Zumo Self Reflection Report

## Benjamin Tyzack 27008109

## Introduction

The aim of this assignment was to create a search and rescue robot using the Pololu 32u4 Zumo Robot and the Arduino IDE. The Zumo is controlled wirelessly by using XBee modules which connect over a serial port and a GUI was created to control the Zumo through.

## The tasks

There were 7 tasks in total to be completed, I was able to complete 6 out of those 7. The tasks are listed below

1. Task 1 - The Zumo can be driven down the corridor using wasd keys through a GUI
2. Task 2 - The Zumo can drive down the corridor wirelessly using line sensors to keep within the corridor lines, the Zumo should stop when it reaches a wall
3. Task 3 - When the Zumo reaches a wall, control should be handed back to the controller who indicates which way the Zumo should turn, once the Zumo has finished turning the controller presses ‘C’ to resume Zumo auto mode
4. Task 4 - When the Zumo gets to a wall it should send a message over the XBee asking which way to turn, the controller will indicate right or left and the Zumo will turn 90 degrees then continue on it’s journey with no further user input
5. Task 5 - When the Zumo gets to a room, the controller should stop the Zumo outside the room and indicate which side the room is on, the Zumo will then complete a room search using the proximity sensors and will return a message over the XBee about whether a person is found or not
6. Task 6 – When the Zumo gets to the T Junction the controller should indicate whether to turn left or right, the Zumo should then go down that side and once it reaches the end of the corridor it will prompt the controller to start a 180 degree turn, the Zumo should then go down the corridor to the other side of the T Junction to search the rest of the corridor
7. Task 7 – The return journey, after reaching the second end of the corridor the Zumo will prompt the controller to start the return home journey, in this journey the Zumo will take the fastest route back to the start only revisiting the rooms in which people were found, if the room is still occupied then the Zumo’s LEDs should blink and should play a tone which acts like a ‘follow me’ guide which only stops once it gets back to the start

## Completed Tasks

I was able to complete tasks 1-6 out of the 7 listed above.

## Task 1 – Manual Control

I programmed the Zumo to be controlled manually by using key inputs from the keyboard. The table below indicates which keys are used and what their uses are:

|  |  |
| --- | --- |
| Key Input | Action |
| ‘w’ ‘a’ ‘s’ ‘d’ | These keys are used to control the movement of the Zumo, ‘w’ moves it forward, ‘s’ moves it backwards, ‘a’ turns it slightly to the left and ‘d’ turns it slightly to the right |
| ‘b’ | This indicates that there is a room to be searched, it also starts the logging room process where the user presses either ‘a’ or ‘d’ to indicate whether the room is on the left or right |
| ‘c’ | Once the room has been logged, the user then presses ‘c’ to start the room search. If the Zumo finds a person then it will sound the buzzer and send a message over the XBee indicating that fact, once the room search is complete the Zumo will turn back into the corridor and wait to continue |
| ‘k’ | This is for emergency stop. The Zumo will stop immediately and wait for further instructions. |
| ‘q’ ‘e’ | These are used to rotate the Zumo 90 degrees right and left, the Zumo encoders are used to make sure the turn is accurate |
| ‘u’ | When the Zumo gets to the end of the corridor pressing ‘u’ will do a 180 degree turn |
| ‘m’ | Changes to automatic mode |

## Task 2 – Autonomous Mode

For this task I used the line sensors to control the Zumo to keep within the corridor walls. To find out how to do this I looked at the Zumo examples Line Solver and Maze Follower in the Zumo32u4 library. In the calibration function that runs at the start of the program, the 3 line sensors are calibrated by scanning the floor to get a base value. When moving through the corridor if one of the left or right sensor’s value goes over that base value e.g. detects the black line then it will turn away from it.

If either the left or right sensor hits the black line AND the middle one does too then this signifies to the Zumo that it had reached a dead end, this could either be a corner, the T Junction or the end of the corridor and it will send a message over the XBee which way to turn.

## Task 3 – Turning a corner 90 Degrees

Once the Zumo reaches a corner it will return to manual mode, at this point the user can then press either ‘q’ or ‘e’ to turn the Zumo 90 degrees in the desired direction. Once this is complete the user can then press ‘m’ to return back to auto mode.

## Task 4 – Turning a corner autonomously

To complete this task I created two functions a turn left 90 and a turn right 90, in these functions I use the Zumo encoders which measure how far the motors have turned. I found the value that the encoder reaches at 90 degrees through trial and error and the value I ended up using is -700 which is around 90 degrees for both the right and left turn.

When the Zumo reaches a corner, it will ask whether to turn right or left and will then perform that task. Once the turn is complete the Zumo will continue its journey down the corridor with no further user input needed.

## Task 5 – Search a room

When the Zumo is in manual mode, the user presses ‘b’ to signify that there is a room and the Zumo will message over the XBee which direction the room is in, when the room has been logged the Zumo will turn in that direction and wait for the user to press ‘c’ to start the search process.

When the Zumo searches a room it turns from left to right while firing off the proximity sensors, if the value of the proximity sensors returns higher then a certain value then a person has been found and the Zumo will message that over the XBee. The message will also say which room the person was found in. If no person is found in that room then that will be messaged over the XBee. Once the Zumo has finished the room search it will leave the room and return to the corridor and wait for further instructions.

In auto mode, the user will still press ‘b’ to signify that there is a room and will log the room the same way as in manual by pressing either ‘a’ or ‘d’ for which direction the room is in, but once the room is logged the Zumo will automatically start the room search with no user input needed. Once the search is complete the Zumo will leave the room and continue down the corridor.

## Task 6 – T Junction

When the Zumo makes it to the T Junction it will ask which way to turn, the user will press either ‘a’ or ‘d’ to indicate left or right. The Zumo will then turn down that side and go down the corridor. Once it reaches the first end, the end count will equal to 3 which allows the Zumo to know that it needs to perform a 180 degree turn so it will prompt the user to press ‘u’ to perform that action. When the has turned around it will go down the corridor and until the user presses ‘p’ the Zumo will not be able to do any room searching. The user presses ‘p’ when the Zumo has reached the T Junction again to re-enable room searching for the second part of the corridor. Once the Zumo has reached the second end of the corridor it will know that it has reached the end of its journey and stop.

## Issues I faced during development

## Proximity Sensors finding objects

When I first tried finding objects using the proximity sensors I was having a lot of issues with it finding objects when there was nothing close by. I found out that after a while my program was looking for a sensor reading that was too low which meant that even if there was an object further away it would detect that object and say it was a person. I found changing the value to around 3-4 made it so it would only detect objects closer to the Zumo which is much better.

## Turning Corners 90 degrees accurately

When I first coded the turning a corner I used delays to turn corners which weren’t very efficient and wouldn’t always be the best method. I then started using the encoders built into the Zumo to measure the distance the motors had travelled to get a more accurate 90 degree turn. This took quite a while to figure out as I initially had a lot of difficulty finding the right value for the encoders for a 90 degree turn. I found out that the encoder values could be a negative number and I had been looking for positive numbers in my trail and error process. I found the value -700 to be the closest value to giving me a 90 degree turn and even though it’s not fully accurate it’s the closest I managed to get.

## Creating a GUI through python and pygame

I decided to create a GUI in python as out of the list of languages that was provided as examples it was the language I knew more about and thought it would be easier to make a GUI in it then the other examples. I ended up using pygame to create my GUI. However I found that creating the GUI in python is a lot more tasking then originally thought as everything in the GUI needs to be drawn out then shown on the GUI and there was just a lot of code needed for such a simple GUI. I originally planned to have buttons on the GUI which can allow the user to control the Zumo from but I ended up with just a simple GUI with the instructions listed and a terminal line which shows any message received from the Zumo. I used threading within my GUI as it allowed me to control the Zumo and for messages to be sent over the XBee simultaneously.

## Sources

The application is developed using Pololu Zumo32u4 libraries which includes:

1. Zumo32u4Motors – Allowed control of the motors
2. Zumo32u4Buzzer – Used to make buzzer sounds when a person is found
3. Zumo32u4ProximitySensors – Detects objects
4. Zumo32u4LineSensors – used to detect black lines and navigate the Zumo around the corridor
5. Zumo32u4Encoders – Used to make turns more accurate

I also used examples from the Zumo32u4 library called LineSolver, MazeFollower and Encoders as a guide to learn how the different sensors and encoders worked.

Stackoverflow.com was also used for help and guidance when I ran into problems that couldn’t solved by looking at the above examples. It was also used to help me solve issues with creating my GUI mainly with how to use threading.