



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 - Microcontrollers</u>"

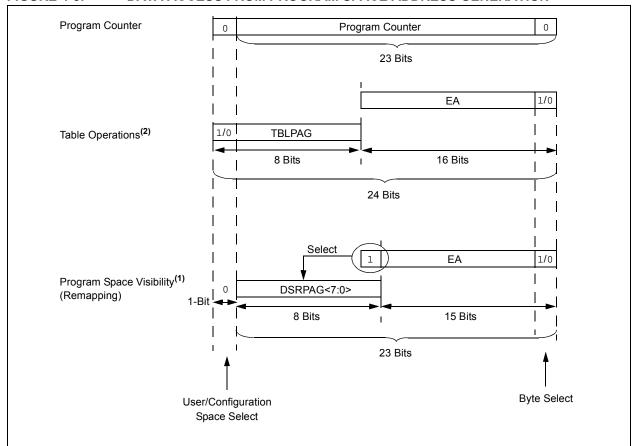
Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, GFX, LVD, POR, PWM, WDT
Number of I/O	84
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128da210-i-pt

TABLE 4-36: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Tyres	Access	Program Space Address						
Access Type	Space	<23>	<22:16>	<15>	<14:1>	<0>		
Instruction Access	User	0	0 PC<22:1>			0		
(Code Execution)		0xx xxxx xxxx xxxx xxx0						
TBLRD/TBLWT (Byte/Word Read/Write)	User	ТВ	LPAG<7:0>	Data EA<15:0>				
		0xxx xxxx		xxxx xxxx xxxx xxxx				
	Configuration	TBLPAG<7:0>		Data EA<15:0>				
		1:	xxx xxxx	xxxx xxxx xxxx xxxx				
Program Space Visibility	User	0		DSRPAG<7:0>(2)		:0> ⁽¹⁾		
(Block Remap/Read)		0	xxxx xxxx		xxx xxxx xxxx xxxx			

- **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 DSRPAG<0>.
 - 2: DSRPAG<9> is always '1' in this case. DSRPAG<8> decides whether the lower word or higher word of program memory is read. When DSRPAG<8> is '0', the lower word is read and when it is '1', the higher word is read.

FIGURE 4-8: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



- Note 1: DSRPAG<8> acts as word select. DSRPAG<9> should always be '1' to map program memory to data memory.
 - 2: The instructions, TBLRDH/TBLWTH/TBLRDL/TBLWTL, decide if the higher or lower word of program memory is accessed. TBLRDH/TBLWTH instructions access the higher word and TBLRDL/TBLWTL instructions access the lower word. Table read operations are permitted in the configuration memory space.

EXAMPLE 5-2: ERASING A PROGRAM MEMORY BLOCK ('C' LANGUAGE CODE)

```
// C example using MPLAB C30
   unsigned long progAddr = 0xXXXXXX;
                                             // Address of row to write
   unsigned int offset;
//Set up pointer to the first memory location to be written
                               // Initialize PM Page Boundary SFR
   TBLPAG = progAddr>>16;
   offset = progAddr & 0xFFFF;
                                             // Initialize lower word of address
   __builtin_tblwtl(offset, 0x0000); // Set base address of erase block
                                             // with dummy latch write
   NVMCON = 0x4042;
                                             // Initialize NVMCON
   asm("DISI #5");
                                             // Block all interrupts with priority <7
                                             // for next 5 instructions
   builtin write NVM();
                                             // check function to perform unlock
                                             // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV
             #0x4001, W0
              W0, NVMCON
       MOV
                                               ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
             #0x0000, W0
       MOV
             W0, TBLPAG
                                              ; Initialize PM Page Boundary SFR
       MOV
             #0x6000, W0
                                               ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV
             #LOW_WORD_0, W2
       MOV
             #HIGH_BYTE_0, W3
                                               ;
       TBLWTL W2, [W0]
                                               ; Write PM low word into program latch
      TBLWTH W3, [W0++]
                                               ; Write PM high byte into program latch
; 1st_program_word
       MOV
            #LOW_WORD_1, W2
                                               ;
       MOV
             #HIGH_BYTE_1, W3
                                               ;
       TBLWTL W2, [W0]
                                               ; Write PM low word into program latch
       TBLWTH W3, [W0++]
                                               ; Write PM high byte into program latch
  2nd_program_word
       MOV #LOW_WORD_2, W2
MOV #HIGH_BYTE_2, W3
                                               ;
       TBLWTL W2, [W0]
                                               ; Write PM low word into program latch
       TBLWTH W3, [W0++]
                                               ; Write PM high byte into program latch
; 63rd_program_word
            #LOW_WORD_63, W2
       MOV
       MOV
              #HIGH_BYTE_63, W3
       TBLWTL W2, [W0]
                                               ; Write PM low word into program latch
       TBLWTH W3, [W0]
                                                ; Write PM high byte into program latch
```

EXAMPLE 5-4: INITIATING A PROGRAMMING SEQUENCE

```
DIST
       #5
                                ; Block all interrupts with priority <7
                                ; for next 5 instructions
MOV.B #0x55, W0
       WO, NVMKEY
                                ; Write the 0x55 key
MOV
MOV.B #0xAA, W1
       W1, NVMKEY
                               ; Write the OxAA key
MOM
                               ; Start the programming sequence
BSET
      NVMCON, #WR
                                ; Required delays
NOP
BTSC NVMCON, #15
                                ; and wait for it to be
BRA
                                ; completed
       $-2
```

10.4.3.2 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. Each register contains two 6-bit fields, with each field being associated with one RPn pin (see Register 10-29 through Register 10-44). The value of the bit field

corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-4).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '000000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-4: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Output Function Number ⁽¹⁾	Function	Output Name
0	NULL ⁽²⁾	Null
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	U1RTS ⁽³⁾	UART1 Request To Send
5	U2TX	UART2 Transmit
6	U2RTS ⁽³⁾	UART2 Request To Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
18	OC1	Output Compare 1
19	OC2	Output Compare 2
20	OC3	Output Compare 3
21	OC4	Output Compare 4
22	OC5	Output Compare 5
23	OC6	Output Compare 6
24	OC7	Output Compare 7
25	OC8	Output Compare 8
28	U3TX	UART3 Transmit
29	U3RTS ⁽³⁾	UART3 Request To Send
30	U4TX	UART4 Transmit
31	U4RTS ⁽³⁾	UART4 Request To Send
32	SDO3	SPI3 Data Output
33	SCK3OUT	SPI3 Clock Output
34	SS3OUT	SPI3 Slave Select Output
35	OC9	Output Compare 9
36	C3OUT	Comparator 3 Output
37-63	(unused)	NC

Note 1: Setting the RPORx register with the listed value assigns that output function to the associated RPn pin.

^{2:} The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

^{3:} IrDA® BCLK functionality uses this output.

REGISTER 10-10: RPINR2: PERIPHERAL PIN SELECT INPUT REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
_	_	_	_	_	_	_	_	
bit 15								

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5-0 INT4R<5:0>: Assign External Interrupt 4 (INT4) to Corresponding RPn or RPIn Pin bits

REGISTER 10-11: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 T3CKR<5:0>: Assign Timer3 External Clock (T3CK) to Corresponding RPn or RPIn Pin bits

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

bit 5-0 T2CKR<5:0>: Assign Timer2 External Clock (T2CK) to Corresponding RPn or RPln Pin bits

REGISTER 10-24: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-14 Unimplemented: Read as '0'

bit 13-8 SCK2R<5:0>: Assign SPI2 Clock Input (SCK2IN) to Corresponding RPn or RPIn Pin bits

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

bit 5-0 SDI2R<5:0>: Assign SPI2 Data Input (SDI2) to Corresponding RPn or RPIn Pin bits

REGISTER 10-25: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5-0 SS2R<5:0>: Assign SPI2 Slave Select Input (SS2IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-43: RPOR14: PERIPHERAL PIN SELECT OUTPUT REGISTER 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 RP29R<5:0>: RP29 Output Pin Mapping bits

Peripheral output number n is assigned to pin, RP29 (see Table 10-4 for peripheral function numbers).

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

bit 5-0 RP28R<5:0>: RP28 Output Pin Mapping bits

Peripheral output number n is assigned to pin, RP28 (see Table 10-4 for peripheral function numbers).

REGISTER 10-44: RPOR15: PERIPHERAL PIN SELECT OUTPUT REGISTER 15⁽¹⁾

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	RP31R5	RP31R4	RP31R3	RP31R2	RP31R1	RP31R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-14 Unimplemented: Read as '0'

bit 13-8 RP31R<5:0>: RP31 Output Pin Mapping bits⁽¹⁾

Peripheral output number n is assigned to pin, RP31 (see Table 10-4 for peripheral function numbers).

bit 7-6 Unimplemented: Read as '0'

bit 5-0 RP30R<5:0>: RP30 Output Pin Mapping bits⁽¹⁾

Peripheral output number n is assigned to pin, RP30 (see Table 10-4 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as '0'.

REGISTER 13-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
_	_	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	_	_
bit 15			•				bit 8

U-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
_	ICI1	ICI0	ICOV	ICBNE	ICM2 ⁽¹⁾	ICM1 ⁽¹⁾	ICM0 ⁽¹⁾
bit 7							bit 0

Legend: HSC = Hardware Settable/Clearable bit

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 15-14 Unimplemented: Read as '0'

bit 13 ICSIDL: Input Capture x Module Stop in Idle Control bit

1 = Input capture module halts in CPU Idle mode

0 = Input capture module continues to operate in CPU Idle mode

bit 12-10 ICTSEL<2:0>: Input Capture Timer Select bits

111 = System clock (Fosc/2)

110 = Reserved

101 = Reserved

100 = Timer1

011 = Timer5

010 = Timer4

001 = Timer2

000 = Timer3

bit 9-7 **Unimplemented:** Read as '0'

bit 6-5 ICI<1:0>: Select Number of Captures Per Interrupt bits

11 = Interrupt on every fourth capture event

10 = Interrupt on every third capture event

01 = Interrupt on every second capture event

00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture x Overflow Status Flag bit (read-only)

1 = Input capture overflow occurred

0 = No input capture overflow occurred

bit 3 **ICBNE**: Input Capture x Buffer Empty Status bit (read-only)

1 = Input capture buffer is not empty, at least one more capture value can be read

0 = Input capture buffer is empty

bit 2-0 ICM<2:0>: Input Capture Mode Select bits⁽¹⁾

111 = Interrupt mode: input capture functions as an interrupt pin only when the device is in Sleep or Idle mode (rising edge detect only, all other control bits are not applicable)

110 = Unused (module disabled)

101 = Prescaler Capture mode: capture on every 16th rising edge

100 = Prescaler Capture mode: capture on every 4th rising edge

011 = Simple Capture mode: capture on every rising edge

010 = Simple Capture mode: capture on every falling edge

001 = Edge Detect Capture mode: capture on every edge (rising and falling), ICI<1:0> bits do not control interrupt generation for this mode

000 = Input capture module turned off

Note 1: The ICx input must also be configured to an available RPn/RPIn pin. For more information, see Section 10.4 "Peripheral Pin Select (PPS)".

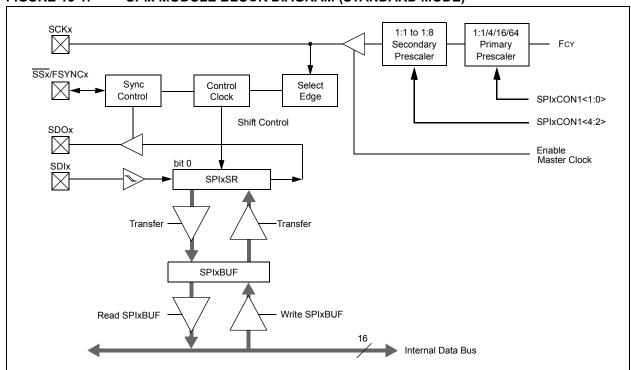
To set up the SPI module for the Standard Master mode of operation:

- 1. If using interrupts:
 - Clear the SPIxIF bit in the respective IFS register.
 - b) Set the SPIxIE bit in the respective IEC register.
 - Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
- Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 1.
- 3. Clear the SPIROV bit (SPIxSTAT<6>).
- Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
- Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Standard Slave mode of operation:

- 1. Clear the SPIxBUF register.
- 2. If using interrupts:
 - Clear the SPIxIF bit in the respective IFS register.
 - b) Set the SPIxIE bit in the respective IEC register.
 - Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
- 3. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 0.
- 4. Clear the SMP bit.
- If the CKE bit (SPIxCON1<8>) is set, then the SSEN bit (SPIxCON1<7>) must be set to enable the SSx pin.
- 6. Clear the SPIROV bit (SPIxSTAT<6>).
- 7. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

FIGURE 15-1: SPIX MODULE BLOCK DIAGRAM (STANDARD MODE)



NOTES:

18.2 USB Buffer Descriptors and the BDT

Endpoint buffer control is handled through a structure called the Buffer Descriptor Table (BDT). This provides a flexible method for users to construct and control endpoint buffers of various lengths and configurations.

The BDT can be located in any available, 512-byte aligned block of data RAM. The BDT Pointer (U1BDTP1) contains the upper address byte of the BDT and sets the location of the BDT in RAM. The user must set this pointer to indicate the table's location.

The BDT is composed of Buffer Descriptors (BDs) which are used to define and control the actual buffers in the USB RAM space. Each BD consists of two, 16-bit "soft" (non-fixed-address) registers, BDnSTAT and BDnADR, where n represents one of the 64 possible BDs (range of 0 to 63). BDnSTAT is the status register for BDn, while BDnADR specifies the starting address for the buffer associated with BDn.

Note: Since BDnADR is a 16-bit register, only the first 64 Kbytes of RAM can be accessed by the USB module.

Depending on the endpoint buffering configuration used, there are up to 64 sets of Buffer Descriptors, for a total of 256 bytes. At a minimum, the BDT must be at least 8 bytes long. This is because the USB Specification mandates that every device must have Endpoint 0 with both input and output for initial setup.

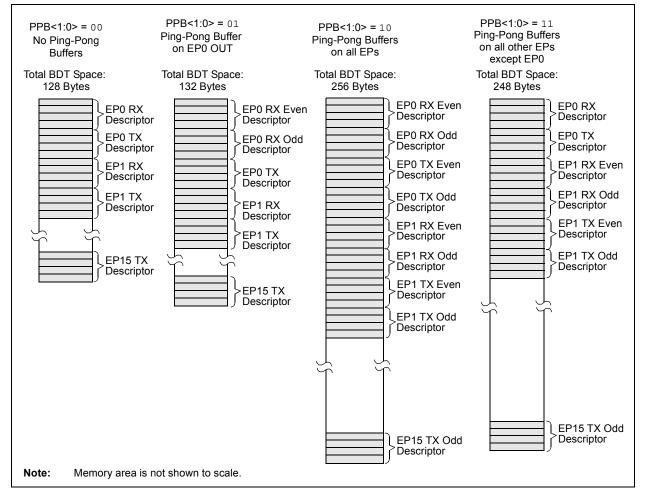
Endpoint mapping in the BDT is dependent on three variables:

- Endpoint number (0 to 15)
- Endpoint direction (RX or TX)
- Ping-pong settings (U1CNFG1<1:0>)

Figure 18-8 illustrates how these variables are used to map endpoints in the BDT.

In Host mode, only Endpoint 0 Buffer Descriptors are used. All transfers utilize the Endpoint 0 Buffer Descriptor and Endpoint Control register (U1EP0). For received packets, the attached device's source endpoint is indicated by the value of ENDPT<3:0> in the USB status register (U1STAT<7:4>). For transmitted packet, the attached device's destination endpoint is indicated by the value written to the Token register (U1TOK).

FIGURE 18-8: BDT MAPPING FOR ENDPOINT BUFFERING MODES



18.4.2 RECEIVING AN IN TOKEN IN DEVICE MODE

- 1. Attach to a USB host and enumerate as described in "Chapter 9 of the USB 2.0 Specification".
- 2. Create a data buffer and populate it with the data to send to the host.
- 3. In the appropriate (even or odd) TX BD for the desired endpoint:
 - Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
 - b) Set up the address register (BDnADR) with the starting address of the data buffer.
 - c) Set the UOWN bit of the status register to '1'.
- When the USB module receives an IN token, it automatically transmits the data in the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Transfer Complete Interrupt Flag, TRNIF (U1IR<3>).

18.4.3 RECEIVING AN OUT TOKEN IN DEVICE MODE

- Attach to a USB host and enumerate as described in "Chapter 9 of the USB 2.0 Specification".
- Create a data buffer with the amount of data you are expecting from the host.
- In the appropriate (even or odd) TX BD for the desired endpoint:
 - a) Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
 - b) Set up the address register (BDnADR) with the starting address of the data buffer.
 - c) Set the UOWN bit of the status register to '1'.
- 4. When the USB module receives an OUT token, it automatically receives the data sent by the host to the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Transfer Complete Interrupt Flag, TRNIF (U1IR<3>).

18.5 Host Mode Operation

The following sections describe how to perform common Host mode tasks. In Host mode, USB transfers are invoked explicitly by the host software. The host software is responsible for the Acknowledge portion of the transfer. Also, all transfers are performed using the Endpoint 0 Control register (U1EP0) and Buffer Descriptors.

18.5.1 ENABLE HOST MODE AND DISCOVER A CONNECTED DEVICE

- Enable Host mode by setting the HOSTEN bit (U1CON<3>). This causes the Host mode control bits in other USB OTG registers to become available.
- Enable the D+ and D- pull-down resistors by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5:4>). Disable the D+ and D-pull-up resistors by clearing the DPPULUP and DMPULUP bits (U1OTGCON<7:6>).
- At this point, SOF generation begins with the SOF counter loaded with 12,000. Eliminate noise on the USB by clearing the SOFEN bit (U1CON<0>) to disable Start-Of-Frame packet generation.
- 4. Enable the device attached interrupt by setting the ATTACHIE bit (U1IE<6>).
- 5. Wait for the device attached interrupt (U1IR<6> = 1). This is signaled by the USB device changing the state of D+ or D- from '0' to '1' (SE0 to J state). After it occurs, wait 100 ms for the device power to stabilize.
- Check the state of the JSTATE and SE0 bits in U1CON. If the JSTATE bit (U1CON<7>) is '0', the connecting device is low speed. If the connecting device is low speed, set the low LSPDEN and LSPD bits (U1ADDR<7> and U1EP0<7>) to enable low-speed operation.
- 7. Reset the USB device by setting the USBRST bit (U1CON<4>) for at least 50 ms, sending Reset signaling on the bus. After 50 ms, terminate the Reset by clearing USBRST.
- 8. In order to keep the connected device from going into suspend, enable the SOF packet generation by setting the SOFEN bit.
- 9. Wait 10 ms for the device to recover from Reset.
- 10. Perform enumeration as described by "Chapter 9 of the USB 2.0 Specification".

REGISTER 18-4: U10TGCON: USB ON-THE-GO CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPPULUP	DMPULUP	DPPULDWN ⁽¹⁾	DMPULDWN ⁽¹⁾	VBUSON ⁽¹⁾	OTGEN ⁽¹⁾	VBUSCHG ⁽¹⁾	VBUSDIS ⁽¹⁾
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-8 Unimplemented: Read as '0'

bit 7 DPPULUP: D+ Pull-up Enable bit

1 = D+ data line pull-up resistor is enabled

0 = D+ data line pull-up resistor is disabled

bit 6 DMPULUP: D- Pull-up Enable bit

1 = D- data line pull-up resistor is enabled0 = D- data line pull-up resistor is disabled

bit 5 **DPPULDWN:** D+ Pull-Down Enable bit⁽¹⁾

1 = D+ data line pull-down resistor is enabled0 = D+ data line pull-down resistor is disabled

bit 4 **DMPULDWN:** D- Pull-Down Enable bit⁽¹⁾

1 = D- data line pull-down resistor is enabled 0 = D- data line pull-down resistor is disabled

bit 3 VBUSON: VBUS Power-on bit⁽¹⁾

1 = VBUS line is powered

0 = VBUS line is not powered

bit 2 **OTGEN:** OTG Features Enable bit⁽¹⁾

1 = USB OTG is enabled; all D+/D- pull-up and pull-down bits are enabled

0 = USB OTG is disabled; D+/D- pull-up and pull-down bits are controlled in hardware by the settings of the HOSTEN and USBEN (U1CON<3,0>) bits

bit 1 VBUSCHG: VBUS Charge Select bit (1)

1 = VBUS line is set to charge to 3.3V

0 = VBUS line is set to charge to 5V

bit 0 **VBUSDIS:** VBUS Discharge Enable bit⁽¹⁾

1 = VBUS line is discharged through a resistor

0 = VBUS line is not discharged

Note 1: These bits are only used in Host mode; do not use in Device mode.

20.1.5 ALRMVAL REGISTER MAPPINGS

REGISTER 20-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	_	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-13 Unimplemented: Read as '0'

bit 12 MTHTEN0: Binary Coded Decimal Value of Month's Tens Digit bit

Contains a value of 0 or 1.

bit 11-8 MTHONE<3:0>: Binary Coded Decimal Value of Month's Ones Digit bits

Contains a value from 0 to 9.

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

bit 5-4 DAYTEN<1:0>: Binary Coded Decimal Value of Day's Tens Digit bits

Contains a value from 0 to 3.

bit 3-0 DAYONE<3:0>: Binary Coded Decimal Value of Day's Ones Digit bits

Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

FIGURE 23-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL

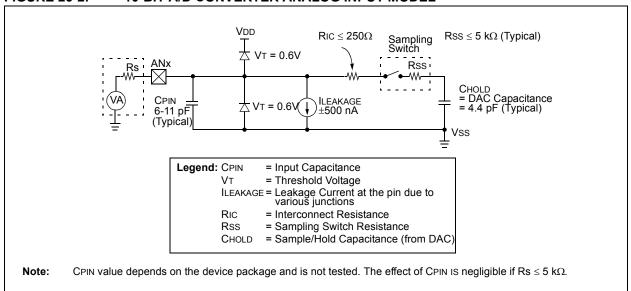
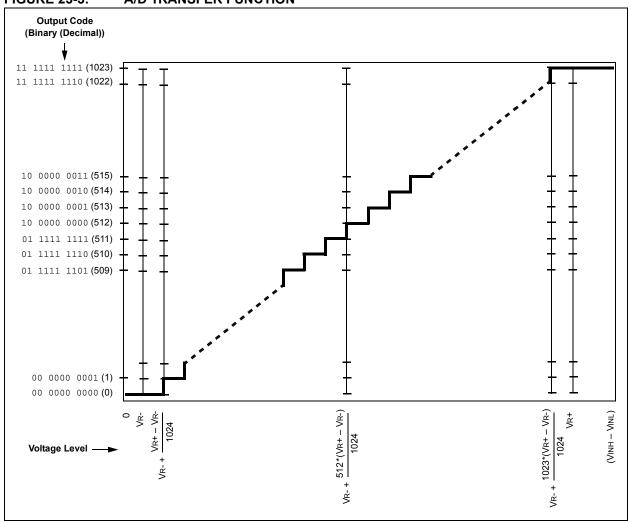


FIGURE 23-3: A/D TRANSFER FUNCTION



REGISTER 24-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3) (CONTINUED)

bit 4 CREF: Comparator Reference Select bits (non-inverting input)

1 = Non-inverting input connects to the internal CVREF voltage

0 = Non-inverting input connects to the CxINA pin

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

bit 1-0 **CCH<1:0>:** Comparator Channel Select bits

11 = Inverting input of the comparator connects to the internal selectable reference voltage specified by the CVREFM<1:0> bits in the CVRCON register

10 = Inverting input of the comparator connects to the CxIND pin

01 = Inverting input of the comparator connects to the CxINC pin

00 = Inverting input of the comparator connects to the CxINB pin

REGISTER 24-2: CMSTAT: COMPARATOR MODULE STATUS REGISTER

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	_	_	_	_	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
_	_	_	_	_	C3OUT	C2OUT	C1OUT
bit 7							bit 0

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

bit 15 CMIDL: Comparator Stop in Idle Mode bit

1 = Discontinue operation of all comparators when device enters Idle mode

0 = Continue operation of all enabled comparators in Idle mode

bit 14-11 Unimplemented: Read as '0'

bit 10 **C3EVT:** Comparator 3 Event Status bit (read-only)

Shows the current event status of Comparator 3 (CM3CON<9>).

bit 9 **C2EVT:** Comparator 2 Event Status bit (read-only)

Shows the current event status of Comparator 2 (CM2CON<9>).

bit 8 C1EVT: Comparator 1 Event Status bit (read-only)

Shows the current event status of Comparator 1 (CM1CON<9>).

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **C3OUT:** Comparator 3 Output Status bit (read-only)

Shows the current output of Comparator 3 (CM3CON<8>).

bit 1 **C2OUT:** Comparator 2 Output Status bit (read-only)

Shows the current output of Comparator 2 (CM2CON<8>).

bit 0 C10UT: Comparator 1 Output Status bit (read-only)

Shows the current output of Comparator 1 (CM1CON<8>).

TABLE 30-10: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	Symbol	Characteristics	Min	Тур	Max	Units	Comments
	VRGOUT	Regulator Output Voltage	1	1.8	_	V	
	VBG	Internal Band Gap Reference	ı	1.2	_	V	
	CEFC	External Filter Capacitor Value	4.7	10	_	μF	Series resistance < 3 Ohm recommended; < 5 Ohm required.
	TVREG		_	10	_	μS	VREGS = 1, VREGS = 0 with WUTSEL<1:0> = 01 or any POR or BOR
			_	190	_	μS	Sleep wake-up with VREGS = 0 and WUTSEL<1:0> = 11
	TBG	Band Gap Reference Start-up Time	_	1		ms	

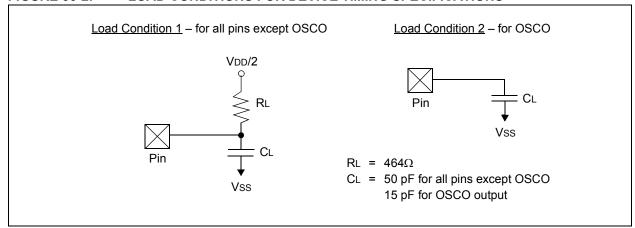
30.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24FJ256DA210 family AC characteristics and timing parameters.

TABLE 30-11: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC

	Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated)				
AC CHARACTERISTICS	Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
	Operating voltage VDD range as described in Section 30.1 "DC Characteristics" .				

FIGURE 30-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



NOTES:

APPENDIX A: REVISION HISTORY

Revision A (February 2010)

Original data sheet for the PIC24FJ256DA210 family of devices.

Revision B (May 2010)

Minor changes throughout text and the values in **Section 30.0 "Electrical Characteristics"** were updated.

INDEX

A		Shared I/O Port Structure	157
A/D Conversion		SPI Master, Frame Master Connection	220
10-Bit High-Speed A/D Converter	325	SPI Master, Frame Slave Connection	220
A/D Converter		SPI Master/Slave Connection	
Analog Input Model		(Enhanced Buffer Modes)	219
Transfer Function		SPI Master/Slave Connection	
AC Characteristics	333	(Standard Mode)	219
ADC Conversion Timing	395	SPI Slave, Frame Master Connection	220
CLKO and I/O Timing		SPI Slave, Frame Slave Connection	220
Internal RC Accuracy		SPIx Module (Enhanced Mode)	213
Alternate Interrupt Vector Table (AIVT)		SPIx Module (Standard Mode)	212
Alternative Master	33	System Clock	141
EPMP	273	Triple Comparator Module	335
Assembler	270	UART (Simplified)	231
MPASM Assembler	360	USB OTG	
WII / COW / COCHIDICI		Device Mode Power Modes	241
В		USB OTG Interrupt Funnel	248
Block Diagram		USB OTG Module	240
CRC	297	Watchdog Timer (WDT)	356
Block Diagrams		С	
10-Bit High-Speed A/D Converter	326		
16-Bit Asynchronous Timer3 and Timer5		C Compilers	
16-Bit Synchronous Timer2 and Timer4		MPLAB C18	
16-Bit Timer1 Module		Charge Time Measurement Unit (CTMU)	
32-Bit Timer2/3 and Timer4/5		Key Features	343
96 MHz PLL	150	Charge Time Measurement Unit. See CTMU.	
Accessing Program Space Using Table		Code Examples	
Operations	77	Basic Sequence for Clock Switching in	
Addressing for Table Registers		Assembly	149
BDT Mapping for Endpoint Buffering Modes		Configuring UART1 I/O Input/Output	
CALL Stack Frame		Functions (PPS)	
Comparator Voltage Reference	341	EDS Read From Program Memory in Assembly	
CPU Programmer's Model		EDS Read in Assembly	
CRC Shift Engine Detail		EDS Write in Assembly	
CTMU Connections and Internal Configuration		Erasing a Program Memory Block (Assembly)	
for Capacitance Measurement	343	I/O Port Read/Write in 'C'	
CTMU Typical Connections and Internal		I/O Port Read/Write in Assembly	
Configuration for Pulse Delay Generation	344	Initiating a Programming Sequence	
CTMU Typical Connections and Internal		PWRSAV Instruction Syntax	
Configuration for Time Measurement	344	Setting the RTCWREN Bit	
Data Access From Program Space		Single-Word Flash Programming	00
Address Generation		Single-Word Flash Programming	06
Graphics Module Overview		('C' Language) Code Protection	
I ² C Module		Code Segment Protection	
Individual Comparator Configurations,		Configuration Options	
CREF = 0	336	Configuration Protection	
Individual Comparator Configurations,		Comparator Voltage Reference	
CREF = 1 and CVREFP = 0	337	Configuring	
Individual Comparator ConfigurationS,		Configuration Bits	
CREF = 1 and CVREFP = 1		Core Features	
Input Capture		CPU	10
On-Chip Regulator Connections		Arithmetic Logic Unit (ALU)	43
Output Compare (16-Bit Mode)	202	Control Registers	
Output Compare (Double-Buffered,	004	Core Registers	
16-Bit PWM Mode)		Programmer's Model	
PCI24FJ256DA210 Family (General)		CRC	00
PIC24F CPU Core		32-Bit Programmable Cyclic	
PSV Operation (Higher Word)		Redundancy Check	297
PSV Operation (Lower Word)		Polynomials	
Reset System		Setup Examples for 16 and 32-Bit Polynomials	
RTCC	285	User Interface	

K		Program Memory	
Key Features	347	Access Using Table Instructions	77
CTMU		Address Construction	75
EPMP		Address Space	45
RTCC		Flash Configuration Words	46
1.100	203	Memory Maps	45
M		Organization	46
Memory Organization	45	Reading From Program Memory Using EDS	78
Microchip Internet Web Site		Pulse-Width Modulation (PWM) Mode	
MPLAB ASM30 Assembler, Linker, Librarian		Pulse-Width Modulation. See PWM.	
MPLAB Integrated Development		PWM	
Environment Software	359	Duty Cycle and Period	204
MPLAB PM3 Device Programmer		D	
MPLAB REAL ICE In-Circuit Emulator System		R	
MPLINK Object Linker/MPLIB Object Librarian		Reader Response	406
Wil Ellar Object Ellinon Wil Elb Object Elbraham		Reference Clock Output	152
N		Register Maps	
Near Data Space	49	A/D Converter	61
-		ANCFG	64
0		ANSEL	64
Oscillator Configuration		Comparators	66
96 MHz PLL	149	CPU Core	50
Clock Selection		CRC	66
Clock Switching		CTMU	62
Sequence		Enhanced Parallel Master/Slave Port	65
CPU Clocking Scheme		Graphics	
Display Clock Frequency Division		I ² C™	56
Initial Configuration on POR		ICN	51
USB Operations		Input Capture	54
Output Compare		Interrupt Controller	52
32-Bit Mode	201	NVM	70
Synchronous and Trigger Modes		Output Compare	55
Output Compare with Dedicated Timers		Pad Configuration	60
		Peripheral Pin Select	
P		PMD	70
Packaging	387	PORTA	58
Details	388	PORTB	58
Marking	387	PORTC	59
Peripheral Enable Bits	156	PORTD	59
Peripheral Module Disable Bits	156	PORTE	59
Peripheral Pin Select (PPS)	164	PORTF	60
Available Peripherals and Pins	164	PORTG	60
Configuration Control	167	RTCC	65
Considerations for Use		SPI	58
Input Mapping	164	System	70
Mapping Exceptions	167	Timers	53
Output Mapping	166	UART	57
Peripheral Priority	164	USB OTG	63
Registers	169	Registers	
Pin Descriptions		AD1CHS (A/D Input Select)	330
100-Pin Devices	8	AD1CON1 (A/D Control 1)	327
121 (BGA)-Pin Devices	11	AD1CON2 (A/D Control 2)	
64-Pin Devices	6	AD1CON3 (A/D Control 3)	
Pin Diagrams		AD1CSSH (A/D Input Scan Select, High)	
100-Pin TQFP	7	AD1CSSL (A/D Input Scan Select, Low)	331
121-Pin BGA	10	ALCFGRPT (Alarm Configuration)	
64-Pin TQFP/QFN	5	ALMINSEC (Alarm Minutes and Seconds Value)	293
Pinout Descriptions	21	ALMTHDY (Alarm Month and Day Value)	292
POR		ALWDHR (Alarm Weekday and Hours Value)	293
and On-Chip Voltage Regulator	354	ANCFG (A/D Band Gap Reference	
Power-Saving		Configuration)	331
Clock Frequency and Clock Switching	155	ANSA (PORTA Analog Function Selection)	
Features		ANSB (PORTB Analog Function Selection)	
Instruction-Based Modes		ANSC (PORTC Analog Function Selection)	
Power-up Requirements		ANSD (PORTD Analog Function Selection)	161
Product Identification System		ANSE (PORTE Analog Function Selection)	161