

# Mikrocontroller R8C/13 und Renesas HEW mit Emulator E8

Ein Softwareprojekt, bei dem der Emulator E8 von Renesas zum Einsatz kommt, wird eingerichtet und in Gang gebracht.

## Installation des High Performance Embedded Workspace (HEW)

Der HEW ist eine integrierte Entwicklungsumgebung unter Windows für Assembler- und C-Programme, die auf Renesas Mikrocontrollern, wie z.B. dem R8C/13, laufen. Bei der Installation des HEW entsteht ein umfangreicher Installationsprozess, der sowohl von CD als auch durch Download von Renesas möglich ist. Ein Update nach der Installation bringt den HEW auf den neuesten Stand, z.B. durch Versions Update.

Drei Komponenten treten hervor, obwohl das System aus viel mehr Komponenten besteht:

- HEW, also die integrierte Entwicklungsumgebung.

- FDT das Flash Development Toolkit), ein eigenständiges Programm zum Programmieren (Flashen) von Renesas Microcontrollern .

- AutoUpdate

Es genügt, bei den zu installierenden Mikrocontroller-Serien nur M16C/60,30,Tiny, 21,10 und R8C/Tiny für die Installation auszuwählen, RSK ... nur, falls man das entsprechende Renesas Starter Kit verwendet.

Am Ende des Installationsvorgangs will AutoUpdate gleich mit dem Update beginnen. Das sollte man zulassen, wenngleich dieser Vorgang ziemlich langwierig sein kann. Er setzt sich zusammen aus: Zusammenstellung der notwendigen Updates sowie deren Downloads und Installation. Außerdem gibt es Informationen und Fragen zu jedem einzelnen Update.

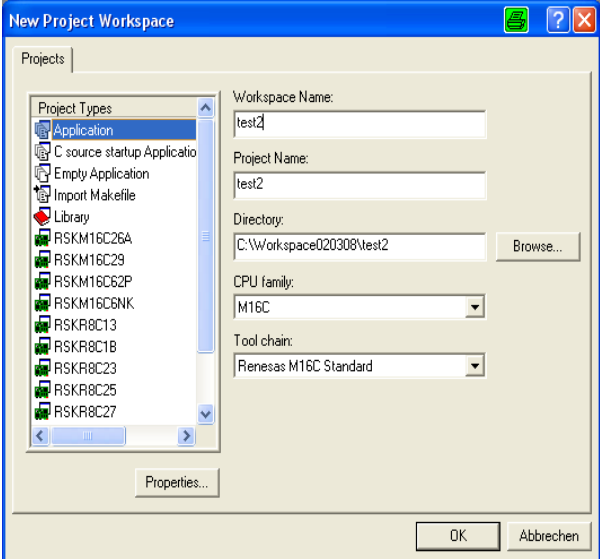
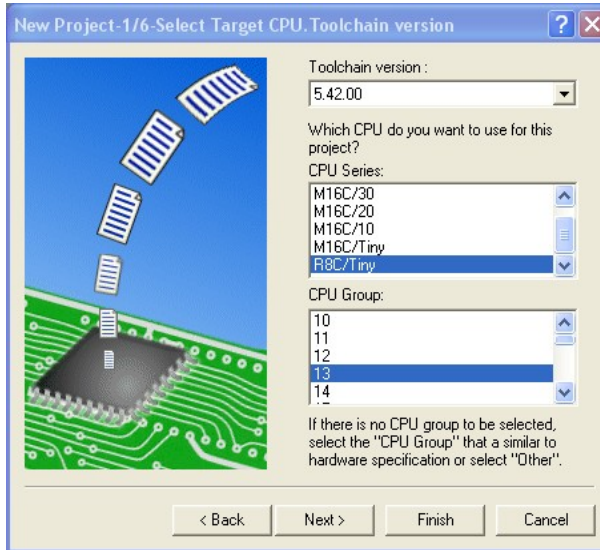
## E8 und Treiber

Diese Beschreibung setzt die Verwendung des Renesas Emulators E8 voraus, ein Hardwareteil, das den R8C auf dem bekannten 32 Pin Carrier über USB mit dem PC verbindet. Der E8 ist bei den Renesas Starter Kits enthalten.

Wurde vor der Installation ein älterer oder fehlerhafter HEW deinstalliert, sind die E8-USB-Treiber davon nicht betroffen. Es gibt sie noch. Im andern Fall meldet sich der Hardware-Assistent von Windows, wenn man den E8 mit dem R8C verbindet und dann den USB-Stecker einsteckt. HEW darf nicht gestartet worden sein. Man folgt den Empfehlungen bis man den Wizzard mit Finish beenden kann.

## Erzeugung eines neuen Projekts mit E8

Neue Projekte entstehen im Ordner, der unter Setup/Options/Workspace im HEW angegeben werden kann. Der Default-Ordner ist [C:\Workspace](#). Man muss allerdings beachten, dass im HEW *Projekt* und *Workspace* das gleiche bedeuten, zumindest ist kein deutlicher Unterschied erkennbar. Ein Projekt wird repräsentiert durch die Datei \*.hws. Also wird zum Öffnen eines Projektes (Workspace) seine hws- Datei geöffnet.

<p><b>Projekt, Workspace, Family</b></p> <p>Im ersten Fenster bei der Eröffnung eines neuen Projekts ist nur der Workspace Name in der ersten Zeile einzutragen, z.B. test2. Das Projekt erhält automatisch den gleichen Namen.</p> <p>Man kann mit Browse ein anderes Directory festlegen als das eingetragene Default-Directory.</p> <p>Der R8C gehört zur M16C-CPU-Familie.</p> <p>Ein Project-Type ist zu wählen, normalerweise Application. RSK... wird gewählt, wenn man mit einem Renesas Starter Kit arbeiten möchte. Die RSK-Projekttypen erscheinen nur, wenn sie installiert wurden.</p>	
<p><b>Toolchain, Target-CPU</b></p> <p>Die neueste Version der Toolchain ist schon eingetragen.</p> <p>Bei den Mitgliedern der M16C-Familie ist R8C/Tiny zu wählen.</p> <p>Der verwendete R8C gehört zur Gruppe 13.</p>	

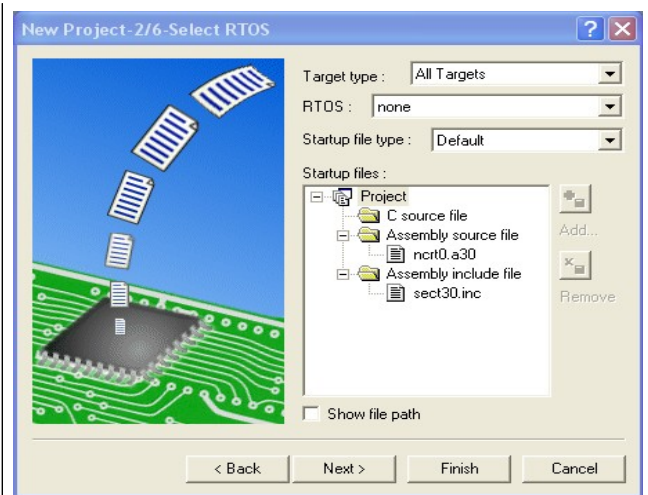
## RTOS

Etwas anderes als none kann nicht angegeben werden und Target type ist dann wohl ohne Bedeutung.

## Startup Files

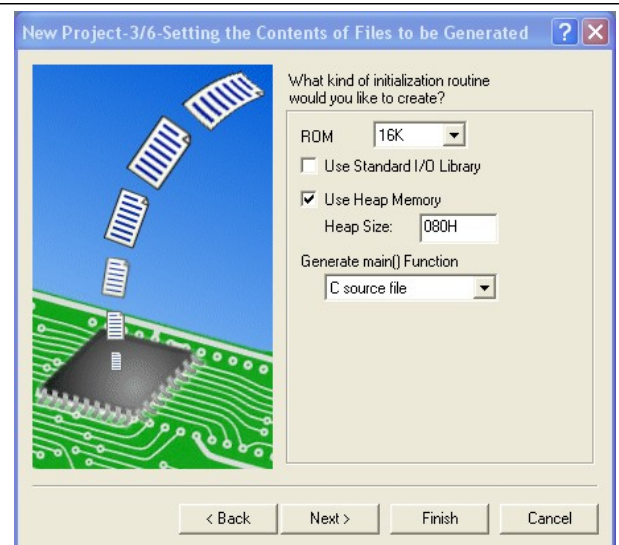
Die Startup Files, die mit Default dem Projekt zugefügt werden, sind genau die, die gebraucht werden (siehe auch unter Dateien) Mit USER können könnten auch andere ausgesucht werden.

In der neuesten Version des HEW werden auch die Dateien `sfr_r813.h` und `sfr_r813.inc` zugefügt. Später im Explorer erscheinen sie aber erst, wenn sie benutzt werden.



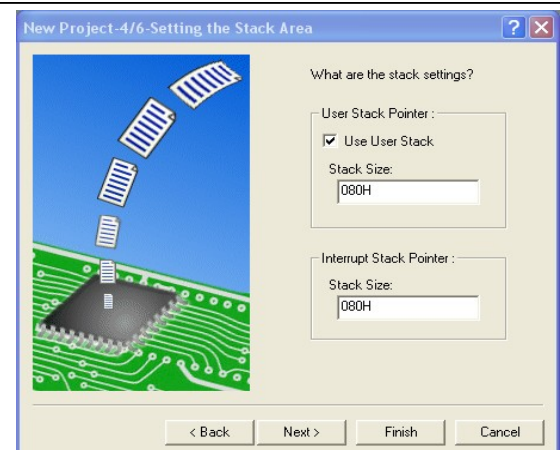
## Bau der Startup Files

Die Eintragungen müssen nicht geändert werden. Wenn man das mit Generate `main()` Function erzeugte Hauptprogramm nicht weiter entwickeln möchte, kann man es später gegen ein eigenes austauschen.



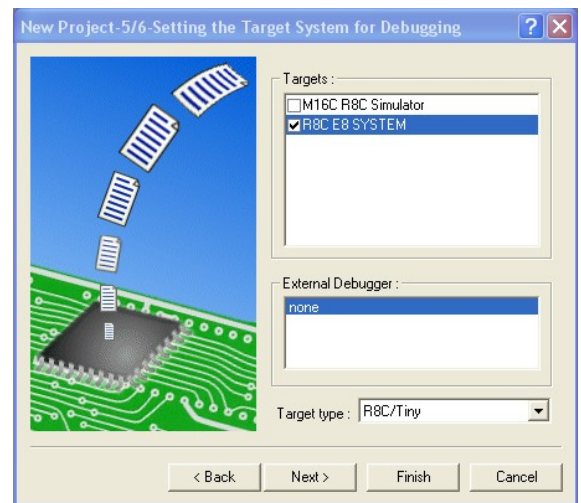
## Stacks

Die Eintragungen können wie vorgeschlagen übernommen werden.



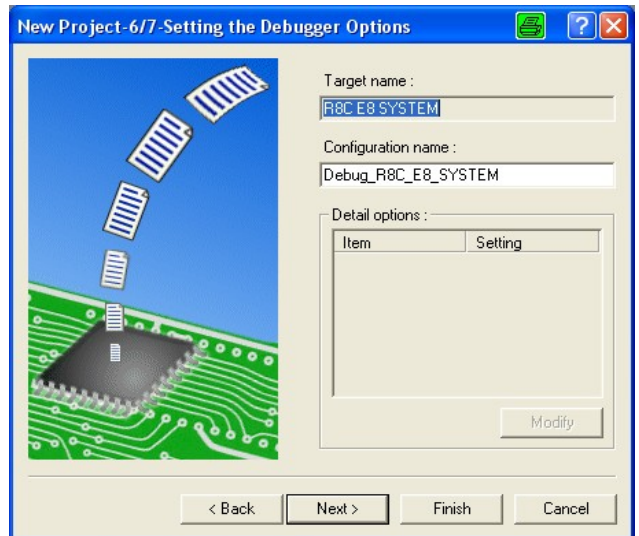
### Debugging

Das Häkchen auf das R8C E8 System zu setzen, weil diese Beschreibung auf dem E8 mit dem tatsächlich angeschlossenen R8C basiert.



### Debugger Option

Man könnte die Bezeichnungen ändern. In der neuesten Version des HEW gibt es noch die Option Core, die beim R8C bedeutungslos ist.

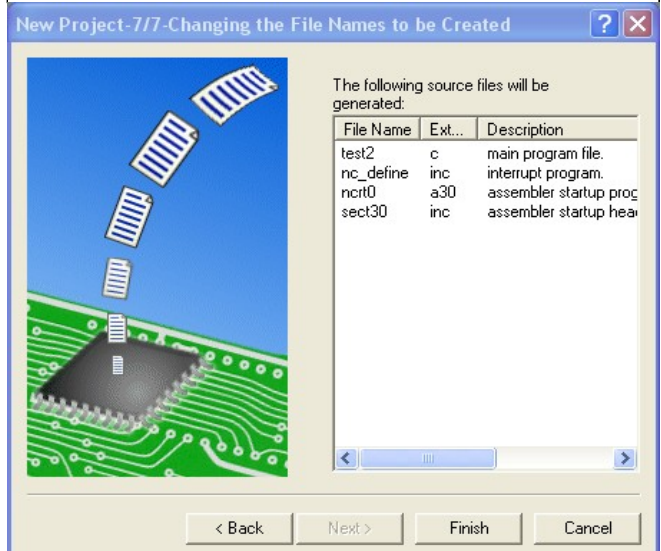


Im letzten Fenster sieht man die Liste der Dateien, die für das Projekt generiert werden. Man könnte die Namen ändern.

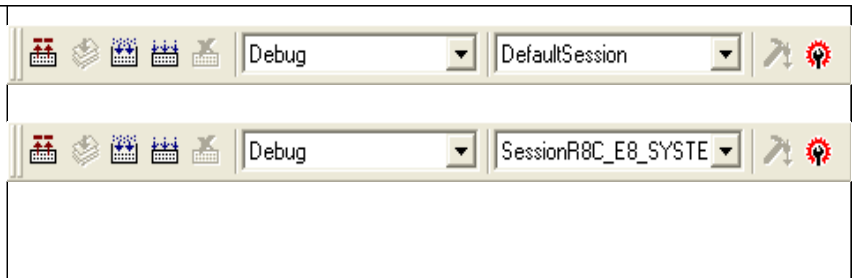
In der neuesten Version des HEW sind auch die Dateien sfr\_r813.h und sfr\_r813.inc in der Liste zu finden, wobei die letztere aber keine Bedeutung zu haben scheint.

Da es keinen Next-Button gibt, ist der Finish-Button zu drücken.

Danach sieht man ein Summary.



Zur Projekterzeugung gehört auch, die R8C\_E8 Session zu wählen. Dieser Eintrag ist zu speichern. Er erscheint dann später automatisch und muss beibehalten werden.

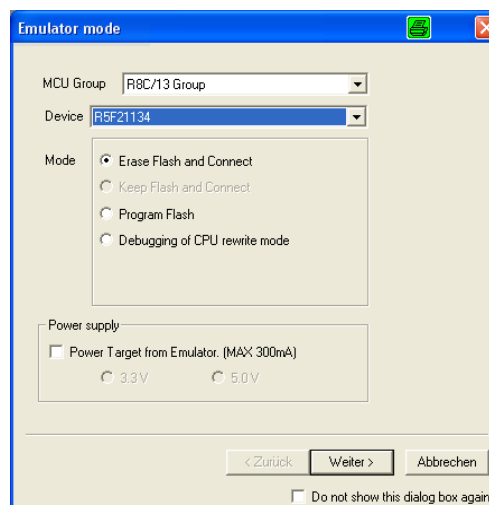


## Projekttyp und Verbindung

Die folgenden Fenster und Einstellungen sind nicht nur nach der Erzeugung eines neuen Projektes von Bedeutung, sondern auch nach dem Öffnen eines vorhandenen, beim Wechsel zwischen den Projekttypen Debug und Release und auch beim Trennen (disconnect) und Wiederherstellen (connect) der Verbindung zum R8C. Deutlich wird das daran, dass in allen Fällen die unten stehende Fensterfolge erscheint, mit der die Verbindung zum E8 eingerichtet oder geändert wird.

Es ist zweckmäßig mit dem Projekttyp Debug zu entwickeln, selbst dann, wenn man nicht debuggen will. Bei der Beschreibung des Projekttyps Release wird das deutlich (siehe unten).

Nach der Erzeugung eines neuen Projektes sind die meisten Angaben in den Fenstern zu machen. Beim Öffnen eines vorhandenen Projektes werden die gespeicherten Angaben nur zur Änderung angeboten.

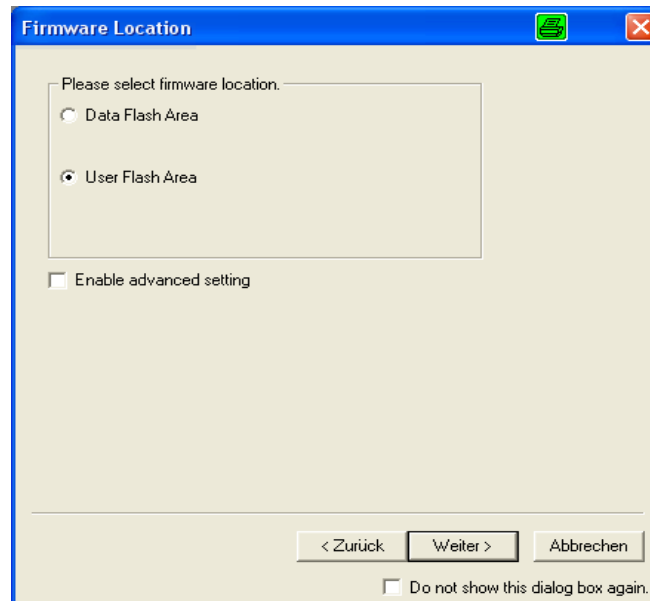


Den R8C/13 gibt es mit unterschiedlichen Speichergrößen. Daher ist unter Device die genaue Bezeichnung R5F21134 erforderlich.

Da mit dem Projekttyp Debug entwickelt wird, ist Erase Flash and Connect einzustellen. Falls die Schaltung über den E8 mit Strom versorgt werden soll, ist Power Target from Emulator mit Häkchen zu versehen und die Spannung anzugeben.

Beim Projekttyp Debug wird zusätzlich zum Anwenderprogramm eine Firmware in den R8C geschrieben. Deshalb muss im nächsten Fenster die location dafür angegeben werden. Der Flash Speicher für das Anwenderprogramm, die User Flash Area, beginnt bei Adresse C000H. Wenn die Firmware dort gespeichert werden soll, muss die Startadresse für Anwenderprogramme, die ROM TOP ADDRESS, in der Datei nc\_define.inc auf C800H gesetzt werden. Weil das später wieder zurück geändert werden muss, sollte, wenn möglich, die Data Flash Area benutzt werden, die in Block A und

Block B aufgeteilt ist. Dort können Anwenderprogramme Daten unverlierbar speichern. Wenn nicht beide Blöcke genutzt werden, ist diese location günstiger. Der Block, der die Firmware aufnehmen soll, kann unter Enable Advanced Settings gewählt werden.



Das dritte Fenster ist hier nicht dargestellt, weil die Baudrate darin nicht geändert werden muss. Mit Button Fertig wird die Verbindung hergestellt. Je nach Vorgeschichte kann es sein, dass der E8 dann seine Software updatet.

Die drei Fenster können auch als Registerkarten gemeinsam erscheinen.

Wichtig ist die untere Symbolleiste, die sich nach dem Verbindungsaufbau öffnet. Mit dem Button Disconnect kann man die Verbindung trennen und mit Connect wieder herstellen. Und es gibt eine ganze Reihe von Buttons zur Steuerung des Programmlaufs bis hin zu Stop.



Der Button Reset Go (Programmstart) ist freigegeben. Wenn man ihn betätigt, kommt die Warnung, dass kein Download zum R8C stattgefunden hätte, selbst wenn der R8C aus dem vorherigen Durchgang ein lauffähiges Programm enthält. Der Vorgang sollte auf jeden Fall abgebrochen werden. Korrekt ist es, fehlerfrei zu compilieren (Build all), den Download zu bestätigen und, wenn erfolgt, das Programm zu starten.

## Entwicklung

Schon bei der Einrichtung eines Projektes erscheint eine Liste mit den dem Projekt zugefügten Dateien, darunter ncr0.a30. (In der neuesten Version des HEW ist ein früherer Fehler in dieser Datei beseitigt.)

---

In ncr0.a30 ist Assembler Source Text eines Startup Programms enthalten. Darin eingebunden sind die Dateien nc\_define.inc und sect30.inc. In sect30.inc ist ggf. eine Änderung vorzunehmen, und zwar in der Interrupt-Vector-Tabelle, natürlich nur, wenn Interrupts benutzt werden.

Zum Beispiel ist für den Timer Y der Dummy-Eintrag von vector 23 zu ersetzen (im folgenden Beispiel heißt der Interrupthandler timer\_y) und die Tabelle ist mit if 1 generell freizugeben:

```
.if 1
.
.
.glb    _timer_y
.lword  _timer_y           ; vector 23      timer y
.
```

Welche Vektoren zu welchen Interrupts gehören, kann man in den Datenblättern des R8C finden.

Die Datei sfr\_r813.h ist eine C-Header-Datei, die nützliche Definitionen im Zusammenhang mit den Registern des R8C/13 (Special Function Register) enthält. Wenn man sie einbindet, muss man sich z.B. nicht mehr um Adressen von Registern kümmern. Allerdings erscheint die Datei sfr\_r813.h nach der Erzeugung eines neuen Projektes erst in der Explorer Leiste, wenn compiliert wurde (Build all), vorausgesetzt, sie wurde eingebunden. Den üblichen Schutz gegen mehrfaches Includieren hat sie allerdings nicht, den muss man zufügen.

```
#ifndef SFR_R813_H
#define SFR_R813_H
.
.
#endif
```

Eigene C-Sourcen und Header Dateien, die man im Projekt verwenden möchte, werden am besten mit dem Windows Explorer in das Projekt kopiert. Zum Projekt gehören diese Dateien aber erst, wenn sie mit Project/Add Files dem Projekt zugefügt werden.

Die C-Sourcen müssen nicht unbedingt im Projektverzeichnis stehen, wohl aber die Header Dateien, da sie meistens projektspezifisch angepasst werden.

Das vom Projekt generierte Hauptprogramm wird lediglich gegen das eigene ausgetauscht. Das folgende minimale Programmbeispiel lässt eine LED blinken, die an Bit 5 von Port 4 (Pin 16) angeschlossen ist. Für jede Hell- und Dunkelphase muss die while-Schleife 100 000mal durchlaufen werden. Zur Vorbereitung müssen der externe Taktgeber und der LED-Ausgang konfiguriert werden.

```
#include "sfr_r813.h"

long n;

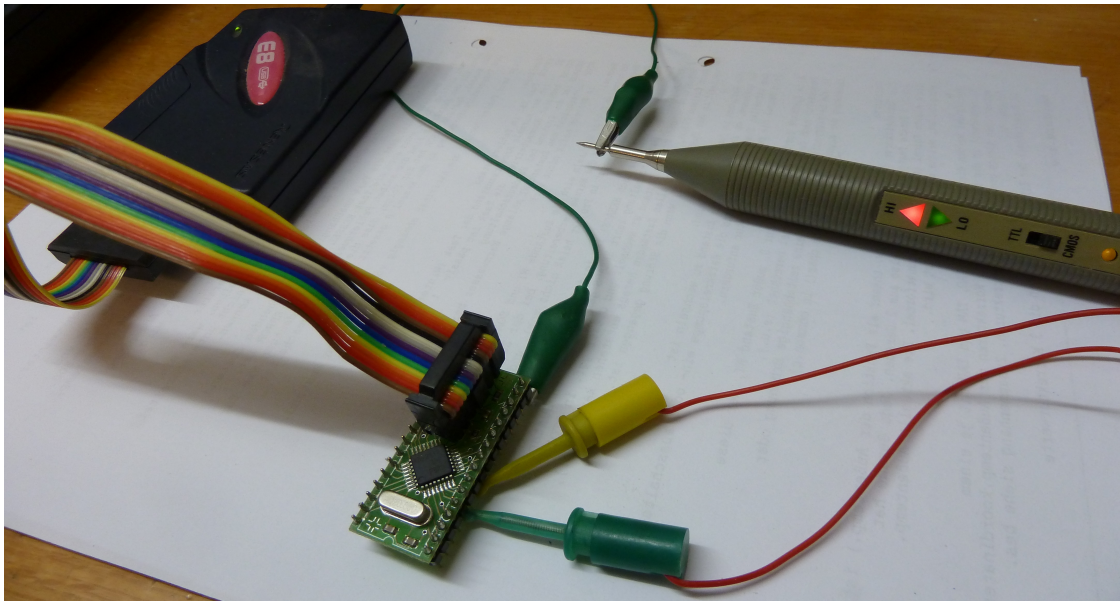
void main(void) {
    asm( "FCLR I");    // Change on-chip oscillator clock to Main clock
    prc0 = 1;         // Protect off
    cm13 = 1;         // Xin Xout
    cm15 = 1;         // XCIN-XCOUT drive capacity select bit : HIGH
    cm05 = 0;         // Xin on
    cm16 = 0;         // Main clock = No division mode
    cm17 = 0;
    cm06 = 0;         // CM16 and CM17 enable
    asm("nop"); asm("nop"); asm("nop"); asm("nop");
    ocd2 = 0;         // Main clock change
    prc0 = 0;         // Protect on

    pd4 = pd4 | 0x20; // LED drive
    drr6 = 1;         //drive capacity high
    drr7 = 1;

    while (1) {
```

```
for (n= 0; n < 100000; n++);  
p4 = p4 ^ 0x20;  
}  
}
```

Der ganz normale Entwicklungszyklus ohne Debuggen ist folgendermaßen: Editieren, Button Build all (Compile), Download zum R8C, Button Reset/Go, Button Stop, Editieren, Button Build all (Compile) usw. Die Stärke des E8 liegt natürlich in den Debug-Funktionen, aber die sollen hier nicht beschrieben werden.



Das Bild zeigt den E8, der auf den R8C aufgesteckt ist und der den Strom liefert. An Pin 16 ist ein Logiktester angeschlossen, der die Impulse, die das oben stehende Programm liefert, sichtbar macht. Pins 5 und 7 versorgen den Logiktester mit Strom.

Wenn ein Programm tut, was es soll, ist die Entwicklung vorerst beendet. Jetzt muss der R8C in der endgültigen Art und Weise programmiert werden. Dazu wird der Projekttyp Release eingestellt.



Der Wechsel zum Projekttyp Release bringt die Fenster zur Verbindungseinstellung auf den Bildschirm (siehe oben). Es ist der Mode Program Flash einzustellen. Die Verbindung zum E8 wird erneuert.

Im wesentlichen bedeutet Release, dass das Anwenderprogramm ohne die Debug Firmware auf den R8C übertragen wird. Das Fenster für den Download erscheint aber erst nach Build all.

Der E8-Stecker ist zu ziehen, damit das Programm starten kann. Wenn das System über den E8 mit Strom versorgt wurde, ist für eine externe Stromversorgung zu sorgen. Weil das Ziehen des Steckers zwecks Programmstart im Projekttyp Debug nicht notwendig ist, ist dieser Projekttyp zum Entwickeln ohne Debuggen vorzuziehen.

Bevor das Projekt geschlossen wird, sollte es wieder auf den Projekttyp Debug zurückgesetzt werden, damit dieser Typ beim nächsten Öffnen erscheint, denn in aller Regel geht dann die Entwicklung weiter. Dass von Release auf Debug gewechselt werden muss, wird dann leicht übersehen. Beim Wechsel auf

---

Debug erscheint das Verbindungsfenster, in dem der Mode Erase Flash and connect einzustellen ist. Auch wenn der R8C mit eigenem Strom betrieben wird und arbeitet, kann der E8-Stecker aufgesteckt werden, wodurch das Programm stoppt. Sollte der R8C ohne Strom sein und der E8 Strom liefern, startet es.

## HEW-FDT

Um unabhängig vom HEW zu flashen, gibt es das Programm Flash Development Toolkit (FDT) als eigenständiges Programm, worauf hier aber nicht eingegangen wird. FDT ist aber auch im HEW mit einer Symbolleiste eingebunden.

Man braucht FDT eigentlich nur für Sonderfälle, denn das Flashen des Programms mit dem HEW über den E8 ist völlig ausreichend.

Die Buttons der Symbolleiste werden aktiv, wenn man mit Button Configure Flash Project die Device eingestellt hat und das Gerät, also den E8.

Mit FDT Connect stellt man die Verbindung her, wonach auf Nachfrage eine andere Firmware in den E8 geladen wird.

Man kann genauso wie beim Projekttyp Release ein Anwenderprogramm mit Hilfe seiner mot-Datei auf den R8C übertragen, worauf hier aber nicht eingegangen wird.

HEW-FDT ist sicher gut zu gebrauchen, wenn mehrere Bausteine identisch programmiert werden sollen oder wenn ein Programm von einem Baustein auf einen anderen zu kopieren ist.