**Texto, Logotipo

Descripción generada automáticamente**

**“TRABAJO PRÁCTICO #2:** **Fundamentos de Programación IoT y Módulos de Desarrollo”**

**Asignatura:** “PROYECTO INTEGRADOR”

**Fecha:** 04/05/2024

**Autor**:

Paola Natalia Alejandra, PANTOJA

**Profesor:**

Técnico superior en Mecatrónica C. GONZALO VERA

Arduino es uno de los microcontroladores de código abierto más populares, está basado en hardware y software fáciles de usar. Esta es la primera diferencia con otras placas y microcontroladores.

Sirve para desarrollar elementos autónomos, o bien conectarse a otros dispositivos o interactuar con otros programas tanto con el hardware como con el software.

Ejemplo: Un motor que suba o baje una persiana basada en la luz que haya y gracias a un sensor conectado al Arduino, o bien para transformar la información de una fuente, como puede ser un teclado, y convertirá la información a algo que entienda por ejemplo un ordenador.

Características principales:

1. **Lenguaje de programación**: Utiliza un lenguaje de programación basado en C/C++, que es relativamente sencillo de aprender y muy accesible para principiantes.
2. **IDE (ENTORNO DE DESARROLLO INTEGRADO)**: Arduino IDE proporciona una interfaz simple y fácil de usar para escribir y cargar código en placas Arduino.
3. **Amplia compatibilidad de hardware:** Es compatible con una amplia gama de placas de desarrollo, desde las más simples como Arduino Uno hasta las más avanzadas como Arduino Due y Arduino Mega.
4. **Librerías y comunidad:** Cuenta con una gran cantidad de librerías y una comunidad activa de usuarios.

MycroPython (RT-Thead):

Nació en 2006 y es una implementación de Python 3 optimizada para microcontroladores y sistemas embebidos. Está escrito principalmente en lenguaje C, fácil de entender y fácil de portar (puede ser rápidamente portado a una amplia gama de MCUs y Chips de módulos).

Características:

1. Interpretación directa: MycroPython permite ejecutar código Python directamente en microcontroladores, lo que simplifica el desarrollo al utilizar un lenguaje de alto nivel como Python.
2. Eficiencia y rendimiento: Aunque Python es conocido por su simplicidad y legibilidad, MycroPython está diseñado para ejecutarse eficientemente en dispositivos con recursos limitados.
3. Soporte de hardware: Es compatible con una variedad de placas de desarrollo, incluyendo algunas basadas en ESP32.
4. Facilidad de integración. Es altamente portable y puede integrarse fácilmente con otros sistemas y protocolos, lo que lo hace ideal para aplicaciones IoT.

Diferencias y aplicaciones en IoT

La principal diferencia entre Arduino y MicroPython está en los lenguajes de programación. Al contrario del lenguaje de Arduino, que es una versión de C/C++ que se compila en la computadora y se transfiere luego como código binario a la placa, un programa en MicroPython se transfiere como código fuente a la placa y recién allí es interpretado y convertido instrucción por instrucción a código binario. Es decir, Arduino utiliza C/C++ y MicroPython utiliza Python.

Este funcionamiento de MicroPython tiene sus ventajas y desventajas cuando lo comparamos con la programación en Arduino.

**Ventajas**: Es más rápido hacer pruebas y prototipos, porque no es necesario volver a compilar el programa cada vez que hacemos una modificación.

**Desventajas**: Se requiere más memoria en la placa, porque se almacena el programa como código fuente (texto) lo que ocupa más lugar que un código binario y además el intérprete de MicroPython debe estar grabado en la placa.

Arduino es ideal para proyectos que requieren un control preciso de hardware y un rendimiento optimizado, especialmente cuando se trabaja con sensores y actuadores específicos. Por otro lado, MicroPython es más adecuado para proyectos donde la simplicidad de la programación y la rápida prototipación son prioritarias, o cuando se requiere integración con otros sistemas basados en Python.

En términos de aplicaciones en IoT, ambas plataformas ofrecen diferentes ventajas, se pueden utilizar en diferentes disciplinas como la programación de robótica, medicina, el cuidado ambiental, y demás.

Se pueden utilizar a su vez diferentes sensores y actuadores para medir y controlar los objetos del proyecto que se realice.

<https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/espressif-systems/esp32/ESP32-SHIELD-detail>

CONTIUNAR ACÁ, VAMOS A INTENTAR HACER EL REPOSITORIO

<https://github.com/RT-Thread/rt-thread/blob/master/README_es.md>

<https://www.genbeta.com/desarrollo/que-micropython-lenguaje-programacion-que-puedes-usar-tu-arduino-probar-tu-navegador>

<https://docs.arduino.cc/arduino-cloud/guides/micropython/>