

Allgemeine Programmstruktur (Assemblerprogrammierung)

Programmkopf

Kurze, funktionelle Beschreibung des Programmes.

Muss als Kommentar geschrieben werden.
Hat keine funktionale Bedeutung.

- Nennung der Zielfunktion
(evtl. Nennung von Teilfunktionen)
- Nennung des Autors/Bearbeiters
- Erstellungs-/Änderungsdatum

Assemblerzuweisungen

Werden zur Laufzeit des Übersetzungsvorgangs genutzt, erzeugen keinen Code!

Beispiele

- Zielprozessor definieren
(erzeugt den Maschinencode für einen bestimmten Typ)
- Dateien einfügen (include; fügt den Quelltext weiterer Dateien ein)
- Bezeichner (Label, Marken) einen Ausdruck (konstant) zuweisen (equal; Label:=Ausdruck)

Speicherzuweisung

Reserviert Speicher im Daten- oder Programmspeicher. Vorbelegung des Speicherplatzes mit Werten möglich und/oder notwendig. Zuweisung von Bezeichnern (Label) möglich.

Datenspeicher

- Variablenvereinbarung
(variable Daten im Datenspeicher, z.B. Messwerte)

Datensegment auswählen
Zieladresse auswählen
Variablen vereinbaren

Programmspeicher

- Konstantenvereinbarung
(feste Werte im Programmspeicher, z.B. ASCII-Zeichen, Korrektortabellen)

Codesegment auswählen
Zieladresse auswählen
Variablen vereinbaren

Programmcode

Sequenzielle Anordnung von Befehlen, zur Steuerung der gesamten Funktionalität des Programms. Inhalt und Umfang werden durch die zu realisierende Applikation bestimmt. Der erzeugte Code wird im Programmspeicher abgelegt.

Der Quelltext muss funktional strukturiert (Unterprogrammnutzung) und mit funktionsbezogenen Marken (Label) und Kommentaren versehen sein.

Am Beginn sollte eine Zieladresse (Codesegment) vereinbart werden!

Initialisierungen

Initialisierung und Aktivierung der für die Funktionalität des Programmes erforderlichen On-Chip-Komponenten.

Bei Bedarf laden von Startwerten.

Wird typischerweise nur einmal durchlaufen!

Hauptprogramm

Beginnt an einer Anfangsadresse (Label) und endet in einer Schleifenfunktion.

Anfadr:



Befehl 1
Befehl 2
Befehl 3
.
Befehl n
Sprungbefehl
(Schleifenfunktion)

Unterprogramme

Programmabschnitte, welche wichtige Teilfunktionen realisieren. Werden aus dem Hauptprogramm, Unterprogrammen oder ISR aufgerufen.

Beginnen mit einer funktionsbezogenen Adressmarke (Label) und enden mit einem Rückkehrbefehl. Enthalten nie eine Endlosschleife!

UP_Mess:

Befehl 1
Befehl 2
Befehl n
Rückkehrbefehl

Interruptservice-Routinen

Programmabschnitte, die interruptgesteuerte Teilfunktionen realisieren. Werden über die Interruptvektortabelle aufgerufen.

Beginnen mit einer funktionsbezogenen Adressmarke (Label) und enden mit einem Rückkehrbefehl. Enthalten nie eine Endlosschleife!

In der ISR benutzte Register müssen gesichert werden.

ISR_Timer1:

Register retten
Befehl
Befehl
.
Rettung aufheben
Rückkehrbefehl

Interruptvektortabelle (IVT)

Ist bei Bedarf einzurichten.

Adresse Befehl mit Sprungziel

; hier beginnt die CPU nach dem Reset
0000: Sprung zum Ende der IVT
; hier wird automatisch bei Interruptereignis 1 hin verzweigt
0001: Sprung zu ISR1
; hier wird automatisch bei Interruptereignis 2 hin verzweigt
0002: Sprung zu ISR2
*
; hier wird automatisch bei Interruptereignis n hin verzweigt
000n: Sprung zu ISRn

On-Chip-Komponenten

Sind bei Bedarf einzurichten.

Initialisiert werden die entsprechenden Spezialfunktionregister.

mögliche Komponenten

- Interruptsystem
- Ports (Ein-/Ausgabe)
- Schnittstellen (UART, I²C, SPI)
- Timer/Counter
- ADU, DAU

Startwerte

Sind bei Bedarf einzurichten.

Register und/oder Variablen sind mit entsprechenden Zahlenwerten zu laden.