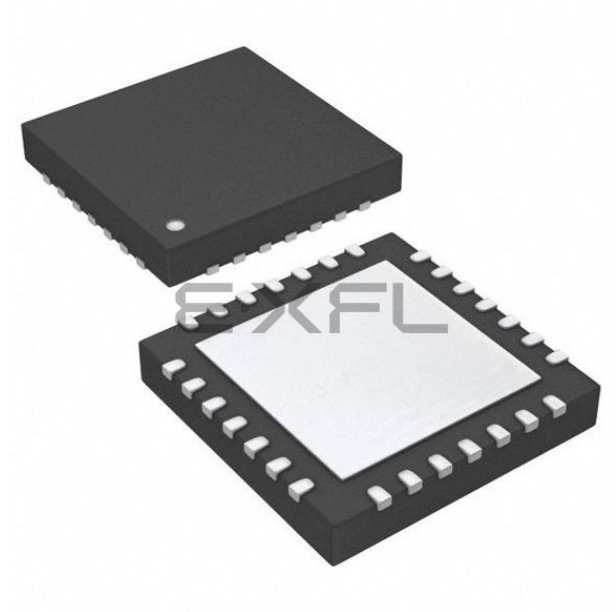


**Welcome to E-XFL.COM**

## 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, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 24 |
| Program Memory Size | 8KB (4K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 19x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°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/pic18f23k22t-i-ml |

**TABLE 2: PIC18(L)F2XK22 PIN SUMMARY**

| 28-SSOP, SOIC 28-SPDIP | 28-QFN, UQFN | I/O | Analog | Comparator | CTMU | SR Latch | Reference | (E)CCP | EUSART | MSSP | Timers | Interrupts | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 27 | RA0 | AN0 | C12IN0- | | | | | | | | | | |
| 3 | 28 | RA1 | AN1 | C12IN1- | | | | | | | | | | |
| 4 | 1 | RA2 | AN2 | C2IN+ | | | VREF-<br>DACOUT | | | | | | | |
| 5 | 2 | RA3 | AN3 | C1IN+ | | | VREF+ | | | | | | | |
| 6 | 3 | RA4 | | C1OUT | | SRQ | | CCP5 | | | T0CKI | | | |
| 7 | 4 | RA5 | AN4 | C2OUT | | SRNQ | HLVDIN | | | $\overline{SS}$1 | | | | |
| 10 | 7 | RA6 | | | | | | | | | | | | OSC2<br>CLKO |
| 9 | 6 | RA7 | | | | | | | | | | | | OSC1<br>CLKI |
| 21 | 18 | RB0 | AN12 | | | SRI | | CCP4<br>FLT0 | | $\overline{SS}$2 | | INT0 | Y | |
| 22 | 19 | RB1 | AN10 | C12IN3- | | | | P1C | | SCK2<br>SCL2 | | INT1 | Y | |
| 23 | 20 | RB2 | AN8 | | CTED1 | | | P1B | | SDI2<br>SDA2 | | INT2 | Y | |
| 24 | 21 | RB3 | AN9 | C12IN2- | CTED2 | | | CCP2<br>P2A(1) | | SDO2 | | | Y | |
| 25 | 22 | RB4 | AN11 | | | | | P1D | | | T5G | IOC | Y | |
| 26 | 23 | RB5 | AN13 | | | | | CCP3<br>P3A(4)<br>P2B(3) | | | T1G<br>T3CKI(2) | IOC | Y | |
| 27 | 24 | RB6 | | | | | | | TX2/CK2 | | | IOC | Y | PGC |
| 28 | 25 | RB7 | | | | | | | RX2/DT2 | | | IOC | Y | PGD |
| 11 | 8 | RC0 | | | | | | P2B(3) | | | SOSCO<br>T1CKI<br>T3CKI(2)<br>T3G | | | |
| 12 | 9 | RC1 | | | | | | CCP2<br>P2A(1) | | | SOSCI | | | |
| 13 | 10 | RC2 | AN14 | | CTPLS | | | CCP1<br>P1A | | | T5CKI | | | |
| 14 | 11 | RC3 | AN15 | | | | | | | SCK1<br>SCL1 | | | | |
| 15 | 12 | RC4 | AN16 | | | | | | | SDI1<br>SDA1 | | | | |
| 16 | 13 | RC5 | AN17 | | | | | | | SDO1 | | | | |
| 17 | 14 | RC6 | AN18 | | | | | CCP3<br>P3A(4) | TX1/CK1 | | | | | |
| 18 | 15 | RC7 | AN19 | | | | | P3B | RX1/DT1 | | | | | |
| 1 | 26 | RE3 | | | | | | | | | | | | $\overline{MCLR}$<br>VPP |
| 8, 19<br>19 | 5, 16<br>16 | VSS | | | | | | | | | | | | VSS |
| 20 | 17 | VDD | | | | | | | | | | | | VDD |

**Note 1:** CCP2/P2A multiplexed in fuses.
**2:** T3CKI multiplexed in fuses.
**3:** P2B multiplexed in fuses.
**4:** CCP3/P3A multiplexed in fuses.

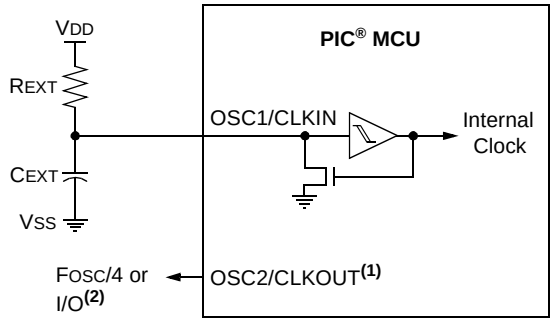
### 2.5.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

#### 2.5.4.1 RC Mode

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by four. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 2-8 shows the external RC mode connections.

**FIGURE 2-8: EXTERNAL RC MODES**




Recommended values: 10 kΩ ≤ REXT ≤ 100 kΩ
CEXT > 20 pF

**Note 1:** Alternate pin functions are listed in **Section 1.0 "Device Overview"**.
**2:** Output depends upon RC or RCIO clock mode.

#### 2.5.4.2 RCIO Mode

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes a general purpose I/O pin.

The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- input threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

## 2.6 Internal Clock Modes

The oscillator module has three independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory calibrated and operates at 31.25 kHz. The LFINTOSC cannot be user-adjusted, but is designed to be stable over temperature and voltage.

The system clock speed can be selected via software using the Internal Oscillator Frequency select bits IRCF<2:0> of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS<1:0>) bits of the OSCCON register. See **Section 2.11 "Clock Switching"** for more information.

### 2.6.1 INTOSC WITH I/O OR CLOCKOUT

Two of the clock modes selectable with the FOSC<3:0> bits of the CONFIG1H Configuration register configure the internal oscillator block as the primary oscillator. Mode selection determines whether the OSC2/CLKOUT pin will be configured as general purpose I/O or FOSC/4 (CLKOUT). In both modes, the OSC1/CLKIN pin is configured as general purpose I/O. See **Section 24.0 "Special Features of the CPU"** for more information.

The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

## 4.3 Master Clear ($\overline{MCLR}$)

The $\overline{MCLR}$ pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the $\overline{MCLR}$ Reset path which detects and ignores small pulses. An internal weak pull-up is enabled when the pin is configured as the $\overline{MCLR}$ input.

The $\overline{MCLR}$ pin is not driven low by any internal Resets, including the WDT.

In PIC18(L)F2X/4XK22 devices, the $\overline{MCLR}$ input can be disabled with the MCLRE Configuration bit. When $\overline{MCLR}$ is disabled, the pin becomes a digital input. See **Section 10.6 "PORTE Registers"** for more information.

## 4.4 Power-on Reset (POR)

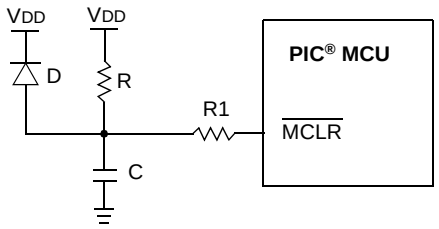
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry either leave the pin floating, or tie the $\overline{MCLR}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified. For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure proper operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{POR}$ bit of the RCON register. The state of the bit is set to '0' whenever a POR occurs; it does not change for any other Reset event. $\overline{POR}$ is not reset to '1' by any hardware event. To capture multiple events, the user must manually set the bit to '1' by software following any POR.

**FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**




**Note 1:** External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

**2:** 15 kΩ < R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

**3:** R1 ≥ 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C, in the event of $\overline{MCLR}$/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

## 7.6 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

The microcontroller itself can both read and write to the internal data EEPROM, regardless of the state of the code-protect Configuration bit. Refer to **Section 24.0 "Special Features of the CPU"** for additional information.

## 7.7 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (TPWRT).The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

## 7.8 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM without exceeding the total number of write cycles to a single byte. Refer to the Data EEPROM Memory parameters in **Section 27.0 "Electrical Specifications"** for write cycle limits. If this is the case, then an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

| **Note:** | If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification. |
|---|---|

**EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE**

```
        CLRF    EEADR               ; Start at address 0
        CLRF    EEADRH              ; if > 256 bytes EEPROM
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   0AAh                ;
        MOVWF   EECON2              ; Write 0AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     LOOP                ; Not zero, do it again
        INCFSZ  EEADRH, F           ; if > 256 bytes, Increment address
        BRA     LOOP                ; if > 256 bytes, Not zero, do it again

        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

**REGISTER 10-2: PORTE: PORTE REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-u/x | R/W-u/x | R/W-u/x | R/W-u/x |
|---|---|---|---|---|---|---|---|
| — | — | — | — | RE3(1) | RE2(2), (3) | RE1(2), (3) | RE0(2), (3) |
| bit 7 | | | | | | | bit 0 |

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

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

bit 3 **RE3:** PORTE Input bit value(1)

bit 2-0 **RE<2:0>:** PORTE I/O bit values(2), (3)

**Note 1:** Port is available as input only when MCLRE = 0.

**2:** Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

**3:** Available on PIC18(L)F4XK22 devices.

**REGISTER 10-3: ANSELA – PORTA ANALOG SELECT REGISTER**

| U-0 | U-0 | R/W-1 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | ANSA5 | — | ANSA3 | ANSA2 | ANSA1 | ANSA0 |
| 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 7-6 **Unimplemented:** Read as '0'

bit 5 **ANSA5:** RA5 Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

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

bit 3-0 **ANSA<3:0>:** RA<3:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

### 14.4.7 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

The CCPxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (PxA/PxC and PxB/PxD). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The PxA, PxB, PxC and PxD output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit.

The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMRxIF bit of the PIR1, PIR2 or PIR5 register being set as the second PWM period begins.

| | |
|---|---|
| **Note:** | When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s). |

**FIGURE 14-19: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRxSYNC = 0)**

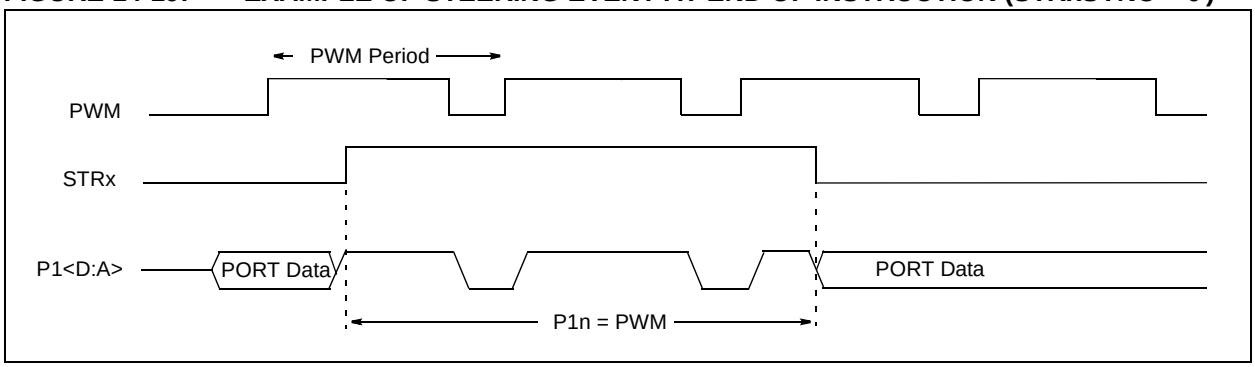



**FIGURE 14-20: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRxSYNC = 1)**

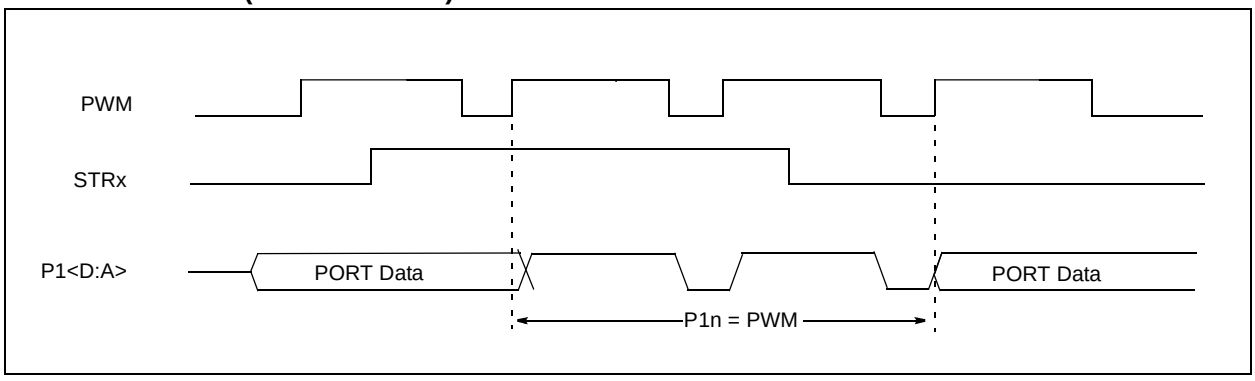
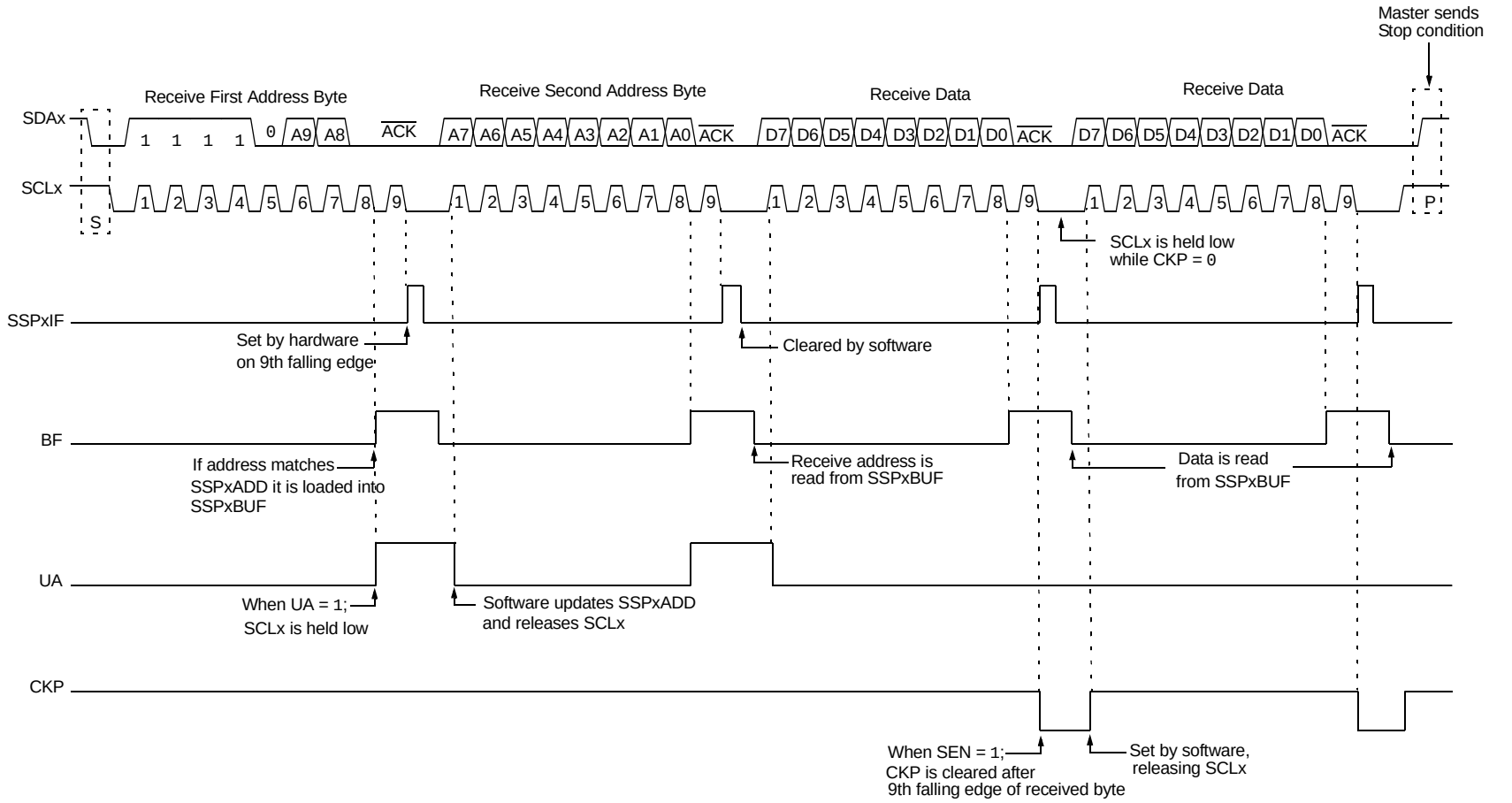


**FIGURE 15-20: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**



### 15.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN, of the SSPxCON2 register. When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSPx module then goes into Idle mode (Figure 15-30).

#### 15.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write does not occur).

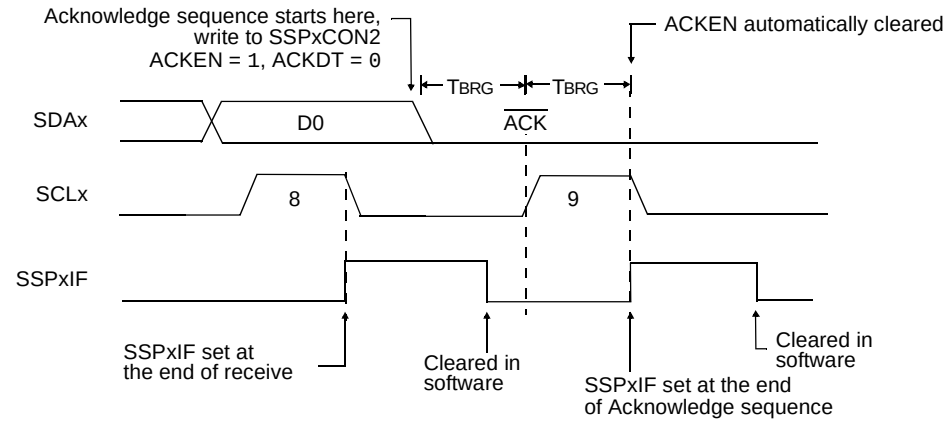
### 15.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN, of the SSPxCON2 register. At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 15-31).

#### 15.6.9.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

**FIGURE 15-30: ACKNOWLEDGE SEQUENCE WAVEFORM**




**Note:** TBRG = one Baud Rate Generator period.

### 16.1.2 EUSART ASYNCHRONOUS RECEIVER

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

#### 16.1.2.1 Enabling the Receiver

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

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

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

Setting the CREN bit of the RCSTAx register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTAx register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTAx register enables the EUSART. The RXx/DTx I/O pin must be configured as an input by setting the corresponding TRIS control bit. If the RXx/DTx pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

#### 16.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 16.1.2.5 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCxIF interrupt flag bit of the PIR1/PIR3 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREGx register.

| | |
|---|---|
| **Note:** | If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 16.1.2.6 "Receive Overrun Error"** for more information on overrun errors. |

#### 16.1.2.3 Receive Data Polarity

The polarity of the receive data can be controlled with the DTRXP bit of the BAUDCONx register. The default state of this bit is '0' which selects high true receive idle and data bits. Setting the DTRXP bit to '1' will invert the receive data resulting in low true idle and data bits. The DTRXP bit controls receive data polarity only in Asynchronous mode. In Synchronous mode the DTRXP bit has a different function.

**FIGURE 16-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**

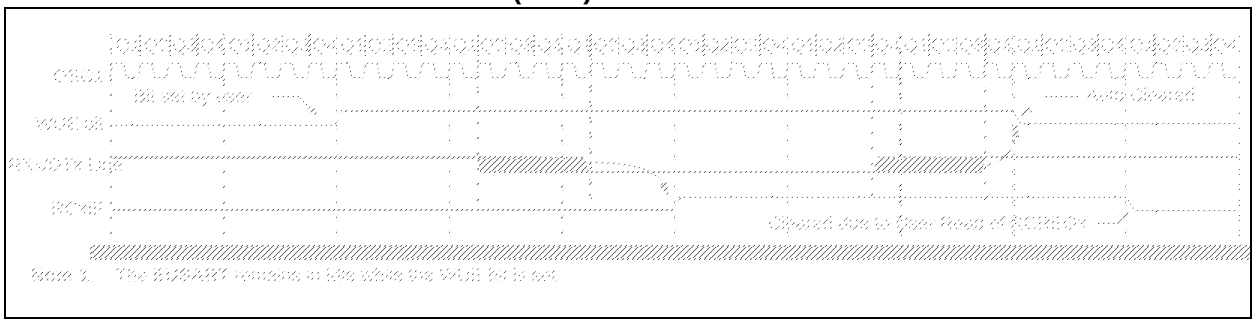
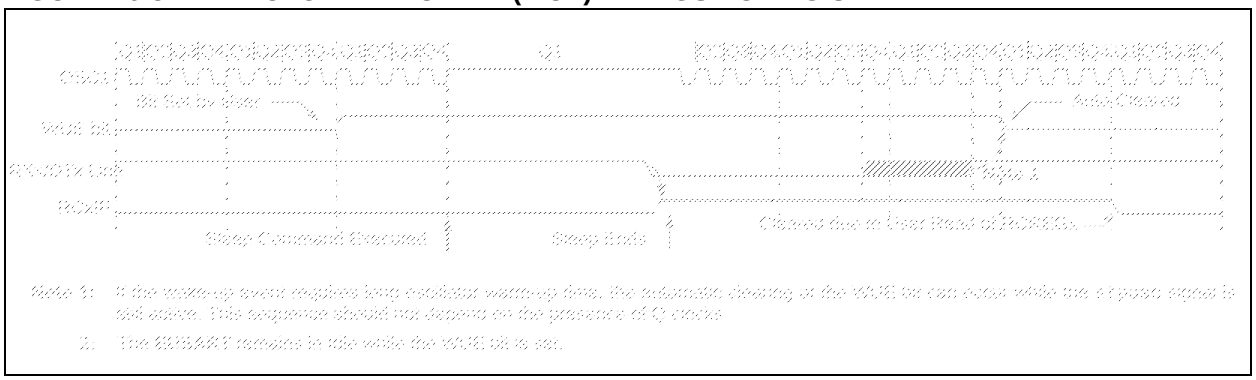


**FIGURE 16-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



## 17.3 Register Definitions: ADC Control

| **Note:** | Analog pin control is determined by the ANSELx registers (see Register 10-2) |
|---|---|

**REGISTER 17-1: ADCON0: A/D CONTROL REGISTER 0**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | CHS<4:0> | | | | | GO/$\overline{\text{DONE}}$ | ADON |
| 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 **Unimplemented:** Read as ‘0’

bit 6-2 **CHS<4:0>: Analog Channel Select bits**
00000 = AN0
00001 = AN1
00010 = AN2
00011 = AN3
00100 = AN4
00101 = AN5**(1)**
00110 = AN6**(1)**
00111 = AN7**(1)**
01000 = AN8
01001 = AN9
01010 = AN10
01011 = AN11
01100 = AN12
01101 = AN13
01110 = AN14
01111 = AN15
10000 = AN16
10001 = AN17
10010 = AN18
10011 = AN19
10100 = AN20**(1)**
10101 = AN21**(1)**
10110 = AN22**(1)**
10111 = AN23**(1)**
11000 = AN24**(1)**
11001 = AN25**(1)**
11010 = AN26**(1)**
11011 = AN27**(1)**
11100 = Reserved
11101 = CTMU
11110 = DAC
11111 = FVR BUF2 (1.024V/2.048V/2.096V Volt Fixed Voltage Reference)**(2)**

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

**Note 1:** Available on PIC18(L)F4XK22 devices only.
**2:** Allow greater than 15 μs acquisition time when measuring the Fixed Voltage Reference.

**EXAMPLE 19-2: CURRENT CALIBRATION ROUTINE**

```
#include "p18cxxx.h"

#define COUNT 500                           //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA
#define ADSCALE 1023                        //for unsigned conversion 10 sig bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0;                              //index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0;     //float values stored for calcs

//assume CTMU and A/D have been set up correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONHbits.CTMUEN = 1;                    //Enable the CTMU
CTMUCONLbits.EDG1STAT = 0;                  // Set Edge status bits to zero
CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;           //drain charge on the circuit
        DELAY;                              //wait 125us
        CTMUCONHbits.IDISSEN = 0;           //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;          //Begin charging the circuit
                                            //using CTMU current source
        DELAY;                              //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;          //Stop charging circuit

        PIR1bits.ADIF = 0;                  //make sure A/D Int not set
        ADCON0bits.GO=1;                    //and begin A/D conv.
        while(!PIR1bits.ADIF);              //Wait for A/D convert complete

        Vread = ADRES;                      //Get the value from the A/D
        PIR1bits.ADIF = 0;                  //Clear A/D Interrupt Flag
        VTot += Vread;                      //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);            //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                   //CTMUISrc is in 1/100ths of uA

}
```

**REGISTER 20-2: SRCON1: SR LATCH CONTROL REGISTER 1**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| SRSPE | SRSCKE | SRSC2E | SRSC1E | SRRPE | SRRCKE | SRRC2E | SRRC1E |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented | C = Clearable only bit |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **SRSPE:** SR Latch Peripheral Set Enable bit
1 = SRI pin status sets SR latch
0 = SRI pin status has no effect on SR latch

bit 6 **SRSCKE:** SR Latch Set Clock Enable bit
1 = Set input of SR latch is pulsed with DIVSRCLK
0 = Set input of SR latch is not pulsed with DIVSRCLK

bit 5 **SRSC2E:** SR Latch C2 Set Enable bit
1 = C2 Comparator output sets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 4 **SRSC1E:** SR Latch C1 Set Enable bit
1 = C1 Comparator output sets SR latch
0 = C1 Comparator output has no effect on SR latch

bit 3 **SRRPE:** SR Latch Peripheral Reset Enable bit
1 = SRI pin resets SR latch
0 = SRI pin has no effect on SR latch

bit 2 **SRRCKE:** SR Latch Reset Clock Enable bit
1 = Reset input of SR latch is pulsed with DIVSRCLK
0 = Reset input of SR latch is not pulsed with DIVSRCLK

bit 1 **SRRC2E:** SR Latch C2 Reset Enable bit
1 = C2 Comparator output resets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 0 **SRRC1E:** SR Latch C1 Reset Enable bit
1 = C1 Comparator output resets SR latch
0 = C1 Comparator output has no effect on SR latch

**TABLE 20-2: REGISTERS ASSOCIATED WITH THE SR LATCH**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---|---|---|---|---|---|---|---|---|---|
| SRCON0 | SRLEN | SRCLK<2:0> | | | SRQEN | SRNQEN | SRPS | SRPR | 329 |
| SRCON1 | SRSPE | SRSCKE | SRSC2E | SRSC1E | SRRPE | SRRCKE | SRRC2E | SRRC1E | 330 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 151 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 151 |
| WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 | 152 |

**Legend:** Shaded bits are not used with this module.

## 23.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

The PIC18(L)F2X/4XK22 devices have a High/Low-Voltage Detect module (HLVD). This is a programmable circuit that sets both a device voltage trip point and the direction of change from that point. If the device experiences an excursion past the trip point in that direction, an interrupt flag is set. If the interrupt is enabled, the program execution branches to the interrupt vector address and the software responds to the interrupt.

The High/Low-Voltage Detect Control register (Register 23-1) completely controls the operation of the HLVD module. This allows the circuitry to be “turned off” by the user under software control, which minimizes the current consumption for the device.

The module’s block diagram is shown in Figure 23-1.

### 23.1 Register - HLVD Control

**REGISTER 23-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER**

| R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| VDIRMAG | BGVST | IRVST | HLVDEN | HLVDL<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 | x = Bit is unknown |

bit 7 **VDIRMAG:** Voltage Direction Magnitude Select bit
1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Reference Voltages Stable Status Flag bit
1 = Internal band gap voltage references are stable
0 = Internal band gap voltage reference is not stable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the voltage detect logic will generate the interrupt flag at the specified voltage range
0 = Indicates that the voltage detect logic will not generate the interrupt flag at the specified voltage range and the HLVD interrupt should not be enabled

bit 4 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
1 = HLVD enabled
0 = HLVD disabled

bit 3-0 **HLVDL<3:0>:** Voltage Detection Level bits(1)
1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Maximum setting
.
.
.
0000 = Minimum setting

**Note 1:** See Table 27-5 for specifications.

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

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

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

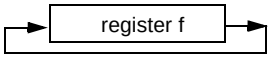
Operation: (f<n>) → dest<n – 1>,
(f<0>) → dest<7>

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 25.2.3 "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 25.2.3 "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

### 25.2.5 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/4XK22 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 28-30: PIC18LF2X/4XK22 TYPICAL IDD: RC_RUN HF-INTOSC with PLL**

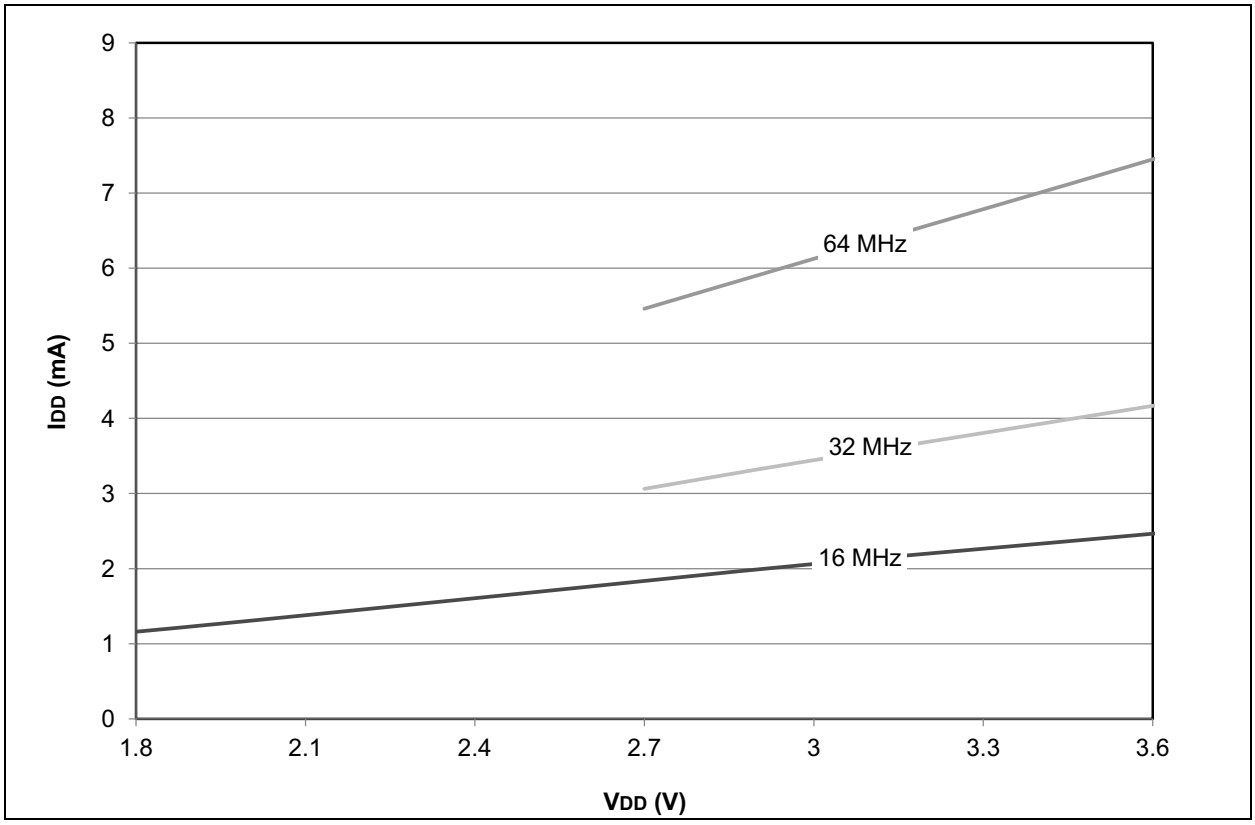



**FIGURE 28-31: PIC18LF2X/4XK22 MAXIMUM IDD: RC_RUN HF-INTOSC with PLL**

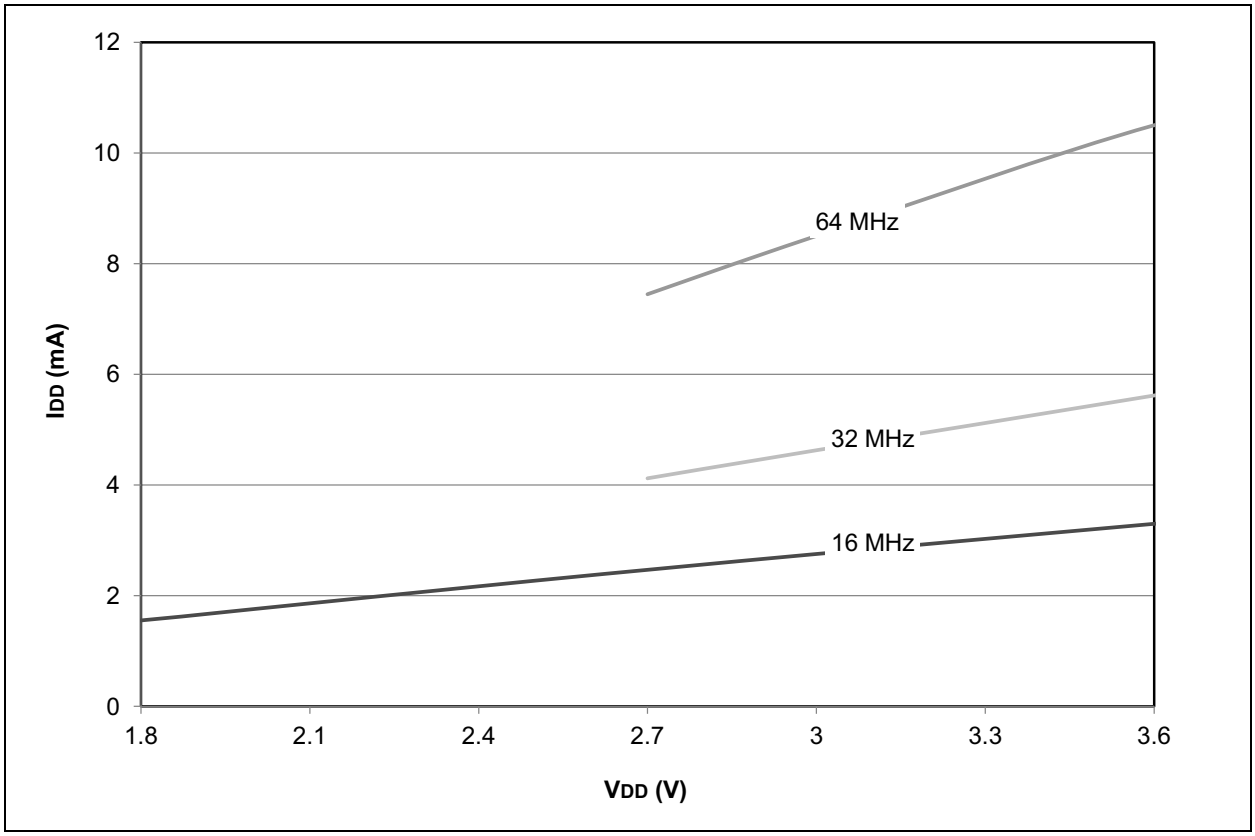


**FIGURE 28-34: PIC18LF2X/4XK22 TYPICAL IDD: RC_IDLE LF-INTOSC 31 kHz**

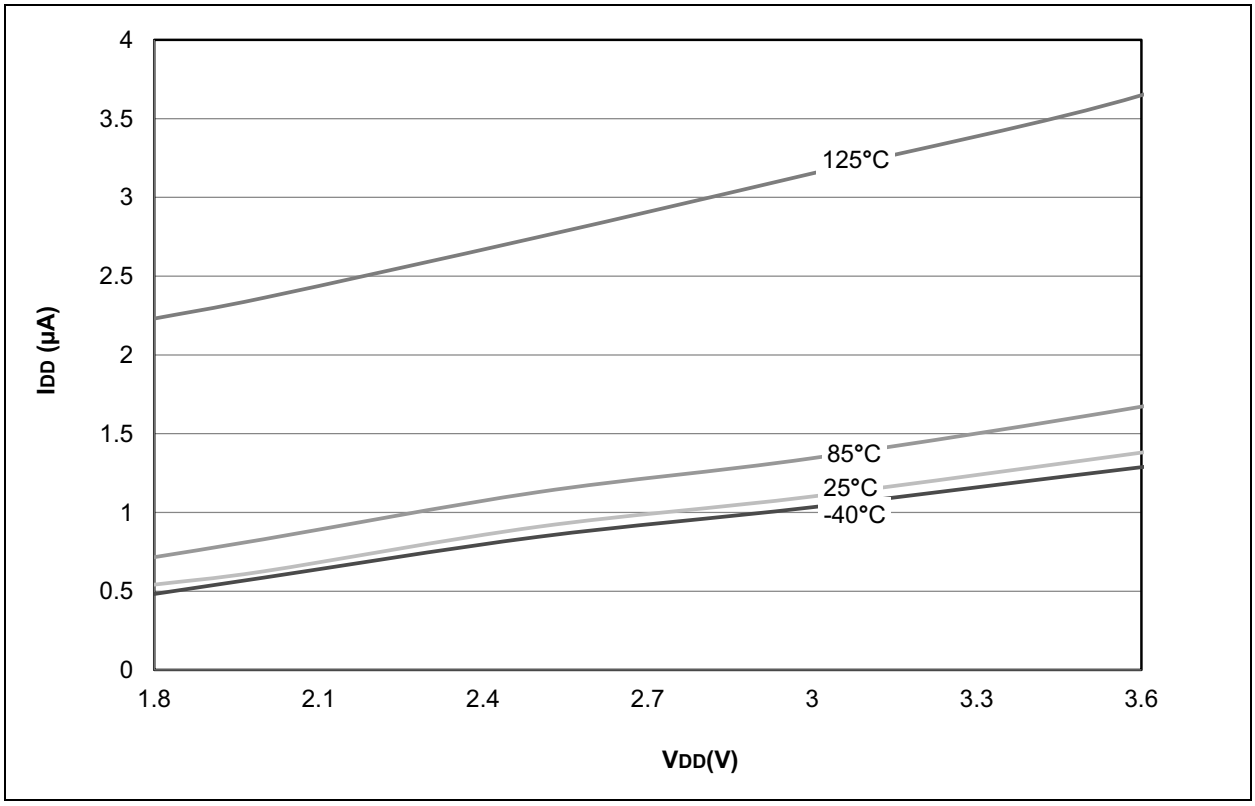



**FIGURE 28-35: PIC18LF2X/4XK22 MAXIMUM IDD: RC_IDLE LF-INTOSC 31 kHz**

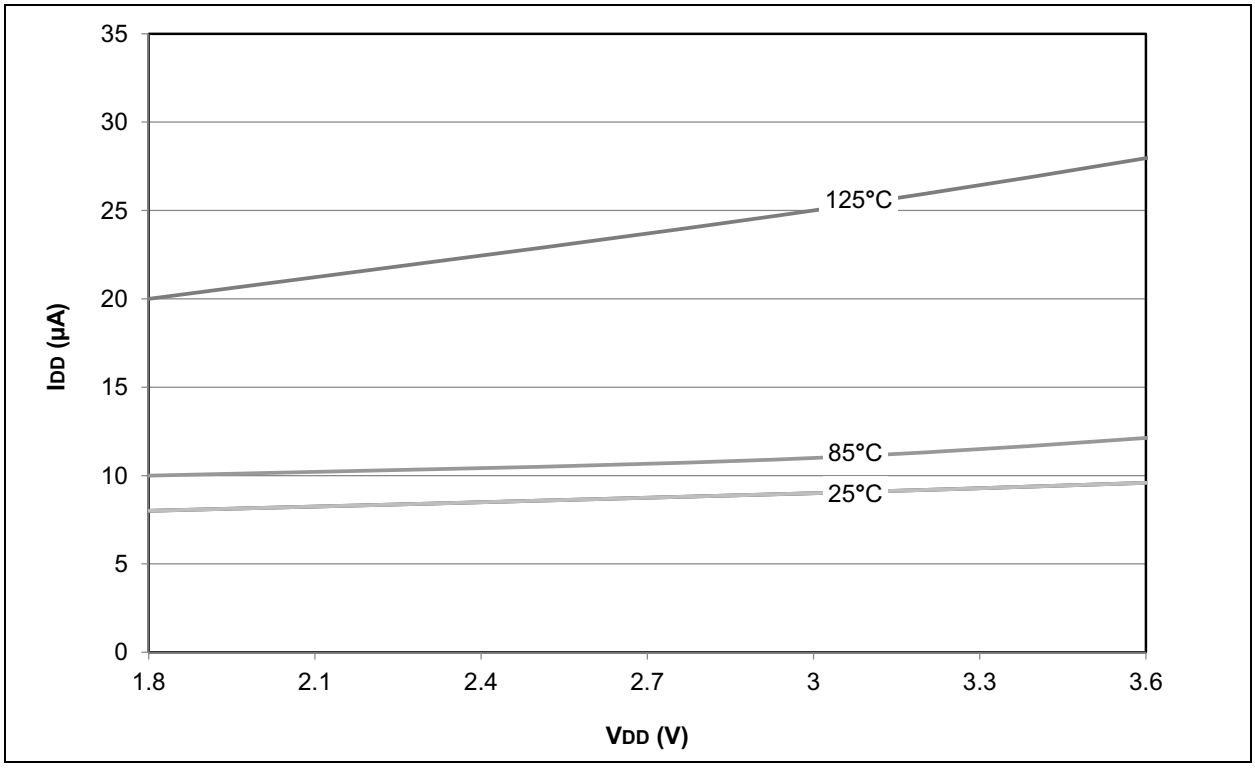


**FIGURE 28-82: PIC18(L)F2X/4XK22 TTL BUFFER INPUT HIGH VOLTAGE**

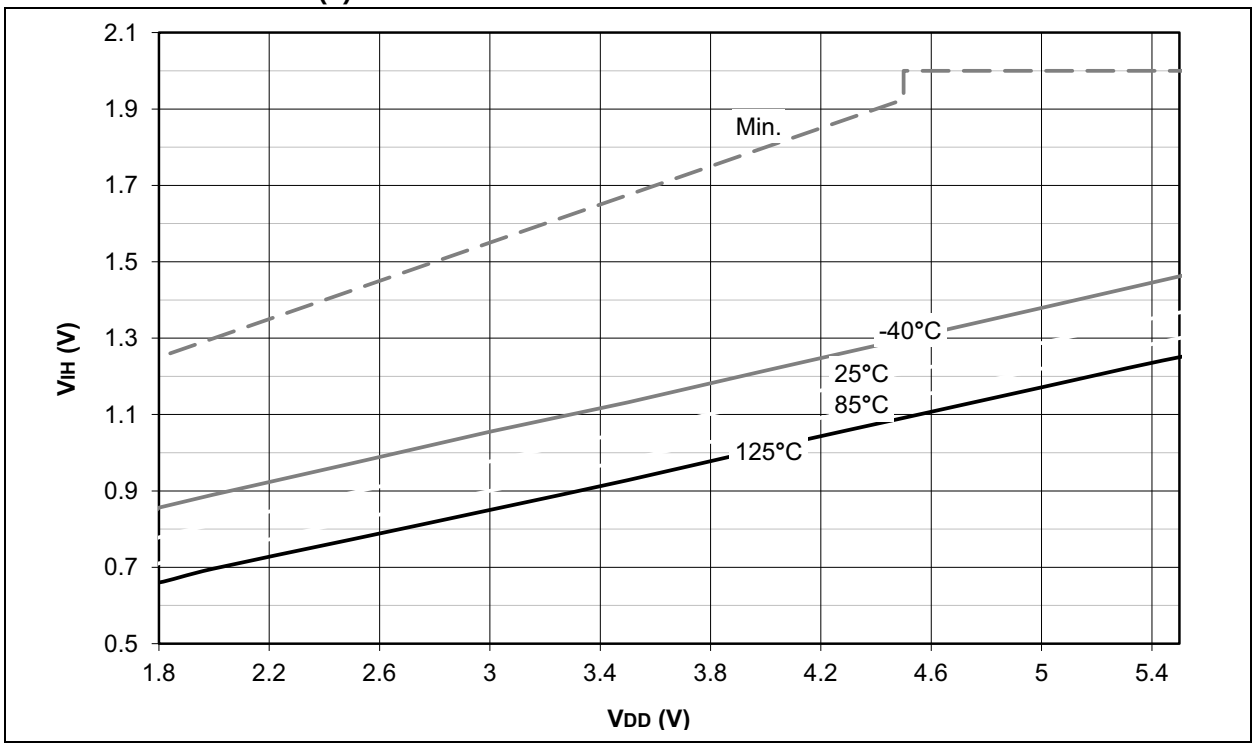



**FIGURE 28-83: PIC18(L)F2X/4XK22 SCHMITT TRIGGER BUFFER INPUT HIGH VOLTAGE**

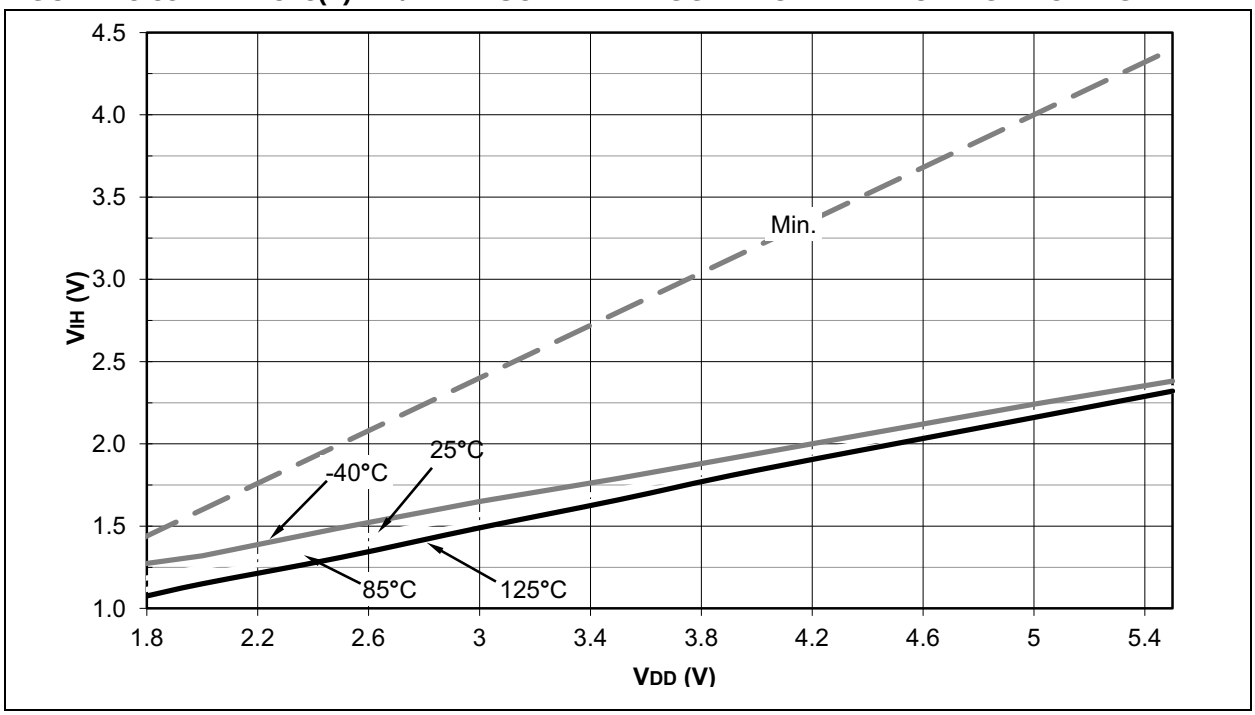


**FIGURE 28-89: PIC18LF2X/4XK22 COMPARATOR OFFSET VOLTAGE, NORMAL-POWER MODE; VDD=1.8V**