*Journal of Harbin Engineering University*
*ISSN: 1006-7043*

Vol 45 No. 11
November 2024

# Temperature Regulation Inside a Steam Power Plant Through Customised Java and WSN: A Modest Nexus

[1]**Saroj kumar Dash,** [2]**Monoj Kumar Sahu**
Gandhi Institute of Excellent Technocrats, Bhubaneswar

**Abstract**

Conventionally wired temperature sensors are used for temperature monitoring in thermal and nuclear power plants. However, recent research has shown that remote measurement techniques provide superior methods for data collection, as well as transmission of data. Wireless temperature sensors can be used in both critical and non-critical zones of the thermal and nuclear power plants. Live monitoring of temperature data from these zones is essential for the safety of the plant and the equipment. This is because the slightest change in behaviour of the system and temperature behaviour can potentially be disastrous. Live monitoring is important for safety of the plant as well as control of the day to day functioning of the plant. Commonly used configuration of Xbee in which Xbee is interfaced with microcontroller, has its own limitations and shortcomings. Data acquisition and monitoring is possible with the help of Xbee radios and Java without interfacing any microcontroller.

It has been demonstrated how thermocouples can be used in WSNs to monitor the remote temperature using Java. Java with customized API has been implemented to capture the temperature data of remotely placed nodes. No peripheral hardware is used with thermocouple while measuring the temperature.

**Keywords**: - API, WSN, Xbee module, Thermocouple, Self-repeating function

## 1. INTRODUCTION

A majority of temperature monitoring systems today make use of wired sensors. These sensors are placed in various regions of the power plant. These systems have stood the test of time however they have problems in terms of installation, scalability and maintenance. There has often been debate regarding over the use of wired or wireless systems, but wireless sensors and Wireless sensor networks reign supreme when it comes to cost, scalability and flexibility.

Wireless Sensor Networks (WSNs) are able to address and simplify a number of issues that are currently being faced by renovation and modernization projects for thermal and nuclear power plants. Highway Addressable Remote Transducers (HARTs) are currently being marketed as a solution for these projects. WSNs can be advantageous in thermal power plants. Here it has covered the entire process, right from coal handling to the electrical substation control. WSNs suffer from limitation in terms of power, which often becomes an important factor when it comes to deployment of the sensor nodes. Battery life and power dissipation are important factors that need to be clearly calculated before deploying WSNs. Sometimes WSN nodes are placed in locations where it is not possible to constantly change their battery packs. Therefore, care must be taken to make sure the battery life of the nodes can be optimized.

Thermal power plants require strict monitoring and control of parameters such as temperature, pressure, humidity throughout the various stages of power generation. They also require special chemicals that need to be stored and monitored as well. These chemicals are used in processes such as cracking and combustion in the power generation process.

WSNs with thousands of nodes become a very complex system where it is often not possible to model them analytically with high confidence. It is often not possible to deploy test beds for these tests. Software simulation offers a viable alternative to this.

A suitable model based on real world assumptions and appropriate frameworks is needed to accurately simulate the behaviour of WSNs. Also, the simulation needs to take into account real world environment, hardware. This information is usually not as accurate as is needed to get a high confidence result. Scalability and performance are the major issues with reference to WSN. A trade-off between scalability and accuracy becomes a major issue when simulating WSN. Simulation of WSNs is another field of interest that can be looked into.

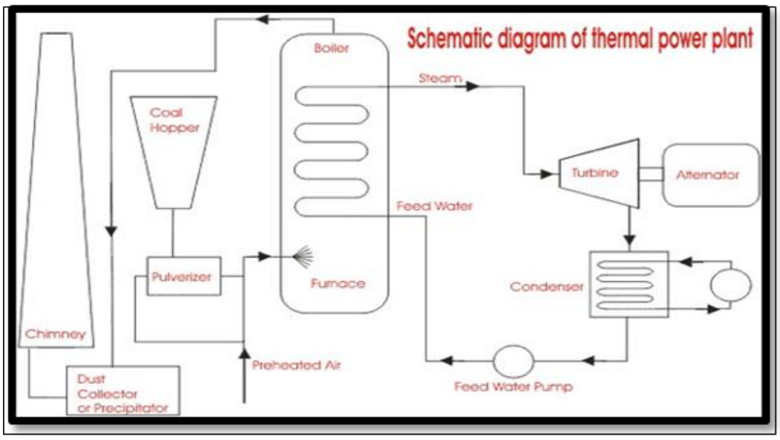


**Figure 1: Power Generation schematic diagram**

**2. System Design, Configuration and Setup**

Xbee radio modules can be configured into three different operating configurations - as a router, as a coordinator or as a repeater. A repeater is a WSN node that just transmits any data it receives to the next node in the WSN. A router is an end node of the WSN. A router can either consume the information it gets from the WSN to perform an actuating action, or it can consume sensor data from the connected sensors and transmit it to the WSN. The coordinator node in the central hub of the WSN and it is responsible for creating and maintaining the network. An Xbee WSN can have only one coordinator node. There is no limit to the number of repeater and router nodes that can be connected to an Xbee WSN. As discussed earlier, Xbee radio modules can be configured to operate in 2 operating modes
- AT and API mode.

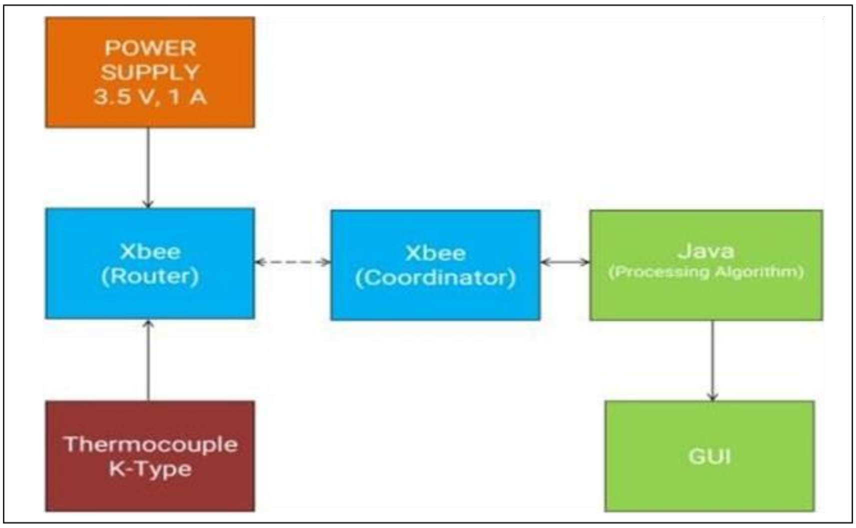


**Figure 2: Block diagram-Wireless sensing of high temperatures**

The AT mode is a basic operating mode which provides basic functionalities such as data transmission in the form of 'data packets', as described by the Xbee reference manual. The API mode has more functionality andallows more flexibility of use. Our system consists of two Xbee radios one in "AT mode" and another in "API mode", and they work as router and coordinator respectively.

The router Xbee node will be placed at a remote location. As stated, our system is modular and can work with any type of thermocouple. For our experiment though, wehave used thermocouple of type-K. The reason for choosing the type-K thermocouple is that it is one of the most used types of thermocouples. The router Xbee is powered by a 3.5V, 1A standalone power supply. The thermocouple has been connected to one of the General Purpose Input Output (GPIO) pins of the router Xbee. The GPIO pins can be configured to work as digital, analog input or output. In this system, it has been configured as an analog input pin to get the thermocouple analog output.

The coordinator Xbee is connected to PC using the USB-serial COM port. Coordinator and router Xbee nodes have been configured to allow them to setup a WSN using the XCTU software. Once the connection between the router and the coordinator nodes has been established, then the process of data transmission starts. The data from the router is sent to the coordinator node. The coordinator node then passes this data to the desktop application. The desktop application is able to perform data collection, processing, analysis, ADC conversion, visualization and logging. The working of the desktop application has been explained in the following sections.

## 3. Design and Experimental Setup

In this set up two Xbee modules (coordinator and router) have been deployed and communication between them is established using Java program. Coordinator module is connected with PC and router is deployed at remote end where LM 35 sensor is connected to sense the temperature. In this WSN, coordinator is configured in API mode of operation and router is configured in AT mode. In API-AT configuration the AT module sends the data and API module reads the same at API frame 19 to retrieve the temperature value. Even if both the radios are configured in API mode, still API frame 19 will be used by remote Xbee to transmit the data. USB-Xbee explorer connects the coordinator with PC. Java program can communicate with an Xbee module with the help of API mode of operation. Java program first fetches the data from collected frames and then converts it to temperature data and same is represented in the form of temperature-time graph and writes the data to disc. [Figure-31]

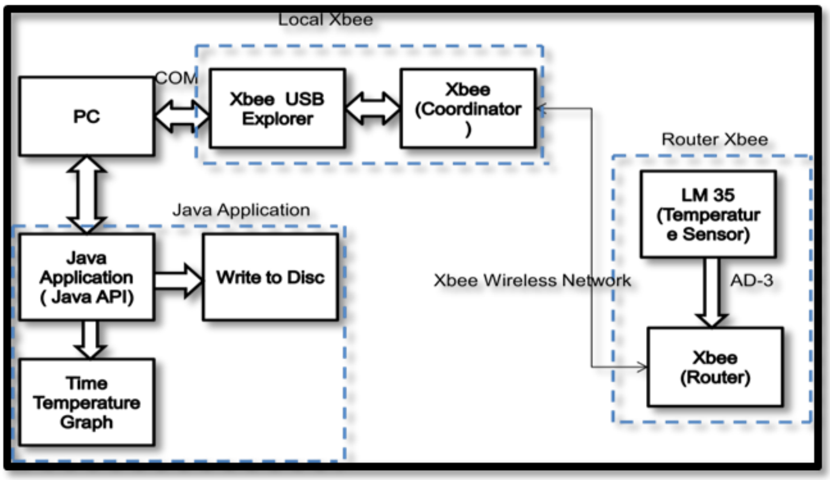


**Figure-3: Investigational setup**

## 4. Communication using Java

After establishment of communication between two Xbees i.e. coordinator and router Xbee, the sensor data is transmitted by router Xbee in the form of packets using API frames. Type of API frame used is 0x92, which makes communication possible between the two Xbee radios (AT-API configuration). API frames have been discussed in detail in Chapter 2. A single IO line on router radio is used to connect the temperature sensor LM 35. Thus total length of the API frame becomes twenty bytes. Java program extracts the data from the 19$^{th}$ byte. All the Xbees present on a network can be controlled and configured with the help of a single Xbee coordinator using Java algorithm.

Operation of Java Xbee program is depicted in figure 3. Initially, main () function is used to create Java class. Local instance is created and local device is configured. Connection between wireless network of local device and Java program is established where it searches for any remote Xbee device using "Node Identifier" parameter of router Xbee. In case remote device is not found, program exits and if it is found then Java program tries to setup the connection with remote device and subsequently configures it with the pre-defined parameters. Thermistor connected to ADC pin of Xbee senses and sends the temperature data to remote Xbee in the form of voltage. Router Xbee configured in AT mode, connected to the network sends the data to coordinator Xbee in the form of packets. Remote data is collected using a self-repeating function. A 0x92 frame is fetched from remote device, by a Java application after every given amount of time. Then an algorithm written in Java is applied to capture the analog data on previously defined I/O line (Hexadecimal value) and subsequently to get the current value of temperature.

## 5. Parsing Xbee WSN Data using the Desktop Application

This section will cover the process of data transmission and analysis between the desktop application and the Xbee nodes in detail. The sampling rate of the remotely placed node has been set using XCTU software. Thermocouple sends the data to remote node which further transmits to coordinator node after the set interval. The data is sent in the form of API frames. According to the terminology defined by Xbee, the API frame 19 with type 0x92 is used to send this data. The length of this frame is 20 bytes. The API frames parsed by the Xbee nodes can contain a wide array of information, such as the 8-bit address, 16 bit-address or MAC ID of the remote node. This information is of relevance to the coordinator to help identify and request information from the remote nodes.

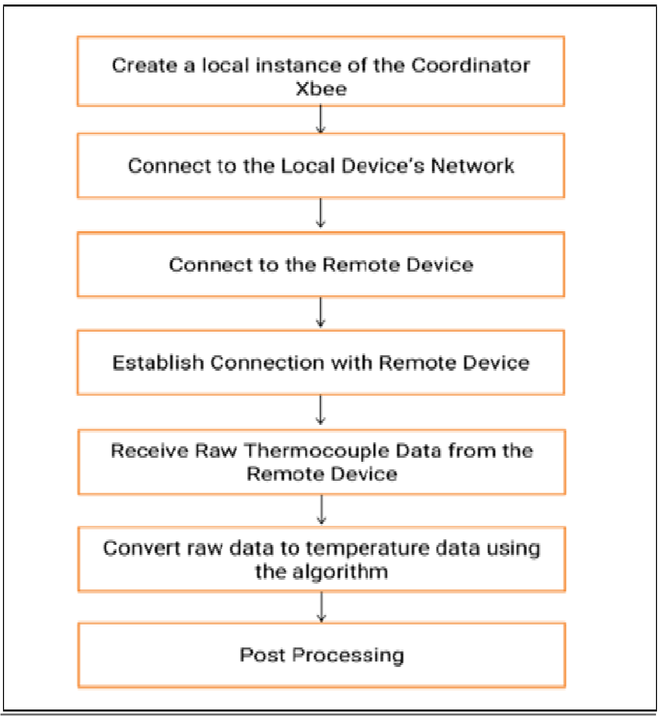


**Figure-4: Flow chart-Wireless sensing of high temperatures**

The desktop application parses this information to ensure that it is connected to the correct remote node and then reads the data present in the 19th byte. The 19th byte contains the analog data samples. The analog data samples are in the hex format and have to be converted to the corresponding integer value. The integer value then undergoes analog to digital data conversion. The integer value is converted into the corresponding voltage value. The voltage value can then be used to get the "cold junction compensated voltage". This voltage is then converted into the corresponding temperature value. The process of conversion of the voltage to temperature has been explained in the following section.

## 6. WSN Process Flow

The underneath figure-5 explains the process flow regarding how the system converts analog sensor data from the thermocouple, connected to the remote node, into thermocouple temperature, visualization, logging and monitoring, sequentially.

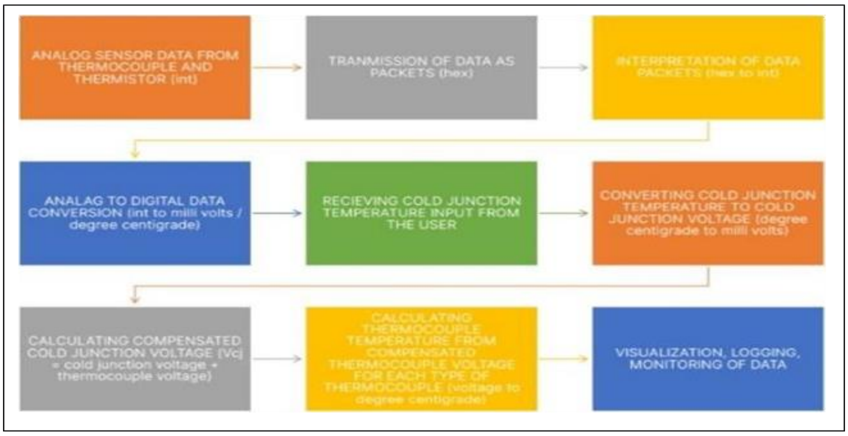


**Figure-5: WSN Process flow**

## 7. Thermocouple Data Processing using Java

Once cold junction voltage is calculated, it is added to thermocouple voltage and then cold junction compensated thermocouple (K-type) temperature is calculated using equation no. 7.2, and same is displayed on the screen. All the corresponding coefficients are taken from the NIST table [70]. The equations are universal for all types of thermocouples, but the coefficients and the voltage range differ per thermocouple type. The WSN detects and connects to the remote node without requiring any input from the user. The data transmission is setup automatically at the predefined sampling rate. Screen shot in figure-6 shows the sensor voltage at the remote node, and theconverted thermocouple temperature in real time. The desktop application is able to perform ADC, cold junction compensation and conversion of the compensated voltage into the corresponding temperature.

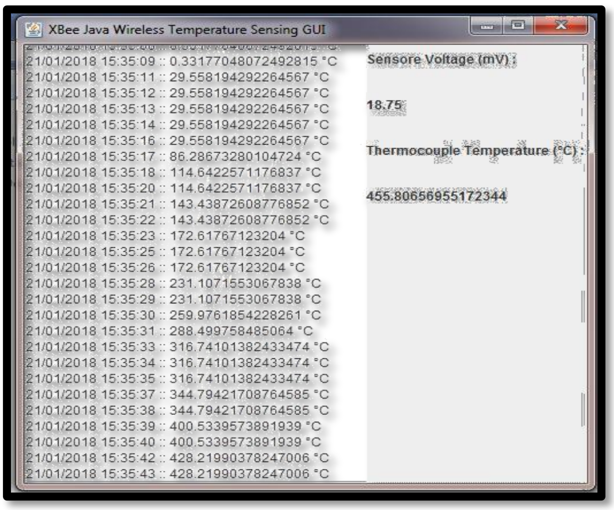


**Figure-6: Screenshot-Sensor voltage and thermocouple temperature in real time**

## 8. Simulation of the Various Thermocouple Equations to Demonstrate thePotential of the

**Proposed System**

The equations and coefficients used to calculate the thermocouple temperature from the thermos-voltages have been released for public use by the US "NIST" with adherence to the International Temperature Scale of 1990 (ITS-90).

The system is able to fetch analog thermocouple data from a wireless node. Subsequently, the software is able to perform analog to digital data conversion, digital signal processing, and cold junction voltage calculation and calculate the cold junction compensated thermocouple temperature from the thermos-voltage automatically. This can be repeated for any type and number of thermocouples ("Type-B, Type-E, Type-K" thermocouple etc.).

Earlier such a complicated sequence of tasks could only be completed through a diverse array of electronic circuits. Such a "hardware" approach, is inflexible, and isnot easy to customize or adapt to different operating requirements and applications. This work is immune to the wide and diverse array of errors such asvoltage drift, ageing and performance variations due to change in ambient temperatures. These problems are inherent to electronic circuits and solutions based on electronic circuits. Our proposed system is able to overcome all of these drawbacks, while at the same time, allows for streamlining of data collection, data processing and data logging for a number of sensors, wireless nodes and locations simultaneously. The system is also able to visualize the thermocouple temperature in the form of a time-thermocouple graph in real time. Such a graph cannot be generated whilst using only electronic circuits. Data visualization helps the user of the system, as well as the designer of the system and process better understand the nature of the system itself.

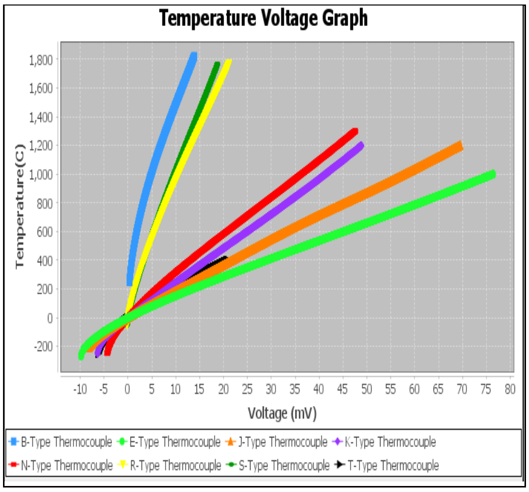


**Figure-7: Voltage-temperature graphs for various thermocouples**

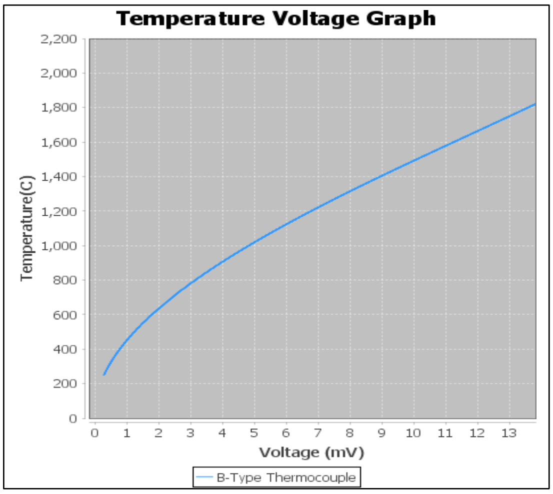


**Figure-8: Voltage-temperature graph of the B- type thermocouple**

**Conclusion**

Java has been interfaced with Xbee radios to capture and process the live data. Communication in AT and API mode has been established between remote and local Xbee respectively. Data acquisition and monitoring is made possible with the help of Xbee radios and Java without interfacing any microcontroller. Performance of these filters is not affected due to heating and drift. Thermocouples have been used in WSNs to monitor the remote temperature using Java. Java with customized API has been implemented to capture the temperature data of remotely placed nodes. No peripheral hardware is used with thermocouple while measuring the temperature.

**Future Scope**

Understanding and analysis of data in WSN will become easier while using the statistical analysis. Wireless monitoring and predictive system is developed to measure and predict the remotely placed high temperature regions. Real time

Statistical analysis on live data could be conveniently operative over these kind of executions.

**REFERENCES**

[1] V. S. Kale, and R. D. Kulkarni, “Real Time Remote Temerature & Humidity Monitoring Using Arduino and Xbee S2,” International Journal of Innovative Research in Electrical,Electronics, Instrumentation and Control Engineering, vol. 4, no. 6, pp. 175-179, June 2016.
[2] P. Kulakowski, “WSNs: Technology, Protocols and Applications,” [Book Review], IEEE Communications Magazine, vol. 46, no. 6, pp. 42-44,2008.
[3] P. Le-Huy, and S. Roy, “Low-Power Wake-Up Radio for WSNs,”Mob. Netw. Appl., vol. 15, no. 2, pp. 226-236, April 2010.
[4] J.M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, “A secure and resilient WSN roadside architecture for intelligent transport systems,” Proceedings of the first ACM conference on Wireless Network Security, Alexandria, VA, USA, March 2008.
[5] R. Nowak, and U. Mitra, “Boundary Estimation in Sensor Networks: Theory and Methods,” Proceedings of the 2nd Workshop on Information Processing in Sensor Networks (IPSN’03), Palo Alto, CA, Apr. 2003.
[6] A. Cerpa, and D. Estrin, “ASCENT: Adaptive Self-Configuring Sensor Networks Topologies,” Proceedings of the 21stAnnual Joint Conference of the IEEE Computer and Communications Societies (InfoCom’02), New York, vol. 3, June 2002.
[7] K. Sohraby et al., “Protocols for Self-Organization of a WSN,” IEEE Personal Communications, vol. 7, no. 5, Oct. 2000.
[8] H. Gupta et al., “Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution,” Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc’03), Annapolis, MD, June 2003.