A

PROJECT REPORT

ON

**AUTOMATED RAILWAY GATE CONTROL**

BY

USING PIC16F877A MICROCONTROLLER

Submitted to Faculty of Engineering

Dr. A.P.J Abdul Kalam Technical University (AKTU)

In partial fulfilment of the requirement for award of degree of

Bachelor of Technology

In

Electronics and Communication Engineering

**SUBMITTED BY: -**

AJAY KUMAR

Roll No. (1742931001)

B. TECH (7th sem/4th Year)

**SUBMITTED TO: -**

Prof. Mr. Ashish Nigam

(Project Head)

(H.O.D)

****

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NARAINA VIDYA PEETH ENGINEERING AND MANAGEMENT INSTITUTE, GANGAGUNJ PANKI KANPUR - 429

**ACKNOWLEDGEMENT**

During this project work, we would like to project report on the title-**AUTOMATIC RAILWAY GATE CONTROL SYSTEM**, We acknowledge with sincere thanks to our project guide Mr. Ashish Nigam for excellent guidance and selfless efforts. Without their co-operative attitude, constant inspiration and dedicated at each and every stage of this project it would not possible to make this project complete. We would also like to express our gratitude and thanks toward the staff and H.O.D. of Electronics & Communication Engineering department for their supper. .

**Abstract**

In this project to lift the railway crossing gate D.C. series motor. Gear arrangement is used. The Crystal oscillator at two points on away from railway crossing gate is used. A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a constant frequency. This frequency is often used to keep track of time, as in quartz wristwatches, to provide a stable clock signal for digital integrated circuits. To send response on micro controller then micro controller

which received a data and complete the circuit when railway will pass through it and the gate will be closed and similarly when the rail will pass through the gate then after some delay open the gate. Hence the motor will operate and with help of gear, and the gate will open. In this way the automatic operation of gate takes place.

**Contents**

* **Introduction**
* **Objective**
* **Components Used**
* **Circuit Diagram**
* **Software Used**
* **Description of Components**
* **Introduction of Proteus 8 Software**
* **Introduction of Keil uVision Software**
* **PIC16F877A microcontroller**
* **L293D Motor Driver Introduction**
* **Crystal oscillator**
* **Servo Motor**
* **APPENDIX A.: PROGRAMMING**

**Introduction**

* In this project to lift the railway crossing gate D.C. series motor. Gear arrangement is used. The Crystal oscillator at two points on away from railway crossing gate is used. A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a constant frequency. This frequency is often used to keep track of time, as in quartz wristwatches, to provide a stable clock signal for digital integrated circuits. To send response on micro controller then micro controller
* which received a data and complete the circuit when railway will pass through it and the gate will be closed and similarly when the rail will pass through the gate then after some delay open the gate. Hence the motor will operate and with help of gear, and the gate will open. In this way the automatic operation of gate takes place.

#### The objective of this project is to create an automatic railway gate control system which can be implemented easily in roads. Generally there are manual gate control system which are maintained by person. As vehicles are increasing day by day it has become more difficult to control the gate manually. As a result often accident occurs and many people become injured badly and sometimes it become very serious when people died due to this type of accidents. This project can help us to reduce accidents in our country by introducing automatic railway gate control system

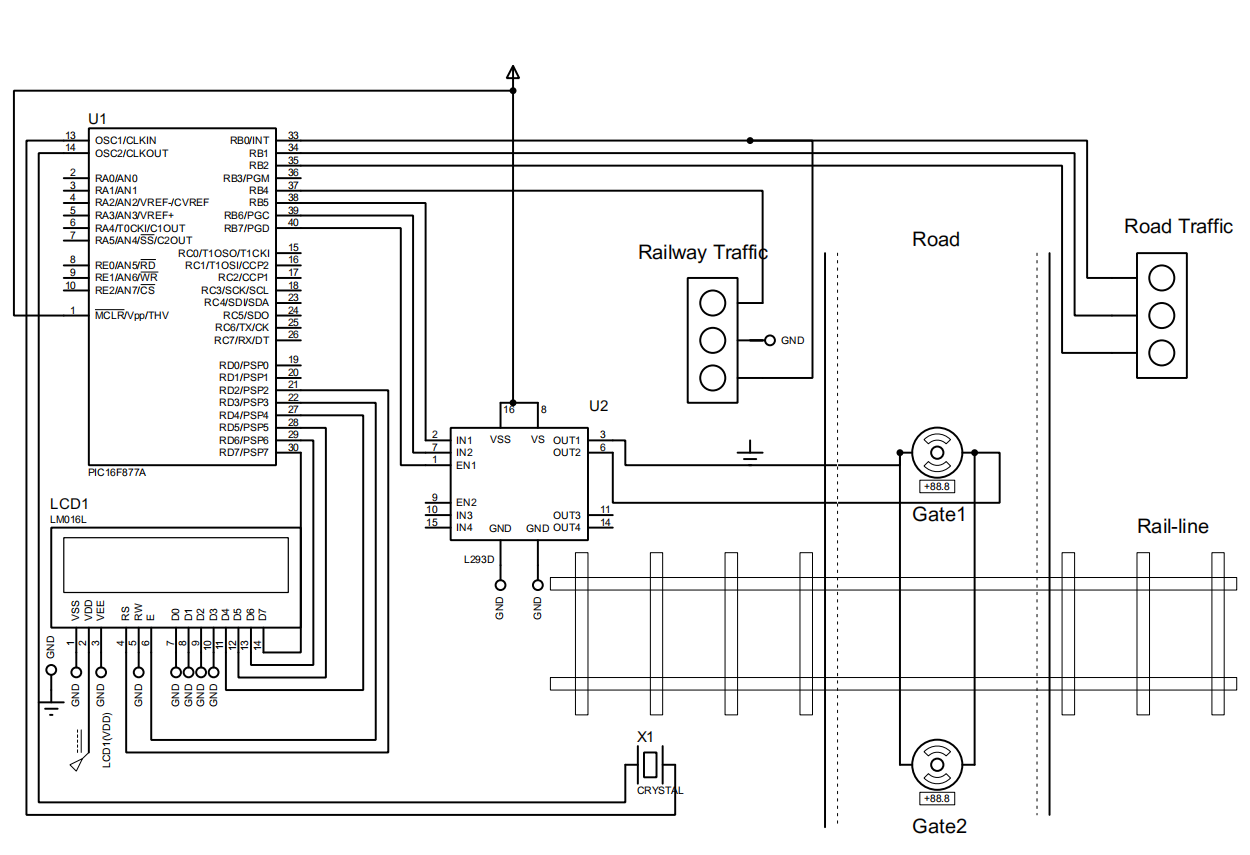
## Objective

#### The objective of this project is to create an automatic railway gate control system which can be implemented easily in roads. Generally there are manual gate control system which are maintained by person. As vehicles are increasing day by day it has become more difficult to control the gate manually. As a result often accident occurs and many people become injured badly and sometimes it become very serious when people died due to this type of accidents. This project can help us to reduce accidents in our country by introducing automatic railway gate control system

## Components Used

* + PIC16F877A Micro-controller
  + Servo Motor
  + Crystal oscillator
  + L293D IC , Dual H-Bridge Motor Driver
  + Resistances
  + LM016L LCD Display
  + Switch

**Circuit Diagram**



**Software Used In Projects**

* Proteus 8 Professional
* Keil uVision

**Introduction of Proteus 8 Professional**

⮚ **Proteus Design Suite** (designed by **Labcenter Electronics Ltd.**) is a software tool set,

mainly used for creating schematics, simulating Electronics & Embedded Circuits and

designing PCB Layouts.

⮚ Proteus ISIS is used by Engineering students & professionals to create schematics &

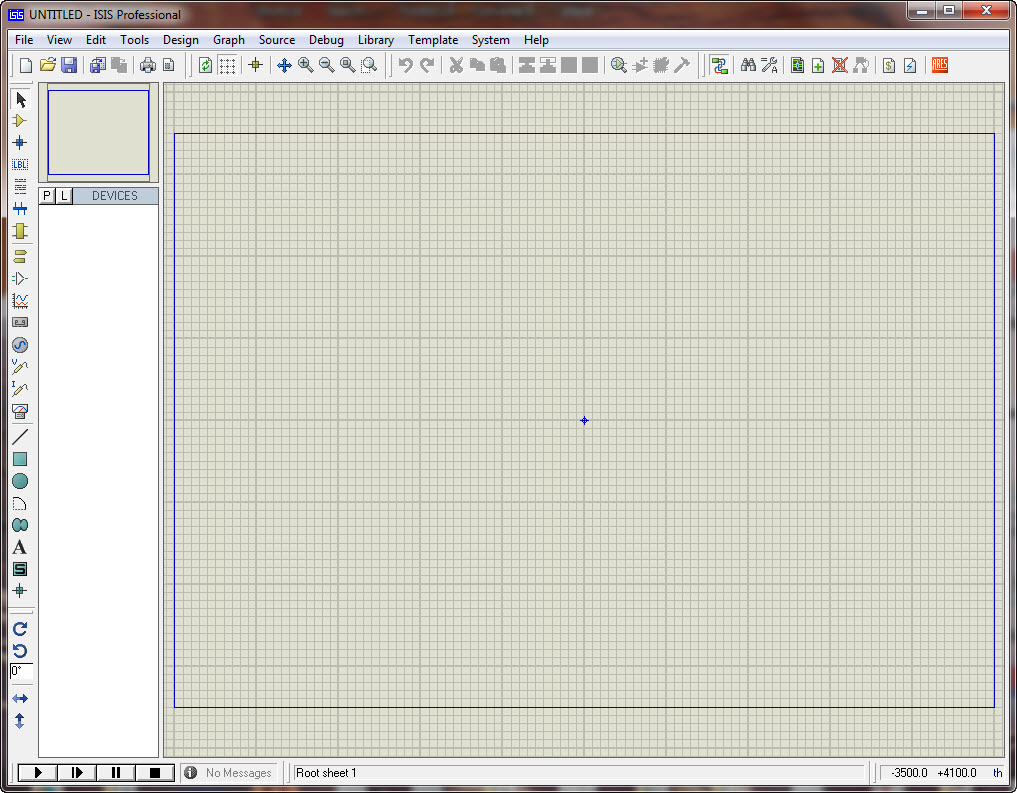
simulations of different electronic circuits.

⮚ Proteus ARES is used for designing PCB Layouts of electronic circuits.It's available in four

languages i.e. English, Chinese, Spanish & French.

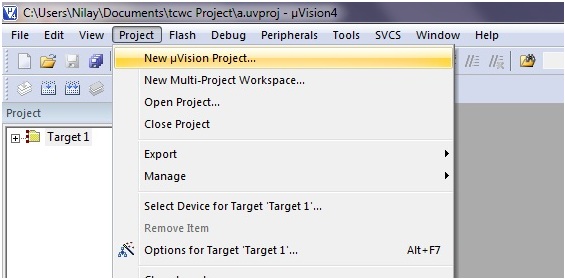
⮚ Proteus is also used for designing/testing programming codes for different

Microcontrollers i.e. Arduino, PIC Microcontroller, 8051 etc.



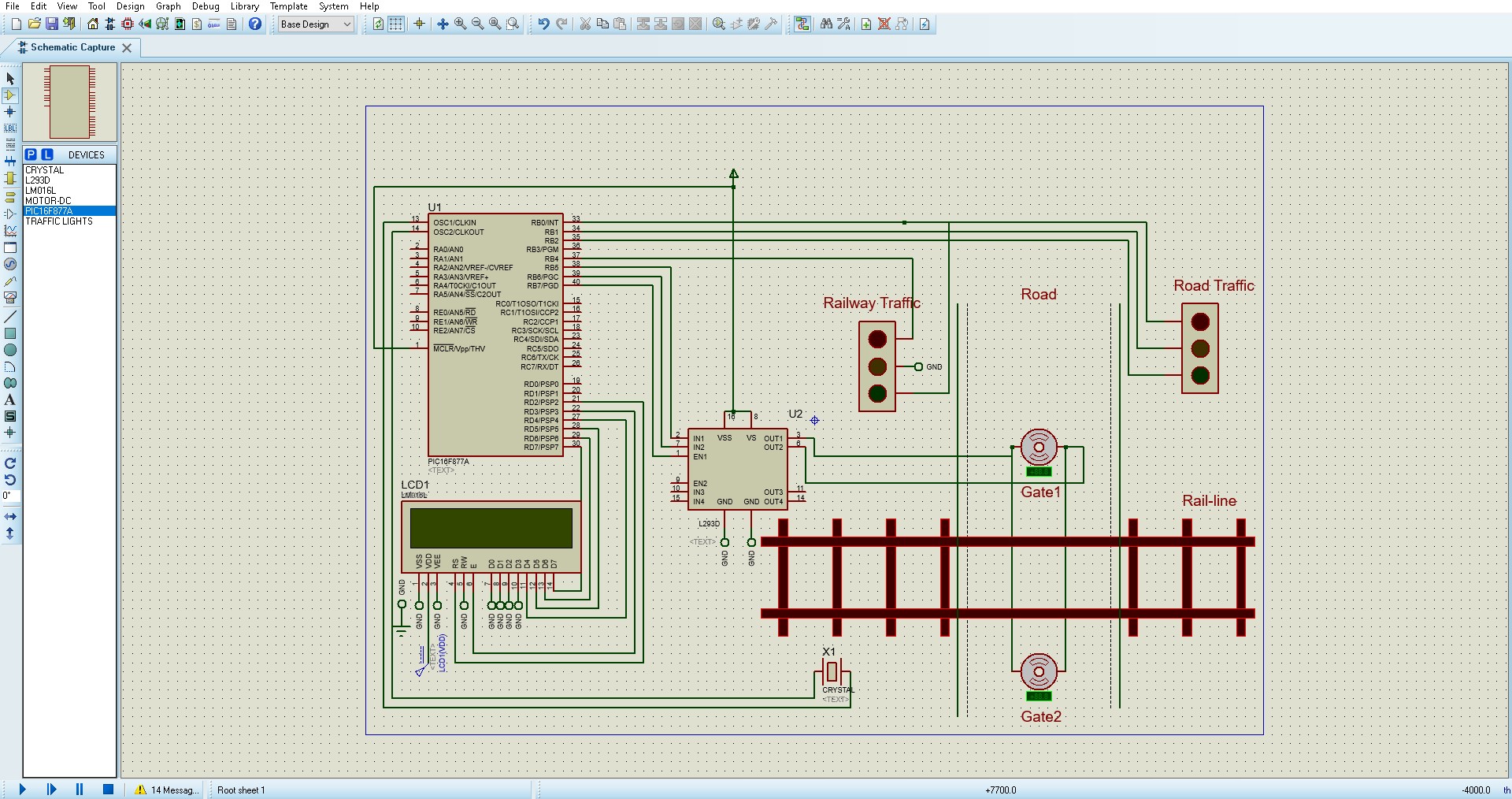
**Introduction of Keil uVision**

* The C programming language is a general-purpose programming language that provides code efficiency, elements of structured programming, and a rich set of operators. C is not a *big* language and is not designed for any one particular area of application. Its generality combined with its absence of restrictions, makes C a convenient and effective programming solution for a wide variety of software tasks. Many applications can be solved more easily and efficiently with C than with other more specialized languages.
* The Cx51 Optimizing C Compiler is a complete implementation of the American National Standards Institute (ANSI) standard for the C language. The Cx51 Compiler is not a universal C compiler adapted for the 8051 target. It is a ground-up implementation, dedicated to generating extremely fast and compact code for the 8051 microprocessor. The Cx51 Compiler provides you with the flexibility of programming in C and the code efficiency and speed of assembly language.
* The C language on its own is not capable of performing operations (such as input and output) that would normally require intervention from the operating system. Instead, these capabilities are provided as part of the standard library. Because these functions are separate from the language itself, C is especially suited for producing code that is portable across a wide number of platforms

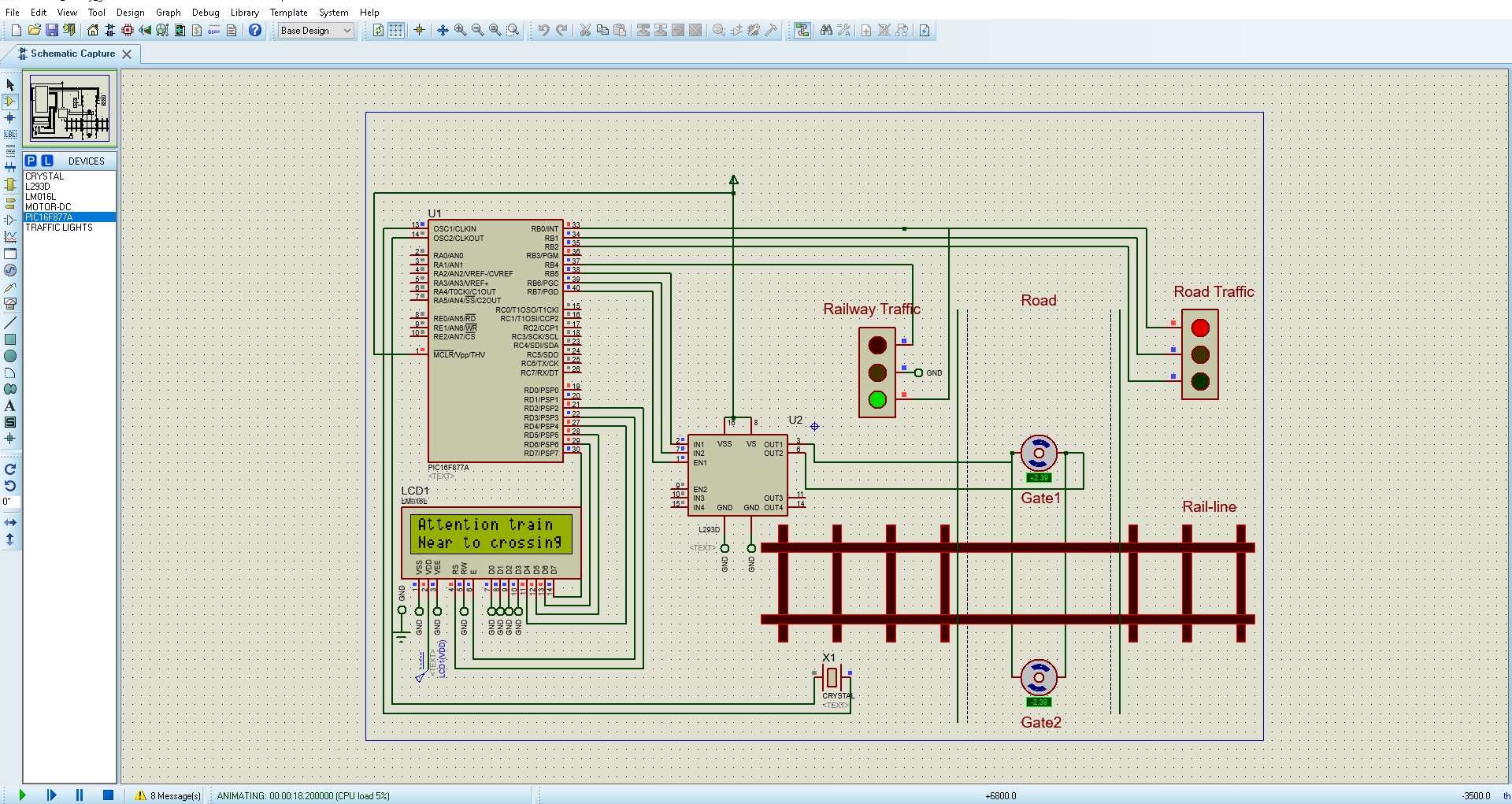


**Simulation Projects on Proteus**

Without Run

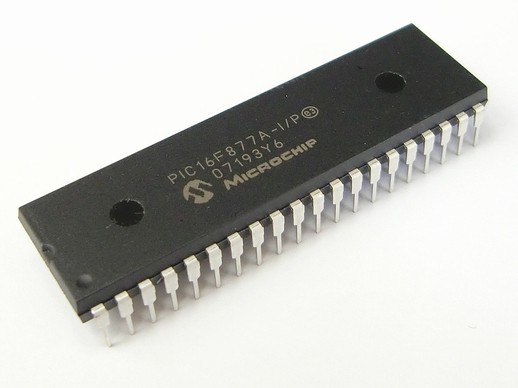


After Running Project on Proteus

**Description of Component**

## ****PIC16F877A microcontroller****

The PIC microcontroller **PIC16f877a** is one of the most renowned microcontrollers in the industry. This microcontroller is very convenient to use, the coding or programming of this controller is also easier. One of the main advantages is that it can be write-erase as many times as possible because it uses **FLASH memory technology**. It has a total number of 40 pins and there are 33 pins for input and output. PIC16F877A is used in many [**pic microcontroller projects**](https://microcontrollerslab.com/pic-microcontroller-projects-for-eee-students/). PIC16F877A also have much application in digital [**electronics circuits**](https://microcontrollerslab.com/electronics-projects/).

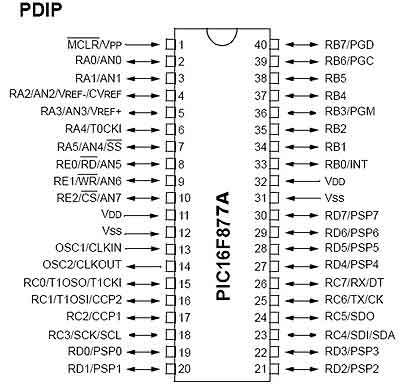


PIC16f877a finds its applications in a huge number of devices. It is used in remote sensors, security and safety devices, home automation and many industrial instruments. An[**EEPROM**](https://microcontrollerslab.com/eeprom-working-interfacing-with-microcontroller/) is also featured in it which makes it possible to store some of the information permanently like transmitter codes and receiver frequencies and some other related data. The cost of this controller is low and its handling is also easy. It is flexible and can be used in areas where microcontrollers have never been used before as in microprocessor applications and timer functions etc.

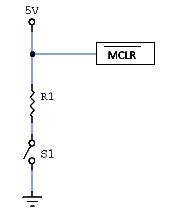
* It has a smaller **35 instructions set**.
* It can operate up to**20MHz frequency**.
* The operating voltage is between 4.2 volts to 5.5 volts. If you provide it voltage more than 5.5 volts, it may get damaged permanently.
* It does not have an internal oscillator like other [PIC18F46K22](https://microcontrollerslab.com/pic18f46k22-microcontroller-introduction/), [PIC18F4550](https://microcontrollerslab.com/introduction-pic18f4550-microcontroller/).
* The maximum current each PORT can sink or source is around 100mA. Therefore, the current limit for each GPIO pin of PIC16F877A is 10 mili ampere.
* It is available in **four IC packaging** such as 40-pin PDIP  44-pin PLCC, 44-pin TQFP, 44-pin QFN

## ****PIN CONFIGURATION AND DESCRIPTION Of**** ****PIC16F877A microcontroller****

* As it has been mentioned before, there are 40 pins of this microcontroller IC. It consists of two 8 bit and one 16 bit timer. Capture and compare modules, serial ports, parallel ports and five input/output ports are also present in it. This picture shows the **pinout diagram** of PIC16F877A.



* **PIN 1: MCLR:**The first pin is the master clear pin of this IC. It resets the microcontroller and is active low, meaning that it should constantly be given a voltage of 5V and if 0 V are given then the controller is reset. Resetting the controller will bring it back to the first line of the program that has been burned into the IC.



* A push button and a resistor is connected to the pin. The pin is already being supplied by constant 5V. When we want to reset the IC we just have to push the button which will bring the MCLR pin to 0 potential thereby resetting the controller.
* **PIN 2: RA0/AN0:**PORTA consists of 6 pins, from pin 2 to pin 7, all of these are bidirectional input/output pins. Pin 2 is the first pin of this port. This pin can also be used as an analog pin AN0. It is built in [analog to digital converter](https://microcontrollerslab.com/analog-to-digital-adc-converter-working/).
* **PIN 3: RA1/AN1:**This can be the analog input 1.
* **PIN 4: RA2/AN2/Vref- :**It can also act as the analog input2. Or negative analog reference voltage can be given to it.
* **PIN 5: RA3/AN3/Vref+:**It can act as the analog input 3. Or can act as the analog positive reference voltage.
* **PIN 6: RA0/T0CKI:**To timer0 this pin can act as the clock input pin, the type of output is open drain.
* **PIN 7: RA5/SS/AN4:**This can be the analog input 4. There is synchronous serial port in the controller also and this pin can be used as the slave select for that port.
* **PIN 8: RE0/RD/AN5:**PORTE starts from pin 8 to pin 10 and this is also a bidirectional input output port. It can be the analog input 5 or for parallel slave port it can act as a ‘read control’ pin which will be active low.
* **PIN 9: RE1/WR/AN6:**It can be the analog input 6. And for the parallel slave port it can act as the ‘write control’ which will be active low.
* **PIN 10: RE2/CS/A7:**It can be the analog input 7, or for the parallel slave port it can act as the ‘control select’ which will also be active low just like read and write control pins.
* **PIN 11 and 32: VDD:**These two pins are the positive supply for the input/output and logic pins. Both of them should be connected to 5V.
* **PIN 12 and 31: VSS:**These pins are the ground reference for input/output and logic pins. They should be connected to 0 potential.
* **PIN 13: OSC1/CLKIN:**This is the oscillator input or the external clock input pin.
* **PIN 14: OSC2/CLKOUT:**This is the oscillator output pin. A crystal resonator is connected between pin 13 and 14 to provide external clock to the microcontroller. ¼ of the frequency of OSC1 is outputted by OSC2 in case of RC mode. This indicates the instruction cycle rate.
* **PIN 15: RC0/T1OCO/T1CKI:**PORTC consists of 8 pins. It is also a bidirectional input output port. Of them, pin 15 is the first. It can be the clock input of timer 1 or the oscillator output of timer 2.
* **PIN 16: RC1/T1OSI/CCP2:**It can be the oscillator input of timer 1 or the capture 2 input/compare 2 output/ PWM 2 output.
* **PIN 17: RC2/CCP1:**It can be the capture 1 input/ compare 1 output/ PWM 1 output.
* **PIN 18: RC3/SCK/SCL:**It can be the output for SPI or I2C modes and can be the input/output for synchronous serial clock.
* **PIN 23: RC4/SDI/SDA:**It can be the SPI data in pin. Or in I2C mode it can be data input/output pin.
* **PIN 24: RC5/SDO:**It can be the data out of SPI in the SPI mode.
* **PIN 25: RC6/TX/CK:**It can be the synchronous clock or USART Asynchronous transmit pin.
* **PIN 26: RC7/RX/DT:**It can be the synchronous data pin or the USART receive pin.
* **PIN 19,20,21,22,27,28,29,30:**All of these pins belong to PORTD which is again a bidirectional input and output port. When the microprocessor bus is to be interfaced, it can act as the parallel slave port.
* **PIN 33-40: PORT B:**All these pins belong to PORTB. Out of which RB0 can be used as the external interrupt pin and RB6 and RB7 can be used as in-circuit debugger pins.

## ****HOW TO PROGRAM THE INPUT AND OUTPUT PORTS****

* As we have studied 5 input and output ports namely PORTA, PORTB, PORTC, PORTD and PORTE which can be digital as well as analog.
* We will configure them according to our requirements. But in case of analog mode, the pins or the ports can only act as inputs. There is a built in A to D converter which is used in such cases. Multiplexer circuits are also used.
* But in digital mode, there is no restriction. We can configure the ports as output or as input. This is done through programming. For PIC the preferable compiler is mikro C pro which can be downloaded from their website.
* There is a register named as ‘TRIS’ which controls the direction of ports. For different ports there are different registers such as TRISA, TRISB etc.
* If we set a bit of the TRIS register to 0, the corresponding port bit will act as the digital output.
* If we set a bit of the TRIS register to 1, the corresponding port bit will act as the digital input.
* For example to set the whole portb to output we can write the program statement as:

TRISB=0;

* Now the port will act as the output port and we can send any value on the output such as

PORTB=0XFF;

* FF represents all 1’s in binary i.e. FF=11111111, now all the pins of port b are high. If we connect LEDs at all the pins then they will all start glowing in this condition.
* If we want to negate the values of the port b we can use the statement:

PORTB=~PORTB;

* Now all the pins of the port b will be low.

## Compiler for PIC16F877A microcontroller

* Three popular compiler which are used to program pic microcontrollers are MPLAB XC8,  Mikro C for pic, PIC CCS compiler and Hi-Tech compiler.
* The official compiler is [MPLAB XC8 compiler](https://www.microchip.com/mplab/compilers) which is developed by manufactures of PIC16F877A.
* We generally recommend Mikro C for pic compiler for beginners and MPLAB XC8 compiler for those who want to learn pic microcontrollers programming from register level bare metal concepts.
* You can go through our [list pic microcontroller compilers](https://microcontrollerslab.com/pic-microcontroller-compiler/) article for further details.

## Main features of PIC16F877A microcontroller

Like all other microcontroller, PIC16F877A also provide built-in useful features as mentioned in this list:

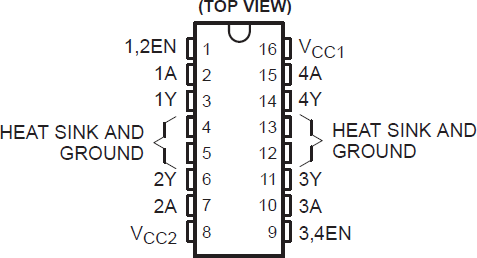
* **Analog to digital converter module :**It has 8 bit ADC module which consists of 8 channels.  We can use 8 analog sensors with this microcontroller.
* **Timers**: It provides three timers timer0, timer1 and timer2.  All these timers can be used either in timer mode or in counter mode.  These timers are used to generate delays, pulse width modulation, counting external events and timer interrupts. TIMER0 is a 8 bit timer and it can operate with internal or external clock frequency. When we use Timer0 in timer mode, we usually operate it with internal frequency and in counter mode, we trigger it with external clock source.  Similarly, TIMER1 is a 16-bit timer and it can also operate in both modes. TIMER2 is also of 8-bit. It is used with PWM as a time base for CCP module.
* **EEPROM :**It also has built-in Electrically erasable read only memory 256 x 8 bytes which can used to store data permanently even if the microcontroller is switched off, data will remain there.  It is usually used with [**electronics lock**](https://microcontrollerslab.com/electronic-lock-pic-microcontroller/)related projects.
* **PWM modules :**It also provide 2 CCP modules. CCP stands for capture compare PWM modules. We can easily generate two PWM signals with this microcontroller. The maximum resolution it supports is 10 bits. you can read[**PWM using PIC16F877A microcontroller tutorial**](https://microcontrollerslab.com/pwm-using-pic16f877a-microcontroller/) for more information and programming.
* **Serial or UART communication pins** : It support one UART channel. UART pins are used for serial communication between digital devices. RC7 pin is a transmitter or RX pin which is pin number 26. RC6 is a receiver or Tx pin which is pin number 25. For additional details, check this complete guide on [serial communication using pic16f877a microcontroller](https://microcontrollerslab.com/serial-communication-using-pic16f877a-microcontroller/).
* **I2C Communication** : PIC16F877A also support I2C communication and its has one module for I2C communication.  Pin#18/RC3 and 23/RC4 are **SCL**and **SDA**pins respectively. SCL is a serial clock line and SDA is serial data line.  [I2C communication tutorial](https://microcontrollerslab.com/i2c-communication-pic-microcontroller/) will help you understand further.
* **Interrupts** : Interrupts have wonderful applications in embedded systems field. If you don’t know about interrupts, I suggest you to get complete understanding about them, you will not get command on embedded programming them. PIC16F877A microcontroller provides 8 types of interrupts namley; External interrupts, timer interrupts, PORT state change interrupts, UART  interrupt, I2C, PWM interrupts. you can read this guide on [pic microcontroller interrupts](https://microcontrollerslab.com/how-use-pic-microcontroller-timers-interrupt/) for additional information.
* **Comparator module** : It has a comparator module which composed of two comparators. They are used for comparison of analog signal similar to comparators in electronics circuits.  Input pins for these comparators are RA0, RA1, RA2 and RA3 and output can measured through RA4 and RA5.
* **Watchdog timer** : WDT is a on chip separate oscillator which runs freely. It is a separate oscillator from OSC1/CLKI.  WDT will also work even if the device is in sleep mode.  It is used to wake up device from sleep mode and also used to generate watchdog timer reset.
* **Sleep mode** : PIC16F877A also provide sleep mode operation. In this mode, device operates at very low power. All peripherals draws minimum amount of current. Wake up from sleep mode from interrupts resources like timer1 interrupt, uart interrupt, EEPROM write completion operation  and many others.
* **Brown out detection** : It also has a brown out detection circuit which detects the significant drop in power supply voltage. If supply voltage drop from a certain limit, it will generate a interrupt signals.  This configuration bit (BODEN) is used to disable or enable this circuitry.
* **Brown out reset** : This option reset the device upon detection of brown out interrupt signal from BODEN signal. if supply voltage goes below threshold for more than 100 micro seconds,
* Programmable code protection, **Brown out reset**will occur and device will remain reset until the voltage raise to its nominal value.  Device checks for voltage after every 72ms.

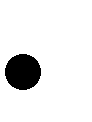
Some other momentous features are listed below :

* Power on Reset
* Multiple oscillator group
* In-Circuit Debugger
* In-Circuit Serial programming
* Low voltage ICSP programming

## L293D Motor Driver Introduction

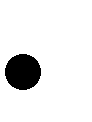
* L293d IC is known as a motor driver. It is a low voltage operating device like other ICs. The other ICs could have the same functions like L293d but they cannot provide the high voltage to the motor. L293d provides the continuous bidirectional Direct Current to the Motor. The Polarity of current can change at any time without affecting the whole IC or any other device in the circuit. L293d has an internal H-bridge installed for two motors.
* H-Bridge is an electrical circuit that enables the load in a bidirectional way. L293d bridge is controlled by external low voltage signals. It may be small in size, but its power output capacity is higher than our expectation. It could control any DC motor speed and direction with a voltage range of 4.5 – 36 Volts. Its diodes also save the controlling device and IC from back EMF. To control the max 600mA amount of current an internal “Darlington transistor sink” installed in it, which could be used to control a large amount of current by providing a small amount of current. It has also internal “pseudo-Darlington source” which amplifies the input signal to control the high voltage DC motor without any interception.
* **IC L293D**

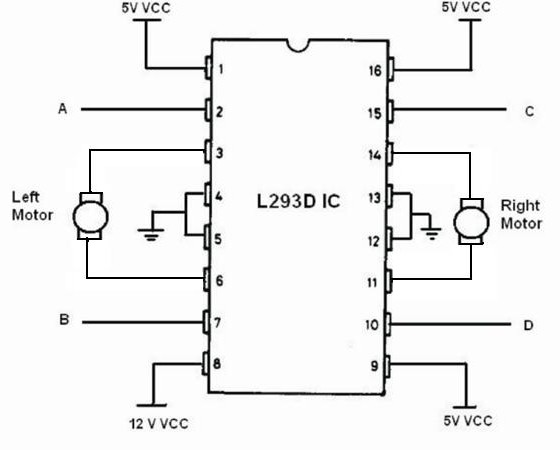


 **Description**:-

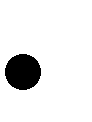
* The L293D are quadruple high-current half-H drivers. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36

V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo- Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 12EN and drivers 3 and 4 enabled by 3,4 When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

 **Block Diagram:-**



(Fig 4.11:- H bridge connection diagram of motor driven IC)

 **Feature:-**

* Wide Supply-Voltage Range: 4.5 V to 36 V
* Separate Input-Logic Supply
* Internal ESD Protection
* Thermal Shutdown
* High-Noise-Immunity Inputs
* Output Current 600m A Per Channel
* Peak Output Current 1.2 A Per Channel
* Output Clamp Diodes for Inductive Transient Suppression

### L293D pinoutIMG_256

### L293D Pin Configuration Details

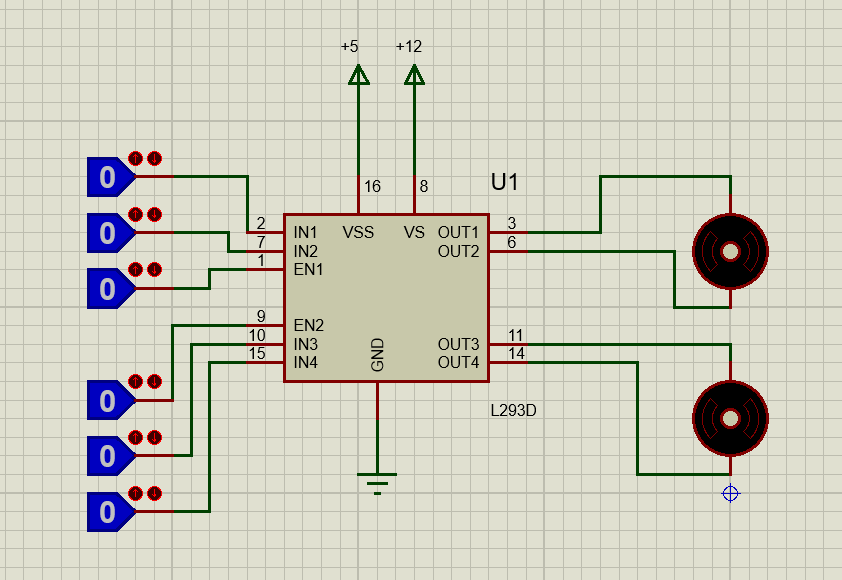
|  |  |
| --- | --- |
| **Pin1 (Enable)** | Pin 1 is known as the enable pin. It has a major effect on Input and output. If there is High logical signal on enable pin (EN) then there will be input and output between pin 2,3,6 & 7 (Input 1, Output 1, Input 2 & Output 2) |
| **Pin2 (Input 1)** | Mostly input means where we provide the input to give the output. But here Input 1 means which polarity we want to give at Output 1. |
| **Pin3 (Output 1)** | Output 1 is the input of the first motor/Motor 1. It attaches to its one end. |
| **Pin4 (Ground)** | The ground pin will attach to the ground of the circuit. |
| **Pin5 (Ground)** | The ground pin will be attached to the ground, and it will remain common with the previous ground. |
| **Pin6 (Output 2)** | Output 2 will attach to the input of the first motor/Motor 1. It will attach to its second end. |
| **Pin7 (Input 2)** | Input 2 will attach to the control button or device to control the Output 2 just like Input 1. |
| **Pin8 (Vcc)** | Pin8 is the voltage pin for Motor. It will device how much power we are going to attach the Motor. This Power should not be more than 36 volts and should not be less than 4.5 volts. |
| **Pin9 (Enable)** | Pin 9 is also the same as Pin 1. It controls the input and output signals. Pin 9 Controls the connection between Input 3, Input 4, Output 3 and Output 4. It also enables the connection when the logic signal will be High (1). |
| **Pin10 (Input 3)** | Input 3 will control the output polarity of the Pin 11 (Output 3) by logic signals. |
| **Pin11 (Output 3)** | Output 3 will be connected to the one end of the second motor. |
| **Pin 12 (Ground)** | Pin 12 will attach to the common ground with all other grounds. |
| **Pin13 (Ground)** | Pin 13 will also be attached to the common ground with all other grounds. |
| **Pin14 (Output 4)** | Pin 14 will attach to the second end of the second motor. |
| **Pin15 (Input 4)** | Pin 15 will control the output polarity of the Pin 14 (Output 4) by logic signals. |
| **Pin16 (Vcc)** | Pin 16 will the Power we will provide to the L293D to activate it or to turn it on. The power level of Pin 16 should be 4.5 – 7Volts. Voltage more than 7 will burn the IC |

### 

### ****FEATURES****

* L293d could be used to control the two motors at the same time.
* It has the ability to control the speed by using the enable pin.
* The direction is also easy to change.
* Voltage supply range is higher than other IC. Voltage range between 4.5-36 volts can easily handle by the IC to the motor.
* The motor has a maximum continuous range of current close to 600mA but the maximum peak current range is 1.2A
* It has an automatic shutdown system on thermal condition.
* Its working range is from 0 – 70 degree which is much higher for any small-sized IC.
* It has an internal back emp protection for IC and the controlling device.

## How to Use L293D?



L293d may have an internal complex circuit, but it is easy to use in real life. Just attach two motors with output pins. Remember at output 1 and output 2 same motor should be connected and it needs to be the same for output 3 and output 4. All ground should be common with both the power supplies provided to the IC. The enable pins are used to control the outputs but when we use the microcontrollers or microprocessor these enable pins can be used to control the speed of the motor.

### Motor Control Pins

The Input pins have a major role to control the direction of the motor. If Input 1 (Pin 2) is High and Enable (Pin 1) is high low but Input 2 (Pin 7) is low than motor will rotate clockwise at motor attached to the Pin 3 & 6. The rotation will be anti-clockwise when Input 2 (Pin 7) & Enable (Pin 1) is high but Input 1 (Pin 2) is low. To control the rotation of Pin 11 & 14 the Pin 9,10 & 11 will be used. If Input 1 & Input 2 has the same logic at the same time or enable 1 pin is low then there will be no rotation. This will be the same for Input 3, Input 4 and enable 2.

**Crystal oscillator**

* A **crystal oscillator** is an [electronic oscillator](https://en.wikipedia.org/wiki/Electronic_oscillator) circuit that uses the mechanical [resonance](https://en.wikipedia.org/wiki/Resonance) of a vibrating [crystal](https://en.wikipedia.org/wiki/Crystal) of [piezoelectric material](https://en.wikipedia.org/wiki/Piezoelectricity#History) to create an electrical signal with a constant [frequency](https://en.wikipedia.org/wiki/Frequency). This frequency is often used to keep track of time, as in [quartz wristwatches](https://en.wikipedia.org/wiki/Quartz_clock), to provide a stable [clock signal](https://en.wikipedia.org/wiki/Clock_signal) for [digital](https://en.wikipedia.org/wiki/Digital_data) [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuits), and to stabilize frequencies for [radio transmitters](https://en.wikipedia.org/wiki/Radio_transmitter) and [receivers](https://en.wikipedia.org/wiki/Radio_receiver). The most common type of piezoelectric resonator used is the [quartz](https://en.wikipedia.org/wiki/Quartz) crystal, so oscillator circuits incorporating them became known as crystal oscillators, but other [piezoelectric](https://en.wikipedia.org/wiki/Piezoelectric) materials including [polycrystalline](https://en.wikipedia.org/wiki/Polycrystalline) ceramics are used in similar circuits.
* A crystal oscillator relies on the slight change in shape of a quartz crystal under an [electric field](https://en.wikipedia.org/wiki/Electric_field), a property known as [electrostriction](https://en.wikipedia.org/wiki/Electrostriction) or inverse [piezoelectricity](https://en.wikipedia.org/wiki/Piezoelectricity). A voltage applied to an [electrode](https://en.wikipedia.org/wiki/Electrode) on the crystal causes it to change shape; when the voltage is removed, the crystal generates a small voltage as it elastically returns to its original shape. The quartz oscillates at a stable resonant frequency, behaving like an [RLC circuit](https://en.wikipedia.org/wiki/RLC_circuit), but with a much higher [Q factor](https://en.wikipedia.org/wiki/Q_factor) (less energy loss on each cycle of oscillation). Once a quartz crystal is adjusted to a particular frequency (which is affected by the mass of electrodes attached to the crystal, the orientation of the crystal, temperature and other factors), it maintains that frequency with high stability.

|  |
| --- |
| **Crystal oscillator** |
| IMG_256  A miniature 16 [MHz](https://en.wikipedia.org/wiki/MHz) [quartz crystal](https://en.wikipedia.org/wiki/Quartz_crystal) enclosed in a [hermetically sealed](https://en.wikipedia.org/wiki/Hermetically_sealed) HC-49/S package, used as the resonator in a crystal oscillator. |

## Operation

* A [crystal](https://en.wikipedia.org/wiki/Crystal) is a [solid](https://en.wikipedia.org/wiki/Solid) in which the constituent [atoms](https://en.wikipedia.org/wiki/Atom), [molecules](https://en.wikipedia.org/wiki/Molecule), or [ions](https://en.wikipedia.org/wiki/Ion) are packed in a regularly ordered, repeating pattern extending in all three spatial dimensions.
* Almost any object made of an [elastic](https://en.wikipedia.org/wiki/Elasticity_(physics)) material could be used like a crystal, with appropriate [transducers](https://en.wikipedia.org/wiki/Transducer), since all objects have natural [resonant](https://en.wikipedia.org/wiki/Resonance) frequencies of [vibration](https://en.wikipedia.org/wiki/Vibration). For example, [steel](https://en.wikipedia.org/wiki/Steel) is very elastic and has a high speed of sound. It was often used in [mechanical filters](https://en.wikipedia.org/wiki/Mechanical_filter) before quartz. The resonant frequency depends on size, shape, [elasticity](https://en.wikipedia.org/wiki/Elasticity_(physics)), and the [speed of sound](https://en.wikipedia.org/wiki/Speed_of_sound) in the material. High-frequency crystals are typically cut in the shape of a simple rectangle or circular disk. Low-frequency crystals, such as those used in digital watches, are typically cut in the shape of a [tuning fork](https://en.wikipedia.org/wiki/Tuning_fork). For applications not needing very precise timing, a low-cost [ceramic resonator](https://en.wikipedia.org/wiki/Ceramic_resonator) is often used in place of a quartz crystal.
* When a crystal of [quartz](https://en.wikipedia.org/wiki/Quartz) is properly cut and mounted, it can be made to distort in an [electric field](https://en.wikipedia.org/wiki/Electric_field) by applying a [voltage](https://en.wikipedia.org/wiki/Voltage) to an [electrode](https://en.wikipedia.org/wiki/Electrode) near or on the crystal. This property is known as [electrostriction](https://en.wikipedia.org/wiki/Electrostriction) or inverse piezoelectricity. When the field is removed, the quartz generates an electric field as it returns to its previous shape, and this can generate a voltage. The result is that a quartz crystal behaves like an [RLC circuit](https://en.wikipedia.org/wiki/RLC_circuit), composed of an [inductor](https://en.wikipedia.org/wiki/Inductor), [capacitor](https://en.wikipedia.org/wiki/Capacitor) and [resistor](https://en.wikipedia.org/wiki/Resistor), with a precise resonant frequency.
* Quartz has the further advantage that its elastic constants and its size change in such a way that the frequency dependence on temperature can be very low. The specific characteristics depend on the mode of vibration and the angle at which the quartz is cut (relative to its crystallographic axes).[[12]](https://en.wikipedia.org/wiki/Crystal_oscillator#cite_note-Virgil1982-12) Therefore, the resonant frequency of the plate, which depends on its size, does not change much. This means that a quartz clock, filter or oscillator remains accurate. For critical applications the quartz oscillator is mounted in a temperature-controlled container, called a [crystal oven](https://en.wikipedia.org/wiki/Crystal_oven), and can also be mounted on shock absorbers to prevent perturbation by external mechanical vibrations.

**Servo Motor**

servomotor is a rotary actuator or linear actuator that allows for precise controlof angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servo motors. Servomotors are not a specific class motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.



## Working Principle :

3 pairs of IR is used in this project.2 pairs of IR is placed at the rail line in such a distance that these IR pair can send signal to arduino to open or close the gate by servo motor. The third IR pair is placed under the gate .If there is any obstacle this IR pair check and send signal to microcontroller to read it so that the gate remain close. The gate become close or open by the rotation of servo motor. This is the main operation principle of this project

**APPENDIX A.: PROGRAMMING**

**This Programming Code Run in Keil uVision and generate Hex file**

**This program save to ( .c) extension,**

**sbit LCD\_RS at RD2\_bit;**

**sbit LCD\_EN at RD3\_bit;**

**sbit LCD\_D4 at RD4\_bit;**

**sbit LCD\_D5 at RD5\_bit;**

**sbit LCD\_D6 at RD6\_bit;**

**sbit LCD\_D7 at RD7\_bit;**

**sbit LCD\_RS\_Direction at TRISD2\_bit;**

**sbit LCD\_EN\_Direction at TRISD3\_bit;**

**sbit LCD\_D4\_Direction at TRISD4\_bit;**

**sbit LCD\_D5\_Direction at TRISD5\_bit;**

**sbit LCD\_D6\_Direction at TRISD6\_bit;**

**sbit LCD\_D7\_Direction at TRISD7\_bit;**

**void f1()**

**{**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**lcd\_out(1,1,"Attention train");**

**lcd\_out(2,1,"Near to crossing");**

**delay\_ms(10);**

**portb=0b10101001;**

**delay\_ms(10);**

**portb=0b00001001;**

**delay\_ms(1300);**

**//delay\_ms(100);**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**lcd\_out(1,1,"Gate is Closed");**

**lcd\_out(2,1,"Please Wait");**

**}**

**void f2()**

**{**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**delay\_ms(10);**

**lcd\_out(1,1,"Start your Car");**

**lcd\_out(2,1,"Gate is Opening");**

**delay\_ms(10);**

**portb=0b11000010;**

**delay\_ms(10);**

**portb=0b00010010;**

**delay\_ms(2000);**

**portb=0b00010100;**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**lcd\_out(1,1,"Gate is now Open");**

**lcd\_out(2,1,"Safe journey");**

**}**

**void fun()**

**{**

**int i,j;**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**lcd\_out(1,1,"Time Left");**

**for(i=1,j=9;i<=10;i++,j--)**

**{**

**Lcd\_Chr(2, 12, (j+'0'));**

**delay\_ms(100);**

**}**

**f1();**

**delay\_ms(1000);**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**lcd\_out(1,1,"Time Left");**

**for(i=1,j=9;i<=10;i++,j--)**

**{**

**Lcd\_Chr(2, 12, (j+'0'));**

**delay\_ms(100);**

**}**

**f2();**

**delay\_ms(1000);**

**}**

**void main()**

**{**

**TRISd=0x00;**

**TRISC=0b11111111;**

**trisb=0b00000000;**

**portb=0b00010100;**

**lcd\_init();**

**lcd\_cmd(\_LCD\_CURSOR\_OFF);**

**lcd\_out(1,5,"Automated");**

**lcd\_out(2,1,"Railway Crossing");**

**while(1)**

**{**

**delay\_ms(1000);**

**fun();**

**}**

**}**

**CONCLUSIONS**

* To save the human life and vehicles from miserable train accidents is a challenge of the era of modern science and technology.
* The working model was fabricated within the laboratory premises. The results exhibit that it is one of the expedient approaches for secure railway system.
* The ultrasonic sensors detect the train and stuck on the level crossing very quickly and communicate with the control unit.
* The control unit takes proper steps which lead the train and vehicles movements either to move forward or to stop to avoid collision.
* Consequently, this is able to play a great contribution to the railway gate automation with reliability and lower cost.
* In future this developed working model will be equipped with GPS to navigate the position of the train and the track to avoid collision between two trains.