# **ABSTRACT**

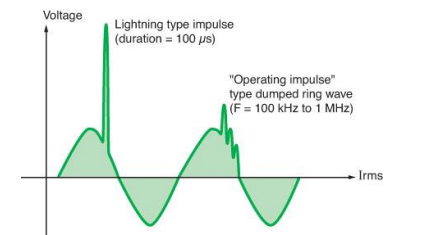
The aim of this project is to develop a high voltage detect mechanism. The fluctuation in AC mains supply is frequent in homes and industries. The abnormal over voltages may be caused due to various reasons such as sudden interruption of heavy load, lightening impulses, switching impulses etc. The sensitive electronic devices in these conditions can get easily damaged. It detects any voltage greater than 230V AC (pre-set value). If the voltage is greater than the pre-set value, it initiates a signal which in turn is given to Arduino nano.. For all the above process to happen, real time monitoring of the data is required. After getting this data, based upon the programming in the memory (ROM/RAM), the controller takes the decision of the tripping of the system. This device can be used directly as a standalone equipment between the mains supply and the load, or it may be inserted between an existing automatic/manual stabilizer and the load.

INTRODUCTION

Over-voltages occur in a system when the system voltage rises over 110% of the nominal rated voltage. There is always a chance for suffering of an electrical power system from abnormal over voltages. These abnormal over voltages may be caused due to various reason such as, sudden interruption of heavy load, lightening impulses, switching impulses etc. These over voltage stresses may damage insulation of various equipment’s and insulators of the power system. Although, all the over voltage stresses are not strong enough to damage insulation of system, but still these over voltages also to be avoided to ensure the smooth operation of electrical power system. The present work aims to develop an over voltage and over current relay for Domestic and small commercial electrical installations using ARDUINO at a cheaper cost.

OVER VOLTAGE

The word over voltage is in use from 1907. According to IEEE standards, Overvoltage is defined as: “Voltage between one phase and ground or between two phases, having a crest value exceeding the corresponding crest of maximum system voltage.” It can also be defined as the voltage in a circuit or part of it is raised above its upper design limit. Also an overvoltage is a voltage pulse or wave which is superimposed on the rated voltage of the network Figure



TYPES OF OVER VOLTAGES

Depending on the duration and the magnitude of the voltage, over voltages are classified into various types as follows:

EXTERNAL OVER VOLTAGES

This type of over voltages originates from atmospheric disturbances, mainly due to lightning. This takes the form of a surge and has no direct relationship with the operating voltage of the line. It may be due to any of the following causes:

1. Direct lightning stroke: A lightening stroke is defined as a direct stroke if it hits either the tower or shield wire or the phase conductor. When the insulator string flashes over by direct hit either to the tower or to the shield wire along the span, it is called back flash.
2. Electromagnetically induced over voltages due to lightning discharge taking place near the line are called 'side stroke'.
3. Voltages induced due to atmospheric changes along the length of the line
4. Electrostatically induced voltages due to presence of charged clouds nearby.

Electrostatically induced over voltages due to the frictional effects of small particles like dust or dry snow in the atmosphere or due to change in the altitude of the line

**Internal Over voltages:**

These over voltages are caused by changes in the operating conditions of the power system. These can be divided into two groups as below:

**1. Switching over voltages or Transient over operation voltages of high frequency:**

A short-duration highly damped, oscillatory, or non-oscillatory overvoltage, having duration of few milliseconds or less is Transient overvoltage. This is caused when switching operation is carried out under normal conditions or when fault occurs in the network. When an unloaded long line is charged, due to Ferranti Effect the receiving end voltage is increased considerably resulting in over voltage in the system. Similarly when the primary side of the transformers or reactors is switched on, over voltage of transient nature occurs.

**2. Temporary over voltages:**

An Oscillatory phase-to-ground or phase-to phase overvoltage that generally exists for long duration (seconds, even minute) and that is un-damped or only weakly damped is Temporary over voltage. Temporary overvoltage usually originate from switching operation or faults (e.g. load rejection, single-phase fault, fault on a high-resistance ground or ungrounded system) or from nonlinearities (Ferro resonance, harmonics), or both. They are characterized by the amplitude, the oscillation frequencies, the total duration or the decrement.

**CAUSES OF OVER VOLTAGES**

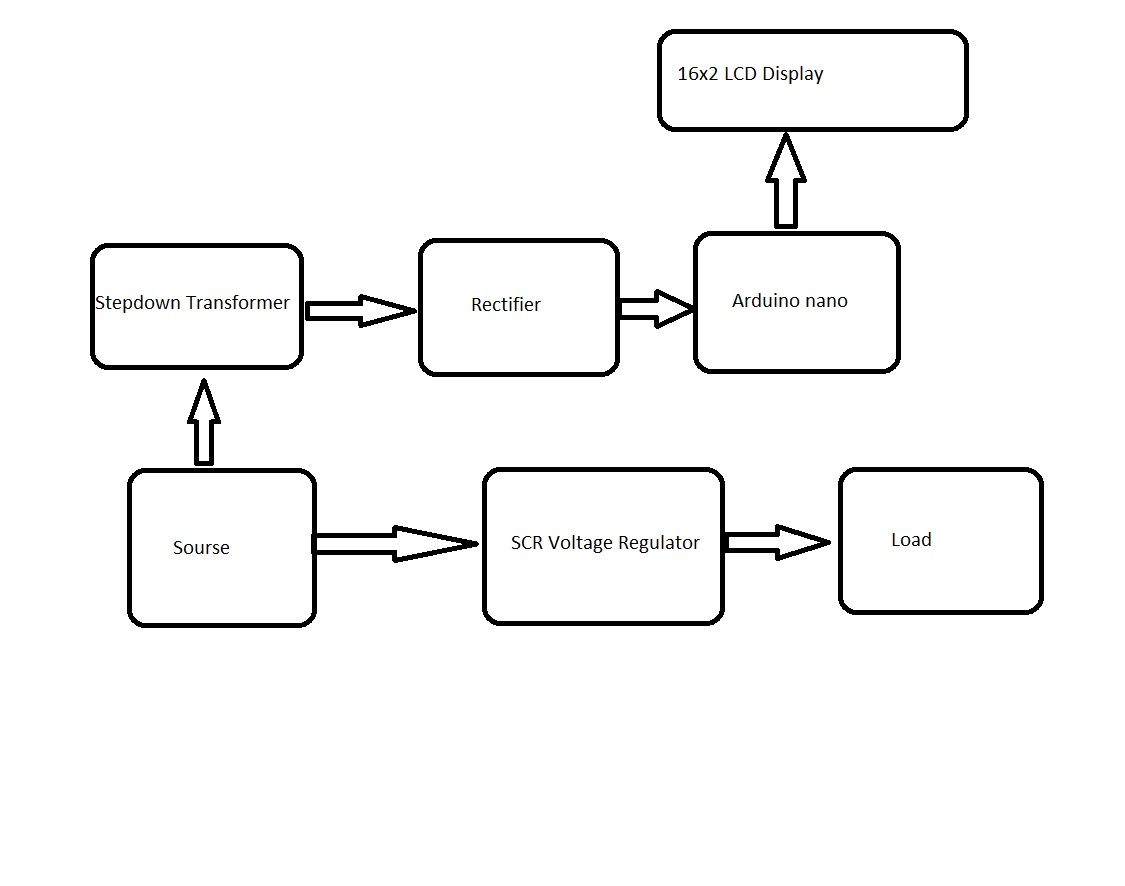
INTERNAL CAUSES: (I) switching surges: A surge, or transient, is a sub cycle overvoltage with a duration of less than a half-cycle of the normal voltage waveform. A surge can be either positive or negative polarity, can be additive or subtractive from the normal voltage waveform, and is often oscillatory and decaying over time. Surges, or transients, are brief overvoltage spikes or disturbances on a power waveform that can damage, degrade, or destroy electronic equipment within any home, commercial building, industrial, or manufacturing facility. Transients can reach amplitudes of tens of thousands of volts. Surges are generally measured in microseconds. Few cases of switching surges are discussed here.

(II) Insulation failure: Electrical breakdown or dielectric breakdown is a long reduction in the resistance of a 12 electrical insulator when the voltage applied across it exceeds the breakdown voltage. This results in the insulator becoming electrically conductive. Electrical breakdown may be a momentary event (as in an electrostatic discharge), or may lead to a discontinuous arc charge if protective devices fail to interrupt the current in a low power circuit. Under sufficient electrical stress, electrical breakdown can occur within solids, liquids, gases or vacuum. However, the specific breakdown mechanisms are significantly different for each, particularly in different kinds of dielectric medium. (III) Arcing ground: Arcing Grounds is a phenomenon which is observed in ungrounded three phase systems. In ungrounded three phase systems operating in a healthy balanced conditions, capacitances are formed between the conductors and ground. The voltage across these capacitances is the phase voltage. Now, in the event of a ground fault, the voltage across the faulty conductor becomes zero while the voltages across the healthy conductors increase by a factor of 1.732. The arc caused between the faulty conductor and the ground gets extinguished and restarts many times, this repeated initiation and extinction of the arc across the fault produces severe voltage oscillations of the order of nearly three to four times the nominal voltage. This repeated arcing across the fault due to the capacitances between the conductors and the ground is known as arcing grounds.

**EXTERNAL CAUSES:**

(I) Lightening: A lightning strike creates over voltages that propagate along any type of electrical cabling (electrical distribution mains, telephone connections, communication bus, etc.), metallic 13 wiring systems or conducting elements of significant length. The consequences of lightning, i.e. the over voltages created on the installations and equipment, can be appreciable over a radius of 10km. • Internal causes do not produce surges of large magnitude. • Experience shows that surges due to internal causes hardly increase the system voltage to twice the normal value. • Generally, surges due to internal causes are taken care of by providing proper insulation to the equipment in the power system. • However, surges due to lightning are very severe may increase the system voltage to several times the normal value. • If the equipment in the power system is not protected against lightning surges, these surges may cause considerable damage. • In fact, in a power system, the protective devices provided against overvoltages mainly take care of lightning surges.

BLOCK DIAGRAM



**DESCRIPTION OF BLOCK DIAGRAM:**

(1) AC Input: This is the input supply from the public utility where the device will be energized. It is also supplied directly to the relay contacts in the device which connects the load to the supply when the supply is within 200V – 240V range.

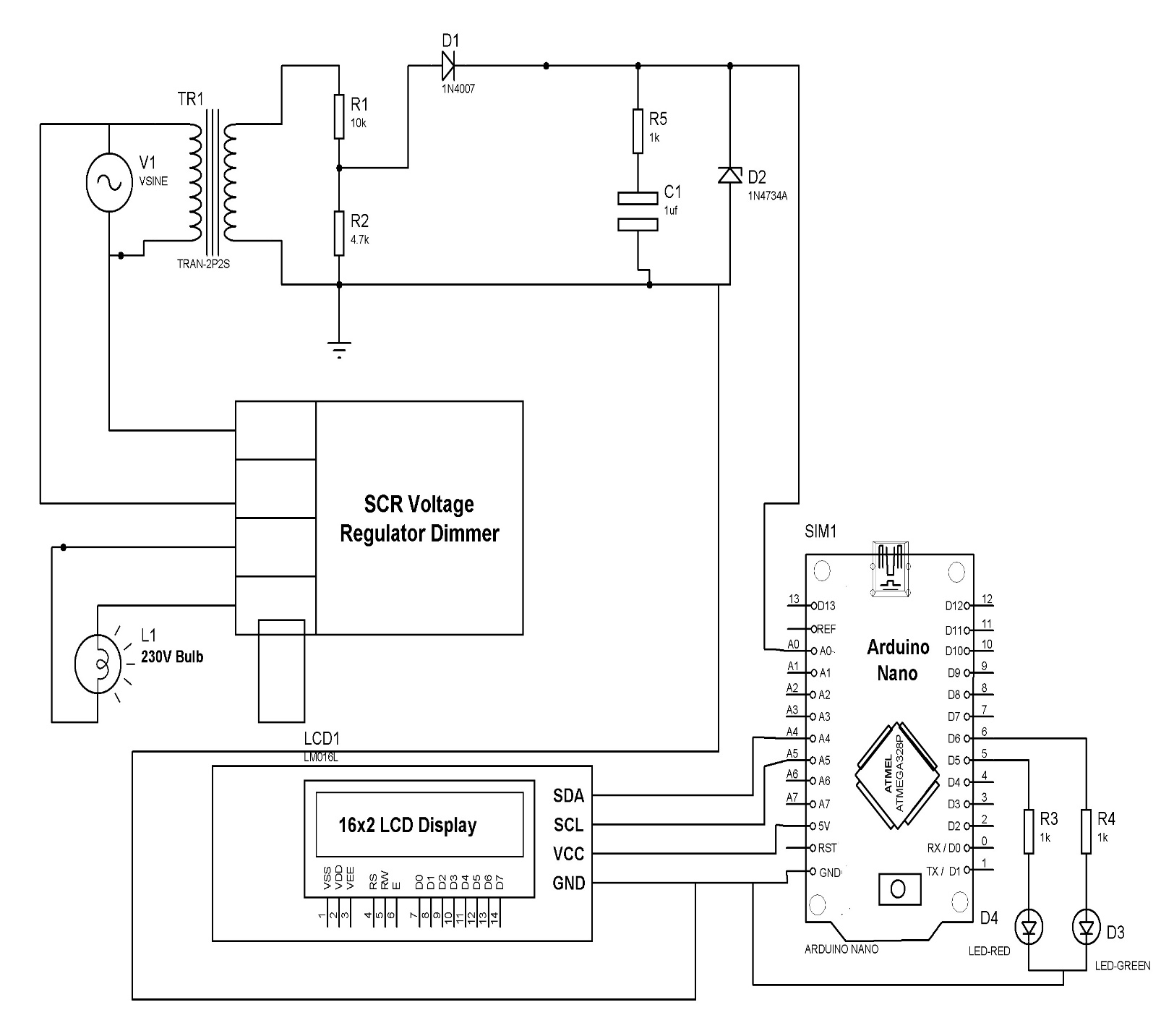
(2) Step down transformer: It steps down the AC supply into 12v on the secondary side. It is therefore a 230/12 v transformer. Any change in the primary reflects in the secondary of the transformer. So any fluctuations in the input is also reflected as a fluctuation in the output.

(3) Rectifier: A transformer, with single diodes for rectification is used to convert the ac voltage to a pulsating dc voltage followed by a filter, comprising of a capacitor to filter out (smooth) the pulsation. After the rectification and smoothening, a sample of the output voltage is fed to the Arduino. This voltage is unregulated and therefore varies as the input mains voltage varies. Since the system is to prevent against over voltage, the transformer was designed and the windings were so selected for the device to be able to sense and withstand input mains voltage up to 300Vac.

(4) Arduino: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. Arduino is the controller used in this project. It compares the input fluctuations with the preset value. If the fluctuations are within the limit then it makes the pin connected to the relay high. This trips the relay

(6) LCD Display: This displays the supply voltage as well as some information at “switch on” or when the supply voltage is out of range of the desired pre-set range of values. The LCD used is having a 16x 2 display.

**Circuit Diagram**

****

**OPERATION**

The ac supply at our homes is usually 230 V. Due to the fluctuations in load, it might vary. A tolerance of +2% is accepted. In case of increase in mains above 2%, the load might get damaged. rectifier converts the ac supply to dc. A filter comprising of a capacitor is connected to smoothen the pulsation. After the rectification and smoothening, a sample of the output voltage is fed to the Arduino. This voltage is unregulated and therefore varies as the input mains voltage varies. As the value of capacitance increases, the ripple content decreases. The capacitor used in this circuit is 1 micro farads. This is followed by a potential divider. The variable of the potential divider is connected to the input of the Arduino. Arduino has five analog input pins and 13 digital output pins. It has an inbuilt analog- digital converter. So, five different loads can be connected at a time. The 13 th pin contains a LED. Arduino takes an input voltage of 5-12 V and gives an output of 5 V or 3.3 V. A preset value with tolerance is given to the Arduino. The Arduino compares the preset value with the analog read value at A0.

If supply not given to system LCD Display on Supply not connected.if voltage voltage greater than threshold voltages then display over voltage detect. Also vary the intensity of bulb depending upon voltage level. We use voltage regulator module for vary voltage. Secondary side of transformer is 12V AC.so vary the voltage at primary side also deflect output at secondary side .this voltage given to arduino nano board Analog pin A0.Arduino process this signal & perform the operation depending upon programming.if voltage is greater than threshold voltage message shown on lcd display also red colour led on & normal opration green led on.we use I2C mudule for lcd display so we perform operation only four wire as shown on circuit diagram.Pin like VCC,GROUND,SDA,SCL.so reduce bunch of wire using I2C module

**HARDWARE**

1. **ARDUINO NANO:**

Arduino is an open-source electronics prototyping platform based on flexible, easy to use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer.

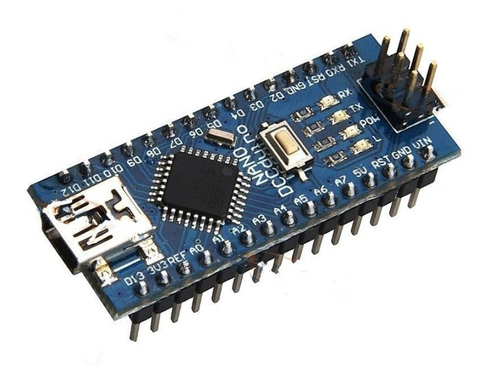
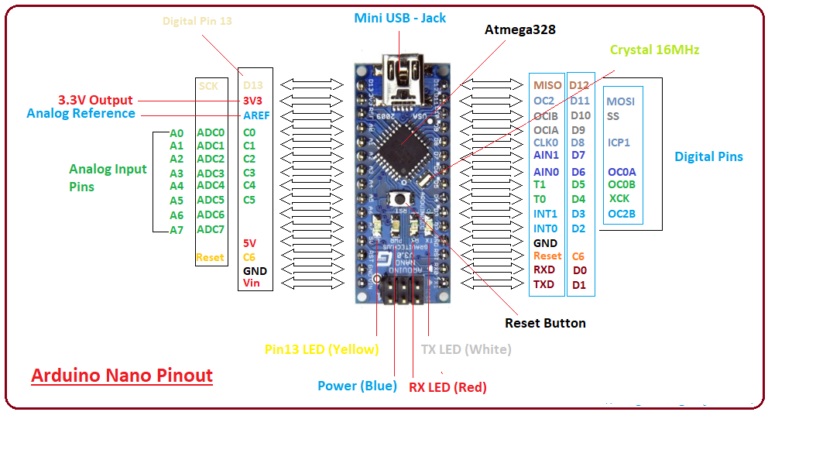


Fig. 5.1 Arduino nano

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that has electrically with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper. Each of the 14 digital pins on the Nano can be used as an input or output, using pin mode, digital write, and digital read functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull up resistor (disconnected by default) of 20-50 kOhms.[8]

****

In addition, some pins have specialized Functions:-

* Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
* External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt function for details.
* PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write function.
* SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the arduino language.
* LED: There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW value, then it's off. The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analog reference function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:
* AREF : Reference voltage for the analog inputs. Used with analog Reference.
* **I2C.** I2C communication is developed using A4 and A5 pins where A4 represents the serial data line (SDA) which carries the data and A5 represents the serial clock line (SCL) which is a clock signal, generated by the master device, used for data synchronization between the devices on an I2C bus

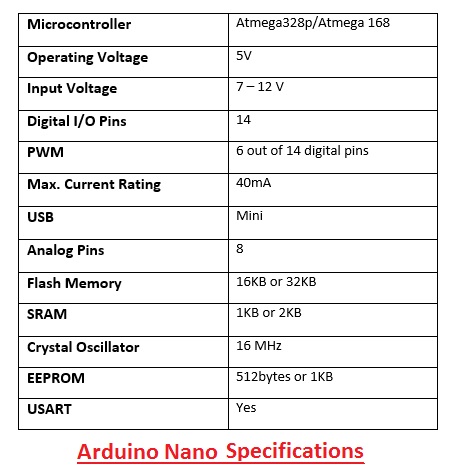
Power :

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.[8]

Memory :

The ATmega328p has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM. Reset Pin of this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.[6]

Features:   
  
• Automatic reset during program download  
• Power OK blue LED  
• Green (TX), red (RX) and orange (L) LED  
• Auto sensing/switching power input  
• Small mini-B USB for programming and serial monitor  
• ICSP header for direct program download  
• Standard 0.1” spacing DIP (breadboard friendly)  
• Manual reset switch

****

|  |  |
| --- | --- |
|  |  |
|  |  |

**TRANSFORMER :**

Transformer used is the step down transformer to reduce 230 volt to +12 volt. It provides isolation from the mains. Step down transformers are mainly designed to reduce electrical voltage.

**RECTIFIER :**

The rectifier is used to convert A.C to D.C voltage. The designed that we have carried out of the rectifier circuit diode 4007 are use.

### LIQUID CRYSTAL DISPLAY:

This is an example for the Parallel Port. This doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all Parallel Ports. It however doesn't show the use of the Status Port as an input. A 16 Character x 2 Line LCD Module to the Parallel Port. A **16x2 LCD** means it can display 16 characters per line and

there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.



**44780 LCD BACKGROUND:**

The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred as EN, RW and RS

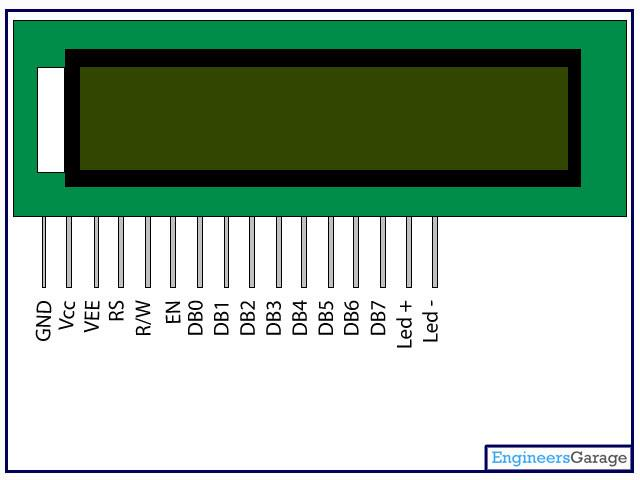
**EN:** The **EN** line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

**RS:** The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

**RW:** The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

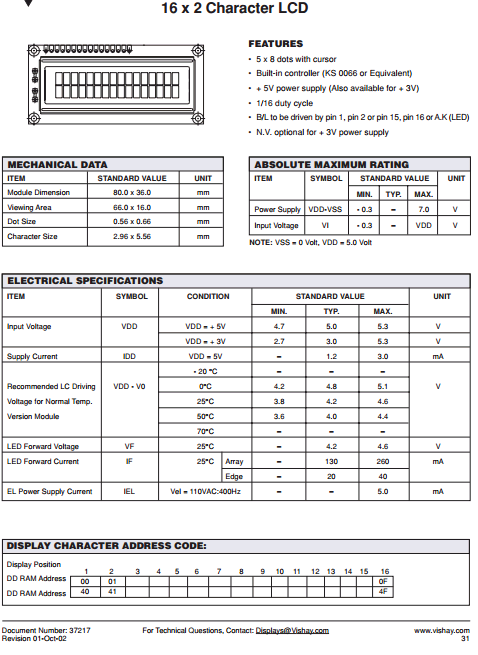
Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

* + 1. **LCD PIN OUT**



**Pin Description :-**

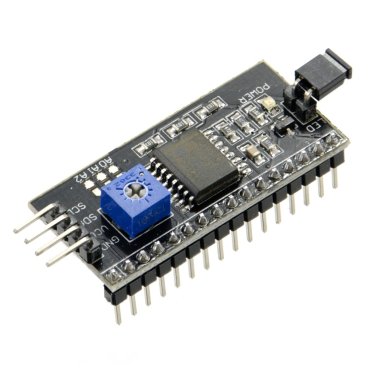
|  |  |  |
| --- | --- | --- |
| Pin no. | Functions | Name |
| 1. | Ground | Ground |
| 2. | Supply voltage 5V(4.7V-5.3V) | Vcc |
| 3. | Contrast adjustment; through a variable resistor | VEE |
| 4. | Selects command resistor when low; and data data resistor when high | Register  Select |
| 5. | Low to write to the register; High to read from the register | Read/Write |
| 6. | Sends data to data pins when high to low pulse given | Enable |
| 7-14 | 8-bit data pins | DB0-DB7 |
| 15. | Backlight Vcc (5V) | LED+ |
| 16. | Backlight Ground (0V) | LED - |

****

## **I2C LCD**

The classic parallel LCD sometimes post a problem for projects that use a lot of [Arduino pins](https://www.teachmemicro.com/arduino-pins-manipulation/). The least amount of pins you can use is six, excluding the power pins and the potentiometer contrast adjust pin. by using an I2C LCD "backpack", the pin use can be reduced to four!

### **I2C LCD Backpack**



At the center of the board is the [PCF8547 controller](https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf) by NXP. The row of pins is attached to the same row of pins on any HD44870-compatible LCD

## Pin Configuration of I2C LCD Display



That black adapter uses the **PCF8574T IC** chip which converts I2C serial data to parallel data for the LCD display. The blue color component you have seen in the above picture is a potentiometer which is used to adjust the brightness of the display.

Connections  
  
First of all we connect i2c pins module as shown in the schematic. Power the LCD module to 5 volts and connect the ground as well. The SDA pin of the i2c module conected to arduinio A5 and the SCL pin to A4. We connect the arduino to USB and we are ready to program. In order to make the LCD work we download LCD library for arduino.

* 5V to your 5V power pin - don't use 3.3V power, the LCD needs 5V for contrast!
* GND to Ground

The **I2C**protocol involves using two lines to send and receive data: a serial clock pin (SCL) that the **Arduino** or Genuino Master board pulses at a regular interval, and a serial data pin (SDA) over which data **is** sent between the two devices.

SOFTWARE

1. **ARDUINO IDE**

Arduino IDE is a lightweight, cross-platform application that introduces programming to novices. It has both an online editor and an on-premise application, for users to have the option whether they want to save their sketches on the cloud or locally on their own computers.

While Arduino IDE is highly-rated by users according to ease of use, it is also capable of performing complex processes without taxing computing resources.

With Arduino IDE, users can easily access contributed libraries and receive up-to-date support for the latest Arduino boards, so they can create sketches that are backed by the newest version of the IDE.

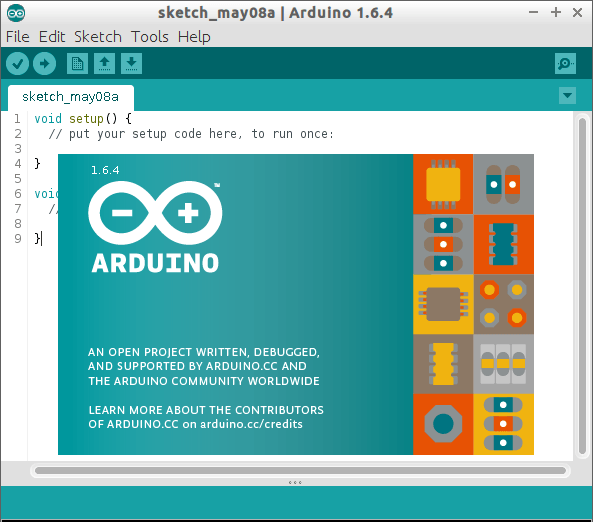
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux.

The Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application written in [Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29), and derives from the IDE for the [Processing programming language](https://en.wikipedia.org/wiki/Processing_%28programming_language%29) and the [Wiring](https://en.wikipedia.org/wiki/Wiring_%28development_platform%29) projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [brace matching](https://en.wikipedia.org/wiki/Brace_matching), and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Arduino programs are written in [C](https://en.wikipedia.org/wiki/C_%28programming_language%29) or [C++](https://en.wikipedia.org/wiki/C%2B%2B). The Arduino IDE comes with a [software library](https://en.wikipedia.org/wiki/Software_library) called "[Wiring](https://en.wikipedia.org/wiki/Wiring_%28development_platform%29)" from the original Wiring project, which makes many common input/output operations much easier. The users need only to define two functions to make an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program:

Setup (): a function that runs once at the start of a program and that can initialize settings.Loop : a function called repeatedly until the board powers off.

A typical first program for a microcontroller simply blinks an [LED](https://en.wikipedia.org/wiki/Light-emitting_diode) on and off. In the Arduino environment, the user might write a program like this:

Power LED (Red) and Integrated LED on Line 13 (Green) on Arduino Compatible Board.

# 

# The Arduino IDE supplies

**Multi-Platform Application**

Arduino IDE works on the three most popular operating systems: Windows, Mac OS, and Linux. Aside from that, the application is also accessible from the cloud. These options provide programmers with the choice of creating and saving their sketches on the cloud or building their programs locally and upload it directly to the board.

**Board Management**

Arduino IDE comes with a board management module, where users can select the board they want to work with at the moment. If they wish to change it, they can do so easily from the dropdown menu. Modifying their selection also automatically updates the PORT infos with the data they need in relation to the new board.

**Straightforward Sketching**

With Arduino IDE, users can create programs called sketches that are built with a text editor. The process is a straightforward one though it has several bells and whistles that make the experience more interactive.

**Project Documentation**

Arduino IDE offers programmers the option to document their projects. This function allows them to keep track of their advancements and any changes they make every time. Apart from that, documentations allow other people to easily employ the sketches to their own boards.

**Simple Sketch Sharing**

Aside from saving and archiving sketches and uploading them to the board, Arduino IDE is also capable of sharing sketches (available only on the cloud version). Each sketch is given its own unique URL that users can share with their colleagues and fellow Arduino hobbyists. The recipient then has access to the code; they can save it in the cloud sketchbook or download it for their own use.

**Vast Library**

Arduino IDE has more than 700 libraries integrated. These were written and shared by members of the Arduino community that other users can utilize for their own projects without having to install anything. This enables programmers to add a different dimension to their sketches.

# **PROTEUS SIMULATION SOFTWARE**

Proteus is a software package for computer-aided design, simulation and design of electronic circuits. It consists of two main parts, the ISIS , the circuit design environment that even the simulator VSM includes, and the ARES , the PCB -Designer. Developer and manufacturer of the software package is the company Lab center Electronics

Proteus 8 is a best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So it is a handy tool to test programs and embedded designs for electronics hobbyist. One can simulate your programming of microcontroller in Proteus 8 Simulation Software.

After Simulating the circuit in Proteus 8 Software one can directly make PCB design with it so it could be a all in one package for students and hobbyists. So now one can have a little bit idea about what is proteus software. Latest version of proteus is proteus 8 demo version available for free of cost. The latest Proteus Software version is [Proteus 8](http://www.geniusdevils.com/2013/03/Proteus-Latest-Version-8-released-download-it-for-free.html).

**DEVELOPMET STAGES & PROCESS:**

The complete development of this system can be divided into the following stages:

* Problem definition stage
* Designing block diagram
* Implementing circuits and components
* Developing algorithm for software
* Writing actual code for Microcontroller
* Compiling the code
* Testing and Running.

* **Problem definition stage**

This is the very first stage to develop any project. It actually defines the aim and the concept of the project.

* **Designing block diagram**

At this stage we have categorized the whole system into different individual modules. These modules (block diagrams) will be helpful in understanding the concept and working of the integrated system. It also simplifies the entire debugging and testing process.

* **Implementing circuits and components**

This is the actual implementation of circuit of each block. At this stage we have actually designed each block separately and finally integrated them into the complete working system.

* **Developing algorithm for software**

To get the logical flow of the software, the development of algorithm is having a prominent role. So that we have analyzed the complete system and organized the algorithm in such a manner that one can understand the complete working of the software.

* **writing actual code for Microcontroller**

After the development of the algorithm and flowchart we have actually translated them in C language for Atmega328 Microcontroller so that it can understand the instructions and run as per our requirement.

* **Compiling the code**

The code is implemented on the computer for which we have used Arduino IDE pre-installed on PC. Proteus software used for simulate working of Microcontroller in real time without burning the software into actual IC. We simulated and compiled our program for error checking. After removing of several compiling errors the program was converted into machine language .

* **Testing and Running**

This time we tested our project for actual working, after loading the software into the microcontroller. Any errors found were removed successfully. This is the last and final stage of development of our project.

**CONCLUSION**

The aim of designing and constructing a low cost Arduino based over voltage protective device was achieved in this work. The device supplies power to the connected load whenever the input supply is within the required pre-set voltage, thereby protecting the output connected loads from un-necessary damages. The device is found to be economical, easier to maintain and repair. The use of Arduino based relay is that the same circuit can be used also as under voltage and over current relay just by changing the coding of the program. Also characteristics like definite time, inverse, very inverse, extremely inverse and many other can be employed. Here we employed definite time and inverse characteristics .The same circuit can be used to trip 5 different loads at the same time.

# **APPLICATIONS**

### Applications of Overcurrent Relay

**Motor Protection:**

Used against overloads and short-circuits in stator windings of motor. Inverse time and instantaneous overcurrent phase and ground Overcurrent relays used for motors above 1000 kW.

**Transformer Protection:**

Used only when the cost of overcurrent relays are not justified. Extensively also at power-transformer locations for external-fault back- up protection.

**Line Protection:**

On some sub transmission lines where the cost of distance relaying cannot be justified.

primary ground-fault protection on most transmission lines where distance relays are used for phase faults.

For ground back-up protection on most lines having pilot relaying for primary protection.

**Distribution Protection:**

Overcurrent relaying is very well suited to distribution system protection for the following reasons:

It is basically simple and inexpensive.

Very often the relays do not need to be directional and hence no PT supply is required.

It is possible to use a set of two O/C relays for protection against inter- phase faults and a separate Overcurrent relay for ground faults.

### Application of over voltage relay

In a sparse power system, the shunt capacitances and series inductances of long transmission lines can cause significant voltage variations between high and low load periods. Overvoltage relays in substations are used to protect power transformers from damage due to high voltage. In the Peruvian system, these relays are usually located only at power transformers or bus bars. But sometimes it is better to trip selected long lines before tripping power transformers to mitigate temporary overvoltage’s. Tripping of power transformers and their connected loads exacerbates the temporary overvoltage leading to the possibility of a widespread black out. Another important issue is coordinating the overvoltage relay settings system wide.

# **BIBILOGRAPHY**

#### IEEE REFERENCES:

##### *International Journal of Scientific Engineering and Technology* (ISSN : 2277-1581)

*Volume No.3 Issue No.9, pp : 1225-1229.*

1. *Bayindir R., Sefa I., Cola I., and Bektas A. (2008)*

*“Fault Detection and Load Protection Using Sensors”, IEEE Transactions on Energy Conversion, Vol. 23, Issue 3, pp. 734– 741.*

1. *Close K. J., and Yarwood J. (1979) “Experimental*

*Electronics for Students” 1st Edition, Chapman and Hall Ltd, Britain.*

1. *Delton T. H. (1989) “Basic Electronic Theory with Projects and Experiment” 3rd Edition; TAB Book Inc. Blue Ridge Summit, P.A.*
2. *Gurevich H., and Vladimir S. (2005) “Electrical Relays: Principles and Applications” CRC Press, London- New York.*
3. *Ian S. L. (2000) “Passive Components for Circuit Design”, Texas, USA, p. 170. ISBN 008051359X.*
4. *Maddock R. J. and Calcute P. (1994) “Electronics: A Course for Engineers” 2nd Edition; Longman Essek, London.*
5. *Martin P. B. (2008) “Programming 8-Bit PIC Microcontrollers in C with Interactive Hardware Simulation” ISBN: 978-0-7506-8960-1.*
6. *Paul H., and Winfield H. (1989) “The Art of*

*Electronics”, Second Edition, Cambridge University Press, Cambridge MA, p. 58, ISBN 0-521-37095-7.*