

**TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG HÓA**

----🙣🕮🙡----

**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐIỀU KHIỂN VỊ TRÍ VÀ TỐC ĐỘ ĐỘNG CƠ DC**

**GVHD: ThS. TRẦN HOÀN**

**SVTH: VŨ QUANG NGỌC**

**MSSV: 2032180683**

**TP. HỒ CHÍ MINH, tháng 9 năm 2021**

**TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG HÓA**

**----🙣🕮🙡----**

**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐIỀU KHIỂN VỊ TRÍ VÀ TỐC ĐỘ ĐỘNG CƠ DC**

**GVHD: ThS. TRẦN HOÀN**

**SVTH: VŨ QUANG NGỌC**

**MSSV: 2032180683**

**TP. HỒ CHÍ MINH, tháng 9 năm 2021**

|  |  |
| --- | --- |
| TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM  KHOA CN ĐIỆN – ĐIỆN TỬ  BỘ MÔN: TỰ ĐỘNG HÓA | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày….tháng…..năm 2021* |

**NHẬN XÉT ĐỒ ÁN CHUYÊN NGÀNH**

**CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |  |  |
| --- | --- | --- |
| **Tên đồ án:** | | |
| **(ĐIỀU KHIỂN VỊ TRÍ VÀ TỐC ĐỘ ĐỘNG CƠ DC)** | | |
| **Sinh viên thực hiện:** | | **Giảng viên hướng dẫn:** |
| Vũ Quang Ngọc | 2032180683 | ThS. Trần Hoàn |
| **Đánh giá Đồ án**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung đồ án: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:**  **Điểm từng sinh viên:**  (Họ tên sinh viên):………..**/10** | | |

|  |  |
| --- | --- |
|  | **Người nhận xét**  (Ký tên và ghi rõ họ tên) |

**LỜI CÁM ƠN**

Trong thời gian làm đồ án chuyên ngành, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy, gia đình và bạn bè.

Em xin gởi lời cảm ơn chân thành đến thầy Th.S Trần Hoàn, giảng viên hướng dẫn đồ án chuyên ngành, người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm đề tài này.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong trường Đai học Công Nghiệp Thực Phẩm TP. Hồ Chí Minh nói chung, các thầy cô trong khoa Công Nghệ Kỹ thuật – Điều Khiển và Tự Động Hóa nói riêng đã dạy dỗ cho em về kiến thức các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đề tài đồ án lần này.

TP. Hồ Chí Minh, ngày 15 tháng 9 năm 2021

Tác giả

***Vũ Quang Ngọc***

|  |  |
| --- | --- |
| TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM  KHOA CN ĐIỆN – ĐIỆN TỬ  BỘ MÔN: TỰ ĐỘNG HÓA | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP.HCM, ngày 15 tháng 9 năm 2021* |

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỒ ÁN: ĐIỀU KHIỂN VỊ TRÍ VÀ TỐC ĐỘ ĐỘNG CƠ DC** | |
| **Giảng viên hướng dẫn: ThS. Trần Hoàn** | |
| **Thời gian thực hiện:** Từ ngày 07/03/2021 đến ngày 15/09/2021 | |
| **Sinh viên thực hiện: Vũ Quang Ngọc** | |
| **Nội dung đề tài:**   * Tìm hiểu tổng quan mô hình điều khiển tốc độ và vị trí động cơ DC Servo * Thiết kế bộ điều khiển PID cho hệ thống * Kiểm tra đánh giá kết quả dựa trên mô hình thực nghiệm | |
| **Kế hoạch thực hiện:**   * Từ ngày 07/03/2021 đến tháng 07/2021: Tìm hiểu tổng quan mô hình điều khiển tốc độ và vị trí động cơ DC Servo * Từ tháng 07/2021 đến tháng 08/2021: Tìm hiểu về bộ điều khiển PID số trên mạng * Từ ngày 29/08/2021 đến ngày 15/9/2021: Viết báo cáo Đồ án chuyên ngành | |
| **Xác nhận của giảng viên hướng dẫn** | TP. HCM, ngày 15 tháng 9 năm 2021  **Sinh viên**  **Vũ Quang Ngọc** |

# MỤC LỤC

[MỤC LỤC i](#_Toc82594861)

[DANH MỤC BẢNG BIỂU iii](#_Toc82594862)

[DANH MỤC HÌNH ẢNH iv](#_Toc82594863)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc82594864)

[1.1 Đặt vấn đề 1](#_Toc82594865)

[1.2 Các công trình nghiên cứu liên quan 2](#_Toc82594866)

[1.3 Mục tiêu đề tài 3](#_Toc82594867)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc82594868)

[2.1 Kit Arduino UNO R3 4](#_Toc82594869)

[2.1.1 Giới thiệu về Arduino UNO R3 4](#_Toc82594870)

[2.1.2 Thông số kỹ thuật và cấu tạo của Arduino UNO R3 5](#_Toc82594871)

[2.2 Module điều khiển động cơ L298N 10](#_Toc82594872)

[2.2.1 Cách thức hoạt động 10](#_Toc82594873)

[2.2.2 Nguyên lý hoạt động 12](#_Toc82594874)

[2.3 Pin sạc 18650 và đế pin 15](#_Toc82594875)

[2.4 Động cơ điện DC Servo 16](#_Toc82594876)

[2.4.1 Nguyên lý làm việc của động cơ DC Servo 17](#_Toc82594877)

[2.4.2 Đặc điểm của động cơ DC Servo 18](#_Toc82594878)

[2.4.3 Ứng dụng của động cơ DC Servo 18](#_Toc82594879)

[2.4.4 Sự khác nhau giữa động cơ Servo và động cơ DC 19](#_Toc82594880)

[2.4.5 Ưu, nhược điểm của động cơ DC Servo 19](#_Toc82594881)

[2.4.6 Động cơ DC Servo được sử dụng trong đề tài 20](#_Toc82594882)

[2.5 Encoder 21](#_Toc82594883)

[2.5.1 Giới thiệu về Encoder 21](#_Toc82594884)

[2.5.2 Giới thiệu về Encoder 21](#_Toc82594885)

[2.5.3 Nguyên lý hoạt động 23](#_Toc82594886)

[2.6 Ngôn ngữ Arduino 26](#_Toc82594887)

[2.7 PID 27](#_Toc82594888)

[2.7.1 Giới thiệu PID 27](#_Toc82594889)

[2.7.2 Hàm truyền 28](#_Toc82594890)

[2.7.3 Đặc tính bộ điều khiển PID 29](#_Toc82594891)

[CHƯƠNG 3: CƠ SỞ THỰC HIỆN 31](#_Toc82594892)

[3.1 Lưu đồ giải thuật 31](#_Toc82594893)

[3.2 Đọc vị trí động cơ 31](#_Toc82594894)

[3.3 Các phương pháp chọn tham số cho bộ diều khiển PID 33](#_Toc82594895)

[3.3.1 Phương pháp Ziegler-Nichols: 33](#_Toc82594896)

[3.3.2 Phương pháp Chien-Hrones-Reswick: 35](#_Toc82594897)

[3.3.3 Phương pháp chỉnh định bằng thực nghiệm (thử và sai): 37](#_Toc82594898)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 40](#_Toc82594899)

[4.1. Thực nghiệm đánh giá độ chính xác của vị trí động cơ: 40](#_Toc82594900)

[4.2. Kết quả điều khiển vị trí mong muốn động cơ: 41](#_Toc82594901)

[4.3. Thực nghiệm đánh giá độ chính xác của tốc độ động cơ: 46](#_Toc82594902)

[4.4. Kết quả điều khiển tốc độ mong muốn động cơ: 47](#_Toc82594903)

[4.5. Hình ảnh mô hình điều khiển tốc độ vị trí động cơ: 52](#_Toc82594904)

[CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI 54](#_Toc82594905)

[5.1 Kết quả đạt được 54](#_Toc82594906)

[5.2 Hạn chế 54](#_Toc82594907)

[5.3 Hướng phát triển của đề tài 54](#_Toc82594908)

[PHỤ LỤC 55](#_Toc82594909)

[Code chương trình: 55](#_Toc82594910)

[Code điều khiển vị trí động cơ 55](#_Toc82594911)

[Code điều khiển tốc độ động cơ 58](#_Toc82594912)

[TÀI LIỆU THAM KHẢO 62](#_Toc82594913)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1 Sự khác nhau giữa động cơ Servo và động cơ DC 19](#_Toc82585380)

[Bảng 2.2 Ảnh hưởng của các thành phần Kp, Ki, Kd 29](#_Toc82585381)

[Bảng 3.1 Các tham số PID theo phương pháp Ziegler-Nichols thứ nhất 33](#_Toc82585385)

[Bảng 3.2 Các tham số PID theo phương pháp Ziegler-Nichols thứ hai 35](#_Toc82585386)

[Bảng 3.3 Các tham số PID theo phương pháp Chien-Hrones-Reswick 1 36](#_Toc82585387)

[Bảng 3.4 Các tham số PID theo phương pháp Chien-Hrones-Reswick 2 36](#_Toc82585388)

[Bảng 3.5 Các tham số PID theo phương pháp Chien-Hrones-Reswick 3 36](#_Toc82585389)

[Bảng 3.6 Các tham số PID theo phương pháp Chien-Hrones-Reswick 4 37](#_Toc82585390)

[Bảng 4.1 Bảng đánh giá thực nghiệm độ chính xác điều khiển vị trí 40](#_Toc82585392)

[Bảng 4.2 Bảng đánh giá thực nghiệm độ chính xác điều khiển tốc độ 46](#_Toc82585393)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Dây chuyền hàn thân vỏ có mức độ tự động hoá 100% 2](#_Toc82594914)

[Hình 2.1 Arduino UNO R3 4](#_Toc82594922)

[Hình 2.2 Thông số phần cứng của Arduino UNO R3 5](#_Toc82594923)

[Hình 2.3 Sơ đồ chân của Arduino UNO R3 9](#_Toc82594924)

[Hình 2.4 Module điều khiển động cơ L298N 10](#_Toc82594925)

[Hình 2.5 Cách thức hoạt động mạch L298 11](#_Toc82594926)

[Hình 2.6 Hoạt động mạch L298 11](#_Toc82594927)

[Hình 2.7 Sơ đồ tổng quát của mạch cầu H sử dụng transistor BJT 12](#_Toc82594928)

[Hình 2.8 A ở mức LOW và B mức HIGH 13](#_Toc82594929)

[Hình 2.9 A ở mức HIGH và B mức LOW 14](#_Toc82594930)

[Hình 2.10 Pin sạc Lithium Li-on Samsung ICR18650 15](#_Toc82594931)

[Hình 2.11 Đế pin 16](#_Toc82594932)

[Hình 2.12 Sơ đồ khối của động cơ Servo 17](#_Toc82594933)

[Hình 2.13 Động cơ DC Servo DSE38BE27-001 20](#_Toc82594934)

[Hình 2.14 Bộ encoder trên thị trường 21](#_Toc82594935)

[Hình 2.15 Encoder kiểu tuyệt đối 22](#_Toc82594936)

[Hình 2.16 Encoder kiểu tương đối 23](#_Toc82594937)

[Hình 2.17 Mã hóa quang học 24](#_Toc82594938)

[Hình 2.18 Hai kênh A và B lệch pha trong encoder 25](#_Toc82594939)

[Hình 2.19 Sơ đồ khối của bộ điều khiển PID 28](#_Toc82594940)

[Hình 2.20 Sơ đồ khối của một hệ thống 28](#_Toc82594941)

[Hình 3.1 Lưu đồ giải thuật 31](#_Toc82594942)

[Hình 3.2 Có được vị trí của động cơ khi quay âm (cùng chiều kim đồng hồ) 32](#_Toc82594943)

[Hình 3.3 Có được vị trí của động cơ quay dương (ngược chiều kim đồng hồ) 32](#_Toc82594944)

[Hình 3.4 Đáp ứng nấc của hệ hở có dạng S 33](#_Toc82594945)

[Hình 3.5 Xác định hằng số khuếch đại tới hạn 34](#_Toc82594946)

[Hình 3.6 Đáp ứng nấc của hệ kín khi k = kth 34](#_Toc82594947)

[Hình 3.7 Đáp ứng nấc của hệ thích hợp phương pháp Chien-Hrones-Reswick 35](#_Toc82594948)

[Hình 3.8 Thực nghiệm đặt Ki và Kd = 0, tăng Kp đến khi có dao động 37](#_Toc82594949)

[Hình 3.9 Thực nghiệm giảm hệ số Kp và tăng hệ số Kd 38](#_Toc82594950)

[Hình 3.10 Thực nghiệm tăng hệ số Ki và điều chỉnh hệ số Kp, Kd 39](#_Toc82594951)

[Hình 4.1 Kết quả khi đặt vị trí mong muốn là vị trí 6 xung 41](#_Toc82594952)

[Hình 4.2 Kết quả khi đặt vị trí mong muốn là 26 xung 41](#_Toc82594953)

[Hình 4.3 Kết quả khi đặt vị trí mong muốn là 260 xung 42](#_Toc82594954)

[Hình 4.4 Kết quả khi đặt vị trí mong muốn là 2600 xung 42](#_Toc82594955)

[Hình 4.5 Kết quả khi đặt vị trí mong muốn là 26000 xung 43](#_Toc82594956)

[Hình 4.6 Kết quả khi đặt vị trí mong muốn là -6 xung 43](#_Toc82594957)

[Hình 4.7 Kết quả khi đặt vị trí mong muốn là -26 xung 44](#_Toc82594958)

[Hình 4.8 Kết quả khi đặt vị trí mong muốn là -260 xung 44](#_Toc82594959)

[Hình 4.9 Kết quả khi đặt vị trí mong muốn là -2600 xung 45](#_Toc82594960)

[Hình 4.10 Kết quả khi đặt vị trí mong muốn là -26000 xung 45](#_Toc82594961)

[Hình 4.11 Kết quả khi đặt tốc độ mong muốn là 1 vòng/giây 47](#_Toc82594962)

[Hình 4.12 Kết quả khi đặt tốc độ mong muốn là 3 vòng/giây 47](#_Toc82594963)

[Hình 4.13 Kết quả khi đặt tốc độ mong muốn là 5 vòng/giây 48](#_Toc82594964)

[Hình 4.14 Kết quả khi đặt tốc độ mong muốn là 7 vòng/giây 48](#_Toc82594965)

[Hình 4.15 Kết quả khi đặt tốc độ mong muốn là 10 vòng/giây 49](#_Toc82594966)

[Hình 4.16 Kết quả khi đặt tốc độ mong muốn là 12 vòng/giây 49](#_Toc82594967)

[Hình 4.17 Kết quả khi đặt tốc độ mong muốn là 15 vòng/giây 50](#_Toc82594968)

[Hình 4.18 Kết quả khi đặt tốc độ mong muốn là 18 vòng/giây 50](#_Toc82594969)

[Hình 4.19 Kết quả khi đặt tốc độ mong muốn là 24 vòng/giây 51](#_Toc82594970)

[Hình 4.20 Kết quả khi đặt tốc độ mong muốn là 36 vòng/giây 51](#_Toc82594971)

[Hình 4.21 Mô hình thực tế 52](#_Toc82594972)

[Hình 4.22 Sơ đồ kết nối của hệ thống điều khiển vị trí 52](#_Toc82594973)

[Hình 4.23 Sơ đồ kết nối của hệ thống điều khiển tốc độ động cơ 53](#_Toc82594974)

[Hình 4.24 Thiết kề phần cứng mô hình điều khiển tốc độ và vị trí động cơ 53](#_Toc82594975)

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Đặt vấn đề

Lĩnh vực điều khiển tự động ngày càng phát triển, đặc biệt là điều khiển chính xác, đã trở thành một phần không thể thiểu của nền công nghiệp hiện đai. Phần lớn các loại máy móc, thiết bị dân dụng hay trong công nghiệp sử dụng động cơ điện, từ động cơ điện trong các máy công cụ, máy CNC, các cánh tay robot,… đến trong những thiết bị gia dụng như máy giặt, điều hòa, máy hút bụi, ngay cả trong máy vi tính. Những thiết bị như vậy yêu cầu độ chính xác cao, tiết kiệm năng lượng, tuổi thọ và chu kỳ bảo dưỡng dài. Một trong những yêu cầu cần được đáp ứng để đạt những chỉ tiêu trên đây là điều khiển được tốc độ động cơ điện một cách ổn định, đáp ứng nhanh, vận hành trơn tru khi xác lập và khi thay đổi trạng thái.

Việc ứng dụng những thuật toán kinh điển vào vấn đề điều khiển tốc độ động cơ đã được nhiều kết quả khả quan. Ví dụ như sử dụng bộ điều khiển PID cho kết quả tốt ở một số đối tượng động cơ. Chỉnh định tham số cho bộ điều khiển PID kinh điển cũng có nhiều phương pháp. Tuy nhiên, với các thuật toán, phương pháp kinh điển, ta phải biết chính xác về đối tượng, hoặc mô hình hóa tương đối chi tiết đối tượng. Một điểm nữa là trong quá trình vận hành, nếu đối tượng thay đổi thì hệ thống có thể mất ổn định hoặc chất lượng điều khiển không còn đáp ứng được yêu cầu.

Trong điều khiển hiện đại, lý thuyết mở cung cấp cho ta một hướng đi mới, xây dựng những hệ điều khiển mờ thuần túy hoặc những hệ mờ lai với mục đích nâng cao chất lượng các bộ điều khiển kinh điển, cũng như điều khiển những đối tượng chưa biết hoặc khó nhận dạng.

Trong khuôn khổ Đồ án, em xin trình bày điều khiển tốc độ và vị trí động cơ DC Servo bằng thuật toán PID, xây dựng các bộ điều khiển này trên nền vi điều khiển Arduino, các kết quả thu được và hướng phát triển đề tài. Do hạn chế kiến thức cũng như về thiết bị hỗ trợ, nếu có sai sót mong thầy góp ý và bỏ qua cho em.

## Các công trình nghiên cứu liên quan

Khi nhắc tới về các động cơ, robot,.. Không thể không nhắc tới sự kiện toàn bộ hệ thống 1200 Robot đã được lắp ráp và đi vào hoạt động sản xuất thử ô tô VinFast kể từ tháng 3/2019.

Xưởng Hàn thân xe VinFast có tổng diện tích 100.000m2 được thiết kế và cung ứng dây chuyền bởi các đối tác hàng đầu thế giới như FFT, EBZ đến từ Đức, HIROTEC đến từ Nhật Bản. Mối liên kết giữa các Robot, giữa con người và hệ thống máy móc thiết bị đều được hoàn chỉnh theo nền tảng công nghệ 4.0.



Hình 1.1 Dây chuyền hàn thân vỏ có mức độ tự động hoá 100%

Việc điều khiển chính xác các vị trí, tốc độ các động cơ robot là một phần quan trọng trong quá trính sản xuất của dây chuyền trong nhà máy.

## Mục tiêu đề tài

Hoàn thành đa số các vấn đề về đề tài điều khiển tốc độ và vị trí của động cơ DC theo mong muốn.

Hiểu được một ít về động cơ điện DC, các đặc tính, phương pháp điều khiển chúng.

Sẽ nâng cao được kỹ năng lập trình, và trao dồi thêm kiến thức về ngôn ngữ lập trình Arduino.

Kích thích sự sáng tạo và đưa ra nhiều hướng giải quyết khi gặp vấn đề khó trong quá trình lập trình.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

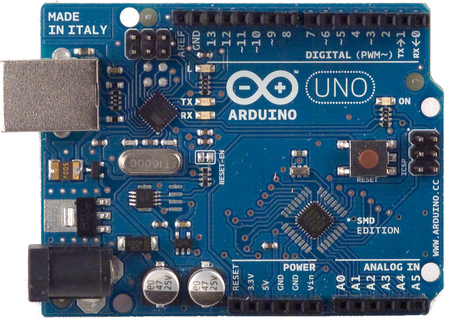
## 2.1 Kit Arduino UNO R3

### 2.1.1 Giới thiệu về Arduino UNO R3

Arduino Uno là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P. Với Arduino chúng ta có thể xây dựng các ứng dụng điện tử tương tác với nhau thông qua  phần mềm và phần cứng hỗ trợ.

Khi arduino chưa ra đời, để làm được một dự án điện tử nhỏ liên quan đến lập trình, biên dịch, chúng ta cần đến sự hỗ trợ của các thiết bị biên dịch khác để hỗ trợ. Ví dụ như, dùng Vi điều khiển PIC hoặc IC vi điều khiển họ 8051..., chúng ta phải thiết kế chân nạp onboard, hoặc mua các thiết bị hỗ trợ nạp và biên dịch như mạch nạp 8051, mạch nạp PIC...

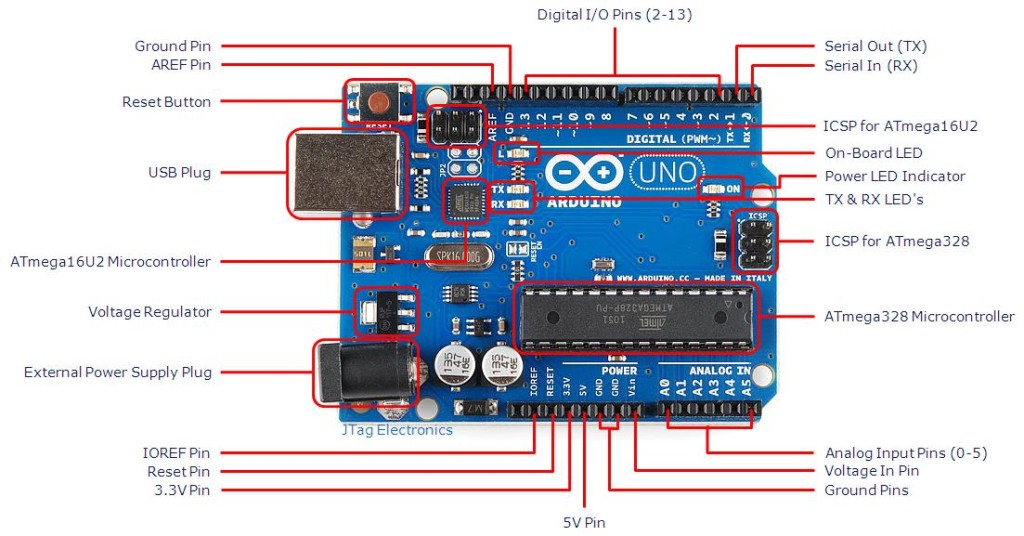
Hiện nay Arduino được biết đến ở Việt Nam rất rộng rãi. Từ học sinh trung học, đến sinh viên và người đi làm. Những dự án nhỏ và lớn được thực hiện một cách rất nhanh, các mã nguồn mở được chia sẻ nhiều trên diễn dàn trong nước và nước ngoài. Giúp ích rất nhiều cho những bạn theo đam mê nghiên cứu chế tạo những sản phẩm có ích cho xã hội.



Hình 2.1 Arduino UNO R3

Trên thị trường có rất nhiều phiên bản Arduino như Arduino Uno R3, Arduino Uno R3 CH340, Arduino Mega2560, Arduino Nano, Arduino Pro Mino, Arduino Lenadro, Arduino Industrial....

### 2.1.2 Thông số kỹ thuật và cấu tạo của Arduino UNO R3



Hình 2.2 Thông số phần cứng của Arduino UNO R3

**Thông số kỹ thuật:**

* Chip điều khiển chính: ATmega328P
* Chip nạp và giao tiếp UART: ATmega16U2
* Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (nếu sử dụng nguồn ngoài từ giắc tròn DC Hshop.vn khuyên bạn nên cấp nguồn từ 6~9VDC để đảm bảo mạch hoạt động tốt, nếu bạn cắm 12VDC thì IC ổn áp rất nóng, dễ cháy và gây hư hỏng mạch).
* Số chân Digital I/O: 14 (trong đó 6 chân có khả năng xuất xung PWM).
* Số chân PWM Digital I/O: 6
* Số chân Analog Input: 6
* Dòng điện DC Current trên mỗi chân I/O: 20 mA
* Dòng điện DC Current chân 3.3V: 50 mA
* Flash Memory: 32 KB (ATmega328P), 0.5 KB dùng cho bootloader.
* SRAM: 2 KB (ATmega328P)
* EEPROM: 1 KB (ATmega328P)
* Clock Speed: 16 MHz
* LED\_BUILTIN: 13
* Kích thước: 68.6 x 53.4 mm

**Các chân năng lượng**

**GND (Ground)**: cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

**5V**: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.

**3.3V**: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.

**Vin (Voltage Input)**: để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.

**IOREF**: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.

**RESET**: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

*Lưu ý:*

Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. mình khuyên bạn nên dùng nguồn từ cổng USB nếu có thể.

Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích.

Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.

Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.

Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

Cấp điệp áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.

Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kĩ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kĩ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2 .Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kĩ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kĩ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2.

**Bộ nhớ**

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

**32KB bộ nhớ Flash**: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.

**2KB cho SRAM** (**S**tatic **R**andom **A**ccess **M**emory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

**2 chân Serial**: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết

**Chân PWM (~): 3, 5, 6, 9, 10, và 11**: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

**Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).  Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

**LED 13**: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

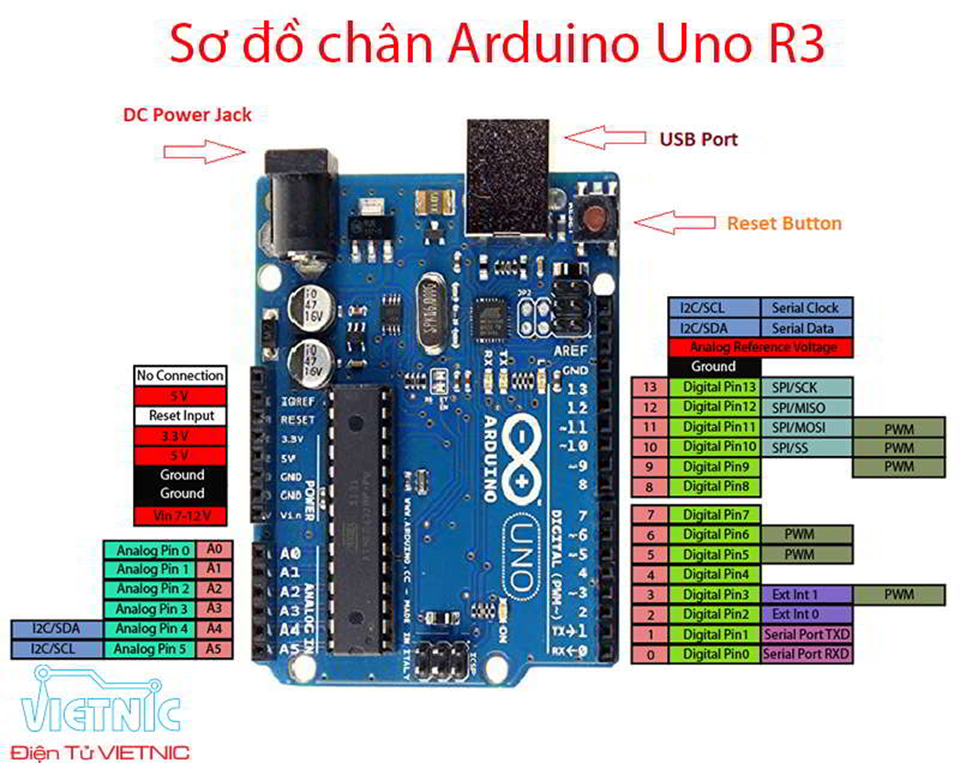
Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V  → 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

**Những lưu ý khi xử dụng Arduino UNO R3:**

Giới hạn điện áp cấp cho Uno là 6 – 20V. Tuy nhiên, dải điện áp khuyên dùng là 7 – 12 V (tốt nhất là 9V). Lý do là nếu nguồn cấp dưới 7V thì điện áp ở ‘chân 5V’ có thể thấp hơn 5V và mạch có thể hoạt động không ổn định; nếu nguồn cấp lớn hơn 12V có thể gấy nóng bo mạch hoặc phá hỏng.

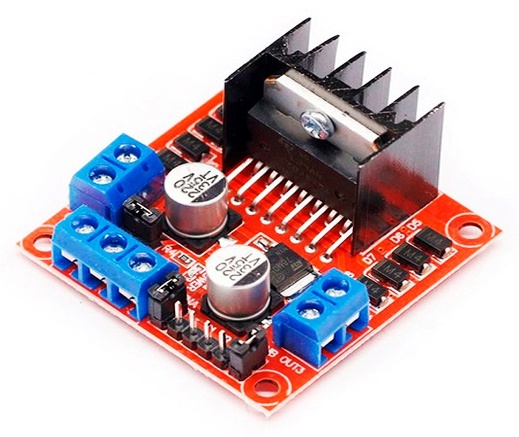
Chúng ta có thể cấp nguồn cho Uno thông qua chân Vin. Cách cấp nguồn này ít được sử dụng. Chân 5V: Chân này có thể cho nguồn 5V từ bo mạch Uno. Việc cấp nguồn vào chân này hay chân 3.3 V đều có thể phá hỏng bo mạch.



Hình 2.3 Sơ đồ chân của Arduino UNO R3

## 2.2 Module điều khiển động cơ L298N

Mạch điều khiển động cơ DC L298 có khả năng điều khiển 2 động cơ DC. Dòng tối đa là 2A mỗi động cơ, mạch tích hợp diod bảo vệ và IC nguồn 7805 giúp cấp nguồn 5VDC cho các module khác (chỉ sử dụng 5V này nếu nguồn cấp <12VDC).



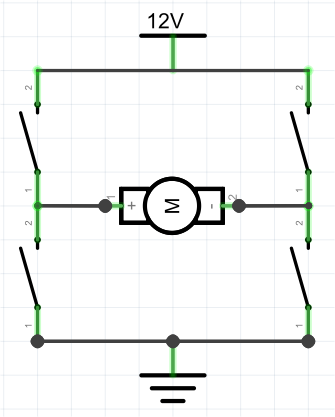
Hình 2.4 Module điều khiển động cơ L298N

**Thông số kỹ thuật:**

* Driver: L298N tích hợp hai mạch cầu H
* Điện áp điều khiển: +5 V ~ +12 V
* Dòng tối đa cho mỗi cầu H là: 2A (=>2A cho mỗi motor)
* Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
* Dòng của tín hiệu điều khiển: 0 ~ 36Ma
* Công suất hao phí: 20W (khi nhiệt độ T = 75 ℃)

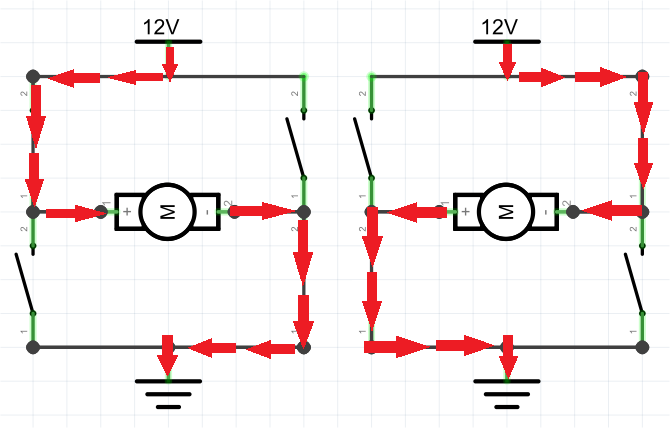
### 2.2.1 Cách thức hoạt động

Xét một cách tổng quát, mạch cầu H là một mạch gồm 4 "công tắc" được mắc theo hình chữ H.



Hình 2.5 Cách thức hoạt động mạch L298

Bằng cách điều khiển 4 "công tắc" này đóng mở, ta có thể điều khiển được dòng điện qua động cơ cũng như các thiết bị điện tương tự.

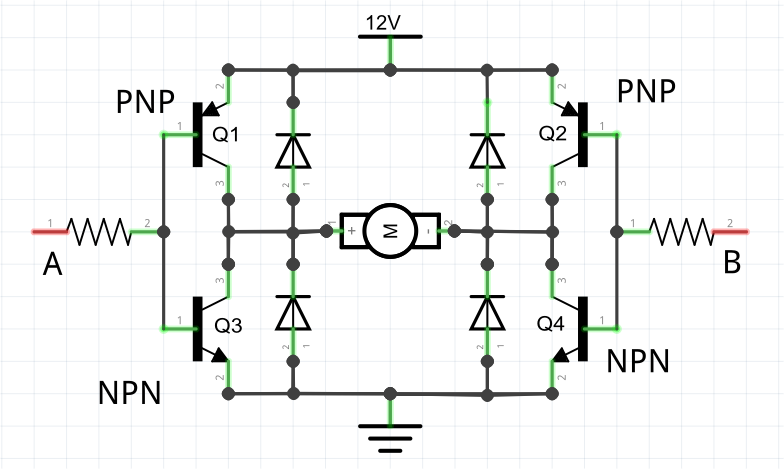


Hình 2.6 Hoạt động mạch L298

4 "công tắc" này thường là Transistor BJT, MOSFET hay relay. Tùy vào yêu cầu điều khiển khác nhau mà người ta lựa chọn các loại "công tắc" khác nhau.

Mạch cầu H dùng transistor BJT là loại mạch được sử dụng khá thông dụng cho việc điều khiển các loại động cơ công suất thấp. Lí do đơn giản là vì transistor BJT thường có công suất thấp hơn các loại MOSFET (relay thì không phải bàn rồi), đồng đời cũng rẻ và dễ tìm mua, sử dụng đơn giản.

Đây là sơ đồ tổng quát của một mạch cầu H sử dụng transistor BJT.

[](http://k1.arduino.vn/img/2014/08/03/0/745_12320-1407043541-0-h-bridge.png)

Hình 2.7 Sơ đồ tổng quát của mạch cầu H sử dụng transistor BJT

Trong sơ đồ này, A và B là hai cực điều khiển. Bốn diode có nhiệm vụ triệt tiêu dòng điện cảm ứng sinh ra trong quá trình động cơ làm việc. Nếu không có diode bảo vệ, dòng điện cảm ứng trong mạch có thể làm hỏng các transistor.

Transistor BJT được sử dụng nên là loại có công suất lớn và hệ số khếch đại lớn.

### 2.2.2 Nguyên lý hoạt động

Theo như sơ đồ trên, ta có A và B là 2 cực điều khiển được mắc nối tiếp với 2 điện trở hạn dòng, Tùy vào loại transistor bạn đang dùng mà trị số điện trở này khác nhau. Phải đảm bảo rằng dòng điện qua cực Base của các transistor không quá lớn để làm hỏng chúng. Trung bình thì dùng điện trở 1k Ohm.

Ta điều khiển 2 cực này bằng các mức tín hiệu [HIGH](http://arduino.vn/reference/high), [LOW](http://arduino.vn/reference/low) tương ứng là 12V và 0V.

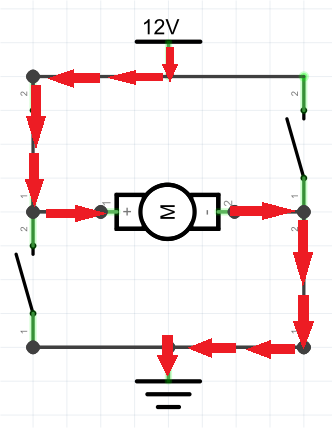
Nhớ lại rằng: Transistor BJT loại NPN mở hoàn toàn khi điện áp ở cực Base bằng điện áp ở cực Collector, trong mạch đang xét hiện tại là 12V. Transistor BJT loại PNP mở hoàn toàn khi điện áp ở cực Base bằng 0V.

Với 2 cực điều khiển và 2 mức tín hiệu HIGH/LOW tương ứng 12V/0V cho mỗi cực, có 4 trường hợp xảy ra như sau:

**A ở mức LOW và B ở mức HIGH**

Ở phía A, transistor Q1 mở, Q3 đóng. Ở phía B, transistor Q2 đóng, Q 4 mở. Dó đó, dòng điện trong mạch có thể chạy từ nguồn 12V đến Q1, qua động cơ đến Q4 để về GND. Lúc này, động cơ quay theo chiều thuận. Bạn để ý các cực (+) và (-) của động cơ là sẽ thấy.

Hình dung dòng điện trong mạch nó như thế này:

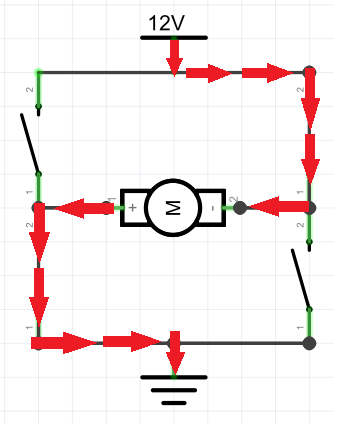


Hình 2.8 A ở mức LOW và B mức HIGH

**A ở mức HIGH và B ở mức LOW**

Ở phía A, transistor Q1 đóng, Q3 mở. Ở phía B, transistor Q2 mở, Q4 đóng. Dó đó, dòng điện trong mạch có thể chạy từ nguồn 12V đến Q2, qua động cơ đến Q3 để về GND. Lúc này, động cơ quay theo chiều ngược.

Bạn có thể hình dung dòng điện trong mạch nó như thế này

[](http://k2.arduino.vn/img/2014/08/04/0/703_88220-1407130257-0-dc4.png)

Hình 2.9 A ở mức HIGH và B mức LOW

**A và B cùng ở mức LOW**

Khi đó, transistor Q1 và Q2 mở nhưng Q3 và Q4 đóng. Dòng điện không có đường về được GND do đó không có dòng điện qua động cơ - động cơ không hoạt động.

**A và B cùng ở mức HIGH**

Khi đó, transistor Q1 và Q2 đóng nhưng Q3 và Q4 mở. Dòng điện không thể chạy từ nguồn 12V ra do đó không có dòng điện qua động cơ - động cơ không hoạt động.

Như vậy, để dừng động cơ, điện áp ở 2 cực điều khiển phải bằng nhau.

## 2.3 Pin sạc 18650 và đế pin

**Pin 18650:** là một loại pin lithium-ion có thể sạc lại. Pin lithium-ion là cuộc cách mạng hóa các thiết bị cầm tay. Pin 18650 là pin có kích thước 18mm x 65mm. Mã pin 18650 dành riêng cho kích thước của pin lithium-ion với nhiều thương hiệu sản xuất như pin Panasonic, Sony, Ansmann, Akasha… đã trở thành tiêu chuẩn vàng mới cho pin có thể thay thế và có thể sạc lại.



Hình 2.10 Pin sạc Lithium Li-on Samsung ICR18650

Ý nghĩa của con số 18650:

Hai số đầu là đường kính viên pin: ở đây là 18mm. Hai số tiếp theo là chiều sài viên pin, khoảng: 65mm. Số 0 cuối cùng để chỉ viên pin có hình trụ.

**Thông số kỹ thuật:**

* Kiểu pin: 18650
* Điện áp trung bình 3.7VDC, sạc đầy 4.2VDC.
* Dung lượng: 2600mAh
* Dòng xả min: 5A
* Dòng xả Max: 10A
* Size: 18 x 65 mm
* Trọng Lượng: 45g.

**Đế pin 18650 4 cell**:

* Các cell pin được nối tiếp với nhau với điện áp tối đa là: 4,2 x 4
* Dây màu đỏ là dương (+)
* Dây màu đen là âm (-)



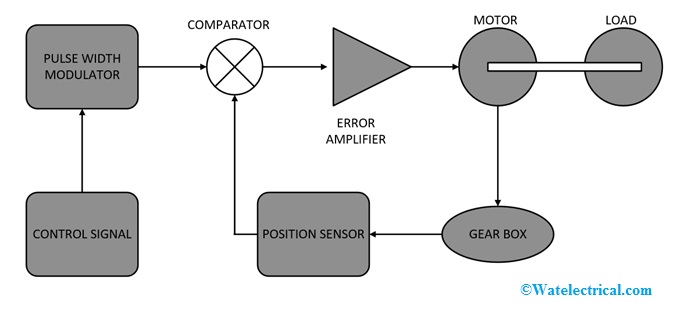
Hình 2.11 Đế pin

## 2.4 Động cơ điện DC Servo

Động cơ DC cùng với cơ chế Servo (hệ thống điều khiển vòng kín) hoạt động như một động cơ Servo về cơ bản được sử dụng như một [bộ chuyển đổi](https://www.watelectrical.com/what-is-piezoelectric-transducer-theory-circuit-formula-applications/) cơ khí trong ngành công nghiệp tự động hóa. Dựa trên khả năng điều khiển vòng kín chính xác, nó có các ứng dụng linh hoạt được sử dụng trong nhiều ngành công nghiệp khác nhau.

Động cơ DC Servo là một động cơ được sử dụng trong các hệ thống servo được biết đến như một động cơ servo. Hệ thống Servo là một hệ thống vòng kín trong đó tín hiệu phản hồi (vị trí, vận tốc, gia tốc, v.v.) điều khiển [động cơ](https://www.watelectrical.com/types-of-electric-motor/). Tín hiệu này hoạt động như một lỗi và dựa trên bộ điều khiển, vị trí hoặc vận tốc chính xác đạt được. Các động cơ được nối với trục đầu ra (tải) thông qua một hộp số để khớp với công suất. Động cơ Servo hoạt động như một bộ chuyển đổi cơ học khi chúng chuyển đổi tín hiệu điện thành vận tốc góc hoặc vị trí.

### 2.4.1 Nguyên lý làm việc của động cơ DC Servo



Hình 2.12 Sơ đồ khối của động cơ Servo

Motor: nó là một động cơ điện một chiều bình thường với cuộn dây trường được kích riêng biệt. Dựa trên bản chất, nó có thể được phân loại thêm thành một động cơ Servo điều khiển trường và động cơ Servo điều khiển [phần ứng](https://www.watelectrical.com/armature-definition-control-method-applications/).

Load: nó được kết nối với một trục cơ của động cơ. Nó có thể là tải công nghiệp hoặc một tải đơn giản của quạt.

Gear box: nó hoạt động như một bộ chuyển đổi cơ học để chuyển đổi đầu ra của động cơ dưới dạng vị trí, gia tốc hoặc vận tốc dựa trên ứng dụng.

Position Sensor: Nó cảm biến vị trí của rôto và đưa nó đến bộ so sánh. Cảm biến thường được sử dụng là cảm biến hiệu ứng Hall (là một loại cảm biến phát hiện sự hiện diện và cường độ của từ trường).

Comparator: Nó so sánh đầu ra của cảm biến vị trí và điểm tham chiếu để tạo ra tín hiệu lỗi và cấp cho bộ khuếch đại. Nếu động cơ chạy với sự điều khiển chính xác, thì sai số bằng không. Hộp số, cảm biến vị trí và bộ so sánh làm cho hệ thống khép kín.

Amplifier: Nó khuếch đại lỗi từ bộ so sánh, để đưa vào động cơ. Nó hoạt động giống như một bộ điều khiển tỷ lệ trong đó độ lợi được khuếch đại cho lỗi trạng thái ổn định bằng không.

Control signal và Pulse Width Modulator:

Dựa trên tín hiệu phản hồi, được điều khiển đưa ra đầu vào cho bộ điều chế độ rộng xung để điều chỉnh đầu vào của động cơ (điện áp hoặc kích từ trường) để điều khiển chính xác hoặc không có lỗi trạng thái ổn định. Bộ điều chế độ rộng xung tiếp tục sử dụng một dạng sóng tham chiếu và bộ so sánh để tạo ra các xung.

Bằng cách làm cho hệ thống khép kín vị trí chính xác, gia tốc hoặc vận tốc sẽ đạt được. Tên động cơ Servo ngụ ý là một động cơ được điều khiển cung cấp đầu ra mong muốn do tác động của phản hồi và bộ điều khiển. Tín hiệu lỗi được khuếch đại và được sử dụng để điều khiển động cơ servo. Dựa trên bản chất của việc tạo ra tín hiệu điều khiển và bộ điều chế độ rộng xung, động cơ servo có thể có các kỹ thuật điều khiển tiên tiến bằng cách sử dụng bộ xử lý tín hiệu kỹ thuật số hoặc chip FPGA.

### 2.4.2 Đặc điểm của động cơ DC Servo

* Quán tính của động cơ servo nên ít hơn để đảm bảo độ chính xác.
* Nó phải có phản ứng nhanh, có thể đạt được bằng cách giữ tỷ lệ mô-men xoắn trên trọng lượng cao.
* Các đặc tính tốc độ mô-men xoắn phải là tuyến tính.
* Hoạt động bốn góc phần tư sẽ được chính xác bằng cách sử dụng các [bộ chuyển đổi khác](https://www.watelectrical.com/ac-to-dc-converter-with-working/) .
* Hoạt động ổn định, bản chất mạnh mẽ.

### Ứng dụng của động cơ DC Servo

Do khả năng điều khiển chính xác và độ chính xác của nó, động cơ Servo có rất nhiều ứng dụng, một số ứng dụng trong số đó như: Công nghiệp tự động hóa, công nghiệp robot, hàng không, công nghiệp sản xuất, đồ chơi và ô tô điều khiển bằng radio,...

### Sự khác nhau giữa động cơ Servo và động cơ DC

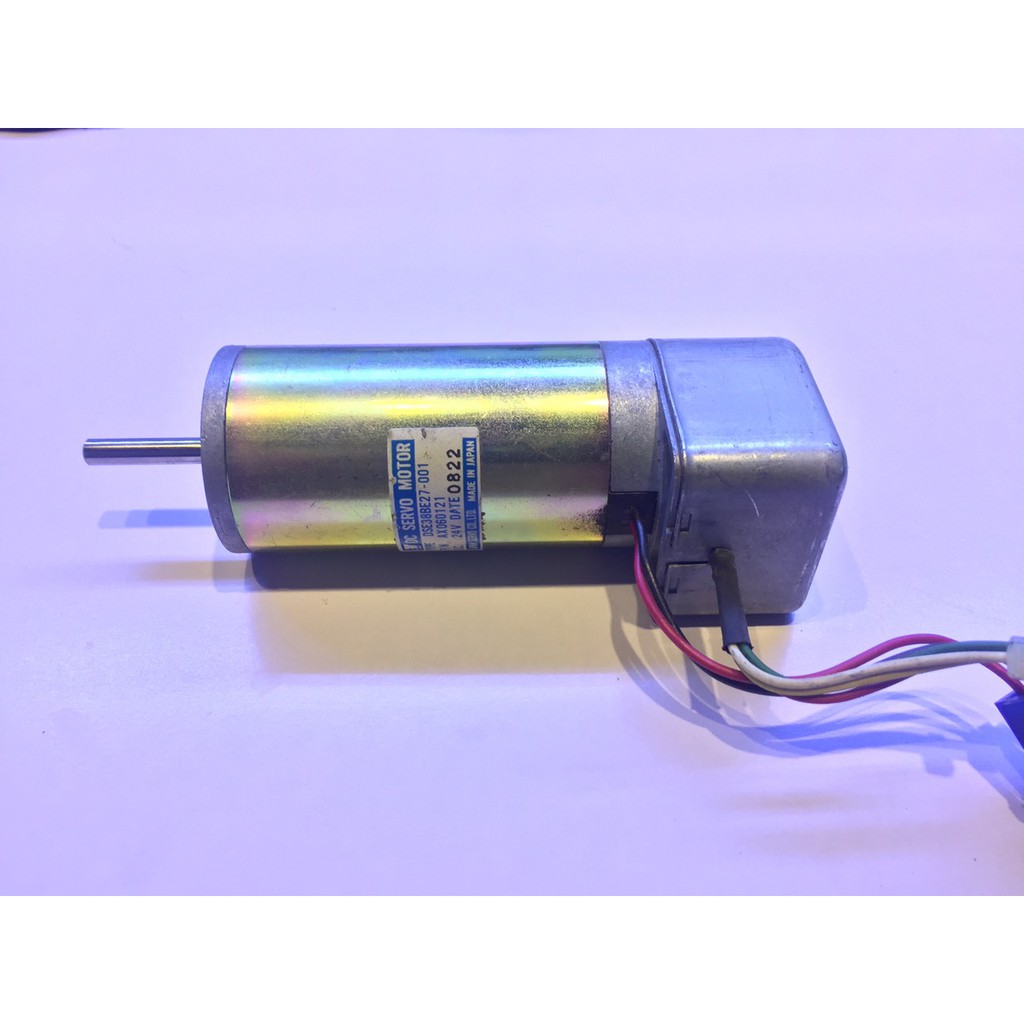
|  |  |
| --- | --- |
| **Động cơ Servo** | **Động cơ DC** |
| Động cơ Servo hoạt động dựa trên nguyên lý của hệ thống điều khiển vòng kín, trong đó động cơ DC cũng là một trong những thành phần | Động cơ điện một chiều hoạt động dựa trên nguyên tắc của Định luật lực Lorentz, tức là khi một dây dẫn mang dòng điện được đặt trong từ trường, nó sẽ chịu một lực. |
| Quán tính của động cơ Servo ít hơn | Quán tính của động cơ điện một chiều nhỏ hơn |
| Động cơ Servo phù hợp hơn cho ngành công nghiệp tự động hóa và robot | Động cơ DC có nhiều ứng dụng |
| Servo motor cần bảo trì | Động cơ DC có bản chất chắc chắn |
| Động cơ Servo đắt hơn so với động cơ DC | Động cơ DC không đắt so với động cơ Servo |
| Do hệ thống điều khiển vòng kín, động cơ Servo sẽ có độ chính xác và độ chính xác cao hơn | Đây không phải là trường hợp của động cơ DC. |

Bảng 2.1 Sự khác nhau giữa động cơ Servo và động cơ DC

### Ưu, nhược điểm của động cơ DC Servo

* Ưu điểm: Kiểm soát chính xác và độ chính xác cao, hoạt động ổn định, phản ứng nhanh, nhẹ và di động,..
* Nhược điểm: Mạch phức tạp độ tin cậy thấp hơn, do có các thành phần vòng kín như bộ khuếch đại, hộp số,..Nên rất tốn kém

### 2.4.6 Động cơ DC Servo được sử dụng trong đề tài

****

Hình 2.13 Động cơ DC Servo DSE38BE27-001

Thông số động cơ DC Servo dùng làm đề tài

* Kiểu: DSE38BE27-001
* Điện áp cấp cho động cơ: 24VDC
* Tốc độ: 4400 vòng/phút
* Điện áp cấp cho encoder: 5VDC
* Encoder: 108 xung/vòng
* Số encoder: 2 encoder đặt lệch 

\* Ghi chú: Để đơn giản và thuận tiện trong việc làm mô hình, trong bài này em sử dụng điện áp cấp cho DC Servo là 12V. Việc sử dụng điện áp bằng một nửa điện áp định mức của động cơ làm giảm tốc độ quay của động cơ xuống khoảng còn một nửa (2200 vòng/phút) so với số vòng quay khi sử dụng đúng điện áp quy định (4400 vòng/phút).

## 2.5 Encoder

### 2.5.1 Giới thiệu về Encoder



Hình 2.14 Bộ encoder trên thị trường

Encoder là thành phần quan trọng của động cơ, giúp chúng ta đọc được tốc độ và vị trí của động cơ, nhờ các xung vuông có tần số thay đổi phụ thuộc vào tốc độ của động cơ.

Trong ngành gia công cơ khí chính xác. Máy CNC là thiết bị được dùng gia công cơ khí chính xác hoàn toàn tự động. Để điều khiển và xác định các góc quay của dao hoặc bàn gá, hiển thị trên máy tính là đường thẳng hoặc góc bao nhiêu độ. Thì bên trong các cánh tay robot máy CNC được bố trí các Encoder làm nhiệm vụ trên.

### 2.5.2 Giới thiệu về Encoder

Encoder có nhiều xuất xứ và chủng loại khác nhau, CNC Khắc đá sẽ phân loại chúng theo 2 loại chính như sau: Encoder tuyệt đối (adsolute encoder) đĩa theo mã nhị phân hoặc mã Gray và Encoder tương đối (encremental encoder) có tín hiệu tăng dần theo chu kỳ.

**Encoder kiểu tuyệt đối**

****

Hình 2.15 Encoder kiểu tuyệt đối

Encoder kiểu tuyệt đối có kết cấu gồm những phần sau: Bộ phát ánh sáng (LED), đĩa mã hóa (có chứa dải băng mang tín hiệu), một bộ thu ánh sáng nhạy với ánh sáng phát ra (photosensor).

Đĩa mã hóa ở encoder tuyệt đối được chế tạo từ vật liệu trong suốt, người ta chia mặt đĩa thành các góc đều nhau và các đường tròn đồng tâm.

Các đường tròn đồng tâm và bán kính giới hạn các góc hình thành các phân tố diện tích.

Tập hợp các phân tố diện tích cùng giới hạn bởi 2 vòng tròn đồng tâm gọi là dải băng. Số dải băng tùy thuộc vào công nghệ sản xuất (chủng loại sản phẩm), ứng với một dải băng ta có một đèn LED và một bộ thu.

**Encoder kiểu tương đối**



Hình 2.16 Encoder kiểu tương đối

Về cơ bản thì Encoder kiểu tương đối đều giống nhau, chỉ khác ở đĩa mã hóa. Ở encoder tương đối thì đĩa mã hóa gồm 1 dải băng tạo xung. Ở dải băng này được chia làm nhiều lỗ bằng nhau và cách đều nhau (có thể chất liệu trong suốt để ánh sáng chiếu qua).

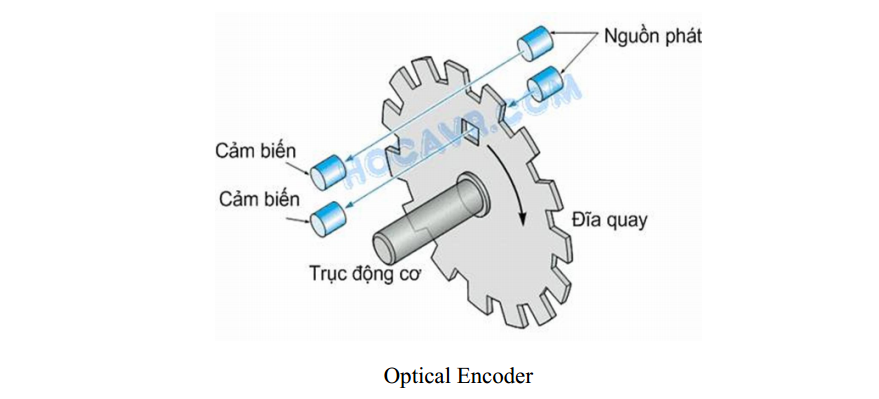
Khi đĩa quay và ánh sáng chiếu được 1 lỗ thì bộ thu (photosensor) nhận được tín hiệu từ đèn LED thì encoder sẽ ghi nhận giá trị lên 1 biến đếm.

Ngoài ra encoder còn được phân loại theo kiểu giao tiếp,v.v…

### 2.5.3 Nguyên lý hoạt động

Để điều khiển số vòng quay hay vận tốc động cơ thì chúng ta nhất thiết phải đọc được góc quay của motor. Một số phương pháp có thể được dùng để xác định góc quay của motor bao gồm Tachometer (thật ra Tachometer đo  vận tốc quay dùng biến trở  xoay, hoặc dùng encoder). Trong đó 2 phương pháp đầu tiên là phương pháp Analog và dùng Optical encoder (Encoder quang thuộc nhóm phương pháp digital). Hệ thống Optical encoder bao gồm một nguồn phát quang (thường là hồng ngoại – infrared), một cảm biến quang đọc tín hiệu từ đầu phát và một đĩa có chia rãnh để cắt đường tín hiệu quang phát ra từ mắt phát quang.

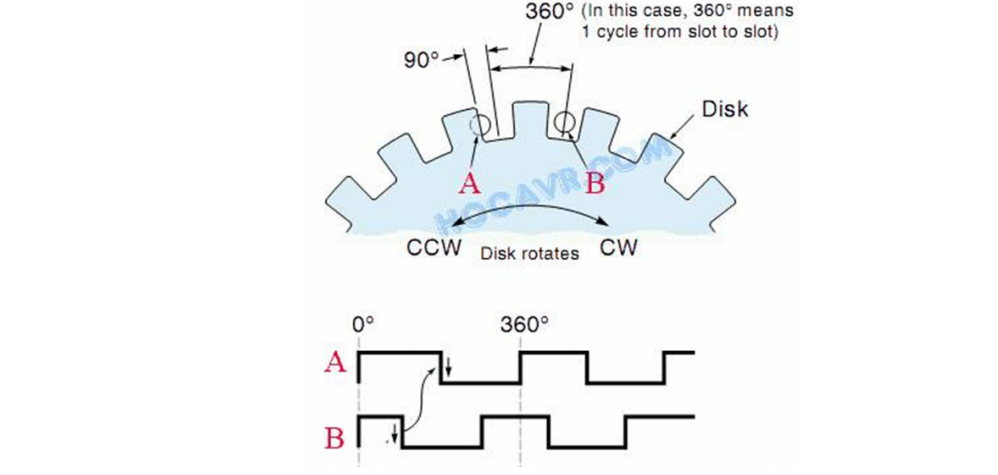
Encoder thường có 3 kênh (3 ngõ ra) bao gồm kênh A, kênh B và kênh I (Index). Trong hình 2 bạn thấy hãy chú ý một lỗ nhỏ bên phía trong của đĩa quay và một cặp phat-thu dành riêng cho lỗ nhỏ này. Đó là kênh I của encoder. Cữ mỗi lần motor quay được một vòng, lỗ nhỏ xuất hiện tại vị trí của cặp phát - thu, hồng ngoại từ nguồn phát sẽ xuyên qua lỗ nhỏ đến cảm biến quang, một tín hiệu x hiện trên cảm biến.



Hình 2.17 Mã hóa quang học

Như thế kênh I xuất hiện một “xung” mỗi vòng quay của motor. Bên ngoài đĩa quay được chia thành các rãnh nhỏ và một cặp thu-phát khác dành cho các rãnh này. Đây là kênh A của encoder, hoạt động của kênh A cũng tương tự kênh I, điểm khác nhau là trong 1 vòng quay của motor, có N “xung” xuất hiện trên kênh A. N là số rãnh trên đĩa và được gọi là độ phân giải (resolution) của encoder. Mỗi loại encoder có độ phân giải khác nhau, có khi trên mỗi đĩa chĩ có vài rãnh nhưng cũng có trường hợp đến hàng nghìn rãnh được chia. Để điều khiển động cơ, bạn phải biết độ phân giải của encoder đang dùng.

Độ phân giải ảnh hưởng đến độ chính xác điều khiển và cả phương pháp điều khiển. Không được vẽ trong hình 2, tuy nhiên trên các encoder còn có một cặp thu phát khác được đặt trên cùng đường tròn với kênh A nhưng lệch một chút (lệch M+0,5 rãnh), đây là kênh B của encoder. Tín hiệu xung từ kênh B có cùng tần số với kênh A nhưng lệch pha 90o. Bằng cách phối hợp kênh A và B người đọc sẽ biết chiều quay của động cơ.



Hình 2.18 Hai kênh A và B lệch pha trong encoder

 Hình  trên  cùng   thể  hiện  sự  bộ  trí  của  2  cảm  biến kênh  A  và B lệch pha nhau. Khi cảm biến A bắt đầu bị che thì cảm biến B hoàn toàn nhận được hồng ngoại xuyên qua, và ngược lại. Hình thấp là dạng xung ngõ ra trên 2 kênh.

Xét trường hợp motor quay cùng chiều kim đồng hồ, tín hiệu “đi” từ trái sang phải. Bạn hãy quan sát lúc tín hiệu A chuyển từ mức xuống thấp (cạnh xuống) thì kênh B đang ở mức thấp. Ngược lại, nếu động cơ quay ngược chiều kim đồng hồ, tín hiệu “đi” từ phải qua trái.

Lúc này, tại cạnh xuống của kênh A thì kênh B đang ở mức cao. Như vậy, bằng cách phối hợp 2 kênh A và B chúng ta không những xác định được góc quay (thông qua số xung) mà còn biết được chiều quay của động cơ (thông qua mức của kênh B ở cạnh xuống của kênh A).

## 2.6 Ngôn ngữ Arduino

Đây là một bo mạch đã được xử lý để thiết kế các tương tác với các thiết bị phần cứng như động cơ robot, cảm biến, ..

Đặc điểm của [ngôn ngữ lập trình](https://magenest.com/vi/ngon-ngu-lap-trinh/) này nằm ở khả năng mở rộng của ngôn ngữ. Arduino là bo mạch đã được xử lý và chuẩn hóa cho phép bạn có thể mở rộng bằng các bo mạch khác. Tùy vào mục đích sử dụng của bạn, bạn có thể cắm chồng các bo mạch mở rộng (shield) lên bo mạch Arduino. Bạn có thể dễ dàng sáng tạo và phát minh ra các ứng dụng tùy ý nhờ vào khả năng mở rộng của ngôn ngữ lập trình Arduino.

Điều khiến ngôn ngữ lập trình Arduino trở nên phổ biến và được ưa chuộng bởi cộng đồng lập trình viên chính là việc đây là một ngôn ngữ cực kỳ dễ học (giống như C/C++). Khả năng mở rộng nhờ vào các bo mạch mở rộng cũng khiến ngôn ngữ này trở thành ưu tiên đối với các lập trình viên. Ví dụ như khi bạn muốn thiết kế khả năng kết nối Internet, bạn có thể sử dụng Ethernet shield, muốn thiết kế khả năng điều khiển động cơ thì có Motor shield, và khả năng kết nối nhận tin nhắn thì có GSM shield, ..

Như vậy với đặc điểm đã được xử lý và chuẩn hóa, Arduino là lựa chọn tối ưu cho việc thiết kế bất cứ ứng dụng phần cứng nào mà bạn cần một cách đơn giản và dễ dàng.

Một số tính năng thú vị của ngôn ngữ Arduino có thể kể đến như: Thiết kế bộ xử lý trung tâm của các loại robot nhờ vào khả năng xử lý dữ liệu của các thiết bị cảm biến, động cơ, thiết kế các trò chơi tương tác như: Tetrix, Mario,..Ứng dụng trong máy bay không người lái.

Để lập trình bằng Arduino, chỉ cần tải xuống IDE , dây kết nối USB loại A-B và một bo mạch Arduino.

Ngôn ngữ lập trình của Arduino chính là C/C++, những khác với lập trình trực tiếp với vi điều khiển, lập trình Arduino dễ dàng hơn rất nhiều nhờ vào các bo mạch mở rộng (library) đa dạng. Tùy thuộc vào loại ứng dụng bạn cần thiết kế, một số loại bo mạch mở rộng hiện có sẵn trên website. Những gì cần làm chỉ đơn giản là kết hợp các bo mạch này để tạo ra ứng dụng như mong muốn của mình.

## 2.7 PID

### 2.7.1 Giới thiệu PID

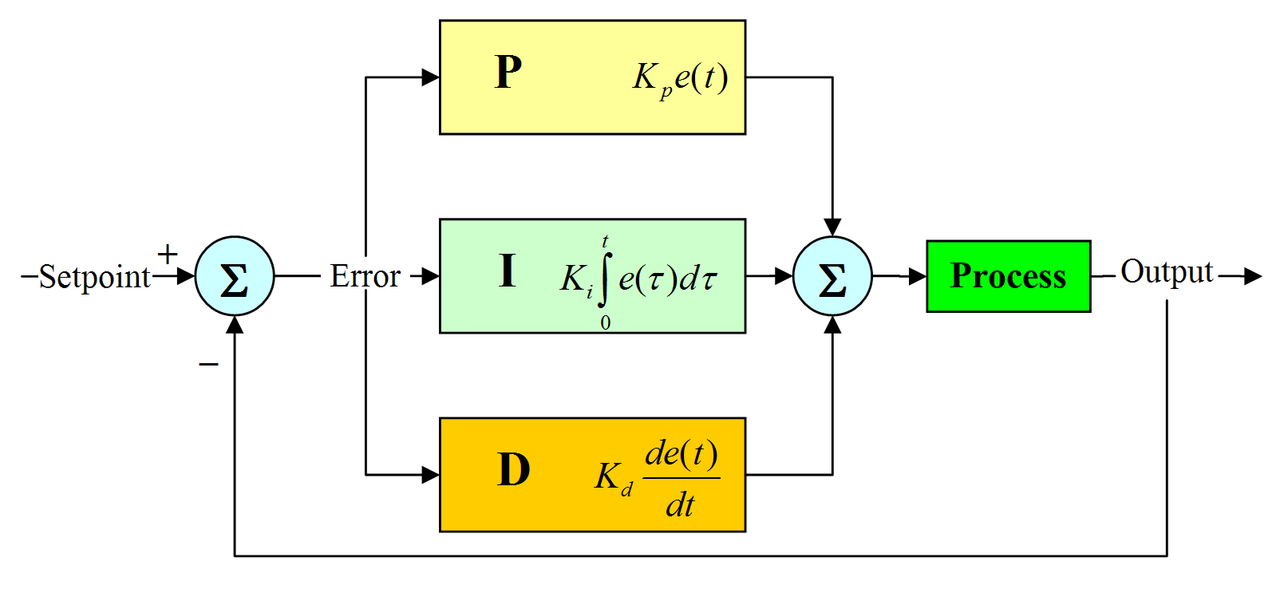
Bộ điều khiển PID (A proportional integral derivative controller) là bộ điều khiển sử dụng kỹ thuật điều khiển theo vòng lặp dụng kỹ thuật điều khiển theo vòng lặp có hồi tiếp được sử dụng rộng rãi trong các hệ thống điều khiển tự động.

Một bộ điều khiển PID cố gắng hiệu chỉnh sai lệch giữa tín hiệu ngõ ra và ngõ vào sau đó đưa ra một một tín hiệu điều khiển để điều chỉnh quá trình cho phù hợp.

Bộ điều khiển kinh điển PID đã và đang được sử dụng rộng rãi để điều khiển các đối tượng SISO bởi vì tính đơn giản của nó cả về cấu trúc lẫn nguyên lý làm việc. Bộ điều chỉnh này làm việc rất tốt trong các hệ thống có quán tính lớn như điều khiển tốc độ, điều khiển mức,... và trong các hệ điều khiển tuyến tính hay có mức độ phi tuyến thấp.

PID là một trong những lý thuyết cổ điển và cũ nhất dùng cho điều khiển tuy nhiên nó vẫn ứng dụng rộng rãi cho đến ngày nay.

Sơ đồ khối của bộ điều khiển PID:



Hình 2.19 Sơ đồ khối của bộ điều khiển PID

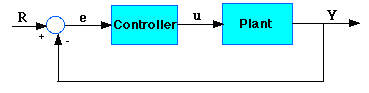
### 2.7.2 Hàm truyền



Bộ điều khiển gồm có 3 thành phần:

* Kp: hệ số tỷ lệ
* Ki : hệ số tích phân
* Kd: hệ số vi phân

Xét 1 hệ thống có sơ đồ khối như sau:

****

Hình 2.20 Sơ đồ khối của một hệ thống

Plant: đối tượng cần điều khiển

Controller: đưa tín hiệu điều khiển đối tượng, được thiết kế để hệ thống đạt đáp ứng mong muốn .

Biến e là thành phần sai lệch, là hiệu giữa giá trị tín hiệu vào mong muốn và tín hiệu ra thực tế. Tín hiệu sai lệch (e) sẽ đưa tới bộ PID, và bộ điều khiển tính toán cả thành phần tích phân lẫn vi phân của (e). Tín hiệu ra (u) của bộ điều khiển bằng:



Lúc này đối tượng điều khiển có tín hiệu vào là (u), và tín hiệu ra là (Y). (Y) được hồi tiếp về bằng các cảm biến để tiếp tục tính sai lệch (e). Và bộ điều khiển lại tiếp tục như trên.

### 2.7.3 Đặc tính bộ điều khiển PID

Thành phần tỉ lệ (Kp) có tác dụng làm tăng tốc độ đáp ứng của hệ, và làm giảm chứ không triệt tiêu sai số xác lập của hệ (steady-state error).

Thành phần tích phân (Ki) có tác dụng triệt tiêu sai số xác lập nhưng có thể làm giảm tốc độ đáp ứng của hệ.

Thành phần vi phân (Kd) làm tăng độ ổn định hệ thống, giảm độ vọt lố và cải thiện tốc độ đáp ứng của hệ.

Ảnh hưởng của các thành phần Kp, Ki, Kd đối với hệ kín được tóm tắt trong bảng sau:

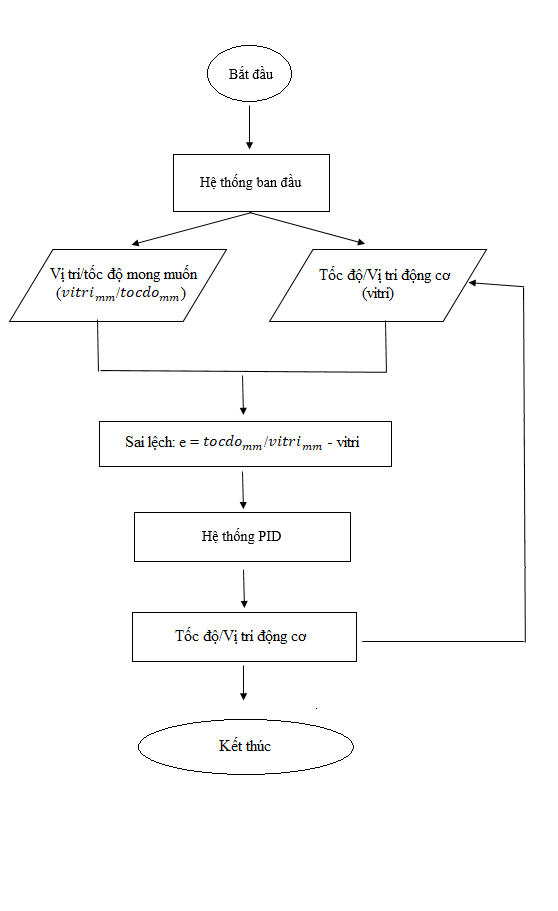
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đáp ứng vòng kín | Thời gian lên | Vọt lố | Thời gian xác lập | Sai số xác lập |
| KP | Giảm | Tăng | Thay đổi nhỏ | Giảm |
| KI | Giảm | Tăng | Tăng | Thay đổi nhỏ |
| KD | Thay đổi nhỏ | Giảm | Giảm | Thay đổi nhỏ |

Bảng 2.2 Ảnh hưởng của các thành phần Kp, Ki, Kd

Lưu ý rằng quan hệ này không phải chính xác tuyệt đối vì Kp, Ki và Kd còn phụ thuộc vào nhau. Trên thực tế, thay đổi một thành phần có thể ảnh hưởng đến hai thành phần còn lại. Vì vậy bảng trên chỉ có tác dụng tham khảo khi chọn Kp, Ki, Kd.

# CHƯƠNG 3: CƠ SỞ THỰC HIỆN

## 3.1 Lưu đồ giải thuật



Hình 3.1 Lưu đồ giải thuật

## 3.2 Đọc vị trí động cơ

Để điều khiển vị trí của động cơ đầu tiên phải đọc được vị trí của động cơ, đặt vị trí ban đầu là 0.

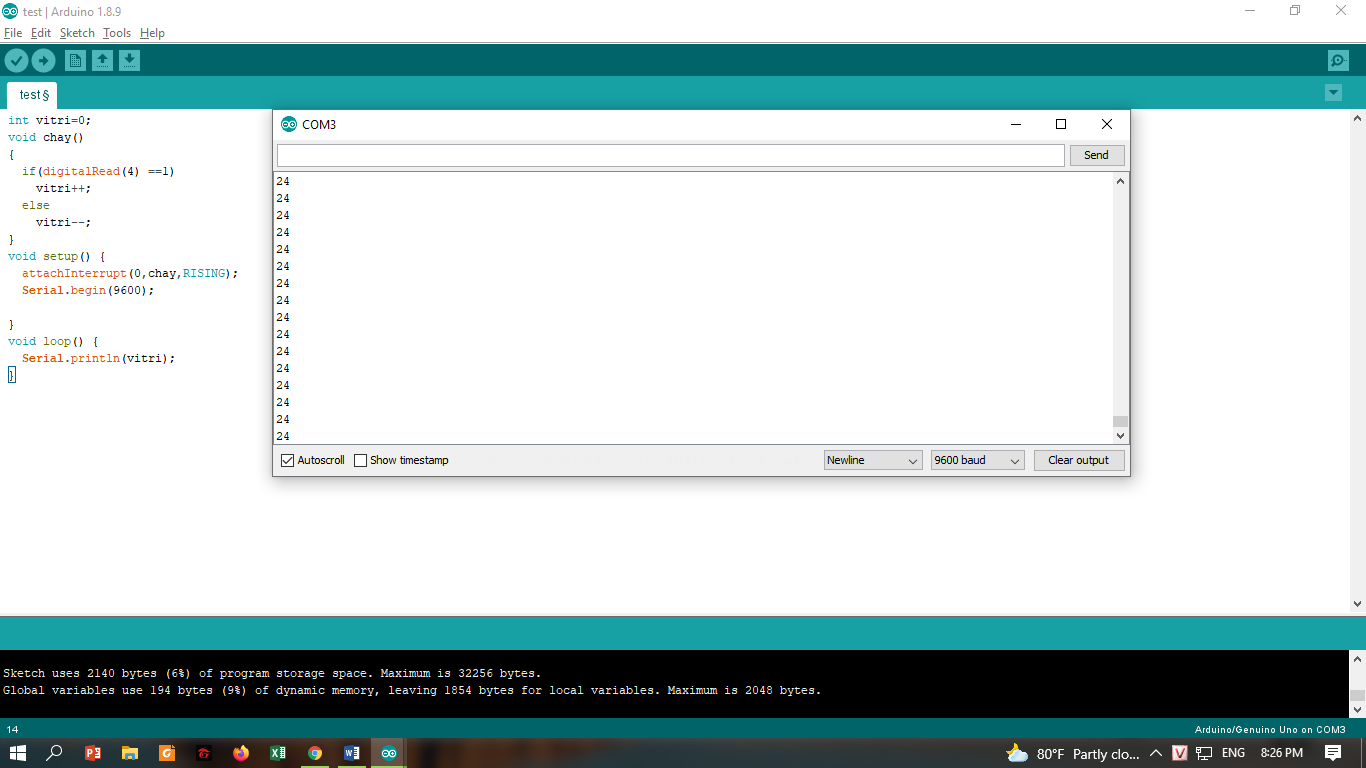
Tiếp đến chúng ta dùng hàm ngắt ngoài để khi có tín hiệu thì nó sẽ làm hành động tiếp theo, có cú pháp: attachInterrupt(interrupt, ISR, mode), trong đó:

* interrupt: Số thứ tự của ngắt. Trên Arduino Uno, có 2 ngắt với số thứ tự là 0 và 1. Em dùng ngắt số 0 vì đã nối với chân digital số 2.
* ISR: Tên hàm sẽ gọi khi có sự kiện ngắt sinh ra, ở đây ta dùng hàm “chay”
* mode: Kiểu kích hoạt ngắt là RISING: kích hoạt khi trạng thái của chân digital chuyển từ mức điện áp thấp sang mức điện áp cao.

Sau khi có tín hiệu ở chân digital số 2, sẽ xét sang hàm “chay”, hàm “chay” được viết rằng: xét chân digital số 4 nếu ở mức cao thì tăng vị trí lên còn mức thấp thì ngược lại.



Hình 3.2 Có được vị trí của động cơ khi quay âm (cùng chiều kim đồng hồ)



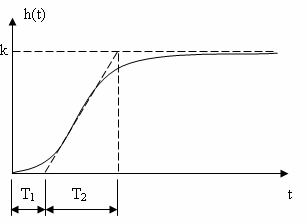
Hình 3.3 Có được vị trí của động cơ quay dương (ngược chiều kim đồng hồ)

## 3.3 Các phương pháp chọn tham số cho bộ diều khiển PID

### 3.3.1 Phương pháp Ziegler-Nichols**:**

Là phương pháp thực nghiệm để xác định tham số của bộ điều khiển P, PI hoặc PID bằng cách dựa vào đáp ứng quá độ của đối tượng điều khiển. Tùy theo đặc điểm của từng đối tượng điều khiển mà Ziegler-Nichols đã đưa ra hai phương pháp lựa chọn tham số:

**Phương pháp Ziegler-Nichols thứ nhất**: phương pháp này áp dụng cho các đối tượng có đáp ứng đối với tín hiệu vào là hàm nấc có dạng chữ S như điều khiển nhiệt độ lò nhiệt, tốc độ động cơ,..



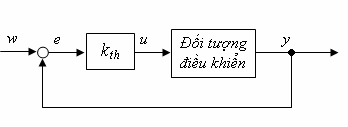
Hình 3.4 Đáp ứng nấc của hệ hở có dạng S

Thông số của các bộ điều khiển được chọn theo bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | kp | TI | TD |
| P | *T2/(k.T1)* | *-* | *-* |
| PI | *0,9T2/(k.T1)* | *T1/0,3* | *-* |
| PID | *1,2T2/(k.T1)* | *2T1* | *0,5T1* |

Bảng 3.1 Các tham số PID theo phương pháp Ziegler-Nichols thứ nhất

**Phương pháp Ziegler-Nichols thứ hai**: Phương pháp này áp dụng cho đối tượng có khâu tích phân lý tưởng như mực chất lỏng trong bồn chứa, vị trí hệ truyền động dùng động cơ… Đáp ứng quá độ của hệ hở của đối tượng tăng đến vô cùng. Phương pháp này được thực hiện như sau:

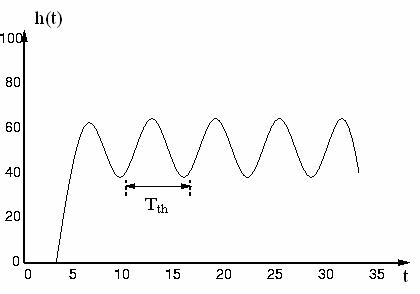


Hình 3.5 Xác định hằng số khuếch đại tới hạn

Thay bộ điều khiển PID trong hệ kín bằng bộ khuếch đại (hình 3.4).

Tăng hệ số khuếch đại tới giá trị tới hạn kth để hệ kín ở chế độ biên giới ổn định tức là h(t) có dạng dao động điều hòa.

Xác định chu kỳ Tth của dao động.



Hình 3. 6 Đáp ứng nấc của hệ kín khi k = kth

Thông số của các bộ điều khiển được chọn theo bảng sau:

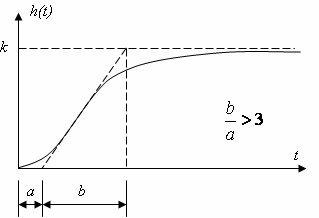
|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | *kp* | *TI* | *TD* |
| P | *0,5kth* | *-* | *-* |
| PI | *0,45kth* | *0,85Tth* | *-* |
| PID | *0,6kth* | *0,5Tth* | *0,125Tth* |

Bảng 3.2 Các tham số PID theo phương pháp Ziegler-Nichols thứ hai

### 3.3.2 Phương pháp Chien-Hrones-Reswick**:**

Phương pháp này cũng áp dụng cho các đối tượng có đáp ứng đối với tín hiệu vào là hàm nấc có dạng chữ S (hình 3.6) nhưng có thêm điều kiện:





Hình 3.7 Đáp ứng nấc của hệ thích hợp cho phương pháp Chien-Hrones-Reswick

Phương pháp Chien-Hrones-Reswick đưa ra bốn cách xác định tham số bộ điều khiển cho bốn yêu cầu chất lượng khác nhau:

* Yêu cầu tối ưu theo nhiễu và hệ kín không có độ quá điều chỉnh:

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | kp | TI | TD |
| P | *3b/10ak* | *-* | *-* |
| PI | *6b/10ak* | *4a* | *-* |
| PID | *19b/20ak* | *12a/5* | *21a/50* |

Bảng 3.3 Các tham số PID theo phương pháp Chien-Hrones-Reswick 1

* Yêu cầu tối ưu theo nhiễu và hệ kín có độ quá điều chỉnh h không vượt quá 20% so với 

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | *kp* | *TI* | *TD* |
| P | *7b/10ak* | *-* | *-* |
| PI | *7b/10ak* | *23a/10* | *-* |
| PID | *6b/5ak* | *2a* | *21a/50* |

Bảng 3.4 Các tham số PID theo phương pháp Chien-Hrones-Reswick 2

* Yêu cầu tối ưu theo tín hiệu đặt trước và hệ kín không có độ quá điều chỉnh:

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | *kp* | *TI* | *TD* |
| P | *3b/10ak* | *-* | *-* |
| PI | *7b/20ak* | *6b/5* | *-* |
| PID | *3b/5ak* | *B* | *a/2* |

Bảng 3.5 Các tham số PID theo phương pháp Chien-Hrones-Reswick 3

* Yêu cầu tối ưu theo nhiễu và hệ kín có độ quá điều chỉnh h không vượt quá 20% so với 

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số  BĐK | *kp* | *TI* | *TD* |
| P | *7b/10ak* | *-* | *-* |
| PI | *6b/5ak* | *B* | *-* |
| PID | *19b/20ak* | *27b/20* | *47a/100* |

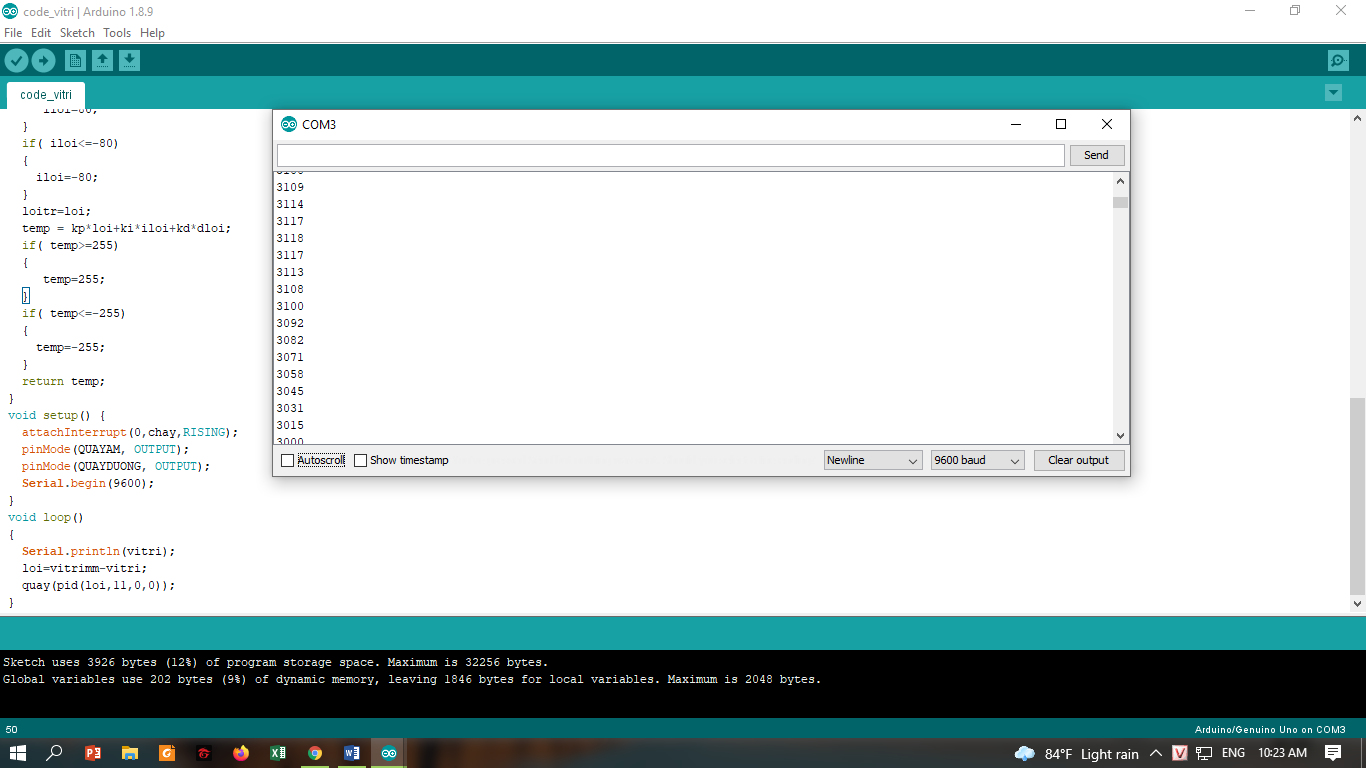
Bảng 3.6 Các tham số PID theo phương pháp Chien-Hrones-Reswick 4

### 3.3.3 Phương pháp chỉnh định bằng thực nghiệm (thử và sai)**:**

Có rất nhiều phương pháp xác định tham số PID, tuy nhiên xác định tham số bằng thực nghiệm vẫn được dùng nhiều nhất vì nó đơn giản, hiệu quả trong phạm vi ứng dụng thực tế, phương pháp được thực hiện như sau:

* Đặt  Ki= Kd = 0. Tăng đến khi hệ thống ổn định giá trị điều chỉnh gần với giá trị mong muốn, chấp nhận vọt lố nhỏ
* Nếu chưa tới giá trị tới hạn thì tăng Ki
* Nếu có vọt lố thì tăng giá trị Kd

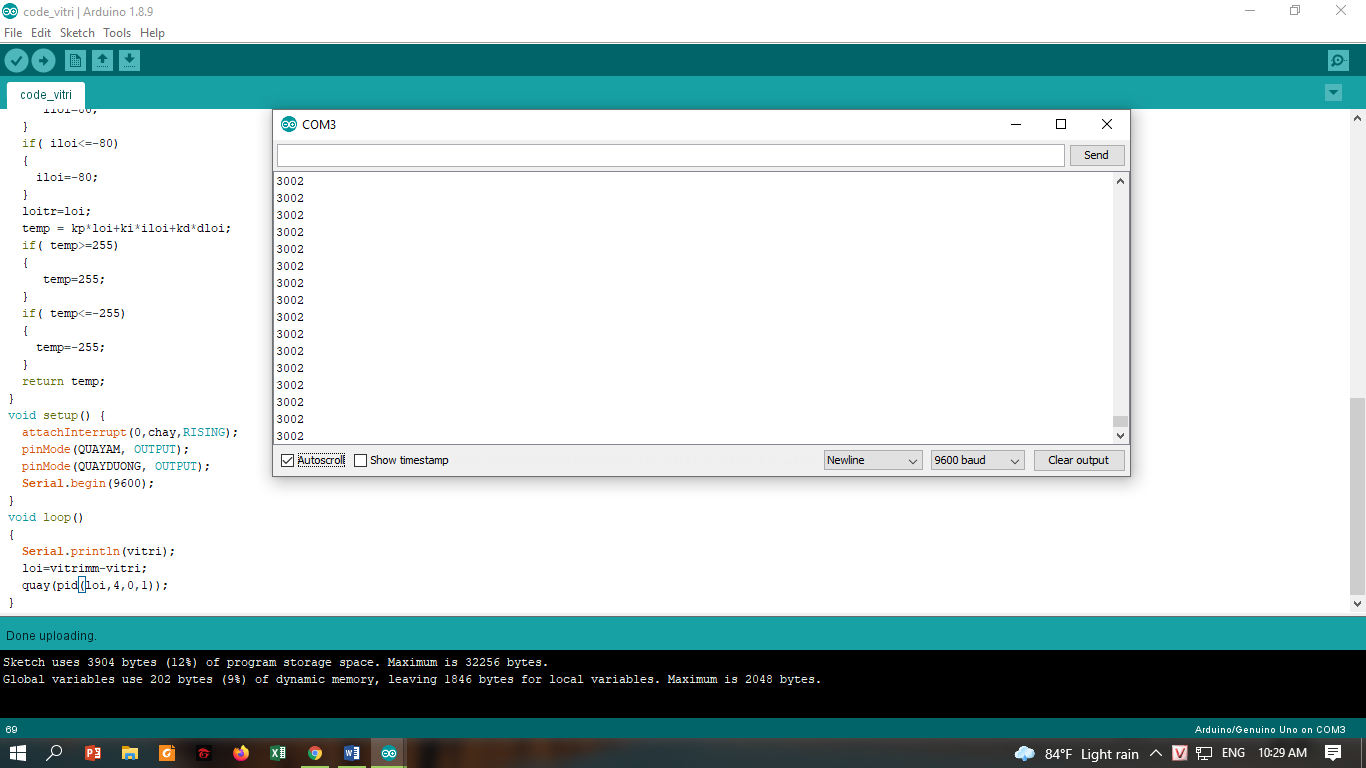
Đầu tiên đặt Ki và Kd = 0, tăng Kp đến khi có dao động:



Hình 3.8 Thực nghiệm đặt Ki và Kd = 0, tăng Kp đến khi có dao động

\*Nhận xét: Hệ thống đến vị trí mong muốn nhanh, bắt đầu dao dộng và rung lắc, chưa tới vị trí mong muốn.

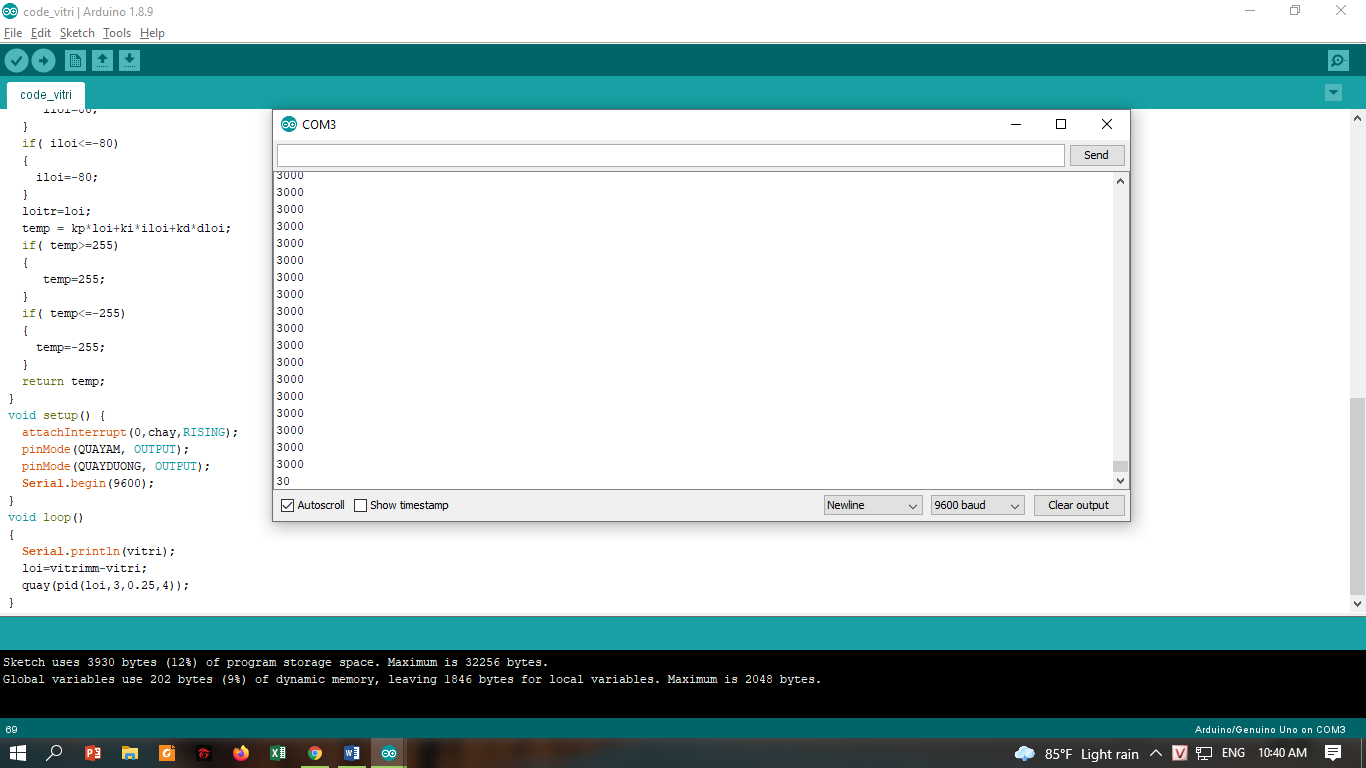
Tiếp theo giảm hệ số Kp và tăng hệ số Kd lên:



Hình 3.9 Thực nghiệm giảm hệ số Kp và tăng hệ số Kd

\*Nhận xét: Hệ thống đến vị trí mong muốn chậm hơn và hệ thống bắt đầu bớt hiện tượng rung và dao động lại, nhưng vẫn chưa đạt được chính xác vị trí mong muốn.

Tiếp theo đó là tăng hệ số Ki và hiệu chỉnh lại Kp, Kd.



Hình 3.10 Thực nghiệm tăng hệ số Ki và điều chỉnh hệ số Kp, Kd

\*Nhận xét: Hệ thống đã đến được vị trí mong muốn và không còn dao động nữa.

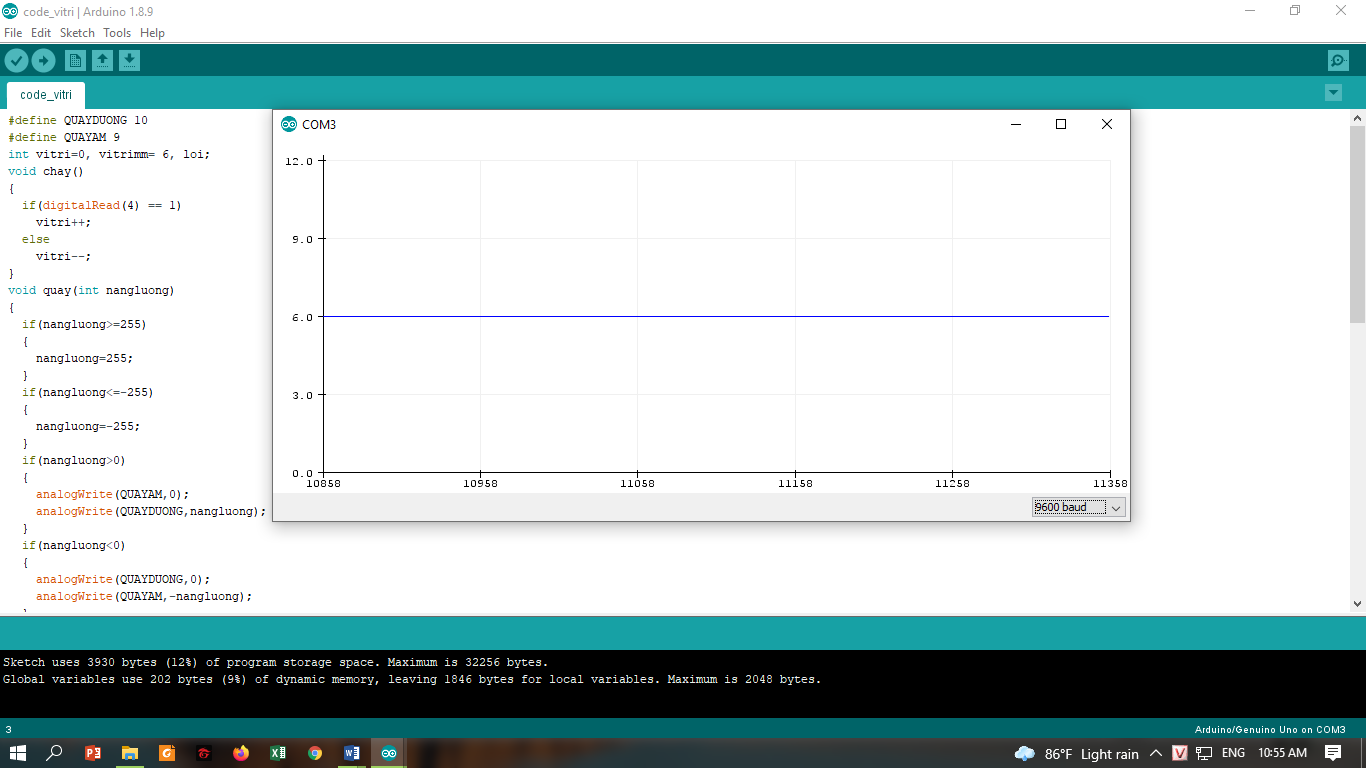
# CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

## 4.1. Thực nghiệm đánh giá độ chính xác của vị trí động cơ:

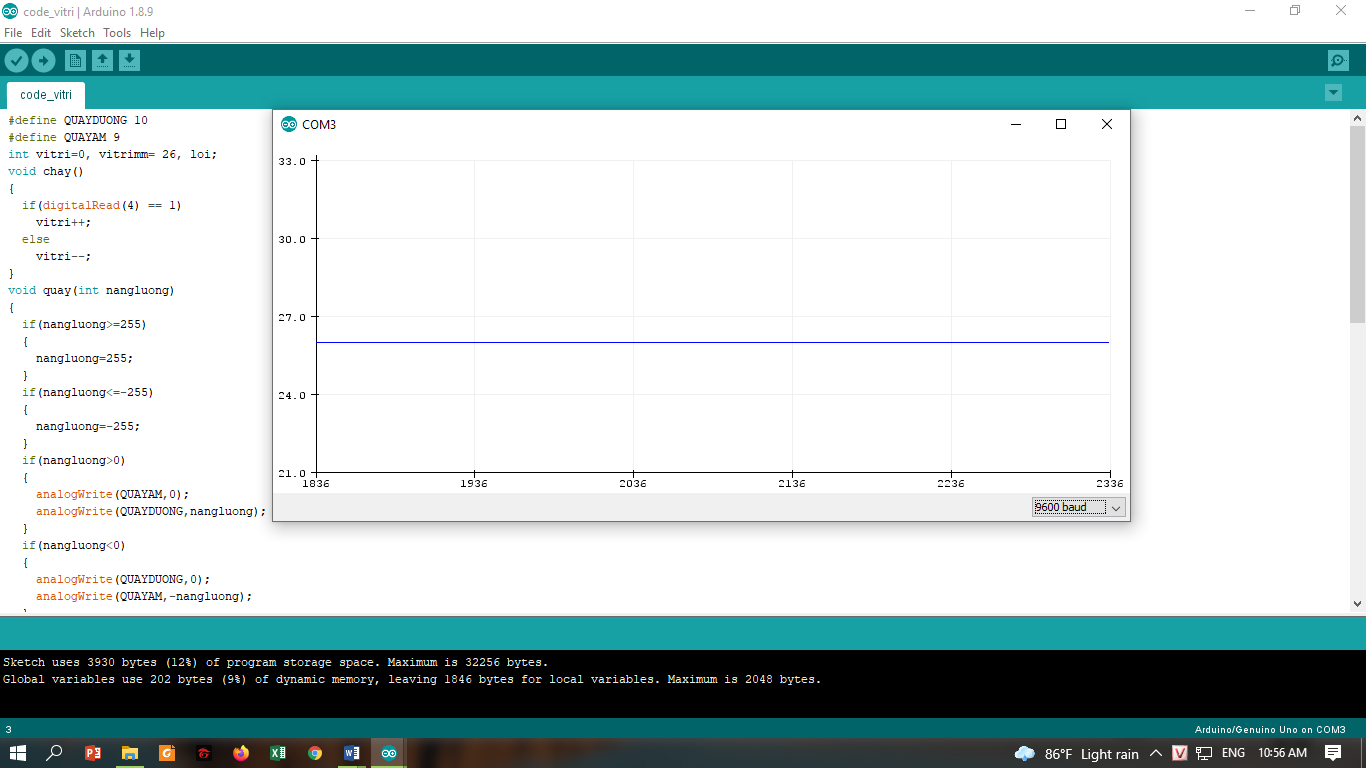
|  |  |
| --- | --- |
| Vị trí mong muốn (xung) | Độ chính xác |
| 6 | 100% |
| 26 | 99% |
| 260 | 100% |
| 2600 | 100% |
| 26000 | 100% |
| -6 | 100% |
| -26 | 99% |
| -260 | 100% |
| -2600 | 100% |
| -26000 | 100% |

Bảng 4.1 Bảng đánh giá thực nghiệm độ chính xác điều khiển vị trí

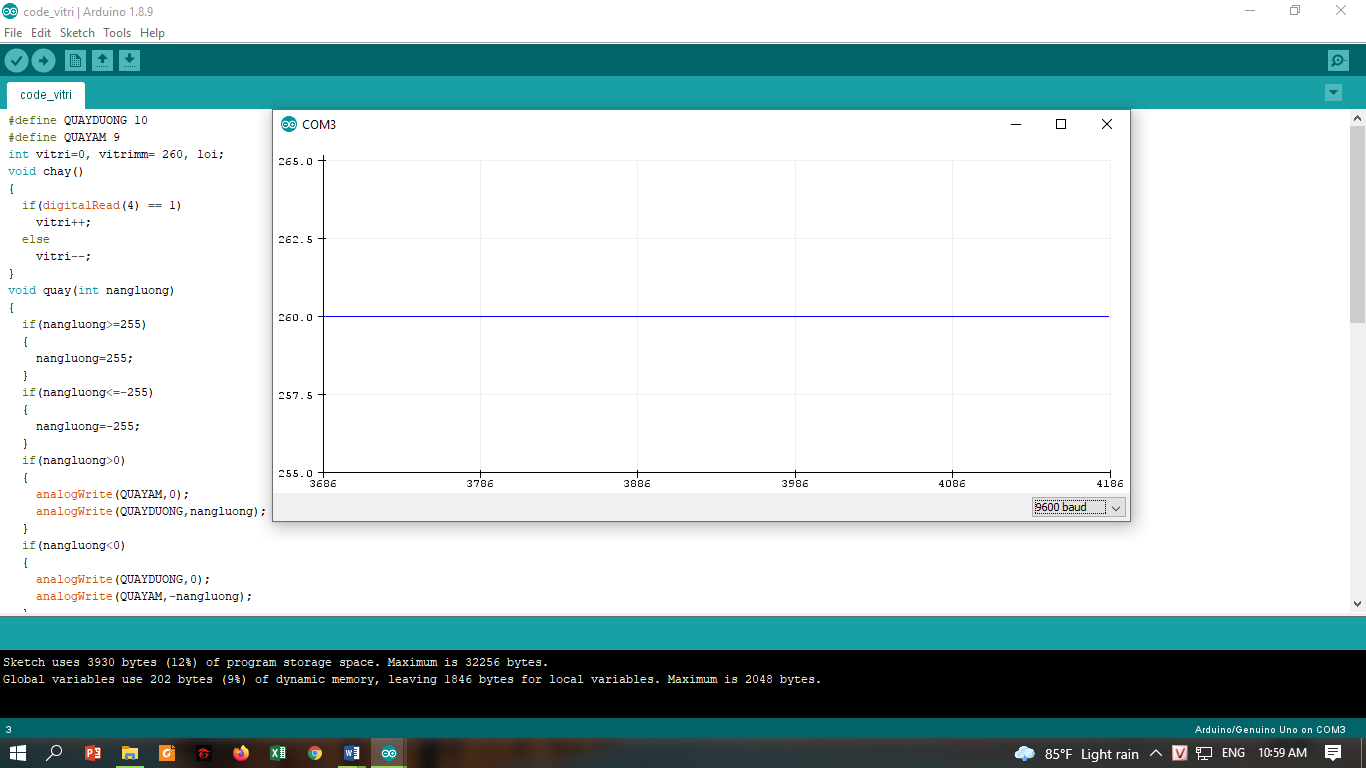
## 4.2. Kết quả điều khiển vị trí mong muốn động cơ:



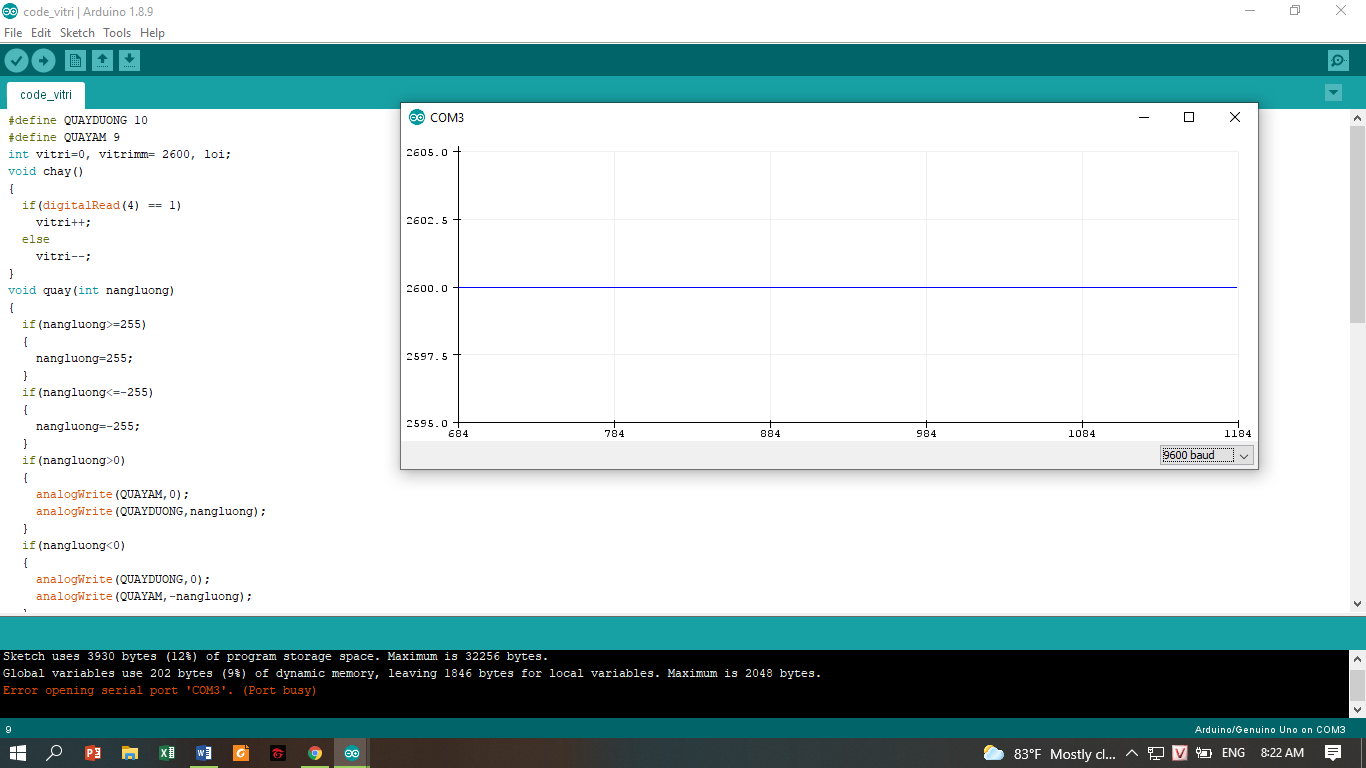
Hình 4.1 Kết quả khi đặt vị trí mong muốn là vị trí 6 xung



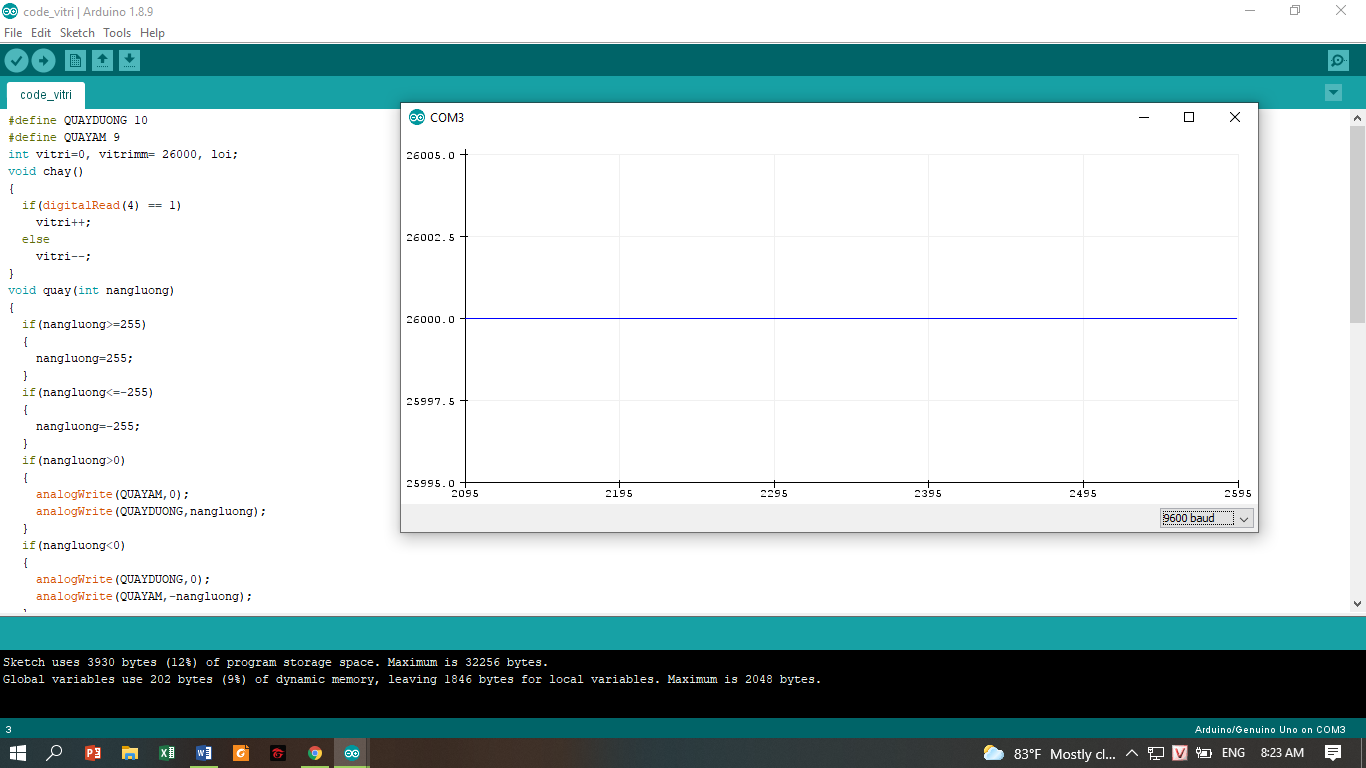
Hình 4.2 Kết quả khi đặt vị trí mong muốn là 26 xung



Hình 4.3 Kết quả khi đặt vị trí mong muốn là 260 xung



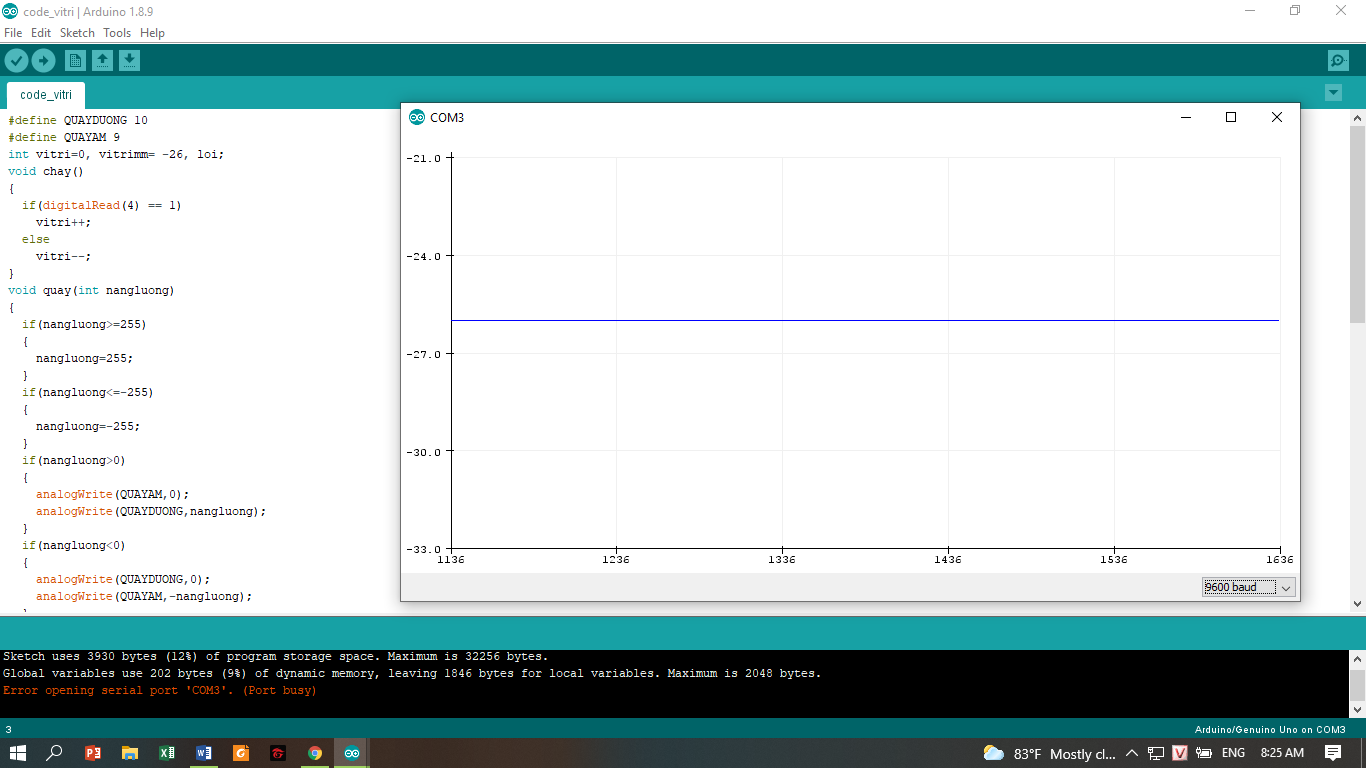
Hình 4.4 Kết quả khi đặt vị trí mong muốn là 2600 xung



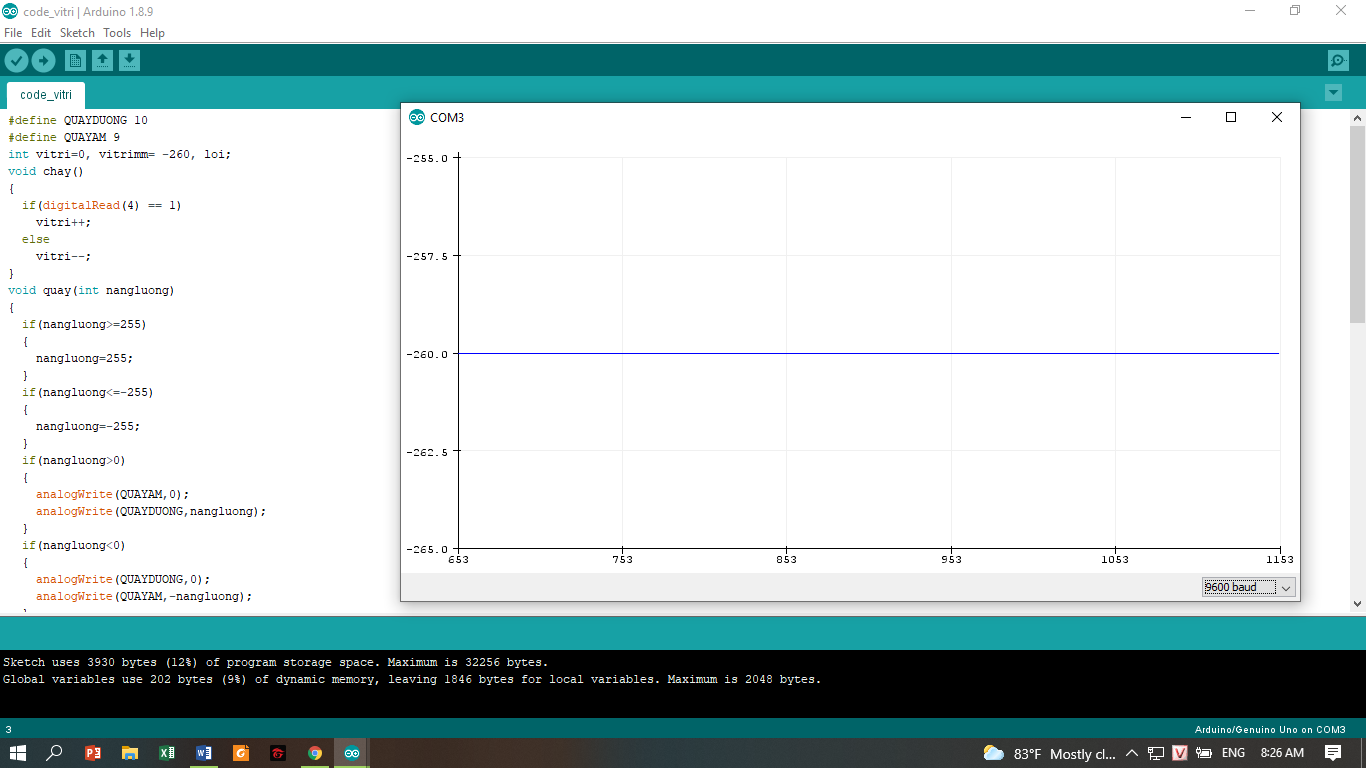
Hình 4.5 Kết quả khi đặt vị trí mong muốn là 26000 xung



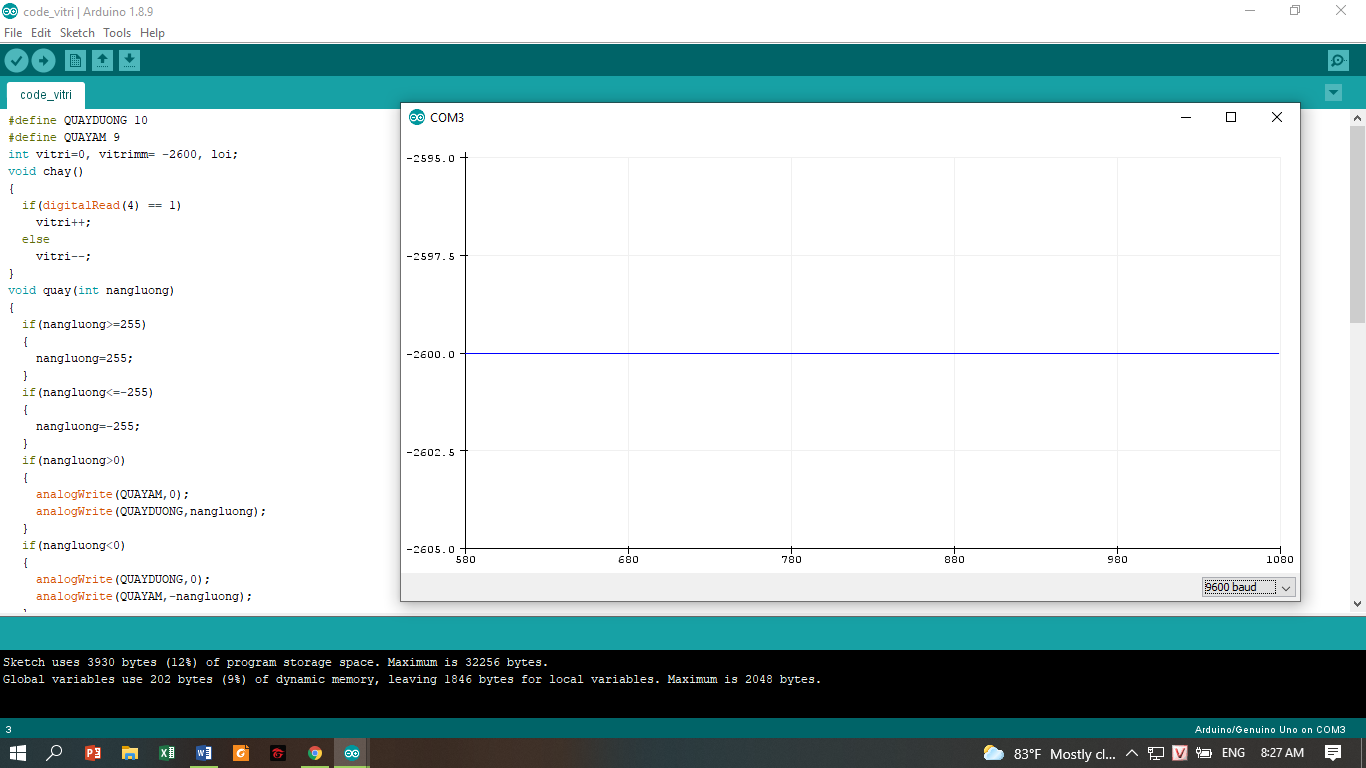
Hình 4.6 Kết quả khi đặt vị trí mong muốn là -6 xung



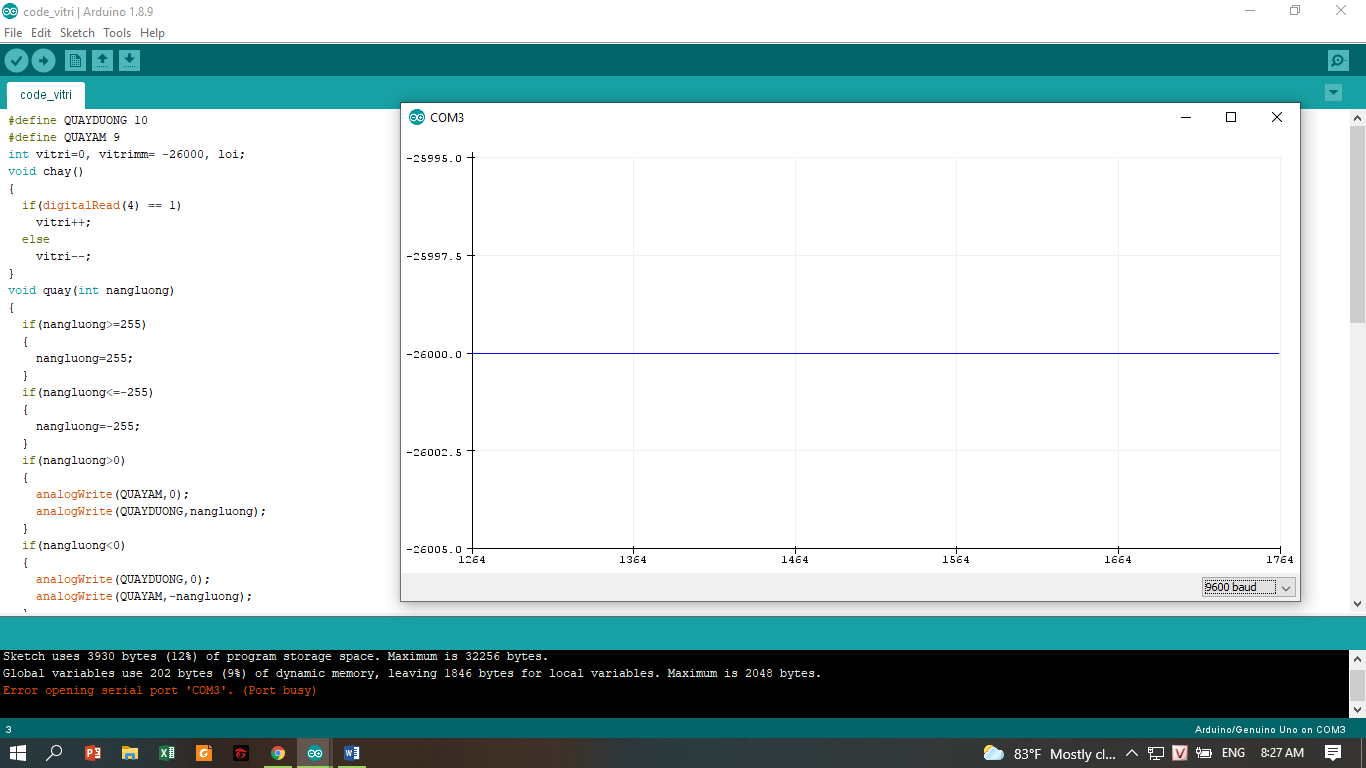
Hình 4.7 Kết quả khi đặt vị trí mong muốn là -26 xung



Hình 4.8 Kết quả khi đặt vị trí mong muốn là -260 xung



Hình 4.9 Kết quả khi đặt vị trí mong muốn là -2600 xung



Hình 4.10 Kết quả khi đặt vị trí mong muốn là -26000 xung

## 4.3. Thực nghiệm đánh giá độ chính xác của tốc độ động cơ:

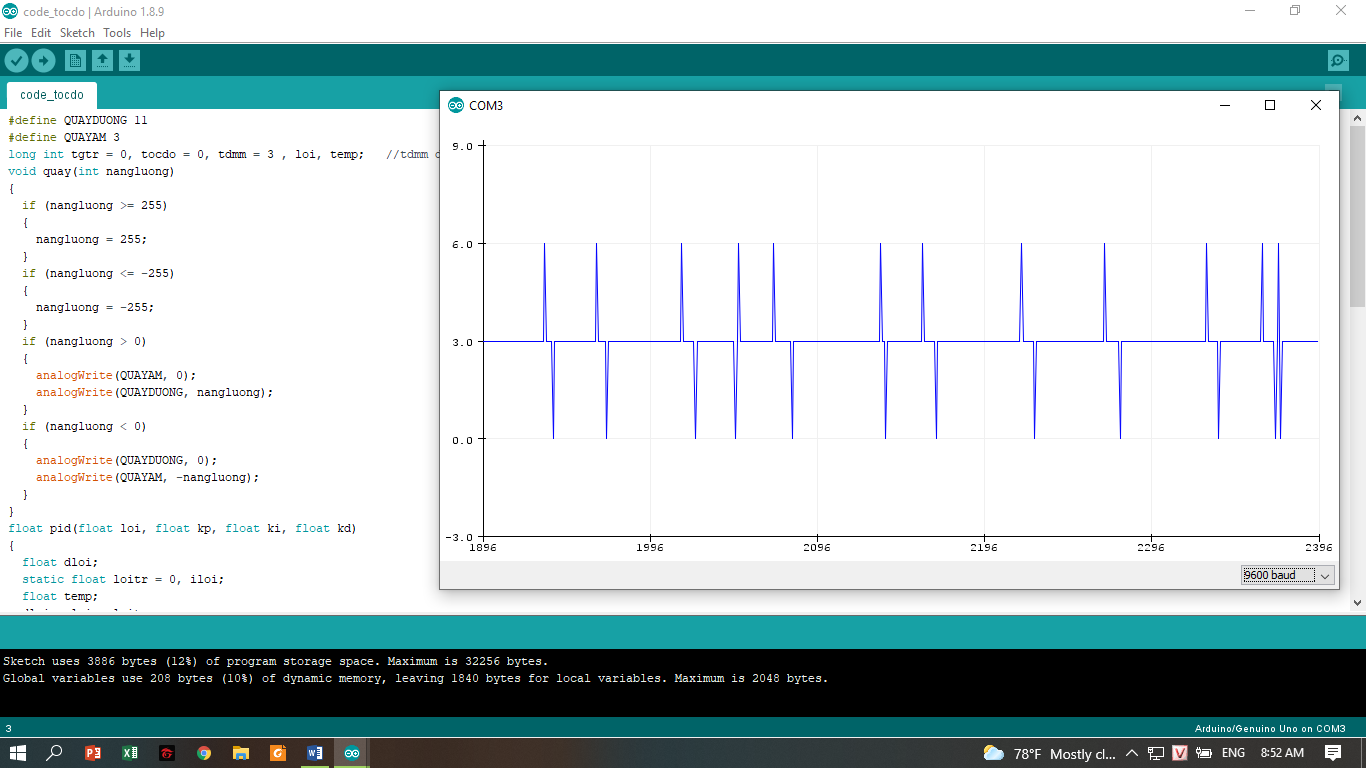
|  |  |
| --- | --- |
| Tốc độ mong muốn | Độ chính xác |
| 1 vòng/giây | 75% |
| 3 vòng/giây | 75% |
| 5 vòng/giây | 75% |
| 7 vòng/giây | 75% |
| 10 vòng/giây | 75% |
| 12 vòng/giây | 75% |
| 15 vòng/giây | 75% |
| 18 vòng/giây | 75% |
| 24 vòng/giây | 75% |
| 36 vòng/giây (max) | 75% |

Bảng 4.2 Bảng đánh giá thực nghiệm độ chính xác điều khiển tốc độ

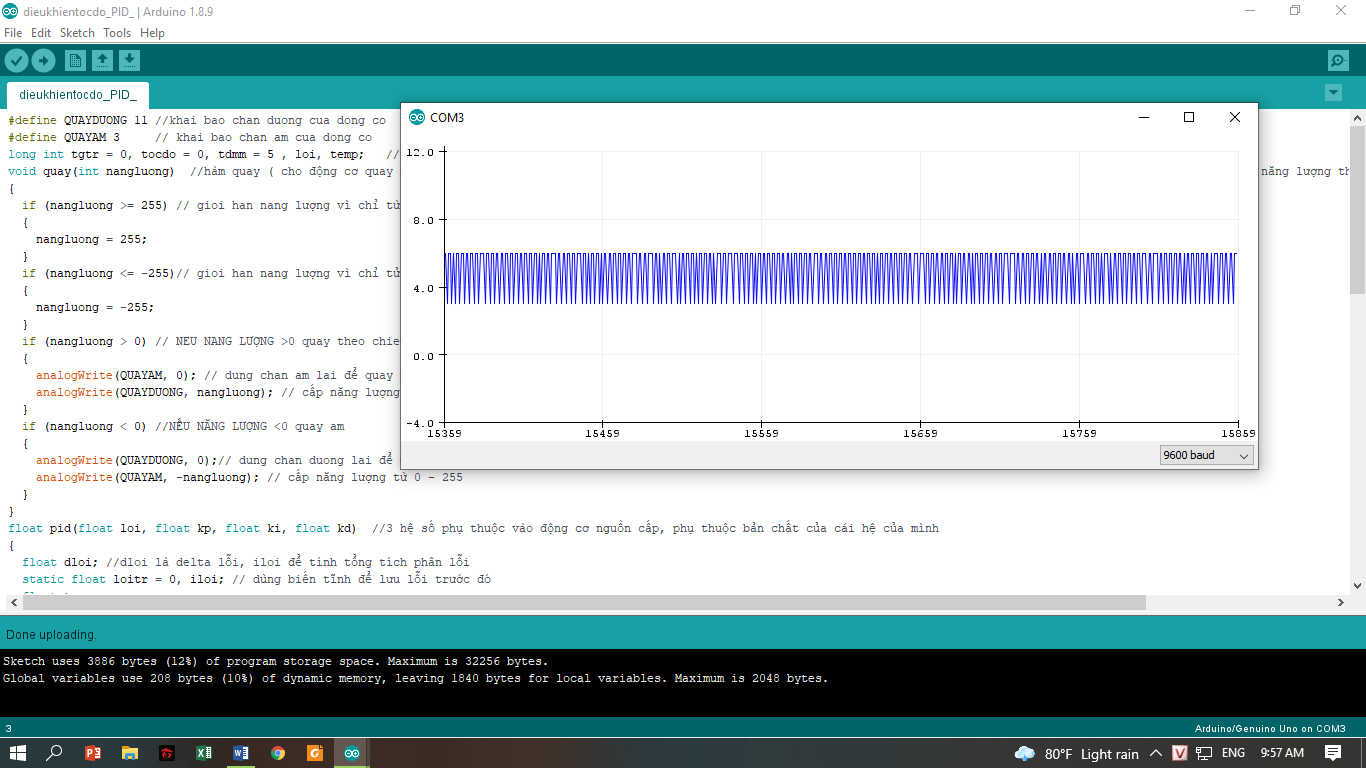
## 4.4. Kết quả điều khiển tốc độ mong muốn động cơ:



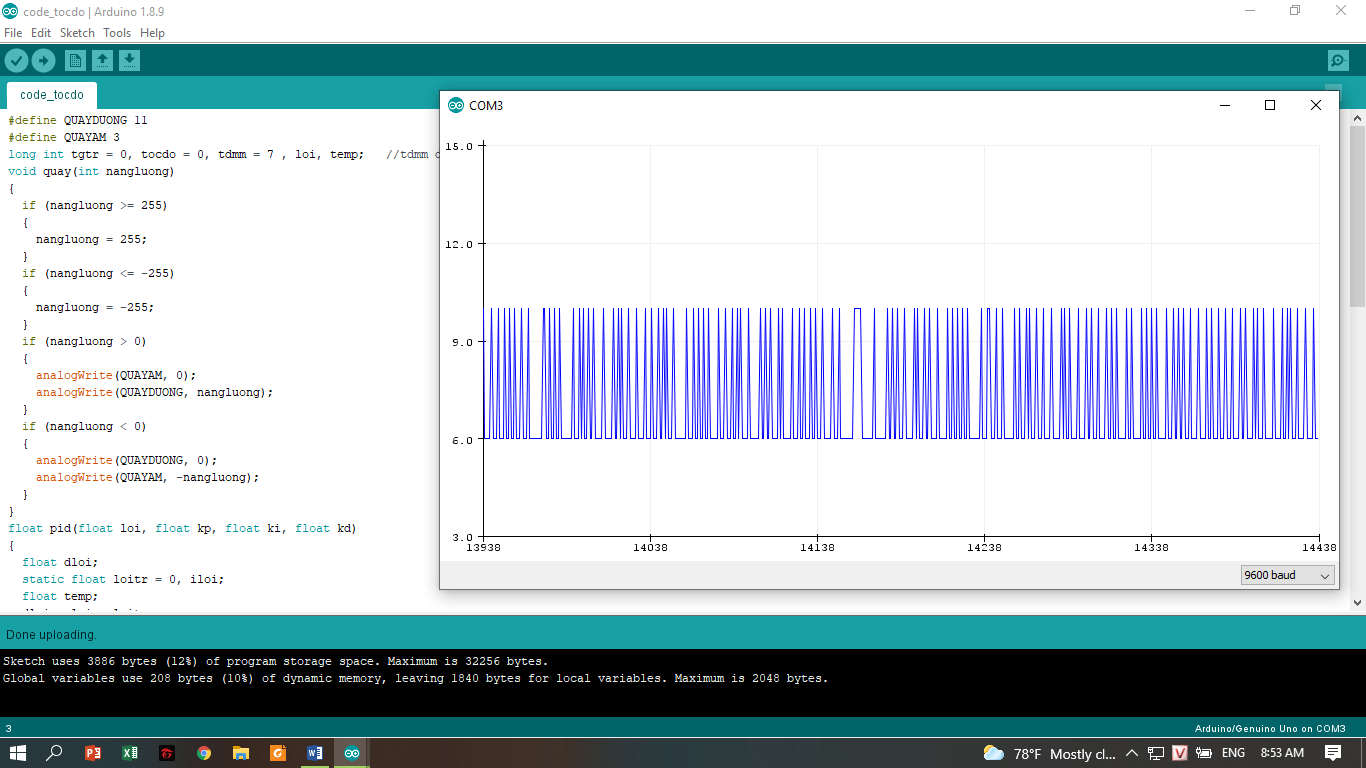
Hình 4.11 Kết quả khi đặt tốc độ mong muốn là 1 vòng/giây



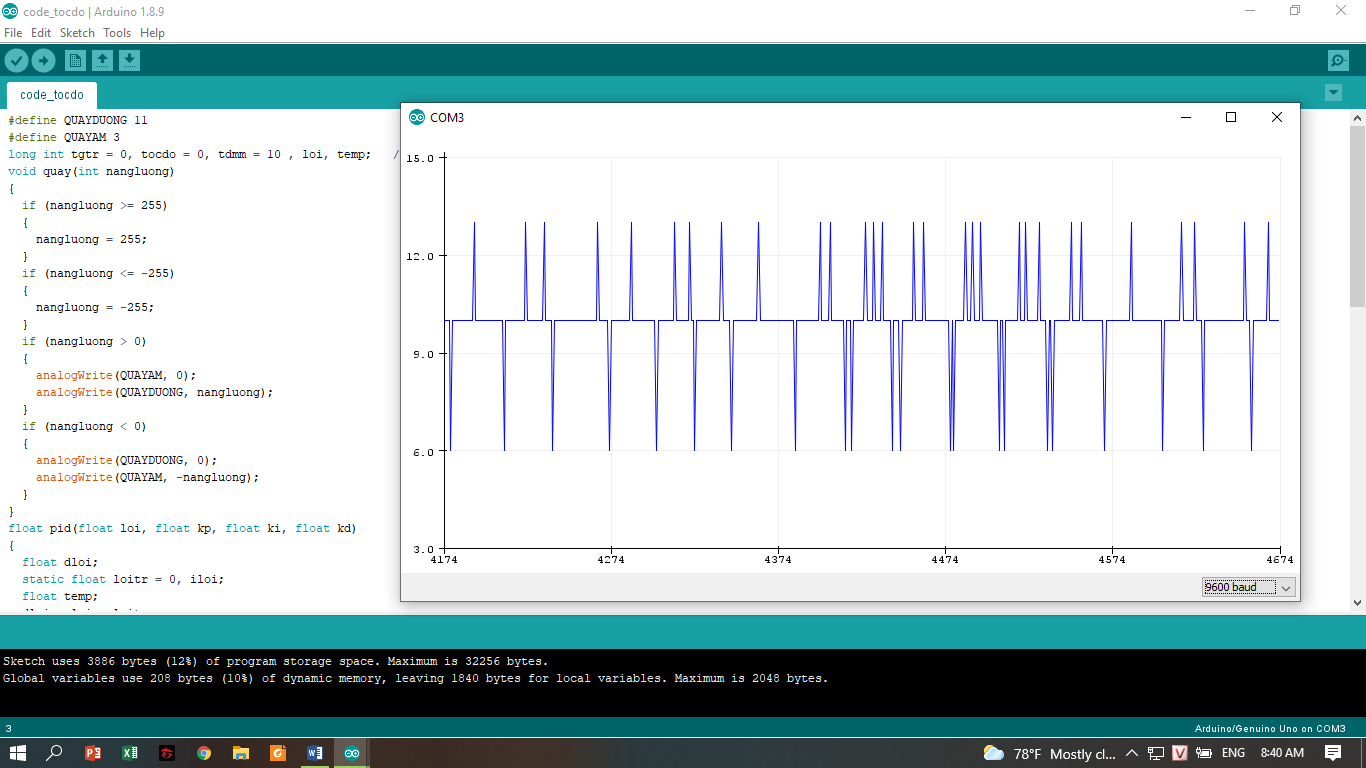
Hình 4.12 Kết quả khi đặt tốc độ mong muốn là 3 vòng/giây



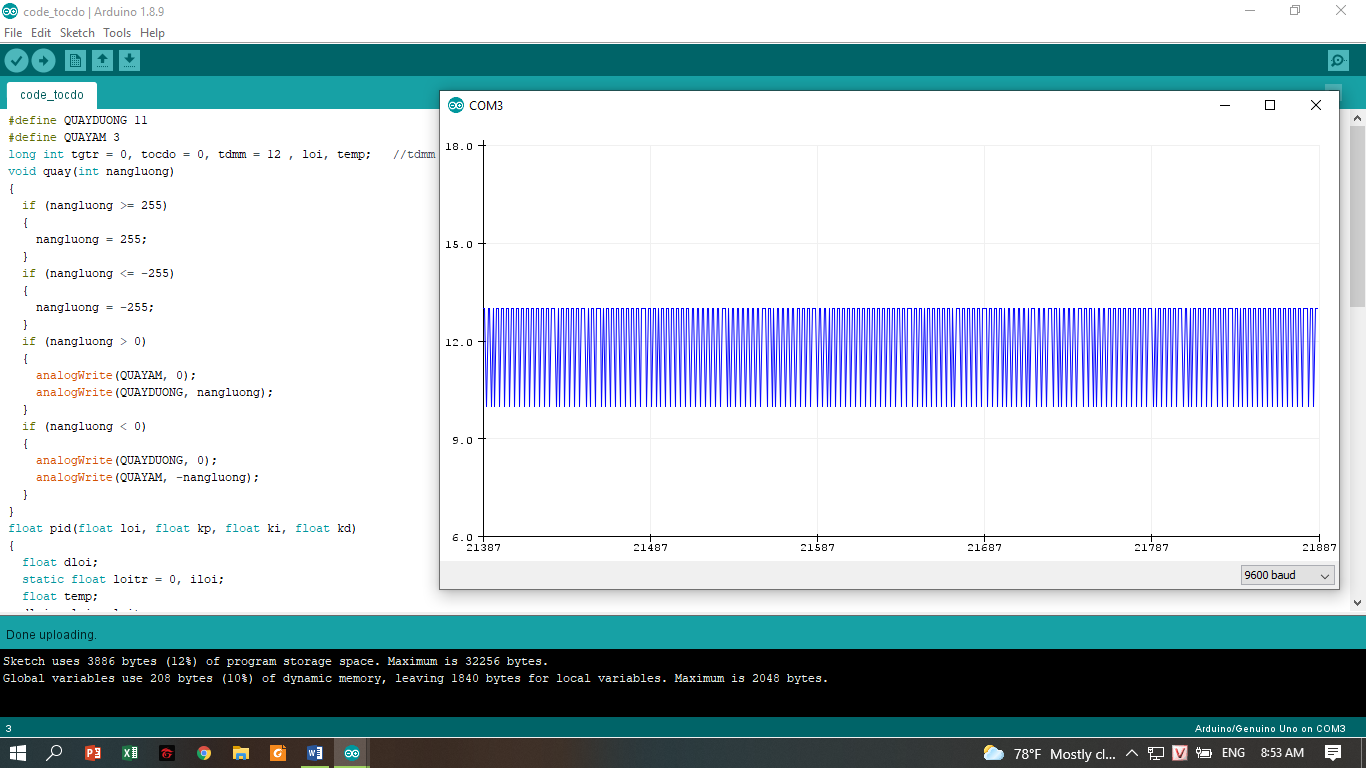
Hình 4.13 Kết quả khi đặt tốc độ mong muốn là 5 vòng/giây



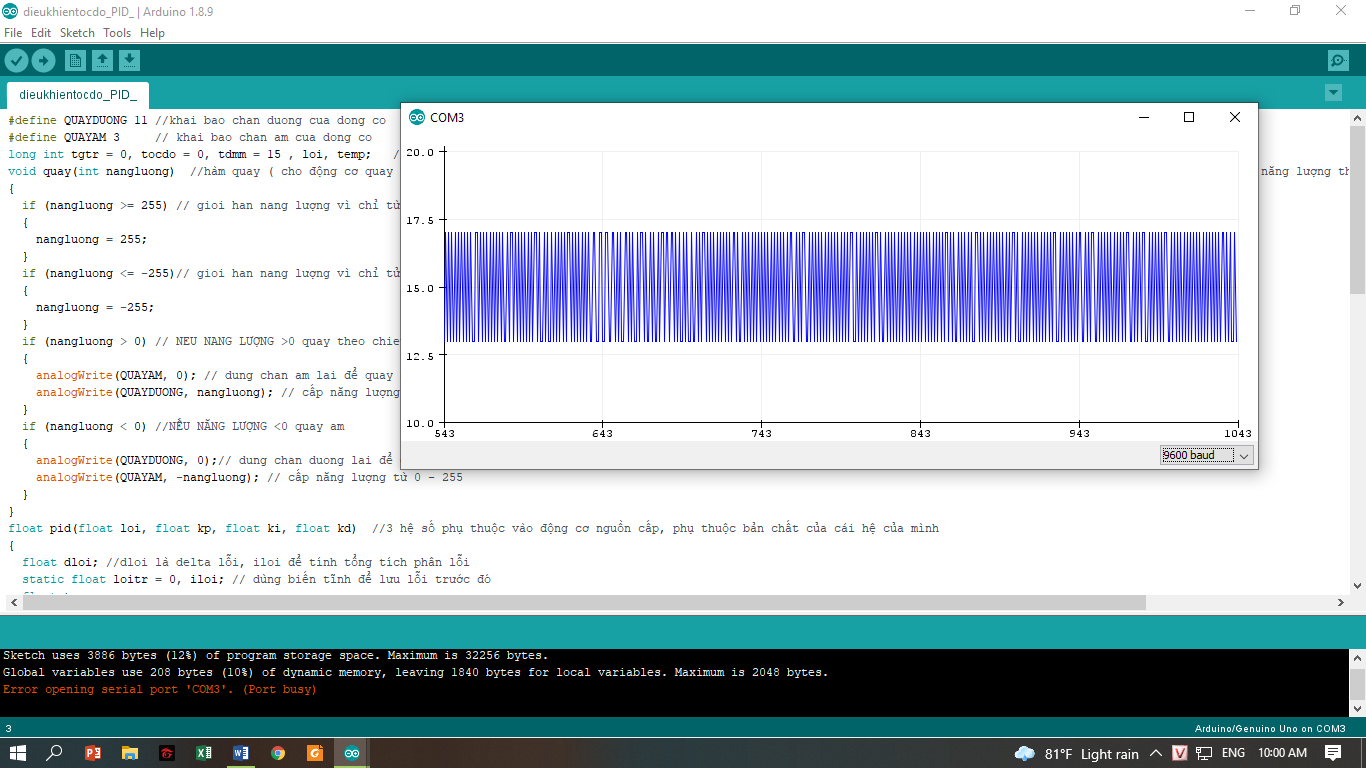
Hình 4.14 Kết quả khi đặt tốc độ mong muốn là 7 vòng/giây



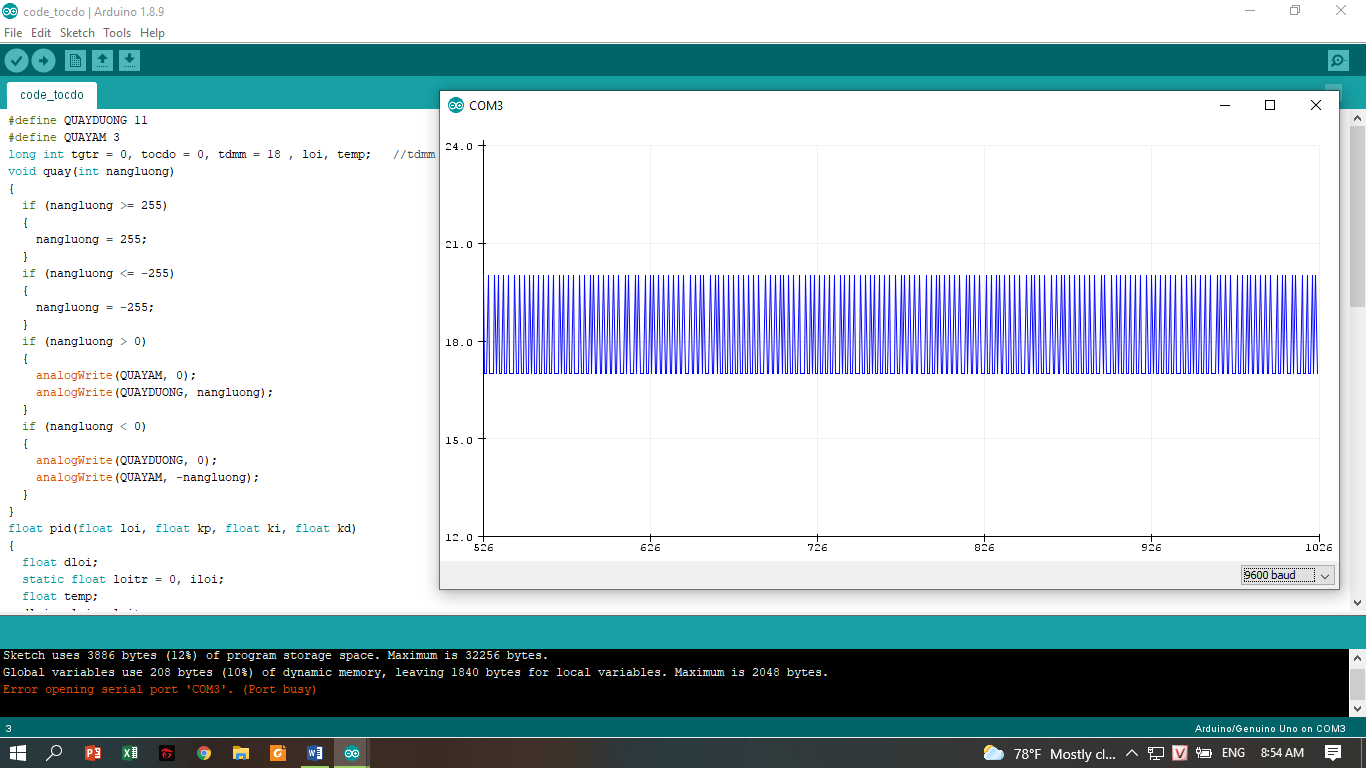
Hình 4.15 Kết quả khi đặt tốc độ mong muốn là 10 vòng/giây



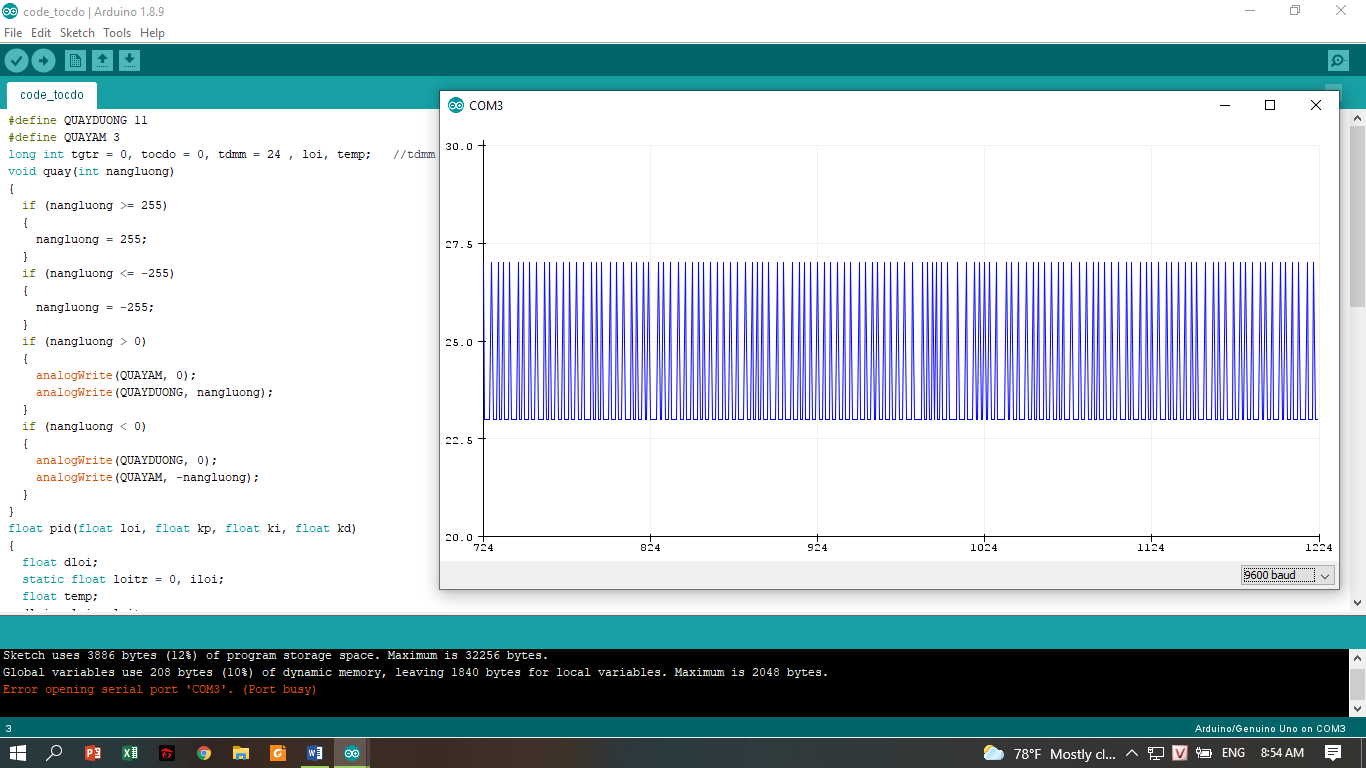
Hình 4.16 Kết quả khi đặt tốc độ mong muốn là 12 vòng/giây



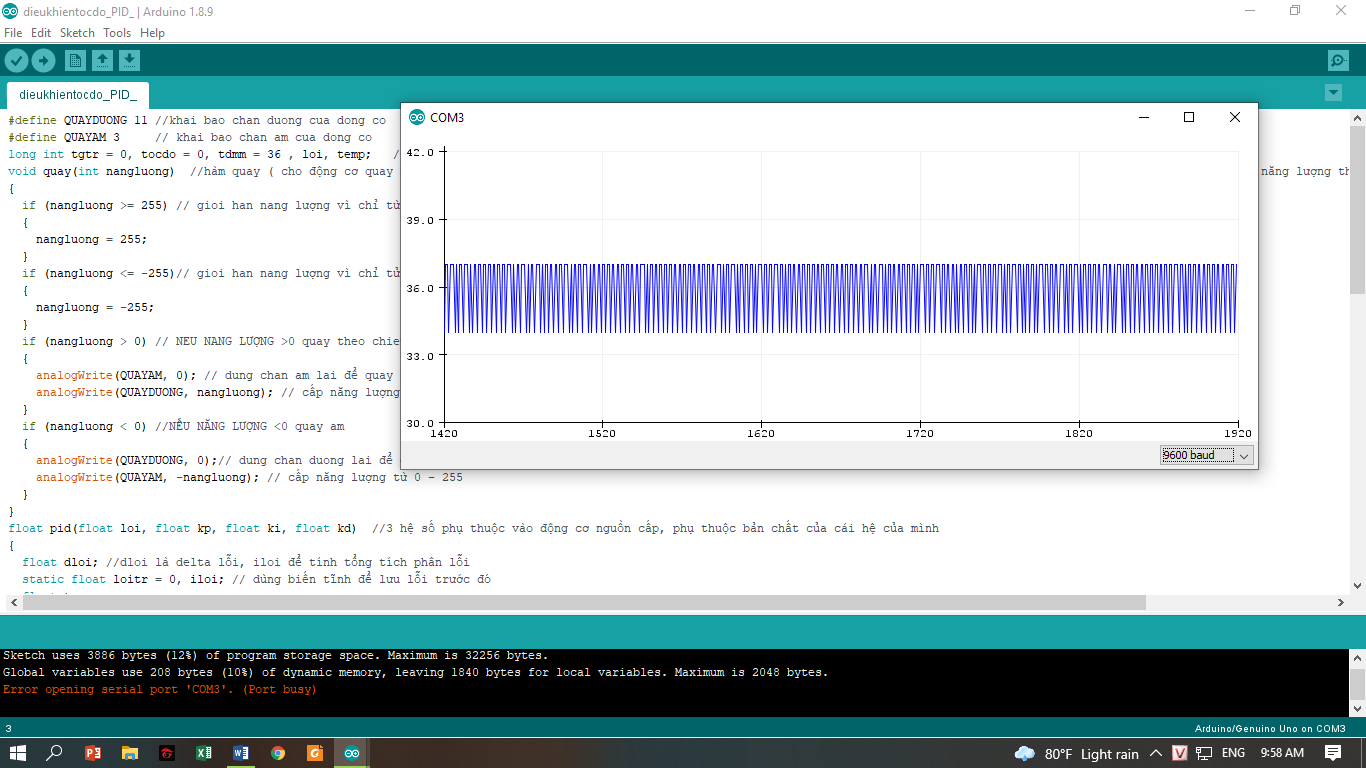
Hình 4.17 Kết quả khi đặt tốc độ mong muốn là 15 vòng/giây



Hình 4.18 Kết quả khi đặt tốc độ mong muốn là 18 vòng/giây



Hình 4.19 Kết quả khi đặt tốc độ mong muốn là 24 vòng/giây

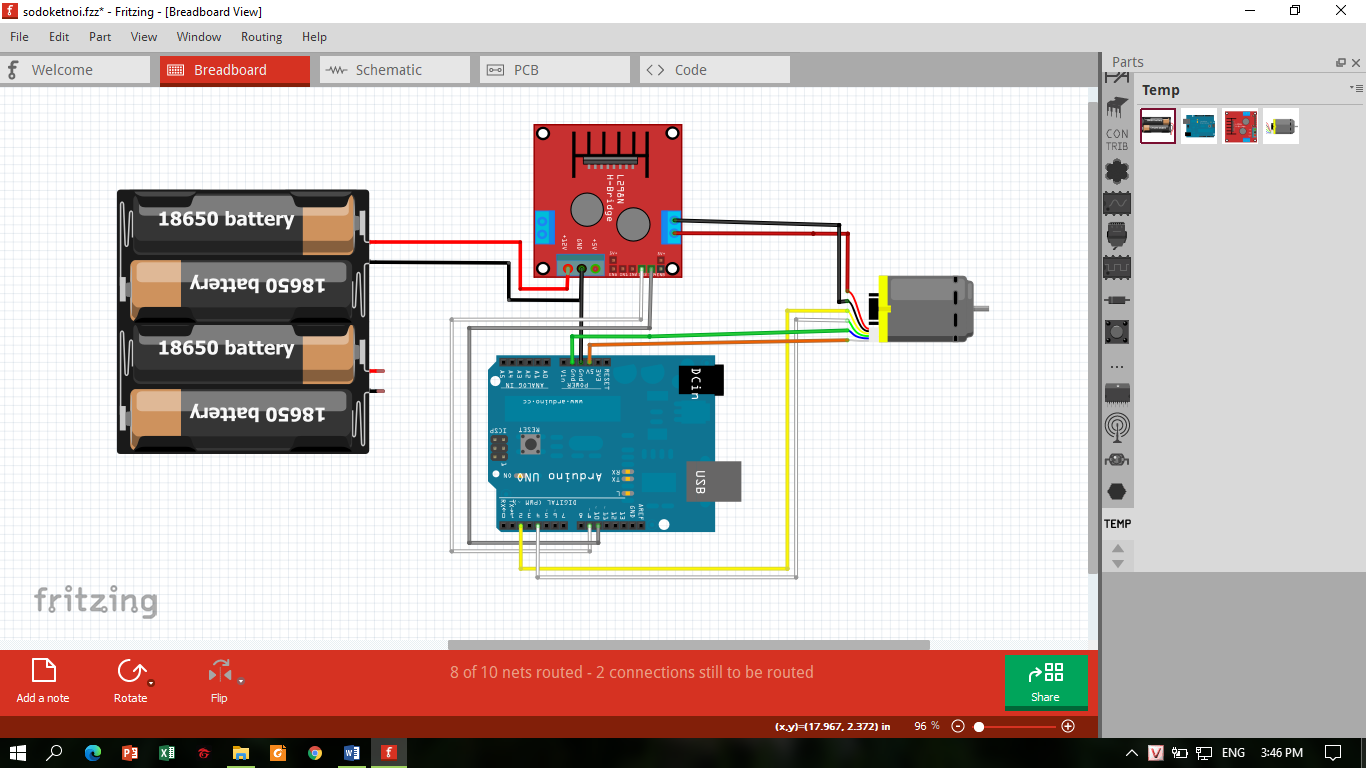


Hình 4.20 Kết quả khi đặt tốc độ mong muốn là 36 vòng/giây

## 4.5. Hình ảnh mô hình điều khiển tốc độ vị trí động cơ:



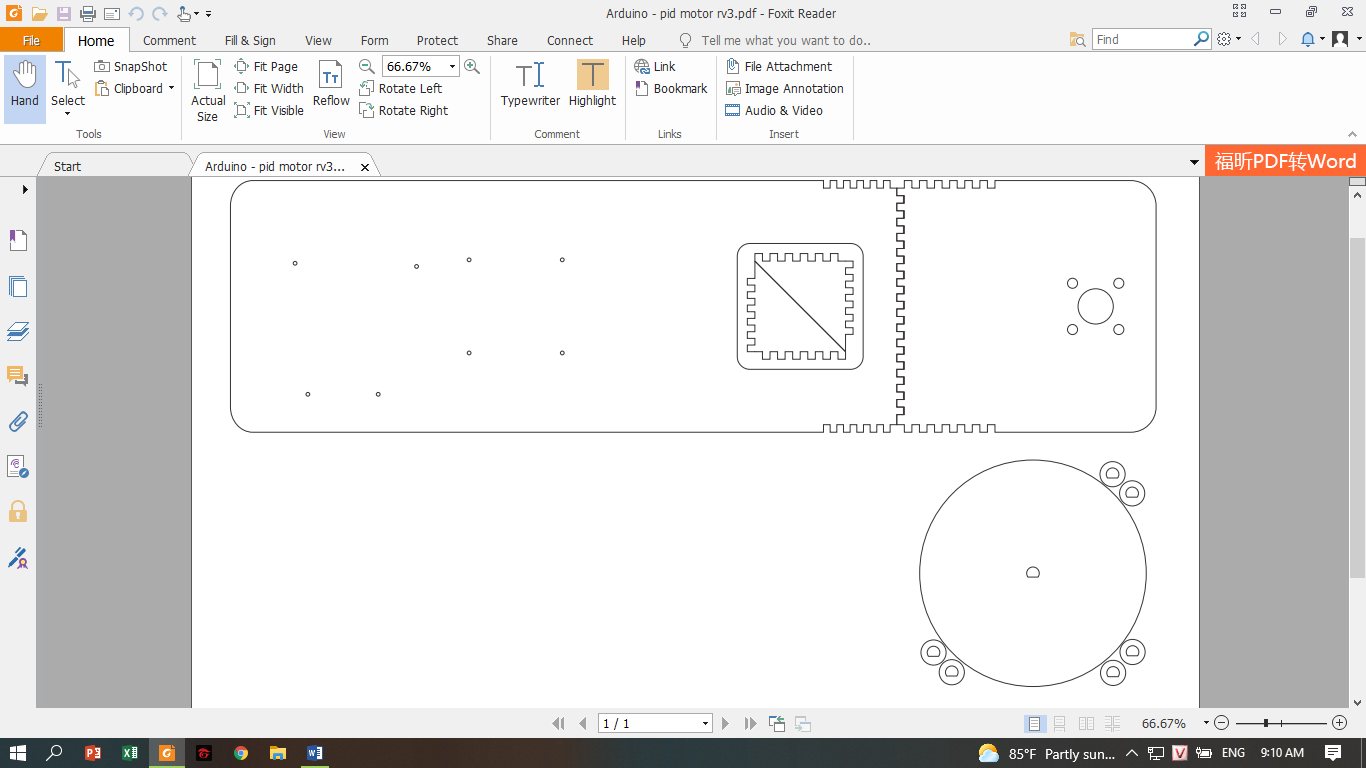
Hình 4.21 Mô hình thực tế



Hình 4.22 Sơ đồ kết nối của hệ thống điều khiển vị trí



Hình 4.23 Sơ đồ kết nối của hệ thống điều khiển tốc độ động cơ



Hình 4.24 Thiết kề phần cứng mô hình điều khiển tốc độ và vị trí động cơ

# CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI

## 5.1 Kết quả đạt được

* Hiểu biết hơn một ít về điều khiển, ở đây là tốc độ và vị trí động cơ DC servo.
* Đã điều khiển được vị trí động cơ DC servo khá chính xác.
* Có kỹ năng viết code hơn sau khi học hỏi và tìm hiểu trên mạng.

## 5.2 Hạn chế

* Tốc độ điều khiển chưa được như mong muốn, xung gai còn nhiều chưa đạt được chính xác tốc độ mong muốn.

## 5.3 Hướng phát triển của đề tài

* Gắn thêm bàn phím ma trận matrix 4x4 để dễ dàng thay đổi thông số vị trí và tốc độ và màn hình LCD để hiển thị chúng.

# PHỤ LỤC

## Code chương trình:

### Code điều khiển vị trí động cơ

#define QUAYDUONG 10

#define QUAYAM 9

int vitri=0, vitrimm= 2600, loi;

void chay()

{

if(digitalRead(4) == 1)

vitri++;

else

vitri--;

}

void quay(int nangluong)

{

if(nangluong>=255)

{

nangluong=255;

}

if(nangluong<=-255)

{

nangluong=-255;

}

if(nangluong>0)

{

analogWrite(QUAYAM,0);

analogWrite(QUAYDUONG,nangluong);

}

if(nangluong<0)

{

analogWrite(QUAYDUONG,0);

analogWrite(QUAYAM,-nangluong);

}

}

float pid(float loi, float kp, float ki, float kd)

{

float dloi;

static float loitr=0, iloi= 0;

float temp;

dloi=loi-loitr;

iloi+=loi;

if( iloi>=80)

{

iloi=80;

}

if( iloi<=-80)

{

iloi=-80;

}

loitr=loi;

temp = kp\*loi+ki\*iloi+kd\*dloi;

if( temp>=255)

{

temp=255;

}

if( temp<=-255)

{

temp=-255;

}

return temp;

}

void setup() {

attachInterrupt(0,chay,RISING);

pinMode(QUAYAM, OUTPUT);

pinMode(QUAYDUONG, OUTPUT);

Serial.begin(9600);

}

void loop()

{

Serial.println(vitri);

loi=vitrimm-vitri;

quay(pid(loi,3,0.25,4));

}

### Code điều khiển tốc độ động cơ

#define QUAYDUONG 11

#define QUAYAM 3

long int tgtr = 0, tocdo = 0, tdmm = 15 , loi, temp; //tdmm:vong / giay (max la 36\*60 2200(v/phut))

void quay(int nangluong)

{

if (nangluong >= 255)

{

nangluong = 255;

}

if (nangluong <= -255)

{

nangluong = -255;

}

if (nangluong > 0)

{

analogWrite(QUAYAM, 0);

analogWrite(QUAYDUONG, nangluong);

}

if (nangluong < 0) //NẾU NĂNG LƯỢNG <0 quay am

{

analogWrite(QUAYDUONG, 0);

analogWrite(QUAYAM, -nangluong);

}

}

float pid(float loi, float kp, float ki, float kd)

{

float dloi;

static float loitr = 0, iloi;

float temp;

dloi = loi - loitr;

iloi += loi;

if (iloi >= 50)

{

iloi = 50;

}

if (iloi <= -50)

{

iloi = -50;

}

loitr=loi;

temp = kp \* loi + ki \* iloi + kd \* dloi;

if (temp >= 255)

{

temp = 255;

}

if (temp <= -255)

{

temp = -255;

}

return temp;

}

void setup() {

TCCR1A = 0;

TCCR1B = 0b111;

pinMode(QUAYAM, OUTPUT);

pinMode(QUAYDUONG, OUTPUT);

Serial.begin(9600);

}

void loop()

{

if (micros() - tgtr >= 1000)

{

tgtr = micros();

tocdo = TCNT1;

TCNT1 = 0;

tocdo= (float)tocdo/1.0\*3.4;

loi = tdmm - tocdo;

temp += pid(loi, 3, 0.001, 5);

quay(temp

Serial.println(tocdo

}

}

# TÀI LIỆU THAM KHẢO

**1. Giáo trình**

[1] Nguyễn Thị Phương Hà, *Trích* “ *Lý thuyết điều khiển tự động”,* Trường Đại học Bách khoa TPHCM

**2. Trang web**

[1] <https://www.youtube.com/watch?v=CabSCbRm2Bo/> (trang hướng dẫn cài đặt Arduino và add thư viện)

[2] <https://www.youtube.com/watch?v=Oai8qMH2yTU/> (trang tham khảo mô hình phần cứng điều khiển động cơ)

[3] <https://www.youtube.com/watch?v=ev5sJ2-JGzA&t=194s/> (trang tham khảo viết code điều khiển động cơ DC servo)

[4] [https://www.youtube.com/watch?v=bMSVXpaZVZY/](https://www.youtube.com/watch?v=ev5sJ2-JGzA&t=194s/) (trang tham khảo hướng dẫn điều khiển PID số)