Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

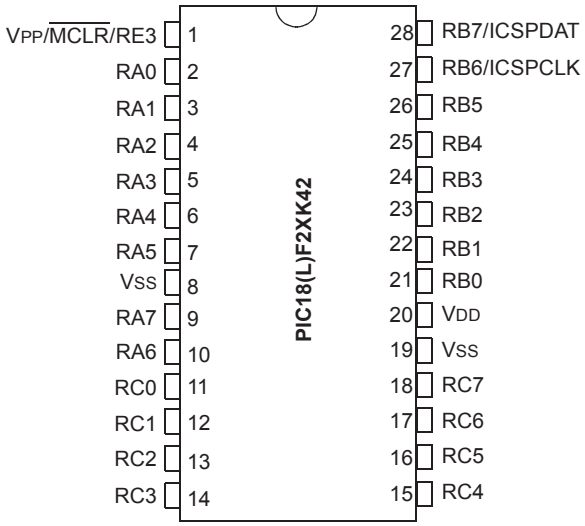
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k42-e-ss

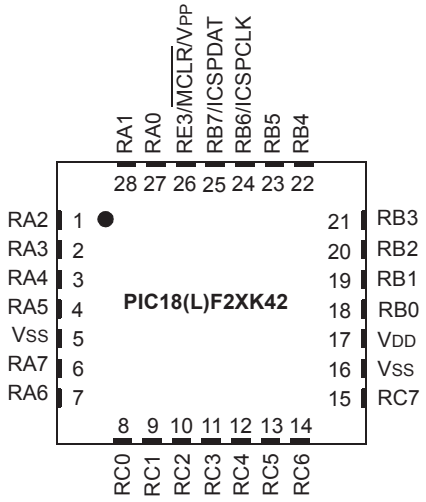
Pin Diagrams

28-pin SPDIP, SOIC, SSOP



Note: See Table 1 for location of all peripheral functions.

28-pin QFN (6x6x0.9mm), UQFN (4x4x0.5mm)



Note 1: See Table 1 for location of all peripheral functions.

2: It is recommended that the exposed bottom pad be connected to VSS, however it must not be the only VSS connection to the device.

REGISTER 8-3: PCON1: POWER CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W/HC-1/u	U-0
—	—	—	—	—	—	$\overline{\text{MEMV}}$	—
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	-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 7-2 **Unimplemented:** Read as '0'

bit 1 $\overline{\textbf{MEMV}}$**:** Memory Violation Flag bit
- 1 = No memory violation Reset occurred or set to '1' by firmware
- 0 = A memory violation Reset occurred (set to '0' in hardware when a Memory Violation occurs)

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

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	90
PCON0	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	95
PCON1	—	—	—	—	—	—	$\overline{\text{MEMV}}$	—	96

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Resets.

REGISTER 11-3: PIR0: PERIPHERAL INTERRUPT Request Register 0

R-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCIF	CRCIF	SCANIF	NVMIF	CSWIF	OSFIF	HLVDIF	SWIF
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	HS = Bit is set in hardware

bit 7 **IOCIF:** Interrupt-on-Change Interrupt Flag bit
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 6 **CRCIF:** CRC Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 5 **SCANIF:** Memory Scanner Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 4 **NVMIF:** NVM Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 3 **CSWIF:** Clock Switch Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 2 **OSFIF:** Oscillator Fail Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 1 **HLVDIF:** HLVD Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 0 **SWIF:** Software Interrupt Flag bit
- 1 = Software Interrupt Flag Enable
- 0 = Software Interrupt Flag Disable

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 11-10: PIR7: PERIPHERAL INTERRUPT REGISTER 7

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	R/W/HS-0/0	R/W/HS-0/0
—	—	INT2IF	CLC2IF	CWG2IF	—	CCP2IF	TMR4IF
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	HS = Bit is set in hardware

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

bit 5 **INT2IF:** External Interrupt 2 Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 4 **CLC2IF:** CLC2 Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 3 **CWG2IF:** CWG2 Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

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

bit 1 **CCP2IF:** CCP2 Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 0 **TMR4IF:** TMR4 Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-2: WDTCON1: WATCHDOG TIMER CONTROL REGISTER 1

U-0	R/W(3)-q/q(1)	R/W(3)-q/q(1)	R/W(3)-q/q(1)	U-0	R/W(4)-q/q(2)	R/W(4)-q/q(2)	R/W(4)-q/q(2)
—	CS<2:0>			—	WINDOW<2:0>		
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 condition

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

bit 6-4 **CS<2:0>:** Watchdog Timer Clock Select bits

111 = Reserved
•
•
•
011 = Reserved
010 = SOSC
001 = MFINTOSC 31.25 kHz
000 = LFINTOSC 31 kHz

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

bit 2-0 **WINDOW<2:0>:** Watchdog Timer Window Select bits

WINDOW<2:0>	Window delay Percent of time	Window opening Percent of time
111	N/A	100
110	12.5	87.5
101	25	75
100	37.5	62.5
011	50	50
010	62.5	37.5
001	75	25
000	87.5	12.5

Note 1: If WDTCCS <2:0> in CONFIG3H = 111, the Reset value of CS<2:0> is 000.

2: The Reset value of WINDOW<2:0> is determined by the value of WDTCWS<2:0> in the CONFIG3H register.

3: If WDTCCS<2:0> in CONFIG3H ≠ 111, these bits are read-only.

4: If WDTCWS<2:0> in CONFIG3H ≠ 111, these bits are read-only.

REGISTER 17-1: DMAxCON0 – DMAx CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	R/W/HS/HC-0/0	U-0	U-0	R/W/HC-0/0	U-0	R/HS/HC-0/0
EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP
bit 7							bit 0

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

bit 7 **EN:** DMA Module Enable bit
- 1 = Enables module
- 0 = Disables module

bit 6 **SIRQEN:** Start of Transfer Interrupt Request Enable bits
- 1 = Hardware triggers are allowed to start DMA transfers
- 0 = Hardware triggers are not allowed to start DMA transfers

bit 5 **DGO:** DMA transaction bit
- 1 = DMA transaction is in progress
- 0 = DMA transaction is not in progress

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **AIRQEN:** Abort of Transfer Interrupt Request Enable bits
- 1 = Hardware triggers are allowed to abort DMA transfers
- 0 = Hardware triggers are not allowed to abort DMA transfers

bit 1 **Unimplemented:** Read as '0'

bit 0 **XIP:** Transfer in Progress Status bit
- 1 = The DMAxBUF register currently holds contents from a read operation and has not transferred data to the destination.
- 0 = The DMAxBUF register is empty or has successfully transferred data to the destination address

19.8 Register Definitions: PPS Input Selection

REGISTER 19-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION

U-0	U-0	U-0	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)
—	—	—	xxxPPS<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	-n/n = Value at POR and BOR/Value at all other Resets
u = Bit is unchanged	x = Bit is unknown	q = value depends on peripheral
'1' = Bit is set	U = Unimplemented bit, read as '0'	m = value depends on default location for that input
'0' = Bit is cleared		

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **xxxPPS<4:3>:** Peripheral xxx Input PORTx Pin Selection bits
See Table 19-1 for the list of available ports and default pin locations.
11 = Reserved
10 = PORTC
01 = PORTB
00 = PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input PORTx Pin Selection bits
111 = Peripheral input is from PORTx Pin 7 (Rx7)
110 = Peripheral input is from PORTx Pin 6 (Rx6)
101 = Peripheral input is from PORTx Pin 5 (Rx5)
100 = Peripheral input is from PORTx Pin 4 (Rx4)
011 = Peripheral input is from PORTx Pin 3 (Rx3)
010 = Peripheral input is from PORTx Pin 2 (Rx2)
001 = Peripheral input is from PORTx Pin 1 (Rx1)
000 = Peripheral input is from PORTx Pin 0 (Rx0)

Note 1: The Reset value 'm' of this register is determined by device default locations for that input.

TABLE 19-1: PPS INPUT REGISTER DETAILS

Peripheral	PPS Input Register	Default Pin Selection at POR	Register Reset Value at POR	Input Available from Selected PORTx		
Interrupt 0	INT0PPS	RB0	5'b0 1000	A	B	—
Interrupt 1	INT1PPS	RB1	5'b0 1001	A	B	—
Interrupt 2	INT2PPS	RB2	5'b0 1010	A	B	—
Timer0 Clock	T0CKIPPS	RA4	5'b0 0100	A	B	—
Timer1 Clock	T1CKIPPS	RC0	5'b1 0000	A	—	C
Timer1 Gate	T1GPPS	RB5	5'b0 1101	—	B	C
Timer3 Clock	T3CKIPPS	RC0	5'b1 0000	—	B	C
Timer3 Gate	T3GPPS	RC0	5'b1 0000	A	—	C
Timer5 Clock	T5CKIPPS	RC2	5'b1 0010	A	—	C
Timer5 Gate	T5GPPS	RB4	5'b0 1100	—	B	C
Timer2 Clock	T2INPPS	RC3	5'b1 0011	A	—	C
Timer4 Clock	T4INPPS	RC5	5'b1 0101	—	B	C

REGISTER 23-3: TxCLK: TIMERx CLOCK REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	CS<4:0>				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	u = unchanged

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CS<4:0>:** Timerx Clock Source Selection bits

CS	Timer1	Timer3	Timer5
	Clock Source	Clock Source	Clock Source
11111-10001	Reserved	Reserved	Reserved
10000	CLC4	CLC4	CLC4
01111	CLC3	CLC3	CLC3
01110	CLC2	CLC2	CLC2
01101	CLC1	CLC1	CLC1
01100	TMR5 overflow	TMR5 overflow	Reserved
01011	TMR3 overflow	Reserved	TMR3 overflow
01010	Reserved	TMR1 overflow	TMR1 overflow
01001	TMR0 overflow	TMR0 overflow	TMR0 overflow
01000	CLKREF	CLKREF	CLKREF
00111	SOSC	SOSC	SOSC
00110	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)
00101	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)
00100	LFINTOSC	LFINTOSC	LFINTOSC
00011	HFINTOSC	HFINTOSC	HFINTOSC
00010	Fosc	Fosc	Fosc
00001	Fosc/4	Fosc/4	Fosc/4
00000	T1CKIPPS	T3CKIPPS	T5CKIPPS

26.1.5 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown by Equation 26-4.

EQUATION 26-4: PWM RESOLUTION

$$Resolution = \frac{log[4(T2PR + 1)]}{log(2)} \ bits$$

Note:	If the pulse-width value is greater than the period the assigned PWM pin(s) will remain unchanged.

TABLE 26-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 26-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

26.1.6 OPERATION IN SLEEP MODE

In Sleep mode, the T2TMR register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, T2TMR will continue from its previous state.

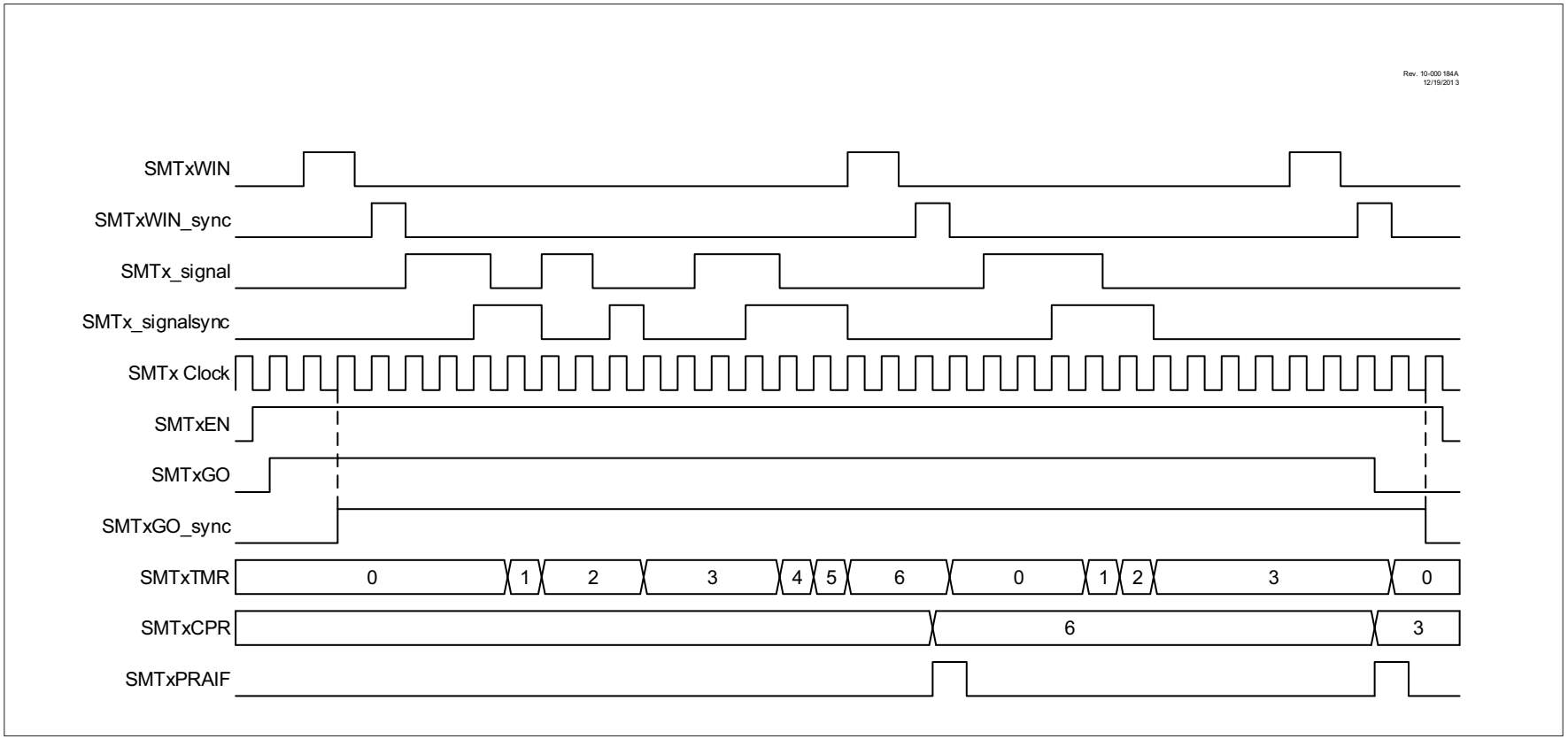
26.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency (FOSC). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 9.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

26.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

FIGURE 27-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



REGISTER 29-7: CLCxGLS0: GATE 0 LOGIC SELECT REGISTER

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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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	

bit 7 **G1D4T:** Gate 0 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 0
0 = CLCIN3 (true) is not gated into CLCx Gate 0

bit 6 **G1D4N:** Gate 0 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 0
0 = CLCIN3 (inverted) is not gated into CLCx Gate 0

bit 5 **G1D3T:** Gate 0 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 0
0 = CLCIN2 (true) is not gated into CLCx Gate 0

bit 4 **G1D3N:** Gate 0 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 0
0 = CLCIN2 (inverted) is not gated into CLCx Gate 0

bit 3 **G1D2T:** Gate 0 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 0
0 = CLCIN1 (true) is not gated into l CLCx Gate 0

bit 2 **G1D2N:** Gate 0 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 0
0 = CLCIN1 (inverted) is not gated into CLCx Gate 0

bit 1 **G1D1T:** Gate 0 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 0
0 = CLCIN0 (true) is not gated into CLCx Gate 0

bit 0 **G1D1N:** Gate 0 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 0
0 = CLCIN0 (inverted) is not gated into CLCx Gate 0

REGISTER 32-3: MD1CARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CH<4:0>(1)				
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	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CH<4:0>:** Modulator Carrier High Selection bits(1)
See Table 32-2 for signal list

Note 1: Unused selections provide an input value.

REGISTER 32-4: MD1CARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CL<4:0>(1)				
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	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CL<4:0>:** Modulator Carrier Low Input Selection bits(1)
See Table 32-2 for signal list

Note 1: Unused selections provide a zero as the input value.

TABLE 32-2: MD1CARH/MD1CARL SELECTION MUX CONNECTIONS

MD1CARH			MD1CARL		
CH<4:0>		Connection	CL<4:0>		Connection
11111-10011	31-19	Reserved	11111-10011	31-19	Reserved
10010	18	CLC4OUT	10010	18	CLC4OUT
10001	17	CLC3OUT	10001	17	CLC3OUT
10000	16	CLC2OUT	10000	16	CLC2OUT
01111	15	CLC1OUT	01111	15	CLC1OUT
01110	14	NCO1OUT	01110	14	NCO1OUT
01101-01100	13-12	Reserved	01101-01100	13-12	Reserved
01011	11	PWM8 OUT	01011	11	PWM8 OUT
01010	10	PWM7 OUT	01010	10	PWM7 OUT

33.2.2 UART 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 pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 4 or 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 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 UART receiver. The FIFO registers and RSR are not directly accessible by software. Access to the received data is via the UxRXB register.

33.2.2.1 Enabling the Receiver

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

- RXEN = 1
- MODE<3:0> = 0h through 3h
- UxBRGH:L = desired baud rate
- RXPPS = code for desired input pin
- Input pin ANSEL bit = 0
- ON = 1

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

Setting the RXEN bit in the UxCON0 register enables the receiver circuitry of the UART. Setting the MODE<3:0> bits in the UxCON0 register configures the UART for the desired asynchronous mode. Setting the ON bit in the UxCON1 register enables the UART. The TRIS bit corresponding to the selected RX I/O pin must be set to configure the pin as an input.

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

33.2.2.2 Receiving Data

Data is recovered from the bit stream by timing to the center of the bits and sampling the input level. In High-Speed mode, there are four BRG clocks per bit and only one sample is taken per bit. In Normal-Speed mode, there are 16 BRG clocks per bit and three samples are taken per bit.

The receiver data recovery circuit initiates character reception on the falling edge of the Start bit. The Start bit, is always a '0'. The Start bit is qualified in the middle of the bit. In Normal-Speed mode only, the Start bit is also qualified at the leading edge of the bit. The following paragraphs describe the majority detect sampling of Normal-Speed mode.

The falling edge starts the baud rate generator (BRG) clock. The input is sampled at the first and second BRG clocks.

If both samples are high then the falling edge is deemed a glitch and the UART returns to the Start bit detection state without generating an error.

If either sample is low, the data recovery circuit continues counting BRG clocks and takes samples at clock counts 7, 8, and 9. When less than two samples are low, the Start bit is deemed invalid and the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit.

When two or more samples are low, the Start bit is deemed valid and the data recovery continues. After a valid Start bit is detected, the BRG clock counter continues and resets at count 16. This is the beginning of the first data bit.

The data recovery circuit counts BRG clocks from the beginning of the bit and takes samples at clocks 7, 8, and 9. The bit value is determined from the majority of the samples. The resulting '0' or '1' is shifted into the RSR.The BRG clock counter continues and resets at count 16. This sequence repeats until all data bits have been sampled and shifted into the RSR.

After all data bits have been shifted in, the first Stop bit is sampled. Stop bits are always a '1'. If the bit sampling determines that a '0' is in the Stop bit position, the framing error is set for this character. Otherwise, the framing error is cleared for this character. See **Section 33.2.2.4 "Receive Framing Error"** for more information on framing errors.

33.2.2.3 Receive Interrupts

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the UART receive FIFO. The UxRXIF interrupt flag in the respective PIR register is set at this time, provided it is not being suppressed.

The UxRXIF is suppressed by any of the following:

- FERIF if FERIE is set
- PERIF if PERIE is set

This suspends DMA transfer of data until software processes the error and reads UxRXB to advance the FIFO beyond the error.

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

- UxRXIE, Interrupt Enable bit in the PIE register
- GIE, Global Interrupt Enable bits in the INTCON0 register

The UxRXIF interrupt flag bit will be set when not suppressed and there is an unread character in the FIFO, regardless of the state of interrupt enable bits. Reading the UxRXB register will transfer the top character out of the FIFO and reduce the FIFO contents by one. The UxRXIF interrupt flag bit is read-only, it cannot be set or cleared by software.

33.4 DMX Mode (UART1 only)

DMX is a protocol used in stage and show equipment. This includes lighting, fog machines, motors, etc. The protocol consists of a control console that sends out commands, and equipment such as theater lights that receive these commands. The protocol is one-way: The console transmits commands and the equipment receives them. Equipment never transmits anything and consoles never receive anything. Also there are no error conditions or re-transmit mechanisms.

DMX, or DMX512 as it is known, consists of a "Universe" of 512 channels. This means that one console can output up to 512 bytes on a single DMX link. Each equipment on the line is programmed to listen to a consecutive sequence of one or more of these bytes.

For example, a fog machine connected to one of the universes may be programmed to receive 4 bytes, starting at byte number 10, and a lighting unit may be programmed to receive one byte starting at byte number 22.

33.4.1 DMX CONSOLE TRANSMITTER

DMX Console mode is configured with the following settings:

- MODE<3:0> = 1010
- TXEN = 1
- RXEN = 0
- TXPOL = 1
- UxP1 = One less than the number of bytes to transmit (excluding the Start code)
- UxBRGH:L = Value to achieve 250K baud rate
- STP<1:0> = 10 for 2 Stop bits
- RxyPPS = TX pin output code
- ON = 1

Each DMX transmission begins with a Break followed by a byte called the 'Start Code'. The width of the BREAK is fixed at 25 bit times. The Break is followed by a "Make After Break" (MAB) idle period. After this idle period, the 1st through 'n'th byte is transmitted, where 'n-1' is the value in UxP1. See Figure 33-6.

Software sends the Start Code and the 'n' data bytes by writing the UxTXB register with each byte to be sent in the desired order. A UxTXIF value of '1' indicates when the UxTXB is ready to accept the next byte.

The internal byte counter is not accessible to software. Software needs to keep track of the number of bytes written to UxTXB to ensure that no more and no less than 'n' bytes are sent because the DMX state machine will automatically insert a Break and reset its internal counter after 'n' bytes are written. One way to ensure synchronization between hardware and software is to toggle TXEN after the last byte of the universe is completely free of the transmit shift register as indicated by the TXMTIF bit.

33.4.2 DMX EQUIPMENT RECEIVER

DMX Equipment mode is configured with the following settings:

- MODE<3:0> = 1010
- TXEN = 0
- RXEN = 1
- RXPOL = 1
- UxP2 = number of first byte to receive
- UxP3 = number of last byte to receive
- UxBRGH:L = Value to achieve 250K baud rate
- STP<1:0> = 10 for 2 Stop bits
- ON = 1
- UxRXPPS = code for desired input pin
- Input pin ANSEL bit = 0

When configured as DMX Equipment, the UART listens for a Break character that is at least 23 bit periods wide. If the Break is shorter than 23 bit times, the Break is ignored and the DMX state machine remains in Idle mode. Upon receiving the Break, the DMX counters will be reset to align with the incoming data stream. Immediately after the Break, the UART will see the "Make after Break" (MAB). This space is ignored by the UART. The Start Code follows the MAB and will always be stored in the receive FIFO.

After the Start Code, the 1st through 512th byte will be received, but not all of them are stored in the receive FIFO. The UART ignores all received bytes until the ones of interest are received. This is done using the UxP2 and UxP3 registers. The UxP2 register holds the value of the byte number to start the receive process. The byte counter starts at 0 for the first byte after the Start Code. For example, to receive 4 bytes starting at the 10th byte after the Start Code, write 009h (9 decimal) to UxP2H:L and 00Ch (12 decimal) to UxP3H:L. The receive FIFO is only 2 bytes deep, therefore the bytes must be retrieved by reading UxRXB as they come in to avoid a receive FIFO overrun condition.

Typically two Stop bits are inserted between bytes. If either Stop bit is detected as a '0' then the framing error for that byte will be set.

Since the DMX sequence always starts with a Break, the software can verify that it is in sync with the sequence by monitoring the RXBKIF flag to ensure that the next byte received after the RXBKIF is processed as the Start Code and subsequent bytes are processed as the expected data.

33.12.3 XON/XOFF FLOW CONTROL

XON/XOFF flow control is selected by setting the FLO<1:0> bits to '01'.

XON/XOFF is a data based flow control method. The signals to suspend and resume transmission are special characters sent by the receiver to the transmitter The advantage is that additional hardware lines are not needed.

XON/XOFF flow control requires full duplex operation because the transmitter must be able to receive the signal to suspend transmitting while the transmission is in progress. Although XON and XOFF are not defined in the ASCII code, the generally accepted values are 13h for XOFF and 11h for XON. The UART uses those codes.

The transmitter defaults to XON, or transmitter enabled. This state is also indicated by the read-only XON bit in the UxFIFO register.

When an XOFF character is received, the transmitter stops transmitting after completing the character actively being transmitted. The transmitter remains disabled until an XON character is received.

XON will be forced on when software toggles the TXEN bit.

When the RUNOVF bit in the UxCON2 register is set then XON and XOFF characters continue to be received and processed without the need to clear the input FIFO by reading the UxRXB. However, if the RUNOVF bit is clear then the UxRXB must be read to avoid a receive overflow which will suspend flow control when the receive buffer overflows.

REGISTER 38-22: ADPREVH: ADC PREVIOUS RESULT REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
PREV<15:8>							
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	

bit 7-0 **PREV<15:8>**: Previous ADC Results bits
If ADPSIS = 1:
Upper byte of FLTR at the start of current ADC conversion
If ADPSIS = 0:
Upper bits of ADRES at the start of current ADC conversion(1)

Note 1: If ADPSIS = 0, ADPREVH and ADPREVL are formatted the same way as ADRES is, depending on the FM bit.

REGISTER 38-23: ADPREVL: ADC PREVIOUS RESULT REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
PREV<7:0>							
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	

bit 7-0 **PREV<7:0>**: Previous ADC Results bits
If ADPSIS = 1:
Lower byte of FLTR at the start of current ADC conversion
If ADPSIS = 0:
Lower bits of ADRES at the start of current ADC conversion(1)

Note 1: If ADPSIS = 0, ADPREVH and ADPREVL are formatted the same way as ADRES is, depending on the FM bit.

REGISTER 41-2: HLVDCON0: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER 0

R/W-0/0	U-0	R-x	R-x	U-0	U-0	R/W-0/0	R/W-0/0
EN	—	OUT	RDY	—	—	INTH	INTL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **EN:** High/Low-voltage Detect Power Enable bit
- 1 = Enables HLVD, powers up HLVD circuit and supporting reference circuitry
- 0 = Disables HLVD, powers down HLVD and supporting circuitry

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

bit 5 **OUT:** HLVD Comparator Output bit
- 1 = Voltage ≤ selected detection limit (HLVDL<3:0>)
- 0 = Voltage ≥ selected detection limit (HLVDL<3:0>)

bit 4 **RDY:** Band Gap Reference Voltages Stable Status Flag bit
- 1 = Indicates HLVD Module is ready and output is stable
- 0 = Indicates HLVD Module is not ready

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

bit 1 **INTH:** HLVD Positive going (High Voltage) Interrupt Enable
- 1 = HLVDIF will be set when voltage ≥ selected detection limit (SEL<3:0>)
- 0 = HLVDIF will not be set

bit 0 **INTL:** HLVD Negative going (Low Voltage) Interrupt Enable
- 1 = HLVDIF will be set when voltage ≤ selected detection limit (SEL<3:0>)
- 0 = HLVDIF will not be set

TABLE 41-2: SUMMARY OF REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
HLVDCON0	EN	—	OUT	RDY	—	—	INTH	INTL	662
HLVDCON1	—	—	—	—	SEL<3:0>				661

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the HLVD module.

CPFSLT **Compare f with W, skip if f < W**

Syntax: CPFSLT f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```

Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1] then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>;

If [W<7:4> + DC > 9] or [C = 1] then
(W<7:4>) + 6 + DC → W<7:4> ;
else
(W<7:4>) + DC → W<7:4>

Status Affected: C

Encoding:

0000	0000	0000	0111

Description: DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register W	Process Data	Write W

Example1:
```
DAW
```
Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

Example 2:
Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

43.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set of the PIC18(L)F2x/4xK42 family of devices. This includes the MPLAB C18 C compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

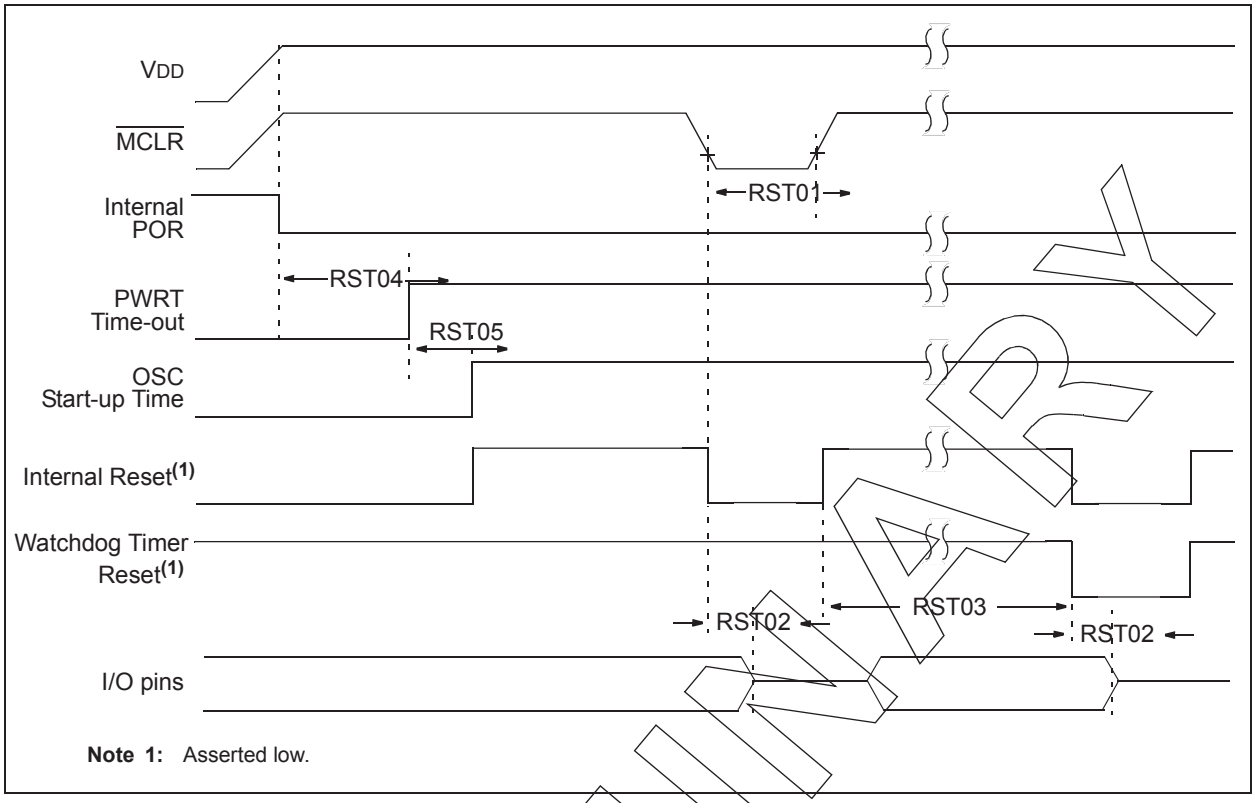
When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing mode. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option, or dialog box within the environment, that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

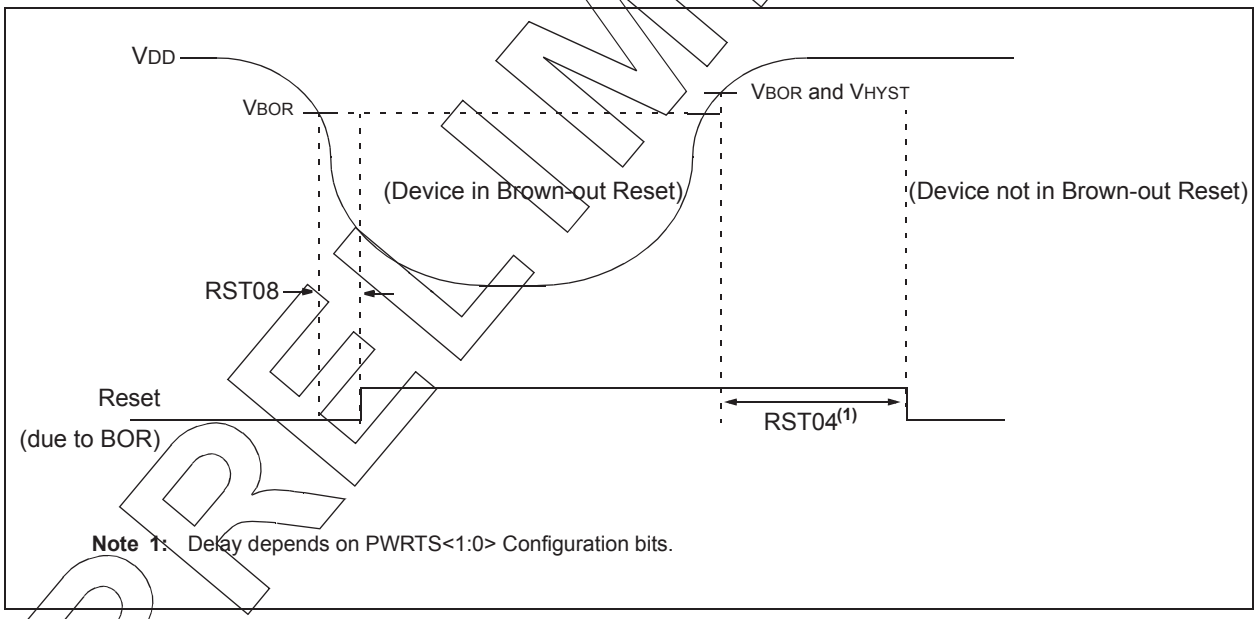
These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

FIGURE 46-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 46-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



Note 1: Delay depends on PWRTS<1:0> Configuration bits.