# MICROCONTROLADORES Y ARDUINO

FUNDACIÓN KINAL

# MICROCONTROLADORES

El microcontrolador nace cuando las técnicas de integración han progresado lo bastante para permitir su fabricación; pero también porque, muy a menudo, tanto en las aplicaciones domésticas como industriales, se tiene la necesidad de sistemas “inteligentes” o, al menos programables.

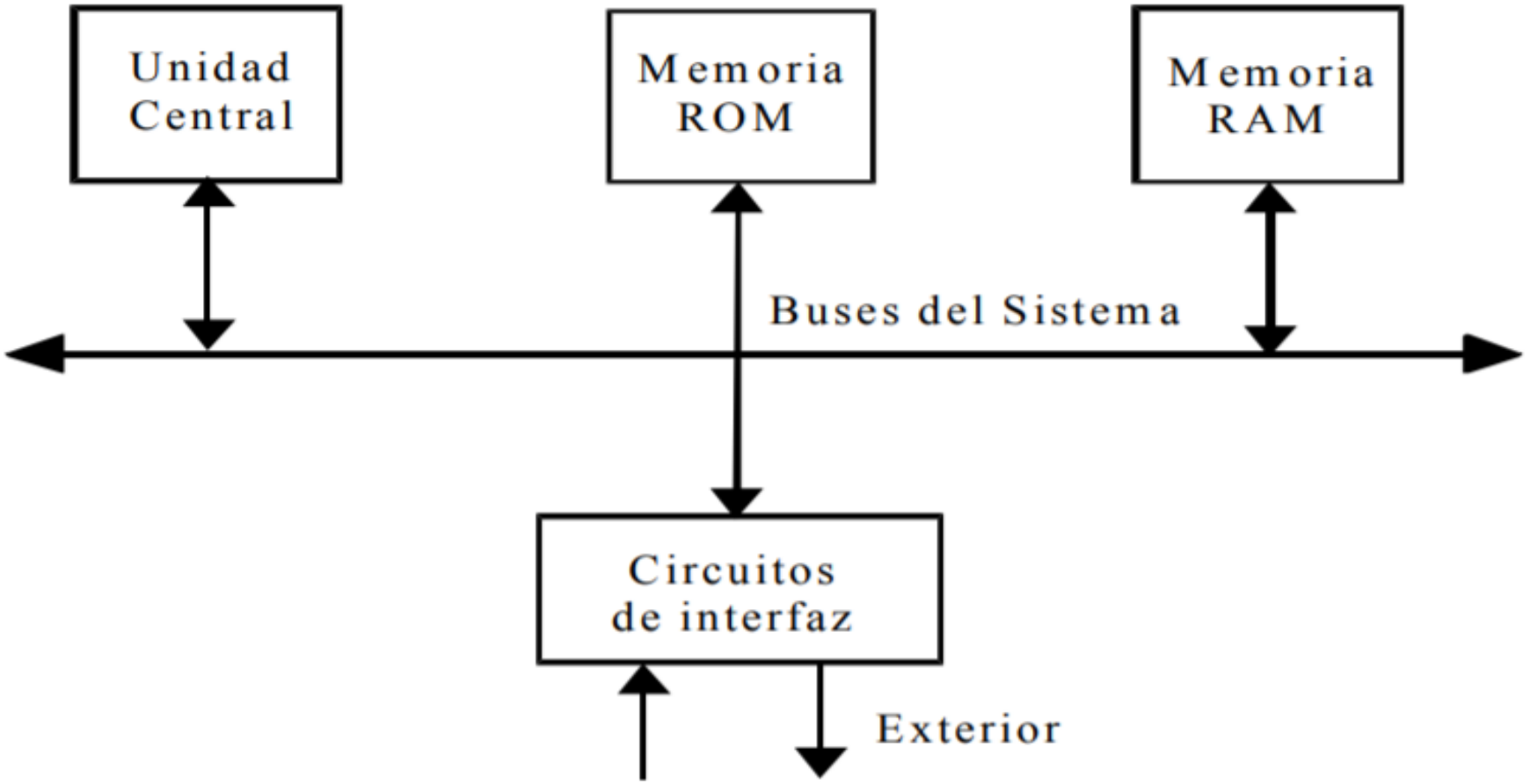
# MICROCONTROLADORES

Un ejemplo muy simple es el programador de una lavadora, el cual debe controlar una cierta cantidad de elementos con ciclos y cadencias perfectamente definidas, pero variables en función del programa seleccionado.

# MICROCONTROLADORES

Otras aplicaciones más técnicas tienen, igualmente, necesidad de sistemas programables. Por ejemplo, una fotocopiadora debe controlar permanentemente un gran número de elementos y de funciones.

# Estructura de un sistema programable



# MICROCONTROLADORES

La unidad central, generalmente constituida por un microprocesador más o menos evolucionado, ejecuta el programa que da vida a la aplicación. Puede tener una diversidad de programas y no necesariamente todos tienen que ser iguales, todo dependerá de la función que tengan que hacer.

# MICROCONTROLADORES

Sin embargo, estos programas tienen en común el hecho de que muy raramente necesitan cálculos complejos y, en cambio, sí suelen incluir numerosas manipulaciones de la información de entrada/salida.

# MICROCONTROLADORES

El programa se almacena en un segundo elemento, que es la memoria ROM. Esta memoria puede constituirse de diferentes formas: UVPROM, EEPROM u OTPROM, cualquiera que sea la que se utilice es una memoria no volátil desde la que se ejecutará el programa una vez alimentado el sistema.

# MICROCONTROLADORES

Para poder trabajar correctamente, nuestro microprocesador necesita, a menudo, almacenar datos temporales en alguna parte, y aquí es donde interviene la memoria RAM, que no necesita ser de grandes dimensiones.

# MICROCONTROLADORES

El último elemento y que, generalmente, es el más importante en una aplicación susceptible de utilizar un microcontrolador es todo lo concerniente a los circuitos de interfaz con el mundo exterior, que relacionará al microprocesador con elementos tan dispares como un motor paso a paso, un display de cristal líquido o una botonera hexadecimal.

# MICROCONTROLADORES

Los primeros microcontroladores venían con memoria ROM, la cual necesitaba de un dispositivo especial para poder guardar programas y no eran reprogramables. En la actualidad aún se pueden encontrar microcontroladores con memoria ROM o sin esta.

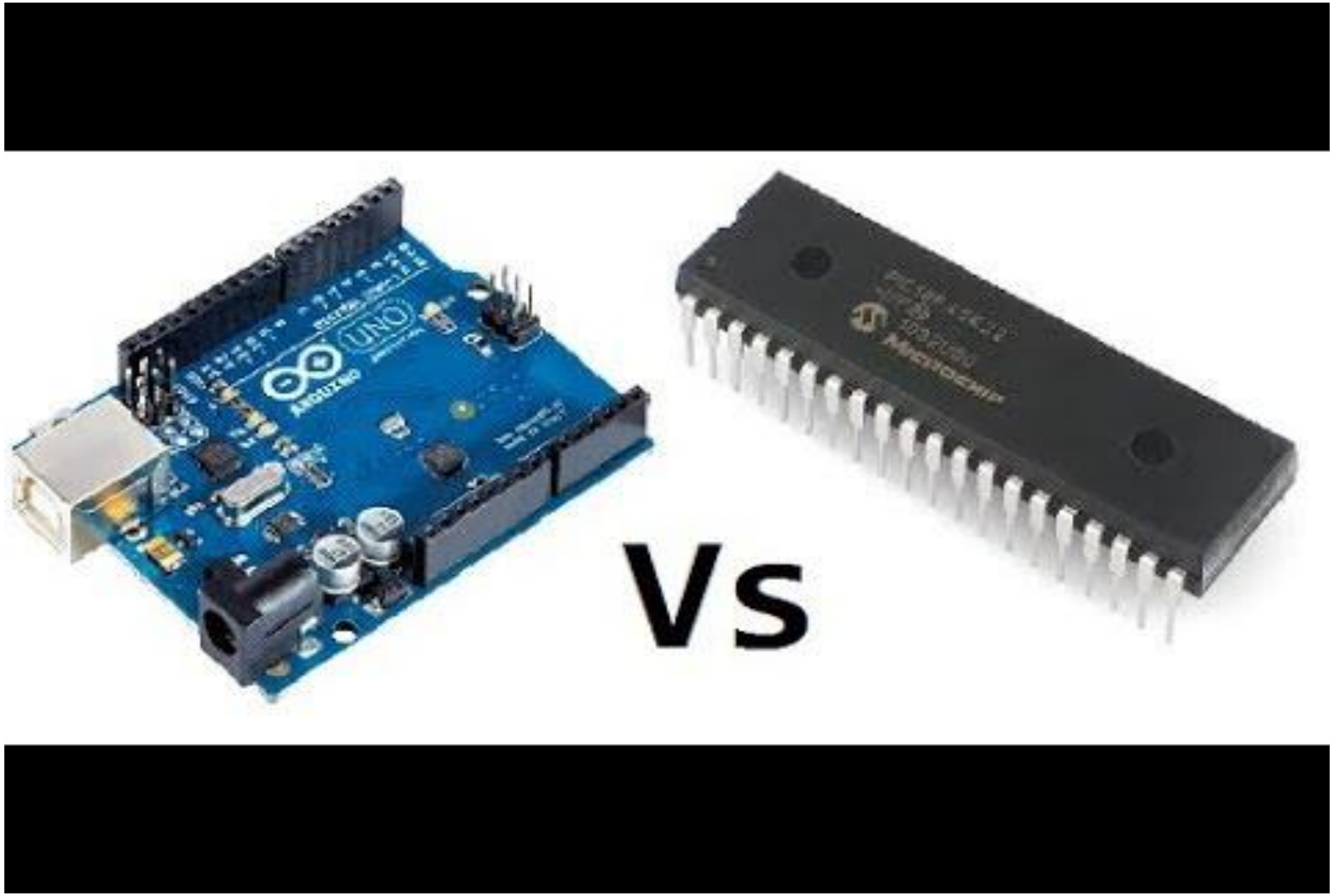
# MICROCONTROLADORES

Los microcontroladores con UVROM tenían la característica de ser reprogramables, para esto necesitaban de luz UV, aunque los costos de fabricación son muy elevados (por la ventana de cuarzo que tienen para la luz UV). Los dispositivos mas recientes vienen con memoria EEPROM las cuales tienen la capacidad de borrar la memoria de manera eléctrica.

# MICROCONTROLADORES

La tecnología más reciente para memorias no volátiles es la FLASH, tecnología que es similar a la EEPROM pero que tiene un muy bajo consumo y tiene mayor velocidad. Entre sus características es que pueden borrar contenido por bloques y no necesariamente tienen que borrar todo lo que tienen almacenado, son muy recomendables en aplicaciones en las que sea necesario modificar el programa a lo largo de la vida del producto, como consecuencia del desgaste o cambios de piezas, como sucede con los vehículos.

# MICROCONTROLADORES



# MICROCONTROLADORES

Raspberry Pi

Arduino Nano

NodeMCU

# ARDUINO

Arduino es una plataforma de desarrollo abierta, concebida para la creación de prototipos y aplicaciones Hardware. Arduino fue creado inicialmente para estudiantes, ya que antes de esto las placas que existían eran caras y tenían una arquitectura cerrada, lo que no las hacía atractivas ni a la comunidad de estudiantes, ni a los aficionados a la electrónica en general.

# ARDUINO

Con la creación de Arduino como un sistema de desarrollo abierto (no necesita licencias) y la constitución de un lenguaje de programación propio, el cual esta basado en lenguaje C y tiene una gran comunidad de programadores, existiendo mucha fuentes y repositorios de programas y librerías en internet.

# ARDUINO

El Hardware de Arduino, consiste en una placa con un microcontrolador Atmel AVR, en función del modelo de placa llevará un microcontrolador u otro, los más usados son el Atmega168, Atmega328, Atmega1280 y el ATmega8.

# MODELOS DE ARDUINO

En la actualidad Arduino ha lanzado al mercado, placas, shields y sensores casi para cualquier cosa, las diferencias fundamentales entre una placa u otra suelen ser, el número de pines que sacan al exterior y la tipología de estos (si son pines digitales, analógicos o de PWM...), la memoria de programa de la que disponemos y la frecuencia de reloj del microcontrolador, o lo que es lo mismo, la velocidad a la que podemos trabajar.

# MODELOS DE ARDUINO

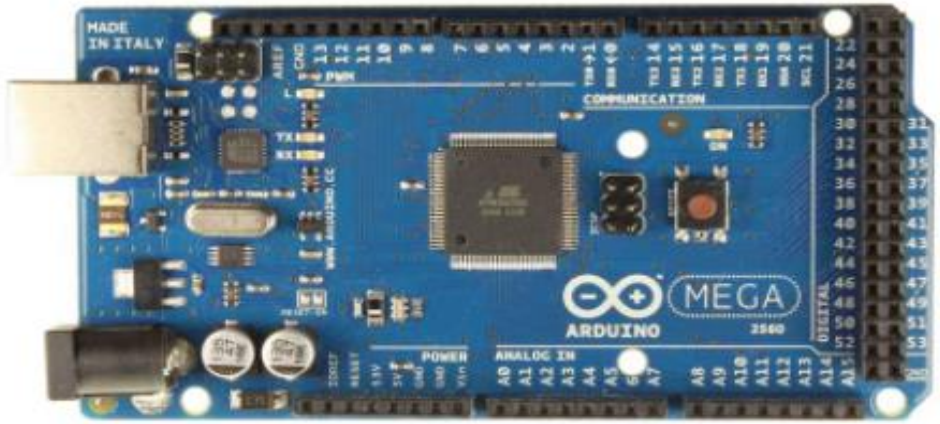
En función de estos tres parámetros, deberemos escoger una placa u otra para nuestros proyectos (dependiendo de nuestras necesidades), ya que puede que tan solo necesitemos unas pocas entradas/salidas y que la velocidad para la aplicación no sea importante o quizás tengamos que gestionar 20 o 30 procesos diferentes y hacerlo de manera casi instantánea, según lo que queramos hacer, podemos ajustarnos a una placa u otra, ya que como es lógico una placa con mayor número de entradas/salidas, mayor memoria de programa y mayor velocidad será más cara y puede que algunos proyectos no necesiten estos recursos.

# MODELOS DE ARDUINO


LEONARDO

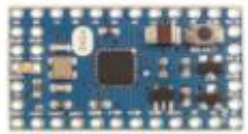

YUN


UNO




MEGA 2560

NANO



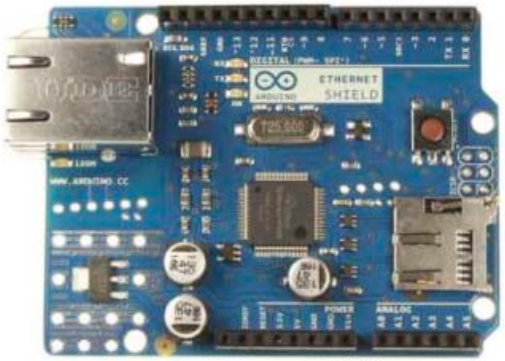
MINI

# MODELOS DE ARDUINO

| Modelo | Microcontrolador | Voltaje de entrada | Voltaje del sistema | Frecuencia de Reloj | Digital I/O | Entradas Analógicas | PWM | UART | Memoria Flash | Interfaz de Programación |
|---|---|---|---|---|---|---|---|---|---|---|
| **Arduino Due** | AT91SAM3X8E | 5-12V | 3.3V | 84MHz | 54 | 12 | 12 | 4 | 512Kb | Nativa USB |
| **Arduino Leonardo** | ATmega32U4 | 7-12V | 5V | 16MHz | 20 | 12 | 7 | 1 | 32Kb | Nativa USB |
| **Arduino Uno - R3** | ATmega328 | 7-12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | USB via ATMega16U2 |
| **Arduino Pro 3.3V/8MHz** | ATmega328 | 3.35 -12V | 3.3V | 8MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| **Arduino Pro 5V/16MHz** | ATmega328 | 5 - 12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| **Ethernet** | ATmega328 | 7-12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| **Arduino Mega 2560 R3** | ATmega2560 | 7-12V | 5V | 16MHz | 54 | 16 | 14 | 4 | 256Kb | USB via ATMega16U2 |
| **Arduino Mini 05** | ATmega328 | 7-9V | 5V | 16MHz | 14 | 6 | 8 | 1 | 32Kb | Cabecera Serial |
| **Arduino Pro Mini 3.3V/8MHz** | ATmega328 | 3.35 -12V | 3.3V | 8MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| **Arduino Pro Mini 5V/16MHz** | ATmega328 | 5 - 12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| **Arduino Fio** | ATmega328P | 3.35 -12V | 3.3V | 8MHz | 14 | 8 | 6 | 1 | 32Kb | Cabecera compatible con FTDI o Inalámbrica via XBee[1] |

# SENSORES PARA ARDUINO

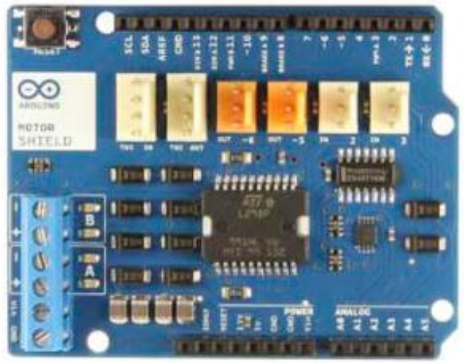
Shields




Ethernet

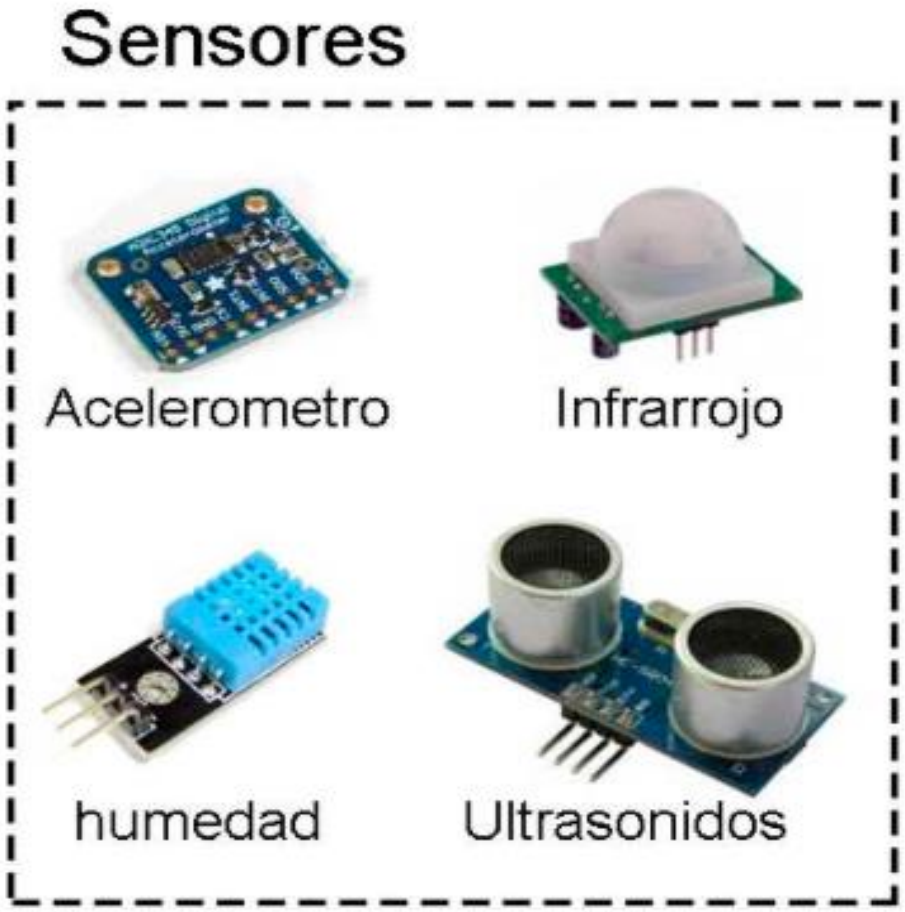
LCD

Wifi




Control de motores

# SENSORES PARA ARDUINO



# LENGUAJE DE ARDUINO

Arduino se programará mediante un editor de texto o "sketch" el cual usa un lenguaje propio creado expresamente para la plataforma Arduino. El Sketch es un entorno de programación sencillo el cual no tiene más que las opciones básicas de programación, verificación del código y carga en la placa Arduino.

# LENGUAJE DE ARDUINO