



PIC-Grundlagen Teil 1

Mikrocontroller haben sich in der Elektronik ein großes Terrain erobert, nicht zuletzt durch ihre riesige Vielfalt an Derivaten. Besonders im Bereich kleiner Anwendungen haben sich die PIC-Mikrocontroller aus dem Hause Microchip durchgesetzt, die wir in der vorliegenden Artikelserie vorstellen wollen.

Allgemeines

Seitdem Mikrocontroller in der Elektronik eingesetzt werden, sind diese auch für den Hobbyelektroniker sehr interessant. Besonders beliebt ist hier die MCS51-Familie, bei der meist ein Mikrocontroller eingesetzt wird, der sein Betriebsprogramm aus einem externen EPROM erhält. Diese Controller sind leistungsfähig und ermöglichen auch die Realisierung größerer Anwendungen.

Aber auch in kleinen Anwendungen erscheint es oft sinnvoll, einen Mikrocontroller einzusetzen, wenn es zum Beispiel darum geht, eine Tastatur auszuwerten und eine Hardware anzusteuern. Für diesen Anwendungsfall wurden die PIC-Mikrocontroller entwickelt, die auf dem Chip über Programm- und Arbeitsspeicher verfügen und somit nur mit wenigen externen Bauelementen beschaltet werden müssen. Daraus ergibt sich auch eine verringerte Anzahl an Pins, so daß die Controller nur wenig Platz auf der Leiterplatte benötigen. Ebenso ist der Preis dieser Mikrocontroller interessant, selbst beim Einsatz in Großserien.

Zwischenzeitlich sind die PIC-Controller in verschiedenen Versionen erhältlich, so daß für fast jede Anwendung ein optimaler PIC verfügbar ist. So sind zum Bei-

spiel Controller mit internem AD-Wandler oder I²C-Bus-Interface erhältlich. Aber auch größere PICs mit mehr Programm- und Arbeitsspeicher sind verfügbar, für umfangreichere Anwendungen.

Durch geschickte Vermarktung der PIC-Mikrocontroller sind diese auch für den

Hobbyelektroniker sehr interessant geworden. So wird der Assembler für die PIC-Controller von Microchip kostenlos zur Verfügung gestellt, und es gibt Programmerroutinen zu verschiedenen Anwendungen, z. B. aus dem Internet.

Zudem bietet Microchip ein Starterkit mit Datenbüchern, dem Assembler, einem Softwaresimulator, einem Programmiergerät und ein paar Mikrocontrollern an, das zu einem günstigen Preis erhältlich ist.

Mit dem Starterkit können für alle derzeit von Microchip verfügbaren PIC-Controller Programme erstellt, getestet und programmiert werden.

Die verschiedenen Mikrocontrollertypen sind in Familien eingeteilt, in denen sich die Mikrocontroller hauptsächlich durch die Größe des Programm- und Datenspeichers sowie der I/O-Pins unterscheiden. Zwischen den Familien gibt es kleine Unterschiede in der Anzahl und der Wortbreite der Befehle.

Die Mikrocontroller verfügen über ein internes EPROM, in dem das auszuführende Programm gespeichert ist. Die ICs sind dabei nur einmal programmierbar und können nicht gelöscht werden. Für die Entwicklung gibt es spezielle Versionen, die über ein Löschenfenster verfügen. Diese können mit einem UV-Löschgerät gelöscht und neu programmiert werden.

Die PIC 16C5x-Familie

Die wesentlichen Eigenschaften der PIC 16C5x-Familie sind:

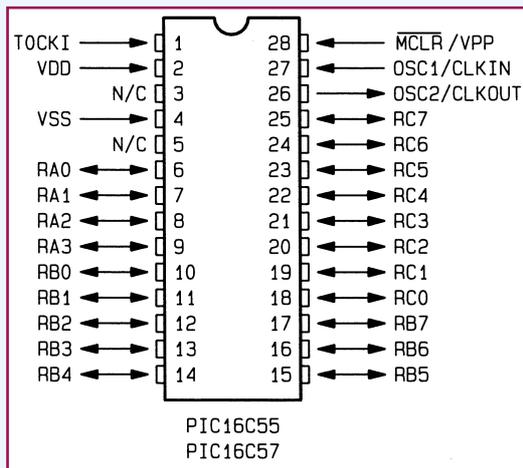
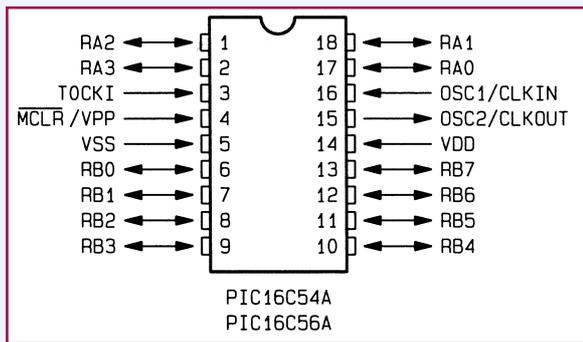
- 33 Ein-Wort-Befehle
- Alle Befehle (bis auf Sprungbefehle) benötigen nur einen Zyklus (200 ns bei 20 MHz).

Tabelle 1: Typen der PIC 16C5x-Familie

Typ	EPROM	RAM	I/O-Ports	Anschlüsse
PIC 16C54A	512 x 12 Bit	25 Byte	12	18pol.
PIC 16C58A	2048 x 12 Bit	73 Byte	12	18pol.
PIC 16C55	512 x 12 Bit	24 Byte	20	28pol.
PIC 16C57	2048 x 12 Bit	73 Byte	20	28pol.

Tabelle 2: Funktion der Anschlußpins

Name	Typ	Funktion
RA0 bis RA3	I/O	Bidirektionale Ein-/Ausgänge Port A
RB0 bis RB7	I/O	Bidirektionale Ein-/Ausgänge Port B
RC0 bis RC7	I/O	Bidirektionale Ein-/Ausgänge Port C
TOCKI	I	Takteingang für Timer 0
MCLR/Vpp	I	RESET-Eingang (Low-Pegel löst Reset aus)
OSC1/CLKIN	I	Takteingang
OSC2/CLKOUT	O	Taktausgang
VDD		Positive Versorgungsspannung
VSS		Negative Versorgungsspannung
N/C		Nicht angeschlossen



tet jedoch nur korrekt, wenn die Betriebsspannung mit mindestens 0,05 V/ms ansteigt. Andernfalls muß der MCLR-Pin mit einer RC-Kombination beschaltet werden (siehe Abbildung 4). Dabei sollte der Widerstand nicht größer als 40kΩ sein.

Architektur

Der PIC-Controller arbeitet im Gegensatz zu anderen Controllern mit zwei getrennten Bussen für Daten und Befehle, so daß er mit einem 8 Bit breiten Datenbus und einem 12 Bit breiten Befehlsbus arbeiten kann. Durch die 12Bit-Befehle ist es möglich, Ein-Wort-Befehle zu verwenden.

971188701A

Bild 1: Anschlußbelegung der PICs 16C54A/58A/55/57

- Wählbarer Oszillator mit einer Taktfrequenz von 0 bis 20 MHz
- 12 Bit breite Befehle
- 8 Bit breiter Datenbus
- 512 x 12 Bit bis 2048 x 12 Bit interner Programmspeicher (EPROM)
- 24 bis 73 Byte Datenspeicher (RAM)
- Direkte und indirekte Adressierung
- Integrierter Watch-Dog-Timer
- Integrierter Zähler/Teiler
- 12 bis 20 I/O-Pins
- Betriebsspannung von 2,5 V bis 6,0 V

Einige Typen der PIC-16C5x-Familie sind in Tabelle 1 aufgeführt, die Anschlußbelegung ist aus Abbildung 1 ersichtlich. Die Funktionen der Anschlußpins sind in Tabelle 2 dargestellt.

Oszillator

Bei der PIC 16C5x-Familie können 4 verschiedene Oszillatortypen gewählt werden:

- LP: langsamer Quarz
- XT: Quarz-Oszillator 100kHz bis 4 MHz
- HS: Quarz-Oszillator 4 MHz bis 20 MHz
- RC: RC-Oszillator

Bei den Entwicklungstypen mit Löschenfenster (tragen in der Typ-Bezeichnung die Erweiterung /JW) ist die Konfiguration des Oszillators über zwei Bit einstellbar und wird bei der Programmierung vorgegeben.

Bei den OTP-Versionen (ohne Löschenfenster) ist dagegen der Oszillatortyp fest vorgegeben, so daß diese bereits mit der entsprechenden Oszillatoreinstellung eingesetzt werden müssen. Sie tragen dann die Erweiterung LP, XT, HS oder RC.

Bei den Typen PIC 16C54A und PIC

16C58A handelt es sich um überarbeitete Versionen (durch das A gekennzeichnet), bei denen auch bei der OTP-Version der Oszillator bei der Programmierung wählbar ist. Hierbei muß lediglich auf die maximale Taktfrequenz geachtet werden, damit ein IC, das mit maximal 4MHz-Taktfrequenz angegeben ist, nicht im HS-Mode mit einem 20MHz-Quarz beschaltet wird.

Die Beschaltung eines Quarz-Oszillators ist in Abbildung 2 dargestellt. Die Werte für die Kondensatoren sind abhängig vom Oszillatormode. Für den LP-Mode sind 33pF-, und für den XT- und HS-Mode 22pF-Kondensatoren zu empfehlen. Bei einigen Oszillatoren kann zusätzlich ein Serienwiderstand RS notwendig sein.

Ebenfalls kann der Mikrocontroller extern mit einem Takt versorgt werden, der dann an den Pin OSC1 anzuschließen ist, wobei der Pin OSC2 nicht beschaltet wird. Dabei muß je nach Taktfrequenz der LP, XT oder HS-Mode gewählt werden.

Für Anwendungen, bei denen eine genaue Oszillatorfrequenz keine Rolle spielt, kann der PIC auch mit einem RC-Oszillator betrieben werden (siehe Abbildung 3). Aus Gründen der Stabilität sollte ein Widerstand im Bereich zwischen 3 kΩ und 100 kΩ und der Kondensator größer als 20 pF gewählt werden. Zu beachten ist hierbei, daß die Oszillatorfrequenz aufgrund von Bauteiltoleranzen und Temperaturdrift stark schwanken kann.

Die durch 4 geteilte Oszillatorfrequenz liegt am Ausgang OSC2 an und ist auch für andere Schaltungskomponenten nutzbar.

Reset

Die Mikrocontroller verfügen über eine interne Reset-Schaltung, die es erlaubt, den MCLR/Vpp-Pin mit der positiven Versorgungsspannung zu verbinden. Nach dem Einschalten der Versorgungsspannung führt der Mikrocontroller dann intern einen Reset durch, währenddessen der Oszillator anschwingt. Diese Funktion arbei-

tigt er, bis auf die Sprungbefehle, für jeden Befehl nur einen Maschinenzklus.

Alle Special-Function-Register (zur Steuerung der Ports, Timer usw.) und der Datenspeicher sind im gleichen Adreßbereich (den File-Registern) angeordnet. So besteht kein Unterschied, ob mit einem Befehl auf ein Byte im Datenspeicher oder auf einen Ausgangsport zugegriffen wird.

Der Controller verfügt über eine 8 Bit breite ALU (Arithmetisch-Logische-Einheit) und ein W-Register (Arbeitsregister). Bei Befehlen mit zwei Operanden findet die Verknüpfung immer zwischen dem W-Register und einem File-Register statt.

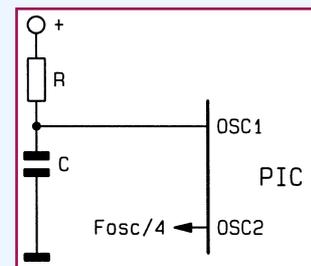


Bild 3: Beschaltung eines RC-Oszillators
971188703A

Der Controller verfügt über eine 8 Bit breite ALU (Arithmetisch-Logische-Einheit) und ein W-Register (Arbeitsregister). Bei Befehlen mit zwei Operanden findet die Verknüpfung immer zwischen dem W-Register und einem File-Register statt.

Der Controller verfügt über eine 8 Bit breite ALU (Arithmetisch-Logische-Einheit) und ein W-Register (Arbeitsregister). Bei Befehlen mit zwei Operanden findet die Verknüpfung immer zwischen dem W-Register und einem File-Register statt.

Der Controller verfügt über eine 8 Bit breite ALU (Arithmetisch-Logische-Einheit) und ein W-Register (Arbeitsregister). Bei Befehlen mit zwei Operanden findet die Verknüpfung immer zwischen dem W-Register und einem File-Register statt.

Programmspeicher

Der Programmspeicher der PIC 16C5x-Familie ist in Seiten zu je 512 x 12 Bit

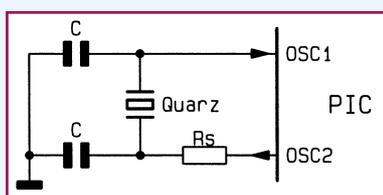


Bild 2: Beschaltung eines Quarzoszillators

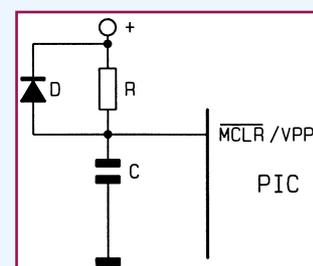


Bild 4: Externe Reset-Schaltung
971188704A

aufgeteilt. Die Adressierung erfolgt über einen Programmzähler (PC). Die Controller PIC 16C54A und PIC 16C55 verfügen über eine Seite Programmspeicher, der mit einem 9 Bit breiten Programmzähler adressiert werden kann. Der PIC 16C56A sowie der PIC 16C57 verfügen über 4 Seiten Programmspeicher und einen 11 Bit breiten Programmzähler.

Die Aufteilung des Programmspeichers ist in Abbildung 5 dargestellt.

Bei der Programmierung ist darauf zu achten, daß der Reset-Vektor (ab dieser Speicherstelle wird das Programm nach einem Reset ausgeführt) am Ende des Programmspeichers liegt, also beim PIC 16C54A und PIC 16C55 auf 1FFh und bei dem PIC 16C56A und PIC 16C57 auf 7FFh. An diesem Programmplatz muß auf jeden Fall ein Sprung zum eigentlichen Programmanfang stehen.

Datenspeicher

Wie schon erwähnt, befinden sich die Register zur Steuerung des Controllers im gleichen Adreßraum wie der Datenspeicher. Die Aufteilung des Speichers erfolgt in Bänke zu je 32 Register.

Somit verfügen der PIC 16C54A und der PIC 16C55 über eine Bank (siehe Abbildung 6). Die unteren Register dienen dabei zur Steuerung des Controllers und die oberen stehen als Datenspeicher zur Verfügung. Damit erklärt sich auch die Tatsache, daß der PIC 16C54A ein Byte mehr RAM aufweist, da ein Register dem PORT C entspricht.

Die Controller PIC 16C56A und PIC 16C57 verfügen über insgesamt 4 Bänke (siehe Abbildung 6), wobei die Bank 0 von 00h bis 1Fh adressierbar ist. Die Bänke 1 bis 3 hingegen beginnen an den Adressen 30h, 50h und 70h und reichen bis 3Fh, 5Fh und 7Fh. Ist nun eine der Bänke 1 bis 3 aktiviert und es wird auf die Adressen 00h bis 0Fh zugegriffen, so erreicht man hiermit die „gespiegelten“ Adressen 00h bis 0Fh der Bank 0. Somit kann aus jeder Bank heraus auf die Register zur Steuerung des Controllers zugegriffen werden. Ebenfalls stehen die RAM-Speicherstellen 08h bis 0Fh der Bank 0 aus jeder anderen Bank heraus

Bild 5: Aufteilung des Programmspeichers 971188705A

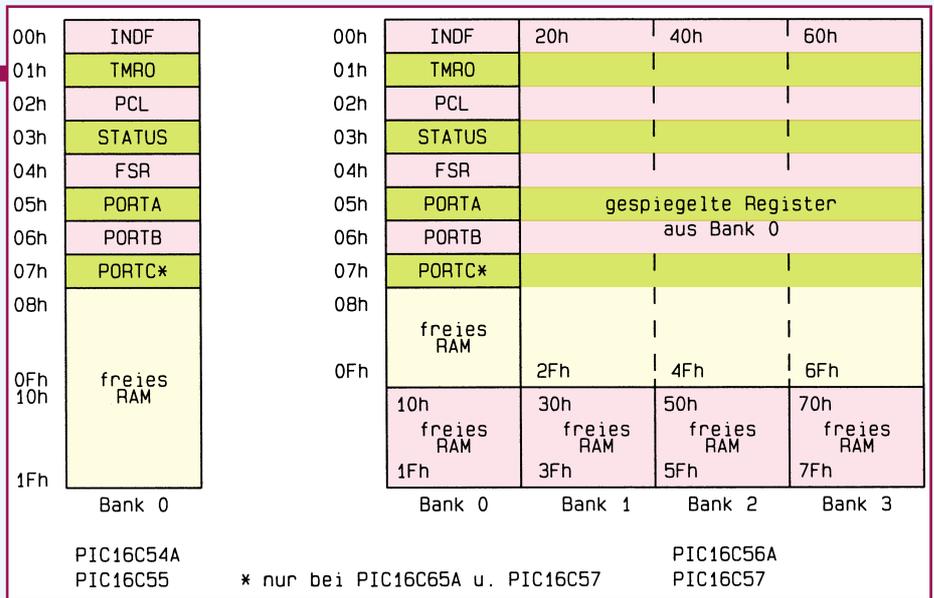
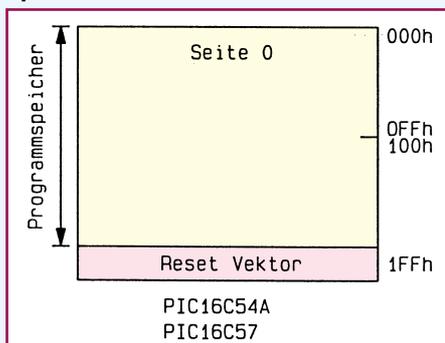


Bild 6: Aufteilung des Datenspeichers 971188706A

zur Verfügung und können somit für globale Variablen genutzt werden.

Die I/O-Ports

Die interne Struktur der I/O-Ports ist in Abbildung 7 dargestellt. Die Ports des PICs sind bidirektional und müssen deshalb als Ein- oder Ausgang konfiguriert werden.

Jedem Port ist ein Tristate-Register (TRIS-Register) zugeordnet, wobei jedes Bit einem Pin entspricht. Nach einem Reset sind diese Register mit „1“ zu beschreiben, so daß

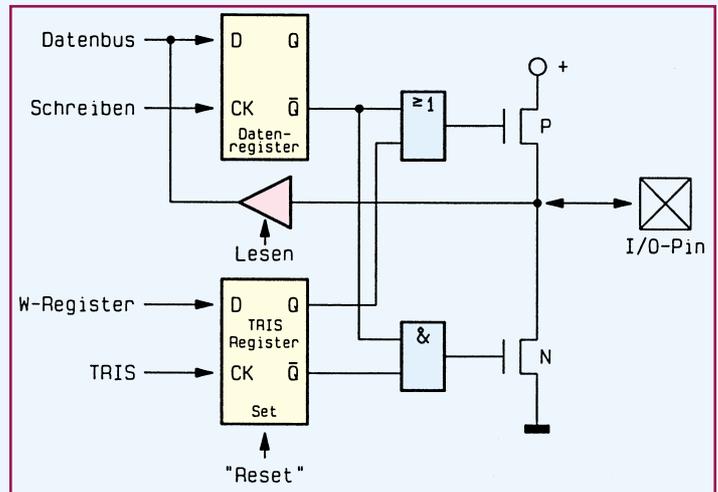


Bild 7: Interne Struktur des I/O-Ports 971188707A

alle Port-Pins als Eingang geschaltet sind. Wie schon zuvor beschrieben, befinden sich die Portregister im gleichen Adreßbereich wie der Datenspeicher, so daß über die entsprechende Speicherstelle der Zustand der Ports lesbar ist.

Um einen Pin als Ausgang zu konfigurieren, muß das entsprechende Bit des Tristate-Registers gesetzt werden. Dann wird der Port über zwei interne Transistoren aktiv auf High- oder Low-Potential gezogen.

Um einen Open-Kollektor-Ausgang zu realisieren, bei dem nur der Transistor nach Masse schaltet, ist das Port-Register zu löschen und über das TRIS-Register der Ausgang zu aktivieren (mit 0 beschreiben) oder zu sperren (mit 1 beschreiben).

Eine Besonderheit bildet das Beschreiben der TRIS-Register, die nicht wie die anderen Register mit im Datenspeicher angeordnet sind. Sie werden über gesonderte Befehle direkt mit Werten aus dem W-Register geladen.

Im zweiten Teil dieser Artikelserie folgt die Beschreibung der Control-Register und des Timers.