UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	AHMAD AMJAD BIN AHMAD DAUD
Date of Birth	:	
Title	:	DEVELOPMENT OF VISION BASED AUTOMATED GUIDED VEHICLE
Academic Session	:	2021/2022

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Supervisor's Signature)

(Student's Signature)

Name of Supervisor
Date:

New IC/Passport Number
Date:

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	AHMAD AMJAD BIN AHMAD DAUD
Thesis Title	DEVELOPMENT OF VISION BASED AUTOMATED GUIDED VEHICLE
Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

MAKLUMAT PANEL PEMERIKSA PEPERIKSAAN LISAN

(only for Faculty of Computer's student)

Thesis ini telah diperiksa dan diakui oleh
This thesis has been checked and verified by

Nama dan Alamat Pemeriksa Dalam :
Name and Address Internal Examiner

Nama dan Alamat Pemeriksa Luar :
Name and Address External Examiner

Nama dan Alamat Pemeriksa Luar :
Name and Address External Examiner

Disahkan oleh Penolong Pendaftar IPS :
Verified by Assistant Registrar IPS

Tandatangan : *Signature*		Tarikh : *Date*	
Nama : *Name*			

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in ....

____

(Supervisor's Signature)

Full Name	: SAIFUDIN BIN RAZALI
Position	: SENIOR LECTURER
Date	:

____

(Co-supervisor's Signature)

Full Name	:
Position	:
Date	:

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student's Signature)

Full Name	: AHMAD AMJAD BIN AHMAD DAUD
ID Number	: TG18031
Date	: 27 JANUARY 2022

DEVELOPMENT OF VISION BASED AUTOMATED GUIDED VEHICLE

AHMAD AMJAD BIN AHMAD DAUD

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Electronics Engineering Technology (Computer System) with Hons.

Faculty of Electrical & Electronics Engineering Technology

UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2022

ACKNOWLEDGEMENTS

In the name of Allah, the Most Merciful and Gracious. First and foremost, I want to thank Allah, the Most Merciful and Compassionate, for providing me with the ability and willingness to begin and accomplish my endeavour. I pray to Allah to continue to inspire and empower me to strive for the welfare of humanity.

First and foremost, I'd want to show our gratitude to our supervisor, Dr. Saifudin Bin Razali, for providing us with the opportunity to demonstrate our engineering abilities and guiding me through an exciting final year project. His guidance has helped me channel and apply our knowledge and ideas while also giving me the freedom and support to experiment with new ideas.

My gratitude also goes to the person who has always seen my potential and encouraged me to pursue it. Their compassion and affection have given me the motivation to always do my best since there is always someone who encourages me to keep going.

Furthermore, we would like to express our gratitude to the Faculty of Electrical and Electronic Engineering Technology at University Malaysia Pahang (UMP) for providing us with the opportunity to participate in a project that has improved our understanding of the engineering world. I'm hoping that all of the experience and information I've received as a result of this project will be really useful in my future profession.

I would also like to thank my loving family members, particularly my parents, who brought me into this world, showered me with boundless love and care, and helped me go through the most difficult phase of my life yet with their encouragement and financial support during the study.

ABSTRAK

Kenderaan berpandu automatik (AGV) adalah bahagian penting dalam sistem pengangkutan logistik terutamanya dalam industri besar. Mereka biasanya digunakan untuk memudahkan pekerja atau majikan terutamanya dalam tugas berat di kilang seperti industri automotif. Keperluan asas untuk robot mudah alih bergerak secara autonomi adalah bahawa ia memahami kedudukan semasanya berhubung dengan persekitarannya, yang disebut sebagai penyetempatan robot mudah alih. Walau bagaimanapun, dalam aplikasi dunia sebenar, gangguan dan gangguan alam sekitar yang pelbagai menghalang penyetempatan robot mudah alih. Matlamat projek ini adalah untuk memetakan dan mengimbas lokasi robot mudah alih dalam persekitaran yang diketahui, statik, dan dalaman. Sensor lidar, jetson nano dan sensor ultrasonik digunakan dalam projek ini untuk menguruskan sasaran untuk pemetaan dan mengimbas alam sekitar terutamanya dalaman. Dengan menggunakan jetson nano sebagai 'otak' projek ini, ia boleh mengawal prototaip atau model untuk berfungsi sensor lain untuk menguruskan pemetaan dan pengimbasan semasa prototaip bergerak. Sensor Lidar dirujuk sebagai "pengimbasan laser" atau "pengimbasan 2D." Kaedah ini mewujudkan gambaran 3D kawasan yang dikaji menggunakan rasuk laser yang selamat mata. Kajian ini dilakukan menggunakan robot mudah alih dua roda untuk mengumpul data mengenai alam sekitar dengan memetakan dan mengimbas masa kemas kini dari masa ke masa untuk mendapatkan pengimbasan penuh di beberapa ruang. Sensor ultrasonik dan lidar digunakan dengan berkesan untuk meningkatkan ketepatan anggaran penyetempatan robot mudah alih yang menggunakan sistem pemacu dan model pengukuran yang dicipta.

ABSTRACT

Automated guided vehicles (AGV) are important part of the logistic transport system especially in big industry. They are used commonly for ease the worker or employer especially in heavy task in factory such as automotive industry. The basic requirement for a mobile robot to move autonomously is that it understands its current position in relation to its surroundings, which is referred to as mobile robot localization. In real-world applications, however, interferences and diverse environmental disturbances hinder mobile robot localization. The goal of this project is to mapping and scanning the location of a mobile robot in a known, static, and indoor environment. The lidar sensor, jetson nano and the ultrasonic sensor are used in this project to manage the target to mapping and scanning the environment especially indoor. By using Jetson Nano as a 'brain' of this project it can control the prototype or model to functioning other sensor to manage the mapping and scanning while the prototype move surrounding. Lidar sensor referred to as "laser scanning" or "2D scanning." The method creates a 3D depiction of the studied area using eye-safe laser beams. The studies were done out utilising a two-wheeled mobile robot to collect data about the environment by mapping and scanning update time by time to get the full scanning in some space. The ultrasonic and lidar sensor was effectively used to increase the estimation accuracy of the mobile robot localisation utilising the created drive system and measurement models.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGV	Automated Guided Vehicle
LiDAR	Light Detection and Ranging
LPR	Local Positioning Radar
DOF	Degree of Freedom
2WD	Two Wheel Drive

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

Automated Guided Vehicle (AGV) conceived in 1954 by Barret Electronics Corporation, Illinois, a manufacturing company in the United States. Since AGV became widespread all over the globe in every logistics environment where goods are transported. Currently, the market of AGV is growing fast and is very dynamic. A market report published by Grand View Research has made a forecast for the period from 2018 to 2025 focusing on the potential growth opportunities of AGV. Despite AGV were invented in the United States, this proposal state that the biggest market of AGV is now situated in Europe. In this system a fleet AGV is organized in centralized way. Task like path planning and allocation of tasks are done by a central entity for all the AGV together. But driven by future requirements like flexibility, robustness, and scalability, the current trends in AGV systems are customization and decentralization as one of the future technologies are dual mode (hybrid) AGV which can operate both manually and automatically, camera vision, Light Detection and Ranging (LiDAR) for navigation, and collision avoidance. The most important reasons why this shift is necessary is the enlargement of the AGV fleet. Over the years, an enormous amount of literature has been accumulated in the research on AGV.

1.1 PROJECT BACKGROUND

Automated guided vehicles (AGV) are commonly used in facilities such as manufacturing plants, warehouse, distribution centre and transhipment terminals. AGV can be referred as mobile robots owing to their reprogram ability. The purpose of AGV is to help reduce costs of manufacturing and increase efficiency in a manufacturing system. It also involved the movement of tools, raw material and work in process between stations or into the storage. These movements must be safely, accurately, efficiently and without any damage to the materials. It is an important system and in element to integrate manufacturing facilities.

In this project to build an AGV to transport the acrylate during laser cutting process, project team have been divided into three different specialize which is, loading and unloading mechanism, AGV control system and mechanical part of AGV. The essential capability of this AGV is ability transfer loads to location through path under computer control by programming. Considering the full potentials and advantages of the Automated Guided Vehicle (AGV) in our livings, it is valuable to do this project, as it also will be the first step towards the creation of more intelligent technology or system. The simplest AGV model may use just a sensor to provide its navigation and can be the complex one with more sensors and advance systems to do the task. They can work or do the task everywhere needed but the safety for the AGV as well as the people and environment surround it must be provided.

The AGVs is just the same as mobile robot, which can move from one place to another to do their task, but mostly the mobile robot is used for difficult task with dangerous environment such as bomb defusing. Furthermore, the mobile robot can be categorized into wheeled, tracked, or legged robot. Although the AGVs may not be glamorous of robots, but their work, which usually menial, are often be essential to the smooth running of factories, offices, hospitals, and even houses. They can work without any complaint around many workplaces all over the world.

1.2 PROBLEM STATEMENT

There are many reasons which yield to the creation of Automated Guided Vehicle (AGV) around the world. Mostly the reason is to overcome the logistic problems that often occurred in the workplaces and to make improvement to the facilities provided in the workplaces. Usually, the AGVs are implemented in factories, hospitals, offices, houses, and even can be found anywhere outdoors without the people surround realized it.

In the industries or factories, the AGVs can ease the physical strain on human workers by performing tiring tasks, such as lifting and towing heavy materials, more efficiently with no signs of fatigue creeping in. They can carry far more than human workers, and their movements can be tracked electronically at all times. Their movements can be timed to feed or collect products or materials from the workcells in the factories.

Besides that, in the hospitals thousands of staff spends a portion of their day moving medical supplies, bedding, medicines and other equipment around large hospitals. By using the AGVs, the strain on the workers can be ease as well as the hospital's system, heavy factory system would be smart and systematic without any bad complaint from the patients and people. AGVs also capable of both cutting cost and releasing more staff hours to tend and care for patients.

Therefore, it is very significant that the valuable knowledge on AGV construction is studied and be further implemented from the result of this project. It is due to its advantages to our own living and technology.

1.3 OBJECTIVE

The objective of this project is to create an AGV model that can automatically move by using vision system. In order to complete this objective, we divided this project into three parts. First part contributes to design and fabrication of the basic mechanism of AGV controlled by arduino and equiped onboard sensors and actuator such as imu, ultrasonic and bumper. All basic mechanism specifications of an AGV will be considered. Second part is to implement Robot Operation System (ROS) to communicate and control AGV. There are several main stages consists in this project such as software design, electronic hardware, theoretical design, algorithm design in assembly language, phyton, c++ programming, mapping algorithm and lidar sensor. The issue to examine is how the robot can mapping and scanning its surroundings. Thirdly, to set up a mobile robot platform for localization in known,static and indoor environment.So, it is like human brain that we programme to scan the environment, but it is limited to the function we give this robot. It is also critical to select the best microcontroller, actuators and sensor to achieve the project goals.

1.4 RESEARCH SCOPE

The model is a four-wheeled and two motor wheel mobile robot that has the ability to navigate in one area and automatically move by using the vision sensor such as LiDAR. In addition, there are use ultrasonic sensor to detected any movement or resistance to prevent any collision or damage to our robot.

This project consists of several main stages, which are theoretical design, mechanical build, electronic hardware and software design, algorithm design in assembly language and mapping algorithm. The matter to be considered is how the robot can navigate and scan surrounding. So, it is like similar to human brain that we program to look surrounding. It is also important to choose the most suitable microcontroller, actuators, and sensors to achieve the project objectives.

CHAPTER 2

LITERATURE REVIEW

INTRODUCTION

This chapter will design some of the Automated Guided Vehicles (AGVs) that are well known are discussed in brief. This chapter we discussing on some literature that give information about automated guided vehicle (AGV) and show how this AGV have been fabricated.

2.1 AGVs CORE TASKS

A complete AGV-system consists of multiple core tasks to be able to operate in complex environments. If we take an e-commerce warehouse as a running example then these core tasks can be easily label. In such an e-commerce situation, there is a continuous stream logistical warehouse system. An order can be seen as an object somewhere in the warehouse that needs to be brought to the picking station where the order can be sent to the customer. In this paper, the authors divide the total AGV-system in the following five core AGV-tasks: (1) Task Allocation [1][2], (2) Navigation [5][6], (3) Path planning [16], (4) Motion planning[16][25], (5) Vehicle Management.

A first core AGV task is Task Allocation. A set of tasks (orders) which has to be distributed to the fleet of AGVs need to be allocated to a specific AGV in an optimal way. The easiest way to solve this is to allocate the task to the AGV which is closest to the position of the ordered object. Once a task is allocated, a next core AGV task is used to find the shortest path to the destination. This task is named Path Planning. It uses a representation of the environment to search for a sequence of segments to reach the goal as fast as possible. For Path Planning, it is important that the AGV can navigate properly in its environment. Thus, Navigation is also an important core AGV task. If a Path

Planning algorithm computes the shortest path for an AGV, this does not mean that the AGV is able to follow that path without any problems.

An unforeseen object or person can block the path or other AGVs may need some segments of the path at the same time. To avoid collisions or situations where multiple AGVs enter a life- or deadlock situation, there is another core AGV task named Motion Planning. This planner tries to avoid collisions with other static or dynamic objects. It tries to avoid deadlock situations and tries to limit the number of vehicles in a particular area. Limiting the numbers of vehicles in an area is called zone control. Once the collision-free path is executed and the AGV reaches its destination, the object can be loaded on the AGV. The exact same tasks are then used to bring the loaded object to the picking station. Parallel with all these core AGV tasks, there is another core task, vehicle management, which controls and monitors the status of an AGV. Some management issues are battery lifetime, maintenance requirements, and error status handling. In the following sections, every core AGV task will be treated and relevant algorithms and techniques will be described.

2.2 TASK ALLOCATION

Task allocation is one of the most challenging AGV tasks. It has as a purpose to assign a set of tasks to a set of robots. This is a constrained optimization problem in which we want the cost of the total assignment to be as low as possible. The optimization of this problem is NP-hard [1], which means that if we have a lot of tasks and robots, the number of solutions will be enormous.

So there does not exist an efficient algorithm which can produce and exact solution to the problem in a finite amount of time. To solve these NP-hard problems, we use approximate optimization techniques named heuristics and meta-heuristics[2]. A task in task allocation can be an object that has to be picked up at a certain location and has to be dropped off at another location in a factory. Another task can be a surveillance task in which a group of AGVs must scan an area in the most efficient way. In the following sections, some properties and solution methods are described.

A. Desired properties

In task allocation for AGV-systems, there are some desired properties [3] to be considered. One of these properties is divisibility. It is important that the total work is divided in an efficient way. The purpose is to have a high usage of every resource. We don't want busy AGVs on one side and idle AGVs on the other at the same time. Another important desired property is fault tolerance. The task allocation must operate correct no matter what failures are faced. To meet future system requirements, some additional important properties are scalability, flexibility, and responsiveness.

• Scalability is the way the system can be enlarged without problems. The task allocation algorithms must keep working with a bigger amount of AGVs without reaching memory or computation limits.

• Flexibility means that the algorithm should, in any case, continue operating by continuously adapting to changes in the system.

• Task allocation should also be responsive; this means that it must have high performance also in dynamic environments. To take all these desired properties into account, proper algorithms are developed. Section 2.2.E covers these algorithms.

B. `Taxonomy of tasks every application requires another kind of task allocation. Gerkey and Mataric [4] have proposed a widely accepted taxonomy for task allocation in multi-robot systems. They divide the tasks into the following categories:

• Single robot tasks (SR): Tasks which only need one robot to be completed.

• Multiple robot tasks (MR): Task which needs more than one robot to be completed.

• Single task robots (ST): Robot which can only perform one single task at a time.

• Multi-task robots (MT): Robots which can perform more tasks simultaneously.

• Instantaneous assignment (IA): Tasks are independent of each other and there is no planning for future allocations. The available information about the task only permits an instantaneous allocation.

• Time-Extended assignment (TA): Tasks are dependent one each other. Future allocations can be planned considering several constraints.

C. Task Constraints

In some applications, tasks are independent of each other and are assigned to a robot from the moment the task is available. The only relevant information about the task is a starting and an ending point. After assignment, the robot can directly execute the task knowing this information. This can be the case in a warehouse environment where an order enters and the task for the AGV is to get the ordered object and bring it to the picking station. Knowing the basic information, the task can be assigned to the robot which is closest to the task. This is the simplest case. But in real-world applications, we are dealing with several constraints [3]:

• Tasks can have a time window in which tasks can have a duration, minimum starting time, and maximum ending time (deadline). These are time constraints related to the specific task and are called temporal constraints.

• There are also constraints which cause that tasks are dependent on each other. These are called precedence constraints. Tasks can be partial ordered, which means that some tasks must be completed before or after another task. Tasks can be coupled, which means that two or more tasks must be executed at the same time. There can also be incompatibility, in which tasks produce or obsolete other tasks.

• Some further kind of constraints which can restrict some assignments are mobility interferences. For instance, when a narrow aisle exists where only one robot can pass on the way to the task.

• A last type of constraints is resource constraints. These can prevent a task to be executed when resources are empty. The AGV then has to be charged before it can execute more tasks.

D. Optimization Objectives

In the allocation of robots to a set of tasks, there is always a certain objective to be optimized. There are several optimization objectives which can be used. At first, there are several elements which can be optimized:

- Cost: Cost that it takes for a robot to execute a task. This can be travel cost like time, distance, or fuel consumption.

- Fitness: How well an agent can perform a task.

- Reward: Gain of completing a task.

- Priority: Urgency of completing a task.

- Utility: The subtraction of cost from reward or fitness.

Knowing these elements, there are some types of objectives possible:

- Min Max: Minimize the cost of the worst agent.

- Egalitarian: Maximize the utility of the worst agent.

- Total Sum: Minimize the sum of individual costs.

- Maximize the sum of individual utilities.

- Minimize the average cost per task.

- Maximize throughput.

2.3 NAVIGATION

For a vehicle to move inside any area, it is obvious that navigation is crucial thus certainly also in AGV-systems. In contrast to the other tasks, it makes sense that this task is already decentralized. All the navigation equipment and software are on board. Each AGV needs to navigate itself inside the environment. Information about the location in the 2D-map of the environment can then be sent to a central computer or to the other devices. This will be a 3-degrees of freedom coordinate set which consists of the X- and Y -coordinates and a rotation angle. In the past decades, there is also been a lot of improvement in navigation systems,[5],[6]. There are some old school techniques which are still used. But there are some newer techniques which will gain more attention in the future. Some of the older and proven techniques are:

• Inductive navigation

• Optical navigation

• Magnetic navigation

• Inertial navigation A well-known and standard in current AGV-systems is the Laser navigation which makes use of the triangulation theory. Some new techniques which are rapidly gaining attention are:

• GPS navigation

• Natural navigation

• Vision guided navigation

The next part of the paper will go through some common types of navigation. In current AGV systems, the AGVs move on a predefined circuit of paths. In academia, sometimes an AGV can move freely into an area without being fixed to a circuit but this is almost not used in the industry. Thus, in current industrial systems, the navigation methods are used to stay on this path. The circuit can be physically present on the factory floor in the form of a tape or a wire or can be virtually present in the internal map of the AGV designed by graphical design software. It is clear that all the old navigation types are navigation methods which use a physical circuit. For future requirements like flexibility, it is not efficient to have fixed physical paths which are difficult to adapt. Virtual navigation is much more flexible as a change in the circuit can be done easily online in the graphical design software. The navigation methods can be divided into physical and virtual path navigation.

A. Physical path navigation *w*ith physical path navigation, the paths are present as physical guidelines on the floor and the navigation is easy as there is only a sensor needed to detect the guideline and make that it is followed. The predefined paths are fixed on the ground using tape or embedded into the ground using a wire. So, the AGV does not actually know its position inside the area map but just stays on the guideline. Marks along the guideline tell the AGV if it has to take a special action like increasing or decreasing speed, or to rotate at a certain degree when being in a curve.

1) Inductive navigation:

This is the first type of navigation used in the first generation of AGVs [7]. In this type of navigation, a wire is embedded into the floor. There is a current that runs through the wire which maintains a magnetic flux. The AGV has a sensor on board which consists of coils which can detect the emitted magnetic flux. The sensor detects deviation from the wire due to a decrease in magnetic flux and sends this to a controller which adapts the speeds of the wheels to go back on track. This is a very accurate method but the wired paths are fixed into the ground and cannot be adapted easily without any damage. However, this technique can still be used in very small aisles to perfectly stay on track.

2) Optical navigation:

In this type of navigation, a colour tape or a painted line with high contrast with the ground colour is placed onto the floor [8]. The AGV has an optical sensor on board which can detect the tape. The sensor detects deviation from the tape and sends this to a controller which adapts the speeds of the wheels to go back on track. As these paths are physically present onto the floor, it is not very flexible to make changes to the circuit. A tape has to be removed and replaced when the circuit has to be changed. Another disadvantage is that the tape can become dirty or can be damaged. Yet it is easier to adapt then the above wire method. This option is also cheap because of the tape and the only use of an optical sensor.

3) Magnetic tape navigation:

The physical guide path is marked with magnetic tape that is placed onto the factory floor [9]. Inside the vehicle, there is a magnetic sensor that can detect the magnetic field. The sensor detects deviation from the tape due to a decrease in the magnetic field and sends this to a controller which adapts the speeds of the wheels to go back on track. It is not very flexible to make changes to the circuit, but still more flexible than the wire option. A tape has to be removed and replaced when the circuit has to be changed. The disadvantage in comparison to a wire is that the tape can become dirty or can be damaged. The advantage of this option is that it is cheap because of the tape and the only use of a magnetic sensor.

B. Virtual path navigation with virtual path navigation, the paths are virtually present inside the local map maintained by the AGV or in the global map of the central unit. This makes it easily adaptable and expandable online. Navigation is more difficult because the AGV needs to know its exact position into the virtual map and needs to calculate deviations from the path in the software.

1) Magnetic spot navigation:

A grid or line of spots is embedded into the floor [10] on specific (*x*,*y*)-coordinates in the area map. The spots can be passive permanent magnets or transponders. Inside the vehicle, there is a magnetic sensor that can detect the spots. By detecting the spots, the AGV can determine its absolute position in the map. Together with an encoder on the wheels which calculates the travelled distance (odometry), this can be very accurate. A disadvantage is that this method is time expensive to install and to modify.

2) Laser navigation:

Laser navigation [11] is currently the most popular method for AGV navigation. A rotating laser is mounted onto the vehicle. For navigation, multiple fixed reference points like reflective strips, are located in the operating area on known coordinates. The coordinates of the reflectors are added to the global map. The emitted laser beams are reflected and scanned by the AGV after which the AGV is able to triangulate its absolute position based on the coordinates of the reflectors. At least three landmarks have to be visible to be able to navigate. This is a very accurate, secure and reliable method and is now used as a standard in a lot of AGV-systems. Disadvantages are the high price and the effort to place all the reflectors in the factory area.

3) GPS navigation:

In GPS navigation [8], satellites with known positions into the global map emit signals which are detected by the GPS-receiver. This receiver can then measure the distance to each satellite. This info is used to determine the absolute position of the receiver using trilateration. At least four satellites have to be visible to be able to navigate. For this technique, a clear line of sight to the sky is needed. This is difficult to obtain in industrial environments. As an alternative, a Local Positioning Radar (LPR) in the factory can be used instead of satellites. The disadvantage of this LPR is that there is a precision of ± 10 cm, which is not very accurate. This can be improved using sensor fusion, see Section IV-C.

4) Natural or contour navigation:

This type of navigation uses a laser to scan the whole environment around the vehicle [8]. Also, a Light Detection and Ranging (LiDAR) sensor can be used. No fixed landmarks like reflectors are needed. The AGV uses the existing environment to navigate. This makes this type of navigation very flexible. Unfortunately, the method is not that precise and robust. Using the scanned map of the environment, the vehicle is able to make a 2D map of its surroundings with all visible features like walls and pillars. When comparing this local map with the whole map of the factory, the robot knows its position into the map. This is also the way people navigate in a known environment. Before this can be done, the robot needs a map of its environment which can be scanned by a person and built up using Simultaneous Localization and Mapping (SLAM). Using SLAM, the robot has to explore the whole area while making local 2D-maps. These maps can then be combined into one large map of everything the robot has already seen. As a consequence, the system becomes a lot more flexible, especially in dynamic environments. The map can be updated every time there is a change in the environment. The disadvantages are that the material for this is expensive and that some transparent materials cannot be detected. Other sensors like sonar sensors can be used as an alternative.

5) Vision guided navigation:

Vision-guided navigation is similar to contour navigation. Instead of using a LiDAR, a stereo camera is used to make images from which 3D-pointclouds can be built. Each pixel of the camera is converted to a point in the 3D-space which is situated before the camera. This 3D-point-cloud consists of points which represent features in the area seen by the camera. This point-cloud can be projected onto a 2D-point-cloud which is a

2D projection of these features. An occupancy grid system can be used to represent the local map of the environment. An occupancy grid [12] is a cell decomposed representation of the environment. The whole area is divided into a grid of small squares. Each square is denoted either as"Unvisited","Occupied" or as"Unoccupied". Also, a probability of occupation can be used. By projecting the 2D-point-cloud of the already seen features onto the occupancy grid, the features seen by the camera are translated into occupied cells. In this way, a map of the environment can be constructed by moving around and continuously projecting the seen 2D-point-cloud onto the occupancy grid. Like in natural navigation, the robot also needs an initial map of its environment which can be scanned by a person when exploring the total area using SLAM. A disadvantage of this method is that camera images are light sensitive.

C. Sensor fusion

In practice, the above navigation methods are not implemented on their own. Because of noisy sensor data, the uncertainty on the measured position would be too large to properly navigate a vehicle. For this reason, these navigation methods are combined with filters or other types of navigation. This is called sensor fusion [13]. Sensor fusion can be divided into direct and indirect fusion. Direct fusion combines sensor data of different homogeneous or heterogeneous sensors. Indirect fusion combines sensor data with information of a prior knowledge of the environment and input.

• In direct fusion, different navigation techniques are combined. Two techniques which are not usable on their own because of high uncertainties are odometry and inertial navigation. These are frequently combined with other navigation techniques to obtain more accurate results. In odometry [8], the robot calculates its new position by knowing its starting position, the distance it already travelled, and the angle it is rotated. By using odometry sensors on the wheels, these distances and angles can be measured. However, this technique cannot be used as a stand-alone navigation technique as it is far from accurate because of slip of the wheels on different surfaces and other uncertainties. Because of this, odometry is used to combine with other navigation techniques to gain more accurate measurements. Inertial navigation [6] adds a gyroscope which detects and corrects the smallest change in the heading of the vehicle. Combination of inertial navigation and other navigation techniques will improve accuracy.

• An example of an indirect technique is a combination of a navigation technique with some form of a Kalman Filter [14],[15]. This filter uses a model of the process to make a prediction of a next state using the current state and the properties of the process. This prediction is combined with the sensor measurement of this next state to obtain a more accurate estimation. The filter considers noise on sensor measurement and on the transition from one state to the next. The Kalman filter is a very commonly used technique to improve accuracy in mobile navigation. In practice, sensor fusion is always implemented to have an accurate position estimation of the robot. Without this, the measurements would be too noisy to properly navigate a vehicle.

2.4 PATH PLANNING

A next core AGV task is path planning [16], [17]. This can be seen as the static planning task of an AGV whereas motion planning, which is going to be handled in the next section, can be seen as dynamic path planning. Static planning means that a basic collision-free path is computed using already known information. No dynamic time-dependent elements are considered, only the already known map with obstacles is used. Using this map, the robot knows its free configuration space and the obstacle space. Path planning can thus be defined as the generation of an obstacle-free path, connecting the start point with the goal point, taking into account the geometric characteristics of obstacles and the kinematic constraints of the robot. Path planning consists of two steps:

• Representation of the free configuration space.

• Using a graph search algorithm to search for the shortest path using this representation although path planning is used to generate shortest paths, it is also frequently used in AGV-systems to calculate the cost to reach a certain goal. This information is frequently used in the task allocation algorithms. In this section, several methods to represent the environment and some search algorithms to compute the shortest path using this representation will be illustrated.

A. Desired properties

The goal of path planning is to generate the shortest path which minimizes an objective function. This objective can be travel time, travel distance or fuel consumption. Algorithms thus must be able to find a solution which connects the starting point and goal by minimizing this objection function. Path planning algorithms also have to be complete. An algorithm is said to be complete if it finds a solution or correctly reports that there is no solution, and this in a finite time. Incomplete planners, on the other hand, does not always find a solution when one exists. Another important property is time complexity. Path planning will be calculated a lot of times and also re-planning of paths will occur frequently. For this reason, the time complexity has to be as small as possible.

B. Representation of the environment

A path planning algorithm has as purpose to compute the shortest path. To do this, the algorithm first needs are presentation of the possible reachable states of the AGV on the environmental map. In the current deployed AGV systems, the paths where these AGVs can move on are predetermined. Thus, a circuit which consists of nodes (intersections) and segments (paths between the intersections) is defined and designed in, for example, graphical software. This network of nodes and segments is then used by a search algorithm to search for the shortest path from point A to B. If there are no predefined paths available which the robot can follow, we need an algorithm to represent the configuration space in such a way that possible paths are generated into the full free configuration space. There are several algorithms for this. These algorithms are unfolded next.

1) Cell decomposition methods:

In cell decomposition methods [18], the total environmental map is decomposed into a grid of cells with a certain size. Each cell is either defined as occupied or non-occupied. We call this also an occupancy grid. All the cells which are marked as occupied represent obstacles like walls, tables, or other structures. In the cells which are unoccupied, the robot is able to move. The occupancy can also be probabilistic. If a robot defines a cell multiple times as occupied because of what it perceives, the probability that the cell is occupied in reality is larger. Using this technique we can get a representation of the total area, knowing in which cells we can move (free configuration space) and in which cells not (obstacle space). This cell-based representation can be translated into a connectivity graph which represents the adjacency relations between cells. A graph search algorithm like an algorithm can then be used to find an optimal path between the cell where the robot is situated and the goal cell. There are some different types of cell decomposition methods:

a) Approximate decomposition:

The size of all the cells is predefined and fixed. A disadvantage is that the complexity grows with the dimensions of the grid. And an obstacle which is smaller than the size of a cell will occupy the whole cell.

b) Adaptive cell decomposition:

A large cell size is chosen at the beginning. If a cell is partially occupied by an obstacle, the cell is divided into four equal parts. This is repeated until each cell is either fully occupied or non-occupied. This approach uses less memory but can result in difficulties in dynamic environments where a robot constantly needs to update its map when seeing other obstacles. The map is decomposed into cells which are based on the map and the locations and shapes of the obstacles. No fixed size of the cells is predefined. The cells take over the shapes of the obstacles so the union of all the free cells represents exactly the free configuration space.

2) Trajectory maps:

In this approach, the total area is covered with generated paths where the robot can move on. The generation of such trajectory maps can be done in several ways. A disadvantage is that the generation of such paths is not dynamic as the whole map has to be reconstructed if something changes in the map. This is not the case in cell-based representation where only the occupancy grid has to be updated.

a) Visibility graph:

In this approach [19], a graph of possible paths to move on is constructed by connecting all the vertices of the obstacles which are represented as polygons with straight lines. The shortest path can then be calculated by using a simple graph search algorithm. One thing to mention here is that if the robot will move on segments which are made by connecting obstacle vertices, the robot will definitely collide with those objects. This can be prevented by enlarging the obstacles with the size of the robot itself. In this way, the robot has a graph of all possible paths it can move on without the possibility to collide with an object. So if its known that the graph consists of non-colliding paths, the only task is to let a graph search algorithm find the shortest path on the graph from the start to the goal.

b) Voronoi diagram:

A Voronoi diagram [20] consists of a graph of lines which are equidistant from the two nearest obstacles. This guarantees that the paths to move on are positioned as far as possible from every obstacle. Obstacles here are also represented as polygons. Once the graph is constructed, a start and end point are added and connected to the graph and a graph search method is used to find an optimal solution. The AGV can then move on the vertices of the 2D-Voronoi diagram.

c) Probabilistic Road Maps (PRM):

A PRM [21] makes use of random samples in the configuration space of the robot. If these random samples are in the free space, then they are connected to neighbouring samples if the line between the two samples is collision-free. This can be the k-nearest neighbours or neighbours within a certain distance. This process of connecting the samples is continued until the graph is dense enough. After the graph is made, start and goal configurations are added to the graph. A graph search algorithm can now be used to connect the start and the goal with the shortest path. When the number of sampled points reaches infinity, a non-optimal path will certainly be found when there exists one.

d) Rapidly exploring random trees (RRT):

In this RRT [22] algorithm, a tree is randomly expanded from the start node. An edge is only added to the tree if it does not cause a collision. Using this technique, an area can be rapidly covered by a tree of paths where an AGV can move on. An algorithm where two trees grow towards each other can also be used. One starting from the start and one starting from the goal, this is called bi-directional search. This is used to fasten up the process. These two trees will then be linked together. If the start and goal node is connected to the tree, a graph search algorithm is used to find the shortest path. If the number of expanded nodes approach infinity, a non-optimal path will certainly be found when one exists.

3) Artificial potential fields:

Using artificial potential fields [23], a robot finds a path where it can move on by a superposition of all potential fields the robot senses. There are repulsive fields emitted by obstacles with a force inversely proportional with the distance to the obstacle. There are also attractive forces which are emitted by goal states. By using this method, it is possible that the robot gets stuck into local minima or that is can find a non-optimal path.

C. Graph search algorithms

In the previous part, alternative representations of the environment are given. These representations are further used by an algorithm to compute the shortest path. Here we want to notice again that these generated representations are only used in situations where predefined paths are missing. But in all current AGV systems, these AGVs move on predefined circuits. These circuits are used as a base to do path planning. Some of the algorithms to do this planning are unfolded next.

1) Algorithm:

The heuristic algorithm [24] is one of the most popular classical graph search algorithms in calculating the least-cost path on a weighted graph. This algorithm uses a weighted graph with nodes as locations and edges between these nodes containing the cost to go from one node to another. Also, a list of unvisited nodes and a list of visited nodes are maintained. The algorithm makes use of a heuristic to find an optimal solution much faster, this can be an estimated cost from a node to the goal node. The algorithm starts from the initial start node and works towards the goal by visiting and evaluating each neighbouring node in the unvisited list. It is an efficient and complete algorithm but has high memory usage. A guarantee to find the optimal shortest path if there is one.

2) Lite-algorithm:

Also different is the use of an extra parameter *rhs* which is introduced to give info about the cost of one-step ahead. Also, a heuristic is used here but one which estimates a cost to the start. This algorithm is an incremental heuristic search algorithm which re-uses trees from previous searches. This to speed up the search process. The D* Lite-algorithm has good results in large and complex areas. It plans shorter paths much faster than algorithms. It is less effective than A* in simple and small areas. Whereas A* cannot do re-planning, Lite can re-plan because it keeps previous information. Because of this, D* Lite algorithm is the most widely used path planning algorithm.

3) Other algorithms:

The previously mentioned algorithms are the most common in path planning for AGVs. But there are more general optimization algorithms which can be used. Some which are already mentioned in the section about task allocation like Tabu Search, Genetic Algorithm, Particle Swarm Optimization, Ant colony algorithms, and Simulated Annealing can be used. But however, the graph search algorithms like A* and D* Lite, are the most common for mobile robotics.

2.5 Vehicle Management

Vehicle management is the simplest core AGV task in that sense that it does not require complex algorithms or techniques. It monitors all the system parameters like battery status, error statuses, and mechanical parameters. All these parameters have to be monitored by a central unit or by the AGV itself while communicating its parameters to other AGVs. These parameters can cause constraints at the task allocation level. When a battery of an AGV is low or it finds itself in an error status, the tasks it was assigned to cannot be completed anymore. We can divide this management of vehicle statuses into a centralized or a decentralized structure:

- When a centralized approach is used, the central controller takes all these parameters into account. If it knows that the battery life of a vehicle is not enough to execute a certain task, then it will give it another less energy-asking task or it will send the AGV to the charging station. When the central unit receives error statuses form an AGV, then the central unit will not involve the AGV into the task allocation optimization anymore. Instead, it will send the AGV to maintenance.

- If decentralized approaches are used, the AGV monitors its own parameters. If it sees that its battery capacity is low, it will, in a market-based approach for instance, not bid on an energy expensive task but will directly head to a charging station. An AGV can also send its status parameters to other neighbouring AGVs so they can maybe take over some tasks when the initial AGV has an error status, needs battery charging, or when it needs maintenance. The AVG deals independently with its own parameters. This core task can be seen as an extra input to the other core AGV tasks. So, when more decentralized task allocation and motion planning algorithms will be used in the future, also a decentralized management of the AGV's system parameters will be used. As the distribution of intelligence and information benefits future requirements, the authors believe that this decentralization of status management will grow in the future.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology is a system of board principles or rules from which specific methods or procedure may be derived to interpret or solve. This chapter explained in details about the procedure of the implementation of the automated guided vehicle project. The methods used in this chapter are aimed to achieve the objectives of the project which will give satisfying results on the performance of the control system in the automated guided vehicle.

3.2 K-Chart Flow

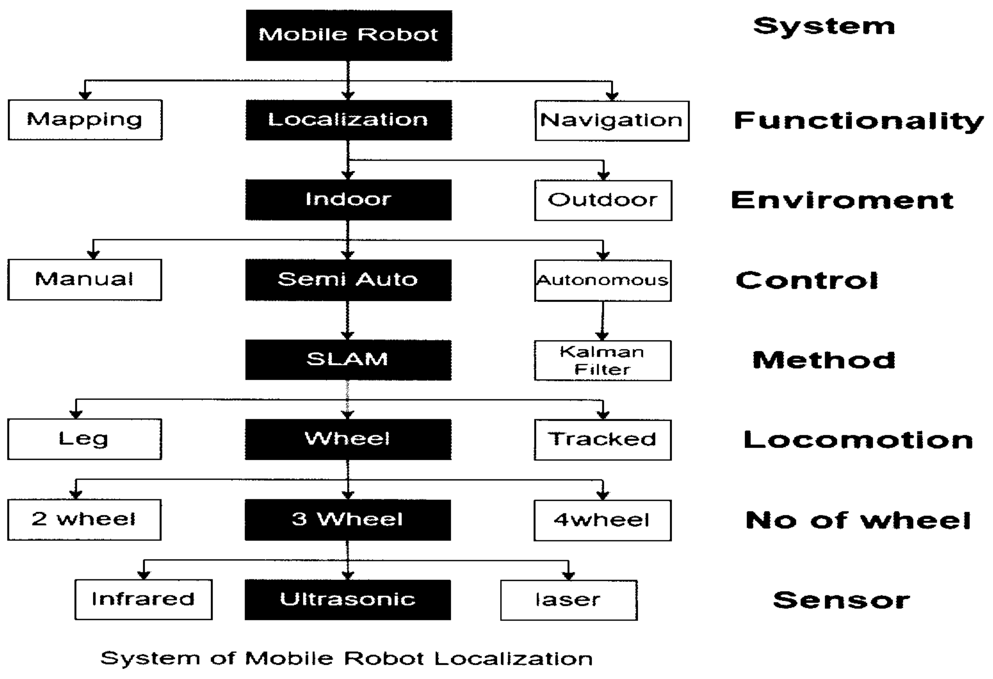


Figure 3.1 : System and stage used

Figure **3.1** shows three main stage we want to manage the project. The purple box is our stage target that we manage to finish the project. Firstly, our system of project or main system is mobile robot. Secondly, we manage finish until second stage which is localization. The concept localization is we want the mobile robot can update the scanning and mapping time to time during the real time. Next is we choose the indoor as our testing or running the robot because if we find out that mostly this robot function in indoor of industry. Fourthly, we choose the control is semi auto. Semi auto control is we trigger the one button that only give the instruction and do that continuously until we trigger another button to do another instruction. Simultaneous Localization and Mapping (SLAM) algorithm that use in Robotic Operating System (ROS) to make the localization updating by during scanning and mapping in environment that we want to mapping. Next, we use the locomotion type of wheel which is three wheel. The wheel is 2 wheel drive (2WD) while another one is use for support body of robot. Lastly, the sensor that we use is ultrasonic.

3.3 Hardware Setup

As illustrated in Figure 3.2, the experimental and suggested procedure was carried out with a 2WD mobile robot. It has a carrying capacity of around 10 kg, which includes all sensors, batteries, and mechanical components. The mobile robot features two rubber wheels, one on each side. Compared to a brushless motor The Arduino Mega ADK board was used to control this mobile robot. Each optical wheel encoder was used to measure the distance travelled. A brushless motor is one that has no brushes. The wheelbase is 285 mm long. Three ultrasonic sensors with dual outputs were used. Situated in front of the mobile robot, with a 25 ° angle between the two sensors. A 9-DOF (degrees of freedom) IMU was mounted to the mobile's centre of gravity.

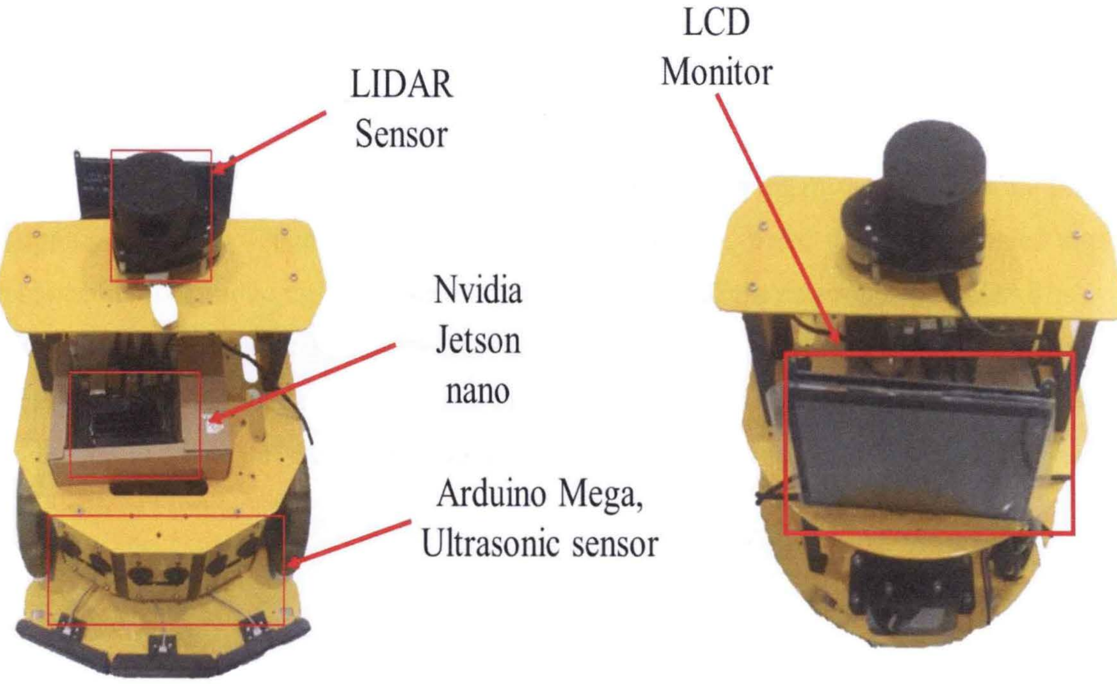


Figure 3.2: Overview of 2WD mobile robot

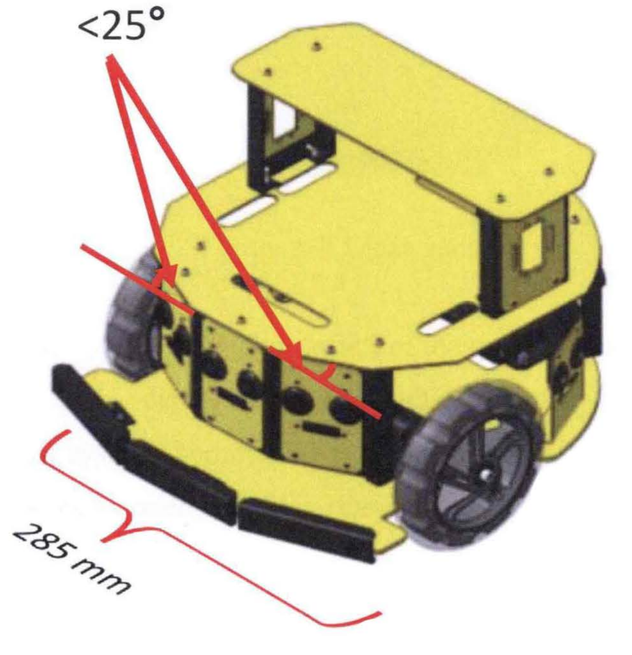


Figure 3.3: 3D modelling of 2WD mobile robot

3.3.1 Lidar Sensor

In this project LiDAR sensors are an essential component in GeoSLAM's mobile mapping solutions. Together with our SLAM algorithm, these two technologies are responsible for producing 3D pictures or "point clouds" of the environment. LiDAR is a form of remote sensing. The pulse from a laser is used in LiDAR technology to collect measurements. These are used to make 3D models of things and maps of environments. Figure 3.4 is the one of example Lidar sensor that we apply into our mode robot that has be shown in Figure 3.2.

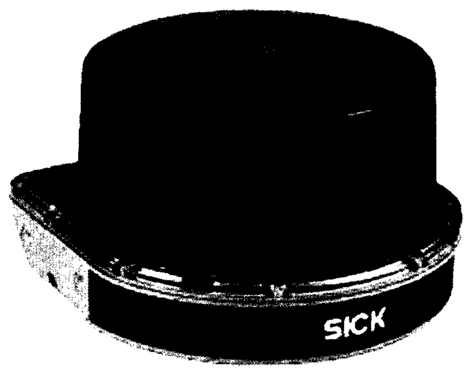


Figure 3.4: Lidar Sensor

It is usually used to examine the surface of the earth, assess information about the ground surface, create a digital twin of an object or detail a range of geospatial information. LiDAR systems harness this technology, using LiDAR on this project data to map two-dimensional models and digital elevation. From handheld to airborne LiDAR, there's a LiDAR system to capture the data we need. The SLAM algorithm, which geo references each measurement or point in the generated point cloud. Each 'point' joins together to form a three-dimensional depiction of the target object or area. The laser range for this project only we use 1 meter because based on the lidar we used in this project.

3.3.2 Jetson Nano

A microSD card serves as the boot device and main storage for the developer kit. It's critical to have a card that's both fast and large enough for your tasks; a 32GB UHS-1 card is the minimum required. Due to the limited physical memory of the Jetson Nano 2GB Developer Kit, many projects will make use of swap space on the microSD Card. As a result, 64GB or larger microSD cards are recommended. MicroSD cards with high endurance are also recommended as Figure 3.5 below.

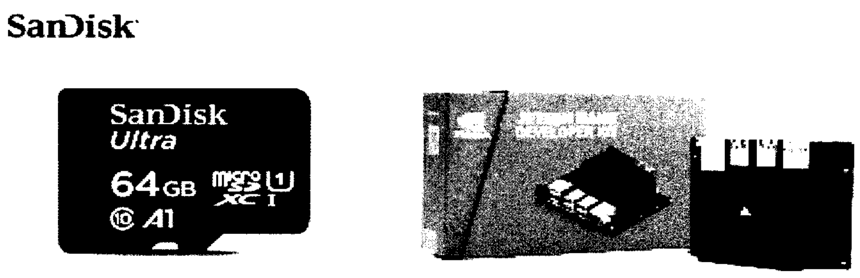


Figure 3.5 : Micro SD card and Jetson Nano

A. USB-C cable and Power Bank as Power Supply

The Jetson Nano 2GB Developer Kit can be powered by any USB-C power supply, but it does not support USB-C Power Delivery. This means that typical USB-C power supply can be utilised, but they will all default to delivering 5V3A due to the lack of power negotiation. We need a decent quality power source and cable that can deliver 5V=3A to the developer kit's USB-C connector to power the developer kit. A USB power supply's stated power output capability can be found on its label, although not every power supply claiming "5V=3A" will actually deliver this. For this we used two power bank as shown in Figure 3.6 as power supply for LCD monitor and for Jetson nano as power supply to turn on the devices.

Figure 3.6 :Power Bank and Type C USB cable

B. Write Image to the microSD Card

We use a computer with an Internet connection and the ability to read and write SD cards, either through a built-in SD card port or an adaptor, to prepare your microSD card. We saved the Jetson Nano 2GB Developer Kit SD Card Image on PC after downloading it. Follow the instructions below to write the image to your microSD card, depending on the type of computer you're using: Windows, macOS, or Linux are all viable options. Use a graphical programme like Etcher or the command line as shown in Figure 3.7 to write the SD card image.

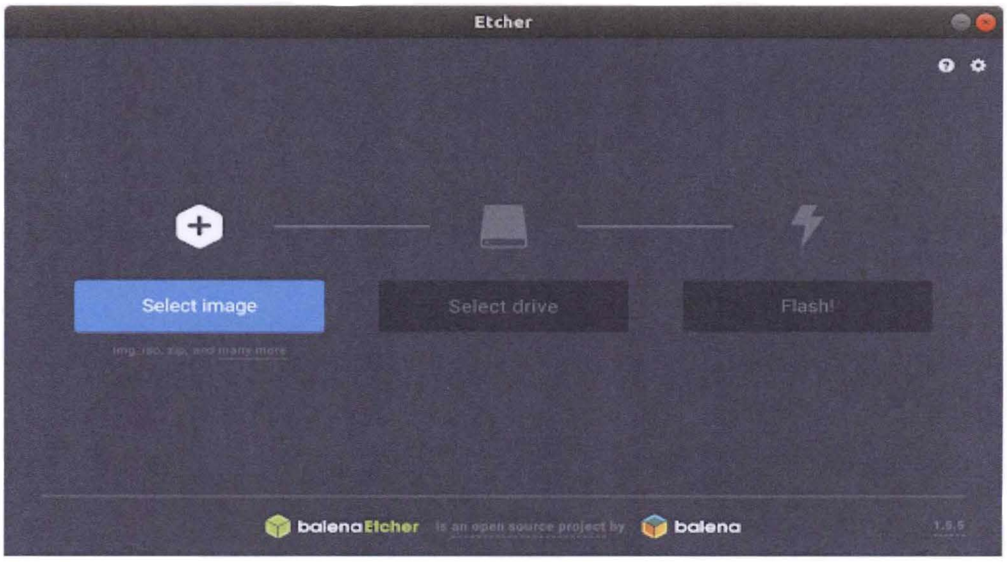

Etcher
Select image
img, iso, zip, and many more
Select drive
Flash!
balenaEtcher is an open source project by balena
1.5.5

Figure 3.7: Interface of Etcher software

"Select image" and choose the zipped image file downloaded earlier was clicked. MicroSD card has been inserted. If you have no other external drives attached, Etcher will automatically select the microSD card as target device. Otherwise, click "Change" and choose the correct device.

Lastly click ‘Flash !’ such as Figure 3.8 to allow Etcher proceed.

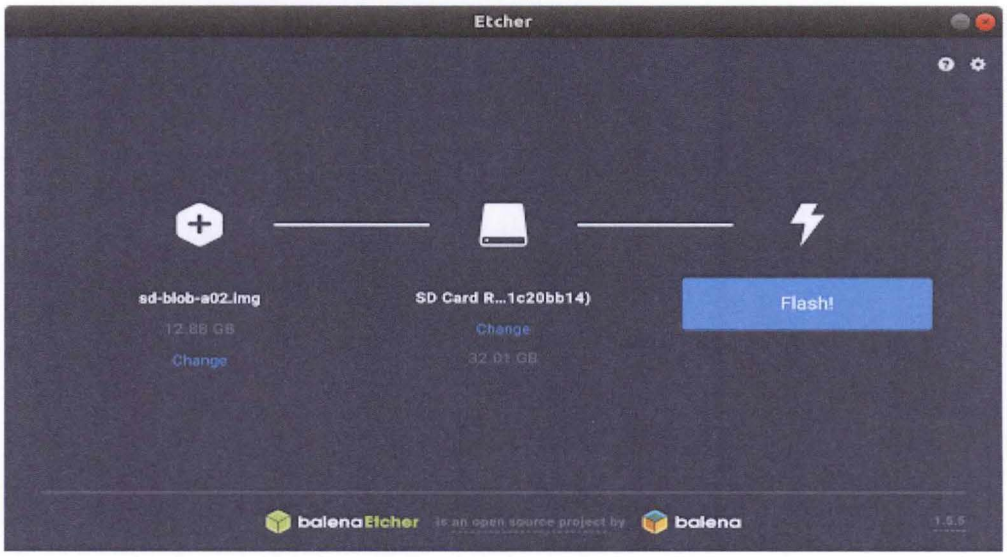


Figure 3.8: Procedure flash image into SD Card in Etcher

Micro SD was inserted as Figure 3.9 shown below.

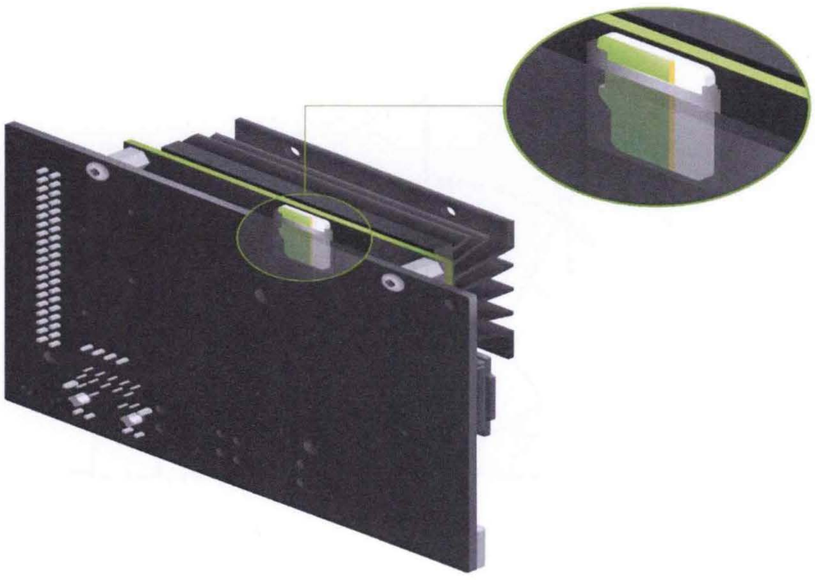


Figure 3.9: Example of Micro SD card insert into Jetson Nano

3.3.3 Dual Ultrasonic Sensor

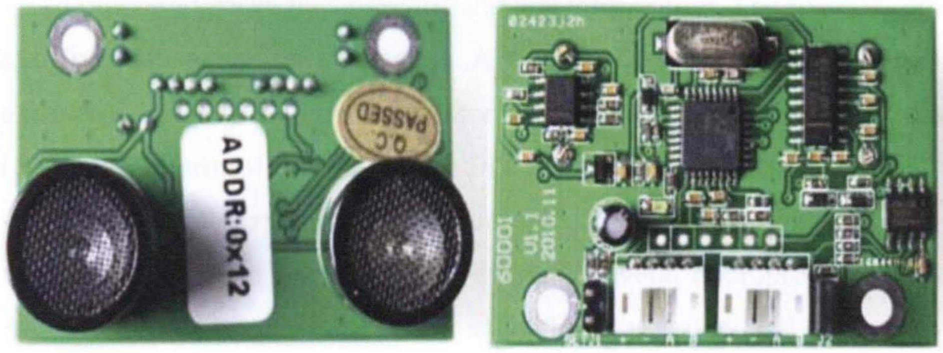


Figure 3.10 : Dual Ultrasonic

As demonstrated in Figure 3.10, a dual ultrasonic range finder sensor was used for this project. The ultrasonic sensor's core idea was a combination of transmitter and receiver, with the distance calculated by the time it took for the signal to arrive at the receiver. It is based on the RS485 interface and can connect up to 32 units in an RS485 network with various addresses. It enables the precise distance of barriers or landmarks in the ultrasonic field of view to be determined, as shown in Figure 3.11.

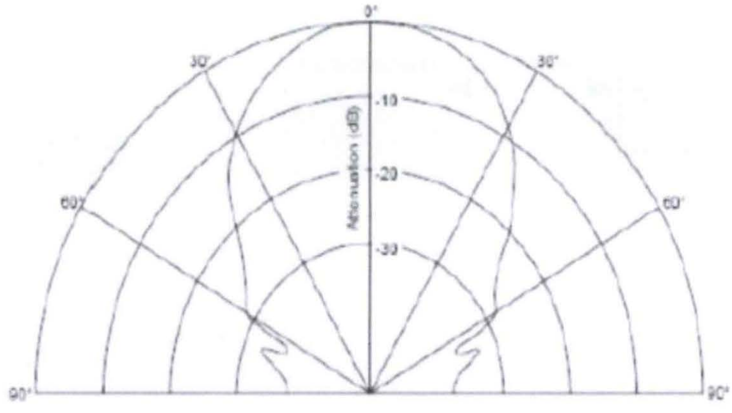


Figure 3.11 : Detection of two ultrasonic ranges

The full programming of the Arduino software for this dual ultrasonic sensor is attached at Appendix A.

3.3.4 Arduino

The rosserial package is used to send this information. The rosserial protocol is used to deliver data over a serial interface. A rosserial-server is a computer running ROS, and a rosserial-client is the microprocessor that receives sensor data and transfers it to the server in the form of ROS messages in a client-server rosserial implementation. In this version, rosserial-server is a publishing node, whereas rosserial-client is a subscriber node, albeit this can sometimes be reversed. Several microprocessor types, including Arduino, STM32, embedded linux, and others, are supported by the Rosserial-client package. The rosserial-server package comes in two flavours: Python and C++.This Figure 3.12 shown the related between Arduino and ROS.

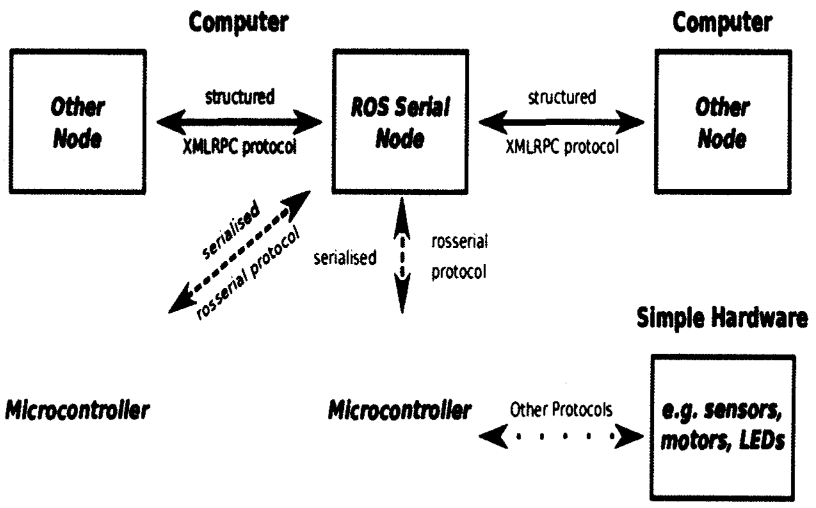


Figure 3.12 : Structure of Arduino and ROS

3.3.5 LCD display

For this project, LCD was included in our model robot mainly for display the output of the mapping by using RViz Software that has been prepared by application open source which is ROS. Other than that, we can run the program by monitor the terminal that has been use to program the ROS in this display. Because this project had the movement of robot. So, to ease the project running we want this display attach to the robot shown as Figure 3.13.So, we can monitor this robot easily.

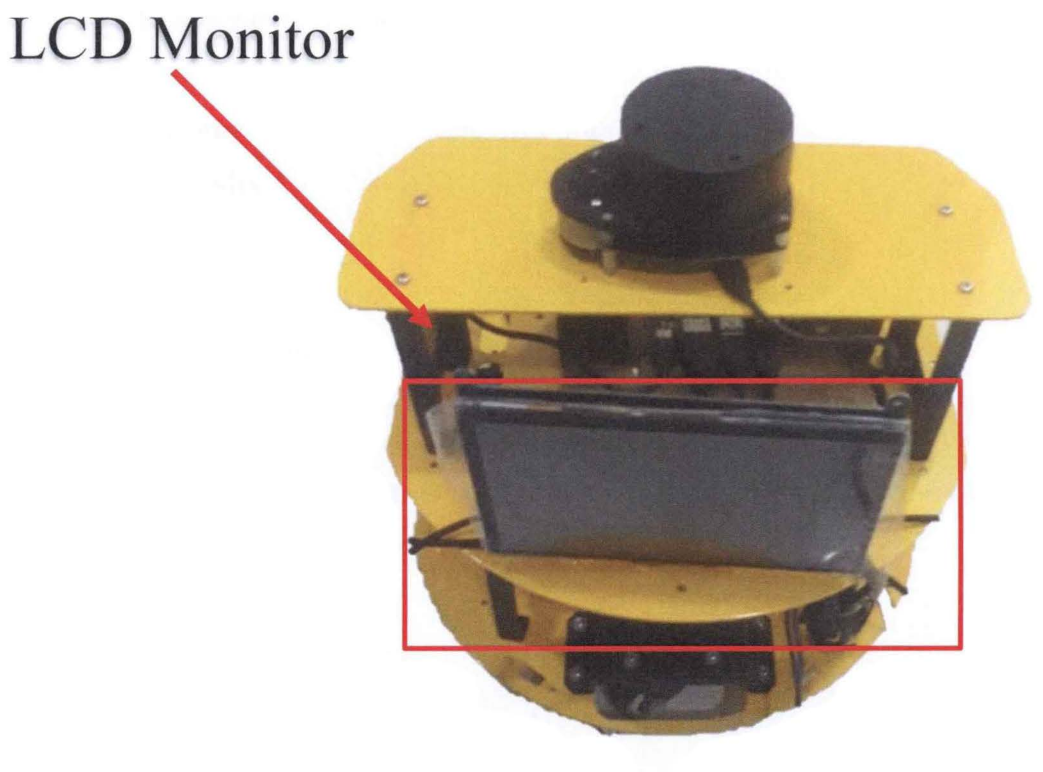


Figure 3.13 : View of LCD attach to the robot

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter to discuss the main hardware setup and explain the movement that has been setup to this prototype (Figure 3.2). Firstly, part we will review the the hardware setup in the prototype in some of view of the prototype. Then, discuss more deeply about the program that has been write into Arduino. The program code has been shown in Appendix A.

4.2 View of Complete Prototype

The Figure 4.1 shown below is the front view of the prototype that we can see certain component such as Lidar sensor, Jetson nano and ultrasonic sensor. This design was fix to run this project function and move efficiency.

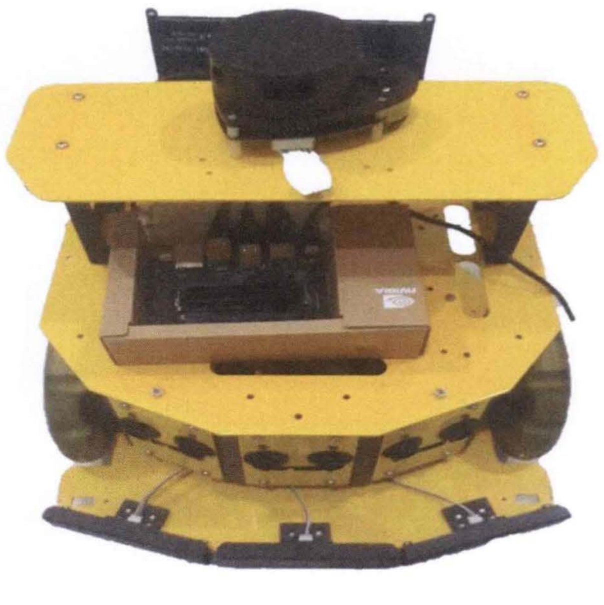


Figure 4.1 : Front View of prototype

For back view of prototype as shown in Figure 4.2 we only put the power bank as the power supply for the LCD monitor for monitor the program in LCD monitor and the same time view of mapping run by ROS software which is RViz to show the result for mapping surrounding of enviroment.

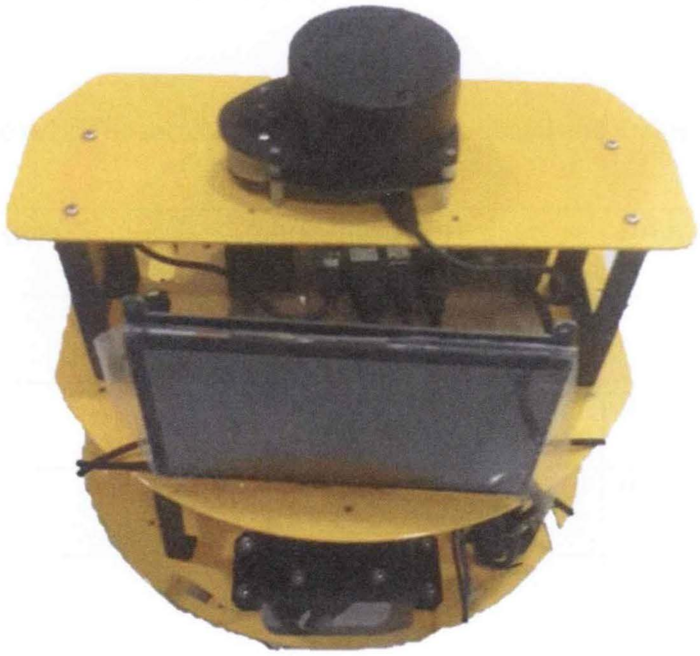


Figure 4.2 : Back view of Prototype

4.3 Dual Ultrasonic Error Variation

The goal of this experiment is to determine the dual ultrasonic sensor's inaccuracy and accuracy. This dual ultrasonic sensor has a range of around 40 to 3000 mm and a resolution of 10 mm. In this experiment, the errors of all three dual ultrasonic sensors were measured, and the variance of error was calculated using the errors of these three sensors. Table 4.1 shows the discrepancy between the real distance and an ultrasonic reading.

Table 4.1 : The measurement of ultrasonic actual and real reading

No.	Actual (mm)	Ultrasonic, (mm)	Error, (mm)
1	0	-	-
2	100	100	0
3	200	200	0
4	300	310	10
5	400	400	0
6	500	500	0
7	600	610	10
8	700	700	0
9	800	800	0
10	900	910	10
11	1000	1010	10
12	1100	1100	0

13	1200	1210	10
14	1300	1300	0
15	1400	1400	0
16	1500	1500	0
17	1600	1610	10
18	1700	1690	-10
19	1800	1790	-10
20	1900	1900	0
21	2000	2000	0
22	2100	2100	0
23	2200	2220	20
24	2300	2320	20
25	2400	2410	10
26	2500	2510	10
27	2600	2630	30
28	2700	2670	-30
29	2800	2820	20
30	2900	2910	10
31	3000	3000	0

4.4 Discussion

For the Arduino coding the information is sent using the rosserial package. Data is delivered over a serial interface using the rosserial protocol. In a client-server rosserial implementation, a rosserial-server is a computer running ROS, and a rosserial-client is the microcontroller that receives sensor data and sends it to the server in the form of ROS messages. Rosserial-server is a publishing node in this version, while rosserial-client is a subscriber node, though this can be flipped on occasion.

The Rosserial-client package supports a variety of microprocessor types, including Arduino, STM32, embedded Linux, and others. There are two versions of the rosserial-server package: Python and C++.

For the ultrasonic result that shown in table tell us the reading that can be show that actual measurement in RViz application of ROS.So the RViz can detect the surrounding distance by reading from Ultrasonic sensor that has been attached to the prototype (Figure 3.2). That has been small error for ultrasonic in actual and the real reading because error of environment and accuracy of the measurement in real reading.

Lastly for body part or we known as prototype was be attach with this part which is Lidar Sensor, LCD display, Jetson nano and ultrasonic sensor. This part rolling the function respectively and work the function to combine into jetson nano as an operation the whole system for this project.

CHAPTER 5

CONCLUSION

5.1 Conclusion

The project's result is that we were able to develop an AGV model using Arduino and fitted with onboard sensors and actuators such as ultrasonic bumpers as our application technology to create an autonomous mobile robot for localization. The designed localization system combines two different types of sensors wheel encoders and ultrasonic sensors. Only owing to uncertainty measurement from wheel slippage, wheel misalignment, and tiny odometer resolution is it impossible to determine exact location of mobile robot using wheel encoder.

Lastly, the system noise covariance and the measurement noise covariance are used to calculate the variance of error for odometry and ultrasonic, respectively. This thesis about mobile robot localization and the measurement of ultrasonic has been discussed.

5.2 Recommendation

This project can be used as a foundation for developing a totally autonomous mobile robot in the future, with improvements to the mapping approach, path planning, and control system. Mobile robot localisation using static settings, where it knows the location with some ambiguity, has been discussed in this research. If the mobile robot can self-localize in a dynamic environment, it becomes more challenging and exciting. Other than that, this project can move forward with navigation system by using GPS and camera to visualize environment and the movement more like autonomous like AGV mobile robot like in other big industry.

REFERENCES

[1] X. Jia and M. Q. Meng, "A survey and analysis of task allocation algorithms in multi-robot systems", IEEE International Conference on Robotics and Biomimetics, ROBIO 2013, pp. 2280–2285, 2013

[2] S. Nesmachnow, "An overview of metaheuristics: accurate and efficient methods for optimisation", International Journal of Metaheuristics, vol. 3, no. 4, pp. 320–347, 2014.

[3] A. R. Mosteo and L. Montano, "A survey of multirobot task allocation", 2010, pp. 1–27.

[4] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems",International Journal of Robotics Research, vol. 23, no. 9, pp. 939–954, 2004

[5] G. J. Cawood and I. A. Gorlach, "Navigation and locomotion of a low-cost Automated Guided Cart", Proceedings of the 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2015, pp. 83–88, 2015.

[6] H. M. Barbera, J. P. C. Quinonero, M. A. Z. Izquierdo, and A. G. Skarmeta, "i-Fork: a flexible AGV system using topological and grid maps", in IEEE International Conference on Robotics & Automation, IEEE, 2003, pp. 2147–2152.

[7] J. Song, "Electromagnetic Induction Sensor of Navigation System for Spraying Robot *", Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011), pp. 175–181, 2011.

[8] C. Feledy and M. S. Luttenberger, "A State of the Art Map of the AGVS Technology and a Guideline for How and Where to Use It", University of Lund, Lund, Tech. Rep., 2017.

[9] S. Kamewaka and S. Uemura, "A magnetic guidance method for automated guided vehicles", IEEE Transactions on Magnetics, vol. 23, no. 5, pp. 2416–2418,1987.

[10] S. Y. Lee and H. W. Yang, "Navigation of automated guided vehicles using magnet spot guidance method", Robotics and Computer-Integrated Manufacturing, vol. 28, no. 3, pp. 425–436, 2012.

[11] U. Andersson, "Laser Navigation System for Automatic Guided Vehicles", Lulea, Tech. Rep., 2013.

[12] A. M. Santana, K. R. Aires, R. M. Veras, and A. A. Medeiros, "An approach for 2D visual occupancy grid map using monocular vision", Electronic Notes in Theoretical Computer Science, vol. 281, pp. 175–191, 2011.

[13] W. Elmenreich, "An introduction to sensor fusion", *Vienna University of Technology, Austria*, pp. 1–28,2002.

[14] G. Welch and G. Bishop, "An Introduction to the Kalman Filter", in Siggraph Course. 8, 2006, pp. 1-16.

[15] S. W. Yoon, S. B. Park, and J. S. Kim, "Kalman filter sensor fusion for Mecanum wheeled automated guided vehicle localization", Journal of Sensors, vol. 2015, pp. 1–8, 2015.

[16] V. Kunchev, L.Jain, V. Ivancevic, and A. Finn, "Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review", in KES2006 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, 2006, pp. 537–544.

[17] S. G. Anavatti, S. L. Francis, and M. Garratt, "Pathplanning modules for Autonomous Vehicles: Current status and challenges", ICAMIMIA 2015 - International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation, Proceeding - In conjunction with Industrial Mechatronics and Automation Exhibition, IMAE, pp. 205–214, 2016.

[18] Ahmad Abbadi and Vaclav Prenosil, "Safe Path Planning Using Cell Decomposition Approximation",International Conference Distance Learning, Simulation and Communication, no. May, 2015.

[19] J. O'Rourke and I. Streinu, "The vertex-edge visibility graph of a polygon", Computational Geometry: Theory and Applications, vol. 10, no. 2, pp. 105–120, 1998.

[20] Adam Dobrin, "A Review of Properties and Variations of Voronoi Diagrams", World Applied Sciences Journal, vol. 21, no. 1, pp. 21–29, 2013.

[21] L. E. Kavraki, P. Svestka, J.-C. Latombe, and Mark H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[22] S. M. LaValle and J. J. Kuffner, "Rapidly-Exploring Random Trees: Progress and prospects", Algorithmic and computational robotics: New directions, 2000.

[23] C. Warren, Global path planning using artificial potential fields, 1989.

[24] D. H. Kim, N. T. Hai, and W. Y. Joe, "A Guide to Select Path Planning Algorithm for Automated Guided Vehicle (AGV)", Lecture Notes in Electrical Engineering, vol. 465, pp. 587–596, 2018.

[25] K. M.R. L. Moorthy and W. H. Guan, "Deadlock Prediction and Avoidance in an AGV System", Tech.Rep., 2000

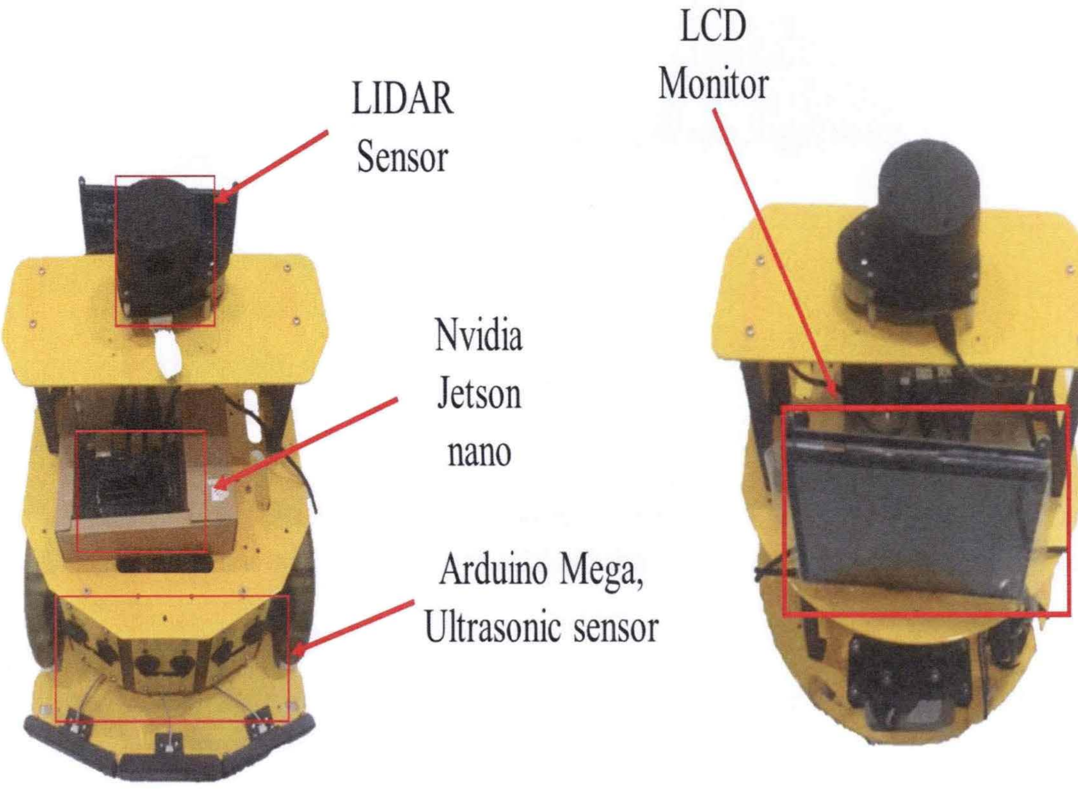
APPENDICES

Appendix A: Coding of Arduino for movement

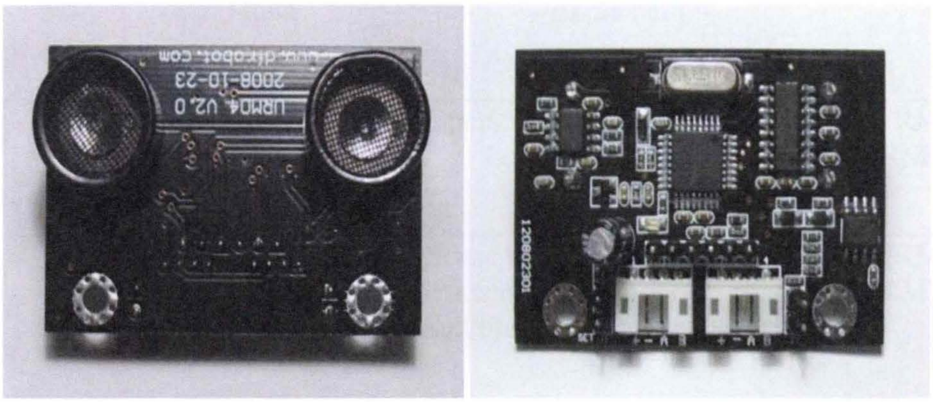
```
#include <ros.h>
#include <geometry_msgs/Twist.h>
|
#define IN1 6
#define IN2 4
#define IN3 7
#define IN4 5

void onTwist(const geometry_msgs::Twist& msg)
{
  //forward
  if(msg.linear.x < 0)
  {
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    analogWrite(IN3,-90);
    analogWrite(IN4,-90);

  }
  //backward
  else if(msg.linear.x > 0)
  {
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    analogWrite(IN3,90);
    analogWrite(IN4,90);


 //right
 else if(msg.angular.z < 0)
 {
   digitalWrite(IN1,HIGH);
   digitalWrite(IN2,HIGH);
   analogWrite(IN3,-90);
   analogWrite(IN4,90);
 }
```

```
  else if(msg.angular.z > 0)
  {
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    analogWrite(IN3,90);
    analogWrite(IN4,-90);

  }
  else
  {
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    analogWrite(IN3,0);
    analogWrite(IN4,0);

  }
}



ros::Subscriber<geometry_msgs::Twist> sub("cmd_vel",onTwist);

ros::NodeHandle nh;

void setup() {
  // put your setup code here, to run once:
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);

  nh.initNode();
  nh.subscribe(sub);
}

void loop() {

  nh.spinOnce();
}
```

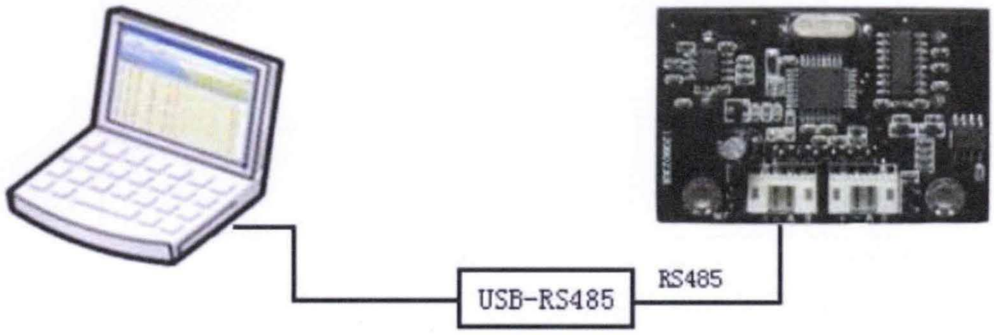
Appendix B: Prototype



Appendix C: Ultrasonic RS485



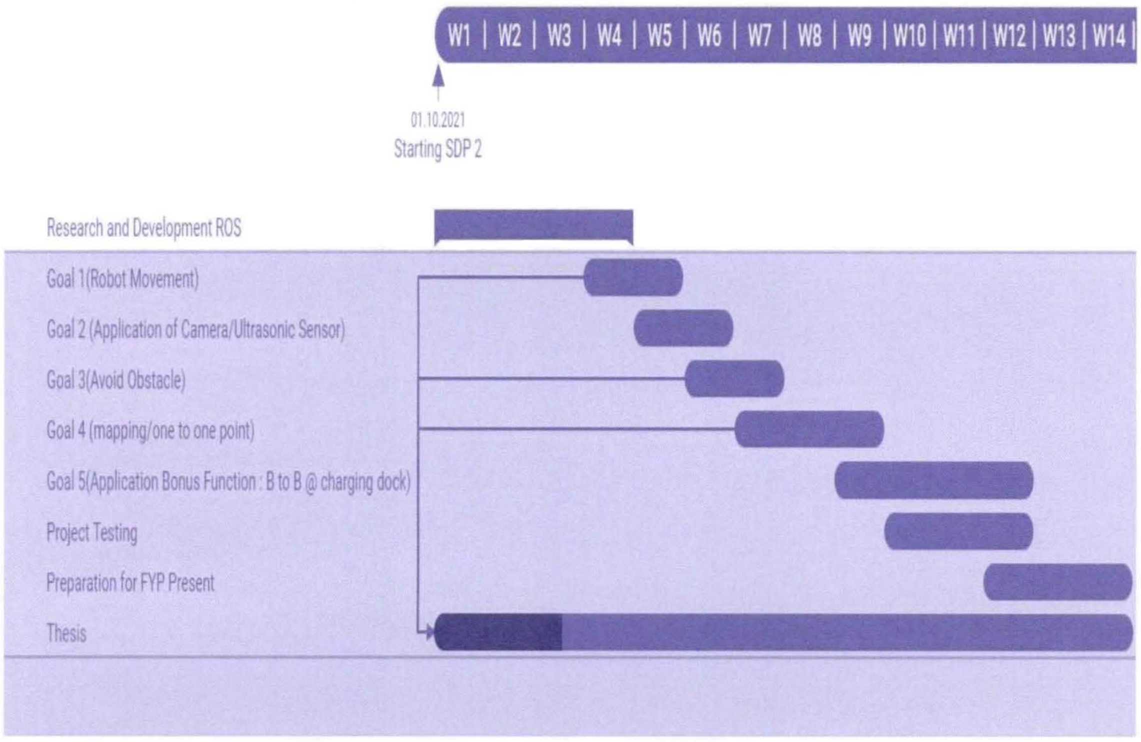
- **RS485 Interface：Two connectors**
  - +: +5V DC Power +5V
  - -: GND Ground
  - **A**: A RS485 A(+)
  - **B**: B RS485 B(-)
- **ISP Pin**: For factory firmware uploading
- **Communication LED**: As the device is powered up, this LED will flash four times which indicates that the sensor is working properly. This LED will also flash when it is communcating with other devices.
  - **Jumper A**: Not in use
  - **Jumper B**: When the sensor is working under a network, only the Jumper B for the first Device and the last Device need to be bridged.



Appendix D: List of Budget

No	Item	Explaination	Cost
1	Jetson nano	Operation System Linux	1xRM647.80 =RM647.80
2	Ultrasonic Sensor	Sense the obstacle and measure distance between model robot and obstacle	3xRM9.90 =RM29.70
3	Protoype model robot	To move and carry all the sensor that has install on it.	1xRM350 .00 =RM350.00
4	Power Bank 5V x 3A	Supply the power to jetson nano and LCD	2xRM35.00 =RM70.00
5	LCD 7 inch	Dislpay the output from jetson nano	1xRM328.00 =RM328.00
6	SD card 64GB & Adapter	Flash the image for jetson nano to run Linux Operation System	1xRM69.00 =RM69.00
7	Lidar Sensor	Mapping and scanning	1xRM403.76 = RM403.76
8	Battery 12V 7.2Ah	Supply the power for wheel encoder	1xRM48.50 = RM48.50
9	Arduino Mega ADK	Microcontroller for wheel encoder	1xRM138.00 = RM138.00
		Total	RM2084.76

Appendix E: Gantt chart progress planning


W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 | W13 | W14 |
01.10.2021
Starting SDP 2
Research and Development ROS
Goal 1(Robot Movement)
Goal 2 (Application of Camera/Ultrasonic Sensor)
Goal 3(Avoid Obstacle)
Goal 4 (mapping/one to one point)
Goal 5(Application Bonus Function : B to B @ charging dock)
Project Testing
Preparation for FYP Present
Thesis