

Microcontroladores y protocolos de comunicación Serial

Ing. Yeiner Arias Esquivel

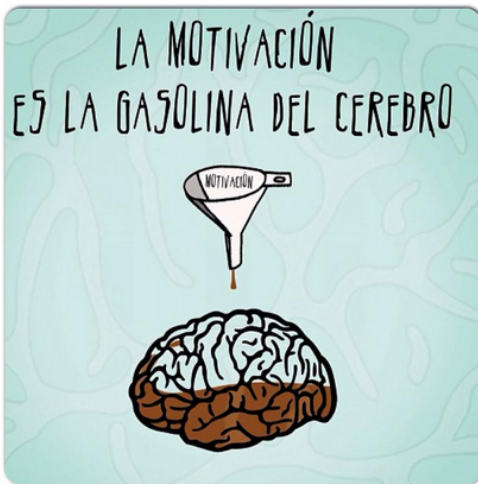
Tecnológico de Costa Rica
Área Académica de Ingeniería Mecatrónica
Microprocesadores y microcontroladores

2 de octubre de 2018

Agenda

- 1 Motivación
- 2 Microcontrolador
 - Definiciones
 - Comparación con CPU
 - Características
 - Arquitectura
- 3 Protocolos Seriales típicos
 - Definición
 - I²C
 - SPI
 - UART RS232
- 4 Referencias

Motivación sobre microcontroladores y protocolos de comunicación



Definición

Pregunta 1

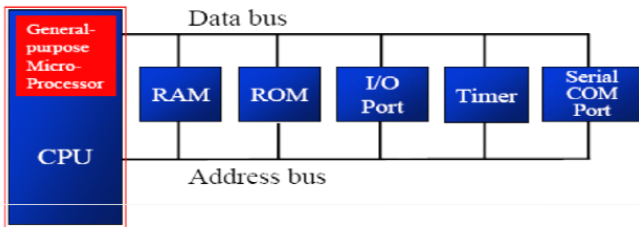
¿Qué es un microcontrolador?

¿Qué es un microcontrolador?

Es un sistema computacional integrado en un chip (SOC).

- Núcleo de procesamiento
- Memoria para datos, código
- Dispositivos periféricos básicos
- **Orientado a aplicaciones específicas**

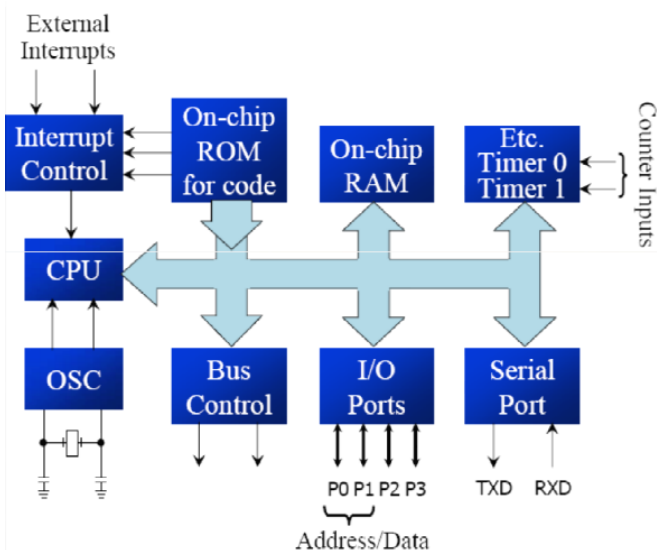
CPU vs Microcontrolador



Microcontroller



Arquitectura típica de un MCU



Orientación en diseño con MCU

- Se enfoca el diseño en una aplicación específica solventa un problema en un proceso.
- Bajo consumo de energía.
- Soporta el ambiente de trabajo para el que fue diseñado.
- Mínima interacción Humano-MCU, sistemas autónomos.

Características necesarias en un MCU

- Respuesta en tiempo real
- Manejo versátil de interrupciones: Priorización
- Eficiente uso de la memoria de programa
- Capacidad de interacción con señales analógicas. ADC integrados.
- Capacidad de comunicación con sistemas basados en microprocesador, RS232, USB, SPI, I²C

¿Que debo conocer en un MCU?

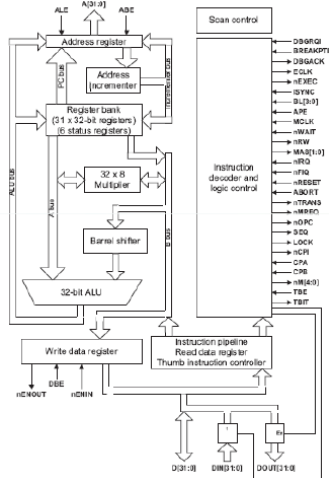
- Arquitectura y modelo de programación
- Administración de la memoria: Mapa, métodos de acceso, etc
- Protocolos de comunicación interna y externa que soporta el MCU
- Administración de I/O: mapeo, interacción con los buses, interrupciones, configuración, resolución, etc.

Arquitectura y modelo de programación

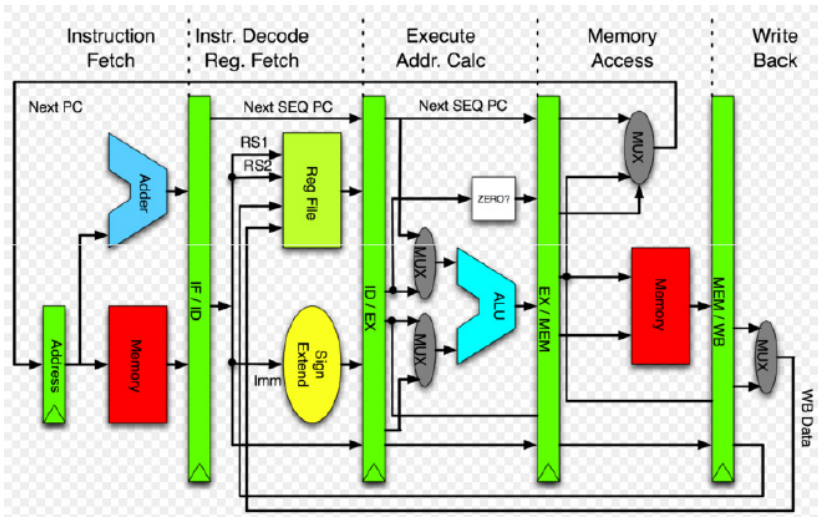
- Nivel de complejidad en manejo de datos y operaciones
- Las arquitecturas favoritas
 - ARM
 - MIPS

ARM: Diagrama de la arquitectura

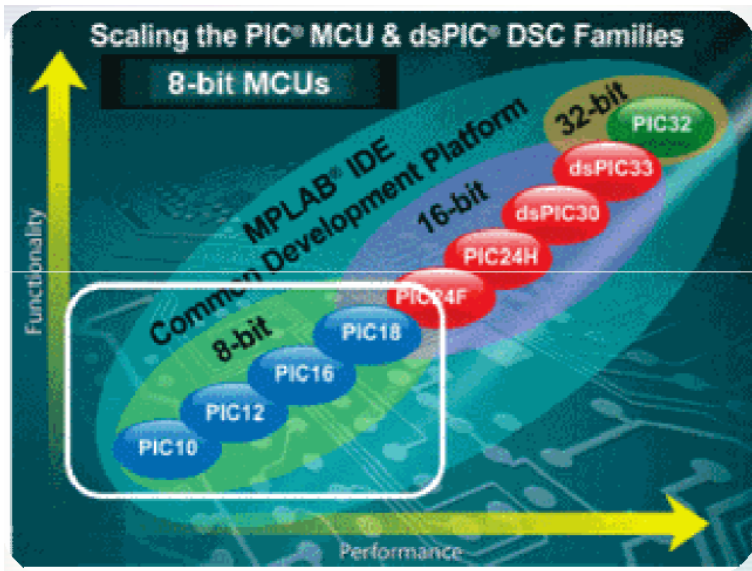
- Arquitectura RISC de 32 bits
- El 75 % de todas los sistemas empuotrados usan ARM core



Arquitectura MIPS



Evolución de los PIC



Protocolos de comunicación en MCU

Los MCU necesitan comunicar su núcleo con memorias y periféricos para ellos usa protocolos comerciales:

- I²C
- UART
- SPI
- USB

Protocolos I²C y SPI

Protocolos de comunicación en serie

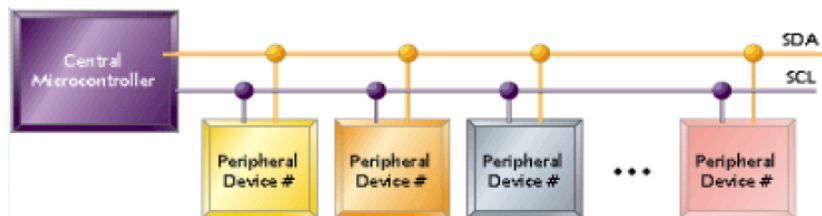
- Diseñados para transmisiones a corta distancia: “inside the box”
- Fácil implementación
- Bajo costo
- Velocidad baja (unos cuantos Mbps)

I²C

¿Qué es I²C?

- Es la abreviación en ingles para : “Inter-integrated circuit bus”
- Desarrollada en los 80s por Philips Semiconductor para TV.
- Entre los dispositivos que se comunican por I²C se encuentran EEPROMS, sensores térmicos, RTC.
- Se usa en aplicaciones con conexión a subprocesos. Por ejemplo sintonizadores, codificadores decodificadores.
- Se puede operar en tres rangos de velocidad:
 - Slow (hasta 100 Kbps)
 - Fast (400 Kbps)
 - High-speed (3.4 Mbps) – I²C v.2.0
- El alcance esta limitado a tres metros punto a multipunto.

Configuración del bus I²C



- El bus consiste en dos cables: Serial data (SDA) y Serial clock (SCL)
- Half-duplex, síncrono, multi-master bus
- No se requiere lógica de selección de chip ni arbitraje del bus.

Operación del protocolo I²C



- Nodo maestro envía señal de inicio (start= S) y controla la señal de reloj.
- Seguidamente se envía un identificador de 7 bits que representa al nodo con el que se va a establecer conexión.
- El nodo maestro envía un bit de WR/!R para indicar el tipo de operación del nodo con maestro. $WR/!R = 0 \rightarrow$ Nodo esclavo recibe.
- Nodo receptor emite una señal de acknowledge (ACK).
- Nodo transmisor (maestro o esclavo) transmite un byte de datos.

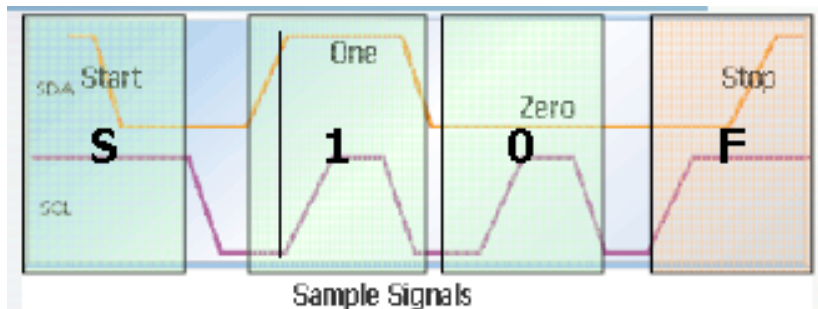
Operación del protocolo I2C



- El Nodo que recibe el byte envía un bit de ACK cuando se completa la transmisión.
- Repita pasos 5 y 6 tantas veces como bytes a transmitir.
- Cuando el nodo que transmite es el nodo maestro, este controla la finalización de la transmisión [F bit] luego del ultimo byte a transmitir
- Cuando el nodo que recibe es el maestro, este no notifica el ultimo byte sino que transmite el bit de paro [F bit] para indicar al transmisor que la comunicación se finaliza

I^2C

Control de señalización I^2C



- **Start:** cambio de alto a bajo en SDA con SCL en alto
- **Stop:** Cambio de bajo a alto en SDA con SCL fijo en Alto
- **ACK:** el nodo receptor pone SDA en bajo mientras el transmisor lo deja libre y lo lee
- **Datos:** se lleva a cabo cuando SCL esta en bajo, y se valida mientras SCL esta en alto.

Características adicionales de I²C

- “Clock stretching”: Cuando un dispositivo esclavo (RX) necesita mas tiempo para procesar un bit, este puede sostener SCL en bajo y el Maestro (TX) espera a que SCL sea liberado por nodo RX para enviar un nuevo bit
- “General call” broadcast: permite transmisión multipunto del TX a los RX
- Direccionamiento extendido: para ambientes de operación con muchos nodos se implemento direccionamiento extendido a 10 bits.

I²C

I²C en pocas palabras...

- Ventajas:
 - Muy eficiente en comunicación de dispositivos (SOC) que son accedidos ocasionalmente.
 - Direccionamiento simple de implementar
 - El costo y la complejidad son fijos al aumentar el número de dispositivos.
- Desventajas:
 - La complejidad de interactuar con componentes de software es elevada vrs otros esquemas de comunicación serie.

¿Qué es SPI?

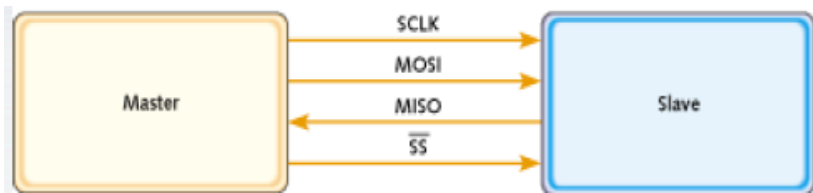
- Es un protocolo de comunicación serie que viene del ingles: "Serial Peripheral Interface"
- Creado por Motorola para su familia de microcontroladores MC68HCxx
- Velocidad de operación de varios Mbps (mejor rendimiento en velocidad que I2C)

Aplicaciones

- Comunicación con dispositivos de memoria EEPROM, Flash, RTC, etc.
- Excelente rendimiento en aplicaciones que generan cadenas de datos, ejemplo ADC.
- Comunicación Full duplex, mejora el rendimiento en comunicaciones con subsistemas dentro de la aplicación

SPI

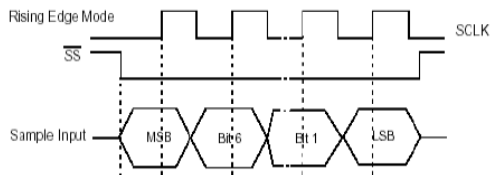
Configuración de un bus SPI



- Enlace serial síncrono funcionando en full duplex de punto a punto.
- Relación maestro - esclavo
- Dos señales para datos
 - MOSI – Master data Output, Slave data Input
 - MISO - Master data Input, Slave data Output
- Dos señales de control
 - Reloj
 - Selección de punto (Slave Select)

SPI

¿Cómo funciona SPI?

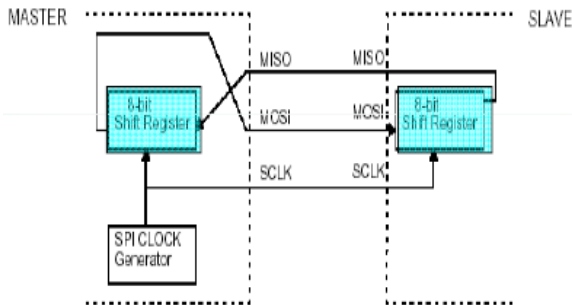


- Dos parámetros, Clock Polarity (CPOL) y Clock Phase (CPHA), determinan el flanco activo del reloj
- Para establecer la comunicación el maestro y el esclavo deben acordar los parámetros de enlace.

CPOL	CPHA	Flanco activo
0	0	Positivo
0	1	Negativo
1	0	Negativo
1	1	Positivo

Protocolo SPI

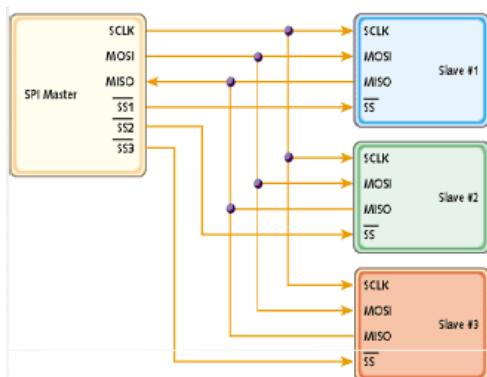
- La interfase SPI solamente define cuales son las líneas de comunicación y el flanco de reloj en que estas son activas.
- No existe control de flujo, ni ACK para confirmar la recepción de un dato.
- La implementación en hardware se hace con un registro de corrimiento.



SPI

SPI vs. I²C

- Para comunicaciones punto a punto, SPI es simple y eficiente
 - Mas efectividad del ancho de banda, y comunicación bidireccional
- SPI tiene limitaciones de escalamiento debido al control de acceso al bus individual por dispositivo (!SSX).



Resumen de comparación I²C vrs. SPI

- I²C y SPI proveen un buen soporte para comunicaciones con periféricos lentos que son accedidos intermitentemente, especialmente EEPROMS y RTC.
- I²C permite extensión de dispositivos conectados al bus sin incrementar complejidad del hardware para accederlos.
- SPI ofrece mejor desempeño en velocidad pero posee problemas de escalamiento.

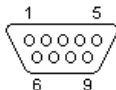
Comunicación Serial UART

Responde al estándar RS-232. Fue desarrollada en la década de los 60 para gobernar la interconexión de terminales y MODEM. Está patrocinada por la EIA (Asociación de Industrias Eléctricas).

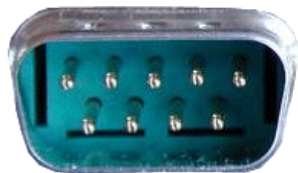
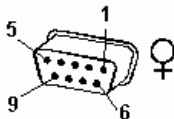
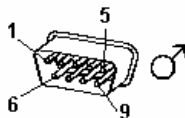
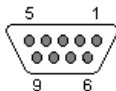
Es la forma más sencilla de comunicar un microcontrolador con una PC.

Conector RS232

Conector DB9 macho
Conector del PC



Conector DB9 hembra



Pinout de conector típico a la PC

Número de Pin	Nombre	Descripción
1	CD	(Carrier Detect) Entrada
2	RXD	(Receive Data) Entrada
3	TXD	(Transmit Data) Salida
4	DTR	(Data terminal Ready) Salida
5	GND	(Ground)
6	DSR	(Data Set Ready) Entrada
7	RTS	(Request To Send) Salida
8	CTS	(Clear To Send) Entrada
9	RI	(Ring Indicator) Entrada

Para comunicarse con la PC solo se necesitan 3

Para comunicarse con un microcontrolador bastan con 3 líneas (visto desde el PC):

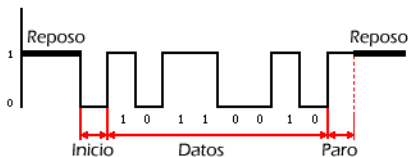
- 3, TXD, Transmisión de datos.
- 2, RXD, Recepción de datos.
- 5, GND, Masa.

La velocidad de transmisión es un dato importante y es la cantidad de información que es enviada por segundo. Se muestra en unidades de baudios (bits por segundo). En la PC suele utilizarse típicamente 4800 o 9600.

Comunicación del PIC con la PC

Enviar datos de forma serial mediante el protocolo RS232 de forma asíncrona, la cuál es la más usada en el PIC, consiste solamente en hacer que una salida esté a nivel alto o bajo en el momento adecuado.

Cuando no se transmite, la salida está a nivel alto, y cuando empieza el bit de inicio, esta pasa a ser de nivel bajo. A 4800 baudios la duración del bit debe ser de $1/4800 = 208$ microsegundos. 8 bits de datos siguen al bit de inicio, cada uno de un periodo. Un nivel alto de más de un periodo es un bit de parada. Debe tenerse en cuenta que primero se mandan los bits menos significativos o LSB (Least Significant Bit).



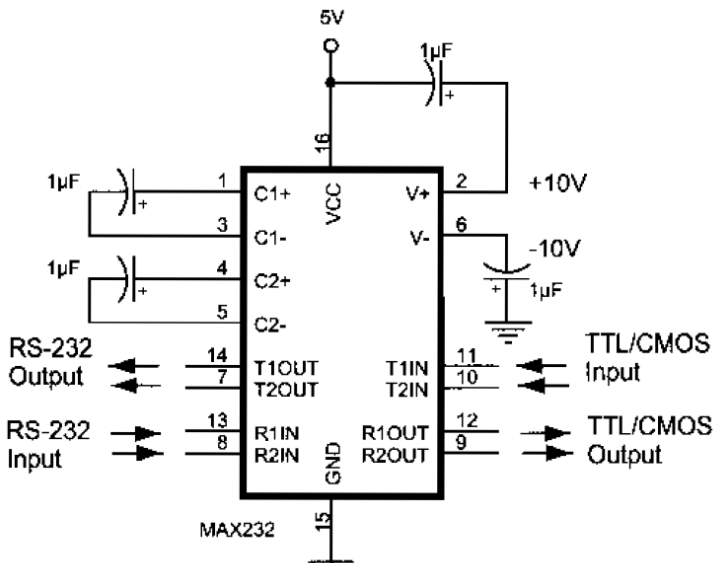
Comunicación del PIC con la PC

El protocolo RS232 utiliza lógica negativa, un nivel alto (1) es de -3V a -15 voltios, y un nivel bajo (0) es de +3V a +15 voltios. Usualmente los valores son +/-12 voltios.

Puesto que un PIC utiliza niveles TTL es necesario un conversor de niveles. El mas utilizado es el circuito integrado MAX232 de Dallas Semiconductor.

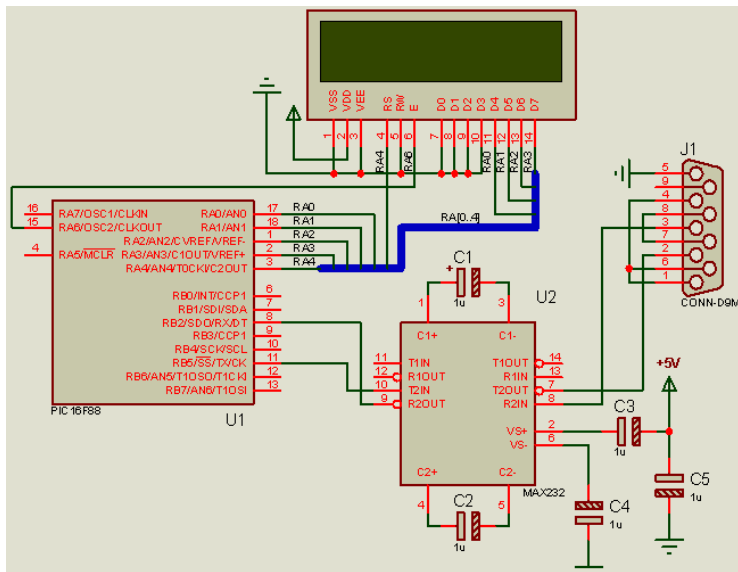
El MAX232 convierte los niveles RS232 a TTL y viceversa con sólo una fuente de 5 voltios. Contiene dos driver TTL-RS232 y dos driver RS232-TTL.

MAX232



UART RS232

Ejemplo de comunicación Serial PIC utilizando Proteus



Referencias



Carvajal J.

Introducción al diseño de sistemas electrónicos basados en microcontrolador

Presentación de clases Estructura de microprocesadores. Escuela de Electrónica. ITCR



Sin Nombre.

<http://programarpicenc.com/articulos/usart-uart-de-los-microcontroladores-pic-con-el-max232/>