

Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

Microcontrolador vs Microprocesador

Diferencia principal entre un microcontrolador (Arduino) y un microprocesador (Raspberry Pi) son las capacidades de entradas y salidas, así como el rendimiento de la CPU.

Analogía: Arduino es un Autómata programable, Raspberry Pi es un Ordenador, así que a la hora de decidirse que utilizar para un proyecto pensar que usaríamos un autómata o un Ordenador.

Un resumen de como funciona una MCU y como agregar un programa:

<http://www.electronicaestudio.com/microcontrolador.htm> que es diferente a como funciona un microprocesador como los que tenemos en nuestro ordenador o portátil.

Para programación en tiempo real el HW a utilizar es el Arduino, para programación intensiva con gran cantidad de datos usaríamos una Raspberry Pi o un PC.

En un proyecto grande la elección es usar ambos, cada uno en la tarea que mejor hace. Por ejemplo, la recolección de datos, supervisión del entorno, envío de alarmas, accionar motores, etc.. lo dejaremos para el arduino, el tratamiento de los datos recogidos, el interfaz gráfico de usuario, envío de correos, etc... lo dejaremos para un ordenador o una raspberry pi o similar.

Diferencias entre el microprocesador y el microcontrolador, características al usarlos en la implementación de sistemas digitales programables:

- CPU
- Memorias RAM y ROM
- Velocidad de Operación
- Tamaño
- Costes
- Interferencias (ruido)
- Tiempo de desarrollo

El uso de una u otra tecnología depende del fin que se espera, pues debido a sus características propias, los microcontroladores y los microprocesadores pueden adquirir variados y diferentes espacios de implementación, por ejemplo, los microprocesadores se han desarrollado fundamentalmente orientados al mercado de los ordenadores personales y las estaciones de trabajo, pues allí se requiere una elevada potencia de cálculo, el manejo de gran cantidad de memoria y una gran velocidad de procesamiento. Mientras que los microcontroladores están concebidos fundamentalmente para ser utilizados en aplicaciones puntuales, es decir, aplicaciones donde el microcontrolador debe realizar un pequeño número de tareas, al menor costo posible. En estas aplicaciones el microcontrolador ejecuta un programa almacenado permanentemente en su memoria, el cual trabaja con algunos datos almacenados temporalmente e interactúa con el exterior a través de las líneas de entrada y salida de que dispone.

| | Microprocesadores | Microcontroladores |
|------------------------|--|---|
| CPU | El microprocesador tiene mucha más potencia de cálculo, por lo cual solamente realiza sus funciones con lo que tiene (datos) y su algoritmo o programa establecida. | Es una de sus partes principales, la cual se encarga de dirigir sus operaciones. |
| Memorias RAM y ROM | Son dispositivos externos que lo complementan para su óptimo funcionamiento. | Las incluye en un solo circuito integrado. |
| Velocidad de Operación | Rápida | Lenta en comparación con la de un microprocesador |
| Tamaño | La configuración mínima básica de un Microprocesador está constituida por un Microprocesador, una memoria RAM, una memoria ROM, un decodificador de direcciones, lo cual lo convierte en un circuito bastante engorroso. | El Microcontrolador incluye todo estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores, como por ejemplo, la disminución en el tamaño del circuito impreso por la reducción de los circuitos externos. |
| Costos | Para el Microprocesador, el costo es muy alto en la actualidad. | El costo para un sistema basado en Microcontrolador es mucho menor. |
| Interferencias | Son más susceptibles a la interferencia electromagnética debido a su tamaño y a su cableado externo que lo hace más propenso al ruido. | El alto nivel de integración reduce los niveles de interferencia electromagnética |
| Tiempo de desarrollo | El tiempo de desarrollo de un microprocesador es lento. | Por el contrario, el de un microcontrolador es rápido. |

Un buen curso sobre microcontroladores es accesible desde

http://www.itescam.edu.mx/portal/asignatura.php?clave_asig=MTF-1021&carrera=IMCT-2010-229&id_d=206.

Se trata de una asignatura de microcontroladores.

Más información sobre microcontroladores: <https://sites.google.com/site/electronicsscience20/Micro/pic-assembler/2-microcontroladores>

Raspberry Pi es un ordenador de placa reducida o (placa única) (SBC) de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi. El diseño incluye un System-on-a-chip Broadcom BCM2837, que contiene un procesador central (CPU) ARM1176JZF-S a 1.2 GHz quad-core ARMv8, un procesador gráfico (GPU) VideoCore IV, y 1GB de memoria RAM.

Los sistemas operativos soportados son distribuciones Linux para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora)

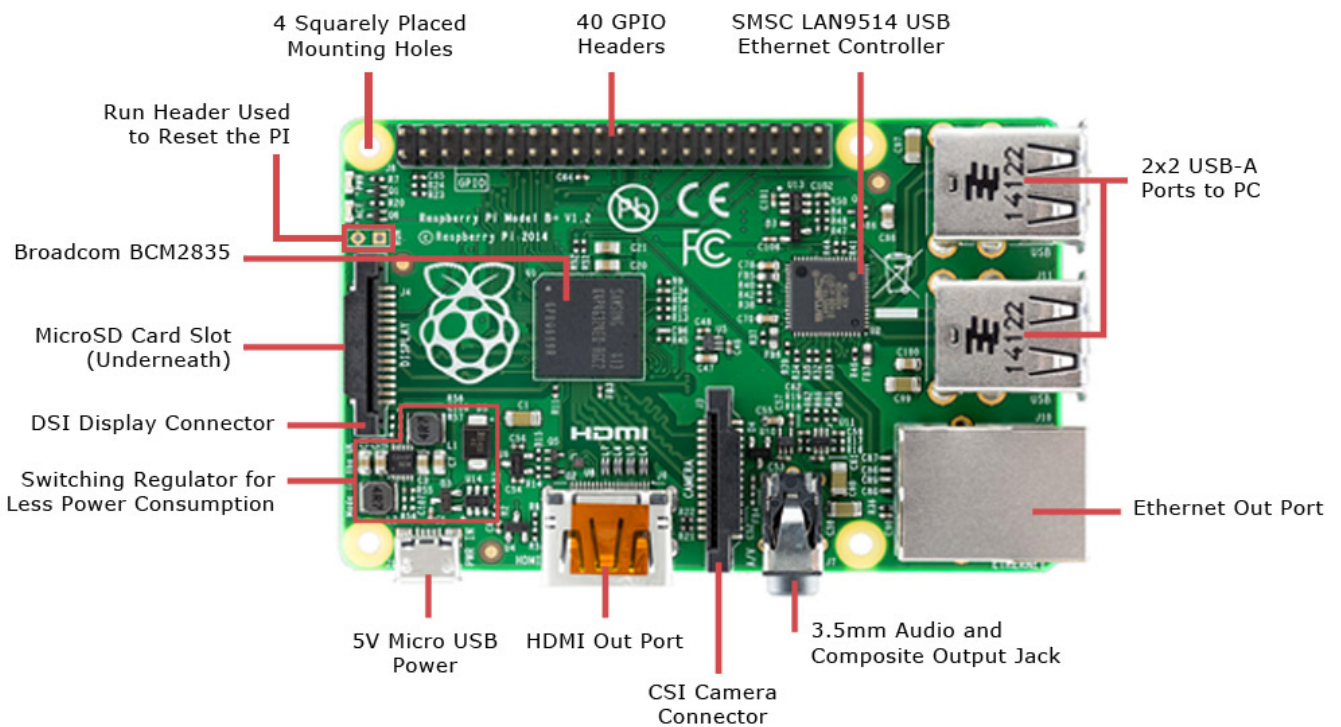
Web principal: <http://www.raspberrypi.org/>

Especificaciones técnicas: http://es.wikipedia.org/wiki/Raspberry_Pi#Especificaciones_t.C3.A9cnicas

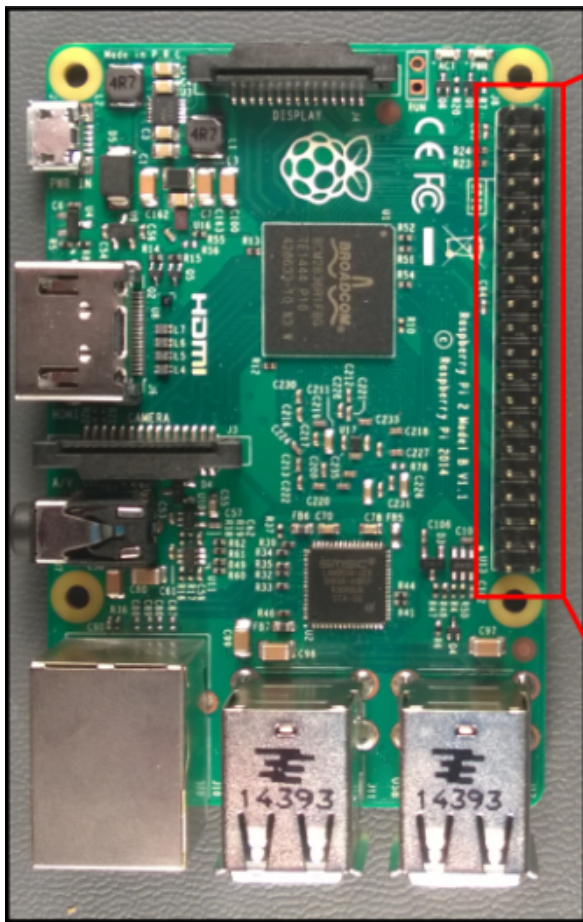
El System-on-a-chip Broadcom BCM2835: <http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

Raspberry Pi:

GPIO Pinout Diagram



GPIO:

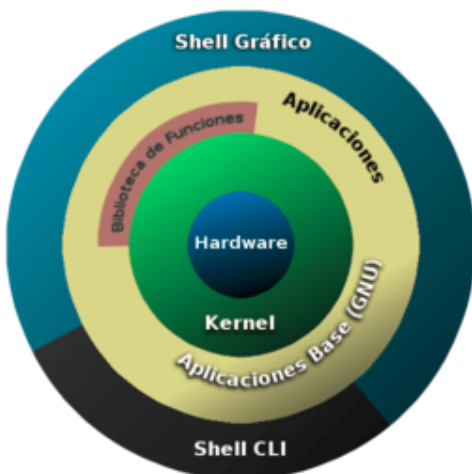


| | | | | |
|-----------|----|--|----|----------|
| 3.3V PWR | 1 | | 2 | 5V PWR |
| I2C1 SDA | 3 | | 4 | 5V PWR |
| I2C1 SCL | 5 | | 6 | GND |
| GPIO 4 | 7 | | 8 | UART0 TX |
| GND | 9 | | 10 | UART0 RX |
| GPIO 17 | 11 | | 12 | GPIO 18 |
| GPIO 27 | 13 | | 14 | GND |
| GPIO 22 | 15 | | 16 | GPIO 23 |
| 3.3V PWR | 17 | | 18 | GPIO 24 |
| SPI0 MOSI | 19 | | 20 | GND |
| SPI0 MISO | 21 | | 22 | GPIO 25 |
| SPI0 SCLK | 23 | | 24 | SPI0 CS0 |
| GND | 25 | | 26 | SPI0 CS1 |
| Reserved | 27 | | 28 | Reserved |
| GPIO 5 | 29 | | 30 | GND |
| GPIO 6 | 31 | | 32 | GPIO 12 |
| GPIO 13 | 33 | | 34 | GND |
| GPIO 19 | 35 | | 36 | GPIO 16 |
| GPIO 26 | 37 | | 38 | GPIO 20 |
| GND | 39 | | 40 | GPIO 21 |

También intel saca su alternativa a raspberry: <http://www.intel.es/content/www/es/es/do-it-yourself/edison.html>

Más información: <https://aprendiendoarduino.wordpress.com/2016/06/25/arduino-vs-raspberry-pi-2/>

Un sketch de Arduino no es un sistema operativo: https://es.wikipedia.org/wiki/Sistema_operativo. Un sistema operativo es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software, ejecutándose en modo privilegiado respecto de los restantes (aunque puede que parte de él se ejecute en espacio de usuario)



Se puede decir que el sketch de Arduino es la aplicación que gestiona directamente los recursos de HW sin necesidad de un SO o un kernel intermedio.



En el caso de raspberry Pi, el programa o sketch se ejecuta como una aplicación sobre un sistema operativo y para interaccionar con el HW necesita de la interacción con el sistema operativo.

Para entender qué es el microcontrolador dentro de Arduino, leer:

<https://aprendiendoarduino.wordpress.com/2015/02/25/como-conseguir-un-arduino-gratis/>

¿Podría convertir un arduino en un ordenador? ¿Cómo? ¿Es práctico?. Arduino como un ordenador:

- Salida TV:
 - <https://code.google.com/p/arduino-tvout/>
 - <http://playground.arduino.cc/Main/TVout>
 - <https://code.google.com/archive/p/arduino-tvout/wikis/FunctionalDescription.wiki>
 - <http://www.instructables.com/id/TV-Out-with-Arduino/>
- Pantalla táctil: <http://tienda.bricogeek.com/shields-arduino/521-arduino-tft-touch-shield-v20.html>
- Teclado: <http://playground.arduino.cc/code/Keypad>, <http://abedulengenharia.blogspot.com.es/2011/07/arduino-y-teclado-3x4.html>
- Separar los procesos en diferentes MCUs, mejor con chips específicos, pero se puede hacer usando varios microcontroladores, uno ejecuta el SO, otros acceso a disco (controladora), otro muestra datos por pantalla (tarjeta gráfica), tarjeta de sonido, etc...
- Disco Duro, leer y escribir ficheros. Acceso a una tarjeta SD. El sketch compilado se podría guardar en una tarjeta SD y un bootloader adecuado podría leer de la tarjeta SD y cargarlo en la memoria de programa para ejecutarlo. Ver ejemplo <http://baldwisdom.com/bootdrive/>
- Sistema Operativo: un sketch diseñado para interaccionar con el HW y poder correr aplicaciones sobre el. Ver <http://antipastohw.blogspot.com.es/2009/11/4-operating-systems-for-arduino.html>

También es posible convertir un ordenador en un microcontrolador <http://www.instructables.com/id/HackTurn-PC-into-a-microcontroller-for-free/?ALLSTEPS>

Microcontroladores 8 bits y 32 bits

A principios de 1970 TI produjo el primero de los microcontroladores el TMS 1000. Aunque Intel hizo anteriormente el microcontrolador de 4 bits Intel 4004, necesitaba de una circuitería externa para funcionar, por lo que el TMS

1000 es considerado el primer microcontrolador completo en un chip.

El tamaño de la palabra es un aspecto importante en la arquitectura de procesadores. La mayoría de los registros de un Microprocesador/Microcontrolador tienen el tamaño de la palabra y las operaciones que hace la ALU es manejando operandos cuyo tamaño es el tamaño de la palabra, así como la cantidad de datos transferidos a memoria y dirección utilizada para designar una localización de memoria a menudo ocupa una palabra.

El tamaño de palabra de un microprocesador/microcontrolador influye principalmente en el tamaño de datos que puede manejar y la cantidad de memoria RAM que puede usar, así como la velocidad de procesamiento.

También los valores que pueden tomar las variables dependen del tamaño de la palabra. http://es.wikipedia.org/wiki/Palabra_%28inform%C3%A1tica%29

Arduinos con procesadores de 8 bits:

- Arduino UNO: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- Arduino MICRO: <https://www.arduino.cc/en/Main/ArduinoBoardMicro>
- Arduino MEGA: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Arduinos o compatibles con procesadores de 32 bits:

- Arduino Due: <http://www.arduino.org/products/boards/4-arduino-boards/arduino-due>
- Arduino 101: <https://www.arduino.cc/en/Main/ArduinoBoard101>
- Arduino ZERO: <https://www.arduino.cc/en/Main/ArduinoBoardZero>
- Arduino MKR1000: <https://www.arduino.cc/en/Main/ArduinoMKR1000>
- ESP8288: <http://espressif.com/en/products/hardware/esp8266ex/overview>
- STM32F103C8: http://wiki.stm32duino.com/index.php?title=Blue_Pill

Anuncios

Esta entrada se publicó en Arduino, Curso Avanzado 2017, Hardware, Microcontrolador, microprocesador y está etiquetada con Arduino, Curso Avanzado 2017, Hardware, Microcontroladores, Microprocesadores en 11 agosto, 2017 [<https://aprendiendoarduino.wordpress.com/2017/08/11/microcontrolador-vs-microprocesador-3/>] .