

Conectividad USB 2.0



**Octubre
de 2012**

Instructores

Ing. Gamaliel Torres Sánchez

Ing. Julio Lugo Jiménez

Ing. Gabriel Pool Balam

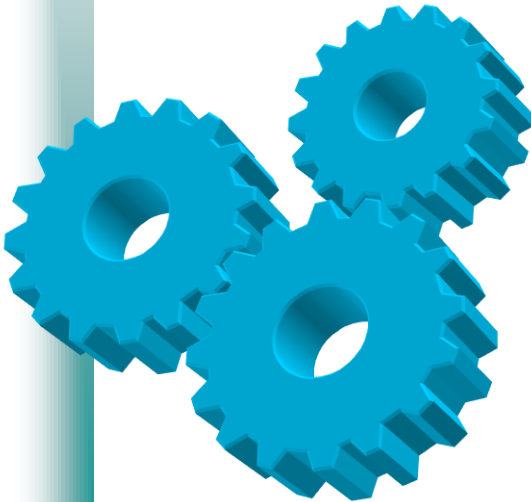
Ing. Juan Molina Amaro



1.- Introducción a la familia de Pic's 18Fxx5x

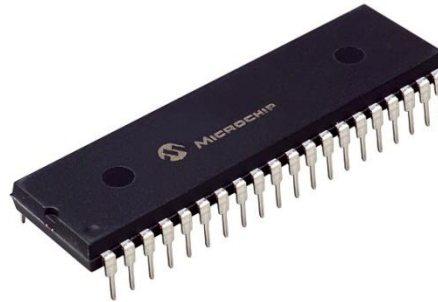
2.- Introducción al Compilador CCS

3.- Introducción al USB



Tema 1

Características de la familia del PIC18Fxx5x



Características de la familia PIC18Fxx5x

La nueva familia PIC18F se presentó como una nueva gama de Micro controladores RISC de 8bits con arquitectura Harvard que incluía los últimos avances y la mayoría de los periféricos para potenciar miles de aplicaciones centradas en las áreas de automoción, comunicaciones, ofimáticas, productos de gran consumo y control industrial.

Con la nueva familia Microchip ha resuelto dos grandes metas:

- * Compatibilidad: con los PIC16F y sus periféricos; soportando Ensamblador y C.
- * Elevado rendimiento: máxima eficacia del compilador C; elevada productividad que alcanza un rendimiento de 10MIPS a 10MHz.

Características

Aportaciones específicas de la gama mejorada:

- Tecnología “nanoWatt”: reduce el consumo de energía.
- Espacio de direccionamiento lineal: desaparecen los bancos.
- Compatibilidad de software.
- Compatibilidad de hardware.
- Orientación al lenguaje C.
- Autoprogramabilidad: Pueden escribir su memoria de programa internamente.
- Diversidad de periféricos integrados.

Actualmente Microchip dispone de la familia PIC24, dsPIC30F/33F de 16bits y los PIC32MX de 32 bits.

Principales aplicaciones

- Universal Serial Bus is a bus standard for serial communication. Originally intended as a replacement for serial and parallel connections on a PC to connect mice, keyboards and printers, it has grown to be the standard interface between a PC and its many and varied peripherals.
- With the addition of Embedded host and OTG, the interface is now expanded beyond the PC to any platforms that would benefit from its features, including:
 - Home audio – for digital file sharing, audio connections
 - Audio for automotive – for MP3 connectivity
 - Handheld scanners and meters – to download data via a USB key to a PC
 - Industrial control – to upgrade software, download diagnostics for analysis
 - Medical patient monitors - to upgrade software, download diagnostics for analysis

Una extensión del USB llamada "[USB-On-The-Go](#)" (*sobre la marcha*) permite a un puerto actuar como servidor o como dispositivo - esto se determina por qué lado del cable está conectado al aparato. Incluso después de que el cable está conectado y las unidades se están comunicando, las 2 unidades pueden "cambiar de papel" bajo el control de un programa. Esta facilidad está específicamente diseñada para dispositivos como [PDA](#), donde el enlace USB podría conectarse a un PC como un dispositivo, y conectarse como servidor a un teclado o ratón. El "USB-On-The-Go" también ha diseñado 2 conectores pequeños, el mini-A y el mini-B, así que esto debería detener la proliferación de conectores miniaturizados de entrada.

Diagrama de Pines

44-Pin TQFP

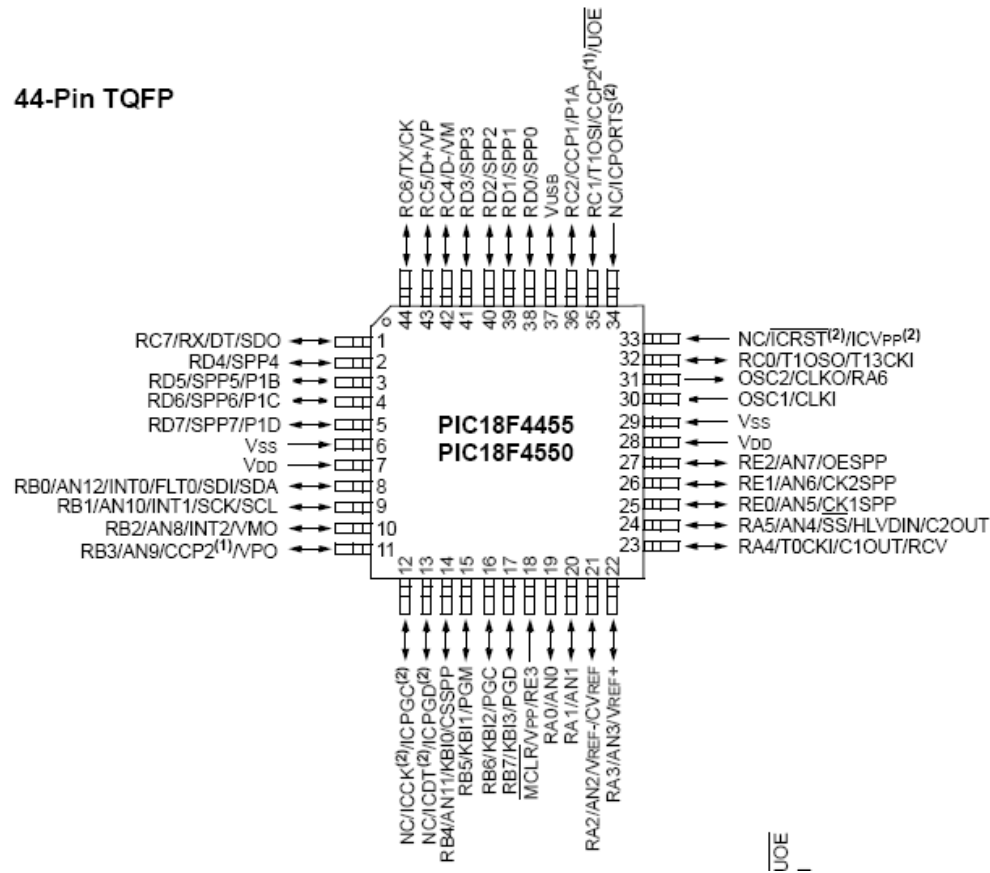
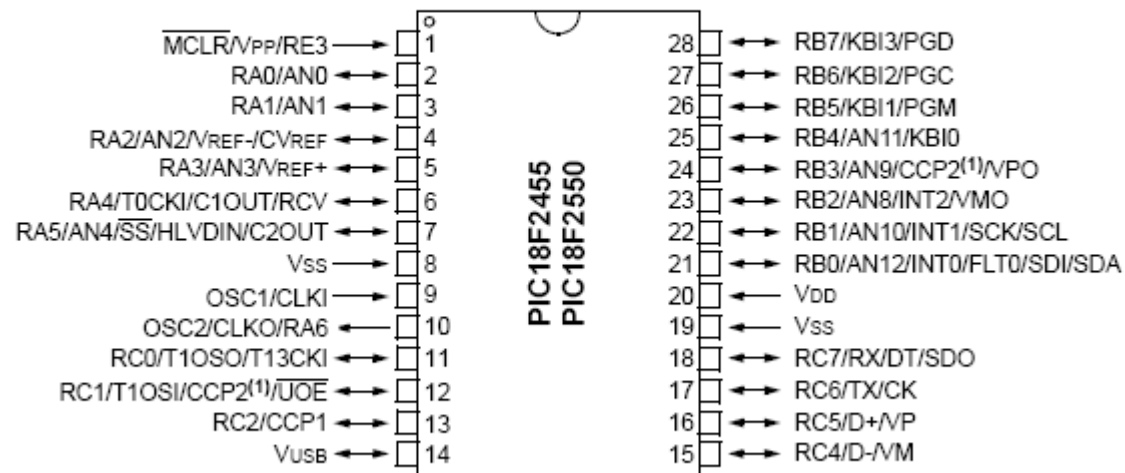
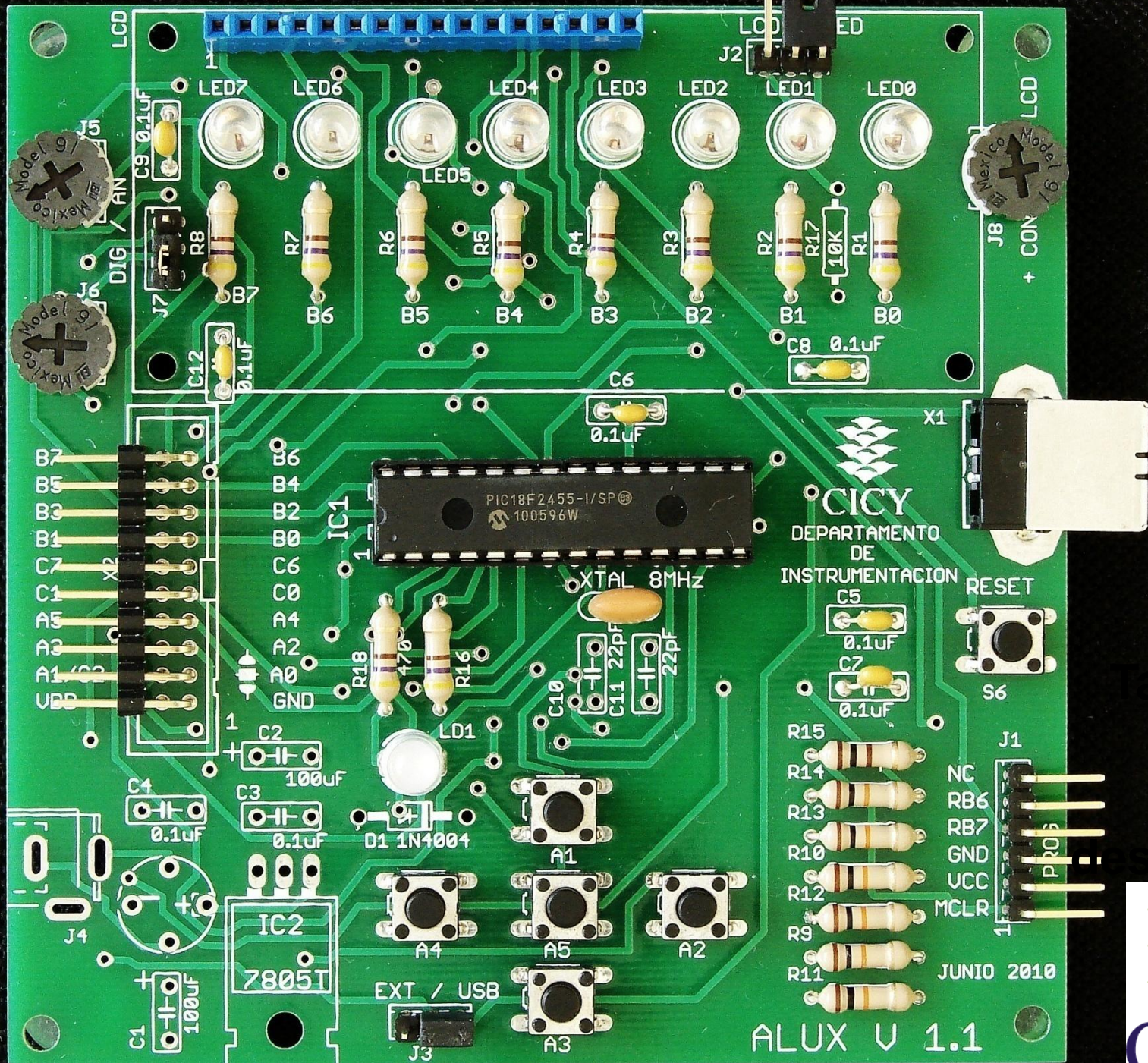


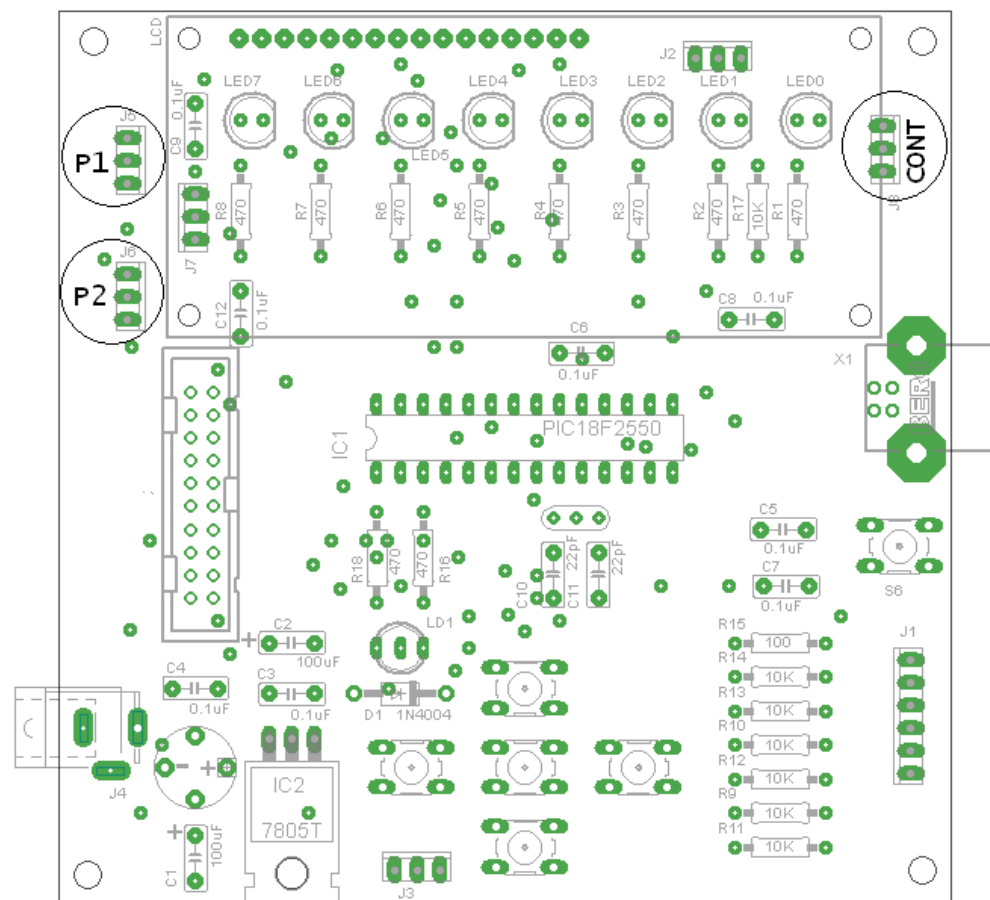
Diagrama de Pines

28-Pin PDIP, SOIC





Distribución de componentes





Sheet: 1/1

Características del módulo USB

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

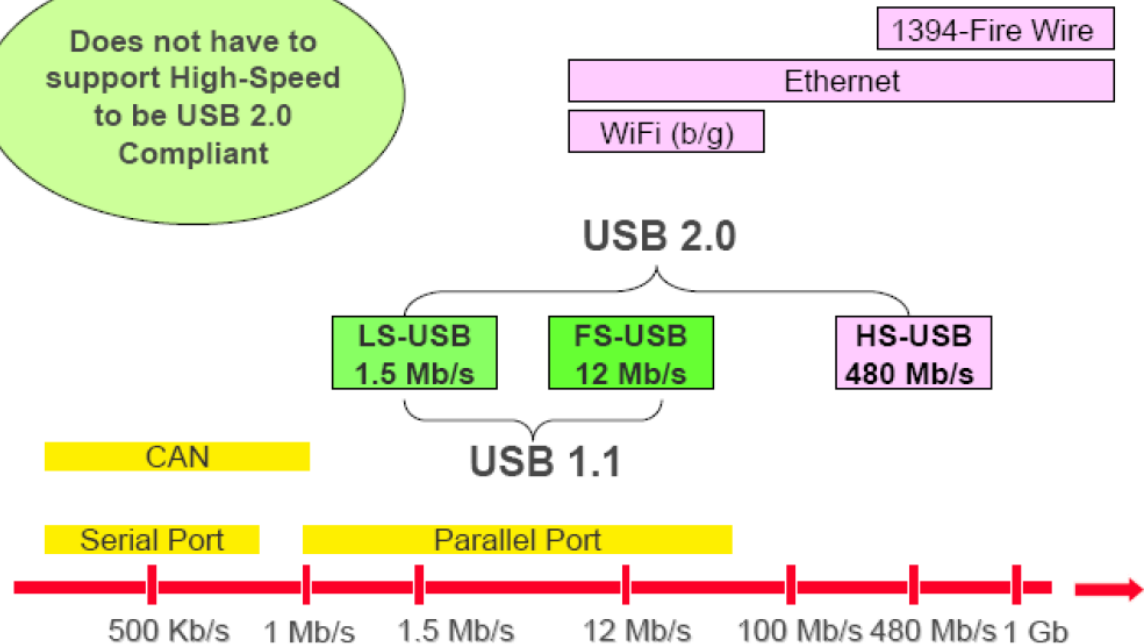
Características del módulo USB



Buses and Speed Comparison

**USB Complete
VS
Datasheet**

Does not have to
support High-Speed
to be USB 2.0
Compliant



USB Complete VS Datasheet

Table 6-1: USB controller chips that are compatible with popular microcontroller families are available from many sources.

Compatibility	Manufacturer	Chips	Bus Speed
Atmel AVR	Atmel	AT43USB35x, AT76C713	Full
Freescale/Motorola 68HC05	Freescale Semiconductor	68HC05JB3/4	Low
Freescale/Motorola 68HC08	Freescale Semiconductor	68HC08JB8	Low
Freescale/Motorola PowerPC	Freescale Semiconductor	MCF5482	Full/High
Infineon C166	Infineon	C161U	Full
Intel 80C186	AMD	Am186CC	Full
Intel 8051	Atmel	AT89C513x	Full
	Cypress Semiconductor	EZ-USB, EZ-USB FX	Full
		EZ-USB FX2	Full/High
	Prolific Technology	PL-23xx	Full
		PL-25xx	Full/High
	Silicon Laboratories	C8051F32x	Full
	Standard Microsystems Corporation (SMSC)	USB97Cxxx, USB222x	Full, Full/High
	Texas Instruments	TUSB3210/3410	Full
		TUSB6250	Full/High
Microchip PIC16	Microchip Technology	PIC16C7x5	Low
Microchip PIC18	Microchip Technology	PIC18F2455/2550/ 4455/4550	Full/High
STMicroelectronics ST7, ST9	STMicroelectronics	ST7265X, ST7263, ST92163	Low, Full

OSCILLATOR CONFIGURATIONS

FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM

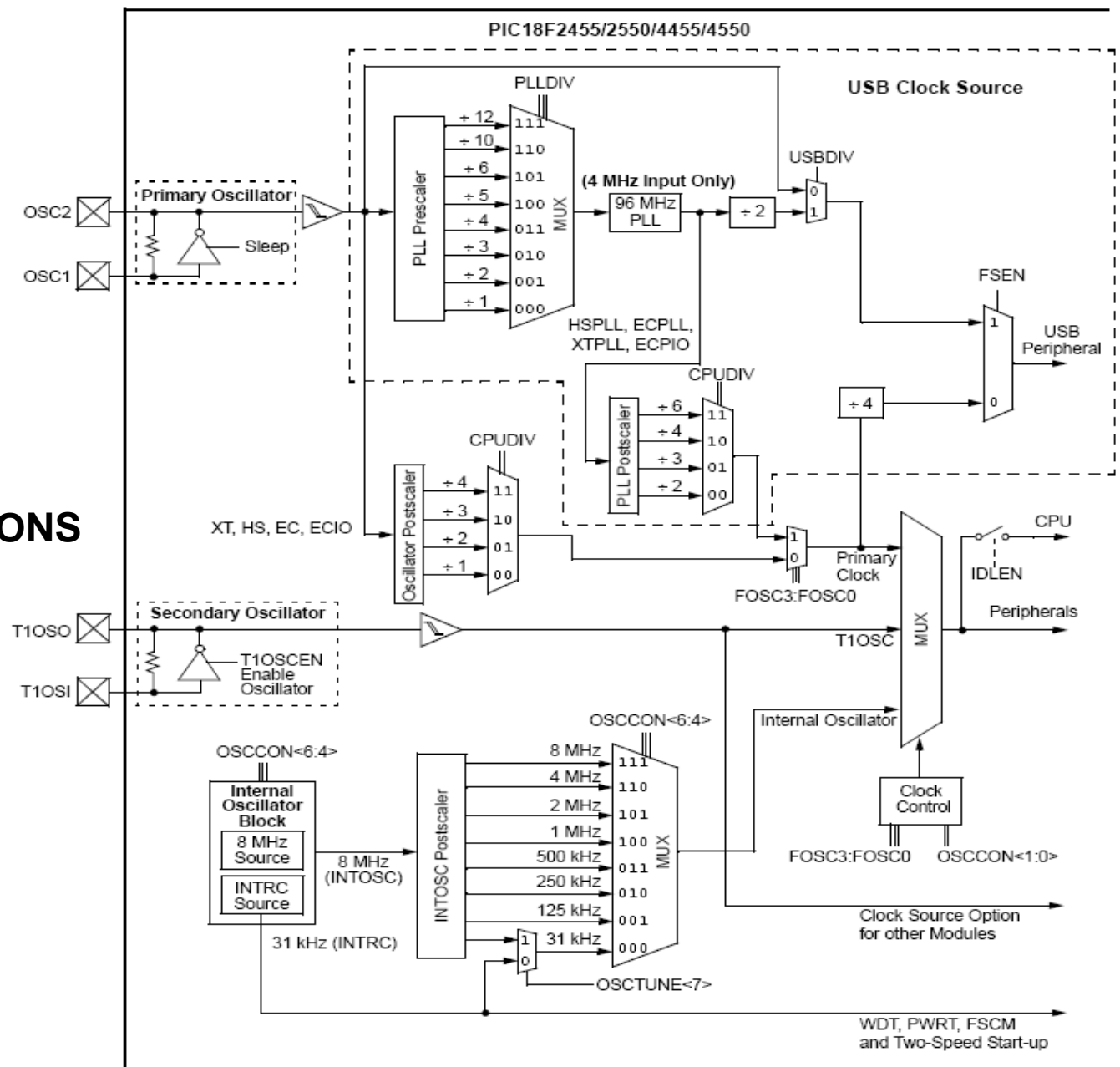
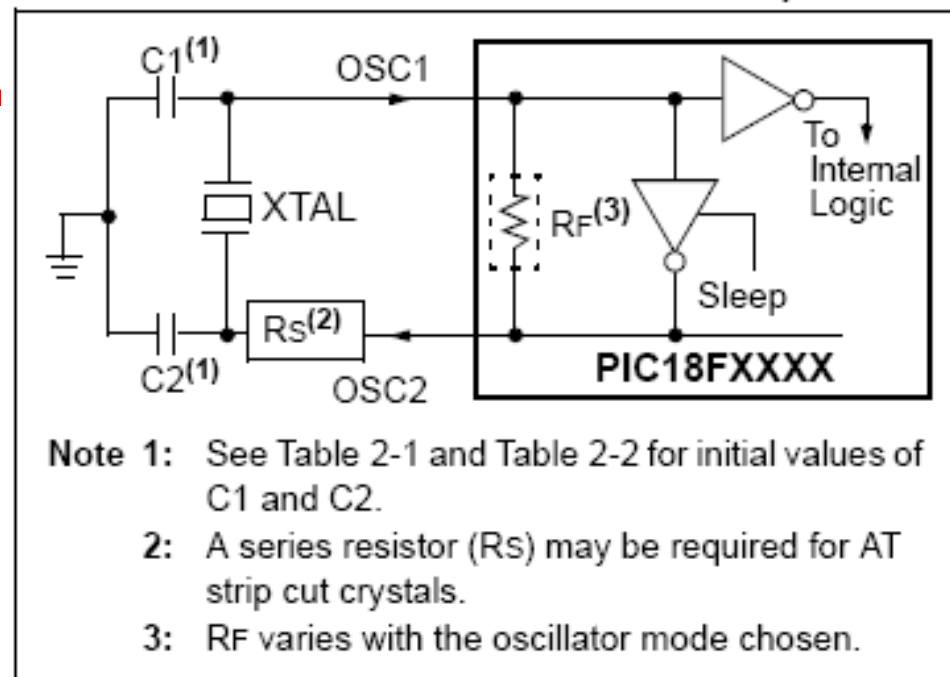


FIGURE 2-2: CRYSTAL/CERAMIC RESONATOR OPERATION (XT, HS OR HSPLL CONFIGURATION)

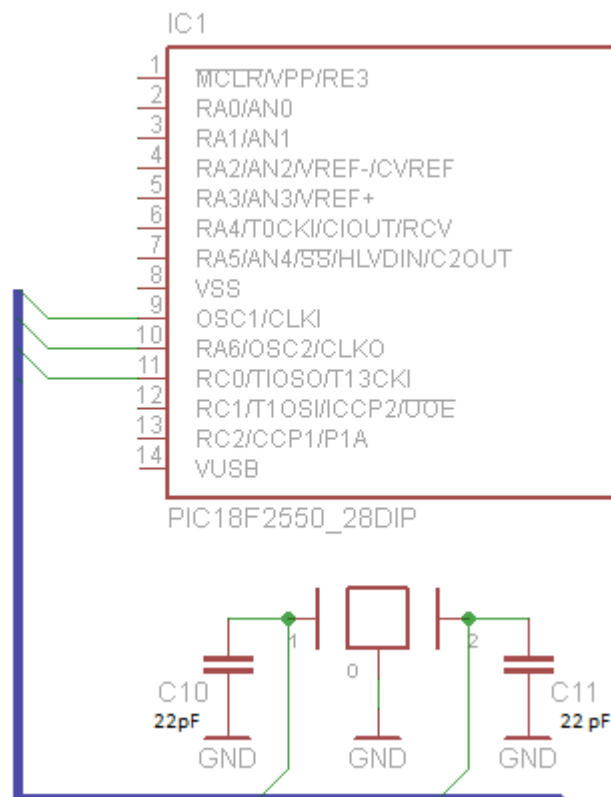


Xtal

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	Freq	OSC1	OSC2
XT	4.0 MHz	33 pF	33 pF
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

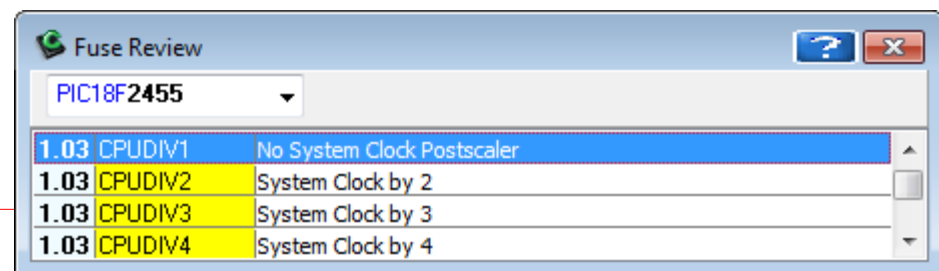
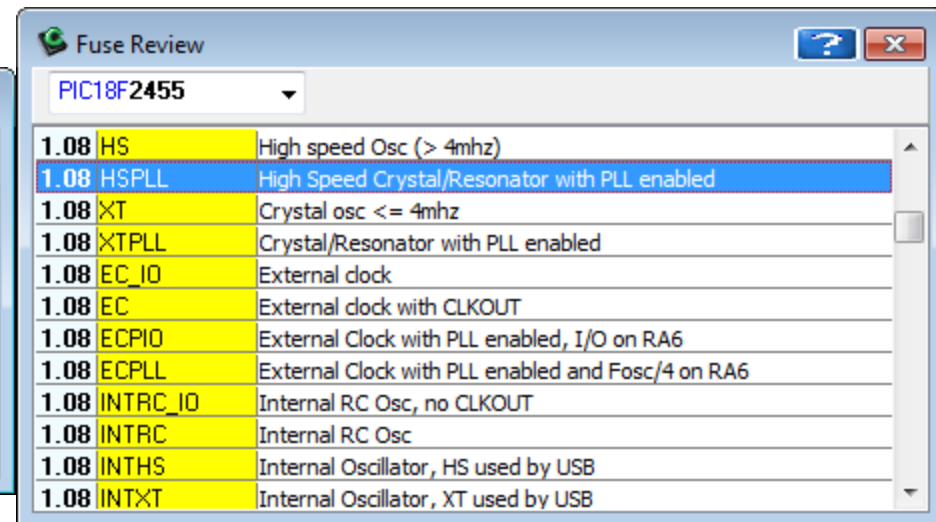
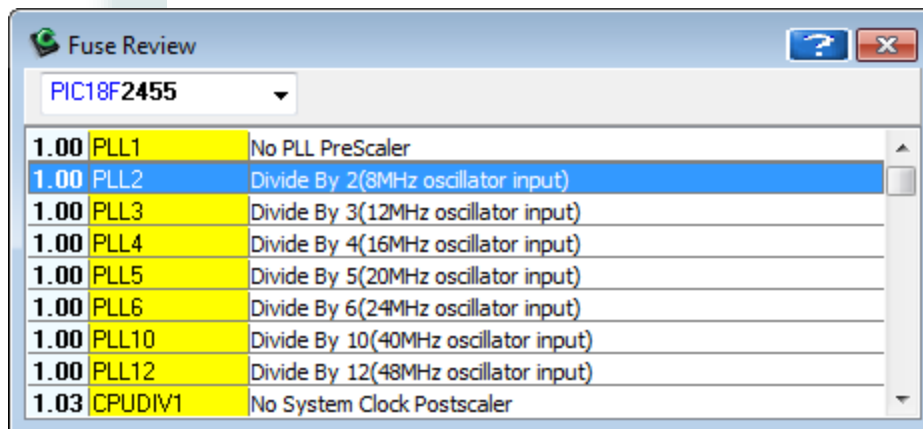
OSCILLATOR CONFIGURATIONS



Ver componentes [>>](#)

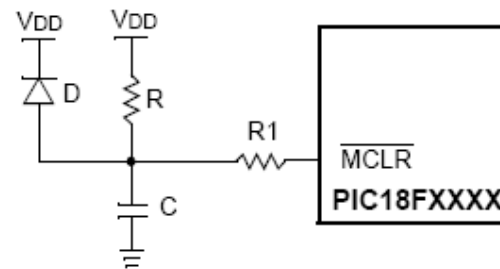
OSCILLATOR CONFIGURATIONS

```
#fuses HSPLL, PLL2, CPUDIV1
// Paso 2: Ajuste el fusible del PLL de acuerdo al Xtal que utilice
// No olvide que PLL1 = Para un Xtal de 4Mhz
//           PLL2 = Para un Xtal de 8Mhz
//           PLL3 = Para un Xtal de 12Mhz
//           PLL4 = Para un Xtal de 20Mhz , etc.
#use delay (clock=48000000)
```



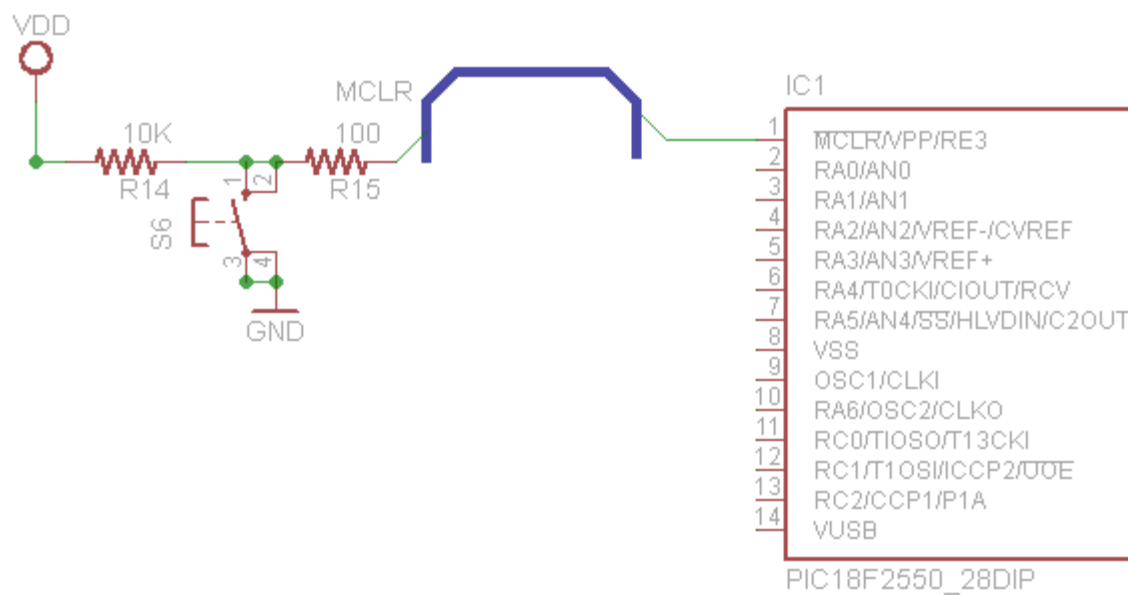
Reset

FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V_{DD} POWER-UP)



- Note 1:** External Power-on Reset circuit is required only if the V_{DD} power-up slope is too slow. The diode D helps discharge the capacitor quickly when V_{DD} powers down.
- 2:** $R < 40 \text{ k}\Omega$ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.
- 3:** $R1 \geq 1 \text{ k}\Omega$ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor C, in the event of $\overline{\text{MCLR}}/\text{VPP}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

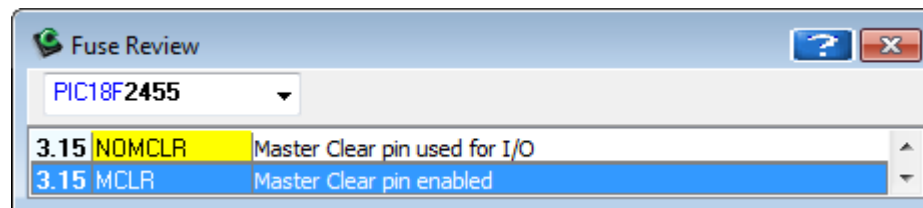
Reset

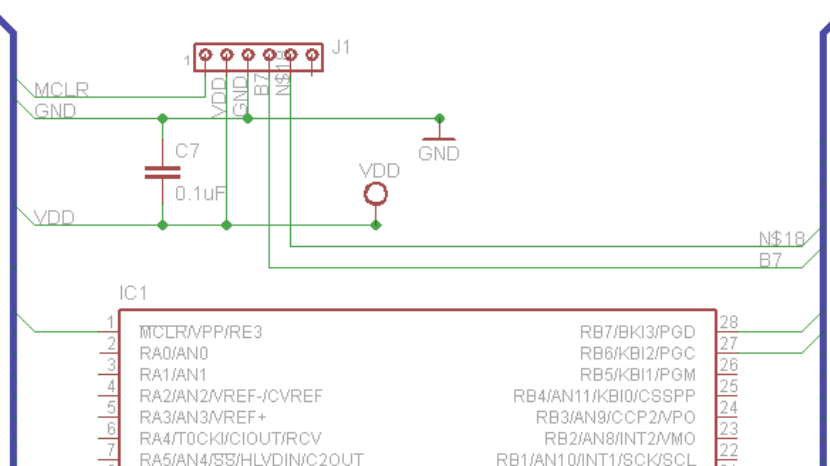


Ver componentes >>

Reset

#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL2, CPUDIV1, VREGEN, MCLR, NOPBADEN





ICSP

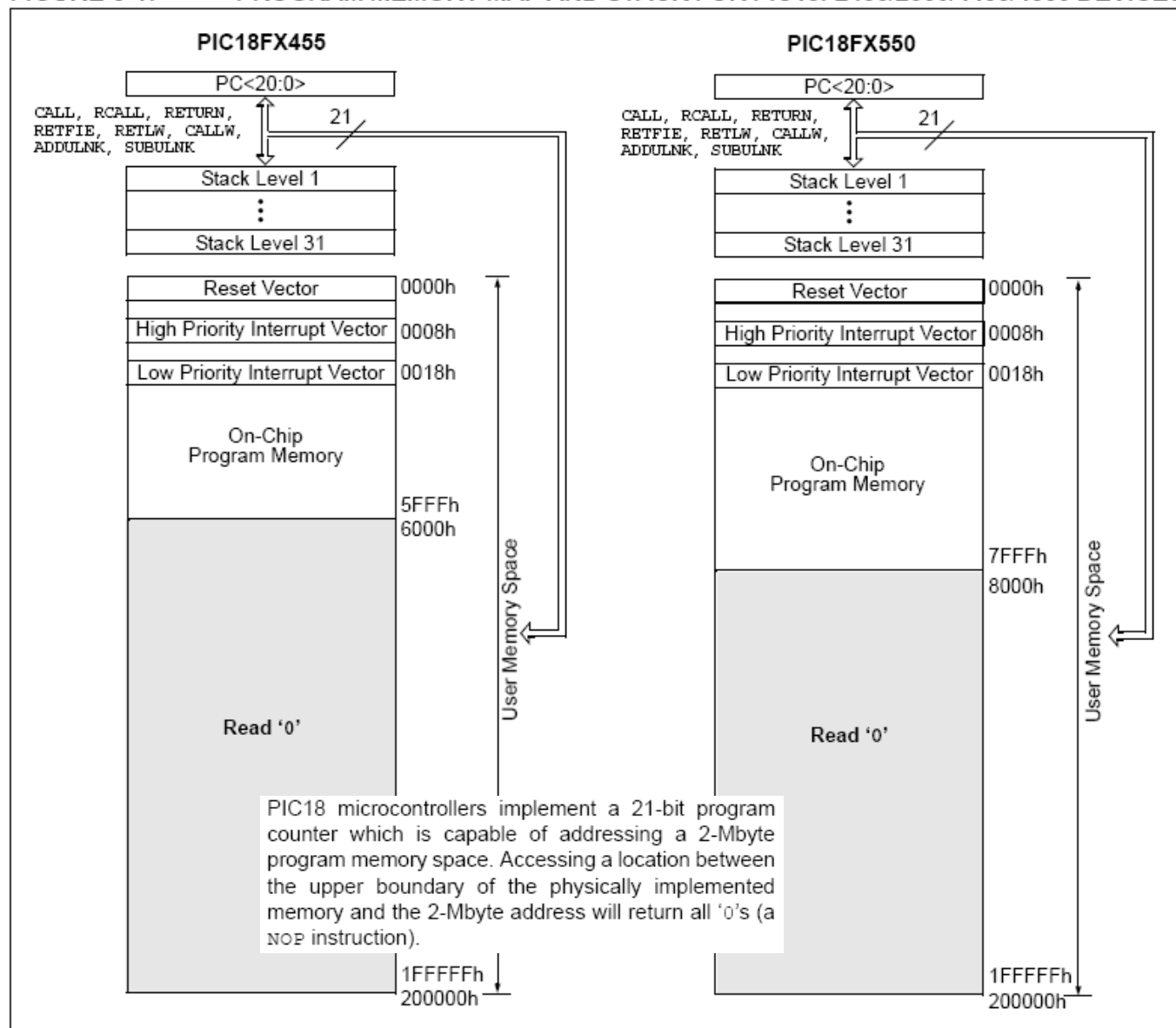
Ver componentes >>

TABLE 25-5: EQUIVALENT PINS FOR LEGACY AND DEDICATED ICD/ICSP™ PORTS

Pin Name		Pin Type	Pin Function
Legacy Port	Dedicated Port		
MCLR/VPP/RE3	NC/ICRST/ICVPP	P	Device Reset and Programming Enable
RB6/KBI2/PGC	NC/ICCK/ICPGC	I	Serial Clock
RB7/KBI3/PGD	NC/ICDT/ICPGD	I/O	Serial Data

Legend: I = Input, O = Output, P = Power

FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2455/2550/4455/4550 DEVICES



Memory Map

Register

TABLE 5-1: SPECIAL FUNCTION REGISTER MAP FOR PIC18F2455/2550/4455/4550 DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	— ⁽²⁾	F7Ch	UEP12
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	— ⁽²⁾	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	FB9h	— ⁽²⁾	F99h	— ⁽²⁾	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	— ⁽²⁾	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL	F97h	— ⁽²⁾	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE ⁽³⁾	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD ⁽³⁾	F75h	UEP5
FF4h	PRODH	FD4h	— ⁽²⁾	FB4h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	— ⁽²⁾	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	— ⁽²⁾	F70h	UEP0
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	— ⁽²⁾	F6Fh	UCFG
FEeh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAEh	RCREG	F8Eh	— ⁽²⁾	F6Eh	UADDR
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8Dh	LATE ⁽³⁾	F6Dh	UCON
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽³⁾	F6Ch	USTAT
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	— ⁽²⁾	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	— ⁽²⁾	F68h	UIR
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2 ⁽¹⁾	F87h	— ⁽²⁾	F67h	UFRMH
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	— ⁽²⁾	F66h	UFRML
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	— ⁽²⁾	F85h	— ⁽²⁾	F65h	SPPCON ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	— ⁽²⁾	F84h	PORTE	F64h	SPPEPS ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	— ⁽²⁾	F83h	PORTD ⁽³⁾	F63h	SPPCFG ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA ⁽³⁾
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	— ⁽²⁾
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	— ⁽²⁾

- Note** 1: Not a physical register.
 2: Unimplemented registers are read as '0'.
 3: These registers are implemented only on 40/44-pin devices.

FIGURE 17-5:

IMPLEMENTATION OF USB RAM IN DATA MEMORY SPACE

USB RAM

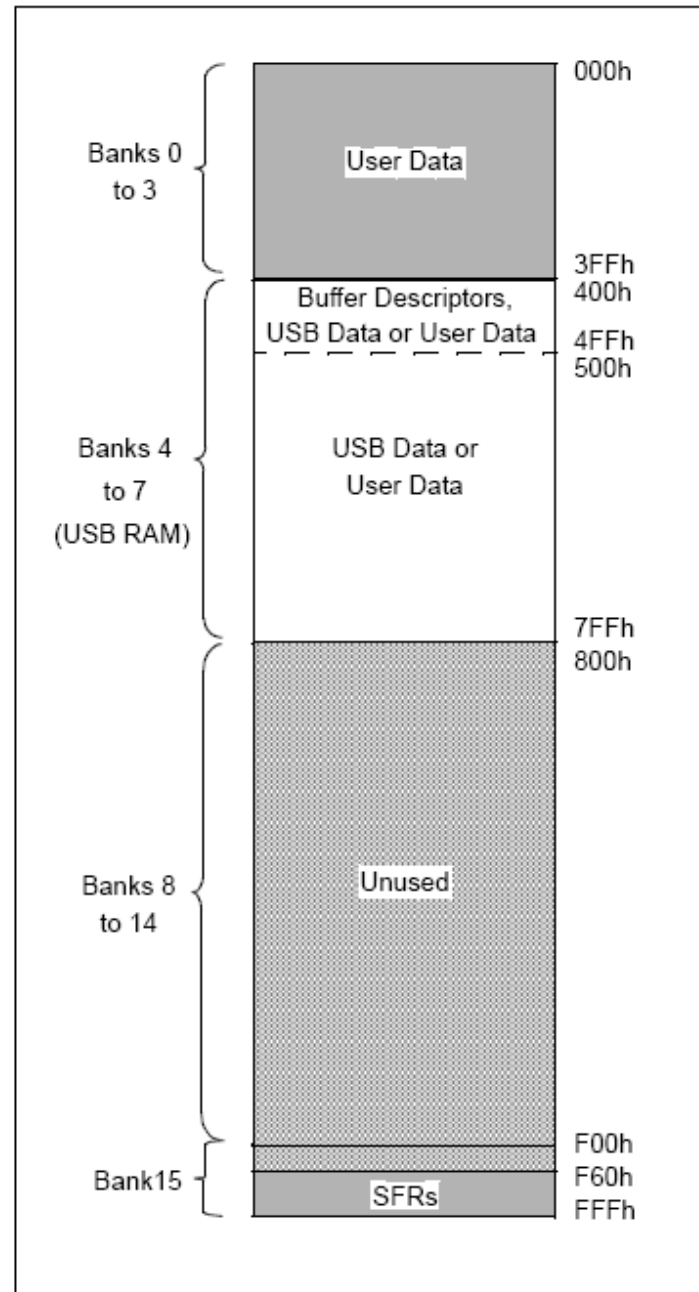
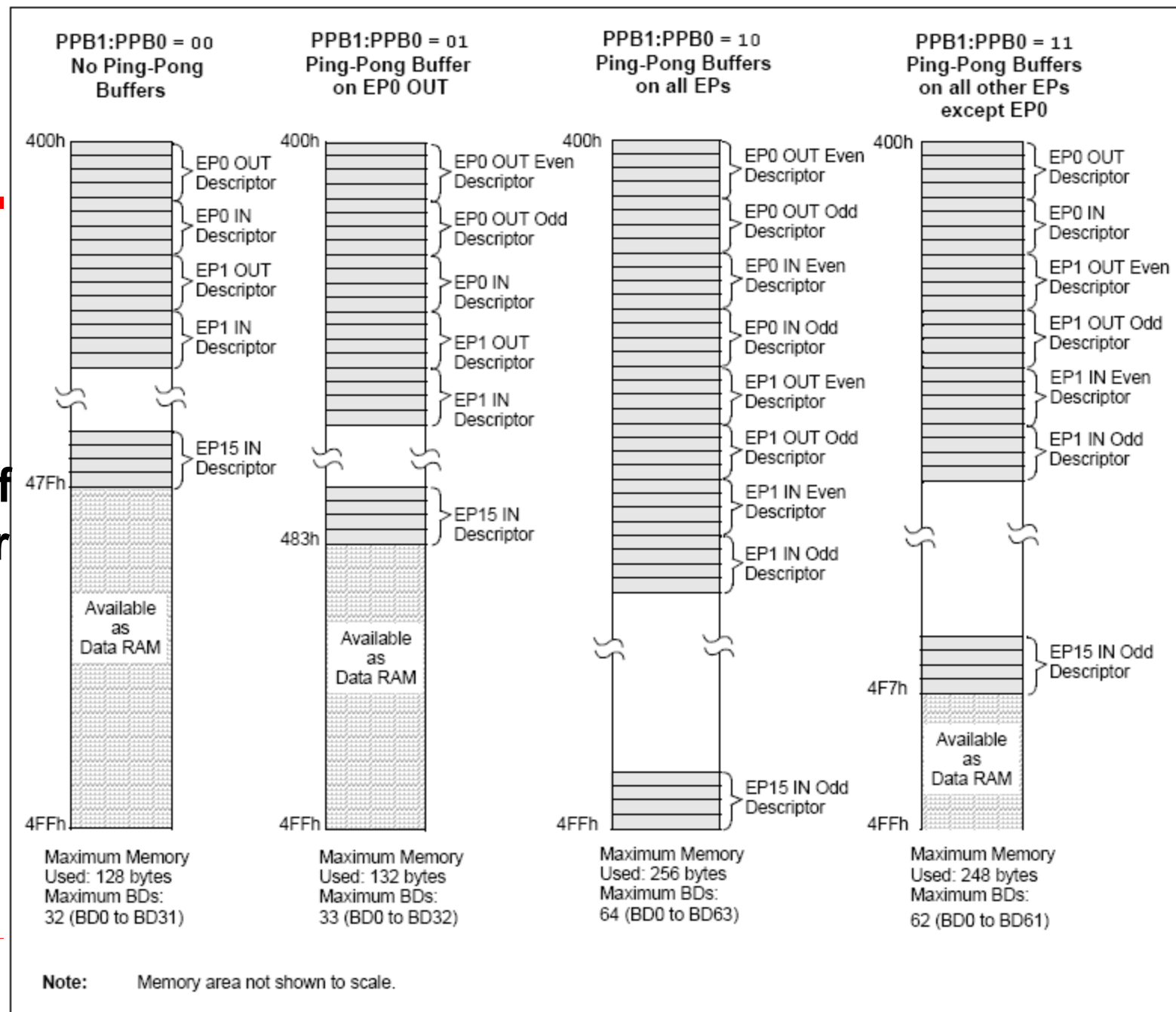


FIGURE 17-7: BUFFER DESCRIPTOR TABLE MAPPING FOR BUFFERING MODES



**Memory of
Descriptor**

Data of Descriptor

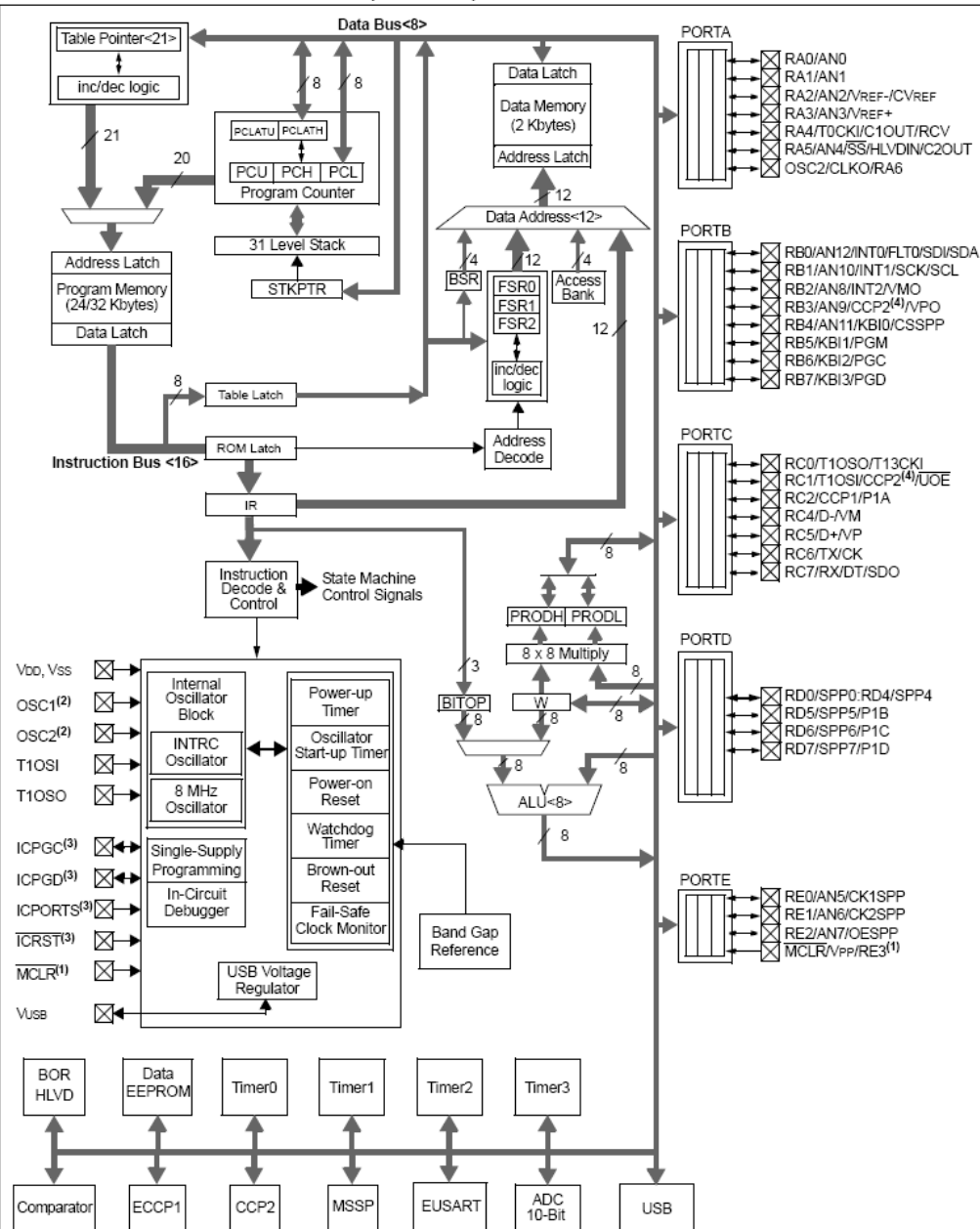
```
//device descriptor
char const USB_DEVICE_DESC[] ={
    USB_DESC_DEVICE_LEN,           //the length of this report
    0x01,                          //constant DEVICE (0x01)
    0x10,0x01,                     //usb version in bcd
    0x00,                          //class code (if 0, interface defines class. FF is vendor defined)
    0x00,                          //subclass code
    0x00,                          //protocol code
    USB_MAX_EPO_PACKET_LENGTH,     //max packet size for endpoint 0. (SLOW SPEED SPECIFIES 8)
    0xD8,0x04,                     //vendor id (0x04D8 is Microchip)
    0x0B,0x00,                     //product id
    0x01,0x00,                     //device release number
    0x01,                          //index of string description of manufacturer. therefore we point to
    0x02,                          //index of string descriptor of the product
    0x00,                          //index of string descriptor of serial number
    USB_NUM_CONFIGURATIONS        //number of possible configurations
};
```

```
#DEFINE USB_TOTAL_CONFIG_LEN      32 //config+interface+class+endpoint
```

Puertos

FIGURE 1-2:

PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM



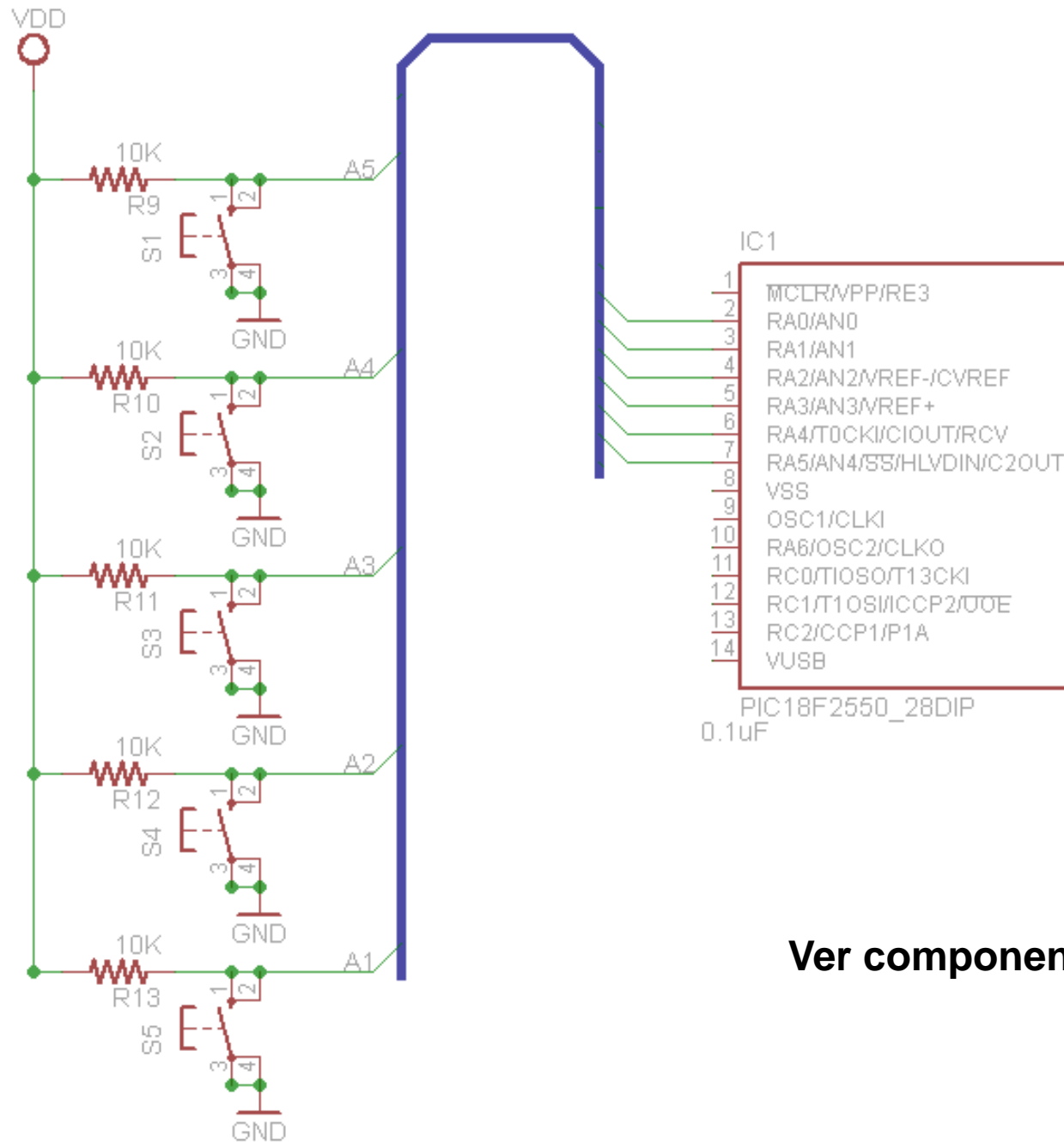
Note 1: RE3 is multiplexed with MCLR and is only available when the MCLR Resets are disabled.

2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Configurations" for additional information.

3: These pins are only available on 44-pin TQFP packages under certain conditions. Refer to Section 25.9 "Special ICPORT Features (Designated Packages Only)" for additional information.

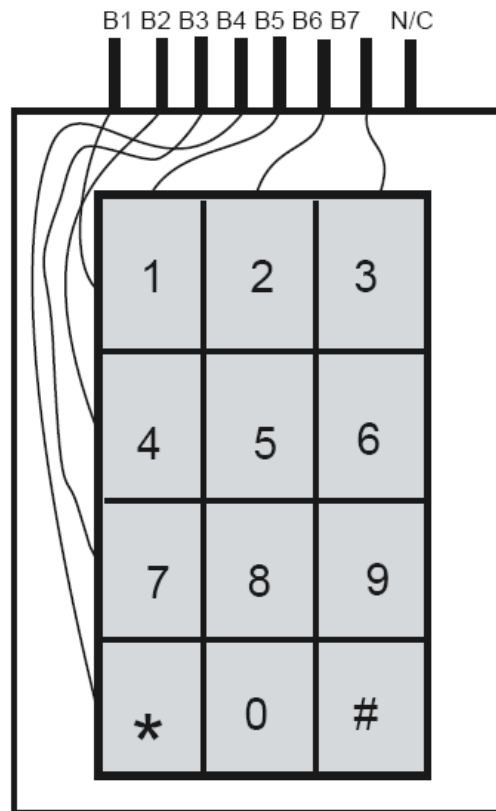
4: RB3 is the alternate pin for CCP2 multiplexing.

Puertos

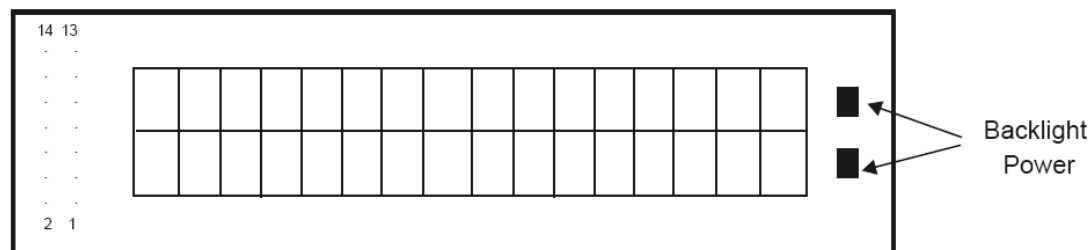


Ver componentes >>

Puertos



3x4 Keypad and
16x2 LCD



LCD Module

Puertos

1	Ground	GND
2	+5	+5
3	Contrast	Pot 10
4	RS H-Data L-Control	B1
5	R/W H-Read L-Write	B2
6	Enable	B0
7	Data 0	N/C
8	Data 1	N/C
9	Data 2	N/C
10	Data 3	N/C
11	Data 4	B4
12	Data 5	B5
13	Data 6	B6
14	Data 7	B7

Display Addresses:

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

TEMA 2

Compilando código con CCS

Manejo de puertos

1. Salida de datos digitales (puertos.c)
2. Entrada de datos digitales (leer_puerto.c)
3. Manejo de puertos apuntando a los registros de memoria (DAQ.c)

Ver [memoria](#)

Manejo del LCD

LCD (display.c)

1. Mostrar caracteres de manera directa
2. Mostrar caracteres almacenados en una variable

2: I/O pins have diode protection to VDD and VSS.



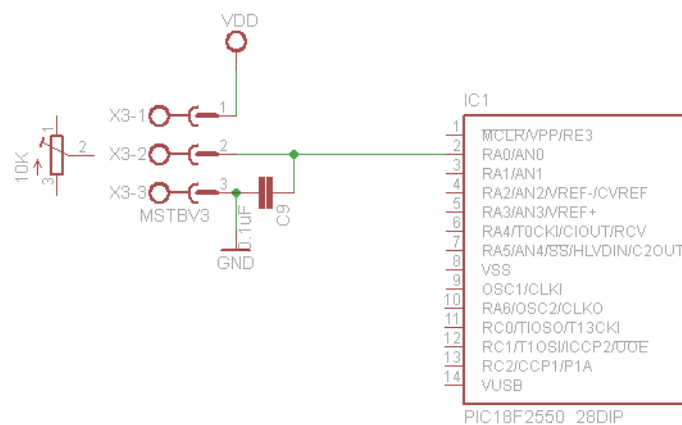
Comparators

FIGURE 22-1:

COMPARATOR I/O OPERATING MODES

<p>Comparators Reset CM2:CM0 = 000</p>	<p>Comparators Off (POR Default Value) CM2:CM0 = 111</p>
<p>Two Independent Comparators CM2:CM0 = 010</p>	<p>Two Independent Comparators with Outputs CM2:CM0 = 011</p>
<p>Two Common Reference Comparators CM2:CM0 = 100</p>	<p>Two Common Reference Comparators with Outputs CM2:CM0 = 101</p>
<p>One Independent Comparator with Output CM2:CM0 = 001</p>	<p>Four Inputs Multiplexed to Two Comparators CM2:CM0 = 110</p>
<p>A = Analog Input, port reads zeros always D = Digital Input CIS (CMCON<3>) is the Comparator Input Switch * Setting the TRISA<5:4> bits will disable the comparator outputs by configuring the pins as inputs.</p>	

ADC

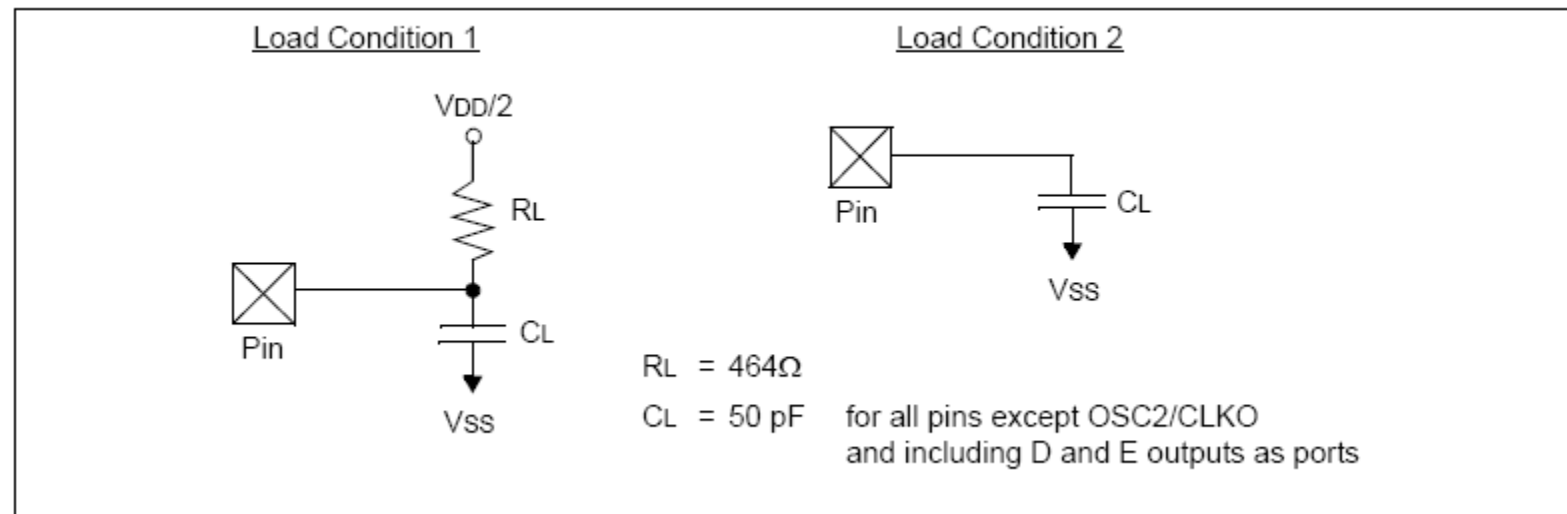


Ver componentes [>>](#)

Consideraciones de diseño:

1. Impedancia de entrada

FIGURE 28-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

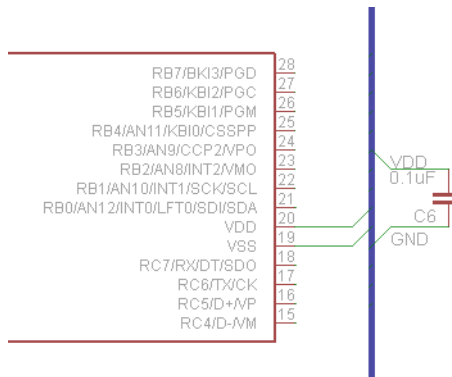


ADC

1. Leer un dato analógico de 8 bits (adc.c)
2. Leer un dato analógico de 10 bits

Consideraciones de diseño:

2. Fuente de poder



Las pistas de Vcc y GND no deben ser delgadas.

Los capacitores debe de estar lo más cerca del pin de Vcc

Consideraciones de diseño:

3. Cristal

Debe de estar lo más pegado al los pines de entrada correspondientes al crystal del microcontrolador

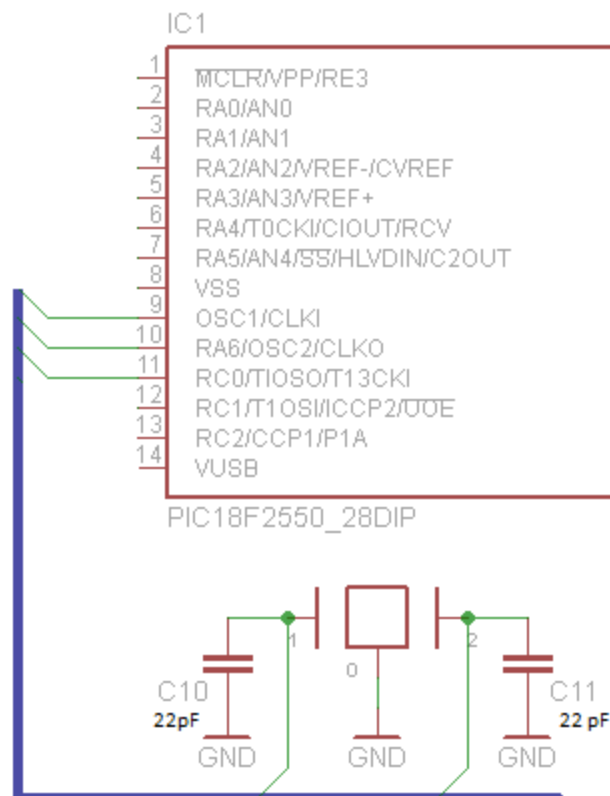
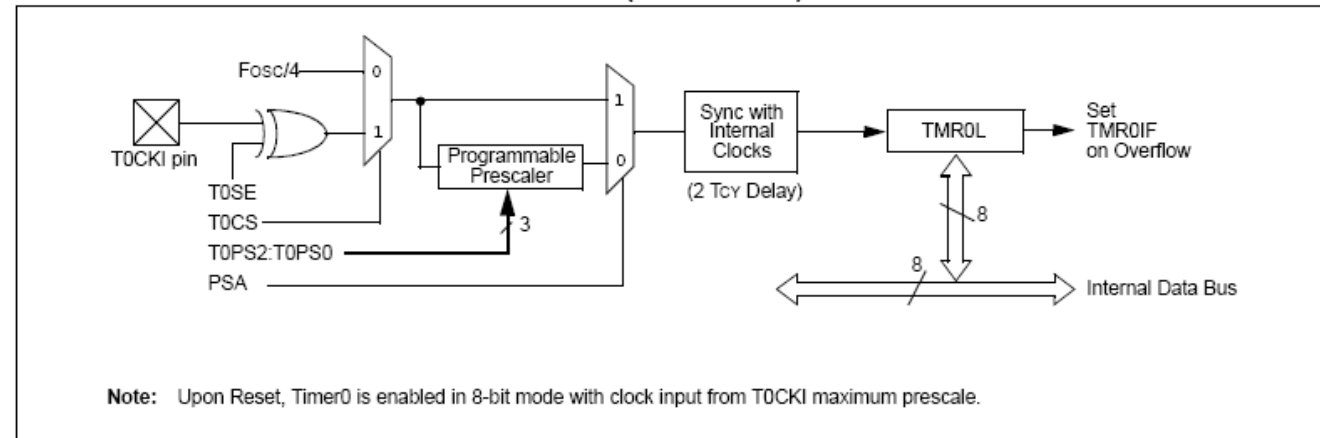
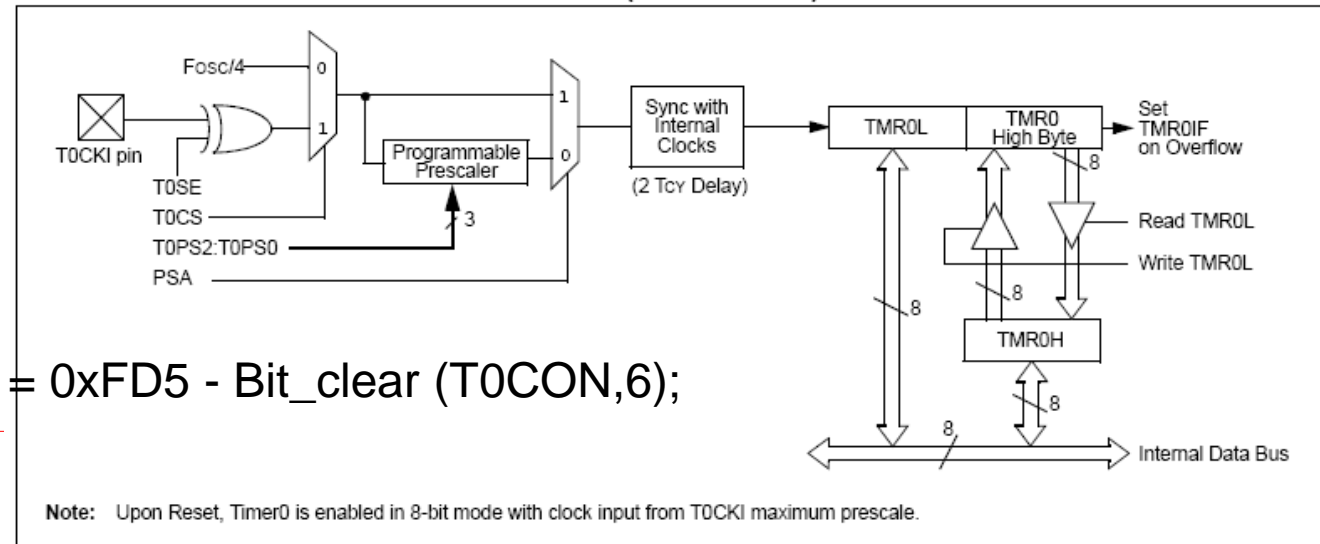


FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Timer 0

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



#byte T0CON = 0xFD5 - Bit_clear (T0CON,6);

Timer 1

FIGURE 12-1: TIMER1 BLOCK DIAGRAM

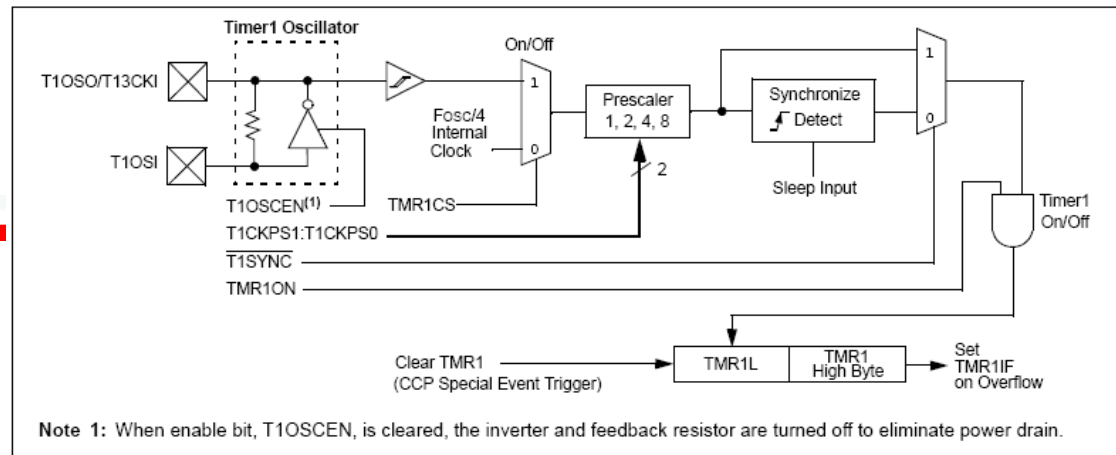
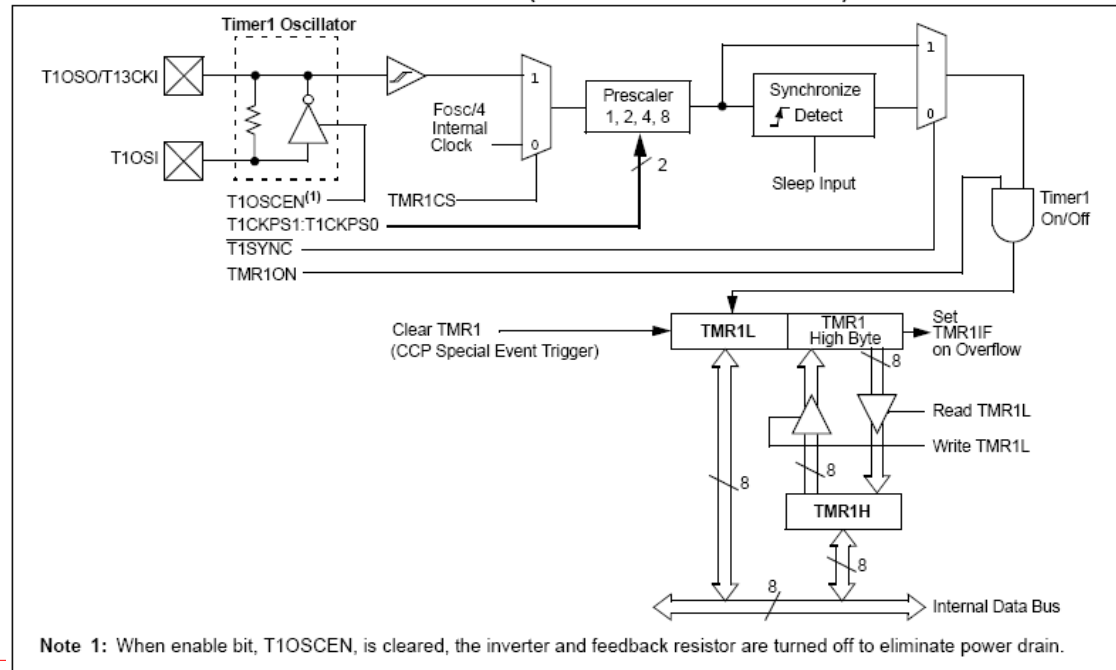
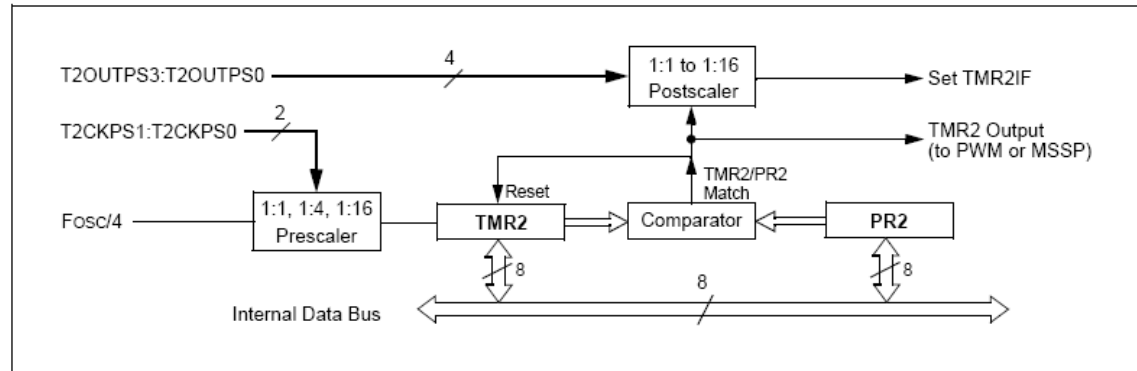


FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Timer 2

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



Timer 3

FIGURE 14-1: TIMER3 BLOCK DIAGRAM

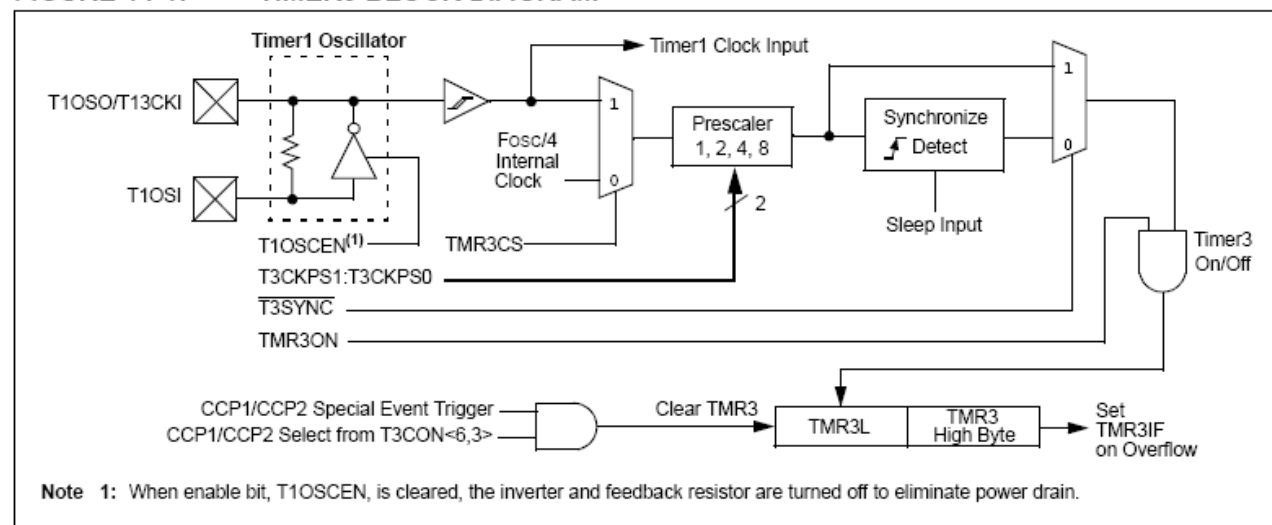
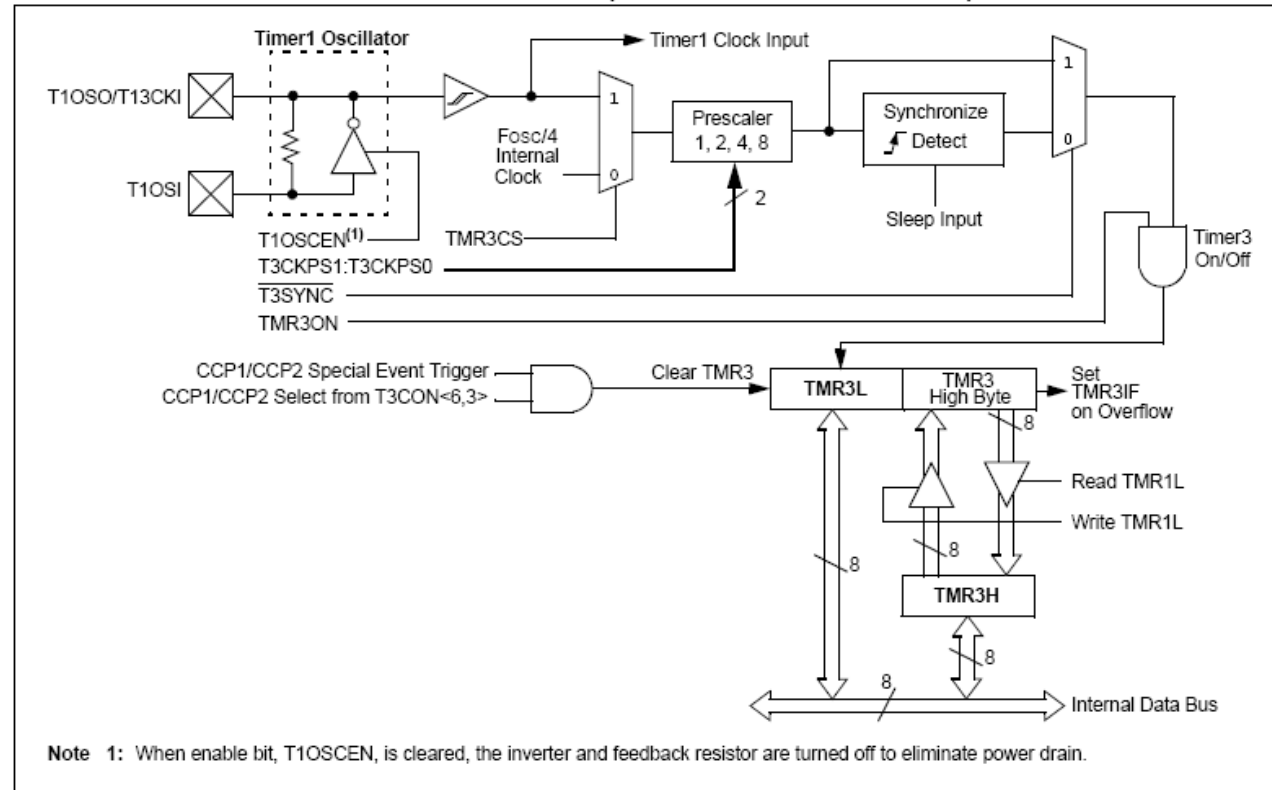


FIGURE 14-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



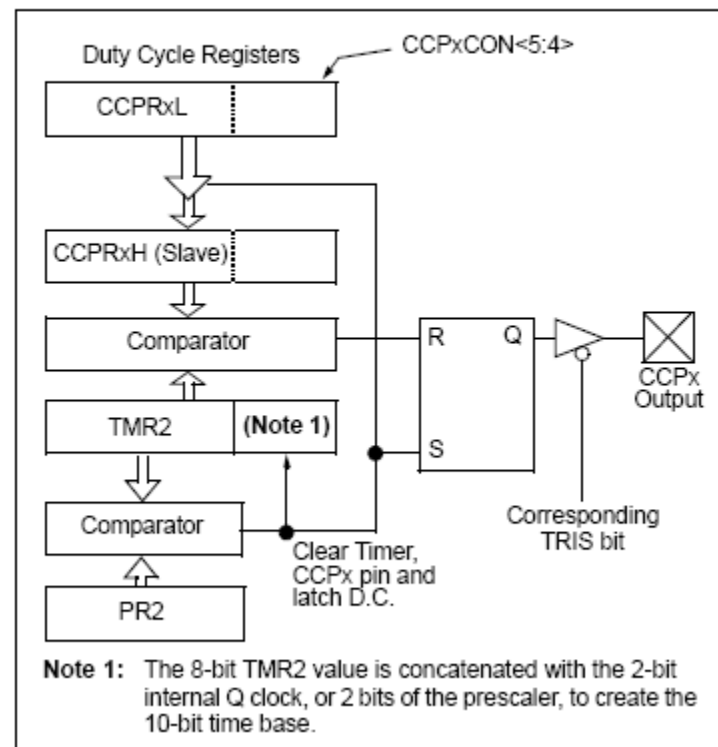
Timer

Manejo de los timer (timer.c)

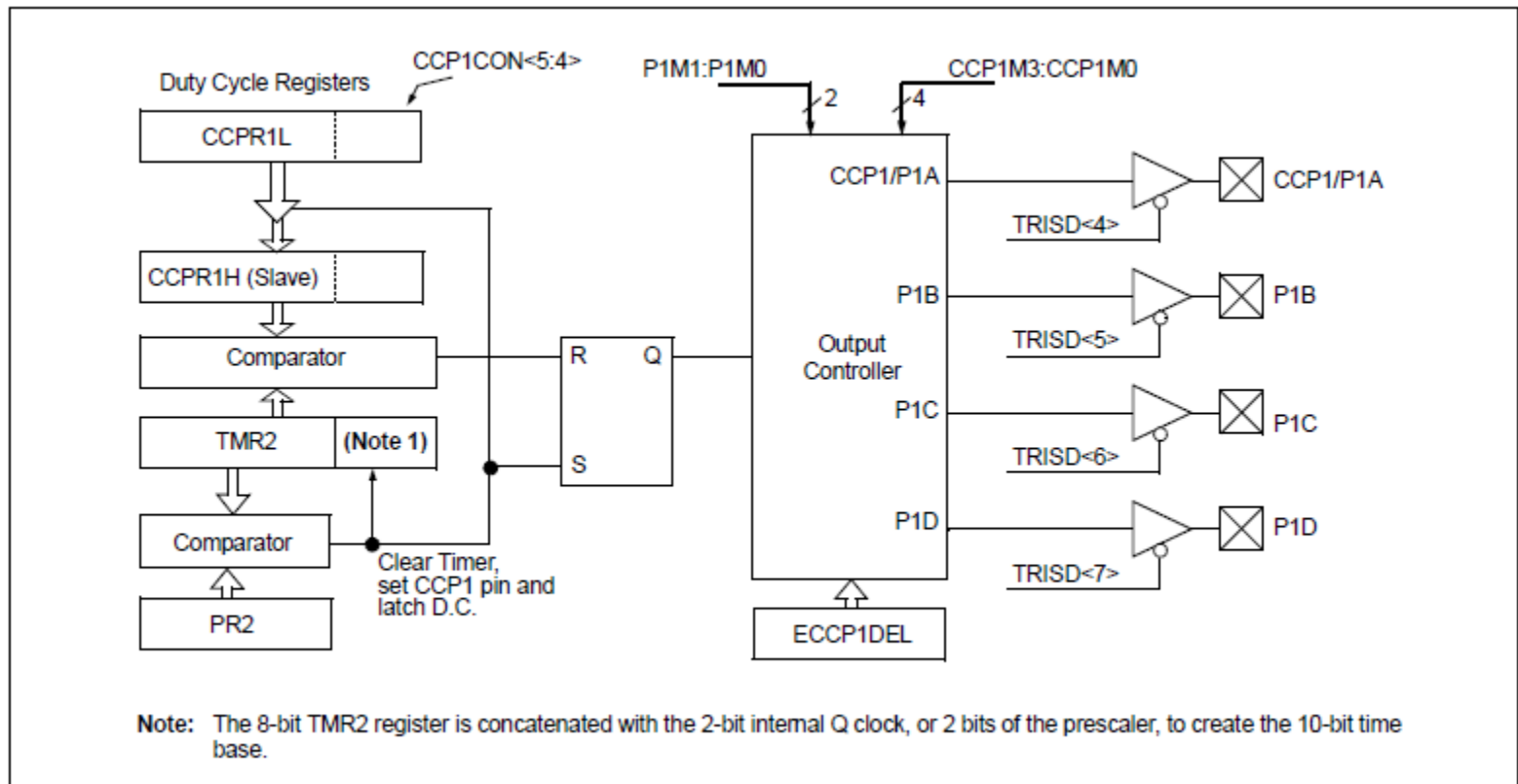
1. Configurar timer 0 como contador
2. Configurar timer 1 como reloj

PWM

FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



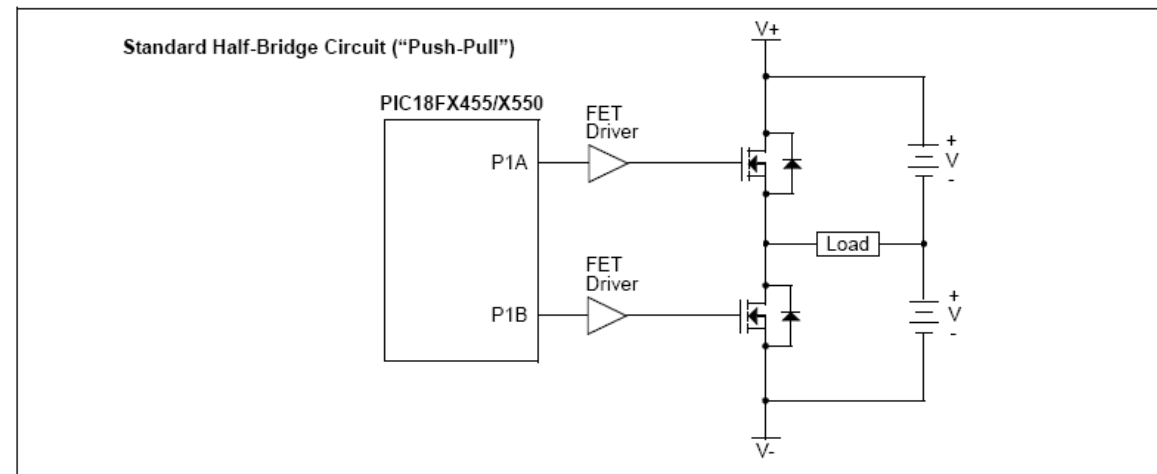
Enhanced PWM



NOTA: The ECCP module is implemented only in 40/44-pin devices.

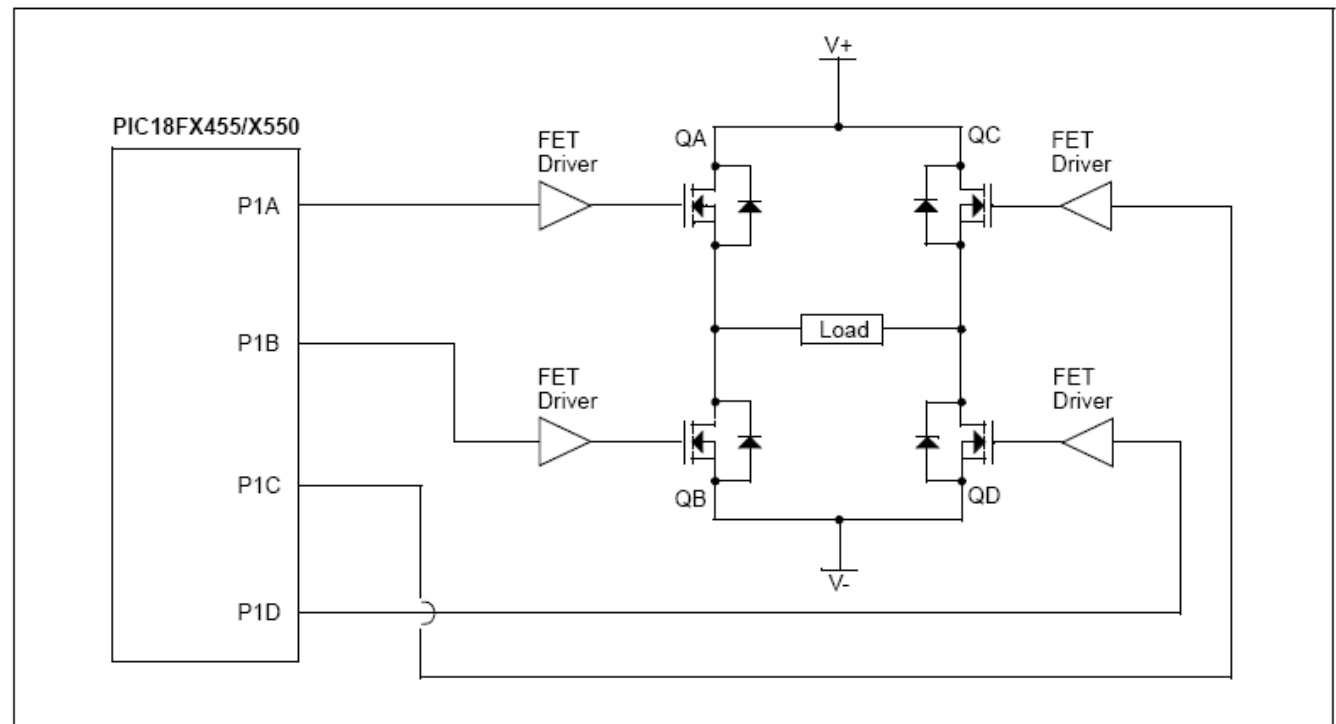
PWM

FIGURE 16-5: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS



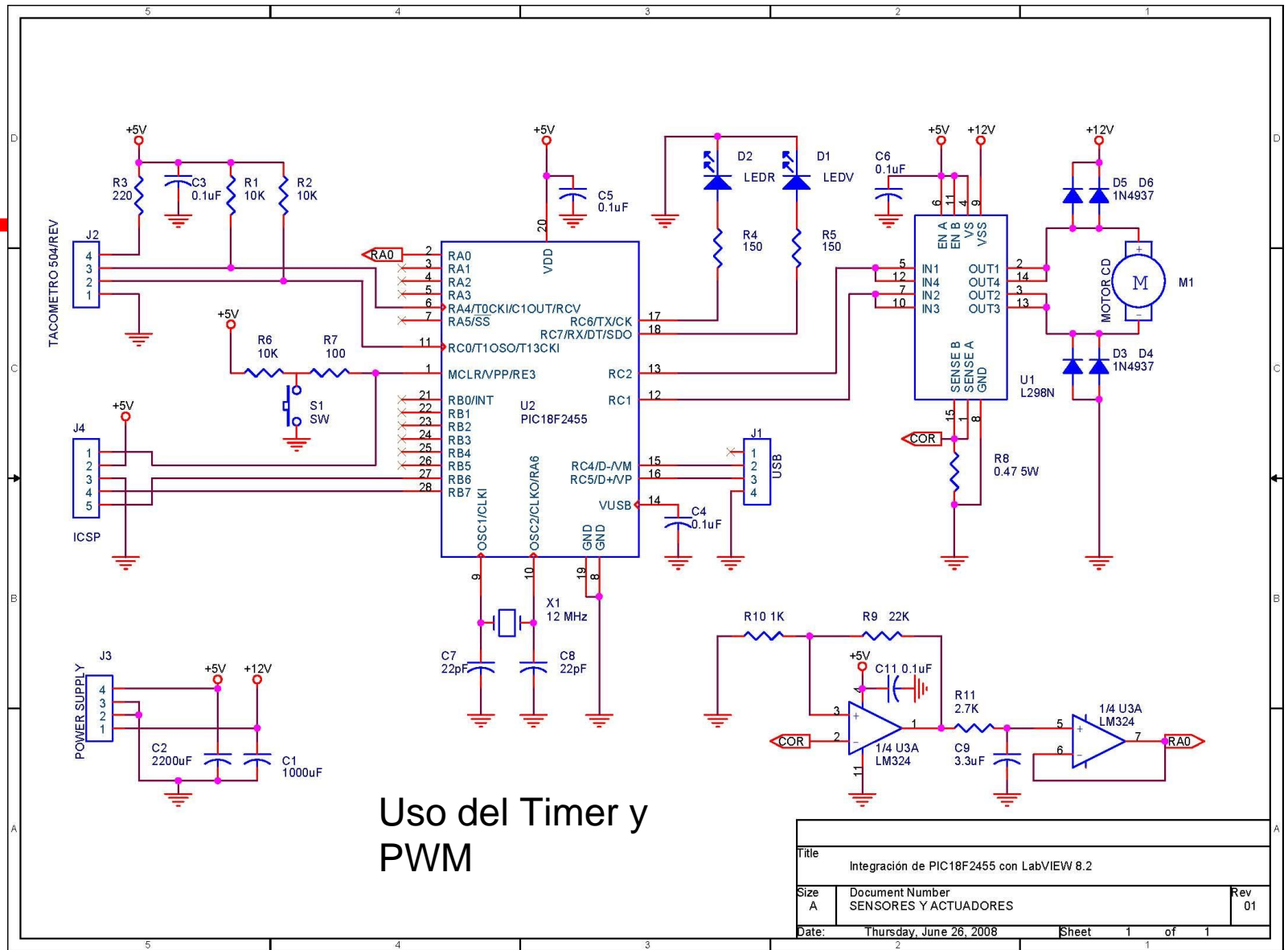
PWM

FIGURE 16-7: EXAMPLE OF FULL-BRIDGE APPLICATION



PWM

Uso del módulo del pwm del PIC (pwm.c)



Puerto Serie

Uso del puerto serie del PIC (serie.c)

Resumen

Aplica para Timer 0,1 y 3:

¿Cómo determino el modo de trabajo del timer?

```
Setup_timer0(mode);
```

```
Setup_timer0(RTCC_DIV_2 | RTCC_EXT_L_TO_H);
```

¿Cómo determino el tiempo de interrupción?

```
Set_timer0(val) = 256 - ((tiempo_int * val_xtal) / (4 * preescaler))
```

```
Tiempo_int = ( 4*(256-set_timer0(val))* preescaler)/val_xtal
```

Preescaler = al valor (en lugar de la "x") que sigue al RTCC_DIV_x

¿Cómo ajusto el nuevo valor del registro del timer?

```
Set_timer0(value);
```

Nota: Para el modo de 16 bits sustituya el valor de 256 por 65536

Resumen

Aplica para PWM (CCP Module):

```
Setup_ccp1(CCP_PWM);
```

```
Setup_timer_2 (mode, period, postscale);
```

```
Setup_timer_2 (T2_DIV_BY_4, 127, 1);
```

$\text{Frequency} = \text{Fosc} / (4 * \text{mode} * (\text{period} + 1) * \text{postscale})$

De donde mode = al valor (en lugar de la "x") que sigue al T2_DIV_BY_x

```
set_pwm1_duty(valor_pwm1);
```

NOTA: El valor del duty cycle del PWM no siempre empieza con el valor de 0 y termina con el valor máximo 16838

Resumen

¿Qué modo de PWM se desea?

// Modo alternativo para configurar el PWM (sustituye la instruccion: setup_power_pwm_pins())

PWMCON0= 0x0040;

//setup_power_pwm_pins(PWM_COMPLEMENTARY,PWM_COMPLEMENTARY,PWM_COMPLEMENTARY,PWM_OFF);

//PPWM los modulos 0, 1 y 2 se establecen como PWM complementarios

¿Cómo determino la frecuencia de trabajo del PWM?

// Inicializando el power PWM

period = 399;

//Frequency = $F_{osc} / (4 * mode * (period+1) * postscale) = 4000000 / 4 * 4 * (399+1) * 1 = 625 \text{ Hz}$ (igual que en PWM (CCP Module)

//De donde mode = al valor que sigue a WM_CLOCK_DIV_

Resumen

¿Cómo determino el tiempo muerto?

//tiempo muerto en uS = (dead_time * PWM_DEAD_CLOCK_DIV_X)/val_xtal= (15*4)/4000000= 15 uS

// ... de donde dead time es un valor entero entre 0 y 15

//dead time for reactive loads = 15 microSeg

SUSTITUYENDO VALORES ...

```
setup_power_pwm(PWM_CLOCK_DIV_4|PWM_FREE_RUN|PWM_DEAD_CLOCK_DIV_4,1,0,period,0,1,15);
```

AJUSTANDO EL DUTY CYCLE

```
set_power_pwmX_duty(Duty1);
```

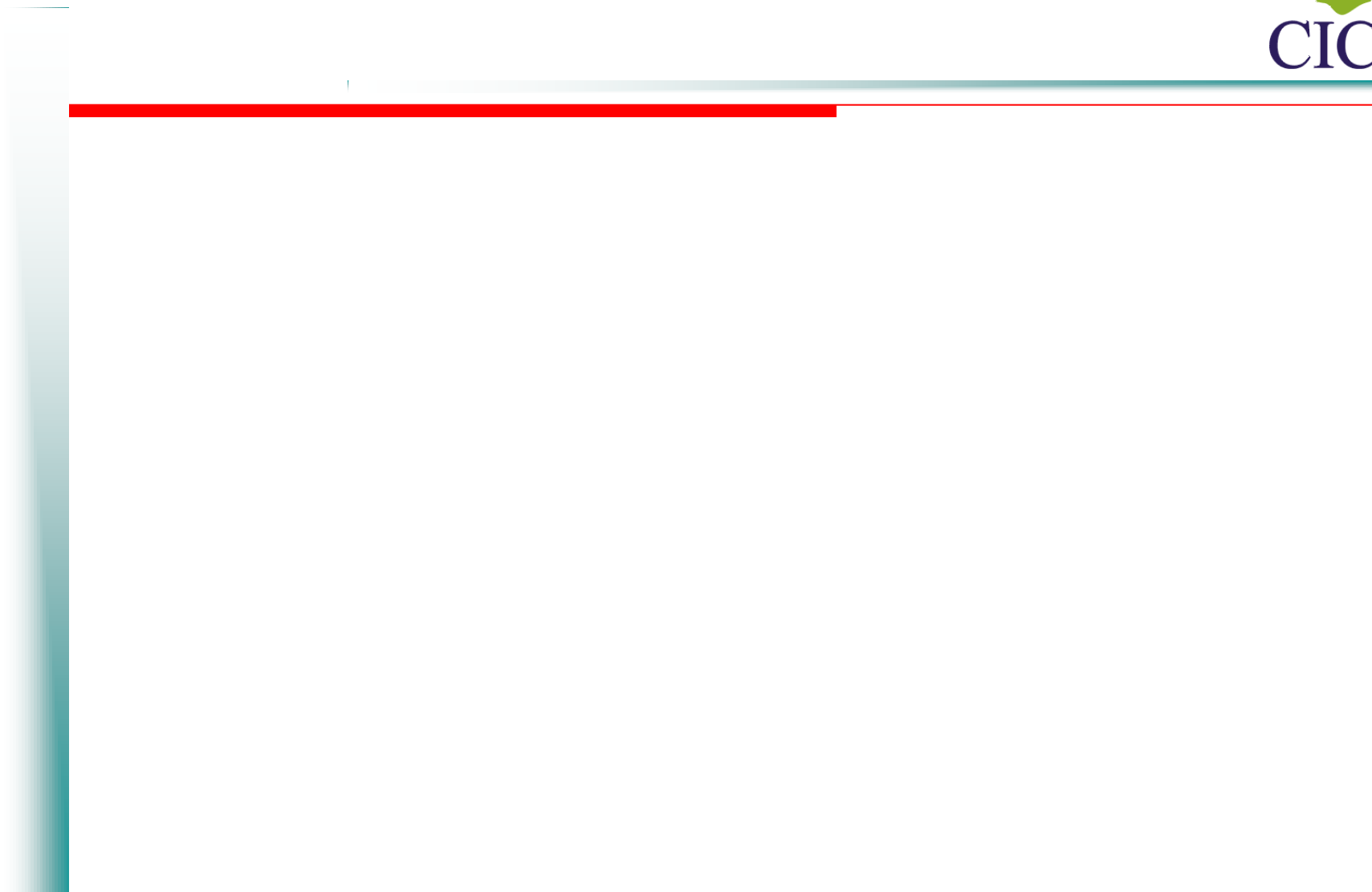
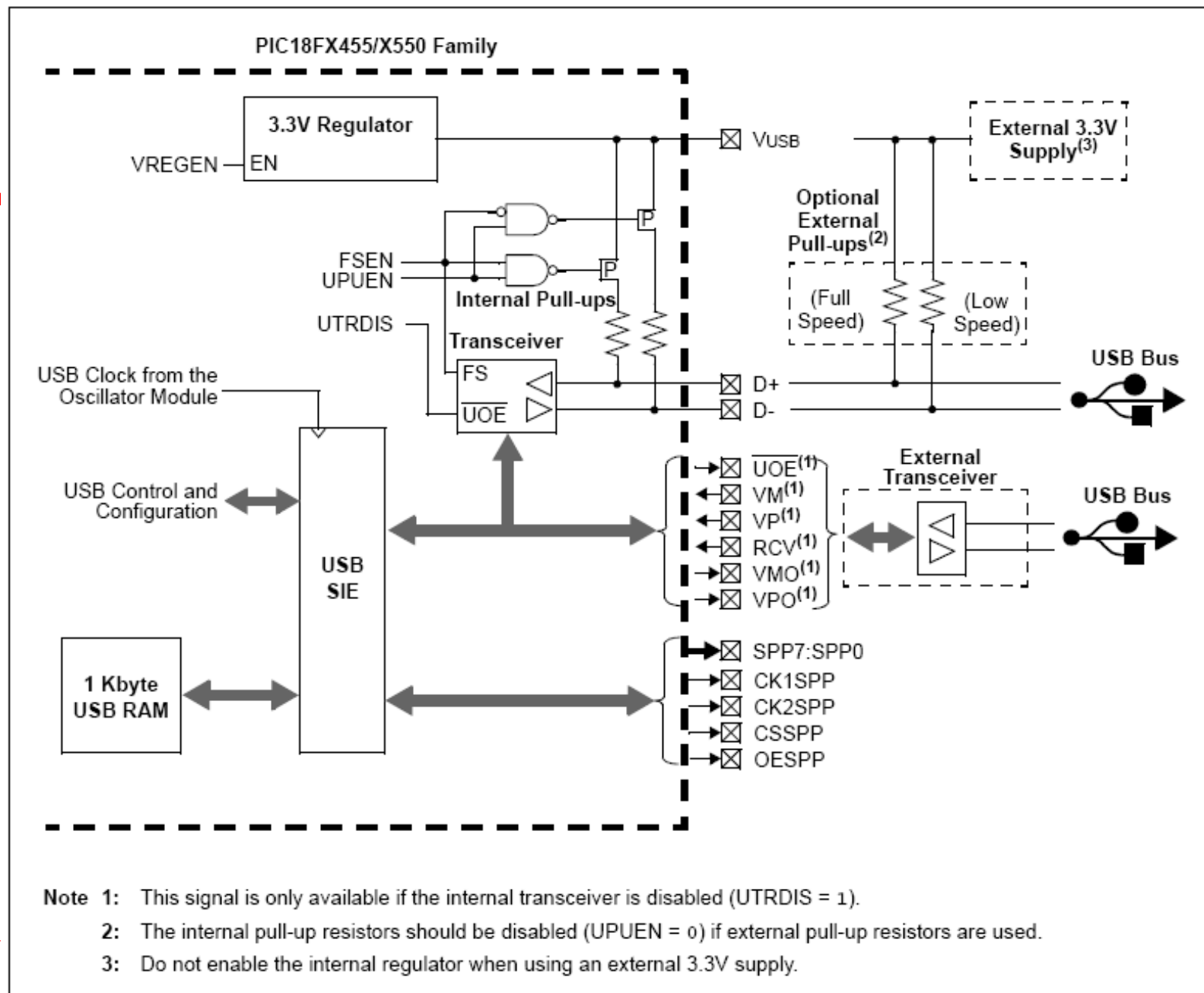
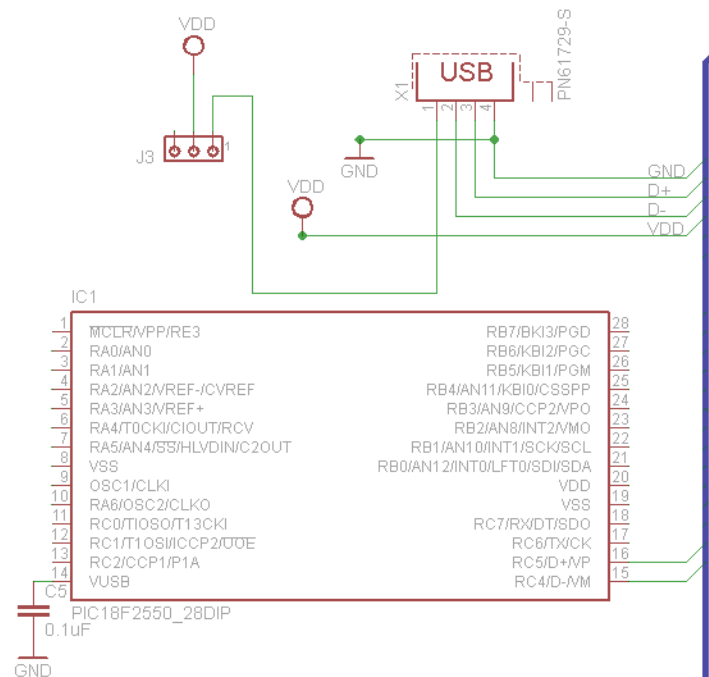


FIGURE 17-1: USB PERIPHERAL AND OPTIONS



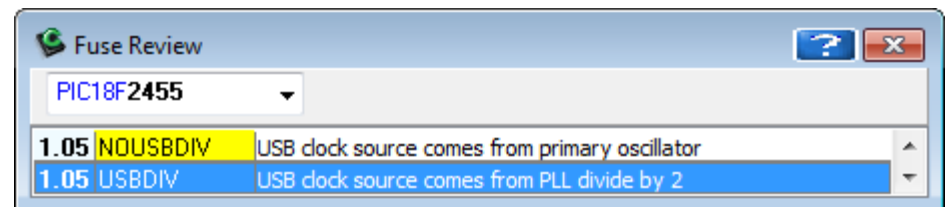
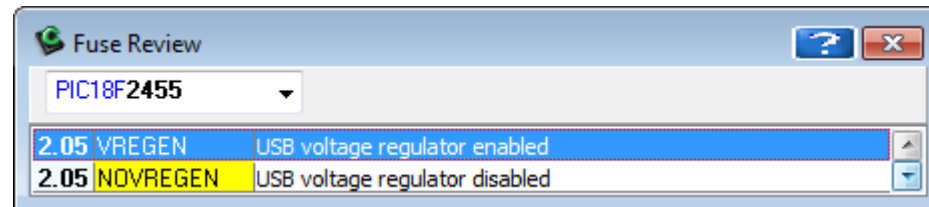
USB



Ver componentes >>

USB

```
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL2, CPUDIV1, VREGEN, MCLR, NOPBADEN
```



USB

DAQ.c

usb_desc_hid.h

```
45  #DEFINE __USB_DESCRIPTOR__
46
47  #ifndef USB_CONFIG_PID
48      #define USB_CONFIG_PID 0x003F
49  #endif
50
51  #ifndef USB_CONFIG_VID
52      #define USB_CONFIG_VID 0x04D8
53  #endif
54
55  #ifndef USB_CONFIG_BUS_POWER
56      //valid range is 0..500
57      #define USB_CONFIG_BUS_POWER 100 //
58  #endif
59
60  #ifndef USB_CONFIG_VERSION
61      //version number that is stored into c
62      //range is 00.00 to 99.99
63      #define USB_CONFIG_VERSION 0x0100
64  #endif
65
66  #ifndef USB_CONFIG_HID_TX_SIZE
67      //valid range is 0-255
68      #define USB_CONFIG_HID_TX_SIZE 64
69  #endif
70
71  #ifndef USB_CONFIG_HID_RX_SIZE
72      //valid range is 0-255
73      #define USB_CONFIG_HID_RX_SIZE 64
74  #endif
```


USB

```
#include <pic18_usb.h>           //Microchip PIC18Fxx5x Hardware layer for CCS's PIC USB driver
#include <usb_desc_scope.h>       //descriptors del Pic USB
#include <usb.c>                  //handles usb setup tokens and get descriptor reports

usb_init();                      // inicializamos el USB
usb_task();                      // habilita periférico usb e interrupciones
usb_wait_for_enumeration();      // esperamos hasta que el PicUSB sea configurado por el host

if(usb_enumerated())             // si el Pic está configurado via USB
{
    if (usb_kbhit(1))             // si el endpoint de salida contiene datos del host
    {
        usb_get_packet(1, dato, 64); // cojemos el paquete de tamaño 64bytes del EP1 y almacenamos en dato

        usb_put_packet(1, dato, 64, USB_DTS_TOGGLE); //y enviamos el mismo paquete de tamaño 64bytes del EP1 al PC
    }
}
```

USB

Notas:

A diferencia del Host (la PC), el PIC puede saber cuando ya le llegaron datos nuevos procedentes del puerto USB

El Device (el PIC) no puede enviar datos al Host hasta que las pipes estén configuradas, por lo que no se recomienda enviar datos sin que el Host se los solicite.

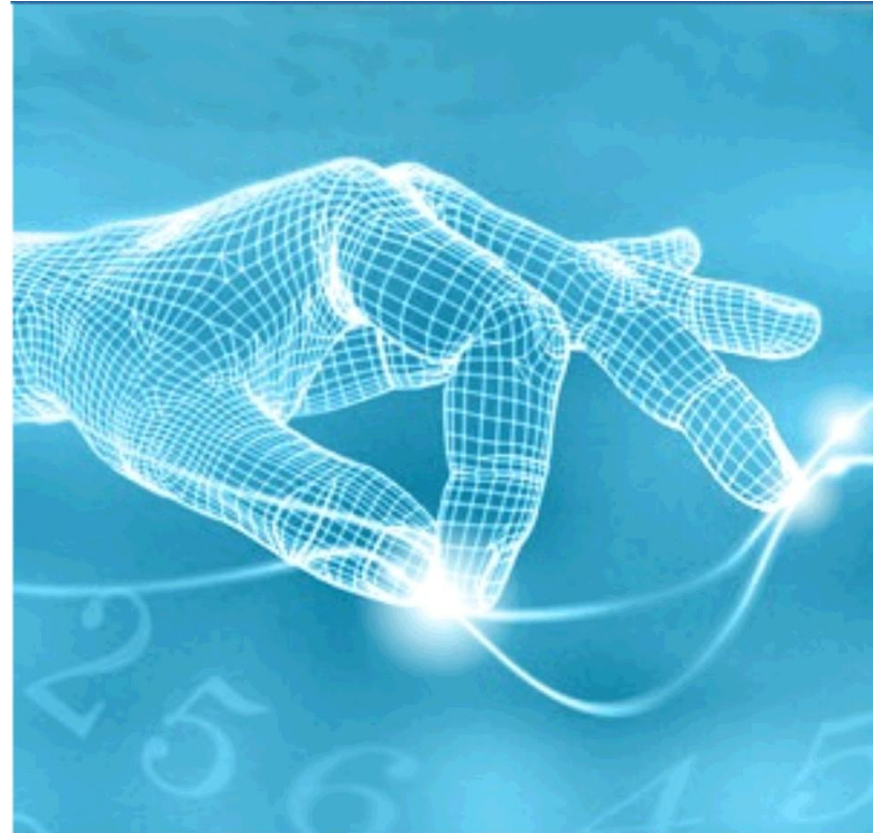
Tema 3

Introducción al USB

Los desarrolladores del USB

El grupo inicial se formó por:

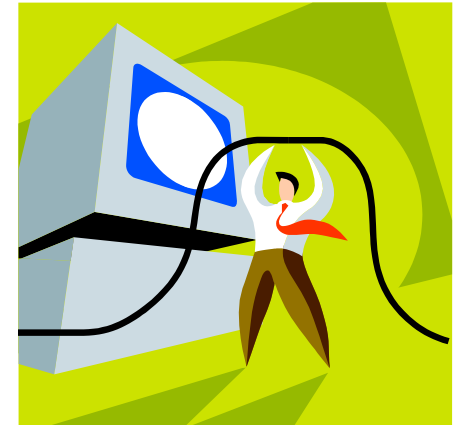
- 1. Compaq,**
- 2. Intel,**
- 3. IBM,**
- 4. Microsoft,**
- 5. NEC,**
- 6. Northern Telecom y**
- 7. Digital Equipment.**



Introducción al USB

ES UNA INTERFAZ DE 4 HILOS...

Pin	Nombre	Color del cable	Descripción
1	VCC	Rojo	+5v
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Tierra



La máxima longitud del cable es de 5 m (AWG20) o 80 cm (AWG28).

D+ y D- son de par cruzado.

La cubierta exterior es cobre trenzado con un escudo de aluminio.

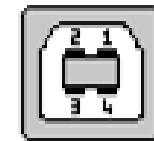
Introducción al USB

Tengo un cable USB...

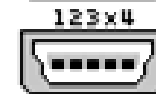
...¿Cuál es el pin 1?



A



B



MINI

4 pin USB A / USB B / mini-USB jack connector



A



B



MINI

4 pin USB A or USB B plug connector

LA ESPECIFICACIÓN USB Y SU VELOCIDAD DE TRANSFERENCIA



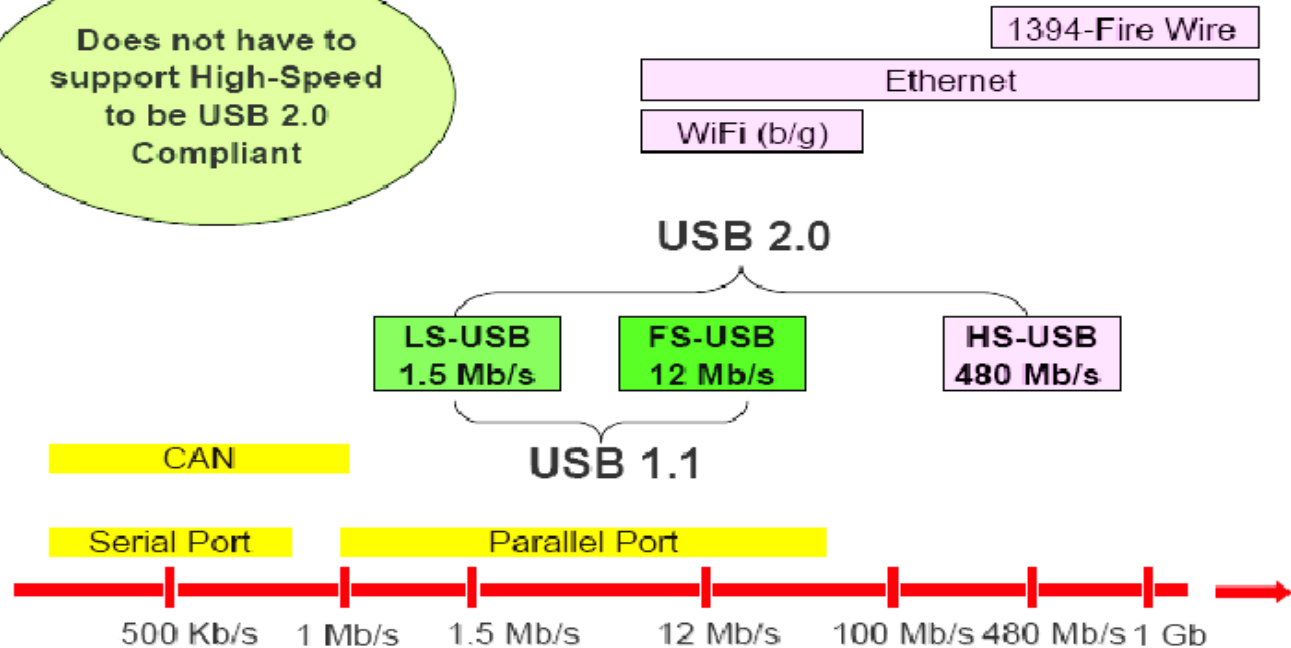
USB				
	LOW SPEED 1.5 Mbps	FULL SPEED 12Mbps	HIGH SPEED 480Mbps	SUPER SPEED 4.8Gbps
1.0	■			
1.1	■	■		
2.0	■	■	■	
3.0	■	■	■	■

USB vs OTROS BUSES



Buses and Speed Comparison

Does not have to support High-Speed to be USB 2.0 Compliant



Introducción al USB

Otras características relevantes son:

**Es posible conectar hasta 127 dispositivos
en un único Host.**



Introducción al USB

**Detecta y configura automáticamente
sin reiniciar la PC...**



**... y para una gran
variedad de dispositivos
no es necesario instalar
software adicional !**

Introducción al USB

No hay conflictos de IRQ'S



Introducción al USB

Puede funcionar para una gran variedad de periféricos.



Desventajas del USB

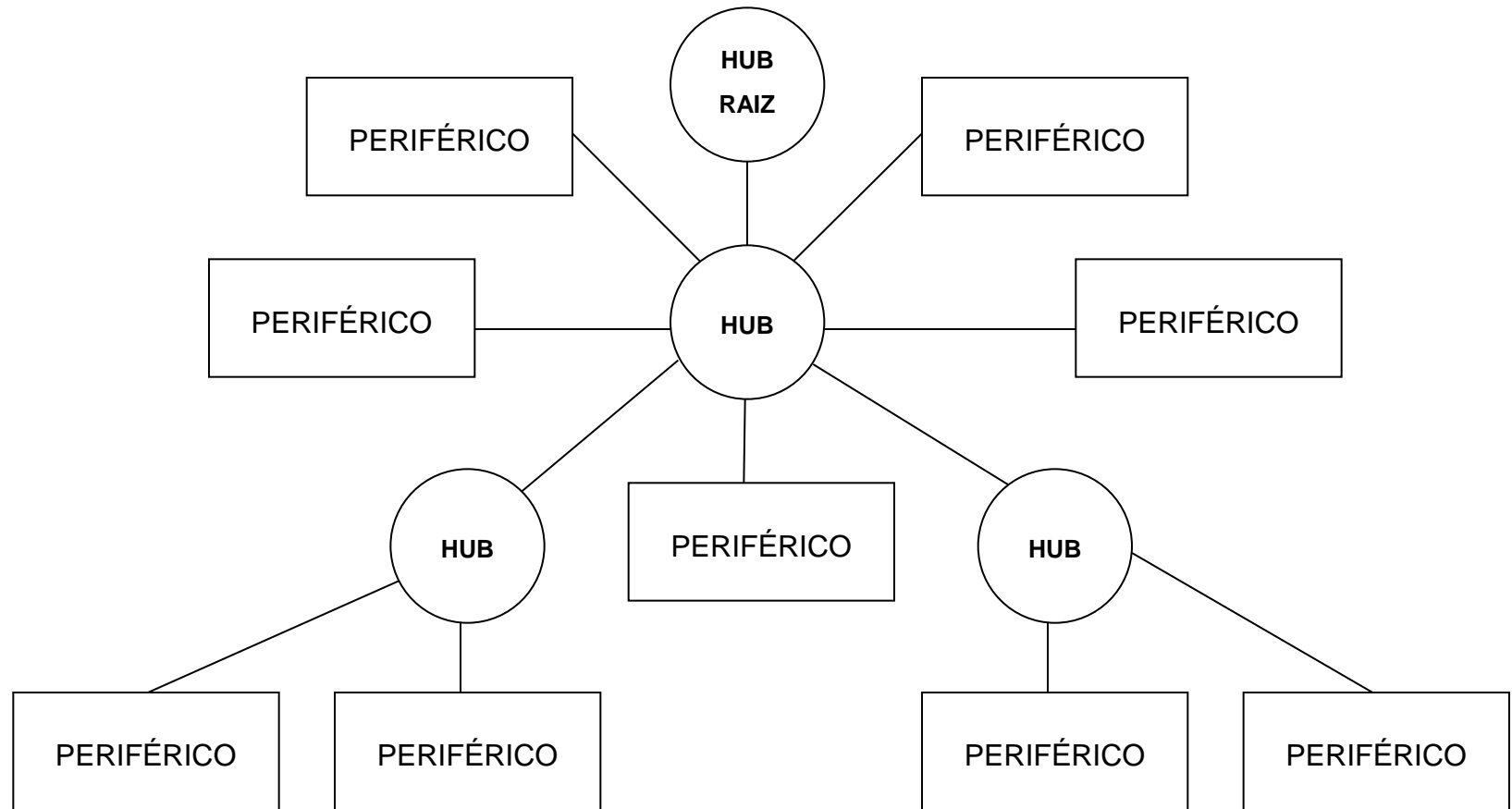
El ancho de banda debe repartirse entre todos los dispositivos conectados a él.



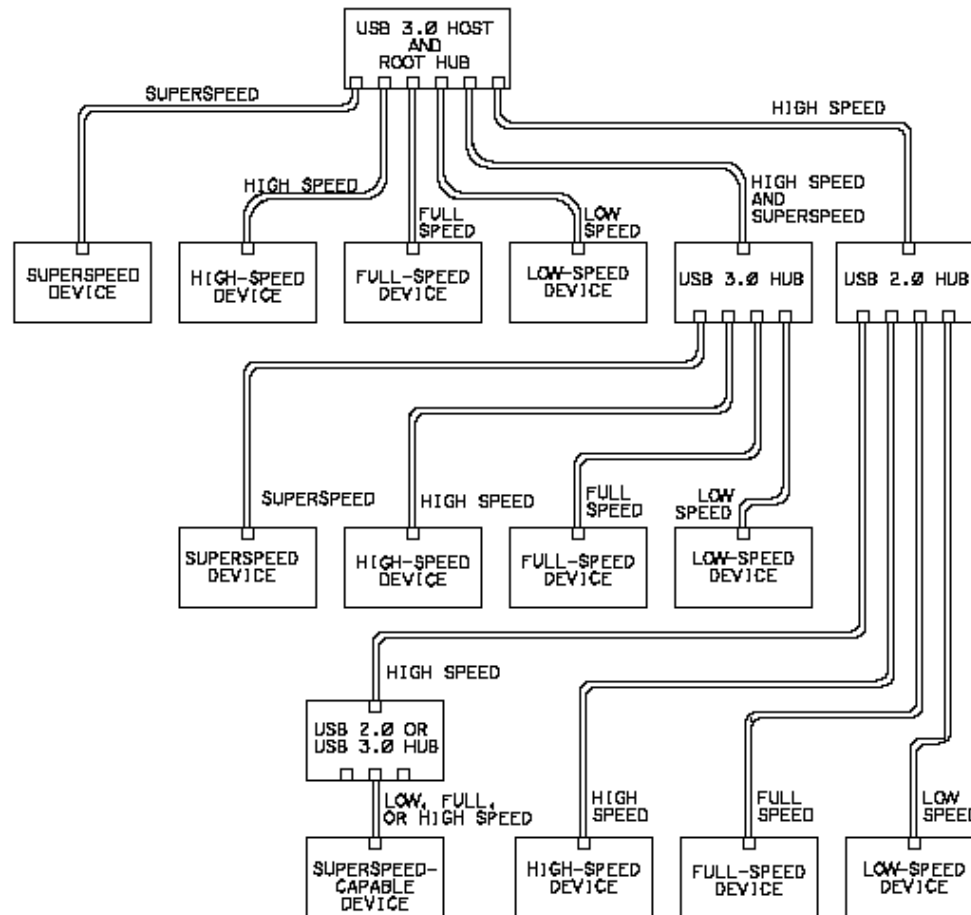
No es posible enviar un mensaje simultáneamente a múltiples dispositivos USB.

Topología

Topología del bus



Topología del bus USB considerando dispositivos V. 1.0, 1.1, 2.0 y 3.0



Máxima longitud entre un dispositivo y el Host

Un segmento de cable puede tener una longitud máxima de 5 metros.



Se puede incrementar la longitud de un enlace USB hasta 30 metros usando cables, 5 hubs y un dispositivo.



#Detalles

Tipos de transferencias



•Control



•Interrupt



•Masiva
“Bulk”



•Isócrona

Una transferencia para cada aplicación

Tipo de Transferencia	Uso
CONTROL	Utilizada por todos los dispositivos durante la enumeración. También para darle órdenes personalizadas al dispositivo.
INTERRUPCIÓN	Útil cuando los datos deben transmitirse sin demora. Garantiza la más rápida respuesta del Host. Una sola transferencia x marco.
MÁSIVA	Útil para transferir datos cuando el tiempo no es crítico. Se pueden transferir grandes cantidades de datos sin atascar el bus. Con un bus inactivo es el tipo más veloz de transferencia.
ISÓCRONA	Útil cuando los datos deben llegar a un ritmo constante y los errores ocasionales son tolerados. Una sola transferencia por marco.

Protocolo de transmisión

El *Host* administra el flujo de datos en el bus.



**Múltiples periféricos
podrían querer transferir
datos al mismo tiempo...**

**...es así como el *Host*
divide el tiempo disponible
en segmentos llamados
“frames” o “microframes”.**

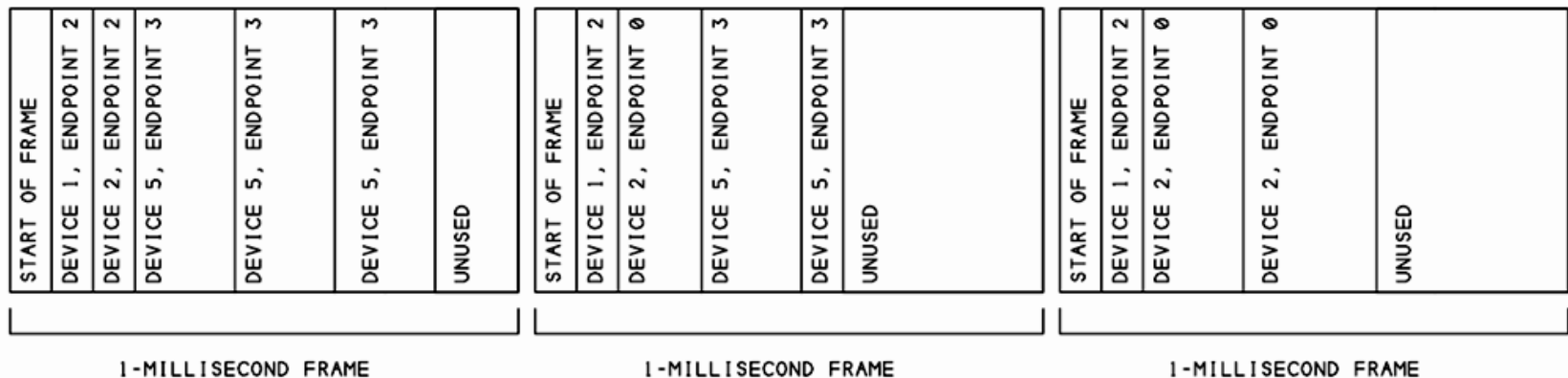
División del tiempo en marcos (*frames*).



A velocidades *Low* y *Full*, el *Host* programa las transacciones con los dispositivos dentro de marcos de 1 ms.

El *Host* puede programar transacciones en cualquier lugar dentro de un marco.

A velocidad *High* los marcos son de 125 μ s.





Introducción al USB



El voltaje nominal de VBUS es +5V.

- El valor real puede ser un poco más o un poco menos.
- Los voltajes mínimo y máximo permitidos (usb 2.0) son:

Hub Type	Minimum Voltage	Maximum Voltage
High Power	4.75	5.25
Low Power	4.4	5.25

- Adicionalmente las condiciones de uso pueden causar que el voltaje sea tan bajo como 4.07 V



Administración de la Energía



- Un dispositivo que requiera hasta 100 mA puede ser alimentado con la energía de cualquier *Host* o *Hub*.
- Ningún dispositivo podrá utilizar más de 100 mA, antes que el *Host* lo haya configurado para usar más corriente.
- Un dispositivo que requiera hasta 500 mA puede usar la alimentación del bus cuando se conecta a un *Hub* autoalimentado o a cualquier *Host* excepto los *Host* alimentados con batería.

La Tarjeta de Presentación del Dispositivo: El Descriptor.

Device Descriptor

Configuration Descriptor

Interface Descriptor

Class Descriptor

EndPoint #1

⋮

EndPoint #n

Report Descriptor (únicamente HID)

String Descriptor



Device Descriptor

- **Contiene la información básica del dispositivo USB como:**
 - **Especificación de USB (1.0, 1.1 o 2.0)**
 - **Identificación del Vendedor (VID)**
 - **Identificación del Producto (PID)**
 - **Número posible de configuraciones**
 - **Versión del producto**
 - **Número de serie del producto**
 - **Nombre del Vendedor**
 - **Nombre del Producto**



Configuration Descriptor

- Puede haber más de un *Configuration Descriptor*.
- Especifica las características y habilidades del dispositivo USB como:
 - Especifica si el dispositivo es autoalimentado o requiere energía del *Host*.
 - Cantidad de energía que utiliza el dispositivo.
 - Contiene al *Interface Descriptor*.



Interface Descriptor

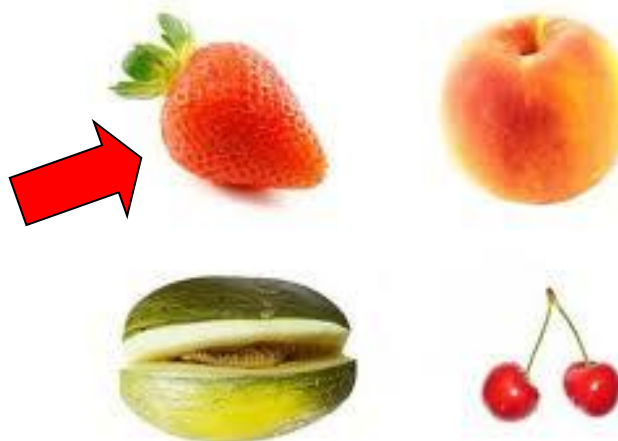
- **Contiene los siguientes datos:**
 - **Identifica con un número a la interface.**
 - **Contiene al *Class Descriptor*.**
 - **Contiene al (o los) *EndPoint Descriptor*.**



Class Descriptor

Contiene los siguientes datos:

- **Define la clase del dispositivo (HID, Impresora, Hub, Almacenamiento Masivo, etc).**
- **Clave del idioma que maneja el dispositivo.**



EndPoint Descriptor



Contiene los siguientes datos:

- El número de id. del *EndPoint*.
 - Define si el *EndPoint* es de Entrada o Salida.
 - Este parámetro se considera visto desde la perspectiva del *Host*.
 - Es de Salida cuando el *EndPoint* recibirá datos enviados desde el *Host*.
 - Es de Entrada cuando el *EndPoint* transmitirá datos hacia el *Host*.
 - Define el tamaño máximo del paquete a transmitir a través del *EndPoint*.
 - El tipo de transferencia que usará el *EndPoint*.
 - La latencia máxima de transmisión.
-

Report Descriptor (HID)

- **Contiene los siguientes datos:**
 - Define la subclase de dispositivo HID (Teclado, Mouse, HID Genérico, etc).
 - Define la cantidad de bytes de entrada y de salida así como el significado de cada byte, que el dispositivo utiliza en su comunicación con el *Host*.
 - Es completamente personalizable.



String Descriptor

Contiene las cadenas de texto a ser mostradas por el sistema operativo al conectar el dispositivo.

- **Nombre del Vendedor.**
- **Nombre del Producto.**
- **Clave que identifica el idioma en que están escritas las cadenas de texto.**





EndPoint

- Un *EndPoint* es un buffer que almacena múltiples bytes. Es un bloque de memoria o registro dentro del dispositivo.



- Un *EndPoint* se define con un número del 0 al 15 , un sentido de transmisión IN o OUT.

Pipes

Una *Pipe* es una asociación entre el EndPoint del dispositivo y el software del *Host* Controlador.

El *Host* debe establecer una *Pipe* con cada EndPoint con el que quiera comunicarse.



La Enumeración

- El *Host* la realiza cuando se conecta o se desconecta un dispositivo USB del bus.
- Es un intercambio de información cuyo propósito es aprender del dispositivo y asignarle un “device driver”.
- El proceso incluye asignarle una dirección al dispositivo, seleccionar una configuración que especifique: **requerimientos de energía, número y tipo de endpoints, clase del producto, tipo de transferencia y otras características.**



El Proceso de Enumeración

1. El usuario conecta un dispositivo al puerto USB.
2. El *Hub* detecta al dispositivo.
3. El *Host* aprende del nuevo dispositivo.
4. El *Hub* detecta si un dispositivo es *low* o *full speed*.
5. El *Hub* reinicia el dispositivo.
6. El *Host* aprende si un dispositivo es *full* o *high speed*.
7. El *Hub* establece una ruta de señal entre el dispositivo y el bus.

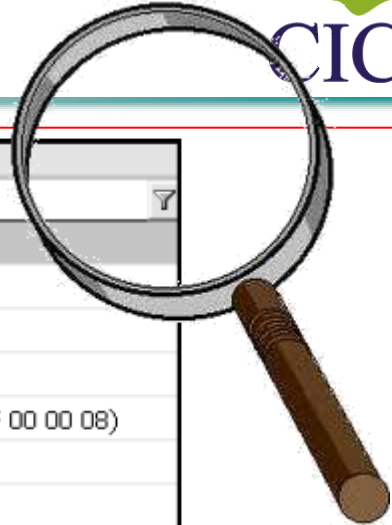






















El Proceso de Enumeración

8. El *Host* envía un comando de requerimiento para aprender el máximo tamaño del paquete de la ruta por defecto.
9. El *Host* asigna una dirección al dispositivo.
10. El *Host* aprende acerca de las habilidades del dispositivo.
11. El *Host* asigna y carga un *Driver* para el dispositivo.
12. El *Driver* del dispositivo elige una configuración y configura al dispositivo.



La Enumeración bajo la lupa



Item	Device	Payload
 Reset (2.3 s)		
 Suspended (114.0 ms)		
 Reset (10.0 ms)		
 High speed Detection Handshake		
 + GetDescriptor (Device)	0 (5)	8 bytes (12 01 00 02 FF 00 00 08)
 Reset (10.0 ms)		
 High speed Detection Handshake		
 + SetAddress (5)	0 (5)	No data
 + GetDescriptor (Device)	5	18 bytes (12 01 00 02 FF 00 00 08 ...)
 + GetDescriptor (Configuration)	5	9 bytes (09 02 2E 00 01 01 00 A0 32)
 + GetDescriptor (Configuration)	5	46 bytes (09 02 2E 00 01 01 00 A0 ...)
 + GetDescriptor (String lang IDs)	5	4 bytes (04 03 09 04)
 + GetDescriptor (String iProduct)	5	24 bytes (18 03 57 00 69 00 6E 00 ...)
 + GetDescriptor (String lang IDs)	5	4 bytes (04 03 09 04)
 + GetDescriptor (String iProduct)	5	24 bytes (18 03 57 00 69 00 6E 00 ...)
 + GetDescriptor (Device)	5	18 bytes (12 01 00 02 FF 00 00 08 ...)
 + GetDescriptor (Configuration)	5	9 bytes (09 02 2E 00 01 01 00 A0 32)
 + GetDescriptor (Configuration)	5	46 bytes (09 02 2E 00 01 01 00 A0 ...)
 + GetStatus (Device)	5	2 bytes (00 00)
 + SetConfiguration (1)	5	No data

Preguntas ?

Gracias por su atención!!!