

基于功能依赖的无线通信系统设计

韩 中¹, 王泽奉², 贾晓婷^{3*}, 张起荣³, 韩 强³, 杨 雪¹

¹鞍山师范学院人工智能学院, 辽宁 鞍山

²北京邮电大学玛丽女王海南学院, 海南 陵水县

³琼台师范学院信息科学与技术学院, 海南 海口

收稿日期: 2025年4月23日; 录用日期: 2025年5月22日; 发布日期: 2025年5月30日

摘 要

近些年, 随着电子信息技术的快速发展, 无线通信产品的使用变得越来越广泛。然而, 现实中无线通信产品的安装调试却非常困难, 由于通信双方的数据传输存在着盲盒效应, 使得产品安装调试很不便。为此提出了基于功能依赖的无线通信系统设计, 通过设计关键的通信数据控制模块, 保障通信过程的可靠性。研究依据功能依赖原理把通信系统进行划分、并设计出相应的通信功能模块。依据通信的物理性依赖, 设计了嵌入式WIFI通信硬件的监控和管理功能模块; 针对通信双方设备连接的状态, 开发了通信的服务端和访问端(终端)的连接操控模块; 为了保证连接双方通信的有效性, 研发了通信数据收发测试模块。最后, 研究对设计的功能模块进行了测试验证, 结果表明研究方法是有效的可行的。

关键词

无线系统, 功能依赖, 通信服务

Design of Wireless Communication System Based on Functional Dependence

Zhong Han¹, Zefeng Wang², Xiaoting Jia^{3*}, Qirong Zhang³, Qiang Han³, Xue Yang¹

¹College of Artificial Intelligence, Anshan Normal University, Anshan Liaoning

²Queen Mary School Hainan, Beijing University of Posts and Telecommunications, Lingshui Hainan

³College of Information Science and Technology, Qiongtai Normal University, Haikou Hainan

Received: Apr. 23rd, 2025; accepted: May 22nd, 2025; published: May 30th, 2025

Abstract

In recent years, with the rapid development of electronic information technology, the use of wireless

*通讯作者。

文章引用: 韩中, 王泽奉, 贾晓婷, 张起荣, 韩强, 杨雪. 基于功能依赖的无线通信系统设计[J]. 计算机科学与应用, 2025, 15(5): 723-734. DOI: 10.12677/csa.2025.155144

communication products has become increasingly widespread. However, in reality, the installation and debugging of wireless communication products are very difficult, as there is a blind box effect in the data transmission between the communication parties, which makes product installation and debugging very inconvenient. A wireless communication system design based on functional dependencies has been proposed to ensure the reliability of the communication process by designing key communication data control modules. Research divides communication systems based on the principle of functional dependency and designs corresponding communication functional modules. Based on the physical dependence of communication, a monitoring and management module for embedded WIFI communication hardware has been designed; We have developed a connection control module for the communication server and access terminal (terminal) based on the connection status of devices between the two parties; In order to ensure the effectiveness of communication between the two parties, a communication data transmission and reception testing module has been developed. Finally, the research conducted testing and verification on the designed functional modules, and the results showed that the research method is effective and feasible.

Keywords

Wireless System, Functional Dependency, Communication Services

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近些年来,无线通信技术的应用更加广泛[1],促进了WIFI、NFC、蓝牙等近场无线通信技术的出现[2][3]。其中,WIFI技术由于它传输距离远与传输效率高的特性而被人们优先使用[4]。WIFI功能不只具有无线上网的功能,用户还可以通过WIFI网络进行远距离的通信[5],甚至实现了对某些设备的智能控制[6],使生活更加智能化。目前,嵌入式系统在日常生活中使用非常普遍[7][8],于是,就带来了嵌入式终端访问网络资源的问题[9][10]。而对于嵌入式系统的WIFI能否实现通信连接[11]、上网获取网络资源成为了一个焦点问题[12][13]。所以,研究基于功能依赖的无线通信系统的设计是非常必要的[14][15]。鉴于嵌入式系统的WIFI无线通信应用的广泛性,本文主要以WIFI无线通信为基础展开研究设计[16]。

WIFI是使用IEEE802.11b的协议进行通信的,它能够个人电脑、手持设备等多种嵌入式终端连接起来[12],形成一个无线的局域网,实现设备之间的无线通信[13]。WIFI通信这些年来主要用在PC机上,现在要把WIFI通信引进到嵌入式系统中来。但是,对于一个普通的嵌入式终端来说,这方面功能的使用就没有这么广泛,而且有些技术保密与限制机制的存在,造成了技术壁垒。于是,对它进行研究、形成并提供经典范式还是非常有意义的。

嵌入式产品所支持的操作系统有Vxworks、IOS、以及谷歌公司的Android等多个种类,其中Android操作系统有多种优势而被广泛使用,首先因其占有开源代码的优势被大多数开发者所选择;其次,Android嵌入式系统是将应用于大型服务器上的Linux系统进行裁剪修改而成的一种新型操作系统,具有使用方便、技术成熟、功能强大、运行稳定等优点;另外,Android作为嵌入式操作系统还有它可以应用于多种硬件平台,以及有利于缩短软件与硬件的开发周期的优势。本文所研究的WIFI通信应用就是基于Android系统开发平台而开发的。

在嵌入式Android系统环境下实现的WIFI通信应用设计,其开发内容主要包括三个部分:通信系统资源监控与管理、通信功能接口开发、系统通信服务。研究最后通过系统开发实验测试验证了设计方法

的可行性和实用性。

2. 功能依赖的通信设计

2.1. 无线通信分析

嵌入式 WIFI 通信系统由硬件系统与软件系统两个大的部分构成，硬件系统采用嵌入式系统与现有的 WIFI 模块的集成，软件系统则需要由用户编程来完成。WIFI 通信系统服务主要包括：服务端与终端的 WIFI 连接服务、通信系统的 WIFI 监控服务、服务端与终端之间的通信服务。通信服务可以从以下几点进行理解。

(1) 对嵌入式硬件系统 WIFI 模块进行识别与管理，硬件系统是通过扫描 WIFI 网络，输入秘钥连接 WIFI 无线上网的。

(2) 对通信系统进行功能分类，划分为不同的独立功能应用模块，通过不同的功能模块实现不同的通信功能操作。

(3) 功能模块的算法实现。整体设计采用了分层模块化程序设计思想，各个模块功能独立实现。这样处理后，系统的可操作性和功能扩展性得以加强，出现错误也好及时发现与纠正。

(4) 通信服务通过无线 WIFI 支撑运行。相互连接的无线 WIFI 系统可以进行多种媒体形式的信息发送和传输，如无线 WIFI 网络办公、文件传递、音乐视频的分享等。

以上情况能够凸显嵌入式无线 WIFI 的性能优势，突破时间空间的限制，为用户的各种活动提供快捷的基础性的应用支撑。

2.2. 功能依赖的通信架构设计

为了实现通信系统的无线通信及其方便维护的要求，这里使用了功能依赖的设计方法，通过此方法设计出支撑系统通信必须的功能模块。支撑整个系统通信的功能模块构成了通信系统架构。功能依赖就是为了实现某项系统目标而按照相互依存关系设计的系列支撑性功能模块，系统和模块两个实体就构成了功能依赖关系。功能依赖的功能模块可以是物理产品、技术方法、技术组织等资源。功能依赖可以使用数学模型进行表达，通过对系统建模来实现。功能依赖方法不同于以往的功能框图法，功能框图法主要是把系统各组成功能使用框图的方式直接呈现出来，对模型设计意义考虑较少；而功能依赖方法的系统设计是按照系统组织部分的依存关系逻辑得到的功能模块，更具科学意义。

为了实现嵌入式系统的 WIFI 通信，要先对硬件 WIFI 模块进行设计，然后进行软件功能设计。模块设计中，硬件设计为现有的嵌入式通用模组，软件是自行设计。软件功能能够体现 WIFI 通信功能，控制嵌入式的服务端服务的开启。服务将通过 Android 操作系统的两个嵌入式(终端)终端进行接入连接，形成一个 WIFI 无线网，都连接到整个服务网，使用终端与终端之间数据通信的方式来实现嵌入式 WIFI 通信。应用程序实现用户之间的一种更加方便的方式去实现通信和信息传递。

通过上述对无线通信系统的描述，可以设无线通信系统为 S ，通信系统 S 的功能单元为 $F(X)$ ，这里称 X 是 $F(X)$ 的依赖，如果 X 存在 N 个分量。

那么就有式(1)：

$$\begin{cases} X = \{X_1, X_2, \dots, X_N\} \\ 5 \leq N \leq 9 \end{cases} \quad (1)$$

式(1)中，按照软件工程的思想，同层次模块分量 N 的取值一般为 5~9 之间，即 $N = 7 \pm 2$ 。

进一步得到 $F(X)$ 的划分，即式(2)：

$$F(X)=\{F(X_1),F(X_2)\cdots F(X_N)\} \tag{2}$$

这里称 $F(X_1)\dots F(X_N)$ 是 $F(X)$ 的多个依赖。

如果 X_N 存在 M 维度上划分，即有式(3)。

$$\begin{cases} X=X_{MN} \\ 5\leq N\leq 9,1\leq M\leq 5 \end{cases} \tag{3}$$

相应的有式(4):

$$F(X)=\{F(X_{1N}),F(X_{2N})\cdots F(X_{MN})\} \tag{4}$$

那么， $F(X)$ 就构成了一个矩阵，其对应每一个分类功能表达可记为 f_{mn} 。

功能矩阵如下式(5):

$$F(X)=\begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mn} \end{bmatrix} \tag{5}$$

根据依赖规则，任一功能分量必须是独立的、不可重复的模块。功能模块则存在下列关系式(6)。

$$\begin{cases} F(X_{MN})\otimes F(X_{IJ})=\Phi \\ M\neq I,N\neq J \end{cases} \tag{6}$$

基于上述功能依赖的通信系统设计思想，嵌入式 WIFI 通信系统的系统架构设计的图示表示如图 1 所示。架构类似一个树形结构，结构展示了嵌入式 WIFI 通信的各个重要的单元模块，以及实现通信过程的模块关系。

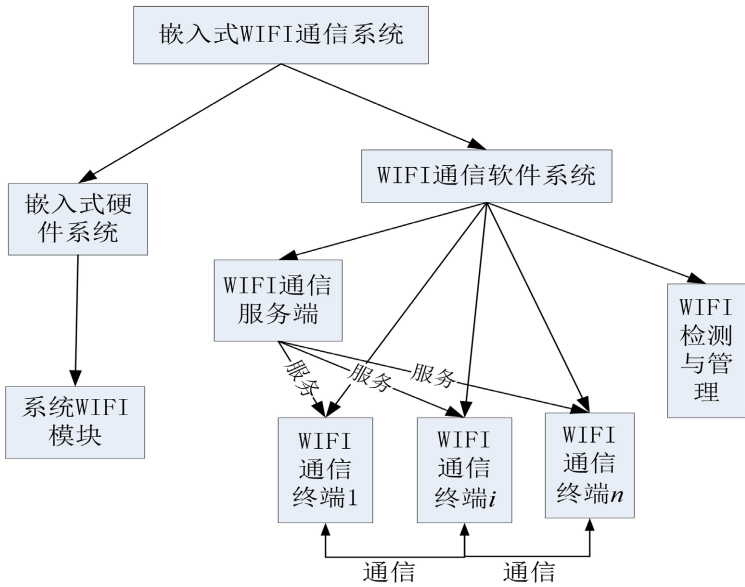


Figure 1. Diagram of the communication system
图 1. 通信系统架构图

2.3. 通信服务的功能依赖设计

按照功能依赖的设计方法设计出通信系统的服务功能。

WIFI 通信系统硬件为嵌入式系统与现有 WIFI 模块的集成, 嵌入式内置 WIFI 驱动、MAC、WIFI 协议、无线安全协议等部分。

WIFI 通信系统软件为 Android 系统应用软件, 可分为以下三个模块: WIFI 连接器、WIFI 通信终端和 WIFI 通信服务端模块。

WIFI 连接器基本功能如图 2 所示:

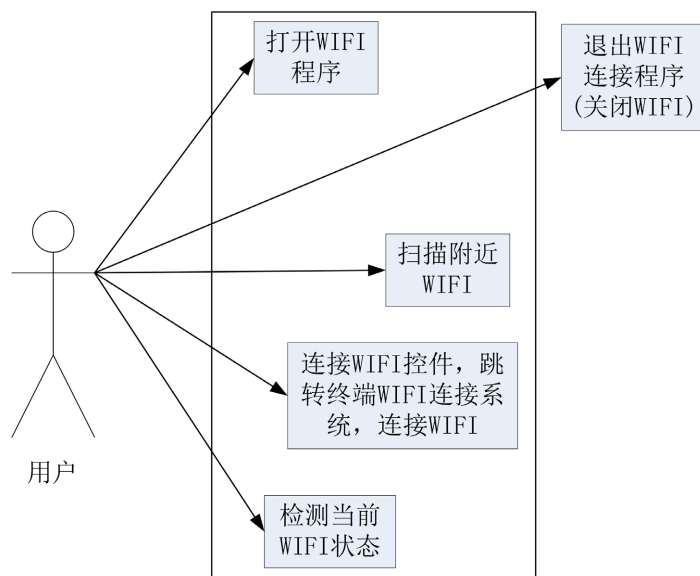


Figure 2. Diagram of the basic functions for WIFI connector

图 2. WIFI 连接器基本功能图

当 WIFI 连接成功后, 返回系统进入通信应用软件 APP (终端), 再控制嵌入式设备(服务端)也接入 WIFI 无线网并开启服务器, 使用终端检测连入服务端, 最后实现终端与服务端通信。

通信流程如图 3 所示:

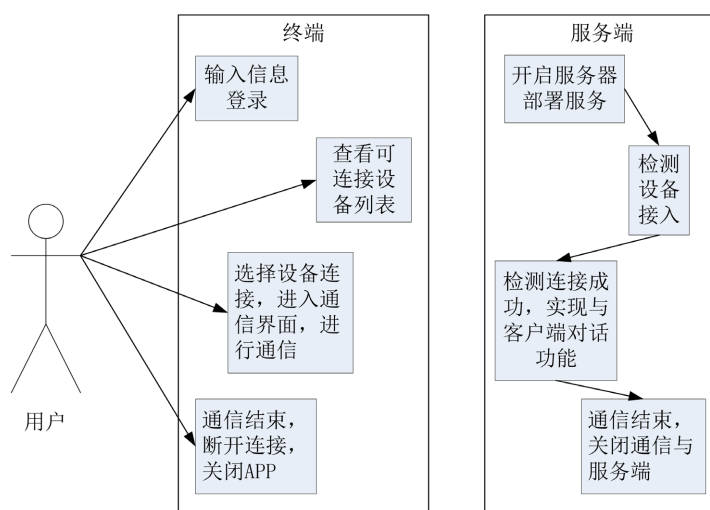


Figure 3. Diagram of the communication processes

图 3. 通信流程图

从整个流程图中可以得出的在整个应用程序运行中,参与方式大部分都采用用户直接参与的方式进行,这就更好地从软件方面实现了需求中提及了的增强客户体验的要求。在通信过程中,服务端可根据用户接入服务方式不同,进行点对点电线通信和一对多信息群发功能的选择处理。

3. 功能模块的实现方法

按照功能依赖方法在实现系统功能模块的同时,还要考虑系统通信连接的稳定性、安全性等问题。通信的稳定性与安全性研究较为广泛,本文仅从软硬件基础连接可视方面进行设计,能够较好的保障通信系统基本的稳定性和安全性。此部分给出了关键的连接设计,解决软硬件相互之间的基本连接问题。进一步为保障通信连接的稳定性与安全性,重点介绍了设备连接设计方法、通信服务连接设计方法。

3.1. 关键连接的设计

除了一些众所周知的英文缩写,如 IP、CPU、FDA,所有的英文缩写在文中第一次出现时都应该给出其全称。文章标题中尽量避免使用生僻的英文缩写。

系统软硬件的连接是通过接口实现的,WIFI 通信的接口分为两类:可视界面的接口和内部函数类接口。前者也就是通常所说的界面,即 UI 开发,后者则是内部调用的函数接口,通常不见。下面给出了常用接口的两大类开发方法。

(1) WIFI 通信设备接口

在 WIFI 通信设备应用打开之后进入欢迎界面模块或者说是主界面,其布局使用了 `RelativeLayout` 整体布局,用 `ImageView` 类来显示图片,然后对其大小、形状、位置、颜色等进行设计,呈现相应的界面设计效果。

主界面对应的内部程序接口如下代码。

```
public class SplashActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        ThreadUtils.runInSubThread(new Runnable()
            public void run(){
        try {
            Thread.sleep(3000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        })
        startActivity (new Intent(getApplicationContext(),
        LoginActivity.class));
```


(2) 收发信内容接口模块

通信信息的内容显示在发送与接收的窗口当中，窗口采用了两个 `Linear Layout` 的布局设计。显示内容包括通信时间，用户头像，发送的消息等。

接收界面与发送界面的布局区别就是为左对齐布局和文本接收框颜色不同，其他部分全部一致。

(3) 通信设备控制接口

管理所有可连接和可交换信息的设备，把所有的通信设备进行列表，其界面设计为 `LinearLayout` 水平线性布局，显示设备标题、颜色、大小、图标等内容。

3.2. 设备连接设计

(1) 设备管理 UI 设计

设备管理设计了一个通过扫描可展示的 `WIFI` 列表界面，列表中显示的是周围可连接的 `WIFI` 设备。界面中包含的可连接设备可进行选择操作，当某设备被选中时，可对其进行发送信息、连接该设备、断开连接等方面的操作。

(2) `WIFI` 监管的实现

`WIFI` 状态的监测功能设计，旨在提升物理设备连接的稳定性。在主功能实现程序中创建了一个 `Wifi Activity` 类，能够实现对 `WIFI` 的监听，见下列功能实现代码。

```
//设置监听
iv_switch.setOnClickListener (OnClickListener);
scan.setOnClickListener (OnClickListener);
connet.setOnClickListener (OnClickListener);
checkNetWorkState.setOnClickListener (OnClickListener);
new Task().start();
```

随后判断当前 `WIFI` 的状态，若打开，则不用再开 `WIFI` 开关，直接点击扫描 `WIFI` 按钮进入下一步即可，否则先开启 `WIFI` 后才能进行扫描阶段。调用 `Wifi Helper.java` 文件检测与扫描 `WIFI`，获取扫描结果。扫描完成后，单击 `WIFI` 连接按钮，程序跳转至嵌入式系统设置中连接 `WIFI` 界面，选择可以连入的 `WIFI` 网络，输入 `WIFI` 秘钥即可连接。再检测一次 `WIFI` 连接状态，设置新的结果(`WIFI` 已开启或是 `WIFI` 已关闭)，编程如下：

```
//检测 WIFI 的当前状态
class Task implements Runnable {
    final handler handler = new Handler();
    public void start(){
        //要将之前发送的消息移除掉，避免发送重复消息
        handler.removeCallbacksAndMessages(null);
        handler.postDelayed(this, 500);
    }
    @Override
    public void run(){
        changeState ();
        handler.postDelayed (this, 500);
    }
}
```

```

    }
    public void changeState () {
    if (checkNetCardState())
        iv_switch.setActivated(true);
        tv_toast.setText ( “wifi 已开启” );
    } else {
        iv_switch.setActivated(false);}

```

由于扫描周围 WIFI 得到的结果与显示结果功能较为复杂, 所以建立一个新的 MyAdapter 类, 构建判断方法测试 WIFI 信号强度, 并对应不同等级强度的 WIFI 信号图片, 与 WIFI 信息一同在主布局界面列表控件中显示出来。

```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View view = null;
    view = inflater.inflate(R.layout.item_wifi_list, null);
    ScanResult scanResult = list.get(position);
    TextView testview = (TextView) view.findViewById(R.id.textView);
    testview.setText(scanResult.SSID);
    ImageView imageView = (ImageView) view.findViewById(R.id.imageView);
    //检测 wifi 强度
    if (Math.abs(scanResult.level) > 80) {
        imageView.setImageDrawable (context.getResources().getDrawable(R.drawable.xg));
    } else if (Math.abs(scanResult.level) > 60) {
        imageView.setImageDrawable (context.getResources().getDrawable(R.drawable.xf));
    } else {
        imageView.setImageDrawable (context.getResources().getDrawable(R.drawable.xe));
    };
    return view;
}
}

```

3.3. 服务连接设计

WiFi 通信服务连接的设计, 重在解决连接的安全性问题。

服务端提供服务是先把服务器启动, 然后检测设备终端连接情况。

① 创建一个线程, 等待其他终端的连接; ② 如果终端连接成功, 分配一个线程给终端; ③ 在该线程内等待用户数据; ④ 最后分配一个线程给终端, 编程如下。

```

public class Server {
    /**
     * @param args
     */
    Public static void main(String[] args){

```



```

try {
    // ①创建一个线程，等待其他终端的连接
    final ServerSocket server = new ServerSocket(8090);
    System.out.println(“---服务器启动---” + new Data().toString());
    new Thread() {
        public void run() {
            while (true) {
                Connection conn = null;
                try {
                    Socket client = server.accept();
                    System.out.println(“---有客户端接入---” + client);
                    //②如果终端连接成功，分配一个线程给终端
                    conn = new Connection(client);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                conn.addOnRecevieMsglistener (new LoginMsgListener (conn));
                conn.addOnRecevieMsglistener (new LoginMsgListener (conn));
                conn.addOnRecevieMsglistener (new LoginMsgListener (conn));
                conn.addOnRecevieMsglistener (new LoginMsgListener (conn));
                //③在该线程内等待用户数据
                ...
            }
        }
    };
}

```

Connection 类中存放连接，断开连接，配置监听器与等待数据的功能程序段。

在连接消息类中，向服务器发送信息提示终端上线与下线，并把下线的终端从列表中移除，以免混淆，协议相应程序如下。

```

public class ConnectionManager {
    public static HashMap<Long, Connection> cons = new HashMap < Long, Connection > ();
    public static BuddyList list = new BuddyList();
    public static void put(long accout, Connection conn) {
        System.out.println(“====账号” + account + “上线了”);
        remove(account, conn);
        User u = Db.getByAccount(account);
        Bubby item = new Buddy();
        item.account = u.account;
        item.avatar = u.avatar;
        item.nick = u.nick;
        list.buddyList.add(item);
    }
    public static void remove(long account) {
        if (conns.containsKey (account)) {
            System.out.println(“====账号” + account + “上线了”);

```

为了终端方便连入服务器，将通信服务端程序生成 Jar 包在远程服务器中一直运行。

4. 通信功能测试

软件设计完成，进入测试环节，根据测试结果对软件系统进行功能优化，增加软件稳定性。测试环境为 Android 4.3 版本以上的嵌入式系统。系统的关键测试点有 WIFI 连接程序测试、WIFI 通信终端测试、WIFI 通信服务端测试。

4.1. WIFI 连接测试

WIFI 连接测试是测通信终端的连接状态，连接状态是通过嵌入式系统应用程序的图标显示给用户。进入设备连接界面，检测当前 WIFI 状态，点击图标打开 WIFI，扫描周围 WIFI 网络，点击连接 WIFI 按钮，在系统中选择 WIFI，输入密码连入 WIFI，设备连接的状态使用下列图标表示，见图 4 和图 5 所示。



Figure 4. WIFI connect closed
图 4. WIFI 连接关闭



Figure 5. WIFI connect open
图 5. WIFI 连接开启

在 WIFI 关闭情况下，检测状态为 WIFI 已关闭，如图 4 所示，点击 WIFI 图标打开 WIFI，WIFI 图标显示为绿色并由关闭图标换成开启图标，显示为 WIFI 连接开启或成功，如图 5 所示。

4.2. WIFI 通信服务测试

WIFI 通信服务模块的各类服务操作可以通过点击应用程序显示的图标实现。进入主界面后，按登录协议输入账号密码登录，进入设备列表界面，选择可通信的设备，进行收发消息通信。规定不能与自己通信，通信时空消息不能发送。

登录成功后，系统跳转至已连接设备列表界面，并显示登录成功，接下来就可以进行信息通信了，例如设备 5 向设备 4 发送消息“你好”，设备 4 收到消息，显示消息“你好”，效果如图 6、图 7 所示。



Figure 6. Message sending from Dev. 4 to Dev. 5

图 6. 设备 4 向设备 5 发送消息



Figure 7. Receipting message in Dev. 5

图 7. 设备 5 接收消息

嵌入式终端设备 4 再给设备 5 发送“你好,很高兴认识你!”的消息,设备 4 完整的收到消息,消息成功地显示在设备 4 上。

5. 总结

提出的基于功能依赖的无线通信应用设计了 WIFI 通信资源的监控与管理问题、实现了 WIFI 通信基

本接口的开发、以及通信的数据信息传输的测试方法等,为 WIFI 通信开发活动提供了基本的范式参考,节约了嵌入式 WIFI 通信应用开发活动的重复浪费,提升工作效率和开发周期。在应用程序开发过程中,对软件功能进行分析,对接口界面进行布局,按照活动逻辑设计应用软件中的布局与相关功能。应用迭代式的功能调试,检测应用程序功能的使用情况。

在开发过程中,研读了国内外论文和期刊,网上的技术博客,实现了以上通信系统基本功能。因此,课题还存在很多需要改进和完善的地方,例如基于 WIFI 通信的效率问题,通信过程的稳定性、可靠性问题, WIFI 通信网络连接优化问题都是后续值得研究。

致 谢

论文受到了多个基金的资助,在此表示感谢。

基金项目

海南省科技厅 2022 年自然科学基金重大专项项目,题为“拓扑向量空间辨识及其多网络车辆智能诊断应用”,项目编号:722RC740。辽宁省教育厅 2023 年度基本科研项目、面上项目“社交网络谣言传播源点定位方法研究”,项目编号:JYTMS20231713。2025 年度海南省高等学校教育教学改革研究重点项目“大学生移动学习力评价指标体系构建研究”,项目编号:Hnjg2025ZD-63。

参考文献

- [1] 虞文武, 聂晓凯, 蔡豫晋, 等. 智能超表面辅助的无线通信网络性能优化综述[J]. 中国科学: 信息科学, 2024, 54(11): 2503-2517.
- [2] 张永薪. 基于嵌入式 4G/WIFI 远程移动监控系统设计[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2021.
- [3] 夏晨阳. 无线通信网络中的干扰管理与优化策略[J]. 产品可靠性报告, 2024(12): 64-65.
- [4] 曹晓宇, 胡小生, 高杰. 基于区域无线通信的线路数据传输研究[J]. 铁路通信信号工程技术, 2024, 21(12): 48-56.
- [5] 邓欣莹, 江水明. 基于无线通信技术在远程用电检查系统设计与实现[J]. 电子技术, 2024, 53(12): 370-371.
- [6] 张澄安, 李保国, 杜志毅, 等. 基于软件无线电的雷达嵌入式通信验证系统[J]. 信号处理, 2021, 37(11): 2041-2053.
- [7] Zhang, H., Wentzloff, D.D. and Kim, H.S. (2016) Software-Defined, WiFi and BLE Compliant Back-Channel for Ultra-Low Power Wireless Communication. 2016 *IEEE Global Communications Conference (GLOBECOM)*, Washington DC, 4-8 December 2016, 1-6. <https://doi.org/10.1109/GLOCOM.2016.7842275>
- [8] 李小孟. Android 和 WIFI 通信在智能家居系统设计中的应用[J]. 电视技术, 2018, 42(5): 107-111.
- [9] 张一鸣, 肖晓萍. 基于 ARM 和 WIFI 通信的智能开关控制器设计[J]. 计算机测量与控制, 2018, 26(8): 83-87, 137.
- [10] 王雪. 基于 MicroPython 的 WIFI 通信在嵌入式系统中的应用[J]. 电子测试, 2021(16): 61-63.
- [11] 王春鸣. 嵌入式通信系统优化设计[J]. 电子测试, 2018(13): 136-137.
- [12] 刘新宇, 周恒, 葛锡云, 等. 水下无线通信装备发展研究[J]. 中国工程科学, 2024, 26(2): 38-49.
- [13] Kim, H.S. and Wentzloff, D.D. (2016) Back-Channel Wireless Communication Embedded in WiFi-Compliant OFDM Packets. *IEEE Journal on Selected Areas in Communications*, 34, 3181-3194. <https://doi.org/10.1109/JSAC.2016.2612078>
- [14] 王学文, 程思齐, 武世勇, 等. 基于功能依赖网的评估指标体系构建[J]. 火力与指挥控制, 2024, 49(7): 97-103.
- [15] 浣顺启, 方哲梅, 王剑波. 基于功能依赖网的体系效能评估方法[J]. 系统工程与电子技术, 2022, 44(7): 2191-2200.
- [16] 王哲, 李建华, 刘子杨, 等. 基于功能依赖的网络信息体系建模及重心分析[J]. 系统工程与电子技术, 2021, 43(10): 2876-2883.