

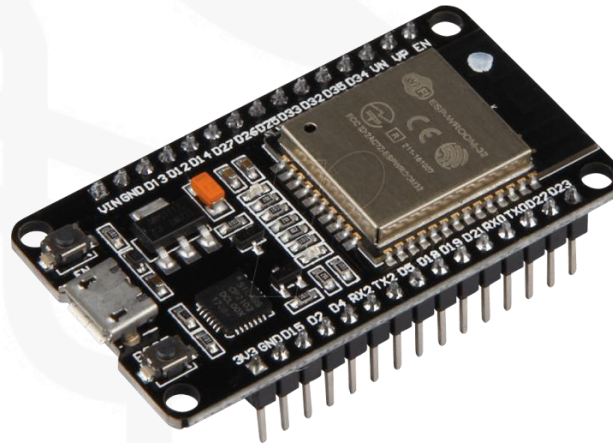
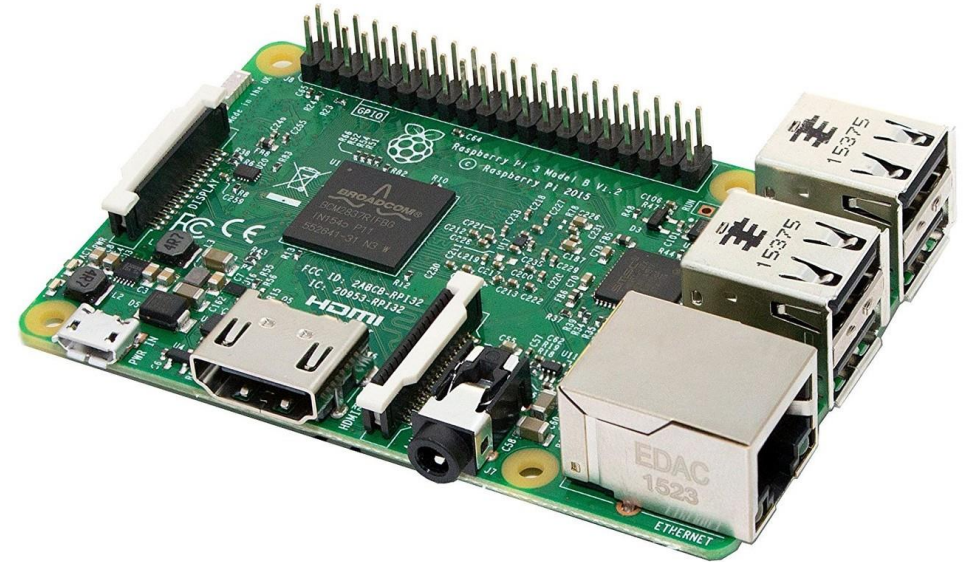
Computación Ubicua

Sesión 2 – Sistemas Inteligentes

Ana Castillo Martínez

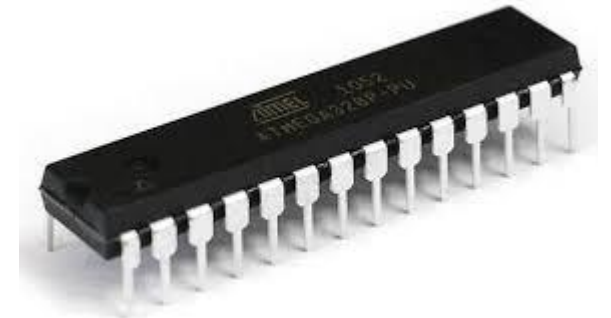
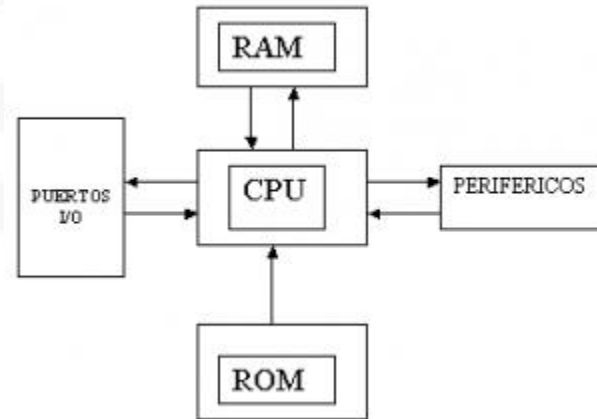
Javier Albert Seguí

¿En qué se diferencian?

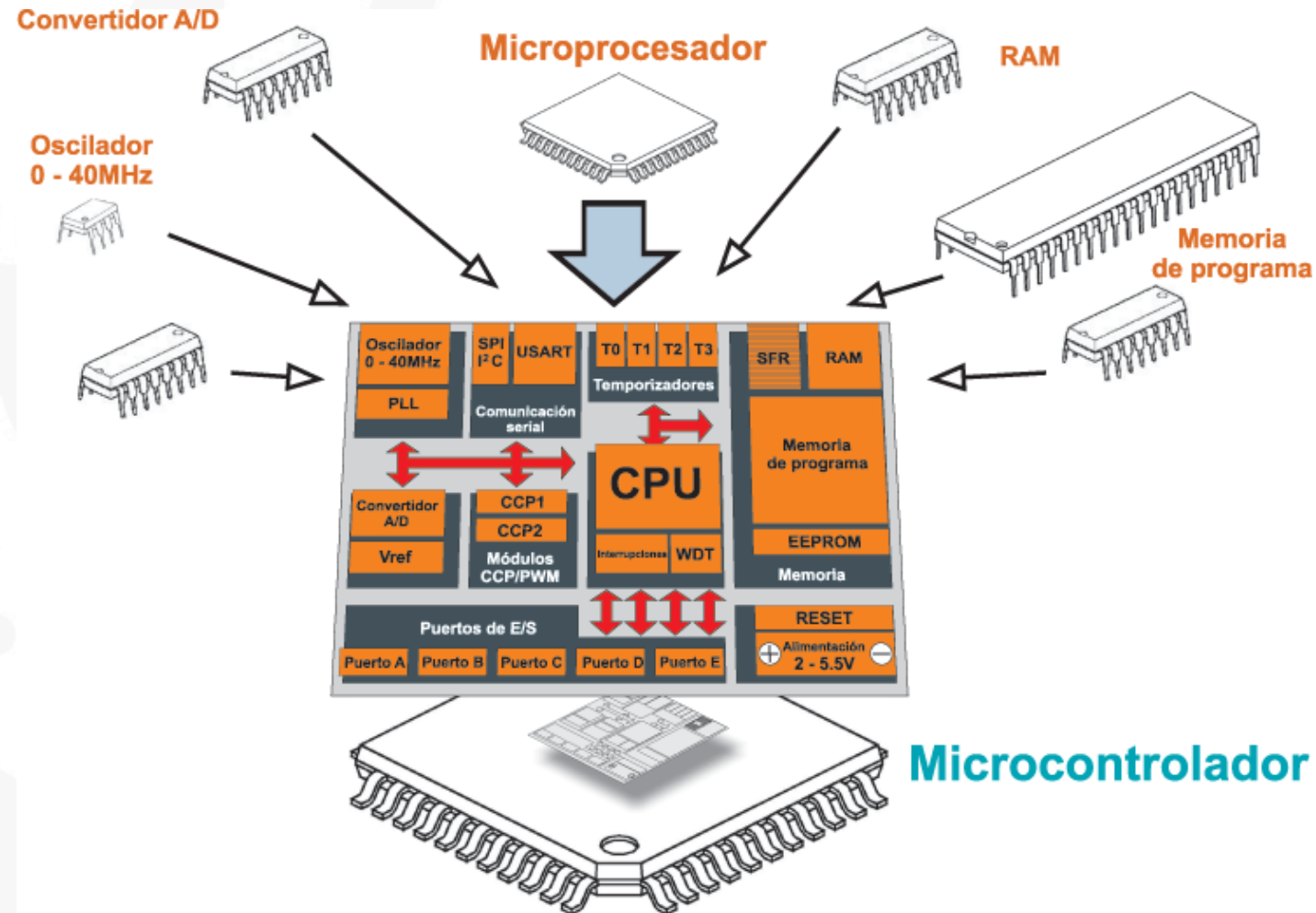


Microcontroladores

- Circuito integrado que en su interior contiene una CPU, unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos
- Un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado



Microcontroladores



Microprocesadores

- Chip integrado basado en silicio con solo una unidad de procesamiento central
- Es el cerebro de un ordenador. Permite realizar operaciones a gran velocidad
- No tiene RAM, ROM, periféricos...



Microprocesador vs Microcontrolador

MICROPROCESADOR	MICROCONTROLADOR
Es una unidad de procesamiento en un chip integrado basado en silicio	Es un subproducto de desarrollo de un microprocesador con una CPU junto a periféricos
No tiene RAM, ROM, unidades IO, temporizadores ni otros periféricos en el chip	Tiene una CPU con RAM, ROM y otros periféricos integrados en un solo chip
Utiliza un bus externo para interactuar con RAM, ROM, pines IO y otros periféricos	Se utiliza un bus de datos interno que no está disponible para el diseño de la placa
Un sistema basado en un microprocesador puede funcionar a alta velocidad debido a su tecnología	Los sistemas basados en microcontroladores funcionan hasta 200MHz dependiendo de su arquitectura
Se utiliza para realizar aplicaciones de propósito general que pueden manejar la carga de datos	Se utiliza en sistemas específicos
Es complejo y costoso, capaz de procesar una gran cantidad de instrucciones	Es simple y barato, y capaz de procesar una menor cantidad de instrucciones

Arduino

- Plataforma de creación de electrónica de Código abierto
- Está basada en Hardware y Software libre, flexible y fácil de utilizar
- Ofrece un entorno de programación (IDE) con el que crear aplicaciones
- Cuenta con una gran comunidad que apoya el Desarrollo, comparte conocimiento, elabora librerías para facilitar su uso, publica proyectos que sirven de apoyo

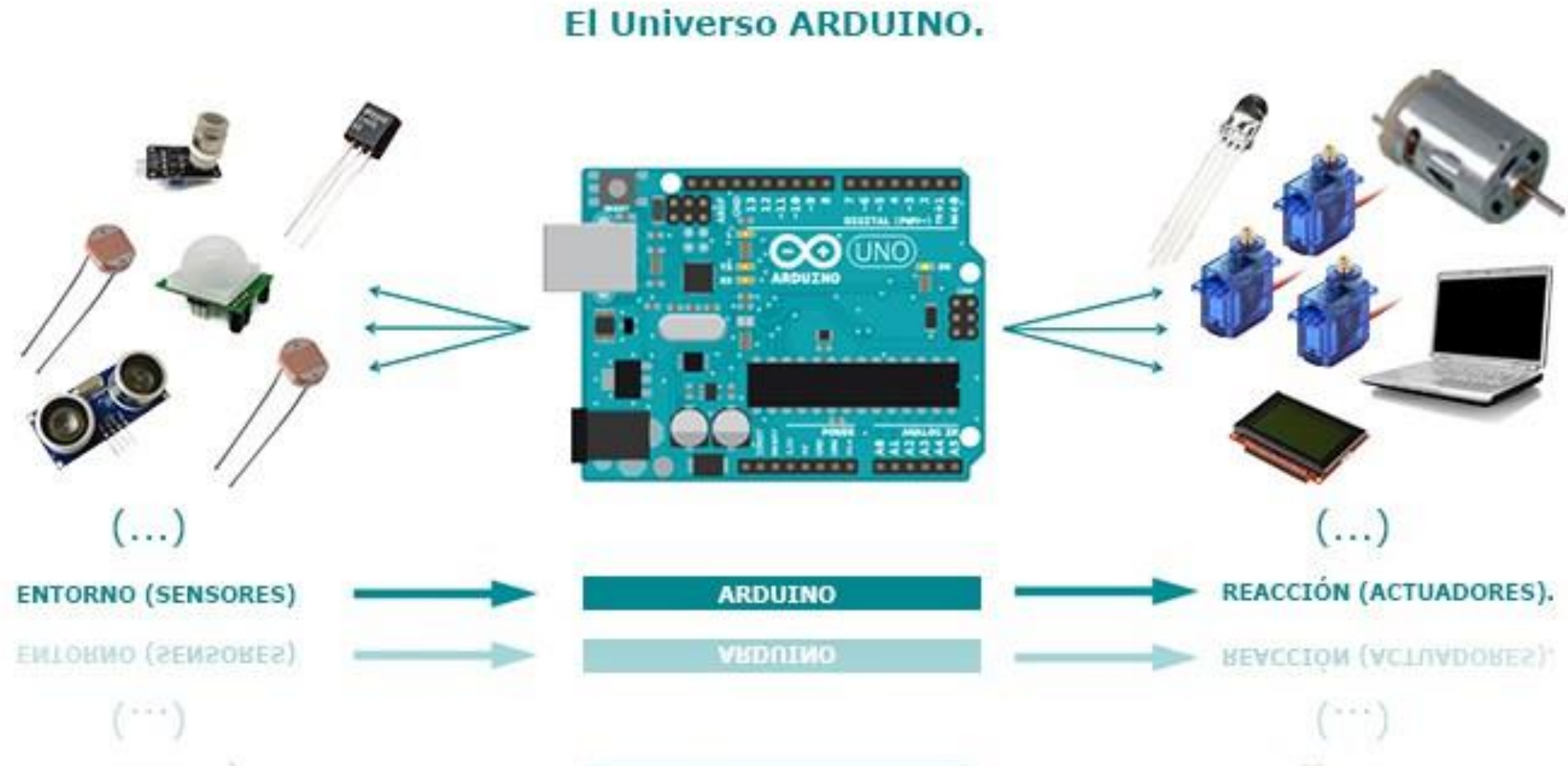
Arduino = HW + SW + Comunidad



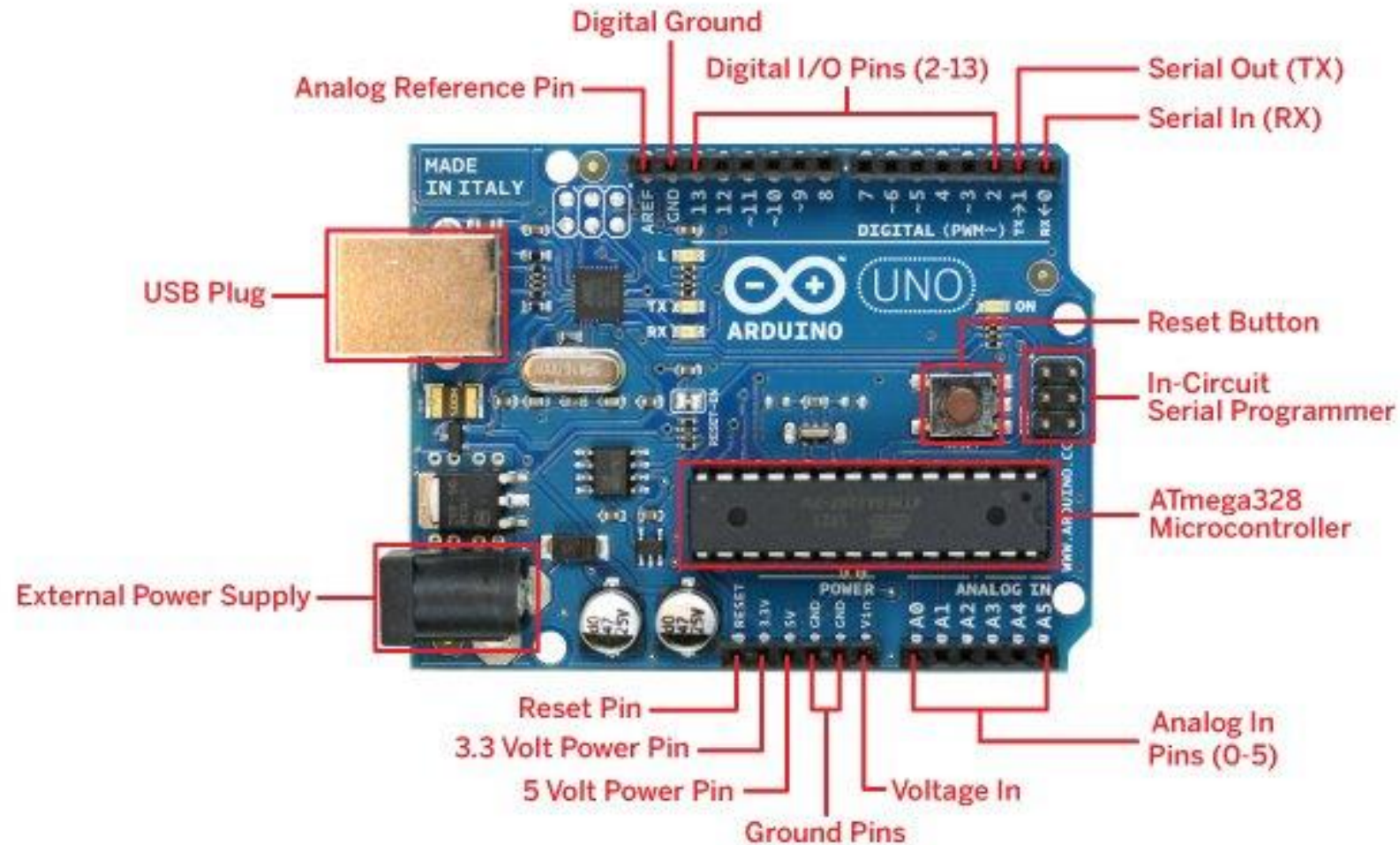
Placa Arduino

- La placa Arduino está basada en un microcontrolador ATMEL, en el cual se pueden grabar instrucciones
- Posee:
 - **Interfaz de entrada** con la que conectar diferentes periféricos
 - **Interfaz de salida** que envía la información procesada a otros periféricos

Placa Arduino



Placa Arduino

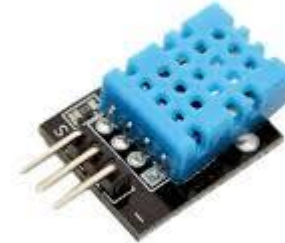
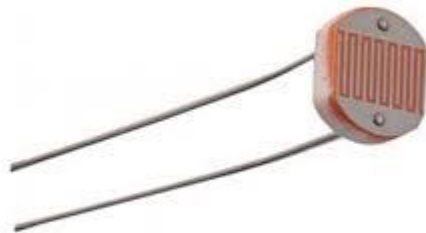


Modelos de Arduino



Sensores

- Dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas
- Se pueden clasificar según los datos de salida en:
 - Analógicos
 - Digitales

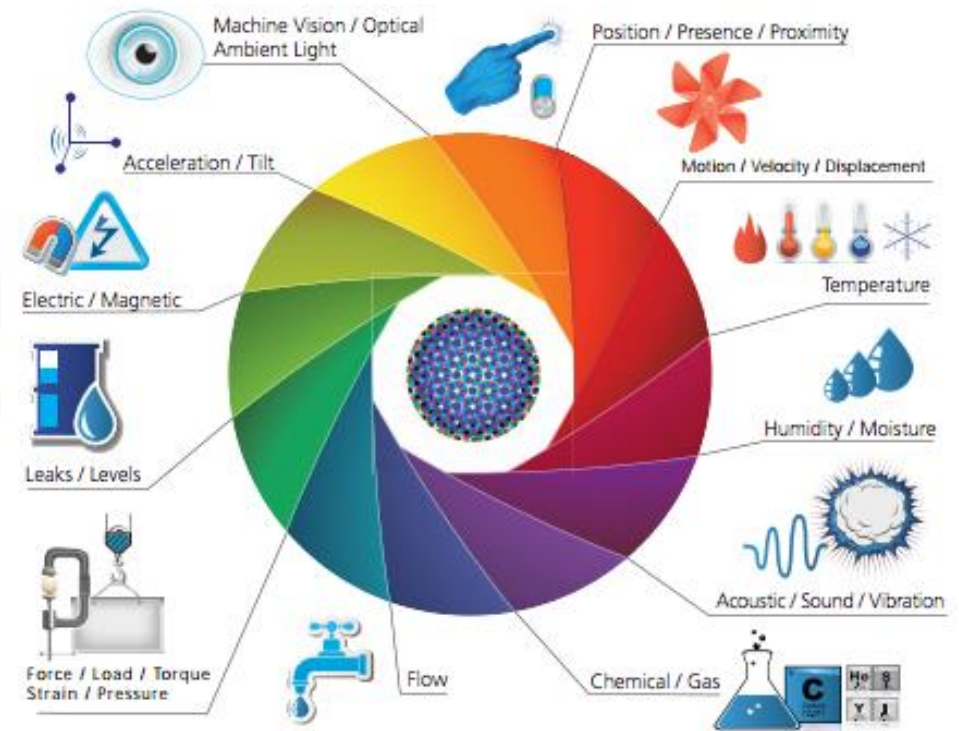


Sensores - Características

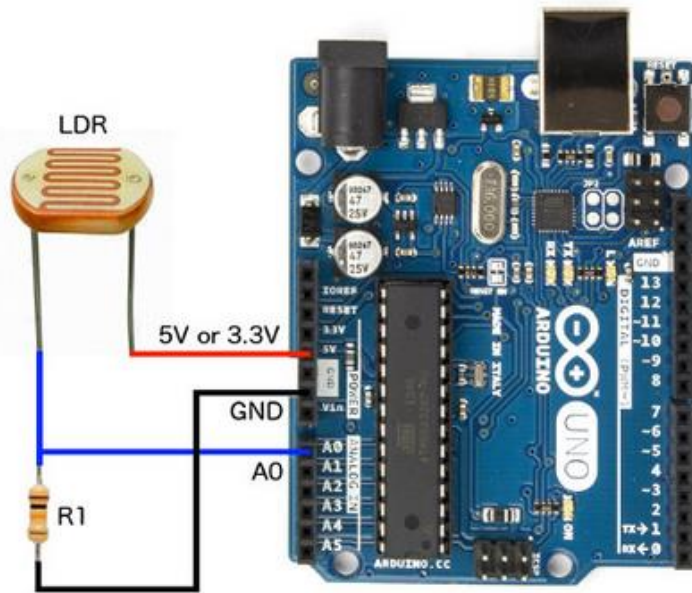
- Rango de medida
- Precisión
- Offset o desviación de cero
- Linealidad o correlación lineal
- Sensibilidad
- Resolución
- Rapidez en respuesta
- Derivas
- Repetitividad

7 SENSORS & ACTUATORS

We are giving our world a **digital nervous system**. Location data using GPS sensors. Eyes and ears using cameras and microphones, along with sensory organs that can measure everything from temperature to pressure changes.

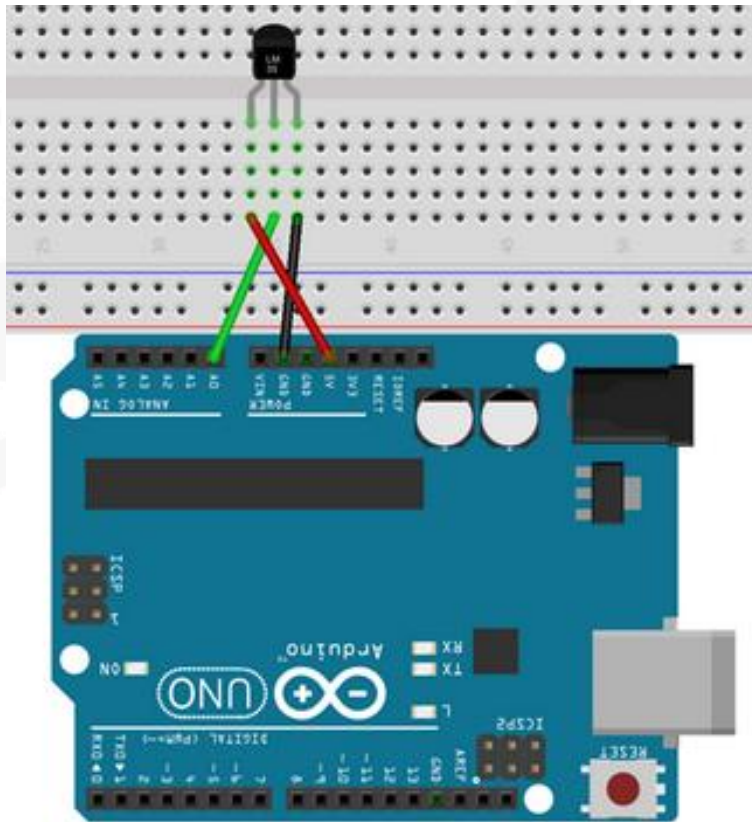


Ejemplo Sensor LDR (Luz)



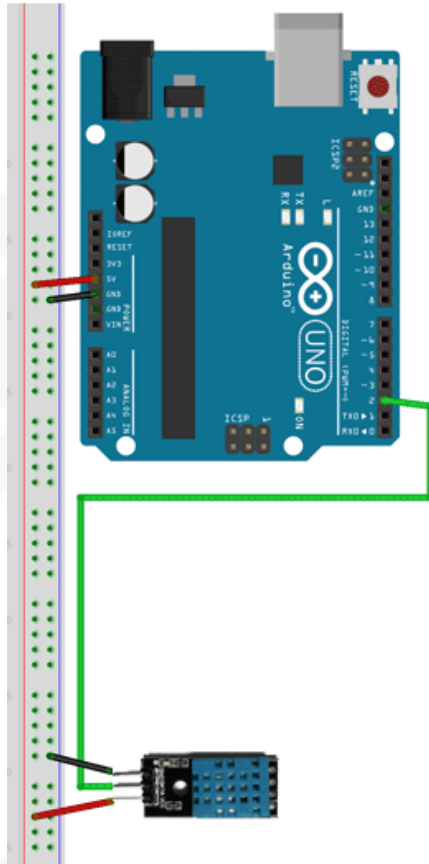
```
int analogPin = 0; //Arduino Analog pin 0
void setup(){
    Serial.begin(9600);
}
Void loop(){
    int lightvalue =analogRead(analogPin);
    Serial.print(lightvalue);
    delay(5000);
}
```


Ejemplo Sensor Temperatura (LM35)



```
int analogPin = 0; // Arduino Analog pin 0
void setup(){
    Serial.begin(9600);
}
Void loop(){
    int tempvalue =analogRead(analogPin);
    tempvalue =(tempvalue *500)/1024;
    Serial.print(temp);
    delay(5000);
}
```

Ejemplo Sensor Temperatura y Humedad (DHT11)



```
#include <DHT.h>

#define DHTPIN 2 //Define the digital pin where the sensor is connected

#define DHTTYPE DHT11

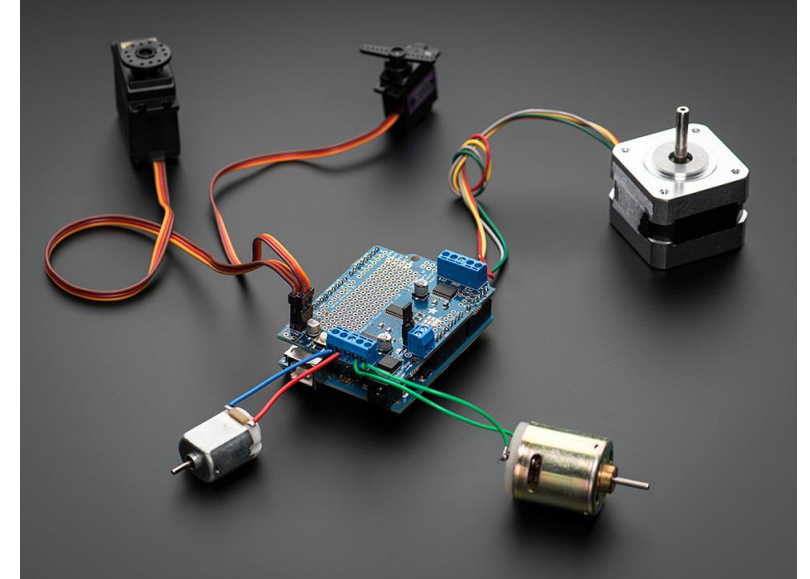
DHT dht(DHTPIN, DHTTYPE); //Initialize the DHT11 sensor

void setup() {
    Serial.begin(9600);
    dht.begin(); //Start the DHT sensor
}

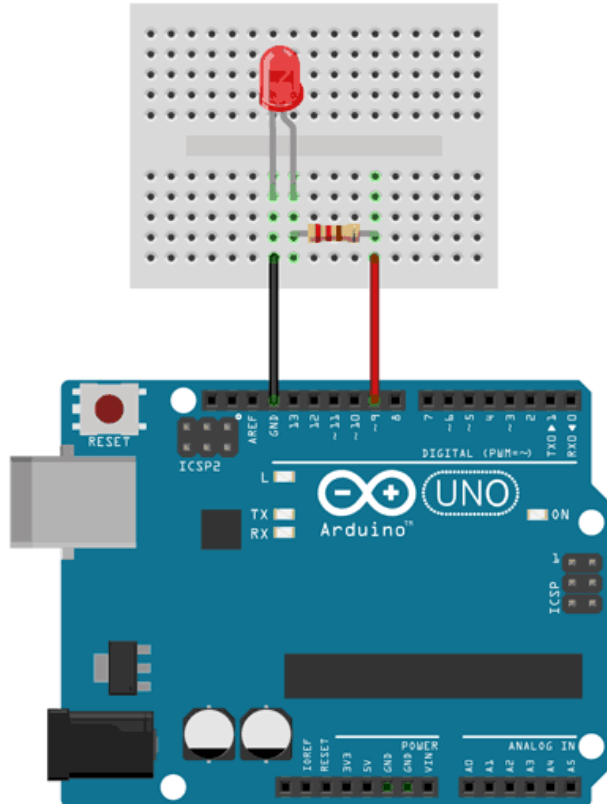
void loop() {
    float h = dht.readHumidity(); //Read relative humidity
    float t = dht.readTemperature(); //Read temperatura in Celsius degrees
    float f = dht.readTemperature(true); // Read temperatura in Fahrenheit degrees
    delay (5000);
}
```

Actuadores

- Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre elemento externo.
- Existen varios tipos de actuadores:
 - Electrónicos
 - Hidráulicos
 - Neumáticos
 - Eléctricos
 - Motores
 - Bombas
 -



Ejemplo Actuador. Led

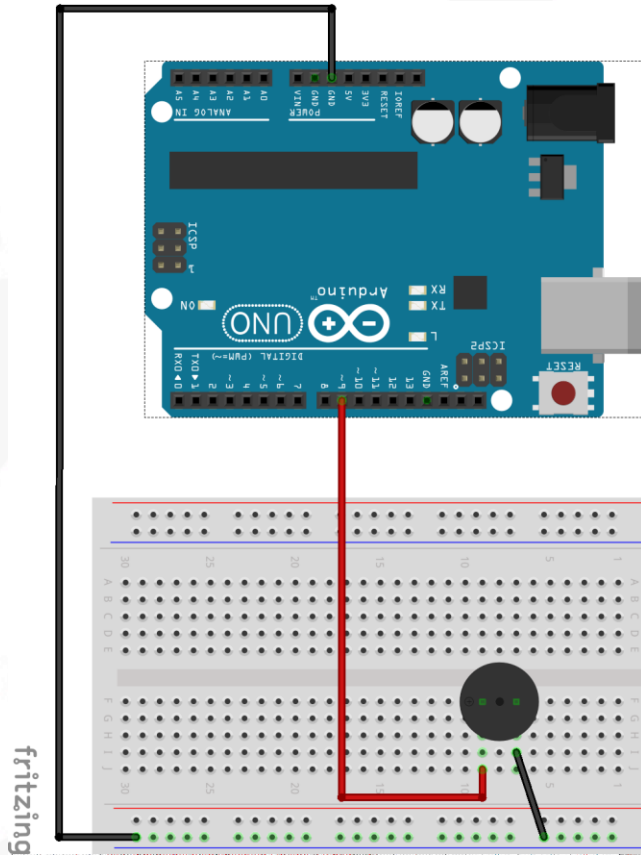


```
int pin = 9; //Arduino Digital pin 9

void setup(){
    pinMode(pin, OUTPUT); //Define pin as output
}

void loop(){
    digitalWrite (pin, HIGH); //Turn the led on
    delay(1000);
    digitalWrite (pin, LOW); //Turn the led off
    delay(1000);
}
```

Ejemplo Actuador. Buzzer

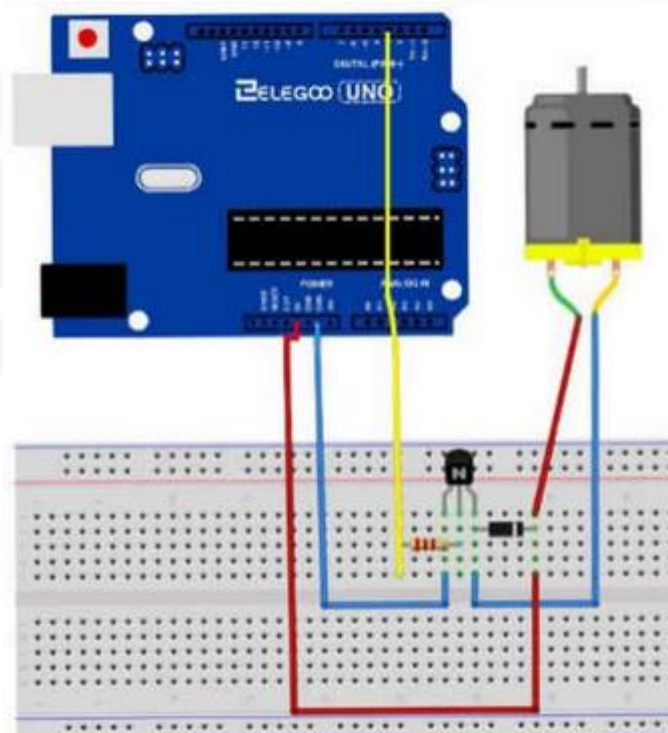


```
int pin = 9; //Arduino Digital pin 9

void setup(){
    pinMode(pin, OUTPUT); // Define pin as output
}

void loop(){
    tone (pin, 1000); //Send 1kHz signal
    delay(1000);
    noTone(pin); //stop the sound
    delay(1000);
}
```

Ejemplo Actuador. Bomba de Agua



```
int pin = 3; //Arduino Digital pin 3

void setup(){
    pinMode(pin, OUTPUT); //Define pin as output
}

void loop(){
    digitalWrite (pin, HIGH); // Turn the pump on
    delay(5000);
    digitalWrite (pin, LOW); //Turn the pump off
    delay(5000);
}
```

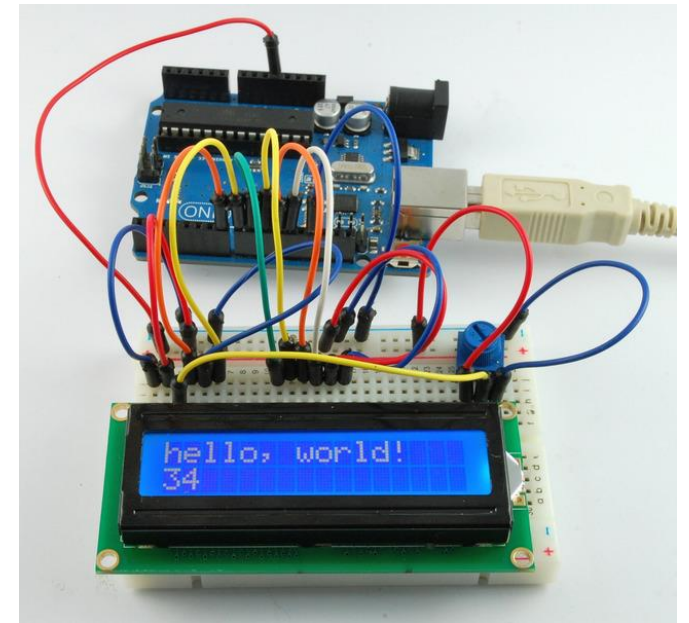

Periféricos

- Dispositivo auxiliar e independiente conectado a la unidad central de procesamiento
- Se consideran periféricos a:
 - las unidades o dispositivos de hardware a través de los cuales Arduino se comunica con el exterior
 - a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.
- En ocasiones para usar un periférico con Arduino, necesitamos un driver para poder mandar órdenes desde Arduino

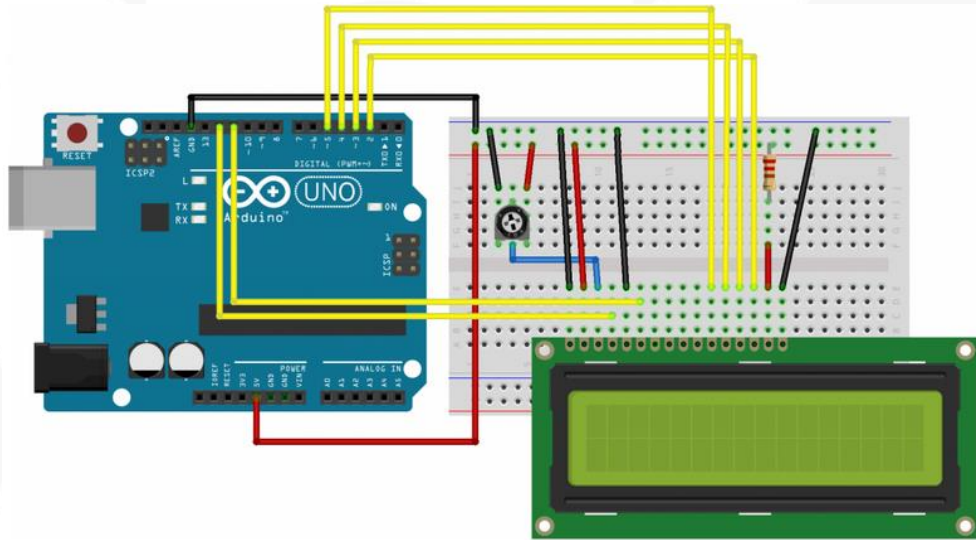
Periféricos

○ Ejemplos de periféricos:

- Pantallas LCD
- Teclados
- Memorias externas
- Cámaras
- Micrófonos
- Impresoras
- Pantalla táctil
- Displays numéricos
- Zumbadores
- Indicadores luminosos,
- ...



Ejemplo Periférico. Pantalla LCD



```
#include <liquidCrystal.h>
```

```
LiquidCrystal lcd(2,3,4,5,11,12) //Declare LCD pins
```

```
void setup(){
```

```
  lcd.begin(16,2); //LCD Dimensions
```

```
  lcd.setCursor(0,0); //Set Cursor position
```

```
  lcd.print("Hello World"); //Enter the desired code
```

```
  lcd.setCursor(0,1);
```

```
  lcd.print("Ubicomp");
```

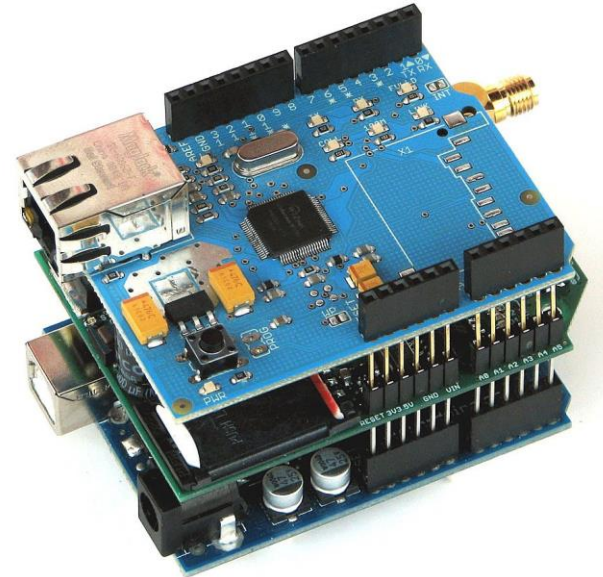
```
}
```

```
void loop(){
```

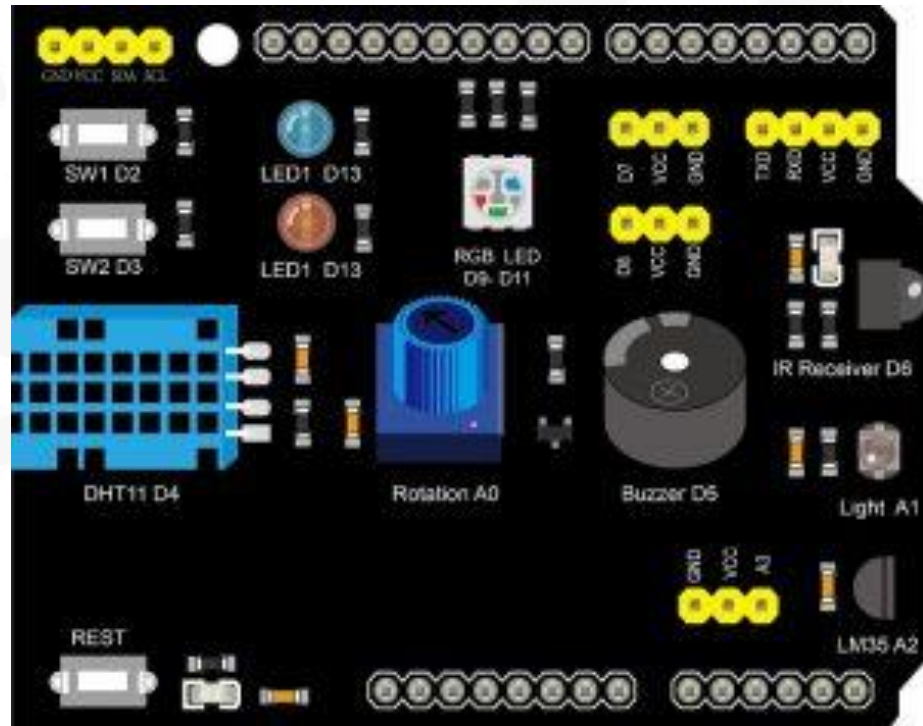
```
}
```

Shields

- Las shields son placas de circuitos modulares que se montan unas encima de otras para dar funcionalidad extra a un Arduino
- Son apilables, pudiendo añadir más de una shield
- Se alimentan a través de los pines GND y 5V



Shields



NodeMCU

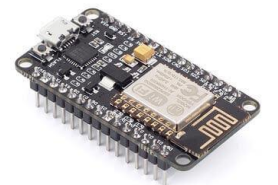
- NodeMCU es un nombre que recoge tanto un firmware Open Source y como a una placa de desarrollo
- Son placas o kits de desarrollo que llevan incorporados un chip que se suele llamar SoC (System on a Chip) que dentro tiene un microcontrolador o MCU
- Inicialmente podía programarse en lenguaje Lua, aunque actualmente también permite usar entornos como Arduino o microPython



NodeMCU

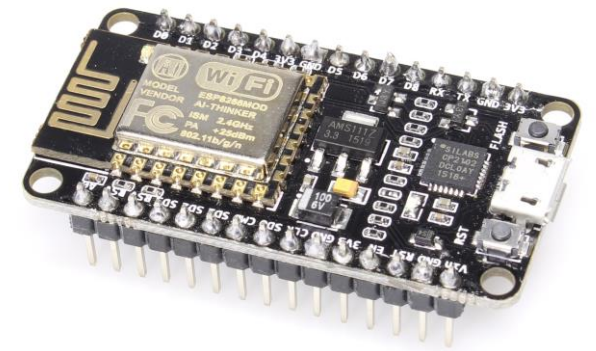
- Existen 3 versiones:

Generación	Versión	Características
1ª	V0.9	El módulo que utiliza es el ESP-12 con una memoria Flash de 4 MB
2ª	V1.0	El chip fue actualizado al ESP-12E, aportando pines extra
3ª	V1.0	Dos de los pines que no se utilizan en la versión V2 se han utilizado como salida de 5V directa del USB y un GND adicional



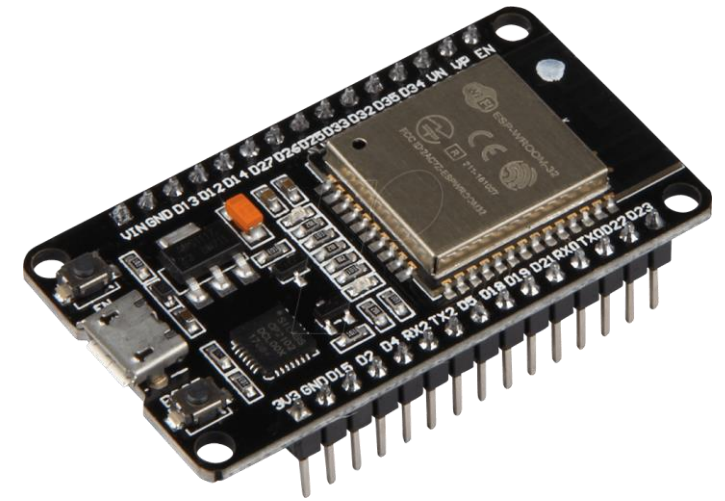
ESP8266

- Chip WIFI de bajo coste que funciona mediante el protocolo TCP/IP
- Incluye un microcontrolador que se puede programar con el entorno de Arduino
- Dispone de entradas/salidas digitales de propósito general (GPIO), así como una entrada analógica



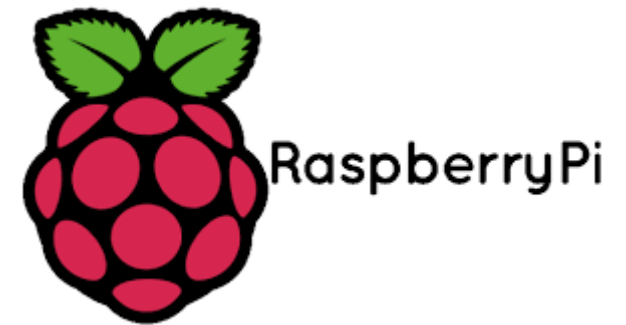
ESP32

- Versión más avanzada que el módulo ESP8266
- Tiene una antena WIFI y un módulo Bluetooth integrados
- Compatibilidad a la hora de programarlo:
 - IDE de Arduino
 - MicroPython
 - RTOS
 - Mongoose OS
 - Espruino

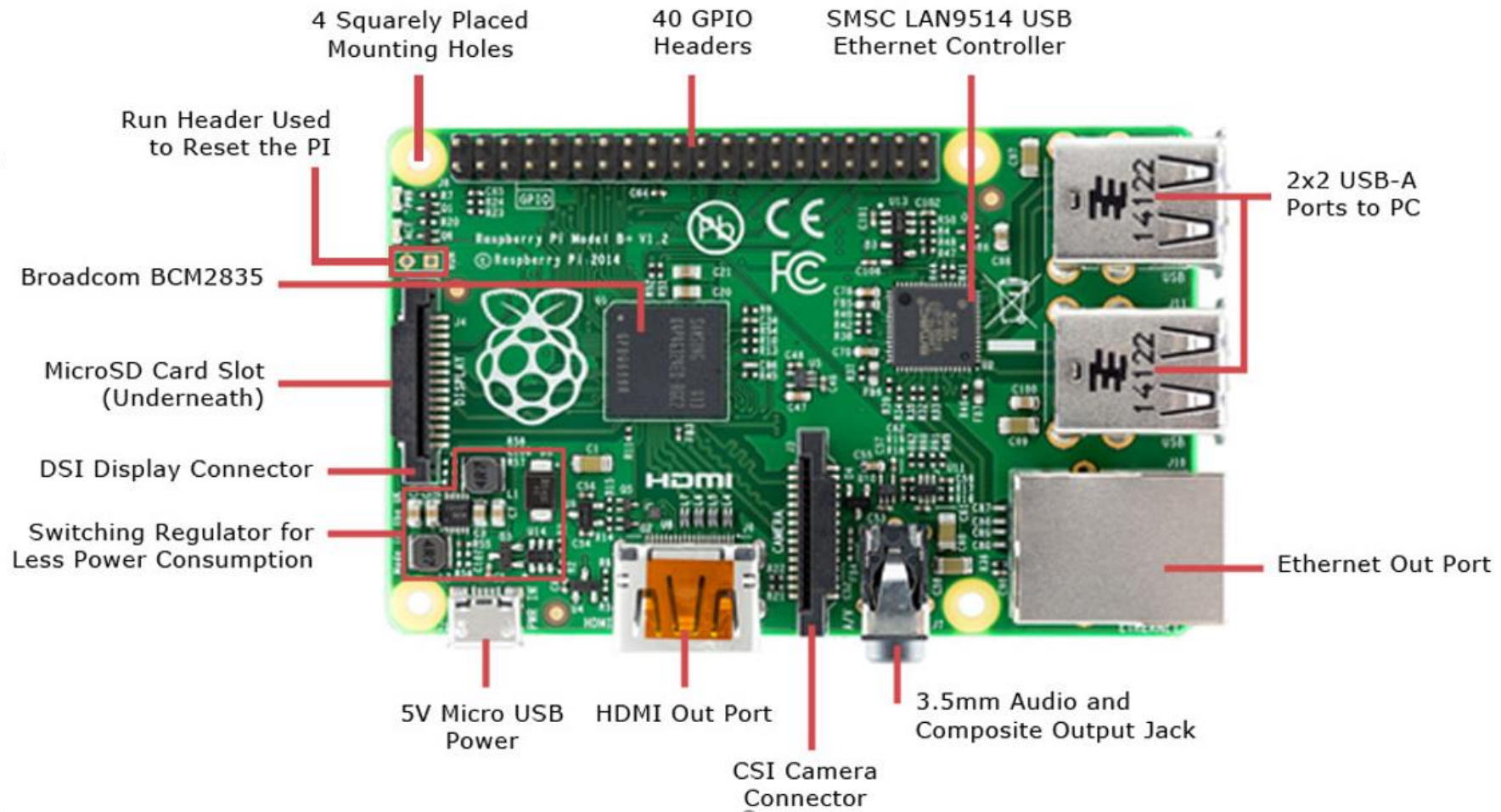


Raspberry Pi

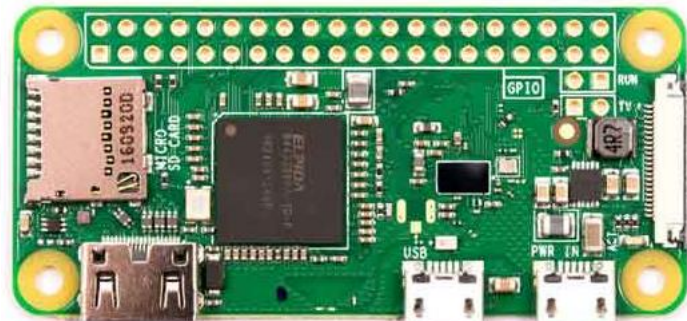
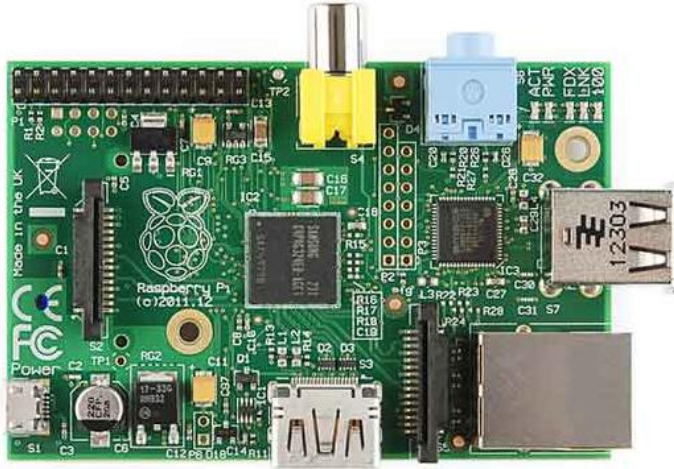
- Raspberry Pi es un ordenador de placa simple (SBC) de bajo coste
- Surge con el objetivo de estimular la enseñanza de la informática en las escuelas
- Software de código abierto, con un sistema operativo basado en Debian llamado Raspbian



Raspberry Pi



Modelos Raspberry Pi



Arduino Vs. Raspberry Pi

	ARDUINO	RASPBERRY PI
Pros	<ul style="list-style-type: none">• Es sencillo de aprender• Buena opción para aplicaciones en tiempo real• No es necesario grandes conocimientos de programación para realizar aplicaciones básicas• Muy sencillo de ampliar con shields y librerías	<ul style="list-style-type: none">• Se conecta fácilmente a Internet• Tiene el software Linux disponible• Se puede programar usando diferentes lenguajes
Contras	<ul style="list-style-type: none">• Es menos potente que la Raspberry Pi• Solo se puede programar con Arduino o C/C++• La conexión a internet es más compleja	<ul style="list-style-type: none">• El acceso al hardware no es en tiempo real. Si la CPU está atascada, la interfaz hardware puede retrasarse• Carece de carga suficiente para cargas inductivas• Carece de un convertidor analógico-digital• El hardware no es de Código abierto

Sistemas Inteligentes

- Un sistema inteligente es aquel sistema capaz de resolver problemas complejos y multidisciplinares de una forma automática dando soporte a las decisiones de un experto
- Son sistemas que reúnen características de comportamiento asimilables al de la inteligencia humana o animal



Historia de los Sistemas Inteligentes

- 1950 – Alan Turing publica “Computing Machinery and Intelligence”
- 1956 – Conferencia en la Universidad de Dartmouth
- 1961 – Marvin Minsky publica “Pasos hacia la inteligencia artificial”
- 1966 – El programa interactivo conversacional ELIZA
- 1972 – Hubert Dreyfus publica “Lo que no pueden hacer los ordenadores”
- 1979 – Un ordenador vence al backgammon
- 1981 – Japón comienza el proyecto “Quinta Generación”

Historia de los Sistemas Inteligentes

- 1987 - Martin Fischles y Oscar Firschein describen los atributos de un agente inteligente
- 1997 – “Deep Blue” vence a Kasparov, campeón mundial de ajedrez
- 2005 – Un ordenador al volante
- 2011 – Watson gana Jeopardy!
- 2014 - Un ordenador supera con éxito el Test de Turing
- 2016 – Microsoft lanza Tay
- 2016 – Alphago vence al Go
- 2017 – Libratus vence al póker

Características de los Sistemas Inteligentes

Objetivo

- Es una cierta situación que el sistema quiere lograr

Inteligencia

- Capacidad del sistema en lograr sus objetivos

Sistematización

- Un sistema es parte de un ambiente o entorno, con una extensión limitada en espacio y tiempo

Conceptualización

- Es el almacenamiento de la información o del conocimiento mediante su representación utilizando términos, estos están relacionados entre sí

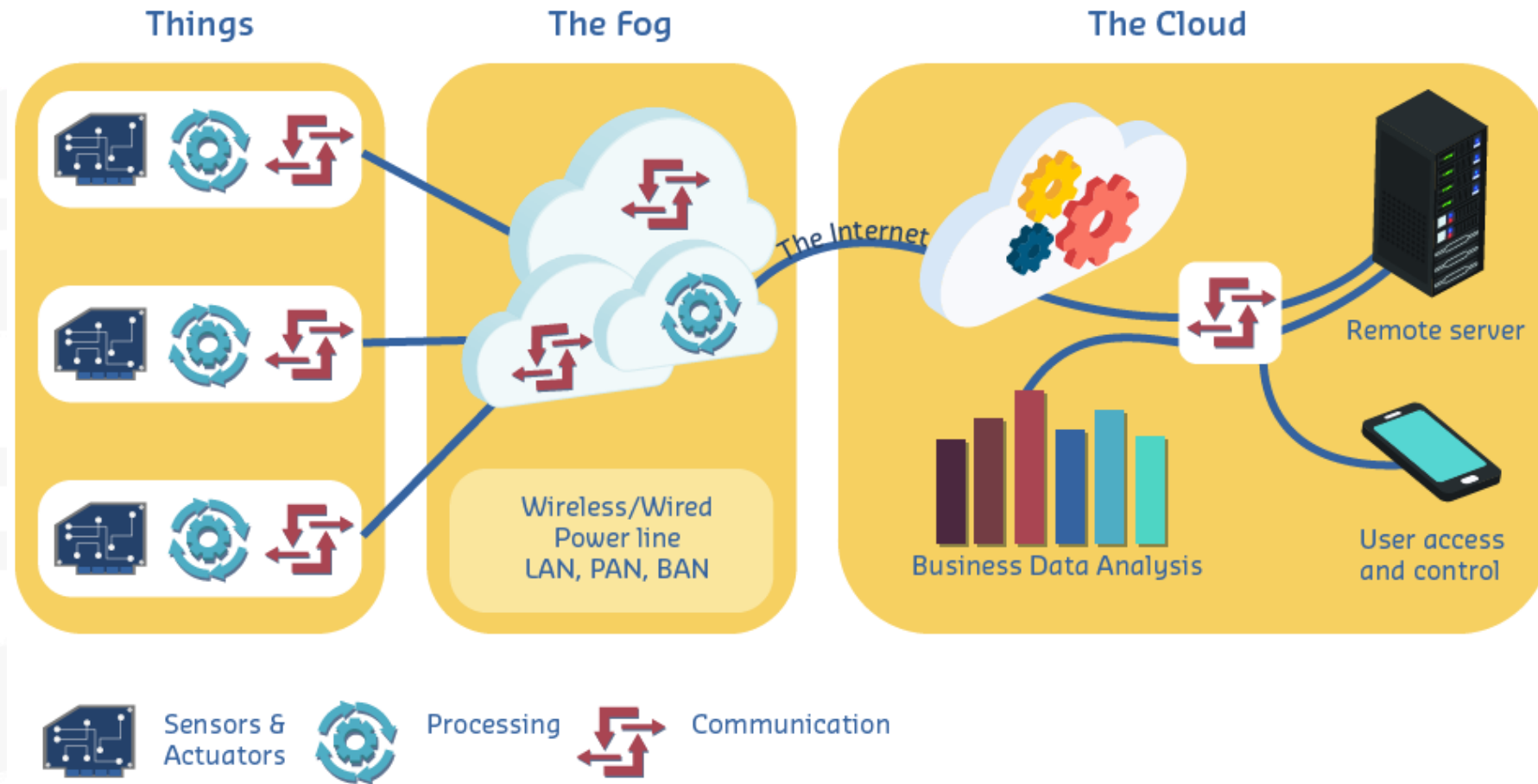
Reglas de actuación

- Relaciona una causa con un efecto de una determinada acción

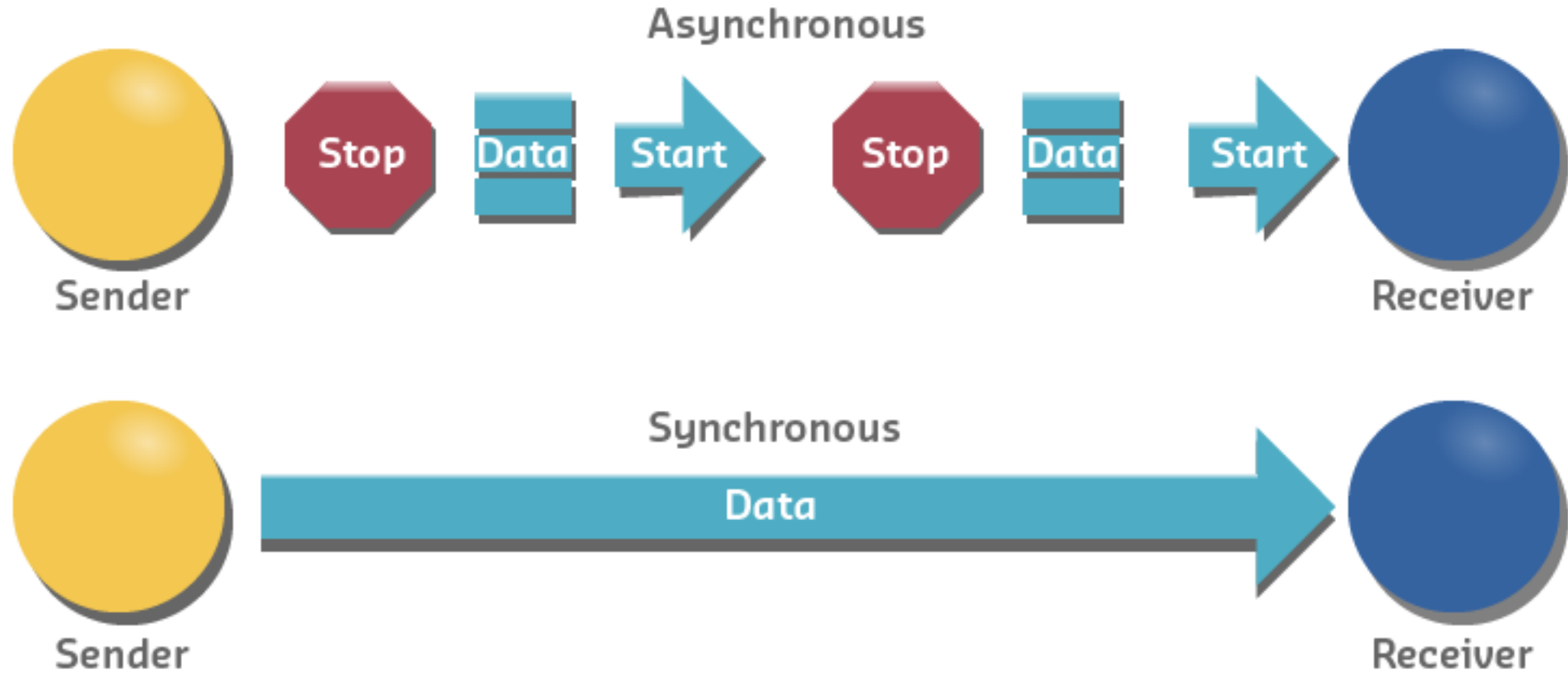
Aprendizaje

- Incluye la fijación y la creación de conceptos compuestos que contienen los conceptos de partes de un objetivo

¿Cómo hacer nuestro sistema inteligente?



Periodicidad de los datos



Ejemplo de un sistema inteligente

Moisture **sensors** in the soil send information about moisture levels.



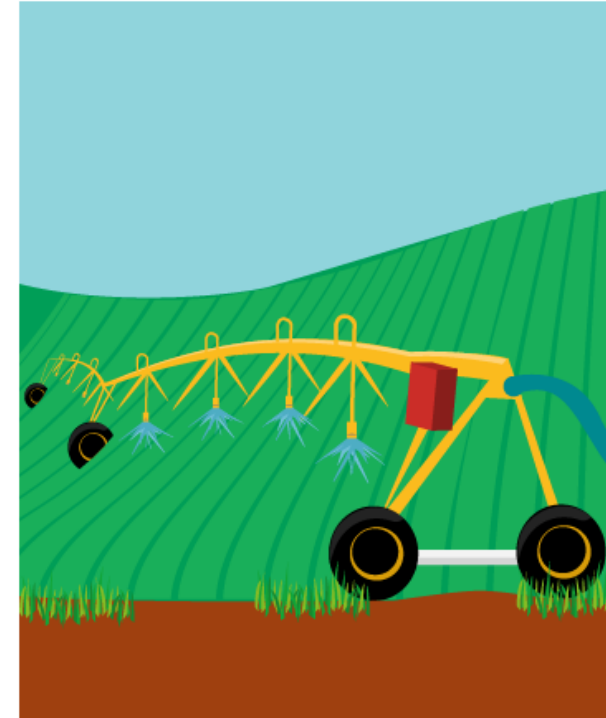
Information from online weather services is also provided.



The system decides if it needs to water and for how long.



Smart Sprinkler System



It then sends the command to the **actuators** (the valve and the solenoid) which start or stop the water flow.