Anleitung zur Struktur eines Programms

Die allermeisten Mikroprozessorprogramme bestehen aus vier logischen Teilen, deren Zweck an beispielhaften Befehlen in der rechten Spalte erläutert wird:

Teil A: Anweisungen an das Übersetzerprogramm (Assembler)

Code (beispielhaft)	Zweck
;rahmen.asm ;Version 25.02.2011 ;Autor: Hartwig Harm DH2MIC .include "tn13def.inc"	Programmname, Versionsdatum und Verfasser sowie Include-File zur Festlegung des Prozessors, für den das Programm bestimmt ist.
.def Wartezeit = r16 .def tmp = r17 .def delay = r03	Definitionen von "sprechenden Namen" für reservierte Namen von Registern, Ports, Variablen, etc.
.equ Frequenz = 123	Zuweisung eines konstanten Wertes zu einem "sprechenden Begriff"
.org $0000	Origin (Beginn) legt die Anfangsadresse des folgenden Codes (Programcounter PC) fest

Teil B: Startbereich des Programms, der nur ein Mal ausgeführt wird

rjmp Start	Sprung von der ersten Adresse (meist 0) auf den eigentlichen Programmbeginn. Dies ist der RESET-Interruptvektor, also die Adresse, an der der Prozessor nach einem RESET seine Arbeit beginnt.
rjmp Interruptroutine1	Sprung auf das Unterprogramm zur Abarbeitung des Interrupt 1
rjmp (weitere Routinen)	Insgesamt gibt es 10 Interrupts
.org $000A Start:	Label - das ist eine Sprungadresse zu der unbedingt, bedingt (abhängig von einer logischen Entscheidung) oder in Form eines Unterprogrammaufrufes verzweigt wird. Hier der Programmbeginn mit einem Origin unmittelbar nach dem Interruptvektorbereich.
ldi tmp, 0b00010101 out ddrb, tmp	Festlegung der Eigenschaften von Ports, hier werden PB4, PB2 und PB0 zu Ausgängen; PB3 und PB1 bleiben Eingänge (default)
loop: sbis pinb, 1 rjmp loop rcall warte_prellen_ab	Programmbereich, mit dem abgewartet wird bis die Starttaste am Port B Pin1, mit der auch die Stromversorgung des ATtiny13 eingeschaltet wurde, wieder gelöst ist (auf HIGH geht). Besonderheit der Würfelplatine!
ldi tmp, 0b00000000 out portb, tmp	Befehlssequenz, mit der alle als Ausgang deklarierten Portanschlüsse auf LOW (GND) geschaltet werden. Für die Eingänge (Inputs) hat das keine Auswirkungen.
ldi tmp, Frequenz mov delay, tmp	Register mit Anfangswerten besetzen. Register r00 mit r15 können nur mit Zwischenschritt über eines der Register r16 mit r31 gesetzt werden

Teil C: Hauptprogramm

Schleife:	Erster Label, auf den aus dem Hauptprogramm immer wieder zurückgesprungen wird
(Gruppe von Befehlen)	Befehle des Hauptprogramms
rjmp Schleife	Letzter Befehl des Hauptprogramms

Teil D: Unterprogramme einschließlich Interrupt-Service-Routinen

Unterprogramm1:	Erster Label, auf den aus dem Hauptprogramm mit einem rcall Befehl verzweigt werden kann
(Gruppe von Befehlen)	Befehle des Unterprogramms
ret	Letzter Befehl des Unterprogramms (Return). Kehrt zu der Stelle im aufrufenden Programm zurück, die auf den Aufruf "rcall Unterprogramm1" folgt. Bei Interrupt-Routinen lautet der letzte Befehl "reti" (Return from Interrupt)
Warten: ldi r16,40 Warten1: ldi r17,250 Warten2: ;innere Schleife dec r17 ; 1 Takt brne Warten2 ; 2 Takte dec r16 brne Warten1 ret ;Rücksprung	Es können (fast) beliebig viele Unterprogramme folgen. Die Reihenfolge ist beliebig. Es ist aber zu empfehlen, zusammen gehörige Teile auch zusammen zu lassen. Nebenstehende Wartezeitroputine wartet übrigens etwas über 40x250x3=30000 Takte Interruptroutinen stehen oft am Schluss oder auch ganz am Anfang. Sie enden statt mit 'ret' mit 'reti'.

Bitte immer Kommentare (beginnen mit einem Semikolon) einbauen, damit man später noch versteht, was man sich beim Schreiben gedacht hat. Und zwar so, dass es auch ein anderer Programmierer versteht!

Wenn Euch etwas auffällt das fehlt: Bitte sagen!

vy 55 de Hartwig DH2MIC

email: dh2mic@darc.de