INTRODUCTION :-

In today's fast-paced world, convenience is key. Dealing with the T**emperature Fluctuations** in our living space can be quite a hassle. For those who live in regions where the afternoons can be sweltering hot and the nights chillingly cold, constantly adjusting your wall fan can be annoying. However, with a little bit of DIY ingenuity and the help of some electronic components, we can transform our ordinary electric fan into an automatic one. Temperature Controlled Fan using an Arduino Uno, LM35 temperature sensor, and a relay.

Objectives

1. Aim is to automatically turn on or off an AC fan depending on the room temperature.

1. Utilizing an Arduino Uno to process the temperature data and control the fan through a relay module.

1. Implementing a relay so that the switch safely turn on & off the AC fan, preventing direct exposure to high voltage.

1. Offers a self made method to maintain comfortable room temperature without manual fan operation.

Sr No.	Title Of Research Paper	Authors	Name of Journal/ Conference	Date of Publication	Brief Summary	Remarks
1)	Design and implementation of automatic room **Temperature Controlled fan using Arduino** Uno and LM35 **heat** sensor	Nur Afiqah Junizan, Amirrudin Abdul Razak, Bohendiran Balakrishnan, W.A.F.W. Othman	International Journal of Engineering Creativity and Innovation (IJECI)	2019	This paper presents design and implementation of an automatic fan system using an Arduino Uno and LM35 temperature sensor, which adjusts fan speed based on room temperature.	PWM (Pulse Width Modulation), Varying with width of digital pulses.

| 2) | Temperature Controlled Fan using IOT | G. Joga Rao , G. Satish , D. Abhinav , P. Mothi Manoj Nageswara Rao , P. Satish Ganesh | International Journal of Scientific Research in Science, Engineering and Technology | Mar-Apr 2018 | This paper presents an Arduino-based temperaturecontrolled fan system using the LM35 sensor and ESP8266 Wi-Fi module. Which automatically adjusts fan speed based on room temperature and allows remote control via mobile phone using IoT technology | IoT and PWM techniques |

3)	Arduino & IOT based Temperature based Speed Control of Fan	Mrs. G Jamuna, Mamidi Sruthilaya, Assistant Professor	International Journal of Engineering, Science and Advanced Technology	Oct 2024	This paper presents an IoTenabled fan system using Arduino Uno, LM35 temperature sensor, and ESP8266 Wi-Fi module to automatically adjust fan speed based on ambient temperature. It features both automated and manual control, with real-time display via an LCD and remote operation through mobile devices.	PWM system & IOT

4)	Centigrade and Body Temperature Based Fan Speed and AC Load Controller	Keertheghaa S., , Ismath Sulthana H., , Arthi A., 3 & Aruna Devi T.	Asian Journal of Applied Science and Technology (AJAST)	Jan-Mar 2025	This paper presents a smart Arduino Uno-based system that automatically adjusts fan speed and AC load based on room and body temperature using a DS18B20 sensor. The system uses PWM and relay control for efficient thermal management and displays real-time data on an LCD, aiming to improve energy efficiency and user comfort.	AC Load management & PWM System

Hardware Tools & Components:-

1. Usb cable
2. LM35 temperature sensor
3. 5V single channel Relay
4. Jumper wires
5. Breadboard
6. An AC Fan

Software Tools:-

1. Arduino IDE
2. Embedded C/C++ Programming language used in the Arduino sketch.

METHODOLOGY :-

1. Designing of the System:
- The design of the system is all about reading temperature using the LM35 sensor and using the data to control the speed of a AC fan ●

   The fan speed is adjusted using PWM signals from the Arduino based on the temperature.

2. Algorithm:
- Reading analog voltage from LM35

- Converting voltage to temperature using the formula:
   Temperature (°C) = (Analog Voltage in mV) / 10

- Checking the temperature range and set fan speed accordingly:
   Below threshold (e.g., <25°C): Fan OFF
   Medium range (25°C–35°C): Medium speed
   Higher range (>35°C): High speed or full speed

- Generating PWM output from Arduino to control the transistor and, in turn, the fan speed.

3. Coding (Arduino Programming):
- Embedded C/C++ using Arduino IDE.

- Using analogRead() for sensor input and analogWrite() to generate PWM signals.

- Including conditional logic to control fan speed based on temperature ranges.