**🧠 2. Detección de ArUco y envío al Arduino (Python)**

Este código:

* Detecta **ArUcos 4x4** del **1 al 50**.
* Dibuja la detección.
* Envía las coordenadas al Arduino como texto:  
  ID,x,y\n

Guárdalo como aruco\_eurobot.py:

import cv2

import numpy as np

import serial

import time

# ---------- CONFIGURACIÓN ----------

CAM\_INDEX = 0 # Cámara USB o PiCam

ARUCO\_DICT = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT\_4X4\_50)

PARAMS = cv2.aruco.DetectorParameters()

SERIAL\_PORT = 'COM3' # Cambia según tu sistema (ej: '/dev/ttyUSB0')

BAUDRATE = 115200

SEND\_INTERVAL = 0.1 # segundos entre envíos

# ---------- SERIAL ----------

try:

ser = serial.Serial(SERIAL\_PORT, BAUDRATE, timeout=0.1)

time.sleep(2)

print("✅ Conectado al Arduino en", SERIAL\_PORT)

except Exception as e:

print("⚠️ No se pudo conectar al Arduino:", e)

ser = None

# ---------- CÁMARA ----------

cap = cv2.VideoCapture(CAM\_INDEX)

if not cap.isOpened():

raise RuntimeError("No se pudo acceder a la cámara")

last\_send\_time = 0

while True:

ret, frame = cap.read()

if not ret:

break

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

corners, ids, \_ = cv2.aruco.detectMarkers(gray, ARUCO\_DICT, parameters=PARAMS)

if ids is not None:

cv2.aruco.drawDetectedMarkers(frame, corners, ids)

for i, corner in enumerate(corners):

marker\_id = int(ids[i][0])

if 1 <= marker\_id <= 50:

pts = corner.reshape((4, 2))

cx, cy = np.mean(pts[:, 0]), np.mean(pts[:, 1])

cv2.circle(frame, (int(cx), int(cy)), 5, (0, 255, 0), -1)

cv2.putText(frame, f"ID:{marker\_id}", (int(cx)+5, int(cy)-5),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0,255,0), 2)

# Enviar cada cierto tiempo

if ser and (time.time() - last\_send\_time > SEND\_INTERVAL):

msg = f"{marker\_id},{int(cx)},{int(cy)}\n"

ser.write(msg.encode('ascii'))

last\_send\_time = time.time()

cv2.imshow("Eurobot ArUco Tracker", frame)

if cv2.waitKey(1) & 0xFF == 27: # ESC

break

cap.release()

if ser:

ser.close()

cv2.destroyAllWindows()

**⚙️ 3. Código Arduino receptor**

Este sketch recibe las coordenadas por serial y muestra qué marcador se ve, además de poder ejecutar acciones diferentes según el ID.

Guárdalo como eurobot\_aruco\_serial.ino:

// Lee líneas con el formato "ID,x,y\n"

String inputLine = "";

void setup() {

Serial.begin(115200);

pinMode(LED\_BUILTIN, OUTPUT);

Serial.println("Arduino listo para recibir coordenadas.");

}

void loop() {

while (Serial.available() > 0) {

char c = Serial.read();

if (c == '\n') {

processLine(inputLine);

inputLine = "";

} else if (c != '\r') {

inputLine += c;

}

}

}

void processLine(String line) {

line.trim();

if (line.length() == 0) return;

int firstComma = line.indexOf(',');

int secondComma = line.indexOf(',', firstComma + 1);

if (firstComma == -1 || secondComma == -1) return;

int id = line.substring(0, firstComma).toInt();

int x = line.substring(firstComma + 1, secondComma).toInt();

int y = line.substring(secondComma + 1).toInt();

Serial.print("Detectado ID=");

Serial.print(id);

Serial.print(" en (");

Serial.print(x);

Serial.print(",");

Serial.print(y);

Serial.println(")");

// Ejemplo de acciones

if (id == 1) {

digitalWrite(LED\_BUILTIN, HIGH);

} else if (id == 2) {

digitalWrite(LED\_BUILTIN, LOW);

}

}

**🧭 4. Cómo probarlo**

1. Conecta la cámara y el Arduino.
2. Abre el puerto correcto en el script (COM3, /dev/ttyUSB0, etc.).
3. Imprime o muestra un **ArUco 4x4 ID 1–10** (puedes generarlos con OpenCV o https://chev.me/arucogen/).
4. Ejecuta el script Python.
5. Verás los marcadores en la ventana de vídeo y los datos llegando al **Monitor Serie del Arduino**.