File: Automation.cpp

File Author(s): Bill Johnson, Andrew Burroughs

Documentation Author(s): Andrew Burroughs

Detailed Description:

This is the C++ implementation of the Automation.hpp header file. This file was created to implement the functions needed to handle the various parts needed for autonomous movement of the robot. This is currently not implemented as autonomy was scrapped due to time constraints.

Issues:

None currently known

Software Documentation

Global Variables

**driveLeftSpeedPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the left motors.

**driveRightSpeedPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the right motors.

**dumpBinPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the dump bin motor.

**shoulderSpeedPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the shoulder motor.

**excavationArmPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the excavation arm motors.

**excavationDrumPublisher**

This is a ROS2 publisher of type Float32 that publishes the speed data for the excavation drum motors.

**Rclcpp::Node::SharedPtr node**

This is a ROS2 node that is used to create the publishers used by this file.

**Position position**

This is a variable of type Position, as defined in AutomationTypes.hpp, that is used to hold the position values of the robot.

**Quaternion orientationQuaternion**

This is a variable of type Quaternion, as defined in AutomationTypes.hpp, that is used to hold the orientation values of the robot.

**EulerAngles orientation**

This is a variable of type EulerAngles, as defined in AutomationTypes.hpp, that is used to hold the orientation values of the robot, expressed in Euler Angles.

**float currentLeftSpeed**

This is a float variable that is used to track the speed that had been sent previously to the left wheel motors and is assumed to be the current motor speed.

**float currentRightSpeed**

This is a float variable that is used to track the speed that had been sent previously to the right wheel motors and is assumed to be the current motor speed.

Function Documentation

**void setNode**(rclcpp::Node::SharedPtr node):

This function is used to create the publishers that are needed by the node to publish the messages generated by the node.

Expected Input:

node – This is a shared ROS2 node that is passed by reference

Expected Results:

The function will set the node to the one passed and will create the publishers needed.

**void setPosition**(Position position):

This function is intended to convert the position coordinates to an orientation quaternion that is then converted to Euler angles using the toEulerAngles function. This was intended to be the starting point for the autonomous movement of the rover but was not fully implemented due to time constraints.

Expected Input:

position – This is a message of type Position, as defined in AutomationTypes.hpp, and contains the x, y, z, and w coordinates of the robot’s location

Expected Results:

The function will create the orientation quaternion, then pass that quaternion to the toEulerAngles and store the result in the orientation variable.

**void changeSpeed**(float left, float right):

This function was intended to change the speeds of the wheel motors. The function checks if the new values are the same as the previous values that are stored in currentLeftSpeed and currentRightSpeed. If the values are different, the new values are stored in the currentLeftSpeed and currentRightSpeed variables, then new Float32 messages are created and published using the driveLeft and driveRight publishers.

Expected Input:

left – This is a float variable with range [-1.0, 1.0] that represents the speed of the left motors

right – This is a float variable with range [-1.0, 1.0] that represents the speed of the right motors

Expected Results:

The function will set the new motor speed values and publish the new values. The motors will adjust to match the new speeds given.

**EulerAngles toEulerAngles**(Quaternion q):

This function is intended to convert the orientation coordinates from a quaternion to the standard Euler angles. The functions are complicated and additional information would be good here.

Expected Input:

q – Variable of type Quaternion, as defined in AutomationTypes.hpp, containing information about the robot’s orientation

Expected Outputs:

This function will return the calculated values of roll, pitch, and yaw from the four values of the quaternion.

**void changeDumpBinSpeed**(float speed):

This function is intended to set the speed of the dump bin motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

**void changeShoulderSpeed**(float speed):

This function is intended to set the speed of the shoulder motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

**void changeArmSpeed**(float speed):

This function is intended to set the speed of the excavation arm motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

**void changeDrumSpeed**(float speed):

This function is intended to set the speed of the excavation drum motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

**void changeLockServoSpeed**(float speed):

This function is intended to set the speed of the dump bin lock mechanism servo motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

**void changeArmServoSpeed**(float speed):

This function is intended to set the speed of the dump bin door servo motor. It receives a float value of the speed and publishes a message to set the speed of the motor.

Expected Input:

speed – This is a float variable with range [-1.0, 1.0] that represents the speed of the motor

Expected Results:

The function should create a message of type Float32, set the data to the speed, then publish the message and cause the motor to adjust its speed.

Change Log:

7/3/2022: Documentation was created

8/3/2022: Added Description and Issues, Software Documentation, and Function Documentation