Software Design Report

Group 5

Yongxuan Li, Michael Ru, Sachin Apiah, Kevin Huo

**Problem Break-down**

Functions

int initialize() //written by Yongxuan(bluetooth) and Sachin(menu)

* starts Bluetooth communication and also contains start selection menu and will return the player (Mario, Bowser, Peach) as an integer

char readBluetooth() //written by Yongxuan

* reads in a character from the Bluetooth signal and returns it

void convertInput(char input, int player, bool & canShoot) //written by Michael

* calls setMotor(*parameters*) and sets the motor values depending on the character accepted through the parameter

void setMotor(int leftMotor, int rightMotor) //written by Sachin

* sets the motor power to each of the motors controlling movement

void launchBall() //written by Sachin

* rotate the ball shooting motor to launch the ball

void checkColour(int & lap, bool & canShoot, int player) //written by Michael

* checks the colour sensor and will perform appropriate actions

void checkTouch(int player) //written by Kevin

* checks the touch sensor and perform appropriate actions

void outputResults() //written by Kevin

* uses the motor encoders and timers to calculate the time and speed and output to the NXT

**Software Plan and Testing**

The plan is to have a looping main function that will correspond to the majority of the program. The program will start with an initialization phase where bluetooth connectivity will start. In this phase, the user will also select a character using the NXT brick buttons. The character chosen will be saved to the brick as a variable. This phase can be tested by displaying a message to the screen when bluetooth connects and the character selection can be tested by using debug mode where the value of the character variable can be seen.

The next phase would consist of the main loop phase where the vehicle is actually moving. This phase would consist of reading in a bluetooth message sent from the computer (which got a signal from a Wii remote). This can be tested by outputting the received data to the NXT screen. The next part would be taking the message and mapping it to motor movement. This can be tested by giving a specific message and making sure that the motors are moving properly. After setting the motors, the brick will then check for input from the sensors. It would check from the touch sensors and the colour sensor. If a specific input is detected then the robot will act accordingly (e.g. a blue colour is detected and the robot would speed up, bump sensor is triggered and the robot would stop moving for a second). These can be tested by manually triggering the input and making sure that the appropriate action is taken. The colour sensor function would also return the current lap number by checking if a specific colour set to be the start/stop. A ball shooter function will also occur when the user has a shooter power-up and clicks the correct button. This can be tested by making sure that the shooter shoots only when both the conditions are satisfied. After three laps the main loop would stop and move onto the final phase.

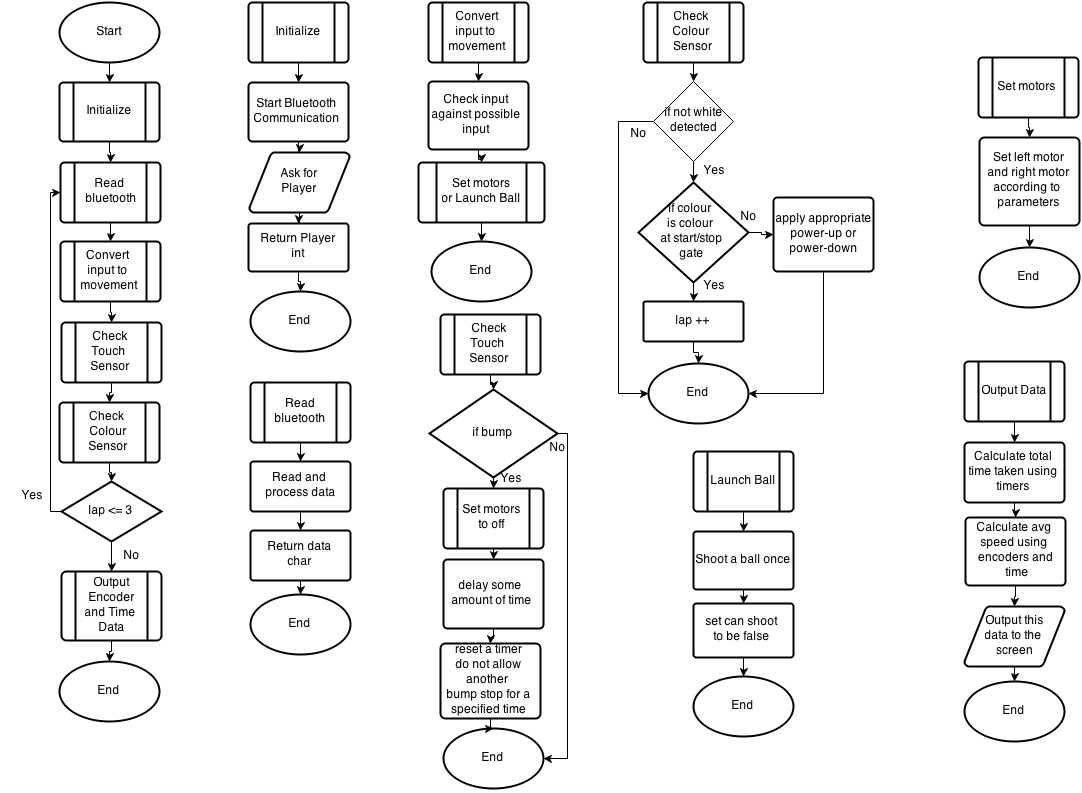
The final phase will consist of using timers and encoders which were reset that the beginning to output the time taken as well as an approximate average speed. This phase can be tested by running the robot along a track which has a known distance as well as using a stopwatch. Then some math could be done to calculate the actual data which can then be compared.

**Revision List**

Previous constraints included: the robot must move in a fluent manner; the program must provide an exceptional Mario-Kart/video game experience; and the player must interact with the robot. These constraints should be changed so that they provide hard limits to be measured against for a pass or fail. Fluidity can be changed to a criteria, as it is very hard to quantify. The Mario-Kart resemblance constraint can be broken into several different constraints: the program must contain elements of mario-kart gameplay, such as shell-shooting, inclusion of racetrack, power-ups, interaction with other karts, wii-remote controlled movement, and different characters. The user must interact with the lego NXT brick both directly (through button pushes) and indirectly (through wii-remote control).

The criteria of incorporating as many sensors as possible has been removed, as the goal is now to optimize the given space with the sensors that must be used, since it is unrealistic to plan for future software upgrades.

The mechanical design of the robot has been modified so that the free spinning wheel is now positioned at the back, and the two motor-controlled wheels are now at the front. This allows for pivotal turning to occur at the front of the car, making it easier for the user to make tight turns. The ball shooter has been repositioned to the back of the car pointing forward, to account for the limited space at the front of the vehicle reserved for the two sensors.



**Flow Chart**

Main