**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



# BỘ MÔN

HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG Ô TÔ

ĐỀ TÀI

ĐIỀU KHIỂN XE DÒ LINE

|  |  |
| --- | --- |
| **SVTH** | **MSSV** |
| **1. PHAN HUY NAM** | **20145553** |
| **2. LÊ VĂN HƯNG** | **20145523** |
| **3. VÕ NHÂN BẢO** | **20145047** |
| **4. LƯƠNG PHÚ QUÝ** | **20145593** |
| **5. NGUYỄN VĂN VŨ**  **6. NGUYỄN XUÂN HUY** | **20145661**  **20145520** |
| **Ngành: CÔNG NGHỆ KỸ THUẬT Ô TÔ** |  |
| **Mã môn học: VACS330333\_ 01CLC\_T789** |  |
| **GVHD: TS. LÊ THANH PHÚC** |  |

Tp. Hồ Chí Minh, tháng 5 năm 2023

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



# BỘ MÔN

HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG Ô TÔ

ĐỀ TÀI

ĐIỀU KHIỂN XE DÒ LINE

|  |  |
| --- | --- |
| **SVTH** | **MSSV** |
| **1. PHAN HUY NAM** | **20145553** |
| **2. LÊ VĂN HƯNG** | **20145523** |
| **3. VÕ NHÂN BẢO** | **20145047** |
| **4. LƯƠNG PHÚ QUÝ** | **20145593** |
| **5. NGUYỄN VĂN VŨ**  **6. NGUYỄN XUÂN HUY** | **20145661**  **20145520** |
| **Ngành: CÔNG NGHỆ KỸ THUẬT Ô TÔ** |  |
| **Mã môn học: VACS330333\_ 01CLC\_T789** |  |
| **GVHD: TS. LÊ THANH PHÚC** |  |

Tp. Hồ Chí Minh, tháng 5 năm 2023

|  |  |
| --- | --- |
| LOGO CLC.JPG | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – Tự do – Hạnh phúc  \*\*\* |

Tp. Hồ Chí Minh, tháng 06 năm 2023

# NHIỆM VỤ BÁO CÁO

|  |  |
| --- | --- |
| Họ và tên sinh viên: | MSSV: |
| 1. Phan Huy Nam | 20145553 |
| 2. Lê Văn Hưng | 20145523 |
| 3. Võ Nhân Bảo | 20145047 |
| 1. Lương Phú Quý 2. Nguyễn Văn Vũ 3. Nguyễn Xuân Huy | 20145593  20145661  20145520 |
| Ngành: Công nghệ kỹ thuật Ô tô | Lớp: 20145CL1 |
| Giảng viên hướng dẫn: TS. Lê Thanh Phúc | |
| Ngày nhận đề tài: 15/03/2022 | Ngày nộp đề tài: |
| 1. Tên đề tài: Điều khiển xe dò Line |  |
| 2. Các số liệu, tài liệu ban đầu: |  |
| 3. Nội dung thực hiện đề tài: Xây dựng nên mô hình xe điều khiển và áp dụng các kiến thức đã học về Aruino, Python để điều khiển xe dò theo Line đường đã được thiết lập sẵn | |
| 4. Sản phẩm: Mô hình xe dò Line |  |
| TRƯỞNG NGÀNH | GIẢNG VIÊN HƯỚNG DẪN |

|  |  |
| --- | --- |
| LOGO CLC.JPG | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*\*** |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |  |
| --- | --- |
| Họ và tên sinh viên: | MSSV: |
| 1. Phan Huy Nam | 20145553 |
| 2. Lê Văn Hưng | 20145523 |
| 3. Võ Nhân Bảo | 20145047 |
| 1. Lương Phú Quý 2. Nguyễn Văn Vũ 3. Nguyễn Xuân Huy | 20145593  20145661  20145520 |

Ngành: Công nghệ kỹ thuật Ô tô Tên đề tài: Điều khiển xe dò Line

Họ và tên Giáo viên hướng dẫn: TS. Lê Thanh Phúc

## NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.......................................................................................................................................

1. Ưu điểm:

.......................................................................................................................................

.......................................................................................................................................

1. Khuyết điểm:

.......................................................................................................................................

.......................................................................................................................................

1. Đề nghị cho báo cáo hay không?

.......................................................................................................................................

1. Đánh giá phân loại:

.......................................................................................................................................

6. Điểm: ......................................(Bằng chữ )

Tp*. Hồ Chí Minh, ngày tháng năm 2023*

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

LỜI CẢM ƠN

*Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành đến Trường Đại học Sư phạm Kỹ thuật Tp.HCM đã đưa môn học Hệ thống điều khiển tự động ô tô vào chương trình giảng dạy.*

*Đặc biệt, nhóm em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn thầy TS. Lê Thanh Phúc đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập. Trong thời gian tham gia lớp học Hệ thống điều khiển tự động ô tô của thầy, nhóm em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để nhóm em có thể vững bước sau này.*

*Hệ thống điều khiển tự động ô tô là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của nhóm được hoàn thiện hơn.*

*Xin kính chúc quý thầy cô mạnh khoẻ, hạnh phúc và thành công trong sự nghiệp trồng người vinh quang.*

*Nhóm em xin chân thành cảm ơn!*

*TPHCM, ngày 04 tháng 06 năm 2023*

### Nhóm sinh viên thực hiện

*Tập thể nhóm*

# MỤC LỤC

Trang bìa phụ TRANG

Trang phiếu nhận xét của giảng viên hướng dẫn ..........................................................

Trang phiếu nhận xét của giảng viên phản biện............................................................

Lời cảm ơn i

[Mục lục ii](#_TOC_250003)

[Danh mục các từ viết tắt iii](#_TOC_250002)

Danh sách các hình, biểu đồ v

[Bảng phân công nhiệm vụ vi](#_TOC_250001)

[Chương 1](#_bookmark0)

[MỞ ĐẦU 1](#_bookmark1)

* 1. [Lý do chọn đề tài 1](#_bookmark2)
  2. [Mục tiêu, nhiệm vụ của đề tài 2](#_bookmark3)
     1. [Mục tiêu của đề tài 2](#_bookmark4)
     2. [Nhiệm vụ của đề tài 2](#_bookmark5)
  3. [Đối tượng, phạm vi nghiên cứu đề tài 2](#_bookmark6)
     1. [Đối tượng nghiên cứu 2](#_bookmark7)
     2. [Phạm vi nghiên cứu 2](#_bookmark8)
  4. [Phương pháp nghiên cứu 2](#_bookmark9)

[Chương 2](#_bookmark10)

[CƠ SỞ LÝ THUYẾT 4](#_bookmark11)

* 1. [Arduino 4](#_bookmark12)
     1. [Chớp LED 4](#_bookmark13)
     2. [Thu thập tín hiệu 6](#_bookmark18)
     3. [Xuất tín hiệu PWM 9](#_bookmark26)
     4. [Thu thập tín hiệu từ cảm biến qua đó điều chế xung PWM thay đổi tốc độ động cơ 112](#_bookmark31)
  2. [Python 14](#_bookmark36)
     1. [Python Shell 14](#_bookmark37)
     2. [Package cho Python Shell 15](#_bookmark40)
     3. [Serial communication giữa laptop và Arduino sử dụng Python 16](#_bookmark43)
        1. [Code Python Shell 16](#_bookmark44)
        2. [Code Arduino 17](#_bookmark46)
     4. [Xử lý ảnh bằng Python 19](#_bookmark50)
        1. [Code Python 19](#_bookmark51)
        2. [Code Arduino 21](#_bookmark53)
     5. [Xử lý ảnh bằng Open CV trong Python 26](#_bookmark64)
        1. [Code Python nhận diện đoạn thẳng 26](#_bookmark65)
        2. [Kết quả 28](#_bookmark67)

[Chương 3](#_bookmark69)

[TÍNH TOÁN, THIẾT KẾ VÀ LẮP ĐẶT 29](#_bookmark70)

* 1. [Chuẩn bị 29](#_bookmark71)
  2. [Lắp đặt 31](#_bookmark76)
  3. [Nối dây 31](#_bookmark77)
  4. [Tiến độ 32](#_bookmark80)

[Chương 4](#_bookmark81)

[GIAO DIỆN CODE XỬ LÝ VÀ ĐIỀU KHIỂN,](#_bookmark82) [CHẠY THỬ NGHIỆM 33](#_bookmark83)

* 1. [Giao diện code điều khiển Arduino 33](#_bookmark84)
     1. [Chi tiết từng đoạn lệnh và câu lệnh trong giao diện code Arduino 33](#_bookmark85)
     2. [Phần code hoàn chỉnh 35](#_bookmark86)
  2. [Giao diện code Python 37](#_bookmark87)
     1. [Chi tiết từng đoạn lệnh và câu lệnh trong giao diện code Python 37](#_bookmark88)
     2. [Phần code hoàn chỉnh 41](#_bookmark93)
  3. [Kết quả dò Line từ Python 43](#_bookmark94)

[Chương 5](#_bookmark98)

[KẾT LUẬN VÀ KIẾN NGHỊ 45](#_bookmark99)

* 1. [Kết luận 45](#_bookmark100)
     1. [Mục tiêu và kiến thức đạt được 45](#_bookmark101)
     2. [Thuận lợi trong quá trình làm project 45](#_bookmark102)
     3. [Khó khăn trong quá trình làm project 46](#_bookmark103)
  2. [Kiến nghị 46](#_bookmark104)

[TÀI LIỆU THAM KHẢO 47](#_TOC_250000)

# DANH MỤC CÁC TỪ VIẾT TẮT

**Arduino IDE**: Arduino Intergrated Development Environment

**COM**: Communication

**GND**: Ground

**LED**: Light Emitting Diode **PWM**: Pulse-Width Modulation **USB**: Universal Serial Bus

**ECT:** Electronic Control Transmission

**CCS:** Cruise Control System **ABS:** Anti – Lock Brake System **LAS:** Lane-keeping Assist

# DANH MỤC CÁC HÌNH, BIỂU ĐỒ

[**Hình 2.1.** Mở code chớp LED 4](#_bookmark14)

[**Hình 2.2.** Giao diện của code chớp LED 5](#_bookmark15)

[**Hình 2.3.** Nạp code chớp LED xuống Arduino 5](#_bookmark16)

[**Hình 2.4.** Hình ảnh thực tế 6](#_bookmark17)

[**Hình 2.5.** Mở chương trình AnalogRead 6](#_bookmark19)

[**Hình 2.6.** Giao diện AnalogRead 7](#_bookmark20)

[**Hình 2.7.** Cách set up 7](#_bookmark21)

[**Hình 2.8.** Giao diện khi nạp 8](#_bookmark22)

[**Hình 2.9.** Serial Monitor 8](#_bookmark23)

**Hình 2.10.** Serial Plotter 8

[**Hình 2.11.** Tín hiệu Serial Monitor 9](#_bookmark24)

[**Hình 2.12.** Tín hiệu Serial Plotter 9](#_bookmark25)

[**Hình 2.13.** Mở code Fade 10](#_bookmark27)

[**Hình 2.14.** Cách bố trí 10](#_bookmark28)

[**Hình 2.15.** Ảnh đang nạp 11](#_bookmark29)

[**Hình 2.16.** Ảnh thực tế sau khi nạp code 11](#_bookmark30)

[**Hình 2.17.** Code để chạy mô phỏng 12](#_bookmark32)

[**Hình 2.18.** Bố trí trên Proteus 12](#_bookmark33)

[**Hình 2.19.** Lấy file hex để nạp vào Proteus 13](#_bookmark34)

[**Hình 2.20.** Kết quả sau khi đưa file hex vào 14](#_bookmark35)

[**Hình 2.21.** Cách mở Python Shell 15](#_bookmark38)

[**Hình 2.22.** Giao diện chương trình Python Shell 15](#_bookmark39)

[**Hình 2.23.** Package Serial Communication 16](#_bookmark41)

[**Hình 2.24.** Pillow Package và OpenCV Package 16](#_bookmark42)

[**Hình 2.25.** Nạp code Python 17](#_bookmark45)

[**Hình 2.26.** Nạp code Arduino 18](#_bookmark47)

[**Hình 2.27.** Kết quả thực tế 18](#_bookmark48)

[**Hình 2.28.** Kết quả từ chương trình Python Shell 19](#_bookmark49)

[**Hình 2.29.** Nạp code Python 21](#_bookmark52)

[**Hình 2.30.** Nạp code Arduino 22](#_bookmark54)

[**Hình 2.31.** Ảnh màu đỏ chụp từ camera laptop 22](#_bookmark55)

[**Hình 2.31.** Ảnh kết quả màu đỏ sáng mạnh ra Arduino 23](#_bookmark56)

[**Hình 2.32.** Ảnh kết quả màu đỏ quả từ Python 23](#_bookmark57)

[**Hình 2.33.** Ảnh màu xanh lá cây chụp từ camera laptop 24](#_bookmark58)

[**Hình 2.34.** Ảnh kết quả màu xanh lá cây sáng mờ ra Arduino 24](#_bookmark59)

[**Hình 2.35.** Ảnh kết quả màu xanh lá cây từ Python 25](#_bookmark60)

[**Hình 2.36.** Ảnh màu xanh dương chụp từ camera laptop 25](#_bookmark61)

[**Hình 2.37.** Ảnh kết quả màu xanh dương ra Arduino 25](#_bookmark62)

[**Hình 2.38**. Ảnh kết quả màu xanh dương từ Python 26](#_bookmark63)

[**Hình 2.39.** Code Python nhận diện đoạn thẳng 27](#_bookmark66)

[**Hình 2.40.** Kết quả thu được từ chương trinh Python 28](#_bookmark68)

[**Hình 3.1.** Kìm điện, tua vít, kìm cắt và súng bắn keo 29](#_bookmark72)

[**Hình 3.2.** Bộ khung xe, băng keo điện, pin 4,2V và khay đựng 30](#_bookmark73)

[**Hình 3.3.** Dây điện, khây đựng pin 30](#_bookmark74)

[**Hình 3.4.** Cáp đực cái 20cm và Arduino Uno, L298 31](#_bookmark75)

[**Hình 3.5.** Sơ đồ mô phỏng nối dây bằng protues 31](#_bookmark78)

[**Hình 3.6.** Chân port trong arduino 32](#_bookmark79)

[**Hình 4.1.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được](#_bookmark89) [khoảng hoành độ giao điểm hợp lí với trường hợp đi xe đi thẳng 40](#_bookmark89)

[...................................................................................................................................40](#_bookmark90)

[**Hình 4.2.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được](#_bookmark91) [khoảng hoành độ giao điểm hợp lí với trường hợp đi xe rẽ phải 40](#_bookmark91)

[**Hình 4.3.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được](#_bookmark92) [khoảng hoành độ giao điểm hợp lí với trường hợp đi xe rẽ trái 41](#_bookmark92)

[**Hình 4.4.**. Hình ảnh dò Line đường đi thẳng 43](#_bookmark95)

[**Hình 4.5.** Hình ảnh dò Line rẽ trái 44](#_bookmark96)

[**Hình 4.6.** Hình ảnh dò Line rẽ phải 44](#_bookmark97)

# BẢNG PHÂN CÔNG NHIỆM VỤ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **MSSV** | **Nhiệm vụ phân công** | **Mức độ** |
| **1** |  |  | Thiết kế, lắp ráp khung xe, Word làm Chương 3, Power point phần 3, lên  lịch họp, phân công | 100%  (Tham gia tích cực) |
| **2** |  |  | Làm word phần Chương  5, Power point Chương 5, Đóng góp xây dựng code | 100%  (Tham gia tích cực) |
| **3** |  |  | Làm word Chương 2, power point Chương 2,  Đóng góp xây dựng code. | 100%  (Tham gia tích cực) |
| **4** |  |  | Làm Word Chương 1,4, làm power point chương 1, 4, Lời cảm ơn, Tổng hợp, chỉnh sửa word và  power point | 100%  (Tham gia tích cực) |
| **5** |  |  | Nghiên cứu và làm Code Python, Word Chương 4. Giải thích code cho các  thành viên còn lại. | 100%  (Tham gia tích cực) |

**Note:** Mọi người trong nhóm đều có thế mạnh sở trường riêng nên việc phân công và làm báo cáo được dựa trên niềm đam mê và khả năng của từng bạn.

Kí tên Trưởng nhóm

**CHƯƠNG 1:**

**MỞ ĐẦU**

## Lý do chọn đề tài

Thế giới công nghệ ngày nay vẫn không ngừng phát triển, và len lỏi tới toàn bộ đời sống chúng ta. Các ngành học về công nghệ phần mềm cũng hết sức đa dạng, từ tầm vĩ mô như sáng tạo ra những ứng dụng để đời cho tới những điều cần thiết cho cuộc sống như điều khiển hệ thống. Chính vì vậy ứng dụng lập trình Python vào Arduino điều khiển có tính thực tế rất cao trong con đường phát triển công nghệ.

## Mục tiêu, nhiệm vụ của đề tài

## Mục tiêu của đề tài

Sinh viên có thể hiểu, nắm bắt các kiến thức cơ bản về các khái niệm, cấu tạo và nguyên lý hoạt động của các hệ thống điều khiển tự động trên ô tô, đặc biệt là hệ thống Lane-keeping Assist.

Dựa trên các kiến thức cơ sở đó, cùng với các tài liệu trên Internet sẵn có để có thể vận dụng và xây dựng nên mô hình xe dò line.

Cuối cùng là chạy thử nghiệm và rút ra được bài học dựa trên xe mô hình của mình, từ đó có thể phát triển thêm thành mô hình xe chạy tự động.

## Nhiệm vụ của đề tài

Đưa ra hình ảnh, các dụng cụ, thiết bị và sự đánh giá khách quan về việc xây dựng nên mô hình xe.

Nghiên cứu cấu tạo, nguyên lý vận hành và sau đó thiết kế để lắp đặt hoàn chỉnh mô hình xe.

Nắm được các kiến thức cơ bản về lập trình Arduino và Python để có thể mô phỏng được xe dò line.

## Đối tượng, phạm vi nghiên cứu đề tài

## Đối tượng nghiên cứu

Ý tưởng chủ yếu dựa trên hệ thống hỗ trợ giữ làn đường (Lane-keeping Assist System – LAS) và theo hướng dẫn từ các tài liệu có sẵn và thông tin trên Internet, để định hướng và xây dựng nên mô hình.

## Phạm vi nghiên cứu

Trong phạm vi hệ thống điều khiển tự động trên ô tô và các hệ thống khác có liên quan

Dữ liệu của bài báo cáo chủ yếu dựa trên bài hướng dẫn Arduino và Python và các thông tin từ nguồn Internet và các tài liệu trong giáo trình.

## Phương pháp nghiên cứu

Trong quá trình nghiên cứu thực hiện đề tài nhóm chúng em có sử dụng một số phương pháp nghiên cứu sau:

* Tra cứu trong các tài liệu, giáo trình kỹ thuật, sách vở,vv
* Tìm kiếm thông tin trên mạng Internet, các website trong và ngoài nước. So sánh và chắt lọc để sử dụng những thông tin cần thiết và đáng tin cậy.
* Tham khảo ý kiến của các giảng viên trong khoa cơ khí động lực và các thầy cô trong khoa Đào tạo chất lượng cao.
* Tổng hợp và phân tích các nguồn dữ liệu thu thập được, từ đó đưa ra những đánh giá và nhận xét của riêng mình.

**CHƯƠNG 2:**

**CƠ SỞ LÝ THUYẾT**

## Arduino

Arduino là một nền tảng mã nguồn mở được sử dụng để xây dựng các dự án điện tử. Arduino bao gồm cả phần cứng bảng mạch (thường được gọi là vi điều khiển) và một phần mềm IDE (Môi trường phát triển tích hợp) chạy trên máy tính của bạn, được sử dụng để viết code và nạp cho phần cứng.

Ngày nay arduino được sử dụng khá phổ biến với những người mới bắt đầu với điện tử, một phần bởi vì mạch arduino dễ mua được trên thị trường. Ngoài ra, Arduino IDE sử dụng phiên bản C ++ đơn giản hóa, giúp việc học lập trình trở nên dễ dàng hơn.

## 2.1.1 Chớp Led

Chuẩn bị: Arduino Uno

Tiến trình thực hiện

Mở code chớp LED.Để mở code chớp LED, ta làm theo như sau.

Vào file – examples – 01.basics – blink

Giao diện mở ra như sau

Graphical user interface, text, application

Description automatically generated

*Hình 2.1. Giao diện của code chớp LED*

Chúng ta nạp code cho arduino và kết quả thu được

A circuit board with a red light

Description automatically generated with low confidenceA close up of a circuit board

Description automatically generated with medium confidence

*Hình 2.2. Hình ảnh thực tế*

**2.1.2. Thu thập tín hiệu**

Chuẩn bị: biến trở 5V

A picture containing person, holding, hand

Description automatically generated

*Hình 2.3 Hình biến trở*

* Mở chương trình đọc tín hiệu đầu vào ta làm như sau:

Vào file – examples – 01.basics – analogreadserial

Giao diện mở ra như sau:

Graphical user interface, text, application, email

Description automatically generated

*Hình 2.4. Giao diện AnalogRead*

* Cách bố trí

Hai đầu ngoài cùng của biến trở, 1 đầu nối lên +5V, 1 đầu nối GND của Arduino. Đây là 2 đầu cấp nguồn cho biến trở. Đầu giữa nối đến chân A0 của Arduino.

A picture containing electronics, electrical wiring, cable, electronic engineering

Description automatically generated

*Hình 2.5. Cách setup*

* Tiến hành nạp code và xem kết quả

A screenshot of a computer

Description automatically generated with low confidenceA picture containing table

Description automatically generatedTable

Description automatically generated

*Hình 2.6 Kết quả xoay biến trở*

**2.1.3. Xuất tín hiệu PWM**

* Chuẩn bị LED 3V và điện trở
* Mở chương trình Fade có trên Arduino IDE

Vào file – examples – 01.basics – fade

Giao diện mở ra như sau:

Graphical user interface, text, application, email

Description automatically generated

*Hình 2.7. Mở code Fade*

* Bố trính như hình

A close up of a circuit board

Description automatically generated with low confidence

*Hình 2.8 Cách bố trí*

* Tiến hành nạp code và xem kết quả

Đèn sáng tỏ rồi mờ dần

A picture containing text

Description automatically generatedA circuit board with red and green lights

Description automatically generated with low confidence

*Hình 2.9. Ảnh thực tế sau khi nạp code*

**2.1.4 Kết luận**

Qua 3 ví dụ cơ bản trên nhóm chúng em đã phần nào hiểu rõ về cách lập trình điều khiển arduino UNO để điều khiển các thiết bị phần cứng. Ngoài ra nhóm chúng em còn hiểu thêm về hoạt động của biến trở và tín hiệu xung PWM để điều độ sáng LED hay tốc độ motor.

**2.2. Python**

Python là ngôn ngữ lập trình hướng đối tượng đơn giản, dễ học, mạnh mẽ, cấp cao. Python có cấu trúc cú pháp ít hơn các ngôn ngữ khác.

Python được thông dịch: Python được trình thông dịch xử lý trong thời gian chạy. Bạn không cần phải biên dịch chương trình của mình trước khi thực hiện nó. Nó tương tự với PERL và PHP.

Python là tương tác (Interactive): Tại một dấu nhắc Python (command line) bạn có thể tương tác trực tiếp với trình thông dịch để viết chương trình Python.

Python là hướng đối tượng: Python hỗ trợ kỹ thuật lập trình hướng đối tượng hoặc kỹ thuật lập trình đóng gói mã trong các đối tượng.

Python là ngôn ngữ của người mới bắt đầu: Python là ngôn ngữ tuyệt vời cho các lập trình viên mới bắt đầu và hỗ trợ phát triển một loạt các ứng dụng từ xử lý văn bản đơn giản, lập trình web, cho đến lập trình game.

**2.2.1. Python Shell**

Python Shell là một công cụ đi kèm sau khi chúng ta cài đặt xong Python. Đây là một IDE (Integrated Development Environment) giúp soạn thảo các mã Python, có nhiều IDE khác giúp phát triển những thứ lớn hơn nhưng nếu chỉ đơn giản là để biết về ngôn ngữ này thì với Python Shell là đủ để sử dụng. Để mở Python Shell: ta vào Start, sreach từ khóa “Python” và click vào IDLE (Python 3.10 64-bit).

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 2.10. Cách mở Python Shell*

**2.2.2. Package cho Python Shell**

Do bài báo cáo có liên quan đến việc ứng dụng Python để xử lý ảnh nên phải cài thêm Package. Đây là các gói chương trình cho phép Python Shell thực hiện được nhiều lệnh và chức năng. Và như đã nói ở trên ngoài việc xử lý ảnh còn có: các hàm toán học, giao tiếp với thiết bị bên ngoài, ...

Việc cài đặt các Package được thực hiện trong cửa sổ Command Prompt của Window. Để mở cửa sổ này, ta có thể Search trên công cụ tìm kiếm của Window [1]

py -m pip install "SomeProject"

- Từ "SomeProject" ở đây là tên của Package đó.

Ví dụ: Pyserial, Opencv - Python

Và đối với bài báo cáo này, nhóm cần cài các Package sau:

**Package Serial Communication**

Việc đầu tiên là kết nối laptop và Arduino nhằm mục đích vận dụng khả năng tính toán mạnh của laptop và khả năng điều khiển thiết bị của Arduino, từ Command Prompt của Window chúng ta gõ lệnh như sau:

py -m pip install pyserial

A picture containing text, screenshot, font

Description automatically generated

*Hình 2.11.* *Package Serial Communication*

**Pillow Package, OpenCV Package**

Thư viện hình ảnh Python bổ sung khả năng xử lý hình ảnh cho trình thông dịch Python của bạn.

Thư viện này cung cấp hỗ trợ định dạng tệp mở rộng, biểu diễn nội bộ hiệu quả và khả năng xử lý hình ảnh khá mạnh mẽ.

Thư viện hình ảnh cốt lõi được thiết kế để truy cập nhanh vào dữ liệu được lưu trữ ở một vài định dạng pixel cơ bản. Nó sẽ cung cấp một nền tảng vững chắc cho một công cụ xử lý hình ảnh chung.

Từ Command Prompt của Window chúng ta gõ lệnh sau:

py -m pip install Pillow

py -m pip install opencv-python

A screenshot of a computer program

Description automatically generated with medium confidence

*Hình 2.12. Pillow Package và OpenCV Package*

**2.2.3. Serial communication giữa laptop và Arduino sử dụng Python**

**2.2.3.1. Code Python Shell**

Nhóm tiến hành nạp code vào chương trình Python Shell

Code Python Shell

import serial

import time

arduino = serial.Serial(port='COM5', baudrate=115200, timeout=0.1)

while True:

data = arduino.readline()

time.sleep(0.01)

arduino.write(data)

time.sleep(0.01)

data1= data.decode('utf-8')

print(data1)

A screenshot of a computer program

Description automatically generated with low confidence

*Hình 2.13. Nạp code Python*

**2.2.3.2. Code Arduino**

Sau khi nap code Python, nhóm sẽ tiếp tục nạp code Arduino để đọc tín hiệu analog từ 1 biến trở ở chân A0, truyền tín hiệu này lên Python và đọc tín hiệu truyền ngược từ laptop xuống chính tín hiệu analog này.

Code Arduino

int x;

void setup() {

Serial.begin(115200);

Serial.setTimeout(1);

pinMode(10,OUTPUT);

}

void loop() {

int sensorValue = analogRead(A0);

Serial.print(sensorValue);

Serial.print('\n');

x = Serial.readString().toInt();

int y=int(x/4);

analogWrite(10,y);

delay(100);

}

A screenshot of a computer

Description automatically generated

*Hình 2.14. Nạp code Arduino*

- Kết quả khi chạy chương trình Python, nhóm thấy được LED nối với chân 11 của Arduino sẽ sáng với độ sáng thay đổi theo giá trị xoay của biến trở.

A close-up of a circuit board

Description automatically generated with low confidence

*Hình 2.15. Kết quả thực tế*

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 2.16. Kết quả từ chương trình Python Shell*

**2.2.4. Xử lý ảnh bằng Python**

**2.2.4.1. Code Python**

* Nhóm tiến hành nạp Code Python

import cv2

from PIL import Image

from PIL.ImageStat import Stat

import serial

import time

cam = cv2.VideoCapture(0)

frame = cam.read()[1]

cv2.imwrite('img2.png', frame)

Hinh1 = Image.open("img2.png")

Hinh1.show()

hinh2 = Image.Image.split(Hinh1)

print(Hinh1.mode)

ThongKe = Stat(Hinh1)

print(ThongKe.mean)

MauDo=ThongKe.mean[0]

XanhLuc=ThongKe.mean[1]

XanhDuong=ThongKe.mean[2]

DanhSach=[MauDo, XanhLuc, XanhDuong]

max\_DanhSach=max(DanhSach)

max\_index = DanhSach.index(max\_DanhSach)

arduino = serial.Serial(port='COM5', baudrate=115200, timeout=0.1)

dem=0

while (dem<10):

dem=dem+1

if max\_index==0:

print('Mau Do')

arduino.write(b'1')

time.sleep(0.1)

if max\_index==1:

print('Xanh Luc')

arduino.write(b'2')

time.sleep(0.1)

if max\_index==2:

print('Xanh Duong')

arduino.write(b'3')

time.sleep(0.1)

arduino.close()

cam.release()

A screen shot of a computer program

Description automatically generated with low confidence

*Hình 2.17. Nạp code Python*

**2.2.4.2. Code Arduino**

**A screenshot of a computer program

Description automatically generated with medium confidence**

*Hình 2.18 Nạp code Arduino*

- Sau đó nhóm chuẩn bị 3 hình nền với 3 màu: Đỏ, xanh lá cây và xanh dương. Đưa 3 hình với 3 màu khác nhau này vào camera của laptop. Nhiệm vụ của chương trình Python là xác định được 3 màu này và truyền kết quả đến Arduino thông qua cổng giao tiếp nối tiếp.

- Sẽ có 3 trạng thái sáng của của Led khi nhận được tín hiệu:

• *Trạng thái 1: Led sáng mạnh khi có màu đỏ xuất hiện*

A picture containing text, display device, computer, output device

Description automatically generated

*Hình 2.19 .Ảnh màu đỏ chụp từ camera laptop*

*• Trạng thái 2: Led sáng mờ khi có màu xanh lá cây xuất hiện*

A computer screen with a green screen on it

Description automatically generated with low confidence

*Hình 2.20 Ảnh màu xanh lá cây chụp từ camera laptop*

*• Trạng thái 3: Led tắt khi có màu xanh dương xuất hiện*

A computer screen with a blue square on it

Description automatically generated with low confidence

*Hình 2.21 Ảnh màu xanh dương chụp từ camera laptop*

**2.2.5. Xử lý ảnh bằng Open CV trong Python**

**2.2.5.1. Code Python nhận diện đoạn thẳng**

*Bước 1:* Thêm thư viện cv2 và numpy bằng 2 dòng lệnh:

import cv2 import

numpy as np

*Bước 2:* Kích hoạt webcam của laptop và đặt tên cho cửa xổ hiển thị hình ảnh bằng 2 dòng lệnh:

cam = cv2.VideoCapture(0)

window\_name = 'Hinh anh tu Camera'

*Bước 3:* Tạo một vòng lặp bằng lệnh while

while(True):

*Bước 4:* Đọc hình ảnh từ webcam của laptop và gán vào biến cụ thể đồng thời nó sẽ trả về giá trị Đúng hoặc Sai để điều khiển vòng lặp

Hinh1 = cam.read()[1]

*Bước 5:* Chuyển đổi ảnh BGR sang ảnh xám và phát hiện cạnh của ảnh xám

gray = cv2.cvtColor(Hinh1, cv2.COLOR\_BGR2GRAY)

edges = cv2.Canny(gray, 50, 150, apertureSize=3)

*Bước 6:* Nhận diện đoạn thẳng ta dùng đoạn lệnh sau:

lines = cv2.HoughLines(edges, rho=1, theta=np.pi/180, threshold=100)

for line in lines:

rho, theta = line[0]

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

cv2.line(Hinh1,(x1,y1),(x2,y2),(255,255,125),2)

*Bước 7:* Hiển thị hình ảnh vừa mới đọc được từ webcam bằng lệnh

cv2.imshow(window\_name,Hinh1)

*Bước 8:* Sử dụng hàm chờ để nhận dữ liệu hình ảnh trực tiếp liên tục từ webcam của laptop

if cv2.waitKey(1) & 0xFF == ord('q'):

break

*Bước 9:* Giải phóng webcam và đóng tất cả cửa sổ của lệnh imshow

cam.release()

cv2.destroyAllWindows()

A screenshot of a computer program

Description automatically generated with medium confidence

*Hình.2.22 Code Python nhận diện đoạn thẳng*

### Kết quả

**A screen shot of a computer

Description automatically generated with low confidence**

*Hình 2.23. Kết quả thu được từ chương trinh Python*

## 

## Chuẩn bị

# Chương 3

**TÍNH TOÁN, THIẾT KẾ VÀ LẮP ĐẶT**

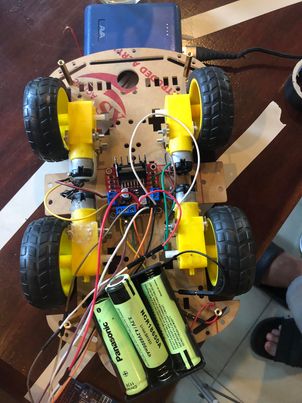
- *Trang bị*

Dụng cụ: Kìm (có sẵn), tua vít (có sẵn), kìm cắt dây điện (có sẵn), súng bắn keo (40k), hàn chì (có sẵn), dây chì.

Vật liệu: Bộ khung xe 4 bánh + L298 (105k), băng keo đen (có sẵn), 3 cục pin sạc 4.2V+ khay đựng pin + cục sạc (133K), cáp đực cái 20cm (18K), Arduino Uno (145K), dây diện đỏ đen (5K)



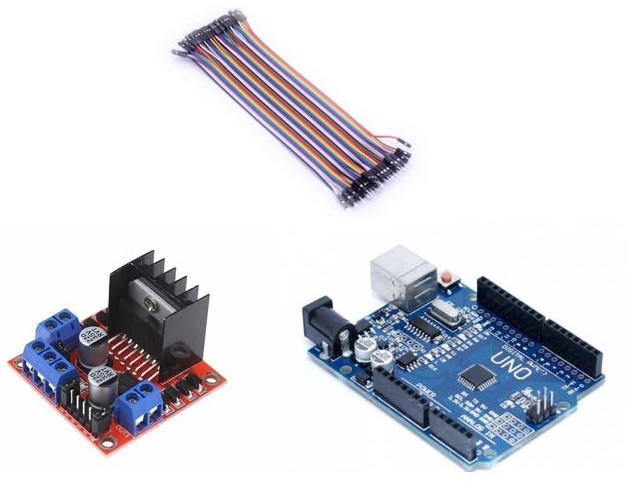
*Hình 3.1. tua vít, kìm cắt*



*Hình 3.2. Bộ khung xe, pin 4,2V và khay đựng*



*Hình 3.3. Dây điện, khây đựng pin*



*Hình 3.4. Cáp đực cái 20cm và Arduino Uno, L298*

## Lắp đặt

Có nhiều cách để bố trí lắp đặt các bộ phận, dưới đây là 1 trong nhiều cách. Đầu tiên là:

* + - Ráp bộ khung xe
    - Để cầu H L298 ở khung dưới cho mạch gọn và thẩm mỹ
    - Pin và mạch arduino để ở phần mặt trên
    - Pin để ở giữa thân xe để xe ổn định

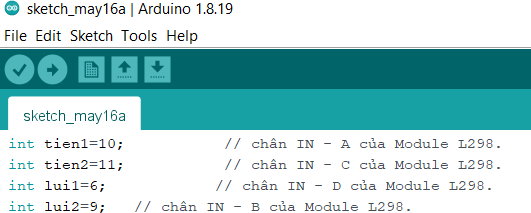
## Nối dây

*Hình 3.5. Sơ đồ mô phỏng nối dây bằng protues*

* + - Giải thích

Dựa vào chân Port trong Arduino để nối dây vào L298 hoặc ta có thể nối dây trước và chỉnh lại phần chân port trong arduino.

Chân Out1, Out2 nối vào động cơ DC bên phải, Out3, Out4 nối vào động cơ DC bên trái.



*Hình 3.6: Chân port trong arduino*

## Tiến độ

* + - Ngày 15/03/2023, bắt đầu lên kế hoạch tiến hành mua các trang thiết bị.
    - Ngày 27/03/2023, hội nhóm lên thư viện clc bắt đầu thiết kế mô hình.
    - Ngày 10/04/2023, nhóm làm xong code để điều khiển chuyển động của xe.
    - Ngày 21/04/2023, Hiệu chỉnh cho xem chạy nhanh bằng cách điều khiển biến

trở.

* + - Ngày 04/05/2022-06/05/2022 code python nhận diện đường thẳng, chưa có chạy

được.

-Ngày 13/05/2023 xe chạy hoàn chỉnh

# Chương 4

**GIAO DIỆN CODE XỬ LÝ VÀ ĐIỀU KHIỂN,** **CHẠY THỬ NGHIỆM**

## Giao diện code điều khiển Arduino

## Chi tiết từng đoạn lệnh và câu lệnh trong giao diện code Arduino

**Bước 1:** Gán 4 chân điều khiển motor của module điều khiển động cơ L298N vào 4 chân điều chế xung PWM của Arduino bằng đoạn lệnh sau:

int tien1=10; // chân IN - 1 của Module L298N. int tien2=11; // chân IN - 3 của Module L298N. int lui1=6; // chân IN - 4 của Module L298N.

int lui2=9; // chân IN - 2 của Module L298N.

**Bước 2:** Khai báo biến x dưới dạng số nguyên để nhận giá trị từ python trả về

int x;

**Bước 3:** Setup chế độ OUTPUT cho 4 chân điều khiển motor của module điều khiển động cơ L298N và để 4 chân này ở chế độ LOW tức là khi chạy code nếu không có thay đổi gì thì 4 chân này sẽ không có điện áp ta sử dụng đoạn lệnh sau. pinMode(tien1,OUTPUT);

pinMode(tien2,OUTPUT); pinMode(lui1,OUTPUT); pinMode(lui2,OUTPUT); digitalWrite(tien1,LOW); digitalWrite(tien2,LOW); digitalWrite(lui1,LOW); digitalWrite(lui1,LOW);

**Bước 4:** Khởi động một cổng Serial để giao tiếp với máy tính với mức baudrate 115200 và đặt thời gian tối đa cho việc xử lý dữ liệu Serial là 1ms Serial.begin(115200);

Serial.setTimeout(1);

**Bước 5:** Tạo 4 chương trình con thể hiện 4 trạng thái khác nhau của xe: Đi Thẳng, Rẽ Trái, Rẽ Phải, Đứng Im

void dithang()

{

analogWrite(tien1,50); analogWrite(lui1,0); analogWrite(tien2,50); analogWrite(lui2,0); delay(200);

}

void disangtrai()

{

resetdongco(); analogWrite(tien1,50); analogWrite(lui1,0); analogWrite(tien2,0); analogWrite(lui2,0); delay(910);

}

void disangphai()

{

resetdongco(); analogWrite(tien1,0); analogWrite(lui1,0); analogWrite(tien2,35); analogWrite(lui2,0); delay(910);

}

void resetdongco()

{

analogWrite(tien1,0); analogWrite(lui1,0); analogWrite(tien2,0); analogWrite(lui2,0); delay(2000);

}

**Bước 6:** Python trả về dạng chuỗi sau đó được chuyển về số nguyên và gán vào

biến x đã được khai báo ở Bước 2 và được đưa vào chương trình chính với câu lệnh sau :

x = Serial.readString().toInt();

**Bước 7:** Sử dụng lệnh điều kiện If để đưa ra các chương trình con ứng với vạch đường ( Đi thẳng, Rẽ trái, Rẽ phải) bằng đoạn lệnh sau:

if (x==1){ dithang();

}

if (x==2) { disangtrai();

}

if (x==3){ disangphai();

}

## Phần code hoàn chỉnh

int tien1=10; // chân IN - 1 của Module L298.

int tien2=11; // chân IN - 3 của Module L298.

int lui1=6; // chân IN - 4 của Module L298.

int lui2=9; // chân IN - 2 của Module L298. int x;

void dithang(); void disangtrai(); void disangphai(); void setup()

{

pinMode(tien1,OUTPUT); pinMode(tien2,OUTPUT); pinMode(lui1,OUTPUT); pinMode(lui2,OUTPUT); digitalWrite(tien1,LOW); digitalWrite(tien2,LOW); digitalWrite(lui1,LOW); digitalWrite(lui1,LOW); Serial.begin(115200); Serial.setTimeout(1);

}

void loop()

{

x = Serial.readString().toInt(); if (x==1){

dithang();

}

if (x==2) { disangtrai();

}

if (x==3){ disangphai();

}

}

void dithang()

{

analogWrite(tien1,50); analogWrite(lui1,0); analogWrite(tien2,50); analogWrite(lui2,0); delay(200);

}

void disangtrai()

{

resetdongco(); analogWrite(tien1,50); analogWrite(lui1,0); analogWrite(tien2,0); analogWrite(lui2,0); delay(910);

}

void disangphai()

{

resetdongco(); analogWrite(tien1,0); analogWrite(lui1,0); analogWrite(tien2,35); analogWrite(lui2,0); delay(910);

}

void resetdongco()

{

analogWrite(tien1,0); analogWrite(lui1,0); analogWrite(tien2,0); analogWrite(lui2,0);

delay(2000);

}

## Giao diện code Python

## Chi tiết từng đoạn lệnh và câu lệnh trong giao diện code Python

**Bước 1:** Khai báo 3 thư viện chính được sử dụng cho quá trình xử lý ảnh, giao tiếp với Arduino để điều khiển và tìm vạch kẻ đường của xe qua đoạn lệnh sau:

import cv2

import numpy as np import serial

**Bước 2:** Khởi tạo giao tiếp giữa python và arduino qua dòng lệnh sau:

arduino = serial.Serial(port='COM5', baudrate=115200, timeout=0.1)

**Bước 3:** Khởi tạo camera mặc định của laptop hoặc một camera bất kỳ khác được kết nối ta dùng lệnh sau:

cam = cv2.VideoCapture(0)

**Bước 4:** Sử dụng biến Hinh1 để nhận ảnh được trả về từ camera tiếp theo chuyển đổi từ ảnh BGR thành ảnh xám sau đó làm mịn và tìm phần biên của ảnh qua đoạn lệnh sau:

Hinh1 = cam.read()[1]

gray = cv2.cvtColor(Hinh1, cv2.COLOR\_BGR2GRAY) blur = cv2.GaussianBlur(gray,(5,5),0)

edges = cv2.Canny(blur, 50, 150, apertureSize=3)

**Bước 5:** Sử dụng chương trình con để tìm vạch kẻ đường bằng cách đưa toàn bộ bức ảnh về hoàn toàn màu đen tương ứng với số nhị phân 0 và tạo vùng nhận ảnh là hình tam giác tương ứng với kích thước 640x480 của camera sau đó đưa vào chương trình chính bằng dòng lệnh sau:

def region\_of\_interest(image): height=image.shape[0] polygons=np.array([

[(0,height),(640,height),(320,-200000)]

])

mask=np.zeros\_like(image) cv2.fillPoly(mask,polygons,255) masked\_image=cv2.bitwise\_and(image,mask)

return masked\_image

cropped\_image = region\_of\_interest(edges)

**Bước 6:** Sử dụng hai lệnh cv2.HougLinesP để tìm vạch kẻ đường và hiển thị ; cv2.HoughLines để tìm vạch kẻ đường và điều khiển trạng thái của xe (Đi Thẳng, Rẽ Phải, Rẽ Trái) , lý do phải tách ra hai lệnh riêng biệt đó là để tránh đường thẳng bị nhiễu khiến cho việc điều khiển trạng thái của xe cũng như hiển thị vạch kẻ đường thiếu chính xác điều này dẫn đến việc xử lý không được mượt và sẽ có sai số đáng kể, đôi khi là code sẽ bị lỗi vì các điều kiện bị mâu thuẫn với nhau .

lines=cv2.HoughLinesP(cropped\_image,2,np.pi/180,120,np.array([]),minLineLeng th=40,maxLineGap=1)

line\_control = cv2.HoughLines(edges, rho=1, theta=np.pi/180, threshold=100)

**Bước 7:** Tạo một chương trình con dùng để hiển thị vạch kẻ đường được xác định ở Bước 5 và đưa vào chương trình chính bằng câu lệnh sau

def display\_lines(image,lines): line\_image = np.zeros\_like(image) if lines is not None:

for line in lines:

x1,y1,x2,y2 = line.reshape(4) cv2.line(line\_image,(x1,y1),(x2,y2),(255,0,0),10)

return line\_image

line\_image = display\_lines(Hinh1,lines)

**Bước 8:** Chồng vạch kẻ đường được tìm thấy lên biến Hinh1 là biến đọc hình ảnh từ camera trả về và hiển thị lên màn hình laptop bằng đoạn lệnh sau

combo\_image= cv2.addWeighted(Hinh1,0.8,line\_image,1,1) cv2.imshow(window\_name,combo\_image)

**Bước 9:** Sử dụng vòng lặp với biến line\_control còn lại ở Bước 6 để điều khiển ở đây ta sử dụng 2 biến để xác định trạng thái xe chạy( Đi thẳng,Rẽ Trái,Rẽ Phải) biến đầu tiên là hoành độ giao điểm giữa đường thẳng y = 400 với đường thẳng bên trái và bên phải và hoành độ giao điểm có giá trị từ 0-640, biến thứ hai là góc theta là

góc giữa trục hoành và đoạn thẳng ngắn nhất được nối tới gốc tọa độ và theta có giá trị trong khoảng từ 0 -𝜋 , ta có đoạn lệnh sau:

for line\_c in line\_control: rho,theta = line\_c[0] a = np.cos(theta)

b = np.sin(theta) x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b)) y1 = int(y0 + 1000\*(a)) x2 = int(x0 - 1000\*(-b)) y2 = int(y0 - 1000\*(a))

GiaoDiem = ((400-y1)/(y1-y2))\*(x1-x2)+x1 if

10<GiaoDiem<150 or 500<GiaoDiem<640: arduino.write(b'1')

print("Di thang") elif

2.3<theta<2.6 and 400<GiaoDiem<500: arduino.write(b'2')

print("Re Trai")

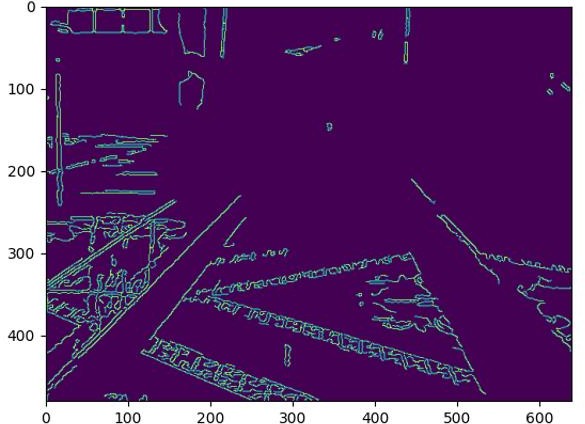
elif

0.35<theta<0.87 and 400<GiaoDiem<500: arduino.write(b'3')

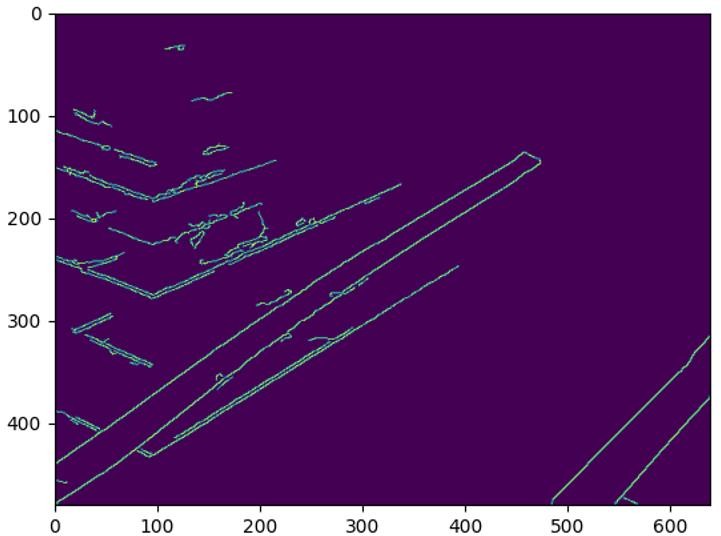
print("Re Phai")

Ở bước này qua quá trình thực nghiệm nhiều lần thì nhóm em tính được:

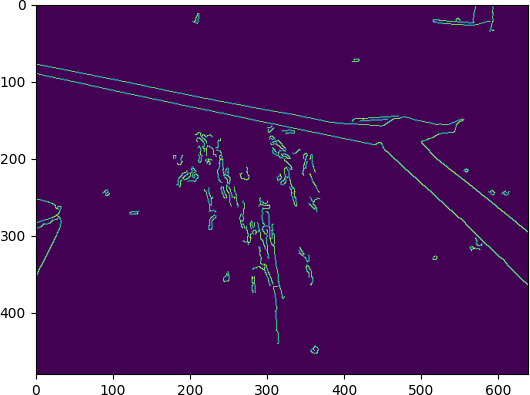
* + - * Trường hợp 1: Khi hoành độ giao điểm của vạch kẻ đường trái từ 10 đến 150 hay vạch kẻ đường phải từ 500 đến 640 thì vạch kẻ đường đó là đường thẳng.
      * Trường hợp 2: Khi hoành độ giao điểm của vạch kẻ đường phải từ 400 đến 600 và góc theta 2.3 đến 2.6 thì vạch kẻ đường đó là Rẽ Trái.
      * Trường hợp 3: Khi hoành độ giao điểm của vạch kẻ đường phải từ 400 đến 600 và góc theta 0.35 đến 0.87 thì vạch kẻ đường đó là Rẽ Phải.



**Hình 4.1.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được khoảng hoành độ giao điểm hợp lí với trường hợp đi xe đi thẳng



**Hình 4.2.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được khoảng hoành độ giao điểm hợp lí với trường hợp đi xe rẽ phải



**Hình 4.3.** Sử dụng thư viện matplotlib.pyplot để tìm tọa độ từ đó xác định được khoảng hoành độ giao điểm hợp lí với trường hợp đi xe rẽ trái

**Bước 10:** Sử dụng hàm chờ để nhận dữ liệu hình ảnh trực tiếp liên tục từ webcam của laptop và đóng kết nối giao tiếp giữa Arduino và python

if cv2.waitKey(1) & 0xFF == ord('q'): break

arduino.close()

**Bước 11:** Giải phóng webcam và đóng tất cả cửa sổ của lệnh imshow

cam.release() cv2.destroyAllWindows()

## Phần code hoàn chỉnh

import cv2

import numpy as np import serial

arduino = serial.Serial(port='COM5', baudrate=115200, timeout=0.1) #Tim 3 diem bang anh nhi phan

def region\_of\_interest(image): height=image.shape[0] polygons=np.array([

[(0,height),(640,height),(320,-200000)]

])

mask=np.zeros\_like(image) cv2.fillPoly(mask,polygons,255) masked\_image=cv2.bitwise\_and(image,mask) return masked\_image

#Hien hinh anh

def display\_lines(image,lines): line\_image = np.zeros\_like(image) if lines is not None:

for line in lines:

x1,y1,x2,y2 = line.reshape(4) cv2.line(line\_image,(x1,y1),(x2,y2),(255,0,0),10)

return line\_image

cam = cv2.VideoCapture(0) window\_name = 'Dectect Line' while(True):

Hinh1 = cam.read()[1]

gray = cv2.cvtColor(Hinh1, cv2.COLOR\_BGR2GRAY) blur = cv2.GaussianBlur(gray,(5,5),0)

edges = cv2.Canny(blur, 50, 150, apertureSize=3) cropped\_image = region\_of\_interest(edges)

lines = cv2.HoughLinesP(cropped\_image,2,np.pi/180,120,np.array([]), minLineLength=40,maxLineGap=1)

line\_control = cv2.HoughLines(edges, rho=1, theta=np.pi/180, threshold=100)

line\_image = display\_lines(Hinh1,lines) # Chong 2 hinh len nhau

combo\_image= cv2.addWeighted(Hinh1,0.8,line\_image,1,1) for line\_c in line\_control:

rho,theta = line\_c[0] a = np.cos(theta)

b = np.sin(theta) x0 = a\*rho

y0 = b\*rho

x1 = int(x0 + 1000\*(-b)) y1 = int(y0 + 1000\*(a)) x2 = int(x0 - 1000\*(-b)) y2 = int(y0 - 1000\*(a))

GiaoDiem = ((400-y1)/(y1-y2))\*(x1-x2)+x1 if 10<GiaoDiem<150 or 500<GiaoDiem<640:

arduino.write(b'1') print("Di thang")

elif 2.3<theta<2.6 and 400<GiaoDiem<500: arduino.write(b'2')

print("Re Trai")

elif 0.35<theta<0.87 and 400<GiaoDiem<500: arduino.write(b'3')

print("Re Phai") cv2.imshow(window\_name,combo\_image) #cv2.imshow(window\_name,Hinh1)

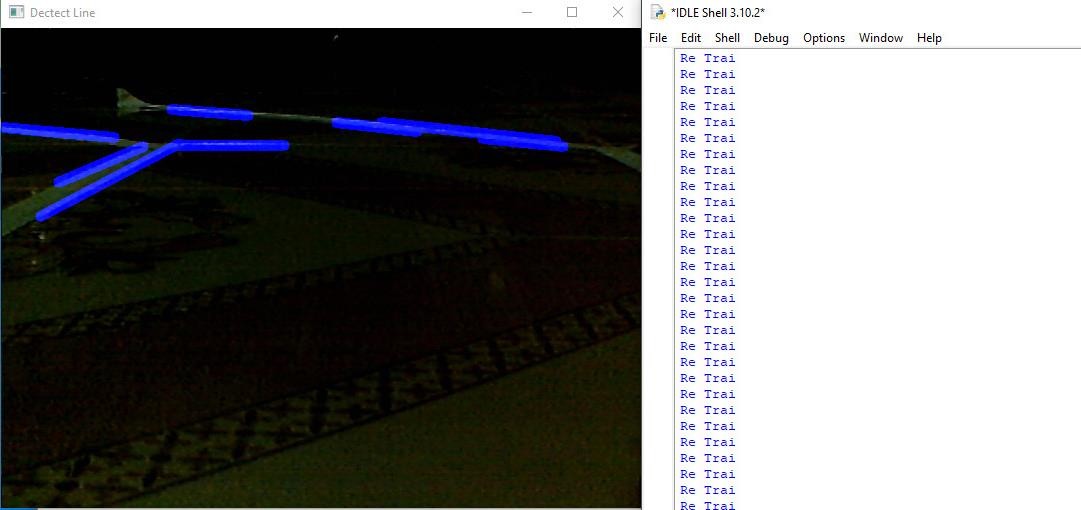
if cv2.waitKey(1) & 0xFF == ord('q'): break

arduino.close() cam.release() cv2.destroyAllWindows()

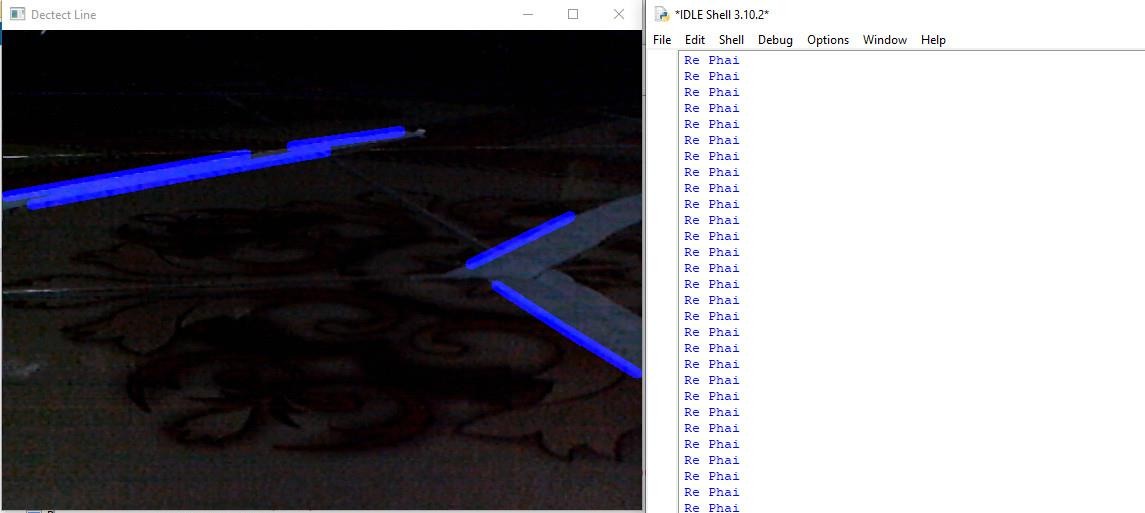
## Kết quả dò Line từ Python



**Hình 4.4.**. Hình ảnh dò Line đường đi thẳng



**Hình 4.5.** Hình ảnh dò Line rẽ trái



**Hình 4.6.** Hình ảnh dò Line rẽ phải

## Kết luận

**Chương 5**

**KẾT LUẬN VÀ KIẾN NGHỊ**

## Mục tiêu và kiến thức đạt được

Thông qua quá trình học kết hợp với việc tìm thêm tài liệu từ các ngồn mở như sách, internet… Hiểu được thức cơ bản về hệ thống điều hệ thống Lane-keeping Assist. Từ đó có được các ý tưởng nhằm phục vụ cho việc thực hiện mục tiêu chính của project đó chính là mô phỏng thực tế hệ thống Lane-keeping Assist bằng xe mô hình.

Nhóm đã hoàn thiện được phần cơ điện của xe đó là cố định và đi dây cho các linh kiện điện tử dựa trên khung xe có sẵn. Trong giai đoạn hoàn thiện này các thành viên trong nhóm có thêm các kỹ năng sử dụng các dụng cụ điện như đồng hồ đo điện, hàn chì, kìm tuốt dây.

Được trang bị thêm các kiến thức cơ bản về ngôn ngữ lập trình Arduino và Python thông qua các bài ví dụ của thầy và các tài liệu tham khảo thêm trên internet.

Ứng dụng được ngôn ngữ lập trình Arduino trong việc điều khiển xe cơ bản như chạy thẳng, lùi, rẽ trái, phải và thay đổi tốc độ của các bánh xe bằng các thay đổi code hoặc sử dụng biến trở.

Có được hình dung cụ thể hơn về cách mà một hệ thống điều khiển tự động trên ô tô hoạt động, cụ thể ở đây là hệ thống Lane-keeping Assit thông qua mô hình thực tế.

Cuối cùng là mục tiêu chính của Project, đó là dò line đường mô hình thông qua xử lý ảnh Python từ đó gửi tín hiệu điều khiển đến Arduino để điều hướng xe giữ line đường một cách tự động, nhóm về cơ bản đã hoàn thành thông qua nhiều lần thử nghiệm và điều chỉnh code.

## Thuận lợi trong quá trình làm project

Được thầy tạo điện kiện lên phòng lap và tận dụng thời gian dư sau tiết học để thực hiện project. Do đó, việc giải đáp thắc mắc cũng như hướng dẫn về project của nhóm được kịp thời và nhanh chóng.

Việc nhiều các cửa hàng bán các linh kiện điện tử xung quanh trường cũng góp phần không nhỏ vào việc cải thiện tiến độ của quá trình làm project. Giúp những linh kiện còn thiếu hoặc hư hỏng được bổ sung và thay thế nhanh hơn.

## Khó khăn trong quá trình làm project

Còn thiếu kiến thức về phần mạch điện nên trong quá trình nối dây cho xe mô hình bị sai khiến pin bị quá nóng, tuy nhiên vấn đề này đã được xử lý và khắc phục kịp thời thông qua sự gợi ý, chỉ dẫn của thầy.

Khó khăn trong việc tìm kiếm tài liệu liên quan đa phần là tiếng Anh và cũng khá ít ví dụ liên quan đến xử lý ảnh bằng Python để tham khảo. Do đó, nhóm vẫn chưa thể tối ưu khả năng điều hướng (rẽ chưa được mượt và ổn định) của xe nhằm giúp kết quả mô phỏng gần với thực tế nhất.

## Kiến nghị

Nhóm xin kiến nghị giảng viên có thể mở thêm nhiều phòng lab để các nhóm có thêm nhiều lựa chọn về thời gian để thực hiện project bên ngoài giờ học. Vì so với một phòng lab hiện tại là không đủ khi có rất nhiều nhóm.

## Tiếng Việt

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | TS Lê Thanh Phúc, “Tài liệu hướng dẫn Arduino”, Trường Đại học Sư phạm  Kỹ thuật TP. HCM. |
| [2] | TS. Lê Thanh Phúc, “Hướng Dẫn Python”, Trường Đại học Sư phạm Kỹ  thuật TP. HCM. |
| [3] | “Python là gì? 13 lý do bạn nên học ngôn ngữ lập trình Python.” Ironhack, 6 July 2021, <https://ironhackvietnam.edu.vn/python-la-gi/>., Ngày truy cập  04/06/2022. |
| [4] | “Xử lý hình ảnh bằng Python”, [https://koodibar.com/posts/xu-ly-hinh-anh-](https://koodibar.com/posts/xu-ly-hinh-anh-voi-python)  [voi-python](https://koodibar.com/posts/xu-ly-hinh-anh-voi-python) , Ngày truy cập 04/06/2022. |
| [5] | “Biết cách sử dụng hệ thống hỗ trợ giữ làn giúp đảm bảo an toàn khi cầm lái.” VinFast, [https://vinfastauto.com/vn\_vi/tim-hieu-he-thong-ho-tro-giu-](https://vinfastauto.com/vn_vi/tim-hieu-he-thong-ho-tro-giu-lan-duong-tren-o-to)  [lan-duong-tren-o-to](https://vinfastauto.com/vn_vi/tim-hieu-he-thong-ho-tro-giu-lan-duong-tren-o-to) . Ngày truy cập 04/06/2022. |
| [6] | “Phát hiện đường thẳng với Hough Transform - OpenCV.” AICurious, [https://aicurious.io/posts/2019-10-24-hough-transform-phat-hien-duong-](https://aicurious.io/posts/2019-10-24-hough-transform-phat-hien-duong-thang/)  [thang/](https://aicurious.io/posts/2019-10-24-hough-transform-phat-hien-duong-thang/) . Ngày truy cập 04/06/2022. |
| [7] | “Tìm đường thẳng bằng hàm Houghlines.” THỊ GIÁC MÁY TÍNH, <https://thigiacmaytinh.com/tim-duong-thang-bang-ham-houghlines/>. Ngày  truy cập 04/06/2022. |

**Tiếng Anh**

|  |  |
| --- | --- |
| [8] | “Why add and subtract 1000 for hough line transformation?” Stack Overflow, 25 June 2020, [https://stackoverflow.com/questions/62585845/why-add-and-subtract-](https://stackoverflow.com/questions/62585845/why-add-and-subtract-1000-for-hough-line-transformation)  [1000-for-hough-line-transformation](https://stackoverflow.com/questions/62585845/why-add-and-subtract-1000-for-hough-line-transformation) . Accessed 4 June 2022. |
| [9] | “Line detection in python with OpenCV | Houghline method.” GeeksforGeeks, 21 February 2022, [https://www.geeksforgeeks.org/line-](https://www.geeksforgeeks.org/line-detection-python-opencv-houghline-method/)  [detection-python-opencv-houghline-method/](https://www.geeksforgeeks.org/line-detection-python-opencv-houghline-method/) . Accessed 4 June 2022. |