



## Zielstellung

Auf dem Infineon-Board „XMC4500 Relax (Lite) Kit“ soll beim Programmstart die rote LED eingeschaltet werden. 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 XMC, Microcontroller++, Professional oder Developer ab der Version 3.6
- Windows XP ... Windows 10
- Installierter USB-Treiber

## Hardware

- XMC4500 Relax (Lite) Kit
- 1 Programmierkabel (Micro USB-Kabel)

## Schaltung

- LED an Pin 1.0

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. Ein neues Projekt anlegen

Starten Sie SiSy und wählen Sie „Neues Projekt erstellen“, vergeben Sie den Projektnamen „Hallo\_XMC“ und wählen Sie das Projektprofil (Vorgehensmodell) „ARM-Vorgehensmodell“ aus.

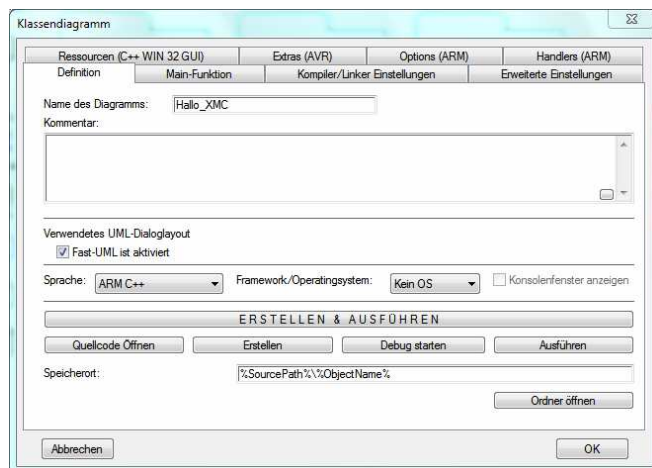


Es öffnet SiSy LibStore und stellt Vorlagen für die weitere Arbeit zur Verfügung. Aktivieren Sie die Vorlage „PEC Framework – Portable Embedded Classes“ und führen Sie den Download aus; dieser kann eventuell einige Zeit in Anspruch nehmen.

## 2. C++-Programm anlegen

Für das Erstellen des Programms für den XMC 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 „Hallo\_XMC“. Wählen Sie die Sprache ARM C++.

Im nächsten Dialogfenster wählen Sie die verwendete Hardware aus, in unserem Fall „XMC4500 Relax Kit“ und den Programmierer „J-Link“. 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:



## 3. Grundlagen

GPIO (General Purpose Input/Output) ist ein allgemeiner Pin, der als Kontakt aus dem Controllergehäuse herausgeführt ist. Der XMC4500 verfügt, je nach Gehäuseform, über die GPIO-Ports PORT 0 bis PORT 6 und über PORT 14 und PORT 15. Die LED ist bereits fest mit dem Pin 1.0 verbunden.

### Hinweis:

Ausführliche Erläuterungen zu den Grundlagen finden Sie in dem „myXMC Lehrbuch“ und in dem „XMC Tutorial“

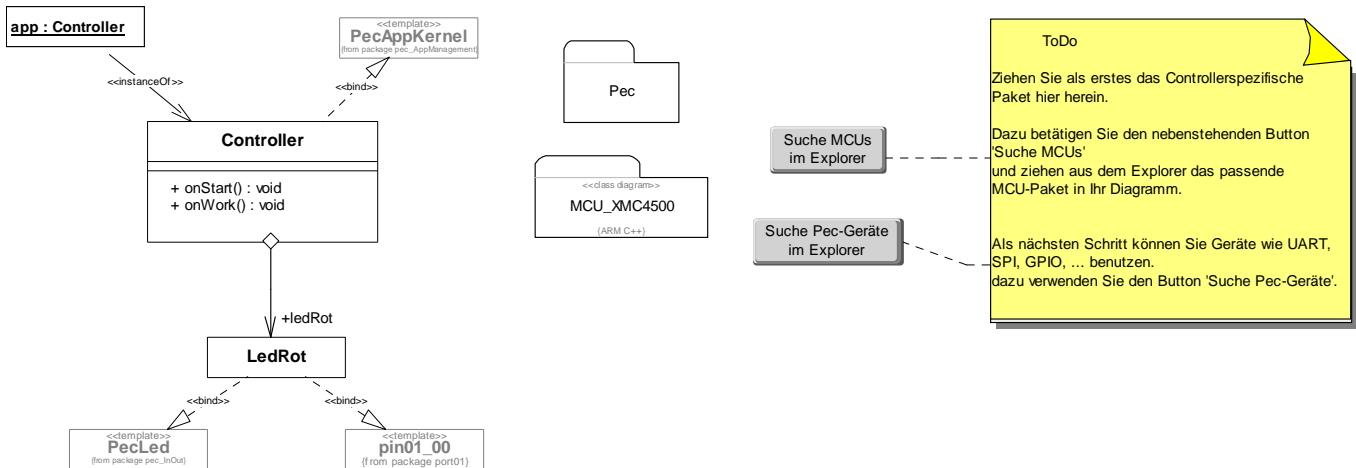
Aktivieren Sie das soeben angelegte Klassendiagramm und öffnen Sie es mit einem Rechtsklick. Es öffnet wieder der SiSy LibStore. Wählen Sie jetzt die Vorlage „PEC Application Grundgerüst (XMC, STM32, AVR)“ und führen Sie den Download aus.

### 4. 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\_XMC4500“ 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. Diese neu angelegte Klasse muß 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. Den „MCU-Explorer“ links oben können Sie schließen.

Geben Sie im Explorer-Fenster den Suchbegriff „PecLed“ ein und ziehen Sie das gefundene Objekt in das Diagrammfenster. Verbinden Sie „PecLed“ mit der Klasse „LedRot“. Der Kante wird automatisch „Realisierung“ zugewiesen. Wie bereits in den Grundlagen beschrieben, ist die rote LED fest mit Pin 1.0 verbunden; hier können Sie eine vorbereitete Klasse verwenden. Geben Sie im Explorer-Fenster den Suchbegriff „pin01-00“ ein. Ziehen Sie das Objekt in das Diagrammfenster, verbinden es mit „LedRot“ und nutzen „Realisierung“.

Ihr Diagramm sollte jetzt der folgenden Abbildung entsprechen



### 5. Quellcode in C++ erfassen

Bis jetzt wurde die Verbindung der LED zum Pin 1.0 festgelegt, die LED soll 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:

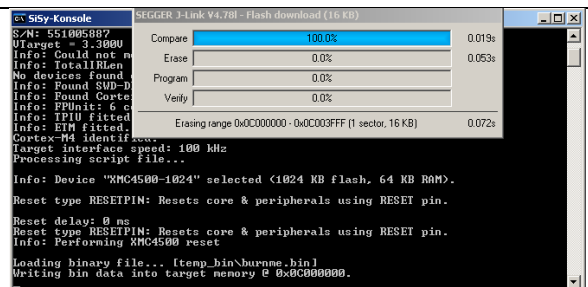
```

Controller::onStart
    ledRot.on();
    
```

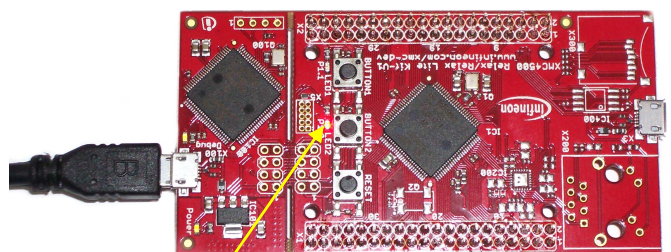
### 6. Programm erstellen, brennen und ausführen

Das Board „XMC4500 Relax (Lite) Kit“ verfügt über eine ISP (In System Programming) Schnittstelle, so kann der Prozessor auf dem Board direkt programmiert werden. Verbinden Sie das Board über das Programmierkabel mit dem USB-Port Ihres Rechners.

Der eingegebene Quellcode muss nun in Maschinencode für den Prozessor übersetzt werden. Aktivieren Sie dazu im Aktionsmenü der Objektbibliothek den Befehl „Erstellen, Brennen, Ausführen“. Bei fehlerfreier Übersetzung liegt das Programm unter dem Namen „Hallo\_XMC.hex“ 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.



Im Anschluß an das „Brennen“ wird das Programm gestartet. In diesem Fall leuchtet die rote LED neben dem mittleren Taster.



Viel Spaß bei weiteren Programmen!

**Hinweis:**  
Nutzen Sie die in SiSy integrierten Tutorials!