Guía de usuario Raspberry Pi Zero W

Guía para realizar la inicialización exitosa de la plataforma de robótica de enjambre

MRTG Hive

Introducción

Esta plataforma esta pensada para ser usada en algoritmos de robótica de enjambre, por lo cual esta guía únicamente sirve como apoyo para poder activar los diferentes módulos ya implementados.

Los módulos disponibles son:
-Control de la velocidad y sentido de los motores
-Accionamiento del servo motor y activación de los sensores ultrasónicos
-Activación de los comandos para el uso de la cámara
-Explicación del módulo de comunicación

Sistema de movimiento

Librerías necesarias

```
from gpiozero import PWMOutputDevice
from gpiozero import DigitalOutputDevice
from time import sleep
```

Comandos disponibles

Comando	Función
AllStop()	Detiene el movimiento de ambos motores
forwardDrive()	Ambos motores van hacia adelante a maxima velocidad
reverseDrive()	Ambos motores van hacia atras a maxima velocidad
spinLeft()	Se cambia de posición hacia la izquierda
spinRight()	Se cambia de posición hacia la derecha
forwardTurnRight()	Se avanza hacia la izquierda
forwardTurnLeft()	Se avanza hacia la derecha

El valor del PWM esta en 1000, este puede llegar a cambiarse según los valores maximos de la Raspberry

Aunque los pines de los motores ya se encuentran soldados si se desea de cambiar el lugar de los motores se puede hacer, unicamente se deben de cambiar los pines utilizados en la programación, el pin Stanby debe tener una señal positiva para activar el driver, si la señal esta en low no existe respuesta de los motores

```
PWM_DRIVE_LEFT = 21             # ENA - H-Bridge enable pin
FORWARD_LEFT_PIN = 26   # IN1 - Forward Drive
REVERSE_LEFT_PIN = 19   # IN2 - Reverse Drive
# Motor B, Right Side GPIO CONSTANTS
PWM_DRIVE_RIGHT = 5             # ENB - H-Bridge enable pin
FORWARD_RIGHT_PIN = 13  # IN1 - Forward Drive
REVERSE_RIGHT_PIN = 6   # IN2 - Reverse Drive
```

e imagenes

rerías necesarias

```
flask import Flask, render_template, Response, request
picamera import PiCamera
gpiozero import LED
threading import Condition
t time
t os
t io
t logging
t socketserver
ttp import server
```

nandos disponibles

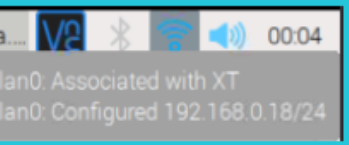
nando	Función
to()	Toma una foto en resolución 640X480
eo(X)	La 'X' simboliza la cantidad de tiempo que se desea realizar para tomar el video
am()	Esta función inicializa todo el stream

o stream se hizo en forma de
ón, pero en el codigo principal
de mandar a llamar el stream
de ir el comando output =
mingOutput()

```
if numero == 3:
    output = StreamingOutput()
    Stream()
```

```
ress = ('', 8000)
```

En address debemos seleccionar un puerto (8000) para conectarnos al server, este numero debe ser diferente al numero puesto para el puerto en la comunicación entre servidor y cliente



Para saber la IP (192.168.0.18) que se debe poner en el navegador para poder ver la transmisión en directo basta con colocarse en la conexión Wifi y ver el numero de IP

sensores ultrasonicos

Librerías necesarías

```
import time
import threading
from threading import Condition
from gpiozero import Servo
import RPi.GPIO as GPIO
```

Comando disponibles

Comando	Función
measureX(Y)	Esta función pone en funcionamieto la detección de objetos, 'X' corresponde al sensor que se esta midiendo y 'Y' a la cantidad de tiempo entre cada medición
servo.min	El servo se pone en la posición minima disponible
servo.mid	El servo se pone en la posición media
servo.max	El servo se pone en la posición maxima disponible
mov(x)	El servo se mueve a X posición dentro de los limites

Los sensores ultrasonicos puede llegar a tomar mediciónes cada 50ms

La unica recomendacíon es no forzar los limites del servomotor para evitar fallas mecanicas con el tiempo

tema de
municación

as necesarías

```
or
ocket
```

los disponibles

	Función
()	Función para poder enviar un mensaje desde el servidor

dor se inicia desde la Raspberry y este
ecutarse de primero para que la conexión
er existosa

te debe de conectarsea la dirección
idor y debe de ser enel puero
cado

```
cket = socket.socket()
cket.connect(('192.168.0.18',6000))
```

idor debe establecer una nueva dirección en la
alojará el servidor y establecer un nuevo puerto
unicación, este debe ser diferente al de la
a

```
ocket = socket.socket()
ocket.bind( ('0.0.0.0',6000))
```