

**硬件课程设计报告（个人）**

——触控感应键鼠的设计

|  |  |
| --- | --- |
| 专业班级： | 信卓2101班 |
| 姓名： | 王梦婷 |
| 学号： | U202114190 |
| 指导教师： | 吴建新 |
| 日期： | 2024年6月12日 |

# 引言

21世纪初，随着智能手机和平板电脑的兴起，触控技术迎来了爆炸性的发展。2007年，苹果公司推出的iPhone采用了多点触控技术，这彻底改变了人们与电子设备交互的方式，使得触控操作更加直观和自然。多点触控技术允许用户通过多个手指同时操作屏幕，实现缩放、旋转等复杂手势。

随后，触控技术不断进步，出现了更多种类的触控屏，如表面声波触控屏、光学触控屏等。这些技术在教育、医疗、零售等多个领域得到广泛应用，极大地丰富了人机交互的方式。

如今，触控技术已经成为现代电子设备中不可或缺的一部分，赢得了广大用户的喜爱。然而，传统的物理键盘和鼠标在便携性和交互体验上仍存在一定的局限性。因此，本课程设计旨在通过实践操作，设计一款基于触控感应技术的键盘与鼠标二合一设备，旨在解决这些问题。

# 项目概述

## 1、课设目的

硬件课程设计是华中科技大学电信学院相关专业本科学生必修的实践性教学环节。华中科技大学电子信息与通信专业的硬件课设旨在通过实践操作培养学生的实践能力、创新思维和技术技能，同时提升团队协作和工程素养，为学生的职业生涯和未来研究打下坚实基础。通硬件课程设计的目的在于使学生通过硬件课程设计教学环节较系统地完成电子系统设计从选题、方案论证、电路设计、电路实现、装配调试、系统测试、总结报告等基本过程，加深对模拟电路、数字逻辑电路、通信电子电路、微机原理等相关课程理论知识的分析理解。过这一过程，学生不仅能够掌握电子电路设计、PCB布局和硬件调试等专业技能，还能了解行业标准、激发研究兴趣，并在跨学科学习中拓宽视野，最终通过成果展示和交流，提高表达和沟通能力，为适应快速变化的技术环境做好准备。

## 2、项目描述

本项目以Xilinx FPGA为核心控制器，通过检测电路，将微小的电容改变量转换成电信号。在采集到电容按键数据后，FPGA由串口驱动与蓝牙芯片连接，使用指定的蓝牙协议将数据发送到上位机或终端设备，用感应电容键盘信号控制终端设备。上位机程序使用C#或Labview编写。该设备通过检测电路，将微小的电容改变量转换成电信号，进而控制终端设备的操作。该设备不仅具有便携性强的特点，还提供了更为直观、便捷的交互体验。用户只需在触控板上轻轻触摸或滑动，即可实现鼠标的点击、移动和键盘的输入功能。完成如下功能：触摸纸上的按键区域，控制PPT翻页或输入文档；无线游戏手柄（在纸上用铅笔绘制出按键区域，触摸后可以控制游戏运行）。

# 设计与实现背景

## **1、**任务设计

1）触控板的设计注重了用户使用的便捷性和操作的精确性。用户只需轻轻触摸板面，即可轻松控制游戏角色进行上下左右的移动。触控板的反应灵敏，能够迅速捕捉到用户的细微动作，并将其转化为准确的指令，通过蓝牙无线传输技术发送至电脑。

2）利用触控板，以无线连接的方式，用户可以轻松控制电脑。在鼠标模式下，轻滑触控板即可使鼠标光标灵活上下左右移动，轻按即可执行左键点击操作。还可以切换至键盘模式，触控板即刻变为虚拟键盘，便于用户快速打字，实现无缝切换的便捷操作体验。

## **2、**实现背景

1）硬件清单

* 电容触控传感器：一块TTP226 8路电容式触摸开关模块。
* FPGA控制平台设计：

使用Xilinx Nexys 4 DDR开发板时，确保稳定供电和高效数据传输。接着，编写FPGA逻辑程序，该程序负责接收、处理电容模块的数据，执行必要的计算或转换，并输出处理结果。

* 蓝牙模块电路：选择HC-08主从一体蓝牙模块。

2）软件开发及平台

* FPGA程序开发

使用Verilog、Vivado平台编写FPGA程序，完成电容数据的采集、处理和传输控制。进行仿真和调试，确保程序的正确性和稳定性。

* 上位机程序开发

使用C#、visual studio平台编写上位机程序，完成蓝牙数据的接收和处理。实现按键信号对终端设备的控制功能。

* 蓝牙协议

基于Bluetooth Specification V4.0 BLE 蓝牙协议，进行数据传输。确保数据传输的可靠性和实时性。

# 项目功能指标

1、功能指标：

1. 感应触摸：检测在触摸纸上绘制的按键区域，并转化为相应的电子信号。键盘布局合理，按键舒适，满足日常打字需求。
2. 数据解析：FPGA被用于处理捕获的电容按键数据。
3. 无线通讯：利用蓝牙模块，将解析后的数据无线地发送到主机或终端，保证传输速度、距离和稳定性。
4. 设备控制：在主机或终端上，基于感应到的触摸信号，实现如点击、文本编辑或游戏操作等功能。使用过程中，设备应保持稳定，不易出现断连、卡顿等现象。

2、性能需求：

1. 蓝牙连接性能：满足一般办公和娱乐需求，无明显延迟。
2. 响应速度：触控感应信号的响应时间应足够快，，确保用户操作的即时反馈。
3. 精度：能够准确检测到微小的电容变化，确保触控操作的精确性。
4. 可靠性：系统在各种环境下稳定运行，不受外界干扰影响。

3、团队分工：

（1）上位机编程---吴永浩：

负责整体系统设计和架构规划。

设计和实现蓝牙模块的集成。

完成电脑端键鼠输入信号处理的设计。

（2）FPGA编程----李鑫瑞：

负责触控板信号的输入处理。

负责触控板性能的升级。

搭建测试平台和环境，对系统进行验证和调试。

（3）系统调试----王梦婷：

负责处理触控板输入信号并传输至蓝牙发送端。

负责对硬件和软件进行系统联调。

搭建测试平台和环境，对系统进行验证和调试。

# 系统设计与实现原理分析

## 系统框图

在本项目中，我们设计了一个高度集成的系统框图，明确了各个模块之间的数据流向和控制逻辑。系统框图不仅展示了系统的整体结构，还体现了模块化设计的优越性，便于后续的维护和升级。

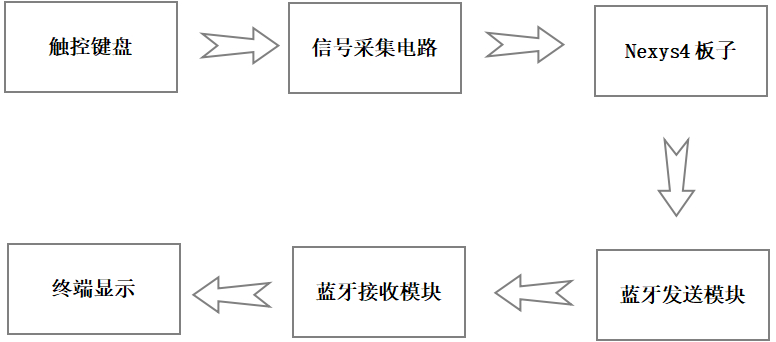


图 1系统框图

## **2、**模块理论分析

（1）触控键盘——TTP226

TTP226 作为一款电容式触摸检测集成电路，其低功耗和宽工作电压的特点使其在本项目中发挥了关键作用。我们详细分析了 TTP226 的信号采集电路，以及如何通过顶层模块 top\_module 进行时钟分频和状态机控制，确保了数据的准确采集和及时传输。

TTP226是一款接触板检测(touch pad detector)IC，提供8个接触键。接触检测 IC 有意取代传统固定pad尺寸的直接按钮键。低功耗和宽工作电压是接触键在DC或AC应用中的特点。

TTP226是一款电容式触摸检测集成电路（IC），它能够提供多达8个触摸键，并且设计用来取代传统的固定尺寸触摸板或按钮。这款IC具备低功耗和宽工作电压的特点，使其适合在直流（DC）或交流（AC）应用中使用。TTP226的工作原理基于电容式感应，能够检测到人手的接近或触摸，从而触发相应的按键动作。它广泛应用于电子设备中，如遥控器、家用电器、工业控制设备等，以提供用户友好的交互界面。當VDD=3V時System clock約為440KHz,大約55Hz更新輸出一次。大多数输出模式工作在直接(direct)或串行(serial)模式。只有当OPS0=0时，输出模式会为矩阵(matrix)类型。

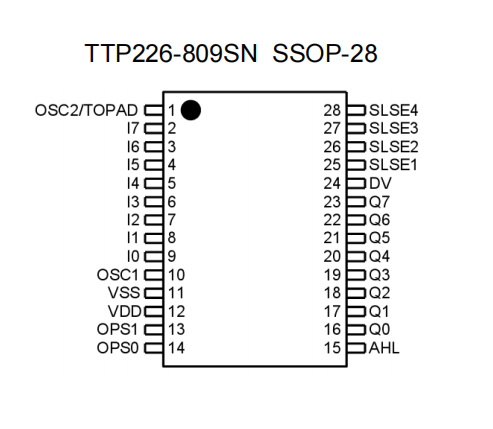


图 2 TTP226封装结构

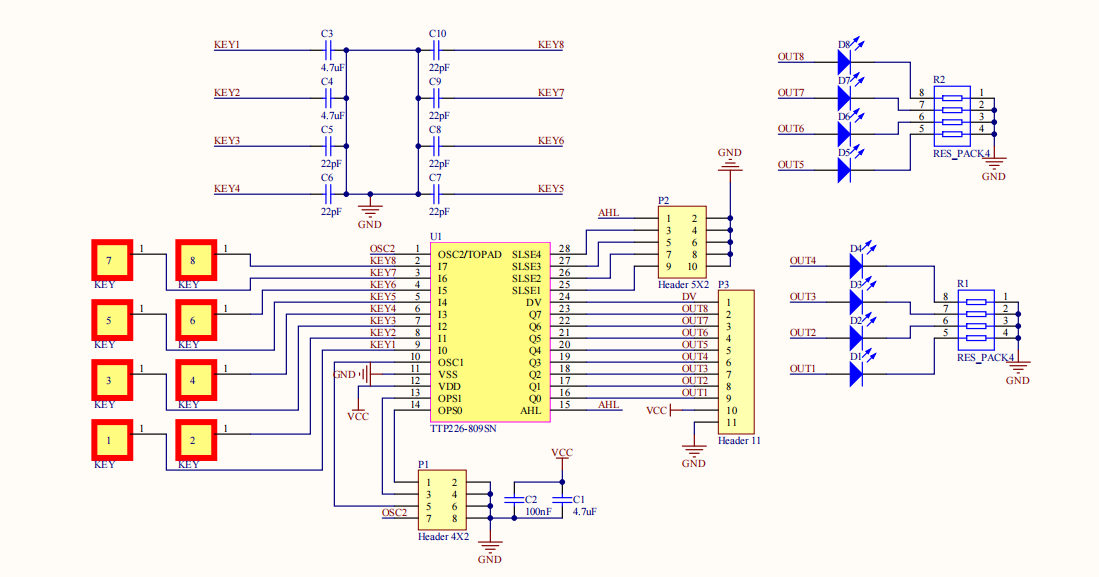


图 3 电路原理图

1. 信号采集电路

触摸输入中读取数据，经过处理后通过UART发送出去。

顶层模块 top\_module：

A.时钟分频器和状态机

计数器 clock\_counter: 用于生成一定间隔的时间信号，这里定义为23位宽，可以计数到5000000。

参数 CLOCK\_DIVIDE: 设置为5000000，用于100MHz时钟频率下生成约100ms的时间间隔。

B.状态机: 在时钟上升沿或重置信号有效时进行状态转换。

当重置信号有效时，将计数器置零，tx\_data和tx\_data\_valid也置零。否则，计数器自增，当计数到CLOCK\_DIVIDE时：将touch\_output扩展为8位并赋值给tx\_data（高4位置零）。tx\_data\_valid置为1，表示数据有效。

若未达到CLOCK\_DIVIDE，则计数器继续自增，tx\_data\_valid保持为0。

1. Nexys4 DDR开发板

Nexys4 DDR开发板作为本项目的核心控制平台，其高性能的 Xilinx Artix-7 FPGA 芯片为数据处理提供了强大的支持。我们详细介绍了开发板的内存、扩展接口、显示控制以及视频音频处理能力，这些都是实现复杂人机交互功能的基础。通过具体的应用示例，我们展示了如何利用 Nexys4 DDR 的丰富资源来构建项目原型。

Nexys4 DDR开发板是来自知名的Digilent公司，基于Xilinx的Artix-7 FPGA系列，以其卓越的性能和丰富的功能，吸引了无数学生和工程师的目光。

这款开发板的核心是一块Xilinx Artix-7 XC7A200T FPGA芯片，它拥有高达200万逻辑单元和716百万门的逻辑能力，为用户提供了强大的数据处理能力。此外，512MB的DDR3 SDRAM内存保证了高速数据处理和存储的需求。



图 4 Nexys4DDR

Nexys4 DDR的扩展接口丰富多样，包括两个Pmod接口和两个Arduino兼容的RJ45接口，这使得用户可以轻松地连接各种扩展模块，实现更多的功能。无论是传感器、电机驱动器还是其他外设，都能轻松与Nexys4 DDR连接。

在显示和控制方面，Nexys4 DDR同样表现出色。板上集成了两个7段显示器和16个LED，用户可以通过编程控制这些显示设备，实现各种状态的指示和简单的信号输出。同时，4个滑动开关也为用户提供了方便的信号输入和配置方式。视频和音频处理也是Nexys4 DDR的一大亮点。它支持VGA视频输出，可以连接显示器显示图形界面；同时，音频接口可以处理音频信号，为多媒体应用提供了可能。存储和通信方面，Nexys4 DDR内置了SD卡插槽和USB接口，方便用户进行数据存储和程序加载，以及与PC的通信。此外，板载JTAG接口也为FPGA的编程和调试提供了方便。

综上所述，Nexys4 DDR开发板以其强大的性能和丰富的功能，成为了电子设计和教育领域的得力助手。无论是学术研究、教学演示还是工业应用开发，它都能提供必要的硬件支持和软件工具，助力用户实现更多的创新和突破。

1. 蓝牙发送与接收模块

HC-08蓝牙模块的集成是实现无线通信的关键。我们深入分析了 HC-08 的特性和功能，包括其无线工作频段、调制方式、发射功率和接收灵敏度等。此外，我们详细描述了如何通过 AT 命令对模块进行编程和配置，以及如何确保模块与各种蓝牙设备进行稳定通信。

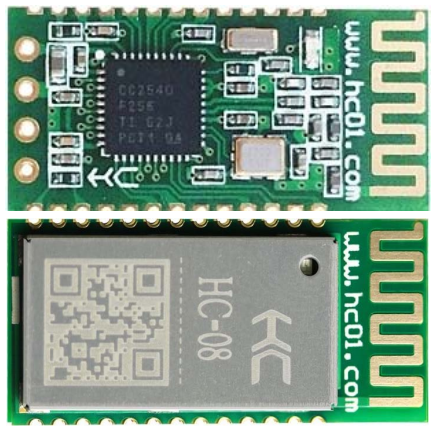
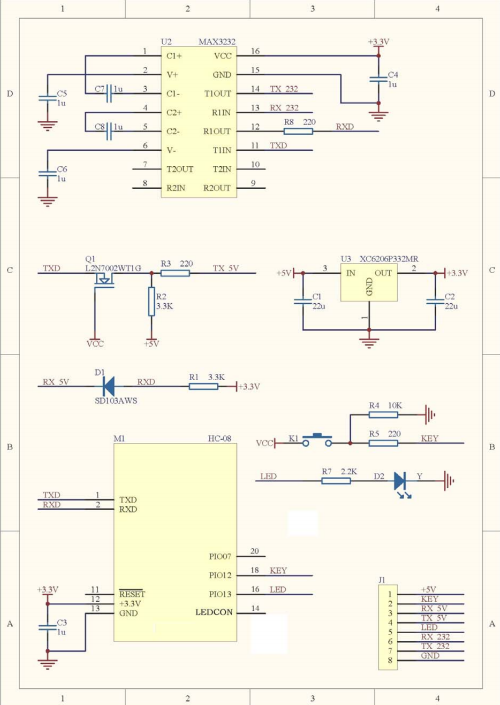


图 5 HC-08具体结构

HC-08是一款流行的蓝牙串口模块，广泛用于各种电子项目中，以实现设备之间的无线通信。以下是HC-08的一些基本特性和功能：

HC-08模块兼容蓝牙2.0+EDR，串口波特率默认是9600bps.无线工作频段为 2.4GHz ISM，调制方式是 GFSK。模块最大发射功率为 4dBm，接收灵敏度-93dBm，空旷环境下和 iphone4s 可以实现 80 米超远距离通信。它可以与多种蓝牙设备进行通信。模块提供了串行通信接口，可以通过TX和RX引脚与微控制器或其他设备连接。HC-08具有自动重连功能，如果连接丢失，它可以自动重新连接到配对的设备。模块可以进入配对模式，允许用户通过蓝牙设备搜索并配对模块。HC-08可以用于各种项目，如智能家居控制系统、无线遥控设备、数据传输等。可以通过AT命令来编程和配置模块的行为，例如设置模块名称、配对密码等。HC-08模块通常体积小巧，便于集成到各种项目中。

Bluetooth Specification V4.0 BLE，即蓝牙低功耗版本，是专为延长设备电池寿命而设计的通信协议。它允许设备在极低的能耗下进行数据交换，这对于小型便携式设备来说是一个理想的选择，例如健康监测器、智能手表和家居自动化设备。BLE通过快速连接和断开机制，优化了数据传输过程，使得设备能够迅速响应用户操作，同时保持低能耗状态。



HC-08模块因其易用性、成本效益和广泛的应用而受到电子爱好者和专业开发者的欢迎。

其部分主要引脚及其功能如下表所示：

|  |  |
| --- | --- |
| STATE | 连接状态引脚(未连接低电平,连接后输出高电平) |
| RXD | 串口输入引脚（TTL 电平） |
| TXD | 串口输出引脚（TTL 电平） |
| GND | 地 |
| VCC | 电源(1.8-3.6V) |

该协议支持广播模式，允许设备发送小数据包而无需建立全双工通信连接，这有助于进一步降低功耗。安全性是BLE设计的核心，它提供了强大的加密和身份验证机制，确保数据在传输过程中的安全。BLE还支持多角色操作，使得单个设备能够同时作为中心设备和外围设备与其他设备通信，增加了应用的灵活性。

BLE简化了设备间的发现和交互过程，保持了与旧版蓝牙技术的向后兼容性，确保了技术的广泛应用和设备的互操作性。

1. 上位机程序——C#编写

用C#语言编写上位机程序是一种常见的选择，尤其是在需要与硬件设备进行通信和控制的时候。C#作为一种面向对象的高级编程语言，我们使用Microsoft Visual Studio作为开发环境。Visual Studio提供了强大的工具和调试功能，能有效提高开发效率。上位机程序常常需要与硬件设备通过串口（如RS232、RS485等）进行通信。C#提供了System.IO.Ports.SerialPort类，简化了串口通信的实现。上位机程序接收到硬件设备的数据后，通常需要进行解析和处理。C#拥有丰富的库和框架，能够高效地处理和解析各种格式的数据。使用C#语言编写上位机程序具有多方面的优点。首先，C#是一种功能强大且类型安全的面向对象编程语言，提供了丰富的特性，如封装、继承和多态，这使得代码更加模块化和易于维护。其次，C#拥有广泛的.NET框架支持，包括大量的库和API，这为开发各种功能提供了便利，如数据访问、网络通信和图形用户界面设计。

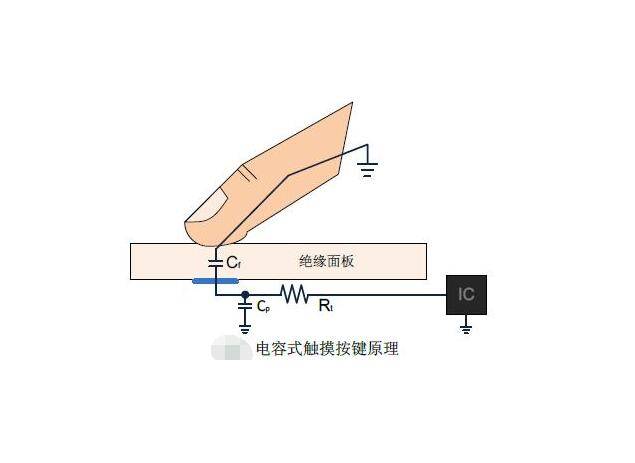
C#的集成开发环境（IDE）Visual Studio提供了强大的开发工具，包括代码编辑器、调试器、设计器和许多其他辅助工具，这些工具可以显著提高开发效率和代码质量。此外，Visual Studio的调试功能非常强大，可以帮助我们快速定位和解决问题。C#语言的跨平台特性也使得编写的上位机程序能够运行在不同的操作系统上，增加了程序的适用性。而且，C#语言的语法清晰，学习曲线相对平缓，对于我们新手开发者来说更加友好。最后，C#社区活跃，拥有大量的教程、文档和第三方库，这为我们在遇到问题时提供了丰富的资源和解决方案。综合这些优点，使用C#编写上位机程序不仅能够提高开发效率，还能够确保程序的质量和可维护性。

## **核心技术**

* **电容触控原理**

电容触控原理基于电容变化的检测。电容是一种存储电荷的能力，而电容触控屏由多层材料构成，其中包括一个导电的触控层，通常覆盖在显示屏的表面。

当用户的手指接触到触控屏时，由于人体是导电的，手指会改变触控屏表面的电容分布。这种变化会导致电荷的重新分布，从而在手指接触点附近形成一个电容值的变化。



电容触控技术的优点包括高灵敏度、快速响应和能够识别多种手势。此外，电容触控屏对环境光和温度变化不敏感，因此具有很好的稳定性。然而，电容屏的一个限制是它们通常不能被非导电物体（如普通笔或手套）操作，这限制了它们在某些特定应用场景中的使用。

## 通信协议——UART串口通信

**1）UART串口通信简介**

UART（Universal Asynchronous Receiver/Transmitter）串口通信是一种通用的异步串行数据总线，用于实现数据的双向通信。，将所需传输的数据一位接一位地传输。在UART通讯协议中，信号线上的状态位高电平代表‘1’，低电平代表‘0’。其特点是通信线路简单，只要一对传输线就可以实现双向通信，大大降低了成本，但传送速度较慢。

UART是一种串行、异步通信总线，可以实现全双工的数据接收和发送。与并行总线相比，UART减少了信号干扰，但数据传输速度相对较慢。UART的工作原理基于异步通信方式，即通信双方不需要共同的时钟信号来同步数据的发送和接收。在UART通信中，数据的传输遵循以下步骤：

发送方在每个字节前发送一个起始位（通常为0），表示数据传输的开始，接着发送数据位（通常为8位），这是实际要传输的数据。然后是可选的校验位（通常为奇校验或偶校验），用于检测数据传输过程中是否出现错误。最后是停止位（通常为1位或2位），表示一个字节的传输结束。

接收方在检测到起始位后，会开始接收数据位，并根据校验位对接收到的数据进行错误检测。如果数据无误，接收方会在下一个字节的起始位到来之前准备好接收下一个字节。

当接收UART检测到起始位时，它将以特定的波特率的频率读取（数据传输速度的度量），以每秒比特数（bps）表示。两个UART必须以大约相同的波特率工作，发送的接收UART之间的波特率只能相差约10%。

**2）工作原理**

数据通信格式如下图3所示，一个UART数据包由10个数据位组成，包含1位起始位，7位有效数据位，1位奇偶校验位和1位停止位。UART串口信号线上空闲时常驻高电平，当检测到低电平下降沿时认为数据传输开始，到停止位时数据传输结束，一共10位数据位组成一个数据包。

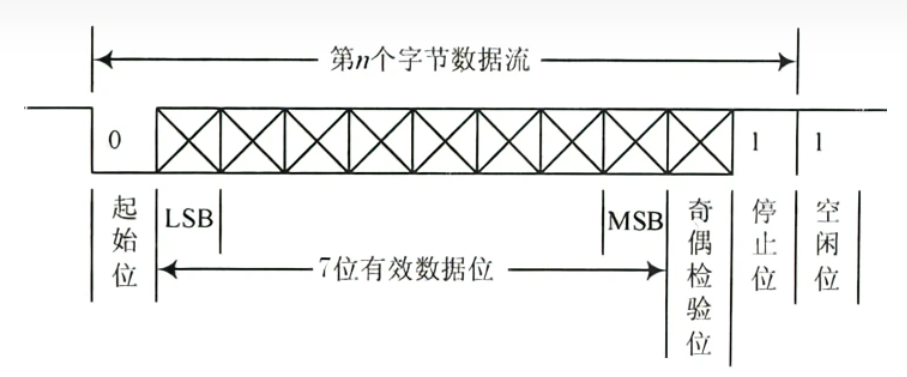


图3 UART数据通信格式

|  |  |
| --- | --- |
| 起始位 | 通信线路上空闲时为“1”，当检测到“0”即下降沿时，认为数据传输开始。 |
| 有效数据位 | 传输开始后传递的需要接收和发送的数据值，可以表示指令或数据。 |
| 奇偶校验位 | 奇偶校验，通过校验传输数据中“1”的个数为奇数个（奇校验）或偶数个（偶校验）来指示传输数据是否正确。 |
| 停止位 | 数据传输结束，传输线恢复常“1”状态。 |

此外，还需关注数据传输波特率，波特率表示一秒内传输了多少个码元数量，一般波特率为300，1200，2400，9600，19200，38400，115200等。例如9600 Baud表示一秒内传输了9600个码元信息，当一个码元只含1 bit信息时，波特率=比特率。我们使用的便是9600Baud。

**3）传输步骤**

1、发送UART（TXD）从数据总线并行接收数据；

2、发送UART（TXD）将起始位、奇偶校验位和停止位添加到数据帧；

3、整个数据包从发送UART（TXD）串行发送到接收UART（RXD），接收UART（RXD）以预先配置的波特率对数据线进行采样；

4、接收UART（RXD）丢弃数据帧中的起始位、奇偶校验位和停止位；

5、接收UART（RXD）将串行数据转换回并行数据，并将其传输到接收端的数据总线。

* **C#编程模拟键盘与鼠标(SendKeys)**

在C#中，可以使用`SendKeys`类和`mouse\_event`函数来实现键盘和鼠标的模拟操作。以下是这两种方法的简介：

**1.SendKeys 类**

SendKeys类是.NET Framework的一部分，用于发送按键到活动窗口。它模拟了键盘输入，可以用于发送单个字符、字符串或特殊键（如箭头键、功能键等）。

\*\*使用 SendKeys 的示例代码\*\*：

public class Program

{

[STAThread]

public static void Main()

{ // 模拟按下字母 'a' 键

SendKeys.Send("a");

// 模拟按下 Enter 键

SendKeys.Send("{ENTER}");

// 模拟按下箭头键

SendKeys.Send("{LEFT}"); // 向左箭头

SendKeys.Send("{RIGHT}"); // 向右箭头

}

}

SendKeys主要用于发送按键到活动窗口，它不会模拟按键的按下和释放动作，而是将按键作为输入发送到当前活动的应用程序。

**2.mouse\_event 函数**

`mouse\_event`函数是一个Windows API函数，用于模拟鼠标的各种动作，包括移动、点击、滚轮滚动等。使用P/Invoke可以将其引入C#程序中。

\*\*使用 mouse\_event 的示例代码\*\*：

```csharp

using System;

using System.Runtime.InteropServices;

public class Program

{

// Windows API 声明

[DllImport("user32.dll")]

private static extern void mouse\_event(uint dwFlags, int dx, int dy, uint dwData, UIntPtr dwExtraInfo);

private const uint MOUSEEVENTF\_LEFTDOWN = 0x02; // 鼠标左键按下

private const uint MOUSEEVENTF\_LEFTUP = 0x04; // 鼠标左键抬起

private const uint MOUSEEVENTF\_MOVE = 0x0001; // 鼠标移动

[STAThread]

public static void Main()

{

// 模拟鼠标移动到屏幕坐标 (100, 100)

mouse\_event(MOUSEEVENTF\_MOVE, 100, 100, 0, UIntPtr.Zero);

// 模拟鼠标左键按下

mouse\_event(MOUSEEVENTF\_LEFTDOWN, 100, 100, 0, UIntPtr.Zero);

// 模拟鼠标左键抬起

mouse\_event(MOUSEEVENTF\_LEFTUP, 100, 100, 0, UIntPtr.Zero);

}

}

mouse\_event函数的dwFlags参数定义了要模拟的鼠标事件类型，`dx`和`dy`参数定义了鼠标移动的相对或绝对坐标，`dwData`通常用于表示鼠标滚轮的滚动量，`dwExtraInfo`是一个额外信息的指针，通常设置为`UIntPtr.Zero`。dx：指定鼠标沿x轴的绝对位置或者从上次鼠标事件产生以来移动的数量，依赖于MOUSEEVENTF\_ABSOLUTE的设置。给出的绝对数据作为鼠标的实际X坐标；给出的相对数据作为移动的mickeys数。一个mickey表示鼠标移动的数量，表明鼠标已经移动。dy：指定鼠标沿y轴的绝对位置或者从上次鼠标事件产生以来移动的数量，依赖于MOUSEEVENTF\_ABSOLUTE的设置。给出的绝对数据作为鼠标的实际y坐标，给出的相对数据作为移动的mickeys数。

# 模块测试

## 触控调试

我们首先研究了触控板的元件手册，这非常重要。因为在研究触控板的设计过程中，我们遭遇了一个意外的挑战。在硬件连接阶段，由于疏忽，我们错误地多连接了一个接口，这导致了后续的代码编写和调试工作陷入了困境。这个错误接口不仅干扰了触控板的正常工作流程，还使得我们花费了大量的时间去排查和定位问题所在。在调试过程中，我们反复检查电路连接，比对设计方案，不断尝试各种可能的解决方案。经过无数次的试验和修正，我们最终确定了问题的根源，并成功排除了这个多余的接口。这次经验虽然给我们带来了不少麻烦，但也让我们更加深入地理解了触控板的硬件结构和工作原理，提升了我们的故障排查和解决能力。

我们特别选择了跑酷游戏作为测试对象，以验证触控键盘的实际效果。在初次试玩时，我们遇到了一些挑战。由于触控键盘的响应速度和游戏内角色的移动速度之间存在差异，导致角色在移动时显得有些卡顿和不流畅。这让我们意识到，需要对代码进行一些修改和优化，以确保游戏的顺畅运行。

经过深入研究了游戏的代码结构，仔细分析了触控键盘的工作原理。通过多次试验和调整，我们最终找到了一种解决方案：在代码中增加适当的延时。这个延时可以让触控键盘的输入信号与游戏角色的移动更加同步，减少卡顿现象的发生。

修改和调试后，我们再次进行了试玩。这次的游戏过程明显比之前流畅了许多。角色在上下左右移动时更加灵活自如，与玩家的触控操作保持高度一致。这不仅让玩家能够更加轻松地掌握游戏节奏，还让他们能够享受到更加真实和沉浸式的游戏体验。

游戏键盘调试好以后，鼠标部分的困惑也迎刃而解。

## UART串口通信蓝牙模块

在调试这部分代码时，我们团队遇到几个问题：

1、配置出错：在将蓝牙模块应用于游戏之前，首先对其进行了相关配置，即连接电脑串口助手，通过AT指令集来设置蓝牙模块的一些相关的信息，如蓝牙名，波特率，连接密码等等。这期间遇到了许多问题，后来发现需要保证蓝牙没有处于通信状态，仅需上电即可。

2、通信失败：无法建立蓝牙连接或无法发送/接收数据。解决办法为：首先要确保蓝牙模块的电源和引脚连接正确。然后，检查是否正确设置了串口通信参数，例如波特率、数据位、停止位和校验位。还可以尝试重新启动蓝牙模块和目标设备，以确保它们之间的通信状态正常。

总的来说，解决UART串口通信蓝牙模块的问题需要耐心和细心。遇到问题时，可以先检查硬件连接和设置，然后逐步排除可能的原因，进行适当的调试和测试。同时，多与其他开发者交流，参考相关资料和文档，也能帮助解决问题。

## 蓝牙调试

根据其通信协议为UART通信协议，应将开发板的RX接蓝牙模块的TX，开发板的TX接蓝牙模块的RX，通信示意图如下图所示：

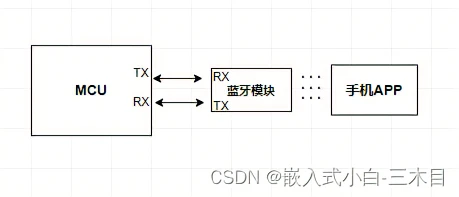


图6 通信示意图

HC-08在通信中充当一个中转站，开发板通过TXD、RXD与模块连接，手机通过蓝牙与模块连接；开发板通过TX、RX发送数据到蓝牙模块，蓝牙模块立即通过蓝牙转发到手机，反之亦然；手机成功与蓝牙模块建立通信后会发送CONNECTED，且此时模块上的指示灯由闪烁变为常亮。

编码过程中，负责蓝牙HC-08模块的同学利用手机和电脑蓝牙助手编程。测试过程中，我们分别测试了电脑端能否接收到正确信号和用手机向接收端发送正确信号时，跟踪观察电脑端操纵的游戏人物能否做出相应的游戏动作。我们着重关注的从数据的准确性，连接的稳定性到借此调试，优化代码。遵循测试原则：遵循测试原则如“尽早测试”、“全面测试”，以提高测试效率和质量。这极大便利了后续系统联调。

## 系统联调模块

在软件开发的过程中，系统联调是一个至关重要的环节。近日，我与同学们在完成各自负责的模块测试后，共同进行了系统联调。在联调过程中，我们遇到了不少预料之外的问题，如模块间数据交互的错误、接口不兼容等。面对这些挑战，我们并没有气馁，而是积极沟通，共同分析原因，寻找解决方案。

经过多次尝试和讨论，我们最终成功解决了这些问题，确保了系统的稳定运行。这次经历让我深刻体会到了系统联调的重要性。它不仅能够帮助我们及时发现并解决潜在的问题，还能提升团队的协作能力和解决问题的能力。在未来的工作中，我将更加重视系统联调这一环节，确保软件的质量和稳定性。

# 课设总结

## 心得体会

基于触控感应键鼠的设计是一个有趣且具有挑战性的硬件课设。下面是一些课设总结及心得体会：这个项目不仅让我深入了解了单片机的应用，也让我在硬件编程和软件设计方面有了更为丰富的实践经验。下面，我将从硬件编程和软件设计两个方面，分享我的一些心得体会。

一、硬件编程方面

1.硬件选型的重要性：

在项目初期，我深刻体会到了选择合适的硬件对于整个项目成功的重要性。单片机型号、蓝牙模块、触控板等每一个部件的选择，都直接影响到项目的进展和最终效果。在选型时，我们不仅要考虑成本，更要考虑部件的性能和兼容性。

2.硬件调试的耐心：

硬件调试是一个既繁琐又需要耐心的过程。在调试过程中，我遇到了许多意想不到的问题，如电路短路、元件损坏等。但正是通过不断的调试和修正，我更加深入地了解了硬件的工作原理和故障排除的方法。

二、软件设计方面

1.编程语言的熟悉：

在软件设计过程中，我选择了C#和verilog语言作为主要的编程语言。通过编写程序，我更加熟悉了语法和特性，也掌握了一些常用的编程算法和数据结构。蓝牙通信是项目中一个关键的环节。在编写蓝牙通信程序时，我深入了解了蓝牙协议的工作原理和通信流程。通过不断的测试和优化，我成功地实现了单片机与蓝牙模块之间的稳定通信。触控功能的实现是项目中另一个难点。在编写触控程序时，我遇到了许多技术难题，如如何准确识别触控点的位置、如何减少误操作等。但正是通过不断的尝试和改进，我们最终实现了稳定可靠的触控功能。

2.系统优化的必要性：

在系统设计中，优化是不可或缺的一环。通过对程序进行优化，我减少了代码的冗余和不必要的计算，提高了系统的运行效率。同时，我也学会了如何合理分配内存和CPU资源，以确保系统的稳定运行。

## 总结

通过本次硬件课程设计，我们成功设计并实现了一款基于触控感应技术的键盘与鼠标二合一设备。该设备不仅具有便携性强的特点，还提供了更为直观、便捷的交互体验。然而，在实际应用中，我们还发现了一些问题和不足之处，如蓝牙连接稳定性、触控精度等方面仍有待提高。未来，我们将进一步优化和改进该设备的设计和实现方案，以满足更多用户的需求和应用场景。同时，我也认识到了团队合作的重要性。在项目的实施过程中，我们面临了硬件选型、电路设计、软件开发等一系列挑战。团队成员通过密切合作，分工明确，各自在硬件编程、FPGA逻辑设计、上位机程序开发等方面发挥了关键作用。特别值得一提的是，在解决蓝牙连接稳定性和触控精度问题上，我们采取了一系列创新的解决方案，显著提升了设备的性能。

个人而言，这个项目不仅加深了我对硬件设计和软件编程的理解，而且锻炼了我的团队合作和问题解决能力。我深刻体会到了硬件编程的耐心和软件设计的细致，也认识到了系统优化在提高设备性能中的重要性。

在未来的学习和工作中，我将继续努力提高自己的能力，为团队做出更大的贡献。总的来说，触控感应键鼠的设计是一个很有挑战性和实践意义的硬件课设。通过这个课设，不仅可以加深对硬件设计和软件编程的理解，还可以锻炼团队合作和问题解决的能力。同时，也能够提高自己的创新思维和实际操作能力。

展望未来，我们计划进一步优化设备的蓝牙连接稳定性和触控精度，以满足更广泛的用户需求和应用场景。最后，我要感谢我的指导教师吴建新老师和所有团队成员的辛勤工作和宝贵建议。

## 参考资料

# TTP226-809SN 8按键触摸检测芯片的详细中文数据手册

Nexys 4 DDR Reference Manua

Nexys4 开发板用户手册

HC-08用户手册V3.3\_2021.03.01