

电子技术课程设计

基于受控无人车的场地探测和 3D 重建系统 预习报告

张蔚桐 2015011493 自 55

陈 巍 2015011481 自 55

目录

1	选题背景及课题简介	2
2	方案比较与选择	3
2.1	主控元件的选择	3
2.2	通信机制的选择	3
2.3	供电方式的选择	4
2.4	摄像头数据传输技术的选择	6
2.5	车辆位置角度信息处理方式的选择	7
2.6	其他注意事项	8
3	基于 FPGA 的数字系统框图	8
4	传感器/执行机构接口电路图	9
4.1	传感器电路	9
4.1.1	数字接口电路	9
4.1.2	模拟接口电路	10
4.2	执行接口电路	11
5	基于 Webench 的电源电路仿真	12
6	PCB 的设计制作	15

1 选题背景及课题简介

根据《智能交通》和《智能家居》的课题方向，我们将选题定位为基于受控无人车的场地探测和 3D 重建系统。

目前，受控无人车，即对无人车发送指令控制无人车的技术已经有了很多现实生活中的应用。然而，这种受控无人车要求控制者能够看到无人车的运动状态，并对无人车进行控制。这样不方便无人车进一步进行无人作业。例如，在地震灾区倒塌的房间中，控制人员无法看到无人车的具体位置，这就需要无人车对周边环境数据进行采集，方便控制人员进行进一步的控制。另外，通过加装场地检测装置，无人车或机器人可以到很多危险场合进行作业和信息的采集。例如，2011 年日本福岛核泄漏之后，东电公司曾派遣如图 1 所示的机器人到反应堆内调查事故情况。



Fig. 1: “蝎形” (“scorpion”) 机器人行进在福岛核电站 2 号反应堆内

通过加装摄像头和辐射监测设备，机器人可以在人类无法承受的辐射量内进行高强度监测作业。

另外，目前 3D 检测和建模技术方兴未艾。利用无人机等设备对建筑物外表面进行 3D 检测建模的技术已经实现。计算机视觉领域内，SfM(Shape from Motion) 等基于单目摄像机的 3D 重建技术已经相对比较成熟，因此，周边技术的成熟为我们的设计提供了相当的基础。同时，我们也考虑采集小车的位置信息，一方面，对于控制人员来说，了解小车的行进路线（相对位置）也是一个很重要的信息点，另外一方面，根据计算机的相关知识，我们也可以知道获取准确的拍摄角度和拍摄位置可以更好的加大 3D 重建的精度，而这方面的应用和算法据我们所查阅的资料来看相对较少，可以进行进一步的创新。

因此我们将选题定位基于受控无人车的场地探测和 3D 重建系统，目标是通过控制无人车的行为，完成对场地内图像信息的采集，并在合适的情况下允许上位机进行 3D 重建，完成对场景 3D 状态的描述工作。根据时间限制和具体的完成情况，我们将整个项目分为如下几个部分进行。

1. 第一部分

- 基本的电源管理和小车运动控制

- 在上位机控制下小车完成相关运动
- 小车安装摄像头，将图像回传上位机

2. 第二部分

- 图像实时回传
- 小车完成相对位置信息的记录和回传
- 利用位置信息进行二维地图的展示工作

3. 第三部分

- 将上位机应用部署到移动平台
- 小车安装避障系统，并可以自动运行
- 三维重建，同时利用位置信息

从上面我们可以看出整个工程量较大，受到两周的开发时间的限制可能不一定全部完成，剩余工程可以作为参加挑战杯等进一步开发项目的发展方向。

2 方案比较与选择

2.1 主控元件的选择

首先我们考虑主控元件的选择，根据我们的技术水平和课题的情况，可供选择的主控芯片为 MCU（单片机）和 FPGA。相比 FPGA，MCU 具有开发相对简单的优势，以串口通信协议为例，大部分 MCU 的系统库中均已经封装了串口通信协议，而 FPGA 相对更接近底层，因此需要自行完成相关的通信协议的封装，这将消耗大量的时间和精力。

然而，相比于 MCU，FPGA 在速度和并行性方面有着很大的优势。根据我们查阅的资料，以 STM32F103 为例，其时钟速度可以达到 72MHz，然而由于串行执行的原因，IO 速度相对于 FPGA 较低；对比我们使用的 Xilinx FPGA，其时钟速度可以达到 100MHz，同时高度并行使得模块之间互相不干扰，保证了 IO 速度等要求。

因此，考虑到性能要求以及课程需要，我们在硬件方面采用了 FPGA 作为主控进行开发。

2.2 通信机制的选择

根据查阅的资料，可供选择的无线通信机制有如下三种：

1. WiFi 信道

这种通信机制需要将 WiFi 模块植入硬件，使得小车可以联网传输信息。上位机或移动平台使用 WiFi 下载信息。这种方式的优势一方面是信息传输距离较远，在一个路由器覆盖的范围内，信息均可以被传输。另一方面可拓展性也比较强，如果设计了合理的联网方式，将进一步解除距离限制，通过将信息上传到云空间，用户可以在任何联网的地区完成信息的收取和对小车的控制。

这种通信机制的缺点是过于复杂，尤其是使用 FPGA 进行开发的情况下，网络通信协议将带来更大的开发时间消耗，不适用于本课程的短期开发。

2. 蓝牙信道

这种通信机制利用蓝牙模块，蓝牙模块已经将蓝牙信道封装成串口的形式，从很大程度上方便了开发。同时，通过对电脑，移动设备上的蓝牙模块进行开发，用户可以不依赖于第三方设备（如路由器）和小车进行通信。虽然通信速度上相比 WiFi 也有一定的损失，并且通信距离受到明显限制，但相比于 WiFi 模块而言，其功耗更小。

3. SPI 信道

后期的开发表明，蓝牙的速度并不能保证摄像头传输的实时性，因此后期考虑采用 SPI 信道替代原有的蓝牙信道，然而相比于蓝牙——串口信道，SPI 的开发不论是在上位机方面还是在 FPGA 方面均比较复杂，因此不一定可以完成。

综合以上三种通信机制的优缺点，我们选择蓝牙信道作为通信手段。

2.3 供电方式的选择

首先通过查阅可能使用到的外设的技术手册，我们大概确定了供电需求为

- 5V@1.5A
- 3.3V@1.5A
- 7.2V@40A

其中 5V@1.5A 负责向 FPGA 5V@1A 以及舵机供电，3.3V 电源负责向外设供电，例如我们采用的摄像头 OV7670 采用的供电电压为 3.3V@100mA。7.2V@40A 电源负责向电机供电，并应接受 PWM 调制来控制电机转速。

通过查阅资料我们发现，供电的难点在 7.2V@40A 电源，而 5V@1.5A 电源和 3.3V@1.5A 电源均可以使用实验室提供的 TPS54160 SMT 芯片完成设计，这一部分内容将在后面基于 Webench 的仿真实现上面进行说明。7.2V@40A 电源的电流限额主要受到电机供电电流要求影响，从实验室提供的数据手册我们可以发现，电机的空载电流 2.4A，最高效率电流 11A，堵转电流 52.8A，根据网上查阅的相关资料，电机的限流应当是最高效率电流的约 4 倍，因此设计 7.2V@40A 电源负责向电机供电。

这里讨论 7.2V@40A 电源的设计问题。因为需要接受 PWM 信号调制，因此我们采用了 H 桥进行设计。H 桥的电路原理图如图 2 所示。

Q1Q4 可以由处于饱和状态的三极管或开关状态的 MOS 管组成 J1 和 J2 分别接入前一级驱动电路产生的两个反相 PWM，设某时刻 J1 为高电平，则 Q3 导通，Q1 截止；由于反相，J2 为低电平，Q2 导通，Q4 截止，电流由通过 Q2，经负载，再经过 Q3 流入地。同理，当某时刻 J1 为低电平时，Q1,Q4 导通，Q2,Q3 截止，电流依次经 Q1,R1,Q4 流入地。综上，任意时刻有且只有一个对角线上的两个 MOS 管是导通的。

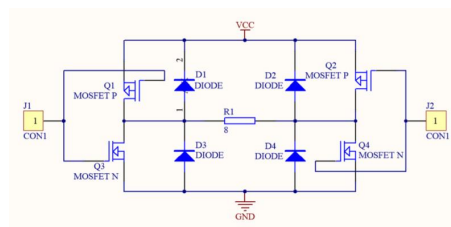


Fig. 2: H 桥的电路原理图

对于电机，我们可以设定刹车模式，即可以通过与上述方法类似的方法将电机两脚短路，使得电机迅速产生电磁制动实现刹车效果

通过搜索符合要求的元件，我们发现合理的解决方案为 bts7960/7961 芯片和自行搭建 H 桥两种方案。

bts7960/7961 芯片是接受 5VPWM 信号调制的，输入电压 5.5V 至 27V@43A 的半 H 桥电路，采用 SMT 封装，输出电流符合要求。然而，由于我们电机的额定输入电压为 7.2V，经过对电机的初步调试我们可以发现电机在 7.2V 输入电压下的转速非常高，导致速度不能得到控制，经过进一步的实验我们发现，在低于数据手册上提供的 5.4V 供电的情况下，电机的转速可以保证一个理想的状态，因此可以初步得到结论即 PWM 信号的占空比不可过高。

而自行搭建 H 桥，因为工艺比较复杂，同时在网上找不到合适的大功率 MOS 管供应商，因此我们放弃了这个方案。

2.4 摄像头数据传输技术的选择

我们采用的 OV7670 摄像头, 可以输出 25fps 640*480 VGA 格式的 RGB 三色图像, 输出采用 8 线同步并行传送因此输出时钟速度可以达到 $25 * 640 * 480 \approx 8\text{MHz}$ 。收到蓝牙信道是串行接口的原因, 蓝牙信道的速度必须达到 8MB/s, 这是一般蓝牙系统很难做到的。因此不论是 FPGA 读取信息还是蓝牙传输信息的速度, 均不如摄像头采集数据的速度, 需要找到合理的解决方式。

经过我们查阅相关资料发现, OV7670 的传输方式有如下三种形式:

- MCU/FPGA 直接采集:

如图 3所示, 这种方法是最简单, 最直接, 但也是最不好实现的方法, 以 MCU 为例, 多数的 CMOS 芯片 (如 OV7670) 的时钟速度可高达 24M, 一般 MCU 的 IO 端口速度根本不可能达到, 所以需要高速 MCU。这对多数用户来讲有些不现实。但也不是完全没有办法在低速上实现采集, 方法也很简单, 那就是降低 CMOS 的输出速度, 不过这需要靠外部的晶振和内部的 PLL 电路以及像素时钟速度, 帧速等多个寄存器共同设置, 并且要和 MCU 的 IO 速度匹配才可实现。但这么做将带来巨大的寄存器设置工作量并导致硬件图像的采集速度可能下降到 0.5 帧以下, 同时带来图像失真的可能。

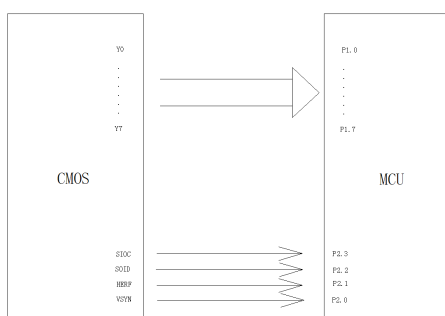


Fig. 3: MCU/FPGA 直接采集

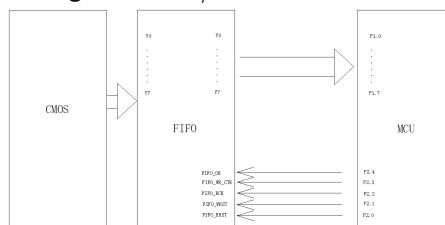


Fig. 4: FIFO 采集

- DMA 方式

这种方式主要用于 MCU 中, FPGA 中这种方式效果不明显, DMA 控制器使得数据可以绕过 CPU 直接进入内存, 但是在 FPGA 中, 同时受到串口通信速度的影响, 这种方式效果也不是很明显。

- FIFO 模块采集

如图 4, 用户只需要按上述时序图控制相关的几个控制引脚即可, 可

以很方便的使用在低速 MCU 上，另外一个好处是，可以直接 IO 口读取数据，读出的数据可以直接送屏，也可以经过 MCU 简单处理；当然也可以不经过 MCU，直接送到屏等外围器件使用。对于 FPGA，以及速度相对较低的串口单元，使用 FIFO 也能很大程度的解决速度不同步的问题。

2.5 车辆位置角度信息处理方式的选择

为了进一步增强 3D 建模的精度，我们希望能够采集车辆的位置信息和角度信息。由于摄像头和车辆是相对固定的，这样也就获得了摄像头的位置信息和角度信息。

根据我们查阅到的资料，车辆的位置信息和角度信息有几种不同的获取方式，如下所示。

- 惯性测量单元（IMU）模块

通过加速度计和陀螺仪的信息，可以直接读取车辆的加速度和角度。经过积分之后可以得到车辆的速度信息和角度信息，这种方法最简单，但是误差最大，加速度计易受到颠簸，碰撞，刹车等造成的脉冲影响，并进一步影响积分的准确性。角速度计（陀螺仪）具有一般性的零点迁移问题，这包括动态的零点波动和静态的温漂。经过积分会导致角度信息相当不准确。这种方法只能作为一种简单的参考，实用时必须加以处理。

- 磁场计

通过对地磁角度的测定，可以直接获取车辆的角度，并通过加速度计等方式获得车辆的速度，这种方式误差相对前一种较小，但是受外界磁场的影响比较严重，尤其是当存在 10A 绕组的电机驱动时，这种方法可以认为无效。

- 码盘

这种方法通过测量车辆四轮的转速经过解算得到车辆的具体位置，相比于之前几种方式，这种方式依赖于更加准确的光电传感器，因此得到的值也相对比较准确，这种方法的难点在于必须对车辆的刚性模型进行建模，同时安装码盘也是一个硬件上的难点。

综合提到的几种方法，我们计划采取 IMU+ 信号处理 + 码盘解算几种方式得到车辆的具体位置，希望通过反馈等方式希望得到更准确的车辆控制。

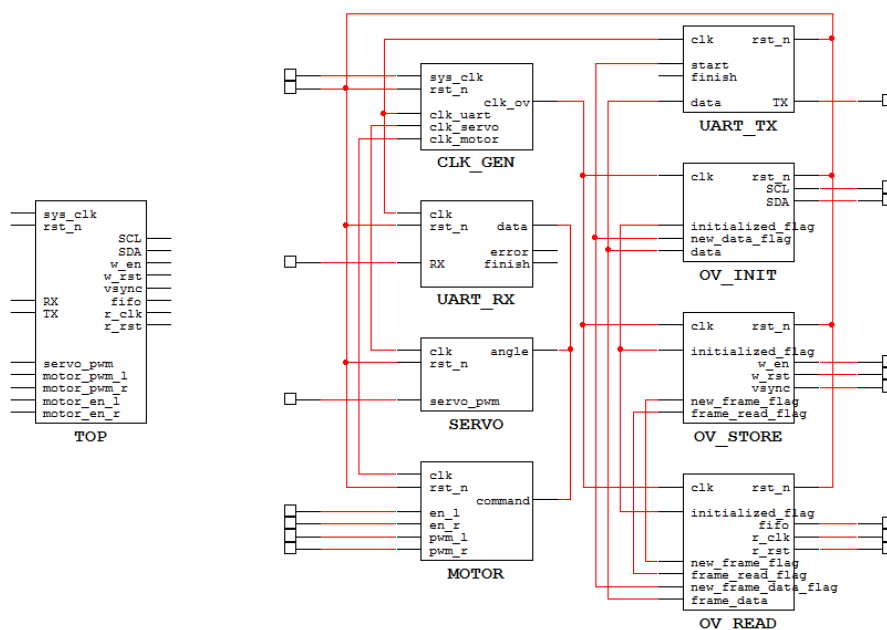


Fig. 5: 基于 FPGA 的数字系统框图

2.6 其他注意事项

在设计对车辆相对位置进行测量的过程中，考虑到 FPGA 在解算浮点数方面比较难以开发，我们选用了 NI myRIO 进行解算和控制，并采用 myRIO 自带的 WiFi 进行相关数据的回传。

NI myRIO 是 NI 针对教学和学生创新应用而最新推出的嵌入式系统开发平台。NI myRIO 内嵌 Xilinx Zynq 芯片，使学生可以利用双核 ARM Cortex-A9 的实时性能以及 Xilinx FPGA 可定制化 I/O，学习从简单嵌入式系统开发到具有一定复杂度的系统设计，采用 LabView 语言进行可视化编程。通过 myRIO 可以大幅简化控制算法如 PID 的调试过程，显著降低开发难度，加速开发进度。

3 基于 FPGA 的数字系统框图

如图 5 所示，左侧为封装的顶层模块，从右侧可以看出共分为 8 个子模块。

• CLK_GEN

该模块对 100MHz 系统时钟分频，输出 4 个时钟分别供串口、舵机、电机、摄像头使用。

- UART_RX

蓝牙串口接收模块，RX 为数据线，每从上位机接收一个码元即触发 finish 信号，并将数据存在 data 寄存器供其他模块使用。同时内部还提供了一个 error 接口，接收数据发生错误时触发。

- UART_TX

蓝牙串口发送模块，TX 为数据线，当触发 start 信号后，将 data 寄存器中的内容发送给上位机，发送完成后会得到一个 finish 信号反馈。

- SERVO

舵机控制模块，从串口接收转向控制指令，输出舵机 PWM 信号。

- MOTOR

电机控制模块，同样从串口接收指令，输出电机使能和 PWM 信号。

- OV_INIT

摄像头初始化模块，SCL 和 SDA 为 SCCB 协议的时钟线和数据线，初始化前 initialized_flag 为低电平，OV_STORE 和 OV_READ 处于失能状态，初始化完成后拉高 initialized_flag，摄像头便可正常工作。

- OV_STORE

将摄像头采集到的图像存储于 FIFO 中，存储完成一帧后触发 new_frame_flag 信号，等待 FPGA 读取图像结束后才继续采集下一帧。

- OV_READ

从 FIFO 中读取一帧图像，并存储在 frame_data 寄存器中，读取完成后触发两个信号，一个是 frame_read_flag，通知摄像头可以继续采集下一帧图像，另一个是 new_frame_data_flag，通知顶层模块将读取到的数据发送给上位机。

4 传感器/执行机构接口电路图

4.1 传感器电路

4.1.1 数字接口电路

这部分主要是完成数字模块之间的接口电路，由于各个数字模块已经将输出电平统一标准化为 TTL 电平，因此这部分的关键主要是接口通信协议的实现，主要有 FPGA 方面实现了 SCCB 接口通信协议，串口通信协议，摄像头数据读取协议等。另外，SIO_C 和 SIO_D 分别为 SCCB 总线的时钟线 and 数据线，为了方面对多个设备进行控制，OV 的 SCCB 两个端口采用的均是 OC 门输出，需要上拉 4.7k 电阻实现输出高低电平，这些需要在设计 PCB 的时候尽可能的考虑到并保证接线的稳定性和摄像头位置的稳定性，因此需要在 PCB 上统一制作并提供对 FPGA 的简单的接线电路，这部分的电路可以如图 6 所示。

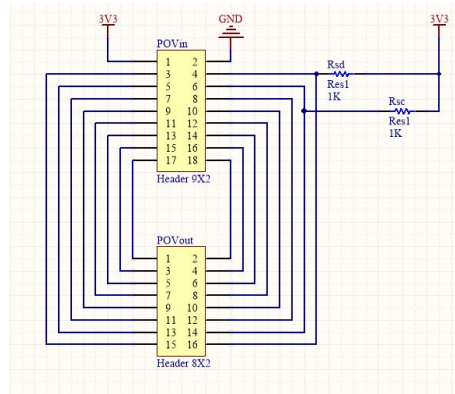


Fig. 6: OV 上拉电阻和转接电路

另一部分的接口电路问题是 3.3V 电平标准和 5V 电平不匹配的问题以及同时设计到的 FPGA 驱动能力不足的问题通过查阅给定的技术手册我们发现 FPGA 输入电流 1A，标准的输出电压为 3.3V，而不论是电机驱动的 PWM 还是舵机驱动的 PWM 均为 5V，这中间需要设计适当的接口电路，同时，由于工程量比较大，担心 FPGA 会出现总线电流过大或无法驱动相关接口电路的情况，因此我们采用了 74HC244 芯片作为接口的辅助电路，74HC244 是一款高速 CMOS 驱动芯片，支持三态门使能，通过 HC244 我们可以顺利的将 3.3V 输出转化为 5V，并提升驱动能力。根据 74HC244 的接线要求，我们在输入端接入下拉电阻 1k 实现相关功能。这部分的电路有如图 7 所示。同时我们将剩下的 74HC244 引脚同样通过排针引出，留待之后使用。

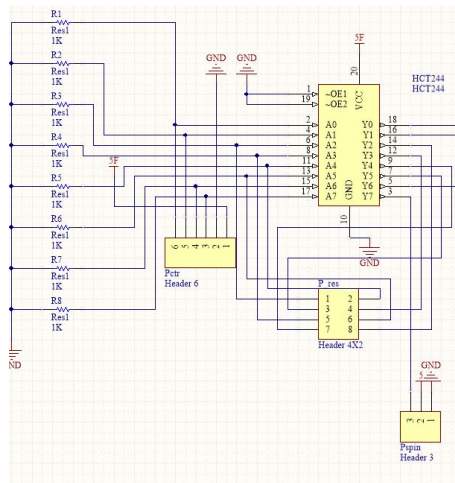


Fig. 7: 74HC244 驱动器电路

4.1.2 模拟接口电路

我们计划采用输出模拟电压的 ENC-03 陀螺仪，这是一种应用科氏力原理的角速度传感器，陀螺输出一个和角速度成正比的模拟电压信号。通过积分电路可以得到相关的角度等。根据 ENC-03 技术手册，我们设计输出放大电路如图 8 所示，整个电路由两个滤波器组成，左侧为低通滤波器，用于滤除传感器高频噪声，右侧为高通滤波-放大电路，用于放大电压信号和滤除低频温漂。

进一步，我们搭建基本积分电路可以得到角度信息，两个信息通过 AD 模块可以返回给主控设备。由于我们计划采用 myRIO 进行位置信息的测量，因此这部分可以直接使用 LabVIEW 语言在 myRIO 上完成，就没有设计电路的必要了。

4.2 执行接口电路

这部分主要描述的是对电机和舵机的驱动电路

对于舵机驱动，单片机输出高电平为 3.3V 的 PWM 脉冲，经过前文提到过的 74HC 模块转 5V 驱动舵机旋转。

对于电机驱动，我们使用 BTN7970

模块，如图 9 所示。这里为了方便后文叙述，两片 IN 引脚被命名为 IN1, IN2，两片 INT 引脚被命名为 INT1, INT2。我们将 INT1, INT2 短接并输入 PWM 波形控制电机转动速度。将 IN1, IN2 分别接高低电平控制电机正转反转。例如 {INT1 = 0, INT2 = 1} 代表电机正转，{INT1 = 1, INT2 = 0} 代表电机反转。当 {INT1 = 0, INT2 = 0} 时，H 桥全部关断，表示不控制电机转动。当 {INT1 = 1, INT2 = 1} 时，H 桥下部导通，电机两端短接，使得电机的机械能通过电磁感应和短路线迅速释放，实现刹车功能。BTN 相关引脚仍为 5V 接线，因此使用 74HC244 完成相关的电平转换和放大工作。

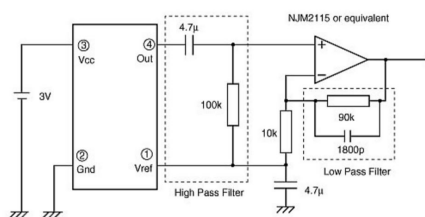


Fig. 8: ENC-03 模块外部电路图

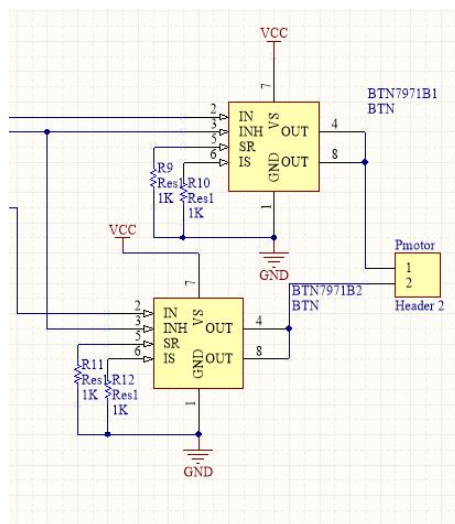
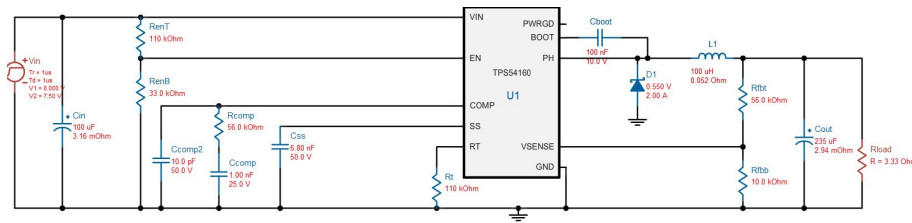


Fig. 9: 电机驱动电路



5 基于 Webench 的电源电路仿真

WEBENCH 是 TI 公司提供的设计环境，通过软件工具提供定制电源等的设计，通过使用 WEBENCH 可以提升设计效率。整体的设计思路如下：

- 输入电压的确定

我们将电池断开，测量电压约为 8.2V，将电池直接接电机，并在电机上施加比较大的阻力，发现电源电压约为 7V，因此我们设计电源时考虑输入电压为 6V9V 即可满足需求。

- 输出电压的确定

由表 1 所示，我们需要向 FPGA 提供 5V 外接电源，对舵机提供 5V 外接电源，对外设提供 3.3V 外接电源，因此设计输出电压为 5V，3.3V。

- 输出电流的确定

首先根据实验室提供的 TI 样片，我们选择了 TI TPS54160 降压模块，TPS54160 器件是一款带有集成型高侧 MOSFET 的 60V,1.5A 的降压稳压器。电流模式控制提供了简单外部补偿和灵活组件选择，功耗较小。

根据 TPS54160 的输出电流，我们限制电源输出电流为 1.5A。同时查阅技术手册可以得到主要的部件的供电信息如表 1 所示，总计需要输出电流 5V@2A,3.3V@500mA，因此需要设计两路 5V 供电电源，其中一路 FPGA 自用（包括驱动器 74HC244），一路给舵机/外设使用。

因此，进入 WEBENCH 中提供需求的输入，输出信息，选择器件为 TPS54160，可以得到元件的原理图。通过调整原理图上的元件的值的的信息，可以选用已有阻值的元件进行仿真，最终确定的电路原理图和响应时间仿真和稳态仿真如下面几张图所示。两个电路结构相同，相关参数差别如表 3 所示。整理仿真结果如表 2 所示。可以看出电源的性能效果还是比较理想的，可以进行实际电路的搭建。

Tab. 1: 主要部件供电信息

部件	电压	电流
FPGA	5V	1A
舵机	5V	800mA
74HC244	5V	100mA
蓝牙模块	5V	100mA
OV7670	3.3V	200mA

Tab. 2: 仿真主要性能指标

参数	3.3V 电源	5V 电源
稳态电压平均值	3.435V	5.326V
稳态电压峰峰值	3.648mV	2.465mV
启动时间	1.368ms	2.379ms
电感电流峰值	1.245A	1.753A

Tab. 3: 元件参数信息

元件名	类别/单位	3.3V 电源	5V 电源
Cin	电解电容/ μF	470	470
Rent	定值电阻/ $\text{k}\Omega$	110	110
RenB	定值电阻/ $\text{k}\Omega$	33	33
Ccomp	独石电容/ nF	1	1
Ccomp2	独石电容/ pF	10	10
Rcomp	定值电阻/ $\text{k}\Omega$	33	56
Css	独石电容/ nF	6.8	6.8
Rt	定值电阻/ $\text{k}\Omega$	110	110
Cboot	独石电容/ nF	100	100
D1	肖特基二极管/型号	SS24FL	SS24FL
L1	定值电感/ μH	100	100
Rfbt	定值电阻/ $\text{k}\Omega$	33	56
Rfbb	定值电阻/ $\text{k}\Omega$	10	10
Cout	电解电容/ μF	470	470

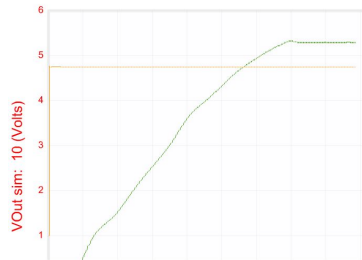


Fig. 10: 5V 电源启动情况

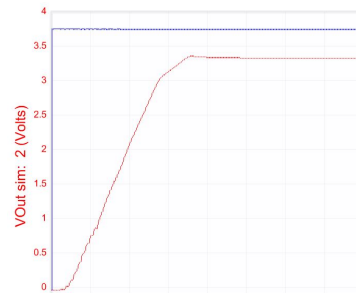


Fig. 12: 3.3V 电源启动情况

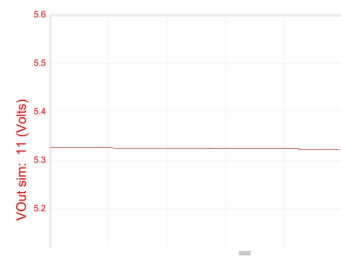


Fig. 11: 5V 电源稳态情况

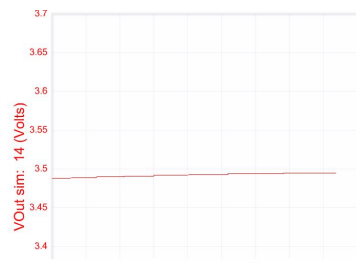


Fig. 13: 3.3V 电源稳态情况

Tab. 4: PCB 线宽和电流的关系

电流 (A)	线宽 (mm)	电流 (A)	线宽 (mm)
6.0	2.5	5.1	2.0
4.2	1.5	3.6	1.2
3.3	1.0	2.8	0.8
2.3	0.6	2	0.5
1.7	0.4	1.3	0.3
0.9	0.2	0.7	0.15

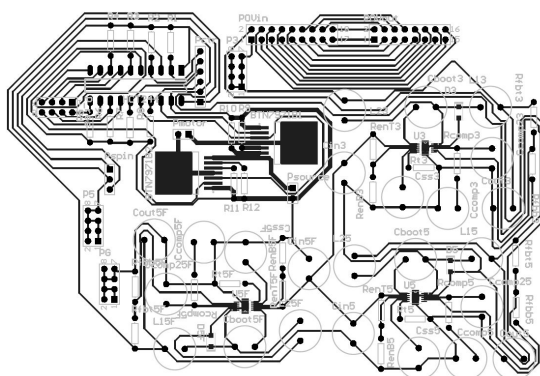


Fig. 14: PCB 版效果图

6 PCB 的设计制作

完成了电源模块的仿真之后就可以开始着手进行 PCB 的设计制作了，在设计制作的过程中，由于考虑到手工焊接的困难，因此尽可能多的选用了插接件代替 SMT 封装元件，如电阻，电容等，同时，根据走线可能涉及的电流大小如图 4 所示，我们设计走线宽度为

- 对 BTN 驱动板，走线 40mil(1mm)，对应电流 3A 左右
- 对电源管理模块，走线 15mil(0.4mm)，引脚处走线 10mil(0.25mm)，对应电流 1.5A 左右

同时对大电流部分加焊锡保护走线，防止过热。整版 PCB 单面布线，背面加装散热片，保证散热顺畅并注意 IC 散热引脚接触良好。同时考虑接线端子，IC 等的位置关系，充分考虑日后安装，打孔位置，保证接线顺利同时散热合理。PCB 如图 14 所示。