Distributed Temperature Monitoring Network with Low Power Consumption and Severe Environment Resistant

Yichen Luo
Xi'an Jiaotong-Liverpool University
Suzhou City, China, Yichen.Luo18@student.xjtlu.edu.cn

Wenhao Lin
Xi'an Jiaotong-Liverpool University
Suzhou City, China, Wenhao.Lin20@student.xjtlu.edu.cn

Xiaohui Zhu*
Xi'an Jiaotong-Liverpool University
Suzhou City, China, Xiaohui.Zhu@xjtlu.edu.cn

Yong Yue
Xi'an Jiaotong-Liverpool University
Suzhou City, China, Yong.Yue@xjtlu.edu.cn

ABSTRACT

This paper proposes a distributed temperature monitoring network using ZigBee and 4G/5G protocols. The network consists of temperature monitoring terminals, coordinators, and a remote server. The temperature monitoring terminals are distributed in the target environment and collect temperature data, which is sent to coordinators and then forwarded to the server. The server stores the data in a database. Compared with other wireless sensor solutions, we optimizes the terminal design for waterproof, high temperature resistance and lower power consumption, so that it has better environmental adaptability and longer working time. Experimental results show that by using a 1000mAh battery, the terminal can continue to operate steadily in severe environment with high humidity and temperature around 80℃ for 20 days until the battery runs out.

CCS CONCEPTS

• **Hardware → Sensor applications and deployments**; *PCB design and layout*.

KEYWORDS

ZigBee, IoT, Distributed sensor, Temperature monitoring, Low power consumption

ACM Reference Format:
Yichen Luo, Wenhao Lin, Xiaohui Zhu, and Yong Yue. 2023. Distributed Temperature Monitoring Network with Low Power Consumption and Severe Environment Resistant. In *ACM International Conference Proceedings Series.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3580219.3580251

1 INTRODUCTION

In logistics warehousing, transportation, manufacturing, agriculture and other industries, it is usually necessary to accurately monitor the temperature of the production environment in real-time, so that the control equipment can adjust the temperature according to the demand. In such application scenarios, a single sensor is limited to the detection range, and it is usually difficult to fully monitor the temperature everywhere in the environment. In order to meet the acquisition requirements, a distributed multi-sensor network is usually applied. In addition, some application scenarios may also contain extreme environments such as high/low temperature, high humidity or high dust. Therefore, the temperature monitoring system also needs to have a good protection ability.

Existing distributed temperature collection solutions using wired distributed sensors are complex and inconvenient, and the use of high temperature and humidity resistant wires increases cost. Additionally, it is difficult to add or delete nodes in a wired sensor network, limiting their applicability.

Wireless sensor networks with battery power offer better flexibility and economy. However, the farthest communication distance and power consumption of the wireless protocol must be considered. For the given application scenario, the distance between sensors in the same temperature monitoring network is typically no more than 200 meters. Within this range, available wireless communication protocols include WiFi, Bluetooth Low Energy (BLE), ZigBee, and others.

As we know, the power consumption of WiFi is much higher than that of BLE and ZigBee. Therefore, IoT devices with low power consumption requirements usually adopt the latter two protocols for communication. For instance, a temperature sensor scheme based on BLE 4.0 described in an invention patent can achieve continuous operation for several years with a large-capacity lithium battery power supply [4]. This system uses BLE for inter-sensor networking and transmits data to a remote server through a BLE-to-4G module.

Despite the low power performance similar to ZigBee, the technical limitations of BLE in networking affect its application range. Bluetooth versions prior to 5.0 do not support mesh networks, so the maximum device number in the same network is less than 7. Therefore, for some application scenarios that require a large number of sensors and have complex networking, BLE is not appropriate. Bluetooth 5.1, released in 2019, addresses these issues [5], but its associated hardware solutions are few and expensive due to the low technology maturity.

For ZigBee, its related software and hardware schemes have been developed for many years and tend to be mature. Among them, the 2007-pro version protocol released by the ZigBee Alliance in October 2007 provides a complete technical specification. This version of the protocol theoretically supports more than 1000 nodes for mesh networking, which can realize dynamic routing transmission and network self-healing [11]. In addition, its power management

CCEAI 2023, Jan. 28-30, 2023, Hainan

ACM ISBN 978-1-4503-9751-3/23/01…$15.00
https://doi.org/10.1145/3580219.3580251

mechanism is also excellent. At present, many semiconductor manufacturers, such as Texas Instruments (TI) and Silicon Laboratories, have developed reliable and low-cost solutions on this basis.

In summary, the temperature monitoring network described in this paper use ZigBee as the communication scheme between temperature sensors. In order to store the data into the remote server, it is also necessary to develop a gateway device that can forward the data form ZigBee network to cloud server through 4G/5G.

In this paper, section 2 shows the system architecture of the temperature monitoring network. Section 3 to 5 details the design scheme of the temperature monitoring terminal, coordinator and server program, respectively. Section 6 shows the test results of the system, including functional verification and terminal power consumption test.

2 SYSTEM ARCHITECTURE

The distributed temperature monitoring system consists of four parts: temperature monitoring terminals, coordinators, Remote Terminal Units (RTUs), and cloud servers. The temperature monitoring terminals (i.e., the end-devices in the ZigBee protocol) are deployed in environments where temperature needs to be collected periodically. As the core nodes of the sensor network, coordinators receive temperature data from terminals and send it to RTUs, which then send the data to a remote server in the cloud via 4G/5G. The server stores the data in a database for access by front-end applications.

The ZigBee protocol is applied for the communication between the temperature monitoring terminal and the coordinator. RTU devices generally use RS-485 protocol to communicate with peripherals. Therefore, the corresponding RS-485 interface is designed in the coordinator to send the temperature data to the RTU. The system architecture is shown in Figure 1.

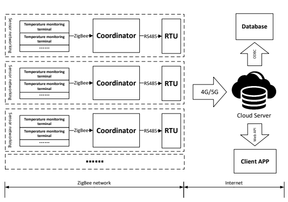


Figure 1: Temperature monitoring network architecture

ZigBee network can be realized by star topology or mesh topology. Considering that the communication distance of most application scenarios is generally no more than 200 meters, the ZigBee protocol based on IEEE 802.15.4 standard can generally achieve the direct communication between two devices within this range [1]. Therefore, the star topology is used to reduce the complexity of the system. Only two device types, end-device and coordinator, exist in the ZigBee network. In addition, if the communication distance between devices is greater than 50 meters and occlusions are included midway, the system can also adopt a mesh topology. That is, routers are added to the network to provide relays for end-devices and coordinators.

Multiple sensor networks can be accommodated simultaneously in this temperature monitoring system. In order to prevent end devices from erroneously connecting to coordinators in other sensor networks, a unique Personal Area Network Identity (PAN-ID) is configured for each ZigBee network. A terminal will only attempt to connect to coordinators with the same PAN-ID [11].

In addition, each ZigBee terminal and coordinator has a unique ID that the server uses to determine which network it belongs to. The coordinator ID is 1 byte long and can represent up to 256 devices, while the terminal ID is 2 bytes long and can represent up to 65536 devices. The server program records the ID values of all equipment in a database and associates them with each ZigBee network.

3 TEMPERATURE MONITORING TERMINAL

3.1 Functional Requirements

The temperature monitoring terminals are distributed in the environment where the temperature needs to be collected. It is responsible for periodically collecting the temperature and transmitting it to the coordinator through the ZigBee network. Limited to some special application scenarios (such as logistics vehicles), the device also needs battery power. Therefore, it must meet the requirements of long endurance time, high/low temperature resistance, and waterproof.

3.2 Element Selection and Electronic Circuit Design

First, the device requires a Micro Controller Unit (MCU) for ZigBee network communication, battery monitoring, and other functions. Considering the requirements of low power consumption, long communication distance, high stability of the device, and ease of use, we uses the CC2530-F256 MCU produced by TI [10]. In order to simplify the terminal hardware design, the digital temperature sensor DS18B20 produced by Maxim Integrated (Maxim) was selected, which does not require the use of Digital-to-Analog Converter (DAC) [9].

In terms of power supply, considering the waterproof requirements of the device, rechargeable lithium battery with dimensions $6 \times 20 \times 80\ mm$ is adopted. The nominal capacity of the battery is $1000mAh$, power supply voltage is between $3.2V$ and $4.2V$. To avoid overcharging and overdischarging the battery, the circuit uses a DW01A lithium battery protection Integrated Circuit (IC) for power management. This IC automatically shuts off the circuit when the battery is charged above $4.2V$ or discharged below $3.2V$, protecting the battery and preventing potential hazards such as explosions. In addition, AMS-1117 3.3V Low Dropout (LDO) regulator produced by Advanced Mono lithic Systems (AMS) is used to reduce the voltage to $3.3V$ and filter circuit ripple to provide a stable working environment for MCU [3].

Based on the above hardware equipment, the circuit of the temperature monitoring terminal is shown in Figure 2.

The schematic diagram in Figure 2 can be divided into five parts: a MCU minimum system module, a temperature detection module, a lithium battery charge and discharge module, a power voltage holding module, and an LDO module. When working with embedded software, this circuit can perform periodic temperature collection, network communication, battery power detection, ZigBee network connection status detection, and power voltage holding.

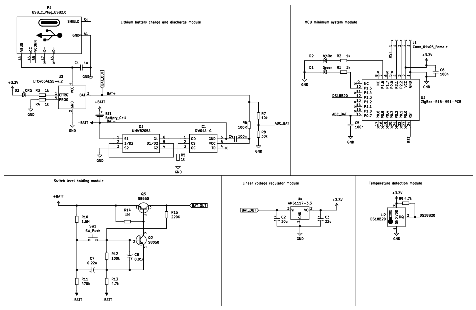


Figure 2: Schematic diagram of temperature monitoring terminal

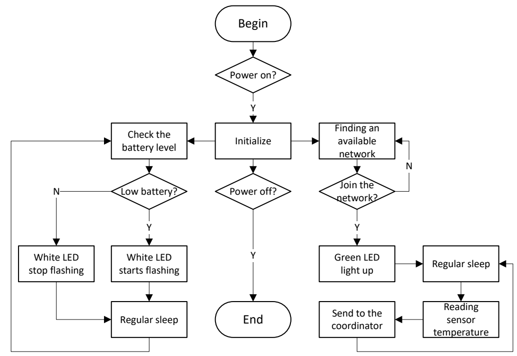


Figure 3: Terminal device workflow

In this circuit, we use a white LED to indicate the working state of the device and the battery power level. The LED lights up when the device is working, and flashes when the battery is low.

The lithium battery charging part uses a dual N-Metal-Oxide Semiconductor (NMOS) to control the charge and discharge switch of DW01A and connects to a Type-C interface for power charging. A yellow LED indicates the battery charging state, lighting up when charging and turning off when the battery is fully charged.

In addition, to reduce the volume of the temperature monitoring terminal and enhance its waterproof ability, We adopted the micro-push button as the power switch. Therefore, we designed a power voltage holding circuit, which can continuously output high voltage after the user press and releases the button to keep the equipment working.

For the power voltage holding module, we use two Bipolar Junction Transistors (BJTs) to form an interlock. As shown in Figure 2, when button SW_1 is pressed, capacitor C_7 discharges and forms a loop with resistor R_{12}, causing Q_2 to become saturated, which drives transistor Q_3 to saturate, and the load circuit turns on. The on-circuit, in turn, acts on the base of transistor Q_2 through resistor R_{15} so that Q_2 and Q_3 remain saturated after the button is released. Conversely, when SW_1 is pressed again, capacitor C_7 discharges in the opposite direction, causing Q_2 and Q_3 to be cut off, and the load circuit to be shut down.

3.3 Embedded Program Design

To realize ZigBee communication, the Z-Stack protocol stack developed by TI is used to control the embedded system based on CC2530 MCU [2]. The embedded program is primarily responsible for controlling the temperature acquisition cycle of the sensor, realizing ZigBee network communication, detecting battery power, etc. According to the polling mechanism of Operating System Abstraction Layer (OSAL), the embedded program first generates two tasks after the device is turned on, including power supply voltage detection and network access detection.

The flow chart of the program is shown in Figure 3.

For the power supply voltage detection task, the MCU reads the voltage data from the DAC every 1 minute after startup. It compares it with the preset threshold value in the program. Based on the comparison results, the program controls the state of the corresponding LED to show the battery level. The threshold setting depends on the size of the divider resistance.

The power supply voltage can be calculated as follows:

$$V_a = \frac{R_7 + R_8}{R_7} V_t \tag{1}$$

Where resistors R_7 and R_8 are divider resistors, which can be found in Figure 2. V_t is the measured voltage of the DAC module, and V_a is the actual supply voltage.

For the network access detection task, the program detects whether there is a coordinator with the same PAN-ID near the terminal by analyzing the signal quality based on the preset detection interval. When a legitimate coordinator is detected, the program proactively requests a communication connection with the coordinator. After successfully connecting to the coordinator, the program will start the temperature collection subtask; otherwise, it continues to detect the presence of other available coordinators.

When the device is connected to the coordinator, the program will turn on the green LED (i.e., the LED that indicate network connection status); When disconnected from the coordinator, the program will turn off the LED.

For the temperature collection subtask, the program will periodically read temperature data form DS18B20 based on the preset time, and send it to the coordinator. The temperature collection period can be flexibly adjusted according to the application requirements.

4 COORDINATOR

4.1 Functional Requirements

Coordinators are deployed in the center of the ZigBee network and responsible for the construction and scheduling of the sensor network. Its function is to receive the temperature data sent by the wireless monitoring terminal and forward it to the RTU via RS-485 protocol. Therefore, the coordinator needs to have ZigBee network reception and management, RS-485 bus output, LED screen display, and other functions.

4.2 Element Selection and Electronic Circuit Design

For coordinator, we still choose CC2530-F256 as the MCU, which is responsible for implementing ZigBee network communication/management and temperature data transmission. The MAX13487 chip of Maxim is used as the RS-485 transceiver for the connection between coordinator and RTU [8]. This chip has the advantages of automatic control of transceiver direction, strong anti Electromagnetic Interference (EMI) ability and high maximum communication rate.

In order to show the temperature data to the user in real-time, we use a 0.96-inch screen to display the temperature, which is controlled by SSD-1306 chip via Serial Peripheral Interface (SPI). In terms of power management, the B0505S-1W DC-DC module is used for isolation to protect the internal circuit and prevent external power supply interference, such as surges and ripples. In addition, AMS1117-3.3V LDO is also used to convert the power supply from $5V$ to $3.3V$ to power MCU and LCD screen.

The coordinator's hardware circuit is designed as Figure 4 shows. It includes four parts: voltage regulator module, MCU minimum system module, Liquid Crystal Display (LCD) module, and RS-485 communication module.

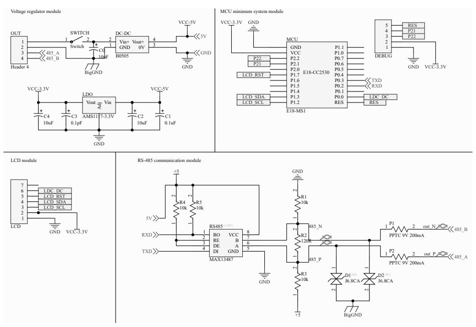


Figure 4: Schematic diagram of coordinator

In the design of the minimum system part of the MCU, we connected the necessary GPIO to communicate with the peripherals, and reserved a program download interface for it.

Transient suppression diodes and self-recovery fuses are used as protection elements on the RS-485 bus to ensure the safety of the internal circuit under high overvoltage/overcurrent conditions such as lightning strikes.

4.3 Embedded Program Design

The embedded program of the coordinator is developed based on Z-Stack, and the network management and message receiving are implemented by its bottom layer. The coordinator program flow chart is shown in Figure 5.

ZigBee protocol with version 2.5.1a does not realize terminal devices' network access and disconnection detection. Therefore, a heartbeat mechanism is designed in the application layer to detect the network access of terminal devices. Since the terminal always sends the temperature regularly, the temperature data is directly used as the content of the heartbeat packet to simplify the program design.

For data transmission, when the coordinator receives the temperature data sent by a terminal, it immediately packages the data into a message with a particular formation, and sends it out through the UART port of MCU. The information is then converted to RS-485 via MAX13487 and sent to the RTU.

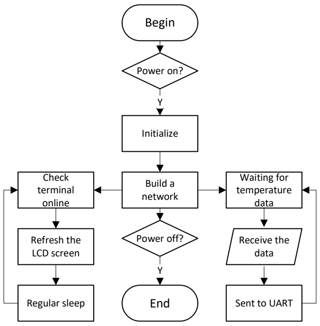


Figure 5: Coordinator workflow

LCD displays is divided into four rows to display text. The first line is the number of terminals currently connected to the network. The second to fourth lines are the scrolling display of the temperature data of each terminal, refreshed every three seconds.

5 SERVER PROGRAM

5.1 Program Architecture

The server program runs on a cloud server to collect temperature data from remote RTUs and save it in a MySQL database. It needs high bandwidth, strong concurrent processing capabilities, and stable communication. To handle potential errors during transmission, the program's communication design must be fault-tolerant, such as adding parity bits to verify data integrity.

The program is divided into three parts, including the TCP/IP communication management, the communication message parsing, and the database operation. The overall architecture design of the server is shown in Figure 6.

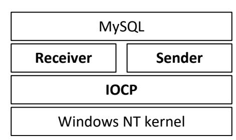


Figure 6: Server architecture

For the TCP/IP communication part, to improve the program's concurrency, Input/Output Complete Port (IOCP) mode is adopted as its communication management model. Unlike traditional high-concurrency network communication programs, IOCP creates threads based on the number of server host CPU cores. Each thread is bound to a core. This mechanism avoids the time-consuming context switching operations of the Windows NT kernel when scheduling threads [7]. Therefore, an IOCP thread can be bound to multiple network ports. When IO events are processed, the thread pulls the events from the queue and executes them in sequence [6].

5.2 Communication Design

The TCP/IP protocol stack provides reliable byte stream transport services at the transport layer, but does not handle packet processing. In actual transmission, packet breaking and packet sticking may occur, where two independent packets are parsed together at the receiving end or one packet is split into two parts at the receiving end.

In addition, as RS-485-based serial communication may have transmission errors such as frame loss or bit reversal, a verification mechanism is needed to find and correct them.

Therefore, to ensure the accuracy and stability of the transmitted data, we designed a lightweight packet encapsulation protocol for the communication between the coordinator and the remote server. The function of each part of the packet is as follows:

- Start byte: 1 byte. Hexadecimal constant 0xFF represents the start of the packet.
- Device type: 1 byte. Used to identify the role of the device on the network. The server is represented by the hexadecimal constant 0x00, and the coordinator is represented by the hexadecimal constant 0x01.
- Message type: 1 byte. Used to determine the received message type. Where, the coordinator temperature message is represented by the hexadecimal constant 0x00, and the response message from the server is represented by the hexadecimal constant 0x80.
- Index: 1 byte. Used to determine whether there is packet loss during transmission. The value is automatically increased once each packet is sent. If the receiver finds that the index of a packet is not continuous with the previous received packet, it proves that there is a packet loss during transmission.
- Body length: 1 byte. Used to specifies the length of the package.
- Body: 1-256 byte. Responsible for transmitting the main content of the message, depending on the requirements.
- Check byte: 1 byte. Uses CRC-8 check algorithm to ensure the correctness of the packet content.

6 SYSTEM TEST AND RESULT ANALYSIS

6.1 Functional Verification

As shown in Figure 7, the coordinator works properly and supports other wireless monitoring terminals to access the network. A terminal with ID 01 is now connected, and the temperature of this terminal is 21.6℃.

Figure 7: Terminal and coordinator prototype

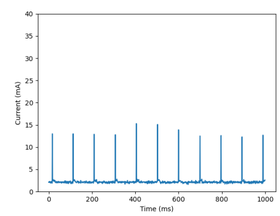
We tested the working condition of the terminal in an environment of 80℃ and 100% humidity. Experimental results show that the terminal can continuously work and collect environmental temperature data. After connecting the coordinator to the RTU device and configuring the IP address and port number, the coordinator will send the temperature data to the server. Experimental results show that the server program works properly.

In addition, we analyze the measurement error of the sensor by obtaining the Mean Square Error (MSE) of the measured data between 10 sensors placed in the same environment and taking measurements at the same time. The experimental results demonstrate that the MSE is small, indicating good measurement accuracy of the sensors.

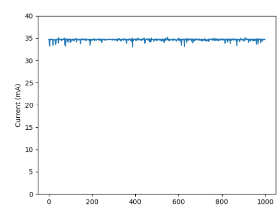
6.2 Power Consumption of Terminal Device

Under the condition of $3.3V$ voltage supply, the power consumption of a device can be obtained by measuring the load current of it. We use an INA-219 digital ammeter chip produced by TI to measure the load current when the terminal is connected and disconnected from the coordinator and send it to the computer for analysis. Among them, after the terminal connects with the coordinator, it sends the temperature data to the coordinator at a time interval of $0.1s$.

After measurement, the average load current of the terminal is $2.13mA$ when connected to the coordinator, and $34.66mA$ when not connected, as shown in Figure 8.



(a) Connect



(b) Unconnect

Figure 8: Current curve of terminal

It can be seen that the terminal's power consumption is higher when searching for a network than after connecting to a coordinator. This is because when not connected, the terminal continuously searches for a nearby coordinator, causing the RF module to work continuously.

The capacity of the battery we choose is $1000mAh$. According to the following formula:

$$C = It \tag{2}$$

Where I is the load current, t is the power-on time, and C is the capacitance of battery. The endurance time of the terminal when connected to the coordinator can be calculated as:

$$t = \frac{1000}{2.13} = 469.48 \ \ (hours) \approx 20 \ \ (days) \tag{3}$$

Therefore, the terminal can work sustainably for about 3 weeks if it remains connected to the coordinator. In practical applications, if the terminal temperature data transmission interval is increased to more than 5*s*, the terminal power consumption can be further reduced.

7 CONCLUSION

This paper proposes a wireless temperature acquisition/monitoring scheme using ZigBee and 4G/5G protocols. ZigBee is used for inter-sensor communication and networking, and data is centralized by the coordinator and sent to a remote server via 4G/5G. A server program has been developed for data storage and access. The system includes embedded development, network communication development, hardware design, and communication data processing. It addresses some shortcomings of traditional sensors, such as deployment inconvenience and low endurance time, and has high temperature resistance and waterproof ability.

In the future, we will continue to expand in many areas. This will include the development of a cross-platform client using Web API, integration into the temperature control system in the application environment, and further improvement of the high temperature resistance and waterproof performance of the equipment.

ACKNOWLEDGMENTS

This research was supported by the Suzhou Science and Technology Project-Key Industrial Technology Innovation (Grant No. SYG202122), the Key Program Special Fund of Xi'an Jiaotong-Liverpool University (XJTLU) (Grant No. KSF-A-19, KSF-E-65, KSF-P-02, KSF-E-54) and the Research Development Fund of XJTLU (RDF-19-02-23).

REFERENCES

[1] 2020. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)* (2020), 1–800. https://doi.org/10.1109/IEEESTD.2020.9144691

[2] Texas Instruments 2020. *Z-Stack 5.20.00 User's Guide.* Texas Instruments, Texas. https://dev.ti.com/tirex/explore/node?node=ANbR0LtTwkMnDCiEhygF7A__pTTHBmu__LATEST

[3] Advanced Monolithic Systems [n. d.]. *AMS1117 1A Low Dropout Voltage Regulator.* Advanced Monolithic Systems. http://www.advanced-monolithic.com/pdf/ds1117.pdf

[4] Wu Anjun, Zhang Guanzhong, Hu Jie, Xu Chunchao, and Yan Li. 2021. A cold storage intelligent control device based on Bluetooth BLE sensor network.

[5] Bluetooth Core Specification Working Group 2019. *Bluetooth Core Specification* (version 1.1 ed.). Bluetooth Core Specification Working Group. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726

[6] Shu Heng. 2015. The Network Model Based on IOCP Memory Control Key Technical Analysis. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES).* 236–239. https://doi.org/10.1109/DCABES.2015.66

[7] M. H. Lucovsky, J. D. Vert, D. N. Cutler, D. E. Havens, and S. R. Wood. 2001. Input/output completion port queue data structures and methods for using same.

[8] Maxim Integrated 2015. *MAX13487E/MAX13488E Half-Duplex RS-485-/RS-422-Compatible Transceiver with AutoDirection Control* (rev. 1 ed.). Maxim Integrated. https://datasheets.maximintegrated.com/en/ds/MAX13487E-MAX13488E.pdf

[9] Maxim Integrated 2019. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer* (rev. 6 ed.). Maxim Integrated, California. https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

[10] Texas Instruments 2011. *A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications* (rev. 2 ed.). Texas Instruments, Texas. https://www.ti.com/lit/ds/symlink/cc2530.pdf?ts=1634268432273&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253Dcc2530%2526nr%253D8073

[11] ZigBee Alliance Board of Directors 2017. *ZigBee Specification* (rev. 22 ed.). ZigBee Alliance Board of Directors. https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf