

ProdLine Classifier

La Giglia, Rodrigo Ariel	33.334.248
Loiotile, Juan Cruz	42.101.782
Marrone, Micaela Abril	45.683.584
Rivas, Nahuel Alberto	44.364.975
Pensel, Nicolás	36.922.906

Martes, Grupo M5

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

Resumen.

El proyecto presenta el desarrollo de una cinta transportadora capaz de clasificar elementos de acuerdo con su color. El sistema integra sensores que identifican la tonalidad de cada objeto y un mecanismo de desvío que dirige los artículos hacia diferentes recipientes, agrupándolos automáticamente según su categoría cromática. Además, se desarrolló una aplicación Android que monitorea el estado del sistema en tiempo real y permite ejecutar acciones remotas, como la detención de emergencia y el reseteo. Esta incorporación complementa la solución, facilitando el control y el seguimiento del proceso desde cualquier dispositivo móvil. La propuesta resulta especialmente útil en líneas de producción o entornos industriales que requieren optimizar el proceso de separación, reduciendo el esfuerzo manual y mejorando la eficiencia en la organización del trabajo.

Palabras claves: cinta transportadora, clasificación por color, automatización industrial, sistema embebido, aplicación android.

1 Introducción

El proyecto consiste en una cinta transportadora diseñada para clasificar productos automáticamente según su color (particularmente, en la implementación llevada a cabo para este proyecto se identifica solo rojo y azul). El sistema cuenta con sensores ubicados al inicio y al final de la cinta. Cuando el sensor de entrada detecta un objeto, la cinta comienza a moverse y lo traslada hasta el clasificador cromático. Allí, según el color identificado, un servomotor posiciona el recipiente que se encuentra en la salida de la cinta. Finalmente, al activarse el sensor final, la cinta se detiene. Si el objeto no es reconocido o no alcanza el final, el sistema también se detiene de forma automática. Además, se incluye un botón de parada de emergencia como medida de seguridad. Este desarrollo resulta útil en procesos industriales donde se necesita ordenar materiales por color, reduciendo la intervención manual y mejorando la organización del trabajo.

Además, el desarrollo incluyó una aplicación Android que se comunica mediante MQTT con un broker en la nube, el cual a su vez se vincula con el sistema embebido. La app informa en tiempo real el estado del sistema, permite realizar una parada de emergencia y resetear la operación.

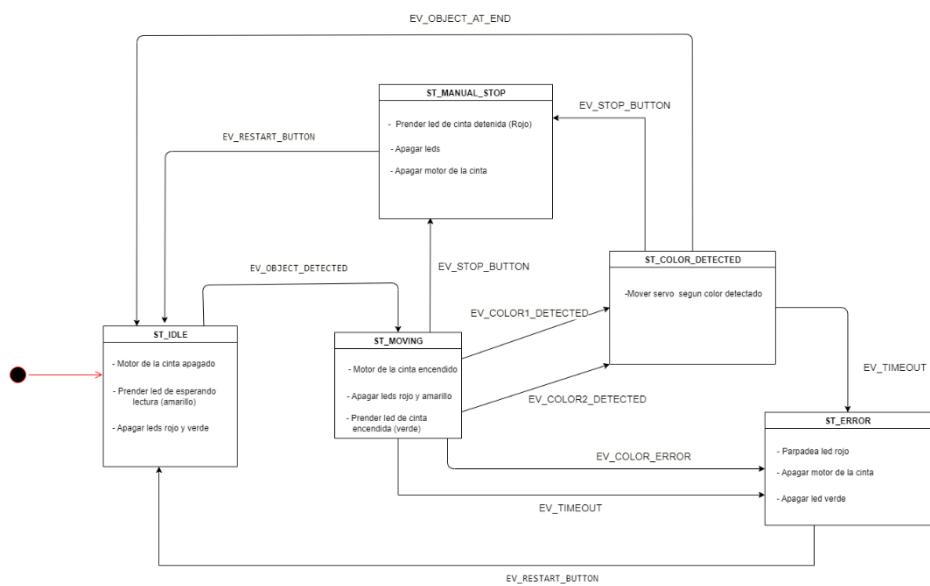
También cuenta con una sección de *Process Log*, donde se visualizan estadísticas sobre la cantidad de paradas y los objetos procesados, y una sección de *Settings* que permite modificar la velocidad de la cinta y consultar los últimos valores registrados por los sensores y el servomotor.

2 Desarrollo

Repositorio GitHub: <https://github.com/UNLAM-SOA/2025-SOA-Q2-M5>

URL proyecto Wowki: <https://wokwi.com/projects/442535339004692481>

Diagrama máquina de estados



Descripción:

El sistema funciona como una máquina de estados finitos que responde a eventos generados por sensores y botones.

Estados Principales

ST_IDLE (Espera):

- Motor de cinta apagado.
- LED amarillo encendido.
- LED verde apagado.
- LED rojo apagado.
- Servo en posición inicial.
- El sistema espera que un objeto sea colocado en la cinta.

ST_MOVING (Objeto en movimiento):

- Motor encendido.

- LED verde encendido.
- LED amarillo apagado.
- LED rojo apagado.
- El sistema cambia de estado al detectar el color, ante la acción de botón de parada o un timeout por color no detectado.

ST_COLOR_DETECTED (Clasificación):

Según el color detectado:

- Rojo → Servo gira 45° (derecha).
- Azul → Servo gira 135° (izquierda).
- El sistema espera a que el objeto llegue al sensor final, tambien puede cambiar de estado ante la accion del boton de parada o timeout.

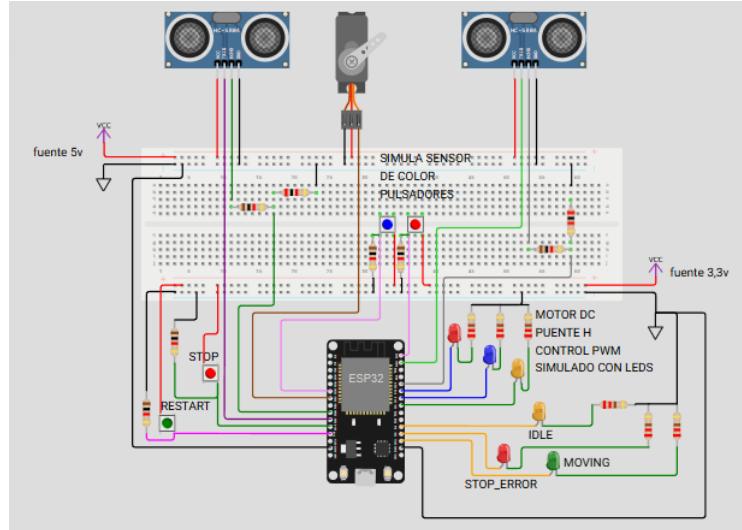
ST_ERROR (Error):

- Motor apagado.
- LED rojo parpadeando.
- LED verde apagado.
- LED amarillo apagado.
- El sistema queda detenido hasta que se presione RESTART.
- Causas típicas:
 - Timeout en detección de color.
 - Timeout en sensor final.
 - Color desconocido.

ST_MANUAL_STOP (Parada de emergencia):

- Motor apagado.
- LED rojo fijo encendido.
- LED verde apagado.
- LED amarillo apagado.
- El sistema queda detenido hasta que se presione RESTART.

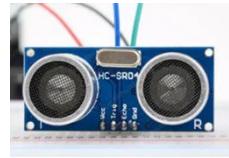
Circuito generado en Wowki



La implementación se realizó en una placa ESP32 con FreeRTOS, permitiendo manejar tareas concurrentes como el parpadeo de LEDs, la lectura de sensores y la ejecución de la máquina de estados.

Sensores y Actuadores Utilizados

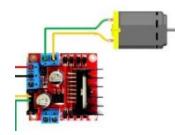
- 2 Sensores Ultrasónicos (HC-SR04). Ambos trabajan midiendo la distancia al objeto y comparándola con un umbral predefinido. Digital (usa pulsos digitales en TRIG/ECHO).
Alimentación: 5 V (consumo ~15 mA).
 - Sensor de inicio (Trigger1/Echo1) : Detecta la colocación de un objeto al inicio de la cinta.
 - Sensor de fin (Trigger2/Echo2): Verifica si el objeto llegó correctamente al final de la cinta.
- Sensor de Color (simulado por botones en este prototipo, reemplazable por TCS230 en la implementación real):
Detecta si el objeto es rojo, azul o desconocido.
En la simulación de Wokwi:
Botón 1 → Rojo Botón 2 → Azul



En la implementación final: El TCS230 medirá la frecuencia asociada a la reflexión de luz en cada filtro RGB. Es Digital (entrega una señal

PWM cuya frecuencia varía según el color detectado). Alimentación: 3.3 V o 5 V (consumo ~3 mA).

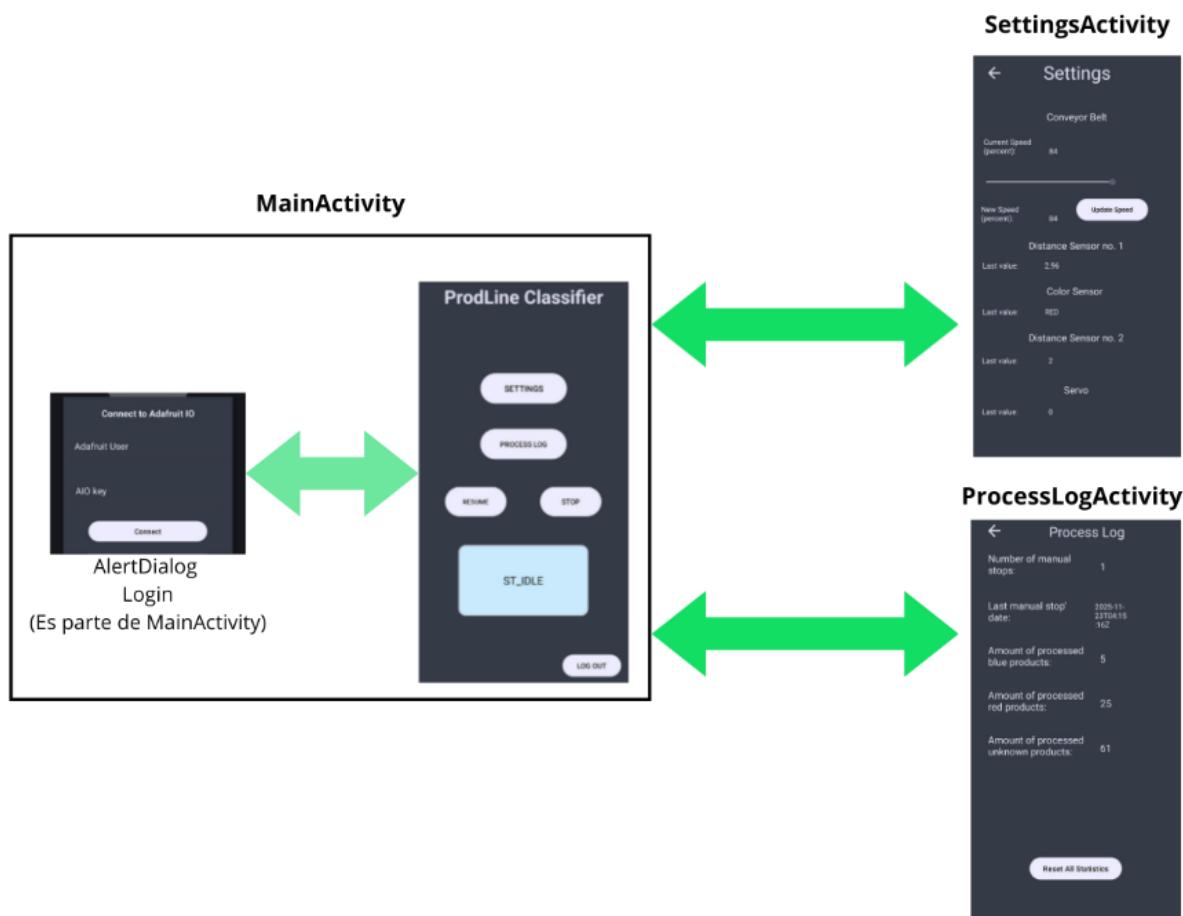
- Botón de Parada de Emergencia (STOP):
Interrumpe inmediatamente la cinta y enciende el LED rojo de error.
- Botón de Reinicio (RESTART):
Restablece el sistema y lo devuelve al estado inicial (ST_IDLE).
Grove - Button es un pulsador momentáneo. Contiene un botón independiente de encendido/apagado momentáneo. "Momentáneo" significa que el botón rebota por sí solo al soltarlo. El botón emite una señal ALTA al presionarlo y BAJA al soltarlo.

- Motor DC (controlado por puente H L298): utiliza dos pines de dirección y un PWM de velocidad; en la simulación se representa con tres LEDs que indican su funcionamiento. Se encarga del movimiento de la cinta transportadora.
Alimentación 9 V.
Consumo: típico 200–500 mA, con picos >1 A según carga.

- Servo Motor (clasificación): se posiciona en un ángulo según la señal PWM recibida en su pin de control. Dependiendo del color detectado. Alimentación: 5V. Consumo: Reposo: ~10 mA. En movimiento sin carga: 100–200 mA

- LEDs indicadores:
 - Amarillo (WAITING): Sistema en espera de objeto (ST_IDLE).
 - Verde (MOVING): Cinta en movimiento con objeto en proceso.
 - Rojo (STOP/ERROR): Encendido fijo → Parada manual.
Intermitente → Error detectado (tiempo excedido o color desconocido).
Dispositivo que permite el paso de corriente en un solo sentido y que al ser polarizado emite un haz de luz. Electrónicamente se conecta en serie con una resistencia limitadora para evitar que reciba más corriente de la que soporta. Se controla con una salida digital: encendido (HIGH) o apagado (LOW).

Aplicación Android

Diagrama de Navegación de Activities



Manual de Usuario – ProdLine Classifier

La aplicación ProdLine Classifier permite monitorear y controlar en tiempo real el sistema de clasificación por color desarrollado en el proyecto. A través de la app, el usuario puede visualizar el estado del proceso, enviar comandos remotos y consultar estadísticas históricas sin necesidad de interactuar físicamente con el equipo.

Requisitos de uso

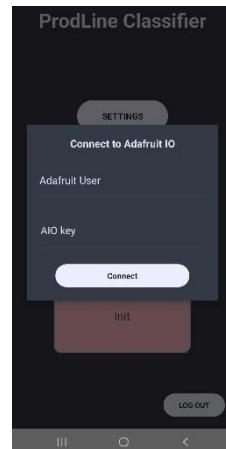
- Conexión a internet activa
- Cuenta en Adafruit IO (para login)
- Sistema embebido en funcionamiento y vinculado al broker MQTT Adafruit.

Inicio de sesión

Al abrir la aplicación por primera vez, aparece una ventana emergente solicitando credenciales: Adafruit User y AIO Key.

Una vez ingresados:

- Se inicia la comunicación con el broker MQTT
- Se actualiza automáticamente el estado del sistema
- Si las credenciales son incorrectas, se solicitarán nuevamente.



Pantalla principal

- Botón de RESUME: Reanuda el funcionamiento del sistema en el caso de una detención manual , estado ST_MANUAL_STOP.
- La funcionalidad de RESUME puede realizarse también realizando un movimiento de shake con el celular.
- Botón de STOP: Detiene la cinta de forma manual. Solo permite la detención si el sistema se encuentra en los estados ST_MOVING o ST_COLOR_DETECTED.

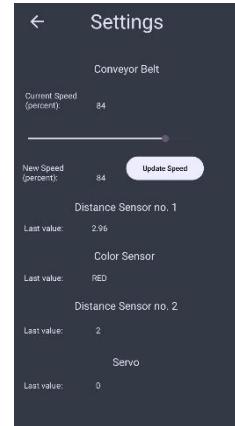


- Botón PROCESS LOG: Accede al registro histórico de eventos y productos procesados.
- Botón SETTINGS: Permite configurar parámetros del sistema y visualizar el estado de sensores.
- Botón LOG OUT: Cierra la sesión y borra las credenciales almacenadas.
- Indicador de estado: La pantalla muestra el estado actual del sistema, por ejemplo:
 - ST_IDLE (en espera)
 - ST_MOVING(cinta en movimiento)
 - ST_COLOR_DETECTED (color detectado)
 - ST_MANUAL_STOP (detención manual)
 - ST_ERROR (error o emergencia)

El color de fondo cambia según el estado para facilitar la interpretación visual.

Pantalla Settings

- Visualización de la velocidad actual de la cinta, Current Speed.
- Ajustar la velocidad de la cinta New Speed y enviarla al embebido mediante Update Speed.
- Ver los últimos valores reportados por:
 - Sensor de distancia nro. 1.
 - Sensor de color.
 - Sensor de distancia nro. 2.
 - Posición del Servo motor.



Pantalla Process Log

Muestra estadísticas acumuladas, entre ellas:

- Número de paradas manuales.
- Última fecha registrada de parada manual.
- Cantidad de productos procesados por color:
 - Red.
 - Blue.
 - Unknown

Incluye un botón para reiniciar las estadísticas si se desea comenzar un nuevo ciclo de producción.



Consideraciones de uso

Si el sistema entra en estado de error, no podrá iniciarse hasta ser reseteado desde el equipo físico, esto brinda una medida de seguridad forzando al responsable a acercarse a verificar visualmente la situación del sistema.

La app refleja la información en tiempo real; cualquier cambio físico se ve actualizado en pantalla.

Si la conexión se interrumpe, los valores se mostrarán en su última lectura disponible.

3 Conclusión

El proyecto permitió desarrollar un sistema de clasificación por color utilizando una cinta transportadora controlada por una ESP32, logrando automatizar el proceso de detección y separación de objetos de manera funcional. A lo largo del trabajo se integraron sensores, actuadores y una máquina de estados que respondió correctamente a las distintas situaciones del sistema, cumpliendo con los objetivos planteados al inicio.

Uno de los principales desafíos fue la implementación del sensor de color, ya que su comportamiento depende mucho de la iluminación, la distancia y el tipo de superficie del objeto. Esto hizo necesario realizar varias pruebas y calibraciones hasta obtener resultados aceptables. Además, la comunicación mediante MQTT también presentó cierta complejidad, especialmente para mantener la sincronización entre el sistema embebido, el broker en la nube y la aplicación móvil sin generar errores o pérdidas de datos.

La incorporación de la app Android aportó un gran valor al proyecto, ya que permitió monitorear el estado del sistema en tiempo real, enviar comandos de control y consultar estadísticas sin necesidad de interactuar físicamente con el hardware. Esto hizo que la solución fuera más completa y cercana a un entorno industrial real.

En conclusión, el desarrollo no solo cumplió los objetivos propuestos, sino que también dejó aprendizajes importantes para futuras mejoras, como avanzar hacia una detección de color más precisa, reforzar la robustez de la comunicación y ampliar las funcionalidades de supervisión y control del sistema.

4 Referencias

1. ElectronicWings, “TCS3200 Color Sensor Interfacing with ESP32,” [Online]. Available: <https://www.electronicwings.com/esp32/tcs3200-color-sensor-interfacing-with-esp32>
2. Seeed Studio, “Grove - Button,” [Online]. Available: <https://wiki.seeedstudio.com/Grove-Button/>
3. Naylamp Mechatronics, “Tutorial de uso del módulo L298N,” [Online]. Available: https://naylampmechatronics.com/blog/11_tutorial-de-uso-del-modulo-l298n.html.
4. R. Santos, “ESP32 Touch Pins with Arduino IDE,” Random Nerd Tutorials. [Online]. Available: <https://randomnerdtutorials.com/esp32-touch-pins-arduino-ide/>
5. UNLaM - Sistemas Operativos Avanzados, “Sistemas embebidos e Internet de las Cosas,” WikiSOA-UNLaM. [Online]. Available: https://www.soa-unlam.com.ar/wiki/index.php/PUBLICO:Sistemas_embebidos_e_Internet_d_e_las_Cosas
6. UNLaM - Sistemas Operativos Avanzados, “Apunte Teórico Sistema Operativo Android” WikiSOA-UNLaM. [Online]. Available: <https://soa-unlam.com.ar/material-clase/Android/Apunte%20Teorico%20Android.pdf>