MAZE SLOVING ROBOT WITH LIVE MONITORING

Sappidi Sriya, Challa Ruchitha, Nagabandi Praneeth, Gumadala Sanjay, Bobby k Simon,

Guide: Dr.Kowdodi Siva Prasad

Computer Science and Engineering (Internet of Things)
Hyderabad Institute of Technology And Management
Medchal,India

ABSTRACT
This project presents a sophisticated maze-solving robot, equipped with advanced live monitoring capabilities, meticulously designed to autonomously navigate and solve intricate maze environments.The robot's navigation prowess is underpinned by an array of sensors, prominently featuring ultrasonic and infrared technologies, which are adept at detecting and avoiding obstacles while simultaneously mapping the layout of the maze. These sensors feed data into an embedded microcontroller, which processes the information in real-time. To determine the optimal path to the maze's exit, the microcontroller employs sophisticated algorithms such as Depth-First Search (DFS) or Breadth-First Search (BFS), renowned for their efficiency in solving complex mazes.A key feature of the robot is its live monitoring capability, enabled by a wireless communication module. This module is responsible for transmitting critical data, including the robot's status, position, and sensory inputs, to a remote monitoring station. The real-time feedback provided through this system allows users to observe
the robot's progress dynamically and make necessary adjustments on the fly, ensuring optimal performance and adaptability in diverse environments. The robot's compact chassis design ensures superior maneuverability, even in the most confined spaces, while a robust power management system guarantees extended operational periods, making the robot both efficient and reliable.The integration of live monitoring capabilities not only significantly enhances the robot's functionality but also offers invaluable insights into its performance and decision-making processes. This feature is particularly advantageous for educational purposes, providing students and researchers with a tangible demonstration of theoretical concepts in robotics and automation. Additionally, the real-time data acquisition and analysis capabilities make the robot a potent tool for research, especially in fields requiring precise navigation and obstacle avoidance, such as search and rescue operations. Here, the ability to receive immediate feedback and monitor the robot's actions in real-time is crucial, as it can directly influence the success of missions in dynamic and potentially hazardous environments.The development of this maze-solving robot represents a significant leap forward in the realms of robotics, automation, and remote monitoring technologies. By addressing real-world challenges with enhanced efficiency, this project underscores the potential of advanced robotics to transform a variety of applications. The robot's design, characterized by a harmonious blend of sophisticated sensor arrays, intelligent pathfinding algorithms, and robust real-time monitoring systems, exemplifies the cutting-edge advancements in technology. This convergence of features not only demonstrates the practical applications of theoretical research but also paves the way for future innovations in autonomous systems, highlighting their capability to operate effectively in real-world scenarios and to provide critical insights that can drive further technological progress.
Keywords: maze-solving robot, autonomous navigation, live monitoring, ultrasonic sensors, infrared sensors, flood-fill algorithm, real-time processing, wireless communication.

1. INTRODUCTION

Introducing a revolutionary maze-solving robot with live monitoring capabilities. This cutting-edge autonomous system is equipped with advanced sensors and algorithms, enabling it to navigate complex mazes while avoiding obstacles in real-time. The incorporation of live monitoring through wireless communication allows for dynamic observation of the robot's progress and decision-making, providing valuable insights for educational, research, and practical applications. Compact yet robust, this robot represents a significant advancement in robotics and remote monitoring technology, promising efficient and effective solutions for a variety of real-world challenges

2. LITERATURE SURVEY

1.Intuitive solution for Robot Maze Problem using Image Processing

Path finding is an important problem in robot design and automation that requires quick error-free solutions that rely on external environment. Automated mobile robotic systems employ various techniques to determine the path that the robot needs to follow to reach the destination to perform its function. Path finding problems can utilize various algorithms to solve the problem. Sensor data can be used as a reference to determine the path to be followed from the start point to the destination. However, this technique is highly localized and does not provide the ability to make decisions by taking global constraints or conditions into consideration. Image processing techniques are employed extensively to provide a solution based on global conditions. The proposed method involves use of image processing to process the acquired image of the maze from a mounted camera system. The processing steps are used to provide steps which the robot can follow to reach from its current position to the final position. This project implements a universal algorithm to allow the robot to maneuver autonomously.

2.Intelligent Maze Solving Robot Based on Image Processing and Graph Theory Algorithms

The most important task for maze solving robots is the fast and reliable finding of its shortest path from its initial point to its final destination point. This paper proposes an intelligent maze solving robot that can determine its shortest path on a line maze based on image processing and artificial intelligence algorithms. The image of the line maze is captured by a camera and sent to the computer to be analyzed and processed by a program developed using Visual C++ and OpenCV libraries and based on graph theory algorithms. The developed program solves the captured maze by examining all possible paths exist in the maze that could convey the robot to the required destination point. After that, the best shortest path is determined and then the instructions that guide the car-like robot to reach its desired destination point are sent to the robot through Bluetooth. The robot follows the received guide path to reach its destination. The proposed approach works faster than the traditional methods which push the robot to move through the maze cell by cell in order to find its destination point.Moreover, the proposed method allows the maze solving robot to avoid trapping and falling in infinity loops.

3 .Survey on techniques used in Autonomous Maze Solving Robot

This paper presents a survey on techniques used in Autonomous Maze Solving Robot and Micromouse. Autonomous movement is an important feature which allows a robot to move freely from one point to another without the mediation from human being. The micromouse is required to solve any kind of maze in shortest interval of time. Autonomous movement within the unknown area requires the robot to investigate, situate and plan the outlaying area. By solving a maze, the referenced algorithms and their pros and cons can be studied and analyzed.

4 Maze solving robot using image processing

Maze solving problem involves determining the path of a mobile robot from its initial position to its destination while traversing through environment consisting of obstacles. In addition, the robot must follow the best possible path among various possible paths present in the maze. Applications of such autonomous vehicles range from simple tasks like robots employed in industries to carry goods through factories, office buildings and other workspaces to dangerous or difficult to reach areas like bomb sniffing, finding humans in wreckage, etc. Existing robots employed in labyrinth problems use long process of training and are incapable of adjusting to dynamic environments. The method proposed here involves image processing and path finding algorithm; which works faster because of beforehand acquiring the maze's data rather than going through the maze cell by cell. The entire maze is captured to determine the possible paths and using Direction Envelope Algorithm for finding the best route. This approach gives robot a prognosis and avoids being trapped or falling in loops

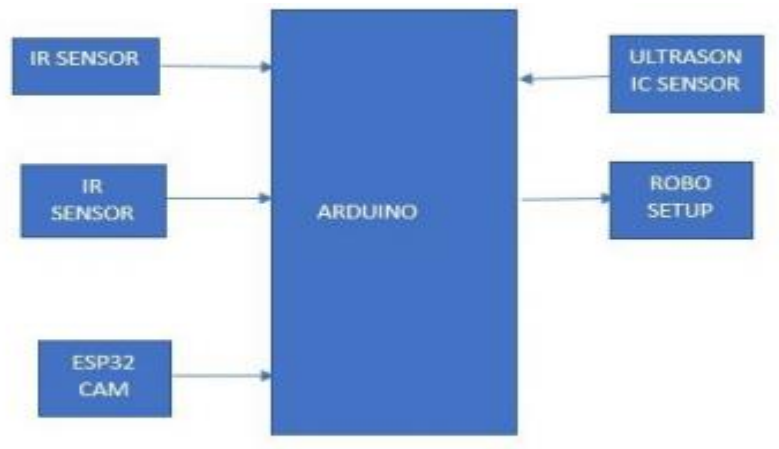
3. PROPOSAL METHOD

The proposed system aims to design an IoT-based maze-solving robot leveraging Arduino, ultrasonic sensors, IR sensors, a robust robotic setup, and an ESP32 CAM module. This innovative approach combines hardware components with IoT capabilities to create an efficient maze-solving mechanism.At its core, the system utilizes Arduino microcontrollers to control the robot's movement and sensor interactions. Ultrasonic sensors enable the robot to detect obstacles and navigate through the maze by measuring distances accurately. IR sensors enhance its navigation capabilities by detecting specific markers or lines within the maze layout, providing additional guidance.The robust robotic setup ensures the stability and reliability of the robot's movements, allowing it to traverse the maze with precision. By integrating an ESP32 CAM module, the system gains the ability to capture realtime images or video footage of the maze environment. This feature can be leveraged for advanced maze analysis, remote monitoring, or even implementing image recognition algorithms for more complex maze-solving strategies.The IoT aspect of the system enables remote control and monitoring functionalities. Users can interact with the robot, receive status updates, and even view live footage of its progress through a connected mobile or web application. This connectivity also opens up possibilities for collaborative maze-solving scenarios, where multiple users can contribute to solving the maze remotely.

ADVANTAGES

- Remote Monitoring and Control
- Data Logging and Analysis
- Enhanced Navigation
- Educational

4.BLOCK DIAGRAM



4.1 ARDUINO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

4.2 Ultrasonic sensors

Ultrasonic sensors measure distance by using ultrasonic waves.The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception. An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

4.3 ESP32 CAM

NODEMCU ESP32

ESP32 camera is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

4.4 ROBOSETUP

The RoboSetup phase encapsulates the initial configuration and assembly of themazesolving robot, bridging the gap between concept and tangible creation. At its core, RoboSetup involves the meticulous arrangement of hardware components, including the Arduino microcontroller, ultrasonic sensor, IR sensor, and ESP32CAM camera module, onto a sturdy chassis or framework.Physically, this entails precision in wiring and connections, ensuring secure attachment and optimal positioning of each component. Each wire and module placement is critical, laying the groundwork for seamless functionality and efficient operation

5.APPENDIX

5.1 ROBOT RUNNING

```
// Define pin numbers for ultrasonic sensor
#define TRIG_PIN 9
#define ECHO_PIN 10
//#define ENA 6 // PWM pin for motor speed control
#define IN1 7 // Motor control input 1
#define IN2 8 // Motor control input 2
//#define ENA2 11 // PWM pin for motor speed control
#define IN11 12 // Motor control input 1
#define IN22 13 // Motor control input 2
// Define pin numbers for IR sensors
#define LEFT_IR_PIN 2
#define RIGHT_IR_PIN 3
// Define threshold distance for obstacle detection (in centimeters)
#define OBSTACLE_DISTANCE_THRESHOLD 20
void setup() {
// Initialize serial communication
Serial.begin(9600);
// Set ultrasonic sensor pins as input/output
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
// Set IR sensor pins as input
41pinMode(LEFT_IR_PIN, INPUT);
pinMode(RIGHT_IR_PIN, INPUT);
//pinMode(ENA, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
//pinMode(ENA2, OUTPUT);
pinMode(IN11, OUTPUT);
pinMode(IN22, OUTPUT);
//int motorSpeed = 60;//0-255) to change speed
//// int motorSpeed1 =55;
// analogWrite(ENA, motorSpeed);
//analogWrite(ENA2, motorSpeed1);
}
void loop() {
// Measure distance using ultrasonic sensor
long duration, distance;
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
42distance = duration * 0.034 / 2
// Read IR sensor values
```

```
int leftIRValue = digitalRead(LEFT_IR_PIN);
int rightIRValue = digitalRead(RIGHT_IR_PIN);
Serial.println(leftIRValue);
Serial.println(rightIRValue);
Serial.println(distance);
delay(1000);
// Check if obstacle is detected
if (distance < OBSTACLE_DISTANCE_THRESHOLD) {
// Obstacle detected, decide which direction to turn based on IR sensor readings
turnRight();
} else {
// No obstacle, move forward
moveForward();
}
if (rightIRValue == LOW) {
// Obstacle on the right, turn left
turnRight();
} else if (leftIRValue == LOW) {
// Obstacle on the left, turn right
turnLeft();
43} else {
// Obstacle in front, stop
//stop();
}
}
void moveForward()
{ digitalWrite(IN1, 0);
digitalWrite(IN2, 250);
digitalWrite(IN11, 0);
digitalWrite(IN22, 250);
}
void turnLeft()
{ digitalWrite(IN1, 150);
digitalWrite(IN2, 0);
digitalWrite(IN11, 150);
digitalWrite(IN22, 0);
delay(100);
digitalWrite(IN1, 0);
digitalWrite(IN2, 150);
digitalWrite(IN11, 150);
digitalWrite(IN22, 0);
delay(500);
44}
void turnRight()
{ digitalWrite(IN1, 0);
digitalWrite(IN2, 0);
digitalWrite(IN11, 0);
digitalWrite(IN22, 0);
```

```
delay(100);
digitalWrite(IN1, 150);
digitalWrite(IN2, 0);
digitalWrite(IN11, 0);
digitalWrite(IN22, 150);
delay(500);
}
void stop()
{ digitalWrite(IN1, 0);
digitalWrite(IN2, 0);
digitalWrite(IN11, 0);
digitalWrite(IN22, 0);
}
```

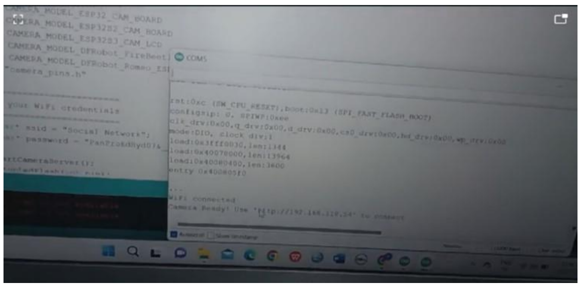
5.2 ESP32 CAM

```
#include "esp_camera.h"
#include <WiFi.h>
45// #include "DHT.h"
// DHT dht2(2,DHT11);
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
// Ensure ESP32 Wrover Module or other board with PSRAM is selected
// Partial images will be transmitted if image exceeds buffer size
// You must select partition scheme from the board menu that has at least 3MB
APP space.
// Face Recognition is DISABLED for ESP32 and ESP32-S2, because it takes
up from 15
// seconds to process single frame. Face Detection is ENABLED if PSRAM is
enabled as well
// ===================
// Select camera model
// ===================
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has
PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
//#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
46#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
//#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
// * Espressif Internal Boards *
//#define CAMERA_MODEL_ESP32_CAM_BOARD
//#define CAMERA_MODEL_ESP32S2_CAM_BOARD
//#define CAMERA_MODEL_ESP32S3_CAM_LCD
//#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
//#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
```

```
#include "camera_pins.h"
// ===========================
// Enter your WiFi credentials
// ===========================
const char* ssid = "Social Network";
const char* password = "PanProEdHyd07& ";
void startCameraServer();
void setupLedFlash(int pin);
void setup()
{ Serial.begin(115200);
// dht2.begin();
Serial.setDebugOutput(true);
47Serial.println();
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
48config.frame_size = FRAMESIZE_UXGA;
config.pixel_format = PIXFORMAT_JPEG; // for streaming
//config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
config.fb_location = CAMERA_FB_IN_PSRAM;
config.jpeg_quality = 12;
config.fb_count = 1;
// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(config.pixel_format ==
PIXFORMAT_JPEG){ if(psramFound()){
config.jpeg_quality = 10;
config.fb_count = 2;
config.grab_mode = CAMERA_GRAB_LATEST;
} else {
```

```
// Limit the frame size when PSRAM is not available
config.frame_size = FRAMESIZE_SVGA;
config.fb_location = CAMERA_FB_IN_DRAM;
}
49} else {
// Best option for face detection/recognition
config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
config.fb_count = 2;
#endif
}
```

6. OUTPUT



```
CAMERA_MODEL_ESP32_CAM_BOARD
CAMERA_MODEL_ESP32S2_CAM_BOARD
CAMERA_MODEL_ESP32S3_CAM_LCD
CAMERA_MODEL_DFRobot_FireBeet
CAMERA_MODEL_DFRobot_Romeo_ESI
"camera_pins.h"

your WiFi credentials

ssid = "Social Network";
password = [illegible]

rtCameraServer();
```

```
rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0
...
WiFi connected
Camera Ready! Use 'http://192.168.119.34' to connect
```

Fig 6.1. Link providing from ESP32 Cam

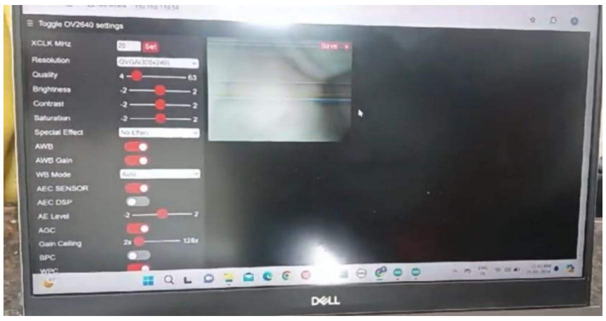


Fig 6.2. Live Monitoring

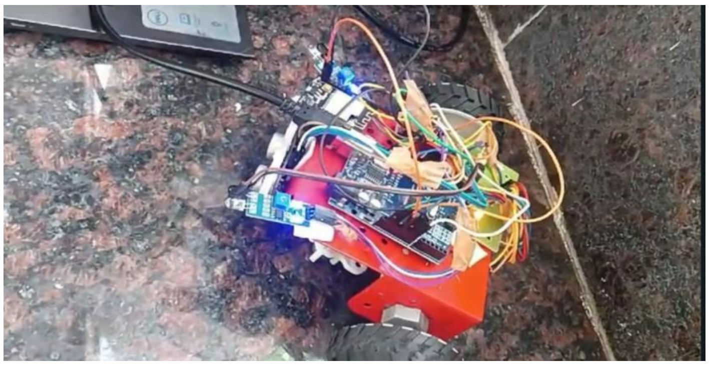


Fig 6.3. Robot is detecting the Obstacle

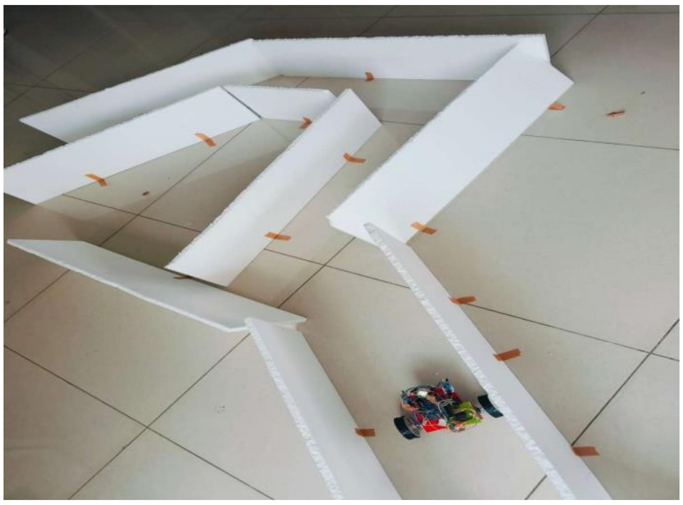


Fig 6.3.Robot Running In Maze

7.CONCLUSION

An IoT-based maze-solving robot leveraging Arduino, ultrasonic sensor, IR sensor,ESP32CAM, and a meticulously designed setup represents a formidable blend of technology with potential applications in various domains. By integrating remote monitoring, data logging, and cloud connectivity, this robot offers unprecedented levels of control, adaptability, and scalability. However, its implementation demands a careful balance between complexity, cost, and reliability. While the advantages of real-time monitoring, data analysis, and remote control are compelling, challenges such as power management, security considerations, and compatibility issues require thorough attention. Nonetheless, with the right approach to design implementation, and continuous refinement, this IoT-powered robot stands poised to revolutionize maze navigation, educational exploration, and beyond, offering a glimpse into the exciting possibilities at the intersection of robotics and IoT technology

8.REFERENCES

1.https://www.google.com/search?q=Intuitive+solution+for+Robot+Maze+Problem+using+Image+Processing&rlz=1C1JJTC_enIN1008IN1008&oq=Intuitive+solution+for+Robot+Maze+Problem+using+Image+Processing&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDExNDdqMGo3qAIIs AIB&sourceid=chrome&ie=UTF-8
2.https://www.google.com/search?q=Intelligent+Maze+Solving+Robot+Based+on+Image+Processing+and+Graph+Theory+Algorithms&rlz=1C1JJTC_enIN1008IN1008&oq=Intelligent+Maze+Solving+Robot+Based+on+Image+Processing+and+Graph+Theory+Algorithms&gs_lcrp=EgZja HJvbWUyBggAEEUYOdIBCDIxMDRqMGo3qAIIsAIB&sourceid=chrome&ie=UTF-8
3.https://www.google.com/search?q=Survey+on+techniques+used+in+Autonomous+Maze+Solving+Robot&rlz=1C1JJTC_enIN1008IN1008&oq=Survey+on+techniques+used+in+Autonomous+Maze+Solving+Robot&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDEzMDVqMGo3qAIAsAI A&sourceid=chrome&ie=UTF-8