Temperature Logging System using Can Protocol

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 PROTOCOL

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).

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 temperature in form of voltage as shown below.
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.

\[ V_{\text{OUT}} = 10 \text{ mV/°C} \times T \]

where
- \( V_{\text{OUT}} \) is the LM35 output voltage
- \( T \) is the temperature in °C

2.2.2 OPAMP LM358

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

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

Microchip Technology Inc (published on 2014-2017) has been described about the STM32F0 is an ARM Cortex-M0-based microcontroller that combines high performance with low power consumption. It is designed for applications requiring high performance and power efficiency, such as industrial automation, embedded systems, and consumer electronics. The STM32F0 series includes various models with different flash memories and memory speeds, allowing for flexibility in choosing the right device for your specific needs.
The method study of components for the Temperature logging system are studied in detail. The method study mainly 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.

![Figure 3.1: Temperature Profile](image)

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 40 seconds. 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.

![Figure 3.2: Values in word document](image)

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

3.5 SETUP FILE

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

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

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

![Figure 3.3: Values in Notepad](image)
6. Now type the location where the logged data value should be stored in computer

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

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 file created”.

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

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.

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

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

![Temperature Profile First Floor Server Hall](image)

<table>
<thead>
<tr>
<th>Bay 1</th>
<th>Bay 2</th>
<th>Bay 3</th>
<th>Bay 4</th>
<th>Bay 5</th>
<th>Bay 6</th>
<th>Bay 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM Rck</td>
<td>COM Rck</td>
<td>COM Rck</td>
<td>COM Rck</td>
<td>COM Rck</td>
<td>COM Rck</td>
<td>COM Rck</td>
</tr>
</tbody>
</table>

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

![STATUS](image)

<table>
<thead>
<tr>
<th>Status</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main LT Panel</td>
<td>---</td>
</tr>
<tr>
<td>RJIO LT Panel</td>
<td>---</td>
</tr>
<tr>
<td>DG-1</td>
<td>---</td>
</tr>
<tr>
<td>DG-2</td>
<td>---</td>
</tr>
<tr>
<td>JlO SMPS-1</td>
<td>---</td>
</tr>
<tr>
<td>JlO SMPS-2</td>
<td>---</td>
</tr>
<tr>
<td>Rcom SMPS-1</td>
<td>---</td>
</tr>
<tr>
<td>Rcom SMPS-2</td>
<td>---</td>
</tr>
<tr>
<td>PAC-1</td>
<td>---</td>
</tr>
<tr>
<td>PAC-2</td>
<td>---</td>
</tr>
<tr>
<td>PAC-3</td>
<td>---</td>
</tr>
<tr>
<td>PAC-4</td>
<td>---</td>
</tr>
<tr>
<td>PAC-5</td>
<td>---</td>
</tr>
</tbody>
</table>

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

![Serial Port](image)

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

![Message Box](image)

23. Click on Exit button to close the application

![Exit Button](image)

**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.

   ![File Folders](image)

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

This project is concerned with 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

[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
[6]. ISSN: 2278-3075, Volume-8 Issue-6, April 2019
[7]. S. Vijayalakshmi Vehicle control system implementation