



Einführung in die Microchip PIC-Controller

vergleichende Betrachtung zu anderen Microcontrollern

CLUG 12.2.2016

Matthias Kupfer

maku@kupfer-it.de

Übersicht

Überblick über Microchip PIC-Familie

Eigenschaften (Vergleich zu AVR)

Technische Voraussetzungen

Aufbau & Struktur

Entwicklungszyklus & Werkzeuge

Assembler

Compiler

Besonderheiten

Einfache Beispiele/Demo

Informationsquellen

Überblick PIC-Familie

PIC Controller entstanden aus PIC1650 von General Instruments

Microchip PIC-Controller sind RISC-Chips

33 (Baseline) bis 70 (Enhanced) Befehle

kompatibles Pinout

Controller mit unterschiedlicher Pin-Anzahl können getauscht werden

Harvard-Architektur

Überblick PIC-Familie

PIC-Controller haben grundsätzlich einfaches Namensschema (Einige Sonderfälle bleiben unberücksichtigt)

PIC xx $yzzz$ *

xx mit den Werten **10**, **12**, **16** für 8 Bit mit 6, 8 oder 14-64 Pins, sowie **24** für 16 Bit und **32** für 32 Bit

y für den Speichertyp oder die Ausführung: **F** für Flash, **C** für EEPROM

zzz * 2-4 stellige Nummer für den konkreten Microcontrollertyp ggf. mit ergänzenden Buchstaben für Weiterentwicklung

Wir betrachten einen (alten) verbreiteten PIC16F84A

18 Pin, 20 MHz, 1024 Words Flash, 68 Bytes RAM, 64 Bytes EEPROM

Words bedeutet Befehle der angegebenen Familie, hier 14 Bit breit

Eigenschaften

PIC benötigen 1 Zyklus pro Befehl, bei Sprungbefehlen auch 2 Zyklen (AVR: 1-4)

Ein Zyklus benötigt 4 Takte (AVR: 1)

RISC Befehlssatz fester Länge (AVR: nicht orthogonaler Befehlssatz variabler Länge)

Befehlssatz hängt vom Typ ab (AVR: weitestgehend einheitlicher Befehlssatz)

PICs gibt es mit vielen Unterstützungen (USB, Ethernet, CAN, RF usw.) (AVR: nur sehr wenige ausgewählte μC)

Eigenschaften

Original Programmer (inkl. Debugger) PicKit 3 ~ 83,- €
(zuzüglich Software) (AVR: AVR ISP mk II 52,- €, vglb.
Dragon 73,- €)

Alle (auch sehr alte) μ C erhältlich (AVR: alte μ C nicht
mehr verfügbar)

Speicher in Bänken organisiert (AVR: linearer Speicher)

Nur ein Register (W) (AVR: 32 Register)

Einfache μ C (PIC16F84A) nur sehr wenig Peripherie,
kein ADC, kein PWM (AVR: ADC, PWM)

Technische Voraussetzungen

mindestens zum Betrieb erforderlich sind (keine interne Taktquelle)

Quarz (bis 20 MHz)

2 Kondensatoren (22 pF)

zum Flashen erforderlich ist

Programmierertools von Microchip oder

Eigenbau-Brenner (siehe Beispiel) oder

ICSP (erfordert eigene Schaltung oder Adapter)

Aufbau und Struktur

PIC16F84A verfügt über folgende Anschlüsse

2 Spannungsversorgung (VSS, VDD)

2 Takt (OSC1, OSC2)

OSC1 auch als Takteingang, OSC2 als Taktausgang
möglich

1 Reset (MCLR, auch zur Programmierung)

GPIO-Port A (RA0-4)

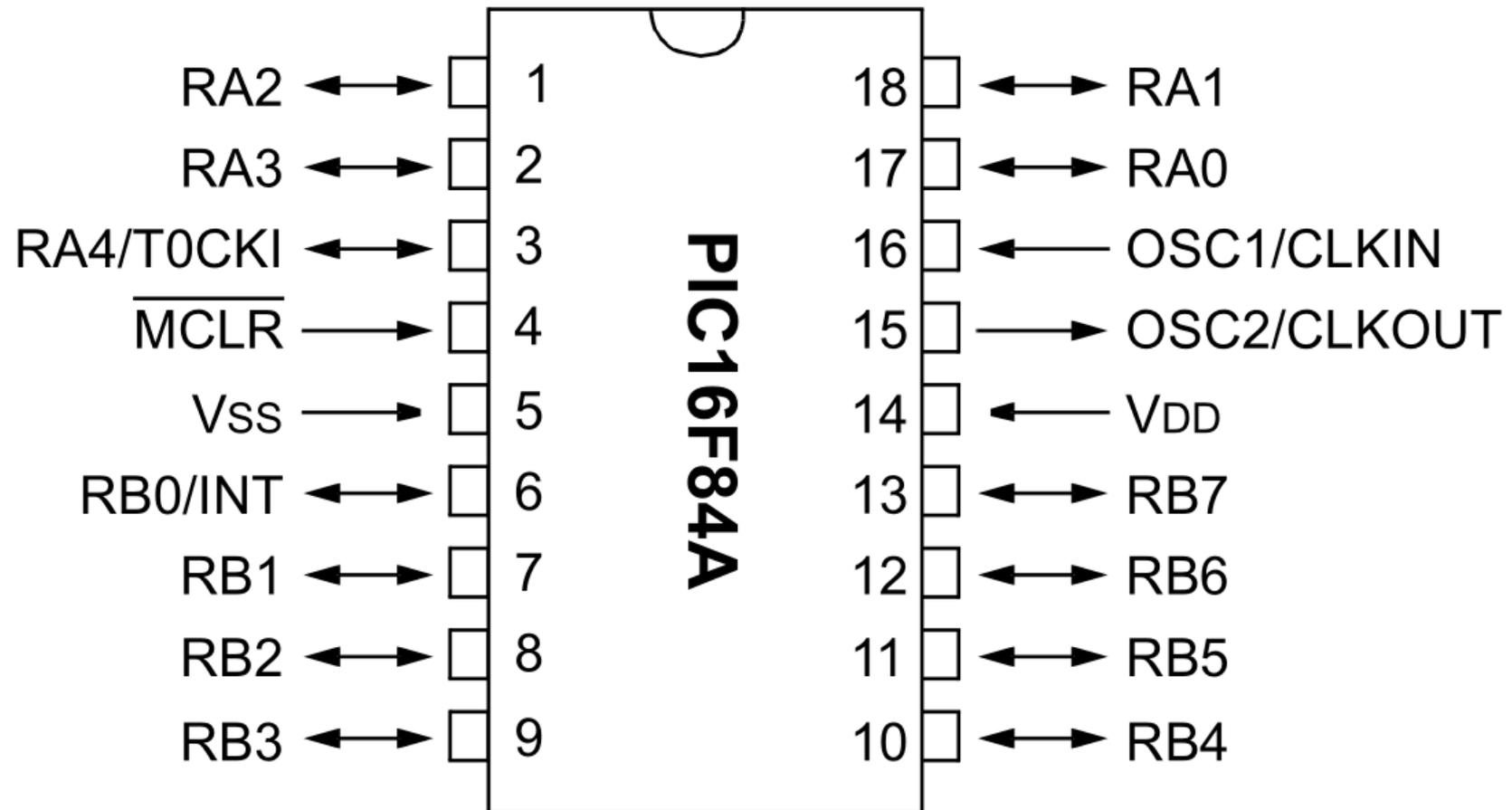
RA4 auch als Clock Input für Timer möglich

GPIO-Port B (RB0-7)

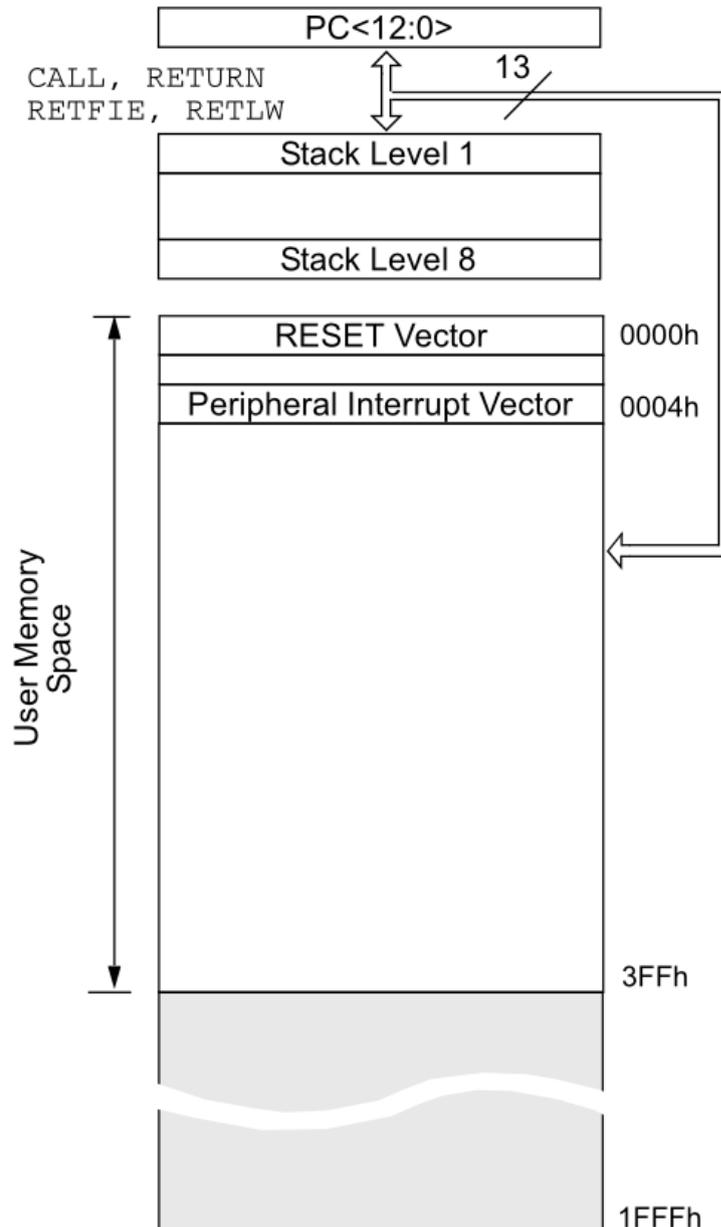
RB0 auch als externer Interrupt Pin möglich

Aufbau und Struktur

PDIP, SOIC



Aufbau und Struktur



Stack

Bis zu 8 Stufen

Interrupt-Vektoren

Programm-Speicher

hier 14 Bit-Worte

Aufbau und Struktur

File Address		File Address
00h	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾
01h	TMR0	OPTION_REG
02h	PCL	PCL
03h	STATUS	STATUS
04h	FSR	FSR
05h	PORTA	TRISA
06h	PORTB	TRISB
07h	—	—
08h	EEDATA	EECON1
09h	EEADR	EECON2 ⁽¹⁾
0Ah	PCLATH	PCLATH
0Bh	INTCON	INTCON
0Ch		
	68 General Purpose Registers (SRAM)	Mapped (accesses) in Bank 0
4Fh		
50h		
7Fh		
	Bank 0	Bank 1

□ Unimplemented data memory location, read as '0'.

Note 1: Not a physical register.

Speicherorganisation

2 Bänke

Bänke faktisch nur für
Register relevant

Register teilweise auch
abgebildet

Register bis 0Bh/8Bh

68 Speicherzellen

(2. Bank auf 1. abgebildet)

teilweise nicht nutzbarer
Bereich

Entwicklungszyklus & Werkzeuge

Zum Entwickeln stehen C/C++-Compiler und Assembler zur Verfügung

Die fertigen Programme liegen als HEX-File vor und werden auf den Microcontroller geflasht

Ablauf entspricht grundsätzlich auch den anderen Microcontrollern (z.B. AVR)

Besonderheit gegenüber AVR: Alle Konfigurationen sind in einem HEX-File

Entwicklungszyklus & Werkzeuge

Microchip bietet C/C++-Compiler und Assembler zusammen MPLAB X IDE

Freie C/C++ Compiler besitzen nur eingeschränkte Optimierung,
andere Compiler kostenpflichtig, teilweise Binärpakete für Linux
verfügbar

MPLAB XC Compiler (siehe Beispiel)

MPASM

Nur für Windows

Sonstige Werkzeuge und Simulationssoftware kostenpflichtig

Freie Software

gpasm Assembler aus Paket gputils

C Compiler aus dem sdcc (mit Einschränkungen – großer Code)

gpsim als einfacher Simulator

IDE verfügbar z.B. pikLab

Besonderheiten

Keine Multiplikation verfügbar

Keine gesonderten Vergleichsfunktion

Variable == 0

```
movf var, f
```

```
btss STATUS, Z
```

Statt movf auch beliebiger anderer logische arithmetischer Befehl möglich

Zustand wiederherstellen nur mit Trick möglich

Register W wiederherstellen:

```
swapf w_temp, f
```

```
swapf w_temp, w
```

Grund: swapf verändert als einziger Befehl nicht die Status Bits

Beispiele/Demo

Makefiles, Hexfiles

Kleine Assemblerbeispiele

- LED blinken, Blinkmuster

- LCD Ansteuerung

- IR Sender (ohne Demo nur Quellcode)

Demo Programmieren und Brennen

Demo Programme auf PIC16F84A

Bonus: Disassembler schnell entwickelt

Informationsquellen

Hilfreiche Infos zu PIC und deren Programmierung:

Datenblätter von www.microchip.com

www.sprut.de

www.mikrocontroller.net