Thomas Brefeld, Jr.

Chan-Jin Chung, PhD

Senior Project 1

ROS River

Abstract:

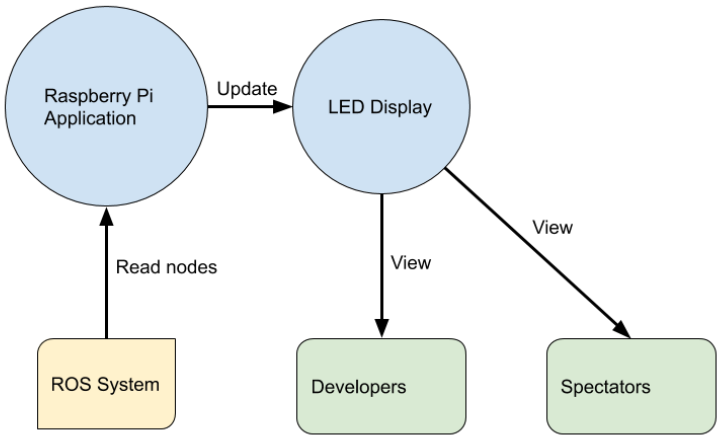
Understanding what the Robot Operating System (ROS) is currently performing is only available to developers actively working within that environment. The aim of ROS River was to create a display that would relay relative information about the ROS system to spectators and external developers. ROS River displays simplified versions of the current ROS system state and predefined task names. ROS River performs all these tasks effectively, but has room to become better. Next step for ROS River is to improve management of devices and offer more control messages being displayed.

Introduction:

ROS River is an application that will display information about the Robot Operating System (ROS) to the user via a LED display. This application displays current tasks, issues, and other debugging information desired by the user. ROS River has uses in autonomous vehicle development where persons outside the vehicle can see and understand the car's current task. The application runs on a Raspberry PI B and is connected via ethernet to the main ROS core. The Raspberry PI subscribes to certain nodes within the main ROS core and displays the information to the LED Display. In the future, the subscribed nodes can be changed on a website that can be accessed through the Raspberry Pi's Bluetooth. This will allow for quick updates to which topics will be shown on the display.

System Diagram:

The system is designed to be an independent system outside of the ROS system it is connected to. The Raspberry Pi connects to the ROS core through ethernet and no setup is required on the ROS core side, this allows the program to run with a wide veracity of programs with little or no changes to ROS River. Since the ROS River runs in a separate system from the ROS core the display can continue to display the status of the ROS core even if a fatal error occurs within the ROS core environment.



Limitations and future additions:

One limitation is to be able to connect to the ROS core the IP of the computer running the ROS core must be pre defined within the ROS River environment. This can be solved in the future with a more advanced connection protocol. Another current limitation is the reduced flexibility of the application, since the only way to manipulate the program is to SSH into the ROS river environment, this adds a layer of difficulty when changing the program. For future development we can implement a website over bluetooth to enable quick and easy changes. The final limitation is the program only has one brightness setting for the display, with the addition of a Light Dependent Resistor (LDR) the program would be able to adjust the brightness of the display in accordance with the environment. Another limitation is how fast the display can update, which is currently 500ms, this can be improved by changing the display language to C++ from python.

Hardware:

Part	Price
Raspberry Pi 3B	$35
DC Voltage Regulator 12v to 5v	$23
WS2812B LED Display 8 x 32 Pixels	$24
Miscellaneous (project box, wires, and connectors)	$25
Total Cost	**$107**

Test Cases:

User Function Name	User Function Description	Expected output or results	Self-Testing Results	Notes
ROS communication	Test that the Raspberry Pi can connect in many different situations	Successful communication between the two systems	Communication between a master Laptop and Raspberry Pi works as expected	Currently, need IP of Raspberry Pi and Master Computer
ROS node update	Test that the Raspberry Pi can see node updates	Successful node update	The Raspberry Pi can receive node updates in a callback effectively	
LED Display communication	Test that the Raspberry Pi properly displays to the LED display	The display is visible and updates	The LED display can print scrolling text	Need to add more functionality than just text
LED Displays communication for status	Test that the Raspberry Pi properly displays the status number to the LED	The display is visible and updates independently from the text	The display does show the status along without affecting the scrolling text	
ROS master connection reliability	Can the raspberry reconnect to the master if connection is lost	The Raspberry Pi will wait until it is able to reconnect to the master	The raspberry pi does wait and will reconnect to the master	
More functions for LED Display	Add a status bar at the bottom for quick color notifications	Will display a color 1 high and 32 pixels wide at the bottom of the display without interfering with the scrolling text	The LED display can display status bar at the bottom	Limited to manual operation mode

User Function Name	User Function Description	Expected output or results	Self-Testing Results	Notes
Automatic Launching	The Raspberry Pi and application will launch without user	When powered the Raspberry Pi will boot and launch the application without user	The Raspberry Pi is able to boot and launch the Application is a reasonable amount of time	
LED Display rigidity	Can the LED display hold onto the glass for an extended period of time	It is expected that the LED Display will be able to hold itself against the glass without falling	After leaving the Display on the glass there seems to be no loss in suction	
Subscribers to manipulate the program	Host subscribers that allow the user to edit the program by publishing to ROS topics	The topics should be easily available and reliable for manipulating data	The subscribes successfully allow manipulation of the program	
LED Display status	The LED Display should show one LED for use in identifying update rate	It is expected that the LED will blink on and off every update of the display	The program successfully updates the LED for status	In the future will add an option to turn this off

Conclusion:

The application does perform as expected and does achieve the goal set out at the beginning of the semester. Even with the limitations defined prior, the program is still useful in its current state and does accurately display information about the ROS system. With the developments defined in the limitations and future additions ROS River will be a very capable and useful system for developers.

Links:

Github: https://github.com/Thomasbrefeld/ROS_River

Demo: https://youtu.be/Xvg98JJ5lbI