PROJECT REPORT

On

“FOUR QUADRANT DC MOTOR DRIVER USING MICORCONTROLLER”

Submitted in the partial fulfillment of the requirement for the

Fourth Year of Engineering

(Electrical Engineering)

By

Babasaheb G. Chavan (1921321293019)

Pawan S. Lekurwale (1921321293020)

Ashish S. Gadekar (1921321293021)

Under the guidance of

Prof. N.S. HUSSAIN

Department of Electrical Engineering,

Jawaharlal Nehru Engineering College, Aurangabad.

(2022-2023)

CERTIFICATE

This is to certify that project titled

"FOUR QUADRANT DC MOTOR DRIVER USING MICORCONTROLLER"

Submitted By

Babasaheb G. Chavan (1921321293019)

Pawan S. Lekurwale (1921321293020)

Ashish S. Gadekar (1921321293021)

Is completed as per requirement of

Dr. Babasaheb Ambedkar Technological University Lonere (Raigad).

In partial fulfillment of Fourth Year of Engineering

(Electrical Engineering)

For Academic year 2022-2023

Prof. N.S. Hussain	**Prof. B.T. Deshmukh**	**Dr. H.H. Shinde**
Guide	H.O.D.	Principal

Jawaharlal Nehru Engineering College, Aurangabad.

ACKNOWLEWLEGEMENT

Through this we present our sincere thanks to the countless individuals who have helped us in completing this work. First and foremost, we would like to thank our project guide **Prof.N.S.HUSSAIN** for his valuable guidance and help. At every step he provided us with his Valuable time, meticulous guidance and moral support throughout the dissertation entitled:

"FOUR QUADRANT DC MOTOR DRIVER USING MICORCONTROLLER"

Also, we are very thankful to **Prof. B. T. DESHMUKH,** Head of Department, Electrical Engineering, MGM's Jawaharlal Nehru Engineering College, Aurangabad for his suggestio0ns, encouragement and constant support. We are also grateful to **DR. H. H. SHINDE,** Principal, MGM's Jawaharlal Nehru Engineering College, Aurangabad.

Last but never the less, we thank our family and friends for their support and encouragement.

Babasaheb G. Chavan (1921321293019)

Pawan S. Lekurwale (1921321293020)

Ashish S. Gadekar (1921321293021)

LIST OF CONTENTS

CONTENTS

LIST OF FIGURES

ABSTRACT

Speed control of a machine is the most vital and important part in any industrial organization. This paper is designed to develop a four-quadrant speed control system for a DC motor using microcontroller. The motor is operated in four quadrants i.e., clockwise, counter clock-wise, forward brake and reverse brake. It also has a feature of speed control. The four-quadrant operation of the dc motor is best suited for industries where motors are used and as per requirement they can rotate in clockwise, counter-clockwise and also apply brakes immediately in both the directions. In case of a specific operation in industrial environment, the motor needs to be stopped immediately. In such scenario, this proposed system is very apt as forward brake and reverse brake are its integral features. Instantaneous brake in both the directions happens as a result of applying a reverse voltage across the running motor for a brief period and the speed control of the motor can be achieved with the PWM pulses generated by the microcontroller. The microcontroller used in this project is from 8051 family. Push buttons are provided for the operation of the motor which are interfaced to the microcontroller that provides an input signal to it and controls the speed of the motor through a motor driver IC. The speed and direction of DC motor has been observed on digital CRO. Microcontroller programming has been written in assembly language by using notepad and it has been converted in hex file by using micro vision Kiel. The burning of programming in the 8051-microcontroller chip has been done by using positron boot loader software.

Keywords: DC motor, AT89C51 Microcontroller, Motor Driver (L293D), Voltage Regulator (LM7805), Push Buttons, PWM.

CHAPTER 1

INRODUCTION

1.1 General Introduction

DC machines play a very important role in industries and in our daily life. The outstanding advantage of DC machines is that they offer easily controllable characteristics. This paper is designed to develop a four-quadrant speed control system for a DC motor using microcontroller. The motor is operated in four quadrants i.e., clockwise, counter clock-wise, forward brake and reverse brake. It also has a feature of speed control. The four-quadrant operation of the dc motor is best suited for industries where motors are used and as per requirement they can rotate in clockwise, counter-clockwise and also apply brakes immediately in both the directions. In case of a specific operation in industrial environment, the motor needs to be stopped immediately. In such scenario, this proposed system is very apt as forward brake and reverse brake are its integral features. In this work the concept of four quadrant speed control i.e., clockwise movement, anticlockwise movement, instantaneous forward braking and instantaneous reverse braking of a dc motor with the help of microcontroller through motor driver (L293D) has been proposed.

1.2 Objective

In this work the concept of four quadrant speed control i.e., clockwise movement, anticlockwise movement, instantaneous forward braking and instantaneous reverse braking of a dc motor with the help of Arduino through motor driver (L293D) has been proposed. The same application is used in many areas such as to control the rudder of airplane, electric bicycle or an electric car. The motor is built from a stationary magnet and rotating coils. The brushes supply electric current to the coil that is close to the motor magnet pole pieces .As the rotor turns, the polarity of each coil is reversed and sustained rotation of motor is achieved. The direction of rotation depends on the polarity of brushes. The torque applied on the rotor depends on the current passing through the coils. The steady speed depends on the current flowing through the coils and load being driven by the motor.

CHAPTER 2

LITERATURE SURVEY

A four-quadrant or multiple-quadrant operation is required in industrial as well as commercial applications. These applications require both driving and braking, i.e., motoring and generating capability. Some of these applications include electric traction systems, cranes and lifts, cable laying winders, and engine test loading systems [2].

The different quadrant operations drive the motor with normal as well as reversal of both voltage and currents so as to run as well as to break the motor either in forward or reverse directions. A DC motor may operate in one or more modes (or quadrant) in variable speed applications [4]. The major advantage of using DC motor is that the ease of its control. The speed of the DC motor is controlled by applying a variable DC input for below rated speed control. For above rated speeds, the motor is controlled by applying variable current through its field winding. For reversing the direction of rotation, either polarity of the supply voltage (which is applied to armature terminals) or the direction of field current has to be changed. By using DC motors, it is possible to obtain smooth speed control over a wide range in clockwise as well as anti-clockwise directions [1]. The torque of a DC motor is proportional to the armature current which in turn depends on the difference between back emf and applied voltage.

Therefore, it is possible to make the motor to develop positive or negative torque simply by controlling the applied voltage to a greater or lesser than the back emf. Thus, an armature-controlled DC machine is inherently capable of operating different modes or quadrants, generally it is known as four-quadrant operation of a motor [3]. In multi-quadrant operation or four quadrant operation, motor accelerates or decelerates depending on whether motor torque is lesser or greater than load torque. During motor acceleration, it should supply not only the load torque, but an additional component of load current to overcome the inertia. Motor positive torque produces the acceleration in forward direction. In this, the motor speed is positive when the motor is rotating in forward direction [6]. During motor deceleration, the resultant or dynamic torque has a negative sign. This torque assists with motor developed torque and maintains the motion by extracting the energy from stored energy. Hence the motor torque is considered as negative if it produces deceleration. A motor can be controlled in such a way that it operates in two cases; motor action and braking action [7].

Motor action converts the electric energy into mechanical energy and it produces forward motion, hence it called as motoring action, whereas braking action converts mechanical energy to electrical energy which gives forward braking motion, it is termed as generator [3].

Similarly, these two actions are performed in case motor operating in reverse direction, i.e., (reverse motoring and reverse braking actions). Before beginning our discussion on how a DC motor works in 4 quadrants, we will look at the four-quadrant operation of a motor driving a hoist load as shown in figure below [5]. This hoist consists of a cage with or without any load. A rope, generally made up of a steel wire is wounded on a drum to raise the cage and a balance weight. This balance weight or counterweight magnitude is greater than that of empty cage, but less than the loaded cage. For each quadrant of operation, direction of rotation, w, load torque, TL, and motor torque Tm are shown in figure. Consider that the load torque is constant and independent of motor speed [6].

CHAPTER 3

SYSTEM DEVELOPMENT

The project work has been divided into two parts. In the first part simulation is done using proteus software and in second part a prototype model is developed and the result is verified using a prototype hardware model.

3.1 The total project consists of four sections i.e.

1. System Overview:
2. Four Quadrant Operation of DC Motor:
3. Pulse Width Modulation:
4. Motor Driver IC:

3.1.1 System Overview:

The design was broken down into different modules to simplify the circuit design. Figure1 describes the block diagram of overall system for the four-quadrant speed control of dc motor.

The circuit uses standard power supply comprising of a step-down transformer from 230V to 12V and the four diodes forming a bridge rectifier that delivers pulsating dc which is unregulated is regulated to constant 5V dc. The output of the power supply which is 5V is connected to the 40pin of microcontroller and ground is connected to 20pin. Pin no 1 to 7 of port 1 are connected to switches. Pin no 21, 22, 23 of microcontroller are connected to input 1,2, enable pins of motor driver L293D.Pin 3 and 6 are connected to motor terminals.

3.1.2 Four Quadrant Operation of DC Motor:

There are four possible modes or quadrants of operation using a DC Motor which is depicted in Figure 2. When DC motor is operating in the first and third quadrant, the supplied voltage is greater than the back emf which is forward motoring and reverse motoring modes respectively, but the direction of current flow differs. When the motor operates in the second

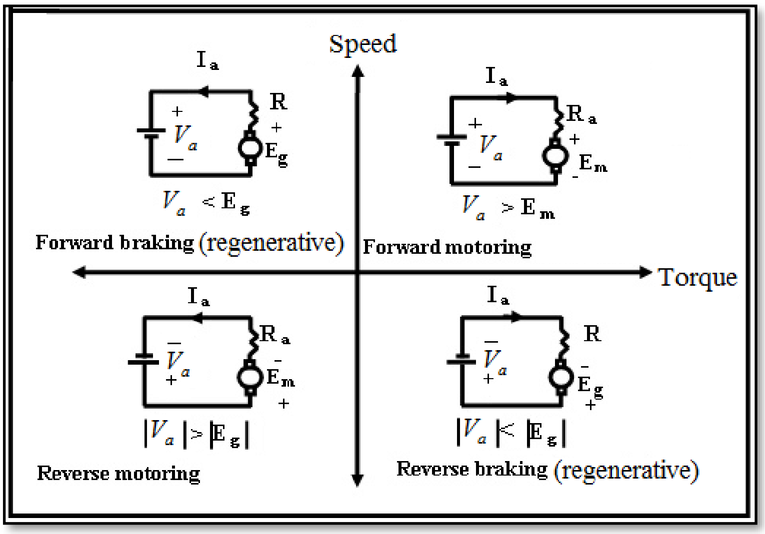


Figure 3.1 : Four Quadrant Operation Characteristics of DC Motor

and fourth quadrant the value of the back emf generated by the motor should be greater than the supplied voltage which are the forward braking and reverse braking modes of operation respectively, here again the direction of current flow is reversed.

3.1.3 Pulse Width Modulation:

Pulse width Modulation (PWM) is the term used to describe using a digital signal to generate an analogue output signal. PWM is one of the powerful techniques used in control systems today. This is usually used to control the average power to a load in a motor speed control

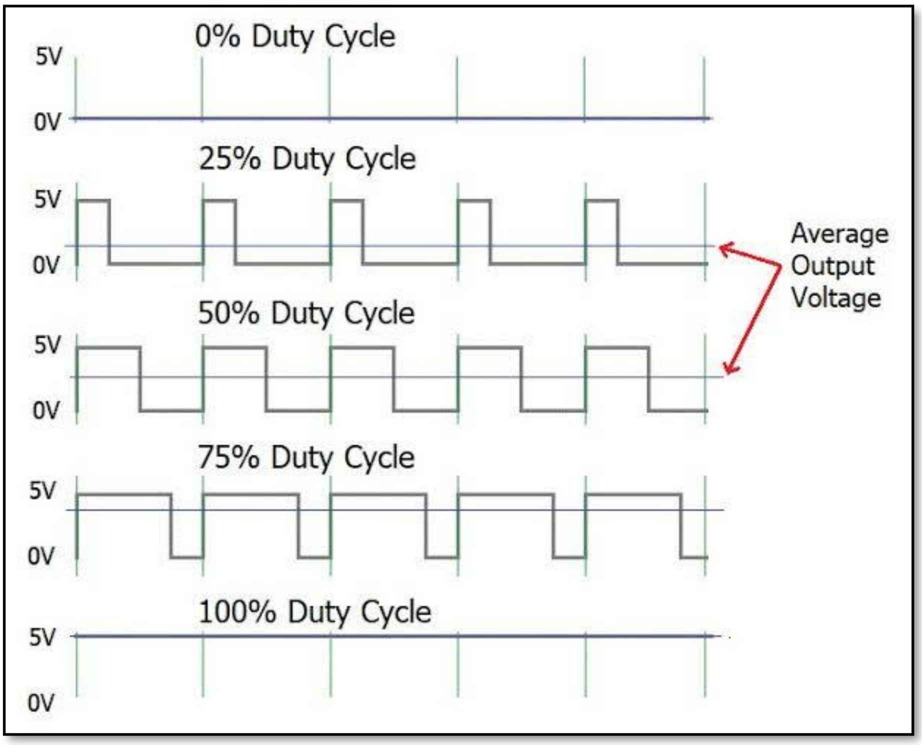


Figure 3.2 : Pulse Width Modulation

circuit. It is used in wide range of application which includes: speed control, power control, measurement and communication.

Pulse-width modulation (PWM) is a commonly used technique for controlling power to an electrical device, made practical by modern electronic power switches. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. The term duty cycle describes the proportion of on time to the regular interval or period of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on.

The main advantage of PWM is that power loss in the switching devices is very low. When a switch is off there is practically no current, and when it is on, there is almost no voltage drop across the switch. Power loss, being the product of voltage and current, is thus in both cases close to zero. PWM works also well with digital controls, which, because of their on/off nature, can easily set the needed duty cycle. PWM has also been used in certain communication systems where its duty cycle has been used to convey information over a communications channel. The duty cycle determines the speed of the motor. The desired speed can be obtained by changing the duty cycle. The PWM in microcontroller is used to control the duty cycle of DC motor. The

PWM pulses generated from the microcontroller are viewed for various duty cycles in the simulation done in proteus software.

3.1.4 Motor Driver IC:

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively. Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high impedance state.

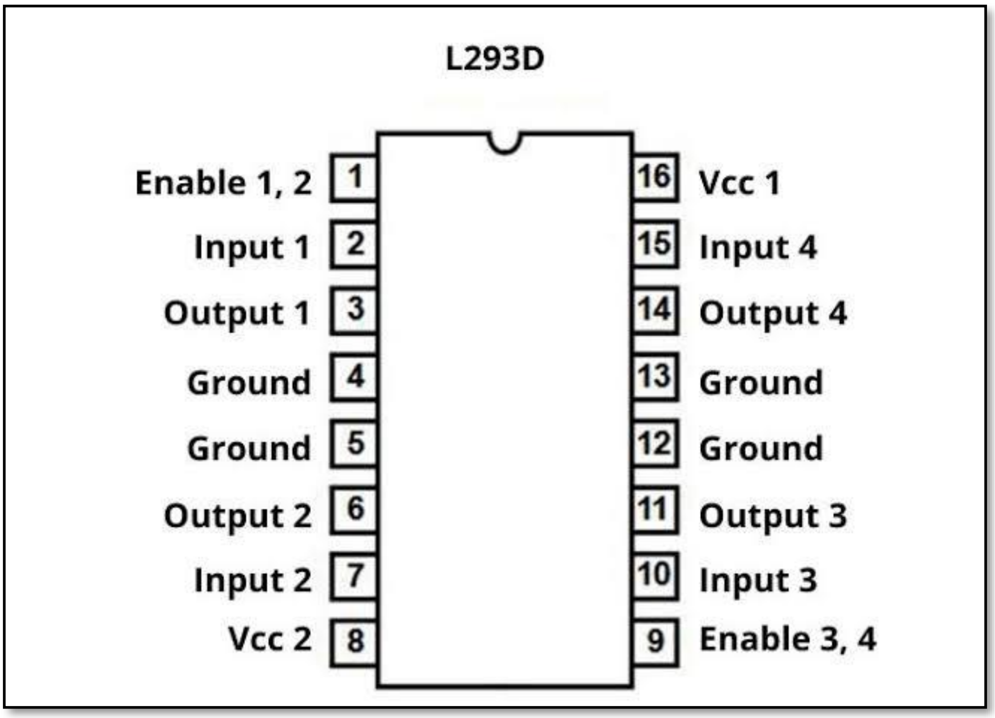


Figure 3.3 : LM293D Motor Driver IC

3.2 System Development :-

3.2.1 Microcontroller AT89C51 :

Features :

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory.
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description :

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configuration :

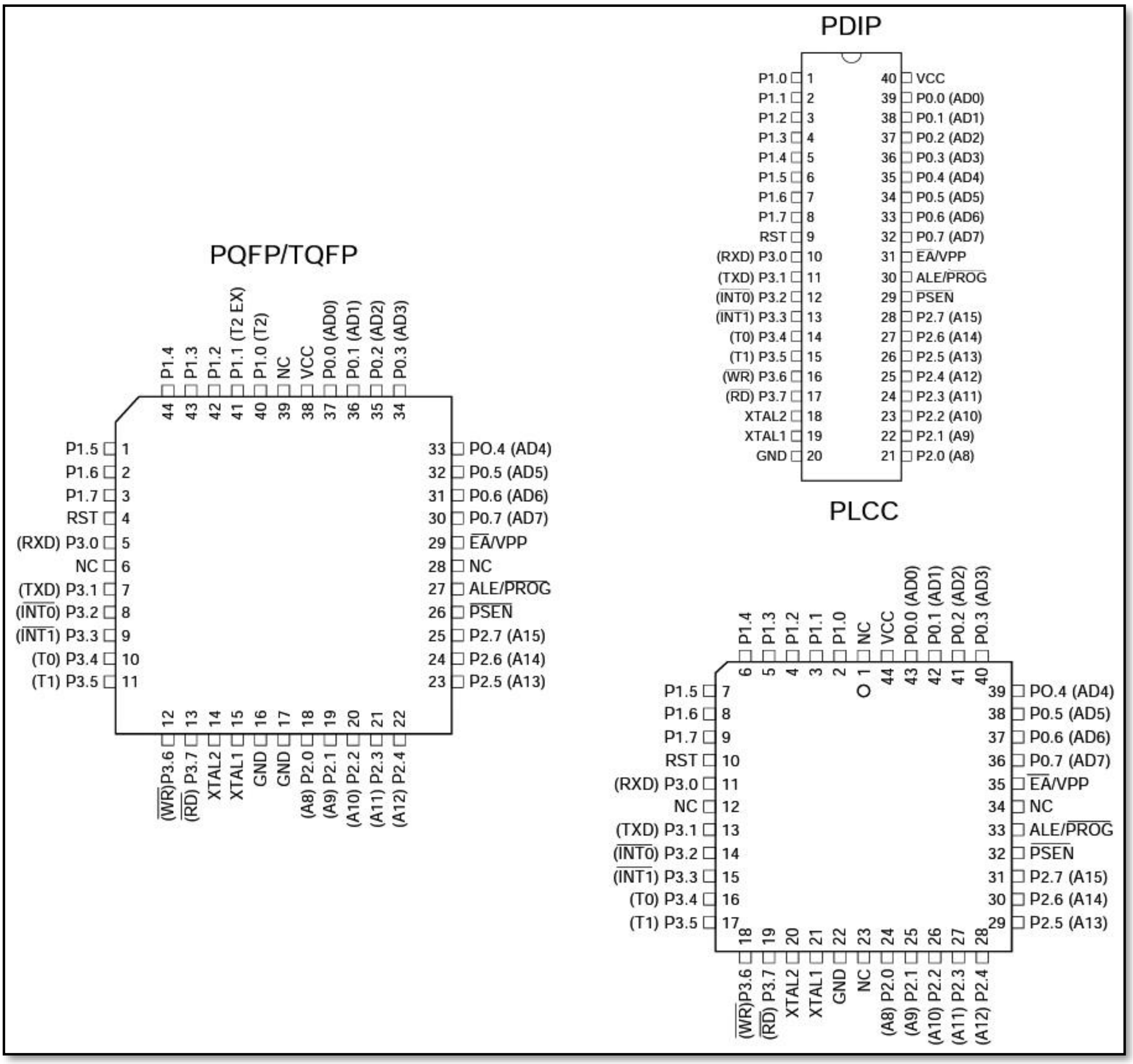


Figure 3.4 : Pin Configuration of AT89C51 Microcontroller

Block Diagram :

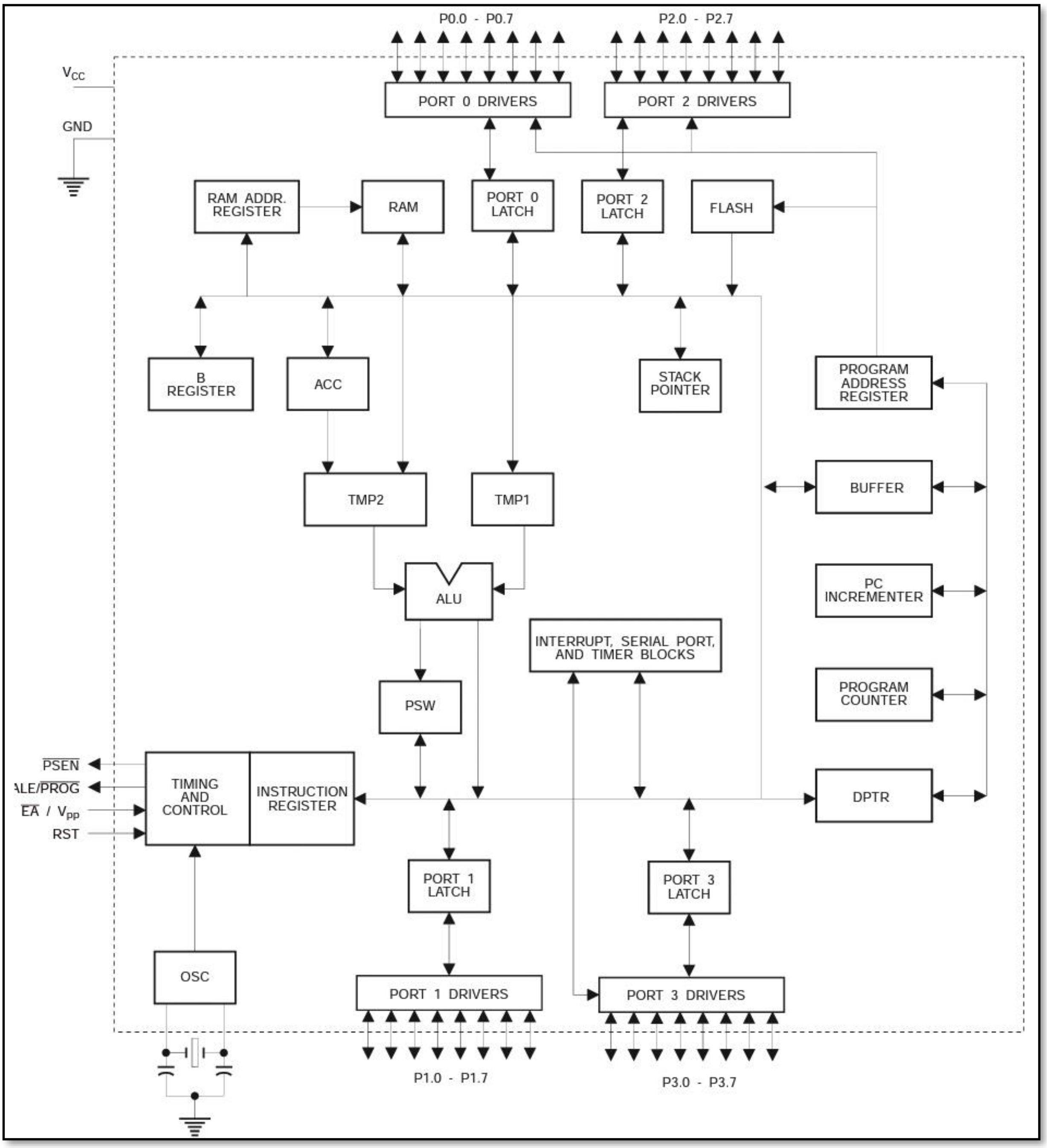


Figure 3.5 : Block Diagram of AT89C51 Microcontroller

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description :

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high impedance inputs.

Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pullups. Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Table 3.1 : Pin Description

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory. When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA /VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA internally latched on reset. EA will be should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming, for parts that require 12-volt VPP.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

3.2.2 Power Supply:

The power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an AC voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes. A rectifier is used for the converting a.c. supply into d.c supply. A bridge IC is used for that purpose. To get the power supply for the circuit , 230V ac main voltage is step down by the step-down transformer X1 which is rectified by the bridge rectifier into the DC voltage and is filtered out by the capacitor C1 of (100uF, 25V) rating. The filtered output fed to the input pin i.e. pin no.1 of the regulator IC 7805 and 7812. This gives the fixed 5V and 12V DC power Supply respectively. The block diagram of power supply is shown in fig below.

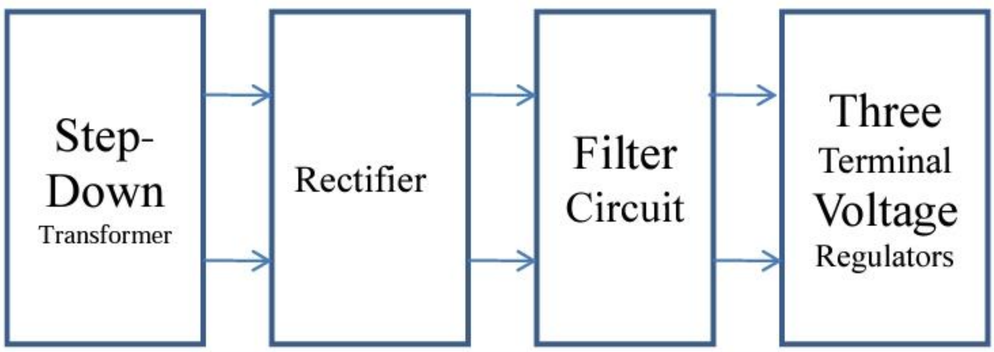


Figure 3.6 : Block Diagram of Power Supply

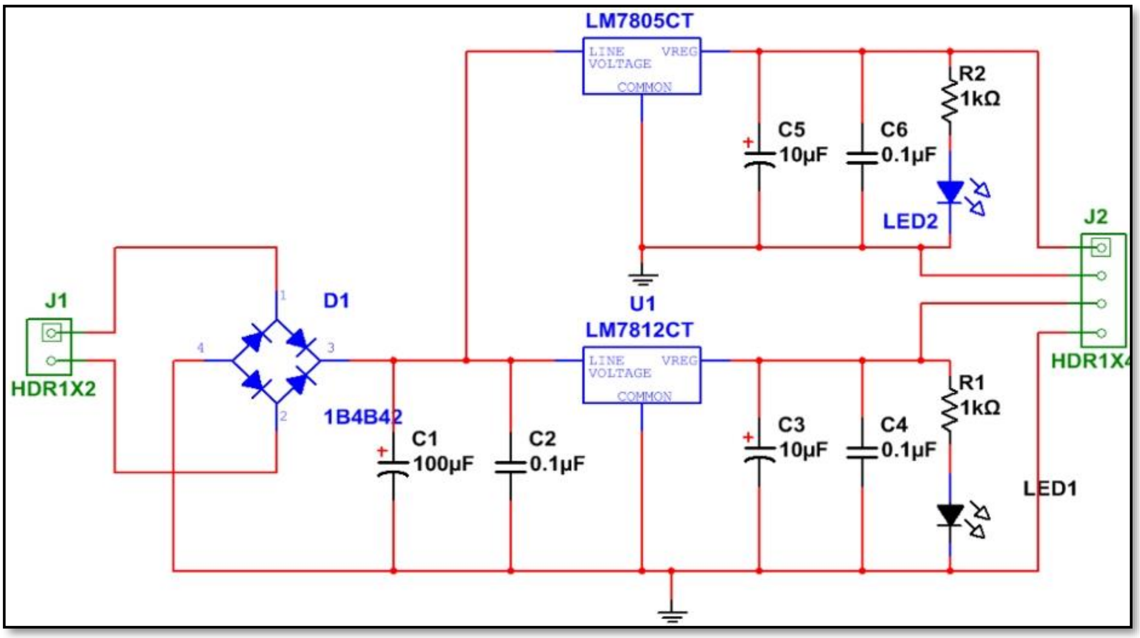


Figure 3.7 : Circuit Diagram of Power Supply

3.2.3 Crystal Oscillator (16 MHz)

The 16 MHz Crystal Oscillator module is designed to handle off-chip crystals that have a frequency of 416 MHz the crystal oscillator s output is fed to the System PLL as the input reference. The oscillator design generates low frequency and phase jitter, which is recommended for USB operation. This document contains an overview of the on-chip

Figure 3.8 : Quartz Crystal Oscillator

oscillator design and parameters for the crystal model, which are derived from both simulation and empirical data analysis. Recommendations and requirements for selecting a 16 MHz crystal are also covered. Finally, the document has guidelines and a detailed description of oscillator circuit design and PCB layout. It is strongly recommended that you follow the crystal specification and crystal PCB layout guidelines in this document.

An electronic circuit or electronic device that is used to generate periodically oscillating electronic signal is called as an electronic oscillator. The electronic signal produced by an oscillator is typically a sine wave or square wave. An electronic oscillator converts the direct current signal into an alternating current signal. The radio and television transmitters are broad casted using the signals generated by oscillators. The electronic beep sounds and video game sounds are generated by the oscillator signals. These oscillators generate signals using the principle of oscillation. There are different types of oscillator electronic circuits such as Linear oscillators – Hartley oscillator, Phase-shift oscillator, Armstrong oscillator, Clapp oscillator, Colpitts oscillator, and so on, Relaxation oscillators – Royer oscillator, Ring oscillator, Multivibrator, and so on, and Voltage Controlled Oscillator (VCO). In this article, let us discuss in detail about Crystal oscillator like what is crystal oscillator, a crystal oscillator circuit, working, and use of crystal oscillator in electronic circuits.

An electronic circuit that is used to generate an electrical signal of precise frequency by utilizing the vibrating crystal’s mechanical resonance made of piezoelectric material. There are different types of piezoelectric resonators, but typically, quartz crystal is used in these types of oscillators. Hence, these oscillator electronic circuits are named as crystal oscillators. The quartz crystal oscillator circuit diagram can be represented as follows:

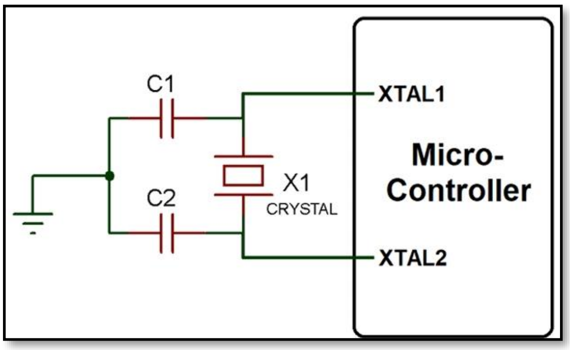


Figure 3.9 : Oscillator Circuit

These quartz crystal oscillators are highly stable, consists of good quality factor, they are small in size, and are very economical. Hence, quartz crystal oscillator circuits are superior compared to other resonators such as LC circuits, turning forks, and so on. Generally, 8MHz crystal oscillator is used in microprocessors and microcontrollers. The equivalent electrical circuit also represents the crystal action of the crystal. The basic components used in the circuit, inductance L1 represent crystal mass, capacitance C1 represents compliance, resistance R1 represents the crystal's internal structure friction, and C0 is used to represent the capacitance that is formed because of crystal's mechanical molding . The quartz crystal oscillator circuit diagram consists of series resonance and parallel resonance, i.e., two resonant frequencies. If the reactance produced by capacitance C1 is equal and opposite to the reactance produced by inductance L1, then the series resonance occurs.

3.2.4 Bridge Rectifier :

A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), current that flows in only one direction, a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components. The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification. In positive half cycle only two diodes (1 set of parallel diodes) will conduct, in negative half cycle remaining two diodes will conduct and they will conduct only in forward bias only. Bridge rectifier is used to maintain the proper DC polarity at the input to the circuit, irrespective of telephone line polarity. It comprises of four diodes connected to form a bridge. It uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in fig below.

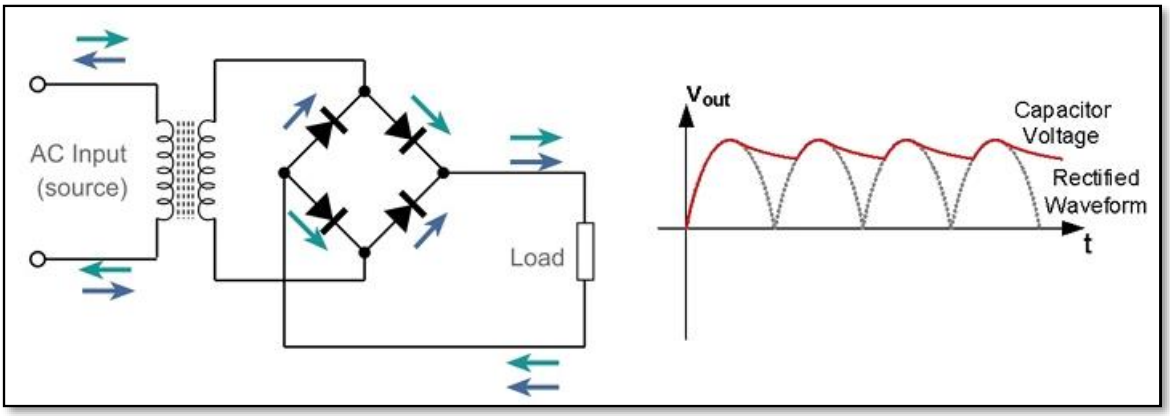


Figure 3.10 : Circuit Diagram of Rectifier

3.2.5 7805 Voltage Regulator :

Voltage regulators are very common in electronic circuits. They provide a constant output voltage for a varied input voltage. In our case the 7805 IC is an iconic regulator IC that finds its application in most of the projects. The name 7805 signifies two meaning, “78” means that it is a positive voltage regulator and “05” means that it provides 5V as output. So, our 7805 will provide a +5V output voltage. The output current of this IC can go up to 1.5A. But the IC suffers from heavy heat loss hence a Heat sink is recommended for projects that consume more current. For example, if the input voltage is 12V and you are consuming 1A, then (12-5) * 1 = 7W. This 7 Watts will be dissipated as heat. 7805 as +5V Voltage Regulator: This is a typical application circuit of the 7805 IC. We just need two capacitors of vale 33uf and 0.1uf to get this IC working. The input capacitor 0.33uF is a ceramic capacitor that deals with input inductance problem and the output capacitor 0.1uF is also a ceramic capacitor that adds to the stability of the circuit. These capacitors should be placed close to the terminals for them to work effectively. Also, they should be of ceramic type, since ceramic capacitors are faster than electrolytic. 7805 as adjustable output Regulator: This IC can also act as an adjustable output voltage regulator, meaning you can also control the output voltage for your desired value using the below circuit. Here, the input voltage can be anywhere between 9V-25V, and the output voltage can be adjusted using the value of resistance R1 and R2. The value can be calculated using the below formulae.

$$V_O = V_{xx} + \left(\frac{V_{xx}}{R1} + I_Q\right)R2$$

Where, Vxx=5, IQ = 5*10-3

LM78XX Technical Specifications

- TO-220 Package
- The Output current is 1.5 Ampere
- Immediate short circuit shutdown function
- Immediate over heat shutdown function
- Low price
- Authentic to use in commercial devices
- Accurate and fixed 12V output
- Maximum input voltage is 35V DC

Low standby current only 8mA

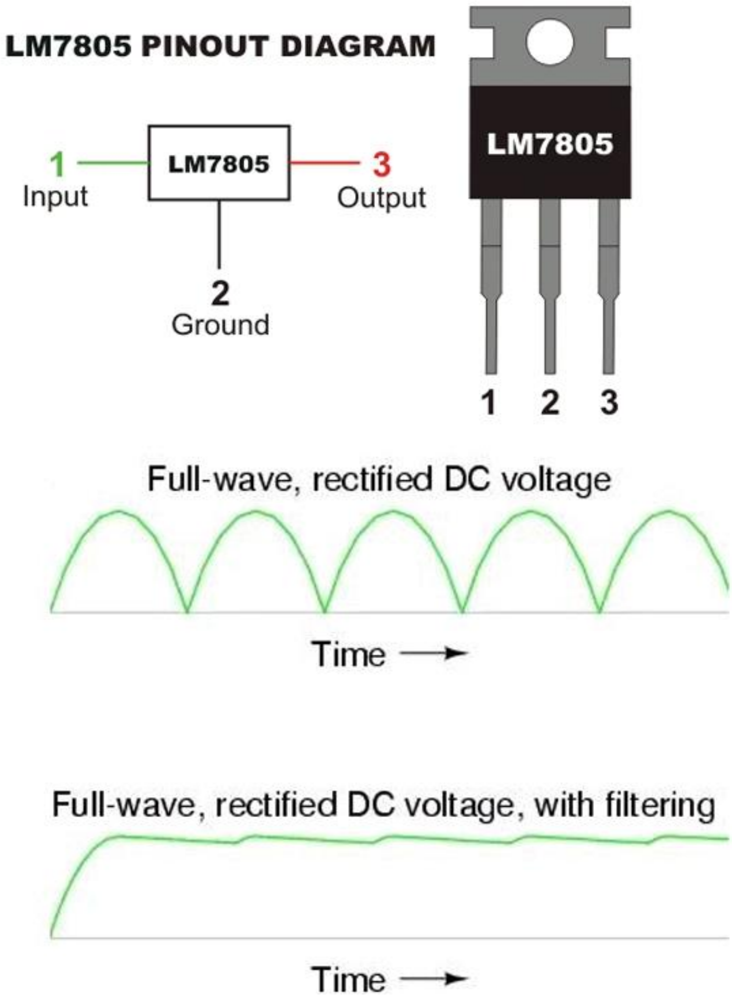


Figure 3.11 : Pinout and Output Waveform

3.2.6 Capacitor :

Power supplies, capacitors are used to smooth (filter) the pulsating DC output after rectification so that a nearly constant DC voltage is supplied to the load. The pulsating output of the rectifiers has an average DC value and an AC portion that is called ripple voltage. Filter capacitors reduce the amount of ripple voltage to a level that is acceptable. It should be noted that resistors and inductors can be combined with the capacitors to form filter networks. Here we will concentrate on capacitive filters only. In a filter circuit the capacitor is charged to the peak of the rectified input voltage during the positive portion of the input. When the input goes negative, the capacitor begins to discharge into the load. Smooth the output of the rectifier a reservoir capacitor is used - placed across the output of the reciter and in parallel with the load. The smoothing works because the capacitor charges up when the voltage from the rectifier rises above that of the capacitor and then as the rectifier voltage falls, the capacitor provides the required current from its stored charge. In this way the capacitor is able to provide charge when it is not available from the rectifier, and accordingly the voltage varies considerably less than if the capacitor were not present. The capacitor smoothing will not provide total voltage stability; there will always be some variation in the voltage. In fact, the higher the value of the capacitor, the greater the smoothing, and also the less current that is drawn, the better the smoothing.

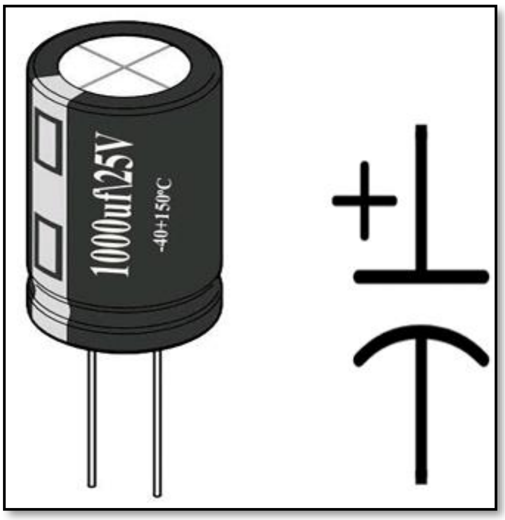


Figure 3.12 : Capacitor and Symbol

3.2.7 Resistor :

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits.

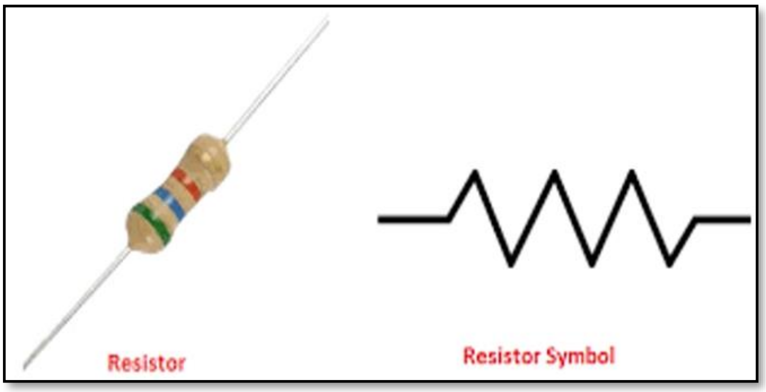
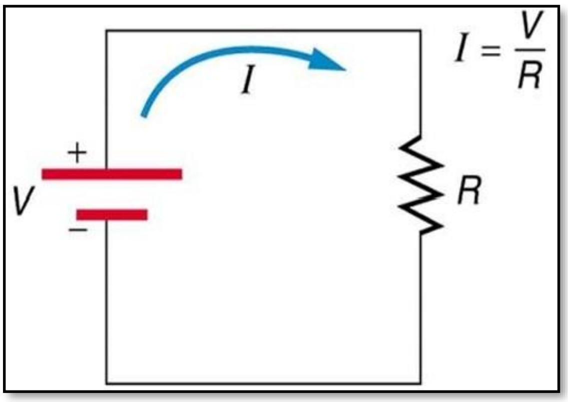


Figure 3.13 : Resistor and Symbol

The unit of resistance, given by 1Ω = 1 V/A



3.2.8 LED :

An LED is a two-lead semiconductor light source, which emits lights when activated. When an appropriate voltage is applied to the LED terminal, then the electrons are able to recombine with the electron holes within the device and release energy in the form of photons. This effect is known as electroluminescence. The color of the LED is determined by the energy band gap of the semiconductor.

The forward voltage requires to turn ON a LED, depends on the color of the LED. If you are feeding the exact value of forward voltage then you can connect a LED directly to the source. If the voltage is a higher than use a resistance in series with the LED, to calculate the value of the resistance use formula

$$R = (VS - VLED * X) / ILED$$

Were,

VS is the supply voltage

VLED is the forward voltage of

LED X is the number of LED connected in series

LED is LED current

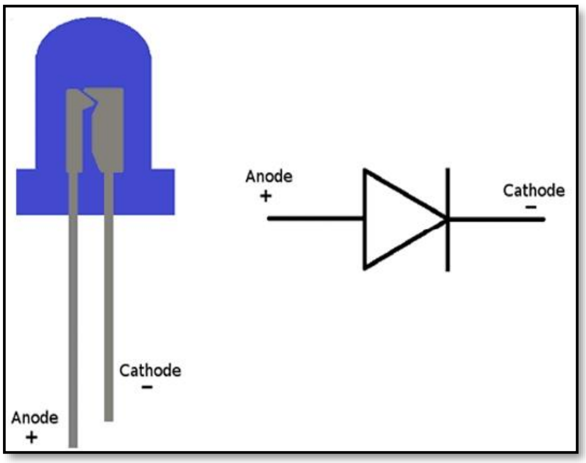


Figure 3.14 : LED and Symbol

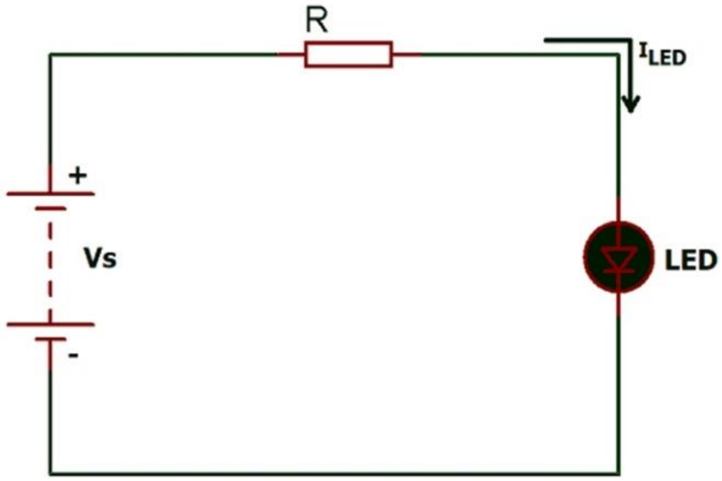


Figure 3.15 : LED Circuit Diagram

Features and Technical Specifications :

- Superior weather resistance
- 5mm Round Standard Directivity
- UV Resistant Eproxy

- Forward Current (IF): 30mA
- Forward Voltage (VF): 1.8V to 2.4V
- Reverse Voltage: 5V
- Operating Temperature: -30℃ to +85℃
- Storage Temperature: -40℃ to +100℃
- Luminous Intensity: 20mcd

3.2.9 Reset Circuit :

One task of the power-on reset (POR) is ensuring that the processor starts at a known address when power is first applied. To accomplish that task, the POR logic output holds the processor in its reset state when the processor's power supply is first turned on. The POR's second task is to keep the processor from starting its operation from that known address until three events have occurred: the system power supplies have stabilized at the appropriate levels; the processor's clock(s) has (have) settled; and the internal registers have been properly loaded. The POR accomplishes this second task through an onboard timer, which continues to hold the processor in its reset state for a prescribed period of time. That timer triggers after the processor's power supply reaches a specific voltage threshold. After a set time elapses, the timer expires, causing the POR output to become inactive, the processor's data sheet specifies the required duration of the timer's delay. The timer, incidentally, is what differentiates a POR from a voltage detector, a device that also detects a voltage threshold, but does not time an event.

During powering-on the 328P's reset pin is low until the capacitor C5 is sufficiently charged. How long this takes depends mainly on the start-up behavior of Vcc. When DTR is low the capacitor will charge faster because RN2D (from capacitor C5 to ground) is bypassed. But the main thing is that the power-on reset and the external reset have the same effect when the rising voltage passes a certain threshold a timer is started and after it times out program execution starts at 0x0000 or at the Boot Reset Address .So, in the powering-on situation it makes no sense to differentiate between Power on Reset and External Reset.

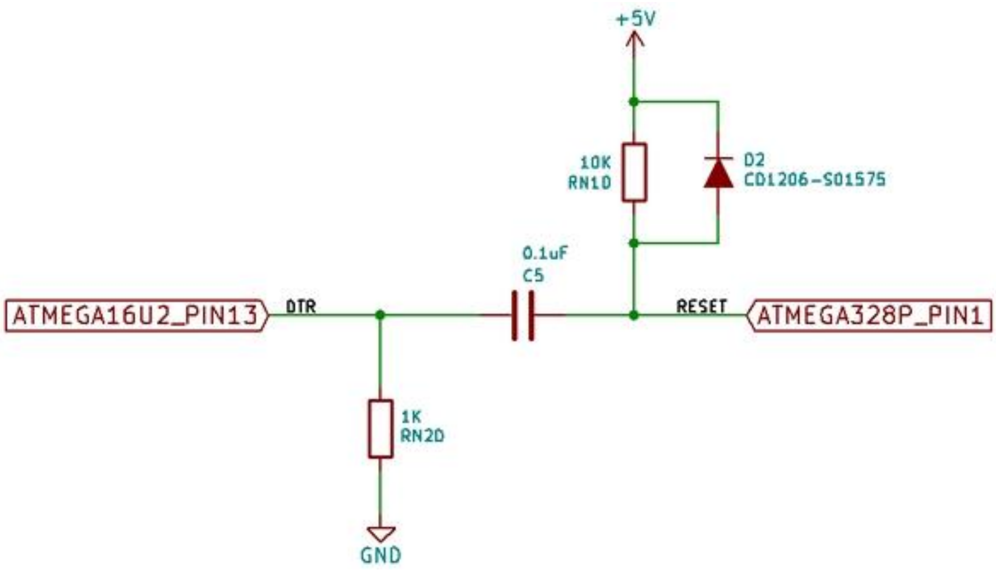


Figure 3.16 : Reset Circuit

3.2.10 Switches :

Push Buttons are normally-open tactile switches. Push buttons allow us to power the circuit or make any particular connection only when we press the button. Simply, it makes the circuit connected when pressed and breaks when released. A push button is also used for triggering of the SCR by gate terminal. These are the most common buttons which we see in our daily life electronic equipment's. Some of the applications of the Push button are mentioned at the end of the article. When connecting in between of supply and the circuit we should only connect the wires with both the legs of the Push-Button as shown in the circuit below:

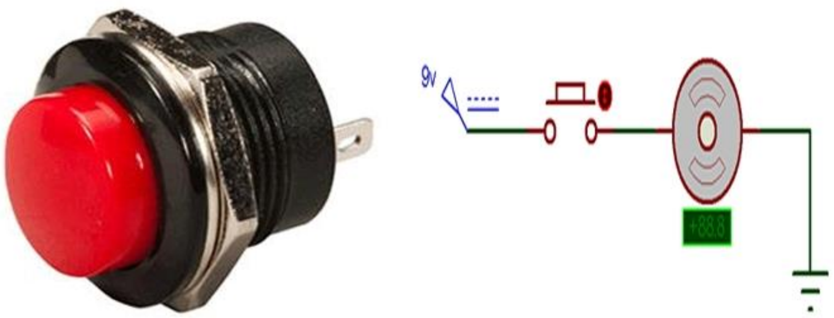


Figure 3.17 : Switch

A Push-Button can also be used for the triggering purpose like of SCR. An SCR is a gate-controlled Switch which needs a triggering pulse. So, for this we can add a Push button in the circuit to give a triggering pulse, a push button switch is a small, sealed mechanism that completes an electric circuit when you press on it. When it's on, a small metal spring inside makes contact with two wires, allowing electricity to flow. The body of the switch is made of non-conducting plastic.

CHAPTER 4

EXPERIMENTATION, ANALYSIS & DISCUSSION

4.1 System Specifications & Estimation:

4.1.1 System Implementation:

The implementation of this project work requires two software’s.

These are:

1. Kiel (µVision)

2. Proteus

4.1.1.1 Kiel (µVision):

The main working of Kiel compiler is to convert the high-level language into the Hex code.

Step 1: Create a Project

Create a New Project for the Evaluation Board

Create a project with initialization files and the main module:

1. In the main µVision menu, select Project ,New µVision Project. The Create New Project window opens up.

2. Create a suitable folder in the normal fashion and name your project. We will use C:\USB and the project name will be USB. When you save the project the project file name will be USB.uvprojx.

3. The Select Device for Target window opens. Select STM32F429ZITx:

4. Click on OK and the Manage Run-Time Environment window opens:

5. Expand the various options as shown and select CMSIS:Core, Device:Startup. Most devices provide additional hardware abstraction layers that are listed under the Device component. The STM32Cube HAL is a list of available drivers for the STM32F429. It requires a framework.

Select STM32Cube Framework (API):Classic. For more information, click on the link STM32Cube Framework which opens the documentation (red circle).

6. In the SEL. Column will be some orange blocks. Click on Resolve and these will turn to green.

7. Click OK to close this window

8. In the Project window expand all the items and have a look at the files µVision has added to your project:

Add the main.c file:

1. Right click on Source Group 1 and select Add New Item to Group

'Source Group 1'…

2. In the window that opens, select User Code Template. Select 'main' module for STM32Cube. It initializes the STM32Cube HAL and configures the clock system.

3. Click on Add

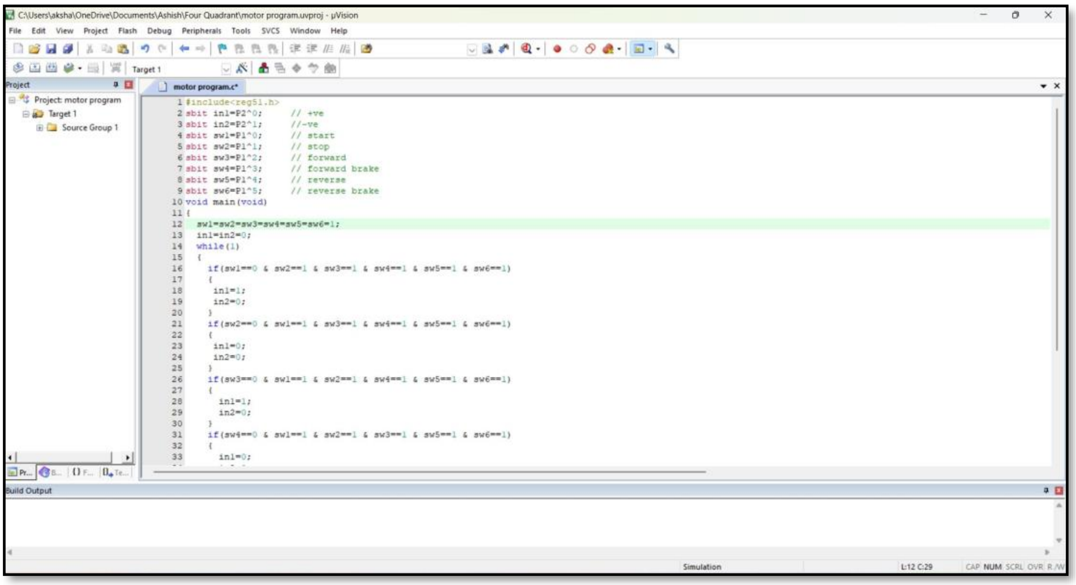


Figure 4.1 : Keil Software

Set the CPU Clock Speed:

The external crystal oscillator on the development kit has a frequency of 8 MHz.

1. Select Target Options or ALT-F7 and select the Target tab. Enter 8 MHz in the Xtal (MHz) box.

2. Select the C/C++ tab.

3. Enter HSE_VALUE=8000000 in the Define box. The HSE_VALUE represents the crystal frequency. This will set the CPU clock to 168 MHz in system_stmf4xx.c.

4. Click on OK to close this window.

5. Select File Save All or press

6. Compile the project source files: There will be no errors or warnings displayed in the Build Output window. If you get any errors or warnings, please correct this before moving on to configure the ST-Link V2 Debug Adapter.

4.1.1.2 Proteus 8 Professionals :

Proteus is a complete development platform from product concept to design completion. Its advantages are intelligent principal layout, hybrid circuit simulation and accurate analysis, single-chip software debugging, single-chip and peripheral circuit co-simulation, PCB automatic layout and wiring.

Lab center, a British company and Proteus software developer, has been developed around the world for nearly 20 years. It is currently the most powerful and cost-effective EDA tool in more than 50 countries. It has been named the best EDA tool by EWW CAD REVIEW ROUNDUP [2]. It is one step ahead of other competitors in philosophy, continuous model development and software upgrade thus to ensure first-class technology.

Proteus software product structure as shown in the following figure 1, Proteus is a complete embedded system software and hardware design simulation platform, Proteus ISIS is an intelligent schematic input system, system design and Simulation of the basic platform to achieve the combination of single-chip microcomputer simulation and spice circuit simulation. It has the functions of analog circuit simulation, digital circuit simulation, system simulation composed of single chip microcomputer and its peripheral circuit, RS232 dynamic simulation, I2C debugger, SPI debugger, keyboard and LCD system simulation, and various virtual instruments, such as oscilloscope, logic analyzer, signal generator, etc. ARES is a high-level PCB wiring editing software. The schematic diagram designed in ISIS can automatically export the network table after confirming that the device is packaged correctly. PCB layout and wiring can use 2D tools to design the PCB frame in the board Edge board side layer, set the wiring strategy, select the automatic or artificial device layout for wiring, and carry out DRC. (Design Rules Check) and ERC (Electrical Rules Check) can output Gerber files in layers for PCB boarding.

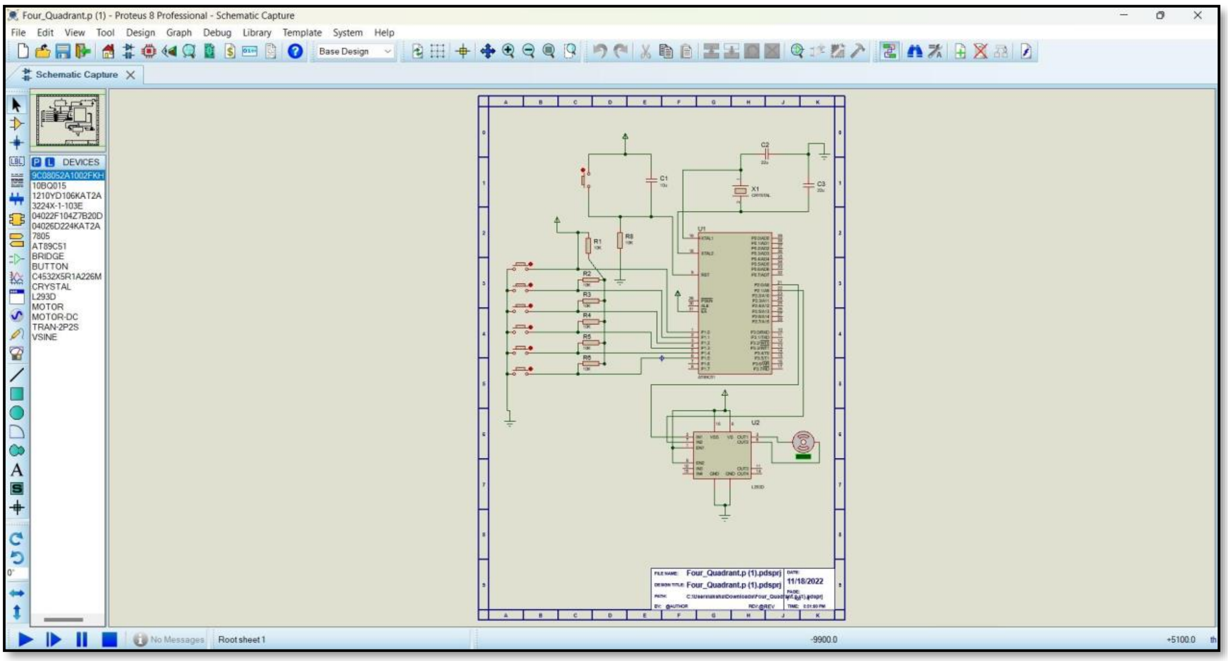


Figure 4.2 : Proteus Software

Proteus virtual system model combines SPICE circuit simulation of mixed mode, dynamic device and microcontroller model to realize the complete collaborative simulation based on microcontroller design. For the first time, it is possible to develop and test such designs before the physical prototype comes out. Proteus software products include Proteus VSM, VSM for ARM7/LPC2XXX, VSM for 51/52, VSM for AVR, VSM for PIC24, Proteus PCB Design, Advanced Simulation Feature (ASF). Proteus virtual system model combines SPICE circuit simulation of mixed mode, dynamic device and microcontroller model to realize the complete collaborative simulation based on microcontroller design. For the first time, it is possible to develop and test such designs before the physical prototype comes out. Proteus software products include Proteus VSM, VSM for ARM7/LPC2XXX, VSM for 51/52, VSM for AVR, VSM for PIC24, Proteus PCB Design, Advanced Simulation Feature (ASF).

4.2 Circuit Diagram :

The response of the motor connected can be seen visually according to the program fed into the microcontroller and the operations are carried accordingly. It is the easiest way to check whether the hardware will get the desired output. The changes can be made to get the desired output and the operation can be carried out accordingly.

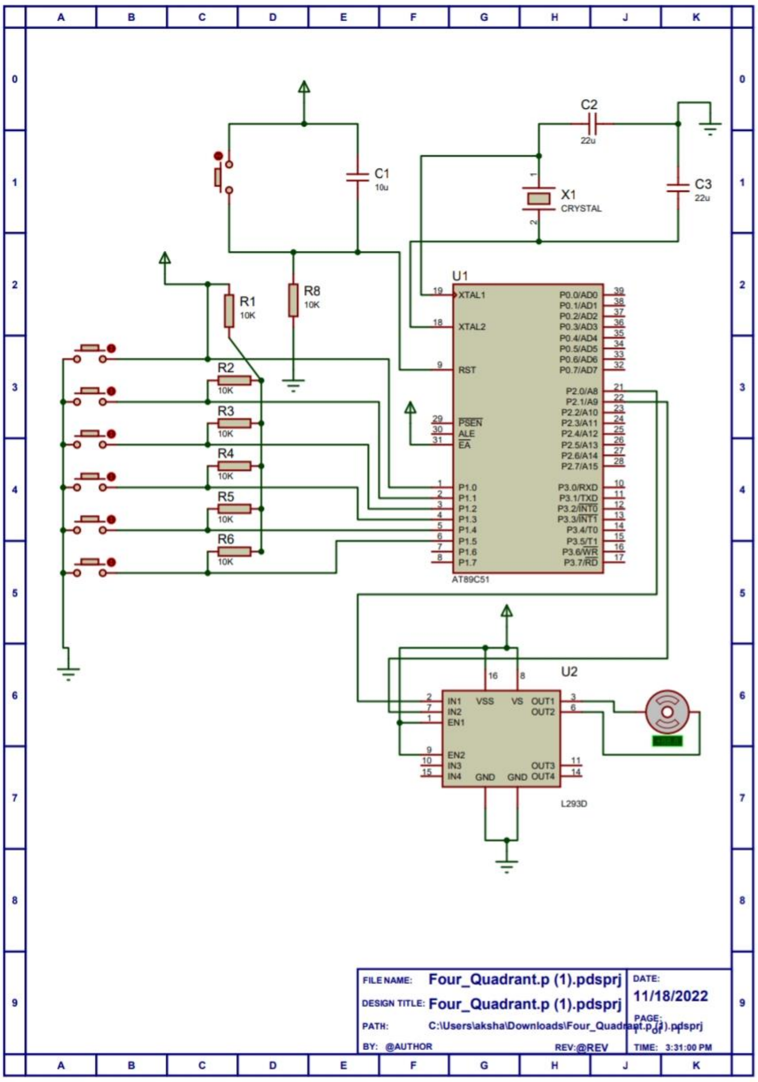


Figure 4.3 : Circuit Diagram using Proteus Software

4.3 Program :

```
#include<reg51.h>
sbit in1=P2^0;    // +ve
sbit in2=P2^1;    //-ve
sbit sw1=P1^0;    // start
sbit sw2=P1^1;    // stop
sbit sw3=P1^2;    // forward
sbit sw4=P1^3;    // forward brake
sbit sw5=P1^4;    // reverse
sbit sw6=P1^5;    // reverse brake
void main(void)
{
        sw1=sw2=sw3=sw4=sw5=sw6=1;
        in1=in2=0;
        while(1)
        {
                if(sw1==0 & sw2==1 & sw3==1 & sw4==1 & sw5==1 & sw6==1)
                {
                 in1=1;
                 in2=0;
                }
                if(sw2==0 & sw1==1 & sw3==1 & sw4==1 & sw5==1 & sw6==1)
                {
          in1=0;
   in2=0;
                }
```

```
                if(sw3==0 & sw1==1 & sw2==1 & sw4==1 & sw5==1 & sw6==1)
                {
                        in1=1;
                        in2=0;
                }
                if(sw4==0 & sw1==1 & sw2==1 & sw3==1 & sw5==1 & sw6==1)
                {
                        in1=0;
                        in2=0;
                }
                if(sw5==0 & sw1==1 & sw2==1 & sw3==1 & sw4==1 & sw6==1)
                {
                        in1=0;
                        in2=1;
                }
                if(sw6==0 & sw1==1 & sw2==1 & sw3==1 & sw4==1 & sw5==1)
                {
                        in1=0;
                        in2=0;
                }
        }
}
```

4.4 Hardware Description:

This circuit is used for the four quadrant DC motor speed control operation. Here seven switches are interfaced to microcontroller to control the speed of the motor.

When the starting switch is pressed the motor starts rotating at full speed being driven by a motor driver IC L293D that receives control signal continuously from the microcontroller.[1] When clockwise switch is pressed the motor rotates in the forward direction as per the logic provided by the program from the microcontroller to the motor driver IC.[1] While forward brake is pressed a reverse voltage is applied to the motor by the

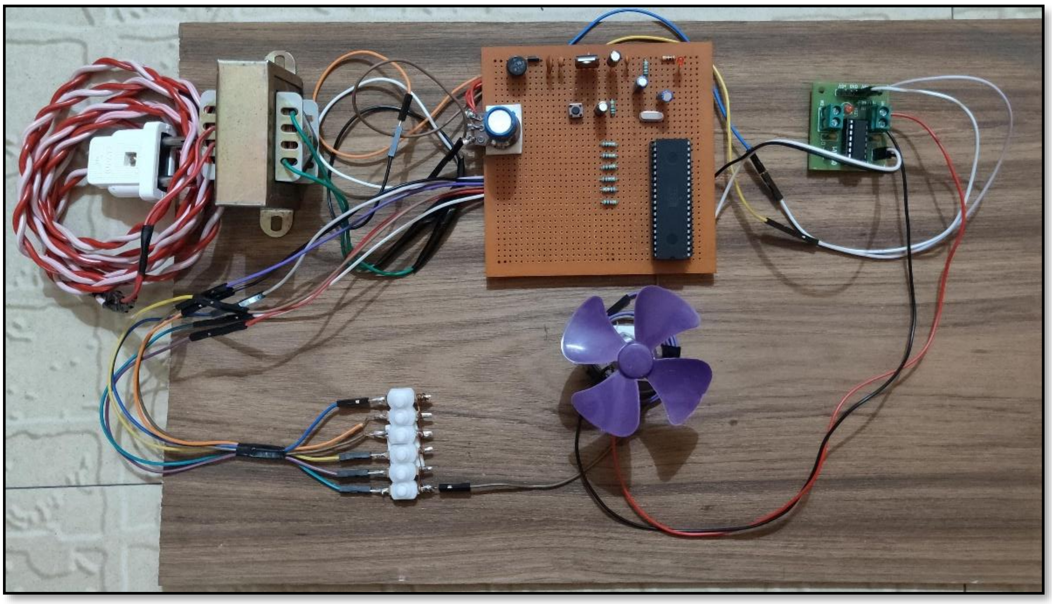


Figure 4.4: Assembly of Hardware Components

motor driver IC by sensing reverse logic sent by the microcontroller for a short time period due to and reverse brake switch is pressed the microcontroller delivers a logic to the motor driver IC that develops for very small time a reverse voltage across the running motor due to which instantaneous brake situation happens to the motor.[1] PWM switch is used to rotate the motor at varying speeds by delivering from the microcontroller a varying duty cycle to the enable pin of the motor driver IC.[1] It starts from 100% duty cycle and reduces in steps of 10% when it is pressed again and finally reaches to 10% duty cycle and the process repeats.[1] Stop button is used to switch OFF the motor by driving the enable pin to ground from the microcontroller command accordingly.

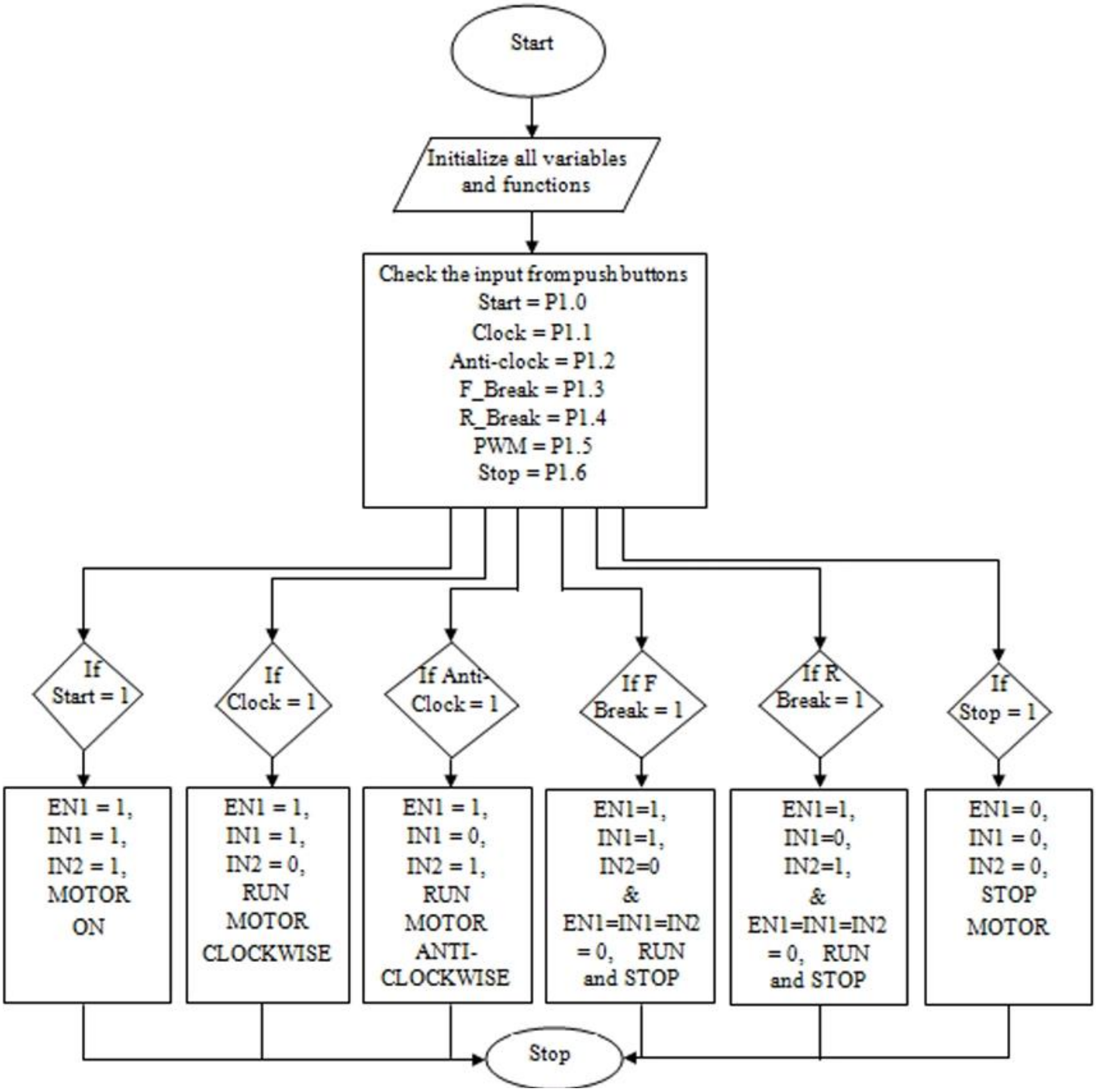


Figure 4.5: Flow Chart Diagram of System

4.5 Result:

The practical implementation of the four-quadrant control of the DC motor is shown in figure 4.4. The hardware is designed and the operation has been done based upon the program written in the AT89C51 microcontroller for the four-quadrant operation of the DC motor and the speed is also controlled by using PWM technique which instantaneous brake situation is applied to the motor. The project model is supplied with 230V ac supply from the circuit. The 230V supply is given to the primary of the transformer which gives 12V output. This 12V ac supply fed to bridge rectifier, which converts it into 12V dc. This 12V dc is given to voltage regulator which provides 5V dc.

CHAPTER 5

CONCLUSION

The prototype hardware model for the four-quadrant dc motor speed control using microcontroller is designed. A simulated model has been developed by proteus software and then result has been verified using a prototype hardware model. In the proposed model, we can use the PWM technique for controlling the speed of dc motor. As the on-time duty cycle increases the speed of dc motor also increases. The waveform of input pulse of dc motor has been taken for forward and reverse braking mode and it has been observed that amplitude of waveform became high for very short duration and after that amplitude becomes zero. In the experimental result it has been observed that some harmonics are occurred. It is due to different nonlinear electronic components such as diodes, transistors etc. present in the prototype developed model. It has been observed that amplitude of waveform obtained in proteus simulated model is 5v while it is 3.5v for waveform obtained from experimental result i.e., amplitude of waveform obtained from experimental result is less. It is due to the fact that some voltage drop has been taken place across resisters used in the prototype developed model. It is practical and highly feasible in economic point of view and has an advantage of running motors of higher ratings. It gives a reliable, durable, accurate and efficient way of speed control of a DC motor. The program is found to be efficient and the results with the designed hardware are promising. The developed control and power circuit functions properly and satisfies the application requirements. The motor is able to operate in all the four quadrants successfully. Regenerative braking is also achieved.

CHAPTER 6

REFERENCES

1. B.K Bose., Power electronics and motor drives recent technology advances, Proceedings of the IEEE International Symposium on Industrial Electronics, IEEE, 2002, pp 22-25.

2. "DS1103 PPC Controller Board", Germany: dSPACE, July 2008. Janice Gillispie Mazidi. "Books on Microcontroller: 8051 microcontroller and embedded systems"

3. Maiocchi.G., "Books on DC motors: Driving DC Motors "BL. Theraja. "DC Motors and drives "

4. Muhammad Ali Mazidi and Janice Gillispie Mazidi, "The 8051 Microcontroller and Embedded Systems, Pearson Prentice Hall Publication".

5. S.M.Rangdal1, Prof. G.P.Jain2 "Speed Control of Dc Motor Using Microcontroller" International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No.02, Issue No. 12, December 2014 ISSN (online): 2348 – 7550.

6. Y. S. E. Ali, S. B. M. Noor, S. M. Uashi and M. K Hassan" Microcontroller Performance for DC Motor Speed Control" O-7803-8208,2003 IEEE.

7. https://nevonprojects.com

8. https://www.wikipedia.com