

Explicación de Código - Comunicación OpenMV y Teensy

Código de OpenMV

Este código se ejecuta en la cámara OpenMV y está diseñado para detectar el color amarillo en la imagen capturada, y enviar un dato por UART a otro microcontrolador (como Teensy).

1. Importaciones de librerías necesarias:

- ``sensor``, ``image``, ``time`` son módulos propios de OpenMV para procesamiento de imagen.
- ``pyb.UART`` es el módulo de comunicación serie.

2. Configuración de la cámara:

- ``sensor.reset()`` reinicia la cámara.
- ``sensor.set_pixformat(sensor.RGB565)`` establece el formato de color.
- ``sensor.set_framesize(sensor.QVGA)`` define resolución 320x240.
- ``sensor.skip_frames(time = 2000)`` espera 2 segundos para estabilizar imagen.
- Se desactiva el auto-gain y auto-whitebalance para una detección de color más precisa.

3. UART:

- ``uart = UART(3, 19200)`` inicializa UART3 con velocidad de 19200 baudios (P4=TX, P5=RX).

4. Detección de amarillo:

- Se define un umbral de color amarillo en RGB565: ``(30, 100, -10, 40, 40, 100)``.
- ``img.find_blobs(...)`` busca zonas (blobs) que cumplan con ese umbral.

5. Comunicación:

- Si encuentra amarillo: se envía '1' por UART.
- Si no: se envía '0'.
- Además se imprimen mensajes para depuración en OpenMV IDE.

6. Dibujo:

- Se dibujan rectángulos y cruces sobre los blobs detectados.

Explicación de Código - Comunicación OpenMV y Teensy

Código en Teensy con zirconLib

Este código se ejecuta en un microcontrolador Teensy y está diseñado para recibir datos de la cámara OpenMV por UART, interpretarlos y controlar un motor usando la librería Zircon.

1. Librerías:

- ``Arduino.h``: funciones básicas de Arduino.
- ``zirconLib.h``: contiene funciones específicas para controlar hardware (como motores).

2. `setup()`:

- ``InitializeZircon()`` inicializa la plataforma Zircon.
- ``Serial.begin(9600)`` inicializa comunicación con la PC.
- ``Serial1.begin(19200)`` configura UART1 para comunicar con OpenMV.

3. `loop()`:

- Verifica si hay datos disponibles en Serial1 (UART con OpenMV).
- Si recibe '1': se detectó amarillo, se imprime el mensaje y activa el motor con ``motor1(100, 0)``.
- Si recibe '0': no se detectó amarillo, detiene el motor con ``motor1(0, 0)``.
- Si recibe otro dato: lo imprime como "dato no esperado".

Este código permite tomar decisiones en tiempo real según la información visual detectada por la cámara.