

MIP

der Initialisierungs-Wizzard für Programme für den c167 Mikrocontroller

Franz Fiala

LIT 94

Bevor eine Idee auf einem Mikrocontroller als Programm auch wirklich läuft, muß eine wichtige Hürde überwunden werden: die Initialisierung aller beteiligten Hardwareeinheiten, allen voran jene im Mikrocontroller selbst wie der Timer, Interrupts, Schnittstellen... Die richtigen Initialisierungsparameter einzustellen, ist geprägt durch Tests kurzer Programmsequenzen, stets bewaffnet mit einem Datenblatt, immer hoffend, jetzt möge sich doch schon endlich dieser Timer einschalten oder möge jener Interrupt ausgelöst werden.

Das MIP (Mikrocontroller Initialisierungs-Programm) gibt dem Anfänger wie dem Profi ein wertvolles Hilfsmittel zur menügeführten Generierung einer Initialisierungsfunktion `Ini_Fkt()` in die Hand.

Alle Einstellungen im Controller bilden sich in Menüs ab.

Übersicht zur MIP-Anwendung

Mit Hilfe des Mikrocontroller-Initialisierungs-Programms (MIP) wird dem c167-Programmierer die Initialisierung der Peripheriekomponenten erleichtert. Hierzu sind folgende Schritte nötig: Konfiguration der gewünschten Peripheriekomponenten unter der grafischen Oberfläche des MIP-Tools. Abspeichern der Konfiguration im gewünschten Arbeitsverzeichnis unter dem Namen `datei.name.cfg`. Einstellen der gewünschten Export-Optionen. Exportieren der gespeicherten Konfiguration zum entsprechenden C-Quellcode. Der exportierte C-Quellcode wird unter dem Namen `datei.name.c` abgespeichert.

Die Initialisierung mit dem MIP-Tool ist an diesem Punkt abgeschlossen. Die erzeugte Initialisierungs-Datei (`datei.name.c`) kann nun im C-Programm des c167-Programmierers eingebunden, und die dort enthaltene Initialisierungs-Funktion aufgerufen werden.

Das Beispielprogramm

Es soll ein einfacher Frequenzzähler mit dem Timerblock GPT1 erzeugt werden. Timer 3 und Timer 2 sollen als 32-Bit Counter initialisiert werden und mit Hilfe von Timer 4 sollen jede Sekunde die Register T3 und T2 ausgelesen und abgespeichert werden. Das Eingangssignal muß an T3IN = P3.6 angelegt werden.

Folgende Einstellungen müssen nun getätigt werden, um die oben genannten Timerfunktionen zu erhalten:

Betriebsart (T3M)	Counter
Zähltakt oder Flankenwahl (T3I)	Steigende (Fallende) Flanke
Timer/Counter 3 starten (T3R)	aktivieren

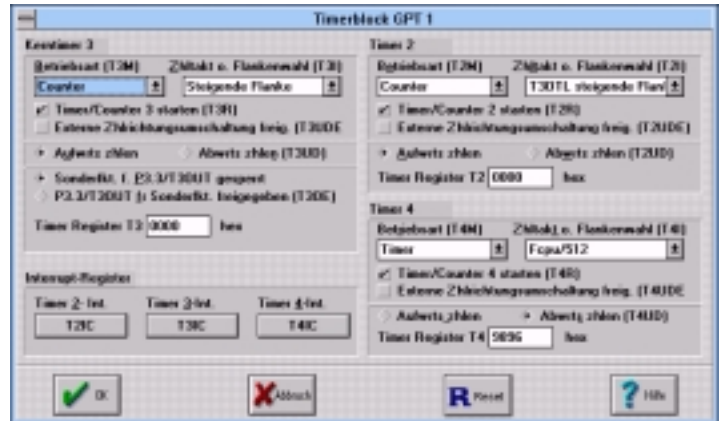
Betriebsart (T2M)	Counter
Zähltakt oder Flankenwahl (T2I)	T30TL beide Flanken
Timer/Counter 3 starten (T2R)	aktivieren

Betriebsart (T4M)	Timer
Zähltakt oder Flankenwahl (T4I)	Fcpu/5 12
Timer/Counter 3 starten (T4R)	aktivieren
Abwärts zählen (T4UD)	aktivieren
Timer Register T4	9896 hex
(20 MHz (CPU-Frequenz)/5 12)	

Diese Einstellungen werden in genau dieser Art in einem Menü eingetragen.

Erzeugung des Initialisierungs-Codes mit 'MIP'

Nach dem Aufruf von 'MIP' muß zunächst der Menüpunkt 'Datei'-'Neu' ausgewählt werden, um einen leeren Datensatz zu erzeugen. Anschließend wird über die Menüpunkte 'Peripherie'-'Timer'-'Timereinheit 1 (GPT1)' die entsprechende Dialogbox geöffnet:



Das Ergebnis kann sich sehen lassen! Kein Stöbern in Datenblättern, keine Bitjonglierereien, alles erledigt MIP und erzeugt eine saubere Initialisierungsfunktion `Ini_Fkt()`:

```

/***** C-Quellcode *****/
/* Erstellt am 30.08.1995 um 01:42:57 */

#include "NONAME.H"

void Ini_Fkt(void)
{
    /* Initialisierung des Timerblocks GPT1 */
    _bfd(T2CON, 0x0007, (INI_T_T21 << 0));
    _bfd(T2CON, 0x0038, (INI_T_T2M << 3));
    _bfd(T3CON, 0x0007, (INI_T_T31 << 0));
    _bfd(T3CON, 0x0018, (INI_T_T3M << 3));
    _bfd(T4CON, 0x0007, (INI_T_T41 << 0));
    T4UD = INI_T_T4UD;
    T4 = INI_T_T4;
    T41C = INI_T_T41C;
    T2R = INI_T_T2R;
    T3R = INI_T_T3R;
    T4R = INI_T_T4R;
    _einit();
}

```

Auch bei der Auswahl eines bereits gespeicherten Projekts erhält man eine übersichtliche Darstellung, welche Werte einer CPU initialisiert werden:

```

BUS      Resetwert
BUS1     Resetwert
BUS2     Resetwert
BUS3     Resetwert
BUS4     Resetwert
STACK    Resetwert
ASC0     Resetwert
SSC      Resetwert
GPT1     veränderte Parameter: T2I, T2M, T3I, T3M, T4I, T4UD, T4, T4C, T2R, T3R, T4R,
GPT2     Resetwert
CAPCOM1  Resetwert
CAPCOM2  Resetwert
WD1      Resetwert
PWM      Resetwert
ADU      Resetwert
Interrupts veränderte Parameter: T4C,
Ports    Resetwert

```

Didaktisch wertvoll

Dieses Tool besitzt auch einen erheblichen didaktischen Wert. Auswirkungen von Initialisierungsmaßnahmen am Initialisierungscode können unmittelbar veranschaulicht werden. Bitmanipulationen in C werden demonstriert.

Praktisch Freeware

Ein weiteres Feature dieses nützlichen Programms ist der Preis. Um 100,- Schilling (einfach Ihrer Bestellung beilegen) kann dieses hilfreiche Tool, gemeinsam mit einer Beschreibung bei Firma WILLERT SOFTWARE TOOLS GmbH, Herminenstraße 17b, D-31675 Bückeburg, Tel.: 0049-57 22-2 40 81, Fax: 0049-57 22-2 40 83 bezogen werden. Eine komplette Darstellung der Eigenschaften können Sie über LIT-xx bei der Redaktion der *PCNEWS edit* beziehen. □