开源机器人项目

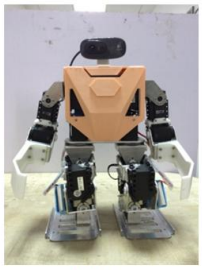
HANDS FREE

HANDS FREE 是一个面向机器人研究、开发的开源软硬件系统。她有完备与科学的框架，以优秀的嵌入式系统框架为核心，精良的电路、机械设计为支撑，帮您快速实现多种形态的机器人。本系统包含机器人导航，SLAM，计算机视觉等模块，并拥有自己上层软件和调试系统。她支持国外其他的开源项目，如 ROS, MPRT, PIXHAWK 等，这一切都为您带来了无比的便捷和快乐！

如果你觉得“哎呦不错”的话，就一起加入进来吧！！！

了解我们：


斯通/Stone


巴克


基隆/JiLong

无人机系统

最新资料和代码请到： https://github.com/HANDS-FREE

最全资料请去百度云： http://pan.baidu.com/s/1c201NC

HANDS FREE 网页介绍：

http://www.adv-ci.com/

http://www.rosclub.cn/post-265.html

http://wiki.exbot.net/HandsFree

HANDS FREE 交流群：521037187 （Hands Free Community）

小车视频展示：

http://v.youku.com/v_show/id_XMTUyODk4NTUzNg==.htm

核心技术展示：

http://v.youku.com/v_show/id_XMTU0NzgwNzc3Mg==.html?from=y1.7-1.2

购买链接：

https://shop145029875.taobao.com/?spm=a1z10.3-c.0.0.zpwB3d

ROS 学习社区推荐：

EXBOT ： http://blog.exbot.net/

EXBOT 交流群：109434898（群 1 已满） 426334501（群 2）

ROSCLUB ： http://www.rosclub.cn/ ROSCLUB 交流群 ： 184903125

EXBOT 已经有很长的历史了，里面有很多 ROS 的使用攻略，以及一些专题的深入讨论 。ROSCLUB 刚起步不久，里面有很多机器人系统方面的文章和 ROS 的使用攻略，HANDS FREE 和 ROSCLUB 是友好的合作关系，所以很多使用攻略和问题也会发布在 ROSCLUB 上。

第一章 简介

1.0 送给入门者忠告

1. 机器人是一个复杂系统，对理论和工程知识的要求综合度是非常高的，如果你没有足够的激情，不断学习的心态，独立解决困难的决心，建议你还是不要入这个神坑。
2. 即使你有上述的基本素质，机器人也不是想玩就能玩的，至少不是你有钱买个平台你就有能力玩起来的,虽然我自己也很菜,但是我还是得给新手指指方向。
我觉得如果你能满足以下条件再开始你的 HANDS FREE 开发之旅会比较好一点
（1）. C++编程功底要够好
最基本的是要能看懂或者看过 stephen prata 的 C++ Primer Plus ，如果对 C++有一定自己的理解，那么对你以后看复杂系统的代码会更好一点，
可以参考书籍： Effective_C++ More_Effective_C++ STL 源码剖析并且了解一点 boost
（2）. 对机器人的整体系统有一定的了解：
虽然你可能只是一个 SLAM，计算机视觉，机器学习等局部问题的研究者，但是如果你要玩整机那就得对整体系统从机械结构到运动学动力学，电路，嵌入式等有一些最最基本的了解。
这本书可以帮你入门，请确保你可以看懂或者看过：
自主移动机器人导论 （美）西格沃特（Siegwart,R.），（美）诺巴克什（Nourbakhsh,I.R.）
（3）. ROS 可能是大部分人入门的工具，但是在你对它还没有一定基本了解的前提下不要买我们的平台，因为你可能会失望，除非你不用 ROS 来构建你的机器人系统。
请确保你看懂 beginner level 的教程： http://wiki.ros.org/ROS/Tutorials
如果你的经验还不够多，请多看几遍。
请确保你看过比较正式的书籍，随便一本你看懂了就行：
比如你已经大概看懂 ros by example 1 indigo 前 8 章
另外我再推荐一个国外的 ROS 入门 ppt：
http://u.cs.biu.ac.il/~yehoshr1/89-685/

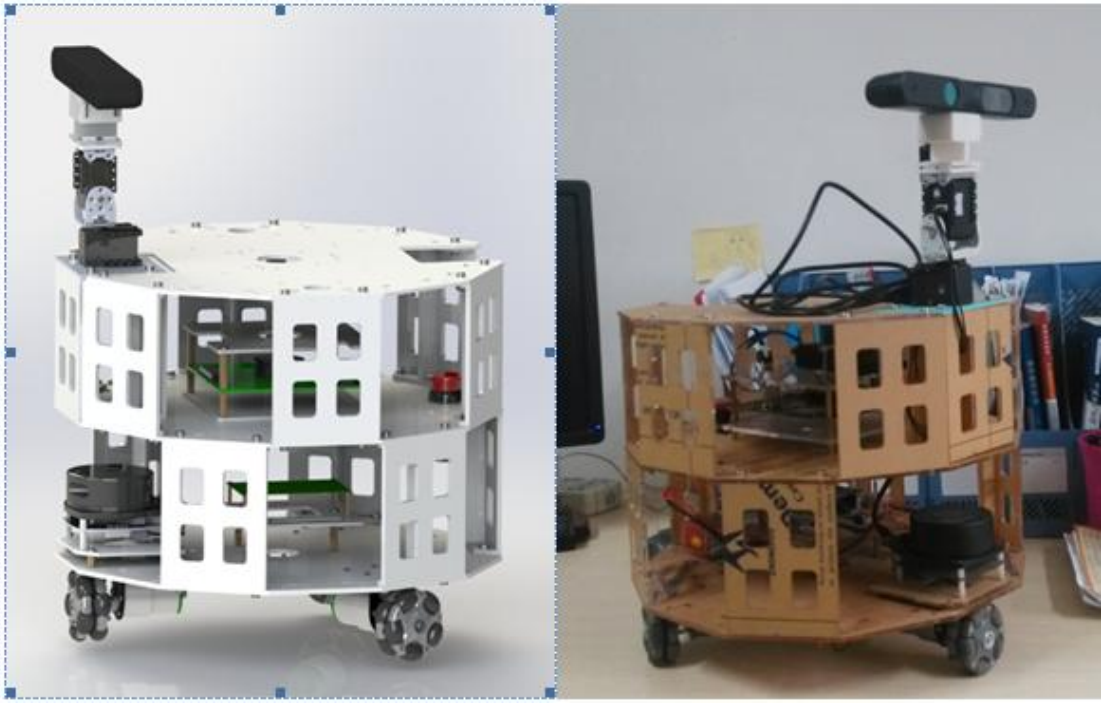
如果你以上条件你都 ok， 那么入手实物机器人就比较科学了，当然你要知道的远不止这些，很多细节的问题和工程经验我是列不完的。
PS： 以上是个人建议，我也很乐意有大神来修正或者帮忙补全。

1.1 平台介绍

Hands Free Robot 是 Hands Free Team 基于 Hands Free 开源项目建立的一系列实物机器人研究项目，目前以 3WD ，2WD 平台开源建设的重点，另外还有无人机和人形机器人方面的研究。

移动机器人由一个三轮式底盘（3WD）或者差速型底盘（2WD）和一个数字舵机云台（pan-and-tilt head）组成，适合于运动控制算法研究，机器人导航避障算法，SLAM，多机协同研究等研究，也是一个不错的 ROS 入门小神器。

目前已经发布了的平台代号是： 基隆 3WD ，基隆 2WD 和 斯通 2WD 以及即将发布的斯通 2WD_PLUS ，当然按照我们的发展轨迹，只要时间充足，我们还会陆续推出其他类型的平台，可能会有大载重的坦克型和 4 驱动麦克纳姆轮，以及还有飞行器的平台，所有的这些平台都会在一个统一的框架下进行开发。



HANDS FREE 机器人平台--基隆

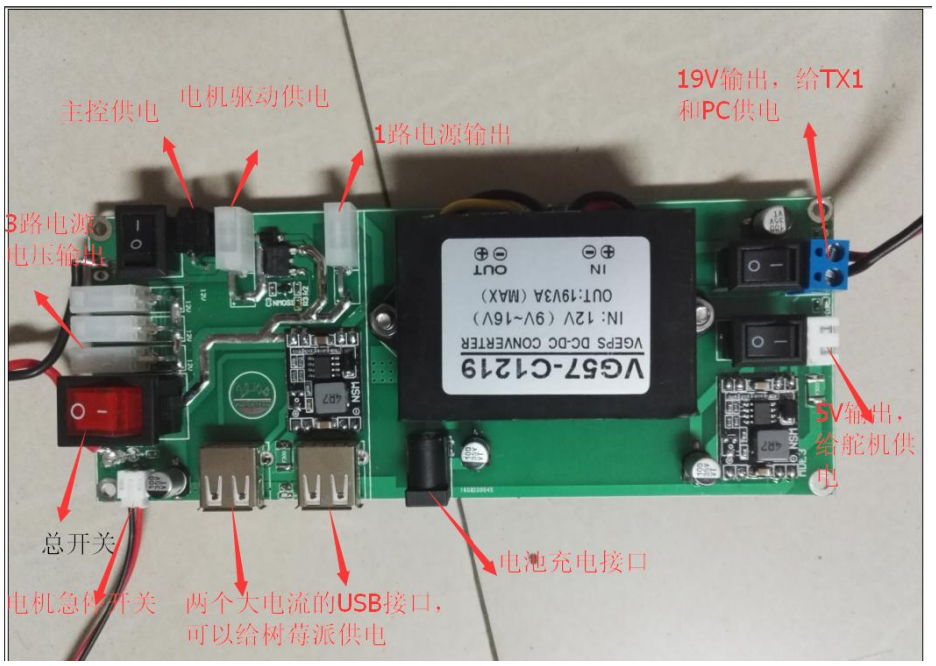
基隆：大小 35CM 直径，40CM 高度，亚克力机身，体重小于 3KG，总功率不超过 15W，电池电压 12V，额定负载能力可达 4 到 6KG，额定最大速度 0.8M/S。

HANDS FREE 机器人平台—斯通

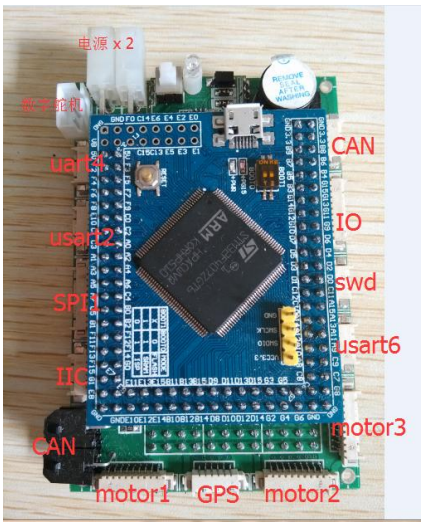
斯通：大小 35CM 直径，80CM-100CM 高度，玻璃纤维的机身，体重小于 3KG，总功率不超过 15W，电池电压 12V，额定负载能力可达 6KG 以上，额定最大速度 0.8M/S。

传感器和设备支持：

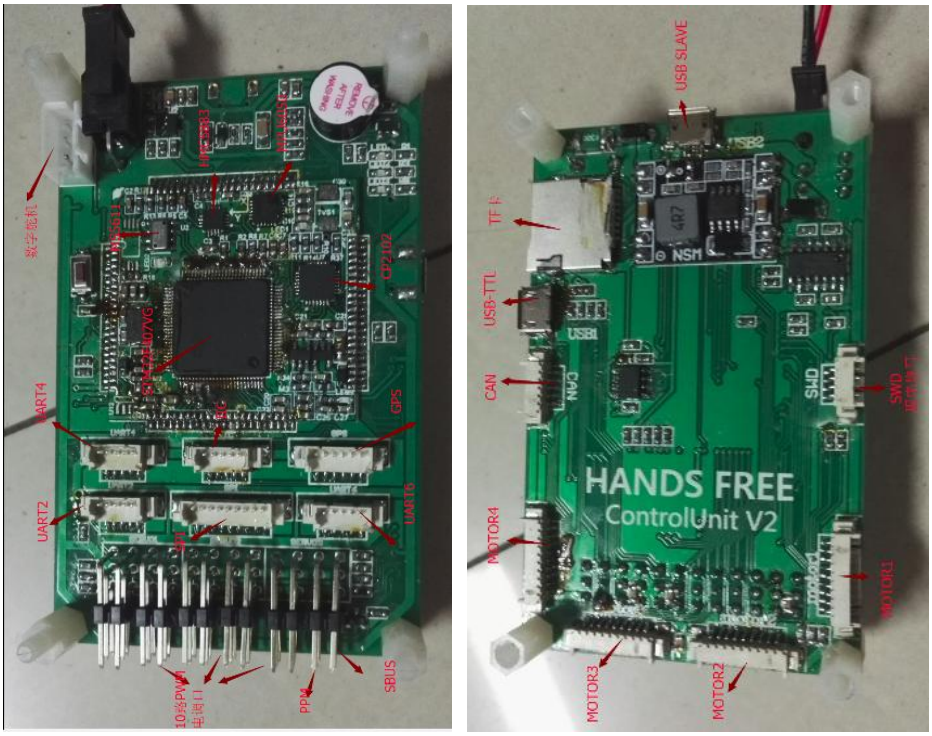
常用的激光雷达（RPLIDAR,HOKUYO），华硕的 xtion，英伟达的 TK1，树莓派，Hands Free 云台，机械设计尽量解耦，主要是方便玩家自己 DIY。由于机器人的系统的供电系统复杂，所以 HANDSFREE 专门设计了电源分配板来管理车体的供电。



嵌入式环境：



CONTROL_UNIT_V1



CONTROL_UNIT_V2

目前发布了两款控制器： CONTROL_UNIT_V1 和 CONTROL_UNIT_V2，详细介绍可以参考后面的文档。简单的说是底层实时性控制是 STM32F4 跑了一个实时性操作系统 UCOSIII 实现的，通过自己定制的机器人通信协议 HFLINK（ROBOLINK）和上位机通信，接口：USB 串口。

电机参数：目前斯通和基隆车型的电机是一样的。
电机电压：12V
电机最大功率：7W
电机编码器线束： 390
最终脉冲数： 390*4 = 1560 （即轮子转一圈单片机收到 1560 个计数，用于速度环足够了）

1.2 平台运动原理

3WD 平台运动原理：

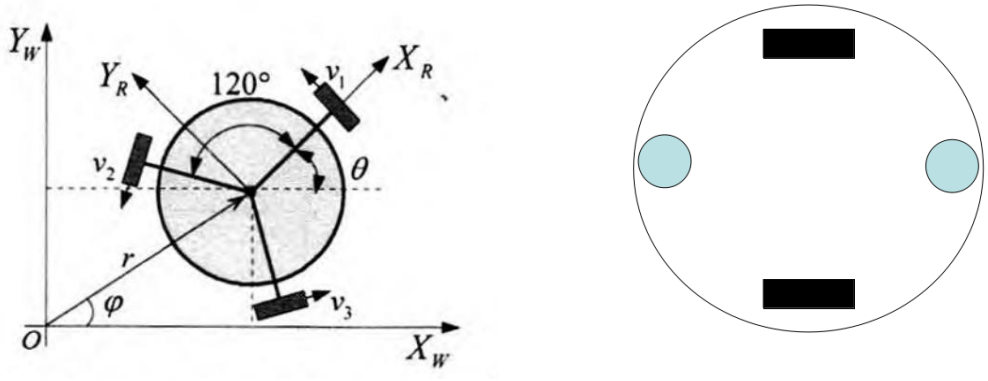


图 1-3 3WD 和 2WD 运动原理

3WD 采用 3 个电机驱动的 3 个全向轮作为其运动机构，优点是具有 3 个运动自由度，即前后，左右，旋转，因此可以在一个平面内朝任何一个方向运动，所以该机器人运动过程中机动性是很强的，最终可能会在自主运动过程中穿越狭小空间表现出优势；缺点是由于存在力的抵消，所以运动效率不高，而且跑起来有轻微的抖动，这是由于轮子结构引起的，当然也可以通过购买昂贵的全向轮来解决这个问题，还有就是在运动过程中遇到电线或者电梯门槛会卡住。

2WD 平台运动原理：

2WD 差速平台除了底盘的运动结构和 3WD 不一样，其他的都一样，采用 2 个驱动轮，前后各一个脚轮作为底盘运动机构，具备前后和旋转的运动能力，由于比 3WD 多了一个支撑点，所以负载能力强了一点，可以到 6KG，而且运动效率比 3WD 高一些，但是没有 3WD 的全向运动特性。

虽然视频中演示的是 3WD，但是 2WD 的效果同样好，个人感觉全向特性也不是强需求，就好像人一般情况下不会横着走路一样。综合比较之下，HANDS FREE 之后还是以 2WD 的开发为主，以及之后发布的平台也是以 2WD 为主。

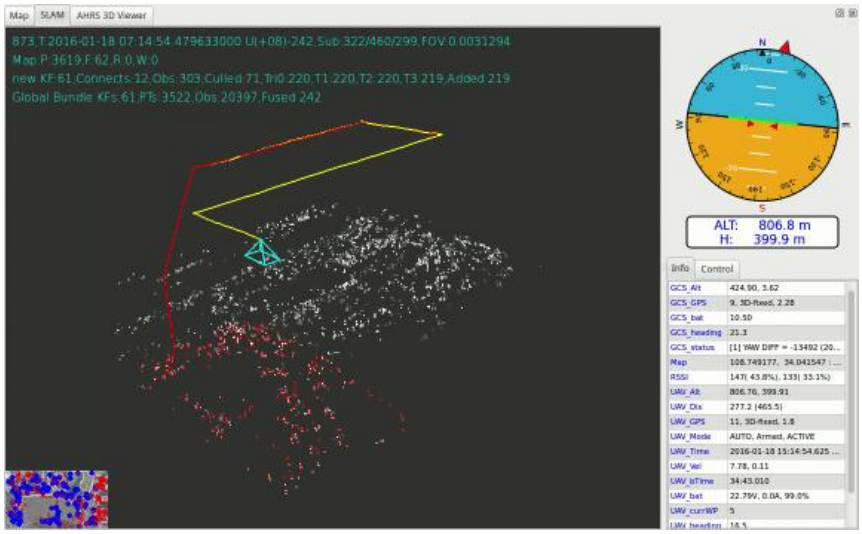
1.3 HANDSFREE 的无人机系统介绍

无人机组的研究成果是 HANDS FREE 团队的核心技术力量，该研究项目由西北工业大学布树辉教授领导，并取得了诸多成果，详细介绍请关注布老师的个人网站： http://www.adv-ci.com/

由于目前无人机系统时我们团队的核心技术力量但不是开源的重点，所以在这里就不做过多介绍，如果你喜欢可以你去网站或者通过其它途径了解更多。

综合数据链路

地面站软件



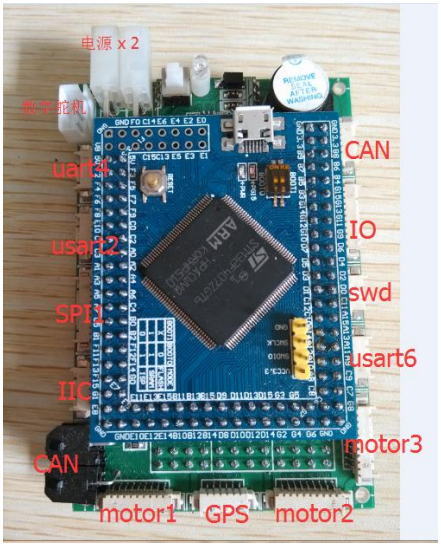
第二章 硬件介绍

2.1 HANDSFREE 控制器介绍

使用 STM32F4 为下位机，为了方便，我们给它起了个名字叫 Hands Free Control Unit 。目前发布了两款 Control Unit V1 和 Control Unit V2，正在销售的是最新版的。

Control Unit V1:

使用的是第三方设计的 F4 核心板，关于核心板详细资料请看他们的店铺 https://item.taobao.com/item.htm?spm=a1z0k.7385961.1997985097.d4918993.F7657F&id=523383164199&_u=e1g8nedo4164



CONTROL_UNIT_V1

基于 STM32F407ZGT6 为主芯片
LQFP 引脚个数：144；
内核：Cortex™-M4；
工作频率：168MHz；210 DMIPS/1.25 DMIPS/MHz；
存储资源：1024K Byte Flash ，192KByte SRAM；
1 路 USB 接口
1 路 SD 卡接口

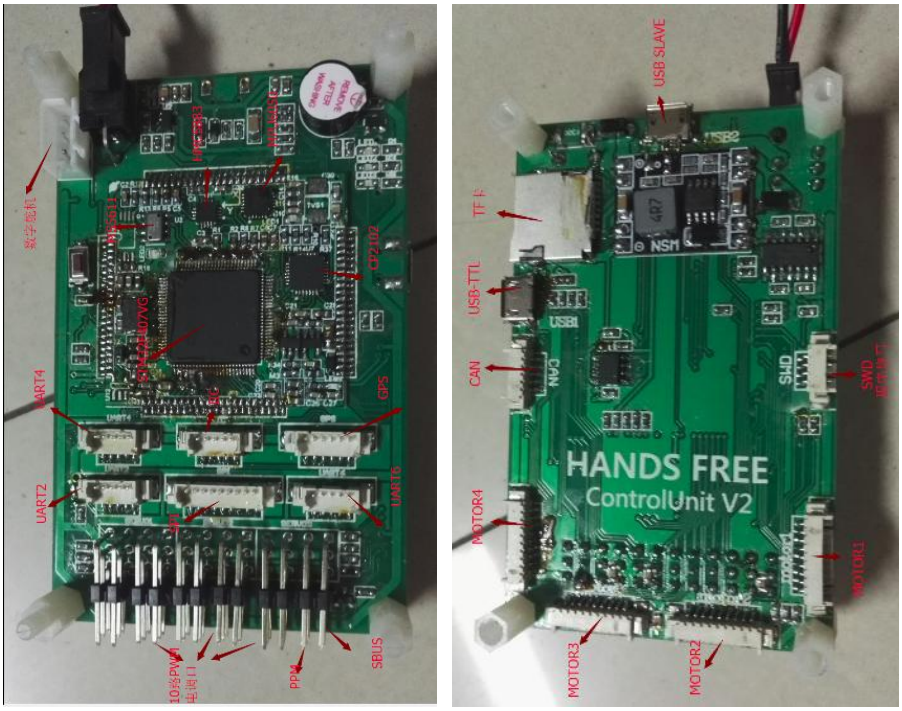
Control Unit V1 资源列表:

MPU6050 加速计陀螺仪
HMC5883L 磁力计
MS5611 气压计
1 路数字舵机控制器 ：支持 AX12，优必选等型号的数字舵机，用于控制 Hands Free Team 开发的数字舵机云台，或者控制 Hands Free 人形机器人

3 路电机控制接口：支持 Hands free Team 开发的电机驱动，可用于搭载 Hands Free 3WD 轮式机器人或者 Hands Free 2 轮平衡车
10 路电调口：用于控制模拟舵机或者电调，用于搭建多轴飞行器
1 路 SBUS 采集和 1 路 PPM 采集：用于采集航模遥控器的信号，PPM 接口还可用于扩展空速计
一个 GPS 接口：支持 APM 接口的 GPS USART3
1 路 CAN 总线：板载 3 个 CAN 接口，可用于通信，或者扩展 Hands Free IMU 高精度单轴陀螺仪
1 路 IIC 接口：用于扩展 IIC 设备
1 路 SPI 接口
1 路 GPIO 扩展接口：提供了 6 个普通 GPIO 引脚
三路串口接口： USART2 UART4 USART6
一个 USB 串口：采用 CP2102 USB 串口芯片 USART1，可 USB 线升级固件和通信
1 个大容量 EEPROM 存储
1 个 SWD 程序烧写接口

Control Unit V2:

整个板子都是自主设计的，4 层板设计的核心板和 2 层板的底板，集成度更加的高，板子的大小只有身份证的大小，而且功能比 V1 更加的丰富，特别是增加了一个电机控制口，总共是 4 个电机控制口，基本上能支持常用的车型。



CONTROL_UNIT_V2

可以说 Control Unit 是飞控的配置，用来玩小车是有点土豪。

2.2 HANDSFREE 电机驱动介绍

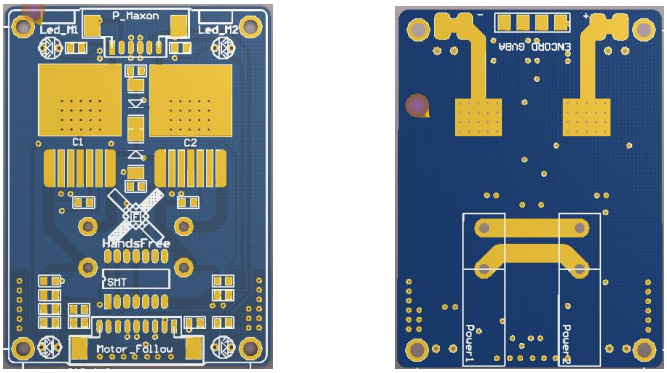
Hands Free Motor Drive 是 Hands Free Team 根据 Hands Free 开源项目的标准开发的一款电机驱动器，是 Hands Free 平台的组成部分之一，板载 4 个 led 分别显示电机供电，编码器供电，电机正转，电机反转。目前共发布了两款 Motor Drive V1 和 Motor Drive V2。 正在销售的是最新版。

Motor Drive V1 电路综述：

前几套版本：采用市面上一般的设计思路，使用 BTS7960B 半桥驱动芯片的组合，通过 74hc244 线路驱动器，将 3.3V 抬高到 5V，连接到 bts 驱动芯片上的相应引脚。由于采购不到较窄(74hc244 市面上多为宽体面积较大)的非反相驱动器，我们采用了二级非门的结构(特性相对于 74hc244 较差)代替 74hc244。另外将 bts 电机驱动芯片的 GND 和 STM32，非门的逻辑 GND 用磁珠连接，简单地抑制部分干扰。

在后续的驱动升级中，驱动芯片将替换成 BTN7971B ，性能也会有一定提高。此外还要废弃直插型 LED、非门，引入光耦等其他较为可靠的隔离方式，电机输出接头也需要更换，对于铜柱的尺寸，待统一之后，可能要换为 3mm。

(对于涉及模拟电路部分，我们团队还没有一个较为专业的人士，只能靠实验以及吸取网上经验的方式，如果有同学对这块比较熟悉，也欢迎将方案提供给我们。当然，作为开源的一部分，也支持大家自行下载 PCB 原理图，开板用于个人学习使用。)



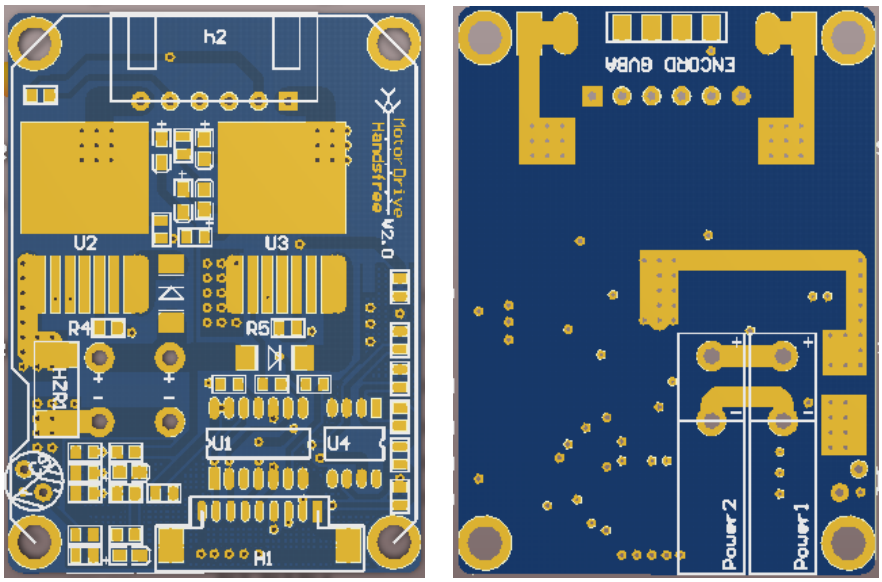
Motor Drive V1

详细参数：

铜柱间距： 1200mil × 1600mil (40mil = 1mm) — 2MM 铜柱
板子大小： 1400mil × 1900mil
输入电源大小： 5~35V
输入接口： [电源输入接口][扩展电源接口][HANDS_FREE 标准电机接口]
输出接口： <主方案> [瑞士 ESCAP 16 电机接口] - MOLEX1.25mm 6P
<备选方案> VCC5.0 GND 编码器 AB 相 焊盘*4
电机+ 电机- 焊盘*2
状态显示：
[电机正反转 LED 显示] [输入电源 LED 显示] [信号电源 LED 显示]

PS：这系列芯片除了大电流，还很稳定，虽然用到 3WD 这种小型电机上有点大材小用，不过耐用是个优点，看在搞科研搞学习的份上就不惜成本了呗，而且以后我们开发大型载重的平台时也可以用这个驱动，当然用它搞 DIY 也是不错的选择。

Motor Drive V2 电路综述：



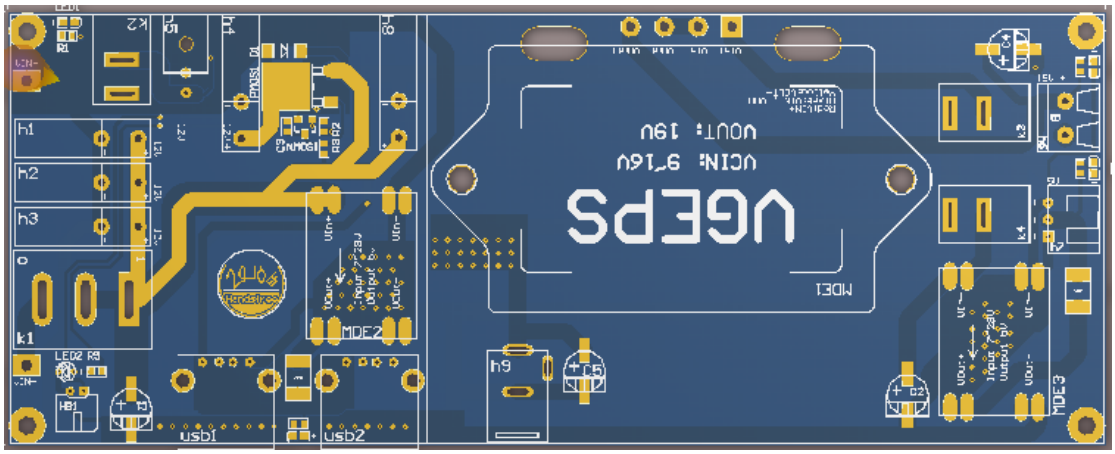
Motor Drive V1

与 V1 相比， V2 多了电流检测和防反接的功能。
电流采样：0.34V/A Max:<6A(瞬态) 建议软启动下使用。

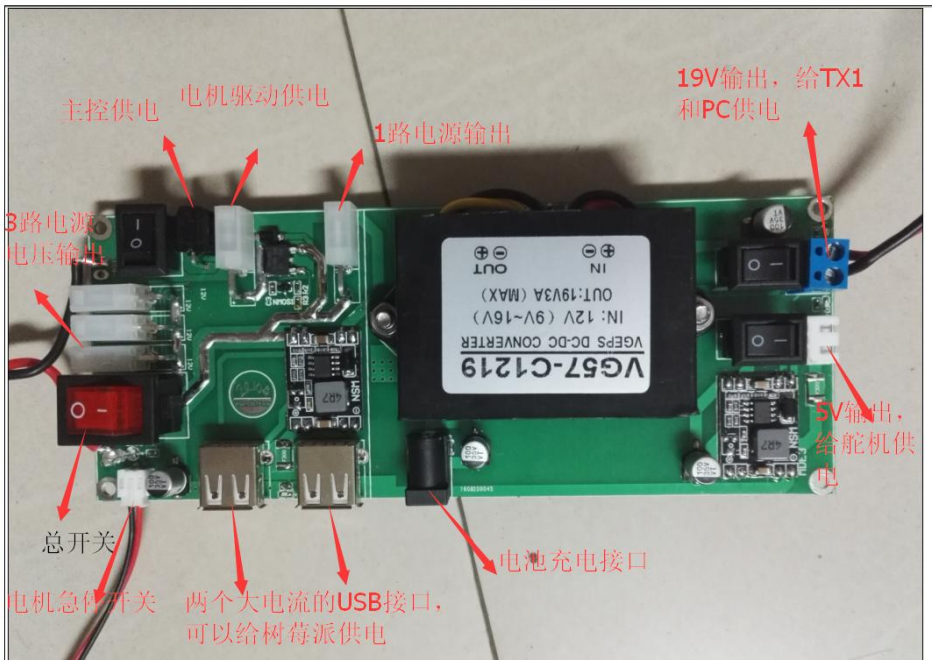
2.2 HANDSFREE 电源分配板介绍

Hands Free Power Manager V1 是 Hands Free Team 根据 Hands Free 开源项目的标准设计的一款电源分配板，附带多路开关和电源转换功能，满足机器人多样的电源需要。

原理图片：



实物图片：



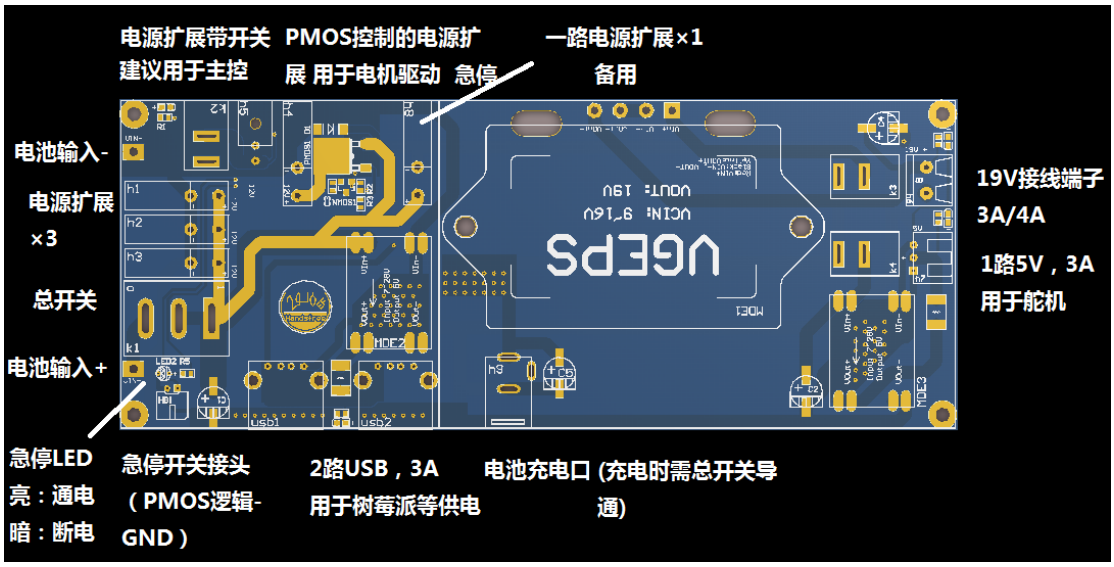
板载模块：

7~24V 转 5V, 3A 电源模块 ×2

9~16V 转 19V，3A/4A 电源模块 ×1

板载电源口：

PMOS 控制电源扩展口 ×1
[开关]5V，3A(舵机供电)电源口 ×1
[开关]19V,3A/4A(TX1 供电，小型 PC 供电)电源口 ×1
5V,3A 双 USB 母头(树莓派供电)
普通电源扩展口 ×4
[开关]电源扩展口(主控供电) ×1

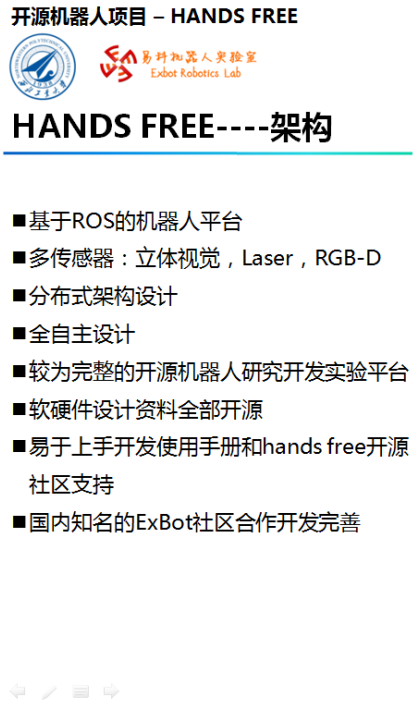


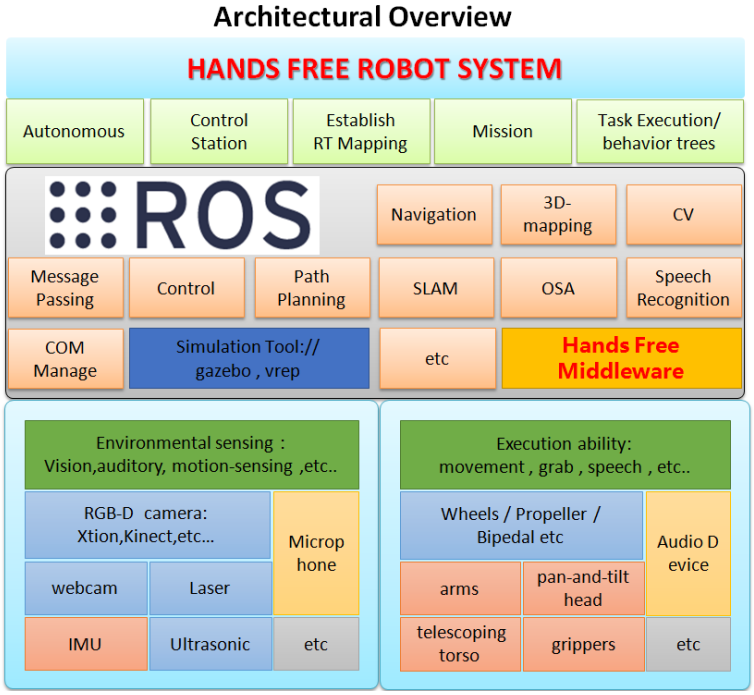
功能说明图

第三章 软件架构介绍

3.1 整体框架介绍

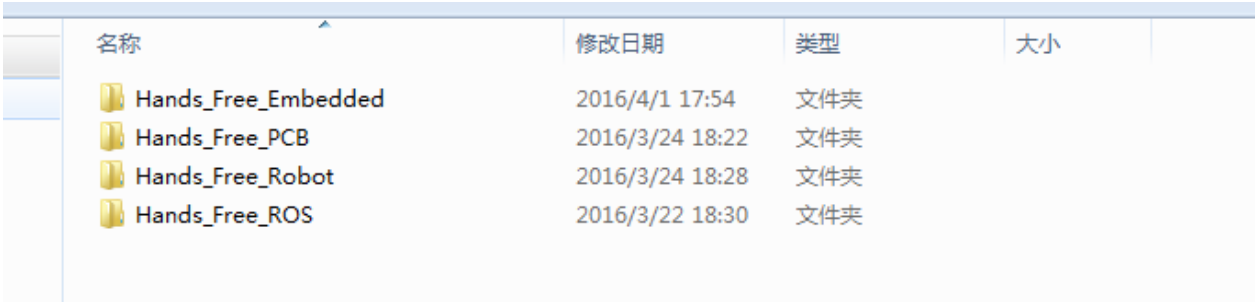
先放一张我们的美好蓝图：
路漫漫兮其修远，还请各位憋吐槽





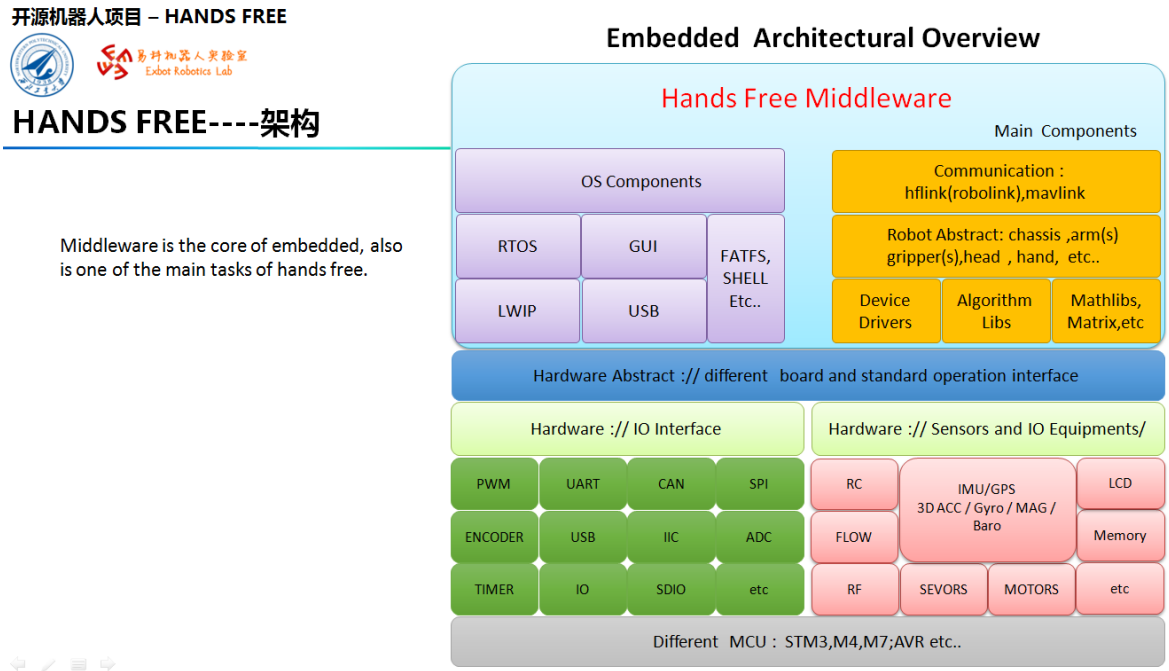
虽然以上我们都有所涉及，但是要完善也挺困难的，所以我们主要是搭建一个平台而不是都自己去实现，我们尽可能使用各种工具来实现自己的目的。

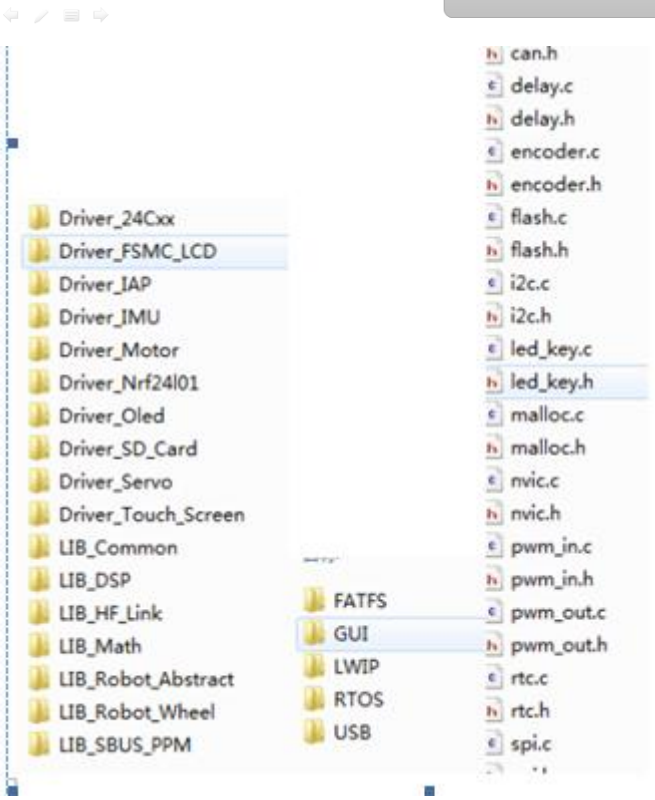
资源总结： 目前是这几大组成部分，以下一一介绍一下这些资料。



1. Hands_Free_Embedded

Hands_Free_Embedded 是目前 handsfree 的核心，里面都是各种驱动包和算法库和 RTOS，也就是以下内容　ps：移动平台的嵌入式部分目前有操作系统版和裸奔版的固件，使用的是 UCOSIII 系统。





机器人可能使用到的设备非常多：
各种伺服设备：数模舵机，直流电机，三相电机，步进电机等
各种传感器：加速计，陀螺仪，磁力计，超声，红外，GPS 等
各种 IO 设备：交互类的 LCD，触摸屏， 遥控器，蓝牙设备， EEPROM，SD 卡等
而且每种设备还会有许多性能不一的不同型号，嵌入式软件方面则需要各种驱动代码，数学运算库，以及和机器人密切相关的控制算法，运动学和动力学模型 等，当然还需要很多功能软件好维护体系的正常运行，为了丰富软件的功能和提高软件的可靠性，最好再有个操作系统以及其组件 USB,文件系统，TCIP/IP 等。
没有一个比较健全的底层系统，或许简单的研究不会觉得有什么问题，但是一旦系统变得复杂后开发调试就会大大受阻，从而限制你去学习和研究复杂系统。当然 HADNS FREE 已经有一个比较健全和鲁邦的开源系统了，并且会继续完善，这也是我们推荐初学者了解 HANDS FREE 的原因。
所以开源的真正重点是在这里，希望通过 handsfree，能建立一个取之不尽的驱动库和算法库，而这一点，在国内是很少有的，基本上在各大论坛也只能零散的找到一些别人用过的代码，当然，handsfree 的发展，我们也无法预测，所以若是让大家失望也求憋吐槽

为什么把**建设重点**放在，底层库：
第一，因为现在没有能力去创建一个机器人的上层框架譬如机器人导航，SLAM，计算机视觉种种，这方面我们也在探索。
第二，没有必要，国外已经有很多好的开源项目，诸如 ROS,PIXHWAK，MRPT，机器人仿真可以用 gazebo 和 vrep，甚至底层也有 arduino 这样的存在，所以 handsfree 主要是一个交流圈，以及帮助入门者去使用这些工具
第三，阻碍初学者入门和一些理论研究者去实现自己的想法的，往往是底层平台的问题，并不是每个研究者都是 DYI 硬件大神，或者有些硬件大神也需要工具建立快速原型的时候，就需要 handsfree 的这个底层库了，其实这和 ROS 一样，装几个包，roslaunch 一下就可以感受机器人是个啥玩意一样，装好 handsfree 的底层库，调用一下，你就可以让机器人动起来，从而实现自己的 DIY，当然前提是要会 C++。

这个底层库和 arduino 不一样的是，handsfree 底层库和硬件主要都是为机器人服务的，它有自己的硬件接口标准和实现，而不是 arduino 那种排针和杜邦线满天飞的节奏，而且我们使用的是 stm32F4，自然性能上也高出许多倍，这里不是比较谁优谁劣，只是 handsfree 有自己的不一样。

2.Hands_Free_PCB

Handsfree PCB 囊括了 schlib、pcblib 和 pcb 文件及标准三大部分，其中 schlib 和 pcblib 规范了一些常用的 PCB 元器件及命名规则。

Sch Library:

CHP_XXX	芯片类	
ELE_XXX	普通元件类	例：电阻电容开关等
HDR_Header_XXX	插头接口类	
MDE_XXX	模块类	例：MPU6050 NRF24L01
ELE_Standard_XXX	[规范常用]	普通元件类
HANDS_FREE_XXX	[规范常用]	接口类

Pcb Library:

CHP_XXX	芯片类	
ELE_XXX	普通元件类	例：电阻电容开关等
HDR_XXX	插头接口类	
MDE_XXX	模块类	

后缀规则：

[大小规格]_[D/P][V/H]_X*X

D：孔类封装　　P：贴片封装

V：竖直方向　　H：横向

例：HDR_Header_2.54_DV_1*2

>>　2.54 间距 直插 直排 1*2P 排针/排座

Pcb Document:

一个完整的 Pcb 文件应该包括 PrjPCB, PcbDoc, SchDoc 及其其使用资料(原理图.pdf, step 文件, 依赖 Lib 版本，相关原理说明等)。

我们希望 PCB 部分也能够像代码一样，在个人学习的前提下，相互地分享和借鉴，避免再重复地制作库和封装、查找基本电路原理等工作，把更多的精力放在 PCB 的布局美观化和更高深的电路原理层面上。而由于电路部分的开发周期的局限性，PcbLib 的更新周期也不太稳定，所以也更需要大家的加入和分享。

3. Hands_Free_Robot

关于 handsfree 的各种机器人实体的资料都在这个目录下，目前有三款平台，3WD,2WD,和人形。文件包括机器人的全部机械模型资料以及机械的说明和标准，以及机器人的使用手册。

Handsfree 主要是完善底层库，和基于底层库搭建更多好玩的平台。所以后面还可能有其他的平台，，也欢迎玩家使用 handsfree 的标准搭建平台，在这里分享你的设计。

4. Hands_Free_ROS

这个文件夹全是 hands free 移动平台在 ROS 下的实现代码，里面有基于 hflink 的串口，udp 底层驱动，机器人的 urdf 模型，在 3wd 在 gazebo 和 vrep 里的仿真案例，以及 3wd 在真实世界的导航和 SLAM 的代码，使用 xtion 的一些 ros 案例等等。

Hands free 像 turtlebot 一样封装了 robot hardware，增加了 ros controller，所以不管是实物和仿真之间的切换，还是从 3wd 切换到 2wd，甚至切换控制方式都是很方便的，具体会在 ROS 篇介绍。

关于 ROS 的玩法，推荐国内一个很不错的博客 exbot
http://blog.exbot.net/
以及另外一个社区 ROSCLUB：
http://www.rosclub.cn/

3.2 购买平台和平台搭建方案

HANDSFREE 的淘宝链接：
https://shop145029875.taobao.com/?spm=a1z10.1-c.0.0.nTuSbH

机器人的设备种类繁多，由于我们精力有限，而且主要是搞软件和方案，所以目前只出售整机，而且尽可能简化了出售的东西，这就意味着买家要自己去单独选购配件。
发货单：机械底盘，电路板全套，电子线材齐全，电池和电源分配板一套（包含充电器），顺丰快递包邮，客服现在是虚设的，大家有问题直接在群里问吧。
你收到快递后还需要买导航配件和上位机板才能跑起来。

1.导航配件

如果你没有经费，你可以只选择激光导航或者视觉导航，建议入门还是从激光导航开始，不然你的学习难度就会比较大了。
2D 激光雷达 rplidar 价格：2399　有钱的可以买 hokuyo 激光雷达
https://item.taobao.com/item.htm?spm=a230r.1.14.1.8EDGgF&id=960853638 8&ns=1&abbucket=16#detail
RGBD camera xtion pro live　价格：1099
https://item.taobao.com/item.htm?spm=a230r.1.14.1.dLh6dW&id=522922112 022&ns=1&abbucket=6#detail

2.上位机板

可以直接上笔记本的啊，只不过把笔记本放在车上调，真是累的慌，你可能需要坐在地上调了，由于我们自己要搞多机协同，自然就不能上本了，所以推荐有资金的买一个 TK1 或者买 TX1 ，或者买个英特尔的 edsion，只要 799 ，树莓派我们也试过，不过目前没试出什么好效果，当然了，我们目前主要还是笔记本和 TK1 上测试代码，至于其它的以后有空再说
Arm 板上位机 jetson nvidia tk1　价格：1600
https://item.taobao.com/item.htm?spm=a230r.1.14.19.qHoD6f&id=40502280 534&ns=1&abbucket=6#detail
无线网卡，给 jetson nvidia tk1 用的
https://item.taobao.com/item.htm?spm=a1z09.2.0.0.uu4eB6&id=4191122384 8&_u=t1g8nedoeba8

3. 其它配件：

一个 HDMI 转 VGA 的线，自备一个显示器一个，一个路由器（用于建立 TK1 和笔记本通信的局域网），TK1 只有一个 USB，可能还需要一个有源 USBHUB，可能还要一套无线键鼠....

第四章　组装

4.1 机械组装

4.2 电路组装

4.3 固件升级

新一代的平台是组装好出售的，固件升级请参考 OpenRE 下的手册。
https://github.com/HANDS-FREE/OpenRE/tree/fe0746295762fb1883fa3e08fe9b75337d6987d6

4.4 简单测试

烧好固件后（出厂是烧好的），基本的工作就完成了，接下来你就可以用车来跑导航了，不过在跑导航之前，我们可以测试一下车能不能动。

如果你有 SBUS 遥控器，你可以接上去体验一下，或者你还可以在 ros 环境下使用 HANDS FREE TEAM 写好的驱动，用键盘去测试，看车能不能正常的个个方位移动。

如果以上一切正常，那么恭喜你，你可以开始你的移动机器人开发之旅了，在这之后你可能还会遇到很多问题，不过不用担心，HANDS FREE 或许是个值得依赖的平台，当然，变成大神也不是一蹴而就的，我们的精力也有限，有些问题还需要你自己去探索^_^（大神请无视这句话）

第五章　ROS 下开发

在 ROS 测试之前，我想在强调一下你已经看过第一篇的预防针的内容了。

系统环境 Ubuntu14.04 + ROSindigo

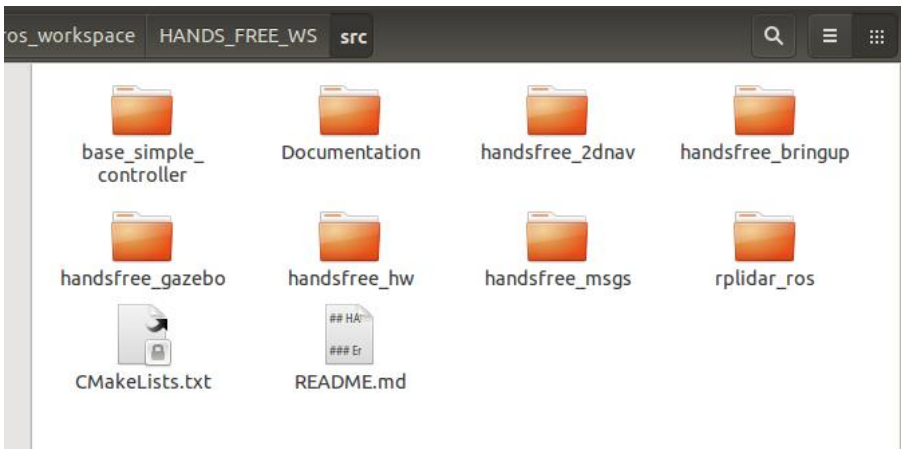
5.1 配置 HANDSFREE 的 ROS 开发环境

第一步：

新建一个 ros workspace 没并且把源码下载放在 src/目录下

https://github.com/HANDS-FREE/ROS_DEMO

最终效果如下 src/ 目录下的文件



第二步：

运行 src/Documentation/environment_config.sh 安装依赖包

第三步：

把 handsfree_hw/src/main.cpp 的第七行的配置文件路径 "/home/kedou/ros_workspace/HANDS_FREE_WS/src/handsfree_hw/config.txt" 改成你自己的路径

第四步：

编译： 在你的 ros workspace 下运行 catkin_make

退到 src 目录外打开终端编译：

```
HANDS_FREE_WS> pwd
/home/kedou/ros_workspace/HANDS_FREE_WS
HANDS_FREE_WS> catkin_make
```

编译成功：

```
hw.cpp.o
[100%] Building CXX object handsfree_hw/CMakeFiles/handsfree_hw_node.dir/src/hf_
hw_ros.cpp.o
Linking CXX executable /home/kedou/ros_workspace/HANDS_FREE_WS/devel/lib/rplidar
_ros/rplidarNodeClient
[100%] Built target rplidarNodeClient
Linking CXX executable /home/kedou/ros_workspace/HANDS_FREE_WS/devel/lib/rplidar
_ros/rplidarNode
[100%] Built target rplidarNode
Linking CXX shared library /home/kedou/ros_workspace/HANDS_FREE_WS/devel/lib/lib
base_simple_controller.so
[100%] Built target base_simple_controller
Linking CXX executable /home/kedou/ros_workspace/HANDS_FREE_WS/devel/lib/handsfr
ee_hw/handsfree_hw_node
[100%] Built target handsfree_hw_node
HANDS_FREE_WS>
```

同时把你的 source ros workspace/devel/setup.bash 放到.bashrc 文件里
这一步，在 ros 初级教程里有介绍，不知道的自己去看。

5.2 让小车动起来

Tips ： 每次打开新的终端运行 ROS 命令之前你都要 source 一下你的 workspace。或者你可以把它加进 .bashrc 具体请看 ros 的 wiki 基本教程。

第一步：

用 PC 的 USB 连上主控的 USBTTL 接口，请记得轻轻地插进去，别太用力。确保正常上电，此时你听到蜂鸣器响了一下并且灯开始不停的闪。确保你的电路，电机，驱动都连上，急停开关是打开的（驱动的灯是亮的）。然后就可以开始测试了。

第二步：

先打开终端运行：
roslaunch handsfree_hw handsfree_hw.launch
仔细的看下这个界面，确保没有问题，如果出现 timeout 的问题，则说明不成功。

使用的是 HNADSFREE 的控制器并出现 timeout 的：

1. 串口没插好或者不是 ttyUSB0
2. 权限不对
3. 固件烧写错了
4. ubuntu 系统问题

使用的不是 HNADSFREE 的控制器并出现 timeout 的，除了之前的 4 点还可能

是你的 USBTTL 芯片不支持 1M 以上的波特率。此时可以通过降低波特率来解决。只要第二步没问题，说明硬件都调通了，剩下的基本不会有什么问题。

第三步：

在打开新的终端运行

roslaunch handsfree_hw keyboard_teleop.launch

如果一切正常的话，此时你就可以像移动 ros 乌龟一样，用键盘控制你的小车了。至此，平台硬件测试完毕，剩下的就看软件发挥了。

5.3 建图导航

在正式导航之前，你要建图，因为你下载的文件里只有我们实验室的地图，而你要建一个你身边的地图才能玩起来。

建图：

（1）运行机器人抽象节点：

roslaunch handsfree_hw handsfree_hw.launch

（2） 运行激光节点

如果你用 rplidar：

roslaunch rplidar_ros rplidar.launch

如果你用的是 hokuyo：

roslaunch handsfree_bringup hokuyo.launch

（3） 运行建图节点：

roslaunch handsfree_2dnav move_base_gmapping.launch

(4) 打开 rviz，并选择 handsfree_2dnav/rviz/ HANDSFREE_Robot.rviz 配置文件

rosrun rviz rviz

然后应该可以正常建图了，

(5) 保存地图：

rosrun map_server map_saver -f my_map

生成了 2 个文件.pgm　和 .yaml，
地图就保存到了当前目录，建议你把地图都放在 handsfree_2dnav/map/下便于统一管理。

导航:
（1）运行机器人抽象节点：
roslaunch handsfree_hw handsfree_hw.launch

（2） 运行激光节点
如果你用 rplidar：
roslaunch rplidar_ros rplidar.launch
如果你用的是 hokuyo：
roslaunch handsfree_bringup hokuyo.launch

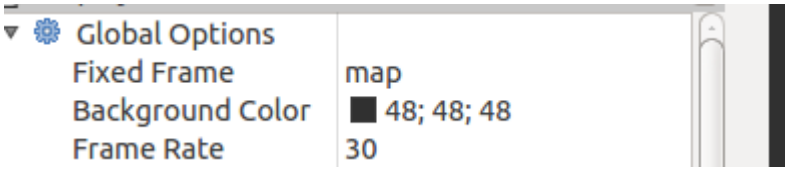
（2） 运行导航节点：
先把 handsfree_2dnav/launch/move_base_amcl_5cm.launch
如下：

```
<launch>
  <master auto="start"/>
  <node name="map_server" pkg="map_server" type="map_server" args="$(find handsfree_2dnav)/map/lab.yaml" respawn="false" >
  </node>
  <include file="$(find handsfree_2dnav)/move_base_config/amcl_node.xml"/>
  <include file="$(find handsfree_2dnav)/move_base_config/move_base.xml"/>
</launch>
```

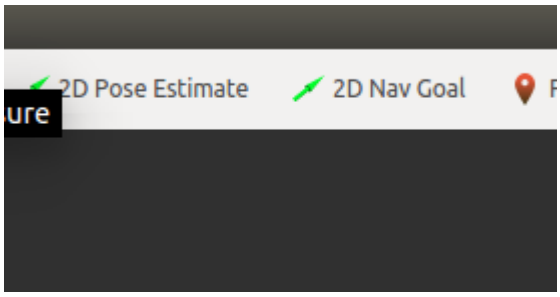
把 args="$(find handsfree_2dnav)/map/lab.yaml"　改成自己刚刚建的那个图，保存更改。

然后打开终端运行：
roslaunch handsfree_2dnav move_base_amcl_5cm.launch

(4) 打开 rviz，并选择 handsfree_2dnav/rviz/ HANDSFREE_Robot.rviz 配置文件
rosrun rviz rviz
并选择 map frame 。



（5）此时你看到你的机器人在你刚刚建好的这个地图上，但是地图中显示的机器人位置可能并不是你机器人真实的位置。



于是，先用左边和这个 pose estimate 初始化你机器人的位置，也就是调整机器人在地图中的位置。让他和现实一致。然后再用 右边的 2d nav goal 给机器人发一个比较近的(大概一两米左右)目标位置。 如果一切正常，机器人都会自己跑到你给的那个位置去。

好了，剩下的以后可能会在写，不过更重要的还是自己去探索学习，也希望你可以在研究过程中记录下你的一些心得分享到 HANDS FREE，愿意的话可以帮忙完善此教程。

常见问题

这一篇是一些常见问题答案的总结，每一个开发者都可以在此留下你纠结一个问题并成功解决后的心得。你把你的遇到的问题和答案以及心得写好以后保存成文档命名为“常见问题和解决方法”发给群主，我就会整理到手册的该篇下。我先开个头，回答几个我觉得有必要的问题和遇到最多的问题。也希望后面有人不要再问这种瞬间暴露你水平的问题了。

问题一： 我们到底是一群什么人。

由学生老师和社会人士组成，主要是学生，绝大部分是研究生：

有西北工业大学，中国科学院，厦门大学等各大学校的，研究方向不一，自动化，软件工程，计算机视觉，机器学习，集成电路，SLAM 等等都有，我们都是好朋友，从本科开始基本都是全国机器人大赛 ROBOCUP 里面的老司机，也参加过机器人国际赛以及 robomaster 这类比赛，获奖无数。然而那都是以前的事了，现在基本是以科研为主，一般的比赛没兴趣。

所以我们和一般网上卖平台的是不一样的，不管是初衷还是技术含量还是我们的将来的目标，都不是一个人生观的，相信你可以通过我们的介绍文档感受到这一点。

问题二： 我们是为了挣钱吗。

想挣钱，但不是为了。 我们的学习和科研任务都很重，并且都是和机器人有关的，顺便搞 HANDS FREE 的初衷是为了交流技术并未社会做点贡献，这里强调了顺便，因为反正我们已经经历了千辛万苦折腾出来了，那独乐了不如众乐乐。实际上我们研发测试的支出远大于我们卖平台的收入，所以挣个毛线的钱，除非能卖出许多许多台。

作为 leader 的我的心态就是，别人爱买就买，不买就算了，所以那些打着买平台的招牌和我扯家常，讨价还价，问些犹豫不决毫无意义的问题，或者手册和淘宝页面上说的很清楚的问题，我基本不理。 所以在这里说句不好意思，并给出我们的苦衷，也希望后面的买平台人尽量问有实际意义的问题，我肯定乐意回答。

当然术业有专攻，我更喜欢和你交流，互相学习，而不是一味的回答你那扣不起我回答欲的问题。

问题三： 平台能搞 SLAM 吗，视觉导航吗，或者平台支持 xtion 吗，支持 kinect 吗等等。

在这里我给出肯定的回答： 能，可以，支持。

凡是问此类问题的人，都是没提前预习功课的并且瞬间暴露水平的，因为平台本身只是个执行机构，它不依赖也不限制你使用什么传感器或者你在它身上运行什么软件。你会 SLAM，或者视觉导航你就能在这个平台上验证你的软件。所以凡是你觉得他能做的功能，他都是可以实现的，只是问题在于软件上，也就是使用这个平台的你能不能让他实现而和平台无关。