Proceedings of the
International Conference on Mechanical Engineering and Renewable Energy 2013
(ICMERE2013) 1 - 3 May 2014, Chittagong, Bangladesh

ICMERE-2013-PI-046

DEVELOPMENT OF A CLUSTER MAZE SOLVER ROBOT SYSTEM

Ahamed Nasif Hossain Aoyon[1,*], **Md. Tazul Islam**[2], **Mahmudul Hasan Russel**[3] **and Tarun Kanti Roy**[4]

[1,2]Department of Mechanical Engineering, CUET, Chittagong 4349, Bangladesh
[3,4]Department of Electrical and Electronics Engineering, CUET, Chittagong 4349, Bangladesh

[1,*]aoyon7@gmail.com, [2]tazul2003@yahoo.com, [3]mahmudulrussel.eee.bd@gmail.com, [4]tkroy777@gmail.com

Abstract*-The International Autonomous Robotic Competition has been a prestigious event among engineering students all over the world. The challenge of iARC'13 was to prepare 2 autonomous line-following robots which can communicate with each other so to simultaneously solve 2 mazes via line-following. So the aim of our research project was, one of the robot have to solve the maze according to the turn indicators and the other one have to follow the first one by communicating between them. The design method of the iARC Maze Solver consisted of two stages. The first stage was to design and construct individual components. The second stage of the design process was to interface the components to form the final prototype. We have to develop an artificial intelligence system which is able to decide the shortest path by analyzing the combination of turn indicators and generate a detailed field map of the game field. The individual components of this maze solver robot system consist of the motor control system, navigation sensor array, nRF24L01 wireless communication system and a mapping system for navigating the maze intelligently. A test plan was developed in order to validate the overall performance of the cluster maze solver system. The robots of our cluster maze solver system were successful to navigate within an unknown area by exploring, localizing and mapping its surroundings. The robots were able to learn the maze, find all possible routes and solve it using the shortest one.*

Keywords: Autonomous, Maze solver, Communication, Artificial Intelligence, Mapping

1. INTRODUCTION

Maze-solving robots are one kind of vastly complicated line following robots which are equipped with advanced algorithms to solve mazes. In 15th march of last year (2013) Indian institute of technology arranged a maze solving competition named 2nd International autonomous robotic competition (iARC). The challenge of this competition was to develop two maze-solving robot which can work simultaneously in a cluster architecture by communicating wirelessly with each other so as to simultaneously solve the maze. Another challenge of this competition was to develop these maze solving robots in a compact 200x200x200 mm form factor. The mazes ware formed by concentric circles connected to each other at some specific locations. There were two similar mazes with one maze having dimensions in a ratio greater than 1 radially. The 1st maze contained some random turn indicators which direct our 1st maze solving robot to the goal. The 1st maze solving robot had to scan through the whole maze and compute the pathway to the goal. Then it transmits the pathway wirelessly to the 2nd maze solving robot which uses this pathway to navigate through the 2nd maze to reach the goal.

The aim of this paper is to present our maze-solving robot's design. It's essentially an integrated system in which an embedded CPU, which monitors an assortment of sensors, computes the pathway to the goal, and directs the motors. Although a lot of ideas have been proposed for the design, yet none was proven to be ideal which still makes the design a challenging engineering project with plenty of confusing alternatives.

Following overview of maze will cover the challenges that we faced in the game field of iARC. A broad description of the robot including its components, sensors, control software and the maze-solving algorithm will then follow. Finally we conclude this paper with key results and conclusions that could improve future designs.

2. OVERVIEW OF MAZE

A maze is a tour puzzle in the form of a complex branching passage through which the solver must find a route. In everyday speech, both maze and labyrinth denote a complex and confusing series of pathways, but technically the maze is distinguished from the labyrinth, as the labyrinth has a single through-route with twists and turns but without branches, and is not designed to be as difficult to navigate. The pathways in a maze or labyrinth are fixed (pre-determined) but where the paths can change during the game are categorized as tour puzzles. The Cretan labyrinth is the oldest known maze.

In the game field of iARC mazes ware formed by concentric circles connected to each other at some specific locations (as shown in Fig. 1). There were two similar mazes with one maze having dimensions in a ratio greater than 1 radially. The goal was located at the center of the maze and the bot had start from a starting zone located at the outer circle of the maze. The starting zone was painted white and white circular lines on green field denotes the pathways of the maze. The turn indicators and the goal was painted black.

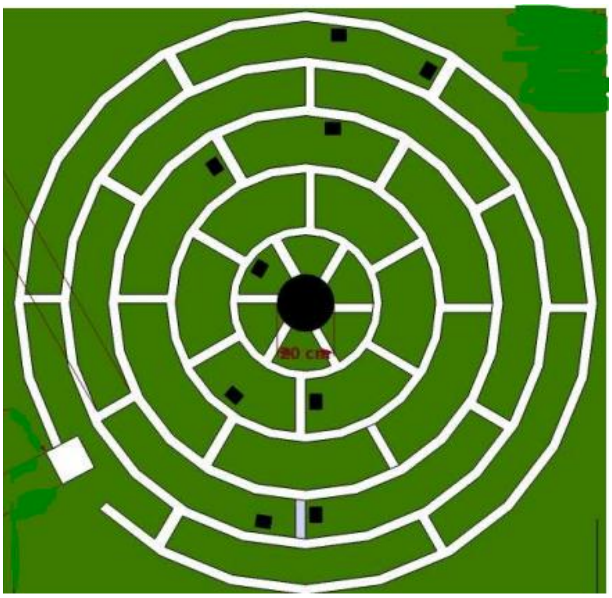

Fig 1: iARC Game field

3. OVERVIEW OF MAZE SOLVING ROBOT

The maze-solving robots are consists of following basic components:

- Custom designed chassis
- Processing Unit (ATMega2560)
- Analog sensor array (8 Photo transistors)
- L293D Motor driver.
- 2200mAh Li-Ion battery as power source.
- nRF Communication module.

In addition of these, there are two motors with a RPM of 120 per min and they can easily carry a weight of about 500gm. Controlling the motors is responsible for safely moving the wireless robot through the maze. The chases need to carry the board, circuits, sensors and batteries and the chases are designed to turn 90 and 180. If the and track branching or ending are available in the routing make the decision based on the algorithms and the detected final path is to be stored in to the registers in the ATMega2560. The programs are transfers using USB data cable.

3.1 Chassis and motors

The chassis is 200x200x200 mm, made of poly-carbonate. It moves on two bulky wheels and a single cluster wheel at front. The diagram above (Figure 2) illustrates how different parts of the system are arranged about the chassis. Two DC motors drive the rear-wheels independently. The front contains a cluster wheel so it can be easily steered. Consequently, the bots can make smooth turns on a curved path by rotating the rear-wheels at different speeds [1]. It can also do in place turns, about the center of mass, by rotating the rear wheels in opposite directions at the same speed. The Li-Ion battery, which represents a considerable portion of the total mass, is attached to the rear end of the chassis to set the center of mass at a mid-point between the rear wheels. That is necessary to avoid skidding during in place turns.

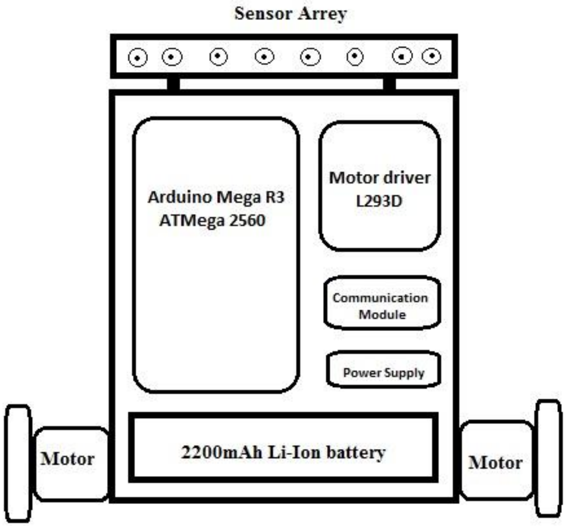

Fig 2: Maze solving bot chassis

The DC motor rotates at a speed that's proportional to its operating voltage. The voltage regulators, and bypass capacitors help provide a stable voltage all the time. Also, rapid successive switching of the motors is avoided in the software to evade spikes on the power line. Thus, the motors speed is almost constant.

Knowledge of the constant rotational speed of the motors and the duration they're switched on, gives enough information to track the displacement of the bot as it moves [2]. To drive the DC motors an integrated H-Bridge circuit is used. The L293D motor driving IC facilitates independent switching of both motors.

3.2 Processing unit

The microcontroller is the processing unit, which monitors the sensors, executes the search algorithm to change the bot's heading, and controls the motors correspondingly. A micro-controller is the most adequate unit to be used compared to other processing units like microprocessors or DSPs. Microcontrollers have built in RAM, Flash ROM, I/O ports, interrupt controllers and counters, which makes them compact, cheap, and simple. We have used Arduino MEGA, which is an open source microcontroller board based on the ATmega2560 [7] as the processing unit our maze-solving robot. By plugging in a USB cable into the Arduino MEGA, we can be able to upload the programs to the microcontroller and also by providing the external adapter to the Arduino MEGA, we can activate the Arduino MEGA board. Some its specifications are described below:

Specifications:

- 54 Digital I/O Pins
- 16 Analog Input Pins
- Flash Memory 256 KB

- ❖ SRAM 8 KB
- ❖ EEPROM 4 KB
- ❖ Clock Speed 16 MHz

3.3 Sensors

The maze-solving makes use of only a single type of sensors, namely photo transistors. The sensors provides high levels of gain, and standard devices are low cost and can be used for detecting a wide range of colors accurately.

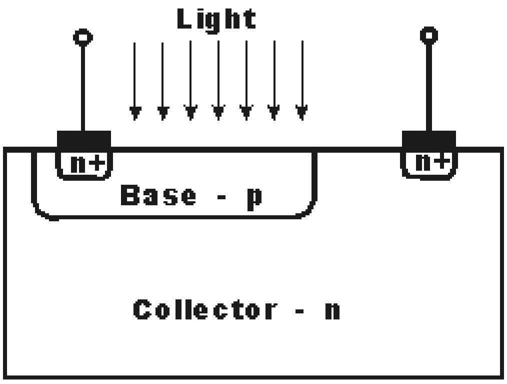

Fig 3: Photo transistor's working principle

Although ordinary transistors exhibit the photosensitive effects if they are exposed to light, the structure of the phototransistor is specifically optimized for photo applications. The photo transistor has much larger base and collector areas than would be used for a normal transistor (as shown in Fig. 3). As a result these sensors are better optimized for light sensitivity than other visual sensors like LDR.

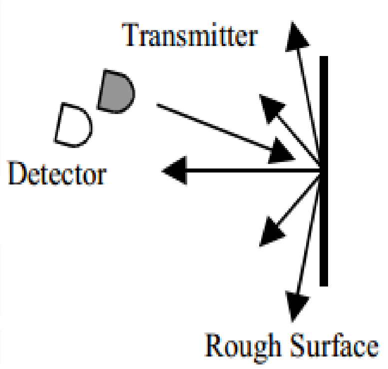

Fig 4: Effect of surface-emissivity

We used 8 phototransistors in our sensor array. The outer sensors were used for detecting the path intersections in the maze and the turn indicators. The inner four sensors were used for smooth navigation of the robot. Green LEDs works as the emitter of our sensor array and this light gets reflected by the maze and our phototransistors pick-up this light and analyze it spectrum (as shown in Fig. 4). Thus it differentiates between the open zone, path and turn indicator of the maze and give the processing unit proper instruction to navigate through the maze.

3.4 Motor driving system

The robot uses DC electric motors, and in order to control and drive them, a system incorporating a power converter/regulator is needed. The power from the chassis (External) battery must be translated to the 12V needed by the DC motors, and regulated in such a way as to provide speed, acceleration, and directional control to the robot. L293D is a dual H-Bridge motor driver, so with one IC we can interface two DC motors which can be controlled in both clockwise and counter clockwise direction (as shown in Fig. 5). L293D has output current of 600mA and peak output current of 1.2A per channel [6]. Moreover for protection of circuit from back EMF output diodes are included within the IC. The output supply (VCC2) has a wide range from 4.5V to 36V, which has made L293D a best choice for DC motor driver.

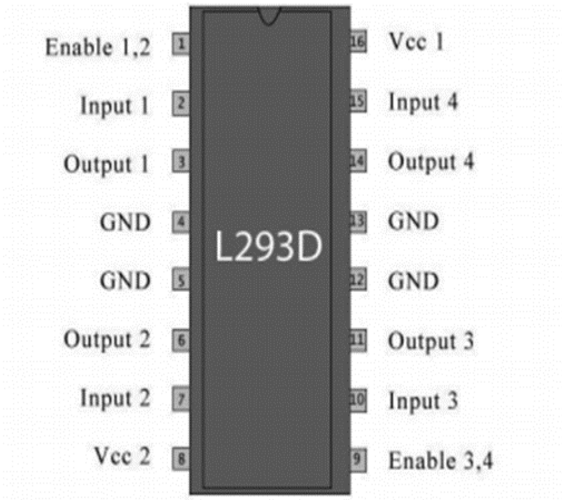

Fig 5: L293D motor driver

Specifications:

- ❖ Supply Voltage Range 12V
- ❖ 36V Output current capability per driver
- ❖ Separate Input-logic supply
- ❖ It can drive small DC-geared motors, bipolar stepper motor.
- ❖ Pulsed Current 1.2-A Per Driver
- ❖ Thermal Shutdown
- ❖ High-Noise-Immunity Inputs

3.5 Power

The power supply block is made up of a 11.9Volt 2200mAh Li-Ion battery feeding a power distribution board that provides regulated supplies of 5V for the microcontroller and the sensors, 11.9Volt for the motors.

To evade false-triggering of the sensors, we've avoided rapid-successive switching of the motors in the software. The Li-Ion battery adds a considerable weight to the bot but it is the only resort when dry batteries fail to satisfy the power consumption requirements for a reasonable duration during development.

The following analysis shows the power consumption of the system:

- ❖ Motor: 12V – 900mA.
- ❖ LEDs: 5V – 70mA.

- ❖ Sensors: 5V – 2mA.
- ❖ Microcontroller: 5V – less than 200 µA
- ❖ Total Power Consumption = 4.951Watt

3.6 Communication module

We use 2.4GHz nRF24L01 module for the wireless communication between out maze solving robots. These RF modules are cheap, robust and powerful as a result they are a great way to communicate wirelessly between robots (as shown in Fig. 6). We used the Nordic nRF24L01 variant of this RF module which integrates a complete 2.4GHz RF transceiver, RF synthesizer, and baseband logic including the hardware protocol accelerator supporting a high-speed SPI interface for the application controller.

Nordic nRF24L01 is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz. This frequency band occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps and a bandwidth of less than 2MHz at 2Mbps. nRF24L01 can operate on frequencies from 2.400GHz to 2.525GHz.

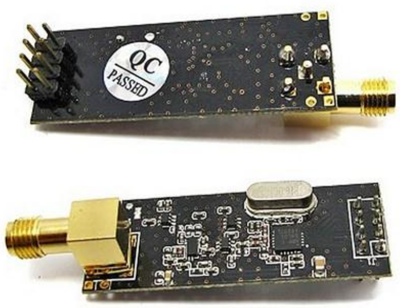

Fig 6: 2.4GHz nRF24L01 module for the wireless communication

These nRF24L01 transceivers send and receive data in 'packets' of several bytes at a time. There is built-in error correction and resending, and it is possible to have one unit communicate with up to 6 other similar units at the same time. These modules communicates through a Serial Peripheral Interface (SPI) with the microcontroller. The register full map is accessible through the SPI, contains all configuration registers in the nRF24L01 and is accessible in all operation modes of the chip.

Specifications:

- ❖ 2.4GHz ISM band operation
- ❖ 126 RF channels
- ❖ 1 to 32 bytes dynamic payload length
- ❖ 1.9 to 3.6V supply voltage
- ❖ 4-pin hardware SPI

4. MAZE SOLVING ALGORITHM

Mazes built for iARC contests are generally not simple ones and are usually built in a challenging way, which is difficult to solve by traditional algorithms directly. What makes the maze-solving algorithm more complicated is developing one for an embedded system [4]. Embedded systems are ones that impose tight time and memory constraints on the system. These constraints are mainly:

1. Real-time processing.
2. Limited memory.
3. Power limitations.

The main challenge of 1st maze-solving bot of cluster system was not only to solve the maze but also communicate simultaneously with 2nd bot so that it can also solve the maze by receiving maze pathway from the 1st bot. At first the 1st bot had to solve the maze and generate a pathway so that it can be followed by 2nd bot. There were turn indicators on the maze for the 1st bot so that it can use those information to solve the maze.

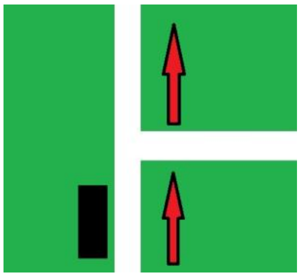

Fig 7: Go straight

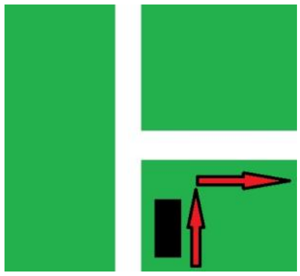

Fig 8: Turn right

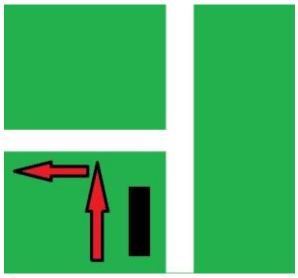

Fig 9: Turn left

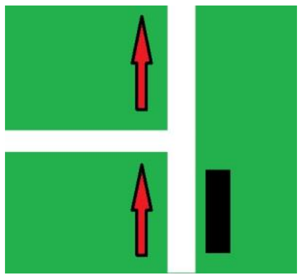

Fig 10: Go straight

The 1st bot uses a simple but effective algorithm to explore through the maze and reach the goal [3]. It starts from the starting zone of the maze and scans through the maze for turn indicators. When it faces a turn indicator, if it’s a left turn indicator then it rises a left flag (as shown in Fig. 9) and if it’s a right turn indicator then right flag (as shown in Fig. 8). When the bot reaches a left turn junction and left flag is raised then it takes a right turn otherwise it goes straight and updates the field map. The same principle also used for right turn (as shown in Fig. 7 & 10). After passing each junction the 1st bot sends the last turn direction in the field map to the 2nd bot. The algorithm as follows:

Algorithm:

While goal not reached
    If a turn indicator is detected
        If it’s a left turn indicator
            Fag is equal left flag
        If it’s a right turn indicator
            Fag is equal right flag
    If a left turn junction is reached
        If it’s a left junction and flag is equal left flag
            Turn left
        Else
            Go straight
        If it’s a right junction and flag is equal right flag

Turn right
Else
Go straight
Update field map and send map to 2nd bot

The 2nd bot also starts from the starting zone and explore through the maze until it reaches a junction. Then it request for the turn direction of that junction from the 1st bot and turns accordingly the received turn direction [5]. The algorithm as follows:

Algorithm:
While goal not reached
If a turn junction is reached
Request the turn direction of that junction from 1st bot
Wait until turn direction is received
If received turn direction is left
Turn left
If received turn direction is right
Turn right
If received turn direction is straight
Go straight

5. RESULT

Our maze-solving bots were prototypes and used in the iARC Bangladesh round and international round. The 1st maze-solving bot was able to solve the maze in 95 seconds and second bot took 123 seconds, even though; the algorithm is not guaranteed to do so in all mazes.

6. CONCLUSION

Maze-solving involves Control Engineering and Artificial Intelligence. Using a good algorithm can achieve the high efficiency of finding the goal of the maze. The proposed maze-solving algorithm works better but our main hurdle was to develop a cluster system within which two bots can work simultaneously to solve a common problem and we successfully overcame this hurdle in this competition.

7. REFERENCES

[1] Huo Zhe Fu. “A Design and Realize for Micromouse’s Mazesolving”, Microcomputer Appfications.vol 9, pp59-62. 2008.
[2] Jin Liang Fang, Yi Jun Zhou. “Micormouse Based on ARM of IEEE Standard”, Machine Building and Automation.vol.5, pp99-101, 2008
[3] Kazuo Sugihara, John Smith. “Genetical algorithms for adaptive motion planning of IEEE Trans.USA: SIM, 19997. MicroMouse, California State University at Northridge, http//homepage.mac.com/SBenkovic/MicroMouse/index.html
[4] Dieguez A R,Sanz R,Lopez J. “Deliberative on-line local path plannning for autonomous mobile robot”,Journal of Intelligent and Robotic Systems, vol. 37, pp1-19, 2003.
[5] Tondra De, Dfew Hall. “The Inception of Chedda: A detailed design and analysis of Micromouse”, University of Nevada, Las Vegas Senior Design. vol. 5, pp39-44, 2004.
[6] www.ti.com/lit/ds/symlink/l293d.pdf
[7] http://www.atmel.com/devices/atmega2560.aspx