HiLL PUBLISHING

Advances in Computer and Communications, 2023, 4(5), 328-333
https://www.hillpublisher.com/journals/acc/
ISSN Online: 2767-2875

Design and Implementation of Communication Scheme Between Microcontroller and Android Application Based on RS232 Technology

Wenkai Sun*, Weisheng Liu, Qiyue Gao

School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China.

How to cite this paper: Wenkai Sun, Weisheng Liu, Qiyue Gao. (2023) Design and Implementation of Communication Scheme Between Microcontroller and Android Application Based on RS232 Technology. *Advances in Computer and Communication*, 4(5), 328-333.
DOI: 10.26855/acc.2023.10.012

Received: September 28, 2023
Accepted: October 26, 2023
Published: November 22, 2023

***Corresponding author:** Wenkai Sun, School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China.

Abstract

In the development of applications that combine software and hardware, communication problems between devices are often encountered. For example, in an industrial control system, the microcontroller acts as the lower computer and needs to receive control commands from the upper computer. This requires communication between the upper and lower computers to transmit the commands effectively. In this paper, we design and implement communication between the microcontroller module (lower unit) and the Android host (upper unit) using RS232 technology in the context of the "intelligent extermination robot" project. The article first introduces the relevant technology used, and then explains how to establish communication between the MCU main control board and the Android terminal host. It also covers the design and analysis of the communication messages, providing valuable references for those with similar needs.

Keywords

RS232 technology, Microcontroller, Android system, serial communication

Introduction

In early 2020, a new coronavirus swept through the country with great speed, reigniting public health and safety concerns that had been dormant for a long time [1]. The end of the epidemic still requires maintaining safety precautions, especially for public environments. Many places are currently relying on manual disinfection. However, manual disinfection poses long-term risks to the health of disinfection personnel due to prolonged exposure to viruses in the environment. Additionally, continuous manual operation can lead to fatigue and reduced work efficiency. Based on this, an "Intelligent Elimination Robot" has been developed to replace manual disinfection work in various environments, such as airports, railway stations, and high-speed railways. This robot can efficiently complete the task of environmental disinfection. The communication component of the intelligent elimination robot consists of three parts. The communication component of the intelligent extermination robot comprises three modules: the microcontroller main control board module, the Android host module, and the autonomous mobile chassis module. The communication requirements are illustrated in Figure 1. The MCU main control board module is responsible for controlling various hardware operations such as the UV disinfection switch, the left and right arm swing, fan switching, and spray switching. The Android host module serves as the core control module of the system, facilitating communication between the MCU main control board module and the autonomous mobile chassis module. It is responsible for debugging the corresponding parameters, setting up and controlling the movement of the autonomous mobile chassis, and enabling the system to perform automatic extermination functions for households. In this paper, we will provide a detailed introduction to the implementation of communication between the MCU main control board module and the Android host module. We will also explain the design and parsing of communication messages.

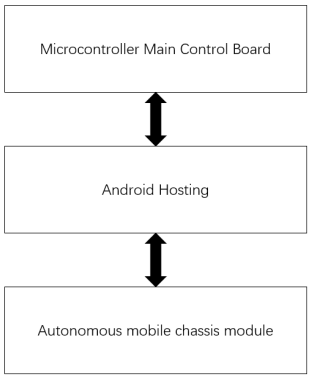

Figure 1. Schematic diagram of the communication part.

1. Introduction to related technologies

1.1 Serial communications

The serial port is the main interface for communication between the display and control equipment and the signal processing board. It is also an important interface for the transmission of protocol data frame communication between the display and control equipment and other equipment and between equipment and equipment [2]. Serial communication means that the serial port sends and receives bytes bit by bit. Although bit-byte (byte) serial communication is slow, the serial port can use one wire to send data while simultaneously receiving data on another wire. Serial communication protocol refers to the specification of the data packet's content, which includes the start bit, body data, parity bit, and stop bit. Both sides must agree on the consistent format of the data packet to properly send and receive data. Serial communication protocol is based on the serial port, allowing communication between two devices to follow a specific agreement. This agreement defines the data frame format and transmission mode that both sides must adhere to [2]. In serial communication, the commonly used protocols include RS-232, RS-422, and RS-485.

1.2 RS232 communication technology

RS232 standard interface (also known as EIA RS-232) is one of the commonly used serial communication interface standards. It was developed in 1970 by the U.S. Electronic Industry Association (EIA) in collaboration with the Bell System, modem manufacturers, and computer terminal manufacturers. Its full name is "Technical Standard for Serial Binary Data Exchange Interface between Data Terminal Equipment (DTE) and Data Communication Equipment (DCE)" [3].

RS232 uses ASCII characters for data transmission. The communication technology is mainly composed of three parts: ground, transmitter and receiver, which can be combined together to realize the complete function of data transmission and reception. With the continuous development of communication technology, most of the devices can be connected together using certain communication standards, for microcontrollers and PC host computer is no exception, it can use the role of RS232 communication standards to realize the data connection [4]. RS232 has the advantages of fewer signal lines, flexible baud rate selection, longer transmission distance and so on.

1.3 Android

Android is an open-source, Linux-based operating system for mobile devices, such as smartphones and tablets. Android was developed by the Open Handset Alliance, led by Google and other companies. Android provides a unified approach to application development, meaning that developers only need to develop for Android in order for their applications to run on various Android-powered mobile devices. Android provides a unified approach to application development, meaning that developers only need to develop for Android in order for their applications to run on various Android-powered mobile devices [5].

2. Communication scheme design and realization

In the project of the "Intelligent Extermination Robot", the Android host utilizes the Android main control board

with RS232 and RS485 interfaces. This choice is made because the physical distance between the microcontroller module and the Android host is only 50 cm, the amount of communication data is small, and there is no need for networking functionality. To realize full-duplex communication, it is necessary to use two physical wires. This method offers simple and convenient manipulation, as well as easy debugging. It also has a unique advantage for transmitting a small number of commands. Therefore, choosing the RS232 technology standard as the data transmission method is an efficient and convenient way to meet the project requirements.

2.1 Communication message design

To establish communication between the microcontroller board and the Android terminal host, it is essential to establish a common message format. This will facilitate the parsing of data sent by each party in accordance with the agreed format. In this design, the microcontroller side uses a structure to encapsulate the message format, while the Android side uses a class to encapsulate the message format. When designing the message format, consideration is given to the generality of the message and the high efficiency of transmission. The communication data format consists of four parts: the start data, the function domain, the data domain, and the error detection, as shown in Table 1.

Table 1. Communication message format

Starting data	Functional domain	Data domain		Error detection
Head (protocol header)	Command (function)	DataLen (length)	Data (dataframe)	CheckSum (Checksum)
5Byte	1Byte	1Byte	xByte	1Byte
"$J636"	command code	Number of data frame bytes x	data frame	Binary inverse summation

1. starting data (Head): protocol header, accounting for 5 bytes, the data is fixed data $ J636.

2. function domain (Command): command code, accounting for 1 byte, the data range of 1 ~ 255, part of the function as shown in Table 2, the host in the process of the command description refers to the Android interface display board (Android side), the slave refers to the robot control motherboard (microcontroller side).

Table 2. Command design

Command value	Functional Description
0x01	Device connection handshake command. During the reboot process, the host needs to use the handshake command to query whether the slave is online or not.
0x11	The command is used for the working status of the HSR mode.

3. data domain: DataLen is the length of data, accounting for 1 byte, is the length of valid data in the data frame, the effective range of 0 ~ 247; Data is the data frame, the number of bytes for the number of bytes accounted for by the data frame, is not fixed, the use of the small end of the model for storage. (low byte in the front, high byte in the back), the definition of the format for the struct structural format, different businesses use different types of structural data.

4. error detection (CheckSum): check and, accounting for 1 byte, the calculation method for the binary inverse code sum. Binary inverse summation algorithm code description is as follows (MCU main control board).

```
u8 Cal_Checksum(uc8 *buf, u16 size)
{
    u16 cksum = 0;
    for(; size > 0; size--)
        cksum += *buf++;
    cksum = (cksum>>8) + (cksum&0xFF);
    return (u8) ~cksum;
}
```

Annotation:

Calculation range: Command (function) + DataLen (length) + Data (data frame).

Microcontroller main control board and Android host communication using RS232 technology standard, file path

using /dev/ttyS1, baud rate using 115200, flags parameter value using 0.

2.2 Communication Methods Realized by the MCU Main Control Board

First, define two constants, which represent the serial device path and baud rate respectively, the code is as follows.

```
#define SERIAL_PORT "/dev/ttyS1" // serial device path
#define BAUDRATE B115200 // baud rate
```

Step 1: Open the serial device and configure the serial port properties, part of the reference code is as follows.

```
int serial_fd; // serial port file descriptor
struct termios serial_config.
// Open the serial device
serial_fd = open(SERIAL_PORT,0_RDWR | 0_NOCTTY | 0_NONBLOCK);
if(serial_fd < 0)
{
    perror("Failed to open serial port device");
    exit(EXIT_FAILURE).
}
// Configure serial port properties
memset(&amp;serial_config,0,sizeof(serial_config));
// Control mode flag
serial_config.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD.
// Input mode flag, controls the terminal input method
serial_config.c_iflag = IGNPAR; // ignore parity errors
.....
```

Step 2: Write data to the serial port and send it to the Android side, the reference code is as follows.

```
char data[] = "Hello!";
int len = strlen(data);
write(serial_fd,data,len).
```

Step 3: Read the data in the serial port and receive the data from the Android side, the reference code is as follows.

```
char buf[256].
int n = read(serial_fd,buf,sizeof(buf));
if(n > 0)
{
    buf[n] = "\0";
    printf("Received data: %s\n",buf);
}
```

Finally, after reading and writing data, the serial device needs to be closed with the code close(serial_fd);.

2.3 Android side host to realize the communication method

Android application development, the realization of serial port communication, there are many mature packages to complete the communication function. The package used in this article is com.aill.androidserialport.SerialPort. the basic use of the step consists of three steps, as follows.

Step 1: Open the serial port and create the object serialPort with the code SerialPort serialPort = new SerialPort(new File("/dev/ttyS1"), 115200, 0);.

Step 2: Write data to the serial port, first from the serial port object to get the output stream OutputStream outputStream = serialPort.getOutputStream (); then write the byte data outputStream.write(message); here message message message needs to be Byte array type byte[]. After writing the data, the microcontroller can receive the written byte data at the main control board.

Step 3: After sending the data from the main control board of the microcontroller, the data in the serial port can be read. When reading the data, you may encounter a sub-packet situation, that is, you can't read the correct and complete data at one time, the solution can be to let the thread reading the data sleep for a period of time when reading the data, and then wait for all the data to be received and then read out at one time to avoid most of the sub-packet situation. First get the input stream from the serial port object InputStream inputStream = serialPort.getInputStream

(), next use the loop to read the data, part of the reference code is shown below.

```
while (!exit) { // exit Initial value is false
    try {
        if (inputStream.available() > 0) {
            received Thread.sleep(200);
            byte[] buffer = new byte[inputStream.available()];
            int size = inputStream.read(buffer);
            .....}}}
```

2.4 Communication message parsing

The communication message parsing is divided into two parts, one part is the Android side transmits the message to the MCU main control board side and the MCU main control board side parses the data, and the other part is the MCU main control board side transmits the message to the Android side and the Android side parses the data.

1. The microcontroller side parses the data: Microcontroller side in the receipt of data sent over by the Android side, the main use of memcpy function to complete the received byte stream data buf converted to structural data CmFormat structure type, CmFormat structure definition and the Android side is similar to the end, not repeat.

2. the Android side of the parsing data: Android end of the application using Java language development, first in accordance with the communication message format definition class CmFormat, part of the code is shown below.

```
public class CmFormat {
    private byte[] head;       // protocol header
    private byte command;   // function
    private byte dataLen;    // length
    private byte[] data;       // data frame
    private byte checkSum;   // Checksum
}
```

After receiving the byte data sent from the main control board of the microcontroller, use the method getCmFormatByBytes to parse the data items one by one in accordance with the communication message format, and then encapsulated into a CmFormat object, the specific code is shown below, and the method returns an object of type CmFormat.

```
public static CmFormat getCmFormatByBytes(byte[] bytes) {
    CmFormat cmFormat = new CmFormat();
    try {
        byte[] head = new byte[5];
        System.arraycopy(bytes, 0, head, 0, 5);
        cmFormat.head = head;
        cmFormat.command = bytes[5];
        cmFormat.dataLen = bytes[6];
        byte[] data = new byte[cmFormat.dataLen];
        System.arraycopy(bytes, 5 + 1 + 1, data, 0, data.length);
        cmFormat.data = data;
        cmFormat.checkSum = bytes[bytes.length - 1];
    } catch (Exception e) {
        e.printStackTrace();
    }
    return cmFormat.
}
```

After the MCU main control board and Android end complete the communication function and correctly parse the data sent by the other party, the subsequent development of business functions can be completed. After parsing the data, the need to verify the integrity of the data, as shown in Figure 2, the method used to verify the data is the binary inverse code summing, and then compared with the value of the CheckSum field in the subsection of the communication message design, the receiving end of the received data first on the Command (Function) + DataLen (Length) + Data (Data Frame) three types of data binary inverse code summing, and the value and the received checkSum field

value comparison, the value and the received checkSum field value comparison, the subsequent development of business functions can be completed. The value is compared with the received checkSum field value, if it is equal then the data is considered legal, otherwise, it is not considered legal and the data is discarded.

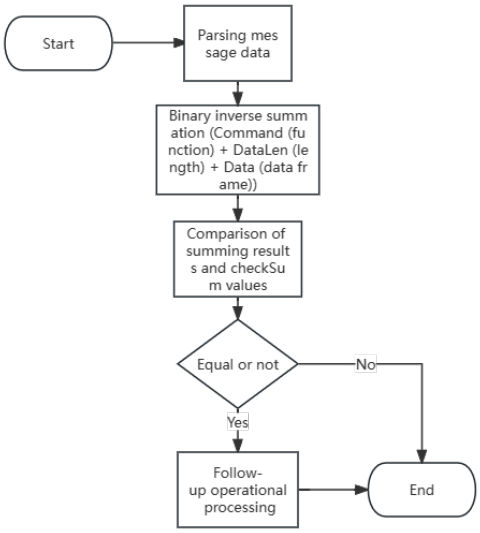

Figure 2. Data validation process.

3. Conclusion

In this paper, the background of the project "Intelligent Extermination Robot", RS232 as the communication technology, a detailed description of the project microcontroller main control board and the Android host how to realize the communication, as well as the communication message design, communication message error detection methods and communication message parsing methods, for the relevant needs of personnel to provide a communication reference program and a communication message design format. It provides a communication reference program and a communication message design format for people who have related needs. The intelligent extermination robot using this communication scheme operates stably and meets the requirements for transmission of control commands of the communication module, and it has been proved in practice that this design scheme achieves the expected results, with the advantages of accurate quasitransmission, high efficiency and low cost.

Funding

University of Science and Technology Liaoning 2022 Da Chuang Project (Project No.: S202210146068).

References

[1] Wang Bin, Zhao Zihang, Gao Hui, et al. Design of control system for intelligent extermination robot [J]. Artificial Intelligence and Robotics, 2022, 37(1): 40.

[2] Li L. Design of a highly reliable serial port communication protocol [J]. Modern Navigation, 2019(4):308.

[3] Li Yongzhong, ed. Modern Microcomputer Principles and Interface Technology [M]. Xi'an University of Electronic Science and Technology Press, 2013, p. 303.

[4] Wen Hao, Lei Shouning. Application of RS232 communication technology in microcontroller communication [J]. Information Communication, 2020(2):130.

[5] Rookie Tutorials. Android Overview What is Android [OL]. 2023. https://www.runoob.com/android/android-overview.html.