

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

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	21
Program Memory Size	32KB (11K x 24)
Program Memory Type	FLASH
EEPROM Size	
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24hj32gp202-h-so

Email: info@E-XFL.COM

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

4.2 Data Address Space

The CPU has a separate 16 bit wide data memory space. The data space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory maps is shown in Figure 4-3.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to the bytes within the data space. This arrangement gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when EA<15>=0) is used for implemented memory addresses, while the upper half (EA<15>=1) is reserved for the Program Space Visibility area (see Section 4.6.3 "Reading Data from Program Memory Using Program Space Visibility").

PIC24HJ32GP202/204 and PIC24HJ16GP304 devices implement up to 2 Kbytes of data memory. Should an EA point to a location outside of this area, an all-zero word or byte will be returned.

4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte addressable, 16 bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC[®] devices and improve data space memory usage efficiency, the PIC24HJ32GP202/204 and PIC24HJ16GP304 instruction set supports both word and byte operations. As a consequence of byte accessibility, all effective address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word that contains the byte, using the LSB of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register that matches the byte address. All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or when translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap is generated. If the error occurred on a read, the instruction underway is completed. If the instruction occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the Least Significant Byte. The Most Significant Byte is not modified.

A sign-extend instruction (SE) is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, user applications can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

4.2.3 SFR SPACE

The first 2 Kbytes of the Near Data Space, from 0x0000 to 0x07FF, is primarily occupied by Special Function Registers (SFRs). These are used by the PIC24HJ32GP202/204 and PIC24HJ16GP304 core and peripheral modules to control the operation of the device.

SFRs are distributed among the modules that they control, and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'. A complete listing of implemented SFRs, including their addresses, is shown in Table 4-1 through Table 4-22.

Note:	The actual set of peripheral features and interrupts varies by the device. Refer to										
	the corresponding device tables and										
	pinout diagrams for device-specific										
	information.										

4.2.4 NEAR DATA SPACE

The 8 Kbyte area between 0x0000 and 0x1FFF is referred to as the Near Data Space. Locations in this space are directly addressable via 13-bit absolute address field within all memory direct instructions. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a working register as an address pointer.

TABLE 4-16: PORTA REGISTER MAP FOR PIC24HJ32GP202

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISA	02C0	-	-	-	-	-	-	-	-	_	-	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	001F
PORTA	02C2	—	_	—	—	_	—	_	—	_	_	—	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	_	_	_	_	_	_	_	_	_	_	_	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	_	_	_	_	_	_	_	_	_	_	_	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-17: PORTA REGISTER MAP FOR PIC24HJ32GP204 AND PIC24HJ16GP304

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISA	02C0	—	—	—	—	—	TRISA10	TRISA9	TRISA8	TRISA7	-	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	001F
PORTA	02C2	_	_	_	_	_	RA10	RA9	RA8	RA7	_	_	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	_	_	_	_	_	LATA10	LATA9	LATA8	LATA7	_	_	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	_	_	_	_	_	ODCA10	ODCA9	ODCA8	ODCA7	_	_	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-18: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODCB15	ODCB14	ODCB13	ODCB12	ODCB11	ODCB10	ODCB9	ODCB8	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	0000

Legend: x = unknown value on Reset, - = unimplemented, read as '0'. Reset values are shown in hexadecimal for PinHigh devices.

TABLE 4-19: PORTC REGISTER MAP FOR PIC24HJ32GP204 AND PIC24HJ16GP304

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISC	02D0	—	—	—	—	—	—	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	03FF
PORTC	02D2	_	—	—	—	—	_	RC9	RC8	RC7	RC6	RC5	RC4	RC4	RC2	RC1	RC0	xxxx
LATC	02D4	_	_	_	_	_	_	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC4	LATC2	LATC1	LATC0	xxxx
ODCC	02D6	_	_	_	_	_	_	ODCC9	ODCC8	ODCC7	ODCC6	ODCC5	ODCC4	ODCC4	ODCC2	ODCC1	ODCC0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

4.6 Interfacing Program and Data Memory Spaces

The device architecture uses a 24-bit-wide program space and a 16 bit wide data space. The architecture is also a modified Harvard scheme, which means that the data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (Program Space Visibility)

Table instructions allow an application to read or write to small areas of the program memory. This capability makes the method ideal for accessing data tables that need to be updated periodically. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look ups from a large table of static data. The application can only access the least significant word of the program word.

4.6.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Page register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the Most Significant bit of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 8-bit Program Space Visibility register (PSVPAG) is used to define a 16K word page in the program space. When the Most Significant bit of the EA is '1', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike table operations, this limits remapping operations strictly to the user memory area.

Table 4-24 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, and D<15:0> refers to a data space word.

	Access		Program Space Address									
Access Type	Space	<23>	<22:16>	<15>	<14:1>	<0>						
Instruction Access	User	0		PC<22:1>		0						
(Code Execution)			0xx xxxx x	xxx xxx	x xxxx xxx0							
TBLRD/TBLWT	User	TB	LPAG<7:0>		Data EA<15:0>							
(Byte/Word Read/Write)		0	xxx xxxx	xxxx xx	xx xxxx xxxx							
	Configuration	TB	LPAG<7:0>		Data EA<15:0>							
		1	xxx xxxx	xxxx x	xxx xxxx xxxx							
Program Space Visibility	User	0	PSVPAG<7	/:0>	Data EA<14:	0> ⁽¹⁾						
(Block Remap/Read)		0	XXXX XXXX	۲.	xxx xxxx xxxx xxxx							

TABLE 4-24: PROGRAM SPACE ADDRESS CONSTRUCTION

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.

NOTES:

5.2 RTSP Operation

The Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user application to erase a page of memory, which consists of eight rows (512 instructions) at a time, and to program one row or one word at a time. The 8-row erase pages and single row write rows are edge-aligned from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

The program memory implements holding buffers that can contain 64 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the buffers sequentially. The instruction words loaded must always be from a group of 64 boundary.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register. A total of 64 TBLWTL and TBLWTH instructions are required to load the instructions.

All table write operations are single-word writes (two instruction cycles) because only the buffers are written. A programming cycle is required for programming each row.

5.3 **Programming Operations**

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. The processor stalls (waits) until the programming operation is finished.

The programming time depends on the FRC accuracy (see Table 22-18) and the value of the FRC Oscillator Tuning register (see Register 8-4). Use the following formula to calculate the minimum and maximum values for the Row Write Time, Page Erase Time, and Word Write Cycle Time parameters (see Table 22-12).

EQUATION 5-1: PROGRAMMING TIME



For example, if the device is operating at $+125^{\circ}$ C, the FRC accuracy will be $\pm 5\%$. If the TUN<5:0> bits (see Register 8-4) are set to `bl11111, the minimum row write time is equal to Equation 5-2.

EQUATION 5-2: MINIMUM ROW WRITE TIME

 $T_{RW} = \frac{11064 \ Cycles}{7.37 \ MHz \times (1 + 0.05) \times (1 - 0.00375)} = 1.435 ms$

The maximum row write time is equal to Equation 5-3.

EQUATION 5-3: MAXIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064 \ Cycles}{7.37 \ MHz \times (1 - 0.05) \times (1 - 0.00375)} = 1.586 ms$$

Setting the WR bit (NVMCON<15>) starts the operation, and the WR bit is automatically cleared when the operation is finished.

5.4 Flash Memory Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page, which can be accessed using this link, contains the latest updates and additional information.

Note: In the event you are not able to access the product page using the link above, enter this URL in your browser: http://www.microchip.com/wwwproducts/ Devices.aspx?dDocName=en530271

5.4.1 KEY RESOURCES

- Section 5. "Flash Programming" (DS70191)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All related dsPIC33F/PIC24H Family Reference Manuals Sections
- Development Tools

5.5 Control Registers

The two SFRs that are used to read and write the program Flash memory are:

NVMCON: Flash Memory Control Register

• NVMKEY: Nonvolatile Memory Key Register

The NVMCON register (Register 5-1) controls which blocks need to be erased, which memory type is to be programmed and the start of the programming cycle.

NVMKEY (Register 5-2) is a write-only register that is used for write protection. To start a programming or erase sequence, the user application must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 5.3** "**Programming Operations**" for further details.

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0				
—	—	—	—		—	—					
bit 15							bit 8				
U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0				
—	—	—	—	IPL3 ⁽²⁾	PSV	—					
bit 7							bit 0				
Legend:		C = Clear only	/ bit								
R = Readable b	oit	W = Writable	bit	-n = Value at	POR	'1' = Bit is set					
0' = Bit is cleare	ed	ʻx = Bit is unkr	nown	U = Unimplemented bit, read as '0'							

REGISTER 7-2: CORCON: CORE CONTROL REGISTER⁽¹⁾

bit 3 IPL3: CPU Interrupt Priority Level Status bit 3⁽²⁾ 1 = CPU interrupt priority level is greater than 7 0 = CPU interrupt priority level is 7 or less

Note 1: For complete register details, see Register 3-2.

2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8
U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'	
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	Iown
bit 15	NSTDIS: Inter	rrupt Nesting D	isable bit				
	1 = Interrupt n	nesting is disab	led				
	0 = Interrupt n	lesting is enable	led				
bit 14-7	Unimplement	ted: Read as '	D'				
bit 6	DIV0ERR: Ari	thmetic Error S	Status bit				
	1 = Math error 0 = Math error	r trap was caus r trap was not o	sed by a divide caused by a di	e by zero ivide by zero			
bit 5	Unimplement	ted: Read as '	D'	2			
bit 4	MATHERR: A	rithmetic Error	Status bit				
	1 = Math error	r trap has occu	irred				
	0 = Math error	r trap has not c	occurred				
bit 3	ADDRERR: A	ddress Error T	rap Status bit				
	1 = Address e	error trap has o	ccurred				
	0 = Address e	error trap has n	ot occurred				
bit 2	STKERR: Sta	ck Error Trap S	Status bit				
	1 = Stack erro	or trap has occu	urred				
		or trap has not o		.,			
DIT 1		cillator Failure	Trap Status b	IT			
	$\perp = Oscillator$ 0 = Oscillator	failure trap has	s occurred				
hit 0	Unimplement	ted: Read as '	n'				
	ennpionen						

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0						
_		—	—		—	—	—						
bit 15							bit 8						
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0						
—	_	—	—	_	—	U1EIE	—						
bit 7							bit 0						
Legend:													
R = Readable	bit	W = Writable	bit	U = Unimplemented bit, read as '0'									
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown						
bit 15-2	Unimplemen	ted: Read as '	0'										
bit 1	U1EIE: UART	1 Error Interru	pt Enable bit										
	1 = Interrupt r	equest enable	d										
	0 = Interrupt r	equest not ena	abled										

REGISTER 7-10: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

bit 0 Unimplemented: Read as '0'

TABLE 10-1:REMAPPABLE PERIPHERAL INPUTS⁽¹⁾

Input Name	Function Name	Register	Configuration Bits
External Interrupt 1	INT1	RPINR0	INT1R<4:0>
External Interrupt 2	INT2	RPINR1	INT2R<4:0>
Timer 2 External Clock	T2CK	RPINR3	T2CKR<4:0>
Timer 3 External Clock	T3CK	RPINR3	T3CKR<4:0>
Input Capture 1	IC1	RPINR7	IC1R<4:0>
Input Capture 2	IC2	RPINR7	IC2R<4:0>
Input Capture 7	IC7	RPINR10	IC7R<4:0>
Input Capture 8	IC8	RPINR10	IC8R<4:0>
Output Compare Fault A	OCFA	RPINR11	OCFAR<4:0>
UART 1 Receive	U1RX	RPINR18	U1RXR<4:0>
UART 1 Clear To Send	U1CTS	RPINR18	U1CTSR<4:0>
SPI 1 Data Input	SDI1	RPINR20	SDI1R<4:0>
SPI 1 Clock Input	SCK1IN	RPINR20	SCK1R<4:0>
SPI 1 Slave Select Input	SS1IN	RPINR21	SS1R<4:0>

FIGURE 10-3:

Note 1: Unless otherwise noted, all inputs use the Schmitt input buffers.

10.6.2.2 Output Mapping

In contrast to inputs, the outputs of the peripheral pin select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Like the RPINRx registers, each register contains sets of 5-bit fields, with each set associated with one RPn pin (see Register 10-10 through Register 10-22). The value of the bit field corresponds to one of the peripherals, and that peripheral's output is mapped to the pin (see Table 10-2 and Figure 10-3).

The list of peripherals for output mapping also includes a null value of '00000' because of the mapping technique. This permits any given pin to remain unconnected from the output of any of the pin selectable peripherals.

MULTIPLEXING OF REMAPPABLE OUTPUT FOR RPn



U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	—	—			IC8R<4:0>		
bit 15		·					bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	_			IC7R<4:0>		
bit 7							bit 0
Legend:							
R = Readab	le bit	W = Writable	bit	U = Unimple	mented bit, rea	d as '0'	
-n = Value a	t POR	'1' = Bit is set	t	'0' = Bit is cle	eared	x = Bit is unkr	nown
bit 15-13	Unimplemer	nted: Read as '	0'				
bit 12-8	IC8R<4:0>: /	Assign Input Ca	apture 8 (IC8)	to the correspo	onding RPn pir	ı	
	11111 = Inp i	ut tied to Vss					
	11001 = Inp	ut tied to RP25					
	•						
	•						
	•						
	00001 = Inpl	ut fied to RP1					
hit 7 5		at de Bood os '	0'				
bit 4 0		Aggian Input Co	\cup	to the correspond	onding DDn nir		
DIL 4-0	11111 - Inn			to the conespo	Shulling REIT pli	I	
	11001 = Inpl	ut tied to RP25					
	•						
	•						
	•						
	00001 = Inp	ut tied to RP1					
	00000 = Inp i	ut tied to RP0					

REGISTER 10-5: RPIR10: PERIPHERAL PIN SELECT INPUT REGISTER 10

REGISTER 10-11: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
—	—	—	RP3R<4:0>						
bit 15							bit 8		
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
—	—	—			RP2R<4:0>				
bit 7							bit 0		

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

bit 15-13 Unimplemented: Read as '0'

bit 12-8 RP3R<4:0>: Peripheral Output Function is Assigned to RP3 Output Pin (see Table 10-2 for peripheral function numbers)

bit 7-5 Unimplemented: Read as '0'

bit 4-0 RP2R<4:0>: Peripheral Output Function is Assigned to RP2 Output Pin (see Table 10-2 for peripheral function numbers)

REGISTER 10-12: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
	—	—			RP5R<4:0>				
bit 15							bit 8		
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
_	_	—	RP4R<4:0>						
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimplemented bit, read as '0'					
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown		
bit 15-13	Unimplemen	ted: Read as '	0'						
bit 12-8	bit 12-8 RP5R<4:0>: Peripheral Output Function is Assigned to RP5 Output Pin (see Table 10-2 for peripheral function numbers)								
			-1						

bit 7-5 Unimplemented: Read as '0'

bit 4-0 RP4R<4:0>: Peripheral Output Function is Assigned to RP4 Output Pin (see Table 10-2 for peripheral function numbers)

14.1 Output Compare Modes

Configure the Output Compare modes by setting the appropriate Output Compare Mode bits (OCM<2:0>) in the Output Compare Control register (OCxCON<2:0>). Table 14-1 lists the different bit settings for the Output Compare modes. Figure 14-2 illustrates the output compare operation for various modes. The user

TABLE 14-1: OUTPUT COMPARE MODES

application must disable the associated timer when writing to the output compare control registers to avoid malfunctions.

Note: See Section 13. "Output Compare" (DS70209) in the "dsPIC33F/PIC24H Family Reference Manual" for OCxR and OCxRS register restrictions.

OCM<2:0>	Mode	OCx Pin Initial State	OCx Interrupt Generation
000	Module Disabled	Controlled by GPIO register	—
001	Active-Low One-Shot	0	OCx Rising edge
010	Active-High One-Shot	1	OCx Falling edge
011	Toggle Mode	Current output is maintained	OCx Rising and Falling edge
100	Delayed One-Shot	0	OCx Falling edge
101	Continuous Pulse mode	0	OCx Falling edge
110	PWM mode without fault protection	0, if OCxR is zero 1, if OCxR is non-zero	No interrupt
111	PWM mode with fault protection	0, if OCxR is zero 1, if OCxR is non-zero	OCFA Falling edge for OC1 to OC4

FIGURE 14-2: OUTPUT COMPARE OPERATION



15.0 SERIAL PERIPHERAL INTERFACE (SPI)

- Note 1: This data sheet summarizes the features of the PIC24HJ32GP202/204 and PIC24HJ16GP304 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to "Section 18. Serial Peripheral Interface (SPI)" (DS70206) of the "dsPIC33F/PIC24H Family Reference Manual", which is available from the Microchip website (www.microchip.com).
 - 2: Some registers and associated bits described in this section may not be available on all devices. Refer to Section 4.0 "Memory Organization" in this data sheet for device-specific register and bit information.

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices can be serial EEPROMs, shift registers, display drivers, Analog-to-Digital Converters (ADCs), and so on. The SPI module is compatible with Motorola[®] SPI and SIOP.

Each SPI module consists of a 16-bit shift register, SPIxSR (where x = 1 or 2), used for shifting data in and out, and a buffer register, SPIxBUF. A control register, SPIxCON, configures the module. Additionally, a status register, SPIxSTAT, indicates status conditions.

The serial interface consists of these four pins:

- · SDIx (serial data input)
- SDOx (serial data output)
- SCKx (shift clock input or output)
- SSx (active-low slave select)

In Master mode operation, SCK is a clock output. In Slave mode, it is a clock input.



FIGURE 15-1: SPI MODULE BLOCK DIAGRAM

15.1 SPI Helpful Tips

- 1. In Frame mode, if there is a possibility that the master may not be initialized before the slave:
 - a) If FRMPOL (SPIxCON2<13>) = 1, use a pull-down resistor on SSx.
 - b) If FRMPOL = 0, use a pull-up resistor on $\frac{1}{SSx}$.

Note:	This	insures	that	t the	first	fra	ame
	transn	nission	after	initializ	ation	is	not
	shifted	d or corru	pted.				

- 2. In non-framed 3-wire mode, (i.e., not using SSx from a master):
 - a) If CKP (SPIxCON1<6>) = 1, always place a pull-up resistor on SSx.
 - b) If CKP = 0, always place a pull-down resistor on SSx.
- **Note:** This will insure that during power-up and initialization the master/slave will not lose sync due to an errant SCK transition that would cause the slave to accumulate data shift errors for both transmit and receive appearing as corrupted data.
- FRMEN (SPIxCON2<15>) = 1 and SSEN (SPIxCON1<7>) = 1 are exclusive and invalid. In Frame mode, SCKx is continuous and the Frame sync pulse is active on the SSx pin, which indicates the start of a data frame.
 - Note: Not all third-party devices support Frame mode timing. Refer to the SPI electrical characteristics for details.
- In Master mode only, set the SMP bit (SPIxCON1<9>) to a '1' for the fastest SPI data rate possible. The SMP bit can only be set at the same time or after the MSTEN bit (SPIxCON1<5>) is set.
- 5. To avoid invalid slave read data to the master, the user's master software must guarantee enough time for slave software to fill its write buffer before the user application initiates a master write/read cycle. It is always advisable to preload the SPIxBUF transmit register in advance of the next master transaction cycle. SPIxBUF is transferred to the SPI shift register and is empty once the data transmission begins.

15.2 SPI Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page, which can be accessed using this link, contains the latest updates and additional information.

Note: In the event you are not able to access the product page using the link above, enter this URL in your browser: http://www.microchip.com/wwwproducts/ Devices.aspx?dDocName=en530271

15.2.1 KEY RESOURCES

- Section 18. "Serial Peripheral Interface (SPI)" (DS70206)
- Code Samples
- Application Notes
- · Software Libraries
- Webinars
- All related dsPIC33F/PIC24H Family Reference Manuals Sections
- Development Tools

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5	 ACKDT: Acknowledge Data bit (when operating as I²C master, applicable during master receive) Value that will be transmitted when the software initiates an Acknowledge sequence. 1 = Send NACK during Acknowledge 0 = Send ACK during Acknowledge 						
bit 4	ACKEN: Acknowledge Sequence Enable bit (when operating as I ² C master, applicable during master receive)						
	 1 = Initiate Acknowledge sequence on SDAx and SCLx pins and transmit ACKDT data bit. Hardware clear at end of master Acknowledge sequence 0 = Acknowledge sequence not in progress 						
bit 3	RCEN: Receive Enable bit (when operating as I^2C master)						
	1 = Enables Receive mode for I^2C . Hardware clear at end of eighth bit of master receive data byte 0 = Receive sequence not in progress						
bit 2	PEN: Stop Condition Enable bit (when operating as I ² C master)						
	 1 = Initiate Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence 0 = Stop condition not in progress 						
bit 1	RSEN: Repeated Start Condition Enable bit (when operating as I ² C master)						
	 1 = Initiate Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of master Repeated Start sequence 						
	0 = Repeated Start condition not in progress						
bit 0	SEN: Start Condition Enable bit (when operating as I ² C master)						
	 1 = Initiate Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence 0 = Start condition not in progress 						

17.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

- Note 1: This data sheet summarizes the features of the PIC24HJ32GP202/204 and PIC24HJ16GP304 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to "Section 17. UART" (DS70188) of the "dsPIC33F/PIC24H Family Reference Manual", which is available from the Microchip website (www.microchip.com).
 - 2: Some registers and associated bits described in this section may not be available on all devices. Refer to Section 4.0 "Memory Organization" in this data sheet for device-specific register and bit information.

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the PIC24HJ32GP202/204 and PIC24HJ16GP304 device family. The UART is a full-duplex asynchronous system that can communicate with peripheral devices, such as personal computers, LIN, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the UxCTS and UxRTS pins and also includes an IrDA[®] encoder and decoder.

The primary features of the UART module are:

- Full-Duplex 8-bit or 9-bit Data Transmission through the UxTX and UxRX pins
- Even, odd or no parity options (for 8-bit data)
- One or two stop bits
- Hardware Flow Control Option with UxCTS and UxRTS pins
- Fully Integrated Baud Rate Generator with 16-bit prescaler
- Baud rates ranging from 10 Mbps to 38 bps at 40 MIPS
- 4-deep first-in-first-out (FIFO) Transmit Data Buffer
- 4-deep FIFO Receive Data Buffer
- · Parity, framing and buffer overrun error detection
- Support for 9-bit mode with Address Detect (9th bit = 1)
- · Transmit and Receive interrupts
- · A separate interrupt for all UART error conditions
- Loopback mode for diagnostic support
- · Support for Sync and Break characters
- · Support for automatic baud rate detection
- IrDA[®] encoder and decoder logic
- + 16x baud clock output for $\mbox{IrDA}^{\mbox{$\mathbb{R}$}}$ support

A simplified block diagram of the UART module is shown in Figure 17-1. The UART module consists of the following key hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

FIGURE 17-1: UART SIMPLIFIED BLOCK DIAGRAM



DC CHARACTER	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$							
Parameter No. ⁽³⁾ Typical ⁽²⁾ Max			Doze Ratio	Units		Conditions		
Doze Current (IDC	2E) ⁽¹⁾							
DC73a	41	51	1:2	mA				
DC73f	20	28	1:64	mA	-40°C	3.3V	40 MIPS	
DC73g	19	24	1:128	mA				
DC70a	40	46	1:2	mA		3.3V	40 MIPS	
DC70f	18	20	1:64	mA	+25°C			
DC70g	18	20	1:128	mA				
DC71a	40	46	1:2	mA				
DC71f	18	25	1:64	mA	+85°C	3.3V	40 MIPS	
DC71g	18	20	1:128	mA				
DC72a	39	55	1:2	mA				
DC72f	18	30	1:64	mA	+125°C	3.3V	40 MIPS	
DC72g	18	25	1:128	mA				

Note 1: IDOZE is primarily a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDOZE measurements are as follows:

 Oscillator is configured in EC mode and external clock active, OSC1 is driven with external square wave from rail-to-rail with overshoot/undershoot < 250 mV

- · CLKO is configured as an I/O input pin in the Configuration word
- All I/O pins are configured as inputs and pulled to Vss
- MCLR = VDD, WDT and FSCM are disabled
- · CPU, SRAM, program memory and data memory are operational
- No peripheral modules are operating; however, every peripheral is being clocked (defined PMDx bits are set to zero and unimplemented PMDx bits are set to one)
- CPU executing while(1) statement
- JTAG is disabled
- 2: Data in the "Typ" column is at 3.3V, +25°C unless otherwise stated.

FIGURE 22-8: OC/PWM MODULE TIMING CHARACTERISTICS

TABLE 22-27: SIMPLE OC/PWM MODE TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{array}{l} \mbox{Standard Operating Conditions: 3.0V to 3.6V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ & -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$					
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур	Max	Units	Conditions	
OC15	Tfd	Fault Input to PWM I/O Change	—	_	Tcy + 20	ns	_	
OC20	TFLT	Fault Input Pulse-Width	Tcy + 20	_	—	ns	—	

Note 1: These parameters are characterized but not tested in manufacturing.

TABLE 23-15: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +150^{\circ}C$ for High Temperature							
Param No. Symbol		Characteristic	Min	Тур	Max	Units	Conditions		
Reference Inputs									
HAD08	IREF	Current Drain	_	250 —	600 50	μA μA	ADC operating, See Note 1 ADC off, See Note 1		

Note 1: These parameters are not characterized or tested in manufacturing.

2: These parameters are characterized, but are not tested in manufacturing.

TABLE 23-16: ADC MODULE SPECIFICATIONS (12-BIT MODE)⁽³⁾

AC		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)								
CHARACTERISTICS		Operating temperature $-40^{\circ}C \le TA \le +150^{\circ}C$ for High Temperature								
Param No.	Symbol	Characteristic Min Typ		Тур	Max	Units	Conditions			
ADC Accuracy (12-bit Mode) – Measurements with External VREF+/VREF- ⁽¹⁾										
HAD20a	Nr	Resolution ⁽³⁾	1.	2 data bi	ts	bits	—			
HAD21a	INL	Integral Nonlinearity	-2	_	+2	LSb	Vinl = AVss = Vrefl = 0V, AVdd = Vrefh = 3.6V			
HAD22a	DNL	Differential Nonlinearity	> -1	_	< 1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V			
HAD23a	Gerr	Gain Error	-2	—	10	LSb	Vinl = AVss = Vrefl = 0V, AVdd = Vrefh = 3.6V			
HAD24a	EOFF	Offset Error	-3	—	5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V			
	AD	C Accuracy (12-bit Mode	e) – Meas	uremen	ts with In	ternal V	/REF+/VREF- ⁽¹⁾			
HAD20a	Nr	Resolution ⁽³⁾	1	2 data bi	ts	bits	—			
HAD21a	INL	Integral Nonlinearity	-2	—	+2	LSb	VINL = AVSS = 0V, AVDD = 3.6V			
HAD22a	DNL	Differential Nonlinearity	> -1	—	< 1	LSb	VINL = AVSS = 0V, AVDD = 3.6V			
HAD23a	Gerr	Gain Error	2		20	LSb	VINL = AVSS = 0V, AVDD = 3.6V			
HAD24a EOFF C		Offset Error	2	_	10	LSb	VINL = AVSS = 0V, AVDD = 3.6V			
		Dynamic I	Performa	nce (12	-bit Mode	e) ⁽²⁾				
HAD33a	Fnyq	Input Signal Bandwidth	_	—	200	kHz	—			

Note 1: These parameters are characterized, but are tested at 20 ksps only.

2: These parameters are characterized by similarity, but are not tested in manufacturing.

3: Injection currents > | 0 | can affect the ADC results by approximately 4-6 counts.

Note the following details of the code protection feature on Microchip devices:

- · Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949=

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007-2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 978-1-62076-440-4

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and mulfacture of development systems is ISO 9001:2000 certified.