Descripción de la Obra

Software: SENSORA\_SIMPLE\_ANGLE

El software SENSORA\_SIMPLE\_ANGLE es un módulo educativo que forma parte del ecosistema SensoraCore, diseñado para la monitorización de sensores angulares mediante potenciómetros conectados al ESP32 DevKit V1. A través de esta plataforma, el ESP32 ejecuta firmware en MicroPython que lee valores analógicos del potenciómetro conectado a un pin GPIO y los transmite mediante socket TCP a través de WiFi. La interacción con el sistema se realiza mediante una aplicación de escritorio desarrollada en Python con PySide6 que actúa como cliente TCP para recibir y procesar los datos.

• DISEÑO

El sistema se basa en una arquitectura cliente-servidor donde el ESP32 actúa como servidor TCP y la aplicación de escritorio como cliente. El ESP32 DevKit V1 ejecuta un firmware desarrollado en MicroPython que lee continuamente el pin analógico GPIO 32 donde se conecta un potenciómetro de entre 1kΩ y 100kΩ. El microcontrolador convierte las lecturas ADC (0-4095) a valores angulares usando una fórmula lineal y transmite estos datos a través de sockets TCP por WiFi. La aplicación cliente, desarrollada en Python con PySide6, se conecta al ESP32, recibe los datos en tiempo real, los procesa mediante algoritmos de calibración por regresión lineal y los visualiza en gráficas interactivas.

• IMPLEMENTACIÓN

El firmware del ESP32 está programado en MicroPython utilizando las librerías nativas machine, network y socket. El microcontrolador configura el pin GPIO como ADC con resolución de 12 bits y atenuación de 11dB para leer el rango completo de 0-3.3V del potenciómetro. El firmware establece un servidor TCP en el puerto 8080 y acepta conexiones entrantes de la aplicación cliente. La aplicación de escritorio está desarrollada en Python 3.x con PySide6 para la interfaz gráfica, matplotlib para visualización de datos, scikit-learn para regresión lineal y openpyxl para exportación a Excel.

Las funciones que contiene este software son:

• Lectura Analógica Continua: el ESP32 lee el potenciómetro conectado al GPIO con resolución de 12 bits (0-4095).

• Conversión Angular: convierte valores ADC a ángulos usando la fórmula linear angle = (adc\_value / 4095.0) \* 270.0 - 135.0.

• Servidor TCP: establece comunicación por socket TCP en puerto 8080 para transmitir datos a la aplicación cliente.

• Monitoreo en Tiempo Real: la aplicación cliente recibe datos continuamente y actualiza gráficas en tiempo real.

• Calibración por Regresión Lineal: permite al usuario establecer puntos de referencia para mejorar la precisión del sensor.

• Visualización Gráfica: muestra gráficos de líneas con datos históricos usando matplotlib integrado en PySide6.

• Exportación de Datos: genera archivos Excel (.xlsx) con datos tabulados, gráficos y metadatos de calibración.

• Filtrado Digital: aplica promedio móvil configurable para reducir ruido en las mediciones.

• Control de LED: permite encender/apagar el LED integrado del ESP32 como indicador visual desde la aplicación.

• VALIDACIÓN

La validación del software se llevó a cabo mediante pruebas de Funcionamiento, en las que se confirmó que el sistema integrado ESP32-Cliente es capaz de:

• Establecer conexión WiFi estable del ESP32 a la red local configurada en wifi\_config.py.

• Crear servidor TCP en puerto 8080 y aceptar conexiones entrantes de múltiples clientes.

• Leer valores analógicos del potenciómetro con resolución ADC de 12 bits (0-4095) de forma continua.

• Convertir lecturas ADC a ángulos usando fórmula lineal con rango de -135° a +135° (270° total).

• Transmitir datos en tiempo real mediante sockets TCP sin pérdida significativa de paquetes.

• Procesar algoritmos de regresión lineal en la aplicación cliente con coeficientes de determinación R² superiores a 0.95.

Asimismo, se verificó que la comunicación TCP entre el ESP32 y la aplicación cliente de escritorio mantiene estabilidad durante sesiones prolongadas de monitoreo, que la calibración por regresión lineal mejora la precisión del sensor reduciendo errores de linealidad a menos del 1%, y que la exportación de datos genera archivos Excel válidos con gráficos integrados para análisis posterior.

• PLATAFORMA

Hardware:

• ESP32 DevKit V1: Microcontrolador de 32 bits con procesador Xtensa LX6 dual-core, equipado con WiFi 802.11b/g/n y Bluetooth integrados. Incluye múltiples pines GPIO con conversores ADC de 12 bits para lectura de sensores analógicos. El pin GPIO se utiliza específicamente para la lectura del potenciómetro.

• Potenciómetro Rotativo: Sensor de posición angular que varía su resistencia proporcionalmente al ángulo de rotación. Ofrece un rango mecánico típico de 270° con linealidad mejor que ±1%. Se conecta entre 3.3V, GND y la señal al GPIO 32.

• LED Integrado: LED incorporado en el ESP32 conectado al GPIO 2, utilizado como indicador visual de estado que puede controlarse remotamente desde la aplicación cliente.

Software:

Para el firmware del ESP32 se emplea MicroPython con las librerías nativas machine para control de hardware, network para conectividad WiFi, socket para comunicación TCP y time para temporizadores. El desarrollo y la carga del firmware se realizan utilizando entornos como Visual Studio Code (con extensiones para MicroPython) y Thonny, que facilitan la edición, depuración y transferencia de scripts al ESP32. La aplicación cliente se desarrolla en Python 3.x utilizando PySide6 para la interfaz gráfica moderna, matplotlib integrado con FigureCanvasQTAgg para visualización científica de datos, scikit-learn para algoritmos de regresión lineal, openpyxl para exportación de reportes Excel y numpy para procesamiento numérico eficiente.

• AMBIENTE

El sistema opera como una plataforma educativa distribuida para el aprendizaje de instrumentación y sistemas embebidos IoT. El ESP32 DevKit V1 ejecuta firmware MicroPython que establece un servidor TCP en puerto 8080, lee continuamente el potenciómetro conectado al GPIO 32 con resolución ADC de 12 bits, convierte los valores (0-4095) a ángulos usando transformación lineal (-135° a +135°), y transmite los datos por WiFi a clientes conectados. La aplicación de escritorio desarrollada en Python actúa como cliente TCP, conectándose al ESP32 mediante su dirección IP local, recibiendo datos de forma asíncrona usando QThread para evitar bloqueos de interfaz, procesando los datos mediante algoritmos de calibración por regresión lineal implementados con scikit-learn, y presentando visualizaciones interactivas en tiempo real con matplotlib. El sistema incluye funcionalidades avanzadas como calibración multipoint para mejorar precisión, exportación automática de datos a Excel con gráficos integrados, control remoto del LED indicador, y filtrado digital configurable, proporcionando una experiencia educativa completa que abarca desde conceptos básicos de sensores analógicos hasta técnicas avanzadas de procesamiento de señales y comunicaciones de red en sistemas embebidos.