CHAPTER ONE
INTRODUCTION

1.1 General concepts

Voting is a crucial process to reveal the opinion of a group on an issue that is under consideration. In addition security is indeed an essential part of voting. The term electronic voting depicts to the use of some electronic means in voting and ensure the security, reliability and guarantee. Compared to the traditional voting, electronic voting is considered to have many greater potential benefits, these benefits include better accuracy by eliminating the negative factor of human error and increased speed for tally computation [1].

The Arduino is an electronic development board, it composed of an electronic circuit with open source microcontroller on a single panel. It programmed by computer and it designed to make the process of using interactive electronics in multi -disciplinary projects more easily.

1.2 Problem Statement

In the traditional manual voting process could occurs several problems such as the one person to vote more than once or the possibility of something goes wrong during the process of collections and calculations the votes. May a person will vote for a candidate only to prevent the embarrassment, without conviction his opinion.

1.3 Objectives

The main aims of this study are to:

- Design and implementation of an electronic voting machine to insure the accuracy and reliability of the voting process.
- Design program allows to a person to vote only once time and prevent embarrassment by not reveal the identity of the candidate who has been voted.

1.4 Methodology

- Study all the previous works related to the problem.
- Study and understand different component such as microcontroller and relays.
- Use of Protues software to model of design and simulate the proposed system.
- Build the proposed system based on the Arduino.
- programming Arduino using c language.
- Evaluate the performance of voting system based on simulation and practical results.

1.5 Thesis Layout

This thesis consists of five chapters including chapter one. Chapter two presents theoretical backgrounds, introduction of voting technologies, microcontroller definition and introduction of Arduino. Chapter three concentrates on system's architecture; hardware construction, software code and simulation. Chapter four discusses implementation of the system, testing and experimental results. Finally, chapter five gives the conclusion and recommendations of the research.

CHAPTER TWO
THEORETICAL BACKGROUND

2.1 Voting Technologies

Here is an overview of voting technologies; to introduce the subject of electronic voting, and also to provide some historical context. These include paper ballots, mechanical lever machines, punch cards, mark sense, and direct recording electronic voting systems.

2.1.1 Paper ballot

Paper ballot systems use official ballots with the names of all candidates and issues printed on them. Voters mark boxes next to the candidates or issues of their choice in private, and drop the completed ballot into a sealed ballot box. This system, also known as the “Australian ballot” because it was first adopted in the Australian State of Victoria in 1856, was first used in the United States in a statewide election in New York in 1889 [1].

2.1.2 Mechanical lever voting

In mechanical lever voting systems shown in Figure 2.1, the voter makes a selection by pulling a lever assigned to a candidate or issue choice identified by a printed strip. When the voter opens the privacy curtain and exits the voting booth, the levers are automatically returned to their original positions. As the levers return to position, each causes a wheel to turn one-tenth of a full rotation for the counted vote. This wheel serves as the ones position of the count for the particular lever. After each full rotation of this wheel, this wheel causes a tens wheel to turn one-tenth of a full rotation. Similarly, the tens wheel updates a hundreds wheel [1].

If the mechanical connections work properly and the counting wheels are initially set to zero, the number of votes cast is measured by the position of each counter when the polls close. Mechanical lever machines were invented by Thomas Edison as a way to deter the vote fraud (such as ballot stuffing) that was occurring at the time. The first official use of a mechanical lever voting machine was in New York in 1892. By 1930, they were in almost every large city in the United States. In the1960s, over half of the country's votes were counted by mechanical lever machines. These machines are no longer made, and are being replaced with mark sense or direct recording electronic voting systems [1].

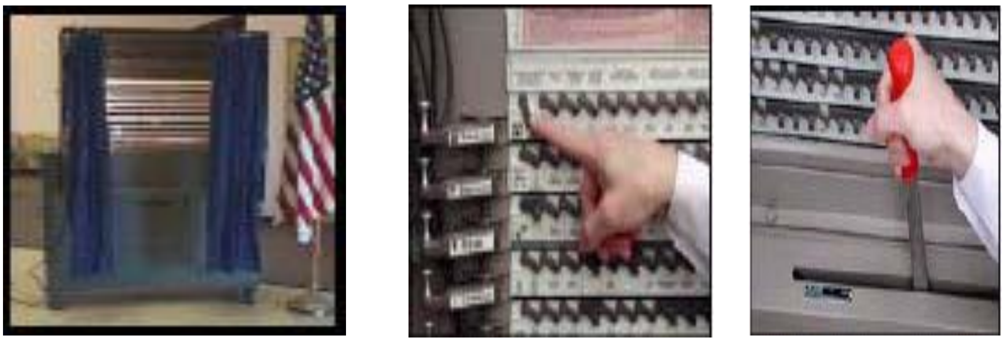

Figure 2.1: Mechanical lever machines

2.1.3 Punch card

In punch card voting systems shown in Figure 2.2, voters punch holes in cards to indicate their candidate or issue choice. The cards are printed with numbers (with the list of candidates and issue choices printed separately in a book), or the candidate names and issue choices may be printed directly on the ballot next to the location of the hole to be punched. The ballot is either dropped in a ballot box or fed into a computerized tabulating device. The first punch cards and computerized tally machines were used in Georgia in 1964 [1].

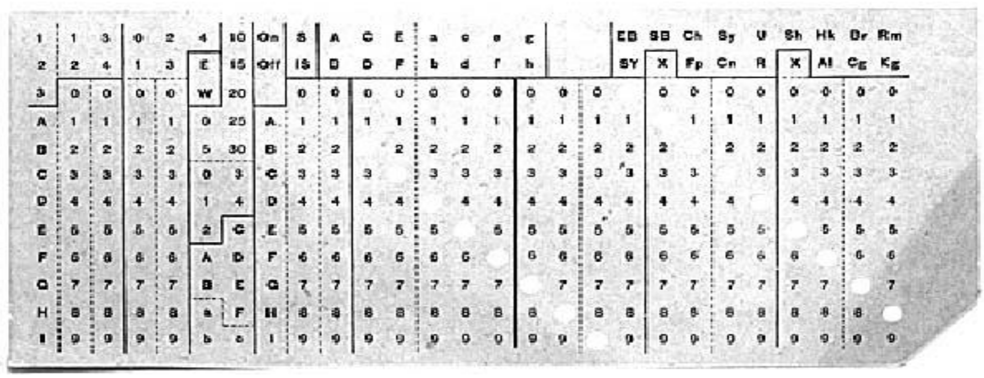

Figure 2.2: Punch card ballot

2.1.4 Mark sense

Mark sense voting systems allow voters to record their choices by filling in a circle, rectangle or oval, or by completing an arrow on a ballot card with candidate names and issue choices printed on it. Ballot cards are then dropped in a ballot box or fed into a computerized tabulating device. This device selects the darkest mark in a group as the vote using "dark mark logic". Mark sense technology, often referred to as "optical scan". has also been used for applications such as standardized testing [1].

2.1.5 Direct recording electronic

Direct Recording Electronic (DRE) voting systems shown in Figure 2.3, are an electronic implementation of mechanical lever voting systems. As in lever voting systems, there is no ballot and the choices are visible to the voter on the front of the machine. Voters use touch-screens, push-buttons, possibly keyboards or other devices to enter their choices into electronic storage such as smart cards, diskettes, or memory cartridges. Choices are added to the choices of all other voters [1].

Figure 2.3: DRE machine

2.2 Microcontroller

Not too long ago, the work on the industry of the electronic circuit meant building a complex design, using hundreds of different components like resistor, capacitor, inductor, transistor and so on. The design of the electronic circuit was fixed, and every circuit was wired to do one specific application, and making changes or modify a small part of it required you to cut wires, welding, solder connections and more [2].

Due to technological development in the field of semiconductors and the invention of Integrated Circuits (IC), it became possible to develop a complete electronic circuit on a small chip size may not exceed a few millimeters or less. Which produced a special generation of electronic circuits called microcontroller. The microcontrollers can be described as a mini programmable computer to perform many function. Thus, electronic system industry has shifted from purely electronic design to software commands. Software is easier to modify than hardware. With a few key presses, you can radically change the logic of a device [2].

Microcontroller is a general-purpose device, but one which is meant to fetch data, perform limited calculations on that data, and control its environment based on those calculations. The prime use of microcontroller is to control the operation of a machine using a fixed program that is stored in read only memory Read Only Memory (ROM) and that does not change over the lifetime of a system [3] .

2.2.1 Architecture of microcontroller

The block diagram of a typical microcontroller is shown in Figure 2.4, which is true computer on a chip. The design incorporates all of the features found in microprocessor control process unit (CPU): Arithmetic Logic Unit (ALU), Program Counter (PC), Stack Pointer (SP) and registers. It also has added the other features needed to make complete computer: ROM, Random Access Memory (RAM), parallel Input/Output I/O, serial I/O, counter and a clock circuit [4].

Microcontrollers are sometimes called embedded microcontrollers, which just means that they are part of a larger device or system, unlike a general purpose computer, which also includes all of these components, a microcontroller is designed for a specific task to control a particular system [3].

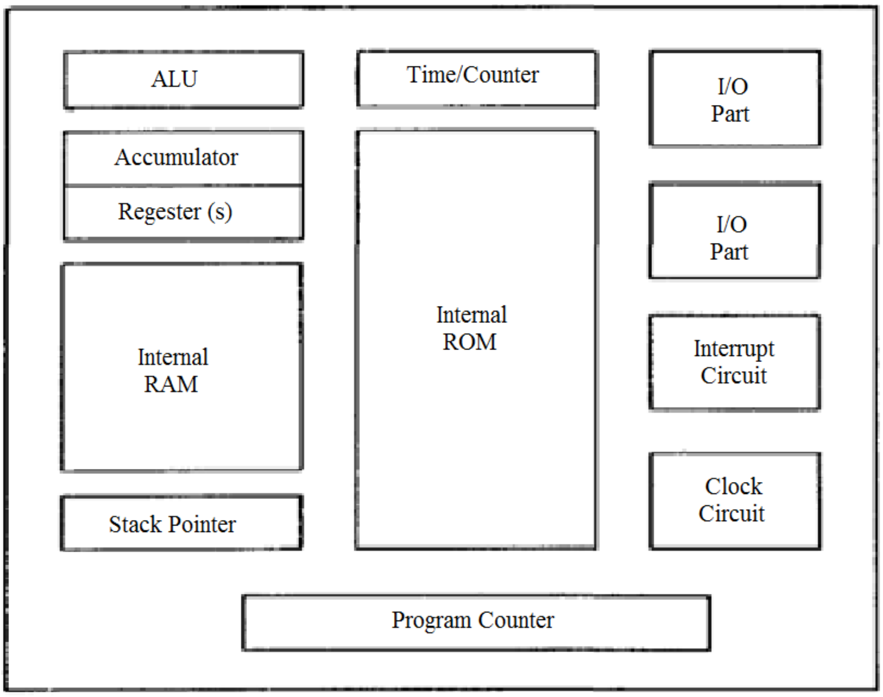

Figure 2.4: Microcontroller architecture

Even at a time when Intel presented the first microprocessor with the 4004 there was already a demand for microcontrollers The contemporary TMS1802 from Texas Instruments, designed for usage in calculators, was by the end of 1971 advertised for applications in cash registers, watches and measuring instruments. The TMS 1000, which was introduced in 1974, already included RAM, ROM, and I/O on-chip and can be seen as one of the first microcontrollers, even though it was called a microcomputer [5]. The first controllers to gain really widespread use were the Intel 8048, which was integrated into PC keyboards, and its successor, the Intel 8051, as well as the 68HCxx series of microcontrollers from Motorola

2.2.2 Components of microcontroller

Figure 2.5 shows the basic layout of microcontroller, the following list contains the modules typically found in a microcontroller [4].

i. Processor core: The CPU of the controller. It contains the arithmetic logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file, etc).

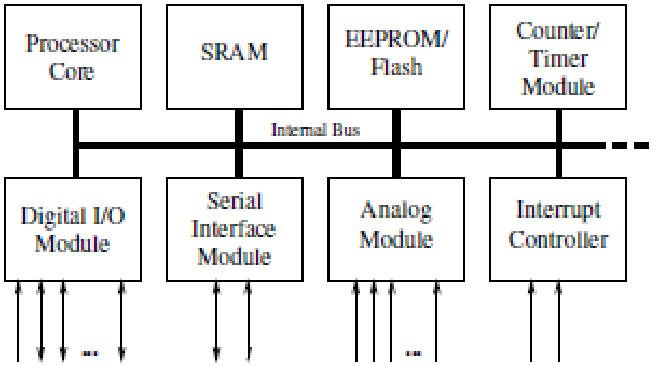

Figure 2.5: Basic layout of a microcontroller

ii. Memory: The memory is sometimes split into program memory and data memory. In larger controllers, a Direct Memory Access (DMA) controller handles data transfers between peripheral components and the memory.

iii. Interrupt controller: Interrupts are useful for interrupting the normal program flow in case of (important) external or internal events. In conjunction with sleep modes, they help to conserve power.

iv. Timer/Counter: Most controllers have at least one and more likely three timer/counters, which can be used to timestamp events, measure intervals, or count events. Many controllers also contain Pulse Width Modulation (PWM) outputs, which can be used to drive motors or for safe breaking Antilock Brake System (ABS). Furthermore, the PWM output can, in conjunction with an external filter, be used to realize a cheap digital/analog converter.

v. Digital I/O: Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 90, depending on the controller family and the controller type.

vi. Analog I/O: Apart from a few small controllers, most microcontrollers have integrated analog/digital converters, which differ in the number of channels (2-16) and their resolution (8-12 bits). The analog module also generally features an analog comparator. In some cases, the microcontroller includes digital/analog converters.

vii. Interfaces: Controllers generally have at least one serial interface which can be used to download the program and for communication with the development PC in general. Since serial interfaces can also be used to communicate with external peripheral devices, most controllers offer several and varied interfaces like Serial Peripheral Interface (SPI) and Serial Communications Interface (SCI).

viii. Watchdog timer: Since safety-critical systems form a major application area of microcontrollers, it is important to guard against errors in the program and/or the hardware. The watchdog timer is used to reset the controller in case of software “crashes”.

All components are connected via an internal bus and are all integrated on one chip. The modules are connected to the outside world via I/O pins [4].

2.2.3 Appliances of microcontroller

Today, microcontroller production counts are in the billions per year, and the controllers are integrate into many appliances we have grown used to, like:

• Household appliances (microwave, washing machine, coffee machine).

• Telecommunication (mobile phones).

• Automotive industry (fuel injection, ABS) etc.

• Aerospace industry.

• Industrial automation.

Microcontrollers are used in almost all type of electronic equipment, from coffeepots to laser printer. They are even incorporate in computer systems. They can be found in floppy drives, Compact Disk-Read Only Memory (CD-ROM) drivers, and video cards. A microcontroller is often embedded into the electronic circuit boards of these computer devices [5]. Microcontrollers are also used heavily in industry. They have taken the place of the relays, solid-state devices, and other discrete components. The microcontroller has one chip (built-in) peripheral devices. These one chip peripherals make it possible to have single chip microcomputer system [3]. There are few more advantages of built-in peripherals:

- Built-in peripherals have smaller access times hence speed is more.
- Hardware reduces due to single chip microcomputer system.
- Less hardware reduced Printed Circuit Board (PCB) size and increase reliability of the system.

2.2.4 Advantage of microcontroller

In addition to its small size, there are a number of advantages to using microcontrollers in industry. Some of the major advantages of microcontroller are that they are [3]:

i. Reusable: The typical microcontroller is programmable, which means it is reusable. This is especially advantageous for prototyping control circuitry. When developing a complex control system, it is not unusual for it to fail when first applied. As a matter of fact, a complex control project may need to be rewritten and/or rewired many times before it meets design expectations. The fact that the control circuit can be modified by programming rather than rewiring is very advantageous for fast project prototype development.

ii. Dependable: Integrated circuits, such as the microcontroller, are much more dependable than relays. Before microcontrollers, control circuitry relied on many electromechanical relays and timers to control the system. Relays depend on electromagnets to move armature and contact parts, so they eventually wear out due to mechanical friction. Relays are also susceptible to damage caused by dust, dirt, corrosion, rust, insects, and other contaminants that can interfere with the moving parts. Microcontrollers have no moving parts. This provides a much higher rate of reliability.

iii. Cost effective: Microcontrollers can be produced at lower costs than their electromechanical predecessors. Also, microcontrollers can be reprogrammed if the designed application does not work correctly or if the application for its use changes.

iv. Energy efficient: Because the majority of the circuitry is made from integrated circuits, the energy cost of using a microcontroller is much less than if using individual components of a relay-type logic circuit. Relay logic uses numerous relays wired in series and parallel to form control circuit conditions similar in function to logic gates. A microcontroller consumes less electrical energy than conventional electromechanical devices [3].

2.2.5 Disadvantage of microcontroller

There are a few disadvantages to using microcontrollers. The two most prominent disadvantages are:

i. Programming complexity: Special skills are required to program the microcontrollers. This requires a higher level of training for some personnel. In addition, there are many different programming languages to choose from. This can lead to a compatibility problem when attempting to merge two dissimilar systems into one control system.

ii. Electrostatic sensitivity: most microcontrollers are composed of Complementary Metal-Oxide Semiconductor (CMOS) integrated circuitry. CMOS can be damaged easily by a static charge [3].

2.3 Arduino

In 2005 Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis came up with an idea, aiming to provide an device. The device needed to be simple, easy to connect to various things (such as relays, motors, and sensors), and easy to program. It also needed to be inexpensive. They selected the AVR family of 8-bit microcontroller (MCU or µC) devices from Atmel and designed a self-contained circuit board with easy to use connections, wrote bootloader firmware for the microcontroller, and integrated it all into a simple development environment that used programs called “sketches” . The result was the Arduino.

Since then the Arduino has grown in several different directions, some versions getting smaller than the original, and some getting larger. The common element among all of them is the Arduino run-time library that is supplied with the Arduino development environment, and the on-board bootloader firmware that comes preloaded on the microcontroller of every Arduino board.

2.3.1 Definition of Arduino

Arduino is an open source physical computing platform based on a simple I/O board and a development environment that implements the processing language [2]. Arduino is composed of two major parts there are:

i- The Arduino hardware:

The Arduino board is a small microcontroller board, which is a small circuit (the board) that contains a whole computer on a small chip (the microcontroller).

ii- The Arduino (IDE):

The Integrated Development Environment (IDE) is a special program running on the computer that allows you to write sketches (a little computer program) for the ARDUINO board in a simple language, the sketch tells the board what to do [2].

2.3.2 Types of Arduino

Over the years the designers have developed a number of board designs. The newer versions have more advanced processors with more memory and enhanced I/O features, but for the most part they use the same pin-out arrangements and will work with existing add-on boards called shields, and various add-on components such as sensors, relays, and actuators. The newer versions of the Arduino will also run most of the sketches created for older models, but sketches written for the latest versions may, or may not, work with older models.

i. Arduino Uno

The Uno is most common version of ARDUINO and the one labeled as the classic ARDUINO. It is compatible with most available ARDUINO shields.

ii. The Arduino Due

The Arduino Due is the second iteration of the classic Arduino and offers more features for advanced users. The Due's processor is faster, has more memory, and more I/O ports. It does not support many shields. Because of the faster CPU, the Arduino Due runs on a lower voltage 3.3V over the Uno's 5V. This means it cannot always support the same devices.

iii. The Arduino Mega

The Mega is the second most commonly encountered version of the Arduino family. It uses the same 5V power supply as the Uno, so many of the Arduino shields are also compatible with the Mega. The Arduino Mega comes in two types:

- Arduino Mega 2560

The Arduino Mega boasts 256kB of memory (8 times more than the Uno). It also had 54 input and output pins, 16 of which are analog pins, and 14 of which can do PWM. However, all of the added functionality comes at the cost of a slightly larger circuit board.

- Arduino Mega ADK

The ADK is similar to the 2560; it also has a programmable Universal Serial Bus (USB) host chip installed. This specialized version of the Arduino is basically has been specifically designed for interfacing with Android smartphones.

iv. Arduino Leonardo

The Leonardo is not a common board, but has similar features to the Uno, including the 5V power supply and the processing power. It is a good board for those who need more input and output ports than the Arduino Uno, but do not need the horsepower or size of the Due.

v. Arduino NG, Diecimila, and the Duemilanove

Legacy versions of the Arduino Uno consist of the NG, Diecimila, and the Duemilanove. The important thing to note about legacy boards is that they lack particular feature of the Arduino Uno. Some key differences:

a- The Diecimila and NG use an ATMEGA168 chips (as opposed to the more powerful ATMEGA328).

b- Both the Diecimila and NG have a jumper next to the USB port and require manual selection of either USB or battery power.

c- The Arduino NG requires that you hold the rest button on the board for a few seconds prior to uploading a program.

Table 2.1 shows features of each Arduino type in processor, memory, digital I/O and analogue I/O

Table 2.1: Features of each Arduino type

Features	Arduino Uno	Arduino Due	Arduino Mega	Arduino Leonardo
Processor	16MHz ATmega328	84MHz AT91SAM3X8E	16MHz ATmega2560	16MHz ATmega32u4
Memory	2KB SRAM, 32KB flash	96KB SRAM, 512KB flash	8KB SRAM, 256KB flash	2.5KB SRAM, 32KB flash
Digital I/O	14	54	54	20
Analogue I/O	6 input, 0 output	12 input, 2 output	16 input, 0 output	12 nput, 0 output

2.2.3 Advantage of Arduino over other microcontrollers

i. Ready to use: The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a micro-controller, serial communication interface, Light Emitter Diode (LED) and headers for the connections. You don't have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that's it.

ii. The Arduino can be programmed by a USB cable, not a serial port. This feature is useful, because many modern computers don’t have serial ports.

iii. Examples of codes: Another big advantage of Arduino is its library of examples present inside the software of Arduino.

iv. The Arduino is a multiplatform environment; it can run on Windows, Macintosh, and Linux.

v. The Arduino is an open source hardware and software

2.3.4 Disadvantage of Arduino

On the other hand, the size of the structure of the Arduino consider as disadvantage ,unlike the other micro controller, which allow to make the size of the projects as small as possible. Also the higher cost of the Arduino than other microcontroller is another disadvantage.

CHAPTER THREE

SYSTEM HARDWARE AND SOFTWARE CONSIDERATIONS

3.1 Introduction

There are two conceptually separate parts of system modules the hardware which is the body of the system, and the software which is used to control the system. The hardware consists of a number of relays, pushbuttons, and a Liquid Crystal Display (LCD) to display the result. The software consists of two parts; the code which is written and installed in microcontroller and simulation which describes the system behavior in the ideal state.

3.2 System Hardware Consideration

The practical circuit of this thesis consists of several electronic component that integrated with each other to complete the voting process and display the results of the process. This chapter shows these components with the explanation of each component. Figure 3.1 shows the block diagram of the system.

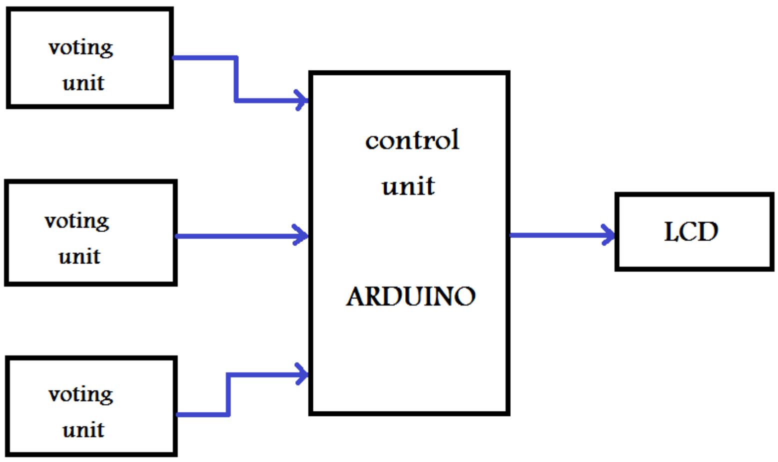

Figure 3.1: Block diagram of the system

3.2.1 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 microcontroller operating at 5V with 2KB of RAM, 32 KB of flash memory for storing programs and 1 Kb of Electrically Erasable Programmable Read Only Memory (EEPROM) for storing parameters. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button, as show in Figure 3.2. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

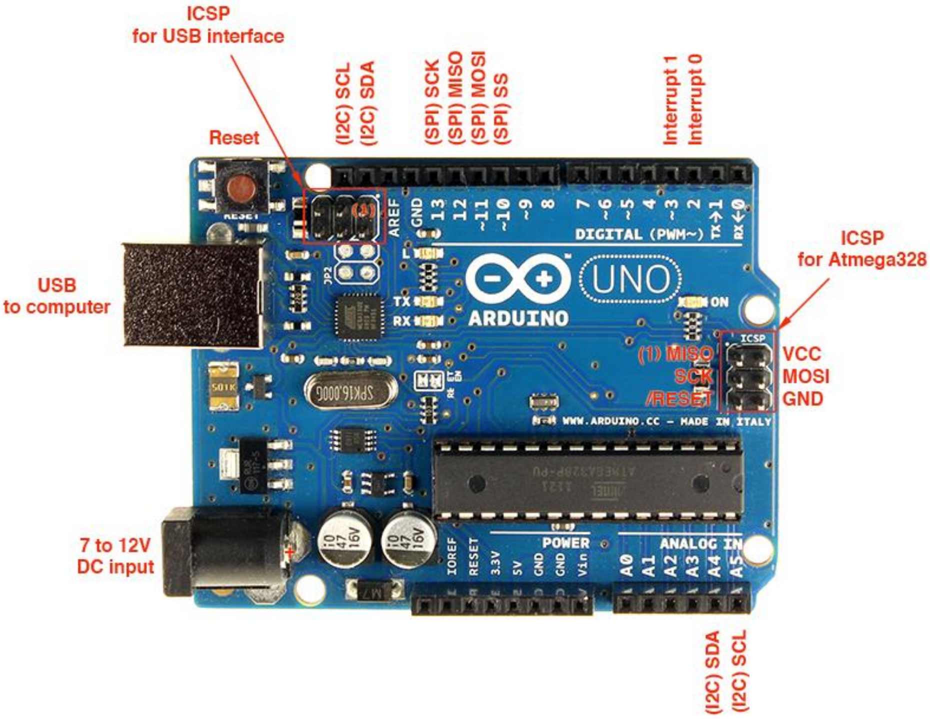

Figure 3.2 : ARDUINO UNO

In this thesis Arduino Uno revision 3 (Arduino Uno R3) will be used, which added SDA, SCL and IOREF pins to the previous versions.

- Arduino Uno R3 specifications:

Table 3.1 shows the specification of the Arduino Uno R3.

- Power supply:

The Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter or battery. Leads from a battery can be inserted in the GND and Vin pin headers of the power connector.

Table 3.1: specification of Arduino Uno R3

Microcontroller	ATmega328
Operating voltage	5 V
Recommended Input Voltage	7 – 12 V
Input Voltage Limits	6 – 20 V
Digital I/O Pins	14– 6 of which can be PWM
Analog Input Pins	6
Analog Input Pins	6
Maximum DC Current per I/O pin at 5VDC	40 ma
Maximum DC Current per I/I pin at 3.3 VDC	50 ma
Flash Memory	32 kB
SRAM Memory	2 kB
EEPROM	1 kB
Clock Speed	16 MHz

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. Thus, the recommended range is 7 to 12 volts [6].

- Power pins

a. Vin: The input voltage ,when the Uno board using an external power source we can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
b. 5V: This pin outputs a regulated 5V from the regulator on the board. Supplying voltage via the 5V or 3.3V pins can damage the board.
c. 3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.
d. GND: Ground pins.
e. IOREF: This pin on the Uno board provides the voltage reference with which the microcontroller operates.

- Input and output pins

Each of the 14 digital pins on the Uno can be used as an input or output, by using software commands as (pinmode(), digitalwrite(), and digitalread() functions). Each pin operate at 5 volts and it can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller [6]. In addition, there are some pins have specialized functions:

a. Serial: 0 (RX) and 1 (TX) pins, (RX) pin used to receive Transistor-Transistor Logic (TTL) serial data from host computer to Arduino, and (TX) pin used to transmit TTL serial data from the Arduino to another device. These pins are connected to the corresponding pins of the ATmega16U2 USB-to-TTL. These pins are also internally connected via 1K resistors to the USB chip, so that data from the USB interface can be sent/received to pins 0 and 1.
b. External interrupts: pin2 and pin3, these pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
c. PWM: The pins 3, 5, 6, 9, 10, and 11, used to create a square wave with a specific duty cycle, Provide 8-bit PWM output with the analogwrite() function.

d. SPI: the pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK). A serial protocol that uses a clock line (SCK) along with Master Out Slave In (MOSI) while the data going from the master to the slave ,and Master In Slave out (MISO) while the data going from the slave to the master, to transfer information. Also commonly use a slave select (SS) line to select one slave out of multiple ones.
e. Analogue to Digital (A/D): This input converts an analogue voltage in to a digital representation.
f. LED: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
g. I2C: A4 or SDA pin and A5 or SCL pin, is another serial communications method using a bidirectional data.
h. AREF: Reference voltage for the analog inputs. Used with analogReference().
i. The In Circuit Serial Programming (ICSP): Another way to programming either the main Atmega328 processor (most common) or the Atmega16U2 USB interface processor (only if required).
j. Reset: Bring this line LOW to reset the microcontroller.

3.2.2 LCD display

LCD draws its definition from its name itself. It is combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. LCDs are economical, easily programmable, have no limitation of displaying special and even custom characters unlike in seven segments. LCD is composed of several layers which include two polarized panel filters and electrodes. Light is projected from a lens on a layer of liquid crystal. This combination of colored light with the gray-scale image of the crystal (formed as electric flows through the crystal) forms the colored image. This image is then displayed on the screen [7].

The principle behind the LCD’s is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light, which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD. Thus that particular area will become dark compared to other. The LCD works on the principle of blocking light. While constructing the LCD's, a reflected mirror is arranged at the back. An electrode plane is made of indium-tin oxide which is kept on top and a polarized glass with a polarizing film which is also added on the bottom of the device. The complete region of the LCD has to be enclosed by a common electrode and above it should be the liquid crystal matter. Next comes to the second piece of glass with an electrode in the form of the rectangle on the bottom and on top is another polarizing film. It must be considered that both of the pieces are kept at right angles. When there is no current, the light passes through the front of the LCD it will be reflected by the mirror and bounced back. As the electrode is connected to a battery the current from it will cause the liquid crystals between the common-plane electrode and the electrode shaped like a rectangle to untwist. Thus the light is blocked from passing through. That particular rectangular area appears blank; all these steps are shown in Figure 3.3 [7].

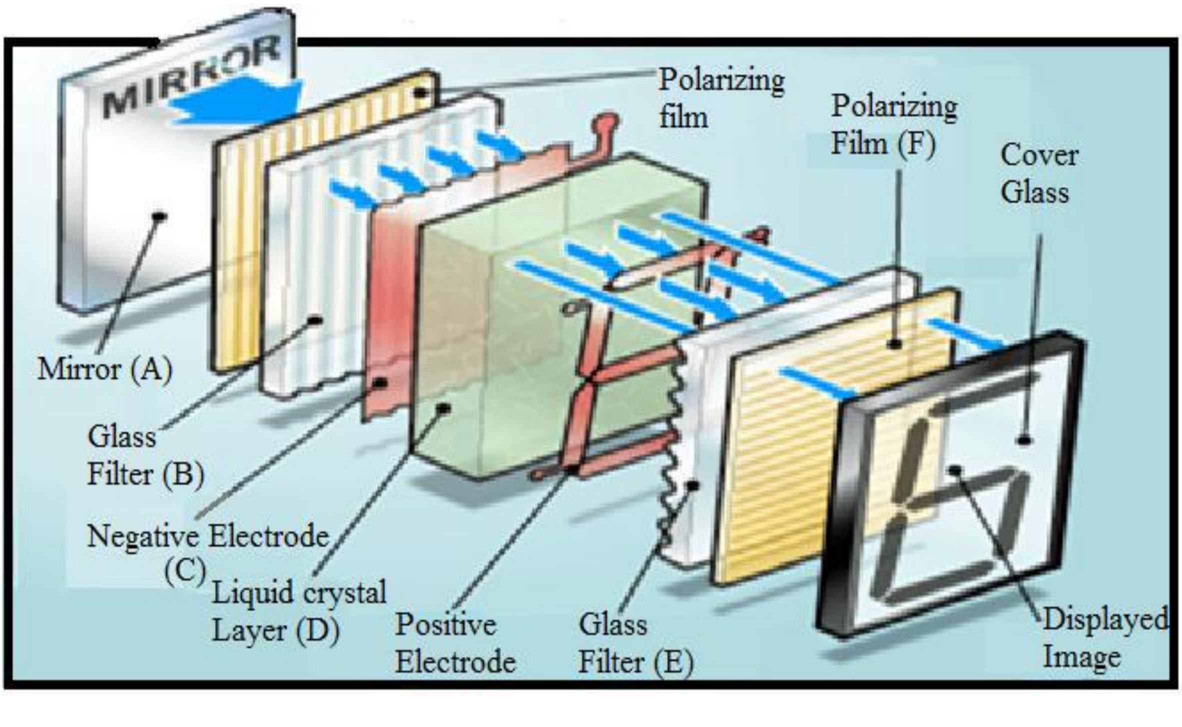

Figure 3.3: LCD layer diagram

A 16x2 LCD shown in Figure 3.4 is a very basic module and is very common in various devices and circuits. These modules are preferred over seven segment displays and other multi segment LEDs. There are many advantages when compared to seven segment displays. They are: LCDs can display characters, numbers and even graphics [8]. 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. Table 3.2 shows the pin of LCD 16x2 and the function of each pin.

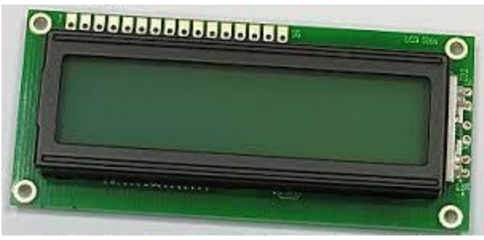

Figure 3.4: 16x2 LCD

Table 3.2: LCD 16x2 pins description

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply Voltage (5V)	Vcc
3	Contrast Adjustment through a Variable Resistor	Vee
4	Selects Command Register when Low, Data Register 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 a High to Low Pulse is Given	Enable
7-14	8-bit data pins	DB0-DB7
15	Backlight Vcc (5V)	LED+
16	Backlight Ground (0V)	LED-

This LCD has two registers:

i. 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.

ii. 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.

3.2.3 Relay

A relay is usually an electromechanical device that is actuated by an electrical current. The current flowing in one circuit causes the opening or closing of another circuit. Relays are like remote control switches and are used in many applications because of their relative simplicity, long life, and proven high reliability. It takes a relatively small amount of power to turn on a relay but the relay can control something that draws much more power.

- **Parts of relay**

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core, a movable iron armature, and one or more sets of contacts.

- **Types of relay**

There are two basic classifications of relays:

i. Electromechanical relay:

The electromechanical relay have moving parts, this relay can be classified into:

a- General purpose relay: Most versions of the general-purpose relay have one to eight poles and can be single or double throw. These are found in computers, copy machines, and other consumer electronic equipment and appliances.

b- Power relay: The power relay is capable of handling larger power loads 10-50 amperes or more. They are usually single-pole or double-pole units.

c- Contactor: A special type of high power relay, it’s used mainly to control high voltages and currents in industrial electrical applications.

Because of these high power requirements, contactors always have double-make contacts.

d- Time-delay relay: The contacts might not open or close until sometime interval after the coil has been energized. This is called delay-on-operate. Delay-on-release means that the contacts will remain in their actuated position until some interval after the power has been removed from the coil. A third delay is called interval timing. Contacts revert to their alternate position at a specific interval of time after the coil has been energized. The timing of these actions may be a fixed parameter of the relay, or remotely adjusted through an external circuit.

Electromechanical relays have some advantages include lower cost, no heat sink is required, multiple poles are available and they can switch AC or DC with equal ease.

ii- Solid state relay:

The solid state relays have no moving parts. These active semiconductor devices use light instead of magnetism to actuate a switch. The light comes from an LED, or light emitting diode. When control power is applied to the device's output, the light is turned on and shines across an open space. On the load side of this space, a part of the device senses the presence of the light, and triggers a solid state switch that either opens or closes the circuit under control. Often, solid state relays are used where the circuit under control must be protected from the introduction of electrical noises. Solid state relays have some advantages include low EMI/RFI, long life, no moving parts, no contact bounce, and fast response. While the drawback to using a solid state relay is that it can only accomplish single pole switching.

- **Principle of the relays work**

All relays contain an electric coil, which is powered by AC or DC current. When the applied current or voltage exceeds a threshold value, the coil activates the armature, which operates either to close the open contacts or to open the closed contacts. When a power is supplied to the coil, it generates a magnetic force that actuates the switch mechanism. The magnetic force is, in effect, relaying the action from one circuit to another.

The first circuit is called the control circuit while the second is called the load circuit. Usually a spring will pull the switch open again once the power is removed from the coil.

3.3 System Software Consideration

The system is programmed by C language, which it is a family of general-purpose, high-level programming languages where its design emphasizes ease of use. The code for this system is organized as a number of functions that are defined within the modules which they are related to. Ideally, this system could be entirely contained within a Proteus simulation.

3.3.1 System code

The code that is used to programmed the Arduino is written by the C language in the Arduino 1.6.6 program.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
const int yes1 = 2;
const int yes2 = 3;
const int yes3 = 4;
const int no1 = A0;
const int no2 = A1;
const int no3 = A2;
const int nu1 = A3;
const int nu2 = A4;
const int nu3 = A5;
int Y1 = 0;
int Y2 = 0;
int Y3 = 0;
int N1 = 0;
int N2 = 0;
int N3 = 0;
int NUT1= 0;
int NUT2 = 0;
int NUT3 = 0;
```

```
void setup()
{
pinMode(yes1, INPUT);
pinMode(yes2, INPUT);
pinMode(yes3, INPUT);
pinMode(no1, INPUT);
pinMode(no2, INPUT);
pinMode(no3, INPUT);
pinMode(nu1, INPUT);
pinMode(nu2, INPUT);
pinMode(nu3, INPUT);
lcd.begin(16, 2);
Serial.begin(9600);
lcd.setCursor(3, 0);
lcd.print("Electronic");
lcd.setCursor(0,1);
lcd.print("Voting Machine");
delay(2000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Click on your ");
lcd.setCursor(0,1);
lcd.print("choice");
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("wait for 5 sec");
delay(3000);
lcd.clear();}
void loop()
Y1 = digitalRead(yes1);
Y2 = digitalRead(yes2);
Y3 = digitalRead(yes3);
N1 = digitalRead(no1);
N2 = digitalRead(no2);
```

```
N3 = digitalRead(no3);
NUT1 = digitalRead(nu1);
NUT2 = digitalRead(nu2);
NUT3 = digitalRead(nu3);
if ((Y1 == HIGH)&& (Y2 == LOW)  && (Y3 == LOW)) {
lcd.setCursor(0,0);
lcd.print("yes 1 ");
delay(200);}
else if ((Y1 == LOW) && (Y2 == HIGH)  && (Y3 == LOW)) {
lcd.setCursor(0,0);
lcd.print("yes 1 ");}
else if ((Y1== LOW) && (Y2== LOW)  && (Y3== HIGH)) {
lcd.setCursor(0,0);
lcd.print("yes 1 ");}
else if ((Y1== HIGH) && (Y2== HIGH)  && (Y3== LOW)) {
lcd.setCursor(0,0);
lcd.print("yes 2 ");}
else if ((Y1== LOW) && (Y2== HIGH)  && (Y3== HIGH)) {
lcd.setCursor(0,0);
lcd.print("yes 2 ");}
else if ((Y1== HIGH) && (Y2== LOW)  && (Y3== HIGH)) {
lcd.setCursor(0,0);
lcd.print("yes 2 ");}
else if ((Y1== HIGH) && (Y2== HIGH)  && (Y3== HIGH)) {
lcd.setCursor(0,0)     }
else if ((Y1== LOW) && (Y2== LOW)  && (Y3== LOW)) {
lcd.setCursor(0,0);
lcd.print("yes 0 ");}
if ((N1 == HIGH)&& (N2 == LOW)  && (N3 == LOW)) {
lcd.setCursor(9,0);
lcd.print("No 1 ");
delay(200)    }
else if ((N1 == LOW) && (N2 == HIGH)  && (N3 == LOW)) {
lcd.setCursor(9,0);
lcd.print("No 1 ");}
```

```
else if ((N1== LOW) && (N2== LOW)  && (N3== HIGH)) {
lcd.setCursor(9,0);
lcd.print("No 1 ");}
else if ((N1== HIGH) && (N2== HIGH)  && (N3== LOW)) {
lcd.setCursor(9,0);
lcd.print("No 2 ");}
else if ((N1== LOW) && (N2== HIGH)  && (N3== HIGH)) {
lcd.setCursor(9,0);
lcd.print("No 2 ") }
else if ((N1== HIGH) && (N2== LOW)  && (N3== HIGH)) {
lcd.setCursor(9,0);
lcd.print("No 2 ");  }
else if ((N1== HIGH) && (N2== HIGH)  && (N3== HIGH)) {
lcd.setCursor(9,0);
lcd.print("No 3 "); }
else if ((N1== LOW) && (N2== LOW)  && (N3== LOW)) {
lcd.setCursor(9,0);
lcd.print("No 0 ");}
if ((NUT1 == HIGH)&& (NUT2 == LOW)  && (NUT3 == LOW)) {
lcd.setCursor(0,1);
lcd.print("Nut 1 ");
delay(200);}
else if ((NUT1 == LOW) && (NUT2 == HIGH)  && (NUT3 == LOW)) {
lcd.setCursor(0,1);
lcd.print("Nut 1 ");   }
else if ((NUT1== LOW) && (NUT2== LOW)  && (NUT3== HIGH)) {
lcd.setCursor(0,1);
lcd.print("Nut 1 "); }
else if ((NUT1== HIGH) && (NUT2== HIGH)  && (NUT3== LOW)) {
lcd.setCursor(0,1);
lcd.print("Nut 2 "); }
else if ((NUT1== LOW) && (NUT2== HIGH)  && (NUT3== HIGH)) {
lcd.setCursor(0,1);
lcd.print("Nut 2 ");   }
else if ((NUT1== HIGH) && (NUT2== LOW)  && (NUT3== HIGH)) {
```

```
lcd.setCursor(0,1);
lcd.print("Nut 2 ");  }
else if ((NUT1== HIGH) && (NUT2== HIGH)  && (NUT3== HIGH)) {
lcd.setCursor(0,1);
lcd.print("Nut 3 "); }
else if ((NUT1== LOW) && (NUT2== LOW)  && (NUT3== LOW)) {
lcd.setCursor(0,1);
lcd.print("Nut 0 ");
    }
}
```

3.3.2 System simulation

The computer simulation is very useful part, it allows the engineer to test the systems and observe their behavior before it is built in the real situation. voting circle for each person consists of three relays and three pushbuttons to represent the three options yes, no and neutral. The three relays connected with each other in way that the voting in one of the options lead to the separation of the two other options completely, and also that allow to vote on each option only once. The voting circle for one person is represented in Proteus as shown in Figure 3.5.

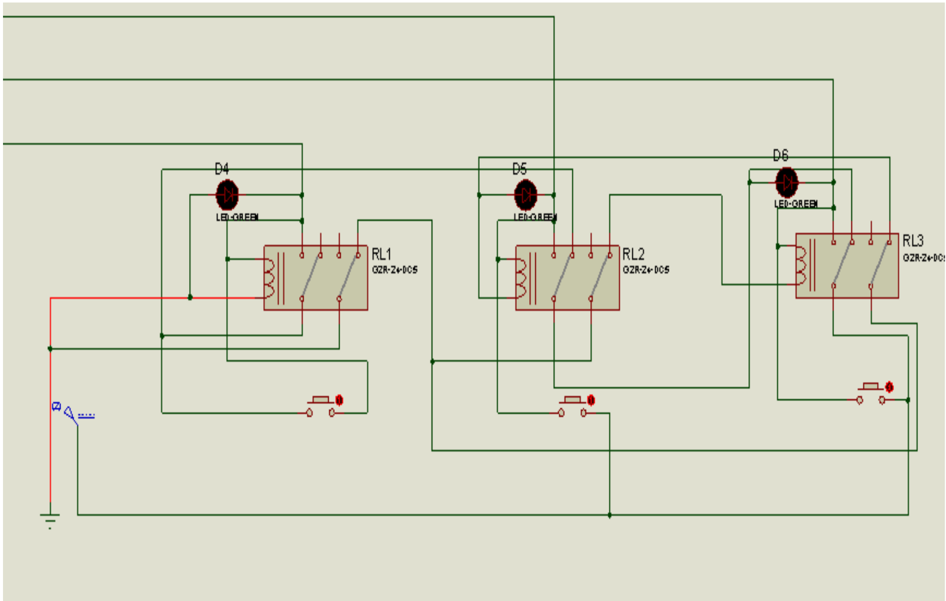

Figure 3.5: Voting circle for one person

LCD 16*2 was connected with the Arduino to display the result of the voting process as shown in Figure 3.6.

The code is installed in ARDUINO by double click on mouse and selected the code as in Figure 3.7.

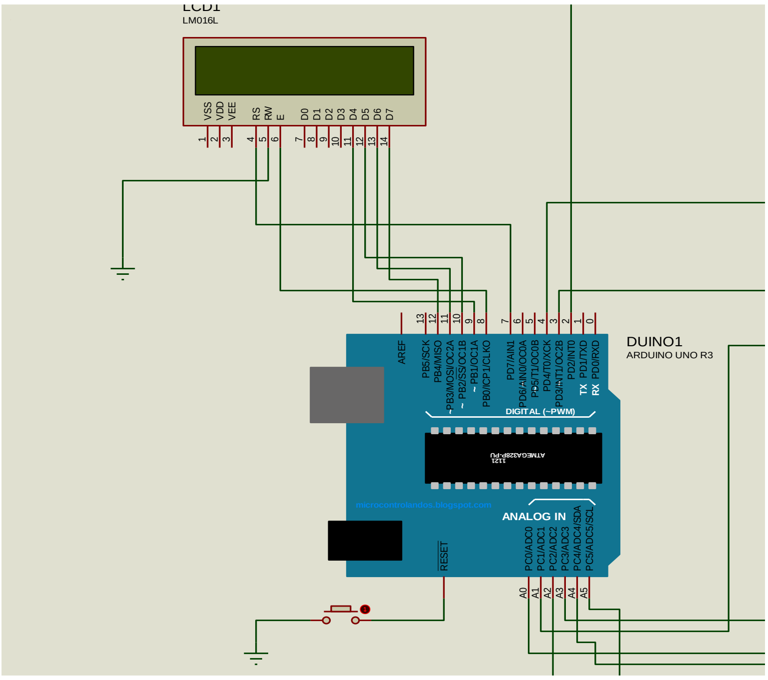

Figure 3.6**:** Connect the LCD with Arduino

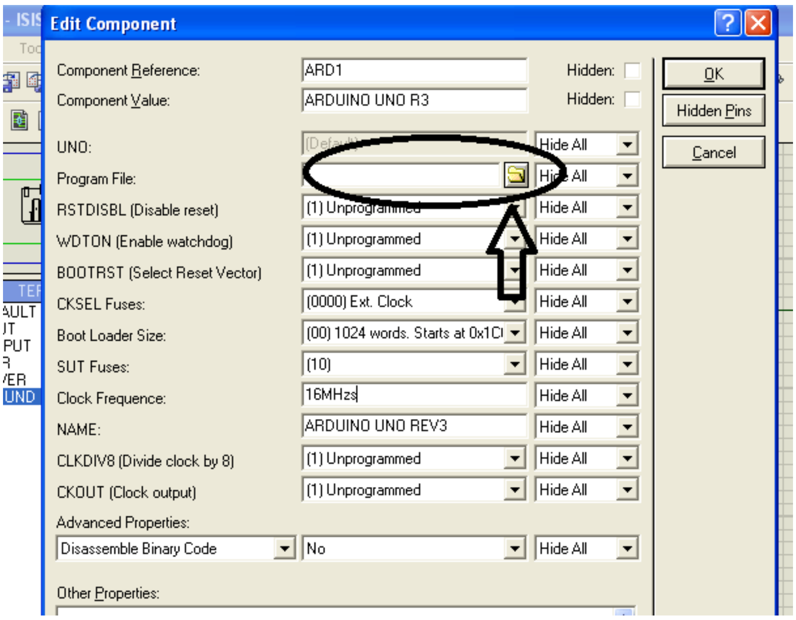

Figure 3.7**:** Install code in the arduino

Finally, Figure 3.8 shows the overall design of the simulation after connected the output of the voting circles and the LCD to the arduino, as well as download the program in the arduino. By clicking play button simulation will run ,the LCD directly display "electronic voting machine".

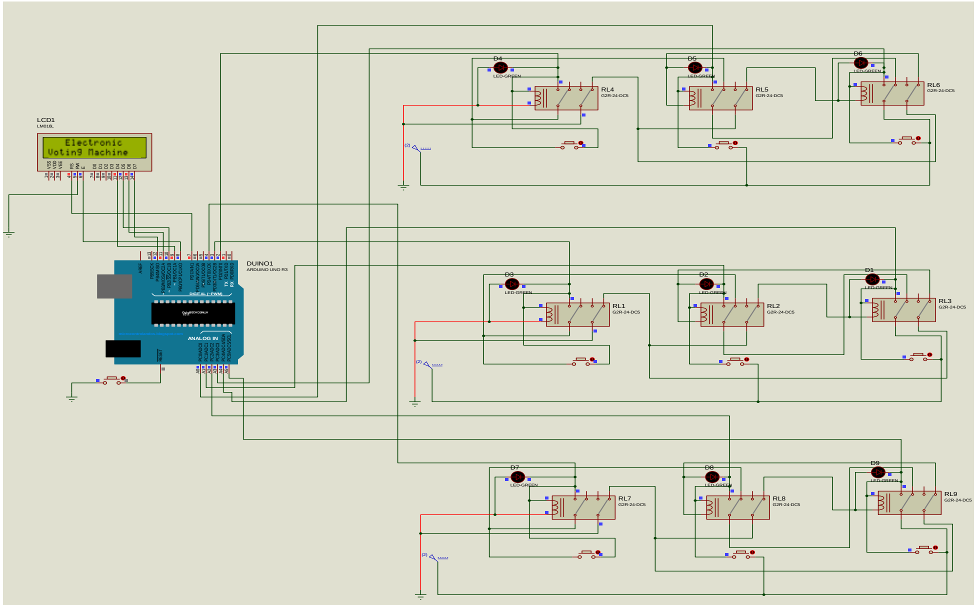

Figure 3.8: Schematic diagram of the system

Then, the system asks voters to voting through the message "click on your choice", requests to wait for a few seconds as shown in Figure 3.9, then the result will display on LCD

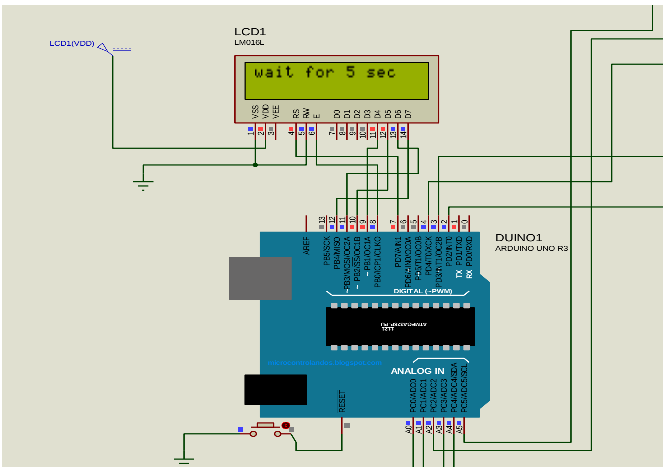

Figure 3.9: System wait voting

If we assume as an example, that each person of voters clicked on a different option, the result of the of the voting will appear on the screen as shown in Figure 3.10.

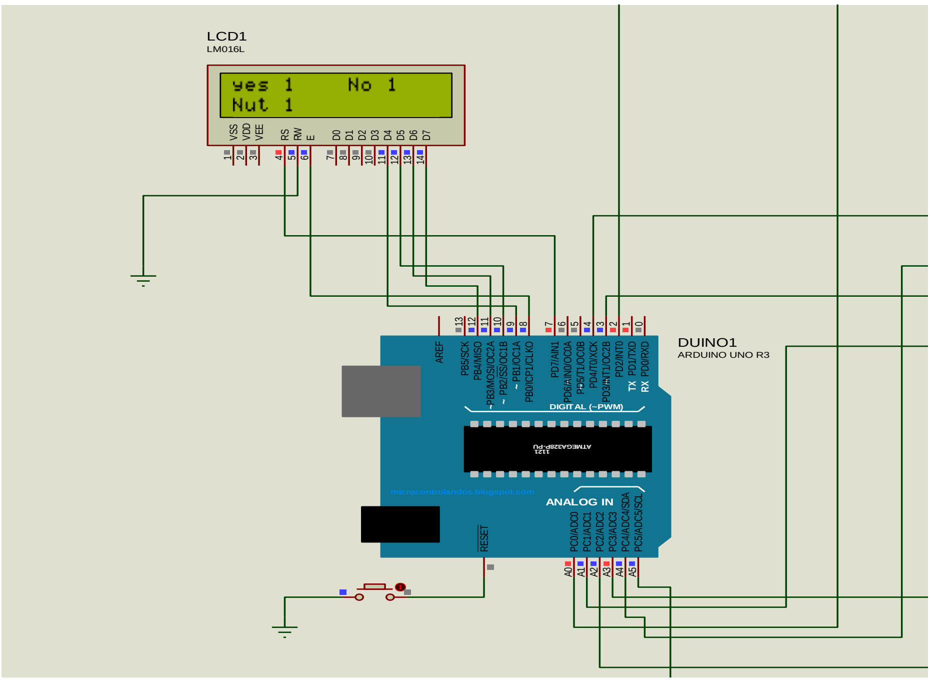

Figure 3.10: Tested the simulation

CHAPTER FOUR

SYSTEM IMPLEMENTAION AND TESTING

4.1 System Implementation

The voting circuits is designed in the real situation by connecting the three relays by the same way that described in the simulation. Three voting circuits are designed and connected with each other to be used by three person (voter1, voter2 and voter3) as shown in Figure 4.1.

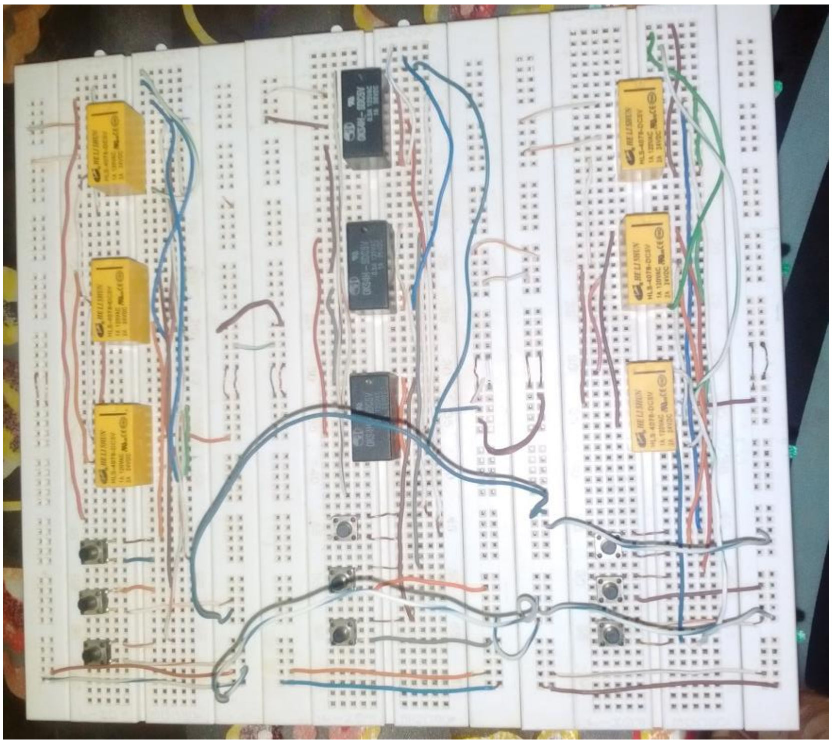

Figure 4.1: Voting circuits

This voting circles connected to the Arduino by connected the bush-button of "yes" signals of the voter1, voter2 and voter3 to the pins 2, 3 and 4 of the Arduino, bush-button of "no" signals are connected to the pins A0, A1 and A2 of the Arduino, while bush-button of "neutral" signals are connected to pins A3, A4 and A5 in the Arduino as shown in Figure 4.2.

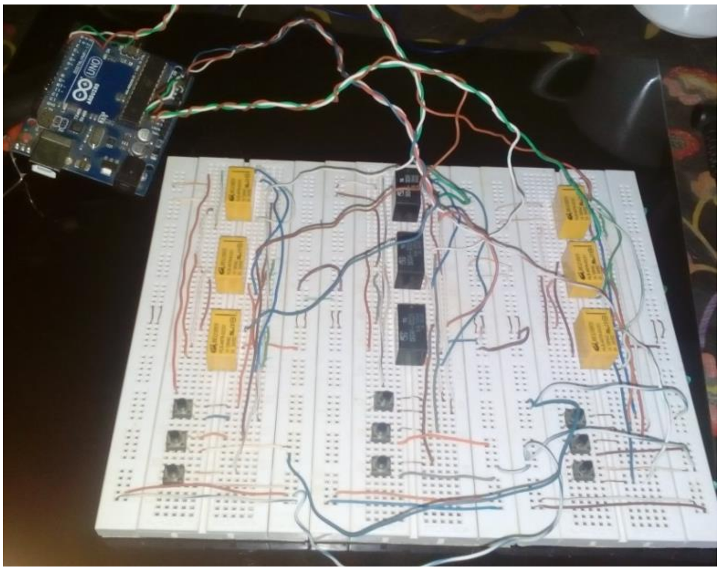

Figure 4.2: Voting circuits with Arduino

The LCD is connected with the Arduino by connecting pin0 and pin1 from LCD with GND and VCC in Arduino respectively, connecting pin3, pin4 and pin5 from the LCD to pin7, GND and pin8 in the Arduino, finally connecting the pins D4, D5, D6 and D7 from the LCD with the pins 9,10,11 and 12 in the Arduino respectively as shown in Figure 4.3. In addition to connecting a potentiometer with pin2 to adjust the intensity of lighting of the LCD.

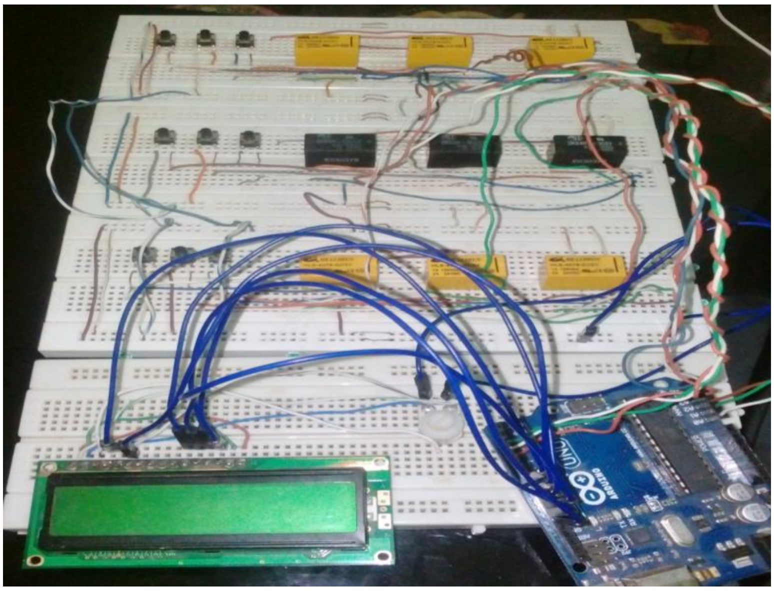

Figure 4.3: Connecting of LCD

The system is powered via an external power supply, this power supply can transform a 220V to 12V or 5V, the source 5V used to powered this system, as shown in Figure 4.4.

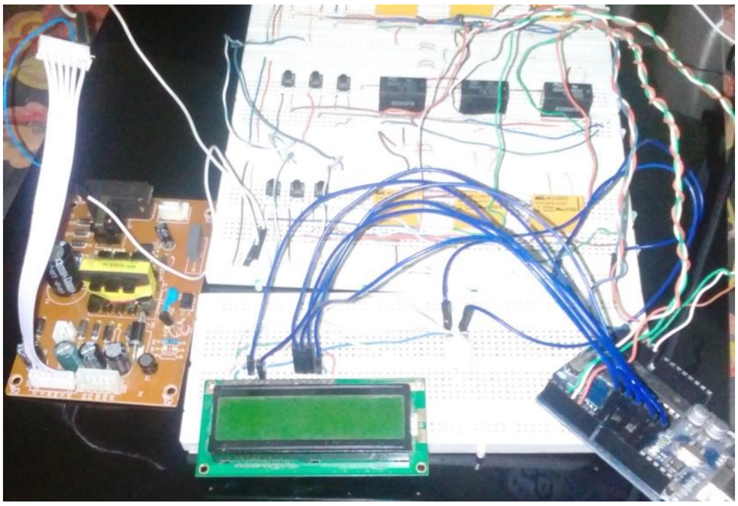

Figure 4.4 : Installing power supply

4.2 System Testing

When powered the system the LCD will display the message "electronic voting machine as shown in Figure 4.5.

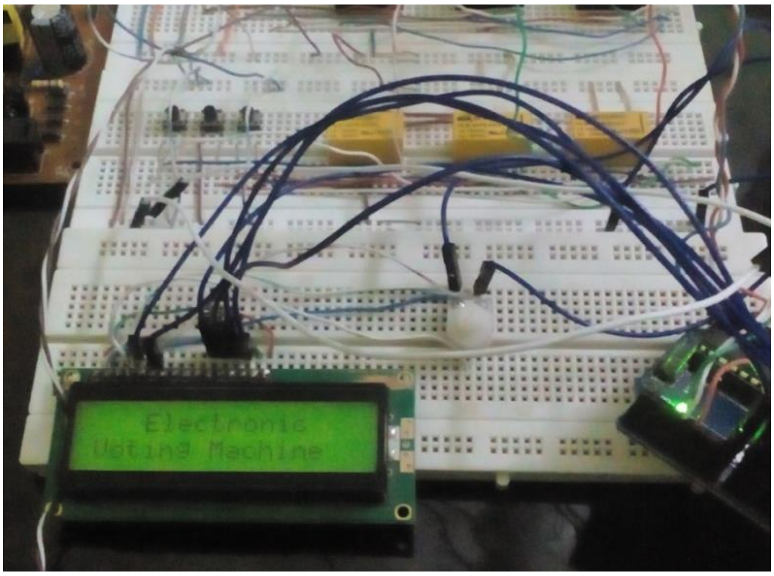

Figure 4.5: Circuit with power supply

Then the system display the message "click in your choice" on the LCD, after a few seconds the LCD display the message " wait 5 second "as shown in Figure 4.6. Then the system display the result on the LCD.

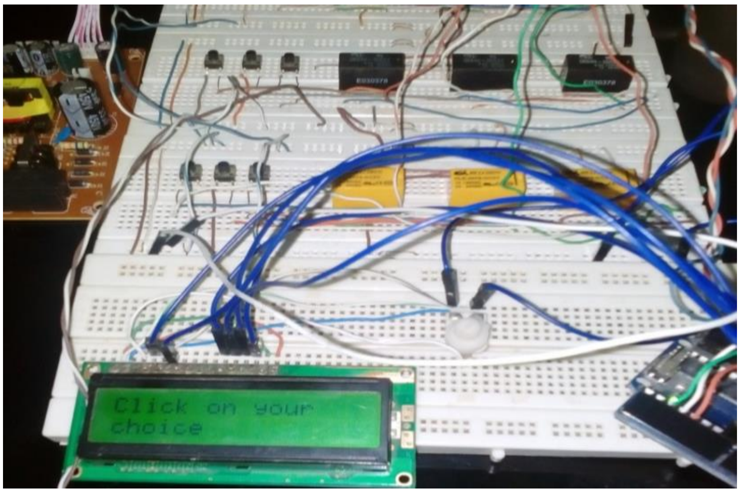

Figure 4.6: Choice entering

- **Cases of system operation**

- If no person do the voting process, Figure 4.7 shows the result of this case.

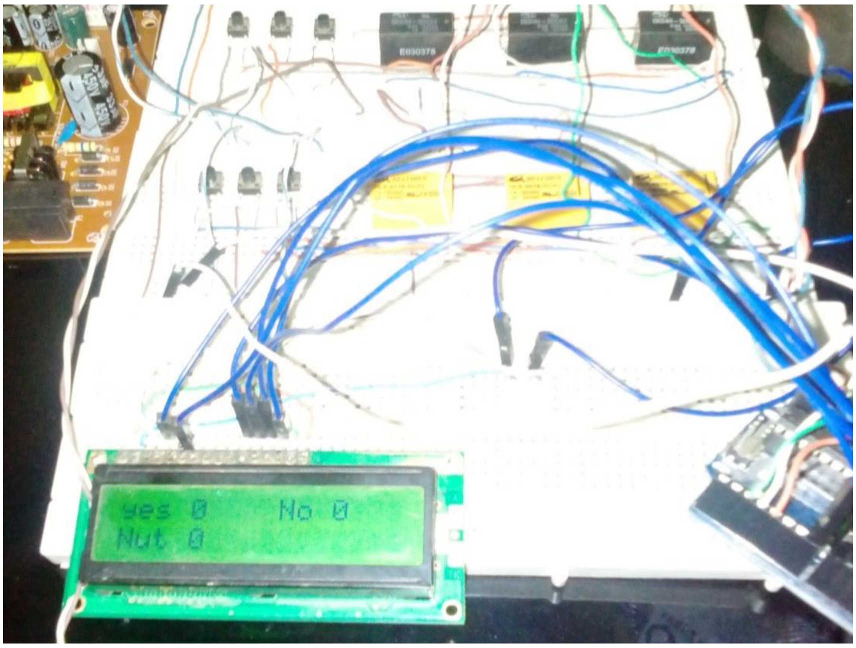

Figure 4.7: No voting case

- If a one from the voters voted "YES" , and the other voted "NO", and the third did not voting, Figure 4.8 shows the result of this case.

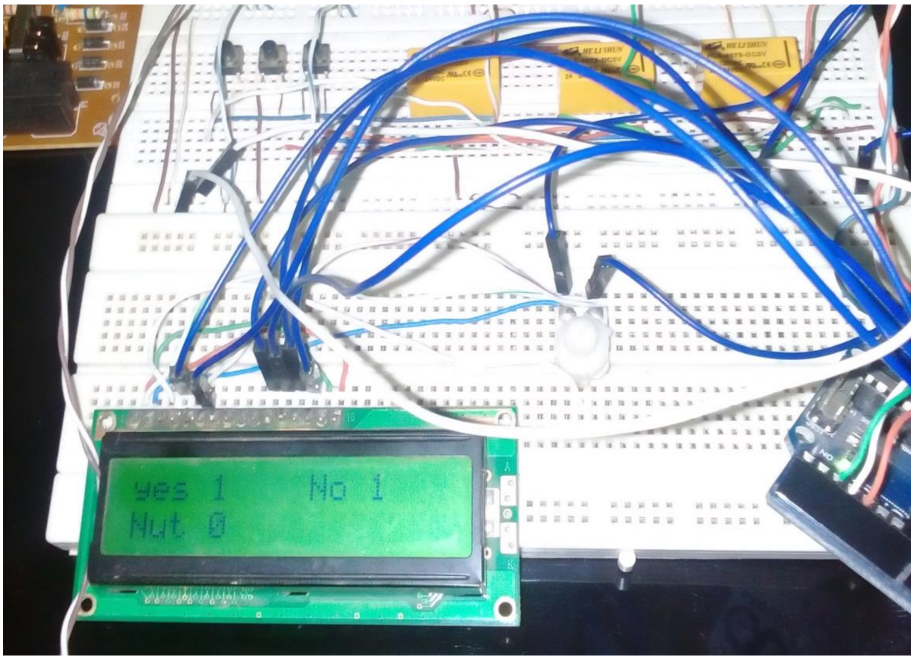

Figure 4.8: Different voting (1 yes and 1 no)

- If the three voters voted "YES", Figure 4.9 shows the result of this case.

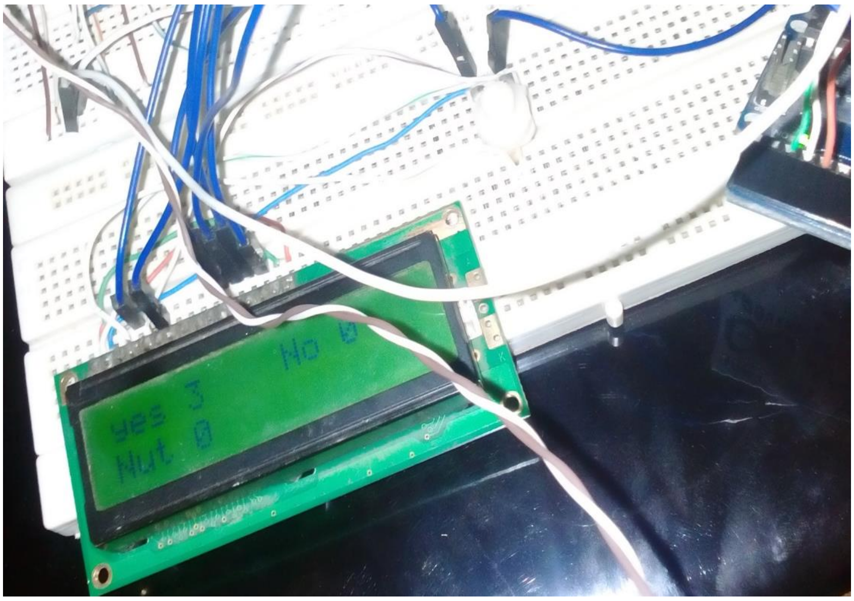

Figure 4.9: Similar voting (3 yes)

- If two voters voted "YES" and one voted "NO", Figure 4.10 shows the result of this case.

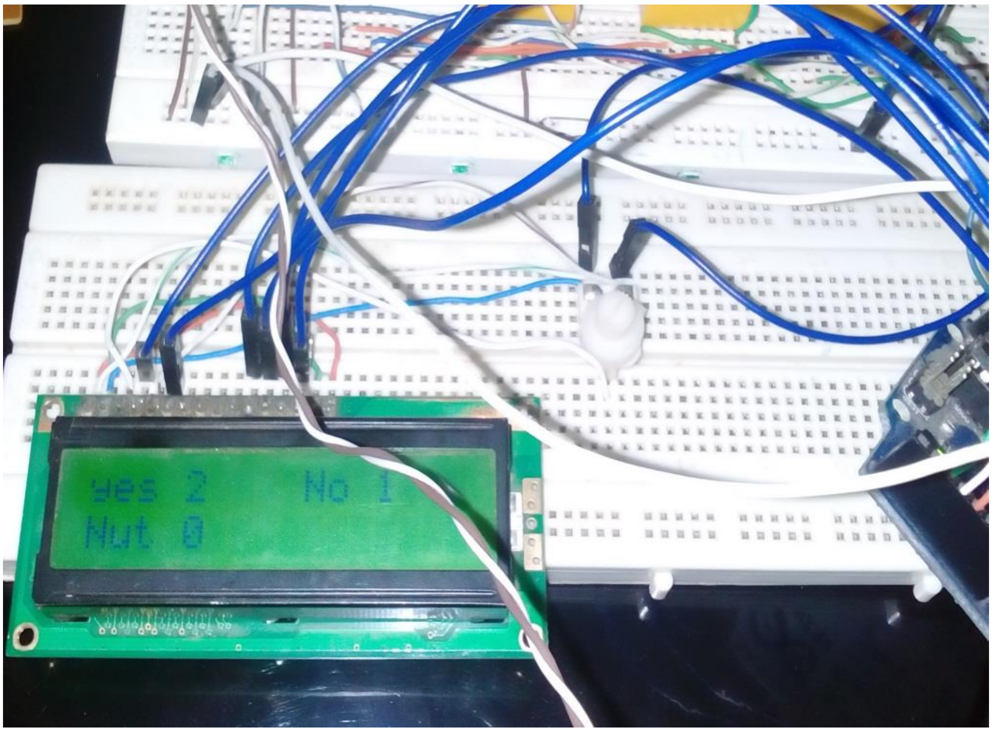

Figure 4.10: Different voting (2 yes and 1 no)

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The main objective of this research is facilitate the voting process and insure the accuracy and reliability of the voting process. In this thesis the electronic voting machine is designed. The electronic components that were used has been clarified and explained in manner that how linking them with each other and how they work. After designing and testing found that the control circuit has been implemented successfully and the system is very suitable so it contribute to solve some of the problems, as well as that of the control circuit is easy to design and programming.

5.2 Recommendations

- Using a memory unit for storing the results of previous voting process.
- Use a wireless unit between a voting unit and the control unit such as RF module.
- Use a solid state relays instead of the electrical relays to reduce power consumption.

REFERENCES

[1] Lelia Barlow, "An Introduction to Electronic Voting", November 2003.

[2] Massimo Banzi, "Getting Started with Arduino", Second Edition, September 2011.

[3] David Harris and Sarah Harris, "Digital Design and Computer Architecture", Second Edition, Morgan Kaufmann, 2012.

[4] Kenneth J. Ayala, "The 8051 Microcontroller Architecture, Programming and Application", Western Carolina University, 1991.

[5] A. P. Godse, "Microprocessor and Microcontroller Systems", Technical Publications Pune, 2007.

[6] Allan G. Smith, "Introduction to ardunio", 2011.

[7] Robert H. Chen, "Liquid Crystal Displays Fundemental Physics and Technology", Willy Co, 2011.

[8] Swetha J1 "Rfid based Automated Bank Locker System", 2014.