ISSN: 2278-0181
ICEECT - 2020 Conference Proceedings

Temperature Logging System using Can Protocol

Sivaprasath.K[1], Anusha.S[2], Gaurav Kumar.S[3], Jobish Revanth.S[4], Nandhini.S[5]
1 Assistant Professor, Department of ECE, Sri Shakthi Institute of Engineering & Technology, Coimbatore.
2,3,4,5 Under Graduate Student, Department of ECE, Sri Shakthi Institute of Engineering & Technology, Coimbatore.

Abstract: **The main purpose of this project is to monitor temperature using can protocol. A temperature monitoring system will help to avoid heat buildup at your Telecom locations like huts and other network industrial "IT" locations like server rooms and data centers. CAN (controller area network) is a bus system, which was originally developed for automotive application in the early 1980's.CAN is a serial communication protocol that may be used to transfer up to 8 bytes within a single message. The main objective of this project is to monitor and store the temperature value of the device continuously and also in this we can store temperature value of multiple device at the same time. It is an embedded based project which is a combination of hardware and software. Hardware includes connection of temperature sensor, Buzzer and CAN module. Software programming is done using Microsoft visual studio (.net).**

1. INTRODUCTION

1.1 TEMPERATURE MEASUREMENT

Temperature measurement in today's industrial environment encompasses a wide variety of needs and applications. To meet this wide array of needs the process controls industry has developed a large number of sensors and devices to handle this demand, many processes must have either a monitored or controlled temperature.

In our project is to monitor and store the temperature value of a multiple server continuously at the same time We will set the maximum and minimum value of the temperature so that if the temperature value does not lies between the maximum and minimum value the buzzer we are measuring a server temperature using CAN protocol.

1.2 CAN PROTCOL

Steve Corrigan SLOA101B–August 2002–Revised May 2016 described about CAN (Controller Area Network) is a serial bus system, which was originally developed for automotive applications in the early 1980's. CAN is a serial communication protocol that may be used to transfer up to 8 data bytes within a single message. CAN application tells communication controller to send/receive messages with a certain identifier (11 or 29 bits), the communication controller tries to send immediately, but can be delayed by current traffic and by arbitrating against higher-priority messages currently transmitted by other nodes. Messages have a transmission latency that depends on the concurrent higher-priority messages; changing the message set/priority/transmit rate results in changes of the communication timing that are difficult to predict.

1.3 PROBLEM STATEMENT

In existing model, the temperature values of the multiple devices are displayed on the LCD but the temperature value cannot be store and also, we cannot monitor and store value of multiple devices.

The manufacturing cost of other protocol is very high and it is less efficient. In order to improve the efficiency and monitor and store the temperature value of a multiple server continuously.

1.4 OBJECTIVE

The objective of our project is

i. To affordable for all small area region.

ii. To reduce the cost using CAN protocol.

iii. To increase the efficiency using CAN protocol.

iv. The main objective of this project is to monitor and store the temperature value of a multiple device continuously.

v. No need of man power

vi. Fast and efficient.

1.5 SCOPE

i. This project is to monitor and store the server temperature value of a multiple device continuously.

ii. Digital value is converted to CAN format using CAN module. This CAN format has start bit and end bit which indicates the starting and ending of the CAN packet.

iii. These data are entered in Notepad continuously and stored with date and time, which is used in JIO server.

iv. Using cooler, the server heat will be reduced.

v. This project is to product the server from the damage.

1.6 APPLICATIONS

i. It is used to sense the server temperature.

ii. It stores the previous temperature data in Notepad.

iii. It is used to monitor multiple server temperatures.

2.PROPOSED TEMPERATURE MONITORING ALGORITHM

The aims of this project are to monitor and store the server temperature value of a multiple device continuously. We proposed a system in which the server temperature monitoring is done with ease at a suitable distance with the help of CAN protocol.

2.1 PROBLEM IDENTIFICATION

In Industry to provide the temperature values of the multiple devices are displayed on the LCD but the temperature value cannot be store and also, we cannot monitor and store value of multiple devices. To overcome this, we are using that monitor and store the server temperature value of a multiple device continuously. At the same time, we will set the maximum and minimum value of the temperature so that if the temperature value does not lie between the maximum and minimum value the buzzer will turn on so that we easily avoid the server temperature.

This project its concern to temperature monitoring, we can also maintain the temperature by connecting it with the temperature cooling system so that temperature can be maintained.

2.2 HARDWARE DESCRIPTION

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers. Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result. The system using temperature sensors with Microcontrollers is an exclusive work that can automatically monitor & control the temperature according to the user requirement. The system comprises of

i Temperature sensor to know the ambience temperature.

ii CAN module

iii ADC – Op-Amp Lm358

iv Microcontroller

It is an embedded based project which is a combination of hardware and software. Hardware includes connection of Temperature Sensor, Buzzer and CAN module. Software programming is done using Microsoft visual studio(.net).

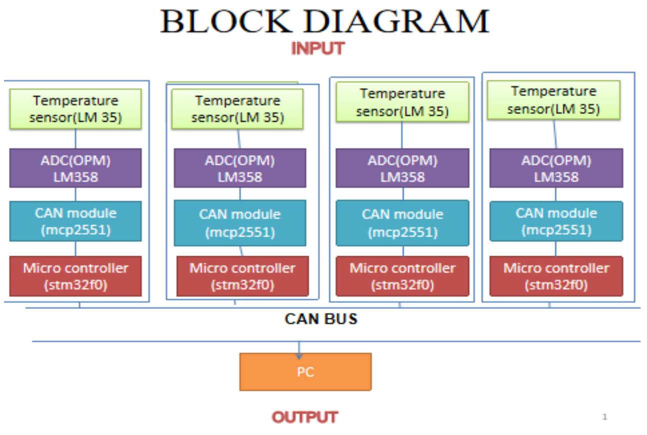

BLOCK DIAGRAM
INPUT
Temperature sensor(LM 35)
ADC(OPM) LM358
CAN module (mcp2551)
Micro controller (stm32f0)
Temperature sensor(LM 35)
ADC(OPM) LM358
CAN module (mcp2551)
Micro controller (stm32f0)
Temperature sensor(LM 35)
ADC(OPM) LM358
CAN module (mcp2551)
Micro controller (stm32f0)
Temperature sensor(LM 35)
ADC(OPM) LM358
CAN module (mcp2551)
Micro controller (stm32f0)
CAN BUS
PC
OUTPUT

Figure 2.1: Block Diagram

Temperature sensor will measure through temperature and this analog value will be converted to digital value by operational amplifier which act as ADC.

These digital values are converted to CAN packet using CAN module. This CAN format has start bit and end bit which indicates the starting and ending of the CAN packet. These data are entered in notepad continuously and stored with date and time. The fig 3.1 shows the block diagram of our proposed system which uses Temperature sensor and ADC.

2.2.1 TEMPERATURE SENSOR LM35

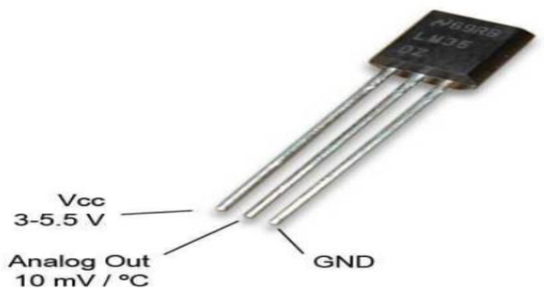


Figure 2.2: Temperature Sensor lm35

N Dinesh Kumar., (2014) described about LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any microcontroller that has ADC function or any development platform like Arduino. Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. Now, you can measure the temperate in form of voltage as shown below.

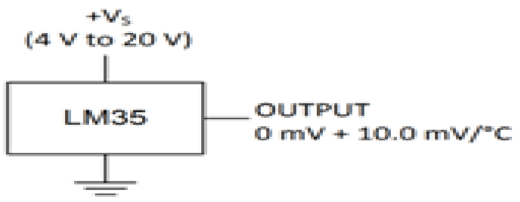


Figure2.3: Pin configuration of LM35

If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can be converted into temperature using the below formulae.

V_{OUT} = 10 mv/°C × T

where

- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

. 2.2.2 OPAMP LM358

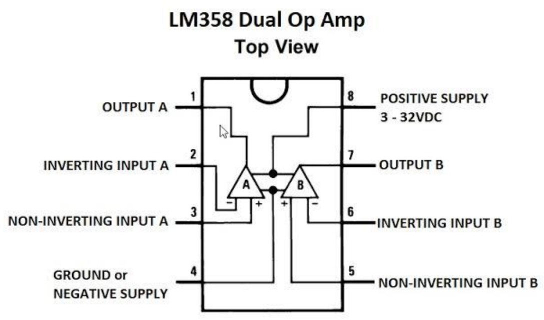


Figure 2.4: LM358 Op Amp Top View

Karumuri Anusha Reddy., (2019). described about The LM358 IC is a great, low power and easy to use dual channel op-amp IC. It is designed and introduced by national semiconductor. It consists of two internally frequency compensated, high gain, independent op-amps. This IC is designed for specially to operate from a single power supply over a wide range of voltages. The LM358 IC is available in a chip sized package and applications of this op amp include conventional op-amp circuits, DC gain blocks and transducer amplifiers. LM358 IC is a good, standard operational amplifier and it is suitable for your needs. It can handle 3-32V DC supply & source up to 20mA per channel. This op-amp is apt, if you want to operate two separate op-amps for a single power supply. It's available in an 8-pin DIP package.

2.2.3 CAN MODULE MCP2551

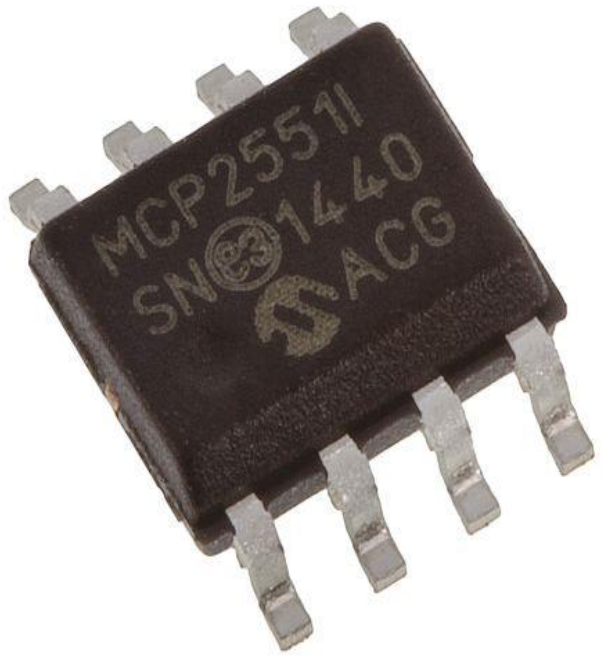


Figure 2.5: CAN Module

Microchip Technology Inc (published on 2014-2017) has been described about the MCP2551 is a high-speed CAN, fault-tolerant device that serves as the interface between a CAN protocol controller and the physical bus. The MCP2551 provides differential transmit and receive capability for the CAN protocol controller and is fully compatible with the ISO-11898 standard, including 24V requirements. It will operate at speeds of up to 1 Mb/s. Typically, each node in a CAN system must have a device to convert the digital signals generated by a CAN controller to signals suitable for transmission over the bus cabling (differential output). It also provides a buffer between the CAN controller and the high-voltage spikes that can be generated on the CAN bus by outside sources (EMI, ESD, electrical transients, etc.).

2.2.4 MICROCONTROLLER STM32F0

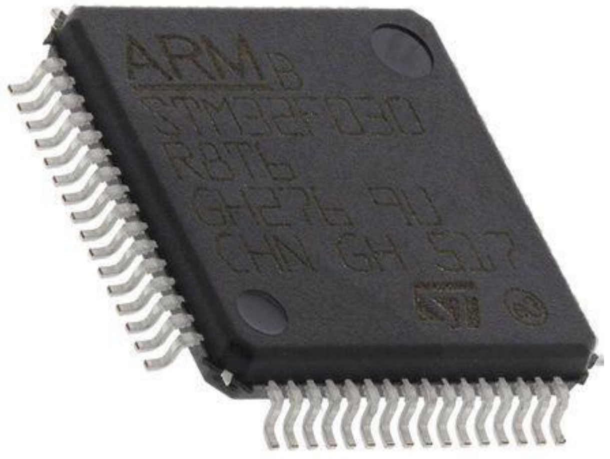


Figure2.6: Microcontroller STM32F0

STMicroelectronics STM32 F0 Entry-level ARM Cortex-M0 MCUs deliver 32-bit performance while featuring the essentials of the STM32 family and are particularly suited for cost-sensitive applications. STM32 F0 MCUs combine real-time performance, low-power operation, and the advanced architecture and peripherals of the STM32 platform. The STM32F0x0 is highly competitive in traditional 8-bit and 16-bit markets and eliminates the need to manage different architectures and the associated development overhead. The STM32F0x1 line provides a high integration of functions and covers a wide range of memory sizes and packages, bringing flexibility to cost-sensitive applications. The STM32F0x2 line provides rich connectivity with crystal-less USB 2.0 and a CAN bus interface, making it the ideal choice for communication gateways, smart-energy devices or game terminals. The STM32F0x8-line operating at 1.8V ±8% is well suited for use in portable consumer applications such as smartphones, accessories and media devices. The STM32 turns the one-architecture-fits-all concept into reality. The STM32 Nucleoboard provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

2.2.5 CAN MODULE

SLLS983J –JUNE 2009–REVISED SEPTEMBER 2019, The ISO1050 isolated transceiver described about The MCP2551 is a high-speed CAN, fault-tolerant device that serves as the interface between a CAN protocol controller and the physical bus. The MCP2551 provides differential transmit and receive capability for the CAN protocol controller and is fully compatible with the ISO-11898 standard, including 24V requirements. It will operate at speeds of up to 1 Mb/s. Typically, each node in a CAN system must have a device to convert the digital signals generated by a CAN controller to signals suitable for transmission over the bus cabling (differential output). It also provides a buffer between the CAN controller and the high-voltage spikes that can be generated on the CAN bus by outside sources (EMI, ESD, electrical transients, etc.)

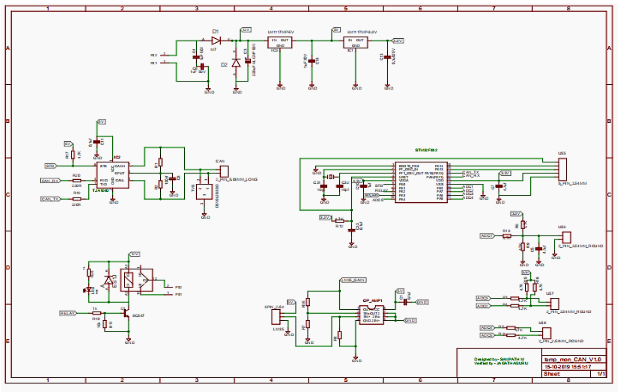


Figure2.7: CAN module

2.3 SOFTWARE DESCRIPTION

VISUAL STUDIO (.net)

The graphical user interface describes anything your application displays to the user. It is the primary way you interact with the user and allow him or her to interact with you. we can create professional interfaces with minimal effort. Graphical User Interfaces have been around for many years. Visual C++ 2008 provides a powerful and flexible development environment for creating Microsoft Windows–based and Microsoft .NET–based applications. Here we use Visual C++ 2008 in an integrated development system.

2.4 CAN PROTOCOL

S. Vijayalakshmi., (2013) descried about CAN application tells communication controller to send/receive messages with a certain identifier (11 or 29 bits).(Rev. 02 — 10 November 2006 Application note) the communication controller tries to send immediately, but can be delayed by current traffic, and by arbitrating against higher-priority messages currently transmitted by other nodes_ Messages have a transmission latency that depends on the concurrent higher-priority messages; changing the message set/priority/transmit rate results in changes of the communication timing that are difficult to predict. Stuart Robb East Kilbride, Scotland. (AN1798) described about Identifiers can be added to an existing network by only changing the sender and relevant receivers, but the message timing changes for all nodes. Arbitration between concurrent messages is non-destructive and therefore efficient, but limits the bit rate (< 1 Mbit/s) and the topology (no star or ring, no repeaters) and has severe safety issues.

3. RESULTS AND DISCUSION

In this project the temperature is sensed and monitored as explained in sub-chapter 3. The overall specification of the project along with its analysis are detailed in this chapter. In addition, the results and the working of temperature logging system are discussed.

3.1 SERVER TEMPERATURE

This project is to monitor and store the temperature value of a multiple devices continuously. At the same time, we will set the maximum and minimum value of the temperature so that if the temperature value does not lie between maximum and minimum value buzzer will turn on

3.2 METHOD STUDY

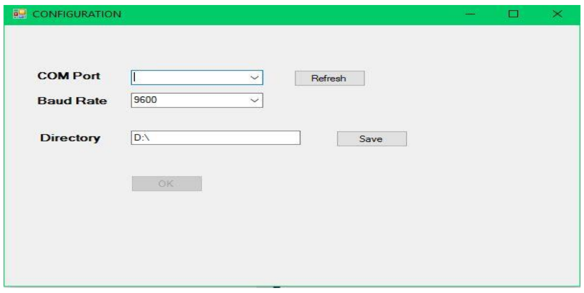
The method study of components for the Temperature logging system are studied in detail. The method study manly involves the different operation involved in the temperature logging system and sequence of its operation.

3.3 INSTALLATION OF SETUP FILE

1. Click on setup file and run it.
2. Then click on next button
3. Browse and select the location, where the file need to installed
4. Then click on next button
5. Again, click on next button
6. Now your file will be installed successfully
7. Then click on close button to exit from the window

3.4 FRONT PANEL

In the front panel there are COM port, Baud rate and directory. The COM port displays the port were the product is connected, Baud rate shows the frequency of the board connected, Directory displays the location were the values are to be stored.

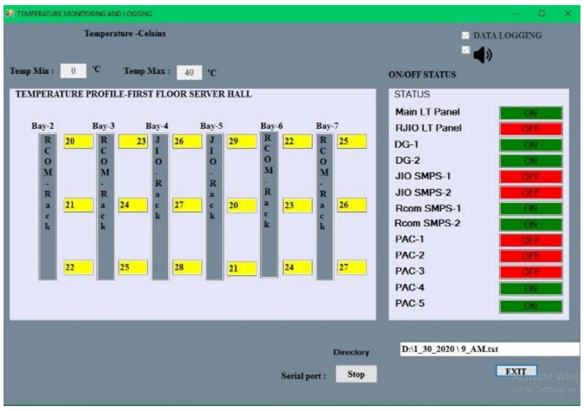


CONFIGURATION

COM Port:
Baud Rate: 9600
Directory: D:\

Figure 3.1: Temperature Profile

In this fig the there are several servers are monitored and the status of the server are also shown. We can also control the server by switching the servers ON and OFF. We can set the minimum and maximum` values of the temperature. If the value does not lie between minimum and maximum value the buzzer will turn on for 40seconds. Once the buzzer turns on the storing of values stops, after the value lies between the minimum and maximum value the storing process starts from the beginning not from the place where the process stops.



TEMPERATURE MONITORING AND LOGGING

Temperature -Celsius

Temp Min : 0 'C   Temp Max : 40 'C

DATA LOGGING ✓

TEMPERATURE PROFILE-FIRST FLOOR SERVER HALL

Bay-2		Bay-3		Bay-4		Bay-5		Bay-6		Bay-7	
RCOM-Rack	20	RCOM-Rack	23	JIO-Rack	26	JIO-Rack	29	RCOM-Rack	22	RCOM-Rack	25
	21		24		27		20		23		26
	22		25		28		21		24		27

ON/OFF STATUS

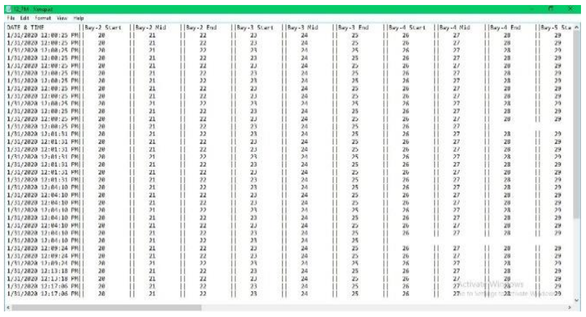
STATUS	
Main LT Panel	ON
RJIO LT Panel	OFF
DG-1	ON
DG-2	ON
JIO SMPS-1	OFF
JIO SMPS-2	OFF
Rcom SMPS-1	ON
Rcom SMPS-2	ON
PAC-1	OFF
PAC-2	OFF
PAC-3	OFF
PAC-4	ON
PAC-5	ON

Directory: D:\1_30_2020 \ 9_AM.txt

Serial port : Stop

Figure 3.2: Values in word document

In this page the temperature values of the several servers are stored with date and time. Time includes hours minutes and seconds



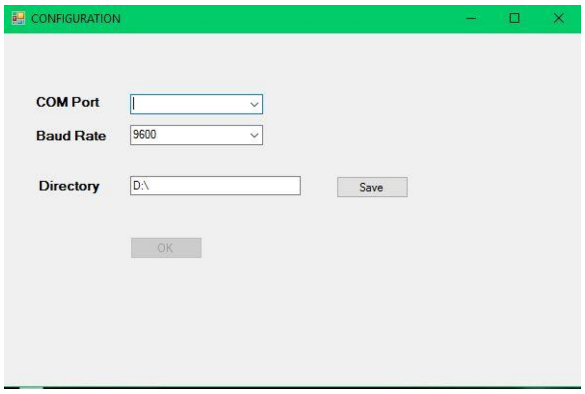
12_PM - Notepad

```
File  Edit  Format  View  Help
DATE & TIME          ||Bay-2 Start  ||Bay-2 Mid  ||Bay-2 End  ||Bay-3 Start  ||Bay-3 Mid  ||Bay-3 End  ||Bay-4 Start  ||Bay-4 Mid  ||Bay-4 End  ||Bay-5 Sta
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:00:25 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:01:31 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:04:10 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||
1/31/2020 12:09:24 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:09:24 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:09:24 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:13:18 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:13:18 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:17:06 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
1/31/2020 12:17:06 PM||     20      ||    21     ||    22     ||     23      ||    24     ||    25     ||     26      ||    27     ||    28     ||    29
```

Figure 3.3: Values in Notepad

3.5 SETUP FILE

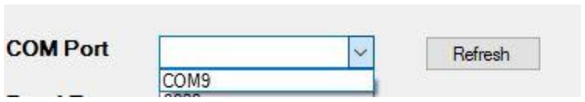
1. Click on the application file, **CONFIGURATION** form will open.



CONFIGURATION

COM Port:
Baud Rate: 9600
Directory: D:\

2 Click on **COM Port**, it will show the available COM Port

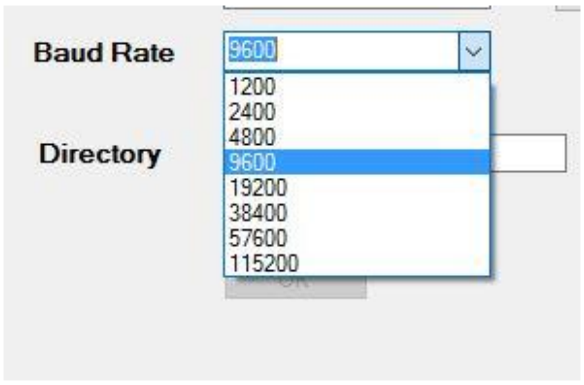


COM Port: COM9

3.If no COM Port is detected then click on **Refresh button** to scan for COM Port once again

4. Now select the COM Port where the hardware is connected

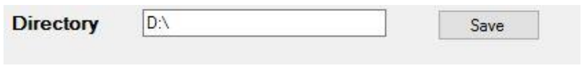
5.Next click on Baud Rate and select the **Baud Rate** as 9600 in the combo box



Baud Rate: 9600

```
1200
2400
4800
9600
19200
38400
57600
115200
```

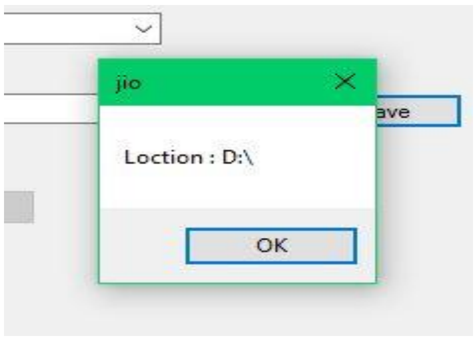
6. Now type the location where the logged data value should be stored in computer



Directory: D:\

7. click on **Save button** to create the text document in that location

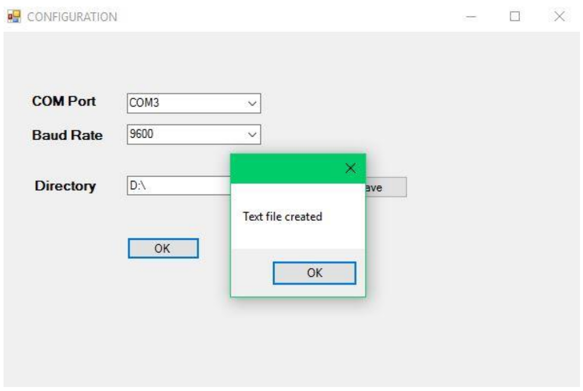
8. After clicking Save button, the location where the text document will be created is displayed in message box



jio

Loction : D:\

9. Next click on **OK Button** then text file will be created in the location where we typed and a message box will be displayed as "Text fie created".

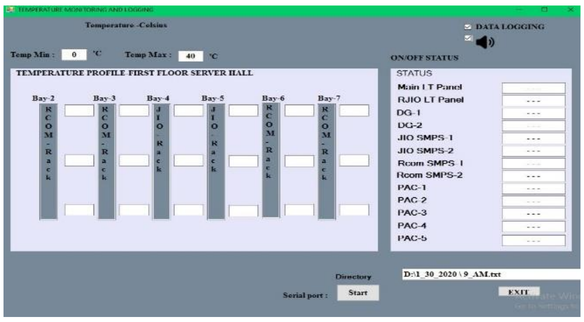


CONFIGURATION

COM Port: COM3
Baud Rate: 9600
Directory: D:\

Text file created

10. After clicking Ok in message box **TEMPERATURE MONITORING AND LOGGING** form will be opened



TEMPERATURE MONITORING AND LOGGING

Temperature -Celsius

DATA LOGGING

Temp Min : 0 'C Temp Max : 40 'C

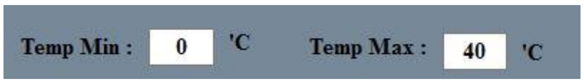
TEMPERATURE PROFILE FIRST FLOOR SERVER HALL

ON/OFF STATUS

STATUS	
Main LT Panel	---
RJIO LT Panel	---
DG-1	---
DG-2	---
JIO SMPS-1	---
JIO SMPS-2	---
Rcom SMPS-1	---
Rcom SMPS-2	---
PAC-1	---
PAC-2	---
PAC-3	---
PAC-4	---
PAC-5	---

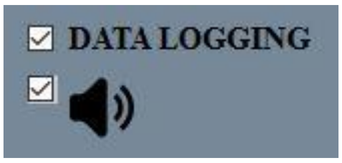
Directory: D:\1_30_2020 \ 9_AM.txt

11. Next set the minimum and maximum temperature value so that the buzzer will turn on when temperature value measured does not lies between minimum and maximum temperature value.

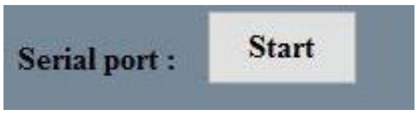


Temp Min : 0 'C Temp Max : 40 'C

12. Next check if Data logging and Buzzer are marked enable so that these functions will work in the application, if these functions are not required then unmark them so that they are not enable

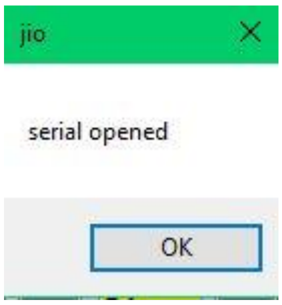


13. Now click on **Start button** so that serial port will become open and start receiving data.



14. Once the **Start button** is click it will be disabled and **Stop button** will be enabled

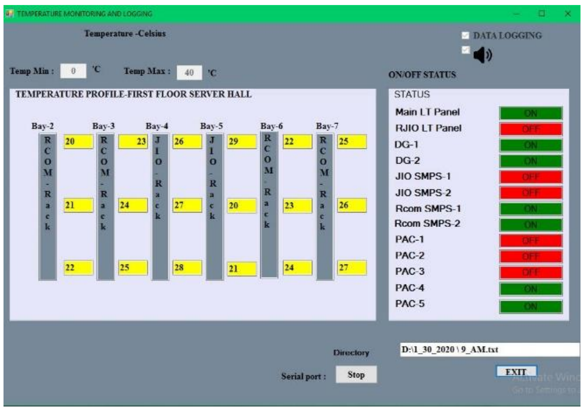
15. If **Start button** is clicked it will show a message box that serial opened.



jio

serial opened

16. Then data will be start to logging in the front panel.



TEMPERATURE MONITORING AND LOGGING

Temperature -Celsius

DATA LOGGING

Temp Min : 0 'C Temp Max : 40 'C

TEMPERATURE PROFILE-FIRST FLOOR SERVER HALL

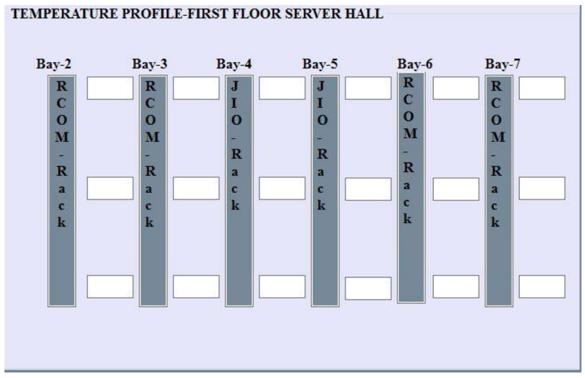
Bay-2	Bay-3	Bay-4	Bay-5	Bay-6	Bay-7
20	23	26	29	22	25
21	24	27	20	23	26
22	25	28	21	24	27

ON/OFF STATUS

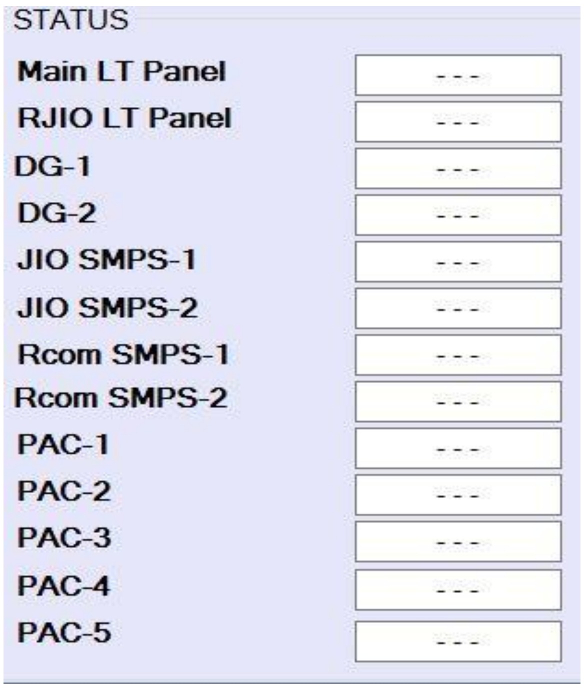
STATUS	
Main LT Panel	ON
RJIO LT Panel	OFF
DG-1	ON
DG-2	ON
JIO SMPS-1	OFF
JIO SMPS-2	OFF
Rcom SMPS-1	ON
Rcom SMPS-2	ON
PAC-1	OFF
PAC-2	OFF
PAC-3	OFF
PAC-4	ON
PAC-5	ON

Directory: D:\1_30_2020 \ 9_AM.txt

17. Here the SERVER temperature value will be displayed.



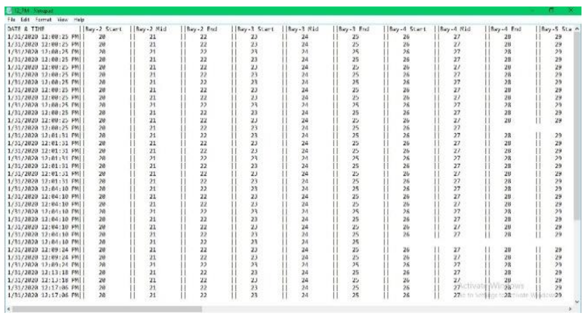
18. Here the SERVER On/Off status will be displayed



STATUS

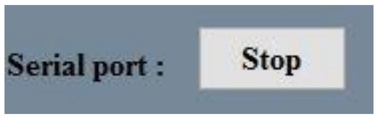
Main LT Panel	- - -
RJIO LT Panel	- - -
DG-1	- - -
DG-2	- - -
JIO SMPS-1	- - -
JIO SMPS-2	- - -
Rcom SMPS-1	- - -
Rcom SMPS-2	- - -
PAC-1	- - -
PAC-2	- - -
PAC-3	- - -
PAC-4	- - -
PAC-5	- - -

19. At the same time data will be stored in Text document along with date and time.

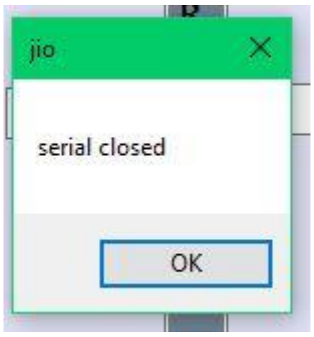


```
DATE & TIME            ||Bay-2 Start ||Bay-2 Mid ||Bay-2 End ||Bay-3 Start ||Bay-3 Mid ||Bay-3 End ||Bay-4 Start ||Bay-4 Mid ||Bay-4 End ||Bay-5 Sta
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:00:25 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:01:31 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26    ||   27     ||   28     ||   29
1/31/2020 12:04:10 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25
1/31/2020 12:09:24 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:09:24 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:09:24 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:13:18 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:13:18 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:17:06 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
1/31/2020 12:17:06 PM||    20      ||   21     ||   22     ||   23       ||   24     ||   25     ||   26       ||   27     ||   28     ||   29
```

20. Click on **Stop button** to close the serial port so that data will be stopped.



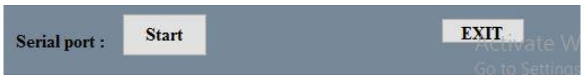
21. Once **Stop button** is clicked a message box will display that serial closed.



serial closed

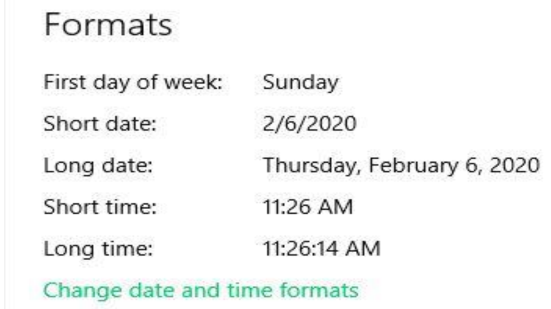
22. Once the **Stop button** is clicked then **Start button** will become enable and **Stop button** will become disable, so that if we want to start receiving data again then click on **Start button**

23. Click on **Exit button** to close the application



3.6 DATE AND TIME FORMAT

1. If the file name is not created properly like the below format then check the date and time format, check if the date and time is in the following format.

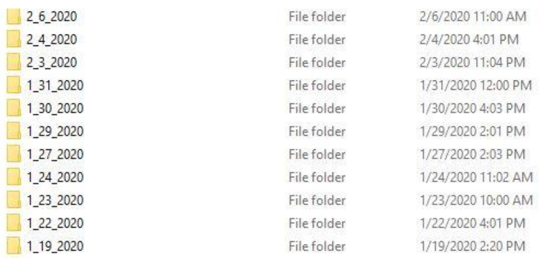


Formats

First day of week:	Sunday
Short date:	2/6/2020
Long date:	Thursday, February 6, 2020
Short time:	11:26 AM
Long time:	11:26:14 AM

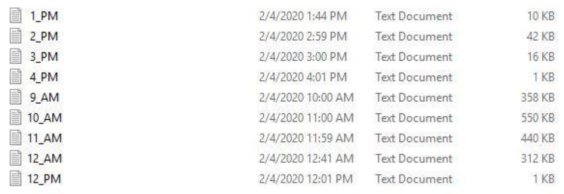
2. If date and time is in correct format then File name should be created in following format.

3. First folder should be created in the name of date.



2_6_2020	File folder	2/6/2020 11:00 AM
2_4_2020	File folder	2/4/2020 4:01 PM
2_3_2020	File folder	2/3/2020 11:04 PM
1_31_2020	File folder	1/31/2020 12:00 PM
1_30_2020	File folder	1/30/2020 4:03 PM
1_29_2020	File folder	1/29/2020 2:01 PM
1_27_2020	File folder	1/27/2020 2:03 PM
1_24_2020	File folder	1/24/2020 11:02 AM
1_23_2020	File folder	1/23/2020 10:00 AM
1_22_2020	File folder	1/22/2020 4:01 PM
1_19_2020	File folder	1/19/2020 2:20 PM

4. Second text document created in the name of time.



1_PM	2/4/2020 1:44 PM	Text Document	10 KB
2_PM	2/4/2020 2:59 PM	Text Document	42 KB
3_PM	2/4/2020 3:00 PM	Text Document	16 KB
4_PM	2/4/2020 4:01 PM	Text Document	1 KB
9_AM	2/4/2020 10:00 AM	Text Document	358 KB
10_AM	2/4/2020 11:00 AM	Text Document	550 KB
11_AM	2/4/2020 11:59 AM	Text Document	440 KB
12_AM	2/4/2020 12:41 AM	Text Document	312 KB
12_PM	2/4/2020 12:01 PM	Text Document	1 KB

4. CONCLUSION

This project its concern to temperature monitoring, we can also maintain the temperature by connecting it with the temperature cooling system so that temperature can be maintained. we can select TCP IP interface format for successful design of web-based control system in large area network. The project was completed within the given amount of time and the project ended on the successful note.

REFERENCES

[1]. N. Dinesh Kumar," Monitoring and controlling of temperature using can architecture, Best journal vol 2, Issue 6, Jun 2014,61-68
[2]. Florian Hartwich, "The Configuration of the CAN bit timing", BOSCH 6th international CAN Conference 2nd to 4th November, Turin (Italy)
[3]. Jadsonlee da Silva Sá*Monitoring of Temperature Using Smart Sensors Based on CAN Architecture publisher: IEEE
[4]. Proceedings of the 15th International Conference on Electronics, Communications and Computers (CONIELECOMP 2005)
[5]. Karumuri Anusha Reddy. Monitoring of Patients Health in Hospitals using CAN Protocol International Journal of Innovative Technology and Exploring Engineering (IJITEE)
[6]. ISSN: 2278-3075, Volume-8 Issue-6, April 2019
[7]. 5. Steve Corrigan," Introduction to the controller area network (CAN)", Texas instruments, Application report, SLOA101B-August 2002-Revised May-2016
[8]. 6. Stuart Robb," CAN bit timing requireents", Freescale semiconductor Inc. (2004)
[9]. 7.S. Vijayalakshmi Vehicle control system implementation
[10]. Using CAN protocol International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering