IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 10, Issue 2 Ver. III (Mar – Apr. 2015), PP 79-85
www.iosrjournals.org

Programming the PIC16F877A Microcontroller for Speed Control of a DC Motor

Patrick Olabisi, Israel Megbowon and Isaiah Akintunde
Electrical & Computer Engineering Dept., Bells University of Technology, Ota, Nigeria

Abstract: *Considering the versatility of usefulness and ease of control of the DC motor both for domestic, industrial and entertainment, research works have continued unabated on further and better ways to achieve its control. Among the many methods used in the control of the speed of a DC motor the microcontroller – based method was adapted in a closed-loop real time control system approach making use of the PIC16F877A with a program in Flow Code developed to implement a feasible control system. The PIC16F877A microcontroller was programmed and used for the control of the speed of DC motors at the desired speed range of up to 500 – 600 rpm with variation of load. An LCD was used to display the feedback motor speed signal fed back to the controller.*
Keywords: *DC Motor, Speed Control, Microcontroller, Flow Code Programming, Control Systems.*

I. Introduction

Direct current (DC) motors are very important integral parts of drive mechanisms both for domestic, entertainment, and industrial uses. The wide applications of dc motors are largely dependent on the variability of their capacities of power and running speeds. The possibility of variation and the ease of control of the drive speed and direction of dc motors have helped to increase the range of areas in which they are being applied, which include such areas as rolling mills, electric vehicle tractions, electric trains, electric bicycles, guided vehicles, home electric appliances, toys and robotic manipulators, etc. [1-7].

Characteristics of the DC motor that are required to be controlled or changed at anytime usually include speed, torque and direction of rotation. Important speed–related requirements for efficient and intelligent dc motor speed control include maintaining constant speed with variation of load, varying the running speed to suit particular needs, for energy saving, soft starting and stopping to increase motor reliability and availability. Normally, in the conventional DC motors which is known to have high starting torque and a high no-load speed, the speed is poorly regulated in the sense that the speed changes when the amount of load it drives changes and would require additional supply potentials to maintain a steady running speed. But in the permanent magnet DC motors, the torque-speed characteristic is usually linear because the flux is generated by the permanent magnet rather than the wire wound poles in conventional dc motors.

II. Background Knowledge

Controlling the speed of dc motors may not necessarily require measuring the running speed as in feedback or closed loop speed controller. The feedback speed controller is usually better than the open loop speed controller, but is much more complex and it is not required in very simple designs or applications. Speed control in DC motor drive systems are commonly operated with the closed-loop control[8], whereby the feedback speed signal is generated through tachometers or optical encoders coupled to the motor shaft[9]. The drive speed is captured by the sensor and the information is fed back to the controller. An error signal, which is the difference between the process speed variable (PS) from the speed sensor and the set or reference speed variable (RS) is generated from it. Based on the error signal, the controller makes adjustments when and where necessary to keep the motor speed at the reference (required) value. This self-correcting feature of the closed-loop speed control makes it preferable over open-loop speed control in many applications.

Nevertheless, the closed-loop approach making use of speed sensor or shaft transducer have the disadvantages of likely significantly affecting the mechanical robustness of the drive [10], changes in the transducer's characteristics due to mechanical wear can adversely affect the stable performance of the closed loop system, and as an additional component, it increases the cost and size and reduces the reliability of the motor system [8,11]. These have led to considerable works being done in developing speed controllers that do not use explicit speed sensors. Such approach is known as the speed sensorless control.

The fields of artificial intelligence (AI) such as Fuzzy Logic (FL) and Artificial Neural Network (ANN) and computer engineering have, in recent times, brought in radically different and intelligent approach to the control of industrial processes. Artificial intelligent control of dc motors seeks to understudy or learn the experiences of human operators and the basis on which motor speed variation is carried out. From this it is able to estimate values or levels to which the speed should be set with respect to demand/load requirements.

[12] noted that the development and application of faster, smaller and cheaper microprocessors to the control of processes and systems is now common place. In controlling DC motor drive systems, the computer serves as an integral part of the feedback control system to carry out the control action or supervise the control operation in real-time in according to an algorithm stored inside the computer and/or microprocessor memories. The main objective of this work is to efficiently program the PIC16F877A microcontroller using flow code and to interface the feedback control system on an LCD display unit in the design of a DC motor speed controller to operate at desired speed regardless of changes that may occur in the load.

III. Related Works

The use of and advancement in microcontroller technology in the recent times have allowed for dc motors to be controlled more efficiently and at a lower cost than ever before. Atmel AT89C52 microcontroller was used in [13] to generate pulse-width modulated (PWM) signals to control the speed of dc motor. The software was written in 'C' and compiled using Keil C compiler which generates Intel hex code for the microcontroller. [14] made use of the Motorola MC68HC11E9 microcontroller to feed a chopper with high frequency PWM signal in order to be able to control the speed of a dc motor. They found the microcontroller as a reliable electronic device which can be applied for the speed control of all sizes of dc motors with high precision and that the protection and power cut-off system yielded appreciable response.

The PIC16F72 was used in [15] in the control method for DC motor armature voltage control that involves the pulse width modulation (PWM) based on the H-bridge motor circuit. The adaptive fuzzy controller designed and implemented in [1] is an alternative method of using microcontroller to control the speed of a dc motor. The design showed good velocity tracking performance under both load and no-load conditions. The fuzzy logic controller (FLC) was also adopted in [16] for the control of the speed of brushless dc motors. It was noted that that wide range of speed control is possible by means of fuzzy controller and that the controller design does not require explicit knowledge of the motor and load characteristics. Results were obtained without peak overshoot and with negligible steady state error. More and more works are being done to control the speed of dc motors by making use of microcontroller chips to prove the ease and cost-effectiveness of the approach.

IV. Design Methodology

The block diagram of the dc motor control system in this work is shown in Fig. 2. It is a closed-loop control system in which a 40 pin 8 bit CMOS flashmic microcontroller chip (PIC16F877A) shown in Fig. 1 was programmed to control the speed of the dc motor at desired speed in real time. The overall circuit connection is shown in Fig. 3. Two supply lines are used: one for the motor, while the other is for the controller and display circuitry. The speed of the motor is controlled by varying the voltage entering the motor through the potentiometer and the rpm is displayed on a 16x2 LCD display unit.

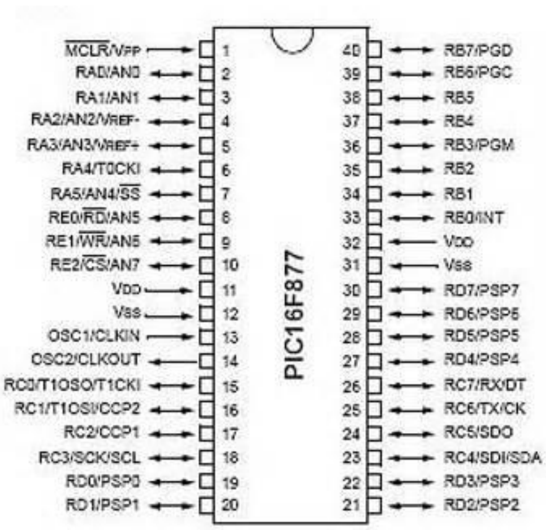


Figure 1: Pin Configuration of the PIC16F877A.

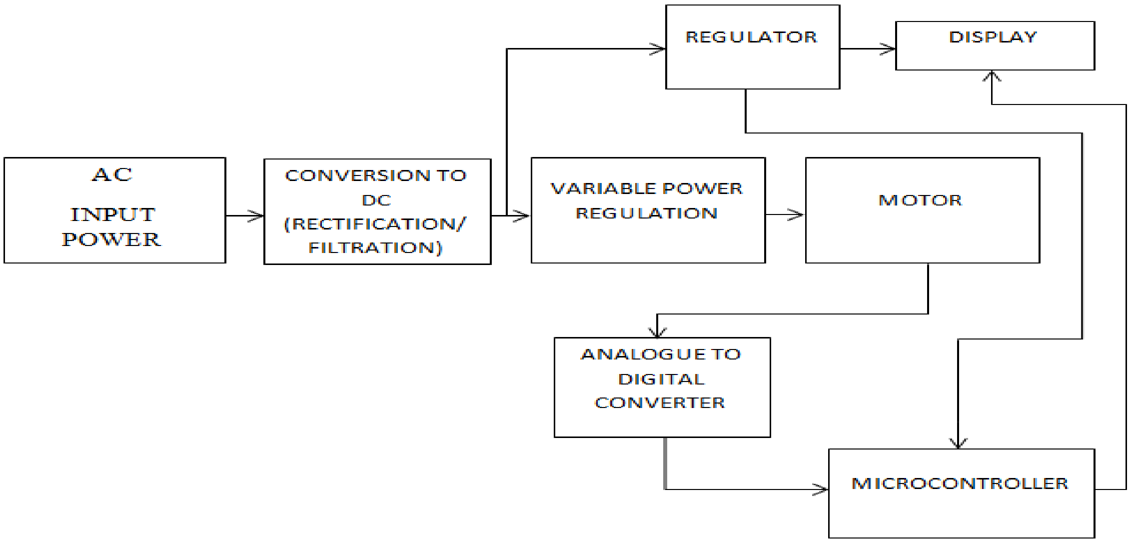


Figure 2: Block diagram of the DC motor speed controller.

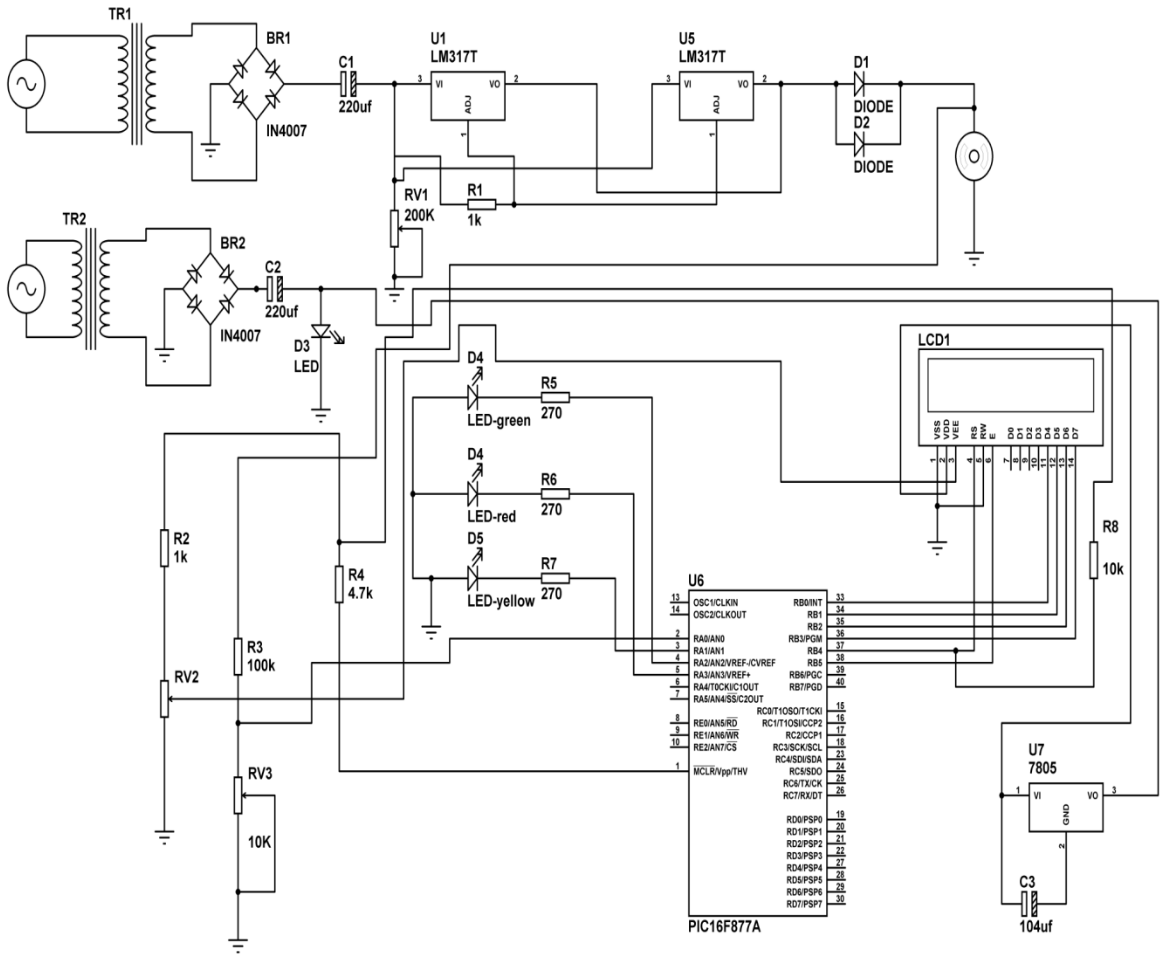


Figure 3: Circuit diagram of the DC motor speed controller

V. Software Implementation

For software implementation, Proteus design suite 8 developed by Labcenter Electronics was used to draw the circuit diagram for the DC motor speed controller, while Flow Code v5.2 was used to program the microcontroller in assembly language.

5.1 Algorithm and Programming in Flow Code

Flow Code is a graphical programming software developed by Matrix Multimedia, which allows those with little or non-programming experience to create complex automated systems without writing traditional codes line by line. Tools located in the program will compile the flowchart and download the hexadecimal program to the microcontroller chip. The microcontroller is the brain of the whole DC motor speed control system. It receive the desired voltage from the potentiometer to determine the speed of the motor.

The actual speed will be compared with the desired speed and the correction will be done by the microcontroller to always maintain the DC motor speed at the desired speed. 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 Appendix A and these flowcharts are then translated into assembly language and compiled.

Shown in Fig. 4 is the display of the voltage supplied to drive the motor and the corresponding speed of the motor. Shown in Fig. 5 is a picture of the completed work displaying the controlled sample motor, the power supply module, and the control/display module.

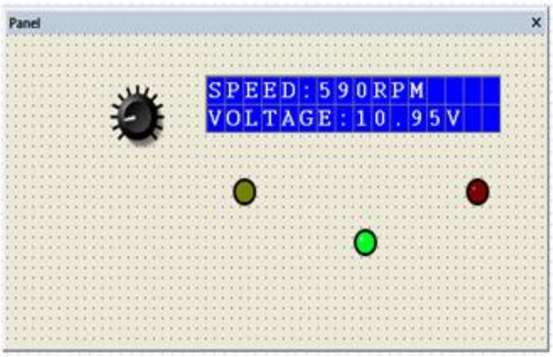


Figure 4: Display of motor speed at 590 rpm.

Figure 5: Picture of the Completed Work.

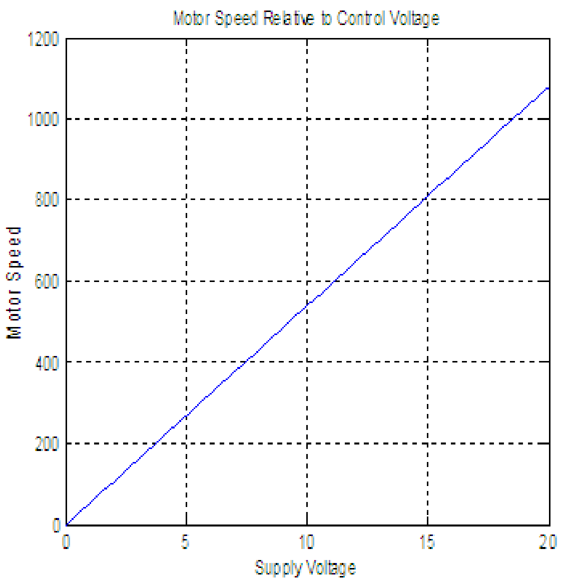
VI. Results Analysis

Table 1 gives the corresponding speed (rpm) of the dc motor as the supply voltage is been varied from 0 volt through to 20 volts at the step of 0.5 volt. Shown in Fig. 6 and Fig. 7 are the plots of the motor speed and the average motor speed against the control voltage as displayed by the digital display of the project shown in Fig. 4 respectively. These plots were carried out making use of MATLAB 2010a. The graphs plot and establish the linear relationship that exit between the control voltage and the motor speed.

Table 1: Supply voltage and motor speed.

Supply Voltage (V)	Speed (rpm)	Average speed (rpm)
0.0	0	0
0.5	26	13
1.0	53	40
1.5	80	67
2.0	107	94
2.5	134	121
3.0	161	148
3.5	188	175
4.0	215	202
4.5	242	229
5.0	269	256
5.5	296	283
6.0	323	310
6.5	350	337
7.0	377	364
7.5	404	391
8.0	431	418
8.5	458	445

9.0	485	472
9.5	512	499
10.0	539	526
10.5	566	553
11.0	593	580
11.5	620	607
12.0	647	634
12.5	674	661
13.0	701	688
13.5	728	715
14.0	755	742
14.5	782	769
15.0	809	796
15.5	836	823
16.0	863	850
16.5	890	877
17.0	917	904
17.5	944	931
18.0	971	958
18.5	998	985
19.0	1025	1012
19.5	1052	1039
20.0	1079	1066



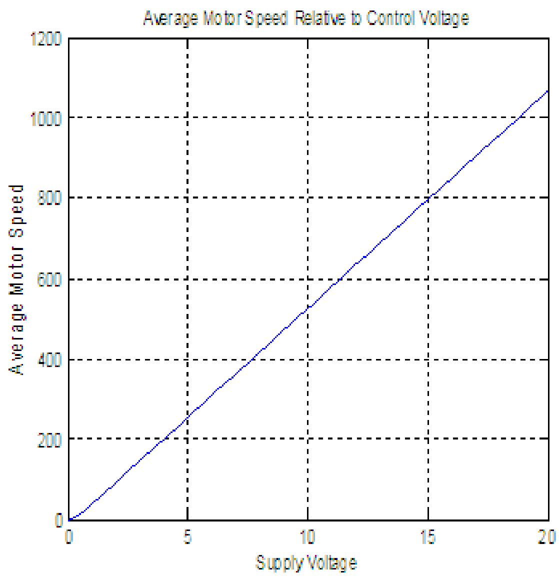


Figure 6: Motor speed against control voltage. **Figure 7:** The average motor speed against control voltage

VII. Conclusions

This work demonstrated the possibility of achieving ease of control of DC motors by making use of the microcontroller. The use of Flow Code programming approach made programming the microcontroller much easier than is obtainable with other programming methods and languages for DC motor control projects.

References

[1]. Tipsuwan, Y and Chow, M-Y, (1999), "Fuzzy Logic Microcontroller Implementation for DC Motor Speed Control," Proceedings of the IEEE 25th Annual Conference of the Industrial Electronics Society, 1999 (IECON '99), Volume 3,, pp.1271-1276.

[2]. Uma, G.; Chellamuthu, C, (2000), "Design and implementation of constant frequency soft switched regulated power supply for aerospace applications," Proceedings of the IEEE International Symposium on Industrial Electronics, 2000 (ISIE 2000), Volume 2,pp.735-740.

[3]. Carroll, B.; Klug, D.; Dimitrov, L.; Marmara, S.; Montemayor, R.; Schaffer, A.; Shaub, P.; Tse, E.; and Thompson, M. (1999), "Design of a motor speed controller for a lightweight electric vehicle," Proceedings of IEEE Conference on Electrical Insulation, Electrical Manufacturing & Coil Winding, Cincinnati OH, 1999, 26-28 Oct, 1999, pp.559-562.

[4]. Ismail, A.; and Sharaf, A.M., (2002), "An efficient neuro-fuzzy speed controller for large industrial DC motor drives ," Proccedings of the 2002 IEEE International Conference on Control Applications, Vol. 2, pp.1027-1031.

[5]. Soliman, H.F.; Sharaf, A.M.; Mansour, M.M.; Kandil, S.A.; and El-Shafii, M.H., (1994), "A fuzzy logic tunable speed controller for a rectifier fed PMDC motor drives," Proceedings of the 26th IEEE 1994 Southeastern Symposium on System Theory, Athens, OH, 20-22 March, 1994, pp.22-26.

[6]. Castagnet, T. and Nicolai, J. (1992), "Digital drive for home appliances DC motor," IEE Colloquium on Variable Speed Drives and Motion Control, London, 4th Nov, 1992, pp.6/1-6/4.

[7]. Al-Duwaish, H.; and Al-Hamouz Z. (1999), "Adaptive Output Feedback Controller for Wind Turbine Generators Using Neutral Networks", Electrical Machines and Power Systems, Vol.27, pp. 465-479, (ISI).

[8]. Takeshita, T.; and Matsui, N., (1994), "Sensorless Brushless DC motor drive with constant EMF identifier," Proceedings of the IEEE 20th International Conference on Industrial Electronics, Control and Instrumentation, 1994 (IECON '94), Bologna, Italy, Vol. 1, 5-9 Sept, 1994, pp.14-19

[9]. Gao, X.Z.; and Ovaska, S.J. (1999), "A New Fuzzy filterwith application in motion control systems," Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC '99, Tokyo, 12-15 October, 1999, Vol. 3, pp.280-285.

[10]. Jahmeerbacus, M.I.; Oolun, M.K.; Bhurtun, C.; and Soyjaudah, K.M.S. (1999), "Speed senorless control of a converter-fed DC motor," 1999 IEEE AFRICON, Cape Town, 28 Sept – 01 Oct, 1999, Vol. 1, pp. 453-456.

[11]. Matsui, N. (1996), "Sensorless PM brushless DC motor drives," IEEE Transactions on Industrial Electronics, Vol. 43, Issue 2, pp.300-308.

[12]. Umez-Eronini, I.C. (1999), System Dynamics and Control, PWS Publishing Company, pp.121-126.

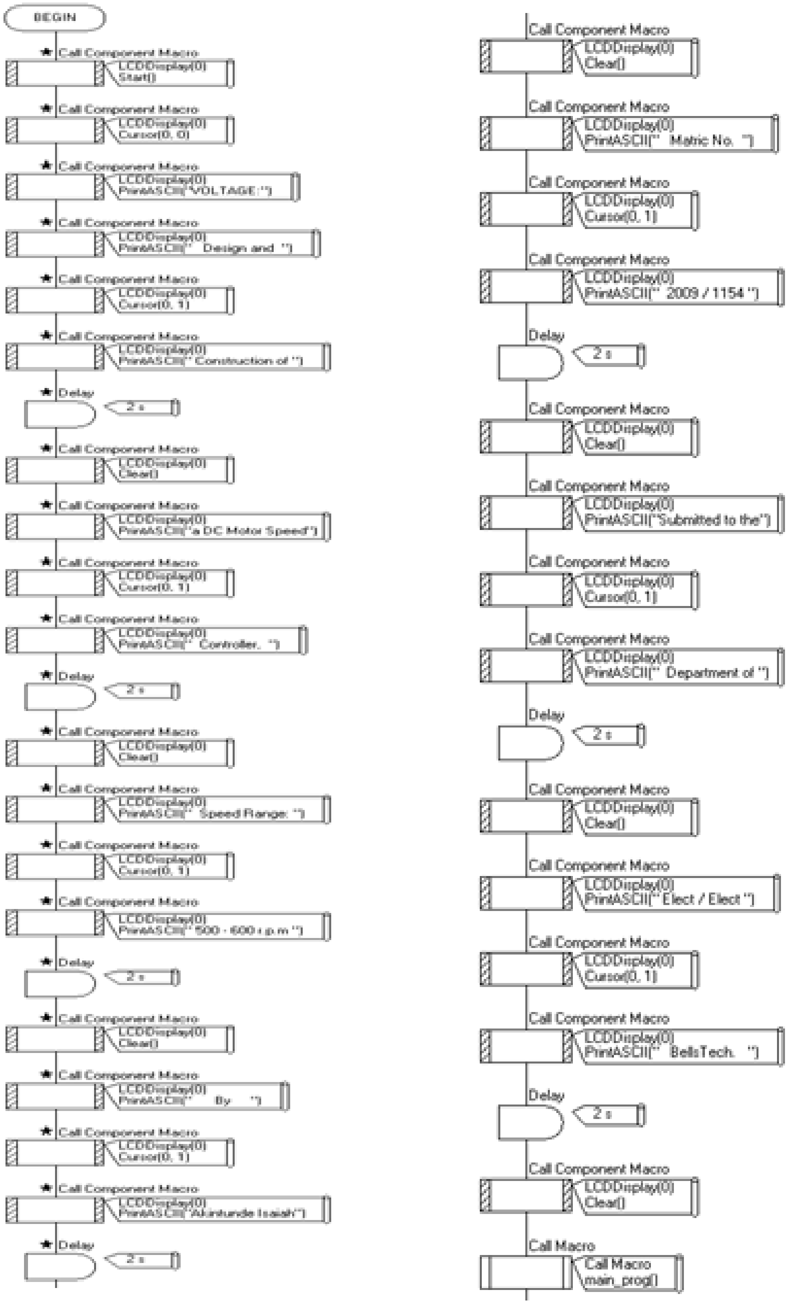
[13]. Akshay Mathur,(2008), "Microcontroller-Based DC Motor Speed Controller," Electronics For You, March 2008, pages 92 – 96.

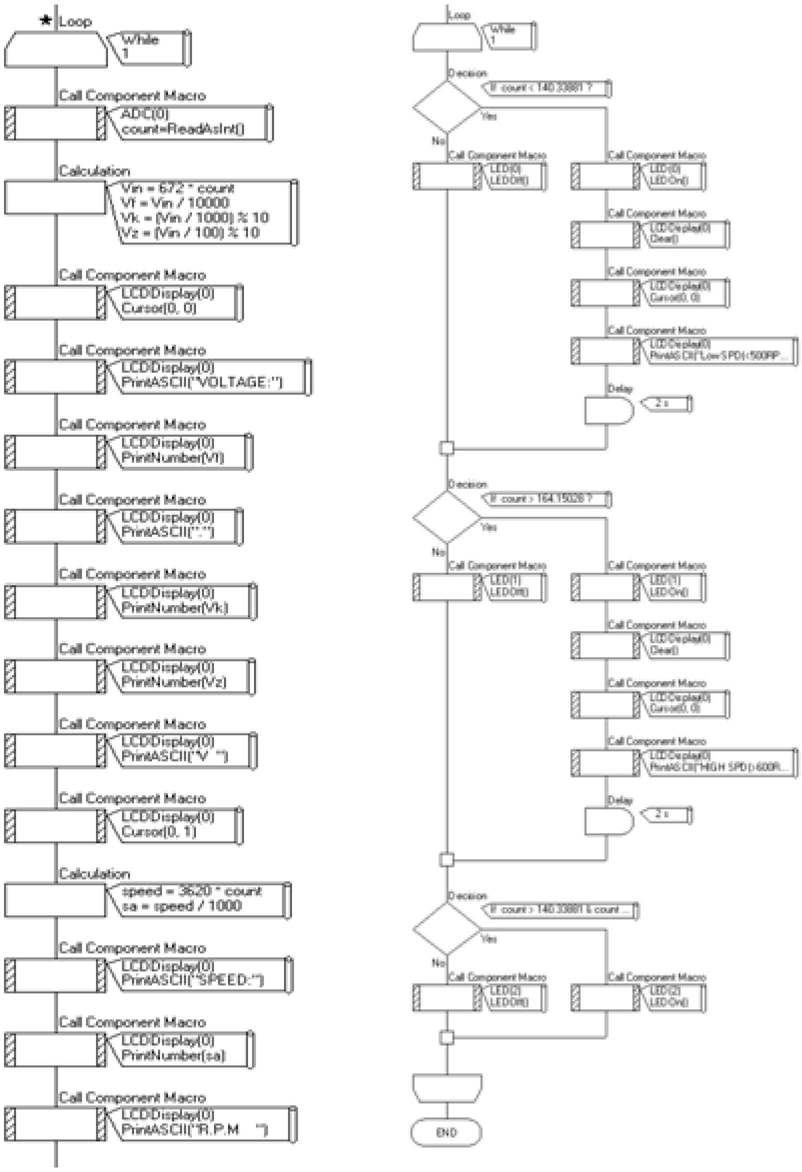
[14]. Ali Y.S.A.; Noor, S.B.M.; Bashi, S.M.; and Hassan M.K., (2003) , Microcontroller Performance for DC Motor Speed Control System," Proceedings of the IEEE National Power and Energy Conference, (PECon), Bangi, Malaysia, 2003, pp.104-109.

[15]. Shrivastava, S.; Rawat, J. and Agrawa, A., (2012), "Controlling DC Motor using Microcontroller (PIC16F72) with PWM," International Journal of Engineering Research (IJER), Volume No.1, Issue No.2, pp : 45-47 01 Dec. 2012

[16]. Kumar S.N. and Kumar, S.C. (2010), "Design and Implementation of Adaptive Fuzzy Controller for Speed Control of Brushless DC Motors," International Journal of Computer Applications, Volume 1 – No. 27, 2010, pp.36-41.

Appendix A: Flow Code Programme Chart




Loop
While
1
Call Component Macro
ADC(0)
count=ReadAsInt()
Calculation
Vin = 672 * count
Vf = Vin / 10000
Vk = (Vin / 1000) % 10
Vz = (Vin / 100) % 10
Call Component Macro
LCDDisplay(0)
Cursor(0, 0)
Call Component Macro
LCDDisplay(0)
PrintASCII("VOLTAGE:")
Call Component Macro
LCDDisplay(0)
PrintNumber(Vf)
Call Component Macro
LCDDisplay(0)
PrintASCII(".")
Call Component Macro
LCDDisplay(0)
PrintNumber(Vk)
Call Component Macro
LCDDisplay(0)
PrintNumber(Vz)
Call Component Macro
LCDDisplay(0)
PrintASCII("V ")
Call Component Macro
LCDDisplay(0)
Cursor(0, 1)
Calculation
speed = 3620 * count
sa = speed / 1000
Call Component Macro
LCDDisplay(0)
PrintASCII("SPEED:")
Call Component Macro
LCDDisplay(0)
PrintNumber(sa)
Call Component Macro
LCDDisplay(0)
PrintASCII("R.P.M ")
Loop
While
1
Decision
If count < 140.33881 ?
Yes
No
Call Component Macro
LED(0)
LEDOff()
Call Component Macro
LED(0)
LEDOn()
Call Component Macro
LCDDisplay(0)
Clear()
Call Component Macro
LCDDisplay(0)
Cursor(0, 0)
Call Component Macro
LCDDisplay(0)
PrintASCII("Low SPD(<500RP...
Delay
2 s
Decision
If count > 164.15028 ?
Yes
No
Call Component Macro
LED(1)
LEDOff()
Call Component Macro
LED(1)
LEDOn()
Call Component Macro
LCDDisplay(0)
Clear()
Call Component Macro
LCDDisplay(0)
Cursor(0, 0)
Call Component Macro
LCDDisplay(0)
PrintASCII("HIGH SPD(>600R...
Delay
2 s
Decision
If count > 140.33881 & count ...
Yes
No
Call Component Macro
LED(2)
LEDOff()
Call Component Macro
LED(2)
LEDOn()
END