A
PROJECT REPORT ON

AUTOMATIC SPEED CONTROL OF MOTOR DEPENDING ON TEMPERATURE

Submitted By:
LOHITHA BOMMISETTY

Introduction : In today's world, where automation and efficiency are paramount, the ability to control devices based on environmental conditions has become increasingly important. This project leverages the capabilities of microcontrollers, temperature sensors, and motor drivers to create a system that dynamically adjusts the speed of a motor in response to changes in temperature.

Purpose : The primary purpose of this project is to design and implement a system that automatically regulates the speed of a DC motor (or fan) based on real-time temperature readings. This system will utilize an Arduino microcontroller to read data from a temperature sensor, process that data, and then adjust the motor's speed accordingly through a motor driver.

Objective : The primary objective of the project is to develop a system that automatically adjusts the speed of a motor (or fan) in response to changes in ambient temperature. To create a responsive system that can adapt to environmental changes, ensuring optimal performance of the motor while maintaining energy efficiency and enhancing user comfort.

Components:

The Automatic Speed Control of Motor Based on Temperature project involves several key components, each serving a specific function in the overall system. Below is a detailed explanation of each component:

1. **Arduino Microcontroller**
   - **Overview**: The Arduino is an open-source electronics platform based on easy-to-use hardware and software. It consists of a microcontroller (such as the ATmega328 on the Arduino UNO) that can be programmed to perform various tasks.
   - **Function in the Project**:
     - **Data Processing**: It reads the temperature data from the temperature sensor and processes it to determine the appropriate motor speed.
     - **Control Logic**: It executes the control logic to adjust the motor speed based on the temperature readings.

    - **PWM Generation**: It generates a Pulse Width Modulation (PWM) signal to control the speed of the motor.

2. Temperature Sensor

- **Overview**: A temperature sensor detects and measures temperature. Common sensors used in this project include:
    - **TMP36**: An analog temperature sensor that outputs a voltage proportional to the temperature in Celsius.
    - **DHT11**: A digital sensor that provides both temperature and humidity readings.
- **Function in the Project**:
    - **Temperature Measurement**: It provides real-time temperature readings, which are crucial for determining the necessary adjustments to the motor speed.
    - **Signal Output**: The TMP36 outputs an analog voltage, while the DHT11 provides digital data, both of which can be read by the Arduino.

3. Motor Driver (e.g., L293D)

- **Overview**: A motor driver is an electronic circuit that allows the control of a motor’s speed and direction. The L293D is a popular dual H-bridge motor driver that can control two DC motors or one stepper motor.
- **Function in the Project**:
    - **Motor Control**: It receives control signals from the Arduino and regulates the power delivered to the motor.
    - **Direction Control**: It allows the motor to be driven in both forward and reverse directions, depending on the control signals.
    - **Speed Control**: It uses PWM signals from the Arduino to adjust the speed of the motor based on the temperature readings.

4. DC Motor or Fan

- **Overview**: A DC motor converts electrical energy into mechanical energy, allowing it to perform work. In this project, a small DC motor or fan is used to provide cooling or heating.

- **Function in the Project**:
    - **Actuation**: The motor or fan provides the necessary airflow to regulate temperature in the environment.
    - **Variable Speed**: The speed of the motor is adjusted automatically based on temperature, allowing for efficient temperature management.

5. Power Supply

- **Overview**: A power supply provides the necessary voltage and current to operate the Arduino, motor driver, and motor. The requirements depend on the specifications of the components being used.
- **Function in the Project**:
    - **Powering Components**: It ensures that all components receive adequate power for operation.
    - **Voltage Regulation**: Depending on the components, different voltage levels may be required (e.g., 5V for Arduino and 9V for the motor).

6. Breadboard and Jumper Wires

- **Overview**: A breadboard is a reusable platform for prototyping electronic circuits, while jumper wires are used to make connections between components.
- **Function in the Project**:
    - **Circuit Prototyping**: The breadboard allows for easy assembly and modification of the circuit without soldering.
    - **Connections**: Jumper wires connect the various components, enabling communication and power distribution among them.

Steps to Design the Circuit (TinkerCad)

Place Components:

Drag and drop the following components onto the workspace:

- **Arduino UNO**
- **Temperature Sensor (TMP36 or DHT11)**

- **Motor Driver (L293D)**
- **DC Motor or Fan**
- **Power Supply (battery or DC power source)**
- **Breadboard (optional)**

Wiring the Components:

Temperature Sensor (TMP36):

- Connect the left pin (VCC) to the 5V pin on the Arduino.
- Connect the middle pin (Vout) to an analog pin on the Arduino (e.g., A0).
- Connect the right pin (GND) to the GND pin on the Arduino.

Motor Driver (L293D):

- Connect the IN1 and IN2 pins to two digital pins on the Arduino (e.g., pin 9 and pin 10).
- Connect the ENA pin to a PWM-capable digital pin on the Arduino (e.g., pin 3).
- Connect the motor terminals to the output pins (OUT1 and OUT2) of the L293D.
- Connect the power supply to the L293D (VCC and GND).
- Connect the GND of the L293D to the GND of the Arduino.

DC Motor:

- Connect the motor terminals to OUT1 of the L293D.

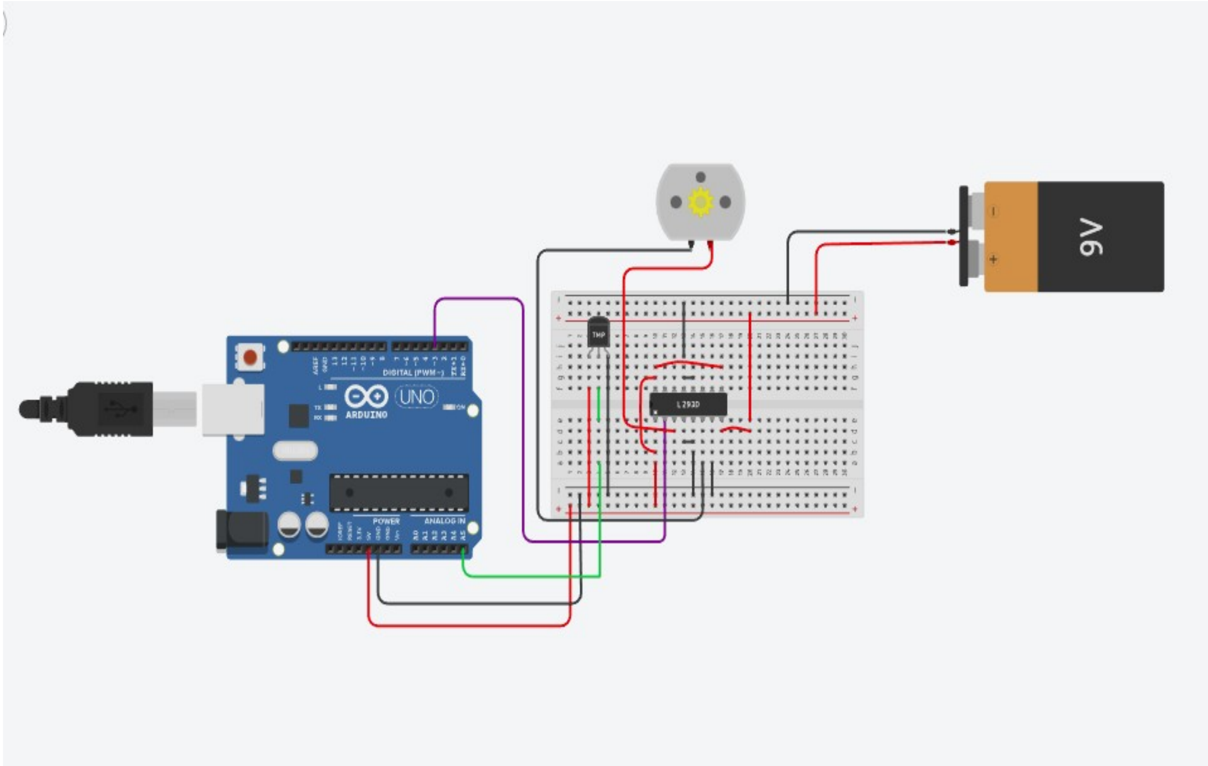
Power Supply:

- Connect the power supply to the motor driver, ensuring it matches the voltage requirements of the motor.

Final Connections:

- Ensure all grounds (GND) are connected together (Arduino, motor driver, and power supply).
- Double-check connections to avoid short circuits.

SCHEMATIC:



Program :

```
int motorpin = 3;

int tmpPin = A5;

void setup()
{
  Serial.begin(9600);
  pinMode(motorpin,OUTPUT);
}
void loop()
{
  int reading = analogRead(tmpPin);
  float volt = reading *(5.0/1024.0);
  float temp = (volt - 0.5)*100;
  Serial.print(temp);
  delay(1000);
```

```
  if(temp <28)
  {
    analogWrite(motorpin,0);
  }
  else if(temp <30)
  {
    analogWrite(motorpin,138);
  }
  else if(temp <35)
  {
    analogWrite(motorpin,150);
  }
  else if(temp <38)
  {
    analogWrite(motorpin,190);
  }
  else
  {
    analogWrite(motorpin,225);
  }
}
```

Applications:

1. HVAC Systems

In heating, ventilation, and air conditioning (HVAC) systems, this technology can be used to adjust the speed of fans and compressors based on temperature readings.

2. Computer Cooling Systems

In computers and servers, fans are used to dissipate heat. An automatic speed control system can adjust fan speeds based on the internal temperature of the components.

3. Home Applications

In refrigerators ,washing machines,dryers,dishwashers, freezers, motors control compressors and fans. Automatic speed control can optimize performance based on the internal temperature.

4. Greenhouses and Agriculture

In agricultural settings, controlling the speed of ventilation fans based on temperature can help maintain optimal growing conditions in greenhouses.

5. Aquarium Systems

In aquarium setups, water pumps and filtration systems can be controlled based on water temperature to maintain a healthy environment for aquatic life.

Conclusion:

Overall, the automatic speed control of motors based on temperature is a practical and effective solution for optimizing performance in various systems.