IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 4, Aug-Sept, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

An Automatic Fan Speed Controller Leveraging Internet Of Things

Gokul Ashok Nanmaran[1], Neha Shivhare[2], Aparna A[3], Kayalvizhi Jayavel[4]

[1,2,3,4] **School of Computing, SRM University, Chennai, Tamil Nadu 600047, India**

Abstract
The normal fans in household as well as offices have a manual speed control system. It is regulated using a rotary switch which internally involves a potentiometer. The potentiometer adjusts current flow to the fan and hence, the speed is adjusted. This project involves the use of temperature sensor. The temperature sensor will read through changes in room temperature. This data will be processed by a microcontroller (in our case, Arduino), to regulate fan speed automatically. The fan regulation will be done with range of temperatures specified for a specific fan speed. Temperature extreme values are fixed, based on the requirements, to which fan speed is mapped. Thus, the speed of the fan can be actuated. This system can further be extended to air conditioning systems and larger workspaces.

Keywords: IoT, Internet of Things, Fan Speed Controller, Arduino, Temperature, Potentiometer.

1. Introduction

Information technology has become an integral part of our daily life. According to Information Technology Association of America, information technology is defined as "the study, design, development, application, implementation, support or management of computer-based information systems." Information technology has served as a big change agent in different aspect of business and society. It has proven game changer in resolving economic and social issues.The growing ubiquity of mobile technology and internet-connected devices will present new threats and increase the challenges for IT security managers. Application complexity is growing and web services are the primary means of integrating into the cloud.

2. Literature Survey

The normal fans in household as well as offices have a manual speed control system. It is regulated using a rotary switch which internally involves a Potentiometer. The Potentiometer adjusts current flow to the fan and hence, the speed is adjusted. This system creates a few problems in terms of user comfort. For example, the speed of fan needs to be changed at regular time intervals as the temperature and weather conditions keep changing throughout the day. Also, in the mornings or at night, it might become cold.

Hence, either the fan speed needs to be adjusted before sleeping or the consumer has to wake at the inappropriate time and manually set it. It can be difficult for people sensitive to changing temperatures and may cause sickness, especially small kids and babies. The organization of the paper is as follows: Section 3 deals with proposed system. Section 4 deals with problem definition. Section 5 deals with methodology and section 6 presents conclusion and future works.

3. Proposed System

The proposed system aims to overcome the drawbacks of the existing system. The new system involves the use of temperature sensor. The temperature sensor will read through changes in room temperature. This data will be processed by a micro-controller (in our case, Arduino), to regulate fan speed automatically.

The fan regulation will be done with range of temperatures specified for a specific fan speed. Temperature extreme values are fixed based on the requirements to which fan speed is mapped. Thus, the speed of the fan can be actuated. This system can further be extended to air conditioning systems and larger work-spaces. [1]

Our Automatic Fan Speed Controller is aimed at automating the speed of the fan according to the room temperature. A temperature sensor is used, which provides the temperature values of the surroundings. These values are then used by the micro-controller, which then actuates the fan speed. There will be specific fan speed value set for a certain range of temperature.[2] Temperature extreme values are fixed based on the requirements to which fan speed is mapped.

4. Problem Definition

The normal fans in household create a few problems in terms of user comfort. For example, the speed of fan needs to be changed at regular time intervals as the temperature and weather conditions keep changing throughout the day. Also, in the mornings or at night, it might become cold. Hence, either the fan speed needs to be adjusted before sleeping or the consumer has to wake at the inappropriate time and manually set it. It can be difficult for people sensitive to changing temperatures and may cause sickness, especially small kids and babies.[3]

The Temperature Detection Module is responsible for reading temperature from the sensor and send required data to Fan Speed Control module. The temperature sensor (LM35) returns analog input to the Arduino board when called. This module comprises of two parts:

- Temperature parser and
- Passing data to fan controller.

The data received from LM35 needs to be converted to recognized temperature reading formats like Celsius, Fahrenheit etc. before the data starts making sense. For conversion of temperature to Celsius, a factor of 0.48828125 needs to multiplied to the received value. The received temperature must be within the specified minTemp – maxTemp range. If the temperature is outside the range, the fan will be stopped. However, if the temperature is within the allowed range, then instructions with temperature and fan speed as parameters are sent to the fan controller. The temperature reading is updated in real time on the LCD screen for user reference.

The Fan control module, controls the speed of the fan depending on the temperature values recorded by LM35 temperature sensor above.

5. Methodology

We followed the waterfall model for development.

5.1 Requirement Elicitation

Functional Requirements - Functional Requirements defines functions of a software system or its components. They may be calculations, technical details and other technical functionality that define what a system is supposed to accomplish.[4] The system’s main functionality is to regulate the speed of fan. Based on the temperature values read, it will actuate the fan regulator. Hence controlling its speed is the main functional requirement.

Hardware Requirements - The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the whole system. The following were the hardware required:

1. Micro-controller (Arduino)
2. DC fan
3. 12V dc battery
4. LM-35 Temperature Sensor
5. Bread Board
6. Jumper Wires
7. Transistors & Diodes
8. Resistors & Capacitors
9. LED & Buzzer

Software Requirements - The software that was required for the project was the Arduino IDE. Knowledge of Arduino programming (Embedded-C) was also required.

Non-Functional Requirements - Efficiency - Determines how efficiently the controller works, it depends on the processing capacity of it and is different for different micro controllers. Other Non-functional requirements were Reliability, Safety Measures and Standards.

5.2 Modules

The research project has two modules:

- Temperature-Detection-Module
- Fan Speed Control Module

Temperature Detection Module - We use LM35 temperature sensor to detect the temperature as shown in figure 1. LM35 is a precision IC temperature sensor with its output proportional to the temperature (in °C). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes. [5]

With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self-heating and does not cause more than 0.1° C temperature rise in still air. The operating temperature range is from -55°C to 150°C. The output voltage varies by 10mV in response to every oC rise/fall in ambient temperature, i.e., its scale factor is 0.01V/ °C.

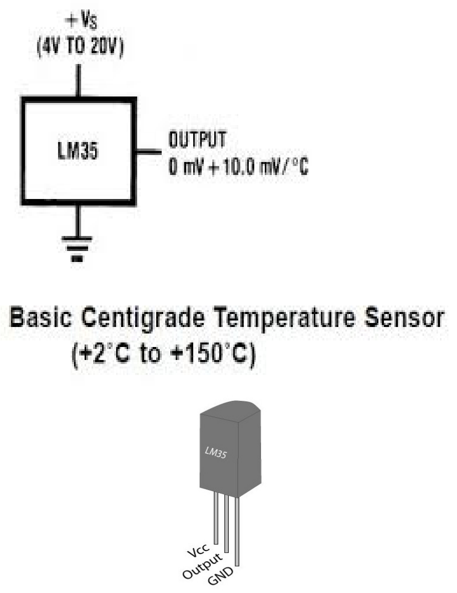

Figure 1: LM35 Sensor Components and its outer view

The Temperature Detection Module is responsible for reading temperature from the sensor and send required data to Fan Speed Control module. The temperature sensor (LM35) returns analog input to the Arduino board when called. This module comprises of two parts:

1. Temperature parser
2. Passing data to fan controller

1. Temperature parser – The data received from LM35 needs to be converted to recognized temperature reading formats like Celsius, Fahrenheit etc. before the data starts making sense. For conversion of temperature to Celsius, a factor of 0.48828125 needs to multiplied to the received value.

- Passing data to fan controller – The received temperature must be within the specified minTemp – maxTemp range. If the temperature is outside the range, the fan will be stopped. However, if the temperature is within the allowed range, then instructions with temperature and fan speed as parameters are sent to the fan controller. The temperature reading is updated in real time on the LCD screen for user reference.

- Fan Speed Control Module - This module, controls the speed of the fan depending on the temperature values recorded by LM35 temperature sensor above. A dc fan image is given in figure 2.

Figure 2. DC Fan

This fan can be used for cooling the heated systems as in laptops, mobile devices, etc. This module takes input from

the sensor, processes it and actuates the fan’s motor. For processing, it uses a map function.
map (value, from Low, from High, to Low, to High)
fan Speed = map (temp, tempMin, tempMax, 32, 255);

5.3 System Design

Activity Diagram - Activity diagrams model the internal logic of a complex operation.

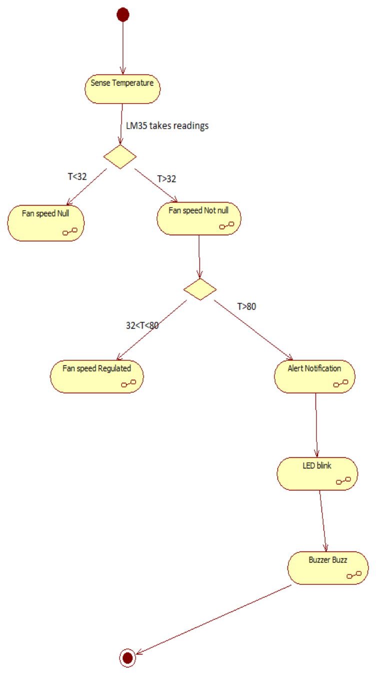

Figure 3. Use Case Diagram for fan speed controller

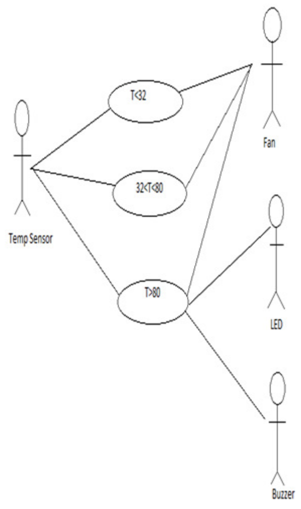

Figure 4: Sequence Diagram for fan speed controller

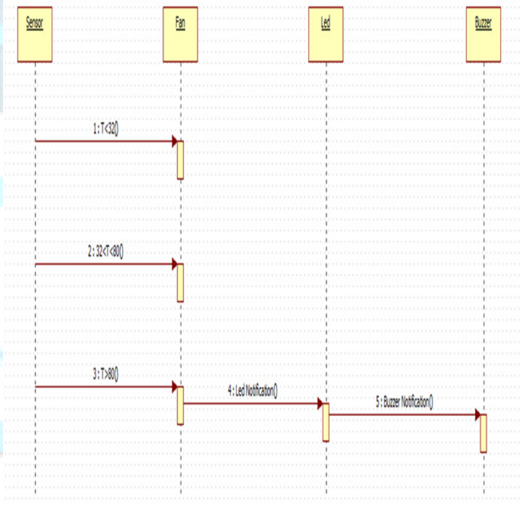

Figure 5 Class Diagram 1 for fan speed controller

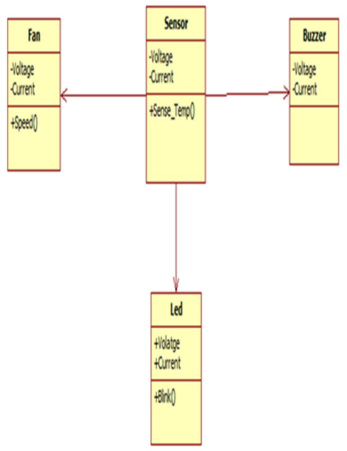

Figure 6 Class Diagram 2 for fan speed controller

System Architecture - The system architect establishes the basic structure of the system. This diagram explains each and every stage of request processing from client to destination. Defining the essential core design features and elements that provide the framework. The systems architect provides the architects view of the users' vision.

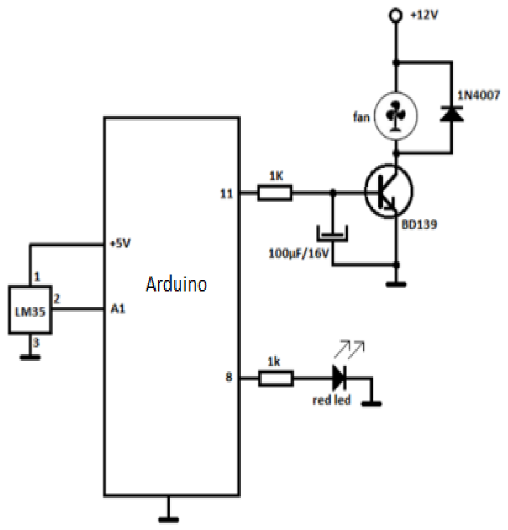

Figure 7 Circuit configuration for fan speed controller

E-R Diagram - Entity – Relationship model is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

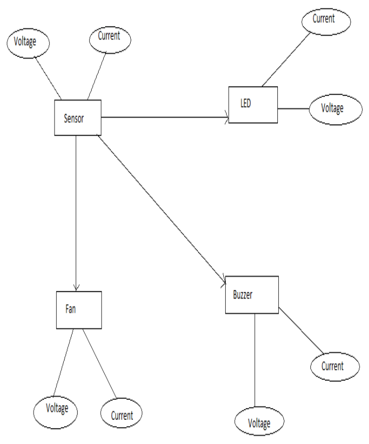

Figure 8 E-R diagram for fan speed controller

Figure 3, 4, 5,6,7,8 indicate the different aspects of the diagrams for fan speed controller. In an embedded system development process, under Implementation, following keys points are followed:

1. Developing the circuit schematics
2. Obtaining the PCB's
3. Procuring the components
4. Assembling the components
5. Testing each component individually and then integrate all components for integrated testing

While implementing, code optimization and Coding guidelines have to be taken into full consideration.

5.4 Implementation and Screenshots

The circuit diagram was built and the temperature in different rooms were observed. According to threshold settings, the indication to the fan was sent and fan speed was controlled automatically.

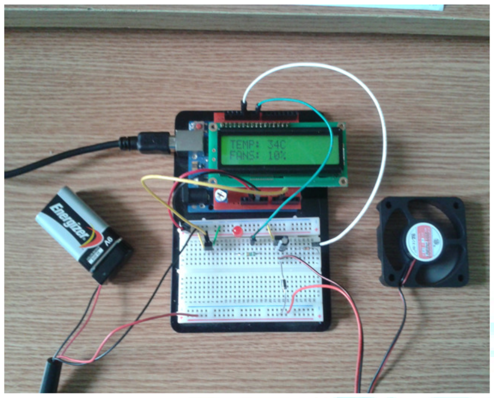

Figure 9. Experimental Set up

The above screen shot demonstrates our system. It has a temperature sensor, a DC fan, a LCD Display, a LED & a Battery.

5.5 Testing

In order to test the system we considered few test cases and executed them to check the proper functionality of the system. The following are some of our test cases:

- Provide least temperature value, here the temperature limits are from 23-30 degree celcius. Hence, our first test-case will be for temperature value below 23 that is 20 degree celcius.

For this use-case, required output: Speed of Fan is 0.
Explanation: Since, the temperature values are mapped from 23-30 to Fan speed values 0-100.

2. Here, we consider temperature value between 23 to 30. Hence expected output is that the fan's speed will be somewhere between 0 to 100, excluding 0 and 100.

3. Consider the third test-case, here temperature is raised above the highest temperature value i.e. 30 degree celcius. Expected output is that the fan should have 100% speed, or the maximum speed.

Table 1 shows the fan speed according to the temperature values

Table 1 Fan speed vs Temperature

Se no	Temperature value	Fan speed
1	20° C	0
2	22° C	20
3	24° C	40
4	26° C	60
5	28° C	80
6	>30° C	100

6. Conclusion and Scope for Future Work

In this paper, we designed a system that automatically controls the speed of the fan using the temperature values sensed by the sensor. Project involving the use of temperature sensor has been successfully accomplished. The temperature sensor reads through room temperature. This data is processed by a microcontroller to regulate fan speed automatically. The fan regulation is hence done with range of temperatures specified for a specific fan speed.

This system can further be extended to air conditioning systems, Laptop cooling systems and larger work-spaces by adding additional sensors on request like humidity, rain, light, etc. Also mobile based monitoring can be added as an additional feature. The same fan speed control can be done through Internet, if the device information reaches the cloud, as a future scope of this research.

References

[1]http://www.electroschematics.com/9540/arduino-fan-speed-controlled-temperature/
[2].http://arduino.cc/en/reference/map
[3].Dr. K.V.K.K. Prasad,"Embedded/Real-time Systems: Concepts, Design, Programming, Black Book" November 2003.
[4] http://esd.cs.ucr.edu/
[5] http://www.embedded.com/