File: communication\_node.cpp

Author: Bill Johnson

Documentation Author: Andrew Burroughs

Detailed Description:

This node handles communication between the robot and the client. It receives information published by the power distribution node, motor nodes, and the logic node, then wraps it into messages that get sent to the client to display the information in the client side GUI. The client sends joystick information, keystroke information, and information about various buttons to the communication node and the node creates and publishes topics with the various information. It will be able to receive an image from the ZED node and publish this to the client, allowing the user to stream video.

Issues:

Not currently known if the client side GUI is receiving the information that is being passed. It currently does not display the information if it is being received.

Software Documentation

Global Variables

**std\_msgs::msg::Empty empty**

Creates a ROS2 message of type Empty that is published when the robot needs to stop running.

**bool silentRunning**

Boolean variable to track whether to send data back to the client.

**int new\_socket**

Global int variable used to track the socket port number.

**rclcpp::Node::SharedPtr nodeHandle**

This is an instance of the ROS2 pointer and is used to create the ROS2 node.

**std::string robotName**

This is a string variable used to hold the name of the robot, allowing for multiple robots to broadcast with different names.

**bool broadcast**

Boolean value used to determine whether the robot broadcasts its IP address to clients.

Function Documentation

**void insert**(float value, uint8\_t\* array):

Description of function

This function takes the value of the float and converts in into a byte array, then stores this byte array in the array parameter.

Expected Input

float value – Value to be stored.

uint8\_t\* array – Array to store the value in.

Expected Results

The float value will be stored in the array passed into the function.

**void insert**(int value, uint8\_t\* array):

Description of function

This function takes the value of the int and converts in into a byte array, then stores this byte array in the array parameter.

Expected Input

int value – Value to be stored.

uint8\_t\* array – Array to store the value in.

Expected Results

The int value will be stored in the array passed into the function.

**float parseFloat**(uint8\_t\* array):

Description of function

This function parses a uint8\_t array into a float variable using bit shifting.

Expected Input

uint8\_t\* array – byte array with a float value represented.

Expected Output

value – Float value of the byte array.

**int parseInt**(uint8\_t\* array):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void send**(BinaryMessage message):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void send**(std::string messageLabel, const messages::msg::VictorOut::SharedPtr victorOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void send**(std::string messageLabel, const messages::msg::TalonOut::SharedPtr talonOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void send**(std::string messageLabel, const messages::msg::Power::SharedPtr power):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void zedPositionCallback**(const messages::msg::ZedPosition::SharedPtr zedPosition):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void powerCallback**(const messages::msg::Power::SharedPtr power):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void talon1Callback**(const messages::msg::TalonOut::SharedPtr talonOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void talon2Callback**(const messages::msg::TalonOut::SharedPtr talonOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void talon3Callback**(const messages::msg::TalonOut::SharedPtr talonOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void talon4Callback**(const messages::msg::TalonOut::SharedPtr talonOut):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void zedImageCallback**(const sensor\_msgs::msg::Image::SharedPtr inputImage):

Description of function

This function will send the received image to the client. It currently does nothing except print a line to the logger that it received an image.

Expected Input

inputImage – Image from the ZED camera

Expected Results

The node should publish the image to the client. It currently only prints a message to the console.

**std::string getAddressString**(int family, std::string interfaceName):

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void printAddresses**():

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void reboot**():

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

**void broadcastIP**():

Description of function

Expected Input, Range(s) of Input

Expected Outputs / Results, Range

Change Log:

7/3/2022: Documentation was created