

Der Mikrocontroller R8C/13

Ein Schnelleinstieg

Um schnell und kostengünstig mit einem Mikrocontroller vertraut zu werden und zu einem Programmier-Erfolg zu kommen, eignet sich besonders der R8C/13 von Renesas.

- Der SMD-Baustein ist auf einem 32poligen DIL-Carrier mit 20 MHz-Quarz für ca. 13 Euro erhältlich. Intern arbeitet der R8C/13 mit 16 Bit. Er hat 16 KByte Programmspeicher, 2 x 2 KByte Datenspeicher und 1 KByte RAM. (www.reichelt.de, www.glyn.de, www.elektor.de)
- Man erhält kostenlos die nötige Software, und zwar den Monitor/Debugger KD30, den C-Compiler NC30 mit Entwicklungsumgebung HEW (mitgelieferte CD).
- Das compilierte Programm kann über eine serielle Schnittstelle vom PC in den Flash-Speicher des R8C geschrieben werden. Auch die nötige Software, das Flash Development Toolkit FDT gibt es kostenlos (mitgelieferte CD).
- Die Zeitschrift Elektor hat ab Heft 11/2005 eine Artikelserie über die Arbeit mit dem R8C gebracht (www.elektor.de). In Heft 12/05 sind die Installation der Software und der Einstieg in die Programmierung beschrieben.

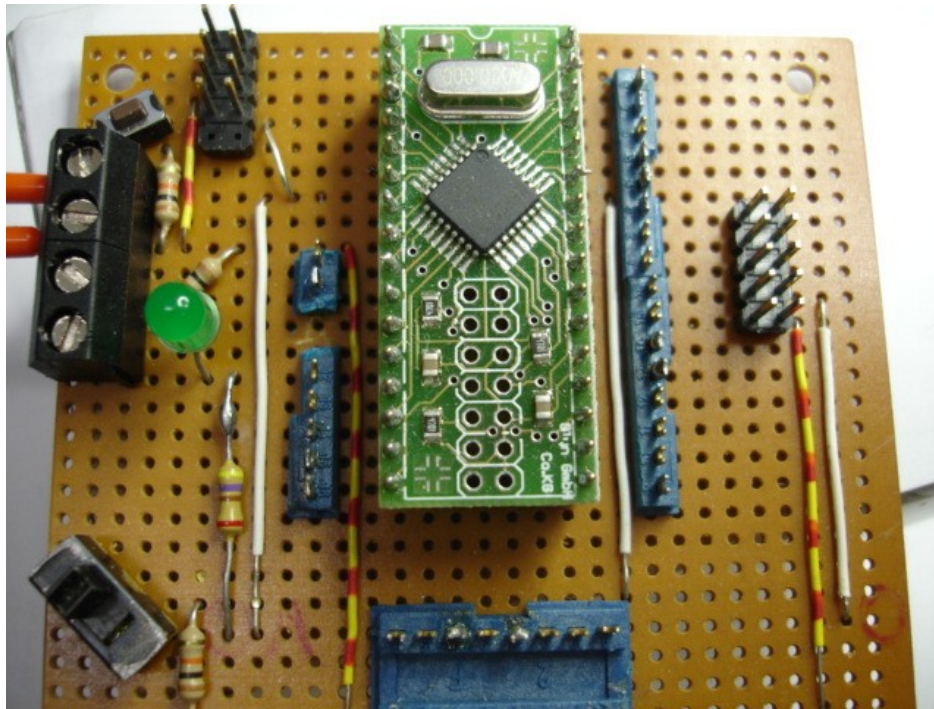
Hardware

Die mitgelieferten Pfostenstiftreihen müssen in den Carrier eingelötet werden. Damit sie richtig ausgerichtet sind, steckt man sie zuerst in eine 32polige DIL-Fassung und lötet dann.

Die Pin-Nummern des Carriers entsprechen denen des R8C-SMD-Bausteins.

Die Schaltung auf dem Carrier ist ein 20 MHz-Quarzoszillator. (Der R8C hat auch einen internen 8 MHz-Oszillator) Ein Pfostenstecker zum Anschluss eines externen Emulator/Debuggers kann eingelötet werden.

Die meisten Anschlüsse des R8C sind mit Bits von Ports verbunden, haben aber je nach Konfiguration auch andere Funktionen. Um diese zu ergründen, muss man das über 200 Seiten umfassende Datenblatt (auf CD) studieren. Einen ersten Hinweis gibt auch die genaue Pin-Bezeichnung. Manche Konfigurationen sind nur erfolgreich, wenn zuvor in einem Protect-Register der Schutz aufgehoben wird, was z.B. erforderlich ist, um die Datenrichtung der Bits des Registers P0 zu ändern!



Der R8C/13 Carrier in einer Experimentierschaltung

Für den Einstieg genügt eine Schaltung, ähnlich der oben dargestellten, die die folgenden Pins verwendet:

Pin	Funktion	Bezug zur Schaltung oben
1	RxD1 Eing. serielle Schnittstelle 1 (Flashen vom PC)	Pfostenstecker links oben
32	TxD11 Ausg. serielle Schnittstelle 1 (Flashen vom PC)	
3	Reset (0 V)	Taster links oben
28	Mode (VCC Normal, 0 V Flash-Mode jeweils nach Reset)	Schalter links unten
7	VCC Stromversorgung 3 - 5,5 V	die beiden oberen Löt клемmen
5	VSS Stromversorgung 0 V	
10	RxD0 Eingang der seriellen Schnittstelle 0	die beiden unteren Löt клемmen
11	TxD0 Ausgang der seriellen Schnittstelle 0	

Die grüne LED ist an ein Bit von Port P1 angeschlossen, das als LED-Treiber konfiguriert werden kann.

Zum Anschluss der seriellen Schnittstelle 1 an den PC ist ein RS232-Interface erforderlich. Am besten verwendet man (fliegend) den Baustein MAX232, der über den Pfostenstecker links oben mit Strom versorgt werden kann.

Software

Fertige Programme

Am schnellsten kommt man zu einem funktionsfähigen System, indem man ein fertiges Programm verwendet. Die Flash-Datei eines Programms wird auch als Motorola-File bezeichnet und trägt die Dateinamenerweiterung „mot“. Die mot-Datei ist das Endprodukt eines Compilervorgangs.

Für den Flash-Vorgang muss auf dem PC das Flash-Development-Toolkit (FDT) installiert werden. Dazu wird die Datei auf der CD \Software\Flasher_FDT\fdtv304r00.exe gestartet. Man erhält zwei Programme FDT und FDT Basic. Es genügt, nur mit FDT Basic zu arbeiten.

Am Ende des Installationsvorgangs müssen Einstellungen vorgenommen werden, was aber auch später noch möglich ist (Options / New Settings). Hier die wichtigsten:

- Die Bezeichnung der Renesas Flash Device: R5F21134
- Die serielle Schnittstelle des PC, über die geflasht werden soll: COM1, COM2
- Speed: 9600 Baud

Zu empfehlen ist auch, unter Options ein Häkchen bei „Auto disconnect“ zu setzen. Und: ohne die Option „Erase Device Before Program“ geht das Flashen nur beim ersten Mal.

Nachdem die serielle Verbindung hergestellt ist, wird beim R8C der Flash-Mode erzeugt (0 V an Mode-Pin danach Reset). Unter „User Area“ im FDT wird der Pfad und der Name der zu übertragenden mot-Datei eingetragen und dann kann mit Program Flash die Programmierung gestartet werden.

Im unteren Teil des FDT-Fensters wird der ganze Vorgang protokolliert. Wenn es zum Abbruch wegen eines Fehlers kommt, hat das wahrscheinlich mit einer unzureichenden Verbindung zu tun. Vor jedem neuen Flash-Versuch sollte der Flash-Mode beim R8C neu erzeugt werden.

Nachdem das Flashen gelungen ist, wird der Mode-Schalter zurückgestellt und ein Reset ausgeführt, der das Programm startet.

Achtung!

Die neuere Version 3.07 des FDT bietet von Hause aus in der Konfiguration nur den Emulator E8, aber keine serielle Schnittstelle für die Übertragung an.

Das kann man ändern, indem man die Konfigurationsdatei R5F21134.fcf (für R8C/13) entsprechend anpasst. Diese Datei ist zu finden in einem ewig langen PfadRenesas.... FDT3.07/Kernels/ProtD.... Am besten sucht man sie. Die Checksum-Zeilen am Ende müssen an den Anfang geschoben werden, und die Checksum ist gleich 0 zu setzen. Die Zeile Com = No ist zu ändern in Com = Yes.

Natürlich muss FDT3.07 unter Options/New Settings neu konfiguriert werden.

Programmentwicklung

Dazu braucht man die IDE (Integrated Development Environment) von Renesas, genannt HEW (High-performance Embedded Workshop). Man installiert von der CD aus dem Ordner Software, und zwar in der angegebenen Reihenfolge:

-
- Monitor/Debugger KD30
Start von \Software\kd30400r1\KD30V410R1_E_20041203.exe (Pfad C:\MTOOL bestätigen)
 - C-Compiler NC30 mit dem HEW
Start von \Software\nc30v530r0_hew\nc30wav530r02_2_ev.exe (Alle Vorschläge annehmen. Site-Code für die Registrierung nicht beachten.
AutoUpdate durch Renesas bestätigen und gleich durchführen lassen. Das kann 10 min dauern. Es gibt keine aussagekräftige Fortschrittsanzeige. Wenn das AutoUpdate gelungen ist, kann auf das Update von der CD, die ohnehin veraltet sein kann, verzichtet werden.)
 - Debugger-Package (integriert KD30 in den HEW)
Start von \Software\Debugger Package\m16cdebuggerv100r01.exe (Neustart des Rechners)

Der High-performance Embedded Workshop aus der Programmgruppe Renesas kann gestartet werden.

Projekte (auch Project Workspace) liegen in je einem Ordner gleichen Namens. Diese Ordner befinden sich im Ordner Workspace. Ein Projekt wird verkörpert durch die Datei *.hws, die auch mit Pfad anzugeben ist, wenn man ein vorhandenes Projekt öffnen will.

Aber zuvor muss ein neues Projekt erzeugt werden. Dazu sind Angaben in mehreren Fenstern zu machen:

Unter „Workspace Name“ ist der Name des Projektes anzugeben und bei „CPU family“ ist M16C zu wählen. Im nächsten Fenster ist wichtig, unter „CPU series“ R8C/Tiny einzutragen. Die Voreinstellungen in den anderen Fenstern können übernommen werden, sofern man nicht debuggen will.

Nach Fertigstellung erscheint das Fenster der HEW. Dort ist vom Projekttyp Debug auf Release zu wechseln. Man erkennt, dass die beiden (immer benötigten) Assemblerdateien ncr0.a30 und sect30.inc bereits im Projekt vorhanden sind. Aber auch ein C-Hauptprogramm-Rumpf ist zu sehen. Dieses Programm kann man öffnen, erweitern und kompilieren (Button Build All).

So vorzugehen, wird eher die Ausnahme sein. Man hat vielmehr einen eigenen Hauptprogramm-Rumpf und einzubindende Header- und Quelldateien. Diese Dateien kopiert man in den betreffenden Ordner. Das vom HEW erzeugte C-Hauptprogramm ersetzt man durch das eigene (Löschen und das eigene entsprechend umbenennen). Auch die Datei sect30.inc ist ggf. zu überschreiben. Damit diese Dateien auch zum Projekt gehören, wird Project / Add Files ausgeführt.

Der Compiler wird immer eine Warnung zeigen, auch wenn das Programm fehlerlos ist. Die Warnung bezieht sich auf die begrenzte Verwendungsdauer des HEW. Was der Compiler nicht weiß: unter 64 K Programmcode gibt es keine Begrenzung. Also ist die Warnung ohne Bedeutung.

Die vom Compiler erzeugte mot-Datei, beispielweise für ein Projekt demo, ist zu finden in: Workspace\demo\demo\Release\demo.mot. Das ständige Flashen zum Testen ist ziemlich umständlich. Schneller und einfacher geht das mit einem externen Emulator/Debugger, z.B. dem E8 von Renesas.