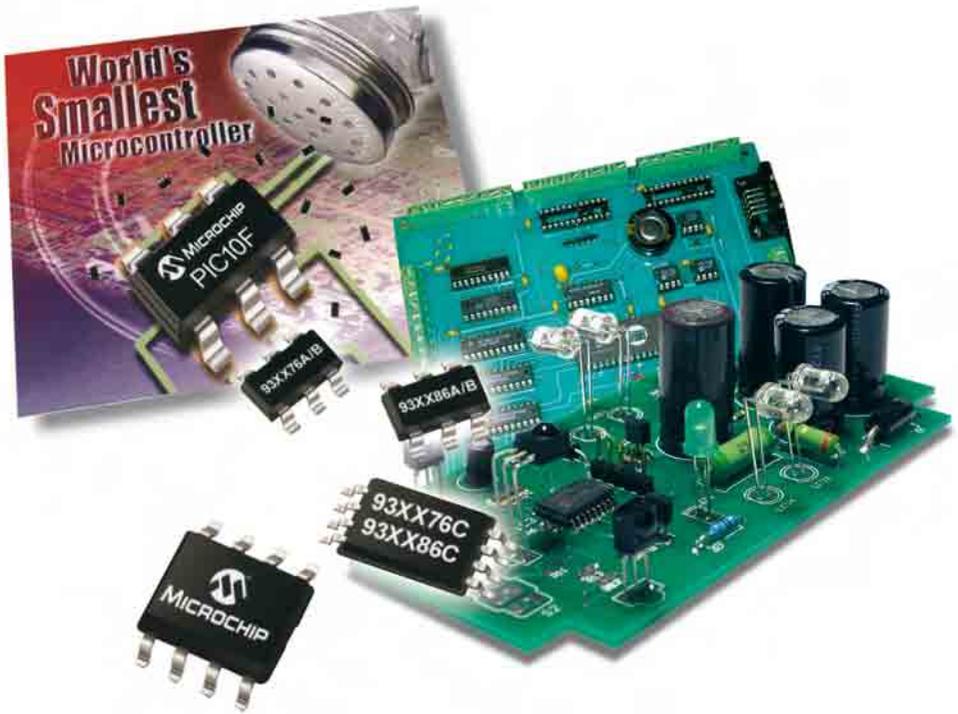


Anne & Manfred König



Das große

2. überarbeitete Auflage

# PIC-Micro Handbuch

# Vorwort

Die erste Auflage dieses Buches wurde Ende des Jahres 2004 fertiggestellt. Seitdem haben sich nicht nur die Microcontroller verändert, sondern vor allem die Welt um sie herum. Kann man ein Microcontroller-Buch nach zweieinhalb Jahren in die zweite Auflage schicken, ohne es total umzugestalten?

Microchip ist es erfreulicherweise gelungen, trotz aller neuen Anforderungen dem Basiskonzept treu zu bleiben. Bei der Durchsicht des Buches haben wir zu unserer Freude festgestellt, daß das meiste, was wir damals geschrieben haben, auch heute nicht von gestern ist.

Schon bei der ersten Auflage war es nicht mehr möglich, alle Details in einem Buch sinnvoller Dicke unterzubringen. Wir hatten deshalb das Augenmerk auf die grundlegende Funktionsweise der PICs gelenkt. In dem großen Kapitel über die PIC18-Familie wurde gezeigt, wie der breitere Programmspeicher zusammen mit einer leichten Modifikation der Architektur den PIC auch für größere Programme komfortabel macht.

Der Trend zu größeren PICs wurde einerseits fortgesetzt (PIC24), andererseits hat Microchip aber auch sehr viel Aufwand in die „kleinen“ PICs investiert. Einen Überblick über die Neuerungen der letzten Zeit geben wir gleich am Anfang des ersten Kapitels. Wesentlich verändert wurde das Kapitel über MPLAB.

Obwohl wir im ersten Kapitel die fundamentalen Grundlagen der Architektur und der Funktionsweise von PIC-Microcontrollern beschreiben, ist dies kein Anfängerbuch. Wir setzen elementare Kenntnisse in der Elektronik und der Datenverarbeitung voraus.



# Inhalt

<b>PIC-Microcontroller im Jahr 2007</b> .....	15
Neues bei den 8 Bit-PICs .....	15
Baustein-Auswahl .....	16
<b>1 Grundlagen</b> .....	19
1.1 Architektur und Funktionsweise .....	20
1.2 Befehlsstruktur .....	22
1.3 Der Befehlssatz .....	24
1.4 Datenspeicher .....	25
1.5 Indirekte Daten-Adressierung .....	27
1.6 Programmspeicher .....	27
1.7 I/O-Ports .....	30
1.8 Special Function Register .....	33
1.9 Interrupts .....	33
1.10 Oszillator .....	37
1.11 Timer .....	39
1.11.1 Timer0 .....	39
1.11.2 Timer1 .....	40
1.11.3 Timer2 .....	41
1.11.4 Watchdog-Timer .....	41
1.12 SLEEP-Modus .....	42
1.13 RESET .....	43
1.13.1 Power-on RESET .....	44
1.13.2 Brown-out RESET .....	45
1.13.3 /MCLR- und WDT-RESET .....	45
1.14 Hardware-Module .....	46
1.14.1 AD-Wandler .....	47
1.14.2 Komparator .....	49
1.14.3 Komparator mit SR-FlipFlop .....	51
1.14.4 CCP/ECCP-MODUL .....	52
1.15 Konfiguration .....	56

<b>2</b>	<b>Serielle Kommunikationen</b> . . . . .	58
2.1	Eigenschaften serieller Schnittstellen . . . . .	58
2.1.1	Bitmanagement . . . . .	59
2.1.2	Bitfelder . . . . .	59
2.1.3	Master und Slaves . . . . .	60
2.2	SSP-Modul (SPI und I2C) . . . . .	61
2.2.1	SPI-Betriebsweise . . . . .	62
2.2.2	SPI-Beispiel . . . . .	63
2.2.3	SPI-Initialisierung . . . . .	64
2.2.4	SPI-Probleme? . . . . .	64
2.2.5	I2C-Betriebsweise mit SSP/MSSP-Modul. . . . .	65
2.2.6	I2C-Initialisierung . . . . .	66
2.2.7	I2C-Betriebsweise ohne Hardware-Modul . . . . .	66
2.3	USART-Modul. . . . .	67
2.3.1	Asynchroner Betrieb (UART) . . . . .	67
2.3.2	Addressable USART (AUSART). . . . .	70
2.3.3	Initialisierung . . . . .	70
2.3.4	Enhanced USART (EUSART). . . . .	73
2.3.5	RS232-Anwendung . . . . .	73
2.4	CAN-Bus . . . . .	75
2.4.1	Nun folgt ein Auszug aus der Diplomarbeit von Klaus Koschinsky: . . . . .	76
2.4.2	Beispiel eines CAN-Programms . . . . .	78
2.5	LIN-Bus . . . . .	80
2.5.1	LIN-Betriebsweisen . . . . .	81
2.5.2	LIN-Firmware-Lösung. . . . .	82
2.6	USB . . . . .	88
2.6.1	Einstiegshilfen von Microchip . . . . .	89
2.6.2	Antworten auf Einsteigerfragen . . . . .	92
<b>3</b>	<b>PIC18</b> . . . . .	96
3.1	Architektur und CPU . . . . .	97
3.2	Programmspeicher . . . . .	99
3.3	Zugriffe auf den Programmspeicher. . . . .	99
3.3.1	Lesen des Programmspeichers . . . . .	100
3.3.2	Schreiben in den Programmspeicher . . . . .	100
3.4	Datenspeicher. . . . .	103
3.4.1	Adressierung der File-Register . . . . .	103
3.4.2	Return Stack. . . . .	104
3.4.3	Fast Register Stack. . . . .	106
3.5	I/O Ports . . . . .	107
3.5.1	LAT-Register . . . . .	107
3.5.2	PORTA bis PORTL . . . . .	109

3.6	Timer	110
3.6.1	Gepufferte 16-Bit-Timer-Register	111
3.6.2	TMR0	112
3.6.3	Watchdog Timer	113
3.6.4	Timer1 und Timer3	113
3.6.5	TIMER2 (und TIMER4)	115
3.7	Interrupts	116
3.8	Reset	120
3.9	Hardware-Module	120
3.10	Neue Befehle	121
3.10.1	Befehlsstruktur	121
3.10.2	Statusregister	122
3.10.3	Befehle mit File-Register-Argument	122
3.10.4	Neue Arithmetikbefehle	124
3.10.5	Bit-Befehle	125
3.10.6	Toggle	125
3.10.7	Befehle mit indirekter Adressierung	126
3.10.8	Zwei-Wort-Befehle	128
3.10.9	Relative Verzweigungsbefehle	131
3.10.10	Neue bedingte Skip-Befehle	132
3.10.11	Multiplikation	132
3.11	Kompatibilität	134
3.11.1	Hardware-Kompatibilität	135
3.11.2	Assembler-Kompatibilität	135
3.11.3	Was sagt MPASM18?	139
3.11.4	Fazit (Checkliste)	143
3.11.5	Abwärtskompatibilität	144
<b>4</b>	<b>Power Management</b>	<b>146</b>
4.1	Betriebsarten	147
4.2	Oszillatorklassen	147
4.3	OSCCON-Register	148
4.4	Power-managed Modes	149
4.5	Moduswechsel im RUN-Zustand	150
4.5.1	SLEEP-Befehl	150
4.5.2	Rückkehr zum Primary RUN-Mode	151
4.6	Aufwachen aus IDLE und SLEEP	152
4.6.1	Aufwachen durch Interrupt	152
4.6.2	Aufwachprozeß	153
4.7	Oszillatorwechsel	153

<b>5</b>	<b>Die neuen PIC10F</b> .....	156
5.1	Die Eigenschaften im Überblick: .....	156
5.2	Hardware-Eigenschaften .....	157
5.2.1	Speicherausbau und vorhandene Module .....	157
5.2.2	Gehäuseformen und Pinanzahl .....	157
5.2.3	Interner RC-Oszillator .....	157
5.2.4	In-circuit serielle Programmierung .....	159
5.2.5	Die inneren Werte; 5X-Core und seine Folgen .....	160
5.2.6	Die Pins im Einzelnen: .....	161
5.2.7	Das Komparator-Modul .....	164
5.2.8	Der 8-Bit AD-Wandler .....	165
<b>6</b>	<b>rfPIC</b> .....	166
6.1	Der PIC-Teil .....	167
6.2	Der HF-Teil .....	167
6.2.1	Maximale Datenrate .....	167
6.2.2	Modulation .....	167
6.2.3	Sendefrequenzen .....	168
6.2.4	Schaltbare Ausgangsleistung .....	169
<b>7</b>	<b>PIC-Programmieren in Assembler</b> .....	170
7.1	Zahlenformate .....	170
7.1.1	Worte aus zwei Bytes .....	171
7.1.2	Negative Zahlen .....	172
7.1.3	Bruchrechnung .....	173
7.1.4	Rechnen mit Exponentialformaten .....	175
7.1.5	Entscheidung über Formate .....	177
7.1.6	Genauigkeit .....	179
7.2	Funktionen .....	180
7.3	Verwendung von Makros .....	181
7.4	Programmstrukturen .....	187
7.5	Modulare Programmierung .....	188
7.5.1	Modulkompetenzen .....	189
7.5.2	Variablen .....	191
7.5.3	Flags .....	192
7.6	Ereignisse erfassen .....	193
7.6.1	Regelmäßiges Abfragen (Pollen) .....	195
7.6.2	Zeitereignisse .....	195
7.6.3	Warten auf Flanken .....	198
7.6.4	Erfassung per Interrupt .....	199
7.7	Organisation von Software Timern .....	201
7.7.1	Software Timer-Beispiel .....	202

7.7.2	Timer-Genauigkeit . . . . .	205
7.8	Hauptschleifen . . . . .	206
7.8.1	Asynchrone Schleife . . . . .	207
7.8.2	Getriggerte Hauptschleifen . . . . .	208
7.8.3	Getaktete Hauptschleifen . . . . .	208
<b>8</b>	<b>Entwicklungssystem MPLAB 7.60 . . . . .</b>	<b>212</b>
8.1	Installation . . . . .	212
8.2	Die ersten Schritte . . . . .	213
8.3	Überblick, die Menüpunkte betreffend . . . . .	217
8.3.1	File: . . . . .	217
8.3.2	Edit: . . . . .	218
8.3.3	View: . . . . .	221
8.3.4	Project: . . . . .	227
8.3.5	Debugger: . . . . .	228
8.3.6	Programmer: . . . . .	230
8.3.7	Tools: . . . . .	231
8.3.8	Configure: . . . . .	232
8.3.9	Window: . . . . .	235
8.3.10	Help: . . . . .	236
8.4	Der Assembler MPASM . . . . .	237
8.4.1	TITLE-Anweisung . . . . .	237
8.4.2	IF-Anweisung . . . . .	238
8.4.3	LIST-Anweisung . . . . .	239
8.4.4	INCLUDE-Anweisung . . . . .	239
8.4.5	__CONFIG-Anweisung . . . . .	240
8.4.6	__IDLOCS-Anweisung . . . . .	240
8.4.7	EQU-Anweisung . . . . .	240
8.4.8	CBLOCK-Anweisung . . . . .	240
8.4.9	#DEFINE-Anweisung . . . . .	241
8.4.10	ORG-Anweisung . . . . .	242
8.4.11	BANKSEL-Anweisung, PAGESEL-Anweisung . . . . .	243
8.4.12	FILL-Anweisung . . . . .	243
8.4.13	END-Anweisung . . . . .	244
8.4.14	Herstellen von EEPROM-Daten mit MPASM . . . . .	244
<b>9</b>	<b>Der ICD2, der In-circuit Debugger und Programmer . . . . .</b>	<b>247</b>
9.1	Das ICD2-Interface . . . . .	251
9.1.1	MCLR . . . . .	252
9.1.2	VCC . . . . .	252
9.1.3	GND . . . . .	252
9.1.4	PGC und PGD . . . . .	253

9.2	Der Debugger-Modus . . . . .	253
9.3	Reservierungen und Einschränkungen des Debug-Modus . . . . .	254
9.4	Der Programmier-Modus . . . . .	254
9.5	Bedienung des ICD2 . . . . .	255
9.6	Inbetriebnahme des ICD2 . . . . .	256
9.6.1	Software . . . . .	256
9.6.2	Hardware . . . . .	256
<b>10</b>	<b>Demo Boards und Developer's Kits . . . . .</b>	<b>259</b>
10.1	Grundausrüstung . . . . .	259
10.1.1	Stromversorgung . . . . .	259
10.1.2	Oszillatorteil . . . . .	259
10.1.3	V.24-Treiber . . . . .	261
10.1.4	LED-Reihe . . . . .	261
10.1.5	Tasten . . . . .	262
10.1.6	großes Lochrasterfeld, mit GND- und +5V-Anschlüssen . . . . .	263
10.1.7	ein Potentiometer . . . . .	263
10.2	Überblick . . . . .	263
10.3	Kurze Vorstellung einiger dieser Werkzeuge . . . . .	264
10.3.1	PICDEM1: . . . . .	264
10.3.2	PICDEM2 plus . . . . .	265
10.3.3	PICDEM3: . . . . .	268
10.3.4	PICDEM4: . . . . .	268
10.3.5	PICKIT1 . . . . .	271
10.3.6	PICKit 2 Development Programmer/Debugger . . . . .	273
10.3.7	PICDEM MSC . . . . .	274
10.3.8	PICDEM CAN . . . . .	276
10.4	Praktische Anwendungen . . . . .	278
10.4.1	Das PICDEM 2 plus . . . . .	278
10.4.2	MCP251X CAN Development Kit . . . . .	280
<b>11</b>	<b>Peripheriebausteine . . . . .</b>	<b>282</b>
11.1	Schnittstellenwandler . . . . .	282
11.1.1	CAN . . . . .	282
11.1.2	IRDA . . . . .	285
11.1.3	LIN . . . . .	286
11.1.4	I2C-I/O-Expander . . . . .	287
11.2	Speicherbausteine . . . . .	289
11.2.1	SPI-Bus/Microwire . . . . .	289
11.2.2	I2C-Bus . . . . .	289
11.3	Operationsverstärker und Komparatoren . . . . .	291
11.3.1	MCP654X . . . . .	291

11.3.2	MCP604X	292
11.3.3	MCP6S2X	293
11.3.4	Die Low cost-OPs	294
11.3.5	Linear building blocks, lineare Bausteine	295
11.4	AD-Wandler	296
11.5	DA-Wandler	297
11.6	Digitale Potentiometer	298
11.6.1	Innere Werte	298
11.6.2	Typenvielfalt	299
11.6.3	Verwendungszweck	299
11.7	Temperatursensoren	300
11.8	Reset-Generatoren	303
11.8.1	Innere Werte	304
11.8.2	Gehäusevarianten	304
11.8.3	Alternativer Verwendungszweck	304
11.8.4	Auswahlliste:	305
11.9	LDO-Spannungsregler	305
11.10	MOSFET-Treiberbausteine	307
11.10.1	Low-side-Treiber TC4421	308
11.10.2	High-side-Treiber LTC1154	309
11.11	Uhrenbausteine	310
11.11.1	DS1302	310
11.11.2	DS1307	310
11.12	Netzteil-design	312
	<b>Sachverzeichnis</b>	<b>315</b>

# 10 Demo Boards und Developer's Kits

Die Firma Microchip bietet zu ihren Controllern und Peripheriebausteinen eine Reihe von Demo Boards und Developer's Kits an.

Sinn und Zweck dieser Boards ist es, Sachverhalte zu demonstrieren, die Funktionsweise von Bausteinen zu zeigen und den Entwickler schnell in die Lage zu versetzen, das Erlernete gleich anzuwenden.

Die Demo Boards muß man aber nicht entsorgen, wenn man einmal alles verstanden hat, was sie demonstrieren. Sie lassen sich als intelligente Laborgeräte zum Testen, Experimentieren oder Kommunizieren vielfältig verwenden.

## 10.1 Grundausrüstung

Alle Demo Boards besitzen eine Grundausrüstung an Hardware. Die einzelnen Komponenten möchten wir hier kurz besprechen.

### 10.1.1 Stromversorgung

Der Stromversorgungsteil auf den Demo Boards ist darauf ausgelegt die Versorgung per Netzgerät zu ermöglichen. Bei den Boards PICDEM2, PICDEM2 plus und PICDEM4 kann alternativ noch eine 9 Volt-Blockbatterie angesteckt werden. Das ist außerordentlich nützlich, um mobil zu sein.

### 10.1.2 Oszillatorteil

Der Grundausrüstung des Oszillatorteils beschränkt sich auf die Beschaltung des Hauptoszillators an OSC1 und OSC2. Diese ist aber komplett. Es ist sowohl der RC-Oszillator realisierbar, als auch XT mit einem Quarz und HS, z. B. mit einem Quarzoszillator in einem 14poligen DIL-Gehäuse. Das RC-Glied ist über einen Jumper an OSC1 geschaltet, und der Quarzoszillator ist gesteckt. Nur der Quarz mit seinen Ballastkondensatoren wird gelötet. Will man also wieder einen anderen Oszillatortypen haben, muß man diese drei Bauelemente eben wieder auslöten.

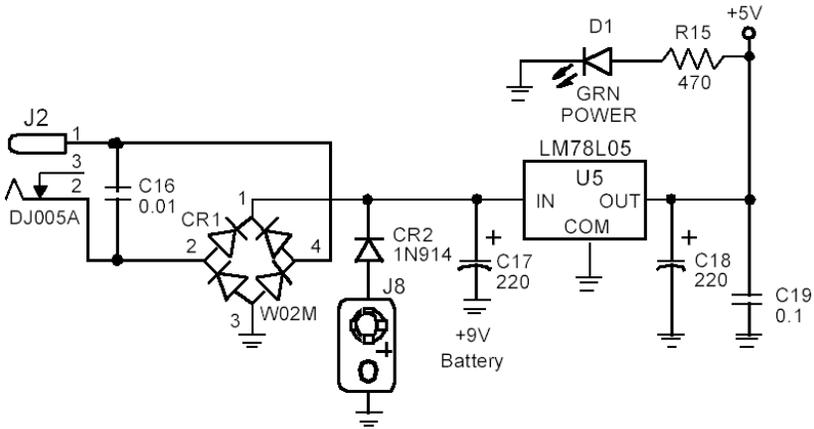


Abbildung 10.1: typische Stromversorgungsschaltung auf den Demo Boards

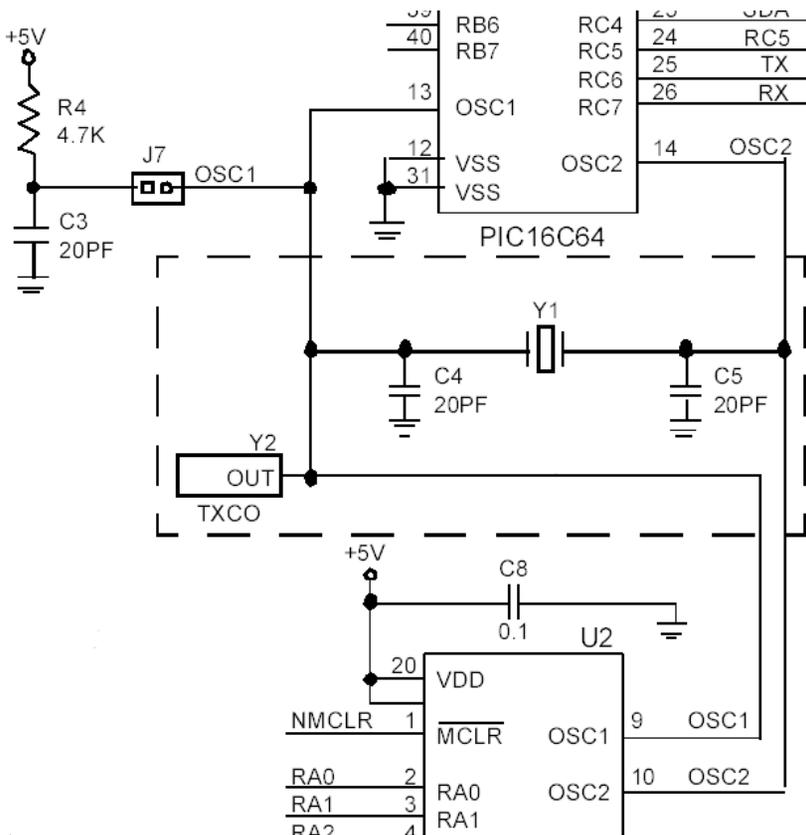


Abbildung 10.2: Die externen Oszillatorschaltungsmöglichkeiten

### 10.1.3 V.24-Treiber

Diese Schnittstelle gehört zu den langjährigen Standards bei den Demo Boards von Microchip. Mit einem normalen 1:1-V.24-Kabel kann damit auf einfache Weise eine Verbindung zum PC hergestellt werden.

Der Treiberbaustein kann dabei ein MAX232A oder ein LT1280A sein, das ist egal. Wichtig ist, daß er vorhanden ist, denn das externe Anstöpseln eines derartigen Treibers ist lästig.

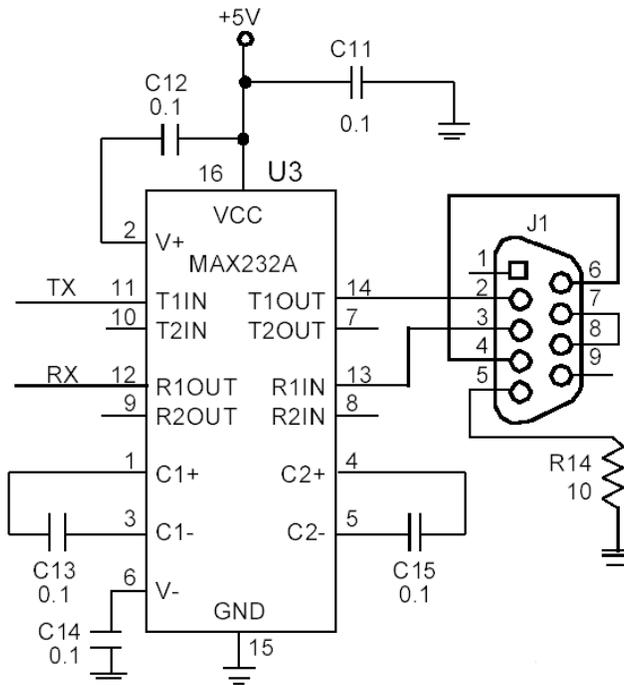


Abbildung 10.3: Die V.24-Treiberschaltung mit der 9poligen DSUB-Buchse

### 10.1.4 LED-Reihe

Zum schnellen Anzeigen eines digitalen Zustands oder eines Bytes konnte man seit frühester Zeit die LED-Reihe verwenden. Den Fortschritt eines Prozesses mit den LEDs verfolgen zu können ist auch sehr praktisch. Aus diesem Grunde sind die LEDs nach wie vor auf den Demo Boards vorhanden. Daß es beim PICDEM2 plus nur 4 Stück sind, wird durch die LCD-Anzeige ausgeglichen.

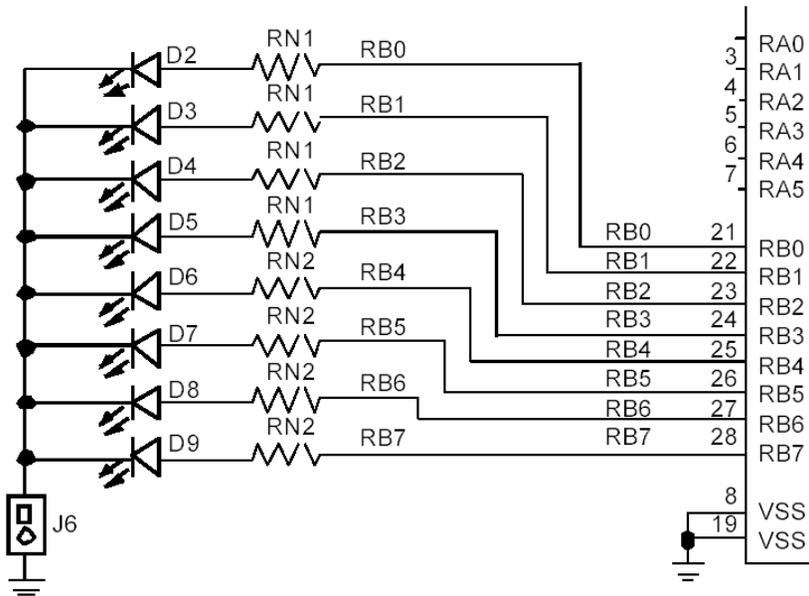


Abbildung 10.4: Die LEDs mit ihren Vorwiderständen

Die Pins, an denen die LEDs angeschlossen sind, sind nicht immer die gleichen, und die Anzahl auch nicht. Ganz früher, bei dem PICDEM1, war die LED-Reihe nicht per Jumper abschaltbar, was jetzt mittlerweile Standard ist.

### 10.1.5 Tasten

Ein paar Tasten auf einem Test-Board sind so wertvoll wie ein kleines Steak. Es gibt immer Funktionen zu starten oder eine Eingabe per Menü. Für diese Aufgabe ist eine Taste von Nöten. Zur Standardausrüstung gehören drei Tasten.

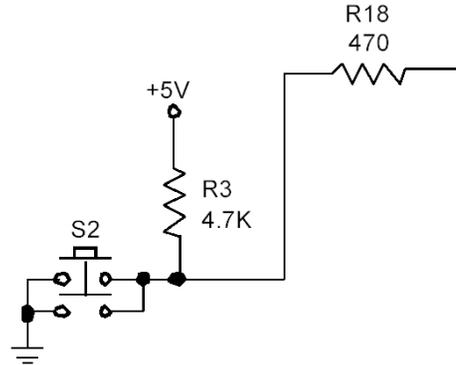
- MCLR
- RTCC, T0CKI
- RA1, RB0, RC2

Die MCLR-Taste ist natürlich nicht direkt einlesbar, aber unbedingt nötig.

Die zweite Taste ist mit dem PortA4, dem Eingang des Timer0 (früher RTCC genannt) verbunden. Damit ist sie vom Controller direkt einlesbar, sie kann aber auch den Timer0 hochzählen.

Die dritte Taste ist nicht immer am gleichen Pin zu finden. Beim PICDEM1 ist sie am PortA1. Sie hat keine spezielle Funktion. Beim PICDEM2 ist diese Taste am

PortC2. Also an einem CCP-Pin, wodurch indirekt auch ein Interrupt ausgelöst werden kann. Beim PICDEM2 plus und beim PICDEM4 ist sie zum PortB0 gewandert. Das ist der INT-Eingang, dessen interruptauslösende Flanke programmiert werden kann.



**Abbildung 10.5:** Eine typische Taste der Demo Boards

### 10.1.6 großes Lochrasterfeld, mit GND- und +5V-Anschlüssen

Ein bißchen Platz für einige bedrahtete Bauelemente ist auf jedem Demo Board von Microchip vorhanden. Das ist gut, denn jeder hat irgendwann mal ein besonderes Bauteil, das er ansprechen möchte oder dessen Signale er erfassen möchte. Die Größe der vorhandenen Lochrasterfelder ist voll in Ordnung. Bei mehreren Bauteilen wird dann ohnehin ein spezieller Prototyp hergestellt.

### 10.1.7 ein Potentiometer

Kein Demo Board ohne Poti. Da es immer auch analoge Signale zu erfassen gibt, haben die meisten PICs einen AD-Wandler onboard. Um die AD-Wandlung zu demonstrieren ist demzufolge auf jedem Demo Board ein Poti. Beim PICDEM4 finden Sie sogar 4 davon. Der Widerstandswert ist typischerweise 10 k $\Omega$ .

## 10.2 Überblick

Die eine Art von Demo Boards ist für Controller mit einem bestimmten Gehäuse entworfen worden. Sie sind großzügig ausgestattet.

Bezeichnung	Verwendungszweck
PICDEM 14A	Für die PIC14000
PICDEM 1	Für 18-, 28- und 40-Pin-Controller der Familien C5X, CE62X, C71, 84, 17C4X
PICDEM 2 plus	Nachfolger von ,2': für 28- und 40-Pin-Gehäuse 16C6X, 16C7X, 16F87X, 18CXX2, 18FXXX
PICDEM 3	Für die LCD-Controller 16C92X
PICDEM 4	Für 8-, 14- und 18-Pinner: 12F629/675, 16F62X, 16F81X, 18F1220, ...
PICKIT 1	Für 8- und 14-Pinner: 12F629/675, 16F639/676
PICDEM 17	Für die PIC17CXX

Eine weitere Kategorie von Demo Boards ist die Reihe derer, die konzipiert sind, um ganz spezielle Eigenschaften zu zeigen.

rfPIC Development Kit	Demonstration einer HF-Verbindung im 315–433-MHz-Bereich. Es sind unterschiedliche Receiver- und Transmitter-Module verfügbar.
PICDEM.net	TCP/IP Demonstrations-Kit
PICDEM LIN	Demo Board für den LIN-Controller 16C43X
PICDEM USB	Demo Board für den USB-Controller 16C7X5
MCP251X CAN	CAN-Entwickler-Board
MCP250XX	CAN-Entwickler-Board
PICDEM MSC	Für die mixed signal Controller 16C781/2; es sind einige Zusatz-Boards verfügbar

Für die dsPIC gibt es eine Reihe weiterer Demo Boards, die wir hier nicht näher ansprechen möchten. Die obige Liste erhebt übrigens keinen Anspruch auf Vollständigkeit.

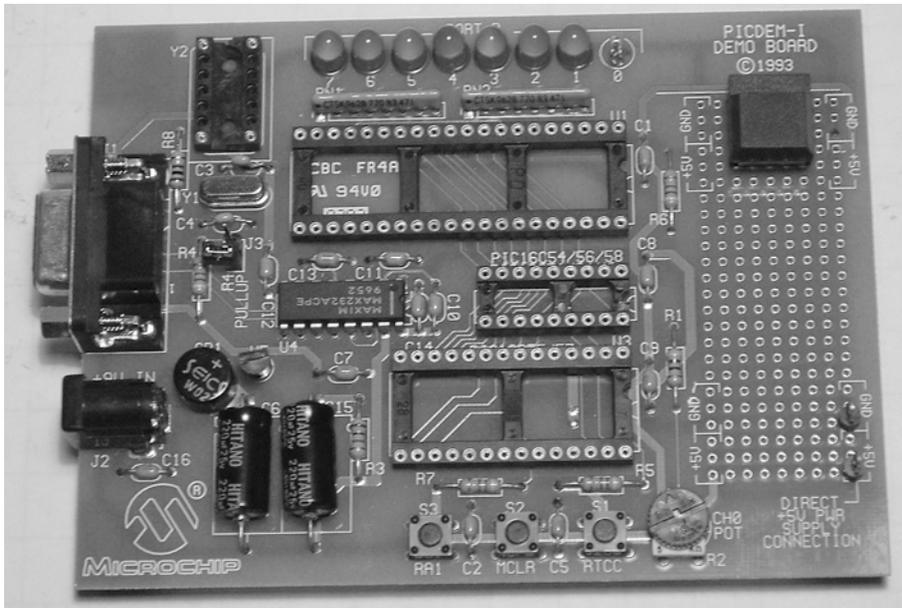
## 10.3 Kurze Vorstellung einiger dieser Werkzeuge

### 10.3.1 PICDEM1:

Der erste Allrounder von Microchips Demo Boards. In dieses Demo Board können alternativ folgende PIC-Typen eingesetzt werden:

- PIC16C5X, 18 Pins
- PIC16C5X, 28 Pins

- PIC17C4X, 40 Pins
- PIC16CXX, 18 Pins, mit Einschränkungen



**Abbildung 10.6:** Das PICDEM1 für die 5X-Familie und die 40 Pin-17er

Außer den oben erwähnten Standardperipherieelementen bietet das PICDEM1 keine weiteren Features. Es ist das einzige 5X-Demo Board.

### 10.3.2 PICDEM2 plus

Ein vielseitiges Demo Board mit Stammbaum. Das heißt, daß es der Nachfolger des PICDEM2 ist. Für folgenden PICs kann das PICDEM2 und 2plus verwendet werden:

- PIC16CXX, 28 und 40 Pins
- PIC16FXXX, 28 und 40 Pins

Über die Standardausrüstung hinaus ist bei diesem Demo Board noch zusätzlich ein Timer1-Oszillator vorhanden. Er ist mit einem Uhrenquarz und zwei Ballastkondensatoren bestückt.

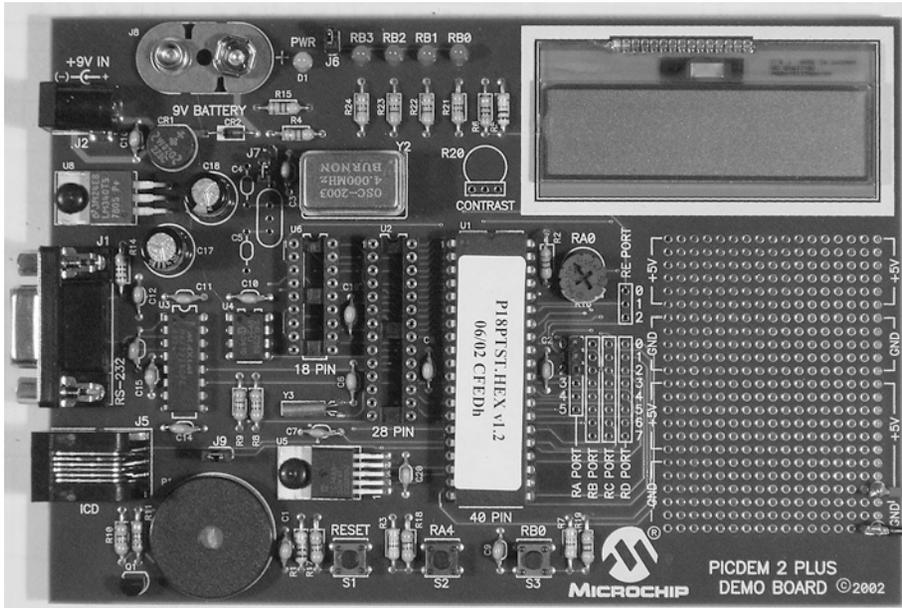


Abbildung 10.7: Das PICDEM2 plus mit LCD-Anzeige

Das PICDEM2 plus bietet des weiteren ein 2 x 16 Charakter großes LCD-Display mit integriertem Controller an, das über einen 4 Bit breiten Datenbus und 3 Steuerleitungen vom Test-PIC angesprochen werden kann.

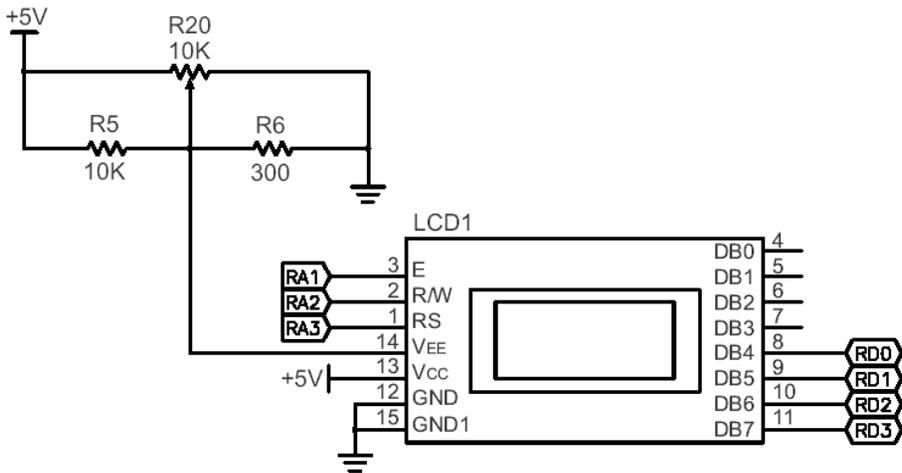
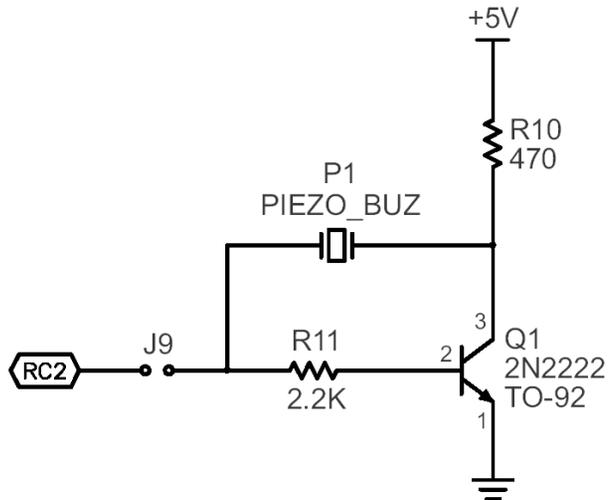


Abbildung 10.8: LCD-Anzeige mit Controller und 4-Bit-Datenbus

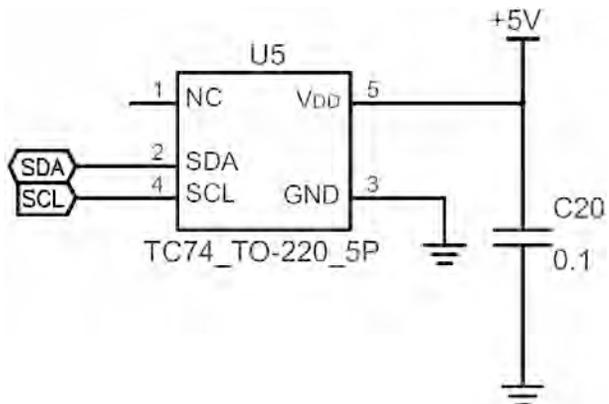
Das einzige Demo Board, das sich akustisch bemerkbar machen kann ist das PICDEM2 plus. Es hat einen Buzzer. Ein Demonstrationsprogramm kann Lautstärke und Tonhöhe in gewissen Grenzen verändern.



**Abbildung 10.9:**  
Der Schreihals des  
PICDEM2 plus

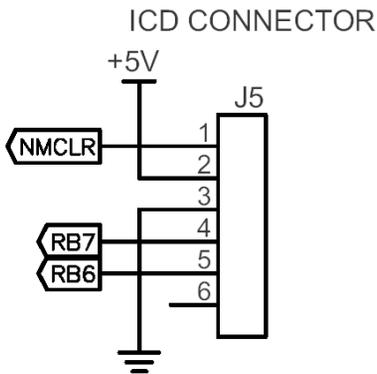
Ein serielles I2C-EEPROM ist auch noch vorhanden. Mit einer Größe von 256 kBit gehört es zu den großen in seiner Klasse. Mit dem PICDEM2 plus können Sie also auch Ihre I2C-Routinen für die EEPROM-Behandlung testen.

Die vorletzte Besonderheit des PICDEM2 plus ist der Temperatursensor TC74. Das eben erwähnte Demonstrationsprogramm ist auch dafür gerüstet, die Temperatur in Grad Celsius anzuzeigen.



**Abbildung 10.10:** Der  
Temperatursensor TC74

Das letzte nützliche Zuckerl ist der ICD2-Steckverbinder. Hierüber kann der ICD2, für Typen die das Debuggen von Hause aus unterstützen, als Debugger angeschlossen werden. Typen, die das Debuggen unterstützen und in diesem Board einen geeigneten Sockel finden sind z. B.: PIC16F876A oder PIC16F877A. Es muß nur ein entsprechender PIC in den Sockel gesteckt werden, die richtige Oszillatorschaltung erstellt werden und das Demo Board muß bestromt werden. Dann kann es schon losgehen.



**Abbildung 10.11:** Die ICD2-Verbindung des PICDEM2 plus

### 10.3.3 PICDEM3:

Ein Spezialist für die 16C92X-Controller mit LCD-Interface. Wie die Bezeichnung Spezialist schon sagt, ist es ausschließlich für diese Typen verwendbar und auch nur zu diesem Zweck.

Über die Standardausrüstung hinaus ist logischerweise ein LC-Display vorhanden.

### 10.3.4 PICDEM4:

Das PICDEM4 ist wie das PICDEM 2 plus ein sehr universelles Demo Board mit vielen Möglichkeiten und für sehr viele PIC-Typen.

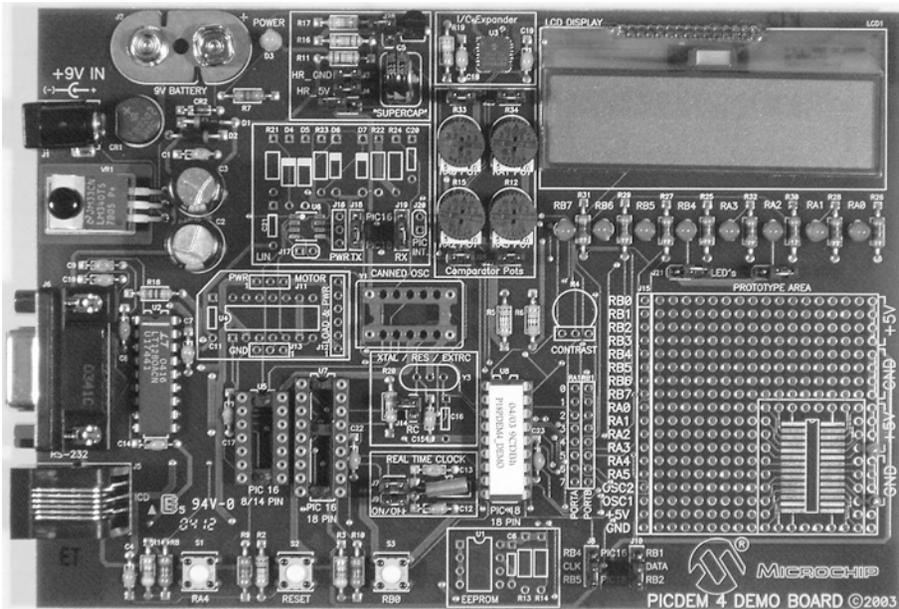


Abbildung 10.12: Das PICDEM4, die Wollmilchsau

Dieses Board bietet für folgende PICs eine tolle Testplattform:

PIC16-Familie:

8 Pin	PIC12F629	PIC12F635	PIC12F675	PIC12F683	
14 Pin	PIC16F630	PIC16F636	PIC16F676	PIC16F684	PIC16F688
18 Pin	PIC16F627A PIC16F628A	PIC16F648A	PIC16F684	PIC16F818 PIC16F819	PIC16F87 PIC16F88

PIC18-Familie:

18 Pin	PIC18F1220	PIC18F1320			
--------	------------	------------	--	--	--

Kleinere PIC18-Derivate als die hier erwähnten Typen gibt es derzeit keine. 8-Pinner waren zwar einmal angekündigt, aber davon hat man wieder Abstand genommen.

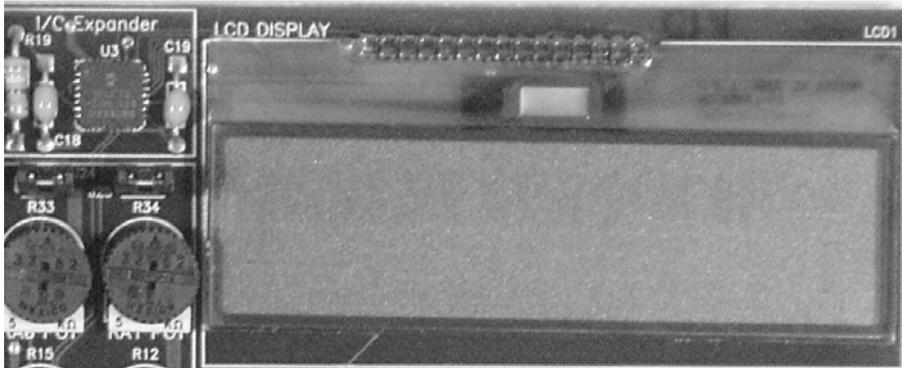
Nun zu den einzelnen Teilschaltungen auf dem Demo Board:

### 2x16 Liquid Crystal Display

Dieses LC-Display mit integriertem Controller kann per SPI-Bus über den I/O-Expander angesprochen werden. Der Befehlssatz des I/O-Expanders PIC16LF72

wird in einer Menge von Programm-Files dargelegt, die alle auf der dem Demo Board beiliegenden CD-ROM enthalten sind.

Der I/O-Expander ist kein Standardbauteil wie der MCP34016. Hierbei handelt es sich um einem fest vorprogrammierten PIC16LF72 mit RC-Oszillator und SPI-Bus-Interface zum Controller.



**Abbildung 10.13:** LCD-Anzeige am I/O-Expander

In-circuit serielle Verbindung zum Programmieren und Entwickeln mit dem ICD2

Je nach PIC-Type kann über diese Verbindung mit dem ICD2 entweder nur programmiert oder aber auch entwickelt (Debug-Modus) werden. Wie im Kapitel über den ICD2 dargelegt, unterstützt nicht jeder PIC von sich aus das Debuggen durch den ICD2. In der obigen Liste der von diesem Demo Board unterstützten PICs sind es nur die 18er und die PIC16F81X und PIC16F8X, die über diese Verbindung debuggen werden können.

Da alle Ziel-PICs ohnehin gesockelt sind, kann problemlos ein entsprechender Header verwendet werden. Dann muß der ICD2 aber auf dem Header angesteckt werden.

8 LEDs, 4 Potentiometer und 3 Taster

Diese einfachen Ein- und Ausgabeelemente scheinen zwar unspektakulär, aber es macht einen deutlichen Aufwand, wenn man sie sich selbst schaffen muß. Gedacht ist es schnell, aber zur Realisierung braucht man dann doch einige Zeit. Also: ‚Danke, daß sie vorhanden sind.‘

Platine ist vorbereitet für ein EEPROM, eine H-Brücke (Motortreiber) und einen LIN-Transceiver

Ein externes serielles EEPROM ist auch so ein Bauelement, welches man immer mal wieder in einer Schaltung braucht. Für diesen Fall ist auf dem Demo Board

ein Bereich reserviert, wo genau ein EEPROM, ein Abblockkondensator für die Stromversorgung und zwei pull-up-Widerstände für die Leitungen SDA und SCL bestückt werden können. Zwei Jumper stellen wahlweise die Verbindungen zum PIC16XXX oder zum PIC18XXX her. Eben solche bestückbaren Bereiche gibt es für einen Motortreiber und den LIN-Treiber MCP201 und deren umgebende Bauteile.

#### Die Oszillatorvielfalt

Die Beschaltung für den Hauptoszillator an OSC1 und OSC2 ist Standard. An dieser Stelle möchte ich allerdings den Uhrenquarz für den Timer1 erwähnen. Er kann bequem per Jumper zugeschaltet werden oder nicht. Er ist bereits mitsamt seinen Ballastkondensatoren bestückt, und das ist gut so. Dieser Quarz ist nicht so problemlos wie der ‚große‘ Quarz. Nach unserer Erfahrung schwingt nicht jeder mit dem PIC. Sie sollten sich unbedingt erst Muster besorgen, bevor Sie diese Teile in hohen Stückzahlen bestellen.

#### Flexible Gestaltung der Stromversorgung

Beim PICDEM4 geht der Stromversorgungsteil über den standardmäßigen Umfang hinaus. Das PICDEM4 hat zusätzlich einen Gold-Cap an Bord. Damit kann man gleich Versuche in einer realitätsnahen Situation durchführen.

- Also mit und ohne externer Stromversorgung.
- Reduzieren der Leistungsaufnahme für die Zeit, in der nur der Gold-Cap für das Überleben sorgt.

Das Umschalten auf Power-managed Modes kann hier sofort und ohne besonderen Aufwand getestet werden.

#### Allgemein verwendbares Lochrasterfeld für Erweiterungen

Dieses Feld gibt es auch hier, aber eine Besonderheit ist doch noch anzumerken. In einem Teil davon sind noch Löt-Pads für ein SO-Gehäuse. Diese Gehäuse drücken immer massiver in den Entwickleralltag herein, daß kaum jemand daran noch vorbei kommt. Aber SO ist gar nicht so schlimm und Widerstände in 1208er Größe sind auch nicht tragisch. Gut, jetzt arbeiten wir grundsätzlich nur noch mit der Pinzette in der Hand. Aber was soll's.

Dieses Board ist rundweg gelungen. Wer sich Geld und Zeit sparen will, der hat das PICDEM2 plus und das PICDEM4 in der Schublade.

### 10.3.5 PICKIT1

Das PICKIT1 ist das erste Demo Board mit einer USB-Anbindung an den PC. Es ist sehr vielseitig und bietet Lehrstoff in vielerlei Hinsicht.

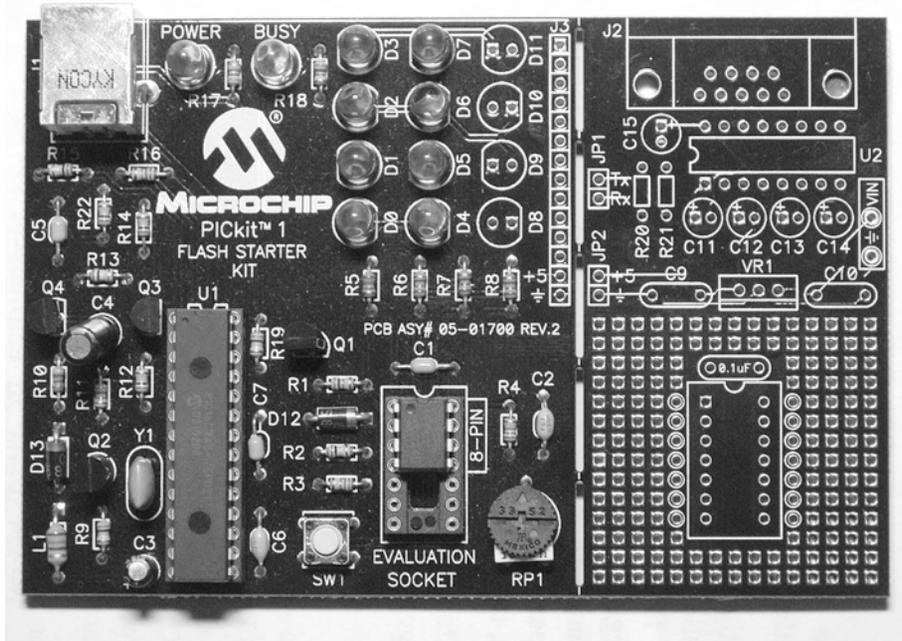


Abbildung 10.14: Das PICKIT1 mit USB-Schnittstelle

Folgende Einsatzgebiete umfaßt das PICKIT1:

#### Demo Board für kleine PICs

Dieses sehr kleine Board bietet nur einen Taster und ein Poti aber bis zu 12 LEDs. Ein Erweiterungsfeld mit einem V.24-Treiber und einem Lochrasterfeld ist zwar nicht bestückt, aber das ist schnell passiert.

Wenn auch hier keine große Hardware vorliegt, dann gibt es doch einige Tutorials, die grundlegende Programmieretechniken vermitteln. So z. B. das Entprellen von Tasten und das Ansprechen von vielen LEDs mit nur wenigen Pins.

#### Hilfsmittel zum Wiederherstellen der Kalibrierwerte für den internen RC-Oszillator

Bei diesem Vorhaben wird eine 2,5 kHz-Referenzfrequenz aus dem USB-Controller als Vergleich herangezogen. Der interne RC-Oszillator sollte dann wieder mit einer Genauigkeit von 1 % laufen.

## USB-Demo

Der dritte Zweck dieses Kits ist wohl die Demonstration einer USB-Verbindung mit dem PC. Dazu werden alle Programme mit ihren Quell-Files dargelegt.

Mit Hilfe des zur Verfügung gestellten GUIs (graphic user interface) können die PICs programmiert werden. Es wird ebenfalls für die Restauration des Kalibrierwertes benötigt.

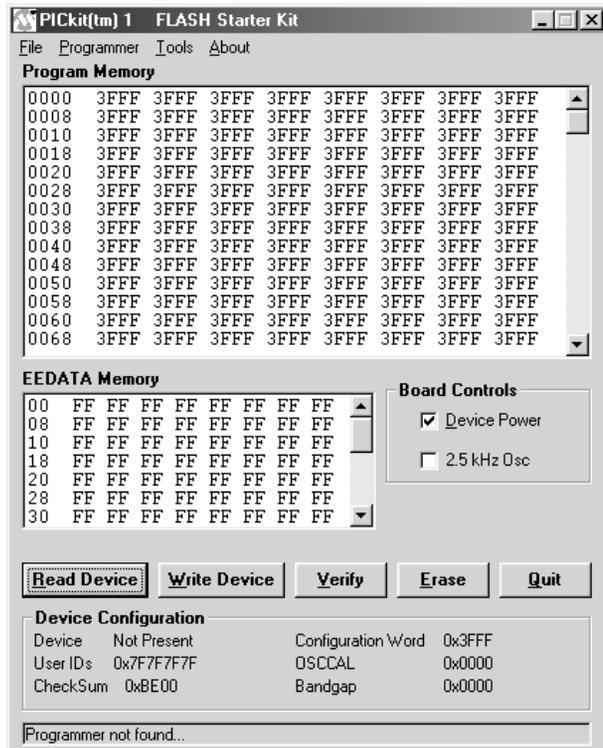


Abbildung 10.15: Das GUI des PICKIT1

### 10.3.6 PICKIT 2 Development Programmer/Debugger

Das PICKIT 2 ist ein kleiner, kostengünstiger ICD2. Also ein ICSP-Programmer und -Debugger. Es ist voll ins MPLAB 7.XX integriert. Zusammen mit dem PICKIT 2 Programmer Application v2.30 werden viele baseline, midrange und PIC18F, PIC24, dsPIC30, dsPIC33 Familien auf Flash-Basis unterstützt. Innerhalb des MPLAB 7.60a ist es nur eine bescheidene Auswahl von unterstützten Bausteinen:

PIC16F87, 88

PIC16F883, 884, 886, 887

PIC16F913, 914, 916, 917

PIC16F946

Folgende Typen können ohne weiteres programmiert werden, jedoch nur mit ICD-Header im Debug-Modus arbeiten:

PIC12F683

PIC16F684, 685, 687

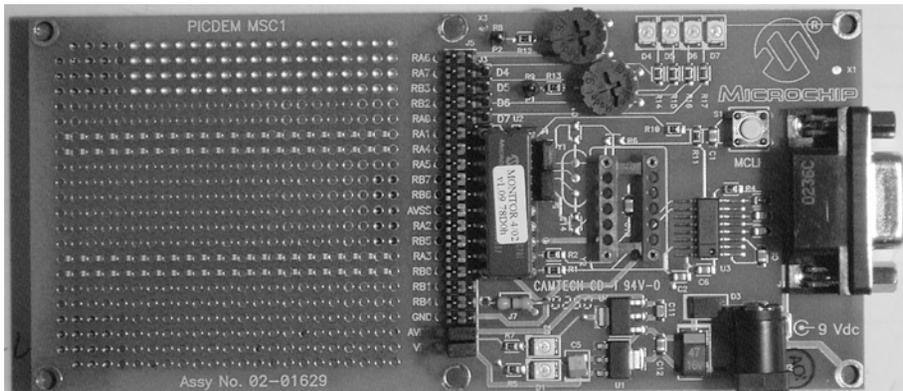
PIC16F688, 689, 690



Dieses Kit gibt es mit der kompletten Dokumentation: Beschreibungen, PC-Software, PIC-Firmware als Quellcode und Schaltplan. Damit soll der Benutzer in die Lage versetzt werden, sein eigenes USB-Gerät zu realisieren.

### 10.3.7 PICDEM MSC

Dieses Demo Board ist speziell für die Controller 16C781/782 entworfen worden. Es soll den Entwickler im Kennenlernen dieses Controllers unterstützen. Dafür wird auch wieder ein GUI zur Verfügung gestellt, mit dessen Hilfe die einzelnen Module konfiguriert werden können.



**Abbildung 10.16:** Das PIC MSC-Demo Board

Welche Zusatzmodule gibt es?

- Timer0-Modul
- Timer1-Modul mit Gate-Eingang
- Referenzspannungsmodul
- Programmierbares Low-Voltage-Detect-Modul (PLVD)
- Analog/Digital-Wandler (8 Bit)
- Digital/Analog-Wandler (8 Bit)
- Operationsverstärker
- Komparatormodul
- Programmable Switch Mode Controller (PSMC)

Für jedes dieser Module gibt es eine Extraseite im GUI, auf der alle Eigenschaften mit den Control-Registern definiert werden können.

Beim D/A-Wandlermodul gibt es sogar noch die Möglichkeit, eine benutzerdefinierte Kurvenform auszugeben.

Mit dem PSMC lässt sich ein PIC-gesteuertes Schaltnetzteil aufbauen.

Also eine komplette Plattform für diese PICs, die zum Messen ,Steuern und Regeln hervorragend geeignet sind.

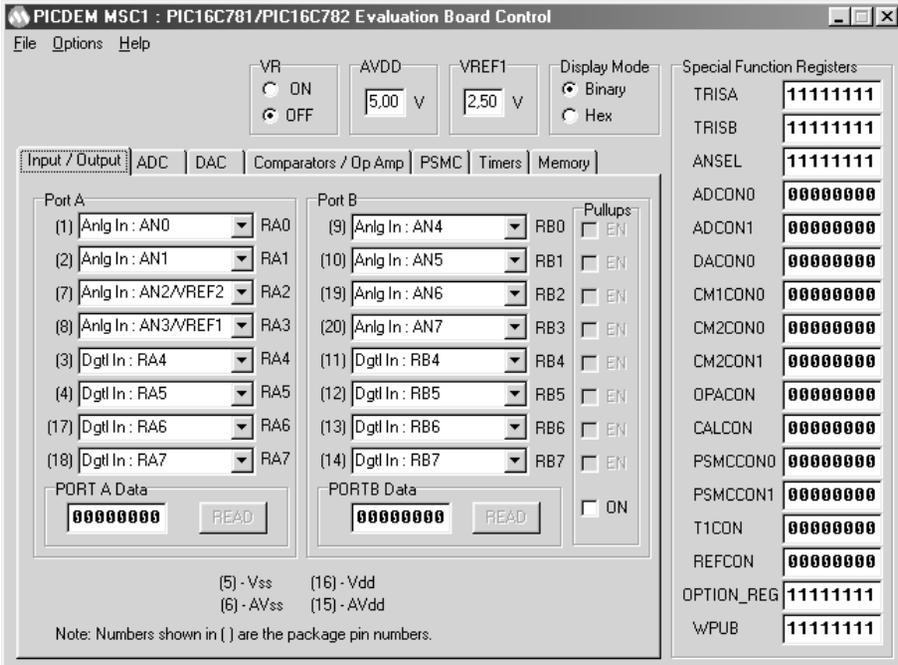


Abbildung 10.17: Das GUI des PIC-MSC-Kits

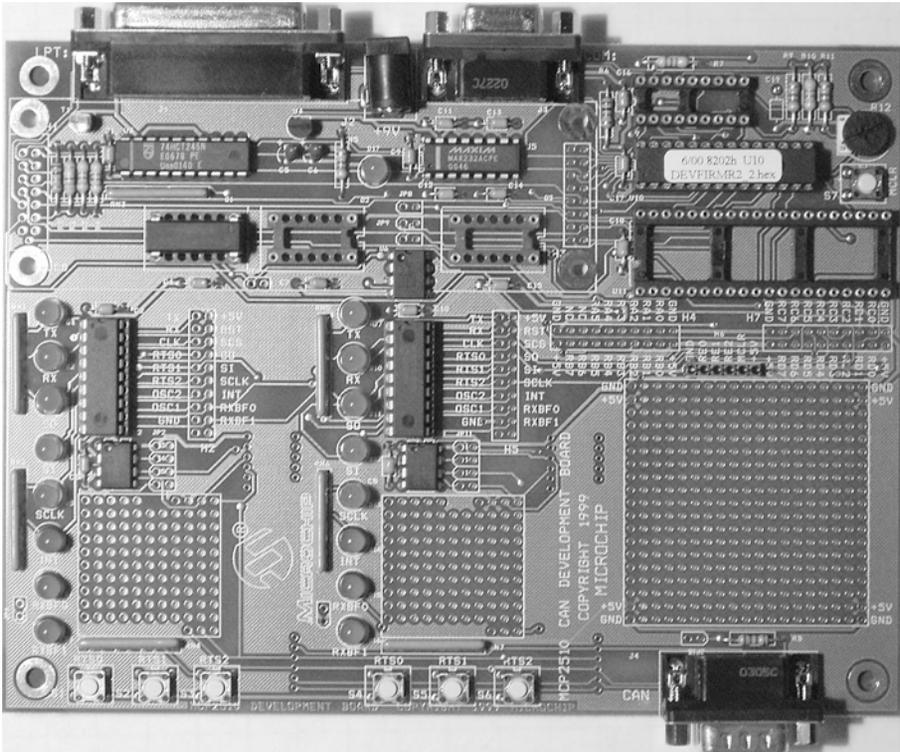
### 10.3.8 PICDEM CAN

Für den CAN-Bus gibt es derzeit mindestens zwei Demo Boards. Das MCP2510 CAN Development Board stellt zwei CAN-Knoten zur Verfügung.

Der linke Knoten ist direkt über die parallele Schnittstelle mit dem PC verbunden. Zu dessen Bedienung wird eine Windows-Software installiert, die es ermöglicht, den MCP2510/2515 komplett zu programmieren. Für die Erstellung der Filter und Masken gibt es eine spezielle Unterstützung.

Der rechte Knoten ist mit einem vorprogrammierten PIC verbunden. Dieser Knoten kann einen analogen Wert absetzen. Drei Tasten und drei LEDs sind auch noch da für die Kommunikation mit dem PC-Knoten. Ein Zeitparameter im PIC läßt sich sogar über den CAN-Bus vom PC aus setzen.

Ein 9poliger DSUB-Stecker stellt den CAN-Bus zur Verfügung. Damit läßt sich ein selbst erstellter CAN-Knoten an den PC-Knoten anschließen und testen. In diesem Zusammenhang war uns dieses Demo Board eine riesige Hilfe.



**Abbildung 10.18:** Das MCP2510 Demo Board

Das MCP250XX CAN Development Board stellt drei CAN-Knoten und einen Programmiersockel für die MCP250XX zur Verfügung.

Ein PC-Knoten per paralleler Schnittstelle wie gehabt und zwei Knoten mit den Standalone-CAN-Controllern MCP250XX. Der rechte dieser beiden Knoten ist zur Demonstration seiner Fähigkeiten mit Tasten, LEDs, Potis und einem Buzzer ausgestattet. Der linke Knoten hat ein Lochrasterfeld zu Füßen und bietet damit für den Entwickler die optimale Freiheit, seine Ideen zu realisieren.

Auch dieses Board hat wieder einen 9poligen DSUB-Stecker, wodurch auch hier problemlos eine eigene Schaltung angeschlossen werden kann.

Rechts neben dem PC-Knoten ist noch ein Programmiersockel. Daneben sorgt ein Schaltregler für die nötige Programmierspannung.

Die MCP250XX werden in diesem Sockel zum realen Einsatz vorbereitet. Sie müssen ja wissen, wie oft sie welche Informationen mit welchen Frame ID's über

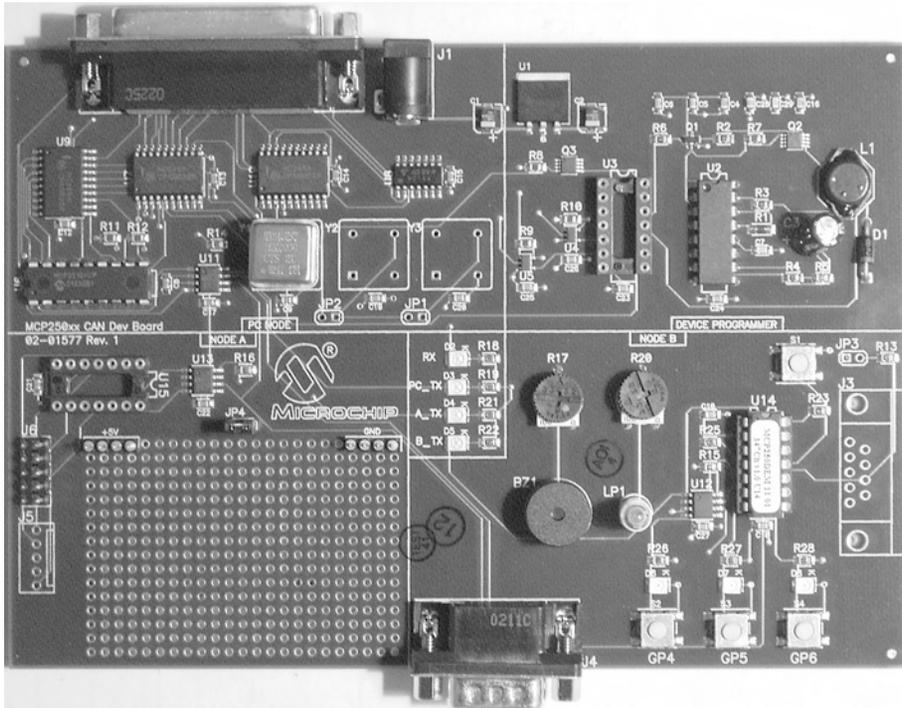


Abbildung 10.19: Das MPC250XX Demo Board

den Bus schicken sollen. Um die richtigen Informationen empfangen zu können, müssen auch noch die Filter und Masken des Eingangspfades vorprogrammiert werden.

## 10.4 Praktische Anwendungen

Um die Nützlichkeit dieser Kits darzulegen, möchten wir zwei Beispiele bringen:

### 10.4.1 Das PICDEM 2 plus

Mit diesem Board haben wir schon so manche Tests schnell mal realisiert. Es beinhaltet wie alle anderen Boards auch eine komplette Stromversorgung aller darauf befindlichen Komponenten. Es ist nur eine 9-Volt-Batterie oder ein 9-V-Netzteil nötig, um es in Betrieb zu nehmen. Wir haben das Netzteil des PICSTART plus verwendet.

Es sind voll funktionsfähige Controller-Schaltungen, die mit Hilfe von mitgelieferten Controllern sofort in Betrieb genommen werden können. Bei diesem konkreten Board sind es zwei Controller. Ein PIC16 und ein PIC18. Das Programm in diesen vorprogrammierten Controllern besteht aus vier Komponenten.

- Voltmeter
- Buzzer
- Thermometer
- Uhr

Die Bedienung erfolgt über die Tasten RA4 und RB0.

Im Voltmeter-Modus wird die Spannung am Poti R16 permanent angezeigt. Der Wert bewegt sich zwischen 0 und 5 Volt. Der PortRA0, an dem das Poti hängt, kann leicht zweckentfremdet werden und zum Messen einer anderen Spannung zwischen 0 und 5 Volt verwendet werden.

Im Buzzer-Modus wird mit Hilfe des PWM-Moduls eine Frequenz an einen Piezosummer ausgegeben. Frequenz und Puls-Pause-Verhältnis sind per Tasten veränderbar.

Im Thermometer-Modus wird der auf dem Board befindliche Temperatursensor TC74 ausgelesen und zyklisch dargestellt.

Im Uhren-Modus wird eine Echtzeit gestartet. Basis dessen ist der Timer1 mit einem 32 kHz-Quarz. Per Tasten ist die Uhr einstellbar.

Wie viele andere Demo Boards auch, hat das PICDEM 2 plus ebenfalls einen V.24-Treiber, den MAX232 o.ä.

Die Daten, die in den eben besprochenen Modi an die LCD-Anzeige ausgegeben werden, gehen gleichzeitig auch über die V.24 an die 9-polige Steckverbindung. Mit einem Terminal-Programm sollte es kein Problem sein, die jeweilige Menüausgabe auch auf einem Computerbildschirm darzustellen.

Sie können mit diesem Board natürlich auch selbst entwickelte Programme testen. Dazu brauchen Sie nur den ICD2, den Sie über die entsprechende Steckverbindung mit dem Demo Board verbinden. Dann können Sie den eingesteckten Controller löschen, programmieren und debuggen. Das Ganze natürlich komfortabel von der Entwicklungsumgebung MPLAB IDE aus. Sie müssen nur für die Spannungsversorgung und die richtige Clock-Beschaltung sorgen.

Diese Demo Boards werden in unserem Hause gerne und oft als selbst konfigurierte Testeinrichtungen und schnelle Experimentier-Boards genutzt.

### 10.4.2 MCP251X CAN Development Kit

Mit diesem Board war es uns problemlos möglich ein selbst entwickeltes CAN-Gerät anzusprechen. Problemlos heißt nicht mühelos! Aber das liegt nicht an diesem Kit oder dem CAN-Baustein, sondern an der CAN-Thematik an sich. Die mitgelieferte Software unterstützte uns sehr gut. Das Ergebnis der Einarbeitung war, daß CAN gar nicht so schwierig ist wie anfänglich vermutet, im Gegenteil.

Wenn man nur die ganzen Filter und Masken nimmt, ist man als Anfänger leicht zu verwirren. Von den Timing-Einstellungen brauchen wir gar nicht erst zu reden. Aber das macht auch die mitgelieferte Software.

Was ist denn nun auf den CAN-Bus abgelaufen? Kein Problem. In einem separaten Fenster, der mitgelieferten PC-Software, sehen Sie, welche Frames über den Bus gegangen sind, und welche Daten sie beinhalteten.

Die Hardware-Installation dieses Entwickler-Kits ist auch kein Problem. Sie brauchen nur einen Parallel-Port.

Nach der Software-Installation müssen Sie sich durch etwas Literatur kämpfen, aber dann bleiben keine Wünsche offen. Nicht nur die dem Demo Board beiliegende Literatur steht zur Verfügung. Von Microchip gibt es einige Applikationsnoten, (z. B. AN733) und die Diplomarbeit von Herrn Koschinsky. Darin enthalten ist übrigens auch Software, die Sie leicht modifizieren können und schon ist Ihr eigener CAN-Knoten programmiert.

An etwas Literatur kommen Sie wie erwähnt nicht herum, weil wie bei jedem Standard-Bus drei große Fragen im Raum stehen:

1. Was muß ich?
2. Was darf ich?
3. Was kann ich mir schenken?

Übrigens: Wenn die Gestaltung der Befehle, respektive ‚Frames‘ einem selbst überlassen ist, dann hat man es wesentlich leichter mit diesen Fragen. Wenn Sie CAN-open verwenden müssen, dann wird's heftig. Hier stellen sich die Fragen dann etwas anders:

1. Welche Befehle brauche ich?
2. Welche Befehle kann ich mitnehmen? Welche kann ich vergessen?
3. Welche Befehle muß ich unterstützen, obwohl ich ihren Sinn auf Anhieb nicht verstehe oder akzeptiere?

Hinter dieser Aussage steckt eine leidvolle SCPI-Erfahrung.

# Sachverzeichnis

2x16 Liquid Crystal Display . . . . . 261, 269

## A

Access-Bank . . . . . 118, 123  
 acquisition time . . . . . 44, 48  
 Addressable USART . . . . . 65, 70  
 Adreßräume . . . . . 93, 98  
 AD-Wandler . . . . . 43, 47  
 Anwendungs-Timer. . . . . 196, 201  
 Architektur . . . . . 16, 20, 92, 97  
 Assembler-Arithmetik. . . . . 165, 170  
 Assembler-Kompatibilität . . . . . 130, 135  
 Asynchrone Schleife . . . . . 202, 207  
 AUSART . . . . . 65, 70  
 Autodecrement . . . . . 122, 127  
 Autoincrement. . . . . 122, 127

## B

Bank-Select-Register. . . . . 118, 123  
 Basis-Timer . . . . . 196, 201  
 Befehlsbreite . . . . . 93, 98  
 Befehlssatz . . . . . 20, 24  
 Befehlsstruktur . . . . . 18, 22  
 Botschafts-Flag . . . . . 184, 189  
 Botschaftsvariablen. . . . . 186, 191  
 Brown-out RESET . . . . . 41, 45  
 Bruchrechnung . . . . . 168, 173  
 BSR . . . . . 118, 123  
 BTG. . . . . 120, 125

## C

CAN-Bus . . . . . 70, 75  
 Capture-Modus . . . . . 47, 52

CCP/ECCP-MODUL .....	47, 52
color-coding .....	212, 219
Compare-Modus .....	48, 53
context saving .....	126, 131
Cosinus .....	175, 180
<b>D</b>	
daisy-chain .....	289, 298
Datenspeicher .....	21, 25, 98, 103
deadband delay .....	50, 56
„debug“-Option .....	245, 246, 253, 254
DECF .....	133, 138
Demo Boards .....	251, 259
Divisionsverfahren .....	169, 174
dominant .....	76, 81
<b>E</b>	
Enhanced PWM .....	50, 55
Enhanced USART .....	68, 73
Ereignisse .....	188, 193
EUSART .....	68, 73, 77, 82
Exponentialformaten .....	170, 175
<b>F</b>	
Fail save Clock Monitor .....	148, 153
Fast Register Stack .....	101, 106
File Select Register .....	23, 27
find in project files .....	220
find-in-files .....	217, 222
FSR .....	23, 27
<b>G</b>	
Getaktete Hauptschleifen .....	203, 208
Getriggerte Hauptschleifen .....	203, 208
Gold-Cap .....	263, 271, 302, 310
<b>H</b>	
Hardware-Kompatibilität .....	130, 135
HF-Sender .....	161, 166
High-side-Treiber LTC1154 .....	309

High-side-Treiber MIC5016 .....	300
hohe Priorität .....	113, 118

**I**

I2C .....	56, 61
I2C-I/O-Expander .....	278, 287
ICD2 .....	239, 247
ICD2-Header .....	241, 249
IDLE .....	142, 147
INCF .....	133, 138
Indirekte Daten-Adressierung .....	23, 27
internen RC-Oszillatorblock .....	142, 147
Interrupt-Routine .....	31, 35
Interrupts .....	29, 33
INTRC .....	142, 147
I/O Ports .....	26, 30, 102, 107

**K**

Komparator .....	45, 49, 282, 291
Konfiguration .....	51, 56

**L**

Latency .....	112, 117
LAT-Register .....	102, 107
LFSR .....	125, 130
LIN-Bus .....	75, 80
LIN-Pegelwandler MCP201 .....	277, 286
Logarithmus .....	175, 180
Low-side-Treiber TC4421 .....	299, 308

**M**

Makro .....	176, 181, 186
MCLR-RESET .....	42, 46
MCP 2515 .....	70
MCP 2510 .....	273, 282
MCP 2515 .....	75, 273, 282
Modul .....	184, 189
MOVFF .....	124, 129
MSSP-Modul .....	56, 61
Multiplikation .....	127, 132

**N**

Nanowatt	141, 146
Negative Zahlen	167, 172
niedrige Priorität	113, 118

**O**

Operationsverstärker	283, 292
OSCCON	143, 148
Oszillator	33, 37

**P**

Pegelwandlerbaustein	276, 285
PIC10F	151
PIC16C745/765	84, 89
PIC18	91, 96
PICDEM CAN	267, 276
PICDEM MSC	265, 274
PICDEM1:	256, 264
PICDEM2 plus	257, 265
PICDEM4:	260, 268
PICKIT1	263, 271
PICKit1	86, 91
pipelining	17, 21
PORT-Register	103, 108
postincrement	122, 127
Power Management	141, 146
Power-on RESET	40, 44
preincrement	122, 127
Programmable gain amplifier	284, 293
Programmable Switch Mode Controller (PSMC)	266, 275
Programmspeicher	23, 27, 93, 98
Programmstruktur	182, 187
Project Wizard	209, 214
PWM-Modus	49, 54

**R**

Read/Modify/Write-Befehle	27, 31
Read-modify-write-Problematik	157, 161
Reset	115, 120
RESET	39, 43
Return Stack	104, 99

rezessiv .....	76, 81
rfPIC .....	161, 166
RS232-Schnittstelle .....	54, 59
RUN .....	142, 147

**S**

seriellen EEPROM-Bausteine .....	280, 289
SFR .....	29, 33
single supply .....	283, 292
Sinus .....	175, 180
Skip-Befehle .....	127, 132
SLEEP .....	142, 147
SLEEP-Modus .....	38, 42
Special Function Register .....	29, 33
SPI .....	56, 61
SSP-Modul .....	56, 61
STKPTR .....	100, 105
SWDTEN .....	108, 113

**T**

Temperatursensor TC74 .....	291, 300
Thermometer DS1620 .....	292, 301
TIMEOUT .....	193, 198
Timer0 .....	35, 39, 105, 110
Timer1 .....	36, 40, 108, 113
TIMER2 .....	110, 115
Timer2 .....	37, 41
Timer3 .....	108, 113
TIMER4 .....	110, 115
Timer-Tick .....	195, 200
TMR0 .....	107, 112
TRIS .....	26, 30
TRIS-Register .....	103, 108
Tristate .....	26, 30

**U**

Uhrenquarz .....	263, 271
UHRZEIT .....	198, 203
universal programming module .....	247, 255
USB .....	83, 88
USB-Demo Board .....	84, 89, 265, 273
USB-Schnittstelle .....	264, 272

**V**

V.24-Treiber .....	253, 261
Variablenstruktur .....	186, 191

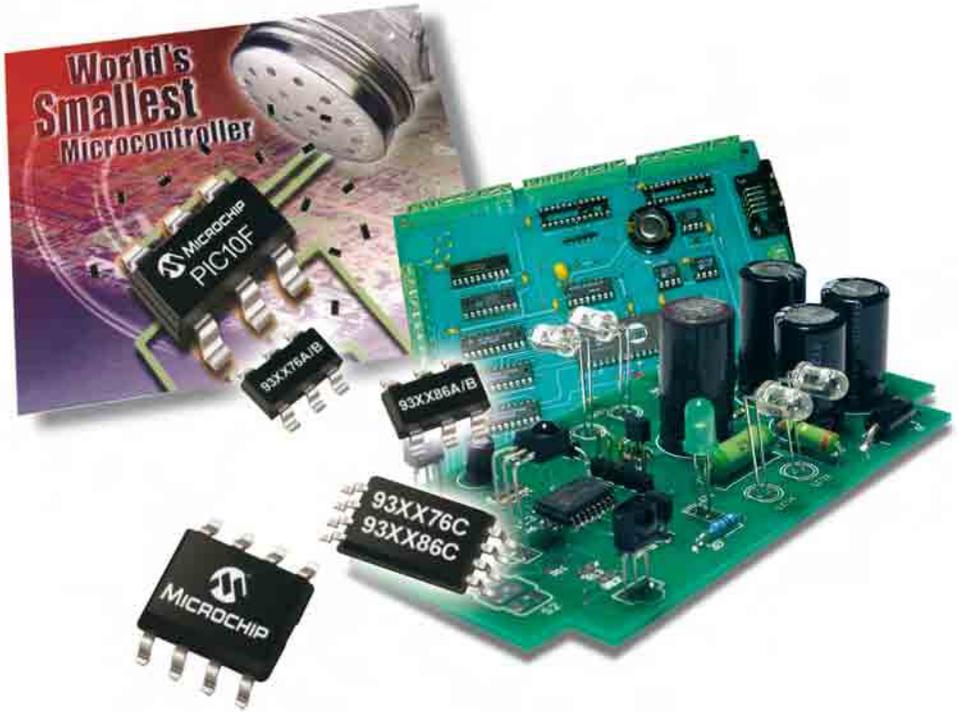
**W**

Wake-up .....	146, 151
Watchdog Timer .....	37, 41, 108, 113
WDTEN .....	108, 113
WDT-RESET .....	41, 45

**Z**

Zeitereignisse .....	190, 195
Zustände .....	188, 193
Zwei-Wort-Befehle .....	123, 128

Anne & Manfred König



Das große

2. überarbeitete Auflage

# PIC-Micro Handbuch



**Auf CD-ROM:**

- MPLAB
- PIC-Programme
- Beispielcode