A PROJECT REPORT ON

TEMPERATURE CONTROLLED DC FAN USING MICROCONTROLLER

Group S-4

Name: Tarala Trilokesh

Roll: B190367EC

Email: tarala_b190367ec@nitc.ac.in

Name: T.V.S Tanuj

Roll: B190875EC

Email: tanuj_b190875ec@nitc.ac.in

Name: Topireddy Janardhan Reddy

Roll: B190167EC

Email: topireddy_b190167ec@nitc.ac.in

Name: Swagath N S

Roll: B191069EC

Email: swagath_b191069ec@nitc.ac.in

Name: Swarnendu Mondal

Roll: B190772EC

Email: swarnendu_b190772ec@nitc.ac.in

Name: Mahesh Reddy S

Roll: B180743EC

Email: mahesh_b180743ec@nitc.ac.in

Department of Electronics and Communication Engineering

National Institute of Technology Calicut

Introduction:

The aim of the project is to control the speed of a Fan (DC Motor) according to the temperature of the environment. The more the temperature the more will be the speed of the fan and maintains the coolness of the surroundings instead of manual control. The fan doesn't go on for temperature below 0°C and slowly varies its speed from minimum to maximum over 0 to 150°C. The circuit is simulated in Proteus Simulation Software.

Circuit Principle:

The main principles used in the project are ADC (Analog to Digital Conversion), DAC(Digital to Analog Conversion) and PWM (Pulse Width Modulation). The operation of the circuit starts from reading the temperature of the environment using the LM35 Temperature Sensor. It is connected to ADC0804 which converts the analog signal of the sensor into digital. The digital output of the ADC is sent to the 8051 MC, which calculates the temperature by scaling it and then certain logic has been implemented to find the duty cycle of the PWM Pulse. Based on the duty cycle, we will know the time duration of the high and low pulses of the PWM pulse. We will be using timer 0 in mode 3 to on the motor and off the motor for the times calculated above. We used the motor driver L293D which is an H-Bridge Motor driver, which connects the DC Motor (to replicate a fan) and supplies the motor with sufficient current and power to run.

The Components used in the circuit are -

LM35

The LM35 is a temperature sensor with precession whose output voltage varies depending on the temperature around it. It's a small integrated circuit that can test temperatures from -55°C to 150°C. It can be easily connected to any microcontroller with an ADC feature, as well as any development platform such as Arduino.

If the temperature is 0 degrees Celsius, the output voltage would also be 0 degrees Celsius. For every degree Celsius increase in temperature, the voltage will rise by 0.01V (10mV).

The voltage can convert into temperature using the below formulae.

Vout = 10mv/°C × T

ADC0804

The ADC0804 is a popular ADC module for projects that require an external ADC. It's a single channel 8-bit ADC module with 20 pins. It can calculate one ADC value from 0V to 5V with a precision of 19.53mV when the voltage reference (Vref –pin 9) is +5V. (Step size). That is, for every 19.53mV increase on the input side, the output side would increase by one bit.

This IC is ideal for use with microprocessors such as the Raspberry Pi, Beaglebone, and other similar devices. Alternatively, it can be used as a stand-alone ADC module. Every ADC requires a clock to function. Here the advantage is the clock comes inbuilt for this IC.

L293D

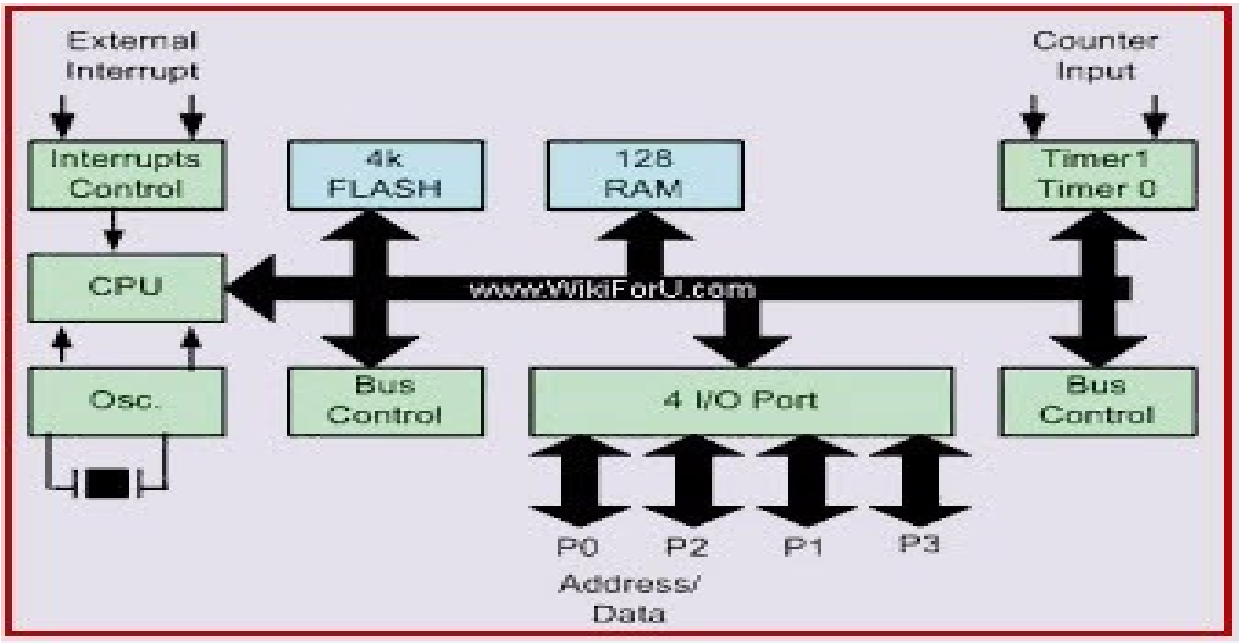
The L293D is a 16-pin motor driver IC that is widely used. It is primarily used to drive motors, as the name implies. A single L293D IC can drive two DC motors at the same time, and the two motors' directions can be operated independently. So, if we have motors with an operating voltage of less than 36V and a current of less than 600mA, and we want to power them with digital circuits like Op-Amps, 555 timers, digital gates, or even Micron rollers like Arduino, PIC, ARM, and so on...

8051

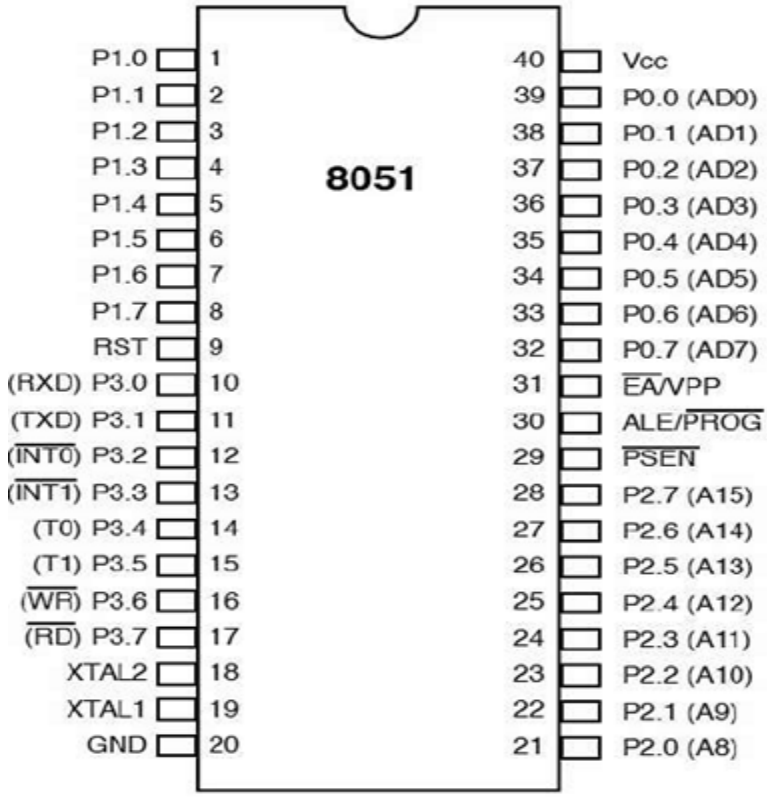
The 8xC51 contain a 128 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a six-source, four-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuit.

In addition, the device is a low power static design which offers a wide range of operating frequencies down to zero. Two software selectable modes of power reduction—idle mode and power-down mode are available. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

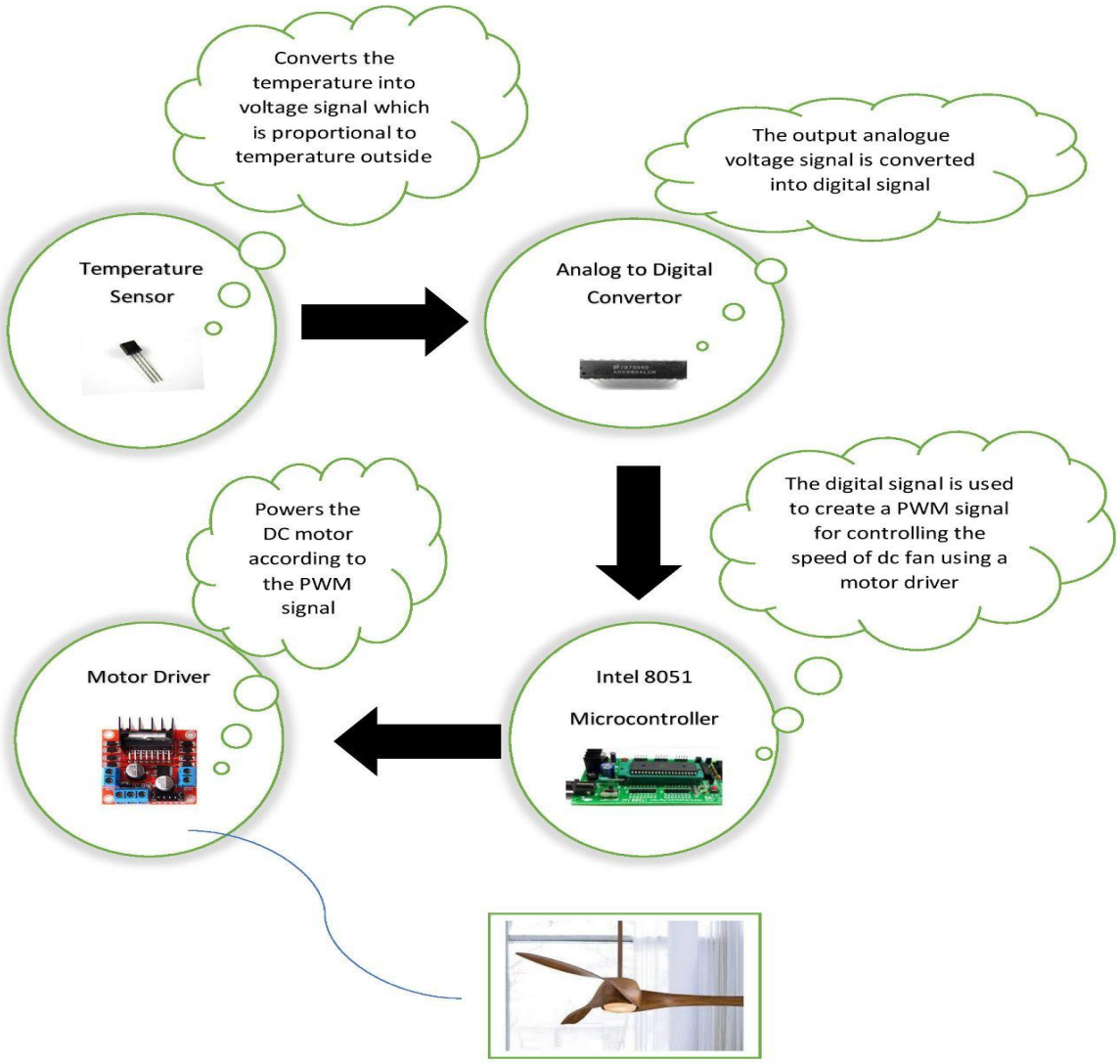
Block Diagram of 80C51



Pin Diagram of 80C51

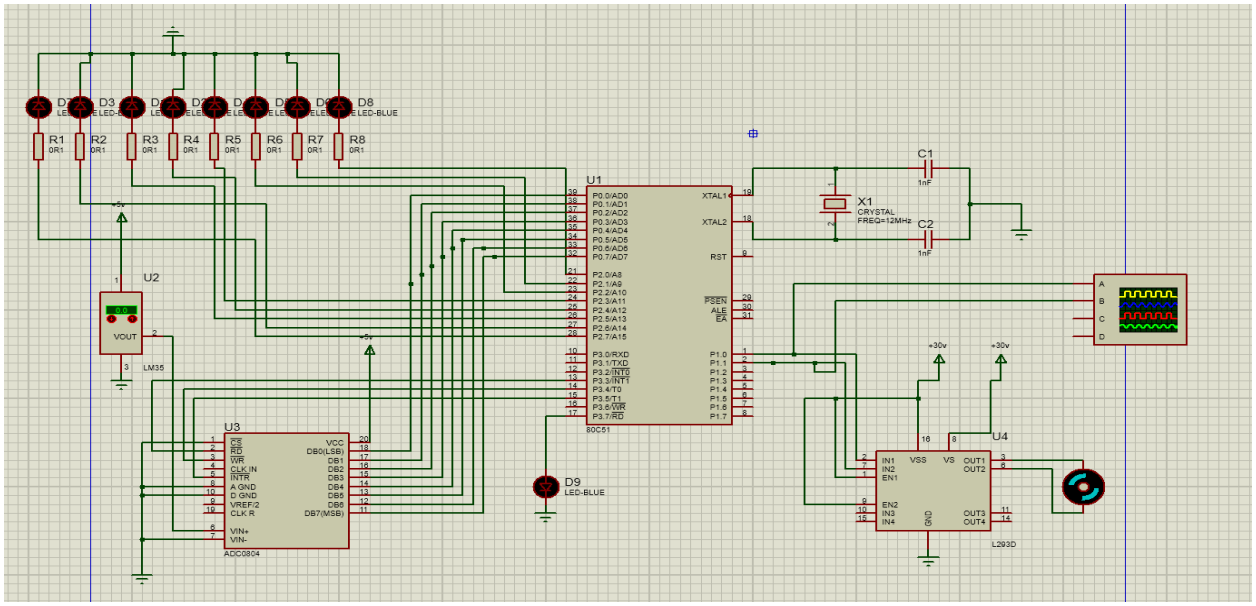


Block Diagram of Temperature controlled DC fan motor :



Circuit Design:

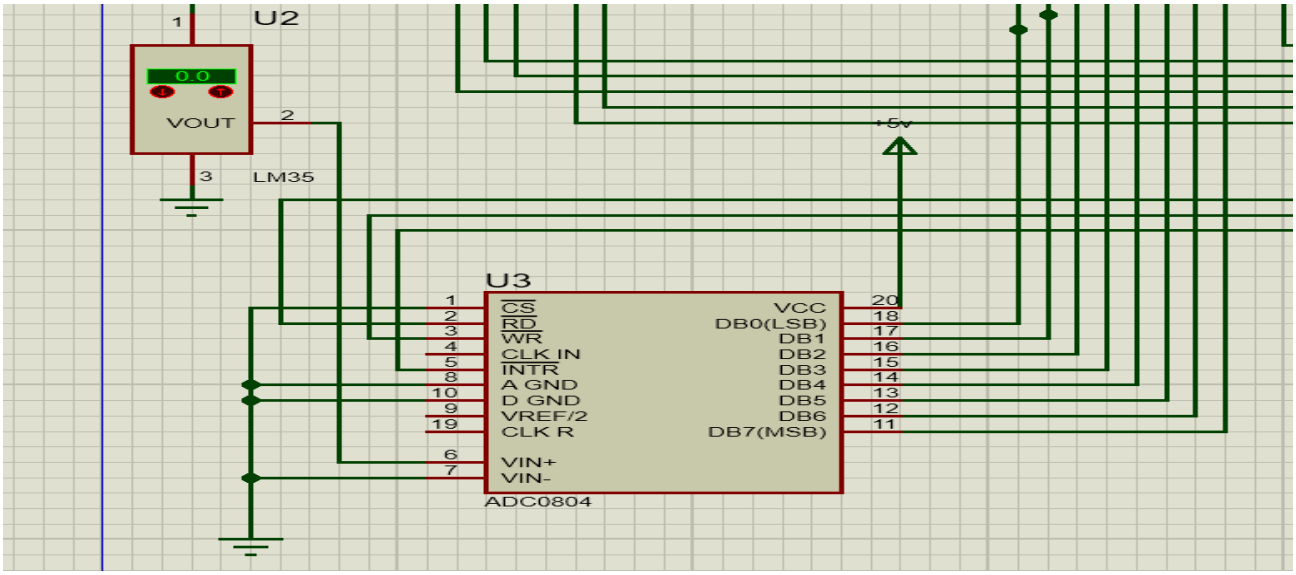
The whole circuit has been simulated in the Proteus Simulation Software. The reason to choose Proteus over EdSim 51 is that it has a temperature sensor and all the components of the circuit can be shown clearly, whereas Edsim51 is a default circuit containing some other components which are not used in the circuit, creating a confusion in the reader's head. A DC Motor is used here to replicate a fan.



Circuit Diagram of the Temperature Controlled Fan

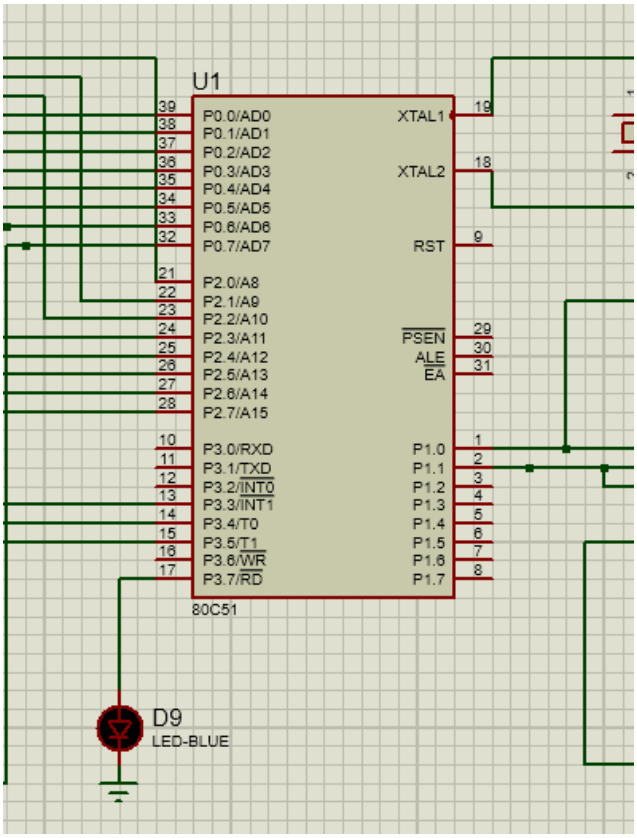
The working of the circuit is explained in the following steps, each describing the specific part of the circuit

1) Temperature Detection & ADC Conversion



The above picture contains the LM35 Sensor and ADC0804 Converter. The sensor detects the temperature of the environment and sends the analog value of that to the ADC Converter. The converter converts that analog value into digital value and sends it to the 8051 MC.

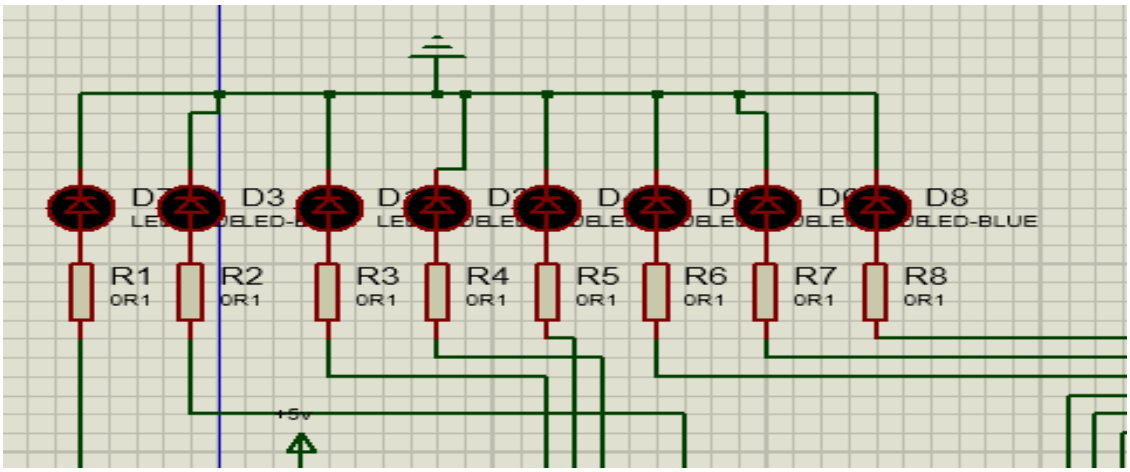
2) 8051 MC and Digital Value Indicator



The picture in the left indicates the 8051 MC which is the heart and brain of the whole project. It receives the digital output of the ADC through the P0 port. We then used the logic to scale the temperature value to 255, which should be the input for the PWM pulse.

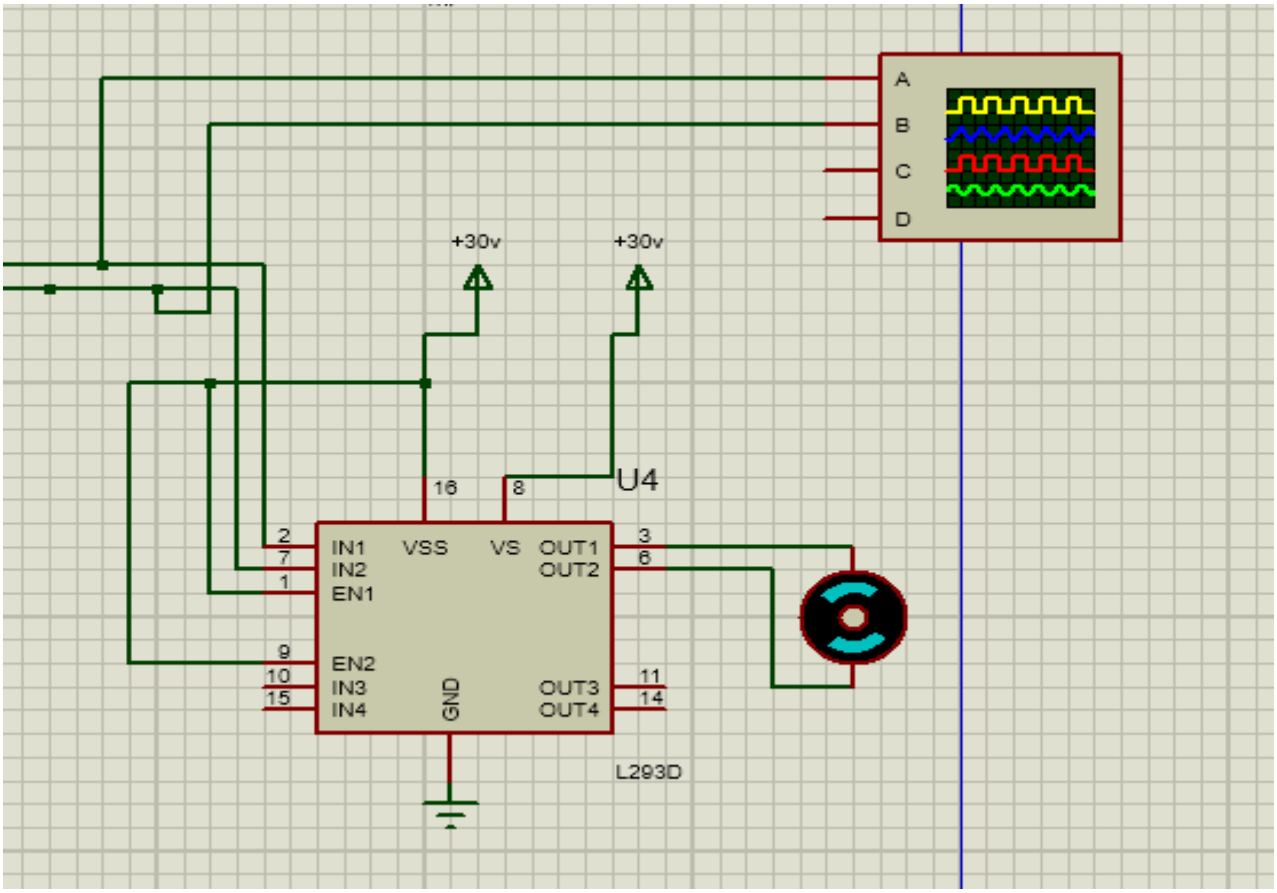
From the observation we did, we have to multiply the digital value by 3 to scale the value to 255.

We loaded that value into Port P2, which is connected to a series of LED's to indicate the digital value of temperature scaled to 255. The LED connected to P3.7 is used to detect if there is any fault in the circuit or the MC.



The above picture shows the digital value of temperature scaled to 255 in the form of an 8-bit binary number, which is connected to Port 2. The resistors used act as pull up resistors to provide LED's with required amounts of current. The left-most led is the LSB and right-most led is the MSB for the 8-bit binary number.

3) Motor and Motor Driver



The above picture contains DC Motor which is connected to the 8051MC via L293D Motor Driver to provide the DC Motor with sufficient current. Here, we are rotating the motor in an A.C.W sense, so for that IN2 pin must be high and IN1 pin must be low. The IN1 pin is always set to low and we control the IN2 pin using PWM.

The box kind of thing in the top right corner is the Digital Oscilloscope, which is used here to capture the PWM Pulse, which essentially is the output of the circuit.

Logic:

1. LM35 will convert the temperature to analogue voltage signal using the following logic :

   $for\ T = 0°C \Rightarrow voltage = 0;$

   $for\ T > 0°C \Rightarrow voltage = (T * 10)\ mV$

2. This analogue Voltage signal is given as input to ADC0804 (analogue to digital convertor) which will convert signal to digital signal based on the signals from 8051 microcontroller. 8051 sends the following to ADC to complete the conversion:
   a. Set the read bar pin of ADC0804 to disable data lines.
   b. Connect the CS(chip select) pin to ground which will make ADC0804 on.
   c. Set the EOC(INTR bar here) pin because the ADC will make this pin LOW after the conversion of analog signal into digital.
   d. To start conversion we need to give a low to high pulse to write bar, so clear it first and then Set it.
   e. Check for the EOC pin to become HIGh to LOW which denotes that the conversion is complete.
   f. Clear the read bar pin to enable Data lines, which will send the Digital signal of analog signal of LM35 to Port 0 of intel 8051 microcontroller.
3. A led is connected to P3.7 of 8051 microcontroller which is used to check whether the circuit is functioning or not, if it glows it denotes that microcontroller is functioning correctly otherwise it denotes that it has a fault.

4. We want to create a PWM signal whose duty cycle is proportional to temperature, For that we are using the timer 0 in mode 0 which counts 0 to 255, the max value of digital signal from LM35 is 85 approx so we multiplied it by 3 to make it approx. 255, and this value is now used to create the PWM signal, the motor will be on for this many seconds (i.e., the digital signal value from adc * 3). and it is off for the remaining time (i.e., 256 minus(-) its ON time)
5. This digital signal value (from adc * 3) is outputted to Port 0 which is connected to Leds connected in common anode mode and and a logic 1 should be driven from the microcontroller pin in order to glow the LED. This can be used for visualising the duty cycle of PWM signal.
6. This PWM signal is fed to LM293D motor driver, The IN1 pin is always set to low and we control the IN2 pin using PWM by connecting it to P1.1(PWM is generated at this pin in the 8051). Thus the fan rotates according to PWM signal.

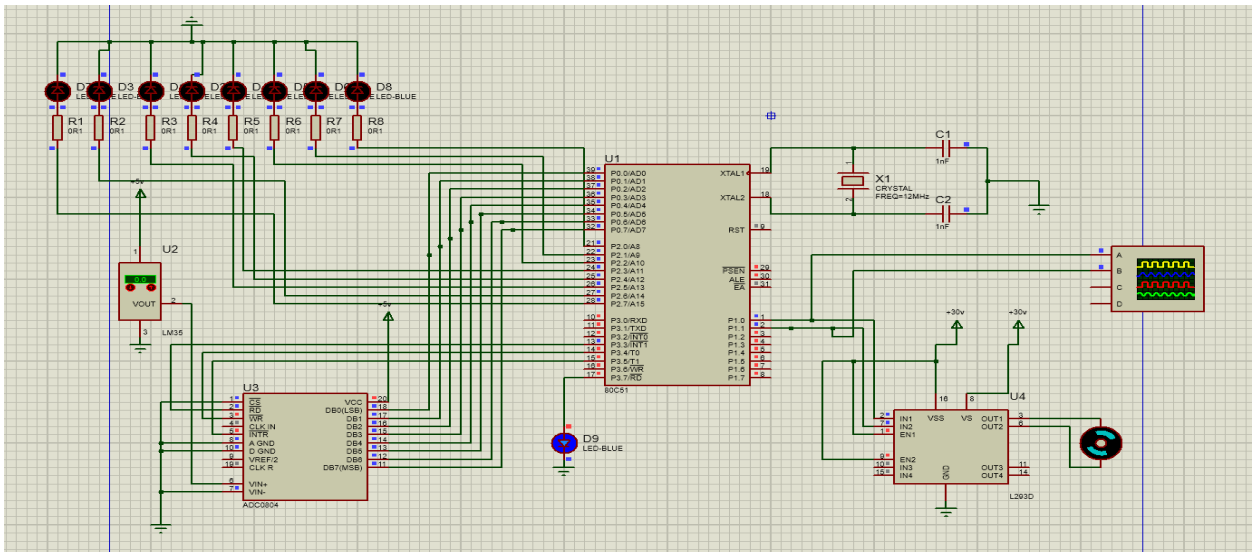
Programming :

```
      org   0000h
Start:
      SETB P3.3 ; set the read bar pin to disable the data lines
      CLR P3.4   ; clear the write bar pin
      SETB P3.5 ; set the EOC pin
      SETB P3.4 ; set the write bar pin (giving a low to high pulse to start conversion)

stay: JB P3.5, stay ; wait till the analog to digital conversion is complete
      CLR P3.3     ; clear read bar pin to enable data lines this will send output to 8051
      SETB P3.7   ; set this pin to glow a led connected to this pin
     MOV TMOD,#00H ; Timer0 in Mode 0
     MOV R1, P0           ;digital signal's value is stored in R1 register
     MOV A, R1          ;and then is copied to A register  for multiplying operation
     MOV B, #3          ;to multiply A with 3 B is  loaded with 3
     MUL AB              ; multiplying with 3
     MOV P2,A     ; this multiplied value is then copied to P2 which is connected to leds
     CPL A           ; this multiplied value is complemented for loading it into Timer0
     MOV R7, A     ;the complemented value is copied to R7 register
     SETB TR0      ; timer 0 is started

     CLR P1.1                  ;Clearing the PWM in
LCALL HIGH_DONE         ;Calling the procedure to perform High pulse
     SETB TR0                  ;Starting the Timer
LCALL LOW_DONE          ;Calling the procedure to perform Low pulse
     JMP Start                 ;Jumping to the Start

HIGH_DONE:
     SETB P1.1                  ;Setting the PWM Pulse to High
     CLR TF0                    ;Clearing the Timer Flag
     MOV TH0, R7                ;Moving the appropriate value to set the Timer
      stay1: JNB TF0, stay1     ;Waiting till the High Pulse is completed
     CLR TF0                    ;Clearing the Timer Flag
     RET
```

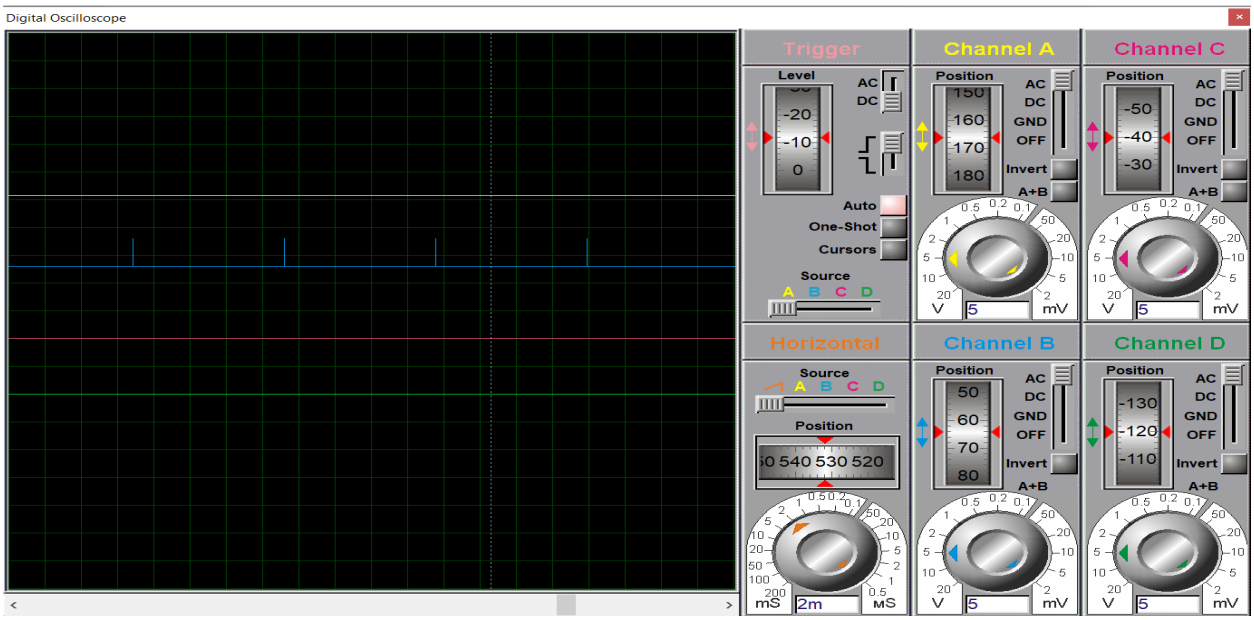
```
LOW_DONE:
    CLR P1.1              ;To lower the PWM Pulse
    MOV A, #0FFH          ;Toading 255 into A
    CLR C
    SUBB A, R7            ;Finding the duration of time to lower the PWM pulse
    MOV TH0, A            ;Loading the Timer with required value
    stay2: JNB TF0, stay2 ;Waiting till the low Pulse is completed
    CLR TF0               ;Clearing the Timer Flag.
    RET                   ;Returning to the Main Function.



    END                   ;End of the Code
```

Results

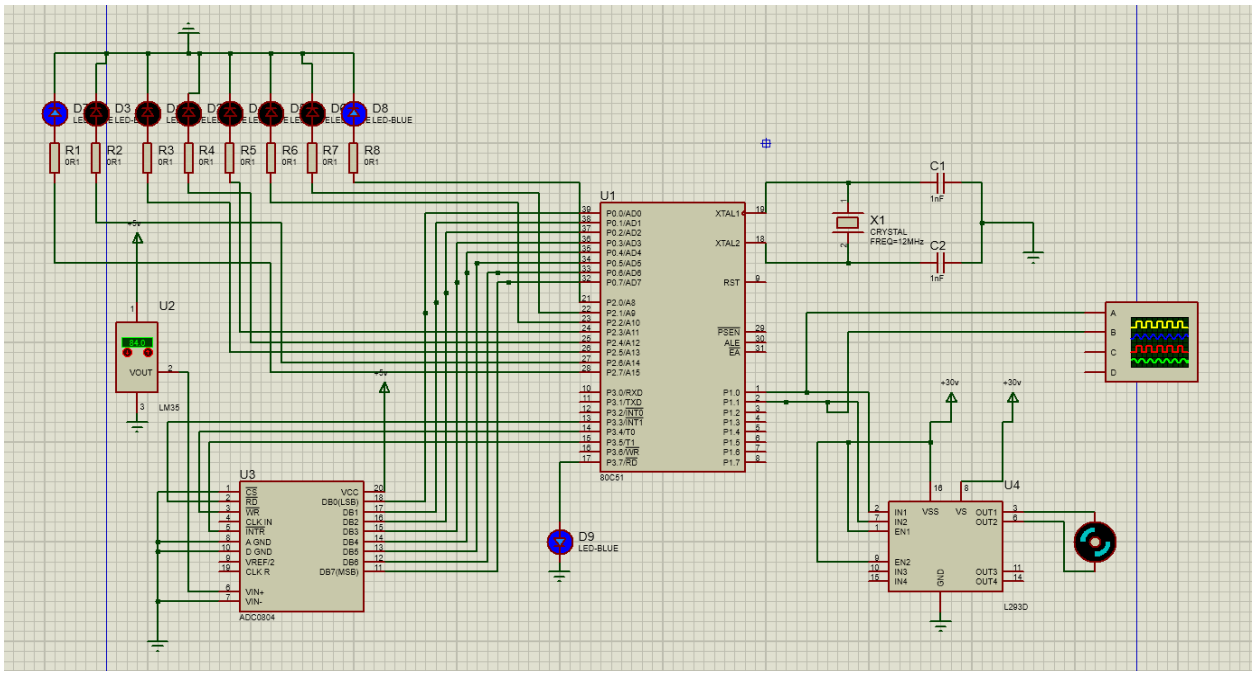
In all the results shown below, the blue line represents the PWM pulse, which varies the speed of the motor. Since in screenshots, the speed of the motor cannot be shown, Duty Cycle of the PWM pulse is itself the output. The yellow signal is always low, since the motor is A.C.W sense.



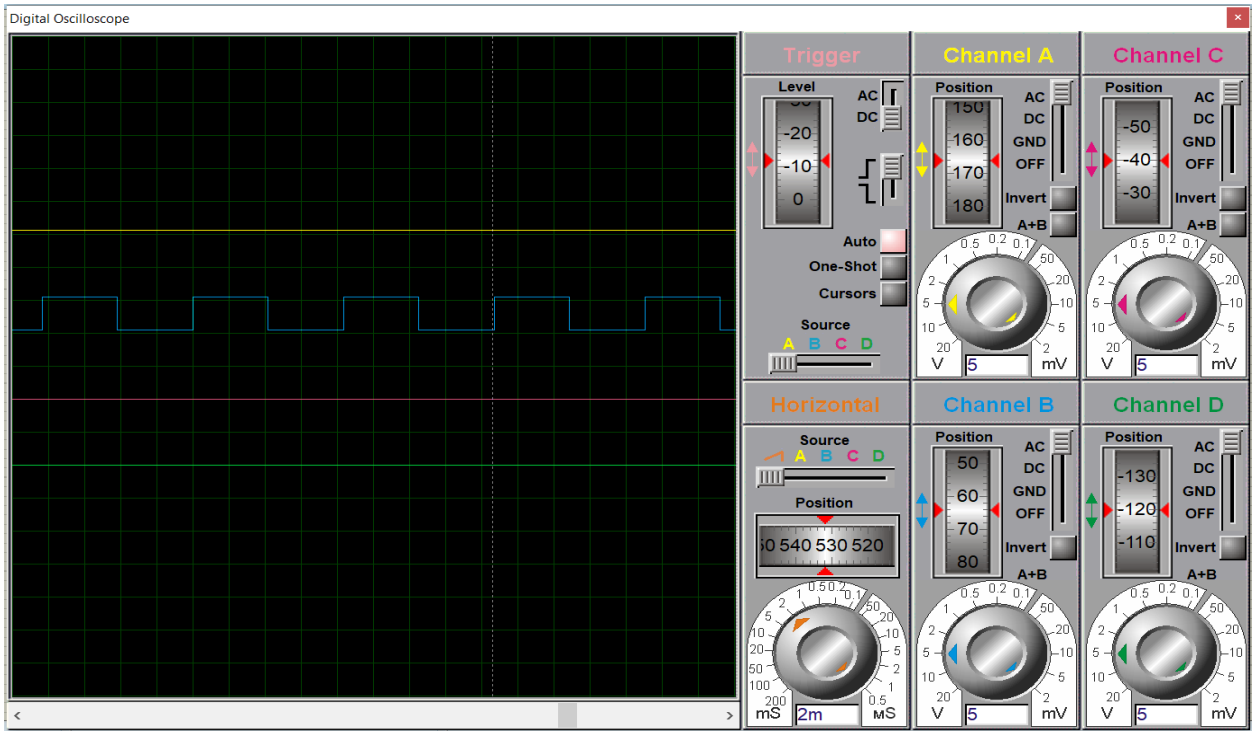
Simulation of the Circuit when the temperature of the sensor is set to 0°C. It can be seen that the digital output of the temperature is 0 (from the LED's)



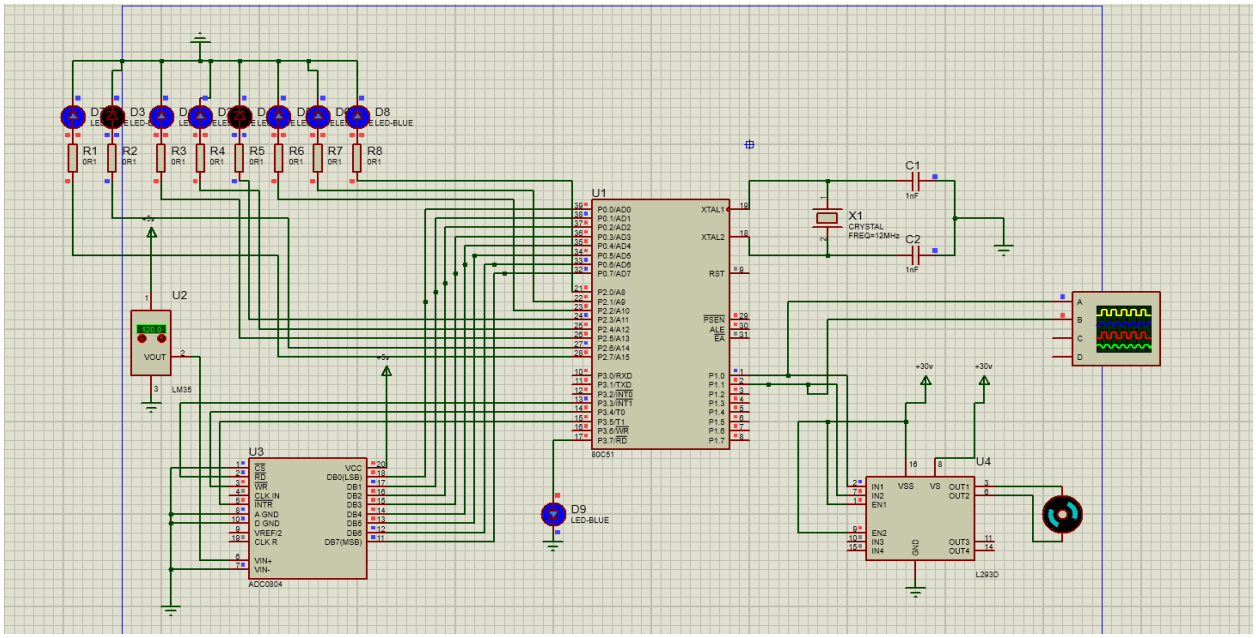
It can be seen that the width of the PWM Pulse if 0 (blue graph)



Simulation of the Circuit when the temperature is set to 80°C. It can be seen that the LED's show a value of 128, which is half of 255, so is 80°C which is half of the max temperature.

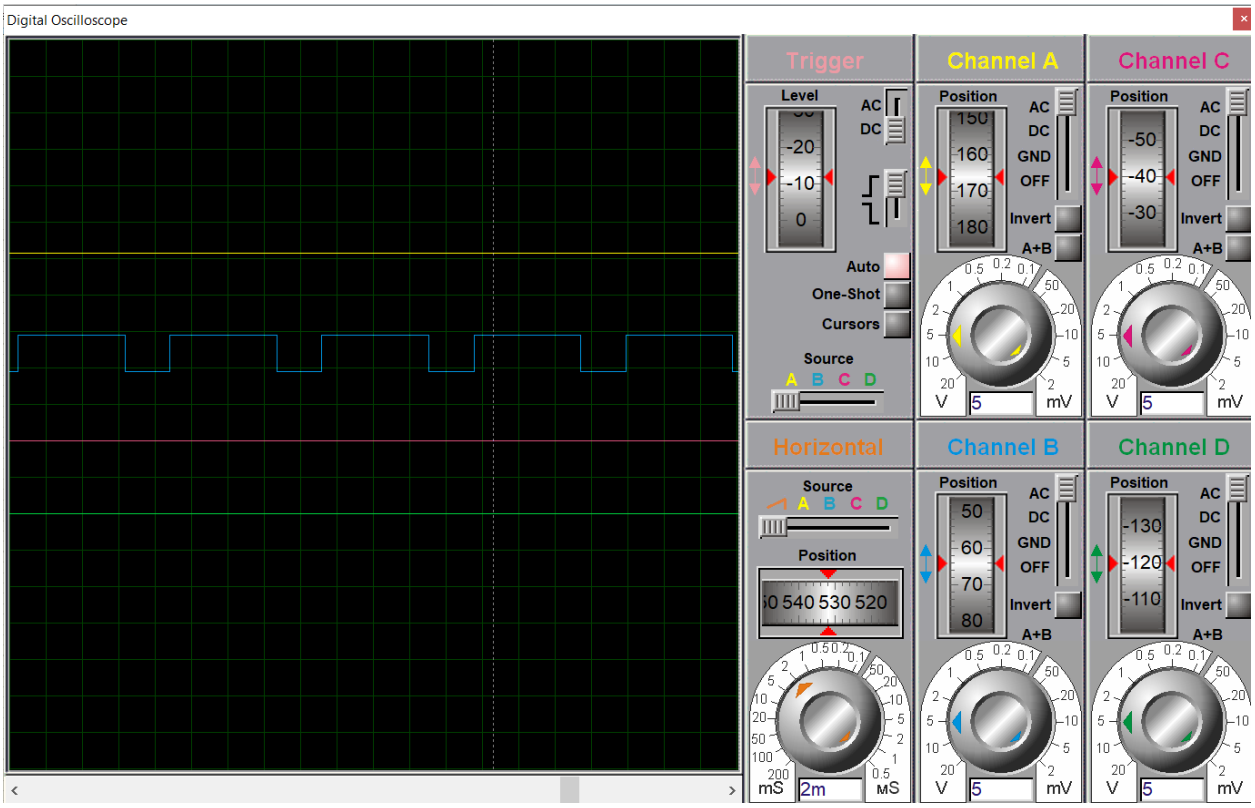


It can be that the duty cycle of the PWM pulse is almost 50%, since the temperature here is also nearly equal to half of the maximum value that LM35 can measure.

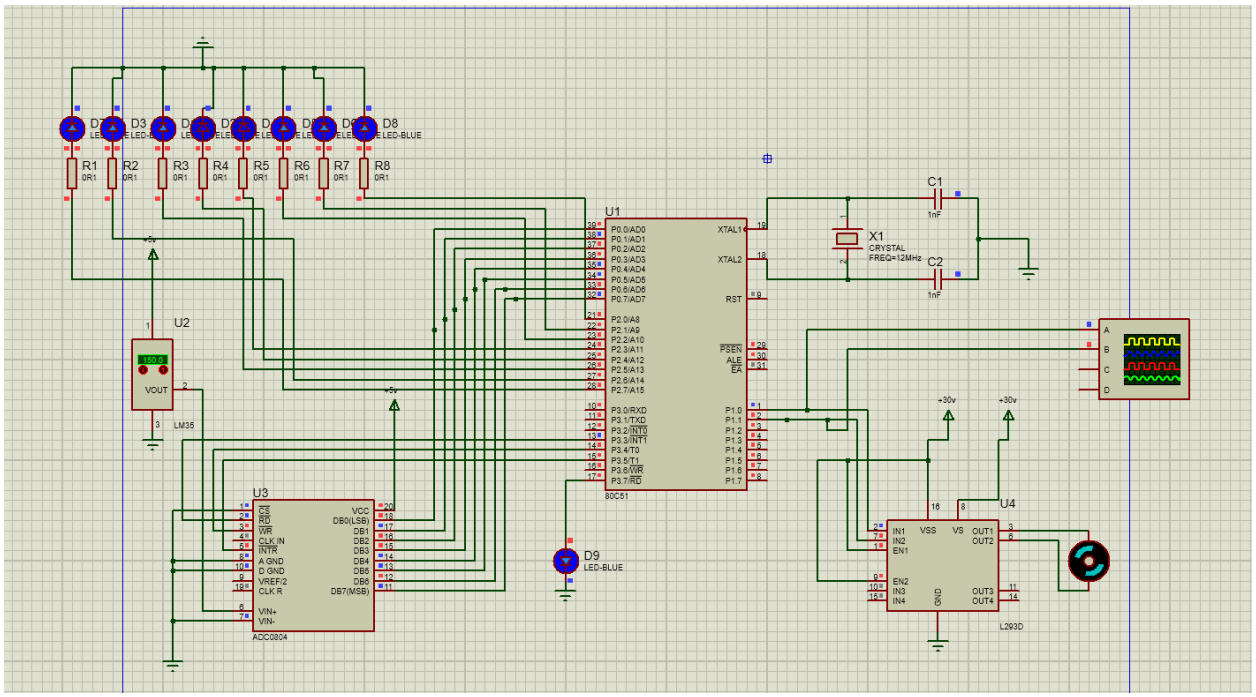


At this point, the temperature of the sensor is set to 112°C which is like 75% of max temperature.

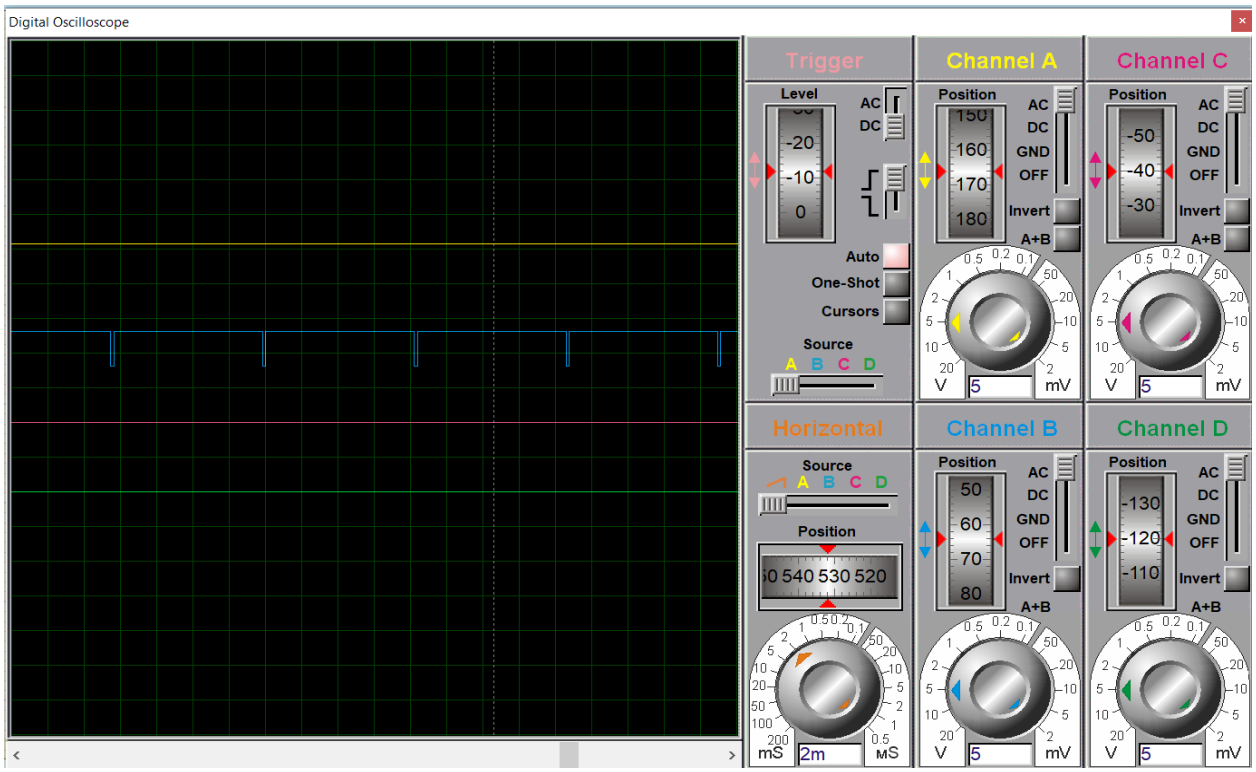
And similarly, the LED's are showing a value of 192 which is 75% of 255.



Here, it can be seen that the Duty Cycle of the PWM pulse is almost like 75% which is in par with 112°C which is 75% of the max temperature of LM35.



Here, The temperature has been set to 150C which is maximum and similarly the LED's can be seen glowing fully, giving a value of 255 which is maximum of PWM Pulse.



Here, it can be seen that the Duty Cycle of the PWM pulse is almost 100% which is in part with the temperature which is 100% of the maximum.

Conclusion:

The basic idea of this project is to automate the fan speed according to the temperature without any need for us to change the speed of the fan every time the temperature changes, we used intel 8051 microcontroller to control the fan speed, a temperature sensor named LM35 gives analog voltage directly proportional to the temperature, and this analog voltage signal is then converted into a digital signal using ADC0804 and this digital signal is used to make a PWM signal which is fed to the DC motor, The duty cycle is directly proportional to the temperature, hence the Dc motor speed will be proportional to the temperature.In this way, our main objective of the project is achieved.

Attachments:

The proteus file of the project for simulation is uploaded in drive in this link...

References:

The images & data related to 8051 are taken from -
https://www.watelectronics.com/8051-microcontroller-architecture/

The information about the temperature sensor, ADC & Motor Driver have been taken from the official data sheets -

https://www.ti.com/lit/ds/symlink/lm35.pdf

https://www.ti.com/lit/ds/symlink/adc0804-n.pdf

https://www.ti.com/lit/ds/symlink/l293.pdf