

*Programación para mecatrónica

- **C/C++:** Son lenguajes de bajo nivel, lo que significa que ofrecen un control muy preciso sobre el hardware. Son extremadamente rápidos y eficientes. La mayoría de los microcontroladores, como los de la plataforma Arduino, se programan en una variante de C++. Son ideales para sistemas en tiempo real donde la velocidad es crítica.
- **Python:** Es un lenguaje de alto nivel, conocido por su sintaxis clara y legible. Es muy popular para aplicaciones de robótica, análisis de datos, visión por computadora e inteligencia artificial. Es menos eficiente en sistemas embebidos, pero es excelente para desarrollar la lógica de control o la interfaz de usuario en una computadora que se comunica con un robot.
- **Lenguaje de Arduino (Basado en C++):** Es una versión simplificada de C++ diseñada para la plataforma Arduino. Su facilidad de uso y la gran cantidad de bibliotecas disponibles lo hacen perfecto para principiantes y para la creación rápida de prototipos.

Código

```
// Se incluye la biblioteca para controlar el servo motor
#include <Servo.h>

// Se crea un objeto 'Servo' llamado 'miServo'
Servo miServo;

// La función 'setup' se ejecuta una sola vez al inicio
void setup() {
  // Se le indica al servo en qué pin digital está conectado (ej. pin 9)
  miServo.attach(9);
}

// La función 'loop' se ejecuta continuamente
void loop() {
  // Mueve el servo a la posición de 0 grados
  miServo.write(0);
  // Espera 1000 milisegundos (1 segundo)
  delay(1000);

  // Mueve el servo a la posición de 90 grados
  miServo.write(90);
  // Espera 1000 milisegundos (1 segundo)
  delay(1000);
}
```

Explicación de código

#include <Servo.h>: Esta línea importa la biblioteca que contiene las funciones necesarias para controlar un servo.

Servo miServo;: Declara una variable de tipo Servo que representa nuestro motor.

`void setup()`: Aquí se configura el hardware. `miServo.attach(9)`; le dice al programa que el servo está conectado al pin digital 9 del Arduino.

`void loop()`: El código dentro de esta función se repite indefinidamente.

`miServo.write(0)`:: Envía un comando al servo para que se mueva a la posición de 0 grados.

`delay(1000)`:: Pausa la ejecución del programa durante 1000 milisegundos.

`miServo.write(90)`:: Mueve el servo a la posición de 90 grados.

Otra opción de programación

C / C++ (Arduino)

- **Uso principal:** Es el lenguaje estándar para programar microcontroladores como Arduino. De hecho, el lenguaje Arduino está basado en C/C++ con librerías específicas para manejo de hardware.
- **Estructura básica:**
 - `void setup() {`
 - `// Se ejecuta una vez al iniciar`
 - `}`
 - `void loop() {`
 - `// Se ejecuta de forma repetitiva`
 - `}`

Python

- **Uso en mecatrónica:** Ideal para desarrollo rápido, análisis de datos, visión artificial e integración con ROS (Robot Operating System) y sistemas IoT
- **Ejemplo con ROS:**
 - `import rospy`
 - `from std_msgs.msg import Float64`
 - `rospy.init_node('robot_arm_controller')`
 - `pub = rospy.Publisher('/robot_arm/joint1_position_controller/command', Float64, queue_size=10)`
 - `def move_robot_arm(position):`
 - `pub.publish(position)`
 - `if __name__ == '__main__':`
 - `move_robot_arm(0.5) # Mover a 0.5 radianes`

Publica comandos de posición para controlar un brazo robótico

- **Conexión Arduino-Python:** Puedes controlar placas Arduino desde Python usando librerías como pySerial, ideal para prototipos y pruebas

MicroPython

- Versión reducida de Python optimizada para microcontroladores (Arduino, ESP32, etc.), usable en sistemas con recursos limitados
- **Ejemplo simple:**
- `print("Hello, World!")`

Y controlar un LED:

```
import pyb
```

```
pyb.LED(1).on()
```

Otros entornos útiles

- **MATLAB / Simulink:** Muy usado en educación e investigación. Permite diseñar algoritmos de control y simular sistemas antes de implementarlos en hardware (por ejemplo usando placas Arduino)
- **Java, MATLAB y Python en control industrial:** C/C++ para control en tiempo real, mientras que Python o MATLAB se usan para análisis de datos o aprendizaje automático
- **Entornos gráficos (visual programming):** Facilitan la enseñanza, especialmente a principiantes. Ejemplos como miniBloq permiten programar robots Arduino mediante bloques visuales