

# Sistemas Embebidos

## Trabajo Práctico N°1 – Año 2025

### Conversores AD y DA: Protocolos de Comunicación Serial Cableados

#### *Integrantes:*

- Arrieta, Nahuel
- Del Longo, Micaela
- Maglione, Andrés

#### Introducción

Este informe describe la implementación de un sistema de control de LED y monitoreo de un sensor LDR (Light Dependent Resistor) utilizando una plataforma basada en Arduino, una interfaz web y un servidor desarrollado con Flask. Se detalla el diseño de la arquitectura del sistema y el flujo de datos entre la interfaz web y el hardware.

#### Arquitectura del Sistema

El sistema consta de tres componentes principales:

- Frontend (`index.html` y `script.js`): Permite a los usuarios interactuar con el sistema mediante controles de deslizamiento (sliders) y botones.
- Backend (`arduino_web_server.py`): Procesa las solicitudes del usuario y envía comandos al Arduino a través de un puerto serie.
- Arduino (`led_ldr_controller.ino`): Recibe los comandos, ajusta el brillo de los LED y envía datos de sensores al servidor.

La comunicación entre el microcontrolador y la interfaz web se realiza a través de WebSockets y un puerto serie, permitiendo el control remoto de los dispositivos conectados al Arduino.

#### Código de Arduino

El Arduino está programado para recibir comandos desde el puerto serie, modificar el estado de los LED y enviar la información actualizada a la interfaz web. Los valores de los LED y el sensor LDR se envían en el siguiente formato:

`L1:valor;L2:valor;L3:valor;D13:valor;A3:valor;`

Donde cada "valor" representa el estado de cada dispositivo.

### Backend: Flask y SocketIO

El servidor Flask utiliza Flask-SocketIO para establecer una comunicación en tiempo real entre el Arduino y la interfaz web. Se encarga de:

- **serial\_reader()**: Leer los datos provenientes del Arduino y emitir actualizaciones a la interfaz web mediante WebSockets.
- **set\_control(data)**: Recibir comandos desde la interfaz y enviarlos al Arduino a través del puerto serie.

El backend utiliza hilos para la lectura continua de datos desde el puerto serie sin bloquear la ejecución del programa.

### Frontend: HTML, JavaScript, jQuery y Socket.IO

La interfaz permite a los usuarios controlar los LED mediante sliders y botones. Además, muestra en tiempo real el estado de los LED y la lectura del sensor LDR. Se utilizan eventos de WebSocket para actualizar la información y enviar comandos al servidor.

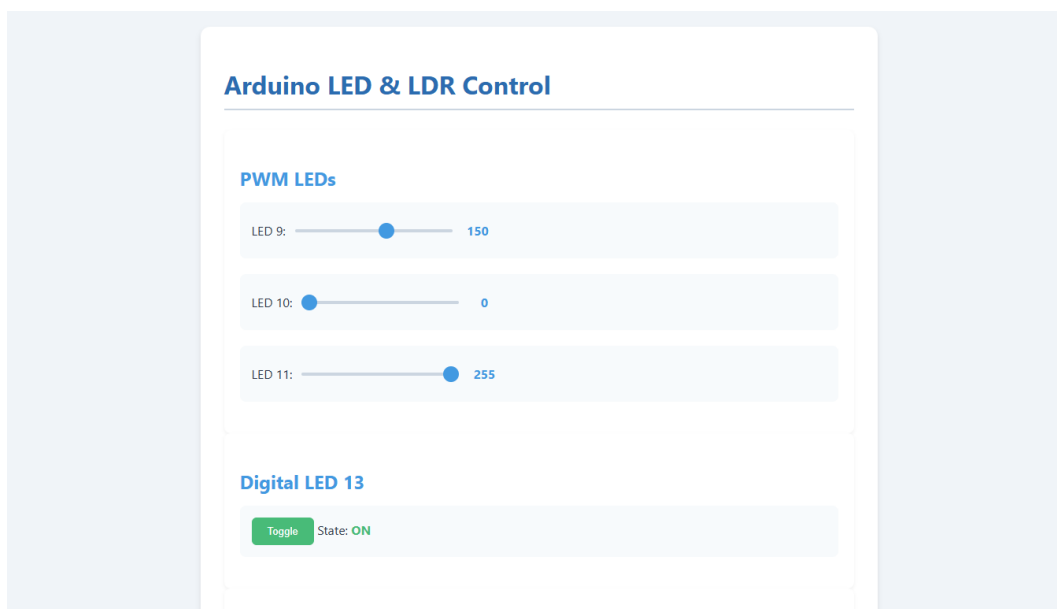


Figura 1: Captura de pantalla de la aplicación web en la parte para controlar los LED.

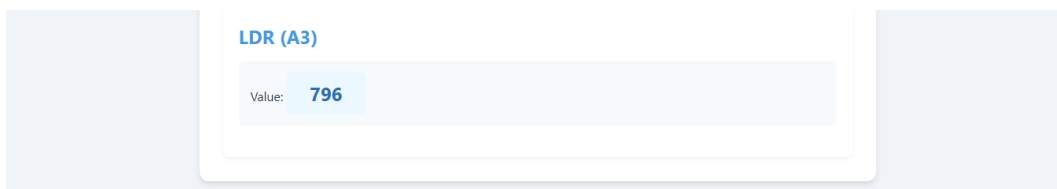


Figura 2: Captura de pantalla de la aplicación web en la parte con la información del sensor LDR.

## Flujo de Datos

El sistema sigue el siguiente flujo de datos:

1. Entrada de datos desde la interfaz web:
  - El usuario ajusta un deslizador para cambiar la intensidad de un LED (PWM) o presiona un botón para encender/apagar el LED digital (pin 13).
  - `script.js` captura estos eventos y envía los datos mediante WebSockets al servidor Flask.
2. Procesamiento en el servidor Flask:
  - El servidor Flask recibe los datos mediante el evento `control_led` de WebSocket.
  - Convierte los datos a un formato compatible y los envía a Arduino a través del puerto serie.
3. Recepción y ejecución en Arduino:
  - Arduino recibe los comandos a través del puerto serie.
  - Interpreta los datos y ajusta los pines de salida para modificar la intensidad de los LEDs o cambiar el estado del LED digital.
  - También lee el valor del sensor LDR (A3) y prepara un mensaje con el estado actual del sistema.
4. Envío de datos de Arduino al servidor:
  - Arduino envía el estado actual de los LEDs y la lectura del LDR a través del puerto serie en formato `clave:valor;`.
  - El servidor Flask recibe estos datos, los almacena en un diccionario y los transmite a la interfaz web mediante WebSockets.
5. Actualización en la interfaz web:
  - `script.js` recibe los datos de WebSocket y actualiza la interfaz de usuario con los nuevos valores.
  - Se actualizan los deslizadores de los LEDs, el estado del LED digital y la lectura del sensor LDR en la web en tiempo real.

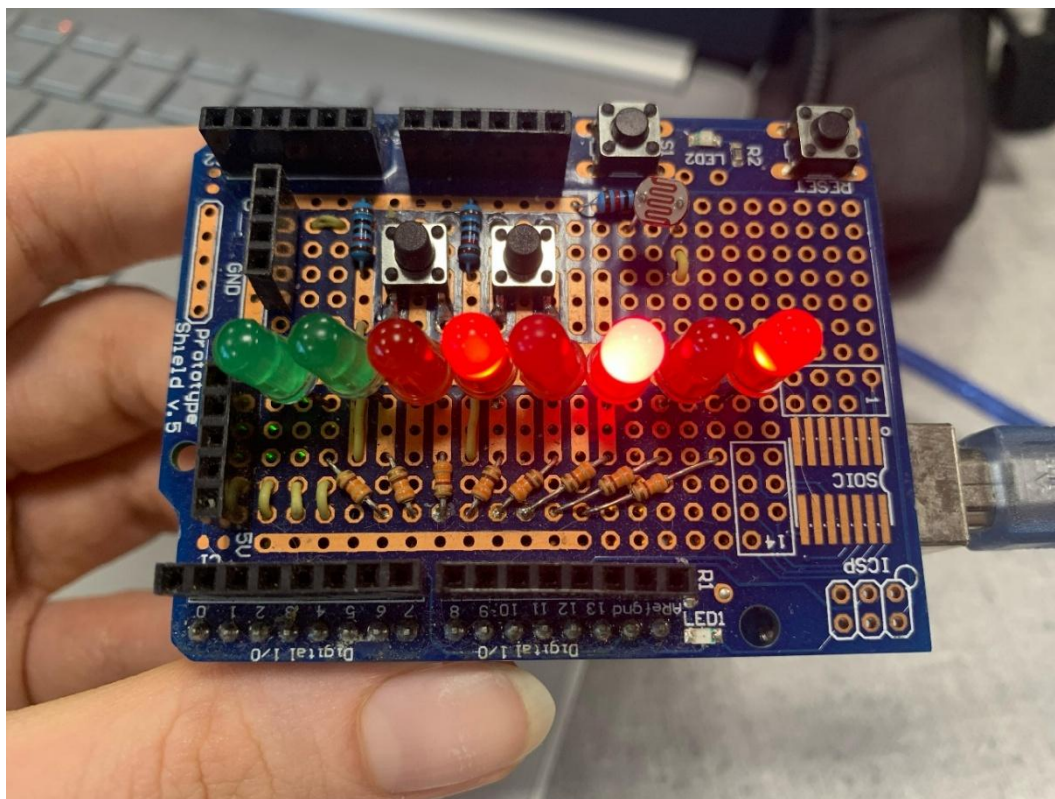


Figura 3: Foto del Arduino en funcionamiento con las configuraciones mostradas en la Figura 1.