

# ỨNG DỤNG NỀN TẢNG ARDUINO VÀ LABVIEW TRONG THU THẬP DỮ LIỆU ĐỘNG CƠ

## APPLYING ARDUINO AND LABVIEW PLATFORM IN AUTOMOTIVE ENGINE DATA ACQUISITION

Lê Khánh Tân

Trường Đại học Sư phạm Kỹ thuật TP.HCM, Việt Nam

Ngày toà soạn nhận bài 27/11/2018, ngày phản biện đánh giá 24/12/2018, ngày chấp nhận đăng 15/01/2019

### TÓM TẮT

Cuộc cách mạng công nghiệp 4.0 và công nghệ điều khiển thông minh qua máy tính sẽ dẫn đến việc điều khiển ô tô thông qua máy tính ngày càng phát triển mạnh. Việc điều khiển xe bằng máy tính sẽ giúp con người giảm bớt được thao tác khi vận hành, đồng thời việc xử lý các sự cố xảy ra cũng chính xác và nhanh hơn. Để làm được điều đó thì việc thu thập dữ liệu từ động cơ về máy tính là rất quan trọng, sau đó những tín hiệu này sẽ được dùng để xử lý và thực thi mệnh lệnh của người điều khiển và mục đích cuối cùng là giúp động cơ hoạt động ổn định theo mong muốn của chúng ta. Đề tài này sẽ ứng dụng phần mềm LabVIEW và Arduino để xây dựng thiết bị thu thập dữ liệu trên động cơ Toyota Yaris 2SZ-FE phục vụ cho nhu cầu giảng dạy, học tập và nghiên cứu của sinh viên.

**Từ khóa:** Hệ thống điều khiển; thu thập dữ liệu; điều khiển bằng máy tính; LabVIEW; Arduino; cảm biến; cơ cấu chấp hành.

### ABSTRACT

The 4.0 industrial revolution and computer-controlled intelligent technology will lead to the development of controlling automobiles through computers. Controlling the car by computer will help people reduce working while operating, and the treatment of the incident in the street is also accurate and faster. To do this, it is important to collect data from the automotive to the computer, then these signals will be used to process and execute the command of the operator and the ultimate goal is to keep the engine working stable according to all conditions. This project will use LabVIEW and Arduino software to build a data acquisition device on the Toyota Yaris 2SZ-FE engine for serving of teaching, learning and research of students.

**Keywords:** Control system; data acquisition; computer control; LabVIEW; Arduino; sensor; actuators.

### 1. GIỚI THIỆU VỀ ĐỀ TÀI

Với mục tiêu là tạo điều kiện cho người học được dễ dàng tiếp cận với thiết bị mới trong học tập, và rèn luyện kỹ năng kiểm tra, chẩn đoán, thu thập thông tin theo hướng ứng dụng công nghệ thông tin vào chuyên ngành ô tô.

Xuất phát từ nhu cầu thực tế trên, tác giả nghiên cứu quyết định thực hiện đề tài “**Ứng dụng nền tảng Arduino và LabVIEW**

**trong thu thập dữ liệu trên ô tô**” với mong muốn tạo ra một mô hình có thể áp dụng vào giảng dạy ngay học phần mà mình đang đảm trách.

Sản phẩm đề tài sau khi hoàn thành sẽ cung cấp cho người học mô hình thu thập dữ liệu và điều khiển từ máy tính các dữ liệu hoạt động của động cơ. Qua đó người học tiếp cận nhanh và ứng dụng tốt vào các bài thực hành và trong nghiên cứu khoa học.

## 2. THIẾT KẾ THIẾT BỊ THU THẬP DỮ LIỆU VÀ ĐIỀU KHIỂN

### 2.1 Sơ lược về động cơ Toyota Yaris 2SZ – FE và phương pháp thu thập dữ liệu

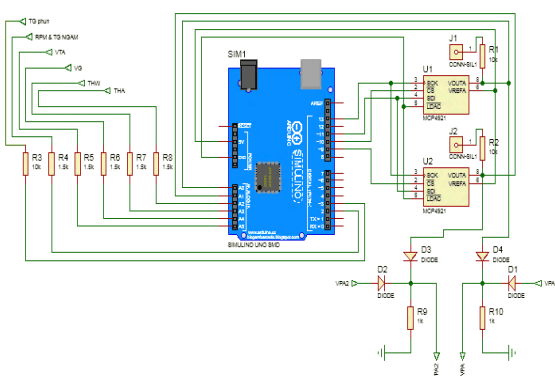
Động cơ trên xe Toyota Yaris sử dụng loại động cơ xăng, 4 kỳ, 4 xy lanh thẳng hàng. Hệ thống đánh lửa sớm điện tử (Electronic Spark Advance - ESA) sử dụng bobine đơn. Hệ thống phun xăng điện tử (Electronic Fuel Injection - EFI). [1]

Ta thực hiện thu thập tín hiệu từ các cảm biến: nhiệt độ nước làm mát, nhiệt độ khí nạp, vị trí bướm ga, cảm biến lưu lượng khí nạp, lấy tín hiệu IGT để đo thời gian ngậm điện của bobine và tốc độ động cơ, lấy tín hiệu từ một chân #10 hoặc #20 trên mỗi kim phun về ECU động cơ để đo thời gian mà kim phun nhiên liệu với mỗi chu kỳ của máy. [2]

Thu thập tín hiệu cảm biến vị trí bàn đạp ga là tín hiệu kép loại tuyến tính.

### 2.2 Thiết kế phần cứng thu thập dữ liệu

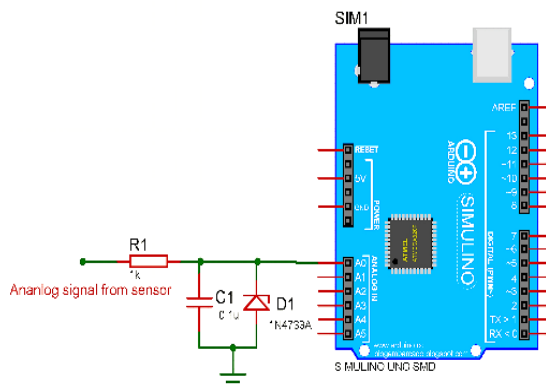
- Sử dụng board Arduino để thu thập dữ liệu từ các hệ thống trên động cơ:



**Hình 1.** Hình vẽ mô phỏng mạch điện tổng thể

Tín hiệu analog của các cảm biến vị trí bàn đạp ga số 1 (VPA) và 2 (VPA2), cảm biến nhiệt độ khí nạp (THA), cảm biến nhiệt độ nước làm mát (THW), cảm biến khối lượng khí nạp (VG), cảm biến vị trí bướm ga (VTA) được thu thập về ECU qua một con điện trở 1,5 KΩ về chân analog A0, A1, A2, A3, A4, A5 trên board Arduino. [3]

- Thiết kế cơ chế ổn định tín hiệu đầu vào và bảo vệ tín hiệu cảm biến khỏi các xung nhiễu:



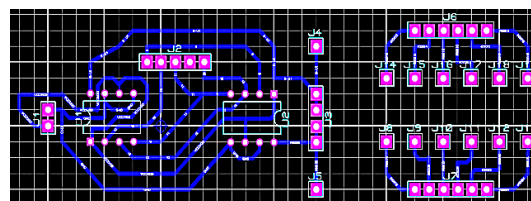
**Hình 2.** Sơ đồ thực tế mạch trích tín hiệu analog từ cảm biến về board Arduino.

Tín hiệu từ cảm biến trên động cơ được đưa qua bộ lọc thông thấp gồm R1 và C1 giúp lọc các tín hiệu xung gai cao tần, đồng thời được ghim áp bởi diode zener 5V giúp bảo vệ tín hiệu cảm biến khỏi các điện áp cao. Tổng trở kháng của mạch trích tín hiệu analog từ cảm biến trên động cơ rất lớn nên phần cứng thu thập dữ liệu chỉ đóng vai trò như thiết bị đo và lọc tín hiệu. Thực nghiệm cho thấy hoàn toàn không gây ảnh hưởng đến các thông số làm việc của động cơ.

Tín hiệu góc ngậm đánh lửa và tốc độ động cơ được lấy từ chân tín hiệu IGT từ ECU xuống bobine thông qua một con điện trở 1,5KΩ về chân digital 2 trên board Arduino.

Tín hiệu thời gian phun được lấy từ chân tín hiệu #10 hoặc #20 hoặc #30 hoặc #40 sau một trong bốn kim phun qua điện trở 10KΩ về chân digital 3 trên board Arduino. [2]

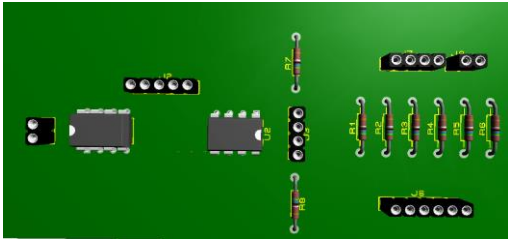
- Sử dụng board Arduino và IC DAC MCP 4921 để giả tín hiệu của cảm biến vị trí bàn đạp ga điều khiển động cơ:



**Hình 3.** Mạch in thiết kế trên Proteus điều khiển và thu thập dữ liệu

Tín hiệu analog giả tín hiệu cảm biến vị trí bàn đạp ga (VPA) được thu thập từ chân tín hiệu số 8 trên Microchip MCP 4921 thứ I được đưa vào chân A0 board Arduino.

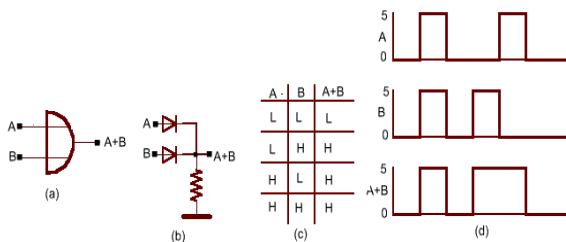
Tín hiệu analog giả tín hiệu cảm biến vị trí bàn đạp ga (VPA2) được thu thập từ chân tín hiệu số 8 trên Microchip MCP 4921 thứ II được đưa vào chân A1 board Arduino. [3]



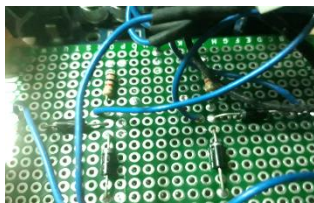
Hình 4. Mạch thiết kế 3D trên Proteus cùng 2 chip DAC MCP4921

Chân số 1 và số 6 của cả hai MCP 4921 nối với nhau và nối với chân 5V board Arduino. Chân số 2 MCP 4921 thứ I nối với chân ra số 10 board Arduino để nhận xung điều chỉnh. Chân số 2 MCP 4921 thứ II nối với chân ra số 9 board Arduino để nhận xung điều chỉnh. Chân số 3 của cả hai MCP 4921 nối với nhau và nối về chân ra 13 board Arduino. Chân số 4 của cả hai MCP 4921 nối với nhau và nối với chân ra 11 board Arduino. Chân số 5 và 7 của cả hai MCP 4921 nối với nhau và nối về chân GND board Arduino. Chân số 8 MCP 4921 thứ I đưa tín hiệu đi điều khiển giả tín hiệu chân VPA của cảm biến bàn đạp ga trên ECU động cơ. Chân số 8 MCP 4921 thứ II đưa tín hiệu đi điều khiển giả tín hiệu chân VPA2 của cảm biến bàn đạp ga trên ECU động cơ. [3]

- Thiết kế mạch OR để ghép nối và bảo vệ cảm biến bàn đạp ga:

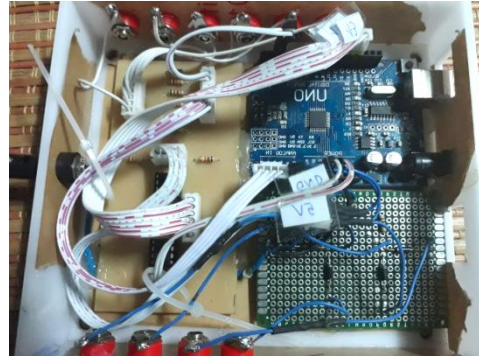


Hình 5. Các chế độ hoạt động của cổng OR.



Hình 6. Mạch "OR" thiết kế trên mô hình thật

Mỗi mạch OR gồm 2 diode và 1 điện trở 1KΩ. Mỗi đầu Anode của diốt được nối với tín hiệu đi điều khiển. Hai đầu Cathode của diode được nối với nhau và nối với một đầu của điện trở và nối xuống mass.



Hình 7. Thiết kế hộp bộ thu thập và điều khiển động cơ

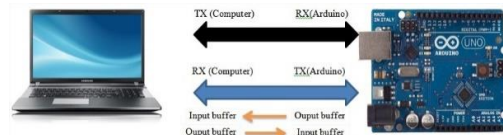


Hình 8. Thiết kế bên ngoài hộp điều khiển

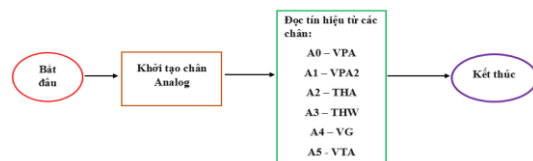
### 2.3 Thiết kế phần mềm giao tiếp thu thập dữ liệu và điều khiển từ máy tính trên nền tảng Arduino và LabVIEW

- Lập trình Arduino

Arduino sau khi thu thập dữ liệu qua các chân trên board Arduino sẽ gửi dữ liệu qua giắc nối USB type B trên Arduino của một đầu cáp nối đến đầu nối còn lại USB type A trên máy tính. [3]

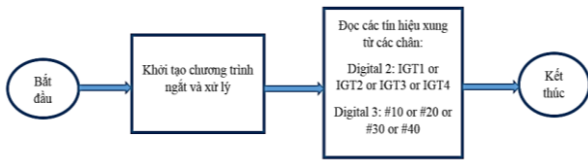


Hình 9. Giao tiếp giữa máy tính và Arduino



Hình 10. Sơ đồ khối đọc tín hiệu từ các chân tín hiệu

Các tín tiện của cảm biến sẽ đưa về các chân Analog từ A0 đến A5 trên board mạch Arduino. [3]

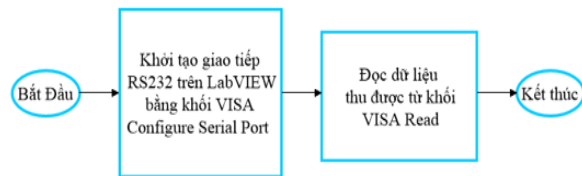


Hình 11. Sơ đồ khối đọc tín hiệu xung IGT

Tín tiện thời gian tích trữ năng lượng trên hệ thống đánh lửa (thời gian ngậm) IGT sẽ đưa về chân digital 2 để đếm xung xác định số vòng quay động cơ. Tín hiệu Điều khiển kim phun sẽ đưa về chân digital 3 trên board mạch Arduino để xác định thời gian phun.

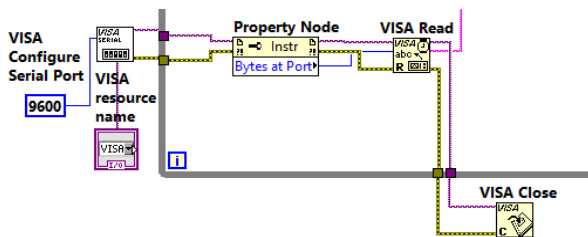
- Lập trình giao tiếp giữa board arduino và LabVIEW

Cách giao tiếp RS232 sử dụng trên LabVIEW:



Hình 12. Sơ đồ khối giao tiếp giữa Arduino và LabVIEW

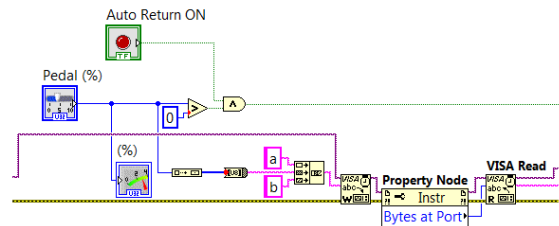
Việc lập trình trên LabVIEW sẽ bắt đầu với việc khởi tạo các khối hàm VISA. Đây là những khối hàm chức năng đã được xây dựng sẵn nhằm giúp LabVIEW có thể kết nối với các phần cứng bên ngoài. [4]



Hình 13. Mô hình khối giao tiếp VISA của LabVIEW

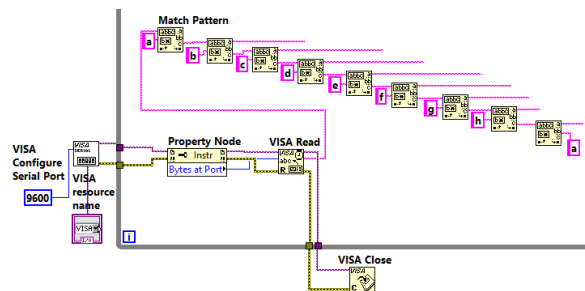
Các dữ liệu sau được truyền máy tính sẽ được chuyển đổi thành các mảng (array). [4]

Unsigned by Array → String



Hình 14. Mô tả các khối hàm chuyển đổi mảng

Sau khi nhận được tín hiệu truyền đến từ giao tiếp RS232, chúng ta sẽ bóc tách riêng biệt từng gói dữ liệu đã đặt tên trong code Arduino ra để tiến hành xử lý, đọc và hiển thị trên biểu đồ, thông số đã qua tính toán của Sub VI. [5]



Hình 15. Mô hình khối tách tín hiệu trong LabVIEW [5]

### 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

#### 3.1 Thử nghiệm với động cơ hoạt động cầm chừng

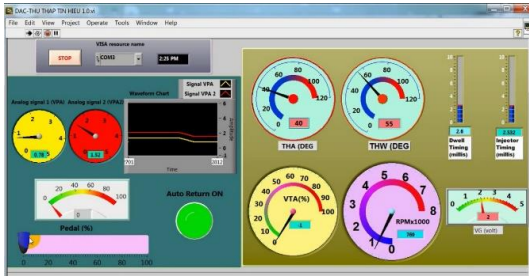
Động cơ hoạt động với chế độ cầm chừng không tải.

Bướm ga đóng hoàn toàn với góc mở 0%.

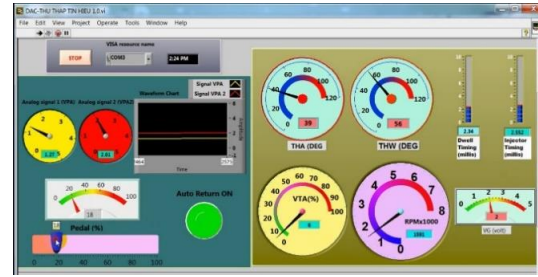
Bàn đạp ga đóng hoàn toàn với góc đạp 0%

Động cơ nằm trong xưởng động cơ với nhiệt độ môi trường tại xưởng bình thường.

Tín hiệu bàn đạp ga dạng Volt hiển thị qua biểu đồ và đồng hồ khi góc mở bàn đạp ga 0% [6]



Hình 16. Giao diện chương trình lúc không tải (VTA 0 %)



Hình 18. Giao diện chương trình lúc động cơ không tải (~1500 rpm)

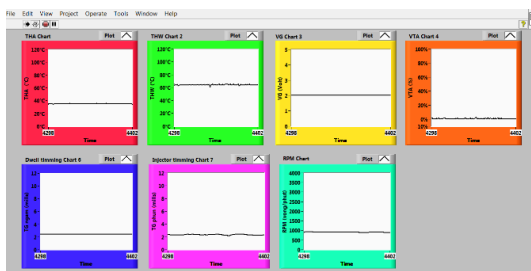
## KẾT QUẢ

Kết quả đo được thể hiện đầy đủ trên giao diện LabVIEW của máy tính với các tín hiệu đo được ở thử nghiệm 1 như sau:

- Tốc độ động cơ: ~700 rpm
- Góc mở bướm ga: ~ 0 %
- Thời gian ngậm: 2.6 millis
- Thời gian phun: ~2.6 millis
- Cảm biến khối lượng khí nạp: 2 Volt
- Góc mở bàn đạp: ~0%
- Tín hiệu bàn đạp VPA: 0.78 Volt
- Tín hiệu bàn đạp VPA2: 1.52 Volt

## Ý NGHĨA

Kiểm tra sơ bộ tất cả các tín hiệu cảm biến trên động cơ được thu thập lên máy tính.



Hình 17. Bảng đồ thị kết quả chương trình lúc không tải (VTA 0 %)

### 3.2 Thử nghiệm với động cơ hoạt động trong số vòng quay từ 1500 – 2000 vòng/phút

Xe hoạt động với chế độ không tải.

Tốc độ động cơ trong khoảng 1500 – 2000 vòng/phút.

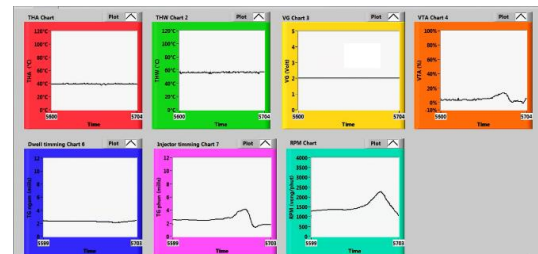
Động cơ nằm trong xưởng động cơ với nhiệt độ môi trường tại xưởng bình thường. [6]

## KẾT QUẢ

Kết quả đo được thể hiện đầy đủ trên giao diện LabVIEW của máy tính với các tín hiệu đo được ở thử nghiệm 2 như sau:

Khi kéo thanh Pedal góc đạp bàn đạp ~20%: tốc độ động cơ: ~1600 rpm, thời gian ngậm: 2.2 millis, thời gian phun 2.5 millis, VG: 2 volt, tín hiệu bàn đạp ga: VPA = 1.27 volt, VPA2 = 2.01 volt.

Khi bật công tắc Return ON: tín hiệu bàn đạp giảm dần về tại vị trí volt cảm chừng, thanh kéo bàn đạp trên giao diện LABVIEW hồi về 0 %, tín hiệu giảm về giá trị cảm chừng.



Hình 19. Bảng đồ thị kết quả chương trình lúc không tải (~1500 rpm)

## Ý NGHĨA

Cơ chế tự trả về khi bật công tắc và giữ nguyên góc bàn đạp khi công tắc off là một lợi thế trong việc điều khiển động cơ trên máy tính:

Đem lại sự chân thực khi điều khiển bàn đạp trên máy tính. Duy trì hay tăng tốc theo mong muốn mà không cần giữ lâu vào thanh bàn đạp ga trên máy tính cũng như đạp ga trên mô hình.

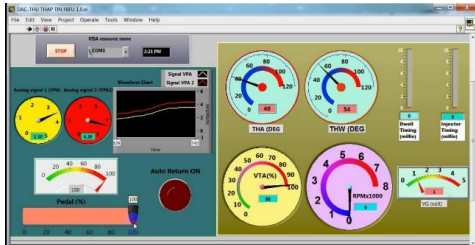
Là bước phát triển cho việc thiết kế một cơ cấu điều khiển động cơ xe qua máy tính cho người khuyết tật về chân không đạp được bàn đạp

### 3.3 Thử nghiệm mở bướm ga từ 0% đến 100% lúc động cơ không nổ máy

Điều khiển kéo bàn đạp ga trên giao diện chương trình LabVIEW

Bàn đạp ga mở 100%

Công tắc ON, động cơ không nổ máy [6]



Hình 20. Bảng điều khiển bàn đạp ga kéo lên góc mở 100%

#### KẾT QUẢ:

Khi kéo thanh kéo bàn đạp lên 100% thì tín hiệu bướm ga mở 100%.

Khi đạp bàn đạp ga mô hình, bướm ga mở 100%.

Khi nhả bàn đạp mô hình, kéo thanh mô phỏng bàn đạp trên máy tính thì tín hiệu góc mở luôn theo tín hiệu mở nào có giá trị cao hơn.

#### Ý NGHĨA

Kiểm tra được cơ chế điều khiển thuận tiện của hệ thống, có thể tùy chỉnh lựa chọn phương thức điều khiển theo mong muốn

người điều khiển và không gây hư hại cho hệ thống trên xe.

Có thể phát triển hệ thống điều khiển động cơ theo 2 phương thức điều khiển: gián tiếp bằng máy tính và trực tiếp trên xe. Đây là cơ sở để thiết kế các phương thức điều khiển cho ô tô thông minh sau này.

#### 4. KẾT LUẬN

Đề tài đã thiết kế và thi công hoàn chỉnh mô hình “Ứng dụng nền tảng Arduino và LabVIEW trong thu thập dữ liệu động cơ”. Đề tài có nhiều ý nghĩa quan trọng trong công tác đào tạo và về mặt khoa học cũng như thực tiễn.

Đáp ứng các yêu cầu giao tiếp giữa Arduino và LabVIEW thông qua giao tiếp RS232.

Thiết kế giao diện LabVIEW dễ nhìn, thông tin rõ ràng, chính xác.

Thiết kế board mạch Arduino gọn gàng, dễ dàng lắp đặt và kết nối.

Dữ liệu trên động cơ sẽ được truyền liên tục lên trên máy tính và việc điều khiển động cơ cũng trở nên dễ dàng trên máy tính. Mô hình giúp cho người học dễ dàng quan sát, thí nghiệm và vận hành trên thiết bị. Điều này có ý nghĩa đặc biệt quan trọng đối với cuộc cách mạng công nghiệp 4.0 đang bùng nổ nói chung và công nghệ xe tự hành nói riêng.

#### TÀI LIỆU THAM KHẢO

- [1] Đinh Ngọc Ân, “Trang bị điện trên ô tô máy kéo”, Nhà xuất bản đại học và trung học chuyên nghiệp Hà nội, 1980
- [2] PGS-TS. Đỗ Văn Dũng, “Trang bị điện và điện tử trên ô tô hiện đại”, NXB Đại học Quốc gia Tp Hồ Chí Minh, 2004.
- [3] Michael McRoberts, “Beginning Arduino” second edition, Technology in action, America, 2008.
- [4] National Instruments Corporation. (2006), “LabVIEW Basic Course Manual”, North Mopac, Austin, Texas.
- [5] S.Sumathi, P.Surekha, “LabVIEW based Advanced Instrumentation System” India, 2007
- [6] James D.Halderman, “Diagnosis and Troubleshooting of Automotive Electrical, Electronic and Computer System”. New Jersey, America, 2012.

#### Tác giả chịu trách nhiệm bài viết:

Lê Khánh Tân

Trường Đại học Sư phạm Kỹ thuật TP.HCM

Email: tanlk@hcmute.edu.vn