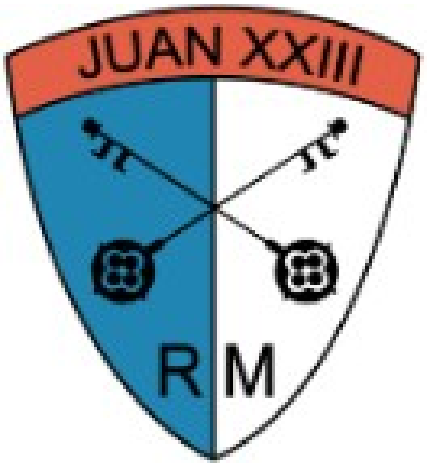


INSTITUTO JUAN XXIII

TALLER

Procedimientos Técnicos

Arduino

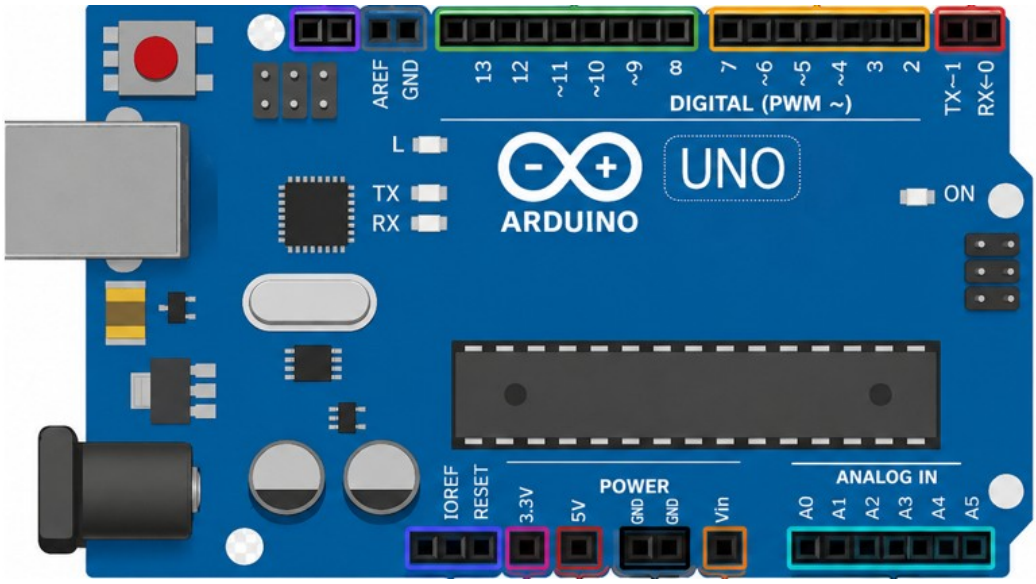
2º año

Introducción

Arduino es una plataforma de software y hardware libres basada en una placa que incluye circuitos electrónicos con entradas y salidas, analógicas y digitales, que conectan el mundo físico con el mundo virtual, a partir de la cual se pueden crear objetos electrónicos interactivos. Con estas placas es posible enviar o recibir información de cualquier sistema electrónico conectado a ellas. Son capaces de leer entradas (ejemplo: sensores y botones) y convertirlas en una salida (ejemplo: encender luces y motores).

Arduino UNO

A continuación, se explican los pines que posee el Arduino UNO para realizar las conexiones en las prácticas.



Pines digitales (2 a 7 y 8 a 13): Son entradas o salidas digitales. El pin 13, además es un led integrado en el Arduino.

Los pines que empiezan con **~** son los denominados PWM (Pulse Width Modulation).

Pines 0 (RX) y 1(TX): cumplen la función de recepción y transmisión en la comunicación con otros módulos mediante el uso del puerto Serie. Se pueden usar como pines digitales, pero existen ciertas reglas estrictas y limitaciones.

Pines GND: Son la masa o tierra del circuito (0V).

Pines analógicos (A0 a A5): son únicamente entradas analógicas.

Pin Vin: Entrada de voltaje externo.

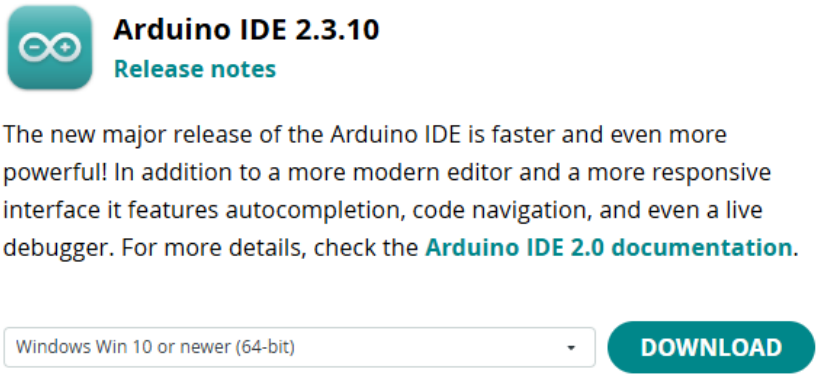
Pines de 3.5V y 5V: son salidas reguladas a esos voltajes.

NOTA: Para poder realizar las conexiones, se recomienda utilizar cables Dupont.

Instalación Arduino IDE

Para poder trabajar con el Arduino UNO de forma física, utilizamos Arduino IDE. Actualmente, se encuentra disponible para ser descargada del siguiente link:

https://www.arduino.cc/en/software/#ide

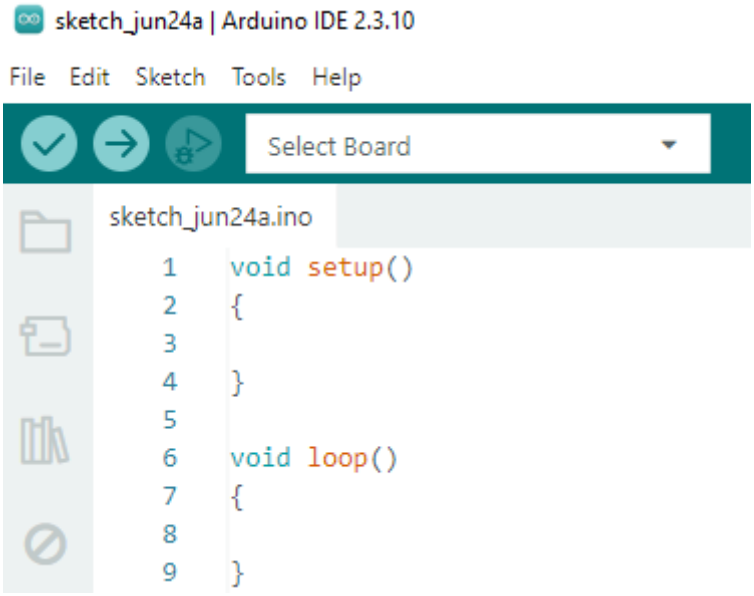


Debe seleccionarse el Sistema Operativo que tenga la computadora donde se utilizará y presionar DOWNLOAD.

Tras realizar la instalación y ejecutarlo, es posible que solicite permisos de Firewall y también, instalar algunos complementos. Es importante dar los permisos e instalar todo lo solicitado ya que utilizará los puertos serie (USB) para poder comunicarse entre el Arduino UNO y la computadora.

Entorno Arduino IDE

Al abrir Arduino por primera vez, nos encontraremos con un archivo que muestra las siguientes funciones:



La función **void setup( )** se ejecuta una única vez al iniciar el programa y se utiliza para realizar configuraciones iniciales.

La función **void loop( )** se ejecuta repetidamente en bucle y es donde debemos programar lo que queremos que suceda.

IMPORTANTE: para abrir y cerrar estas funciones se utilizan las { }. Si falta una de ellas, el programa tirará un error ya que no puede delimitar su comienzo y su fin. También es necesario respetar las mayúsculas y minúsculas en todas las líneas de código tal como se indican ya que si no indicará error de sintaxis.

Por encima de la función void setup( ) suele realizarse la declaración de variables que utilizará el programa.

Tipos de variables

Existen muchos tipos de variables, los más utilizados en nuestras prácticas son:

- Para números enteros (sin coma): **int**
- Para números reales (números enteros muy grandes o con coma): **float**
- Para caracteres (letras, símbolos): **char**

Ejemplos de declaración:

```
int num;
char caracter;
float coma;
```

IMPORTANTE: toda línea de código que no sea una función, estructura condicional o de repetición (explicado más adelante), entre otras que se verán otros años, lleva punto y coma (;) al finalizar la misma.

Función void setup ( )

Dentro de esta función, configuramos la velocidad en baudios del puerto USB que permite enviar y recibir datos, así como también los pines que se utilizarán y cómo serán utilizados (entradas o salidas).

Ejemplo:

```
void setup()
{
Serial.begin(9600);   //configura velocidad a 9600 baudios
pinMode(8, INPUT);    //configura pin 8 como entrada
pinMode(13, OUTPUT);  //configura pin 13 como salida
}
```

Función void loop ( )

En esta función deberemos indicar qué queremos que realice nuestro programa. Para ello utilizaremos algunas funciones que se explican en el siguiente ejemplo:

```
1   void setup()
2   {
3   pinMode(3, OUTPUT);     //configura pin 3 como salida
4   }
5
6   void loop()
7   {
8   digitalWrite(3, HIGH);  //envía un 1 al pin 3 (tensión)
9   delay(1000);            //espera 1 segundo
10  digitalWrite(3, LOW);   //envía un 0 al pin 3 (0v)
11  delay(1000);            //espera 1 segundo
12  }
```

HIGH y **LOW** se utilizan para indicar o no el paso de tensión y así lograr que, por ejemplo, un LED encienda.

La función **delay( )** trabaja en milisegundos (1s = 1000ms) y genera una pausa del tiempo indicado. Si no ponemos un delay( ), no se podrá percibir por el ojo humano el encendido y apagado del mismo.

Recordar que, al estar dentro del void loop( ), esto se repetirá infinitamente hasta que se frene el programa o se desconecte el Arduino.

Comunicación puerto serie

Para entender cómo funciona, desarrollamos un ejercicio el cual implementará el uso de un botón y un LED mediante la siguiente lógica:

ACCIÓN	RESULTADO
Presionar el botón	Envía **s** por el puerto serie
Puerto serie recibe A	Se enciende el LED
Puerto serie recibe B	Se apaga el LED