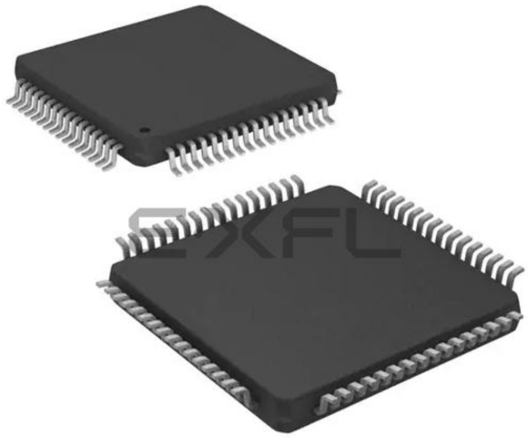
Microchip Technology - PIC16F1946-E/PT Datasheet



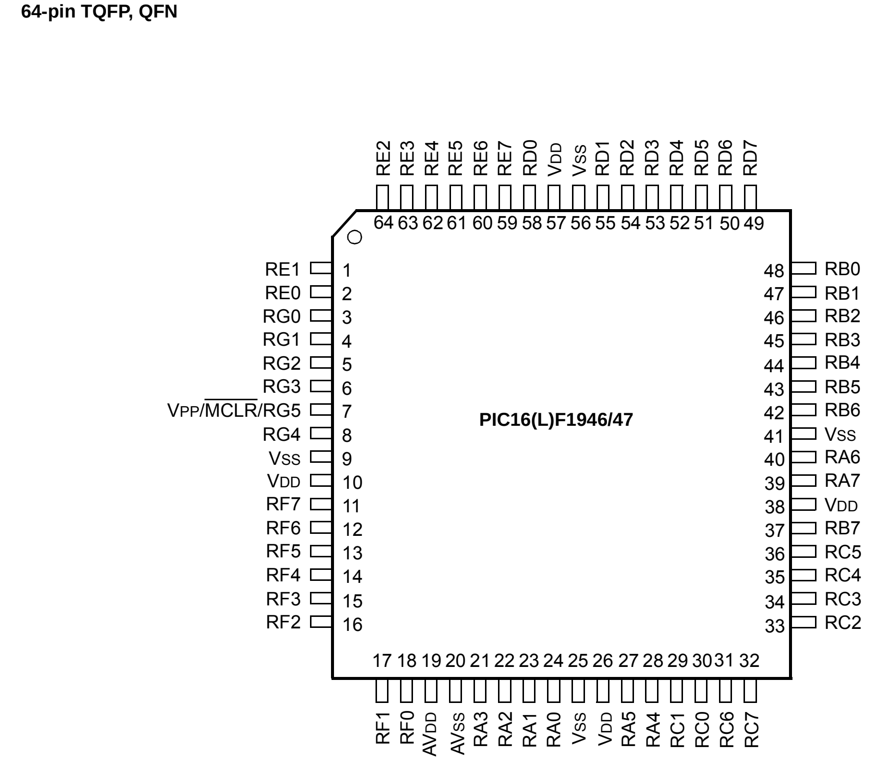
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1946-e-pt

Pin Diagram – 64-Pin TQFP/QFN (PIC16(L)F1946/47)



Note 1: Pin location selected by APFCON register setting. Default location.

2: Pin function can be moved using the APFCON register. Alternate location.

3: QFN package orientation is the same. No leads are present on the QFN package.

Note: AVDD and AVSS are dedicated power connection pins for the on-board analog circuits of the PIC® microcontroller. The separate power pins help eliminate digital switching noise interference with the analog circuitry inside the device, especially on larger devices with more I/O pins and larger switching currents on the VDD/VSS pins. Customers typically connect these to the appropriate VDD or VSS connections on the PCB, unless there is a lot of noise on the external power rails. In those situations, they will add additional noise filtering components (like capacitors) on the AVDD/AVSS pins to help ensure good solid supply to the analog modules inside the device.

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1946

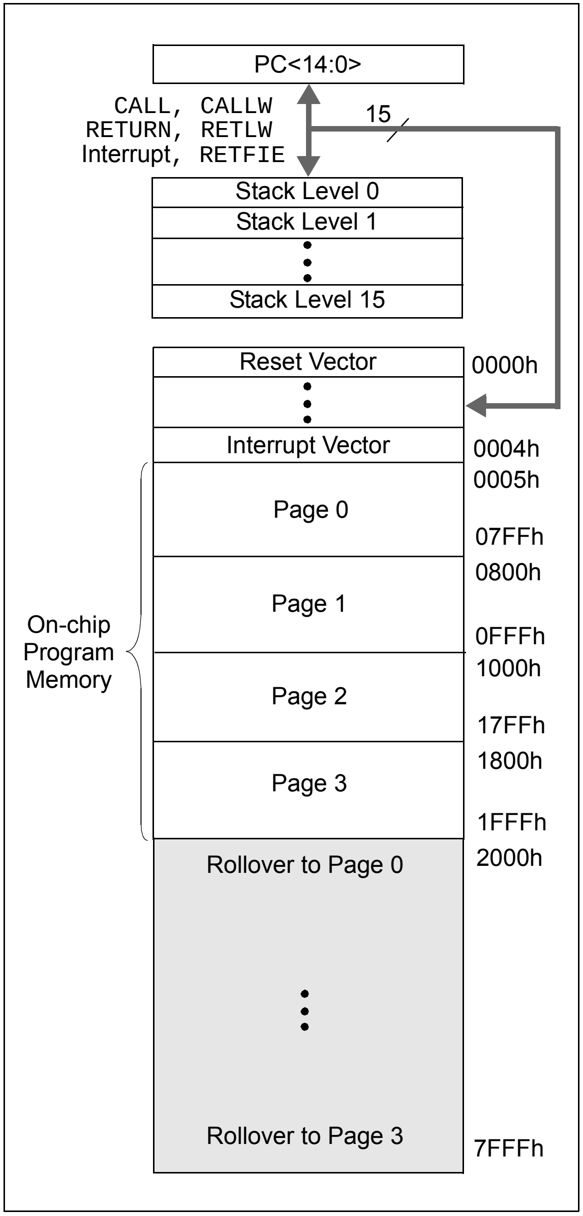


FIGURE 3-2: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1947

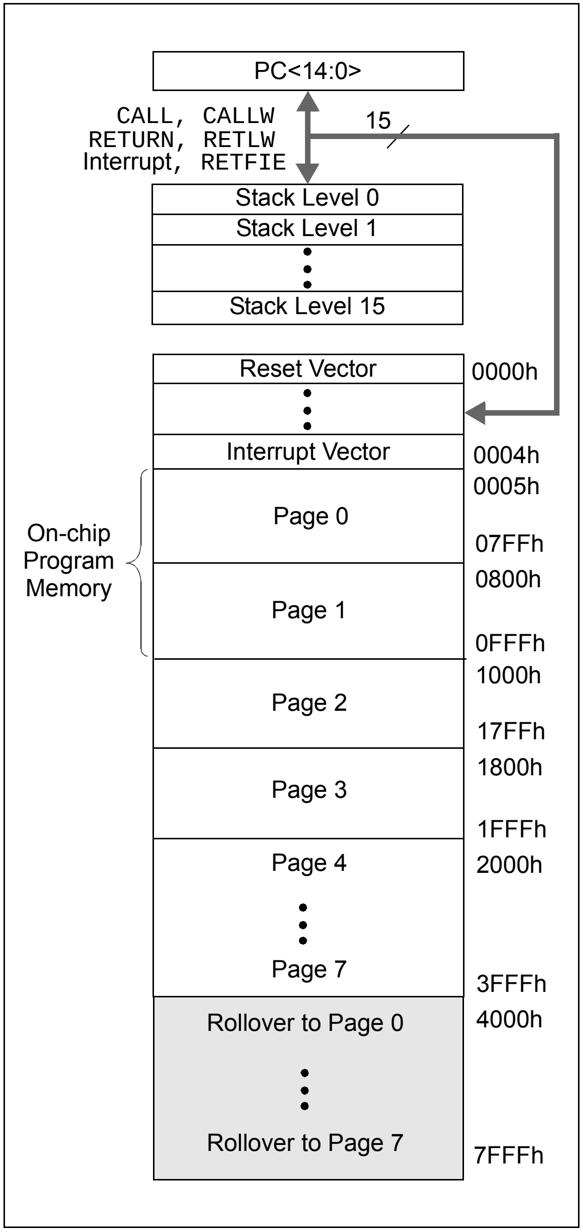


TABLE 3-4: PIC16(L)F1946/47 MEMORY MAP, BANKS 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	—	28Ch	PORTF	30Ch	TRISF	38Ch	LATF
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	—	20Dh	WPUB	28Dh	PORTG	30Dh	TRISG	38Dh	LATG
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	PORTD	08Fh	TRISD	10Fh	LATD	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	LATE	190h	ANSELE	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADRL	211h	SSP1BUF	291h	CCPR1L	311h	CCPR3L	391h	—
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADRH	212h	SSP1ADD	292h	CCPR1H	312h	CCPR3H	392h	—
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	EEDATL	213h	SSP1MSK	293h	CCP1CON	313h	CCP3CON	393h	—
014h	PIR4	094h	PIE4	114h	CM2CON1	194h	EEDATH	214h	SSP1STAT	294h	PWM1CON	314h	PWM3CON	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	EECON1	215h	SSP1CON1	295h	CCP1AS	315h	CCP3AS	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSP1CON2	296h	PSTR1CON	316h	PSTR3CON	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	—	217h	SSP1CON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	CCPR2L	318h	CCPR4L	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RC1REG	219h	SSP2BUF	299h	CCPR2H	319h	CCPR4H	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TX1REG	21Ah	SSP2ADD	29Ah	CCP2CON	31Ah	CCP4CON	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SP1BRGL	21Bh	SSP2MSK	29Bh	PWM2CON	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SP1BRGH	21Ch	SSP2STAT	29Ch	CCP2AS	31Ch	CCPR5L	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RC1STA	21Dh	SSP2CON1	29Dh	PSTR2CON	31Dh	CCPR5H	39Dh	—
01Eh	CPSCON0	09Eh	ADCON1	11Eh	CM3CON0	19Eh	TX1STA	21Eh	SSP2CON2	29Eh	CCPTMRS0	31Eh	CCP5CON	39Eh	—
01Fh	CPSCON1	09Fh	—	11Fh	CM3CON1	19Fh	BAUD1CON	21Fh	SSP2CON3	29Fh	CCPTMRS1	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 16 Bytes	3A0h	General Purpose Register 80 Bytes[1]
												32Fh			
												330h	General Purpose Register 64 Bytes[1]		
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h	Accesses 70h – 7Fh	170h	Accesses 70h – 7Fh	1F0h	Accesses 70h – 7Fh	270h	Accesses 70h – 7Fh	2F0h	Accesses 70h – 7Fh	370h	Accesses 70h – 7Fh	3F0h	Accesses 70h – 7Fh
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

Legend: ▒ = Unimplemented data memory locations, read as ‘0’.

Note 1: Not available on PIC16F1946.

TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 1											
080h(2)	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
081h(2)	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
082h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
083h(2)	STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	---1 1000	---q quuu
084h(2)	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
085h(2)	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
086h(2)	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
087h(2)	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
088h(2)	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000
089h(2)	WREG	Working Register								0000 0000	uuuu uuuu
08Ah(1, 2)	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
08Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 000x	0000 000u
08Ch	TRISA	PORTA Data Direction Register								1111 1111	1111 1111
08Dh	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
08Eh	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
08Fh	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
090h	TRISE	PORTE Data Direction Register								1111 1111	1111 1111
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
092h	PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	C3IE	CCP2IE	0000 0000	0000 0000
093h	PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	-000 0-0-	-000 0-0-
094h	PIE4	—	—	RC2IE	TX2IE	—	—	BCL2IE	SSP2IE	--00 --00	--00 --00
095h	OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	—	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	00-- 11qq	qq-- qquu
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
098h	OSCTUNE	—	—	TUN<5:0>						--00 0000	--00 0000
099h	OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		0011 1-00	0011 1-00
09Ah	OSCSTAT	T1OSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	00q0 0q0-	qqqq qq0-
09Bh	ADRESL	A/D Result Register Low								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	A/D Result Register High								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF1	ADPREF0	0000 -000	0000 -000
09Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Unimplemented, read as '1'.

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM

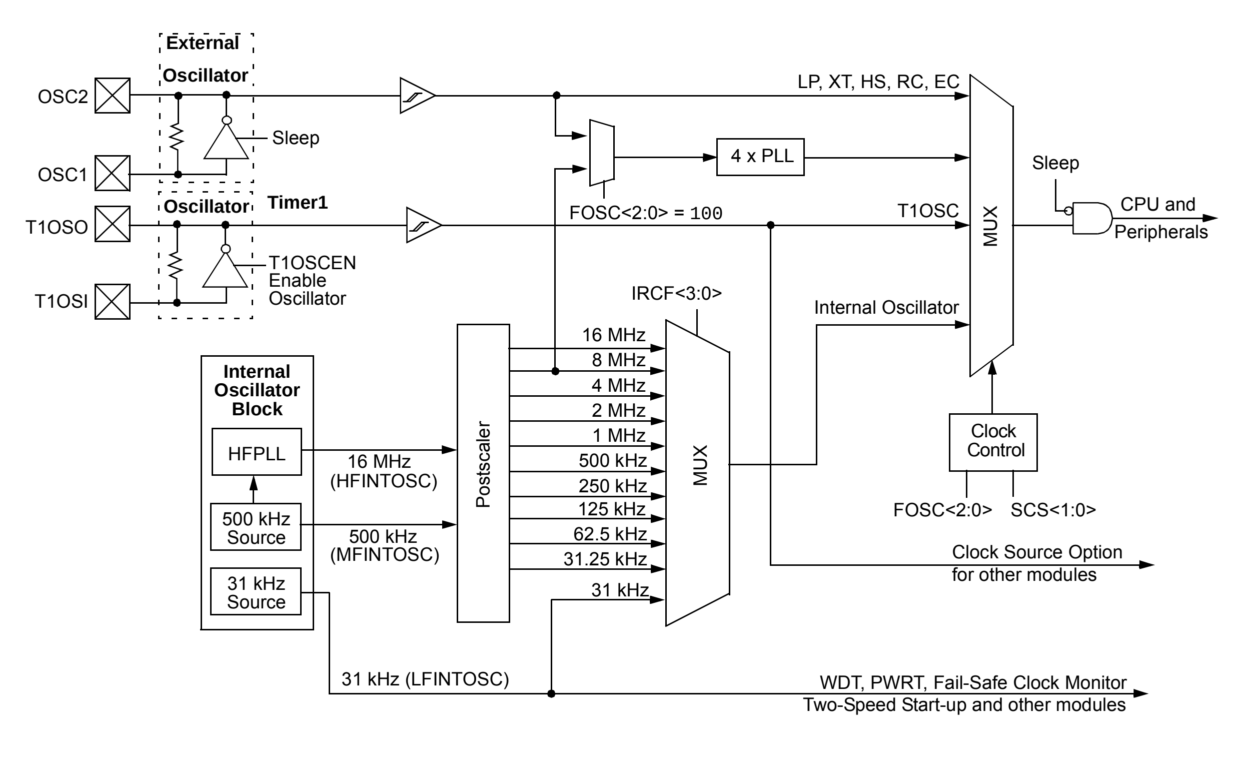
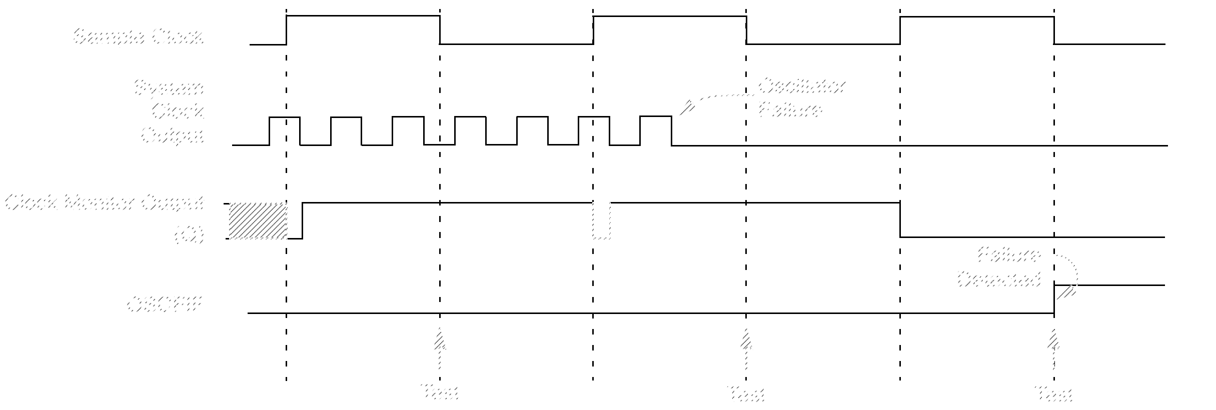


FIGURE 5-10: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

6.12 Power Control (PCON) Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Reset ($\overline{\text{BOR}}$)
- Reset Instruction Reset ($\overline{\text{RI}}$)
- Stack Overflow Reset (STKOVF)
- Stack Underflow Reset (STKUNF)
- $\overline{\text{MCLR}}$ Reset ($\overline{\text{RMCLR}}$)

The PCON register bits are shown in Register 6-2.

6.13 Register Definitions: Power Control

REGISTER 6-2: PCON: POWER CONTROL REGISTER

R/W/HS-0/q	R/W/HS-0/q	U-0	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	—	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-m/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or set to ‘0’ by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or set to ‘0’ by firmware

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to ‘1’ by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (set to ‘0’ in hardware when a $\overline{\text{MCLR}}$ Reset occurs)

bit 2 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to ‘1’ by firmware
0 = A RESET instruction has been executed (set to ‘0’ in hardware upon executing a RESET instruction)

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

11.4 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.
8. Repeat steps 6 and 7 as many times as required to reprogram the erased row.

11.5 User ID, Device ID and Configuration Word Access

Instead of accessing program memory or EEPROM data memory, the User ID’s, Device ID/Revision ID and Configuration Words can be accessed when CFGS = 1 in the EECON1 register. This is the region that would be pointed to by PC<15> = 1, but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 11-3.

When read access is initiated on an address outside the parameters listed in Table 11-3, the EEDATH:EEDATL register pair is cleared.

TABLE 11-2: USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS (CFGS = 1)

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8006h	Device ID/Revision ID	Yes	No
8007h-8008h	Configuration Words 1 and 2	Yes	No

EXAMPLE 11-3: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  EEADRL           ; Select correct Bank
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    EEADRL           ; Store LSB of address
    CLRF     EEADRH           ; Clear MSB of address

    BSF      EECON1,CFGS      ; Select Configuration Space
    BCF      INTCON,GIE       ; Disable interrupts
    BSF      EECON1,RD        ; Initiate read
    NOP                       ; Executed (See Figure 11-1)
    NOP                       ; Ignored (See Figure 11-1)
    BSF      INTCON,GIE       ; Restore interrupts

    MOVF     EEDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     EEDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```

TABLE 12-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	P3CSEL	P3BSEL	P2DSEL	P2CSEL	P2BSEL	CCP2SEL	P1CSEL	P1BSEL	123
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	131
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	330
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	330
LCDSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	330
LCDSE4	SE39	SE38	SE37	SE36	SE35	SE34	SE33	SE32	330
LCDSE5	—	—	SE45	SE44	SE43	SE42	SE41	SE40	330
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	131
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	298
RC2STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	298
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				282
SSP2STAT	SMP	CKE	$D/\overline{A}$	P	S	$R/\overline{W}$	UA	BF	281
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	197
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	297
TX2STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	297
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	131

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTC.

23.3 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully On and fully Off states. The PWM signal resembles a square wave where the high portion of the signal is considered the On state and the low portion of the signal is considered the Off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of On and Off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the On time to the Off time and is expressed in percentages, where 0% is fully Off and 100% is fully On. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 23-3 shows a typical waveform of the PWM signal.

23.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for CCP modules ECCP1, ECCP2, ECCP3, CCP4 and CCP5.

The standard PWM mode generates a Pulse-Width modulation (PWM) signal on the CCPx pin with up to ten bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PRx registers
- TxCON registers
- CCPRxL registers
- CCPxCON registers

Figure 23-4 shows a simplified block diagram of the PWM operation.

Note 1: The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.

2: Clearing the CCPxCON register will relinquish control of the CCPx pin.

FIGURE 23-3: CCP PWM OUTPUT SIGNAL

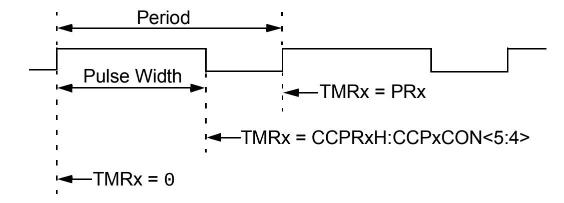
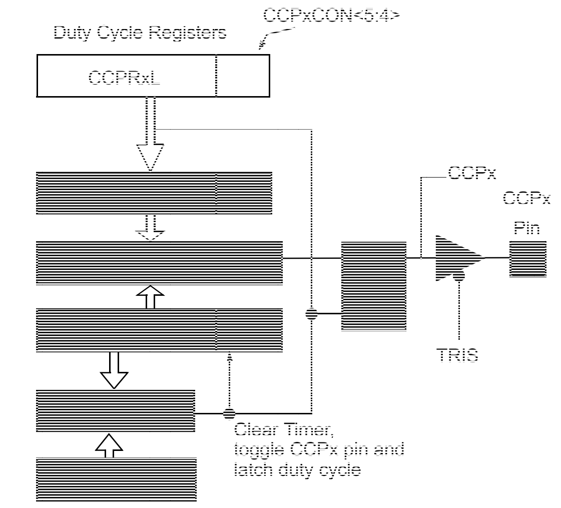


FIGURE 23-4: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: The 8-bit timer TMRx register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.

2: In PWM mode, CCPRxH is a read-only register.

REGISTER 24-3: SSPxCON2: SSPx CONTROL REGISTER 2

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Cleared by hardware S = User set

bit 7 **GCEN:** General Call Enable bit (in I^{2}C Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPxSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I^{2}C mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in I^{2}C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I^{2}C Master mode only)[1]
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDAx and SCLx pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in I^{2}C Master mode only)[1]
1 = Enables Receive mode for I^{2}C
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in I^{2}C Master mode only)[1]
SCKx Release Control:
1 = Initiate Stop condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (in I^{2}C Master mode only)[1]
1 = Initiate Repeated Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enable bit[1]
In Master mode:
1 = Initiate Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

Note 1: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I^{2}C module is not in Idle mode, this bit may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

25.1.2.3 Receive Interrupts

The RCxIF interrupt flag bit of the PIR1/PIR3 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCxIF interrupts are enabled by setting the following bits:

- RCxIE interrupt enable bit of the PIE1/PIE4 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

25.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCxSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCxREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = `1`) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit.

25.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated If a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCxSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

25.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set, the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

25.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCxSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCxIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

TABLE 25-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	299
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	299
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	90
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	91
PIE4	—	—	RC2IE	TX2IE	—	—	BCL2IE	SSP2IE	94
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	95
PIR4	—	—	RC2IF	TX2IF	—	—	BCL2IF	SSP2IF	98
RC1REG	EUSART1 Receive Register								292*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	298
RC2REG	EUSART2 Receive Register								292*
RC2STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	298
SP1BRGL	EUSART1 Baud Rate Generator, Low Byte								300*
SP1BRGH	EUSART1 Baud Rate Generator, High Byte								300*
SP2BRGL	EUSART2 Baud Rate Generator, Low Byte								300*
SP2BRGH	EUSART2 Baud Rate Generator, High Byte								300*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	131
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	292
TX2STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	297

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used for asynchronous reception.
* Page provides register information.

25.4.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RXx signal, the RXx signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDxCON register starts the auto-baud calibration sequence (Figure 25.4.2). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPxBRGL begins counting up using the BRG counter clock as shown in Table 25-6. The fifth rising edge will occur on the RXx/DTx pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPxBRGH:SPxBRGL register pair, the ABDEN bit is automatically cleared, and the RCxIF interrupt flag is set. A read operation on the RCxREG needs to be performed to clear the RCxIF interrupt. RCxREG content should be discarded. When calibrating for modes that do not use the SPxBRGH register the user can verify that the SPxBRGL register did not overflow by checking for 00h in the SPxBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 25-6. During ABD, both the SPxBRGH and SPxBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPxBRGH and SPxBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

Note 1: If the WUE bit is set with the ABDEN bit, the auto-baud detection will occur on the byte following the Break character (see **Section 25.4.3 "Auto-Wake-up on Break"**).

2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.

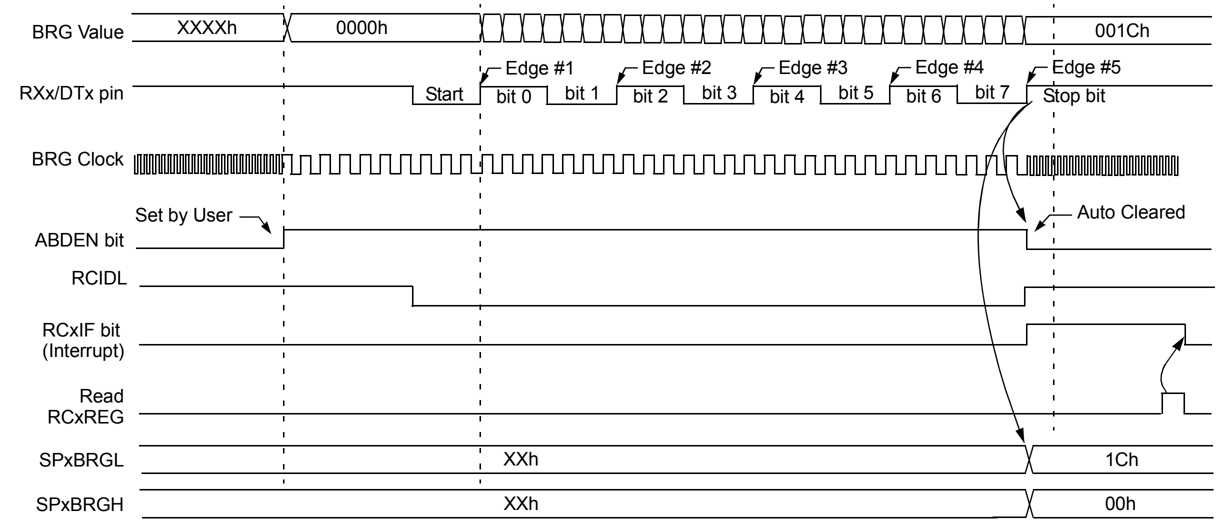
3: During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPxBRGH:SPxBRGL register pair.

TABLE 25-6: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	FOSC/64	FOSC/512
0	1	FOSC/16	FOSC/128
1	0	FOSC/16	FOSC/128
1	1	FOSC/4	FOSC/32

Note: During the ABD sequence, SPxBRGL and SPxBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

FIGURE 25-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.

TABLE 27-9: SUMMARY OF REGISTERS ASSOCIATED WITH LCD OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	90
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		326
LCDCST	—	—	—	—	—	LCDCST<2:0>			329
LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	330
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	330
LCDDATA2	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	330
LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	330
LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	330
LCDDATA5	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	330
LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	330
LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	330
LCDDATA8	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	330
LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	330
LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	330
LCDDATA11	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	330
LCDDATA12	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0	330
LCDDATA13	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0	330
LCDDATA14	—	—	SEG45 COM0	SEG44 COM0	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0	330
LCDDATA15	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1	330
LCDDATA16	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	330
LCDDATA17	—	—	SEG45 COM1	SEG44 COM1	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	330
LCDDATA18	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2	330
LCDDATA19	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2	330
LCDDATA20	—	—	SEG45 COM2	SEG44 COM2	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2	330
LCDDATA21	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3	330

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by the LCD module.

29.2 Instruction Descriptions

ADDFSR	Add Literal to FSRn
Syntax:	[*label*] ADDFSR FSRn, k
Operands:	$-32 \le k \le 31$ $n \in [0, 1]$
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal ‘k’ is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap around.

ADDLW	Add literal and W
Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal ‘k’ and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ADDWFC	ADD W and CARRY bit to f
Syntax:	[*label*] ADDWFC f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location ‘f’. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed in data memory location ‘f’.

ANDLW	AND literal with W
Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND’ed with the 8-bit literal ‘k’. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ASRF	Arithmetic Right Shift
Syntax:	[*label*] ASRF f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

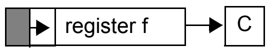


FIGURE 31-21: IDD, MFINTOSC MODE, FOSC = 500 kHz, PIC16LF1946/47 ONLY

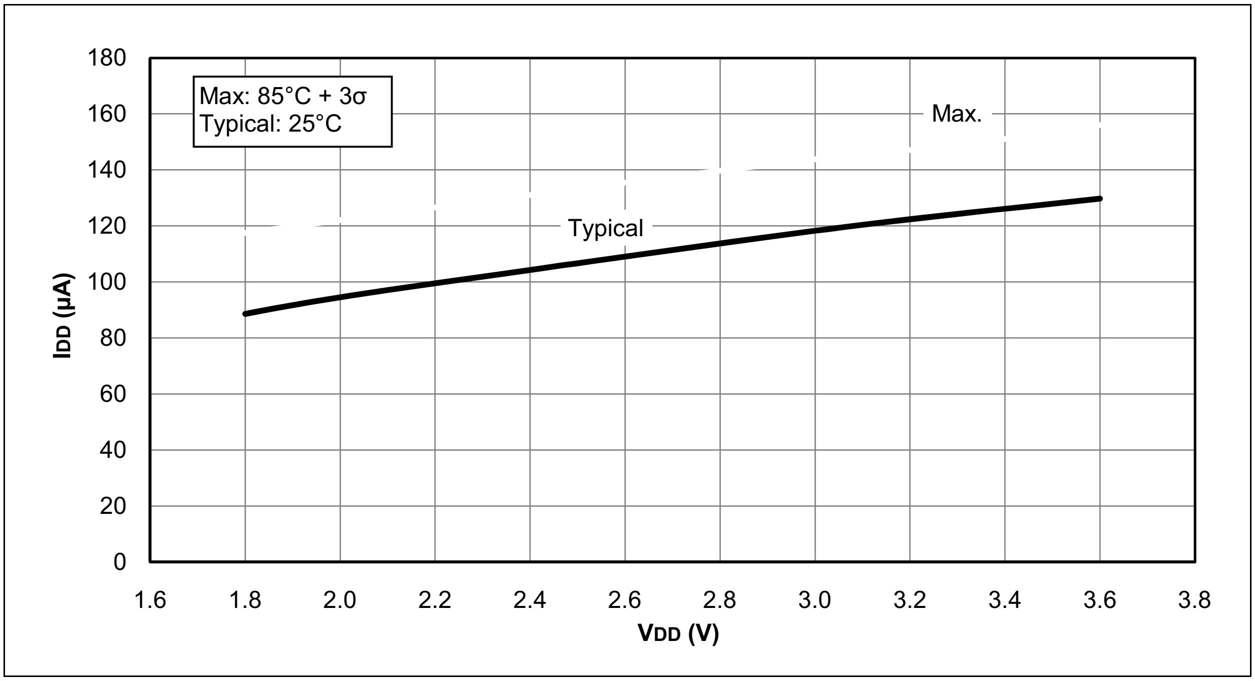


FIGURE 31-22: IDD, MFINTOSC MODE, FOSC = 500 kHz, PIC16F1946/47 ONLY

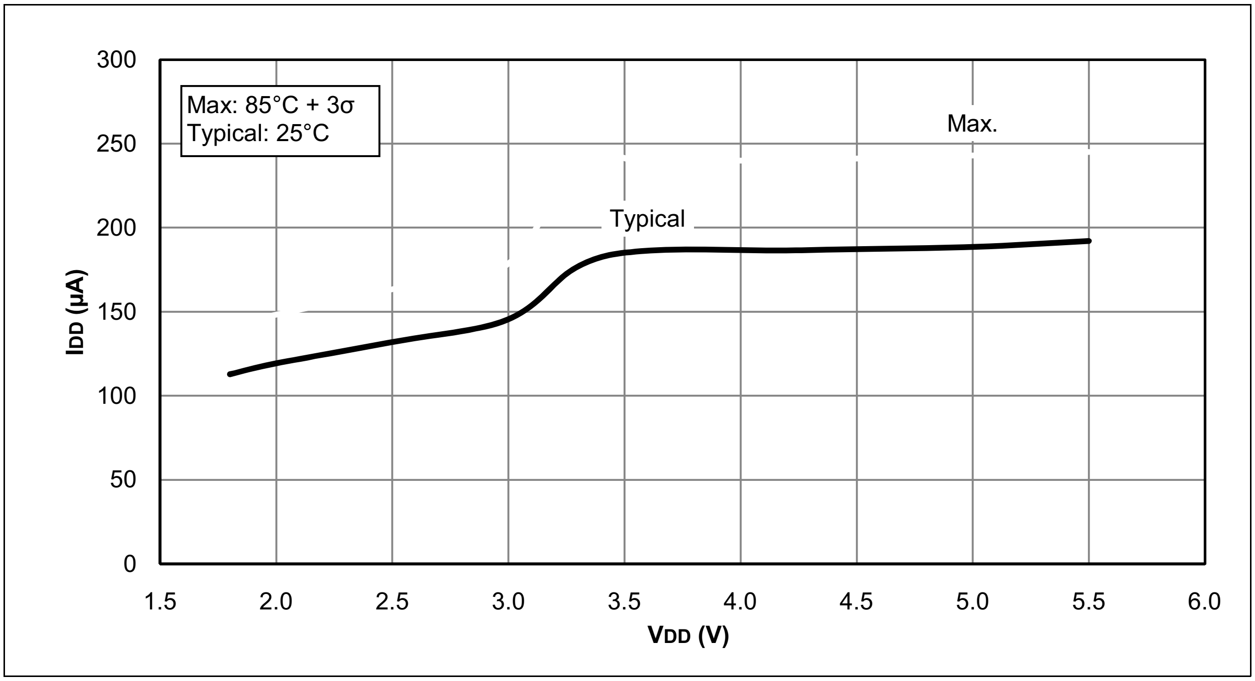


FIGURE 31-43: IPD, CAPACITIVE SENSING (CPS) MODULE, MEDIUM-CURRENT RANGE, CPSRM = 0, PIC16LF1946/47 ONLY

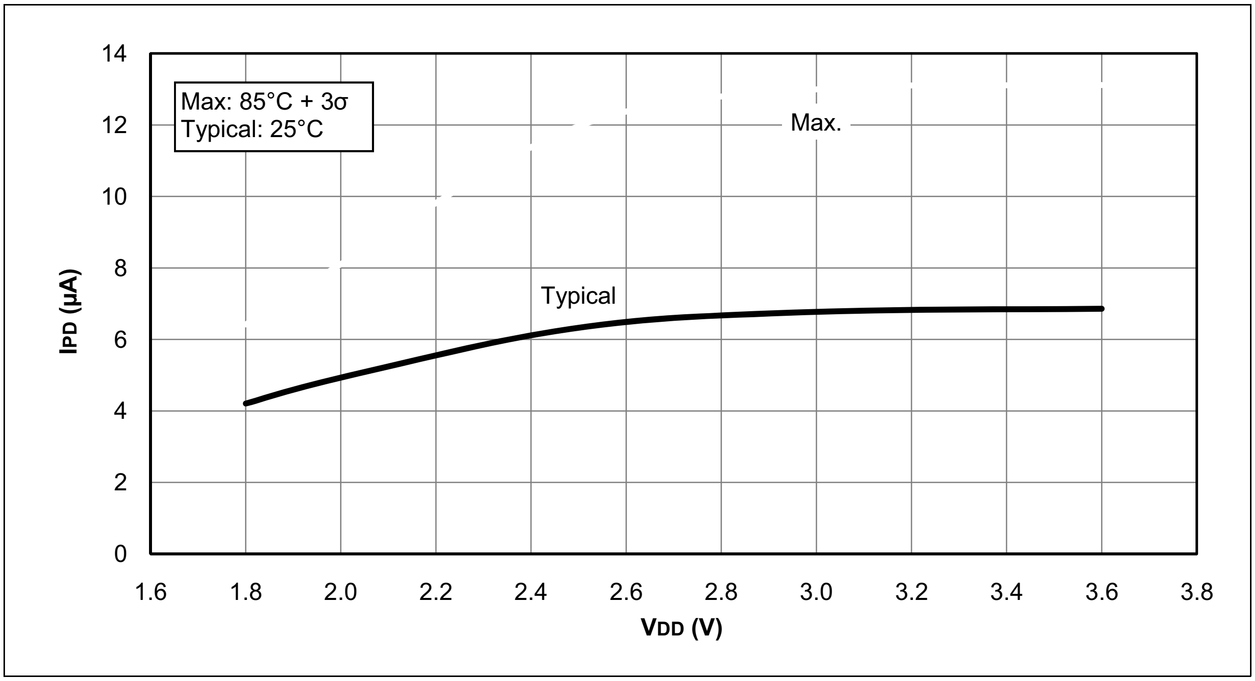
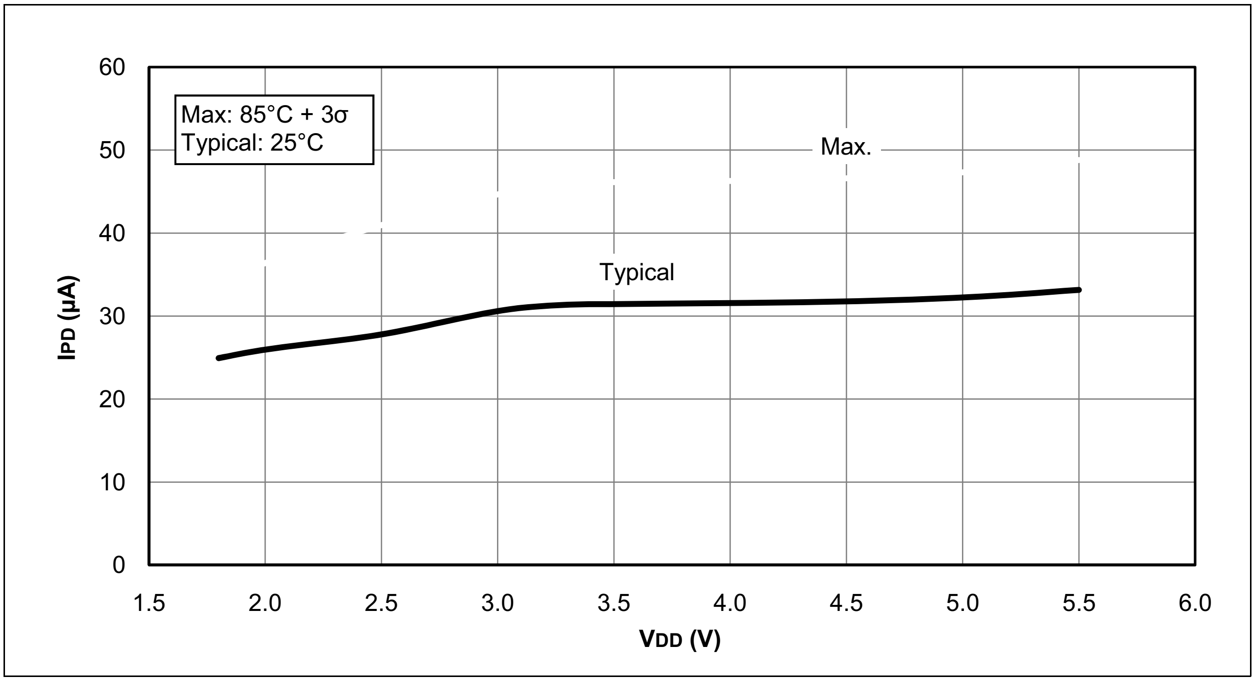
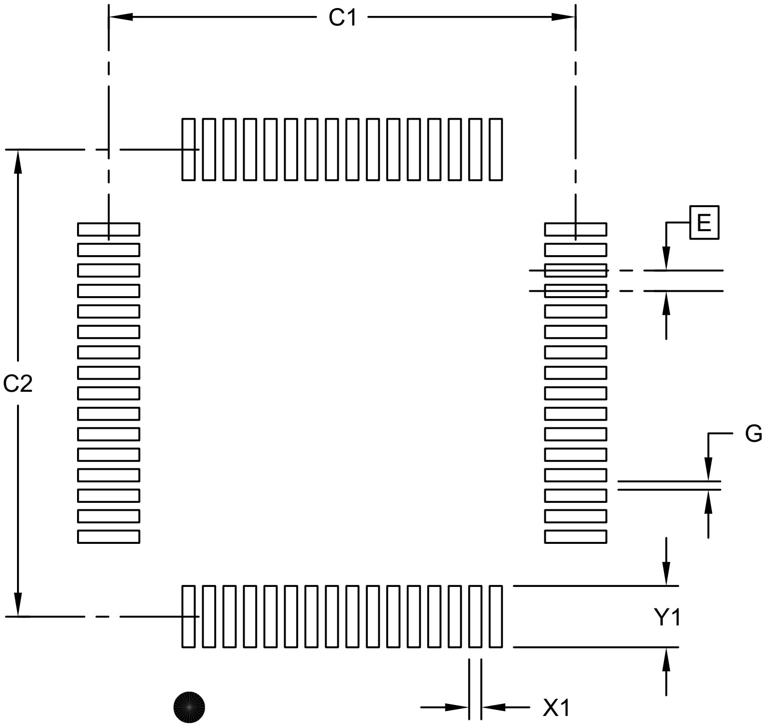


FIGURE 31-44: IPD, CAPACITIVE SENSING (CPS) MODULE, MEDIUM-CURRENT RANGE, CPSRM = 0, PIC16F1946/47 ONLY



64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		Units	MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2085B Sheet 1 of 1

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
══ ISO/TS 16949 ══

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-1072-0