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	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf721-i-so

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0											
00h(2)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 module Register								xxxx xxxx	uuuu uuuu
02h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
03h(2)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h(2)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
06h	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah(1),(2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	—	$\overline{T1SYNC}$	—	TMR1ON	0000 -0-0	uuuu -u-u
11h	TMR2	Timer2 module Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register Low Byte								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register High Byte								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	DC1	B1	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRES	ADC Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	--00 0000

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

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.
2: These registers can be addressed from any bank.
3: Accessible only when SSPM<3:0> = 1001.
4: This bit is unimplemented and reads as '1'.
5: See Register 6-2.

TABLE 3-6: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset/ Brown-out Reset(1)	$\overline{\text{MCLR}}$ Reset/ WDT Reset	Wake-up from Sleep through Interrupt/Time out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/ 104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	--xx xxxx	--xx xxxx	--uu uuuu
PORTB	06h	xxxx ----	xxxx ----	uuuu ----
PORTC	07h	xxxx xxxx	xxxx xxxx	uuuu uuuu
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 000x	0000 000x	uuuu uuuu(2)
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu(2)
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 -0-0	0000 -0-0	uuuu -u-u
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
SSPBUF	13h	xxxx xxxx	xxxx xxxx	uuuu uuuu
SSPCON	14h	0000 0000	0000 0000	uuuu uuuu
CCPR1L	15h	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR1H	16h	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP1CON	17h	--00 0000	--00 0000	--uu uuuu
RCSTA	18h	0000 000x	0000 000x	uuuu uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
ADRES	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	--00 0000	--00 0000	--uu uuuu
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	--11 -111	--11 -111	--uu -uuu
TRISB	86h	1111 ----	1111 ----	uuuu ----
TRISC	87h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PCON	8Eh	---- --qq	---- --uu(1,5)	---- --uu
T1GCON	8Fh	0000 0x00	uuuu uxuu	uuuu uxuu
OSCCON	90h	--10 qq--	--10 qq--	--uu qq--
OSCTUNE	91h	--00 0000	--uu uuuu	--uu uuuu
PR2	92h	1111 1111	1111 1111	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 3-8 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 3-6: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

Register	Address	Power-on Reset/ Brown-out Reset(1)	$\overline{\text{MCLR}}$ Reset/ WDT Reset	Wake-up from Sleep through Interrupt/Time out
SSPADD	93h	0000 0000	0000 0000	uuuu uuuu
SSPMSK	93h	1111 1111	1111 1111	uuuu uuuu
SSPSTAT	94h	0000 0000	0000 0000	uuuu uuuu
WPUB	115h	1111 ----	1111 ----	uuuu ----
WPUA	95h	--11 1111	--11 1111	--uu uuuu
IOCB	116h	0000 ----	0000 ----	uuuu ----
IOCA	96h	--00 0000	--00 0000	--uu uuuu
TXSTA	98h	0000 -010	0000 -010	uuuu -uuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
FVRCON	9Dh	q000 --00	q000 --00	uuuu --uu
ADCON1	9Fh	-000 ----	-000 ----	-uuu ----
PMDATL	10Ch	xxxx xxxx	xxxx xxxx	uuuu uuuu
PMADRL	10Dh	0000 0000	0000 0000	uuuu uuuu
PMDATH	10Eh	--xx xxxx	--xx xxxx	--uu uuuu
PMADRH	10Fh	---0 0000	---0 0000	---u uuuu
ANSELA	185h	---1 -111	---1 -111	---u -uuu
ANSELB	186h	--11 ----	--11 ----	--uu ----
ANSELC	187h	11-- 1111	11-- 1111	uu-- uuuu
PMCON1	18Ch	1000 -000	1000 -000	1000 -000

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If V_{DD} goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 3-8 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

4.3 Interrupts During Sleep

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

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to the **Section 19.0 "Power-Down Mode (Sleep)"** for more details.

4.4 INT Pin

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

4.5 Context Saving

When an interrupt occurs, only the return PC value is saved to the stack. If the ISR modifies or uses an instruction that modifies key registers, their values must be saved at the beginning of the ISR and restored when the ISR completes. This prevents instructions following the ISR from using invalid data. Examples of key registers include the W, STATUS, FSR and PCLATH registers.

Note:	The microcontroller does not normally require saving the PCLATH register. However, if computed GOTOs are used, the PCLATH register must be saved at the beginning of the ISR and restored when the ISR is complete to ensure correct program flow.

The code shown in Example 4-1 can be used to do the following.

- Save the W register
- Save the STATUS register
- Save the PCLATH register
- Execute the ISR program
- Restore the PCLATH register
- Restore the STATUS register
- Restore the W register

Since most instructions modify the W register, it must be saved immediately upon entering the ISR. The SWAPF instruction is used when saving and restoring the W and STATUS registers because it will not affect any bits in the STATUS register. It is useful to place W_TEMP in shared memory because the ISR cannot predict which bank will be selected when the interrupt occurs.

The processor will branch to the interrupt vector by loading the PC with 0004h. The PCLATH register will remain unchanged. This requires the ISR to ensure that the PCLATH register is set properly before using an instruction that causes PCLATH to be loaded into the PC. See **Section 2.3 "PCL and PCLATH"** for details on PC operation.

EXAMPLE 4-1: SAVING W, STATUS AND PCLATH REGISTERS IN RAM

```
MOVWFW_TEMP           ;Copy W to W_TEMP register
SWAPFSTATUS,W         ;Swap status to be saved into W
                      ;Swaps are used because they do not affect the status bits
BANKSELSTATUS_TEMP    ;Select regardless of current bank
MOVWFSTATUS_TEMP      ;Copy status to bank zero STATUS_TEMP register
MOVF    PCLATH,W      ;Copy PCLATH to W register
MOVWF   PCLATH_TEMP   ;Copy W register to PCLATH_TEMP
:
:(ISR)                ;Insert user code here
:
BANKSELSTATUS_TEMP    ;Select regardless of current bank
MOVF    PCLATH_TEMP,W ;
MOVWF   PCLATH        ;Restore PCLATH
SWAPFSTATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                      ;(sets bank to original state)
MOVWFSTATUS           ;Move W into STATUS register
SWAPFW_TEMP,F         ;Swap W_TEMP
SWAPFW_TEMP,W         ;Swap W_TEMP into W
```

REGISTER 6-1: PORTA: PORTA REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	RA5	RA4	RA3(1)	RA2	RA1	RA0
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 5-0 **RA<5:0>:** PORTA I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

Note 1: RA<3> is input only.

REGISTER 6-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	U-1	R/W-1	R/W-1	R/W-1
—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0
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 5-4 **TRISA<5:4>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

Note 1: TRISA<3> is unimplemented and read as 1.

REGISTER 6-3: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	WPUA3(2)	WPUA2	WPUA1	WPUA0
bit 7							bit 0

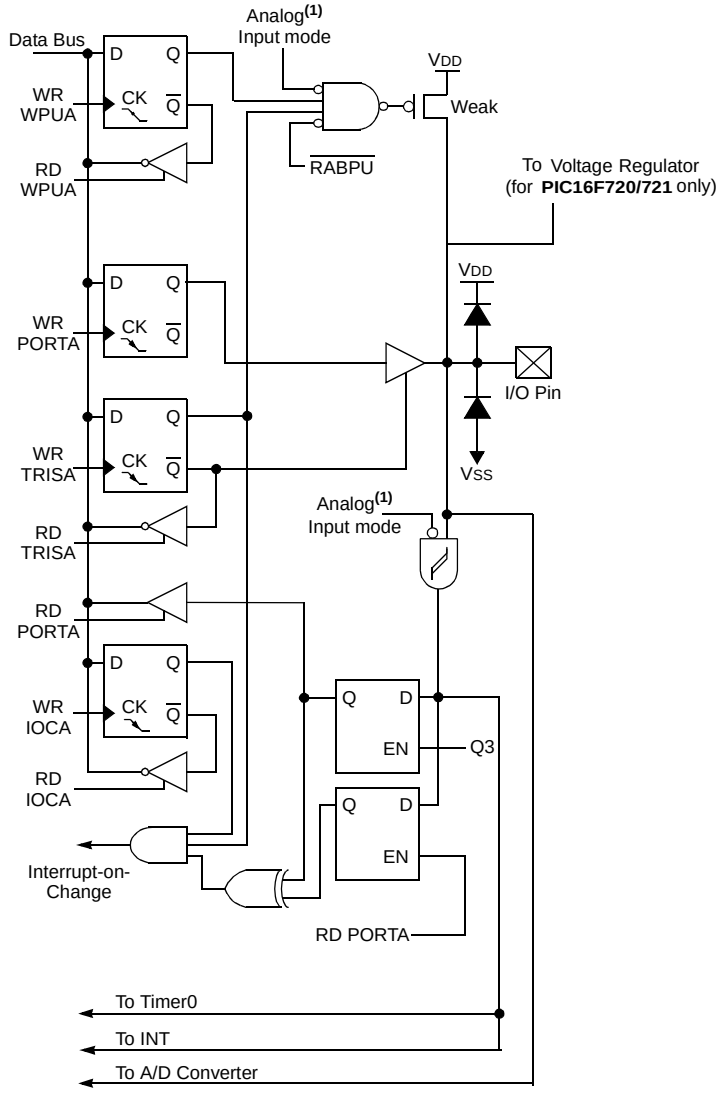
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

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

bit 5-0 **WPUA<5:0>:** Weak Pull-up PORTA Control bits
1 = Weak pull-up enabled(1)
0 = Weak pull-up disabled

Note 1: Enabling weak pull-ups also requires that the $\overline{\text{RABPU}}$ bit of the OPTION_REG register be cleared.
2: If MCLREN = 1, WPUA3 is always enabled.

FIGURE 6-3: BLOCK DIAGRAM OF RA2



Note 1: ANSEL determines Analog Input mode.

7.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- Internal clock source (INTOSC) is contained within the oscillator module and derived from a 500 kHz high-precision oscillator. The oscillator module has eight selectable output frequencies, with a maximum internal frequency of 16 MHz.
- The External Clock mode (EC) relies on an external signal for the clock source.

The system clock can be selected between external or internal clock sources via the FOSC bits of the Configuration Word 1.

7.3 Internal Clock Modes

The oscillator module has eight output frequencies derived from a 500 kHz high-precision oscillator. The IRCF bits of the OSCCON register select the postscaler applied to the clock source dividing the frequency by 1, 2, 4 or 8. Setting the PLLEN bit of the Configuration Word 1 locks the internal clock source to 16 MHz before the postscaler is selected by the IRCF bits. The PLLEN bit must be set or cleared at the time of programming; therefore, only the upper or low four clock source frequencies are selectable in software.

The internal oscillator block has one internal oscillator and a dedicated Phase-Locked Loop that are used to generate two internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 500 kHz (MFINTOSC). Both can be user-adjusted via software using the OSCTUNE register (Register 7-2).

7.3.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as system clock source when the device is programmed using the oscillator selection or the FOSC<1:0> bits in the CONFIG1 register. See **Section 8.0 “Device Configuration”** for more information.

In INTOSC mode, CLKIN is available for general purpose I/O. CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, Calibration, test or other application requirements.

In INTOSCIO mode, CLKIN and CLKOUT are available for general purpose I/O.

7.3.2 FREQUENCY SELECT BITS (IRCF)

The output of the 500 kHz MFINTOSC and 16 MHz HFINTOSC, with Phase-Locked Loop enabled, connect to a postscaler and multiplexer (see Figure 7-1). The Internal Oscillator Frequency Select bits (IRCF) of the OSCCON register select the frequency output of the internal oscillator. Depending upon the PLLEN bit, one of four frequencies of two frequency sets can be selected via software:

If PLLEN = 1, HFINTOSC frequency selection is as follows:

- 16 MHz
- 8 MHz (default after Reset)
- 4 MHz
- 2 MHz

If PLLEN = 0, MFINTOSC frequency selection is as follows:

- 500 kHz
- 250 kHz (default after Reset)
- 125 kHz
- 62.5 kHz

Note:	Following any Reset, the IRCF<1:0> bits of the OSCCON register are set to ‘10’ and the frequency selection is set to 8 MHz or 250 kHz. The user can modify the IRCF bits to select a different frequency.

There is no start-up delay before a new frequency selected in the IRCF bits takes effect. This is because the old and new frequencies are derived from INTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the Table 23-2 in **Section 23.0 “Electrical Specifications”**.

7.3.3 INTERNAL OSCILLATOR STATUS BITS

The internal oscillator (500 kHz) is a factory-calibrated internal clock source. The frequency can be altered via software using the OSCTUNE register (Register 7-2).

The Internal Oscillator Status Locked bit (ICSL) of the OSCCON register indicates when the internal oscillator is running within 2% of its final value.

The Internal Oscillator Status Stable bit (ICSS) of the OSCCON register indicates when the internal oscillator is running within 0.5% of its final value.

REGISTER 10-1: FVRCON: FIXED VOLTAGE REFERENCE REGISTER

R-q	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
FVRRDY	FVREN	TSEN	TSRNG	—	—	ADFVR1	ADFVR0
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
q = Value depends on condition			

bit 7 **FVRRDY(1):** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not active or stable
1 = Fixed Voltage Reference output is ready for use

bit 6 **FVREN:** Fixed Voltage Reference Enable bit
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
0 = Temperature indicator is disabled
1 = Temperature indicator is enabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = VOUT = VDD - 4VT (High Range)
0 = VOUT = VDD - 2VT (Low Range)

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

bit 1-0 **ADFVR<1:0>:** A/D Converter Fixed Voltage Reference Selection bits
00 = A/D Converter Fixed Voltage Reference Peripheral output is off
01 = A/D Converter Fixed Voltage Reference Peripheral output is 1x (1.024V)
10 = A/D Converter Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
11 = A/D Converter Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)

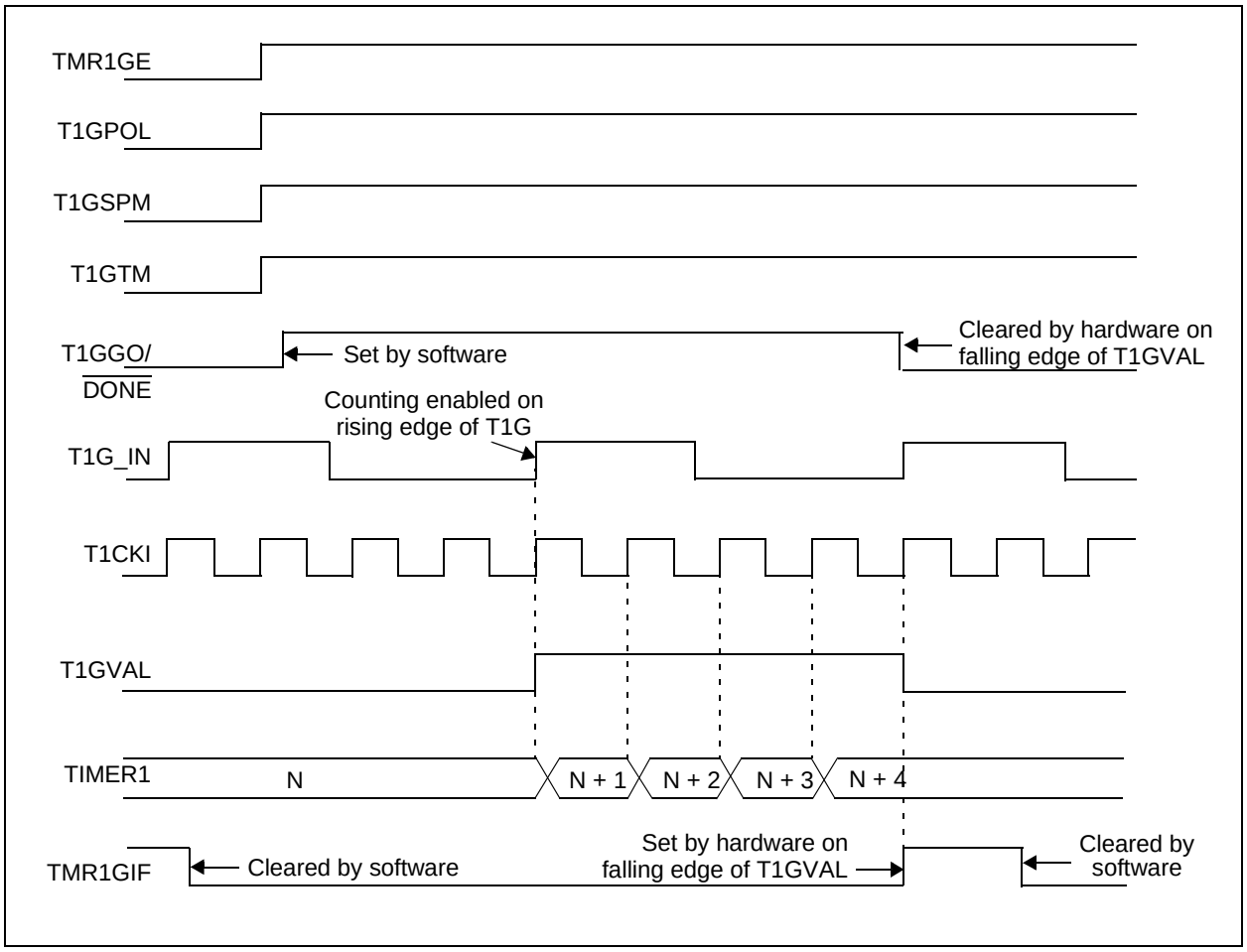
Note 1: FVRRDY is always '1' for the PIC16F720/721 devices.
2: Fixed Voltage Reference output cannot exceed VDD.
3: See **Section 11.0 "Temperature Indicator Module"** for additional information.

TABLE 10-2: SUMMARY OF ASSOCIATED FIXED VOLTAGE REFERENCE REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
FVRCON	FVRRDY	FVREN	TSEN	TSRNG	—	—	ADFVR1	ADFVR0	81

Legend: x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for Fixed Voltage Reference.

FIGURE 13-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



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 '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. Refer to **Section 16.1.2.4 "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 AUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. Refer to **Section 16.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

16.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the AUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

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

- RCIE interrupt enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit of the PIR1 register will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

16.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the AUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.

16.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by setting the AUSART by clearing the SPEN bit of the RCSTA register.

16.1.2.6 Receiving 9-bit Characters

The AUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set, the AUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

REGISTER 17-5: SSPMSK: SSP MASK REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
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 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect I^2C address match
0 = The received address bit n is not used to detect I^2C address match

bit 0 **MSK<0>:** Mask bit for I^2C Slave Mode, 10-bit Address
I^2C Slave mode, 10-bit Address (SSPM<3:0> = 0111):
1 = The received address bit '0' is compared to SSPADD<0> to detect I^2C address match
0 = The received address bit '0' is not used to detect I^2C address match
All other SSP modes: this bit has no effect.

REGISTER 17-6: SSPADD: SSP I^2C ADDRESS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
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 **ADD<7:0>:** Address bits
Received address

TABLE 17-3: REGISTERS ASSOCIATED WITH I^2C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	RABIE	TMR0IF	INTF	RABIF	37
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	38
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								131
SSPADD	ADD<7:0>								150
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	148
SSPMSK(2)	MSK<7:0>								150
SSPSTAT	SMP(1)	CKE(1)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	137
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	52

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by SSP module in I^2C mode.

Note 1: Maintain these bits clear in I^2C mode.
2: Accessible only when SSPM<3:0> = 1001.

18.5 Writing to Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT<1:0> of the Configuration Word Register 2. Flash program memory must be written in 32-word rows. See Figure 18-2 for more details. A row consists of 32 words with sequential addresses, with a lower boundary defined by an address, where PMADR<4:0>= 00000. All row writes to program memory are done as 32-word erase and one to 32-word write operations. The write operation is edge-aligned. Crossing boundaries is not recommended, as the operation will only affect the new boundary, wrapping the data values at the same time. Once the write control bit is set, the Program Memory (PM) controller will immediately write the data. Program execution is stalled while the write is in progress.

To erase a program memory row, the address of the row to erase must be loaded into the PMADRH:PMADRL register pair. A row consists of 32 words so, when selecting a row, PMADR<4:0> are ignored. After the Address has been set up, then the following sequence of events must be executed:

1. Set the WREN and FREE control bits of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set the WR control bit of the PMCON1 register.

To write program data, it must first be loaded into the buffer latches (see Figure 18-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATA and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Set the WREN control bit of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set the WR control bit of the PMCON1 register.

All 32 buffer register locations should be written to with correct data. If less than 32 words are being written to in the block of 32 words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then, the sequence of events to transfer data to the buffer registers must be executed.

When the LWLO bit is '1', the write sequence will only load the buffer register and will not actually initiate the write to program Flash:

1. Set the WREN and LWLO bits of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
3. Set control bit WR of the PMCON1 register to begin the write operation.

Note:	Self-write execution to Flash memory cannot be done while running in low power PFM and Voltage Regulator modes. Therefore, executing a self-write will put the PFM and voltage regulator into High Power mode for the duration of the sequence.

To transfer data from the buffer registers to the program memory, the last word to be written should be written to the PMDATH:PMDATL register pair. Then, the following sequence of events must be executed:

1. Clear the LWLO bit of the PMCON1 Register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set control bit WR of the PMCON1 register to begin the write operation.
4. Two NOP instructions must follow the setting of the WR bit.

This is necessary to provide time for the address and to be provided to the Program Flash Memory to be put in the write latches.

Note:	An ICD break that occurs during the 55h - AAh – Set WR bit sequence will interrupt the timing of the sequence and prevent the unlock sequence from occurring. In this case, no write will be initiated, as there was no operation to complete.

No automatic erase occurs upon the initiation of the write; if the program Flash needs to be erased before writing, the row (32 words) must be previously erased.

After the "BSF PMCON1, WR" instruction, the processor requires two cycles to set up the erase/write operation.

The user must place two NOP instructions after the WR bit is set. These two instructions will also be forced in hardware to NOP, but if an ICD break occurs at this point, the forcing to NOP will be lost.

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	—	CFGS	LWLO	FREE	—	WREN	WR	RD	155
PMCON2	Program Memory Control Register 2 (not a physical register)								—
PMADRH	—	—	—	Program Memory Read Address Register High Byte					156
PMADRL	Program Memory Read Address Register Low Byte								157
PMDATH	—	—	Program Memory Read Data Register High Byte						156
PMDATL	Program Memory Read Data Register Low Byte								156

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

TABLE 23-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param. No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CC01*	TccL	CCP Input Low Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCP Input High Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCP Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 23-7: PIC16F720/721 A/D CONVERTER (ADC) CHARACTERISTICS

Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD01	NR	Resolution	—	—	8	bit	
AD02	EIL	Integral Error	—	—	±1.7	LSb	VDD = 3.0V
AD03	EDL	Differential Error	—	—	±1	LSb	No missing codes VDD = 3.0V
AD07	EGN	Gain Error	—	—	±1.5	LSb	VDD = 3.0V
AD07	VAIN	Full-Scale Range	VSS	—	VDD	V	
AD08*	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 23-13: I^2C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
100*	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	—		
101*	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	—		
102*	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
103*	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
90*	TSU:STA	Start condition setup time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91*	THD:STA	Start condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92*	TSU:STO	Stop condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110*	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	
	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) I^2C bus device can be used in a Standard mode (100 kHz) I^2C bus system, but the requirement $T_{SU:DAT} \geq 250$ 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 T_R max. + $T_{SU:DAT}$ = 1000 + 250 = 1250 ns (according to the Standard mode I^2C bus specification), before the SCL line is released.

FIGURE 24-5: PIC16F720/721 MAX. IDD vs. FOSC OVER VDD, MFINTOSC

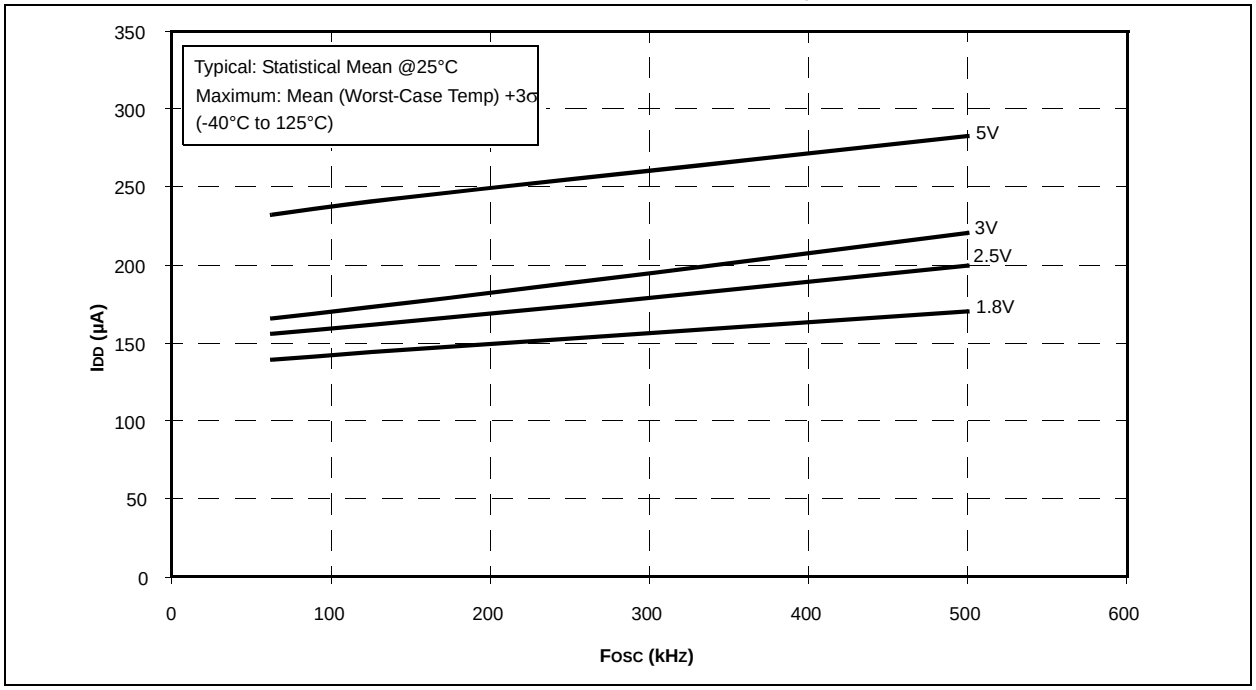


FIGURE 24-6: PIC16F720/721 TYPICAL IDD vs. FOSC OVER VDD, MFINTOSC

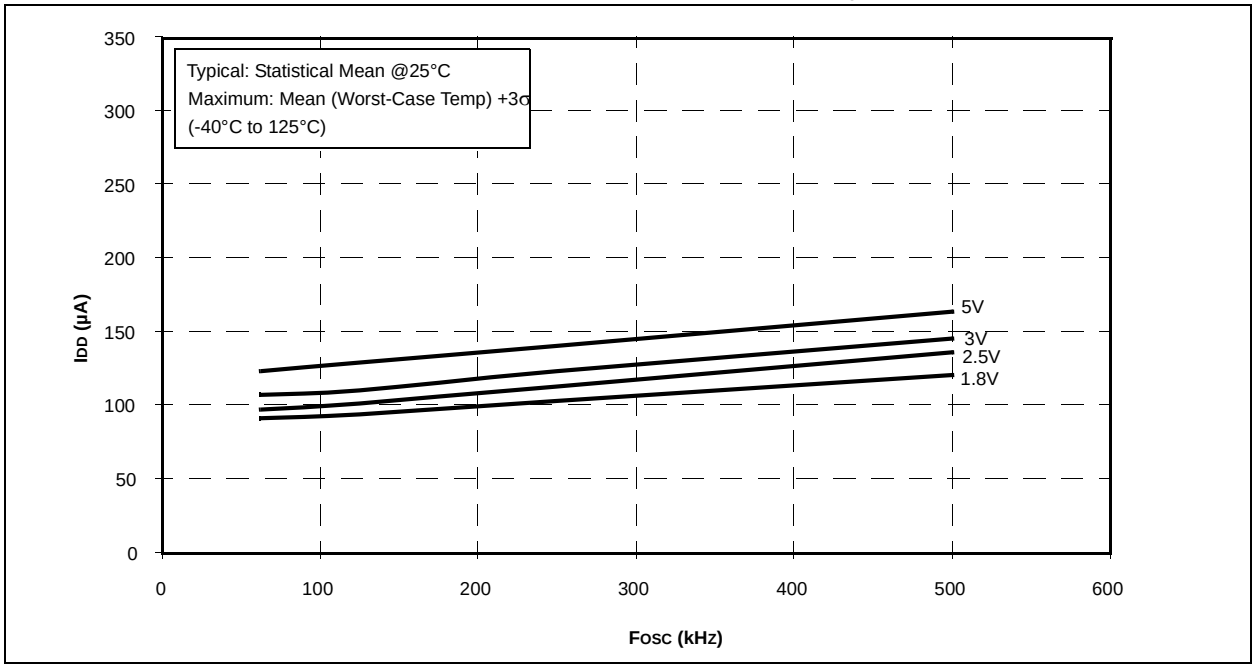


FIGURE 24-22: TTL INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

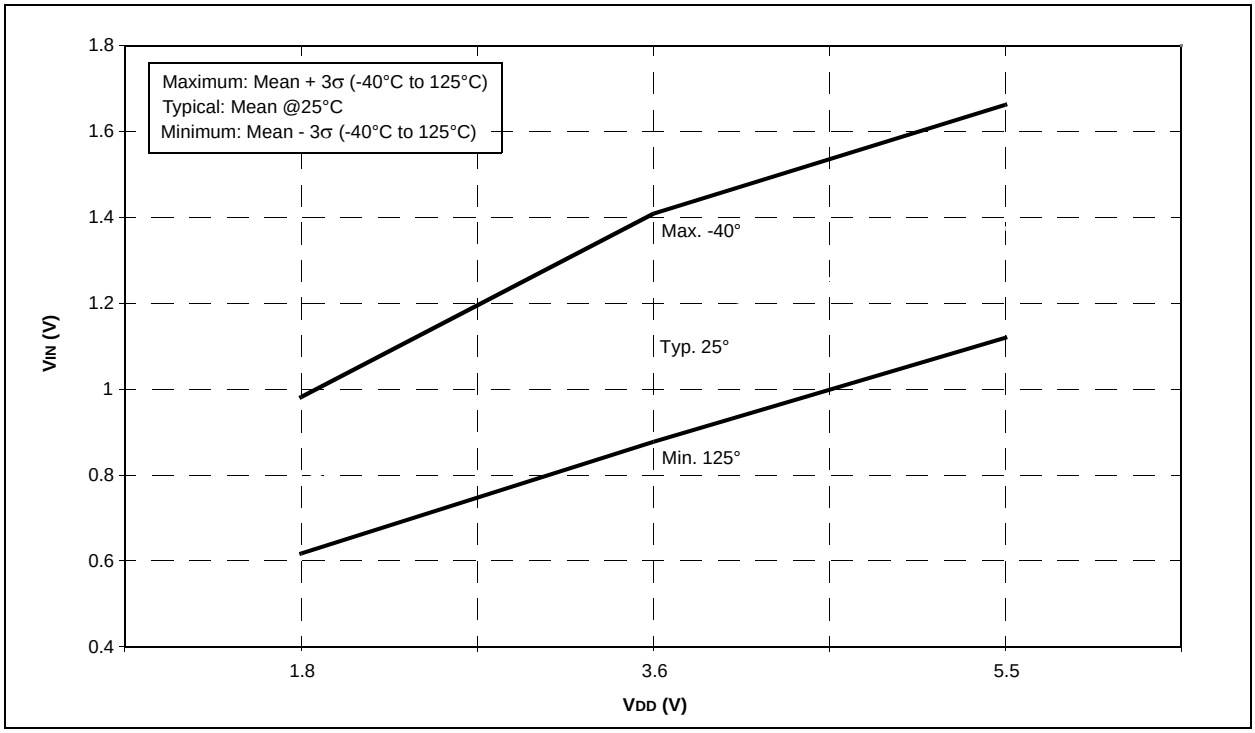
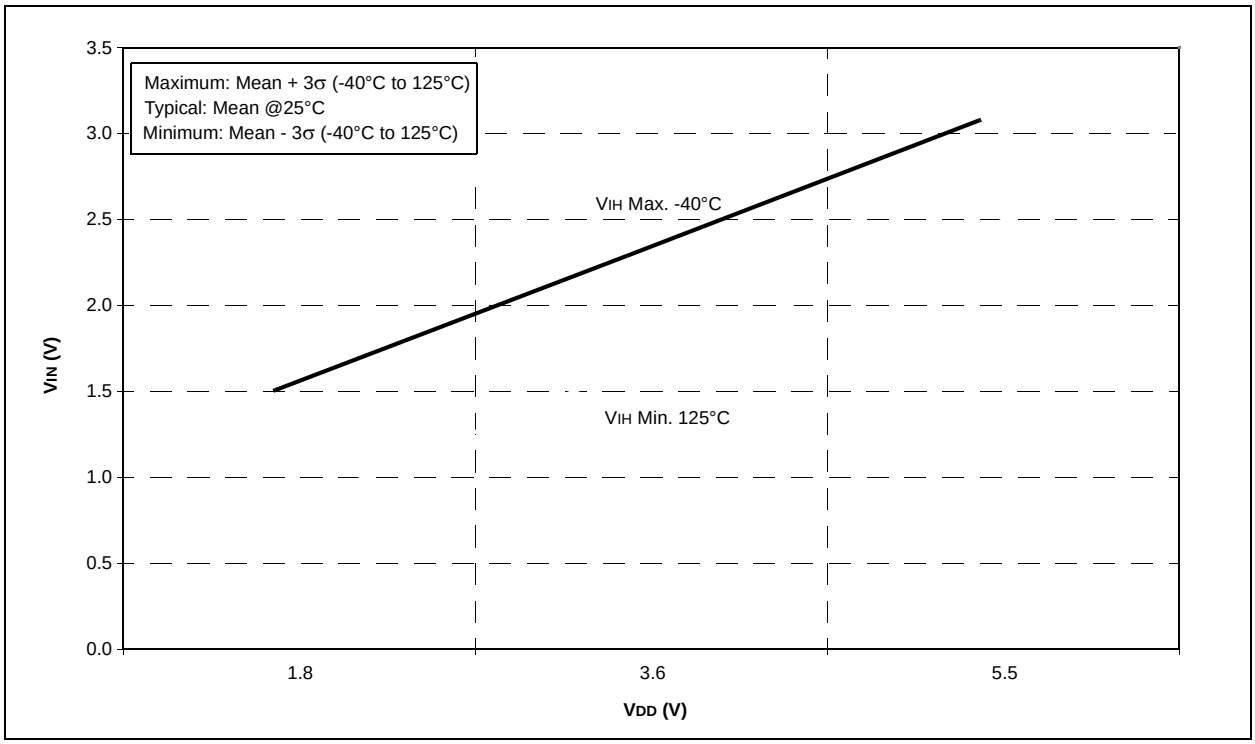


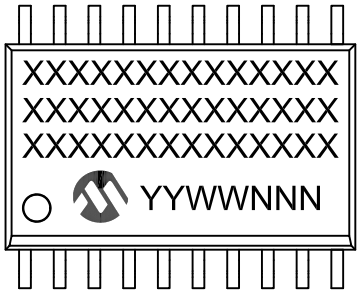
FIGURE 24-23: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE



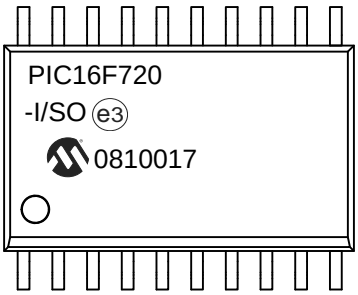
NOTES:

25.1 Package Marking Information

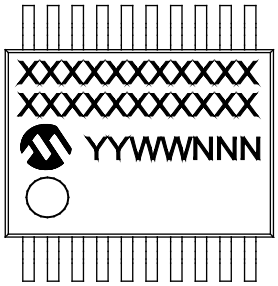
20-Lead SOIC (7.50 mm)



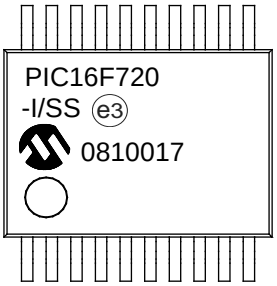
Example



20-Lead SSOP (5.30 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

***** Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (September 2010)

Original release of this document.

Revision B (March 2011)

Updated the Electrical Specifications section.

Revision C (September 2011)

Reviewed title; Updated Table 1 and Table 1-1; Reviewed the Memory Organization section; Updated Section 3.6, Figures 3-4 and 3-5, Register 4-1 and Figure 4-2; Updated Registers 8-1 and 8-2; Reviewed the Oscillator Module section; Updated Table 10-1, Figures 11-1, 12-1 and Register 18-1; Updated the Summary of Registers Tables; Updated the Electrical Specifications section; Updated the DC and AC Characteristics Graphs and Charts section; Updated the Packaging Information section; Updated the Product Identification System section.

Revision D (February 2013)

Updated Table 1-1, Table 15-4 and Table 16-5; Updated the Electrical Specifications section; Updated the DC and AC Characteristics Graphs and Charts section; Other minor corrections.

Revision E (August 2013)

Deleted Example 18-2; Revised Table 23-7.

Revision F (December 2015)

Updated Table 2-1 and Table 23-7; Updated Register 7-1; Added 7.3.3 Section; Other corrections.

APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This shows a comparison of features in the migration from another PIC® device, the PIC16F720, to the PIC16F721 device.

B.1 PIC16F690 to PIC16F721

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F690	PIC16F721
Max. Operating Speed	20 MHz	16 MHz
Max. Program Memory (Words)	4K	4K
Max. SRAM (Bytes)	256	256
A/D Resolution	10-bit	8-bit
Timers (8/16-bit)	2/1	2/1
Oscillator Modes	8	4
Brown-out Reset	Y	Y
Internal Pull-ups	RA<5:0>, RB<7:4>	RA<5:0>, RB<7:4>
Interrupt-on-change	RA<5:0>, RB<7:4>	RA<5:0>, RB<7:4>
Comparator	2	0
EUSART	Y	Y
Extended WDT	Y	N
Software Control Option of WDT/BOR	Y	N
INTOSC Frequencies	31 kHz - 8 MHz	500 kHz - 16 MHz
Pin Count	20	20

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.