

Universidad Autónoma del Caribe

|  |
| --- |
| MANUAL DE USUARIO |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Yamith J. Romero Aldana  Elian D. Ruidiaz Marino | Saul A. Pérez Pérez  Carlos G. Díaz Sáenz  Pablo D. Bonaveri |  |

TABLA DE CONTENIDO

[INTRODUCCIÓN 4](#_Toc202139145)

[Descripción General 4](#_Toc202139146)

[Objetivo General 4](#_Toc202139147)

[Desarrollar una herramienta software educativa que facilite el aprendizaje de sistemas de adquisición de datos de sensores angulares, proporcionando una experiencia práctica e interactiva en el manejo de hardware y software embebido. 4](#_Toc202139148)

[Objetivos Específicos 4](#_Toc202139149)

[CARACTERISTICAS TECNICAS 5](#_Toc202139150)

[Arquitectura del Software 5](#_Toc202139151)

[Componentes principales 5](#_Toc202139152)

[Interfaz de Usuario (anguloSimple\_UI) 5](#_Toc202139153)

[Sistema de Calibración (CalibrationDialog) 5](#_Toc202139154)

[Hilo de Comunicación (AnguloSimpleThread) 5](#_Toc202139155)

[Funcionalidades Especificas 6](#_Toc202139156)

[Monitoreo en Tiempo Real 6](#_Toc202139157)

[Sistema de Calibración Avanzado 6](#_Toc202139158)

[Visualización Gráfica 6](#_Toc202139159)

[Exportación de Datos 6](#_Toc202139160)

[HARDWARE COMPATIBLE 7](#_Toc202139161)

[Microcontrolador Principal 7](#_Toc202139162)

[Sensor Requerido 7](#_Toc202139163)

[Diagrama de Conexiones 7](#_Toc202139164)

[REQUERIMIENTOS DE INSTALACIÓN Y CONFIGURACIÓN 8](#_Toc202139165)

[Requisitos del Sistema 8](#_Toc202139166)

[Dependencias de Software 8](#_Toc202139167)

[Proceso de Instalación 8](#_Toc202139168)

[Manual De Usuario 9](#_Toc202139169)

[Inicio del Sistema 9](#_Toc202139170)

[Calibración del Sensor 9](#_Toc202139171)

[Monitoreo en Tiempo Real 9](#_Toc202139172)

[Exportación de Datos 9](#_Toc202139173)

[ALGORITMOS IMPLEMENTADOS 10](#_Toc202139174)

[Calibración por Regresión Lineal 10](#_Toc202139175)

[Conversión ADC a Ángulo 10](#_Toc202139176)

[Filtrado de Señal 10](#_Toc202139177)

[VALIDACIÓN Y TESTING 11](#_Toc202139178)

[Pruebas de Funcionalidad 11](#_Toc202139179)

[Pruebas de Precisión 11](#_Toc202139180)

[Pruebas de Usabilidad 11](#_Toc202139181)

[APLICACIONES EDUCATIVAS 12](#_Toc202139182)

[Niveles Académicos 12](#_Toc202139183)

[Conceptos Didácticos Cubiertos 12](#_Toc202139184)

[MANTENIMIENTO Y SOPORTE 13](#_Toc202139185)

[Actualizaciones de Software 13](#_Toc202139186)

[Soporte Técnico 13](#_Toc202139187)

[Extensibilidad 13](#_Toc202139188)

[Consideraciones De Seguridad 13](#_Toc202139189)

[Seguridad Eléctrica 13](#_Toc202139190)

[CONCLUSIONES 14](#_Toc202139191)

[REFERENCIAS Y DOCUMENTACIÓN ADICIONAL 15](#_Toc202139192)

[Estándares Aplicados 15](#_Toc202139193)

# INTRODUCCIÓN

## Descripción General

SENSORA\_SIMPLE\_ANGLE es un módulo software especializado diseñado para la monitorización en tiempo real de sensores de ángulo mediante potenciómetros conectados a microcontroladores ESP32. Este software forma parte del ecosistema SensoraCore, una plataforma integral de módulos didácticos para el aprendizaje de sistemas embebidos y adquisición de datos.

El software proporciona una interfaz gráfica intuitiva que permite a estudiantes y profesionales visualizar, calibrar y analizar lecturas angulares provenientes de sensores analógicos, facilitando el entendimiento práctico de conceptos como conversión analógico-digital, calibración por regresión lineal y comunicación serie con microcontroladores.

## Objetivo General

## Desarrollar una herramienta software educativa que facilite el aprendizaje de sistemas de adquisición de datos de sensores angulares, proporcionando una experiencia práctica e interactiva en el manejo de hardware y software embebido.

## Objetivos Específicos

* **Visualización en Tiempo Real:** Mostrar gráficamente las lecturas del sensor de ángulo con actualización continua
* **Calibración Inteligente:** Implementar algoritmos de regresión lineal para calibración precisa del sensor
* **Interfaz Educativa:** Proporcionar diagramas de conexión y documentación integrada
* **Exportación de Datos:** Generar reportes en formato Excel para análisis posterior
* **Comunicación Serie:** Establecer comunicación robusta con microcontroladores ESP32

# CARACTERISTICAS TECNICAS

## Arquitectura del Software

Lenguaje de Programación: Python 3.8+

Framework GUI: PyQt5 con elementos personalizados

Visualización: Matplotlib integrado con FigureCanvasQTAgg

Comunicación: Protocolo serie TCP/UDP para ESP32

Procesamiento de Datos: NumPy y SciPy para cálculos matemáticos

Exportación: OpenPyXL para generación de archivos Excel

## Componentes principales

Interfaz de Usuario (anguloSimple\_UI)

def anguloSimple\_UI(self):

"""

Interfaz principal del módulo con:

- Diagrama de conexiones ESP32

- Controles de monitoreo

- Visualización en tiempo real

- Panel de calibración

"""

Sistema de Calibración (CalibrationDialog)

class AnguloSimpleThread(QThread):

"""

Manejo asíncrono de comunicación con ESP32:

- Recepción de datos serie en tiempo real

- Conversión ADC a valores angulares

- Señales Qt para actualización de interfaz

"""

class CalibrationDialog(QDialog):

"""

Diálogo avanzado para calibración por regresión lineal:

- Entrada manual de puntos de referencia

- Visualización gráfica de la regresión

- Cálculo automático de ecuaciones de calibración

- Estadísticas de calidad de ajuste (R²)

"""

def anguloSimple\_UI(self):

"""

Interfaz principal del módulo con:

- Diagrama de conexiones ESP32

- Controles de monitoreo

- Visualización en tiempo real

- Panel de calibración

"""

Hilo de Comunicación (AnguloSimpleThread)

## Funcionalidades Especificas

Monitoreo en Tiempo Real

* **Frecuencia de Muestreo:** Configurable hasta 100 Hz
* **Rango de Medición:** -135° a +135° (configurable)
* **Resolución ADC:** 12 bits (0-4095) del ESP32
* **Filtrado de Señal:** Promedio móvil para reducir ruido

Sistema de Calibración Avanzado

* **Método:** Regresión lineal por mínimos cuadrados
* **Puntos Mínimos:** 2 puntos de referencia
* **Puntos Recomendados:** 5-10 para mejor precisión
* **Estadísticas:** Coeficiente de determinación R²
* **Persistencia:** Guardado/carga de calibraciones en JSON

Visualización Gráfica

* **Tipo de Gráfico:** Línea continua con marcadores
* **Actualización:** Tiempo real con buffer circular
* **Personalización:** Colores y estilos Bootstrap
* **Zoom y Pan:** Navegación interactiva del gráfico

Exportación de Datos

* **Formato:** Excel (.xlsx) con múltiples hojas
* **Contenido:** Timestamps, lecturas ADC, ángulos calibrados
* **Gráficos:** Gráfico de líneas integrado en Excel
* **Metadatos:** Información de calibración y sesión

# HARDWARE COMPATIBLE

## Microcontrolador Principal

* **Modelo:** ESP32 DevKit V1 o compatible
* **GPIO Utilizado:** Pin 32 (ADC1\_CH4)
* **Alimentación:** 3.3V y GND
* **Comunicación:** WiFi 802.11b/g/n

## Sensor Requerido

* **Tipo:** Potenciómetro rotativo
* **Resistencia:** 10kΩ (recomendado)
* **Rango Angular:** -135° 135°
* **Linealidad:** ±1% o mejor
* **Conexión:** 3 pines (VCC, GND, Señal)

## Diagrama de Conexiones

┌─────────────────────────────────┐

│ ESP32 DevKit V1 │

│ │

│ 3V3 ○ ←── Potenciómetro (+) │

│ GPIO ○ ←── Potenciómetro (S) │

│ GND ○ ←── Potenciómetro (-) │

│ │

└─────────────────────────────────┘

# REQUERIMIENTOS DE INSTALACIÓN Y CONFIGURACIÓN

## Requisitos del Sistema

* **Sistema Operativo:** Windows 10/11, macOS 10.14+, Ubuntu 18.04+
* **Python:** Versión 3.8 o superior
* **RAM:** Mínimo 2GB, recomendado 4GB
* **Espacio en Disco:** 500MB para Implementación de modulo

## Dependencias de Software

*# requirements.txt*

PyQt5>=5.15.0

matplotlib>=3.5.0

numpy>=1.21.0

scipy>=1.7.0

openpyxl>=3.0.9

pyserial>=3.5

## Proceso de Instalación

1. **Clonar Repositorio:** Descargar código fuente
2. **Instalar Dependencias:** pip install -r requirements.txt
3. **Configurar ESP32:** Flashear firmware compatible
4. **Ejecutar Software:** python main.py

# Manual De Usuario

## Inicio del Sistema

1. Conectar ESP32 al puerto USB
2. Verificar conexiones del potenciómetro
3. Ejecutar aplicación SensoraCore
4. Seleccionar módulo "Sensor de Ángulo Simple"

## Calibración del Sensor

1. Hacer clic en "⚙️ Calibrar Sensor"
2. Agregar mínimo 2 puntos de referencia:
   * Posicionar potenciómetro en ángulo conocido
   * Anotar lectura ADC mostrada
   * Ingresar ángulo real de referencia
3. Repetir para diferentes ángulos
4. Hacer clic en "🔧 Realizar Calibración"
5. Verificar calidad de ajuste (R² > 0.95 recomendado)

## Monitoreo en Tiempo Real

1. Hacer clic en "▶️ Iniciar Monitoreo"
2. Observar lecturas en tiempo real
3. Analizar gráfica de tendencias
4. Usar "🗑️ Limpiar Gráfica" para reiniciar

## Exportación de Datos

1. Acumular datos durante sesión de monitoreo
2. Hacer clic en "📊 Exportar Excel"
3. Seleccionar ubicación de guardado
4. Abrir archivo generado para análisis

# ALGORITMOS IMPLEMENTADOS

## Calibración por Regresión Lineal

def perform\_calibration(self):

"""

Implementa regresión lineal: y = mx + b

donde:

- x: valores ADC del sensor

- y: ángulos de referencia conocidos

- m: pendiente (sensibilidad)

- b: intersección (offset)

"""

slope, intercept, r\_value, p\_value, std\_err = linregress(

self.raw\_values, self.reference\_values

)

self.r\_squared = r\_value \*\* 2

## Conversión ADC a Ángulo

def apply\_calibration(self, raw\_value):

"""

Aplica calibración lineal a lectura ADC:

angle = slope \* adc\_value + intercept

"""

if self.is\_calibrated:

return self.slope \* raw\_value + self.intercept

else:

*# Conversión por defecto (sin calibrar)*

return (raw\_value / 4095.0) \* 270.0 - 135.0

## Filtrado de Señal

def apply\_moving\_average(self, new\_value, window\_size=5):

"""

Implementa filtro de promedio móvil:

- Reduce ruido de alta frecuencia

- Mantiene respuesta temporal aceptable

- Configurable según aplicación

"""

self.buffer.append(new\_value)

if len(self.buffer) > window\_size:

self.buffer.pop(0)

return sum(self.buffer) / len(self.buffer)

# VALIDACIÓN Y TESTING

## Pruebas de Funcionalidad

* **Comunicación Serie:** Verificación de protocolo con ESP32
* **Calibración:** Validación con sensores conocidos
* **Visualización:** Pruebas de rendimiento gráfico
* **Exportación:** Integridad de archivos Excel

## Pruebas de Precisión

* **Linealidad:** Error < 1% en rango completo
* **Repetibilidad:** Desviación estándar < 0.5°
* **Temperatura:** Estabilidad en rango 0-40°C
* **Ruido:** Relación señal/ruido > 40dB

## Pruebas de Usabilidad

* **Tiempo de Aprendizaje:** < 30 minutos para usuarios novatos
* **Facilidad de Calibración:** Proceso guiado paso a paso
* **Documentación:** Diagramas integrados en interfaz
* **Robustez:** Recuperación automática de errores

# APLICACIONES EDUCATIVAS

## Niveles Académicos

* **Educación Media:** Conceptos básicos de sensores
* **Técnico Superior:** Sistemas de adquisición de datos
* **Universitario:** Instrumentación y control
* **Posgrado:** Procesamiento avanzado de señales

## Conceptos Didácticos Cubiertos

* **Electrónica Analógica:** Potenciómetros y divisores de voltaje
* **Conversión A/D:** Cuantización y resolución
* **Calibración:** Regresión lineal y estadística
* **Programación:** Interfaces gráficas y threads
* **Comunicaciones:** Protocolos serie y WiFi

# MANTENIMIENTO Y SOPORTE

## Actualizaciones de Software

* **Frecuencia:** Trimestral o según necesidades
* **Canales:** GitHub y repositorio institucional
* **Documentación:** Changelog detallado

## Soporte Técnico

* **Documentación:** Manual completo en línea
* **Issues:** Sistema de tickets en GitHub
* **Capacitación:** Talleres presenciales/virtuales

## Extensibilidad

* **Plugins:** Sistema modular expandible

## Consideraciones De Seguridad

### Seguridad Eléctrica

* **Voltajes Seguros:** Máximo 5V en todas las conexiones

# CONCLUSIONES

SENSORA\_SIMPLE\_ANGLE representa una herramienta educativa robusta y versátil para el aprendizaje de sistemas de sensores angulares. Su diseño modular, interfaz intuitiva y capacidades de calibración avanzadas lo convierten en una solución ideal para instituciones educativas que buscan modernizar sus laboratorios de instrumentación y control.

El software cumple exitosamente con los objetivos planteados, proporcionando una experiencia educativa completa que abarca desde conceptos básicos de electrónica hasta técnicas avanzadas de procesamiento de señales y calibración estadística.

# REFERENCIAS Y DOCUMENTACIÓN ADICIONAL

## Estándares Aplicados

* **IEEE 1451:** Transductores inteligentes
* **ISO 5725:** Precisión y exactitud de medición
* **IEC 61131:** Sistemas de control programable