

Aufbau und Funktion von Mikrocontrollern am Beispiel des ATtiny2313

6. Dezember 2005

Inhaltsverzeichnis

1 Grundlagen	1
1.1 Was ist ein Mikrocontroller?	1
1.2 Was ist ein Programm?	2
2 Aufbau eine Mikrocontrollers	2
2.1 Blockdiagramm des ATtiny2313	2
2.2 Die ALU	3
2.3 Die Register	3
2.4 Die Busse	3
2.5 Der Takt	4
2.6 Die Speicher	4
2.7 Die Peripherie	4
2.7.1 bei unser Frequenzmessung verwendete Peripherie	5
3 Programmierung eines Mikrocontrollers	6
3.1 Wie schreibe ich ein Programm?	6
3.1.1 Typischer Ablauf bei der Softwareentwicklung	6
3.1.2 Hilfsmittel für die Softwareentwicklung	6
3.2 Wie kommt das Programm in den Mikrocontroller?	7
4 Glossar	7

1 Grundlagen

1.1 Was ist ein Mikrocontroller?

Ein Mikrocontroller ist im wesentlichen ein, in der Regel nicht ganz so leistungsstarker Computer, der auf einem einzigen Chip untergebracht ist. Er besitzt neben einer CPU und einem Zentralspeicher diverse Peripheriebausteine für die verschiedensten Aufgaben (Kommunikation, Digital/Analog-Wandlung, Zeitmessung, etc.).

Heute werden Mikrocontroller wegen ihres geringen Preises und universellen Einsetzbarkeit in sehr vielen elektronischen Geräten verwendet (Handys, Taschenrechner, Armbanduhren, Waschmaschinen, Radios, Autos, ... etc.).

Im Folgenden wird vor Allem der in unserem Projekt zur Frequenzmessung eingesetzten ATtiny2313 behandelt.

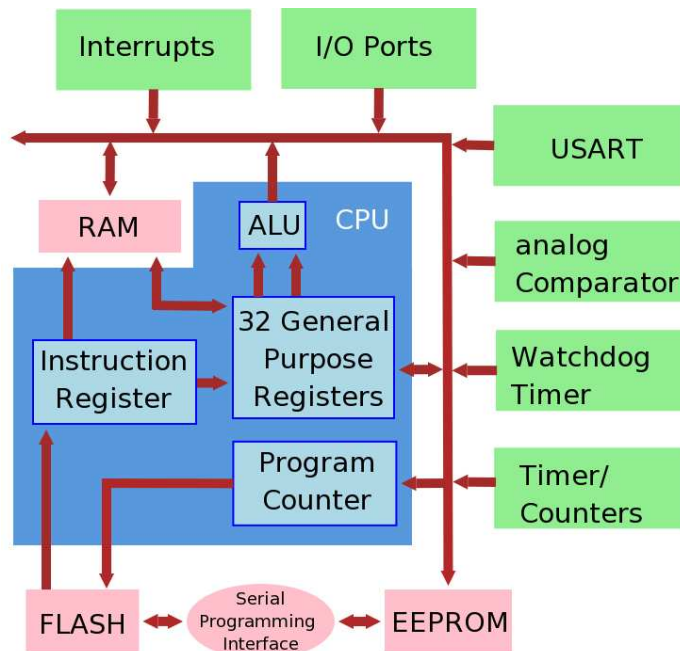
1.2 Was ist ein Programm?

Ein Programm ist eine Liste von Befehlen, die von einem Computer Schritt für Schritt abgearbeitet werden können. Ein Programm kann auf viele Arten und Weisen dargestellt werden, z.B.:

- als Quellcode in einer höheren Programmiersprache wie C oder Java
- als Assemblercode
- als binärer Maschinencode, d.h. lauter nullen und einsens. Ein Programm kann nur in dieser Form von einem Computer ausgeführt werden
- als Ablaufdiagramm, z.B. PAP, Nassy-Schneidermann Stuktogramme

2 Aufbau eine Mikrocontrollers

2.1 Blockdiagramm des ATtiny2313



2.2 Die ALU

Die ALU (engl. **A**rithmetic **L**ogic **U**nit) ist das Kernstück eines jeden Mikrocontrollers. Sie führt die einzelnen Befehle aus. Insgesamt unterscheidet man die folgenden Befehlsarten:

- Transportbefehle (z.B. lesen aus und schreiben in das RAM)
- arithmetische, logische und Shiftbefehle (z.B. Addieren, bitweise UND, Linksshift, etc.)
- Sprungbefehle (Sprung an eine bestimmte Stelle im Programm)
- Unterprogrammbefehle
- Trap-Befehle (Auslösen eines Interrupts)

2.3 Die Register

Ein Register ist ein Stück Speicher. Im Gegensatz zu anderen Speichern wie, z.B. dem RAM kommen Registern in der Regel besondere Aufgaben zu. Es gibt insgesamt drei Arten von Registern:

1. Datenregister

in ihnen stehen Daten die von einer Funktionseinheit des Mikrocontrollers als Eingabe benötigt werden (z.B. die Operanden für eine Addition, ein Byte das über die serielle Schnittstelle gesendet werden soll) oder in ihnen stehen Daten, die als Ergebnis einer Operation entsandt sind (z.B. Ergebnis einer Addition, ein Byte was über die serielle Schnittstelle empfangen wurde.

2. Steuerregister

durch setzen und löschen ihrer Bits (auch Flags genannt) lassen sich wie mit Schaltern Einstellungen an den einzelnen Funktionseinheiten des Mikrocontrollers vornehmen

3. Statusregister

ihre Bits geben wie Kontrolllämpchen Auskunft über den Zustand der einzelnen Funktionseinheiten des Mikrocontrollers

2.4 Die Busse

Die Busse werden für den Datentransport von einer Funktionseinheit zur anderen benötigt. Meist hat ein Mikrocontroller nur zwei Busse einen für Daten und einen für Adressen. Der Adressbus des ATtiny2313 ist 16Bit und sein Datenbus 8Bit breit.

2.5 Der Takt

Der Takt ist meist ein Rechtecksignal, das dafür sorgt, dass im Mikrocontroller alles einen geordneten Gang geht. Der Mikrocontroller wechselt mit jedem Taktschritt (z.B. jeder steigenden Flanke) von einem elektrisch stabilen Zustand in einen anderen. Dabei wird stets ein Schritt in der Programmausführung gemacht. Das kann, muss aber nicht immer der Ausführung eines Befehls durch die ALU entsprechen.

2.6 Die Speicher

Im ATtiny2313 gibt es insgesamt drei Arten von Speicher:

1. Flash

Im Flash sind der Programmcode und einige Daten nicht-flüchtig gespeichert, d.h. die Daten bleiben auch noch nach Abschalten der Versorgungsspannung erhalten. Der Flash kann in der Regel nicht durch das Programm selber gelesen oder beschrieben werden. Um den Flash zu beschreiben wird deshalb meistens ein spezielles Programmiergerät benötigt. Der Flashspeicher des ATtiny2313 kann laut Herstellerangaben ca. 10.000 mal gelöscht und neu beschrieben werden.

2. EEPROM

Im EEPROM (**E**lectrical **E**rasable **P**rogramming **O**nly **M**emory) können Daten nicht-flüchtig gespeichert werden. Der EEPROM kann im Gegensatz zum Flashspeicher vom Programm selber gelesen und beschrieben werden. Dies ist jedoch relativ langsam und umständlich. Der EEPROM des ATtiny2313 kann laut Herstellerangaben 100.000 mal gelöscht und neu beschrieben werden.

Deshalb kann man den EEPROM für kaum veränderliche Daten wie Messwerte oder Konstanten, die im Flash keinen Platz mehr hatten verwenden. Meistens wird er aber überhaupt nicht verwendet.

3. RAM

Im RAM (**R**andom **A**ccess **M**emory) können Daten flüchtig gespeichert werden, d.h. die Daten gehen nach Abschalten der Versorgungsspannung verloren. Im Gegensatz zum EEPROM kann das Programm schnell und einfach lesen und beschreiben. Das RAM des ATtiny2313 ist praktisch unbegrenzt gelöscht und wieder beschrieben werden.

Deshalb wird das RAM meist zum Speichern von stark veränderlichen Daten, wie z.B. Variablenwerten verwendet.

2.7 Die Peripherie

Jeder Mikrocontroller besitzt eine Vielzahl von Peripherie. Dies sind meist bestimmte Funktionseinheiten. Der ATtiny2313 besitzt zum Beispiel die folgenden Peripheriebausteine:

- I/O (Input/Output) Ports
- Interrupt Einheit
- einen 8-Bit Timer/Counter
- einen 16-Bit Timer/Counter
- 4 PWM(Pulsweiten Modulation)-Kanäle
- einen analoger Komparator
- einen programmierbarer Watchdog Timer mit integriertem Oszillator
- ein USI - Universelles Serielles Interface
- einen voll duplex USART - Universal Synchronous Asynchronous Receiver and Transmitter

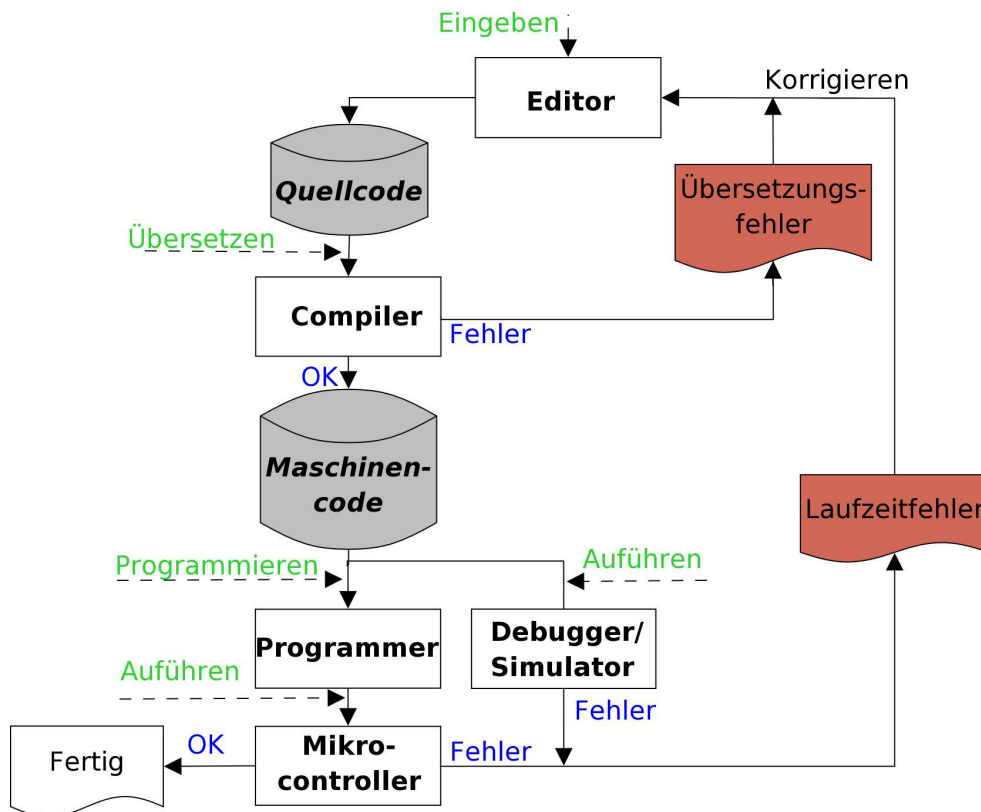
2.7.1 bei unser Frequenzmessung verwendete Peripherie

- 16-Bit Timer/Counter zur Messung der Periodendauer
- USART zur Übertragung der Messwerte
- I/O Ports zum einlesen und ausgeben von digitalen Signalen
- einige Interrupts, um schnell auf Ereignisse zu reagieren (z.B. Zählerüberlauf)

3 Programmierung eines Mikrocontrollers

3.1 Wie schreibe ich ein Programm?

3.1.1 Typischer Ablauf bei der Softwareentwicklung



3.1.2 Hilfsmittel für die Softwareentwicklung

Wie man sehen konnte wird die Softwareentwicklung in den einzelnen Stufen durch unterschiedliche Programme unterstützt, z.B. Editor, Compiler, Assembler, Debugger, Programmierer etc. Sind alle oder viele dieser in einer Softwareoberfläche vereint, so spricht man von einer IDE (Integrated Development Environment).

Für die AVR's gibt es eine kostenlose IDE vom Hersteller, das AVRStudio. Diese beinhaltet neben einem Editor und einem Assembler auch einen Programmierer und einen Debugger/Simulator. Zusätzlich gibt es eine Variante des linux C-Compilers gcc, für die AVR's, der in das AVR-Studio integriert werden kann.

3.2 Wie kommt das Programm in den Mikrocontroller?

Ist das Programm einmal geschrieben und muss es noch irgendwie in den Mikrocontroller gelangen. Dazu hat der ATtiny2313 spezielle Leitungen über die mit Hilfe eines geeigneten Programmiergerätes und einem passenden Programm der Maschinencode in den Flash-Speicher übertragen werden kann.

4 Glossar

Assembler Assembler ist einerseits eine sehr einfache Programmiersprache in der jeder Maschinenbefehl eines Prozessors einfach durch eine für den Menschen (halbwegs) verständliche Abkürzung ersetzt wird. So kann man z.B. mit ADD r20,r21 die Inhalte der Register 20 und 21 addieren.

Andererseits wird aber auch das Programm, das von der Assembler-Sprache in den Maschinencode übersetzt, Assembler genannt.

Compiler Ein Programm was von einer höheren Programmiersprache, z.B. C oder Java, in den Maschinencode eines Prozessors übersetzt.

Debugger Ein Debugger ist ein Programm, das bei der Auffindung und Behebung von Programmierfehlern, Bugs genannt, helfen soll. In einem Debugger kann man in der Regel ein Programm Schritt für Schritt ausführen und dabei die Werte von Variablen und den Programmablauf beobachten kann. Mit Hilfe dieser Informationen findet man viele Programmierfehler, dann deutlich schneller.

PWM Bei der Pulsweitenmodulation kurz PWM wird das Verhältnis zwischen der Zeit, die ein Rechtecksignal auf HIGH und der Zeit die es auf LOW ist, proportional zu einem digitalen Wert verändert. Die Frequenz des Rechtecksignals bleibt dabei aber unverändert.

Schickt man des PWM-Signal anschließend durch einen RC-Tiefpass, so ist die sich am Ausgang einstellende Spannung proportional zu dem digitalen Wert, durch den das Signal moduliert wurde. Man kann so also einen einfachen Digital-Analog-Wandler herstellen.

Simulator Ein Simulator ist ein Programm, das versucht möglichst genau das Verhalten eines Mikrocontrollers bei der Ausführung eines Programms zu simulieren.

Timer/Counter Ein Timer/Counter ist ein universeller digitaler Zähler zum zählen von Ereignissen und Messen von Zeitspannen.

Watchdog Timer Jeder hat schon mal erlebt, dass sein Computer abstürzt oder hängen bleibt. Der einzige Ausweg besteht oft darin den Computer, zur Not durch ziehen des Steckers, neu zu starten.

Bei Mikrocontrollern ist das bei sehr komplizierten Programmen auch nicht anders. Der Watchdog Timer ist deshalb eine vom restlichen Mikrocontroller völlig unabhängige Einheit, die den Mikrocontroller reseted, wenn in dieser einem bestimmten Zeitraum nicht durch setzen eines Bits betätigt, dass er noch ordnungsgemäß funktioniert.

Literatur

- [1] Dr.-Ing. Flik, Thomas: Mikroprozessortechnik und Rechnerstrukturen, Springer (6. Aufl., Berlin Heidelberg, 2005)
- [2] Müller Helmut; Walz Lothar : Mikroprozessortechnik, Vogel (5. Aufl., Würzburg, 1999)
- [3] Volpe, Safinaz; Francesco P. Volpe: AVR-Mikrocontroller-Praxis, Elektor-Verlag (2. Aufl.,Aachen, 2001)
- [4] Heiß, Hans-Ulrich; Dr.-Ing. Flik, Thomas: Vorlesungsfolien Informatik 2 (SS 2005)
- [5] ATtiny2313 Datasheet: http://www.atmel.com/dyn/resources/prof_documents/doc2543.pdf