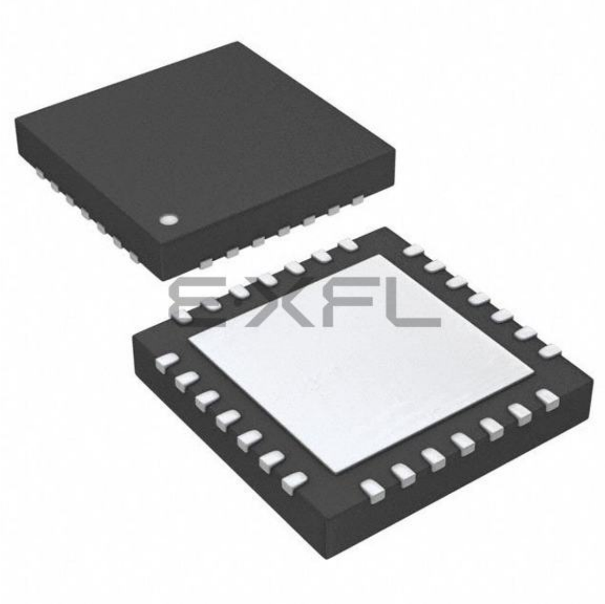
Microchip Technology - PIC16F722A-E/ML Datasheet



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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x8b
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/pic16f722a-e-ml

PIC16(L)F72X Family Types

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	High-Endurance Flash Memory (bytes)	I/O's(2)	8-bit ADC (ch)	CapSense (ch)	Timers (8/16-bit)	AUSART	SSP (I^2C/SPI)	CCP	Debug(1)	XLP
PIC16(L)F707	(1)	8192	363	0	36	14	32	4/2	1	1	2	I	Y
PIC16(L)F720	(2)	2048	128	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F721	(2)	4096	256	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F722	(4)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F722A	(3)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723	(4)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723A	(3)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F724	(4)	4096	192	0	36	14	16	2/1	1	1	2	I	Y
PIC16(L)F726	(4)	8192	368	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F727	(4)	8192	368	0	36	14	16	2/1	1	1	2	I	Y

Note 1: I - Debugging, Integrated on Chip; H - Debugging, Requires Debug Header.

2: One pin is input-only.

Data Sheet Index: (Unshaded devices are described in this document.)

1: DS41418 PIC16(L)F707 Data Sheet, 40/44-Pin Flash, 8-bit Microcontrollers

2: DS41430 PIC16(L)F720/721 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers

3: DS41417 PIC16(L)F722A/723A Data Sheet, 28-Pin Flash, 8-bit Microcontrollers

4: DS41341 PIC16(L)F72X Data Sheet, 28/40/44-Pin Flash, 8-bit Microcontrollers

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F722A/723A has a 13-bit program counter capable of addressing a 2K x 14 program memory space for the PIC16(L)F722A (0000h-07FFh) and a 4K x 14 program memory space for the PIC16(L)F723A (0000h-0FFFh). Accessing a location above the memory boundaries for the PIC16(L)F722A will cause a wrap-around within the first 2K x 14 program memory space. Accessing a location above the memory boundaries for the PIC16(L)F723A will cause a wrap-around within the first 4K x 14 program memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F722A

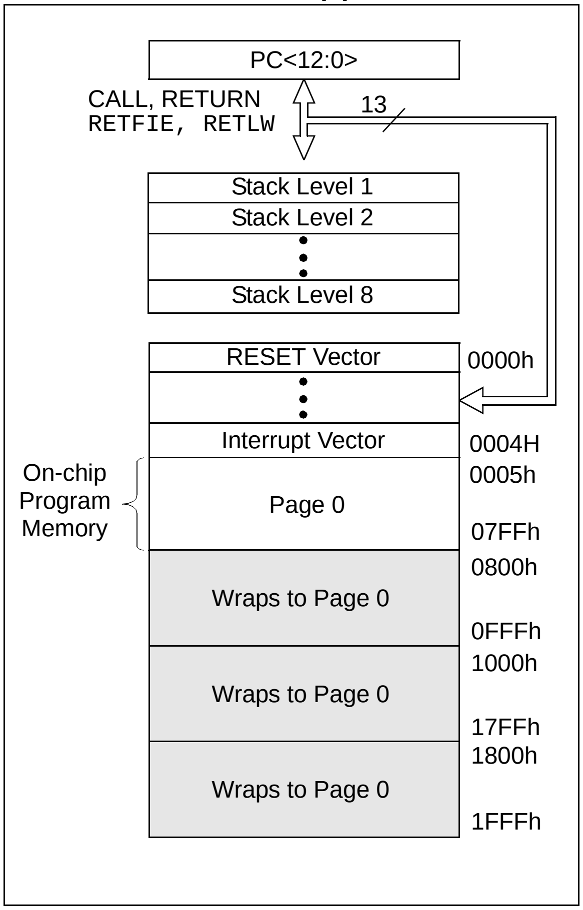
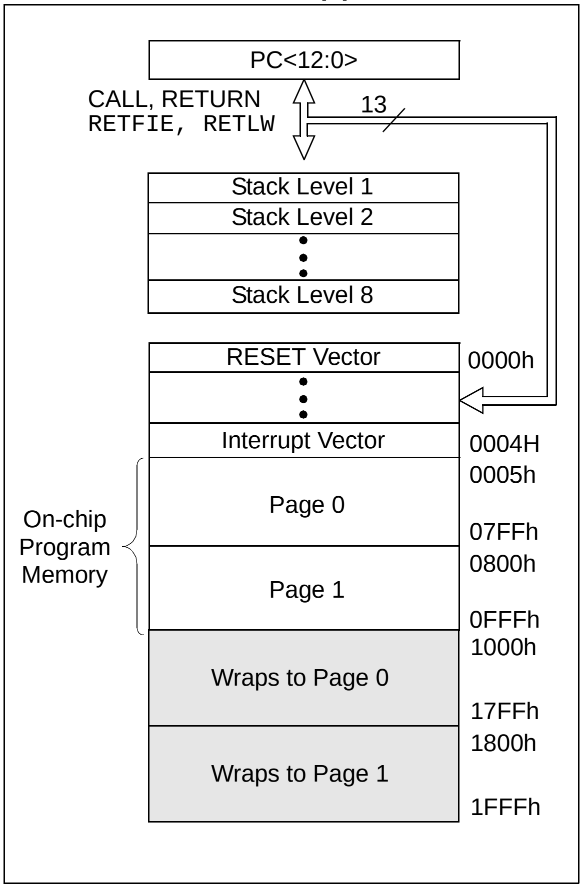


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F723A



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPRs) and the Special Function Registers (SFRs). Bits RP0 and RP1 are bank select bits.

RP1	RP0		
0	0	→	Bank 0 is selected
0	1	→	Bank 1 is selected
1	0	→	Bank 2 is selected
1	1	→	Bank 3 is selected

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank are mirrored in another bank for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 bits in the PIC16(L)F722A and 192 x 8 bits in the PIC16(L)F723A. Each register is accessed either directly or indirectly through the File Select Register (FSR), (Refer to **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (refer to Table 2-1). These registers are static RAM.

The Special Function Registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

7.6 External Clock Modes

7.6.1 OSCILLATOR START-UP TIMER (OST)

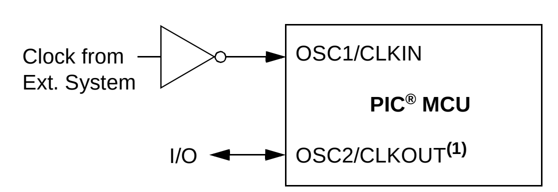
If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations on the OSC1 pin before the device is released from Reset. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

7.6.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O. Figure 7-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 7-2: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Alternate pin functions are described in **Section 6.1 "Alternate Pin Function"**.

7.6.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 7-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

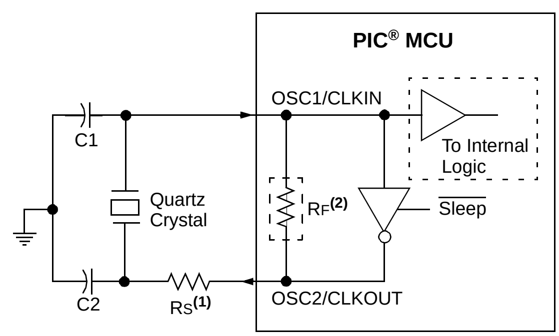
LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 7-3 and Figure 7-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

FIGURE 7-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

2: The value of RF varies with the Oscillator mode selected.

> **Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.
>
> **2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.
>
> **3:** For oscillator design assistance, reference the following Microchip Applications Notes:
>
> - *AN826, Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices* (DS00826)
> - *AN849, Basic PIC® Oscillator Design* (DS00849)
> - *AN943, Practical PIC® Oscillator Analysis and Design* (DS00943)
> - *AN949, Making Your Oscillator Work* (DS00949)

8.2 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out using ICSP™ for verification purposes.

Note:	The entire Flash program memory will be erased when the code protection is turned off. See the "*PIC16(L)F72X Memory Programming Specification"* (DS41332) for more information.

8.3 User ID

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are reported when using MPLAB IDE. See the "*PIC16(L)F72X Memory Programming Specification"* (DS41332) for more information.

13.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 13-1 for a block diagram of Timer2.

13.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented.

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

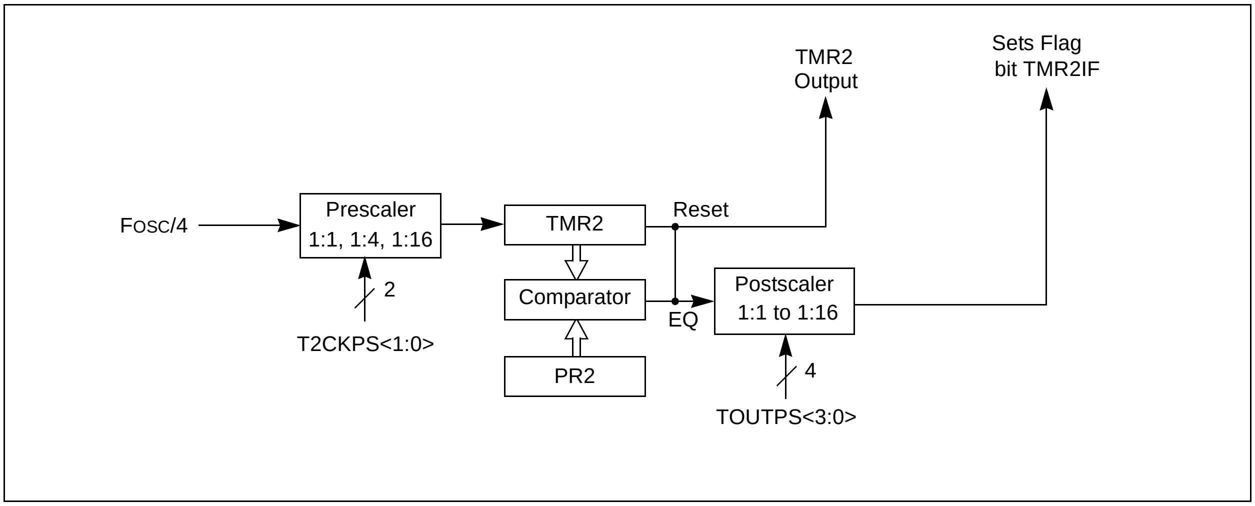
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

> **Note:** TMR2 is not cleared when T2CON is written.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



14.1 Analog MUX

The capacitive sensing module can monitor up to 8 inputs. The capacitive sensing inputs are defined as CPS<7:0>. To determine if a frequency change has occurred the user must:

- Select the appropriate CPS pin by setting the CPSCH<2:0> bits of the CPSCON1 register
- Set the corresponding ANSEL bit
- Set the corresponding TRIS bit
- Run the software algorithm

Selection of the CPSx pin while the module is enabled will cause the capacitive sensing oscillator to be on the CPSx pin. Failure to set the corresponding ANSEL and TRIS bits can cause the capacitive sensing oscillator to stop, leading to false frequency readings.

14.2 Capacitive Sensing Oscillator

The capacitive sensing oscillator consists of a constant current source and a constant current sink, to produce a triangle waveform. The CPSOUT bit of the CPSCON0 register shows the status of the capacitive sensing oscillator, whether it is a sinking or sourcing current. The oscillator is designed to drive a capacitive load (single PCB pad) and at the same time, be a clock source to either Timer0 or Timer1. The oscillator has three different current settings as defined by CPSRNG<1:0> of the CPSCON0 register. The different current settings for the oscillator serve two purposes:

- Maximize the number of counts in a timer for a fixed-time base
- Maximize the count differential in the timer during a change in frequency

14.3 Timer Resources

To measure the change in frequency of the capacitive sensing oscillator, a fixed-time base is required. For the period of the fixed-time base, the capacitive sensing oscillator is used to clock either Timer0 or Timer1. The frequency of the capacitive sensing oscillator is equal to the number of counts in the timer divided by the period of the fixed-time base.

14.4 Fixed-Time Base

To measure the frequency of the capacitive sensing oscillator, a fixed-time base is required. Any timer resource or software loop can be used to establish the fixed-time base. It is up to the end user to determine the method in which the fixed-time base is generated.

> **Note:** The fixed-time base can not be generated by the timer resource the capacitive sensing oscillator is clocking.

14.4.1 TIMER0

To select Timer0 as the timer resource for the capacitive sensing module:

- Set the T0XCS bit of the CPSCON0 register
- Clear the T0CS bit of the OPTION register

When Timer0 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer0. Refer to **Section 11.0 “Timer0 Module”** for additional information.

14.4.2 TIMER1

To select Timer1 as the timer resource for the capacitive sensing module, set the TMR1CS<1:0> of the T1CON register to ‘11’. When Timer1 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer1. Because the Timer1 module has a gate control, developing a time base for the frequency measurement can be simplified using either:

- The Timer0 overflow flag
- The Timer2 overflow flag
- The WDT overflow flag

It is recommended that one of these flags, in conjunction with the toggle mode of the Timer1 gate, is used to develop the fixed-time base required by the software portion of the capacitive sensing module. Refer to **Section 12.0 “Timer1 Module with Gate Control”** for additional information.

TABLE 14-1: TIMER1 ENABLE FUNCTION

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	On
1	1	Count Enabled by input

16.1 AUSART Asynchronous Mode

The AUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH Mark state which represents a '1' data bit, and a VOL Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. Refer to Table 16-5 for examples of baud rate configurations.

The AUSART transmits and receives the LSb first. The AUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

16.1.1 AUSART ASYNCHRONOUS TRANSMITTER

The AUSART transmitter block diagram is shown in Figure 16-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

16.1.1.1 Enabling the Transmitter

The AUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

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

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the AUSART. Clearing the SYNC bit of the TXSTA register configures the AUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the AUSART and automatically configures the TX/CK I/O pin as an output.

Note 1:	When the SPEN bit is set the RX/DT I/O pin is automatically configured as an input, regardless of the state of the corresponding TRIS bit and whether or not the AUSART receiver is enabled. The RX/DT pin data can be read via a normal PORT read but PORT latch data output is precluded.
2:	The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

16.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

16.1.1.3 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the AUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

16.1.2.8 Asynchronous Reception Setup:

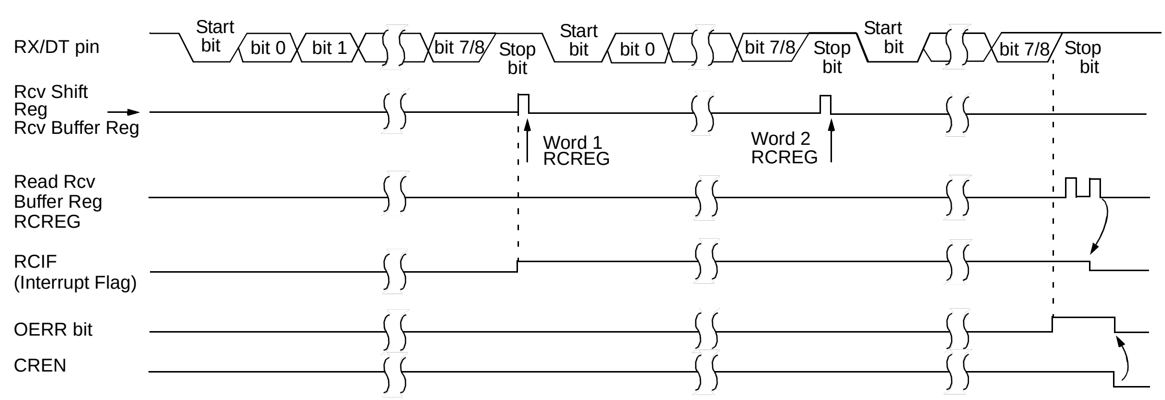
1. Initialize the SPBRG register and the BRGH bit to achieve the desired baud rate (refer to **Section 16.2 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Enable reception by setting the CREN bit.
6. The RCIF interrupt flag bit of the PIR1 register will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE bit of the PIE1 register was also set.
7. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
8. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

16.1.2.9 9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRG register and the BRGH bit to achieve the desired baud rate (refer to **Section 16.2 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. Enable 9-bit reception by setting the RX9 bit.
5. Enable address detection by setting the ADDEN bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit of the PIR1 register will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit of the PIE1 register was also set.
8. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device’s address.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 16-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

REGISTER 16-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	—	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don’t care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** AUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as ‘0’

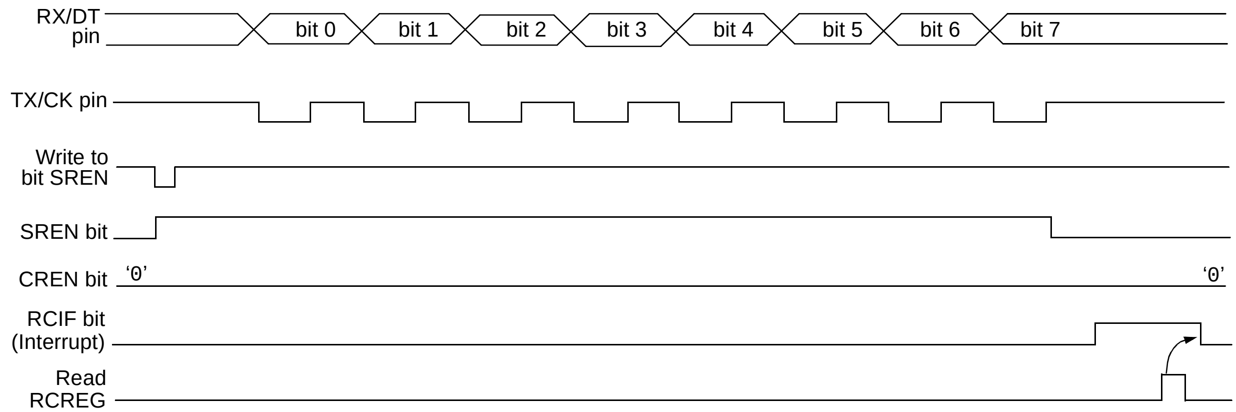
bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Synchronous mode.

FIGURE 16-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Synchronous Master mode with bit SREN = 1 and bit BRGH = 0.

TABLE 16-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
RCREG	AUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for synchronous master reception.

REGISTER 18-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	PMA12	PMA11	PMA10	PMA9	PMA8
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **PMA<12:8>:** Program Memory Read Address bits

REGISTER 18-5: PMADRL: PROGRAM MEMORY ADDRESS LOW REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0
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 7-0 **PMA<7:0>:** Program Memory Read Address bits

TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH PROGRAM MEMORY READ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PMCON1	Reserved	—	—	—	—	—	—	RD	168
PMADRH	—	—	—	Program Memory Read Address Register High Byte					169
PMADRL	Program Memory Read Address Register Low Byte								169
PMDATH	—	—	Program Memory Read Data Register High Byte						168
PMDATL	Program Memory Read Data Register Low Byte								168

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Program Memory Read.

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW	Return with literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the 8-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.
Words:	1
Cycles:	2

<u>Example:</u>

```
      CALL TABLE;W contains
      table
          ;offset value
TABLE •   ;W now has table value
      •
      •
      ADDWF PC  ;W = offset
      RETLW k1  ;Begin table
      RETLW k2  ;
      •
      •
      •
      RETLW kn  ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

TABLE 23-13: I²C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
SP100*	THIGH	Clock high time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1CB	250	ns	CB is specified to be from 10-400 pF
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
SP111	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 24-20: PIC16LF722A/723A MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD}, INTOSC MODE

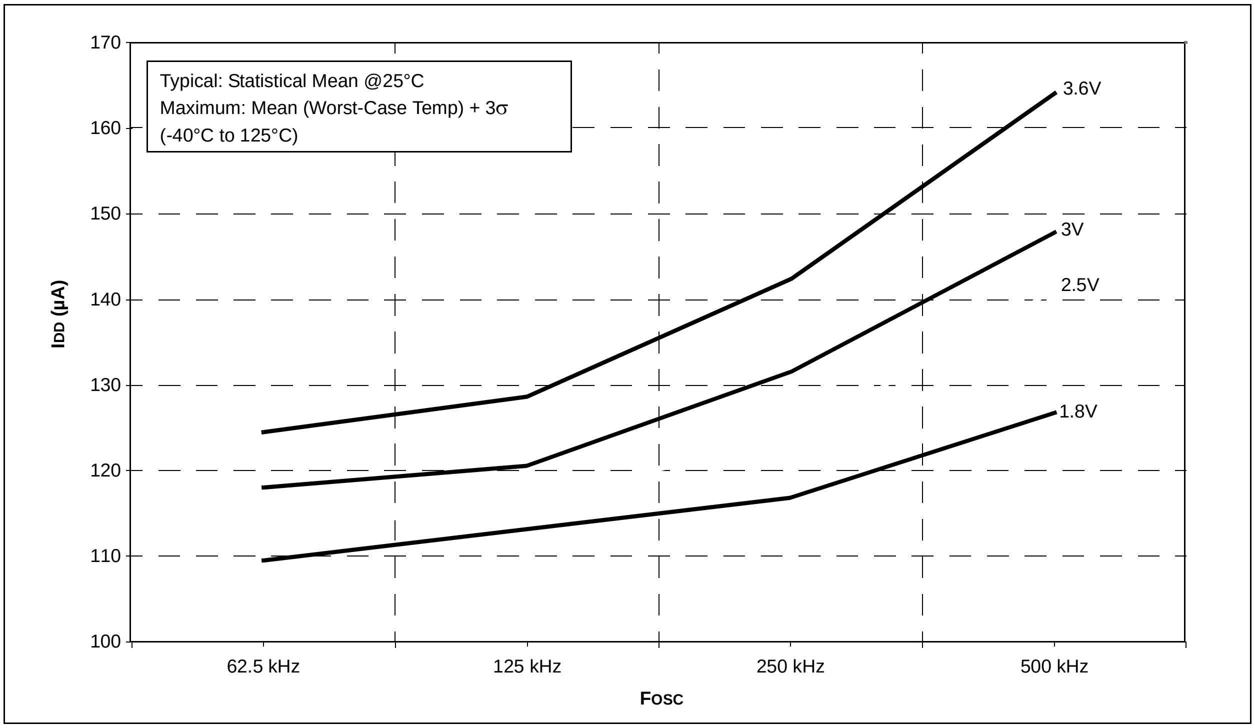


FIGURE 24-21: PIC16F722A/723A MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD}, INTOSC MODE, V_{CAP} = 0.1µF

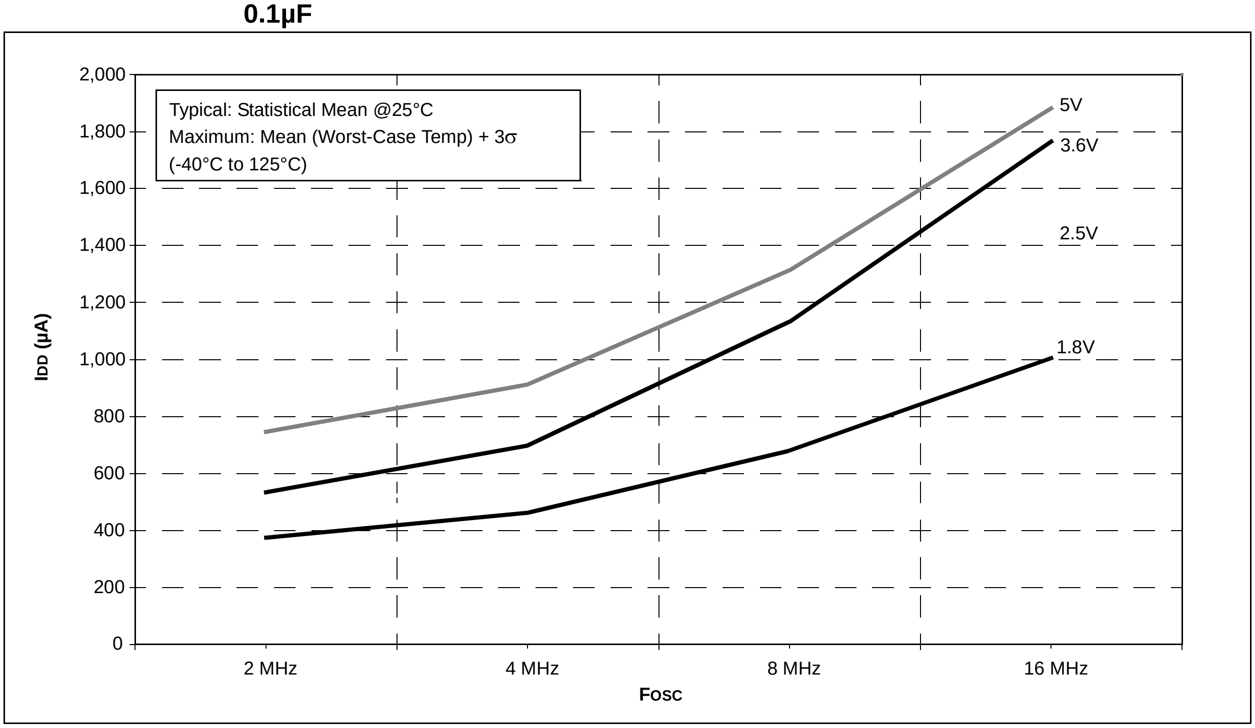


FIGURE 24-30: PIC16LF722A/723A TYPICAL BASE IPD vs. VDD

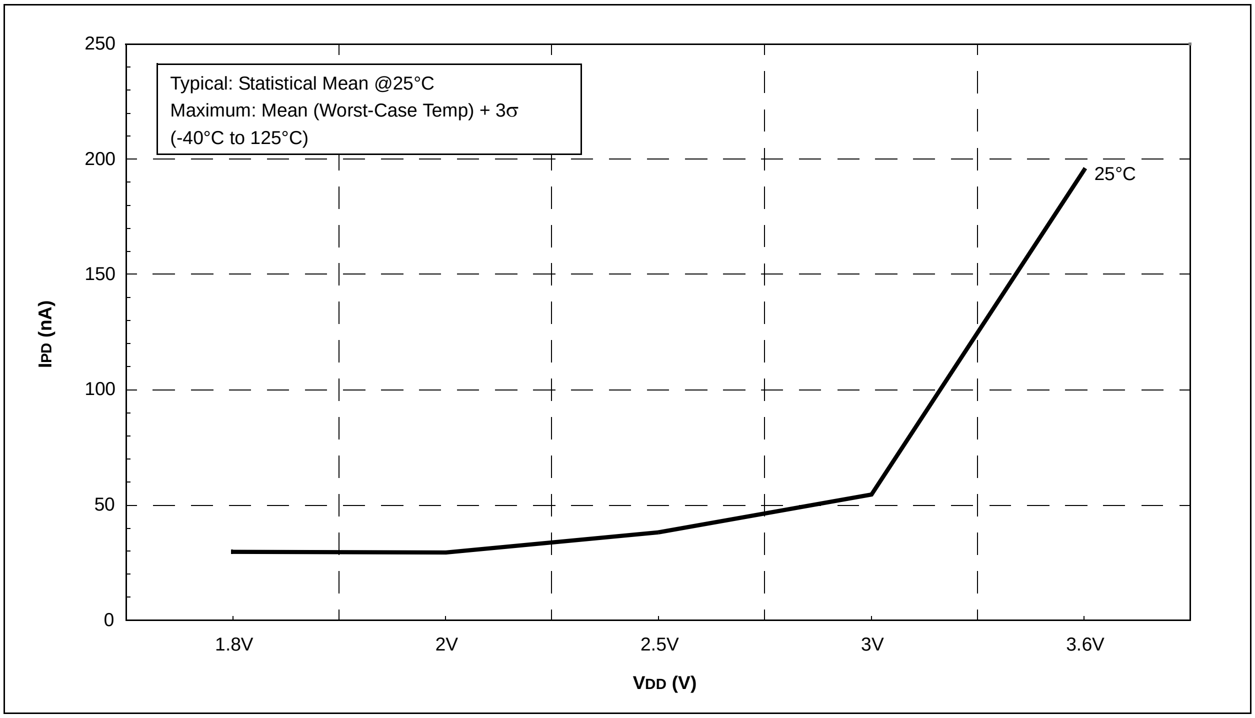


FIGURE 24-31: PIC16F722A/723A FIXED VOLTAGE REFERENCE IPD vs. VDD, VCAP = 0.1µF

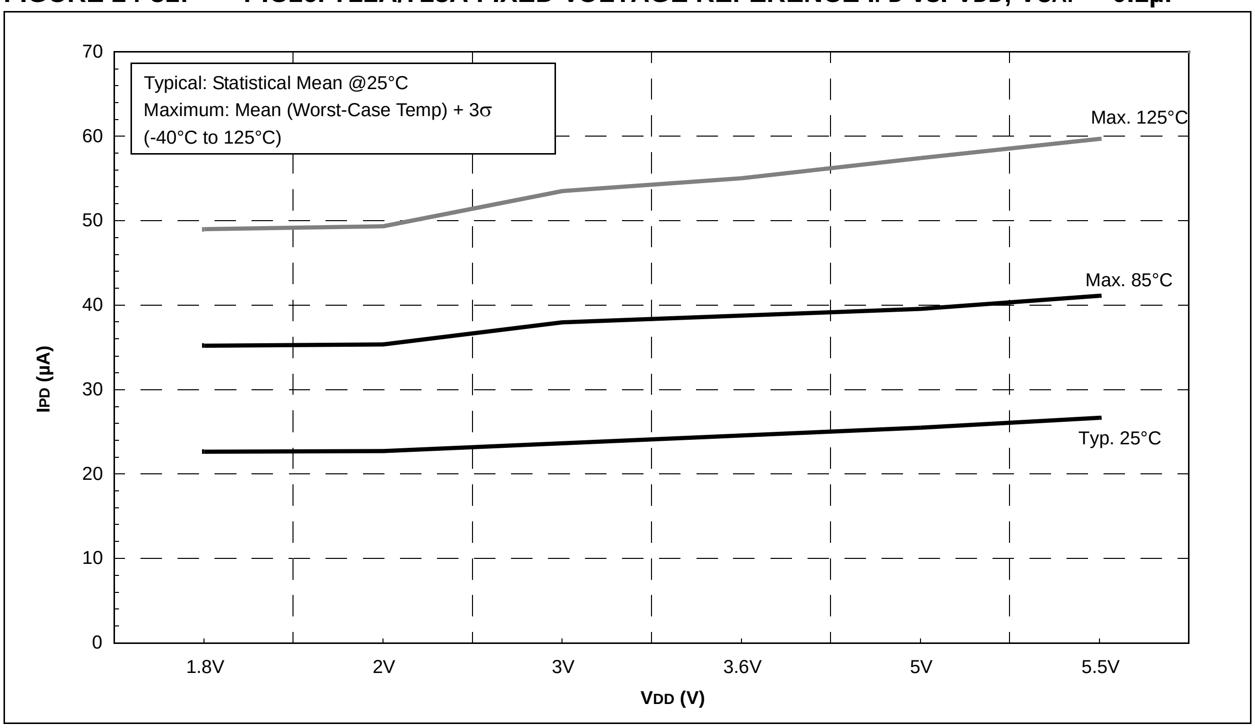


FIGURE 24-40: PIC16LF722A/723A CAP SENSE LOW POWER IPD vs. VDD

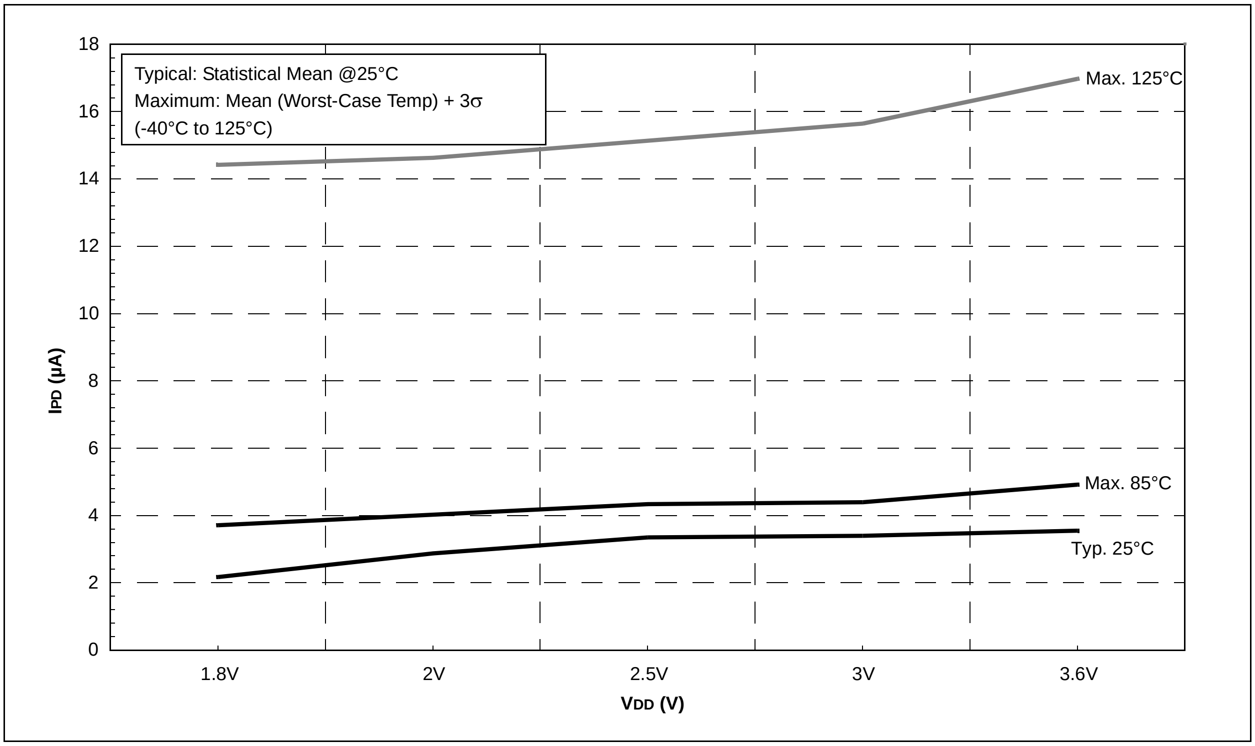
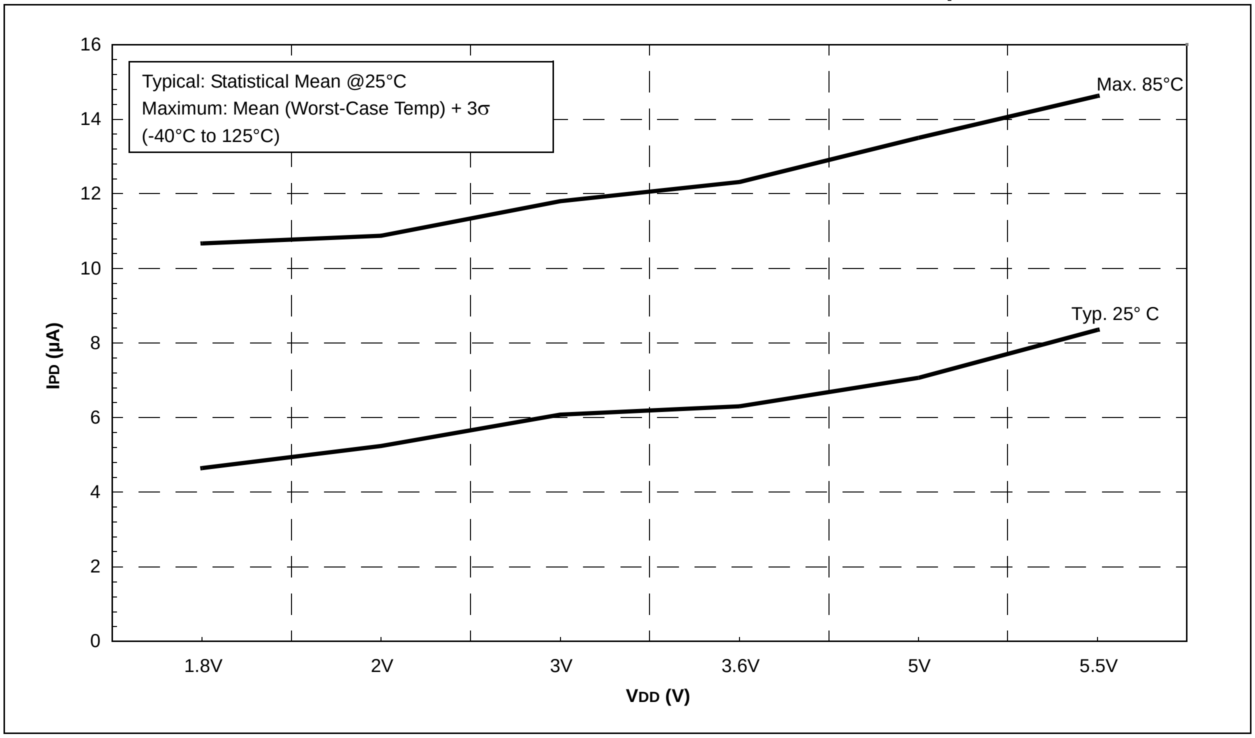
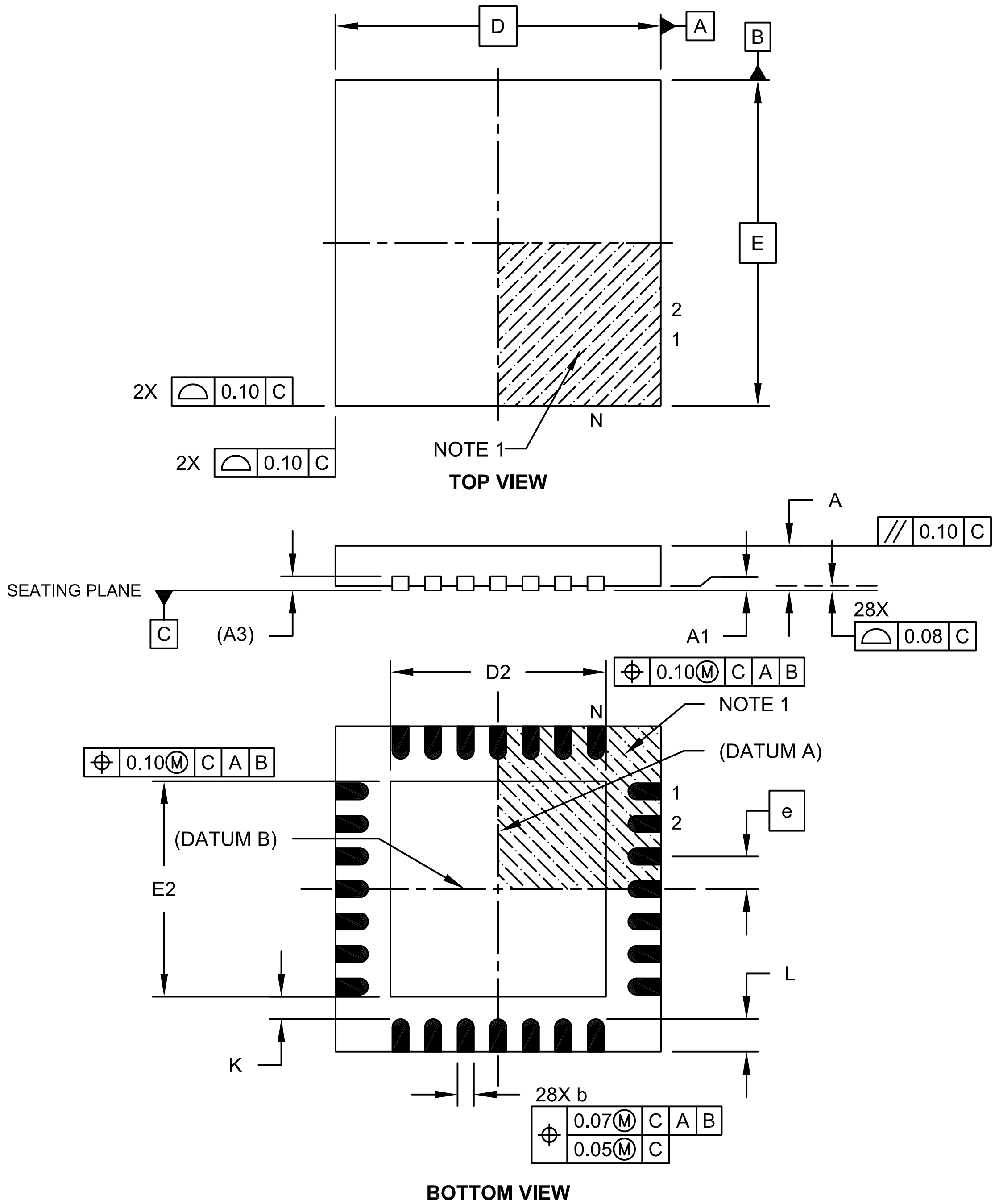


FIGURE 24-41: PIC16F722A/723A T1OSC 32 kHz IPD vs. VDD, VCAP = 0.1µF



28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-152A Sheet 1 of 2

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (April 2010)

Original release of this data sheet.

Revision B (January 2012)

Updated the data sheet to new format; Updated Figure 9-1 and Register 9-1; Updated the Packaging Information section; Updated the Product Identification System section; Other minor corrections.

Revision C (03/2016)

Updated Table 2-1, Table 6-1 and Table 6-3; Updated Register 14-2; Other minor corrections.

APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC® devices to the PIC16F722A/723A family of devices.

Note:	This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

Note:	The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the oscillator mode may be required.

B.1 PIC16F77 to PIC16F722A/723A

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F77	PIC16F722A/723A
Max. Operating Speed	20 MHz	20 MHz
Max. Program Memory (Words)	8K	4K
Max. SRAM (Bytes)	368	192
A/D Resolution	8-bit	8-bit
Timers (8/16-bit)	2/1	2/1
Oscillator Modes	4	8
Brown-out Reset	Y	Y
Internal Pull ups	RB<7:0>	RB<7:0>
Interrupt-on-change	RB<7:4>	RB<7:0>
Comparator	0	0
USART	Y	Y
Extended WDT	N	N
Software Control Option of WDT/BOR	N	N
INTOSC Frequencies	None	500 kHz - 16 MHz
Clock Switching	N	N

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://www.microchip.com/support