School of Information Technologies Faculty of Engineering & IT

PROJECT REPORT COVERSHEET - GROUP ASSESSMENT

Unit of Study: ELEC1601

Tutorial Time: 2020 S2 Monday 6:00PM

Group name: M18 Group 5

Assignment name: Group Project

DECLARATION

We the undersigned declare that we have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that failure to comply with the *Student Plagiarism: Coursework Policy and Procedure* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Project team members				
Student name	**Student ID**	**Participated**	**Agree to share**	**Signature**
Yaxi Yang	**470413559**	**Yes**	**Yes**	**Yaxi Yang**
Ayad Farhat	**490383120**	**Yes**	**Yes**	**Ayad Farhat**
Emily Xiong	**490613153**	**Yes**	**Yes**	**Emily Xiong**
Wenxi Zheng	**500061017**	**Yes**	**Yes**	**Wenxi Zheng**
Xingyi Yang	**500335859**	**Yes**	**Yes**	**Xingyi Yang**
Weiting Hung	**309338468**	**Yes**	**Yes**	**Weiting Hung**

Level 2, SIT Building, J12
The University of Sydney
NSW 2006 Australia
T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
ABN 15 211 513 464
CRICOS 00026A

1. INTRODUCTION

The purpose of the project was to design a robot which could be both manually controlled by implementing a two way communication bluetooth system, and automatically navigate through a maze. The manual component involved connecting the master bluetooth to the joystick, which sends x and y values to the slave bluetooth on the robot, allowing the robot to be manually controlled. The automatic component utilises control flow logic, with a pair of infrared sensors and ultrasound sensor, to smoothly navigate through a maze.

2. PROJECT PROCESSES

2.1 Project Organisation

Our project was divided into two separate parts: manual control and maze navigation. We have planned out our project schedule on a weekly basis. In week 9 lab, We started coding and circuit connection to get each component working. In week 10 lab, our main goal was to achieve smooth navigation as well as improving joystick performance. In week 11 lab, we worked on bluetooth connection and improving navigation control flow.
Our main tools for project management were Zoom during labs and WeChat throughout the week. This allowed all members to update their progress on their assigned tasks, as well as collaborating and solving issues together as a team. At the end of every lab session, the source code was emailed to all team members. Furthermore, our progress was recorded on Trello, which allowed all team members to keep track of the project details.

2.2 Task Division

The manual part was divided into 4 main tasks: joystick, bluetooth connection, coding servo motor directions, manual control - joystick performance improvement. On the other hand the maze navigation was divided into 3 tasks: coding / circuit Connection - Infra-Red, automatic navigation - control flow logic improvement, movement function - adjust turning angles. On-campus students were involved in circuit connection and coding of both parts and were involved in the majority of the tasks , while online students were involved in the coding of the joystick performance improvement, servo motor directions, movement function - adjust turning angles and coding / circuit connection - joystick (manual).

2.3 Problems

When we worked on automatic navigation, one of the major issues was turning. Initially, we only had two pairs of infrared sensors, which we used for detecting the front and left. However, the right side was undefined, and had resulted in the robot having difficulty deciding which way to turn. Ultimately, we used an ultrasonic sensor to return the distance to the obstacle in front of the robot. And we used the infra-red sensors to return a boolean value indicating whether there was any obstacle on the left and right side respectively. This significantly improved its ability to navigate through the maze smoothly and effectively.
With the manual part of the task, the main issue was that the center position vector <X,Y> of the joystick in idle mode was always different and fluctuating. This gave us a non-stable center position for the vector representing the speed when we were converting the values of the x and y axis into the speed of the servo motor. To resolve this issue, we simply used a constant base value for rescaling and assumed it to be a reasonable approximation. However, as a consequence, the actual servo speed was always off from its expected mapping speed.

3. MATERIALS

The materials and components that we used for our project are:

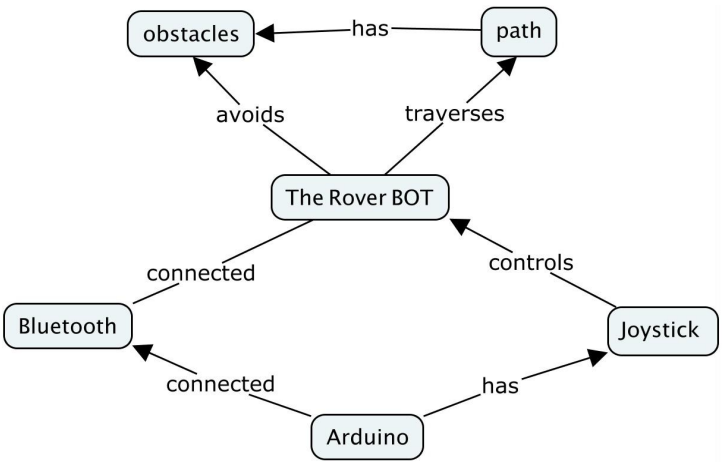
- 9-pin joystick x 1:
  The X and Y axes are two 10k potentiometers which control 2D movement by generating analog signals. When the module is in working mode, it will output two analog values, representing two directions. Commonly used in video games controllers.
  Operating Voltage: 5V/3.3V Compatible.
- IR LED and receiver pair x 2
  It is suitable for use in infrared remote control units with high power requirements, free air transmission systems, infrared source for optical counters and card readers.
- Rover bot x 1
  2 wheels smart robot car
- Ultrasonic sensor x 1
  Ultrasonic sensors use SONAR to determine the distance of an object like bats do.
  Operating voltage: 5V
- Bluetooth board x 2
  A Serial Bluetooth module. It can be easily used with Arduino/Seeedstudio for transparent wireless serial communication.
  Operating Voltage: 3.3V
- Resistors
- wires
- Batteries
- Buzzer x1
- Arduino breadboard set x 1

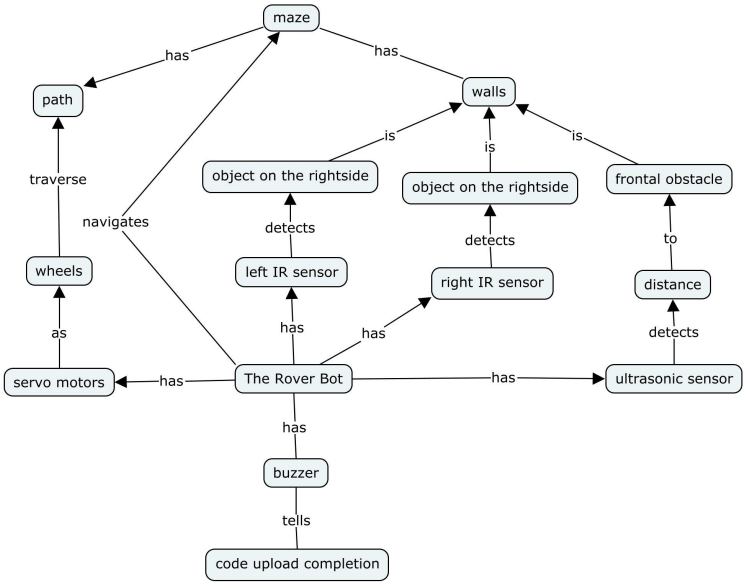
4. IMPLEMENTATION

Our group used separate code for the manual/automatic tasks. For the manual control part, we are using bluetooth so that the joystick connected to the Arduino board could control the robot remotely. For the maze navigation task, we are simply using the sensors with our code.

4.1 Knowledge Modelling - CMap

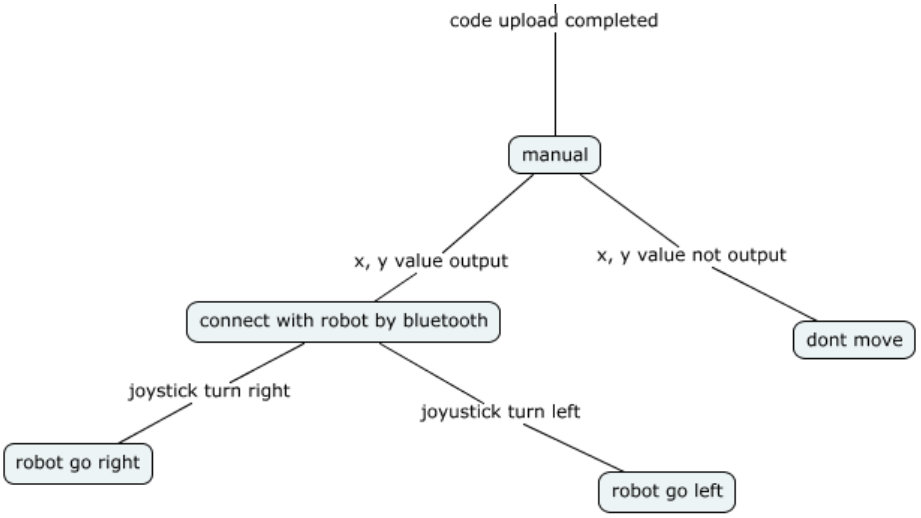
Manual (left) Auto (right)



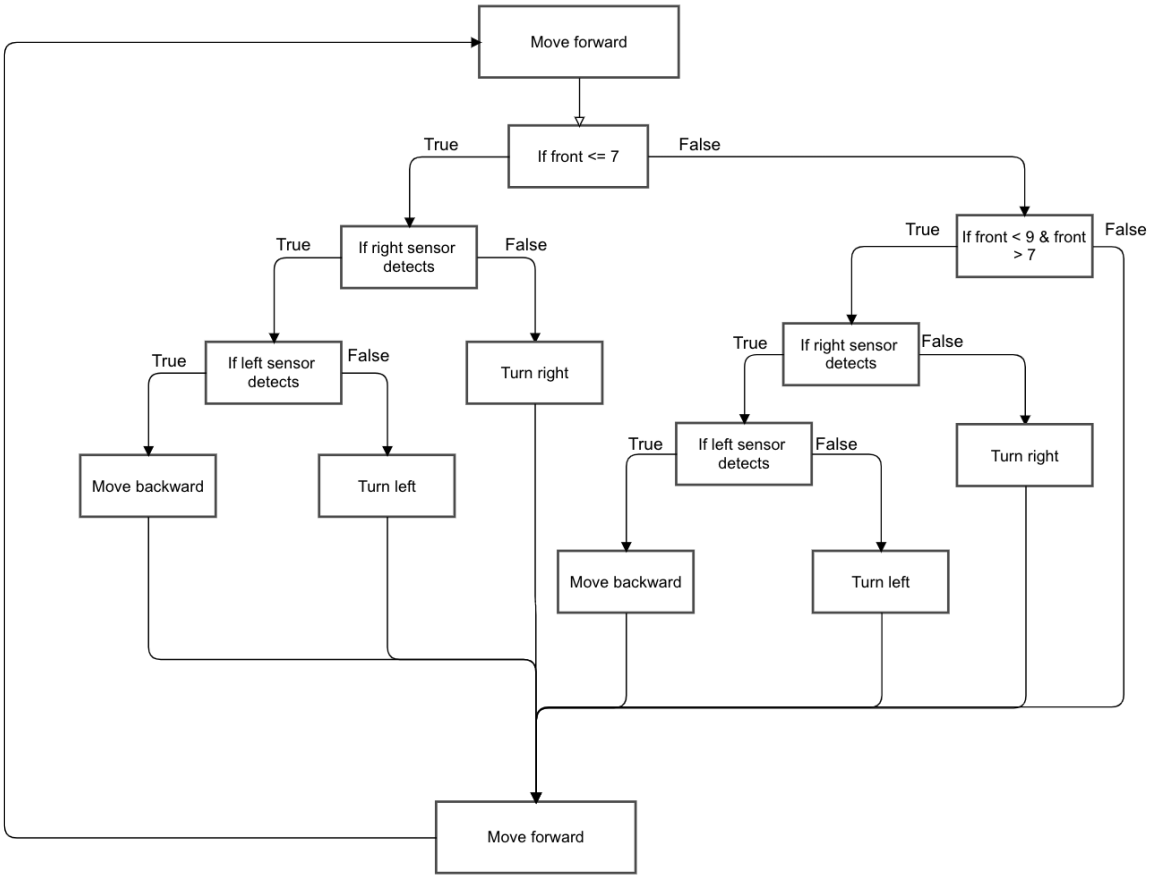


4.2 Flow Chart

4.2.1 Manual



4.2.2 Automatic



4.3 Pseudocode

4.3.1 Manual

[Remote control]
Declare bluetooth serial
Initialise variable for joystick and bluetooth
Program set up
    Set up joystick X,Y input pin
    Set up bluetooth master connection
        Send MasterMode through bluetooth connection
        Send MasterName through bluetooth connection
        Snd AutoConnection=False through bluetooth connection
        Flush bluetooth connection and wait
        Send Inquirable=True through bluetooth connection
        Flush bluetooth connection and wait
    Search bluetooth slave connection until found
        Read from bluetooth connection char by char
        Construct command and search SlaveName command
    Send connection command
    Read connection result until found
Auto-repeats main loop
    Read X,Y value from joystick
    Construct (X,Y,0) command
    Send (X,Y,0) command through bluetooth connection
    Wait until 100ms passed since last command sent

[Rover bot]
    Declare left servo
    Declare right servo
    Declare bluetooth serial
    Initialise variable for servo and bluetooth
    Program set up
        Set up bluetooth slave connection
            Send SlaveMode through bluetooth connection
            Send SlaveName through bluetooth connection
            Snd AutoConnection=False through bluetooth connection
            Send AutoConnection=True through bluetooth connection
            Flush bluetooth connection and wait
            Send Inquirable=True through bluetooth connection
            Flush bluetooth connection and wait
    Auto-repeats main loop
        Read from bluetooth connection until (X,Y,0) command is constructed
        Extract position vector <X,Y>
        If position vector is nearby center, stop both left and right servo motor
            Deattach left wheel from pin 13
            Deattach right wheel from pin 12
        Otherwise, use position vector to update left and right wheel speed
            Declare initial left wheel speed value
            Declare initial right wheel speed value

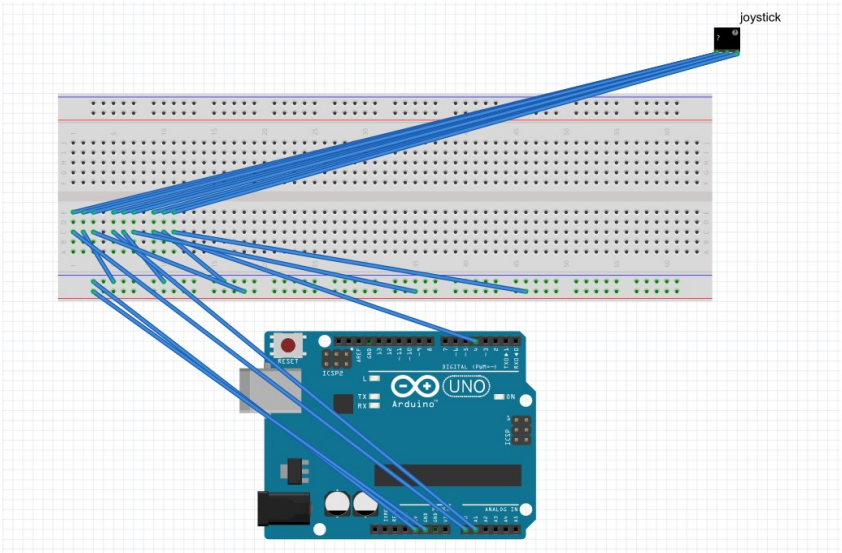
```
        Calculate radianz of position vector
        Rescale radianz to speed adjustment
        Calculate final speed value for left and right wheel
        Attach left wheel from pin 13
        Attach right wheel from pin 12
        Update left wheel speed
        Update right wheel speed
```

4.3.2 Automatic

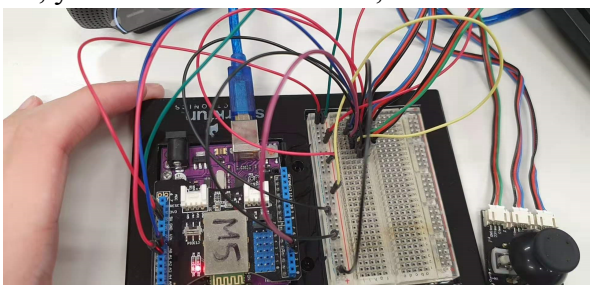
```
Declare left servo
Declare right servo
Initialise variable for frontal distance
Program set up
    Play buzzer for one second
    Attach left wheel to pin 13
    Attach right wheel to pin 12
    Set up left IR sensor
    Set up right IR sensor
    Set data rate to 9600 bps
Auto-repeats main loop
    Update distance to the front wall
    Check leftside for blockage
    Check rightside for blockage
    If frontal distance is less or equal to 7cm
        If left is blocked by wall and right is clear
            Move back for 2ms
            Turn right for 4ms
        Else if right is blocked and left is clear
            Move back for 2ms
            Turn left for 4ms
        Else if both sides are blocked or both are clear
            Move back for 2ms
    Else if frontal distance is 7-9cm
        Stop movement for 5ms
        If left is blocked by wall and right is clear
            Turn right for 6ms
            Stop for 1ms
        Otherwise just turn left for 2ms and stop for 1ms
    Otherwise when frontal distance is greater than 9cm
    Keep moving forward
    Delay for 0.1ms
Function to set up ultrasonic sensor
    Set pin 6 as output
    Set pin 5 as input
    Measure distance
    Return a distance in centimeter
```

4.4 Prototype (circuit diagrams)

4.4.1 Manual

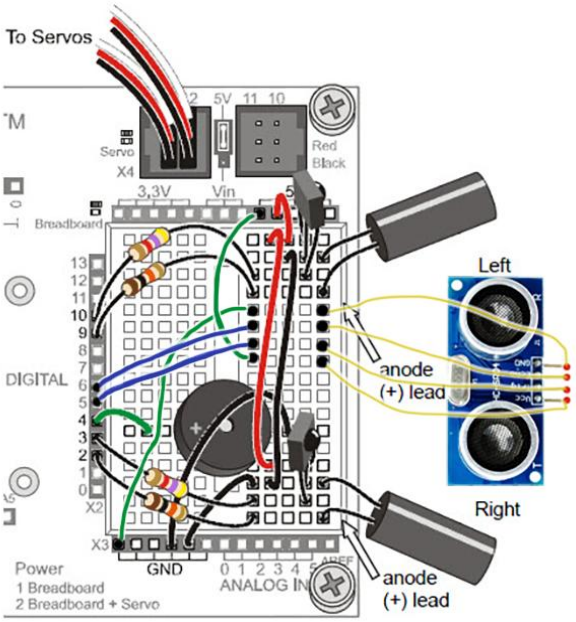


The above circuit diagram demonstrates our joystick connection to the Arduino breadboard. The joystick is connected to the Arduino board and it controls the robot's movement through bluetooth. There are 9 pins connected to the joystick. The e vcc pin is connected to the 5v, the GRD pin is connected to the ground; x axis is connected to A0, y axis is connected to A1, and z axis is connected to D4.

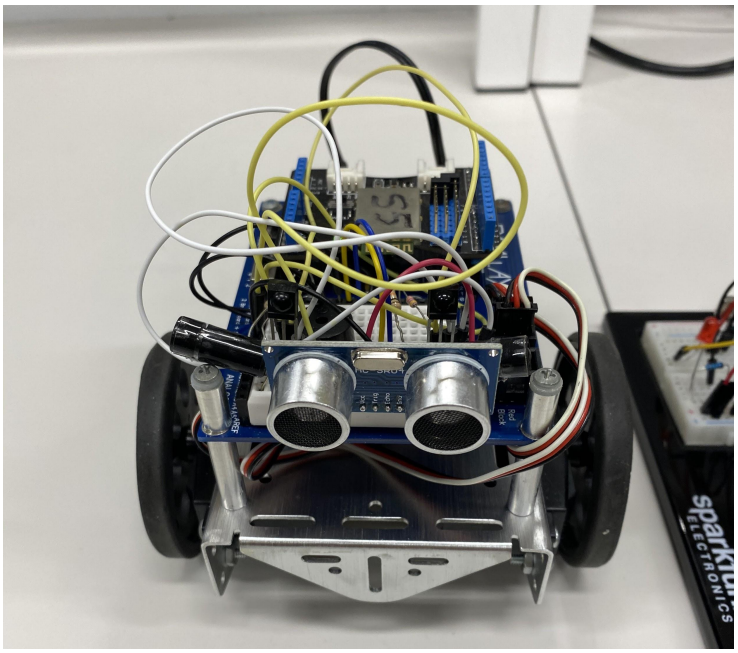


4.4.2 Automatic

For this part, we used the same circuit base as the one we learned in our week 7 lab [1]. Instead of the resistor they provided, we switched to two pairs of 4.7k and 10k resistors to reduce the sensitivity of the IR sensors, so that they detect a shorter distance. The buzzer rings for one second to indicate that our code has finished uploading. One IR sensor faces the left side to detect whether the path is blocked by the wall on the left side, and the other IR sensor does the same for the right side. On top of this model, we also use an ultrasonic sensor to measure the distance to the front end.



Here is a picture of our final robot:



5. CONCLUSION

5.1 Project Summary

Overall, our group was organised in a goal oriented manner. Due to our clearly defined weekly goals and reasonable task division, we were able to make significant progress each week. Within our limited given time of 3 weeks, we successfully put together a project that was able to finish its desired tasks, with an impressive task completion speed and minimal errors.

5.2 Improvement

For the manual part, we could synchronise the center vector position between master and slave, so that Rover bot can support the dynamic value to rescale and calculate servo speed. It could also alert if the center vector position is beyond the threshold from the theoretical value, since this will result in uneven sensitivity to control the servo motor.
For the maze navigation task, there were always a few bumps on the wall before we reached the end point. This was because while the robot was traversing the path automatically, we did not have any way to ensure that the robot was in the middle of the track. If we were given more time to improve this flaw, we could have replaced the IR sensors with a pair of ultrasonic sensors, so that the distance to both sides of the wall can also be measured. This can allow us to always position the robot in the middle of the path, and therefore avoid bumping into the wall.

6. BIBLIOGRAPHY

1. https://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino/chapter-7-navigating-infrared-10
2. https://canvas.sydney.edu.au/courses/25935/pages/11-dot-3-4-robot-vision
3. https://canvas.sydney.edu.au/courses/25935/pages/servo-motor-and-boe-shield-bot-manual-navigation
4. https://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino/chapter-4-boe-shield-bot-17
5. https://www.seeedstudio.com/Bluetooth-Shield-V2-p-2416.html
6. https://canvas.sydney.edu.au/courses/25935/pages/11-dot-2-22-serial-communication-via-bluetooth
7. https://canvas.sydney.edu.au/courses/25935/pages/11-dot-2-23-data-transfer-via-blueto oth

7. APPENDICES

Source code: https://github.sydney.edu.au/yyan6117/2020_S2_ELEC1601_M18G5
Trello: https://trello.com/b/MdOngLcJ/2020elec1601m1805