

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO  
**Hệ thống nhúng &  
Thiết kế giao tiếp nhúng**

ĐỀ TÀI:

**BỘ ĐIỀU KHIỂN BƠM XĂNG  
ĐỘNG CƠ TỪ BÌNH XĂNG**

Giảng viên hướng dẫn      TS. Võ Lê Cường

Sinh viên thực hiện :

Họ tên	MSSV
Lê Huy Hoàng	20192868
Vũ Tuấn Minh	20214015
Lê Minh Tiến	20193132

*Hà Nội, 06/2024*



## LỜI NÓI ĐẦU

Bơm xăng đóng vai trò then chốt trong hệ thống nhiên liệu, cung cấp nhiên liệu cho động cơ hoạt động hiệu quả. Việc điều khiển bơm xăng tối ưu góp phần tiết kiệm năng lượng, bảo vệ môi trường và nâng cao hiệu suất động cơ. Nhận thức được tầm quan trọng đó, nhóm chúng em xin trình bày đề tài "Bộ điều khiển bơm xăng động cơ từ bình xăng", nhằm mục tiêu thiết kế, chế tạo và đánh giá hiệu quả hoạt động của bộ điều khiển này.

Báo cáo đi sâu nghiên cứu các loại bơm xăng hiện có, phân tích yêu cầu điều khiển, thiết kế phần cứng và phần mềm cho bộ điều khiển, đồng thời chế tạo và thử nghiệm thực tế. Phương pháp nghiên cứu bao gồm nghiên cứu tài liệu, phân tích mô phỏng và thử nghiệm thực nghiệm.

Chúng em hy vọng mang đến những thông tin hữu ích về bộ điều khiển bơm xăng động cơ từ bình xăng, góp phần nâng cao hiệu quả sử dụng nhiên liệu và bảo vệ môi trường.

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Sự cần thiết của bộ điều khiển bơm xăng động cơ từ bình xăng

### 1.1.1. Đặt vấn đề

- Bơm xăng là bộ phận thiết yếu trong hệ thống nhiên liệu của động cơ, đảm nhiệm chức năng cung cấp nhiên liệu cho động cơ hoạt động.
- Việc điều khiển bơm xăng truyền thống thường dựa vào các phương pháp cơ học hoặc điện đơn giản, dẫn đến một số hạn chế:
  - + Không đảm bảo cung cấp lượng nhiên liệu phù hợp với nhu cầu của động cơ ở mọi điều kiện hoạt động.
  - + Gây lãng phí nhiên liệu, gia tăng khí thải và giảm hiệu suất động cơ.
  - + Khó khăn trong việc điều chỉnh và kiểm soát hoạt động của bơm xăng

### 1.1.2. Tầm quan trọng của Ứng dụng

- Cải thiện hiệu suất động cơ:
  - + Điều khiển chính xác lượng nhiên liệu cung cấp cho động cơ ở mọi điều kiện hoạt động, giúp động cơ hoạt động hiệu quả hơn, tiết kiệm nhiên liệu và giảm thiểu khí thải.
  - + Tối ưu hóa quá trình đốt cháy nhiên liệu, nâng cao công suất và mô-men xoắn của động cơ.
- Giảm thiểu ô nhiễm môi trường:
  - + Hạn chế lượng nhiên liệu dư thừa, giảm khí thải độc hại ra môi trường.
  - + Góp phần bảo vệ môi trường và ứng phó với biến đổi khí hậu.
- Nâng cao trải nghiệm người dùng:
  - + Động cơ vận hành êm ái, mượt mà hơn.
  - + Tăng tuổi thọ động cơ và các bộ phận liên quan.
  - + Giảm chi phí bảo dưỡng và sửa chữa xe.

## 1.2 Mục tiêu của hệ thống

- Cung cấp lượng nhiên liệu phù hợp cho động cơ:
  - Hệ thống điều khiển bơm xăng dựa trên các yếu tố như tốc độ động cơ, tải trọng, áp suất khí nạp, nhiệt độ động cơ,... để cung cấp lượng nhiên liệu chính xác cho động cơ hoạt động hiệu quả nhất.
  - Việc cung cấp nhiên liệu phù hợp giúp tối ưu hóa hiệu suất đốt cháy, tiết kiệm nhiên liệu, giảm thiểu khí thải độc hại và cải thiện khả năng vận hành của động cơ.
- Duy trì áp suất nhiên liệu ổn định:
  - Hệ thống đảm bảo áp suất nhiên liệu cung cấp cho động cơ luôn ổn định trong mọi điều kiện hoạt động, từ đó giúp động cơ khởi động dễ dàng, vận hành trơn tru và có mô-men xoắn cao.

- Áp suất nhiên liệu ổn định cũng góp phần bảo vệ các bộ phận trong hệ thống nhiên liệu khỏi hư hỏng do rung động hay va đập.
- Tăng cường độ tin cậy và an toàn:
  - Hệ thống sử dụng các cảm biến và bộ xử lý hiện đại để giám sát và điều khiển hoạt động của bơm xăng một cách chính xác và hiệu quả.
  - Hệ thống cũng được trang bị các tính năng bảo vệ như ngắt bơm khi áp suất nhiên liệu quá cao hoặc quá thấp, giúp đảm bảo an toàn cho người sử dụng và bảo vệ động cơ khỏi hư hỏng.
- Tiết kiệm năng lượng và bảo vệ môi trường:
  - Việc cung cấp nhiên liệu chính xác và duy trì áp suất nhiên liệu ổn định giúp động cơ hoạt động hiệu quả hơn, giảm thiểu lượng nhiên liệu tiêu hao và lượng khí thải độc hại ra môi trường.
  - Hệ thống góp phần nâng cao ý thức tiết kiệm năng lượng và bảo vệ môi trường cho người sử dụng.
- Tăng cường khả năng kiểm soát và điều chỉnh:
  - Hệ thống cho phép người sử dụng dễ dàng điều chỉnh các thông số hoạt động của bơm xăng như lưu lượng, áp suất,... để phù hợp với nhu cầu sử dụng và điều kiện vận hành.
  - Khả năng điều chỉnh linh hoạt này giúp tối ưu hóa hiệu suất hoạt động của động cơ và đáp ứng nhu cầu đa dạng của người sử dụng.

### 1.3 Tìm hiểu các bài báo về bộ điều khiển bơm xăng động cơ từ bình xăng

Dựa trên mục tiêu đã đề ra, nhóm đã tìm hiểu về các bộ điều khiển có trên thị trường hiện nay:

- Bộ chế hòa khí: [\*\*NGHIÊN CỨU CHẾ TẠO, THỬ NGHIỆM BỘ TIẾT KIỆM XĂNG CHO XE Ô TÔ LẮP ĐỘNG CƠ SỬ DỤNG BỘ CHẾ HÒA KHÍ\*\* – Lê Văn Anh, Ngô Quang Tào, Lê Đình Mạnh, Nguyễn Thế Anh.](#)  
 Bài báo này nói về việc giảm thiểu tiêu hao nhiên liệu cho ô tô. Bài báo thảo luận về việc sử dụng quán tính để giảm thiểu tiêu hao nhiên liệu cho xe có động cơ chế hòa khí. Các tác giả đã thiết kế một thiết bị ngắt nhiên liệu sử dụng quán tính để cất lưu lượng khí vào động cơ khi xe di chuyển bằng quán tính. Điều này giúp giảm thiểu lượng nhiên liệu lãng phí. Trong các bài kiểm tra, thiết bị đã giảm thiểu 36% lượng tiêu hao nhiên liệu trong môi trường kiểm soát và 6% trong điều kiện lái xe thực tế. Các tác giả tin rằng thiết bị này có thể giảm thiểu đáng kể lượng tiêu hao nhiên liệu cho ô tô, đặc biệt là những xe di chuyển ở khu vực đồi núi.
- Hệ thống phun xăng trực tiếp (GDI): [\*\*CHARACTERISTICS OF GDI ENGINE FLOW STRUCTURES\*\* - N. J. Beavis, S. S. Ibrahim, P. K. Manickam, W. Malalasekera.](#)

Bài báo này nói về đặc điểm của cấu trúc dòng chảy động cơ phun xăng trực tiếp (GDI). Bài báo thảo luận về những lợi ích của động cơ GDI so với động cơ phun nhiên liệu cổng truyền thống. Những lợi ích này bao gồm hiệu quả nhiên liệu được cải thiện, giảm thiểu khả năng gõ và giảm thiểu hydrocacbon không cháy. Tuy nhiên, động cơ GDI cũng đi kèm với những thách thức như biến đổi chu kỳ, hiện tượng đốt cháy bất thường và phát thải hạt bụi tăng lên. Các tác giả thảo luận về một mô hình 3D-CFD được phát triển để dự đoán trường dòng trong xi lanh qua các kỳ nạp và nén. Mô hình này đã được xác nhận bằng dữ liệu thí nghiệm từ động cơ nghiên cứu quang học một xi lanh. Kết quả cho thấy mô hình này nói chung là chính xác trong việc dự đoán trường dòng. Tuy nhiên, có một số khác biệt giữa mô hình và dữ liệu thí nghiệm, chẳng hạn như thời điểm tách tia van nạp khỏi đầu xi lanh.

- Hệ thống phun xăng điện tử (EFI): [\*\*COMPARATIVE INVESTIGATION OF ELECTRONIC FUEL INJECTION IN TWO-WHEELER APPLICATIONS: A REVIEW - K Venkata Mohan Krishna Reddy\*\*](#)

Bài báo này đánh giá về hệ thống phun xăng điện tử (EFI) trên các loại xe hai bánh. Hệ thống này đã thay thế cho chế hòa khí, giúp kiểm soát chính xác hơn lượng hỗn hợp không khí-xăng đi vào động cơ, cải thiện hiệu suất hoạt động và giảm thiểu khí thải.

Tiêu chí	Bộ chế hòa khí	Hệ thống phun xăng điện tử (EFI)	Hệ thống phun xăng trực tiếp (GDI)
<b>Nguyên tắc hoạt động</b>	Dựa vào áp suất âm trong buồng đốt để hút nhiên liệu từ trong bình xăng và hòa trộn với không khí theo tỷ lệ nhất định.	Sử dụng các kim phun điện tử để phun nhiên liệu vào đường dẫn khí hoặc trực tiếp vào buồng đốt.	Sử dụng các kim phun điện tử áp suất cao để phun nhiên liệu trực tiếp vào buồng đốt.
<b>Cấu tạo</b>	Gồm các bộ phận chính như: buồng phao, bộ khuếch đại, vòi phun, van gió, van khí phụ, giắc ga,...	Gồm các bộ phận chính như: bơm nhiên liệu, kim phun, cảm biến, bộ điều khiển điện tử (ECU),...	Gồm các bộ phận chính như: bơm nhiên liệu áp suất cao, kim phun áp suất cao, cảm biến, bộ điều khiển điện tử (ECU),...
<b>Ưu điểm</b>	- Cấu tạo đơn giản, dễ sản xuất, giá	- Kiểm soát chính xác lượng nhiên liệu và tỷ	- Hiệu suất đốt cháy cao, tiết kiệm nhiên

	thành rẻ. - Dễ dàng sửa chữa, bảo dưỡng.	lệ không khí-nhiên liệu, giúp cải thiện hiệu suất động cơ và tiết kiệm nhiên liệu. - Giảm thiểu khí thải độc hại. - Khởi động dễ dàng trong mọi điều kiện thời tiết.	liệu tối ưu. - Giảm thiểu khí thải độc hại. - Tăng cường công suất động cơ.
<b>Nhược điểm</b>	- Khó kiểm soát lượng nhiên liệu và tỷ lệ không khí-nhiên liệu, dẫn đến lãng phí nhiên liệu và gia tăng khí thải. - Khó khởi động trong thời tiết lạnh. - Không đáp ứng được tiêu chuẩn khí thải ngày càng khắt khe.	- Cấu tạo phức tạp, giá thành cao. - Yêu cầu độ chính xác cao trong chế tạo và lắp đặt. - Dễ bị hư hỏng do cặn bẩn trong nhiên liệu.	- Cấu tạo phức tạp, giá thành cao. - Yêu cầu độ chính xác cao trong chế tạo và lắp đặt. - Dễ bị hư hỏng do cặn bẩn trong nhiên liệu. - Có thể gây ra hiện tượng bám cặn carbon trên piston và xi lanh.
<b>Ứng dụng</b>	Được sử dụng phổ biến trên các loại xe máy đời cũ, xe máy giá rẻ.	Được sử dụng phổ biến trên các loại xe máy hiện đại, xe ô tô.	Được sử dụng trên các loại xe máy và xe ô tô cao cấp.
<b>Tương lai</b>	Dần được thay thế bởi các hệ thống phun xăng điện tử và phun xăng trực tiếp.	Có tiềm năng phát triển mạnh mẽ trong tương lai nhờ khả năng tiết kiệm nhiên liệu và giảm thiểu khí thải tối ưu.	Có tiềm năng phát triển mạnh mẽ trong tương lai nhờ khả năng tiết kiệm nhiên liệu và giảm thiểu khí thải tối ưu. Tuy nhiên, cần khắc phục các nhược điểm về cấu tạo phức tạp và giá thành cao.

Dựa trên các ưu điểm của Hệ thống phun xăng điện tử (EFI), nhóm đã quyết định chọn thực hiện xây dựng và thiết kế hệ thống này.

## 1.4 Phân công công việc

**Bảng 1. Phân công nhiệm vụ**

<b>Lê Huy Hoàng</b>	<ul style="list-style-type: none"><li>- Đồng thiết kế cứng mềm</li><li>- Mô hình hóa hệ thống sử dụng FSM</li><li>- Thiết kế giao tiếp ngoại vi Atmega16 với LCD</li><li>- Xác định các chỉ tiêu kỹ thuật</li></ul>
<b>Vũ Tuấn Minh</b>	<ul style="list-style-type: none"><li>- Mô hình hóa hệ thống sử dụng SystemC</li><li>- Thiết kế giao tiếp ngoại vi Atmega16 với khối điều khiển máy bơm và động cơ</li><li>- Tổng hợp, cấu hình và biên dịch phần mềm</li><li>- Thực hiện mô phỏng và kiểm thử</li></ul>
<b>Lê Minh Tiến</b>	<ul style="list-style-type: none"><li>- Đồng thiết kế cứng mềm</li><li>- Tìm hiểu các hệ thống bơm xăng trên thị trường</li><li>- Xác định tính năng của hệ thống</li><li>- Mô hình hóa hệ thống sử dụng các mô hình UML</li><li>- Thiết kế giao tiếp ngoại vi Atmega16 với khối nút bấm</li></ul>



## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP NGHIÊN CỨU

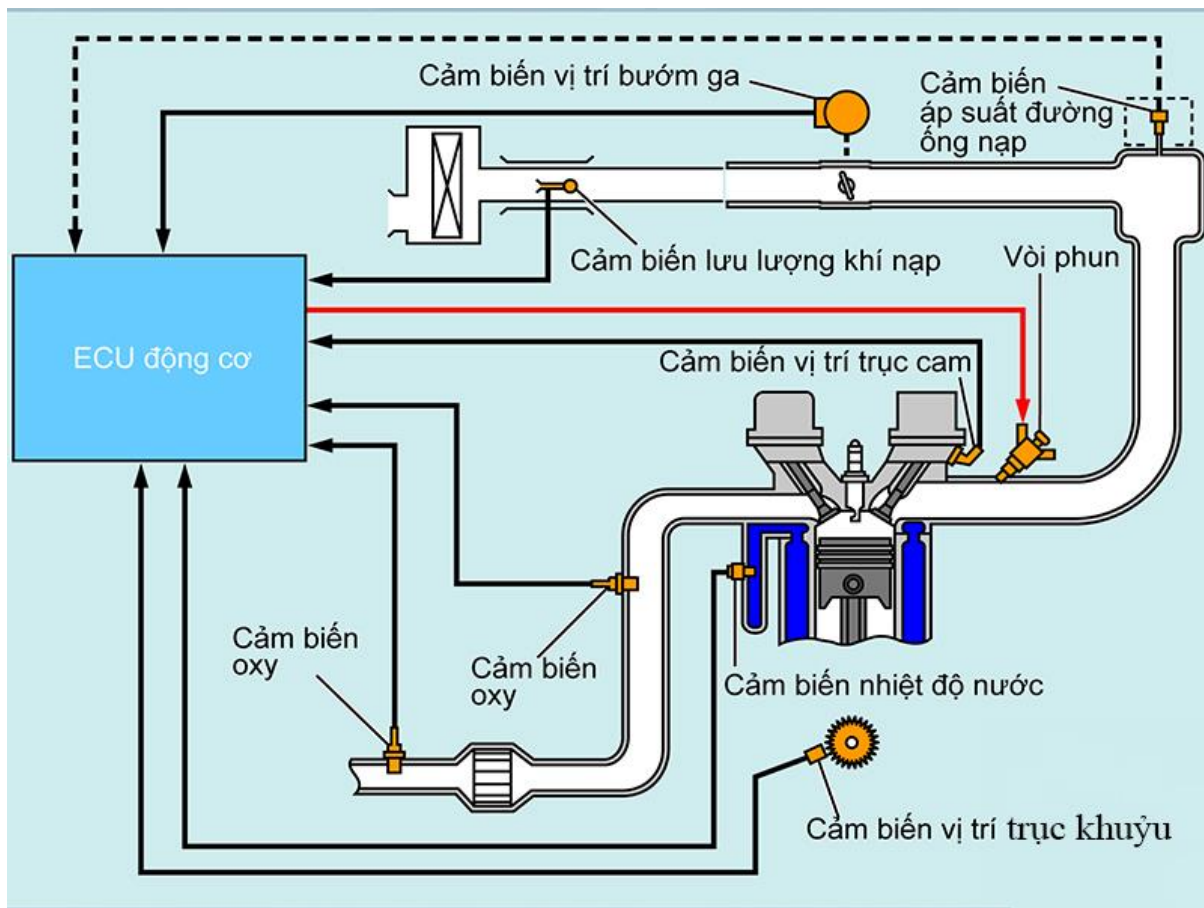
### 2.1 Cơ sở hoạt động của hệ thống phun xăng thật

#### 2.1.1 Hệ thống phun xăng điện tử (EFI) là gì?

- Hệ thống cung cấp nhiên liệu cho động cơ sử dụng cảm biến và bộ xử lý điện tử để điều khiển lượng nhiên liệu và thời điểm phun xăng chính xác hơn so với hệ thống phun xăng cơ khí truyền thống.
- Ưu điểm: Tiết kiệm nhiên liệu, giảm thiểu khí thải, nâng cao hiệu suất động cơ, khởi động dễ dàng, hoạt động êm ái.
- Nhược điểm: Chi phí cao hơn, cấu tạo phức tạp hơn, yêu cầu bảo dưỡng định kỳ.

#### 2.1.2 Cấu tạo:

- ECU động cơ: ECU này tính thời gian phun nhiên liệu tối ưu dựa vào các tín hiệu từ các cảm biến.
- Cảm biến lưu lượng khí nạp hoặc cảm biến áp suất đường ống nạp: Cảm biến này phát hiện khối lượng không khí nạp hoặc áp suất của ống nạp.
- Cảm biến vị trí trục khuỷu: Cảm biến này phát hiện góc quay trục khuỷu và tốc độ của động cơ.
- Cảm biến vị trí trục cam: Cảm biến này phát hiện góc quay chuẩn và thời điểm của trục cam.
- Cảm biến nhiệt độ nước: Cảm biến này phát hiện nhiệt độ của nước làm mát.
- Cảm biến vị trí bướm ga: Cảm biến này phát hiện góc mở của bướm ga.
- Cảm biến oxy: Cảm biến này phát hiện nồng độ của oxy trong khí xả.



#### 2.1.4. Phân loại:

Có hai loại hệ thống EFI được phân loại theo phương pháp phát hiện lượng không khí nạp.

##### a. L - EFI (Loại điều khiển lưu lượng không khí)

Loại này sử dụng một cảm biến lưu lượng khí nạp để phát hiện lượng không khí chạy vào đường ống nạp.

Có hai phương pháp phát hiện: Một loại trực tiếp đo khối không khí nạp, và một loại thực hiện các hiệu chỉnh dựa vào thể tích không khí.

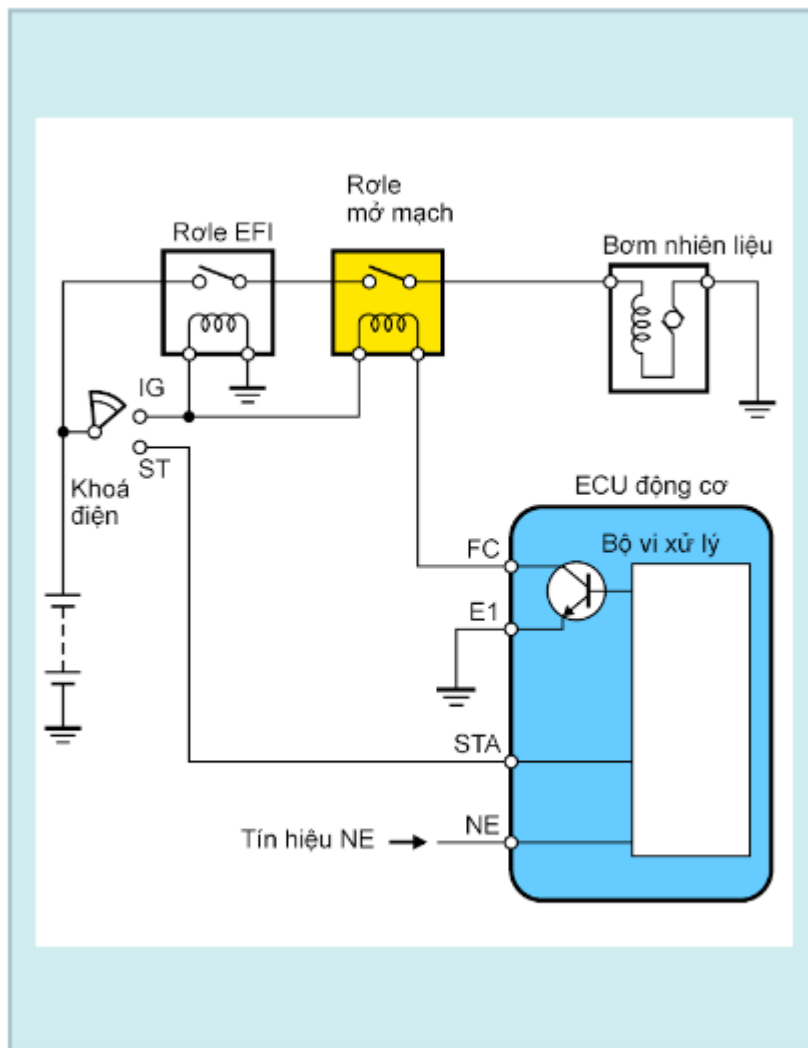
##### b. D - EFI (Loại điều khiển áp suất đường ống nạp)

Loại này đo áp suất trong đường ống nạp để phát hiện lượng không khí nạp theo tỷ trọng của không khí nạp.

#### 2.1.3 Nguyên lý điều khiển bơm nhiên liệu

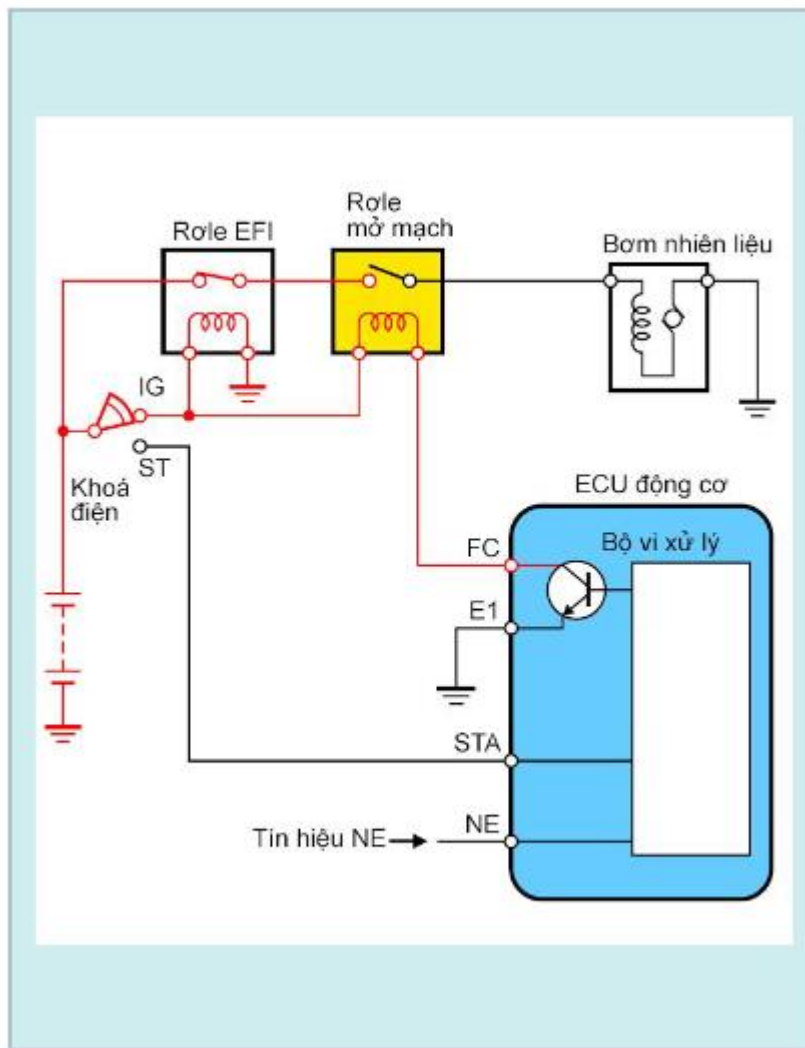
##### a. Hoạt động cơ bản

Bơm nhiên liệu chỉ hoạt động khi động cơ đang chạy. Thậm chí khi khóa điện được bật đến vị trí ON, nếu động cơ chưa nổ máy, thì bơm nhiên liệu sẽ không làm việc



(1) Khóa điện ở vị trí ON :

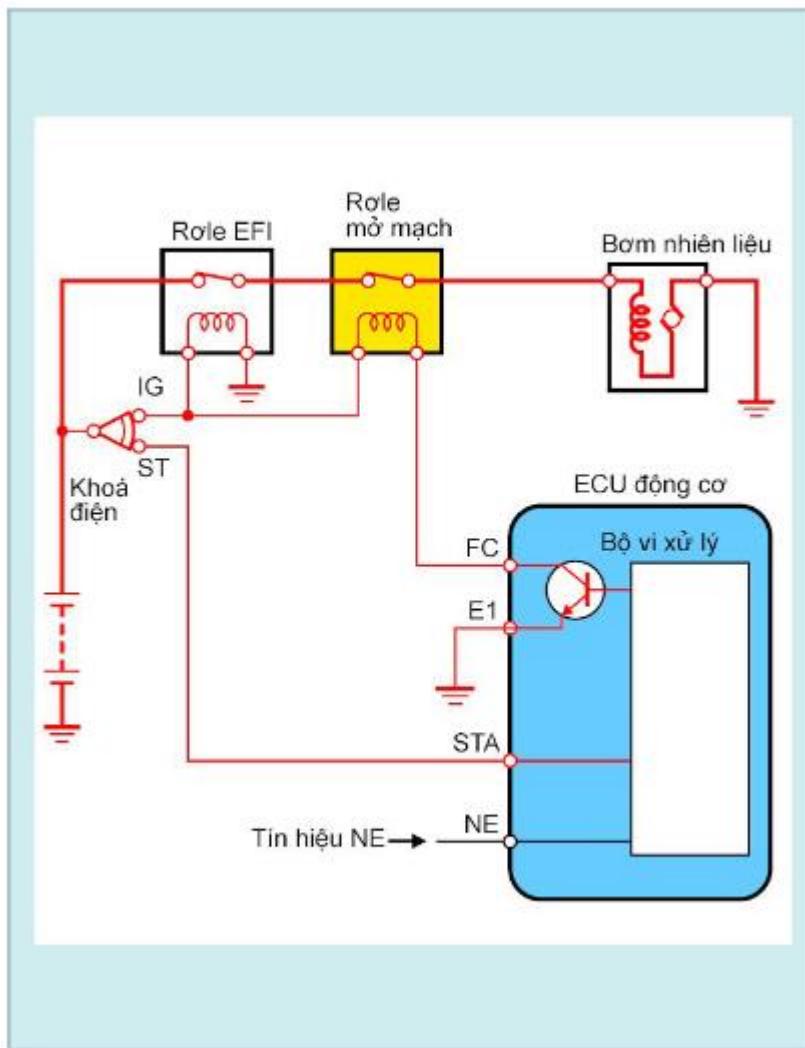
Khi bật khóa điện ở vị trí IG, rơ lê EFI bật mở



(2) Khóa điện ở vị trí START :

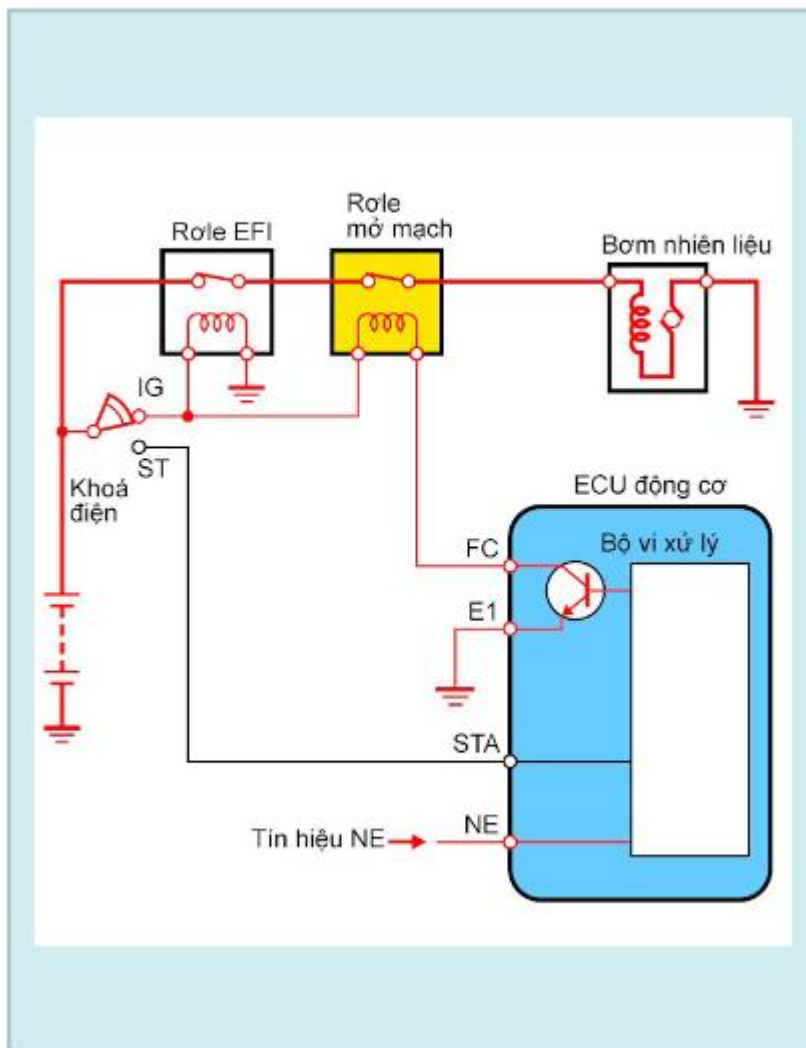
Khi động cơ quay khởi động, một tín hiệu STA (tín hiệu máy khởi động) được truyền đến ECU động cơ từ cực ST của khóa điện.

Khi tín hiệu STA được đưa vào ECU động cơ, động cơ ON tranzito này và rơ le mở mạch được bật ON. Sau đó, dòng điện được chạy vào bơm nhiên liệu để vận hành bơm.



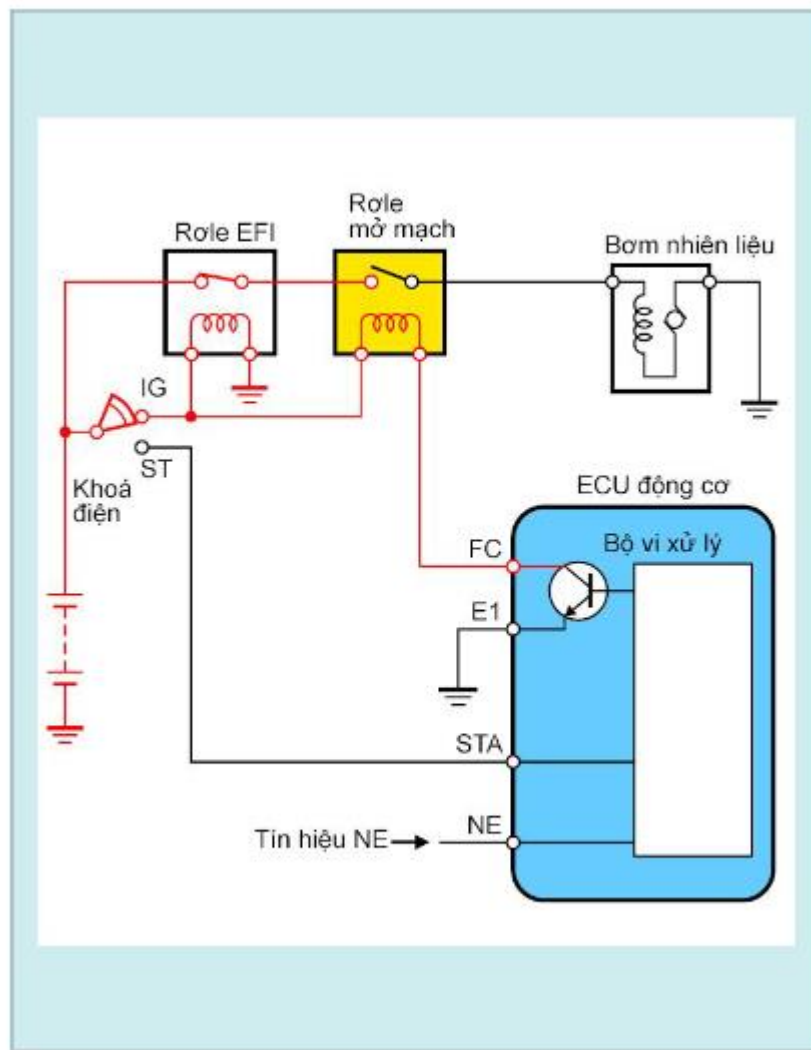
### (3) Động cơ quay khởi động :

Cùng một lúc khi động cơ quay khởi động, ECU động cơ nhận tín hiệu NE từ cảm biến vị trí của trục khuỷu, làm cho tranzito này tiếp tục duy trì hoạt động của bơm nhiên liệu.



(4) Nếu động cơ tắt máy :

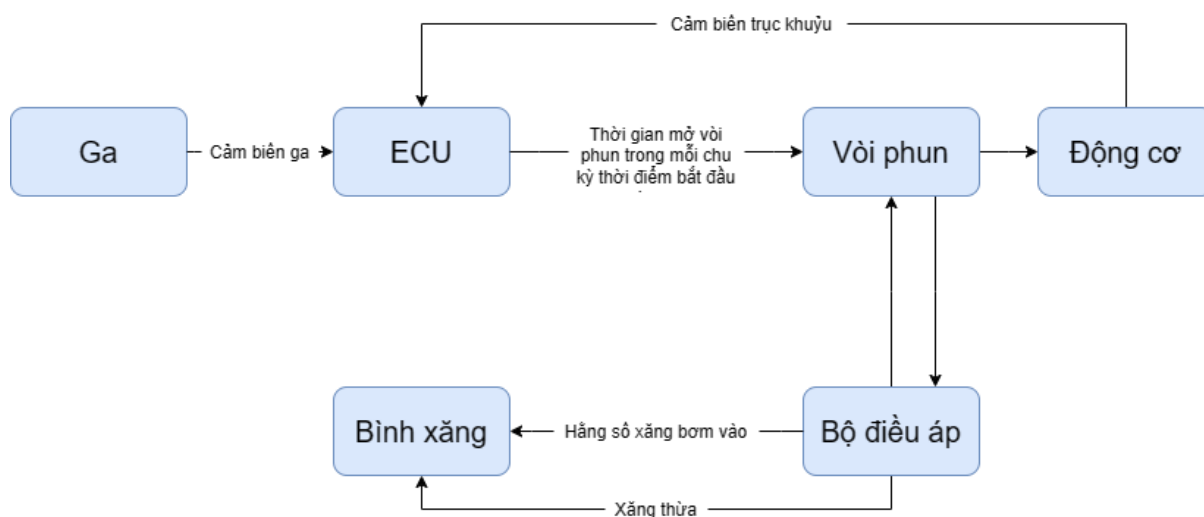
Thậm chí khi khóa điện bật ON, nếu động cơ tắt máy, tín hiệu NE sẽ không còn được đưa vào ECU động cơ sẽ ngắt tranzito này, nó ngắt role mở mạch, làm cho bơm nhiên liệu ngừng lại



### 2.1.5 Ưu điểm và hạn chế:

- Ưu điểm:
  - Tiết kiệm nhiên liệu.
  - Giảm thiểu khí thải.
  - Nâng cao hiệu suất động cơ.
  - Khởi động dễ dàng.
  - Hoạt động êm ái.
- Hạn chế:
  - Chi phí cao hơn so với hệ thống phun xăng cơ khí.
  - Cấu tạo phức tạp hơn.
  - Yêu cầu bảo dưỡng định kỳ.

## Hệ thống phun xăng điện tử





## CHƯƠNG 3. THIẾT KẾ VÀ PHÂN TÍCH HỆ THỐNG

### 3.1 Xác lập chỉ tiêu kỹ thuật

Dựa trên nguyên lý hoạt động của hệ thống thực đã trình bày và ý tưởng thiết kế hệ thống Nhúng cho đề tài trên, thực hiện phân tích và đưa ra các yêu cầu, đặc tả cho hệ thống.

#### 3.1.1 Xác định bài toán

Các hệ thống bơm xăng ở trên đều có điểm chung là: động cơ hoạt động ở các mức khác nhau với lượng xăng tiêu thụ khác nhau thì máy bơm vẫn hoạt động với công suất không đổi, lượng xăng dư thừa sẽ được quay trở lại bình xăng. Do vậy, nhóm sẽ phát triển hệ thống nhúng điều khiển bơm xăng sao cho giảm thiểu lượng xăng thừa, đáp ứng vừa đủ theo nhu cầu của động cơ.

Bài toán: Điều khiển bơm xăng sao cho áp suất trong đường ống là ổn định (bỏ qua bộ điều áp).

Nguyên lý: khi động cơ hoạt động ở các mức độ khác nhau thì lượng xăng tiêu thụ sẽ khác nhau. Khi lượng xăng tiêu thụ thay đổi sẽ làm mất cân bằng giữa lượng xăng được bơm vào từ bơm xăng và lượng xăng được lấy ra bởi động cơ, dẫn đến áp suất trong đường ống thay đổi. ECU có nhiệm vụ nhận biết sự thay đổi áp suất và tiến hành thay đổi tốc độ bơm sao cho áp suất trong đường ống quay trở về mức ổn định.

#### 3.1.2 Yêu cầu chức năng

Lựa chọn đại lượng tốc độ để đặc trưng cho động cơ và bơm. Có nghĩa là, nhu cầu của động cơ được thể hiện theo tốc độ quay mong muốn. Điều khiển tốc độ bơm để đáp ứng nhu cầu đó.

Người điều khiển có thể tương tác với hệ thống Nhúng thông qua phím nhấn đơn giản, có các chức năng như bật/tắt, tăng/giảm tốc độ.

Hệ thống có khả năng chuyển đổi tín hiệu điều khiển thành tốc độ động cơ theo lý thuyết, đo lường tốc độ thực của động cơ, từ đó tính toán, điều chỉnh tốc độ bơm xăng và các thông số như áp suất. Áp suất dao động quanh mức chấp nhận được.

Hệ thống có khả năng hiển thị các thông số tính toán lên màn hình hiển thị đơn giản.

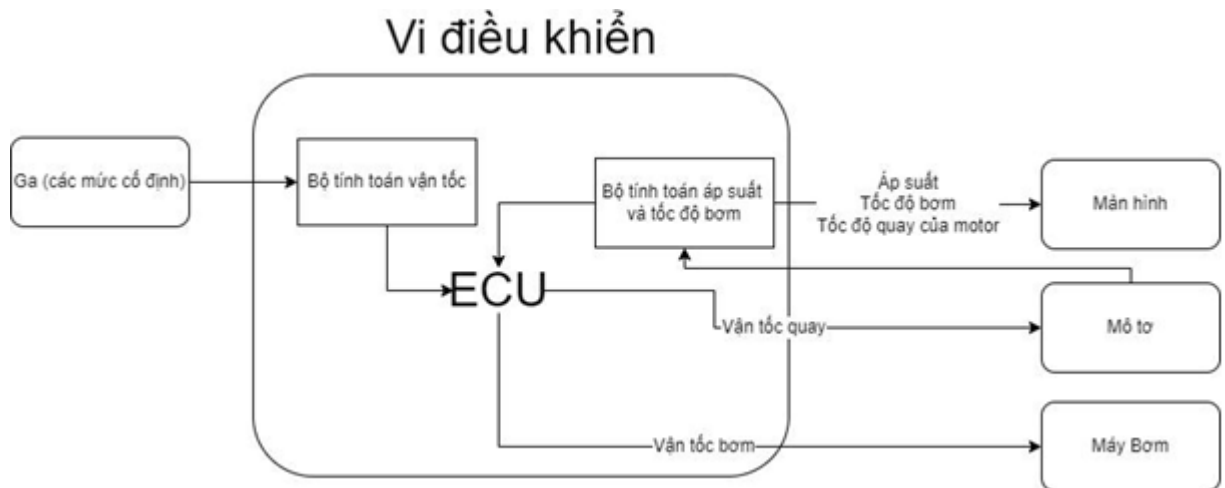
### 3.1.3 Yêu cầu phi chức năng

Hệ thống đáp ứng và phản hồi nhanh lại các tương tác chức năng.

Hệ thống hoạt động theo đúng lý thuyết và ý tưởng đề xuất, không có lỗi phát sinh.

### 3.1.3 Hệ thống dựa trên lý thuyết

Do hệ thống bơm xăng điện tử hoạt động phụ thuộc rất nhiều vào các cảm biến để làm đầu vào cho việc tính toán, cho nên một số thành phần và chức năng nằm ngoài khả năng của các thành viên nhóm. Vì vậy nhóm sẽ tối giản bài toán để dễ dàng thực hiện.



Hình 3. Xây dựng hệ thống dựa trên lý thuyết

Thay tín hiệu ga thành tín hiệu nút bấm.

Loại bỏ bộ điều áp, mô phỏng tín hiệu áp suất bằng cách tính lượng xăng bơm vào và lấy ra mỗi phút.

Sử dụng bộ encoder để tính số vòng quay của động cơ. Coi điều kiện môi trường là hoàn hảo.

### 3.1.4 Định nghĩa tham số

Bảng giá trị ban đầu

Thông số	Giá trị
Tốc độ quay của động cơ	0 – 2500 vòng/phút
Tốc độ quay của bơm	0 – 500 vòng/phút

<b>Lượng xăng tiêu thụ</b>	2l/h (với vận tốc 60km/h) = 33ml/phút
<b>Áp suất, thể tích trong 1 đoạn đường ống 10cm<sup>3</sup></b>	4 kg/cm <sup>2</sup> 10ml

Bảng giá trị vận tốc với tốc độ quay và lượng xăng tiêu thụ

<b>Tốc độ (km/h)</b>	<b>0</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>	<b>60</b>
<b>RPM (vòng/phút)</b>	0	500	1000	1500	2000	2500	3000
<b>Lượng xăng (ml/phút)</b>	0	5	10	15	20	25	30
<b>PRPM (vòng/phút)</b>	0	100	200	300	400	500	600

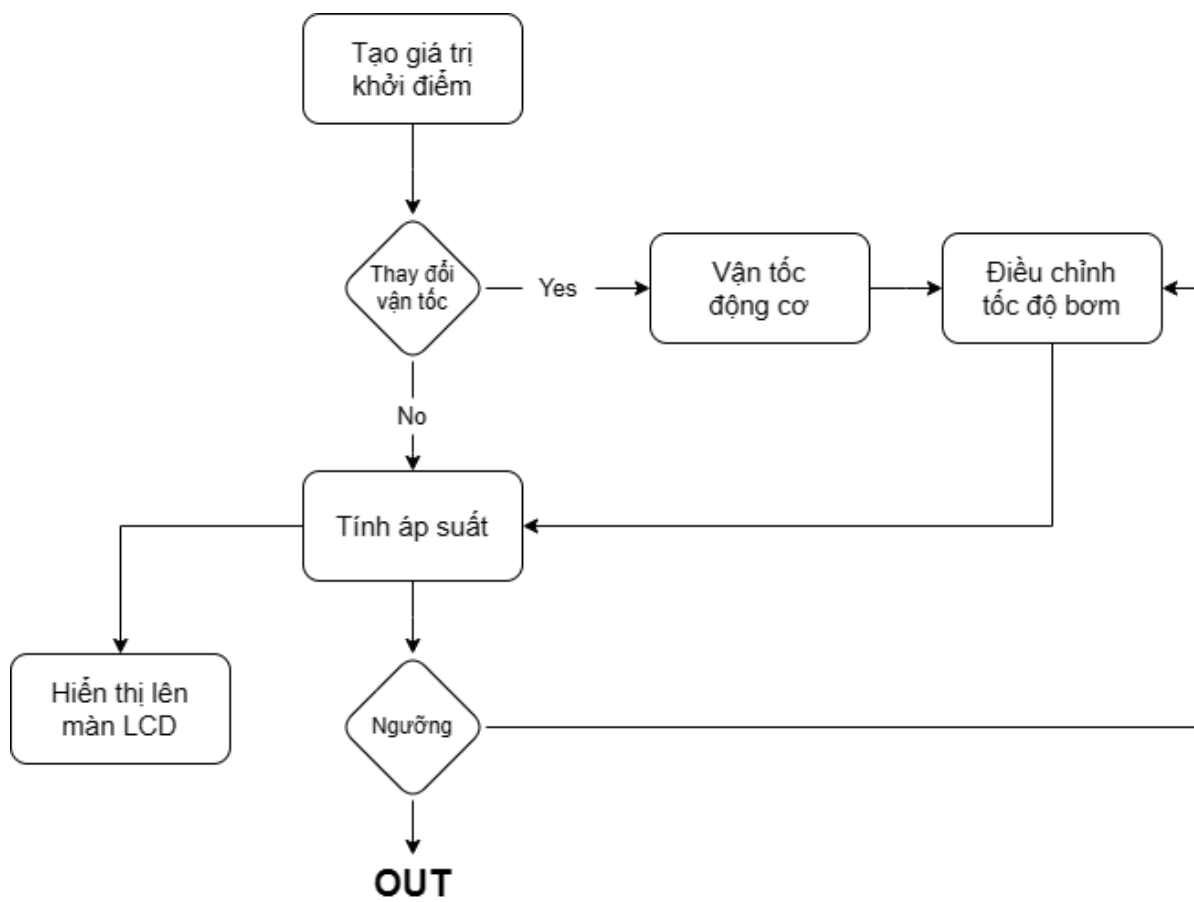
### 3.1.5 Tính toán các tham số cần thiết

- Lượng xăng tiêu thụ mỗi vòng quay: 0,01ml
- Lượng xăng hao hụt mỗi phút khi tăng 1km/h: 1ml
- Máy bơm phải quay 20 vòng để bơm thêm 1ml/phút
- Công thức tính áp suất sau mỗi phút:
- $$P' = \frac{P_a * (V - V_{tiêu\ thụ} + V_{bơm})}{V}$$
- Ngưỡng thay đổi áp suất x:

$$P_a - x = \frac{P_a(V')}{V}$$

$$P_a + x = \frac{4(V' + V_{tăng})}{V}$$

$$\Rightarrow x = 0,2$$



**Hình 4. Cơ chế cân bằng áp suất**

## 3.2 Mô hình hóa hệ thống

### 3.2.1 Sơ đồ Use-case

Các chức năng chính của hệ thống được xây dựng theo ý tưởng thiết kế, bao gồm:

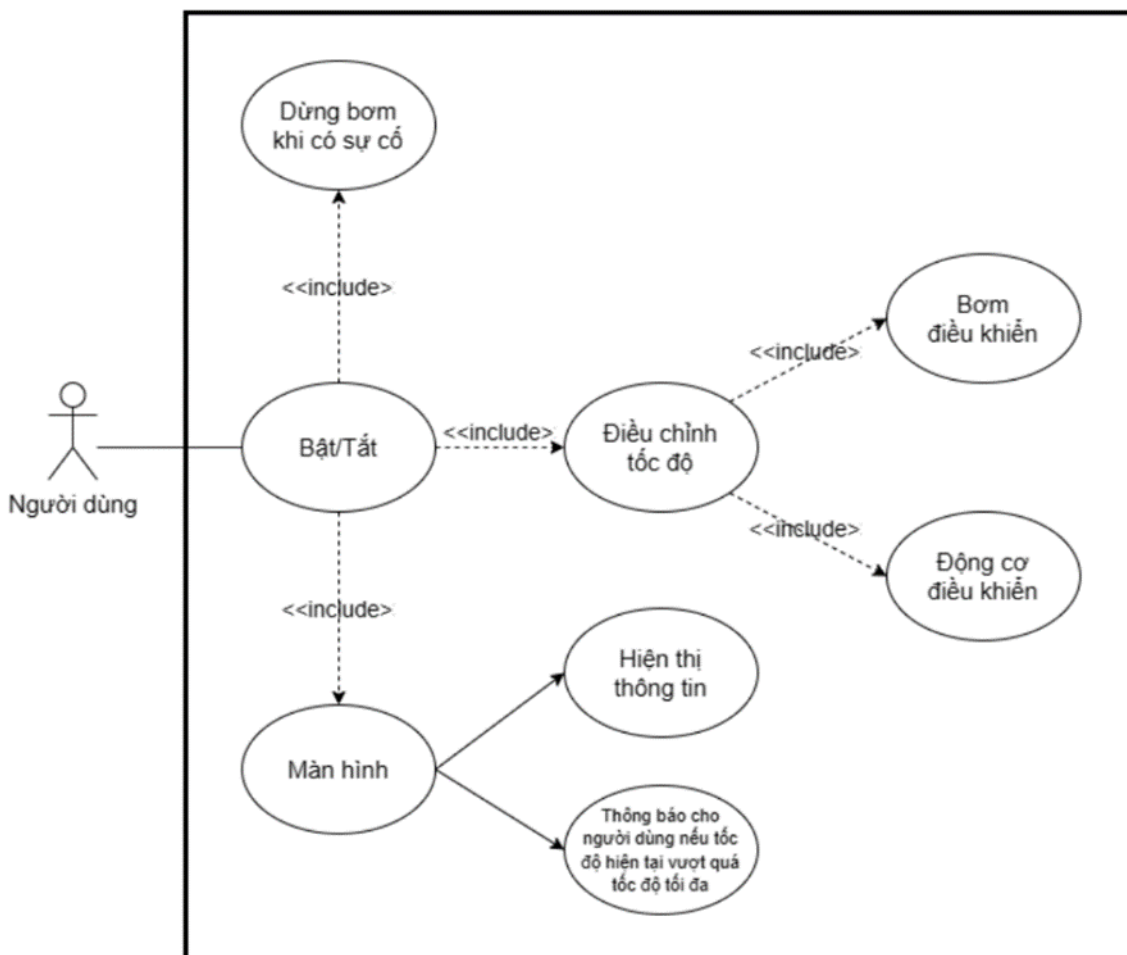
Khởi động/dừng hệ thống

Điều khiển tăng/giảm tốc độ của động cơ và bơm

Hiển thị thông số của hệ thống điều khiển bơm xăng như tốc độ, áp suất

Báo hiệu khi đạt tốc độ mong muốn

Dừng bơm khi xảy ra sự cố, tai nạn

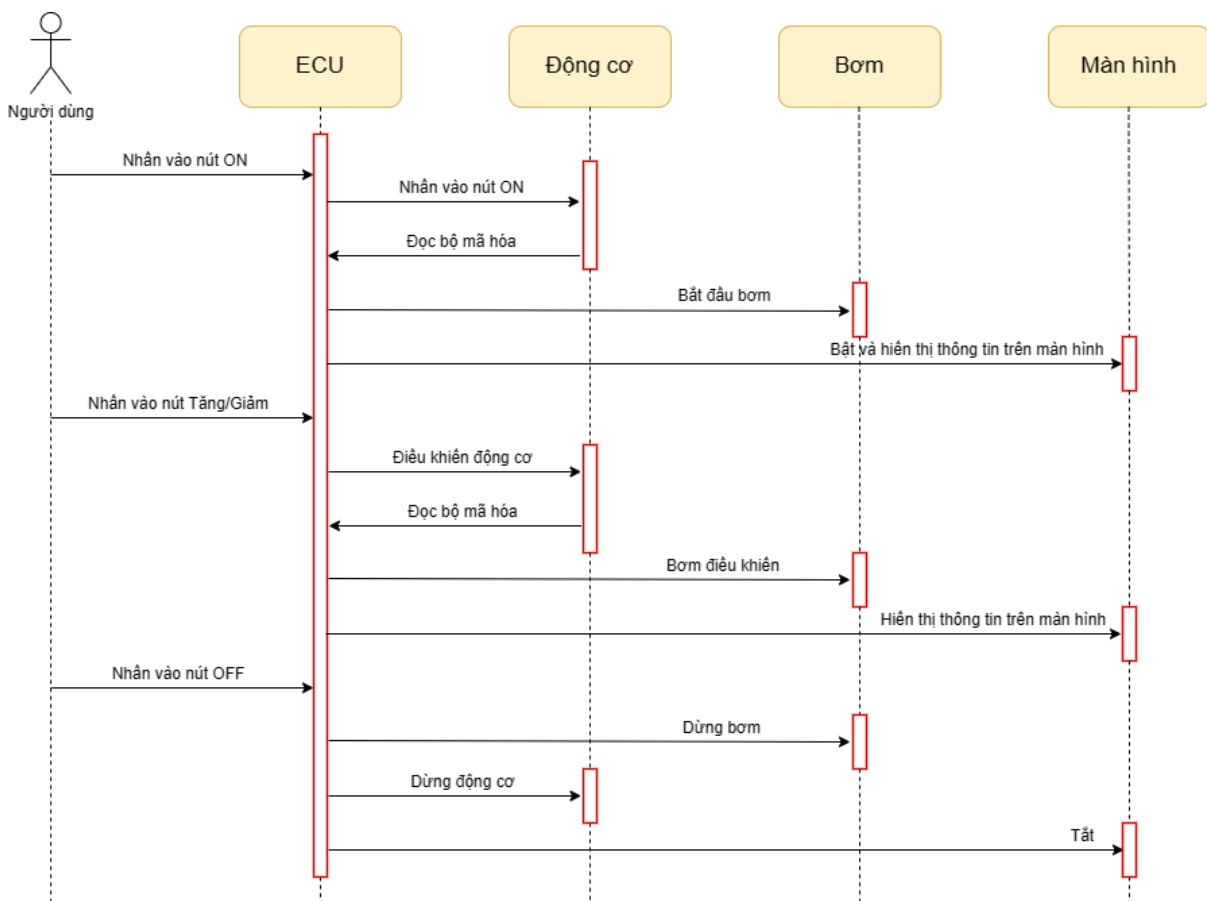


### 3.3.2 Sơ đồ tuần tự

Sơ đồ Sequence diagram thể hiện sự tương tác giữa các đối tượng trong hệ thống theo ràng buộc thời gian.

Ban đầu, khi khởi động bằng cách nhấn nút ON, bộ điều khiển ECU cấp nguồn cho bơm. Bơm quay một vài vòng để tạo áp suất cho hệ thống. Sau đó, bộ điều khiển kích hoạt động cơ quay. Encoder đo tốc độ động cơ, gửi về bộ điều khiển ECU. Sau khi tính toán cân bằng áp suất, ECU điều chỉnh tốc độ bơm, bật và hiển thị thông tin lên màn hình.

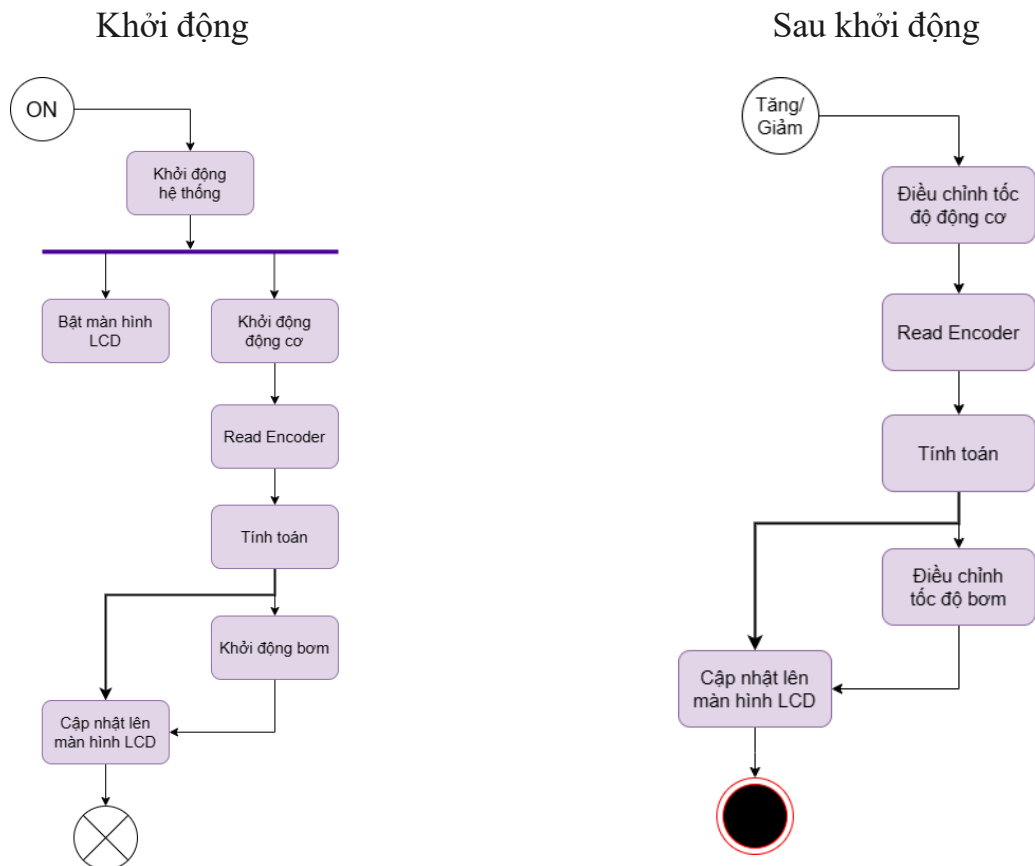
Sau khi đã khởi động, khi người dùng nhấn tăng/giảm tốc độ, bộ điều khiển điều chỉnh tốc độ động cơ theo lý thuyết. Encoder đo tốc độ động cơ, gửi về bộ điều



### 3.2.3 Sơ đồ hoạt động

Activity Diagram là sơ đồ tập trung vào mô tả các hoạt động, luồng xử lý bên trong hệ thống. Nó có thể được sử dụng để mô tả các quy trình nghiệp vụ trong hệ thống, các luồng của một chức năng hoặc các hoạt động của một đối tượng.

Dựa trên sơ đồ Use case và Sequence, xây dựng sơ đồ Activity thể hiện các luồng hoạt động bên trong hệ thống khi khởi động và khi tăng/giảm tốc độ.



### 3.2.4 Mô hình hóa bằng FSM

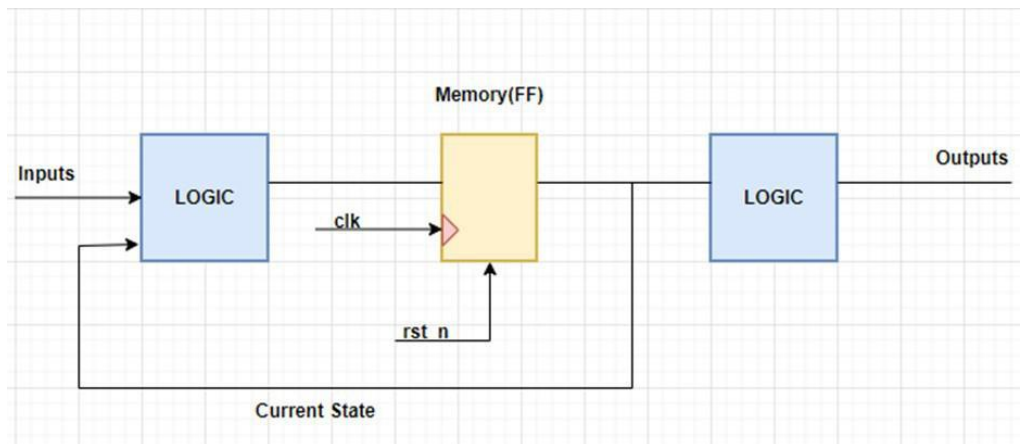
#### 3.2.4.1 Khái niệm

**Máy trạng thái hữu hạn** là mô hình toán học dùng trong thiết kế máy tính và các mạch tuần tự. Máy trạng thái hữu hạn bao gồm nhiều trạng thái có liên quan với nhau, và tại một thời điểm, máy sẽ ở trong một trạng thái duy nhất. Trạng thái mà máy đang ở, trong bất kỳ thời điểm nào gọi là trạng thái hiện tại (current state), trạng thái mà máy sẽ chuyển đến kế tiếp tính từ thời điểm hiện tại gọi là trạng thái kế tiếp.

Trong đời sống hàng ngày, chúng ta có thể bắt gặp rất nhiều thiết kế áp dụng máy trạng thái ví dụ như thang máy (elevator), đèn giao thông (traffic light)...

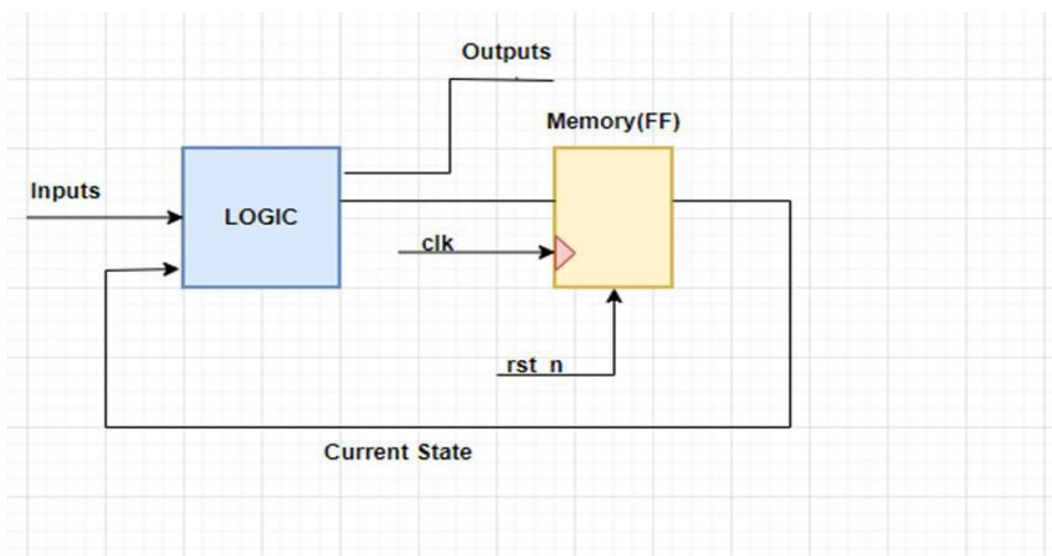
#### 3.2.4.2 Phân loại:

**Máy trạng thái Moore (Moore machine):** Máy trạng thái Moore là máy trạng thái mà dữ liệu ngõ ra được quyết định duy nhất bởi trạng thái hiện tại.



Hình 9. Moore State Machine

**Máy trạng thái Mealy (Mealy machine):** Máy trạng thái Mealy là một máy trạng thái mà dữ liệu đầu ra được quyết định bởi trạng thái hiện tại và các dữ liệu ngõ vào.



Hình 10. Mealy State Machine

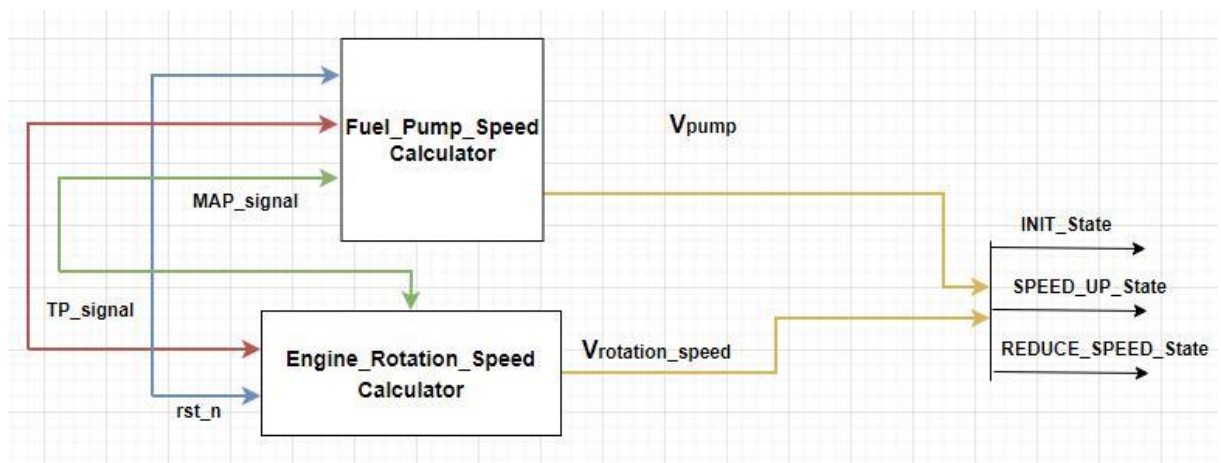


Qua hai khái niệm này, chúng ta có thể dễ dàng nhận thấy hầu hết các máy trạng thái hữu hạn trong thiết kế ASIC là máy trạng thái Mealy, khi trạng thái kế tiếp được quyết định bởi trạng thái hiện tại và dữ liệu ngõ vào

Ta đơn giản hóa bài toán hệ thống bơm xăng cho động cơ từ bình xăng, với các tín hiệu TP\_signal (tín hiệu của các mức bướm ga từ cảm biến bướm ga), MAP\_signal (tín hiệu áp suất từ cảm biến áp suất), rst\_n (tín hiệu ECU điều khiển động cơ hoạt động).

Khi các tín hiệu đi qua các bộ tính toán vận tốc bơm xăng của máy bơm xăng từ bình xăng đến động cơ và bộ tính toán vận tốc quay của động cơ ta ra được vận tốc bơm xăng và vận tốc quay của động cơ.

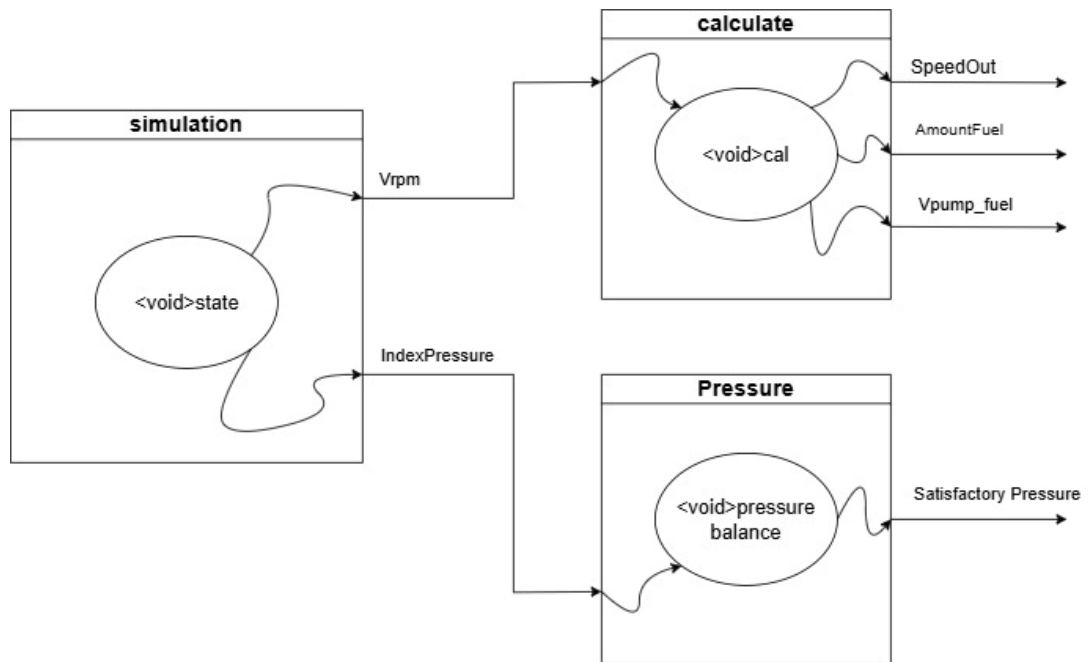
Với từng khoảng vận tốc quay của máy bơm và động cơ ta có thể chia ra làm 3 trạng thái: INIT\_State (trạng thái động cơ khởi động), SPEED\_UP\_State (trạng thái tăng tốc), REDUCE\_SPEED\_State (trạng thái giảm tốc).



Hình 11. Mô hình FSM của hệ thống điều khiển bơm xăng cho động cơ

### 3.2.3 Mô hình hóa bằng SystemC

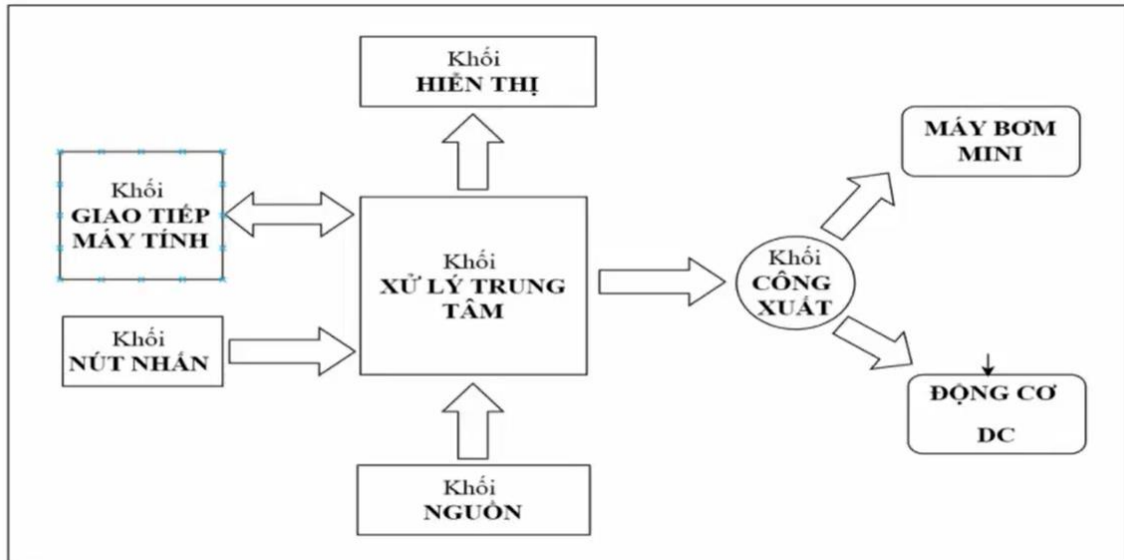
Dựa vào các dữ kiện đã phân tích tổng hợp được ta có được các Module sẽ được thể hiện trong phần mô hình hóa bằng ngôn ngữ SystemC như sau:



- Module simulation: Module này đóng vai trò sản sinh tín hiệu để các module calculate và Pressure xử lý.
- Module calculate: Module này thực hiện tính toán dựa trên dữ liệu đầu vào và xuất ra các thông số về tốc độ (SpeedOut), lượng xăng đã bơm (AmountFuel) và tốc độ cần thiết của máy bơm (Vpump\_fuel).
- Module Pressure: Module này thực hiện việc cân bằng áp suất dựa trên dữ liệu đầu vào từ IndexPressure và xuất ra áp suất ổn định (SatisfactoryPressure).

## CHƯƠNG 4. ĐỒNG THIẾT KẾ CỨNG MỀM

### 1. Sơ đồ khối hệ thống



Hệ thống được chia nhỏ thành các module thực hiện các chức năng cụ thể, như sau:

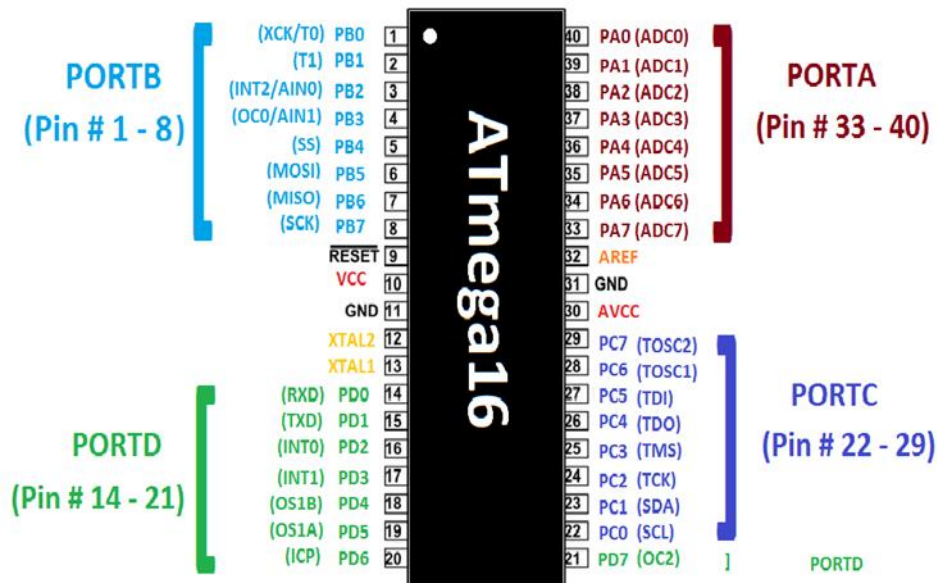
- Module phím nhấn: Thực hiện các chức năng bật/tắt, tăng tốc và giảm tốc cho hệ thống.
- Module điều khiển tốc độ quay bằng điều chỉnh độ rộng xung PWM: Thực hiện điều khiển động cơ và máy bơm. Thay đổi độ rộng xung PWM để thay đổi điện áp trung bình cấp cho động cơ và máy bơm. Độ rộng xung càng lớn thì động cơ chạy càng nhanh, bơm càng mạnh.
- Module đo tốc độ quay thực của động cơ DC bằng encoder: Khi động cơ quay, thực hiện đo tốc độ thực, đơn vị [vòng/s] hoặc [vòng/phút].
- Module tính toán cân bằng áp suất: Dựa trên tốc độ quay thực của động cơ, tính toán và điều chỉnh các thông số tốc độ bơm, áp suất, lượng nhiên liệu tiêu thụ.
- Module hiển thị các thông số hệ thống lên màn hình.

Các thông số điện của mạch:

Điện áp của mạch là 12 V. Dòng điện của mạch là 1A

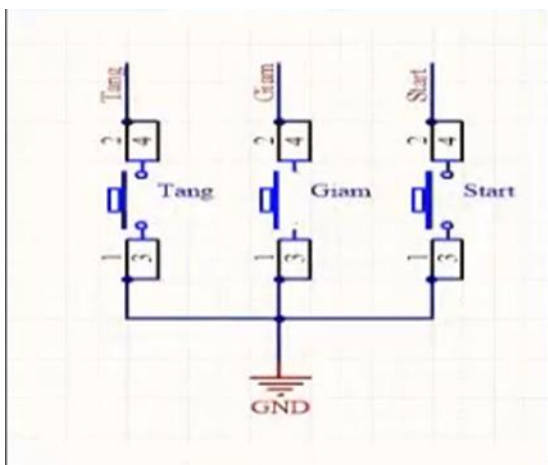
### 2. Lựa chọn giải pháp cứng mềm

## 2.1 Khối xử lý trung tâm



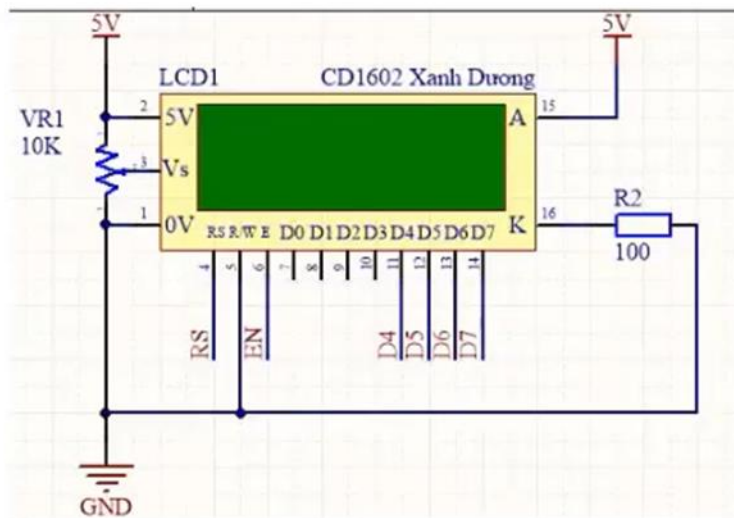
- Chọn vi điều khiển là Atmega16, thuộc họ vi điều khiển AVR
- Có 32 chân GPIO
- Điện áp hoạt động: +2.7- 5.5 V đối với Atmega16L
- Tần số hoạt động: 0-8 MHz đối với Atmega16

## 2.2 Khối phím nhấn



Gồm các nút nhấn để điều khiển động cơ và máy bơm gồm Start, Tăng, Giảm. Các nút nhấn đều có điện trở kéo lên nối với một nguồn để đặt trạng thái ngõ vào là HIGH khi không nhấn nút.

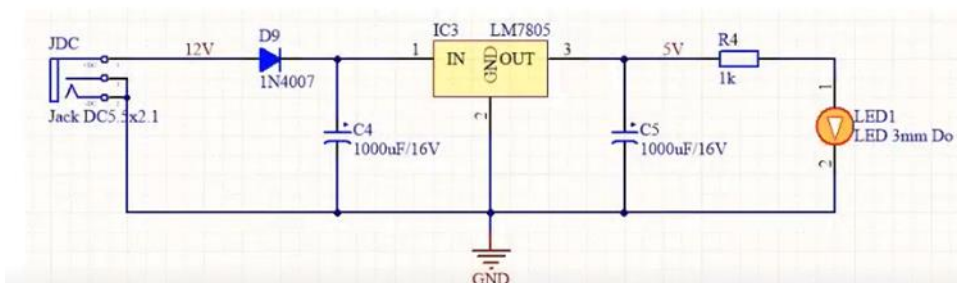
## 2.3 Khối hiển thị LCD 16x2



Chức năng hiển thị các thông số hệ thống như tốc độ động cơ, tốc độ bơm và lượng nhiên liệu tiêu thụ qua từng thay đổi tăng giảm.

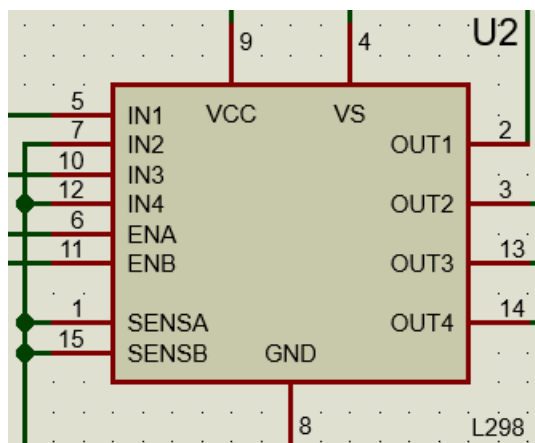
Viết drive hiển thị cho LCD file lcd.c

#### 2.4. Khối nguồn



Mạch này được dùng để bảo vệ những mạch điện hoạt động ở điện áp 5V. Nếu nguồn điện biến thiên đột ngột thì mạch vẫn sẽ hoạt động nhờ có IC 7805 giữ được điện áp ở ngõ ra OUT 5V không đổi.

## 2.5 Khối điều khiển thiết bị bằng L298N kết hợp PWM



Do khối động cơ và máy bơm chạy với điện áp 12V nên ATmega16 không thể điều khiển trực tiếp. IC L298N có nhiệm vụ điều khiển motor dựa trên tín hiệu nhận được từ ATmega16.

ATmega16 sẽ tiến hành điều khiển motor gián tiếp thông qua L298N bằng cách sử dụng Timer/Counter 1 trong Atmega16. Đây là một bộ timer/counter 16 bit với mode Fast PWM. Trong mode này, bộ timer/counter sẽ đếm từ TCNT1 đến giá trị tối đa của nó (ICR1), sau đó trở về 0 và bắt đầu lại. Ban đầu, xung sẽ ở mức HIGH, khi đếm đạt đến giá trị của thanh ghi OCR1A/B, nó sẽ chuyển xuống mức LOW; khi đạt đến giá trị ICR1, xung sẽ chuyển lại lên mức HIGH và bắt đầu lại quá trình. Qua đó, ta có thể điều chỉnh Duty Cycle của PWM.

Điểm cao nhất ICR1=60000. Tần số PWM tương ứng là 133Hz.

## 2.6 Khối đo tốc độ quay thực của động cơ

- Sử dụng Encoder xung tín hiệu (pulse signal encoder)
- Một loại encoder tạo ra các xung tín hiệu khi quay
- Các xung này được sử dụng để đo và phản hồi về vị trí, tốc độ và hướng quay của một đối tượng. Mỗi lần đối tượng quay một đơn vị (ví dụ: một góc hoặc một đoạn đường), encoder sẽ tạo ra một xung, và số lượng xung này được đếm để xác định vị trí hoặc tốc độ của đối tượng.
- Triển khai với Atmega16:

$$v_{dc} = \frac{n_s}{T_s \times N_e}$$

Trong đó:

N<sub>s</sub>: số xung tín hiệu trong T<sub>s</sub>

N<sub>e</sub>: resolution của encoder

Đơn vị: vòng/s, vòng/phút (rpm)

## CHƯƠNG 5. THIẾT KẾ GIAO TIẾP NGOẠI VI

### 5.1 Giao tiếp ngoại vi giữa Atmega16 và LCD

- **Cấu hình DDR (Data Direction Register):**

```
DDRA  |= (1<<PA7) | (1<<PA6) | (1<<PA5) | (1<<PA4) | (1<<PA3) | (1<<PA2) |
(0<<PA1) | (0<<PA0);
```

Các chân PA7, PA6, PA5, PA4, PA3, PA2 được thiết lập là đầu ra (1). Các chân PA1 và PA0 được thiết lập là đầu vào (0).

- **Cấu hình PORT (Data Register):**

```
PORTA |= (0<<PA7) | (0<<PA6) | (0<<PA5) | (0<<PA4) | (0<<PA3) | (0<<PA2) |
(0<<PA1) | (0<<PA0);
```

Tất cả các chân PA7, PA6, PA5, PA4, PA3, PA2, PA1, và PA0 đều được kéo xuống (0).

### 5.2. Giao tiếp ngoại vi giữa Atmega16 và Button

- **Cấu hình DDR:**

```
DDRB = 0x00;
```

Tất cả các chân của PORTB được thiết lập là đầu vào.

□ **Cấu hình PORT:**

PORTB = 0x00;

Không bật điện trở kéo lên, các chân này vẫn ở trạng thái thấp.

### 5.3 Giao tiếp ngoại vi giữa Atmega16 và Motor

□ **Cấu hình DDR:**

DDRD |= (0<<PD7) | (0<<PD6) | (1<<PD5) | (1<<PD4) | (0<<PD3) | (0<<PD2) | (1<<PD1) | (0<<PD0);

Các chân PD5, PD4, và PD1 được thiết lập là đầu ra (1). Các chân PD7, PD6, PD3, PD2, và PD0 được thiết lập là đầu vào (0).

□ **Cấu hình PORT:**

PORTD |= (1<<PD1);

Chân PD1 được kéo lên mức cao (1).

### 5.4 Cài đặt Timer/Counter và Interrupt

// Khai báo timer 0 mode normal chia 64 đếm 1ms

TCCR0 = (0<<CS02) | (1<<CS01) | (1<<CS00); // Prescaler = 1/64

TCNT0 = 130; // giá trị đầy của timer0;

// Khai báo timer2 mode normal chia 64 đếm 1ms

TCCR2 = (1<<CS22) | (0<<CS21) | (0<<CS20); // Prescaler = 1/64

TCNT2 = 130; // giá trị đầy của timer2

// Khai báo fast PWM của timer 1 có chu kỳ mới sử dụng ICR1

TCCR1A |= (1<<COM1A1) | (1<<WGM11) | (0<<WGM10) | (1<<COM1B1);

TCCR1B |= (1<<WGM13) | (1<<WGM12) | (1<<CS10); // Prescaler = 1

TCNT1H = 0x00; // Giá trị đầy = 0

TCNT1L = 0x00;



```

ICR1H = 0xEA; // Giá trị đỉnh = 60000, dùng để thiết lập chu kì PWM
ICR1L = 0x60;

OCR1AH = 0x00; // thanh ghi so sánh được khởi tạo tại giá trị 0
OCR1AL = 0x00;

OCR1BH = 0x00;

OCR1BL = 0x00;

TIMSK = (1 << TOIE2) | (1 << TOIE0); // bật ngắt tràn timer0 và timer2

MCUCR = (1 << ISC11) | (1 << ISC10) | (1 << ISC01) | (1 << ISC00); // cài đặt
ngắt ngoài int0 và int1 sẽ được kích hoạt ở sườn lên

```

## CHƯƠNG 6. PHÁT TRIỂN PHẦN MỀM

### 6.1 Driver LCD

Với yêu cầu hiển thị các thông số lên màn hình LCD ta cần phát triển một driver để thực hiện việc hiển thị theo mong muốn.

#### 6.1.1 Gửi lệnh tới LCD

```
void LCD_Command(unsigned char cmd) {  
    // Gửi 4 bit cao của lệnh  
    LCD_Port = (LCD_Port & 0x0F) | (cmd & 0xF0);  
    cbi(LCD_Port, RS);    // RS = 0: chế độ lệnh  
    sbi(LCD_Port, EN);    // EN = 1: kích hoạt LCD  
    _delay_us(1);        // Đợi 1 micro giây  
    cbi(LCD_Port, EN);    // EN = 0: kết thúc kích hoạt  
  
    _delay_us(100);      // Đợi 100 micro giây  
  
    // Gửi 4 bit thấp của lệnh  
    LCD_Port = (LCD_Port & 0x0F) | (cmd << 4);  
    sbi(LCD_Port, EN);    // EN = 1: kích hoạt LCD  
    _delay_us(1);        // Đợi 1 micro giây  
    cbi(LCD_Port, EN);    // EN = 0: kết thúc kích hoạt  
    _delay_ms(2);        // Đợi 2 mili giây  
}
```

Hàm này gửi lệnh (8 bit) đến LCD bằng cách chia lệnh thành hai phần: 4 bit cao và 4 bit thấp, và gửi từng phần một. (chân RS của LCD = 0)

#### 6.1.2 Gửi ký tự tới LCD

```
void LCD_Char(unsigned char data) {  
    // Gửi 4 bit cao của dữ liệu  
    LCD_Port = (LCD_Port & 0x0F) | (data & 0xF0);  
    sbi(LCD_Port, RS);    // RS = 1: chế độ dữ liệu
```

```

sbi(LCD_Port, EN);    // EN = 1: kích hoạt LCD
_delay_us(1);         // Đợi 1 micro giây
cbi(LCD_Port, EN);    // EN = 0: kết thúc kích hoạt

_delay_us(100);       // Đợi 100 micro giây

// Gửi 4 bit thấp của dữ liệu
LCD_Port = (LCD_Port & 0x0F) | (data << 4);
sbi(LCD_Port, EN);    // EN = 1: kích hoạt LCD
_delay_us(1);         // Đợi 1 micro giây
cbi(LCD_Port, EN);    // EN = 0: kết thúc kích hoạt
_delay_ms(2);         // Đợi 2 mili giây
}

```

Hàm này gửi một ký tự (8 bit) đến LCD bằng cách chia ký tự thành hai phần: 4 bit cao và 4 bit thấp, và gửi từng phần một. (chân RS của LCD = 1)

### 6.1.3 Gửi chuỗi ký tự tới LCD

```

void LCD_String(char *str) {
    int i;
    for (i = 0; str[i] != 0; i++) {
        LCD_Char(str[i]);
    }
}

```

Hàm này gửi một chuỗi ký tự đến LCD bằng cách gọi hàm LCD\_Char cho từng ký tự trong chuỗi.

### 6.1.4 Gửi chuỗi ký tự tới vị trí xác định trên LCD

```

void LCD_String_xy(char row, char pos, char *str) {
    if (row == 0 && pos < 16)
        LCD_Command((pos & 0x0F) | 0x80); // 0x80: Đặt con trỏ ở đầu dòng 1
    if (row == 1 && pos < 16)
        LCD_Command((pos & 0x0F) | 0xC0); // 0xC0: Đặt con trỏ ở đầu dòng 2
    LCD_String(str);
}

```

Hàm này đặt con trỏ tại vị trí xác định (dòng và cột) trên LCD và gửi chuỗi ký tự bắt đầu từ vị trí đó.

### 6.1.5 Bật hoặc tắt LCD

```
void LCD_OnOff(int state) {
    LCD_Command(0x02); // Đặt con trỏ về đầu
    _delay_ms(2);
    if (state == 1) {
        LCD_Command(0x0C); // Display on, cursor off
        LCD_Command(0x28); // Chế độ 4-bit với 2 dòng và font 5x7
        _delay_ms(2);
        LCD_String_xy(0, 0, "Erpm |");
        LCD_String_xy(0, 6, "Prpm |");
        LCD_String_xy(0, 13, "P ");
    }
    if (state == 0) {
        LCD_Command(0x01); // Clear screen
        LCD_Command(0x08); // Display off, cursor off
    }
}
```

Hàm này bật hoặc tắt LCD và có thể đặt lại các cài đặt hiển thị tùy thuộc vào trạng thái (state). Nếu state = 1, LCD sẽ được bật và hiển thị các tiêu đề "Erpm |", "Prpm |" và "P". Nếu state = 0, màn hình sẽ bị xóa và tắt.

## 6.2 Driver nút bấm

### 6.2.1 Nút tăng tốc

```
_delay_ms(5);

if(state == 1){

    if((PINB & (1 << button_up)) == 0) // Nếu nút tăng tốc được nhấn

    {

        *PWM1 += Temp1; // Tiến hành tăng giảm PWM theo biến Temp1

        if(*PWM1 >= 60000) // Giá trị tối đa của PWM = ICR1 = 60000

            *PWM1 = 60000;

        while((PINB & (1 << button_up)) == 0); // đảm bảo hàm if chỉ chạy 1 lần

    }

}
```

```
}
```

### 6.2.2 Nút giảm tốc

```
_delay_ms(5);  
if(state == 1){  
    if((PINB & (1 << button_down)) == 0) // Nếu nút giảm tốc được nhấn  
    {  
        if(*PWM1 > 0) // nếu PWM dương thì tiến hành giảm  
        *PWM1 -= Temp1;  
        if(*PWM1 <= 0) // giá trị bé nhất của PWM = 0  
        *PWM1 = 0;  
        while((PINB & (1 << button_down)) == 0); // đảm bảo hàm if chỉ  
        chạy 1 lần  
    }  
}
```

### 6.2.3 Nút bật tắt

```
_delay_ms(5);  
if((PINB & (1 << button_start)) == 0) // Nếu nút bật tắt được nhấn  
{  
    state ^= 1; // Đảo giá trị của biến state  
    LCD_OnOff(state); // Kết hợp với hàm LCD_OnOff của driver LCD  
    if(state == 1){ // Nếu mới bật thì mặc định PWM có giá trị = Temp1  
        *PWM1 = Temp1;  
    }  
    if(state == 0) // Nếu tắt thì PWM = 0 ➔ Động cơ tắt  
    {
```

```

        *PWM1 = 0;
    }

    while((PINB & (1 << button_start)) == 0); // đảm bảo hàm if chỉ chạy 1
lần

}

```

### 6.3 Chu trình hoạt động của hệ thống

- Hệ thống bơm xăng có 2 nhiệm vụ chính: Điều khiển tốc độ động cơ sử dụng PWM và đọc tốc độ động cơ thông qua encoder để tiến hành xử lý tín hiệu

```

ISR (TIMER2_OVF_vect)
{
    if(CountT2 >= 101) // sau 101ms
    {
        CountT2=0;

        if((PINB & (1 << button_up)) == 0) // button_up được nhấn
        {
            Speed_up(&PWM1, Temp1);
        }

        if((PINB & (1 << button_down)) == 0) // button_down được nhấn
        {
            Speed_down(&PWM1, Temp1);
        }

        if((PINB & (1 << button_start)) == 0) // On/Off được nhấn
        {
            On_Off(&PWM1, Temp1);
        }
        _delay_ms(1);

        OCR1BH = PWM1 >>8 ;
        OCR1BL = PWM1 & 0x00FF ;

    }
    CountT2++;
    TCNT2 = 130;
}

```

- Người dùng sẽ cung cấp dữ liệu cho hệ thống hoạt động thông qua nút nhấn, các nút nhấn có nhiệm vụ bật/tắt và tăng/giảm tốc độ động cơ thông qua điều chế PWM.
- Vì số lượng chân ngắt ngoài của vi điều khiển là có hạn nên ngắt ngoài sẽ được ưu tiên sử dụng với encoder của động cơ. Module nút nhấn sẽ sử dụng ngắt tràn bộ đếm thay vì ngắt ngoài, hệ thống sẽ tiến hành kiểm tra module nút nhấn mỗi 101ms.

```

ISR(INT0_vect) // ngắt ngoài int0;
{
    encoderE++; // Pulse per revolution = 200
}

ISR(INT1_vect)
{
    encoderP++; // Pulse per revolution = 200
}

ISR (TIMER0_OVF_vect) { // timer0 overflow sau 1ms
    if(CountT0 >= 100) // sau 100ms
    {
        if(encoder == 0){
            GICR &= ~(1 << INT0); // tắt int0 để lấy dữ liệu
            Rpm = encoderE * 3;
            encoderE = 0;
            GICR |= (1 << INT1); // bật int1 để tiếp tục đếm
        }
        if(encoder == 1){
            GICR &= ~(1 << INT1); // tắt int1 để lấy dữ liệu
            Prpm = encoderP * 3;
            encoderP = 0;
            GICR |= (1 << INT0); // bật int0 để tiếp tục đếm
        }
        encoder ^= 1;
        CountT0 = 0;
    }
    CountT0++;
    TCNT0 = 130;
}

```

- Encoder của động cơ và máy bơm sẽ được đọc thông qua ngắt ngoài, hệ thống trong mô phỏng sử dụng encoder với 200 xung / vòng quay.
- Tuy nhiên trong 1 thời điểm bất kỳ thì vi điều khiển chỉ có thể xử lý duy nhất 1 ngắt ngoài, vì vậy 2 ngắt ngoài cho encoder động cơ và máy bơm sẽ được bật tắt luân phiên mỗi 100ms.
- Từ dữ liệu encoder đọc được trong 100ms, ta có thể tính được số vòng quay của động cơ trong 1 phút (rpm) thông qua công thức:  $\text{encoder} * (1000 * 60) / (200 * 100) = \text{encoder} * 3$

```

void fuelPump(unsigned int step){ // điều chỉnh tốc độ máy bơm để cung cấp xăng cho động cơ
    float rightValue = Rpm / 5.0;
    if(rightValue == 0)
        PWM2 = 0;
    else if(Prpm < rightValue - 5 ){
        PWM2 += step;
    }
    else if(Prpm > rightValue + 5 ){
        PWM2 -= step;
    }
    OCR1AH = PWM2 >> 8 ;
    OCR1AL = PWM2 & 0x00FF ;
}

```

- Từ dữ liệu tốc độ động cơ tính được dựa theo encoder, ta có thể tính được tốc độ cần thiết của máy bơm để giữ cho động cơ hoạt động. Hàm trên sẽ liên tục tăng/giảm tốc độ máy bơm thông qua điều chế PWM để đạt được tốc độ cần thiết

```

void pressure_balance() // tính áp suất theo thể tích xăng được thêm vào và tiêu thụ trong 1s
{
    V_old = V_new;

    Gpm = (float) Rpm / (100.0 * 60.0);
    Pump = (float) Prpm / (20.0 * 60.0);
    V_new += Pump; // cộng thêm thể tích xăng được bơm vào
    V_new -= Gpm; // trừ đi thể tích xăng được tiêu thụ
    P = 4.0 * V_new / V_old;
}

```

- Từ tốc độ động cơ và máy bơm ta có thể tính được lượng xăng bơm vào cũng như lượng xăng được tiêu thụ. Qua đó có thể dễ dàng tính được áp suất của đường ống thông qua biểu thức toán học

```

while (1)
{
    _delay_ms(100);
    fuelPump(1000);
    pressure_balance();
    display();
}

```

- Vòng lặp hoạt động chính của vi xử lý sẽ gồm 3 hàm chính:

+ fuelPump: điều chỉnh tốc độ máy bơm cho phù hợp với tốc độ động cơ

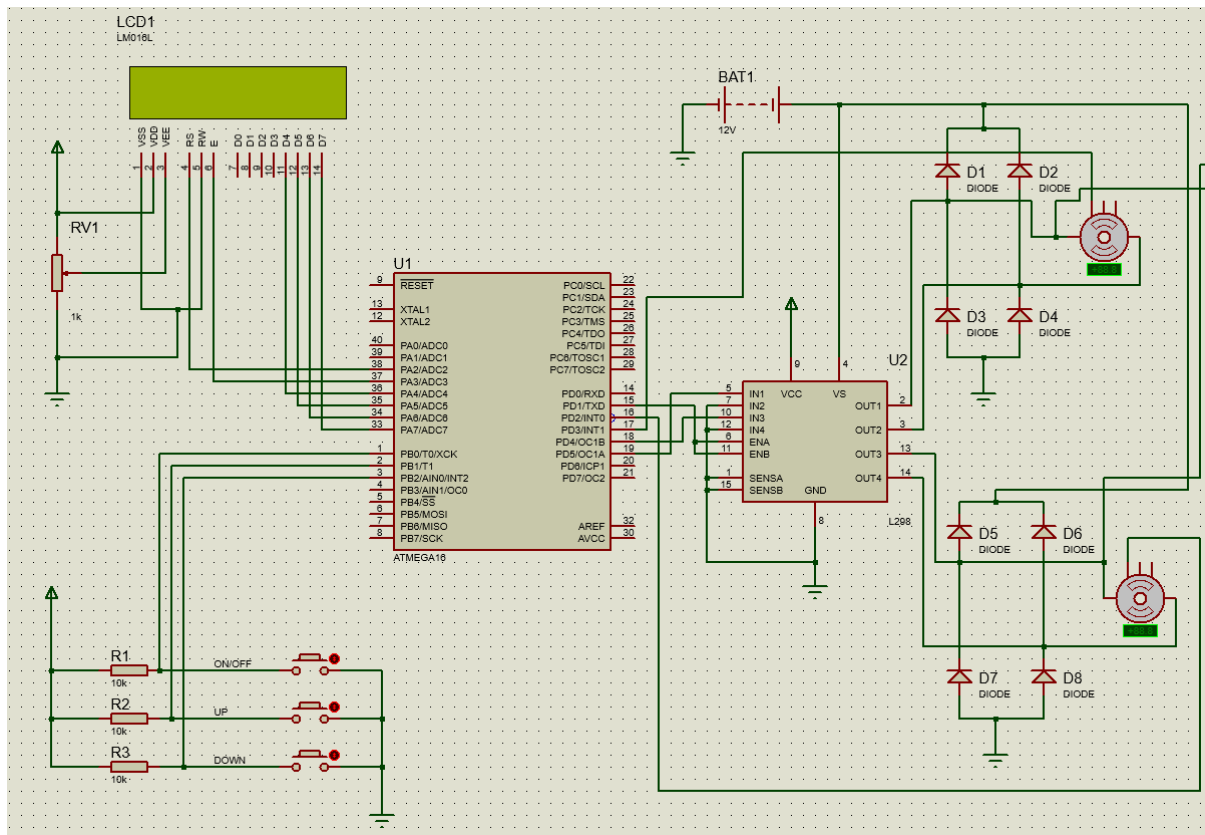
+ pressure\_balance: tính toán áp suất của đường ống thông qua biểu thức toán học

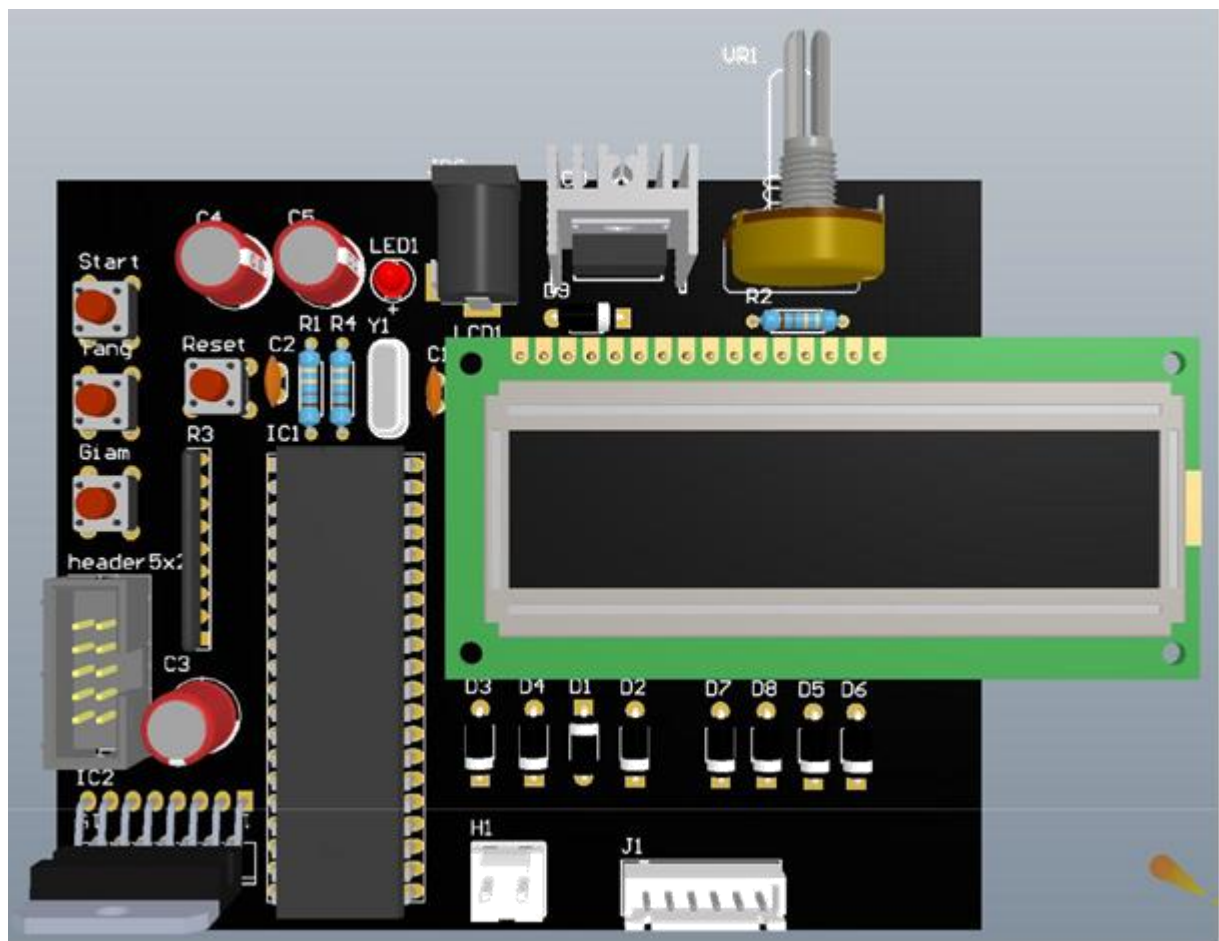
+ display: hiển thị các giá trị tính được lên màn hình LCD



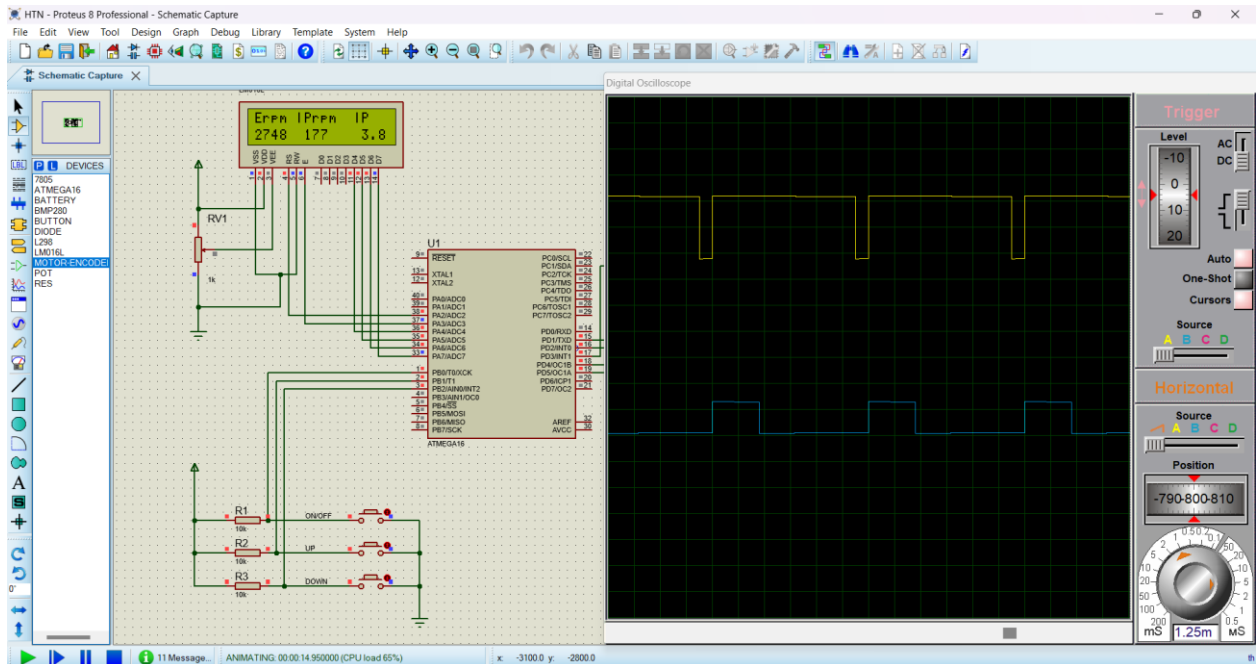
## CHƯƠNG 7. TRIỂN KHAI VÀ MÔ PHỎNG

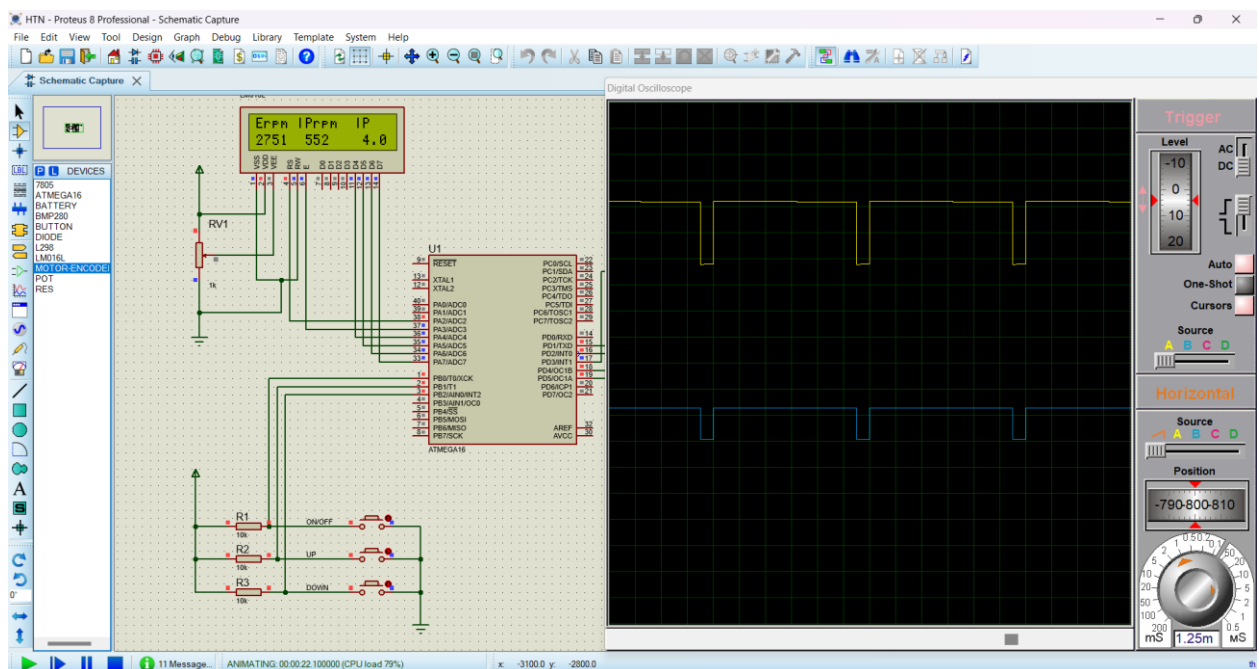
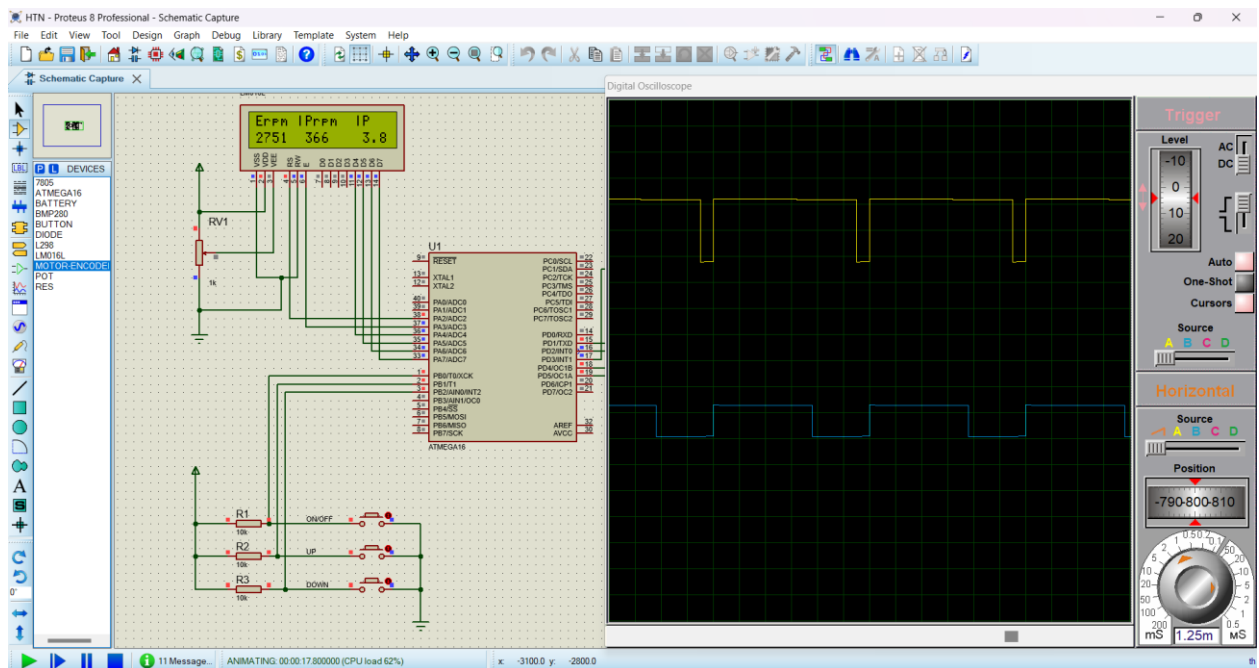
Qua các bước đã thực hiện ở các phần trên, chúng em sử dụng công cụ mô phỏng mạch Proteus để tiến hành mô phỏng lại hoạt động của hệ thống. Ta có sơ đồ mô phỏng như hình dưới:





## 7.1 Quá trình mô phỏng





- Chú thích: đường màu vàng: PWM động cơ; đường màu xanh: PWM máy bơm

- Trong trường hợp mô phỏng ở trên, tốc độ ban đầu của động cơ ở gần mức tối đa, ta tiến hành phân tích:

+ Do động cơ ở tốc độ cao mà bơm xăng hoạt động ở tốc độ thấp, nên xăng tiêu thụ nhiều hơn xăng bơm vào ==> áp suất của đường ống lúc đầu giảm xuống còn 3.8kg/cm<sup>2</sup>

+ Sau đó hệ thống liên tục thực hiện căn chỉnh tốc độ máy bơm dựa theo hàm đã lập trình, ta có thể thấy Duty Cycle và tốc độ của máy bơm dần tăng lên

+ Đến khi đạt được tốc độ cần thiết, tốc độ của máy bơm sẽ duy trì ổn định và áp suất của đường ống sẽ là mức lý tưởng  $4\text{kg/cm}^2$

➔ Trường hợp tốc độ máy bơm quá lớn so với tốc độ động cơ sẽ tương tự nhưng ngược lại

## **CHƯƠNG 8. PHÂN TÍCH, ĐÁNH GIÁ KẾT QUẢ**

Nhóm đã thực hiện được việc mô phỏng lại hệ thống bơm xăng mới, tự động điều khiển tốc độ bơm tùy thuộc vào tốc độ của động cơ.

Hệ thống đã hoạt động theo yêu cầu đặt ra ban đầu: tốc độ điều khiển theo ý muốn, áp suất được cân bằng ở ngưỡng gần  $4\text{kg/cm}^2$ . Tuy nhiên, các chức năng vẫn chưa hoạt động thật tốt, tốc độ được điều khiển bằng xung PWM không chính xác như mong muốn, áp suất và tốc độ bơm vẫn còn dao động, chưa về mức ổn định sau một khoảng thời gian.

### **Hướng phát triển**

Trong tương lai, nhóm sẽ hoàn thiện hàm cân bằng áp suất, để áp suất và tốc độ bơm trở nên ổn định sau một khoảng thời gian cân chỉnh. Nhóm cũng sẽ tìm thêm phương pháp để điều chỉnh tín hiệu đầu ra hiệu quả hơn.