CompE571 Embedded Operating Systems Term Project: Project Proposal

***Title: Bot Sensing Simulator with Raspbian OS Task Optimization***

Names: Daniel Foster (817895293) and Vahe Ohanian (818027724)

Contact/Email: [dfoster1130@gmail.com](mailto:dfoster1130@gmail.com) and [vaheohanian@gmail.com](mailto:vaheohanian@gmail.com)

Project Description:

We will implement a sensing system for a bot/vehicle designed for scientific exploration of unknown environments. The bot, implemented using a Raspberry Pi 3 Model B microcontroller and tri-wheel bot chassis kit, will move in fixed patterns that will allow our sensing module to pick up measurements and move according to these measurements. For example, according to outside environmental conditions such as temperature or light intensity, the bot will be able to schedule simultaneous tasks/movements from sensory inputs and showcase the management of different priority tasks. Our sensing module will be implemented using the Raspberry Pi 3 microcontroller and a Sense Hat add-on board capable of measuring temperature, humidity, pressure, and movement; all of these sensors would be realistic components of a ground, space, or air vehicle. For the OS component of this project, we plan to modify the Raspbian OS for optimization of our particular tasks. By modifying the kernel for our Raspberry Pi, we hope to increase execution time and provide real-time outputs that are accurate for different priority tasks.

Reasoning/Applicability:

Interested by the concept of the Mars rover that was deadlocked by priority inversion, we thought a bot that could manage low, medium, and high priority tasks with priority inheritance would be unique and would be very applicable to real-world scientific and research scenarios. In a hard real-time system such as this, multitasking must be done without error or risk large financial or information loss. We believe that extending the default functionality of the provided Raspbian OS is not only a difficult enough task to consider this project unique and satisfactory for this class, but we think that this skill will likely help us in the future as the Raspberry Pi is growing in its popularity and useful applications.

Difficult Components:

As we have never worked with a Raspberry Pi or Python before, we hope to be able to pick up Python quickly and implement the Sense Hat add-on board without error. Furthermore, as we have never coded or modified instructions in the kernel of an OS, and we are still not to the point where we feel comfortable doing so, we will need to do extensive research on Raspbian OS, as well as researching task scheduling algorithms and other possibilities for kernel modification. Lastly, our choice of project, which is incredibly applicable in the real world as stated above, has limitless options for further development, some of which we may explore or discuss implementing over the course of project work.

Hardware and Software Implementation:

For this project we plan to implement the following hardware and software –

1. Hardware

* Raspberry Pi 3 Model B
* Raspberry Pi Sense Hat add-on board
* Tri-wheel bot chassis with battery pack and dual motors
* Battery pack/on-chassis power
* L298N motor drive controller board
* Miscellaneous wires, connectors, adaptors

1. Software

* Python (Sensing modules and bot control)
* Raspbian OS (task management/scheduling)

Metrics:

We will measure the success of our project based on:

* Reliability of sensing inputs and outputs
* Optimization of Raspbian OS
* Response-time/execution time
* Task management and scheduling
* Energy efficiency/memory management (*if time permits*)

Experiments/Results:

All deadlocks should be prevented, and as different test environments are ran using the different sensing measurements, scheduling between tasks should be done with execution time and reliability in mind. For example, if the bot is to interact with a light source (ex. flashlight), it should stop the bot or move it in a different direction. If it encounters this task, along with another lower priority task (ex. high heat reading requiring movement in the opposite direction), it should execute the higher priority task and stall the lower priority task. Results include all sensors giving measurements in real time and with a high percentage of correctness, as well as good response time overall for the system; both of these results would be necessities in a real-world, scientific application.

Deliverables:

1. Progress Report

* Bot assembly and movement (no interfacing with sensors at this point)
* Raspberry Pi setup and Sense Hat implementation
* Single sense demo and operation
* Raspbian OS introductory interfacing

1. Project Demo/Final Report

* Working bot with sensory inputs/outputs
* Live demonstration/video of sample test sensory environment (using at least 2 sensing mechanisms)
* Raspbian OS modification for enhanced metrics (ex. task scheduling)
* Pertinent data that supports results
* Problems faced/modification to project design

Timeline/Milestones:

|  |  |  |
| --- | --- | --- |
| Task Number | Description | Delivery Date |
| 0 | Submit project proposal | September 25th, 2017 |
| 1 | Obtain hardware/software components (shipping) | October 1st, 2017 |
| 2 | Assemble bot components and research necessary hardware components | October 11th, 2017 |
| 3 | Set-up Raspberry Pi 3 Model B, configure/install Raspbian OS | October 11th, 2017 |
| 4 | Raspberry Pi Sense Hat implementation/interfacing | October 16th, 2017 |
| 5 | Implement sensing module (ex. light sensitivity) | October 18th, 2017 |
| 6 | Start research and modification of Raspbian OS, look at scheduling and task management as it relates to our system, implement simple bot movement | October 25th, 2017 |
| 7 | Prepare progress report and sample demo | October 30th, 2017 |
| 8 | Implement remaining sensor modules (ex. gyroscope, temperature, etc.) | November 13th, 2017 |
| 9 | Modify Raspbian OS for maximum optimization of our project and confirm working bot sensing system | November 29th, 2017 |
| 10 | Test using practical mock environments and collect data | December 6th, 2017 |
| 11 | Prepare final report and presentation, demo working system with at least 2 unique sensors | December 11th, 2017 |