**U**

**P**

**T**



**Universidad politÉcnica de Tulancingo**

**CARRERA:**

**INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**ASIGNATURA:**

**Seminario de proyectos**

**ALUMNO:**

**José Manuel Amador Cruz**

**David Guajardo Gregorio**

**Ángel Salas Sánchez**

**DOCENTE:**

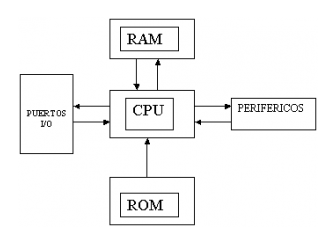
**Ing. Arturo Negrete Medellín**

**Mayo- agosto 2020**

**SELECCIÓN DEL MICROCONTROLADOR**

**Microcontrolador**

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.



Toda microcomputadora requiere de un programa para que realice una función específica. Este se almacena normalmente en la memoria ROM. No esta de mas mencionar que sin un programa, los microcontroladores carecen de utilidad. El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario le escribe, es por esto que la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. El carácter programable de los microcontroladores simplifican el diseño de circuitos electrónicos. Permiten modularidad y flexibilidad, ya que un mismo circuito se puede utilizar para que realice diferentes funciones con solo cambiar el programa del microcontrolador. Las aplicaciones de los microcontroladores son vastas, se puede decir que solo están limitadas por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y el automatismo, en la industria del entretenimiento, en las telecomunicaciones, en la instrumentación, en el hogar, en la industria automotriz, etc.

**PROGRAMA DEL MICROCONTROLADOR**

Los microcontroladores están diseñados para interpretar y procesar datos e instrucciones en forma binaria. Patrones de 1’s y 0’s conforman el lenguaje maquina de los microcontroladores, y es lo único que son capaces de entender. Estos 1’s y 0’s representan la unidad mínima de información, conocida como bit, ya que solo puede adoptar uno de dos valores posibles: 0 ó 1.

La representación de datos, instrucciones y señales en forma de bits resulta dificultosa y tediosa para aquellas personas que no estén familiarizadas con el sistema de numeración binario. Aún para los usuarios expertos no resulta tan evidente la interpretación de instrucciones en forma binaria o lenguaje máquina (el lenguaje maquina se le conoce también como lenguaje de bajo nivel debido a que las instrucciones no son propias del lenguaje humano). Es por esto que la programación comúnmente se lleva a cabo en un lenguaje de alto nivel, es decir, un lenguaje que utilice frases o palabras semejantes o propias del lenguaje humano. Las sentencias de los lenguajes de alto nivel facilitan enormemente la programación ya que son familiares a nuestra manera de comunicarnos. Lenguajes como el C o BASIC son comúnmente utilizados en la programación de microcontroladores.Otro tipo de lenguaje mas especializado es el lenguaje ensamblador. El lenguaje ensamblador es una lista con un limitado numero instrucciones a los cuales puede responder un microcontrolador. Estas instrucciones son palabras o abreviaciones que representan las instrucciones en lenguaje maquina del microcontrolador.

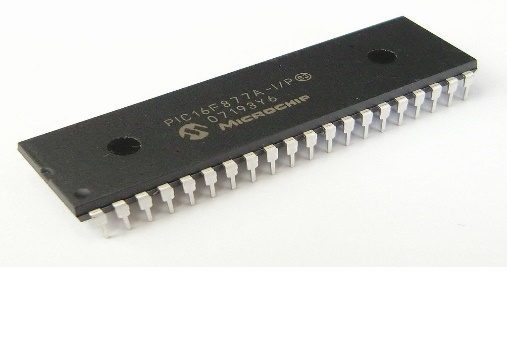
Las instrucciones en lenguaje ensamblador, también conocidas como nemotécnicos, son fáciles de entender y permiten operar directamente con los registros de memoria así como con las instrucciones intrínsecas del microcontrolador. Es por esto que el lenguaje ensamblador es sin lugar a dudas el lenguaje por excelencia en la programación de microcontroladores, ya que permite hacer un uso eficiente de la memoria y minimizar el tiempo de ejecución de un programa.

Cualquiera que sea el lenguaje que se utilice en la programación de microcontroladores, es de lo mas recomendable profundizar en su arquitectura interna, ya que con este conocimiento se pueden aprovechar mas y mejor las capacidades de un microcontrolador dado.

**MICROCONTROLADORES**

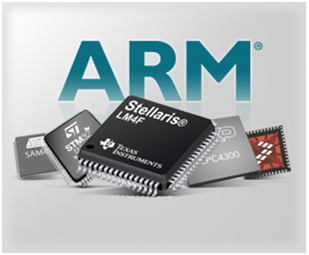
**PIC**

El PIC es una familia de microcontroladores de Microchip. Son muy populares y vienen en una amplia gama de opciones. El número de pines, tipo de encapsulado y periféricos, es lo que ofrecen en una variedad de combinaciones casi infinita.



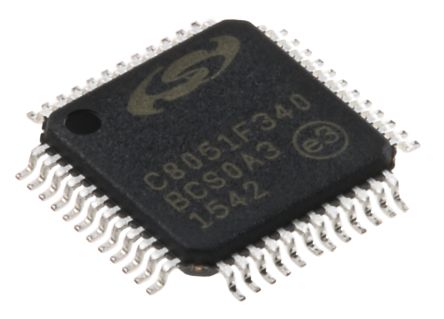
**ARM Cortex – M.**

La serie ARM Cortex M de 32 -bit es uno de los cores más ​​utilizados en la actualidad. ARM no está fabricando y vendiendo microcontroladores, sino que concede licencias de su arquitectura a otros fabricantes de chips. Muchas compañías ofrecen microcontroladores Cortex M incluyendo ST Microelectronics, Freescale Semiconductor, Silicon Labs, Texas Instruments y Atmel. Los microcontroladores de la serie Cortex M son de bajo costo, potentes y ampliamente utilizados.



**8051**

El microcontrolador 8051 de 8 bits fue desarrollado por Intel en 1980. Es el núcleo del microcontrolador más antiguo y el más utilizado en la actualidad. El 8051 ya está disponible en las versiones modernas y mejoradas, y es vendido por al menos 8 diferentes fabricantes de semiconductores. Por ejemplo, el popular chip Bluetooth de bajo consumo de energía a partir del CRS (CSR101x) utiliza un núcleo 8051.



**Atmel AVR**

La línea del microcontrolador AVR de Atmel es mejor conocido por ser el cerebro de la mayoría de las versiones del Arduino. Así que para muchos fabricantes es fácil realizar una transición de un Arduino a un microcontrolador Atmel AVR. Aunque por lo general es posible conseguir algunos de los otros microcotroladores con características similares, o mejores, por varios dólares más barato.



**RASPBERRY PI**

Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, con el objetivo de estimular la enseñanza de informática en las escuelas.3​4​5​ El modelo original se convirtió en más popular de lo que se esperaba6​, hasta incluso vendiéndose fuera del mercado objetivo para usos como robótica.



**COMPARACION**

|  |  |  |  |
| --- | --- | --- | --- |
| MICROCONTROLADOR | CARACTERIRTICAS | VENTAJAS | DESVENTAJAS |
| PIC | \*Núcleos de CPU de 8/16 bits con  \*Arquitectura Harvard modificada  \*Memoria Flash y ROM disponible desde 256 bytes a 256 kilobytes  \*Puertos de entrada/salida (típicamente 0 a 5.5 voltios)  \*Temporizadores de 8/16/32 bits  \*Conversores analógico /digital de 8-10-12 bits  \*Módulos de captura y comparación PWM  \*Controladores LCD  \*Periférico MSSP para comunicaciones I²C, SPI, y I²S  \*Memoria EEPROM interna con duración de hasta un millón de ciclos de lectura/escritura  \*Soporte de interfaz USB  \*Soporte de controlador Ethernet | \* Los microcontroladores PIC son muy fácil de grabar.  \*permite controlar, programar y soncronizar tareas electrónicas a través del tiempo simplemente realizando una correcta programación.  \*En el mercado existen varios softwares que nos ayudan a programar un microcontrolador de este tipo  \* Existe una gran diversidad de microcontroladores PIC en el mercado de Microchip y ésta también es una gran ventaja, ya que podemos elegir entre diversas características que uno no tiene pero otro si, | \*se necesitan llamar a muchas instrucciones para realizar una tarea en particular. Esto siempre y cuando el proyecto sea complejo.  \*los PIC no son tan baratos como uno los puede esperar. |
| ARM Cortex – M. | \*Núcleo ARM Cortex-M0 de 50 MHz.  \*Flash de hasta 32 KB.  Memoria SRAM de hasta 8 KB.  \*Periféricos seriales: I2C Fast-mode Plus, dos SPI, UART.  \*ADC de 10 bits y 8 canales.  \*Nivel de consumo de energía activo más bajo (solo 130 μA/MHz)  \*GPIO de alta velocidad de hasta 42 clavijas. | \*ARM Cortex-M0, el núcleo Cortex-M más pequeño y eficiente energéticamente  \*es la alternativa más reciente de bajo costo a los diseños de 8 y 16 bits.  \*La serie LPC1100L ofrece el mejor nivel de costos y consumo de energía de la industria e incluye la serie LPC1102,  \*El microcontrolador ARM de 32 bits más pequeño del mundo. La serie LPC11C00  \*Incluye el primer microcontrolador CAN y transceptor con controladores CANopen; mientras que la serie LPC11U00  \*ofrece un rendimiento de USB sólido a bajo precio. La serie LPC1200, la última generación de Cortex-M0 de NXP.  \*proporciona un alto nivel de eficiencia energética e incluye características especiales para el control industrial. | \*precios elevados  \*programación compleja  \*utilización para proyectos robustos |
| 8051 | \*Arquitectura basada en el 80C51 con CPU de 8 bits.  \*Procesador booleano con soporte de operación sobre bits.  \*Memoria de programa integrada de 8k × 8 EPROM (caso 87C52).  \*Sin memoria de programa interna ROMless (caso 80C32).  \*Memoria RAM interna de 256 × 8 .  \*Tres contadores  \*temporizadores de 16-bit (counter/timers).  \*Un canal serie asíncrono Full duplex para las comunicaciones RS-232.  \*Capacidad de direccionamiento de memoria 64k ROM y 64k RAM.  \*Modos de funcionamiento en bajo consumo (Idle mode Power-down mode).  \*Compatible con las tecnologías digitales CMOS y TTL.  \*Frecuencias de trabajo de reloj comprendida ente 3,5 y 33MHz.  Oscilador interno.  \*6 fuentes de interrupción con distintos niveles de prioridad.  \*2 interrupciones externas.  \*3 interrupciones para los contadores/temporizadores (timers).  \*1 interrupción para el puerto serie. | \*Bajo Coste. Simplificación de stock  \*reducido tamaño placa, simplificación fabricación.  \*Alto grado de integración/Simplificación de periféricos.  \*Disminuye el hardware, placas menos complejas,  \*aumenta la fiabilidad al reducirse el número de componentes.  \*Aumento de capacidad y velocidad de ejecución. Disminución consumo. Utilización de tecnologías MOS, CMOS o HCMOS | \*Necesidad de herramientas y sistemas de desarrollo que incrementan el tiempo y el coste de diseño.  \*Programación: necesidad de programación por máscaras en grandes series y de diferentes tipos en pequeñas series. |
| Atmel AVR | \*Fabricante: Atmel (Microchip).  \*Voltaje de operación: 1.8 a 5.5 VDC.  \*Arquitectura de CPU: 8 bit AVR  \*Memoria flash: 32 KB.  \*Memoria RAM: 2 KB.  \*EEPROM: 2 KB.  \*Frecuencia de operación: 20 Mhz.  \*Pines de IO: 23  \*Canales ADC: 10.  \*Interfaces: UART, TWI, SPI.  \*Temperatura de Operación: -40° a 85° C | Las capacidades de facilidad de uso, alto rendimiento, consumo eficiente de energía y estrecha densidad de código de la arquitectura AVR la convierten en una propuesta muy seria para ingenieros de diseño que trabajan en una amplia gama de aplicaciones y mercados. Son la opción ideal para tecnología inalámbrica y las amplias oportunidades que se abren con la Internet de las cosas. Hay una amplitud y gama de MCU de AVR de 8 bits disponible. Los últimos 15 años ciertamente han traído intensa actividad para la AVR | \*para aplicaciones poco complejas  \*normalmente utilizado por principiantes y amateurs  \*para mejor funcionamiento se deben conectar complementos aumentando la arquitectura haciendo la mas complejas por ejemplo conectar un cristal externo y no usar el oscilador interno. |
| RASPBERRY PI | \*CPU + GPU: Broadcom BCM2837B0, Cortex-A53  \*(ARMv8) 64-bit SoC @ 1.4GHz  \*RAM: 1GB LPDDR2 SDRAM  \*Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE  \*Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)  \*GPIO de 40 pines  HDMI  \*4 puertos USB 2.0  \*Puerto CSI para conectar una cámara.  \*Puerto DSI para conectar una pantalla táctil  \*Salida de audio estéreo y vídeo compuesto  Micro-SD  \*Power-over-Ethernet (PoE) | ofrecer una instalación de hardware lista para usarse que libera al usuario de la tediosa compilación de la base técnica necesaria para realizar sus proyectos DIY.  \* Raspberry Pi traen consigo todos los componentes para que el miniordenador funcione de forma independiente.  \* la mayoría de modelos son compatibles con cualquier red (Ethernet, redes inalámbricas o ambas).  \* el Raspberry Pi no solo puede servir como plataforma de programación, sino que también se utilizapara operar aplicaciones como nubes privadas, servidores web o servidores VPN. | la instalación requiere de mucha paciencia, pues no solo se tiene que seleccionar y descargar el sistema operativo, sino guardarlo en una tarjeta de memoria SD adecuada. La consiguiente instalación y configuración del software del sistema no es tampoco tan sencilla como cabría esperar del entorno Raspberry. |

**CONCLUSION**

**En el mercado existen diversos microcontroladores, que se pueden utilizar en proyectos de innovación, pero para nuestro caso y en base al estudio realizado llegamos a la conclusión que la mejor opción es Atmel en este caso en la placa Arduino, ya que el proyecto no es tan complejo como para utilizar una rasperry pi, ni los datos de procesamiento son tantos, ni se requierer de tantos puestos de entrada y salida, y aunque es barato comple con los requerimientos ocupados para nuestro proyecto, sin mencionar que requiere de la programación mas sencilla entre los microcontroladores.**