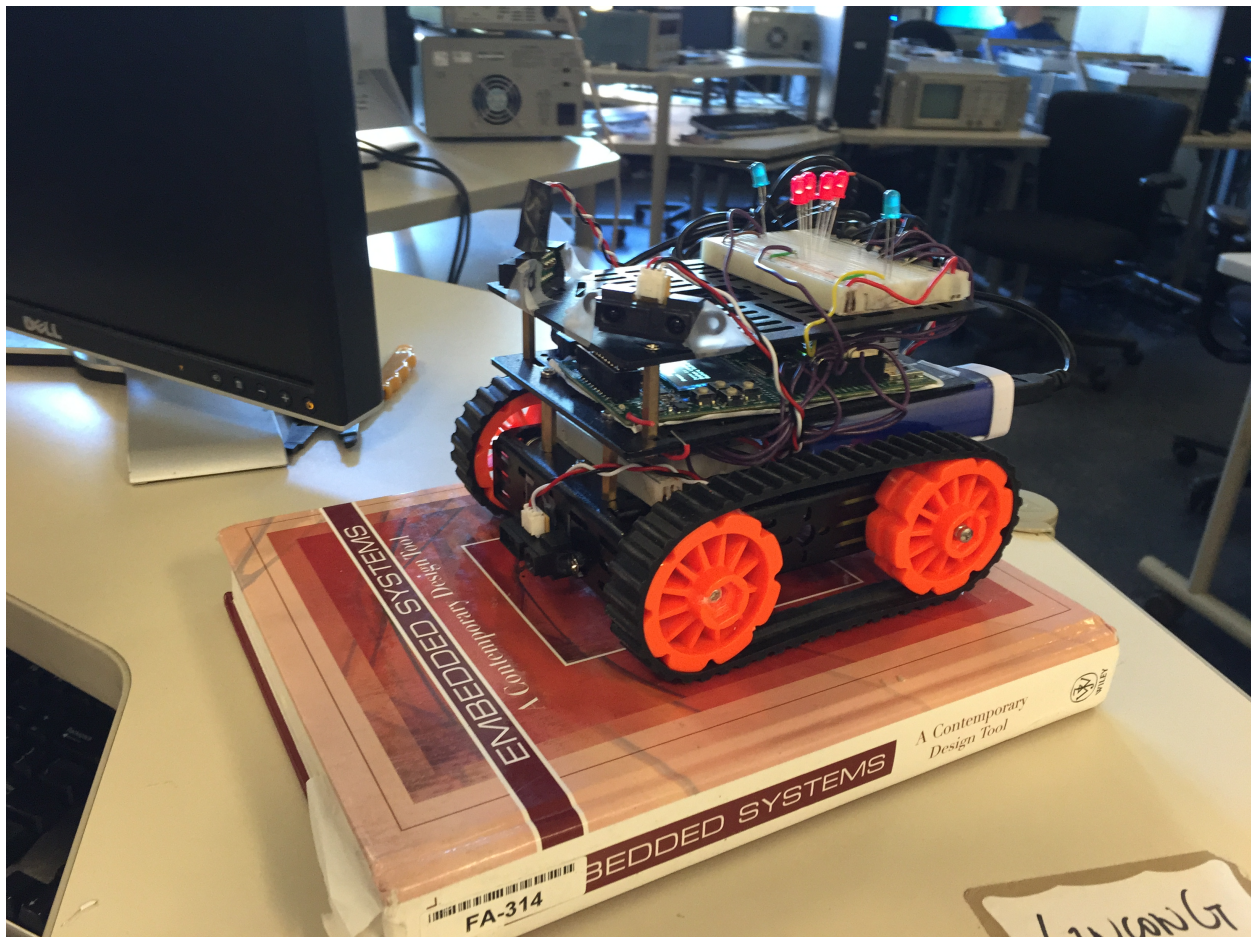


*University of Washington
Department of Electrical Engineering
EE 472, Spring 2015*

Report for Lab 5: The RoboTank

Completing the RoboTank System.

*Report by:
Denny Ly
Eeshan Londhe
Ruchira Kulkarni*



INTRODUCTION

The purpose of this lab is to develop the RoboTank system. After completing Lab 4, we had our tank controllable via keypad button inputs on the Stellaris board. However, for Lab 5, our goals for the RoboTank system were to implement Bluetooth controllability through a Mac OS app called Blueterm. To achieve this, we learned how UART works to be able to use Bluetooth to serially send/receive data through the Blueterm app. In addition to this, we wanted to create a self-driving autonomous system that avoids all collisions. To improve the wireless controllability, we will also be adding a semi-autonomous mode where the tank avoids collisions even when you are controlling the tank through the app. The final product will be a fully functional Bluetooth controlled, collision avoiding RoboTank System.

The key subsystems of the RoboTank are:

- Distance Sensors (ADC's)
- PWM Signals (for the motors)
- Bluetooth connectivity
- Blueterm (Mac OS app)
- Autonomous driving mode
- User Interface
- Keypad Interface

DATASHEETS AND REFERENCES

Some good references to do this lab are to look at the data sheet for both the Stellaris board and the microcontroller as well as looking at the FreeRTOS website to learn their API's. Also, we used the Bluetooth module's data sheet to configure Bluetooth connectivity to our tank through a Mac OS application called Blueterm.

- Stellaris LM3S8962 Evaluation Board User's Manual
Stellaris.pdf
- Texas Instruments Stellaris LM3S8962 Microcontroller Data Sheet (Manual)
lm3s8962.pdf
- Texas Instruments ADC Oversampling Techniques for Stellaris Family Microcontrollers
spma001a.pdf
- Sharp GP2Y0A21YK0F Distance Sensor
1489_Sharp_GP2Y0A21YK0F.pdf
- Optrex Dot Matrix Character LCD Module User's Manual
OptrexMan.pdf
- Toshiba TB6612FNG dual DC motor driver IC
TB6612FNG.pdf
- Bluetooth module guid Bluetooth_guide.pdf
- Bluetooth module datasheet rn-42-ds.pdf

- <http://www.freertos.org/>

CONTROL FLOW

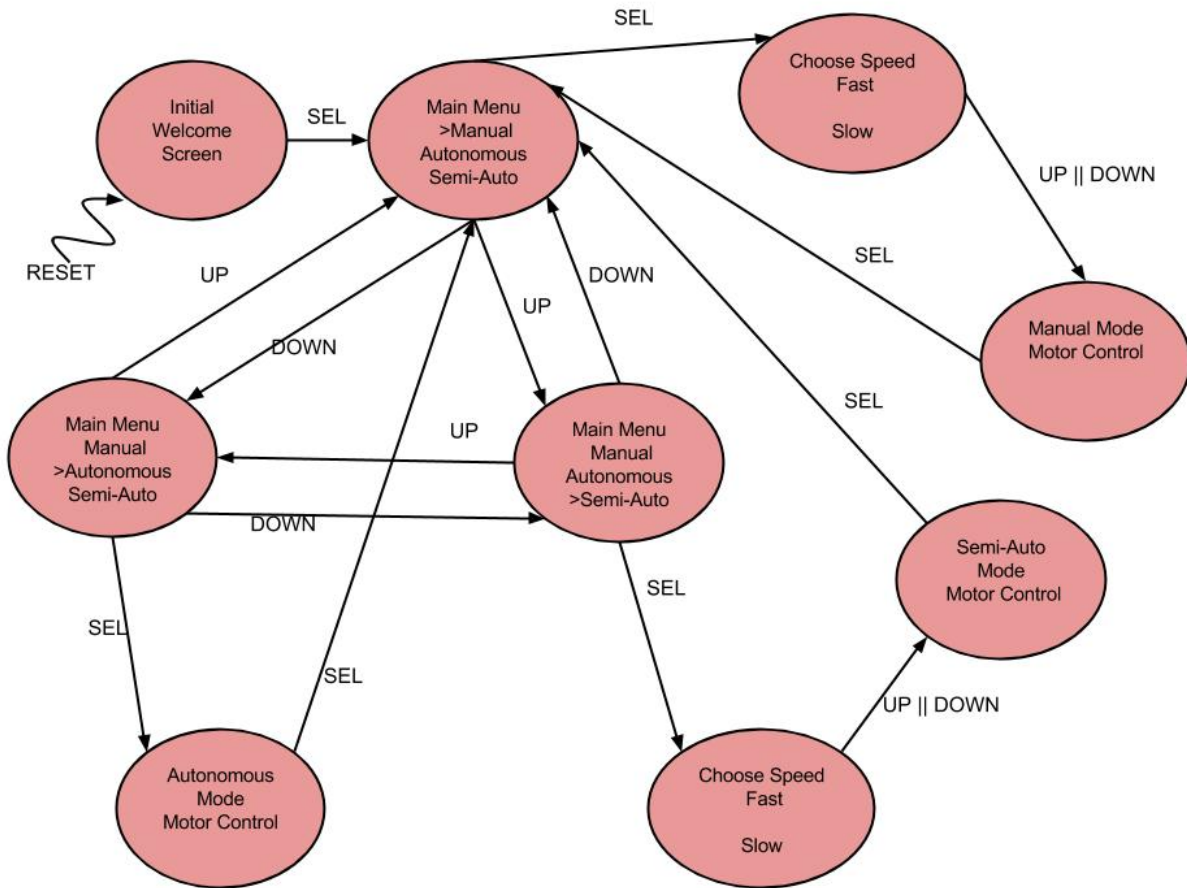
The flow diagram shows the flow of the code throughout the RoboTank System. The system begins with a simple menu screen welcoming the user asking it to press the select key to begin.

Once the select key is pressed, the user is prompted with three modes of operation: Manual, Autonomous, and Semi-Autonomous mode. The user can cycle through the options by pressing the up and down arrow keys on the keypad interface. To choose an option, the user must press select. If manual or semi-autonomous mode are selected, the user may choose a speed option for the tank. The two options are fast and slow and the selection can be made using the up and down arrows.

If the autonomous mode is selected, the tank will begin to move automatically depending on the distance sensor inputs. If the manual mode is selected, the tank will begin to move depending on what the user inputs into the system. The user can either use the keypad interface or send inputs through Bluetooth to control the tank. The tank can move forward, backwards, right, left, up-right, up-left, back-right, and back-left. If semi-autonomous is selected, the user can still control the tank until the one of the distance sensors reads that the tank is about to hit another object. When the tank knows it is about to hit something, the tank will automatically move out of the dangerous situation.

While the tank is in one of these modes of operation, the OLED will display the distance values of the four sensors in millimeters. The tank will also begin to create a sound through the speaker when the tank is too close to an object.

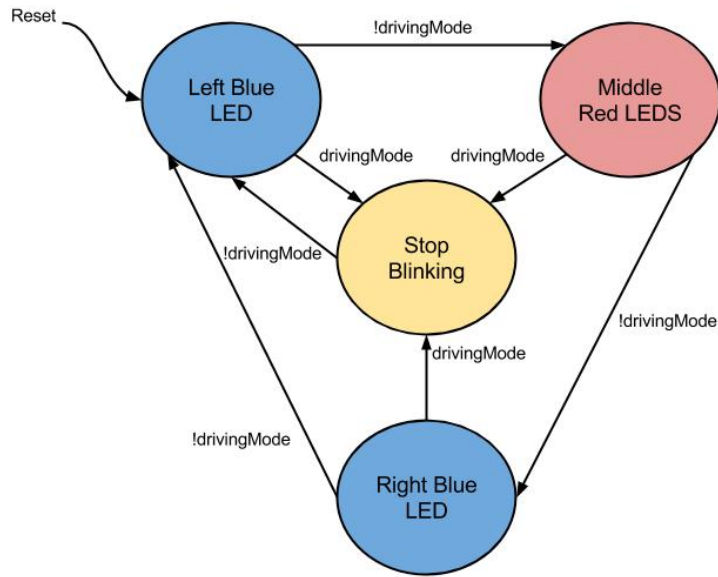
The user may press select to get out of any of these modes and return to back to the menu where the user can decide on the mode of operation. It is important to note that the tank is not operational once autonomous, manual, or semi-autonomous is initiated. This way the key presses control the menu, not the tank, and vice-versa.



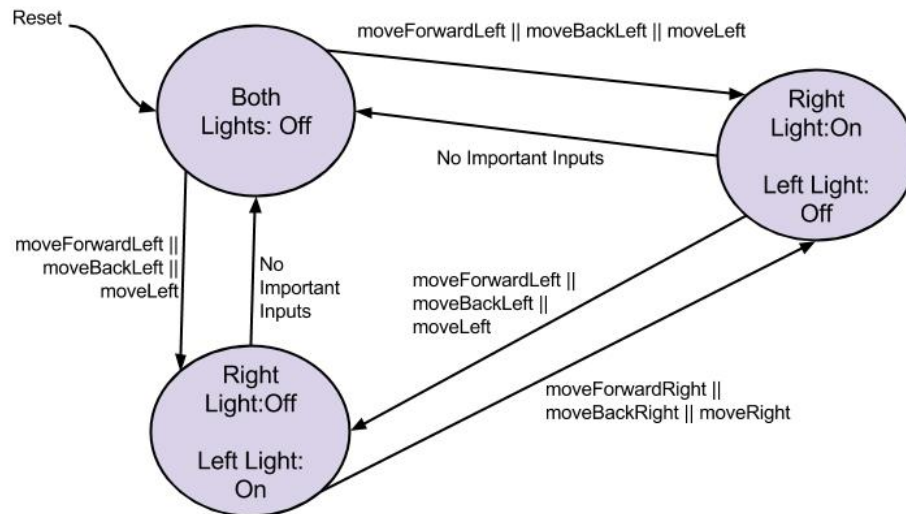
EXTRA FLOW DIAGRAMS

These extra diagrams corresponds to the extra lights we have on the tank and the different types of functions we have for the LEDs. The first flow diagram is on the menu lights that occur when we are selecting through the menu. There are three LEDs taking turns being on and this occurs until we reach one of the driving modes in the tank system. The second flow diagram corresponds to the turning lights on the tank. These LEDs are the same ones from the menu lights, but they have are functioning differently. If the tank is moving left or up-left, the left LED will turn on to indicate the tank is turning left. If the tank is moving right or up-right, the right LED will turn on to indicate the tank is turning right. The last flow diagram cooresponds to the brake lights. The brake lights turn on when the tank is in driving mode, but the tank has no input that commands it to move. When the tank is moving the brake lights should be on.

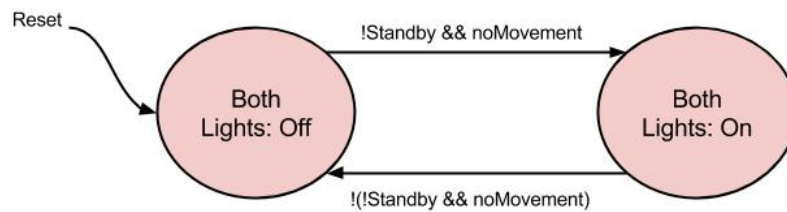
Menu Lights



Turning Lights



Brake Lights



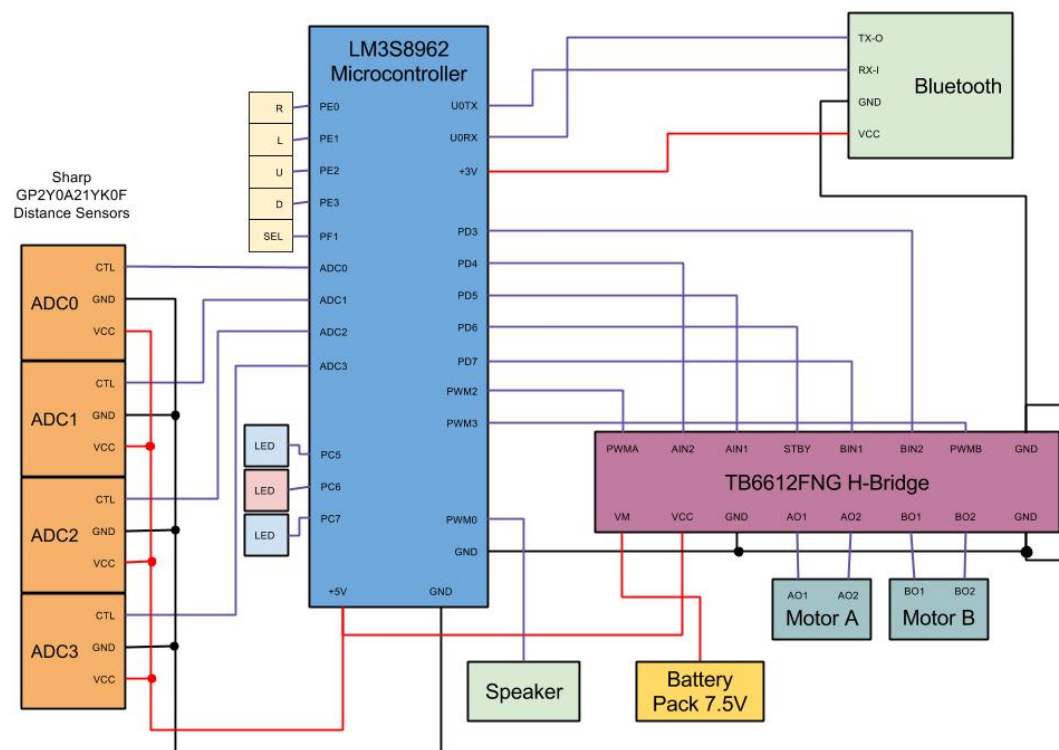
HARDWARE SCHEMATIC

In lab 5, we kept most of the peripherals that were used in lab 4 with an addition of a couple other new things added to the microcontroller. Originally we had buttons to control the menu and the motor, distance sensors, speaker, and an H-Bridge wired straight from the microcontroller to the two motors on the tank. We added a Bluetooth and a couple LEDs to our microcontroller and the H-Bridge is also getting 7.5 V from the battery pack.

The Bluetooth is powered by the 3.3 V output from the microcontroller. The Bluetooth talks to the microcontroller by connecting the TX-O to the U0RX port and the RX-I to the U0TX port. The Bluetooth is required to pair with another machine before there is any communication between the two pieces of hardware.

The LEDs we added to the tank is turned on from the GPIO in port C. They are given 3.3 V when the microcontroller has a logic output of 1 and 0 V when the logic output is 0. There are two functions for these lights. When the tank is in the menu and not in one of the driving modes, the lights go through a constant pattern from left to right. When the tank is in a driving mode, the lights will represent the brake lights and the turning lights.

The H-Bridge was powered by the 5 V from the microcontroller in the previous lab, but is being powered by a 7.5 V battery pack in lab 5. Since the microcontroller will also run off of a 5 V battery pack, there will not be enough current to power both the motor and the microcontroller from this one power supply.



MEMORY DIAGRAM

This memory diagram shows how all of the functions relate to each other. It displays the hierarchy Top-Down to explain how and when a function would be called. The global variables are also displayed and this shows how the functions interact with the global variables to get/set their values.



TIME SPENT

- Design: 5 hours
- Coding: 20 hours
- Test/Debug: 10 hours
- Documentation: 5 hours

Total: 40 hours

Our team members spent an equal amount of time doing equal amount of work. See README for work distribution.

CONCLUSION

In conclusion, this lab was very valuable to learn about Bluetooth connectivity as an added peripheral. We learned a lot about how the UART protocol works in order to serially send data via Bluetooth. Putting together the tank on the hardware and software side was very time

consuming but we learned a lot about how to create a complicated system by using many smaller sub-systems to achieve our goals. Our final product was a fully autonomous tank that avoided all collisions in its path. In addition, we were able to control the tank through Bluetooth using an app called Blueterm. Overall, we had a lot of fun creating this tank and learned a lot about Embedded System. Please see this small video that showcases the tank's functionality:

<https://www.youtube.com/watch?v=dcReAuvuEgM&feature=youtu.be>