***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： | 自动避障蓝牙可变速小车 |
| 专 业： | 计算机科学与技术 |
| 班 级： | ACM1601 |
| 学 号： | U201614644 |
| 姓 名： | 曾康 |
| 同组成员： | 郑汉清 孙卓正 杨子灿 |

目 录

[1 课程设计概述 3](#_Toc29153489)

[1.1 课程设计目的 3](#_Toc29153490)

[1.2 设计任务 4](#_Toc29153491)

[1.3 设计要求 4](#_Toc29153492)

[2 系统总体设计 5](#_Toc29153493)

[2.1 设计目的 5](#_Toc29153494)

[2.2 硬件部分设计 5](#_Toc29153495)

[2.3 软件部分设计 6](#_Toc29153496)

[3 硬件平台设计与实现 7](#_Toc29153497)

[3.1 设计目的 7](#_Toc29153498)

[3.2 实验步骤 7](#_Toc29153499)

[3.3 故障与调试 8](#_Toc29153500)

[3.4 存在的不足和缺陷 8](#_Toc29153501)

[4 操作系统设计与实现 9](#_Toc29153502)

[4.1 设计目的 9](#_Toc29153503)

[4.2 实验步骤 9](#_Toc29153504)

[5 应用设计与实现 14](#_Toc29153505)

[5.1 设计目的 14](#_Toc29153506)

[5.2 实验步骤 14](#_Toc29153507)

[5.3 故障与调试 19](#_Toc29153508)

[5.4 存在的不足和缺陷 19](#_Toc29153509)

[6 总结与心得 20](#_Toc29153510)

[6.1 实验总结 20](#_Toc29153511)

[6.2 实验心得 20](#_Toc29153512)

[6.3 意见与建议 20](#_Toc29153513)

[致谢 21](#_Toc29153514)

[参考文献 22](#_Toc29153515)

# 课程设计概述

## 课程设计目的

《系统能力培养综合实践之蓝牙小车》的具体目标包括：

目标1:基于FPGA开发板能设计满足设计任务要求的蓝牙小车，通过所设计系统在FPGA资源消耗、系统的性能、功能完备、操作的便利、存在的问题等因素的分析，评价解决方案可能存在的对社会和环境等方面的影响，并对设计方案进行优化。

目标2:能根据功能验证的需要，设计仿真脚本，掌握FPGA仿真波形、EDA软件资源、操作系统、专用硬件模块等数据收集、分析与处理的方法，通过数据分析验证所设计硬件、软件及系统。

目标3：熟悉硬件、软件设计主流工具（EDA软件、FPGA开发平台、操作系统和软件开发调试环境等）的功能、特点及使用方法，掌握基于主流设计工具进行软硬件协同设计开发的流程和方法。

目标4：能根据设计任务要求组建团队，成员分工明确、合理。能理解多学科背景下的团队中各角色的定位与责任，能够胜任个人承担的角色任务，并能与团队其他成员有效沟通。

目标5：软硬件开发过程中各个功能模块和系统集成的版本管理和控制，能够运用项目管理工具完成项目决策和管理。

目标6：能设计开发全新的软硬件功能模块并集成到系统中去，这需要通过不断学习掌握新技术和新方法才能做。

## 设计任务

1. 硬件平台搭建的实践准备
2. 基于MIPS CPU的硬件平台搭建、
3. 自定制接口模块的设计
4. MIPS CPU硬件平台的中断处理
5. HOS操作系统的构建与运行
6. 搭建操作系统开发环境，编译HOS操作系统内核，下载到MIPS CPU硬件平台运行。
7. 从内核到应用
8. 蓝牙模块及马达驱动模块硬件实现
9. 蓝牙小车软件应用实现
10. 蓝牙小车应用扩展和创新

## 设计要求

硬件开发工具和软件开发环境搭建，实现一个简单的基于MIPS CPU处理器的硬件平台，并运行简单测试程序。

通过IP集成的方式搭建一个基于MIPS CPU的嵌入式系统。

设计实现自定制的IP模块。

MIPS CPU处理器硬件平台的中断功能实现进行实践，在系统中添加中断控制器，并测试BootLoader程序，为下一步操作系统的实践做准备。

搭建操作系统开发环境，编译HOS操作系统内核，下载到MIPS CPU硬件平台运行。

掌握HOS中从应用层到系统内核层的完整调用路径，在HOS内核中添加系统调用，实现“Hello World”应用程序

在上面构建的软硬件平台上设计实现蓝牙模块和马达驱动模块

在HOS操作系统上开发应用软件，实现蓝牙小车的基本功能。

添加新的软硬件模块，实现创新性、开拓性的功能。

# 系统总体设计

## 设计目的

利用fpga搭建MIPS处理器，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。

MIPS处理器使用总线结构统一整个系统的输入输出以及数据存储。利用统一的中断寄存器管理终端响应中断的产生。

利用hos系统和mips处理器相互配合，形成一个软硬件配合的系统。

## 硬件部分设计

### 硬件平台的搭建

搭建基本的硬件平台，支撑Hos系统的运行。自定义额外的模块，首先在MIPS处理器部分，编写verilog代码，实现各个模块的硬件逻辑。自定义各个接口的含义和功能，处理好信号通路。

### 信号处理

定义各个模块发出的中断信号，构建信号通路。

### 包含的模块

蓝牙控制模块、超声波模块、马达驱动模块。

## 软件部分设计

### Hos系统的编译运行

将原始版本的hos系统进行编译和运行。

本地运行环境：

windows 10 version 1909

cgywin

mips编译器、gcc编译器

### 增加系统调用

定义系统调用编号，增加系统调用。

### 增加设备驱动中断处理

增加蓝牙和超声波系统调用中断。

自己定义系统中断的编号，然后编写识别中断的函数，函数在识别中断类型后可以做出相应的动作。

# 硬件平台设计与实现

## 设计目的

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。

将提供的.bit 文件通过烧写到 FPGA 开发板，并观察其运行。

根据不同功能需求，在硬件平台上扩展编写需要的模块，实现额外的硬件功能。

## 实验步骤

### 安装实验需要的vivado以及环境配置

安装vivado软件，安装实验需要的各种环境。

### 基于 MIPSfpga 的简单硬件平台搭建

构建一个由 IP方式实现 FPGA 上的跑马灯，烧写比特流测试。

### 自定制模块设计

编写额外的自定义模块，首先明确各个信号的输入输出，明确输出信号和输入信号的演算过程。

编写verilog代码实现硬件逻辑。对于本来fpga板子过高的时钟频率需要对其进行分频，对于时钟上升沿和下降沿的信号变化需要分清。

## 故障与调试

在查看vivado工程的ip核的时候，软件陷入崩溃，在查阅了相关资料后，问题依旧，在尝试了使用管理员模式打开软件以及修改系统路径等方法后都无法成功。为了解决这个问题，组内更换了多次系统和电脑，偶然间就解决了这个问题，至今为止也不知道是什么原因导致解决这一问题。

## 存在的不足和缺陷

Vivado软件优化较差，往往不能充分利用电脑的CPU性能，对于多核心CPU的支持较差，导致综合等工作的运行很慢。

# 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Hos 操作系统的环境，学习使用gcc等linux平台的软件，熟悉命令行的工作方式。

将生成的hos系统镜像下载到fpga板子上，运行软硬件协同的系统，实现软件功能。

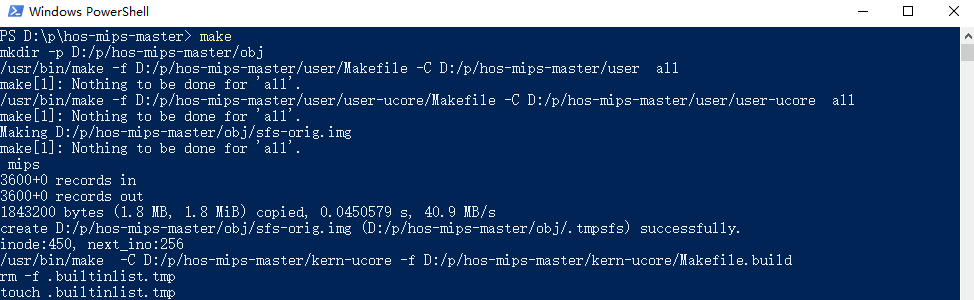
熟悉操作系统的原理，阅读和修改源码以实现额外的功能。

## 实验步骤

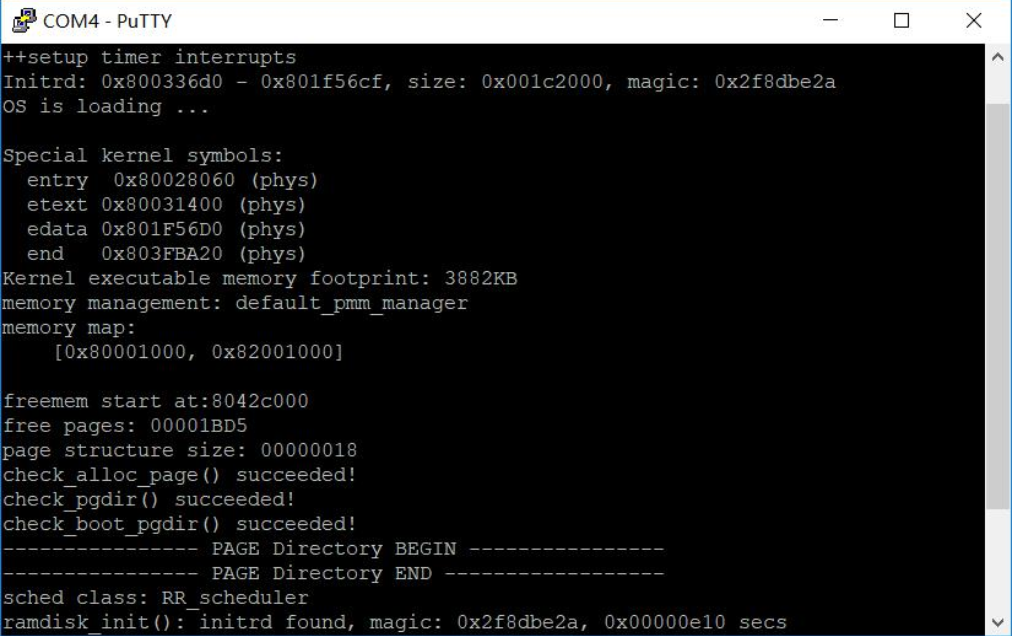
### Hos系统的构建和运行

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。

将cygwin系统的路径写入系统变量中，然后进入Hos系统makefile的目录，对于整个目录运行make。



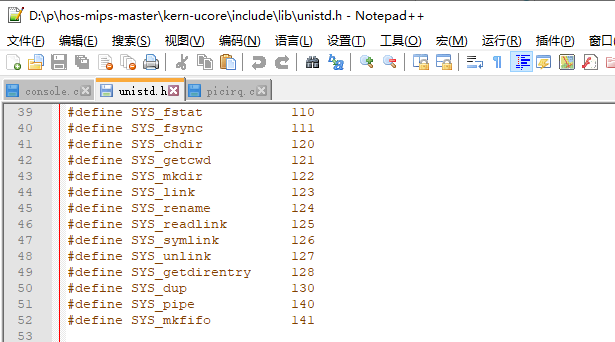
Make完成后，运行run.bat的批处理文件，查看是否可以成功把hos系统下载到fpga的板子上。



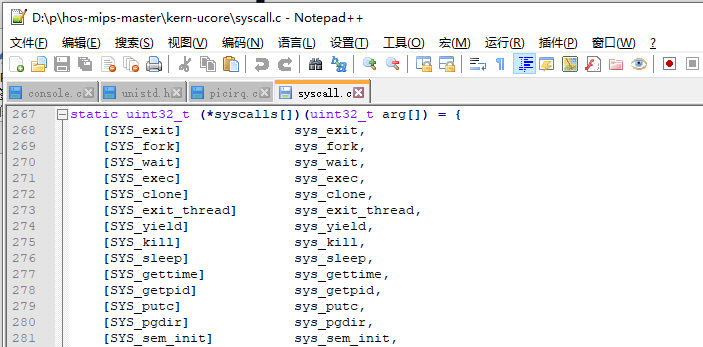
在查看到上面那幅图的画面后，就说明当前Hos系统已经成功运行在了mips处理器上了。

### 在hos系统上增添系统调用（内核）

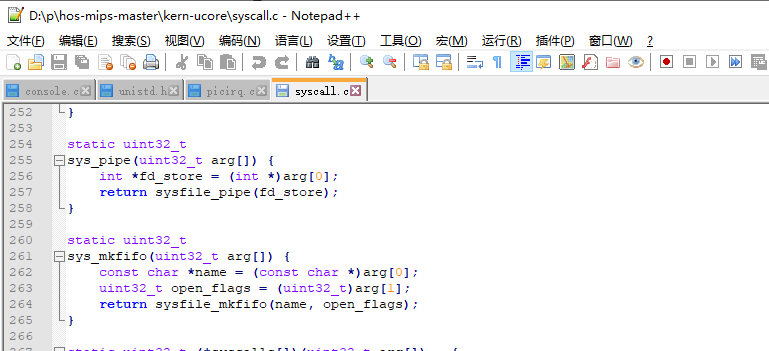
首先增添和定义新的系统调用，在kern-ucore\include\lib\unistd.h文件中增加新的系统调用编号。



同样，也要在同一个文件中增加函数的地址入口，可以看出，系统调用的原理是系统保存了一个函数地址入口表。

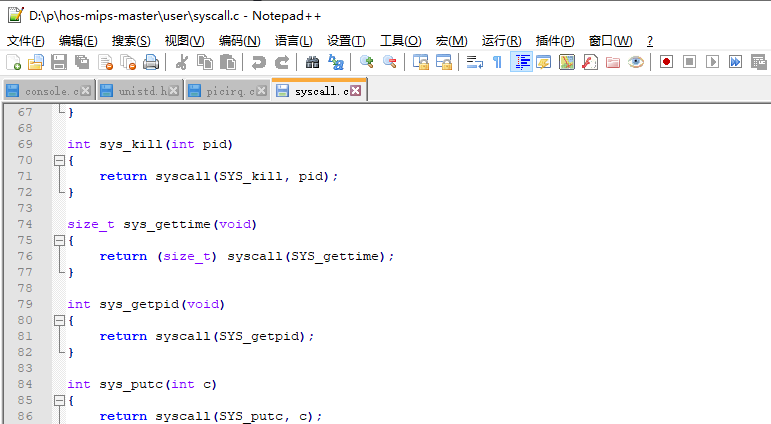


然后需要在库文件中增加新的函数定义，这个函数先在内核态的文件中增加定义，文件路径user\syscall.c。

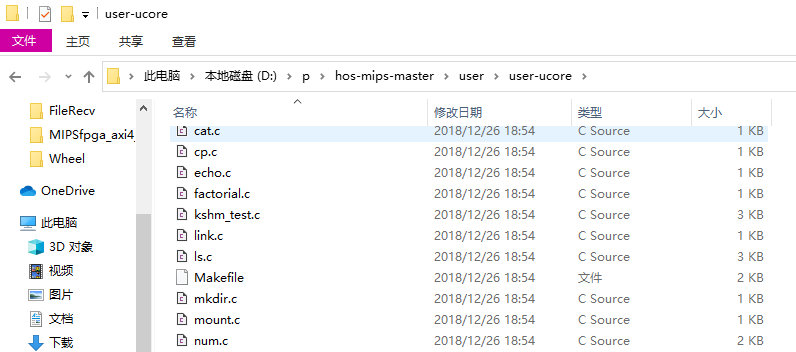


### 在用户态的文件中增加系统调用的函数

在user\syscall.c的文件中添加函数的具体定义。

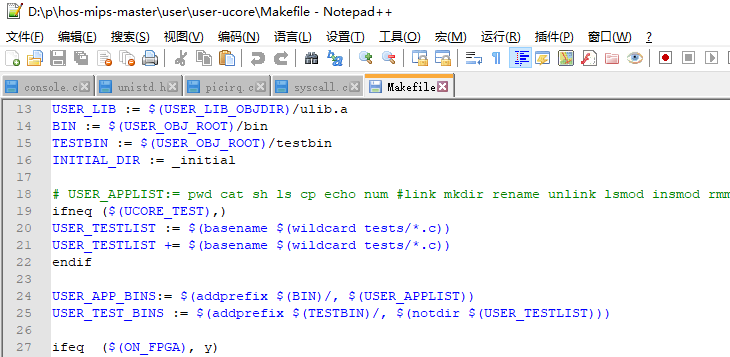


最后需要在user\user-ucore文件夹下，新建文件，文件名就取为新增加系统调用的名字，这里需要写的是一个带main函数的完整程序。



可以看到，这里文件夹包含了许多命令文件。

最后，在user\user-ucore\Makefile中增加新的命令调用。



# 应用设计与实现

## 设计目的

实现一个蓝牙控制的小车，该小车还支持自动避障的功能。蓝牙芯片采用hc-06芯片，马达芯片采用l293d，超声波芯片型号采用rcwl-1601。

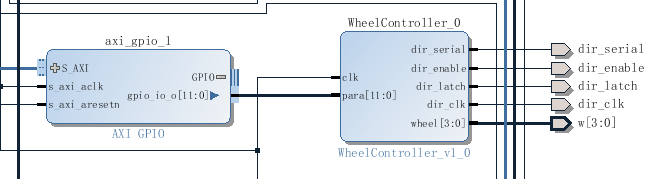
实现以 UART 串口协议为基础的蓝牙芯片总线外设 AXI 总线接口模块，然后实现以 PMOD 接口协议为基础的 L293D 驱动板总线外设 AXI 总线接口模块。超声波模块也采用AXI总线模块。

## 实验步骤

### 马达驱动模块实现（含变速实现）

FPGA部分，马达用自定义编写和封装的模块控制。利用gpio控制器存储方向和速度数据，传给马达控制模块。

总体模块连线图如下所示。



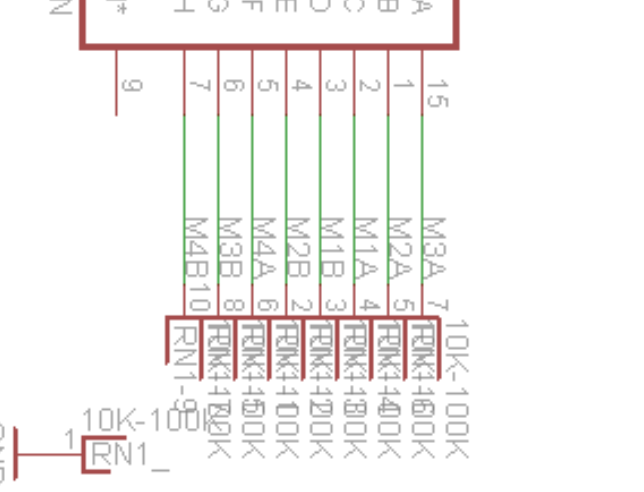
通过dir\_ser和dir\_clk始终控制输入数据到8位移位寄存器，时钟频率为1000Hz。数据传输完成之后给dir\_latch锁存信号置为1，则8位移位寄存器的内容被送进8位存储寄存器中锁存。

这样，就可以根据串行锁存器的原理编写相应的verilog代码，从而依次输出dir\_serial、dir\_enable、dir\_latch、dir\_clk的信号了。

通过PWM，控制CPU发给马达驱动芯片的数据的不同占空比来控制电压的高低，经过马达驱动芯片转化之后给马达驱动相应的电压。通过测试轮子的转速，我们可以得到一个几个固定的值，在verilog代码中，我们声明了一个计数器，计数器根据时钟频率变化，计数器有数据范围，超出范围则归零。

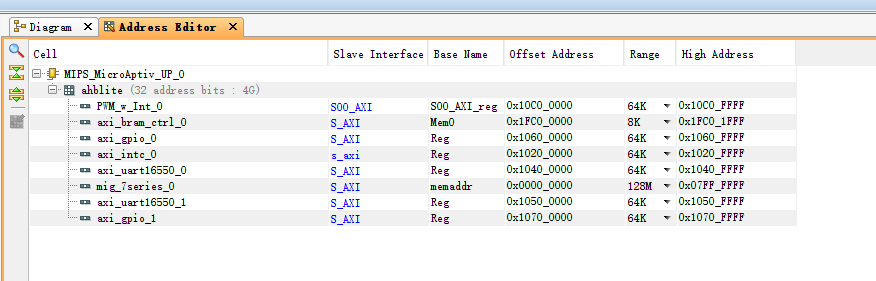
如果计数器小于这个固定的值，那么马达模块输出驱动信号1，否则输出0。这样就可以输出固定的电压，使得轮子驱动。如果我们调整这个固定的值，那么就可以实现变速功能。

控制轮子的方向一共8位数据，查阅手册，每位对应的轮子可以看到。



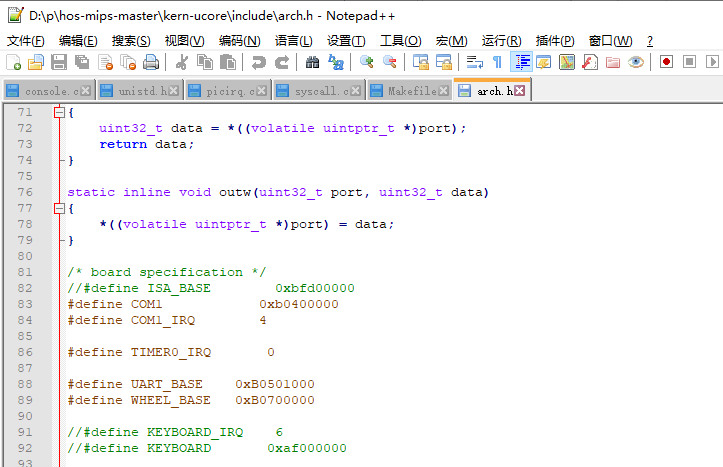
软件部分直接对于指定的缓冲区写入方向和速度数据，就可以直接控制马达的运行。我们在vivado软件中开辟了一部分特定的空间来存储方向和速度数据，只需要向这个缓冲区内写数据，就可以实现马达驱动。

缓冲区的地址在硬件和软件部分需要相互对应。



这里马达驱动的地址是0x1070\_0000，那么就需要在arch.h中显式地声明这个地址以对应。

这里，马达驱动缓冲区对应成0XB0700000。



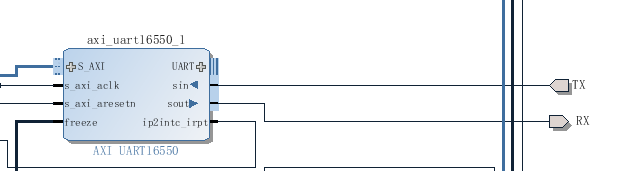
### 蓝牙驱动模块实现

这里也同样需要在硬件和软件的缓冲区之间做好地址的对应。对应方法和马达驱动类似，不再赘述。

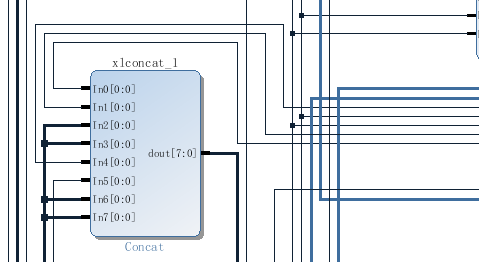
FPGA部分采用uart 16550芯片，利用芯片的rx和tx接口连接蓝牙芯片的rx和tx接口，实现数据的存储、读取以及同步。16550芯片在处理蓝牙数据的时候会发出中断信号，MIPS系统接受到信号可以给hos系统使用。

软件部分，hos系统接受到蓝牙发出的中断信号，然后对于中断寄存器检查，识别中断类型，读取蓝牙信号。

连线图如下。

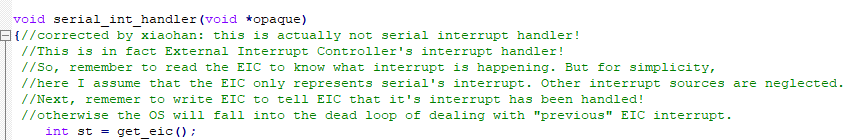


蓝牙芯片收到信号会产生中断，中断信号会送给中断寄存器，进而引发系统中断的产生。这里，我们可以自定义中断。



这里，我们可以看到，整个系统利用了四个中断位，0号中断是超声波测距中断（之后会介绍）；1号中断就是蓝牙中断（连接的是ip2intc\_irpt接口）；4号中断是原来系统保留的中断，并没有用处，我们这里将它保留，不做改变；最后，就是5号中断，这是响应键盘中断。

我们在console.c中发现了这样的一个函数：



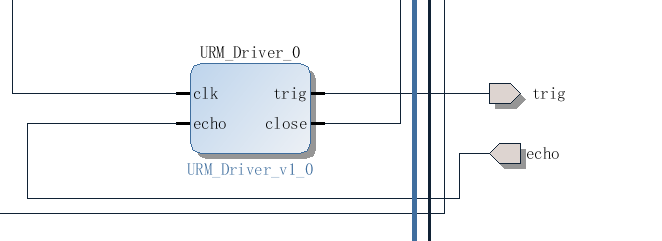
经过研究发现这就是响应中断，并且处理中断的函数。根据函数的注释，我们可以得到信息：硬件系统发出中断信号，就会触发这个函数，这个函数需要实现的功能就是识别中断并且响应中断。识别中断可以通过查看之前的中断寄存器（一共9位），然后得到中断类型。

### 超声波模块

超声波模块通过trig接口触发，连续产生8个波形，如果存在一个声波传回并且被接收，那么就会收到echo信号。

这里我们利用超声波来实现避障的功能。超声波模块每隔一定时间发出trig信号，通过接收信号的达到时间，测算前方障碍物的距离。如果距离过短，那么就会发出中断信号给中断寄存器。

hos系统会判断中断类型然后做出反应，小车会右转避障。



原理其实十分简单，就是不间断地重复测算障碍物距离，因为声速很快，所以相当于可以“实时”地得到距离前方障碍物的距离。如果检测到障碍物，就可以发出中断信号，中断的处理同上。

## 故障与调试

之前实现蓝牙小车的基本功能时，仅仅使用了系统调用方式来驱动小车的运行。在实现小车的超声波避障时，由于需要系统自己产生中断控制小车运行，所以不得不采用系统中断的方式控制。

之前采用过系统调用和蓝牙中断的混合方式，结果发现小车不能及时响应超声中断，原因是系统调用的实现方式是阻塞式的，中断无法打断。

## 存在的不足和缺陷

整体功能还可以进行进一步的丰富，蓝牙小车避障方式比较单一。

# 总结与心得

## 实验总结

实现了一个完整的小车，小车可以遥控同时具有一定的自主功能，还是挺有成就感的。

## 实验心得

理解了硬件和软件的协同工作模式，同时对于操作系统的系统调用和中断有了不少认识。

## 意见与建议

希望提供更多一点的实验解说材料，感觉材料有点少。

# 致谢

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: 2020-01-05** |