

1AEL0256 - Microcontroladores

Semana 2

Semestre 2025-2

Profesor: Kalun José Lau Gan

Agenda:

- La familia PIC18 de Microchip
- El Curiosity Nano PIC18F57Q43
- Manejo de la documentación (hojas técnicas y repositorios)
- El MPLAB X IDE
- Primer ejemplo implementado

Preguntas previas:

- ¿Debo de tener los materiales para esta semana?
 - Esta semana 2 iniciaremos con las experiencias de laboratorio con el PIC18F57Q43 de manera dirigida, es decir, el profesor hará toda la experiencia de implementación. Se recomienda tener los materiales para las experiencias de la semana 3.
- El lenguaje Assembler para el microcontrolador lo debemos de saber previamente o lo vamos a ver en las clases?
 - Se los va a atender durante las sesiones con el compromiso de que el alumno practique.
- ¿Cuáles son los softwares que necesito instalar en mi PC?
 - ~~MPLAB X versión 6.20 (la mas actual)~~
 - ~~XC8 versión 2.50 (la mas actual)~~
 - MPLABX v6.25, XC8 v3.10
- ¿Qué cable se usa para el Curiosity Nano?
 - Cable USB a MicroUSB de datos (versión antigua)
 - Cable USB a USC-C (versión 2024)

Preguntas previas:

- ¿Hay algún libro sobre este nuevo microcontrolador?
 - No
- Al conectar el Curiosity Nano a la PC solo se enciende, no lo detecta. ¿Qué puede ser?
 - El cable USB que has usado no transporta datos por ende no lo reconoce, cambia de cable.
- ¿Cómo puedo saber si mi PC ha detectado correctamente el Curiosity Nano?
 - Al conectar el Curiosity Nano a la PC, si es que es la primera vez, la PC instalará de manera automática los controladores (drivers) y aparecerá como popup una ventana de explorador de archivos de Windows y se visualizará una unidad de disco correspondiente al Curiosity Nano. (Verificado en Windows 10 y 11)
- ¿El microcontrolador se usa en la industria o solamente con fines académicos? ¿Se puede conectar otro microcontrolador al microcontrolador que estamos usando mediante pines?
 - Los microcontroladores los encuentras en casi todos los equipos electrónicos, desde un reloj hasta el equipo electrónico mas avanzado.
 - Es posible conectar dos microcontroladores y establecer un canal de comunicación entre ellos, mediante el uso de comunicación serial en sus diferentes tipos: I2C, SPI ó UART.

Preguntas previas:

- ¿Es necesario tener un osciloscopio en el curso? ¿Alguna referencia de modelos de scopes pequeños?
 - Es opcional y nos servirá para visualizar señales que cambian de forma en el tiempo, como el PWM.



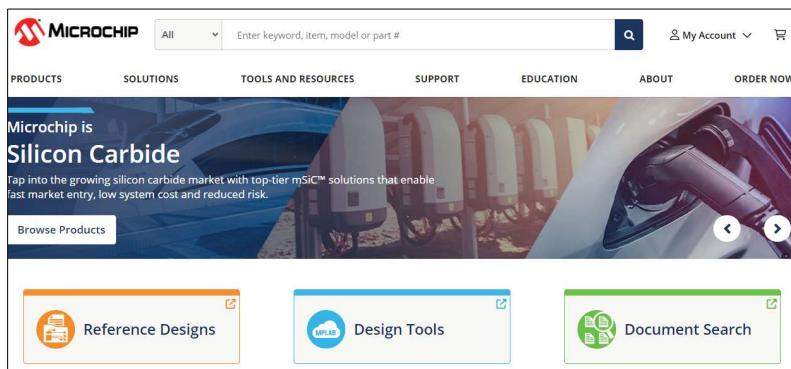
Preguntas previas

- ¿Pueden coexistir diferentes versiones de MPLABX y XC8 en una misma PC?
 - Si, pero de preferencia dejar la versión mas actual solamente a fin de evitar eventuales conflictos.
- Al terminar la instalación del MPLABX me aparece la opción de descargar e instalar el XC8, XC16 y XC32. ¿Cuál de todos selecciono?
 - Solo el XC8
- Los bits de configuración que se han modificado. ¿Serán los mismos para cada proyecto que vayamos a realizar?
 - Si, se configuran los mismos 5 bits de configuración para cada proyecto.

Mercado de los microcontroladores

- Actualmente hay preferencias en usar microcontroladores de 8 bits y de 32 bits:
 - Aplicaciones de baja a mediana escala de complejidad -> 8bits
 - Aplicaciones de mediana a alta escala de complejidad -> 32bits
- Se ha disminuido el uso de microcontroladores de 16 bits, muy posible debido a similitud de costos de manufactura y estrecha relación con los de 32 bits.
- Los precios entre micros de 32bits y 8 bits están muy similares.
- Los µC de 8 bits naturalmente consumen menos energía.

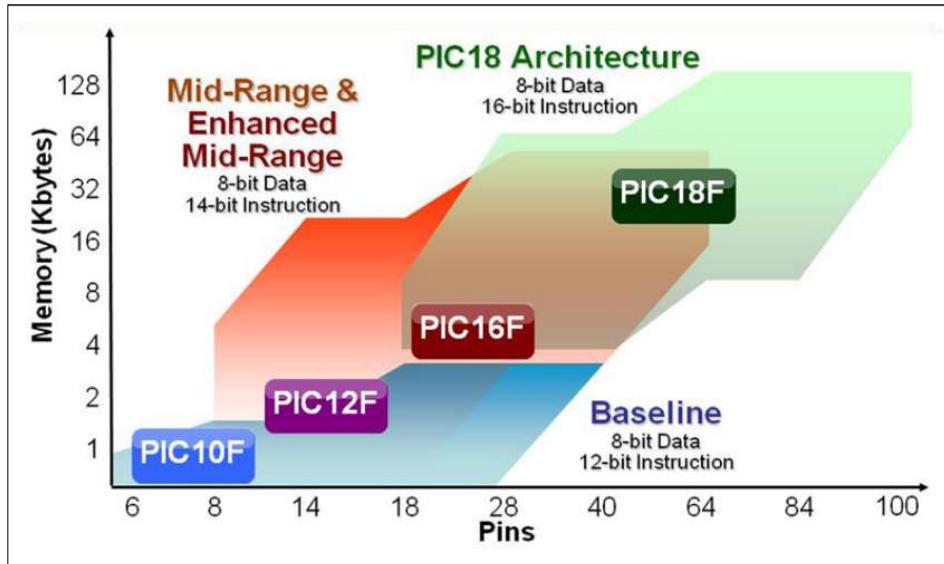
Microchip INC



- División de semiconductores de General Instrument, en 1989 se independizó y se formó Microchip INC.
- Actual líder en el mercado de microcontroladores de **8 bits**.
- En Julio del 2016 compró Atmel y todo su portafolio de productos.
- También adquirió Micrel, Microsemi, KeeLoq, HI-TECH Software, Novocell, ISSC, etc.

Microchip INC

- Familias de microcontroladores PIC de 8 bits



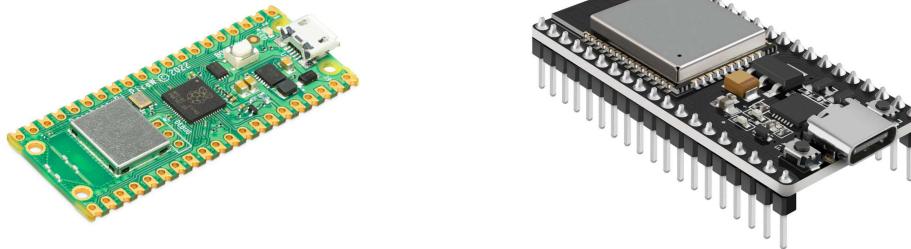
ST Microelectronics

- Líder en microcontroladores de 32bits (STM32)



Nuevos contrincantes en los micros de 32 bits:

- Raspberry Pi Foundation: RP2040 y RP2350
 - Costo muy reducidos
 - Flexibilidad
- Xtensa Tensilica LX7: ESP32
 - Ampliamente utilizado para aplicaciones de IoT



Familia PIC18 de Microchip

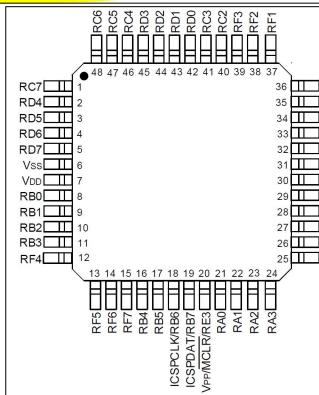
- Primeros dispositivos PIC18: ej. PIC18F452
- Siguiente generación de PIC18: ej. PIC18F4550
- Actualización de generación de PIC18: Familia K ej. PIC18F45K50
- Familias “Q” de PIC18: Q10, Q40/41, Q43, Q71, Q83/84 ej. PIC18F57Q43
- Nuevas familias Q PIC18: Q20, Q24 (released 27/11/2023)

Product Family	Pin Count	Program Flash Memory (kB)	RAM (kB)	Data EEPROM (kB)	8-bit PIC® Microcontrollers										Communications		User Interface		Low Power and System Flexibility	
					Intelligent Analog			Waveform Control ⁽¹⁾			Logic and Math		Safety and Monitoring		Communications		User Interface		Low Power and System Flexibility	
PIC18-Q40/41	14-20	16-64	1-4	512	12 ⁽²⁾	✓	8	✓ ⁽³⁾	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PIC18-Q43	28-48	32-128	2-8	1024	12 ⁽²⁾	✓	8	✓	✓	✓	3	✓	✓	✓	✓	✓	✓	✓	✓	✓
PIC18-Q83 ⁽²⁾ /84 ⁽²⁾	28-48	64-128	8-13	1024	12 ⁽²⁾	✓	8	✓	✓	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PIC18-Q10	28-40	16-128	1-3.6	256-1024	10 ⁽²⁾	✓	5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PIC18-Q71	28-48	16-64	1-4	256	12 ⁽²⁾	✓	8 ⁽²⁾ /10	2	✓	✓	✓	3	✓	✓	✓	✓	✓	✓	✓	✓

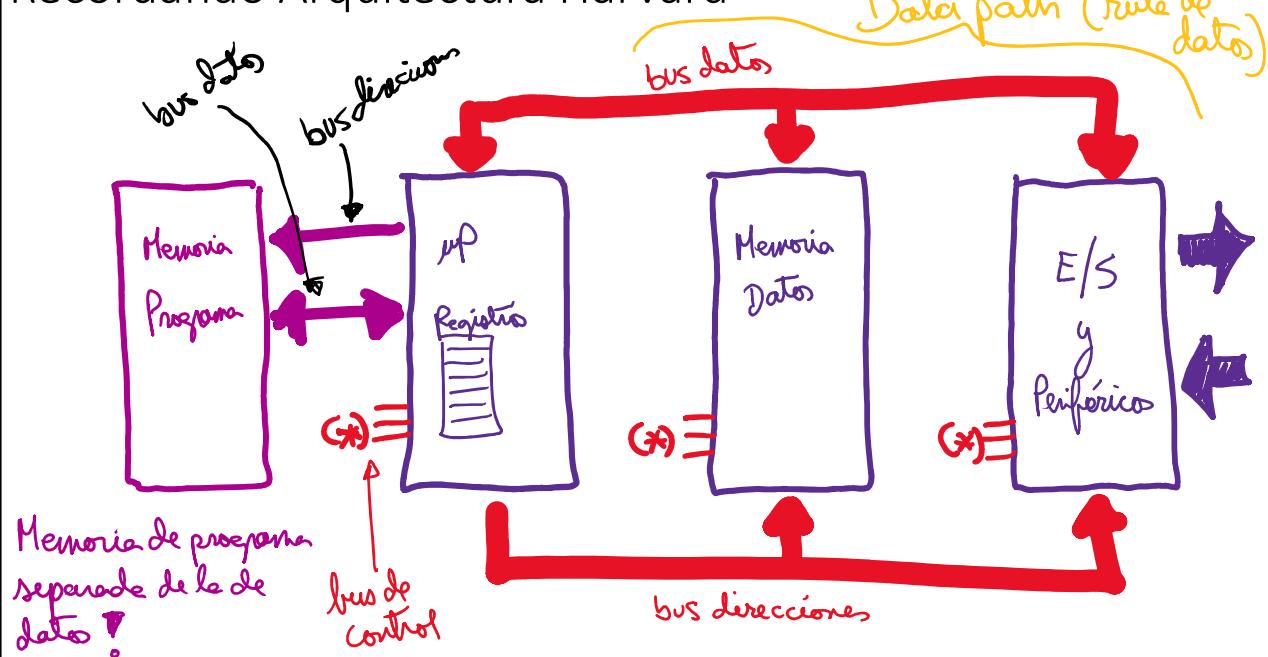
El Microcontrolador PIC18F57Q43

Device	Program Memory Flash (bytes)	Data SRAM (bytes)	Data EEPROM (bytes)	Memory Access Partition/ Device Information Area	I/O Pins/ Peripheral Pin Select	8-Bit Timer with H/T/ 16-Bit Dual PWM/ CCP	Complementary Waveform Generator	Signal Measurement/ Timer Numerically Controlled Oscillator	Configurable Logic Cell	12-Bit ADC/C (channels)	8-Bit DAC	Comparator/ Zero-Cross Detect	High-Low Voltage Detect	SPI/I2C	UART/ with Protocol Support	Direct Memory Access (DMA)	Windowed Watchdog Timer	16-Bit CRC with Scanner	Vectorized Interrupts	Peripheral Module Details	Temperature Indicator	
PIC18F27Q43	128k	8192	1024	Y/Y	25/Y	3/4	3/3	3	1	3	8	24	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y
PIC18F47Q43	128k	8192	1024	Y/Y	36/Y	3/4	3/3	3	1	3	8	35	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y
PIC18F57Q43	128k	8192	1024	Y/Y	44/Y	3/4	3/3	3	1	3	8	43	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y

- Encapsulado de 48 pines
- Amplio rango de voltaje de alimentación: 1.8V – 5.5V DC
- Bajo consumo energético (menor a 800nA en modo SLEEP y alrededor de 1mA en operación con 4MHz, 5V)
- Amplio rango de temperatura: -40°C – +85°C (grado industrial)
- Hasta 64MHz de frecuencia en CPU
- Memoria de programa de 128Kbytes y memoria de datos de 8Kbytes
- Periféricos nuevos (detector de cruce por cero, CLC, etc)
- Interrupciones vectorizadas
- Módulo PPS para asignación de funciones en los pines

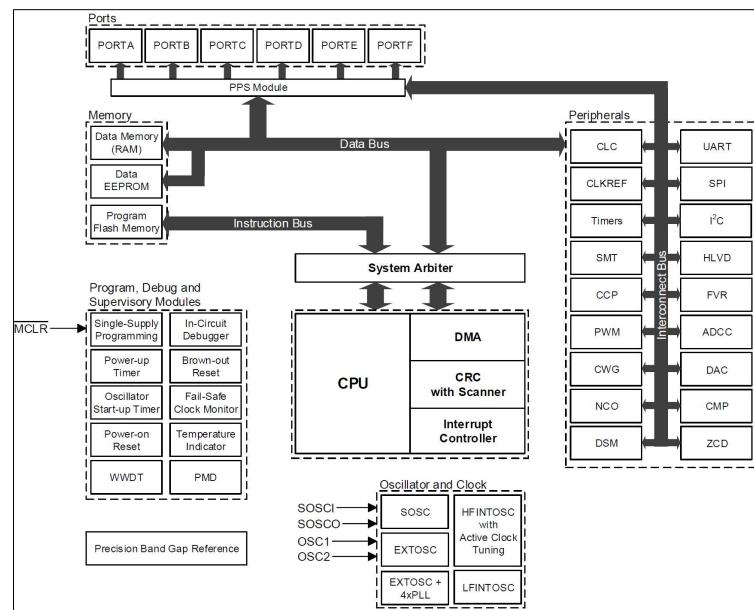


Recordando Arquitectura Harvard



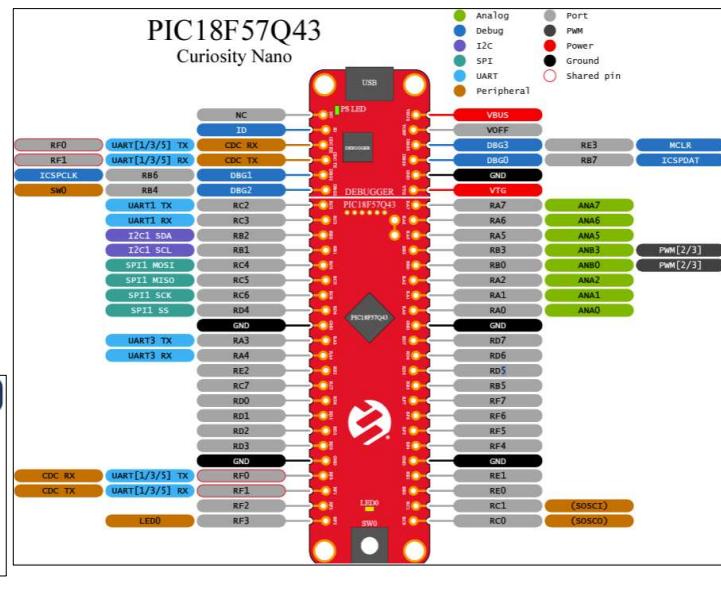
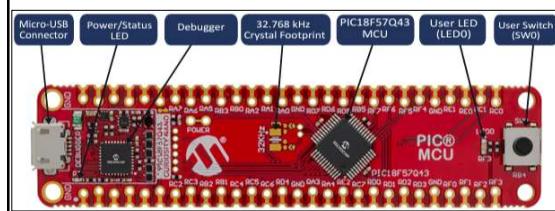
El Microcontrolador PIC18F57Q43

- Arquitectura Harvard:
 - Memoria de programa separada de la memoria de datos y con buses independientes
- Cambio en organización de la memoria de datos
- Módulo Timer0 mejorado
- ADC mejorado de 12bits
- Módulo generador de voltaje de referencia para el ADC



El módulo Curiosity Nano PIC18F57Q43

- Plataforma de desarrollo que integra programador/depurador y microcontrolador PIC18F57Q43.
- Integra un LED (RF3) y un pulsador (RB4), ambos activos en bajo, el pulsador requiere resistencia pull-up.



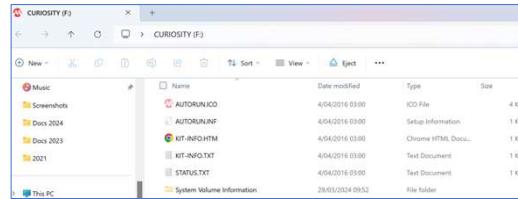
Si es que no se usa el Curiosity Nano

- Necesitamos un microcontrolador en formato DIP para ser usado en el protoboard (ejemplo PIC18F45Q43)
- Necesitamos un programador (ejemplo PICKIT5)



Funciones del debugger/programmer del Curiosity Nano PIC18F57Q43

- Proveer energía al módulo (dos líneas)
 - VBUS (conexión directa de la energía del Puerto USB conectado)
 - VTG (conexión de salida de voltaje del sistema ajustable/controlado integrado en el módulo)
- Programar el microcontrolador onboard (el PIC18F57Q43)
- Proveer un puerto serial en la PC para comunicación UART
- Al conectarlo a la PC, aparecerá una unidad de disco llamado “Curiosity”

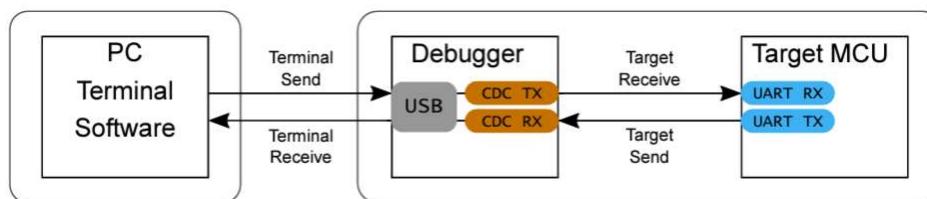


Aclaración sobre la unidad de disco Curiosity

- Al conectar el Curiosity Nano PIC18F57Q43 a la PC aparecerá una unidad de disco llamado "Curiosity".
- Esta unidad de disco **NO DEBE DE USARSE COMO UNIDAD DE ALMACENAMIENTO!, NO ES UNA MEMORIA USB!**
- Esta unidad de disco sirve para hacer drop-off de un firmware

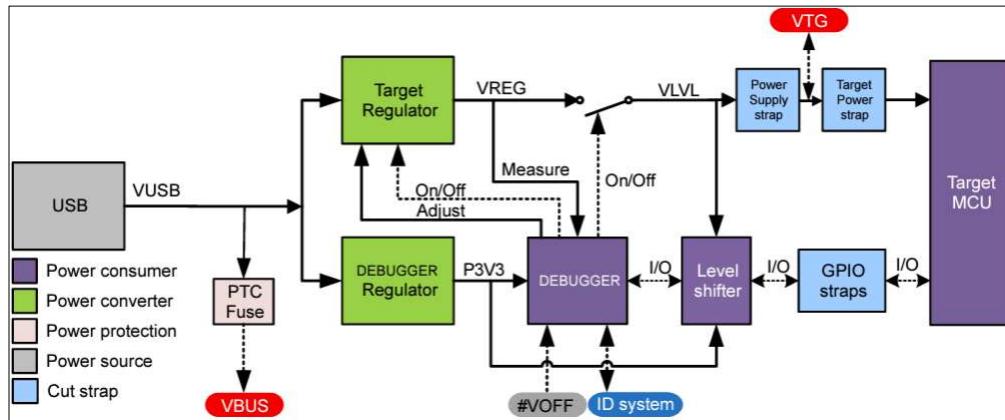
El módulo Curiosity Nano PIC18F57Q43

- Al conectar el Curiosity Nano a la PC con el cable de datos USB-MicroUSB se habilitará un puerto serial para hacer labores de depuración/comunicación
- Dirigirse al Administrador de Dispositivos del Windows (dentro del Panel de Control) para que puedan identificar el puerto serial que se ha habilitado (COMx)
- Software terminal serial puede ser el PuTTY, HyperTerminal, Arduino IDE



El módulo Curiosity Nano PIC18F57Q43

- Posee dos terminales de voltaje de alimentación disponibles para el usuario:
 - VBUS: Conexión directa de la alimentación del puerto USB (5V DC)
 - VTG: Alimentación ajustable y controlable a través del MPLAB X



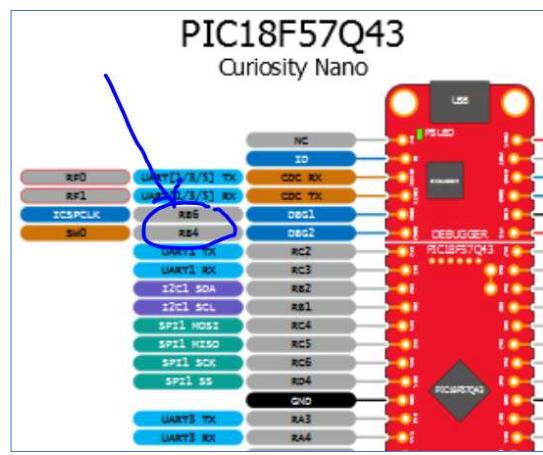
Recomendaciones al usar el Puerto VTG

•NO CONECTAR LOS PINES “VBUS” Y “VTG” A LA VEZ!!!

- El ajuste de voltaje de este puerto se realiza en el MPLABX
- Antes de grabar el dispositivo en el MPLABX, entrar a propiedades del proyecto y activar la energía (opción Power)

El módulo Curiosity Nano PIC18F57Q43

- No encuentro los pines RB4 y RB6 en el Curiosity Nano...
- Los puedes encontrar no rotulados y siguiendo el diagrama de pines en la documentación.



Comentario:

← Akira Sebastian Nakamura Montenegro

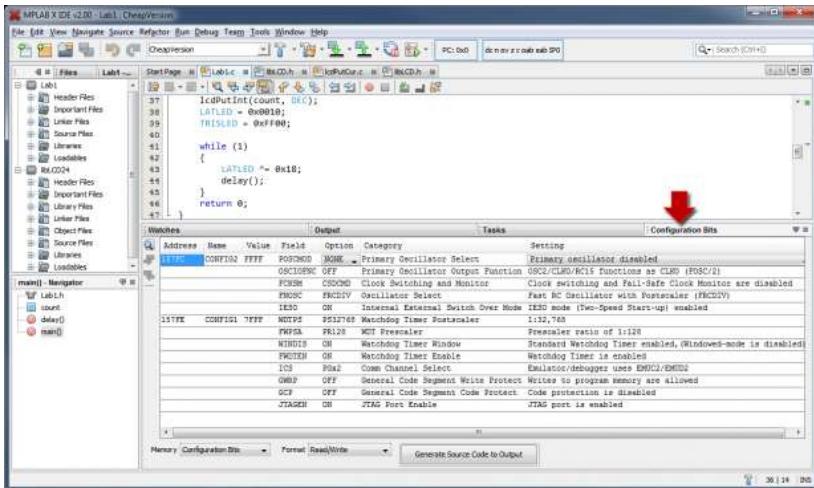
Akira Sebastian Nakamura Monte... 1:06 PM

seguirá siendo importante los procesos de fabricación ahora que la inteligencia artificial planea optimizar los procesos en el mundo computacional, en su conferencia el ceo de nvidia menciona que a su parecer los procesos pasaran a segundo plano en pos de mejores algoritmos

Left: Akira Sebastian Nakamura Montenegro 1:40 PM

- Pero dichos algoritmos. ¿En dónde son ejecutados? En hardware!
- Se tiene que construir el hardware primero!

El MPLAB X IDE



- Descargable desde el siguiente link:

<https://www.microchip.com/mplab/mplab-x-ide>

<https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>

El MPLAB X IDE



- Entorno de desarrollo integrado (Integrated Design Environment - IDE) para los microcontroladores PIC y AVR de Microchip.
- Disponible para Windows, Linux y macOS
- Desarrollo basado en proyecto
- Emplea el compilador XC (XC8, XC16 y XC32), descargable de manera separada.
- Soporte a herramientas de desarrollo propias de Microchip (módulos Curiosity Nano, programadores PICKIT, depuradores ICD, etc)
- Última versión: v6.25 (lanzado el 4 de marzo del 2025)

MPASM vs. PICASM (XC8 Assembler)

- MPASM fué el lenguaje de programación hasta la v5.35 del MPLABX, actualmente en obsolescencia.
 - XC8 PIC Assembler es el nuevo formato de lenguaje y soportado por la nueva versión ~~6.20~~ 6.25
 - Las instrucciones en los microcontroladores no han variado, solo la sintaxis de programación (el core del microcontrolador es el mismo).
 - Evolución: MPLAB -> MPLAB X (32bits) -> MPLAB X (64bits)
 - Documentación: /Program Files/Microchip/xc8/v3.10/docs/MPLAB_XC8_PIC_Assembler_User_Guide.pdf
 - Última versión del XC8: v3.10

Importancia de tener las hojas técnicas de los IC's a usar:

- Las hojas técnicas (datasheet) son proporcionadas por el fabricante del IC's y se detallan todas las funcionalidades, capacidades, configuraciones, limitaciones, etc de dicho dispositivo, es la información mas fiel.
 - En nuestro caso tendremos siempre presente la hoja técnica del microcontrolador PIC18F57Q43 rev. H (actualizado al 2024)

	MICROCHIP	PIC18F27/47/57Q43																																																																																																																				
<hr/>																																																																																																																						
28/40/44/48-Pin, Low-Power, High-Performance Microcontroller with XLP Technology																																																																																																																						
<hr/>																																																																																																																						
<h2>Introduction</h2> <p>The PIC18-Q43 microcontroller family is available in 28/40/44/48-pin devices for real-time control applications. This family features a 12-bit Analog-to-Digital Converter with Computation (ADC) automating Capacitive Voltage Divider (CVD) techniques for advanced capacitive touch sensing, averaging, filtering, oversampling and threshold comparison. This family showcases a new 16-bit Pulse-Width Modulator (PWM) module which provides dual independent outputs on the same time base. Additional features include vectored interrupt controller with free running timer, two serial communication modules (I²C, SPI), a 16-bit USART, a 16-bit Full Duplex Serial Asynchronous Receiver-Transmitter (UART) with support for asynchronous, Digital Multiplex (DMX), Digital Addressable Lighting Interface (DALI) and Local Interconnect Network (LIN) protocols, Serial Peripheral Interface (SPI), I²C, memory features including Memory Access Partition (MAP) to support users in data protection and bootloader applications, and Device Information Area (DIA), which stores factory calibration values to help improve temperature sensor accuracy.</p>																																																																																																																						
<h2>PIC18-Q43 Family Types</h2>																																																																																																																						
Table 1. Devices Included in This Data Sheet																																																																																																																						
<table border="1"> <thead> <tr> <th>Device</th> <th>Programmable Flash (bytes)</th> <th>EEPROM (bytes)</th> <th>SRAM (bytes)</th> <th>Code ROM (bytes)</th> <th>Code RAM (bytes)</th> <th>Memory Access Protection Area</th> <th>Data Information Area</th> <th>I²C (bytes)</th> <th>SPI (bytes)</th> <th>USART (bytes)</th> <th>USART with LIN (bytes)</th> <th>16-bit USART (bytes)</th> <th>DMX (bytes)</th> <th>CDC (bytes)</th> <th>Configurable Watchdog Timer</th> <th>RTC Measurement Timer</th> <th>Microcontroller Oscillator</th> <th>Configurable Logic Cell</th> <th>8-Bit DAC</th> <th>Comparator</th> <th>Zero-Cross Detect</th> <th>High-Low Water Detect</th> <th>gPMC</th> <th>UART with Serial Support</th> <th>Direct Memory Access (DMA)</th> <th>Watchdog Timing</th> <th>16-Bit CRC with Buffer</th> <th>Watchdog Interrupt</th> <th>PortPin with Pull-Up Driver</th> <th>Temperature Indicator</th> </tr> </thead> <tbody> <tr> <td>PIC18F27Q43</td> <td>128K</td> <td>8192</td> <td>1024</td> <td>YY</td> <td>YY</td> <td>25Y</td> <td>3/4</td> <td>3/3</td> <td>3</td> <td>1</td> <td>3</td> <td>85</td> <td>1</td> <td>2/1</td> <td>1</td> <td>2/1</td> <td>4/1</td> <td>4/1</td> <td>6</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> </tr> <tr> <td>PIC18F47Q43</td> <td>128K</td> <td>8192</td> <td>1024</td> <td>YY</td> <td>YY</td> <td>56Y</td> <td>3/4</td> <td>3/3</td> <td>3</td> <td>1</td> <td>3</td> <td>85</td> <td>1</td> <td>2/1</td> <td>1</td> <td>2/1</td> <td>4/1</td> <td>6</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> </tr> <tr> <td>PIC18F57Q43</td> <td>128K</td> <td>8192</td> <td>1024</td> <td>YY</td> <td>YY</td> <td>44Y</td> <td>3/4</td> <td>3/3</td> <td>3</td> <td>1</td> <td>3</td> <td>85</td> <td>1</td> <td>2/1</td> <td>1</td> <td>2/1</td> <td>4/1</td> <td>6</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> <td>Y</td> </tr> </tbody> </table>			Device	Programmable Flash (bytes)	EEPROM (bytes)	SRAM (bytes)	Code ROM (bytes)	Code RAM (bytes)	Memory Access Protection Area	Data Information Area	I ² C (bytes)	SPI (bytes)	USART (bytes)	USART with LIN (bytes)	16-bit USART (bytes)	DMX (bytes)	CDC (bytes)	Configurable Watchdog Timer	RTC Measurement Timer	Microcontroller Oscillator	Configurable Logic Cell	8-Bit DAC	Comparator	Zero-Cross Detect	High-Low Water Detect	gPMC	UART with Serial Support	Direct Memory Access (DMA)	Watchdog Timing	16-Bit CRC with Buffer	Watchdog Interrupt	PortPin with Pull-Up Driver	Temperature Indicator	PIC18F27Q43	128K	8192	1024	YY	YY	25Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y	PIC18F47Q43	128K	8192	1024	YY	YY	56Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y	PIC18F57Q43	128K	8192	1024	YY	YY	44Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y
Device	Programmable Flash (bytes)	EEPROM (bytes)	SRAM (bytes)	Code ROM (bytes)	Code RAM (bytes)	Memory Access Protection Area	Data Information Area	I ² C (bytes)	SPI (bytes)	USART (bytes)	USART with LIN (bytes)	16-bit USART (bytes)	DMX (bytes)	CDC (bytes)	Configurable Watchdog Timer	RTC Measurement Timer	Microcontroller Oscillator	Configurable Logic Cell	8-Bit DAC	Comparator	Zero-Cross Detect	High-Low Water Detect	gPMC	UART with Serial Support	Direct Memory Access (DMA)	Watchdog Timing	16-Bit CRC with Buffer	Watchdog Interrupt	PortPin with Pull-Up Driver	Temperature Indicator																																																																																								
PIC18F27Q43	128K	8192	1024	YY	YY	25Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y																																																																																										
PIC18F47Q43	128K	8192	1024	YY	YY	56Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y																																																																																											
PIC18F57Q43	128K	8192	1024	YY	YY	44Y	3/4	3/3	3	1	3	85	1	2/1	1	2/1	4/1	6	Y	Y	Y	Y	Y	Y	Y	Y	Y																																																																																											

Revisión de documentos

- Hoja técnica del PIC18F57Q43
 - <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/PIC18F27-47-57Q43-Data-Sheet-40002147F.pdf> (rev. F)
 - <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/PIC18F27-47-57Q43-Microcontroller-Data-Sheet-XLP-DS40002147.pdf> (rev. H 2024)
- Hoja técnica del Curiosity Nano PIC18F57Q43
 - <http://ww1.microchip.com/downloads/en/DeviceDoc/PIC18F57Q43-Curiosity-Nano-HW-UserGuide-DS40002186B.pdf>
- MPLAB XC8 PIC Assembler User Guide
 - //Program Files/Microchip/xc8/v3.10/docs/MPLAB_XC8_PIC_Assembler_User_Guide.pdf

Procedimiento para desarrollar una aplicación con el microcontrolador PIC18F57Q43 en los laboratorios

1. Análisis del problema y ver los requerimientos (puertos E/S, tipo de señales, velocidad, consumo energético, etc)
2. Desarrollamos el hardware (el circuito implementado)
3. Elaboramos el algoritmo (Flowchart, Nassi-Schneiderman, etc)
4. Redactamos el código en un lenguaje de programación (Assembler, BASIC, C, Python, etc) en el IDE
5. Compilar y realizar la pruebas (simulación (SW), emulación (HW), programación (HW))

Importancia de los comentarios en un código fuente

- Cuando uno desarrolla un programa, en cualquier lenguaje de programación, es fundamental colocar comentarios.
- Los comentarios no añaden espacio de memoria luego de la compilación.
- Los comentarios sirven para recordar ideas, configuraciones, procesos, algoritmos, etc que le permitan al programador en un tiempo después ver lo que hizo en dicho momento.
- En MPASM ó XC8 PICASM los comentarios van antecedidos por un punto y coma (;)

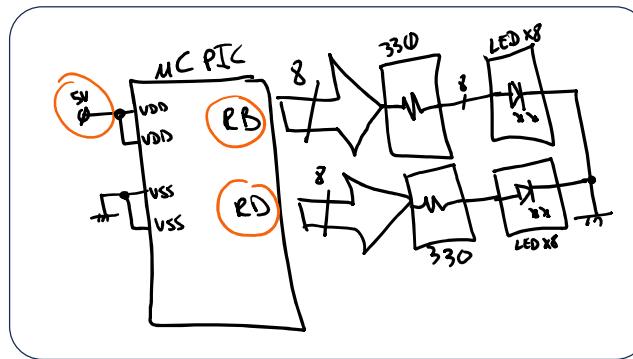
Consideraciones importantes al usar simuladores

- El uso de simuladores ha permitido acelerar los procesos de validación de circuitos eléctricos y electrónicos, **pero** no es un determinante a la hora de validar en forma física.
- En la mayoría de casos en ingeniería electrónica el producto final es algo físico por lo que no solamente podemos fiarnos de una simulación y dar por sentado que la propuesta funcione correctamente.
- En Proteus hay elementos que no se muestran en el momento de hacer simulaciones.
- El software Proteus a la fecha no posee módulo de simulación del microcontrolador PIC18F57Q43

Sobre el MPLAB X IDE

1. Crear un proyecto (seleccionar Standalone Project)
2. Seleccionar el dispositivo microcontrolador (PIC18F57Q43)
3. Seleccionar la herramienta “pic-as” (XC8 PIC Assembler)
4. Crear el archivo header (*.inc) e incluirle los bits de configuración (Window / Target Memory Views / Configuration Bits), dicho archivo debe de estar en la carpeta “Header Files”
5. Crear el archivo fuente (*.s) e incluir el archivo header dentro del cuerpo, dicho archivo debe de estar en la carpeta “Source Files”
6. Para compilar: 
7. El archivo generado de la compilación tiene extensión *.hex ó *.elf
8. Para grabar en el Curiosity Nano: 

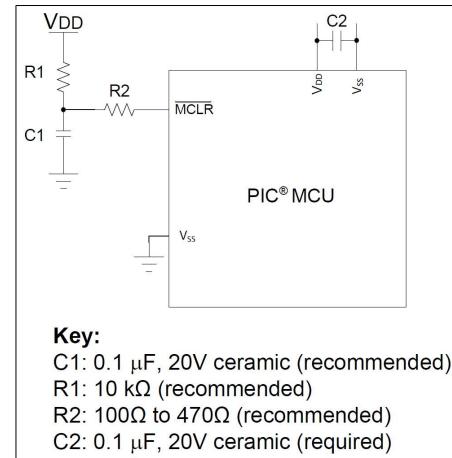
Hay dos pines de Vdd y dos pines de Vss, es necesario conectar todos?



Se deben de conectar todos los pines de alimentación para que el microcontrolador pueda obtener mayor capacidad de corriente en caso lo requiera la aplicación.

Detalles técnicos iniciales

- Voltaje de alimentación del PIC18F57Q43
 - **Voltajes menores a -0.5V son perjudiciales**
 - Voltaje de operación máximo 5.5V
 - Voltaje de operación mínimo 1.8V



Ejemplo inicial

- Configuration bits:

300000	CONFIG1	FC	-	-	-	-
	4	EXTOSC	OFF	External Oscillator Selection	Oscillator not enabled	
	7	RSTOSC	EXTOSC	Reset Oscillator Selection	EXTOSC operating per FEXTOSC bits (device manufac	
300001	CONFIG2	FF	-	-	-	-
	1	CLKOUTEN	OFF	Clock out Enable bit	CLKOUT function is disabled	
	1	PRIWAY	ON	PRLOCKED One-Way Set Enable bit	PRLOCKED bit can be cleared and set only once	
	1	CSWEN	ON	Clock Switch Enable bit	Writing to NOSC and NDIV is allowed	
	1	FCMEN	ON	Fail-Safe Clock Monitor Enable bit	Fail-Safe Clock Monitor enabled	
300002	CONFIG3	3D	-	-	-	-
	1	MCLRE	EXTMCLR	MCLR Enable bit	If LVP = 0, MCLR pin is MCLR; If LVP = 1, RE3 pin f	
	2	PWRIS	PWRT_64	Power-up timer selection bits	PWRT set at 64ms	
	1	MVECEN	ON	Multi-vector enable bit	Multi-vector enabled, Vector table used for interrupt	
	1	IVTIWAY	ON	IVTLOCK bit One-way set enable bit	IVTLOCKED bit can be cleared and set only once	
	1	LPBOREN	OFF	Low Power BOR Enable bit	Low-Power BOR disabled	
	0	BOREN	OFF	Brown-out Reset Enable bits	Brown-out Reset disabled	
300003	CONFIG4	DF	-	-	-	-
	3	BORV	VBOR_1P9	Brown-out Reset Voltage Selection bits	Brown-out Reset Voltage (VBOR) set to 1.9V	
	1	ZCD	OFF	ZCD Disable bit	ZCD module is disabled. ZCD can be enabled by settin	
	1	PPS1WAY	ON	PPSLOCK bit One-Way Set Enable bit	PPSLOCKED bit can be cleared and set only once; PPS	
	1	STVREN	ON	Stack Full/Underflow Reset Enable bit	Stack full/underflow will cause Reset	
	0	LVP	OFF	Low Voltage Programming Enable bit	HV on MCLR/VPP must be used for programming	
	1	XINST	OFF	Extended Instruction Set Enable bit	Extended Instruction Set and Indexed Addressing Mode	
300004	CONFIG5	9E	-	-	-	-
	1F	WDTCP	WDTCP_31	WDT Period selection bits	Divider ratio 1:65536; software control of WDTPS	
	0	WDTE	[OFF]	WDT operating mode	WDT Disabled; SWDTEN is ignored	
300005	CONFIG6	FF	-	-	-	-
	7	WDTIWS	WDTIWS_7	WDT Window Select bits	Window always open (100%); software control; keyed	
	7	WDTSC	SC	WDT input clock selection	Software Control	

Ejemplo inicial

- Configuration bits:
 - External Oscillator Selection: OFF
 - Powerup Timer: 64ms
 - Brownout Reset: OFF
 - Low Voltage Programming: OFF
 - Watchdog Timer: OFF

Ejemplo inicial

- Plantilla de un programa en XC8 PIC Assembler:

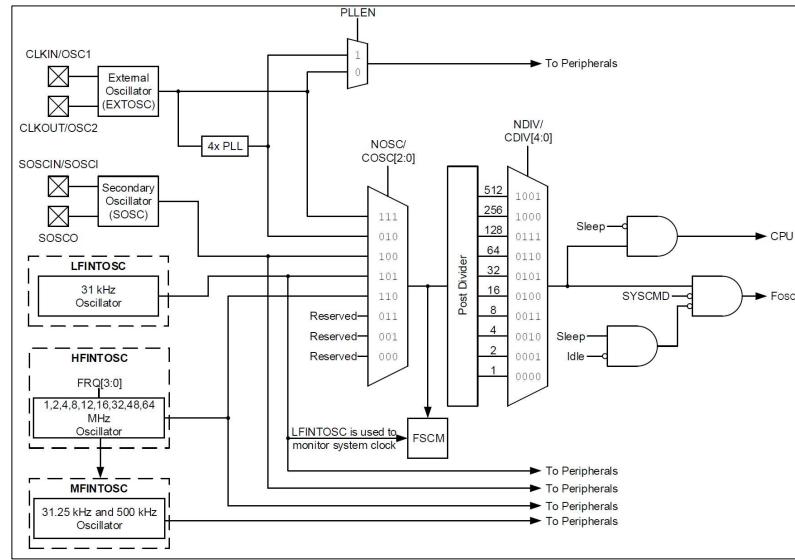
```

1      PROCESSOR 18F57Q43
2      #include "cabecera.inc"
3
4      PSECT principal, class=CODE, reloc=2, abs
5      principal:
6          ORG 000000H          ;Vector de RESET
7          bra configuro      ;Salto hacia etiqueta configuro
8
9          ORG 000020H          ;Zona de programa de usuario
10         configuro:          ;Configuraciones de la aplicación (puertos, periféricos, etc)
11             ;Detalle de lo que va a hacer el microcontrolador
12             bra inicio          ;Salto hacia etiqueta inicio
13
14         inicio:
15             ;Detalle de lo que va a hacer el microcontrolador
16             bra inicio          ;Salto hacia etiqueta inicio
17
18         end principal

```

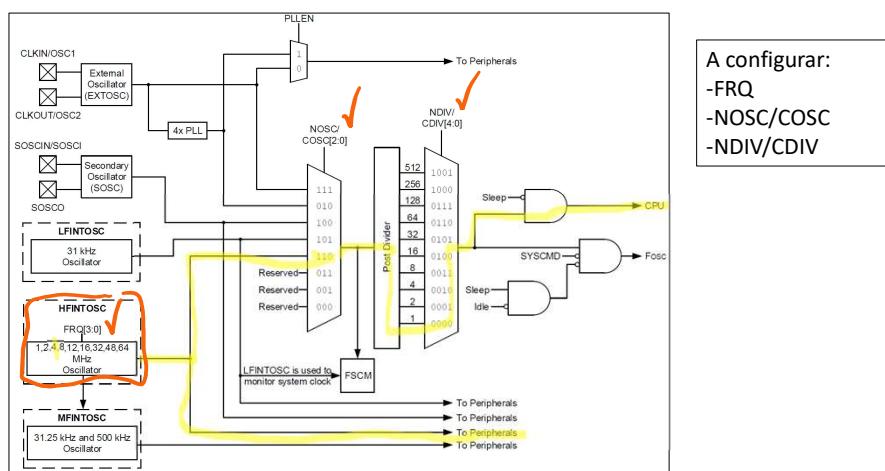
Detalles técnicos del modulo de oscilador (fuente de reloj)

- Oscilador externo (oscilador primario) sirve para que se obtenga la Fuente de reloj de manera externa, ya sea por un cristal o una Fuente de onda cuadrada)
- Oscilador secundario sirve para colocar un cristal de 32.768KHz y hacer aplicaciones de reloj en tiempo real con el Timer1.
- El PLL de este microcontrolador sirve para incrementar la frecuencia de trabajo, hasta 64MHz.
- Usaremos el oscilador interno HFINTOSC para el CPU en todas nuestras aplicaciones.
- MFINTOSC no puede derivarse hacia el CPU, solo hacia periféricos.



Configuración inicial de la fuente de reloj

Para obtener 4MHz al CPU y periféricos a partir del oscilador interno:



Configuración inicial de fuente de reloj

Registros implicados en la configuración del reloj:

A configurar:
 -FRQ
 -NOSC/COSC
 -NDIV/CDIV

12.6 Register Summary - Oscillator Module

Address	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xAC	ACTCON	7:0	ACTEN	ACTUD			ACTLOCK		ACTORS	
0xAD	OSCCON1	7:0			NOSC[2:0]				NDIV[3:0]	
0xAE	OSCCON2	7:0			COSC[2:0]				CDIV[3:0]	
0xAF	OSCCON3	7:0	CSWHOLD	SOSCPWR		ORDY	NOSCR			
0xB0	OSCTUNE	7:0					TUN[5:0]			
0xB1	OSCFRQ	7:0							FRQ[3:0]	
0xB2	OSCSTAT	7:0	EXTOR	HFOR	MFOR	LFOR	SOR	ADOR		PLLR
0xB3	OSCEN	7:0	EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN		PLLEN

Configuración inicial de fuente de reloj

Registro OSCCON1:

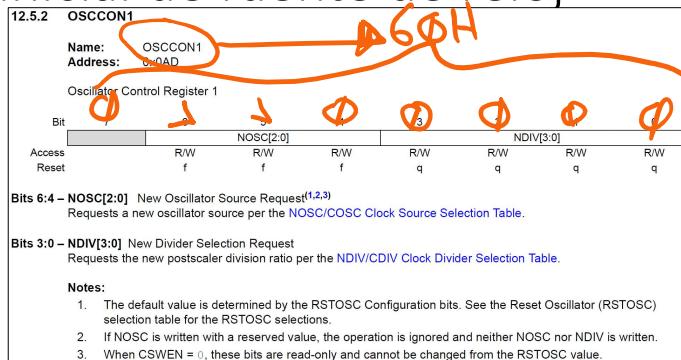


Table 12-2. NOSC/COSC Clock Source Selection Table

NOSC / COSC	Clock Source
111	EXTOSC ⁽¹⁾
110	HFINTOSC ⁽²⁾
101	LFINTOSC
100	SOSC
011	Reserved
010	EXTOSC + 4xPLL ⁽³⁾
001	Reserved
000	Reserved

Table 12-3. NDIV/CDIV Clock Divider Selection Table

NDIV / CDIV	Clock Divider
1111-1010	Reserved
1001	512
1000	256
0111	128
0110	64
0101	32
0100	16
0011	8
0010	4
0001	2
0000	1

Configuración inicial de fuente de reloj

Registro OSCFRQ:

12.5.6 OSCFRQ							
Name: OSCFRQ		Address: 0x051					
HFINTOSC Frequency Selection Register							
Bit	7	6	5	4	3	2	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0

Bits 3:0 – FRQ[3:0] HFINTOSC Frequency Selection

FRQ	Nominal Freq (MHz)
1111-1001	Reserved
1000	64
0111	48
0110	32
0101	16
0100	12
0011	8
0010	4
0001	2
0000	1

Configuración inicial de fuente de reloj

Registro OSCEN:

12.5.8 OSCEN							
Name: OSCEN		Address: 0x052					
Oscillator Enable Register							
Bit	7	6	5	4	3	2	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0

Bit 7 – EXTOEN External Oscillator Enable

Value	Description
1	EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0	EXTOSC can be enabled by a peripheral request

Bit 6 – HFOEN HFINTOSC Enable

Value	Description
1	HFINTOSC is explicitly enabled, operating as specified by OSCFRQ
0	HFINTOSC can be enabled by a peripheral request

Bit 5 – MFOEN MFINTOSC Enable

Value	Description
1	MFINTOSC is explicitly enabled
0	MFINTOSC can be enabled by a peripheral request

Bit 4 – LFOEN LFINTOSC Enable

Value	Description
1	LFINTOSC is explicitly enabled
0	LFINTOSC can be enabled by a peripheral request

Bit 3 – SOSCEN Secondary Oscillator Enable

Value	Description
1	SOSC is explicitly enabled, operating as specified by SOSCPWR
0	SOSC can be enabled by a peripheral request

Bit 2 – ADOEN ADCRC Oscillator Enable

Value	Description
1	ADCRC is explicitly enabled
0	ADCRC may be enabled by a peripheral request

Bit 0 – PLLEN PLL Enable⁽¹⁾

Value	Description
1	EXTOSC multiplied by the 4x system PLL is used by a peripheral request
0	EXTOSC is used by a peripheral request

Note:

- 1. This bit only controls external clock source supplied to the peripherals and has no effect on the system clock.

Ejemplo inicial

- Configuraremos el PIC18F57Q43 para que funcione a 4MHz con el HFINTOSC

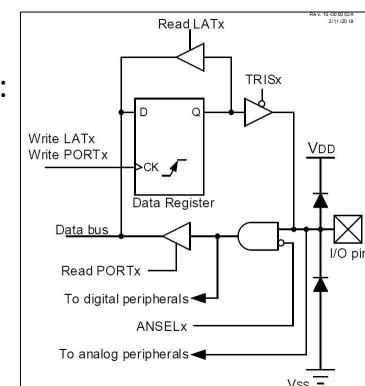
```

1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT principal, class=CODE, reloc=2, abs
5  principal:
6      ORG 000000H          ;Vector de RESET
7      bra configuro        ;Salto hacia etiqueta configuro
8
9      ORG 000020H          ;Zona de programa de usuario
10 configuro:
11      ;Configuraciones de la aplicación (puertos, periféricos, etc)
12      ;Configuración de la fuente de reloj para 4MHz con el HFINTOSC
13      movlb 0H              ;Bank 0 donde estan los registros de conf del OSC
14      ;OSCFRQ = 02H
15      movlw 02H              ;muevo literal 02H a wreg
16      movwf OSCFRQ, 1         ;muevo contenido de wreg hacia OSCFRQ
17      ;OSCCON1 = 60H
18      movlw 60H              ;muevo literal 60H a wreg
19      movwf OSCCON1, 1         ;muevo contenido de wreg hacia OSCCON1
20      ;OSCEN = 40H
21      movlw 40H              ;muevo literal 40H a wreg (0100 0000)
22      movwf OSCEN, 1           ;muevo contenido de wreg hacia OSCEN
23
24 inicio:
25     ;Detalle de lo que va a hacer el microcontrolador
26     bra inicio            ;Salto hacia etiqueta inicio
27
28 end principal

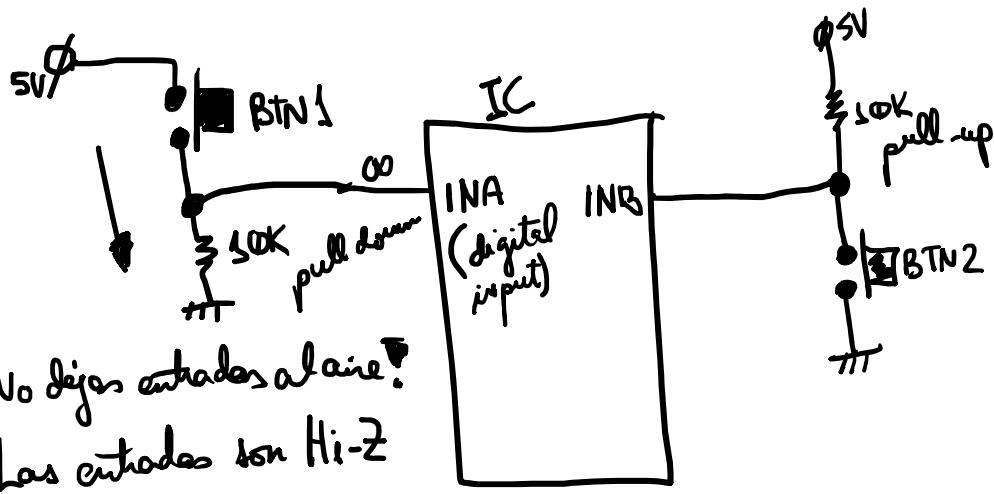
```

Manejo de puertos de E/S en el PIC18F57Q43

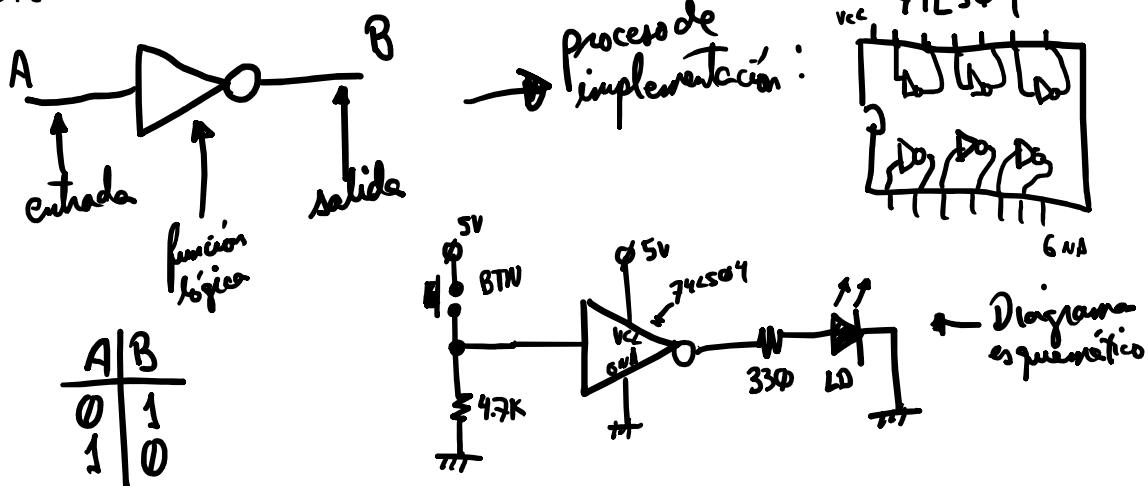
- Revisar 19.0 en la hoja técnica
- Por defecto los puertos están como entradas analógicas.**
- Se tienen los siguientes registros para manipular los puertos:
 - TRISx Para configurar el sentido del puerto (entrada ó salida), cero para salida y uno para entrada.
 - PORTx Para leer el puerto
 - LATx Para escribir el puerto
 - ANSELx Para configurar el puerto en analógico o digital, uno para analógico y cero para digital.
 - WPUx Para habilitar las resistencias de pullup: 1 activado, 0 desactivado
 - SLRCON Para configurar la velocidad de respuesta en el puerto configurado como salida.



Diferencia entre pulsadores activos en alto y activos en bajo



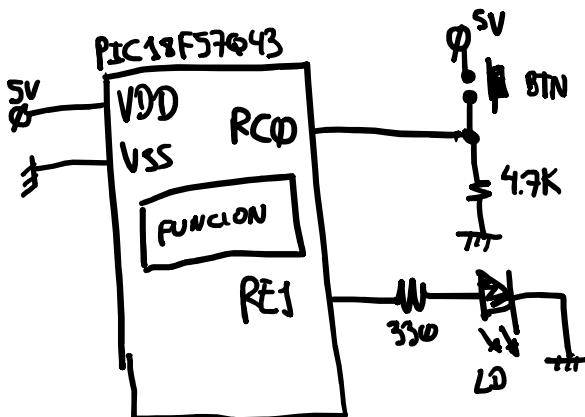
Primer ejemplo 2025-1: Negador lógico de un bit



¿Cómo migro el presente diseño hacia el microcontrolador?

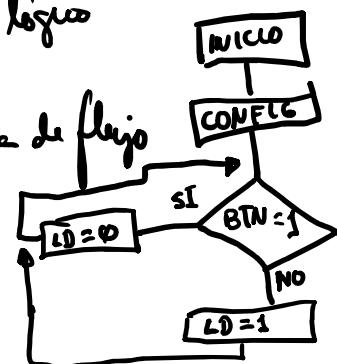
Primer ejemplo 2025-1: Negador lógico de un bit

¿Cómo migro el presente diseño hacia el microcontrolador?



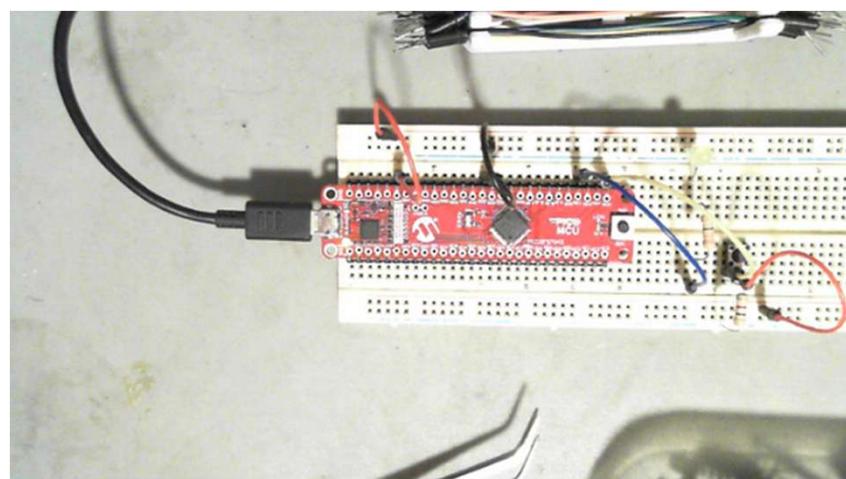
Hay que meterle la funcionalidad del negador lógico en el uC!

→ Diagrama de flujo



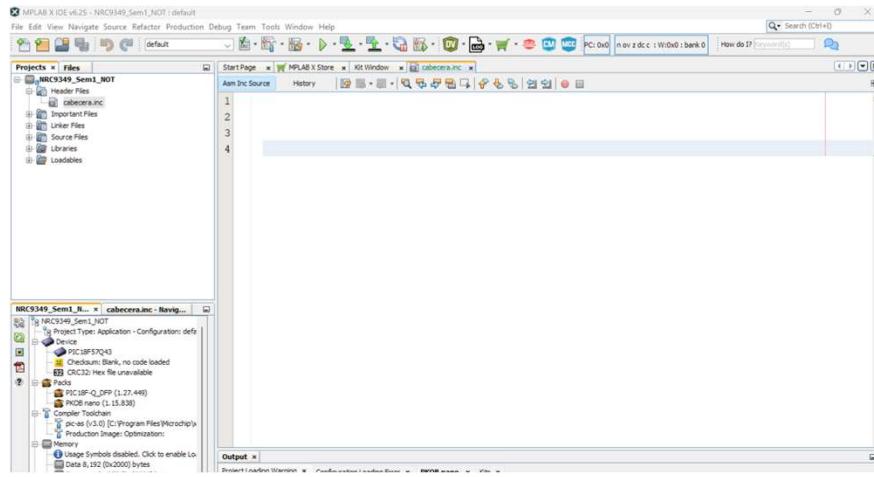
Primer ejemplo 2025-1: Negador lógico de un bit

¿Cómo migro el presente diseño hacia el microcontrolador?



Primer ejemplo 2025-1: Negador lógico de un bit

¿Cómo migro el presente diseño hacia el microcontrolador?



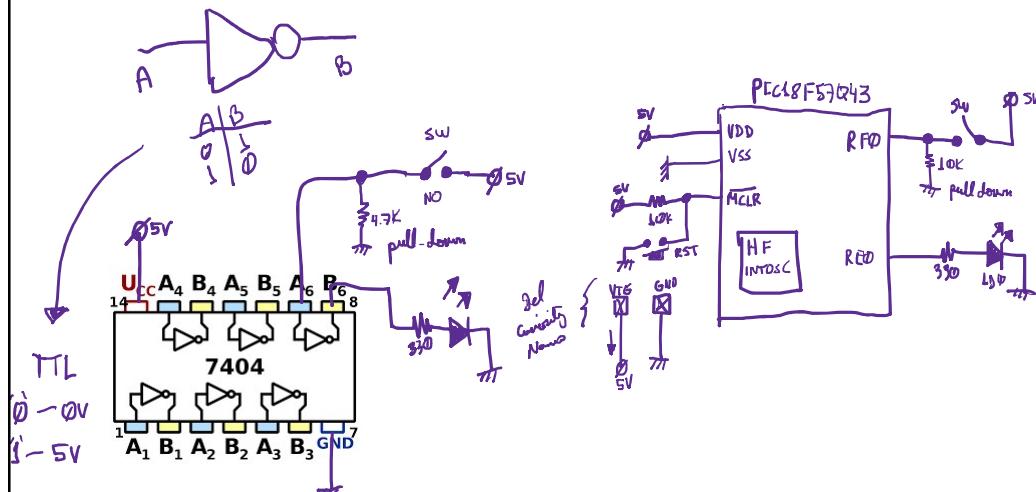
Primer ejemplo 2025-1: Negador lógico de un bit

```

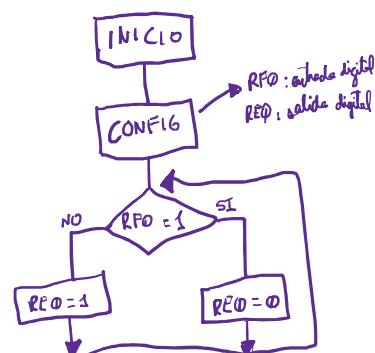
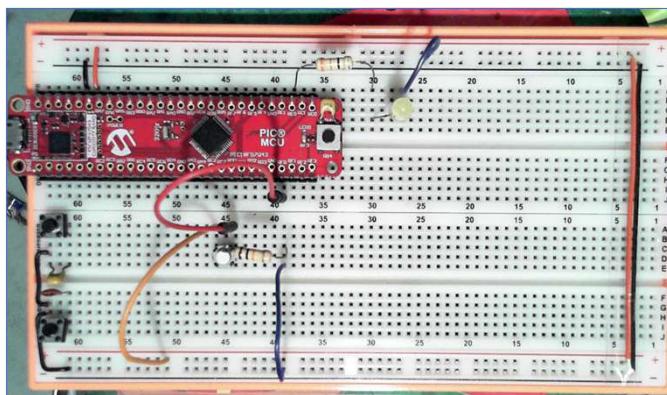
5      PROCESSOR 18F57Q43      ;modelo de CPU
6      ;llamada a la cabecera
7      #include "cabecera.inc"
8
9      ;Creación de un program section:
10     PSECT upcino, class=CODE, reloc=2, abs
11     upcino:
12         ;empieza el código fuente
13         ORG 000000H      ;vector de RESET
14         bra configuro
15
16         ORG 000100H      ;Zona de programa de usuario
17     configuro:
18         ;Configuracion de fuente de reloj para que funcione el CPU
19         ;Vamos a configuracion el HFINTOSC como fuente de reloj a 4MHz
20         movlb 0H          ;nos vamos al Bank0
21         movlw 60H          ;HFINTOSC como fuente y divisor a 1:1
22         movwf OSCCON1, 1
23         movlw 02H
24         movwf OSCFRQ, 1   ;HFINTOSC trabaja a 4MHz
25         bcf OSCEN, 6, 1   ;encendemos el HFINTOSC
26
27         ;Configuracion de las E/S
28         movlb 4H          ;nos vamos al Bank4
29         ;RE1 como salida digital (LED)
30         bcf TRISE, 1, 1   ;RE1 como salida
31         bcf ANSEL1, 1, 1   ;RE1 como digital
32         ;RC0 como entrada digital (BOTON)
33         bcf TRISC, 0, 1    ;RC0 como entrada
34         bcf ANSELC, 0, 1    ;RC0 como digital
35
36         ;Recien aca esta el codigo para el negador lógico
37         inicio:
38             btfs PORTC, 0, 1  ;preguntamos si el boton se machucó
39             bra nolopresione  ;Falso, salta a etiqueta nolopresione
40             bra silopresione  ;Verdad, salta a label silopresione
41
42         nolopresione:
43             bsf LATE, 1, 1    ;encendemos el LED
44             bra inicio        ;retorna a inicio parapreguntar de nuevo
45
46         silopresione:
47             bcf LATE, 1, 1    ;apagamos el LED
48             bra inicio        ;retorna a inicio parapreguntar de nuevo
49
50     end upcino

```

Primer ejemplo 2025-2: Negador lógico



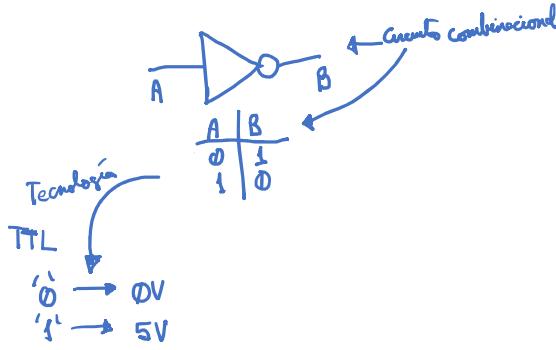
Primer ejemplo 2025-2: Negador lógico



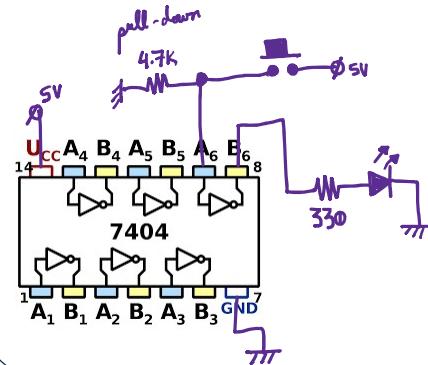
Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

Análisis del problema: Revisión de conceptos



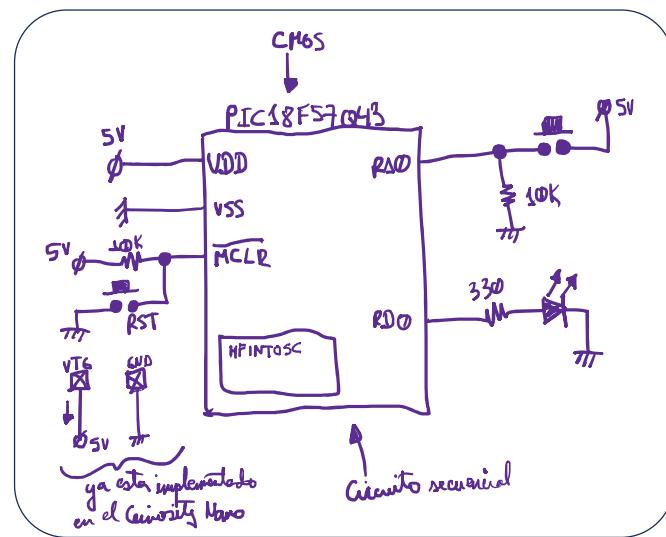
Implementación con circuito 7404



Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

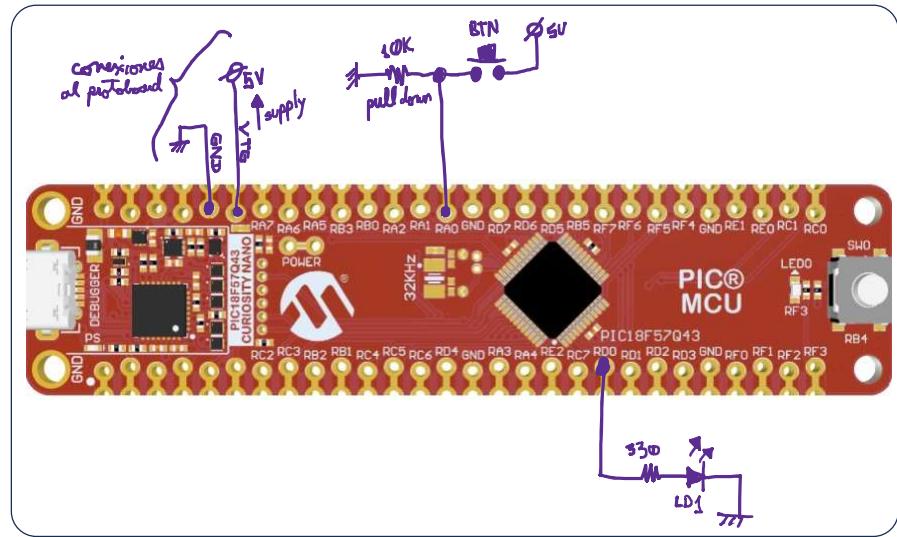
Diagrama esquemático de la solución empleando el microcontrolador PIC18F57Q43



Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

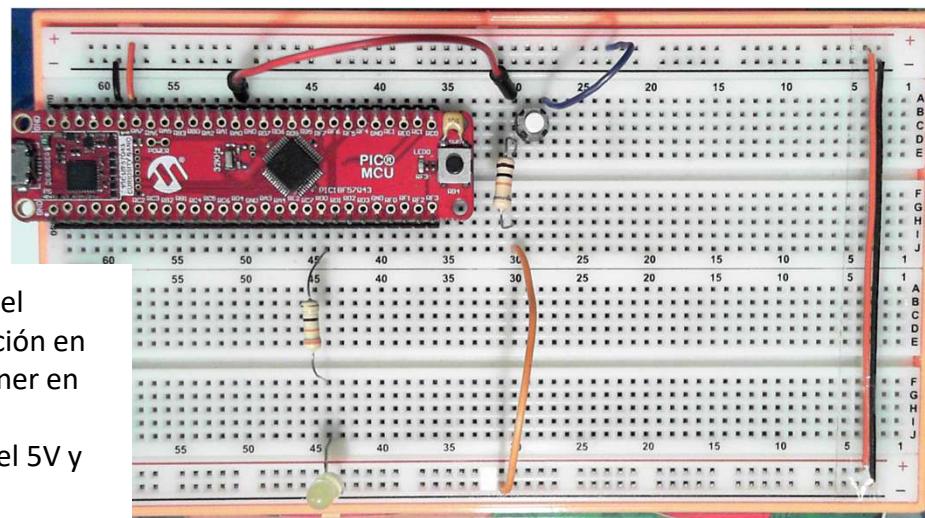
Diagrama de conexiones de la solución empleando el Curiosity Nano PIC18F57Q43



Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

Implementación del circuito de la solución en un protoboard, tener en cuenta los rieles horizontales para el 5V y GND

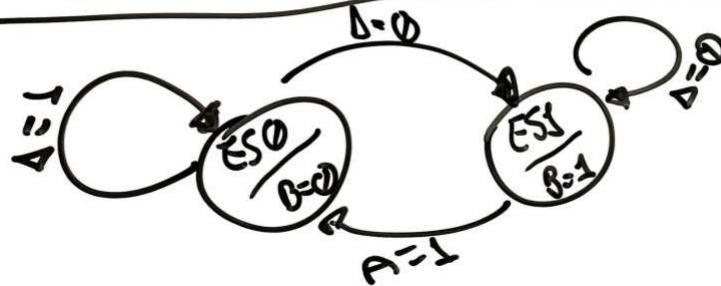


Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

Modelo en MEF Moore de la NOT

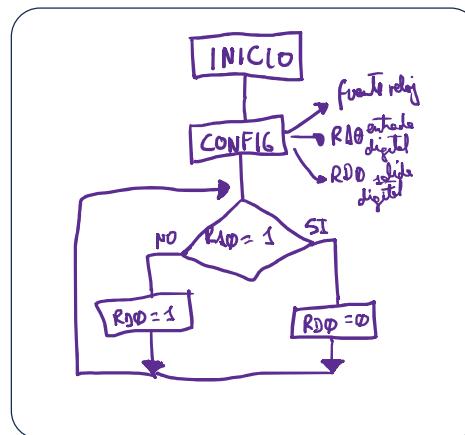
Modelando la NOT en MEF Moore :

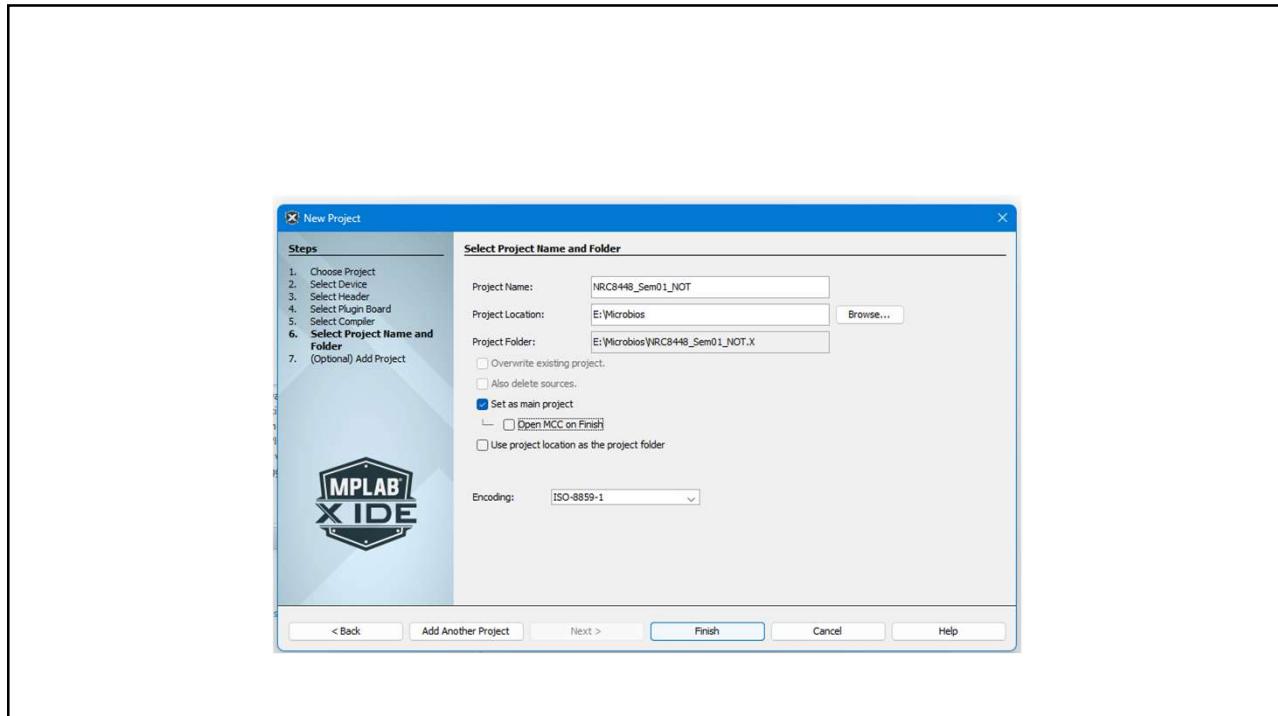
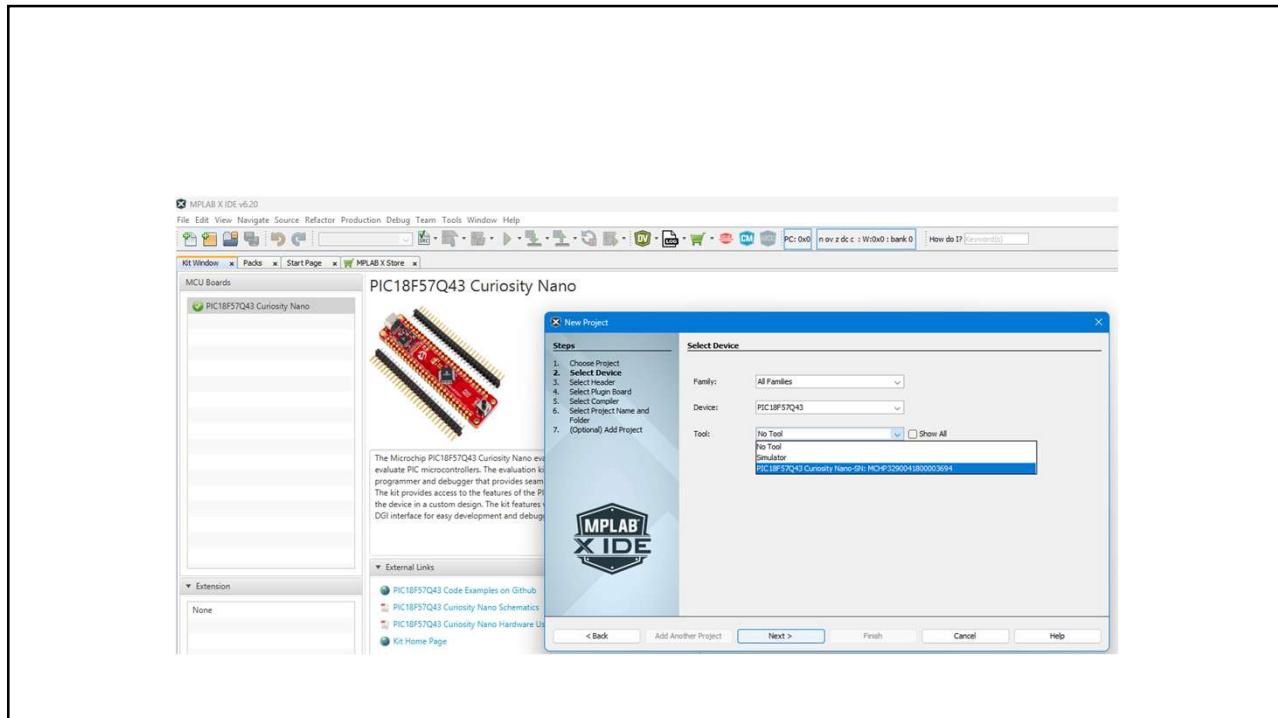


Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

Algoritmo en diagrama de flujo para emular la compuerta NOT en el microcontrolador PIC18F57Q43





Primer ejemplo 2025-2

- Implementar una compuerta NOT de un bit con el microcontrolador PIC18F57Q43

Código propuesto, tener en cuenta los comentarios en este primer ejemplo

```

1 ;Directiva de procesador, para indicarle al compilador
2 PROCESSOR 18F57Q43
3 ;Llamada al archivo de cabecera
4 #include "cabecera.inc"
5
6 ;Crear una sección de programa (PSECT)
7 PSECT upcino, class=CODE, reloc=2, abs
8 upcino:
9     ORG 000000H      ;posición de memoria de programa: VECTOR DE RESET
10    bra configuro   ;salto a etiqueta configuro
11
12    ORG 000100H      ;posición de memoria de programa 100H
13 configuro:
14        ;configurar la fuente de reloj
15        movlb 0H          ;me voy al BANK0
16        movlw 60H
17        movwf OSCCON1, 1   ;NOSC->HFINTOSC, NDIV->1:1
18        movlw 02H
19        movwf OSCFRC, 1    ;HFINTOSC a 4MHz
20        movlw 40H
21        movwf OSCEN, 1     ;HFINTOSC enabled
22        ;configurar las E/S
23        movlb 4H          ;me voy al BANK4
24        bsf TRISA, 0, 1    ;RA0 como entrada
25        bcf ANSELA, 0, 1   ;RA0 como digital
26        bcf TRISE, 0, 1    ;RD0 como salida
27        bcf ANSELD, 0, 1   ;RD0 como digital
28
29 inicio:
30     btfss PORTA, 0, 1  ;Pregunto si RA0 es uno
31     bra esfalso         ;es falso, salta a etiqueta esfalso
32     bra esverdadero     ;es verdadero, salta a etiqueta esverdadero
33 esfalso:
34     bsf LATD, 0, 1      ;RD0 es uno
35     bra inicio          ;regreso a preguntar RA0
36 esverdadero:
37     bcf LATD, 0, 1      ;RD0 es cero
38     bra inicio          ;regreso a preguntar RA0
39
40 end upcino           ;cierra del program section upcino

```

Ejercicios propuestos:

- Modificar el circuito y diagrama de flujo para que la compuerta NOT implementada anteriormente sea de dos bits.
- Implementar el funcionamiento de una compuerta XOR de un bit.
- Implementar un multiplexor de 2 a 1 de un bit

Fin de la sesión!