

SISTEMA DE CONTROL REMOTO DE SERVOS CON EL ROVER PARA MOVER LA CAMARA. UTILIZANDO ESP32.

REALIZADO POR: DEL VILLAR LICONA JOSÉ ANTONIO

Se solicito como parte del proyecto poder manejar de forma más precisa la mira en hora de hacer las pruebas del sistema de visión por computadora, inicialmente se prevé que estos sea la antesala de un posible algoritmo de seguimiento de objetos, por lo mientras el inicio de esta etapa 1 es poder controlar de forma manual el movimiento de los servos para poder hacer las pruebas de forma más fácil y cómoda. En este documento daremos la lista de acciones realizadas para realizar este propósito.

MANEJO DE LA ESP32 - GENERAL DRIVER FOR ROBOTS

Como sabemos esta placa esp32 de waveshare con nombre “General Drivers For Robots” es diseñada para poder manejar de forma mas sencilla con los servos ST3215 de la misma marca waveshare, esto servos se maneja por TTL en ves de PWM como cualquier servo comercial. En la siguiente imagen podrán observarse tanto el controlador (ESP32) y los servos (ST3215).

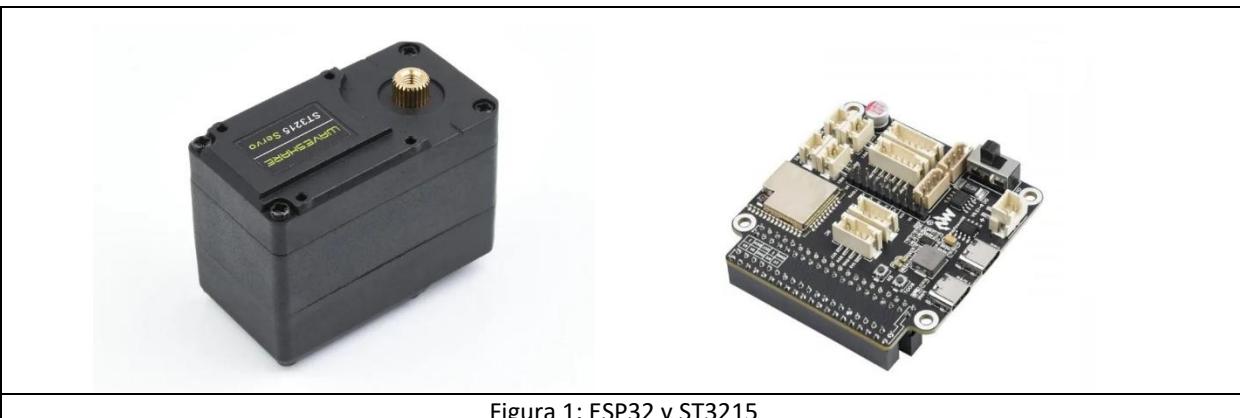


Figura 1: ESP32 y ST3215

Al hacer la conexión debía entre estos artículos fue montado sobre un Robert de wavesahre para una mejor estabilidad en una etapa dos se espera poder hacer que las llantas se muevan y el se pueda mover con mayor libertad.

COMO PREPARAR NUESTRA ESP32 – GENERAL DRIVER FOR ROBOT

La compañía de waveshare no es muy explicita y te hace dar demasiadas vueltas, Pero logramos obtener un método más sencillo de programar esta tarjeta de desarrollo.

- I. Paso uno descargar e instalar “Arduino IDE” de la página oficial.
- II. Paso dos entrar a Files y buscar preferencias
- III. Paso tres buscar URL adicionales y pegar esto: https://espressif.github.io/arduino-esp32/package_esp32_index.json
- IV. Paso cuatro después deben ir a <https://www.silabs.com/software-and-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads> y descargar el que necesiten ya sea Linux o Windows
- V. Paso cinco conecte su ESP32 a su dispositivo y actualice los drivers ya sea desde terminal (linux) o administrador de dispositivos (Windows)

PSEUDOCODIGO – PLANEACION DEL PROGRAMA PARA ESP32- GENERAL DRIVER FOR ROBOTS

Este programa busca establecer un control robusto y flexible de servos STS3215 conectados al ESP32 mediante comunicación TTL, con comandos recibidos desde la Jetson. La idea es lograr que los servos se muevan en dos ejes (X y Y) de forma suave, segura y parametrizable, permitiendo:

- Movimientos básicos (arriba, abajo, izquierda, derecha) controlados desde Python.
- Ajustes dinámicos de sensibilidad (cambiar el tamaño del paso).
- Modos de operación (rápido o delicado) que modifican velocidad, aceleración y paso.
- Sincronización inicial de la posición real de los servos para evitar saltos bruscos.



```
1 INICIO
2
3 Definir pines RXD y TXD para comunicación con servos
4 Definir límites de movimiento en eje X y Y
5 Definir variables de posición actual (posX, posY)
6 Definir parámetros iniciales: paso, velocidad, aceleración
7
8 SETUP:
9     Iniciar comunicación Serial con Jetson a 115200 baudios
10    Iniciar comunicación serial2 con servos a 1,000,000 baudios
11    Asociar objeto 'st' a Serial2
12    Esperar 1 segundo
13    Leer posición actual de servo Y (ID=2)
14        Si falla, asignar valor por defecto
15    Leer posición actual de servo X (ID=1)
16        Si falla, asignar valor por defecto
17
18 LOOP:
19    Si hay datos disponibles en Serial:
20        Leer carácter recibido
21
22    Si carácter == 'U':
23        Incrementar posY en 'paso' (respetando límites)
24    Si carácter == 'D':
25        Decrementar posY en 'paso' (respetando límites)
26    Si carácter == 'L':
27        Incrementar posX en 'paso' (respetando límites)
28    Si carácter == 'R':
29        Decrementar posX en 'paso' (respetando límites)
30
31    Si carácter == '+':
32        Aumentar tamaño del paso (máx. 250)
33    Si carácter == '-':
34        Disminuir tamaño del paso (mín. 5)
35
36    Si carácter == '1':
37        Configurar modo rápido (velocidad alta, aceleración alta, paso grande)
38    Si carácter == '2':
39        Configurar modo delicado (velocidad baja, aceleración baja, paso pequeño)
40
41    Enviar nueva posición a servo X (ID=1) con velocidad y aceleración actuales
42    Enviar nueva posición a servo Y (ID=2) con velocidad y aceleración actuales
43
44 FIN LOOP
45
```

PSEUDOCODIGO – PLANEACION DEL PROGRAMA PARA LA JETSON NANO

El objetivo es crear una interfaz de control en consola que permita al usuario manejar el robot (o servos conectados al ESP32) usando el teclado de la computadora.

- El programa abre un puerto serial (/dev/ttyUSB0) para comunicarse con el ESP32.
- Captura teclas en tiempo real sin necesidad de presionar Enter.
- Traduce las teclas en comandos simples (U, D, L, R, +, -, 1, 2) que el ESP32 entiende y convierte en movimientos de los servos.
- Permite cambiar sensibilidad (tamaño del paso) y modos de operación (rápido/agresivo o lento/delicado).
- Ofrece una salida clara en consola para que el usuario sepa qué acción se ejecutó.
- Finaliza la comunicación de forma segura al presionar Q o con Ctrl+C.

```
1  INICIO
2
3  Definir puerto serial (/dev/ttyUSB0)
4  Intentar abrir conexión a 115200 baudios
5      Si falla -> mostrar error y salir
6      Si éxito -> mostrar mensaje de conexión
7
8  Definir función getch():
9      Captura una tecla sin necesidad de Enter
10
11 Mostrar menú de control en consola:
12     Flechas = mover robot
13     + / - = ajustar sensibilidad
14     1 / 2 = cambiar modos
15     Q      = salir
16
17 BUCLE PRINCIPAL:
18     Leer tecla con getch()
19
20     Si tecla es ESC (secuencia ANSI):
21         Leer siguiente caracteres
22         Si 'A' -> enviar 'U' al ESP32 (arriba)
23         Si 'B' -> enviar 'D' al ESP32 (abajo)
24         Si 'C' -> enviar 'R' al ESP32 (derecha)
25         Si 'D' -> enviar 'L' al ESP32 (izquierda)
26
27     Si tecla es '+':
28         Enviar '+' al ESP32 (aumentar paso)
29     Si tecla es '-':
30         Enviar '-' al ESP32 (reducir paso)
31
32     Si tecla es '1':
33         Enviar '1' al ESP32 (modo rápido/agresivo)
34     Si tecla es '2':
35         Enviar '2' al ESP32 (modo lento/delicado)
36
37     Si tecla es 'q' o 'Q':
38         Mostrar mensaje de cierre
39         Romper bucle
40
41     Refrescar salida en consola
42
43     Si usuario interrumpe con Ctrl+C:
44         Mostrar mensaje de interrupción
45
46     Cerrar puerto serial
47     Mostrar mensaje de desconexión
48
49 FIN
```

RESULTADOS

Lo pedido fue logrado y documentados los programas se encontrarán en el repositorio, los resultados fueron satisfactorio agregamos una foto a lo que se logró.



Logramos hacer que los movimientos no tuvieran retraso y los movimientos fueron delicados para cuando el seguimiento de forma manual sede, sea perfecto. Ahora por último lo que mandamos a la terminal.

```
sudo /home/jetson/motor_control
jetson@nano:~$ fish
Welcome to fish, the friendly interactive shell
jetson@nano ~> cd motor_control/
jetson@nano ~/motor_control> sudo python3 servos.py
[sudo] contraseña para jetson:
✓ Conectado exitosamente a: /dev/ttyUSB0

=====
CONTROL DE ROBOT - MODO CONSOLA
=====
[↑][↓][←][→] : Mover Robot
[+] / [-] : Ajustar Sensibilidad
[1] / [2] : Cambiar Modos
[Q] : Salir del programa
=====

MOVIMIENTO: ABAJO RDAndo paso...
```