Int.J.HIT.TRANSC:ECCN.Vol.7: Issue 2A (2021) Page 22-30 ISSN: 0973-6875

Available Online at www.hithaldia.in/locate/ECCN
All Rights Reserved

ORIGINAL CONTRIBUTION

Speed Control of a PMDC Motor using MATLAB with Arduino Interface

P. Roy[1], D. Dey[2] and S. Biswas[3]

[1] *Electrical Engineering, Haldia Institute of Technology*

[2] *Electrical Engineering, Haldia Institute of Technology*

[3] *PG Student, Electrical Engineering,IIEST (Shibpur)*

ABSTRACT

A closed loop speed control scheme for permanent magnet direct current (PMDC) motor is presented in this article. A Proportional plus Integral (PI) controller is designed in MATLAB Simulink. Control signal received from the computer is fed to an Arduino microcontroller board that generates the necessary pulse width modulated (PWM) signal that drives the PMDC motor through a driver IC. Speed of the motor shaft is received through a tacho-generator which is fed back to the controller block in MATLAB. The controller has been designed in such a way that the motor speed can be maintained at the set reference value even if there are sudden changes in load. Hardware experimental results are reported to demonstrate performance of the developed scheme.

KEYWORDS—PMDC motor, PI controller, L293D chip, Arduino UNO microcontroller.

1. INTRODUCTION

PMDC motor is used in specific applications where high starting torque is required at low speeds. In comparison to conventional DC motors, PMDC motor has few advantages such as no requirement of field excitation arrangement, small size of motor due to the absence of field coil, improved efficiency due to absence of field excitation, cheaper and economical for fractional kW rated applications.As the magnetic flux in the air gap is fixed and cannot be controlled externally, so speed of a PMDC motor can only be varied by armature control[1, 2].

Over the years, several speed control schemes for PMDC motors have been proposed by different researchers. Speed control of PMDC motor or other types of DC motor can be controlled by different controllers, such as fuzzy logic controller [3, 6], PI controller to achieve the controlling signal with different types of microcontroller [3-7]. If fuzzy controller is used to control the speed of DC motor, then much more manual adjustment by trial and error method will be required to reach the optimum performance level as fuzzy controller uses asymmetric membership function. On the other hand, the manual adjustment can be avoided by using PI controller. Although PI controller generates superior performance [7], but if hardware components are used to transfer the controller signal to DC motor, it will not be cost effective. To make the PI controller economically feasible, Arduinomicrocontroller has been used in the present work to generate required PWM signal, so that same level of performance can be achieved at lower hardware cost and complexity.

In this work, the conventional armature speed control method is used, but with a new technique. The requirement of hardware control circuit is eliminated as the control circuit is designed in MATLAB and that controls the

*Corresponding Address: *piyajun@gmail.com*

PMDC motor using Arduino, a microcontroller with a programmable suite. It is mostly used in robotics to control the forward, backward or other type of motions of DC motor [8]. In order to keep the motor speed constant under variable mechanical load, the speed control circuit in MATLAB needs to be designed with the proper feedback mechanism. Designing the control circuit in MATLAB eliminates the power loss in the control circuit and also provides the opportunity to design a wide range of different control circuit configurations without further investments for making different hardware models of control circuit.

This article reports the results of hardware implementation of a PMDC drive system that attempts to maintain the speed at a reference value in response to varying loads. A closed loop (Proportional-Integral) PI control circuit is designed using MATLAB Simulink and control signals from the PC is used to control a PMDC motor through Arduino[9] interface with L293D (motor driver chip) [10].

2. MODELLING OF PMDC MOTOR FOR SPEED CONTROL

To work with the speed control scheme of a PMDC motor, the understanding of the mathematical model of the motor and the appropriate controlling method is necessary.

PMDC motor model

This section describes the method used to identify the final transfer function of a PMDC motor from its equivalent circuit parameters [11], as shown in Figure. 1.

From the equivalent circuit as shown in Figure. 1, the armature supply voltage (V_a) can be related to the back emf (E_g) as:

$$V_a = E_g + I_a R_a + L_a \frac{dI_a}{dt} \quad (1)$$

Where, I_a is armature current (A), R_a and L_a are armature resistance (Ω) and inductance (H) respectively.

Torque equation of the PMDC motor is given by

$$T_d = J\frac{d\omega}{dt} + B\omega + T_L \quad (2)$$

Here, T_d is torque developed (Nm), T_L is load torque (Nm), J is moment of inertia (Kg/m^2), B is friction coefficient and ω is angular velocity (rad/s) of the motor.

Assuming the value of rotor friction of the motor to be negligible, consider B=0. Therefore, the torque equation (2) can be simplified as:

$$T_d = J\frac{d\omega}{dt} + T_L \quad (3)$$

As the motor is of PMDC type, therefore the flux φ is constant. Then the back emf of the motor can be expressed as

$$E_g = K\omega \quad (4)$$

And torque is

$$T_d = KI_a \quad (5)$$

Where, K is a constant.

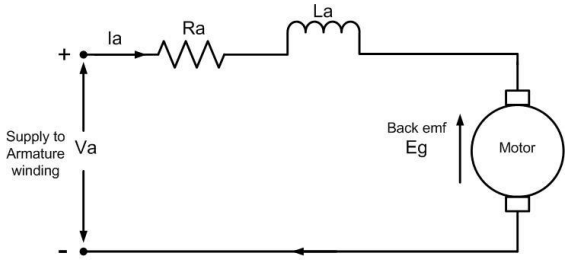


Fig. 1. PMDC motor equivalent circuit

Since the aim is to keep the speed of the motor constant independent of the applied mechanical load, therefore the output of the drive system is speed (ω) and input is reference armature voltage (V_a). Figure.2 shows the block diagram of the PMDC motor, using (1) – (5).

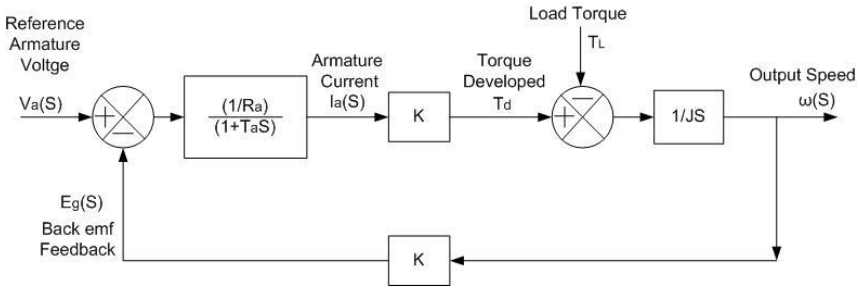


Fig. 2. PMDC motor block diagram [12]

Now combining (1) – (5), transfer function of the PMDC motor can be derived as:

$$\frac{\omega(S)}{Va(S)} = \frac{(\frac{K}{Ra})/\{JS(1+TaS)\}}{1+(\frac{K^2}{Ra})/JS(1+TaS)} \quad (6)$$

Where, armature time constant, $T_a = \frac{L_a}{R_a}$.

Electromechanical time constant of the motor is given by:

$$T_m \quad (7)$$

At the time of starting of the motor, assuming load torque T_L = 0 Nm and also considering negligible armature inductance L_a, the derived transfer function is

$$\frac{\omega(S)}{Va(S)} = \frac{(1/K)}{(1+STm+S^2TaTm)} \quad (8)$$

As $T_a << T_m$ [7], therefore $(1 + ST_m + S^2T_aT_m)$ can be written as $(1 + ST_m)(1 + ST_a)$. Hence,

$$\frac{\omega(S)}{Va(S)} = \frac{(1/K)}{(1+STm)(1+STa)} \quad (9)$$

The final transfer function given by (9) will be used to model the PMDC motor in MATLAB.

Controller circuit modelling

As the speed of the motor is to be maintained constant at the reference value even under variable mechanical load, proper design of the controller circuit is necessary. The common controller circuits are proportional (P), proportional-integral (PI) and proportional-integral-derivative (PID) control circuits. Among these, PI controller is used as P controller has the capability of fast response and I part has the ability to make the steady state speed error to zero [11, 13].

In Figure. 3, block diagram of a basic PI controller is shown. The controller is placed before the PMDC motor model as shown in Figure. 4.

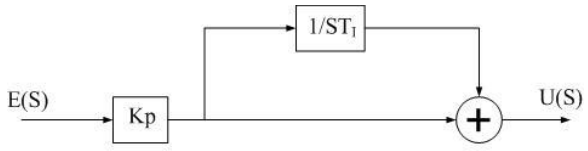


Fig. 3. PI controller

From Figure. 3, the relation between E(S) and U(S) is as

$$\frac{U(S)}{E(S)} = K_P + \frac{K_I}{S} \qquad (10)$$

Where, K_P and K_I are proportional and integral gains respectively and also, $K_I = \frac{K_P}{T_I}$. Here, T_I is integral time.

3. MATLAB SIMULATION OF PMDC MOTOR WITH PI CONTROLLER

Before implementation of the speed control circuit of motor in hardware, MATLAB simulation is done for cross verification of the system. To drive the motor from a PWM signal [14], a chopper module has been inserted between the PI controller block and the PMDC motor block as shown in Figure. 4.

Parameters of the PMDC motor used for simulations are given below:

Voltage rating (V_a): 12volt

Armature resistance (R_a): 0.5Ω

Armature inductance (L_a): 0.01H

Rated armature current (I_a): 10A

Rated speed (N_r): 3000rpm = 314.1593rad/s

Back emf constant (K): 0.0954volt-s/rad

Moment of inertia (J): 0.4Kg-m^2

Armature time constant ($T_a = L_a/R_a$): 0.02s

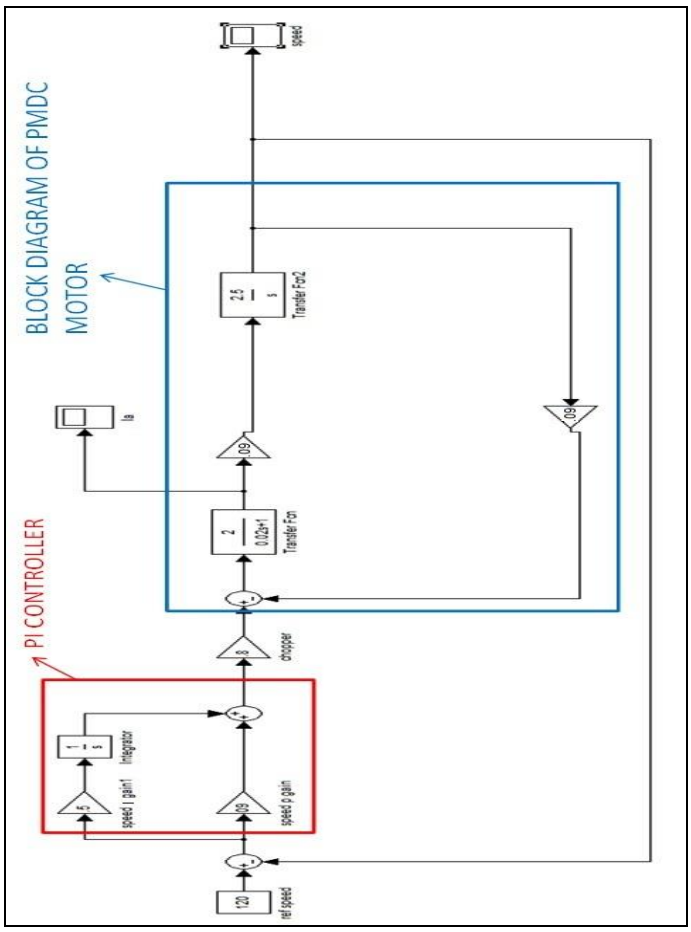


Fig. 4. PMDC motor block diagram with PI controller under no load codition

Simulated speed-time curve of the PMDC motor with PI controller is shown in Figure. 5.The reference speed of the motor is set to 120 rad/s in the MATLAB simulink model shown inFigure.4 shows that as anticipated, speed of the motor is found to remain constant at the set value of 120 rad/s under no-load.

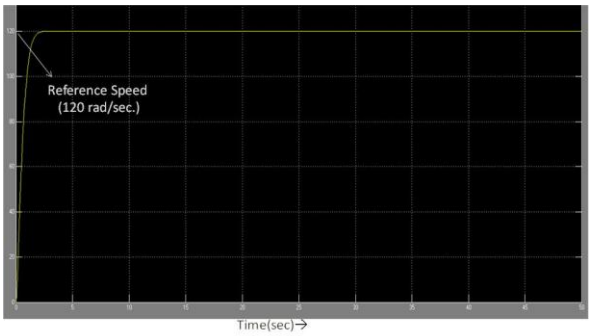


Fig. 5. Speed-time curve of PMDC motor with PI controller

4. HARDWARE IMPLEMENTATION

After verification of the simulation model, the challenge is to run a PMDC motor at a set value of speed under variable load conditions in hardware setup. The controller circuit remains in MATLAB Simulink. Interfacing between the hardware and software model is done by Arduino microcontroller with L293D, a motor driver chip.

PMDC motor

Two identical PMDC machines are used in the study as shown in Figure. 6.

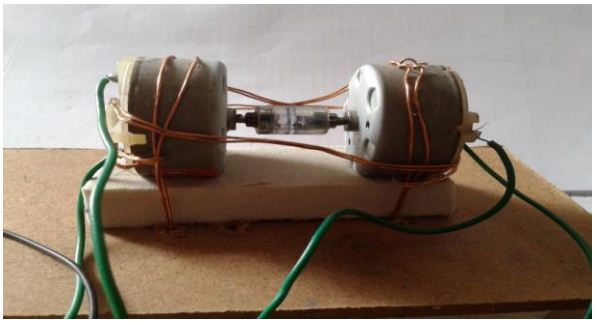


Fig. 6. PMDC motor and generator at coupled condition

One is used as a motor for which the speed is to be maintained at a fixed value at different mechanical loads. The second one is used as a tacho generator to sense the speed of motor and produce a voltage signal at its output terminal that is proportional to the speed. The shafts of the motor and generator being directly coupled, tacho generator output voltage can directly be used as a feedback signal proportional to the motor speed.

Arduino Board

The Arduino microcontroller board as shown in Figure. 7 is used as a bridge between the PMDC motor and MATLAB Simulink model of the drive setup.

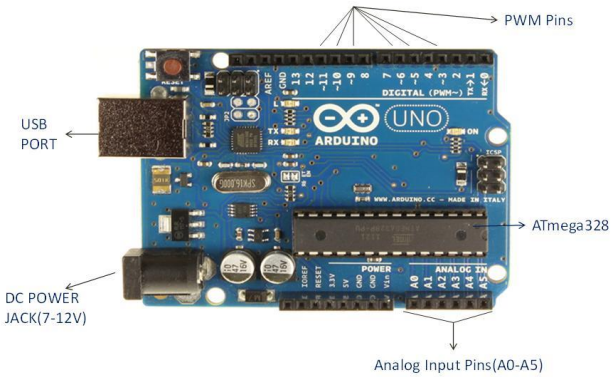


Fig. 7. Arduino UNO board [15]

The Arduino board is used to transmit the controller output signal to the motor via motor driver chip L293D. A detailed specification of the Arduino board is given in Table I.

TABLE I
DETAIL SPECIFICATIONS OF ARDUINO UNO [16]

Components & Parameters	Types & Values
Microcontroller	ATmega328
Operating	5V
Input Voltage	7-12V
Input Voltage	6-20V
Digital I/O Pins	14 (of which 6 provide
Analog Input	6
DC Current per	40 mA
DC Current for	50 mA
Flash Memory	32 KB (ATmega328) of
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

L293D – a motor driver chip[17,18]

The motor driver chip (L293D) is connected between Arduino board and the PMDC motor.As

the Arduino output signal strength is less than the motor voltage and current ratings, therefore Arduino output cannot be used to directly drive the motor. The driver chip amplify the controller signal received from Arduino to match the required voltage, current rating for the motor.The L293D driver chip can be used to drive two motors simultaneously, though it has been used to drive only one motor is the present work. In Figure.8, the terminals of L293D necessary to drive a single motor are marked.

Hardware connection of motor-generator with arduino and L293D

Schematic diagram of the entire hardware set up with the Arduino board, motor driver chip, motor, and generator is shown in Figure.8.

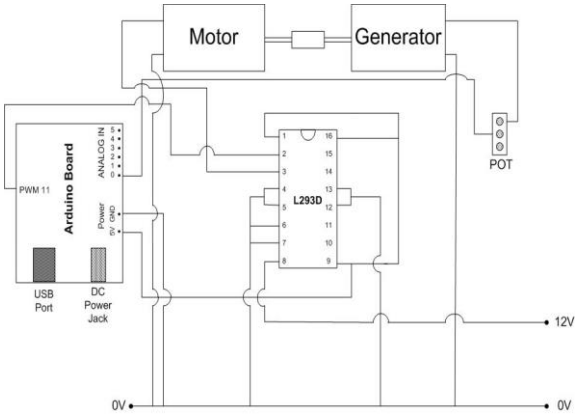


Fig. 8. Layout of hardware connection

The PWM signal is transmitted from Arduino pin 11 to L293D pin 2. From the chip pin 3 the PWM signal goes to the motor. All the 5V terminals are shorted and 0V terminals are shorted as shown. To drive the motor through L293D chip, a 12V supply is given to the chip pin 8. From the output of the generator the voltage is dropped by using a pot, because the Arduino cannot deal with a voltage having high value. Then Pot transmits the generator output voltage to arduino analog pin 0. Via the USB port, the arduino connected to MATLAB Simulink model at a computer and an AC to DC adapter is connected at DC power jack of arduino.

Photograph of the actual set up is shown in Figure.9.

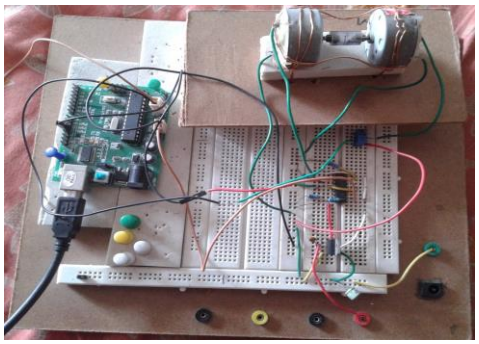


Fig. 9. Hardware connection

MATLAB Simulink model of PI controller and feedback

Fig.10 shows the MATLAB simulation part of the drive system. Here, the reference speed is kept at 150rad/s. Then the speed signal enters to the PI controller block, with proper gain. Output of PI controller then enters to rounding function block, where the fractional data is converted to integer data with round up, which is accepted by the Arduino. Next block is Arduino 1 Analog Write Pin 11, where the analog signal is converted into PWM signal at pin 11of Arduino. The signal comes from Arduino pin 11 then enters to the L293D input pin 2. In feedback path, Arduino 1 Analog Read Pin 0 receives the output of tacho generator and the signal goes to the input to compare with the reference speed and to generate error speed signal if any. After the analog read block, a gain block is used. The value of the gain is 0.249. Arduino can read up to 1024 and can write up to 255. Therefore, (255/1024) = 0.249. Now, this block converts the value that is read by the Arduino in the 0 - 255 range, and then through the feedback path it goes to the comparator for being compared with the

reference value to generate error signal which finally goes to the PI Controller [19]. The Arduino IO setup block shows the type of the Arduino board, port number (COM 8) at which the board is connected.

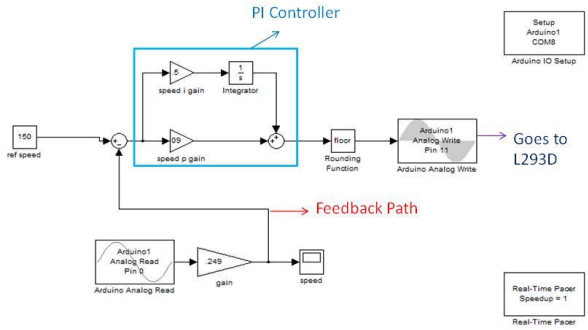


Fig. 10. Simulink model of controller

RESULT AND ANALYSIS

Fig. 11 shows the speed – time curve of the motor.

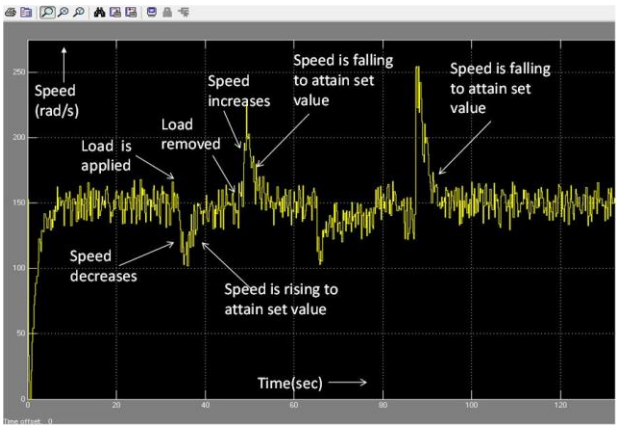


Fig. 11. Speed-time curve of PMDC motor with PI controller under variable load with Arduino interface

As seen in the figure, at around 33s, mechanical load is suddenly applied to the shaft of the motor and instantaneously there is a fall of speed, but the motor is found to regain the reference speed of 150rad/s within few seconds. At 47s the external load is removed and there is an instantaneous increase in speed, which again comes back to its reference speed. This process motor responding to sudden load changes are made to happen several times during the run as shown in Figure. 11.

In Figure. 11 the speed –time curve is not smooth like thesimulatedplot in Figure. 5, which is totally designed in MATLAB Simulink. This is because of the fact that in software the all parameters are ideal, but in hardware there are some unavoidable parameters and disturbances that get introduced in the system.

5. CONCLUSION AND FUTURE SCOPE

This article presents a new scheme for closed loop speed control of PMDC motors under variable load conditions. Novelty of the work is in designing a MATLAB Simulink model of the controller that avoids otherwise complicated hardware circuitry. The motor is interfaced with the software based PI controller through an Arduino UNO microcontroller and L293D motor driver IC.

Hardware experimental results presented in the article demonstrates the capability of the controller in maintaining the speed of the motor at a set reference value irrespective of different applied mechanical loads.

In place of the Arduino UNO board, if higher versions of Arduino are used along with MOSFET or other FET based motor drivers rather than using L293D, then higher capacity motors can be dealt with. Also, using the basic concepts of the proposed technique, other types of motors can be controlled.

REFERENCES

[1] J. Larminie and J. Lowry, Electric Vehicle Technology Explained, 2nded.Hoboken, NJ, USA: Wiley, 2012.

[2] C. Mi, A. Masrur, and D. Gao, Hybrid Electric Vehicles: Principles andApplications with Practical Perspectives. Hoboken, NJ, USA: Wiley,2011.

[3] H. R. Jayetileke, W. R. de Mei, H. U. W. Ratnayake, "Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due", ISSN: 2151-1802

[4] Julio Noel Hernández-Pérez, Jesús Ebert Giral-Salas, Roberto Morales-Caporal, Rafael Ordoñez-Flores, Miguel Ángel Morales-Flores, "Speed and current control of a permanent-magnet DC servo motor using a real-time microcontroller", ISBN:978-1-4799-1460-9

[5] Y. S. E. Ali, S. B. M. Noor, S. M. Bashi, M. K. Hassan, "Microcontroller performance for DC motor speed control system", ISBN: 0-7803-8208-0

[6] AhmetIsik,OktayKarakaya, P.AlperÖner, Mehmet KubilayEker, "PMDC motor speed control with fuzzy logic algorithm using PIC16F877 micro controller and plotting data on monitor", ISBN**:** 978-1-4244-3429-9

[7] R. Sankar, S. Ramareddy, "Closed loop controlled PMDC motor drive" ISSN **:** 0537-9989

[8] National Instruments- Robotics Fundamentals Series: Motors, Publish Date: Mar 16, 2016

[9] https://www.arduino.cc

[10] http://www.arduino.cc/documents/datasheets/L293D.pdf

[11] Ned Mohan,Tore M. Undeland, William P. Robbins, Power Electronics: Converters, Applications, and Design, 3rded, India, Wiley, 2009, pp. 377 – 391.

[12] Gopakumar, K., Power Electronics and Electrical Drives, Video Lectures 1-25, Centre for Electronics and Technology, Indian Institute of Science, Bangalore.

[13] Choudhury, d. Roy, Modern Control Engineering, India, PHI, 2012, pp. 323 – 346..

[14] Cengelci, S. U. Sulistijo, B. O. Woo, R. Teodrescu, and F.Blaabjerg, "A new medium-voltage PWM inverter topology for adjustable-speed drives," IEEE Trans. Ind. Appl. , vol. 35, no. 3, pp. 628–637, May/Jun. 1999.

[15] https://www.arduino.cc/en/Guide/Windows

[16] https://www.arduino.cc/en/main/arduinoBoardUno

[17] https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF

[18] http://www.ti.com/lit/ds/symlink/l293.pdf

[19] https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM