电子技术课程设计

基于受控无人车的场地探测和 3D 重建系统终结报告

张蔚桐 *2015011493* 自 *55*
陈　崴 *2015011481* 自 *55*

目录

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。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。

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

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。

1.1 具体完成情况

这一节总结一下提到的项目内容的完成情况

经过两周的具体调试，我们实现了第一部分的所有具体要求，包括小车的驱动模块，摄像头的安装和测试，蓝牙模块通信的测试等，并完成了整个项目的整合。但是由于蓝牙速度较慢，经过了长时间的调试，我们仍然不能实现图像的实时回传，同时，由于图像数据量较大，回传图像对蓝牙压力也比较大导致蓝牙模块并不是很稳定。虽然如此，整个项目的工作量，完成度和可展示性还是非常理想的，尤其是对基于 FPGA 对裸 CMOS 摄像头 OV7670 的封装等为代表的工作花费了我们大量的心血，总结起来还是比较满意的。

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 STMF103 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。

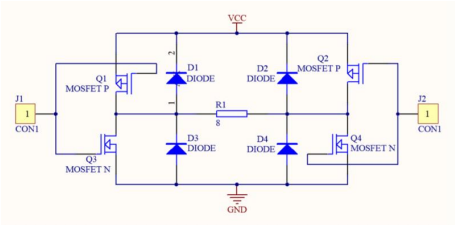


Fig. 2: H 桥的电路原理图

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，那么就是降低 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

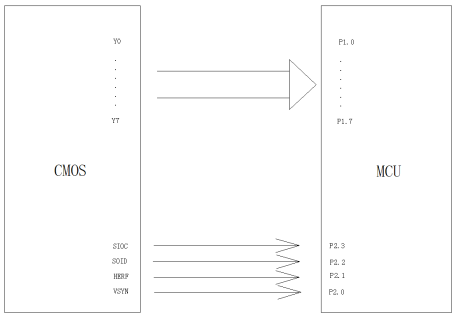


Fig. 3: MCU/FPGA 直接采集

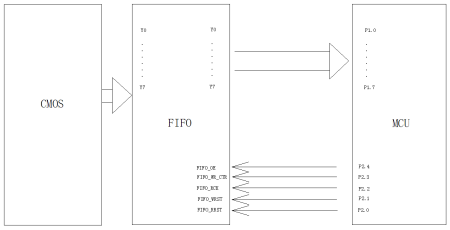


Fig. 4: FIFO 采集

0.5 帧以下，同时带来图像失真的可能。

- 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。

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。

2.7 实际项目的修正和改动

实际项目进行过程中，我们尽可能的对蓝牙的速度进行了利用，然而由于蓝牙速度受限，导致实际上图像回传不能实时化，并且对蓝牙通信速度上限的冲击导致蓝牙出现不稳定情况。在这种情况下，我们希望采用 SPI-WiFi-TCP 通信协议，然而 TCP 协议封装复杂，经过我们查阅相关资料，TCP 的引脚数量甚至超过了我们使用的 FPGA 的全部可控引脚数量，‘同时 TCP 通信机制也比较复杂，会对是 FPGA 层面的封装还是上位机的编写带来巨大的困难，因此我们最终放弃了这种思路。

同时，对于陀螺仪和测速方面，由于我们使用的 FPGA 引脚已经全部被占用，导致基于 FPGA 的相关开发变得不现实，因此我们选用了 NI myRIO 进行相关的开发，并完成了陀螺仪模块。对于测速模块，我们发现不论是车轮还是电机均很难安装码盘，因此我们采用了对车轮进行标注并利用红外进行测速，并基于 myRIO 调通了相关的功能，然而由于我们设计的车辆行驶速度偏慢，因此相关的测速变得没有实际意义，最终我们放弃了这个模块的整合工作。

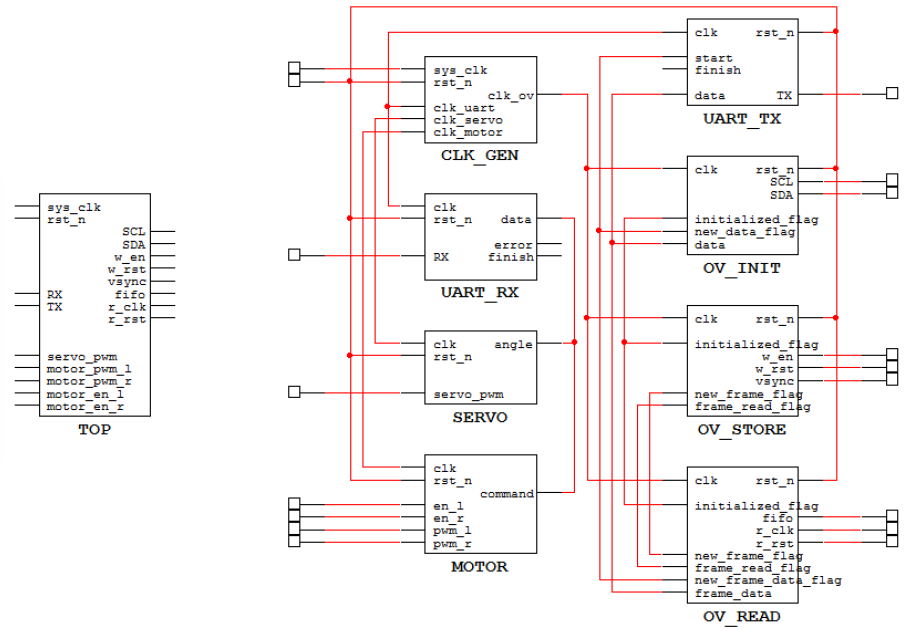


Fig. 5: 基于 FPGA 的数字系统框图

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 总线的时钟线和数据线,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

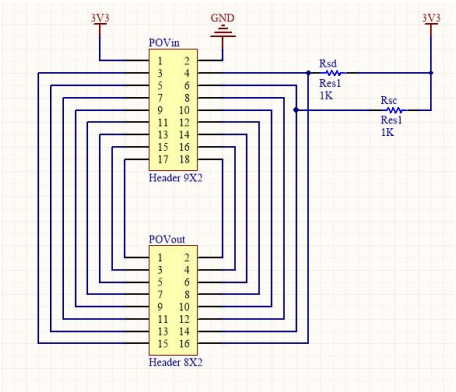


Fig. 6: OV 上拉电阻和转接电路

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。

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。

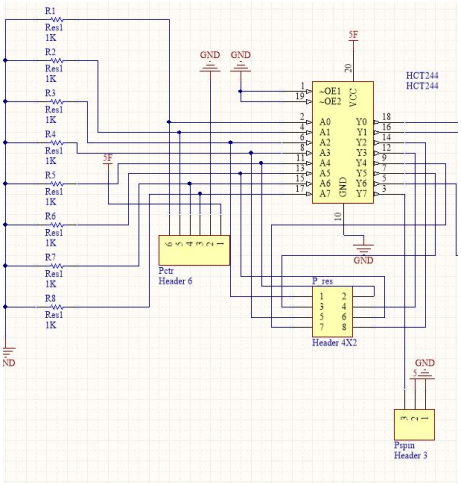


Fig. 7: 74HC244 驱动器电路

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。

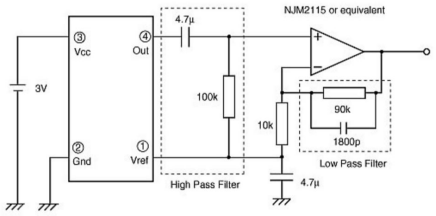


Fig. 8: ENC-03 模块外部电路图

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。

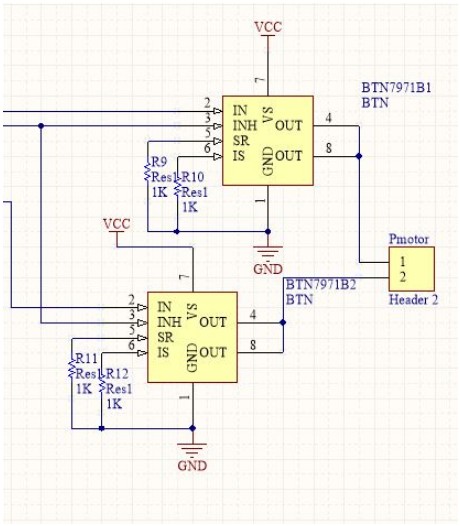


Fig. 9: 电机驱动电路

5 基于 Webench 的电源电路仿真

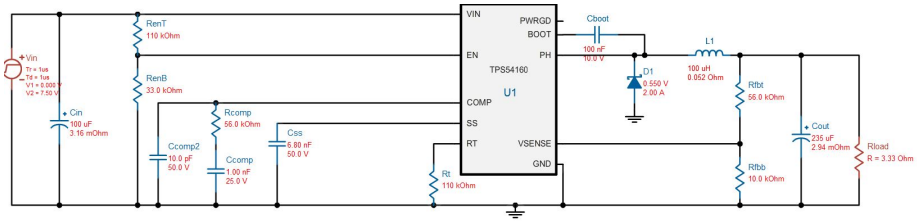
WEBENCH 是 TI 公司提供的设计环境，通过软件工具提供定制电源等的设计，通过使用 WEBENCH 可以提升设计效率。整体的设计思路如下：

- 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/kΩ	110	110
RenB	定值电阻/kΩ	33	33
Ccomp	独石电容/nF	1	1
Ccomp2	独石电容/pF	10	10
Rcomp	定值电阻/kΩ	33	56
Css	独石电容/nF	6.8	6.8
Rt	定值电阻/kΩ	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/kΩ	33	56
Rfbb	定值电阻/kΩ	10	10
Cout	电解电容/μF	470	470

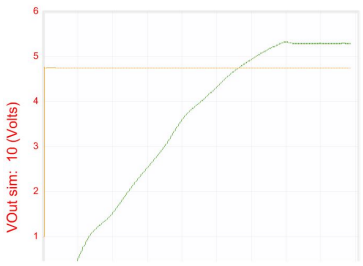


Fig. 10: 5V 电源启动情况

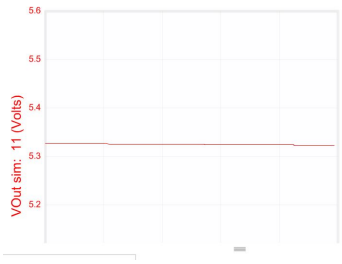


Fig. 11: 5V 电源稳态情况

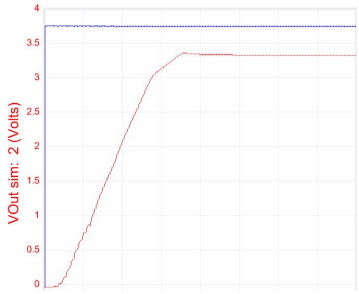


Fig. 12: 3.3V 电源启动情况

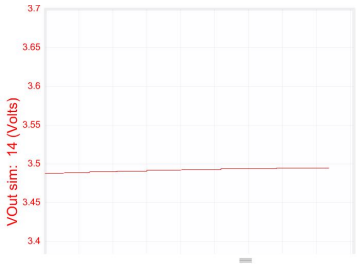


Fig. 13: 3.3V 电源稳态情况

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

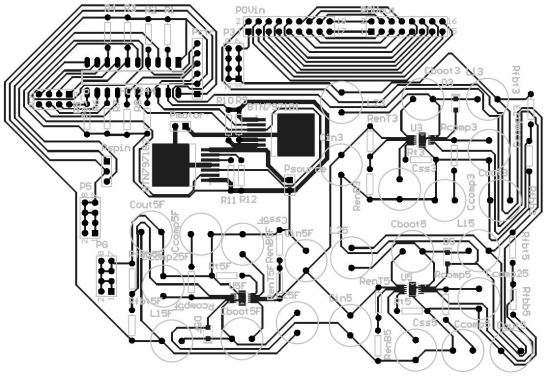


Fig. 14: PCB 版效果图

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。

7 课题开发与调试中出现的问题分析

7.1 数字电路部分

数电部分完全使用 FPGA 进行开发，遇到的困难几乎都来自 ov7670 图像传感器模块。

首先，图像传感器在工作前需要先进行寄存器初始化配置，读写寄存器使用的是 SCCB 协议。一开始调试时，待写入的寄存器的地址和内容都存放在文本文件中，且因为数据是串行写入，各个寄存器的初始化是顺序执行的，所以初始化语句都写在 initial 块中，又由于每次读或写一个寄存器都需要 SCCB 协议中的起始、应答、结束等信号，这就会产生大量对时钟线重复的操作，为了使代码看起来更简洁，这些相同的操作都封装成了 task。这样一来，尽管在仿真测试时代码都正常运行，但电路却是不可综合的。

最后，解决这些问题的方式是重写代码，将各寄存器地址和内容都以常量形式存在 FPGA 中，以 FSM 状态的跳转模拟各语句的顺序执行，大量相同的操作也只能重复相同的代码。其中还发现寄存器配置需要一定的反应时间，因此每写入一字节后都加入几毫秒的延时。

图像传感器的初始化只进行一次，而图像的采集和读取是无限循环的，且必须在初始化完成后才能进行，所以使用了一个全局的信号来判断初始化是否完成。前文已经分析过使用 FIFO 模块采集图像的好处，但是在我们查阅到的资料中，绝大部分不带 FIFO 的传感器封装成 18 个引脚，带 FIFO 的传感器封装成 22 个引脚，而为了节省 FPGA 的 IO，我们选购的传感器是带 FIFO 模块且封装为 18 个引脚的，这种封装的资料几乎找不到，开发完全没有前车之鉴，只能依赖于 ov 公司和 FIFO 芯片的数据手册。这些数据手册中提供的都是各 IC 引脚的时序，而封装后外部引脚的时序只能靠分析和推测。在硬件调试时，我们合理地使用 FPGA 上的 LED 进行辅助调试，通过 LED 的状态来分析电路状态，从而找出代码中的错误，大大节省了开发时间。

通过蓝牙串口模块回传数据时，理论上传输速率可以达到 1.3 秒每帧，但实际测试中发现，如果以这么高的速率发送数据，蓝牙模块的数据吞吐量已经达到极限，会发生误码、丢包、甚至蓝牙连接中断的现象。受硬件性能所限，只能在传输一字节后加入至少 100us 的延时间隔，使蓝牙模块能够较稳定地工作，最终展示时，需要采用串口替代蓝牙来加速传输速度。

7.2 模拟电路部分

模拟电路部分主要包括电源驱动模块和电机驱动模块问题，虽然通过查阅技术手册和基于 WEBENCH 的仿真可以看到基本的电路，但是从电路原理

图到电路实际工作还有很远的距离。

首先是电路原理图的设计，通过查阅资料，我们发现实验室提供的 TPS54160 模块输出电流为 1.5A，设计的 5V 电源不能很好的驱动 FPGA（1A）和舵机（800mA 堵转）。因此我们设计 FPGA 和 PCB 上的其他元件一路供电，舵机和其他外设一路供电。这种设计虽然降低了电路的负载，但是显然不可避免地增加了电路的规模和后续布线，制版，焊接的压力。

我们将整个电路进行整合为一块 PCB，这样从最大程度上减少了可能存在的飞线和不稳定的情况，然而过大体量的电路带来了布线，制版，焊接的困难。使用 AD 模块制作 PCB 的过程中，首先遇到的问题是 AD 不能提供我们需要的元件封装库，因此我们参照元件的技术手册自行设计了封装图。其次，包括 74HC244 模块等模块以及电阻电容等存在贴片（包括 SOP，SSOP 封装）以及插接封装（包括 DIP 封装），经过无数次的布线尝试，我们尽可能的选择插接件来降低后续布线，焊接的难度。

接下来的布线部分，首先遇到的问题是 AD 的自动布线完全无法胜任如此大体量的布线工作，最开始我们根据体量希望能够采用双面板来降低布线难度，然而实验室很难提供相关的加工工具。因此我们选择单面板布线，并只能采用手动布线方式，整个布线过程浪费了整整一周的时间。

布线中的另一个方面的困难是线宽的选择，一方面 TPS54160 不到 15mil 的引脚宽度要求布线最小宽度小于 15mil，另一方面 BTN 大电流供电要求布线宽度尽可能宽，经过一系列的调整和选择，我们确定了适合负载电流的布线宽度，保证每一条支路的电流不过载。然而，10mil 的线宽虽然满足了大部分的需求，却对后续打孔，焊接的技术难度带来了巨大的麻烦。另外，开始布线时部分地方布线过密，导致出现了 VCC，GND 之间距离过小的问题，之后的 PCB 对这些部分尽可能的进行了调整。

之后是 PCB 的印制工作，在小学期开始前，我们曾请教过助教有关大体量密集 PCB 雕刻的问题，发现抵达了雕刻机的上限，之后学期内具体采用的是转印-腐蚀方式，然而这种方式对于技术水平仍有一定的要求，一方面，版面过大导致散热过快，转印不能顺利实现，因此采取的解决方式是采用多次转印的方式确保转印实现。另一方面，我们发现转印过程中由于线宽问题也可能导致一些线出现中断等问题，具体的解决方式是采用 mark 笔进行添加等，并对 PCB 的连接情况进行了细致的检查。

之后的钻孔和焊接方面也遇到了相当大的困难，一方面超过 200 个孔使得钻孔体力消耗相当大，另一方面孔径过小，焊盘过小导致钻孔伤及焊盘。这些问题我们通过自身的努力都已经尽可能的解决和避免了。焊接过程中，首先遇到的问题是 SMT 的焊接工作。以 TPS54160 为例，其宽度为 3mm，一侧共有 5 个引脚，引脚宽度和引脚之间的间隙均不超过 20mil，不

仅如此，其下部还有一个散热引脚。初次焊接出现了巨大的问题，在助教的帮助下完成了第一个芯片的焊接，然而后续的焊接也非常的吃力。这个问题的解决方式是我们取李兆基大楼找电子工艺实习的师傅学习了一天的 SMT 焊接技术，并顺利完成了相关的焊接操作，使得该项技术达到了实用层次，如图15是我们的最终成果。

焊接技术的另一个问题是插接件焊盘丢失的问题，一些插接件由于打孔的问题出现了焊盘的不完整等问题。具体的解决方式很大程度上依靠了陈崴同学优秀的焊接技术，同时，对于完成焊接的电路，进行逐点测试短路并对短路部分予以隔断。同时对可能存在的电流容量不足的线路进行了上锡保证导电性。

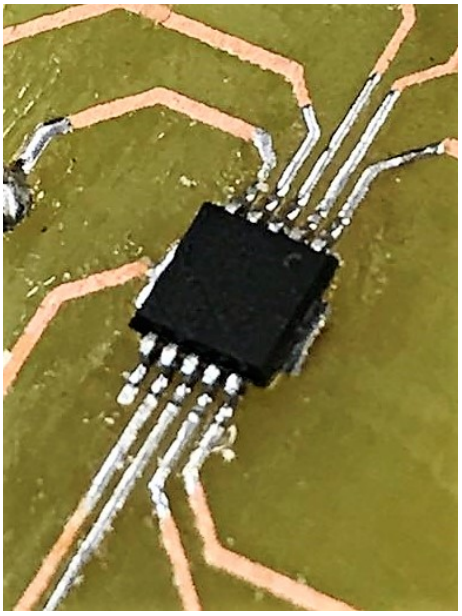


Fig. 15: 手焊 SMT 11 脚芯片 TPS54160 图

电路组装完成后，仍然出现了大量的问题，首先解决了大量的虚焊，短路的问题。其次，为了固定摄像头模块，我们在 PCB 上预留了相关位置，然而由于原理图出现镜像对称问题，摄像头前后两排引脚翻转，导致摄像头被反向上电出现了一些损坏。为了解决这个问题，我们不得不设计另一个 PCB 将摄像头引脚反置，然而，由于开始设计的过程中没有考虑这方面的问题，导致新构建的 PCB 遮挡了一些排针和引脚，这个问题的解决方法是我们另购了 L 形排针，将本来直插的排针改良为横接，基本上解决了这个问题。

在电路实际运行的过程中，我们发现由于电机驱动力不足，我们不得不提高 PWM 占空比来提升电机电流，然而，当电流过大时电源电压被拉低导致后续电路失效供电失败，这个问题的解决方法是严格控制电机 PWM 占空比，保证了电路可以正常运行。另外，我们进一步处理了电路接触不良的问题，由于具体的问题和解决方法比较简单，这里省略。

8 项目创新点

8.1 摄像头的编写

8.2 上位机的编写

我们采用 C++/Qt 编写了自己的上位机程序负责遥控，图像解算，图像显示和图像存储。相比其他小组采用的手机端下载的成型 APP，我们自行开发的程序可定制性更强，效率更高。同时，不论采用串口和蓝牙，回传的数据量解算都是相当大的，采用现成的程序根本无法完成相关任务。我们通过不断的调试，采用了多线程技术来采集，解算，显示图像，保证了上位机工作的稳定性和效率，保证上位机不成为整个项目的瓶颈。

同时我们编写了精美的用户界面，方便用户通过键盘和鼠标进行控制小车的速度。整个上位机的工作量甚至在一定程度上超过了 C++ 小学期同类型大作业的工作量，并涉及到大量的 API，如蓝牙，串口，以及 openGL 等软件工程技术，提高了我们的软件能力以及软件和硬件交互能力。

8.3 PCB 的处理

我们设计了一块整合的 PCB，包括三个开关电源，一个驱动电路和若干外围电路。采用 TPS54160 电源模块，虽然极大的增大了开发的难度，但是提高了电源的效率，比起线性电源不到 50%，甚至更低，仅为 30% 的效率，我们的理论效率可以达到 84%。同时，通过将 PCB 的整合，我们尽可能的避免了飞线等情况，提升了电路的稳定性。整个 PCB 的难度和体量均为整个课程组里最大的之一。

9 工作日志

简要的工作日志如表5所示，由于我们采用了免费、开源的分布式版本控制系统 Git，并将所有代码托管在面向开源及私有软件项目的托管平台 Github 中，我们可以很简单的管理我们的工程以及检查我们的进度，同时有效加速团队间的协作进度以及防止可能存在的错误修改，提高开发效率。详细的工作日志如附加文件所示。详细的工作日志请参阅提交记录

10 项目总结和感想

经过两周的努力以及之前的一些铺垫，我们完成了整个项目从规划设计到具体实现的各种环节，有了比较大的收获。

Tab. 5: 工作日志表

小学期之前		主要完成了蓝牙模块，摄像头模块，上位机，电路原理图和 PCB 布线的分模块的工作，并分模块进行了检验，其中上位机顺带完成了一个上位机自检测试程序。基于 WEBENCH 完成了电源模块的设计，并综合了电机驱动到一块 PCB 上，完成了复杂的布线工作，为小学期做好了相关的准备				
工作日 / 周		周一	周二	周三	周四	周五
第一周	上午	综合了假期的工作进度，摄像头模块和上位机模块进行了联调	PCB 版印刷完毕，检查相关问题以及准备打孔工作	审查 PCB 发现 PCB 出现印刷质量问题，发回实验室重新制作	赴李兆基大楼完成 SMT 以及绝大部分插接件的焊接工作	修正 PCB 问题和错误，整车组装空载测试
	下午	PCB 原理图交付实验室印刷，上位机模块发布稳定版本提供后续使用	PCB 打孔，焊接工作（包括 SMT 焊接以及插接件焊接）	赴李兆基大楼进行 PCB 打孔，刷松香以及学习 SMT 相关焊接技术并进行练习	完成 PCB 的焊接工作，检查 PCB 电路的工作性能并进行调整	赴中发电子市场购买其他电子元件，并为实验室提供相关电子元件
第二周	上午	完成基于 Lab-VIEW (myRIO) 的陀螺仪配置	小车落地遥控，检查通信协议，准备验收	验收	视频录制，器件归还	展示工作
	下午	设计完成红外测速功能	修正电路，检查蓝牙速度上线，寻找代替蓝牙信道的替代方式，通报验收	休整，准备项目结题	制作展示 PPT，器件归还	– –

首先是项目的规划，跟进方面。在项目规划过程中，我们在 7 月份就提出了一份明确的规划方案，之后的工作均按照方案实行。在合作方面，我们队友之间配合紧密，合理的采用了 Github 等团队合作，代码托管工具加速了我们的合作进程。技术水平的相近保证了我们团队分工均匀有效，每个人都可以发挥自己的特长。模块分工清晰明确，陈崴同学负责了几乎所有的 Verilog 代码的实现，而张蔚桐负责了所有的上位机实现和模拟电路部分。这种分工方式使得我们不论是合并工作量还是检查错误都有着明确的方向，也不会发生队友之间相互“甩锅”的情况。

其次是对于硬件软件电路的关系的理解，通过这个项目我们了解到，虽然相比于模拟电路，数字电路的稳定性和可展示性更强；相比于硬件底层的封装，同样工作量的软件工程，或者是顶层的综合更加“出彩”，但是模拟电路，底层封装的重要性是不可忽视的。底层电路是整个项目的基础，更好的了解底层的情况有助于我们对顶层进行更好的封装。例如，我们采用的摄像头模块是 18 引脚的底层封装，相比于其他组用的高级摄像头模块，带来了大量的工作量，然而，我们的摄像头造价更低，不考虑其他硬件瓶颈限制，我们的摄像头采集速度也更快，采集方式也更灵活。

这次课程设计，我们从底层硬件电路层面到上位机纯软件层面进行了设计，实现了一个从硬件到软件有机整合的项目，也得到了相比于之前做的软件项目不同的进步。通过和硬件层面的了解，我们了解到一些事实存在的行业瓶颈，了解到了软硬件结合的过程中硬件电路的重要性。通过实际硬件电路的了解，设计，制作这些过程，我们增长了硬件电路调试设计的很多经验，如 PCB 设计的合理性等，方便了我们之后进一步的硬件项目的设计。

11 项目完成情况

项目完成情况良好，蓝牙图像回传收到前面提到的数据量和速度的问题受限，采用串口代替蓝牙可以得到比较好的图像，一些图像如下所示，作品视频可以参考视频文件