

Sistemas Embebidos en Tiempo Real

Maestría en Inteligencia Artificial
2024

1

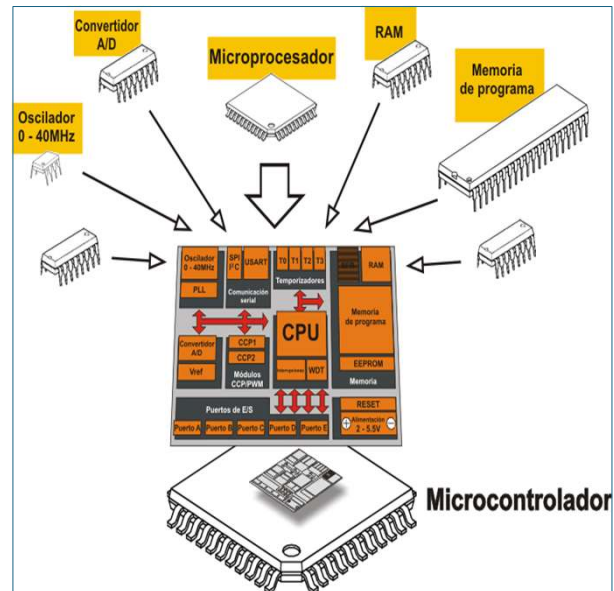
Sesión 12/11/2024

- Primeras implementaciones en físico con el Arduino UNO
- Conexión a sensores y actuadores

2

¿Por qué microcontroladores en Sistemas Embebidos?

- Los microcontroladores son la parte esencial de un sistema embebido.
- A diferencia de una computadora, para que un microcontrolador cumpla la función de un sistema embebido es necesario conectarle sensores y actuadores.



3

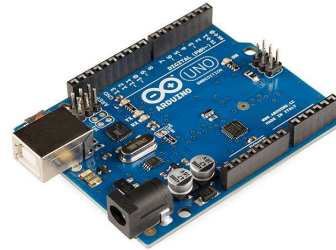
¿Microcontrolador? ¿Microprocesador?

- Un microcontrolador está encaminado básicamente hacia aplicaciones concretas en donde, el espacio, y número de componentes es mínimo, además, los cambios o ampliaciones futuras del sistema son casi nulos. Por otro lado, un microprocesador se destina a sistemas donde su expansión a corto o mediano plazo es factible. A pesar de que un microprocesador es más rápido que un microcontrolador para la ejecución de sus instrucciones, en la mayoría de los casos es necesario interconectarlo con dispositivos periféricos.
- Un microcontrolador, puede ser utilizado con un mínimo número de componentes en trabajos específicos y en un amplio rango de aplicaciones, tales como; los sistemas de control de alarmas, tableros de control en la industria automotriz, en la instrumentación médica, en los teclados de computadora, en los sistemas portátiles de almacenamiento de datos, en equipos de laboratorio, etc.

4

Arduino

- Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.
- El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de multiples diseños.
- Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

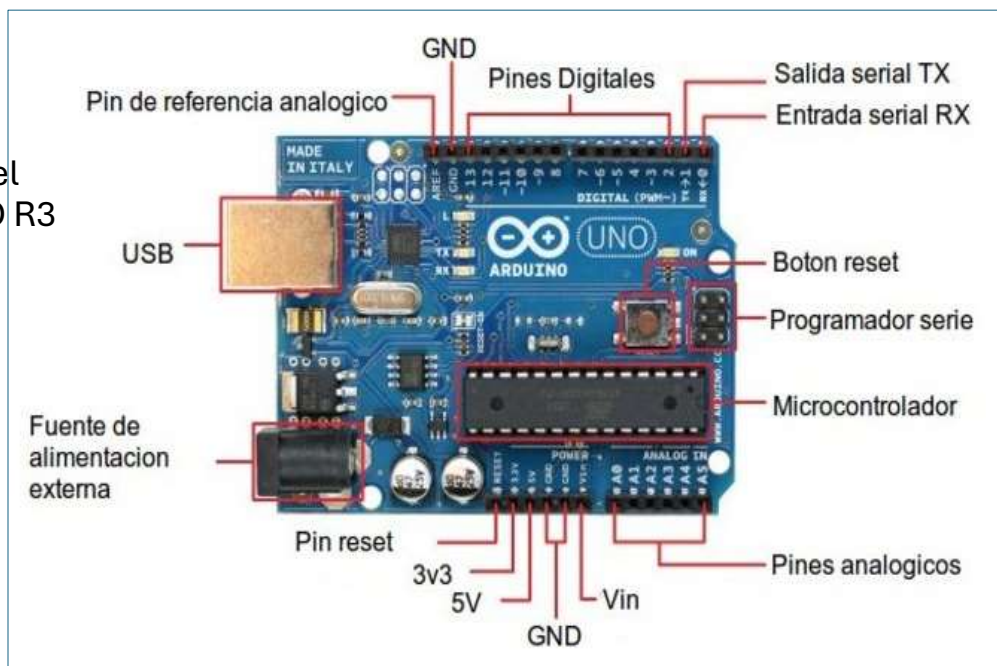


Microcontrolador	ATMEGA328
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7V-12V
Límites máximo y mínimo de Vin	6V – 20V
Pines Input/Output digitales	14 (6 para salida PWM) de 40mA máximo cada uno
Pines de entrada Analógico	6
Memoria de programa FLASH	32kB (0,5 usado para Bootloader)
Memoria SRAM	2kB
Velocidad de reloj (Clock)	16MHz

5

Arduino

- Estructura del Arduino UNO R3



6

Arduino shields:

- En Arduino, los Shields son tarjetas electrónicas especialmente diseñadas para conectarse con las placas base de Arduino.
- Estas placas son creadas para funciones específicas y normalmente vienen acompañadas con librerías (software) para poder utilizarlas fácilmente.
- Ejemplos de algunos shields:

- Arduino GSM Shield
- Arduino Ethernet Shield
- Arduino WiFi Shield
- Arduino Wireless SD Shield
- Arduino USB Host Shield
- Arduino Motor Shield



7

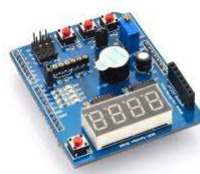
Arduino shields:



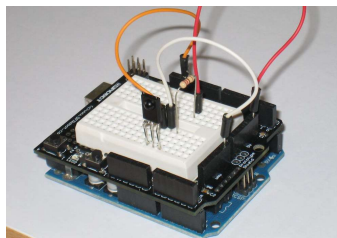
Relay board



ESP8266 board



I/O shield



Breadboard shield



DHT11 shield

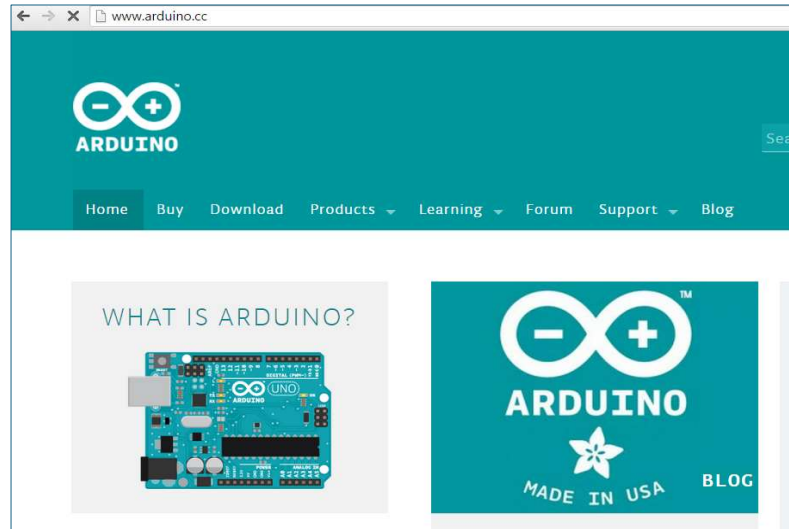


LCD Shield

8

Arduino IDE

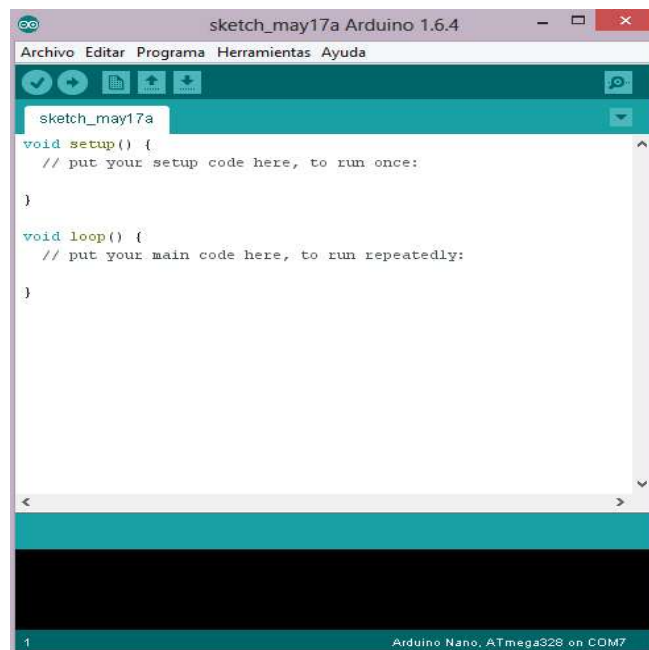
- Entorno de desarrollo para Arduino (freeware).
- Soporta diferentes plataformas tanto Arduino como plataformas third-party
- El lenguaje de programación es similar al C
- Extenso repertorio de librerías



9

Arduino sketch

- Los sketch son los archivos fuente y tienen extensión *.ino



10

Estructura de un Sketch

```
int pin;    // crea la variable de tipo entero pin
int numero = 1; // crea la variable de tipo entero numero y le a
// valor 1
```

Área para declarar y definir variables

```
void setup()
{
  ...
  ...
  ...
}
```

Área de configuración
Setup () es un conjunto de código que solo se ejecuta una vez cuando Arduino arranca

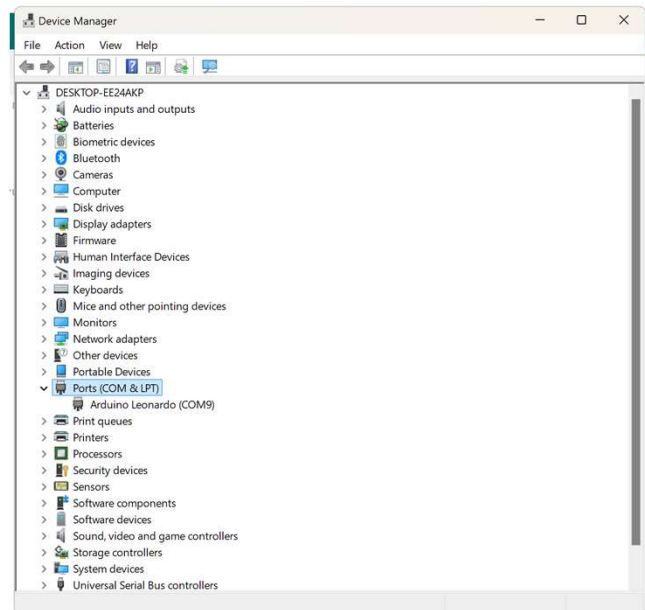
```
void Loop()
{
  ...
  ...
  ...
}
```

Loop() Es la parte principal del sketch, aquí se encuentra el código que el microcontrolador ejecutará en forma repetitiva mientras esté energizado

11

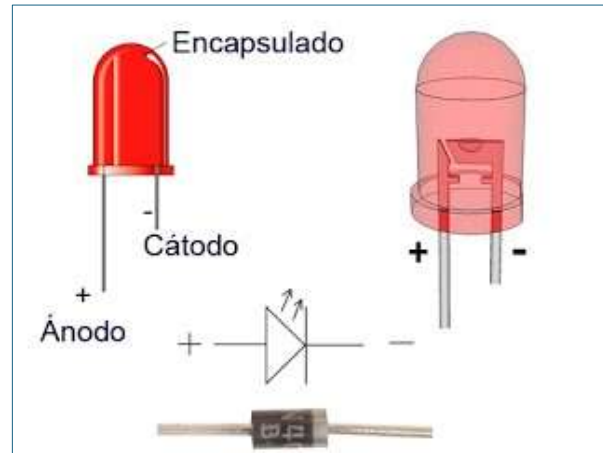
Arduino

- Es importante abrir el Administrador de Dispositivos del Windows para verificar si se detectó correctamente el Arduino en la PC y se creó el puerto serial



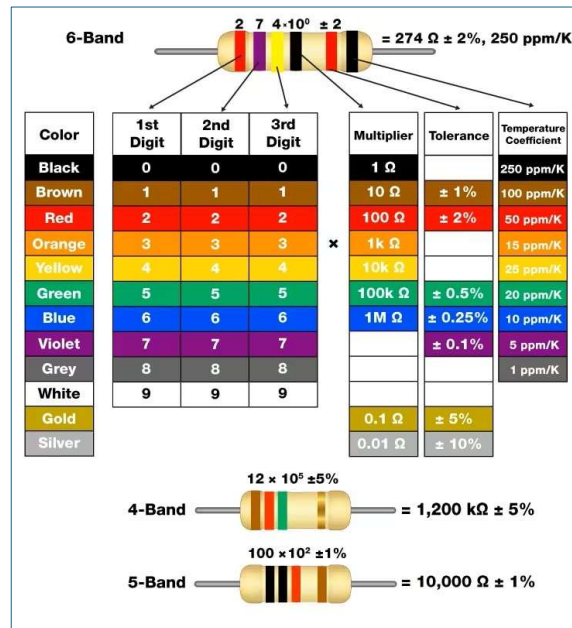
12

Dispositivos LED



13

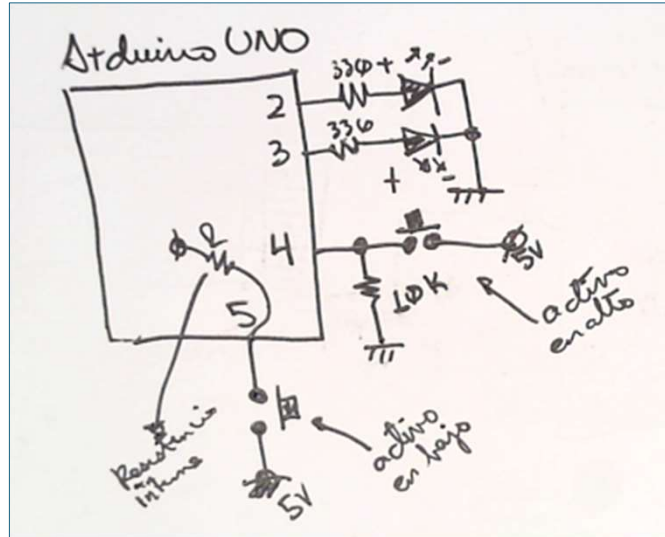
Código de colores de las resistencias



14

Arduino

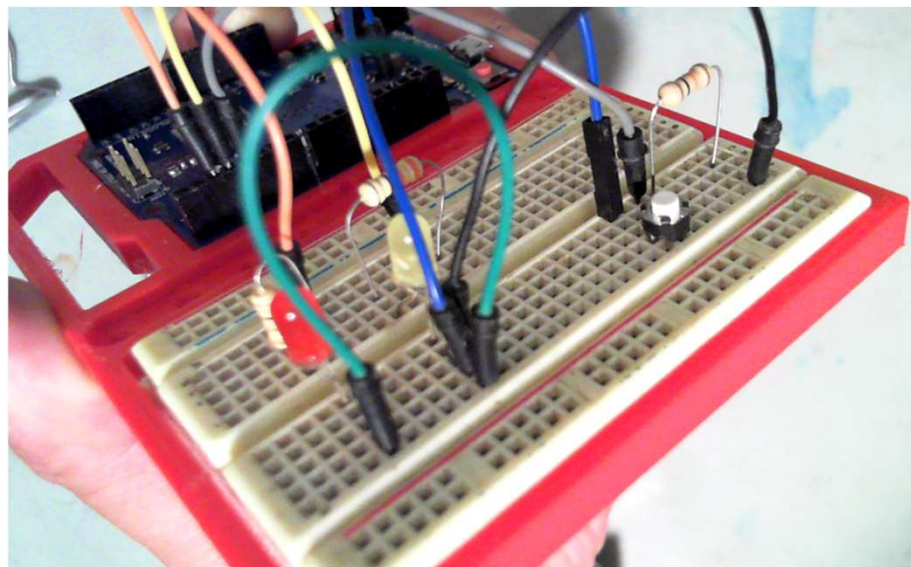
- Ejemplo de implementación



15

Arduino

- Ejemplo de implementación



16

Arduino

- Ejemplo de implementación

```

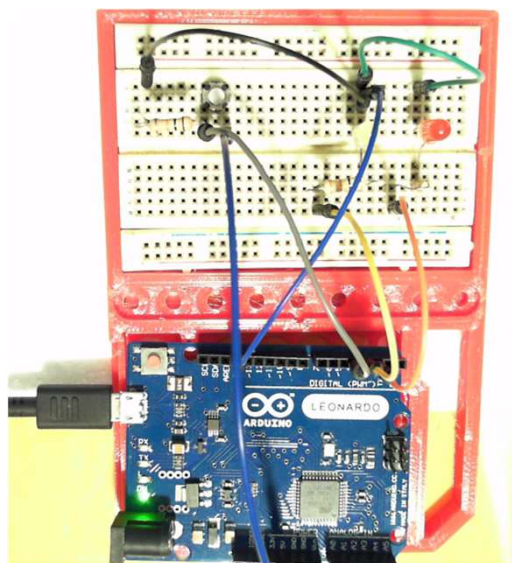
sketch_nov12a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Leonardo
sketch_nov12a.ino
1 //Primera prueba de circuito en físico empleando Arduino UNO
2 //Desarrollado por Kalun
3 //Noviembre del 2024
4
5 void setup() {
6   // put your setup code here, to run once:
7   pinMode(2, OUTPUT); //pin2 como salida
8   pinMode(3, OUTPUT); //pin3 como salida
9   pinMode(4, INPUT);  //pin4 como entrada
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14   if(digitalRead(4) == 1){
15     //cuando es verdad
16     digitalWrite(2, HIGH);
17     digitalWrite(3, LOW);
18     delay(100);
19     digitalWrite(2, LOW);
20     digitalWrite(3, HIGH);
21     delay(100);
22   }
23   else{
24     //cuando es falso
25     digitalWrite(2, LOW);
26     digitalWrite(3, LOW);
27   }
28 }
Output

```

17

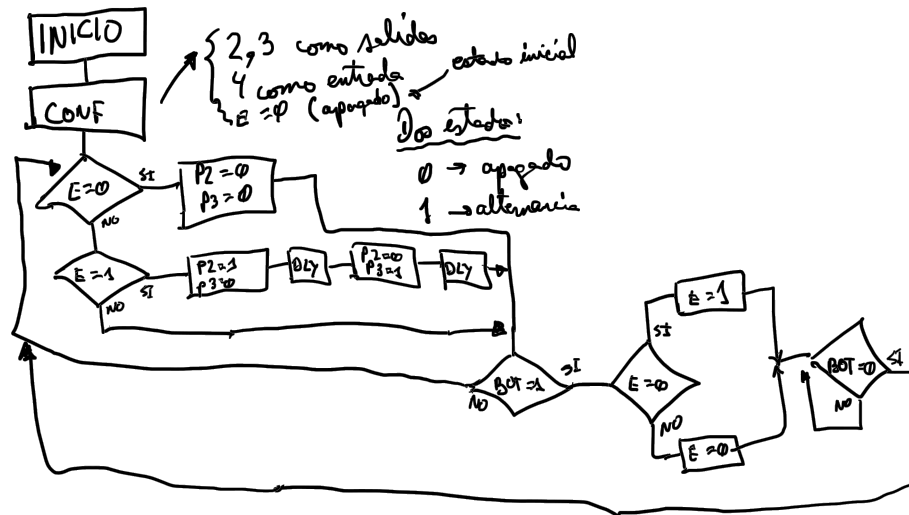
Arduino

- Ejemplo de implementación



18

¿Cómo haría para que se enclave la función de alternancia de los LEDs en el ejemplo anterior?



19

¿Cómo haría para que se enclave la función de alternancia de los LEDs en el ejemplo anterior?

```

6  byte estado = 0; //estado inicial es apagado
7
8  void setup() {
9    // put your setup code here, to run once:
10   // voy a configurar primero los pines de los LEDs como salida
11   pinMode(2, OUTPUT); //pin2 como salida LED rojo
12   pinMode(3, OUTPUT); //pin3 como salida LED amarillo
13   pinMode(4, INPUT); //pin4 como entrada pulsador
14 }
15
16 void loop() {
17   // put your main code here, to run repeatedly:
18   if(estado == 0){
19     digitalWrite(2, LOW);
20     digitalWrite(3, LOW);
21   }
22   else{
23     digitalWrite(2, HIGH);
24     digitalWrite(3, LOW);
25     delay(500);
26     digitalWrite(2, LOW);
27     digitalWrite(3, HIGH);
28     delay(500);
29   }
30   if(digitalRead(4) == 1){
31     if(estado == 0){
32       estado = 1;
33     }
34     else{
35       estado = 0;
36     }
37   }
38   while(digitalRead(4) == 1);
39 }

```

20

Fin de la sesión