

Trabajo Práctico #2

Fundamentos de Programación IoT y Módulos de Desarrollo

1- Investigación de las plataformas de desarrollo:

Documentar las características principales de Arduino (PlatformIO) y MicroPython (RT-Thread), destacando sus diferencias y aplicaciones en IoT.

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. Algunas de sus características principales:

- Hardware asequible y accesible.
- Las placas Arduino están basadas en microcontroladores AVR o ARM. Estos microcontroladores son potentes y versátiles.
- Ecosistema de software robusto: Arduino cuenta con un entorno de desarrollo integrado (IDE) que es fácil de usar y está disponible de forma gratuita para Windows, Mac y Linux.
- Lenguaje de programación sencillo: Arduino utiliza un dialecto simplificado de C/C++ como lenguaje de programación.
- Amplia gama de shields y módulos.
- Comunidad activa y soporte abundante.
- Flexibilidad y versatilidad

En resumen, Arduino es una plataforma versátil, accesible y fácil de usar que permite a cualquier persona, desde principiantes hasta expertos, crear proyectos electrónicos de manera rápida y sencilla. Su combinación de hardware asequible, software amigable y una gran comunidad de usuarios lo convierte en una opción popular para proyectos de electrónica en todo el mundo.

PlatformIO es un ecosistema de código abierto para el desarrollo de IoT centrado en microcontroladores y computadoras de placa única como Arduino, ESP8266, ESP32 y otros. Proporciona una experiencia de desarrollo unificada en diferentes plataformas de desarrollo y arquitecturas de microcontroladores.

A continuación, se ofrece las características y componentes clave de PlatformIO:

- IDE multiplataforma.
- Administrador de biblioteca.
- Sistema de compilación unificado.
- Configuración del proyecto.
- Integración con sistemas de control de versiones.
- Soporte para múltiples marcos.
- Depuración y pruebas unitarias.

PlatformIO simplifica el proceso de desarrollo de proyectos de IoT al proporcionar una plataforma unificada y fácil de usar que abstrae muchas de las complejidades asociadas con el desarrollo multiplataforma para microcontroladores.

En resumen, PlatformIO es una herramienta poderosa para desarrollar proyectos de Arduino, ofreciendo un entorno de desarrollo integrado, soporte para múltiples placas y microcontroladores, funciones avanzadas de autocompletado y depuración, gestión de bibliotecas, y más. Es una opción excelente para aquellos que buscan llevar sus proyectos de Arduino al siguiente nivel.

MicroPython es una implementación ligera del lenguaje de programación Python optimizada para ejecutarse en microcontroladores y sistemas integrados. Permite a los desarrolladores escribir código Python que puede interactuar directamente con el hardware, lo que lo convierte en una opción popular para proyectos de IoT y desarrollo integrado.

Aquí tienes algunas características principales de MicroPython:

- Pequeño y Eficiente.
- Interfaz con Hardware.
- Sintaxis de Python.
- Soporte Multiplataforma.
- Librerías y Módulos Adicionales
- Licencia Abierta.

RT-Thread, por otro lado, es un sistema operativo en tiempo real (RTOS) de código abierto que proporciona un amplio conjunto de componentes y herramientas para construir sistemas integrados. Está diseñado para ser altamente configurable y modular, lo que permite a los desarrolladores adaptar el sistema operativo a los requisitos específicos de sus proyectos.

Estas son algunas de las funciones y características clave de RT-Thread:

- Multitarea en tiempo real.
- Arquitectura Modular y Escalable.
- Soporte Multiplataforma.
- Rico Conjunto de Componentes y Middleware.
- Bajo Consumo de Recursos.
- Desarrollo de Aplicaciones Basado en Hilos.
- Soporte para Desarrollo y Depuración

La integración de MicroPython con el sistema operativo RT-Thread permite a los desarrolladores aprovechar las características y capacidades de ambas plataformas.

RESUMEN: Arduino con PlatformIO y MicroPython con RT-Thread son dos enfoques diferentes para el desarrollo de aplicaciones IoT (Internet de las cosas), cada uno con sus propias características y ventajas. La elección entre ellas dependerá de las necesidades específicas del proyecto y las preferencias del desarrollador.

Comparación de las diferencias claves y las aplicaciones en IoT de ambos:

Arduino con PlatformIO:

- **Lenguaje de Programación:** Arduino se basa en el lenguaje de programación Wiring, que es similar a C/C++. Con PlatformIO, los desarrolladores pueden escribir código en C/C++ utilizando el entorno de desarrollo integrado (IDE) de PlatformIO, que proporciona características avanzadas como la gestión de bibliotecas, la depuración integrada y el soporte para múltiples plataformas de hardware.
- **Facilidad de Uso:** Arduino es conocido por su facilidad de uso y su enfoque orientado a principiantes. Es ideal para aquellos que están comenzando en el mundo de la electrónica y la programación.
- **Compatibilidad con una amplia gama de hardware:** Arduino es compatible con una amplia variedad de placas de desarrollo, incluyendo las populares placas Arduino UNO, Nano, Mega, y muchas otras. Esto permite a los desarrolladores elegir la plataforma que mejor se adapte a sus necesidades.
- **Librerías y Ecosistema:** Arduino cuenta con un amplio ecosistema de librerías y herramientas desarrolladas por la comunidad, lo que facilita la creación de proyectos IoT utilizando una variedad de sensores, actuadores y dispositivos periféricos.

MicroPython con RT-Thread:

- **Lenguaje de Programación:** MicroPython es una implementación de Python diseñada para sistemas embebidos. Permite a los desarrolladores escribir código en Python, un lenguaje de programación de alto nivel y fácil de aprender, lo que simplifica el proceso de desarrollo y reduce el tiempo de desarrollo.
- **Eficiencia y Bajo Consumo de Recursos:** MicroPython está optimizado para ejecutarse en dispositivos con recursos limitados, como microcontroladores. Esto lo hace ideal para aplicaciones IoT donde se requiere eficiencia en el uso de recursos y bajo consumo de energía.
- **Soporte para Hardware Embebido:** MicroPython ofrece un conjunto de librerías y módulos que permiten interactuar directamente con el hardware del dispositivo, como GPIO, UART, SPI, y más. Esto facilita el control y la comunicación con sensores, actuadores y otros dispositivos periféricos.
- **RTOS (Sistema Operativo en Tiempo Real):** RT-Thread es un sistema operativo en tiempo real que proporciona capacidades avanzadas de multitarea y gestión de recursos. Al combinar MicroPython con RT-Thread, los desarrolladores pueden crear aplicaciones IoT complejas que requieren manejo de tareas concurrentes y gestión eficiente de recursos.

Aplicaciones en IoT:

- **Arduino con PlatformIO:** Ideal para proyectos IoT simples y prototipos rápidos donde la facilidad de uso y la compatibilidad con una amplia gama de hardware son importantes. Puede ser utilizado en aplicaciones como monitoreo ambiental, control de dispositivos domésticos, y más.
- **MicroPython con RT-Thread:** Adecuado para aplicaciones IoT más avanzadas que requieren eficiencia en el uso de recursos y la capacidad de manejar tareas concurrentes. Puede ser utilizado en aplicaciones como sistemas de automatización industrial, dispositivos médicos, y más.

En resumen, tanto Arduino con PlatformIO como MicroPython con RT-Thread son herramientas poderosas para el desarrollo de aplicaciones IoT, cada una con sus propias fortalezas y casos de uso específicos.

Investigar sobre diferentes módulos shield disponibles para ESP32 y su aplicación en controladores IoT.

El ESP32 es un microcontrolador de bajo costo y alto rendimiento que se ha vuelto muy popular en el desarrollo de proyectos IoT debido a su potencia y versatilidad.

Los módulos Shield son placas de expansión que se conectan directamente al ESP32 para agregar funcionalidades adicionales. Aquí tienes algunos módulos Shield comunes disponibles para el ESP32 y sus posibles aplicaciones en controladores IoT:

- 1- ESP32-CAM Shield: Este módulo incluye una cámara OV2640 y una ranura para tarjeta microSD. Es ideal para aplicaciones de visión por computadora y vigilancia en tiempo real, como cámaras de seguridad, sistemas de reconocimiento facial y sistemas de monitoreo remoto.
- 2- ESP32 OLED Display Shield: Este módulo cuenta con una pantalla OLED integrada que puede mostrar texto, gráficos y datos en tiempo real. Es útil para crear interfaces de usuario simples y mostrar información como el estado del sistema, datos de sensores y notificaciones en dispositivos IoT.
- 3- ESP32 LoRa Shield: Este módulo agrega funcionalidad de comunicación de largo alcance mediante el uso del protocolo LoRa. Es útil para aplicaciones que requieren comunicación inalámbrica de larga distancia, como sistemas de telemetría, redes de sensores remotos y dispositivos de seguimiento.
- 4- ESP32 Relay Shield: Este módulo incluye relés que permiten controlar dispositivos eléctricos de alto voltaje y corriente. Es útil para aplicaciones de automatización del hogar, control de iluminación, sistemas de riego automático y control de dispositivos industriales.
- 5- ESP32 Motor Driver Shield: Este módulo incluye controladores de motor que permiten controlar motores DC, motores paso a paso y servomotores. Es útil para aplicaciones de robótica, control de vehículos, sistemas de posicionamiento y control de movimiento.
- 6- ESP32 Environmental Sensor Shield: Este módulo incluye sensores de temperatura, humedad, presión atmosférica y calidad del aire. Es útil para aplicaciones de monitoreo ambiental, agricultura inteligente, sistemas de control de clima interior y sistemas de gestión de la calidad del aire. Estos son solo algunos ejemplos de los módulos Shield disponibles para el ESP32 y sus posibles aplicaciones en controladores IoT. La elección del módulo depende de los requisitos específicos del proyecto, como la funcionalidad requerida, el presupuesto, el consumo de energía y la facilidad de integración.