



UNIVERSIDAD ESAN
FACULTAD DE INGENIERÍA

Manual Técnico de TomatAIT

Robot estacionario con cinta transportadora para detección de madurez de tomates

Curso: Robótica

Docente: Calderón Niquin, Marks

Integrantes:

Andrade Ortiz Alessandro

Condor García Kheyela

Mantilla Huaman Joe

LIMA-PERÚ

2025

ÍNDICE

1. Lista de Componentes:	3
1.1. Componentes en de Cinta Transportadora:.....	3
1.2. Componentes de Módulo de Detección:.....	5
1.3. Componentes Electrónicos.....	6
2. Construcción de Robot Estacionario:	9
2.1. Ensamblaje de Cinta Transportadora.....	9
2.2. Ensamblaje de Módulo de Detección.....	10
2.3. Ensamblado de componentes electrónicos.....	11
2.4. Ensamblado Final.....	13
● Fijación del Motor a la Cinta Transportadora.....	13
● Montaje del Módulo de Detección sobre la Cinta.....	13
● Posicionamiento del Sensor Ultrasónico (Detección de Entrada).....	14
● Integración del Mecanismo de Actuación.....	14
3. Instalación del Software:	15
3.1. Código Arduino MEGA.....	15
3.2. Código en Raspberry Pi Pico W.....	21
3.3. Instalación del software en la Raspberry Pi 5.....	24
4. Uso básico de TomatAIT:	28

1. Lista de Componentes:

1.1. Componentes en de Cinta Transportadora:



Figura 1: Vista en Corel Draw de Carril principal

La Figura 1 muestra los componentes estructurales más largos, diseñados para ser cortados con láser.

- **Rieles Laterales (Dos piezas largas con orificios):** Estas piezas conforman los **laterales longitudinales** del marco de la cinta transportadora. Su función es establecer la **longitud total** de la cinta y servir como soporte principal para los ejes de tracción y los rodamientos en sus extremos. Son la base de apoyo estructural.
- **Plataforma de Base (Una pieza rectangular grande):** Este rectángulo se utiliza como la **plataforma interna** que se coloca debajo de la banda azul, proporcionando una superficie plana y firme sobre la cual el material transportado (como el tomate) se deslizará, evitando que la banda se hunda.

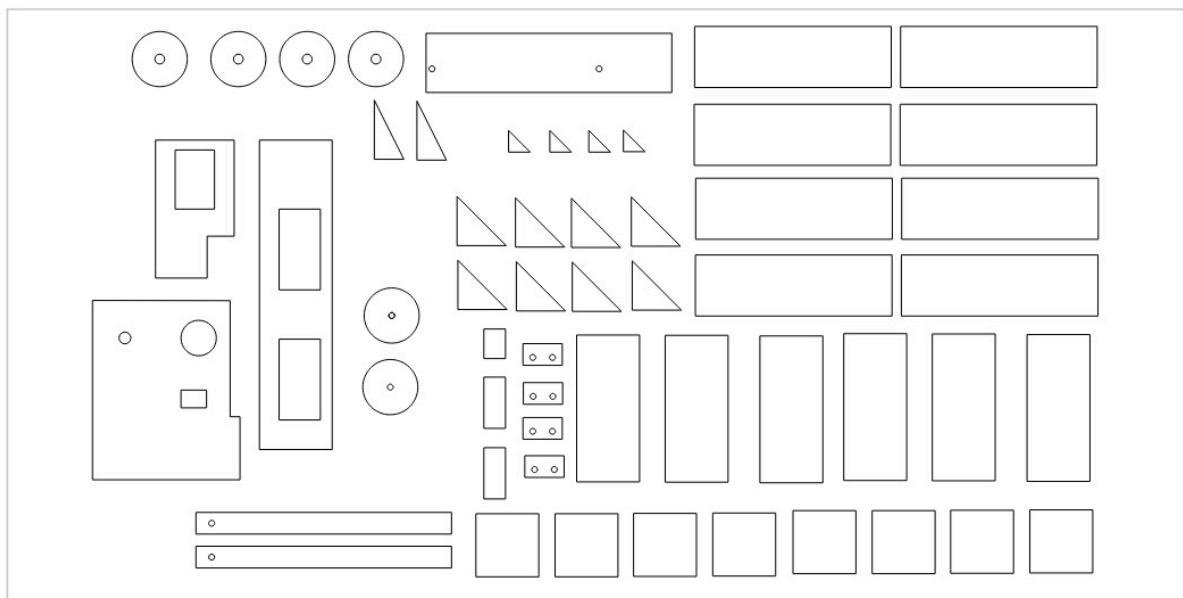


Figura 2: Vista en Corel Draw de Soportes

La Figura 2 muestra una colección de piezas auxiliares y soportes optimizados para el corte (nesting). Estas piezas aportan el soporte vertical y los puntos de montaje.

- **Soportes Verticales y Patas:** Se observan piezas rectangulares de diferentes tamaños que actúan como **patas o soportes verticales** para elevar la estructura del carril del suelo (visible en la Figura 3).
- **Soportes para Ejes y Rodamientos (Piezas circulares y cuadradas pequeñas):** Los círculos y pequeños rectángulos perforados están diseñados para fijar y asegurar los **rodamientos** y los **ejes** dentro de los rieles laterales de la Figura 1, garantizando que giren libremente.
- **Refuerzos y Piezas de Ensamblaje (Triángulos y piezas variadas):** Las pequeñas piezas triangulares y rectangulares más cortas se utilizan para dar **rigidez transversal** a la estructura y funcionar como **uniones encajables** (\textit{finger joints}) que conectan los rieles laterales con los soportes verticales.

1.2. *Componentes de Módulo de Detección:*

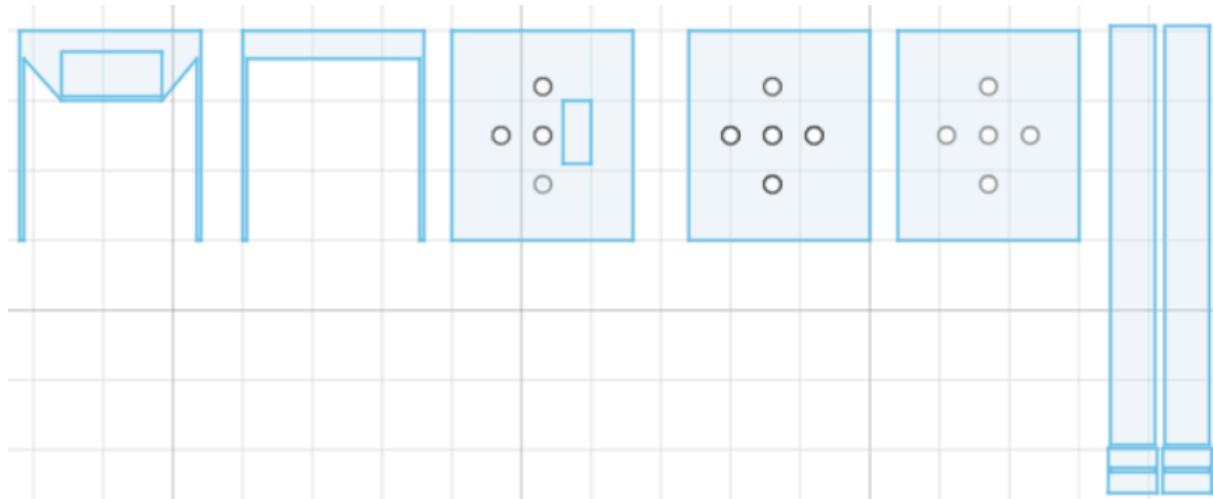


Figura 3: Vista en Fusion 360

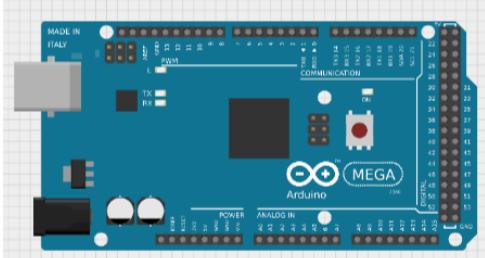
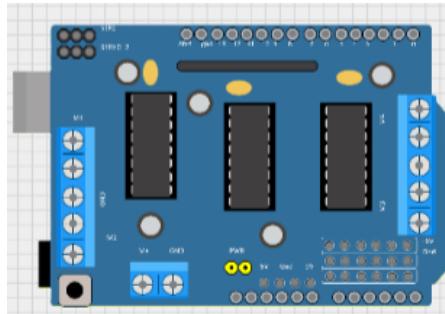
La Figura 3 muestra la vista de diseño de las piezas estructurales del Módulo de Detección, las cuales serán ensambladas para formar la caja que alberga la electrónica.

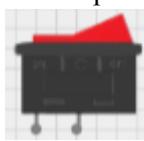
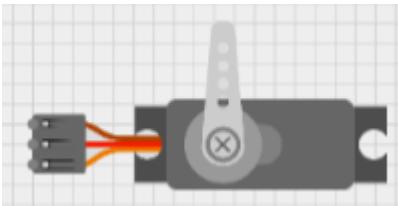
Los componentes principales que se utilizarán en el ensamblaje son:

- **Paredes Laterales (Las piezas más largas a la izquierda):** Son las paredes que definen la **altura y profundidad** del módulo. Estas piezas tienen grandes cortes rectangulares en la parte superior, que actúan como marcos o huecos para montar la placa superior de control y proporcionar acceso a la electrónica interior. En el ensamblaje (Figura 5), estas se convierten en los lados izquierdo y derecho del módulo central.
- **Soportes Inferiores/Patas (Dos piezas planas con base curva):** Estas son las "patas" del módulo, que se extienden hacia adelante. Su función principal es **dar estabilidad** y elevar el módulo para que pueda colocarse **sobre la cinta transportadora**.
- **Placas de Montaje Interno (Las piezas cuadradas con círculos):** Estas son cruciales para el montaje. Contienen orificios precisos para fijar componentes clave:

- **Orificio Central Grande:** Diseñado para la **lente de la cámara** o el sensor de visión que mirará hacia abajo, sobre la cinta.
- **Orificios Periféricos:** Utilizados para montar **sensores de distancia** (como el ultrasónico) o para atornillar la estructura interna de la electrónica (Arduino, motor driver).
- **Paneles Frontal/Trasero (Piezas delgadas a la derecha):** Se utilizan para cerrar la parte superior del módulo o actuar como tapas, algunas con ranuras para cables o interruptores.

1.3. Componentes Electrónicos

Imagen	Descripción
<p>Arduino Mega</p> 	Placa de control principal que ejecuta el código, procesa la señal de los sensores y emite las señales de control.
<p>Motor Driver (l293d)</p> 	Chip/Módulo controlador de motor (Puente H L293D). Recibe señales de bajo voltaje del Arduino y suministra la potencia necesaria (corriente) a los motores DC.
<p>Motor DC</p> 	Motorreductor DC con caja de engranajes, encargado de suministrar el torque y movimiento a la cinta transportadora.
<p>Sensor Ultrasonico</p>	Sensor Ultrasónico (HC-SR04), utilizado para

	<p>medir la distancia del objeto (tomate) y determinar si se encuentra en la zona de detección.</p>
<p>Interruptor</p> 	<p>Interruptor de encendido/apagado general que activa y desactiva el funcionamiento completo del sistema y del motor.</p>
<p>Servomotor</p> 	<p>Actuador de control angular, utilizado para el mecanismo de clasificación o desvío de los objetos sobre la cinta.</p>
<p>LEDs (13 uds)</p> 	<p>Diodos emisores de luz (13 unidades) utilizados para optimizar la iluminación.</p>
<p>Resistencias (13 uds)</p> 	<p>Resistencias de 330 Ohms (13 unidades) utilizadas para limitar la corriente y proteger tanto a los LEDs como a los pines de salida del Arduino.</p>
<p>Fuente de Energía</p> 	<p>Fuente de alimentación externa de 12 Voltios, requerida para suministrar la alta potencia (voltaje y corriente) necesaria para el Motor Driver (L293D) y el motor DC.</p>
<p>Raspberry Pi Pico</p> 	<p>Su función es actuar como el punto de comunicación inter-placas para el envío de los resultados del modelo YOLOv8 (desde la RPi 5) hacia el Arduino (para la actuación del clasificador).</p>
<p>Raspberry Pi 5</p>	<p>Realiza el procesamiento de la cámara (captura y gestión de imágenes) y la ejecución del modelo de visión YOLOv8 para la</p>

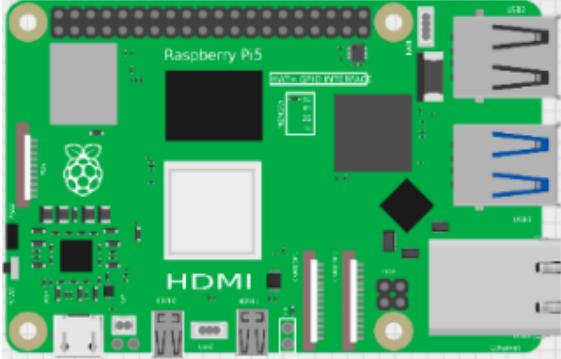
	detección de madurez.
Argom Tech USB HD Webcam 	Captura continuamente el flujo de video/imágenes sobre la cinta transportadora. El procesamiento de estas imágenes es cargado a la Raspberry Pi 5 para ser analizado por el modelo YOLOv8.

Tabla 1: Componentes electrónicos principales de TomatAIT

2. Construcción de Robot Estacionario

2.1. *Ensamblaje de Cinta Transportadora*

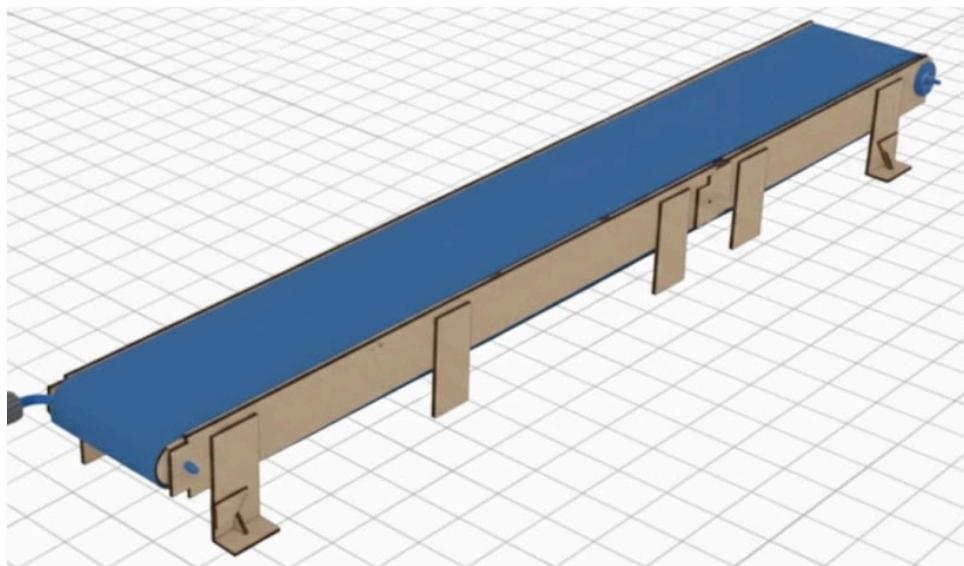


Figura 4: Cinta Transportadora ensamblada

El objetivo de ensamblaje es crear la estructura robusta y funcional que se muestra en la Figura 3, utilizando uniones machihembradas o encajables sin necesidad de adhesivos excesivos.

- **Montaje del Marco Longitudinal:** Se toman los dos **Rieles Laterales** (Figura 1) y se insertan en las ranuras de los **Soportes Verticales y Patas** más grandes (Figura 2). Este paso define la altura y la longitud de la cinta.
- **Fijación de la Plataforma de Base:** La **Plataforma de Base** (el rectángulo grande de la Figura 1) se coloca internamente, encajando en las ranuras inferiores de los rieles laterales, asegurando que la banda de la cinta tenga un apoyo firme en el centro.
- **Instalación de Ejes y Rodamientos:** En los extremos de los rieles laterales, se utilizan los **Soportes para Ejes y Rodamientos** (Figura 2) para fijar el eje de tracción (donde se conectará el motor) y el eje pasivo, que mantendrán la tensión de la banda.
- **Colocación de la Banda:** Una vez que la estructura es rígida, la **banda de la cinta** (el material azul) se coloca y se tensa alrededor del eje de tracción y el eje pasivo.

El resultado final es la **Cinta Transportadora ensamblada** (Figura 4), una unidad mecánica completa lista para ser acoplada al motor y a la estructura de la caja de control/detección.

2.2. *Ensamblaje de Módulo de Detección*

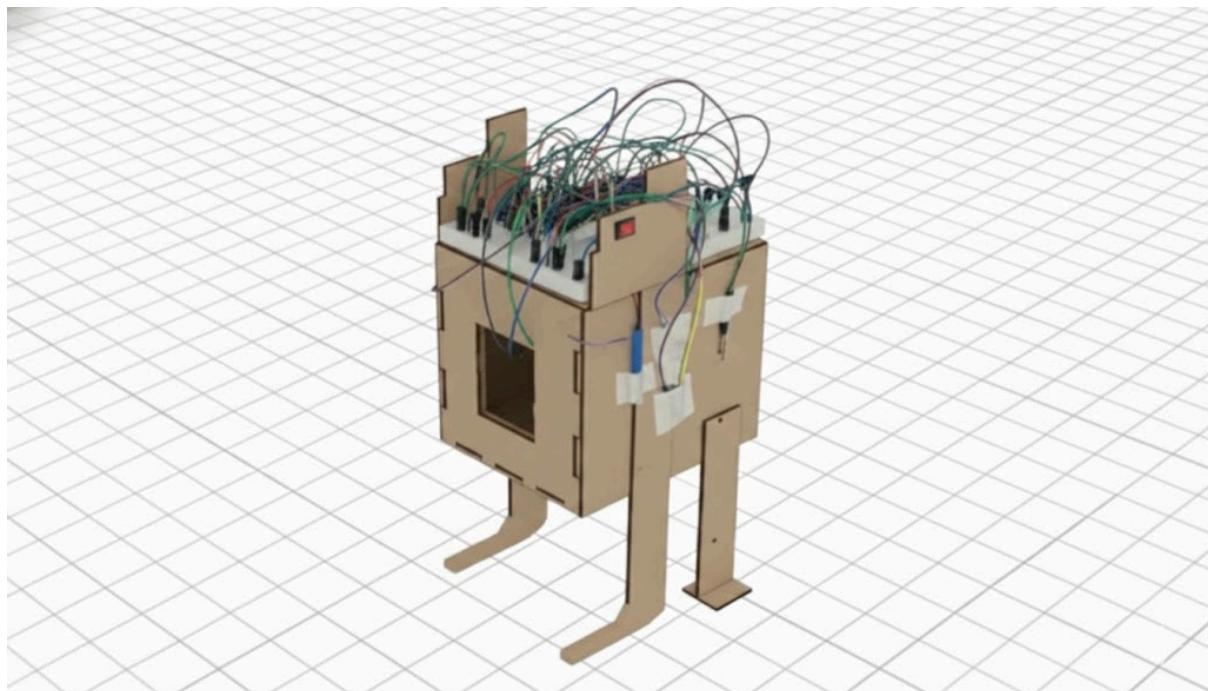


Figura 5: Módulo de Detección Ensamblado (referencial)

El ensamblaje de la Figura 5 tiene como objetivo construir una estructura de caja robusta y elevada a partir de las piezas planas cortadas con láser de la Figura 3.

- **Ensamblaje del Cuerpo Central (Caja):** Las **Paredes Laterales** (piezas largas) se unen perpendicularmente con el **Panel Frontal** y las **Placas de Montaje Interno** mediante uniones encajables. Esto forma la caja cuadrada que define el volumen interno para la electrónica.
- **Montaje de Soportes y Estabilización:** Los **Soportes Inferiores/Patas** se encajan en la parte inferior de la caja central. Estos soportes aseguran que la caja se mantenga estable y permiten que el módulo se "monte a horcajadas" sobre los rieles laterales de la cinta transportadora.
- **Integración de Sensores:** Una de las **Placas de Montaje Interno** se fija en el plano inferior del módulo, con el orificio central alineado sobre la cinta para permitir que la cámara o el sensor de visión enfoque el objeto transportado.
- **Cierre Superior:** La placa superior se fija en la parte superior, proporcionando una superficie para el **montaje y cableado** del circuito de control (visible en la Figura 5, con cables saliendo de la parte superior).

El resultado final (Figura 5) es un soporte vertical autónomo de madera (MDF) con una cavidad interna, listo para recibir los componentes electrónicos (Raspberry Pi, Arduino, drivers y sensores) y colocarse sobre la Cinta Transportadora.

2.3. *Ensamblado de componentes electrónicos*

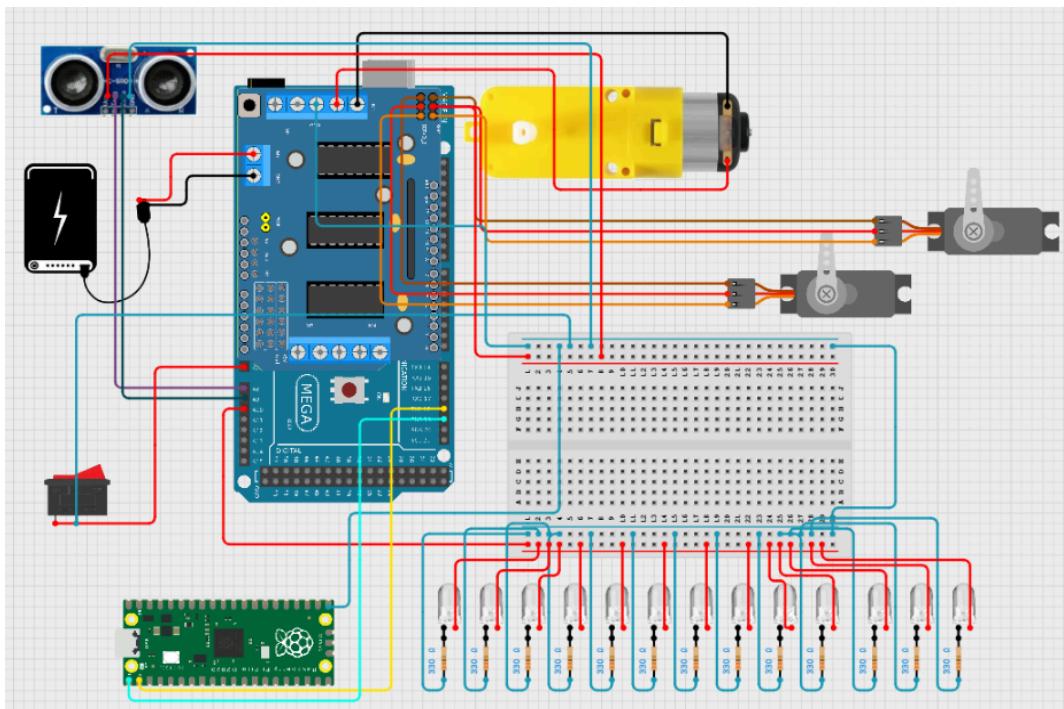


Figura 6: Conexiones de Componentes Electrónicos

El ensamblaje electrónico se centra en la integración del control de potencia (Driver/Motor) con la lógica de control (Arduino Mega) y el procesamiento de IA (Raspberry Pi 5).

- Integración del Módulo de Potencia y Control

Driver L293D (o similar): El Driver se monta sobre el Arduino Mega (o se conecta vía cables) y se alimenta de la Fuente de Energía Externa de 12V. Este driver gestiona la potencia del Motor DC (motorreductor de la cinta) y los Servomotores (actuadores de clasificación).

Motor DC y Servomotores: El motor de la cinta se conecta a las salidas del Driver L293D, mientras que los Servomotores se conectan directamente a los pines PWM del Arduino Mega o del shield (ej., pines 9 y 10), ya que el L293D no los controla.

- Integración de la Raspberry Pi 5 y Visión

Raspberry Pi 5 (RPi 5): Esta placa se instala para gestionar el procesamiento de la cámara. Recibe la señal de la Webcam USB (Argom Tech), donde se carga y ejecuta el modelo YOLOv8 para clasificar la madurez del tomate.

Comunicación Interna (MicroPython): La RPi 5, una vez que obtiene el resultado del modelo, utiliza un canal de comunicación (posiblemente a través de un módulo MicroPython / Pico o serial) para enviar la orden de acción (ej., "Maduro", "Verde", "Mohoso") al Arduino Mega.

- Sensores e Indicadores

Sensores: El Sensor de Distancia Ultrasónico y el Interruptor se cablean directamente a las entradas analógicas o digitales del Arduino Mega.

LEDs e Iluminación: Los 13 Diodos emisores de luz se conectan a los pines de salida del Mega (a través de sus Resistencias de 330Ω para limitar la corriente). Estos LEDs se utilizan para optimizar la iluminación del área de detección o como indicadores de estado del sistema.

2.4. *Ensamblado Final*

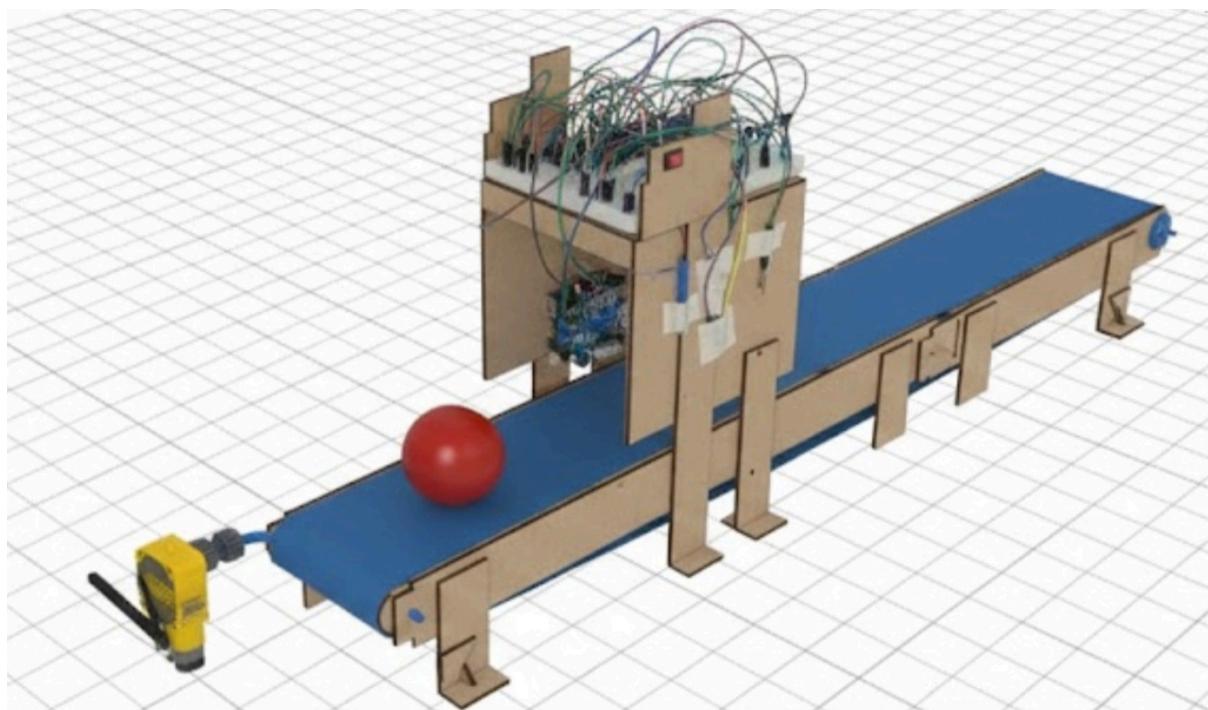


Figura 7: Vista general de ensamblado final (referencia)

Esta fase describe cómo las subestructuras de madera y los módulos electrónicos se unen para formar el prototipo final.

- **Fijación del Motor a la Cinta Transportadora**
 - El **Motor DC con caja reductora** se fija a un soporte de madera independiente, diseñado para alinear su eje con el **eje de tracción de la cinta transportadora** (Figura 3).
 - La unión entre el eje del motor y el eje de la cinta se realiza mediante un **acople flexible** (coupling) para compensar ligeras desalineaciones y garantizar una transmisión de movimiento suave y constante.
- **Montaje del Módulo de Detección sobre la Cinta**

- El **Módulo de Detección ensamblado** (Figura 5), que ya contiene la electrónica interna (Arduino Mega, RPi 5 y el cableado de la tapa superior), se coloca directamente sobre la estructura de la cinta transportadora.
- Las **patas curvas** del Módulo de Detección (Figura 5) están diseñadas para encajar y dar estabilidad a horcajadas sobre los **rieles laterales** de la Cinta Transportadora (Figura 3). Esta ubicación central permite que el sensor de visión y el mecanismo de actuación (servomotores) queden perfectamente **alineados sobre la banda azul**.

- **Posicionamiento del Sensor Ultrasónico (Detección de Entrada)**

- El **Sensor de Distancia Ultrasónico** se fija en el **extremo de entrada** de la Cinta Transportadora.
- Su función es **detectar la llegada** de un objeto (como el tomate) y emitir la señal al Arduino Mega para activar el proceso de lectura de la cámara y el motor de la cinta, iniciando el ciclo de clasificación.

- **Integración del Mecanismo de Actuación**

- Los **Servomotores**, si son parte del mecanismo de desvío, se montan en el exterior del Módulo de Detección o justo después de este, en un punto donde puedan empujar o desviar el objeto según la clasificación de la Raspberry Pi 5.
- El **Interruptor (Switch)** se instala en el orificio reservado de la placa superior del Módulo de Detección, actuando como el **punto de control manual** del sistema.

El sistema final, visible en la vista general, es una unidad compacta y autónoma que permite el flujo, la detección visual y la eventual clasificación de objetos.

3. Instalación del Software

3.1. Código Arduino MEGA

```
1 #include <AFMotor.h>
2 #include <Servo.h>
3
4 // =====
5 // CONFIGURACIÓN DE HARDWARE (ARDUINO MEGA)
6 // =====
7
8 // 1. MOTOR DE LA CINTA (Shield L293D - Puerto M1)
9 AF_DCMotor motorCinta(1);
10
11 // 2. SERVOS
12 Servo servoMaduro; // Pin 9
13 Servo servoVerde; // Pin 10
14
15 const int PIN_SERVO_MADURO = 9;
16 const int PIN_SERVO_VERDE = 10;
17
18 // 3. SENSORES Y BOTONES
19 const int switchPin = A7; // Interruptor Maestro
20 const int trigPin = A8; // Sensor Ultrasonico Trig
21 const int echoPin = A9; // Sensor Ultrasonico Echo
22 const int pinLed = A10; // Led indicador
23
24 // =====
25 // CALIBRACIÓN DE TIEMPOS
26 // =====
27
28 const int DISTANCIA_STOP = 12;
29
30 // VELOCIDAD DE TRABAJO (BAJADA A 70)
```

```
31 const int VELOCIDAD_TRABAJO = 70;
32
33 // TIEMPOS DE VIAJE
34 // Base para MADURO: 2 segundos
35 const unsigned long TIEMPO_VIAJE_MADURO = 2000;
36
37 // Base para VERDE: 1 segundo MÁS que el Maduro (Antes era +2000)
38 // Esto hace que active 1 segundo antes que la configuración previa
39 const unsigned long TIEMPO_VIAJE_VERDE = TIEMPO_VIAJE_MADURO + 1000;
40
41 // Tiempos de retención del brazo
42 const unsigned long TIEMPO_RETENCION_MADURO = 3000;
43 const unsigned long TIEMPO_RETENCION_VERDE = 5000;
44
45 // Tiempo de posicionamiento bajo cámara
46 const unsigned long TIEMPO_POSICIONAMIENTO = 800;
47 // =====
48
49 void setup() {
50     Serial.begin(9600);
51     Serial.println("SISTEMA TOMATAIT: LISTO (VELOCIDAD 70)");
52
53     // Comunicación con PICO W
54     Serial1.begin(9600);
55
56     // CONFIGURACIÓN MOTOR
57     motorCinta.setSpeed(VELOCIDAD_TRABAJO);
58     motorCinta.run(RELEASE);
59
60     pinMode(switchPin, INPUT_PULLUP);
61     pinMode(trigPin, OUTPUT);
62     pinMode(echoPin, INPUT);
63     pinMode(pinLed, OUTPUT);
64
65     servoMaduro.attach(PIN_SERVO_MADURO);
```

```

66     servoVerde.attach(PIN_SERVO_VERDE);
67
68     // === POSICIONES INICIALES DIFERENTES ===
69     // Maduro: Reposo en 180 (Lado A)
70     servoMaduro.write(180);
71
72     // Verde: Reposo en 0 (Lado B - Contrario)
73     servoVerde.write(0);
74 }
75
76 void loop() {
77     // === 1. VERIFICACIÓN MAESTRA ===
78     if (digitalRead(switchPin) == HIGH) {
79         apagarSistema();
80         return;
81     }
82
83     // === 2. BANDA AVANZANDO ===
84     digitalWrite(pinLed, HIGH);
85
86     // Velocidad constante a 70
87     motorCinta.setSpeed(VELOCIDAD_TRABAJO);
88     motorCinta.run(FORWARD);
89
90     // --- FILTRO DE SENSOR ---
91     long distancia = medirDistancia();
92
93     if (distancia > 0 && distancia <= DISTANCIA_STOP) {
94         // Doble confirmación
95         delay(50);
96         long confirmación = medirDistancia();
97         if (confirmación > 0 && confirmación <= DISTANCIA_STOP) {
98             Serial.println("TOMATE DETECTADO. POSICIONANDO...");
```

// Avanzamos para ponerlo bajo la cámara

99 if (!esperarSeguro(TIEMPO_POSICIONAMIENTO)) return;

```

101         // PAUSA LÓGICA PARA FOTO (Banda sigue rodando suave a 70)
102         Serial.println("Esperando clasificación IA (En movimiento)...");
103         while(Serial1.available() > 0) { Serial1.read(); }
104         bool datoRecibido = false;
105         char clasificación = 'N';
106         while (!datoRecibido) {
107             if (digitalRead(switchPin) == HIGH) { apagarSistema(); return; }
108             if (Serial1.available() > 0) {
109                 clasificación = Serial1.read();
110                 if (clasificación != '\n' && clasificación != '\r' &&
111                     clasificación != ' ') {
112                     datoRecibido = true;
113                 }
114             }
115             delay(10);
116         }
117
118         Serial.print("Clasificación: "); Serial.println(clasificación);
119
120         // === ACTUACIÓN SERVOS ===
121         motorCinta.setSpeed(VELOCIDAD_TRABAJO);
122         motorCinta.run(FORWARD);
123
124         if (clasificación == 'R') {
125             Serial.println("Viajando a MADURO...");
126             if (!esperarSeguro(TIEMPO_VIAJE_MADURO)) return;
127             Serial.println("SERVO MADURO (Normal)");
128
129             // Maduro usa lógica normal (180 -> 90 -> 180)
130             golpearNormal(servidor, TIEMPO_RETENCION_MADURO);
131         }
132         else if (clasificación == 'U' || clasificación == 'H') {
133             Serial.println("Viajando a VERDE...");
134             if (!esperarSeguro(TIEMPO_VIAJE_VERDE)) return;
135             Serial.println("SERVO VERDE (Invertido)");

```

```

136         // Verde usa lógica invertida (0 -> 90 -> 0)
137         golpearInvertido(servoVerde, TIEMPO_RETENCION_VERDE);
138     }
139     else {
140         Serial.println("Descarte.");
141         if (!esperarSeguro(5000)) return;
142     }
143     Serial.println("Fin de ciclo. Esperando salida del sensor...\\n");
144     esperarSeguro(2000);
145 }
146 }
147 delay(50);
148 }
149
150 // =====
151 // FUNCIONES AUXILIARES
152 // =====
153
154 void apagarSistema() {
155     digitalWrite(pinLed, LOW);
156     motorCinta.run(RELEASE);
157     // Guardar brazos en sus respectivos lados
158     servoMaduro.write(180);
159     servoVerde.write(0);
160 }
161
162 bool esperarSeguro(unsigned long tiempoMs) {
163     unsigned long inicio = millis();
164     while (millis() - inicio < tiempoMs) {
165         if (digitalRead(switchPin) == HIGH) {
166             apagarSistema();
167             return false;
168         }
169         delay(10);
170     }

```

```

171     return true;
172 }
173
174 long medirDistancia() {
175     digitalWrite(trigPin, LOW); delayMicroseconds(2);
176     digitalWrite(trigPin, HIGH); delayMicroseconds(10);
177     digitalWrite(trigPin, LOW);
178     long duración = pulseIn(echoPin, HIGH, 30000);
179     if (duración == 0) return 999;
180     return (duración * 0.034 / 2);
181 }
182
183 // --- FUNCIÓN GOLPE NORMAL (Para Maduro) ---
184 // Reposo: 180 -> Activo: 90
185 void golpearNormal(Servo &s, unsigned long tiempoRetencion) {
186     s.write(135);
187     if (!esperarSeguro(tiempoRetencion)) return;
188     s.write(180);
189     if (!esperarSeguro(500)) return;
190 }
191
192 // --- FUNCIÓN GOLPE INVERTIDO (Para Verde) ---
193 // Reposo: 0 -> Activo: 90
194 void golpearInvertido(Servo &s, unsigned long tiempoRetencion) {
195     s.write(25);
196     if (!esperarSeguro(tiempoRetencion)) return;
197     s.write(0);
198     if (!esperarSeguro(500)) return;
}

```

3.2. Código en Raspberry Pi Pico W

```

1 import network
2 import time
3 from machine import UART, Pin

```

```

4 from umqtt.simple import MQTTClient
5 # ===== CONFIGURACIÓN =====
6 SSID = "iPhone de Joe Jose"      # Tu WiFi
7 PASSWORD = "jugodemaracuya"      # Tu Contraseña
8 BROKER_IP = "172.20.10.9"        # <-- ;VERIFICA ESTA IP CON IPCONFIG!
9 TOPIC = b"robotica/frutas"      # Tema MQTT
10 # =====
11 # Comunicación con Arduino Mega
12 # GPO (TX Pico) -> Arduino Mega Pin 19 (RX1)
13 # GP1 (RX Pico) -> Arduino Mega Pin 18 (TX1)
14 uart = UART(0, baudrate=9600, tx=Pin(0), rx=Pin(1))
15
16 # 1. Conectar al WiFi
17 wlan = network.WLAN(network.STA_IF)
18 wlan.active(True)
19 wlan.connect(SSID, PASSWORD)
20
21 print("Conectando al WiFi (Hotspot iPhone)...")
22 intentos = 0
23 while not wlan.isconnected() and intentos < 20:
24     time.sleep(1)
25     intentos += 1
26     print(".")
27
28 if wlan.isconnected():
29     print('WiFi Conectado:', wlan.ifconfig())
30 else:
31     print('Error: No se pudo conectar al WiFi. Revisa la clave.')
32
33 # Función para procesar mensajes
34 def mensaje_recibido(topic, msg):

```



```

66 # Bucle Principal

67 while True:

68     try:

69         client.check_msg()

70         time.sleep(0.01)

71     except OSError:

72         try:

73             print("Reconectando...")

74             client.connect()

75             client.subscribe(TOPIC)

76     except:

77         pass

```

3.3. Instalación del software en la Raspberry Pi 5

```

1 import cv2

2 from ultralytics import YOLO

3 import sys

4 import paho.mqtt.client as mqtt

5 import time

6 # ===== CONFIGURACIÓN =====

7 # IP DE TU LAPTOP (Broker Mosquitto)

8 # Debe ser la misma que pusiste en la Pico W

9 BROKER_ADDRESS = "172.20.10.9"

10 TOPIC = "robotica/frutas"

11

12 # Umbral de confianza para disparo automático (75%)

13 CONF_UMBRAL = 0.75

14 # =====

15

16 print("Cargando modelo YOLO (best.pt) ...")

17 try:

18     model = YOLO('best.pt')

```

```

19 except Exception as e:
20     print(f"Error cargando modelo: {e}")
21     sys.exit(1)
22
23 # Configurar Cliente MQTT
24 client = mqtt.Client("RaspberryPi_Auto")
25
26 def run_auto_system():
27     print(f"Conectando a MQTT en {BROKER_ADDRESS}...")
28     try:
29         client.connect(BROKER_ADDRESS, 1883, 60)
30         print("CONEXIÓN MQTT EXITOSA.")
31     except Exception as e:
32         print(f"🔴 Error conectando al Broker: {e}")
33         print("    -> Verifica que Mosquitto esté corriendo en tu"
34 Laptop.)
35         print("    -> Verifica que el Firewall de Windows permita el"
36 puerto 1883.")
37     sys.exit(1)
38
39     print("Iniciando cámara...")
40     # Usamos backend V4L2 y formato MJPG para más velocidad
41     cap = cv2.VideoCapture(0, cv2.CAP_V4L2)
42     cap.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc('M', 'J', 'P',
43 'G'))
44     cap.set(cv2.CAP_PROP_FRAME_WIDTH, 800)
45     cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 600)
46     cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
47
48 if not cap.isOpened():
49     print("Error crítico: No se detecta cámara USB.")
50     sys.exit(1)
51
52 print("\nSISTEMA AUTOMÁTICO ACTIVO.")

```

```

53     print("Esperando tomate estático frente a la cámara...")
54
55     while True:
56
57         ret, frame = cap.read()
58
59         # Mostrar ventana (Si tienes monitor en la Pi)
60         cv2.imshow("Camara Clasificadora", frame)
61
62         if cv2.waitKey(1) & 0xFF == ord('q'): break
63
64         # 1. ANALIZAR CADA FRAME
65
66         # verbose=False para no llenar la consola de basura
67         results = model(frame, verbose=False)
68
69
70         for result in results:
71
72             for box in result.boxes:
73
74                 conf = float(box.conf)
75
76                 # Solo nos interesa si supera el 75% de certeza
77                 if conf > CONF_UMBRAL and conf > mayor_conf:
78
79                     mayor_conf = conf
80
81                     cls_id = int(box.cls)
82
83                     mejor_clase = model.names[cls_id]
84
85                     # 2. SI ENCONTRAMOS UN TOMATE CLARO
86
87                     if mejor_clase:
88
89                         print(f"DETECTADO: {mejor_clase} (Confianza:
90                               mayor_conf:.2f})")
91
92
93                     # Enviar MQTT a la Laptop
94
95                     client.publish(TOPIC, mejor_clase)
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
915
915
916
917
917
918
918
919
919
920
921
922
923
924
925
925
926
927
927
928
928
929
929
930
931
932
933
934
935
935
936
937
937
938
938
939
939
940
941
942
943
944
944
945
946
946
947
947
948
948
949
949
950
951
952
953
953
954
955
955
956
956
957
957
958
958
959
959
960
961
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
1651
1652
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1660
1661
1661
1662
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1670
```

```

87         print(f" Mensaje '{mejor_clase}' enviado. Pausando 5
88 seg...")
89
90         # 3. PAUSA (COOLDOWN)
91
92         # Esperamos 5 segundos para que:
93         # - La Laptop reciba el mensaje.
94         # - La Pico W se lo pase al Arduino.
95         # - El Arduino mueva la banda y clasifique.
96
97         time.sleep(5)
98
99         # Limpiar buffer de cámara (leer frames viejos para tener
100 imagen fresca)
101
102         for _ in range(5): cap.read()
103
104         print("Listo para el siguiente...")
105
106
107         cap.release()
108
109         cv2.destroyAllWindows()
110
111
112     if __name__ == "__main__":
113
114         run_auto_system()

```

4. Uso básico de TomatAIT

- **Preparación e Inicio**

- **Encendido de Energía:** Conecte la **Fuente de Energía Externa de 12V** al Driver L293D y, por separado, encienda la Raspberry Pi 5.
- **Activación del Sistema:** Active el **Interruptor (Switch)** en el Módulo de Detección. Esto energiza el circuito de control del Arduino Mega y enciende los **LEDs** utilizados para optimizar la iluminación del área de detección.

- **Monitoreo de Espera:** La Raspberry Pi 5 y el Arduino Mega inician, pero el **motor DC** de la cinta permanece **detenido**. El sistema está esperando la detección de un tomate.

- **Detección y Transporte**

- **Carga del Tomate:** Coloque un **tomate** en la parte inicial de la Cinta Transportadora, frente al **Sensor Ultrasónico**.
- **Detección de Entrada:** El Sensor Ultrasónico detecta el tomate. Esta señal informa al Arduino Mega que debe iniciar el proceso de clasificación.
- **Movimiento de la Cinta:** El Arduino Mega activa el Driver L293D para encender el **Motor DC**, transportando el tomate hacia el **Módulo de Detección**.

- **3. Procesamiento y Clasificación**

- **Captura de Imagen:** Mientras el tomate pasa por debajo del Módulo de Detección, la **Webcam USB** captura continuamente imágenes y las envía a la **Raspberry Pi 5**.
- **Inferencia de IA:** La Raspberry Pi 5 ejecuta el modelo de **visión YOLOv8** sobre las imágenes para determinar la clase de madurez (Verde, Pintón o Maduro).
- **Comunicación de Resultado:** La Raspberry Pi 5 utiliza el canal de comunicación (MicroPython/Pico) para enviar el resultado final de la clasificación al **Arduino Mega**.

- **Actuación y Detención**

- **Decisión del Motor:** El Arduino Mega recibe la orden. Si el tomate está **dentro del rango de detección efectivo**, el motor continúa moviéndose.

- **Clasificación (Actuación Futura):** En una implementación completa, el Arduino Mega activaría el **Servomotor** para desviar el tomate a su contenedor según la clasificación recibida (ej., desviación para "Maduro").
- **Detención de la Cinta:** Una vez que el tomate abandona la zona de detección/actuación, el Arduino Mega detiene el **Motor DC** (liberando el control del motor, RELEASE), y el sistema vuelve al estado de espera para el siguiente tomate.