



### Zielstellung

Es soll eine Mikrocontrollerlösung entwickelt werden, bei der auf Tastendruck die rote LED eingeschaltet wird. Realisieren Sie diese Aufgabenstellung in der Sprache C mit einem Struktogramm.

### Voraussetzungen

Für die Abarbeitung dieses Beispiels sind Kenntnisse in einer Programmiersprache und der UML von Vorteil. Sie benötigen folgende Software und Hardware:

#### Software

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

#### Hardware

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

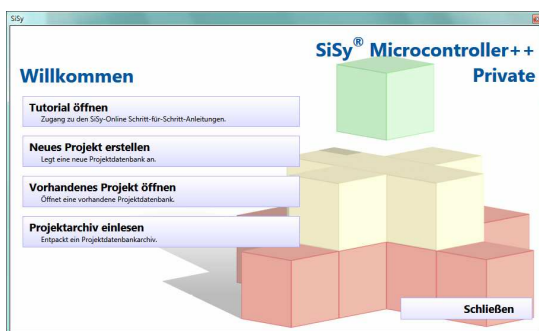
#### Schaltung:

Port B.0 = Taster  
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 Hilfefunktionen, z.B. Syntax zu Befehlen oder Druckmöglichkeiten, finden Sie im Benutzerhandbuch von SiSy.

### 1. Ein neues Projekt anlegen

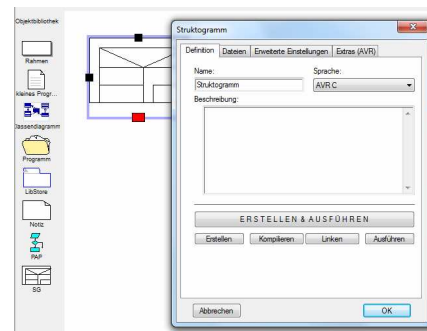
Starten Sie SiSy und wählen Sie „Neues Projekt erstellen“, vergeben Sie den Projektnamen und bestätigen Sie mit „Projekt anlegen“. Wählen Sie das Projektprofil (Vorgehensmodell) „AVR-Vorgehensmodell“ aus.



Legen Sie im weiteren Verlauf die AVR-Grundeinstellungen (Hardware-Konfiguration) für das Projekt fest. SiSy LibStore öffnet, ohne für Struktogramme Vorlagen anzubieten. Kehren Sie zu SiSy zurück und arbeiten Sie ohne Diagrammvorlage.

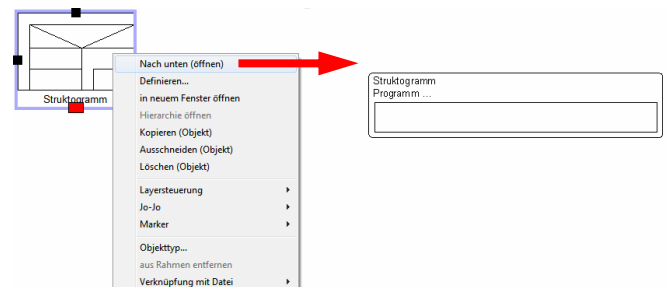
### 2. C-Programm anlegen

Für das Erstellen des Programms für den AVR Mikrocontroller ziehen Sie per Drag & Drop aus der Objektbibliothek ein Objekt vom Typ „SG“ (Struktogramm) in das Diagrammfenster. Legen Sie per Definition (rechte Maustaste/Definieren) den Namen für das Struktogramm fest und die Sprache „AVR C“.



### 3. Struktogramm entwickeln

Für die Entwicklung des Struktogramms muss das Struktogrammfenster geöffnet werden; Kontextmenü → Nach unten (öffnen).

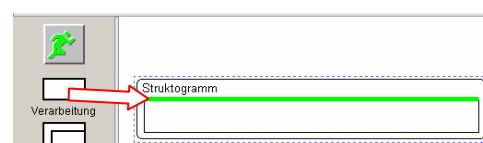


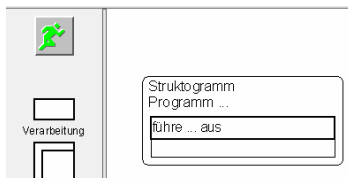
Beim Einfügen einzelner Struktogramm-Elemente in das Struktogramm sind die möglichen Positionen grün unterlegt. Bewegen Sie den Mauszeiger an die gewünschte Stelle, die grüne Linie wird zur Kontrolle breiter.

Führen Sie folgende Arbeitsschritte aus, um den Algorithmus für die oben genannte Aufgabe zu entwerfen:

Für das Editieren selektieren Sie das betreffende Element per Doppelklick. Mit ESC verlassen Sie den Editiermodus

1. Konfigurieren Sie Port B.0 als Eingang und Port B.1 als Ausgang. Ziehen Sie dafür eine „Verarbeitungsroutine“ aus der Objektbibliothek in das Struktogramm.



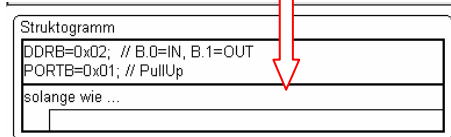
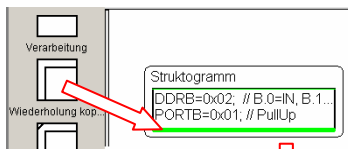


Verarbeitungsroutine:

```

DDRB=0x02; // B.0=IN, B.1=OUT
PORTB=0x01; // PullUp
    
```

2. Fügen Sie eine „WHILE-Schleife“ ein.

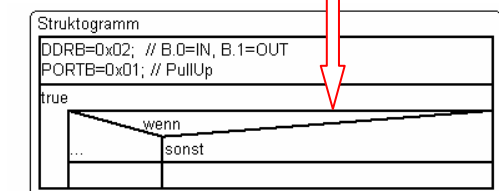
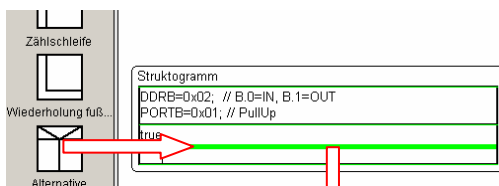


While-Schleife:

```

true
    
```

3. Ergänzen Sie die While-Schleife mit einer „Alternative“ per Drag&Drop. Achten Sie darauf, dass die Alternative innerhalb der While-Schleife liegt. Geben Sie den Titel und die Bedingung für die Alternative ein.

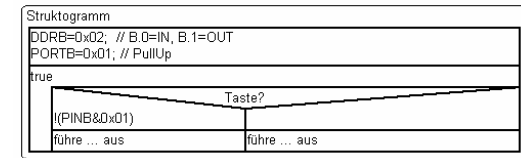
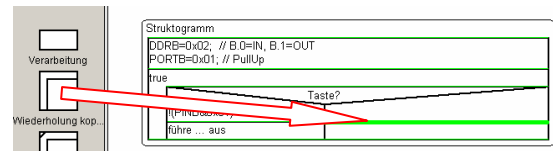
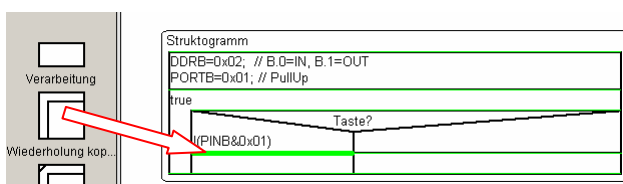


Alternative: wenn  
Taste?

```

Alternative: ...   Alternative: sonst
!(PINB&0x01)     // bleibt leer
    
```

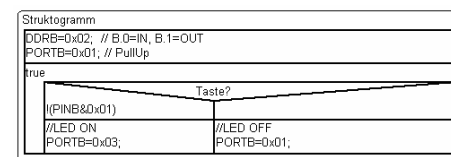
4. Als nächstes sind die Aktionen (DO-Elemente) in der Alternative zu ergänzen. Ziehen Sie dafür erneut jeweils eine „Verarbeitungsroutine“ in die Alternative des Struktogramms.



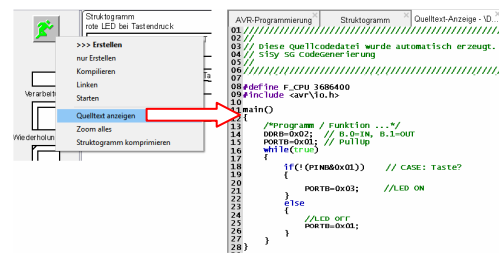
```

//LED ON   //LED OFF
PORTB=0x03; PORTB=0x01;
    
```

5. Tragen Sie den C-Quellcode zum Ein- und Ausschalten der LED in die DO-Elemente ein.



6. Generieren Sie den Quellcode für dieses Struktogramm und lassen Sie sich den Quellcode anzeigen.



**4. Programmtest**

Verbinden Sie das Board mit dem PC und aktivieren Sie aus dem Aktionsmenü den Befehl „>>> Erstellen“, damit brennen Sie das Programm auf den Controller. Es öffnet das myAVR ProgTool und zeigt das Protokoll an. Für den Test des Programms stecken Sie auf dem Board die Verbindungen.

