

Programación básica para lectura de sensores con Arduino y Raspberry Pi

Tecnología e Informática | Pensamiento Computacional

Descripción del Curso

Dirigido a estudiantes de 15 a 16 años, este curso pertenece a Pensamiento Computacional y se centra en la Unidad 3: Integración de la lectura de sensores con salidas visuales o registro. En esta unidad se trabajará la lectura de sensores (reales o simulados) y su integración con salidas que faciliten la interpretación: impresión en consola para monitoreo básico, registro en archivos CSV para análisis posterior y una interfaz gráfica simple que permita visualizar lecturas en tiempo real. El objetivo es desarrollar la capacidad de transformar señales del mundo físico en información usable, comprender flujos de datos y comunicar hallazgos de manera clara. Se combinarán actividades prácticas de programación, manejo de datos y diseño de interfaces para fomentar un aprendizaje activo y colaborativo.

La unidad busca que el estudiante diseñe, implemente y evalúe soluciones que permitan observar, registrar y presentar datos de sensores de forma legible y usable. Se enfatizará la importancia de la legibilidad de las visualizaciones, la calidad de los registros y la facilidad de interpretación de la información, conectando la teoría con situaciones reales como monitoreo ambiental, robótica educativa y proyectos de makers. Al finalizar, los estudiantes deberían ser capaces de integrar una lectura de sensores con salidas en consola, generar archivos CSV con datos estructurados y crear una interfaz gráfica básica que muestre lecturas en tiempo real y permita un análisis posterior. Se fomentará el pensamiento computacional mediante la construcción de flujos de datos, la selección de salidas adecuadas para distintos fines (monitoreo, registro y visualización) y la comunicación de resultados a distintos públicos. El aprendizaje será progresivo, partiendo de conceptos simples de lectura y salida de datos, hasta la implementación de una GUI y la generación de reportes que faciliten la toma de decisiones basada en datos.

Competencias

- Pensamiento lógico y algorítmico para interpretar datos de sensores y transformarlos en salidas comprensibles.
- Diseño y desarrollo de soluciones de visualización y almacenamiento de datos (con salida en consola, archivos CSV y una interfaz gráfica básica).
- Programación en Python y manejo de bibliotecas para lectura de sensores, generación de CSV y visualización básica (por ejemplo, matplotlib y Tkinter).
- Habilidad para analizar, interpretar y comunicar resultados de manera clara, considerando la usabilidad y legibilidad de las visualizaciones.
- Colaboración, planificación de proyectos y ética en manejo de datos y seguridad de la información.

Requerimientos

- Acceso a un ordenador con conexión a Internet y sistema operativo actualizado.
- Python 3.x y un entorno de desarrollo (Thonny, PyCharm, VS Code u otro similar).
- Bibliotecas necesarias: pandas, matplotlib, numpy y tkinter (u otra biblioteca de GUI simple).
- Capacidad para instalar software y, si es posible, acceso a hardware de sensores o, alternativamente, utilizar simuladores de lectura de sensores.
- Proyecto práctico que combine lectura de sensores (real o simulada) con salidas en consola, registro en CSV y una interfaz gráfica; entrega del código y de un informe en un repositorio o carpeta compartida.
- Conocimientos previos de programación básica y manejo de datos en archivos CSV.

Unidades del Curso

Unidad 1: Unidad 1: Introducción a sensores y plataformas (Arduino y Raspberry Pi)

Objetivos de Aprendizaje

- Identificar y describir los componentes clave de un sensor (tipo de salida, rango, modo de conexión) y su interfaz con Arduino y Raspberry Pi, diferenciando entre entradas analógicas y digitales.
- Explicar cómo se conectan sensores básicos a Arduino y a Raspberry Pi, considerando pines, voltaje de alimentación y requerimientos de la placa.
- Distinguir entre sensores analógicos y digitales y justificar la necesidad de convertir señales analógicas para la lectura en Raspberry Pi.

Contenidos Temáticos

1. Tema 1: Componentes de sensores y sus salidas

Descripción de las salidas analógica y digital, rangos de operación y métodos de conexión típicos.

2. Tema 2: Interfaces con Arduino

Cómo leer entradas digitales y analógicas en Arduino (pines, voltaje, funciones analogRead y digitalRead).

3. Tema 3: Interfaces con Raspberry Pi

Limitaciones de la Pi para lectura analógica y opciones para lectura analógica mediante ADC externo (SPI/I2C, MCP3008, etc.).

4. Tema 4: Buenas prácticas y seguridad

Consejos de cableado, alimentación segura y reducción de ruido en electrónica básica.

Actividades

- **Actividad 1: Exploración de sensores y plataformas** - Tema: Reconocer físicamente sensores comunes y sus salidas. - Descripción: Los estudiantes identifican componentes como potenciómetros, interruptores y sensores de

temperatura, y describen su salida, rango y forma de conexión. - Puntos clave: diferencias entre salidas analógicas y digitales; interpretación del rango de voltaje; conceptos de alimentación y señal. - Aprendizajes: Capacidad de clasificar sensores y prever su conexión en Arduino y Raspberry Pi.

- **Actividad 2: Lectura de sensor digital en Arduino** - Tema: Lectura de un sensor digital (por ejemplo, un pulsador) con Arduino. - Descripción: Se implementa un sketch que lee digitalRead y genera una salida en consola. - Puntos clave: configuración de pines, debouncing, interpretación de valores HIGH/LOW. - Aprendizajes: Comprender la lectura digital y su representación en Arduino.
- **Actividad 3: Lectura de sensor analógico en Arduino y preparación para Raspberry Pi** - Tema: Lectura analógica con Arduino y preparación de datos para transferencia a Pi. - Descripción: Se lee con analogRead, se convierte a voltaje y se documenta el mapeo a una unidad útil. - Puntos clave: conversión raw a voltaje ($R = \text{valor}/1023$), calibración básica. - Aprendizajes: Manejo de señales analógicas y pasos para transportar datos a Raspberry Pi.
- **Actividad 4: Introducción a la lectura analógica en Raspberry Pi** - Tema: Uso de un ADC externo (p. ej., MCP3008) para leer un sensor analógico desde Raspberry Pi. - Descripción: Se configura SPI, se implementa lectura de un canal y se imprime el valor crudo y convertido. - Puntos clave: configuración de bus SPI, manejo de datos binarios y conversión a voltaje. - Aprendizajes: Comprender la necesidad de un ADC con Raspberry Pi y cómo leer un sensor analógico.

Evaluación

- Evaluación de Objetivo Específico 1: Participación en la identificación y descripción de sensores y sus salidas durante las actividades prácticas.
- Evaluación de Objetivo Específico 2: Desarrollo de sketches en Arduino que lean sensores analógicos y/o digitales, y segmentos de código en Raspberry Pi que lean sensores a través de un ADC externo.
- Evaluación de Objetivo Específico 3: Informe corto de calibración y mapeo de lecturas a unidades útiles (voltaje, temperatura, etc.).
- Evaluación global: Proyecto corto que demuestre la conexión de un sensor, la lectura en Arduino y la obtención de una lectura convertida, más una breve comparación entre plataformas.

Unidad 2: Programación de lecturas de sensores en Arduino (C/C++) y Raspberry Pi (Python)

Objetivos de Aprendizaje

- Escribir código en Arduino para leer sensores analógicos y/o digitales y convertir la lectura cruda a unidades útiles (p. ej., voltaje, temperatura).
- Escribir código en Raspberry Pi (Python) para leer sensores, usando un ADC externo para lectura analógica y convertir a unidades útiles.

- Calibrar y mapear lecturas para mejorar la precisión de las mediciones y comparar enfoques entre Arduino y Raspberry Pi.

Contenidos Temáticos

1. Tema 1: Lectura de sensores en Arduino (C/C++)

Uso de analogRead y digitalRead, conversión de lecturas crudas a voltaje y unidades útiles, y validación de resultados.

2. Tema 2: Lectura de sensores en Raspberry Pi (Python) con ADC externo

Configuración de MCP3008 (SPI), lectura de canales analógicos y conversión a voltaje en Python.

3. Tema 3: Conversión y calibración de lecturas

Algoritmos de mapeo (p. ej., map() en Arduino y cálculos en Python) para estimar unidades físicas y calibración básica.

4. Tema 4: Validación y comparación entre plataformas

Comparación de lecturas entre Arduino y Raspberry Pi para el mismo sensor y análisis de discrepancias.

Actividades

- **Actividad 1: Lectura de sensor analógico en Arduino** - Tema: Implementar un sketch que lea un sensor analógico y convierta a voltaje. - Descripción: Paso a paso desde la lectura hasta la visualización en la consola y el registro de valor convertidos. - Puntos clave: configuración de referencia, conversión analogRead a voltaje, interpretación de resultados. - Aprendizajes: dominio de la lectura analógica y conversión simple en Arduino.
- **Actividad 2: Lectura de sensor en Raspberry Pi con ADC externo** - Tema: Instalar y usar MCP3008 para leer un sensor analógico desde Raspberry Pi con Python. - Descripción: Configuración de SPI, lectura de canal, conversión a voltaje y mostración en consola. - Puntos clave: interacción SPI, manejo de bytes, conversión de datos. - Aprendizajes: lectura analógica en Raspberry Pi y conversión de señales.
- **Actividad 3: Calibración y mapeo de lecturas** - Tema: Calibrar sensores para obtener unidades útiles más precisas. - Descripción: Aplicar fórmulas de calibración y comparar resultados entre Arduino y Pi. - Puntos clave: fórmulas de escalamiento, offset, validación de resultados. - Aprendizajes: habilidad para calibrar lecturas y aumentar la precisión.
- **Actividad 4: Informe de comparación entre plataformas** - Tema: Documentar y analizar diferencias de lectura entre Arduino y Raspberry Pi para el mismo sensor. - Descripción: Presentar resultados, discutir causas y proponer mejoras. - Puntos clave: análisis de fuentes de error, conclusiones prácticas. - Aprendizajes: capacidad de análisis comparativo y comunicación de hallazgos.

Evaluación

- Evaluación del Objetivo Específico 1: Calidad y exactitud de la lectura y conversión en Arduino.

- Evaluación del Objetivo Específico 2: Implementación funcional de lectura en Raspberry Pi con ADC externo y conversión a unidades útiles.
- Evaluación del Objetivo Específico 3: Capacidad de calibrar y justificar mejoras en las lecturas.
- Evaluación del Objetivo Específico 4: Informe de comparación entre plataformas con análisis de resultados.

Unidad 3: Unidad 3: Integración de la lectura de sensores con salidas visuales o registro

Objetivos de Aprendizaje

- Diseñar salidas de datos: impresión en consola, registro en archivos (CSV) y visualización básica para facilitar la interpretación de lecturas.
- Desarrollar una interfaz gráfica o gráfica simple en Python para visualizar lecturas en tiempo real.
- Crear un flujo de datos que registre y presente lecturas de forma clara y usable, con consideraciones de usabilidad y legibilidad.

Contenidos Temáticos

1. Tema 1: Salida en consola y registro en archivos

Imprimir lecturas en consola y grabarlas en archivos CSV para análisis posterior.

2. Tema 2: Visualización básica de datos

Creación de gráficos simples en Python (matplotlib) o una interfaz gráfica básica (Tkinter) para mostrar lecturas en tiempo real.

3. Tema 3: Flujo de datos y usabilidad

Diseño de un flujo de datos estable y fácil de interpretar, con consideraciones de legibilidad y organización de la información.

4. Tema 4: Integración entre Arduino y Raspberry Pi

Cómo coordinar lecturas desde ambas plataformas y consolidar datos para salida y registro.

Actividades

- **Actividad 1: Registro en consola y CSV** - Tema: Imprimir lecturas en consola y guardar un historial en CSV. - Descripción: Se implementa logging desde Arduino y/o Raspberry Pi para generar archivos con columnas de tiempo y valor leído. - Puntos clave: formato de CSV, manejo de tiempo, consistencia entre lecturas. - Aprendizajes: capacidad de registrar y exportar datos para análisis.
- **Actividad 2: Visualización en tiempo real con Python** - Tema: Crear un gráfico que actualice las lecturas en tiempo real. - Descripción: Utilizar matplotlib o Tkinter para mostrar lecturas en un gráfico sencillo. - Puntos clave: bucle de actualización, manejo de datos, claridad visual. - Aprendizajes: interpretación visual de los datos y feedback rápido.

- **Actividad 3: Interfaz gráfica básica** - Tema: Construir una interfaz simple para mostrar estados y lecturas. - Descripción: Crear widgets que muestren valores y tendencias, con botones de inicio/parada. - Puntos clave: usabilidad, lectura continua, actualizaciones en la interfaz. - Aprendizajes: habilidad para desarrollar interfaces simples que acompañen a la lectura de sensores.
- **Actividad 4: Integración de datos entre Arduino y Raspberry Pi** - Tema: Sincronizar lecturas procedentes de ambas plataformas y consolidarlas en un único registro. - Descripción: Diseñar un flujo de datos consistente y un informe de resumen. - Puntos clave: sincronización temporal, coherencia de unidades, registro centralizado. - Aprendizajes: capacidad de integrar múltiples fuentes de datos y presentarlas de forma unificada.

Evaluación

- Evaluación en OBJETIVO ESPECÍFICO 1: Implementación de salidas en consola y archivos CSV de las lecturas de sensores.
- Evaluación en OBJETIVO ESPECÍFICO 2: Calidad y funcionalidad de la visualización en tiempo real o interfaz gráfica (claridad, legibilidad, respuesta en tiempo real).
- Evaluación en OBJETIVO ESPECÍFICO 3: Cohesión y usabilidad del flujo de datos y del registro, así como la calidad del informe de análisis.