BLOCK DIAGRAM

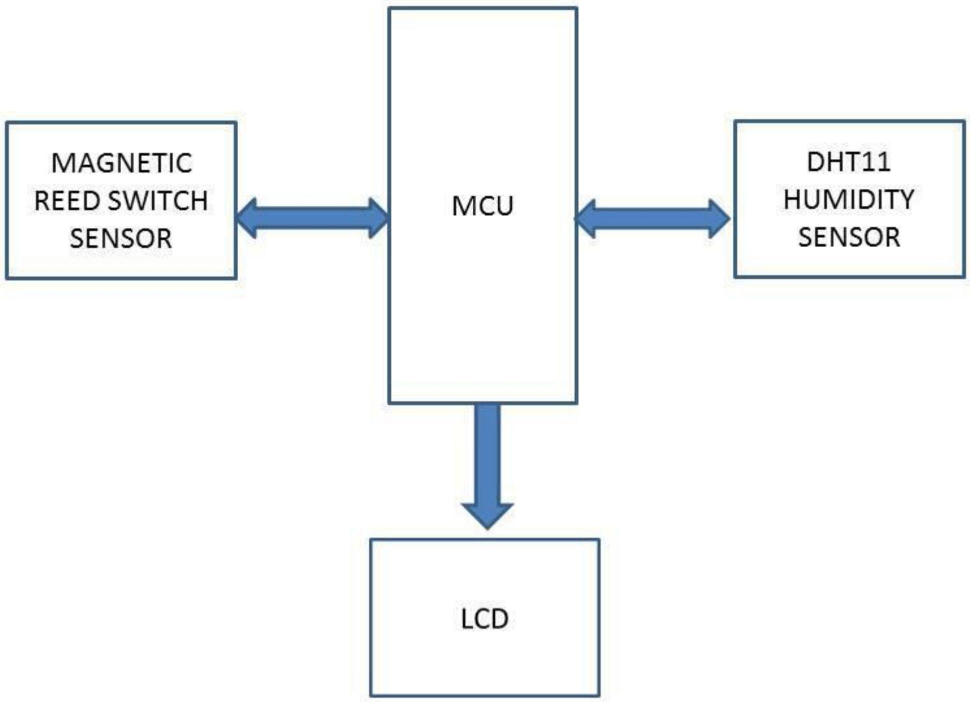

FIG 1: BLOCK DIAGRAM OF PROJECT

COMPONENT LSIT AND ITS DESCRIPTION

4.1 REED SWITCH SENSORS:

The sensors which we have used are magnetic reed switch. There is some small distance between the two poles of the magnetic reed switch. When they are exposed to external magnetic field the magnetic reed switch which have two the opposite poles are attracted towards each other and the contacts close when the force exceeds the spring force of reed switch.

Reed switches come in two varieties called normally open (normally switched off) and normally closed (normally switched on).

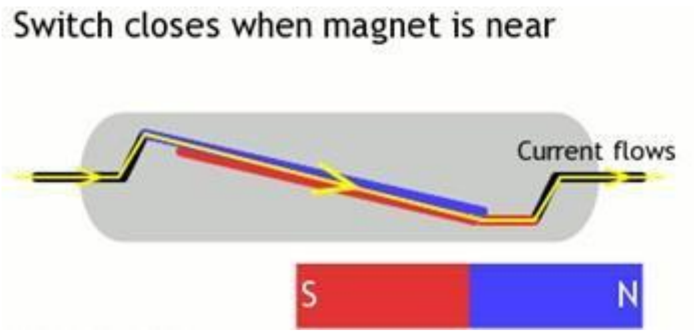

FIG 3:NORMALLY OPEN SWITCH

Normally open

In a reed switch, the two contacts (which look like metal reeds) are made from magnetic material and housed inside a thin glass envelope. (You can see this quite clearly in our top photo.) One of the contacts (sometimes called “blades”) is a magnetic north pole, while the other is a south pole. As you bring a magnet up to the switch, it affects the contacts in opposite ways, attracting one and repelling the other, so they spring together and a current flows through them. A reed switch like this is normally open (NO) (normally off), unless a magnet is positioned right next to it, when it switches on.Take the magnet away and the contacts—made from fairly stiff and springy metal—push apart again and return back to their original positions.

Normally closed

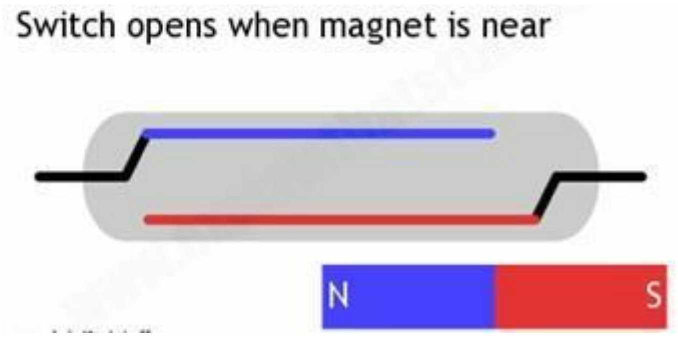

FIG 4: NORMALLY CLOSED SWITCH

You can also get reed switches that work the opposite way. The two contacts are normally snapped together. When you bring a magnet up to the switch, the lower contact is attracted to the magnet, the upper one is repelled, so the contacts split apart, opening the switch and breaking the circuit. Reed switches like this are called normally closed (NC) (normally switched on), and they switch off when you bring a magnet up to them.

Although reed switches can be designed in various ways, generally *both* contacts move (not just one) and they make a flat, parallel area of contact with one another (rather than simply touching at a point), because that helps to extend the life and reliability of the switch. Also, where I've exaggerated the movement of the contacts to make it easier to see, real reed switches have contacts that are only a few microns (millionths of a meter) apart—roughly ten times thinner than a human hair—so the movement isn't visible to the naked eye.

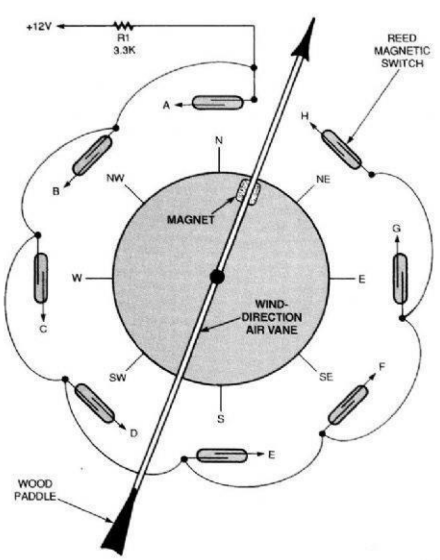

FIG5:ARRANGEMENT OF SENSORS

4.2 DIODE MATRIX LOGIC:

Diode matrix is used as control store or micro program in many early computers. At any instant any single row of diode is active. Here diode matrix is used as a encoder. The output of diode matrix is in the form of bcd.Suppose M1 switch is closed so values ofA,B,C will be 111 so 0111 will be at the output of diode matrix.

Where encoders are needed for non-standard applications, they can also be implemented using a diode matrix, such as the decimal-to-BCD_{2421} encoder.

In this example, as any one of the ten switches is closed +5V is an applied to just one of the horizontal line. Any diode that has its anode connected to that horizontal line and its cathode connected to a vertical line (that is held at zero volts by a resistor connected to Gnd) will conduct.

When current flows through any of the resistors, the top of that resistor will be at +4.4V (i.e. +5V minus a 0.6V drop in across the diode), which will be seen by the output as logic 1.

For example if switch 6 is closed, the two diodes connected between line 6 and columns X_3 and X_2 will conduct, making outputs X_3 and X_2 logic 1 and giving a $binary_{2421}$ output word of 1100_2 (or 2+4 $=6_{10}$).

This particular diode matrix will therefore give an output in BCD_{2421} code from 0000_{2421} to 1111_{2421} for closure of switches 0 to 9.

Many other output sequences are possible therefore, by using different arrangements of the diode positions.

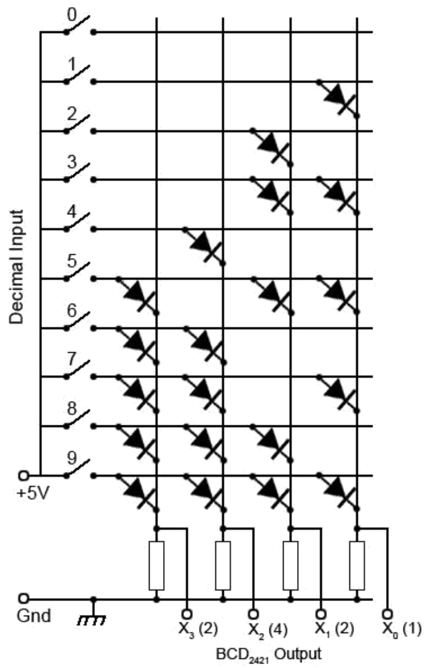

FIG 6 : CIRCUIT OF DIODE MATRIX

	MSB	BCD_{8421}		LSB
Decimal	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

FIG 7: TRUTH TABLE OF DIODE MATRIX

4.3 DHT11 (HUMIDITY SENSOR)

This project is about a simple humidity sensor based on 8051 microcontroller. Humidity sensor is also called hygrometer. This circuit can sense relative humidity (RH) from 20% to 95% at an accuracy of 5%. The humidity information is displayed on a 16×2 LCD display. A relay is also provided which is set to be active when the humidity crosses a certain trip point. The circuit is mains operated and it is very easy to install. DHT11 is the humidity sensor used here. The details and working of the DHT11 humidity sensor is given below.

DHT11 is a low cost humidity cum temperature sensor which has a digital output. Capacitive method is used for sensing the humidity and a thermistor is used for measuring the temperature. The sensor can sense relative humidity from 20% to 95% at a resolution of 5%. Temperature measurement is up to 50°C at a resolution of 2°C. The communication with the microcontroller is through a single wire. The basic communication scheme is given in the image below.

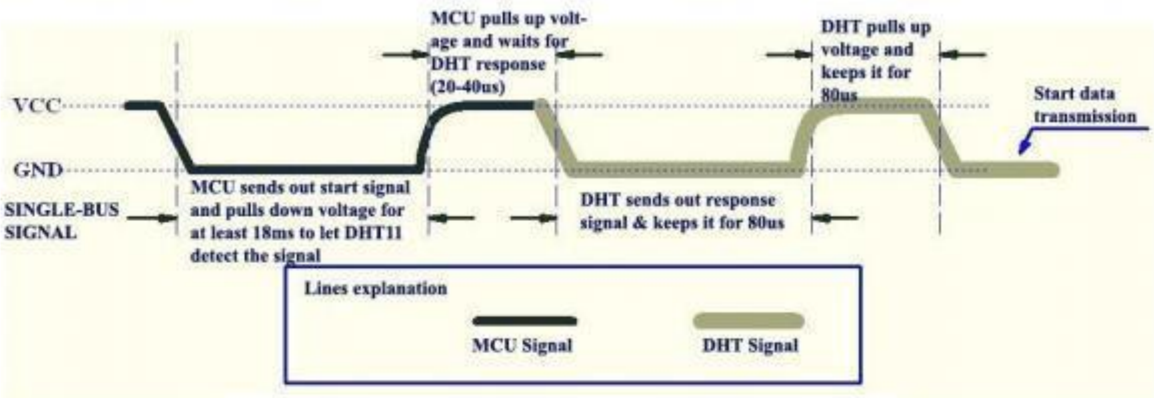

FIG 8: TIMING DIAGRAM OF DHT11

The to and fro communication with DHT11 sensor is very easy. Pin 2 of the DHT11 is connected to the port pin of the 12icrocontroller. The connection scheme is shown in the image below. The data pin (pin2) of the DHT11 requires an external 10K pull-up resistor.

The communication protocol is explained as follows. The MCU (microcontroller unit) first sends a low signal of width 18Ms to the DHT11. After this signal, the MCU pulls up the communication line and waits for the response from DHT11. It make take up to 2. To 40Us. Then the DHT11 pulls down the communication line and keeps it low for 80Us. Then DHT11 pulls up the line and keeps it high for 80Us. Then the DHT pulls down the line for 50Us and the next high pulse will be the first bit of the data. The data is send in bursts of 8 bits. Each high pulse of the burst indicates a data signal. The 50Us low signals between the data bits are just spacers. The logic of the data bit is identified by measuring the width of it. A 26 to 28Us wide pulse indicates a “LOW” and 70Us wide pulse indicates a “HIGH”. In simple words, an pulse narrower than 50Us can be taken as a “LOW” and wider than 50us can be taken as a “HIGH”.

The first 8 bits of the data burst represents the integral value of the relative humidity, second 8 bits represent the decimal value of the relative humidity, third 8 bits represent the integral value of the temperature data, and the last 8 bits represent the decimal value of the temperature data, For DHT11 the decimal values are always zero and we are measuring the relative humidity only in this project. So we need to just concern about the first 8 bits of data, that is the integral part of the relative humidity data.

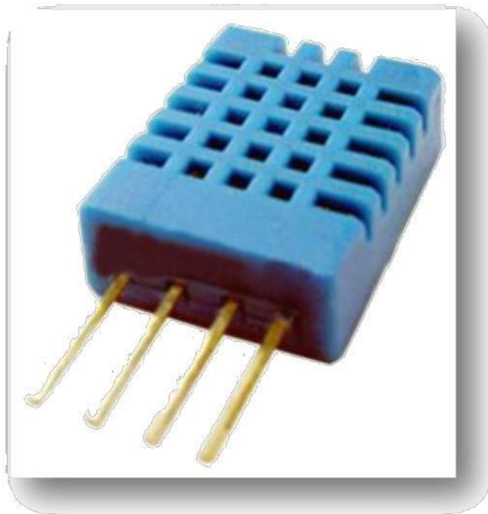

FIG 9: DHT11

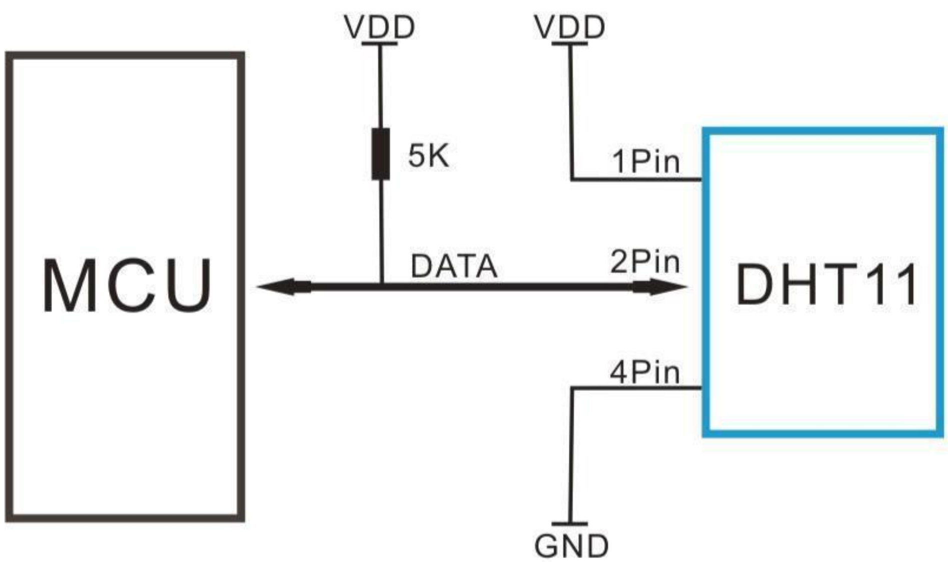

FIG 10: INTERFACING DHT11 WITH MCU

4.4 Atmel 89S52 Microcontroller:

Features

- 8K Bytes of In-System Programmable (ISP) Flash Memory – Endurance: 10,000 Write/Erase Cycles.
- 4.0V to 5.5V Operating Range.
- Fully Static Operation: 0 Hz to 33 MHz.
- Three-level Program Memory Lock.
- 128 x 8-bit Internal RAM.
- 32 Programmable I/O Lines.
- Two 16-bit Timer/Counters.
- Six Interrupt Sources.
- Full Duplex UART Serial Channel.
- Low-power Idle and Power-down Modes.
- Interrupt Recovery from Power-down Mode.
- Watchdog Timer.
- Dual Data Pointer.
- Power-off Flag.
- Fast Programming Time.
- Flexible ISP Programming (Byte and Page Mode).
- Green (Pb/Halide-free) Packaging Option.

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

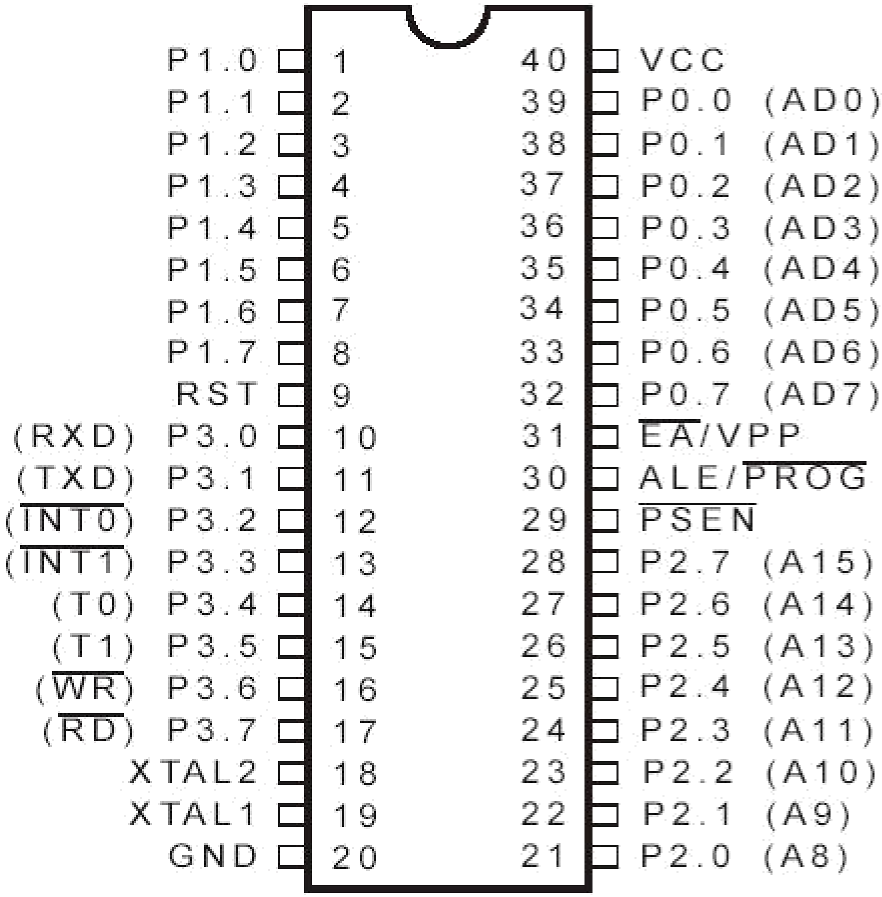

FIG 11 : PIN DIAGRAM OF AT89S52

4.5 LCD(LIQUID CRYSTAL DISPLAY)

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.

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.

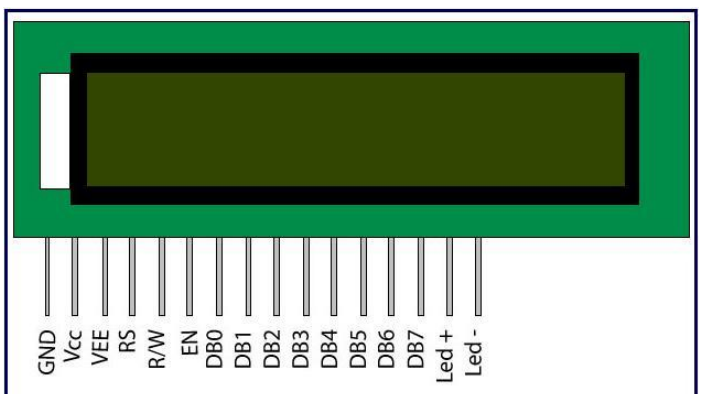

FIG 12: LIQUID CRYSTAL DISPLAY

Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and 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	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

4.6 OTHER COMPONENTS

Along with the above mentioned components we have also used following components:

- General Circuit Board
- Soldering Iron
- Connecting wires
- LED
- Push button switch
- Crystal
- Capacitors
- Resistors

PROGRAM LOGIC AND WORKING

Data out pin of the DHT11 is connected to P3.5 of the microcontroller. At the start of the MAIN loop, P3.5 is held high. Then it is made low and an 18Ms delay routine (DELAY1) is called. Then P3.5 is made high. This forms the first 18Ms wide start signal for the DHT11. Now the communication line is high and the microcontroller polls the status of this line and waits there until it is low. It becomes low when the DHT11 sends back the response signal. Then the microcontroller waits for the second response signal which is a high signal. When this high signal is received the microcontroller waits for the next low signal and after this low signal, data transmission starts.

I have already mentioned that, each data bit is represented by a high signal and the width of it determines the logic. So whenever the microcontroller receives a data pulse, Timer1 is started and the program waits there until the data pulse vanishes. Then the timer is stopped. Now the TL1 i.e. Timer1 low register contains the width of the data pulse. Then the width is compared with 50 by subtracting 50 from the TL1 count. If carry flag (PSW.7) is set it means that the width is less than 50Us and if carry flag is not set it means that the width is higher than 50Us.

If width is less than 50Us it indicates a low signal and ACC.0 is cleared. If the width is higher than 50Us it indicates a high signal and ACC.0 is set. Then TL1 register and PSW.7 bit are cleared. Then the LOOP is iterated 8 times and accumulator is rotated left during the beginning of each iteration. The rotate left (RL) instruction is used because during each iteration you updates ACC.0 only and you have to save that bit before the next update. This is achieved by shifting it to left each time.

The last update of ACC.0 does not require a position shift because it is in the right place it is supposed to be. That is the reason behind placing the RL A instruction at the beginning of the LOOP. After the 8 th iteration the LOOP is exited without rotating the accumulator. Now the content of the accumulator is equal to the integral value of the current relative humidity in percentage. Then subroutine DINT is called for initializing the display. Then subroutine TEXT1 is called which displays “Hygrometer”. Then subroutine LINE2 is called for shifting the cursor to 2nd line. Then subroutine HMDTY is called for displaying the humidity value in percentage. Then subroutine CHECK is called for checking whether the humidity value is above or below 65%. If humidity is above 65%, relay is activated and else the relay is deactivated. The 2S delay subroutine DELAY2 is called. 2 second delay is given because you can take readings from the DHT11 once in every 2 seconds only. This also makes the display stable. Then the program jumps to MAIN label and the entire process is repeated.

The output from reed switch sensor are given to diode matrix for encoding which are then feed to the microcontroller. A is connected with P1.0,B is connected to P0.1 and C is connected to P0.2.

ASSEMBLED PROJECT LAYOUT

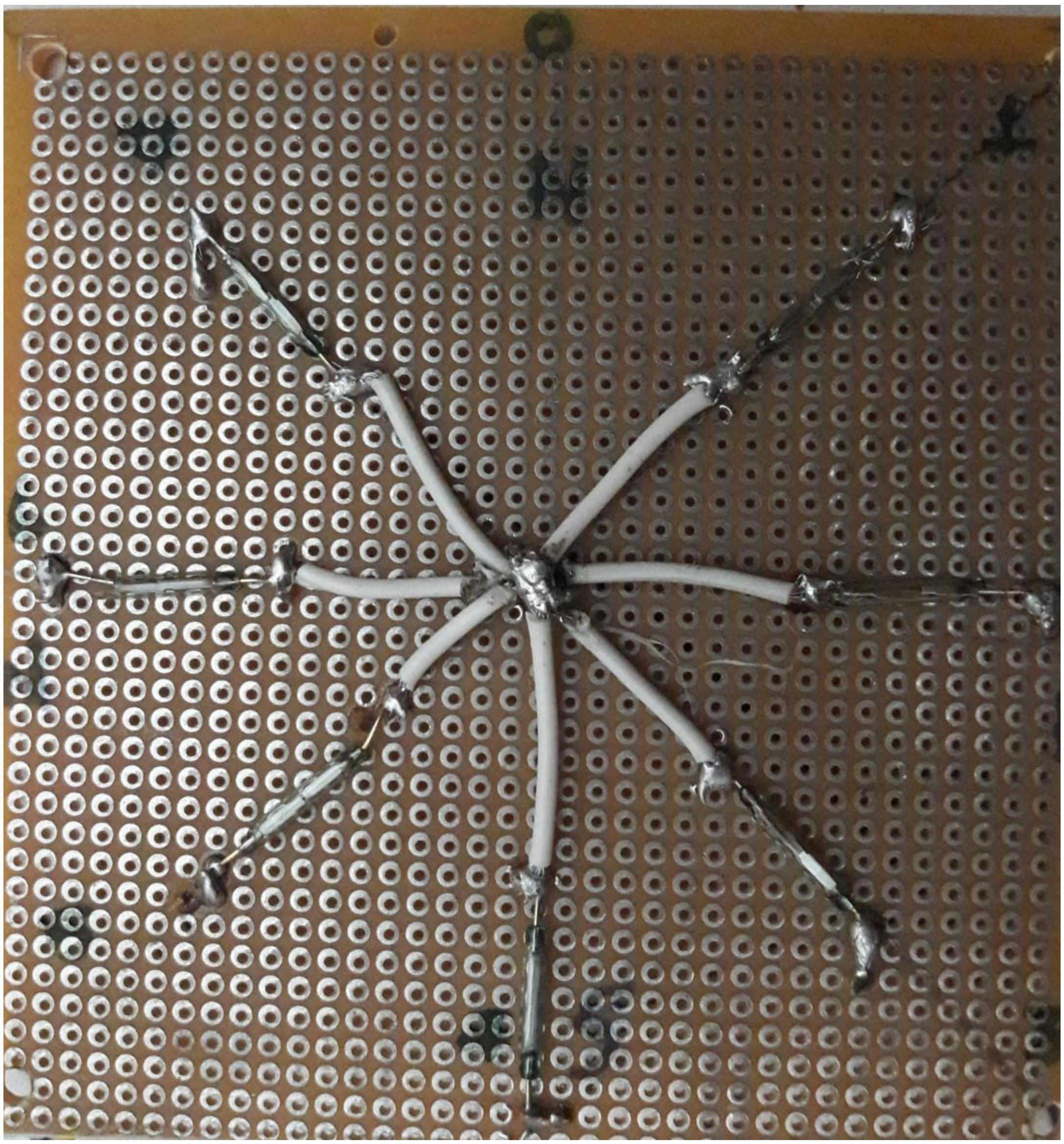

FIG 13: REED SWICH SENSOR CIRCUIT

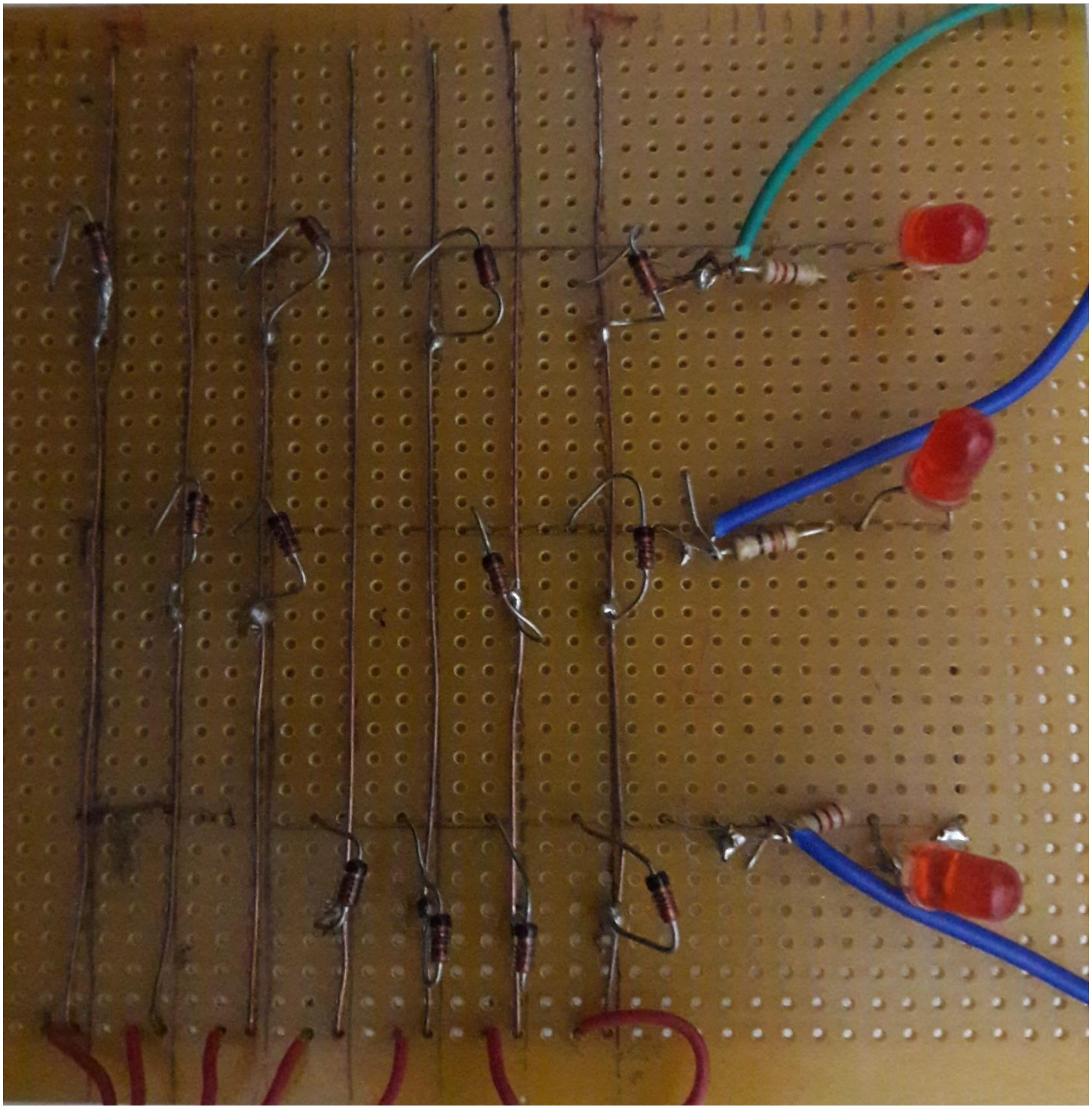

FIG 14: DIODE MATRIX

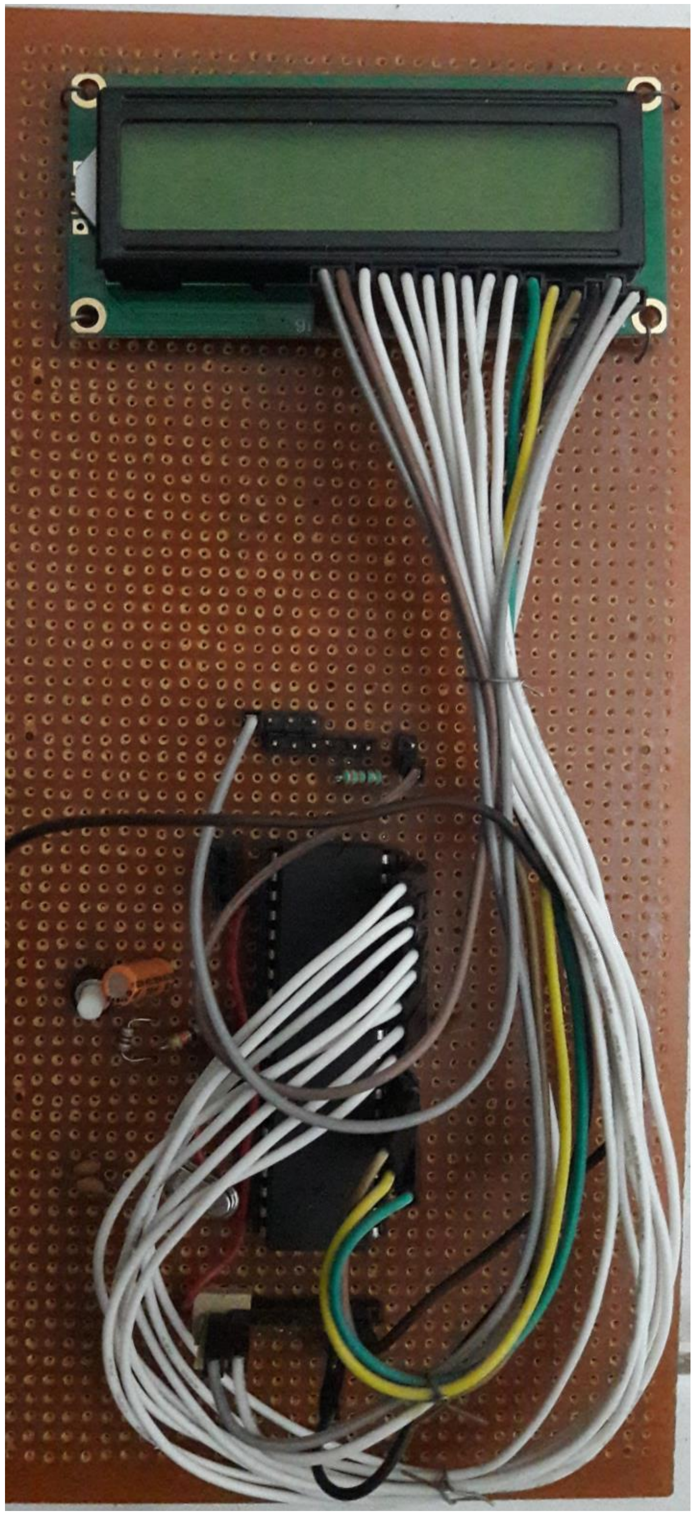

FIG 15: FRONT SIDE

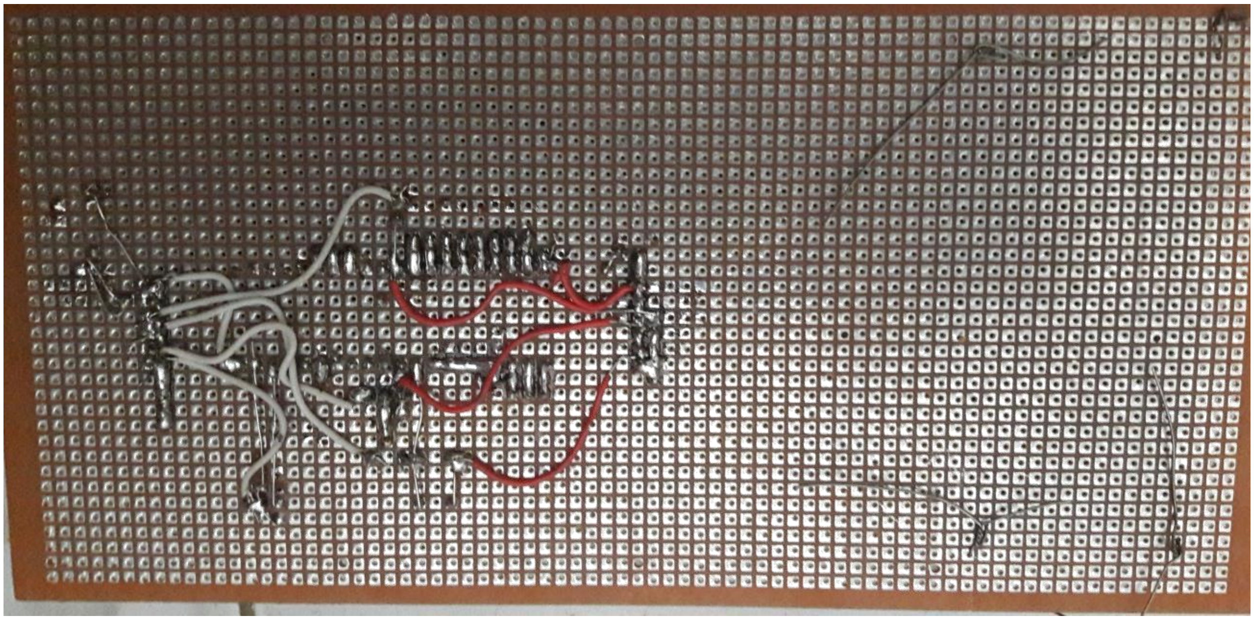

FIG 16 : SOLDERING SIDE