



Zielstellung

Auf dem Board „STM32F4-Discovery“ sollen beim Programmstart die rote und grüne LED in kurzen Zeitabständen aufleuchten und damit ein „Blinklicht“ erzeugen. Der Mikrocontroller ist ein 32-Bit Controller.

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 sollten mit der Bedienung von SiSy und den Diagrammen der UML vertraut sein.

Software

- SiSy Ausgabe STM32 (ARM), Microcontroller++, Professional oder Developer ab der Version 3.6
- Windows XP ... Windows 10
- Installierter USB-Treiber

Hardware

- 1 STM32F4-Discovery
- 1 Programmierkabel (Mini USB-Kabel)

Schaltung

- rote LED an GPIO Port D.12
- grüne LED an GPIO Port D.13

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

1. Ein neues Projekt anlegen

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

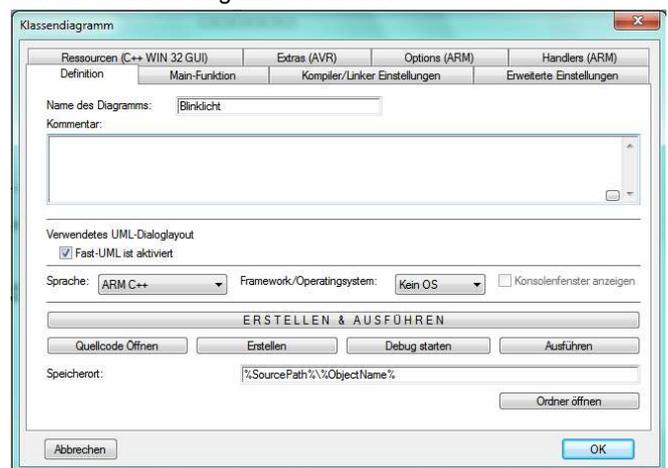


Es öffnet SiSy LibStore und stellt Vorlagen für die weitere Arbeit zur Verfügung. Aktivieren Sie die Vorlage „PEC Framework – Portable Embedded Classes“ und führen Sie den Download aus; dieser kann eventuell einige Zeit in Anspruch nehmen.

2. C++-Programm anlegen

Für das Erstellen des Programms für den STM32 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 „Blinklicht“. Wählen Sie die Sprache ARM C++.

Im nächsten Dialogfenster wählen Sie die verwendete Hardware aus, in unserem Fall „STM32F407-Discovery“ und den Programmierer „ST-Link V2“. Damit sind alle Einstellungen abgeschlossen und ein Programm-Grundgerüst steht zur Verfügung. Zur Kontrolle können alle Einstellungen noch einmal überprüft werden, indem Sie das Objekt „Klassendiagramm“ markieren und mittels rechter Maustaste „Definieren“ auswählen. Sie erhalten folgendes Fenster:



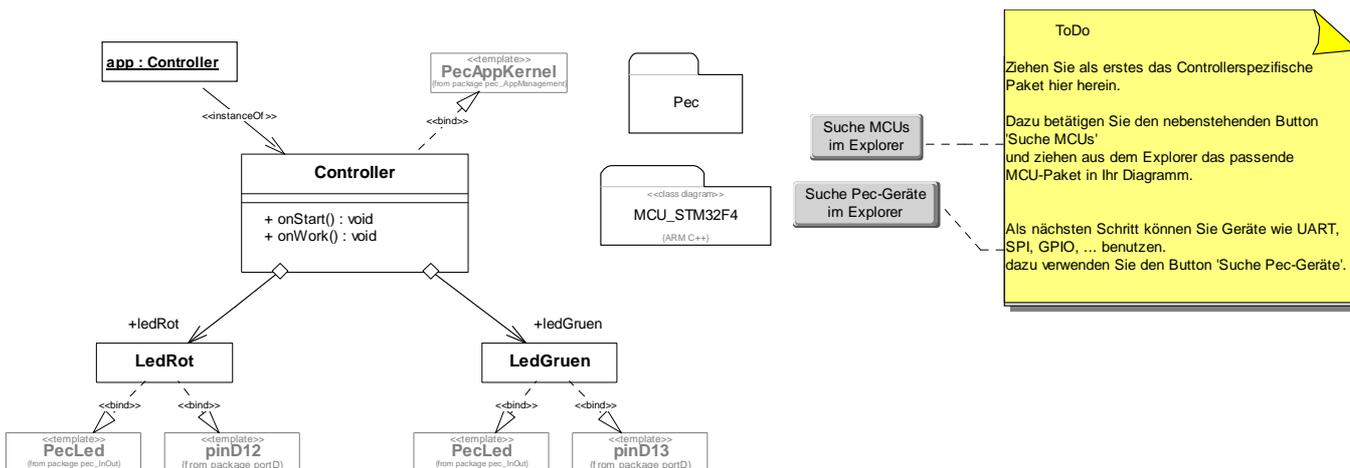
Das geladene Programmgerüst steht Ihnen nun uneingeschränkt für die Weiterverarbeitung zur Verfügung. Die Ausgabegeräte (LEDs) sollen vom Prozessorport GPIO D gesteuert werden. Die Realisierung erfolgt über GPIO Pin 12 und 13.

Wählen Sie das soeben angelegte Klassendiagramm und öffnen Sie es mit einem Rechtsklick. Es öffnet wieder der SiSy LibStore. Wählen Sie jetzt die Vorlage „Application Grundgerüst für PEC Anwendungen (XMC, STM32, AVR)“ und führen Sie den Download aus.

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_STM32F4“ 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 Klasse „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 „LedRot“, wählen Sie „Aggregation“ für diese Verbindung. Wiederholen Sie dieses Vorgehen für die Klasse „LedGruen“. Schließen Sie oben links den „MCU-Explorer“.

Geben Sie im Explorer-Fenster den Suchbegriff „PecLed“ ein und ziehen Sie das Objekt 2x in das Diagrammfenster. Verbinden Sie jeweils ein „PecLed“ mit der Klasse „LedRot“ 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.12 und D.13 und verwenden vorbereitete Klassen. Geben Sie im Explorer-Fenster den Suchbegriff „pinD12“ ein. Ziehen Sie das Objekt in das Diagrammfenster, verbinden es mit „LedRot“ und nutzen „Realisierung“. Wiederholen Sie den Vorgang für „pinD13“ mit „LedGruen“. Ihr Diagramm sollte jetzt der folgenden Abbildung entsprechen.



ToDo

Ziehen Sie als erstes das Controllerspezifische Paket hier herein.

Dazu betätigen Sie den nebenstehenden Button 'Suche MCUs' und ziehen aus dem Explorer das passende MCU-Paket in Ihr Diagramm.

Als nächsten Schritt können Sie Geräte wie UART, SPI, GPIO, ... benutzen. dazu verwenden Sie den Button 'Suche Pec-Geräte'.

4. Quellcode in C++ erfassen

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

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

```

Controller::onStart
    ledRot.blink();
    ledGruen.blink();
  
```

5. Programm erstellen, brennen und ausführen

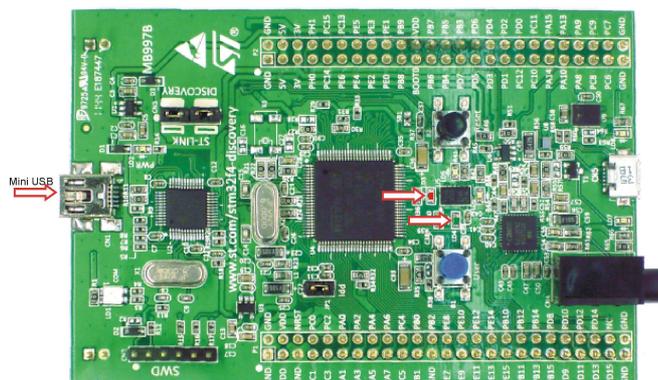
Das Board „STM32F4-Discovery“ verfügt über eine ISP (In System Programming) Schnittstelle, so kann der Prozessor auf dem Board direkt programmiert werden. Verbinden Sie das Board „STM32F4-Discovery“ über das Programmierkabel mit dem USB-Port Ihres Rechners.

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 „Blinklicht.hex“ vor und wird automatisch auf den FLASH-Programmspeicher des Prozessors gebrannt.

In Abhängigkeit Ihrer Konfiguration erhalten Sie im Ausgabefenster eine Fortschrittsmeldung, die bei fehlerfreier Aktion sofort wieder ausgeblendet wird.

6. Mikrocontrollerlösung testen

Zum Testen des Programms ist es erforderlich, dass das Board „STM32F4-Discovery“ angeschlossen ist. Da die LEDs fest mit dem Prozessorport D verbunden sind, erübrigt sich für dieses Beispiel das Patchen von LED zu Pin. Das Programm startet automatisch und die LEDs auf Ihrem Board blinken.



Hinweis:

Nutzen Sie die in SiSy integrierten Tutorials!