



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	18
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	·
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f631t-e-ss

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

## TABLE 1-1: PINOUT DESCRIPTION – PIC16F631

Name	Function	Input Type	Output Type	Description
RA0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	C1IN+	AN		Comparator C1 non-inverting input.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
	ULPWU	AN	_	Ultra Low-Power Wake-up input.
RA1/C12IN0-/ICSPCLK	RA1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	C12IN0-	AN	_	Comparator C1 or C2 inverting input.
	ICSPCLK	ST	_	ICSP™ clock.
RA2/T0CKI/INT/C1OUT	RA2	ST	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	T0CKI	ST	_	Timer0 clock input.
	INT	ST	_	External interrupt pin.
	C1OUT	_	CMOS	Comparator C1 output.
RA3/MCLR/Vpp	RA3	TTL	-	General purpose input. Individually controlled interrupt-on- change.
	MCLR	ST	_	Master Clear with internal pull-up.
	Vpp	ΗV	_	Programming voltage.
RA4/T1G/OSC2/CLKOUT	RA4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	T1G	ST	_	Timer1 gate input.
	OSC2		XTAL	Crystal/Resonator.
	CLKOUT	_	CMOS	Fosc/4 output.
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	T1CKI	ST	—	Timer1 clock input.
	OSC1	XTAL	_	Crystal/Resonator.
	CLKIN	ST	—	External clock input/RC oscillator connection.
RB4	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
RB5	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
RB6	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
RB7	RB7	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
RC0/C2IN+	RC0	ST	CMOS	General purpose I/O.
	C2IN+	AN	—	Comparator C2 non-inverting input.
RC1/C12IN1-	RC1	ST	CMOS	General purpose I/O.
	C12IN1-	AN	—	Comparator C1 or C2 inverting input.
RC2/C12IN2-	RC2	ST	CMOS	General purpose I/O.
	C12IN2-	AN	—	Comparator C1 or C2 inverting input.
RC3/C12IN3-	RC3	ST	CMOS	General purpose I/O.
	C12IN3-	AN	_	Comparator C1 or C2 inverting input.
RC4/C2OUT	RC4	ST	CMOS	General purpose I/O.
	C2OUT	_	CMOS	Comparator C2 output.
RC5	RC5	ST	CMOS	General purpose I/O.
Legend: AN = Analog input	or output	CMOS	=CMOS	compatible input or output
IIL = TTL compat HV = High Voltage	ble input	ST= XTAL=	Schmitt Crvstal	I rigger input with CMOS levels

## 3.5.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See **Section 3.5.4 "Frequency Select Bits (IRCF)**" for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<2:0> bits of the OSCCON register = 000) as the system clock source (SCS bit of the OSCCON register = 1), or when any of the following are enabled:

- Two-Speed Start-up IESO bit of the Configuration Word register = 1 and IRCF<2:0> bits of the OSCCON register = 000
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit of the OSCCON register indicates whether the LFINTOSC is stable or not.

### 3.5.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz (LFINTOSC)

Note:	Following any Reset, the IRCF<2:0> bits
	of the OSCCON register are set to '110'
	and the frequency selection is set to
	4 MHz. The user can modify the IRCF bits
	to select a different frequency.

### 3.5.5 HFINTOSC AND LFINTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power (see Figure 3-6). If this is the case, there is a delay after the IRCF<2:0> bits of the OSCCON register are modified before the frequency selection takes place. The LTS and HTS bits of the OSCCON register will reflect the current active status of the LFINTOSC and HFINTOSC oscillators. The timing of a frequency selection is as follows:

- 1. IRCF<2:0> bits of the OSCCON register are modified.
- 2. If the new clock is shut down, a clock start-up delay is started.
- 3. Clock switch circuitry waits for a falling edge of the current clock.
- 4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
- CLKOUT is now connected with the new clock. LTS and HTS bits of the OSCCON register are updated as required.
- 6. Clock switch is complete.

See Figure 3-1 for more details.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the oscillator tables of **Section 17.0** "**Electrical Specifications**".

#### 3.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or the internal oscillator.

FIGURE 3-7:	TWO-SPEED START-UP	
HFINTOSC /		
OSC1	←Tost	
OSC2		
Program Counter	PC-N (PC	XPC + 1X
System Clock		

#### 4.0 I/O PORTS

There are as many as eighteen general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

#### 4.1 **PORTA and the TRISA Registers**

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 4-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). The exception is RA3, which is input only and its TRIS bit will always read as '1'. Example 4-1 shows how to initialize PORTA.

Reading the PORTA register (Register 4-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write

#### **REGISTER 4-1:** PORTA: PORTA REGISTER

operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch. RA3 reads '0' when MCLRE = 1.

The TRISA register controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

The ANSEL register must be initialized to Note: configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

#### EXAMPLE 4-1: **INITIALIZING PORTA**

BCF	STATUS, RP	0;Bank 0
BCF	STATUS, RP	1;
CLRF	PORTA	;Init PORTA
BSF	STATUS, RP	1;Bank 2
CLRF	ANSEL	;digital I/O
BSF	STATUS, RP	0;Bank 1
BCF	STATUS, RP	1;
MOVLW	0Ch	;Set RA<3:2> as inputs
MOVWF	TRISA	;and set RA<5:4,1:0>
		;as outputs
BCF	STATUS, RP	0;Bank 0
	BCF BCF CLRF BSF BCF MOVLW MOVWF BCF	BCFSTATUS, RPBCFSTATUS, RPCLRFPORTABSFSTATUS, RPCLRFANSELBSFSTATUS, RPBCFSTATUS, RPMOVLWOChMOVWFTRISABCFSTATUS, RP

U-0	U-0	R/W-x	R/W-x	R-x	R/W-x	R/W-x	R/W-x
—	—	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0
Logond							

Legena:			
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 > VIн
	0 = Port pin is < VIL

#### **REGISTER 4-2:** TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead 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'

TRISA<5:0>: 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> always reads '1'.

bit 5-0

TRISA<5:4> always reads '1' in XT, HS and LP Oscillator modes. 2:

# 7.0 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 7-1 for a block diagram of Timer2.

# 7.1 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 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, MCLR Reset, Watchdog Timer Reset or Brown-out Reset).

Note: TMR2 is not cleared when T2CON is written.



### 9.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 9-3 shows the two output formats.

### FIGURE 9-3: 10-BIT A/D CONVERSION RESULT FORMAT



### EXAMPLE 9-1: A/D CONVERSION

;This code block configures the ADC ; for polling, Vdd reference, Frc clock ;and AN0 input. ; ;Conversion start & polling for completion ; are included. BANKSELADCON1; MOVLWB'01110000'; ADC Frc clock MOVWFADCON1; BANKSELTRISA; BSF TRISA,0;Set RA0 to input BANKSELANSEL; BSF ANSEL, 0; Set RA0 to analog BANKSELADCON0; MOVLWB'10000001';Right justify, MOVWFADCON0; Vdd Vref, AN0, On CALLSampleTime;Acquisiton delay BSF ADCON0,GO;Start conversion BTFSCADCON0,GO;Is conversion done? GOTO\$-1;No, test again BANKSELADRESH; MOVFADRESH, W; Read upper 2 bits MOVWFRESULTHI; store in GPR space BANKSELADRESL; MOVFADRESL,W;Read lower 8 bits MOVWFRESULTLO; Store in GPR space

### 9.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

## 11.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCP1/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 11-6). This mode can be used for Half-Bridge applications, as shown in Figure 11-9, or for Full-Bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in Half-Bridge power devices. The value of the PDC<6:0> bits of the PWM1CON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 11.4.6 "Programmable Dead-Band Delay mode"** for more details of the dead-band delay operations. Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.





# FIGURE 11-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS



## FIGURE 11-15: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PRSEN = 0)



### 11.4.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PRSEN bit in the PWM1CON register.

If auto-restart is enabled, the ECCPASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the ECCPASE bit will be cleared via hardware and normal operation will resume.

#### FIGURE 11-16: PWM AUTO-SHUTDOWN WITH AUTO-RESTART ENABLED (PRSEN = 1)





# FIGURE 11-19: SIMPLIFIED STEERING BLOCK DIAGRAM

# 13.3 Enabling SPI I/O

To enable the serial port, SSP Enable bit SSPEN of the SSPCON register must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRISB and TRISC registers) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<7> bit cleared
- SCK (Master mode) must have TRISB<6> bit cleared
- SCK (Slave mode) must have TRISB<6> bit set
- SS must have TRISC<6> bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRISB and TRISC) registers to the opposite value.

# 13.4 Typical Connection

Figure 13-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data Slave sends dummy data
- Master sends data Slave sends data
- Master sends dummy data Slave sends data



#### FIGURE 13-2: SPI MASTER/SLAVE CONNECTION

### 13.12.4 TRANSMISSION

When the  $R/\overline{W}$  bit of the incoming address byte is set and an address match occurs, the  $R/\overline{W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit, and pin RB6/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RB6/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 13-10). An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the ACK pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not ACK), then the data transfer is complete. When the ACK is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave then monitors for another occurrence of the Start bit. If the SDA line was low (ACK), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RB6/SCK/SCL should be enabled by setting bit CKP.





# 15.2 Instruction Descriptions

ADDLW	Add literal and W
Syntax:	[label] ADDLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \to (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

BCF	Bit Clear f
Syntax:	[ <i>label</i> ]BCF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$0 \rightarrow (f < b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

ADDWF	Add W and f					
Syntax:	[label] ADDWF f,d					
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$					
Operation:	(W) + (f) $\rightarrow$ (destination)					
Status Affected:	C, DC, Z					
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.					

BSF	Bit Set f				
Syntax:	[label]BSF f,b				
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$				
Operation:	$1 \rightarrow (f < b >)$				
Status Affected:	None				
Description:	Bit 'b' in register 'f' is set.				

ANDLW	AND literal with W					
Syntax:	[ <i>label</i> ] ANDLW k					
Operands:	$0 \leq k \leq 255$					
Operation:	(W) .AND. (k) $\rightarrow$ (W)					
Status Affected:	Z					
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W reg- ister.					

BTFSC	Bit Test f, Skip if Clear				
Syntax:	[ <i>label</i> ] BTFSC f,b				
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$				
Operation:	skip if (f <b>) = <math>0</math></b>				
Status Affected:	None				
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded, and a $NOP$ is executed instead, making this a 2-cycle instruction.				

Syntax:	[ <i>label</i> ] ANDWF f,d						
Operands:	$0 \le f \le 127$ $d \in [0,1]$						
Operation:	(W) .AND. (f) $\rightarrow$ (destination)						
Status Affected:	Z						
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.						

AND W with f

ANDWF

MOVF	Move f						
Syntax:	[ <i>label</i> ] MOVF f,d						
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$						
Operation:	(f) $\rightarrow$ (dest)						
Status Affected:	Z						
Description:	The contents of register 'f' is moved to a destination dependent upon the status of 'd'. If $d = 0$ , destination is W register. If $d = 1$ , the destination is file register 'f' itself. $d = 1$ is useful to test a file register since status flag Z is affected						
Words:	1						
Cycles:	1						
Example:	MOVF FSR, 0						
	After Instruction W = value in FSR register Z = 1						

MOVWF	Move W to f					
Syntax:	[label] MOVWF f					
Operands:	$0 \leq f \leq 127$					
Operation:	$(W) \rightarrow (f)$					
Status Affected:	None					
Description:	Move data from W register to register 'f'.					
Words:	1					
Cycles:	1					
Example:	MOVW OPTION F					
	Before Instruction					
	OPTION = 0xFF					
	W = 0x4F					
	After Instruction					
	OPTION = 0x4F					
	W = 0x4F					

MOVLW	Move literal to W							
Syntax:	[ <i>label</i> ] MOVLW k							
Operands:	$0 \le k \le 255$							
Operation:	$k \rightarrow (W)$							
Status Affected:	None							
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.							
Words:	1							
Cycles:	1							
Example:	MOVLW 0x5A							
	After Instruction W = 0x5A							

NOP	No Operation				
Syntax:	[label] NOP				
Operands:	None				
Operation:	No operation				
Status Affected:	None				
Description:	No operation.				
Words:	1				
Cycles:	1				
Example:	NOP				

FIGURE 17-1: PIC16F631/677/685/687/689/690 VOLTAGE-FREQUENCY GRAPH, -40°C  $\leq$  TA  $\leq$  +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.





## TABLE 17-2: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$								
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Тур†	Max.	Units	Conditions
OS06	Twarm	Internal Oscillator Switch when running <sup>(3)</sup>	—		_	2	Tosc	Slowest clock
OS07	Tsc	Fail-Safe Sample Clock Period <sup>(1)</sup>	—		21	—	ms	LFINTOSC/64
OS08 HFosc	Internal Calibrated HFINTOSC Frequency <sup>(2)</sup>	±1%	7.92	8.0	8.08	MHz	VDD = 3.5V, 25°C	
		±2%	7.84	8.0	8.16	MHz	$2.5V \le VDD \le 5.5V$ , 0°C $\le$ TA $\le$ +85°C	
			±5%	7.60	8.0	8.40	MHz	$\begin{array}{l} 2.0V \leq V \text{DD} \leq 5.5 \text{V}, \\ -40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C} \text{ (Ind.)}, \\ -40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C} \text{ (Ext.)} \end{array}$
OS09*	LFosc	Internal Uncalibrated LFINTOSC Frequency	—	15	31	45	kHz	
OS10*	TIOSC ST	HFINTOSC Oscillator	—	5.5	12	24	μS	VDD = 2.0V, -40°C to +85°C
		Wake-up from Sleep	—	3.5	7	14	μs	VDD = 3.0V, -40°C to +85°C
			—	3	6	11	μS	VDD = 5.0V, -40°C to +85°C

\* 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.

**Note 1:** Instruction cycle period (TcY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at 'min' values with an external clock applied to the OSC1 pin. When an external clock input is used, the 'max' cycle time limit is 'DC' (no clock) for all devices.

2: To ensure these oscillator frequency tolerances, VDD and Vss must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

3: By design.















#### 16 Typical: Statistical Mean @25°C 14 Maximum: Mean (Worst-case Temp) + 3o (-40°C to 125°C) 85°C 12 25°C 10-Time (µs) -40°C 8 6 4 2 0 2.5 3.0 3.5 4.5 5.0 5.5 2.0 4.0 VDD (V)

#### FIGURE 18-36: TYPICAL HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE









