DEVELOPER'S MANUAL

The execution of the program starts from the main(Strin args[]) function of GUIApp1 class. The main function creates and displays the JFrame form the default constructor of GUIApp1 class is called. Within the constructor initComponents() function is called. The initComponents() function initializes all the JTextField, JLabel, Jbutton, JProgressBar anf JSlider and adds them to the JFrame form. It also sets the border, foreground, background, font etc of the various GUI components and assigns ActionListener and ChangeListener to the buttons and sliders respectively.
After the execution of initComponents() method, object of SerialPortConnection class is made then the listSerialPorts() method is called which lists all the available ports in the JComboBox. All the buttons except the Connect button are disabled by using setEnabled(boolean) function and remains disable till the time a successful connection has been made. To make a succeccful connection, user needs to select a correct COM port from JComboBox and click on the connect button is clicked then the connect(portName) method of SerialPortConnection class which tries to connect the selected COM port with the GUI. If connection is successful then button of the GUI becomes enabled and ReadThread is started which writes "T" on the output stream using writeOnTerminal(String) method so as to tell rotot that the sensors value needs to be sent by it. When the data ia available in the input stream serialEvent method of SerialEventHandler class implementing SerialPortEventListener interface is called.
After the successful connection user click on any button and the action associated with that button is performed by writing on the output stream. If the connection is unsuccessful corresponding exception is displayed on the screen.
To disconnect from the robot user clicks on dissconnect button. After this the ReadThread is stopped and removeSerialPorts() and removeGUIComponents() is called to close the serial port and removes the readings of sensors.

Methods

1. **jButtonBuzzerActionPerformed()**

   Input: None
   Return: void
   Description: Checks the current status of the buzzer stored in the variable statusBuzzer. If statusBuzzer is 0 then it writes "7" on the terminal to turn ON the buzzer and if statusBuzzer is 1 then it writes 9 on the terminal to turn OFF the buzzer and correspondingly updates the statusBuzzer.

2. **jButtonForwardMotionActionPerformed()**

   Input: None
   Return: void
   Description: Makes the robot move forward by writing "8" on the output stream.

3. **jButtonRightMotionActionPerformed()**

   Input: None
   Return: void
   Description: Makes the robot move right by writing "4" on the output stream.

4. **jButtonBackwardMotionActionPerformed()**

   Input: None
   Return: void
   Description: Makes the robot move backward by writing "2" on the output stream.

5. **jButtonLeftMotionActionPerformed()**

   Input: None
   Return: void
   Description: Makes the robot move left by writing "6" on the output stream.

6. **jButtonStopMotionActionPerformed()**

   Input: None
   Return: void
   Description: Makes the robot stop by writing "5" on the output stream.

7. **jButtonCOMConnectActionPerformed()**

   Input: None
   Return: void
   Description: Gets the selected COM port from the JComboBox by using getSelectedItem() function and calls the connect(String portName) function from SerialPortConnection class and passes the selected port as a parameter to make connection.

8. **jButtonCOMDisconnectActionPerformed()**

   Input: None
   Return: void
   Description: Calls removeGUIComponents() method to disable all the enabled GUI components, calls removeSerialPorts() method to close the output and input stream and closes the serial port. It also enables the connect button by calling setEnabled() function

9. **getSliderValueLeftMotor()**

   Input: None
   Return: void
   Description: Gets the changed value of slider by calling the getValue() function and then sets the value of the label by calling setText() function.

10. **jTextFieldLeftMotorActionPerformed()**

    Input: None
    Return: void
    Description: Gets the value entered in the jTextFieldLeftMotor by calling getText() method and then sets the slider according to that value by calling setValue() method on jSliderLeftMotor.

11. **getSliderValueRightMotor()**

    Input: None
    Return: void
    Description: Gets the changed value of slider by calling the getValue() function and then sets the value of the label by calling setText() function.

12. **jTextFieldRightMotorActionPerformed()**

    Input: None
    Return: void
    Description: Gets the value entered in the jTextFieldRightMotor by calling getText() method and then sets the slider value by calling setValue() method on jSliderRightMotor

13. **jButtonSetVelocityActionPerformed()**

    Input: None
    Return: void
    Description: Gets the left and right motor velocity from their respective TextFields and performs a check whether any of the velocity is left blank. If so, then prompts the user to set a valid velocity. Once a valid velocity has been entered then it writes "R" on the outputstream to indicate that the velocity of the robot has to changed and then writes both the velocities on the outputstream.

14. **jButtonResetVelocityActionPerformed()**

    Input: None
    Return: void
    Description: Resets the velocity of left and right motors of robot by writing "S" on the output stream and then sending the initial velocity of robot, i.e., 255.

15. **getSliderValueServo1()**

    Input: None
    Return: void
    Description: Gets the value of angle by which the servo motor s1 has to be rotated by calling the method getValue() of JSlider class and sets the text field to that value using setText() method of JTextField class

16. **getSliderValueServo2()**

    Input: None
    Return: void
    Description: Gets the value of angle by which the servo motor s2 has to be rotated by calling the method getValue() of JSlider class and sets the text field to that value using setText() method of JTextField class

17. **getSliderValueServo3()**

    Input: None
    Return: void
    Description: Gets the value of angle by which the servo motor s3 has to be rotated by calling the method getValue() of JSlider class and sets the text field to that value using setText() method of JTextField class

18. **jButtonForwardMovementActionPerformed()**

    Input: None
    Return: void
    Description: Sends "U" which is 0x54 in hex to the robot to identify next values are for forward movement by some distance. Gets the value of the distance by which robot has to move forward from the jTextFieldDistance by calling getText() method. The value obtained is divided and modulo by 255 to send the values greater than 255.

19. **jButtonBackwardMovementActionPerformed()**

    Input: None
    Return: void
    Description: Sends "V" which is 0x55 in hex to the robot to identify next values are for backward movement by specified distance. Gets the value of the distance by which robot has to move forward from the jTextFieldDistance by calling getText() method. The value obtained is divided and modulo by 255 to send the values greater than 255.

20. **jButtonRightRotationActionPerformed()**

    Input: None
    Return:
    Description: Sends "W" which is 0x56 in hex to the robot to identify next values are for right rotation of robot by specified angel. Gets the value of the angel by which the robot has to rotate from the jTextFieldDistance by calling getText() method. The value obtained is divided and modulo by 255 to send the values greater than 255.

21. **jButtonLeftRotationActionPerformed()**

    Input: None
    Return: void
    Description: Sends "X" which is 0x57 in hex to the robot to identify next values are for right rotation of robot by specified angel. Gets the value of the angel by which the robot has to rotate from the jTextFieldDistance by calling getText() method. The value obtained is divided and modulo by 255 to send the values greater than 255.

22. **jButtonServo1ActionPerformed()**

    Input: None
    Return: void
    Description: Sends 0x80 to identify that the next value is angel to be rotated by servo motor 1. Gets the value of the angle by which motor has to be rotated from the jTextFieldServo1 by calling getText method and then sends the value to the outputstream to rotate the motor.

23. **jButtonServo2ActionPerformed()**

    Input: None
    Return: void
    Description: Sends 0x81 to identify that the next value is angel to be rotated by servo motor 2. Gets the value of the angle by which motor has to be rotated from the jTextFieldServo2 by calling getText method and then sends the value to the outputstream to rotate the motor.

24. **jButtonServo3ActionPerformed()**

    Input: None
    Return: void
    Description: Sends 0x82 to identify that the next value is angel to be rotated by servo motor 3. Gets the value of the angle by which motor has to be rotated from the jTextFieldServo3 by calling getText method and then sends the value to the outputstream to rotate the motor.

25. **jButtonLCDPrintActionPerformed()**

    Input: None
    Return: void
    Description: Sends 0x83 to identify that next 3 bytes are to print on LCD. Gets the row no, column no and the character to be printed on the LCD screen by calling getText method and then sends these values to the output stream

26. **jButtonBarGraphLedActionPerformed()**

    Input: None
    Return: void
    Description: Sends 0x84 to identify that next value is to glow specified bar graph LED. Gets the number of bar graph led to be glown from the jTextFieldBarGraphLed by calling getText method and then writes the corresponding hex value on the output stream

27. **jTextFieldServo1ActionPerformed()**

    Input: None
    Return: void
    Description: Gets the value from the jTextFieldServo1 by calling getText method, then converts it into an integer value and finally sets the value of slider by calling setValue() method.

28. **jTextFieldServo2ActionPerformed()**
    Input: None
    Return: void
    Description: Gets the value from the jTextFieldServo2 by calling getText method, then converts it into an integer value and finally sets the value of slider by calling setValue() method.

29. **jTextFieldServo3ActionPerformed()**
    Input: None
    Return: void
    Description: Gets the value from the jTextFieldServo2 by calling getText method, then converts it into an integer value and finally sets the value of slider by calling setValue() method.

30. **listSerialPorts()**

    Input: None
    Return: void
    Description: List all the available ports in JComboBox. Calls removeEventListener() method of SerialPort class to stop the event that occurs while reading from terminal and also closes the serial port, input and output stream

31. **removeGUIComponents()**

    Input: None
    Return: void
    Description: enable disable various components of GUI and removes text from JTextFiel and jLabel and set the value of all the jProgressBar to 0.

32. **connectGUIComponents()**
    Input: None
    Return: void
    Description: Enable all the jTextField and jLabels and starts the ReadThread to start reading the sensors value.

33. **connect(String)**

    Input: String
    Return: void
    Description: Connects GUI with the selected COM port. Calls connectToPort(portName) method of SerialPortConnection class in order to connect to the robot. If connection is not possible due to some exception then displays the corresponding exception message in a dialog box.

34. **setSerialEventHandler()**

    Input: None
    Return: void
    Description: Sets the serial event handler by adding the event listener.
    addEventListener() and notifyOnDataAvailable() method of serialport class is called to set the serial event handler.

35. **Sharp_Distance_Sensor_estimation (int value)**

    Input: int
    Return: int
    Description: Converts the analog value of sharp IR sensor into distance in mm by using the formula:
    distance = 10.00*(2799.6*(1.00/(Math.pow(value,1.1546))))

36. **setIcon()**

    Input: None
    Return: void
    Description: Sets the icon of GUI by calling setIconImage(Image) method

37. **removeSerialPorts()**

    Input: None
    Return: void
    Description: Calls removeEventListener() method of SerialPort class to stop the event that occurs while reading from terminal and also closes the serial port, input and output stream

38. **serialPorts()**

    Input: None
    Return: string[]
    Description: Ports are enumerated using getPortIdentifiers() method of CommPortIdentifier class and stored in an array using ArrayList and then converted into String array and returned to the calling function

39. **writeOnTerminal(String serialMessage)**

    Input: String
    Return: void
    Description: Converts the serialMessage into bytes by using the function getBytes() and then writes those bytes on the outputstream by using the function write(bytes). It then calls the function flush() to empty the outputstream.

40. **connectToPort(String portName)**

    Input: String
    Return: int
    Description: Calls the getPortIdentifier(portName) method of CommPortIdentifier class to obtain a CommPortIdentifier object and then open the communication channel of that port by using open(String, int) method of CommPortIdentifier class. After this it sets the baud rate, databits, stopbits and parity of the selected port. The function -
    Returns 1 if successful connection has been made
    Returns 2 if UnsupportedCommOperationException is thrown
    Returns 3 if PortInUseException is thrown
    Returns 4 if NoSuchPortException is thrown

41. **setInputOutputStream()**

    Input: None
    Return: void
    Description: Gets the input stream and output stream of the connected COM port by calling the function getInputStream() and getOutputStream().