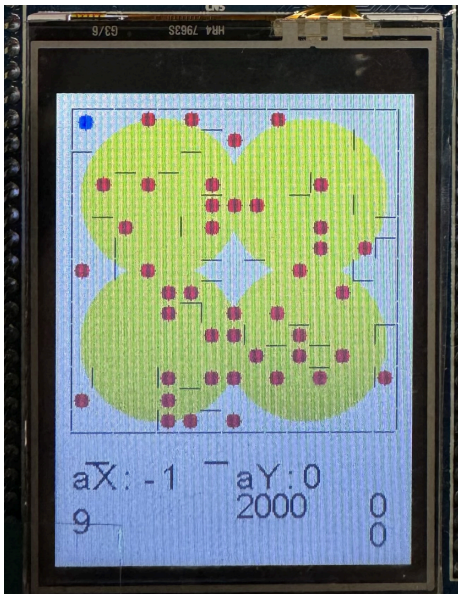
Jonathan Tang

Real Time Operating System - Final Project Report



Project Management summary and learning

Throughout the project, I was able to initialize a maze and a marble. Walls, holes, and waypoints are generated in maze initialization. The marble is able to move according to the movement of the gyro. Furthermore, the green and LEDS displays the status of the energy remaining. The energy is consumed by the disruptor, and the disruptor can be triggered by the button interrupts. The physics task handles gyro and marble movement, energy consumption and disruptor conditions, time remaining and collision checking. Winning and losing conditions are also implemented. Most of the functions meet the requirement from the client.

	Expected hours	Hours consumed	Status	Comment
Create a task diagram	1	1	completed	The task diagram displays the tasks of the program and the variables that are shared amongst the tasks.
Create the maze and holes	5	8	completed	Implemented a struct of maze, walls, and holes, and waypoints.
Create a marble that is control by the physics task	5	5	completed	The marble moves according to to the properly
Gyro task to control marble movement	5	5	completed	The marble moves according to to the properly
Button task to enable physics	5	3	completed	The button enables disruptor, and the ball can move through walls and holes.

disrupter.				
Physics task that controls of velocity, energy, recharging rate	8	10	completed	The physics task correctly handles the gyro movement, energy and disruptor.
Debugging, unit testing, etc	1	2	completed	Developed some screen displays for testing purposes

Unit tests:

Cutting points:

1. Ball movement checks: The gyro, button, physics, all affects the movement behavior of the drone. Therefore, we want to add unit tests for each task specifically to check whether the task interacts with the drone properly.
    - In progress: Checked that the gyro is correctly collecting values. Later have to test the ball movement after implementing the physics

2. Energy consume and recharge task: The energy is constantly consumed and recharged. Careful checks for energy behavior is essential.
    - Still pending. Will start implementing energy once maze and ball is configured.

Functional tests:

1. Maze display test: The LCD displays the mazes and holes correctly.
2. Maze randomization test: Different mazes are displayed throughout tests.
3. Button test: The interrupt is correctly configured and the button works.
4. Gyro test: The gyro is correctly configured and the button works.
5. LCD Timer check test: The timer that refreshes LCD runs and updates the LCD periodically.
6. Energy mutex test: Energy data is initialized properly and the mutex protects the data.
7. Ball mutex test: ball data is initialized properly and the mutex protects the data.
8. Physics test: The gyro and buttons correctly interact with the ball with the correct physics.

Summary statement:

- In this week I have made a task plan, task diagram, created cutting points and also made a risk assessment table. I have not implemented any code yet.

- In week 2, I got some more clarifications on the project. Planned more functional tests and had a better vision on the project. I have not implemented any code yet.

- In week 3, I started working on the maze configuration and ball movement. For ball movement, the gyro driver is modified so it takes both x and y axis readings. Next step is to implement the physics to allow the ball to move. FOr maze configuration, I have made the idea of implementing a function to convert sizes to the LCD display. Will then work on configuring walls, holes and waypoints.

- In week 4, I mainly continued on the maze configuration. I spent time understanding the requirements of the maze, and how to generate the walls. I have created a struct for the maze cells, walls, and holes. Also each cell had 4 pointers that connected the adjacent walls to keep track of the wall generation.

- In week 5, I finished everything remaining perfectly. Most of the time was spent on collision checking, button disruptor, timer, energy conditions, waypoints and LEDs.

Summary effort:

- In week 1, I worked 2 hours on a task plan, task diagram, cutting points and a risk assessment table. Therefore I completed 4.5% (2/44) of the project. Expected hours for these completed tasks were 2 hours (2/2), which is 1.00x the rate I expected.

- In week 2, I worked 1 hour on functional test planning. Therefore I completed 6.8% (3/44) of the project. Expected hours for these completed tasks were 3 hours (1/3), which is 0.33x the rate I expected.

- In week 3, I worked 3 hours on the Gyro task to control ball movement, 1 hour on maze configuration. Therefore I completed 16% (7/44) of the project. Expected hours for these completed tasks were 6 hours (4/6), which is 0.67x the rate I expected.

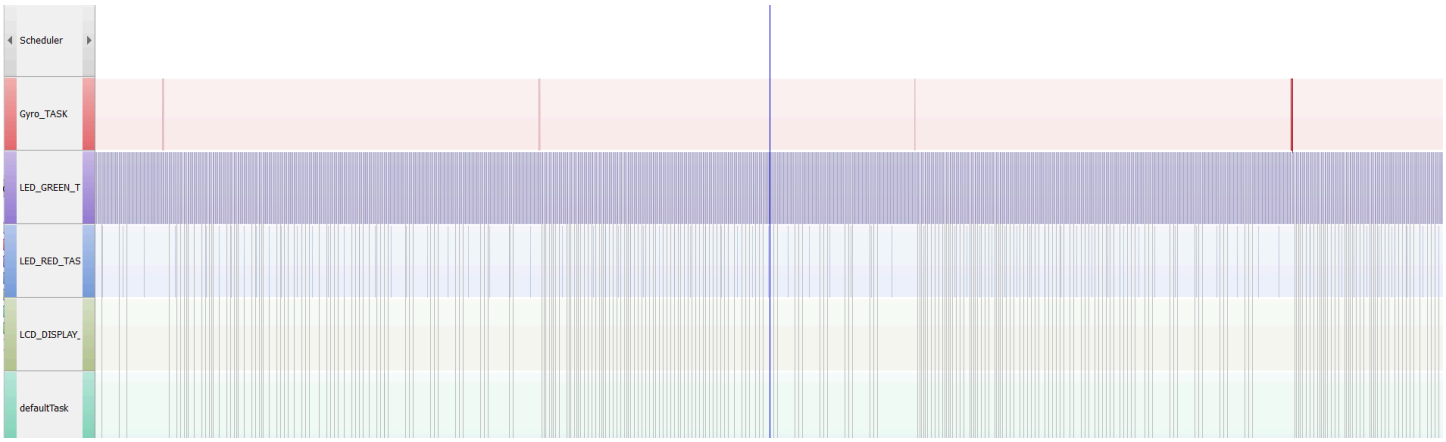
- In week 4, I worked 2 hours on the maze configuration, 1 hour on marble initialization. Therefore I completed 22.7% (10/44) of the project. Expected hours for these completed tasks were 6 hours (3/6), which is 0.5x the rate I expected.

- In week 5, I worked 2 hours on creating a struct for maze, 6 hours on gyro movement, 1 hour on testing of the marble movement. Therefore I completed 43.2% (19/44) of the project. Expected hours for these completed tasks were 8 hours (9/8), which is 1.125x the rate I expected.

- In week 6, 7 hours was spent on collision checking, 2 hours button disruptor, 1 hour on timer, 2 hours on energy conditions, 1 hour on waypoints and 2 hours LEDs.. Therefore I consumed 77.3% (34/44) of the expected project hours. Expected hours for these completed tasks were 12 hours (15/12), which is 1.25x the rate I expected.

Jonathan Tang

Analysis of solutions:



RT Tasks:
The SystemView figure looked very messy because my red and green LED tasks are constantly polling while my gyro (20ms) and LCD displays tasks (100ms) are called by periodic timers. All tasks have pretty short execution time and have lots of time in IDLE. deadlines are also met and tasks are cooperating fine. Green and red LED tasks are polling because it was the simplest way. If I have to improve it, I would it to be event flag or timer driven to save energy from constantly running and polling.

Physics Update:
Physics tasks are all handled in the gyro task. Therefore I only needed 1 mutex for the task to keep all drone and maze data safe. I gyro movement is calculated to check if it hits the border. Also collision checking is done. The logic of collision checking is, before drawing the drone on the LCD, if the drone overwrites a pixel that is black, then there is a collison. Same concept for collision of holes and waypoints. This method was perfect and was fairly simple. Also energy calculation, disruptor, waypoints checking are done in the task. The disadvantage of this method is that different functionalities are not separated in different tasks and could be hard to read and debug.

Data Configuration:
In the config files, about 80% of the variables were useful, and about half of them were flexible. Due to efficiency of maze to LCD conversions, the cell size and maze size could not be changed. Some simple variables like minimum activation energy, time remaining, and drone size can be changed easily. If I have another 2 weeks, I would improve my task structures and make variables more flexible in config. These could make the project more energy efficient, and more config generic.