PIC 16F877A Microcontroller Based Multiple DC Motors Controller

Suroor Moaid Dawood, Dr.Rabee' Hashim Thejeel
Department of Electrical Engineering-University of Basrah
Basrah-Iraq

***Abstract*- This project is mainly concerned on two DC motors speed, direction and operating period control system using three microcontrollers PIC 16F877A, one is the master and the two others are the slaves. It is a closed-loop real time control system, where rotary shaft encoder is coupled to the motor shaft to provide the feedback speed signal to the slave microcontroller. Pulse Width Modulation (PWM) technique is used to vary the voltage supplied to the DC motor armature(with fixed field voltage), thus controlling the motor speed. The PWM signal is generated by the slave microcontroller. This signal will guide the motor driver to vary the voltage supplied to motor to maintain at the required speed. The master microcontroller is programmed with the desired motor number, speed, operating period and direction of the motor from the 4×4 matrix keypad. Besides, it also lets the user monitor the performance of the system easily across the 4× 20 LCD display. The serial peripheral interfacing (SPI) mode is used for the communication between the master and the two slaves. The master sends the data which include motor number, speed, operating period and it's direction to the slaves by SPI communication technique.**

***Index Term*-- PIC 16F877A , DC motor, PWM, SPI, H-Bridge.**

I. INTRODUCTION AND RELATED WORKS

The design initiative is to have two slave processors with the same format, which uses one 16F877A as a dedicated CPU to control one DC motor. A single master 16F877A CPU controls and links the two slave CPUs. This design becomes a multiple processor environment that has a single master which takes the control commands from a user and passes the necessary control functions to an appropriate slave to perform the operations. With this design concept, there will be virtually no limit on the number of slaves in the system[1]. Expansion the role of existing small scale PIC microcontrollers into a multiple-microcontrollers system is the main focus of this work to resolve limited resource issues in meeting complex project needs[2]. The selected project approach is to modulate a design that uses a popular PIC 16F877A microcontroller to control a DC motor, and link each module through specialized protocols between the modules and the central command microcontroller responsible for communications between the system and the user. An additional focus of this project is the use of a high power DC motor control circuit with a heat sink[3]. Direct current motors have variable characteristics and are used extensively in variable-speed drives. DC motor can provide a high starting torque and it is also possible to obtain speed control over wide range[4]. Driving a system at the demanded speed and rotation is the main purpose of DC motor speed control[5]. The speed of a DC motor is directly proportional to the armature supply voltage[6]. A DC motor provides excellent speed control for acceleration and deceleration with effective and simple torque control[7].

A reasonable number of works have found in the literature, regarding the employment of microcontrollers devices for the control of DC motors, the paper of Yousef S. Ettomi[8] had described Motorola 68HC11 microcontroller based adjustable speed closed loop DC motor drive. An IGB switch is used in buck configuration with PWM technique to control the armature voltage of the motor. EA AI CHOON[4] had shown in his thesis how PIC 16F877A can be used for DC motor speed control. The controller will maintain the speed at desired speed when there is a variation of load. Gyorgy Gyorok and et.al.[9] had introduced a work dealt that the microcontroller controls of the motor speed by PWM, further determines the expected value by prediction algorithm and for the prediction an exact DC motor model needed is create. Payal P.Raval and Prof.C.R.mehta[10] had published a study on control techniques of PIC 16F877A microcontroller and MOSFET, mechanism assignments of analyzed by mainly focusing with the modeling and simulation of DC motor using MATLAB.

II. DESIGNE STEPS

To achieve a low cost and simple solution to the control of multiple high power DC motors, simple PIC 16F877A microcontroller is designated to meet this goal[11]. The first step is to design a circuit that uses the minimum resources of this processor to control a DC motor's speed, direction and operating periods. This module has to be fully tested and proven effective in controlling a single DC motor. The second step is to design a protocol link that uses the available resources of the controller to conduct the necessary communications between different modules of control through a variety of actions associated with the motors. The third step is to test the integrity of the communication protocols to ensure there is no miscommunication or unanticipated behavior in the controlled motors.

III. SOFTWARE DESIGN

Since there are two slaves and a single master in this control system design, two kinds of software are needed for this project. The major proof of concept in this project heavily relies on the software design. This paper will describe the master, slave protocol, and communication.

1- MASTER SOFTWARE

The master is the controlling microcontroller which handles all the controlling sequences, such as the interface between a user and the system, making sure the right motors are running

in the specified time, direction and speed. The master controller oversees major system components such as the keypad, LCD, and slave microcontrollers that run the motors.
In operation, the master starts with the keypad and LCD display module, handling interactions between the user's inputs and system's response. The keypad has been used to detect the user's input. The LCD displays the user commands and the two slaves response. A major portion of the software design in this project is the communication between the master and the two slaves. All the communications are initiated by the single master. Once the master has processed an action selected by the user input, it determines which action was chosen and transmits the instructions to the appropriate slave using serial communication (SPI). To control the two slaves, every slave has a unique slave select pin. There are predefined four bytes protocols that a master sends to both slaves in the system. Every communication sequence consists of the master broadcasting four bytes on the shared bus lines consisting of a clock (CLK), data out (DOUT), and data in (DIN). The first half byte is the direction of the motor, the second half of the first byte and the second byte is the motor speed, and the last two bytes is the motor running time period. Once the master receives the slave selected number from the keypad switch , it sends the four bytes according to the selected keypad's switches by the user. When all instructions are sent to the slaves, the master will return to start and wait for the next control sequence from the user through the keypad.

2- SLAVE SOFTWARE

The slave is in charge of executing what the master has commanded. It does not start processing information until the master is ready to send. A slave select pin has in each slave for SPI communication protocol to differentiate them. In the start, any one of slaves waits for its start command from the master to signal the start condition with a status of "0xxx" for slave1 and "1xxx" for slave2 as an initiate of the sending information. Once it receives the correct number, the slave waits for another status condition to receive the remain DC motor control bytes information including direction, speed and period of motor operation. When the slave is finished receiving the rest of the instructions, it activates the motor accordingly. The motor speed (12 bits) 1.5 byte is used to generate PWM (Pulse Width Modulation) signal to regulate the motor speed. The motor running period (16 bits) two bytes are used to determine when to shut down the gate of the PWM signal that will eventually stop the motor. When that is accomplished, the slave is ready for the next set of information from the master.

3-THE COMMUNICATION

There are basically four I/O lines (clock, data in, data out, and slave select) used in SPI communication. These are all shared as a serial bus between a master and two slaves. In each action of the serial communication bits streams, there are total of four bytes either transmit or receive between a master and any particular slave CPU. The predefined bytes are: (1) motor direction half byte, (2) speed 1.5 bytes, and (3) time period two bytes. The SPI of the 16F877A pin allocations are[11,12]:

1- Serial Clock (SCK) (RC3)
2- Serial Data In (SDI) (RC4)
3- Serial Data Out (SDO) (RC5)
4- Slave Select ($\overline{SS}$) (RA5) SS

Each chip needs its own program to operate the SPI port. The SPI outputs (SCK and SDO) need to be set as outputs in each MCU. The operation is controlled by SFRs (Special Function Registers) SSPSTAT (Synchronous Serial Port Status register) and SSPCON (Synchronous Serial Port Control register).
An algorithm has to be developed to make the microcontroller reads the input and respond accordingly. Therefore, this algorithm is established and represented as a flowchart as shown in Fig.1 for master PIC microcontroller, Fig.2 and Fig.3 for slave microcontroller. These flowcharts are then translated into C language and compiled using MikroC program.

IV. HARDWARE DESIGN

Although the hardware design appears complex, it has a lot of duplications due to multiple CPUs in the system. There are four major parts in this design: (1) the single master that handles the user's commands and communications via keypad and LCD module, (2) the two slaves that generates the PWM signals to drive the DC motors, (3) a simple four wires serial links between a single master and two slaves, and (4) the two DC motors H-bridge driver circuits.

1- MASTER CONTROL CIRCUIT

The master interface circuit has a standard 4x4 keypad, 20x4 LCD module, and four software controlled wires for the serial interface buses to the slave PICs. The keypad is directly interfaced to the PORTD (RD0-RD7). The LCD module is connected to the master through PORTB. RB5 (LCD_E) and RB4 (LCD_RS) are used for E (enable) and RS (register select) controls on the LCD display module. The SPI serial interface is accomplished through RC3 (marked as CLK) is used to generate the clock, RC5 (marked as DOUT) is the control data output from a master to the slaves, and RC4 (marked as DIN) is a return data line from the slaves to a master. RE0 as an output pin from the master connected to RA5 which is slave select pin of slave1 and finally, RE1 as an output pin from the master connected to RA5 which is slave select pin of slave2.

2- SLAVE CONTROL CIRCUIT

The serial communication interface is implemented through RC3 (CLK) is used to accept the clock signal from the master, RC5 (DOUT) is used to read the command bytes from the master, and RC4 (DIN) is used to send the data to the master, RA5 as slave select pin. The PWM signal is generated from the TMR2 timer via interrupt control on RC2 and RC1 pins to gate this PWM to a proper direction (either forward or reverse control of the motor). There is a rotary shaft encoder speed sensor on each motor. The signals are monitored on RC0 pin of each slave to provide feedbacks on the motor's speed.

3-MOTOR CONTROL CIRCUIT

The motor driver circuit is a standard H-bridge design. These bridge on-off controls are made through an IRFP460 power MOSFET that can handle 20A DC current[13]. The circuit can control a motor in either forward or reverse direction depending on the PWM signal that is coming in at its P.F or P.R terminal. The two rotary shaft encoders produce a clean feedback signal to the slave CPUs. The motor circuit H-bridge design is presented in Fig.4.

Table I shows the PIC 16F877A pin connection of the designed project.

V. MODELING OF SEPARATELY EXCITED DC MOTOR

From Fig.5.(Equivalent circuit of a separately exited DC motor), The armature voltage equation is given by[14]:

$$V_a = E_b + i_a R_a + L_a (di_a/dt) \qquad (1)$$

Now the torque balance equation will be given by:

TABLE I
PIC 16F877A PIN CONNECTION FOR TWO DC MOTORS SPEED CONTROL SYSTEM

Pin Name	Pin No.	Description	Application
V_{DD}	**11,32**	Positive Supply (+5V)	Power Supply to chip
V_{SS}	**12,31**	Ground Reference	Ground Reference
OSC1	**13**	For oscillator or resonator	Connected to resonator 4MHz with 22pF
OSC2	**14**		
MCLR	**1**	Reset Input	Always connected to +5V
Port B (RB0-RB7)	**33-40**	Input/Output pin	LCD Connection
Port D (RD0-RD7)	**19-22 & 27-30**	Input/Output pin	Keypad connection
Port E (RE0)	**8**	Input/Output pin	Connected to RA5/$\overline{SS}$ of Slave1
(RE1)	**9**	Input/Output pin	Connected to RA5/$\overline{SS}$ of Slave2
(RA5/$\overline{SS}$)	**7**	Input pin	SPI slave select input
Port C (RC0/T1CKI)	**15**	Input/Output pin	Count feedback speed pulses from shaft encoder
(RC1/CCP2)	**16**	Capture/Compare/PWM2	Output of duty cycle(PWM2) to control DC motor speed in CCW
(RC2/CCP1)	**17**	Capture/Compare/PWM1	Output of duty cycle(PWM1) to control DC motor speed in CW
(RC3/SCK)	**18**	Input/output pin	Synchronous serial clock I/O for SPI mode
(RC4/SDI)	**23**	Input pin	SPI data In
(RC5/SDO)	**24**	Output pin	SPI data Out

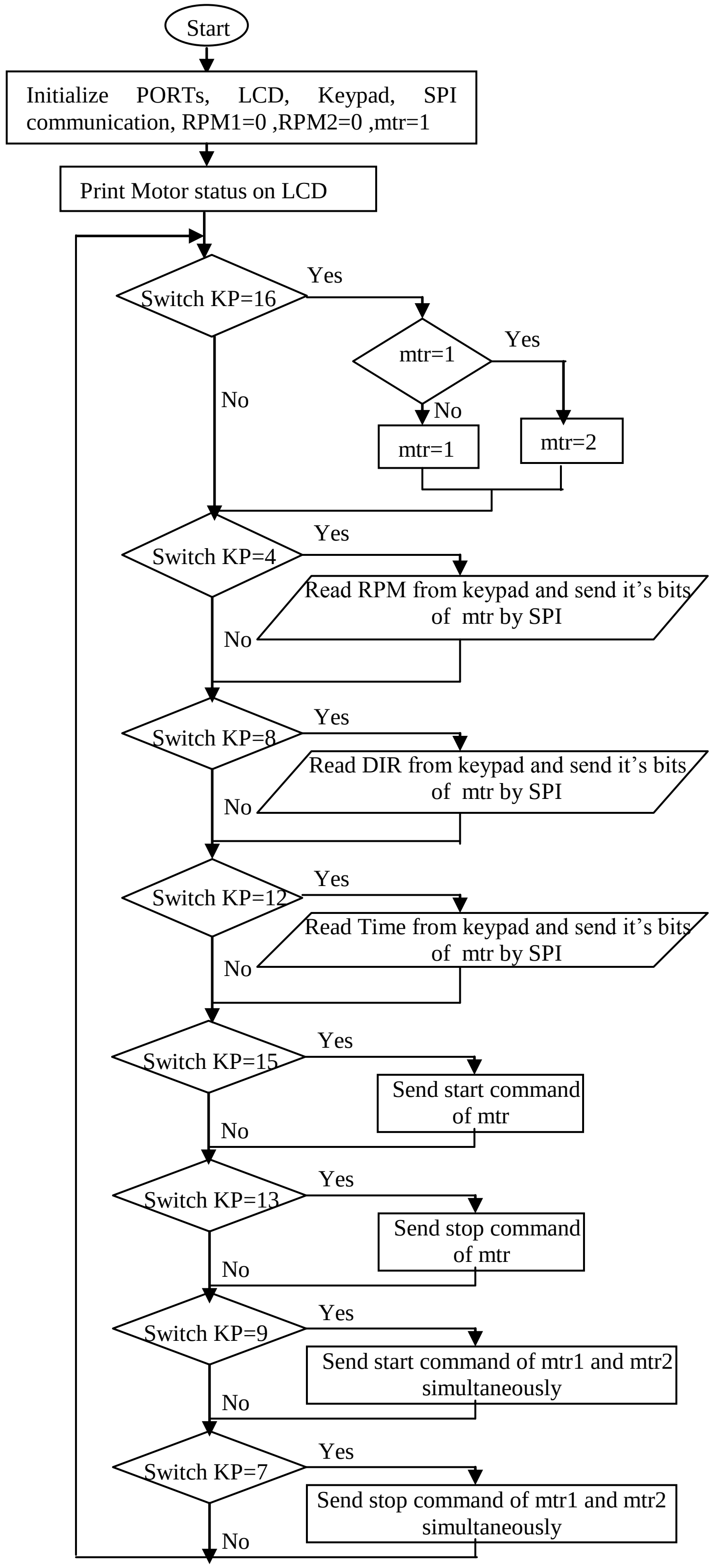


Fig.1 Flow Chart of Master MikroC Program

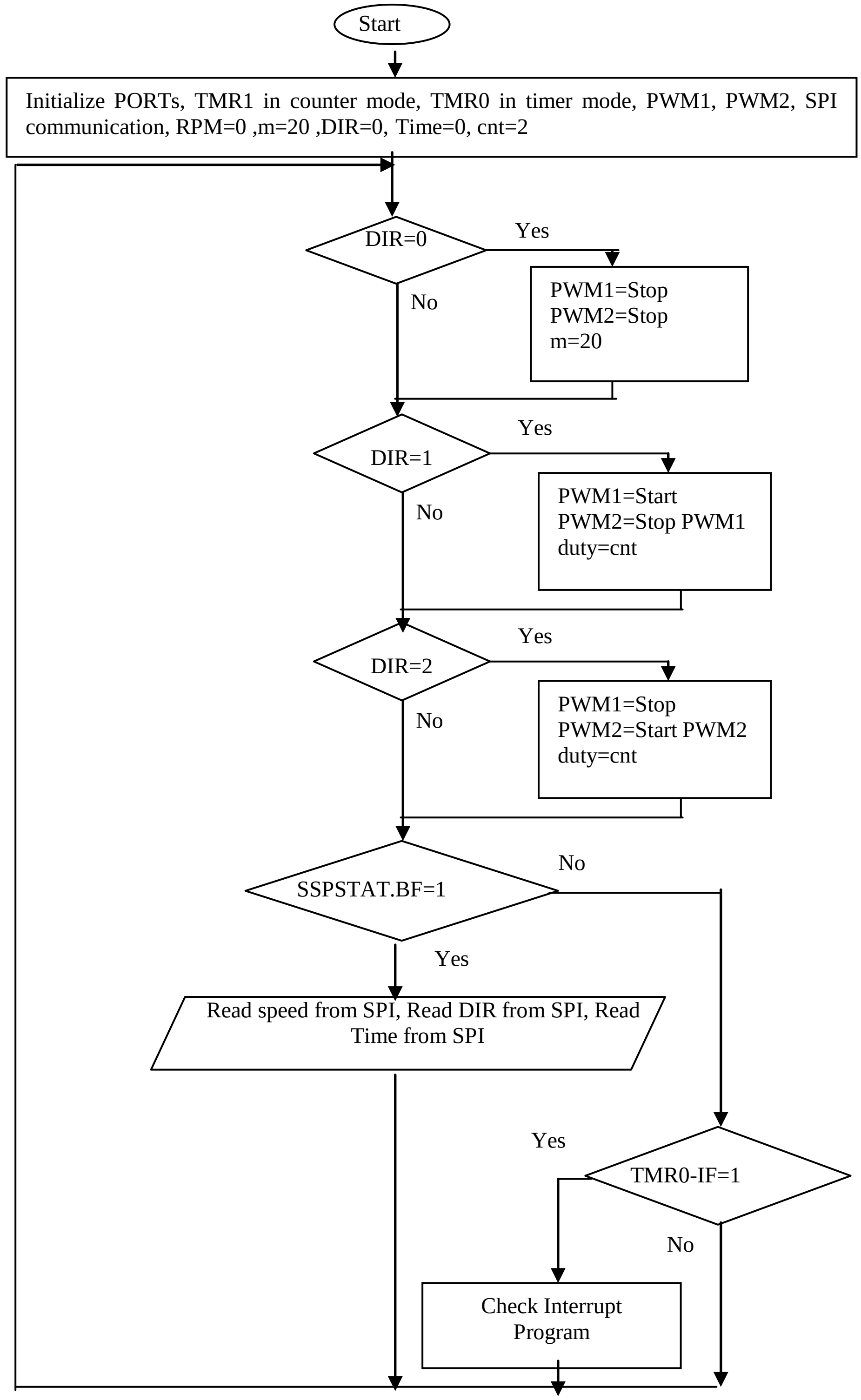


Fig.2 Flow Chart of Main MikroC Program in Slave

$T_m = J_m d_\omega/dt + B_m\omega + T_L$ (2)

where:

V_a is the armature voltage (V).

i_a is the armature current (A).

R_a is the armature resistance (Ω).

J_m is moment of inertia (kg/m²)

B_m is friction coefficient of the motor (Nm.sec/rad).

T_L is load torque in Nm.

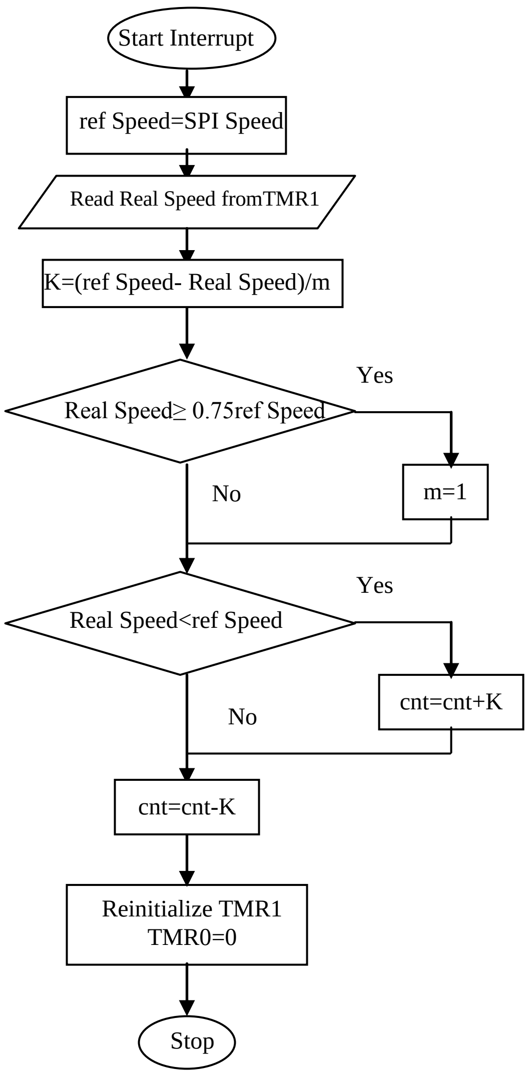


Fig.3 Interrupt MikroC Program in Slave

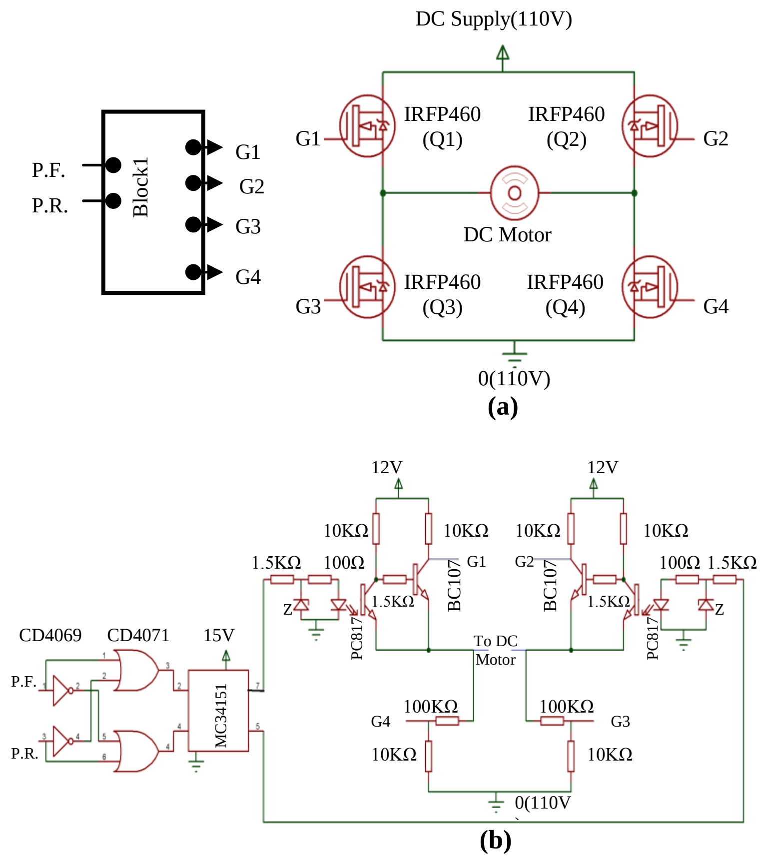


Fig.4 (a)The H-bridge DC motor driver (b) Block1

ω is angular velocity (rad/sec).

E_b is back EMF of the motor (V).

L_a is the armature inductance (H).

T_m is the mechanical torque developed (Nm).

Friction in rotor of motor is very small (can be neglected), so $B_m = 0$

Therefore, new torque balance equation will be given by:

$T_m = J_m d\omega/dt + T_L$ (3)

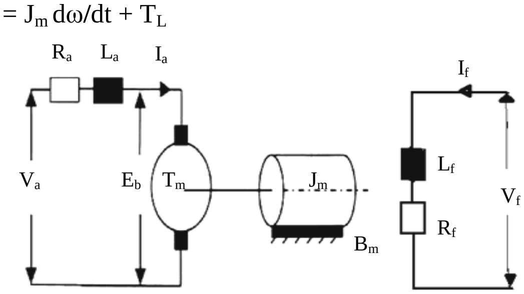


Fig.5 Equivalent circuit of a separately exited DC motor

Taking field flux as Φ (wb) and back EMF constant as K (v.sec/wb.rad). Equation for back EMF of motor will be:

$E_b = K\,\Phi\,\omega$ (4)

Also, $T_m = K\,\Phi\,I_a$ (5)

VI. H-Bridge DC Motor Driver

An H-Bridge is an electronic power circuit that allows motor speed and direction to be controlled. In this project DC motors are controlled from some kind of brain or microcontrollers PIC 16F877A to accomplish a mechanical goal. The microcontroller provides the instructions to the motors, but it cannot provide the power required to drive the motors. An H-bridge circuit inputs the microcontroller instructions and amplifies them to drive a mechanical motor. The H-bridge takes in the small electrical signal and translates it into high power output for the mechanical motor. An H-Bridge circuit allows a large DC motor to be run in both direction with a low level logic input signal[15].

The H-Bridge electronic structure is explicit in the name of the circuit -**H-bridge**. The power electronics actually form a letter H configuration, as shown in Fig.6(a) The switches are symbolic of the electronic Power MOSFETs which are used for switching. If it is desired to turn the motor on in the **forward** direction, switches 1 and 4 must be closed to power the motor as in Fig.6(b) below. If it is desired to turn the motor on in the **reverse** direction, switches 2 and 3 must be closed to power the motor as in Fig.6(c).

VII. PWM METHOD

Pulse width modulation (PWM) is a method for binary signals generation, which has two signal periods (high and low). The width (W) of each pulse varies between 0 and the period (T). The main principle is control of power by varying the duty cycle. Here the conduction time to the load is controlled. The duty cycle can be varied from 0 to 1 by varying t_{on} or T[16] .

Therefore, the average output voltage V_{avr} can be changed between 0 and V_{in} by controlling the duty cycle, thus, the power flow can be controlled.

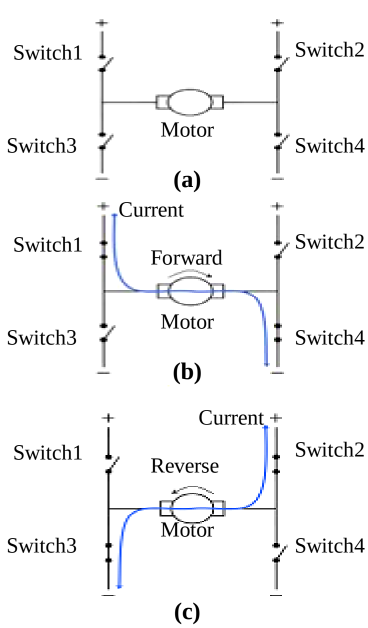


Fig.6.(a)H-Bridge Topology.(b)Forward Direction.(c)Reverse Direction.

The on-off switching is performed by power MOSFETs. A MOSFET is a device that can turn very large currents on and off under the control of a low signal level voltage. PWM signal shown in Fig.7.

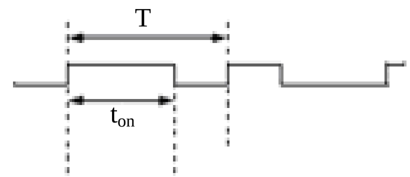


Fig.7 PWM signal

The average voltage that supply to DC motor is given by[7]:

$$V_{avr} = \frac{t_{on}}{T} \times V_{in} \qquad (6)$$

where:

V_{avr}= average voltage supply to DC motor.

t_{on} = time ON of switches.

T = period of PWM.

$\frac{t_{on}}{T}$ = duty cycle.

VIII. TWO DC MOTORS SPEED CONTROLLER

For precise speed control of dc motors system, closed-loop control is normally used. Basically, the block diagram of the speed control is shown in Fig.8. The speed, which is sensed by sensing devices (rotary shaft encoders E50S8-1000-3-T-24)[17] is compared with the reference speed to generate the error signal and to vary the armature voltage of the motor. The schematic diagram of the presented project shown in Fig.9.

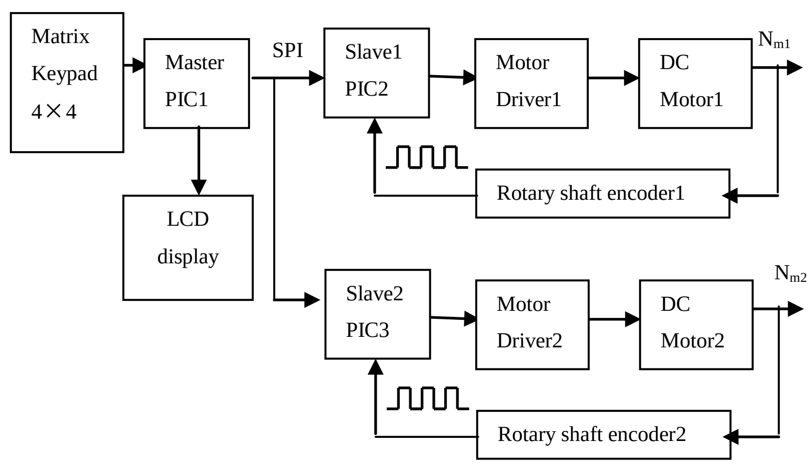


Fig.8. Block Diagram of two DC Motors controller

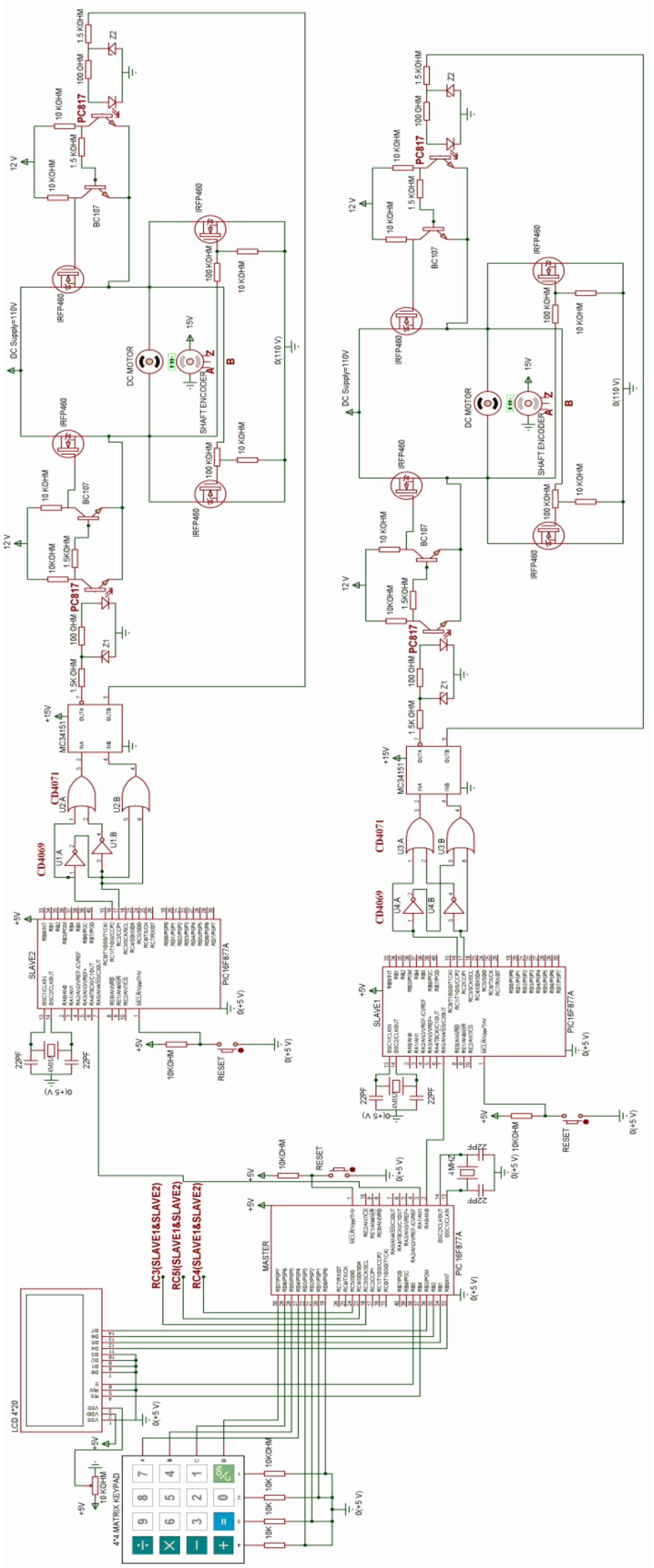


Fig. 9. Schematic Diagram of a PIC 16F877A Based on two DC Motor Controller

The predefined command bytes of the serial communication between master and two slaves and their associated functions are found in Table II.

The prototype system that has one master and two slaves with two DC motors is presented in picture shown in Fig.10.

TABLE II
COMMAND BITS AND FUNCTIONS

#	Command bits	Control Function
0	0000	Motor Direction Control of Slave1
1	0001	First Part of Slave1 Speed Motor Control
2	0010	Second Part of Slave1 Speed Motor Control
3	0011	Third Part of Slave1 Speed Motor Control
4	0100	First Part of Slave1 Time Motor Control
5	0101	Second Part of Slave1 Time Motor Control
6	0110	Third Part of Slave1 Time Motor Control
7	0111	Fourth Part of Slave1 Time Motor Control
8	1000	Motor Direction Control of Slave2
9	1001	First Part of Slave2 Speed Motor Control
10	1010	Second Part of Slave2 Speed Motor Control
11	1011	Third Part of Slave2 Speed Motor Control
12	1100	First Part of Slave2 Time Motor Control
13	1101	Second Part of Slave2 Time Motor Control
14	1110	Third Part of Slave2 Time Motor Control
15	1111	Fourth Part of Slave2 Time Motor Control

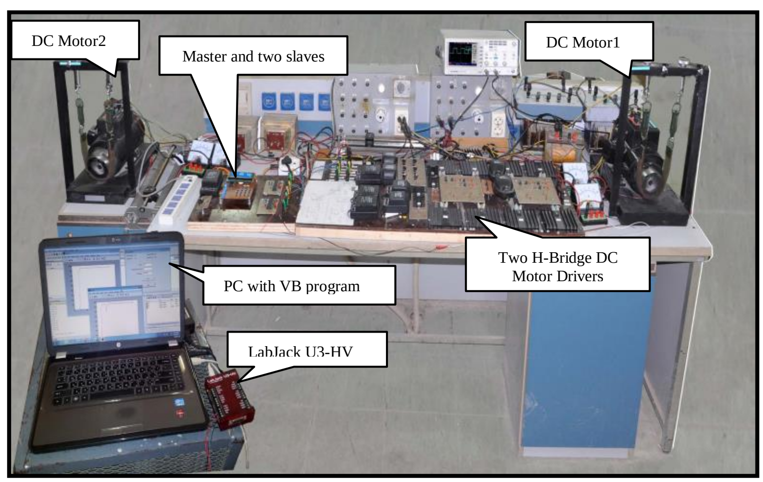
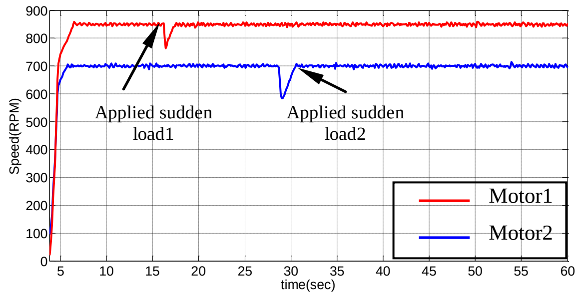


Fig.10. Picture of the designed project

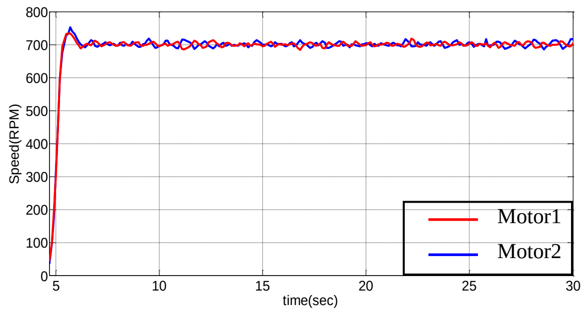
IX. THE EXPERMENTAL RESULTS

To monitoring the system response, the curves of speed versus time has been drawn for different speeds, operating periods, with and without sudden load applied on the DC motors. The experimental results of this presented work obtain by PC using Visual Basic and LabJack U3-HV device[18]as a data consist of (time (sec), speed of DC motor1 (RPM) and speed of DC motor2 (RPM)). In Fig.11(a) both motors are operated simultaneously for 60sec but at different speeds, motor1 run at 850 RPM with sudden load applied at 15.716749 sec and motor2 run at 700 RPM with sudden load applied at 13.3915sec.



(a)

In Fig.11(b) the two motors run simultaneously at 700 RPM for 30 sec without any sudden load.



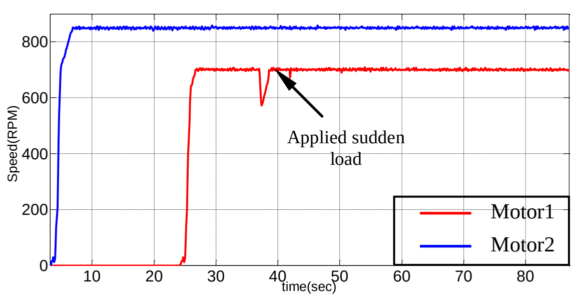
(b)

The small difference between the two DC motors speeds belong to the difference in the DC motors parameters as field resistance(R_F) and armature resistance (R_a) as in Table.III.

TABLE III

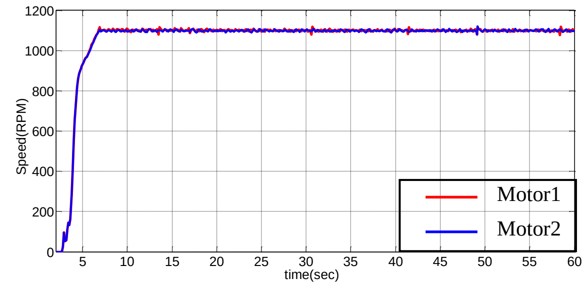
R_{F1}	826Ω
R_{F2}	865.5 Ω
R_{a1}	13.902 Ω
R_{a2}	11.903 Ω

In Fig.11(c) the motors are operated each one alone with different speeds, motor2 run at 850 RPM without sudden load for 85 sec and motor1 run at 700 RPM for 60sec start at 24.3876sec with sudden load applied at 36.99126sec.



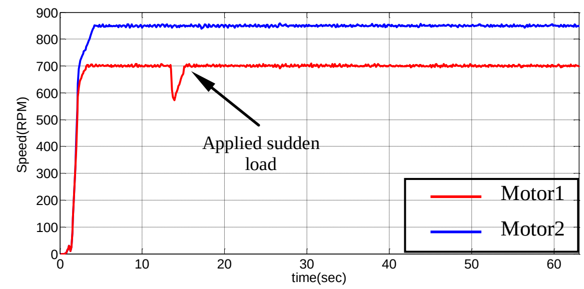
(c)

While in Fig.11(d) the two motors run simultaneously at 1100 RPM for 60 sec without any sudden load.



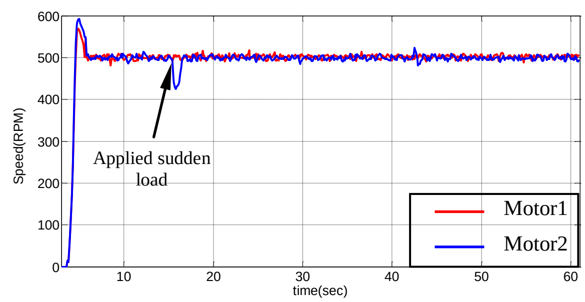
(d)

Fig.11(e) show that both motors are operated simultaneously for 63 sec but at different speeds, motor1 run at 700 RPM with sudden normal load applied at 13.446266 sec and motor2 run at 850 RPM without any sudden load.



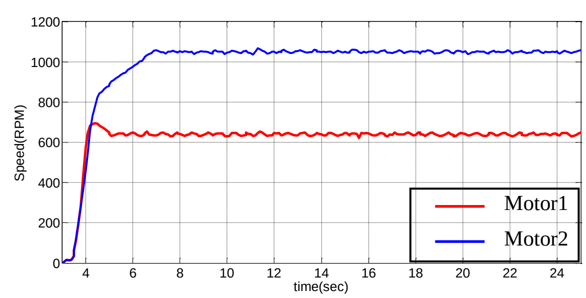
(e)

Fig.11(f) represented that both motors are operated simultaneously at the same speed 500 RPM for 60 sec but motor1 run without any sudden load and motor2 run with sudden load applied at 15.343489 sec.



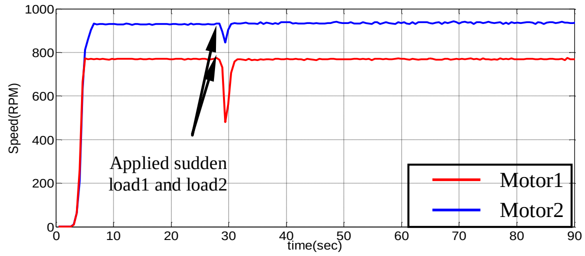
(f)

From Fig.11(g) you can see that both motors are operated simultaneously without any sudden load for 25 sec but at different speeds, motor1 run at 650 RPM and motor2 run at 1050 RPM.



(g)

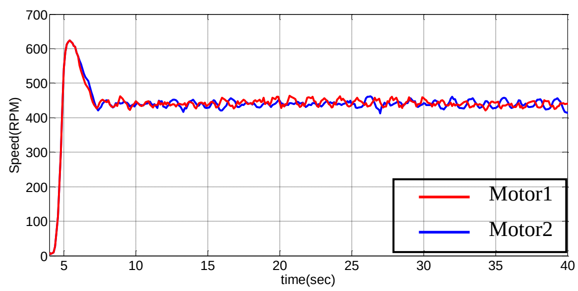
In Fig.11(h) the two DC motors run in the same time for 90 sec but at difference speeds and sudden loads, motor1 run at 770 RPM and motor2 run at 930 RPM. The sudden loads applied on both DC motors at the same instant which is 28.335sec.



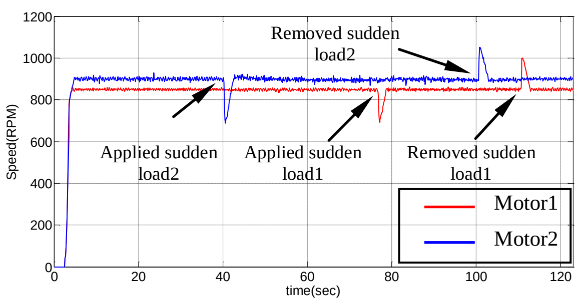
(h)

Fig.11(i) shown that the two motors run simultaneously at 450 RPM for 40 sec without any sudden load.

Fig.11(j) represented that both motors are operated simultaneously for 123sec at different speeds and sudden applied loads. Motor1 run at speed 850 RPM with sudden applied load at 76.439sec and the instant of load removal is 110.539sec, but motor2 run with speed 900 RPM with sudden applied load at 40.283 sec and the instant of this load removal is 100.5598 sec.

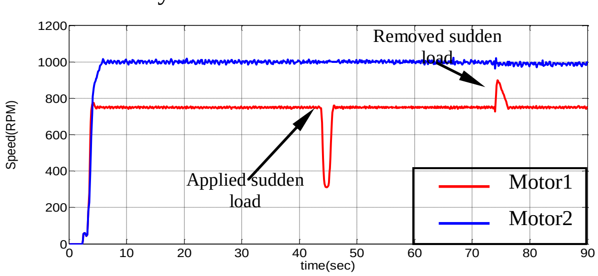


(i)



(j)

In Fig.11(k) both motors are operated simultaneously FOR 90sec at different speeds. Motor1 run at speed 750 RPM with sudden applied load at 43.52565 sec and the instant of load removal is 73.8746 sec, but motor2 run with speed 1000 RPM without any sudden load.



(k)

Fig.11.Experimental Results

From these results it is clearly observed how the system performance and speed control are improved.

X. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

This paper has presented a study of the two DC motors speed ,direction of rotation and their operating period controllers design using PIC microcontroller 16F877A. The controller will maintain the speed at desired speed when there is a variation of load. By varying the PWM signal from microcontroller slaves to the motor driver, motor speed can be controlled back to desired value. The proposed method for control reduces the number of components because the microcontroller can integrate in one package all the functions. Thus, the proposed technique suited for industrial applications.

This project gives a reliable, durable, accurate and efficient way of speed control of two DC motors.

For future works, some recommendations have been suggested in order to improve the presented project:

1-Hardware Design Implementation

It can be used another rotary shaft encoder of 5000 pulses (as an example) for more number of slots. So, this will allow us to increase the resolution of shaft encoder by increasing the number of pulses. Thus, it will reduce the error besides, time for getting data and for controller to take action also can be reduced. So, the DC motors speed response will become better.

2- Software Design Improvement

It is possible to use a fuzzy logic microcontroller which combine the idea of fuzzy logic in microcontroller to improve the system performance or implement the conventional fuzzy PID controller with PC as a controller. To monitor the performance of the system, use Visual Basic 6.0, this program is able to send data (desired speed from user) to microcontroller and plot a graph of detected speed versus time this is done by using RS232 for serial communication interface with PIC microcontroller.

REFERENCES

[1] Steve Hsiung, Tyson McCall and Corinne Ransberger, "Implementing Simple Protocols in Multiple Processors Control Applications", Engineering Technology Department, Old Dominion University, Norfolk, American Society for Engineering Education Annual Conference & Exposition, 2005.

[2] Dr. Steve C. Hsiung, "The Use of PIC Microcontrollers in Multiple DC Motors Control Applications", Journal of Industrial Technology, Volume 23, Number 3 - July 2007 through September 2007.

[3] S. Hsiung and A. Dean, "The Design and Implementation of Multiple Processors in Control of Multiple High Power DC Motors", Dept. of Engineering Technology, Old Dominion University, 2006, IJME INTERTECH.

[4] EA AI Choon," DC Motor Speed Control Using Microcontroller PIC 16F877A", MSC.Thesis, Faculty of Electrical Engineering University Technology Malaysia, MARCH 2005.

[5] Mohd Amir Fikri Bin Awang, " DC Motor Speed Controller", MSC.Thesis, Faculty of Electrical and Electronics Engineering University Malaysia Pahang, 11 NOVEMBER 2010.

[6] Ahmed M. Adday and Dr. Mazin Z. Othman, "Combined Armature and Field Fuzzy Speed Control of a DC Motor for Efficiency

Enhancement",Al-Rafidain Engineering, Vol.20, No. 6 , PP.117-129, December 2012.

[7] Prahlad Kumar Sahoo and Nirmal Kumar Barik, " Speed Control of Seperately Exited DC Motor Using Self Tuned Fuzzy PID Controller",MSc.thesis, National Institute of Technology Rourkela, Orissa, 2010-2011.

[8] Yousef S. Ettomi Ali, " Microcontroller Based Adjustable Speed Closed-Loop DC Motor Drive", PHD.Thesis, Universiti Putra Malaysia, February 2004.

[9] Gyorgy Gyorok, Margit Mako, and Jozsef Lakner, " Sensorless D.C. Motor Control with Predictive Sensig Method", Alba Regia University Center, Obuda University e-Bulletin, Vol. 1, No. 1, 2010.

[10] Payal P.Raval and Prof.C.R.mehta, " Modeling, Simulation and Implementation of Speed Control of DC Motor Using PIC 16F877A", International Journal of Emerging Technology and Advanced Engineering, ISBN 2250-2459, Volume 2, Issue 3, March 2012.

[11] PIC16F877A Data Sheet, Microchip Technology Inc., 2003.

[12]M.Bates,"Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation", Elsevier, ISBN-13: 978-0-7506-8028-8, 2006.

[13] IRFP460 data sheet, International Rectifier.

[14] A. Faizy and Sh. Kumar," DC Motor Control Using Chopper", MSc. thesis, Department of Electrical Engineering, National Institute of Technology, Rourkela, India, 2010.

[15] V.Sieben," A High Power H-Bridge", ARVP - Autonomous Robotic Vehicle Project, September 2003 - Revision 1.0.

[16] S. Kamal Viswanath, M. Gowtham, P. Ashok Kumar and K. Rohit Naik " Speed Control of a DC Motor Using Microcontroller 8051", a project report, Gokaraju Rangaraju Institutes of Engineering and Technology, January 2011.

[17] E50S8-1000-3-T-24 Rotary shaft encoder Data Sheet.

[18] LabJack U3 User’s Guide (-LV & -HV), (Hardware Revision 1.30) Revision 1.07 September 18th,2008 ,LabJack Corporation.