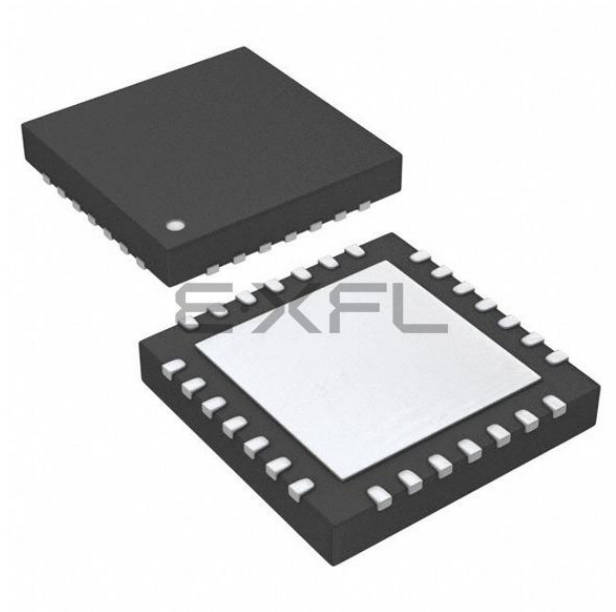


Welcome to E-XFL.COM

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.

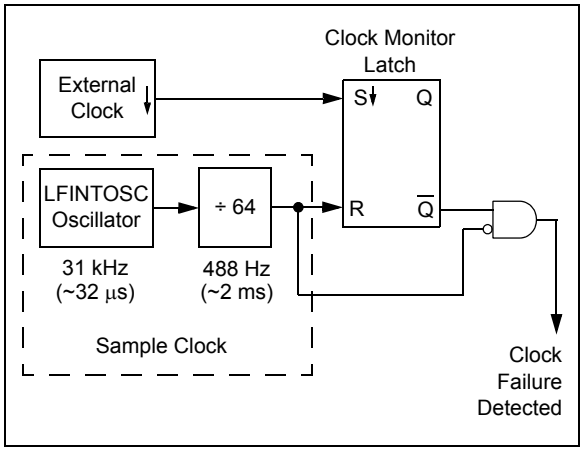
Applications of "Embedded - Microcontrollers"

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	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
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-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f27k42-e-ml

7.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (LP, XT, HS, ECL/M/H and Secondary Oscillator).

FIGURE 7-9: FSCM BLOCK DIAGRAM



7.4.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 7-9. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

7.4.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM overwrites the COSC bits to select HFINTOSC (`3'b110`). The frequency of HFINTOSC would be determined by the previous state of the FRQ bits and the NDIV/CDIV bits. The bit flag OSFIF of the respective PIR register is set. Setting this flag will generate an interrupt if the OSFIE bit of the respective PIR register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation, by writing to the NOSC and NDIV bits of the OSCCON1 register.

7.4.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or changing the NOSC and NDIV bits of the OSCCON1 register. When switching to the external oscillator or PLL, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON1. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSCFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSCFIF flag will again become set by hardware.

9.8 Interrupt Setup Procedure

1. When using interrupt priority levels, set the IPEN bit in INTCON0 register and then select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPRx Control register.

Note:	At a device Reset, the IPRx registers are initialized, such that all user interrupt sources are assigned to high priority.

2. Clear the Interrupt Flag Status bit associated with the peripheral in the associated PIRx Status register.
3. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate PIEx Control register.
4. If the vector table is used (MVECEN = 1), then setup the start address for the Interrupt Vector Table using the IVTBASE register. See **Section 9.2.2 "Interrupt Vector Table Contents"**.
5. Once the IVTBASE is written to, set the Interrupt enable bits in INTCON0 register.
6. An example of setting up interrupts and ISRs using assembly and C can be found in Examples 9-3 and 9-4.

9.9 External Interrupt Pins

The PIC18(L)F26/27/45/46/47/55/56/57K42 devices have three external interrupt sources which can be assigned to any pin on different ports based on the PPS settings. Refer **Section 17.0 "Peripheral Pin Select (PPS) Module"** for possible rerouting options. The external interrupt sources are edge-triggered. If the corresponding INTxEDG bit in the INTCON0 register is set (= 1), the interrupt is triggered by a rising edge. If the bit is clear, the trigger is on the falling edge.

When a valid edge appears on the INTx pin, the corresponding flag bit, INTxF in the PIRx registers, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxE. Flag bit, INTxF, must be cleared by software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1 and INT2) can wake-up the processor from Idle or Sleep modes if bit INTxE was set prior to going into those modes. If the Global Interrupt Enable bit, GIE/GIEH, is set, the processor will branch to the interrupt vector following wake-up. Interrupt priority is determined by the value contained in the interrupt priority bits, INT0IP, INT1IP and INT2IP of the IPRx registers.

9.10 Wake-up from Sleep

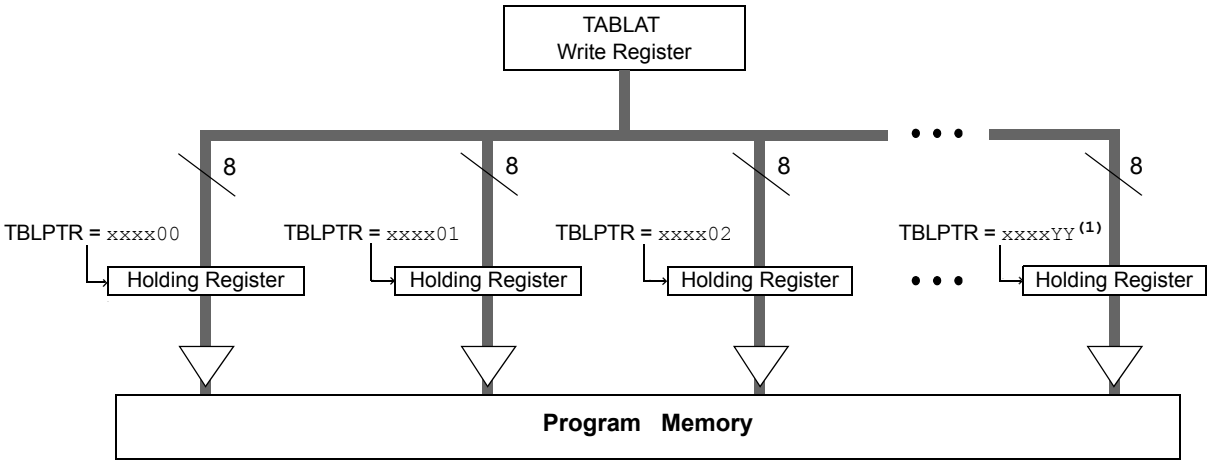
The interrupt controller provides a wake-up request to the CPU whenever an interrupt event occurs, if the interrupt event is enabled. This occurs regardless of whether the part is in Run, Idle/Doze or Sleep modes. The status of the GIEH/GIEL bits has no effect on the wake-up request. The wake-up request will be asynchronous to all clocks.

9.11 Interrupt Compatibility

When the MVECEN bit in Configuration Word 2L is cleared (Register 5-3), the Interrupt Vector Table feature is disabled and interrupts are compatible with previous high performance 8-bit PIC18 microcontroller devices. In this mode, the Interrupt Vector Table priority has no effect.

When the IPEN bit is also cleared, the interrupt priority feature is disabled and interrupts are compatible with PIC®16 microcontroller mid-range devices. All interrupts branch to address 0008h since the interrupt priority is disabled.

FIGURE 13-8: TABLE WRITES TO PROGRAM FLASH MEMORY



Note 1: Refer to Table 5-4 for number of holding registers (e.g., YY = 3F for 64 holding registers).

13.1.6.1 Program Flash Memory Write Sequence

The sequence of events for programming an internal program memory location should be:

1. Read appropriate number of bytes into RAM. Refer to Table 5-4 for Write latch size.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the block erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the n-byte block into the holding registers with auto-increment. Refer to Table 5-4 for Write latch size.
7. Set REG<1:0> bits to point to program memory.
8. Clear FREE bit and set WREN bit in NVMCON1 register.
9. Disable interrupts.
10. Execute the unlock sequence (see **Section 13.1.4 "NVM Unlock Sequence"**).
11. WR bit is set in NVMCON1 register.
12. The CPU will stall for the duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update each write block of memory. An example of the required code is given in Example 13-4.

Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the bytes in the holding registers.

14.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the program memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 14-1: CRC EXAMPLE

Rev. 10-000206A
1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^2 + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010-
```
(1)

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:
SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: 0x6BA2

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always '1' and will always be treated as a '1' by the CRC for calculating the CRC check value. This bit will be read in software as a '0'.

22.7 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 22-2. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

TABLE 22-2: OPERATING MODES

Peripheral	Bit Name Prefix
Timer2	T2
Timer4	T4
Timer6	T6

REGISTER 22-1: TxCLK: TIMERx CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CS<3:0>			
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 7-4 **Unimplemented:** Read as '0'

bit 3-0 **CS<3:0>:** Timerx Clock Selection bits

CS<3:0>	T2TMR	TMR4	TMR6
	Clock Source	Clock Source	Clock Source
1111	Reserved	Reserved	Reserved
1110	CLC4_out	CLC4_out	CLC4_out
1101	CLC3_out	CLC3_out	CLC3_out
1100	CLC2_out	CLC2_out	CLC2_out
1011	CLC1_out	CLC1_out	CLC1_out
1010	ZCD_OUT	ZCD_OUT	ZCD_OUT
1001	NCO1OUT	NCO1OUT	NCO1OUT
1000	CLKREF_OUT	CLKREF_OUT	CLKREF_OUT
0111	SOSC	SOSC	SOSC
0110	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)
0101	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)
0100	LFINTOSC	LFINTOSC	LFINTOSC
0011	HFINTOSC	HFINTOSC	HFINTOSC
0010	Fosc	Fosc	Fosc
0001	Fosc/4	Fosc/4	Fosc/4
0000	Pin selected by T2INPPS	Pin selected by T4INPPS	Pin selected by T6INPPS

REGISTER 22-6: TxHLT: TIMERx HARDWARE LIMIT CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PSYNC	CKPOL	CKSYNC	MODE<4:0>				
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 **PSYNC:** Timerx Prescaler Synchronization Enable bit(1, 2)
1 = TxTMR Prescaler Output is synchronized to Fosc/4
0 = TxTMR Prescaler Output is not synchronized to Fosc/4

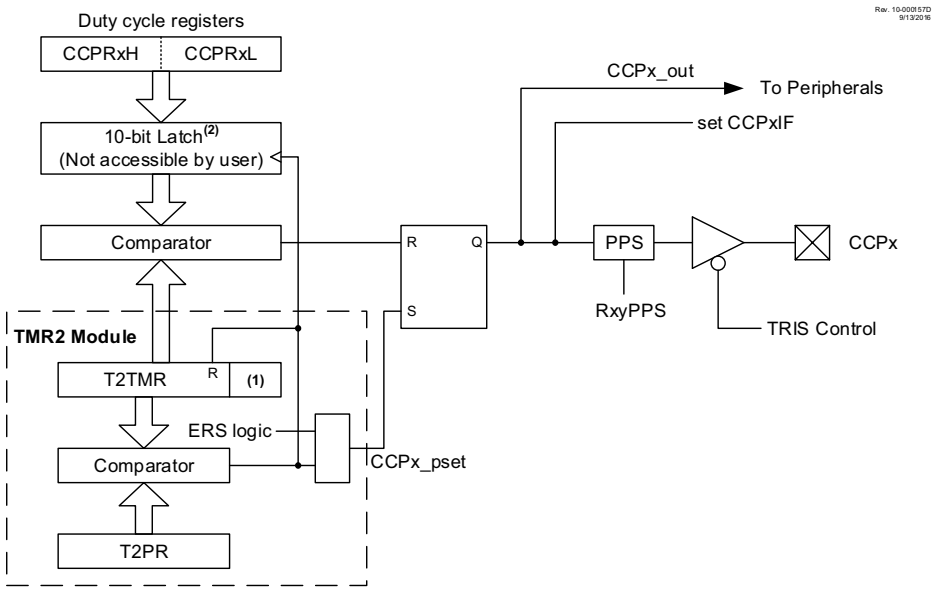
bit 6 **CKPOL:** Timerx Clock Polarity Selection bit(3)
1 = Falling edge of input clock clocks timer/prescaler
0 = Rising edge of input clock clocks timer/prescaler

bit 5 **CKSYNC:** Timerx Clock Synchronization Enable bit(4, 5)
1 = ON register bit is synchronized to T2TMR_clk input
0 = ON register bit is not synchronized to T2TMR_clk input

bit 4-0 **MODE<4:0>:** Timerx Control Mode Selection bits(6, 7)
See Table 22-1 for all operating modes.

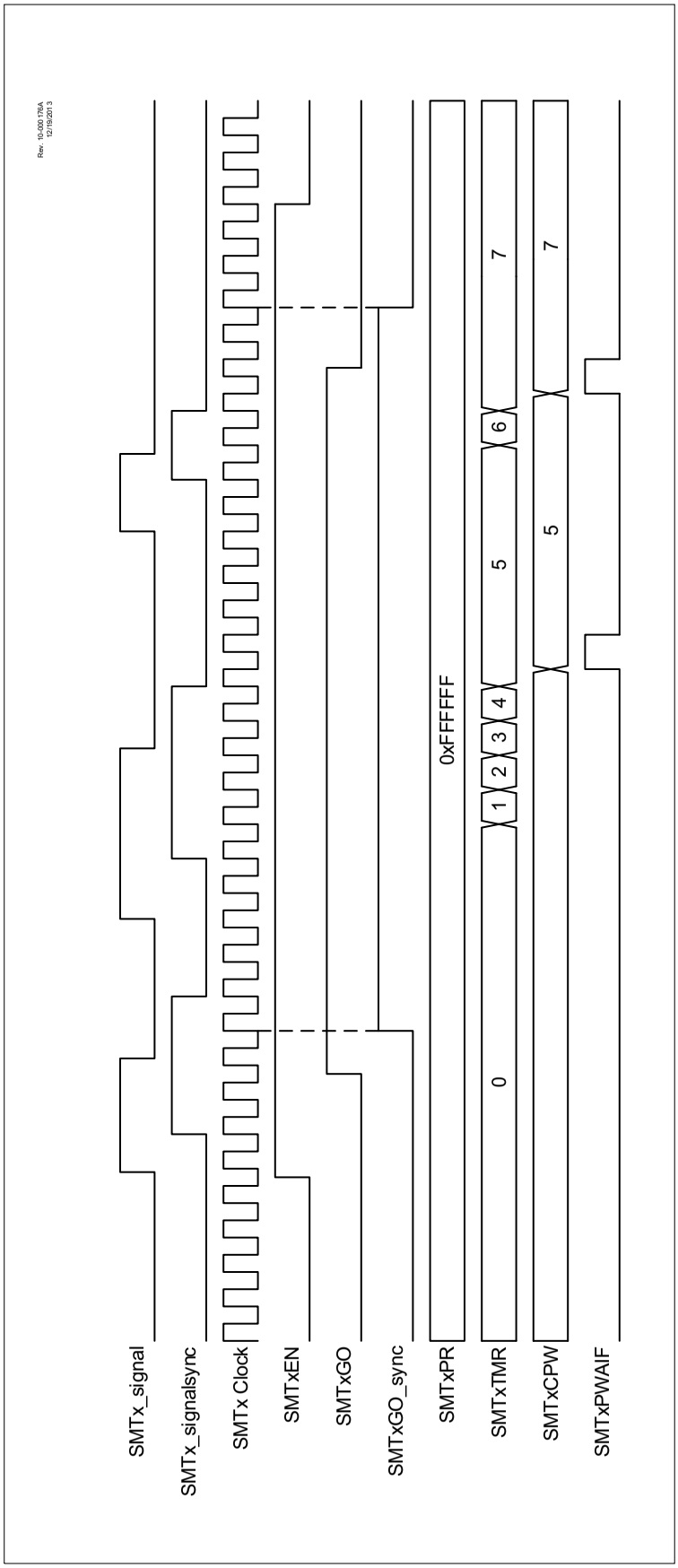
Note 1: Setting this bit ensures that reading TxTMR will return a valid data value.
2: When this bit is '1', Timer2 cannot operate in Sleep mode.
3: CKPOL should not be changed while ON = 1.
4: Setting this bit ensures glitch-free operation when the ON is enabled or disabled.
5: When this bit is set then the timer operation will be delayed by two TxTMR input clocks after the ON bit is set.
6: Unless otherwise indicated, all modes start upon ON = 1 and stop upon ON = 0 (stops occur without affecting the value of TxTMR).
7: When TxTMR = TxPR, the next clock clears TxTMR, regardless of the operating mode.

FIGURE 23-4: SIMPLIFIED PWM BLOCK DIAGRAM



Notes: 1. 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time-base.
2. The alignment of the 10 bits from the CCPR register is determined by the CCPxFMT bit.

FIGURE 25-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM



REGISTER 25-4: SMT1CLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<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 7-3 **Unimplemented**: Read as '0'

bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits

111 = Reference Clock Output
110 = SOSC
101 = MFINTOSC/16 (32 kHz)
100 = MFINTOSC (500 kHz)
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = FOSC
000 = FOSC/4

REGISTER 26-5: CWGxSTR(1): CWG STEERING CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD(2)	STRC(2)	STRB(2)	STRA(2)
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 **OVRD:** Steering Data D bit

bit 6 **OVRC:** Steering Data C bit

bit 5 **OVRB:** Steering Data B bit

bit 4 **OVRA:** Steering Data A bit

bit 3 **STRD:** Steering Enable bit D(2)
- 1 = CWGxD output has the CWG data input waveform with polarity control from POLD bit
- 0 = CWGxD output is assigned to value of OVRD bit

bit 2 **STRC:** Steering Enable bit C(2)
- 1 = CWGxC output has the CWG data input waveform with polarity control from POLC bit
- 0 = CWGxC output is assigned to value of OVRC bit

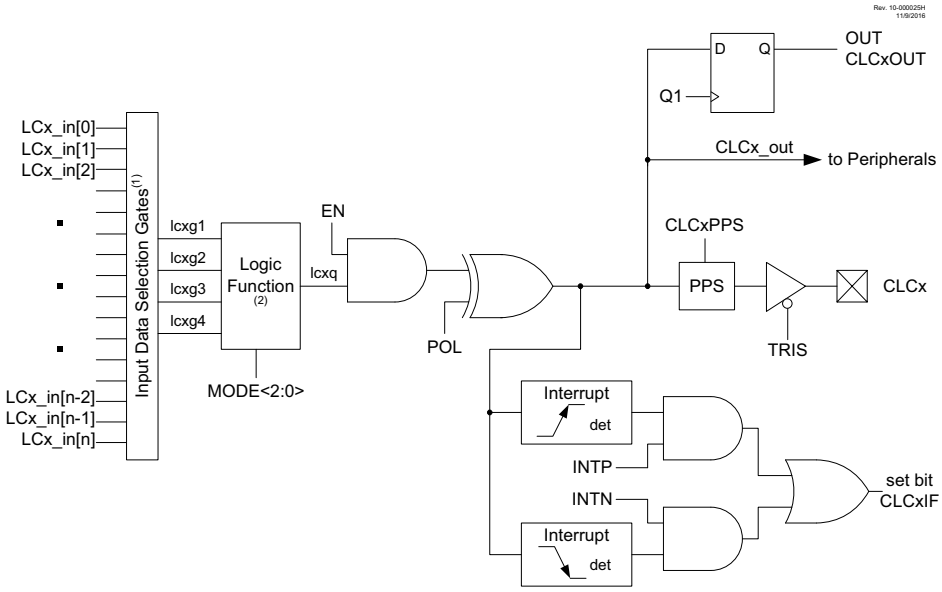
bit 1 **STRB:** Steering Enable bit B(2)
- 1 = CWGxB output has the CWG data input waveform with polarity control from POLB bit
- 0 = CWGxB output is assigned to value of OVRB bit

bit 0 **STRA:** Steering Enable bit A(2)
- 1 = CWGxA output has the CWG data input waveform with polarity control from POLA bit
- 0 = CWGxA output is assigned to value of OVRA bit

Note 1: The bits in this register apply only when MODE<2:0> = 00x (Register 26-1, Steering modes).

2: This bit is double-buffered when MODE<2:0> = 001.

FIGURE 27-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 27-2: Input Data Selection and Gating

2: See Figure 27-3: Programmable Logic Functions.

27.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

27.1.1 DATA SELECTION

There are 32 signals available as inputs to the configurable logic. Four 32-input multiplexers are used to select the inputs to pass on to the next stage.

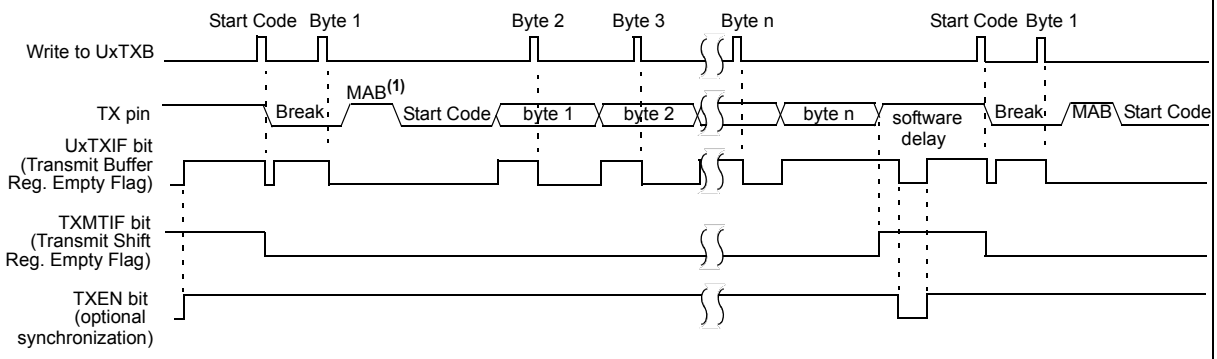
Data selection is through four multiplexers as indicated on the left side of Figure 27-2. Data inputs in the figure are identified by a generic numbered input name.

Table 27-1 correlates the generic input name to the actual signal for each CLC module. The column labeled 'DyS<4:0> Value' indicates the MUX selection code for the selected data input. DyS is an abbreviation for the MUX select input codes: D1S<4:0> through D4S<4:0>.

Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 27-3 through Register 27-6).

Note: Data selections are undefined at power-up.

FIGURE 31-6: DMX TRANSMIT SEQUENCE



Note 1: The MAB period is fixed at 3-bits period.

31.5 LIN Modes (UART1 only)

LIN is a protocol used primarily in automotive applications. The LIN network consists of two kinds of software processes: a Master process and a Slave process. Each network has only one Master process and one or more Slave processes.

From a physical layer point of view, the UART on one processor may be driven by both a Master and a Slave process, as long as only one Master process exists on the network.

A LIN transaction consists of a Master process followed by a Slave process. The Slave process may involve more than one Slave where one is transmitting and the other(s) are receiving. The transaction begins by the following Master process transmission sequence:

1. Break
2. Delimiter bit
3. Sync Field
4. PID byte

The PID determines which Slave processes are expected to respond to the Master. When the PID byte is complete, the TX output remains in the Idle state. One or more of the Slave processes may respond to the Master process. If no one responds within the inter-byte period, the Master is free to start another transmission. The inter-byte period is timed by software using a means other than the UART.

The Slave process follows the Master process. When the Slave software recognizes the PID then that Slave process responds by either transmitting the required response or by receiving the transmitted data. Only Slave processes send data. Therefore, Slave processes receiving data are receiving that of another Slave process.

When a Slave sends data, the Slave UART automatically calculates the checksum for the transmitted bytes as they are sent and appends the inverted checksum byte to the slave response.

When a Slave receives data, the checksum is accumulated on each byte as it is received using the same algorithm as the sending process. The last byte, which is the inverted checksum value calculated by the sending process, is added to the locally calculated checksum by the UART. The check passes when the result is all '1's, otherwise the check fails and the CERIF bit is set.

Two methods for computing the checksum are available: legacy and enhanced. The legacy checksum includes only the data bytes. The enhanced checksum includes the PID and the data. The C0EN control bit in the UxCON2 register determines the checksum method. Setting C0EN to '1' selects the enhanced method. Software must select the appropriate method before the Start bit of the checksum byte is received.

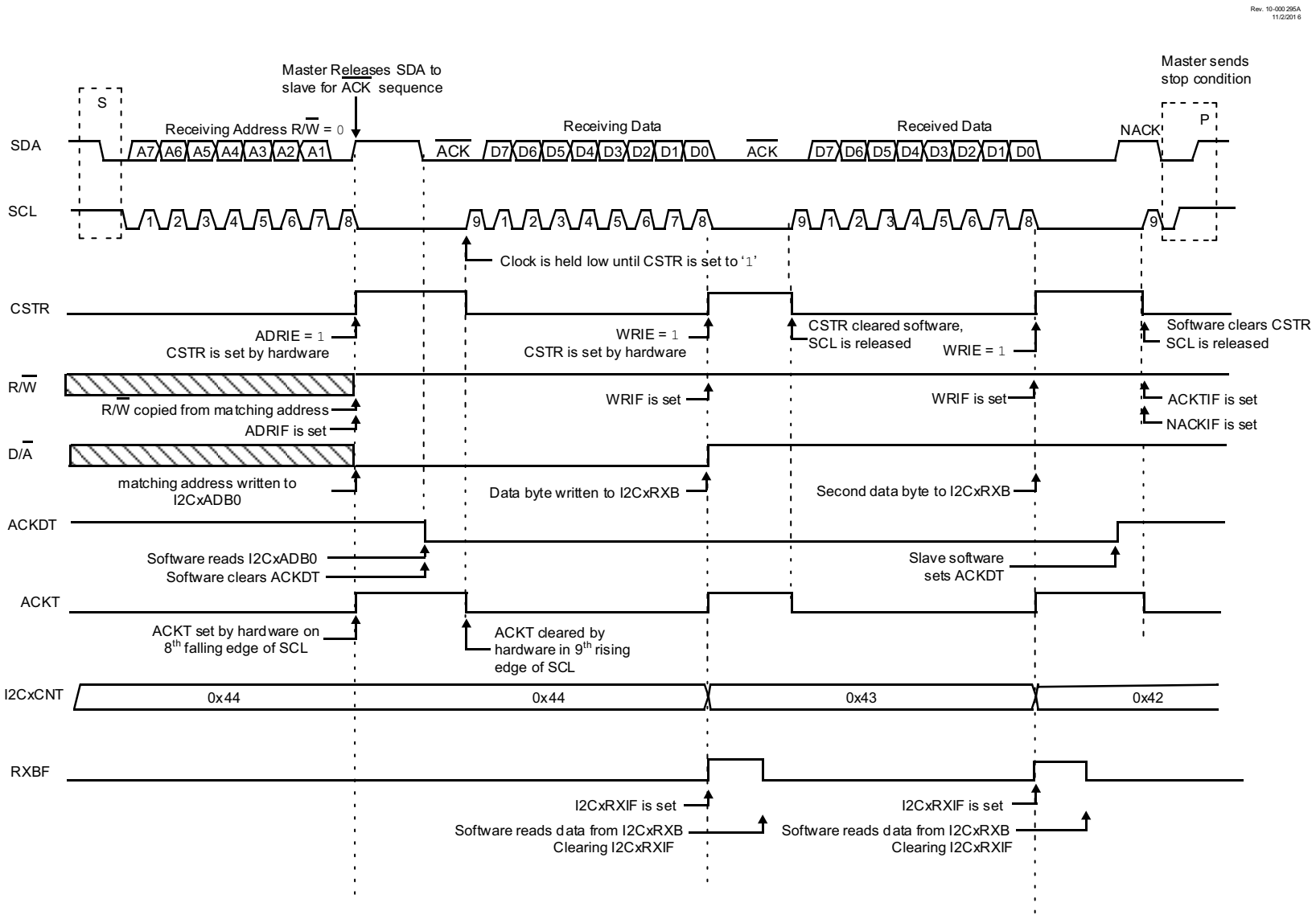
31.5.1 LIN MASTER/SLAVE MODE

The LIN Master mode includes capabilities to generate Slave processes. The Master process stops at the PID transmission. Any data that is transmitted in Master/ Slave mode is done as a Slave process. LIN Master/ Slave mode is configured by the following settings:

- MODE<3:0> = 1100
- TXEN = 1
- RXEN = 1
- UxBRGH:L = Value to achieve desired baud rate
- TXPOL = 0 (for high Idle state)
- STP = desired Stop bits selection
- C0EN = desired checksum mode
- RxyPPS = TX pin selection code
- TX pin TRIS control = 0
- ON = 1

Note:	The TXEN bit must be set before the Master process is received and remain set while in LIN mode whether or not the slave process is a transmitter.

FIGURE 33-8: I²C SLAVE, 7-BIT ADDRESS, RECEPTION NO I2CxCNT (ACKTIE = 0, ADRIE = 1, WRIE = 1)



39.10 Register Definitions: HLVD Control

Long bit name prefixes for the HLVD peripheral is shown in Table 39-1. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

TABLE 39-1:

Peripheral	Bit Name Prefix
HLVD	HLVD

REGISTER 39-1: 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

REGISTER 39-2: HLVDCON1: LOW-VOLTAGE DETECT CONTROL REGISTER 1

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	SEL<3:0>			
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 = Bit is unchanged

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **SEL<3:0>:** High/Low Voltage Detection Limit Selection bits
Refer to Table 44-14 for voltage detection limits.

TABLE 39-2: 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	657
HLVDCON1	—	—	—	—	SEL<3:0>				658

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

ANDWF **AND W with f**

Syntax: ANDWF f {,d {,a}}

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

0001	01da	ffff	ffff

Description: The contents of W are AND'ed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `ANDWF REG, 0, 0`

Before Instruction
W = 17h
REG = C2h
After Instruction
W = 02h
REG = C2h

BC **Branch if Carry**

Syntax: BC n

Operands: $-128 \le n \le 127$

Operation: if CARRY bit is '1'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0010	nnnn	nnnn

Description: If the CARRY bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example:
```
HERE    BC   5
```
Before Instruction
PC = address (HERE)
After Instruction
If CARRY = 1;
PC = address (HERE + 12)
If CARRY = 0;
PC = address (HERE + 2)

RRNCF **Rotate Right f (No Carry)**

Syntax: RRNCF f {,d {,a}}

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

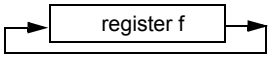
Operation: (f<n>) → dest<n – 1>,
(f<0>) → dest<7>

Status Affected: N, Z

Encoding:

0100	00da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
If 'a' is '0', the Access Bank will be selected (default), overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: RRNCF REG, 1, 0

Before Instruction
REG = 1101 0111
After Instruction
REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction
W = ?
REG = 1101 0111
After Instruction
W = 1110 1011
REG = 1101 0111

SETF **Set f**

Syntax: SETF f {,a}

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

Operation: FFh → f

Status Affected: None

Encoding:

0110	100a	ffff	ffff

Description: The contents of the specified register are set to FFh.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: SETF REG, 1

Before Instruction
REG = 5Ah
After Instruction
REG = FFh

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3F66h	PWM7CON	EN	—	OUT	POL	—	—	—	—	358
3F65h	PWM7DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	360
3F65h	PWM7DCH	DC								360
3F64h	PWM7DCL	DC1	DC0	—	—	—	—	—	—	360
3F64h	PWM7DCL	DC		—	—	—	—	—	—	360
3F63h	—	Unimplemented								
3F62h	PWM8CON	EN	—	OUT	POL	—	—	—	—	358
3F61h	PWM8DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	360
3F61h	PWM8DCH	DC								360
3F60h	PWM8DCL	DC1	DC0	—	—	—	—	—	—	360
3F60h	PWM8DCL	DC		—	—	—	—	—	—	360
3F5Fh	CCPTMRS1	P8TSEL		P7TSEL		P6TSEL		P5TSEL		359
3F5Eh	CCPTMRS0	C4TSEL		C3TSEL		C2TSEL		C1TSEL		359
3F5Dh - 3F5Bh	—	Unimplemented								
3F5Ah	CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F59h	CWG1AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F58h	CWG1AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F57h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F56h	CWG1CON0	EN	LD	—	—	—	MODE			424
3F55h	CWG1DBF	—	—	DBF						431
3F54h	CWG1DBR	—	—	DBR						431
3F53h	CWG1ISM	—	—	—	—	IS				427
3F52h	CWG1CLK	—	—	—	—	—	—	—	CS	426
3F51h	CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F50h	CWG2AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F4Fh	CWG2AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F4Eh	CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F4Dh	CWG2CON0	EN	LD	—	—	—	MODE			424
3F4Ch	CWG2DBF	—	—	DBF						431
3F4Bh	CWG2DBR	—	—	DBR						431
3F4Ah	CWG2ISM	—	—	—	—	IS				427
3F49h	CWG2CLK	—	—	—	—	—	—	—	CS	426
3F48h	CWG3STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F47h	CWG3AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F46h	CWG3AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F45h	CWG3CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F44h	CWG3CON0	EN	LD	—	—	—	MODE			424
3F43h	CWG3DBF	—	—	DBF						431
3F42h	CWG3DBR	—	—	DBR						431
3F41h	CWG3ISM	—	—	—	—	IS				427
3F40h	CWG3CLK	—	—	—	—	—	—	—	CS	426
3F3Fh	NCO1CLK	PWS			—	CKS				454
3F3Eh	NCO1CON	EN	—	OUT	POL	—	—	—	PFM	453
3F3Dh	NCO1INCU	INC								457
3F3Ch	NCO1INCH	INC								456
3F3Bh	NCO1INCL	INC								456

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

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

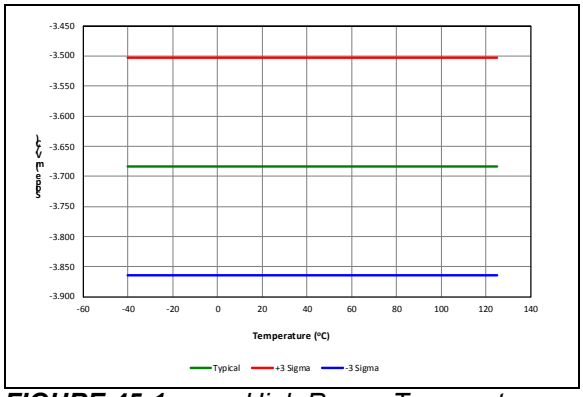


FIGURE 45-1: *High Range Temperature Indicator Voltage Sensitivity Across Temperature*

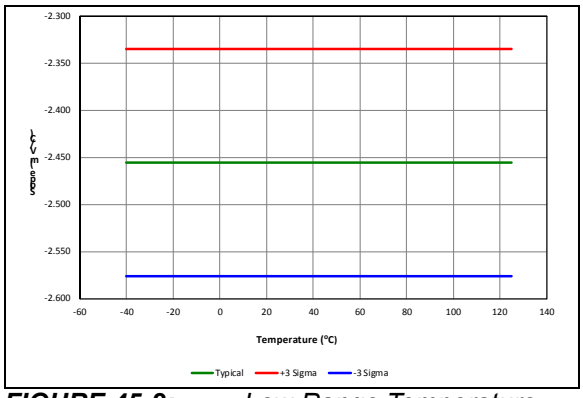


FIGURE 45-2: *Low Range Temperature Indicator Voltage Sensitivity Across Temperature*

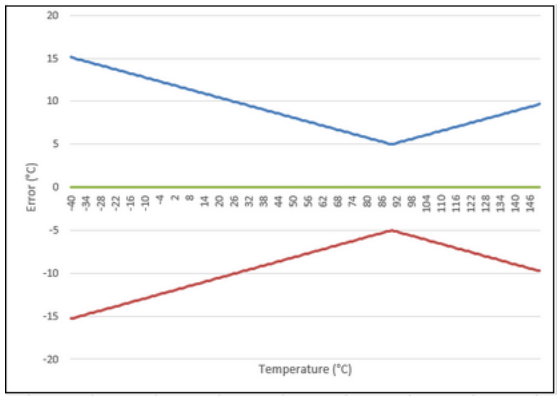
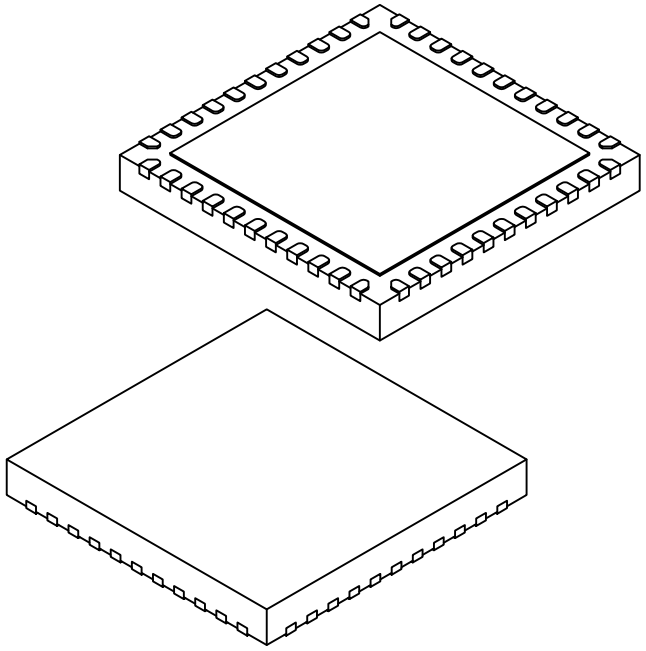


FIGURE 45-3: *Temperature Indicator Performance Over Temperature*

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

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 Pins	N		44	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		8.00 BSC	
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.25	6.45	6.60
Terminal Width	b	0.20	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
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 C04-103D Sheet 2 of 2