



### Zielstellung

Entwickeln Sie eine Mikrocontrollerlösung, bei der auf dem myAVR Board MK2 ein Taster eine LED schaltet. Realisieren Sie diese Mikrocontroller-Anwendung mit der Technik des Programmablaufplanes und in der Sprache Assembler.

### Voraussetzungen

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. In dieser Beispielanwendung wird die Aufgabenstellung in einem Programmablaufplan modelliert und aus diesem PAP der Quellcode in Assembler generiert. Das generierte Programm kann sofort auf den Controller gebrannt werden.

### Software

- SiSy Ausgabe AVR, Microcontroller++ oder Professional
- Windows XP ... Windows 10

### Hardware:

- myAVR Board MK2
- 1 Programmierkabel (USB-Kabel)

### Schaltung:

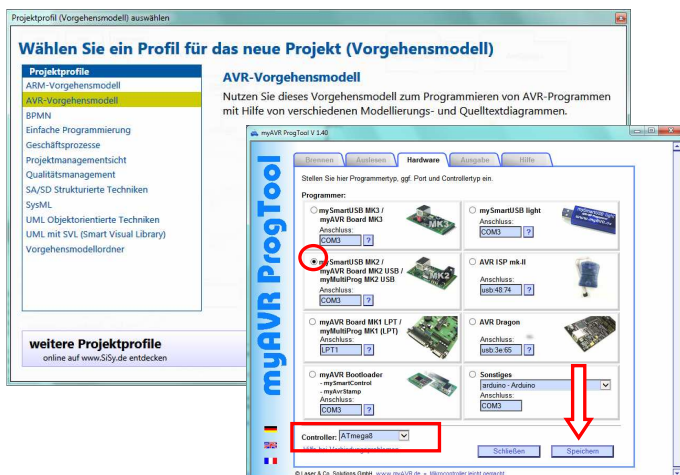
- Port B.0 = Taster 1
- Port B.1 = LED

Im SiSy LibStore finden Sie Beispielprogramme und Programmvorlagen zum Download, die kontinuierlich aktualisiert werden. Eine ausführliche Beschreibung zum SiSy LibStore und der Hilfsfunktionen, z.B. Syntax zu Befehlen oder Druckmöglichkeiten, finden Sie im Benutzerhandbuch von SiSy.

## 1. Einfache Programmentwicklung aus einem PAP

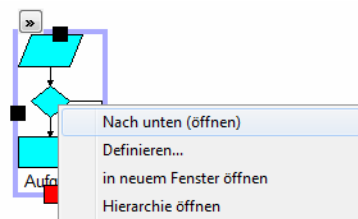
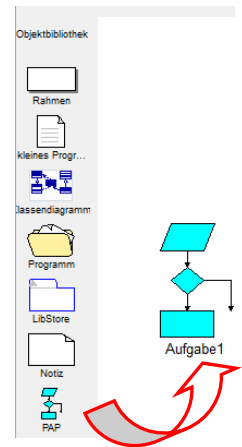
### Vorbereitung

Starten Sie SiSy und legen Sie ein neues Projekt an. Wählen Sie das AVR-Vorgehensmodell und nehmen Sie die Grundeinstellungen für die verwendete AVR Hardware vor.



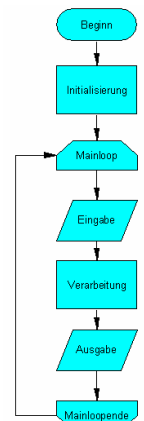
Es öffnet SiSy Libstore, bietet aber keine Vorlagen an. Kehren Sie zurück zu SiSy, Sie erhalten ein leeres Diagramm.

Ziehen Sie aus der Objektbibliothek ein Objekt vom Typ „PAP“ in das leere Diagramm. Benennen Sie das Objekt mit „Aufgabe1“. Der nächste Schritt ist das Aufstellen des Programmablaufplanes. Dazu muss das Diagramm unter dem Symbol geöffnet werden. Wählen Sie aus dem Kontextmenü (rechte Maustaste) „Nach unten (öffnen)“, damit Sie in dieses Diagramm gelangen.



### Grundstruktur laden

Wenn ein Diagramm leer ist, öffnet SiSy LibStore und bietet, soweit vorhanden, Vorlagen für die weitere Arbeit an. Führen Sie den Download für die Vorlage „Grundgerüst PAP AVR Assembler“ aus. Sie erhalten ein Grundgerüst wie in der Abbildung gezeigt.

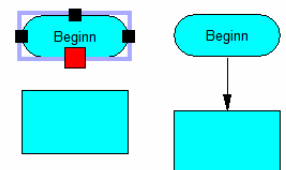


### Logik entwerfen

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.

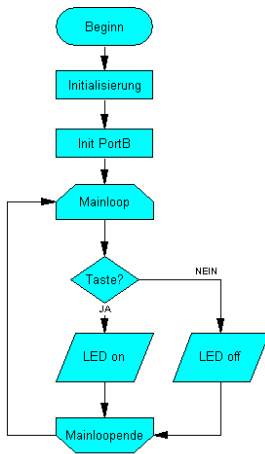
Ein Objekt wird im Diagramm ergänzt, indem der entsprechende Objekttyp in der Objektbibliothek ausgewählt und per Drag&Drop an die entsprechende Position im Diagramm gezogen wird.

Verbindungen zwischen den Objekten können über den rot markierten Verteiler von selektierten Objekten hergestellt werden. Dazu ist das Ausgangsobjekt zu selektieren und mit dem Mauscursor von dem roten Verteiler bei gedrückter linker Maustaste eine Verbindung zum Zielobjekt zu ziehen.



Für das Einfügen von Objekten zwischen bestehende Verbindungen ist zuvor die bestehende Verbindung zu löschen. Für die Benennung der Objekte öffnen Sie den Definieren-Dialog aus dem Kontextmenü (rechte Maustaste auf dem Objekt).

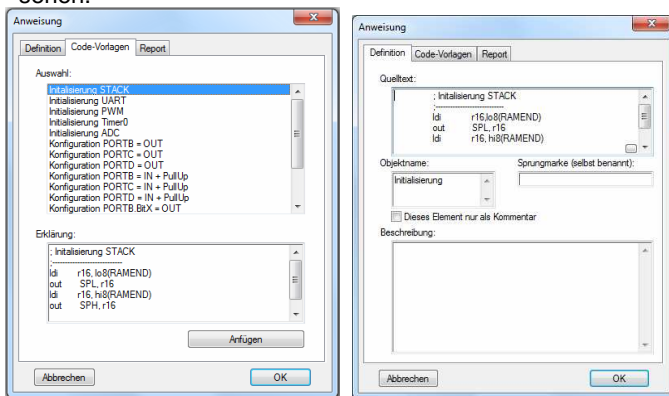
Erweitern Sie die Vorlage entsprechend der folgenden Abbildung.



**Befehle eingeben**

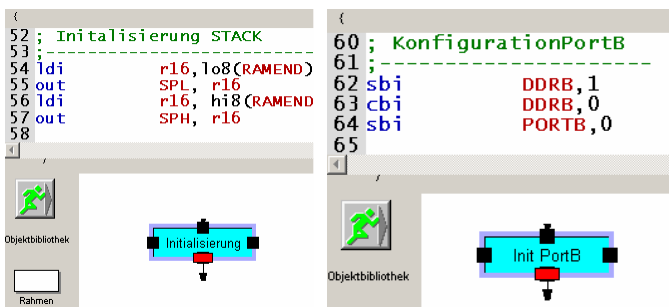
Nachfolgend soll aus dem Programmablaufplan Assemblerquellcode generiert werden. Dazu ist es nötig, die einzelnen Elemente des PAP mit den entsprechenden Assembleranweisungen zu versehen. Dafür gibt es mehrere Möglichkeiten:

- 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.



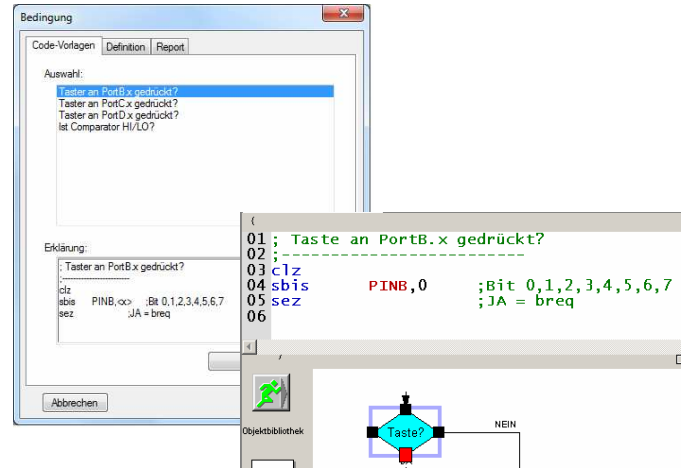
- Die zweite Möglichkeit besteht beim Selektieren von Elementen und Eintragen in den Quellcodeeditor oberhalb des Diagrammfensters.

Geben Sie die gezeigten Quellcodes in die Objekte ein!



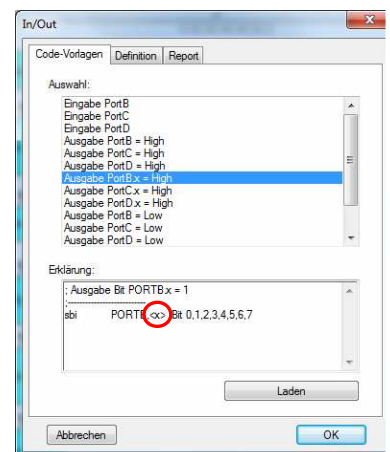
Bedingungen haben spezielle Vorlagen, die eine Codegenerierung bei übersichtlichem Programmablaufplan vereinfachen.

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.)

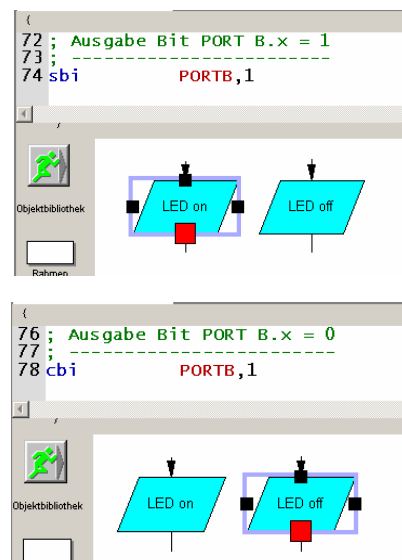


Das Element „IN/OUT“

verfügt auch über spezifische Vorlagen. Diese sind gegebenenfalls mit Ergänzungen versehen. Dazu sind spitze Klammern als Platzhalter in den Vorlagen eingefügt.

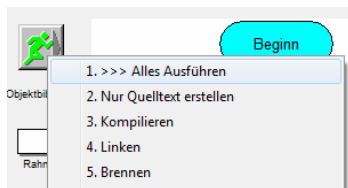


Ergänzen Sie den Quellcode der gezeigten Elemente!



### Übersetzen, Brennen und Test

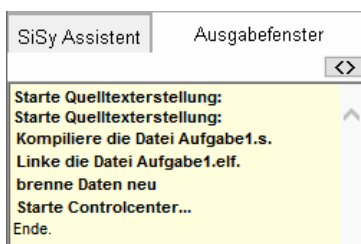
Sind alle betreffenden Elemente mit Quellcode hinterlegt, kann aus dem Programmablaufplan der komplette Quellcode generiert, kompiliert, gelinkt und auf den Mikrocontroller übertragen werden. Die gewünschte Funktion kann aus dem Aktionsmenü gewählt werden.



Hinweis:

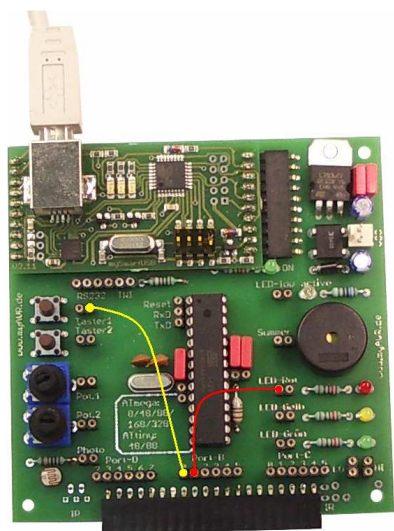
Beachten Sie, dass für das Brennen des Controllers das Programmierkabel angeschlossen sein muss und bei Bedarf eine geeignete Spannungsquelle anzuschließen ist.

Im Ausgabefenster werden die jeweiligen Aktionen angezeigt. Compilerfehler werden mit der entsprechenden Zeilennummer angezeigt. Mit Klick auf die Fehlermeldung gelangt man zum entsprechenden Objekt.



Nachdem das Programm erfolgreich übersetzt und auf den Controller übertragen wurde, öffnet das SiSy ControlCenter. Mit diesem Beispielprogramm erfolgt keine Kommunikation mit den PC; das SiSy ControlCenter kann geschlossen werden.

Für den Test der Anwendung müssen auf dem Board noch die entsprechenden Verbindungen hergestellt werden, wie in der Abbildung gezeigt, nutzen Sie die Patchkabel.



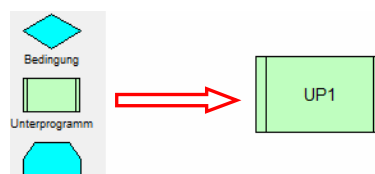
Verbindungen auf dem myAVR Board MK2

## 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.

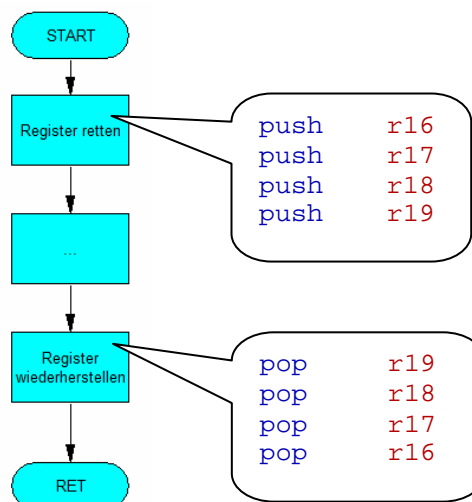
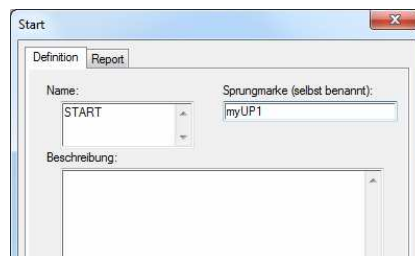
### Anlegen eines Unterprogramms

Ziehen Sie den Objekttyp „Unterprogramm“ aus der Objektbibliothek in das gewünschte Diagramm. Mit Doppelklick auf dem Objekt oder über das Kontextmenü -> Definieren können Sie dem Unterprogramm einen Namen geben.




Damit ist ein Objekt angelegt, welches im aktuellen Diagramm als Aufruf (call) des Unterprogramms zu verstehen ist. Die Funktionalität des Unterprogramms wird in einem gesonderten Programmablaufplan für das Unterprogramm entworfen. Dazu ist das Diagramm „unter“ bzw. „hinter“ dem Objekt „Unterprogramm“ zu öffnen. Für das Öffnen des zugehörigen Diagramms wählen Sie auf dem Objekt aus dem Kontextmenü -> Nach unten (öffnen).

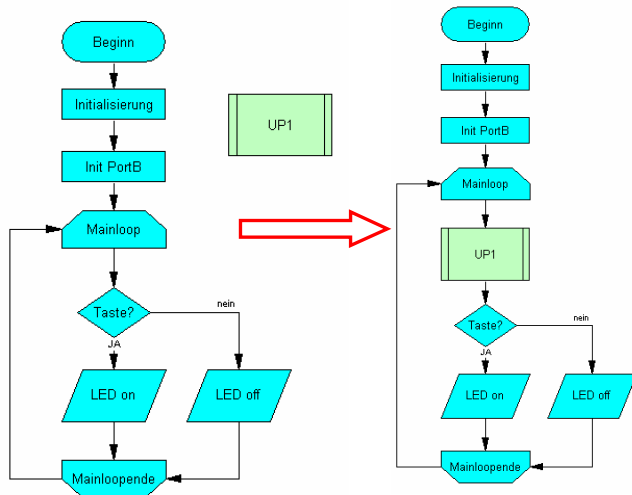
SiSy LibStore öffnet und Sie erhalten eine Vorlage für die Grundstruktur von Unterprogrammen. Bitte laden Sie die Vorlage „Grundgerüst Unterprogramm UP AVR PAP“. Auf dem Objekt „START“ können Sie eine benutzerdefinierte Sprungmarke festlegen (Rechtsklick -> Definieren), die durch den Codegenerator erstellt und verwendet werden soll. Die Vorlage muss entsprechend der vorgesehen Logik abgeändert werden.



### Ein Unterprogramm aufrufen

Das Unterprogrammssymbol muss zum Aufruf an der entsprechenden Position im Programmablaufplan eingefügt werden. Der Codegenerator erzeugt dann entsprechend einen Unterprogrammaufruf und den Code für das Unterprogramm selbst. Dazu ist in das Diagramm zurückzukehren, in dem das Objekt „Unterprogramm“ angelegt wurde (aus dem Kontextmenü → nach oben... oder  aus der Werkzeugleiste).

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



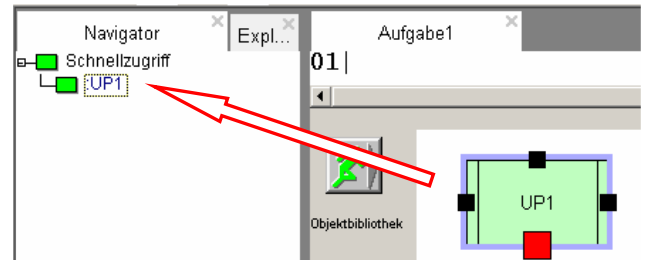
Einfügen des Unterprogramms in den Programmablaufplan

### Unterprogramme mehrmals benutzen

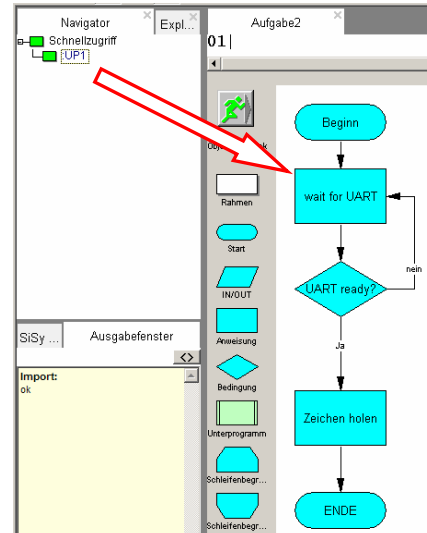
Ein wesentliches Merkmal von Unterprogrammen ist, dass diese von verschiedenen Stellen im Programm aufgerufen (*call*) werden können und auch dorthin zurückkehren (*return*). Um diese Möglichkeit zu nutzen, bietet SiSy das Anlegen von Referenzen.

Um ein Unterprogramm zu referenzieren (wiederholend zeigen und einbinden) gehen Sie wie folgt vor:

1. zeigen Sie im Navigator das 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



Objekt im Navigator/Schnellzugriff hinzufügen



Objekte aus dem Navigator referenzieren