A-**MAZE**-ING

Faculty:

Dr. Ashutosh Mishra

Team Members:

1) Pranav Kandukuru- B210612EC
2) Pothuguntla Tejaswini- B210726EC
3) Mutyala Veera Abhi Nanda- B210708EC
4) Mariserla Ajay Kumar Naidu- B210742EC
5) Nethala Vijay Praneeth- B210658EC
6) Nagalapuram Yoshith Sai- B211291EC

INTRODUCTION

A-MAZE-ING, as the name itself suggests, is an amazing maze game. A-MAZE-ING is an amazing E-Toy (Electronic toy) to everyone, that gives a thrill and kind of joy in solving it. We know that many people are interested in solving mazes that are seen in magazines. This toy best suits them. This is a 3d maze game where the players are supposed to make balls reach the goal by moving the maze. The game is controlled by Arduino Uno. Arduino is an open-source electronics platform based on easy-to-use hardware and software. The Arduino is programmed to navigate through a Maze. This maze is a platform that is tiltable in all directions and at any sort of angle.

To make this game more interesting we introduced four different modes. The difficulty level of the game increases as we move to higher modes. The different modes are such as:

- Normal mode
- Inverse mode
- Slomo mode
- God mode

COMPONENTS

- Arduino UNO
- Joystick module
- Cardboard maze
- Small Metallic Ball / Marble
- Micro Servo Motors
- Ultrasound sensors
- Bread Board
- LCD

WORKING

The players use the joystick for moving the maze. The movement of the maze is made possible by 2 Servo motors that are placed perpendicular to each other for easy and multi-axis movement. The movement of motors based on the mode is controlled by the Arduino Uno. The Arduino code has been written for the mode controls. Shifting from one mode to another mode can also be done in between. LCD screen displays the current mode of the game.

Selection of modes is done by choosing 1,2,3,4 in serial monitor. For coming out of the mode, ‘0’ is selected. This displays ‘BYE WORLD’ on LCD. Based on the analog values of joystick, the Arduino Uno converts into digital values which will be given to servo motors. The digital values cause the angle of tilt in the motor.

NORMAL MODE

In this mode, the movement of the maze will be in the direction of the joystick moved.

INVERSE MODE

The movement of the maze will be in the direction opposite to that of the direction of the joystick. There is a 180-degree phase away from the expected.

SLOMO MODE

This mode is bit interesting. In this mode there will be a delay of 2 seconds in the movement of maze to that of joystick movement. This was made by introducing a delay function in the code.

GOD MODE

This is an exciting mode. For this mode we installed proximity sensors into the circuit. The motion of maze can be controlled by hand gestures. Two sensors were used, one helps in to and fro motion and other for sideways movement.

CODE

```
#include <Servo.h>
#include <LiquidCrystal.h>

Servo servo1; //x-axis
Servo servo2; //y-axis
int joyX = 0; //analog pins
int joyY = 1;
int state;
int flag=0;
int until=1;

int servoVal;

const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int trigPin1 = 6;
const int echoPin1 = 2;

const int trigPin2 = 4;
const int echoPin2 = 13;

long duration;
int distance;

void setup()
{
  servo1.attach(3);
  servo2.attach(5);
  Serial.begin(9600);

  lcd.begin(16, 2);

  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);

  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
}


void loop()
{
```

```
if (Serial.available()>0)
{
  state=Serial.read();
  flag=0;
  lcd.setCursor(0,0);
  lcd.print(" A-MAZE-ING");
}

if (state == '1')
{while(until != '0')
  {
    servoVal = analogRead(joyX);
    servoVal = map(servoVal, 0, 1023, 130, 180);
    servo1.write(servoVal);

    servoVal = analogRead(joyY);
    servoVal = map(servoVal, 0, 1023, 130, 180);
    servo2.write(servoVal);
    delay(50);

    if(flag==0)
    {
      Serial.println("NORMAL MODE");
      lcd.setCursor(0, 1);
      lcd.print(" NORMAL MODE ");
      flag=1;
    }

    if (Serial.available()>0)
    {
      until=Serial.read();
    }
  }
  Serial.println("BYE WORLD");
  until = 1;
}

else if (state == '2')
{while(until != '0')
  {
    servoVal = analogRead(joyX);
    servoVal = map(servoVal, 1023, 0, 130, 180);
    servo1.write(servoVal);

    servoVal = analogRead(joyY);
    servoVal = map(servoVal, 1023, 0, 130, 180);
    servo2.write(servoVal);
    delay(50);

    if(flag==0)
    {
     Serial.println("INVERSE MODE");
     flag=1;
     lcd.setCursor(0, 1);
     lcd.print(" INVERSE MODE");
    }

    if (Serial.available()>0)
    {
     until=Serial.read();
    }
   }
   Serial.println("BYE WORLD");
   until = 1;
  }

  else if (state == '3')
  {while(until != '0')
   {
    servoVal = analogRead(joyX);
    servoVal = map(servoVal, 0, 1023, 130, 180);
    servo1.write(servoVal);

    servoVal = analogRead(joyY);
    servoVal = map(servoVal, 0, 1023, 130, 180);
    servo2.write(servoVal);
    delay(1000);

    if(flag==0)
    {
     Serial.println("SLOMO MODE");
     flag=1;
     lcd.setCursor(0, 1);
     lcd.print(" SLOMO MODE  ");
    }

    if (Serial.available()>0)
    {
     until=Serial.read();
    }
  }
  Serial.println("BYE WORLD");
  until = 1;
}

else if (state == '4')
{
  while(until != '0')
  { digitalWrite(trigPin1, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1, HIGH);

    duration = pulseIn(echoPin1, HIGH);
    distance = duration*0.034/2;

    Serial.print("Distance: ");
    Serial.println(distance);

    servoVal = map(distance, 0, 30, 130, 170);
    servo1.write(servoVal);

    digitalWrite(trigPin2, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin2, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin2, HIGH);

    duration = pulseIn(echoPin2, HIGH);
    distance = duration*0.034/2;

    Serial.print("Distance: ");
    Serial.println(distance);

    servoVal = map(distance, 0, 30, 130, 170);
    servo2.write(servoVal);

    delay(400);

    if(flag==0)
    {
      Serial.println("GOD MODE");
```

```
        flag=1;
        lcd.setCursor(0, 1);
        lcd.print(" GOD MODE    ");
      }

      if (Serial.available()>0)
      {
        until=Serial.read();
      }
    }
    Serial.println("BYE WORLD");
    until = 1;
  }
}
```

HARDWARE

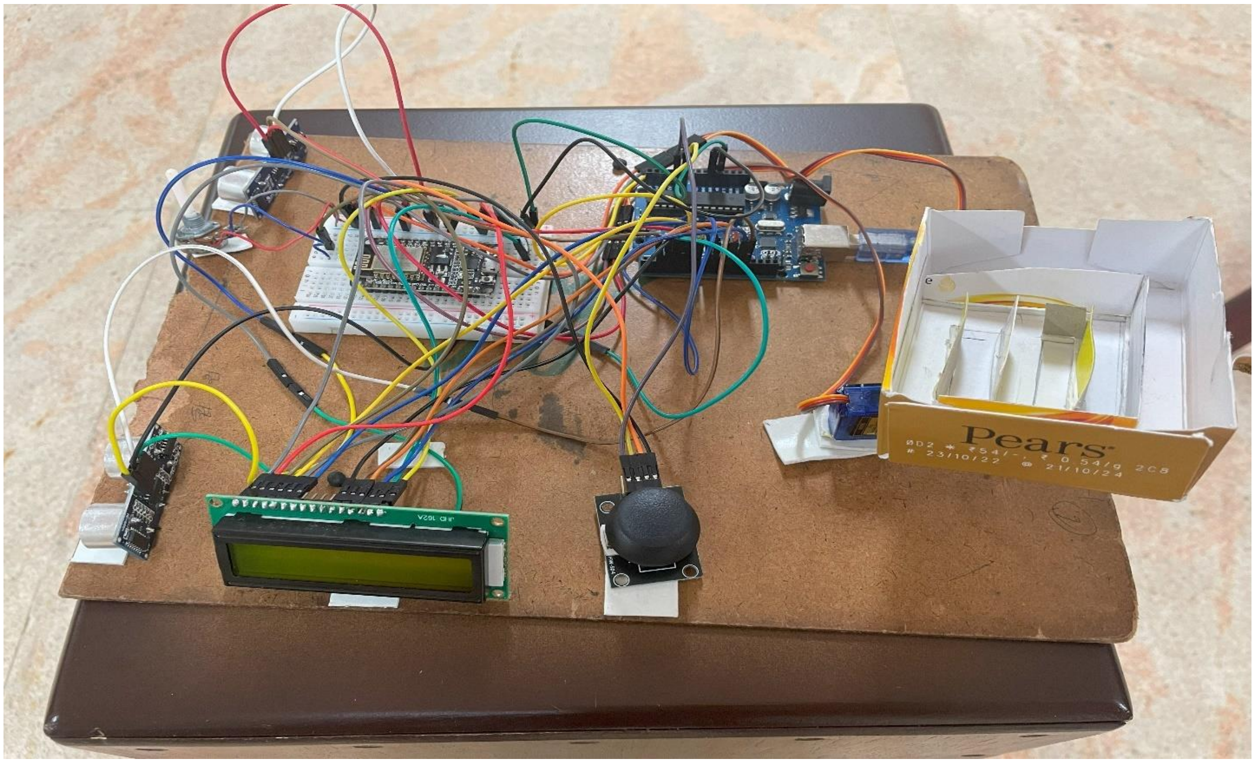

Fig.1: Hardwarwe implementation

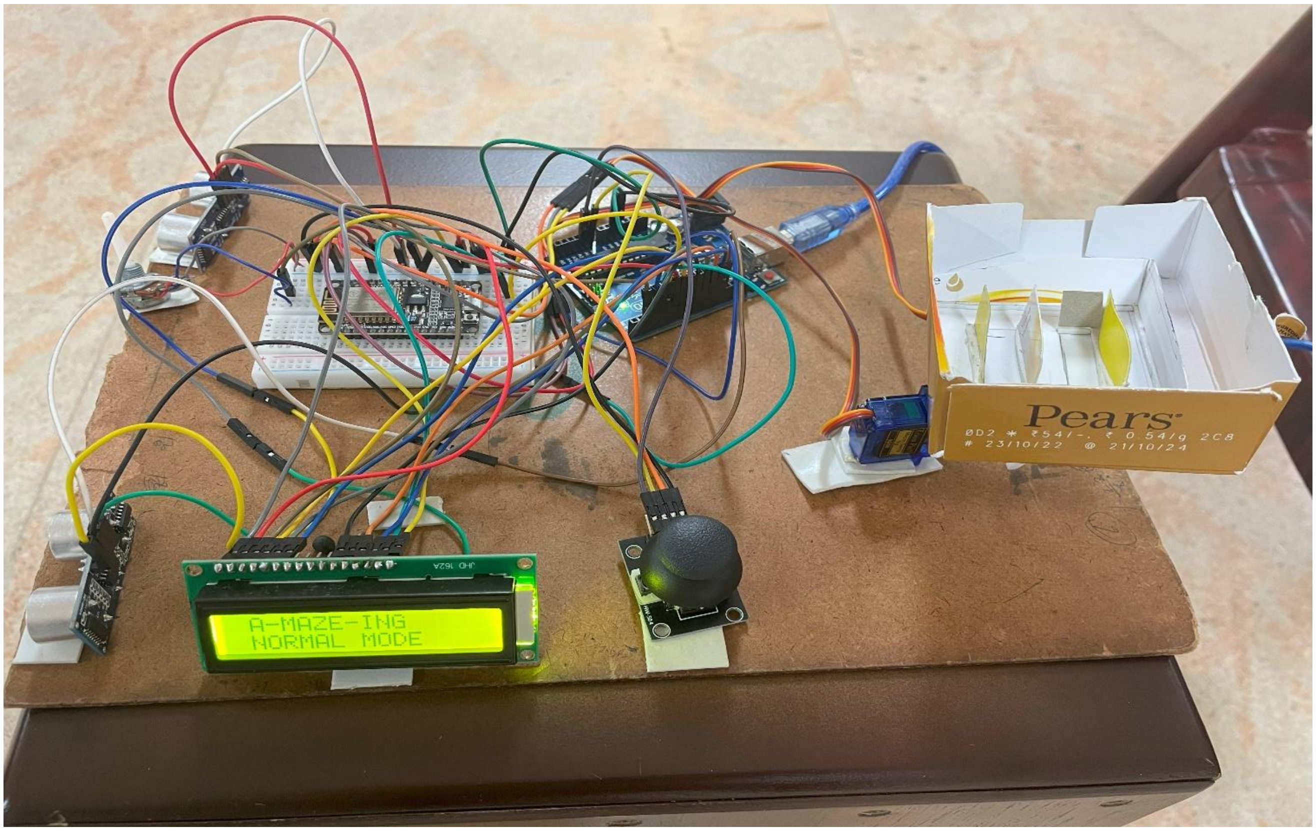

Fig.2: LCD displaying, the game is in normal mode.

CONCLUSION

The project A-MAZE-ING is a wonderful Arduino application. This is a fun toy for children. Different modes of this game makes both children and youth more exciting to play. There are many games that can be made with the help of Arduino. This project helped us in gaining some knowledge in interfacing of Arduino with joystick, servomotors, proximity sensors, LCD.