

Programming a Computer and a Microcontroller to Control the Speed and Direction of DC Motors

¹Ademola Abdulkareem, ²Dike U. Ike, ³Adewale A. Ajao, ⁴Adelakun A. A

^{1,2,3,4}Department of Electrical and Information Engineering, Covenant University, Nigeria.

ABSTRACT: Direct current (DC) motor has already become an important machine used in many applications across a wide range of powers and speeds. The ease of control and excellent performance of the DC motors will ensure that their applications will continue to grow in the future. This work is mainly concerned with DC motor speed, direction and active state (either start or stop) control by using microcontroller PIC18F452 via Pulse Width Modulation (PWM) technique, where its signal is generated in microcontroller. The PWM signal will request the motor driver to vary the voltage supply across the motor to maintain its speed. A program written in Visual Basic 6.0 is developed to provide a graphic user interface (GUI) to enable the control of the DC motor from a computer.

Keywords: DC motor, GUI, Microcontroller, Pulse Width Modulation, Visual Basic.

1. INTRODUCTION

Direct current (DC) motors have variable characteristics and are used extensively in variable-speed drives. An important advantage of DC motors is that their speed and direction could be controlled more readily than those of AC motors [1].

There are several types of applications where the load on the DC motor varies over a speed range. These applications may demand high-speed control accuracy and good dynamic responses i.e. in home appliances, washers, dryers and compressors are good examples. In automotive, fuel pump control, electronic steering control, engine control and electric vehicle control are good examples of these. In aerospace, there are a number of applications, like centrifuges, pumps, robotic arm controls, gyroscope controls and so on.

Motion control plays an important role in automation; manufacturing plants in industries like chemical, pharmaceutical, plastic and textile all require motion control, and it may be a flat belt application, flow-control application or mixing of substances. Different types of motors such as AC, DC, servo or stepper are used depending upon the application. Of these, DC motors are widely used because controlling a DC motor is somewhat easier than other kinds of motors. DC motor driver changes the speed and direction of motion of the motor.

The aim of this work is to program a microcontroller to control the direction and speed of a dc motor. This system

will be able to control the DC motor speed (at desired speed regardless the changes of load) and direction through a computer.

2. MATERIALS AND METHODS

The two stages of this work are the hardware development and software development.

The first stage i.e. the hardware development, there are three main sections and they are to design the;

- i. DC power supply circuit.
- ii. communication bridge circuit, incorporating the serial communication standard RS232.
- iii. motor controller circuit, this includes the circuit for the PIC16F877A microcontroller.

The second stage i.e. the software development, there are two main sections, which are to;

- i. develop embedded system software using the MikroC PRO for PIC16F877A microcontroller.
- ii. develop a Windows based application software to control the DC motor from the PC.

- iii. simulate the control system using Proteus software.

The block diagram of the system is shown in Fig. 1 below.

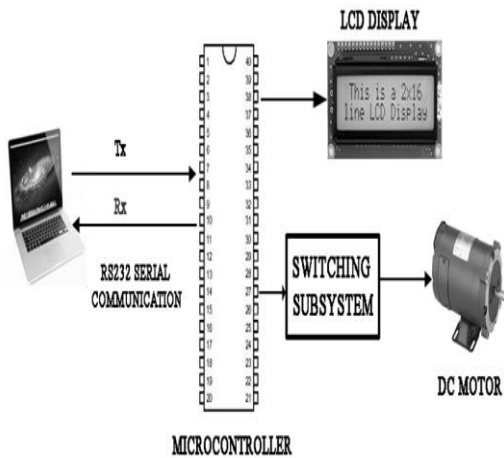


Figure 1: Block Diagram of the System

2.1 Hardware Implementation

The components that was used for the design and development of this system include: DC motor, power supply (5 V, 12 V), microcontroller (PIC18F452), RS232 serial communication of microcontroller, DC Motor driver (L293D) etc.

2.1.1 Power Supply Subsystem

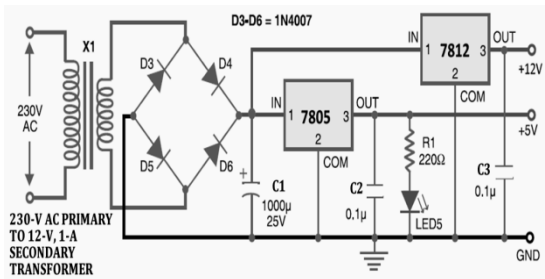


Figure 2: Circuit Diagram for the Power Supply

Here, a 230-V-to-12-V transformer steps down a 230-V, 50-Hz line voltage to a 12-V, 50-Hz secondary voltage. A bridge rectifier (network of four diodes) then rectifies the secondary voltage into a pulsed DC voltage with a peak of 10.6 V and a frequency of 100 Hz (1.4 V is lost during

rectification due to diode biasing voltages, and it appears that the output is double in frequency due to the negative swings being converted into positive swings by the rectifier). The average DC voltage at the output is equal to 0.636 of the peak rectified voltage.

A filter capacitor is then used to delay the discharge time and hence “smoothen” out the pulses. The capacitor must be large enough to store a sufficient amount of energy to provide a steady supply of current to the load. If the capacitor is not large enough or is not being charged fast enough, the voltage will drop as the load demands more current. A general rule for choosing the capacitance of a capacitor C is to use the following relation:

$$R_{load} \gg \frac{1}{f} \dots\dots\dots 1$$

$$V_{ripple} = dV = V_{peak} - V_{minimum} \dots\dots\dots 2$$

$$V_{peak} = \sqrt{2}V_{rms} \dots\dots\dots 3$$

$$\text{From } I_{load} = C \frac{dV}{dt} \dots\dots\dots 4$$

$$\text{Where } dt = \frac{1}{frequency} \dots\dots\dots 5$$

The frequency from the AC mains supply is 50-Hz; the full wave rectified voltage frequency is twice the mains voltage, which is 100-Hz.

From equation 5:

$$dt = \frac{1}{100} = 0.01 \text{ seconds}$$

From equation 2:

$$V_{peak} = \sqrt{2} * 10.6 = 15 - V$$

The total load current I_{load} is less than 0.7-mA, we use $I_{load} = 0.5\text{-mA}$

$$dV = 15 - 9 = 6 -V$$

$$C_1 = I_{load} \frac{dt}{dV} = 0.5 * \frac{0.01}{6} = 8.33 \text{ mF} = 833\text{-}\mu\text{F}$$

1000- μF , 25-V capacitor is used as C_1 in the project, because it is close to 833- μF and its maximum charge/discharge voltage is 25-V. C_2 and C_3 are used to eliminate voltage spikes at the output; it should be around 0.1 μF or larger. The appropriate current suitable for the LED used is 20-mA, the voltage drop across the LED is 2-V, therefore the resistance used is 150 - Ω as calculated below. 200 - Ω is close to the calculated value. $R_1 =$

$$\frac{V_{source} - V_{drop}}{0.020} = \frac{5 - 2}{0.02} = 150 - \Omega$$

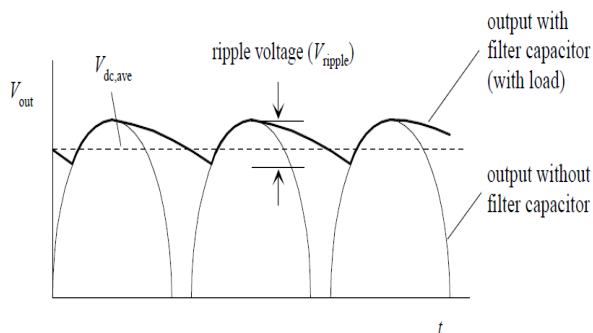


Figure 3: Smoothened full-wave rectified DC voltage

Where f is the rectified signal's frequency (50 Hz). The ripple voltage (deviation from DC) is approximated by:

$$V_{ripple} = \frac{I_{load}}{fC}$$

..... 6

2.2 Switching Subsystem

The switching subsystem consists of;

I. Bipolar Junction Transistors (BJT)

N-channel BJTs are used as the solid state switches in this project. When input voltage is across into the base of the BJT, the required base voltage V_B required to make current to flow through the emitter and collector (I_{CE}) is about 0.7 V. In order to achieve this, we connect a resistor R in series with the base node of the BJT. The desired current ($I_{desired}$) is about 43-mA.

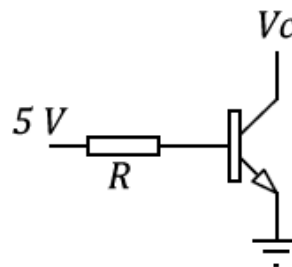


Figure 4: Configuration of BJT

$$R = \frac{V_{source} - 0.7}{I_{desired}} = \frac{5 - 0.7}{43\text{mA}} = 100\text{-}\Omega$$

The base current required in making the TIP4 transistor used throughout the project to act as a switch is about 43mA as calculated. When sufficient base current is applied, the TIP41 transistor moves from the cut-off region to the saturation region.

II. Diodes

Diodes are made use of in this work for;

- rectification (making use of rectification diodes)
- indication (making use of light emitting diodes, LEDs)
- protection

A diode placed in reverse bias across a relay's coil eliminates voltage spikes by going into conduction before a large voltage can form across the coil. The diode must be rated to handle currents equivalent to the maximum current that would have been flowing through the coil before the current supply was interrupted. A good general-purpose diode used in this system that works well for such applications is the 1N4004 diode. [2]

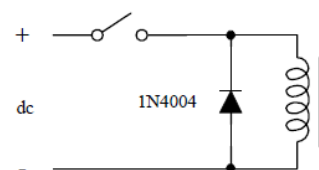


Figure 5: Diode Protection of a DC-Drive Coil

III. SPDT and DPDT Relays

One SPDT (Single Pole Double Throw) relay and One DPDT (Double Pole Double Throw) relay are used in the project. The SPDT is used to start and stop the DC motor, while the DPDT relay is used for changing the direction of the DC motor. They are energized by 5-V input from the microcontroller. Microcontroller pins A0 and A1 are used to start/stop the motor and change the

direction of the motor respectively. Each output of the port is fed into a Bipolar Junction Transistor (TIP41) which in response to the voltage applied to its base switches a relay (SPDT or DPDT).

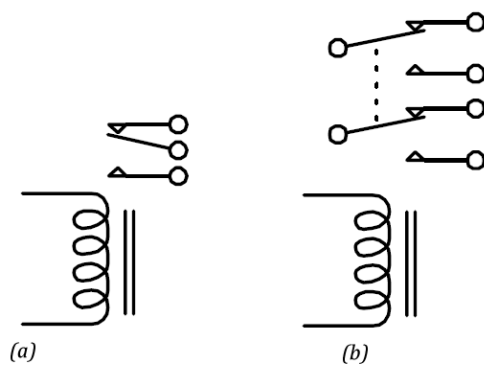


Figure 6: (a) An SPDT Relay Circuit Schematic
(b) A DPDT Relay Circuit

IV. DC Motor Driver (H-Bridge)

The L293D is a quadruple high-current half-H driver. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. It is designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

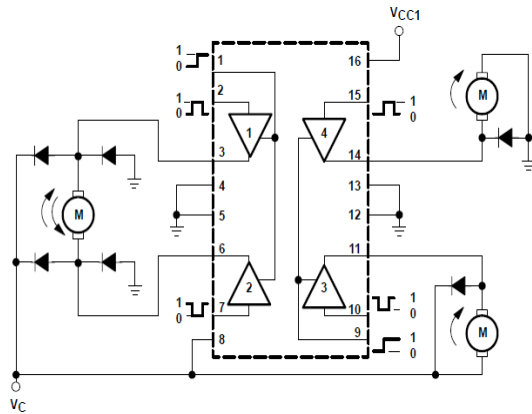


Figure 7: Pin Composition of L293D, DC Motor Driver

2.3 Microcontroller PIC18F452

Microchip Inc. developed the *PIC18F* series of microcontrollers for use in high-pin count, high-density, and complex applications. The *PIC18F* microcontrollers offer cost-efficient solutions for general purpose applications written in C that use a Real-Time Operating System (RTOS) and require a complex communication protocol stack such as TCP/IP, CAN, USB, or ZigBee. PIC18F devices provide flash program memory in sizes from 8 to 128Kbytes and

data memory from 256 to 4Kbytes, operating at a range of 2.0 to 5.0 volts, at speeds from DC to 40MHz [3], [4].

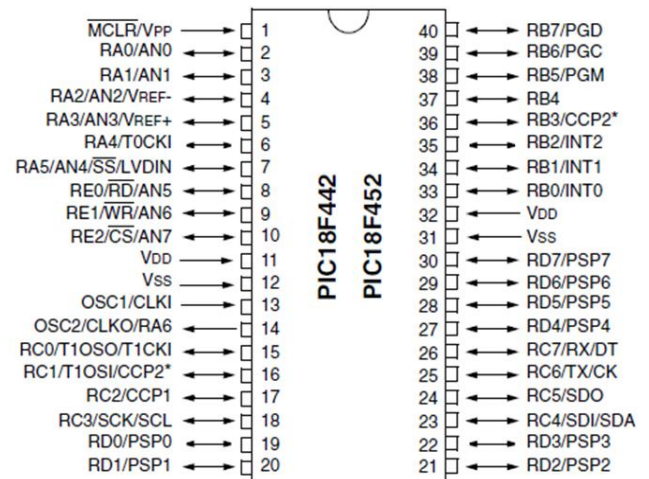


Figure 8: PIC18F452 Microcontroller Pin Configuration

2.4 LCD Display

A LCD display is specifically manufactured for the use with microcontrollers, which means that it cannot be activated by standard IC circuits. It is used to display different messages on a miniature liquid crystal display. The LCD display described here is based on the HD44780 controller (Hitachi) and displays messages in two lines with 16 characters each. Different symbols such as letters of alphabet, Greek letters, punctuation marks, mathematical symbols etc. can be displayed on it. It is also possible to display symbols created by the user. Other useful features include automatic message shift (left and right), cursor appearance, LED backlight etc [5].



Figure 9: A 16 by 2 LCD module

2.5 RS232 Serial Communication

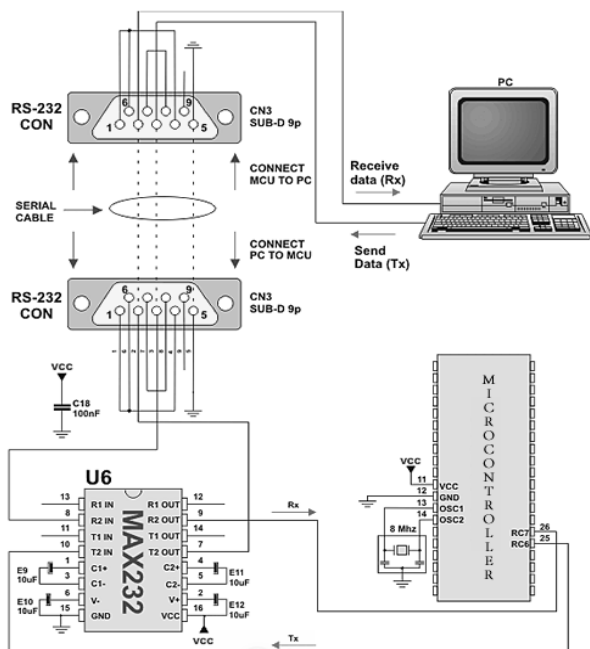


Figure 10: RS232 Serial Communication

The voltage levels specified by the RS232 protocol are $\pm 12V$. A logic HIGH signal is at $-12V$ and a logic LOW signal is at $+12V$. PIC microcontrollers, on the other hand, normally operate in the range of $0V$ to $5V$ voltage levels, the RS232 signals must be converted within the range of $0V$ to $5V$ input to a microcontroller.

Similarly, the output of the microcontroller must be converted to $\pm 12V$ before sending it to the receiving RS232 device. The voltage conversion is carried out with RS232 converter chip i.e. MAX232, manufactured by Maxim Inc. The MAX232 chip receives the TX signal from the input pin of the microcontroller and converts it to RS232 levels. Comparably, the serial data received by the MAX232 chip is converted into microcontroller voltage levels and then sent to output pin. Note that correct operation of the MAX232 chip requires four capacitors to be connected to the chip.

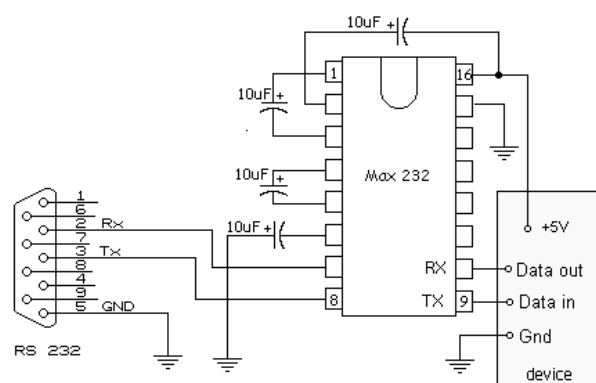


Figure 11: Pin layout of MAX232 connected to an RS232 connector (left) and microcontroller (right)

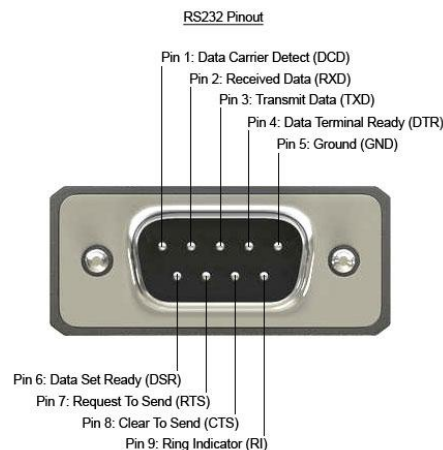


Figure 12: An RS232 Connector (male) and its Pin Layout

2.6 Software Implementation

For the software implementation, mikroC for PIC (version 5. 6) is used to program microcontroller in high level language called “C”. Also, Visual Basic 6.0 is used to program the personal computer to provide the user interface for the purpose for DC motor speed and direction control.

Microcontroller acts as the brain of the whole DC motor direction and speed control system. It will receive the desired input (start, stop, direction and desired speed) from user through PC that interfaces with the microcontroller. These inputs are ASCII (American Standard Code for Information Interchange) characters e.g. “a”, “b” etc. They are easily sent via RS232 and easily received by the microcontroller.

An algorithm has to be developed to make the microcontroller to read the input and respond accordingly. Therefore, the algorithm is established and represented by a flowchart in Fig. 13b. This flowchart is then translated into C programming language and compiled to hexadecimal format using MIKROC for PIC software development tool.

Visual basic 6.0 provides the Graphical User Interface (GUI) on the personal computer running on Microsoft Windows Operating System. It sends to and receives ASCII characters (alphabets) from the microcontroller. These characters represents a particular DC motor control function e.g. “a” is for start, “b” is for start, “c” is for increase speed, “d” is for decrease speed , “e” is for clockwise rotation and “P” is for anticlockwise rotation.

An algorithm has to be developed to make the DC Motor Control application software (developed using Visual basic 6.0) send ASCII input to the microcontroller and receive back the sent input (a proof that the microcontroller has received it inputs from the PC). Therefore, the algorithm is established and represented by a flowchart in Fig. 13. This flowchart is then translated into Visual Basic programming language and

compiled to a Standard EXE (executable) format using Visual basic 6.0 [4], [5], [6], [7].

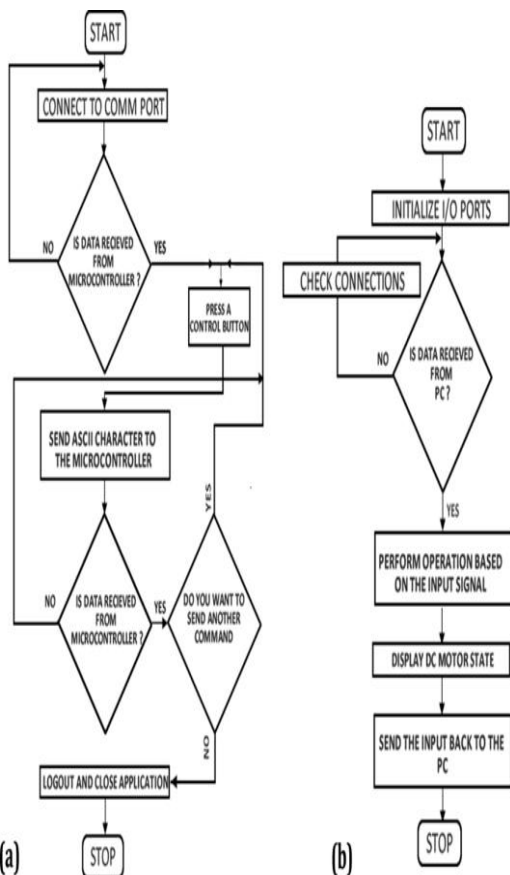


Figure 13: (a) Flowchart for DC motor Controller GUI Application (b) Flowchart for the PIC18F452 Microcontroller Operation

3. RESULTS

The DC Motor Controller software went through series of testing , to find out if;

- i. there is communication between the application and the Microsoft Windows® created virtual COMM PORT
- ii. all the DC motor control buttons are receiving the byte they sent to the microcontroller (sent after clicking)

Proteus ISIS 7 Professional was used during simulation to simulate the hardware module of the system . Virtual Serial Port Driver software was used to bridge the simulated hardware module on ISIS 7 Professional.

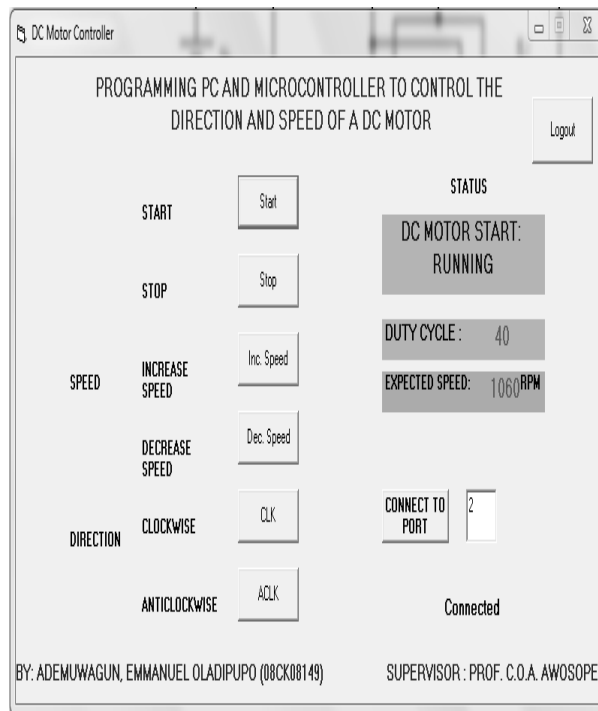


Figure 14: Real time Picture of the DC Motor Controller Software in Operation

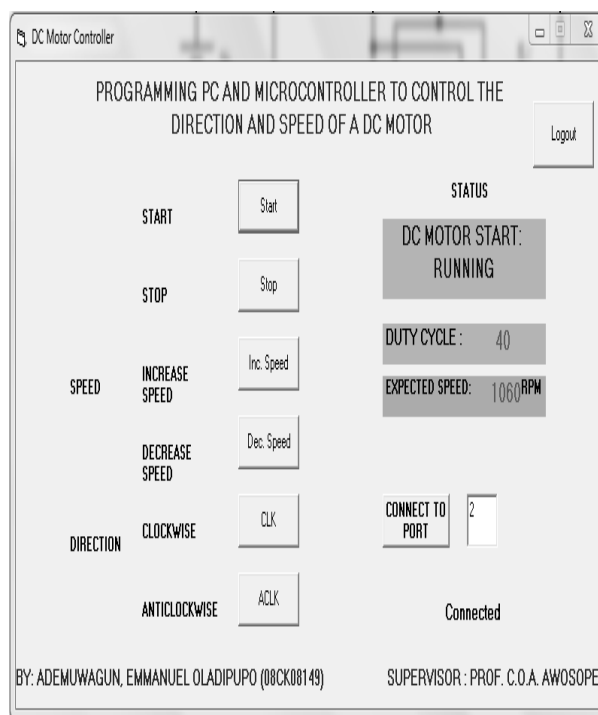
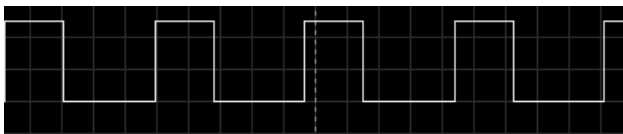


Figure 15: DC Motor Controller Testing

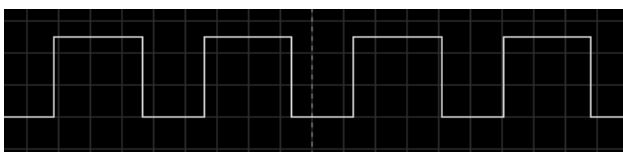


Figure 16: Virtual Serial Port Driver Software User Interface

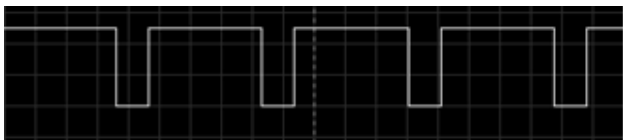
The basic test conducted is to program a LCD module to display the various states of the DC motor. This method is good but not sufficient enough, as there could be wrong circuit connection in the switching subsystem. Therefore, an oscilloscope was connected to the conductor connected to the DC motor from the L293D pins 3 and 6. Fig. 17 shows various output waveforms of the DC motor when operating at different speeds (Duty Cycle controlled).



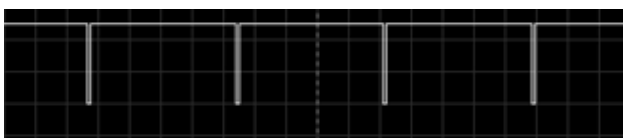
(a) Output voltage for 40% Duty Cycle



(b) Output voltage for 60% Duty Cycle



(c) Output voltage for 80% Duty Cycle



(d) Output voltage for 100% Duty Cycle

Figure 17: (a), (b), (c), and (d) Shows an Oscilloscope Displayed Output Waveform of the Voltage Powering the DC Motor

4. CONCLUSION

Applications of high performance DC motor drives in area such as rolling mills, chemical process, electric trains, robotic manipulators and the home electric appliances require direction and speed control to perform tasks. DC motors have direction and a speed control capability, which means that speed and direction of rotation can be changed at anytime to meet new condition.

The aim of this work which is to program a personal computer and a microcontroller to control the direction and speed of a DC motor has been achieved. By reversing the polarity of DC motor contacts, the direction of a DC motor can be reversed. Also varying the PWM signal from microcontroller to the motor driver, motor speed can be controlled.

In conclusion, with the aid of a GUI (Graphic User Interface) based application software on a PC (Personal Computer), interfacing with the PC with a programmed microcontroller-based DC motor controller circuit using RS232 serial communication standard together as a communication medium between the two, the direction and speed of a DC motor can effectively be controlled and monitored in real time.

REFERENCES

- [1] John Bird, Electrical and Electronic Principles and Technology, 4th ed., Kidlington: Newnes, 2010.
- [2] Paul Scherz, Practical electronics for inventors, New York: McGraw-Hill, 2000.
- [3] "World of Microcontrollers". Mikroelektronika: 2 April 2013. Retrieved from <http://www.mikroe.com/chapters/view/74/pic-basic-book-chapter-1-world-of-microcontrollers>.
- [4] Dogan Ibrahim, Advanced Microcontroller Projects In C, Kidlington: Newnes, 2010.
- [5] "PICMicrocontrollers Programming in BASIC". Mikroelektronika: 12 April 2013. Retrieved from <http://www.mikroe.com/chapters/view/74/pic-basic-book-chapter-1-world-of-microcontrollers>.

[6] S. J. Chapman, *Electric Machinery Fundamentals*, New York: McGraw-Hill, 2002.

[7] Rohit, G., Ruchika, L. & Subhransu, P., (2012). Thyristor Based Speed Control Techniques of DC Motor: A Comparative Analysis. *International Journal of Scientific and Research Publications*. Retrieved March 27, 2013, from http://www.ijsrp.org/research_paper_jun2012/ijsrp-June-2012-24.pdf