Zielstellung

Auf dem myAVR Board MK2 sollen beim Programmstart alle optischen Ausgabegeräte (rote, gelbe und grüne LEDs) leuchten. Als Mikrocontroller wird der ATmega8 verwendet. Die Ausgabegeräte (LEDs) werden vom Prozessorport D gesteuert.
In diesem Schnelleinstieg wird das Programm in der Programmiersprache C++ mit PEC realisiert. PEC (Portable Embedded Classes) ist eine portable Klassenbibliothek für eingebettete Systeme für verschiedene Mikrocontroller.

Voraussetzungen

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

Software

- SiSy-Ausgabe AVR, Microcontroller++, Professional oder Developer ab der Version 3.6
- Windows XP … Windows 10
- installierter USB-Treiber

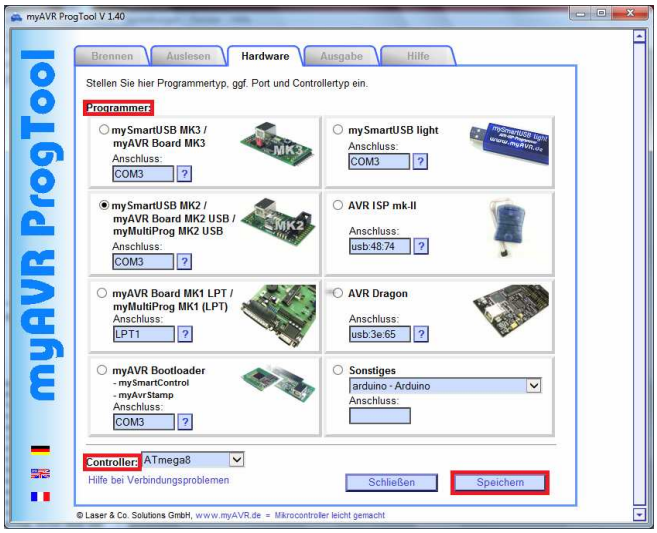
Hardware

- ein bestücktes myAVR Board MK2 (alternativ myAVR Board light)
- Programmierkabel (USB)
- Patchkabel

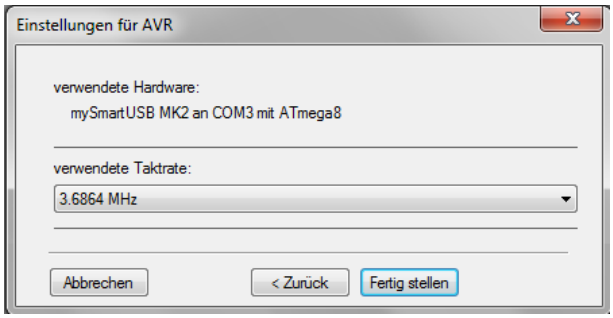
Im SiSy LibStore finden Sie Beispielprogramme und Programmvorlagen zum Download, die kontinuierlich aktualisiert werden. Eine ausführliche Beschreibung zum SiSy LibStore und der Hilfefunktionen, z.B. Syntax zu Befehlen oder Druckmöglichkeiten, finden Sie im Benutzerhandbuch von SiSy.

1. Ein neues Projekt anlegen

Starten Sie SiSy, wählen Sie „Neues Projekt erstellen“ und vergeben Sie einen Programmnamen, z.B. „Alle Lichter an“. Aktivieren Sie das „AVR-Vorgehensmodell“ und bestätigen Sie „dieses Profil auswählen“. Es öffnet das Fenster vom “myAVR ProgTool“. Wählen Sie in diesem Ihren Programmer sowie den Controller aus; „Speichern“ Sie.



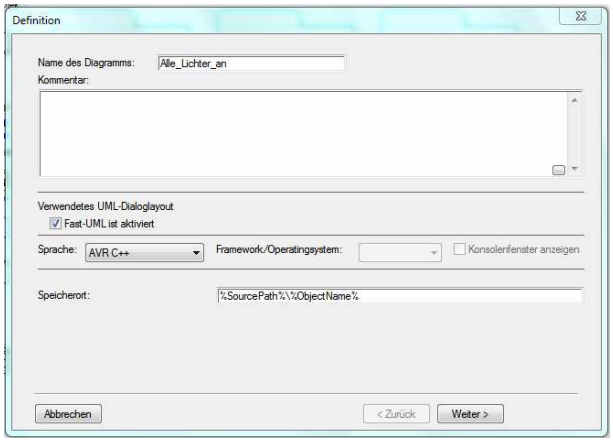
In dem folgenden Dialogfenster werden Ihnen diese Angaben zur Kontrolle angezeigt. Die empfohlene Taktrate ist voreingestellt. Sie können diese ändern.



Als nächstes öffnet SiSy LibStore und bietet Ihnen Vorlagen für die weitere Arbeit an. Aktivieren Sie die Vorlage „PEC Framework – Portable Embedded Classes“ und führen Sie den Download aus; dieser kann eventuell einige Zeit in Anspruch nehmen.

2. C++-Programm anlegen

Für das Erstellen des Programms für den AVR Mikrocontroller ziehen Sie per Drag & Drop aus der Objektbibliothek ein Objekt „Klassendiagramm“ in das Diagrammfenster der geladenen Vorlage. In dem aufgeblendeten Dialogfenster benennen Sie das Klassendiagramm mit dem Namen „Alle_Lichter_an. Verwenden Sie die Sprache „AVR C++“ .

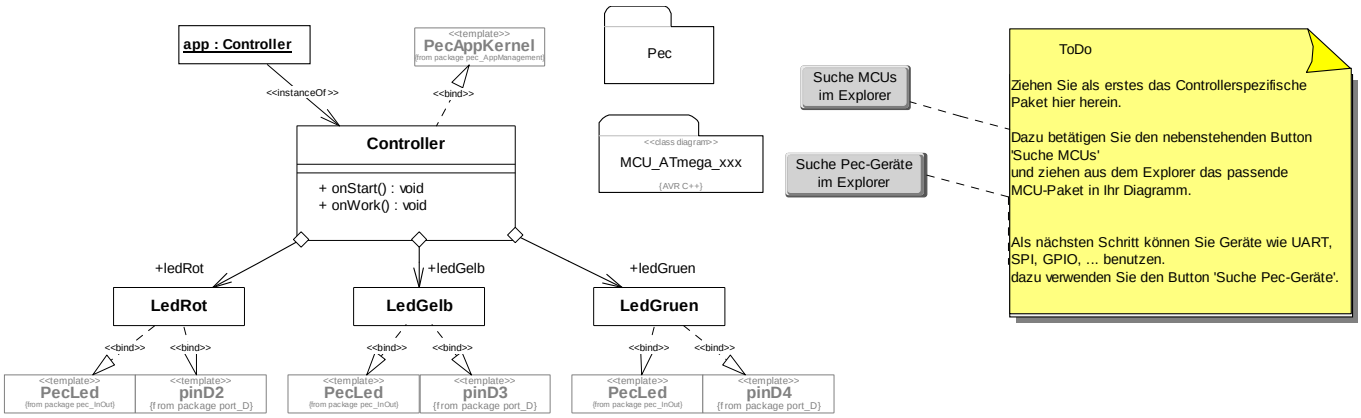


Im folgenden Fenster „Hardware“ nutzen Sie die Vorgaben die in Schritt 1 eingestellt wurden und klicken Sie auf „Weiter“ und „Fertig stellen“. Wählen Sie das soeben angelegte Klassendiagramm und öffnen Sie es mit einem Rechtsklick. Es öffnet wieder der SiSy LibStore. Wählen Sie jetzt die Vorlage „PEC Application Grundgerüst (XMC, STM32, AVR)“ aus.

Die Ausgabegeräte (LEDs) sollen vom Prozessorport GPIOD gesteuert werden. Die Realisierung erfolgt über GPIO Pin 12 und 13.

3. C++-Programm modellieren

Aktivieren Sie im Diagrammfenster die Schaltfläche „Suche MCUs im Explorer“. Oben links erscheint das Fenster „MCU-Explorer“. Ziehen Sie das Objekt „MCU_ATMega_xxx“ in das Diagrammfenster. Aus der Objektbibliothek ziehen Sie ein Objekt „Klasse“ in das Diagrammfenster und vergeben den Namen „LedRot“; nutzen Sie die Standardvorgabe „eine normale Klasse“ und bestätigen Sie Ihre Einstellung. Wiederholen Sie dieses Vorgehen für die Klassen „LedGelb“ und „LedGruen“. Diese neu angelegten Klassen müssen dem Controller zugeordnet werden. Dafür markieren Sie die Klasse „Controller“ und ziehen vom Verteiler (rotes Quadrat) eine Verbindung zur Klasse wählen Sie „Aggregation“ für diese Verbindung. Wiederholen Sie dieses Vorgehen für die beiden anderen Klassen. Schließen Sie oben links den „MCU-Explorer“.
Geben Sie im Explorer-Fenster den Suchbegriff „PecLed“ ein und ziehen Sie das Objekt 3x in das Diagrammfenster. Verbinden Sie jeweils ein „PecLed“ mit der Klasse „LedRot“, „LedGelb“ und „LedGruen“. Den Kanten wird automatisch „Realisierung“ zugewiesen.
Wie bereits in der Zielstellung festgelegt, sollen die LEDs vom Prozessorport D gesteuert werden. Wir nutzen hier die Pins D.2 bis D.4 und verwenden vorbereitete Klassen. Geben Sie im Explorer-Fenster den Suchbegriff „pinD2“ ein. Ziehen Sie das Objekt in das Diagrammfenster, verbinden es mit „LedRot“ und nutzen „Realisierung“. Wiederholen Sie den Vorgang für „pinD3“ mit „LedGelb“ und „pinD4“ für „LedGruen“.
Ihr Diagramm sollte jetzt der folgenden Abbildung entsprechen.



4. Quellcode in C++ erfassen

Bis jetzt wurden die Verbindungen der LEDs zum Port D festgelegt, die LEDs sollen beim Programmstart auch leuchten. Ergänzen Sie die Programmvorlage wie folgt:

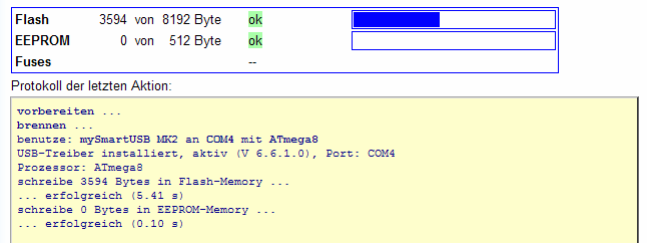
Markieren Sie in der Klasse „Controller“ die Operation „onStart(): void“. Im Beschreibungsfenster geben Sie folgenden Quellcode ein:

```
ledRot.on();
ledGelb.on();
ledGruen.on();
```

5. Programm erstellen, brennen und ausführen

Das myAVR Board verfügt über eine ISP (In System Programming) Schnittstelle, so kann der Prozessor auf dem myAVR Board dirket programmiert werden. Verbinden Sie das myAVR Board über das Programmierkabel mit dem USB-Port Ihres Rechners.

Der eingegebene Quellcode muss nun in Maschinencode für den AVR Prozessor übersetzt werden. Aktivieren Sie dazu im Aktionsmenü der Objektbibliothek den Befehl „Erstellen, Brennen, Ausführen“. Bei fehlerfreier Übersetzung liegt das Programm unter dem Namen „Alle_Lichter_an.elf“ vor und wird automatisch auf den FLASH-Programmspeicher des Prozessors gebrannt. In Abhängigkeit Ihrer Konfiguration erhalten Sie im Ausgabefenster eine entsprechende Meldung über den Fortschritt der Aktion.



Bei fehlerfreien Aktionen verschwindet dieses Fenster und das myAVR ControlCenter wird eingeblendet. Für dieses Beispiel wird das myAVR ContolCenter nicht benötigt, das Fester kann geschlossen werden.

6. Mikrocontrollerlösung testen

Zum Testen des Programms ist es erforderlich, Port D mit den LEDs zu verbinden. Ziehen Sie das USB-Kabel ab.

Verbinden Sie die LEDs mit dem Prozessorport D entsprechend der Abbildung. Nutzen Sie Patchkabel!

Prüfen Sie die Verbindungen und schließen Sie das USB-Kabel an.

Das Programm startet automatisch und die LEDs auf Ihrem Board leuchten.

Viel Erfolg bei weiteren Programmen!