

ỨNG DỤNG THƯ VIỆN ArduinoIO PHẦN MỀM MATLAB TRONG HỆ THỐNG  
ĐIỀU KHIỂN TỰ ĐỘNG  
APPLICATION ArduinoIO TOOLBOX (MATLAB/SIMULINK) IN AUTOMATIC  
CONTROL SYSTEMS

**TS. Đỗ Trung Hải, Trần Đức Quân**

Khoa Điện - Đại học Kỹ thuật Công nghiệp

TÓM TẮT

MATLAB là một môi trường tính toán số và lập trình, được thiết kế bởi công ty MathWorks. Trong quá trình thiết kế các hệ thống điều khiển tự động, Matlab là một công cụ thiết yếu giúp người thiết kế mô hình hóa mô phỏng hệ thống, ngoài ra Matlab còn có thể là một phần trong hệ thống đó – thực hiện thuật toán điều khiển. Bản báo cáo này trình bày các bước ứng dụng thư viện ArduinoIO trong công cụ Simulink của Matlab để thực hiện thu thập dữ liệu, điều khiển hệ thống điều khiển tự động và thực hiện một ví dụ về cụ thể.

ABSTRACT

MATLAB is a numerical computing and programming environment, designed by the MathWorks. During the design of the automatic control systems, Matlab is an essential tool to help designer simulation modeling systems, besides Matlab also can be part of them – the part implements control algorithms. This report presents the steps to use ArduinoIO library in Matlab Simulink toolbox to collect data, control automated control systems and perform a specific example.

Key word: Matlab, Simulink, Arduino, ArduinoIO.

**1. Mở đầu**

**MATLAB** là một môi trường tính toán số và lập trình, được thiết kế bởi công ty MathWorks. MATLAB cho phép tính toán số với ma trận, vẽ đồ thị hàm số hay biểu đồ thông tin, thực hiện thuật toán, tạo các giao diện người dung, liên kết với những chương trình máy tính viết trên nhiều ngôn ngữ lập trình khác và truyền thông kết nối thiết bị thực qua các cổng kết nối tiếp, song song của máy tính. Với thư viện Toolbox, MATLAB cho phép mô phỏng tính toán, thực nghiệm nhiều mô hình trong thực tế và kỹ thuật.

Đặc biệt khi kết hợp với các bo mạch thu thập dữ liệu, Matlab có thể đóng vai trò là trung tâm điều khiển trong hệ thống điều khiển số.

Arduino là một hệ thống sản xuất các bo mạch mã nguồn mở được hình thành và phát triển từ năm 2005. Do các bo mạch là mã nguồn mở nên đến nay hệ thống này đã phát triển rất mạnh mẽ và có thư viện hỗ trợ cho người sử dụng rất đa dạng, phong phú. Là một thiết bị phần cứng, Arduino có thể hoạt động độc lập với chức năng thực hiện các

luật điều khiển, kết nối với máy tính, hoặc một thiết bị Arduino khác, các thiết bị điện tử khác...

Thư viện ArduinoIO là một thư viện trong bộ công cụ Simulink hỗ trợ các bo mạch Arduino làm việc với Matlab-Simulink. Kết hợp các bo mạch Arduino và thư viện ArduinoIO, Matlab có thể tiến hành thu thập dữ liệu, thực hiện thuật toán điều khiển dễ dàng để điều khiển đối tượng thực.

**2. Bo mạch Arduino và thư viện ArduinoIO**

**2.1. Bo mạch Arduino**

Các bo mạch **Arduino** thật ra là bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

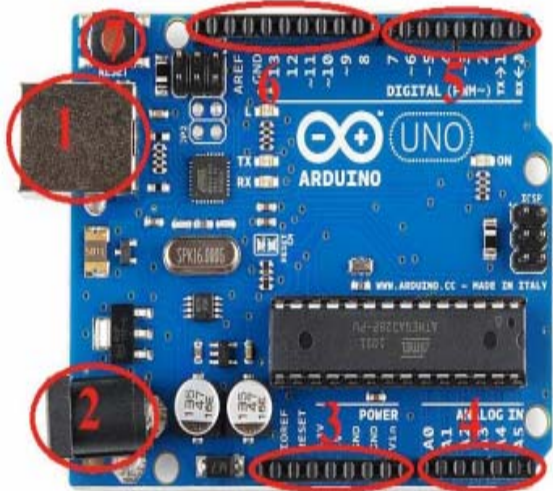
Bộ mạch ArduinoUNO là bộ mạch thông dụng nhất.

ArduinoUno sử dụng chip Atmega328. có 14 chân vào/ra số, 6 chân vào tương tự, thạch anh dao động 16Mhz.

**Một số thông số kỹ thuật như sau:**

Vi điều khiển	Atmega328
Điện áp hoạt động	5V
Nguồn cấp	7-12V
Số đầu vào/ra số	14 (6PWM)
Đầu vào tương tự	6
Dòng điện vào/ra số	40 mA
Bộ nhớ chương trình	32 KB
Xung nhịp	16 MHz

**Sơ đồ chân ArduinoUNO:**



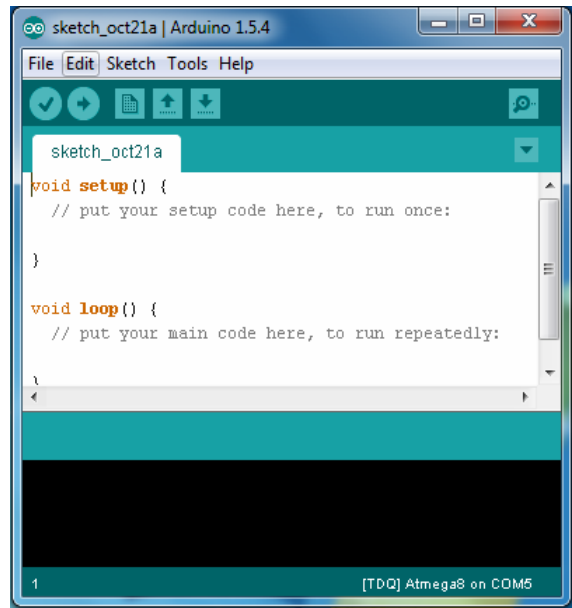
Hình 1: ArduinoUNO

- USB (1): Arduino sử dụng cáp USB để giao tiếp với máy tính. Thông qua cáp USB chúng ta có thể Upload chương trình cho Arduino hoạt động, ngoài ra USB còn là nguồn cho Arduino.
- Nguồn cấp một chiều cho Arduino UNO (2,3), 7÷12V
- Đầu vào tương tự (4), A0÷A5.
- Đầu vào/ra số (5,6), D0÷D13.

**Môi trường lập trình:**

Môi trường lập trình cho Arduino được tải về từ trang web <http://arduino.cc/en/Main/Software>. Sau

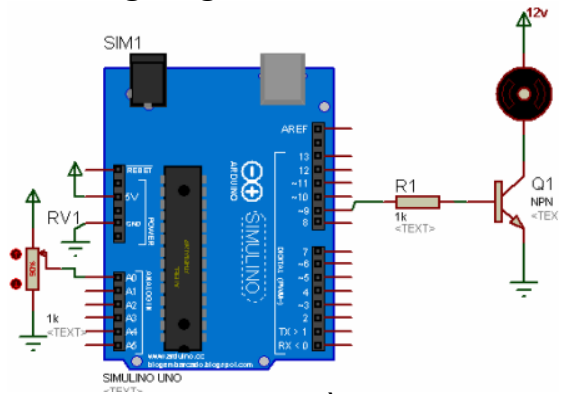
khi cài đặt xong thì giao diện chương trình như sau:



Hình 2: Môi trường lập trình Arduino

Để tìm hiểu lập trình cho Arduino có thể tìm hiểu qua các ví dụ và phần trợ giúp chi tiết trong Arduino.

**Ví dụ điều khiển tốc độ động cơ một chiều bằng xung PWM:**



Hình 3: Sơ đồ mạch

Mã nguồn:

```
int potPin = 0;
int transistorPin = 9;
int potValue = 0;
void setup() {
  pinMode(transistorPin, OUTPUT);}
void loop() {
  potValue = analogRead(potPin) / 4;
  analogWrite(transistorPin, potValue);
}
```

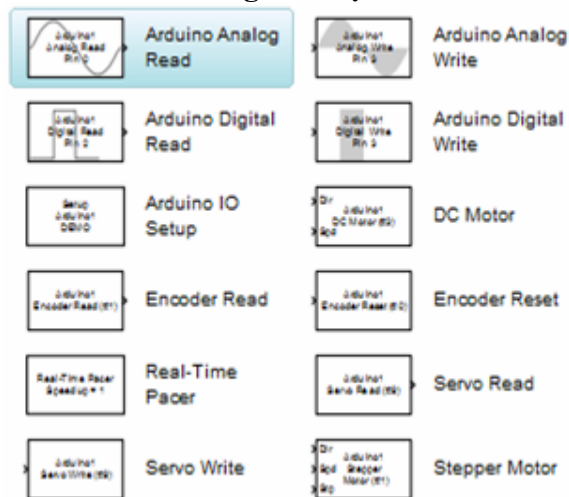
## 2.2. Thư viện ArduinoIO

Thư viện **ArduinoIO** gồm hai gói phần mềm mã nguồn mở: gói mã lập trình ngôn ngữ cho bo mạch Arduino và gói mã lập trình .m cho các khối trong môi trường Simulink. Để sử dụng được thư viện này, Matworks khuyến cáo sử dụng phiên bản Matlab 2012a trở lên.

### Các bước cài đặt thư viện ArduinoIO:

- Tải và giải nén thư viện ArduinoIO từ trang web <http://mathworks.com>.
- Tải gói phần mềm xuống bo mạch ArduinoUNO. Gói phần mềm trong thư mục ArduinoIO/pde.
- Thêm thư viện ArduinoIO cho Matlab/Simulink: Đưa thư mục làm việc của Matlab đến thư mục ArduinoIO. Chạy tệp install\_arduino.m để thêm thư viện ArduinoIO cho Simulink.

### Các khối trong thư viện ArduinoIO:



Hình 1: Thư viện ArduinoIO

- Khối chức năng Arduino IO setup: thiết lập cài đặt giao tiếp với Arduino. Khi kết nối Arduino vào máy tính sẽ tạo ra một cổng giao tiếp nối tiếp (ví dụ Com3, Com4, ...). Người sử dụng phải khai báo cho Matlab biết Arduino được kết nối vào cổng giao tiếp nào.
- Khối chức năng Real-Time Pacer: Cài đặt cho Simulink chạy với thời gian thực.
- Khối chức năng Arduino Analog Read: đọc giá trị ADC trên các đầu vào analog của Arduino. Do bo mạch ArduinoUNO có thể

biến đổi điện áp tương tự 0÷5V từ đầu vào tương tự A0÷A5 thành giá trị số 10bit nên khối này sẽ nhận được kết quả từ 0÷1024 tương ứng với giá trị điện áp ở các đầu vào tương tự được khai báo.

- Khối chức năng Arduino Digital Read: đọc giá trị các đầu vào số của Arduino. Kết quả khối này có thể là 0 hoặc 1 theo đầu vào số được khai báo.

- Khối chức năng Arduino Digital Write: ghi giá trị 0 hoặc 1 ra các đầu ra số được khai báo.

- Khối chức năng Arduino Analog Write: xuất giá trị tương tự trên các đầu ra tương tự của Arduino. Arduino coi các chân có chức năng điều khiển PWM như là các chân xuất ra được tín hiệu tương tự. Do Arduino sử dụng thanh ghi 8bit để điều khiển PWM nên giá trị của khối Arduino Analog Write nhận được từ 0÷255 tương ứng với xung PWM có độ rộng xung từ 0÷100%. Tần số PWM của ArduinoUNO là 980Hz.

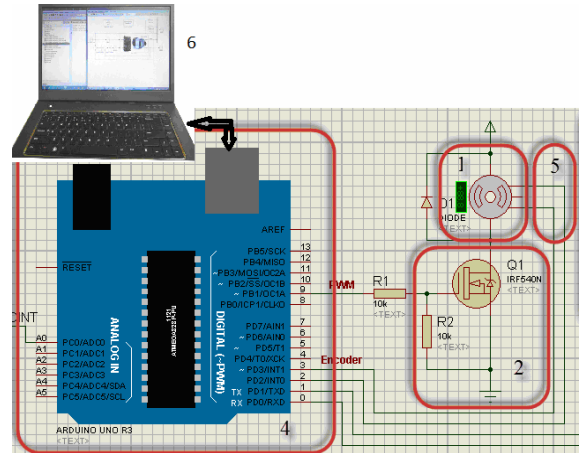
- Khối chức năng Encoder Read: thiết lập và đọc giá trị bộ đếm xung của Arduino. Thư viện ArduinoIO hỗ trợ cảm biến tốc độ mã hóa dưới dạng xung (Encoder) loại tương đối 2 kênh lệch pha nhau 90° điện. Trên bo mạch ArduinoUNO có chân 2 và 3 hỗ trợ nhận tín hiệu xung từ Encoder. ArduinoUNO sẽ tăng hoặc giá trị đếm khi có sự thay đổi trạng thái của tín hiệu xung Encoder tùy theo chiều quay của đĩa Encoder. Do đếm theo sườn xung như vậy nên ArduinoUNO đã thực hiện tăng độ phân giải của Encoder lên 4 lần. Kết quả của khối này là số xung ArduinoUNO đếm được trong 100ms.

- Khối chức năng Encoder Reset.
- Khối chức năng DC Motor: điều khiển động cơ một chiều. Khối này yêu cầu phải sử dụng bo mạch điều khiển động cơ một chiều của Arduino.
- Khối chức năng Stepper Motor: điều khiển động cơ bước. Khối này yêu cầu phải sử dụng bo mạch điều khiển động cơ bước của Arduino.
- Khối chức năng Servo Read, Servo Write: điều khiển động cơ servo.

### 3. Sử dụng bo mạch ArduinoUNO và thư viện ArduinoIO điều khiển tốc độ động cơ một chiều

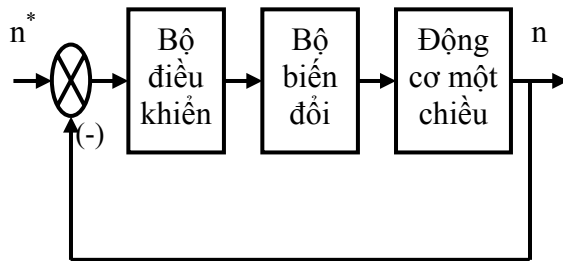
#### 3.1. Thông số động cơ

Hãng sản xuất	YASKAWA
Mã hiệu	UGFMED 03SR121
Điện áp định mức	24V
Công suất định mức	50W
Tốc độ định mức	1500(vòng/phút)
Encoder	5V, 400 xung/vòng



Hình 5: Sơ đồ nguyên lý hệ thống

#### 3.1. Sơ đồ cấu trúc điều khiển hệ thống



Hình 4: Sơ đồ cấu trúc hệ thống

#### 3.2. Sơ đồ nguyên lý hệ thống

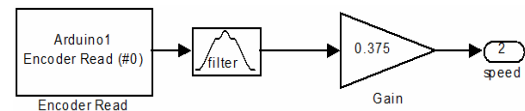
1. Động cơ một chiều
2. Bộ biến đổi xung áp
3. Mạch tạo tín hiệu đặt
4. Bộ ghép nối Arduino
5. Tín hiệu phản hồi tốc độ
6. Máy tính (Matlab/Simulink)

#### 3.3. Xây dựng cấu trúc điều khiển hệ thống sử dụng Matlab/Simulink

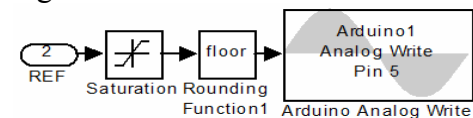
Sử dụng các khối trong thư viện ArduinoIO để xây dựng cấu trúc điều khiển hệ thống với bộ điều khiển được thực hiện trên Matlab/Simulink trên hình 6.

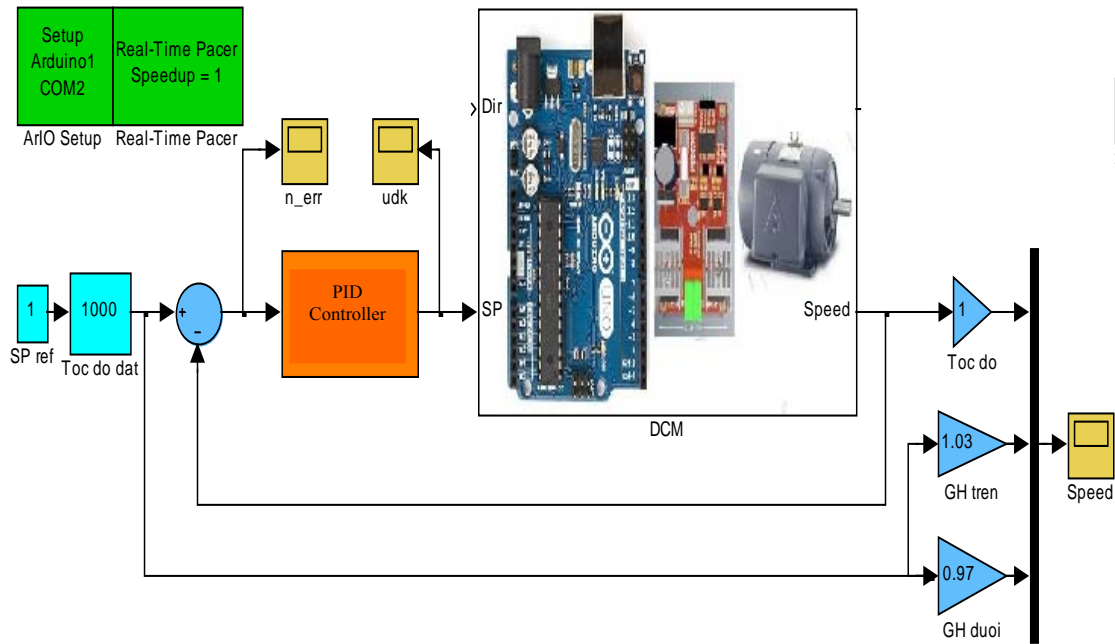
Chi tiết khối Động cơ một chiều (DCM) gồm:

- Đọc tín hiệu tốc độ động cơ, sử dụng khối Encoder Read.



- Xuất tín hiệu từ bộ điều khiển ra bộ biến đổi (xuất tín hiệu PWM), sử dụng khối AnalogWrite.

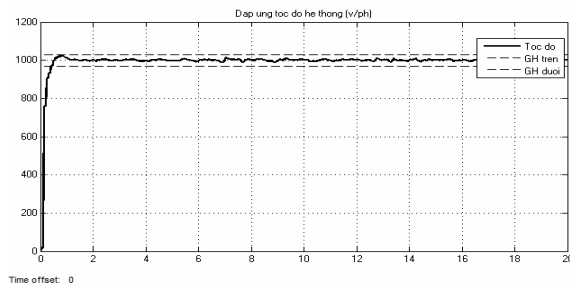




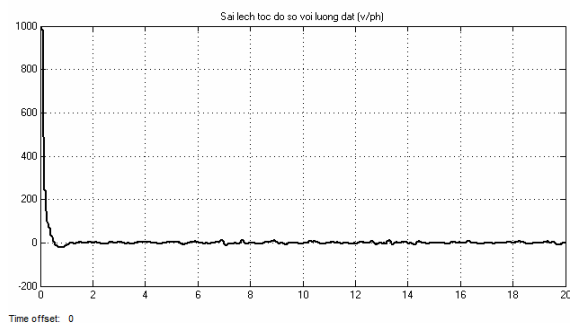
Hình 6: Sơ đồ cấu trúc điều khiển hệ thống sử dụng Matlab/Simulink

### 3.4. Đáp ứng tốc độ của động cơ

Đáp ứng tốc độ động cơ với tín hiệu đặt  $N_{ref} = 1000$  v/ph.



Hình 7. Đáp ứng tốc độ động cơ  
Sai lệch tốc độ động cơ so với tín hiệu đặt:



Hình 8. Sai lệch tốc độ động cơ khi không tải với tín hiệu đặt

Từ hình 7 và hình 8 ta thấy:

- Đáp ứng tốc độ động cơ khi luôn bám theo tín hiệu đặt.
- Thời gian xác lập nhanh, khoảng 0.6s.
- Lượng quá điều chỉnh rất nhỏ.
- Bộ điều khiển thực hiện trên Matlab/Simulink điều khiển hệ thống đạt chất lượng tốt.

### 4. Kết luận

Từ các kết quả nghiên cứu và thực nghiệm ở trên ta thấy: Kết hợp Matlab/Simulink và bo mạch Arduino với thư viện ArduinoIO có thể thực hiện tốt việc thu thập dữ liệu, tính toán và điều khiển trong các hệ thống điều khiển tự động.

Ngoài ra, thư viện ArduinoIO là thư viện mã nguồn mở nên người dùng hoàn toàn có thể chỉnh sửa, thêm, bớt các công cụ cần thiết cho từng ứng dụng cụ thể. Có thể khai thác triệt để khả năng tính toán mạnh mẽ và thực hiện các thuật toán phức tạp của Matlab để điều khiển các hệ thống điều khiển tự động phức tạp, yêu cầu lượng tính toán lớn.

## TÀI LIỆU THAM KHẢO

[1]. Bùi Quốc Khánh – Phạm Quốc Hải – Dương Văn Nghi, *Điều chỉnh tự động truyền động điện*, NXB Khoa học và kỹ thuật, 1999.

[2]. Nguyễn Phùng Quang, *Matlab & Simulink dành cho kỹ sư điều khiển tự động*, NXB Khoa học và kỹ thuật, 1996.

[3]. Arduino, <http://arduino.cc>

[4]. Mathworks, <http://mathworks.com>