

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN ĐIỀU KHIỂN VÀ TỰ ĐỘNG HOÁ



BÁO CÁO ĐỒ ÁN 2

ĐỀ TÀI: THIẾT KẾ XE GIAO HÀNG TỰ ĐỘNG

GIẢNG VIÊN HƯỚNG DẪN

THẠC SỸ. TRẦN QUỐC TIẾN DŨNG

SINH VIÊN THỰC HIỆN

PHAN TRUNG HẬU – 1913307

NGUYỄN NGỌC HOÀNG – 1913443

TP. HỒ CHÍ MINH, 2023

MỤC LỤC

CHƯƠNG 1. NHIỆM VỤ ĐỀ TÀI	1
CHƯƠNG 2. GIỚI THIỆU PHẦN CỨNG	2
2.1. Vi xử lý STM32F407	2
2.3. Module Sim A9G GSM GPRS GPS BDS.....	5
2.4. Động Cơ DC Servo GM25-370.....	7
2.5. USB UART CP2102	9
2.6. Camera Raspberry Pi V2 IMX219 8MP	10
2.7. Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh	11
2.8. Pin Sạc 18650 3350mAh 3.7V.....	12
2.9. Bánh Xe V3 80mm Khớp Lục Giác 12mm	12
2.10. Tính toán thông số của xe dựa vào các thông số phần cứng	13
2.10.1 Thông số phần cứng.....	13
2.10.2 Tính toán các thông số của xe	13
2.11. Sơ đồ hoàn thiện phần cứng.....	14
CHƯƠNG 3. THIẾT KẾ PHẦN MỀM.....	16
3.1. Xử lý tín hiệu GPS.....	16
3.2. Điều khiển tốc độ 2 động cơ	18
3.2.1. Bộ điều khiển PID	19
3.2.2. Kết quả tìm thông số PID và đáp ứng của 2 động cơ	20
3.3.2 Các thông số cần chú ý trong chương trình	21
3.4. Chương trình điều khiển xe thông qua Bluetooth trên Raspberry pi 4.....	22
3.4.1. Lưu đồ thuật toán	22
3.4.2. Ứng dụng trên điện thoại giúp giao tiếp Bluetooth với raspberry	22
3.4.3. Nhận xét	23
3.5. Chương trình trên raspberry pi giúp thu thập dữ liệu từ camera	24
CHƯƠNG 4. SẢN PHẨM VÀ KẾT QUẢ	25
4.1. Sản phẩm.....	25
4.2. Kết quả	26
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI	31

5.1. Kết luận	31
5.2. Hướng phát triển đề tài	31

MỤC LỤC HÌNH ẢNH

Hình 1. Vi xử lý STM32F407	2
Hình 2. Raspberry pi 4.....	3
Hình 3. Cấu hình chân Raspberry pi 4	4
Hình 4. Module Sim A9G GSM GPRS GPS BDS.....	5
Hình 5. Cách hoạt động của module	6
Hình 6. Cấu hình chân của Module A9G	6
Hình 7. Động Cơ DC Servo GM25-370.....	7
Hình 8. USB UART CP2102.....	9
Hình 9. Camera Raspberry Pi V2 IMX219 8MP	10
Hình 10. Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh	11
Hình 11. Pin Sạc 18650 3350mAh 3.7V	12
Hình 12. Bánh Xe V3 80mm Khớp Lục Giác 12mm.....	12
Hình 13. Bản vẽ của xe trên AutoCAD	14
Hình 14. Schematic của các thiết bị trên xe	15
Hình 15. Sơ đồ khối hoạt động của xe	15
Hình 16. Lưu đồ giải thuật quá trình xử lý GPS	16
Hình 17. Cách thức hoạt động của Traccar	18
Hình 18. Hàm truyền rời rạc bộ điều khiển PID	19
Hình 19. Đáp ứng của động cơ 1 với thông số PID đã tìm được	20
Hình 20. Đáp ứng của động cơ 2 với thông số PID đã tìm được	20
Hình 21. Lưu đồ thuật toán chương trình điều khiển động cơ trên stm32f4.....	21
Hình 22. Lưu đồ thuật toán chương trình điều khiển xe thông qua Bluetooth trên Raspberry pi 4.....	22
Hình 23. Giao diện trên điện thoại của ứng dụng Blue Dot.....	23
Hình 24. Hình ảnh xe nhìn từ trên	25
Hình 25. Hình ảnh xe nhìn từ bên hông	26
Hình 26. Đoạn đường xe chạy trên Traccar	27
Hình 27. Góc nhìn của xe ở lề đường.....	27
Hình 28. Góc nhìn của xe khi chạy trên đường.....	28
Hình 29. So sánh kết quả với thực tế.....	29

Hình 30. So sánh kết quả với thực tế.....	29
---	----

MỤC LỤC BẢNG

Bảng 1. Thông số kỹ thuật của Vi xử lý STM32F407	3
Bảng 2. Thông số kỹ thuật của Raspberry pi 4	5
Bảng 3. Sơ đồ chân của động cơ	8
Bảng 4. Thông số kỹ thuật của Động Cơ DC Servo GM25-370.....	8
Bảng 5. Thông số kỹ thuật của Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh	11
Bảng 6. Cấu trúc dữ liệu GPS nhận được dưới dạng gói NMEA	17
Bảng 7. Kết quả gửi lên server	28

DANH MỤC TỪ VIẾT TẮT

Từ khoá	Ý nghĩa
ADVs	Autonomous delivery vehicles
GPS	Global Positioning System
USB	Universal Serial Bus
DC (Động cơ)	Direct Current
UART	Universal Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface
HTTP	HyperText Transfer Protocol
PWM	Pulse Width Modulation
PID	Proportional Integral Derivative

CHƯƠNG 1. NHIỆM VỤ ĐỀ TÀI

Xe giao hàng tự động (Autonomous delivery vehicles) là phương tiện có khả năng cảm nhận môi trường để di chuyển, vận chuyển hàng hoá tự động mà có sự can thiệp của con người, mà dựa vào sự hỗ trợ của GPS, cảm biến, camera, radar,... để xác định vật cản và hoạch định đường đi.

Lợi ích của việc sử dụng xe giao hàng tự động:

- Tiết kiệm nhiên liệu, giảm khí thải, bảo vệ môi trường.
- Giảm được các vụ tai nạn giao thông.
- Giảm chi phí nhân công, vận chuyển, tăng thời gian hoạt động giúp doanh nghiệp đạt năng suất cao hơn.
- Giúp tăng số lượng người mua qua thương mại điện tử.

Nhưng để đưa xe giao hàng tự động vào hoạt động thì đảm bảo một số yêu cầu:

- Hoạt động an toàn, tuân thủ luật giao thông.
- Có khả năng kiểm soát các hoạt động của xe như phanh, còi, xi nhan, tốc độ, ... và nhận biết được các tín hiệu của các phương tiện trên đường khác.
- Có khả năng nhận diện các vật cản để tránh va chạm, để giữ an toàn giao thông cũng như hàng hoá trên xe.
- Có khả năng kiểm soát và phát hiện các lỗi hệ thống để cảnh báo và khắc phục kịp thời.

Nhu cầu về phương tiện giao hàng tự động (ADV) đang tăng lên do sự bùng nổ của lĩnh vực thương mại điện tử, và được dự đoán sẽ tiếp tục bùng nổ trong tương lai. Cùng với sự xuất hiện của đại dịch Covid, việc giao nhận hàng không tiếp xúc giữa người với người càng được ưu tiên, dẫn đến đây là một cơ hội cho lĩnh vực nghiên cứu xe giao hàng tự động phát triển.

Cho nên ở mức độ đại học, chúng em chọn đề tài **THIẾT KẾ XE GIAO HÀNG TỰ ĐỘNG** cho Đồ án chuyên ngành.

CHƯƠNG 2. GIỚI THIỆU PHẦN CỨNG

2.1. Vi xử lý STM32F407



Hình 1. Vi xử lý STM32F407

Kit STM32F4DISCOVERY thúc đẩy khả năng của vi điều khiển hiệu suất cao STM32F407, cho phép người dùng dễ dàng phát triển các ứng dụng. Kit bao gồm một bộ công cụ gỡ lỗi nhúng ST-LINK, một gia tốc số ST-MEMS, một micro số, một audio DAC với trình điều khiển loa tích hợp lớp D, đèn LED, nút bấm và cổng micro-AB USB OTG.

Các đặc điểm tính năng chính

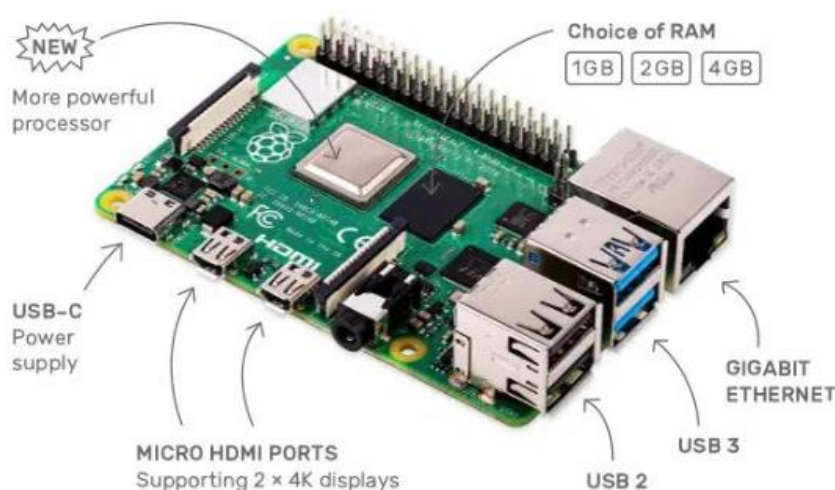
- Bộ vi xử lý STM32F407VGT6: 32-bit ARM Cortex – M4 với lõi FPU, bộ nhớ Flash 1MB, RAM 192KB với kiểu chân LQFP100.
- Trình gỡ lỗi ST-LINK/V2 trên STM32F4DISCOVERY.
- USB ST-LINK với khả năng kiểm tra lại và ba giao diện khác nhau:
 - Cổng gỡ lỗi.
 - Cổng COM ảo.
 - Lưu trữ khối lượng.
- Nguồn cung cấp: thông qua USB hoặc nguồn điện bên ngoài 3V – 5V.
- Gia tốc kế 3 trục LIS302DL hoặc LIS3DSH ST MEMS.
- Micro số đa hướng dùng cảm biến âm thanh MP45DT02 ST-MEMS.
- CS43L22 bộ chuyển đổi âm thanh/kỹ thuật số âm thanh công suất thấp (DAC) với trình điều khiển loa tích hợp lớp D.

- Tám đèn LED:
 - LD1 (đỏ/xanh) cho giao tiếp USB.
 - LD2 (đỏ) led báo nguồn 3.3V.
 - Bốn led của người dùng, LD3 (màu cam), LD4 (xanh lục), LD5 (màu đỏ) và LD6 (màu xanh lam).
 - 2 đèn led OTG USB LD7 (xanh lá cây) VBUS và LD8 (màu đỏ) quá dòng điện.
- Hai nút nhấn (người dùng và reset).
- USB OTG FS có cổng kết nối micro-AB.
- Header mở rộng I/O cho LQPF64 để kết nối nhanh với bo mạch gốc và dễ dàng sử dụng.

Thông số kỹ thuật	Mô tả
Nguồn cấp	3V – 5V DC
Bộ nhớ	1 MB Flash, 192 KB RAM
Hỗ trợ mạch nạp	ST – LINK/V2
Tần số xung Clock	168 MHz

Bảng 1. Thông số kỹ thuật của Vi xử lý STM32F407

2.2. Raspberry pi 4



Hình 2. Raspberry pi 4

Raspberry pi 4 là máy tính cỡ nhỏ sử dụng hệ điều hành Linux. Raspberry Pi 4 là vi xử lý 4 lõi và có ba phiên bản khác nhau có ba dung lượng RAM khác nhau. Pi 4 sử dụng mini HDMI và hỗ trợ hai cổng cho hai màn hình 4K.

Cấu hình chân Raspberry pi 4:

FUNCTION		PIN	PIN	FUNCTION	
3V3	3V3	1	2	5V	5V
GPI02	SPI3 MOSI/SDA3	3	4	5V	5V
GPI03	SPI3 SCLK/SCL3	5	6	GND	GND
GPI04	SPI4 CE0 N/SDA 3	7	8	GPI014	GPI014
GND	GND	9	10	GPI015	GPI015
GPI017		11	12	GPI018	GPI018
GPI027	SPI6 CE1 N	13	14	GND	GND
GPI022	SDA6	15	16	GPI023	GPI023
3V3	3V3	17	18	GPI024	GPI024
GPI010	SDA5	19	20	GND	GND
GPI09	RXD4/SCL4	21	22	GPI025	GPI025
GPI011	SCL5	23	24	GPI08	GPI08
GND	GND	25	26	GPI07	GPI07
GPI00	SPI3 CE0 N/TXD2/SDA8	27	28	GPI01	GPI01
GPI05	SPI4 MISO/RXD3/SCL3	29	30	GND	GND
GPI06	SPI4 MOSI/SDA4	31	32	GPI012	GPI012
GPI013	SPI5 MISO/RXD5/SCL5	33	34	GND	GND
GPI019	SPI6 MISO	35	36	GPI016	GPI016
GPI026	SPI5 CE1 N	37	38	GPI020	GPI020
GND	GND	39	40	GPI021	GPI021
	I2C				Ground
	UART				5V Power
	SPI				3V3 Power

Hình 3. Cấu hình chân Raspberry pi 4

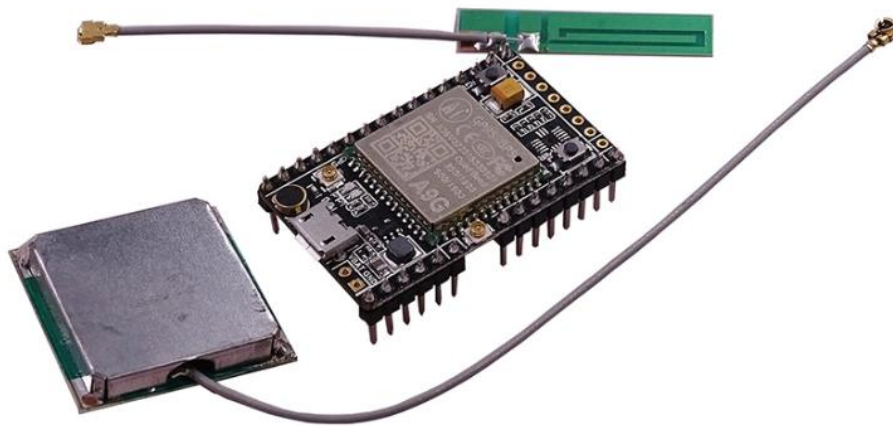
Các tính năng và thông số kỹ thuật của Raspberry pi 4:

CPU	Quad-core Cortex-A72 (64-bit) @ 1.5GHz
GPU	H264 (1080p60 decode, 1080p30 encode) OpenGL ES 3.0 graphics, H.265 (4kp60 decode)
RAM	1GB, 2GB, 4GB.
Dải điện áp hoạt động	5V với dòng tối thiểu 3A
CỔNG GPIO	28 chân I / O

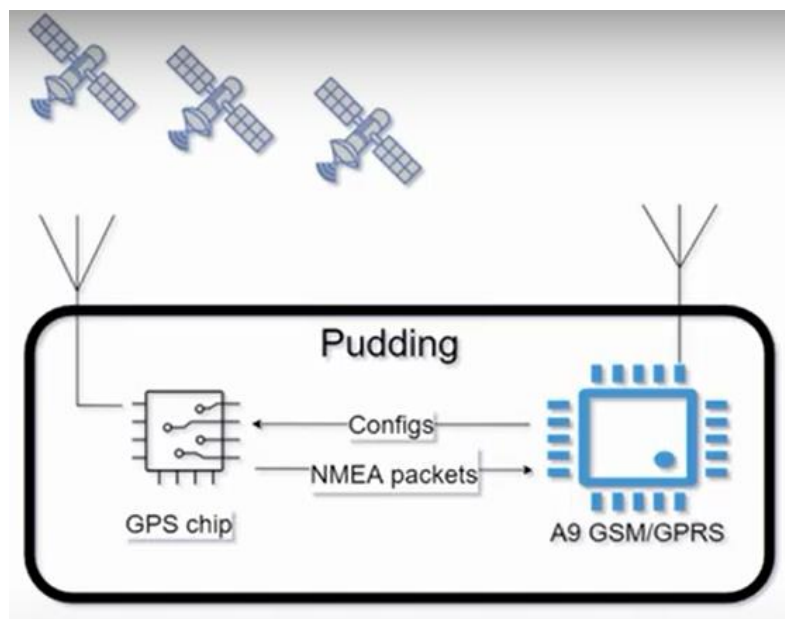
LAN	Có
PoE	Cho phép
WIFI	Có
Bluetooth	5.0
Thẻ SD	Có
HDMI	2 Cổng Màn hình 4k (mini-HDMI)
PWR Exp Header	Không có
Nguồn cấp	Giắc cắm nguồn DC, Cổng USB-C mini
Kết nối mở rộng	40 chân (SPI, I 2 C, LCD, UART, PWM, SDIO)
USB	2×2.0 , 2×3.0
Máy ảnh	CSI
Display	DSI
Nhiệt độ hoạt động	0-50 độ

Bảng 2. Thông số kỹ thuật của Raspberry pi 4

2.3. Module Sim A9G GSM GPRS GPS BDS



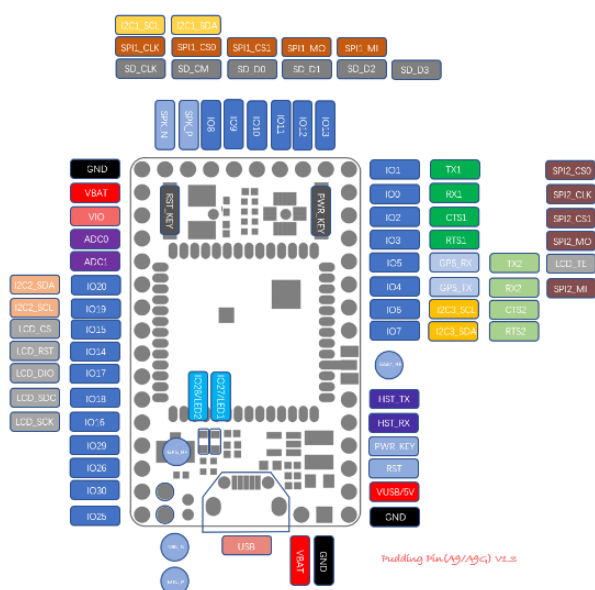
Hình 4. Module Sim A9G GSM GPRS GPS BDS



Hình 5. Cách hoạt động của module

Module Sim A9G GSM GPRS GPS BDS là phiên bản nâng cấp của Module Sim A7 từ nhà sản xuất Ai-Thinker với nhiều cải tiến vượt trội cho khả năng hoạt động ổn định và tiết kiệm năng lượng, mạch có thiết kế nhỏ gọn, tích hợp đầy đủ các tính năng như các module Sim hiện có trên thị trường như Sim900, Sim800,..., ngoài ra mạch còn được tích hợp thêm khả năng định vị GPS, khe cắm thẻ nhớ Micro SD và Microphone cho tính năng Voice.

Cấu hình chân:



Hình 6. Cấu hình chân của Module A9G

Các tính năng kỹ thuật:

- 1 A9G module
- 29 GPIOs cách nhau 2.45mm.
- 1 khe SIM card
- 1 khe TF card
- Cổng giao tiếp: I2C, USB, UART, SDMMC, GPIO, SPI, ADC
- 1 Giao tiếp GPRS
- 1 Giao tiếp GPS
- 1 giao tiếp Micro USB
- Nguồn cung cấp: 5V hoặc 3.8 – 4.2 V
- 1 nút nguồn, 1 nút reset
- 2 LED
- 1 microphone
- Sai số ngang: 2.5 m
- Sai số độ cao: 3.5 m

2.4. Động Cơ DC Servo GM25-370



Hình 7. Động Cơ DC Servo GM25-370

Động Cơ DC Servo GM25-370 DC Geared Motor có cấu tạo bằng kim loại cho độ bền và độ ổn định cao, được sử dụng trong các mô hình robot, xe, thuyền,..., hộp giảm tốc của động cơ có nhiều tỉ số truyền giúp bạn dễ dàng lựa chọn giữa lực kéo và tốc độ (lực kéo càng lớn thì tốc độ càng chậm và ngược lại), động cơ sử dụng nguyên liệu chất lượng cao (lõi dây đồng nguyên chất, lá thép 407, nam châm từ tính mạnh,...) cho sức

mạnh và độ bền vượt trội hơn các loại giá rẻ trên thị trường hiện nay (sử dụng lõi dây nhôm, nam châm từ tính yếu).

Sơ đồ chân của động cơ:

M1	Dây cấp nguồn cho động cơ
GND	Dây cấp nguồn cho Encoder, 0VDC.
C1/A	Kênh trả xung A
C2/B	Kênh trả xung B
VCC	Dây cấp nguồn cho Encoder 3.3~5VDC
M2	Dây cấp nguồn cho động cơ

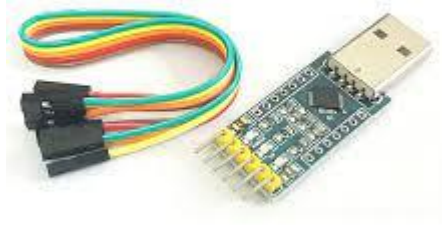
Bảng 3. Sơ đồ chân của động cơ

Thông số kỹ thuật:

Điện áp sử dụng	12 VDC
Đường kính	25 mm
Encoder	Cảm biến từ trường Hall, có 2 kênh AB lệch nhau giúp xác định chiều quay và vận tốc của động cơ, đĩa Encoder trả ra 11 xung/1 kênh/ 1 vòng
Điện áp cấp cho Encoder hoạt động	3.3 – 5 VDC
Tỉ số truyền	34:1
Dòng không tải	150 mA
Dòng chịu đựng tối đa khi có tải	750 mA
Tốc độ không tải	250 RPM
Tốc độ chịu đựng tối đa khi có tải	140 RPM
Lực kéo Moment định mức	4.3 Kg.cm
Lực kéo moment tối đa	5.2 Kg.cm
Chiều dài hộp số	21 mm
Số xung Encoder mỗi kênh trên 1 vòng quay trực chính	374 xung

Bảng 4. Thông số kỹ thuật của Động Cơ DC Servo GM25-370

2.5. USB UART CP2102



Hình 8. USB UART CP2102

Mạch chuyển USB UART CP2102 sử dụng IC CP2102 được dùng để chuyển giao tiếp từ USB sang UART TTL và ngược lại với tốc độ và độ ổn định cao, Driver của mạch có thể nhận trên tất cả các hệ điều hành hiện nay: Windows, Mac, Linux, Android,...

Mô tả chân:

- TXD: chân truyền dữ liệu UART TTL (3.3VDC), dùng kết nối đến chân nhận RX của các module sử dụng mức tín hiệu TTL 3.3~5VDC.
- RXD: chân nhận dữ liệu UART TTL (3.3VDC), dùng kết nối đến chân nhận TX của các module sử dụng mức tín hiệu TTL 3.3~5VDC.
- GND: chân mass hoặc nối đất.
- 5V: Chân cấp nguồn 5VDC từ cổng USB, tối đa 500mA.
- DTR: Chân tín hiệu DTR, thường được dùng để cấp tín hiệu Reset nạp chương trình cho mạch Arduino.
- 3.3V: Chân nguồn 3.3VDC (dòng cấp rất nhỏ tối đa 100mA), không sử dụng để cấp nguồn, thường chỉ sử dụng để thiết đặt mức tín hiệu Logic.

2.6. Camera Raspberry Pi V2 IMX219 8MP



Hình 9. Camera Raspberry Pi V2 IMX219 8MP

Camera Raspberry Pi V2 IMX219 8MP là phiên bản Camera Module dành cho Raspberry Pi mới nhất sử dụng cảm biến ảnh IMX219 8-megapixel từ Sony thay cho cảm biến cũ là OV5647. Với cảm biến IMX219 8-megapixel từ Sony, Camera Module cho Raspberry Pi đã có được sự nâng cấp vượt trội về cả chất lượng hình ảnh, video cũng như độ bền.

Camera Raspberry Pi V2 IMX219 8MP có thể sử dụng với Raspberry Pi để chụp hình, quay phim với chất lượng HD 1080p30, 720p60 hoặc VGA90, cách sử dụng cũng như lập trình với Camera Module trên Raspberry Pi cũng rất dễ dàng, chỉ cần cắm vào cổng Camera trên Raspberry Pi và qua 1 vài bước thiết lập là có thể dùng được.

Thông số kỹ thuật:

- Camera Raspberry Pi V2 8MP dùng cho máy tính nhúng Raspberry Pi.
- Cảm biến IMX219 từ Sony.
- Số điểm ảnh: 8MP.
- Lens: Fixed focus
- Camera specifications:
 - CCD size : 1/4inch
 - Aperture (F) : 2.0
 - Focal Length : 3.04mm
 - Angle of View (diagonal) : 62.2 degree
- Camera Resolution: 3280 x 2464 pixel stills

- Video Resolution: HD 1080p30, 720p60 and 640x480p90 video
- Dimensions: 25mm x 23mm x 9mm
- Connector: ribbon connector
- Interface: CSI

2.7. Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh



Hình 10. Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh

Thông số kỹ thuật

Dung lượng pin	10000mAh
Nguồn vào	USB-micro: 5V/2A (10W) USB-C: 5V/2A (10W)
Nguồn ra	5V/2A (10W)
Wireless Output	Không
Loại pin	Li-Po
Công nghệ/Tiện ích	Smart AI, PD, Quick Charge
Trọng lượng	217g
Hãng sản xuất	Innostyle

Bảng 5. Thông số kỹ thuật của Pin sạc dự phòng Innostyle Powergo Smart AI 10000mAh

2.8. Pin Sạc 18650 3350mAh 3.7V



Hình 11. Pin Sạc 18650 3350mAh 3.7V

Thông số kỹ thuật:

- Dung lượng: 3350 mAh
- Điện áp: 3.7-4.2V
- Size 18x65mm

2.9. Bánh Xe V3 85mm Khớp Lục Giác 12mm



Hình 12. Bánh Xe V3 85mm Khớp Lục Giác 12mm

Thông tin chi tiết

- Chất liệu: Nhựa
- Đường kính bánh xe: 85mm
- Độ dày lớp xe : 10mm
- Chiều rộng bánh xe : 30mm
- Trục xe phi 4mm.

2.10. Tính toán thông số của xe dựa vào các thông số phần cứng

2.10.1 Thông số phần cứng

- Ta có 2 bánh xe gắn với động cơ có đường kính là: $d = 85\text{mm}$.
- Vận tốc tối đa ghi trên động cơ 250 rpm. Nhưng khi chạy thực tế thì vận tốc tối đa của động cơ đạt được là $V_{\max} = 200\text{ rpm}$.
- Moment định mức $T = 4.3\text{ KG.CM}$.
- Dòng chịu đựng của động cơ khi có tải tối đa là : $I_{1\max} = 750\text{ mA}$.
- Điện áp của mỗi viên pin là 3.7-4.2V và có dung lượng là 3350 mAh.
- Giả sử khi mắc nối tiếp 3 pin lại thì ta sẽ có bộ nguồn với điện áp trung bình là
- $V = 12\text{ (V)}$ và nguồn có dung lượng là $I_{t1} = 3350\text{ mAh}$.
- Nguồn 5V cung cấp các vi xử lý và máy tính nhúng có dung lượng $I_{t2} = 10\text{Ah}$.
- Dòng tiêu thụ tối đa của máy tính nhúng là $I_{t2\max} = 3\text{ A}$.
- Dòng tiêu thụ tối đa của Stm32F4 là $I_{t3\max} = 160\text{ mA}$.
- Dòng tiêu thụ tối đa của Module Sim A9G là $I_{t4\max} = 1.14\text{ mA}$.

2.10.2 Tính toán các thông số của xe

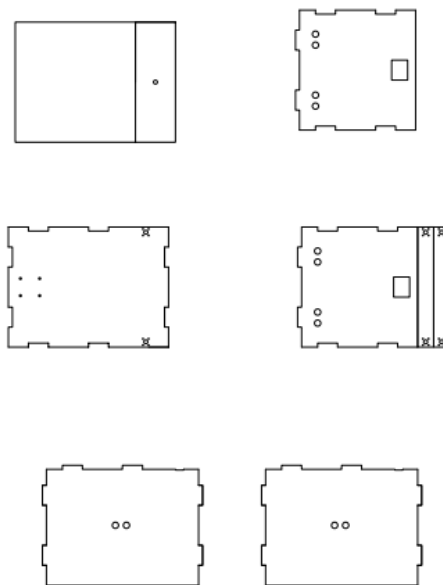
Với các thông số ở trên ta có thể tính toán được một số thông số của xe là :

- Vận tốc tối đa của xe là : $V_{\text{xmax}} = V_{\max} \pi d = 3.2\text{ (Km/h)}$
- Lực đẩy của xe là : $F = \frac{4T}{d} = 2.02\text{ (KG)} = 19.83\text{ (N)}$
- Thời gian trung bình xe chạy được khi sạc đầy pin và động cơ tiêu thụ dòng tối đa là : $t1 = \frac{It1}{2I1\max} = 2.23\text{ (giờ)}$
- Dòng tiêu thụ tối đa của các thiết bị điều khiển là: $I_{dk} = I_{2\max} + I_{3\max} + I_{4\max} = 3.161\text{ (A)}$.
- Thời gian hoạt động tối thiểu của các thiết bị điều khiển trên xe khi sạc đầy pin là : $t2 = \frac{It2}{I_{dk}} = 3.16\text{ (giờ)}$

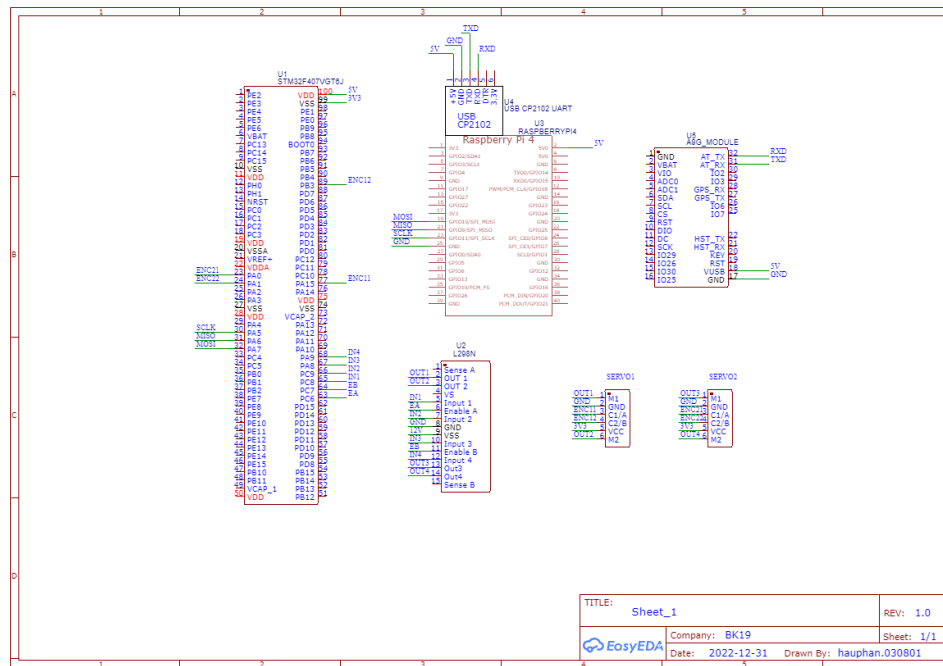
Nhận xét : Các thông số của xe được tính toán dựa trên các thông số định mức của từng linh kiện phần cứng của xe. Những tính toán này chỉ mang tính chất giúp ta ước lượng được các thông số vận hành của xe dựa trên lý thuyết. Nó không phải những thông số thực tế khi xe vận hành.

2.11. Sơ đồ hoàn thiện phần cứng.

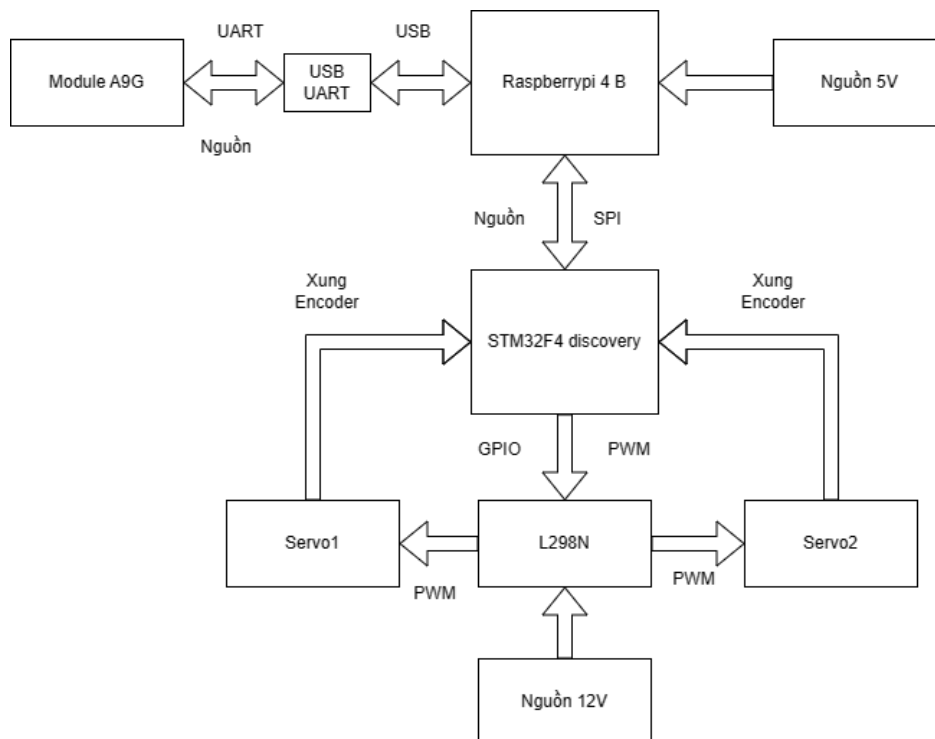
Xe có dạng hình hộp chữ nhật với kích thước: 40x30x30 cm



Hình 13. Bản vẽ của xe trên AutoCAD



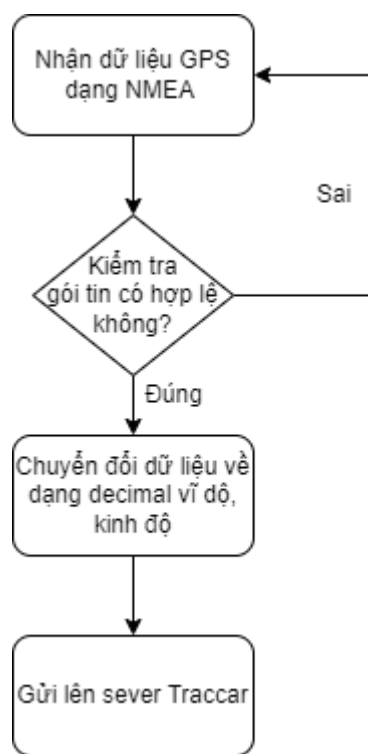
Hình 14. Schematic của các thiết bị trên xe



Hình 15. Sơ đồ khối hoạt động của xe

CHƯƠNG 3. THIẾT KẾ PHẦN MỀM

3.1. Xử lý tín hiệu GPS



Hình 16. Lưu đồ giải thuật quá trình xử lý GPS

Cấu trúc dữ liệu GPS nhận được dưới dạng gói NMEA:

TALKER ID	XX	All talker IDs usable
SENTENCE ID	GGA	
UTC of position	hhmmss.ss	Fixed length 2 digits after dot
Latitude	llll.lllllll	Fixed length 4 digits before and 7 after dot
Hemisphere of latitude	N/S	N if value of latitude is positive
Longitude	lllll.lllllll	Fixed length 5 digits before and 7 after dot
Hemisphere of longitude	E/W	E if value of longitude is positive
GPS quality indicator	X	0: GNSS fix not available 1: GNSS fix valid 4: RTK fixed ambiguities 5: RTK float ambiguities

Number of satellites used for positioning	XX	Fixed length 01 for single digits
HDOP	XX.X	Variable/fixed length 1 digit after dot, variable before
Altitude geoid height	(-)X.XX	Variable/fixed length 2 digits after dot, variable before
Unit of altitude	M	
Geoidal separation	(-)X.XX	Variable/fixed length 2 digits after dot, variable before
Unit of geoidal separation	M	
Age of differential data		Empty field
Differential reference station ID		Empty field

Bảng 6. Cấu trúc dữ liệu GPS nhận được dưới dạng gói NMEA

Ví dụ:

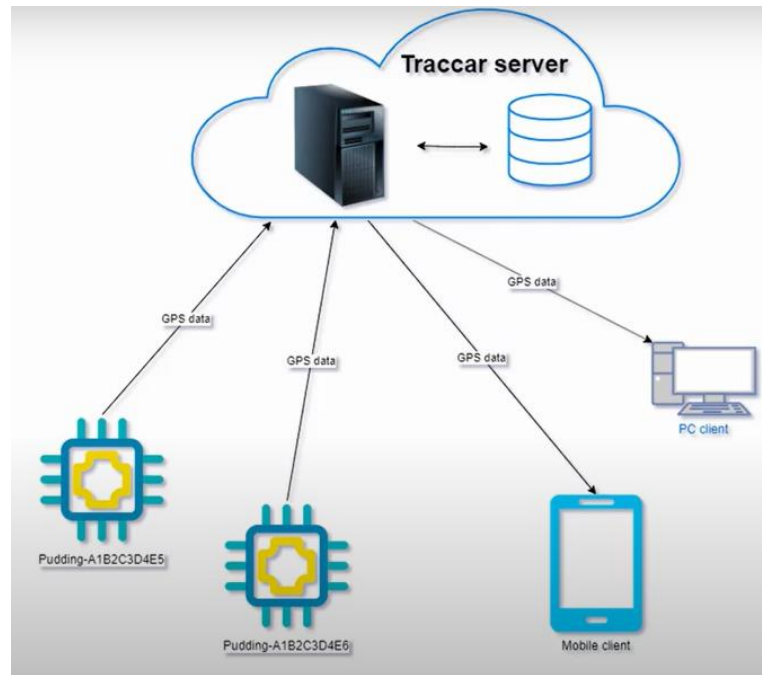
\$GNGGA,185833.80,4808.7402397,N,01133.9325039,E,5,15,1.1,470.50,M,45.65,M,,*75

Vì gói tin NMEA hiển thị giá trị GPS dưới dạng độ và phút nên cần chuyển về decimal degree để web sever có thể hiểu đúng giá trị.

Đoạn code chuyển đổi GPS dạng tin NMEA sang decimal degree.

```
int temp = (int)(gpsInfo->rmc.latitude.value/gpsInfo->rmc.latitude.scale/100);
double latitude = temp+(double)(gpsInfo->rmc.latitude.value - temp*gpsInfo->rmc.latitude.scale*100)/gpsInfo->rmc.latitude.scale/60.0;
temp = (int)(gpsInfo->rmc.longitude.value/gpsInfo->rmc.longitude.scale/100);
double longitude = temp+(double)(gpsInfo->rmc.longitude.value - temp*gpsInfo->rmc.longitude.scale*100)/gpsInfo->rmc.longitude.scale/60.0;
```

Kết nối với server Traccar, ở đồ án 2 này nhóm em dùng sẵn demo của traccar.



Hình 17. Cách thức hoạt động của Traccar

Để gửi giá trị GPS lên Traccar server, ta dùng giao thức HTTP (HyperText Transfer Protocol)

Hàm HTTP mà nhóm dùng:

```
Http_Post(const char* domain, int port, const char* path, uint8_t* body, uint16_t bodyLen, char* retBuffer, int bufferLen)
```

Ví dụ khi gửi một giá trị lên sever Traccar sẽ có dạng:

```
http://demo.traccar.org:5055/?id=11223344554321&lat=10.87834041961587&lon=106.80626571938194
```

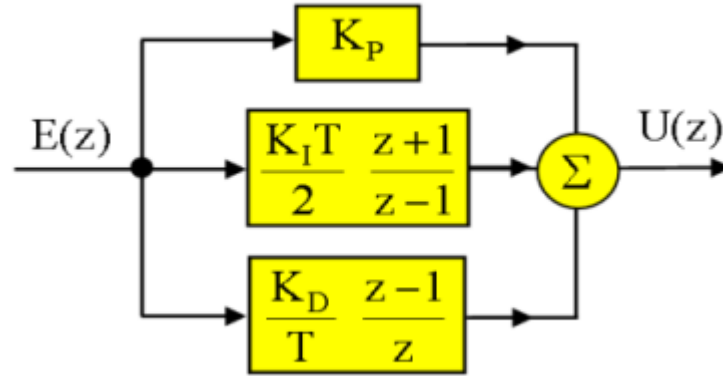
3.2. Điều khiển tốc độ 2 động cơ

Trong giai đoạn này xe chỉ mới vận hành nhằm mục đích đo độ chính xác của cảm biến GPS và khảo sát môi trường thực tế qua camera. Do đó xe sẽ vận hành ở chế độ có người điều khiển và giám sát. Vì vậy việc điều khiển tốc độ hai bánh xe ở thời điểm này là chưa cần có độ chính xác cao. Vì những lý do đó nhóm quyết định sử dụng bộ bộ điều khiển PI có thông số cố định để điều khiển vận tốc hai bánh xe.

Ưu điểm của bộ điều khiển PID trong việc điều khiển vận tốc bánh xe là nó đơn giản, dễ phát triển. Tuy nhiên nó không đáp ứng tốt với những thay đổi từ môi trường bên ngoài (vd: xe thay đổi tải, xe đi lên dốc hoặc xuống dốc v.v) hoặc tốc độ đặt thay đổi trong phạm vi quá lớn. Tuy nhiên những khuyết điểm đó sẽ được nhóm giải quyết trong thời gian tiếp theo của đồ án bằng cách sử dụng các bộ điều khiển để thay đổi thông số điều khiển của bộ PID dựa vào môi trường (vd : bộ điều khiển MRAC ..).

3.2.1. Bộ điều khiển PID

Trong khuôn khổ đề tài này nhóm quyết định chọn bộ điều khiển PID rời rạc để hiện thực bộ điều khiển PID.



Hình 18. Hàm truyền rời rạc bộ điều khiển PID

Từ hàm truyền rời rạc của bộ điều khiển PID ta suy ra biểu thức tính giá trị điều khiển điều khiển đầu ra dựa vào sai số đầu vào như sau:

$$U_{PID} = K_p E(z) + \frac{K_i T_s E(z)(z^{-1} + 1)}{2(1 - z^{-1})} + K_d \frac{1 - z^{-1}}{T_s} E(z)$$

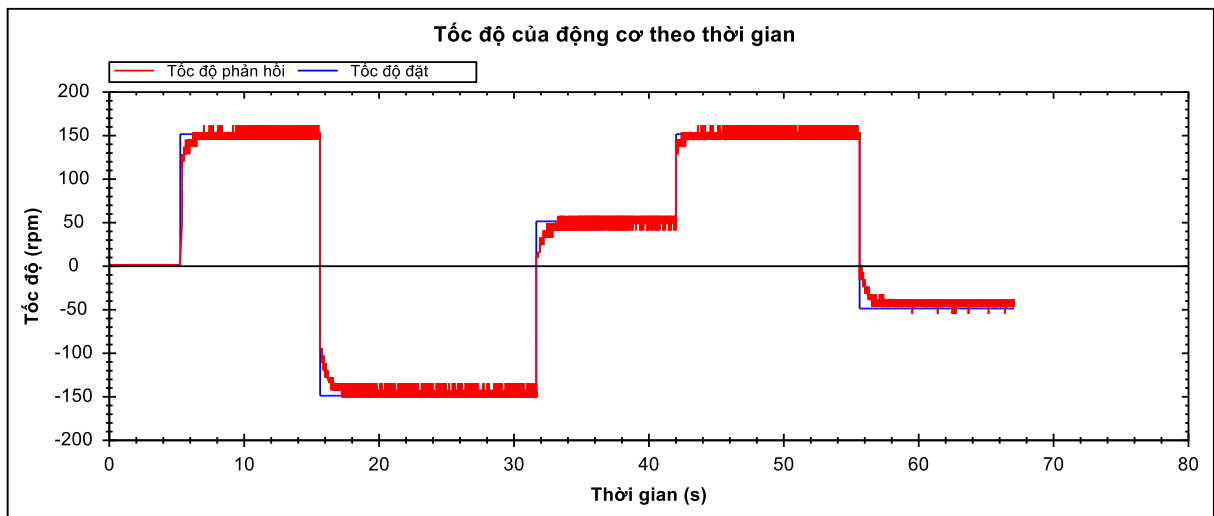
$$U_{PID}(1 - z^{-1}) = K_p E(z)(1 - z^{-1}) + \frac{K_i T_s}{2} E(z)(z^{-1} + 1) + K_d \frac{(1 - z^{-1})^2}{T_s} E(z)$$

$$u_{PID}(k) - u_{PID}(k-1) = K_p (e(k) - e(k-1)) + \frac{K_i T_s}{2} (e(k) + e(k-1)) + \frac{K_d}{T_s} (e(k) - 2e(k-1) + e(k-2))$$

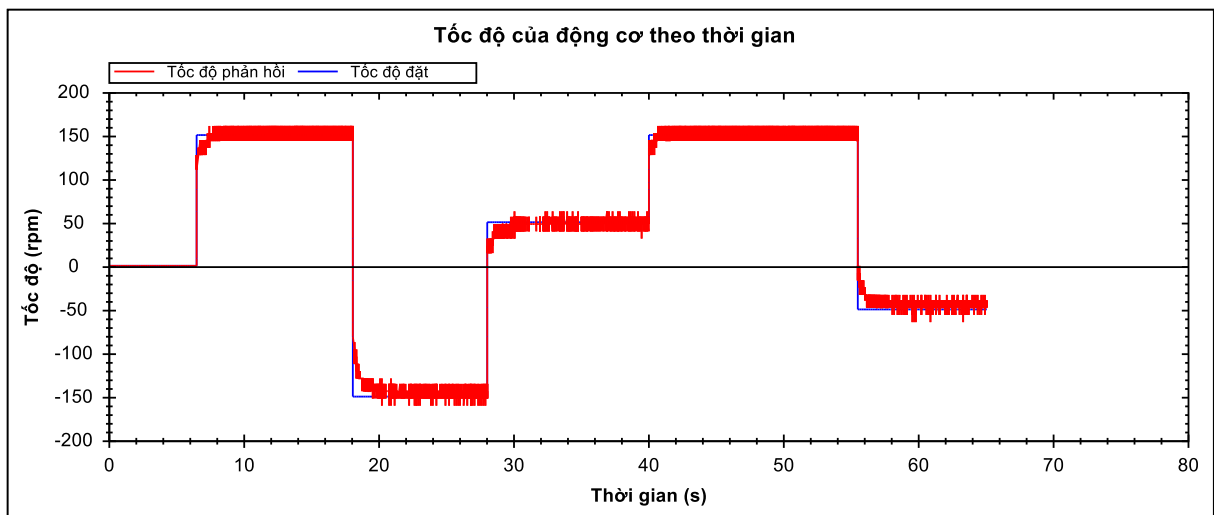
3.2.2. Kết quả tìm thông số PID và đáp ứng của 2 động cơ

Sau khi tìm thông số PID bằng phương pháp thử công với thời gian lấy mẫu cho bộ điều khiển PID là $T_s = 5\text{ms}$, thì thông số điều khiển PID cho cả hai động cơ tìm được là $K_p = 15$, $K_i = 30$, $K_d = 0$.

Với đáp thông số PID như trên thì ta có được đáp ứng của hai động cơ ở chế độ không tải như sau:



Hình 19. Đáp ứng của động cơ 1 với thông số PID đã tìm được

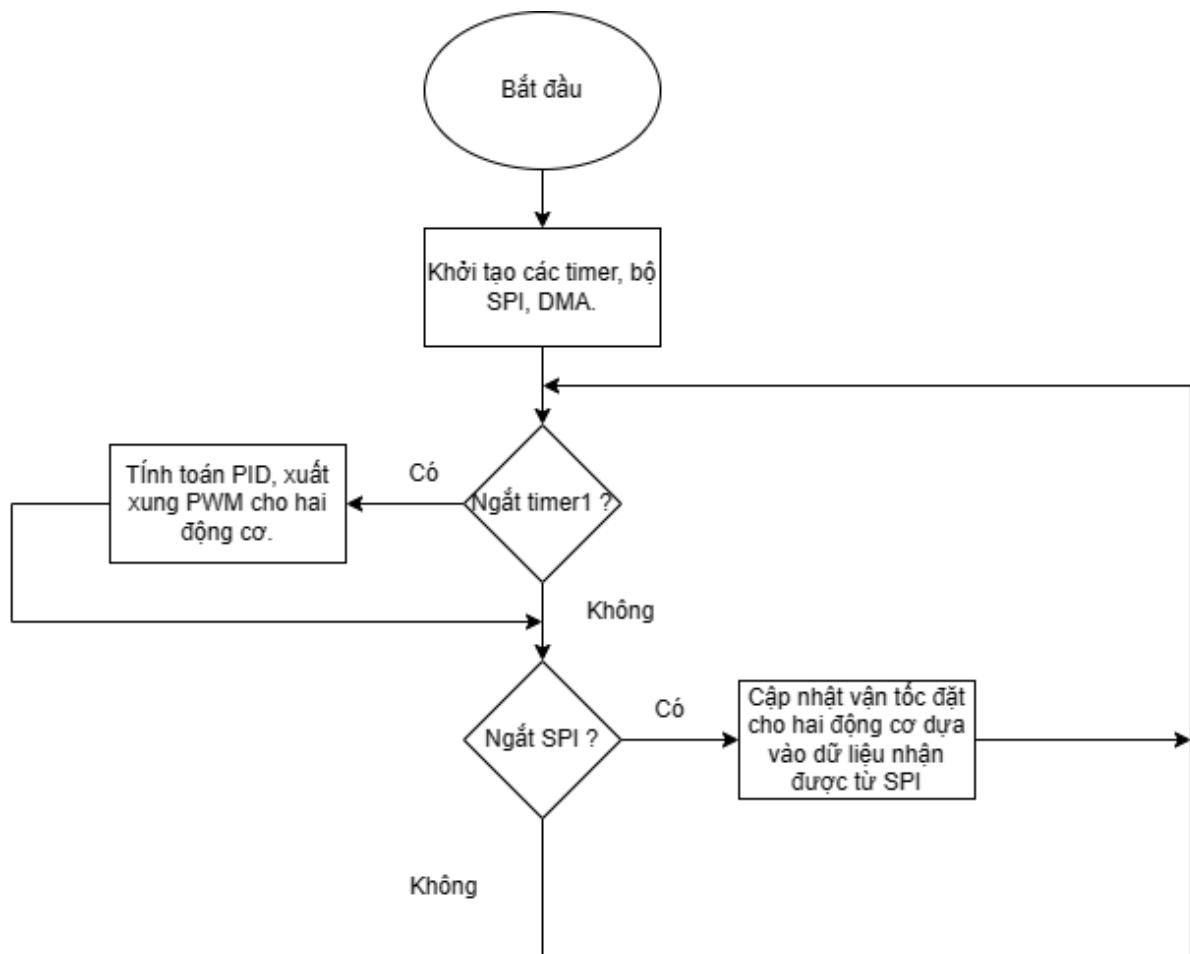


Hình 20. Đáp ứng của động cơ 2 với thông số PID đã tìm được

Nhận xét: Đáp ứng của hai động cơ với thông số PID đã tìm được khá tương đồng. Như vậy ta có thể dễ dàng điều khiển xe hoạt động theo ý mình bằng cách thay đổi tốc độ đặt của hai động cơ.

3.3. Chương trình điều khiển động cơ trên stm32f4

3.3.1. Lưu đồ thuật toán



Hình 21. Lưu đồ thuật toán chương trình điều khiển động cơ trên stm32f4

3.3.2 Các thông số cần chú ý trong chương trình

3.3.2.1 Tần số phát xung PWM

Vì để giảm tiếng ồn khi động cơ hoạt động nhóm quyết định chọn tần số phát xung PWM là 20KHz. Việc phát xung cho PWM cho hai động cơ sẽ do Timer 8 thực hiện.

3.3.2.2 Thời gian lấy mẫu và điều khiển

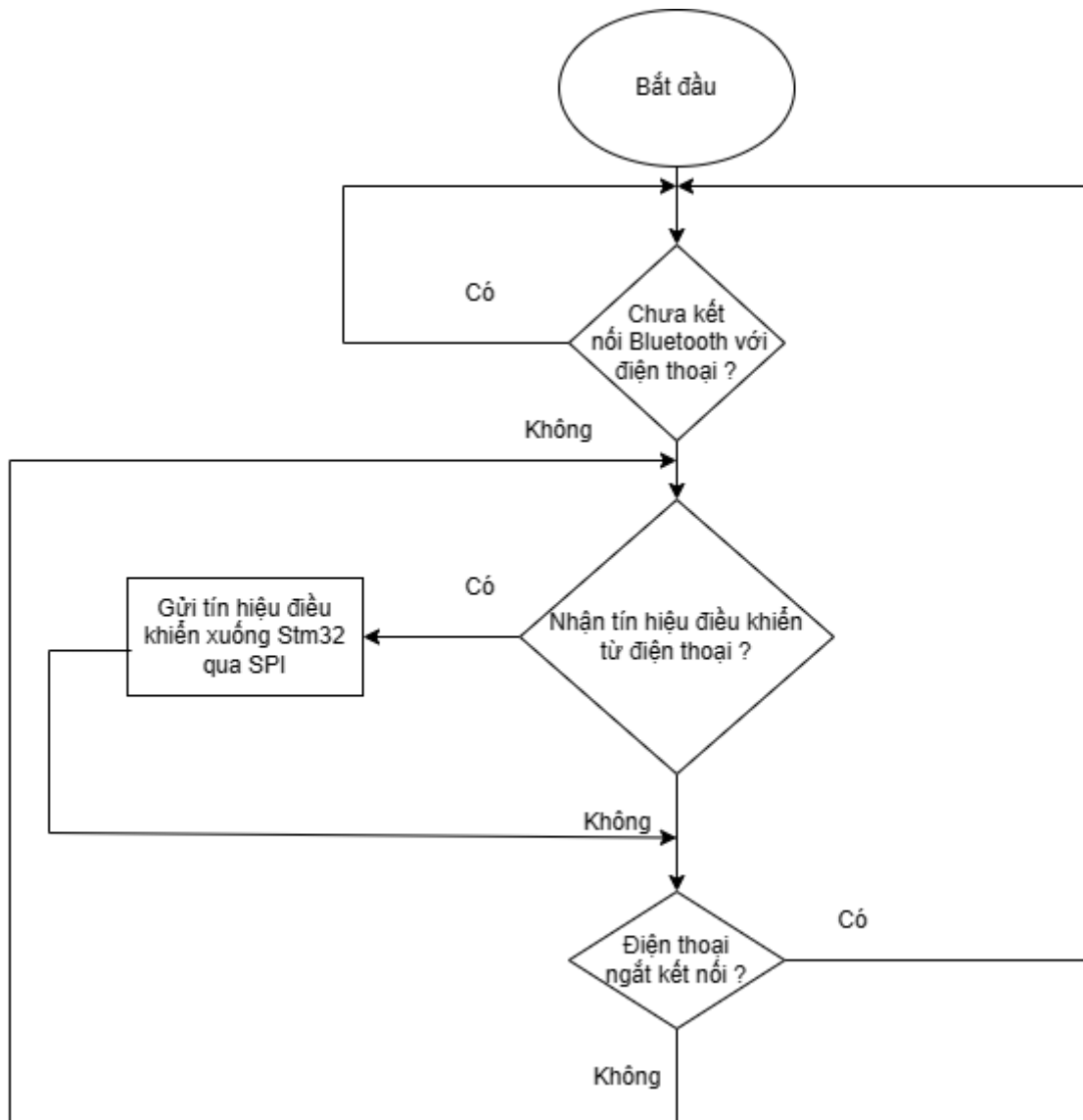
Để việc điều khiển động cơ được chính xác nhóm quyết định lấy thời gian lấy mẫu tín hiệu từ các encoder và điều khiển cho hai bộ PID là 5ms. Việc lấy mẫu này sẽ do Timer1 thực hiện.

3.3.2.3 Chế độ SPI

Vì Stm32f4 là thiết bị nhận tín hiệu điều khiển từ raspberry nên nhóm quyết định cấu hình cho Stm32f4 hoạt động ở chế độ Slave. Bộ SPI1 sẽ được kích hoạt và thực hiện chức năng này.

3.4. Chương trình điều khiển xe thông qua Bluetooth trên Raspberry pi 4

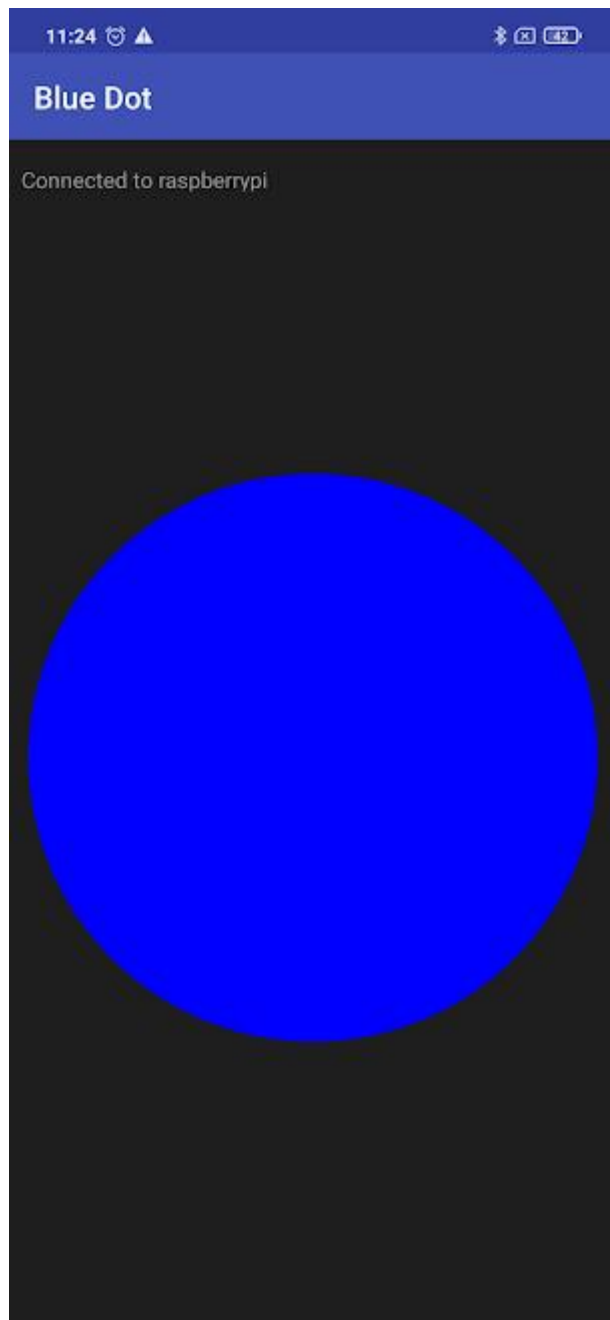
3.4.1. Lưu đồ thuật toán



Hình 22. Lưu đồ thuật toán chương trình điều khiển xe thông qua Bluetooth trên Raspberry pi 4

3.4.2. Ứng dụng trên điện thoại giúp giao tiếp Bluetooth với raspberry

Với mục tiêu là đơn giản và hiệu quả nhóm quyết định chọn công cụ Blue Dot để thực hiện việc giao tiếp bằng Bluetooth giữa raspberry và điện thoại. Giao diện người dùng trên điện thoại của nó khá đơn giản và có sẵn API để ta thực hiện điều khiển xe nên việc công cụ này là cụ thích hợp để đáp ứng mục đích của nhóm.



Hình 23. Giao diện trên điện thoại của ứng dụng Blue Dot

3.4.3. Nhận xét

Mặc dù đây là chương trình phụ nhưng nó có tính chất rất quan trọng. Nó giúp chúng ta có thể thu thập dữ liệu, kiểm tra cảm biến, quan sát quá trình vận hành của xe trong môi trường thực tế.

3.5. Chương trình trên raspberry pi giúp thu thập dữ liệu từ camera

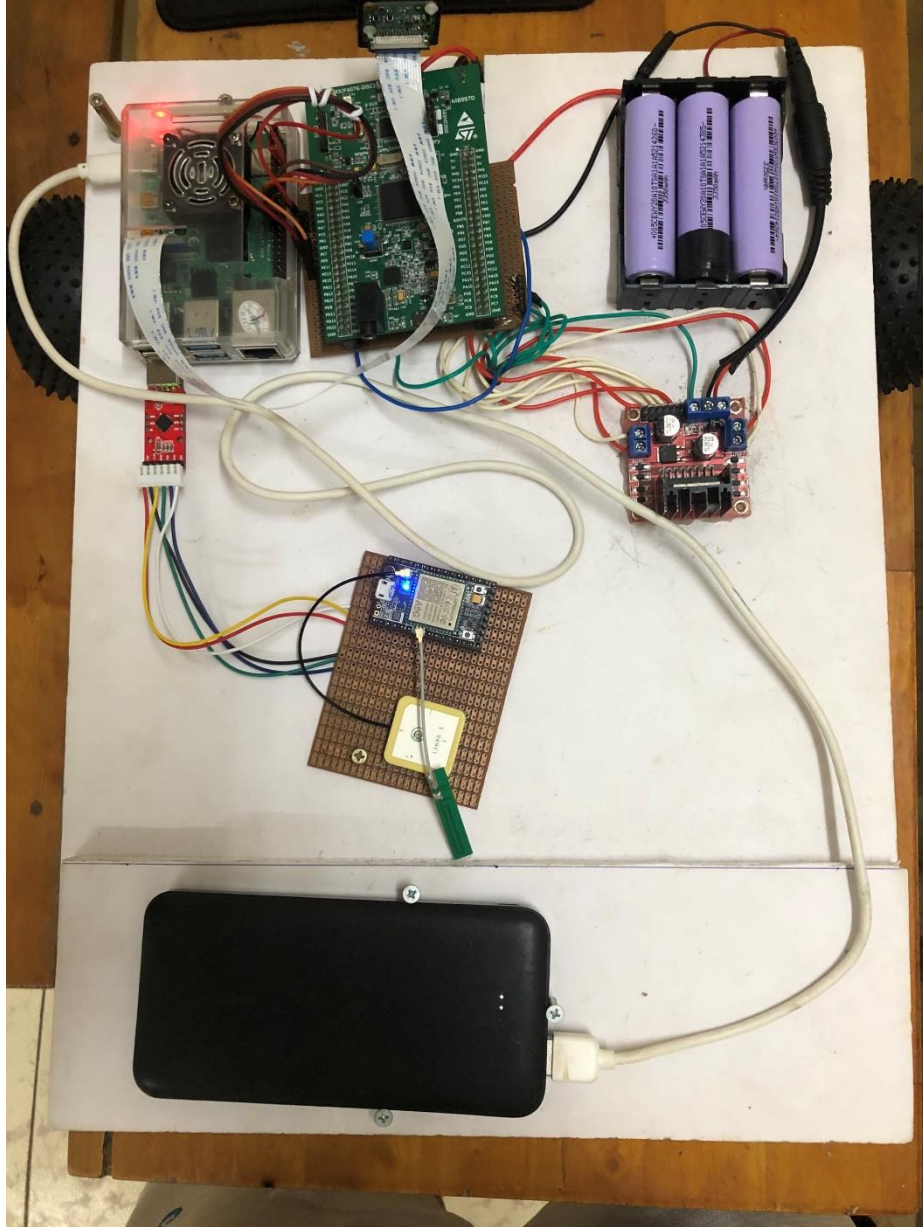
```
import picamera

camera = picamera.PiCamera()
camera.resolution = (1280,854 )
camera.start_recording('my_video.h264')
camera.wait_recording(500)
camera.stop_recording()
```

Giải thích: Đoạn chương trình trên nhằm mục đích thu thập dữ liệu dưới dạng video. Trong video mỗi khung ảnh có độ phân giải là 1280x854. Và mỗi lần chạy đoạn chương trình trên thì ta sẽ thu được một đoạn video có độ dài là 500 giây. Khi ta sử dụng đoạn chương trình này kết hợp với việc ta cho xe chạy ngoài thực tế với sự giám sát của con người thì ta sẽ thu được dữ liệu mong muốn.

CHƯƠNG 4. SẢN PHẨM VÀ KẾT QUẢ

4.1. Sản phẩm



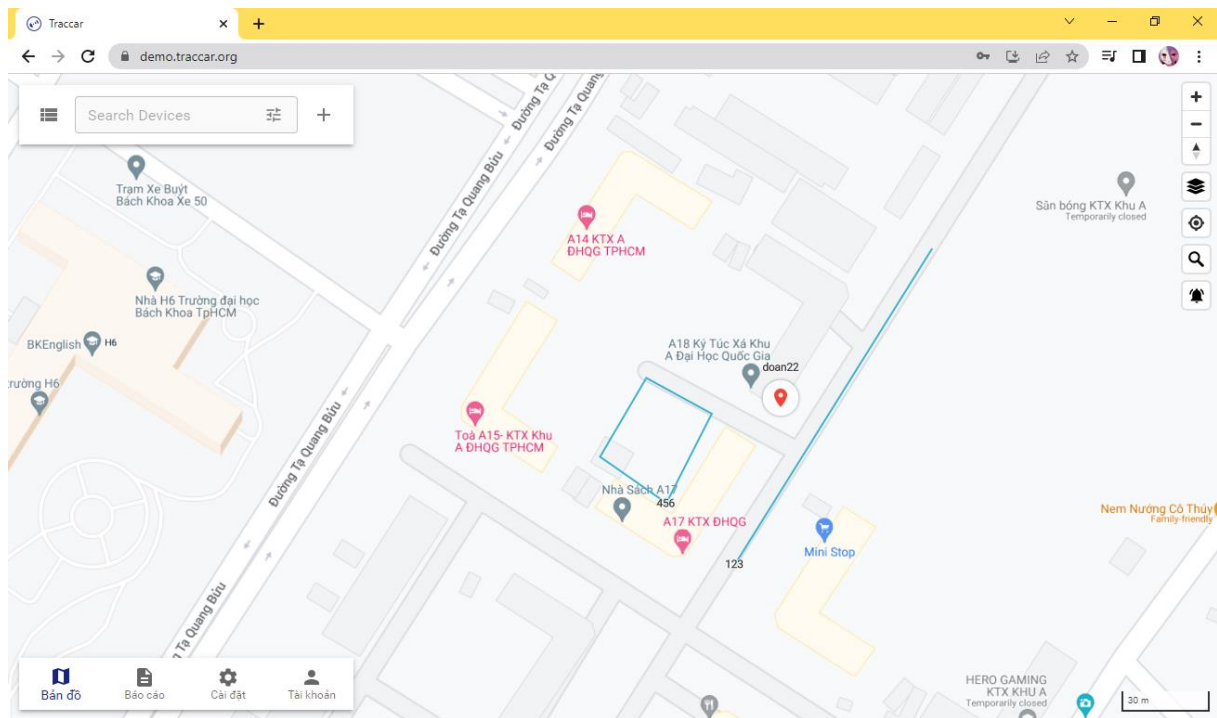
Hình 24. Hình ảnh xe nhìn từ trên



Hình 25. Hình ảnh xe nhìn từ bên hông

4.2. Kết quả

Nhóm em cho xe chạy thử trong khuôn viên ký túc xá, 2 đoạn đường đi:



Hình 26. Đoạn đường xe chạy trên Traccar

Kết quả thu thập dữ liệu từ môi trường thực tế bằng camera của xe:



Hình 27. Góc nhìn của xe ở lề đường



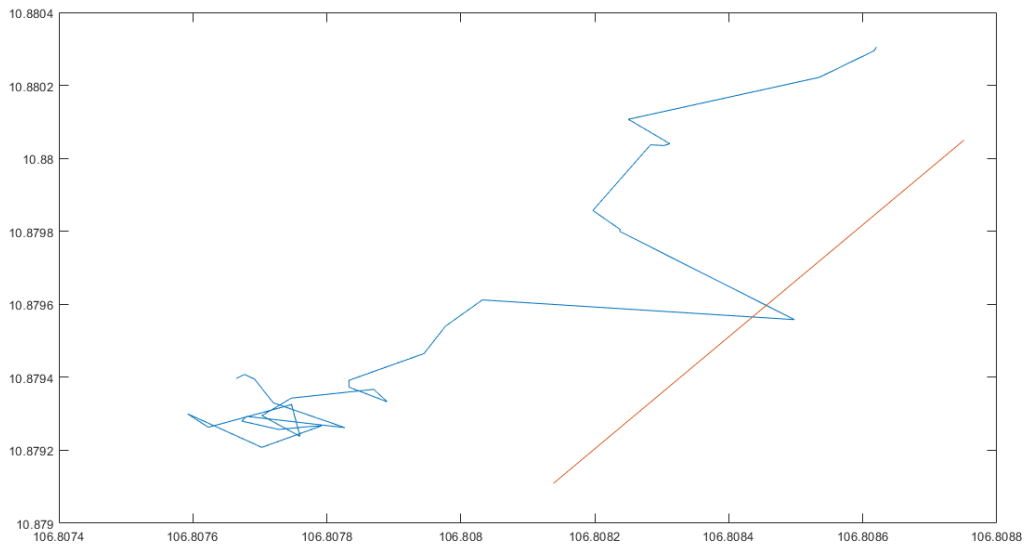
Hình 28. Góc nhìn của xe khi chạy trên đường

Kết quả thu được khoảng 7s gửi lên server Traccar một lần

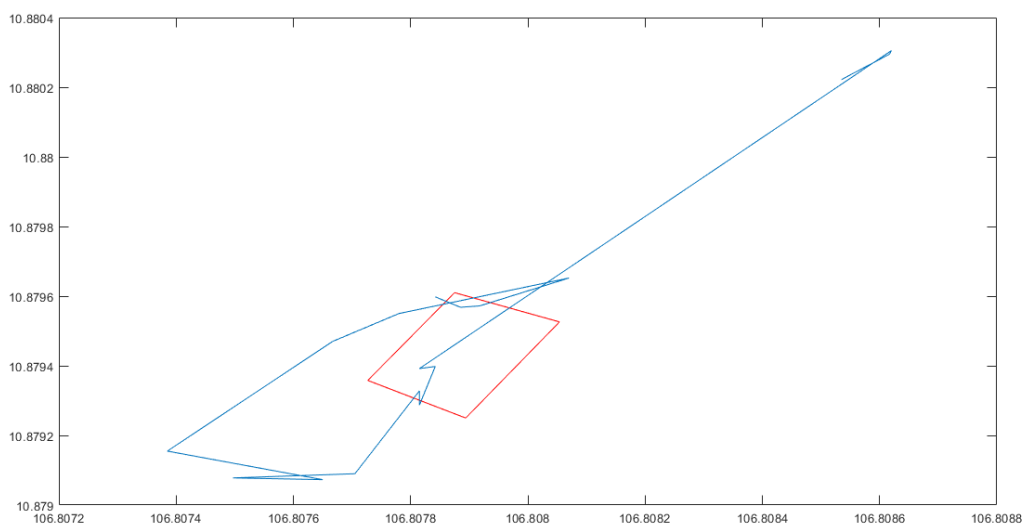
Valid	Time	Latitude	Longitude	Altitude	Speed
TRUE	2022-12-09 02:38:54	10.880535	106.808632	0 m	3.0 kn
TRUE	2022-12-09 02:40:20	10.880843	106.808853	0 m	3.0 kn
TRUE	2022-12-09 02:40:27	10.880738	106.808818	0 m	3.0 kn
TRUE	2022-12-09 02:40:34	10.880738	106.808818	0 m	3.0 kn

Bảng 7. Kết quả gửi lên server

Từ kết quả thu được nhóm em vẽ đồ thị lên MATLAB để so sánh, màu đỏ là đường đi thực tế của xe, còn màu xanh là đường đi vẽ lên từ dữ liệu GPS.



Hình 29. So sánh kết quả với thực tế



Hình 30. So sánh kết quả với thực tế

Nhận xét:

- Với việc sử dụng camera với góc thu ảnh là 60 độ thì góc nhìn của xe theo trực quan là khá hẹp. Tuy nhiên việc đánh giá góc nhìn của xe và chất lượng hình ảnh có đủ để thực hiện các giải thuật xử lý ảnh để điều khiển xe bám đường và tránh vật cản được hay không, thì trong thời gian tiếp theo nhóm sẽ thực hiện các giải thuật xử lý ảnh và đánh giá khách quan hơn. Hiện tại nhóm chỉ mới dừng ở việc

thu thập dữ liệu để làm tiền đề cho việc xử lí ảnh và điều khiển trong thời gian tiếp theo.

- Kết quả thu được từ GPS quá sai lệch so với đường đi thực tế, điều này nguyên nhân là do Module GPS không ổn định trong lúc nhận kết quả gửi về.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

5.1. Kết luận

Hiện tại, nhóm em chưa hoàn thành mục tiêu đề ra. Chỉ mới điều khiển được xe chạy, nhưng qua trình thu thập và xử lý dữ liệu từ GPS có nhiều lỗi, nguyên nhân chính là do dữ liệu thu thập từ GPS không ổn định, chưa chính xác nên nhóm em chưa điều khiển được.

5.2. Hướng phát triển đề tài

Sau khi hoàn thành xong sản phẩm ở mức độ đề án 2 này nhóm em định:

- Đối với phần cứng: hoàn thiện hoàn chỉnh quá trình lắp ráp xe đảm bảo tính thẩm mỹ, và độ ổn định. Sử dụng Module GPS khác để đảm bảo giá trị vị trí thu được chính xác hơn.
- Đối với phần điều khiển động cơ nhóm em dự định xây dựng bộ điều khiển tự chỉnh MRAC. Nhằm mục đích giúp phần điều khiển động cơ thích nghi tốt với thay đổi từ bên ngoài cũng như bám sát với tốc độ đặt hơn.
- Đối với phần GPS: Kết hợp thêm thuật toán di chuyển của xe giữa 2 điểm. Xây dựng giao diện để khách hàng có thể đặt xe.
- Đối với xử lý ảnh: nhóm em định sử dụng xử lý ảnh để phát hiện vật cản, làn đường để giúp cho xe có khả năng tự vận hành tốt hơn.

TÀI LIỆU THAM KHẢO

1. Long Phan. (10/12/2021). *Raspberry Pi 4 là gì? sơ đồ chân, tính năng và ngoại vi của*. Truy cập từ: <https://blog.mecsu.vn/raspberry-pi-4-la-gi/>
2. *Autonomous Delivery Vehicles revamping global logistics for last-mile delivery*. Truy cập từ: <https://aiworldschool.com/research/autonomous-delivery-vehicles-revamping-global-logistics-for-last-mile-delivery/>
3. [Online] https://www.makerfabs.com/desfile/files/a9g_product_specification.pdf
4. [Online] Truy cập từ:
https://ai-thinker-open.github.io/GPRS_C_SDK_DOC/en/hardware/pudding-dev-board.html
5. *NMEA Solution Output Format*. Truy cập từ:
<https://anavs.com/knowledgebase/nmea-format/>