***Project Review Paper***

***On***

**Automatic Room Temperature Controller**

**Submitted for the requirement of**

**Project course**

BACHELOR OF ENGINEERING

**COMPUTER SCIENCE & ENGINEERING**

****

**Submitted to: Submitted By:**

**Project Teacher (Supervisor)**

Shuvendu Das (12496) **NAME UID**

Sahil Kaundal 21BCS8197

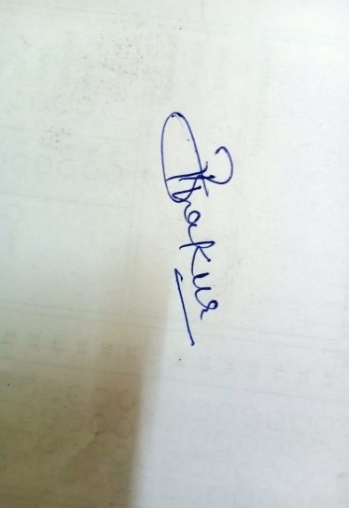
Rohit Kumar Goswami21BCS10083

Pratik Kumar Dey21BCS8183

Feroj Khan 21BCS8168

**Co Supervisor**

Bhanu Thakur (12251)

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CHANDIGARH UNIVERSITY, GHARUAN**

**June 2022**

**INTRODUCTION**

The concept of this paper is to create an automatic room temperature control system to control the temperature of a room. This circuit maintains the temperature of the room in a particular range. Automatic control plays an ever-increasing role in human way of life. During winter and summer season, room temperature must be maintained in order to ensure the human convenience. This project mainly includes the temperature control of surrounding in winter and summer. It consists of Temperature sensing unit, LCD module, a fan and heater. It will operate based on the value or range of Temperature in the system which is detected by Temperature of the sensor. The Temperature sensor detect the temperature of the room. The Temperature sensor is connected to the Arduino. It converts the analog input to digital value. It is use to switch on the heater and fan. The LCD module is also connected to the Arduino. The module displays the current temperature. The LCD display used is a 16x12 Alphanumeric display.

**1.1 Project Background**

Nowadays, the air conditioning is widely used especially in warm countries including India. Usually, the conventional air conditioning is always cooling the room depending on the fixed temperature setting and is not automatically adjusted for the comfort of the users. In the central air conditioning control field, excellent real-time, high reliability, and good intelligence are proposed by many researchers. The traditional PID algorithm is, in fact, still playing a main role in the control process. The air conditioning system has become a field to be researched to improve the user convenience by applying intelligent system.

While the enhanced air conditioning system is being designed, the consideration of the type of control system must be included in a modeling design. In particular the controller must be able to avoid the inefficiency of having the air conditioning operate all the time. Several control options were considered at presence sensing circuit, which would turn the air conditioning off when people are not in the room with the air conditioning and a temperature sensor input, which would change the air conditioning operation depending on room temperature. Based on the observation of the using the present conventional air conditioning application, it always working all the time without a systematic control. Therefore, the control of the air conditioning is adjusted through a feedback control system to monitor and maintain a constant temperature based on the data input from the sensor.

**1.2 Problem Statement**

The problem happens when the air conditioning is still functioning although in the event of cold weather. The function is uncontrolled and must be manually turned on and off. Sometimes it can lead to high usage of electricity which in turn raises the electricity bill when the user forgot to switch it off. The system also does not have the capacity to adjust the room temperature regardless of the ambient temperature. To address the problem, the automatic room temperature control that can control the temperature automatically is proposed. The advantages of such a system are less energy usage, and provides more convenient to the consumers.

**1.3 Objectives**

The objectives of this project are:

1. To record voltage outputs across the DHT 11 sensor, fan and heater when they are in operation.

2. To test its applicability in different temperature ranges as measured by the DHT11 temperature sensor.

3. To analyses the performance of the controller.

**1.4 Scope of The Project**

The project is making use of a fan and a heater with the circuit systems in it. The scope of this project is:

* To design a system that sets the desired temperature value range.
* Automatic temperature sensor monitoring the recent change of temperature within the range.
* Heater/fan functioning within desired temperature range(s).
* Automatically sets the room temperature within 20 to 25 degrees Celsius.

**1.5 Justification and Significance**

Advancement of technology has made automation become part of our lives. A home is usually the most occupied place. Areas in the home that are usually occupied by people, such as the living room and bedrooms need to be maintained within an acceptable temperature range. The problem becomes more acute in areas of the home that are occupied by infants. Adults could possibly find their way around thermal discomforts, but infants may not. Other rooms in home that are used as storage for perishable food items also need to be thermally regulated in order to prevent accelerated decay of such items. Hence this project is a need of the hour.

**LITERATURE REVIEW**

**2.1 Related work**

In the past few years, the need for automation has increased and has been widely applied to cooling and heating systems. There are plenty of commercial temperature control systems which can be bought from manufacturers or inventors, and also, quite a lot of work has been published in this area.

In a related invention (Hedges,1947), R.E. Hedges invented an automatic temperature control systems intended at automatically controlling the temperature of an object or a region so as to continuously maintain the said temperature at a constant value. In his design, the temperature control system can only control a single heat source. It is difficult therefore to attain temperature control of systems emitting simultaneously and in combination of the desirable features of large capacity, high response, and accuracy.

In 1992 (Ian,1992), Ian Bell invented a self-programmable heating and cooling system, which is based on temperature control system. This system is not easy to operate and can only be operated by experts as it has to be interfaced with a computer anytime the re-programming is needed.

M. R. Levine (Levine, 2013) invented an automatic temperature adjusting system for an air conditioner room. The automatic temperature adjusting system for the air conditioner room was made simple in operation and was capable of monitoring the temperature of the human body at any time in the air-conditioned room and transmits the corresponding signals to the air conditioner in time. The air conditioner conducts adjustment, so that the health of people is guaranteed. However, this is also a one-time programming and it is needed to be interfaced with the computer anytime reprogramming is needed, hence, the operation of the system becomes complex.

Other automatic temperature controller systems that have been designed and published include;

* Automatic temperature control for transport airplanes (Hedges, 1947).
* Wireless thermostat and room environment control system. (Tate and Ries,1990)
* Automatic temperature control system. (Minoru, 1996),
* Automatic heater controller. (Poll,2006).
* Design of automatic temperature control system on laser diode of erbium-doped fiber source, in: Proceedings of (Chengxiang, Zhenhua, Xu and Feng, 2011).
* Design of automatic temperature-control circuit module in tunnel microwave heating system. (Fu, Wang and Yang, 2011).

These designs are not easy to use in terms of programming and temperature adjustment. The systems work on the benefits of using temperature adjustable and fan temperature control 12 systems. These systems are either one time programmable or need analog adjustment which is not accurate and more difficult to use.

More recent, real time-based temperature control using Arduino was published (Amoo et al,2014). The system uses Arduino based on ATMEL 89C51, which is just one of the applications of Arduino. The system is not simple in terms of operation as it is tedious to change the reference temperature. The system is similar to the ones presented by the authors in the following systems:

* A precision temperature controller using embedded system. (Pimpalgaonkar et al, 2011).
* Design and experiment about temperature control system of sealing machine based on Fuzzy PID. (Cao et al, 2011).

Other works based on temperature control do exist in different areas and different applications. Such work such as electric cable interference temperature monitoring in power transmissions, server room temperature measurement using Bluetooth embedded system (Loup et al, 2011), control system for communication room using wireless temperature monitoring system (Bing and Wenyao, 2011) and temperature sensor and Zigbee based temperature measurement (Pengfei et al, 2011) do exist. These systems have the same problem of cost as well as the need for experts in re-programming.

This leaves a gap for the design of a system that is not only simple, cost effective, efficient, easy to program and one that minimizes components so as to reduce the size of the design. The automated room temperature controller system thus comes in handy to provide solutions to the gaps that can be noted in the previous systems. It is simple, cost effective and can be used to provide real-time air conditioning for both home and commercial institutions.

**2.2 Sensors**

A sensor can be defined as a device that measures a physical quantity such as temperature, light, motion or any other physical phenomenon and converts it into an electrical signal (Mengistie, 2018). Examples of sensors include temperature sensors, humidity sensors, motion detection sensors, oxygen sensors and infrared sensors. The output of sensors is usually current, charge and voltage. Temperature sensors are devices that are temperature sensitive, and respond to changes in temperature.

There are 2 kinds of temperature sensors:

1) Contact sensors

2) Non-Contact sensors.

However, the 3 main types are thermometers, resistance temperature detectors, and thermocouples. All three of these sensors measure a physical property (i.e., volume of a liquid, current through a wire), which changes as a function of temperature. In addition to the 3 main types of temperature sensors, there are numerous other temperature sensors available for use.

**2.2.1 Contact Sensors**

Contact temperature sensors measure the temperature of the object to which the sensor is in contact by assuming or knowing that the two (sensor and the object) are in thermal equilibrium, in other words, there is no heat flow between them (Augustin & Fröhlich, 2019).

Examples include;

* Thermocouples
* Resistance Temperature Detectors (RTDs)
* Full System Thermometers
* Bimetallic Thermometers

**2.2.2 Non-contact sensors**

These temperature sensors measure the thermal radiant power of the Infrared or Optical radiation received from a known or calculated area on its surface or volume within it.

An example is the pyrometer

**2.3 TMP36 Sensor**

The TMP36 temperature sensor is an easy way to measure temperature using an Arduino! The sensor can measure a fairly wide range of temperature (-50°C to 125°C), is fairly precise (0.1°C resolution), and is very low cost, making it a popular choice.

Unlike a thermistor, the TMP36 does not have a temperature sensitive resistor. Instead, this sensor uses the property of diodes; as a diode changes temperature the voltage changes with it at a known rate. The sensor measures the small change and outputs an analog voltage between 0 and 1.75VDC based on it. To get the temperature we just need to measure the output voltage and a little bit of math.

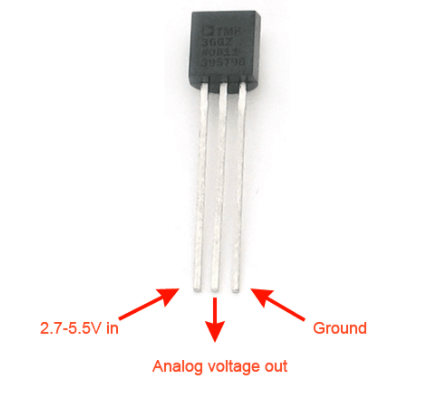
****

Figure 2.1 TMP36 Temperature sensor

Table 2.1. Electrical characteristics of TMP36 sensor

Electrical characteristics

|  |  |
| --- | --- |
| Maximum power supply | 5V DC |
| Minimum power supply | 2.7VDC |
| Minimum current supply | 0.5mA |
| Maximum current supply | 2.5mA |

The TMP36 has 3 pins as shown below;

Pin 1: IN 2.7-5.5V DC

Pin 2: Analog Voltage Out

Pin 3: Ground

**2.4 Microcontroller**

It can be defined as a self-contained system that incorporates a microprocessor for sensing input from the real world and controlling devices based on the input. They consist of a processor, memory and some peripherals. Microcontrollers are used in digital applications for automation and as control units. They work according to a program that is written in their processors. These programs are often written using assembly languages but also high-level languages such as C, C++ PASCAL and Java (Mengistie, 2018).

**2.4.1 Alf and Vegard’s RISC Processor (AVR)**

A Reduced Instruction Set Computer (RISC) is one whose Instruction Set Architecture (ISA) allows it to have fewer cycles per instruction was developed in 1996 by Atmel corporation. They are loaded with an Integrated Development Environment (IDE) and a C compiler. They contain output and input parts, timers, interrupts, USART.12C interface channels and on chip analog comparators. They are fabricated with Modified Harvard Architecture which allows what is contained in the instruction memory to be assessed as if it were data. However, they are difficult for beginners to work with (Barrett and Pack, 2006).

**2.4.2 8051 Microcontroller**

It was developed by Intel Corporation in 1980. It processes 8-bit data at a time and their name results from the fact that they can all be programmed using 8051 assembly language (Mazidi, n.d.). Their main drawback is their language of programming. The assembly language used is complex and difficult to use.

**2.4.3 Programmable Interface Controller (PIC)**

It was originally developed by General Instruments Microelectronics Division. They are affordable and perform better than 8051 microcontrollers. They are better off than 8051 microcontrollers because they have small low pic count devices, however like 8051 microcontrollers they are difficult and cumbersome to program hence cannot be recommended for a beginner.

**2.5 Arduino**

Arduino is a company in Italy that sells Printed Circuit Boards (PCBs) called Arduino boards. Arduino is an open-source electronics design platform which is designed for prototyping and programming with Atmel microcontrollers (Arduino, 2018). They interact with physical environment through sensors. They can be made to respond to physical phenomenon and hence are widely used for automation purposes.

Some of Arduino older boards include;

* Duemilanove
* Duecimila
* NG Rev C
* Arduino NG 240
* Arduino V2
* Arduino Extreme
* Arduino USB V2

**2.5.1. Arduino UNO Rev 3**

The word UNO means one in Italian and was chosen to mark the release of Arduino software Integrated Development Environment (IDE) (Evans, 2011). This is one type of Arduino boards that is widely used due to the following feature

* They have 8-bit bidirectional input/output pins with internal pull up resistors
* 1 KB Electronic Erasable Programmable ROM
* 32 KB flash memory
* 2 KB of RAM

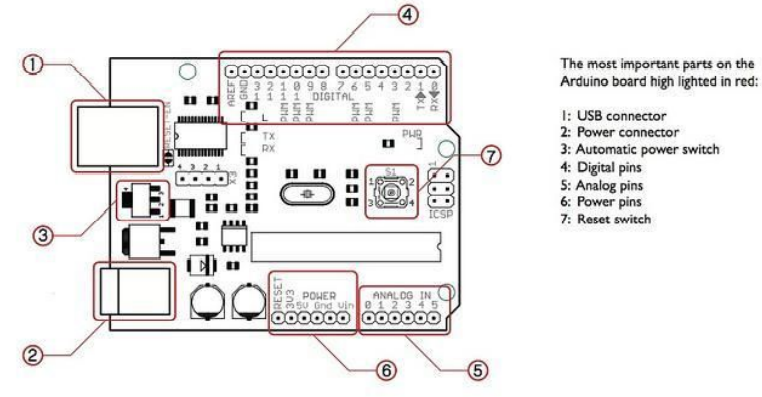


Figure 2.2. Arduino uno rev 3 pins

Arduino UNO is a microcontroller board based on the ATmega328 datasheet. It has 14 digital input/output pins (6 can be used as PWM outputs), 6 analog inputs, a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button.

Examples of inputs: temp sensor, motion sensor, distance sensor, switch etc.

Examples of outputs: -light, screen, motor etc.

**Technical specifications**

* Microcontroller- ATmega328
* Operating Voltage- 5V
* Supply Voltage (recommended)- 7-12V
* Maximum supply voltage (not recommended)- 20V
* Digital I/O Pins 14- (of which 6 provide PWM output)
* Analog Input Pins- 6
* DC Current per I/O Pin- 40 mA
* DC Current for 3.3V Pin- 50 mA
* Flash Memory- 32 KB (ATmega328) of which 0.5 KB used by bootloader
* SRAM 2 KB- (ATmega328)
* EEPROM- 1 KB (ATmega328)
* Clock Speed- 16 MHz
* LED\_builtin- 13
* Length-68.6mm
* Width-53.4mm
* Weight-25g

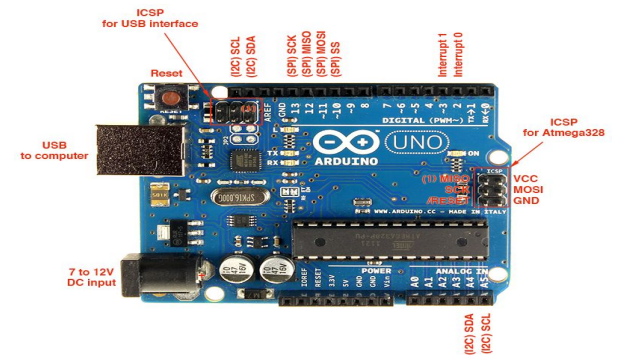
****

Figure 2.3 Arduino Uno

* **USB TO PC:** It is used to communicate Arduino via Universal Serial Bus to PC for programming/sending data to Arduino serially. 19
* **7V to 12V DC input: -** For external supply voltage. The recommended voltage range is 7-12V. A 9V battery can also be used to power the Arduino UNO.
* **Reset Button: -** It resets the Arduino board if pressed.
* **ICSP: -**An abbreviation for In Circuit Serial Programming which consists of MOSI, MISO, SCK, RESET, VCC and GND. It is either used to Programme USB or Microcontroller. It allows the user to Programme the microcontroller when it is in circuits i.e. it allows to Programme ATmega328p-pu directly with AVR instructions without using Arduino IDE.
* **SDA: -** SERIAL Data; it is the bidirectional data line that is used by 12c.
* **SCL: -** Serial Clock; it is used to indicate that data is ready on bidirectional data line that is used by 12C.
* **AREF: -** Analog Read Reference; it is mainly used for analog reference () function calls, as default it is not required but to use it you have to add some voltage source between 0V to 5V in AREF pin which will be considered as accurate reference voltage.
* **GND: -** Ground
* **SCK: -** Serial Clock, which is used by SPI (Serial Peripheral Interface). It is clock generated by a master which is used to clock the data to the slave.
* **MOSI: -** Master out Slave In; the data transmitted from master to slave
* **MISO: -** Master in Slave Out; the data transmitted to master from slave
* **SS: -**Slave Select; it is used to select the ‘slave’. Make high to SS pin to deactivate and make low to activate it.
* **INT1&INT0: -** These are hardware interrupts. It calls the ISR (Interrupt Service Routine) when the pin change occurs
* **TX: -** Transmit; it is used to transmit TTL serial data. Also referred to as ‘outwards’ since it transfers data from Arduino to the connected peripheral.
* **RX: -**Receive; it is used to receive TTL serial data. It is also referred to as inwards since it receives data from external hardware to Arduino.
* **Vin: -** Voltage In; if Arduino board is powered from USB then nothing is obtained from the Vin pin. But if the board is powered directly with an external supply, then that supply is obtained directly from Vin pin. However, the supply voltage obtained at the Vin pin is usually lesser by 1V than the voltage supplied due to reverse polarity protection diode.
* **5V Pin: -** it is used to power external components connected to Arduino which needs 5V.
* **3.3V Pin: -** it is used to power external components connected to Arduino that need 3.3V.
* **IORef: -** Input Output Voltage Reference; it allows shields connected to Arduino board to check whether the board is running at 3.3V or 5V

**Programming of arduino uno.**

* Arduino UNO is programmed in C/C++ language. The first step to programming Arduino is to download the Arduino software package (desktop version) from the Arduino download page. Once downloaded and installed, when one launches, it gives the page below:

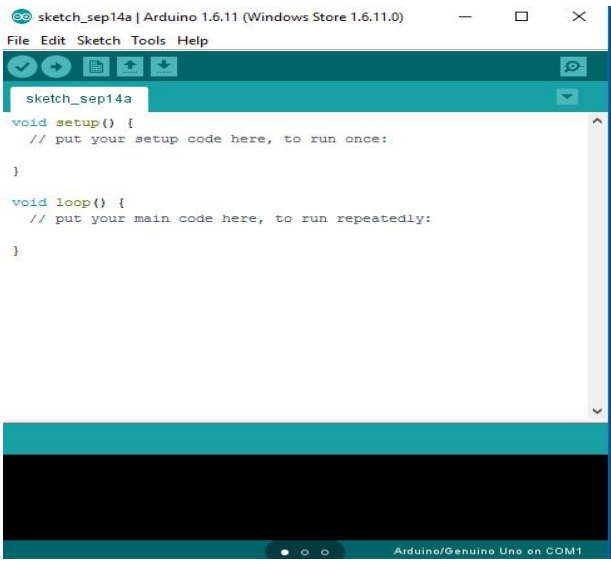
****

Figure 2.4.A sketch of Arduino IDE window

Every sketch needs two void types. The syntax of the Arduino IDE includes the setup () and loop (). Void is essentially a type of function that doesn’t return any value. Setup () is where you want to do any initialization steps while loop () is when you want to run the code over and over again.

In Arduino there are two types of special functions. These are pinMode and digitalWrite. pinMode takes two variables; the pin number and whether it’s an input or output pin. digitalWrite also takes two values; the pin number and the level whether high or low. High also means on-state and low means off-state.

**2.6 LCD (Liquid Crystal Display)**

The most basic and commonly used LCDS are the 16×12 because they are cheap, easy to program and can display wide range of characters. This is a 16-pin device which displays 16 × 2 characters. In this project the 16×12 LCD is used to display the state of heater or fan and the TMP36 readings.

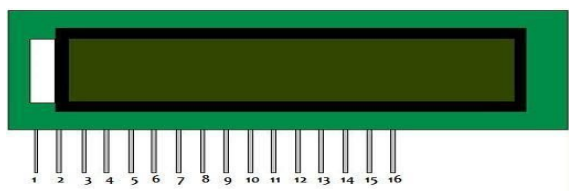


Figure 2.5. 16 ×12 Liquid Crystal Display.

When using 8-bit configuration all 8 data pins (DB0-DB7) are used while only 4 data pins (DB4DB7) are used in a 4-bit configuration.

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Function** | **Symbol** |
| 1 | Ground (0V) | VSS |
| 2 | Supply Voltage (5V) | VDD |
| 3 | Contrast adjustment; through a variable resistor(potentiometer) | V0 |
| 4 | Selects command register when low; and data register when high | RS |
| 5 | Low to write to the register; High to read from the register | RW |
| 6 | Sends data to data pins when a high to low pulse is given | E |
| 7 | 8-bit data pins | D0 |
| 8 | 8-bit data pins | D1 |
| 9 | 8-bit data pins | D2 |
| 10 | 8-bit data pins | D3 |
| 11 | 8-bit data pins | D4 |
| 12 | 8-bit data pins | D5 |
| 13 | 8-bit data pins | D6 |
| 14 | 8-bit data pins | D7 |
| 15 | Backlight Vcc (5V) | A |
| 16 | Backlight Ground (0V) | K |

Table 2.2. LCD pins ant their connections

**References**

The web sites that provide the information’s:

[www.microchip.co.uk](http://www.microchip.co.uk)

[www.quora.com](http://www.quora.com)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[www.wikipedia.com](http://www.wikipedia.com)

[www.engineeringproject.com](http://www.engineeringproject.com)

[www.electronicsworldforyou.com](http://www.electronicsworldforyou.com)

[www.tinkercad.com](http://www.tinkercad.com)

[www.arduinoide.com](http://www.arduinoide.com)

Arduino. (2018). Arduino - Software. <https://doi.org/28/12/2008s>