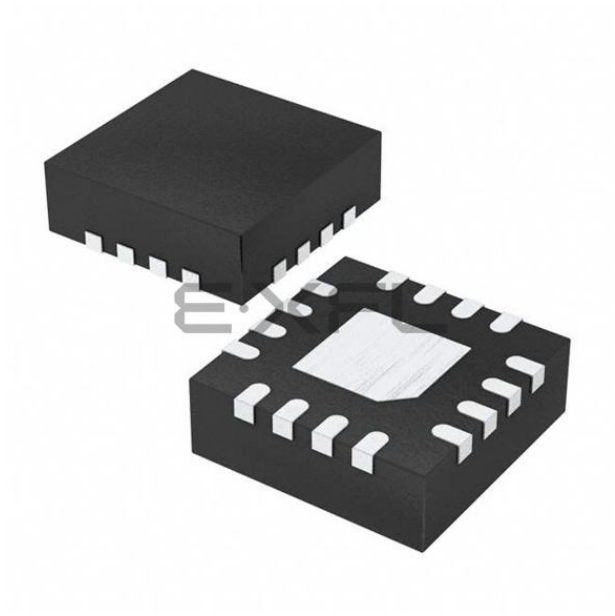


Welcome to E-XFL.COM

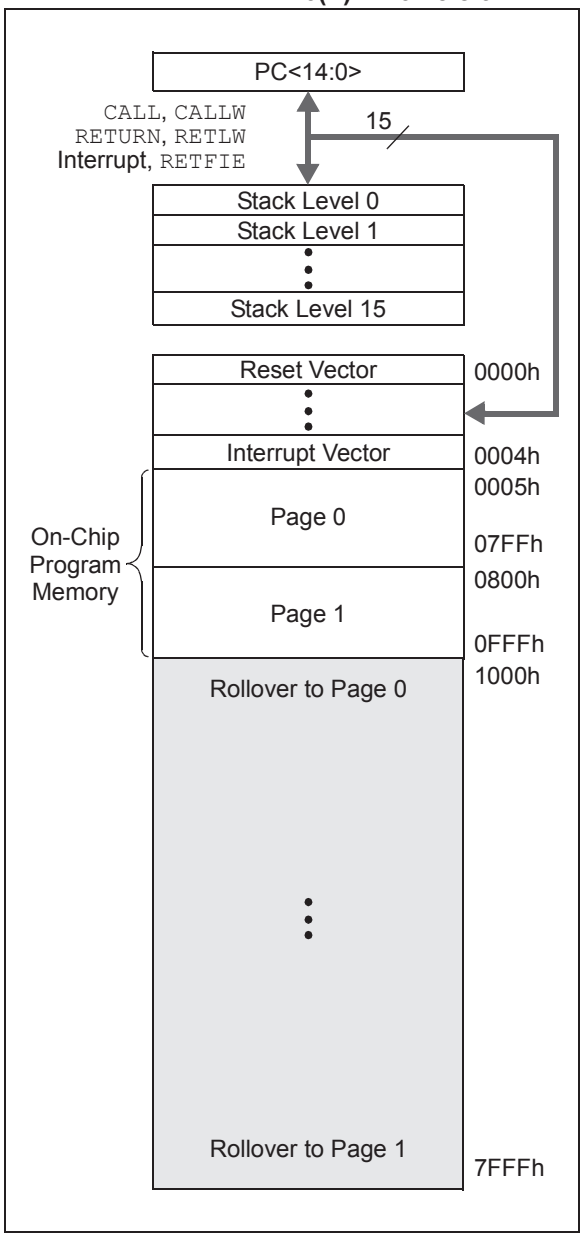
What is "Embedded - Microcontrollers"?

"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, POR, PWM, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 8x10b; D/A 1x5b, 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-VQFN Exposed Pad
Supplier Device Package	16-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1764-e-ml

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1764/5/8/9



3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of `RETLW` instructions. The second method is to set an FSRn to point to the program memory.

3.2.1.1 `RETLW` Instruction

The `RETLW` instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: `RETLW` INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The `BRW` instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the `BRW` instruction is not available, so the older table read method must be used.

3.2.1.2 Indirect Read with FSRn

The program memory can be accessed as data by setting bit 7 of the FSRnH register and reading the matching INDFn register. The `MOVIW` instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDFn registers. Instructions that access the program memory via the FSRn require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSRn.

The high directive will set bit 7 if a label points to a location in program memory.

TABLE 3-12: PIC16(L)F1764/5 MEMORY MAP (BANKS 27-30)

	Bank 27		Bank 28		Bank 29		Bank 30
D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—
D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—
D8Eh	PWMEN	E0Eh	—	E8Eh	—	F0Eh	—
D8Fh	PWMLD	E0Fh	PPSLOCK	E8Fh	—	F0Fh	CLCDATA
D90h	PWMOUT	E10h	INTPPS	E90h	RA0PPS	F10h	CLC1CON
D91h	PWM5PHL	E11h	T0CKIPPS	E91h	RA1PPS	F11h	CLC1POL
D92h	PWM5PHH	E12h	T1CKIPPS	E92h	RA2PPS	F12h	CLC1SEL0
D93h	PWM5DCL	E13h	T1GPPS	E93h	—	F13h	CLC1SEL1
D94h	PWM5DCH	E14h	CCP1PPS	E94h	RA4PPS	F14h	CLC1SEL2
D95h	PWM5PRL	E15h	—	E95h	RA5PPS	F15h	CLC1SEL3
D96h	PWM5PRH	E16h	COG1INPPS	E96h	—	F16h	CLC1GLS0
D97h	PWM5OFL	E17h	—	E97h	—	F17h	CLC1GLS1
D98h	PWM5OFH	E18h	—	E98h	—	F18h	CLC1GLS2
D99h	PWM5TMRL	E19h	T2INPPS	E99h	—	F19h	CLC1GLS3
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS	E9Ah	—	F1Ah	CLC2CON
D9Bh	PWM5CON	E1Bh	T3GPPS	E9Bh	—	F1Bh	CLC2POL
D9Ch	PWM5INTE	E1Ch	T4INPPS	E9Ch	—	F1Ch	CLC2SEL0
D9Dh	PWM5INTF	E1Dh	T5CKIPPS	E9Dh	—	F1Dh	CLC2SEL1
D9Eh	PWM5CLKCON	E1Eh	T5GPPS	E9Eh	—	F1Eh	CLC2SEL2
D9Fh	PWM5LDCON	E1Fh	T6INPPS	E9Fh	—	F1Fh	CLC2SEL3
DA0h	PWM5OFCON	E20h	SSPCLKPPS	EA0h	RC0PPS	F20h	CLC2GLS0
DA1h	—	E21h	SSPDATPPS	EA1h	RC1PPS	F21h	CLC2GLS1
DA2h	—	E22h	SSPSSPPS	EA2h	RC2PPS	F22h	CLC2GLS2
DA3h	—	E23h	—	EA3h	RC3PPS	F23h	CLC2GLS3
DA4h	—	E24h	RXPPS	EA4h	RC4PPS	F24h	CLC3CON
DA5h	—	E25h	CKPPS	EA5h	RC5PPS	F25h	CLC3POL
DA6h	—	E26h	—	EA6h	—	F26h	CLC3SEL0
DA7h	—	E27h	—	EA7h	—	F27h	CLC3SEL1
DA8h	—	E28h	CLCIN0PPS	EA8h	—	F28h	CLC3SEL2
DA9h	—	E29h	CLCIN1PPS	EA9h	—	F29h	CLC3SEL3
DAAh	—	E2Ah	CLCIN2PPS	EAAh	—	F2Ah	CLC3GLS0
DABh	—	E2Bh	CLCIN3PPS	EABh	—	F2Bh	CLC3GLS1
DACh	—	E2Ch	PRG1FPPS	EACh	—	F2Ch	CLC3GLS2
DADh	—	E2Dh	PRG1RPPS	EADh	—	F2Dh	CLC3GLS3
DAEh	—	E2Eh	—	EAEh	—	F2Eh	—
DAFh	—	E2Fh	—	EAFh	—	F2Fh	—
DB0h	—	E30h	MD1CHPPS	EB0h	—	F30h	—
DB1h	—	E31h	MD1CLPPS	EB1h	—	F31h	—
DB2h	—	E32h	MD1MODPPS	EB2h	—	F32h	—
DB3h	—	E33h	—	EB3h	—	F33h	—
DB4h	—	E34h	—	EB4h	—	F34h	—
DB5h	—	E35h	—	EB5h	—	F35h	—
DB6h	—	E36h	—	EB6h	—	F36h	—
DB7h	—	E37h	—	EB7h	—	F37h	—
DB8h	—	E38h	—	EB8h	—	F38h	—
DB9h	—	E39h	—	EB9h	—	F39h	—
DBAh	—	E3Ah	—	EBAh	—	F3Ah	—
DBBh	—	E3Bh	—	EBBh	—	F3Bh	—
DBCh	—	E3Ch	—	EBCh	—	F3Ch	—
DBDh	—	E3Dh	—	EBDh	—	F3Dh	—
DBEh	—	E3Eh	—	EBEh	—	F3Eh	—
DBFh	—	E3Fh	—	EBFh	—	F3Fh	—
DC0h — DEFh		E40h — E6Fh		EC0h — EEFh		F40h — F6Fh	

Legend: ☐ = Unimplemented data memory locations, read as '0',

REGISTER 4-2: CONFIG2: CONFIGURATION WORD 2 (CONTINUED)

bit 1-0 **WRT<1:0>:** Flash Memory Self-Write Protection bits

4 kW Flash Memory (PIC16(L)F1764/8):

11 = Off Write protection is off
10 = Boot 0000h to 01FFh are write-protected, 0200h to 0FFFh may be modified by PMCON control
01 = Half 0000h to 07FFh are write-protected, 0800h to 0FFFh may be modified by PMCON control
00 = All 0000h to 0FFFh are write-protected, no addresses may be modified by PMCON control

8 kW Flash Memory (PIC16(L)F1765/9):

11 = Off Write protection is off
10 = Boot 0000h to 01FFh are write-protected, 0200h to 1FFFh may be modified by PMCON control
01 = Half 0000h to 0FFFh are write-protected, 1000h to 1FFFh may be modified by PMCON control
00 = All 0000h to 1FFFh are write-protected, no addresses may be modified by PMCON control

Note 1: The LVP bit cannot be programmed to '0' when Programming mode is entered via LVP.

2: The $\overline{\text{DEBUG}}$ bit in the Configuration Words is managed automatically by device development tools, including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.

3: See V_{BOR} parameter for specific trip point voltages.

6.12 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON registers are updated to indicate the cause of the Reset. Table 6-3 and Table 6-4 show the Reset conditions of these registers.

TABLE 6-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STKOVF	STKUNF	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	0	1	1	1	0	x	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	1	0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
0	0	u	1	1	u	0	1	1	Brown-out Reset
u	u	0	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	u	0	u	u	u	u	u	$\overline{\text{MCLR}}$ Reset during Normal Operation
u	u	u	0	u	u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep
u	u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

TABLE 6-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
$\overline{\text{MCLR}}$ Reset during Normal Operation	0000h	---u uuuu	uu-- 0uuu
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1(1)	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

Legend: u = unchanged; x = unknown; - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit (GIE) is set, the return address is pushed onto the stack and the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

10.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 15-bit address of the program memory location being read.

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the operating voltage range of the device.

The Flash program memory can be protected in two ways: by code protection ($\overline{CP}$ bit in the Configuration Words) and write protection (WRT<1:0> bits in the Configuration Words).

Code protection ($\overline{CP}$ = 0) disables access, reading and writing to the Flash program memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a bulk erase to the device, clearing all Flash program memory, Configuration bits and User IDs.(1)

Write protection prohibits self-write and erase to a portion or all of the Flash program memory, as defined by the WRT<1:0> bits. Write protection does not affect a device programmer's ability to read, write or erase the device.

Note 1:	Code protection of the entire Flash program memory array is enabled by clearing the $\overline{CP}$ bit of the Configuration Words.

10.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 32K words of program memory. When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADRL register.

10.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash program memory accesses.

Control bits, RD and WR, initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash program memory.

10.2 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

Note:	If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

See Table 10-1 for erase row size and the number of write latches for Flash program memory.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC16(L)F1764	32	32
PIC16(L)F1765		
PIC16(L)F1768		
PIC16(L)F1769		

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

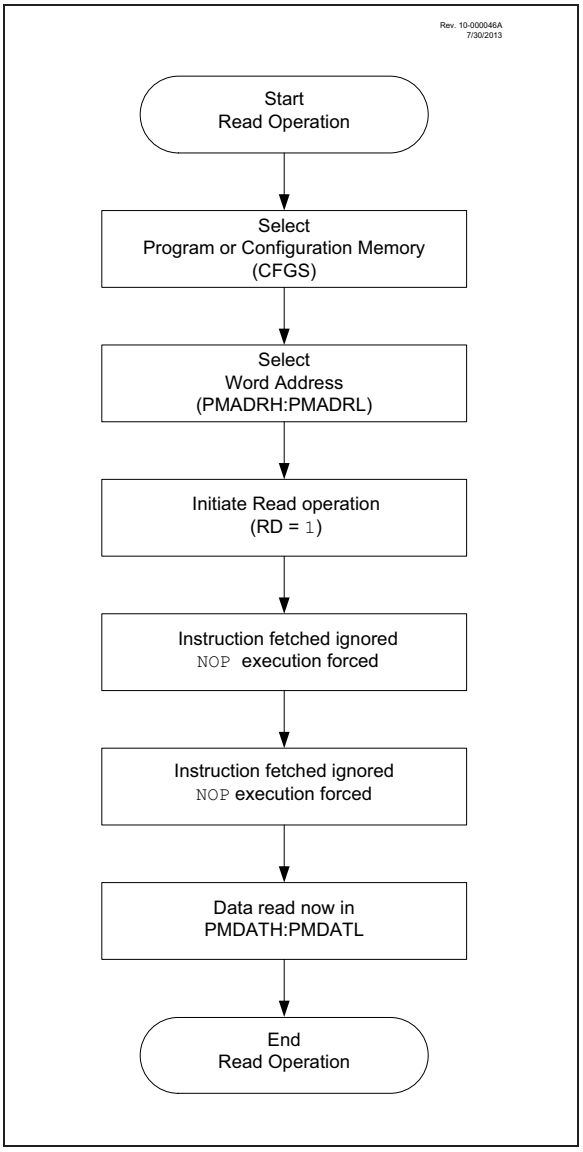
1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit, RD, of the PMCON1 register.

Once the read control bit is set, the Program Flash Memory controller will use the second instruction cycle to read the data. This causes the second instruction, immediately following the "`BSF PMCON1,RD`" instruction, to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

The PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

Note: The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART



REGISTER 11-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA<5:4>		—	SLRA<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	
u = Bit is unchanged	x = Bit is unknown	U = Unimplemented bit, read as '0'
'1' = Bit is set	'0' = Bit is cleared	-n/n = Value at POR and BOR/Value at all other Resets

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> Pins:
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> Pins:
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 11-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	INLVLA<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	
u = Bit is unchanged	x = Bit is unknown	U = Unimplemented bit, read as '0'
'1' = Bit is set	'0' = Bit is cleared	-n/n = Value at POR and BOR/Value at all other Resets

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> Pins:
1 = Port pin digital input operates with ST thresholds
0 = Port pin digital input operates with TTL thresholds

15.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, *"Use and Calibration of the Internal Temperature Indicator"* (DS00001333) for more details regarding the calibration process.

15.1 Circuit Operation

Figure 15-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 15-1 describes the output characteristics of the temperature indicator.

EQUATION 15-1: V_{OUT} RANGES

$$\text{High Range: } V_{OUT} = V_{DD} - 4V_T$$

$$\text{Low Range: } V_{OUT} = V_{DD} - 2V_T$$

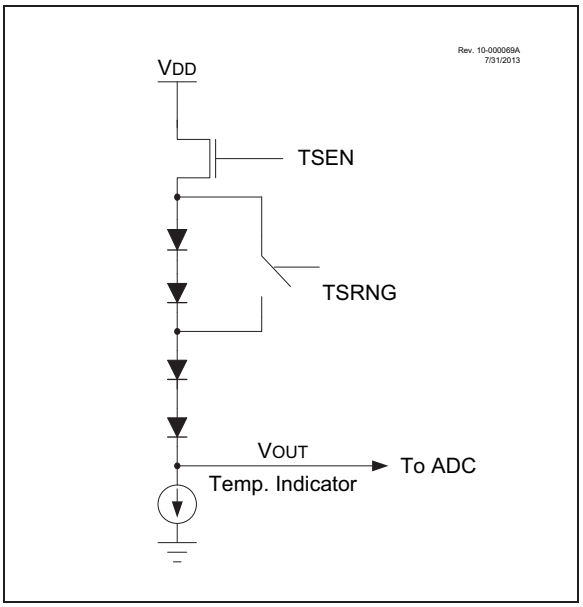
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher V_{DD} is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low-voltage operation.

FIGURE 15-1: TEMPERATURE CIRCUIT DIAGRAM



15.2 Minimum Operating V_{DD}

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

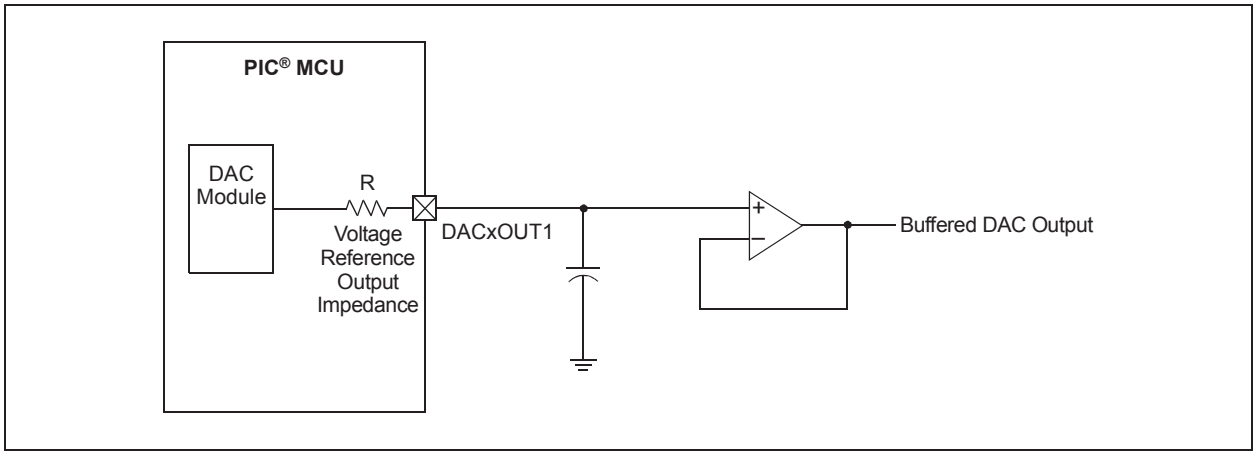
When the temperature circuit is operated in high range, the device operating voltage, V_{DD}, must be high enough to ensure that the temperature circuit is correctly biased.

Table 15-1 shows the recommended minimum V_{DD} vs. range setting.

TABLE 15-1: RECOMMENDED V_{DD} VS. RANGE

Min. V_{DD}, TSRNG = 1	Min. V_{DD}, TSRNG = 0
3.6V	1.8V

FIGURE 18-3: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



18.5 Operation During Sleep

When the device wakes up from Sleep as the result of an interrupt or a Watchdog Timer time-out, the contents of the DACxCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

18.6 Effects of a Reset

A device Reset affects the following:

- DAC is disabled
- DAC output voltage is removed from the DACxOUT1 pin
- The REF<9:0> reference selection bits are cleared

REGISTER 18-4: DACLD: DAC BUFFER LOAD REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	DAC2LD(1)	DAC1LD
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	q = value depends on configuration bits

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **DAC2LD:** DAC2 Double-Buffer Load bit(1)
- 1 = DAC2REFH:DAC2REFL values are transfered to the double-buffer; bit is cleared automatically by hardware
- 0 = DAC2REFH:DAC2REFL double-buffers remain unchanged

bit 0 **DAC1LD:** DAC1 Double-Buffer Load bit
- 1 = DAC1REFH:DAC1REFL values are transfered to the double-buffer; bit is cleared automatically by hardware
- 0 = DAC1REFH:DAC1REFL double-buffers remain unchanged

Note 1: PIC16(L)F1768/9 only

TABLE 18-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE DACx MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
DAC1CON0	EN	FM	OE1	—	PSS<1:0>		—	NSS	193
DAC2CON0(1)	EN	FM	OE1	—	PSS<1:0>		—	NSS	193
DAC1REFH	REF<9:x> (x Depends on FM bit)								194
DAC2REFH(1)	REF<9:x> (x Depends on FM bit)								194
DAC1REFL	REF<x-1:0> (x Depends on FM bit)								194
DAC2REFL(1)	REF<x-1:0> (x Depends on FM bit)								194
DACLD	—	—	—	—	—	—	DAC2LD(1)	DAC1LD	195

Legend: — = unimplemented location, read as '0'. Shaded cells are not used with the DACx module.

Note 1: PIC16(L)F1768/9 only

EQUATION 20-2: R-C CALCULATIONS

V_{peak} = external voltage source peak voltage
f = external voltage source frequency
C = series capacitor
R = series resistor
V_C = Peak capacitor voltage
ϕ = Capacitor induced zero crossing phase advance in radians
T_ϕ = Time ZC event occurs before actual zero crossing

$$Z = \frac{V_{PEAK}}{3x10^{-4}}$$

$$X_C = \frac{1}{(2\pi fC)}$$

$$R = \sqrt{Z^2 - X_C{}^2}$$

$$V_C = X_C(3x10^{-4})$$

$$\phi = Tan^{-1}\left(\frac{X_C}{R}\right)$$

$$T_\phi = \frac{\phi}{(2\pi f)}$$

EQUATION 20-3: R-C CALCULATIONS EXAMPLE

V_{rms} = 120

$$V_{peak} = V_{rms} \cdot \sqrt{2} = 169.7$$

f = 60 Hz

C = 0.1 µf

$$Z = \frac{V_{peak}}{3 \times 10^{-4}} = \frac{169.7}{3 \times 10^{-4}} = 565.7\ kOhms$$

$$X_C = \frac{1}{(2\pi fC)} = \frac{1}{(2\pi \cdot 60 \cdot 1 \cdot 10^{-7})} = 26.53\ \text{kOhms}$$

$$R = \sqrt{Z^2 - X_C{}^2} = 565.1\text{k Ohms (computed)}$$

R = 560 kOhms (used)

$$Z_R = \sqrt{\left(R^2 + X_c{}^2\right)} = 560.6\ kOhm\ \text{(using actual resistor)}$$

$$I_{peak} = \frac{V_{peak}}{Z_R} = 302.7 \cdot 10^{-6}$$

$$V_C = X_C \cdot I_{peak} = 8.0\ \text{V}$$

$$\phi = Tan^{-1}\left(\frac{X_C}{R}\right) = 0.047\ radians$$

$$T_\phi = \frac{\phi}{(2\pi f)} = 125.6\ \mu s$$

REGISTER 26-4: PWMxCLKCON: PWMx CLOCK CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	PS<2:0>			—	—	CS<1:0>	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **PS<2:0>:** Clock Source Prescaler Select bits
- 111 = Divide clock source by 128
- 110 = Divide clock source by 64
- 101 = Divide clock source by 32
- 100 = Divide clock source by 16
- 011 = Divide clock source by 8
- 010 = Divide clock source by 4
- 001 = Divide clock source by 2
- 000 = No Prescaler

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **CS<1:0>:** Clock Source Select bits
- 11 = Reserved
- 10 = LFINTOSC (continues to operate during Sleep)
- 01 = HFINTOSC (continues to operate during Sleep)
- 00 = FOSC

32.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100).

When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven.

When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

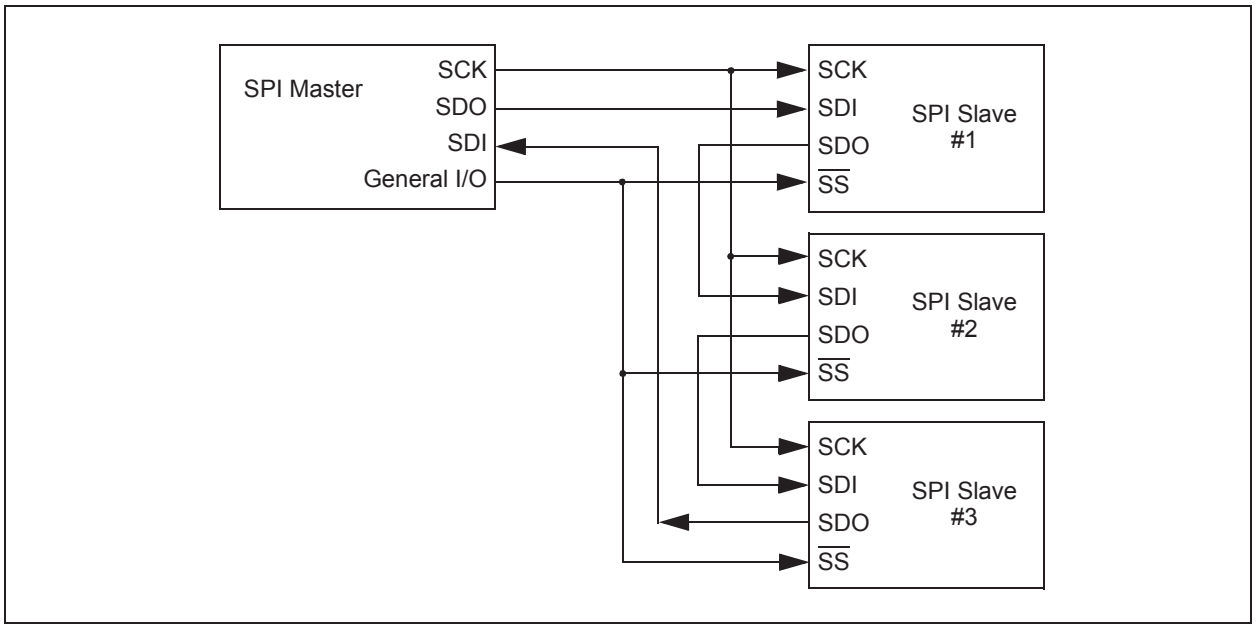
Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to V_{DD}.

2: When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SS}$ pin control.

3: While operating in SPI Slave mode, the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

FIGURE 32-7: SPI DAISY-CHAIN CONNECTION



33.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 33-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two-character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCxREG register.

33.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCxSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSELx bit must be cleared for the receiver to function.

33.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds, then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 33.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCxREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 33.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

33.1.2.3 Receive Interrupts

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

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

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

TABLE 33-2: SUMMARY OF 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
ANSELA	—	—	—	ANSA4	—	ANSA<2:0>			137
ANSELB(1)	ANSB<7:4>				—	—	—	—	143
ANSELC	ANSC<7:6>(1)		—	—	ANSC<3:0>				148
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	442
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	101
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	102
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	105
RC1REG	EUSART Receive Data Register								436*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	441
RxyPPS	—	—	—	RxyPPS<4:0>					154
SP1BRGL	BRG<7:0>								443
SP1BRGH	BRG<15:8>								443
TRISA	—	—	TRISA<5:4>		—(2)	TRISA<2:0>			136
TRISB(1)	TRISB<7:4>				—	—	—	—	142
TRISC	TRISC<7:6>(1)		TRISC<5:0>						147
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	440

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous reception.

* Page provides register information.

Note 1: PIC16(L)F1768/9 only.

2: Unimplemented, read as '1'.

33.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the Internal Oscillator Block (INTOSC) output. However, the INTOSC frequency may drift as VDD or temperature changes and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 5.2.2.3 "Internal Oscillator Frequency Adjustment"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 33.4.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

33.5.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXxSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXxSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCxSTA register enables the EUSART.

33.5.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 33.5.1.3 "Synchronous Master Transmission"**), except in the case of Sleep mode.

If two words are written to the TXxREG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in the TXxREG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXxREG register will transfer the second character to the TSR and the TXIF bit will now be set.
5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

33.5.2.2 Synchronous Slave Transmission Setup

1. Set the SYNC and SPEN bits, and clear the CSRC bit.
2. Clear the ANSEL bit for the CK pin (if applicable).
3. Clear the CREN and SREN bits.
4. If interrupts are desired, set the TXIE bit of the PIE1 register, and the GIE and PEIE bits of the INTCON register.
5. If 9-bit transmission is desired, set the TX9 bit.
6. Enable transmission by setting the TXEN bit.
7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
8. Start transmission by writing the Least Significant eight bits to the TXxREG register.

TABLE 36-4: I/O PORTS (CONTINUED)

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	V_{OH}	**Output High Voltage(4)**					
D090		Standard I/O Ports	V_{DD} – 0.7	—	—	V	I_{OH} = 3.5mA, V_{DD} = 5V I_{OH} = 3mA, V_{DD} = 3.3V I_{OH} = 1mA, V_{DD} = 1.8V
D090A		High Drive I/O Ports	V_{DD} – 0.7 — —	— V_{DD} - 0.7 V_{DD} - 0.7	— — —	V V V	I_{OH} = 10mA, V_{DD} = 2.3V, HIDCx = 1 I_{OH} = 37mA, V_{DD} = 3.0V, HIDCx = 1 I_{OH} = 54mA, V_{DD} = 5.0V, HIDCx = 1
		Capacitive Loading Specs on Output Pins					
D101*	COSC2	OSC2 Pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101A*	C_{IO}	All I/O Pins	—	—	50	pF	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in EXTRC mode.

2: Negative current is defined as current sourced by the pin.

3: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

4: Including OSC2 in CLKOUT mode.

Note: Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

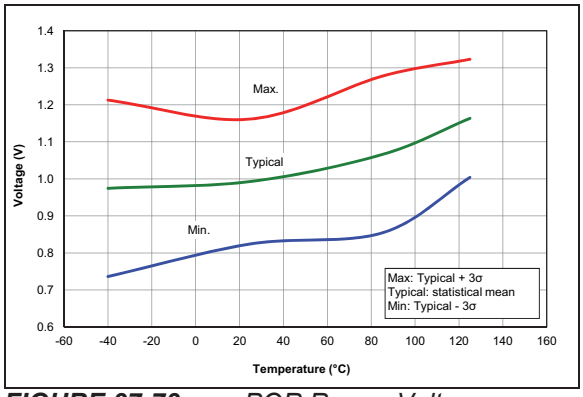


FIGURE 37-73: *POR Rearm Voltage, NP Mode, PIC16LF1764/5/8/9 Only.*

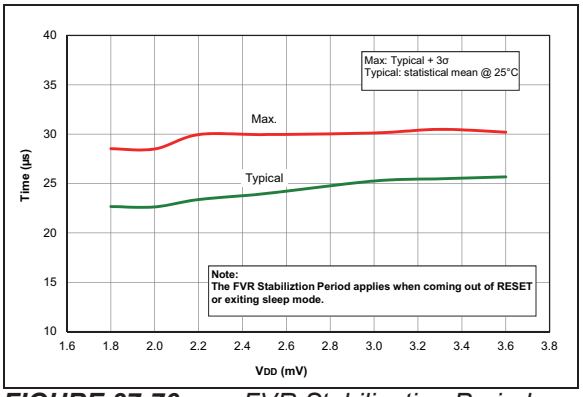


FIGURE 37-76: *FVR Stabilization Period, PIC16LF1764/5/8/9 Only.*

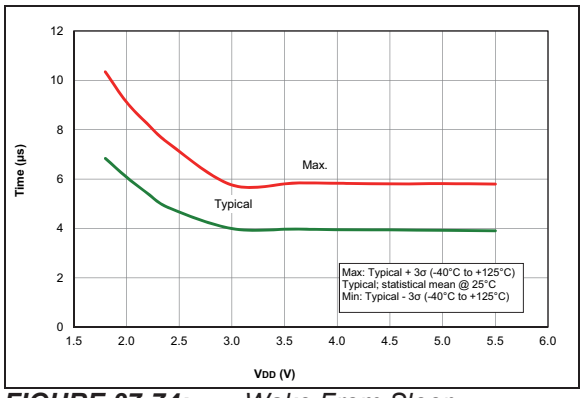


FIGURE 37-74: *Wake From Sleep, VREGPM = 0.*

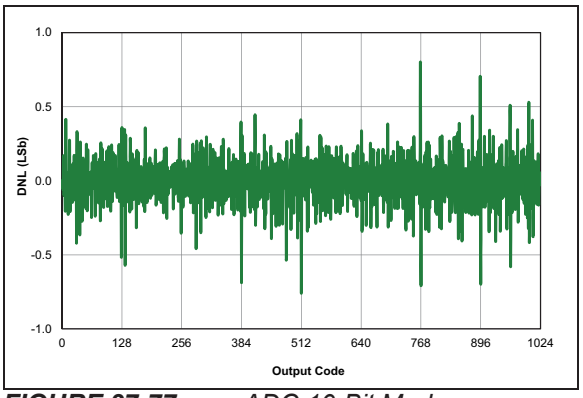


FIGURE 37-77: *ADC 10-Bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS, 25°C.*

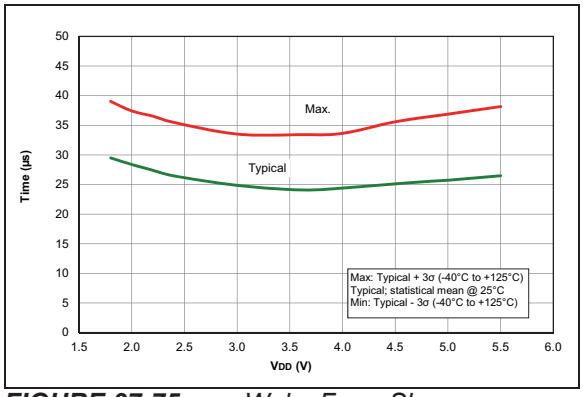


FIGURE 37-75: *Wake From Sleep, VREGPM = 1.*

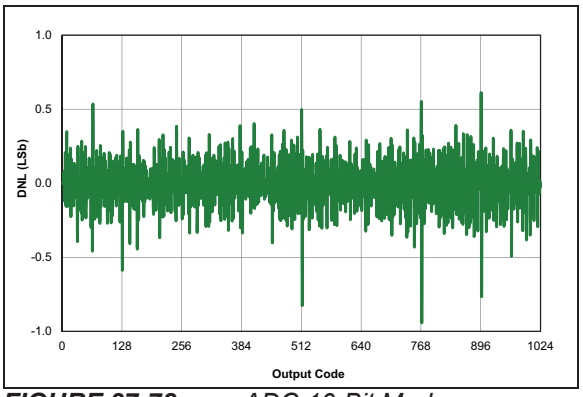
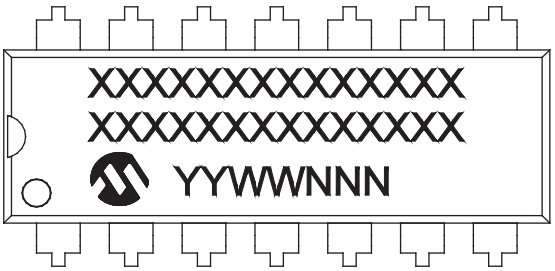


FIGURE 37-78: *ADC 10-Bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 4 µS, 25°C.*

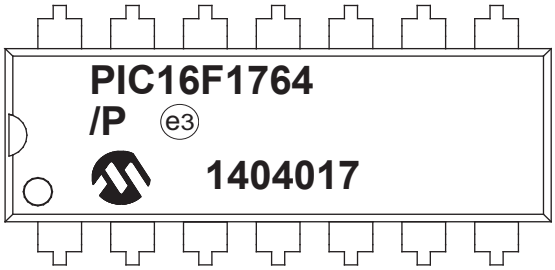
39.0 PACKAGING INFORMATION

39.1 Package Marking Information

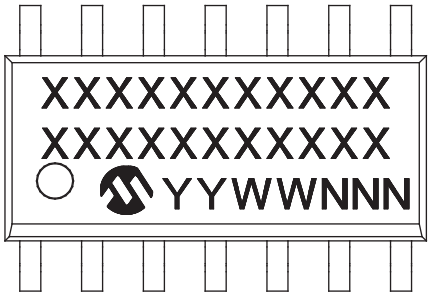
14-Lead PDIP (300 mil)



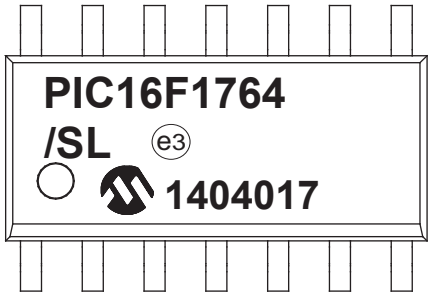
Example



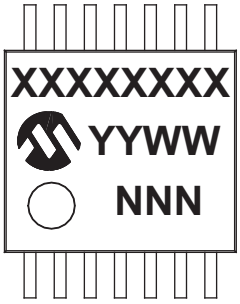
14-Lead SOIC (3.90 mm)



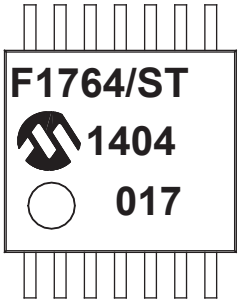
Example



14-Lead TSSOP (4.4 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC® designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC16F1764, PIC16LF1764, PIC16F1765, PIC16LF1765, PIC16F1768, PIC16LF1768, PIC16F1769, PIC16LF1769	
Tape and Reel Option:	Blank T	= Standard packaging (tube or tray) = Tape and Reel(1)
Temperature Range:	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package:(2)	ML P SL SO SS ST	= QFN = PDIP = SOIC = SOIC = SSOP = TSSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16LF1764- I/P
Industrial temperature
PDIP package

b) PIC16F1769- E/SS
Extended temperature,
SSOP package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

2: Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.