**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 controlling of fan. 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.

**COMPONENTS**

**1.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).

**2.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, 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.

**3.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 and ARM.

**4.8051:**

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.

**CIRCUIT DESIGN**

**Temperature Detection & ADC Conversion:**

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.

**8051 MC and Digital Value Indicator:**

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 LEDs 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 resistors used act as pull up resistors to provide LEDs 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.

**Motor and Motor Driver:**

The above picture contains DCM which is connected to the 8051MC via L293D Motor Driver to provide the DCM with sufficient current. 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 ⇒ 𝑣𝑜𝑙𝑡𝑎𝑔𝑒 = 0;

𝑓𝑜𝑟 𝑇 > 0°𝐶 ⇒ 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:

1. Set the read bar pin of ADC0804 to disable data lines.
2. Connect the CS (chip select) pin to ground which will make ADC0804 on.
3. Set the EOC (INTR bar here) pin because the ADC will make this pin LOW after the conversion of analog signal into digital.
4. To start conversion, we need to give a low to high pulse to write bar, so clear it first and then Set it.
5. Check for the EOC pin to become HIGH to LOW which denotes that the conversion is complete.
6. 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 a logic 1 should be driven from the microcontroller pin in order to glow the LED. This can be used for visualizing 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.

**CODE**

; org 0000h

; jmp Start

org 0000h

SETB P1.0

CPL P1.0

Start:

SETB P3.3;

CLR P3.4;

SETB P3.5;

SETB P3.4;

stay: JB P3.5, stay

CLR P3.3

SETB P3.7;

;cpl p3.7;

MOV TMOD,#00H ; Timer0 in Mode 0

MOV R1, P0;

MOV A, R1;

MOV B, #3;

MUL AB;

MOV P2,A;

CPL A;

MOV R7, A

SETB TR0 ;

CLR P1.1;

LCALL HIGH\_DONE;

SETB TR0

LCALL LOW\_DONE;

JMP Start

HIGH\_DONE:

SETB P1.1

CLR TF0

MOV TH0, R7

stay1: JNB TF0, stay1;

CLR TF0

RET

LOW\_DONE:

CLR P1.1

MOV A, #0FFH

CLR C

SUBB A, R7

MOV TH0, A

stay2: JNB TF0, stay2;

CLR TF0

RET

Loop:

jmp Loop

END

**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.

**Circuit Diagram**

