**HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**BACHELOR THESIS PROJECT**

**Serving Mobile Robot For Restaurant**

**Firmware Design**

**LY DUC TRUNG**

trung.ld181930@sis.hust.edu.vn

**Ngành Kỹ thuật điện**

**Chuyên ngành Hệ thống điện**

|  |  |
| --- | --- |
| **Supervisor:** | Le Minh Thuy, Ph. D  Chữ ký của GVHD |
| **Department:** | Instrumentation and industrial informatics |
| **School:** | School of Electrical & Electronics Engineering |
| **HA NOI, 08/2022** | |

**ĐỀ TÀI TỐT NGHIỆP**

Biểu mẫu của Đề tài/khóa luận tốt nghiệp theo qui định của viện, tuy nhiên cần đảm bảo giáo viên giao đề tài ký và ghi rõ họ và tên.

Trường hợp có 2 giáo viên hướng dẫn thì sẽ cùng ký tên.

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

**Lời cảm ơn**

Đây là mục tùy chọn, nên viết phần cảm ơn ngắn gọn, tránh dùng các từ sáo rỗng, giới hạn trong khoảng 100-150 từ.

**Tóm tắt nội dung đồ án**

Tóm tắt nội dung của đồ án tốt nghiệp trong khoảng tối đa 300 chữ. Phần tóm tắt cần nêu được các ý: vấn đề cần thực hiện; phương pháp thực hiện; công cụ sử dụng (phần mềm, phần cứng…); kết quả của đồ án có phù hợp với các vấn đề đã đặt ra hay không; tính thực tế của đồ án, định hướng phát triển mở rộng của đồ án (nếu có); các kiến thức và kỹ năng mà sinh viên đã đạt được.

Sinh viên thực hiện

Ký và ghi rõ họ tên

**MỤC LỤC**

[CHAPTER 1. GENERAL INTRODUCTION ABOUT MOBILE ROBOT 1](#_Toc109256327)

[1.1 Overall information about mobile robots 1](#_Toc109256328)

[1.1.1 Mobile Robot Introduction 1](#_Toc109256329)

[1.1.2 Mobile robot in service delivery 2](#_Toc109256330)

[1.2 Serving mobile robot design planning 6](#_Toc109256331)

[1.2.1 Design planning of project 6](#_Toc109256332)

[1.2.2 Scopes of work 7](#_Toc109256333)

[CHAPTER 2. FIRMWARE DESIGN FOR SERVING MOBILE ROBOT 8](#_Toc109256334)

[2.1 Robot software platforms 8](#_Toc109256335)

[2.2 The basis for selecting ROS 2 (second version of ROS) 9](#_Toc109256336)

[2.3 ROS 2 Introduction 9](#_Toc109256337)

[2.3.1 Communication 10](#_Toc109256338)

[2.4 Serving mobile robot firmware architecture 10](#_Toc109256339)

[2.5 Raspberry Pi 4B 10](#_Toc109256340)

[2.5.1 The change of the ROS version 10](#_Toc109256341)

[2.5.2 Ubuntu version 13](#_Toc109256342)

[2.5.3 zRAM 14](#_Toc109256343)

[2.5.4 ROS 2 Galactic 15](#_Toc109256344)

[2.6 ESP32 Microcontroller 15](#_Toc109256345)

[2.6.1 15](#_Toc109256346)

[2.7 Motor driver block 15](#_Toc109256347)

[2.7.1 EPS32 firmware 15](#_Toc109256348)

[2.7.2 Tạo tiêu đề các cấp 20](#_Toc109256349)

[2.7.3 Định dạng phần nội dung các chương, mục 20](#_Toc109256350)

[2.7.4 Hình vẽ - Đồ thị 20](#_Toc109256351)

[2.7.5 Bảng biểu 22](#_Toc109256352)

[2.7.6 Phương trình 24](#_Toc109256353)

[2.8 Tạo tham chiếu chéo giữa các đoạn văn bản 27](#_Toc109256354)

[2.9 Tạo danh mục tài liệu tham khảo 27](#_Toc109256355)

[2.10 Cập nhật lại các chú thích và tham chiếu 31](#_Toc109256356)

[2.11 Tạo danh mục hình vẽ 31](#_Toc109256357)

[2.12 Tạo danh mục bảng biểu 32](#_Toc109256358)

[2.13 Tạo trang mục lục 32](#_Toc109256359)

[2.14 Qui cách đóng quyển 33](#_Toc109256360)

[CHAPTER 3. SỬ DỤNG CÁC BIỂU ĐỒ 35](#_Toc109256361)

[3.1 Giới thiệu về biểu diễn bằng đồ thị 35](#_Toc109256362)

[3.2 Đồ thị kiểu bánh 35](#_Toc109256363)

[3.3 Đồ thị kiểu thanh ngang 36](#_Toc109256364)

[3.4 Đồ thị kiểu cột đứng 37](#_Toc109256365)

[3.5 Đồ thị kiểu đường 37](#_Toc109256366)

[3.6 Đồ thị kiểu diện tích 37](#_Toc109256367)

[CHAPTER 4. KẾT LUẬN 39](#_Toc109256368)

[4.1 Kết luận 39](#_Toc109256369)

[4.2 Hướng phát triển của đồ án trong tương lai 39](#_Toc109256370)

[References 40](#_Toc109256371)

[PHỤ LỤC 41](#_Toc109256372)

**DANH MỤC HÌNH VẼ**

[Figure 1: Scheduling latency in an idle environment. (a) ROS 1.0 (b) ROS 2.0 [3] 11](#_Toc109251523)

[Figure 2: Scheduling latency in a stressed environment (a) ROS 1.0 (b) ROS 2.0 [3] 12](#_Toc109251524)

[Figure 3: Maximum communication latency given the data size in environments with network traffic 13](#_Toc109251525)

[Figure 4: System requirements from official website for ROS 2 Galactic Geochelone 13](#_Toc109251526)

[Figure 5: Ubuntu 20.04.4 LTS on Raspberry Pi 4 14](#_Toc109251527)

[Figure 6: Initial amount of RAM on Rapsberry Pi 4 14](#_Toc109251528)

[Figure 7: No-load performance of RAM on Raspberry Pi 4 14](#_Toc109251529)

[Figure 8: Motor driver block 15](#_Toc109251530)

[Figure 9: Flowchart of setup() function for EPS32 firmware 16](#_Toc109251531)

[Figure 10: Flowchart of loop() function for EPS32 firmware 17](#_Toc109251532)

[Figure 11: JSON form of receiving data of ESP32 microcontroller 17](#_Toc109251533)

[Figure 12: Commonly­used Frequencies and Resolutions 18](#_Toc109251534)

[Figure 13: Flowchart of drive\_motors() function for EPS32 firmware 18](#_Toc109251535)

[Figure 14: Transmission speed and Real transmission speed of serial port 19](#_Toc109251536)

**DANH MỤC HÌNH VẼ**

[Bảng 1.1 Thống kê các thiết bị và giá thành 12](#_Toc20580109)

# GENERAL INTRODUCTION ABOUT MOBILE ROBOT

## Overall information about mobile robots

### Mobile Robot Introduction

Throughout history, the idea of mechanical automata that can resemble or replace human activities has originated in the mythologies of many cultures around the world. However, this becomes more popular in the 19th and especially 20th century. A machine with the ability to carry out a complex series of actions automatically is called robot. Mobile robot is a robot that is not attached to the environment and is able to move in a certain surrounding space.

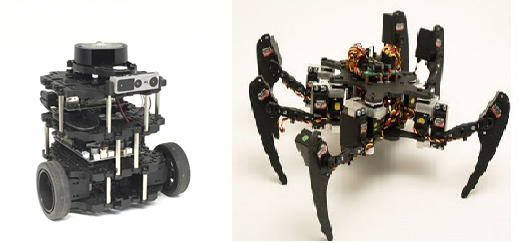
A mobile robot can be designed with an autonomous system, which can self-navigate within their environment or relies on guidance devices to follow a pre-defined route. The autonomy of the mobile robot must be guaranteed with aspects of energy and self-making decision. The level of autonomy can be defined based on the commands of human operator, from operate with desired wheel velocities to achieve desired robot longitudinal and angular velocities, follow path or trajectory to self-operate inside of the known or unknown environment.

The hardware structure of mobile robot is a combination of mechanical and electronic parts, mainly:

* Mechanical parts: rigid and moving parts (body, wheels, etc.)
* Actuators: electrical motors (DC, stepper, servomotor, etc.)
* Sensors: rotation encoders, lidar, ultrasonic sensors, accelerometer & gyroscope sensor, etc.
* Computers:  micro-controllers, embedded computer, …
* Power unit: batteries, solar panels, …

Related to the working environment, mobile robot can be classified into three principal groups:

* Ground mobile systems: They are designed to operate on the ground and included various platform such as mobile vehicles with wheels or caterpillar, legged robots or mimic some other type of animal locomotion.



* Aerial mobile systems: This group consists of systems that can fly in a certain aerial space or orbit the Earth.



* Water and under water mobile system: We can find many kinds of ships, boats, submarines, autonomous underwater vehicles, etc.



Nowadays, with the rapid development of technology, mobile robot is gradually applied to replace human activities in various fields, especially in hazardous or inaccessible environments (planets, minefields, radioactive environments, etc.). Moreover, applying robot in production can also help human to reduce labor costs, and increase productivity. The application field of mobile robot is expanded day by day namely medical service, operation support, cleaning applications, consumer good stores, military and security, etc. It can be concluded that in the near future, mobile robots can become a part of our life with a great support from household chores to discover unknown planets.

### Mobile robot in service delivery

Different from industrial application where the role of robot is to replace humans for carrying out dangerous tasks in hazardous environment, robots for service are more likely to assist humans for working. We can easily find the support of mobile robot in many aspects such as hospitality, agriculture or education. Another application of mobile robot must be mentioned is to deliver goods. The general function of robot in this application is to carry a payload from a source to its destination. With the support of mobile robot, human can focus on higher value activities. The problem of labor shortage for low-value tasks can also be solved and productivity is also increased because robots don’t need breaks or downtime.

One significant example of using robot in delivery is for restaurants where they can deliver food to the table and carry dirty dishes back to the kitchen. Many technology companies are racing to implement their own delivery robots to meet the zooming upward demand day by day. Below are some design models of serving robot in the market.

“Servi”: “Servi” is a production of Bear Robotics, a robotics and artificial intelligence company, and SoftBank Robotics Group.



“Servi” can autonomously travel and carry food when customer select a table to serve on the screen. It can also move while detecting and avoiding obstacles such as people and objects with high accuracy. “Servi” is said to improve the efficiency of the establishment and the quality of customer service. 

|  |  |
| --- | --- |
| ***Main specs*** | ***Value*** |
| Maximum Load Weight | 35kg |
| Navigation Method | SLAM |
| Sensor | LiDAR, 3D Camera |
| Communication | Wi-Fi 2.4/5 GHz |
| Working time | 8 – 12 hours/single charge |
| Maximum speed | 0.6 m/s |
| Rental fee | 738 USD/month |

Beside “Servi” robot, many other robotics company also introduced their invention of serving robot.





These robots’ design shares the same technologies which will be described in detail below:

* Sensors: To localize and navigate, several range sensors, cameras, and radars can be added. One of the most popular sensors that are used in mobile robot is a LIDAR (Light Detection And Ranging), which is normally set up on the top of the robot to achieve maximum coverage. The point cloud produced by the LIDAR sensor contains a huge amount of data about the surroundings. To distinguish, categorize, and identify the objects from the raw point cloud data, advanced data processing is used. Additional sensors like radar or ultrasonic sensors are also added to improve the preciseness about the environment. They are implemented to measure the distance and detect near obstacles around the robot. Besides that, to enable the calculation for odometry, rotation encoders are equipped with robot’s motors. Some other sensors like accelerometers, gyroscopes are also attached on the robot to measure navigation tasks.
* Localization and mapping: Many features can be used to solve these tasks. The recent modern solution becoming more and more popular is using Simultaneous Localization And Mapping method (SLAM) – the method of constructing or updating a map of an uncharted area while keep tracking robot’s position.
* Communication: Wireless technology is used by many mobile robot platforms to connect with other robots, human machine interfaces, and off-line computing resources. Numerous mobile robots are outfitted with wireless technologies including Bluetooth, Wi-Fi, Wireless LAN, and others.
  + Wi-Fi or Wireless LAN: Wireless Local Area Networks, often known as Wi-Fi or WLAN, are based on a set of specifications known as 802.11 from the Institute of Electrical and Electronics Engineers (IEEE). Wi-Fi operates mostly in the unlicensed 2.4GHz spectrum of radio frequency. It makes it possible for someone to use a wireless access point to connect to the Internet using a computer or PDA that supports wireless. A hot spot is the geographic area that one or more access points cover. Wi-Fi is increasingly often utilized for Internet access outside despite being designed for mobile devices and local area networks. ControlNet: ControlNet employs a single media link with (cheap) RG-6 coaxial cables and a bus. It is built on its own physical and data link layer. It has up to 99 nodes, a 5 Mbit/s throughput, data upload/download, and P2P communication.
  + Bluetooth: Most wireless technologies, including Bluetooth and the IrDA standard, provide users the option of bolstering their local wireless network. In the workplace, in labs, or at home, Bluetooth technology, developed by Ericsson in 1994, is utilized in place of wires. a wireless technology that uses a short radio frequency cable. (2012) IACSIT Press, Singapore, IPCSIT vol. 34, International Conference on System Engineering and Modeling (ICSEM 2012) The scientific community is moving distance to 2.4GHz bandwidth to replace unlicensed equipment. Broadband 1 MB per second speech and data exchanges are supported via Bluetooth devices, which typically have a range of around 10 meters. Parts of Bluetooth have been included into a variety of mobile devices, including mobile phones, PDAs, and other wireless equipment, because of Bluetooth's benefits, including inexpensive prices, low power consumption, and nature may be turned in different directions. Around 200 million Bluetooth devices will be sold in 2001, according to research by In-Stat/MDR and Frost & Sullivan, and one billion in 2006. As a result, Bluetooth technology is now being used for mobile robot controllers.

## Serving mobile robot design planning

### Design planning of project

The market for autonomous delivery robots is still at the beginning. One drawback limiting applications of service mobile robot is the high cost for renting or buying. Therefore, in this project, our purpose is to design a mobile robot for restaurants with the ability to self-navigate and can totally replace the role of servant to serve food or drink to customers. Modern technologies and system are integrated in the robot design, but still ensure the suitable price.

Serving mobile robot specification:

* Sensor: LiDAR, Ultrasonic sensor, IMU.
* Navigation method: SLAM.
* Communication: Wi-Fi.
* Body weight: kg.
* Maximum load weight: kg.
* Maximum speed:  m/s.

### Scopes of work

#### Analysis & decision making

This is the first step of designing a mobile robot. General plans for designing the robot structure are outlined, together with researching hardware components which is available on the market and suitable for robot using’s purpose.

#### Hardware design

After analysis to get the design planning, the hardware of robot is begun with calculation of main mechanical and electronic parts.

Scopes of work include:

* Mechanical design: Consider the kinematic model to design robot’s frame and do the calculation to select suitable actuators (DC motor).
* Electronic system design: Select and combine electronic devices: power supply, sensors, computers, actuator drive, … to complete the robot electronic system. Design the motor controller for controlling speed of the robot.

#### Firmware design

System firmware is designed to combine the data taken from peripherals and preprocess all those data using algorithms.

* Sensors: Configure multiple I2C buses in Raspberry Pi 4. Read data from sensors and create Nodes for communicating between ROS 2 and all the sensors.
* Motors: Configure communication method between ESP32 and Raspberry Pi 4 to control the motors and receive data from encoders.
* Algorithms:
  + Program PID controller in Python for driving motor.
  + Program Fuzzy Logic library in C++ for PID controller.
  + Program Kalman Filter and Median Filter in Python for sensors.
  + Program Kinematic Model for calculating Odometry data of the robot.

#### Software design

Mainly work with ROS 2 framework and focus on improving and customizing Navigation Stack for the robot, which includes tasks:

* Create 3D model of robot and restaurant environment for simulation tests. Build transform frame, which represents the position and orientation of different robot's parts corresponding to each other and to the map frame when robot moves.
* Mapping environment with SLAM.
* Localize the robot on the map based on data from odometry, imu sensor and lidar.
* Plan path around obstacles and control the robot as it follows the planned path.
* Convert sensor data into a 2d costmap representation of the robot’s surrounding world.
* Create robot’s operation flow using behavior trees.

#### Experiment & results

The result of speed controller for robot by Fuzzy PID is analyzed and compared with classic PID controller.

Several specific serving cases for robot are also implemented in real life to evaluate working performance of the robot.

# FIRMWARE DESIGN FOR SERVING MOBILE ROBOT

## Robot software platforms

Platforms have recently attracted more interest in the robotics community. Hardware platforms and software platforms both have their own subcategories. A robot software platform consists of the tools needed to create robot application programs, including hardware abstraction, low-level device control, sensing, recognition, SLAM (simultaneous localization and mapping), navigation, manipulation, and package management, libraries, debugging, and development tools. Software platforms have made it possible for many individuals to contribute to the creation of robots, and robot hardware is now being created using the interface that software platforms give.

The smartphone field and the robot field are advancing along the same path. The robot software platform is in a dynamic stage where anybody may become the market leader, even if there is still more to be done compared to the smartphone operating system.



The most popular robot software systems are those that are listed below:

* ROS: Robot Operating System, Open Robotics - U.S.
* OpenRTM: National Institute of Adv. Industrial Science and Technology (AIST) – Japan.
* ERSP: Evolution Robotics Software Platform, Evolution Robotics – Europe.
* OPROS: ETRI, KIST, KITECH, Kangwon National University - South Korea.

## The basis for selecting ROS 2 (second version of ROS)

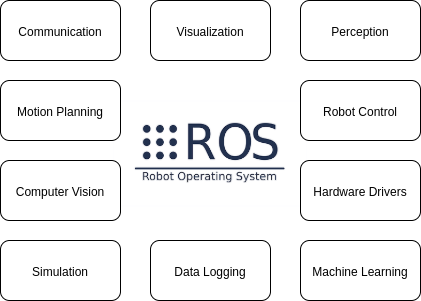
ROS 2 is a robot software platform which helps to reduce the developing and programming time. By entering a few standardized codes, one may quickly convert a non-ROS system to a ROS system without having to completely rewrite the system and programs. Additionally, ROS offers a variety of widely used tools and software and enables users to concentrate on the things they are interested in or would want to contribute to, which eventually cuts down on the time required for development and maintenance.

* The program can be reused: A user may concentrate on the feature they want to build and get the matching package for the other functionalities. They may also reuse the application they created by sharing it with others.

## ROS 2 Introduction

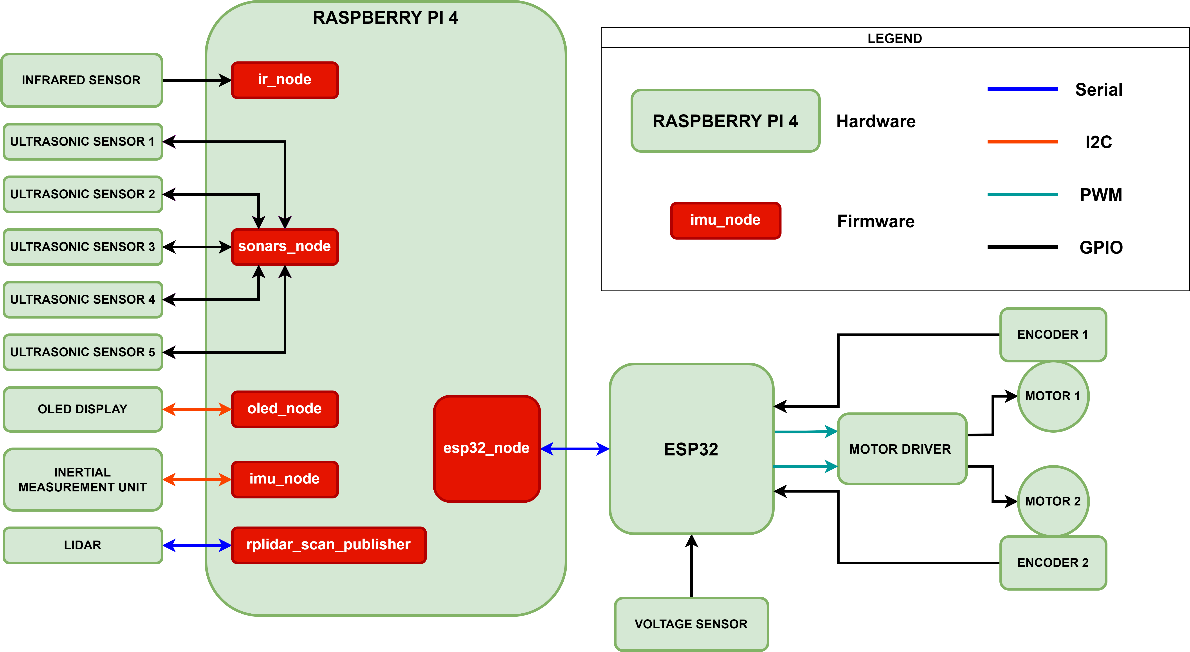
The Robot Operating System 2 (ROS 2) is a collection of open source software libraries, hardware driver framework, state-of-the-art algorithms and robust developer tools to develop a control system for robots.

ROS 2 provides us several components for building and developing a robot or a multi-robot system. Some of them are still being developed and improved, thus they are not yet available. This section will introduce the main elements of ROS 2.



### Communication

## Serving mobile robot firmware architecture



Raspberry Pi 4 and personal computers can both run the ROS 2 system. All of the nodes on the Raspberry Pi 4 are limited to managing peripherals. The other nodes, however, including the SLAM node and the navigation stack, are executed on a personal computer. Each peripheral has an associated node that reads data from the sensor and publishes it to the ROS 2 system.

It takes the right hardware to control motors and receive pulses from encoders, both of which are time-sensitive processes. On the other side, the Raspberry Pi 4 is an inappropriate piece of hardware for applications that need precise timing. The Raspberry Pi 4's RPi.GPIO module is the best option for managing peripherals in Python. It is not, however, appropriate for real-time or timing-sensitive applications. This is due to the fact that Python's garbage collection schedule is unpredictable for developers. Additionally, because Linux is a multitasking operating system and another process could be given priority over the CPU, generating jitter in the program, Linux is not suited for real-time applications. The approach is to use an appropriate microcontroller unit, such as the ESP32 microcontroller, to manage this timing-critical activity. A hardware analog-to-digital converter is also absent from all Raspberry Pi computers. This serves as another evidence that the ESP32, with its 16 ADC pins, is an appropriate MCU for supplying the motor driver with PWM pulses.

## Raspberry Pi 4B

### The change of the ROS version

While the robot and humans share the same workspace in real time, if there are any system latency-related problems, the user might sustain bodily harm. In order to ensure reliable functioning, real-time limitations must be met. Researchers have suggested a number of methods to enable ROS to operate in real-time because it does not currently carry out these requirements. Data Distributed Services (DDS) was designed as middleware for internal communication in ROS 2.0 to facilitate real-time communication. Data is sent via the Real-Time Publish-Subscribe (RTPS) protocol, multicasting, and connectionless transmission techniques like UDP/IP.

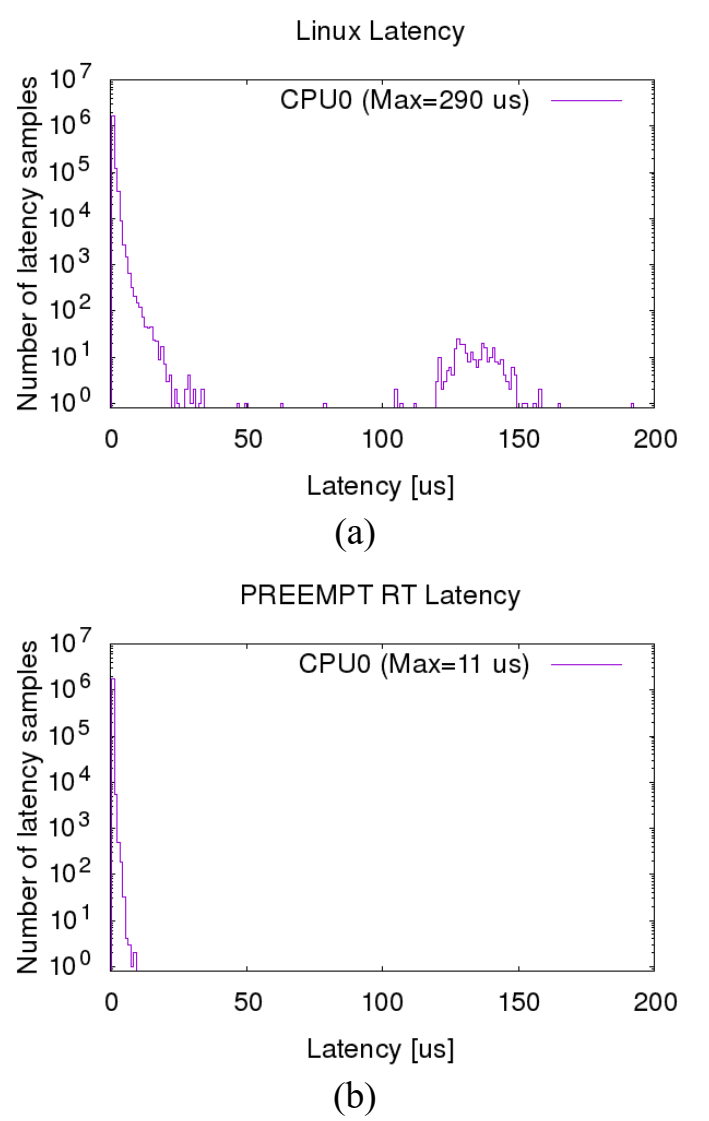


Figure : Scheduling latency in an idle environment. (a) ROS 1.0 (b) ROS 2.0 [3]

As can be observed, the recorded maximum latency, which is 290 us as opposed to ROS 2.0's 11 us, is much greater. The ROS 1.0 findings also display a wider distribution than the ROS 2.0 results, where the majority of the measured data samples are clustered below the highest value. As a result, ROS 2.0 enabled deterministic behavior and more effectively fulfilled the strict temporal requirements than ROS 1.0. Additionally, ROS 1.0 operated inconsistently since it had a wider spread than ROS 2.0. As a result, the ROS 2.0 system worked steadily and offered better real-time performance than ROS 1.0.



Figure : Scheduling latency in a stressed environment (a) ROS 1.0 (b) ROS 2.0 [3]

The greatest delay for ROS 1.0 and ROS 2.0, respectively, was estimated to be 1280 us and 26 us. The maximum latency was higher for both the ROS 1.0 and ROS 2.0 systems when compared to the findings of the performance assessments in idle situations. In this instance, the 1 ms deadline was not met by ROS 1.0. However, ROS 2.0 was still able to achieve the deadline despite with a modest increase in the maximum latency.

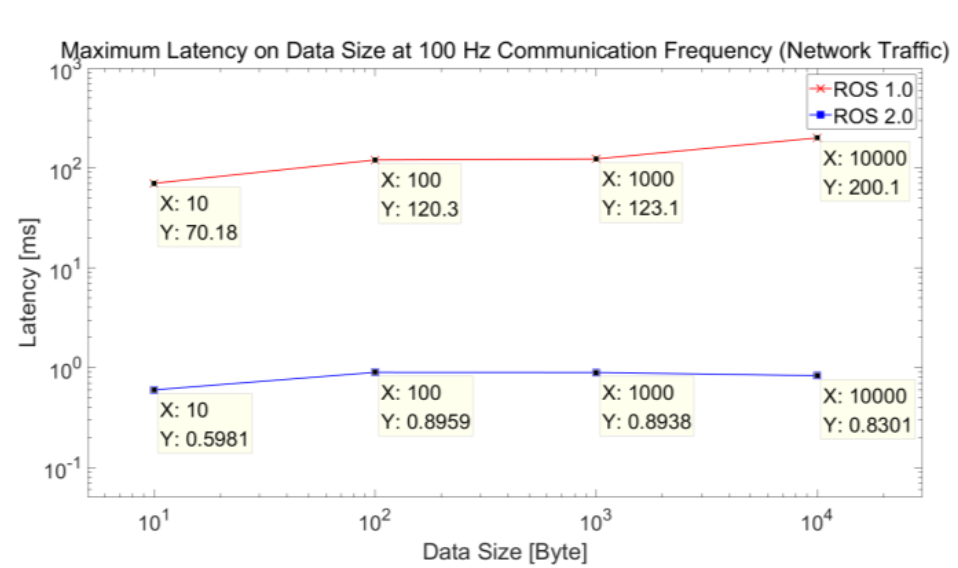


Figure : Maximum communication latency given the data size in environments with network traffic

The outcomes in a dynamic context with network traffic are shown in Fig. 3. The maximum latency was much reduced in ROS 2.0 than in ROS 1.0 in both the steady and idle conditions. While ROS 2.0 demonstrated a little improvement in performance in unreliable situations, latencies in ROS 1.0 made it challenging to transmit and receive messages at the proper time. Additionally, it did not meet the requirements for real-time communication, which makes it challenging to run the system in unsteady conditions.

According to the performance evaluation findings based on data size, ROS 1.0 was unable to meet real-time communication layer requirements and struggled to function properly in unstable environments. The fact that ROS 2.0 has a shorter latency allows it to satisfy these requirements, and because network traffic has a negligible impact on measured performance, steady system operation is achievable even in an unstable environment.

### Ubuntu version

The project has been implemented at the time that Galactic was the latest release of ROS 2.

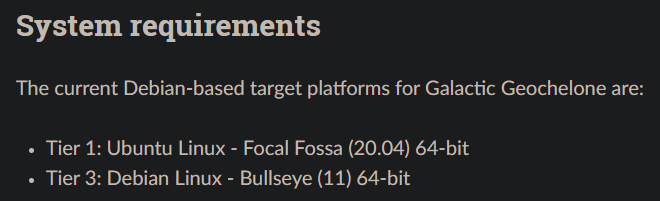


Figure : System requirements from official website for ROS 2 Galactic Geochelone

To have the best support from Operating System, single-board computer Raspberry Pi 4 has Ubuntu Linux – Focal Fossa (20.04) 64-bit installed.

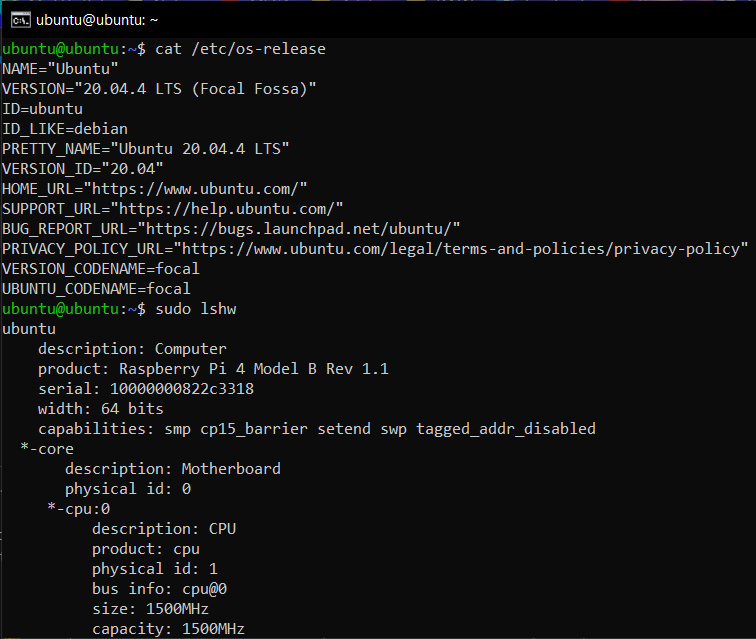


Figure : Ubuntu 20.04.4 LTS on Raspberry Pi 4

### zRAM

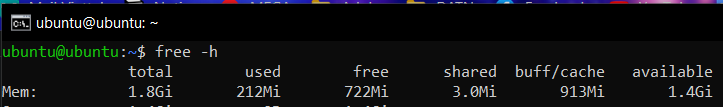


Figure : Initial amount of RAM on Rapsberry Pi 4

For this project, Raspberry Pi 4 is the version which has 1.8 Gigabytes of original RAM.

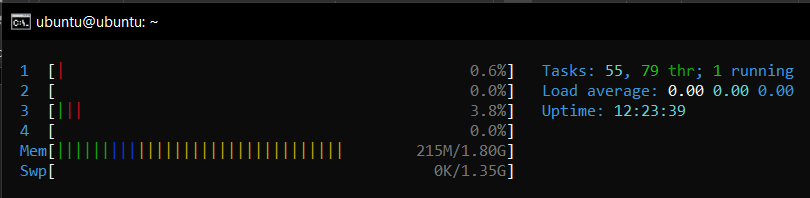


Figure : No-load performance of RAM on Raspberry Pi 4

As observed, when the system is running with no load, the remaining amount of RAM for ROS 2 tasks is 1.4 Gigabytes. During the process of building and running ROS 2 packages, the lag time is quite large and there is a phenomenon where the Raspberry Pi 4 crashes leading to a reboot. To overcome this problem, zRAM was configured for the system of Raspberry Pi 4.

zRam is

https://haydenjames.io/linux-performance-almost-always-add-swap-part2-zram/

### ROS 2 Galactic

## ESP32 Microcontroller

### 

## Motor driver block

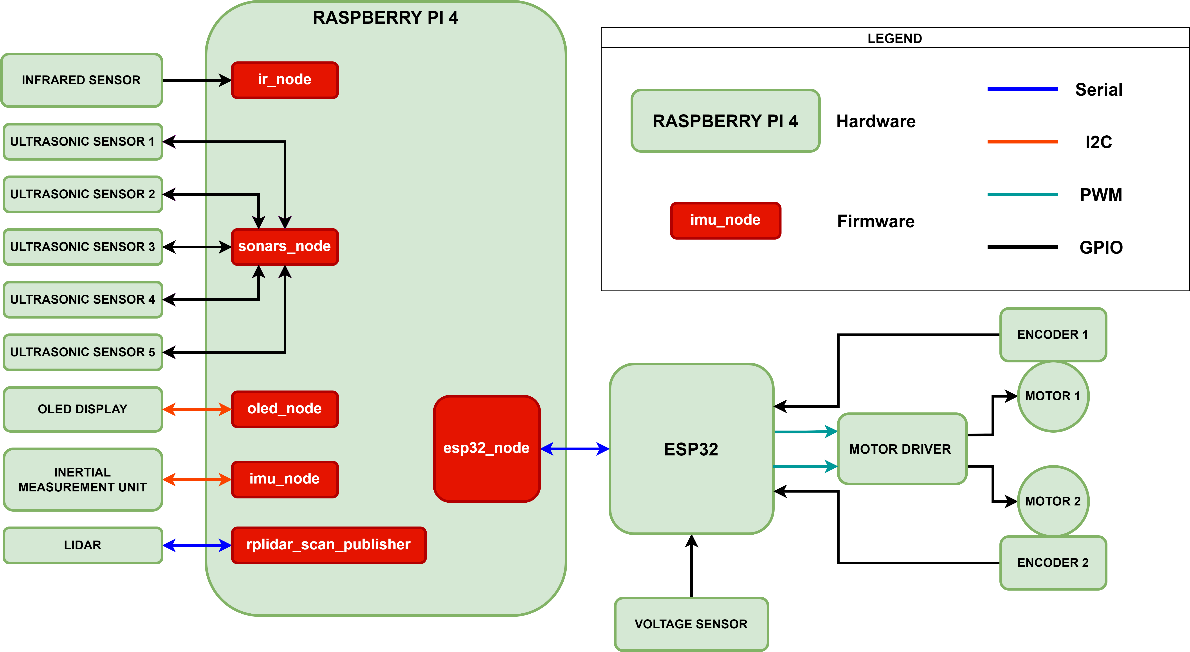


Figure : Motor driver block

Motor driver block includes ESP32, motor driver TB6612FNG, 2 motors and 2 encoders.

### EPS32 firmware

The setup() and loop() procedures are the two primary operations in an ESP32 firmware. When a sketch launches, the setup() function is called. Variables, modes, libraries, and other things are initialized using it. Every time the board is powered on or reset, the setup() code will only be called once. Following the creation of a setup() method, which initializes and sets the starting settings, the loop() function does exactly what it says and performs successive loops, enabling your application to adapt and alter.

#### Setup() function

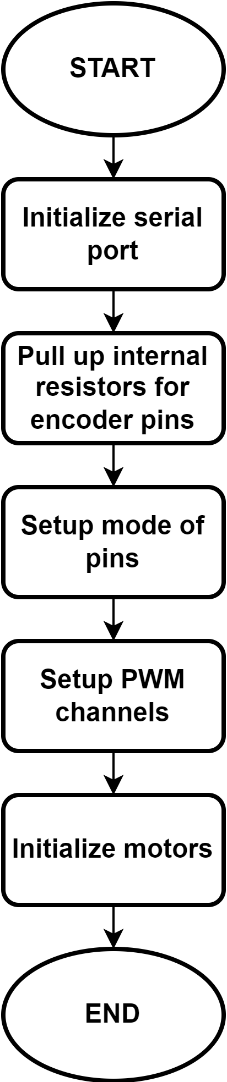


Figure : Flowchart of setup() function for EPS32 firmware

The setup() function includes five successive processes as the Fig. 9. The crucial part in this series of procedures is initializing baud rate for serial port which will be explained later. Pins on the ESP32 microcontroller are configured as input mode or output mode. ESP32's internal resistors pull the input pins, namely the encoder pins, up to a high logic level. By doing this, the pins are prevented from entering an invalid floating condition. The next step is to set up two PWM channels to regulate the speed of the system's two motors.

#### Loop() function

The loop() function includes three vital sections which are receiving data, drive motor and send data back to Raspberry Pi 4.

##### Receiving data from Raspberry Pi 4

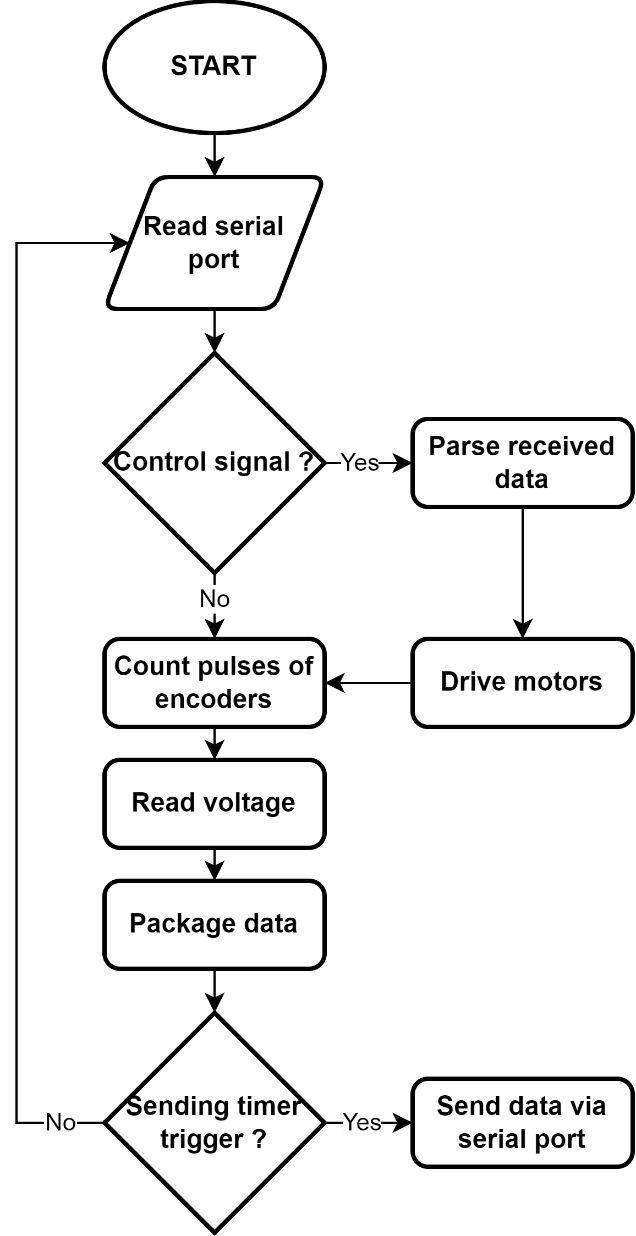


Figure 10: Flowchart of loop() function for EPS32 firmware

The data ESP32 acquired from serial port have the form of JavaScript Object Notation (JSON). The form of data for instance can be observed in Fig. 10:

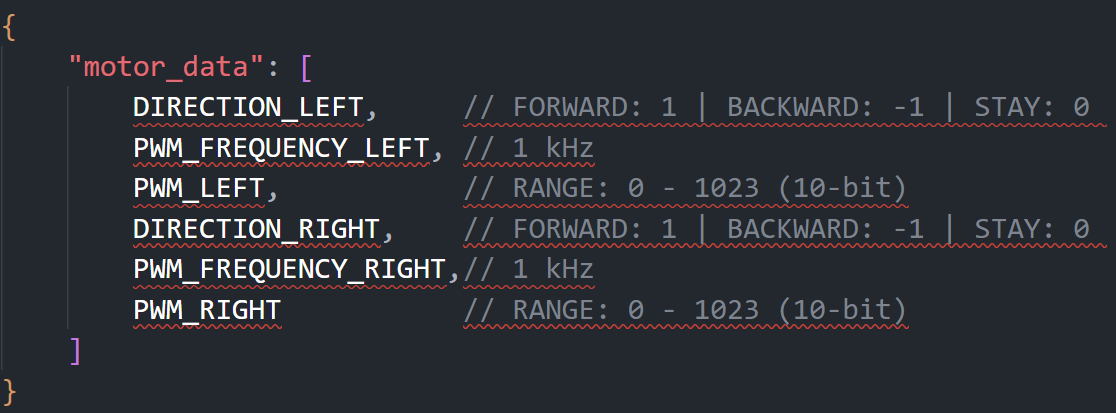


Figure : JSON form of receiving data of ESP32 microcontroller

The direction of the motors is indicated by the direction variable, which has a value of 1, -1, or 0. PWM frequency value selected is 1000 Hz so that the mechanical response of the motor is smooth enough so that there is no sense of tripping due to voltage changes and the motor driver is not overload.

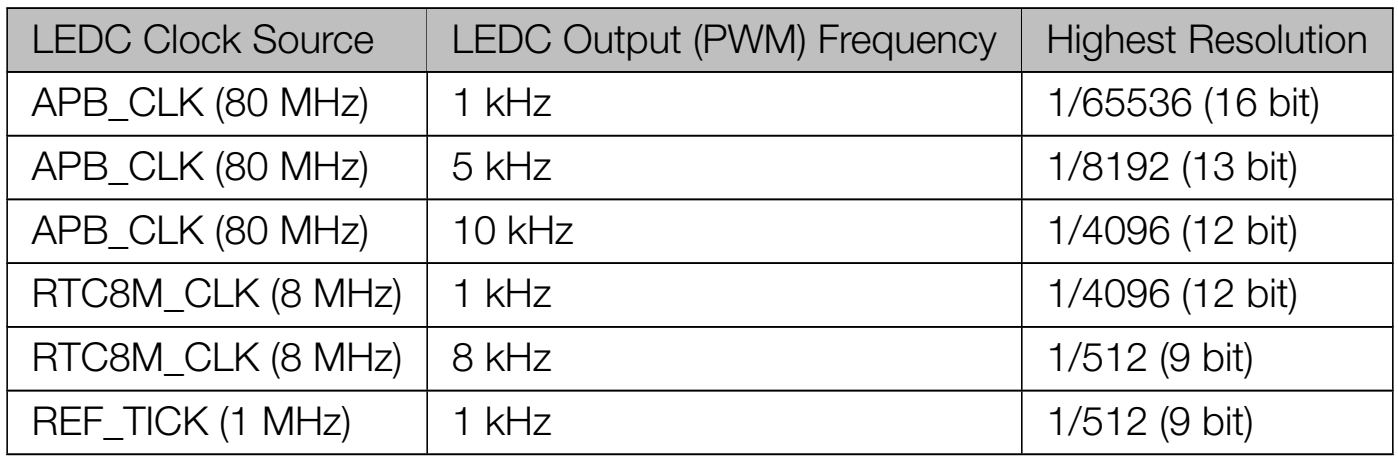


Figure : Commonly­used Frequencies and Resolutions

With a resolution of 10-bit and a PWM frequency of 1 kHz, the motors will operate more smoothly when the resolution is changed between two steps. Moreover, this resolution is within the range of values that make ESP32 works properly as in Fig. 12.

##### Driving motors



Figure : Flowchart of drive\_motors() function for EPS32 firmware

Parsed JSON data will be saved to user-define struct DataReceive before extracted for using in drive\_motor() function. With a program, only one line of code can be run at a time. This causes the motor voltage to happen at two different times causing a delay on one side of the motor. we fixed this by using the STANDBY pin of the TB6612FNG motor driver.

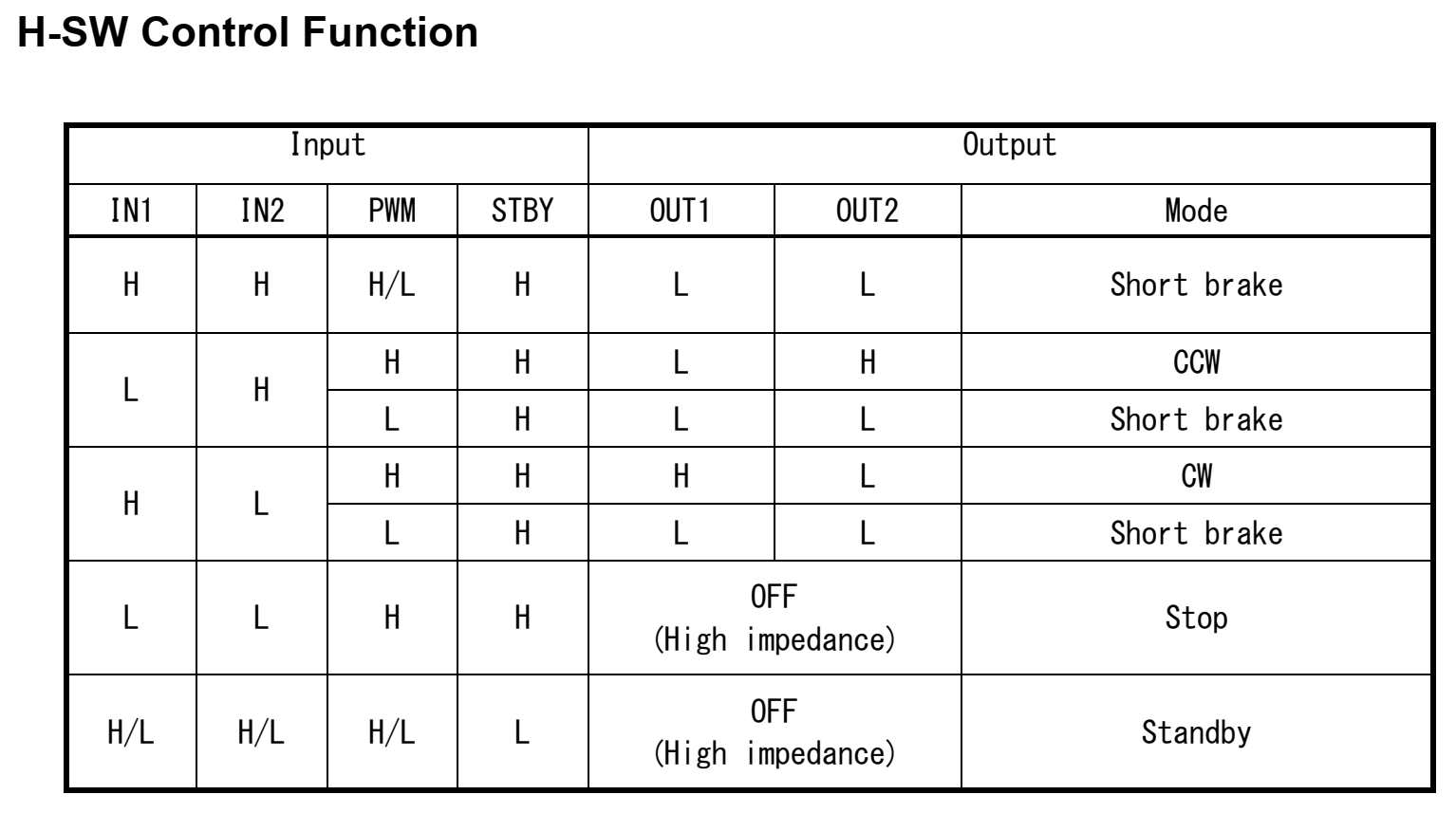


Figure : H-SW Control Function of TB6612FNG motor driver

Both of the TB6612FNG's output PWM pins are disabled while the STANDBY pin is at the low logic level (high impedance mode). The STANDBY pin is switched to a high logic level before the PWM channel value is set using this motor driver function.

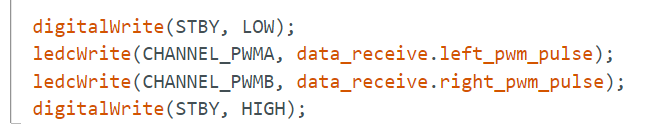


Figure : Solution for instantaneous operating of motors

##### Sending data to Raspberry Pi 4

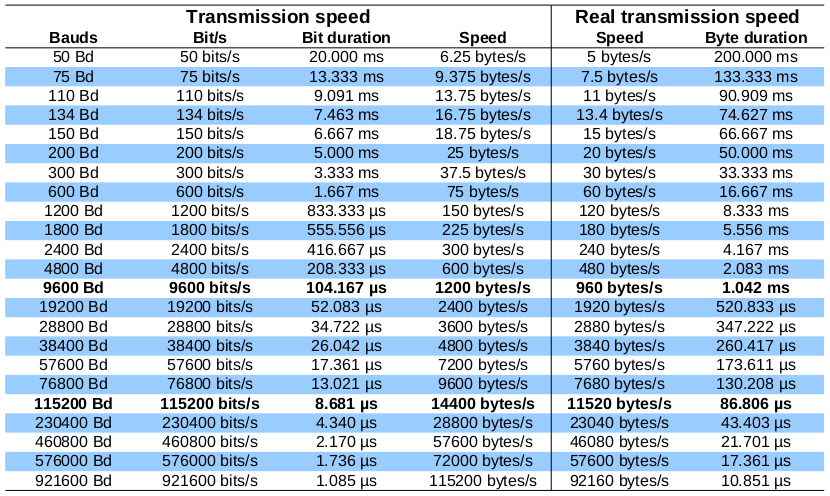
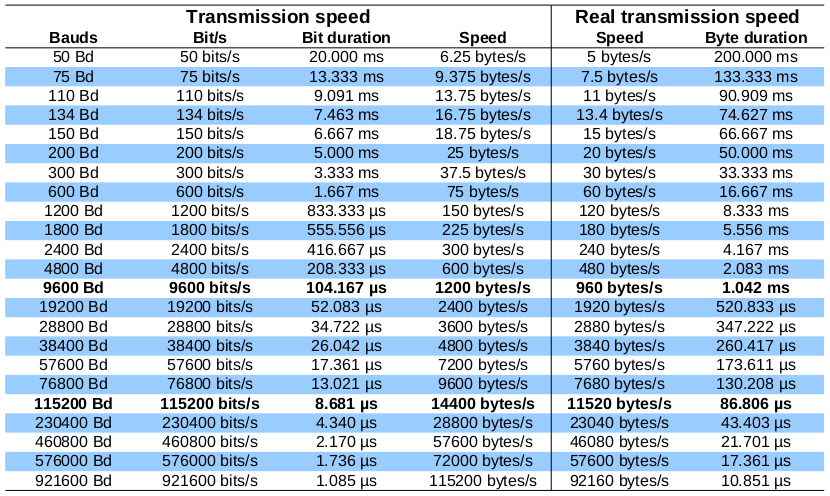


Figure : Transmission speed and Real transmission speed of serial port

### Tạo tiêu đề các cấp

Sử dụng tối đa 3 cấp tiêu đề (ví dụ: 1 hoặc 1.1 hoặc 1.1.1), nếu vẫn muốn tạo thêm các mức khác thì dùng a, b, c… hoặc a), b), c)…hoặc các gạch đầu dòng.

Để tạo tiêu đề với cấp mong muốn: đánh tiêu đề cần tạo, chọn bằng cách bôi đen và chọn “Cap 1” hoặc “Cap 2” hoặc “Cap 3” tương ứng từ thanh công cụ. Khi đó tiêu đề sẽ được tự động đánh số và định dạng.



### Định dạng phần nội dung các chương, mục

Người sử dụng đánh máy nội dung cần trình bày, sau đó chọn toàn bộ đoạn văn bản đó 🡪 bấm chọn Normal trên thanh công cụ. Khi đó phần văn bản vừa tạo sẽ được định dạng đúng theo format yêu cầu của phần nội dung đồ án tốt nghiệp.



Lưu ý: người sử dụng không thay đổi đặc tính của các kiểu style (Normal, Cap 1, Cap 2, Cap 3…) đã được tạo sẵn.

### Hình vẽ - Đồ thị

Hình vẽ hoặc đồ thị (gọi tắt là hình vẽ) có hiệu quả cao khi sử dụng để minh họa cho các nội dung cần tóm lược, do vậy nên được sử dụng để tránh việc đưa các thông tin quá dài.

Hình vẽ có kích thước chiều rộng không quá 75% của chiều rộng nội dung phần chữ, căn lề giữa (trừ các trường hợp đặc biệt có thể rộng hơn hoặc sử dụng trang ngang kiểu Landscapse ).



Chú thích của hình vẽ đặt dưới hình, căn lề giữa, thứ tự hình vẽ theo số thứ tự của chương tương ứng. Để tạo chú thích cho hình vẽ thực hiện như sau:

1. Bấm chọn hình vẽ và bấm phải chuột chọn Insert Caption



1. Chọn New Label 🡪 đánh chữ “Hình” 🡪 chọn OK



1. Sau đó bấm Numbering để tạo định dạng thứ tự cho các chú thích (Hình 1.1, Hình 1.2…) 🡪 tích chọn “Include chapter number” 🡪 chọn Heading 1 🡪 chọn “period”



1. Sau đó đánh nội dung chú thích vào sau chữ Hình….



Với các hình vẽ tiếp theo không cần thực hiện lại các bước trên, chỉ cần chọn hình vẽ 🡪 bấm phải chuột 🡪 Chọn Insert Caption 🡪 bấm Label và chọn Hình 🡪 OK.



### Bảng biểu

Tương tự như hình vẽ, bảng biểu nên có chiều rộng không quá 75% chiều rộng phần chữ của nội dung. Tiêu đề bảng biểu đặt phía trên bảng với cách tạo định dạng tương tự. Bảng biểu nên bố trí để nằm trọn vẹn trong một trang, tránh việc cùng một bảng bị ngắt sang trang khác.



1. Chọn toàn bộ bảng biểu và bấm phải chuột chọn Insert Caption
2. Chọn New Label 🡪 đánh chữ “**Bảng**” 🡪 chọn OK



1. Tại ô Position: chọn Above selected item
2. Bấm Numbering để tạo định dạng thứ tự cho các chú thích (Hình 1.1, Hình 1.2…) 🡪 tích chọn “Include chapter number” 🡪 chọn Heading 1 🡪 chọn “period”



1. Sau đó đánh nội dung chú thích vào sau chữ Bảng….

Bảng . Thống kê các thiết bị và giá thành

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Hạng mục** | **Số lượng** | **Đơn giá** | **Thành tiền** |
| 1 | Đèn bàn | 10 | 10 | 100 |
| 2 | Quạt trần | 10 | 100 | 1000 |
| 3 | Quạt bàn | 10 | 50 | 500 |
| 4 | Bàn học | 10 | 120 | 1200 |
| 5 | Ghế văn phòng | 10 | 70 | 700 |

Với các bảng biểu tiếp theo không cần thực hiện lại các bước trên, chỉ cần chọn hình vẽ 🡪 bấm phải chuột 🡪 Chọn Insert Caption 🡪 bấm Label và chọn Bảng 🡪 OK.

### Phương trình

Để tạo đánh số tự động cho các phương trình thực hiện như sau:

1. Chèn một bảng gồm {1 dòng & 3 cột} tại vị trí muốn có phương trình;

Chỉnh chiều rộng cột 1 khoảng 15% của bảng, chỉnh chiều rộng cột 3 khoảng 15-20% của bảng; còn lại sẽ là cột 2 (khoảng 70% của bảng)

|  |  |  |
| --- | --- | --- |
| Cột 1 (15%) | Cột 2 (~65-70%) | Cột 3 (15-20%) |

1. Bấm chuột vào ô của cột 2 🡪 chọn Insert trên thanh công cụ 🡪 Chọn Equation 🡪 chọn Insert New Equation.



Khi đó sẽ có bảng như sau:



Bấm vào “Type equation here” 🡪 sau đó chọn căn lề trái trên thanh công cụ, mục đích để các chữ trong cột 2 sẽ căn lề trái.



1. Chọn toàn bộ bảng 🡪 Bấm phải chuột chọn Insert Caption 🡪 New Label và tạo nhãn mới có chữ “PT” (tương tự như khi tạo chú thích với Hình và Bảng ở mục 1.2.5 và 1.2.6) 🡪 sẽ tạo ra chú thích với cấu trúc ví dụ như “PT 1.1”
2. Cắt và dán toàn bộ phần chú thích “PT 1.1” vào cột 3, sẽ có bảng như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT . |

1. Chọn toàn bộ bảng 🡪 chọn nút mở rộng của All Border trên thanh công cụ 🡪 chọn No Border



Kết quả sẽ là một bảng không có đường biên dùng để soạn phương trình:



1. Chọn lại toàn bộ bảng này và chọn Insert trên thanh công cụ 🡪 chọn Equation 🡪 “Save Selection to Equation Gallery…”



Khi đó sẽ hiện ra hộp thoại yêu cầu nhập tên của mẫu phương trình vừa tạo, người dùng có thể đánh chữ tạo tên là “Phuong trinh” để dễ nhớ và bấm OK.



Như vậy trong thư viện của Equation đã có một mẫu soạn thảo phương trình với số thứ tự của phương trình tự động thay đổi.

1. Đánh máy phương trình cần tạo vào bảng vừa tạo tại vị trí "Type equation here"
2. Để tạo phương trình tại các vị trí mong muốn khác:

Bấm Insert 🡪 Equation 🡪 kéo thanh trượt xuống dưới và chọn Phuong trình (theo tên vừa đặt)



Khi đó sẽ hiện ra bảng không có đường biên để người dùng soạn phương trình tiếp theo, số thứ tự của phương trình sẽ tự động tăng. Đánh máy phương trình vào ô “Type equation here”

|  |  |  |
| --- | --- | --- |
|  |  | PT . |

Thực hiện tương tự cho các phương trình khác.

## Tạo tham chiếu chéo giữa các đoạn văn bản

Khi viết đồ án nếu cần tham chiếu tới một mục khác hoặc hình vẽ hoặc bảng biểu hoặc phương trình…thì có thể thực hiện tự động. Tham khảo ví dụ sau:

“*Giá trị trung bình của các kết quả thí nghiệm đã được mô tả ở bảng……..; các đánh giá ở mục…. hoàn toàn phù hợp với kết quả được thể hiện ở hình…..*”

Ở ví dụ này cần tham chiếu đến đầu mục, bảng, hình vẽ sẽ thực hiện như sau:

Bấm chuột vào chỗ cần chèn tham chiếu 🡪 chọn Reference 🡪 chọn Cross-reference 🡪 hiện bảng thoại 🡪 chọn mục tương ứng của Reference type:

1. Tham chiếu tới chương, mục 🡪 chọn "Heading" 🡪 với mục “Insert reference to” chọn loại tương ứng là “Paragraph number” 🡪 chọn đầu mục tương ứng trong "For which numberred item:" 🡪 OK
2. Tham chiếu tới hình vẽ, bảng biểu: chọn mục "Reference type" tương ứng với kiểu tham chiếu và thực hiện tương tự như tham chiếu đầu mục.



## Tạo danh mục tài liệu tham khảo

Lưu ý: tài liệu tham khảo là các tài liệu được trích dẫn trong đồ án, không phải là các tài liệu đã đọc. Cách thức trích dẫn và tạo danh mục tài liệu tham khảo theo các bước sau:

* Bước 1: nhập thông tin chi tiết của từng tài liệu tham khảo.
* Bước 2: trích dẫn tài liệu tại các mục cần thiết.
* Bước 3: tạo danh mục tài liệu tham khảo

1. Bước 1: nhập thông tin chi tiết của từng tài liệu tham khảo

Chọn "Reference" trên thanh công cụ 🡪 "Manager Sources" 🡪 hiện hộp thoại "Source Manager" 🡪 chọn "New" để tạo chỉ mục cho tài liệu mới



Thực hiện tương tự các bước trên khi có nhiều tài liệu tham khảo, kết quả sẽ là một cơ sở dữ liệu của các tài liệu dự tính dùng để tham khảo như ví dụ sau:



1. Bước 2: trích dẫn tài liệu tham khảo trong nội dung đồ án

Đặt chuột tại vị trí cần chèn tài liệu tham khảo 🡪 Chọn "Reference" trên thanh công cụ 🡪 chọn kiểu trích dẫn tài liệu trong mục Style là IEEE 🡪 sau đó chọn "Insert Citation" 🡪 chọn tài liệu mong muốn.



Kết quả:

Tóm tắt nội dung của đồ án tốt nghiệp Tóm tắt nội dung của đồ án tốt nghiệp Tóm tắt nội dung của đồ án tốt nghiệp [1] Tóm tắt nội dung của đồ án tốt nghiệp Tóm tắt nội dung của đồ án tốt nghiệp Tóm tắt nội dung của đồ án tốt nghiệp [2] Tóm tắt nội dung của đồ án tốt nghiệp.

1. Bước 3: tạo danh mục tài liệu tham khảo của đồ án

Chuyển tới trang muốn tạo danh mục "TÀI LIỆU THAM KHẢO" và thực hiện theo hướng dẫn sau:

* Tạo trang TÀI LIỆU THAM KHẢO
* Đặt chuột tại ví trí đầu trang 🡪 chọn tab Reference
* Chọn Style kiểu "IEEE"
* Bấm vào Bibliography 🡪 "Work Cited".



Trong trường hợp phần số thứ tự của các tài liệu tham khảo bị lệch dòng so với phần chữ như sau:



Khi đó bấm chuột vào chữ bất kỳ trong danh mục tài liệu tham khảo đang có 🡪 bấm phải chuột và chọn "Paragraph":



Cửa sổ Paragraph sẽ hiện ra 🡪 chọn Special trong mục Indentation 🡪 bấm mũi tên xuống và chọn "(none)" 🡪 chọn OK.



Khi đó toàn bộ các số thứ tự sẽ được căn thẳng hàng với phần chữ của tài liệu tham khảo.



## Cập nhật lại các chú thích và tham chiếu

Trong quá trình soạn thảo đồ án người viết có thể thêm/bớt các đầu mục, xóa các đoạn văn không cần thiết, thêm trang, bớt trang… khi đó cần phải cập nhật lại danh mục các tham chiếu chéo. Các bước cập nhật tham chiếu chéo thực hiện như sau:

* Bước 1: bấm vào chỗ bất kỳ trong văn bản và bấm đồng thời Clt-A để lựa chọn toàn bộ văn bản (hoặc từ tab Home chọn Select rồi chọn Select All).
* Bước 2: bấm phải chuột tại chỗ bất kỳ của văn bản đã được lựa chọn sau đó chọn "Update Field" hoặc bấm phím F9.



* Bước 3: chọn Update entire table và bấm OK.

Có thể sử dụng tùy chọn "Update page number only" nếu quá trình soạn thảo chỉ làm thay đổi số thứ tự các trang.

## Tạo danh mục hình vẽ

Tạo một trang trắng tại vị trí muốn đặt "Danh mục hình vẽ", bấm chọn vị trí đầu trang và thực hiện các thao tác sau:

* Chọn tab "References" 🡪 bấm chọn "Table of Figures":



* Chọn "Hình" trong mục "Caption label" và bấm OK



Danh mục các hình vẽ sẽ được tạo tại trang mong muốn.

Để cập nhật lại đánh số trang, tên hình vẽ…thực hiện như sau:

* Đưa chuột vào vị trí danh mục hình vẽ
* Bấm phải chuột và chọn "Update Field" sau đó tùy chọn chỉ cập nhật số trang hoặc cập nhật cả trang và cả tên, thứ tự hình vẽ.

## Tạo danh mục bảng biểu

Tạo một trang trắng tại vị trí muốn đặt "Danh mục bảng biểu ", bấm chọn vị trí đầu trang và thực hiện tương tự như mục 1.6; tuy nhiên sẽ chọn mục "Bảng" trong mục "Caption label" và bấm OK.



Danh mục các bảng biểu sẽ được tạo tại trang mong muốn. Việc cập nhật lại tên bảng biểu, số trang…tương tự như với danh mục hình vẽ.

## Tạo trang mục lục

Tạo một trang trắng tại vị trí muốn đặt "Mục lục ", bấm chọn vị trí đầu trang và thực hiện các thao tác sau:

* Chọn tab "References" 🡪 bấm chọn "Table of Contents":



* Chọn kiểu "Automatic Table 1".
* Mục lục sẽ được tạo tại trang mong muốn.

Việc cập nhật lại mục lục thực hiện tương tự như với danh mục hình vẽ.

## Qui cách đóng quyển

Phần bìa trước chế bản theo qui định; bìa trước và bìa sau là giấy liền khổ. Sử dụng keo nhiệt để dán gáy khi đóng quyển thay vì sử dụng băng dính và dập ghim.





Phần gáy ĐATN cần ghi các thông tin tóm tắt sau:

Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên

Ví dụ:

**2019.1 - VẬT LÝ KỸ THUẬT - NGUYỄN VĂN A - 20141234**

Qui cách ghi chữ phần gáy như hình sau:



# SỬ DỤNG CÁC BIỂU ĐỒ

## Giới thiệu về biểu diễn bằng đồ thị

Trong rất nhiều lĩnh vực cần phải trình bày, giới thiệu các thông tin liên quan tới con số, thống kê hay các dữ liệu khác. Các dữ liệu đo đạc, tính toán thường được thu thập dưới dạng bảng biểu; tuy nhiên bảng biểu chỉ thích hợp khi trình bày các số lượng nhỏ các số liệu, đồng thời không cung cấp các đánh giá trực quan về xu hướng của dữ liệu thu được.

Đồ thị có khả năng cung cấp hình ảnh trực quan, dễ hiểu giúp người đọc nhanh chóng nắm bắt được ý tưởng muốn nhấn mạnh, muốn trình bày. Người trình bày cần lựa chọn đúng loại đồ thị và không nên sử dụng các đồ thị quá màu mè; lựa chọn tên đồ thị ngắn gọn, dễ hiểu. Các loại đồ thị thường gặp là:

* Kiểu bánh (Pie charts)
* Kiểu thanh ngang & dọc (kiểu cột) (Horizontal & Vertical bar charts)
* Kiểu đường & Kiểu phân bố (Line charts & Scatter diagrams)
* Kiểu diện tích (Area charts)

Phần tiếp theo sẽ khuyến cáo về phạm vi sử dụng của từng loại đồ thị này.

## Đồ thị kiểu bánh

Phạm vi sử dụng:

* Dùng để biểu thị tỷ lệ phần trăm (%)
* Biểu diễn mối liên hệ tương quan tỷ lệ
* Không nên dùng quá nhiều miếng (tối đa 6 miếng) trong một đồ thị



Khi muốn nhấn mạnh một đại lượng:

* Để diễn tả phần quan trọng: đặt phần quan trọng này ở phía trên, bên phải, tính từ vị trí 1 giờ
* Khi cần nhấn mạnh: có thể kéo phần nhô này ra khỏi đồ thị (Hình 2.1 nhấn mạnh về tỷ trọng phần trăm của ngô là nhỏ nhất)



Hình . Đồ thị kiểu bánh

## Đồ thị kiểu thanh ngang

Phạm vi sử dụng:

* Khi muốn so sánh độ lớn hoặc kích thước
* Không nên dùng quá 5 thanh trong một đồ thị

Khi muốn nhấn mạnh một đại lượng:

* Sử dụng vị trí các thanh hợp lý để diễn tả ý muốn nhấn mạnh; không nên đặt các thanh ngẫu nhiên vì có thể gây phân tán suy nghĩ và không diễn tả được ý.
* Dùng các màu khác biệt nhiều để diễn tả đại lượng quan trọng.

So sánh 02 đồ thị trong Hình 2.2 sẽ thấy đồ thị bên trái biểu diễn được ý tưởng muốn nhấn mạnh.



Hình . Đồ thị kiểu thanh ngang

## Đồ thị kiểu cột đứng

Phạm vi sử dụng:

* Khi muốn diễn tả sự thay đổi theo thời gian
* Không nên dùng quá 5 cột trong một đồ thị

Khi muốn nhấn mạnh một đại lượng:

* Khi trình bày nên giản lược đồ thị, bỏ những dữ liệu không cần thiết
* Xem xét dùng đồ thị con để diễn tả sự đóng góp của các thành phần vào sự thay đổi theo thời gian
* Tô màu, gạch chéo hoặc dùng mũi tên để diễn tả những điểm đặc biệt

Hình . Đồ thị kiểu cột đứng

## Đồ thị kiểu đường

Phạm vi sử dụng:

* Biểu diễn xu hướng biến đổi của dữ liệu
* Có tác dụng so sánh nhiều dữ liệu theo thời gian
* Không nên dùng quá 3 đường dữ liệu trong một đồ thị

Khi muốn nhấn mạnh một đại lượng:

* Dùng đường nét đậm
* Đồ thị có nhiều đường: dùng nét đậm và màu nổi bật



Hình . Đồ thị kiểu đường

## Đồ thị kiểu diện tích

Phạm vi sử dụng: phù hợp để biểu diễn so sánh sự thay đổi về số lượng theo thời gian

Các lưu ý:

* Phần đáy đồ thị nên dành cho đại lượng có giá trị lớn nhất (Hình 2.5 nhân mạnh mục tư vấn đóng góp tăng trưởng mạnh nhất so với các mục khác)
* Dùng màu đậm nhất cho phần diện tích đáy; màu đậm sẽ có tác dụng tạo hiệu ứng như “neo” đồ thị, người đọc sẽ nhìn thấy và cảm thấy chắc chắn, hợp mắt
* Các tên chú thích nên để nằm ngang cho dễ đọc
* Đồ thị kiểu này cần nhiều thời gian để quan sát, do vậy nếu sử dụng cho poster thì cần dành nhiều thời gian cho độc giả tìm hiểu.



Hình . Đồ thị kiểu diện tích

# KẾT LUẬN

## Kết luận

Nội dung phần kết luận này tùy thuộc vào từng đồ án. Lưu ý trong phần kết luận không nên có bất cứ phương trình, biểu đồ hay bảng biểu nào. Cần trình bày rõ nội dung đồ án tốt nghiệp đã đáp ứng đầy đủ các yêu cầu của đề bài hay chưa. Trình bày về ý nghĩa của các kết quả thu được, các đánh giá nhận xét về tính khả thi, tính chính xác của kết quả, tính thực tế của đồ án…Cần lưu ý hạn chế sử dụng các tính từ, trạng từ mạnh trong khi miêu tả kết quả đạt được, cần đảm bảo tính trung thực của các kết luận.

Trình bày các kiến thức mà sinh viên đã đạt được sau khi thực hiện đồ án tốt nghiệp. Đồng thời trình bày về các kỹ năng đã học được (kỹ năng tự tìm kiếm tài liệu, tổng hợp thông tin, kỹ năng chế bản, kỹ năng trình bày, viết báo….).

## Hướng phát triển của đồ án trong tương lai

Nêu tóm tắt hướng mở rộng của đề tài trong tương lai nếu có. Đây là mục tùy chọn vì phụ thuộc vào loại đề tài.

# References

|  |  |
| --- | --- |
| [1] | Espressif-Systems, *ESP32 Series Datasheet,* 2022. |
| [2] | Toshiba-Cooperation, *TB6612FNG Datasheet,* 2007. |
| [3] | J. Park, R. Delgado and B. W. Choi, "Real-Time Characteristics of ROS 2.0 in Multiagent Robot Systems: An Empirical Study," vol. 8, pp. 154637-154651, 2020. |

# PHỤ LỤC

1. **Chi tiết số liệu thí nghiệm**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có).

1. **Chi tiết các bước tính toán**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có).

1. **Chi tiết sơ đồ mô phỏng**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trìn