

桌面仿生机械手

(G2-stm32)

使用手册



2023 年 02 月

声明

本着为用户提供更好服务的原则，杭州众灵科技有限公司（以下简称“众灵”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，众灵不能完全保证该文档在任何时段的时效性与适用性。众灵有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，您可以定时访问众灵官方网站或者与众灵工作人员联系。感谢您的包容与支持！

警告

电路板相关部分，避免接触金属物体或人体，否则容易造成短路或者静电击穿。

目录

声明	2
警告	2
第一章 快速使用	4
第一课 产品介绍	5
第二课 蓝牙操控	6
第三课 手柄操控	9
第四课 手套操控	10
第二章 基础教学	12
第一课 认识 KPZ32	13
第二课 安装串口驱动	18
第三课 keil 软件的安装和使用	20
第四课 32 程序的文件结构	22
第五课 下载程序的方法	24
第六课 出厂程序讲解	26
常见问题解决办法:	28
附录:	29
售后服务	30

第一章 快速使用

注意：成品到手可玩

第一课 产品介绍

【本课目标】：

本课认识桌面仿生机械手 G2

【本课内容】：



上图为本店的桌面仿生机械手 Gihand2。

硬件部分：

手腕处安装的是本店的 PM10S 舵机（即扭矩为 $10\text{N}\cdot\text{m}$ 的单轴 PWM 舵机），可控制机械手左右转动。

五根手指上分别安装有一个 9g PWM 小舵机，每根手指都可以进行张开闭合操作。

控制板使用的是本店的 KPZ32，也就是 32 核心板+底板。

供电使用适配器进行供电，控制板上的电源接线柱处固定好适配器对接线，适配器对接线再连接适配器即可。

套餐：

套餐分为手柄遥控版和手套控制版。

手柄遥控版：可使用手柄控制手腕左右转动及每根手指张开闭合。

手套控制版：可使用手套控制手腕左右转动及每根手指张开闭合。

【本课小结】：

本课认识了桌面仿生机械手 G2。

第二课 蓝牙操控

【课程目标】：

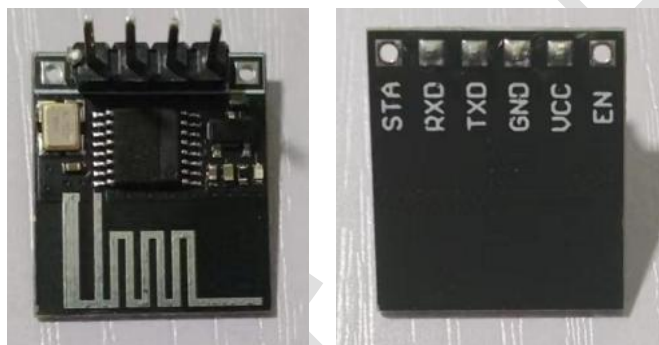
本课学习使用微信小程序控制仿生手掌。

【课程内容】：

微信小程序控制是基于蓝牙模块来实现的。

第一步：蓝牙模块安装 (若为成品则已安装好)

蓝牙模块如下所示，两张图分别为正反面：



蓝牙模块插到控制板底板上的方法如下所示：

注意排针的位置，

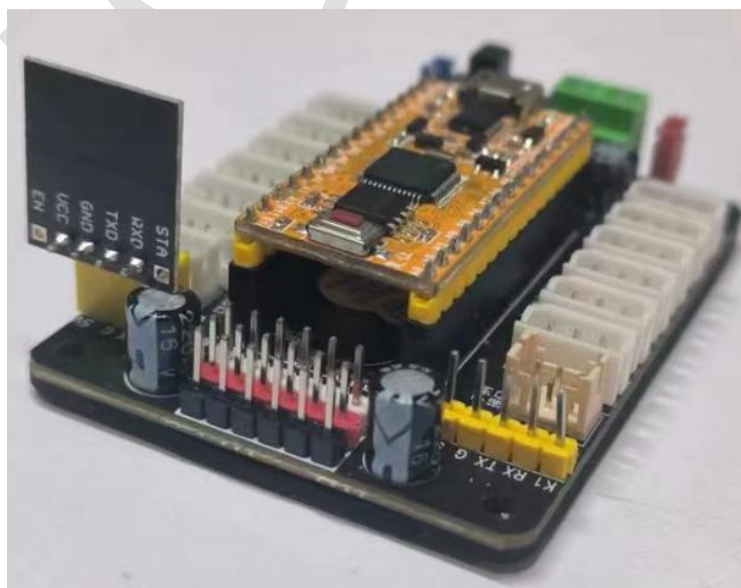
针脚对应关系(控制板底板上的 K1 针脚没有用，注意不要插错了)：

蓝牙模块 RXD ----> 控制板底板的 TX

蓝牙模块 TXD ----> 控制板底板的 RX

蓝牙模块 GND ----> 控制板底板的 G

蓝牙模块 VCC ----> 控制板底板的 5V



第二步：操控说明：

注意不要在手机设置页面连接蓝牙

打开微信小程序搜索【众灵机器人】，打开小程序以后，找到【仿生机械手掌】，在右上角的蓝牙图标中找到蓝牙，然后点击连接，连接成功以后会显示【连接成功】。



然后具体操作可看下图：



注意：此处我们使用的是仿生机械手掌（Gihand）界面，在此界面中我们仅用到了上半部分的滑杆和按钮（上图中用红框框起来的部分）。

可以通过滑杆或按钮控制每个手指张开闭合，还可以控制手腕左转右转。

【本课小结】：

注意：

1. 控制板中需要下载我们的出厂程序；
2. 使用微信小程序前要确保蓝牙模块已安装到底板上对应位置；

注意不要在手机设置中连接蓝牙，要在微信小程序中连接蓝牙。

第三课 手柄操控

【本课目标】：

本课学习使用手柄来操控桌面仿生手 G2（购买的套餐中如果不包括手柄，可跳过本节课）。

【本课内容】：

第一步：将手柄电池装到手柄中

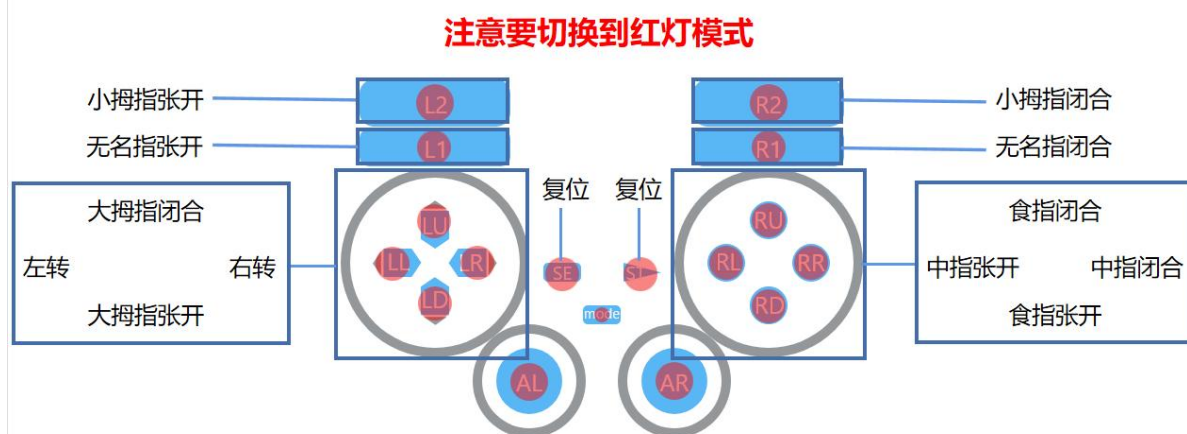
第二步：将机械手开机（打开底板电源开关）

第三步：打开手柄开关（将开关拨到 ON 那一侧即为打开手柄开关）

第四步：按 mode 键切换到红灯模式（红灯和绿灯都亮即为红灯模式）

第五步：使用手柄遥控机械手

手柄控制机械手的按键及对应功能如下：



【本课小结】：

注意：

1. 确定 32 核心板中下载的是我们的出厂程序；
2. 使用手柄前要确保手柄电池已安装到手柄中；
3. 注意要切换到红灯模式：按 mode 键切换到红灯模式（红灯和绿灯都亮即为红灯模式）；
4. 手柄开启后长时间未进行操作，手柄会自动休眠（手柄开关是打开的并且手柄上的红绿灯都是熄灭状态即为休眠，当然也有可能是手柄电池没电了），按 start 键可激活手柄。

第四课 手套操控

【本课目标】：

本课学习使用体感手套来操控桌面仿生手 G2 (购买的套餐中如果不包括体感手套, 可跳过本节课)。

【课程内容】：

第一步：无线同步模块安装 (若为成品则已安装好)

无线同步模块如下图所示：



无线同步模块插到控制板底板上的方法如下所示：

注意排针的位置，

针脚对应关系(控制板底板上的 K1 针脚没有用，注意不要插错了)：

无线同步模块 TXD ----> 控制板底板的 RX

无线同步模块 RXD ----> 控制板底板的 TX

无线同步模块 GND ----> 控制板底板的 G

无线同步模块 VCC ----> 控制板底板的 5V



第二步：操作说明：

使用手套控制之前需要先进行校准（只需校准一次即可，成品是已校准好的，可直接控制机械手），校准方式如下：

校准		
1、手握拳，保持水平，按下校准按键，听到“嘀”一声后		
2、手张开，手指竖直，保持水平，再次按下校准按键，听到长鸣一声，校准完成		

校准完成后，就可以使用手套进行控制了。

【本课小结】：

本课学习了使用仿生手套控制桌面仿生机械手 G2。

第二章 基础教学

第一课 认识 KPZ32

【本课目标】：

本课了解 STM32 核心板及底板上的资源

【课前准备】：

STM32 核心板、底板

【本课内容】：

一、STM32 核心板简介：

1.认识单片机

单片机全称是单片微型计算机，说计算机大家都知道它内部主要包含微处理器 CPU、硬盘、内存条等部件，一个单片机内部也包含了微处理器内核、程序存储器、数据存储器等，单片机的内核相当于计算机主板上的 CPU，单片机的程序存储器相当于计算机的硬盘，单片机的数据存储器相当于计算机的内存条，另外，编写过计算机运用程序的人都知道，计算机是按程序命令一条条执行语句完成所需的功能，单片机也是按程序命令一条条执行语句完成所需的功能，从这里可以看出，单片机与计算机实在是太相似了，这就是可以把它称为计算机的原因，另外，单片机拥有的这么多的结构部件都是集成在单一的一块集成电路芯片上的，加上体积微小，所以全称就是单片微型计算机，简称单片机。

单片机与普通集成电路的区别是：普通集成电路功能是固定死的，使用者无法更改，单片机的功能是可以编写程序进行更改的。事实上，由于单片机只是用在电子产品线路板上的一个集成电路芯片，完成一些常用的电气检测与控制功能，把它称为微型计算机太过夸大其词，于是又有人把它改名称为微控制器，英文名称：Micro Control Unit，缩写为 MCU，不管称为单片微型计算机还是微控制器或者 MCU，它本质上始终是用在电子产品线路板上的一个集成电路芯片，没什么神奇之处。

2.单片机的用途

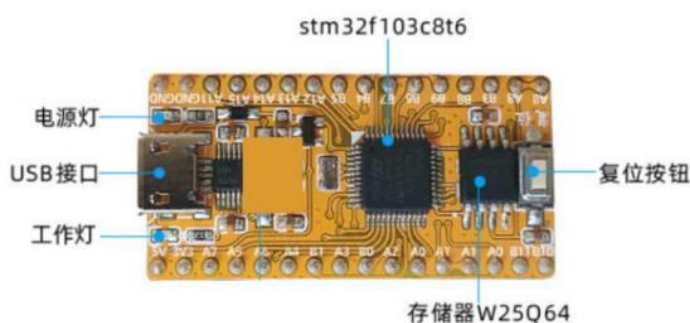
单片机用途十分广泛，比如常见的家用电器洗衣机、空调、电磁炉等内部有单片机，现在的智能化仪器仪表内都有单片机，工业生产上的数控机床有位移检测用的光栅尺，光栅尺连接的控制仪表内就有单片机。

现在这个时代的电子产品，普遍都在使用单片机，所以学好单片机是非常重要的。

3.本课程所使用的 32 芯片

STM32 是一个微控制器产品系列的总称，目前这个系列中已经包含了多个子系列，分别是：STM32 小容量产品、STM32 中容量产品、STM32 大容量产品和 STM32 互联型产品；按照功能上的划分，又可分为 STM32F101xx、STM32F102xx 和 STM32F103xx 系列等，本学习套件使用的是 stm32f103c8t6。

本课程使用的 STM32 开发板的详细组成信息如下图所示。



接口介绍:

【USB 接口】：主要用于下载程序和上位机通信使用。

【工作灯】：工作指示灯。

【电源灯】：控制板供电时，电源灯亮起，不亮时异常。

【复位按钮】：按下时复位重启。

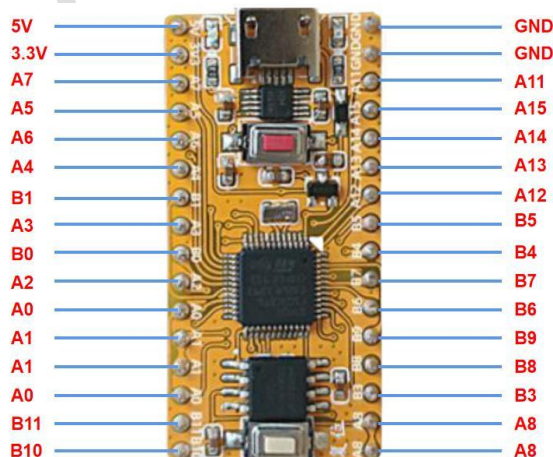
【主控芯片】：stm32f103c8t6。

【存储芯片】：W25Q64 主要用于存储舵机动作组。

注意：有的板子可能还有下载按钮，是靠近 usb 接口处的一个红色小按钮，主要是下载程序时使用。如果板子上没有下载按钮，则无需按此按钮即可直接下载程序。

【下载按钮】：下载程序时使用。

输入/输出端口 (I/O)

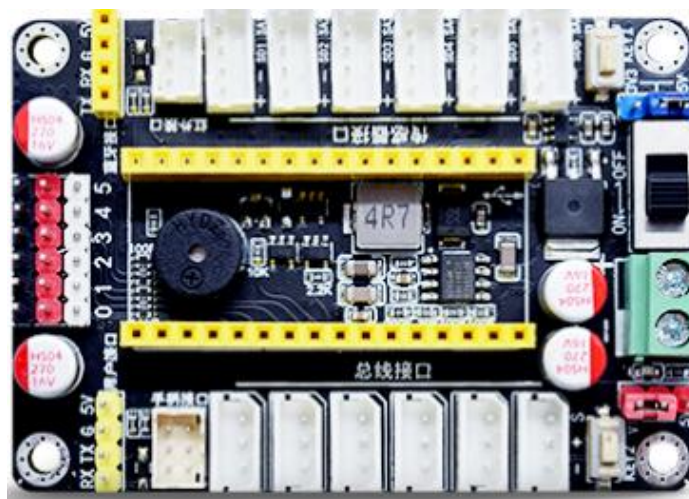


如上图所示为 STM32 核心板的引脚分布。

此处只需了解即可，在实际使用时用的是映射到底板上的各接口对应的 stm32 的引脚。

二、底板简介：

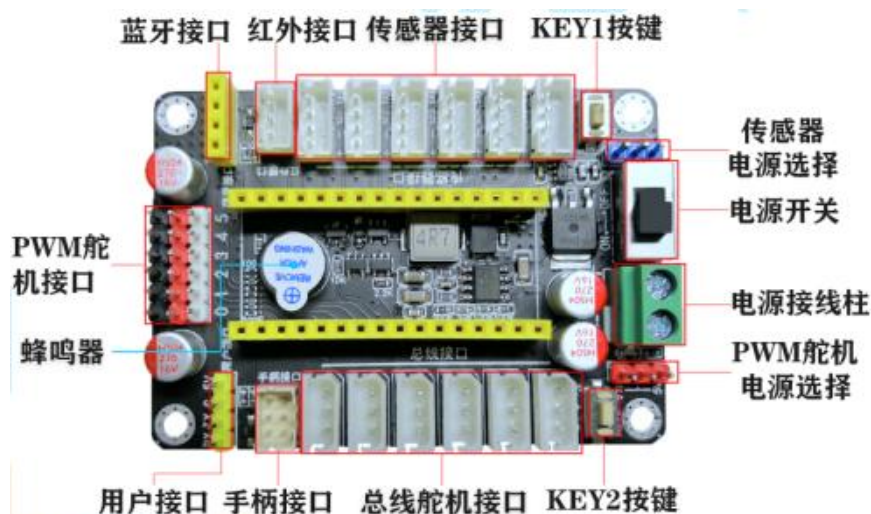
我们的底板如下图所示：



STM32 核心板与底板的插接方法如下图所示，注意不要插反：



底板接口参数如下所示：



底板接口介绍如下：

【**电源接线柱**】：供电电压 5~8.4V，电源功率可根据实际插接设备数量自行搭配，一般电源额定电流输出能力要在 3A 以上，由实际情况而定。

【**电源开关**】：通断电源的作用。

【**KEY1/2**】：预留 2 路按键，具体功能根据具体程序定义。

【**传感器接口**】：6 路传感器接口，接口电压可选择；和 5V 短接时，传感器接口电压为 5V，和 3V3 短接时，传感器接口电压为 3.3V。

【**红外接口**】：可接入红外接收头，搭配红外遥控使用，可控制总线执行设备，或者作为其他触发源使用。

【**蓝牙接口**】：可插入蓝牙模块，作为输入指令信号。

【**PWM 舵机接口**】：PWM 舵机接口，可控制 6 路 PWM 舵机，该接口的电源取决于 PWM 舵机电源选择处跳线帽的接法；和 5V 短接时，板子舵机接口使用 5V3A 的板载电源，当和 VIN 短接时，舵机接口电源和输入电源保持一致。

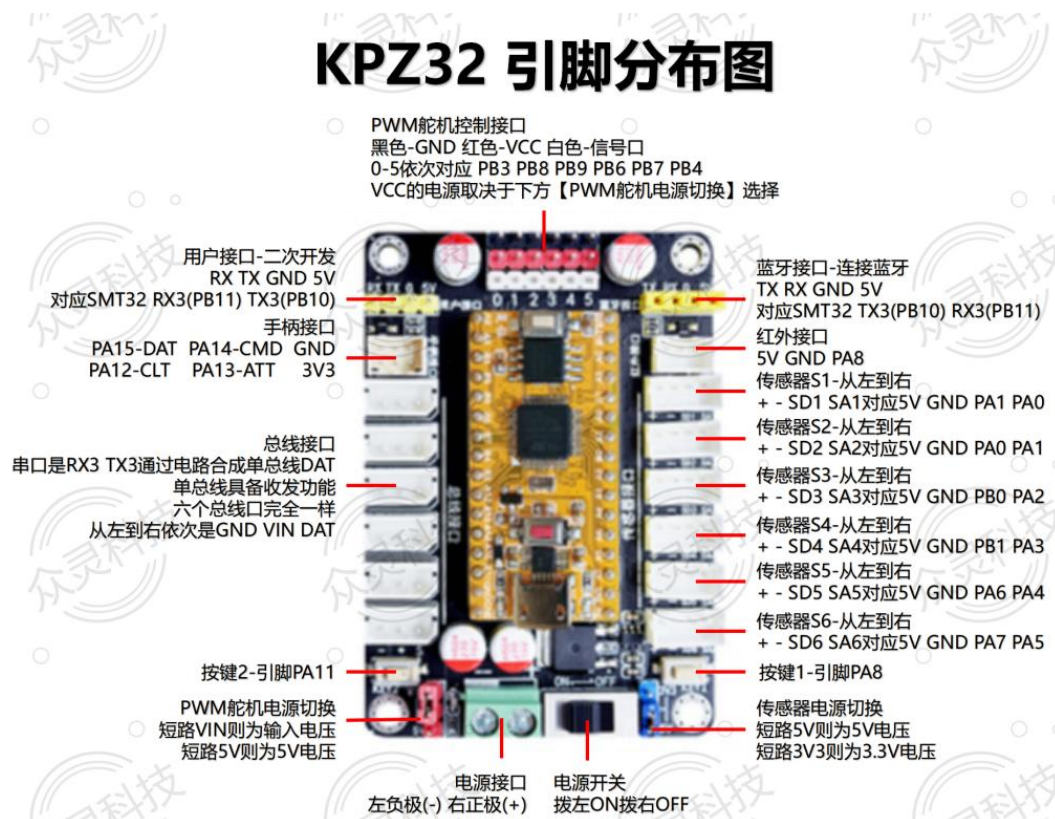
【**蜂鸣器**】：声音报警/提示作用。

【**用户接口**】：用于接无线同步模块或者二次开发时作为通信串口使用。

【**手柄接口**】：可搭配 6P 线接入 PS2 手柄接收器，搭配手柄遥控使用，可控制总线执行设备，或者作为其他触发源使用，注意：我们的 STM32 开发板不是用的此手柄接口，我们的手柄控制主要是通过我们的多功能模块来实现的。

【**总线舵机接口**】：主要接执行设备，如总线舵机，总线 MP3、总线马达等总线设备，理论接口上可串联 255 个设备，由于线的承载能力，建议每个端口串联不超过 5 个设备为宜；也可接总线通信设备，比如接总线双路驱动模块、总线多功能模块、总线语音识别模块、总线 wifi 模块等。

底板引脚分布图如下所示：



【本课小结】：

本课学习了 STM32 及底板上的资源，并对底板上的接口有了一定的了解，为之后我们的操作奠定了基础。

第二课 安装串口驱动

【课程目标】：

本课学习如何在电脑上安装 CH340 串口驱动程序，安装此驱动程序，电脑才能和控制板正常通讯。

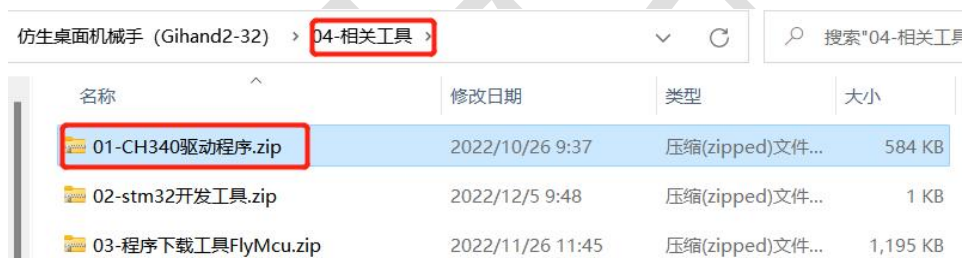
注意：本课与第一章第二课【安装串口驱动】完全相同，如果你已经安装过了串口驱动程序，可以跳过本课。

小提示：

1. 如果你的电脑之前安装过 CH340 串口驱动程序，那么再次安装 CH340 串口驱动程序时，它会显示安装失败。
2. 如果安装串口驱动程序显示预安装成功，可以换一根数据线连接开发板然后进行安装，直到显示安装成功。

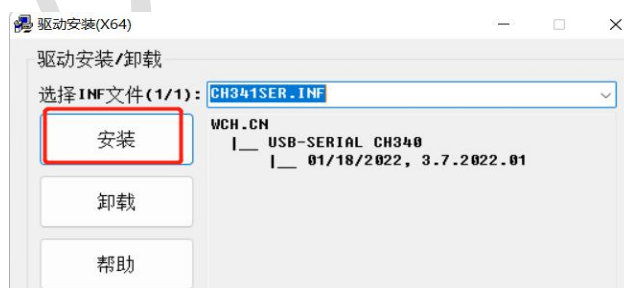
【课程内容】：

一、文件位置



二、安装方式

用数据线连接 STM32 核心板和电脑，将 CH340 驱动程序解压之后，双击打开，然后点击‘安装’，等待安装成功即可。（注意：如果之前安装过 CH340 串口驱动程序，那么安装时会显示安装失败）



三、验证是否安装成功

安装完成后，32 核心板用数据线连接电脑，打开电脑的【设备管理器】中的【端口】，可以看到如下图所示的 CH340 的端口，就表示串口驱动安装成功。



【课程小结】：

本课学习了如何在电脑上安装 CH340 串口驱动程序，安装此驱动程序，电脑才能和控制板正常通讯。

小提示：

1. 如果你的电脑之前安装过 CH340 串口驱动程序，那么再次安装 CH340 串口驱动程序时，它会显示安装失败。
2. 如果安装串口驱动程序显示预安装成功，可以换一根数据线连接开发板然后进行安装，直到显示安装成功。

第三课 keil 软件的安装和使用

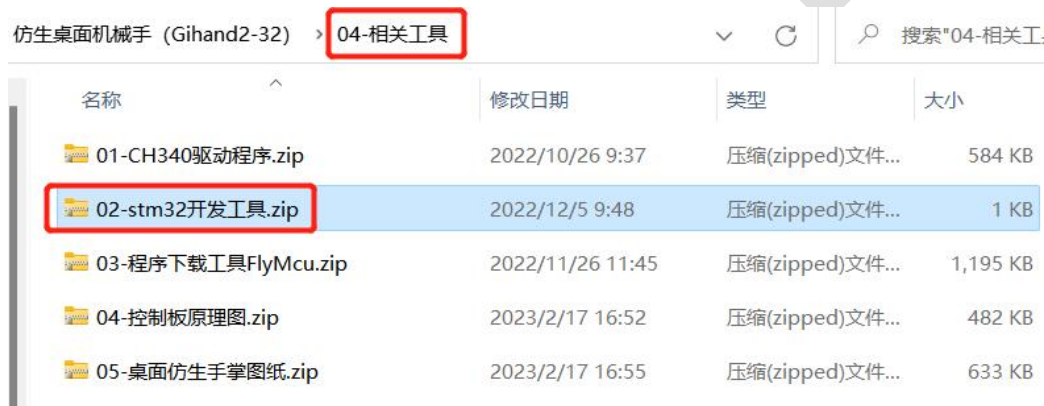
【本课目标】：

本课了解 keil 软件的安装和使用

【本课内容】：

一、keil 软件的安装

在资料中的‘相关工具’-->‘stm32 开发工具’中可以找到 keil 软件下载链接下载即可。



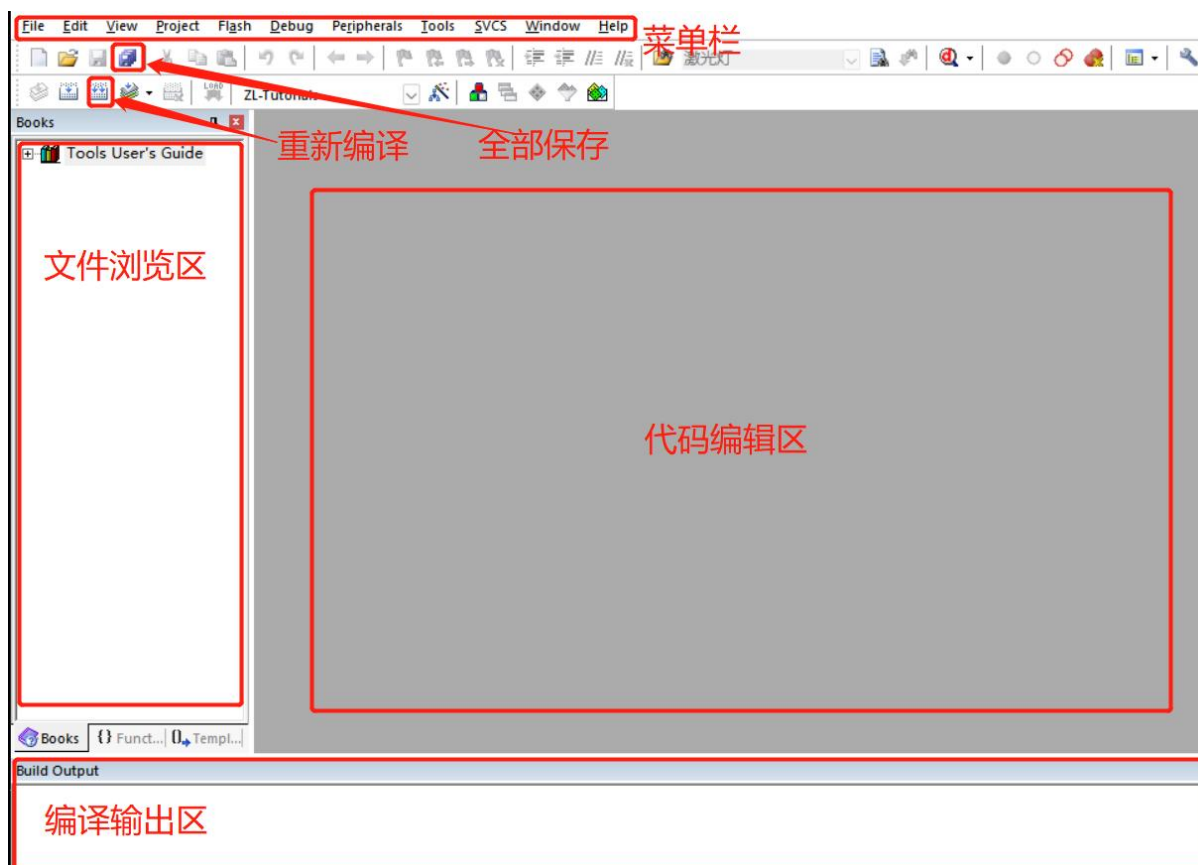
可在网上百度一下 keil 软件安装教程，然后根据步骤安装即可。

注意：安装完成后，需要将我们提供的 pack 包也双击安装。



二、keil 软件的使用

Keil 软件安装好之后，可以双击打开 keil 软件，然后会看到下图所示界面（图中只标出了比较常用的几个部分）



菜单栏：文件、编辑等操作。

文件浏览区：当打开一个项目后，此处可以看到该项目包含的所有文件，双击文件名可在代码编辑区中查看文件中的内容。

代码编辑区：编写代码的区域。

编译输出区：程序编译之后，会在编译输出区打印编译的信息及结果。

重新编译：当代码编写好之后，点击重新编译可以将程序重新编译成一个 hex 文件，使用烧录工具烧录该 hex 文件到开发板中即可。

全部保存：当代码编写好之后，点击全部保存，可以将该项目中的所有文件都保存好，不用再一个一个去保存了。

关于 keil 软件的更详细的使用还可在网上自行百度，了解更多。

【本课小结】：

本课了解了 keil 软件的安装和使用

第四课 32 程序的文件结构

【课程目标】：

本课了解本课程中 32 程序的文件结构

【本课内容】：

一个 32 程序包含多个文件夹，在每个文件夹中又包含了多个依赖文件，此处以我们的出厂程序源码为例，其中的文件结构如下所示：

名称	修改日期	类型	大小
CMSIS	2023/2/17 13:38	文件夹	
Libraries	2023/2/17 13:38	文件夹	
LIST	2023/2/17 13:38	文件夹	
OUT	2023/2/17 13:38	文件夹	
Project	2023/2/20 18:07	文件夹	
src	2023/2/17 13:38	文件夹	
Startup	2023/2/17 13:38	文件夹	
USER	2023/2/17 13:38	文件夹	
ADC.gif	2015/7/17 10:34	GIF 图片文件	31 KB
FlyMcu.exe	2015/9/9 15:48	应用程序	2,564 KB

其中：

OUT：编译好之后的文件都存放在 OUT 文件夹中，编译好之后的文件后缀为 hex

Project：项目主文件所存放的位置

src：存放资源文件，如延时、led 灯的点亮熄灭等代码都放在该文件夹中

USER：存放主程序，如初始化、循环等代码都放在该文件夹中

FlyMcu.exe：烧录工具，烧录程序时需要使用的工具

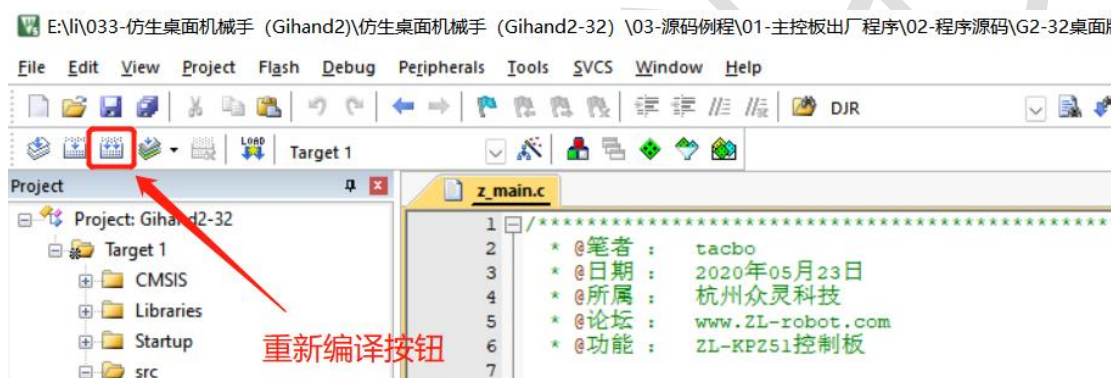
OUT 文件夹中编译好的 hex 文件如下所示：

名称	修改日期	类型	大小
Kibot1-32.axf	2021/6/9 16:12	AXF 文件	1,392 KB
Kibot1-32.hex	2021/6/9 16:12	HEX 文件	129 KB

project 文件夹中的项目主文件如下所示，我们打开此项目时只需双击如下所示项目主文件即可：

02-程序源码 > G2-32桌面版出厂例程-20210202 > Project					搜索"Project"
名称	修改日期	类型	大小		
Gihand2-32.uvgui.Administrator	2023/2/20 18:07	ADMINISTRATO...	90 KB		
Gihand2-32.uvgui.TB	2021/6/9 16:12	TB 文件	71 KB		
Gihand2-32.uvopt	2023/2/20 18:07	UVOPT 文件	16 KB		
Gihand2-32.uvproj	2023/2/20 18:07	磳ision4 Project	22 KB		

当您想对某个项目进行二次编辑时，可以找到该项目 project 文件夹下的后缀名为 uvproj 的文件双击打开即可（注意：前提是电脑上已经安装好了 keil 软件），改好之后点击‘重新编译’按钮即可生成 hex 文件。



【本课小结】：

本课了解了本课程中 32 程序的文件结构。

第五课 下载程序的方法

【课程目标】：

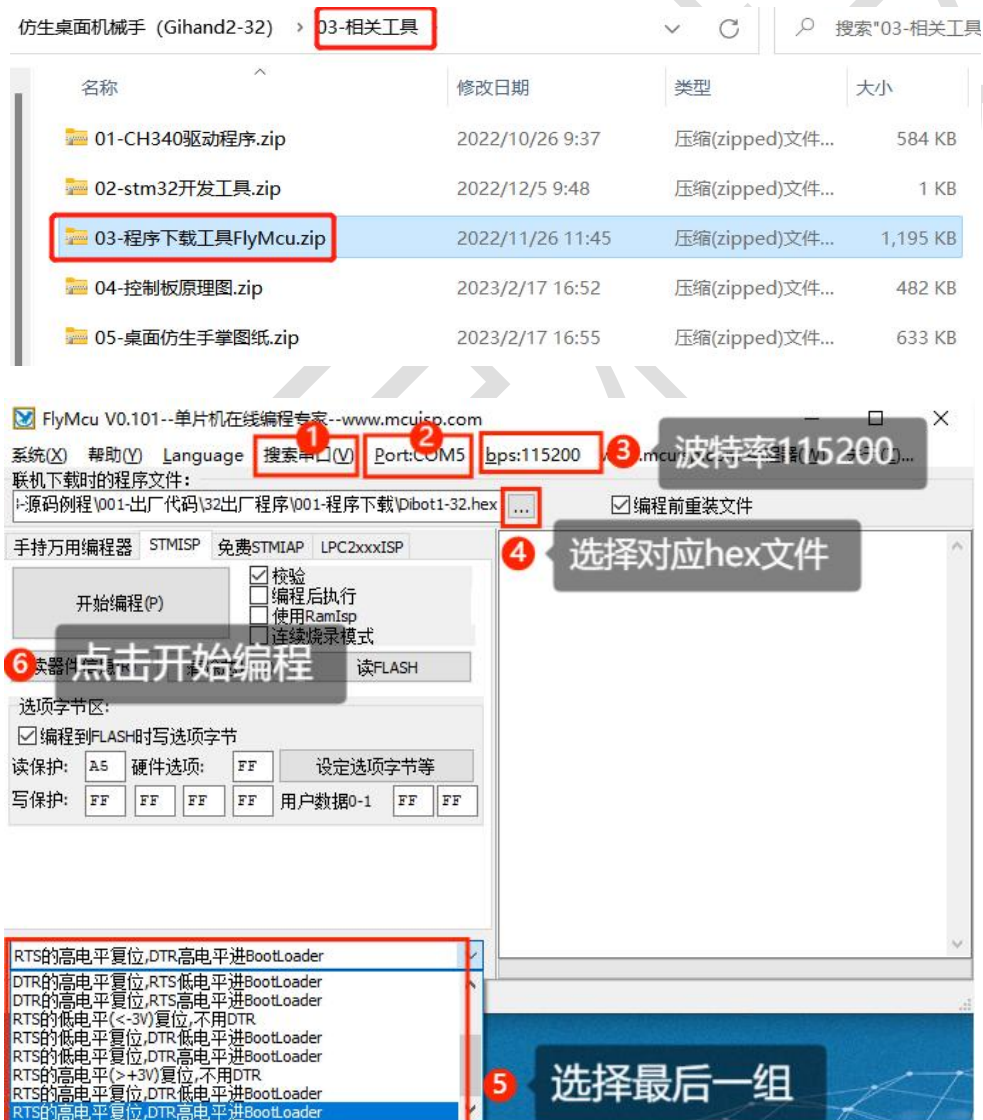
本课学习烧录 STM32 程序的方法

【本课内容】：

第一步、确保电脑已安装好串口驱动程序。

第二步、程序烧录方法

使用 FlyMcu 工具将程序下载到控制板中。



注意：

如果点击‘开始编程’之后未能成功下载程序，查看串口是否选对。

如果串口是对的，则看一下开发板上靠近 usb 接口处是否有一个红色的下载按钮，如果有下载按钮，则按住下载按钮，然后点击‘开始编程’即可烧录程序。

【本课小结】：

本课学习了 32 程序的下载方法。



第六课 出厂程序讲解

【课程目标】：

本课对出厂程序进行简单的讲解

【本课内容】：

出厂程序效果

开机后底板板载蜂鸣器鸣叫 3 声，表示开机成功，程序开始运行。

板载 led 灯每隔 1 秒亮 1 秒，代表程序正常执行中；

功能：

可控制手腕左右转动；

可控制每根手指张开闭合。

控制方式：

手柄控制、手套控制。

代码截图（代码较多，此为部分截图）

```
142 int main(void) {
143     setup_rcc(); //初始化时钟
144     setup_global(); //初始化全局变量
145     setup_gpio(); //初始化IO口
146     setup_nled(); //初始化工作指示灯
147     setup_beep(); //初始化定时器
148     setup_djio(); //初始化舵机IO口
149     setup_w25q64(); //初始化存储器W25Q64
150     setup_ps2(); //初始化ps2手柄
151     setup_systick(); //初始化滴答时钟，1s增加一次millis()的值
152     setup_uart1(); //初始化串口1 用于下载动作组
153     setup_uart3(); //初始化串口3 用于底板总线、蓝牙、loram
154     setup_others(); //初始化其他
155     setup_dj_timer(); //初始化定时器2 处理舵机PWM输出
156     setup_start(); //初始化启动信号
157     setup_interrupt(); //初始化总中断

158     setup_sensor(); //初始化传感器
159
160     //kinematics 90mm 105mm 98mm 150mm
161     setup_kinematics(90, 105, 98, 150, &kinematics);
162     IWDG_Init();
163
164
165     while(1) {
166         loop_nled(); //循环执行工作指示灯，500ms跳动一次 和声音公用一个IO口 这里在声音功能启用的时候就
167         loop_uart(); //串口数据接收处理
168         loop_action(); //动作组批量执行
169         loop_ps2_data(); //循环读取ps2手柄数据
170         loop_ps2_button(); //处理手柄上的按钮
171         loop_ps2_car(); //处理小车电机摇杆控制
172         loop_monitor(); //定时保存一些变量
173         //loop_sensor(); //循环处理传感器
174     }
175
176 }
```

初始化部分：
主要是手柄初始化和
串口初始化

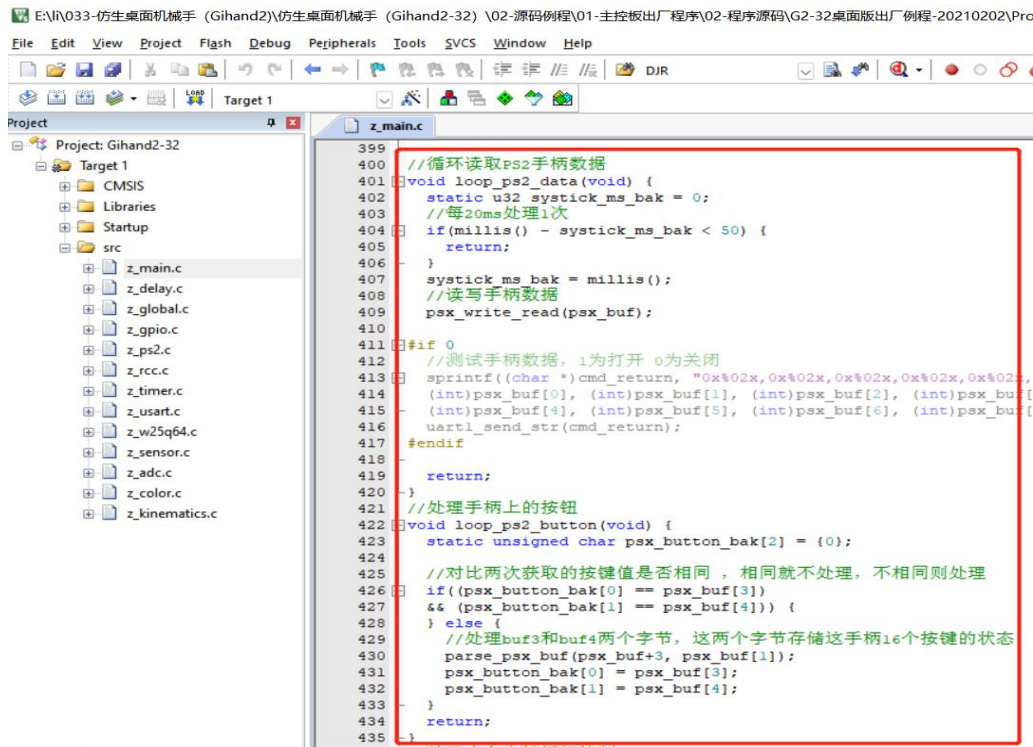
循环执行部分：
主要是循环处理手柄数据
和串口指令解析执行

代码分析

出厂程序代码主要分为两大部分：初始化部分、手柄控制部分、串口指令控制部分。

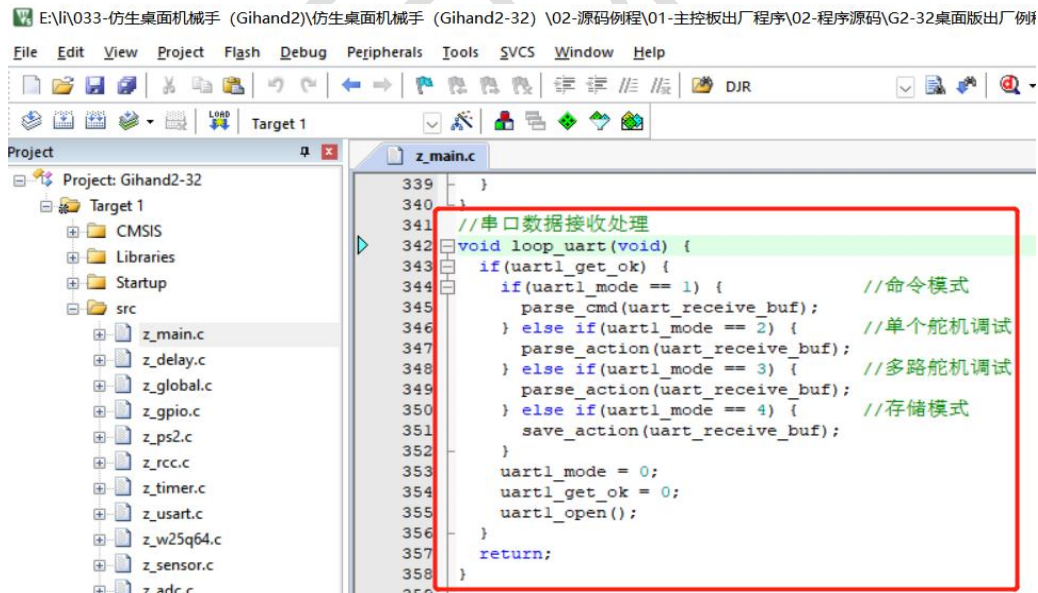
初始化部分：主要是对 led 灯、有源蜂鸣器、手柄、串口进行初始化。

手柄控制部分：loop_ps2_data 函数处理手柄数据。



```
399 //循环读取ps2手柄数据
400 void loop_ps2_data(void) {
401     static u32 systick_ms_bak = 0;
402     //每20ms处理1次
403     if(millis() - systick_ms_bak < 50) {
404         return;
405     }
406     systick_ms_bak = millis();
407     //读写手柄数据
408     psx_write_read(psx_buf);
409
410     #if 0
411     //测试手柄数据, 1为打开 0为关闭
412     sprintf((char *)cmd_return, "0x%02x,0x%02x,0x%02x,0x%02x,0x%02x,0x%02x",
413         (int)psx_buf[0], (int)psx_buf[1], (int)psx_buf[2], (int)psx_buf[3],
414         (int)psx_buf[4], (int)psx_buf[5], (int)psx_buf[6], (int)psx_buf[7]);
415     uart1_send_str(cmd_return);
416     #endif
417     return;
418 }
419 //处理手柄上的按钮
420 void loop_ps2_button(void) {
421     static unsigned char psx_button_bak[2] = {0};
422
423     //对比两次获取的按键值是否相同，相同就不处理，不相同则处理
424     if((psx_button_bak[0] == psx_buf[3])
425     && (psx_button_bak[1] == psx_buf[4])) {
426     } else {
427         //处理buf3和buf4两个字节，这两个字节存储这手柄16个按键的状态
428         parse_psx_buf(psx_buf+3, psx_buf[1]);
429         psx_button_bak[0] = psx_buf[3];
430         psx_button_bak[1] = psx_buf[4];
431     }
432     return;
433 }
```

串口指令控制部分：通过对串口接收数据进行判断来执行不同的功能。



```
339 }
340
341 //串口数据接收处理
342 void loop_uart(void) {
343     if(uart1_get_ok) {
344         if(uart1_mode == 1) { //命令模式
345             parse_cmd(uart_receive_buf);
346         } else if(uart1_mode == 2) { //单个舵机调试
347             parse_action(uart_receive_buf);
348         } else if(uart1_mode == 3) { //多路舵机调试
349             parse_action(uart_receive_buf);
350         } else if(uart1_mode == 4) { //存储模式
351             save_action(uart_receive_buf);
352         }
353         uart1_mode = 0;
354         uart1_get_ok = 0;
355         uart1_open();
356     }
357     return;
358 }
```

手套控制就是通过串口指令来控制的。

【本课小结】：

本课对出厂程序进行了简单的讲解。

常见问题解决办法：

1. 手柄控制不了了？

可能是电池快没电了，或手柄没有切换到红灯模式。



附录：

手柄收发资料链接：

<https://pan.baidu.com/s/16UdDo207x2hETfa8q8uq9Q>

提取码：8piu

体感手套资料链接：

<https://pan.baidu.com/s/10tTk88rl2Pkj6nRykiw5Mg>

提取码：6njz

售后服务

【产品核对】：收到产品时请及时对照发货清单检查配件是否齐全，以及快递运输过程中产品是否损坏等现象，如有问题请及时联系淘宝客服人员（注：自收到货起三日内没有反应问题，视为产品收货正常）。

【资料索取】：学习资料以百度云网盘链接形式发送，收到物品时请联系客服索取。

【组装接线】：散件组装接线时请按照教程进行组装接线，因操作不正确导致产品损坏的，一切后果由自己承担！如有问题请及时联系客服或售后技术人员。

【品质服务】：提供全程有关产品技术支持（可通过电话/QQ/微信等）。

【售后时间】：10:00---20:00（周一到周六）

【技术支持】：联系淘宝客服

【技术论坛】：<http://www.zl-robot.com>

关注微信公众号
获取更多产品信息

