

Übung 10

Startup-Code, Projekte in C- und Assembler-Code

Ziele dieser Übung:

Sie sollen in der Lage sein,

- 1) Den Startup-Code nachzuvollziehen und allenfalls zu modifizieren.
- 2) Ein kleines Projekt mit C- und Assembler-Dateien zu programmieren.

Verwenden Sie für den Test Ihres Codes den Debugger und das Leguan-Board.

Aufgabe 10.1 Startup-Code, initialisierte und nichtinitialisierte Variablen

In dieser Aufgabe soll das Aufstarten eines Programms mit dem Debugger nachvollzogen werden.

- 1) Generieren Sie in STM32CubeIDE ein neues C-Projekt.
- 2) Fügen Sie einige initialisierte und nichtinitialisierte Variable hinzu.
- 3) Im Map-File können Sie prüfen, auf welchen Adressen ihre Variablen angelegt werden.
- 4) Prüfen Sie den Startup-Code (Datei startup_stm32h743zitx.s). Wo werden die Variablen initialisiert? Setzen Sie an dieser Stelle im Code einen Breakpoint. Die Startup-Datei finden Sie im Projekt im folgenden Pfad: "lib\hal\Core\Startup\startup_stm32h743zitx.s".
- 5) Öffnen Sie im Debugger das Memory-Fenster und zeigen Sie die Adressen an, wo Ihre Variablen angelegt wurden. Vollziehen Sie die Initialisierung der Variablen nach.

Aufgabe 10.2 Projekte mit C- und Assembler-Code

Generieren Sie ein neues C-Projekt und fügen Sie dem Projekt ein zusätzliches Assembler-File hinzu.

- 1) Das Assembler-File soll folgende Variablen und Subroutinen enthalten:
 - Eine word-Variable "asm_var".
 - Eine Subroutine mit der Deklaration int asm_sub (int a, int b), welche den Wert (a-b) zurück gibt.
 - Eine Subroutine asm_callcfunc, welche die C-Funktion cfunc(int a, int b) aufruft und ihr als Parameter die Werte 1 und 2 übergibt.
 - Eine Subroutine asm_modifyvar, welche der Variablen c_var den Wert 0x55 und der Variablen asm_var den Wert 0x11 zuweist.
- 2) Das C-File soll folgende Variablen und Funktionen enthalten:
 - Die word-Variable "c_var".
 - Eine Funktion mit dem Prototyp int cfunc (int a, int b), welche den Wert (a+b) zurück gibt.
- 3) Schreiben Sie das Hauptprogramm in C, welches folgenden Code enthält:

```
int main(void){  
  
    /* local variables (auto) will be on stack */  
    int a = 5;  
    int b = 27;  
    int res;  
  
    /**  
     * endless loop  
     */  
    for(;;){  
        res = asm_sub(a, b);  
        asm_callcfunc();  
        asm_modifyvar();  
        asm_var = 0x22;  
        c_var = 0xAA;  
    }  
    return(0);  
}
```

- 4) Debuggen Sie dieses Programm und prüfen Sie die Werte der Variablen.
- 5) Prüfen Sie im Debugger im Disassembly-Fenster den vom C-Compiler generierten Assembler-Code für den Aufruf von `asm_sub`.
- 6) Vergleichen Sie den vom C-Compiler generierten Assembler-Code der Funktion `cfunc()` mit Ihrem Assembler-Code von `asm_sub`. Was stellen Sie fest?