). Für das XMC4500 Relax (Lite) Kit sind frei verfügbare, fertig lauffähige Beispiele erhältlich bei [www.infineon.com](http://www.infineon.com); für das STM32F4-Discovery auf [www.st.com](http://www.st.com).



**myAVR Board MK2**



**XMC4500 Relax Lite Kit**



**STM32F4-Discovery**

Für die Beispiele zur Programmierung in diesem Handbuch benötigen Sie die nachfolgend aufgeführte Hard- bzw. Software; sie ist exemplarisch zu verstehen.

	Hardware	Software
PC-Programme		SiSy Microcontroller ++ oder SiSy Professional
AVR-Programme	bestücktes myAVR Board MK2 Programmierkabel (USB) Patchkabel	SiSy Microcontroller ++ oder SiSy AVR oder eine andere SiSy Ausgabe mit dem Add-On AVR
STM32-Programme	STM32F4-Discovery Programmierkabel (Mini-USB) STM-Patchkabel STM32-Board-F4D (optional)	SiSy Microcontroller ++ oder SiSy STM32 oder eine andere SiSy Ausgabe mit dem Add-On ARM
XMC-Programme	XMC4500 Relax (Lite) Kit Programmierkabel (Micro-USB) STM-Patchkabel myXMC Board 4500 (optional)	SiSy Microcontroller ++ oder SiSy XMC oder eine andere SiSy Ausgabe mit dem Add-On ARM

Ausführliche Beschreibungen zur Programmierung sind nicht Inhalt dieses Handbuchs. Weiterführende Erklärungen finden Sie in myMCU Lehrbüchern und Tutorials.



### 3 Arbeiten mit SiSy, allgemein

In diesem Kapitel werden die in SiSy verwendeten Begriffe zweckentsprechend definiert und die Handhabung erläutert. Dem Nutzer wird damit aufgezeigt, wie er in SiSy modellieren kann.

#### 3.1 Projektarbeit

##### **Was ist ein SiSy-Projekt?**

Ein SiSy-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.

##### **Neues Projekt erstellen**

In SiSy legen Sie stets ein Projekt an. In dieses Projekt integrieren Sie Ihr Programm bzw. mehrere Programme. Jedes Projekt benötigt einen Namen.

Beim Aktivieren der Schaltfläche „Neues Projekt erstellen“ im Willkommensbildschirm (vgl. Abbildung 1) wird ein neues Projekt 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 „Ordner für Projekte ändern“ kann die vorgeschlagene Pfadangabe geändert werden.

##### Hinweis:

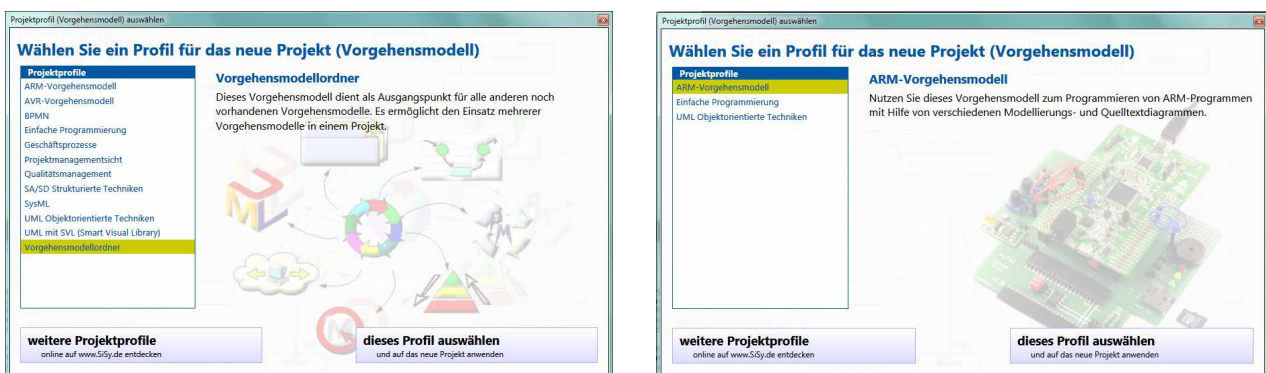
Wird ein bereits vorhandener Projekt- oder Verzeichnisname gewählt, erscheint eine entsprechende Warnung.

##### **Profil auswählen**

Nach der Vergabe eines Projektnamens wird im nachfolgenden Fenster ein Profil (Vorgehensmodell) ausgewählt. Die Auswahl des Profils bestimmt im weiteren Projektverlauf die von SiSy zur Verfügung gestellten Werkzeuge. Je nach Profil können Vorlagen oder Assistenten das Erstellen eines neuen Projektes unterstützen.

In Abhängigkeit der SiSy Ausgabe stehen unterschiedliche Profile (Hauptebenen) zur Verfügung. Zu jedem Profil werden bei der Markierung eine kurze Beschreibung sowie Abbildungen angezeigt.

Mit Auswahl des Profils entscheiden Sie über die Art des Projektes.



**Abbildung 2: Profile in Abhängigkeit der Ausgabe: SiSy Professional (links) und SiSy STM32 (rechts)**

##### Hinweis:

In älteren Ausgaben und anderen Dokumenten wird das Profil noch als Vorgehensmodell (VGM) benannt.

**Vorhandenes Projekt öffnen**

Beim Aktivieren der Schaltfläche „Vorhandenes Projekt öffnen“ (vgl. Abbildung 1) werden bereits angelegte Projekte mit dem Datum der letzten Änderung für die Auswahl gelistet. Ist der Mauszeiger auf dem Projektnamen, werden Informationen angezeigt. Nach Auswahl des gewünschten Projektes wird dieses geöffnet und die abgespeicherten Diagramme und Objekte sowie alle Möglichkeiten für die weitere Bearbeitung stehen zur Verfügung.

**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. Beim Aktivieren der Checkbox „Datum anfügen“ wird das Projekt mit Änderungsdatum gespeichert. 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.*

**Projektarchiv einlesen**

Menü *Projekt/Archiv/Einlesen* oder Schaltfläche „Projektarchiv einlesen“ (Abbildung 1)  
Hierunter versteht man das Einlesen eines Archivs zum Zweck der Rekonstruktion des Projektes. Einlesen bedeutet Entpacken eines archivierten Projektes. Dazu muss das Archiv ausgewählt sein. Beim Aktivieren der Checkbox „Datum entfernen“ wird das Datum nicht im Projektnamen geschrieben.

Hinweis:

*Existiert im Entpacken-Zielpfad bereits ein Projekt, folgt eine Überschreibwarnung.*

**Projektarchiv als Email versenden**

Menü *Projekt/Archiv/Als Email versenden...*

Für den Versand eines Projektes per Email sollten diese zu einem Archiv zusammengefasst werden. Die Funktion „Projektarchiv als Email versenden“ erzeugt von dem aktuell geöffneten Projekt ein Archiv, startet das Standard-Email-Programm Ihres Systems, legt eine neue Email an und fügt das Archiv als Anlage ein. Sie können jetzt die Email schreiben und sofort das Projektarchiv senden.

Hinweis:

*Das Standard Email-Programm Ihres Systems muss ordnungsgemäß eingerichtet und registriert sein, damit diese Funktion korrekt ausgeführt werden kann. Es ist nicht möglich, diese Funktion mit einem WEB-Browser Email-Programm zu nutzen.*

## 3.2 Die Modellierungselemente von SiSy

### Werkzeug

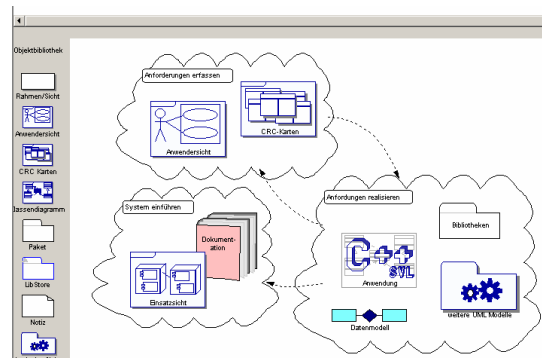
SiSy stellt für die Bearbeitung der Modelle und Teilmodelle Werkzeuge der entsprechenden Methodik bereit. Werkzeuge sind Editoren, mit denen in einem Fenster die grafische Darstellung (Diagramm) der Modelle bearbeitet werden kann.

### Modellelemente

Modelle bestehen aus Elementen, welche zueinander in Beziehung stehen. Diese Modellelemente werden nach bestimmten Regeln (Metamodell) durch SiSy verarbeitet und dargestellt (Notation). Modellelemente sind in der Regel Knoten (Objekte), Kanten (Verbindungen), Rahmen, Referenzen auf Modellelemente und Texte. Prinzipiell kann jedes Modellelement verfeinert werden. Typisch ist es Knoten (Objekte) dadurch zu verfeinern, dass „unter“ diesen Objekten weitere Modellelemente angelegt werden. Dabei entsteht eine Modellhierarchie. Jedes Modellelement verfügt über einen Kurznamen, einen Langnamen, eine Beschreibung und modellinterne Attribute (Inhalt).

### Diagramme

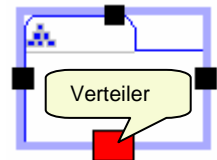
Diagramme sind grafische Darstellungen von Modellen oder Teilmodellen, die mit einem bestimmten Werkzeug erstellt werden. Die Modellierungselemente werden als Objekte in den Diagrammen unter Einhaltung von Regeln zusammengestellt.



### Objekte

Objekte sind mögliche Modellelemente in Diagrammen, z.B. „CRC-Karten“ im „UML-Projekt“. Objekttypen sind konkrete Ausprägungen von Objekten, die in einem Diagramm angelegt wurden, z.B. Objekttyp „Paket“ vom Objekt „CRC-Karte“.

Beim Markieren eines Objektes (Klick mit Maustaste) wird dieses mit einer farbigen Linie umrandet, der Verteiler (rotes Rechteck) ist sichtbar. Vom Verteiler erfolgt die Verbindung von Objekten untereinander.



Objekte besitzen Eigenschaften, die per Definieren-Dialog zugewiesen werden. Jedem Objekt ist ein Name zuzuweisen; weitere Eigenschaften sind objektabhängig; dementsprechend gibt es differenzierte Registerkarten im Definieren-Dialog. Der Definieren-Dialog wird erreicht über das Kontextmenü → Definieren (Objekt markieren → rechte Maustaste → Definieren).

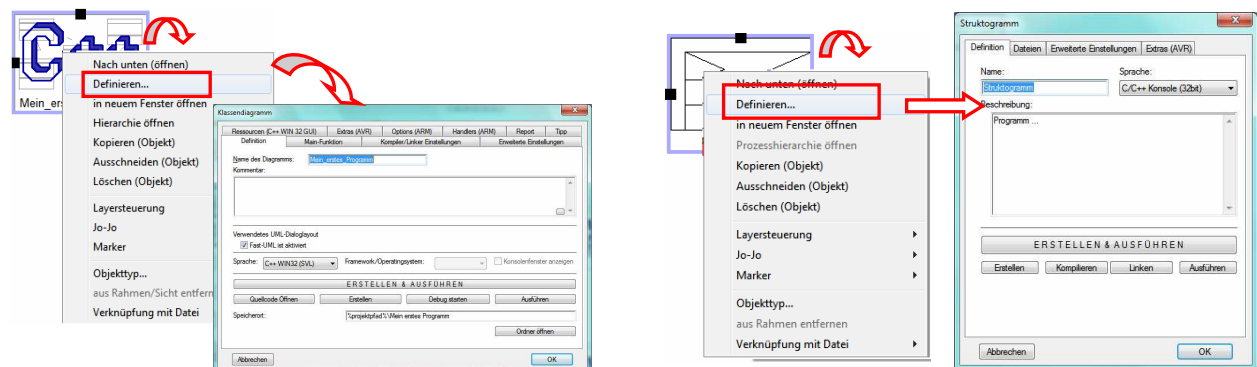


Abbildung 3: Aufruf Definieren-Dialog; links Klassendiagramm und rechts Struktogramm

### **Referenzen**

Die Objekte eines Diagramms können in anderen Diagrammen wieder verwendet werden. Durch das Hineinziehen in das aktuelle Diagramm aus dem Navigator, dem Projekt-Explorer 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 so genannten 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. Inhalte von Rahmen sind in Reports oder einer Dokumentengenerierung auswertbar.

### 3.3 Die Fenster für die Modellierung

Unabhängig von der SiSy Ausgabe und dem Profil sind die Fenster für die Modellierung einheitlich aufgebaut. Die typische Aufteilung der SiSy-Benutzeroberfläche besteht aus Navigator, Explorer, Assistent, Diagrammfenster und Editorfenster. In Abhängigkeit des gewählten Profils und der ausgeführten Aktionen kommen noch Ausgabefenster und das Fenster für das Sequenzdiagramm zur Anzeige. Die Aufteilung zwischen Diagrammfenster und Editorfenster, sowie zwischen den beiden linken Fenstern können Sie je nach Bedarf anpassen.

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.

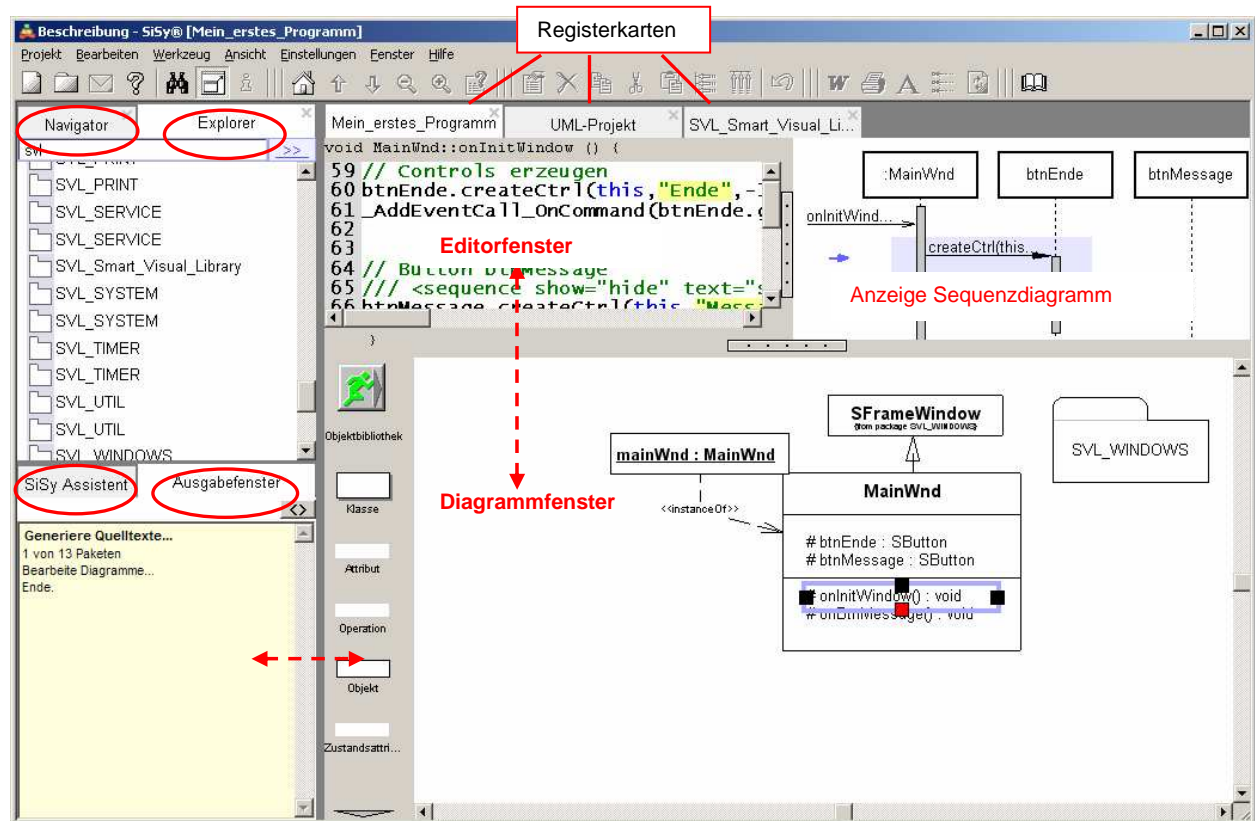


Abbildung 4: SiSy Benutzeroberfläche

#### Navigator

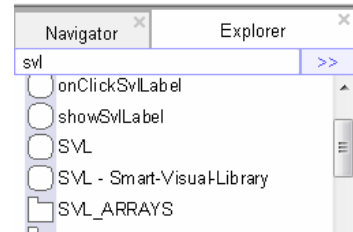
Dieser befindet sich im linken, oberen Teil des Fensters. Er ermöglicht dem Anwender die Orientierung über die in der Projektdatenbank gespeicherten Objekte sowie deren Bereitstellung für die weitere Verwendung. Nach dem Start von SiSy werden neben dem Vorgehensmodell eine Reihe leicht zu handhabender Schaltflächen, Menüs und weitere Hilfsmittel angezeigt.

Beim Start von SiSy ist die Kategorie „Schnellzugriff“ voreingestellt. Mit Klick der rechten Maustaste im Navigator öffnet ein Kontextmenü, aus dem eine Kategorie für die Anzeige gewählt werden kann.

Der „Schnellzugriff“ kann für die individuelle Arbeit genutzt werden. Wiederholt benötigte Diagramme, Objekte usw. können aus dem Diagrammfenster in den Navigator gezogen werden.

## Projekt-Explorer

Der Explorer befindet sich ebenfalls im linken, oberen Teil des Fensters, neben dem Navigator. Er fungiert als Suchfunktion im Projekt. Alle im Projekt enthaltenen Objekte werden gelistet, die im Namen den Suchbegriff oder einen Teil dessen enthalten. Die gefundenen Objekte stehen für die weitere Verwendung im Projekt bereit.



## 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 und VGM bezogen.
- Oft können Beispiele als Vorlagen vom Assistenten geladen werden.

## Ausgabefenster

Das Ausgabefenster wird nur in den Profilen zur Programmierung aktiviert. Während der Generierung des Programmcodes werden die protokollierten Arbeitsschritte sowie Fehlermeldungen angezeigt.

Compilerfehler werden im Ausgabefenster mit der entsprechenden Zeilennummer angezeigt. Mit Klick auf die Fehlermeldung gelangt man zum entsprechenden Objekt bzw. zur Zeile im Quelltext.

## Diagrammfenster

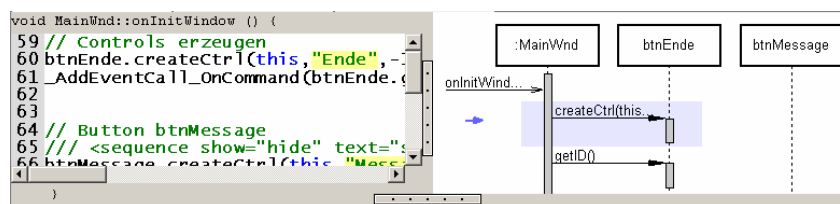
Das Diagrammfenster nimmt die rechte Bildschirmseite ein und ist der Raum, in dem der Nutzer modelliert. Es enthält:

- das ausgewählte Profil
- die Objektbibliothek mit den möglichen Objekten des aktuellen Diagramms
- als weitere Unterteilung das Editorfenster

## Editorfenster

Das Editorfenster dient zur Beschreibung des markierten Objektes; in den Profilen zur Programmierung zum Editieren des Quelltextes.

Bei der Programmierung in C++ wird das Editorfenster noch einmal unterteilt, auf der rechten Seite wird das Sequenzdiagramm zum aktuellen Quelltext gezeigt.



## Bedienelemente/Werkzeuggeste/Objektbibliothek/Aktionsmenü

### Bedienelemente

SiSy bietet, wie bei Windows-Anwendungen üblich, die Steuerung von Befehlen über das Hauptmenü, über die Werkzeuggeste, 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 Schaltfläche in der Werkzeuggeste, die eine schnelle und effiziente Bedienung des Programms ermöglicht. Die Toolbox-Darstellung bietet dem Anwender wichtige Programmfunktionen als direkten Link an.

Werkzeuggeste

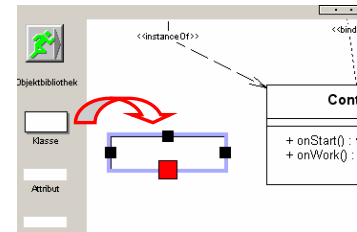


Abbildung 5: Hauptmenü und Werkzeuggeste im geöffneten SiSy-Projekt

Bedienelemente der Werkzeuggeste					
Projektaktionen		Projekt neu	Objektaktionen		Definieren-Dialog
		Projekt öffnen			Löschen
		Projekt als eMail versenden			Kopieren
		Assistent öffnen/schließen			Ausschneiden
		Navigator öffnen			Einfügen
		Projekt-Explorer öffnen erweiterte Suche			Ausrichtung aller markierten Objekte
		Fenstergröße			Rückgängig
		Info-Symbole am Objekt			
Diagramm-aktionen		Home (Einstiegsprofil)	Dokumenten- bearbeitung		QuickDok starten
		Eine Ebene nach oben			Druckmenü öffnen
		Auf Objekt nach unten			Format für Texte
		Ansicht verkleinern			Alles am Raster ausrichten
		Ansicht vergrößern			Aktualisieren
		Dialogfeld Report öffnen			Tutorials öffnen

Objektbibliothek

Die Objektbibliothek unterstützt das Anlegen neuer Objekte. Das gewünschte Objekt wird per Drag&Drop aus der Objektbibliothek in das Diagrammfenster gezogen. Die in der Objektbibliothek enthaltenen Objekte sind vom Profil und der Ebene im Profil abhängig.



Aktionsmenü



Das Aktionsmenü kann in unteren Ebenen in der Objektbibliothek integriert sein. Unterschiedliche Funktionen können daraus sofort aufgerufen werden, ohne die aktuelle Ebene zu verlassen. Der Inhalt des Aktionsmenüs ist abhängig vom Profil und der aktiven Ebene.

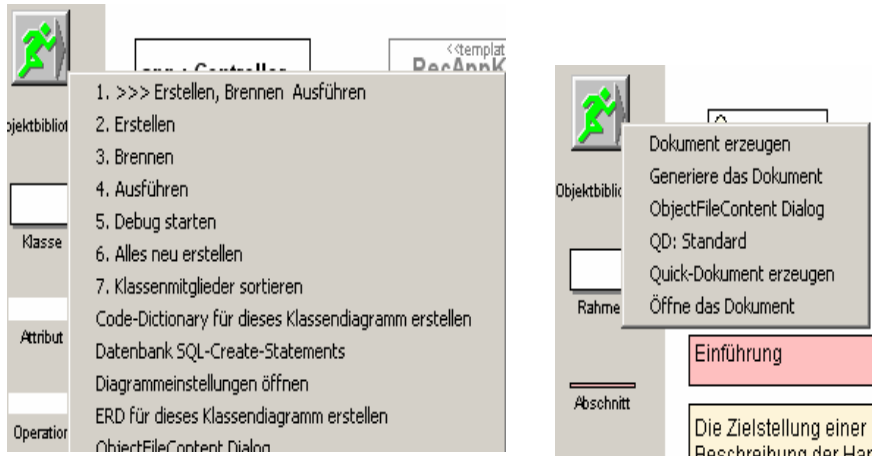
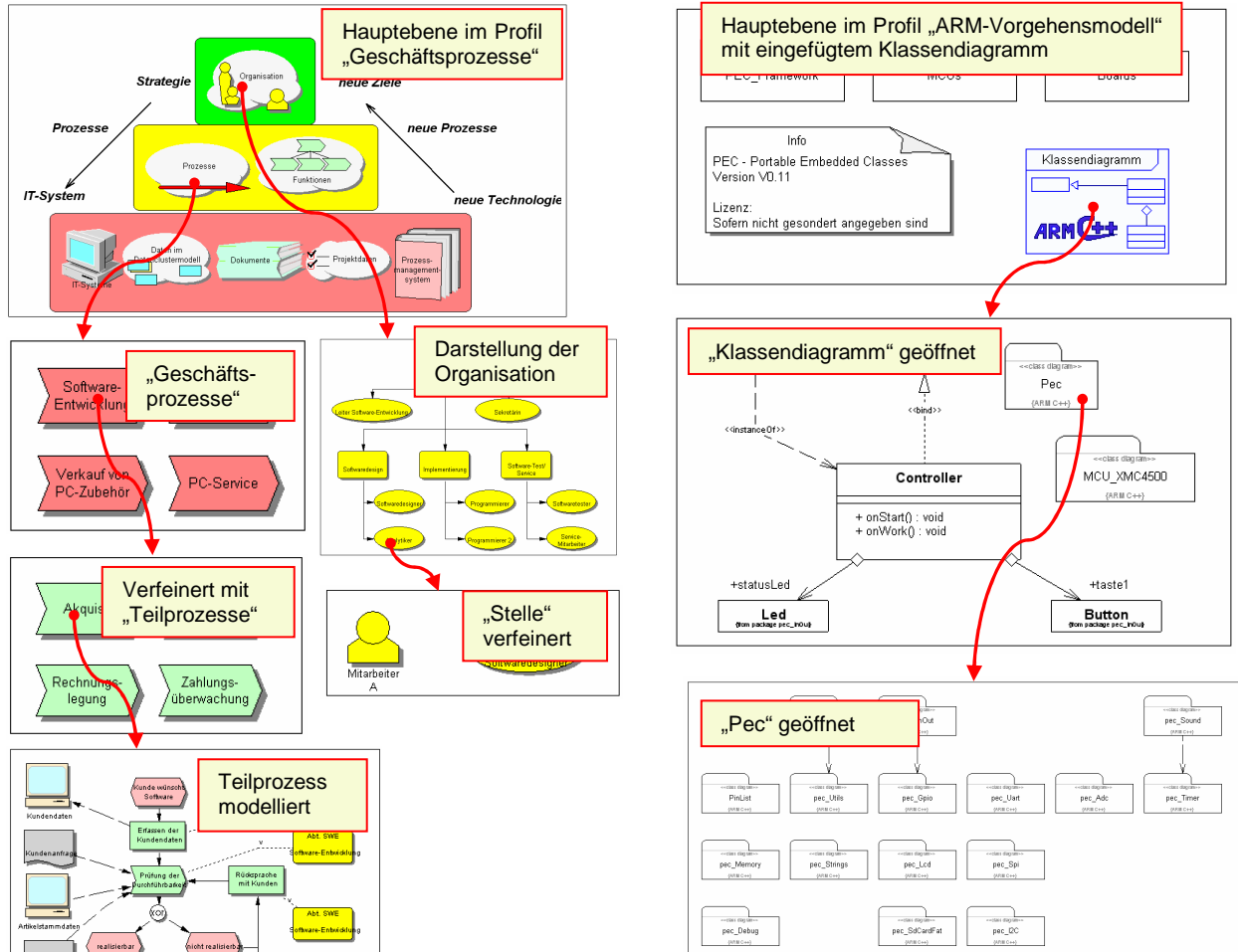


Abbildung 6: mögliche Aktionsmenüs für Mikrocontroller-Programmierung (links) und Dokumentengenerierung (rechts)

### 3.4 Modelle bearbeiten

#### 3.4.1 Modellebenen

Die grafische Darstellung der Modelle (Diagramme) erfolgt in Ebenen. Das ausgewählte Profil beim Anlegen eines Projektes stellt dabei die Hauptebene mit den markanten Objekten dieses Profils dar. Mit der Verfeinerung jedes dieser Objekte (auf dem Objekt „nach unten“) wird eine untere Ebene mit typischen Objekten erreicht, die in dieser Ebene modelliert werden können. Die Tiefenstaffelung der Ebenen ist profilabhängig.




**Abbildung 7: Von der Hauptebene in tiefer liegende Ebenen; links in der Geschäftsprozessmodellierung und rechts im Klassendiagramm**




### 3.4.2 Bewegen in Modellen

Das Bewegen innerhalb der Diagramme und zwischen den Ebenen kann mit verschiedenen Aktionen mittels Tastenkombinationen oder Mausoperationen erreicht werden. Im Folgenden eine kleine Auswahl; eine Übersicht dazu finden Sie im Anhang dieses Handbuches.

#### *Eine Ebene tiefer*

- in der Werkzeugleiste 
- Objekt markieren, aus dem Kontextmenü (rechte Maustaste) „nach unten (öffnen)“
- Objekt markieren, Strg + Enter
- Objekt markieren, Strg + Doppelklick

#### *Eine Ebene höher*

- in der Werkzeugleiste 
- auf Diagrammhintergrund aus dem Kontextmenü „nach oben“
- ESC-Taste



#### *Ein Objekt definieren = Eigenschaftsdialog*

- Objekt markieren, aus Kontextmenü „Definieren“
- Doppelklick mit Maustaste auf Verteiler des Objektes
- Objekt markieren, Alt + Enter

#### *Wechsel zwischen geöffneten Diagrammen*

- Auswahl mittels der Karteikarten über dem Editorfenster

#### *Zoom*

- Mousrad drehen, Zoom erfolgt zum Mauszeiger
- aus Kontextmenü oder Menüleiste/Ansicht „Zoom“ wählen
- in der Werkzeugleiste : mit der Maus Rahmen über zu vergrößernde Objekte ziehen;
- in der Werkzeugleiste : Zoom, dass alles sichtbar ist
- Taste F2: mit der Maus Rahmen über zu vergrößernde Objekte ziehen;
- Taste F2 zweimal: Zoom, dass alles sichtbar ist

#### *Diagrammansicht scrollen*

- Klick mit Maustaste auf Diagrammhintergrund; bei gedrückter Maustaste bewegen
- Scrollbalken benutzen
- Cursortasten

#### *Originalobjekt öffnen*

- Objekt markieren, aus Kontextmenü „Quelldiagramm öffnen“

#### *Referenz(en) eines Objektes öffnen*

- Objekt markieren, Strg + R (=Report) → „alle Diagramme des Objekts“ auswählen

#### *Quelltext öffnen (nur bei Objekten aus denen Quellcode generiert wird)*

- Taste F3
- Funktion aus Aktionsmenü wählen
- Funktion aus Kontextmenü wählen

### 3.4.3 Modellelemente anlegen

Das Original eines Modellelementes wird mittels Anlegen aus der Objektbibliothek erreicht. Dabei wird das gewünschte Element per Drag&Drop von der Objektbibliothek in das dazugehörige Diagrammfenster gezogen. Die Benennung des Objektes sowie weitere Eigenschaften werden im Definieren-Dialog (rechte Maustaste auf Objekt → Kontextmenü → "Definieren") zugewiesen.

Weitere Möglichkeiten zum Anlegen von Objekten können auch das Referenzieren, Kopieren oder Importieren sein (siehe folgende Abschnitte).

Verbindungen zwischen Objekten lassen sich einfach über den Verteiler herstellen.

### 3.4.4 Modellelemente auswählen

Die meisten Modellelemente lassen sich in Diagrammen auswählen (selektieren). Das Selektieren erfolgt in der Regel durch Anklicken (linke Maustaste) mit dem Mauscursor.

Eine Selektion per Klick mit der rechten Maustaste ist ebenfalls möglich, dabei öffnet sich nach dem Auswählen sofort das Kontextmenü für das selektierte Modellelement.

Das Betätigen der Tabulatortaste bewirkt das Auswählen des nächsten Modellelementes in der Reihenfolge des Anlegens der Elemente. Wird über die Auswahl mit der Tabulatortaste das letzte Element erreicht, beginnt die nächste Iteration wieder beim ersten Modellelement im Diagramm. Durch Betätigen der Umschalttaste kann die Iterationsrichtung umgeschaltet werden.

Es ist ebenfalls möglich, mehrere Modellelemente auszuwählen. Bei gehaltener Umschalttaste lassen sich mehrere Objekte (keine Kanten und Rahmen) durch Ziehen einer Markise mit dem Mauscursor auswählen. Über die Menüfolge *Bearbeiten/Diagramm/Alles markieren* können Sie alle Objekte des aktiven Diagramms selektieren.

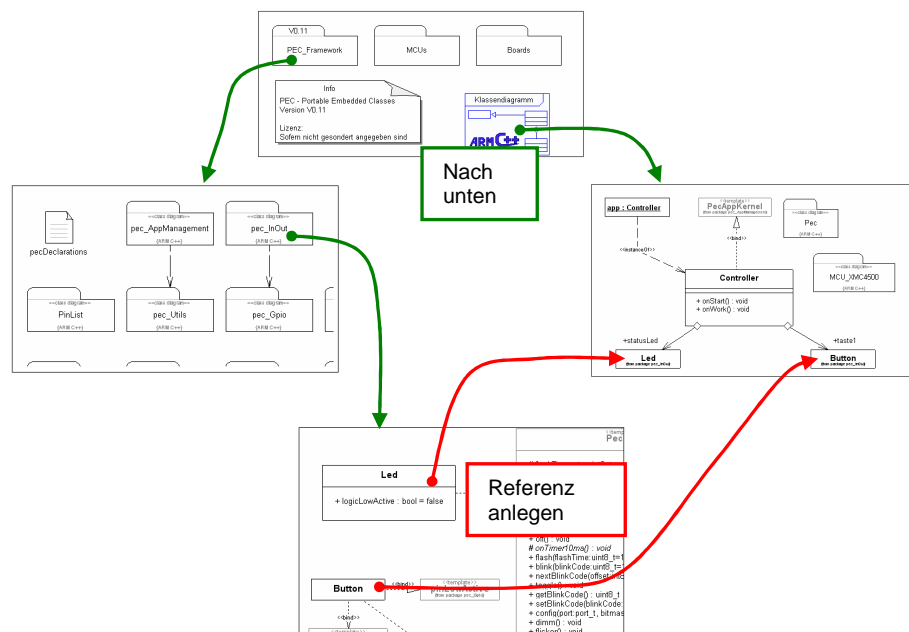
### 3.4.5 Modellelemente referenzieren

In einem Projekt können Objekte wiederholt eingesetzt werden, mit allen einmal definierten Eigenschaften. Zu diesem Zweck wird von dem Originalobjekt eine Referenz angelegt.

Referenzen können erzeugt werden, indem diese per Drag&Drop aus dem Navigator oder dem Explorer in das gewünschte Diagramm gezogen werden.

Des Weiteren können Sie per Drag&Drop aus einem anderen Diagramm gezogen werden. Zu diesem Zweck wird das Objekt mit der Maustaste „angefasst“ und über die Karteikarten am oberen Rand des Editorfensters bewegt bis zum gewünschten Diagramm. Die Diagrammanzeige wechselt. Das Objekt kann an der gewünschten Stelle positioniert werden.

In einigen Diagrammen können Modellelemente nur referenziert werden; das Original wurde in einem anderen Diagramm angelegt. Diese sind in der Objektbibliothek des aktuellen Diagramms rot beschriftet.



### 3.4.6 Modellelemente verschieben

Bestimmte Modellelemente (z.B. Knoten/Objekte) lassen sich im Diagramm frei anordnen. Das Positionieren erfolgt in der Regel durch Drag&Drop mit der Maus. Die Modellelemente werden dabei an einem Raster ausgerichtet. Das Raster erleichtert die Positionierung. Um die Rasterfangfunktion zu unterbinden, muss beim Drag&Drop eines Elementes die Umschalttaste betätigt werden. Viele Modellelemente lassen sich auch über die Cursortasten verschieben. Dabei bewegen sich die Modellelemente in Richtung der betätigten Taste auf dem Raster. Auch dabei lässt sich die Rasterfangfunktion durch die Umschalttaste unterbinden.

Bei Kanten wird ein Stützpunkt verschoben, der in Abhängigkeit der ausgewählten Kantenform zum gewünschten Ergebnis der Darstellung führt.

Rahmen haben nur eine eigene Position, solange diese leer sind. Ein Rahmen bezieht sich in Position und Größe immer auf die darin enthaltenen Modellelemente (umschließen). Beim Verschieben eines Rahmens mit enthaltenen Elementen wird nicht der Rahmen, sondern die enthaltenen Modellelemente verschoben.

### 3.4.7 Modellelemente löschen

Zu löschende Modellelemente müssen zuerst selektiert werden. Die Löschaufforderung erfolgt mittels der Taste „Entfernen“, der Werkzeugleiste oder über das Kontextmenü. Der Löschvorgang selbst ist abhängig von der Art des zu löschenden Modellelementes. Frei stehende Referenzen werden ohne Rückfrage gelöscht.

Originale werden in zwei Stufen gelöscht. Dabei erfolgt zuerst das Entfernen aus dem Diagramm und dann auf Anfrage auch das Löschen aus der Datenbank.

Objekte die verfeinert oder verknüpft sind, lassen sich nicht ohne erhöhten Aufwand aus dem Modell entfernen. Zuerst müssen die Modellbeziehungen des Elementes aufgelöst werden, bis es für den Löschvorgang isoliert ist. Dazu springt in entsprechenden Fällen der Löschassistent an. Dieser gibt dem Anwender die Möglichkeit, zuerst die Auswirkungen auf das Gesamtmodell zu beurteilen und kann dann auf Anforderung komplexe Löschvorgänge, zum Beispiel von ganzen Teilmodellen in der Modellhierarchie, durchführen.

### 3.4.8 Modellelemente verbinden

Knoten und Rahmen können über Kanten verbunden werden (Verbindungen). Verbindungen werden als Linien mit unterschiedlichen Linientypen und -formen sowie Symbolen (Anfang, Ende, Mitte) und diagrammspezifischen Beschriftungen dargestellt. Das Verbinden erfolgt in der Regel von dem Verteiler (rotes Rechteck an der Unterkante) eines selektierten Objektes. Dabei wird per Drag&Drop vom Verteiler die Verbindung vom Quellelement zum Zielelement gezogen. Es ist nicht nötig, exakt den Rand des Zielelementes zu treffen, sondern besser die Verbindung weit in das Zielobjekt hinein zu ziehen. Nach dem Loslassen der linken Maustaste wird die Verbindung erzeugt und die Randpunkte, bis zu denen die Verbindung gezeichnet wird (Clipping), automatisch ermittelt. Sie können während des Drag&Drop durch Klick mit der rechten Maustaste bereits den zukünftigen Stützpunkt der Verbindung festlegen.

### 3.4.9 Die Größe von Modellelementen verändern

In manchen Diagrammen verfügen Modellelemente über Anfassmarken zur Änderung der Größe. Die manuelle Größenänderung von Modellelementen ist jedoch in vielen Diagrammtypen nicht erwünscht. In diesen Diagrammen stehen keine Anfassmarken zur Größenänderung zur Verfügung. Sollte in Ausnahmefällen eine manuelle Anpassung der Objektgröße gewünscht sein, kann dies für selektierte Objekte durch die Kombination von gehaltener Steuerungstaste und den Cursortasten bzw. die Tasten Plus, Minus und Mal des Numerikfeldes erfolgen.

### 3.4.10 Modellelemente untersuchen (Report)

Modellelemente sind in den meisten Fällen nicht isoliert, sondern haben vielfältige Beziehungen zu anderen Modellelementen. Um bei Änderungen, wie zum Beispiel Umbenennen oder Löschen, mögliche Auswirkungen auf das gesamte Modell zu überblicken, bietet der Objektreport die Möglichkeit die Modellbeziehungen des ausgewählten Objektes auszuwerten. Wählen Sie ein Objekt aus und öffnen Sie den Reportdialog über das Hauptmenü, die Werkzeugleiste oder das Kontextmenü (diagrammabhängig). Wählen Sie die erforderliche Modellanfrage und starten den Report. Das Ergebnis der Modellanfrage (z.B. Alle angebotenen Objekte) wird als Liste angezeigt. Sie können dann zu diesen Modellelementen navigieren.

### 3.4.11 Modellelemente kopieren

Um eine Kopie eines Modellelementes zu erzeugen, wählen Sie im Menü oder in der Werkzeugleiste den Befehl „Kopieren“. Sie können dieses Objekt jetzt innerhalb des Projektes in ein Diagramm passenden Typs einfügen. Sollte das Diagramm den Objekttyp nicht akzeptieren, erhalten Sie dazu eine Meldung. Rahmenelemente, wie zum Beispiel „Klassen“, können mit deren Inhalt kopiert werden. Dabei werden jedoch Modellelemente in der Verfeinerung nach unten nicht berücksichtigt (flache Kopie). Innerhalb eines Diagramms können Sie von einzelnen Objekten durch Drag&Drop bei gleichzeitig gedrückter Steuerungstaste Kopien erzeugen.

#### Hinweis:

*Beachten Sie, dass eine Kopie immer dem kopierten Element entspricht. Das heißt, die Kopie eines Originals ist wieder ein neues Original, die Kopie einer Referenz ist eine neue Referenz auf das ursprüngliche Original.*

### 3.4.12 Modellelemente ausschneiden

Zum Ausschneiden eines Modellelements wählen Sie in der Werkzeugleiste oder im Menü *Bearbeiten/Objekt/Ausschneiden*. Sie können dieses Objekt jetzt innerhalb des Projektes in ein Diagramm passenden Typs einfügen. Sollte das Diagramm den Objekttyp nicht akzeptieren, erhalten Sie dazu eine Meldung. Rahmenelemente, wie zum Beispiel „Klassen“, können mit deren Inhalt verschoben werden. Dabei werden auch Modellelemente in der Verfeinerung nach unten berücksichtigt.

#### Hinweis:

*Beachten Sie, dass dieser Vorgang starke Auswirkungen auf die Modellstruktur hat. Vergewissern Sie sich über den Objektreport über die möglichen Auswirkungen.*

### 3.4.13 Diagramme kopieren (flache Kopie)

Sie können Kopien von gesamten Diagrammen anlegen und einfügen. Dabei unterscheidet man zwischen einer flachen Kopie, bei der alle Elemente der gezeigten Ebene (Diagramm) kopiert werden und einer tiefen Kopie, bei der alle Elemente dieser Ebene und der darunter liegenden Ebenen (Teilhierarchie) kopiert werden. Eine flache Kopie erzeugen Sie über die Menübefehle *Bearbeiten/Diagramm/Diagrammexport anlegen (flache Kopie)* und *Bearbeiten/Diagramm/Diagrammimport einfügen (flache Kopie)*.

#### Hinweis:

*Beachten Sie, dass das Einfügen der Diagrammkopie nur in einem Diagramm des gleichen Typs und innerhalb des gleichen Projektes möglich ist.*

### 3.4.14 Modelle kopieren (tiefe Kopie)

Sie können Kopien von gesamten Teilmodellen anlegen und einfügen. Dabei unterscheidet man zwischen einer flachen Kopie, bei der alle Elemente der gezeigten Ebene (Diagramm) kopiert werden und einer tiefen Kopie, bei der alle Elemente dieser Ebene und der darunter liegenden Ebenen (Teilhierarchie) kopiert werden. Eine tiefe Kopie erzeugen Sie über die Menübefehle *Bearbeiten/Diagramm/Diagrammkopie anlegen (tiefe Kopie)* und *Bearbeiten/Diagramm/Diagrammkopie einfügen (tiefe Kopie)*.

#### Hinweis:

*Beachten Sie, dass eine tiefe Kopie dem Wesen nach ein Export und Import von Teilmodellen darstellt. Das Einfügen eines solchen Imports in ein anderes Projekt kann zu Modellkonflikten führen.*

### 3.4.15 Die Modellhierarchie bearbeiten (Jo-Jo)

Die Modellhierarchie kann durch Einfügen neuer Ebenen oder Auflösen von Modell-Ebenen verändert werden (Jo-Jo). Diese Funktion erreichen Sie am besten über das Haupt- oder Kontextmenü. Für das Zusammenfassen einer gewünschten Anzahl von Modellelementen zu einer neuen Ebene, selektieren Sie diese und wählen im Menü *Bearbeiten/Objekt/Jo-Jo/Objekte zusammenfassen*. Es wird ein neues Objekt erzeugt und in das Diagramm eingefügt. Die ausgewählten Objekte wurden dem neuen Objekt untergeordnet. Bei der Funktion *Jo-Jo/Objekt auflösen* werden die untergeordneten Objekte in das aktuelle Diagramm eingefügt und das ausgewählte Objekt gelöscht.

## 3.5 Die Ansicht verändern

### 3.5.1 Fensterinhalt verschieben

Modelle werden oft viel größer als der sichtbare Bereich des Diagrammfensters. Um den sichtbaren Teil eines Diagramms zu verändern (Scroll), haben Sie folgende Möglichkeiten:

- die Scrollleisten des Fensters benutzen
- den Diagrammhintergrund per Drag&Drop schieben
- wenn kein Modellelement ausgewählt ist, mit den Cursortasten scrollen
- Menüpunkt *Ansicht/Fenstermitte* um zum Diagrammursprung zurück zu kehren

#### Hinweis:

*Beachten Sie, der Diagrammursprung liegt nicht wie bei Textverarbeitungen oben links des Arbeitsbereiches, sondern wie bei einer MindMap oder vielen Konstruktionsprogrammen (CAD) in der Mitte (Nullpunkt, Fenstermitte). Das Modell lässt sich somit in alle Richtungen entwickeln. Vergleichen Sie dazu die Position der Scrollbalken. Versuchen Sie auf keinen Fall zu Beginn der Arbeit die Scrollposition oben links herzustellen!*

### 3.5.2 Fensterinhalt vergrößern oder verkleinern

Um einen Überblick über größere Diagramme zu gewinnen, können Sie die Darstellung vergrößern oder verkleinern (Zoom). Eine elegante Möglichkeit das aktuelle Diagramm zu zoomen, ist das Scrollrad der Maus. Sie können aber auch über das Kontextmenü oder den Menüpunkt *Ansicht* feste Zoom-Faktoren einstellen.

Die Funktion *Ansicht/Zoom/Alles* aus dem Menü oder der Werkzeugleiste passt das Diagramm an die Fenstergröße an. Eine weitere Möglichkeit das Diagramm zu zoomen, erhalten Sie über die Tasten des Numerikfeldes:

- Numerikfeld + ... Ansicht vergrößern
- Numerikfeld - ... Ansicht verkleinern
- Numerikfeld \* ... Ansicht alles (einpassen)

### 3.5.3 Fensterinhalt farbig oder als Kontur darstellen

Temporär kann die farbige Darstellung von Elementen unterbunden werden. Dazu können Sie über das Kontextmenü oder das Menü *Ansicht/Darstellung/Konturen* zwischen farbiger oder Konturdarstellung umschalten.

### 3.5.4 Die Schriftart ändern

Die in Diagrammen verwendete Schriftart lässt sich über die Werkzeugleiste oder den Menüpunkt *Einstellungen/Schriftart...* verändern. Dabei sollte die Schriftgröße 10 bis 12 eingehalten werden.

## 3.6 Druckfunktionen in SiSy

Sie haben in SiSy verschiedene Möglichkeiten Projektinhalte wie Grafiken, Übersichten, Quellcodes oder ganze Projektdokumentationen zu drucken. Dabei ist zu berücksichtigen, dass in SiSy bestimmte Informationen wie zum Beispiel die Darstellung eines Programmablaufplanes sichtbare Elemente eines Diagramms sind und andere Teile wie zum Beispiel der Quellcode eines Elementes nur über Dialoge oder bei Selektierung des Elementes sichtbar sind. Je nachdem, welchen Inhalt Sie dokumentieren wollen, richtet sich die Auswahl der betreffenden Druckfunktion.

In SiSy sind u.a. Druckfunktionen integriert, die ein Dokument generieren und dieses nach Microsoft Word exportieren. Für diese Funktion muss auf dem PC MS Word installiert sein. Die Dokumentengenerierung basiert auf Makros; die Vertrauensstellung für Makros muß aktiviert sein.

Vorgehensweise in Word (ab Word 2007):

- Wählen Sie die Schaltfläche „Office“
- Menüfolge: *Word-Optionen/Vertrauensstellungscenter/Einstellungen für das Vertrauensstellungscenter/Einstellungen für Makros*
- Aktivieren Sie bei „Einstellungen für Makros“ die Auswahl „Alle Makros aktivieren“
- Setzen Sie im Unterpunkt „Entwicklermakroeinstellungen“ den Haken bei „Zugriff auf das VBA-Projektobjektmodell vertrauen“

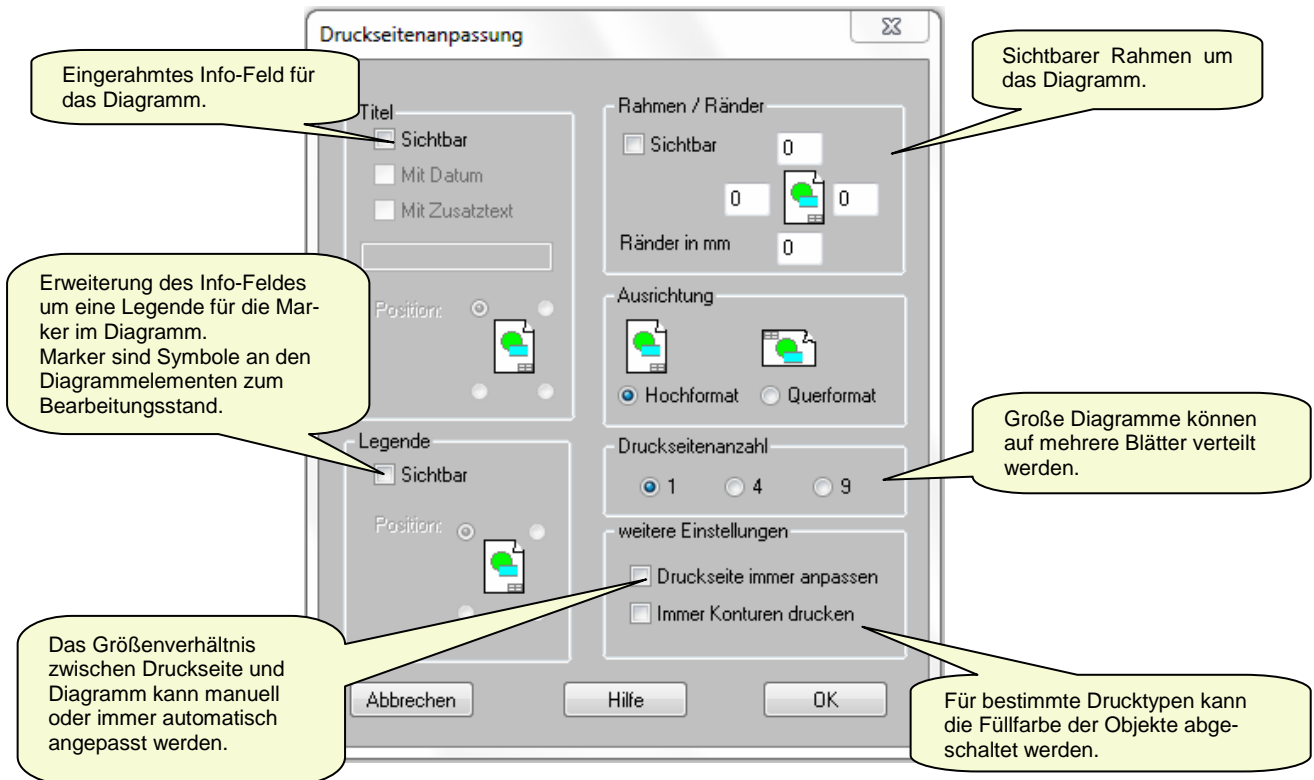
#### Hinweis:

Für die Nutzung anderer Word-Versionen finden Sie Hinweise in unseren FAQs [www.myMCU.de](http://www.myMCU.de) → Shop → Registerkarte „FAQ“ → Suchbegriff „Dokument“

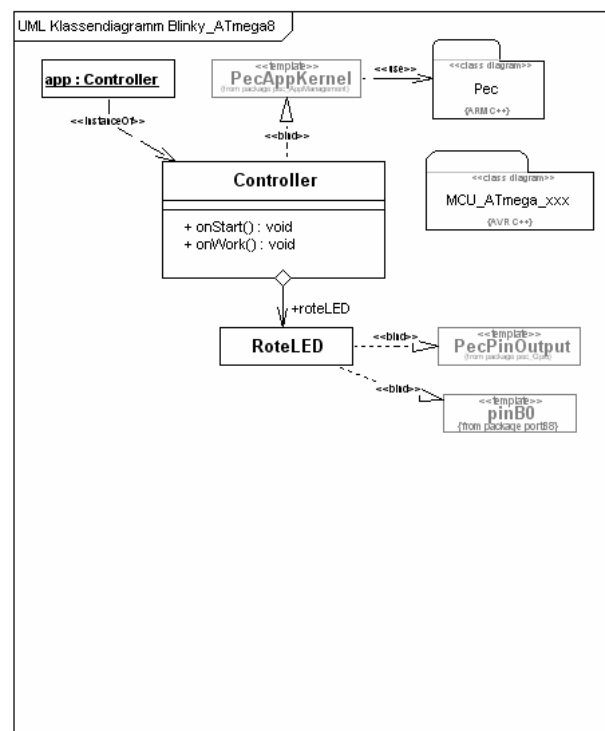
### 3.6.1 Diagramme drucken

Für den Druck eines einzelnen Diagramms, also Druck des sichtbaren Inhalts eines Diagrammfensters, gehen Sie wie folgt vor:

- ggf. Projekt öffnen
- das gewünschte Diagramm öffnen
- die Menüfolge *Bearbeiten/Diagramm/Drucken...* wählen oder das Druckersymbol in der Werkzeugleiste aktivieren
- Sie erhalten den Dialog zum Einrichten der Druckseite, wählen Sie die gewünschten Optionen aus!



- Die Druckvorschau wird angezeigt. Der Druckvorgang kann hier abgebrochen werden, um die Einstellungen zu überarbeiten. Dabei kann die relative Position und das Größenverhältnis der Druckseite zum Diagramm verändert werden. Der Dialog zum Verändern der Einstellungen lässt sich per Doppelklick auf den Selektierungsmarken der Druckseite öffnen.
- Die Druckseitenansicht lässt sich über die Menüfolge *Ansicht/Druckseite* ein- und ausblenden
- Der Druckvorgang kann über das Druckersymbol in der Werkzeugleiste jederzeit gestartet werden. Der Ausdruck erfolgt auf dem Standarddrucker des Systems




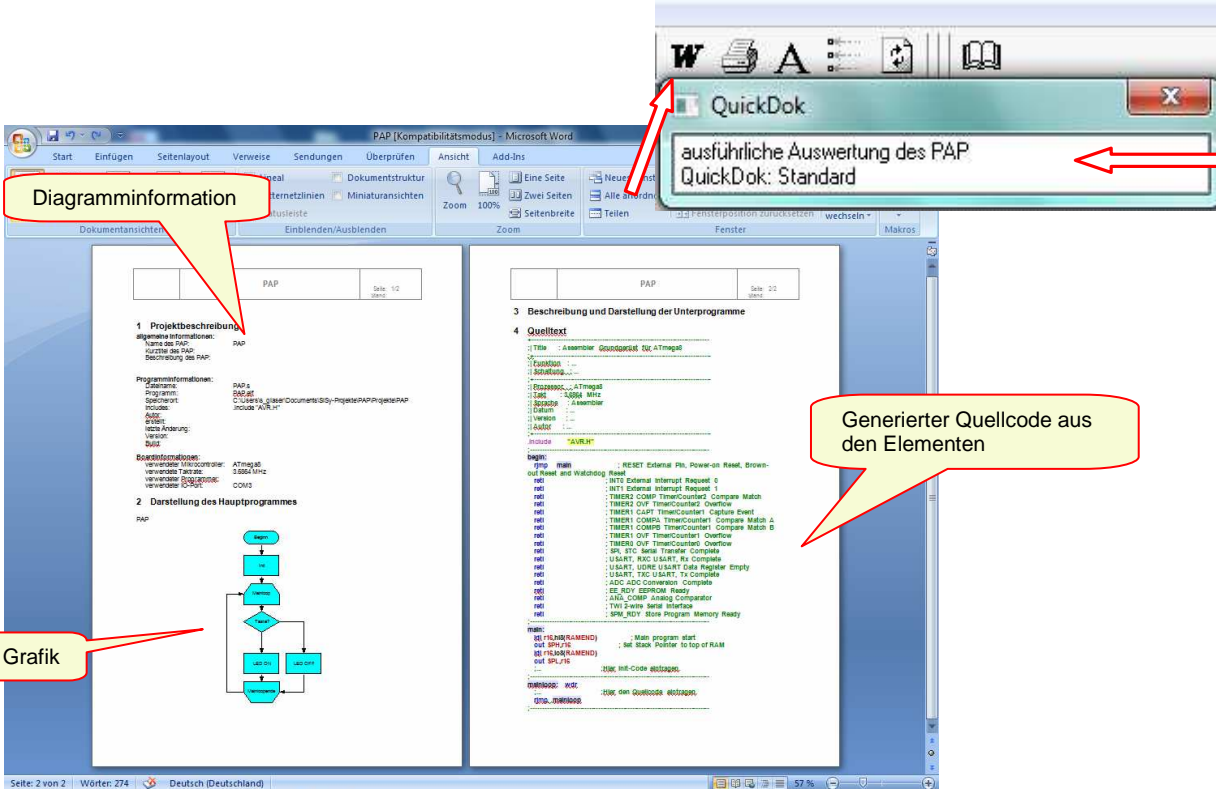
### 3.6.2 Grafiken und Inhalte drucken

Viele Projektinformationen sind kein sichtbarer Teil von erstellten Diagrammen. Diese wurden über Dialoge und Masken eingegeben und stehen als Attribute in der Projektdatenbank zur Verfügung. Um diese Informationen auszudrucken, stellt SiSy für jeden Diagrammtyp eine Reportfunktion zur Verfügung, mit der die wesentlichen Informationen, Inhalte und Attribute des Diagramms und der Objekte in einem Diagramm, als Word-Dokument generiert werden.

Auf dem PC muß MS Word installiert sein.

Die Reportfunktion zum Generieren des Word-Dokumentes aktivieren Sie über das Symbol „QuickDok“ in der Werkzeugleiste.

- Symbol  (QuickDok) in der Werkzeugleiste wählen
- Aus der Liste der angebotenen Reportfunktionen auswählen
- Dateinamen für den Report vergeben
- Makros aktivieren



The screenshot shows a Microsoft Word document titled 'PAP [Kompatibilitätsmodus]'. The document content is divided into sections: '1 Projektbeschreibung', '2 Darstellung des Hauptprogrammes' (which includes a flowchart labeled 'Grafik'), '3 Beschreibung und Darstellung der Unterprogramme', and '4 Quelltext' (which contains assembly code). A 'QuickDok' dialog box is open at the top right, displaying 'ausführliche Auswertung des PAP' and 'QuickDok: Standard'. Red callout boxes highlight the 'Diagramminformation' (Project Description), the 'Grafik' (Flowchart), and the 'Generierter Quellcode aus den Elementen' (Generated source code from elements).



### 3.6.3 Nur Quellcodes drucken

Für das Ausdrucken von Quellcodes bietet SiSy einen speziellen Quellcode-Druckassistenten. Damit ist es möglich, Quellcodes formatiert, in Syntaxfarben und mit Zeilennummern auszudrucken.

kleines Programm, Quellcode drucken

- kleines Programm selektieren
- rechte Maustaste „Definieren...“
- Dialogseite „Quellcode“
- Schaltfläche „>>öffnen“

UML-Klassendiagramm

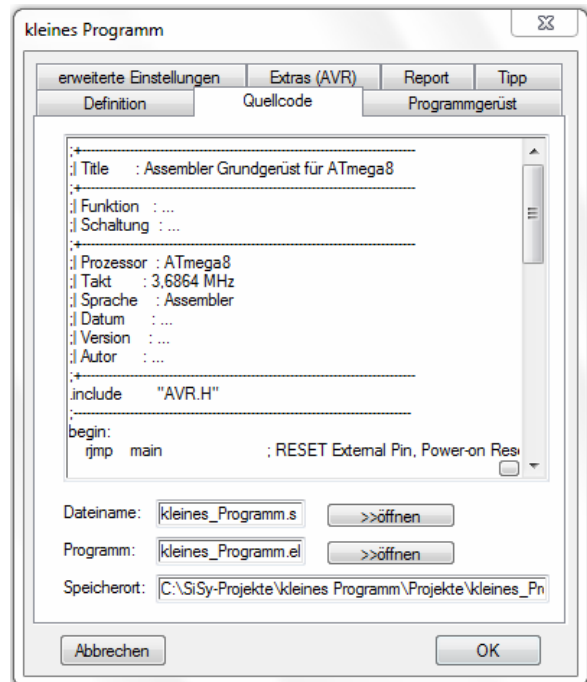
- Klassendiagramm öffnen
- Symbol für „QuickDok“ aktivieren
- Reportfunktion auswählen
- Name vergeben

oder komplettes Dokument

- Aktionsmenü aktivieren
- „Quick-Dokument erzeugen“

Programmablaufplan, Quellcode drucken

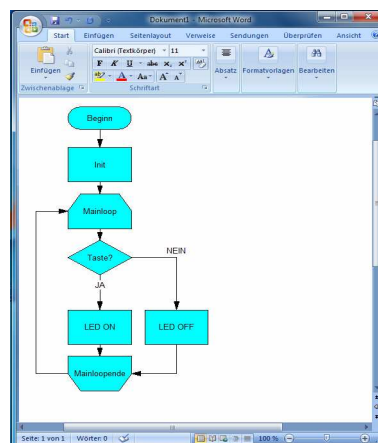
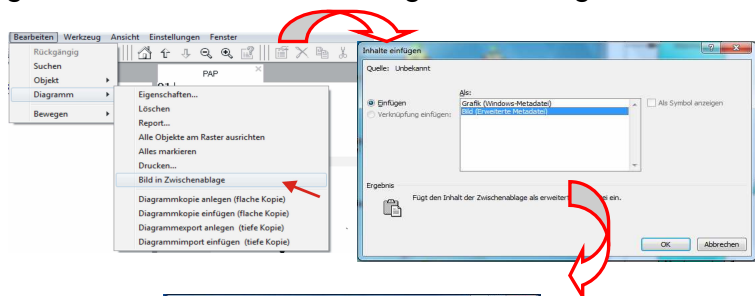
- PAP öffnen
- Aktionsmenü aktivieren
- „Quellcodedatei öffnen/drucken“



### 3.6.4 Nutzen der Zwischenablage

In Projektdokumentationen ist es oft vorteilhaft, die in SiSy erstellten Diagramme als Bilder einzufügen. Die Diagramme werden in SiSy nicht als Bilder gespeichert, sondern zur Laufzeit aus den Modellinformationen generiert. Für die weitere Nutzung der Diagrammbilder existiert die Funktion „Bild in Zwischenablage“. Dabei erstellt SiSy eine skalierbare Vektorgrafik (WMF) und legt diese in die Zwischenablage (Copy). Die Grafik kann nun von anderen Anwendungen über den Befehl „Einfügen“ beliebig weiter verwendet werden.

- Diagramm in SiSy öffnen
- Menüfolge *Bearbeiten/Diagramm/Bild in Zwischenablage* wählen
- Zielanwendung öffnen
- Menüfolge *Start/Einfügen* oder *Start/Einfügen/Inhalte einfügen* wählen
- Gegebenenfalls Grafikformat wählen




### 3.6.5 Dokumentengenerierung

Die in einem Projekt in unterschiedlichen Ebenen und Diagrammen erstellten Modelle, Abläufe, Definitionen, Quelltexte usw. können mit Hilfe der Dokumentengenerierung zu einem Gesamtdokument zusammengefasst werden. Dieses Gesamtdokument kann vervollständigt werden mit Auswertung, Grafiken und Tabellen.

Für die Dokumentengenerierung muß MS Word auf dem PC installiert sein.

#### QuickDok

Mit Hilfe von QuickDok (Symbol  in der Werkzeugleiste) kann die Dokumentengenerierung mit Standardvorlagen genutzt werden. Die Standardvorlagen sind abhängig von der SiSy-Ausgabe und dem aktuellen Diagramm.

Auf Wunsch können diese Vorlagen kundenspezifisch erstellt und angepasst werden.

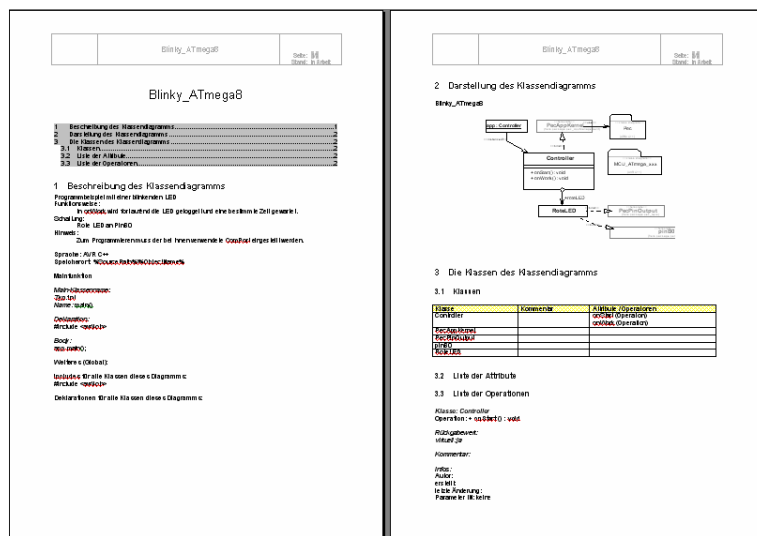


Abbildung 8: Auszug einer Dokumentengenerierung mit QuickDok

#### Freie Dokumentengenerierung

In Abhängigkeit der SiSy Ausgabe gibt es Objekte vom Typ „Dokument“. In der Verfeinerung dieser Objekte kann der Dokumenteninhalt individuell festgelegt werden. Das Vorgehen ist im Handbuch zur Dokumentengenerierung beschrieben.

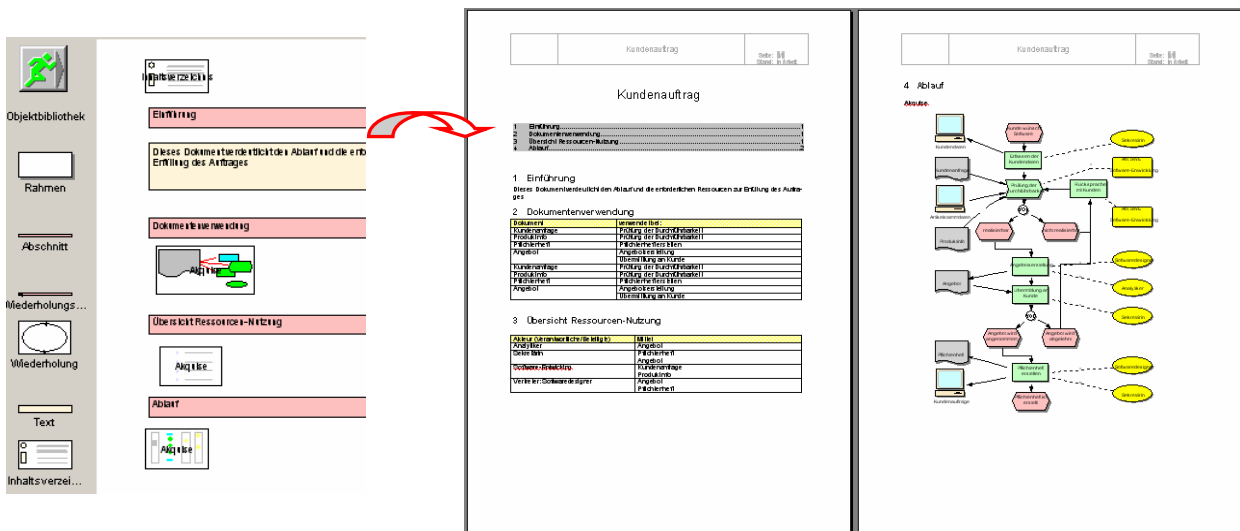


Abbildung 9: links modellierter Dokumenteninhalte in SiSy; rechts generiertes Dokument

### 3.7 SiSy Add-Ons

Das Modellierungswerkzeug SiSy besteht aus seinen Kernkomponenten:

- Anwendungssystem SiSy mit
  - Rahmenanwendung
  - Assistent
  - Diagrammfenster
- Laufzeitbibliotheken
  - Datenbanktreiber (Repository)
  - Grafikbibliotheken
  - Metamodell-Engine
  - SiSy BASIC Interpreter
- Allgemeine Hilfe
- ... und installierten Add-Ons

Add-Ons enthalten zusätzliche Komponenten, SiSy-BASIC-Skripte, erweiterte Hilfen und spezielle Informationen für die Modellierung (Metamodell). Dabei können Add-Ons wiederum aus Add-Ons thematisch zusammengestellt werden. Diese zusätzlichen Bausteine werden während der Laufzeit vor allem durch die Metamodell-Engine verarbeitet. Dadurch können beliebige Modellierungsaufgaben durch SiSy verarbeitet werden. Es ist nur das entsprechende Add-On erforderlich. Für die aktuelle Version von SiSy sind zum Beispiel folgende Add-On Zusammenstellungen verfügbar:

- **AVR:** Programmierung von AVR Mikrocontrollern
  - einfache Programmierung mit dem Zeileneditor in Assembler und C
  - grafische Programmierung mit dem Programmablaufplan in Assembler, dem Struktogramm in C, dem UML-Klassendiagramm in C++ und dem UML-Statechart in C++
  - Codegenerierung mit dem myAVR Code-Wizard für Assembler und C
  - myAVR Werkzeuge für Test und Kommunikation mit AVR Controllern
- **ARM:** Programmierung von ARM Mikrocontrollern, explizit STM32 und XMC
  - einfache Programmierung mit dem Zeileneditor in C
  - grafische Programmierung mit dem Struktogramm in C, dem UML-Klassendiagramm in C++ und dem UML-Statechart in C++
  - Integrierter Debugger für ARM Mikrocontroller
- **SPRG:** einfache Programmierung für Konsolen- und GUI-Anwendungen
  - Integrierte Programmiersprachen C/C++, JAVA, C#, PASCAL, Assembler
  - Integrierter Debugger
- **UML:** objektorientierte Systementwicklung für Konsolen- und GUI-Anwendungen
  - Typische UML-Diagramme z.B.: Klassendiagramm, Anwendungsfalldiagramm, Aktivitätsdiagramm, Zustandsdiagramm, Sequenzdiagramm, u.a.
  - Codegenerierung für C++, C#, JAVA
  - Integrierter Debugger mit Modellnachführung im Schrittbetrieb
- **SVL:** Smart Visual Library
  - Einfache bausteinartige Entwicklung von Windows-Anwendungen (RAD Rapid Application Development) im UML Klassendiagramm
- **SysML:** System Modeling Language
  - Fachübergreifende Systementwicklung mit dem OMG Sprachstandard
- **SA/SD:** Strukturierte Techniken zur Systementwicklung
  - Round Trip Engineering mit den klassischen Analyse und Entwurfswerkzeugen bis zur Codegenerierung, Strukturierte Analyse, Entity Relationship Diagramm mit SQL-Generierung und Struktogramme mit Codegenerierung
- **DOKGEN:** Dokumentengenerierung
  - Generieren von Dokumentationen direkt aus den Modellen heraus.
- **GPM / QM:** Geschäftsprozessmodellierung und Qualitätsmanagement
  - Modellierung von integrierten Managementsystemen

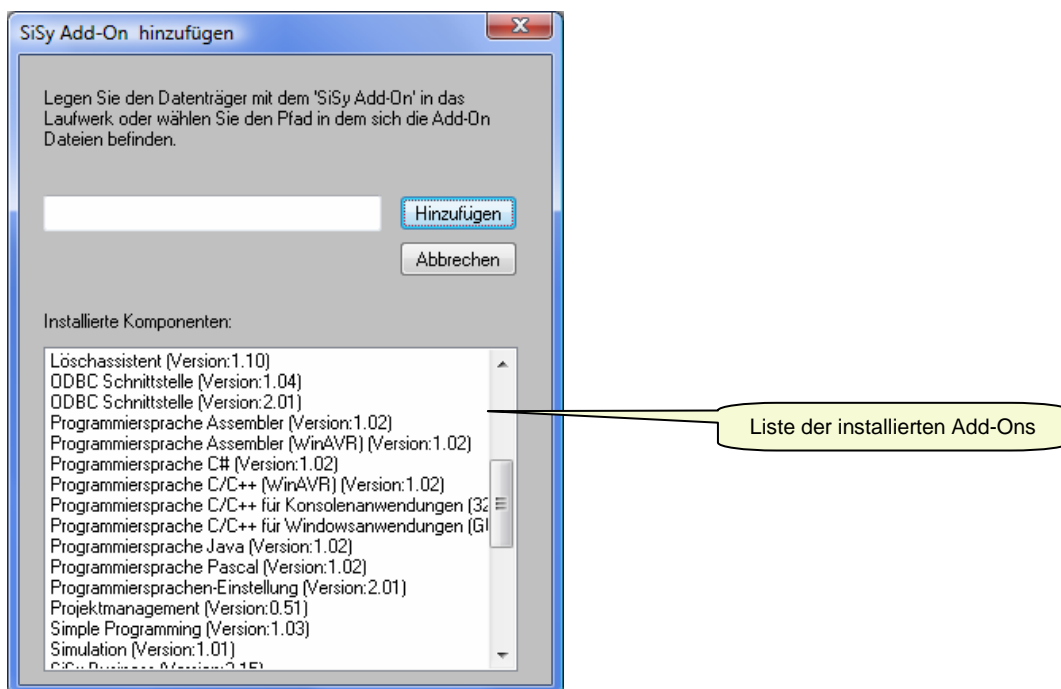
Aus den verfügbaren Add-Ons werden entsprechend der unterschiedlichen Einsatzgebiete des Modellierungswerkzeuges SiSy spezielle Ausgaben zusammengestellt. Die jeweiligen Ausgaben können jederzeit durch weitere Add-Ons ergänzt werden. Es sind unter anderem folgende SiSy-Ausgaben verfügbar (Stand Juni 2016):

### **Ausgabe (Auswahl)**

- SiSy Professional  
umfasst alle verfügbaren Add-Ons
- SiSy Business  
umfasst Add-Ons für Prozessmodellierung, Qualitäts- und Projektmanagement
- SiSy Developer  
umfasst umfangreiche Add-Ons zur Systementwicklung
- SiSy Microcontroller ++  
umfasst die Add-Ons AVR, ARM, ausgewählte Teile der UML, SVL und SysML
- SiSy AVR  
umfasst die Add-Ons AVR, ausgewählte Teile der UML
- SiSy STM32 und SiSy XMC  
umfasst die Add-Ons ARM, ausgewählte Teile der UML und SysML

### **Add-Ons anzeigen**

Für die Anzeige der installierten Add-Ons nutzen Sie den Menüpunkt *Einstellungen/ Add-On hinzufügen*.

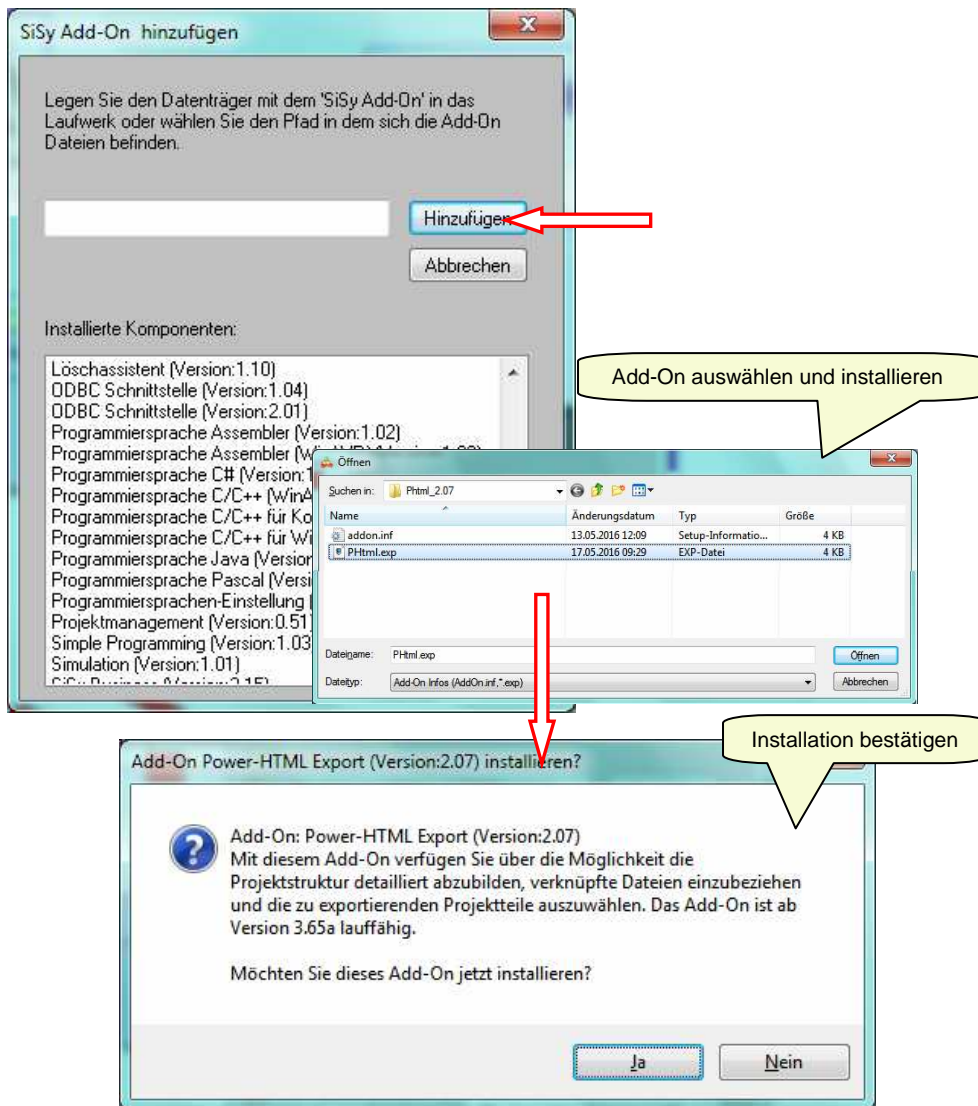


### **Hinweis:**

*Beachten Sie, Änderungen an den Add-Ons haben in jedem Fall Auswirkungen auf die Funktionalität von SiSy und abhängiger Add-Ons.*

## Add-Ons hinzufügen

Add-Ons sind in der jeweiligen SiSy Ausgabe integriert. Das Hinzufügen von Add-Ons ist nur dann erforderlich, wenn Sie ein Add-On zusätzlich erworben haben. Für das Hinzufügen nutzen Sie den Menüpunkt *Einstellungen/Add-On hinzufügen*. Wählen Sie im Add-On Dialog die Schaltfläche „Hinzufügen“ und wählen Sie aus Ihrer Datenquelle das entsprechende Add-On aus. SiSy Add-Ons besitzen die Dateierweiterung \*.exp. Folgen Sie den weiteren Anweisungen.



### Hinweis:

Für die Installation von Add-Ons benötigen Sie Administratorrechte. Beachten Sie, dass Änderungen an den Add-Ons in jedem Fall Auswirkungen auf die Funktionalität von SiSy und abhängiger Add-Ons haben.

## 3.8 SiSy LibStore

SiSy LibStore ist eine online-Sammlung von Vorlagen, Mustern, Beispielen und Bibliotheken. Diese werden Ihnen bei der Arbeit mit SiSy angeboten, sobald bei der Modellierung im jeweiligen Diagramm SiSy LibStore verfügbar ist und Sie online sind. Die Nutzung der im SiSy LibStore verfügbaren Inhalte ist, wenn durch den Autor nicht anderweitig geregelt, an die Lizenzbedingungen Ihrer SiSy-Ausgabe gebunden.

### 3.8.1 ein Projekt mit SiSy LibStore anlegen

Nach dem Start von SiSy legen Sie ein neues Projekt an und wählen ein Profil aus. Es öffnet bereits SiSy LibStore und bietet, soweit für dieses Profil vorhanden, eine Auswahl von Vorlagen an.

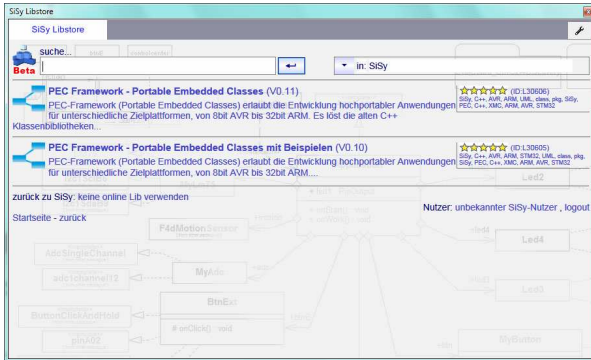


Abbildung 10: SiSy LibStore aus AVR Profil

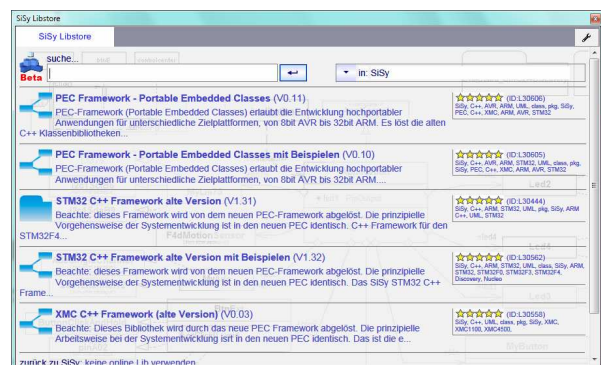
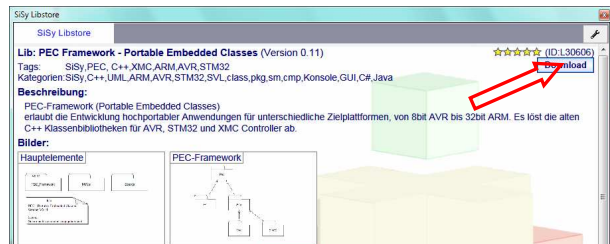


Abbildung 11: SiSy LibStore aus ARM Profil

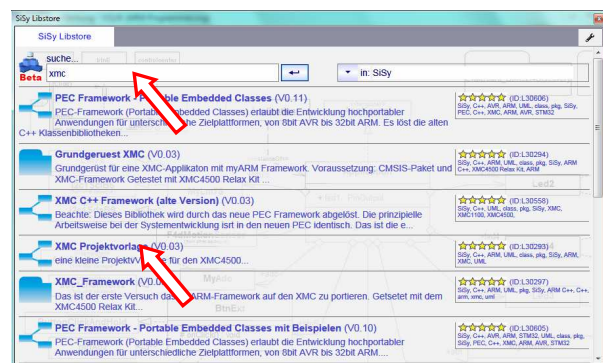
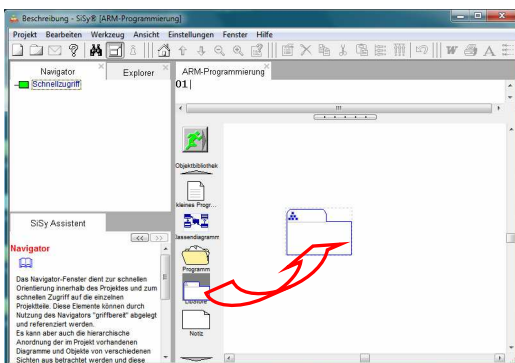
Wählen Sie Ihre gewünschte Vorlage aus und aktivieren Sie im nächsten Fenster die Schaltfläche „Download“. Die ausgewählte Vorlage wird in ein leeres Diagramm geladen; sie steht Ihnen uneingeschränkt zur weiteren Arbeit zur Verfügung.



### 3.8.2 SiSy LibStore in einem Projekt nutzen

Während der Arbeit in einem Projekt können Sie ebenfalls auf den SiSy LibStore zugreifen. Ziehen Sie dafür aus der Objektbibliothek ein Objekt vom Typ „LibStore“ in das geöffnete Diagramm. In Abhängigkeit Ihrer SiSy-Ausgabe und dem Diagrammtyp werden Ihnen von SiSy LibStore Vorlagen, Beispiele, usw. angeboten. Zur Einschränkung können Sie in der Suchzeile Begriffe eingeben, wie z.B. Controllernamen, dann Ihre Auswahl treffen und den Download ausführen.

Des Weiteren öffnet SiSy LibStore in Abhängigkeit von Ausgabe und Profil beim Öffnen von Objekten(auf Objekt *nach unten*).



## 3.9 Die Hilfsfunktionen in SiSy

Nutzen Sie die zahlreichen Hilfen und Vorlagen, die SiSy dem Entwickler bietet!

### 3.9.1 SiSy Online-Tutorials

SiSy Online-Tutorials ermöglichen es dem Anwender sich Schritt für Schritt in ein ausgewähltes Thema einzuarbeiten. Sie sind chronologisch und didaktisch aufgebaut; trotzdem können Abschnitte übersprungen werden. Zu den einzelnen Themen werden sowohl theoretische Grundlagen erörtert als auch ausführliche Beispielprogramme beschrieben. Teilweise erfolgt eine Video-Zusammenfassung.

Der Zugang zum Tutorial kann sofort nach dem Start von SiSy erfolgen, über die entsprechende Schaltfläche (vgl. Abbildung 1) und außerdem auch während der Arbeit mit SiSy über die Menüfolge *Hilfe/Tutorials*.

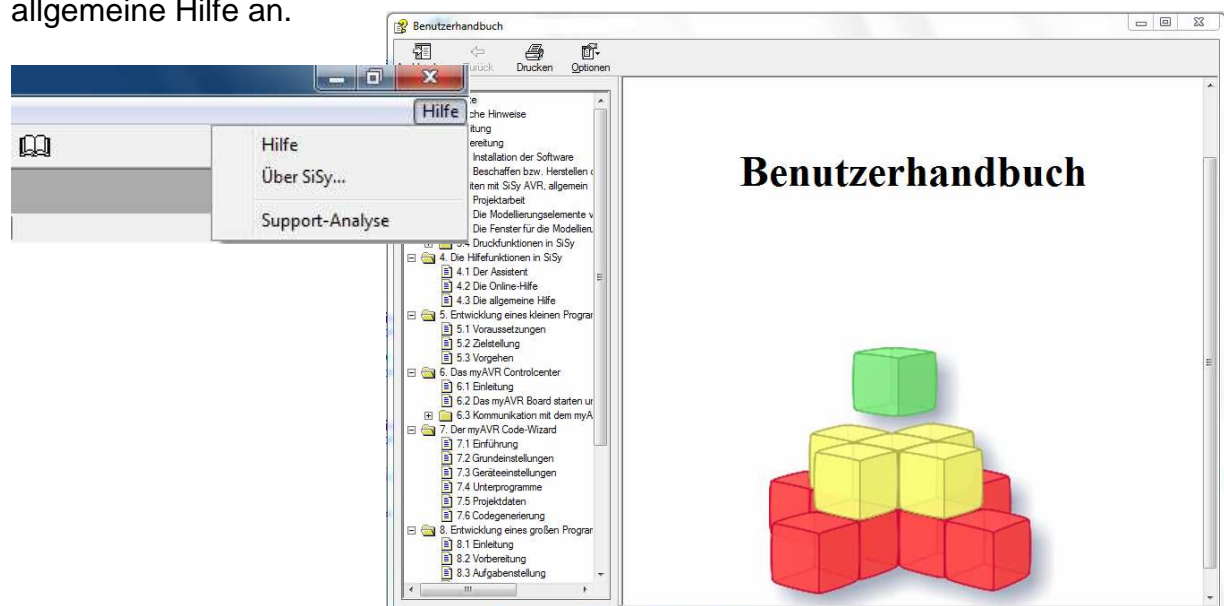
Die vollständige Nutzung der SiSy Tutorials ist, wenn durch den Autor nicht anders weitig geregelt, an die Lizenzbedingungen Ihrer SiSy-Ausgabe gebunden.

Es sind gegenwärtig folgende Tutorials verfügbar (Stand: Juni 2016):

- AVR C++ Tutorial:  
die objektorientierte Programmierung von AVR-Mikrocontrollern
- AVR UML Tutorial:  
vom Einstieg zur komplexen Anwendungsentwicklung für AVR-Mikrocontroller mit UML
- STM32 Tutorial für C/C++ und UML:  
objektorientierte Programmierung von STM32-Mikrocontrollern und Anwendungsentwicklung mit UML
- XMC Tutorial für C/C++ und UML:  
objektorientierte Programmierung von XMC-Mikrocontrollern und Anwendungsentwicklung mit UML
- Micro Graphic Library - µGL:  
vom Einstieg zur komplexen Anwendungsentwicklung mit der microGL für graphische Benutzeroberflächen auf der Basis von UML-Klassendiagrammen
- SVL, Smart Visual Library Tutorial:  
vom Einstieg zur komplexen Anwendungsentwicklung mit der SVL auf der Basis von UML-Klassendiagrammen.

### 3.9.2 Die allgemeine Hilfe

SiSy bietet neben der direkten Hilfe bei der Eingabe von Schlüsselworten auch eine allgemeine Hilfe an.











### 3.9.3 Der Assistent

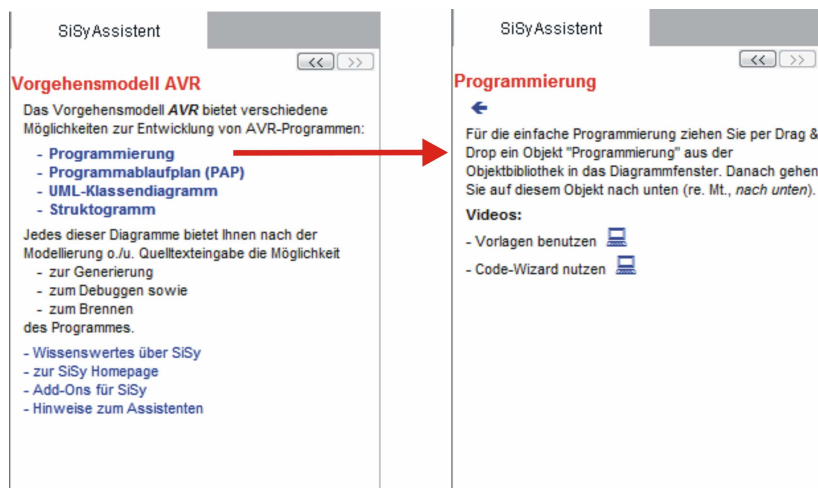
Der Assistent ist hilfreich bei der Unterstützung und Führung des Nutzers im Umgang mit SiSy. Der Assistent kann über die Werkzeugleiste (Symbol Fragezeichen) geöffnet werden, falls dies nicht beim Programmstart erfolgte. Der Assistent begleitet den Nutzer im gesamten Projekt. Er liefert immer passende Informationen zum aktuellen Diagrammtyp. Über verschiedene Links können jederzeit weitere Hilfethemen aufgerufen oder Vorlagen in das Projekt geladen werden. Vorteilhaft ist es jedoch, Vorlagen aus dem SiSy LibStore (vgl. Abschnitt 3.8) zu verwenden, da diese kontinuierlich online aktualisiert werden.

*Hinweis: Der Assistent ist auf die jeweilige Ausgabe von SiSy, die verfügbaren Add-Ons und das gewählte Modell bezogen.*

Bedeutung der verwendeten Symboliken im SiSy Assistenten:

	weitere Informationen anzeigen; öffnet eine Hilfedatei (*.chm, *.hlp, *.htm)
	Demovideo zur Handhabung zeigen; öffnet eine Animation oder Videomitschnitt der Bildschirmarbeit (AVI, ScreenCam- oder FLASH-Film)
	entsprechendes Diagramm öffnen, das so geöffnete Diagramm kann über die Schließen-Schaltfläche des Diagramms wieder geschlossen werden
Beispiel 1	eine Diagrammvorlage laden
	Vorschau zur Diagrammvorlage
	zurück zur Startseite des Assistenten
	ein kleines Skript zu Arbeitsschritten anzeigen
	nächster Schritt (Arbeitsschritt);
	Verweis auf den Standardassistenten dieser Sicht;
blaue Schrift, blaue Symbole	sind mit Links hinterlegt und können per Mausklick aktiviert werden

Beispiele für Assistenten:





### 3.9.4 spezielle Hilfen zur Programmierung

#### Syntax

In Anhängigkeit des gewählten Profils kann jeweils aus dem Quellcode des Programms eine spezielle Hilfe aktiviert werden. Diese Hilfe wird erreicht über das Kontextmenü (rechte Maustaste) aus dem Quellcode.

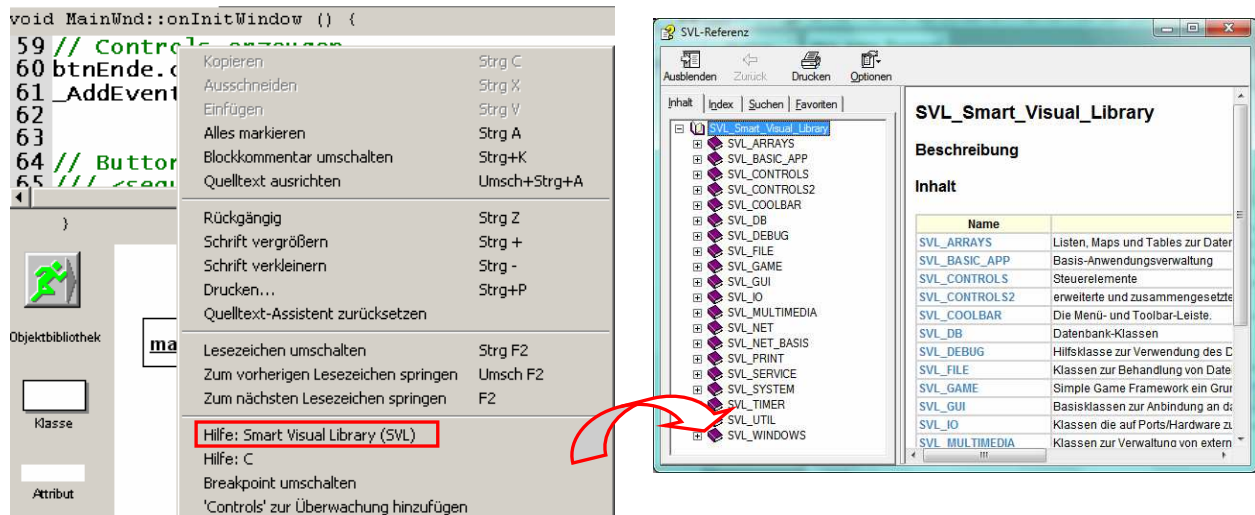


Abbildung 13: Hilfe im SVL-Programm

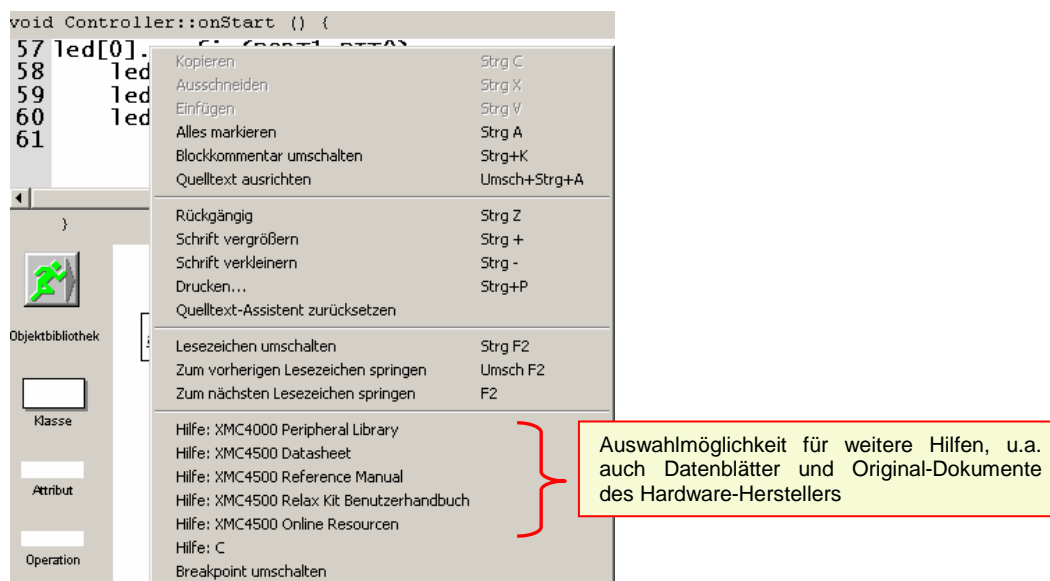


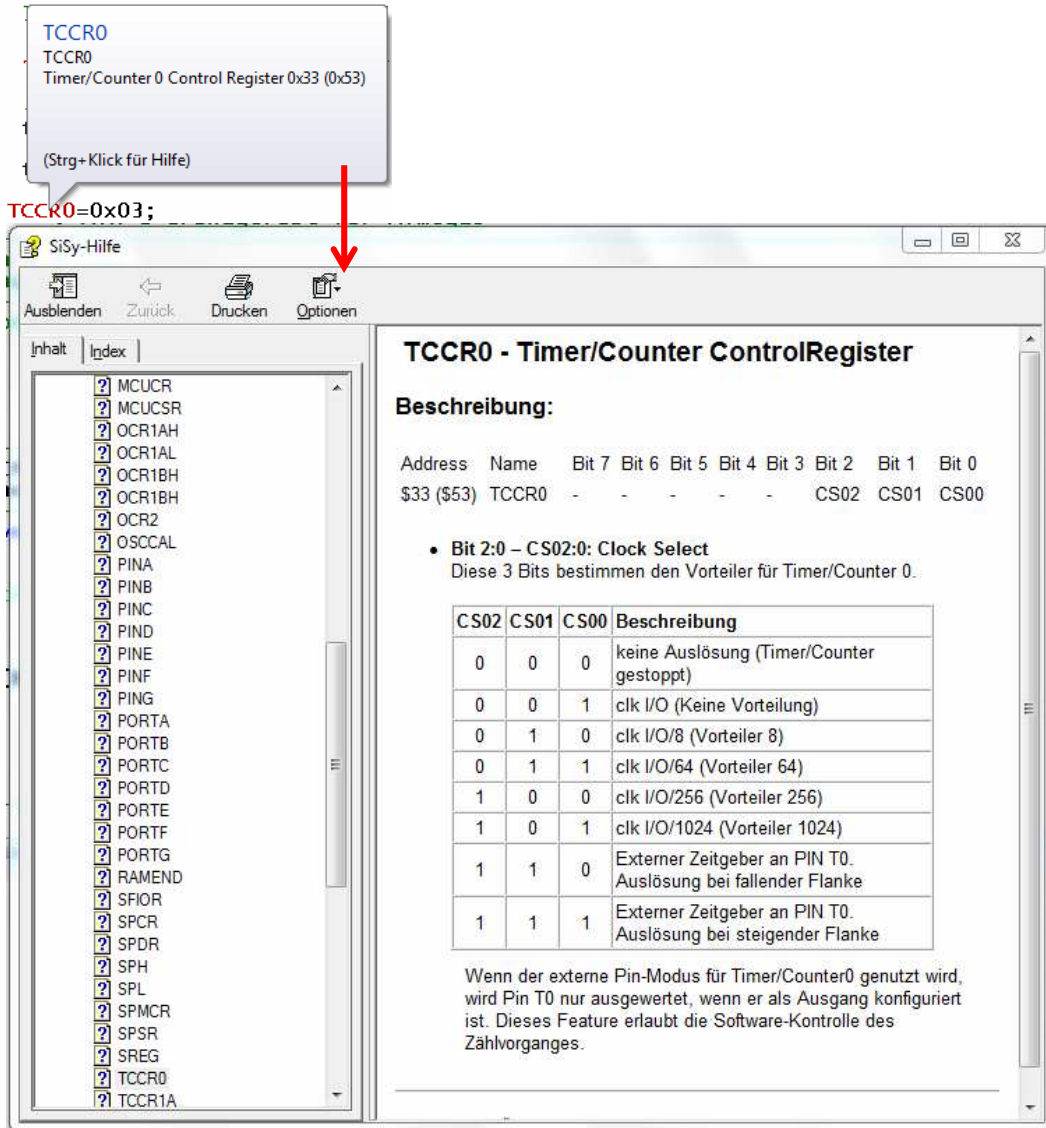
Abbildung 14: Beispiel verschiedener Hilfen für XMC-Controller

#### Online-Hilfe

Bei der Eingabe von Quellcode im dafür vorgesehenen Editorfenster werden reservierte Worte (Bezeichner, Schlüsselworte) der gewählten Programmiersprache durch verschiedenfarbiges Einfärben (Syntaxfarben) hervorgehoben. Zu den hervorgehobenen Bezeichnern existiert in der Regel auch eine kurze Online-Hilfe und eine ausführlichere Hilfe. Die Online-Hilfe ist ein Pop-up-Fenster, welches automatisch eingeblendet wird, wenn Sie mit dem Mauszeiger auf einen Befehl zeigen. In dem Pop-up ist eine kurze Hilfestellung zu dem Bezeichner eingeblendet. Steht eine Hilfe mit ausführlicheren Informationen zur Verfügung, wird diese in dem Pop-up (STRG+Klick für Hilfe) angezeigt.

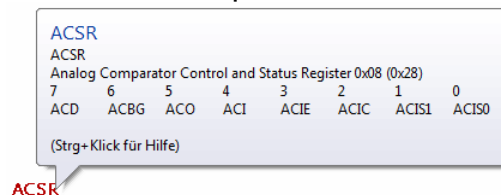
### Schlüsselwort-Hilfe

Bei der Eingabe von bekannten Registernamen wird die Bezeichnung des Registers und dessen Adresse eingeblendet. Beim Betätigen der Taste „STRG“ und gleichzeitigigem Klick auf das Register öffnet eine Hilfedatei mit dem entsprechenden Hilfethema zu dem eingegebenen Register.



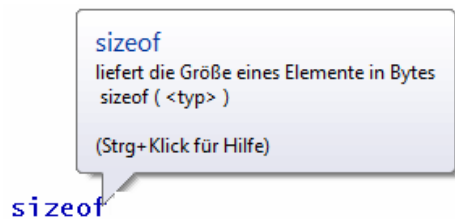
### Bit-Hilfe

Wenn für ein Register detaillierte Informationen zu der Bedeutung/Funktion der einzelnen Bits vorliegen, wird zusätzlich im Pop-up eine Kurzreferenz der Bits angezeigt. Beim Drücken der Taste „STRG“ sowie Klick auf das Register öffnet SiSy eine Hilfedatei mit dem entsprechenden ausführlichen Hilfethema.



## Befehlshilfe

Bei der Eingabe von Befehlen wird in der Regel die Bedeutung bzw. Funktion des Befehls und ein kurzes Syntaxbeispiel eingeblendet. Durch drücken der Taste „STRG“ sowie Klick auf den Befehl öffnet SiSy eine Hilfedatei mit dem entsprechenden ausführlichen Hilfethema.



## SiSy Code-Vervollständigung

Der Codegenerator ist eine integrierte Hilfe in SiSy. Vorteilhaft erweist sich diese Hilfe bei der Generierung des Quellcodes aus einem Programmablaufplan sowie bei der Arbeit mit Klassendiagrammen. Des Weiteren fungiert er als Assistent zum Erstellen von Assembler- und C-Codes für die Programmierung von Mikrocontrollern, was die fehlerhafte Codeeingabe minimiert.

Bei der *Programmierung* springt die Code-Vervollständigung nach der Eingabe von drei zusammenhängenden Buchstaben an. Aus der angezeigten Liste kann der gewünschte Befehl selektiert werden.

```

19 void initApplication()
20 {
21     SysTick_Config(SystemCoreClock/100);
22     // weitere Initialisierungen durchführen
23     // Takt für GPIOD an
24     RCC
25 }
26 int main()
27 {
28     Sys
29     ini
30     do{
31         + RCC_AHB1Periph_ClockCmd(long unsigned int , FunctionalState ) : void
32         + RCC_AHB1Periph_ClockLPMModeCmd(long unsigned int , FunctionalState ) : void
33         + RCC_AHB1Periph_ResetCmd(long unsigned int , FunctionalState ) : void
34         + RCC_AHB1Periph_BKPSRAM
35         + RCC_AHB1Periph_CCMDATARAMEN
36         + RCC_AHB1Periph_CRC
37         + RCC_AHB1Periph_DMA1
38         + RCC_AHB1Periph_DMA2
39         + RCC_AHB1Periph_ETH_MAC
40         + RCC_AHB1Periph_ETH_MAC_PTP
41     } while(1)
42 }

```

Bedingungen im *Programmablaufplan* haben spezielle Vorlagen. Diese sind so konstruiert, dass eine JA/NEIN Entscheidung erzeugt werden kann. Findet der Codegenerator das Schlüsselwort JA oder NEIN an einer der folgenden Verbindungen, setzt er diese in eine entsprechende Sprunganweisung um.

Im *UML Klassendiagramm* erzeugt der Codegenerator eine Instanz der Klasse und ruft die Main-Methode auf. Das Zustandsdiagramm mit Quellcodegenerierung erhalten Sie über ein spezielles Attribut im Klassendiagramm der UML. Dieses Zustandsattribut kann mit einem Zustandsdiagramm verfeinert werden. Dabei wird der Quellcode in spezielle Klassenmethoden generiert.

In den entsprechenden Kapiteln zur Programmierung, zum Programmablaufplan, zum Klassendiagramm und zum myAVR Code-Wizard wird explizit auf die Code-Vervollständigung eingegangen.

## 4 Ausgewählte Methoden/Profile zur Programmierung

In diesem Handbuch wird nicht auf die Grundlagen der Programmierung und Modellierung eingegangen. Detaillierte Handhabung und Methoden bieten Ihnen

- Seminare und Schulungen,  
eine Übersicht zu Inhalt, Terminen und Preisen sowie weiteren Informationen finden Sie unter [www.myMCU.de](http://www.myMCU.de) → Seminare
- Tutorials,  
diese sind aus der installierten SiSy-Ausgabe direkt aufrufbar
- Schnelleinstiege,  
an Hand von spezifischen kleinen Beispielen wird die Handhabung von SiSy aufgezeigt; eine Liste der Schnelleinstiege mit Link finden Sie unter [www.sisy.de](http://www.sisy.de) → Schnelleinstiege
- Handbücher zur Programmierung,  
sie beinhalten theoretische Grundlagen mit praktischen nachvollziehbaren Beispielen zum jeweiligen Fachgebiet  
[www.myMCU.de](http://www.myMCU.de) → Shop → Suchbegriff „Literatur“

In den folgenden Kapiteln wird exemplarisch veranschaulicht, welche Möglichkeiten und Methoden SiSy für die Modellierung und Programmierung bietet.

### 4.1 Programmierung im Klassendiagramm

Mit objektorientierten Programmiersprachen hat der Entwickler mächtige Sprachmittel zur Realisierung komplexer Systeme. C++ ist eine weit verbreitete objektorientierte Standardsprache. Als Visualisierungsmittel objektorientierter Programme gilt die international standardisierte Beschreibungssprache UML (Unified Modeling Language). Die Abbildung zeigt Ihnen eine Kurzübersicht der Modellierungselemente des UML Klassendiagramms.

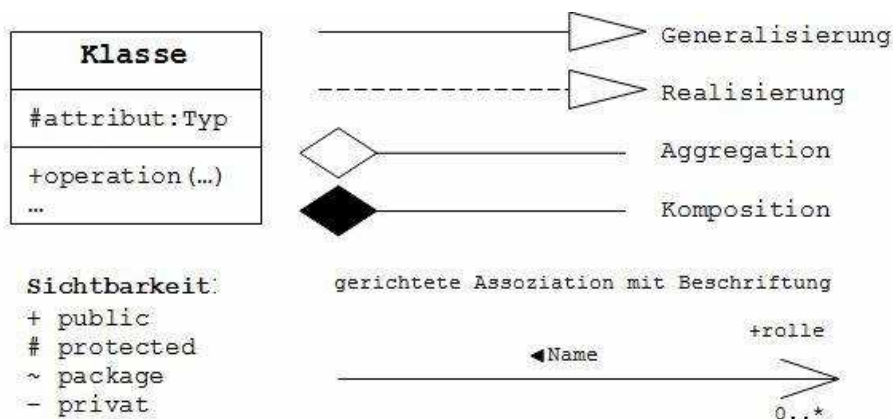


Abbildung 15: Kurzübersicht Elemente des UML-Klassendiagramms

SiSy bietet dem Entwickler das UML Klassendiagramm mit Codegenerierung für C++, AVR C++ und ARM C++.

### 4.1.1 PC-Programm

#### Konsolenprogramm

Konsolenprogramme können in SiSy mit dem UML-Klassendiagramm erstellt werden. Zum Einstieg empfiehlt sich nach dem Start von SiSy das Profil „UML Objektorientierte Techniken“. SiSy LibStore bietet verschiedene Vorlagen an.



Abbildung 16: Profil „UML Objektorientierte Techniken“

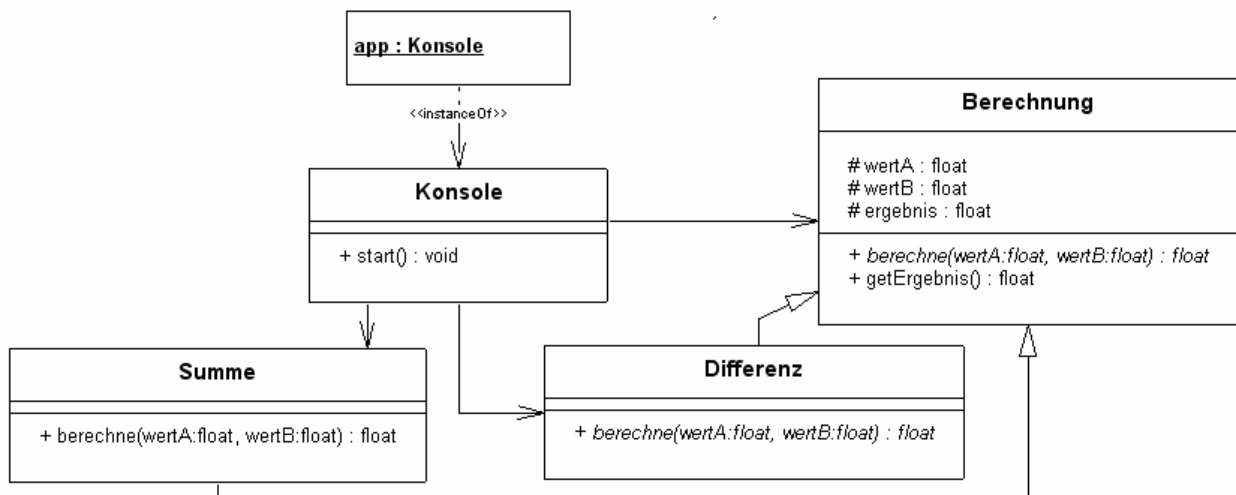


Abbildung 17: vollständiges Klassendiagramm für ein Beispielprogramm zu PC-Konsole

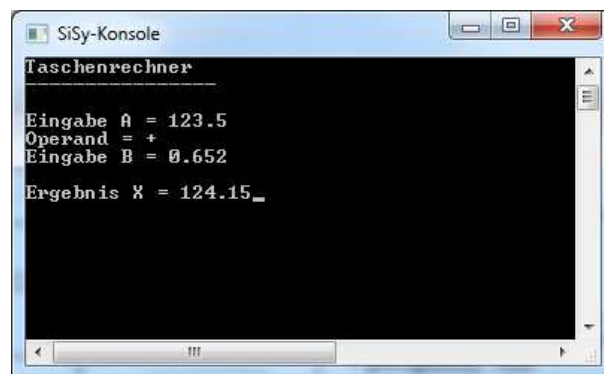


Abbildung 18: mögliches Ergebnis zur PC-Konsole

## Windows-Programmierung mit SVL

Die SVL (Smart Visual Library) ist eine kleine, didaktische Klassenbibliothek für die Programmiersprache C++. Sie unterstützt die einfache Erstellung von Windowsanwendungen speziell mit dem Klassendiagramm in der UML-Entwicklungsumgebung SiSy. Nach dem Start von SiSy ist das Profil „UML mit SVL (Smart Visual Library)“ zu empfehlen und aus dem SiSy LibStore die Vorlage „SVL-Framework mit Beispielen“.

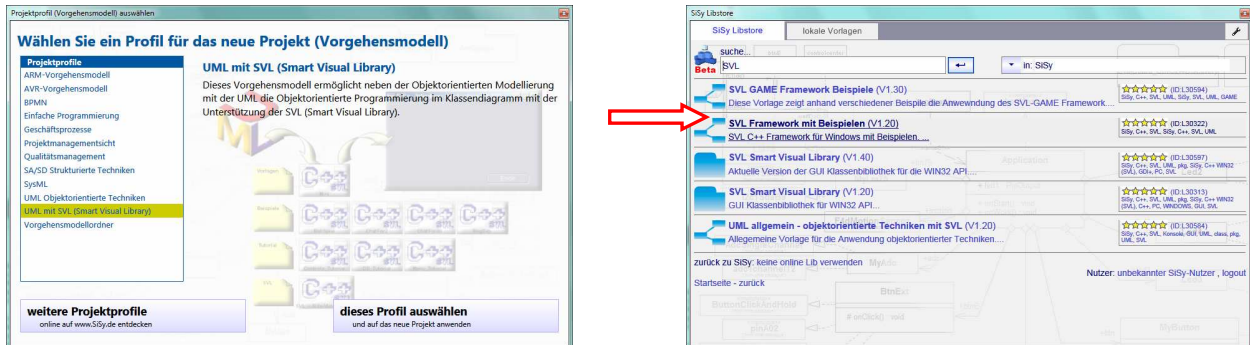


Abbildung 19: Auswahl SVL-Profil und Vorlage aus SiSy LibStore

Nach dem Import der Diagrammvorlage steht Ihnen eine Auswahl an Tutorials und Beispielen zur Verfügung. Das Klassendiagramm „Mein\_erstes\_Programm“ bietet Ihnen ein Grundgerüst für die weitere Arbeit an. In der folgenden Abbildung sehen Sie das verfeinerte Klassendiagramm, welches das Grundgerüst für Ihr Programm sein kann.

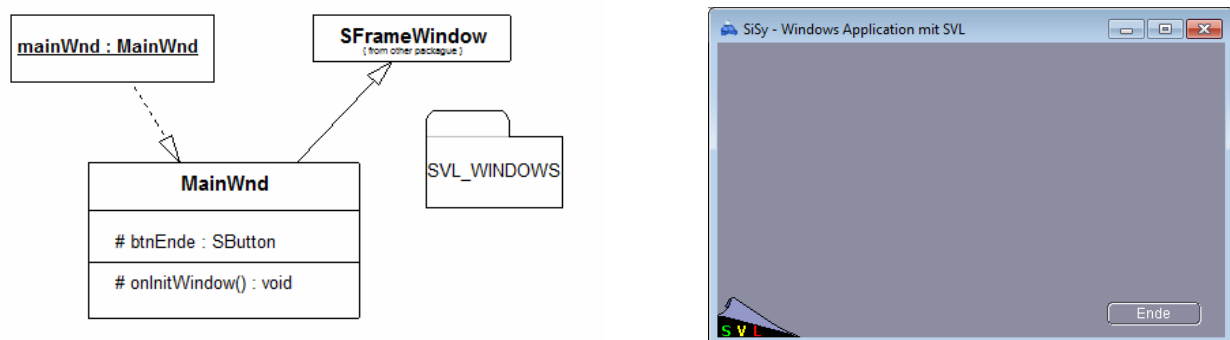
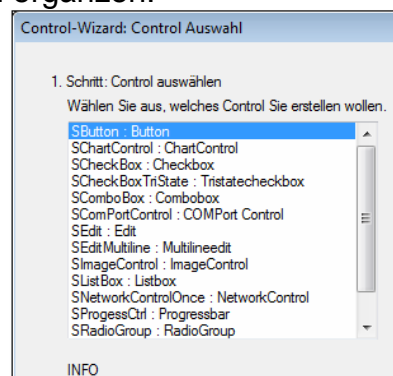
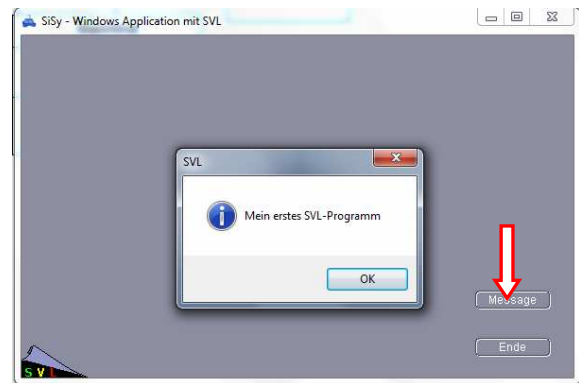
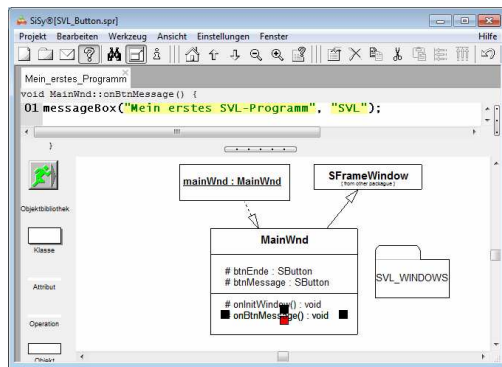


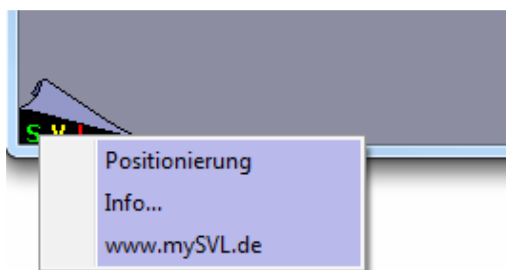
Abbildung 20: SVL "Mein\_erstes\_Programm"; Klassendiagramm und gestartetes Programm

Der Control-Wizard bietet Ihnen vielfältige Möglichkeiten zur Erweiterung des Programms, z.B. für weitere Schaltflächen, Eingabefelder oder Dialogfenster. Der entsprechende Quellcode ist noch zu ergänzen.





Für eine andere Positionierung der Schaltflächen klicken Sie in der linken unteren Ecke auf das SVL-Logo und wählen „Positionierung“. Nun können Sie per Drag&Drop die einzelnen Elemente des Fensters neu positionieren.



### SVL-Tutorial

Im „SVL, Smart Visual Library – Tutorial“ finden Sie eine schrittweise Einführung von den Grundlagen bis zur komplexen Anwendungsentwicklung. Alle Erläuterungen und Beispiele erfolgen mit der SVL und basieren auf UML-Klassendiagrammen.

Die ersten Kapitel sind frei zugänglich; Kapitel zu Projekten sind nur für Nutzer mit der entsprechenden SiSy-Lizenz verfügbar.

Im Folgenden eine Auswahl des Inhalts der Tutorials (Stand 06/2016)

- SVL, Smart Visual Library
  - Basiskonzepte und UML Grundlagen
  - Einfache Steuerelemente
  - Entwicklerwerkzeuge
  - Arbeit mit dem Sequenzdiagramm
  - Programmieren mit dem Zustandsdiagramm
  - Templates

### 4.1.2 Mikrocontrollerprogrammierung in C++

SiSy unterstützt die Programmierung von 8-Bit Mikrocontroller (AVR) und 32-Bit Mikrocontroller (ARM). In den online Tutorials

- AVR C++ Tutorial
- STM32 Tutorial für C/C++ und UML
- XMC Tutorial für C/C++ und UML

ist die Programmierung dieser Prozessorfamilien ausführlich beschrieben. Integrierte Beispiele innerhalb der Tutorials unterstützen die theoretischen Erläuterungen.

Für die Programmierung von Mikrocontrollern in C++ sind in SiSy PEC-Bibliotheken integriert. PEC (Portable Embedded Classes) ist eine portable Klassenbibliothek für eingebettete Systeme verschiedener Mikrocontroller.

Des Weiteren kennt SiSy generische Strukturen, die Templates. Templates sind Vorlagen, die zum Übersetzungszeitpunkt etwas Konkretes erstellen. Die Generierung der konkreten Klassen übernimmt dabei der Codegenerator. Die in den Templates vorgefertigten Struktur- und Verhaltensmerkmale werden in die Zielklasse generiert.

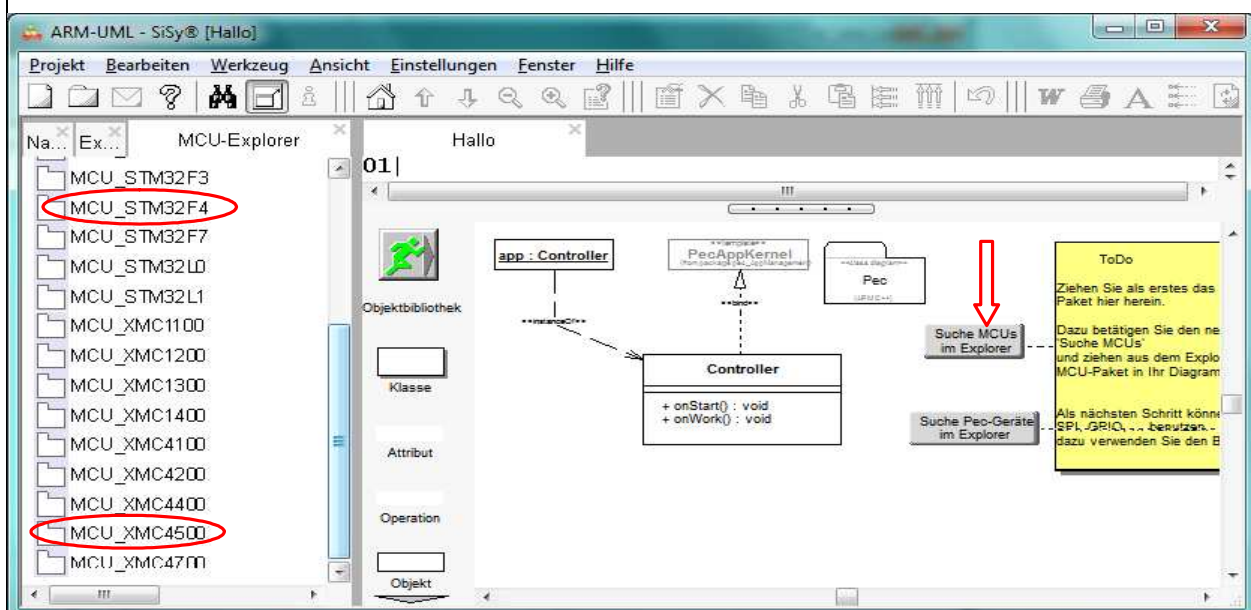
Nach dem Start von SiSy wird folgende Vorgehensweise empfohlen:

	AVR-Mikrocontroller	STM32-Mikrocontroller	XMC-Mikrocontroller
Profil	AVR-Vorgehensmodell	ARM-Vorgehensmodell	
Vorlage aus SiSy LibStore	PEC Framework – Portable Embedded Classes		
Programmiersprache	AVR C++	ARM C++	

Klassendiagramm öffnen

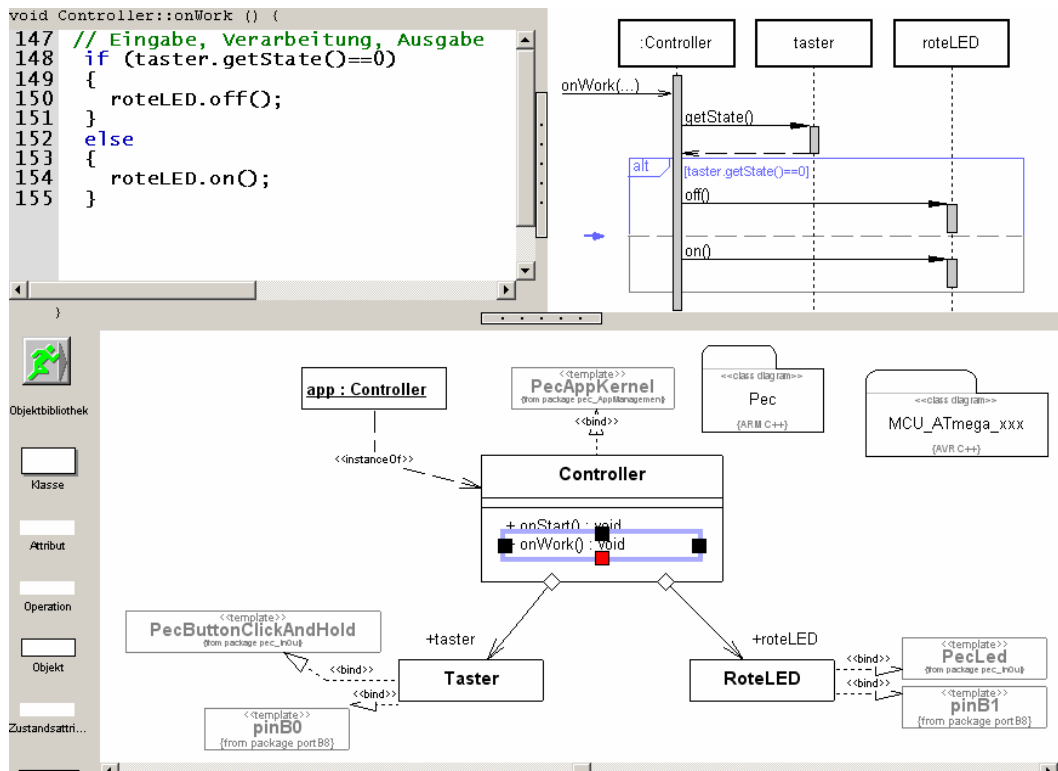
Vorlage aus SiSy LibStore	Application Grundgerüst für PEC Anwendungen (XMC,STM32, AVR)
---------------------------	--

Aus dem MCU-Explorer das controllerspezifische Paket für die verwendete Hardware auswählen und in das Diagramm ziehen.



Klassendiagramm vervollständigen und Quellcode erfassen



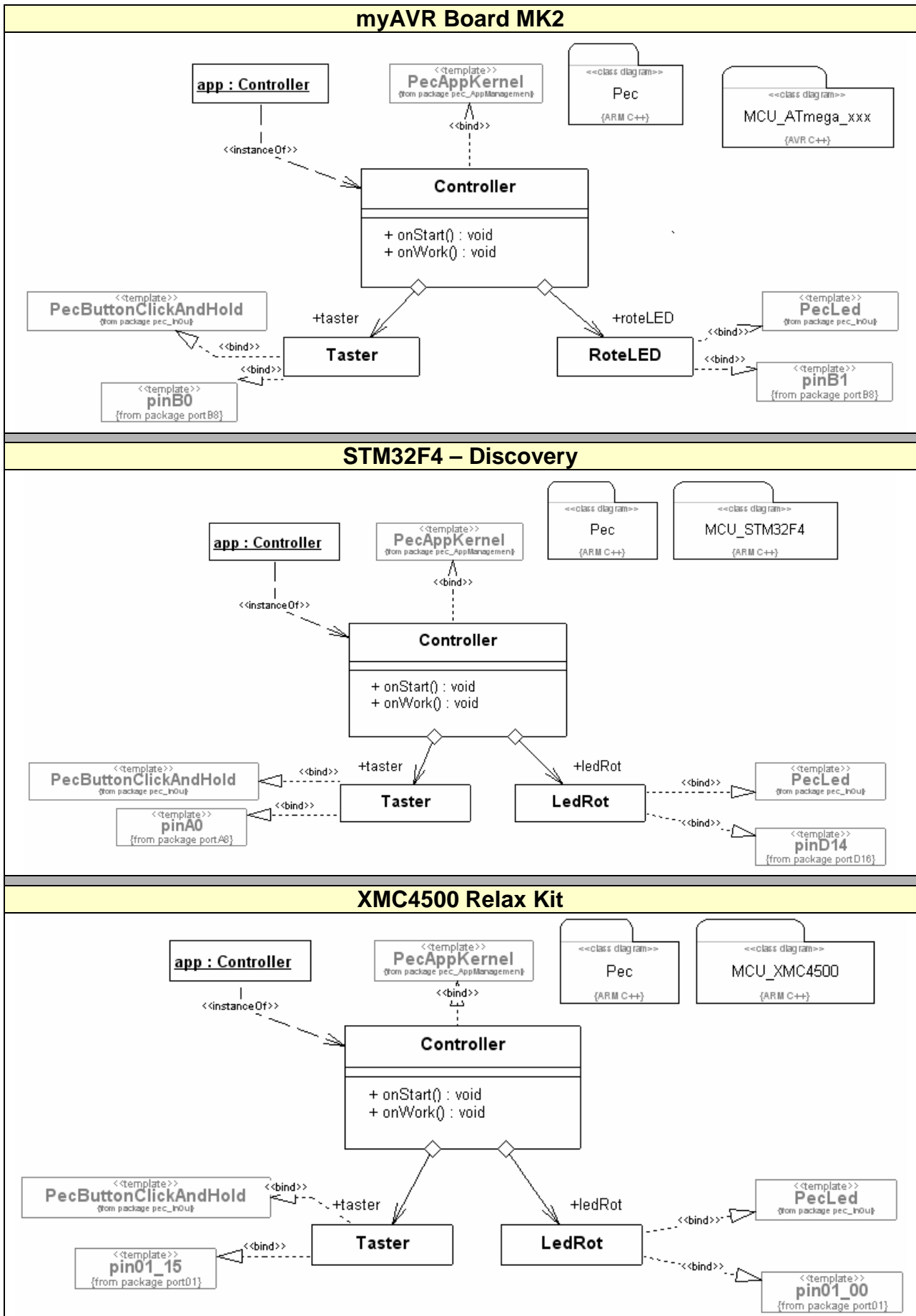


Während der Eingabe des Quellcodes im Editorfenster (Eingabefenster) wird im benachbarten Fenster das Sequenzdiagramm erstellt.

**Beispielanwendung:**

Auf dem Entwicklerboard soll bei Tastendruck eine LED leuchten. In der folgenden Darstellung sind die Klassendiagramme für drei verschiedene Entwicklerboards aufgezeigt:

- myAVR Board MK2 mit einem ATmega8 für die AVR-Programmierung
- STM32F4-Discovery für die STM32-Programmierung (ARM)
- XMC4500 Relax Lite Kit für die XMC-Programmierung (ARM)



## Tutorial

Zur objektorientierten Programmierung von Mikrocontrollern im UML Klassendiagramm Programmierung finden Sie in Tutorials schrittweise Einführungen, von den Grundlagen bis zur komplexen Anwendungsentwicklung. Alle Erläuterungen und Beispiele mit dem Klassendiagramm nutzen die PEC-Bibliotheken.

Die ersten Kapitel sind frei zugänglich; Kapitel zu Projekten sind nur für Nutzer mit der entsprechenden SiSy-Lizenz verfügbar.

Im Folgenden eine Auswahl des Inhalts der Tutorials (Stand 06/2016)

- AVR UML Tutorial
  - theoretische Grundlagen, wie Basiskonzepte, objektorientierte Programmiersprachen, UML Grundlagen
  - Grundstruktur einer Mikrocontrolleranwendung in UML
  - PEC Bibliotheken in SiSy
  - mit Klassen arbeiten
  - UART Kommunikation mit dem PC
  - Analogdaten erfassen
  - Timer, Interrupt
  - mit dem Zustandsdiagramm in SiSy programmieren
- STM32 Tutorial für C/C++ und UML
  - theoretische Grundlagen, wie Basiskonzepte, objektorientierte Programmiersprachen, UML Grundlagen
  - Grundstruktur einer objektorientierten Anwendung
  - mit Klassen arbeiten
  - SystemTick und Templates
  - Kommunikation mit dem PC
  - Analogdaten erfassen
  - Timer, Interrupt
  - Beschleunigungssensor nutzen
  - mit dem Zustandsdiagramm in SiSy programmieren
- XMC Tutorial für C/C++ und UML
  - theoretische Grundlagen, wie Basiskonzepte, objektorientierte Programmiersprachen, UML Grundlagen
  - Grundstruktur einer objektorientierten Anwendung
  - mit Klassen arbeiten
  - SystemTick und Templates
  - Kommunikation mit dem PC
  - Analogdaten erfassen
  - Timer und Interrupt
  - mit dem Zustandsdiagramm in SiSy programmieren

## 4.2 Programmentwicklung aus einem Struktogramm

### 4.2.1 Grundlagen

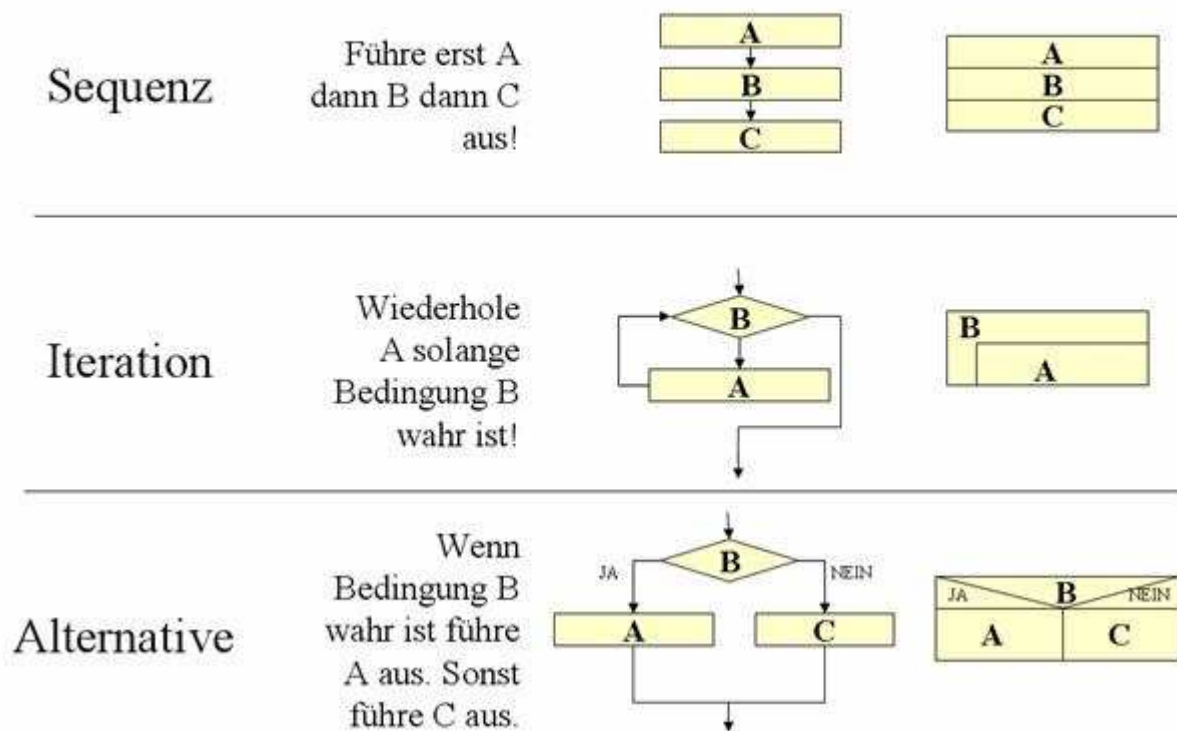
Struktogramme (SG) oder auch Nassi-Shneiderman-Diagramme sind ein Entwurfsmittel der strukturierten Programmierung. Strukturiert meint in dem Fall, dass zur Realisierung eines Algorithmus auf das Verwenden von Sprunganweisungen (Goto, Jump) verzichtet wird. Struktogramme werden als Darstellungsmittel für strukturierte Sprachen wie C oder PASCAL verwendet, da hier im Gegensatz zu Assembler in der Regel auf Sprunganweisungen verzichtet wird.

Für das Formulieren eines Algorithmus stehen dem Entwickler drei normierte Grundstrukturen zur Verfügung:

- Sequenz (Folge von Anweisungen),
- Alternative (Auswahl bzw. bedingte Anweisung),
- Iteration (Schleife, wiederholte Anweisung).

## Grundelemente

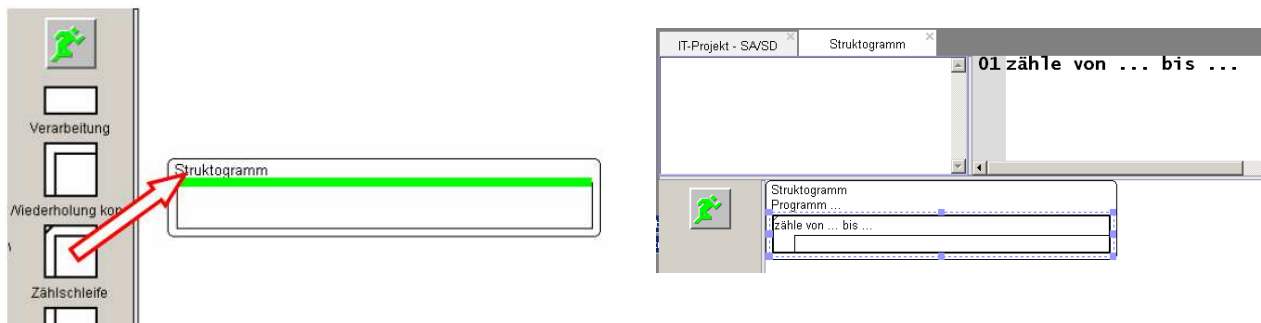
(Text, Programmablaufplan, Struktogramm)



### Struktogramm in SiSy

In SiSy sind die Funktionen zum Erstellen von Struktogrammen für PC- und AVR-Programme integriert. Die Arbeit mit einem Struktogramm wird dabei optimal unterstützt. Beim Einfügen von Struktogramm-Elementen in das Struktogramm sind die möglichen Positionen im Struktogramm mit einer grünen Linie unterlegt. Beim Bewegen des Mauszeigers an die gewünschte Stelle, wird die grüne Linie zur Kontrolle breiter.

Das eingefügte Element enthält einen elementspezifischen Text, z.B. „führe ... aus“, „solange wie ...“, „zähle von ... bis ...“, der mit dem entsprechenden Quellcode zu ersetzen ist. Für das Editieren kann das betreffende Element per Doppelklick selektiert werden oder das Editieren erfolgt in dem separaten Editorfenster.



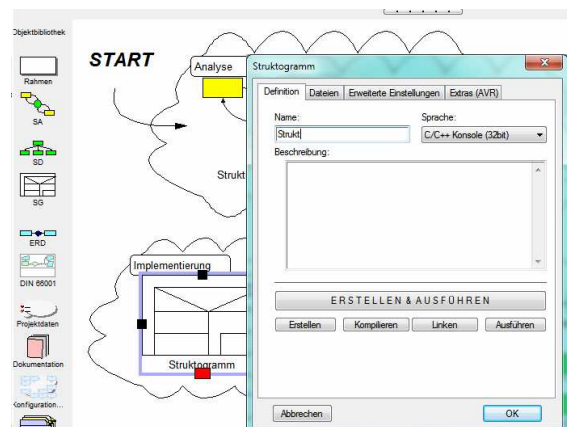
**Hinweis**

Das Struktogramm kann je SiSy-Projekt nur ein Hauptprogramm verwalten. Es ist also nötig, für jedes neue Struktogramm-Projekt auch ein neues SiSy-Projekt zu erzeugen.

**4.2.2 Konsolenprogramme**

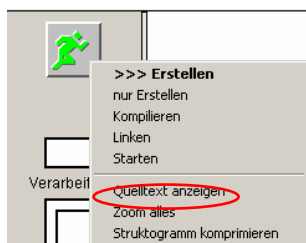
Das Profil „SA/SD Strukturierte Techniken“ bildet die Basis für PC-Programme, die mit Struktogrammen erstellt werden. Für die weitere Arbeit bietet Ihnen SiSy LibStore die Vorlage „IT-Projekt – SA/SD“.

Im Definieren-Dialog des Objektes „Struktogramm“ wird die Programmiersprache ausgewählt, hier „C/C++ Konsole“.



Kontrollieren Sie unter der Registerkarte „Erweiterte Einstellungen“ die Eintragungen der *include*-Dateien (`#include <stdio.h>`, `#include <conio.h>`). Falls benötigt, können hier weitere include-Dateien ergänzt werden.

Von dem fertig erstellten Struktogramm kann der Quellcode angezeigt werden. Die Funktion wird über das Aktionsmenü gestartet.



```

01 ////////////////////////////////////////////////////////////////////
02 //
03 // Diese Quellcodedatei wurde automatisch erzeugt.
04 // SiSy SG CodeGenerierung
05 //
06 ////////////////////////////////////////////////////////////////////
07 //<ObjektNummer>12</ObjektNummer> GoToSiSy:d:12|o:12
08 //
09 /*
10 */
11 */
12
13 #include <stdio.h>
14 #include <conio.h>
15
16
17 int main()
18 {
19     /*Programm / Funktion ...*/
20     for(int i = 0; i < 5; i++)
21     {

```

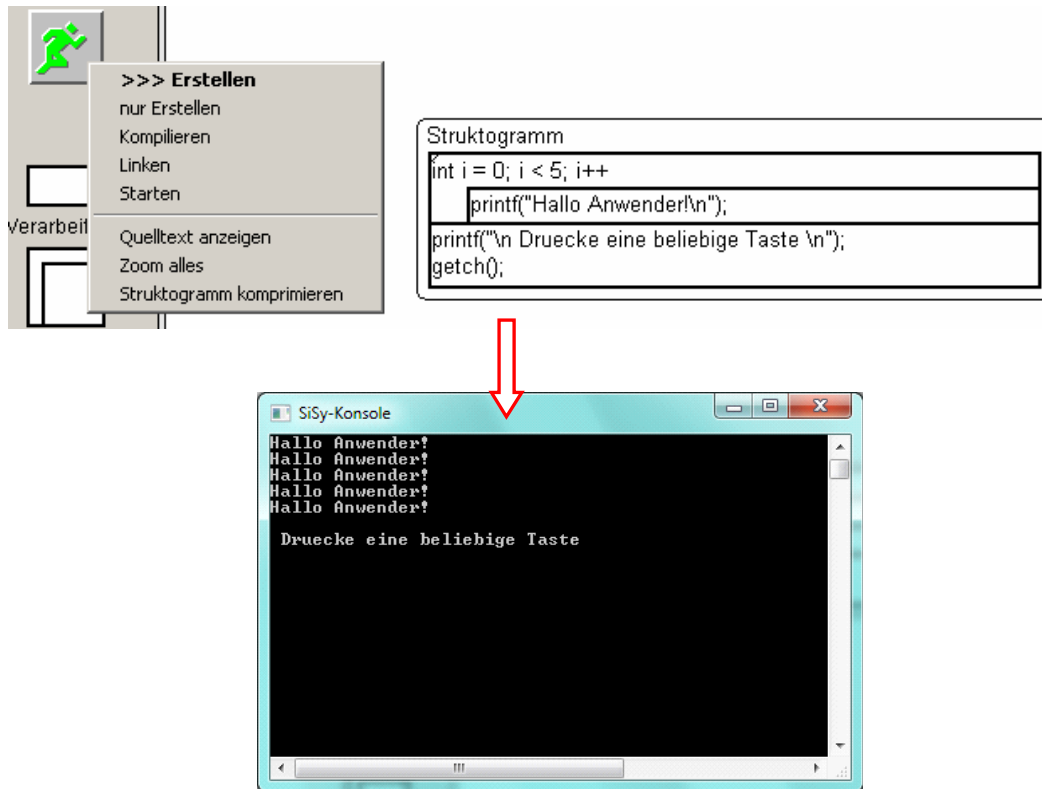
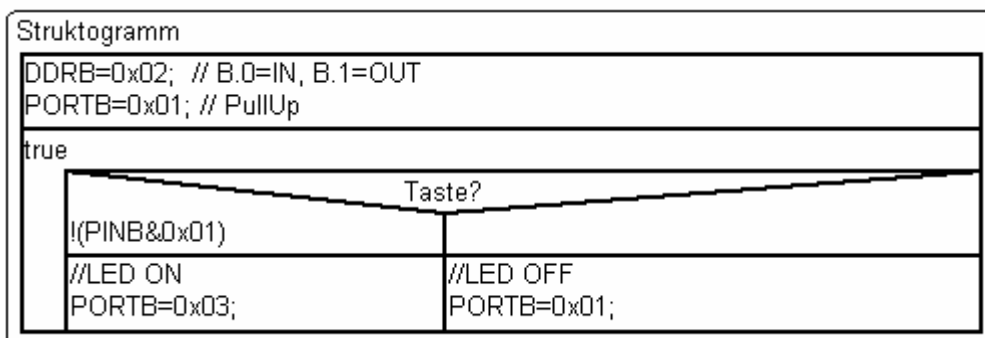
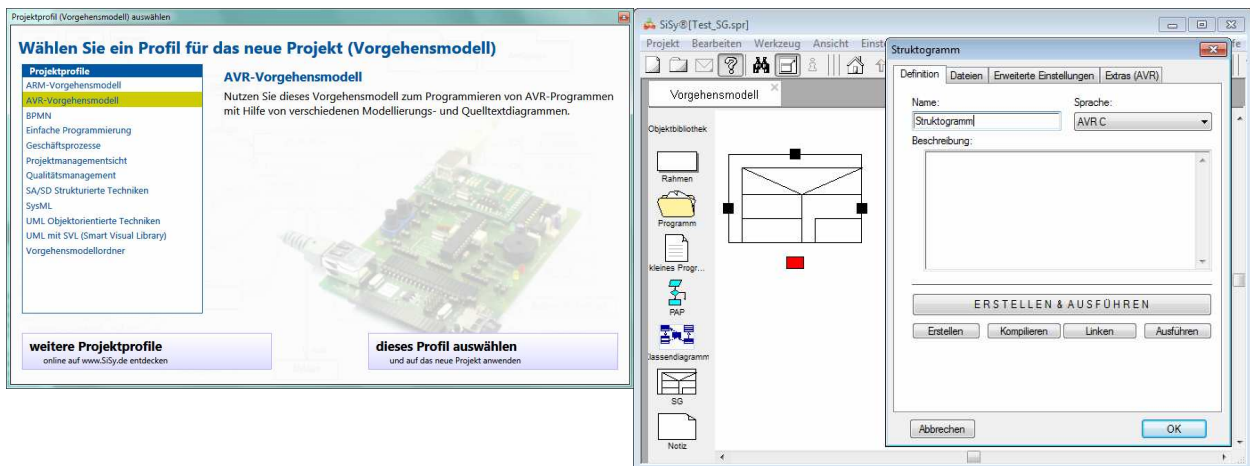


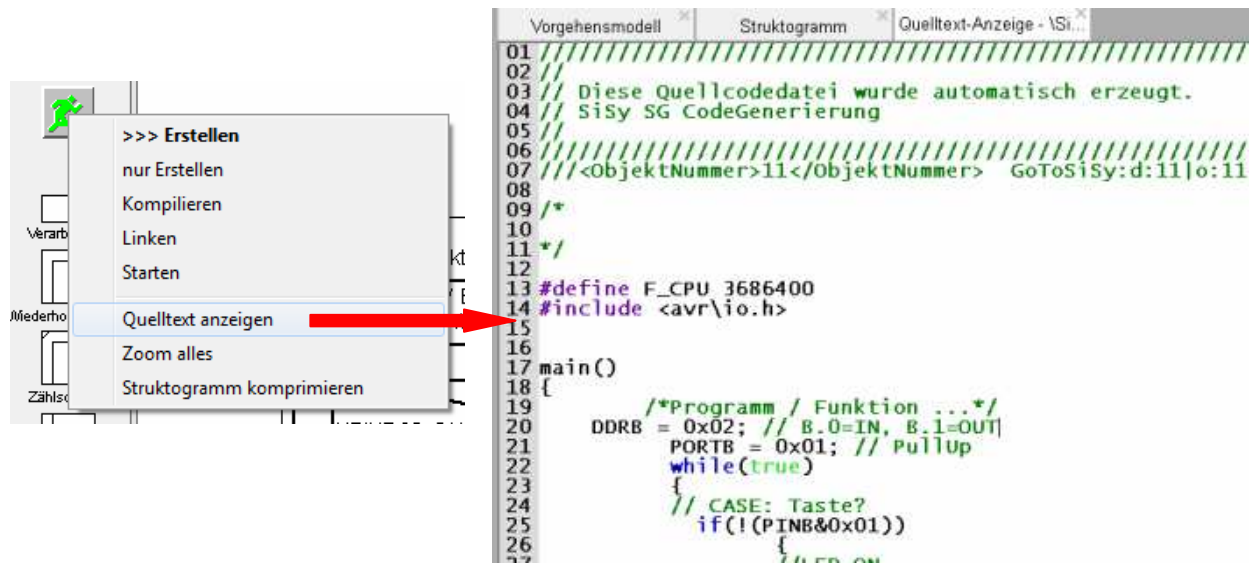
Abbildung 21: Beispiel für Struktogramm und Textausgabe auf Konsole

### 4.2.3 AVR Mikrocontroller-Programme

Das Profil „AVR Vorgehensmodell“ ist für AVR Mikrocontroller-Programme geeignet, die mit einem Struktogramm erstellt werden. SiSy LibStore bietet Ihnen dafür keine Vorlage an, deshalb muß ein Objekt vom Typ „Struktogramm“ per Drag&Drop aus der Objektbibliothek in das leere Diagramm gezogen werden. Im Definieren-Dialog des Struktogramms wird die Programmiersprache „AVR C“ gewählt.



Von dem fertig erstellten Struktogramm können Sie sich den Quellcode anzeigen lassen. Die Funktion wird über das Aktionsmenü erreicht.



Mit dem Befehl „>>> Erstellen“ wird das Programm auf den Controller gebrannt. Im myAVR ProgTool wird das Protokoll angezeigt. Für den Programmtest müssen Sie auf dem Board die Verbindungen stecken.

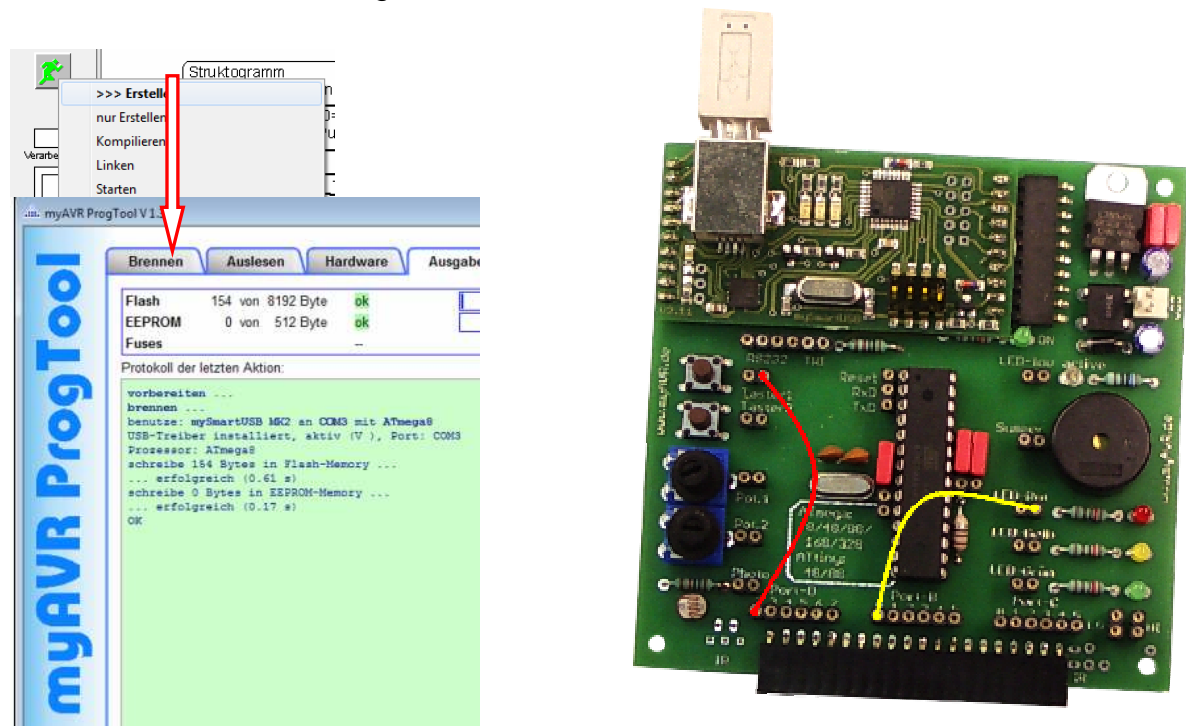


Abbildung 22: Schaltung für das Programmbeispiel "Struktogramm"

## 4.3 Entwicklung Programmablaufplan für AVR Programme

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.

SiSy unterstützt die Programmentwicklung aus einem PAP für AVR Mikrocontroller.

### 4.3.1 Einfache Programmentwicklung aus einem PAP

Zur Modellierung eines Programmablaufplanes wird das Profil „AVR Vorgehensmodell“ benötigt, mit der anschließenden Grundeinstellung für die verwendete Hardware.

SiSy LibStore bietet keine Vorlagen an. Die Modellierung erfolgt in einem leeren Diagramm.

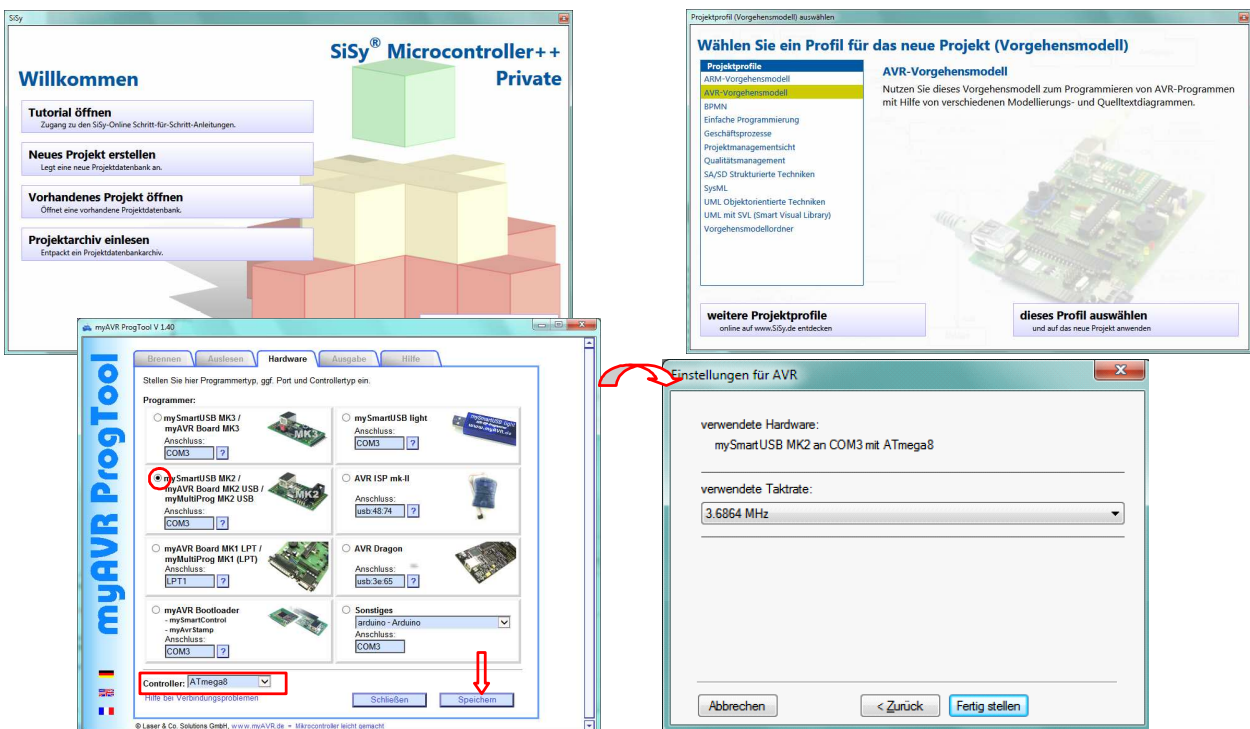


Abbildung 23: Projekt für PAP anlegen und Hardwarekonfiguration einstellen

Ziehen Sie aus der Objektbibliothek ein Objekt vom Typ „PAP“ in das leere Diagramm und öffnen Sie dieses. Dabei bietet SiSy LibStore die Vorlage „Grundgerüst PAP AVR Assembler“ für die weitere Arbeit an.

Für die Abbildung der Programmlogik im PAP muss die Vorlage mit den fehlenden Objekten ergänzt werden. Des Weiteren sind die Objekte durch gerichtete Verbindungen (Kanten) in der Reihenfolge ihrer Abarbeitung zu verbinden.

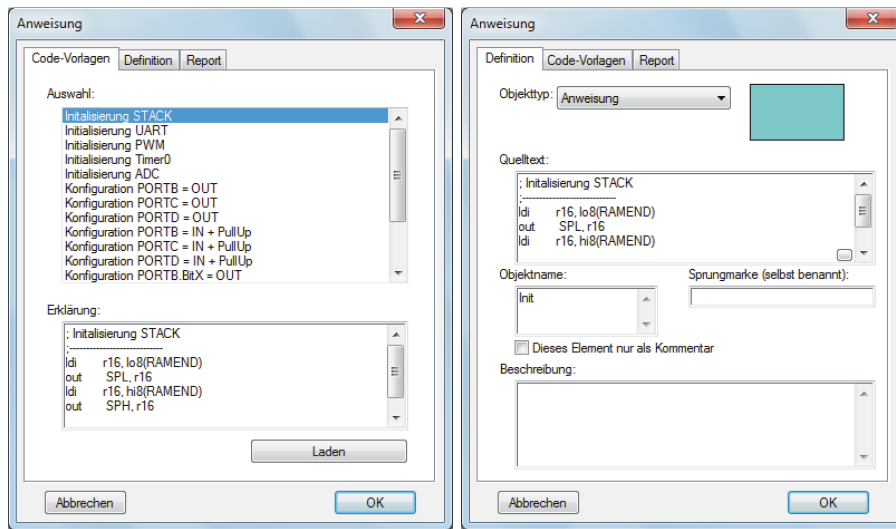
Verbindungen zwischen den Objekten können über den rot markierten Verteiler von selektierten Objekten hergestellt werden. Für das Einfügen von Objekten zwischen bestehenden Verbindungen ist zuvor diese Verbindung zu löschen.

Die einzelnen Elemente des PAP müssen mit den entsprechenden Assembleranweisungen versehen werden. Dafür gibt es folgende Möglichkeiten:

- SiSy bietet beim ersten Definieren eines jeden Objektes typische Code-Vorlagen an, die über die Schaltfläche „Laden“ dem Objekt zugewiesen werden können. Beim Beenden des Definieren-Dialogs mit „OK“, wird die Auswahl im Objekt gespeichert. Beim nächsten Aufruf dieses Dialoges erscheinen die Code-Vorlagen



nicht mehr; das Objekt kann ganz normal bearbeitet werden. In den folgenden Abbildungen sind beide Ausführungen des Dialoges „Definieren“ zu sehen.



- Die zweite Möglichkeit besteht beim Selektieren von Elementen und Einträgen des Quellcodes im Editorfenster.

### Besonderheiten

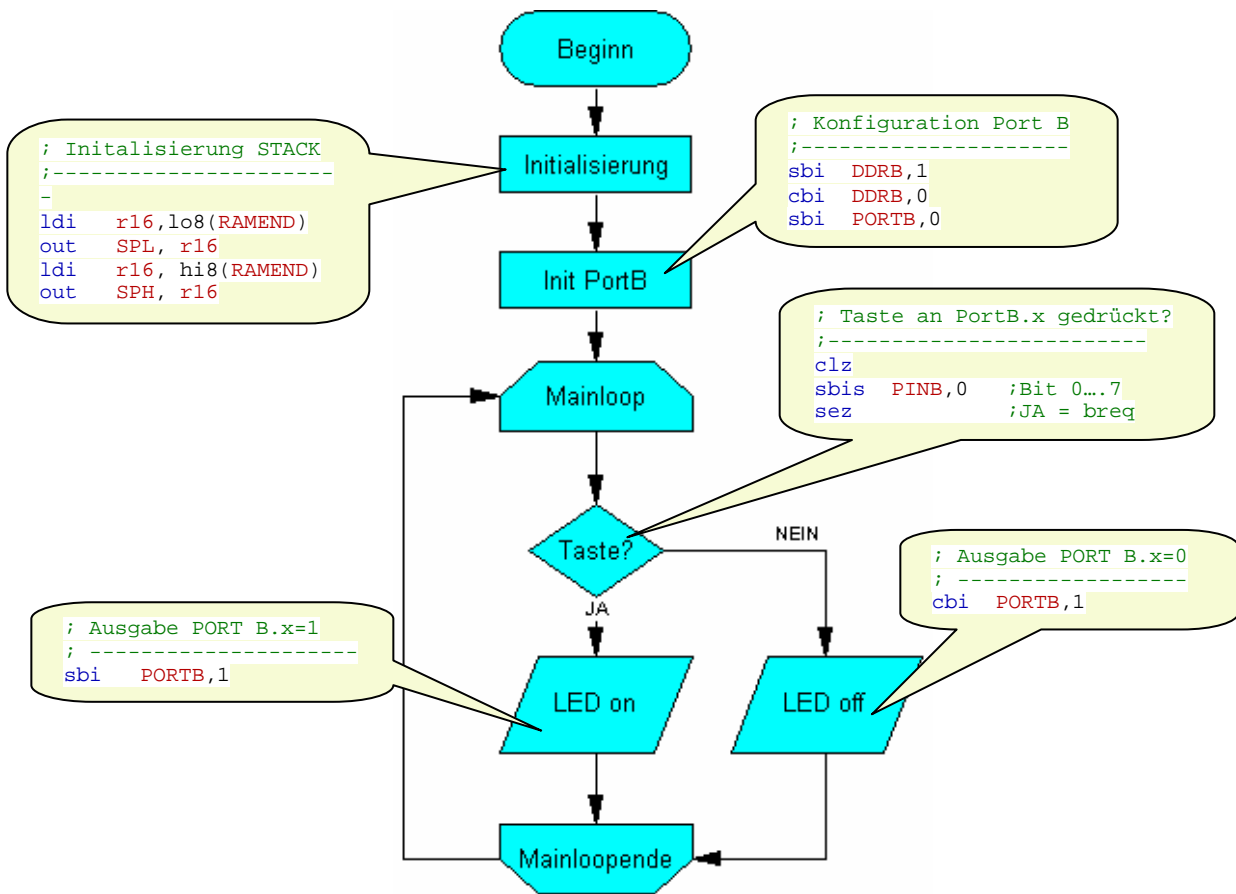
Bedingungen haben spezielle Vorlagen, die eine Codegenerierung bei übersichtlichem Programmablaufplan vereinfachen.

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 kann statt dieser Schlüsselworte auch der Sprungbefehl direkt an eine Kante geschrieben werden (*breq*, *brne*, *brge*, *brlo*, usw.)

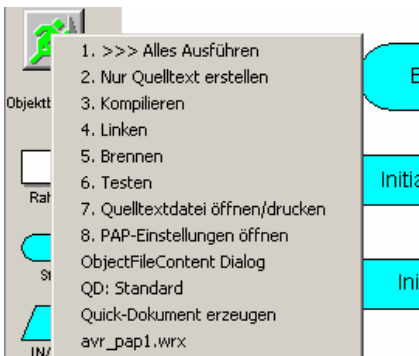
Das Element „IN/OUT“ verfügt auch über spezifische Vorlagen. Diese sind gegebenenfalls mit zu ergänzen. Dazu sind spitze Klammern als Platzhalter in den Vorlagen eingefügt.



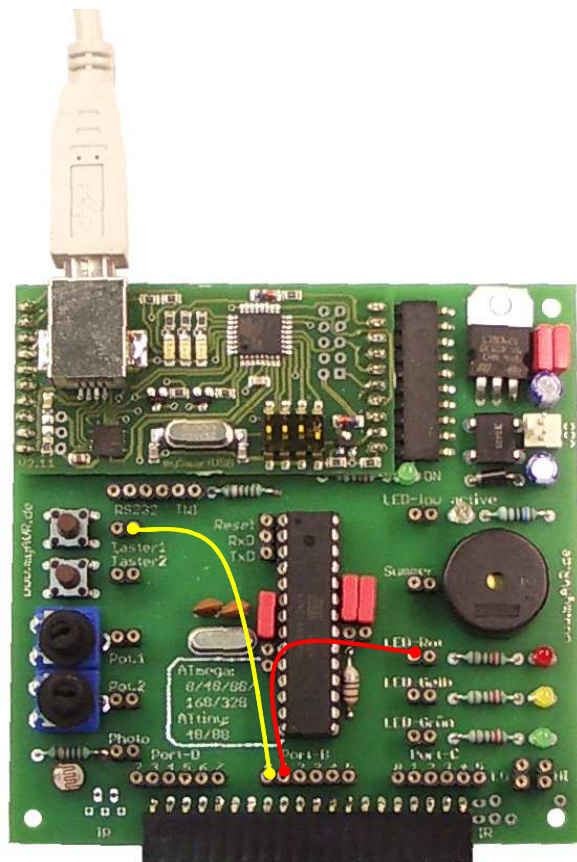
In der folgenden Abbildung ist ein PAP dargestellt für die Funktion, auf dem myAVR Board MK2 bei Tastendruck eine LED einzuschalten.



Kompilieren, Linken und Brennen bzw. „>>>Alles Ausführen“ werden aus dem Aktionsmenü gestartet. Die jeweiligen Aktionen werden im Ausgabefenster angezeigt.



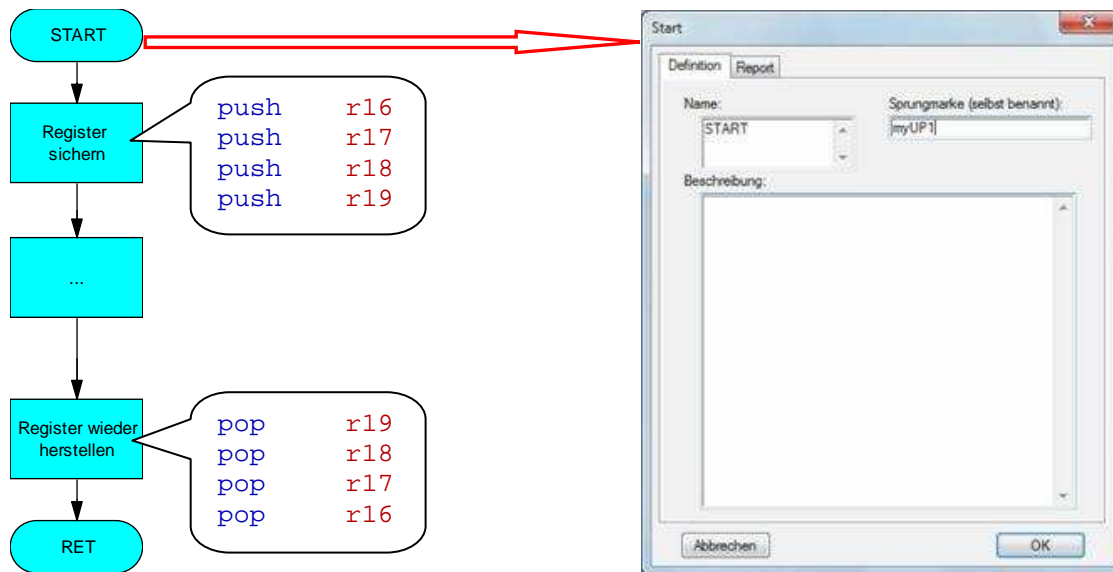
Für den Test der Anwendung müssen auf dem Board noch die entsprechenden Verbindungen hergestellt werden.



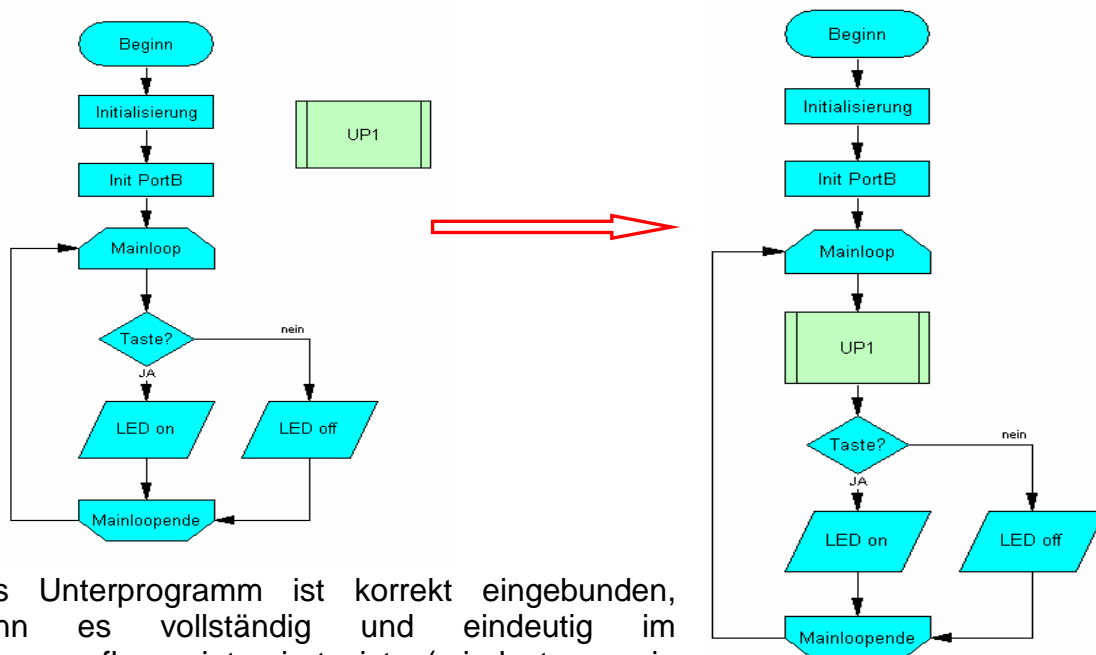
### 4.3.2 Unterprogrammtechnik im PAP

Unterprogramme sind ein wichtiges Gestaltungsmittel für übersichtliche Mikrocontrollerprogramme. Sie werden für in sich abgeschlossene Aufgaben (Verarbeitungsschritte) benutzt, die auch mehrfach im Gesamtprogramm genutzt werden können.

Legen Sie im aktuellen Diagramm ein Objekt vom Typ „Unterprogramm“ an. Die Funktionalität des Unterprogramms wird in einem gesonderten Programmablaufplan für das Unterprogramm entworfen. Dazu ist das „Unterprogramm“ zu öffnen. SiSy LibStore enthält eine Vorlage für die Grundstruktur von Unterprogrammen. Auf dem Objekt „START“ können Sie eine benutzerdefinierte Sprungmarke festlegen, die durch den Codegenerator erstellt und verwendet werden soll. Die Vorlage müssen Sie entsprechend der vorgesehen Logik ändern.



Das Unterprogrammssymbol muss zum Aufruf an der entsprechenden Position im Programmablaufplan eingefügt werden. Der Codegenerator erzeugt dann einen Unterprogrammaufruf und den Code für das Unterprogramm selbst.



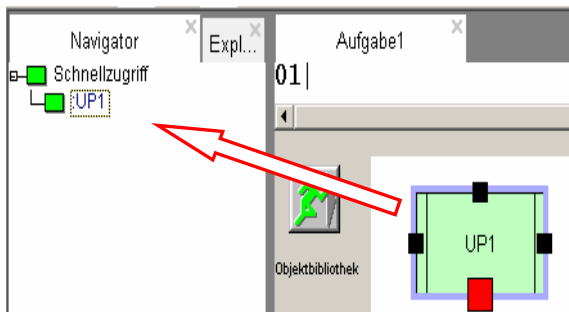
Das Unterprogramm ist korrekt eingebunden, wenn es vollständig und eindeutig im Programmfluss integriert ist (mindestens ein eingehender Pfeil und genau ein ausgehender Pfeil)

### Unterprogramme mehrmals benutzen

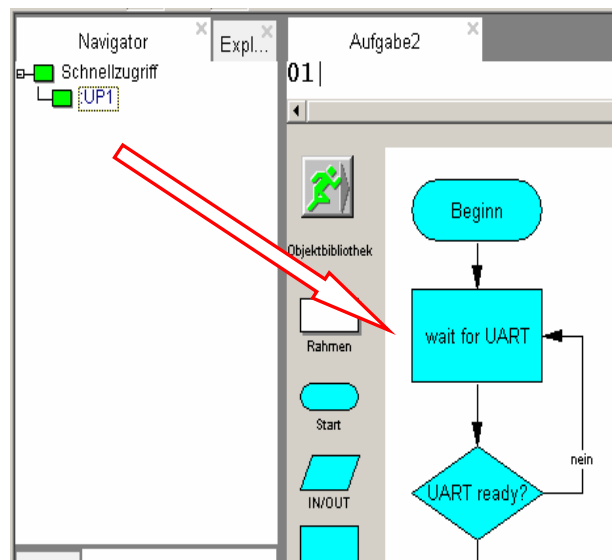
Ein wesentliches Merkmal von Unterprogrammen ist, dass diese von verschiedenen Stellen im Programm aufgerufen (*call*) werden können und auch dorthin zurückkehren (*return*). Um diese Möglichkeit zu nutzen, bietet SiSy das Anlegen von Referenzen. Vergleichen Sie dazu Absatz 3.2.

Für das Referenzieren (wiederholend zeigen und einbinden) gehen Sie wie folgt vor:

1. Zeigen Sie im Navigator das gewünschte Unterprogramm an
  - a. über den Schnellzugriff,
    - dort lässt sich das Original per Drag&Drop ablegen, oder
  - b. über die Navigatorsortierung „Unterprogramme“
    - Navigator -> rechte Maustaste -> Kontextmenü -> Programmablaufpläne.
2. Öffnen Sie das gewünschte Zieldiagramm, in dem das Unterprogramm verwendet werden soll.
3. Ziehen Sie per Drag&Drop das Unterprogramm in das Zieldiagramm, dabei wird nur eine Referenz (Link) auf das Original angelegt.
4. Integrieren Sie die Referenz wie oben beschrieben in den Programmfluss.



**Abbildung 25: Objekt im Navigator/Schnellzugriff hinzufügen**

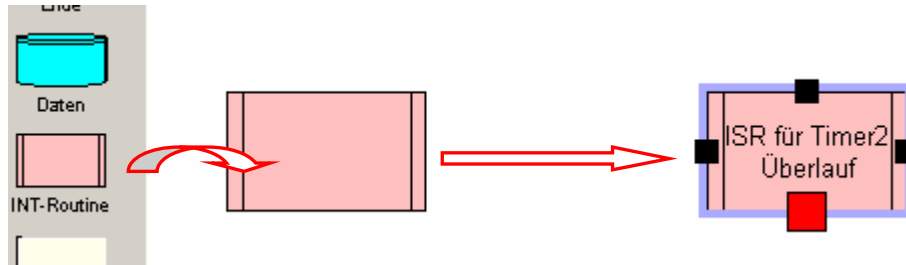


**Abbildung 24: Objekt aus dem Navigator referenzieren**

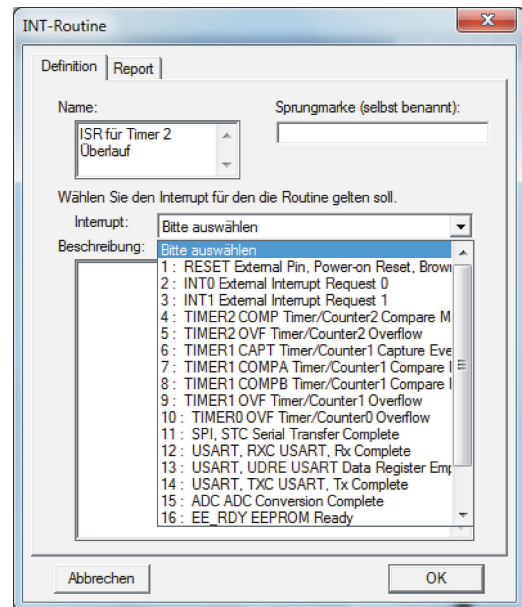
### 4.3.3 Interrupt-Service-Routinen (ISR) im PAP

Interrupt-Service-Routinen (im weiteren ISR) sind besondere Formen von Unterprogrammen. Diese werden von einer Interruptquelle des Mikrocontrollers (Timer, ADC, UART, usw.) bei entsprechenden Ereignissen automatisch an beliebiger Stelle im Programmfluss aufgerufen (Unterbrechung, engl. Interrupt). Es ist demzufolge nicht vorgesehen, eine ISR in den Programmfluss zu integrieren.

Um eine ISR zu erzeugen, ziehen Sie ein Objekt vom Typ INT-Routine aus der Objektbibliothek in das Diagramm des Hauptprogramms und definieren einen Namen.

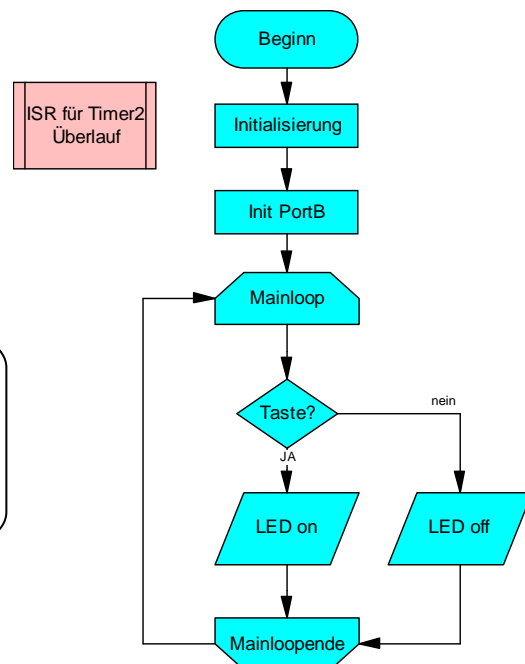
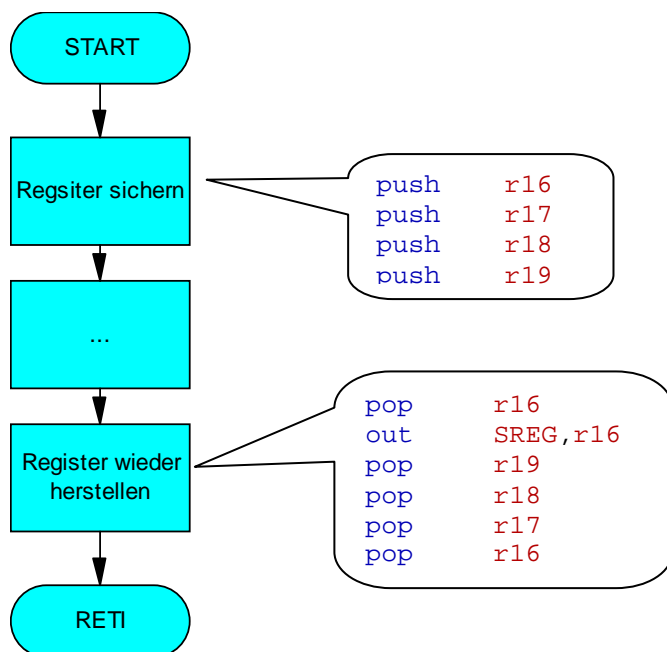


Über den Definieren-Dialog legen Sie die Sprungmarke und den Typ des Interrupts fest. Damit erfolgt durch den Codegenerator die Zuordnung der ISR zum entsprechenden Interruptvektor. Beachten Sie, dass die Liste der Interrupts abhängig vom gewählten Controllertyp ist.



Zum Entwerfen der ISR-Logik öffnen Sie das Objekt.

SiSy LibStore öffnet und bietet ein Grundgerüst ISR PAP AVR zum Download an. Vervollständigen Sie danach die ISR-Logik. Die ISR wird nicht in den Programmfluss integriert.

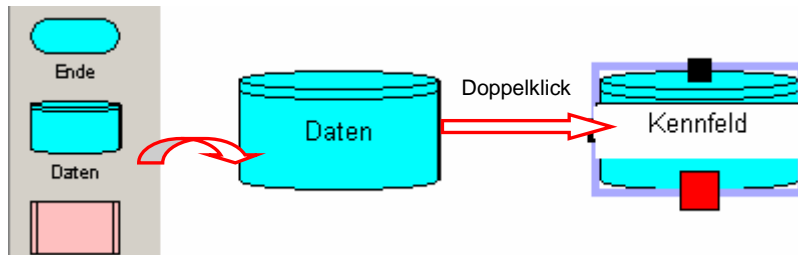


#### 4.3.4 Daten im PAP

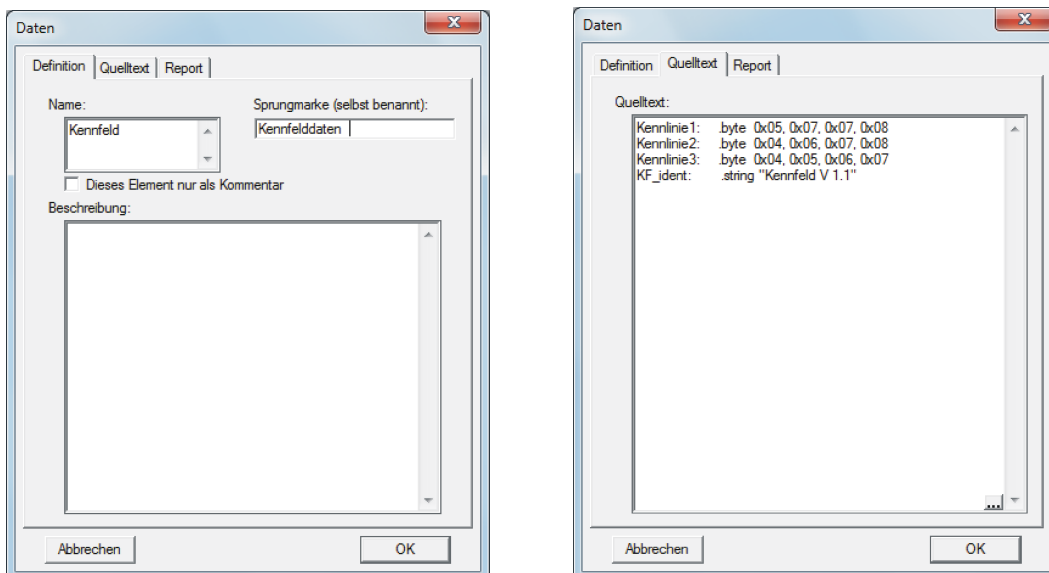
Konstante Daten, die im Programmspeicher (FLASH) des Mikrocontrollers abgelegt werden, können als Datenobjekt deklariert werden. Die im Datenobjekt deklarierten Daten werden durch den Codegenerator immer ans Ende des Quellcodes gestellt. Die zu generierende Marke/Marken für die Datenelemente können vom Entwickler frei gewählt werden.

##### **Anlegen eines Datenobjektes**

Zum Anlegen eines Datenobjektes ziehen Sie das betreffende Objekt in das gewünschte Diagramm und vergeben einen Namen



Die Marke und die Daten selbst können über den Definieren-Dialog festgelegt werden.



##### **Datenobjekt benutzen**

Im Quellcode werden die Daten über die vergebenen Markennamen angesprochen.

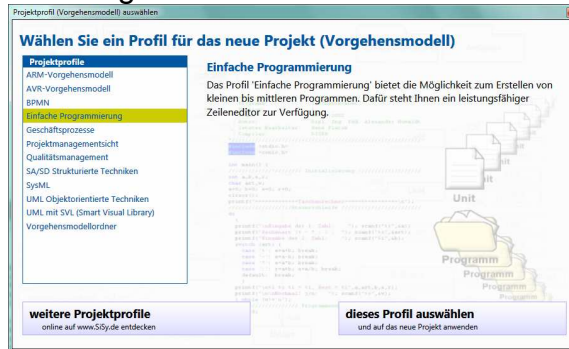
```

{
01 ; Kennfelddaten in Adressregister Laden
02 ldi r30,lo8 (Kennlinie1)
03 ldi r31,hi8 (Kennlinie2)
04 ...
05
06
}

```

### 4.4 Quellcode-Programmierung ohne Modellierung

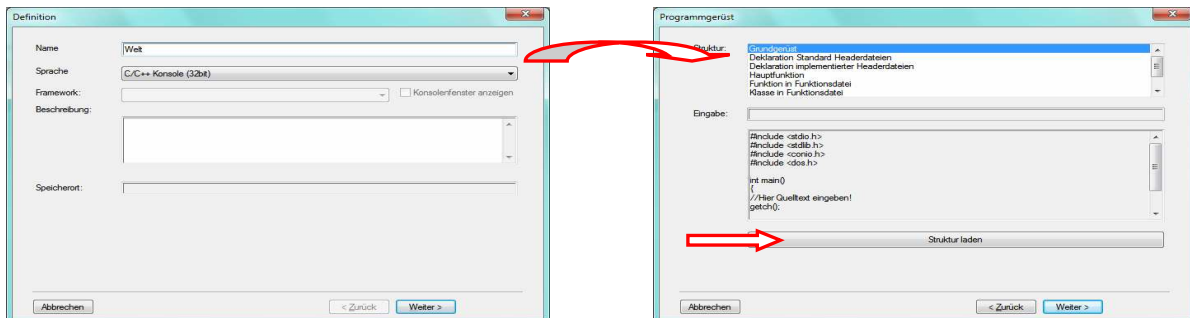
SiSy bietet die Möglichkeit, Programme in einem konventionellen Zeileneditor zu erstellen und zu übersetzen. Bei diesem Vorgehen bietet sich nach dem Start von SiSy das Profil „Einfache Programmierung“ an.



#### 4.4.1 Kleines Programm

##### PC Programme

Das Dialogfeld mit den Informationen zur Hardware-Konfiguration brechen Sie ab; danach aktivieren Sie die Auswahl „leeres Diagramm“. SiSy LibStore öffnet, ohne Vorlagen anzubieten. Kehren Sie zu SiSy zurück. Sie erhalten ein leeres Diagrammfenster. Ziehen Sie per Drag&Drop ein Objekt "kleines Programm" in das Diagrammfenster, wählen Sie die Sprache „C/C++ Konsole (32 Bit)“ und laden Sie aus dem nächsten Fenster die Grundstruktur für ein C-Programm.



Vervollständigen Sie im Quellcodefenster (Editorfenster) den Programmcode:

Aktivieren Sie „>>>Ausführen“. Bei fehlerfreiem Programm öffnet ein DOS Fenster mit der programmierten Ausgabe.



## AVR Programme

Nach der Auswahl des Profils öffnet myAVR ProgTool zur Einstellung der Hardware-Konfiguration.

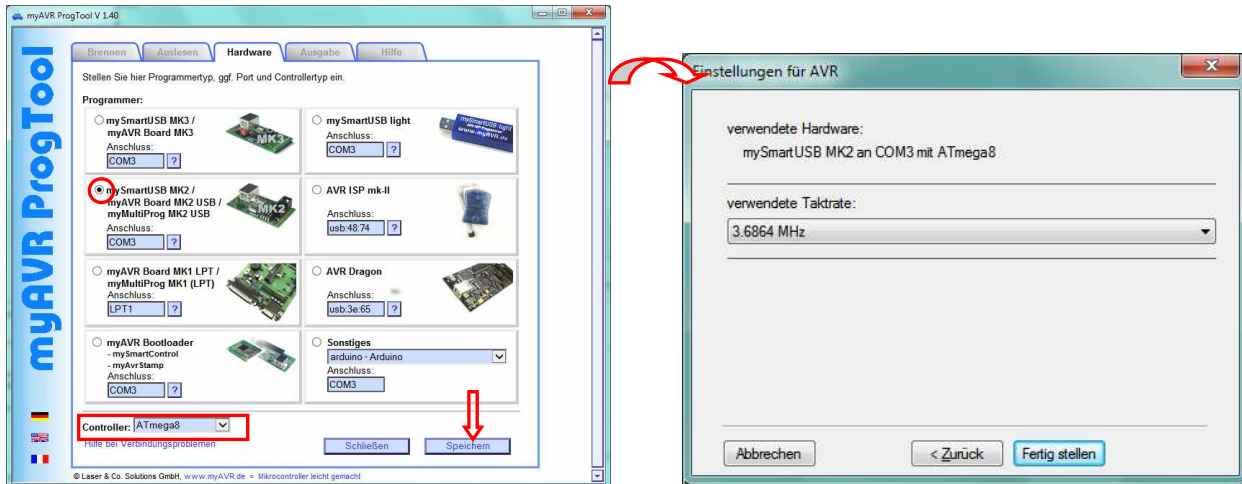


Abbildung 26: Einstellen der AVR Hardwarekonfiguration

### Hinweis:

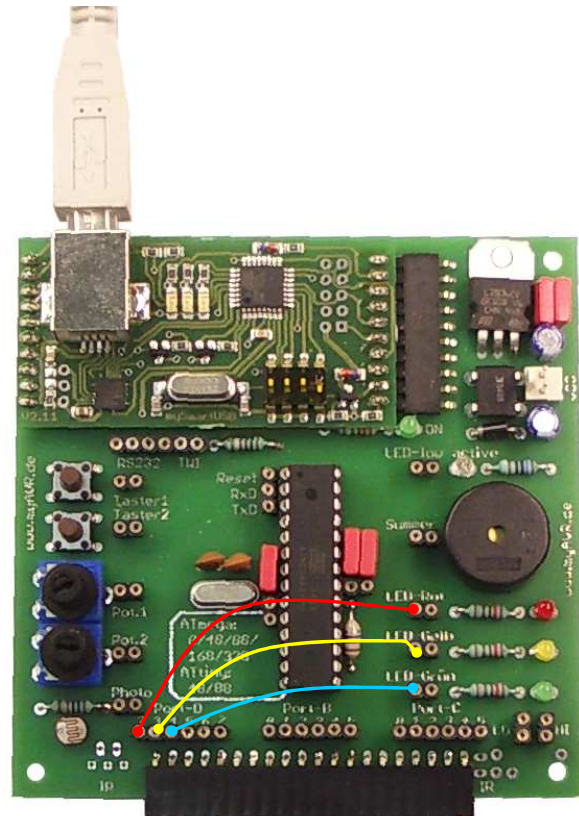
Bei jedem neuen Projekt müssen die Grundeinstellungen zur verwendeten Zielplattform vorgenommen werden (Mikrokontrollertyp, Taktrate, Programmer und I/O-Port). Erfolgen keine Einstellungen, geht SiSy von einem mySmartUSB MK2 Programmer an COM 3 aus. Als Mikrocontroller wird ein Atmega8 mit 3,6864 MHz verwendet.

Aktivieren Sie die Auswahl „leeres Diagramm“. SiSy LibStore öffnet, ohne Vorlagen anzubieten. Kehren Sie zu SiSy zurück und Sie erhalten ein leeres Diagrammfenster. Ziehen Sie per Drag&Drop ein Objekt "kleines Programm" in das Diagrammfenster, wählen Sie die Sprache „AVR Assembler“ und laden Sie aus dem nächsten Fenster die Grundstruktur.

### Hinweis:

Das Programmgerüst darf erst geladen werden, wenn der Zielcontroller ausgewählt wurde; die Vorlagen sind controller-spezifisch.

Im Beispiel wird auf dem myAVR Board MK2 ein Lauflicht erzeugt. Das Lauflicht soll über die LEDs 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.





```

;-----
;* Titel           :Lauflicht für myAVR Board
;*
;* Auszug aus dem Quellcode
;-----
.include          "avr.h"
;-----
;*
;-----
; 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
;-----

```

Der Quellcode muss in Maschinencode für den AVR-Prozessor übersetzt werden. Dafür stehen die Schaltflächen „Kompilieren“ und „Linken“ zur Verfügung. Bei fehlerfreier Übersetzung liegt das Programm als hex-Datei vor und kann auf den Prozessor gebrannt werden über die Schaltfläche „Brennen“.

Im Ausgabefenster vom myAVR ProgTool werden alle Aktionen protokolliert.

### STM32- und XMC-Produkte

Nach der Auswahl des Profils öffnet SiSy LibStore, ohne Vorlagen anzubieten. Kehren Sie zu SiSy zurück, wählen Sie „leeres Diagramm“. Ziehen Sie ein Objekt "kleines Programm" in das Diagrammfenster und wählen Sie die Sprache „ARM C++“. Nach der Auswahl von Hardware und Programmer können Sie die Grundstruktur laden.

	STM32	XMC
Hardware	STM32F407-Discovery	XMC4500 Relax Kit
Programmer	ST-Link V2	J-Link

### Programm für STM32F4 erstellen

Die Aufgabe lautet: Zwei LEDs auf dem STM32F4-Discovery sollen in kurzen Zeitabständen aufleuchten und damit ein „Blinklicht“ erzeugen.

Das Blinklicht soll über die LEDs angezeigt und von dem Prozessorport GPIOD gesteuert werden. Die Realisierung erfolgt über GPIO Pin 12 und 13. Dieses wird mit dem Befehl Bit-Rotation nach rechts verschoben und an den Port GPIOD des Prozessors ausgegeben. Auf Grund der Prozessorgeschwindigkeit muss die Ausgabe des Blinklichtes für unser Auge verzögert werden. Geben Sie den Quellcode ein bzw. ergänzen Sie die Programmvorlage!

Bei der Eingabe des Quellcodes springt nach drei zusammenhängenden Buchstaben die Code-Vervollständigung an und listet alle Bezeichner fortlaufend gefiltert.

```

19 void initApplication()
20 {
21     SysTick_Config(SystemCoreClock/100);
22     // weitere Initialisierungen durchführen
23     // Takt für GPIOD an
24     RCC
25 }
26 int main()
27 {
28     Sys
29     ini
30     do{
31         + RCC_AHB1Periph_CKIDATARAMEN
32         + RCC_AHB1Periph_CRC
33         + RCC_AHB1Periph_DMA1
34         + RCC_AHB1Periph_DMA2
35         + RCC_AHB1Periph_ETH_MAC
36         + RCC_AHB1Periph_ETH_MAC_PTP

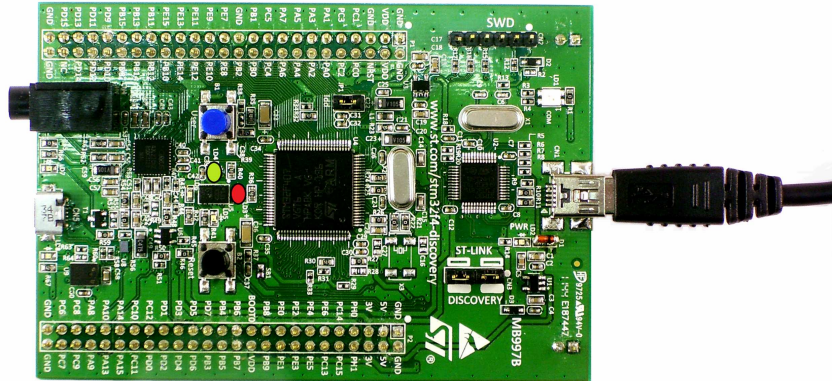
```

```

/-----
// Titel      : Blinklicht auf dem STM32F4-Discovery
//-----
//
// Auszug aus dem Quellcode
//
int main(void)
{
    SystemInit();
    initApplication();
    GPIO_SetBits(GPIOD,GPIO_Pin_13);
    do{
        // Eingabe
        // Ausgabe
        // Verarbeitung
        GPIO_ToggleBits(GPIOD,GPIO_Pin_13|GPIO_Pin_12);
        delay(1000000);
    } while (true);
    return 0;
}
extern "C" void SysTick_Handler(void)
{
    // Application SysTick default 10ms
}
//-----

```

Der eingegebene Quellcode muss in Maschinencode für den ARM-Prozessor übersetzt werden. Wählen Sie dazu die Schaltflächen „Kompilieren“ und „Linken“. Bei fehlerfreier Übersetzung liegt das Programm als „Blinklicht.hex“ vor und kann auf den FLASH-Programmspeicher des Prozessors gebrannt werden. Dazu verbinden Sie das STM32F4-Discovery über das Programmierkabel mit dem USB-Port Ihres Rechners. Zum Brennen wählen Sie die Schaltfläche „Brennen“. Beim Brennen des Beispiels öffnet ein Ausgabefenster und zeigt Protokollausgaben der Aktionen an.



### Programm für XMC erstellen

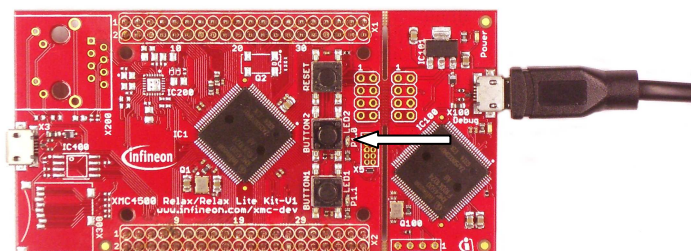
Die Aufgabe lautet: Eine LED auf dem XMC4500 Relax Kit soll eingeschaltet werden. Die LED ist bereits fest mit dem *Pin 1.0* verbunden.

```
//-----
// Titel      : Hallo XMC
//-----
//
// Auszug aus dem Quellcode
//
int main(void)
{
    SystemInit();
    initApplication();
    do{
        // bleibt erst mal leer
    } while (true);
    return 0;
}

extern "C" void SysTick_Handler(void)
{
    // Application SysTick
    // bleibt vorerst auch leer
}
```

Der eingegebene Quellcode muss nun in Maschinencode für den ARM-Prozessor übersetzt werden. Wählen Sie nacheinander die Schaltflächen „Kompilieren“, „Linken“ und „Brennen“ oder „>>>Ausführen“.

Das Programm wird auf den Controller gebrannt und bei fehlerfreier Übertragung leuchtet jetzt eine LED auf dem XMC4500 Relax Kit.

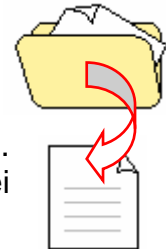


## 4.4.2 Programm mit mehreren Dateien

Für die Entwicklung eines größeren Programms ist es unzweckmäßig, alle Befehle in eine Datei (Unit) zu schreiben. Der Grund dafür ist, dass bei mehr als 60 bis 80 Zeilen Quellcode die Übersicht über die Programmstruktur verloren geht. Selbst die Unterteilung in Unterprogramme reicht ab einer bestimmten Größe von Programmen nicht mehr aus. SiSy erlaubt zwar in kleinen Programmen bzw. je Unit 10 Kilobyte Code. Das sind in Assembler zum Beispiel über 1000 Zeilen. Ein kleines Programm bzw. eine einzelne Unit sollte jedoch nie mehr als 80 bis 120 Zeilen haben. Wird diese Grenze erreicht, sollte das Programm in logische Einheiten (Units, Module) gegliedert werden. Dabei fasst man alles zusammen, was zu einer bestimmten Funktionalität oder einer bestimmten Baugruppe wie zum Beispiel dem AD-Wandler gehört. Physisch entstehen dabei mehrere Dateien, die für sich genommen wieder übersichtlich sind, da diese dann nur 80 bis 120 Zeilen Code enthalten. Das Übersetzungsprogramm (Assembler, Compiler, Linker) sorgt dann dafür, dass alle einzelnen Units zu einem vollständigen Programm zusammengefügt werden.

SiSy bietet für dieses Vorgehen ebenfalls das Profil „Einfache Programmierung“ an.

Für ein Programm mit mehreren Modulen legen Sie zuerst ein Objekt vom Typ „Programm“ an und erfassen im Definieren-Dialog einen Namen und wählen die Sprache aus. Ausgehend von diesem Objekt legen Sie im Diagramm darunter die Unterprogramme an, Objekte vom Typ „Unit“. Benennen Sie eine Unit mit „main“. Diese Unit bildet das Hauptprogramm. Von der Entwicklungsumgebung wird damit erkannt, dass es sich hierbei um das Hauptmodul handelt.



### Konsolenprogramme

In einem kleinen Programm mit mehreren Modulen wird in einer einfachen Anwendung vom Hauptprogramm ein Unterprogramm aufgerufen und ins Hauptprogramm zurückgekehrt. Auf dem Bildschirm werden die Texte dieser Aktionen ausgegeben.

Die Programmiersprache ist „C/C++ Konsole“, es werden die Units „Main“ und „UP1“ verwendet. In dem Dialogfenster zur Definition von „UP1“ ist der Haken gesetzt in den beiden Zeilen

- „Headerdef-Datei erzeugen, Datei mit Deklaration wird generiert“
- „Funktionsdef-Datei erzeugen (.cpp)“

```

Hallo
*.cpp *.h
01 #include <stdio.h>
02 #include <conio.h>
03 #include "UP1.h"
04 int main()
05 {
06     //Hier Quelltext eingeben!
07     printf("Hallo Main");
08     up1();
09     printf("und wieder zurueck!");
10     getch();
11     return 0;
12 }
13
Objektbibliothek
Unit
Main
UP1

```

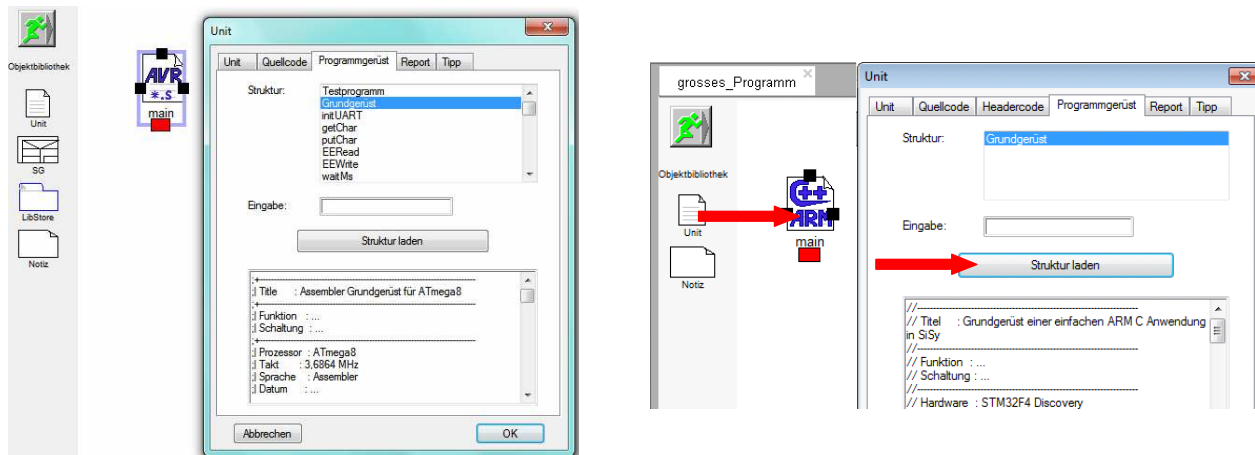


## Mikrocontroller-Programme

Für Mikrocontroller-Programme sind folgende Sprachen im Definieren-Dialog im Objekt „Programm“ auszuwählen:

- AVR: AVR-Assembler
- STM32: ARM C++
- XMC: ARM C++

Die Anzahl der Units ist von der Aufgabenstellung und der Gesamtgröße des Programms abhängig. Für die Unit „main“ kann aus dem Definieren-Dialog der Unit eine Grundstruktur geladen werden.

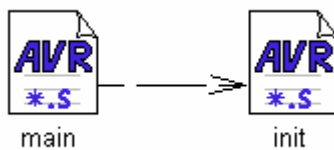


## Auszug aus dem Quellcode der Unit „main“ für AVR-Programme

```

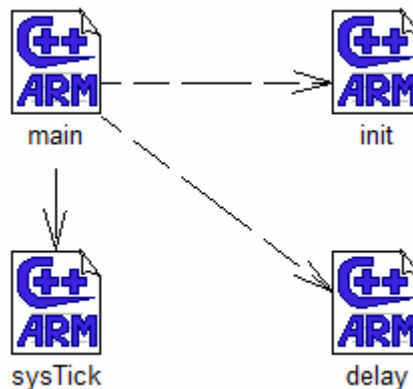
;-----
; ...
main:
    ldi r16,hi8(RAMEND)           ; Main program start
    out SPH,r16                  ; Set Stack Pointer to top of RAM
    ldi r16,lo8(RAMEND)
    out SPL,r16
    rcall  init                  ; Hier Init-Code eingetragen.
;-----
mainloop: sbic    PIND,2          ; Hier Quellcode eingetragen
           rjmp   led_aus
led_an:    sbi    PORTB,0
           rjmp   mainloop
led_aus:   cbi    PORTB,0
           rjmp   mainloop
;-----
#include "init.s"                ; Eintrag bei Verbindung von "main" auf "init"

```



## Auszug aus dem Quellcode der Unit „main“ für ARM-Programme

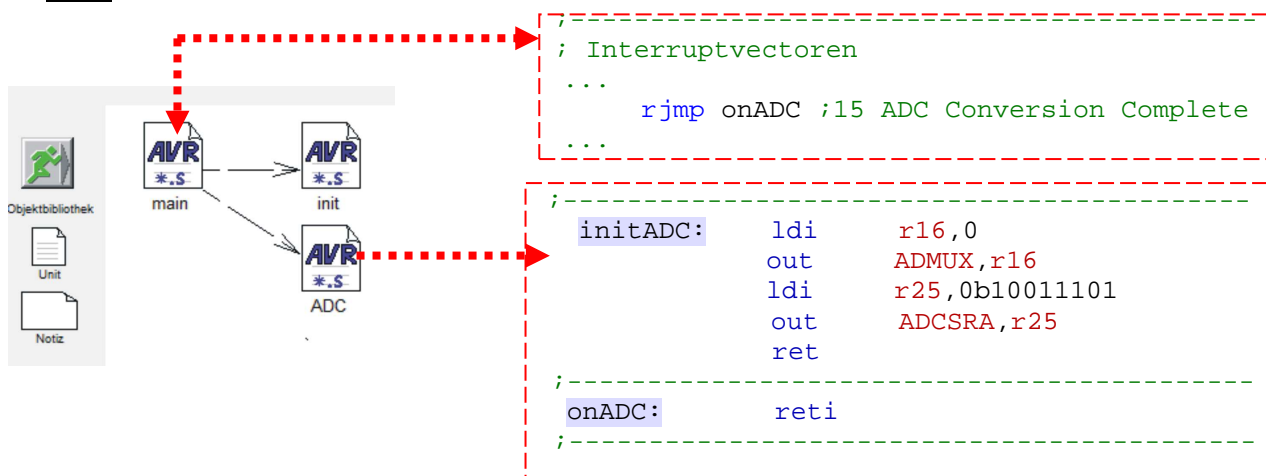
```
//-----
//...
int main(void)
{
    SystemInit();
    initApplication();
    do{
        // Eingabe
        // Verarbeitung
        // Ausgabe
        GPIO_ToggleBits(GPIOD,GPIO_Pin_12);
        delay(10000000);
    } while (true);
    return 0;
}
//-----
```

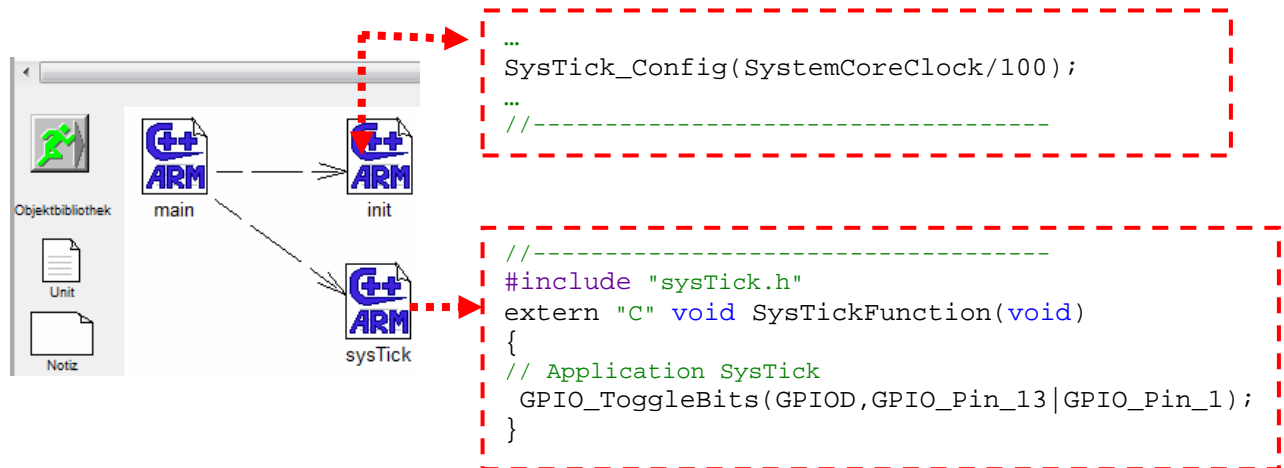


## Interrupt-Service-Routine (ISR) in großen Programmen

Interrupt-Service-Routinen (im weiteren ISR) sind besondere Formen von Unterprogrammen. Diese werden von einer Interruptquelle des Mikrocontrollers (Timer, ADC, UART, usw.) bei entsprechenden Ereignissen automatisch an beliebiger Stelle im Programmfluss aufgerufen (Unterbrechung, engl. Interrupt). Es ist nötig, die Interruptquelle entsprechend zu konfigurieren. Es empfiehlt sich für jedes interruptfähige Gerät eine eigene Unit anzulegen. Diese ist mit der Hauptunit zu verbinden.

### AVR



ARMHinweis:

Die Parameter zum Aufruf der Interrupt-Routinen sind controllerabhängig. Sie sind dem jeweiligen Referenzblatt des Controllers zu entnehmen.

# 5 Weitere Methoden

## 5.1 Objektorientierte Techniken mit der UML

Mit Hilfe unterschiedlicher Diagramme der UML besteht die Möglichkeit der komplexen Anwendungsentwicklung, von der Analyse bis zum lauffähigen Programm. Zu diesen Diagrammen gehören z.B. Anwendungsfalldiagramm, CRC-Karten, Klassendiagramm, Aktivitätsdiagramm usw.

Für die Objektorientierten Techniken mit der UML können Sie in SiSy das Profil „UML Objektorientierte Techniken“ nutzen. SiSy LibStore bietet u.a. die Diagrammvorlage „UML allgemein – objektorientierte Techniken mit SVL“ an. Sie erhalten das Diagramm „Vorgehensmodell“ für die weitere Arbeit.

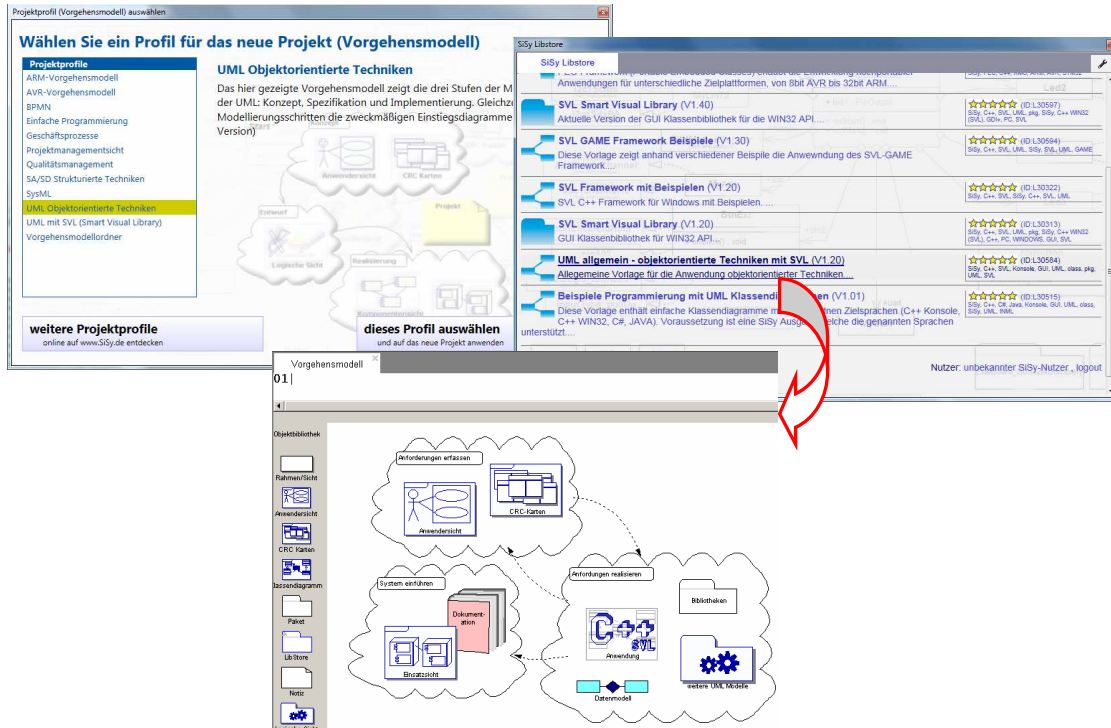


Abbildung 27: Auswahl Profil und Vorlage für objektorientierte Techniken

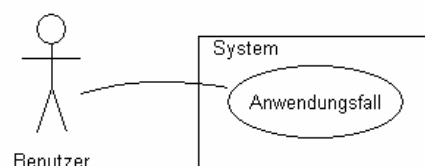
### Objektorientierte Analyse

Die Modellierung beginnt mit der Anwendersicht. Das Anwendungsfalldiagramm ist ein Beschreibungsmittel der UML und stellt das Zusammenspiel zwischen dem System, seinen Anwendungsfällen und dessen Anwendern dar. Ausgehend von den Szenarien, die als Anwendungsfall modelliert wurden, werden in den CRC-Karten die potentiellen Klassenkandidaten aufgenommen.

#### Anwendungsfalldiagramm (Use case)

Das System wird als Rahmen gekennzeichnet. Es besitzt Anwendungsfälle, auf die Anwender zugreifen. Dies wird durch Kanten zwischen Anwendungsfällen und Anwendern gekennzeichnet.

Beim Öffnen des Objektes „Anwendersicht“ bietet SiSy Libstore Vorlagen an; hier ist die Vorlage „Grundgerüst Anwendungsfall“ zu empfehlen.

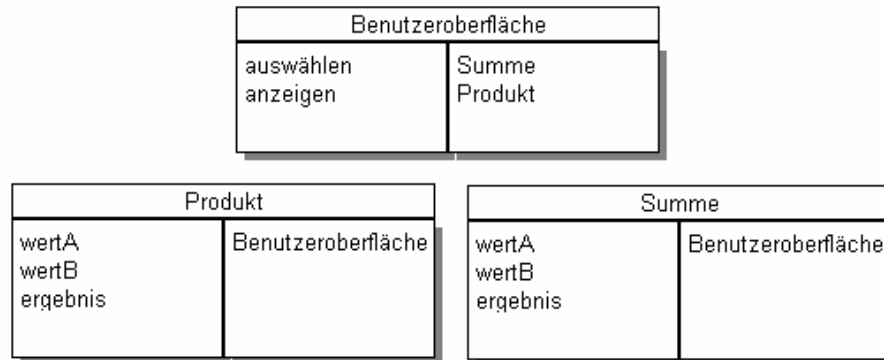




**CRC-Karten**

Aus der verbalen Aufgabenstellung müssen die Klassenkandidaten analysiert werden. Diese werden mit ihren Eigenschaften in CRC-Karten erfasst.

In der Kopfzeile steht der Klassenkandidat, in die linke Spalte werden die Verantwortlichkeiten (Eigenschaften und Verhalten) der Klasse definiert und in der rechten Spalte die Zusammenarbeit der Klasse mit anderen Klassen. Die Erfassung erfolgt im Definieren-Dialog.



**Objektorientiertes Design**

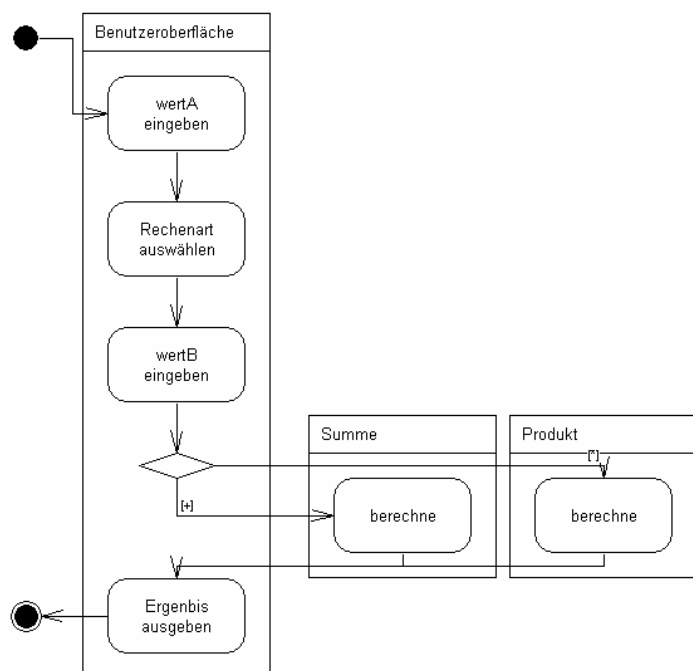
Der objektorientierten Analyse schließt sich ein objektorientiertes Design an. Dabei werden Anwendungsfälle aus dem Anwendungsfalldiagramm mit Verhaltensdiagrammen verfeinert und es wird ein grober Klassenentwurf entwickelt.

**Aktivitätsdiagramm**

Das erwartete Systemverhalten wird in Verfeinerung der Anwendungsfälle modelliert. Das dynamische Verhalten und die Interaktion zwischen Objekten kann in der Form eines Aktivitätsdiagramms dargestellt werden.

Zum Erstellen eines Aktivitätsdiagramms, also der Verfeinerung eines Anwendungsfalls, öffnen Sie im Anwendungsfalldiagramm einen Anwendungsfall. Das Laden einer Vorlage ist nicht erforderlich.

Aus der Objektbibliothek ziehen Sie das Objekt „Partition“ in das Diagrammfenster; dieses Objekt bildet den Rahmen für die weitere Modellierung und es erhält den Namen eines Klassenkandidaten. Für die weitere Modellierung benötigen Sie Objekte vom Typ „Aktion“, diese werden in die Partition eingefügt. Verbinden Sie die Objekte in der logischen Reihenfolge. Je ein Objekt „Start“ und „Ende“ vervollständigen das Diagramm.



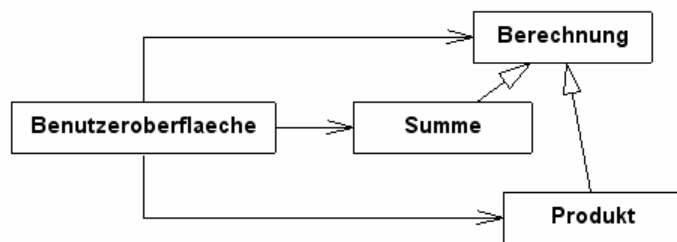
### Klassentwurf

Beim Klassentwurf werden die Klassenkandidaten aus den CRC-Karten in das Klassendiagramm übernommen. Die gemeinsamen Strukturmerkmale der Klassenkandidaten setzen Sie in eine Generalisierung um.

Ausgehend von dem Objekt „weitere UML-Modelle“ (es ist der Objekttyp „Logische Sicht“) erfolgt die Modellierung des Klassentwurfs. Öffnen Sie dieses Objekt und laden keine Vorlage aus dem SiSy Libstore. Ziehen Sie ein „Klassendiagramm“ in das leere Diagrammfenster, tragen Sie im Definieren-Dialog einen Namen ein und wählen Sie die Sprache „C++ Konsole (32bit)“. Öffnen Sie das Objekt und arbeiten Sie ohne Vorlage.

In dem Klassendiagramm werden die im Anwendungsfall erstellten Partions als Klassen angelegt. Ziehen Sie aus der Objektbibliothek eine „Klasse“ in die Diagrammfenster und benennen Sie diese; markieren Sie das Kontrollkästchen „diese Klasse generieren“. Erstellen Sie analog dazu die weiteren Klassen.

Modellieren Sie die Beziehungen der Klassen, indem Sie Verbindungen zwischen den Klassen ziehen und den Verbindungstyp festlegen, z.B. „Assoziation gerichtet“ oder „Vererbung“.



### Ausblick

Den Objektorientierten Techniken (Analyse und Design) schließt sich die Objektorientierte Programmierung an. Dazu gehören die Verfeinerung des Klassendiagramms, die Implementierung und die Programmausführung.

Für den *Feinentwurf* werden den Klassen Operationen und Attribute zugeordnet. Attribute und Operationen besitzen Eigenschaften, wie z.B. Rückgabewerte und Namen, die im Definieren-Dialog festgelegt werden.

In der *Implementierungsphase* wird den Operationen Quellcode zugeordnet.

## 5.2 Prozessmodellierung

Mit SiSy steht ein graphisches Werkzeug für die Modellierung von Geschäftsprozessen, zur QM-konformen Darstellung von Aufbauorganisationen, betrieblichen Funktionen, Dokumentensystemen usw. zur Verfügung. Die erstellten Unternehmensmodelle lassen sich mit der integrierten Dokumentengenerierung in Word-Dokumente exportieren. Wesentlich für die Geschäftsprozessmodellierung mit SiSy ist die Integration verschiedener Modelle mit der Ablaufsicht in Form von erweiterten Prozessketten: WER – WAS – WIE – WOMIT.

Für die Modellierung von Geschäftsprozessen bietet SiSy nach dem Start das Profil „Geschäftsprozesse“. SiSy LibStore bietet für die Geschäftsprozessmodellierung verschiedene Vorlagen an, oft verwendet ist „Vorlage Strategiepyramide“.

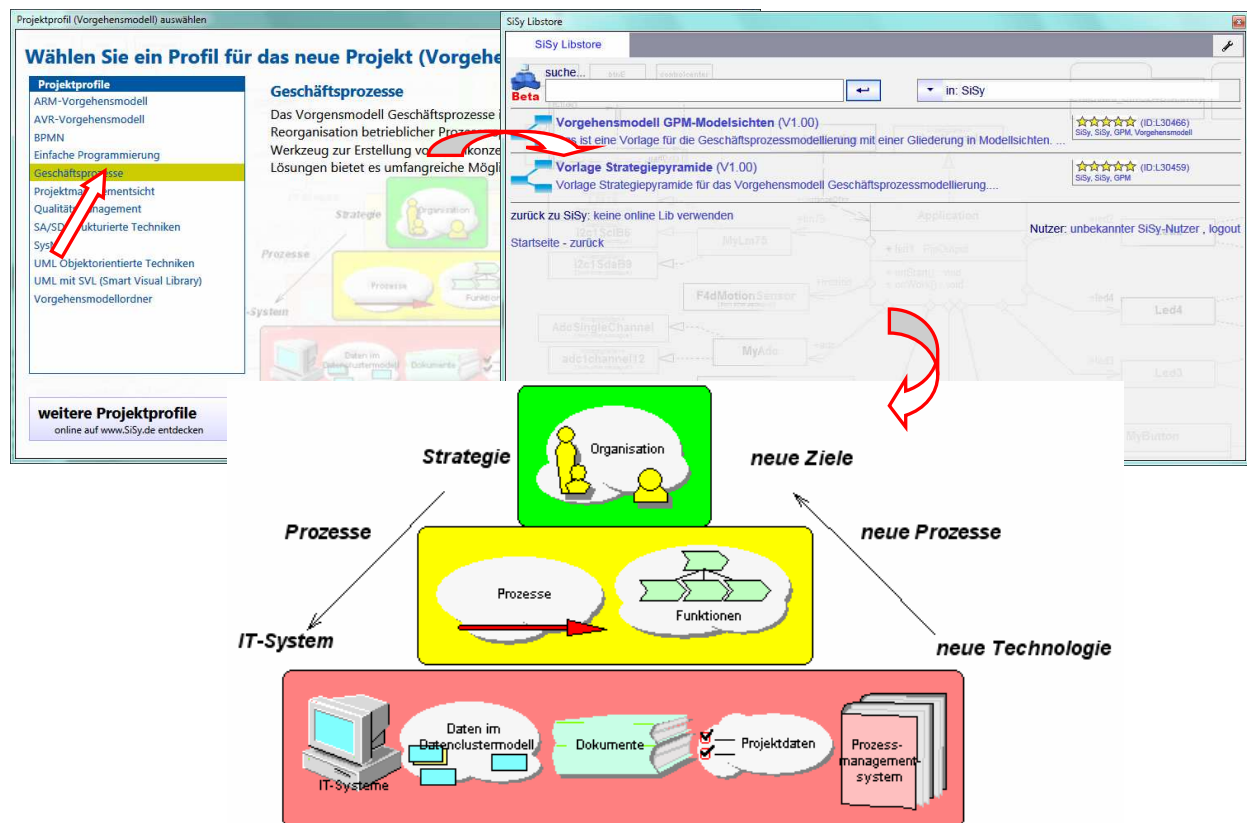
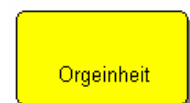


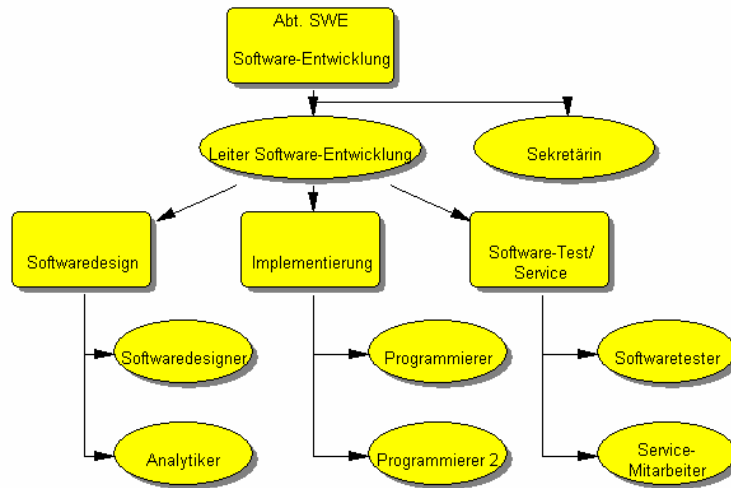
Abbildung 28: Projekt anlegen für Geschäftsprozessmodellierung

Für die Modellierung einer QM-konformen Darstellung erweist sich das Profil „Qualitätsmanagement“ komfortabel. SiSy LibStore bietet keine Vorlagen an, statt dessen kann aus dem Assistenten eine QM-Vorlage ausgewählt werden.

### Organisation darstellen

Für das „WER“ ist es erforderlich, die Organisationsstruktur des Unternehmens zu modellieren. Ziehen Sie dafür die erforderlichen Objekte aus der Objektbibliothek in das Diagrammfenster. Die „Orgeinheit“ kann mit Stellen weiter untergliedert und mit Personen hinterlegt werden. Über den Verteiler können Sie die Objekte mittels einer Kante verbinden und so die Hierarchie darstellen. Der Kante kann über „Definieren“ die Art der Unterstellung (disziplinarisch oder fachlich) zugeordnet werden.



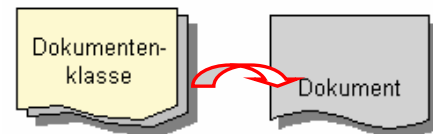


**Abbildung 29: Beispiel einer modellierten Organisationsstruktur**

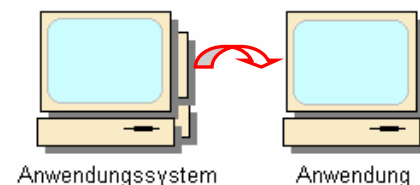
*Ressourcen erfassen*

Das WOMIT in der Prozessmodellierung bezieht sich auf die verwendeten Ressourcen. Dazu gehören Dokumente, Anwendungen, Daten und technische Ressourcen; wobei nicht in jedem Prozess alle Ressourcen zu beachten sind.

Dokumente werden in Verfahren und Arbeitsabläufen erzeugt, verändert und benutzt. In SiSy besteht die Möglichkeit als übergeordnete Einheit Dokumentenklassen anzulegen, in denen Dokumente hinterlegt werden können.

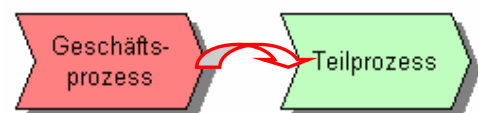


Anwendungssysteme sind übergeordnete Einheiten für systematisch gruppierbare Anwendungen wie Datenverarbeitungssysteme, die mit diesen arbeiten bzw. diese verwalten und bereitstellen. Die komplexen Anwendungssysteme werden in einzelne Anwendungen aufgegliedert.



*Geschäftsprozesse darstellen*

Für das WAS werden im weiteren Ablauf die wichtigsten wertschöpfenden Geschäftsprozesse erfasst und in Teilprozesse untergliedert. Für das WIE besteht die Möglichkeit einer ergänzenden Beschreibung.



*Teilprozesse modellieren*

Ein Teilprozess soll in seinem Ablauf als ereignisgesteuerte Prozesskette (eEPK) modelliert werden. Öffnen Sie den Teilprozess und beschreiben Sie den Ablauf des Prozesses durch weitere Prozesse und Tätigkeiten, fügen Sie die dazu benötigten Ressourcen hinzu; achten Sie auf die Zuordnungsrichtung. Referenzieren Sie bereits angelegte Objekte in das Diagrammfenster. Beachten Sie dazu die Abschnitte zu Referenzen in diesem Handbuch.

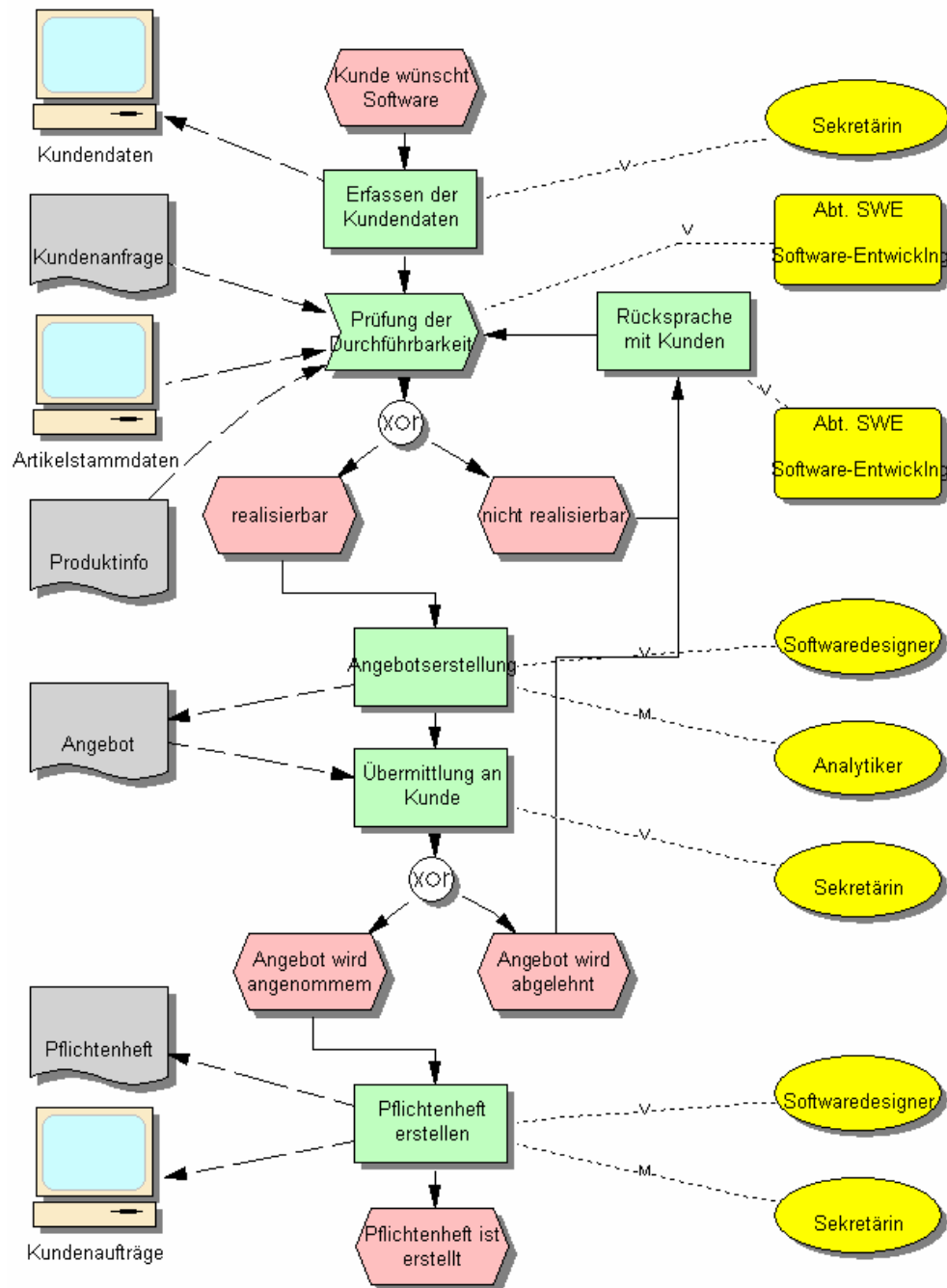


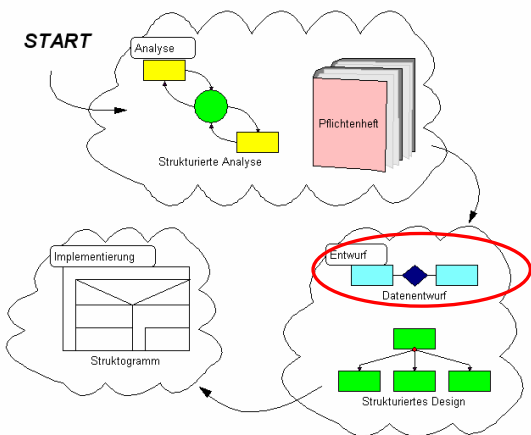
Abbildung 30: modellierter Prozess mit Ressourcen und Zuständigkeiten

### 5.3 Datenmodellierung (ERD/Klassendiagramm)

Auf der Grundlage eines in SiSy modellierten Entity-Relationship-Diagrammes (ERD) besteht die Möglichkeit zur Erstellung einer Zieldatenbank. Die erforderlichen Schritte dafür sind:

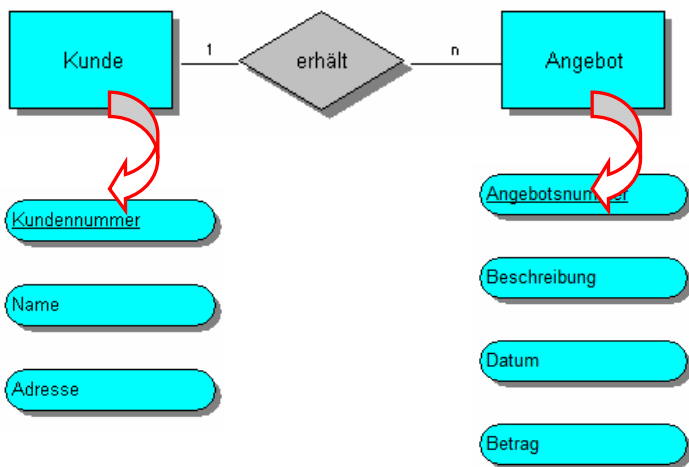
- Anlegen von Entities
- Verbinden der Entities mit Relationship
- Eintragen der Attribute in die Entities, Zuordnung von Primärschlüssel
- Erstellen der anzulegenden Datenbank, Eintragen der Feldnamen und Typen
- SQL ausführen

Für die Datenmodellierung in SiSy wählen Sie das Profil „SA/SD Strukturierte Techniken“. SiSy LibStore öffnet und bietet die Vorlage „IT-Projekt – SA/SD“ an. Sie erhalten das Vorgehensmodell für die weitere Arbeit.

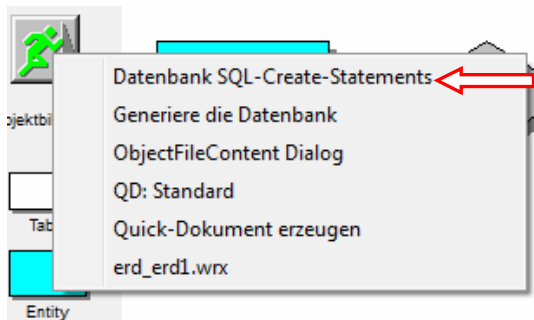


Ausgehend vom Objekt „Datenentwurf“ legen Sie Objekte vom Typ „Entity“ und „Relationship“ an und benennen diese. Verbinden Sie die Objekte und weisen die Kardinalität zu.

Öffnen Sie die Entities und weisen Sie Attribute (Eigenschaften) und Primärschlüssel zu. Im Definieren-Dialog der Attribute werden die Eigenschaften festgelegt. Dazu gehören Name, Typ (Text, Zahl, logisch oder Datum) und Kennzeichnung für Primärschlüssel. Fremdschlüssel werden entsprechend der modellierten Beziehungen vom Programm festgelegt; sie sind nur im Bedarf zu korrigieren.



Aus dem Aktionsmenü kann die Datenbank generiert werden.



## 6 Zusätzliche Werkzeuge

### 6.1 Einführung

Bestimmte Add-Ons installieren als zusätzliche Komponenten externe Programme. Diese sind nach der Installation über den Menüpunkt *Werkzeuge* erreichbar.

### 6.2 Das SiSy ControlCenter

Das SiSy ControlCenter ist ein universelles Terminalprogramm zur Kommunikation mit Mikrocontrollerapplikationen, die über eine serielle Schnittstelle (UART) oder USB Anbindung mit virtuellem COM-Port zum PC verfügen. Es kann für Test- und Debug-Meldungen sowie Visualisierung und Protokollierung von Messdaten genutzt werden. Dazu bietet das SiSy ControlCenter umfangreiche Konfigurationsmöglichkeiten.

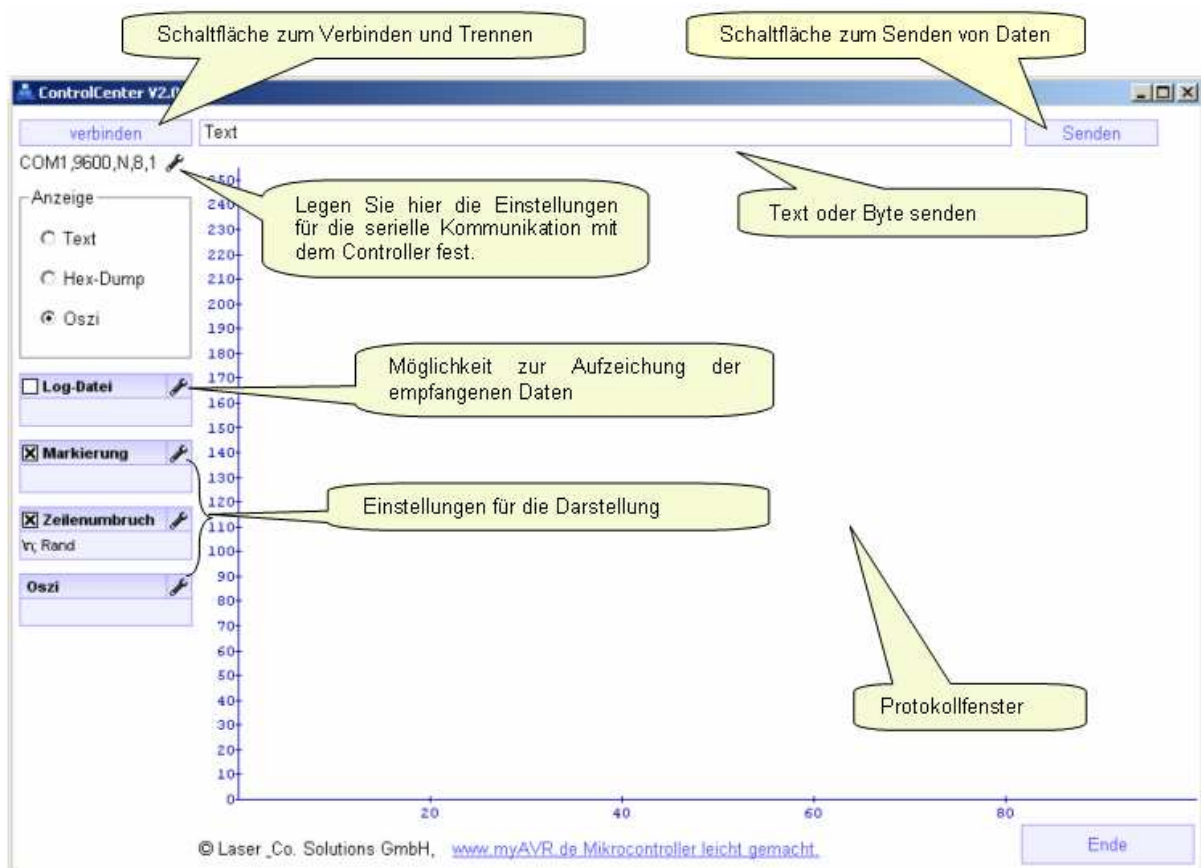


Abbildung 31: Oberfläche SiSy ControlCenter

### 6.2.1 Kommunikation mit dem Controller

Das myAVR Board MK2 verfügt über den USB Programmer mySmartUSB MK2, das myAVR Board MK3 über den USB-Programmer mySmartUSB MK3. Diese Programmer stellen gleichzeitig einen virtuellen COM-Port für die Kommunikation zur Verfügung.

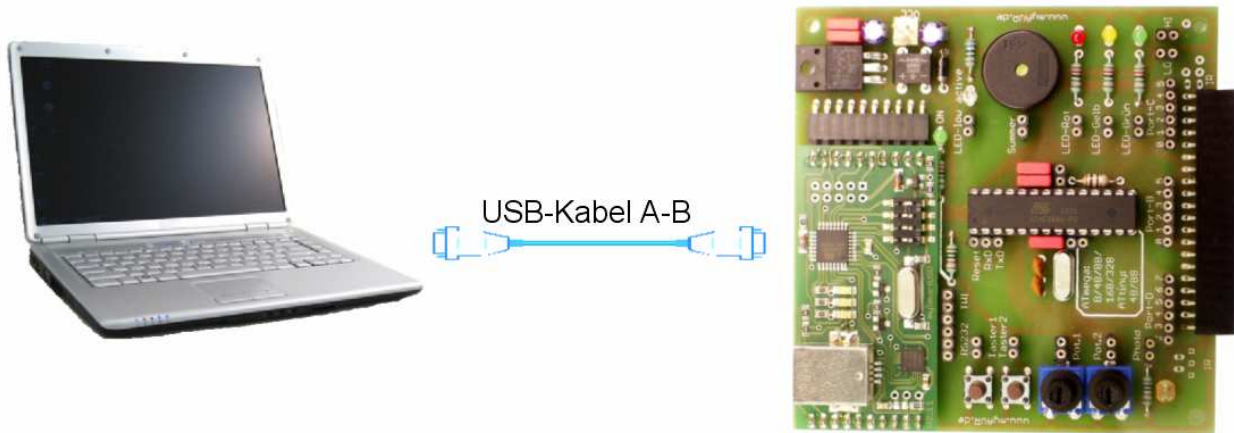


Abbildung 32: : Beispiel mit myAVR Board MK2

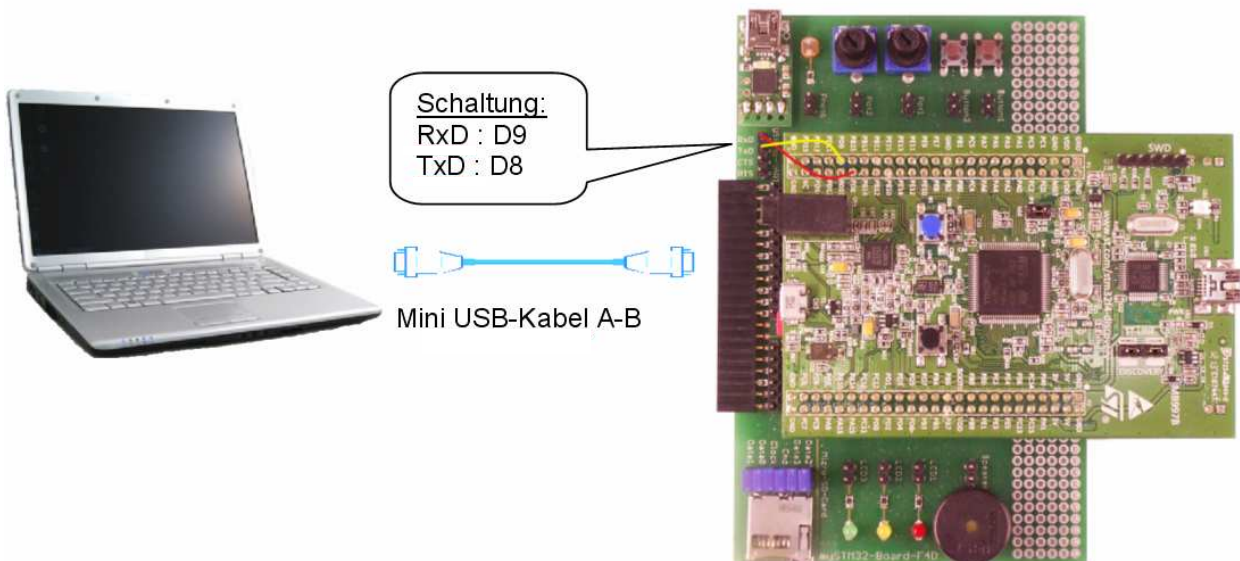


Abbildung 33: Beispiel mit einem mySTM32-Board-F4D



### Einstellungen für die seriellen Verbindung

Für eine erfolgreiche Kommunikation mit dem Mikrocontroller ist es wichtig, dass Sender und Empfänger von seriellen Daten die gleichen Parameter für die Datenübertragung konfiguriert haben. Auf der PC Seite erfolgt die Konfiguration der Kommunikationsparameter im SiSy ControlCenter über das Symbol für die Einstellungen der Verbindung.

Der entsprechende COM-Port ist z.B. aus dem Gerätemanager zu ermitteln.

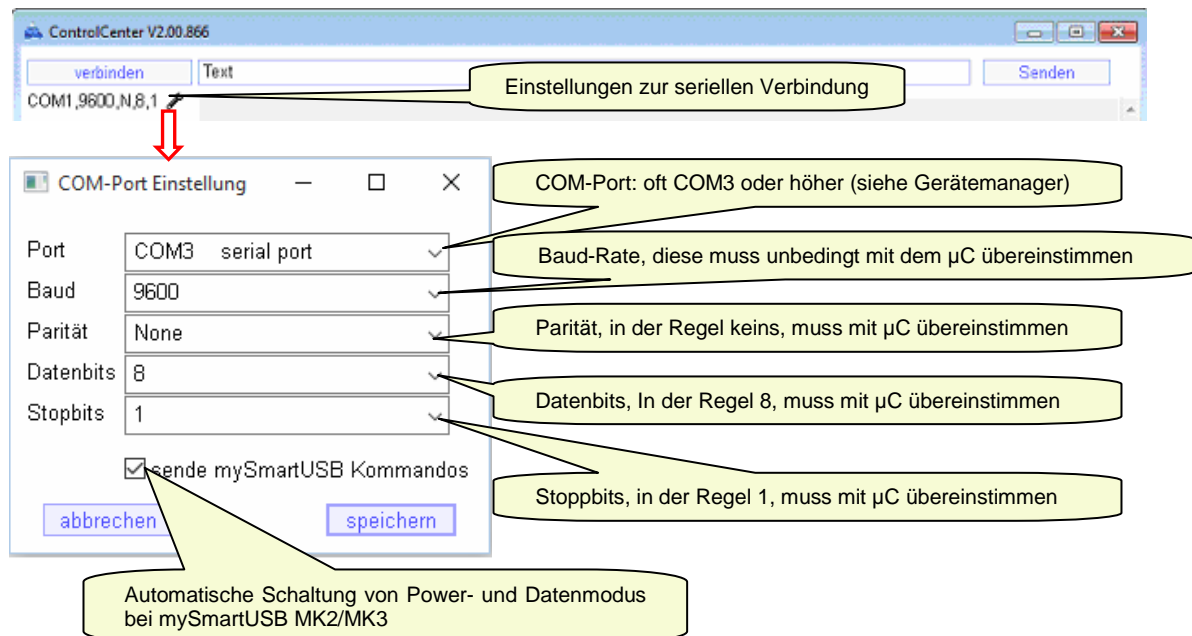


Abbildung 34: Einstellungen für serielle Verbindung

### Besonderheiten für myAVR Systemboards und mySmartUSB MK2/MK3

Bei einem myAVR Systemboard sowie bei den Programmern mySmartUSB MK2/MK3 können Sie die Spannungsversorgung des Boards und die Umschaltung in den Datenmodus aus dem SiSy ControlCenter heraus steuern. Dazu wählen Sie in den Einstellungen „sende mySmartUSB Kommandos“. Eine manuelle Umstellung ist damit nicht nötig.

Im Mikrocontrollerprogramm sind die gleichen Parameter bei der Initialisierung der UART zu wählen.

```

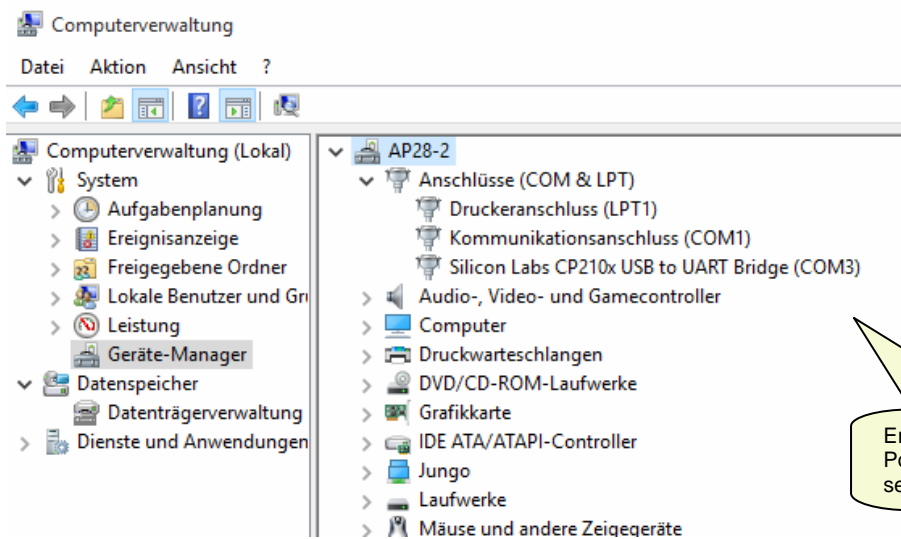
...
public: void onStart()
{
    // hier Initialisierungen durchführen
    wert=0;
    uart.config(9600,UDR);
}
public: void onWork()
...

```

Beispiel für die Konfiguration der UART des Mikrocontrollers mit 9600 Baud.

## Geräte-Manager

Die erfolgreiche Verbindung mit dem Controllerboard kann aus dem Gerätemanager entnommen werden, ebenso der entsprechende COM-Port (Nummerierung).



### 6.2.2 Daten empfangen vom Controller

Das SiSy ControlCenter empfängt Daten über den gewählten COM-Port und stellt diese im Protokollfenster dar. Damit können Statusmeldungen, Fehlermeldungen oder auch Messwerte erfasst werden. Die Voraussetzung ist, dass die serielle Verbindung hergestellt (USB Kabel), korrekt konfiguriert und aktiviert wurde. Die Kommunikation beginnt mit dem Betätigen der Schaltfläche „verbinden“, diese wird ersetzt mit der Schaltfläche „trennen“, womit die Kommunikation beendet werden kann.

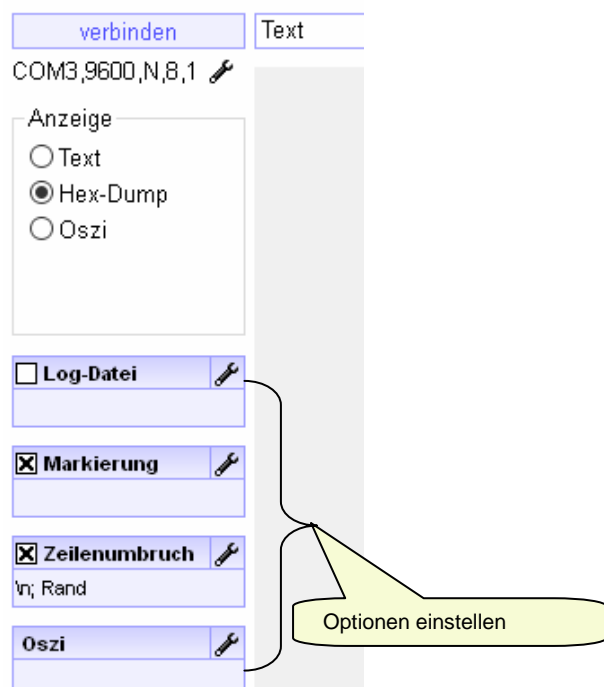
#### Darstellung der empfangenen Daten

Die empfangenen Daten werden fortlaufend im Protokollfenster dargestellt. Der Darstellungsmodus kann während der Kommunikation umgeschaltet werden.

Das SiSy ControlCenter bietet folgende Darstellungsmodi:

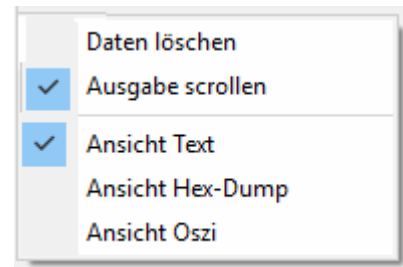
- Text
- Hexadezimal
- Grafik (Oszi)

Für die Darstellungsmodi können über die entsprechenden Schaltflächen Optionen eingestellt werden.



### weitere Funktionen

Für die Darstellungen der Daten im Protokollfenster stehen weitere Funktionen zur Verfügung, z.B. das Löschen der angezeigten Daten. Zu erreichen sind diese Funktionen mit Klick der rechten Maustaste im Protokollfenster, womit ein Fenster zur Auswahl der zusätzlichen Funktionen öffnet. Die Funktionsauswahl ist während der Abarbeitung eines Programms möglich.



### Der Textmodus

Der Textmodus dient zur Visualisierung alphanumerischer Werte im ASCII Format (Zeichenketten). Zahlen, die mit Zeichenketten gesendet werden, müssen vom Mikrocontrollerprogramm zuvor ins ASCII-Format gewandelt werden (siehe *itoa* und *sprintf*). Mit Einstellung der Optionen lassen sich die Ausgaben individuell einstellen.

Über die Schaltfläche „Zeilenumbruch“ lässt sich das Protokollfenster weiter konfigurieren. So kann der Zeilenumbruch z.B. nach einer festgelegten Zeichenanzahl erzwungen werden.

## Optionen

### Zeilenumbruch

Newline-Zeichen ('\n' = 10 = 0x0A)

ganze Zeichenkette:

jedes dieser Zeichen:

nach  Zeichen

am Fensterrand

Über die Schaltfläche „Markierung“ lassen sich im Text festgelegte Zeichen oder Zeichenketten hervorheben.

## Optionen

### Markierung

<input type="checkbox"/>	myAVR	ganze Zeichenkette
<input type="checkbox"/>	e	jedes einzelne Zeichen
<input type="checkbox"/>	'A' 0x0d 10	Bytefolge
<input type="checkbox"/>	\t	jedes einzelne Byte

↓

trennen

COM3,9600,N,8,1

## Der Grafikmodus (Oszi)

Messwerte können auch grafisch visualisiert werden. Dabei werden die Werte fortlaufend und byteweise (Wertebereich 0 bis 255) als Punkte in einem Koordinatensystem dargestellt. Die X-Achse repräsentiert den zeitlichen Verlauf, die Y-Achse den Wertebereich der Daten. Die Anzeige kann als einzelne Punkte oder Linien erfolgen. Die Einstellung der Optionen erfolgt über die Schaltfläche „Oszi“.

## Optionen

### Oszi

Anzeigeart:

- Linie  
 Punkte

Linienstärke und Punktgröße

mittel

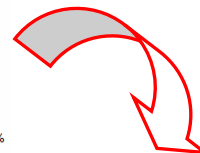
maximale Anzahl der dargestellten Werte: 200



### 6.2.3 Empfangene Daten speichern

Bei der Erfassung von Messdaten die vom Mikrocontroller an den PC gesendet werden ist es oft wichtig, diese in eine Datei zu speichern. Damit wird eine Weiterverarbeitung der Daten in entsprechenden Programmen möglich. Das SiSy ControlCenter ermöglicht es, Protokolldaten aufzuzeichnen (Log-Datei, Rekorderfunktion). Um diese Funktion zu aktivieren, wählen Sie die Schaltfläche „LogDatei“.

In den Optionen muss der Dateiname und Pfad der Log-Datei angegeben werden. Die Weiterverarbeitung der Daten erfolgt entsprechend der Möglichkeiten der Zielanwendung (z.B.: Öffnen, Importieren oder Zwischenablage).



```

Test.txt - Editor
Datei Bearbeiten Format Ansicht ?
Hallo Welt und myAVRHallo Welt und myAVRHallo Welt und myAVR
Hallo Welt und myAVRHallo Welt und myAVRHallo Welt und myAVR
Hallo Welt und myAVRHallo Welt und myAVRHallo Welt und myAVR
Hallo Welt und myAVRHallo Welt und myAVRHallo Welt und myAVR
Hallo Welt und myAVRHallo Welt und myAVRHallo Welt und myAVR
Hallo Welt und mvAVRHallo Welt und mvAVRHallo Welt und mvAVR
  
```

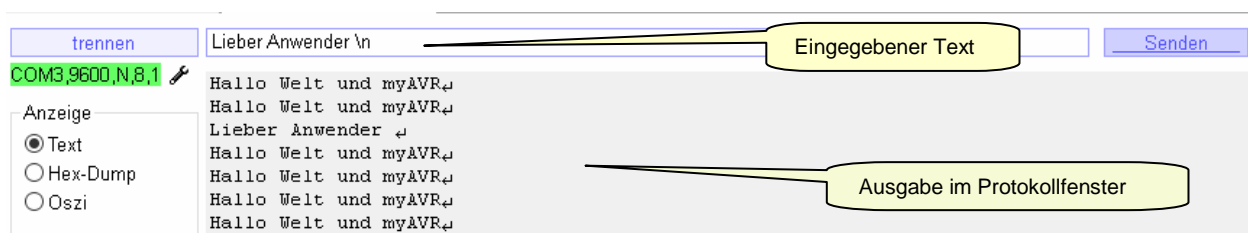
### 6.2.4 Daten an den Controller senden

Über das SiSy ControlCenter können Daten im Text- oder Zahlenformat an den Mikrocontroller gesendet werden. Es können einzelne Zeichen, aber auch Zeichenketten gesendet werden. Die Erfassung der Zeichen bzw. Zeichenkette erfolgt in der Zeile „Text“. Das Senden von Text und Zahlen funktioniert in C-Style:

- \n für Newline (=10=0x0A),
- \r für Return (=13=0x0D),
- \t für Tab (=8),
- \x.. für beliebiges Zeichen als HEX Code (z.B. \x0C für 12=0x0C)

Mit der zugehörigen Schaltfläche „Senden“ wird immer der gesamte Inhalt der Eingabezeile „Text“ gesendet.

In unserem Programmbeispiel auf der folgenden Seite wird die Ausgabe des im Programm hinterlegten Textes (hier „Hallo Welt und myAVR“) einmal mit dem Inhalt der Eingabezeile unterbrochen, sobald die Schaltfläche „Senden“ aktiviert wird.



## Beispielprogramm

Die Daten des folgenden Programmbeispiels sollen als Testdaten dienen.

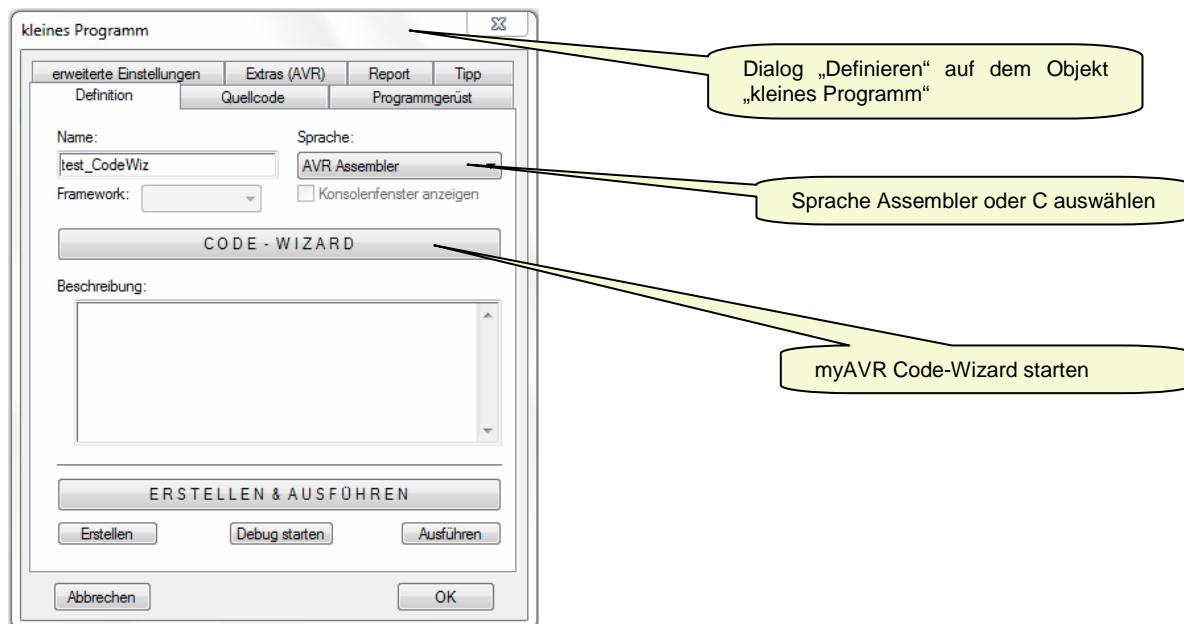
Es wird die Zeichenkette „Hallo Welt und myAVR“ an das SiSy ContolCenter gesendet. Des Weiteren fragt das Programm ab, ob über die Textzeile des SiSy ContolCenters ein Text gesendet wird. Ist das der Fall wird dieser einmal im Protokollfenster ausgegeben.

```
//-----
// Titel      : Datenkommunikation per UART
//-----
// Funktion   : Senden und Empfangen von Daten über UART
//             Entwickelt mit SiSy und PEC (Portable Embedded Classes)
//-----
// Prozessor  : Atmega8 mit 3,6864 MHz
// Sprache    : AVR C++
//-----
class Application : public Controller
{
    // Bausteine und Attribute .....
    protected: uint8_t wert;
    Uart uart;

    // Funktionen .....
    public: void onPower()
    {
        // kleine aber nötige Vorabinitialisierungen
    }
    public: void onStart()
    {
        // hier Initialisierungen durchführen
        wert=0;
        uart.config(9600,UDR);
    }
    public: void onWork()
    {
        if (uart.hasData()) // wenn Daten da sind -> Echo
        {
            char c;
            c = uart.getBytes();
            uart.sendByte(c);
        }
    }
    public: void onTimer10ms() // alle 10 ms
    {
    }
    public: void onTimer100ms() // alle 100 ms
    {
    }
    public: void onTimer1s() // jede Sekunde
    {
        uart.sendString("Hallo Welt und myAVR\n");
    }
    public: void onEvent(const Object& sender, uint8 data)
    {
        // Ereignismeldungen von Appmodulen
        // Sender == Attribut, Data beschreibt das Ereignis
    }
} app; // Anwendungsinstanz
```

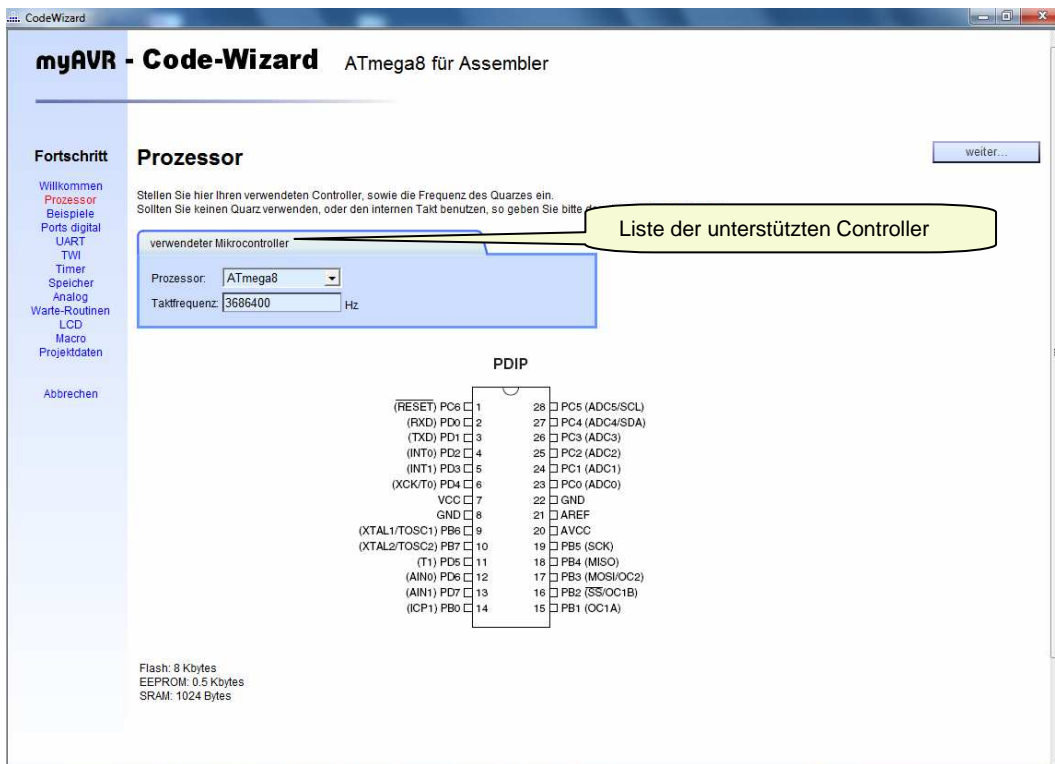
## 6.3 Der myAVR Code-Wizard

Der myAVR Code-Wizard ist ein Assistent zum Erstellen von Assembler- und C-Codes für die Konfiguration und Anwendungsentwicklung von AVR Mikrocontrollern. Dabei wählt der Nutzer Schritt für Schritt im Dialog mit dem Assistenten die gewünschten Konfigurationen und Programmbausteine aus. Der Code-Wizard generiert kompilierfähigen Quellcode in der gewünschten Programmiersprache (AVR Assembler oder AVR C), der als komplette Anwendung geladen wird. Der Entwickler muss nur noch die projektspezifische individuelle Logik ergänzen. Der gesamte Programmrahmen in Form von Hauptprogramm, fertige Initialisierungssequenz, Unterprogramme und Interrupt-routinen wird von Code-Wizard generiert.



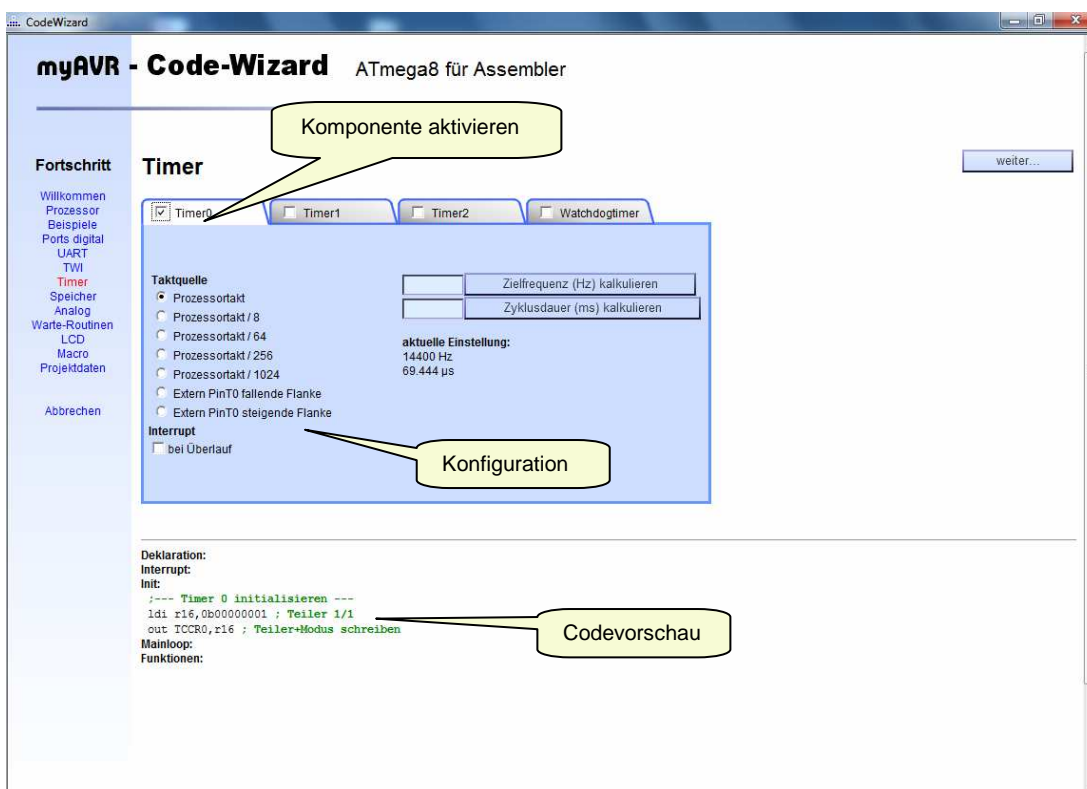
### Grundeinstellungen

Die Zielsprache wurde vor dem Start des Code-Wizards ausgewählt. Für die Generierung von korrektem Quellcode ist es notwendig, den Controllertyp und die Taktfrequenz festzulegen.



### Geräteeinstellungen

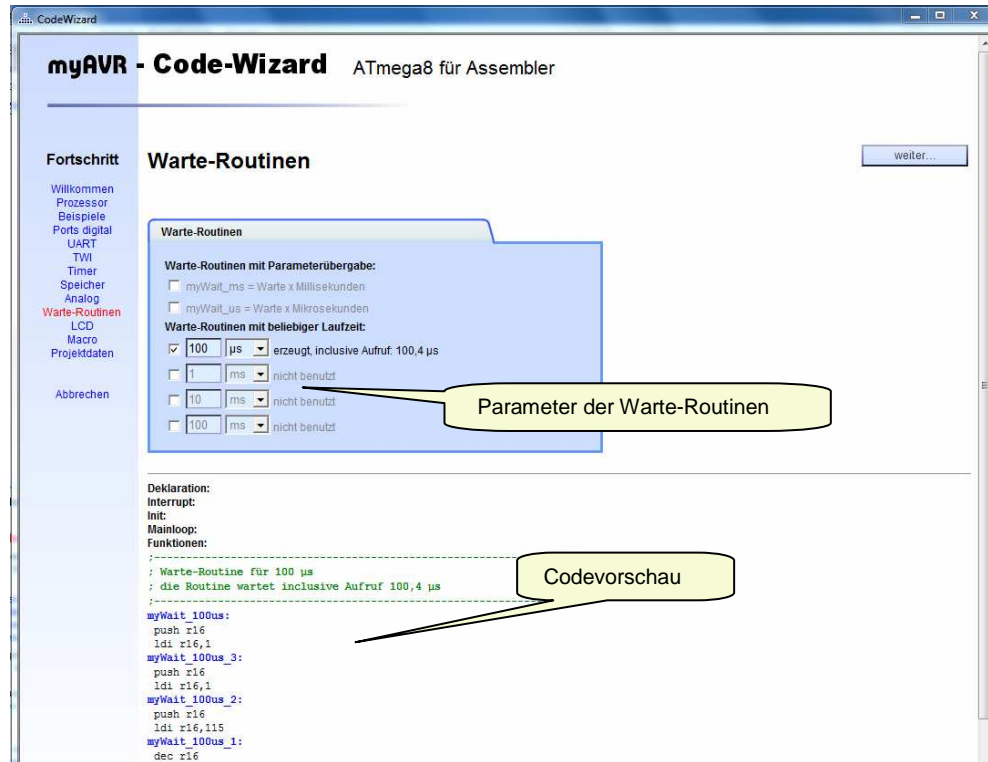
Über den Dialogbereich können Schritt für Schritt die Komponenten des gewählten Controllertyps konfiguriert werden.





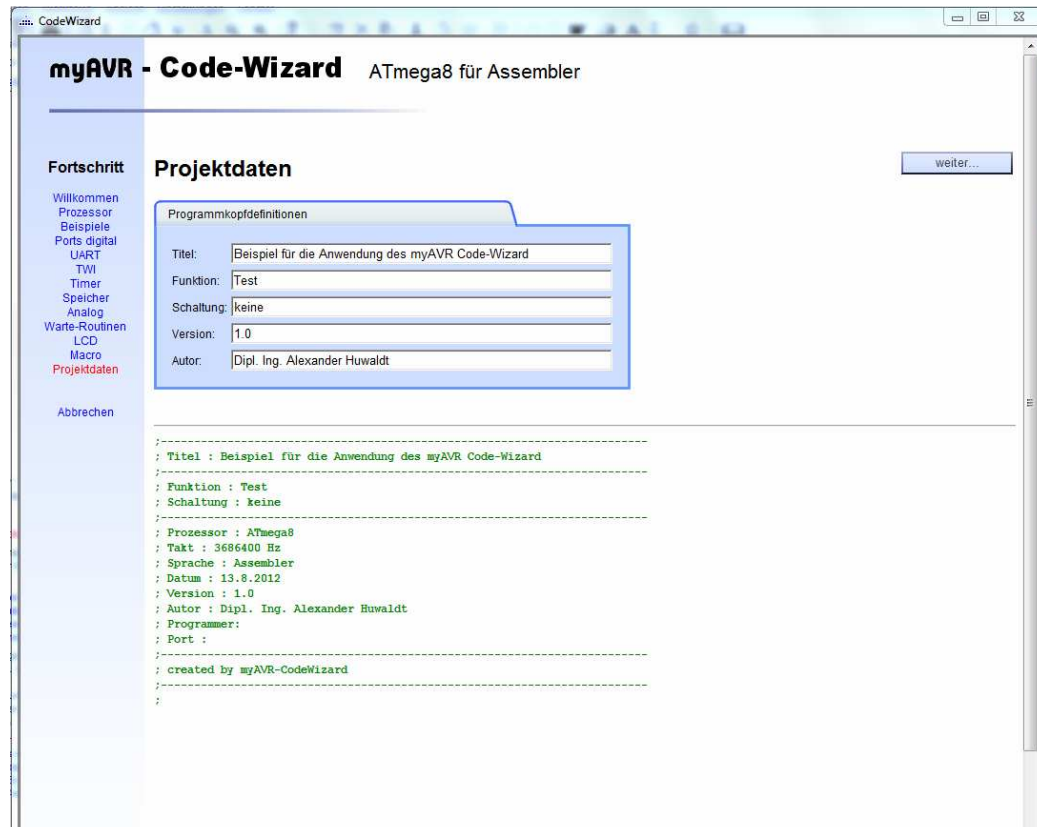
## Unterprogramme

Neben der Möglichkeit die Hardwarekomponenten des gewünschten Controllers zu konfigurieren, bietet der Code-Wizard auch eine Reihe typischer Unterprogramme. So verfügt er über einen Warteroutinen-Rechner zum Generieren präziser Wartefunktionen.



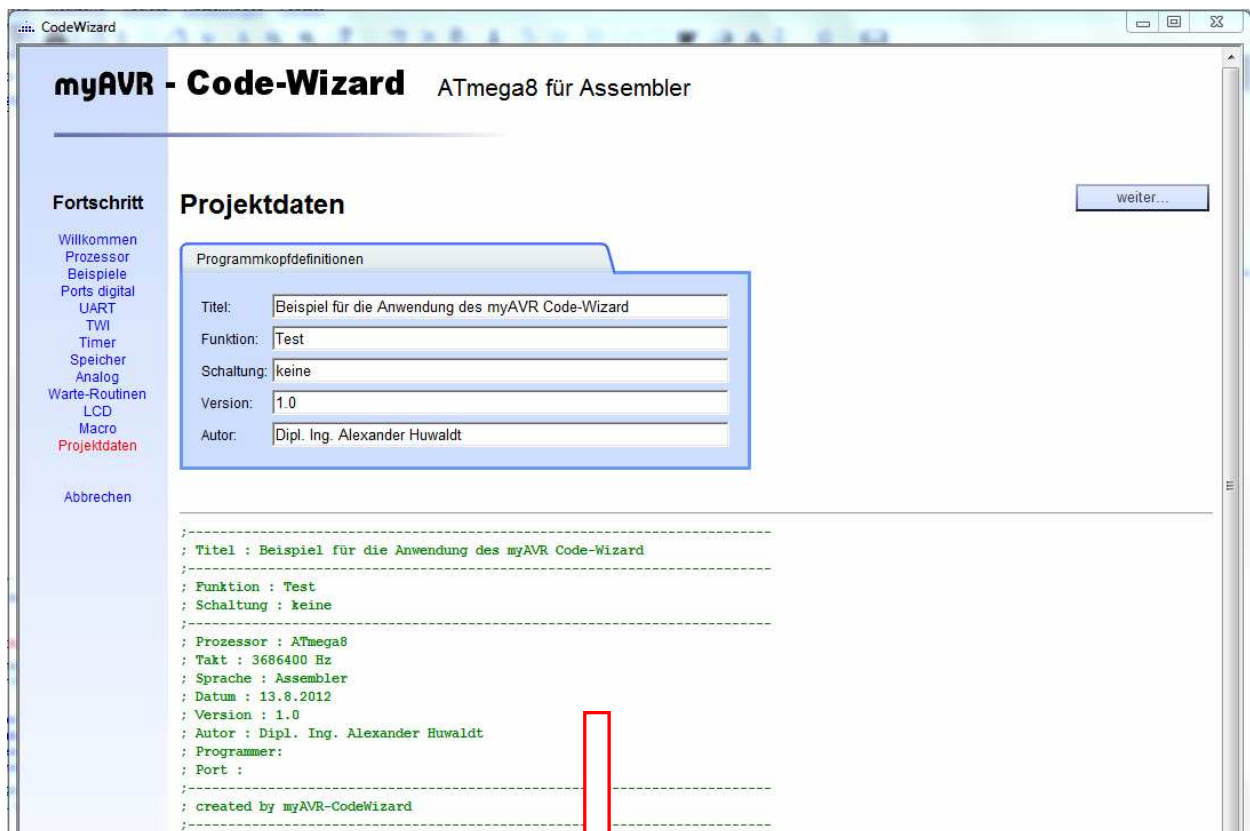
## Projektdaten

Der letzte Punkt im Code-Wizard ist die Eingabe der Projektdaten. Aus diesen wird eine Programmkopfdokumentation generiert und dem kompletten Quellcode vorangestellt.



## Codegenerierung

Der vollständige Quellcode wird zur Kontrolle dem Anwender angezeigt. Es können jetzt noch Änderungen vorgenommen werden, indem man die betreffenden Punkte im Navigationsbereich auswählt und die Parameter ändert. Mit Bestätigung des Quellcodes wird dieser als komplettes Programm eingefügt.



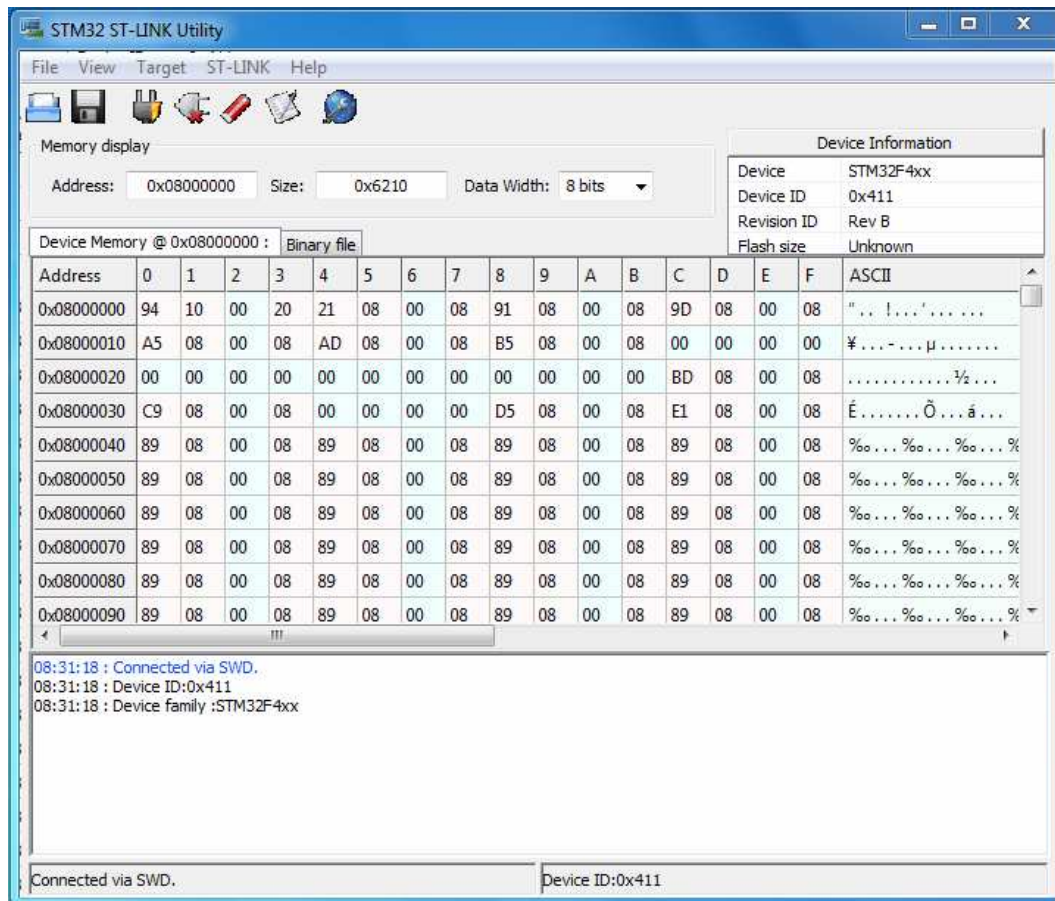
```
//-----
// Titel : Test der seriellen Verbindung
//-----
// Funktion : empfängt und sendet Daten per UART (9600,8,n,1)
// Schaltung : Nullmodemkabel anschließen
//-----
// Prozessor : ATmega8
// Takt : 3686400 Hz
// Sprache : C
// Datum : 20.1.2011
// Version : 1.0
// Autor : Dipl. Ing. Päd. Huwaldt
// Programmier:
// Port :
//-----
// created by myAVR-Code-Wizard
//-----
#define F_CPU 3686400
#include <avr\io.h>
#include <avr\interrupt.h>
//-----
// Initialisierungen
//-----
void init()
{
    // Ports initialisieren
    ...
}
```

## 6.4 Das STM32 ST-Link Utility

ST-Link ist ein Werkzeug zum Programmieren von ARM-Mikrocontrollern. Damit können Sie auch den Programmspeicher komfortabel programmieren.

### Brennen

Der Inhalt der ausgewählten Dateien und des Speichers des ARM-Mikrocontrollers wird im mittleren Bereich des Fensters angezeigt.



### Ausgabe

Die Ausgabe befindet sich im unteren Teil des ST-Link Fensters, die Aktionen werden im Ausgabefenster protokolliert. Es werden Informationen und Dauer der Aktion angezeigt. Zustand und Ergebnis werden durch Signalfarben und Ausgaben dargestellt.

### Signalfarben

- rote Schrift: keine Verbindung
- blaue Schrift: Zustandsinformation
- schwarze Schrift: verbundener Controller
- grüne Schrift: erfolgreiches Brennen

## Speicher anzeigen und ändern

Neben dem Geräte-Informationsbereich enthält das Hauptfenster 2 weitere Bereiche:

- Memory Display
- Memory Data

### Memory Display:

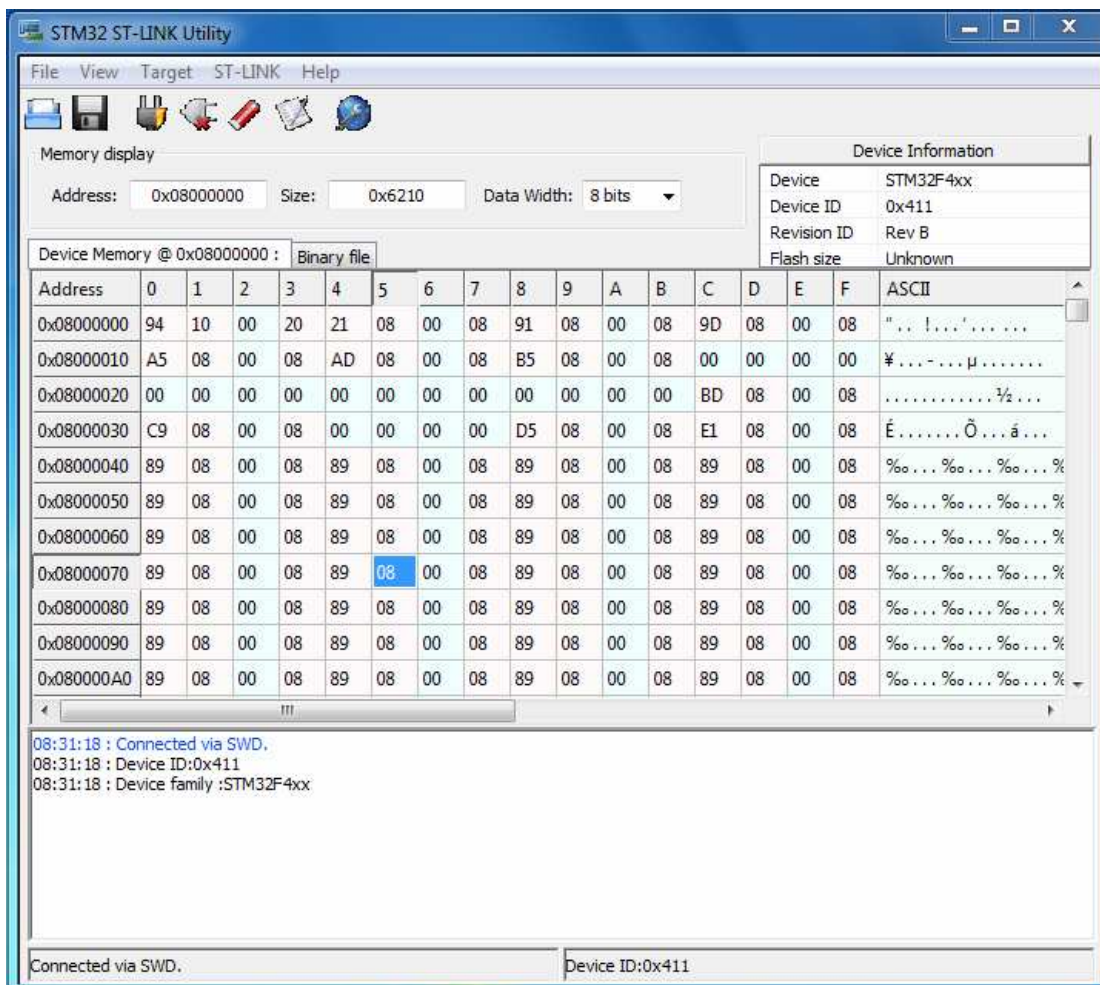
Dieser Bereich enthält drei Eingabefelder:

- *Address*: Startadresse des Speichers, von der Sie lesen wollen.
- *Size*: Anzahl der zu lesenden Daten in Byte (Hexadezimalformat).
- *Data Width*: Breite der angezeigten Daten (8-bit, 16-bit oder 32-bit).

### Memory Data:

In diesem Bereich werden die Daten aus einer binären Datei oder aus dem Speicher eines angeschlossenen Gerätes ausgelesen. Sie können den Inhalt der Datei vor dem Brennen ändern.

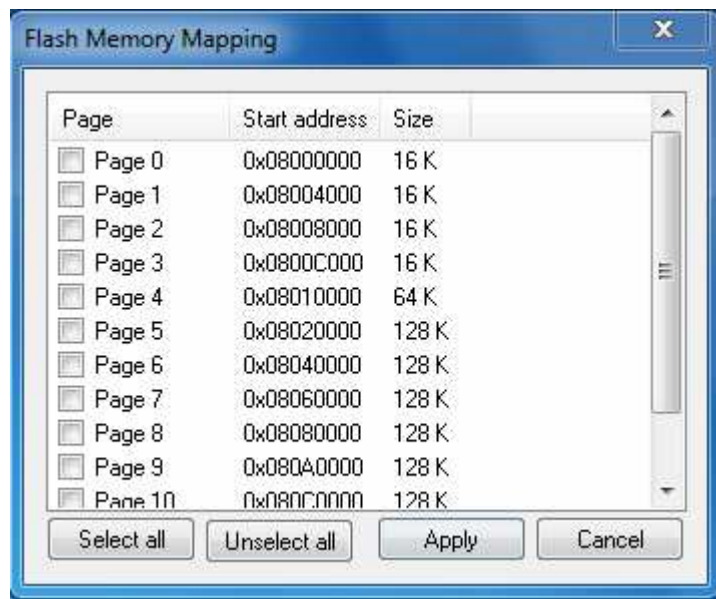
- Um den Inhalt der binären Datei anzeigen zu lassen, klicken Sie auf *File/Open file ...*
- Um den Speicherinhalt eines verbundenen Geräts auszulesen und anzeigen zu lassen, geben Sie die Startadresse, Datengröße und die Datenbreite in die Felder ein und bestätigen dann mit „Enter“.
- Nach dem Lesen von Daten können Sie auch jeden Wert verändern, wenn Sie auf den Wert doppelklicken. Die betroffene Zelle wird markiert.
- Sie können auch den Speicherinhalt in eine binäre Datei über das Menü *File/ Save file as...* abspeichern.



**Flash-Speicher löschen:**

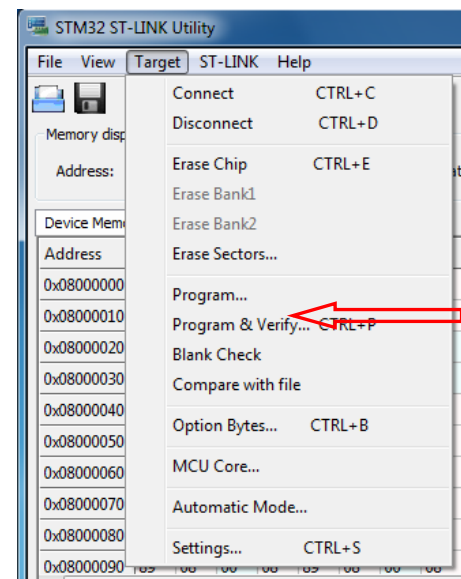
Es gibt zwei Vorgehensweisen den Flash-Speicher zu löschen:

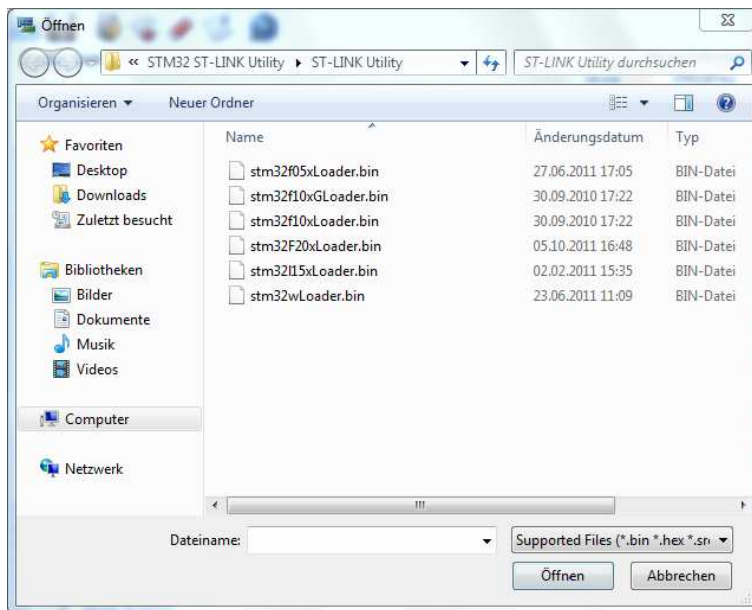
1. Gesamten Flash löschen:  
Löschen Sie alle Flash-Speichersektoren des angeschlossenen Gerätes. Das wird über das Menü *Target/Erase Chip* ausgeführt.
2. Sektorweise den Flash löschen:  
Zur Auswahl der Sektoren klicken Sie im Menü *Target* auf *Erase Sectors...*. Das "Flash Memory Mapping" Fenster zeigt die ausgewählten Sektoren des Flashs an.
  - „Select all“ markiert alle Sektoren
  - „Deselect all“ macht die Markierung rückgängig
  - „Cancel“ bricht den Löschvorgang ab
  - „Apply“ löscht alle markierten Sektoren

**Programm brennen:**

Das STM32 ST-Link kann \*.bin-, \*.hex-, oder \*.srec-Dateien in den Flash oder RAM-Speicher brennen. Um das zu tun, führen Sie diese Schritte aus:

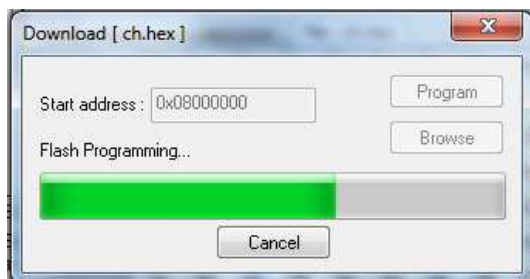
1. Klicken Sie auf *Target/Programm...* (oder *Target/Programm & verify...*, wenn nach dem Brennen die Daten überprüft werden sollen).
2. Es öffnet das „Open file“ Dialogfenster, in diesem können Sie die Datei auswählen und öffnen.





**Abbildung 35: Anzeige der verfügbaren Dateien**

3. In dem „Device programming“ Dialogfenster geben Sie die Startadresse ein, ab welcher das Programm gebrannt werden soll, es kann eine Flash- oder RAM-Adresse sein
4. Zum Schluss klicken Sie auf die Schaltfläche „Program“ und das Programm wird gebrannt. Wenn Sie *Target/Programm & verify...* ausgewählt haben, wird das Programm nach dem Brennen überprüft.

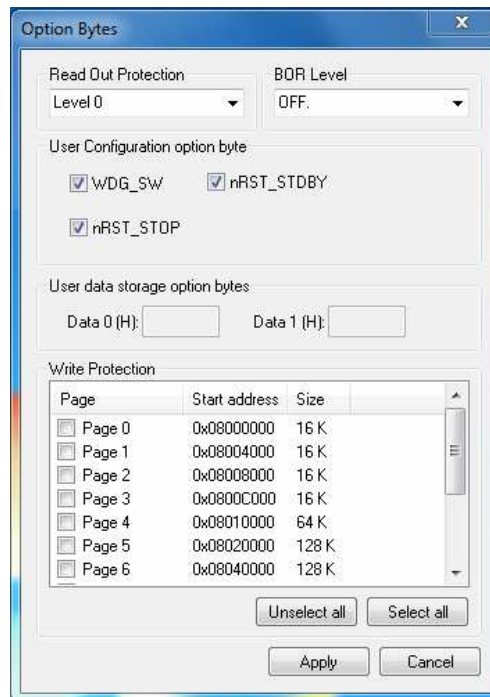


#### Hinweise:

1. Die STM32F2- und STM32F4-Reihe haben unterschiedliche Programmierverfahren, die von der Spannungsversorgung der MCU abhängt. ST-Link verwendet die standardmäßige MCU-Spannungsversorgung. Unter dem Menü *Target/Settings...* kann die Spannungsversorgung geändert werden.
2. Wenn das Gerät schreibgeschützt gelesen wird, dann wird der Schreibschutz deaktiviert. Wenn einige Flash-Sektoren schreibgeschützt sind, dann wird der Schreibschutz während der Programmierung deaktiviert und nach der Programmierung wieder aktiviert.

### Option-Bytes-Konfiguration

Das STM32 ST-Link kann alle Option-Bytes über die „Option Bytes“ Dialogbox ändern, das über *Target/Option Bytes...* geöffnet werden kann.



Die „Option Bytes“ Dialogbox enthält die folgenden Einstellungsmöglichkeiten:

1. „*Read Out Protection*“: Verändert den Leseschutz des Flash-Speichers. Für STM32F2, STM32F4 und STM32L1 Geräte sind folgende Schutzniveaus verfügbar:
  - Level 0: kein Leseschutz
  - Level 1: Flash Leseschutz aktiviert
  - Level 2: Flash Leseschutz und alle Debug-Eigenschaften deaktiviert
 Für die anderen Geräte kann der Leseschutz nur aktiviert oder deaktiviert werden.
  
2. „*BOR Level*“ (Brown Out Reset-Pegel):
 

Diese Liste enthält unterschiedliche Spannungseinstellungen, beim Unterschreiten der Spannungsgrenze wird das Gerät in den Reset-Modus gesetzt. Diese Auswahl ist nur für das angeschlossene STM32 L1 Gerät gültig. Für Geräte mit niedriger Spannungsversorgung kann aus 5 VBOR Einstellungen ausgewählt werden:

  - BOR LEVEL 1:  
Reset-Schwelle im Spannungsbereich von 1,69 bis 1,8 V
  - BOR LEVEL 2:  
Reset-Schwelle im Spannungsbereich von 1,94 bis 2,1 V
  - BOR LEVEL 3:  
Reset-Schwelle im Spannungsbereich von 2,3 bis 2,49 V
  - BOR LEVEL 4:  
Reset-Schwelle im Spannungsbereich von 2,54 bis 2,74 V
  - BOR LEVEL 5:  
Reset-Schwelle im Spannungsbereich von 2,77 bis 3,0 V

Für STM32F2 und STM32F4 Geräte können 4 programmierbare VBOR Einstellungen ausgewählt werden:

- BOR LEVEL 3:  
Spannungsversorgung im Bereich von 2,70 bis 3,60 V
- BOR LEVEL 2:  
Spannungsversorgung im Bereich von 2,40 bis 2,70 V
- BOR LEVEL 1:  
Spannungsversorgung im Bereich von 2,10 bis 2,40 V
- BOR off:  
Spannungsversorgung im Bereich von 1,62 bis 2,10 V

3. „*User Configuration option byte*“:

- WDG\_SW:  
Wenn aktiviert, muss der Watchdog durch die Software aktiviert werden, sonst wird er automatisch beim Einschalten aktiviert.
- nRST\_STOP:  
Wenn nicht aktiviert, wird der Reset im Standby-Modus generiert. Wenn aktiviert, wird kein Reset generiert beim Eintreten in den Standby-Modus.
- nRST\_STDBY:  
Wenn nicht aktiviert, wird ein Reset erzeugt beim Betreten des Stop-Modus (alle Uhren sind gestoppt). Wenn aktiviert, wird kein Reset generiert beim Betreten des Stop-Modus.
- BFB2:  
Wenn nicht aktiviert und die Boot-Pins gesetzt sind, um das Gerät beim Start vom Anwender-Flash zu booten, startet das Gerät von der „Flash bank“ 2, sonst von der „Flash bank“ 1. Diese Option ist nur aktiviert, wenn das Gerät zwei Flash-Banks besitzt.

4. „*User data storage option bytes*“:

Enthält 2 Byte für den „User storage“. Diese Bytes sind nicht verfügbar auf STM32F2-, STM32F4- und STM32L1-Geräten

5. „*Write Protection*“:

Die Flash-Sektoren sind geräteabhängig gruppiert. Hier kann der Schreibschutz für die einzelnen Sektoren aktiviert werden.



## 6.5 Das myAVR ProgTool

**Hinweis:**

Das myAVR ProgTool ist nur in SiSy Ausgaben mit dem Add-On AVR verfügbar.

### 6.5.1 Übersicht zum myAVR ProgTool

Das myAVR ProgTool ist ein Werkzeug zum Programmieren von AVR-Mikrocontrollern. Sie können den Programmspeicher, den EEPROM und die Fuse-/Lock-Bits der unterstützten AVR-Mikrocontroller komfortabel programmieren.

**Wichtig:**

Es muss stets die verwendete Hardware eingestellt sein (Registerkarte „Hardware“).

#### Hardware einstellen

Wählen Sie vor dem Brennen immer die korrekte Programmierhardware und den zu programmierenden Controllertyp aus. Die Einstellungen werden beim Schließen gespeichert und sind nach erneutem Öffnen wieder verfügbar. Mit "?" kann der aktuelle Controller und der verwendete Port ermittelt werden.

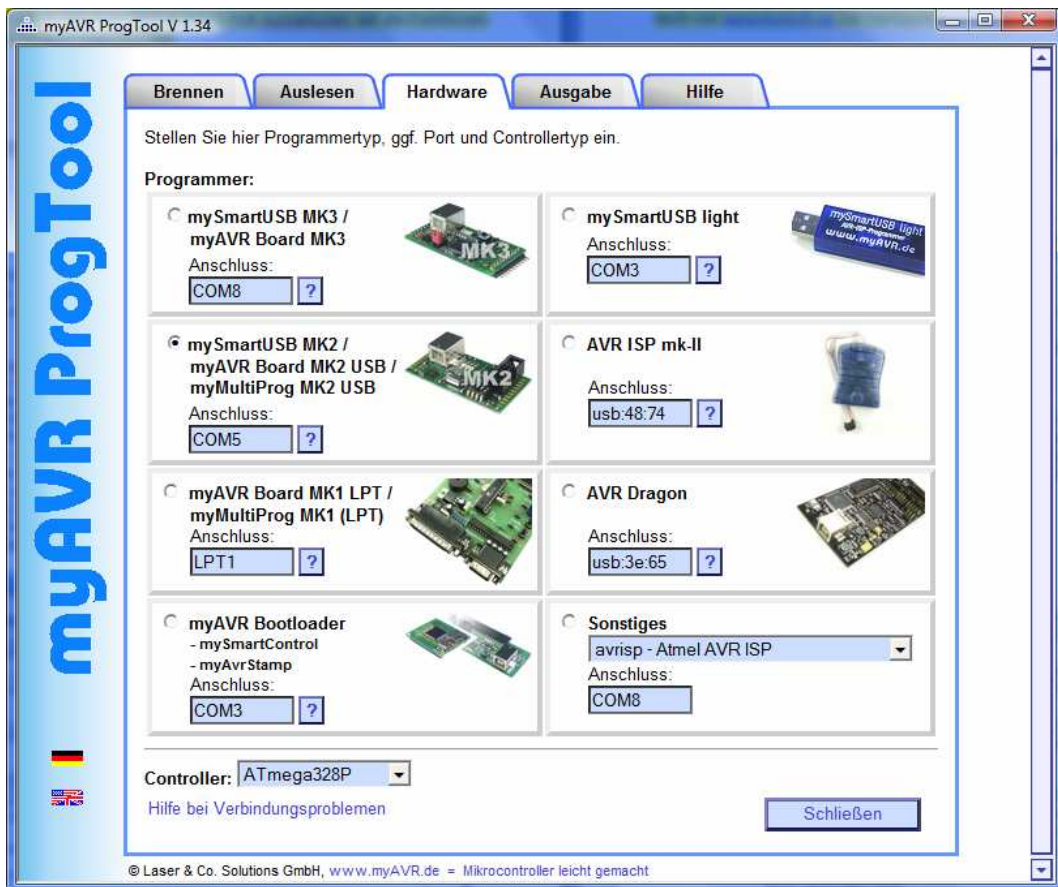


Abbildung 36: Registerkarte "Hardware" im myAVR ProgTool

**Signalfarben:**

gelb: Test läuft

grün: Test war erfolgreich

rot: Test ist fehlgeschlagen

**Info:**

USB-Treiber: 5.4.29.0  
 Geräte-ID: mySmartUSB2-0001  
 Port: COM5  
 Firmware: V2.5  
 Controller: ATmega8

[schließen](#)

**Info:**

USB-Treiber: 5.4.29.0  
 Am angegebenen Port ist zur Zeit kein Gerät angesteckt.  
 Ein Test ist deshalb nicht möglich.  
 Möglicher Port:  
 COM8

[schließen](#)

**Besonderheiten:**

- mySmartUSB, myAVR Board, myMultiProg : → MK2, MK3, light / mySmartControl MK2: → 8K, 16K, 32K / mySmartControl MK3
  - USB-Treiber muss installiert sein (CP210X)
  - mySmartUSB muss angeschlossen sein
  - virtueller COM-Port (COMx) muss aktiv sein
  - Port-Einstellungen sind nicht notwendig, da Programmer automatisch gefunden wird
  - Ziel-Hardware wird vom mySmartUSB mit Spannung versorgt
  - Über "?" kann der verwendete Controller ermittelt werden
  
- JTAG ICE mk-II:
  - LIB-USB-Treiber muss installiert sein
  - JTAG ICE mk-II muss angeschlossen sein
  - Spannungsversorgung der Ziel-Hardware muss eingeschaltet sein
  - korrekte USB Port-Nummer muss im Feld "Anschluss:" eingetragen sein
  - Anschluss und Controller können per "?" ermittelt und überprüft werden
  
- STK 500:
  - Board muss an einem COM-Port angeschlossen sein
  - externe Spannungsversorgung muss angeschlossen sein
  - Board muss eingeschaltet sein (Power)
  - ISP-Pin muss verbunden sein
  - korrekter COM-Port muss im Feld "Anschluss:" eingetragen sein
  - Anschluss wird nicht automatisch ermittelt
  
- AVR ISP mk-II:
  - LIB USB-Treiber muss installiert sein
  - AVR ISP mk-II muss angeschlossen sein
  - Spannungsversorgung der Ziel-Hardware muss eingeschaltet sein
  - korrekte USB Port-Nummer muss im Feld "Anschluss:" eingetragen sein
  - Anschluss und Controller können per "?" ermittelt und überprüft werden
  
- AVR Dragon:
  - LIB USB-Treiber muss installiert sein
  - AVR Dragon muss angeschlossen sein
  - Spannungsversorgung der Ziel-Hardware muss eingeschaltet sein
  - korrekte USB Port-Nummer muss im Feld "Anschluss:" eingetragen sein
  - Anschluss und Controller können per "?" ermittelt und überprüft werden

## Brennen

Der Inhalt der ausgewählten Dateien wird im unteren Bereich des Fensters angezeigt. Die Inhalte der Eingabefelder bleiben nach dem Schließen erhalten und sind nach erneutem Öffnen des Tools wieder verfügbar.

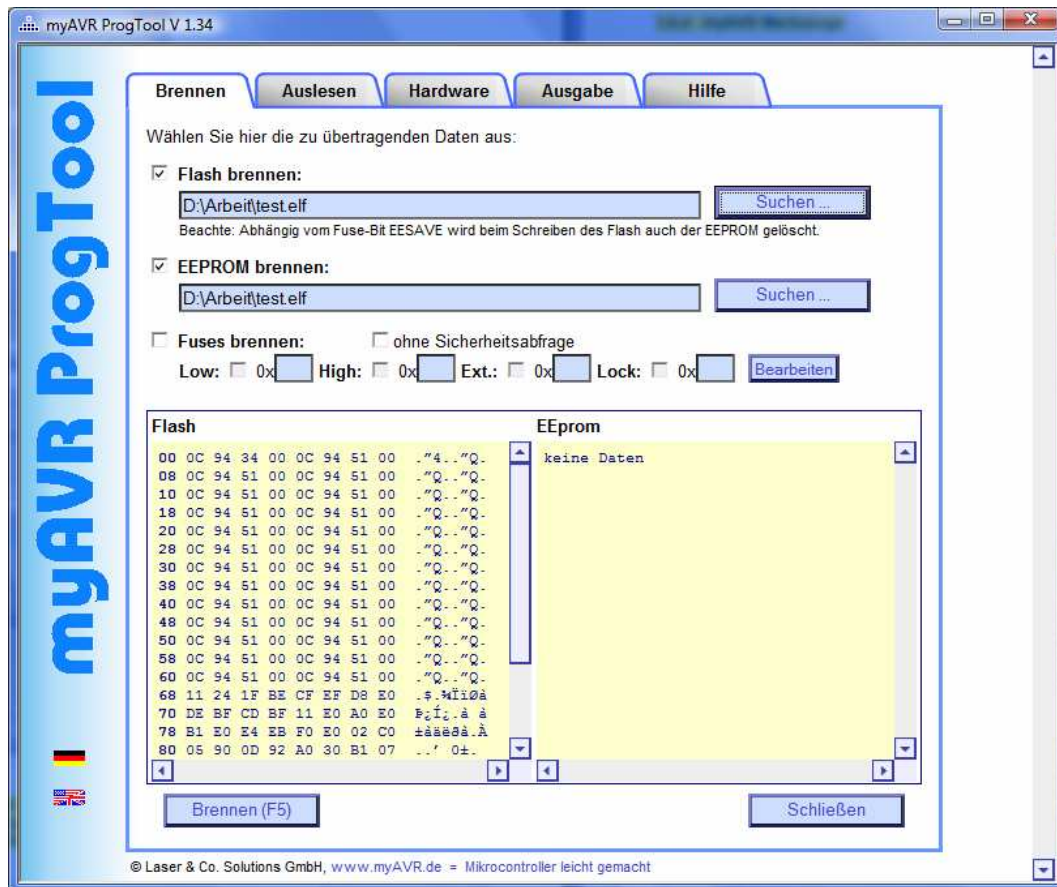


Abbildung 37: Registerkarte „Brennen“ im myAVR ProgTool

### Signalfarben:

*rötlich* gefülltes Datei-Eingabefeld: Datei existiert nicht

*bläulich* gefülltes Datei-Eingabefeld: Datei existiert

*graue Schrift*: Element inaktiv

### Flash brennen

Aktivieren Sie zunächst die Auswahl "Flash brennen". Wählen Sie anschließend per Schaltfläche "Suchen..." die gewünschte Datei (\*.elf, \*.hex, \*.raw, \*.bin) aus. Der Dateiname kann ebenso per Hand in das Eingabefeld eingegeben werden.

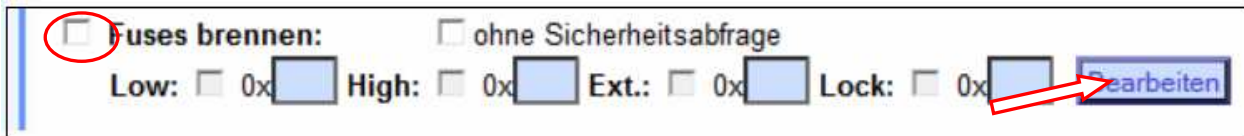
Bei Auswahl einer \*.elf-Datei, wie sie zum Beispiel mit WinAVR erstellt wird, erfolgt eine automatische Konvertierung in das Intel-HEX-Format. Sind in der ELF-Datei die Sektionen ".eeprom", ".fuses" bzw. ".lock" enthalten, werden diese extrahiert und als Werte für EEPROM und die Fuses verwendet.

### EEPROM brennen

Aktivieren Sie zunächst die Auswahl "EEPROM brennen". Wählen Sie anschließend per Schaltfläche "Suchen..." die gewünschte Datei (\*.eep, \*.elf, \*.hex, \*.raw, \*.bin, \*.txt) aus. Der Dateiname kann ebenso per Hand in das Eingabefeld eingegeben werden.

### Fuses brennen

Aktivieren Sie zunächst die Auswahl "Fuses brennen" (auf der Seite "Brennen") sowie die gewünschten Fuses (Low-, High- und Extended Fuse sowie Lock-Bits).



Danach öffnen Sie die Fuse- und Lock-Bit Konfiguration über "Bearbeiten". Tragen Sie die hexadezimalen Werte entsprechend der Datenblätter des zu programmierenden Mikrocontrollers ein. Sie können die Werte auch über die Schaltfläche "Bearbeiten" einstellen.

### Fuse- und Lock-Bits

Sie erreichen die Fuse- und Lock-Bit Konfiguration über "Bearbeiten" auf der Registerkarte "Brennen".

Sie können die Fuse- und Lock-Bits in einer Listenansicht nach Ihren Bedürfnissen konfigurieren. Dazu muss vorher in der Registerkarte "Hardware" der korrekte Controller ausgewählt sein. Wollen Sie die aktuellen Werte auslesen oder verändern, müssen Sie auch den richtigen Programmer einstellen und anschließen.

Beachten Sie, dass bei fehlerhaften Einstellungen der Fuse- und Lock-Bits der Mikrocontroller unter Umständen nicht mehr programmierbar oder gar unerreichbar sein kann.

Weitere Informationen zu den Fuse- und Lock-Bit finden Sie im Kapitel 6.5.2

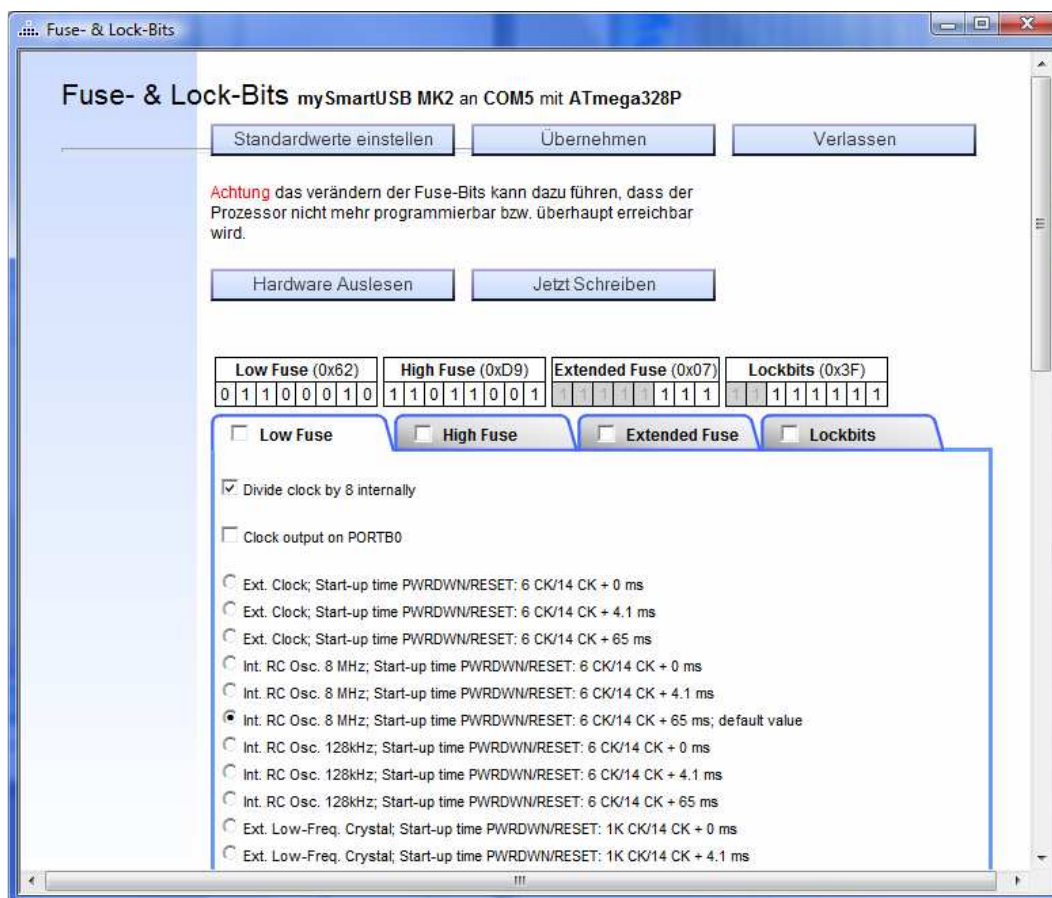


Abbildung 38: Konfiguration Fuse- und Lock-Bits

## Auslesen eines Controllers

Zum Auslesen muss die richtige Hardware (Controller und Programmer) eingestellt sein. Die aktuellen Einstellungen sehen Sie im Kopfteil der Seite.

Lesen Sie die Daten des Flash- oder EEPROM-Speichers durch Betätigen der jeweiligen Schaltfläche aus. Dieser Vorgang kann je nach Speichergröße länger dauern. Es wird der komplette Speicher ausgelesen, haben Sie also etwas Geduld.

Nach erfolgreichem Auslesen werden die gelesenen Daten dargestellt. Nicht belegte Speicherbereiche am Ende der Daten werden vor der Darstellung entfernt.

Sie können die ausgelesenen Daten in eine Datei speichern. Die Speicherung erfolgt im Intel-HEX-Format.

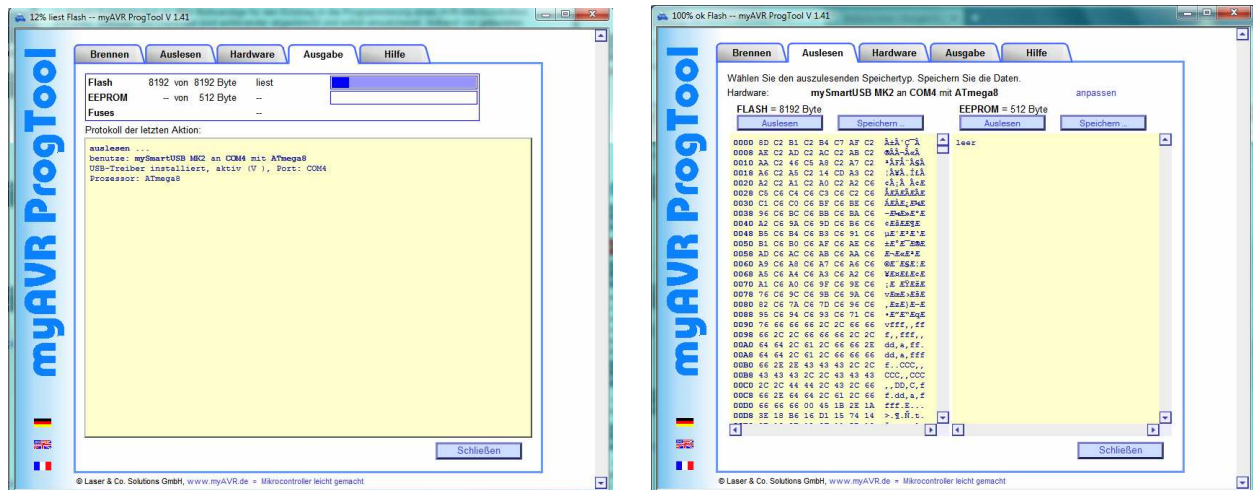
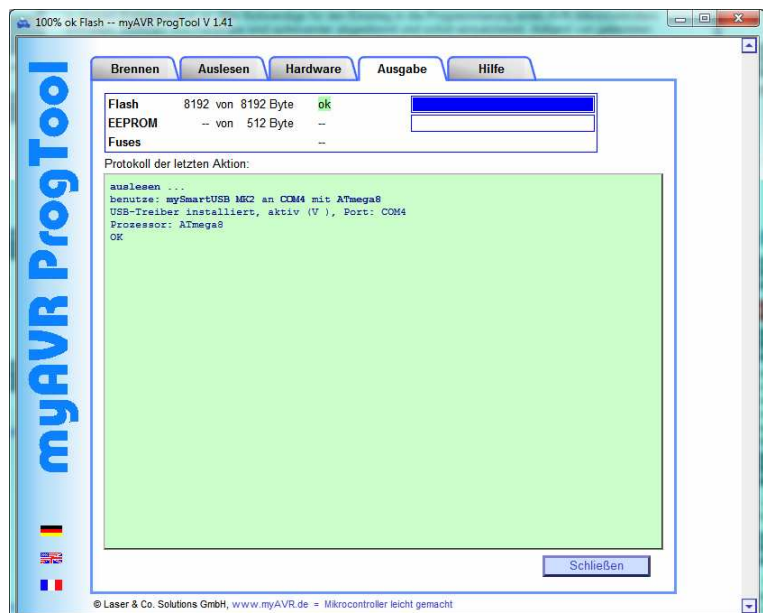


Abbildung 39: Beispiel für FLASH-Auslesen

## Ausgabe

Die jeweils letzte Aktion wird im Ausgabefenster protokolliert. Neben anderen Informationen werden Anzahl der übertragenen Bytes und Dauer der Aktion angezeigt. Zustand und Ergebnis der Aktion wird durch Signalfarben und Ausgaben angezeigt.



## Signalfarben:

*hellgrauer* Hintergrund: keine Aktion, wartend

*hellgelber* Hintergrund: Aktion wird momentan ausgeführt

*hellroter* Hintergrund: Aktion wurde nicht erfolgreich ausgeführt, Fehler

*hellgrüner* Hintergrund: Aktion wurde erfolgreich beendet

**Problembehandlung:**

- Allgemein: Überprüfen Sie, ob alle Kabel richtig angeschlossen sind und überprüfen Sie, ob die Spannungsversorgung der Zielplattform ausreichend ist.
- Bei USB-Verbindungen trennen Sie diese kurz (ca. 20s) um eine Reinitialisierung des USB-Treibers durchzuführen.
- mySmartUSB funktioniert nicht:
  - USB-Treiber (CP210x) nicht installiert -> Treiber installieren
  - mySmartUSB im Datenmodus -> in Programmiermodus schalten (DIP)
- mySmartControl funktioniert nicht:
  - USB-Treiber (CP210x) nicht installiert -> Treiber installieren
  - kein Bootloader vorhanden -> Bootloader nachrüsten

**6.5.2 Einstellungen Fuse- und Lock-Bits für AVR Produkte****Einleitung**

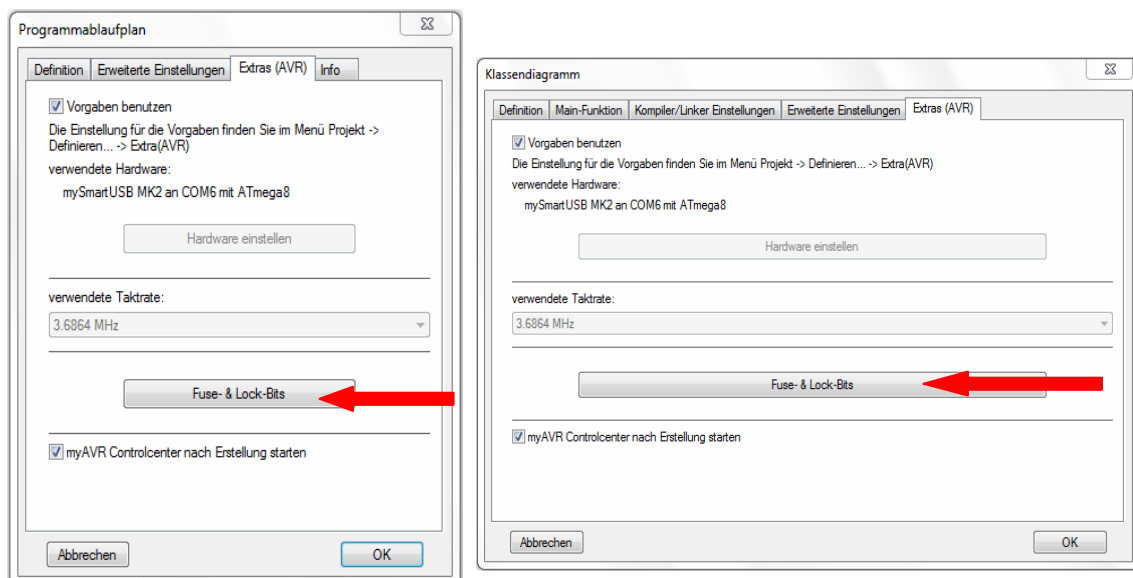
Fuse- und Lock-Bits (engl. fuse = Sicherung, engl. lock = Schloss) nennt man die Bits in bestimmten Registern des AVR zum Konfigurieren des Controllers. Die Fuse-Bits müssen über ein entsprechendes Interface (Software) eingestellt werden. Der normale Programmiermodus verändert die Fuse- und Lock-Bits nicht. Je nach Controllertyp sind unterschiedliche Fuse- und Lock-Bits verfügbar. Die verbindliche und exakte Beschreibung findet man im jeweiligen Datenblatt des Controllers.

Das falsche Setzen der Fuse- und Lock-Bits zählt zu den häufigsten Problemen bei der Programmierung von AVR-Controllern, daher sollte hier mit Umsicht vorgegangen werden. Das Verändern der Fuse-Bits sollte man nicht als Anfänger vornehmen. Das geöffnete Datenblatt zur Überprüfung der Konfiguration ist das wichtigste Instrument, um Fehler zu vermeiden.

**Fuse- und Lock-Bits, Benutzeroberfläche in SiSy AVR**

Sie erreichen die Benutzeroberfläche zum Auslesen, Verändern und Programmieren der Fuse- und Lock-Bits in SiSy wie folgt:

1. Menüfolge *Werkzeuge/myAVR ProgTool*  
→ Registerkarte „Brennen“ → Bearbeiten → Fuse- und Lock-Bits
2. Auf Programm-Objekten wie „kleines Programm“, „Programm“, „PAP“ usw. :  
Kontextmenü → Definieren → Registerkarte „Extras (AVR)“ → Schaltfläche „Fuse- & Lock-Bits“.



Beim Start der Benutzeroberfläche für Fuse- und Lock-Bits wird eine Verbindung zum Controller aufgebaut, der Controllertyp ermittelt, die Definitionen der Fuse- und Lock-Bits des Controllers geladen und die Einstellungen der Fuse- und Lock-Bits des Controllers ausgelesen und angezeigt. Dieser Vorgang kann je nach Controllertyp und Verbindung einige Sekunden dauern. Die Verbindung zum Controller wird, solange diese Benutzeroberfläche offen ist, dauerhaft offen gehalten.

Die Benutzeroberfläche passt sich dem ermittelten Controller und den dazugehörigen Definitionsdaten automatisch an. Es werden immer nur die Optionen angezeigt, die zum ermittelten Controller gehören. Vergleichen Sie dazu immer das betreffende Datenblatt. Es ist nicht zulässig, den Programmer oder den Controller während der Sitzung zu entfernen oder zu wechseln. Dazu ist dieses Fenster zu schließen, danach der Controller oder Programmer zu wechseln und die Benutzeroberfläche erneut zu starten.

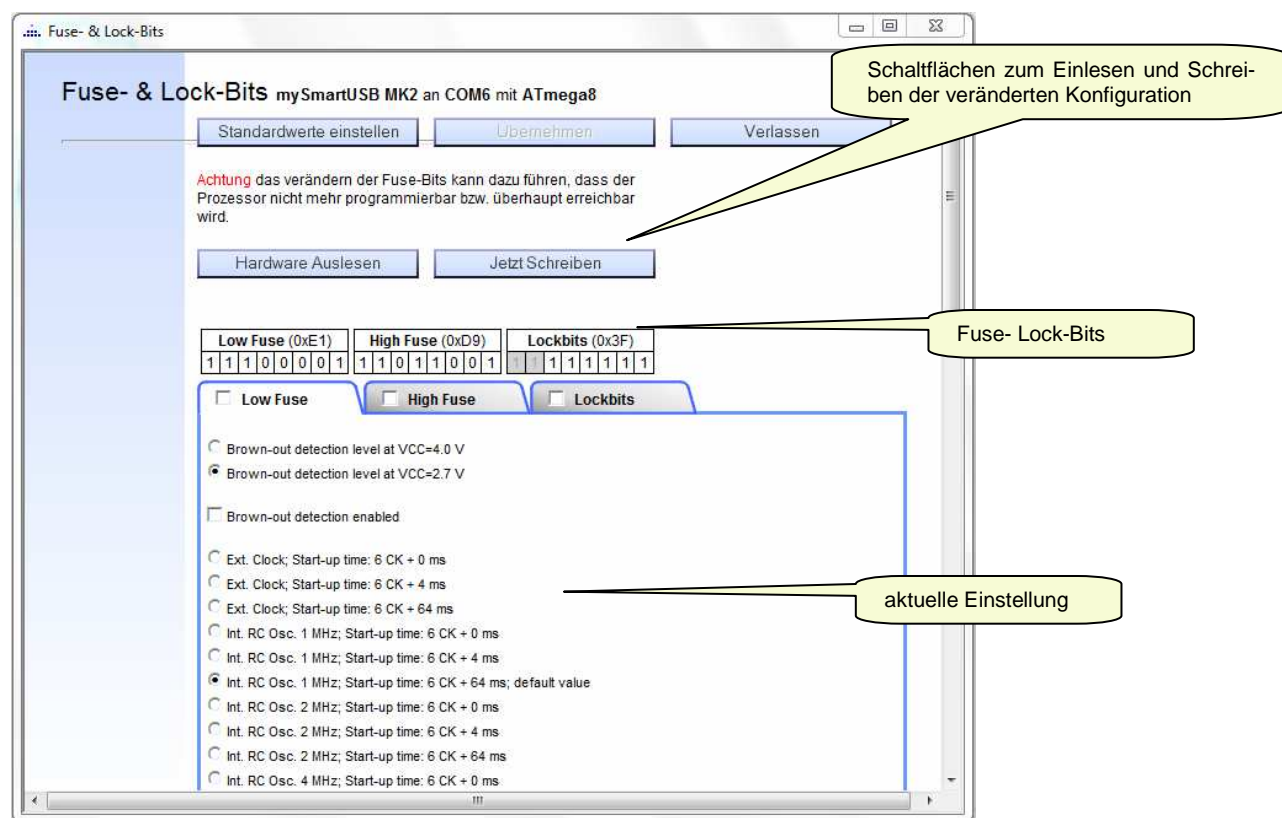


Abbildung 40: Fuse- und Lock-Bits des ATmega8

### Fuse- und Lock-Bits verändern

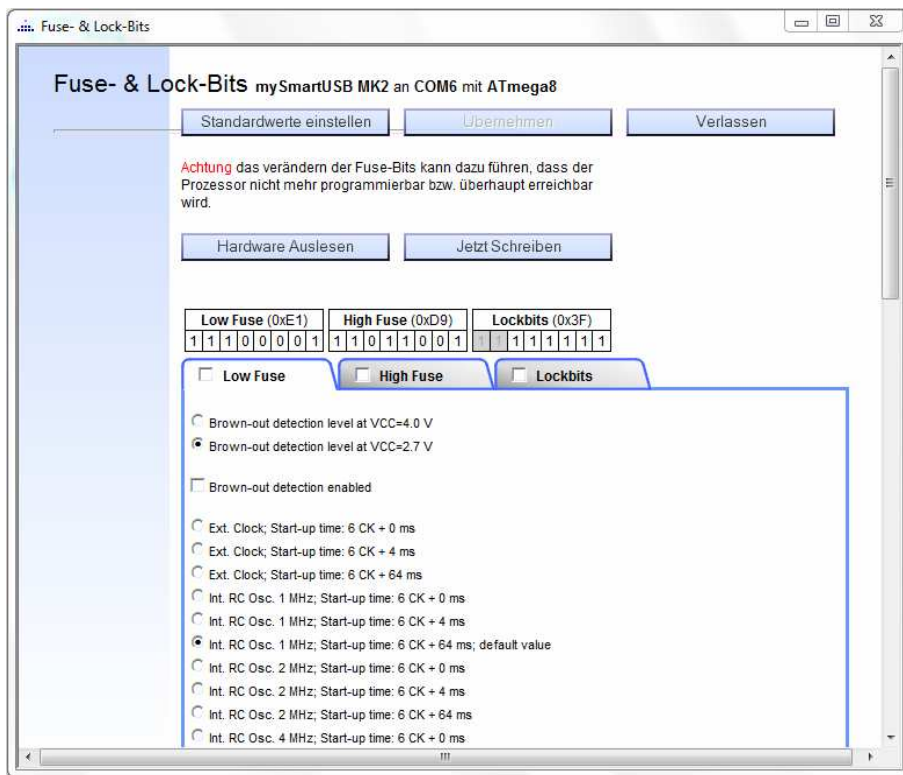
Zum Verändern der Fuse- und Lock-Bits sollte der entsprechende Abschnitt im Datenblatt des Controllers studiert werden. Über die Dialogfelder Low-, High- und Extended-Fuse- sowie Lock-Bits können die einzelnen Optionen bequem ausgewählt werden. Die Änderungen werden im Anzeigebereich für die Fuse- und Lock-Bits visualisiert. Erst mit dem Betätigen der Schaltfläche „Jetzt Schreiben“ werden die neuen Einstellungen an den Controller übertragen.

**Beachte:** Falsche Einstellungen der Fuse- oder Lock-Bits können dazu führen, dass der Controller in der aktuellen Hardware nicht mehr angesprochen werden kann. Häufige Fehleinstellungen durch unerfahrene Entwickler sind:

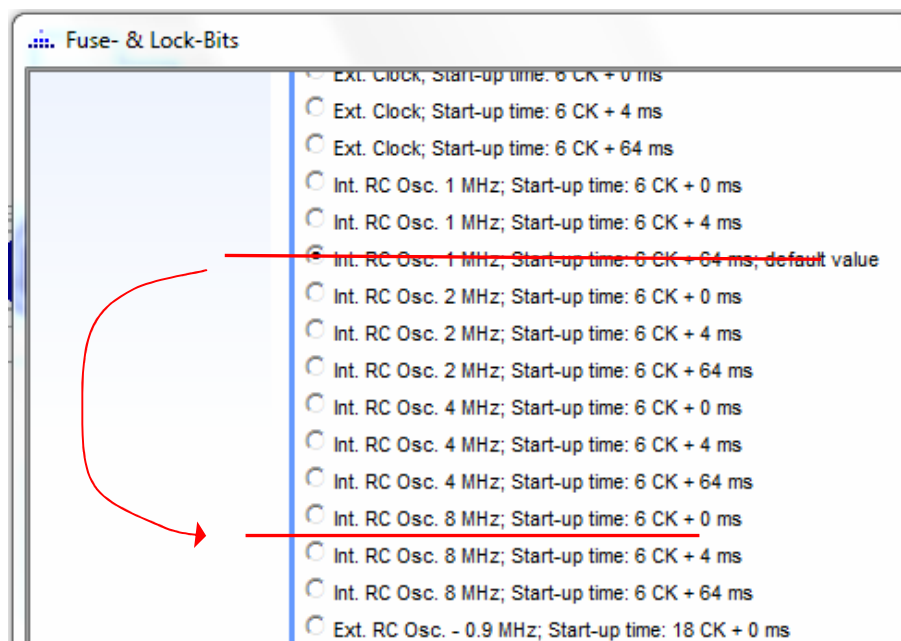
- **Reset disable** -> führt dazu, dass kein ISP mehr möglich ist
- **ISP enable ausgeschaltet** -> führt dazu, dass kein ISP mehr möglich ist
- **Taktquelle umgeschaltet** -> führt u.U. dazu, dass der Controller nicht arbeitet

Im Folgenden wird die Vorgehensweise beschrieben, wie die Taktquelle eines ATmega8 vom internen 1 MHz Oszillator auf intern 8 MHz umgeschaltet wird.

1. Das Board und den Programmer anschließen.
2. Benutzeroberfläche für das Verändern der Fuse- & Lock-Bits starten.

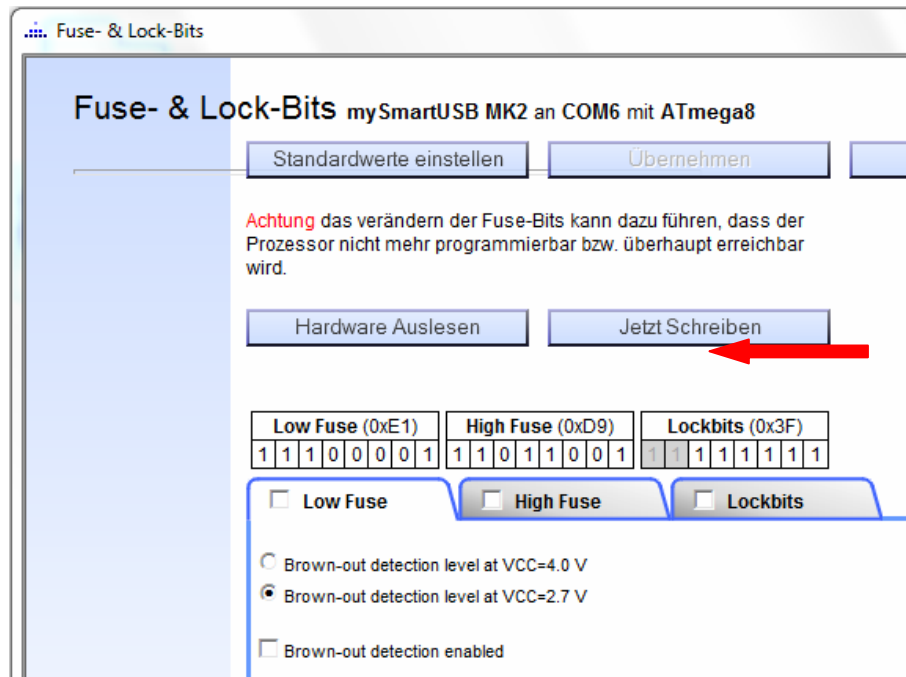


3. In der Liste der Optionen nach unten scrollen und **Int. RC. Osc. 8MHz** auswählen. Vergleichen Sie dazu die Beschreibung im Datenblatt des ATmega8.





- Die Optionsseite zurückscrollen und die veränderten Fuse-Bits überprüfen.



- Die Schaltfläche „Jetzt Schreiben“ wählen und die Sicherheitsabfrage bestätigen. Danach sollten die Einstellungen überprüft werden durch das Betätigen der Schaltfläche „Hardware Auslesen“.

Über die Schaltfläche „Verlassen“ kann die Sitzung zum Verändern der Fuse- und Lock-Bits beendet werden. Die Verbindung zum Controller und Programmer wird dann geschlossen.

## 6.6 Der Debugger

### 6.6.1 Debuggen von SVL- und ARM-Programmen

SiSy verfügt über ein Debug-Interface um C oder C++ Programme zu debuggen. Unter Debugging versteht man bestimmte Arbeitstechniken zur Fehlersuche in laufenden Programmen. Dazu gehören vor allem der Schrittbetrieb, Unterbrechungspunkte, Kontrollausgaben und Datenüberwachung.

Der integrierte Debugger ermöglicht es Ihnen, sowohl auf Quellcodeebene als auch auf Modellebene Fehler zu suchen. Das Starten des Debuggers erfolgt in der Regel über das Aktionsmenü des aktiven Diagramms bzw. über die entsprechende Schaltfläche.

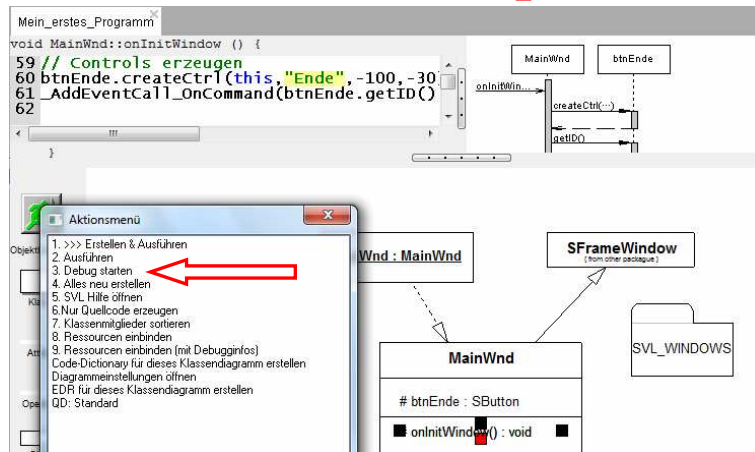


Abbildung 41: Möglichkeiten für Start Debugger

Der Debugger startet und wird mit dem dazugehörigen Quelltext angezeigt. Dabei wird in jedem Fall ein Unterbrechungspunkt beim Programmstart (main/WinMain) gesetzt und das Programm dort angehalten. Jetzt kann der Anwender weitere Unterbrechungspunkte hinzufügen. Die Position, an der ein Programm angehalten wurde, ist im Quelltext als Zeile markiert. Wenn die entsprechende Position im Modell ermittelt werden konnte, wird das Modellfenster nachgeführt und das betreffende Element selektiert.

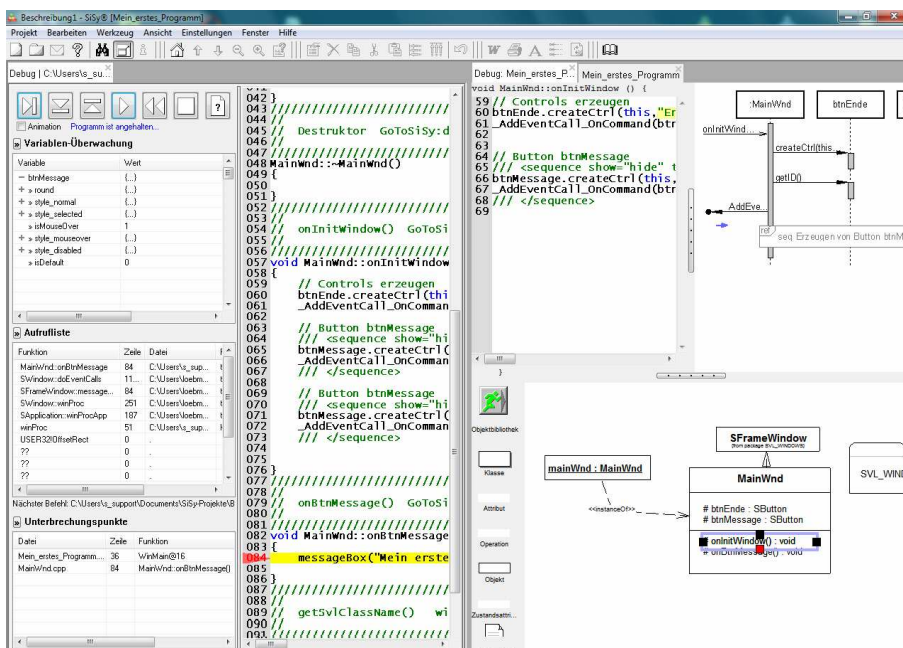







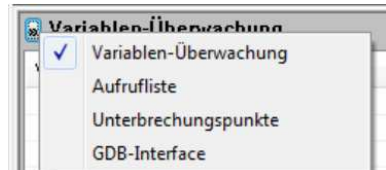


Abbildung 42: SiSy mit geöffneten Debugger im Klassendiagramm

**Die Schaltflächen des Debuggers:**

	<b>Einen Schritt:</b> Das Programm Schritt für Schritt ablaufen lassen. Dies hilft vor allem bei der Fehlersuche, da hier Zeile für Zeile des Quellcodes abgearbeitet wird. Funktionen werden dabei jedoch als ganze Einheit betrachtet.
	<b>Einen Schritt hinein:</b> Verzweigt das Programm zusätzlich in die Einzelanweisungen von aufgerufenen Funktionen.
	<b>Funktion beenden</b>
	<b>Programm laufen lassen</b> Das Programm wird abgearbeitet bis zum nächsten Haltepunkt
	<b>Programm neu starten</b>
	<b>Debugger beenden:</b> Zu jeder Zeit möglich, auch wenn das Programm nicht komplett abgearbeitet wurde.
	<b>Auswahl-Menü:</b> ein Drop-Down Menü wird geöffnet, aus dem wie in der Abbildung dargestellt, verschiedene Debug-Ansichten gewählt werden können.



Die **Variablen-Überwachung** dient dazu, sich während des schrittweisen Abarbeitens des Programms im Debugger die aktuellen Werte der verwendeten Variablen anzeigen zu lassen. Hierzu werden die zu überwachenden Variablen aus dem Quellcodeeditor des Diagrammfensters hinzugefügt. Verwenden Sie dafür aus dem Kontextmenü des Editors „zur Überwachung hinzufügen“

Die **Aufrufliste** zeigt in der Spalte *Funktion* an, welche Unit bzw. Funktion gerade abgearbeitet wird. Dies hilft vor allem beim Abarbeiten von Programmen mit mehreren Units oder Funktionen. In der Spalte *Zeile* wird die aktuelle Quelltextzeile angezeigt, in der sich der Debugger gerade befindet und in der Spalte *Datei* wird die Datei angegeben, die gerade abgearbeitet wird.

**Unterbrechungspunkte** können im Quellcodeeditor des Diagrammfensters hinzugefügt werden. Nutzen Sie dazu den Menüpunkt „Breakpoint umschalten“ im Kontextmenü des Editors.

**Hinweis:**

*Wenn der Anwender durch Nutzung des Debuggers einen Fehler findet und korrigiert, reicht es nicht, das Programm neu zu starten, das Programm muss erst neu erstellt werden, damit die Änderungen übernommen werden.*

## 6.6.2 Der SVL Debug-Monitor

Für die Fehlersuche im laufenden Betrieb einer ARM-Anwendung und einer Windows-anwendung die auf der SVL-Klassenbibliothek basiert, steht Ihnen der Debug-Monitor zur Verfügung. Damit besteht die Möglichkeit im Quelltext zusätzliche Befehle für die Ausgabe von geplanten Aktionen einzugeben, die in einem separaten Fenster ausgegeben werden.

Der Debug-Monitor wird gestartet über die Menüfolge *Werkzeug/Debugmonitor*.

Die SVL bietet intern bereits eine vorbereitete IP-Schnittstelle zum Debug-Monitor, die noch aktiviert werden muß. Dabei kann es zu Warnhinweisen Ihrer Firewall kommen. Es ist für die korrekte Funktion des Debug-Monitors notwendig, die Verbindung zwischen Anwendung und Debug-Monitor nicht zu blockieren.

In der folgenden Darstellung sehen Sie eine typische Anwendung des Debug-Monitors. Dabei fungiert der Debug-Monitor als lokaler Server und die Anwendung als Client. Der Debug-Monitor muss immer zuerst gestartet werden und danach erst die Anwendung. In der SVL-Hilfe (rechte Maustaste im Quelltext) können Sie mit dem Stichwort „SDebug“ detaillierte Informationen zur Nutzung der Debugschnittstelle aufrufen.

The image shows a UML class diagram and a source code editor. The class diagram includes classes like `SFrameWindow`, `SVL_WINDOWS`, `MainWnd`, and `mainWnd : MainWnd`. The source code editor shows the following code snippet:

```

59 debug.stop=false;
60 debug.print("Starte Fensteraufbau");
61 // Controls erzeugen
62 btnEnde.createCtrl(this, "Ende", -100, -30);
63 _AddEventCall_OnCommand(btnEnde.getID(), destroy);
64 debug.print("Ende Fensteraufbau");
65 debug.stop=true;
66

```

A callout box highlights the following code snippet:

```

debug.stop=false;
debug.print("Starte Fensteraufbau");
// Controls erzeugen
btnEnde.createCtrl(this, "Ende", -100, -30);
_AddEventCall_OnCommand(btnEnde.getID(), destroy);
debug.print("Ende Fensteraufbau");
debug.stop=true;

```

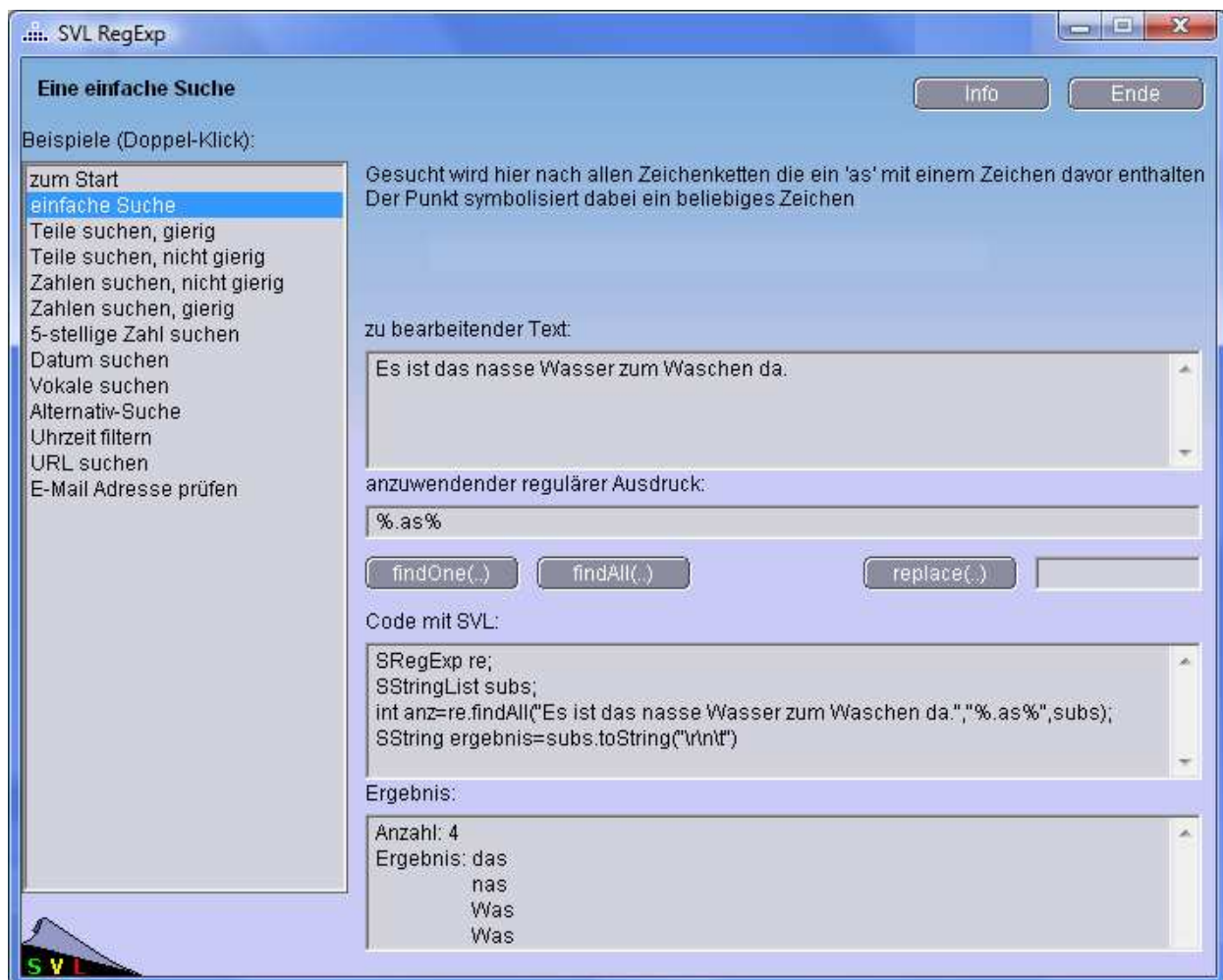
The image shows the SiSy - Windows Application mit SVL interface. The main window displays the text "Starte Fensteraufbau" and "Ende Fensteraufbau". A separate window titled "SVL-Debug-Monitor" is open, showing a green bar with the text "Client akzeptiert". Another window titled "SVL-Debug-Monitor" is also visible, showing a yellow bar with the text "Server ne gestartet". The interface includes buttons for "löschen" and "Ende".

**Hinweis:**

Der SVL Debug-Monitor steht nicht mit jeder SiSy Ausgabe zur Verfügung. Sie benötigen dazu die entsprechenden Add-Ons, beziehungsweise die entsprechende Ausgabe von SiSy.

**6.6.3 Das SVL-Werkzeug RegExp**

Reguläre Ausdrücke sind Regeln für die Suche in Zeichenketten. Man definiert also mit dieser Syntax Suchmuster. Benutzen Sie dieses Werkzeug um die Syntax eines gewünschten regulären Ausdrucks zu testen. Sie erhalten zusätzlich den C++ Code für die Anwendung des getesteten regulären Ausdrucks mit der SVL. Weitere Informationen finden Sie in der SVL-Hilfe (rechte Maustaste im Quelltext) unter den Stichwort „SRegExp“.

**Hinweis:**

SVL Anwendungen können nicht mit jeder SiSy Ausgabe erstellt werden. Sie benötigen dazu die entsprechenden Add-Ons beziehungsweise die entsprechende Ausgabe von SiSy.

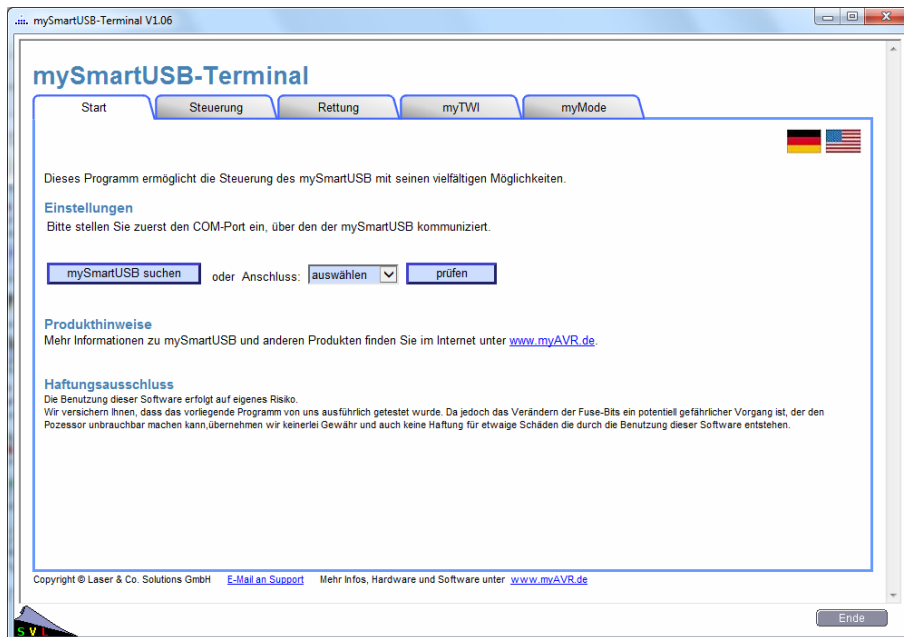
## 6.7 Weitere Werkzeuge

Entsprechend der installierten Add-Ons Ihrer SiSy Ausgabe können weitere Werkzeuge verfügbar sein.

### 6.7.1 mySmartUSB Terminal

Menüfolge *Werkzeug/mySmartUSB Terminal*

Dieses Programm ermöglicht die Steuerung des mySmartUSB MK2 mit seinen vielfältigen Möglichkeiten. Es ist sehr gut geeignet, die einzelnen Schritte eines TWI- oder SPI-Protokolls auszuführen und die Abfolge der Aktionen und Rückmeldungen zu beobachten. Mit diesem Werkzeug lassen sich u.a. viele Controller mit verstellten Fuse-Bits reparieren.



### 6.7.2 mySmartUSB light - SupportBox

Menüfolge *Werkzeug/mySmartUSB light - SupportBox*

Diese Supportbox ist ein kleines Tool auf einer grafischen Oberfläche zum Schalten und Updaten des mySmartUSB light.



## 7 Informationen zu SiSy-Ausgaben

Die Laufzeitkomponenten von SiSy, das Metamodell, Add-Ons sowie spezifische Skripte, Dokumente und Daten sind lizenzpflichtige Produkte. Ausgenommen sind zusätzlich bzw. nachträglich installierte Komponenten Dritter, die von SiSy als Modellierungswerkzeug lediglich zum Beispiel über Skripte angesprochen werden oder in die sich SiSy über offen gelegte Schnittstellen eingebettet hat. Solche Produkte unterliegen den Lizenz und Nutzungsbedingungen des jeweiligen Herstellers bzw. Herausgebers.

### **Ausgaben**

- SiSy Professional  
umfasst alle verfügbaren Add-Ons
- SiSy Business  
umfasst Add-Ons für Prozessmodellierung, Qualitäts- und Projektmanagement
- SiSy Developer  
umfasst umfangreiche Add-Ons zur Systementwicklung
- SiSy Mikrocontroller++  
umfasst die Add-Ons AVR, ARM, ausgewählte Teile der UML, SVL und SysML
- SiSy AVR  
beinhaltet das Add-On AVR, ausgewählte Teile der UML
- SiSy STM32  
umfasst die Add-Ons ARM, ausgewählte Teile der UML und SysML
- SiSy XMC  
umfasst die Add-Ons ARM, ausgewählte Teile der UML und SysML

### **Lizenzmodelle**

- Evaluation License  
erlaubt die Nutzung auf einem Arbeitsplatzrechner für Evaluierungszwecke  
Lizenzlaufzeit 1 Jahr
- Education License  
erlaubt die Nutzung auf einem Arbeitsplatzrechner für Ausbildungszwecke  
Voraussetzung: Bildungseinrichtung, Student, Schüler, keine kommerzielle Nutzung
- Private License  
erlaubt die Nutzung auf einem Arbeitsplatz für private Zwecke
- Single License  
erlaubt die Nutzung auf einem Arbeitsplatz für kommerzielle Zwecke oder in öffentlichen Verwaltungen
- Enterprise License  
erlaubt die Nutzung auf bis zu 20 Arbeitsplätzen für kommerzielle Zwecke oder in öffentlichen Verwaltungen

## Anhang: Tastaturbelegung, allgemein

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

F1	Hilfe wird geöffnet
F2	Zoomen
F3	Quelltext öffnen
F4	Textfeld/Infofeld vergrößern
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 rückgesetzt wird
ESC	Im Diagramm: Nach oben Im Dialog: Bricht ihn ab ohne zu speichern
Tab	In der Reihenfolge, in der die Objekte erstellt wurden, werden sie markiert
Umschalttaste + Enter	Objektbeschreibung wird geöffnet
Leertaste	Objektbeschreibung wird geöffnet
Alt + Enter	Definieren-Dialog wird geöffnet
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
Strg + `-`	Selektiertes Objekt wird verkleinert
Strg + `*`	Ursprüngliche Objektgröße wird wiederhergestellt



**Strg + Cursortasten:**

- 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

Cursortasten            Selektiertes Objekt wird verschoben (in Verbindung mit der Umschalttaste sind größere Schritte möglich)

Numerisch `+`           Diagramm vergrößern

Numerisch `-`           Diagramm verkleinern

Numerisch `\*`           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 <i>Shift</i> und <i>gedrückter linker Maustaste</i> auf Fensterhintergrund und Ziehen eines Rechtecks über zu markierende Objekte.
Verschieben	<i>Drag&amp;Drop im Diagramm</i> Objekt mit linker Maustaste anfassen und verschieben. Objekte werden am Raster verschoben.  <i>Umschalttaste + Drag&amp;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&amp;Drop</i> Maustaste gedrückt halten und Mauszeiger vom Objekt auf den Fensterhintergrund bzw. die Registerkarte führen. Kopie eines Originals → neues Original Kopie einer Referenz → neue Referenz vom selben Original
Referenz erzeugen	<i>Drag&amp;Drop aus Navigator</i> Ziehen des gewünschten Objektes aus dem Navigator in das Diagramm. Es wird eine Referenz des gewählten Objektes erzeugt.  <i>Drag&amp;Drop aus Objektbibliothek</i> Rot beschriftete Objekte können nur als Referenz erzeugt werden. Eine Liste zur Auswahl des gewünschten Typs erscheint.  <i>Strg + Drag&amp;Drop aus Objektbibliothek</i> Eine Liste zur Auswahl der gewünschten Referenz des Originalobjektes erscheint.

	<p><i>Drag&amp;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&amp;Drop aus Objektbibliothek</i> Ein Objekt aus der Objektbibliothek wird im Diagramm angelegt und steht zur Verfeinerung bereit.</p>
Objekt anhängen	<p><i>Drag&amp;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&amp;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&amp;Drop Mittelpunkt</i> Beliebige Gestaltung der Verbindung durch Ziehen mit der Maus.</p>
Verbindung ändern	<p><i>Drag&amp;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>

## Index

Add-On .....	27	Objekt .....	11
Aktionsmenü .....	15	Objektbibliothek .....	15
Aktivitätsdiagramm .....	65	PAP.....	<i>Siehe</i> Programmablaufplan
Anwendungsfalldiagramm.....	64	Profil.....	9
Archiv.....	10	Programmablaufplan.....	27, 48
Assistent .....	14, 32	ProgTool .....	<i>Siehe</i> myAVR ProgTool
Ausgabefenster.....	14	Projekt anlegen.....	9
Code-Vervollständigung.....	35, 58	Projekt archivieren .....	10
Code-Wizard.....	79	Projekt definieren .....	9
ControlCenter .....	71	Projekt öffnen.....	10
CRC-Karte .....	65	Projekt-Explorer .....	14
Debugger .....	98	Quellcodefenster.....	<i>Siehe</i> Editorfenster
Debug-Monitor .....	100	Rahmen .....	12
Definieren-Dialog .....	11, 18, 104	Referenz .....	12, 18, 52, 106
Diagramm .....	11	RegExp .....	101
Diagrammfenster .....	14	Report.....	20
Dokumentengenerierung .....	26	Sequenzdiagramm.....	14
Editorfenster .....	14	SiSy-Benutzeroberfläche .....	13
Entity.Relationship .....	70	Struktogramm .....	44
ERD .....	<i>Siehe</i> Entity.Relationship	SVL.....	38
Explorer .....	<i>Siehe</i> Projekt-Explorer	Template.....	40
Fuse-Bits.....	89	Tutorial.....	31, 40
Kante .....	12	Verteiler .....	11, 19, 48
Kontextmenü.....	17	VGM.....	<i>Siehe</i> Vorgehensmodell
LibStore .....	30	Vorgehensmodell.....	9
Lizenz .....	6	Werkzeug.....	11
Modell .....	11, 16	Werkzeuggeste .....	15, 17
Modellelement .....	11, 18	Willkommenbildschirm .....	7
Navigator .....	13, 52, 106	Zwischenablage.....	25, 76