Work Completed:

During this week, only few things were accomplished. Primarily, I used the commands for the pulse width modulation to make the output of the microcontroller decrease linearly, rather than dropping from high to low as soon as the random time is complete. In order to do this, I used the concept of duty cycle. Duty cycle represents the percentage of time period that the output is high. For example, say that I define the period to be 2 milliseconds and the duty cycle to be 80%, which means that the output will be high 80% of the period and low for the 20% of the period. This will make the motor run at 80% of the maximum speed, which will be 32 RPM for our motor.

![PWM Subroutine](image)

Figure 1: This figure shows the PWM signals of various duty cycle.

**PWM Subroutine**

- 100% duty cycle
  - Run for a random period of time between 2 to 4 seconds.
- 80% duty cycle
  - Run for .2 seconds
- 60% duty cycle
  - Run for .2 seconds
- 40% duty cycle
  - Run for .2 seconds
- 20% duty cycle
  - Run for .2 seconds
- 0% duty cycle
  - Until an input is detected from sensor or the function module
By using this technique the motor can run between 2.8 seconds to 4.8 seconds, rather than the original values of 2 and 4 seconds.

After completing this part of the program, I started the debugging process using the debugger installed with MPLAB (the program used to install the program onto the chip). I was able to debug the random time period subroutine and verified that it worked correctly. However, there were some conflicts with the SP03 subroutine that needs to be resolved in order to further continue the debugging process. This will be completed by the end of this week.

Also, before we left for break we decided to buy a new motor that will decrease the efforts required to create a coupling system for the motor shaft and the shaft connected to the wheel. However, after looking at our budget and the requirements for the motor, I believe that it will be better for us to use the motor we already off. Currently, we have a budget of $200 with which we need to order the PCB for our circuit, pay the people at the machine shop for their tools and help and also by a battery holder. In terms of the motor requirements, the current motor requires a .5 milliamps of current with 12 volts to run at a speed of 40 RPM and provide 2000 g-cm torque. However, the motors that I found online require more than 1 milliamp, which will require more batteries. Also, if we buy motor with a shaft in the middle, we will risk the torque or speed aspect of the motor. So, it is better for us to use the motor we already have, even though it requires more work.

**Future Work:**

By the end of this week, it is crucial to complete the whole program with the inputs and outputs working correctly. Also test the SP03, function module and the motion sensor with the microcontroller to test if it will work. Also, construct the motor circuit so it can be tested soon and give us the opportunity to order the PCB board. The PCB board will be divided into two parts. One part will have the circuit for the motor and the order will have the circuit for the audio control for the speaker.

**Progress Report:**

Currently, we are behind schedule, but hopefully will catch by the end of this week. The program should have been done by the end of last week, but unfortunately it was not done. In terms of the base of the game and wheel, the work will began this week and will be completed within couple of weeks.

**Hours Worked:**

9 hours