Test Code Documentation

Release 1.0.0

Ali Ugur

Jun 24, 2022

CONTENTS:

CHAPTER

ONE

CONTROL HUB MAIN

This python file is for Control Hub to achieve communication between Test Robot and UI

class Control_Hub.control_hub.control_hub.**ClientNode**

This is the main Control Hub communication class

Parameters **Node** (*rclpy.Node*) – Main ROS2 node of the Control Hub

callback_button_state(*request*, *response*)

This callback function runs when a button read request comes from UI

Parameters

- **request** (*ROBOT.goal*) – request coming from UI to read button state
- **response** (*example_interfaces.srv.Trigger*) – response of button read value

Returns response

Return type example_interfaces.srv.Trigger

callback_ui_subscription(*msg*)

This callback function is run when a message comes from UI through topic. When a message is received that message is send to the Test Robot to activate or deactivate the LED.

Parameters **msg** (*string*) – input field data coming from UI

get_result_callback(*future*)

This callback function runs when the LED state response is acquired from Test Robot

Parameters **future** (*future*) – future object to acquire result

goal_response_callback(*future*)

This callback function runs when the LED activation/deactivation request is accepted by the Test Robot

Parameters **future** (*future*) – future object to acquire goal acception

send_goal(*gpio*)

This function is used to send the gpio data to the Test Robot

Parameters **gpio** (*string*) – input field data coming from UI

Control_Hub.control_hub.control_hub.**main**(*args=None*)

This is the main function that starts the ClientNode

CHAPTER

TWO

TEST ROBOT MAIN

This is an example code that is included in the pi_gpio module for ROS2 This example code is modified to work well with Control Hub Test Code

class Test_Robot.test_robot.test_robot.test_robot_server.**GPIOActionServer**

This class is used to handle ROS2 Action server for communication with Control Hub :param Node: Main node of the robot action server :type Node: rclpy.Node

cancel_callback(*goal*)

This callback runs when the current goal is cancelled

Parameters **goal** (*ROBOT.goal*) – Goal coming from Control Hub

Returns Cancel response acceptance

Return type CancelResponse.ACCEPT

destroy()

This function cleans the node

execute_callback(*goal_handle*)

Executes the goal. The goal can be LED control or reading the button state. If the goal is LED control then 3 is sent back as result. If the goal is reading button state then 0 or 1 is sent back as result.

Parameters **goal_handle** (*ROBOT_interface*) – goal from Control Hub

Returns result of execution

Return type ROBOT.result

goal_callback(*goal_request*)

This callback function is used to create goal responce accept message

Parameters **goal_request** (*ROBOT.goal*) – Goal request coming from Control Hub

Returns Goal acception

Return type GoalResponse.ACCEPT

handle_accepted_callback(*goal_handle*)

This callback function handles the goal coming from Control Hub

Parameters **goal_handle** (*ROBOT_interface*) – goal from Control Hub

class Test_Robot.test_robot.test_robot.test_robot_server.**RaspberryPIGPIO**(*pin_id*, *pin_type*)

This class is used for controlling Raspberry Pi GPIO

set_pin(*value*)

This function is used to set pins high or low :param value: pins value ro write :type value: int

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

t

INDEX