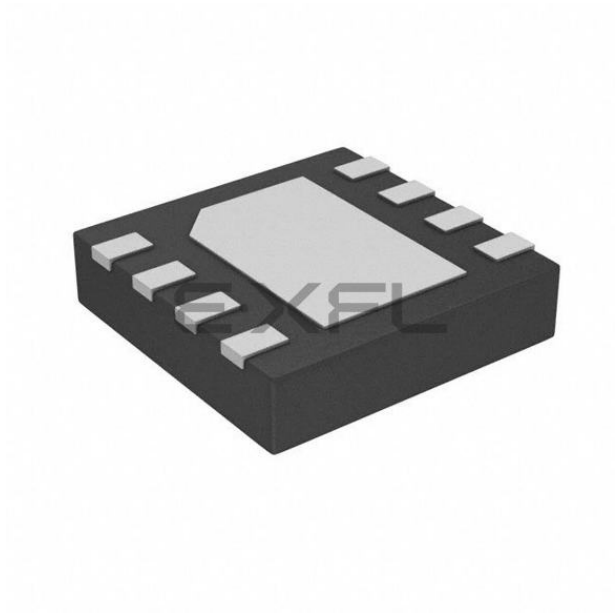


Welcome to E-XFL.COM

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	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12lf1571-e-mf

NOTES:

NOTES:

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 FSR 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 FSR

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 FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH operator will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
  DW     DATA0         ;First constant
  DW     DATA1         ;Second constant
  DW     DATA2
  DW     DATA3
my_function
  ;… LOTS OF CODE…
  MOVLW  DATA_INDEX
  ADDLW  LOW constants
  MOVWF  FSR1L
  MOVLW  HIGH constants ;MSb is set
                         automatically
  MOVWF  FSR1H
  BTFSC  STATUS,C      ;carry from ADDLW?
  INCF   FSR1H,f       ;yes
  MOVIW  0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

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

Addr	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 27											
D8Ch	—	Unimplemented								—	—
D8Dh	—	Unimplemented								—	—
D8Eh	PWMEN	—	—	—	—	—	PWM3EN_A	PWM2EN_A	PWM1EN_A	---- -000	---- -000
D8Fh	PWMLD	—	—	—	—	—	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	---- -000	---- -000
D90h	PWMOUT	—	—	—	—	—	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	---- -000	---- -000
D91h	PWM1PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
D92h	PWM1PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
D93h	PWM1DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
D94h	PWM1DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
D95h	PWM1PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
D96h	PWM1PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
D97h	PWM1OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
D98h	PWM1OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
D99h	PWM1TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
D9Ah	PWM1TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu
D9Bh	PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	PWM1MODE<1:0>		—	—	0000 00--	0000 00--
D9Ch	PWM1INTE	—	—	—	—	PWM1OFIE	PWM1PHIE	PWM1DCIE	PWM1PRIE	---- 000	---- 000
D9Dh	PWM1INTF	—	—	—	—	PWM1OFIF	PWM1PHIF	PWM1DCIF	PWM1PRIF	---- 000	---- 000
D9Eh	PWM1CLKCON	—	PWM1PS<2:0>			—	—	PWM1CS<1:0>		-000 -000	-000 --00
D9Fh	PWM1LDCON	PWM1LDA	PWM1LDT	—	—	—	—	PWM1LDS<1:0>		00-- -000	00-- --00
DA0h	PWM1OFCON	—	PWM1OFM<1:0>		PWM1OFO	—	—	PWM1OFS<1:0>		-000 -000	-000 --00
DA1h	PWM2PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
DA2h	PWM2PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
DA3h	PWM2DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
DA4h	PWM2DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
DA5h	PWM2PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
DA6h	PWM2PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
DA7h	PWM2OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
DA8h	PWM2OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
DA9h	PWM2TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
DAAh	PWM2TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu
DABh	PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	PWM2MODE<1:0>		—	—	0000 00--	0000 00--
DACh	PWM2INTE	—	—	—	—	PWM2OFIE	PWM2PHIE	PWM2DCIE	PWM2PRIE	---- 000	---- 000
DADh	PWM2INTF	—	—	—	—	PWM2OFIF	PWM2PHIF	PWM2DCIF	PWM2PRIF	---- 000	---- 000
DAEh	PWM2CLKCON	—	PWM2PS<2:0>			—	—	PWM2CS<1:0>		-000 -000	-000 --00
DAFh	PWM2LDCON	PWM2LDA	PWM2LDT	—	—	—	—	PWM2LDS<1:0>		00-- -000	00-- --00

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

Note 1: PIC12F1571/2 only.

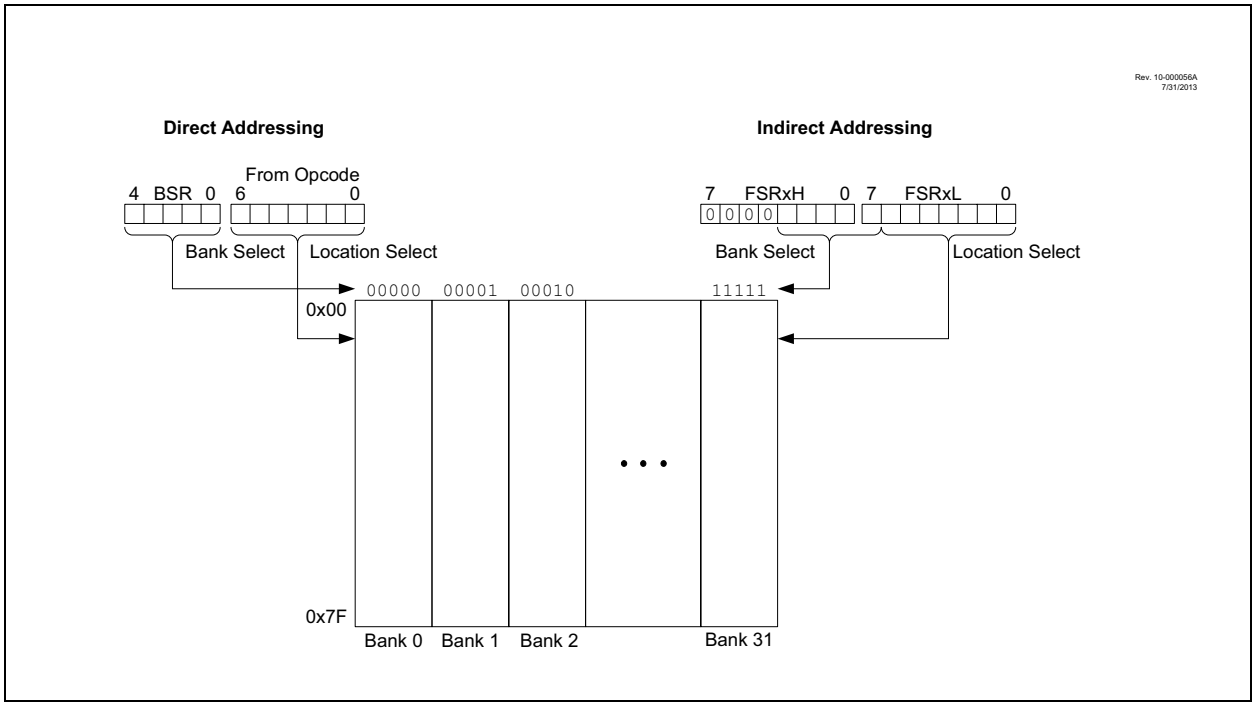
2: PIC12(L)F1572 only.

3: Unimplemented, read as '1'.

3.6.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address, 0x000, to FSR address, 0xFFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

FIGURE 3-10: TRADITIONAL DATA MEMORY MAP



5.5 Register Definitions: Oscillator Control

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>				—	SCS<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 **SPLLEN:** Software PLL Enable bit

If PLLEN in Configuration Words = 1:
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements).

If PLLEN in Configuration Words = 0:
1 = 4x PLL Is enabled
0 = 4x PLL is disabled

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits

1111 = 16 MHz HF
1110 = 8 MHz or 32 MHz HF (see **Section 5.2.2.1 “HFINTOSC”**)
1101 = 4 MHz HF
1100 = 2 MHz HF
1011 = 1 MHz HF
1010 = 500 kHz HF**(1)**
1001 = 250 kHz HF**(1)**
1000 = 125 kHz HF**(1)**
0111 = 500 kHz MF (default upon Reset)
0110 = 250 kHz MF
0101 = 125 kHz MF
0100 = 62.5 kHz MF
0011 = 31.25 kHz HF**(1)**
0010 = 31.25 kHz MF
000x = 31 kHz LF

bit 2 **Unimplemented:** Read as ‘0’

bit 1-0 **SCS<1:0>:** System Clock Select bits

1x = Internal oscillator block
01 = Timer1 oscillator
00 = Clock determined by FOSC<1:0> in Configuration Words

Note 1: Duplicate frequency derived from HFINTOSC.

7.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to **Section 8.0 "Power-Down Mode (Sleep)"** for more details.

7.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

7.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

REGISTER 15-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
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 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result.

REGISTER 15-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<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 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result.

bit 5-0 **Reserved**: Do not use

16.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACxCON1 register.

The DAC output voltage can be determined by using Equation 16-1.

16.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 26-16.

16.3 DAC Voltage Reference Output

The unbuffered DAC voltage can be output to the DACxOUTn pin(s) by setting the respective DACOEn bit(s) of the DACxCON0 register. Selecting the DAC reference voltage for output on either DACxOUTn pin automatically overrides the digital output buffer, the weak pull-up and digital input threshold detector functions of that pin.

Reading the DACxOUTn pin when it has been configured for DAC reference voltage output will always return a '0'.

Note:	The unbuffered DAC output (DACxOUTn) is not intended to drive an external load.

16.4 Operation During Sleep

When the device wakes up from Sleep through 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.

16.5 Effects of a Reset

A device Reset affects the following:

- DACx is disabled.
- DACx output voltage is removed from the DACxOUTn pin(s).
- The DACR<4:0> range select bits are cleared.

EQUATION 16-1: DAC OUTPUT VOLTAGE

IF DACEN = 1

$$DACx\_output = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5} \right) + V_{SOURCE-}$$

Note: See the DACxCON0 register for the available $V_{SOURCE+}$ and $V_{SOURCE-}$ selections.

17.8 Register Definitions: Comparator Control

REGISTER 17-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	CxPOL	—	CxSP	CxHYS	CxSYNC
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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

bit 5 **CxOE:** Comparator Output Enable bit
1 = CxOUT is present on the CxOUT pin; requires that the associated TRISx bit be cleared to actually drive the pin, not affected by CxON
0 = CxOUT is internal only

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator mode is in Normal Power, Higher Speed mode
0 = Comparator mode is in Low-Power, Low-Speed mode

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis is enabled
0 = Comparator hysteresis is disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source; output updated on the falling edge of Timer1 clock source
0 = Comparator output to Timer1 and I/O pin is asynchronous

19.8 Register Definitions: Timer1 Control

REGISTER 19-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON
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 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- 11 = Timer1 clock source is the LFINTOSC
- 10 = Timer1 clock source is the T1CKI pin (on the rising edge)
- 01 = Timer1 clock source is the system clock (FOSC)
- 00 = Timer1 clock source is the instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- 11 = 1:8 Prescale value
- 10 = 1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

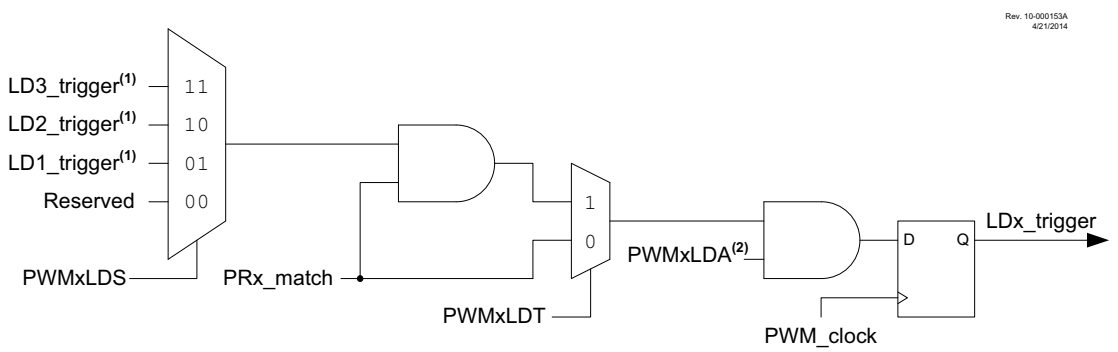
bit 3 **Unimplemented:** Read as ‘0’

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 Synchronization Control bit
- 1 = Does not synchronize the asynchronous clock input
- 0 = Synchronizes the asynchronous clock input with the system clock (FOSC)

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **TMR1ON:** Timer1 On bit
- 1 = Enables Timer1
- 0 = Stops Timer1 and clears Timer1 gate flip-flop

FIGURE 22-2: LOAD TRIGGER BLOCK DIAGRAM



Note 1. The input corresponding to a PWM module's own load trigger is reserved.
2. PWMxLDA is cleared by hardware upon LDx_trigger.

22.1 Fundamental Operation

The PWM module produces a 16-bit resolution pulse-width modulated output.

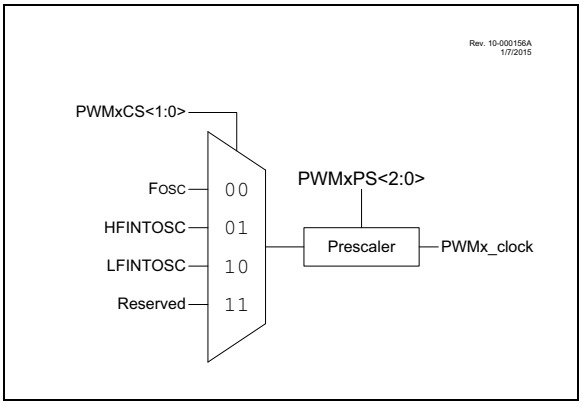
Each PWM module has an independent timer driven by a selection of clock sources determined by the PWMxCLKCON register (Register 22-4). The timer value is compared to event count registers to generate the various events of a the PWM waveform, such as the period and duty cycle. For a block diagram describing the clock sources, refer to Figure 22-3.

Each PWM module can be enabled individually using the EN bit of the PWMxCON register, or several PWM modules can be enabled simultaneously using the mirror bits of the PWMEN register.

The current state of the PWM output can be read using the OUT bit of the PWMxCON register. In some modes, this bit can be set and cleared by software, giving additional software control over the PWM waveform. This bit is synchronized to FOSC/4 and therefore, does not change in real time with respect to the PWM_clock.

Note:	If PWM_clock > FOSC/4, the OUT bit may not accurately represent the output state of the PWM.

FIGURE 22-3: PWM CLOCK SOURCE BLOCK DIAGRAM



22.1.1 PWMx PIN CONFIGURATION

All PWM outputs are multiplexed with the PORT data latch, so the pins must also be configured as outputs by clearing the associated PORT TRISx bits.

The slew rate feature may be configured to optimize the rate to be used in conjunction with the PWM outputs. High-speed output switching is attained by clearing the associated PORT SLRCONx bits.

The PWM outputs can be configured to be open-drain outputs by setting the associated PORT ODCONx bits.

22.1.2 PWMx Output Polarity

The output polarity is inverted by setting the POL bit of the PWMxCON register. The polarity control affects the PWM output even when the module is not enabled.

REGISTER 22-3: PWMxINTF: PWMx INTERRUPT REQUEST REGISTER

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	OFIF	PHIF	DCIF	PRIF
bit 7							bit 0

Legend:		
HC = Hardware Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	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 7-4 **Unimplemented:** Read as '0'

bit 3 **OFIF**: Offset Interrupt Flag bit(1)
- 1 = Offset match event occurred
- 0 = Offset match event did not occur

bit 2 **PHIF:** Phase Interrupt Flag bit(1)
- 1 = Phase match event occurred
- 0 = Phase match event did not occur

bit 1 **DCIF:** Duty Cycle Interrupt Flag bit(1)
- 1 = Duty cycle match event occurred
- 0 = Duty cycle match event did not occur

bit 0 **PRIF:** Period Interrupt Flag bit(1)
- 1 = Period match event occurred
- 0 = Period match event did not occur

Note 1: Bit is forced clear by hardware while module is disabled (EN = 0).

REGISTER 22-4: 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 = Divides clock source by 128
- 110 = Divides clock source by 64
- 101 = Divides clock source by 32
- 100 = Divides clock source by 16
- 011 = Divides clock source by 8
- 010 = Divides clock source by 4
- 001 = Divides 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

REGISTER 22-6: PWMxOFCON: PWMx OFFSET TRIGGER SOURCE SELECT 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
—	OFM<1:0>		OFO(1)	—	—	OFS<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 6-5 **OFM<1:0>:** Offset Mode Select bits
- 11 = Continuous Slave Run mode with immediate Reset and synchronized start when the selected offset trigger occurs
- 10 = One-Shot Slave Run mode with synchronized start when the selected offset trigger occurs
- 01 = Independent Slave Run mode with synchronized start when the selected offset trigger occurs
- 00 = Independent Run mode

bit 4 **OFO:** Offset Match Output Control bit(1)

If MODE<1:0> = 11 (PWM Center-Aligned mode):
- 1 = OFx_match occurs on counter match when counter decrementing, (second match)
- 0 = OFx_match occurs on counter match when counter incrementing, (first match)

If MODE<1:0> = 00, 01 or 10 (all other modes):
Bit is ignored.

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **OFS<1:0>:** Offset Trigger Source Select bits
- 11 = OF3_match(1)
- 10 = OF2_match(1)
- 01 = OF1_match(1)
- 00 = Reserved

Note 1: The OFx_match corresponding to the PWM used becomes reserved.

26.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

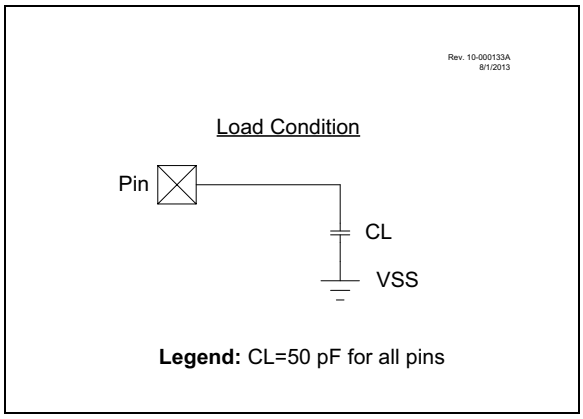
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	CLKIN
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDIx	sc	SCKx
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 26-4: LOAD CONDITIONS



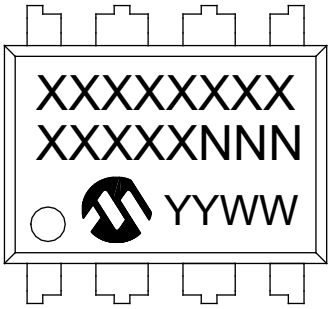
Legend: CL=50 pF for all pins

NOTES:

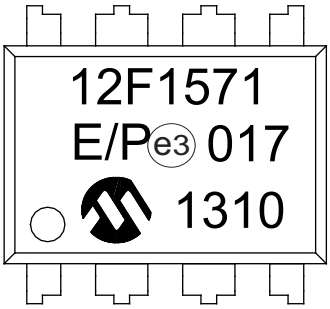
29.0 PACKAGING INFORMATION

29.1 Package Marking Information

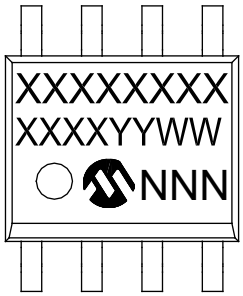
8-Lead PDIP (300 mil)



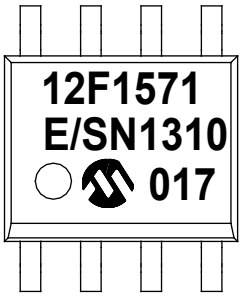
Example



8-Lead SOIC (3.90 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.

TABLE 29-1: 8-LEAD 3x3x0.9 DFN (MF) TOP MARKING

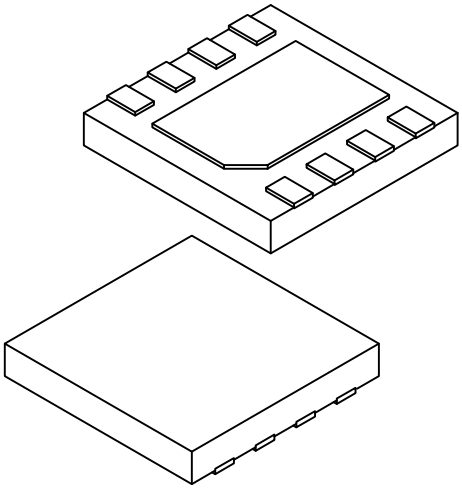
Part Number	Marking
PIC12F1571-E/MF	MFY0/YYWW/NNN
PIC12F1572-E/MF	MGA0/YYWW/NNN
PIC12F1571-I/MF	MFZ0
PIC12F1572-I/MF	MGB0
PIC12LF1571-E/MF	MGC0
PIC12LF1572-E/MF	MGE0
PIC12LF1571-I/MF	MGD0
PIC12LF1572-I/MF	MGF0

TABLE 29-2: 8-LEAD 3x3x0.5 UDFN (RF) TOP MARKING

Part Number	Marking
PIC12F1571-E/MF	MFY0/YYWW/NNN
PIC12F1572-E/MF	MGA0/YYWW/NNN
PIC12F1571-I/MF	MFZ0
PIC12F1572-I/MF	MGB0
PIC12LF1571-E/MF	MGC0
PIC12LF1572-E/MF	MGE0
PIC12LF1571-I/MF	MGD0
PIC12LF1572-I/MF	MGF0

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		8	
Pitch	e		0.65 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.065 REF	
Overall Width	E		3.00 BSC	
Exposed Pad Width	E2	1.40	1.50	1.60
Overall Length	D		3.00 BSC	
Exposed Pad Length	D2	2.20	2.30	2.40
Terminal Width	b	0.25	0.30	0.35
Terminal Length	L	0.35	0.45	0.55
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

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

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-254A Sheet 2 of 2