Franzis- Lernpaket Mikrocontroller und Attiny 45

Ralf Beesner

21.1.2011

1 Einleitung

Ich habe kein sonderliches Programmiertalent, aber da ich inzwischen ein Grundgerüst an Programmen habe, aus denen ich Teile wiederverwerte, wird mein Spaghetti- Code immer länger und passt häufig nicht mehr in einen ATtiny 13.

Da ich die Hardware des „Lernpakets Mikrocontroller“ auch mit ATtiny 45 weiternutzen möchte, habe ich mich mit dem Kommandozeilenprogramm `avrdude` beschäftigt und ein paar kleine Batchdateien beigefügt, die den Umgang mit `avrdude` etwas vereinfachen.

Getestet ist es allerdings nur mit Windows XP (neuere Versionen habe ich nicht).

2 Grundsätzliches

`avrdude` stammt aus der Linux- Welt. Dort sind Kommandozeilen- Programme aus zwei Gründen beliebt und verbreitet:

- sie lassen sich aus Skripten (Batchdateien) heraus aufrufen; mit Skripten lassen sich komplexe Aufgaben automatisieren
- Linux ist modular; die eigentliche Programmfunktionalität steckt häufig in Kommandozeilentools; für die leichtere Bedienbarkeit sorgt eine separate grafische Bedienoberfläche (sofern jemand Lust hatte, sie zu programmieren) ;-)

`avrdude` ist ein GPL- Programm und damit „freie“ Software, der Quellcode liegt offen und steht jedem zur Verfügung, der das Programm weiterentwickeln möchte,

allerdings darf er das Programm nicht ohne Zugang zum geänderten Quellcode und ohne die Datei `COPYING` weiterverbreiten.

Da es auf der Homepage keine kompilierte Windows- Version gibt, habe ich sie beigefügt.

Die große Stärke von `avrdude` ist, dass man in der Konfigurationsdatei `avrdude.conf` eigene Programmer definieren kann.

3 Vorbereitung

Eine weitverbreiteter Programmierer für die serielle Schnittstelle ist Ponyprog (siehe http://www.lancos.com/prog.html). Die Ponyprog- Hardware tut eigentlich dasselbe wie Burkhard Kainkas LP- Mikrocontroller- Hardware, verwendet allerdings ein paar mehr „Angstbauteile“, die den Chip schützen sollen - und sie hat eine leicht abweichende Schnittstellen-Belegung. In der Datei `avrdude.conf` findet man PonyProg unter dem Eintrag „ponyser“:

```
programmer id = "ponyser";
desc = "design ponyprog serial, reset=!txd sck=rts mosi=dtr miso=cts";
type = serbb;
reset = ~3;
sck = 7;
mosi = 4;
miso = 8;
;
```

Ich habe einen neuen Programmer hinzugefügt und ihn „burkhard“ genannt:

```
programmer id = "burkhard";
desc = "Franzis LP Microcontroller";
type = serbb;
reset = ~7;
sck = 3;
mosi = 4;
miso = 8; ;
```

Die Zeile `reset = ~7` bewirkt, daß die RTS- Leitung auf High geschaltet wird und so den Programmer mit Strom versorgt; den eigentlichen Reset muß man wie bei Burkhard Kainkas `LPmikoISP.exe` mit einer kleinen Drahtbrücke anlegen.

Ein zweiter hinzugefügter Programmer „burkhard2“ ist der “Mega8- ISP-Programmer“, den Burkhard Kainka für das Franzis- Pingpong entwickelt hat. Er

benutzt die RTS- Leitung der seriellen Schnittstelle als Reset- Leitung (der Mikrocontroller benötigt daher externe Betriebsspannung). Es ist nur der Eintrag für die Reset- Leitung geändert (das Signal darf nicht invertiert werden): `reset = 7`

Man sollte nun noch `avrdude.exe` und `avrdude.conf` in einen Ordner kopieren, der im Suchpfad des Betriebssystems liegt, dann muß man sie nicht immer im aktuellen Arbeitsordner vorhalten. Wenn man nicht in der Systemsteuerung herumfummeln möchte, kopiert man sie am einfachsten in den Ordner `C:\Windows` . Der liegt standardmäßig im Suchpfad und ist eh so eine Art „Rumpelkammer“, in der Logfiles und irgendwelche Installationsreste landen.

Da Windows direkte Hardwarezugriffe unterbindet, muss man noch einen kleinen Treiber namens `giveio.sys` installieren; dies erfolgt durch Aufruf der Datei `install_giveio.bat` . Der Treiber ist Bestandteil des Avrdude- Pakets.

4 Ein erster Kontakt

Gibt man am Kommandozeilenprompt `avrdude` ein, wird man von zahlreichen Parametern „erschlagen“:

Die meisten bleiben jedoch gleich (z.B. Mikrocontroller-Typ, Programmer und Schnittstelle), so daß man sich nur ein mal damit zu beschäftigen braucht und sie in einer Batchdatei zusammenfassen kann.

Eine kleine Batchdatei `flash.bat` , mit der man ATiny45 flashen kann, enthält nur eine Zeile:

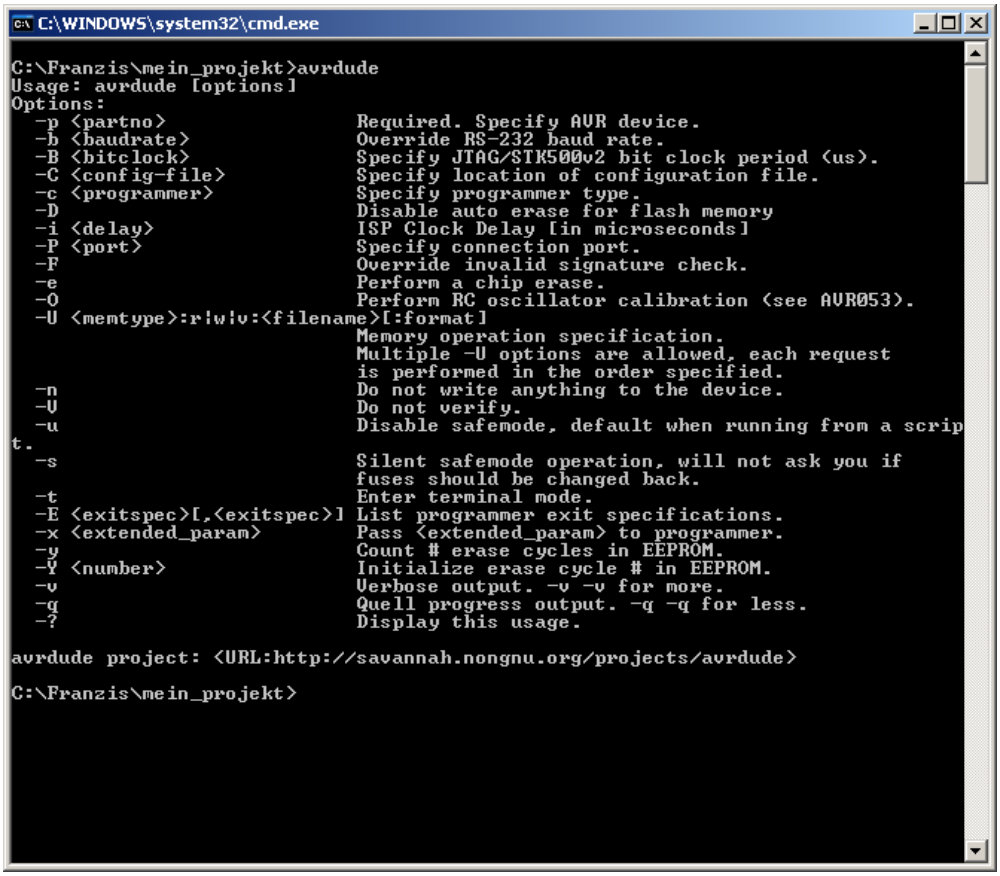
```
avrdude -p t45 -c burkhard -P com1 -U flash:w:%1.hex
```

Der Parameter `%1` ist ein Platzhalter für den Dateinamen (ohne Dateiname- Endung; die wird durch die Batchdatei angefügt).

Ein Anwendungsbeispiel:

Wir haben unter `C:\Franzis\Mein_Projekt` eine Datei `programm1.hex` erzeugt.

Wir starten ein „DOS-Fenster“, wechseln mit `cd C:\Franzis\Mein_Projekt"` in den Arbeitsordner und geben dort `flash programm1` ein. Betätigen wir im „DOS-Fenster“ die (Tastatur-) Cursor- Tasten für „rauf“ und „runter“, können wir in den bereits abgesandten Befehlen scrollen und einen älteren Befehl durch einfaches Drücken der Eingabetaste wiederholen. Das geht weitaus schneller, als sich durch den „Datei- Öffnen“- Dialog einer grafischen Oberfläche zu clicken.



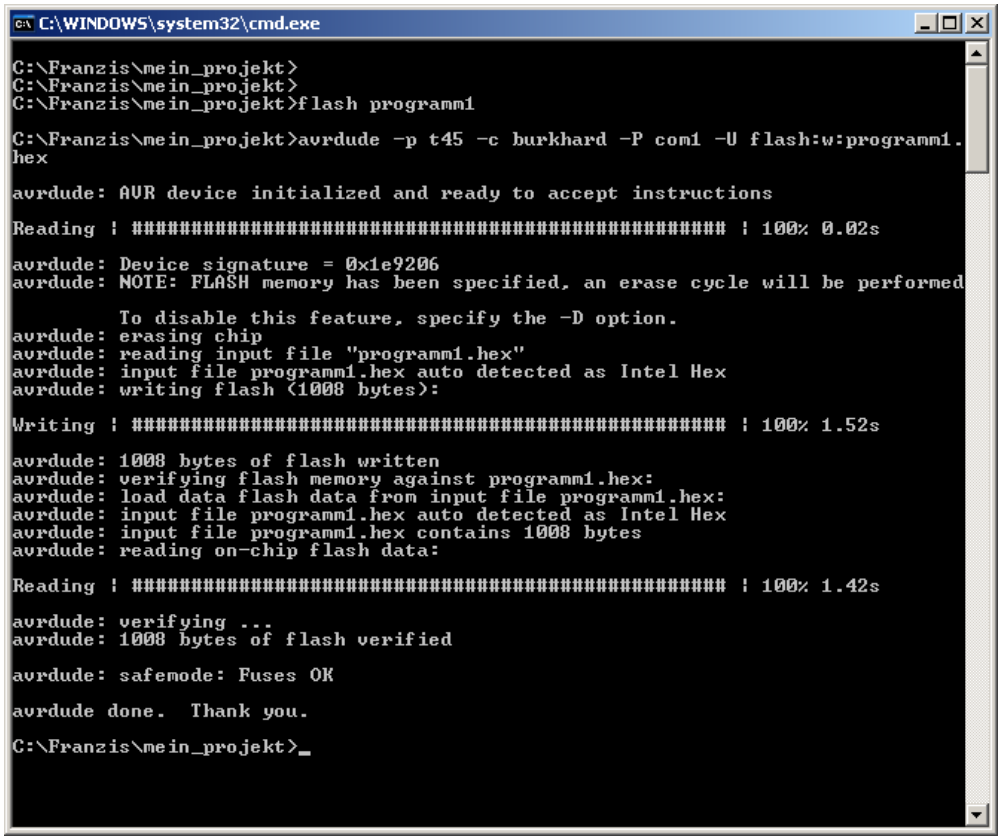
5 Fusebytes ändern

Das Hochladen von Programmen ist sicherlich die häufigste Aufgabe für ein Programmiertool; die zweitwichtigste ist wohl das Umprogrammieren von Fusebits. Meist möchte man die Taktquelle auf einen externen Quarz ändern, den 1/8-Vorteiler aus- oder einschalten oder den Brownout- Detektor aktivieren.

Dies ist der Aufruf von `avrdude` um das Lowfuse- Byte für die Standard- Taktrate (RC- Oszillator 8 MHz; Vorteilung durch 8) zu setzen:

```
avrdude -p t45 -c burkhard -P com1 -U lfuse:w:0x62:m
```

Der Teilstring `0x62` ist das Fusebyte in HEX- Darstellung; das `m` weist avrdude an, diesen Wert als Direkteingabe zu interpretieren und nicht als Filenamen.



6 avrdude- Versionen

Die ursprünglich verwendete avrdude- Version 5.6 hat (wohl nur unter Windows) eine ärgerliche Macke: der Parameter -i , mit dem man die Kommunikation mit dem AVR verlangsamen kann, funktioniert nicht.

Daher gab es Probleme, wenn der Attiny auf niedrige Taktraten geflasht wurde (16 kHz oder 32 kHz). Er war dann anschließend nicht mehr ansprechbar, weil die ISP-Taktrate höher als ein Viertel der AVR- Taktrate war.

Mit der beigefügten Version 5.10 funktioniert der Parameter -i , und mit `-i 1000` lassen sich auch AVRs „wieder einfangen“, die auf 4 KHz (32 kHz- Uhrenquarz plus Vorteiler 1/8) geflasht wurden.

7 Einige vorbereitete Batchdateien

`flash45.bat` enthält neben dem „Einzeiler“ noch ein paar Verwendungsinformationen.

`fuses45.bat` enthält eine Reihe von vorbereiteten Einträgen für gängige Aufgaben. Sie sind alle mit `rem` auskommentiert, man muss nur noch vor dem gewünschten Befehl das `rem` entfernen.

`eeprom45.bat` ermöglicht das flashen des AVR- EEPROM- Bereichs, sofern der Compiler eine *.eep- Datei „ausgespuckt“ hat.

Wenn man in diesen Dateien `t45` durch `t25` bzw. `t85` ersetzt, lassen sich auch ATtiny 25 und ATtiny 85 flashen.

ATtiny 13 haben eine abweichende Belegung der Fusebytes. Daher ist eine Batchdatei für Attiny 13 beigefügt. Mit `fuses13.bat` kann man auch die Brownout-Bits konfigurieren, was mit `LPmikroISP.exe` nicht möglich ist. Aus dem gleichen Grunde ist auch eine Batchdatei zum flashen des ATtiny 13- EEPROMs beigefügt.