Microchip Technology - PIC16LF720-I/SO Datasheet





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	16MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	3.5КВ (2К х 14)
Program Memory Type	FLASH
EEPROM Size	- ·
RAM Size	128 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/pic16lf720-i-so

Email: info@E-XFL.COM

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

4.0 INTERRUPTS

The PIC16(L)F720/721 device family features an interruptible core, allowing certain events to preempt normal program flow. An Interrupt Service Routine (ISR) is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

The PIC16(L)F720/721 device family has 11 interrupt sources, differentiated by corresponding interrupt enable and flag bits:

- Timer0 Overflow Interrupt
- External Edge Detect on INT Pin Interrupt
- · Interrupt-on-change, PORTA and PORTB pins
- Timer1 Gate Interrupt
- A/D Conversion Complete Interrupt
- AUSART Receive Interrupt
- AUSART Transmit Interrupt
- SSP Event Interrupt
- CCP1 Event Interrupt
- Timer2 Match with PR2 Interrupt
- Timer1 Overflow Interrupt

A block diagram of the interrupt logic is shown in Figure 4-1.

FIGURE 4-1: INTERRUPT LOGIC



FIGURE 6-1: BLOCK DIAGRAM OF RA0



6.2.4 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTB pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the SSP, I²C or interrupts, refer to the appropriate section in this data sheet.

6.2.4.1 RB4/AN10/SDI/SDA

Figure 6-7 shows the diagram for this pin. The RB4 pin is configurable to function as one of the following:

- General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
- Analog input for the A/D
- Synchronous Serial Port Input (SPI)
- I²C data I/O

6.2.4.2 RB5/AN11/RX/DT

Figure 6-8 shows the diagram for this pin. The RB5 pin is configurable to function as one of the following:

- General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
- Analog input for the A/D
- USART asynchronous receive
- USART synchronous receive

6.2.4.3 RB6/SCK/SCL

Figure 6-9 shows the diagram for this pin. The RB6 pin is configurable to function as one of the following:

- General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
- Synchronous Serial Port clock for both SPI and $\rm I^2C$

6.2.4.4 RB7/TX/CK

Figure 6-10 shows the diagram for this pin. The RB7 pin is configurable to function as one of the following:

- General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
- USART asynchronous transmit
- USART synchronous clock

FIGURE 6-7: BLOCK DIAGRAM OF RB4



13.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Synchronous or asynchronous operation
- Multiple Timer1 gate (count enable) sources
- · Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity

- Gate Toggle Mode
- Gate Single Pulse Mode
- · Gate Value Status
- Gate Event Interrupt

Figure 13-1 is a block diagram of the Timer1 module.



14.2 Timer2 Control Register

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplem	nented bit, read	d as '0'	
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
bit 7	Unimplemen	ted: Read as '	0'				
bit 6-3	TOUTPS<3:0)>: Timer2 Outp	out Postscaler	Select bits			
	0000 = 1:1 F	Postscaler					
	0001 = 1:2 F	Postscaler					
	0010 = 1:3 F	Postscaler					
	0011 = 1:4 F	Postscaler					
	0100 = 1:5 F	Postscaler					
	0101 = 1:6 F	Postscaler					
	0110 = 1.7 F	ostscaler					
	0111 = 1.8 F						
	1000 = 1.9 F	Postscaler					
	1001 = 1.10 1010 = 1.11	Postscaler					
	1010 = 1:12	Postscaler					
	1100 = 1:13	Postscaler					
	1101 = 1:14	Postscaler					
	1110 = 1 : 1 5	Postscaler					
	1111 = 1:16	Postscaler					
bit 2	TMR2ON: Tir	mer2 On bit					
	1 = Timer2 is	s On					
	0 = Timer2 is	s Off					
bit 1-0	T2CKPS<1:0	>: Timer2 Cloc	k Prescale Sel	ect bits			
	00 = Presca	ller is 1					
	01 = Presca	ler is 4					
	1x = Presca	ller is 16					

REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER

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
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	38
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PR2	Timer2 module Period Register					98			
TMR2	Timer2 module Register				98				
T2CON	_	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	99
\mathbf{I} are not used for Timer?									

TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

15.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCP1 pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPR1L
- CCP1CON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCP1 pin.

Figure 15-3 shows a simplified block diagram of PWM operation.

Figure 15-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, refer to **Section 15.3.8** "Setup for PWM Operation".

FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



The PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle).

FIGURE 15-4: CCP PWM OUTPUT



15.3.1 CCPx PIN CONFIGURATION

In PWM mode, the CCP1 pin is multiplexed with the PORT data latch. The user must configure the CCP1 pin as an output by clearing the associated TRIS bit.

Note: Clearing the CCP1CON register will relinquish CCP1 control of the CCP1 pin.

FIGURE 16-2: AUSART RECEIVE BLOCK DIAGRAM



The operation of the AUSART module is controlled through two registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)

These registers are detailed in Register 16-1 and Register 16-2, respectively.

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.

17.2.6 TRANSMISSION

When the R/W bit of the received address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set and the slave will respond to the master by reading out data. After the address match, an ACK pulse is generated by the slave hardware and the SCL pin is held low (clock is automatically stretched) until the slave is ready to respond. See **Section 17.2.7 "Clock Stretching"**. The data the slave will transmit must be loaded into the SSPBUF register, which sets the BF bit. The SCL line is released by setting the CKP bit of the SSPCON register.

An SSP interrupt is generated for each transferred data byte. The SSPIF flag bit of the PIR1 register initiates an SSP interrupt, and must be cleared by software before the next byte is transmitted. The BF bit of the SSPSTAT register is cleared on the falling edge of the eighth received clock pulse. The SSPIF flag bit is set on the falling edge of the ninth clock pulse. Following the eighth falling clock edge, control of the SDA line is released back to the master so that the master can acknowledge or not acknowledge the response. If the master sends a not acknowledge, the slave's transmission is complete and the slave must monitor for the next Start condition. If the master acknowledges, control of the bus is returned to the slave to transmit another byte of data. Just as with the previous byte, the clock is stretched by the slave, data must be loaded into the SSPBUF and CKP must be set to release the clock line (SCL).



FIGURE 17-12: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)

17.2.10 CLOCK SYNCHRONIZATION

When the CKP bit is cleared, the SCL output is held low once it is sampled low. Therefore, the CKP bit will not stretch the SCL line until an external I^2C master device has already asserted the SCL line low. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (Figure 17-14).

17.2.11 SLEEP OPERATION

While in Sleep mode, the I^2C module can receive addresses of data, and when an address match or complete byte transfer occurs, wake the processor from Sleep (if SSP interrupt is enabled).



FIGURE 17-14: CLOCK SYNCHRONIZATION TIMING



FIGURE 18-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION – NORMAL MODE

18.2 **Code Protection**

When the device is code-protected, the CPU may continue to read and write the Flash program memory. Depending on the settings of the Flash program memory enable (WRT<1:0>) bits, the device may or may not be able to write certain blocks of the program memory. However, reads of the program memory are allowed.

When the Flash program memory Code Protection (CP) bit in the Configuration Word register is enabled, the program memory is code-protected, and the device programmer (ICSP™) cannot access data or program memory.

Note:	Code-protect does not affect the CPU
	from performing a read operation on the
	program memory. For more information,
	refer to Section 8.2 "Code Protection".

PMADRH and PMADRL Registers 18.3

The PMADRH: PMADRL register pair can address up to a maximum of 4K words of program Flash. The Most Significant Byte (MSB) of the address is written to the PMADRH register and the Least Significant Byte (LSB) is written to the PMADRL register.

PMCON1 and PMCON2 Registers 18.4

PMCON1 is the control register for the data program memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, but only set in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental premature termination of a write operation. Setting the control bit WR initiates a write operation. For program memory writes, WR initiates a write cycle if FREE = 0 and an erase cycle if FREE = 1.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. PMCON2 is not a physical register. Reading PMCON2 will read all '0's. The PMCON2 register is used exclusively in the Flash memory write sequence.

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 ${\tt NOP}$ 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.

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.

21.2	Instruction	Descriptions
------	-------------	--------------

Bit Clear f
[label]BCF f,b
$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
$0 \rightarrow (f < b >)$
None
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:	$0 \le k \le 255$				
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:	[label] BTFSC f,b						
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$						
Operation:	skip if (f) = 0						
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.						

ANDWF	AND W with f					
Syntax:	[label] 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'.					

22.0 DEVELOPMENT SUPPORT

The PIC[®] microcontrollers (MCU) and dsPIC[®] digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
- MPLAB[®] X IDE Software
 Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/ MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
 - MPLAB X SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICkit[™] 3
- Device Programmers
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

22.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows[®], Linux and Mac OS[®] X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window
- Project-Based Workspaces:
- Multiple projects
- Multiple tools
- Multiple configurations
- · Simultaneous debugging sessions
- File History and Bug Tracking:
- Local file history feature
- Built-in support for Bugzilla issue tracker

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

Standard Operating Conditions (unless otherwise stated)Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$								
Param. No.	Sym.	Characteristic		Min.	Тур†	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		<u>3Tcy + 40</u> 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.	Тур†	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.

Param. No.	Symbol	Characteristic	Min.	Тур†	Max.	Units	Conditions	
SP70*	TssL2scH, TssL2scL	\overline{SS} to SCK \downarrow or SCK \uparrow input	Тсү		—	ns		
SP71*	TscH	SCK input high time (Slave mode)		TCY + 20	_	-	ns	
SP72*	TscL	SCK input low time (Slave mode)		TCY + 20	—	-	ns	
SP73*	TDIV2SCH, TDIV2SCL	Setup time of SDI data input to SCK edge		100		—	ns	
SP74*	TscH2DIL, TscL2DIL	Hold time of SDI data input to SCK	100	_	—	ns		
SP75*	P75* TDOR SDO data output rise time		3.0-5.5V	_	10	25	ns	
			1.8-5.5V	—	25	50	ns	
SP76*	TDOF	SDO data output fall time		—	10	25	ns	
SP77*	TssH2doZ	SS↑ to SDO output high-impedance		10	-	50	ns	
SP78* TscR	TscR	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns	
			1.8-5.5V	—	25	50	ns	
SP79*	TSCF	SCK output fall time (Master mode)	SCK output fall time (Master mode)		10	25	ns	
SP80*	TscH2doV,	SDO data output valid after SCK edge	3.0-5.5V	_		50	ns	
	TscL2DoV		1.8-5.5V	—	_	145	ns	
SP81*	TDOV2SCH, TDOV2SCL	SDO data output setup to SCK edge		Тсу	_	—	ns	
SP82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge		_	_	50	ns	
SP83*	TscH2ssH, TscL2ssH	SS ↑ after SCK edge		1.5Tcy + 40	—	—	ns	

TABLE 23-11: SPI MODE REQUIREMENTS

These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 24-13: PIC16F720/721 BASE IPD vs. VDD









© 2010-2015 Microchip Technology Inc.



FIGURE 24-34: PIC16F720/721 A/D INTERNAL RC OSCILLATOR PERIOD

FIGURE 24-35: TYPICAL FVR (X1 AND X2) VS. SUPPLY VOLTAGE (V) NORMALIZED AT 3.0V



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 Micro-chip 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