Jelly Jinzhe Liu – AB20328

AB20328@bristol.ac.uk

Mian contribution: Hardware

A clear report showing my contribution to the group project

Group Project 3 2022

Individual report

Content

[Abstract 3](#_Toc132256551)

[Introduction 3](#_Toc132256552)

[Direction 4](#_Toc132256553)

[Hypothesis/ Feasibility 4](#_Toc132256554)

[Motor/actuator 4](#_Toc132256555)

[Processor 5](#_Toc132256556)

[Choosing Arduino 5](#_Toc132256557)

[Components 6](#_Toc132256558)

[GYRO 6](#_Toc132256559)

[Screen 6](#_Toc132256560)

[Mode/Function 6](#_Toc132256561)

[Selfcheck 7](#_Toc132256562)

[Manual mode 7](#_Toc132256563)

[Auto mode 7](#_Toc132256564)

[Design and connection 7](#_Toc132256565)

[Code 10](#_Toc132256566)

[Group Project.ino 10](#_Toc132256567)

[Config.h 11](#_Toc132256568)

[Gyro.ino 12](#_Toc132256569)

[Selfcheck.ino 13](#_Toc132256570)

[Switch\_value\_manual.ino 13](#_Toc132256571)

[Switch\_value\_mode.ino 14](#_Toc132256572)

[Conclusion 16](#_Toc132256573)

[Contribution 16](#_Toc132256574)

[Advantages and Improvements 16](#_Toc132256575)

[Advantages 16](#_Toc132256576)

[Improvements 17](#_Toc132256577)

[Reference 17](#_Toc132256578)

/

# Abstract

This report is based on a group project of selecting and configuring motor and processor, designing a circuit, writing code, and discussing its feasibility.

The part of this report on the group project is to discuss the feasibility of the motor and processor, using the result to choose the motor and integrated it into the maze. The processor needs to choose the model, design the circuits, and write the code. Also, when the image processing is finished, it must use the same port the processor provided.

# Introduction

|  |  |
| --- | --- |
| Table 1 User’s component in Arduino mega | |
| NAME | NUMBER |
| Button | 4 |
| Switch | 1 |
| LCD screen | 1 |
| GYRO | 1 |
| Full-colour LED | 1 |

In the hardware part, the main function is using the motor to drive two knobs in the maze, which can control the pitch and roll. This function is leading the ball running in the guided route in case of not dropping the hole and to the destination. *It uses two servo motors to drive the maze by connecting to Arduino Mega. when the system turns on, Arduino will do a self-check to make sure every component is connected and working well.* Table 1 *shows its component for users.*

*After self-check, the user can choose two modes: auto mode and manual mode. It also includes auto trim, connect to pc and degree in auto mode.*

*The benefit of this plan is to keep the original appearance and function of the maze, so people can still use it like the product itself.*

# Direction

Machines should always serve humans. A useful product not only has perfect hardware but also has good practicality. It is important to choose how to use it.

In our design, the first task or main function is the connection to the PC and the use of two servos to drive the maze. Every part should give a visualization to the user to let them know what the system is doing, and easy to check errors. One of the problems is that the user will place the maze on different planes, therefore, the angle of the maze will change concerning the horizontal plane. In our design, we want it can be auto trim, which is to design a system that the user does not need to trim and still can get horizon.

# Hypothesis/ Feasibility

## Motor/actuator

In this part, our main theme is trying to keep the product itself. Thus, we choose to use two knobs controlled by a servo motor.

The servo motor is a positional (angle) servo driver, which is suitable for the closed-loop control execution module that needs to change and maintain the angle constantly. In our project, we can be adjusting and monitor the angle by PWM. Compared to other motors, the servo motor has good stability and quick response. Therefore, I can control the knob fast and precisely.

Operating Speed: 60/0.15 degree/second (No Load @ 6.0V)

Stall Torque: 9.4 kg/cm @ 4.8V / Stall Torque: 11 kg/cm @ 6V

Running Current: 500mA-900mA @ 6V

Stall Current 2.5A @ 6v

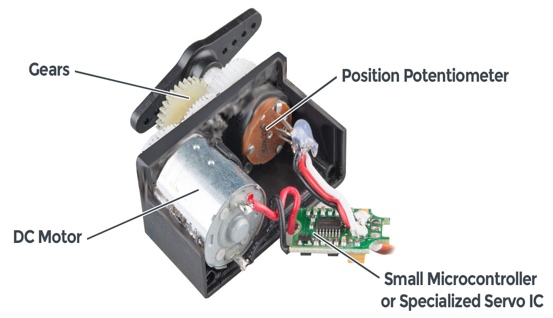


Figure 1 - the structure of servo motor | Table 2 MG996R 180° servo motor features. (MG996R Servo Motor, 2019)

Figure 1 shows servo motor uses gears, which causes it can be high torque, and I can set the degree directly by using its potentiometer. In 6V, the torque can be 11kg/cm (Figure 1 - the structure of servo motor | Table 2 MG996R 180° servo motor features. (MG996R Servo Motor, 2019)). according to my measurement, the force of the knob is \_\_\_\_\_, Which is enough for our project. I did not choose a small servo motor, such as SG92R, So, I want to give some redundancy to prevent subsequent processes from adding weight.

## Processor

The processor’s main function is to drive two motors. It also should take part in connecting different hardware and transferring information. As I discussed with my group member, we will use the computer as the main computing service. Meanwhile, the hardware parts also need the ability to do some light work.

In this project, there are plenty of MCU (Microcontrollers unit) in the market, such as Arduino, Raspberry Pi and so on. As we considered, MCU does not participate in complex calculations. Thus, there is no need to use Raspberry Pi. Moreover, Arduino is our first choice.

### Choosing Arduino

According to the official website, Arduino is an open-source electric platform based on hardware and software. (Arduino, 2018) It can be receiving the input signal and output signal to control the servo motor or other devices, which is easy to use in our project.

In Arduino, there are a few models for users to choose from, such as Uno and Mega.

|  |  |  |
| --- | --- | --- |
|  | Uno | Mega2560 |
| Clock speed | 16MHz | 16MHz |
| Flash memory | 32kB | 256kB |
| EEPROM | 1 | 4 |
| SRAM | 2 | 8 |
| Voltage | 5v | 5v |
| I/O pins | 14 | 54 |
| I/O pins with PWM | 6 | 15 |
| Analog pins | 6 | 16 |

Table 3 The comparison between Arduino Uno and Mega

From Table 3, I compared two popular Arduino controllers, which clearly describe that Arduino Mega is better than Uno. Mega has bigger flash memory than Uno, which is it can store more data. Also, it has fifty-four pin that allows me to insert more hardware. Overall, I decided to choose Mega as our processor. In Arduino, there is another feature call EEPROM(Electrically Erasable Programmable Read-Only Memory). This feature can allow Arduino retain data even after a power outage.

## Components

### GYRO

In this project, we will make sure the maze should put it in horizontal. Thus, I decided to install a GYRO as a sensor to collect the data and analyse to data to balance the maze’s surface. I go to the universities tech hub; they have the GYRO(BNO005) to achieve what I need.

### Screen

To display the information, it is necessary to install a monitor. In Tech Hub, they give me to 2002 screen which can display forty characters that can carry more information instead of using the transition effect of the small screen.

## Mode/Function

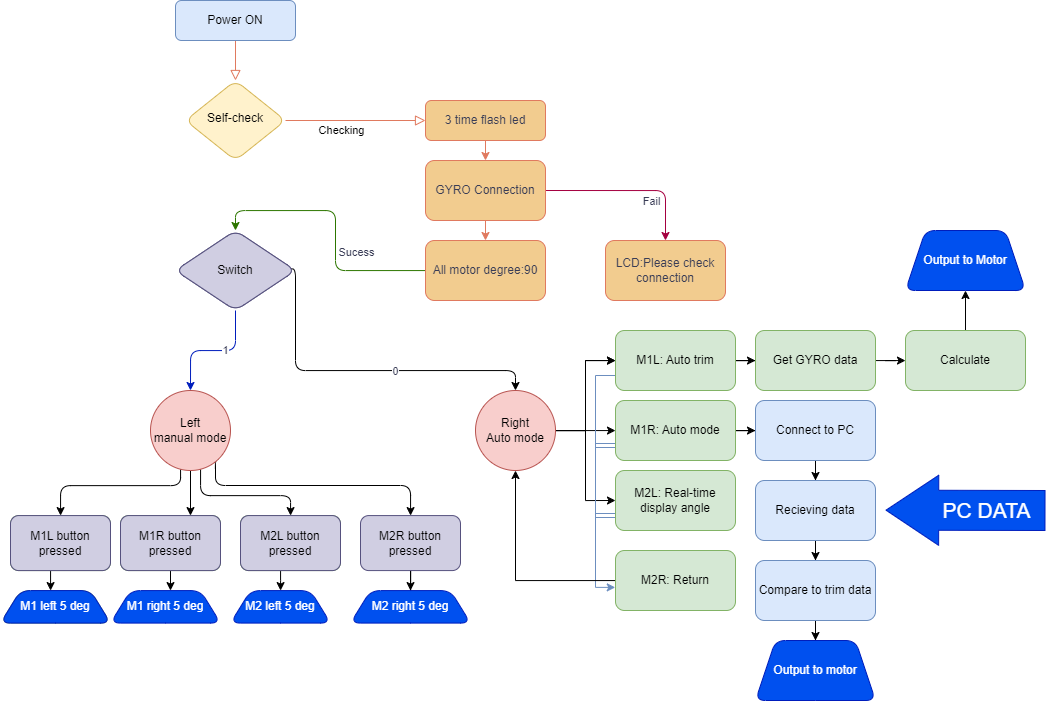


Figure 2 - The function of the hardware

Figure 2 shows the function that I designed for this project. There are three main functions describe below.

### Selfcheck

The selfcheck function can check if every component is connected. For GYRO, it can be checked through the screen. It will display a message if the GYRO is disconnected by Arduino. For led, I used full colour led. In this function, it will flash three channel to check it by visual inspection. If the colour is not white, that means one output is disconnected. For motor, they will turn to 90 degrees to check if it is working.

### Manual mode

By turning the switch, it can be change to manual mode. Manual adjustment can be used for users to manually control the angle of the servo through buttons. It also can control the pitch and roll and used in some situation such as testing and playing.

### Auto mode

In auto mode, there are a few modes user can choose including auto trim, shows degree and connect to pc.

* In auto trim, the maze can automatically balance by the data of GYRO. It can be easily for user without trim it manually. The GYRO should be installed in the maze, it can read the data and transmit to Arduino. There will be a judgment function to analyse the degree. If the degree is lower or higher than set value, then Arduino will calculate the adjustment value and output to motor to realise the trim function.
* Real time display mode can display the axis on screen in real time. So, user can check if it is horizonal and vertical to record the data.
* Connection to PC is using the camera above to capture the image and transmission to PC, PC will give the motor data to Arduino. And Arduino will drive the motor and display the information.

# Design and connection

After determining the pattern, the next step is the design. In this project, it needs to design the circuits and case to make it smaller and smarter. In order to look more like an integrated, we need to make all the components into a whole which needs to connect all components (Table 1).

Firstly, I made a simulation by using Tinker CAD. This software can simulation the basic Arduino and components. It will increase the speed of testing without using physical test on breadboard. The Figure 3 shows the circuits design and simulation.

In this simulation, each component should connect to the digital port in Arduino. In button and switch, they have similar features, which is the closure of the switch to generates a signal. Thus, it should be powered by 5v and put a resistor and signal wire in parallel to avoid short circuits. So, it should get power from the Arduino 5V into the switch/button and connect the signal wire to Arduino digital port. Then, make a resistor in parallel to GND.

In motor connection, the SN996R has three ports which is positive, negative and PWM signal. It just needs to connect to each function. During operation, I found that the servo motor is sometimes out of control. After checking, I found that although the motor can be powered by 5v, other components cannot work properly due to the excessive current demand of the servo motor. Thus, it should be powered by external supply. There are one thing needs to notice that when it uses external power supply, the motor and the Arduino needs to share the same reference voltage which is connect the GND together (green circle in Figure 3). Otherwise, it might cause unstable of the motor, further be damaged.

图示

描述已自动生成

Figure 3 - The simulation of the circuits

After the simulation, I used the laser cut board to make a table and put all component in the table. (Figure 4)

It works well, but it does not meet the needs of portability, so it is necessary to use PCB. PCB is a printed circuit board that can substrate for connection of electronic components and circuit conduction. It should design the wire first and soldering all the components into the PCB. Because all wire will cover in the PCB, thus it can very small and achieve the same function. In Figure 5, it shows the design of this project’s PCB. It can be installed just above the Arduino mega, which is like an extension board.

In fact, this kind of board has never been used by me, because in the later inspection, I found that the production cycle of manufacturers in the market is relatively long, and the cost of independent production is expensive. It is still a good attempt.

图片包含 桌子, 电脑, 游戏机, 男人

描述已自动生成

Figure 4 - The first model

Finally, the electric lab of university gives me a coper board, which allows me to solder the components and wires in it. After I solder it complete as same as the simulation one, here comes this

It can be also insert to the Arduino and it to use, just a little bigger than normal PCB. The two wires are connecting to the GYRO and MOTOR.

For the motor design, after I discussed to my group member, we have better idea. Thus, Yanbo is responsible of 3D design and print and Chenghao did the motor stand as they good at.

**图示, 示意图

描述已自动生成**

Figure 5 - PCB design

# Code

In the code part, I divided the code of the entire project into eight files.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Group Project.ino | Config.h | Init\_setup.ino | Gyro.ino | Selfcheck.ino | Transfer.ino | Switch\_value\_manual.ino | Switch\_value\_mode.ino |

### Group Project.ino

* + In this file, it included the basic running code. This code can be used to run all the functions.

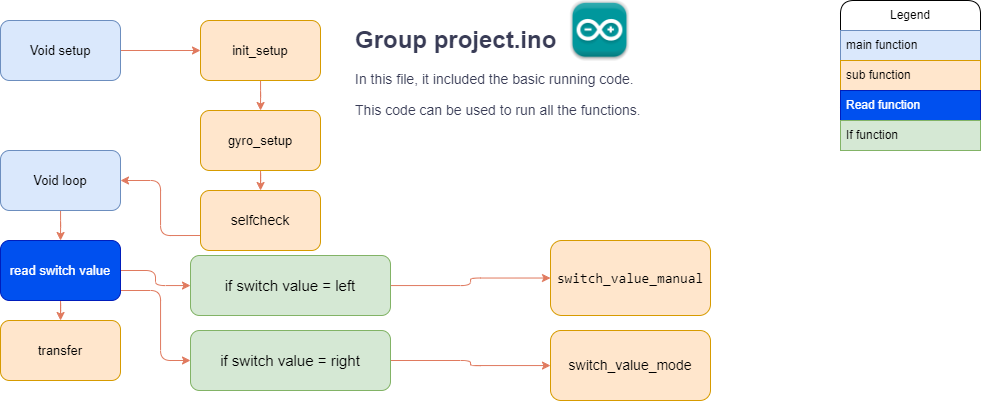


Figure 6 - The structure of Group Project.ino

* + In void setup function, it only runs one time when the Arduino turned on. In this program, it will run three sub function that included init\_setup, gyro\_setup, selfcheck.
  + In void loop function, it contains all the programs. In this program, the main function is checking the switch value. If the switch value is turning left, which means the circuits is cut off. Then, the digital pin cannot receive any voltage level, which is zero. In the same way, the switch will change to one, if the switch is turned to right. All in all, the value can be changed by turning left and right so that Arduino can detect and make actions.

### Config.h

* + In Arduino, config.h is a head file that usually define each pin and libraries.

|  |  |
| --- | --- |
| Config.h | |
| Library | It is a set of pre-written code files designed to solve specific problems or provide specific functions, and developers can import them into their own code for use. A library can contain functions, variables, objects, etc., along with their associated header files, source code, and other resources. |
| adaFruint\_sensor | Many small embedded systems exist to collect data from sensors, analyse the data, and either take an appropriate action or send that sensor data to another system for processing. (Adafruit, 2023) |
| adaFruint\_BNO055 | It is a library for the BNO055 model. This model is a GYRO that I install for our maze, it can provide the class for the hardware sensor that I can configure the sensor and read the data. |
| wire | **It** is a header file in Arduino that provides functions to communicate with I2C |
| LiquidCrystal\_I2C | This library is to use the lcd screen on the Arduino by I2C connection. |
| Servo | Servo library is combined servo function that I can config it easily. |
| Global variable | Is to declare all value in Arduino. |
| LED | Declare each color’s pin. |
| Button | Declare each button’s pin |
| Servo motor | Declare the servo’s pin, degrees and set the current and past degree. |
| Screen | Declare the screen’s I2C address and display’s size. |
| GYRO | Declare the reference degree and other calculation degree name. |
| PC | Declare the Transfer status that used for PC connection and the value for saving the data. |
| Function prototype | In this function, different function code can be decentralized storage in different function. It used to make the program clearly. |
| Transfer | Limit the angle of the motor to prevent damage due to excessive angles. |
| Selfcheck | Check each connection if it is working well. |
| gyro\_setup | Setup the GYRO. |
| gryo\_display | Used to display all angles from GYRO and has difference precision. |
| gryo\_trim | Used to running auto trim. |
| init\_setup | Define the baud rate, lcd, set led to output and servo. Then read the data from EEPROM. |
| switch\_value\_manual | Define the function for manual mode. |
| switch\_value\_mode | Define the function for auto mode. |

### Gyro.ino

* + In this program, it contains all function of GYRO.

图形用户界面, 日程表

描述已自动生成

Figure 7 - The structure of gyro file.

* + In this file, it uses three void function to represent different gyro function.
  + For gyro\_setup, this function is to define the gyro. The code bno.begin() is to check the if the gyro is connected to Arduino. If the gyro is connected, the Arduino will return 1 value to this code. On the opposite, the function will return 0, it will go to if function. In this function, the lcd screen will show the error message to user until the gyro is connected again and press reset button.
  + For gyro\_display, this code describes the display function. In this program, there are some place needs to show different degrees. In this function, it can calculate how much degree needs motor rotate. This function is used for auto trim.
  + For gyro\_trim, it used for auto trim as well. In this function, it can be auto balance by adjust two motors. Firstly, it will read gyro data. Next, there will be four WHILE function to determine the position of the maze board. These four functions can satisfy the angle in any case. Each while function is a loop, if the degree does not conform to the matches. The motor degree will increase or decrease one degree to check if it is match again. It will repeat the above steps until the adjusted angle matches the set angle. User can check its process by checking lcd display or led light.

### Selfcheck.ino

* + In the self-check function, it will run once when the Arduino is power on. At beginning, the lcd display shows welcome and self-test information to let user know what it is doing. Then, led will flash three times as stated above.

### Switch\_value\_manual.ino

* + This function allows user to adjust the motor degree manually by pressing four buttons.

图形用户界面, 应用程序

描述已自动生成

Figure 8 - The chart for manual mode

* + In this chart, I designed LCD refresh function. During each cycle, Arduino will compare the old motor data and new motor data. If new data does not same as old data, the lcd screen will run refresh function. if we insert lcd.clear() code to the normal code, the lcd screen will refresh all the time. Thus, user cannot see the screen clearly.
  + There are four If function that controlling the motor, if user pressed any buttons, the motor would increase or decrease the motor degree by each 5 deg.
  + In case of the readability, the gyro data will not only be showing on the display but also the following decimal part will be omitted, and only the integer will be displayed. The reason of this is to emphasize the main part. In most situation, user only needs to check integer degree. Thus, hiding decimal is necessary.

### Switch\_value\_mode.ino

* + In this mode, it contains the mode function that serves the maze.

图形用户界面, Teams

描述已自动生成

Figure 9 - The chart for switch\_value\_mode

* + From the Figure 9, this chart shows three functions based on gyro and pc.
  + The first mode is auto trim mode. In this while function, it uses two gyro functions as mentioned in gyro part. After finishing auto trim, the value of two motor degree will save to EEPROM and wait for next use.

Connecting to pc is one of the main functions. In this function, it needs to make a protocol to connect to pc. In Arduino, it gives a port that use to connect to pc by USB. It can not only send to data, but also receiving data. Serial.available() is used to detect the signal. This value will be greater than 0 if the data is transmitting. Thus, it can detect the message status. Every time when data is transmitting, it will enter a while function. Serial.parseInt() function will save all data from pc and named “num”.

For example: pc will send 090120 to Arduino, which means motor1=90deg, motor2=120deg. For motor1, the data(090120) will divide by 1000 =90.12. But we define motor1 is an integer, it only takes 90 as an integer. For motor2, it takes data’s remainder, so we can get 120. It needs to combine with GYRO that calculates the absolute degrees. In python, it presumes 90 degree is vertical and horizontal. Thus, the motor needs to rotate 90-90=0 in motor1 and 120-90=+30deg in motor2. To make sure the maze always be balance by putting on different table, we import the absolute vertical and horizontal data from EEPROM that we did auto trim before. So, the exact degree of motor is trim data +0 and trim data +30.

In order to prevent the computer from sending the same data twice to the Arduino, the user cannot judge whether the transmission is completed, a counting function is designed, which can switch back and forth between 1 and 2. So, user can easily check it by watching the display screen.

* + The real time display function also called showing degrees. In this mode, GYRO will collect data each second and drop it to display. On the left side, the data has 4 decimal numbers. On right side, the data only shows integer. This mode is used for testing the maze by checking faster high-precision data and clearer low-precision data.

# Conclusion

In summary, the hardware part of the maze system is designed to use two motor-driven knobs to control the pitch and roll of the maze. This feature enables the ball to follow the guided path and avoid falling into the hole as it moves towards its destination. The maze is controlled by two servo motors connected to an Arduino Mega board. After the system boots, the Arduino performs a self-test to make sure everything is connected and working properly.

The user can choose between two modes: automatic and manual, with additional features such as automatic trimming, connection to a PC and degrees in automatic mode. One of the main benefits of this program is that it retains the original look and functionality of the Maze, allowing people to use it like any other Maze product.

# Contribution

* Design and achieve the whole hardware part.
* Help image processing

# Advantages and Improvements

### Advantages

### Improvements

# Reference

Arduino. (2018, 02 05). *arduino.cc*. Retrieved from https://www.arduino.cc/en/Guide/Introduction

*MG996R Servo Motor*. (2019, 04 03). Retrieved from COMPONENTS101: https://components101.com/motors/mg996r-servo-motor-datasheet