***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： |  |

目 录

[1 课程设计概述 3](#_Toc28852051)

[1.1 课程设计目的 3](#_Toc28852052)

[1.2 设计任务 3](#_Toc28852053)

[1.3 设计要求 3](#_Toc28852054)

[2 系统总体设计 5](#_Toc28852055)

[2.1 硬件部分设计 5](#_Toc28852056)

[2.2 软件部分设计 5](#_Toc28852057)

[3 硬件平台设计与实现 6](#_Toc28852058)

[3.1 设计目的 6](#_Toc28852059)

[3.2 实验步骤 6](#_Toc28852060)

[4 操作系统设计与实现 8](#_Toc28852061)

[4.1 设计目的 8](#_Toc28852062)

[4.2 实验步骤 8](#_Toc28852063)

[4.3 故障与调试 13](#_Toc28852064)

[5 应用设计与实现 15](#_Toc28852065)

[5.1 设计目的 15](#_Toc28852066)

[5.2 实验步骤 15](#_Toc28852067)

[5.3 故障与调试 19](#_Toc28852068)

[5.4 存在的不足和缺陷 19](#_Toc28852069)

[6 总结与心得 20](#_Toc28852070)

[6.1 实验总结 20](#_Toc28852071)

[6.2 实验心得 20](#_Toc28852072)

[6.3 意见与建议 21](#_Toc28852073)

[致谢 22](#_Toc28852074)

[参考文献 23](#_Toc28852075)

# 课程设计概述

## 课程设计目的

实现一个 MIPSfpga 处理器硬件系统与 Hos 操作系统综合的应用蓝牙小车，通过整个系统的实现，增强学生的系统能力，从底层硬件到应用的系统整体观。

学习、了解有关 AXI 总线结构的基本工作原理，设计 AXI 接口外设和可支持操作系统的 MIPSfpga 和自定制 PMOD 模块；在开发的 MIPSfpga 上运行小型的 Hos操作系统，并在小型操作系统上进行应用层面的开发，完成相关的任务。

## 设计任务

主要分为三个部分：硬件（MIPSfpga 处理器）设计，操作系统（Hos）设计，应用（蓝牙小车）设计。

硬件设计部分要求在 Nexys4 DDR FPGA 开发板上搭建一个基于 MIPS 的处理器（软核），并在此基础上设计处理器的外围设备，包括 UART、COM 接口、蓝牙设备等；操作系统设计在清华大学的 ucore 操作系统改造简化而来，熟悉该小型操作系统，了解编译和运行的相关原理，根据源代码和相关编译的流程了解操作系统的具体实现，通过添加相关应用和系统调用实现一定的功能；应用设计基于前两个部分，通过添加相关硬件软件模块，完成一个蓝牙控制的小车，并实现其他额外的功能。

## 设计要求

（1）基于 MIPSfpga 的硬件平台搭建及其测试

（2）Hos 开发调试环境搭建和使用

（3）Hos 操作系统的构建与运行

（4）Hos 操作系统应用开发和系统调用

（5）基于 MIPSfpga 平台的蓝牙小车硬件设计与实现

（6）蓝牙小车操作系统、驱动及应用软件设计开发

（7）蓝牙小车应用扩展和创新

（8）成果验收和检查

（9）总结实践过程，完成实验报告撰写

# 系统总体设计

## 硬件部分设计

### MIPSFPGA处理器

利用 Vivado 平台提供的 IP 模块进行基于 AXI 接口的 MIPSfpga 处理器搭建，其中 IP 模块包括 MIPS MicroAptiv UP、AHB-Lite to AXI Bridge、AXI Interconnect、AXI GPIO、AXI BRAM Controller 和 Block Memory Generator 等，进行相应的连接和设置，可以搭建出一个时钟频率为 50MHZ、物理内存有 128MB 的简单 MIPS 处理器系统，为后续设计接口提供硬件基础。

### 蓝牙和小车马达

蓝牙外设与处理器进行数据传输是通过串口，所以需要在前述处理器系统中添加 UART 的 IP 核模块即可。小车马达通过查阅马达驱动板 L293D 的资料与 PMOD 接口协议原理后，则采用 4 个 32 位寄存器的值进行速度和方向的控制。

## 软件部分设计

### 蓝牙驱动

蓝牙驱动模块通过控制串口的读写进行蓝牙的控制，首先通过设置相应寄存器的值初始化蓝牙，然后从寄存器中读取相应数据进行处理即可。

### 马达驱动

马达驱动则是利用蓝牙模块中获取的数据进行解码分析，然后通过每个寄存器的值控制每个正反转及其马达的 01 占空比即电压，实现驱动轮子的功能。

# 硬件平台设计与实现

## 设计目的

Vivado 开发环境的使用，学习了解 Openocd 和其 JTAG 对 MIPSFPGA 硬件平台调试管理的工作原理，了解 MIPS 交叉编译环境提供的不同机器平台源码到MIPS 机器的编译原理。学习、理解、实践 Vivado IP 开发的流程，掌握 IP 封装、使用、连接的基本方法。同时学习、了解有关 AXI 总线结构的基本工作原理，同时完成一个基本可运行的 MIPSfpga。

## 实验步骤

### 硬件平台搭建准备

搭建基于 Vivado 硬件开发环境、Openocd 嵌入式调试工具、MIPS mti 交叉编译环境、GNU make 工具的开发环境、putty 串口终端的 MIPSfpga 硬件系统。将提供的.bit 文件通过烧写到 FPGA 开发板，并观察其运行。通过搭建的交叉编译器和make 工具编译裸机主程序，将二进制文件下载到烧写好.bit 文件的 FPGA 开发板并观察其运行。

### 基于MIPSFPGA的简单平台搭建

通过动手构建一个由 Vivado IP 开发方式实现 FPGA 上的跑马灯，并对其进行测试。通过 Vivado IP 开发的模式搭建一个使用 AXI 总线及外设模块的 MIPSfpga简单硬件平台，并编写 MIPS 汇编程序或 C 语言程序对其进行测试。

### 中断模块

在简单 MIPSfpga 处理器硬件平台上实现 AXI 总线外设中断处理。在MIPSfpga 中将 uart 和 PWM 的中断信息连接到 CPU 的中断入口，然后在 C 和MIPS 汇编程序中编写中断服务代码，对 MIPSfpga 硬件平台的硬件中断进行测试。

# 操作系统设计与实现

## 设计目的

学习在自己的个人电脑上安装构建（build）Hos 操作系统的环境，将所生成的镜像下载到 MIPSfpga 开发板、并将其运行起来的方法；安装VScode，能对Hos操作系统的源代码进行阅读；熟悉操作系统的原理，能够完成一定的操作系统源代码的阅读、修改，达到所要求的预期功能。

## 实验步骤

### HOS操作系统构建运行

安装 Hos 操作系统的构建涉及到的软件工具：Cygwin、交叉编译器（mips-sde）、Putty、Vivado、OpenOCD 等。

运行 Cygwin，并进入 Hos 源代码所在的目录，然后进行 make。如图 4.1 所示。



图 4.1 Hos 操作系统目录下 make

make 完成后，将构建好的操作系统在之前实现的 MIPSfpga 上运行。为了与调试信息的 run.bat 文件进行区别，则使用 runo.bat 文件，其已经完成了对运行操作系统所需要的所有动作的编辑，所以在当前目录下直接运行 runo.bat 文件，如图 4.2所示。

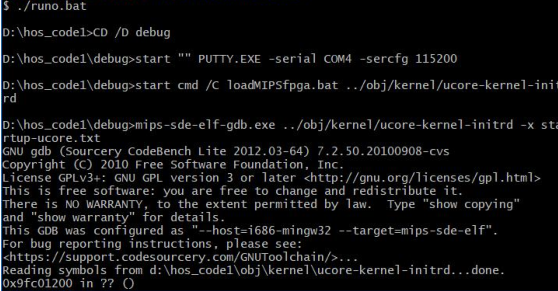


图 4.2 运行 runo.bat

在运行成功之后，可观察 Putty 的窗口中的输出如图 4.3 所示。此时说明 Hos已经成功运行在 MIPSfpga 上。

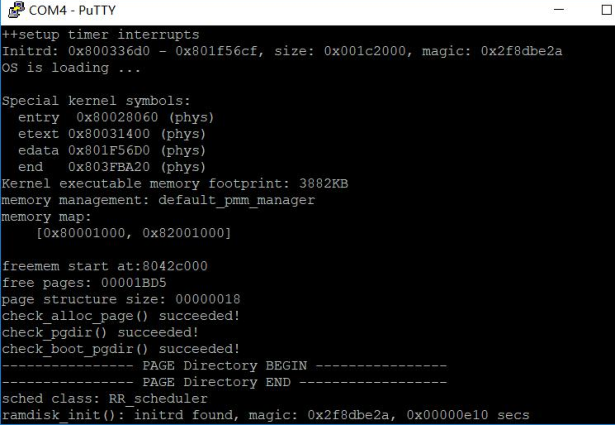


图 4.3 运行 Hos 之后 Putty 窗口的输出

可以执行 ls 命令查看系统中的数据，如图 4.4 所示。

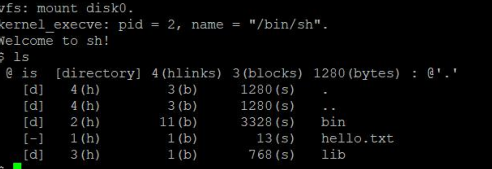


图 4.4 系统中执行 ls

### HOS中添加应用

（1）添加系统内核调用

根据实验要求，添加系统内核调用，首先在 kern-ucore\include\lib\unistd.h 添加系统调用编号，如图 4.5 所示。

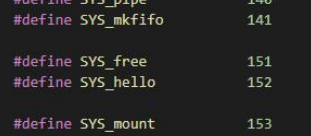


图 4.5 添加系统调用编号

在kern-ucore\syscall.c添加函数sys\_hello和sys\_free，如图4.6和图4.7所示；然后在uint32\_t(\*syscalls[])(uint32\_t arg[])函数中添加函数入口。



图 4.6 添加 sys\_hello 函数



图 4.7 添加 sys\_free 函数

其中 sys\_free 是通过阅读系统内核代码来获取相应的空闲页数并将该数字转换成可以输出的数字。

（2）添加用户调用

然后添加用户调用。在 user\include\unistd.h 添加系统调用编号，如图 4.8 所示。

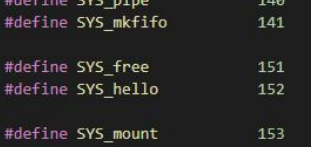


图 4.8 添加用户调用

在 user\include\syscall.h 添加函数声明，如所示；在 user\syscall.c 中添加函数，如图 4.9 和图 4.10 所示。

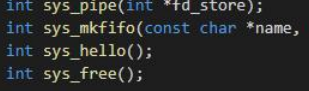


图 4.9 添加函数声明

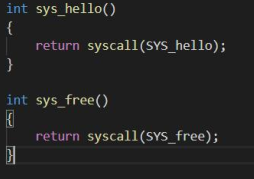


图 4.10 添加函数 sys\_free 和 sys\_hello

（3）添加用户可执行程序

在 user\user-ucore 文件夹下新建文件 hello.c，代码如图 4.11 所示；新建文件free.c，代码如图 4.12 所示。

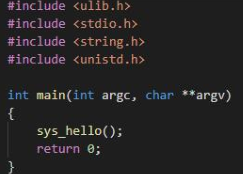


图 4.11 hello.c 文件内容

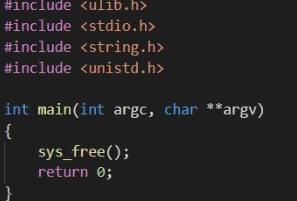


图 4.12 free.c 文件内容

修改文件在 user\user-ucore\Makefile，在 USER\_APPLIST 中添加新的应用名称hello 和 free，如图 4.13 所示。

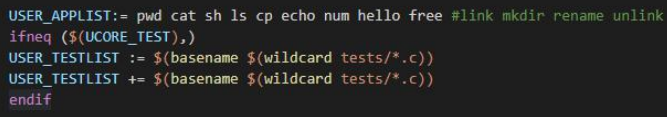


图 4.13 添加应用名称

## 故障与调试

在构建 Hos 镜像的时候遇到如图 4.14 的错误，根据错误提示，是表明缺少相应的 mips-sde-elf-gcc 命令，则经过询问老师，得到相应的环境包，设置好系统变量后错误消除。

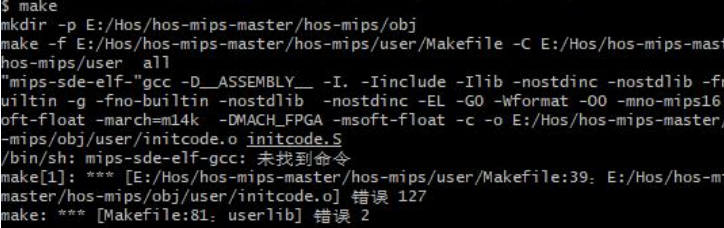


图 4.14 make 出错

# 应用设计与实现

## 设计目的

设计并实现以 UART 串口协议为基础的蓝牙芯片总线外设 AXI 总线接口模块。 设计并实现以 PMOD 接口协议为基础的 L293D 驱动板总线外设 AXI 总线接口模块。将这 2 个外设接口添加到 MIPSfpga 上，进行综合布线，并烧写到 N4ddr 开发板上。利用 MIPS sde 交叉编译器编写一个程序下载到 MIPSfpga 中验证，并测试以上两个模块的正确性。

了解关于无线蓝牙外设的基本工作原理和 L293D 驱动板基本工作原理，以便更好的实现其外设接口的设计。

## 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。

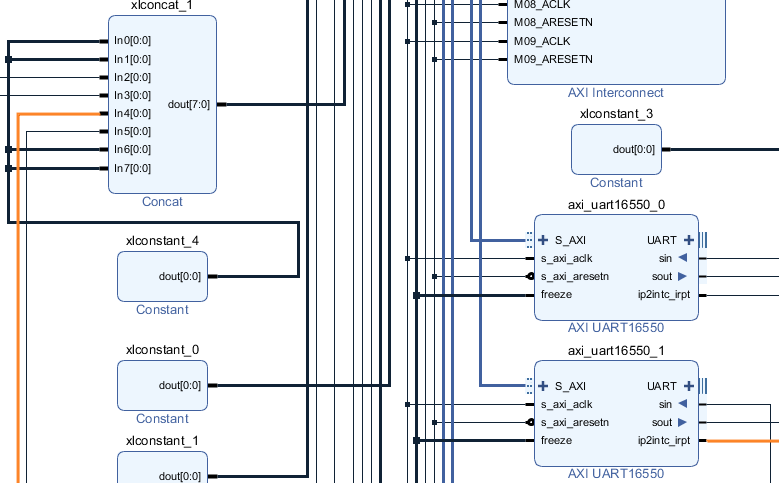


图 5‑1 蓝牙UART模块

### 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 5‑2 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 5‑3 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。

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。

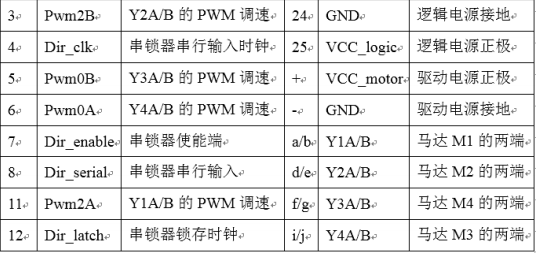


图 5‑2 L293D 驱动板 PCB 封装引脚功能图

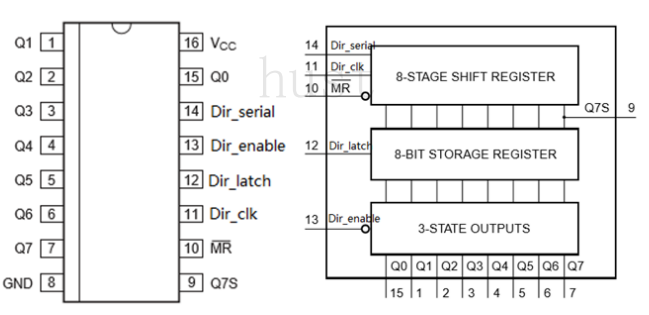


图 5‑3 芯片原理图

### 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 5‑4 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 5‑4 音节频率图

### 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。

本实验采用的红外模块接口说明如图 5‑5 红外模块接口说明所示。



图 5‑5 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

本次蓝牙小车的设计是硬件和软件的结合，其要求我们自己基于 Vivado 平台搭建一个 MIPSfpga 处理器，然后在该处理器的基础上，运行一个微型操作系统，然后在这个操作系统基础上进行应用的设计，通过自己添加的外设接口来控制小车运转。其中涉及到了 CPU 的基本架构知识，操作系统的基本理论，接口方面的设计以及通信协议的使用。

该实验首先从搭建基本的处理器开始，再在这个基础上进行中断控制的添加，然后进行蓝牙外设的控制与操纵，其中碰到几个关键的技术难点如下：处理器与外设之间的交互，操作系统中添加小车的控制模块。

处理器与外设之间的交互是指搭建的 CPU 与蓝牙或者马达驱动之间的数据传输。蓝牙部分是用 UART 模块，所以可以根据 UART 模块接口来进行数据传输。在马达驱动则需要用户自定义的 IP 模块封装，也较为好的是，只需要处理器向马达驱动写值，而不用从马达驱动获取数据到处理器内，也就是说单向往外写，所以该问题也不算难。

操作系统中添加小车的控制模块，我们最终的完成品是通过中断控制的，也可以说是半成品，因为其最好的方法还是在操作系统里面添加设备驱动，这我们经过尝试，但由于对其中的原理了解的不够好，所以没有完成该技术的实现。

## 实验心得

1. 从硬件部分设计到软件接口设计，从 MIPSfpga 处理器再到 Hos 操作系统，每一步都是不可轻视的。

2. 团队之间的协作与帮助很重要，要合理安排每个人的工作。

3. 遇到问题要勇于尝试，不要轻言放弃，可以各种思路去试试。

4. 对操作系统的原理和接口的了解得还不够，多了解一些可能做起来会轻松许

多。

## 意见与建议

本次课程给我带来的收获是不小的，而在对于本课程的建议就是，前面部分分值可以适当放低些，后面扩展部分可以适当增加分值。然后整个小组最好不超过 6人，这样在分工上也会更加合理些。

# 致谢

首先感谢我的组员们，正是因为有他们跟我一起努力，才能完成我们自己的初步目标，才能让我在后面不断失败的时候没有轻言放弃。

再来就是感谢老师和助教，正是他们的辛勤指导和鼓励，我们才可以更好更快地完成我们自己任务，他们在我们困难的时候给的很多指导和建议都对项目的进程有极大的帮助。

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: 2018-12-17** |