



Practica 1 de laboratorio

Arquitectura del computador

Por:

José Rivero C.I: 28.492.353

Edwin Machado C.I: 30.532.641

Microcontroladores

Un microcontrolador es un pequeño dispositivo electrónico que integra en un solo chip un procesador central (CPU), memoria, periféricos de entrada/salida y, en algunos casos, componentes adicionales como temporizadores, contadores y convertidores analógico-digitales. Está diseñado para realizar tareas específicas en sistemas embebidos, donde se requiere control y procesamiento de datos en tiempo real.

Los microcontroladores se utilizan en una amplia variedad de aplicaciones, desde electrodomésticos y dispositivos médicos hasta sistemas de control industrial y automóviles. Su tamaño compacto y su capacidad para ejecutar programas almacenados en memoria lo hacen ideal para aplicaciones donde se necesita un control preciso y eficiente.

Estos dispositivos se programan para realizar funciones específicas, y su programación puede llevarse a cabo mediante lenguajes de programación de bajo nivel o entornos de desarrollo integrados (IDE, por sus siglas en inglés) que facilitan la creación de software para controlar el comportamiento del microcontrolador.

Familia de Microcontroladores Intel

Microcontrolador Intel 8051:

El microcontrolador Intel 8051 es un componente clave en la historia de los microcontroladores y ha sido significativo en el desarrollo de sistemas embebidos (sistema de computación especializado que está diseñado para realizar tareas específicas y está integrado como parte de un dispositivo más grande).

Algunas características que lo destacan incluyen:

- 1.-El 8051 sigue una arquitectura CISC, lo que significa que posee un conjunto de instrucciones más amplio en comparación con los microcontroladores RISC.
- 2.- El 8051 fue uno de los primeros microcontroladores en integrar en un solo chip una unidad de procesamiento central (CPU), memoria RAM, memoria de programa (ROM o EPROM), puertos de entrada/salida y otros periféricos básicos.
- 3.-Algunas versiones del 8051 incluyen un multiplicador y divisor hardware, lo que mejora significativamente el rendimiento en operaciones aritméticas complejas.

Microcontrolador Intel 8052:

El Intel 8052 fue una extensión del 8051 diseñada para abordar las demandas de aplicaciones más avanzadas y complejas. Aunque ya no es tan común como algunos microcontroladores más recientes, sigue siendo utilizado en algunas aplicaciones específicas, especialmente cuando se requieren las características adicionales que ofrece en comparación con el 8051 estándar.

Microcontrolador Intel 8031:

El Intel 8031 fue diseñado para aplicaciones que requerían mayores capacidades de memoria y puertos de entrada/salida en comparación con el 8051 estándar. Aunque ha sido superado por microcontroladores más modernos en muchas aplicaciones, todavía puede encontrarse en algunos sistemas heredados o en aplicaciones específicas donde sus características adicionales son beneficiosas.

Puertos en el Microcontrolador 8051

Un Puerto se refiere a un conjunto de pines o líneas de E/S (entrada/salida) que se agrupan para facilitar la manipulación y la conexión de señales. Estos puertos son interfaces de comunicación entre el microcontrolador y otros dispositivos, como sensores, actuadores, o incluso otros microcontroladores.

En el caso de los Microcontroladores Intel 8051, estos poseen 4 puertos:

Puerto 0:

El puerto 0 es un puerto de propósito dual en las terminales 32-39 del circuito integrado 8051. Se utiliza como un puerto de E/S de propósito general en diseños que requieren un mínimo de componentes. Este puerto se puede convertir en un bus de direcciones y datos multiplexados en diseños más complejos que requieran de memoria externa.

Puerto 1:

El puerto 1 es un puerto dedicado de E/S en las terminales 1-8. Las terminales, designadas como P1.0, P1.1, P1.2, etc., están disponibles para utilizarse como interfaces para dispositivos externos, en caso de requerirse. Ninguna de las terminales del puerto 1 tiene otra función asignada, por lo tanto, sólo se utilizan como interfaces para dispositivos externos. Los circuitos integrados 8032/8052 son la excepción, pues utilizan las terminales P1.0 o P1.1 ya sea como líneas de E/S o como entradas externas del tercer temporizador.

Puerto 2:

El puerto 2 (terminales 21-28) es un puerto de propósito dual que sirve como E/S de propósito general, o como el byte superior del bus de direcciones en diseños

que utilizan memoria externa para código o más de 256 bytes de memoria externa para datos. (Consulte la sección 2.7 Memoria externa).

Puerto 3:

El puerto 3 es un puerto de propósito dual en las terminales 10-17. Se puede utilizar como E/S de propósito general, pero también cumple múltiples funciones ya que sus terminales tienen un propósito alterno relacionado con las características especiales del 8051. En la tabla 2-2 se sintetiza el propósito alterno de los puertos 3 y 1.

Temporizadores en el Microcontrolador 8051

Un temporizador es un componente que cuenta el tiempo y se utiliza para realizar diversas funciones temporales en un sistema embebido. Los temporizadores son esenciales para controlar eventos, medir intervalos de tiempo, generar pulsos precisos y realizar tareas que dependen del tiempo, como la generación de señales PWM (Modulación por Ancho de Pulso), el control de intervalos de muestreo en sistemas de adquisición de datos, entre otros.

Un temporizador consta de una serie de flip-flops de división entre 2, los cuales reciben una señal de entrada como su fuente de reloj. El reloj se aplica al primer flip-flop, el cual divide la frecuencia del reloj entre 2. La salida del primer flip-flop se aplica a la entrada de reloj del segundo flip-flop, que también divide la frecuencia entre 2, y así sucesivamente. Un temporizador con un número n de etapas divide la frecuencia del reloj de entrada entre 2^n , ya que cada etapa sucesiva divide entre 2. La salida de la última etapa se aplica a la entrada de reloj de un flip-flop de desbordamiento del temporizador, o flag, cuyo estado puede ser verificado mediante el software y su establecimiento en 1 puede generar una interrupción.

El 8051 tiene dos temporizadores de 16 bits, cada uno con cuatro modos de operación. Un tercer temporizador de 16 bits con tres modos de operación se ha añadido al 8052. Los temporizadores se utilizan para (a) la temporización de intervalos, (b) el conteo de eventos, o (c) la generación de la tasa de transmisión y recepción en baudios para el puerto serial incorporado. Cada temporizador es de 16 bits, por lo que la decimosexta o última etapa divide la frecuencia del reloj de entrada entre $2^{16} = 65.536$.

Interrupciones en el Microcontrolador 8051

Una interrupción es la ocurrencia de una condición (o evento) que ocasiona la suspensión temporal de un programa mientras que otro programa se encarga de servir a dicha condición.

Las interrupciones cumplen una función importante en el diseño y la implementación de las aplicaciones con microcontroladores. Las interrupciones permiten que un sistema pueda responder a un evento en forma asíncrona y se encargue del evento mientras se ejecuta otro programa. Un sistema controlado mediante interrupciones nos da la falsa percepción de que está realizando muchas cosas en forma simultánea.

Por supuesto que la CPU no puede ejecutar más de una instrucción al mismo tiempo; pero sí puede suspender temporalmente la ejecución de un programa, ejecutar otro, y después regresar al primer programa. En cierta manera, esto es como una subrutina. La CPU ejecuta otro programa (la subrutina) y después regresa al programa original. La diferencia está en que, en un sistema controlado mediante interrupciones, la interrupción es la respuesta a un “evento” que ocurre de manera asíncrona con el programa principal. En otras palabras, no sabemos cuándo se interrumpirá el programa principal.

INTERRUPCIONES DEL TEMPORIZADOR

Las interrupciones del temporizador ocurren cuando la bandera de desbordamiento del temporizador, TFX, se activa después de un desbordamiento de los registros del temporizador, THx/TLx. La bandera de desbordamiento del temporizador, TFX, se borra de manera automática mediante el hardware cuando el 8051 procede a dar servicio a la interrupción.

INTERRUPCIONES DEL PUERTO SERIAL

Las interrupciones del puerto serial ocurren cuando se activa ya sea la bandera de interrupción de transmisión (TI) o la bandera de interrupción de recepción (RI). Una interrupción de transmisión ocurre cuando la transmisión del carácter escrito previamente al SBUF ha terminado. Una interrupción de recepción ocurre cuando un carácter se ha recibido por completo y está esperando en el SBUF para ser leído.

INTERRUPCIONES EXTERNAS

Las interrupciones externas ocurren como resultado de un borde de bajo nivel o negativo en las terminales o del 8051. Éstas son las funciones alternas de los bits P3.2 (terminal 12) y P3.3 (terminal 13) del puerto 3, respectivamente.

Las banderas que generan estas interrupciones son los bits IE0 e IE1 en el registro TCON. Cuando se genera una interrupción externa, la bandera que la generó se borra mediante el hardware cuando se vectoriza a la ISR sólo si la interrupción fue activada mediante una transición. Si la interrupción fue activada debido a un bajo nivel, la fuente externa de petición, y no el hardware, controla el nivel de la bandera de petición.

PRECIOS DE LOS TEMPORIZADORES Y PUERTOS

Microcontrolador Intel 8051:

Tiene un precio aproximado de 0.30 Dolares.

Enlace:

https://www.alibaba.com/product-detail/8051-Integrated-Circuit-Electric-Supplies-Component_1600987263866.html?spm=a2700.7735675.0.0.2592u9cGu9cGmm&s=p

Microcontrolador Intel 8052:

Que se necesita para programar un microcontrolador 8051

Para programar un microcontrolador de la familia Intel 8051, se necesitan algunos elementos básicos. Aquí hay una lista general de lo que podrías necesitar:

1. Microcontrolador Intel 8051:
 - El microcontrolador en sí es el componente principal. Puedes elegir entre diferentes variantes dentro de la familia 8051 según tus necesidades específicas.
2. Kit de Desarrollo o Placa de Evaluación:
 - Para facilitar el desarrollo y la programación, muchos optan por utilizar un kit de desarrollo o una placa de evaluación que ya incluya el microcontrolador, así como otros componentes y periféricos.
3. Herramienta de Programación:

- Necesitarás una herramienta de programación que te permita cargar el código compilado en el microcontrolador. Esto puede ser un programador de microcontroladores específico para la familia 8051.

4. Entorno de Desarrollo (IDE):

- Un entorno de desarrollo integrado (IDE) te permitirá escribir, compilar y depurar tu código. Puedes utilizar un IDE específico para la familia 8051, como Keil u otros que admitan esta arquitectura.

5. Compilador C o Ensamblador:

- Si estás programando en C, necesitarás un compilador C que sea compatible con la arquitectura 8051. Para lenguaje ensamblador, necesitarás un ensamblador compatible.

PLC

Un PLC, que significa "Controlador Lógico Programable" (por sus siglas en inglés, PLC: Programmable Logic Controller), es un dispositivo utilizado en automatización industrial y sistemas de control para monitorear y controlar equipos y procesos en entornos industriales. Los PLCs son ampliamente utilizados en la industria para realizar tareas de control y automatización de maquinaria, procesos de fabricación y sistemas industriales.

Características principales de un PLC:

- 1.- Los PLCs son programables, lo que significa que su comportamiento puede ser definido y modificado mediante software.
- 2.- Los PLCs tienen entradas y salidas digitales y, en algunos casos, también pueden tener entradas y salidas analógicas.
- 3.- Los PLCs pueden comunicarse con otros dispositivos y sistemas a través de interfaces de comunicación, como puertos serie, Ethernet industrial, o incluso redes de campo bus.
- 4.- Muchos PLCs operan en tiempo real, lo que significa que responden a las entradas y generan salidas en un intervalo de tiempo definido y predecible. Esto es crucial en aplicaciones donde el tiempo de respuesta es crítico.
- 5.- Los PLCs están diseñados para ser robustos y confiables en entornos industriales adversos, donde pueden estar expuestos a vibraciones, fluctuaciones de temperatura y otros factores desafiantes.

Que se necesita para Programar un PLC

Programar un PLC (Controlador Lógico Programable) implica varios pasos y requerimientos.

Aquí hay una guía general de lo que se necesita para programar un PLC:

1.Kit de Desarrollo o Placa de Evaluación (opcional):

Puede ser útil tener un kit de desarrollo que incluya el PLC, entradas/salidas, y otros componentes necesarios para el desarrollo y las pruebas.

2.Entorno de Desarrollo (Software de Programación):

El software de programación proporcionado por el fabricante del PLC, como TIA Portal para Siemens, RSLogix para Allen-Bradley, o el software específico según el fabricante y modelo del PLC.

3.Cable de Programación:

Para conectar el PLC a la computadora y permitir la transferencia de programas.

4.Computadora:

Una computadora con el software de programación instalado.

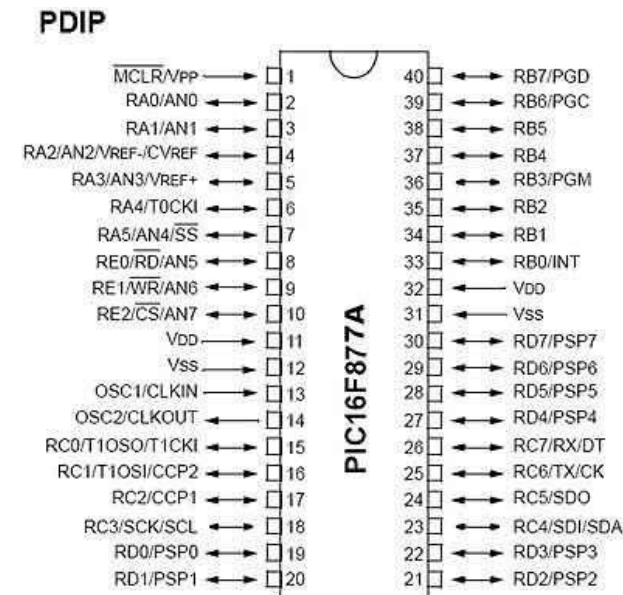
5.Conexión con el PLC:

Dependiendo del PLC, esto puede implicar un cable de programación USB, RS-232, Ethernet, u otras interfaces de comunicación.

Tabla de Pines del PLC:

Pin	Función
---	-----
1	Alimentación (VCC)
2	Alimentación (GND)
3	Entrada Digital 1
4	Entrada Digital 2
5	Salida Digital 1
6	Salida Digital 2
7	Entrada Analógica 1
8	Entrada Analógica 2
9	Salida Analógica 1
10	Salida Analógica 2
11	Comunicación Serie (RX)
12	Comunicación Serie (TX)
13	Común de Entradas Analógicas
14	Común de Salidas Analógicas
15	Tierra (GND)
16	Relé de Salida 1
17	Relé de Salida 2
18	Entrada de Alta Velocidad 1
19	Entrada de Alta Velocidad 2
20	Salida de Alta Velocidad 1

Diagrama de Pines de un PLC:



Costo de un PLC:

El precio de un PLC puede variar bastante dependiendo de su modelo específico. Pero su precio más bajo este alrededor de 20 dólares.

https://spanish.alibaba.com/p-detail/Plc-1600490326431.html?spm=a2700.galleryofferlist.p_offer.d_image.4ebe77a5bRYWzi&s=p

ARDUINO

Arduino es una plataforma de electrónica de código abierto que combina hardware y software libre, diseñada para ser flexible y accesible a creadores y desarrolladores. Su enfoque en hardware libre permite que sus especificaciones sean públicas, permitiendo a otros replicar y crear sus propias placas basadas en su diseño. El software libre de Arduino, como el Arduino IDE, ofrece un entorno de programación abierto para crear diversas aplicaciones para estas placas.

Nacido en 2003 como un proyecto de estudiantes en Italia, Arduino surgió para proporcionar una alternativa económica a las placas de electrónica existentes en ese momento. Ofrece una solución asequible para aquellos interesados en la electrónica y la programación, promoviendo la filosofía de aprender haciendo, fomentando la experimentación y la práctica como herramientas clave para el aprendizaje.

Como programarlo

El Arduino IDE (Integrated Development Environment) es fundamental para programar la placa Arduino. Proporciona a los programadores una interfaz amigable para escribir, editar, depurar y cargar sus programas, conocidos como "sketches", en la placa Arduino. Esta accesibilidad y facilidad de uso son clave para el éxito de Arduino, ya que hace que la programación de dispositivos electrónicos sea más accesible incluso para principiantes en la materia. El IDE simplifica el proceso de desarrollo al ofrecer herramientas intuitivas que facilitan la creación y ejecución de programas para la placa Arduino.

Que se necesita:

- 1.Arduino IDE: el entorno oficial para escribir, compilar y cargar programas en la placa.
- 2.Lenguaje de programación: basado en C/C++, fundamental para programar en Arduino.
- 3.Librerías: disponibles para añadir funciones a proyectos, simplificando el proceso de programación.
- 4.Componentes electrónicos: como resistencias, sensores, motores, esenciales para construir proyectos.
- 5.Breadboard: una herramienta para prototipar circuitos electrónicos de manera rápida y eficiente, importante conocer sus conexiones para un uso efectivo.

Donde comprarlo

https://spanish.alibaba.com/p-detail/ACEBOTT-1600986720139.html?spm=a2700.galleryofferlist.p_offer.d_title.5d7039b037ptKv&s=p

Raspberry Pi

La Raspberry Pi es una computadora compacta y de bajo costo, con dimensiones similares a las de una tarjeta de crédito. Se puede conectar a un monitor o televisor, y utilizar con un teclado y ratón estándar. Funciona con un sistema operativo Linux que permite a personas de todas las edades explorar el mundo de la computación y aprender a programar en lenguajes como Scratch y Python. Esta pequeña computadora es capaz de realizar la mayoría de las tareas habituales de un ordenador de escritorio, desde navegación en internet y reproducción de videos en alta resolución hasta manejo de documentos de ofimática y ejecución de juegos.

Además, la Raspberry Pi tiene la capacidad de interactuar con el entorno exterior, siendo utilizable en una amplia gama de proyectos digitales, desde reproductores de música y vídeo hasta estaciones meteorológicas o dispositivos con cámaras infrarrojas para observar aves. Su versatilidad la hace adecuada para niños y adultos de todo el mundo que deseen adentrarse en la programación y comprender el funcionamiento de las computadoras.

Pasos para programar una Raspberry Pi:

Preparación inicial:

Sistema Operativo: Instala un sistema operativo compatible en la tarjeta microSD de la Raspberry Pi. Puedes usar sistemas como Raspbian, Raspberry Pi OS, o cualquier otro compatible con la placa.

Conexión: Conecta la Raspberry Pi a una pantalla, teclado y ratón, y alimenta la placa con un adaptador de corriente.

Configuración del entorno:

Acceso remoto: Configura la Raspberry Pi para acceder a ella remotamente a través de SSH (Secure Shell) o VNC (Virtual Network Computing).

Actualizaciones: Realiza actualizaciones del sistema operativo y del software de la Raspberry Pi para asegurarte de tener las últimas versiones y correcciones de seguridad.

Entorno de Desarrollo:

Lenguajes de Programación: Puedes programar en varios lenguajes, siendo Python uno de los más populares debido a su simplicidad y versatilidad.

IDE o Editor de Código: Utiliza un entorno de desarrollo integrado (IDE) como Thonny, o un editor de código como Visual Studio Code, Atom o Nano para escribir y ejecutar tu código.

Ejecución de programas:

Escribir código: Utiliza el IDE o editor de código para escribir programas en el lenguaje elegido.

Ejecución: Ejecuta los programas en la Raspberry Pi para realizar diversas tareas, desde controlar hardware hasta ejecutar aplicaciones.

Donde comprarlo:

https://www.alibaba.com/product-detail/Original-Raspberry-Pi-Model-Pi4-4b_1600479423481.html?spm=a2700.7735675.0.0.17f3HQMWHQMW4D&s=p