CHAPTER-1

INTRODUCTION

The DISPLAY BOARD consists of 24 columns and 6 rows. Now the question comes that what are these ROWS and COLUMNS? The rows are made by joining the cathodes and a Common terminal is taken out, similarly the columns are having anodes of the LED. The basic concept behind this project is MULTIPLEXING, it is a way to split the information's and sends them one by one, doing this will save many pins of our microcontroller. We can use any type of LED, like clear lens, diffused lens or RGB LED. But we have used red LED with clear lens.

The moving effect is generated by the shift registers, there are total 8 outputs in the SHIFT REGISTER (74HC595 IC) and using 3 pins (the serial input, clock and the latch) we can control 24 columns. Because of using shift registers we have to use very few pins of the microcontroller and thus we have to program only few pins. This will make our programming easy and simple. The output of the DECADE COUNTER (4017) is connected to the rows according to the order. The output current from the decade counter is very small and does not match to the rated current of the LED, so we have used BJT (bipolar junction transistors) in order to amplify the output current. The display board consists of the LEDs only and don't have any other elements which requires more power like motors etc., so it does not require any extra power source and it will be fully functional with the power from the computer.

1.1. Project Goal

The goals of my projects are as follows:-

1. Design the 24X6 LED display board using 144 clear lens LEDs.
2. To minimize the number of microcontroller pins used, designing the circuit consisting of SHIFT REGISTERS (74HC595 IC) and DECADE COUNTER (4017 IC).
3. Programming of the ARKDUINO BOARD according to the characters which has to be displayed.

All the three aims of our project are achieved one by one and after completion of the programming part our display board is fully functional.

1.2 Methodology to be adopted

The methodology which we are going to adopt in this project is explained in the following points-

- First step is to study the work which has been done in this field. After this we will briefly study about the components used in the project.
- Now the LED matrix will be made by using the clear lens LEDs
- After completion of the display board, the controller circuit will be made. The controller circuit can either be made by simple soldering or by PCB designing.
- The programming of the microcontroller will be done.
- Now the proper base or stand will be made which can hold the display board and the circuit.

CHAPTER-2

COMPONENTS USED IN PROJECT

In this chapter we will discuss the properties of all the basic elements which we have used in our project and also try to put some light on their role in our project. The table below is a quick view of the components used in our project. After this table we will see the brief description of the elements along with their diagram. The knowledge of the basic components is important because this will equip us for improvement in our project in the coming future.

2.1 List of Components

Following are the names and specifications of the components used in making the board and the controller circuit.

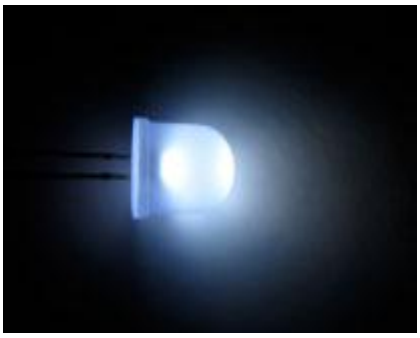
S.No.	**Name of Component**	**Specification**	**Quantity**
1.	LED	Clear lens, 5mm, and red color.	144
2.	Perfboard	Extra large	1
3.	Resistor- for LED for BJT	100Ω 1000Ω	24 6
4.	Decade Counter	4017 IC	1
5.	Shift Register	74HC595 IC	3
6.	BJT	2N3904	6
7.	Breakaway headers	Square, pin length- 11.5mm and pin spacing- 2.54mm.	7

Table: - 2.1 List of components used.

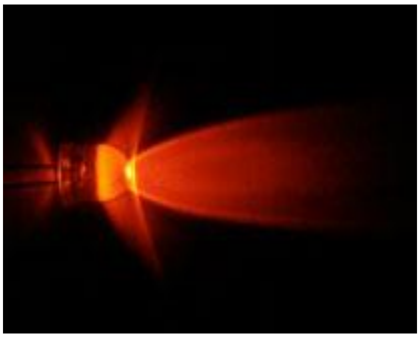
2.2 Description of Components

2.2.1 LED

In our project we have used the clear lens LEDs; we can also use the diffused lens LEDs. The basic difference between the both types is that the covering coating in the clear lens is transparent and just by looking we cannot tell about the color which the LED is going to emit but on the other hand in case of diffused lens LED the covering coating is not clear but it is colored.



Diffused lens



Clear lens

Fig: - 2.1 Types of LED

The brightness of the clear lens LED is much more than the diffused lens and it is also difficult for us to see the clear lens LED directly. In this project the basic concept of selecting the red LED is that, according to the Rayleigh law of scattering, the light ray having shorter the wavelength larger is the scattering and as we all know that the red is having the largest wavelength among all the colors and thus suffers the least scattering. This will make the information displayed on the board, visible from the large distance also.

Brightness: the brightness of the LED is measured in candela or millicandela.

Candela = cd

Millicandela = mcd = 0.001cd

1cd = brightness produced by one candle.

In case of LED used by us -

- Brightness or luminous intensity – (8000 to 10000) mcd
- Forward voltage – (2.0 to 2.4) volt
- Forward current – 20 mA
- Peak forward current – 30 mA
- Power dissipation – 65 mW
- Width – 5 mm

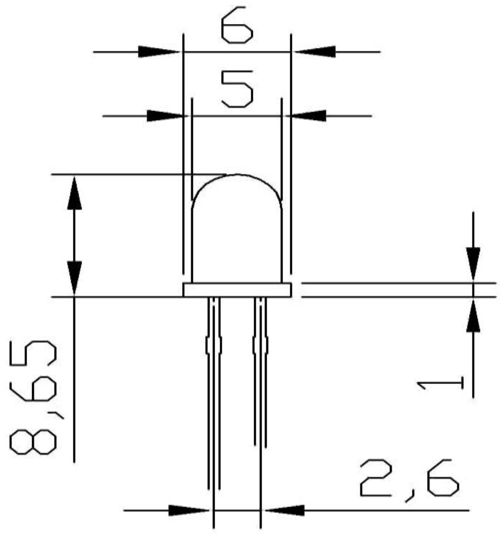


Fig: - 2.2 LED Dimensions

2.2.2 Colour Code Resistance

As we all know that, most of the ICs work on the 5 volts of voltage and the voltage rating of our LED is around 2 volts. Thus it becomes important for us to draw the voltage using some energy dissipating element in series with the LED. For this purpose we use colour code resistance. Now the question comes that what resistance should we use?

It is very simple to calculate the value of the resistance used. The only thing we need to know is the value of the rated current and the rated voltage of the LED used.

Let the applied voltage be V

And the rated voltage be V1

Current rating of the LED is I

Then the required resistance is => R= (V-V1)/I

This resistance is connected in series with the LED. After some small experiments we are able to select our required resistance. We can do this either on the bread board or on MULTISIM software. I have checked my result on both and there is a very small deviation of the result. The table below gives my experimental data.

S.No.	RESISTANCE	VOLTAGE (across LED)
1.	1000Ω	1.76V
2.	200Ω	1.96V
3.	100Ω	1.84V

Table: - 2.2 Experimental values of resistance

On the basis of the above result we found that using the 100Ω resistor will produce the voltage drop of 1.96V, which is closest to the Rated Value of the LED i.e.; 2 volts. So we will connect 100Ω resistor in series with the LED.

Fig: - 2.3 100Ω Colour Code Resistance

2.2.3 Perfboard

Perfboard is a material for prototyping electronic circuit. It is a thin, rigid sheet with holes pre-drilled at standard intervals across a grid, usually a square grid of 2.54 mm (0.1 in) spacing. These holes are ringed by round or square copper pads. Inexpensive Perfboard may have pads on only one side of the board, while better quality Perfboard can have pads on both sides. Since each pad is electrically isolated, the builder makes all connections with either wire wrap or miniature point to point wiring techniques. Discrete components are soldered to the prototype board such as resistors, capacitors and integrated circuits. The substrate is typically made of paper laminated with phenolic resin (FR-2) or a fibreglass-reinforced epoxy laminate (FR-4).

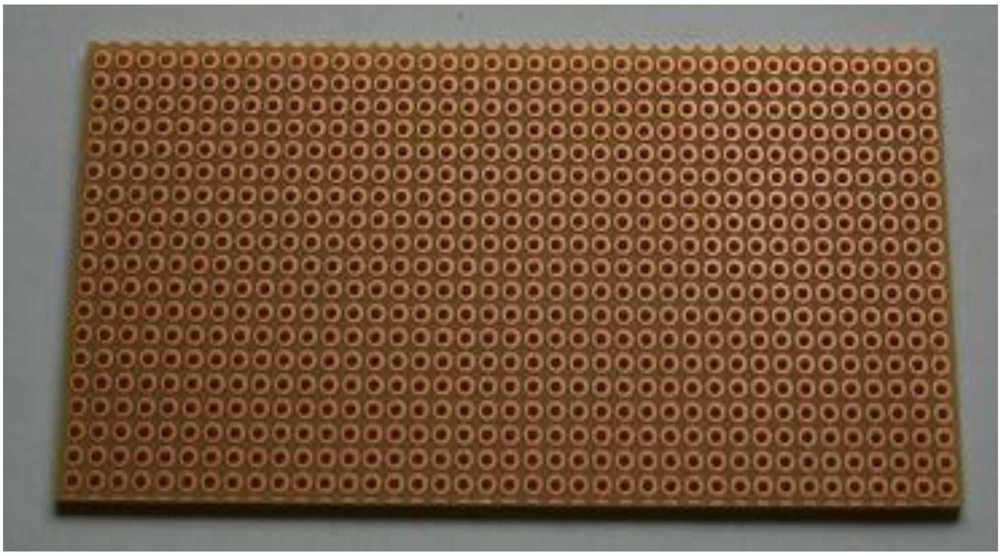


Fig: -2.4 Perfboard

Before building a circuit on Perfboard, the locations of the components and connections are typically planned in detail on paper or with software. Software layout can often be used to generate Perfboard layouts as well. In this case, the designer positions the components so all leads fall on intersections of a 0.1 in grid. When routing the connections more than 2 copper layers can be used, as multiple overlaps are not a problem for insulated wires.

2.2.4 Soldering Iron and Soldering wire

Soldering Iron:

A soldering iron is a hand tool used in soldering. It supplies heat to melt the solder so that it can flow into the joint between two work pieces.

A soldering iron is composed of a heated metal tip and an insulated handle. Heating is often achieved electrically, by passing an electric current (supplied through an electrical cord or battery cables) through a resistive heating element. Cordless irons can be heated by combustion of gas stored in a small tank, often using a catalytic heater rather than a flame. Simple irons less commonly used than in the past were simply a large copper bit on a handle, heated in a flame.

Fig: - 2.5. Soldering Iron

Soldering Wire:

Solder is basically metal wire with a "low" melting point, where low for our purposes means low enough to be melted with a soldering iron. For electronics, it is traditionally a mix of tin and lead. Lead has a lower melting point than tin, so more lead means a lower melting point. Most common lead-based solder you'll find at the gadget store will be 60Sn/40Pb (for 60% tin, 40% lead). There are some other minor variations you're likely to see, such as 63Sn/37Pb. Now, molten metal is a tricky beast, because it behaves a bit like water: Of particular interest is its surface tension. Molten metal will ball up if it doesn't find something to "stick" to. That's why solder masks work to keep jumpers from forming, and why you see surface-mount soldering tricks. In general, metal likes to stick to metal, but doesn't like to stick to oils or oxidized metals. By simply being exposed to air, our parts and boards start to oxidize, and through handling they get exposed to grime (such as oils from our skin). The solution to this is to clean the parts and boards first. That's where flux cores come in to solder. Flux cores melt at a lower temperature than the solder, and coat the area to be soldered. The flux cleans the surfaces, and if they're not too dirty the flux is sufficient to make a good strong solder joint (makes it "sticky" enough).

Fig: - 2.6 Soldering Wire

The above discussed elements are used to design the display board. After designing the display board we need to design the circuit having the shift registers and the decade counter in order to reduce the pins used in microcontroller and to create the moving effect. Now this time we will discuss the elements which are used to design this circuit.

2.2.5 General purpose Transistor (2N3904)

The 2N3904 is a commonly used NPN bipolar junction transistor. A BJT is basically a current amplifier (ex- if we let 1 mA current to the base then we will be getting 100 mA current at collector). The main aim of using the BJT is that the current at the output of the decade counter is smaller than the rated current of the diode. This transistor is used very frequently in small projects and its cost is also quite low.

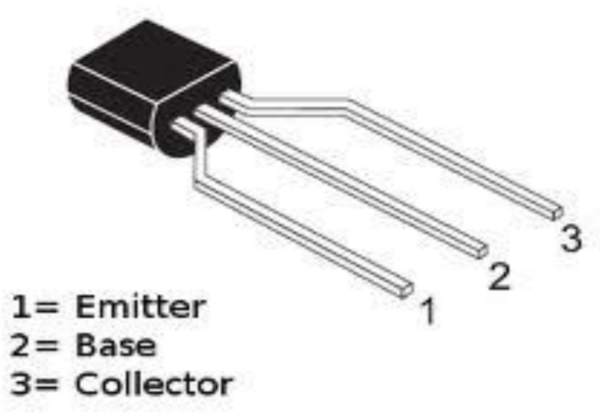


Fig: - 2.7 General purpose transistor (2N3904)

The correct way to see for the emitter, base and collector is to see from the side of the flat part. The first leg is the emitter, the second leg is the base and the last leg is the collector of the transistor.

2.2.6 Breakaway Headers and jumper wires

We used the breakaway headers to connect the circuit board with the microcontroller; the main aim of using this is to make our circuit free from any permanent connection. These are the pointed metallic pin like structure which is connected with each other by the means of the plastic and in a single strip we can have 40 headers.

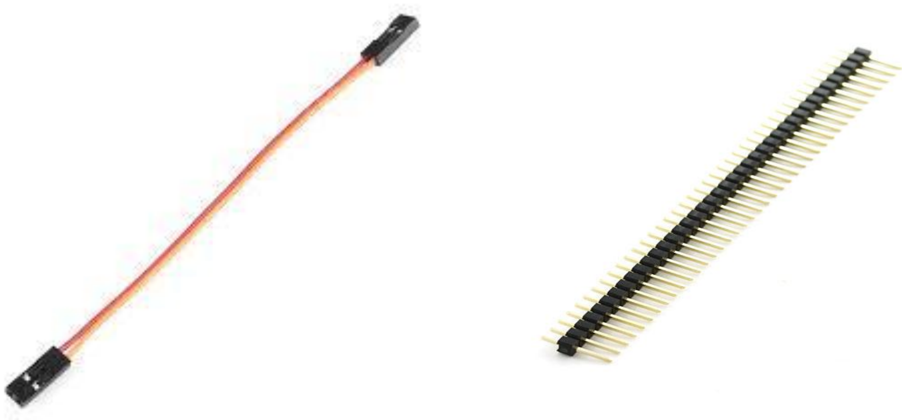


Fig: - 2.8 Jumping wire and Breakaway headers

2.2.7 Decade Counter (4017)

A 4017 integrated circuit is technically called as JOHNSONS 10 STAGE DECADE COUNTER /DEVIDER. A decade counter is the sequential circuit, which is used to count the pulses. As the name suggests it can count the ten clock pulses (0 to 9). These counters are made of the flip-flops; in case of decade counter four flip-flops are required. But they can also be realised using the CMOS. The 4017 IC is able to light the LEDs up in the sequence. These lights up the every LED connected to the output according to the sequence in which they are connected.

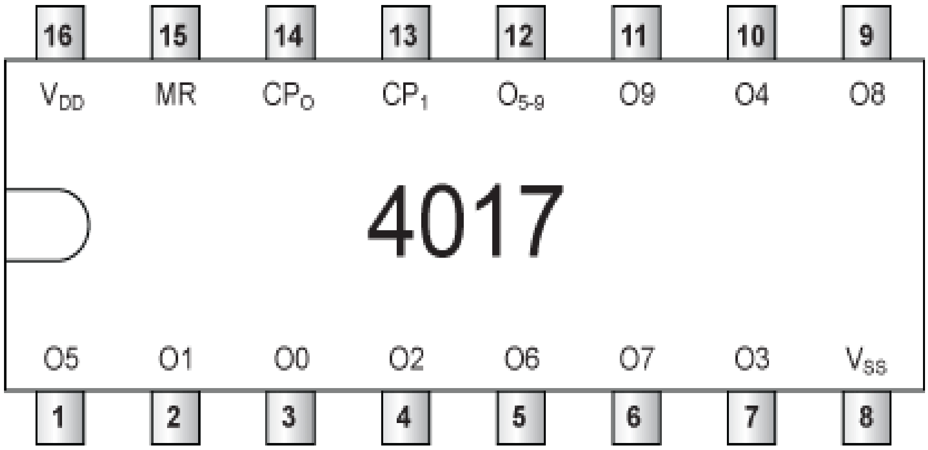


Fig: - 2.9 4017 IC

As we can see from the diagram above, the IC 4017 is a 16 pin dual in line package IC. Pin 1 can be identified from a small depressed circle at the extreme left corner of the IC, or simply one can always remember, the printed side of the IC facing towards you, the pin beginning from the left side of the semi circle notch of every IC is pin 1.

Pin Configuration:

Output Pins: Pin 1 to 7 and Pin 9 and 10 are all outputs of the IC. The output of the Decade Counter is not according to the sequence (1, 2, 3….10), but it is random. Let us see what is that sequence.

Out#1 = Pin#3

Out#2 = Pin#2

Out#3 = Pin#4

Out#4 = Pin#7

Out#5 = Pin#10

Out#6 = Pin#1

Out#7 = Pin#5

Out#8 = Pin#6

Out#9 = Pin#9

Out#10 = Pin#11

Pin#12: This pin is the carryout and it has significance when many ICs are connected in series. In other case it can be left open.

Pin#13: It is the clock enabling point. Logic '1' to this pin will stop the IC 4017 from proceeding and its output will freeze at that instant at the particular output. Even if the clock signal at pin 14 is ON, the output cant shift as long as pin 13 is held at logic'1', therefore this point should be grounded.

Pin#14: It is the clock input of the IC 4017. An external clock signal to this point will make logic '1' to proceed sequentially, beginning from pin 3 and ending at pin 11.

Pin#15: It is the reset pin. It is used when we need to put the output back to pin#3 and begins a fresh cycle.

Pin#8: It is the ground pin.

Pin#16: Supply voltage is applied at this pin.

The 4017 IC is a 16 Pin DIL IC. The role of the decade counter in our project will be discussed in the coming chapter.

2.2.8 Shift Register (74HC595)

Shift Registers are also a sequential circuit like decade counter. It used to multiply the output of the microcontroller. By using this we can save various output pins of the controller. In our project we have used it as serial output, by doing this the output of the controller is shifted to each column one by one and this will produce our moving effect. It has eight output pins (Q0 to Q7) and rest of the pins can be considered as the control pins.

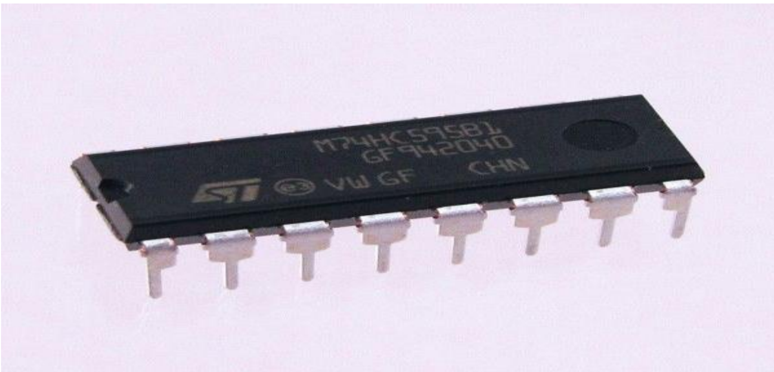
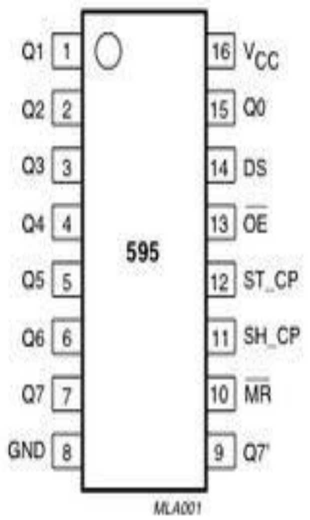


Fig: - 2.10 74HC595 IC



PINS 1-7, 15	Q0 " Q7	Output Pins
PIN 8	GND	Ground, Vss
PIN 9	Q7"	Serial Out
PIN 10	MR	Master Reclear, active low
PIN 11	SH_CP	Shift register clock pin
PIN 12	ST_CP	Storage register clock pin (latch pin)
PIN 13	OE	Output enable, active low
PIN 14	DS	Serial data input
PIN 16	Vcc	Positive supply voltage

Fig: -2.11 8 bit Shift Register

We can control the shift of the register with clock pulses. As we raise the signal going to the clock pin to high, the clock is moved forward one step. And when we pull it low and high again it shifts another. Each time we shift the clock we switch the input to a different one of the eight registers. We are essentially controlling the output of each of the eight pins one at a time, and as we move one clock signal forward, we switch to the next output pin to control. Set the latch pin low then send data through pin#14 and once data has been sent, toggle the clock pin again set the latch pin high.

This was all the important components used in our project. After the knowledge of the basic components, we can now move to the block diagram and the construction of our project.

CHAPTER – 3

BLOCK DIAGRAM OF THE PROJECT

The display board does not require the power from the adaptors. So we are using the power of the computer coming from the USB cables. In the ARDUINO board, there is a little power jumper (next to the USB port). Using this power jumper we can either power the board from the USB port or from the external power supply. As we all know that the LEDs are very less power consuming elements so we will power up the board using the USB port only.

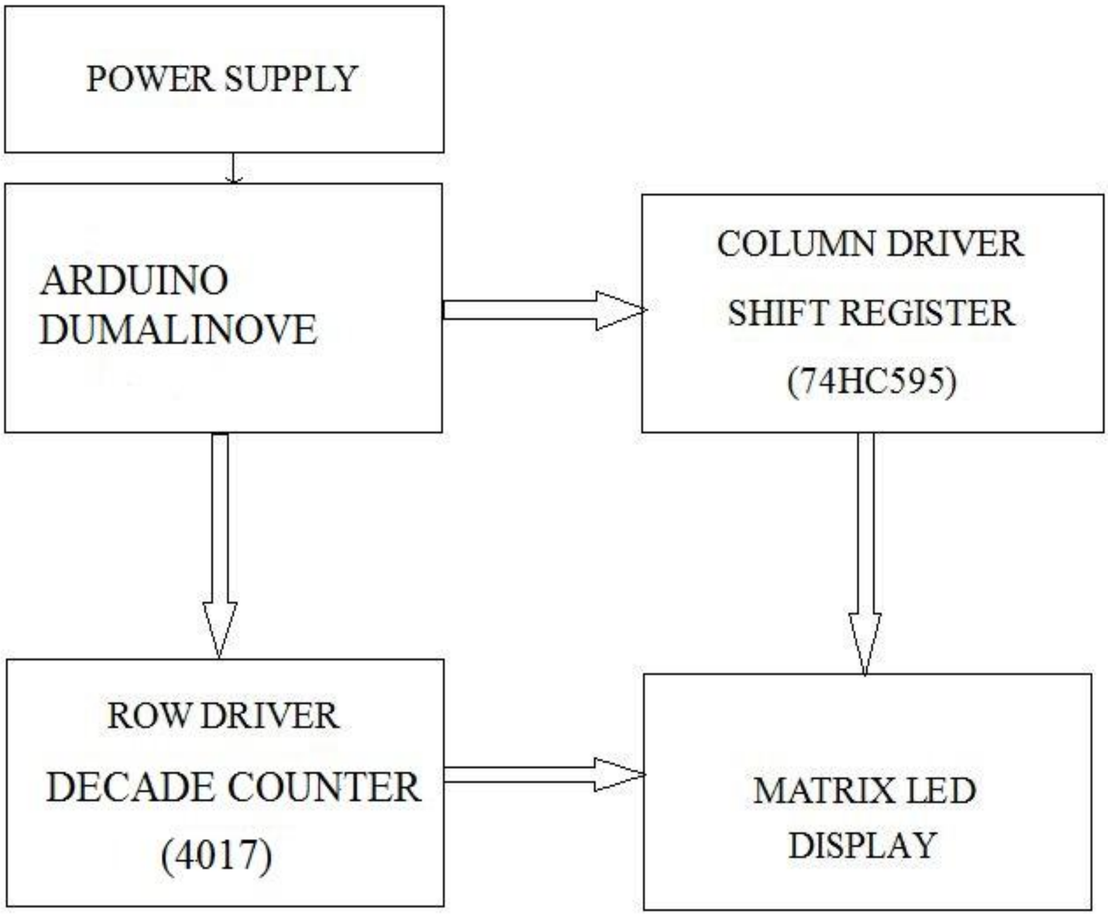


Fig: - 3.1 block diagram representation

The above block diagram is the simple representation of the working of our project. After selection of the power source, this power is given to the ARDUINO board. According to the program we have dumped, the microcontroller will control the shift registers and the decade counters. Now as we have also seen earlier that the shift registers are used to control the columns and the decade counter are used to control the rows. So the SHIFT REGISTERS and the DECADE COUNTER are controlling the animations displayed on the board.

CHAPTER- 4

HARDWARE IMPLEMENTATION

In this chapter we will see how the basic components which we have seen in the chapter-2 are used to build our hardware part. We already know that there are basically two hardware parts in our project, the first one is the display board and the second one is the circuit having registers and the counter. Now we will first see the steps in making the display board and then move to the other part.

4.1 Making the Display Board

Before we start making our display board, we must do some practise of the soldering as it requires a lot. The making of the board requires basically two things, the soldering skills and the great concentration. The soldering skills are necessary because you are going to solder 144 LEDs and also each LED has two legs. Concentration is important because you have to solder the cathode and the anode separately.

4.1.1 Soldering the LEDs

The perfboard taken should be long and broad enough for 24 columns and 6 rows. Now we will see how to solder the LEDs. First take one LED and see for its cathode and anode, for easy note that the longer leg of the LED is the anode and the shorter one is the cathode.

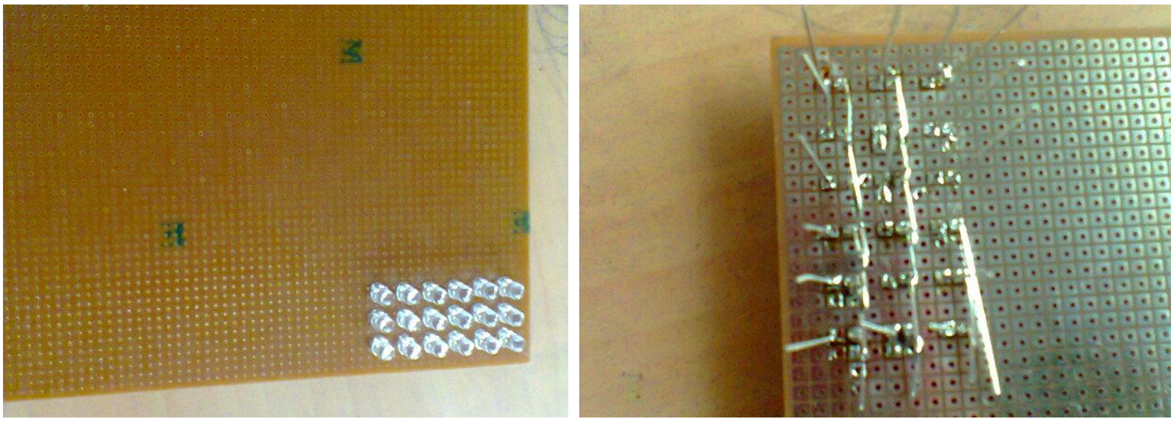


Fig: -4.1 Soldering the LED

Once you are sure about the terminals, insert them in to the two adjacent holes and remember the pattern (that cathode is in the right or the anode) in my case I selected anode to be in the right and cathode to be in the left.

Once you insert the LED in the hole, just bend the on terminal so that LED remains in its place. Now solder the leg which is not bend after doing that straighten the leg which was bend and then solder it too. So this was the complete step to solder the led. Now you can easily solder all the 144 LEDs. The only thing you have to keep in mind that the pattern of the anode and the cathode remains same. The following is the schematic to make the display board.

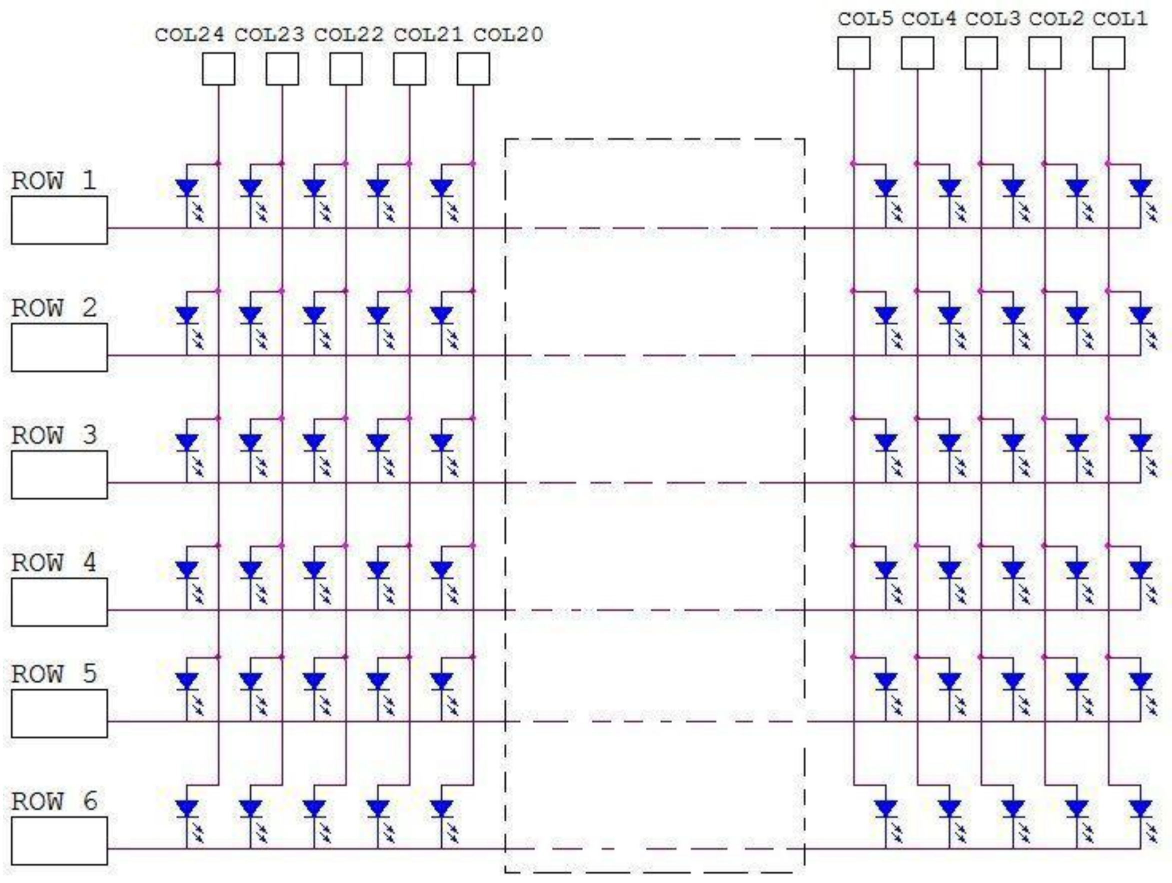


Fig: -4.2 Schematic of the Display Board

4.1.2 Making the Columns

After soldering all the 144 LEDs, the next task is to make the columns. You can also make the rows first but in my case I put the columns below the rows so I was bounded to make it first. So whatever (rows or columns) you make first, keep it low.

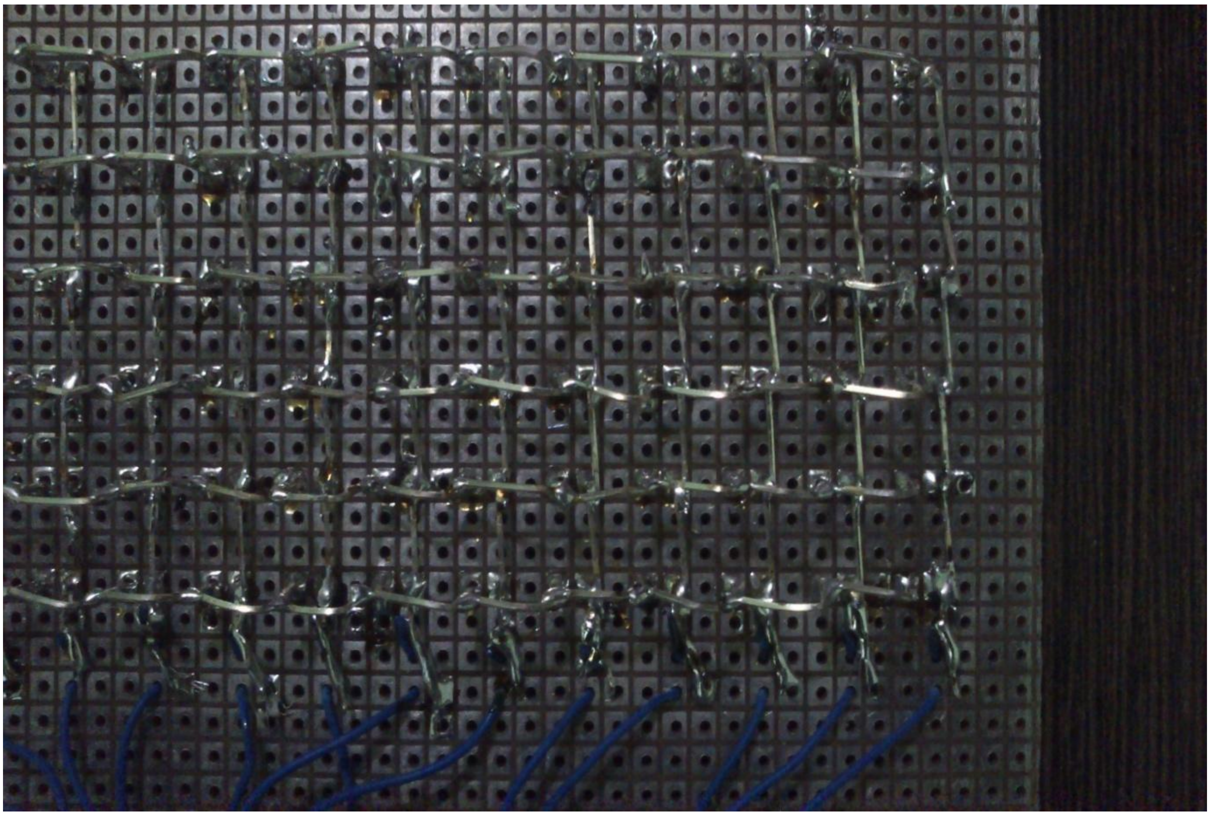


Fig: - 4.3 Making the Columns

Just bend them vertically and cut them by using the cutter, so that they just touch each other after this solder them and you will find that your first column is ready. Repeat the procedure with the remaining LEDs.

4.1.3 Making the Rows

Once you are done with all the 24 columns, you will find that there are only the cathodes of the LEDs which are now left. Repeat the same procedure that we have done while making the columns.

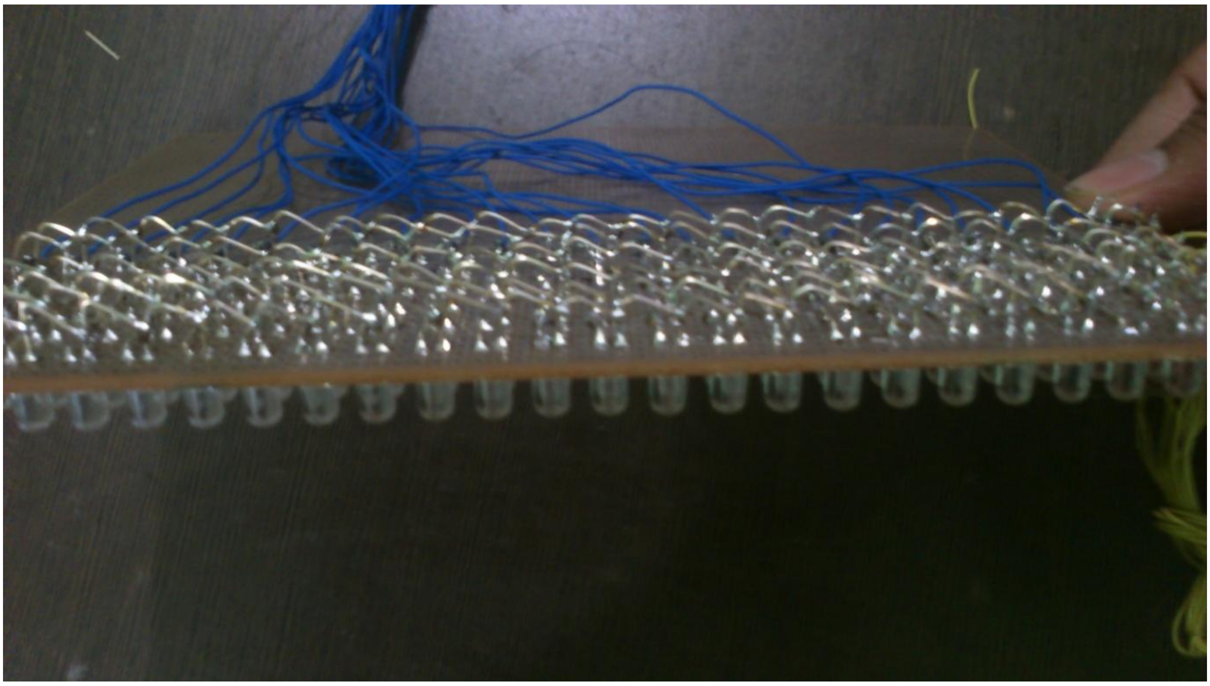


Fig: -4.4 Making the Rows

After completing all the six rows, check the whole circuit that none of the rows or columns is shorted. This board is now ready to follow the principle of multiplexing. It is the principle of selecting one from many. Here also we have various LEDs but in order to display the characters we cannot make high all the LEDs. So we have to select the LEDs according to the pattern. The final form of the display board is shown below.

Fig: -4.5 Front view of the Board

Let us see one example that how multiplexing is done and also the working of this board. Suppose if we want to make the very first LED high then we have to put the Vcc at the first column and the Gnd at the first row. So we can say that the LED, which is at the intersection of the row and the column will be high.

Now suppose we want to flash both circled LEDs in the figure then what will happen.

Fig: -4.6 Multiplexing in the board

In this case we have to connect the first two columns to the Vcc and also at the same time connect the first two rows to the Gnd. But what we will see that, doing this will flash all the four LEDs, which is not our need. Now here comes the multiplexing, by using the microcontroller we will flash the two circled LEDs very frequently that is not easy for the human eye to rule out.

Finally we will take the thin wires out from the each of the columns and the rows. Try to use different colour wires for rows and columns. Doing this will help us connecting them to the shift registers and the decade counter. Now after completing the display board we will move to our next circuit i.e.; controller circuit.

4.2 Implementing the Shift Registers

The shift registers are used to move the information's and are connected to the columns of the display board. In total we require three Shift Register ICs. The 74HC595 have eight outputs and the first eight columns are connected to the first IC and then the next eight to the second IC similarly connect the last eight to the last shift register. Following is the schematic for the circuit.

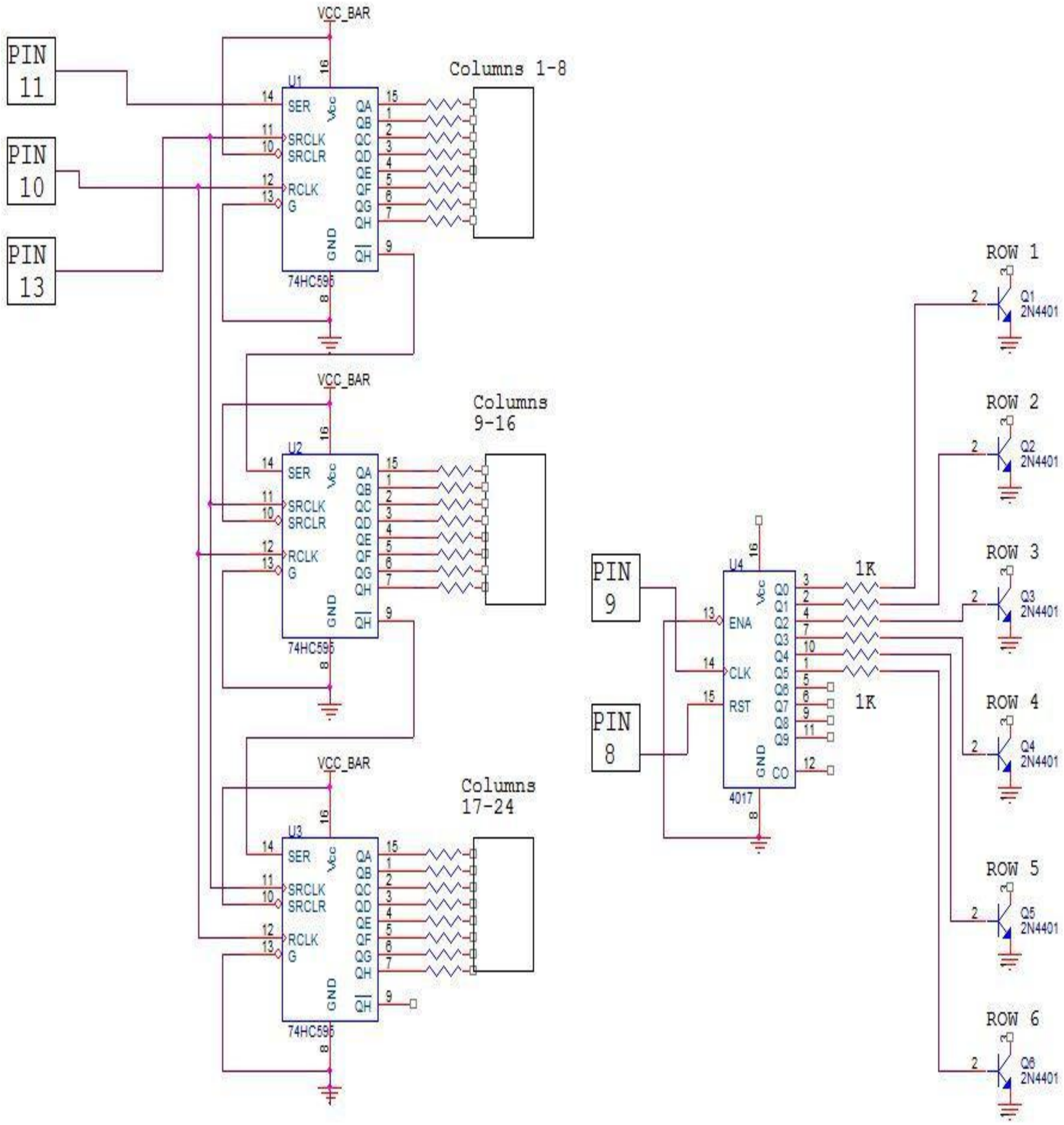


Fig: - 4.7 Schematic of the circuit

In the above schematic we can clearly see that only three pins of the microcontroller are used to control the 24 outputs. Now following the schematic we have to make the circuit on the perfboard. As the 74HC595 is a 16 pin IC we are using the 16 pin IC base. The two diagonal pins of the IC base is bent to avoid the IC to come out from the holes and then the rest of the pins are soldered after that the two pins which are made straight and soldered. This is the simple way to put the IC base on the perfboard. It is good to use the IC base instead of soldering the IC directly.

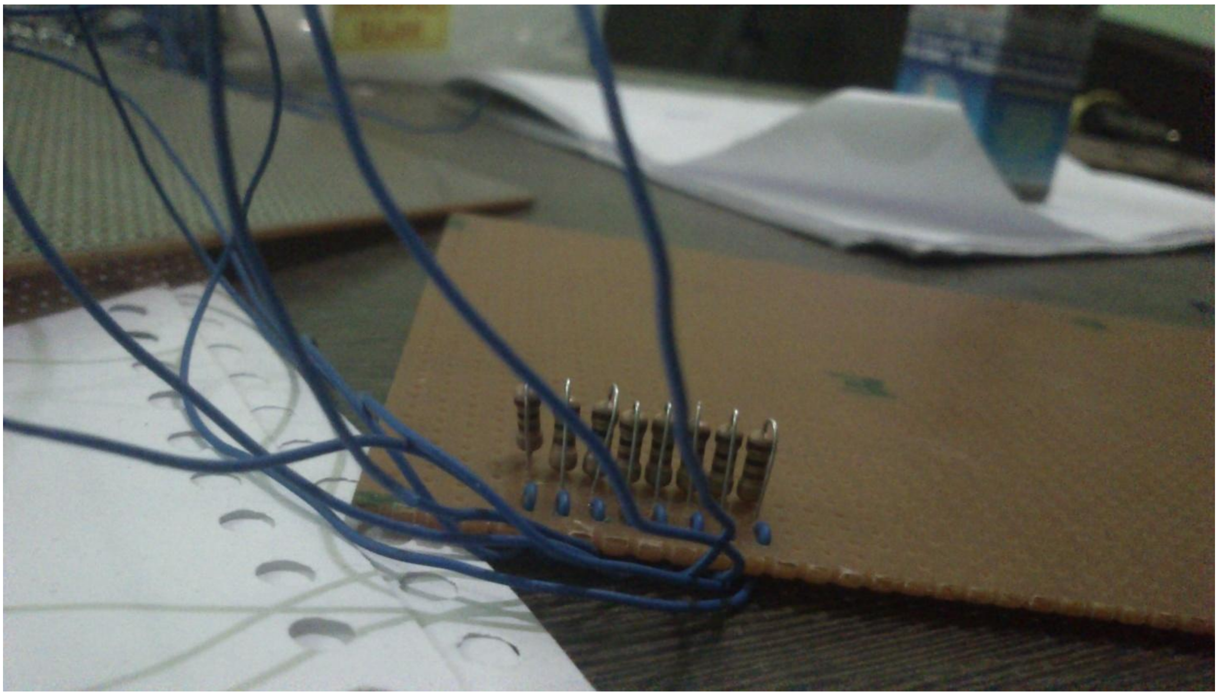


Fig: - 4.8 Soldering the resistors

In order to provide the rated voltage to our LEDs, the 24 colour code resistors having the resistance value of 100Ω is soldered at the output of the 74HC595. After this the other end of the resistor is connected to the respective columns. All the ICs require the Vcc and the Ground to operate, so I made a common ground terminal which later on connects to the ground of the controller and distribute it to the ICs carefully and same is done with the Vcc. Once the shift registers are done then we can move to the decade counter.

4.3 Implementing the Decade Counter

In our circuit we require only a single decade counter IC i.e. 4017. The output from the decade counter is first amplified by using the bipolar junction transistor then sent to the rows. We need to use the current amplifier as the output of the counter does not have sufficient current to run the LEDs. The pin 14 and the pin 15 are connected to the microcontroller and thus we need to take these terminals out and put the breakaway headers at the end. Below is the completed circuit.

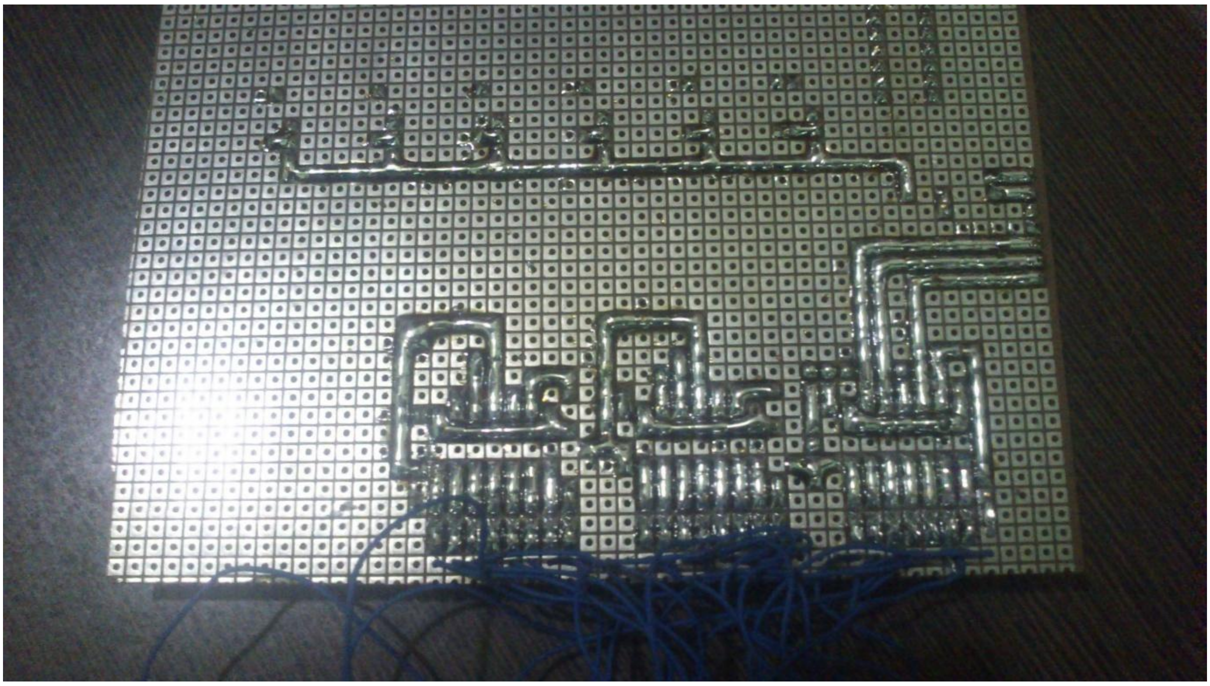


Fig: - 4.9 Top view of the circuit

The BJT used must be properly grounded. When all the ICs are perfectly soldered then take the extend terminals which will be connected to the ARDUINO and put them adjacent to each other. There will be seven inputs. The first one is the Vcc, second is the ground and the rest of the five inputs are from shift registers and the decade counter. The solder line which we made by using the solder wire is a bit difficult to make if we do not know the right way to do it. When we try to make such a line then it will be very tough to join the two blocks, we will see how to make this lines step by step.

Once the circuit has been completed the bottom view will look like the picture below.

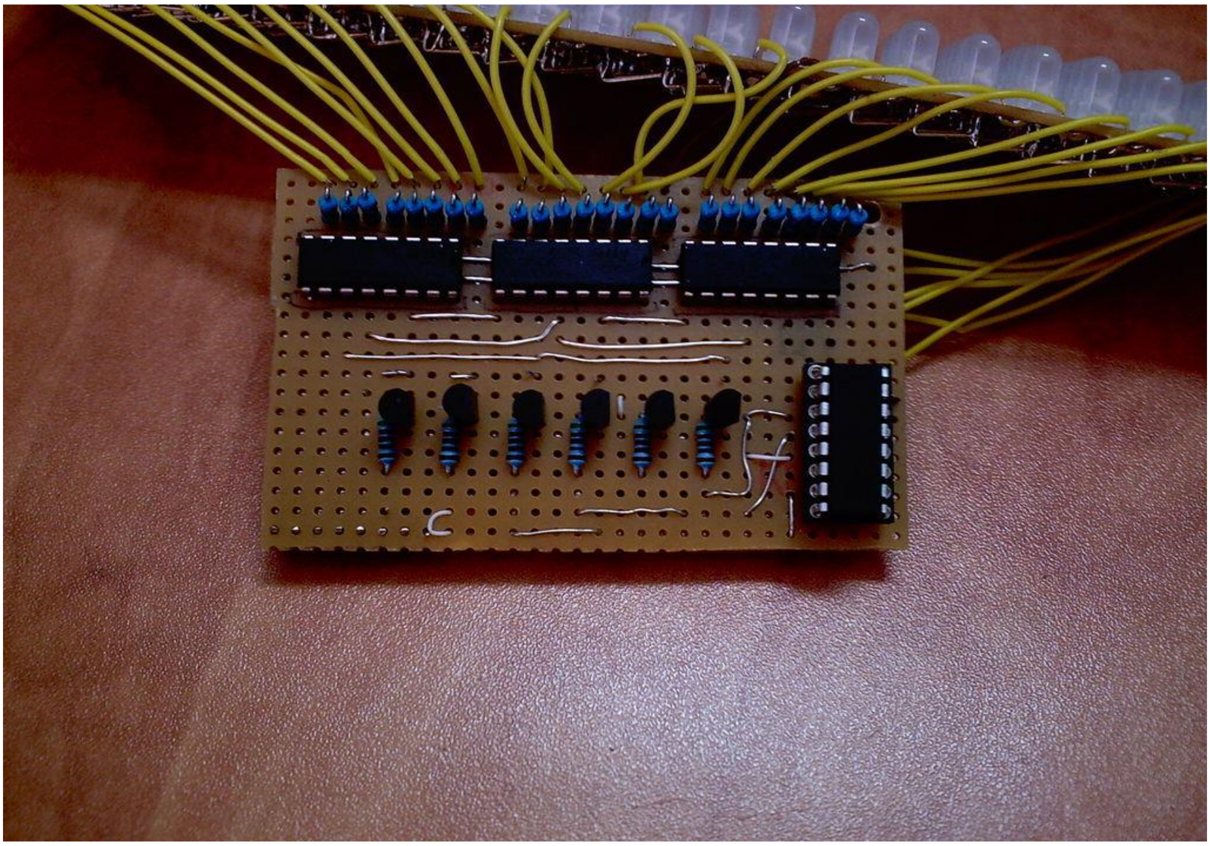


Fig: - 4.10 Bottom view of the circuit

To make the solder lines we have to follow the following steps-

1. Heat the block using soldering iron.

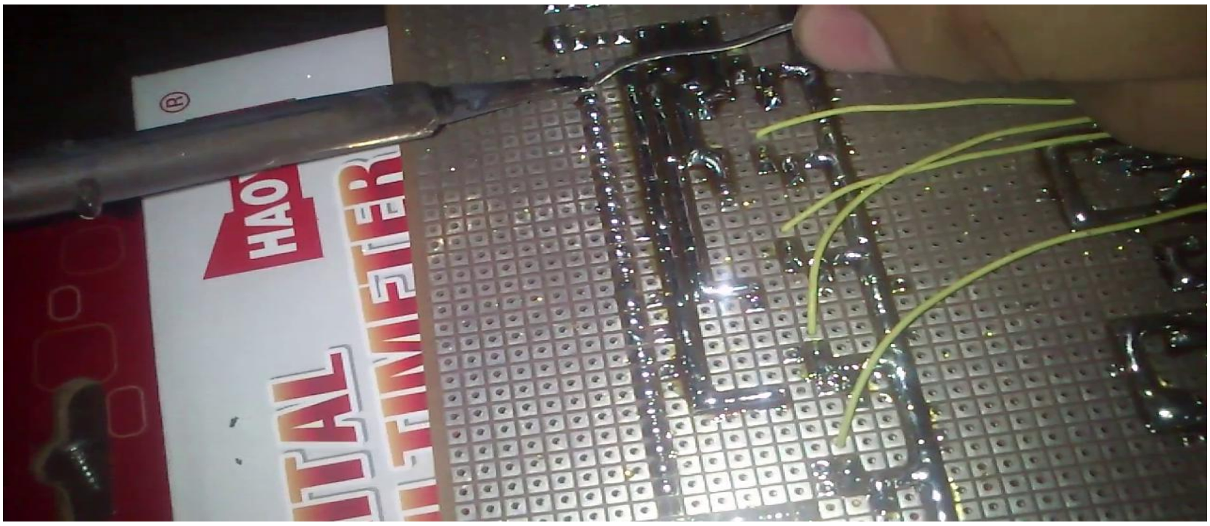


Fig: - 4.11 Heating the board

2. Put the solder on the block and it will melt. You will find that a dot has been made by doing this.

3. Repeat it again on the adjacent block now the trick comes, after the two dots solidify heat them again in between and put some solder. Remove the rod and you will see a small line is there.

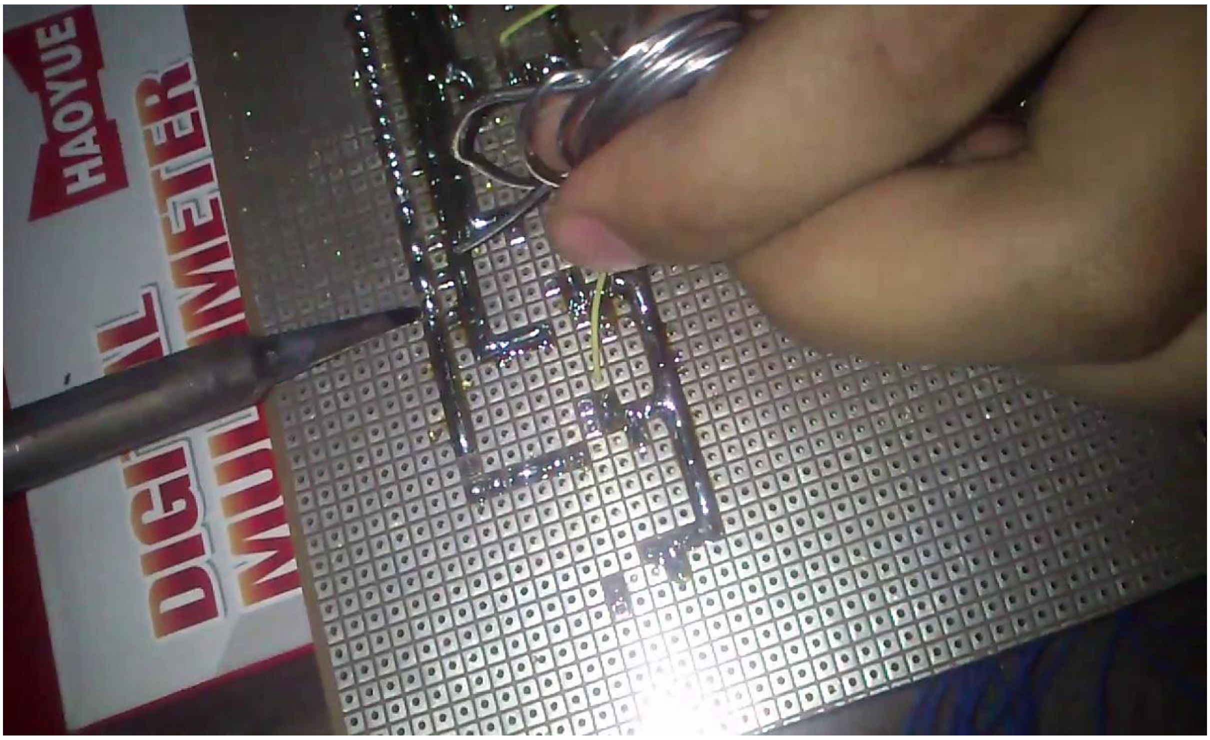


Fig: -4.12 making the solder line on the board

4. Do it again with other two blocks and repeat it according to your need. Making this circuit by the soldering iron and the soldering wire is quite difficult and also there are chances of error. So we can also make this circuit by designing the equivalent PCB. However in my case my circuit operates properly and there is no need for designing the printed circuit board. But I will advice to make this controller board by PCB designing, as it will reduce the labour and it is also more reliable. I am also providing the PCB layout for the controller circuit.

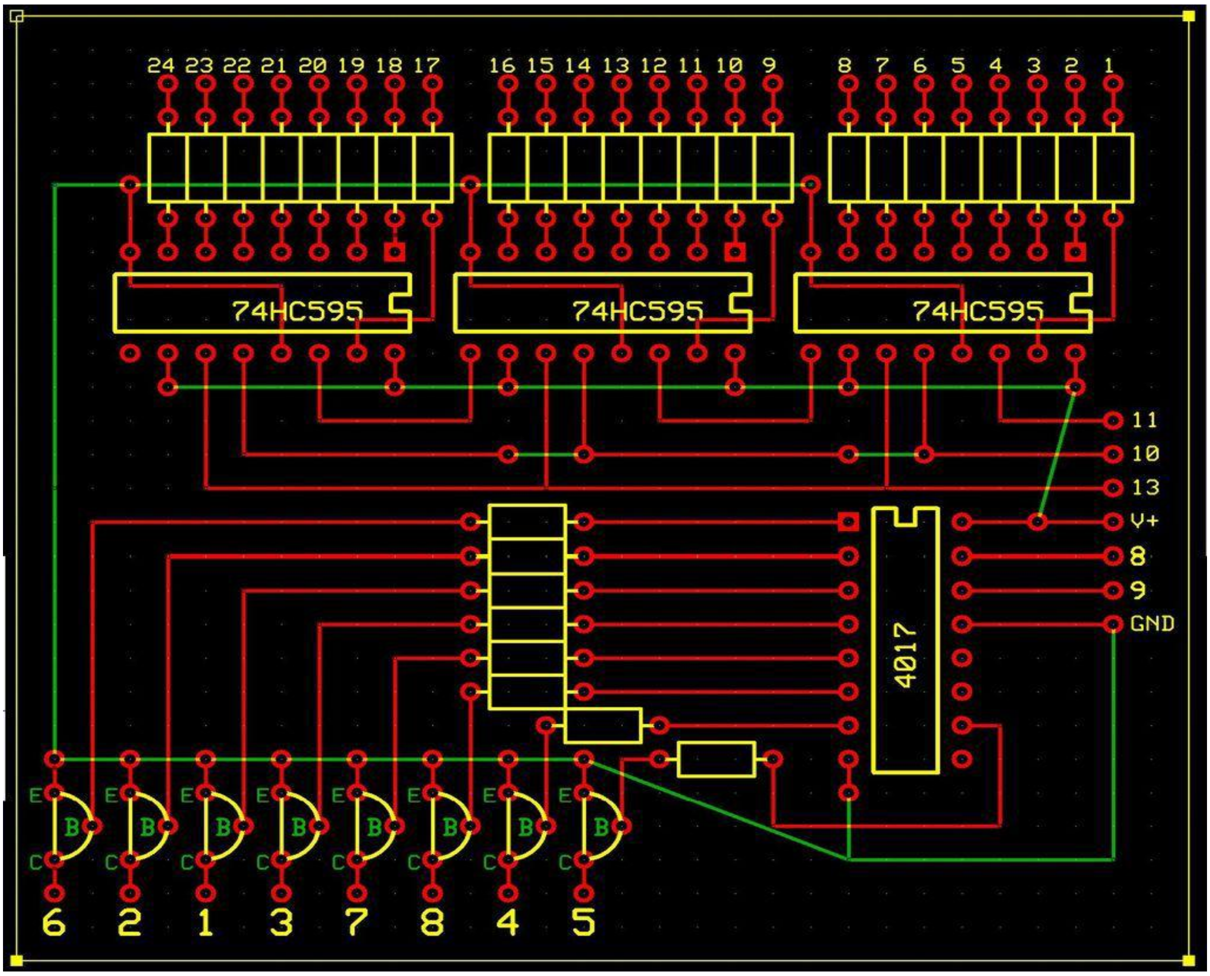


Fig: - 4.13 PCB layout for the controller board

Now we are done with the making of the controller board. Our next step is to program the circuit by using the ARDUINO DUMILANOVE. The ARDUINO will be discussed briefly in the next chapter.

CHAPTER- 5

ARDUINO

5.1 What is ARDUINO?

- In its simplest form, an Arduino is a tiny computer that you can program to process inputs and the outputs going to and from the chip.
- The Arduino is embedded computing platform. It means it interacts with its environment by means of hardware and software.
- The Arduino can be used to develop stand alone interactive objects or it can be connected to a computer to send data to the Arduino and than act on that data.
- The Arduino board is made of an Atmel AVR microprocessor, a crystal or oscillator (enable the controller to operate at correct speed), the voltage regulator (to provide the steady 5V voltage to the microcontroller), the USB connector. The board exposes the microcontroller pins.

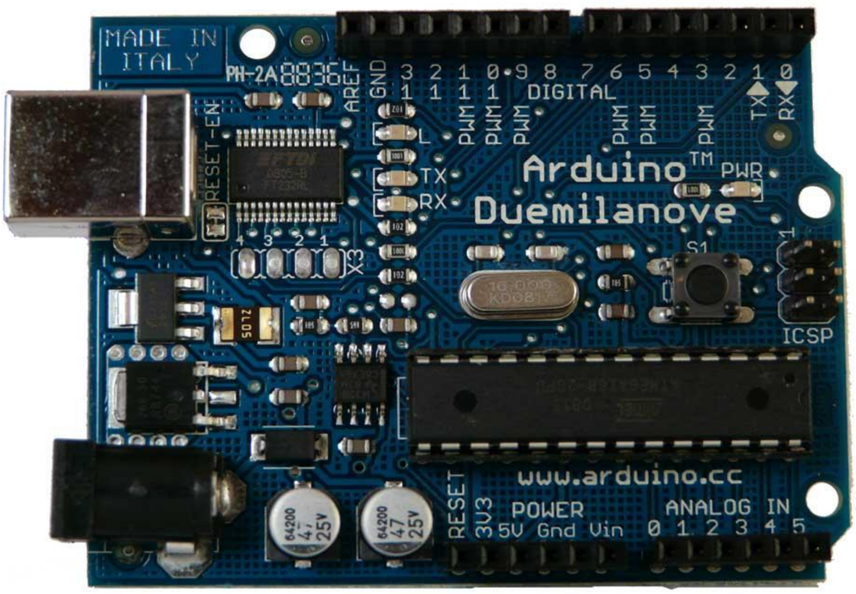


Fig: - 5.1 Arduino Dumilanove

- The Arduino hardware and software are both open source. It means that the code, schematics, design etc. are all open for everyone.
- You can even develop your own board and sell it with other name.
- In our project also we have used one of the clones of the Arduino named ARKDUINO. Some of the clones of Arduino are Freeduino, Boarduino, and Roboduino etc.
- There are many different versions of Arduino available. The most common one is DIECIMILA or DUMILANOVE.
- You can also get Mili, Nano, Uno, Mega etc.

5.2 Arduino IDE

To program the Arduino we use the Arduino IDE. Following are some features of the IDE-

- It is free software that enables us to program in the language that the Arduino understands.
- The program is step by step instruction which is uploaded to the Arduino. Then Arduino will carry out those instructions and interact with the outside worlds.
- The IDE basically consists of two parts, the first part is the editor and the second part is the compiler.
- The editor is where we write our program and the compiler executes the program. The compilation means that the compiler converts the coding to the machine code, which the computer understands and saves the file as .exe file (means executable file). These file when put in some other machine need not require the compilation again and it is ready to use.
- The Arduino IDE is free of cost and it can be downloaded directly from the official website of Arduino in zipped form (around 98 MB).
- The installation of this IDE is different for the different operating systems.
- The IDE and the Arduino coding cannot be used to program any microcontroller as there is a special firmware in the microcontroller which supports the Arduino programming or sketch.

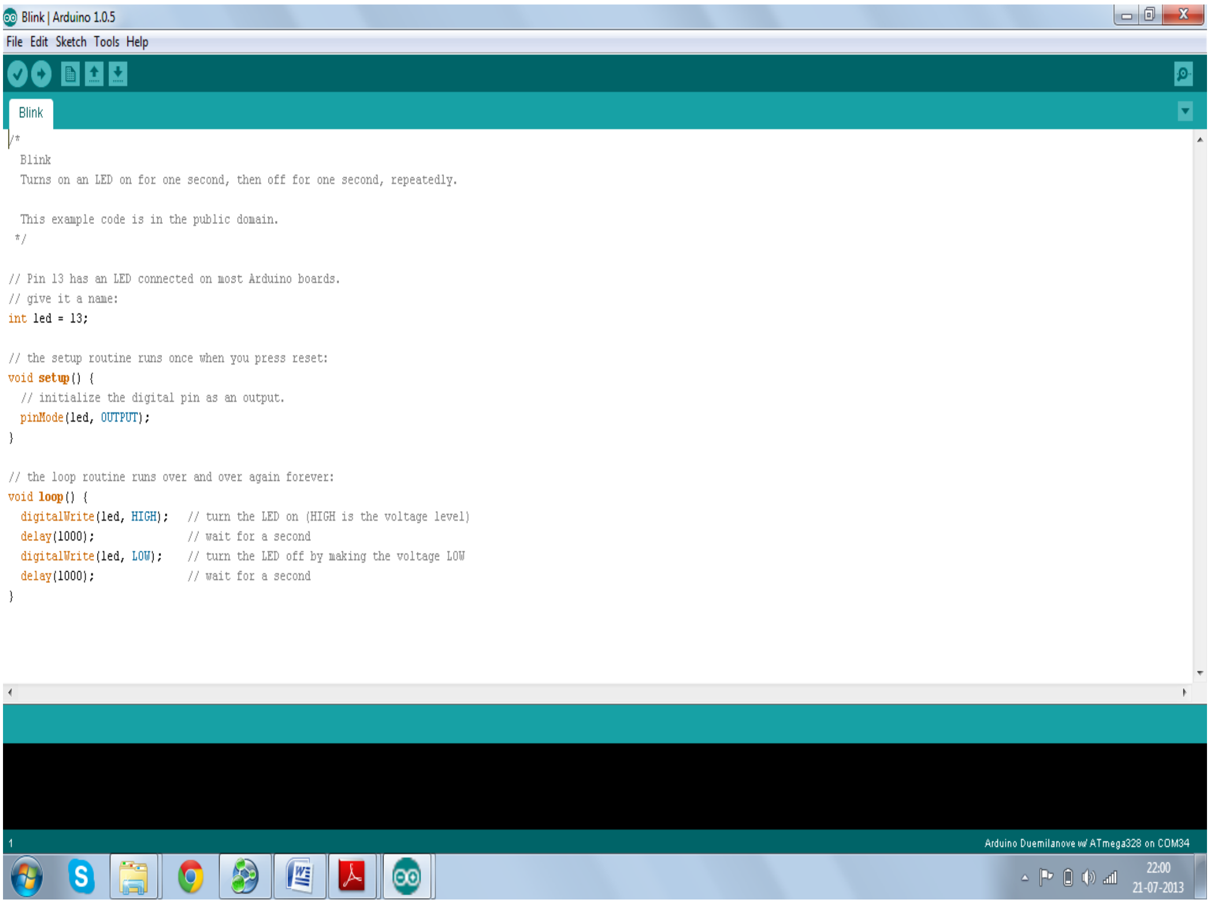


Fig: -5.2 Arduino IDE

5.2.1 The toolbar buttons

Verify/Compile: It checks that the code is correct before we opload it to our Arduino.

Fig: - 5.3 The toolbar buttons

Upload: It will opload the code within the current sketch window to our Arduino. We need to make sure that we have selected the correct board and the port (in the tools menu) before uploading. It is essential that we save our program before we upload it to our board in case strange error causes like system hang or IDE crash. It is also adviced to verify/compile the code before we upload to ensure that there are no errors.

Veryfy/Compile	Checks the code for errors.
Stop	Stops the serial monitor/unhides the other buttons.
New	Creates a new blanck sketch
Open	Shows the list of sketches in our sketch book.
Save	Saves the current sketch.
Upload	Uploads the current sketch in to the Arduino
Serial Monitor	Displays the serial data sent and received to/from the Arduino.

Table: - 5.1 Toolbar buttons

New: The new button will create a completely new and blank sketch. The IDE will ask to enter the name and the location for our sketch and will then give a blanck sketch ready to be coadded.

Open: It will present the list of sketches stored within our sketch book.

Save: It will save the sketch within the sketch window to our sketch file. Once complete we will get a done saving message at the bottom of the window.

Serial monitor: The serial monitor is a very usefull tool, espacially for debugging our code. The monitor displays serial data being sent out from your Arduino (USB or serial board). we can also send serial data back to the Arduino using the serial monitor.

On the left hand side we can also select the Baud Rate that the serial data is to be sebt to/from the Arduino. The Baud Rate is the rate per seccond, that characters (data) is sent to/from the board. the default setting is 9600 baud, ehich means if you have to send a text novel over the serial communications line (in this case our USB cable) then 9600 letters, or symbols of the novel would be sent per second.

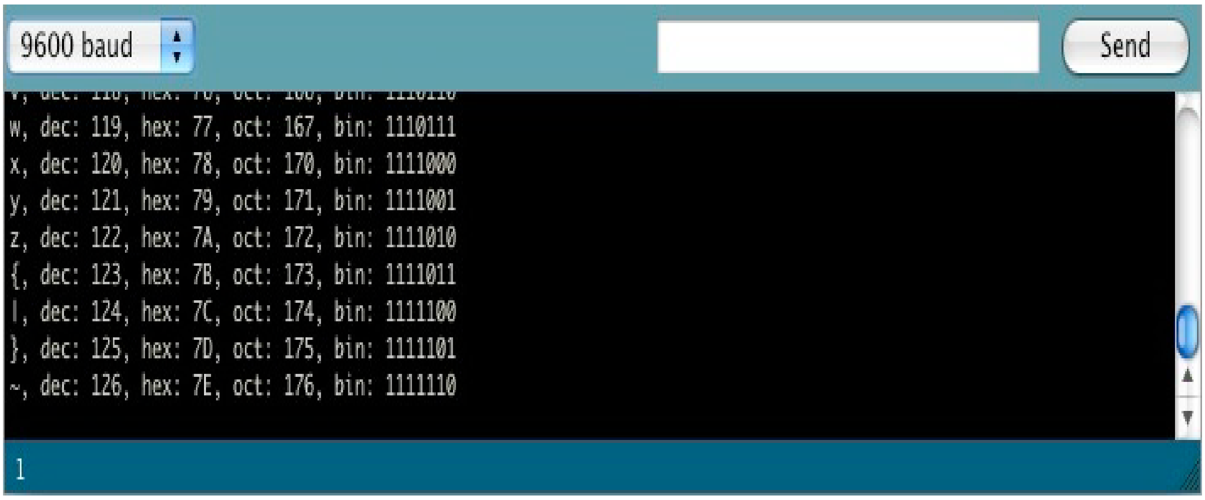


Fig: - 5.4 Serial Monitor

To the reight of this , there is a blank text boxfor you to enter text to send back to the Arduino and a send button to send the text. Note that no serial data can be received or sent by the serial monitor unless we have coded it to do so. The information which we have to display is written in this only.

5.3 Getting Started

Following are the steps involved in setup of the board and the IDE-

- Get Arduino Board or any of its clones with USB cable. In my case I have used one of its clone named ARKDUINO.

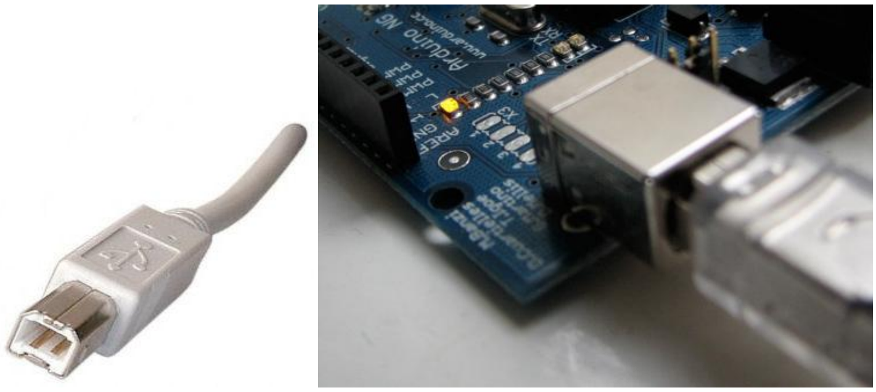


Fig: -5.5 The USB

- Download the IDE from the Arduino website. It would be ziped, unzip it.

- There will be a little power jumper (next to the USB). Using this jumper we can power it according to our need.

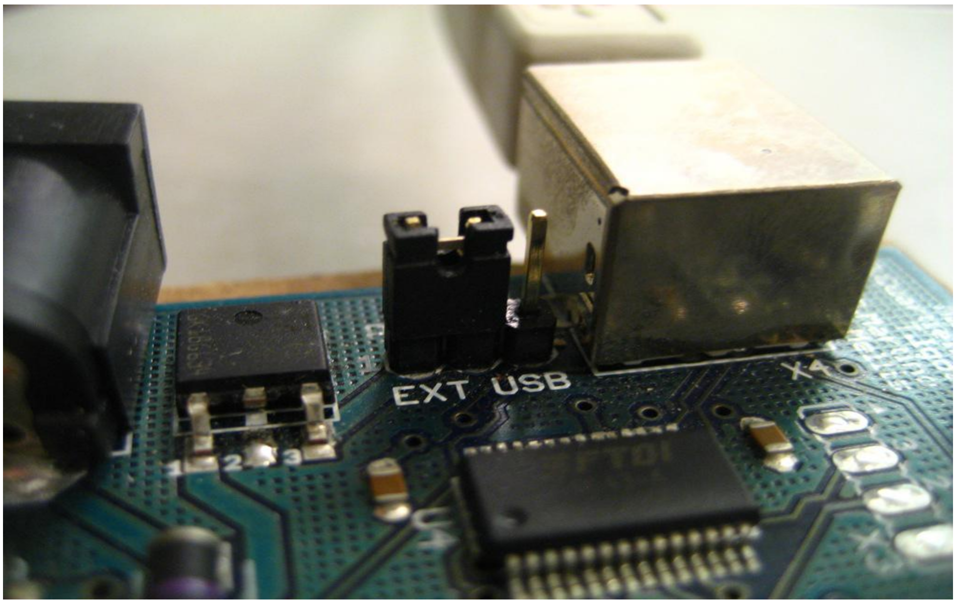
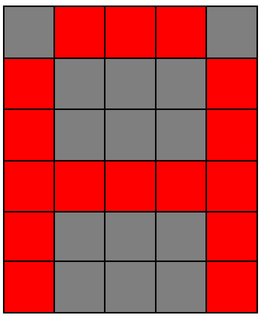


Fig: - 5.6 Power jumper

- Now install the drivers. And you are ready to upload your first sketch.

Now we can edit and burn or upload our program to the board. now click on the serial monitor and write the information which we want to display and then press enter. We can now see the information displayed by our board. after this now I will expalin a small logic which is used to program the board.



B	0	0	0	B01110000,	,
B	0	0	0	B10001000,	,
B	0	0	0	B10001000,	,
B	0	0	0	B11111000,	,
B	0	0	0	B10001000,	,
B	0	0	0	B10001000	

Here we can see that to display the A, we are having the pattern (B01110000, B10001000, B10001000, B11111000, B10001000, B10001000). Similerly we can have different combinations for different lettes, numbers and special symbols.

In Arduino we do not require external burner to dump the progam. There is a inbuilt burner made of Atmega8 on the Arduino board. Using the Arduino is much more convinent and reduces lots of labour. Now we are completed with our project, if there is no problem with the hardware as well as the software part then our project will be a complete working. In the next chapter we will be discussing about the results.

CHAPTER- 6

RESULTS AND CONCLUSION

The display board is fully functional and is able to display the both upper and lower case english letters, the numbers (0 to 9) and some of the special symbols.

Fig: - 6.1 24X6 LED DISPLAY BOARD

6.1 Conclusion:

All the three goals which we have mentioned in our introduction chapter is achieved succesfully. Following are the goals which are achieved at the end of the project-

- We have designed the display board using 144 LEDs.
- We have also made the controller circuit by using one decade counter and three shift registers to control the rows and columns repectively. The aim to use them is to reduce the number of pins required.
- After completing the hardware part, the programing of it has been done succesfully using the ARDUINO.

6.2 Future Scope:

These types of boards are very common now days and the main reason is their simplicity, less consumption of power and small size. There is chance of improvement in all the three aspeccts according to me. We can work to make them more efficient. Their size can also be reduced. However there are much smaller LEDs are there as compared to the LEDs which we have use In our project. They can be made flexible and foldable too. There are many places where they can be implimented. Instead of using the 2D display we can also make the 3D displays to show the animation.

References:

1. https://www.sparkfun.com/products/9590

2. http://www.futurlec.com/Transistors/2N3904.shtml

3. http://www.clarvis.co.uk/version2/4017.html

4. http://www.technologystudent.com/elec1/count1.htm

5. http://www.instructables.com/id/Make-a-24X6-LED-matrix/?ALLSTEPS

6. http://arduino.cc/en/Main/Software

7. http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf

8. https://www.sparkfun.com/datasheets/IC/SN74HC595.pdf

9. https://en.wikipedia.org/wiki/Soldering_iron

10. http://arduino.cc/en/Tutorial/HomePage

11. “Counter Design”, section 6-8, Digital Design, M. Morris Mano| Michael D. Ciletti, Pearson Education.