University Mohamed Khider of Biskra

Faculty *of Science and Technology*

Electrical Engineering Department

Final Year Project Report In View Of Obtaining The Diploma Of:

MASTER

Science And Technology

Embedded System

Theme:

Autonomous Mobile Robot

Presented And Supported By :

Wail Ikhenache

On: July 07,2019

Jury :

.M	HEZABRA Adel	MAA	University of biskra	**President**
.M	BENELMIR Okba	MCB	University of biskra	**Supervisor**
.M	RAHMANI Nacereddine	MAA	University of biskra	**Examiner**

Academic Year: 2020 - 2019

University Mohamed Khider of Biskra

Faculty *of Science and Technology*

Electrical Engineering Department

Final Year Project Report in view of obtaining
The Diploma of:

MASTER

Science and Technology
Embedded System

Theme:

Autonomous Mobile Robot

Presented by:	**Favorable opinion of the supervisor:**	**Signature**
Ikhenache wail	**BENELMIR OKBA**	.

Favorable opinion of the President of Jury:	**Signature**
HEZABRA Adel	.

Stamp and signature

.

University Mohamed Khider of Biskra

Faculty *of Science and Technology*

Electrical Engineering Department

Final Year Project Report in view of obtaining
The Diploma of:

MASTER

Science and Technology
Embedded System

Theme:

Autonomous Mobile Robot

Proposed and directed by : BENELMIR OKBA

Abstract (English and Arabic)

With the rise in navigation technology, autonomous robots are needed for mapping regions that are otherwise dangerous for manual exploration. In this project we build maze solver ,A maze solving robot is a self-contained full autonomous mobile robot, it can capable of transporting itself to the target of an unknown arbitrary two-dimensional maze. The individual components of maze solving robot system consist of the motor control, navigation sensor array, and a mapping system or algorithm for navigating the maze intelligently. The aim of this research is to develop and implement a maze solving robot and find the possible and short path from the starting point to the target.

في بعض الأماكن الخطرة التي يصعب على الإنسان استكشافها هنا يأتي دور الروبوت ذاتي التحكم . في مشروعنا بنينا روبوت متنقل على ثلاث عجلات يحل المتاهة , هو روبوت مستقل بالكامل يمكنه التنقل إلى الهدف في متاهة غير معروفة . يتحتم على الروبوت إيجاد طريقه عبر استكشاف المتاهة مرارا وتكرارا حتى الوصول إلى الهدف بعدها سيتعرف الروبوت على اقصر طريق لحل المتاهة وهذا بمساعدة مجموعة من الحساسات التي تجعل الروبوت مدرك لما يحيط به من عوائق .

Keywords: Collision Avoidance, Flood Fill Algorithm, Maze Solving Robot, Path Finding ,Navigation , Artificial Intelligence

Acknowledgement

First and foremost, I thank Allah the Almighty for this work would never be accomplished without the strength, wiliness and patience He granted me. All great praise goes to Him.

I am enormously thankful to the board of examiners namely: Mr. HEZABRA Adel and Mr. RAHMANI Nacereddine for reviewing my work.

I would like to express my deepest appreciation and gratitude to my warm and sympathetic supervisor Mr. BENELMIR Okba for his precious advice and guidance, and most importantly for his sense of understanding, encouragement, patience and wonderful support.

I also owe special thanks to my roommates, Hamdiken Iskander and Berroussi Zouhir and also to my best friend Kadir Ahmed, Karim Ismail, Lahcen Charif , who were behind the elaboration of this work.

Appreciation also goes to all the administrators and teachers at the Department of electronics in the University of Biskra for their hard work and dedication.

Finally, a heartfelt gratitude to my family and all my friends and colleagues for the encouragement and help they offered.

Dedication

" How to make a Man " .it Is a very short phrase of how my father Abdellatif Ikhenache raised me. He taught me everything I needed to know, helped me how to face life's difficulties ,but he knew when to let me fail to stand up again ; this is the most important lesson, and I know it was hard for him to do that in order to teach me. He always encourage me to try new things ,.I appreciate every lesson he has taught me, especially the hard ones.

My dad is someone who holds you when you cry ,guide you when you break the rules ,shines with pride when you succeed , and has faith in you even when you fail …

Dad thank you for telling me what I'm capable of ,For giving me the support that I needed to build a dream to chase after ,and for believing that I have the talent to reach my goal .

Mom ,thank you for making me realize that I worth everything in this world, That I must be treated as king , and that I should never satisfy for less than what I deserve .

My sisters i just want to tell you I can't imagine my life without you Faten, jihan, Rahma, Malak thank you .

Ben lembarek Bouchra Thanks a lot you are the most precious thing I have

List Of Content

II Embedded System Of Robotics

III Generalities Artificial Intelligence

List Of Figure

I Generalities About Robotics

II Embedded System Of Robotics

III Generalities Artificial Intelligence

IV Third Lab Session

IV Fourth Lab Session

List Of Table

General Introduction

General Introduction

We all used to be fascinated when watching robots on TV whether they were big or small, made to be used in wars or at home. We always asked ourselves and wondered how these robots were made, and how they could do complex movements and processes. After we joined the world of electronics and programming, it became clear to me that all these questions and all those machines that we thought were complex are, in fact, a combination of electronics, programming and mechanics.

In our work, we will go through the presentation of a research project which is no less complicated than any works we were one-day curious about. This research project is about a robot that solves a maze. When you read this, you will have a lot of questions: how, what and why? But we will remove all this confusion in the following chapters of the research project. By in the end, it will be clear to you and hopefully understand the functioning principles of this robot, as well as its basics.

In the first chapter, we will begin with the definition of the field of robotics and robots. Also, we will mention its types, areas of presence, and the design of its movements so that we can choose the right robot for our application, which is a maze solution. We seek to know if it is possible for our robot to actually see the maze and solve it. We will know all about this in the next chapters, so no need to rush. Moreover, the robot interacts with the surrounding area and is equipped with sensors that function as sensory receptors that sense its surroundings. The robot works using "its brain", which is a "Micro-controller", for processing and receiving signals from those sensors, which in turn give orders to certain triggers in the robot to make a response. We will get to know all of these steps in detail in the second chapter where we will talk about the field of " embedded systems" .

We strive for excellence and challenge ourselves to do more, after the robot has solved the maze of course. Can we make our robot recognize the shortest route from all the routes in the maze to solve it? .The first stage is the discovery of the maze, and the final stage is finding the shortest way to solve the maze. Will we be able to achieve this achievement? We'll figure this out together in the third chapter in which we talk about artificial intelligence, multiple algorithms, their types in general, and we mentioned in particular the practices used in the study of mazes. In the end, which is the fourth chapter, we know that many theoretical theories and scientific research will not be useful for the student unless it is tested and applied in real life. Therefore, we chose in our research project the method of labs .to display the last chapter which consists of four (4) "labs", and each one contains a theoretical and applied study at the same time. All of these "labs" are only stages that ultimately enable a robot to solve a maze.

Chapter I

Generalities About Robotics

I.1. Introduction

In the last two decades, various technologies applicable to mobile robots in general and robots in particular have made considerable progress.

Robotics has come a long way, in particular for mobile robots, a similar trend happening as we have considered for computer systems. The transition from mainframe computing through workstations to PCs, which will in all likelihood proceed with handheld units for many applications.

In the past, cell robots were controlled by way of heavy, large, and highly-priced PC systems that should no longer be carried and had to be linked with the aid of cable or wireless devices. Today, however, we can build small mobile robots, with numerous actuators and sensors that are controlled through inexpensive, small, and mild embedded laptop systems that are carried on-board the robot.

There has been a top notch increase of interest in cellular robots. Not simply as interesting toys or inspired by using science fiction stories or movies , however as a ideal device for engineering education, cell robots are used two nowadays at nearly all universities in undergraduate and graduate publications in Computer Science/Computer Engineering, Information Technology, Cybernetics, Electrical Engineering, Mechanical Engineering, and Mechatronics [1][2] .

I.2. Robots Definition

The term robot is derived from Czech word “robota” which capability work, or forced work. A robot is an shrewd computer that can engage with the surroundings to operate particular duties to decrease human effort. Various types of robotic structures are available; however, the majority of robots have some frequent features.

Almost all robots have a movable body, some of them have motorized wheels, whereas others have many small movable segments that are typically made of plastic, wood, or metal. In some robots, countless joints join the man or woman segments of the robotic together. The actuator of the robotic spins the joint with wheels or a pivot segment [3][4] .

Robots are categorized into numerous types based totally on the systems they use: (1) robots that use electric motors as actuators, (2) robots that use a hydraulic system, (3) robots that use a pneumatic device that is driven by compressed gases, and (4) robots that use all of these actuator types. Generally, any robotic system requires a strength source to drive its actuators. Most robots have either a battery or different electricity sources. Hydraulic robots on the whole require a pumping device to pressurize the hydraulic fluid, and pneumatic robots primarily want air compressors or compressed air tanks. In most cases, a microcontroller is used as the Genius of a robot, which is every so often referred to as a microcomputer. All the actuator and circuits are immediately related to microcomputers by way of some interface systems. Another common characteristic is that most robots are programmable; thus, a robot's behavior can be modified by using writing a new program in its microcomputer [4].

Figure I.1.Driven Walking Robot [5]

I.3. Different Branches In The Development Of Robotics

Robotics in contrast to other branches is a moderately new area of engineering. It is a multi-disciplinary domain. The specific branches in the development of Robotics are:

1. **Mechanical Engineering:** Deals with the machinery & structure of the Robots.
2. **Electrical Engineering:** Deals with the controlling & intelligence (sensing) of Robots.
3. **Computer Engineering:** Deals with the movement development and observation of Robots [6].

I.4. Classification Of Robots

Robots are labeled depending upon the circuits of the Robots and the range of application it can perform. The robots are categorized into three types:

I.4.1. Autonomous Robots: A robot is an autonomous device which exists in the bodily world, can experience its environment, and can act on it to gain some dreams .Autonomy is another significant component that determines the complexity of cellular robot systems. Autonomy refers to the functionality of the device to operate in a real-world environment except any external interference for extended durations of time .They have pre-programmed instructions and are given a choice of taking selections based totally on conditions and surroundings. An Artificially smart robotic can even analyze positive behaviors and act accordingly [7][4][8] .

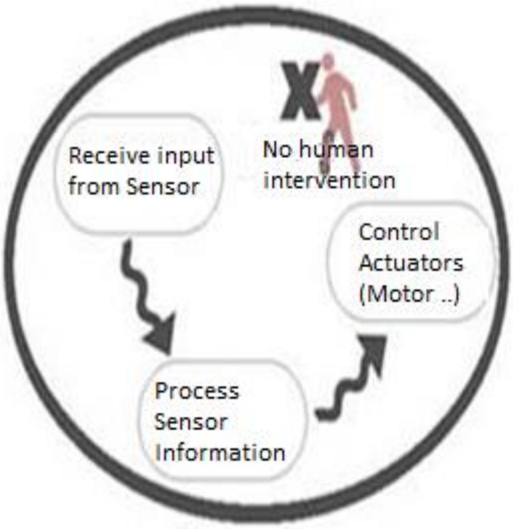


Figure I.2. Autonomous Robots [8]

I.4.2.Controlled Robots: These are robots that require human intervention to accomplish a task. They can either be wire controlled or remote controlled that are guided to perform any variety of complex activities. A remote controlled robotic can be programmed and guided to perform dangerous and complicated duties without being on the spot [8].

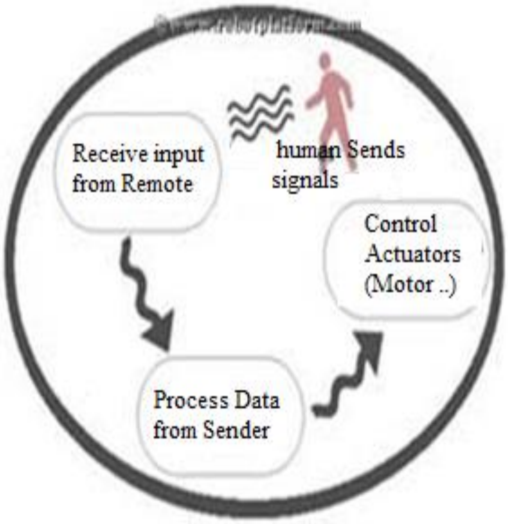


Figure I.3.Controlled Robot [8]

I.4.3.Semi-Autonomous Robots: These robots take the best of each worlds. The brain constructed in helps them perform easy duties and take simple decisions. For complicated duties however, human intervention may additionally be required. Generally the program is designed to take smart decisions on its own until any human input [8].

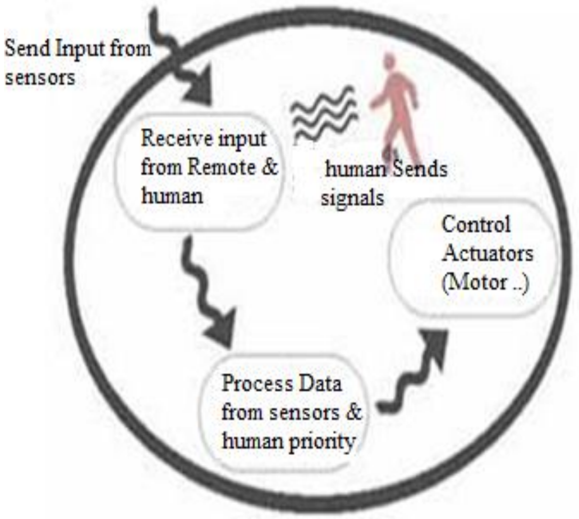


Figure I.4.Semi-Autonomous Robots [8]

I.5.Types Of Robotics

Several types and domains of robotics are concerned in modern day world. They are categorized based on the services they provide :

I.5.1.Industrial Robots: Industrial robots are normally fixed manipulators which perform in a variety of working environments. They function a number general-purpose tasks like Welding, Painting, machining, etc... Robots in the production line are precise examples of such a robotics system. An industrial conveyor belt can also be treated as an industrial robotic that strikes countless products and components from one region of an industry to another [9][10].

Figure I.5.ABB Industrial Robot [10]

I.5.2.Medical Robots: There has been an increasing use of robots in the scientific discipline for surgery, rehabilitation and training. In robotic surgery, the doctor controls the robotic arm using a very small tool that is attached to the robotic arm via a control computer. In such case, the general practitioner makes small cuts to insert the units into the human body. A skinny tube attached to the camera in the front is used to capture the enlarged real-time picture of the phase of the body the place the surgical procedure will take place. The robot matches the movement of the hand of the surgeon, and the software of the laptop tunes the precision two of the movement of the hand such that the surgical procedure will become especially accurate .Medical robots are now not supposed to replace the surgeons however serve as a surgical assistant to the surgeon [9][10] .

Figure I.6.A Robot Performing Surgery [10]

I.5.3.Military Robots: Military robots are developed to be used with the aid of the armed forces such as for searching, search and destroy, rescue, and surveillance operations; bomb disposal robots, transportation robots and reconnaissance drones. Equipped with infrared sensors, these robots react more hastily than humans in emergency and hazardous situations, they are regularly called artificial soldiers [10][9] .

Figure I.7.Foster-Miller TALON [10]

I.5.4.Space Robots: With the advent of robotic technologies, exploration of various celestial bodies has been a reality. Tasks like space manipulation, surface mobility and scientific experiments are performed by space robots. Research on space robotics primarily focuses on two areas of interests: (1) orbital robotics and (2) planetary rovers. Orbital robotics has an interest in the research domain of manipulation and mobility for scenarios such as international space stations and satellite servicing [9][10] .

Figure I.8.Aeryon Scout [10]

I.5.5.Entertainment Robots: These robots are used in quite a number entertainment locations like enjoyment parks, pleasure rides, sports, entertainment, and occasionally home service. Entertainment robots are broadly considered in the context of media and arts, where artists employ superior technologies to create the environment and creative expressions to permit the sensors and actuators to react in accordance to the modifications associated to the viewer. Being incredibly low cost and mass produced, amusement robots are used as mechanical and every now and then interactive toys that can perform various hints and take countless commands [9][10] .

Figure I.9.AIBO, The Entertainment Robot [10]

I.5.6.Robot Locomotion : Locomotion, the capability of a physique to go from one vicinity to another, A mobile robotic desires locomotion mechanisms that allow it to cross unbounded at some stage in its environment. But there are a massive range of feasible approaches to move, and so the determination of a robot's strategy to locomotion is an necessary issue of mobile robotic design. In the laboratory, there are research robots that can walk, jump, run, slide, skate, swim, fly, and, of course, roll. Most of these locomotion mechanisms have been inspired with the aid of their organic counterparts [5][11] .

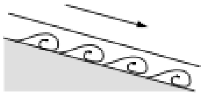
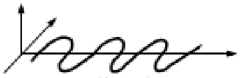
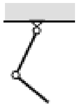
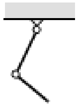
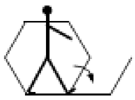
Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	 Longitudinal vibration
Sliding 	Friction forces	 Transverse vibration
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon (see figure 2.2) 

Figure I.10.Locomotion Mechanisms Used In Biological Systems [5]

I.6.Driving Robots

I.6.1. Legged Mobile Robots: Legged locomotion is characterized via a sequence of point contacts between the robotic and the ground. The key blessings include adaptability and maneuverability in rough terrain. Because solely a set of factor contacts is required, the fine of the ground between these factors does no longer remember so lengthy as the robot can preserve adequate ground clearance. In addition, a strolling robot is capable of crossing a gap or chasm so long as its attain exceeds the width of the hole. A remaining benefit of legged locomotion is the conceivable to manipulate objects in the environment with great skill. To make a legged robotic cell each leg ought to have at least two tiers of freedom (DOF). For every DOF one joint is needed, which is normally powered via one servo. Because of this a four legged robot needs at least eight servos to tour around. Figure I.11 indicates the strength consumption of exceptional locomotion concepts. It strikes that the energy consumption of legged locomotion is almost two orders of magnitude greater inefficient than of wheeled locomotion on hard, flat floor (e.g. railway wheel on steel). One motive for this is that wheeled locomotion requires in everyday fewer motors than legged locomotion. But precisely the single set of point contacts presents one of the most complex problem in legged locomotion, the stability problem [5][12].

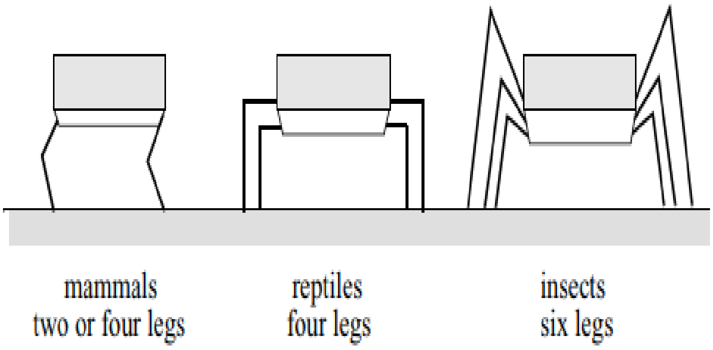


Figure I.11.Arrangement Of The Legs Of Various Animals [11]

- **Stability**: Means an equilibrium which can be measured and observed. is of route a very necessary difficulty of a robot, because it no longer overturn. Stability can be divided into the static and dynamic stability criterion.
  - ✓ **Static Stability** : Means that the robot is stable with no need of motion at every moment of time. A statically stable robot is well balanced and does not fall over when standing. This means that the center of gravity of a robot is within its ground contact base .Static stability is explained by an easy example: Figure I.12 shows a stool with three legs [12][13] .

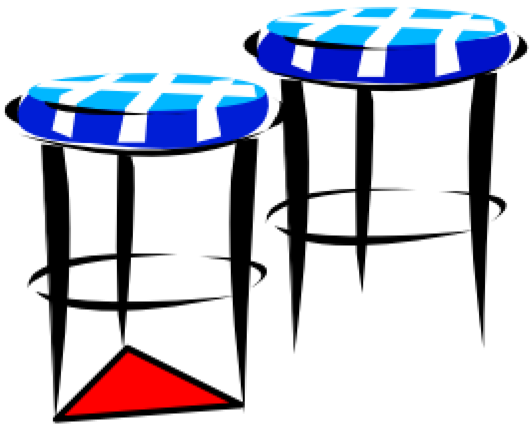


Figure I.12.Example Of A Stool [12]

  - ✓ **Dynamic Stability**: Is where stability is accomplished on movement. Imagine a robot with solely one leg which hops from one location to another and is stable except it maintains moving, but falls over if it stops moving[13] .

- **One Leg**: The minimal number of legs a legged robot can have is, of course, one. Minimizing the number of legs is recommended for a number of reasons. Body mass is in particular necessary to taking walks machines, and the single leg minimizes cumulative leg mass [5].

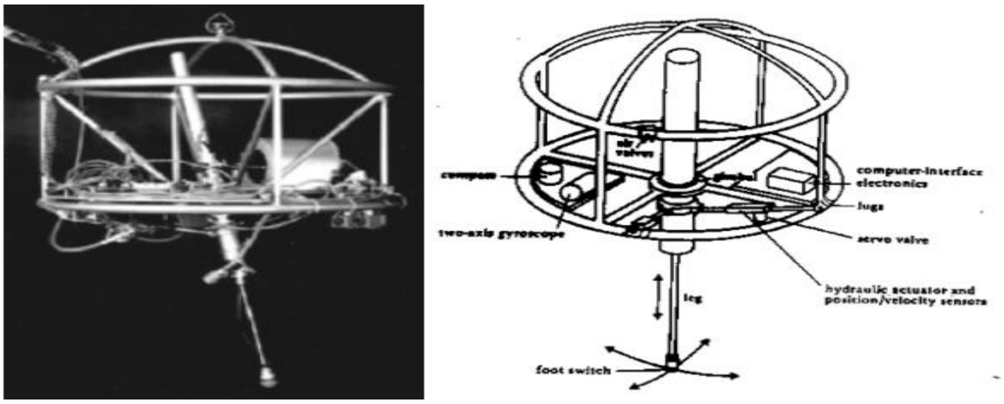


Figure I.13.One Leg Hopper [5]

- **Two Legs**: Two legged robots are already capable to walk, run, jump, dance and travel up and down stairs, but steadiness is nonetheless a problem for bipedal robots, due to the fact they have to be dynamically steady [12] .

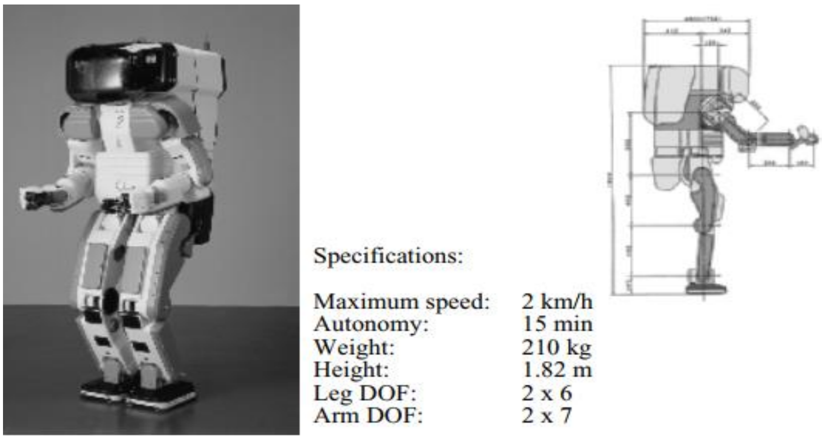


Figure I.14.The Humanoid Robot P2 From Honda, Japan [5]

- **Four Legs**: Walking with four legs is common for most animals and there is an accurate cause to replicate this in robots. Although standing still on four legs is passively stable, on foot stays challenging because to stay stable the robot’s center of gravity should be actively shifted throughout the gait [13][5].

Figure I.15.Aibo (Sony) [12]

- **Six Legs:** More variety of legs presents larger stability. Six legged locomotion is the most popular legged locomotion notion due to the fact of the capacity of static stable walking. The most used static steady gait is the tripod gait, the place each instances the two exterior legs on the one side and the internal leg of the other side are moved together. Due to the possibility of static secure gaits the control complexity is reduced on the one hand, because there is no difficulty of stability manage in general, however on the other hand most six legged robots legs have three ranges of freedom and six legs have to be controlled, so leg coordination turns into greater complex. Six legged robots are regularly stimulated by using nature [13][12] .

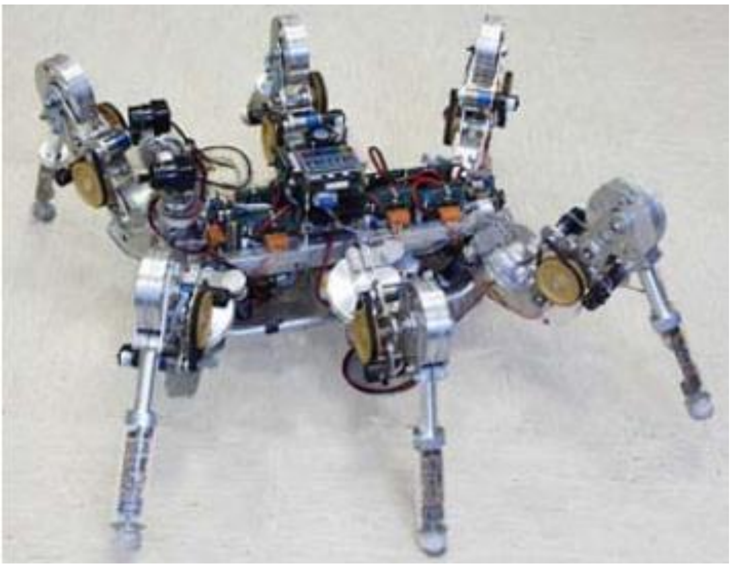


Figure I.16.Lauron [12]

I.6.2 .Tracked Robots: A tracked robot can be viewed as a specific case of a wheeled robot with differential drive. In fact, the only difference is the robot's better maneuverability in tough terrain and its higher friction in turns, due to its tracks and multiple factors of contact with the surface. Tracked robots principally use Skid steer drive which is a modified thought of differential drive. A tracked robotic has two tracks attached at both side of the chassis driven with the aid of two separate motors. They are advised with the aid of transferring these tracks at exclusive speeds in the same/opposite direction. Most of the time these tracked robots skid to change their direction. This makes the threads/tracks to wear off shortly and also they require large vicinity to turn as they slide their entire body in opposition to the ground [14][2] **.**

Figure I.17.EyeTrack Robot [2]

I.6.3.Wheeled Mobile Robots: The most popular locomotion mechanism in man made automobiles is wheeled locomotion; so it is not surprising that it is frequently used in mobile robotics. Reasons for this are the handy mechanical implementation of the wheel, In addition, balance is no longer usually a research trouble in wheeled robot designs, due to the fact wheeled robots are almost constantly designed so that all wheels are in floor contact at all times. Thus, three wheels are sufficient to assurance secure balance, although, as we shall see below, two-wheeled robots can additionally be stable. When greater than three wheels are used, a suspension gadget is required to enable all wheels to maintain floor contact when the robot encounters uneven terrain [12][5].

I.6.3.1.Wheel Types: The starting point of thinking about wheeled locomotion is the wheel itself. In regularly occurring there are 4 major classes of wheels as shown in Figure I.22 .(a) two shows the well-known wheel with two levels of freedom, these are rotation around the wheel axle and around the contact.(b) suggests the castor wheel with two degrees of freedom, rotation around the wheel axle and the offset steering joint. (c) indicates the Swedish 45° and Swedish 90° or omni wheel, which has three tiers of freedom: rotation round the contact point, around the wheel axle and around the rollers.(d) suggests the Ball or spherical wheel, this wheel is omni directional, but it is technically hard to implement [12].

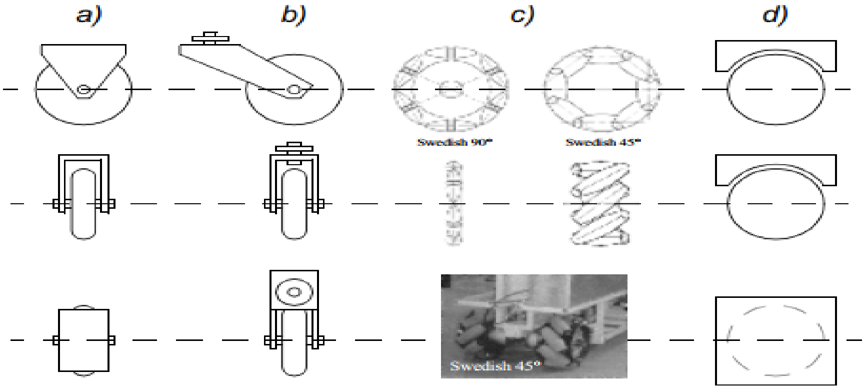
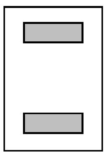
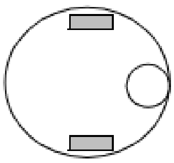
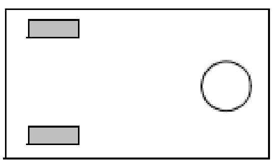
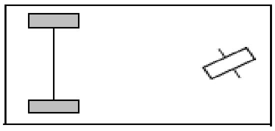
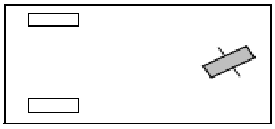
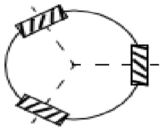
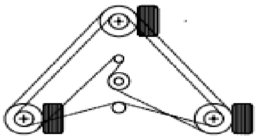
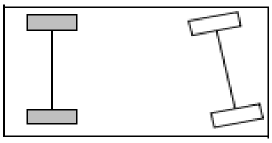
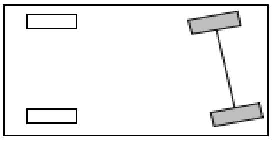
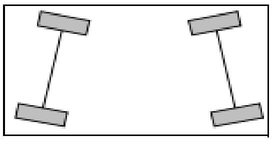
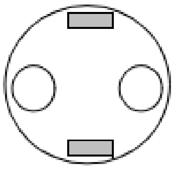
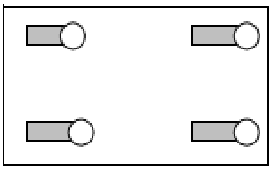
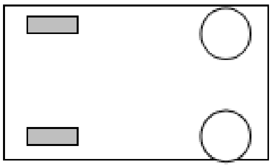
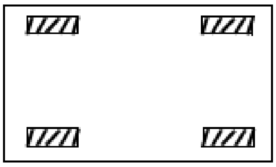
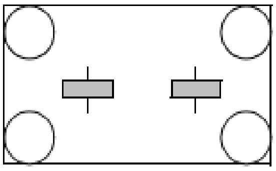
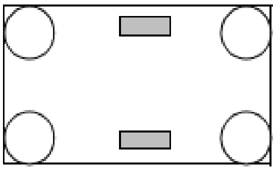


Figure I.18.The Four Basic Wheel Types[5]

I.6.3.2.Wheel Configurations for Rolling Vehicles:

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice

		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
6		Two motorized and steered wheels aligned in center, 1 omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, 1 omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)

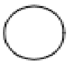
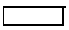
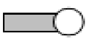
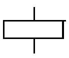
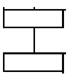
Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

Table I.1.Wheel Configurations For Rolling Vehicles [5]

I.6.3.3.Wheel Control Theory: There are various types of wheels and their arrangements. Once the number of how many to use is determined and what type of wheels the robot will have. Below shows few mechanisms to power and steer your robot [15].

❖ **Differential Drive:** Differential wheeled robot can have two independently driven wheels fixed on a common horizontal axis or three wheels where two independently driven wheels and a roller ball or a castor attached to maintain equilibrium. Here are three fundamental cases which can happen in a differential wheeled robot:

- If both the wheels are driven at the identical velocity and same path (either clockwise or anticlockwise) then the robot tends to spin around its vertical axis

- If both the wheels are pushed in the identical velocity however in the opposite direction (One clockwise and other anticlockwise) then the robotic is more in all likelihood to observe a linear path, either forward or backward based on the motors spin

- If the angular velocities are distinct in terms of values, the robotic makes a curve motion. Lastly, if one of the wheels rotate and the different stays nonetheless then the robotic nearly makes a 90° turn. Manipulating the drive velocity and path can give some interesting power paths [15].

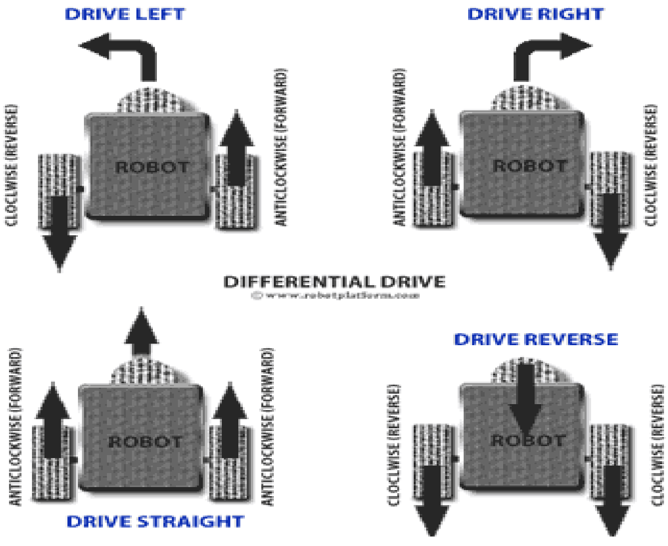


Figure I.19 Differential Drive[15]

❖ **Skid Steering**: Is another driving mechanism implemented on vehicles with either tracks or wheels which uses differential drive concept. Most common Skid steered vehicles are tracked tanks and bulldozers [15].

- Some of the advantages of skid steer drive are:

  1. They have greater traction and especially good for rough terrain
  2. Same concept can be used on both tracked robots and wheeled robots
  3. Since there are no explicit steering wheels, steering mechanism is not required
  4. Since all wheels on each side drive in the same direction, only two motors are enough for driving and steering the robot
  5. This method does not require castor wheels and hence eliminates the problems caused by castors

- Few drawbacks of Skid steering are:
  1. Since it uses skidding or slipping technique, increases wheel / track wear reducing life

     Like differentially driven robots, driving in a straight path is a hard to achieve task as both the motors are expected to drive at exactly the same speed. This can still be taken care by other sensing devices, but adds cost and control mechanisms [15] .

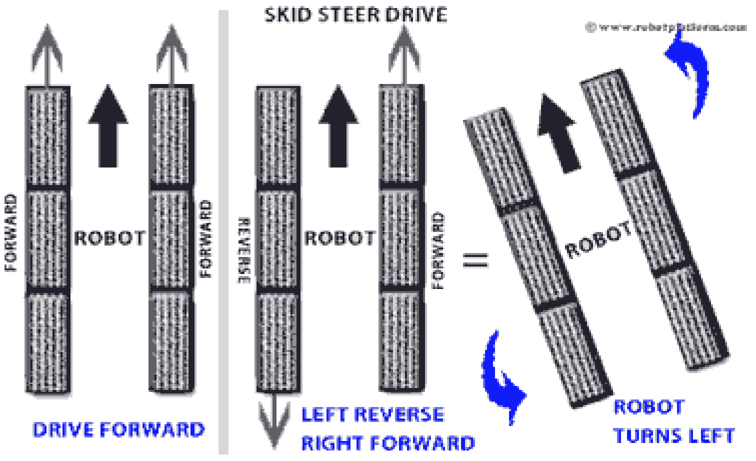


Figure I.20 Skid steering[15]

❖ **Tricycle Drive**: Tricycle robot is designed with a the front steering wheel managed by a motor. The downside is that they can't spin like a differential drive robot and does no longer have a ninety turn due to their constrained radius of curvature. Few robots have both steer and drive managed by way of the front wheel and the rear wheels act as helping wheels to preserve equilibrium[15] .

Figure I.21 Tricycle Drive [15]

❖ **Ackermann Steering:** One of the most frequent configurations observed in cars is Ackerman steering which automatically coordinates the perspective of two front wheels which are constant on a common axle used for guidance and two rear wheels constant on every other axle for driving. This would possibly resemble the tricycle strategy the place the front wheel is replaced with two wheels and an axle. But when two wheels are connected to a tricycle graph axle-articulated drive), then turning causes the robotic to skid. To overcome that downside Ackerman steerage is designed in such a way that when there is a turn, the inner tire turns with a larger perspective than the outer tire and avoids tire slippage. Although there are disadvantages, the draw back is extra parts required; no zero radii flip and extended complexity in design [15].

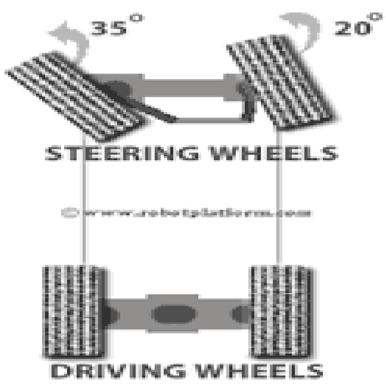


Figure I.22.Ackermann Steering [15]

- **Omni Directional Drive**: Since Omni wheels have smaller wheels attached perpendicular to the circumference of another bigger wheel, they enable wheels to move in any course instantly. In other words, they are Holonomic robots and can pass in any route barring changing the orientation. Omni wheels are reachable in many one-of-a-kind sorts categorized on their size, diameter, design, measurement of smaller wheels attached etc [15].

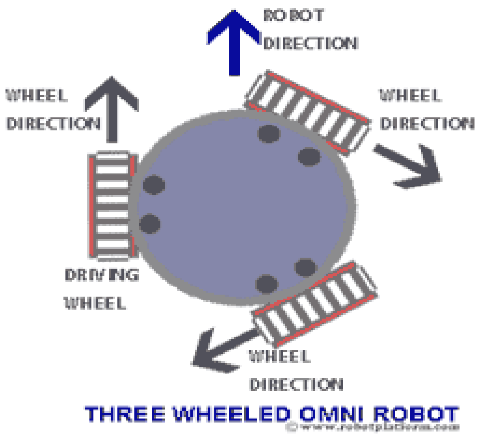


Figure I.23.Wheel design [15]

I.6.4. Aquatic Robots: This type of robots are designed for underwater monitoring operations, which include pollution detection, video mapping and other tasks. This constraint definitely applies to the small-scale swimming robot, where multiple small actuators are wanted for forward/backward propulsion, guidance and diving/surfacing. there are only a few correctly built aquatic robots. Making a robotic to work in water typically ends up in undesirable failures [16][17] .

The first challenge in creating an aquatic robot is to make all the electronics waterproof. There are many ways by which electronics can be made waterproof. Some of the basic ideas are listed below:

1. Enclosing all the circuits and electronics in an air-tight container
2. Enclosing the circuits in oil filled containers which resists water inflow
3. Using hot-glue all over the circuits and then submerging them

Differential drive technique works best on these aquatic robots and driving force can be tails, fins, thrusters, wings, thrusters, paddles, paddle wheels, air pumps etc., based on your robot design. Aquatic robots include robots that sail, submerge or crawl under water [17] .

Figure I. 24.Autonomous submarine USAL [2]

I.6.5.Flying Robots: Aerial robotics serves as a platform for the next generation that consists of a robotic concept emphasized from a small micro aerial vehicle to a large fixed wing multi rotor drone. The application of such a platform widely depends upon the utilization. Starting from aerial surveillance to traffic monitoring and geological and weather survey in agriculture, there is a wide range of applications for aerial robotic [10] .

- **Advantages**

1. First and foremost, flying robots are fun to build and fly
2. Experiments on Remote controlled robots are going on for years. This means there is a huge community which can help you if you like to build one.
3. These robots are extremely useful as surveillance robots

- **Disadvantages**

1. Building a flying robot is expensive when compared to purchasing one
2. Requires extensive research before designing and building one (and even then it might fail)
3. A minor crash can break parts and make your robot irreparable. The parts required to build these flying robots are very expensive making a crash even worst [10][18].

Figure I.25.Predator [19]

I.6.6.Robotic Arm: Robot arms can either be autonomous or remote controlled which are the most common ones, manufactured and used across industries). A typical robot arm has six degrees of freedom which are made by seven metal sections joined by six joints. Robot arms are used because they are fast, precise and do much more work compared to a human being. For example, a Cartesian robot is used for assembly operations, Spherical robot is used for handling machine tools, and articulated robot arms are used for welding, spray painting and other activities [18] .

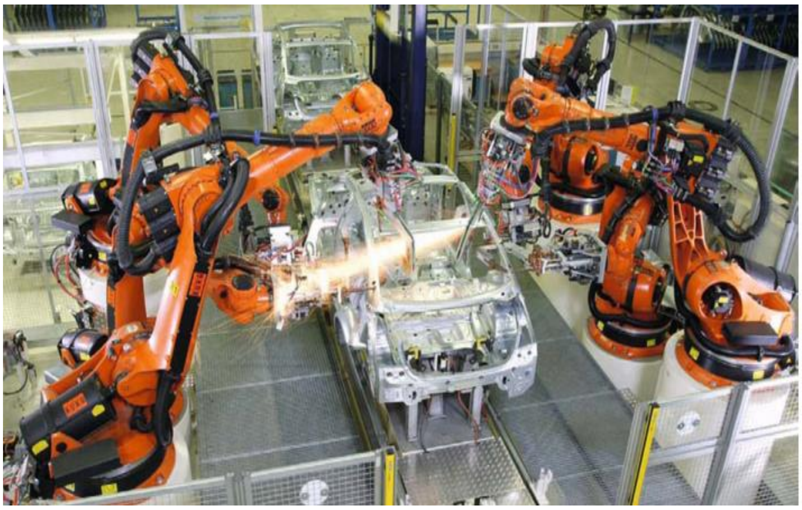


Figure I.26**.** Industrial Application And Individual Manipulator [2]

I.6.7. Hybrid robots: Are those which are designed through combining the characteristics of two or greater robots. For example a hovercraft robotic can be designed to traverse both on land or in water. Another robot would possibly be able to stroll underneath water and additionally swim. A third one would possibly fly and crawl. The mixtures are considerable and it is left to you to combine and create wonders [18] .

Figure I.27. Hybrid Robots [20]

I.7.Conclusion

This chapter contains generalities about robotics and how it works, The relationship between the size and design of the robot and the work and ability of the. It provides also an overview about mobile robots and its types especially of land-based locomotion (wheeled, legged and chained robots), finally presented the drive mechanism which is the differential drive kinematic that I used in my project. The next chapter presents embedded system field and the robot hardware .

Chapter II

Embedded System Of Robotics

II.1.Introduction

In technology ,software and hardware are deeply interrelated. A computer machine is a aggregate of the features of various digital gadgets that act collaboratively with the help of software systems. Computers discover their way into each realm of activity. Some are developed to be as powerful as possible, barring challenge for price, for high-powered purposes in industry and research. Others are designed for the home and office, much less powerful but also less costly. Another class of computer is little recognized, partly because it is little seen. This is the type of computer that is designed into a product, in order to grant its control. The PC is hidden from view, such that the user frequently doesn't know it's even there. This kind of product is referred to as an embedded system [10][21] .

II.2.What is an Embedded system

An embedded device is a aggregate of computer hardware and software—and perhaps extra parts, either mechanical or electronic—designed to function a devoted function. If we take any engineering product that wishes control, and if a PC is incorporated within that product to undertake the control, then we have an embedded system. An embedded system can be described as a gadget whose predominant feature is now not computational, however which is managed by using a computer embedded inside it. These days embedded structures are everywhere, acting in the home, office, factory, auto or hospital [22][21] .

II.3.Common System Components

Any embedded system. Fundamentally, it have to be able to operate arithmetic or logical calculations. This characteristic is supplied by way of the Central Processing Unit (CPU). It operates by using working via a series of instructions, known as a program. Any one of these directions performs a very easy function. However, due to the fact the traditional PC runs so fairly fast, the usual effect is one of very incredible computational power. Many instructions cause mathematical and logical operations to occur. These take vicinity in a part of the CPU referred to as the ALU, the Arithmetic Logic Unit.

In order to have program, there have to be a area to save the executable code and brief storage for run time records manipulation. These take the structure of read-only memory (ROM) and random get entry to reminiscence (RAM), respectively; most embedded systems have some of each. If only a small amount of reminiscence is required, it might be contained within the same chip as the processor. To be of any use the computer need to be able to talk with the outdoor world, and it does this via its input/output in Figure II.1 A generic embedded system [21][22].

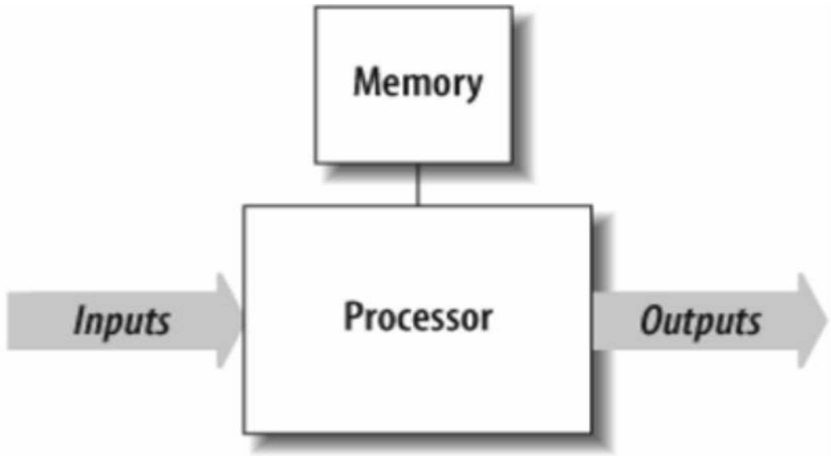


Figure II.1.A Generic Embedded System[22]

II.4. Classifications Of Embedded Systems

Typically, embedded devices can be categorized into quite a few classes. Such classifications are based totally on the processing power, cost, functionality, and architecture. The typical classifications are three as follows.

1. Small-scale embedded system: A small-scale embedded machine is primarily based totally on either 8- or 16-bit architecture . It typically runs on 5 V battery power, having restrained processing power and memory. It typically makes use of small-size flash memory or electrically erasable programmable read- only reminiscence (EEPROM) to keep packages and instructions[10].

2. Medium-scale embedded system: This kind of machine is mainly used for the purpose of digital signal processing (DSP). Mostly, 16±32-bit processor architecture is used in such systems. This machine helps complex software program and hardware layout and needs integrated improvement environments, debuggers, simulators, and code engineering equipment to install and analyze the software. Reduced education set computing (RISC) is the most preferable architecture in such a case, superior RISC laptop (ARM)-based smart records acquisition systems, and automation structures are examples of such embedded systems [10].

3. Sophisticated embedded systems: This kind of embedded device has excessive hardware and software program configuration. Both complex education set computing (CISC) and RISC architectures are supported by such systems. Most of these structures have greater random-access reminiscence (RAM) configuration. It supports the system-on-chip (SOC) concept. The software that runs on embedded structures is frequently a real-time working machine (RTOS) that helps the TCP/IP network protocol. More high-end functions such as high-definition graphics-based media [10].

II.5. Embedded Systems And Robotics

Embedded systems and robotics are the most interrelated phrases in this present day technological era. The revolution of clever phone, wise authentic time working device (RTOS), and system-on-chip technological expertise offers a new dimension to the embedded hardware. In the past, the embedded device used to be a bit problematic to control and a large chunk of assembly-level code used to be to be written to utility the whole system. But as matters preserve altering fantastically drastically, in modern times embedded constructions act as a platform in the development of software/firmware, therefore lowering the improvement time. The shape of the computer moreover keeps altering day by means of using day so as to extend processing electricity and to reduce strength consumption. The enhancement of the RTOS-like Android gives every other new dimension to embedded systems. On the contrary, robotics has evolved to higher-dimensional applications. In formerly years, robots had been only used in industrial and scientific research, however these days robotics has reached a new dimension [10] .

II.6. Definition Of Arduino

Arduino in Figure II.2 is the name of an open-source electronics platform constructed around microcontrollers that are functionally similar to a small computer. Arduino lets you connect the physical world to the world of computers. The thought behind the Arduino board was once no longer unique in any way. On the contrary, there have been different very comparable boards on hand at the time; however what made Arduino special was the Arduino team's approach to the project. The first board was released beneath an open-source licensing model, which was very distinguished for hardware at the time.

Open-source licensing skill that the plan of the board was available for absolutely everyone to copy, reproduce, and adjust in any way. Most technological know-how corporations make their cash creating hardware; they don't inform anybody how they make their products and they take out patents to stop others from copying [23][24].

Figure II.2. Arduino Uno [25]

II.6.1.The Arduino Hardware : There have been a number of Arduino versions, all based on an 8-bit Atmel AVR reduced instruction set computer (RISC) microprocessor. The first board was based on the ATmega8 running at a clock speed of 16 MHz with 8 KB flash memory; later boards such as the Arduino NG plus and the Diecimila (Italian for 10,000) used the ATmega168 with 16 KB flash memory. The most recent Arduino versions, Duemilanove and Uno, use the ATmega328 with 32 KB flash memory and can switch automatically between USB and DC power. For projects requiring more I/O and memory, there's the Arduino Mega1280 with 128 KB memory or the more recent Arduino Mega2560 with 256 KB memory. The boards have 14 digital pins, each of which can be set as either an input or output, and six analog inputs. In addition, six of the digital pins can be programmed to provide a pulse width modulation (PWM) analog output. A variety of communication protocols are available, including serial, serial peripheral interface bus (SPI), and I2C/ TWI. Included on each board as standard features are an in-circuit serial programming (ICSP) header and reset button the pin description of arduino uno as the following figure [26].

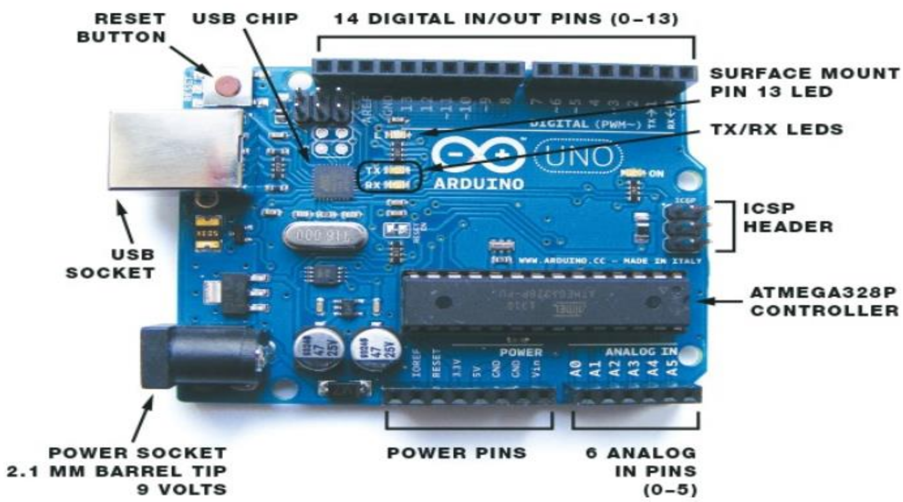


Figure II.3. Arduino Uno Component [27]

II.6. 2.Arduino Software : Software programs, known as sketches, are created on a computer using the Arduino built-in development environment (IDE) as the following figure . The IDE enables you to write and edit code and convert this code into instructions that Arduino hardware understands. The IDE additionally transfers these directions to the Arduino board (a technique called uploading). When you open the Arduino IDE, it ought to seem similar to Figure II.4 two The IDE screen is divided into a toolbar at the pinnacle with buttons for the most normally used functions; the diagram window in the center, the place you'll write or view your programs; and the Serial Output window at the bottom. The Serial Output window displays communication messages between your PC and the Arduino, and also lists any blunders if your layout doesn't assemble properly [28][27].

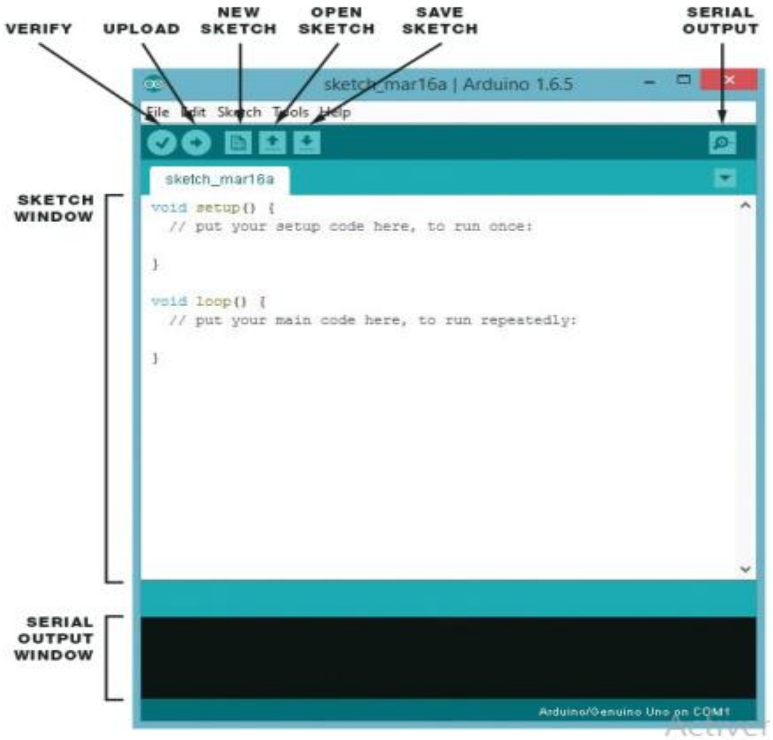


Figure II.4 Arduino Uno Component [27]

II.7. Robot And Sensors

One of the most important tasks of an autonomous system of any kind is to acquire knowledge about its environment. This is done by taking measurements using various sensors and then extracting meaningful information from those measurements. A wide variety of sensors is used in mobile robots (Figure II.2). Some sensors are used to measure simple values such as the internal temperature of a robot's electronics or the rotational speed of the motors. Other more sophisticated sensors can be used to acquire information about the robot's environment or even to measure directly a robot's global position in the following figure example of robot [5].

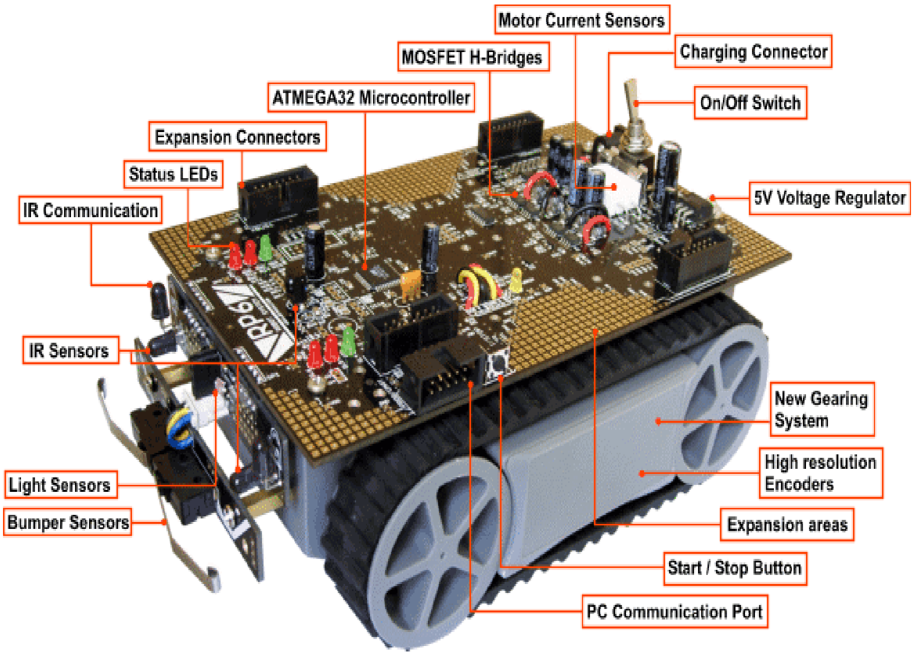


Figure II.5.Autonomous-Robot [29]

II.8. Sensor Classification

Classify sensors using two important functional axes Proprioceptive /exteroceptive and passive/active. Proprioceptive sensors measure values internal to the system (robot), for example, motor speed, wheel load, robot arm joint angles, and battery voltage. Exteroceptive sensors acquire information from the robot's environment, for example ,distance measurements, light intensity, and sound amplitude. Hence exteroceptive sensor measurements are interpreted by the robot in order to extract meaningful environmental features. Passive sensors measure ambient environmental energy entering the sensor. Examples of passive sensors include temperature probes, microphones, and CCD or CMOS cameras .Active sensors emit energy into the environment, then measure the environmental reaction. Because active sensors can manage more controlled interactions with the environment, they often achieve superior performance. However, active sensing introduces several risks: the outbound energy may affect the very characteristics that the sensor is attempting to measure. Furthermore, an active sensor may suffer from interference between its signal and those beyond its control. For example, signals emitted by other nearby robots, or similar sensors on the same robot, may influence the resulting measurements. Examples of active sensors include wheel quadrature encoders, ultrasonic sensors, and laser rangefinders [5].

II.8.1 Analog Sensor VS Digital Sensors

The output generated from sensors can be either analog signals or digital signals.

II.8.1.1 Analog Sensors: Analog Sensors output like figure II.3 a change in electrical property to signify a change in its environment. The change can be a variation in Voltage, Current, Resistance, Charge and Capacitance. Sensor circuits are designed to monitor these changes and provide a voltage difference. This voltage difference, if required can be converted into a digital value and processed further. All modern microcontrollers have Analog to Digital converter circuitry built-in. For example, if we consider a Photo-resistor, the resistance in a Photo-resistor changes with the amount of light falling on it. The Photo-resistor circuitry creates a voltage difference based on the change in resistance and an analog signal is fed into the microcontroller. This analog signal, if required can be further converted into a digital value and processed as per the requirement. Since most microcontrollers work within the 0V to +5V range, the sensor circuitry is designed such that it generates a continuous signal between 0 Volts to +5 Volts as an output [30].

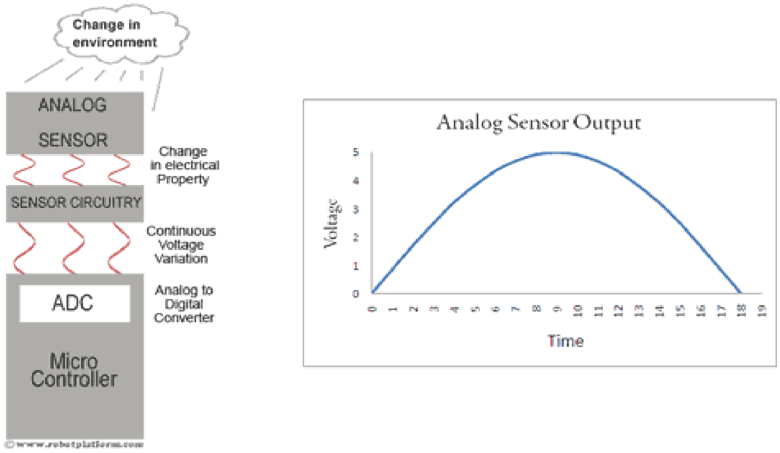


Figure II.6.Analog Sensor [30]

II.8.1.2. Digital Sensors: Unlike Analog sensors, digital sensors produce discrete digital pulses like figure II.4 for a change in its environment. A push button switch is a very good example of a digital sensor. The output of this sensor can be either “ON” or “OFF”, i.e. it can be either 1 or 0. There are other digital sensors which output a series of digital pulses, or binary values. For example, a sensor can output a 10 bit binary value 0000000000 to 1111111111 (decimal equivalent of 0 to 1023) to signify a change. This means a sensor can produce one of 1024 values to suggest a change in its environment. It is important to realize this distinction between analog and digital outputs before selecting a sensor for your robot. Digital signals may seem easy to obtain and process, but involves a lot of calculations. The timer control in a microcontroller in itself is a nightmare. On the other hand, analog signals can be directly fed into a microcontroller, converted into a digital value using its built-in ADC and the information can be used as required [30].

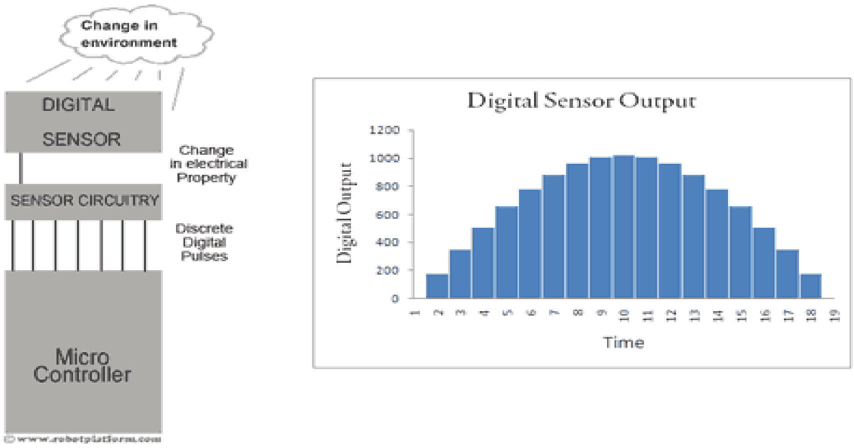


Figure II.7 .Digital Sensor [30]

II.9. Type Of Robot Sensor

There are different sensors to choose from and we will identify the characteristics of few sensors, and also understand why and where they are used.

II.9.1. Motion Detection Sensor: In figure II.5 is infrared motion sensor the most popular type of motion sensor. Infrared radiation basically lies in the range of the electromagnetic spectrum at a wavelength that is much longer than visible light. It cannot be seen, but it can be detected. It is a property that an object generates heat and infrared radiation at the same time and those objects, including the human body, animals and birds, whose radiation is strongest at a wavelength of 9.4 µm [10].

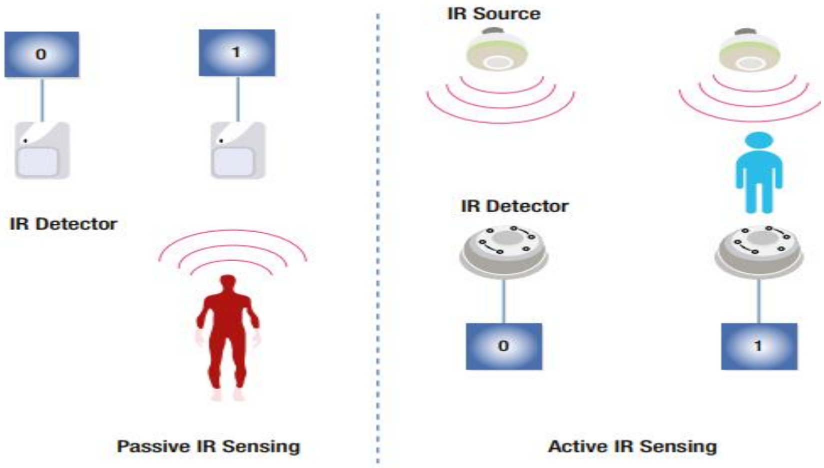


Figure II.8.Motion Detection Sensor [31]

II.9.2. Inertial Measurement Unit (IMU): An inertial measurement unit (IMU) is a device that uses gyroscopes and accelerometers to estimate the relative position, velocity, and acceleration of a moving vehicle. An IMU is also known as an Inertial Navigation System (INS). IMU has three orthogonal accelerometers and three orthogonal gyroscopes. The gyroscope data is integrated to estimate the vehicle orientation while the three accelerometers are used to estimate the instantaneous acceleration of the vehicle. The acceleration is then transformed to the local navigation frame by means of the current estimate of the vehicle orientation relative to gravity [31].

II.9.2.1. Accelerometers: There are five modes of motion sensing: acceleration, vibration (periodic acceleration), shock (instantaneous acceleration), tilt (static acceleration), and rotation. All of these, except rotation, can be measured using accelerometers. Accelerometers are typically either capacitive or piezo-resistive. Capacitive accelerometers are composed of fixed plates attached to a substrate and moveable plates attached to the frame. Displacement of the frame, due to acceleration, changes the differential capacitance, which is measured by the on-board circuitry. Capacitive accelerometers offer high sensitivities and are utilized for low-amplitude, low-frequency devices [31].

II.9.2.2. Gyroscopes: Gyroscopes in figure II.6 measure the angular rate of rotation of one or more axes. Gyroscopes can measure intricate motions accurately in free space. They have no rotating parts that require bearings, and therefore lend themselves to miniaturization and batch fabrication using semiconductor manufacturing processes [31].

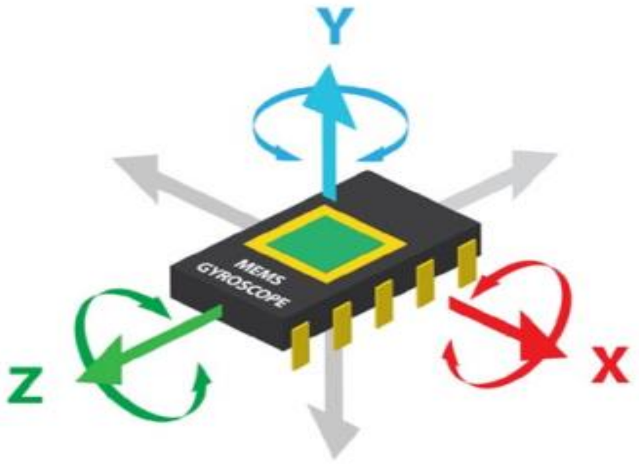


Figure II.9.3D Angular Rotation Measurements [31]

II.9.3. Ground Beacons: Approach to solving the localization problem in mobile robotics is to use active or passive beacons. Using the interaction of on-board sensors and the environmental beacons, the robot can identify its position precisely. Although the general intuition is identical to that of early human navigation beacons, such as stars, mountains, and lighthouses, modern technology has enabled sensors to localize an outdoor robot with accuracies of better than 5 cm within areas that are kilometers in size [5].

II.9.3.1. GPS (Global Positioning System): The most commonly used positioning sensor is a GPS in figure II.7. Satellites orbiting our earth transmit signals and a receiver on a robot acquires these signals and processes it. The processed information can be used to determine the approximate position and velocity of a robot. These GPS systems are extremely helpful for outdoor robots, but fail indoors. They are also bit expensive at the moment and if their prices fall, very soon you would see most robots with a GPS module attached [32] .

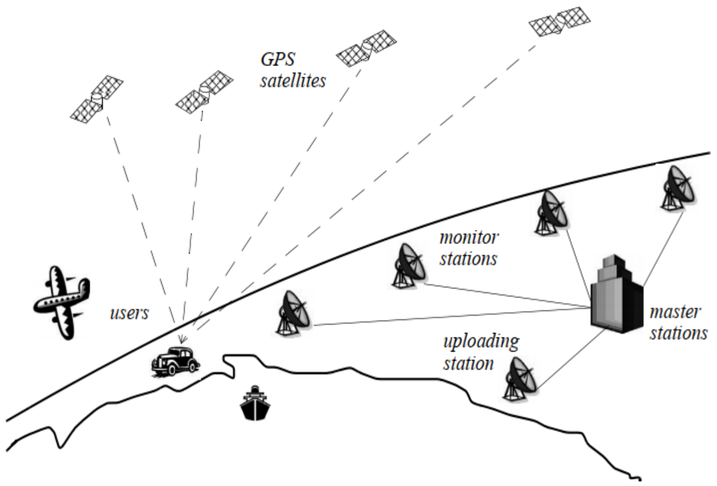


Figure II.10.Calculation Of Position And Heading Based On GPS [5]

II.9.3.2. Digital Magnetic Compass: Similar to a handheld magnetic compass, Digital Magnetic compass provides directional measurements using the earth's magnetic field which guides your robot in the right direction to reach its destination. These sensors are cheap compared to GPS modules, but a compass works best along with a GPS module if you require both positional feedback and navigation. Philips KMZ51 is sensitive enough to detect earth's magnetic field [32].

II.9.3.3. Localization: Localization refers to the task of automatically determining the location of a robot in complex environment. Localization is based on external elements called landmarks which can be either artificially placed landmarks, or natural landmark. In the first approach, artificial landmarks or beacons are placed around the robot, and a robot's sensor captures these signals to determine its exact location. Natural landmarks can be doors, windows, walls, etc. which are sensed by a robots sensor / vision system (Camera). Localization can be achieved using beacons which generate Wi-Fi, Bluetooth, Ultrasound, Infrared, Radio transmissions, Visible Light, or any similar signal [32].

II.9.4. Active Ranging: Active ranging sensors continue to be the most popular sensors in mobile robotics. Many ranging sensors have a low price point, and, most importantly, all ranging sensors provide easily interpreted outputs: direct measurements of distance from the robot to objects in its vicinity. For obstacle detection and avoidance, most mobile robots rely heavily on active ranging sensors. But the local free space information provided by ranging sensors can also be accumulated into representations beyond the robot's current local reference frame. Thus active ranging sensors are also commonly found as part of the localization and environmental modeling processes of mobile robots [5].

II.9.4.1. Time-Of-Flight Active Ranging: Time-of-flight ranging makes use of the propagation speed of sound or an electromagnetic wave. In general, the travel distance of a sound of electromagnetic wave is given by:

$d=c * t$ where :d= distance traveled (usually round-trip); c= speed of wave propagation;

t= time of flight.

It is important to point out that the propagation speed of sound is approximately 0.3 m/ms whereas the speed of electromagnetic signals is 0.3 m/ns, which is 1 million times faster. The time of flight for a typical distance, say 3 m, is 10 ms for an ultrasonic system but only 10 ns for a laser rangefinder. It is thus evident that measuring the time of flight with electromagnetic signals is more technologically challenging. This explains why laser range sensors have only recently become affordable and robust for use on mobile robots. The quality of time-of-flight range sensors depends mainly on uncertainties in determining the exact time of arrival of the reflected signal; inaccuracies in the time-of-flight measurement (particularly with laser range sensors); the dispersal cone of the transmitted beam (mainly with ultrasonic range sensors); interaction with the target (e.g., surface absorption, specular reflections); variation of propagation speed; the speed of the mobile robot and target (in the case of a dynamic target); As discussed below, each type of time-of-flight sensor is sensitive to a particular subset of the above list of factors [5].

II.9.4.1.1. The Ultrasonic Sensor (Time-Of-Flight, Sound) : Ultrasonic sensors in figure II.8 provide an excellent way to measure the distance of the robot from obstacles (providing a measurement of between 2 cm and 20 cm). The ultrasonic module functions by sending a short burst of ultrasonic pulses and then measures the time it takes for the receiver to detect the echo. The module then produces a pulse on the echo output that is equal to the time measured. This time is equal to the distance travelled divided by the speed of sound (340.29 m/sec or 34,029 cm/s), which is the distance from the sensor to the object and back again.[33]

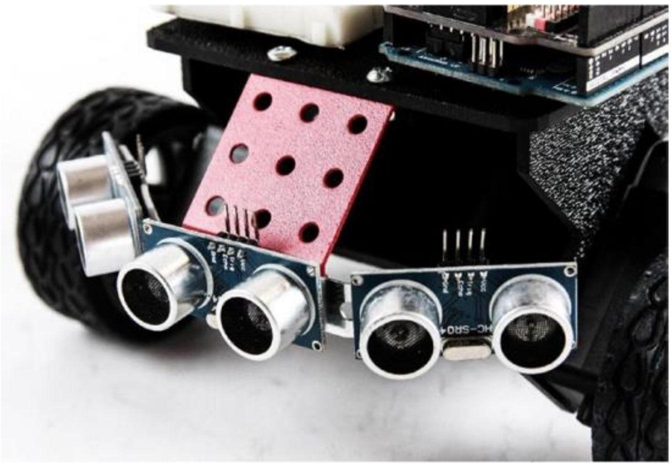


Figure II.11. Ultrasonic sensors mounted [34]

II.9.4.1.2. Laser Rangefinder (Time-Of-Flight, Electromagnetic): The laser rangefinder is a time-off light sensor in the following figure that achieves significant improvements over the ultrasonic range sensor owing to the use of laser light instead of sound. This type of sensor consists of a transmitter which illuminates a target with a collimated beam (e.g., laser), and a receiver capable of detecting the component of light which is essentially coaxial with the transmitted beam. One way to measure the time of flight for the light beam is to use a pulsed laser and then measure the elapsed time directly, just as in the ultrasonic solution described earlier. Electronics capable of resolving picoseconds are required in such devices and they are therefore very expensive. A second method is to measure the beat frequency between a frequency modulated continuous wave and its received reflection. Another, even easier method is to measure the phase shift of the reflected light [5].

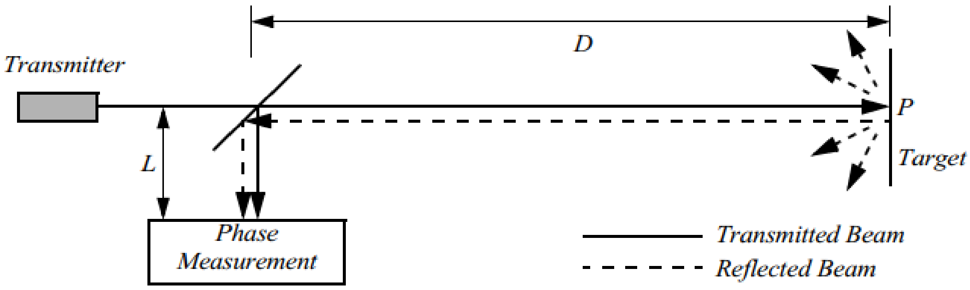


Figure II.12.Schematic Of Laser Range Finding By Phase-Shift Measurement [5]

II.9.4.2. Lidar: Lidar in the following figure is one of the most popular sensors available for robot navigation. The most popular lidar systems are planar, using a rotating mirror to scan the beam from a solid-state laser over an arc of typically 180 degrees or more. Lidar systems with nodding mirrors have been developed that can image three-dimensional space, but their cost, refresh rate, and reliability have not yet reached the point of making them popular for most robot designs. The biggest advantage of lidar systems is that they can detect most passive objects over a wide sweep angle to 10 meters or more and retro reflective targets to more than 100 meters. The effective range is even better to more highly reflective objects. The refresh (sweep) rate is usually between 1 and 100 Hz and there is a trade-off between angular resolution and speed. Highest resolutions are usually obtained at sweep rates of 3 to 10 Hz. This relatively slow sweep rate places some limitations on this sensor for high-speed vehicles [35].

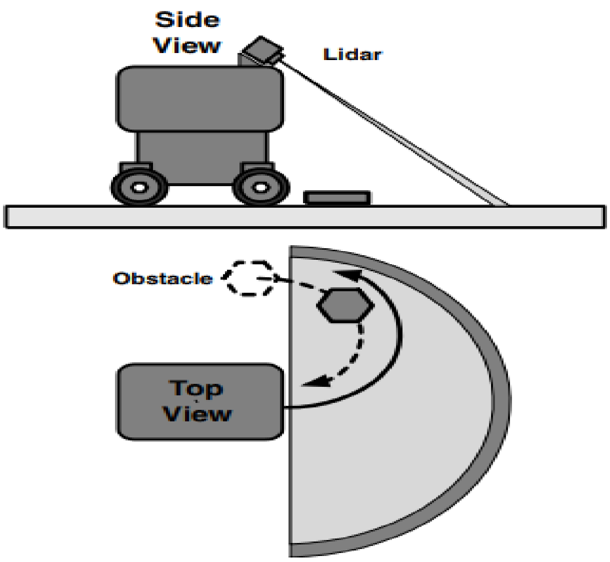


Figure II.13.Downward-Looking Lidar For Collision Avoidance [35]

II.9.5. Light sensor : A Light sensor in the following figure is used to detect light and create a voltage difference. The two main light sensors generally used in robots are Photo-resistor and Photo-voltaic cells. Other kinds of light sensors like Photo-tubes, Photo-transistors, CCD's etc. are rarely used [32].

Figure II.14.Light Sensor [32]

II.9.6. Sound sensor: This sensor in the following figure (generally a microphone) detects sound and returns a voltage proportional to the sound level. A simple robot can be designed to navigate based on the sound it receives. Imagine a robot which turns right for one clap and turns left for two claps. Complex robots can use the same microphone for speech and voice recognition [32].

Figure II.15.Sound Sensor [32]

II.9.7. Temperature Sensor: Semiconductor temperature sensors in following figure are based on the change of voltage across a p-n junction, which exhibits strong thermal dependence. The simplest form of temperature sensor is a silicon diode where the forward bias across the diode has a temperature coefficient of approximately 2.0–2.3mV/°C Few generally used temperature sensor IC's are LM34, LM35, TMP35, TMP36, and TMP37 [31][32] .

Figure II.16.Temperature Sensor [32]

II.9.8. Contact Sensor: Contact sensors like Figure II.14 are those which require physical contact against other objects to trigger. A push button switch, limit switch or tactile bumper switch are all examples of contact sensors. These sensors are mostly used for obstacle avoidance robots. When these switches hit an obstacle, it triggers the robot to do a task [32].

Figure II.17.Contact Sensor [32]

II.9.9. Other Sensor Of Robots: There are hundreds of sensors made today to sense virtually anything you can think of, and it is almost impossible to list all available sensors. Apart from those mentioned above, there are many other sensors used for specific applications. For example: Humidity Sensors measures Humidity; Gas sensors are designed to detect particular gases (helpful for robots which detects gas leaks); Potentiometers are so versatile that they can be used in numerous different applications; Magnetic Field Sensors detect the strength of magnetic field around it pressure sensor measures pressure…etc [32].

II. 10. Robot And Motors:

From conventional robots to space robots, all kinds of robotic systems use several kinds of motors. Most robotic systems use hybrid motor units.

II. 10.1. DC Motor: A dc motor like Figure II.15 converts dc electrical energy into rotational mechanical energy. A major part of the torque generated in the rotor (armature) of the motor is available to drive an external load. The dc motor is probably the earliest form of electric motor. Because of features such as high torque, speed controllability over a wide range**.**

- Torque is approximately proportional to current.
- Speed is approximately proportional to voltage.
- Control circuitry employs electrical switches to deliver power to the motor.
- A controller can govern the motor’s operation using PWM (pulse width modulation) signals[36][37].

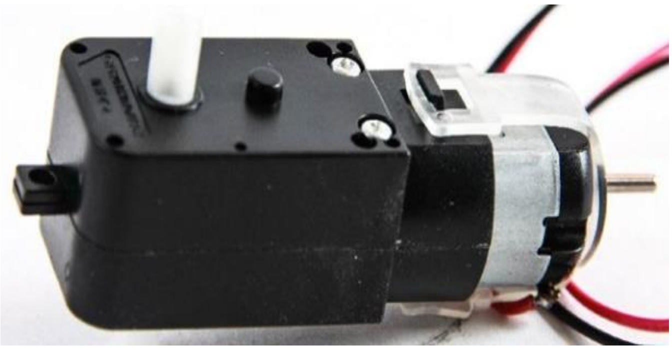


Figure II.18.Dc Motor [28]

II. 10.2. Servomotor : Servomotors in the following figure are motors with motion feedback control, which are able to follow a specified motion trajectory. In a dc servomotor system, both angular position and speed might be measured (using shaft encoders, tachometers, resolvers, rotary-variable differential transformers (RVDTs), potentiometers, etc… and compared with the desired position and speed. The error signal = ([desired response] - [actual response]) is conditioned and compensated using analog circuitry or is processed by a digital hardware processor or microcontroller [36] .

Figure II.19 .Servomotor [34]

II. 10.3. Stepper Motor: For precise positioning and speed control without the use of feedback sensors, the stepper motor like Figure II.17 is the obvious choice. Each time a pulse of electricity is sent to the stepper motor, the shaft of the motor moves a number of degrees that we want, which is the basic principle of the stepper motor. Since the shaft of the motor only moves the number of degrees that it desired when each pulse is delivered to the motor that can be controlled or to control the positioning and speed by using a controller. The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided like poles of a magnet repel and unlike poles attract [10].

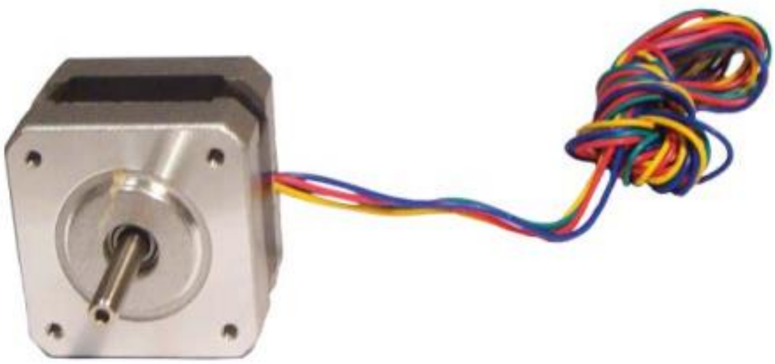


Figure II.20.Stepper Motor [37]

II. 11. Motor Drivers

Most micro controllers, microprocessors, and electronics can only handle a small amount of current. If one pulls too much current, it starts to burn out. Because motors usually easily exceed this maximum current, you generally don't want to connect a motor of any significant size directly to your processor. So, we will use a device called a motor driver or a motor controller. A motor controller is designed for this specific purpose. It uses the low power signal from your micro controller to control a much larger current and/or voltage [34].

II. 11.1. The L298 Driver Module: This dual bidirectional motor driver in the following figure is based on the very popular L298 Dual H-Bridge Motor Driver IC. This module will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a micro-controller requiring just a couple of control lines per motor [38].

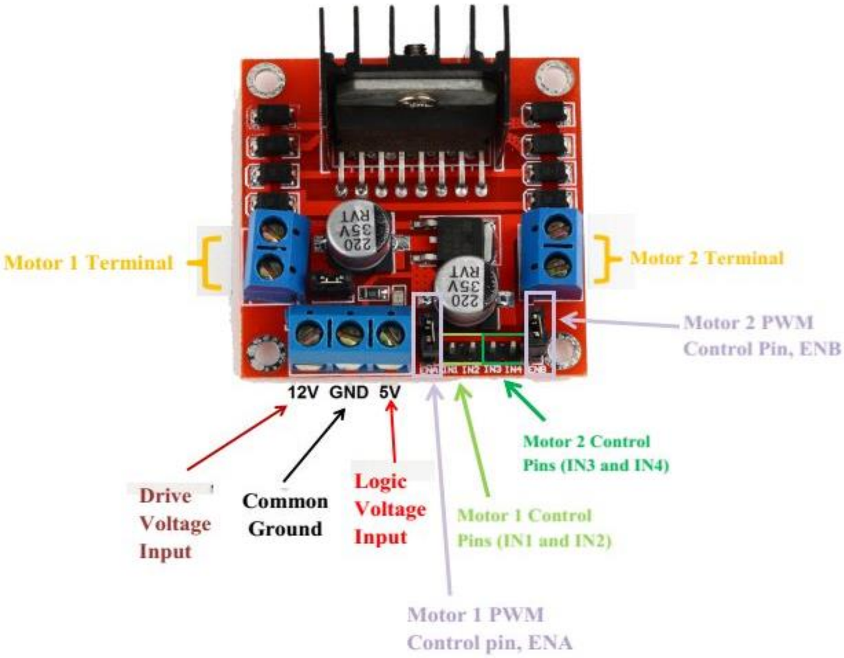


Figure II. 21. L298 Motor Driver Module [38]

Pin	Description	Function
5V	Logic Voltage Input	Power supply to the module, connect to 5V
GND	Common Ground	Connect to ground.
VCC	Drive Voltage Input	Power Supply to the motor, 4.8V to 16V.
IN1, IN2, IN3, IN4	Motor Control Pin	Control motors rotate direction, either clockwise or anticlockwise.
OUT1, OUT2, OUT3, OUT4	Motor Terminal	Connect to the motors.
L1, L2, L3, L4	Motor Status LED	Indicate the status of the motors.
SW1	Share Power Source Selected Switch	ON: Share the drive power to logic power. OFF: Logic power isolated from drive power.
ENA, ENB	Motor PWM Control pin	Control PWM (speed) of the motors.

Table II.1. Pin Definition of L298 Motor Driver Module [38]

Motor 1		
IN1	IN2	Motor Rotation
0	0	Stop rotate
0	1	Clockwise
1	0	Anticlockwise
1	1	Stop rotate

Motor 2		
IN3	IN4	Motor Rotation
0	0	Stop rotate
0	1	Clockwise
1	0	Anticlockwise
1	1	Stop rotate

Table II.2. Theory Of Operation Of L298 Motor Driver Module[38]

II. 11. 1.1. Pulse-Width Modulation: PWM in the figure II 19 is a technique that works on the principle of averaging the signal. If the signal is on for 10% of the total cycle, then when the signal is averaged out, the result is an analog 10% of the two digital extremes. If the highest voltage produced by the GPIO output is 5 V, a repeating digital signal that is only on for 75% of that cycle produces an average voltage that's determined as follows: Vavg= 5 x 0.75 =3.75 If the GPIO output signal was high only 10% of the time, the averaged result is Vavg = 5 V. The on time as a percentage of the total cycle time is known as the duty cycle. Obviously, there is an averaging aspect to all of this. If you applied the 10% signal to the probes of an oscilloscope, you'd see a choppy digital-looking signal. The duty cycle may be there, but the averaging effect is not. The averaging effect is accomplished in several ways. In a light bulb, the element is heated up by the on pulses but does not cool immediately, so its brightness reflects the averaged current flow. A DC motor does not immediately stop when the current is withdrawn, because the rotational inertia keeps the rotor spinning. A meter's pointer does not immediately move back to zero when the current is removed. All of these physical effects have an averaging effect that can be exploited [39].

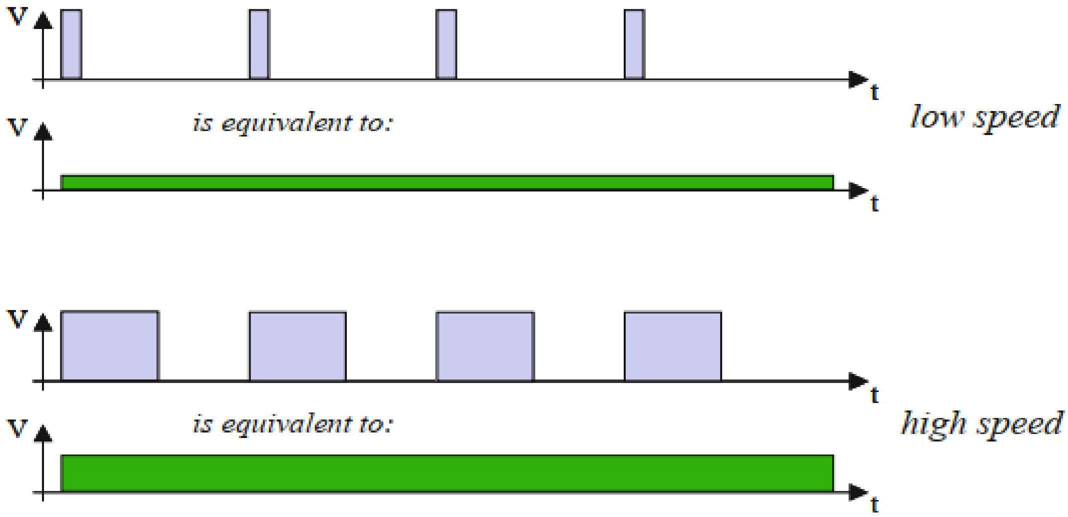


Figure II .22. Two different PWM voltage traces [2]

II. 12 Conclusion

This chapter presents the basic component of robots sensors that make robot discover the world around it ,and actuator that provide the movement to robot , all those component can't act individual they must have Brain another word controller that programmed it to do something useful . We saw the robotic is the best example of embedded system.

Chapter III

Generalities Artificial Intelligence

III.1. Introduction

You can hardly avoid encountering mentions of AI today. You see AI in the movies, in books, in the news, and online. AI is part of robots, self-driving cars, drones, medical systems, online shopping sites, and all sorts of other technologies that affect your daily life in so many ways. Many pundits are burying you in information (and disinformation) about AI, too. Some see AI as cute and fuzzy; others see it as a potential mass murderer of the human race. The problem with being so loaded down with information in so many ways is that you struggle to separate what's real from what is simply the product of an overactive imagination. Much of the hype about AI originates from the excessive and unrealistic expectations of scientists, entrepreneurs, and business persons [40] .

III.2. Artificial Intelligence Definition

The term artificial intelligence stirs emotions. For one thing there is our fascination with intelligence, which seemingly imparts to us humans a special place among life forms. Questions arise such as "What is intelligence?", "How can one measure intelligence?" or "How does the brain work?". Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision [41][42] .

III.3. Types of artificial intelligence

In Figure III.1 the types of artificial intelligence

III.3.1. Type 1:

- **Weak AI**: Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI [42] .
- **Strong AI**: Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities. When presented with an unfamiliar task, a strong AI system is able to find a solution without human intervention [42] .

III.3.2. Type 2:

- **Reactive machines :** Reactive machines are the most basic type of AI system. This means that they cannot form memories or use past experiences to influence present-made decisions; they can only react to currently existing situations – hence "reactive." An existing form of a reactive machine is Deep Blue, a chess-playing supercomputer created by IBM in the mid-1980s. Reactive machines have no concept of the world and therefore cannot function beyond the simple tasks for which they are programmed. A characteristic of reactive machines is that no matter the time or place, these machines will always behave the way they were programmed. There is no growth with reactive machines, only stagnation in recurring actions and behaviors [43].
- **Limited memory** : Limited memory is comprised of machine learning models that derive knowledge from previously-learned information, stored data, or events. Unlike reactive machines, limited memory learns from the past by observing actions or data fed to them in order to build experiential knowledge. Although limited memory builds on observational data in conjunction with pre-programmed data the machines already contain, these sample pieces of information are fleeting. An existing form of limited memory is autonomous vehicles. Autonomous vehicles, or self-driving cars, use the principle of limited memory in that they depend on a combination of observational and pre-programmed knowledge. To observe and understand how to properly drive and function among human-dependent vehicles, self-driving cars read their environment, detect patterns or changes in external factors, and adjust as necessary[43] .
- **Theory of mind :** What constitutes theory of mind is decision-making ability equal to the extent of a human mind, but by machines. While there are some machines that currently exhibit humanlike capabilities, none are fully capable of holding conversations relative to human standards. One component of human conversation is having emotional capacity, or sounding and behaving like a person would in standard conventions of conversation.

  This future class of machine ability would include understanding that people have thoughts and emotions that affect behavioral output and thus influence a "theory of mind" machine's thought process. Social interaction is a key facet of human interaction, so to make theory of mind machines tangible, the AI systems that control the now-hypothetical machines would have to identify, understand, retain, and remember emotional output and behaviors while knowing how to respond to them[43] .

- **Self-awareness :** Self-aware AI involves machines that have human-level consciousness. This form of AI is not currently in existence, but would be considered the most advanced form of artificial intelligence known to man. Facets of self-aware AI include the ability to not only recognize and replicate humanlike actions, but also to think for itself, have desires, and understand its feelings. Self-aware AI, in essence, is an advancement and extension of theory of mind AI. Where theory of mind only focuses on the aspects of comprehension and replication of human practices, self-aware AI takes it a step further by implying that it can and will have self-guided thoughts and reactions[43].

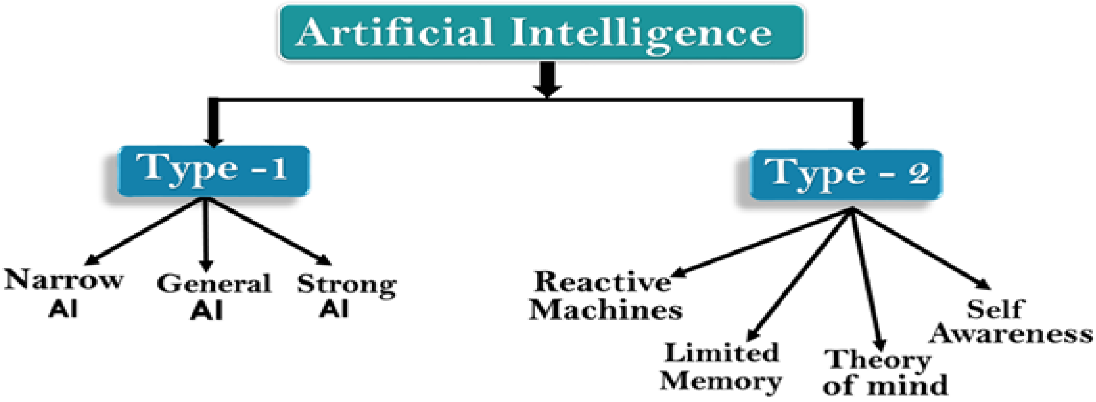


Figure III.1.Types of artificial intelligence [44]

III.4. Machine Learning Definition

Machine learning (ML) is a branch of artificial intelligence that systematically applies algorithms to synthesize the underlying relationships among data and information and writing programs that learn how to perform a task from experience, without being explicitly programmed to do so. For example, ML systems can be trained on automatic speech recognition systems to convert acoustic information in a sequence of speech data into semantic structure expressed in the form of a string of words. ML is already finding widespread uses in web search, ad placement, credit scoring, stock market prediction, gene sequence analysis, behavior analysis, smart coupons, drug development, weather forecasting, big data analytics, and many more applications. ML will play a decisive role in the development of a host of user-centric innovations. The computational characteristic of ML is to generalize the training experience (or examples) and output a hypothesis that estimates the target function. The generalization attribute of ML allows the system to perform well on unseen data instances by accurately predicting the future data [45][46] .

III.5. Machine Learning Categories

III.5.1.Supervised learning: The best analogy for supervised learning in figure III.2 is function approximation or curve fitting. In its simplest form, supervised learning attempts to extract a relation or function fx → y from a training set {x, y}. Supervised learning is far more accurate and reliable than any other learning strategy. However, a domain expert may be required to label (tag) data as a training set for certain types of problems [47] .

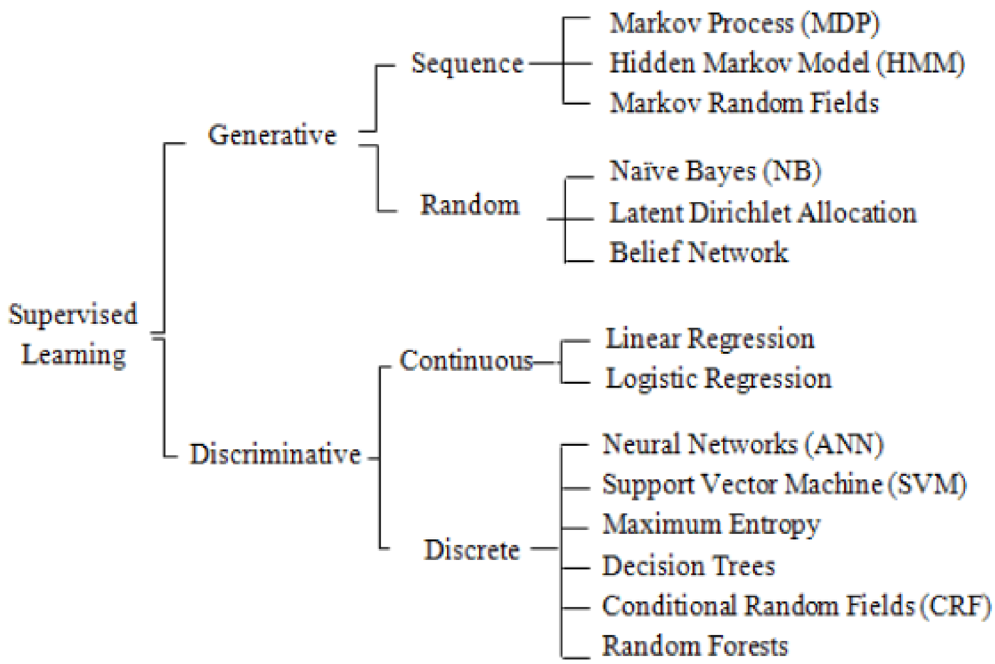


Figure III.2.classification of supervised learning [47]

- **Decision Trees:** The aim with any decision tree in the following figure is to create a workable model that will predict the value of a target variable based on the set of input variables. Financial institutions use decision trees. One of the fundamental use cases is in option pricing, where a binary-like decision tree is used to predict the price of an option in either a bull or bear market . Every tree is comprised of nodes. Each node is associated with one of the input variables. The edges coming from that node are the total possible values of that node. A leaf represents the value based on the values given from the input variable in the path running from the root node to the leaf. Because a picture paints a thousand words, see Figure III.3 for an example. Decision trees always start with a root node and end on a leaf. Notice that the trees don't converge at any point; they split their way out as the nodes are processed [48] .

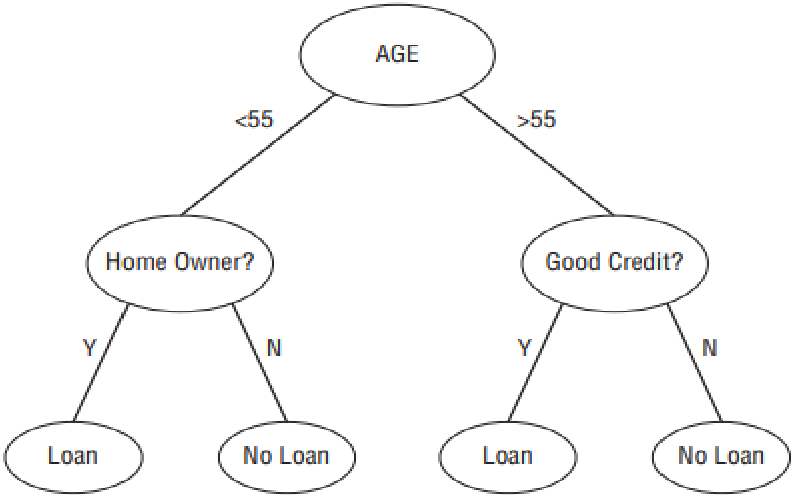


Figure III.3. Decision Trees [48]

- **Bayesian Networks :** The Bayesian Network in the following figure referred to by a few different names: probabilistic directed acyclic graphical model, Bayes Network, Belief Network, or Bayesian Model. Based on a set of variables or parameters, it's possible to predict outcomes based on probabilities. These variables are connected in such a way that the resulting value of one variable will influence the output probability of another, hence the use of networked nodes. A Bayesian Network manages to combine probability theory with graph theory and provides a very handy method for dealing with complexity and uncertainty. Bayesian Networks are found all over the place where uncertainty is in play, which turns out to be a lot of places. Where there is uncertainty, there is probability. Weather forecasting and stock option predictions are examples. The financial industry uses Bayesian Networks a lot to make reasonable predictions even when the data is not complete. Bayesian Networks are the perfect tool for the likes of the insurance, banking, and investment industries. In Figure III.4 a basic grasp of graphs, probability, and bayes Theorem, you can see there are three nodes: Yard Wet, Rain, and Hose. There are two events that would cause the yard to be wet; either the owner had hosed it or it has been raining. In any normal circumstance, you wouldn't hose the yard while it was raining. For each node you can assign true/false values:

Y = Yard wet (True or False) , R = Raining (True or False)

H = Someone using the hose (True or False)

The joint probability would be written as: $p(Y,R,H) = p(Y|R,H)\ p(R|H)\ p(R)$ [48] .

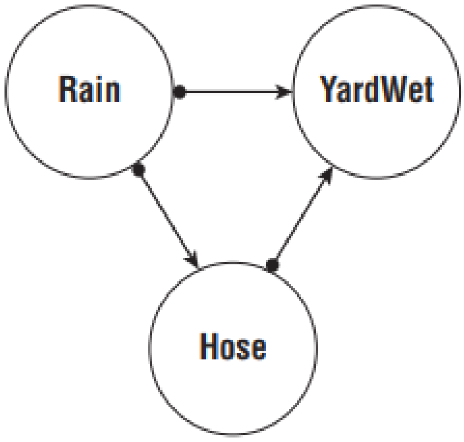


Figure III.4. A basic Bayesian Network [48]

- **Artificial Neural Networks:** Artificial Neural Networks in the following figure are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts. These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are currently impossible for computers. Computers do rote things well, like keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future. Now, advances in biological research promise an initial understanding of the natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then solving problems encompasses a new field in computing. This field, as mentioned before, does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. This field also utilizes words very different from traditional computing, words like behave, react, self-organize, learn, generalize [49] .

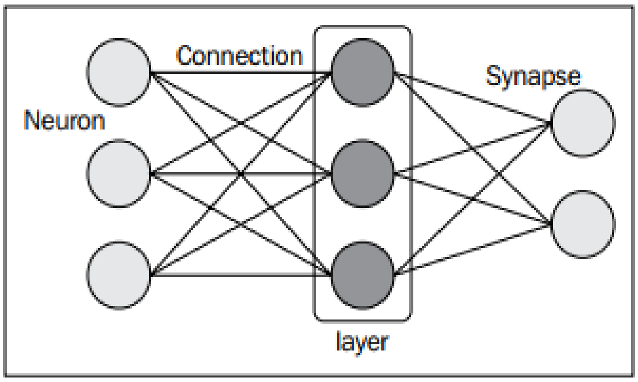


Figure III.5.The neuron structure [47]

III.5.2.Unsupervised Learning: Labeling a set of observations for classification or regression can be a daunting task, especially in the case of a large feature set. In some cases, labeled observations are either not available or not possible to create. In an attempt to extract some hidden association or structures from observations, the data scientist relies on unsupervised learning techniques to detect patterns or similarity in data. The goal of unsupervised learning is to discover patterns of regularities and irregularities in a set of observations. These techniques are also applied in reducing the solution space or feature set similarly to the divide-and-conquer approach commonly used in Computer Science [47] .

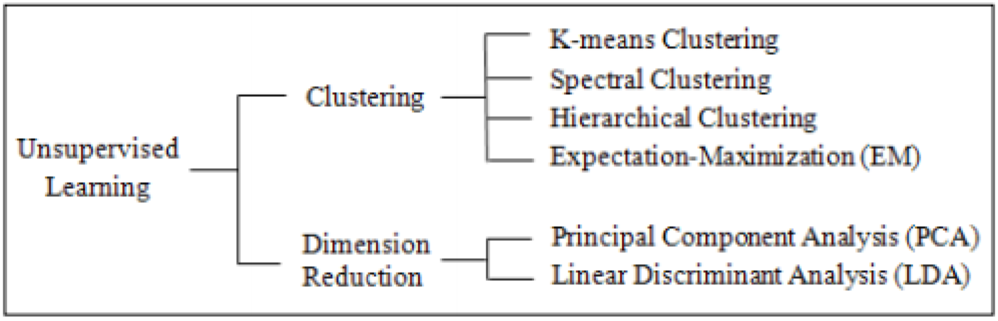


Figure III.6. Classification of supervised learning [47]

- **Clustering :** Problems involving a large number of features for large datasets become quickly intractable, and it is quite difficult to evaluate the independence between features. Any computation that requires some level of optimization and, at a minimum, computation of first order derivatives requires a significant amount of computing power to manipulate high-dimension matrices. As with many engineering fields, a divide-and-conquer approach to classifying very large datasets is quite effective. The objective is to reduce continuous, infinite, or very large datasets into a small group of observations that share some common attributes. This approach is known as vector quantization. Vector quantization is a method that divides a set of observations into groups of similar size. The main benefit of vector quantization is that the analysis using a representative of each group is far simpler than an analysis of the entire dataset the following figure III.7 Visualization of data clustering [47] .

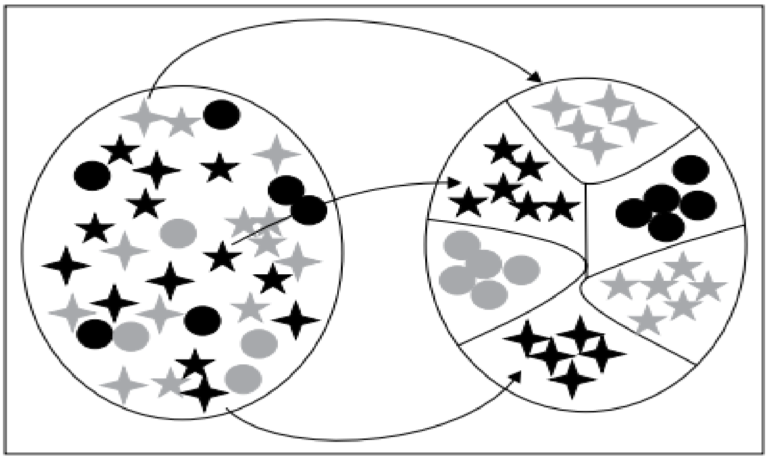


Figure III.7.Visualization of data clustering [47]

- **Dimension Reduction:** Without prior knowledge of the data domain, data scientists include all possible features in their first attempt to create a classification, prediction, or regression model. After all, making assumptions is a poor and dangerous approach to reduce the search space. It is not uncommon for a model to use hundreds of features, adding complexity and significant computation costs to build and validate the model. Noise-filtering techniques reduce the sensitivity of the model to features that are associated with sporadic behavior. However, these noise-related features are not known prior to the training phase, and therefore, cannot be discarded. As a consequence, training of the model becomes a very cumbersome and time-consuming task. Over-fitting is another hurdle that can arise from a large feature set. A training set of limited size does not allow you to create a model with a large number of features. Dimension reduction techniques alleviate these problems by detecting features that have little influence on the overall model behavior the following figure III.8 Visualization of principal components for a 2-dimension mode [47] .

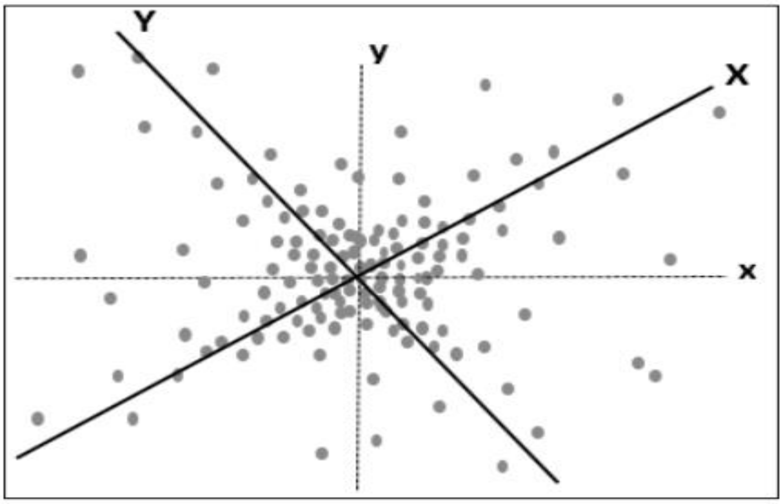


Figure III.8.Visualization of principal components for a 2-dimension model[47]

III.5.3. Reinforcement Learning: All of the learning algorithms described so far—except the clustering algorithms— belong to the class of supervised learning. In supervised learning, the agent is supposed to learn a mapping from the input variables to the output variables. Here it is important that for each individual training example, all values for both input variables and output variables are provided. In other words, we need a teacher or a database in which the mapping to be learned is approximately defined for a sufficient number of input values. The sole task of the machine learning algorithm is to filter out the noise from the data and find a function which approximates the mapping well, even between the given data points. In reinforcement learning the situation is different and more difficult because no training data is available. We begin with a simple illustrative example from robotics, which then is used as an application for the various algorithms in the following figure architecture of the reinforcement learning [47] .

- **Terminologies:** Reinforcement learning introduces a new terminology as listed here, quite different from that of older machine learning techniques:
  - ✓ **Environment:** The environment is any system that has states and mechanisms to transition between states. For example, the environment for a robot is the landscape or facility it operates in.
  - ✓ **Agent:** The agent is an automated system that interacts with the environment.
  - ✓ **State:** The state of the environment or system is the set of variables or features that fully describe the environment.
  - ✓ **Goal or absorbing state or terminal state:** A goal state is the state that provides a higher discounted cumulative rewards than any other state. It is a constraint on the training process that prevents the best policy from being dependent on the initial state.

- ✓ **Action:** An action defines the transition between states. The agent is responsible for performing or at least recommending an action. Upon execution of the action, the agent collects a reward or punishment from the environment.
- ✓ **Policy:** The policy defines the action to be selected and executed for any state of the environment. Best policy. This is the policy generated through training. It defines the model in Q-learning and is constantly updated with any new episode.
- ✓ **Reward:** A reward quantifies the positive or negative interaction of the agent with the environment. Rewards are essentially the training set for the learning engine.
- ✓ **Episode:** This defines the number of steps necessary to reach the goal state from an initial state. Episodes are also known as trials.
- ✓ **Horizon:** The horizon is the number of future steps or actions used in the maximization of the reward. The horizon can be infinite, in which case the future rewards are discounted in order for the value of the policy to converge

- **Concept :** The key component in reinforcement learning is a decision-making agent that reacts to its environment by selecting and executing the best course of actions and being rewarded or penalized for it . You can visualize these agents as robots navigating through an unfamiliar terrain or a maze. Robots use reinforcement learning as part of their reasoning process after all. The following diagram Figure III.9 gives the overview architecture of the reinforcement learning agent:

The agent collects the state of the environment, selects, and then executes the most appropriate action. The environment responds to the action by changing its state and rewarding or punishing the agent for the action.

The four steps of an episode or learning cycle are as follows:

1. The learning agent either retrieves or is notified of a new state of the environment.

2. The agent evaluates and selects the action that may provide the highest reward.

3. The agent executes the action.

4. The agent collects the reward or penalty and applies it to calibrate the learning algorithm.

The action of the agent modifies the state of the system, which in turn notifies the agent of the new operational condition. Although not every action will trigger a change in the state of the environment, the agent collects the reward or penalty nevertheless. At its core, the agent has to design and execute a sequence of actions to reach its goal [47].

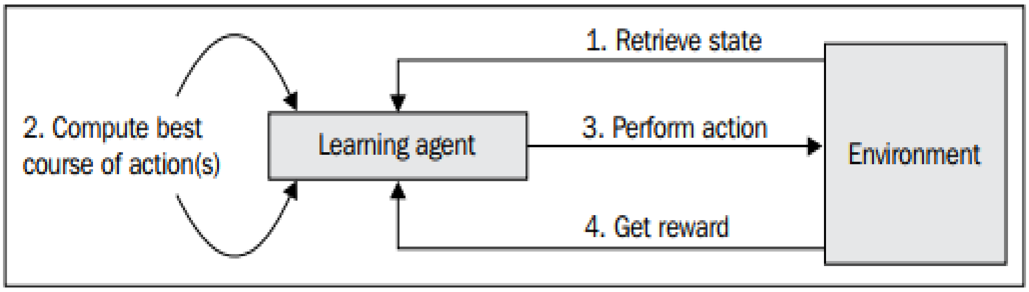


Figure III.9.Architecture of the reinforcement learning [47]

III.6. Maze Solving Algorithms

As the development of Robotics technology is expanding rapidly, the developments in the world of artificial intelligence is also growing very fast. The challenge is how to make a robot to become cleverer in deciding actions based on the circumstances that exist .our goal is to build robot "smart" to solve the maze and find the nearest road to re-discover the maze. It is applying artificial intelligence ,there are a lot of artificial intelligence algorithms but we choose to discuss 3 More known of them to be compared and we will chose one to program our robot with, and test it, and so it is hoped to produce optimal results.[50]

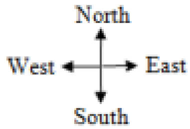
III.6.1. Fill Flood Algorithm: This is one of the most efficient maze solving algorithms. The fill flood algorithm in the following figure is derived from the “Bellman Ford Algorithm”. It paved new methods by which complex and difficult mazes can be solved without having any troubles. The algorithm works by assigning value for all cells in the maze, where these values indicate the steps from any cell to the destination cell . Implementing this algorithm requires to have two arrays. The first array is holding the walls map values, while the other one is storing the distance values [51] .

Array Positions

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

Flood Fill

4	3	2	3	4
3	2	1	2	3
2	1	**0**	1	2
3	2	1	2	3
4	3	2	3	4



Bit No.	7	6	5	4	3	2	1	0
Wall					W	S	E	N

Map Array

array	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
value	7	3	6	3	6	1	4	9	12	5	5	5	11	2	4	5	5	11	12	5	13	9	10	14	13

Distance Array

array	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
value	4	3	2	3	4	3	2	1	2	3	2	1	0	1	2	3	2	1	2	3	4	3	2	3	4

Figure III.10.Fill flood algorithm using one-dimensional arrays [51]

Every time robot arrives in a cell it will perform the following steps

1 - Update the wall map

2 - Flood the maze with new distance value (only if necessary)

3 - Decide which neighboring cell has the lowest distance value

4 - Move to neighboring cell with the lowest distance value

III.6.1.1.Modified Fill Flood Algorithm: The fill flood algorithm works by assigning value for all cells in the maze, where these values indicate the steps from any cell to the destination cell. The array was holding integers values, which will not provide the best look for the simulation results. Therefore, the multidimensional array filled with characters informs of aliphatic letters. In the program character value can be traded as a number using the characters table from Arduino website. For example “a” is equal 97, which should be assumed as zero. The flow chart of the algorithm can be seen in Figure III.12 . The program can be divided into three main parts [51] .

- **Part 1 Solve The Maze :** The robot will start with the first step and then it will enter “solving the maze” loop. This loop will be repeated as long as the robot not reaches the goal. In this loop, the robot will scan the walls of the test maze and decide which adjacent cell has the lowest value. Then, the robot will move to the cell and check if it necessarily to update the distance value,

  [current value = adjacent cell value +1] . After that, the cell value will store in integer “current value” and it will replaced in the maze with character “M” to show the robot movements. The FFA Maze will printed over and over with showing the robot steps until it reaches to the centre of the maze [51] .

- **Part 2 Find The Shorten Path :** In this process the robot position will be on the centre of the maze and it should return back to starting position. This process is almost the same as the process of solving the maze process. The only different is, if the adjacent cell value is Less than the current cell value, [adjacent cell value = current value +1]. The maze will printed until the robot reaches the starting position. After completing this process, the FFA Maze will have new values representing the right orders of steps required to reach the centre of the maze [51] .
- **Part 3: Mark The Shorten Path :** This process is going to mark the shorten path. Since the FFA Maze has the right steps values, marking the shorten path will be straight forward through repeating solve the maze process. Each step the robot takes will be marked with character "X" [51] .

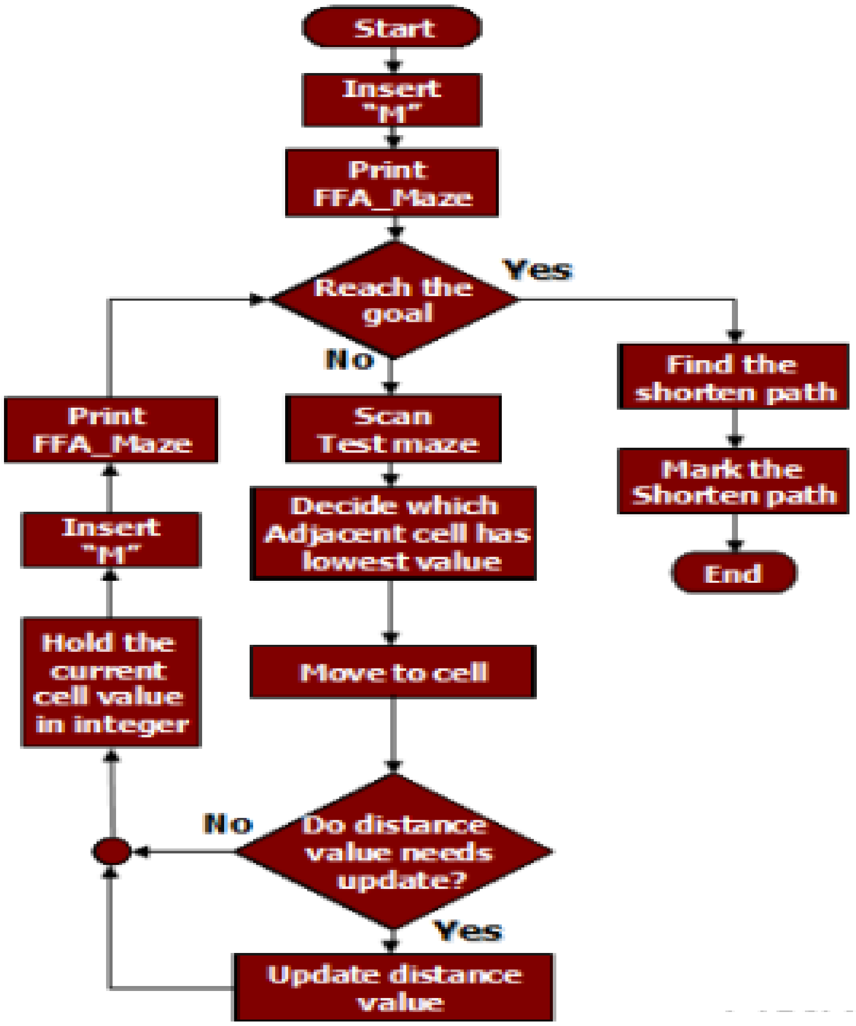


Figure III.11 Fill Flood Flow Chart [51]

III.6.2.A* Search Algorithm: A* is a generic search algorithm which can be used to find solutions for several problems, path finding basically to be one of them. A* algorithm is completed when that location is the goal. In any other case, it helps make note of all that location neighbors for additional exploration. A* could possibly be the most popular path finding algorithm in game AI. The time complexity of this algorithm is depending on heuristic used .

The maze environment will be handled as a graph. This involves breaking the map into different points or locations, which are usually called nodes. These nodes are used to record the progress of the search. In addition to holding the map location, each node has three other attributes that are fitness, goal, and heuristic commonly known as f, g, and h respectively . The purpose of g, h, and f is to quantify how promising a path is up to the present node. The attributes g, h, and f are described in Table III.1. Different values will be assigned to paths between the nodes. Typically, these values would represent the distances between the nodes. The cost between nodes doesn't have to be distance. The cost could be time, if you wanted to find the path that takes the shortest amount of time to traverse. A* using two lists (open list and closed list). The open list contains all the nodes in the map that have not been totally explored yet. The closed list contains of all the nodes that have been totally explored. In addition to the standard open/closed lists, marker arrays can be used to find out whether a state is in the open or closed list [52]. In figure III.13 A* algorithm flow chart.

#	Attribut	Description
1	g	Represent the cost of getting from the source node to the destination node (the summation of all values in the path between the source and destination
2	h	Represent the estimated cost from the source node to the destination node
3	f	Represent the summation of g and hand is the best estimate of the cost for the path going through the source node. F = g+ h

Table III.1 Attributes description [52]

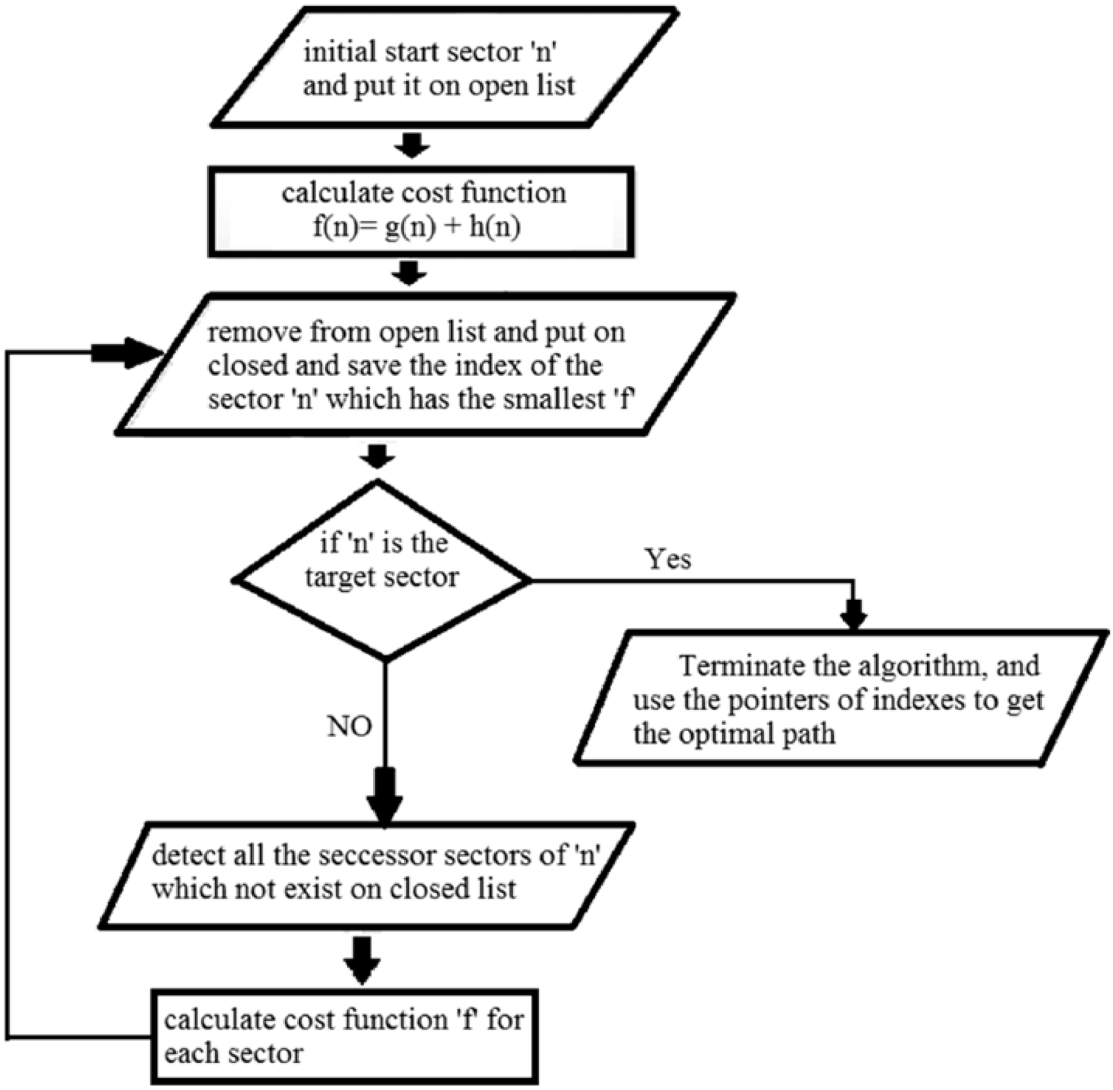


Figure III.12 A* algorithm flow chart [53]

III.6.3.Genetic Algorithms: Genetic algorithms draws inspiration from the work of Charles Darwin and is basically a model of an evolutionary system. The genetic algorithms randomly generate solutions and then continues building on the solutions that are most successful. This means that a GA can find solutions to problems even when the programmer is unaware of how to reach a go o d solution but has knowledge about the properties of such a solution. A genetic algorithm is used to solve a class of maze path finding problems. In particular, we find a complete set of paths directing an agent from any position in the maze towards a single goal [54].

- **How Genetic Algorithms Work:** The GA produces successive generations of individuals, computing their "fitness" at each step and selecting the best of them, when the termination condition arises. the following steps shows a Simple Genetic Algorithm approach [55] .
    1. Create initial random population.
    2. Calculate fitness of the individuals in the population.
    3. Repeat following steps until termination criteria is reached.
        a. Select best fit from current population and generate offspring.
        b. Evaluate fitness of each offspring.
        c. Replace weak individuals from current population with newly generated ones.

III.7. Conclusion

This chapter deals with artificial intelligence, in general, along with its classifications, and given examples. Our purpose is to implement the artificial intelligence inside an embedded system in order to obtain a more elaborated systems that make decisions autonomously. We chose the "Flood-Fill" algorithm because it is the most commonly used one in micro-mouse competitions due to its speed and assurance which run the best and provide the smallest number of moving cells .

In future, we hope to be able to deal with advanced algorithms of artificial intelligence like genetic algorithms and reinforcement learning algorithms .

Chapter IV

Application And Discussion

Chapter IV: Application And Discussion

IV .1.Introduction

A maze is a network of paths, typically from an entrance to exit. The concept of Maze approximately thousand years old .Which was invented in Egypt. From then, many mathematician made various algorithm to solve the maze.Now a days, maze solving problem is an important field of robotics. It is based on one of the most important areas of robot, which is “Decision Making Algorithm” .Google Maps finds the most efficient known route for outside locations. In combination with maps created through satellite imaging, Google uses autonomous vehicles equipped with cameras to detect and efficiently navigate surrounding unknown areas. This technology mapping of foreign regions and the subsequent algorithms that find efficient routes, is also useful to visualize and navigate particularly dangerous terrain. For instance, abandoned mine shafts are not safe for human exploration, but with autonomous mapping, they can be safely explored and updated for modern uses, such as storage and mineral extraction.The goal of this project is to map an unknown field and calculate the best path to travel from the entrance to the exit. A robot with five sensors (one in the front and two on both side and two other sensors that come in between) scans the various mazes built, using one trial for mapping and the second for traveling along the most efficient route. The testing of different algorithms for mapping the maze and navigating paths is backed by the desire to scale down real-world applications of affordable autonomous robots in order to gain a greater understanding of algorithm and robot development [56][57] .

The way that will be chosen to get students interested in this domain is by doing lab sessions; this part contains series of hands-on activities that introduce students to basic robotic concepts using our pedagogic mobile robot ,and we will see how and what it was designed for.

First Lab Session

IV.2.Component And Design Of Robot

The first step in any project is figuring out what pieces are required. A robot needs a few key things to be useful: a way to move, think, and interact with its surroundings. we need two wheels This means to steer we need two separate motors that can be operated independently. we also need a caster that the robot can lean on to glide along. This has the unfortunate downside that the robot really can't go on any surfaces other than smooth floors. we want the brains to be some sort of well-known microcontroller platform. The robot needs to have sensors that allow it to be aware of walls, and obstacles.

IV .2.1.objectives

- Student learn how to solder .
- Learn electronic basics .
- Build an experimental mobile robot.

IV .2.2.Choosing the components

IV .2.2.1.Mechanical: Motors, Gears, Wheels, Chassis :

- **Motors :** Straight BO Motor – 300 RPM VIGOR Original 300 RPM Straight Single Shaft Plastic Gear BO Motor for Arduino /Raspberry-Pi/Robotics. A DC Geared DC motor is a simple DC motor with gear box attached to the shaft of the motor which is mechanically commutated electric motor powered from direct current (DC) 3v normally, but could run higher at the expense of reduced operational life. as shown figure IV.1.

- **Wheels :** as shown figure IV.2
  1) With upgraded tire tread for greater friction
  2) New design wheel for better combination with the motor
  3) Diameter: 65 mm
  4) Width : 28 mm
  5) The material of Wheel: High-strength plastic
  6) Color: Yellow

figure IV.1 Straight BO Motor

figure IV.2 Wheels

- **Chassis :** as the follows figure

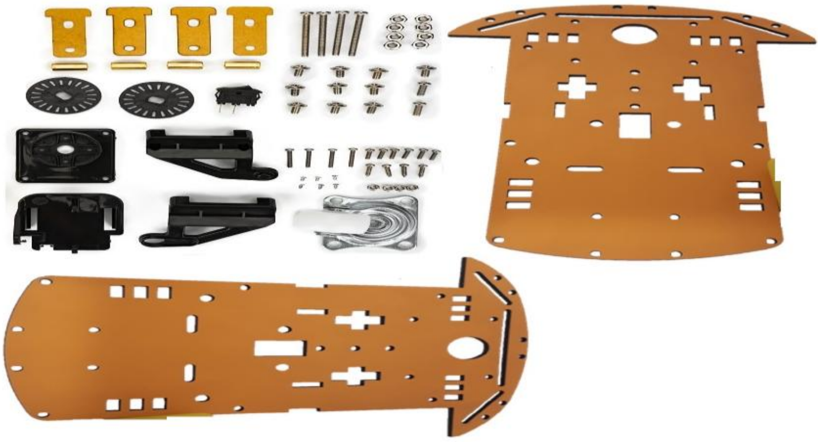


figure IV.3.Chassis of the robot

- **Tools-and-equipment :** Soldering iron, Welding Wire, paste-flux-soldering, wires ,Multimeter as the follows figure

figure IV.4.tools-and-equipment

- **Brains:** arduino uno (we have explained this in chapter 2) as shown figure IV.5
- **Drivers:** l298 motor driver and ultrasonic sensor hc-sr04 (we have explained this in chapter 2) as shown figure IV.6
- **Sensors :**Ultrasonic Sensors HC-SR04 (we have explained this in chapter 2) as shown figure IV.7
- **Communication :** Bluetooth hc-06 as shown figure IV.8
- **Power Supply:** power bank for mobile 10000mah 2 usb ports lcd display as shown figure IV.9

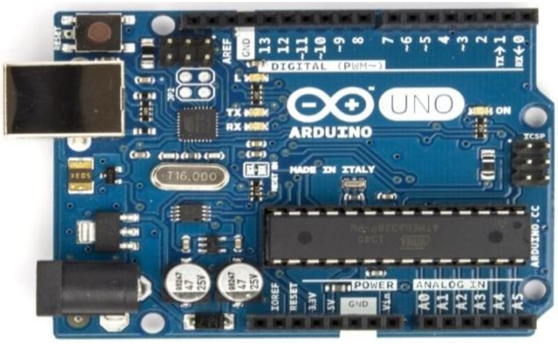


Figure IV.5.Arduino uno

figure IV.7.Ultrasonic Sensors HC-SR04

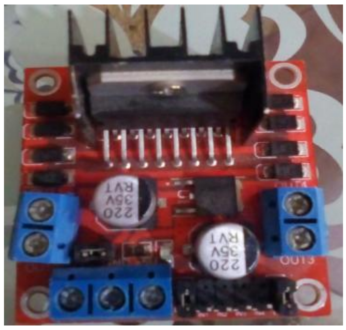


figure IV.6.L298 motor driver

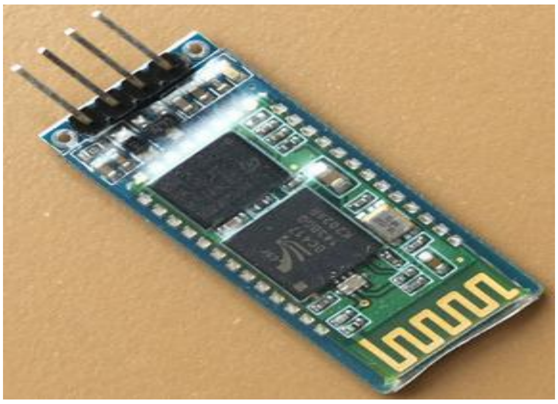


figure IV.8. Bluetooth hc-06

figure IV.9.Power Supply

Part Type	Part Number	Cost	Location
Microcontroller	Arduino uno "1"	2600.00 DA	https://www.dzduino.com/arduino-uno-rev3-dip-original?search=arduino%20%20uno&description=true
Robot Chassis	2WD Smart Robot Car "1"	2500.00 DA	https://www.dzduino.com/index.php?route=journal2/quickview&pid=1657
Motor Drivers	L298 Motor Driver "1"	900,00 DA	https://www.dzduino.com/l298-dual-h-bridge-motor-driver-2
Sensors	Ultrasonic HC-SR04 "5"	500,00 DA	https://www.dzduino.com/Ultrasonic-Sensor-HC-SR04?search=%20ultrasonic%20&description=true
Communication	Bluetooth hc-06	1200.00 DA	https://www.dzduino.com/hc-06-bluetooth-wireless-module?search=Bluetooth%20hc-06%20&description=true

Table IV.1 Places to buy component

IV.2.3.Conclusion

We have now identified the most important components of the robot's composition, Now we will see the complete robot in figure IV.10 and Top design in figure IV.11 with Bottom design in figure IV.12 .The student can program this robot in laboratory to move and interact with environment .

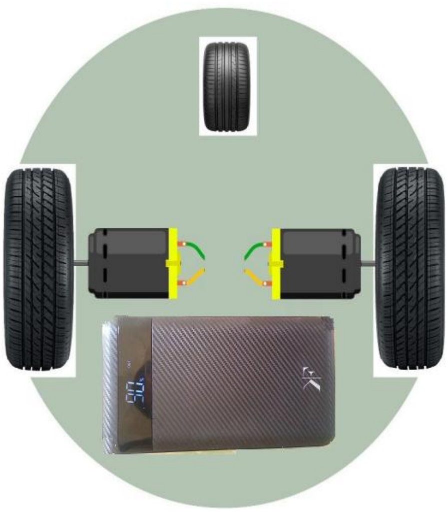


figure IV.12. Bottom design

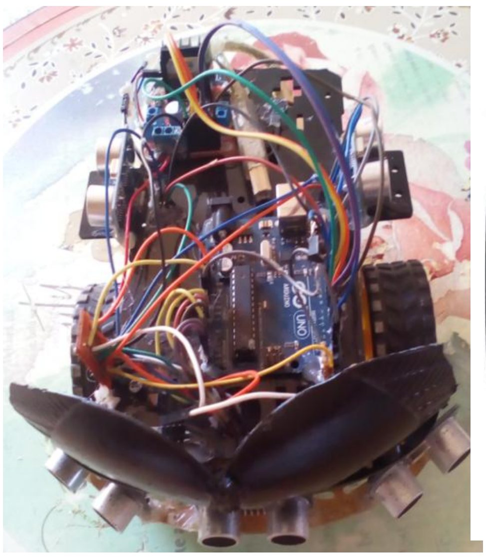


figure IV.10. Robot

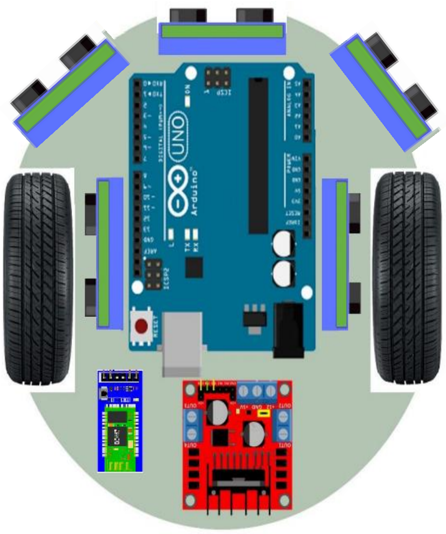


figure IV.11.Top design

Second Lab Session

IV.3.Control by Bluetooth

Is controlled by using Android mobile phone instead of any other method like buttons ,Here only needs to touch button in android phone to control the car in forward, backward, left and right directions. So here android phone is used as transmitting device and Bluetooth module placed in car is used as receiver. Android phone will transmit command using its in-built Bluetooth to car so that it can move in the required direction like moving forward, reverse, turning left, turning right and stop .

IV.3.1.Objective

- Wiring up the Bluetooth.
- Configuring and programming the Bluetooth on arduino uno
- Testing and discussing Bluetooth control.

IV.3.2.Hardware Description

- **Digital Pin Connection:**

Dc Motors	Digital Pin Connection
Motor A (Right)	2,4 (Turn Direction) – 3 (Speed)
Motor B (left)	5,7 (Turn Direction) – 6 (Speed)
Bluetooth	Tx(Bluetooth) → Rx(Arduino)

Table IV.2.Digital Pin Connection

- **circuit diagramme :** The following figure

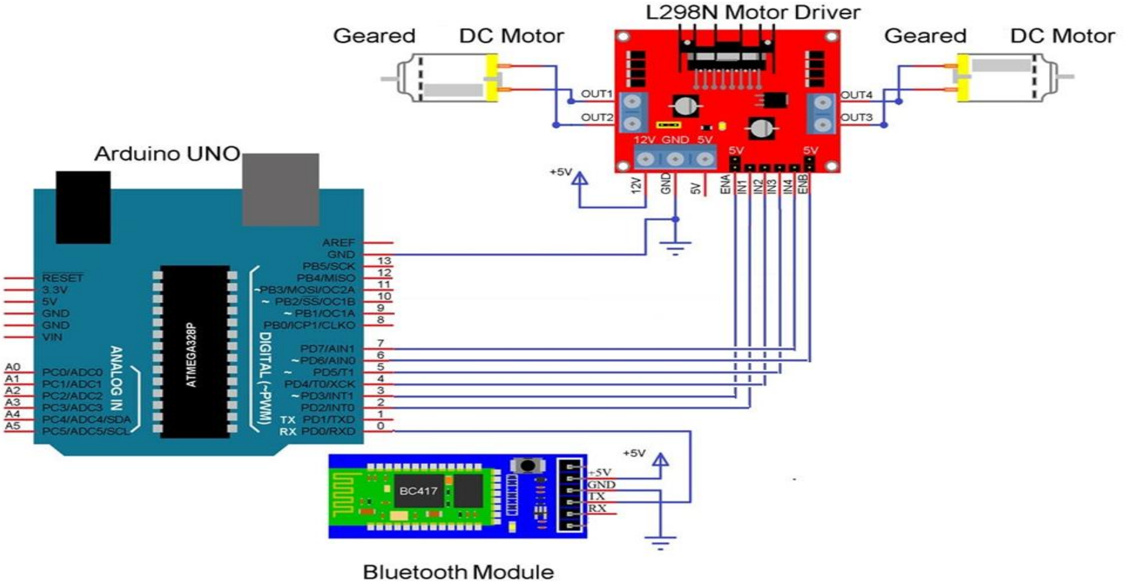


figure IV.13. Circuit Diagramme

- **Note** : The software that we used to draw the circuit diagram is fritzing.

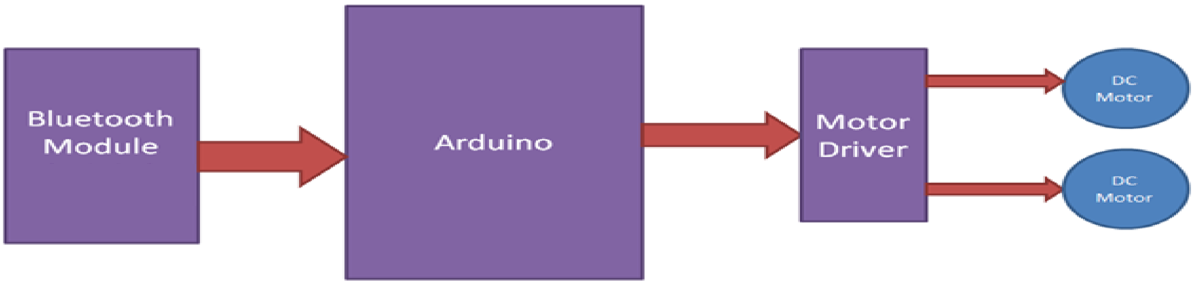


figure IV.14. Block diagram Bluetooth control

- **Flowchart For The Programme:**

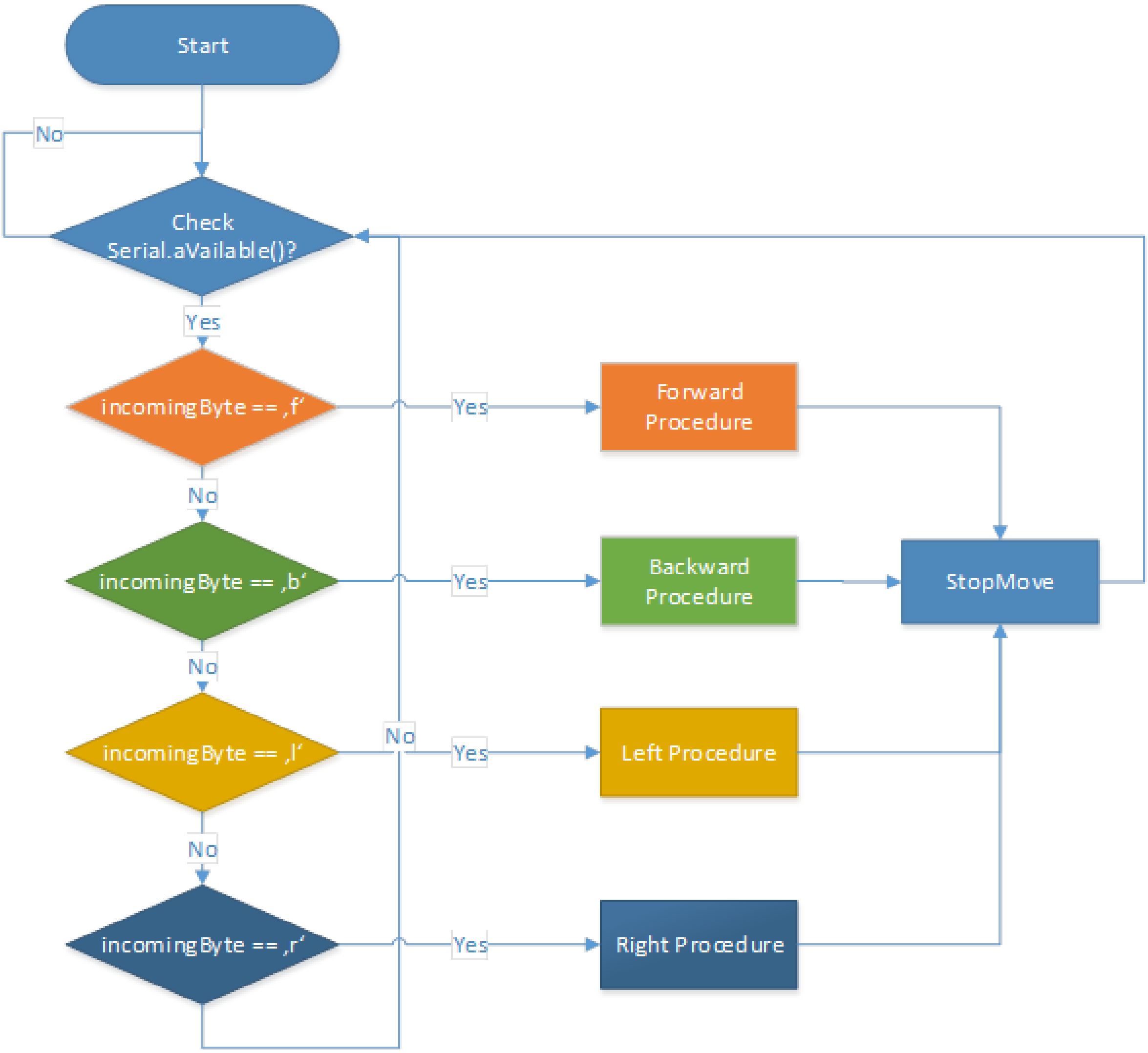


Figure IV.15. Flowchart for algorithm Bluetooth control

IV.3.3.Testing and discussing the Bluetooth control

IV.3.3.1. A Testing the Bluetooth control:

- **Android app remote control Bluetooth :**
  1) **First :**download that app from Google Play Store. as shown figure IV.16
  2) **Second:** Pair your Smartphone with the program using the app: search and pair with the HC-06 Bluetooth signal. When it is correctly paired the Red indicator will turn green, now you will be able to control the program. as shown figure IV.17

Figure IV.16. Arduino Bluetooth Rc Car

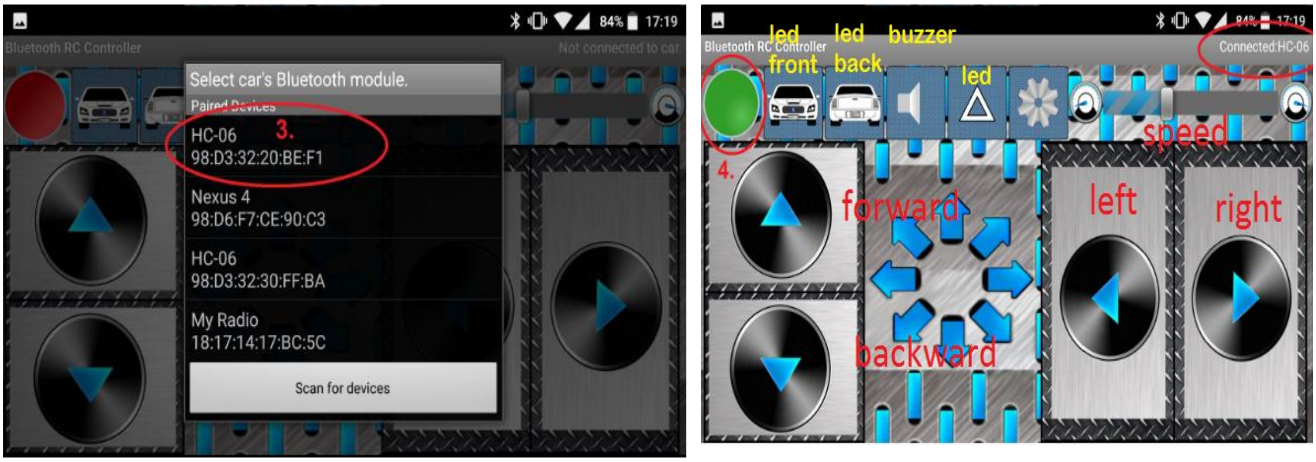


Figure IV.17. Arduino Bluetooth app configuration

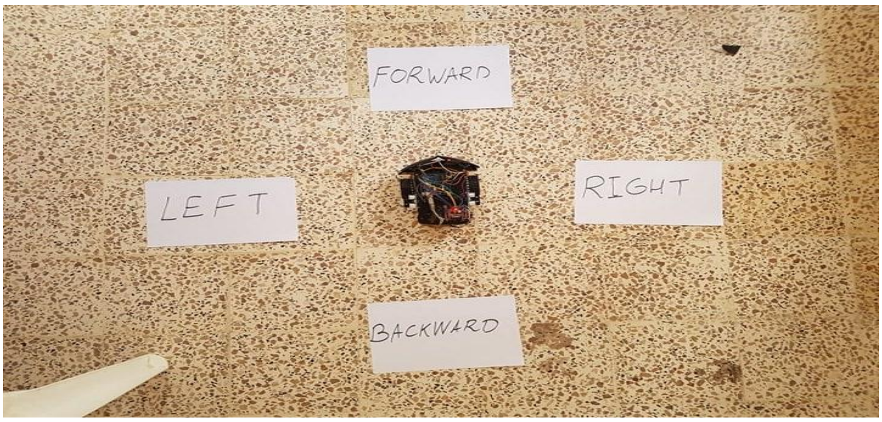


figure IV.18 Area of experiment

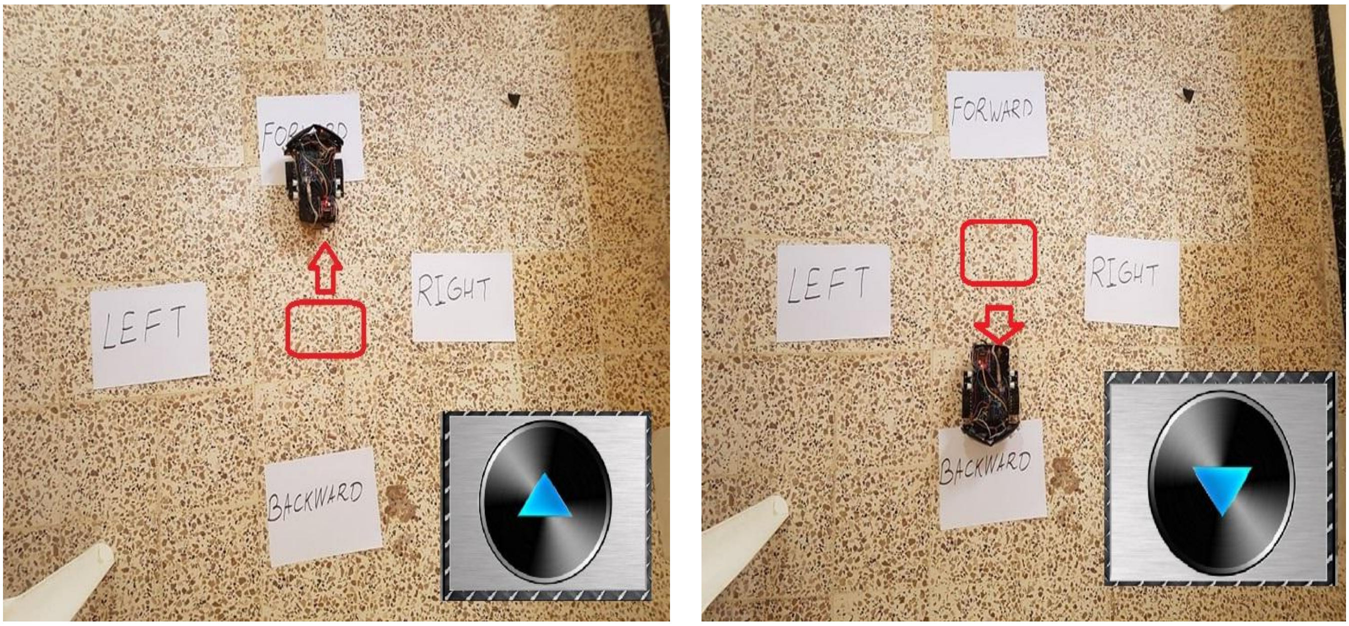


Figure IV.19 Robot moves forward/ backward

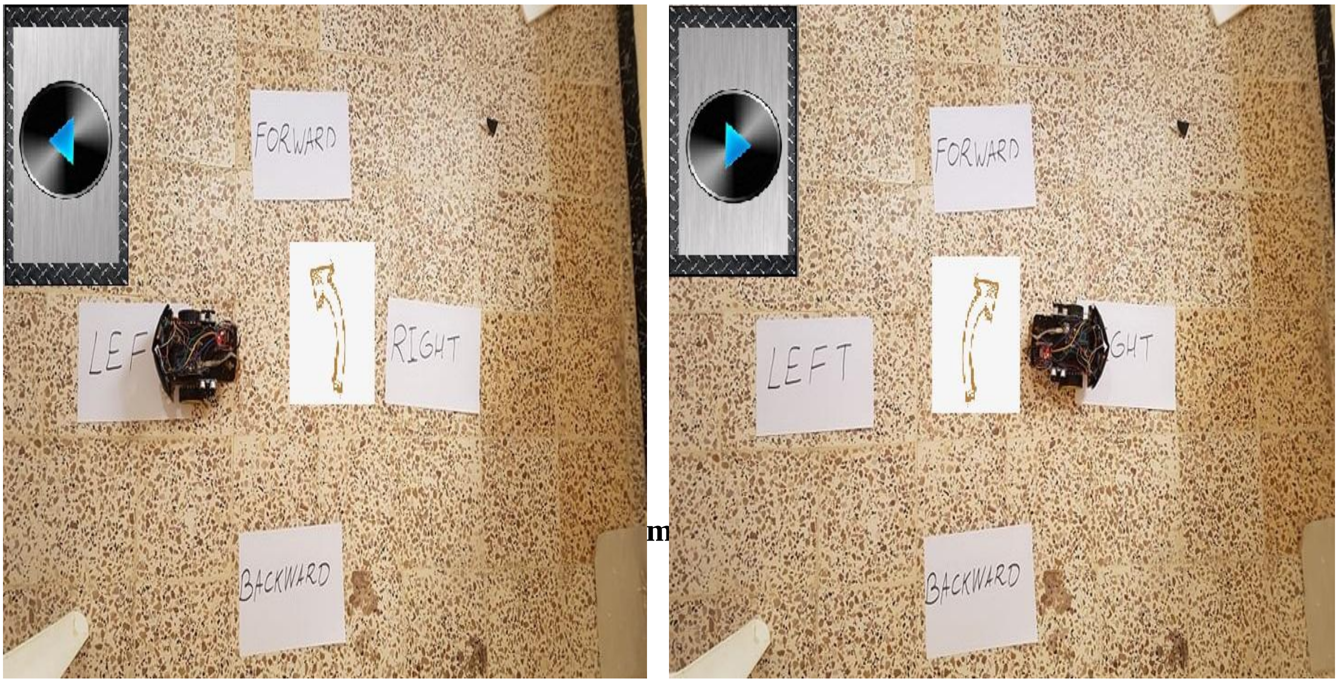


Figure IV.20. Robot rotates 90º and moves to right

IV.3.4.Discussing the Bluetooth control

- **Advantages :**
  1) Bluetooth control is a wireless control.
  2) Operating conditions are 5V and a very low current supply, which means very low power consumption.
  3) Inexpensive as other wireless control
  4) We can send and receive data from phone
- **Disadvantages :**
  1) Existence of an obstacle between the Bluetooth and the phone could hamper the control on robot .
  2) Transmission distance is 9 meters .

IV.3.5.Conclution

In the second lab, we learn how to use Bluetooth in wireless controlling. We need this to drive the robot through a maze in order to learn how to solve it (supervised learning). In the second run, the robot will solve the maze easily. Another task in an unsupervised learning is the robot map the maze and solves it autonomously and memorize the maze routes after each step. In addition, we will be using a Smartphone to see printed maze on it that arduino uno send it using Bluetooth. All these steps will be discussed in the fourth lab. After the second lab, we will see how we can make our robot independent and in no need for any human intervention.

Third Lab Session

IV.4. Autonomous Control by Ultrasonic

Autonomous control means that the robot will have a relative independency in making its decisions with regard to taking trajectories. The feature encapsulates in the fact that robot can avoid the obstacles while moving and choosing its path. HC-SR04 Ultrasonic sensor is the mean that gives the ability to know whether or not there is an obstacle ahead of the robot(we have explained this in chapter 2).

IV.4.1.Objective

- Wiring up ultrasonic sensor.
- Programming the ultrasonic sensor on arduino uno.
- Testing and discussing the autonomous control.

IV.4.2.Hardware description

- **Digital Pin Connection :**

Ultrasonic sensor HC-SR04	Digital Pin Connection
Ultrasonic front	**14,15**
Ultrasonic left	**18,19**
Ultrasonic right	**16,17**

Table IV.3 Digital Pin Connection for Ultrasonic sensor

- **Circuit diagram :**

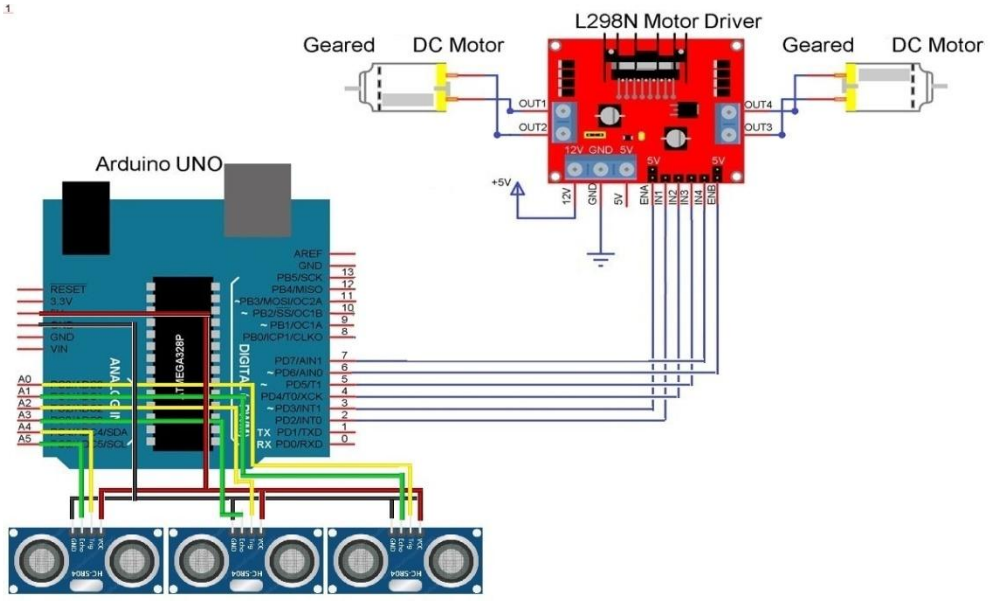


Figure IV.21. Robot rotates 90º and moves to right

IV.4.3.Webots robot simulation

IV.4.3.1.Why Webots :

Webots is an open source robot simulator as shown figure IV.19 that provides a complete development environment to model, program and simulate robots. Thousands of institutions worldwide use it for Research and development and teaching. It is the most efficient solution to quickly get professional results including:

- **The development of complex robotics systems:**

  Webots will help ass to design a new service robot, a tiny toy robot, a big agriculture robot, a vacuum cleaner, a swarm of drones, an autonomous submarine, or whatever robotics system that moves and interacts with its environment through sensors and actuators. The 3D parts we need can be imported from our favorite modeling software. Within minutes, we can add a new sensor or actuator and immediately evaluate its potential benefit on our robotics scenario. We can quickly test and validate control algorithms involving complex data processing, realistic vision and physics simulation [58] .

- **The simulation of autonomous cars :**

  Webots provides as with several tools and models to simulate autonomous cars efficiently: Open Street Map importer , Traffic generator, based on SUMO, Car and Driver programming libraries , Automobile sensors: Lidars, Radars and Cameras, Roads allowing as to build our own road circuits, Traffic lights and signs , Street furniture Buildings and Trees forests [58] .

- **The validation of new robotics research:**

  Because the Webots APIs are open, it is easy to use your favorit libraries and programming languages like c , c++ , phyton , matlab , java, to implement your research prototype. We can use Webots with OpenCV, TensorFlow, Keras, CUDA, various inverse kinematics libraries, genetic algorithms.

- **The setup of practical exercises:**

  Students are motivated by the 3D computer graphics and powerful simulation technology. We can easily setup classroom exercises, providing your student with template simulations as starting points[58] .

- **The training of human pilots:**

Webots provides user interfaces to joysticks, including driving wheels and pedals so that we can build our own driving cockpit simulator. Additionally, it also provides interfaces to virtual reality headsets . Webots has been used by automobile manufacturers to test driver behaviors. It is also used by instructors to teach how to remotely pilot real robots that operate in dangerous areas [58] .

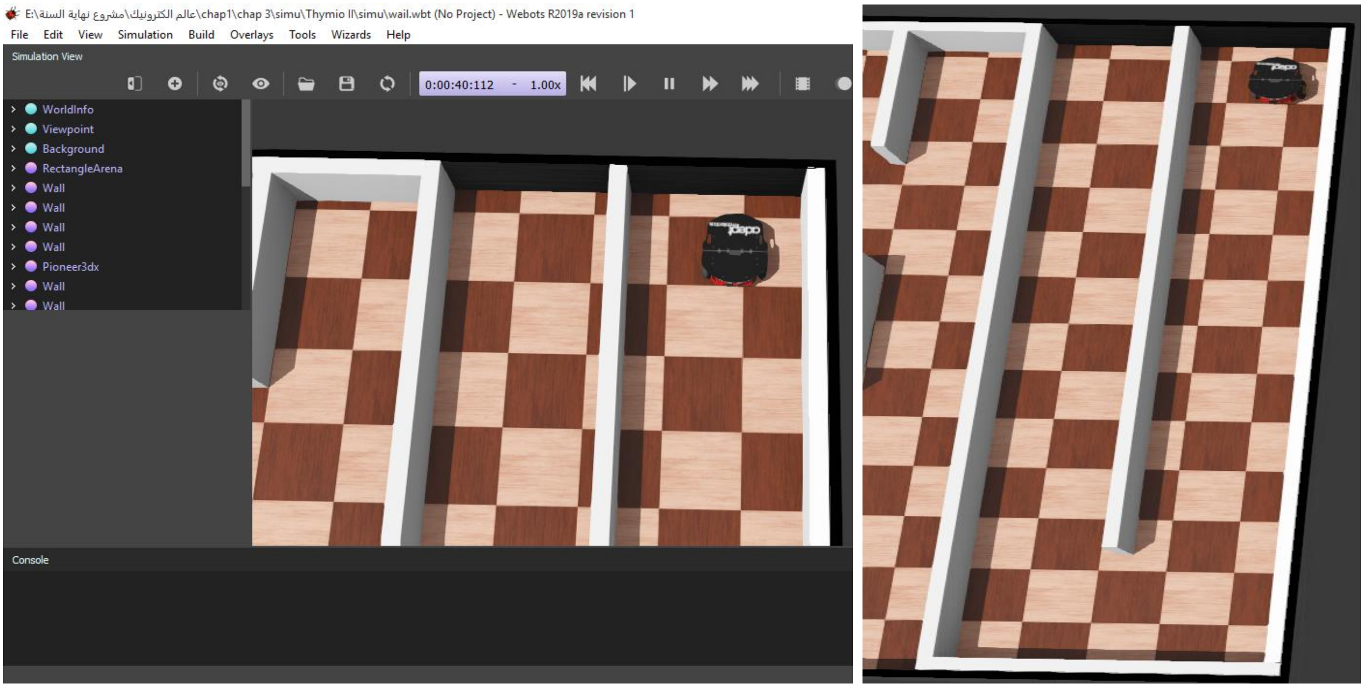


Figure IV.22. Webots simulator

Figure IV.23. Webots robot simulator

- **Flowchart for the program:**

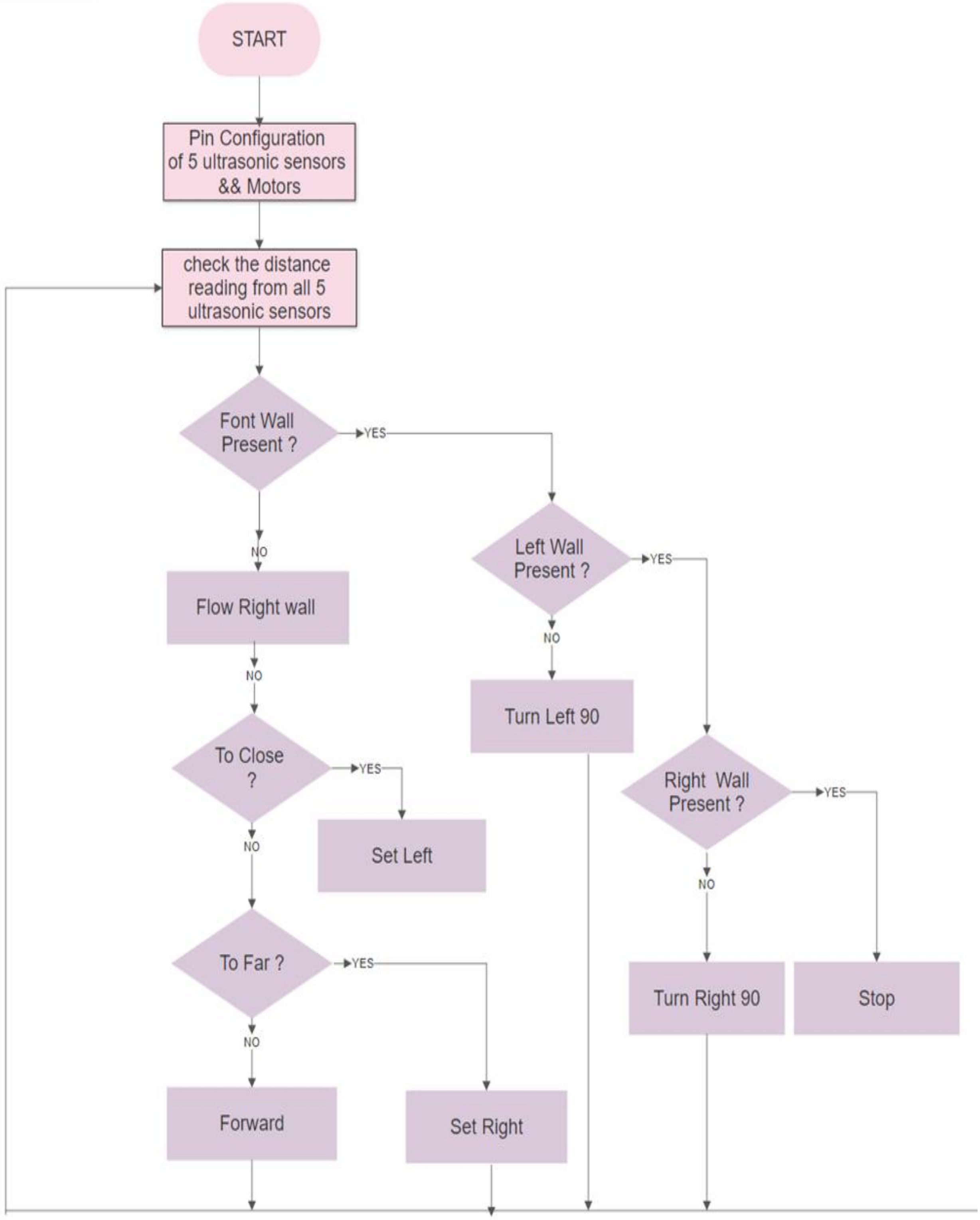


Figure IV.24. Flowchart for programe

IV.4.4.Testing and discussing the Autonomous Control

- **Testing the Autonomous control :**

Figure IV.25. Robot on middle

Figure IV.26. Robot close to right wall

Figure IV.27. Robot close to left wall

Figure IV.28. Robot close to front wall

❖ **Advantages :**

1) Work is by signals, which means no physical contact exists, unlike sensitive whiskers which work by physical contact, so this will be a good option in many life aspects.
2) Recommended operating conditions is 5V, and the current supply is very low at 15mA, which implies very low power consumption.
3) Transmission distance is 4m, and Directivity is1 5°, which is acceptable compared to the good IR characteristics.

❖ **Disadvantages :**

1) Sometimes, the sensors cannot detect the obstacles if the obstacle shape is random or if it contains a lot of bends or corners.
2) Four meters is a small detection distance, in case we want to program of robot to do a navigation process.
3) The sensors can't detect the obstacles if the obstacle face is flat but not parallel to it, and if it forms an obtuse angle with the sensor, as the follows figure:

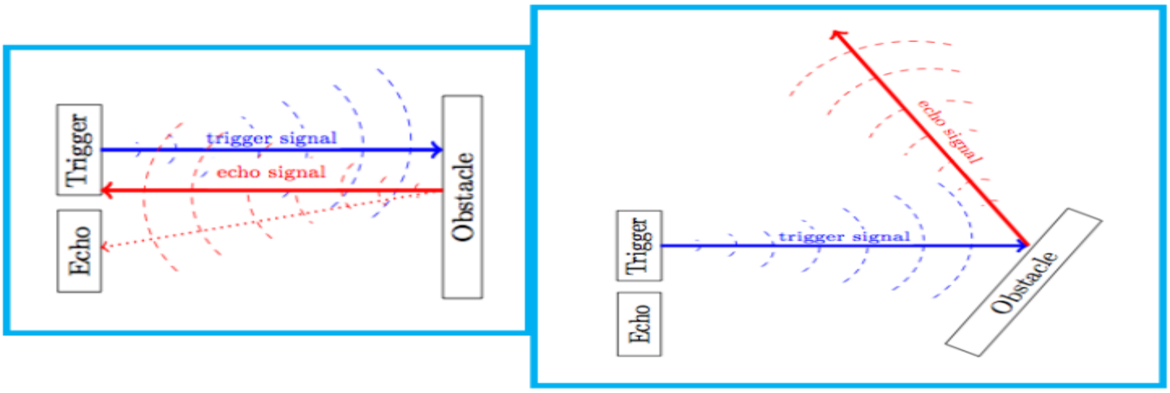


figure IV.29. obstacles detection

- **Note:** All those problem we solve it by add more Ultrasonic sensors three sensors is good, one in the middle and two on both sides.
  In the real project there is always one motor is faster than the other and the caster wheel make problem unbalance we must take attention to this problem and modify the speed of both to be equal as the follows figure .

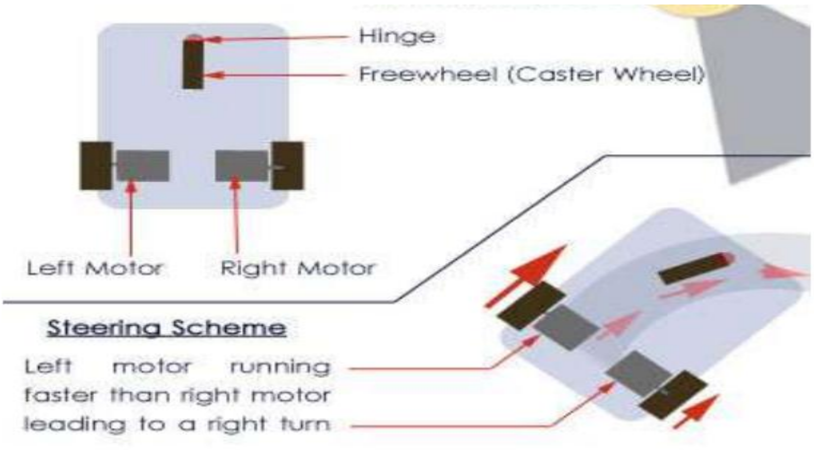


Figure IV.30. problem with motion

IV.4.5. Conclusion

We learn in this lab how to use sensors (Ultrasonic sensor) ,for make our robot detect obstacles and avoid it with south move . And we discovered amazing software that can simulate any robot with Possibility to program with Multiple programming language and also create an efficiency environment to robot .

Fourth Lab Session

IV.5.Maze solving

In mobile robotics, autonomous navigation is an important feature because it allows the robot to independently move from a point to another without human interaction . Numerous techniques and algorithms have been developed for this purpose, each having their own merits.

As a definition , Maze solving is a structured technique and controlled implementation of autonomous navigation which is sometimes preferable in studying specific aspect of the problem . This lab discusses an implementation of a small size mobile robot designed to solve a maze based on the flood-fill algorithm .The maze-solving task is similar to the ones in the Micro Mouse competition where robots compete on solving a maze in the least time possible and using the most efficient way. A robot must navigate from a corner of a maze to the center as quickly as possible. It knows where the starting location is and where the target location is, but it does not have any information about the obstacles between the two.

IV.5.1.Objective

- Learn how to build environment for robot (maze)
- Give robot the ability to make a decision and act by itself (autonomous robot)
- Implement the artificial intelligent in embedded system by applying artificial intelligent algorithm that give the robot the ability to solve the maze and identify the shortest way to solve the maze.

IV.5.2.Robot hardware and enivrement

- **Digital Pin Connection as the follows table :**

Digital Pin Connection	Components
14,15	Ultrasonic front (Tx ,Rx)
10,11	Ultrasonic right inside (Tx,Rx)
12,13	Ultrasonic left inside(Tx,Rx)
18,19	Ultrasonic left(Tx,Rx)
16,17	Ultrasonic right(Tx,Rx)
0,1	Bluetooth Tx,Rx

Table IV.4 . Digital Pin Connection

- **Circuit diagram :** In the follows figure

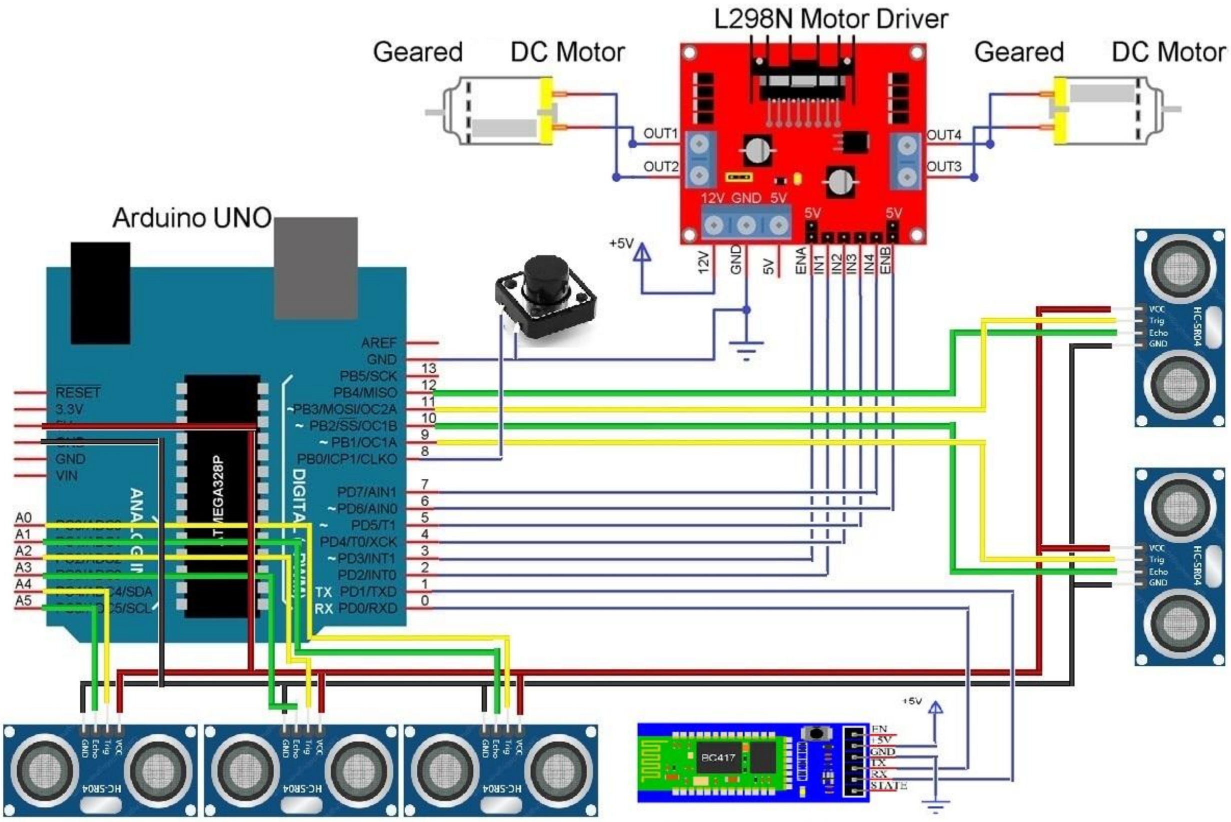


Figure IV.31.Circuit diagram for robot

- **The Maze :**

The maze designed for the robot to solve is of the size of 4×6 cells as shown in figure IV.32 The actual maze constructed, as shown in figure IV 33 The maze was designed so that it will have three paths in order for it to be solved. two of the paths is longer than the first. The robot in figure IV 33 must decide which one of the paths is shorter and solve the maze through that path .

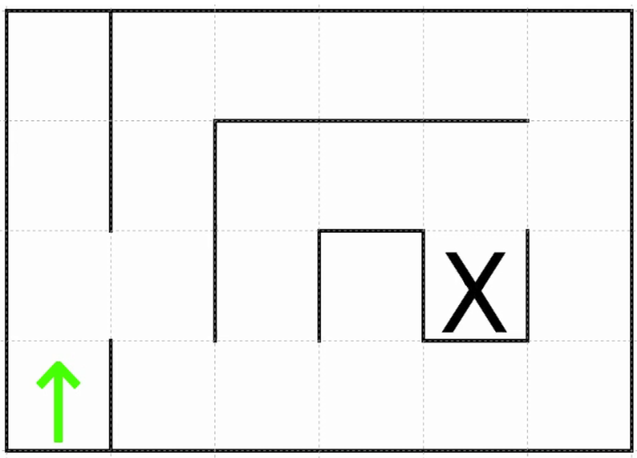


Figure IV.32. The maze designee

Figure IV.33. Real maze

IV.5.3.Algorithm

Choosing an algorithm for the maze robot is critical in solving the maze. In this lab, flood-fill algorithm was chosen to solve the maze due to its balance in efficiency and complexity. Flood Fill Algorithm: Robot maze problems are an important field of robotics and it is based on decision making algorithm. It requires complete analysis of workspace or maze and proper planning . Flood fill algorithm and modified flood fill are used widely for robot maze problem. Flood fill algorithm assigns the value to each node which is represents the distance of that node from center .The flood fill algorithm floods the maze when mouse reaches new cell or node. Thus it requires high cost updates. These flooding are avoided in modified flood fill. Flood fill algorithm is an algorithm that determines the areas that are connected to a node in a multidimensional array.

There are four main steps in the algorithm: Mapping, Flooding, Updating and Turning which are described in the following sub-sections:

- **Mapping The Maze**

  For the robot to be able to solve the maze, it has to know how big the maze is and virtually divides them into certain number of cells that can be used later in calculating the shortest path to the destination. In this lab, a maze of 4x6 cells is used. Between two cells there can be a wall .In total, there are 13 units of cells or walls. This information is stored in an 4x6 array, as shown in figure IV.34 The white units are the cells which the robot can be placed inside. The orange units are the locations for potential walls. The black units indicate wall intersections which are ignored by the algorithm. The external borders of the maze are also ignored as they are fixed boundaries of the maze. Both cells (white) and walls (orange) are set to zero in as their initial conditions.

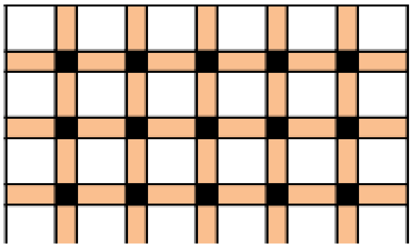


Figure IV.34 .4x6 Array for store information

- **Updating The Wall Data**

  Before the robot decides where it wants to move to, it has to check if it is surrounded by any walls in any of the three directions: right, left and front. The robot reads the distance of any obstacle at each direction and check if the distance in each is more than 20 cm. The ones that exceed 20 cm are updated as "wall" on their respective side. This is illustrated by the flow chart in For the robot to update the correct wall data, it has to know first which direction it is facing. There are four orientations for the robot to be facing: north, south ,east or west. Initial orientation was set at start and the robot keeps tracking of any changes .in the following figure flowchart for Updating The Wall Data.

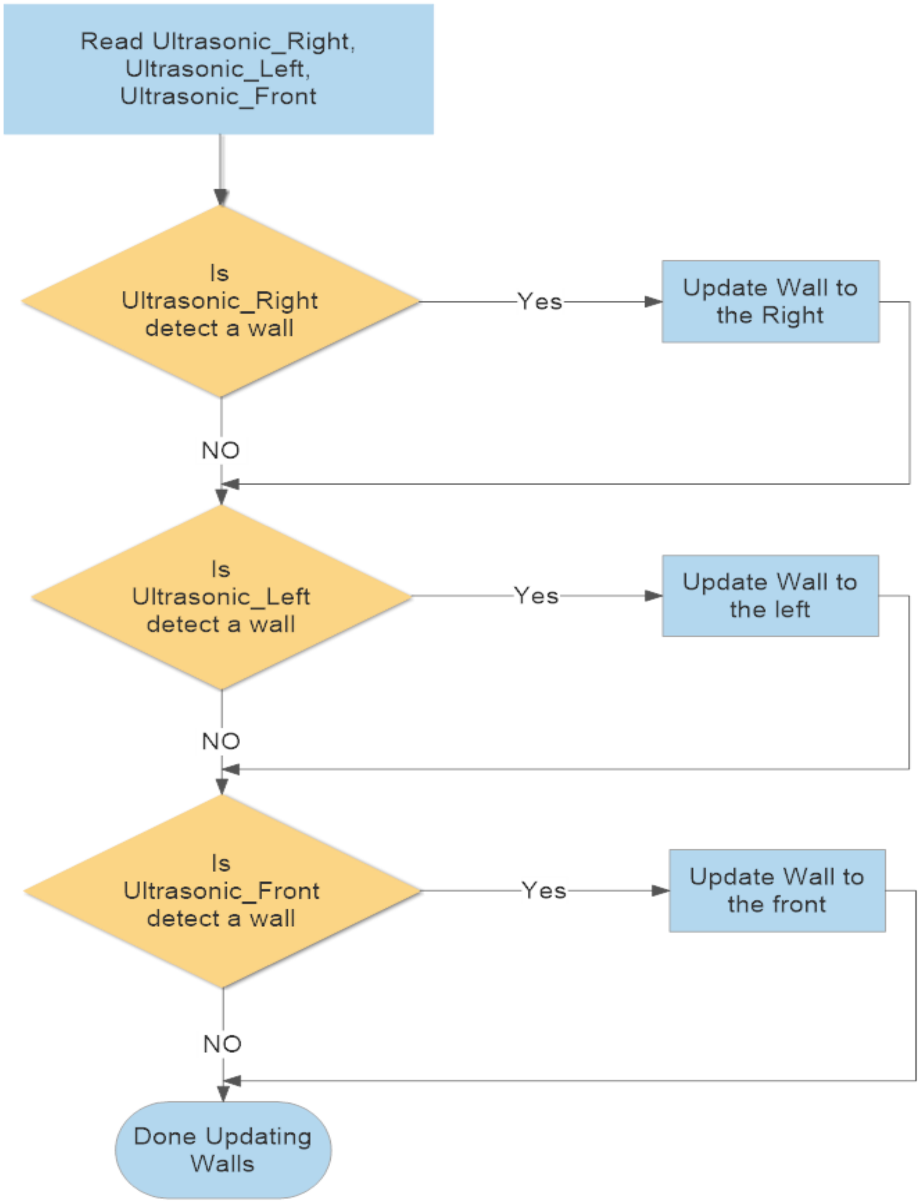
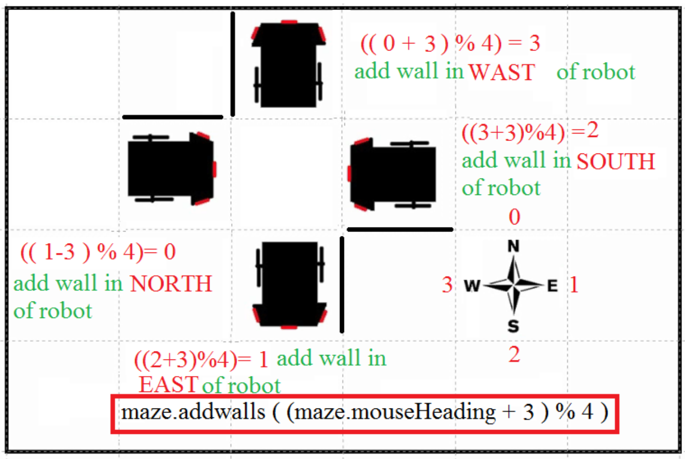


Figure IV.35.Flowchart for Updating The Wall Data

The robot know direction of the wall in proportion to the direction of the robot

- ✓ Wall in the north of the robot in figure IV.36
- ✓ Wall in the East of the robot in figure IV.37
- ✓ Wall in the West of the robot in figure IV.38



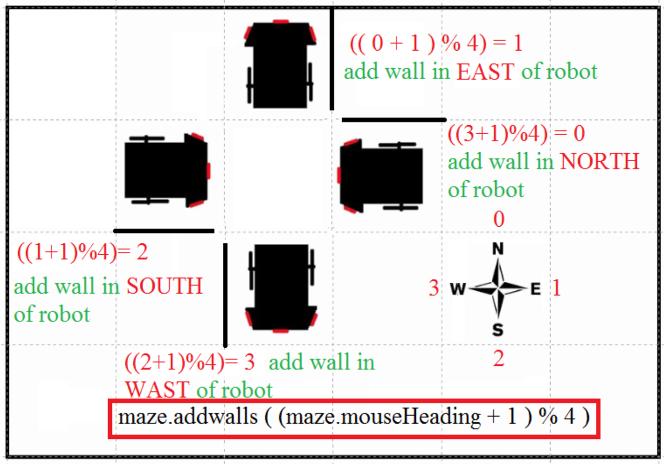


Figure IV.38. Wall in the West of the robot

Figure IV.37. Wall in the East of the robot

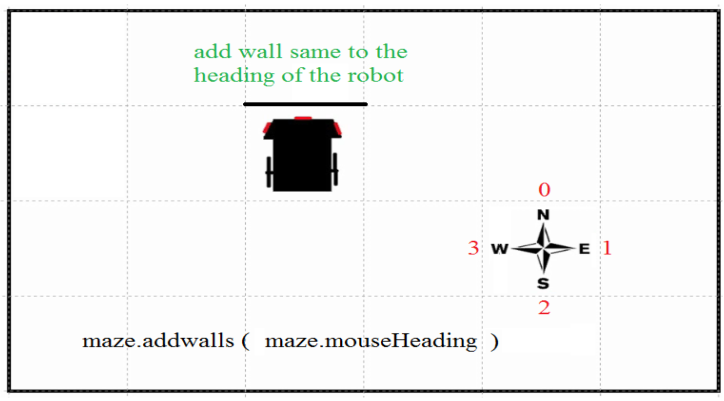


Figure IV.36. Wall in the north of the robot

- **Flooding The Maze**

  After updating the wall information for the current cell, the robot starts to flood the matrix to find the shortest path to the goal . The robot floods the matrix and makes decision by checking one cell at a time. It does the same for all the cells and keeps repeating for several times until a path between the robot and the goal is found. The algorithm assigns a value to each cell based on how far it is from the destination cell. Based on that, the goal cell gets a value of 0. If the robot is standing on a cell with a value of 3, it means it will take the robot 3 steps (3 cells) to reach the destination cell .

- **flowchart for the program as the follows figure :**

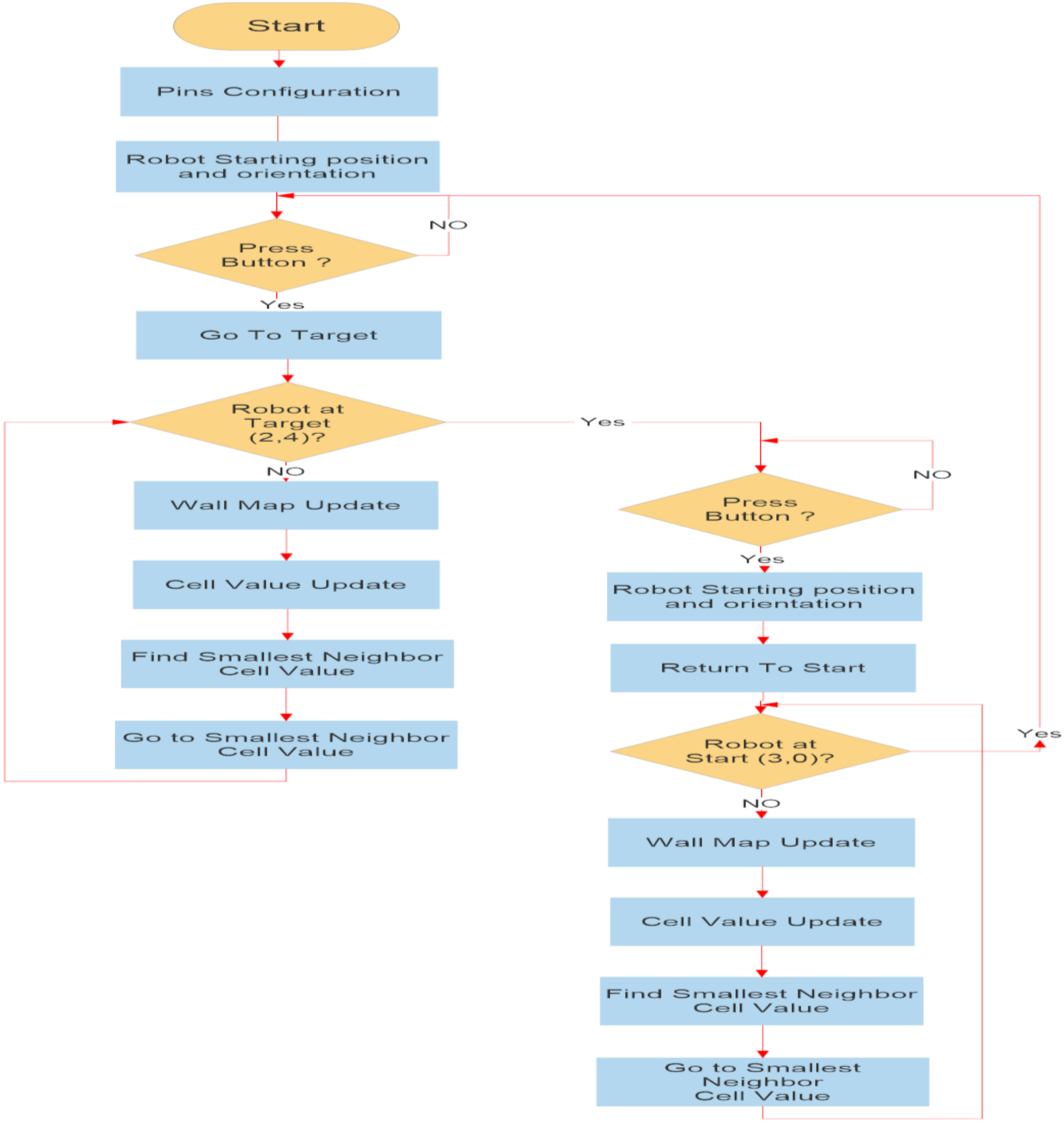


Figure IV.39. Flowchart for maze solver program

IV.5.4.Testing and discussing the maze solver

1) **First Run:** In its first run, the robot discovers the maze, map the walls and find the target location. Each time it moves to a new cell، its mapping matrix is updated with new information about the existence of walls and the distance the cell is located from the target.

```
 -----  -----  -----  -----  -----  -----
|  6      5      4      3      2      3  |
|  5      4      3      2      1      2  |
|  4      3      2      1      0      1  |
|  ^  |   4      3      2      1      2  |
 -----  -----  -----  -----  -----  -----
```

Figure IV.40. The robot is in the cell (3, 0). The robot finds a new wall on the east side. Then the robot will update the value of a cell by placing cells (2, 4) as the destination cell, so the search is done on cell lines (3, 0) to the cell (2, 4). Then the robot will move to a neighboring cell that has the smallest value that the cells (2, 0).

```
 -----  -----  -----  -----  -----  -----
|  6      5      4      3      2      3  |
|  5      4      3      2      1      2  |
|  ^      3      2      1      0      1  |
|  5  |   4      3      2      1      2  |
 -----  -----  -----  -----  -----  -----
```

figure IV.41. The robot is now in the cells (2, 0) In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, is(2, 1).

```
 -----  -----  -----  -----  -----  -----
|  6     5      4      3      2      3  |
|  5     4      3      2      1      2  |
|  6     >  |   2      1      0      1  |
|  7  |  4      3      2      1      2  |
 -----  -----  -----  -----  -----  -----
```

figure IV.42. The robot is now in a cell (2,1). In this position, the robot finds a new wall on the north side. After the robot to update the value of the cell, then the cell (2, 1) will change the value. So that the robot will move to a neighboring cell that has the smallest value, is cell but in this situation there is two cells equal East and west we set the Priority always at east ,the robot now go to (3,1).

```
 -----  -----  -----  -----  -----  -----
|  6     5      4      3      2      3  |
|  5     4      3      2      1      2  |
|  6     5  |   2      1      0      1  |
|  7  |  v      3      2      1      2  |
 -----  -----  -----  -----  -----  -----
```

figure IV.43. the robot is now in the cells (3, 1). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, is (3, 2).

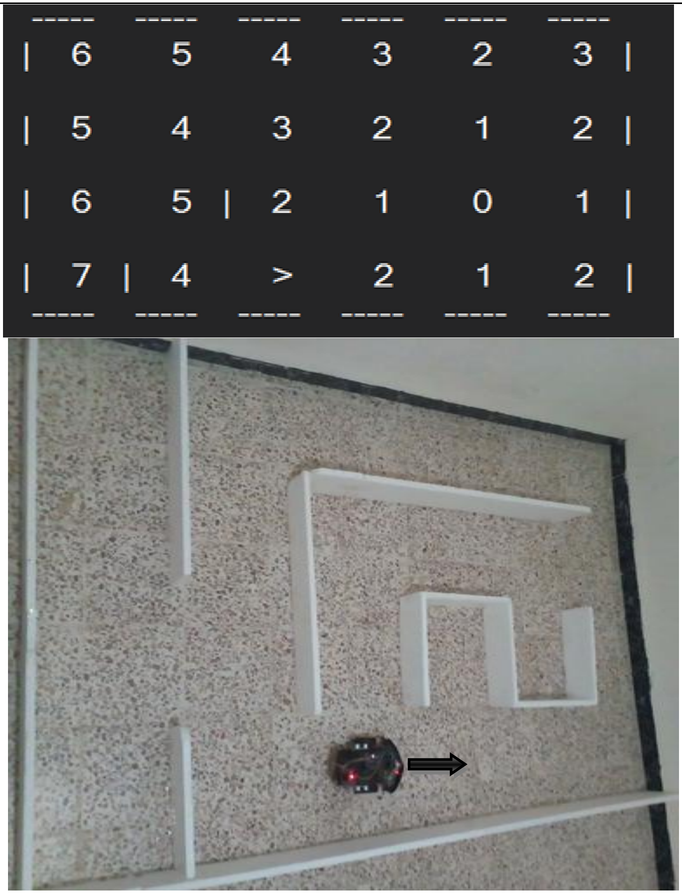


figure IV.44. the robot is now in the cells (3,2). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value ,but in this situation there is two cells equal north and west we set the Priority always at north ,the robot now go to (3,3).

figure IV.45. The robot is now in the cells (3,3). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, ,the robot now go to (3,3).

```
 -----   -----   -----   -----   -----   -----
|  6      5       4       3       2       3   |

|  5      4       3       2       1       2   |

|  6      5   |   2       1       0       1   |
                                -----
|  7  |   4       3       2       >       2   |
 -----   -----   -----   -----   -----   -----
```

figure IV.46. the robot is now in a cell (3,4). In this position, the robot finds a new wall on the wast side. After the robot to update the value of the cell, then the cell (3,4) will change the value. So that the robot will move to a neighboring cell that has the smallest value, is cell but in this situation there is two cells equal south and north we set the Priority always at north ,the robot now go to (3,5).

```
 -----   -----   -----   -----   -----   -----
|  6      5       4       3       2       3   |

|  5      4       3       2       1       2   |

|  6      5   |   2       1       0       1   |
                                -----
|  7  |   4       3       2       3       >   |
 -----   -----   -----   -----   -----   -----
```

figure IV.47. the robot is now in the cells (3,5). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, ,the robot now go to (2,5).

```
 -----  -----  -----  -----  -----  -----
|  6      5      4      3      2      3  |

|  5      4      3      2      1      2  |

|  6      5  |   2      1      0  |   ^  |
                              -----
|  7  |   4      3      2      3      4  |
 -----  -----  -----  -----  -----  -----
```

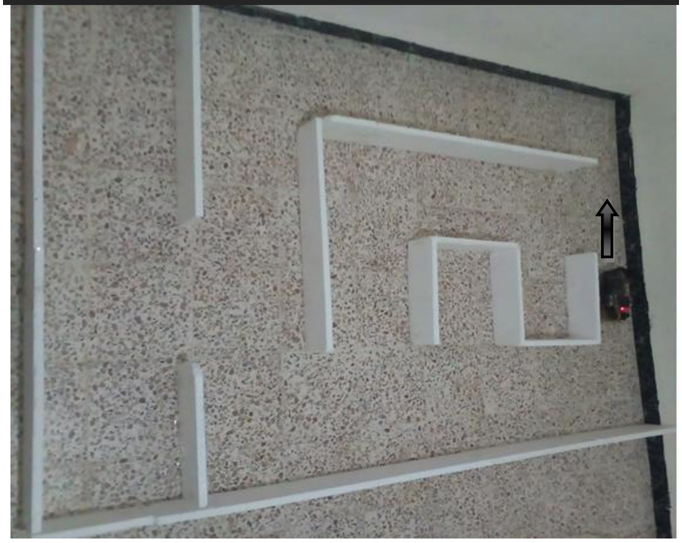


figure IV.48. the robot is now in a cell (2,5). In this position, the robot finds a new wall on the west side. After the robot to update the value of the cell, then the cell (2, 4) will change the value. So that the robot will move to a neighboring cell that has the smallest value ,the robot now go to (1,5).

```
 -----  -----  -----  -----  -----  -----
|  6      5      4      3      2      3  |

|  5      4      3      2      1      ^  |

|  6      5  |   2      1      0  |   3  |
                              -----
|  7  |   4      3      2      3      4  |
 -----  -----  -----  -----  -----  -----
```

figure IV.49. the robot is now in the cells (1,5). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, ,the robot now go to (2,4).

figure IV.50. the robot is now in the cells (1,4). In these cells, the robot does not find a new wall. So the update cell values do not cause any changes in the value of the cell. Then the robot doing the movement to neighboring cells that have the smallest value, ,the robot now go to (2,4).

figure IV.51. the robot is now in a cell (2,4). In this position, the robot finds a new wall on the east side. The robot find his destination, now he will search the path back to the cell (3, 0).

2) **return back:** In the second run the target location become starting point and the last starting point become target location.

Figure IV.52. robot in (2,4) go to (1,4)

figure IV.53. robot in (1,4) go to (1,3)

figure IV.54. robot in (1,3) go to (1,2)

figure IV.55. robot in (1,2) go to (2,2)

figure IV.56. robot in (2,2) go to (3,2)

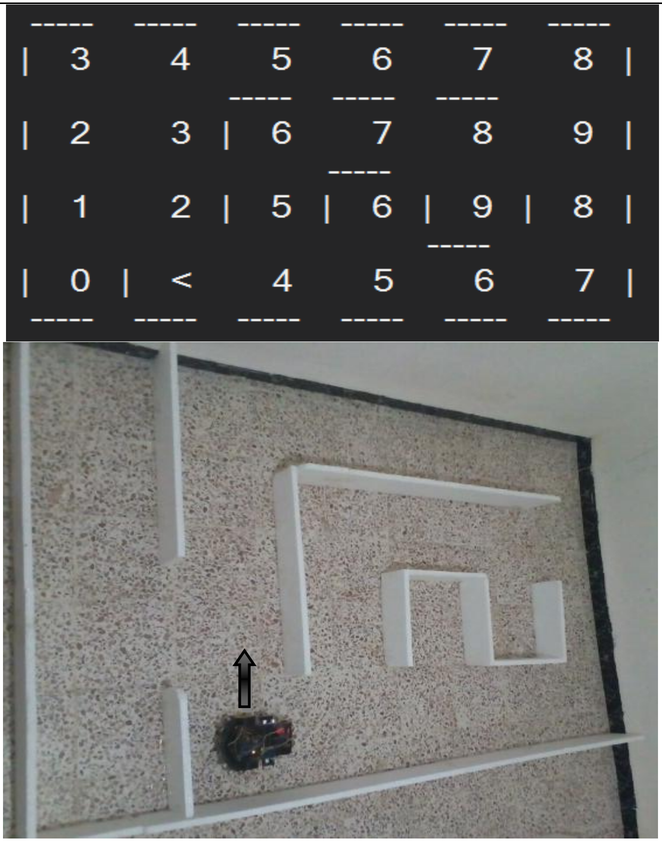


figure IV.57. robot in (3,2) go to (3,1)

figure IV.58. robot in (3,1) go to (2,1)

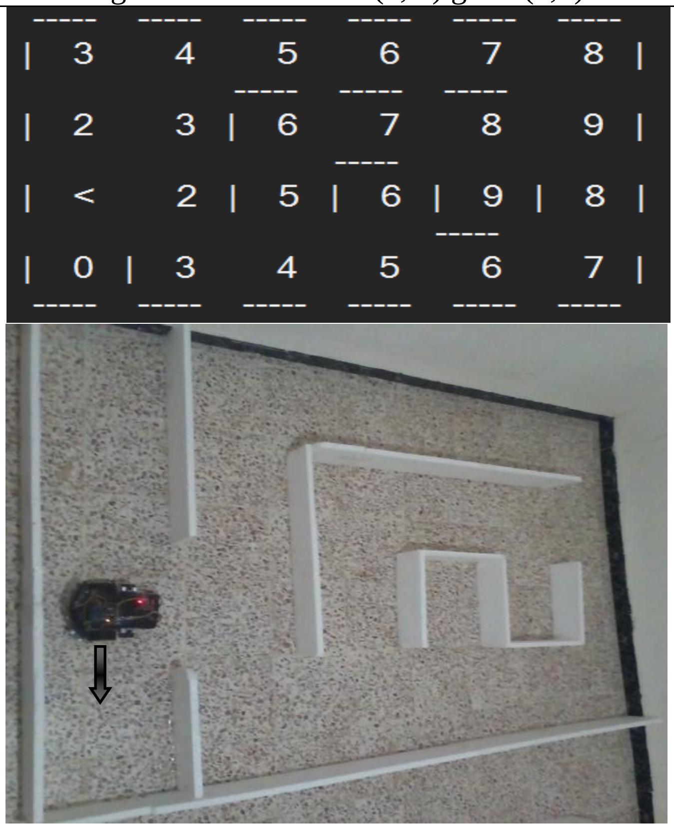


figure IV.59. robot in (2,1) go to (2,0)

3) Second run : when the maze is fully mapped, the robot navigates the maze to reach the goal through the shortest route using this data. When deciding to move the next cell, it chooses the cell that hold the smallest value. This is repeated until the target is reached

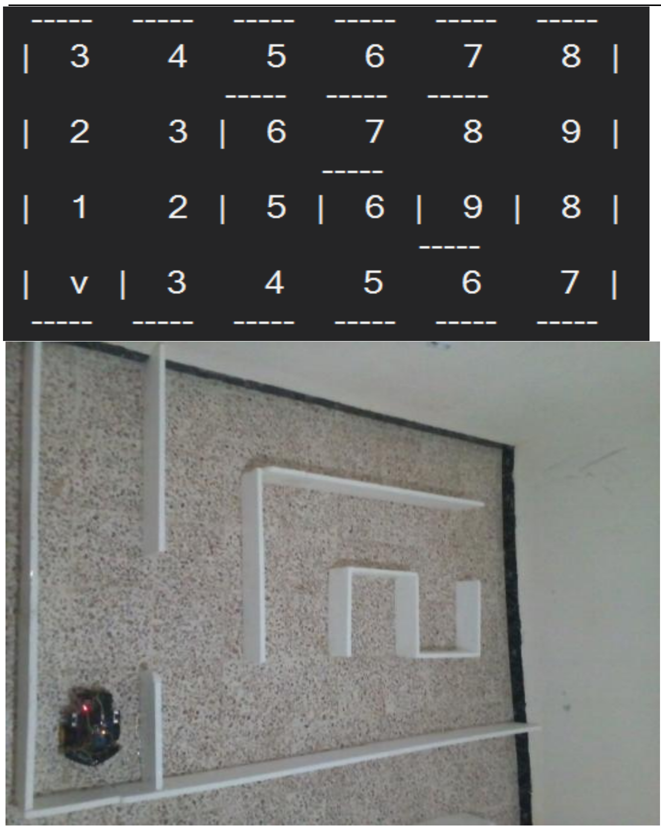


figure IV.60. robot in (2,0) go to (3,0)

figure IV.61. robot in (3,0) go to (2,0)

figure IV.62. robot in (2,0) go to (2,1)

figure IV.63. robot in (2,1) go to (3,1)

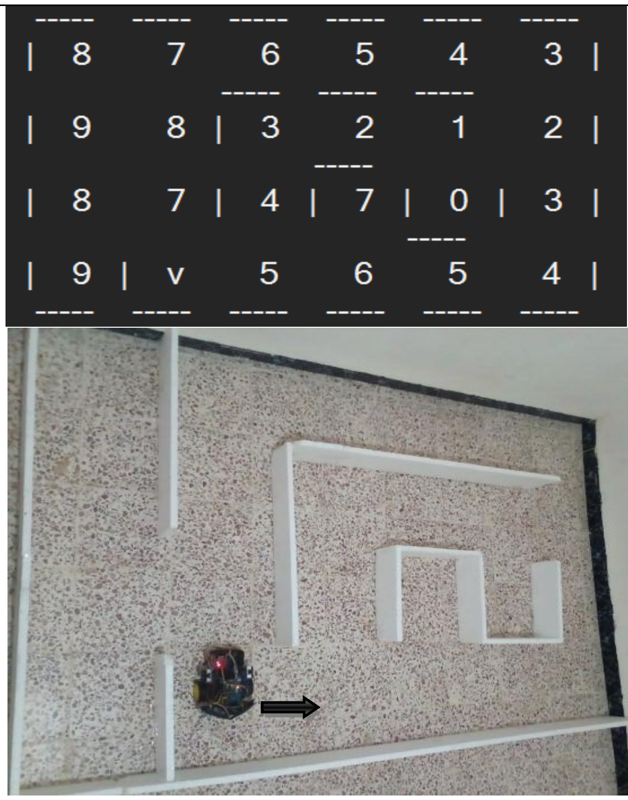


figure IV.64. robot in (3,1) go to (3,2)

figure IV.65. robot in (3,2) go to (2,2)

figure IV.66. robot in (2,2) go to (1,2)

figure IV.67. robot in (2,2) go to (1,2)

figure IV.68. robot in (1,2) go to (1,3)

figure IV.69. robot in (1,3) go to (1,4)

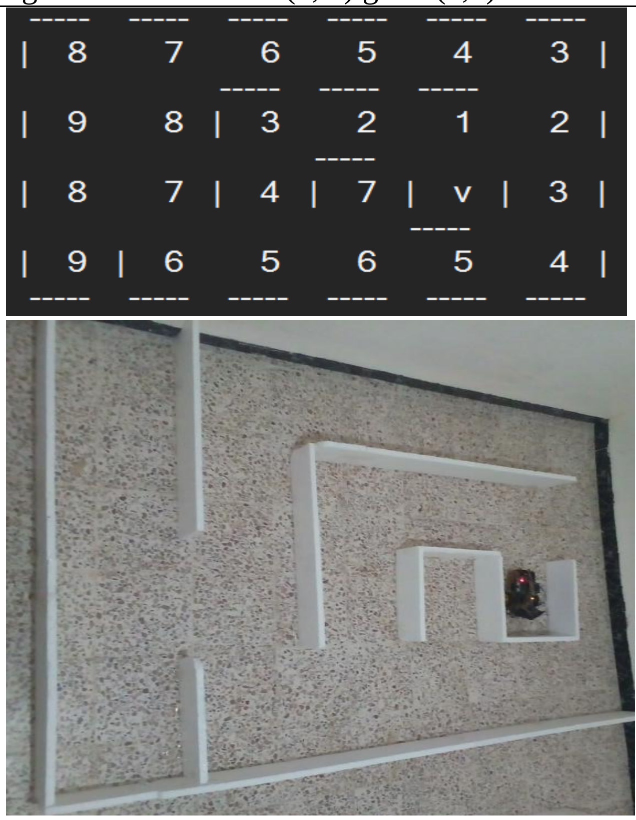


figure IV.70. robot in (1,4) go to (2,4)

Testing	Routes	Number of steps
First Run	(3,0)→ (2,0)→ (2,1)→ (3,1)→ (3,2)→ (3,3)→ (3,4) → (3,5)→ (2,5) → (1,5)→ (1,4) → (2,4)	12
Return home	(2,4)→ (1,4)→ (1,3) → (1,2)→ (2,2)→(3,2)→ (3,1)→ (2,1)→ (2,0)→ (3,0)	10
Second Run	(3,0) → (2,0)→(2,1)→ (3,1) → (3,2)→ (2,2)→ (1,2)→ (1,3)→ (1,4) → (2,4)	10

Table IV.5 . Comparison Of Stages

IV.5.5.Conclution

The robot has successfully able to map the maze in the first and second runs. In its third run it reaches its target cell through the shortest route it has mapped in the previous two runs. Future works may include but not limited to studying the robot's maze-solving capability in a bigger and more complex maze, in particular adaptable the robot is. The maze can be designed and built to be reconfigurable so that the robot will be faced with different challenge each time. In order to improve the quality in wall detection and self correction, a better object sensor, such as a laser range finder, can be employed. A laser range finder is much more costly but provides the ability to the robot to scan its surrounding at a wide angle plane instead of limited directional object detection provided by ultrasonic sensors.

All the program of labs are included in this website https://github.com/wail1995?tab=repositories

We have test Our algorithm on bigger maze in virtual

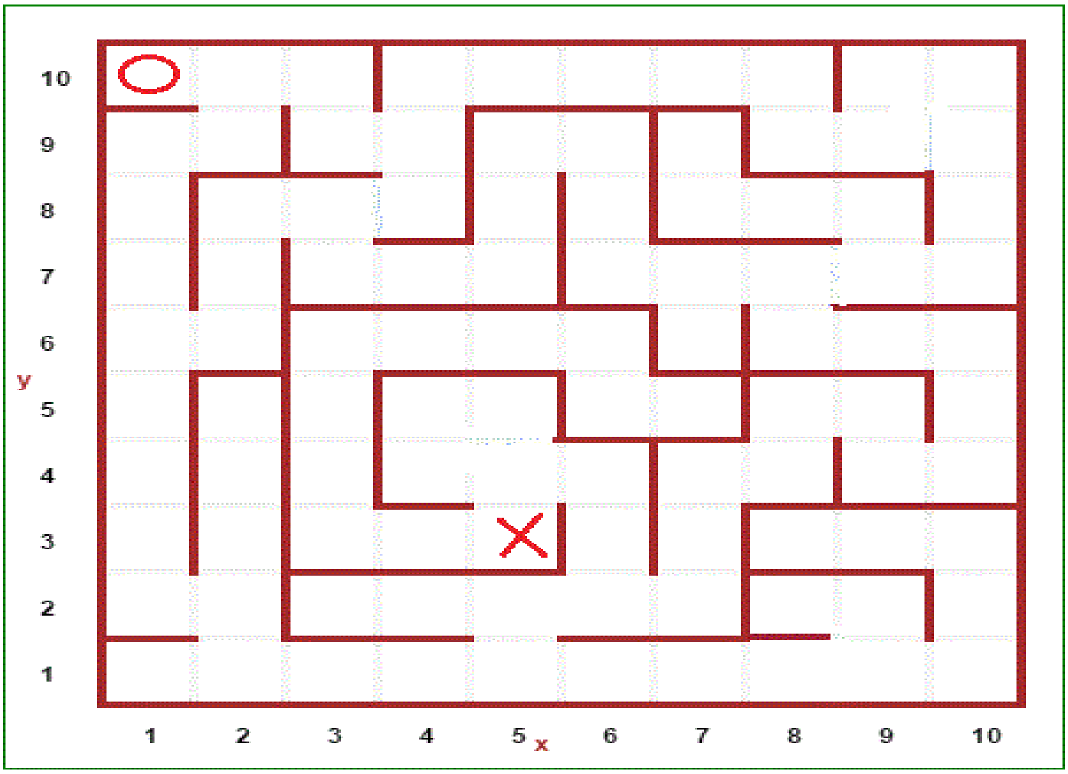


figure IV.71.Real maze

figure IV.72. The Virtual maze

General Conclusion

General Conclusion

The entire world has become more focused on electronics and programming due to the advantages and the facilities it has provided for the human-beings in all domains of life. Also, we are witnessing an advanced stage of intellectual development in the field of electronics, which has made us replace the workforce with intelligent robots that perform all human tasks. Thanks to artificial intelligence, in the next few years, these robots can create other robots, and develop themselves using artificial intelligence. This will be a unique leap in the field of scientific development.

The objective of our research project was achieved by designing a robot and programming it to explore and solve the maze and find out the shortest way out to solve it. Our robot is currently ready for use by students who are conducting research about robotics. Furthermore, the robot we made is equipped with five (5) “ultrasonic sensors" that enables it to discover the world around it, and calculate the distance between obstacles. In addition, the robot has two (2) wheels and each one has a motor with a driver that controls its movement. One of the most important elements responsible for precise control is the “mind” of the Robot which is known as “Arduino”. We wanted to enhance our project and make it more elaborate; We also planned to improve the movement of our robot and make it smoother, as well as trying it on a bigger maze, and we want to use advanced artificial intelligent like reinforcement learning ,and neural network to make our robot more intelligent .

Bibliographie

[1] Raol, Jitendra R., and Ajith K. Gopal. Mobile intelligent autonomous systems. CRC Press, 2016.

[2] Bräunl, Thomas. *Embedded robotics: mobile robot design and applications with embedded systems*. Springer Science & Business Media, 2008.

[3] http://www.robotplatform.com/knowledge/Introduction/Introduction_to_Robots.html.

[4] Zhu, Xiaorui, Youngshik Kim, Mark A. Minor, and Chunxin Qiu. *Autonomous Mobile Robots in Unknown Outdoor Environments*. CRC Press, 2016.

[5] Nourbakhsh, Illah R., and Roland Siegwart. "Introduction to autonomous mobile robots." *The MIT Press, Cambridge, Massachusetts, England, ISBN 0* 262, no. 19502 (2004): 142-150.

]6[https://www.elprocus.com/robots-types-applications/

[7] Joseph, Lentin. *Learning robotics using Python*. Packt Publishing Ltd, 2015.

[8]http://www.robotplatform.com/knowledge/Classification_of_Robots/Classification_of_Robots.html

[9] https://www.electronicshub.org/types-of-robots/

[10] Dey, Nilanjan, and Amartya Mukherjee. *Embedded Systems and Robotics with Open Source Tools*. CRC Press, 2016.

[11] Westervelt, Eric R., Christine Chevallereau, Jun Ho Choi, Benjamin Morris, and Jessy W. Grizzle. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.

[12] Böttcher, Sven. "Principles of robot locomotion." In *Proceedings of human robot interaction seminar*. 2006.

[13] http://www.robotplatform.com/knowledge/Classification_of_Robots/legged_robots.html

[14]http://www.robotplatform.com/knowledge/Classification_of_Robots/tracked_robots.html

[15]http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html

]16 [McGovern, S. T., G. M. Spinks, B. Xi, G. Alici, V. Truong, and G. G. Wallace. "Fast bender actuators for fish-like aquatic robots." In *Electroactive Polymer Actuators and Devices (EAPAD) 2008*, vol. 6927, p. 69271L. International Society for Optics and Photonics, 2008.

[17]http://www.robotplatform.com/knowledge/Classification_of_Robots/aquatic_robots.html

[18] http://www.robotplatform.com/knowledge/Classification_of_Robots/flying_robots.html

[19] Nonami, Kenzo, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer Science & Business Media, 2010.

[20] https://www.sciencenews.org/article/hybrid-robot-merges-flier-two-snakelike-machines

[21] Wilmshurst, Tim. *Designing embedded systems with PIC microcontrollers: principles and applications*. Elsevier, 2006.

[22] Barr, Michael. *Programming embedded systems in C and C++*. " O'Reilly Media, Inc.", 1999.

[23] Edstrom, Brent. *Arduino for Musicians: A Complete Guide to Arduino and Teensy Microcontrollers*. Oxford University Press, 2016.

[24] Olsson, Tony. *Arduino wearables*. Apress, 2012.

[25] Kelly, James Floyd, and Harold Timmis. *Arduino adventures: escape from Gemini station*. Apress, 2013.

[26] Evans, Martin, Joshua J. Noble, and Jordan Hochenbaum. *Arduino in action*. New York: Manning, 2013.

[27] Geddes, Mark. *Arduino Project Handbook, Volume 2: 25 Simple Electronics Projects for Beginners*. No Starch Press, 2017.

[28] Margolis, Michael. *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. " O'Reilly Media, Inc.", 2011.

[29] https://orionrkgit.wordpress.com/autonomous-robot/

[30] http://www.robotplatform.com/knowledge/sensors/robot_sensors.html

[31] McGrath, Michael J., Cliodhna Ni Scanaill, and Dawn Nafus. *Sensor technologies: healthcare, wellness and environmental applications*. Apress, 2014.

[32] http://www.robotplatform.com/knowledge/sensors/types_of_robot_sensors.html

] 33[Cox, Tim. *Raspberry Pi Cookbook for python programmers*. Packt Publishing Ltd, 2014.

]34[Cicolani, Jeff. *Beginning Robotics with Raspberry Pi and Arduino: Using Python and OpenCV*. Apress, 2018.

]35[Holland, John M. *Designing autonomous mobile robots: inside the mind of an intelligent machine*. Elsevier, 2004.

[36] De Silva, Clarence W. *Sensors and actuators: Engineering system instrumentation*. CRC Press, 2015.

[37] Scarpino, Matthew. *Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines*. Que Publishing, 2015.

[38] https://www.instructables.com/id/Tutorial-for-MD-L298-Motor-Driver-Module/

]39[Gay, Warren W., and Stewart Watkiss. *Experimenting with Raspberry Pi*. Apress, 2014.

[40] Mueller, John Paul, and Luca Massaron. *Artificial intelligence for dummies*. John Wiley & Sons, 2018.

]41[Ertel, Wolfgang. *Introduction to artificial intelligence*. Springer, 2018.

[42] https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

[43] https://learn.g2crowd.com/types-of-artificial-intelligence#JL-1

[44] https://www.quora.com/What-are-the-types-of-artificial-intelligence

]45[Awad, Mariette, and Rahul Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Apress, 2015.

[46] Brandewinder, Mathias. *Machine Learning Projects for. NET Developers*. Apress, 2015.

[47] Nicolas, Patrick R. *Scala for machine learning*. Packt Publishing Ltd, 2015.

[48] Bell, Jason. *Machine learning: hands-on for developers and technical professionals*. John Wiley & Sons, 2014.

[49] http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html

[50] Tjiharjadi, Semuil & Wijaya, Marvin & Setiawan, Erwin. (2017). Optimization Maze Robot Using A* and Flood Fill Algorithm. International Journal of Mechanical Engineering and Robotics Research

[51] Alsubaie, M. (2017). *Algorithms for Maze Solving Robot*. [online] Wordpress. Available at: https://mmalsubaie.files.wordpress.com/2013/08/mohamed-alsubaie-final-report-final.pdf [Accessed 21 Apr. 2017].

[52] Barnouti, Nawaf Hazim, Sinan Sameer Mahmood Al-Dabbagh, and Mustafa Abdul Sahib Naser. "Pathfinding in strategy games and maze solving using A search algorithm." *Journal of Computer and Communications* 4.11 (2016): 15.

[53] Wavefront and A-Star Algorithms for Mobile Robot Path Planning - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Flow-chart-of-A-star-algorithm_fig1_319404402 [accessed 26 Jun, 2019]

[54] Jonasson, A., & Westerlind, S. (2016). Genetic algorithms in mazes: A comparative study of the performance for solving mazes between genetic algorithms, BFS and DFS

[55] Choubey, N. S. (2012). A-Mazer with Genetic Algorithm. *International Journal of Computer Applications*, *58*(17).

[56] Arya, I., Gillespie, F., Kapoor, S., Mahesh, K., Ruzinov, D., & Battipaglia, J. (2017). Autonomously Solving Mazes with Robots.

[57] M. Guesbaya, , "Experimental mobile robot controlled by Rapberry Pi", Master thesis, university of biskra, 2017.

[58] https://cyberbotics.com/