

Trabajo practico n2

Nombres: Merwil Gustavo

Apellidos: Quintana Becerra

Arduino y MicroPython:

Arduino y MicroPython Arduino está agregando el lenguaje Python como una opción adicional para programar microcontroladores. La plataforma de elección es MicroPython1. Para cargar scripts de MicroPython en tu placa, necesitas usar un editor de código. Hay dos editores para elegir: Arduino Lab para MicroPython y OpenMV1.

Programando Arduino con MicroPython Aquí te dejo un resumen de los pasos para programar un Arduino Nano RP2040 Connect con Python usando el Arduino Lab para MicroPython IDE2:

Descarga el Arduino Lab para MicroPython IDE desde el sitio web de Arduino e instálalo en tu computadora.

Descarga el firmware de MicroPython para la placa RP2040 desde el sitio web de MicroPython y guárdalo como un archivo .uf2.

Conecta los pines REC y GND de la placa con un cable puente y conecta la placa a tu computadora con un cable micro USB. La placa aparecerá como un dispositivo de almacenamiento en tu computadora.

Arrastra y suelta el archivo .uf2 en la carpeta de la placa para flashear el firmware de MicroPython en la placa. Puedes quitar el cable puente después de este paso.

Abre el Arduino Lab para MicroPython IDE y conecta la placa al IDE. Puedes crear un nuevo archivo para MicroPython y escribir tu código en el editor.

Ejecuta el código y observa la salida en el terminal y en la placa.

Ejemplos de Proyectos Hay varios tutoriales en línea que te pueden ayudar a comenzar con proyectos específicos, como hacer parpadear un LED, controlar un LED con un botón, entre otros3.

Espero que esta información te sea útil para comenzar a trabajar con Arduino y MicroPython. Si tienes más preguntas o necesitas ayuda con algo más, no dudes en preguntar

Ejemplos:

Estación Meteorológica: Puedes crear una estación meteorológica que mida la temperatura, la humedad y la presión atmosférica utilizando sensores como el BME280 y enviar los datos a una base de datos en la nube.

Control de Luz: Puedes controlar la intensidad de una luz utilizando un dimmer y controlarlo a través de una interfaz web.

Termostato IoT: Puedes crear un termostato inteligente que controle la temperatura de tu hogar y que puedas controlar a distancia a través de internet.

Estación de Riego12: Puedes automatizar el riego de tus plantas en función de la humedad del suelo y la temperatura ambiente.

Control de un Motor DC con el Driver L298N1: Puedes controlar la velocidad y la dirección de un motor DC utilizando el driver L298N.

Sensor de Movimiento con Radar de Microondas RCWL-05161: Puedes detectar movimiento en una habitación utilizando el sensor de radar de microondas RCWL-0516.

Envío de Lecturas de Sensores por Correo Electrónico: Puedes enviar las lecturas de un sensor (como el BME280) a tu correo electrónico utilizando el ESP32/ESP8266.

Robot Seguidor de Línea2: Puedes construir un robot que siga una línea en el suelo utilizando sensores de infrarrojos.

Brazo Robótico Controlado por Bluetooth2: Puedes construir un brazo robótico y controlarlo a través de una aplicación móvil utilizando Bluetooth.

Medidor de Calidad del Aire Conectado2: Puedes construir un dispositivo que mida la calidad del aire y envíe los datos a una base de datos en la nube.

Codigos fuentes de los controladores desarrollados:

Los códigos de fuentes de controladores desarrollados se refieren al conjunto de instrucciones escritas en un lenguaje de programación específico que se utilizan para crear y controlar los controladores de un dispositivo. Estos códigos son comprensibles para los humanos, lo que permite entender el funcionamiento del controlador y, cuando sea necesario, modificarlo.

El código fuente de un controlador es un elemento fundamental en el desarrollo de software. Contiene instrucciones y lógica para llevar a cabo tareas específicas. Por ejemplo, si estás desarrollando un controlador para un dispositivo específico, como

una tarjeta de red o una impresora, el código fuente de ese controlador contendrá las instrucciones específicas que el dispositivo necesita para funcionar correctamente.

Además, el código fuente de un controlador puede incluir detalles sobre lo que hace un controlador y cómo funciona internamente. Estos detalles permiten a los usuarios realizar pruebas exhaustivas antes del lanzamiento final, garantizando que el controlador sea fiable y seguro.

Es importante mencionar que antes de liberar un controlador basado en un código de ejemplo, se deben realizar cambios importantes en los controladores de ejemplo. Por ejemplo, si el controlador de ejemplo contiene identificadores únicos globales (GUID), nombres simbólicos de vínculos, nombre del objeto de dispositivo, etiquetas de grupo, definiciones de código de control de E/S (IOCTL), entre otros, estos deben ser cambiados para que se apliquen al controlador y al dispositivo

Demostración del funcionamiento de los controladores en wokwi y el funcionamiento de los controladores con su respectivo proyectos en vsCode

