WSN based Online Parameter Monitoring in Cold Storage Warehouses in Cloud using IOT concepts

Bindu.J.M1, Nikitha2, Namitha.M3 , Pradeep.H.S4

[1,2,3] *B.E Student, Dept. Of ECE, Siddaganga Institute of Technology, Tumakuru, Karnataka.*
[4]*Assistant Professor, Dept. Of ECE, Siddaganga Institute of Technology, Tumakuru, Karnataka.*

Abstract - *Cold storage warehouse management system is a highly competitive business when sensitive products with short shelf life are stored. Such systems require critical monitoring and controlling of temperature and humidity.*

This paper proposes Wireless Sensor Network (WSN) based continuous monitoring of temperature and humidity of the cold storage warehouses using IOT concepts in order to ensure that the stored food products are not decayed due to increase in temperature and humidity.

The proposed solution continuously monitors temperature and humidity using PIC18 microcontroller. These parameters needs to be monitored continuously as they affect nutritional content of the food items stored in the cold storage. Further, the system allows the maintenance authority to keep track of the parameters online using an android application. A web page is also developed for representation of the data values sensed in real time and also for further analysis of the data. The system provides a facility to remotely control the temperature with an android app. An ON/OFF control of air conditioned system is used to maintain the required temperature.

Keywords: *Wireless Sensor Network (WSN), Cold storage, Internet of Things, Relative Humidity, Contiki OS, Thingspeak, MIT app.*

I. INTRODUCTION

A warehouse is a commercial building for storage of goods. Warehouses are used by manufacturers, importers, exporters, wholesalers, transport businesses, customs, etc. Strauss Frito-Lay uses a lot of warehouses to store large amount of Vegetables/ Fruits. These fruits and vegetables (Primarily Potatoes) require a certain temperature to be maintained for proper storage.

When the chiller or the compressor is switched ON, the RH (Relative Humidity) in the air starts diminishing. It also needs to be monitored else the potatoes will rot soon in the absence of water content in the air.

To overcome these issues, a Wireless Sensor Node based design is proposed which will be able to monitor various sections of the warehouse. It then sends the information to a central node in the warehouse which will collect the data and push it on the cloud.

Wireless Sensor Networks (WSN) sometimes called Wireless Sensor and Actuator Networks (WSAN) are spatially distributed autonomous sensors to monitor physical or environmental conditions such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to other locations. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. Today, such networks are used in many industrial and consumer applications such as industrial process monitoring and control, machine health monitoring and so on.

The objective of the paper is to monitor parameters like temperature and relative humidity in cold storage warehouses, so that stored products would not get contaminated due to surrounding conditions. Temperature is controlled by comparing with the threshold. If all sensed temperatures are less than threshold, compressor is switched OFF. If any of the sensed temperature is more than the threshold, then compressor is switched ON.

II. LITERATURE REVIEW

Many techniques were developed in recent years. In this section, a brief literature review is given.

In paper [1], authors have proposed a model that analyses temperature, moisture, light. A web server is used for storage of data values sensed in real time and also for analysis results. User is alerted via messages along with location of the shipment whenever an emergency occurs.

In paper [2], the authors used Zigbee and other wireless technology to construct warehouse environment monitoring network. The temperature, humidity, smoke, light are auto monitored and controlled. They realized monitoring systems on three platforms, including monitor terminal, base station web application and mobile devices.

In paper [3], it is based on a network of collected sensors which collect data and upload it to cloud. Monitoring system is based on Arduino which measures temperature and humidity.

In paper [4], model proposed was used to control corn products. System was provided with location and quality of corn products with tag in real-time. The authors used GUI in which the operator can go through login system and query the corn product by electronic tag. Real-time humidity, temperature as well as stock unit were shown on the GUI page.

In paper [5], the proposed system consists of two major parts namely-Wireless sensor network that monitors all the essential parameters (Temperature, Humidity, CO2 concentration) of the cold storage system and Fuzzy Logic Controller to control the parameters.

In paper [6], authors integrated wireless sensor network with solar energy. By using the voice detection sensors, detection of the animals was made based on their voice frequency. Voice sensors were placed at every corner of the warehouse. Also, sensor nodes collected measured data such as relative humidity, temperature and smoke which were required for determining the warehouse fire danger rate.

In paper [7], the whole system consisted of two wireless sensor networks based on Zigbee and Bluetooth. A relay was connected with the microcontroller to switch between Bluetooth and Zigbee units. After measuring the temperature and humidity, the data was sent to the monitoring unit. The Central monitoring unit receives the monitoring data of sensor units and stores them in the database, and also can call and print that at any time.

III. SYSTEM OVERVIEW

This section describes about the block diagram of the system, which contains two nodes namely Transmitter node and Receiver node.

The block diagram of the Transmitter node is shown in Fig.1. It includes Power supply, Temperature sensor LM-35, Humidity sensor HSM-20G, PIC18 microcontroller, Wi-Fi module ESP8266. The Power to the circuit is given by power supply of 5V. Temperature sensor LM-35 and Humidity sensor HSM-20G are connected to ADC channels of PIC18 microcontroller. The ADC values for temperature and humidity are processed in PIC microcontroller and are sent serially to Wi-Fi module ESP8266. The Wi-Fi module puts the data onto the cloud.

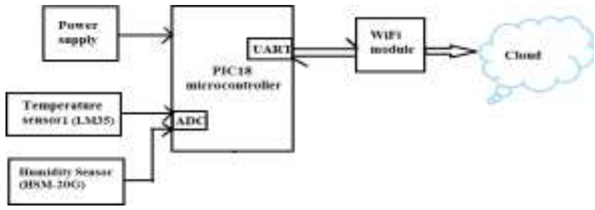

Fig.1: Block diagram of the Transmitter Node

The block diagram of the receiver node is shown in Fig.2. It includes LaunchXL-CC2650 (running with Contiki OS), Temperature sensor LM-35, Raspberry Pi, Relay and compressor. The data pushed onto the cloud by the transmitter node is collecetd by the built-in Wi-Fi module of Raspberry Pi. The current temperature is compared with the threshold and the relay switches ON the compressor if the temperature is more than threshold and turns OFF if the temperature is less than the threshold.

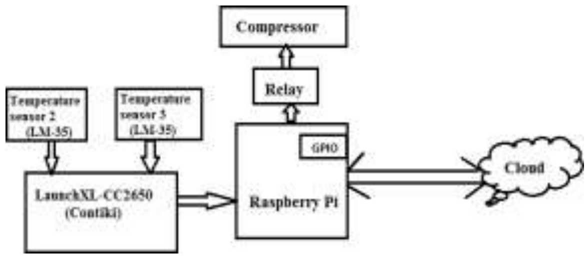

Fig.2: Block diagram of the Receiver Node

IV. HARDWARE IMPLEMENTATION

This section describes the hardware of the system. It includes LM-35, HSM-20G, PIC-18, ESP 8266, Raspberry Pi, LaunchXL- CC2650, Relay.

1. Temperature Sensor : LM35

LM35 is a temperature sensor ranges from -55°C to 150 °C. It draws 60μA from the power supply, so it is very low self-heating & operates in the range of 4V to 20V. It has 0.5°C ensured accuracy. The Vout pin of the LM35 is given to the ADC pin (RA0) of PIC 18 microcontroller as shown in Fig.3.

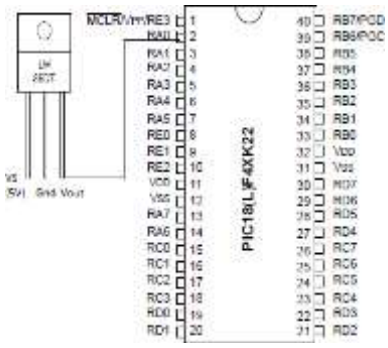

Fig.3: LM35 connected to PIC18

2. Humidity sensor: HSM-20G

HSM 20G convert relative humidity to standard output voltage. It operates in the input voltage range of 5 ± 0.2V. Its output voltage ranges from 1-3.19V. It operates in the range of -20°C to 70°C. The Humidity output pin is given to the ADC pin (RA2) of PIC 18 microcontroller. This is shown in Fig.4.

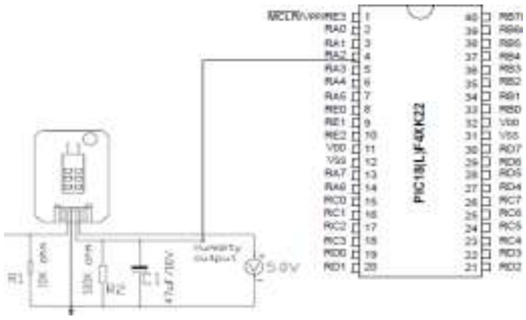

Fig.4: Humidity Sensor connected to PIC18

3. PIC 18 microcontroller

It is a RISC processor which operates in the voltage range of 2.3 to 5V. It has 1024 bytes Data EEPROM. It pushes the data to the cloud through Wi-Fi module. The ADC channels RA0 and RA2 are connected to LM35 and HSM-20G respectively. The Transmitter and Receiver pins of PIC18 (RD6 and RD7) are connected to Transmitter and Receiver pins of ESP 8266 Wi-Fi module as shown in Fig.5.

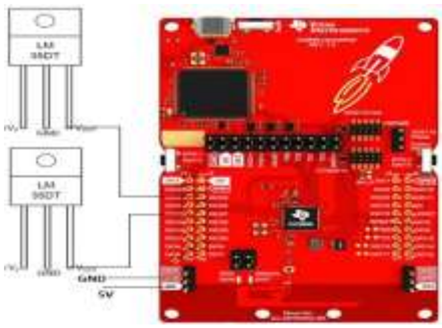

Fig.5: Circuit diagram of PIC18

4. Wi-Fi module ESP 8266

The Wi-Fi module ESP 8266 works on IPv4, TCP, UDP, HTTP, FTP protocols. Its operating voltage ranges from 2.5V to 3.6V and operates under the temperature range of -40°C to 125°C. It sends the temperature and humidity values to the cloud. The Wi-Fi module is connected to the PIC 18 microcontroller as shown in Fig.6. The Transmitter and Receiver pins of ESP 8266 Wi-Fi module (RD6 and RD7) are connected to Transmitter and Receiver pins of PIC 18.

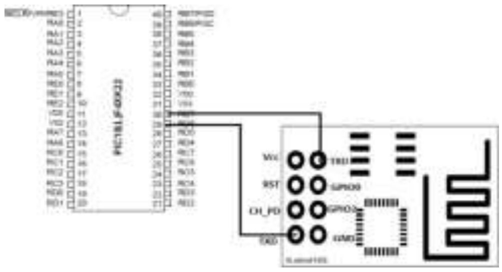

Fig.6: ESP 8266 connected to PIC 18

5. Raspberry pi B3

The Raspberry Pi B3 is a quad-core Broadcom BCM2837 64 bit processor. Its CPU frequency is 1.2GHz. The performance of the Pi 3 is roughly 50-60% faster than the Pi 2 which means it is ten times faster than the original Pi. The Raspberry Pi board runs from a 5V micro-USB power adapter.

The transmitter pin of LaunchXL-CC2650 is connected to the receiver pin of Raspberry Pi. It receives the data from cloud and controls the switching ON and OFF of the compressor based on the comparison between the temperatures and the threshold value. If all sensed temperatures are less than threshold, compressor is switched OFF. If any of the sensed temperature is more than the threshold, then compressor is switched ON. The GPIO 21 of Raspberry Pi is used to control the relay. When the pin is LOW, Common will be connected to Normally Closed terminal of relay. When the pin is high, Common will be connected to Normally Open terminal of relay.

6. LaunchXL - CC2650

The CC26x0 family of microcontrollers build on the ARM Cortex-M3 core to bring high performance 32-bit computing to cost-sensitive embedded microcontroller applications. The radio in the CC26x0 device operates in the 2.4GHz ISM frequency band. It has a Flash of 8KB with 4-way set-associative cache RAM for speed and low power. It also has a 12-bit analog-to-digital converter (ADC) with eight analog input channels. The GPIOs are with analog capability for ADC and comparator.

The ADC channels of CC2650 (DIO23 and DIO25) are connected to Vout pins of two Temperature sensors LM-35 as shown in Fig.7.

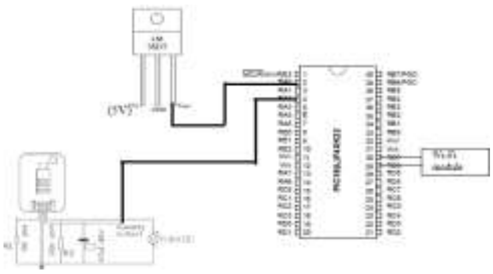

Fig.7: LaunchXL-CC2650

7. Relay

Relay operates as a switch. It is connected to the Raspberry Pi's GPIO pin through which it controls ON and OFF the compressor as shown in Fig.8. By default, common and Normally Open pin are not connected and hence compressor is switched OFF. When temperature is more than threshold, coil of the relay will get energised making a connection between Normally Open and Common. Hence the compressor will switch ON.

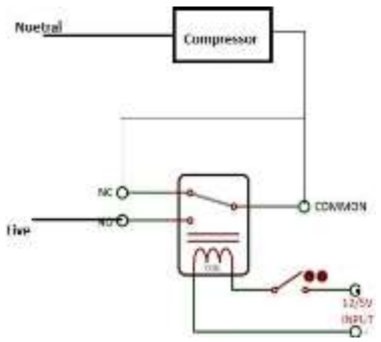

Fig.8: Relay

V. SOFTWARE IMPLEMENTATION

This section describes the software requirements of the system. It includes Contiki OS and the flowchart of the system.

1. Operating system: Contiki

Contiki is an open source operating system for the Internet of Things. It connects tiny low cost, low power microcontrollers to the Internet. It is a powerful toolbox for building complex wireless systems. It supports fully standard IPv6 and IPv4 & Contiki provides multitasking and a built-in Internet Protocol Suite (TCP/IP stack). It requires only about 10 kilobytes of Random Access Memory (RAM) and 30 kilo-bytes of Read Only Memory (ROM).

2. Steps to run a program in contiki

- Install Instant Contiki version 2.7 and VMware player version 12.5.9 softwares.
- Run VMware player and click open a virtual machine.
- Select Instant Contiki Ubuntu 12.04 32-bit.
- Open terminal and open the directory where the program is saved.
- Generate the hex file of the code using make command.
- Burn the hex file onto CC2650 board using Smart Flash Programmer 2.

3. MIT app inventor

App Inventor for Android is an open-source web application provided by Google, maintained by the Massachusetts Institute of Technology (MIT).

MIT app inventor uses a graphical interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. It is also supported with the Firebase Database extension. This allows people to store data on Google's firebase. The flowchart of the android app developed is shown in Fig.9.

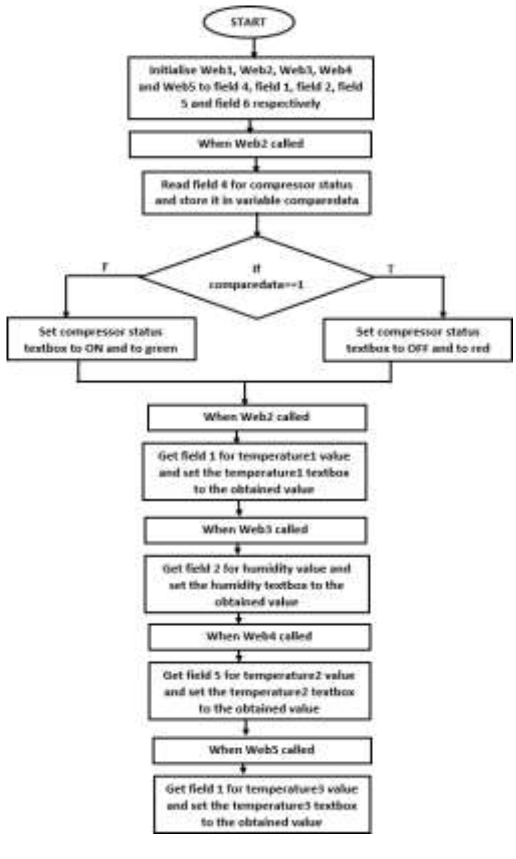

Fig.9: Flowchart of Android App

4. Thingspeak Website

ThingSpeak is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. ThingSpeak has integrated support from the numerical computing software MATLAB from Math Works, allowing ThingSpeak users to analyze and visualize uploaded data.

5. Steps to create a new channel

- Go to website *https://thingspeak.com*
- Create a Microsoft account.
- Sign in to the account created and go to *https://thingspeak.com/channels*
- Click on New channel.
- Enter the channel name and fields name.
- Save the channel.
- Note down the channel ID, Read API key, Write API key.

- Once the channel is created, data can be published by accessing the Thingspeak API with a 'Write key' and a 'Read key' is used to access the channel data.

6. Flowchart

The flowchart of the code implemented at the transmitter node is shown in Fig.10.

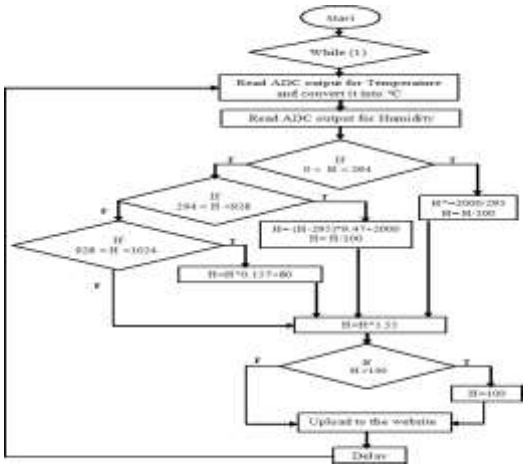

Fig.10: Flow chart of the Transmitter node

The logic includes reading the values of Temperature and Humidity from the respective ADC channels of PIC microcontroller. It is then processed and uploaded to the website. The flowchart of the Receiver node is shown in Fig.11. Raspberry Pi collects the values of Temperature 2 and Temperature 3 from CC2650 running on Contiki OS. It also receives the values of Temperature 1, Humidity, Threshold & previous Compressor status from the website. Temperature is then compared with the threshold and ON/OFF status of compressor is decided. The current compressor status is uploaded back to the website.

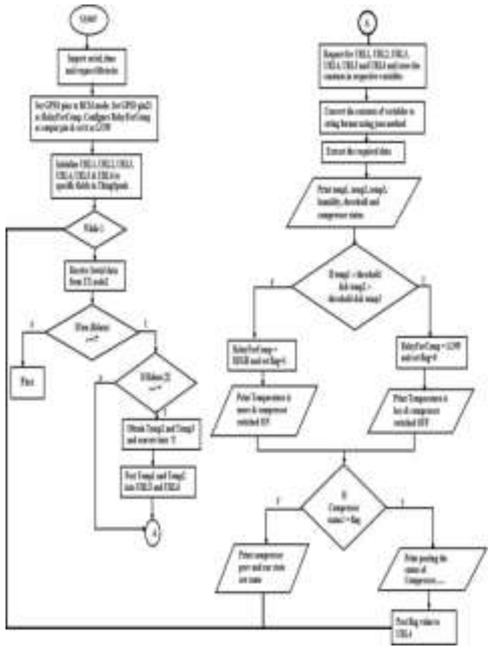

Fig.11: Flowchart of the Receiver node

VI. RESULTS

This section includes the tabulation of the compressor status for different threshold levels and the snapshots of the system.

Tabulation of Compressor Status

The compressor status are tabulated at different thresholds as shown in Table1.

Table 1: Compressor Status

Sl. No	Threshold(˚C)	Temp1 (˚C)	Temp2 (˚C)	Temp3 (˚C)	Compressor Status
1	20	22	29	28	ON
2	30	22	29	28	OFF
3	30	22	30	29	ON
4	40	26	29	30	OFF

Snapshots

Fig.12 indicates the snapshot of the Transmitter Node. It includes Temperature Sensor LM-35, Humidity Sensor HSM-20G, PIC 18 microcontroller, Wi-Fi module ESP8266.

Fig.12: Transmitter node

Fig.13 indicates the snapshot of the Receiver Node. It includes Raspberry Pi, Relay and LaunchXL-CC2650.

Fig.13: Receiver node

Fig.14(a) indicates that when any of the sensed temperature is more than the threshold, the compressor is switched ON and the current status of the compressor is updated in the website as shown in Fig.14(b).

(a)Android App

(b)Thingspeak Website

Fig.14: Compressor ON when any of the temperature is more than the threshold.

The temperature and humidity values are displayed on Raspberry Pi as shown in the Fig.15.

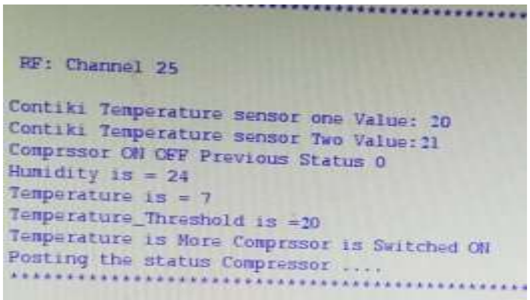

Fig.15: Output displayed on Raspberry Pi

Fig.16(a) indicates that when all the sensed temperatures are less than the threshold, the compressor is switched OFF and the current status of the compressor is updated in the website as shown in Fig.16(b).

(a)Android App

(b)Thingspeak Website

Fig.16: Compressor OFF when all sensed temperature is less than the threshold

The temperature and humidity values are displayed on Raspberry Pi as shown in the Fig.17.

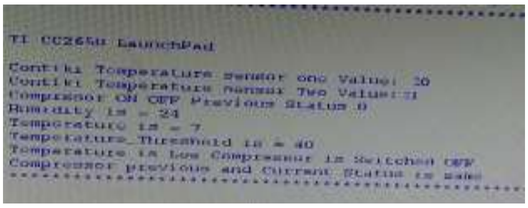

Fig.17: Output displayed on Raspberry Pi

VII. CONCLUSIONS

The system is designed and tested successfully for various temperatures. For different temperature thresholds, compressor is controlled satisfactorily. The temperature and humidity values are sent to the cloud by the Transmitter node. The Receiver node receives the monitored data from different sensor nodes and stores them in the database. The stored values are displayed in the form of a graph in the developed website. Temperature is controlled by comparing with the threshold. The status of the compressor along with the temperature and humidity values are displayed in an android application. Since Wireless Sensor Network (WSN) concept is used, the system is insensitive to distance between Transmitter and Receiver node.

Scope for future work

In this paper, essential parameters like Temperature and Relative Humidity were monitored. Other parameters like CO_2, smoke, light can also be monitored. The number of sensing nodes can be increased, so that temperature and humidity can be monitored and controlled over a large area.

REFERENCES

[1] Amrita Srivastava, Ankita Gulati, "iTrack: IoT frameworkforSmartFoodMonitoring System", International Journal of Computer Applications, 2016.

[2] Qinghua Zhang, Yi Wang, "Warehouse environment monitoring system based on WSN", IEEE 9th Conference on Industrial Electronics and Applications (ICIEA), 2014.

[3] Abel Avitesh Chandra, Seong-Ro Lee, "Advanced Monitoring of cold chain using WSN", International electronic conference on sensors and applications, 2014.

[4] Yi-ping Chen, Jian Wang,"A Corn Warehouse Monitoring System Based on IOT", International Conference on Materials, Manufacturing and Mechanical Engineering, 2016.

[5] A.Sathish Kumar, S.Sudha, "Design of Wireless Sensor Network based Fuzzy Logic Controller for a Cold Storage System", IEEE 7th power India International Conference (PIICON), 2016.

[6] Manoj Kumar Tyagi, Balanagu Raviteja, Rammohanarao Koduri and Gangi Raghu Ram, "Warehouse monitoring and management with the help of WSN and Solar energy", International journal of engineering science and advanced technology, Volume-2, Issue-3, pp 503–507, 2012.

[7] Sandeep Kaushik, Charanjeet Singh, "Monitoring and Controlling in Food Storage System using Wireless Sensor Networks Based on Zigbee & Bluetooth Modules", International Journal of Multidisciplinary in Cryptology and Information Security, Vol.2, no.3, 2013.