

Erstes Assemblerprojekt

1. Neues Assemblerprojekt anlegen (Getting Started)
2. Programmieren

Das Assemblerprogramm hat folgende Grundstruktur:

...

main:

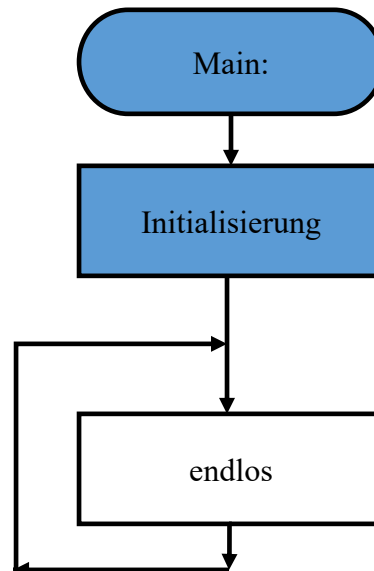
```
bl    startup
ldr   R1,=GPIOC
```

schleife:

```
mov   R0,#0
strb  R0,[R1,ODR]
mov   r0,#1000
bl    HAL_Delay
mov   R0,#0xFF
strb  R0,[R1,ODR]
mov   r0,#1000
bl    HAL_Delay

b     schleife
```

.end



3. Programm assemblieren (übersetzen)

siehe [GettingStartet.docx](#) und Video [GettingStarted.mp4](#) im Verzeichnis Ihrer Entwicklungsumgebung

Beim Assemblieren wird das Programm in Maschinensprache übersetzt.

4. Erläuterung der verwendeten Assemblerbefehle:

main:	Sprungmarke (Label): Sprungmarken können beliebig benannt werden. Sie dürfen nicht mit einer Zahl oder einem Sonderzeichen beginnen. Sie dienen als Ziel von Sprungbefehlen
ldr R1,=GPIOC	Der Port (Anschluss des Mikrocontrollers) wird in R1 ausgewählt Hinweis: GPIOC bedeutet General Purpose Input Output Port C
mov R0,#1000	Move heißt bewege die Zahl 1000 in das Register (die Variable R0) Äquivalent wäre: mov R0,#0x3E8 //Hexadezimalzahl mov R0,#0b001111101000 //Dualzahl # bedeutet: Unmittelbare Adressierung (immediate). Was schlicht und einfach bedeutet: Jetzt kommt eine Zahl Die Zahl 1000 bedeutet in diesem Zusammenhang, dass das Unterprogramm wait_ms 1000ms (=1s) warten soll.
bl HAL_Delay	Das Unterprogramm HAL_Delay wird aufgerufen: bl = Branch Linked

schleife:	Noch eine Sprungmarke, diesmal für die Endlosschleife
mov R0,#0 strb R0,[R1,ODR]	Port LEDs werden auf 0 gesetzt. 0 bedeutet, dass der entsprechende Anschluss des Mikrocontrollers eine Spannung von 0V ausgibt. strb R0,[R1,ODR] gibt den Inhalt von Register R0 auf GPIOC.ODR aus, da zuvor GPIOC in R1 eingetragen wurde.
mov R0,#0xFF strb R0,[R1,ODR]	Port LED wird auf 1 gesetzt. Das bedeutet, dass der entsprechende Anschluss des Mikrocontrollers eine Spannung von 5V ausgibt und die angeschlossene Leuchtdiode (LED) leuchtet
b schleife	branch = Verzweige zu. Sprungbefehl (Goto in C). Das Sprungziel ist die Sprungmarke <i>schleife</i> :
.end	Zeigt dem Assembler an, dass das Assemblerprogramm hier endet.

4. Aufgaben

- Erstellen Sie aus der Vorlage ein neues Projekt mit dem Namen: erstesAssemblerprogramm:
- Bringen Sie das Programm auf Ihrem Mikrocontroller zum Laufen.
 - Dazu muss der Mikrocontroller an USB angeschlossen sein
 - Und die Treiber installiert sein
- Versuchen Sie ein Lauflicht mit verschiedenen Mustern
- Entfernen sie die beiden Anweisungen **bl HAL_Delay**, welches Verhalten zeigt das Programm jetzt
- Programmieren Sie die [Ampel](#)

5. Fragen

- Wie werden Unterprogramme aufgerufen?
- Was bewirkt die Anweisung **b schleife**?
- Welches sind Anweisungen, die dem Datentransport dienen?
- Welche Anweisungen werden nach dem Einschalten nur einmal ausgeführt?
- Welche Anweisungen werden endlos wiederholt? Warum?
- Warum wird ein Unterprogramm HAL_Delay benötigt?
- Wie lautet die Sprunganweisung?

6. Diskussion und Präsentation

Tragen Sie Ihre Ergebnisse im Assembler Wiki auf der Seite: **Erstes Assemblerprojekt** ein.
Diskutieren Sie Ihre Lösungen mit Ihrer Nebenfrau (-mann)
Präsentieren Sie Ihre Lösung im Plenum.