***2019***



**计算机系统能力培养综合实践之**

**蓝牙小车 课程设计报告**

|  |  |
| --- | --- |
| 题 目： | 蓝牙小车 |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS1610 |
| 学 号： | U201614795 |
| 姓 名： | 覃映超 |
| 同组成员： | 孙卓睿 |
|  | 李江山 |
|  | 张 林 |
|  | 龙子吟 |
|  | 张睿欣 |

目 录

[1 课程设计概述 4](#_Toc28901875)

[1.1 课程设计目的 4](#_Toc28901876)

[1.2 设计任务 4](#_Toc28901877)

[1.3 设计要求 5](#_Toc28901878)

[2 系统总体设计 6](#_Toc28901879)

[2.1 硬件部分设计 6](#_Toc28901880)

[2.2 软件部分设计 6](#_Toc28901881)

[3 硬件平台设计与实现 8](#_Toc28901882)

[3.1 设计目的 8](#_Toc28901883)

[3.2 实验步骤 8](#_Toc28901884)

[3.3 故障与调试 9](#_Toc28901885)

[4 操作系统设计与实现 10](#_Toc28901886)

[4.1 设计目的 10](#_Toc28901887)

[4.2 实验步骤 10](#_Toc28901888)

[4.3 故障与调试 16](#_Toc28901889)

[5 应用设计与实现 17](#_Toc28901890)

[5.1 设计目的 17](#_Toc28901891)

[5.2 实验步骤 17](#_Toc28901892)

[5.3 故障与调试 24](#_Toc28901893)

[5.4 存在的不足和缺陷 24](#_Toc28901894)

[6 总结与心得 25](#_Toc28901895)

[6.1 实验总结 25](#_Toc28901896)

[6.2 实验心得 26](#_Toc28901897)

[6.3 意见与建议 26](#_Toc28901898)

[致谢 27](#_Toc28901899)

[参考文献 28](#_Toc28901900)

# 课程设计概述

## 课程设计目的

我国正在实施“中国制造2025”、“互联网+”等重大战略，其目的就在于突破核心关键技术，构筑先发优势，来让我国在未来全球创新生态系统中能占据战略制高点。本次实验的目标就在于通过学生动手实践，学习系统知识，培养出工程实践能力强、创新能力强，具备国际竞争力的高素质复合型人才。

本次课设中，学生需要实现一个MIPS处理器硬件系统和HOs操作系统相结合的应用：智能蓝牙小车，通过实现小车系统，增强学生的系统实践能力，锻炼从底层硬件到应用的系统整体观。

## 设计任务

课设主要分为三个部分：

1. 硬件（MIPSfpga）设计
2. 操作系统（HOS）设计
3. 应用（蓝牙小车）设计。

整体应呈现；硬件-OS-应用的阶梯式发展，由浅入深，首先在硬件设计部分，在Nexys4 DDR开发板上搭建基于MIPS的处理器，并在此基础上同时设计处理器的UART串口、COM 接口、蓝牙设备等；之后进行操作系统部分设计，熟悉小型的操作系统，了解编译和运行的相关原理，根据相关流程熟悉操作系统的具体实现，并通过添加和调用相关应用和系统函数来实现一定的功能；最后进行应用设计，需要基于前两个部分添加相关硬/软件模块实现，搭建出通过蓝牙控制的智能小车。学生应能设计并实现其他额外的、具有自主创新性的新功能。

## 设计要求

1. 硬件部分设计：
2. 基于MIPSfpga的硬件平台搭建及其测试
3. 操作系统部分设计：
4. Hos开发调试环境搭建和使用
5. Hos操作系统的构建与运行
6. Hos操作系统从内核到应用
7. 应用部分设计：
8. 基于MIPSfpga的蓝牙小车硬件设计与实现
9. 蓝牙小车操作系统、驱动及应用软件设计开发
10. 蓝牙小车应用扩展和创新

（4）成果验收和检查答辩；

（5）总结实验过程，完成报告撰写。

# 系统总体设计

## 硬件部分设计

### MIPSfpga处理器

使用Vivado软件，通过IP开发模式搭建一个完整的基于FPGA开发板的MIPSFPGA处理器，同时使用实验提供的包括AXI Interconnect、AXI GPIO、AXI BRAM Controller 和 Block Memory Generator 等部分的IP模块，通过相应的连接处理和硬件设置，最终搭建出物理内存有128MB大小的简单 MIPS 处理器系统，以此为整体系统设计提供硬件基础。

### 蓝牙设备和车辆马达

蓝牙小车的蓝牙部分需要通过串口与处理器进行数据传输，故学生要在之前完成的MIPSFPGA处理器系统中添加相应的UART串口的IP核模块；

小车的车辆驱动马达则通过查阅实验提供的文件中，马达驱动板L293D的资料与 相应的PMOD接口协议原理，了解清楚后，使用四个三十二位寄存器的值，来对智能小车的速度和方向进行精确的调控。

## 软件部分设计

### 蓝牙驱动

智能小车需要使用蓝牙驱动传输数据，故学生需通过控制串口的读写，来进一步控制蓝牙。首先要设置相应寄存器的值，来初始化蓝牙，之后再从寄存器中读取控制数据进行进一步的处理，即可以向智能小车传输数据。

### 马达驱动

智能小车需要使用马达驱动控制速度与方向，这里我们利用蓝牙模块以获取的数据来解码分析，获得数据后再通过每个寄存器的值，控制车轮的正反转向来控制方向，以及车辆马达的01占空比来控制速度，达到驱动车轮行驶的目的。

# 硬件平台设计与实现

## 设计目的

通过学习使用Vivado开发软件，以及了解OpenOCD对简单处理器调试管控的关键性，了解本次课设中设计的MIPSFPGA处理器的编译原理。进一步了解并学习Vivado软件中IP核开发的流程，学习并熟练使用IP的封装和连接的方法。同时也要对处理器中AXI总线结构相应工作原理有所了解，在此基础上构建一个满足需求的、可运行的MIPSfpga处理器。

## 实验步骤

### 硬件平台搭建准备

搭建基于Vivado开发软件、OpenOCD调试工具、MIPS交叉编译环境的完善开发环境、使用putty串口终端的硬件系统。将本次实验中提供的电子流文件烧写到相应的FPGA开发板上，调试并检测其能否运行。

之后再通过已搭建的交叉编译器和GNU make工具来编译开发板主程序，将写好的文件下载到已烧写好FPGA开发板上，观察其能否运行，以及运行后的结果是否符合预期。

### 基于MIPSFPGA的简单平台搭建

在之前的基础上，实际构建一个FPGA开发板上的的跑马灯程序，并进行相应的测试。通过Vivado软件搭建使用 AXI 总线的 MIPSfpga简单硬件平台，之后编写相应的MIPS汇编程序或其他语言程序对其进行测试。

### 中断模块

在完成MIPSfpga处理器硬件的搭建之后，进一步实现 AXI 总线外设中断的处理。这一步要在MIPSfpga处理器中将UART串口和PWM的中断数据，通过相应设置连接到CPU的中断入口，再在C程序和MIPS汇编程序中进一步编写中断的服务代码，以此来对已搭建好的MIPSfpga硬件平台的中断部分进行相应测试。

## 故障与调试

组内成员在使用Vivado软件尝试打开内存接口IP核，查看其配置时，出现了Vivado软件闪退或自动崩溃，无法查看配置界面的情况。在上网查阅故障处理和相应情况后，尝试重装或者用管理员模式打开软件后，依然存在该情况，无法对IP核进行配置，组内成员商议后通过更换电脑再次进行相应实验，解决了这一问题，对搭建好的MIPSFPGA处理器进行了相应配置。

# 操作系统设计与实现

## 设计目的

在这一环节中，学生应学习在个人电脑上构建（build）Hos操作系统的环境的相应方法，并将生成的相应镜像下载到MIPSfpga开发板上，同时令其可以正常运行；更进一步的，学生要能通过VScode软件，阅读并熟悉对应的Hos操作系统源码，熟悉基本工作原理，并能在此基础上对源码进行调试，完成实验要求的预期目标。

## 实验步骤

### Hos操作系统构建运行

首先要安装构建Hos操作系统所必需的工具，这其中主要包括：Cygwin、交叉编译器（mips-sde）、Putty、OpenOCD 等。其中已有部分安装过，此处不再赘述。

Cygwin是一个可以在Window环境下运行的可提供Linux环境的工具。我们首先运行Cygwin，此时需要进入Hos操作系统源代码所在目录，之后在当前目录下make进行编译。make完成后，我们将构建完的操作系统运行在之前实现的 MIPSfpga 上。此处我们为了和调试的run.bat文件区别开来，要使用 runo.bat 文件，该文件早已完成运行操作系统所需要的所有程序编辑，所以我们不需要进行其他操作，直接在当前目录下运行runo.bat文件即可，如图4.1所示。

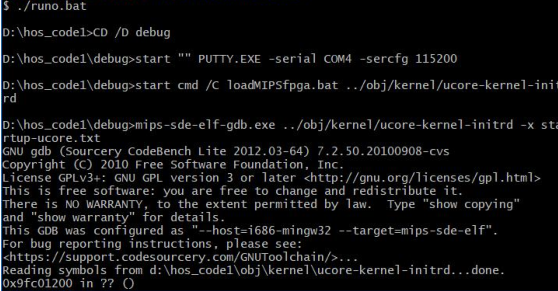


图 4.1 首先运行runo.bat

我们可以看到，运行成功runo.bat文件后，Putty程序窗口中的输出将会如图4.2 所示。这种结果说明了Hos操作系统已经成功地在MIPSfpga上运行着。

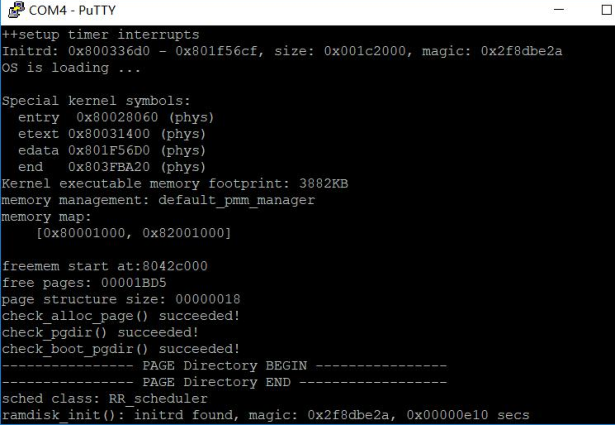


图 4.2 运行完的程序输出

这里我们因为已经运行成功，可以选择执行ls命令观察数据，运行命令后的结果如图4.3所示。

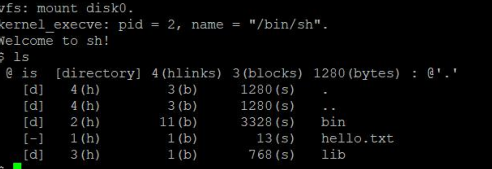


图 4.3 执行ls指令后结果

### Hos中添加应用

（1）添加系统调用

这里我们需要根据实验要求，在Hos操作系统中添加系统内核调用。我们需要在 kern-ucore\include\lib\unistd.h这一文件中加入具体的系统调用编号，之后才能通过调用号调用需要的函数。

之后，我们再在kern-ucore\syscall.c这一文件中添加hello函数和free函数，具体添加内容如图4.4和图4.5所示。

添加完毕具体函数后，我们还需要再在uint32\_t(\*syscalls[])这一函数中添加函数入口，这样才能在后续调用时避免出现错误。



图 4.4 sys\_free函数



图 4.5 sys\_hello函数

这两个函数中，除简单的hello函数外，我们还需要编写sys\_free，通过调用系统内核代码，进而获取此时的空闲页数，并且可以把当前空闲页数格式转换为可以被正常输出的阿拉伯数字，以此满足实验要求

（2）添加用户函数

添加完毕系统调用后，我们再添加用户调用函数，让用户也可以正常调用所需函数。这一步需要我们在user\include\unistd.h这一文件中先添加上系统调用编号，保证可以被调用号调用，这一步与上一部分类似。

之后，我们需要在user\include\syscall.h和user\syscall.c这两个文件中再添加上函数声明和具体的函数内容，这一部分如图 4.6 和图 4.7 所示。



图 4.6 添加具体函数声明

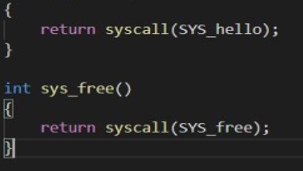


图 4.7 添加具体函数体

（3）添加用户程序

添加完毕用户函数后，我们需要添加具体的用户程序。这一步先在user\user-ucore目录下新建hello.c文件，以及free.c文件，包含我们所需的程序的函数体，这两个函数的具体代码如图4.8和图4.9所示。

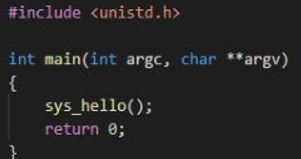


图 4.8 hello函数内容

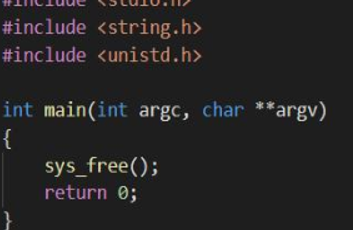


图 4.9 free函数内容

之后，我们再在user\user-ucore\Makefile修改make文件，我们需要在make文件中添加上我们新加入的程序名称，即hello和free，具体如图 4.10 所示。

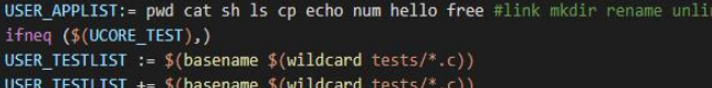


图 4.10 加入具体程序名

## 故障与调试

在这一步中，有几名组员在构建镜像上遇到了同样的问题，即make编译时产生错误提示，警告未找到命令：mips-sde-elf-gcc。询问其他组后，发现有许多人也有这个问题。在询问老师后，得知是环境包没有正确安装，系统变量没有设置好导致。如图4.11所示。按照老师指导正确设置系统变量中路径名解决了这一问题，可以正常产生系统镜像。

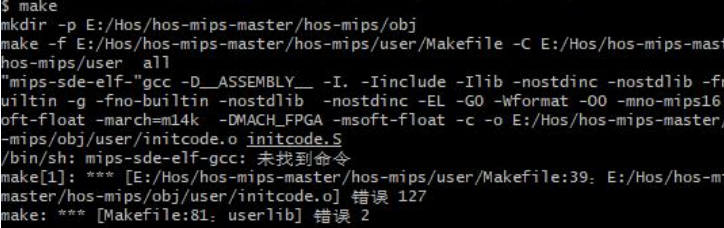


图 4.11 构建镜像时报错

# 应用设计与实现

## 设计目的

本次实验主要是为了实现通过UART串口协议搭建的AXI总线蓝牙接口模块，同时实现通过PMOD接口协议和L293D驱动板原理搭建的AXI总线马达驱动模块，并把这两个外部设备添加到MIPSfpga处理器上，通过对接口进行设置和布线，进而烧写到开发板N4DDR上，达成实验目标，同时使用交叉编译器编写程序后，利用MIPSfpga处理器验证是否添加正确，以及外设模块是否工作正常。

为了更好地完成工作，学生应熟悉蓝牙芯片的基本硬件信息和L293D驱动板的工作原理，这样可以精确无误地设计并实现两个模块的主要功能。

## 实验步骤

### 蓝牙模块

该模块是基于串口方式工作，可以直接使用Vivado提供的UART串口IP核为基础。在硬件平台中添加一个新的UART IP核，并将线路连接好。如图 5‑1 蓝牙UART模块所示。需要注意UART的设备中断输出要连接到硬件中断源整合模块后再接入硬件中断控制器，并且只有ln0-ln5有效。

在完成添加硬件模块后，需要在操作系统中添加串口中断程序，在内核头文件 arch.h 中注册蓝牙串口硬件模块的分配的地址，完善中断处理例程入口函数的中断分发部分的逻辑，将串口中断处理例程注册进去，完成后在 trap 中完善内核态对中断的处理。另外一个需要做的工作就是串口的初始化，初始化的工作最主要的是打开系统对串口中断的响应，通过指明中断号来调用 pic\_enable 函数完成。

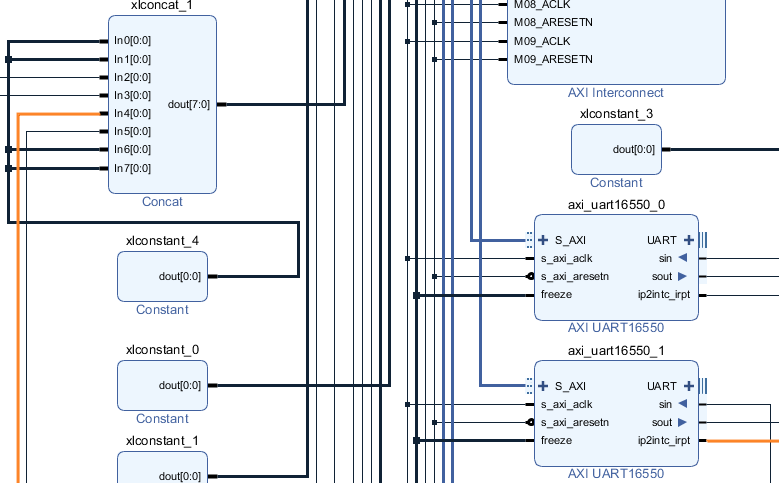


图 5‑1 蓝牙UART模块

在具体的软件功能实现时，首先要定义对应于中断控制器的硬件中断源，如图 5‑2 硬件中断源所示，便于利用中断控制器中的IER寄存器实现开启/关闭不同的硬件中断源。此外还需要定义硬件中断源对应的CPU中断号以及硬件地址，如图 5‑3 硬件地址以及中断号所示，使得系统能够对串口中断响应。

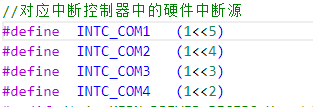


图 5‑2 硬件中断源

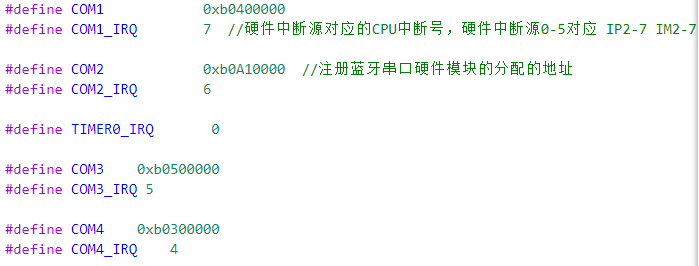


图 5‑3 硬件地址以及中断号

蓝牙中断处理程序的主要功能为根据接收的数据进行分析从而执行不同的操作，其主要代码如图 5‑4 蓝牙中断处理函数所示。

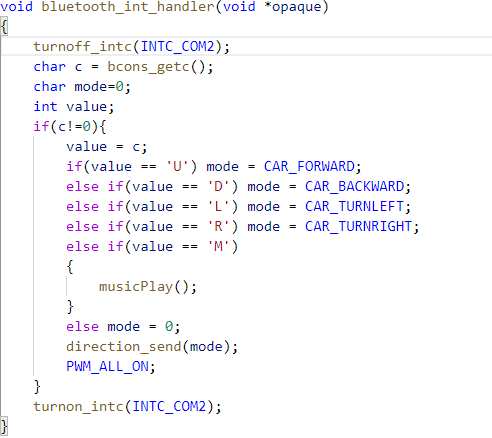


图 5‑4 蓝牙中断处理函数

### 马达驱动模块

本次实验所用的小车使用的是L293D驱动板驱动。该驱动板的输入信号有很多，如图 5‑5 L293D 驱动板 PCB 封装引脚功能图所示，看起来很复杂，但是将板上芯片单独来看就比较清晰。首先，板子上有两块 L293D 芯片，这种芯片常被用来控制电机的转向和转速，每块电机最多可同时控制两个电机，所以两块芯片正好对应着小车的四个驱动电机。L293D 芯片主要的输入有控制信号和 PWM 调速信号。其中每个电机的控制信号有两根，这两根信号的高低电平组合构成小车的正转、反转和停止。

而实际上，L293D 的转向控制信号并不需要我们直接从开发板上接过去。在驱动板上，这些输入已经连接好了，连接在 74HCT595N 芯片的输出上。74HCT595N 芯 片，如图 5‑6 芯片原理图所示，内部由一个移位寄存器、一个锁存器和一个三态输出构成。而我们对小车方向控制的 8 位信号，正是存放在这里。这8位控制信号都是高电平有效，并且需要自己测试某一位对应的控制信号是什么（例如第0位为高电平对应1号马达顺时针旋转等）。

所以我们要实现的接口主要相当于一个串行通信的东西，把传来的8位并行控制位按串行的方式写到74HCT595N 芯片的移位寄存器中，然后Dir\_latch给个上升沿把数据锁存到锁存器中，最后Dir\_enable给低电平，使能三态门，一次控制就算完成了，直到下次控制写入之前小车都会按照上次写的信号运转。这一部分可以采用硬件实现也可以采用软件实现。采用硬件方式实现时，需要使用状态机；软件实现时，只需要将8位信号通过GPIO串口一位一位输出到驱动板上。

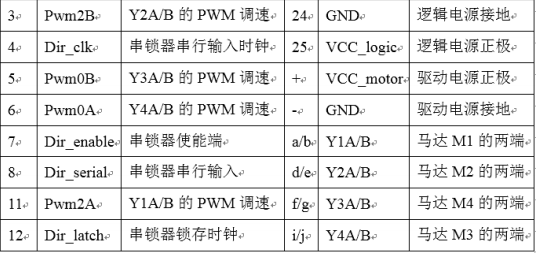


图 5‑5 L293D 驱动板 PCB 封装引脚功能图

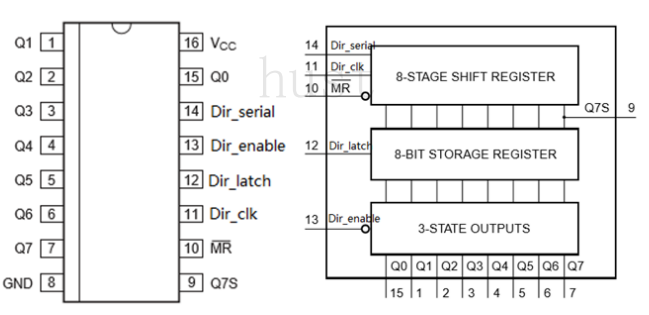


图 5‑6 芯片原理图

### 蜂鸣器模块

通过输出不同频率的脉冲信号到无源蜂鸣器的IO口，就可以发出不同的音节。不同音节对应的频率如图 5‑7 音节频率图所示。为了让蜂鸣器控制信号产生不同的频率，使用一个分频模块实现（可直接修改PWM模块，使其能够产生方波，即占空比为50%的矩形波），不同的音节控制对应不同的分频系数。其转换公式为：

(（系统时钟频率/音节频率）/ 2 ) - 1;



图 5‑7 音节频率图

本次实验采用的歌曲为《always with me》,根据其C调曲谱翻译出音节对应的频率，用来初始化一个音节数组，如图 5‑8 歌曲音节数组所示。

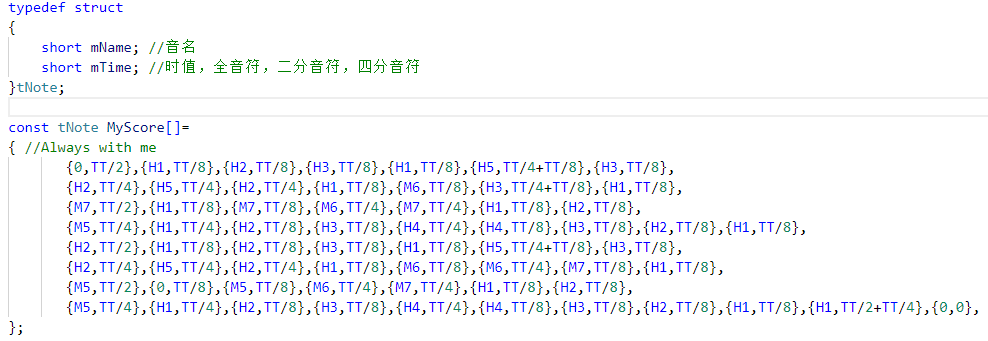


图 5‑8 歌曲音节数组

再通过一个循环遍历数组中的每一个音节，根据音节计算出对应的分频系数，最后再将分频系数传输给分频硬件模块。具体实现如图 5‑9 蜂鸣器处理函数所示。音乐播放模块最好应该通过创建一个新线程来运行，因为在循环遍历数组期间无法进行别的操作，但是技术条件有限就没有实现。



图 5‑9 蜂鸣器处理函数

### 红外线模块

利用红外线模块实现一个预防碰撞辅助功能，当红外线模块检测到前方有障碍时，自动停车。这个模块也采用中断触发的方式，当红外模块检测到障碍时，out输出接口输出低电平，所以只需要在中断处理程序中判断是否为低电平即可。其中断处理函数如图 5‑10 红外模块中断处理函数所示。

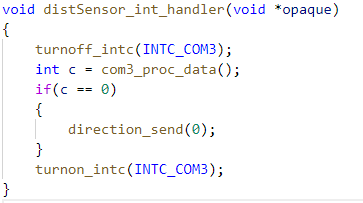


图 5‑10 红外模块中断处理函数

本实验采用的红外模块接口说明如图 5‑11 红外模块接口说明所示。



图 5‑11 红外模块接口说明

## 故障与调试

设备中断信号的输出没有接到有效的位上，导致中断无法响应，查看了MIPSCPU文档后才知道只有0-5位有效，并且硬件中断源0-5对应CPU中的IP2-7、IM2-7。

## 存在的不足和缺陷

1. 蜂鸣器模块与蓝牙模块无法并行执行，缺少多线程。
2. 除了基本功能之外新添加的模块都比较简单。

# 总结与心得

## 实验总结

本次系统能力综合培养中，智能蓝牙小车的设计重点在于硬件和软件相结合，并在其基础上做出创新性的应用。实验难度对于没有选修过接口技术课程的同学来说有些稍高，但对于锻炼学生的系统实践能力，拓宽学生从底层硬件到应用的系统整体观很有帮助。

本次实验首先要从搭建基本的MIPSfpga处理器开始，通过添加中断处理、添加蓝牙接口、马达驱动等部分，最终实现一个功能完善、可正常运行的智能小车。实验的难点主要在于MIPS处理器和蓝牙等外部接口的信息传输，以及小车的转向和速度控制。

对于前者，考虑到蓝牙设备使用UART串口模块，我们可以据此来尝试收发数据，由MIPS处理器向马达驱动模块写入相关值，来达到实验目标；对于后者，我们组在看过L293D驱动板封装的引脚功能图后，分别对控制信号和PWM调速信号进行调试，由占空比控制速度，小车车轮的正反转向控制方向，通过控制设备中断信号来达到实验目标。整体实验任务虽然有难度，但群策群力下也得以结局，最后完成了智能蓝牙小车的基本功能实现，且有余力进行了蜂鸣器、红外传感等额外的创新功能模块的设计和实现。

整体而言，本次实验虽有难度，但却能很好地锻炼学生将知识和实践相结合，在实际工作中培养自己的工程能力，可以说对学生提升个人能力极有益处。

## 实验心得

首先，本次实验硬件设计和软件设计都有一定难度，需要学生仔细阅读实验资料才能有一定头绪，很考验学生的自主学习和总结能力；其次，遇到问题时要学会在组内多讨论、团队协作，有时就可以有不一样的思路，有益于解决问题，如果实在解决不了，也可以询问其他组的同学或老师，不能轻言放弃；最后，我国正在实施“中国制造2025”、“互联网+”等重大战略，无论是课上还是课下，我们都应响应国家号召，好好学习课程知识，培养动手实践能力，通过像系统能力综合培养这样具有深刻意义的课程，锻炼自己，让自己成为具备竞争力的复合型人才。

## 意见与建议

本次课程设计令我获益匪浅，但在初始上手时有些没有头绪，尤其对于没有接触过接口技术等课程的学生而言，最后的蓝牙接口和马达驱动部分难度不小，建议以后的课程中可以适当加入引导，多一些实验指导资料，每个小组分配一两名有接口技术基础的学生，帮助全组更好地完成实验目标，更好地提升个人能力。

# 致谢

首先感谢老师对我们的教导，指导我们实验的重点在哪里，实验时遇到的问题要如何克服，对我们完成实验很有帮助；然后要感谢我的组员们，正因为有大家群策群力，才可以解决问题，顺利完成实验目标。

最后特别感谢 Imagination Community 对本次实验中的支持和贡献，正因为有这些业界公司无私的教学支持才能让我们这些学生在学习中走的更远。

# 参考文献

[1]. 秦磊华，吴非，莫正坤.计算机组成原理. 北京：清华大学出版社，2011 年.

[2]. 袁春风编著. 计算机组成与系统结构. 北京：清华大学出版社，2011 年.

[3]. 张晨曦，王志英. 计算机系统结构. 高等教育出版社，2008 年.

[4]. 周湘贞. 操作系统原理与实践教程. 北京:清华大学出版社，2006.

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  特此声明！  **作者签字:** |
| 二、对课程实验的学术评语（教师填写） |
|  |
| 三、对课程设计的评分（教师填写） |
| |  |  |  |  | | --- | --- | --- | --- | | 评分项目  （分值） | 报告撰写  （30分） | 课设过程  （70分） | 最终评定  （100分） | | 得分 |  |  |  | |
| **指导教师签字: 2019-12-09** |