



Carsten Meyer

Betriebs-Wirth

Embedded-Programmierung für sparsame Feingeister

Selbst entwickelte Mikrocontroller-Programme müssen weder langsam (Basic), kryptisch (C) noch umständlich (Assembler) sein: Pascal erfüllt höchste Anforderungen an Lesbarkeit und Zuverlässigkeit, vor allem, wenn es wie im vorliegenden Fall sinnvoll durch Modula-2-Elemente und hardwarenahe Befehle ergänzt wurde.

Für die Freunde hardwarenaher Programmierung hat die Firma Embedt ein attraktives Päckchen geschnürt: Das Bundle aus Experimentier-Board für die weit verbreiteten Atmel-AVR-Prozessoren, In-Circuit-Programmer zum Anschluss an den PC-Printer-Port und dem AVRco-Pascal-Compiler von e-lab erhält man für etwas über 100 Euro. Wer noch Kenntnisse aus Turbo-Pascal-Zeiten mitbringt, hat in zehn Minuten sein erstes „Hello World“ auf einem angeschlossenen LC-Display stehen.

Auf dem Experimentier-Board sind nicht nur die Portleitungen

des winzigen ATmega8-Prozessors auf Pfostenleisten herausgeführt, es enthält auch schon einige (vom Compiler voll unterstützte) Peripherie: I2C-Bausteine als Porterweiterung mit vier LEDs und Tastern, einen Schrittmotor-Treiber, die serielle RS232-Schnittstelle und einen Anschluss für handelsübliche Punktmatrix-LCDs mit ein bis vier Zeilen und 8 bis 40 Spalten. Der 16 MHz schnelle Atmel-RISC-Prozessor selbst besitzt 25 I/O-Ports, von denen einige mit A/D-Eingängen, UART, PWM-Generator/Timer und TWI (I2C-Bus) doppelbelegt sind. Für Programme und Daten stehen

2 KByte RAM, 512 Byte EEPROM und 8 KByte Flash-Speicher zur Verfügung. Da die meisten Befehle nur einen Taktzyklus benötigen, ist der Prozessor mehr als 40-mal so schnell wie der Klassiker Z80; trotzdem kommt er mit wenigen Milliampere Strom aus.

Postmaterielles Basteln

Die auch einzeln für 25 Euro erhältliche AVRco-Version (Windows 2000/XP) ist genau auf diesen Prozessor zugeschnitten und die Programmgröße deshalb auf rund 6000 Prozessorbefehle begrenzt, entsprechend je nach Anzahl der benötigten Systemfunktionen etwa 600 bis 800 Zeilen Pascal. Das genügt durchaus schon für anspruchsvollere Aufgaben, vor allem, weil die sonst so lästige Peripherie-Ansprache durch sehr weitreichende Systemfunktionen übernommen wird und echte Multitasking-Fähigkeiten bereits ins System integriert sind. Man kann Treiber für verschiedenartige Displays bis hin zum Punktmatrix-LCD, Schrittmotor-Endstufen, I2C-Portbausteine, die integrierten A/D-Wandler, Beeper, DCF77-Signale, PWM-Modulatoren, Inkrementalgeber, RC5-Fernbedienungsempfänger oder Fernsteuer-Servos aus dem Modellbau-Bereich hinzuladen, die allesamt sehr kompakt und schnell als Assembleroutinen implementiert wurden. Mit dem integrierten Simulator lässt sich das Programm bis ins Detail austesten, die Simulation umfasst auch einige der aufgeführten Peripherien.

Den Spaß an der Pascal-Programmierung merkt man den Schöpfern des Compilers schnell an: Die integrierte Entwicklungsumgebung wurde in Delphi erstellt, dem Pascal-Enkel von Borland, auch die Dokumentation spart nicht an Seitenhieben in Richtung C-Gemeinde. Tatsächlich ist der Einsatz von C beispielsweise in militärischen und sicherheitskritischen Bereichen

verpönt – verdeckte Programmierfehler, wie sie sich in atemberaubende C-Konstrukte gern einmal einschleichen, hätten hier fatale Folgen.

Dabei bleiben die Programmierer durchaus selbstkritisch: Ihr Compiler erzeugt keinen hochgradig optimierten, sondern eher konservativen, unverstümmelten Code, bei dem sich an jeder Stelle die zugrunde liegenden Quellen nachvollziehen lassen. Das Einfügen schneller selbst gestrickter Assembleroutinen ist deshalb eine ausgesprochen leichte Übung, eben weil sich die Wege des Compilers nicht in abenteuerlichen Optimierungen verschlingen. Definiert man seine Variablen prozessorgerecht – einige der 32 AVR-Arbeitsregister stehen dafür zur freien Verfügung – ist der Unterschied zu einem optimierenden C-Compiler nur noch marginal, bei entscheidend besserer Les- und Wartbarkeit.

Für Einsteiger in die Controller-Programmierung ist das Paket eine der preiswertesten und einfachsten Möglichkeiten, die leistungsfähige AVR-Prozessorfamilie kennen zu lernen. Der Compiler ist zuverlässig, die Sprache trotz einiger dezenter Erweiterungen gegenüber Niklaus Wirths Standard-Pascal leicht erlernbar, die Entwicklungsumgebung erfordert kaum Einarbeitung. Als einziges Manko neben der Beschränkung auf nur einen Prozessortyp konnten wir lediglich die nicht ganz aktuelle Dokumentation ausmachen, einige Features fanden wir nur in den beigefügten Beispielprogrammen beschrieben. Weitergehende Ansprüche bezüglich anderer Controller-Typen können dann die Vollversionen des Compilers befriedigen. In der Luxus-Ausführung für rund 950 Euro erhält man neben unbegrenzter Code-Größe auch ein FAT16-Filesystem für Multimedia-Cards, einen Ethernet-Stack und einen JTAG-Hardware-Debugger für die Echtzeit-Fehlersuche. (cm)

ATmega8 Entwicklungssystem

Lieferumfang	ATmega8 Application Board, ATmega8 Mini-Prozessor-Modul, Steckernetzteil, ISP-Programmer für PC-Printerport, AVRco Pascal-scm Cross-Compiler/IDE
Hersteller	E-Lab, www.e-lab.de (Compiler) Embedt, www.embedt.de (Hardware)
Systemanf.	Windows 2000 oder XP, empfohlen 300 MHz-Pentium mit min. 256 MByte RAM
Preis	Set aus IDE/Compiler/Simulator, ISP-Programmer und ATmega8 Mini-Modul 63 €, Application Board 39 €

