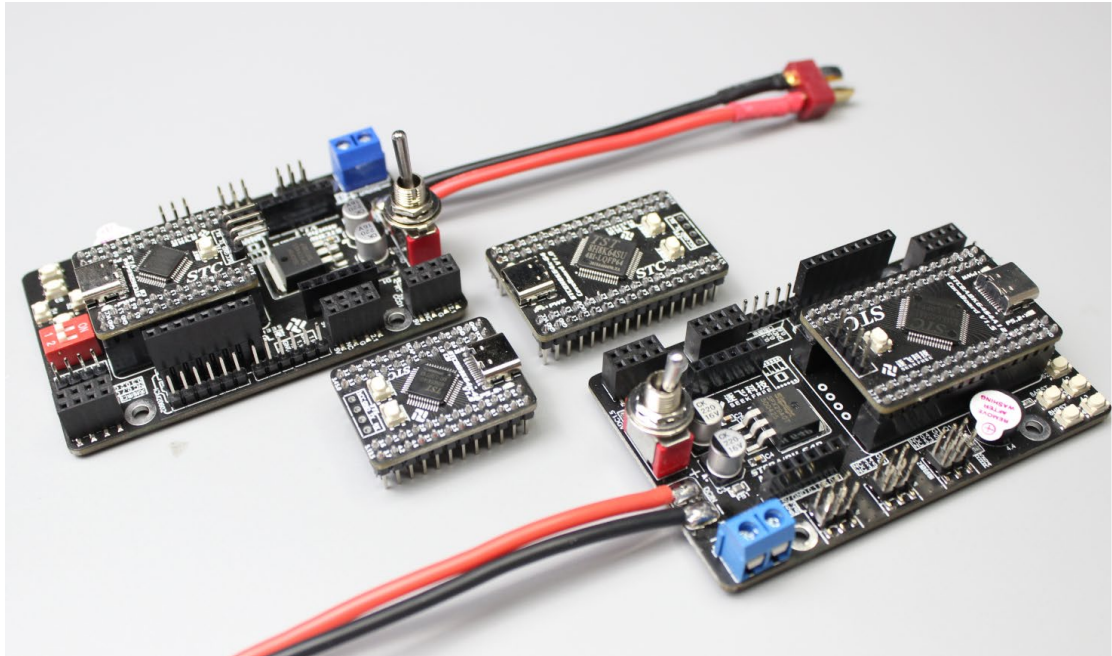


逐飞 STC8H/STC8G/STC8A 智能车开源库 汇总上线

一、前情提要：

STC（宏晶）为第十五届全国大学生智能汽车竞赛的新引入的赞助商之一，在已经公布的规则中，指定了节能组只能使用 STC 的三款单片机作为主控芯片，其中包括最开始公布的两款（STC8G2K64S4、STC8H8K64U）可以免费申请的样片。后来根据广大师生建议，为了便于参赛队伍自行购买方便，对于现在库存量比较大的单片机的型号也允许用于智能车竞赛，扩充的 STC 单片机型号为 STC8A8K64S4。

在这三款单片机中，8G2K,8H8K 都是新品，并且都可以申请免费样片，目前 8G2K 已经量产，并已经根据申请情况免费发放给两百多所高校；8H8K 的 TEST 版本也已经开放申请，带 USB 下载功能，不带 USB 仿真；8A8K 是商用成熟产品，可以直接购买使用。逐飞针对这三款单片机分别开发了针对智能车应用的开源库和 4 种核心板，48 脚 STC8G2K 核心板，64 脚 STC8A8K 核心板，48 脚 STC8H8K 核心板、64 脚 STC8H8K 核心板，同时还带来了两款用于学习的拓展学习板，一块主板可以兼容两种 48 脚芯片的核心板，另一块主板可以兼容两种 64 脚芯片的核心板，由它们组成了第十五届 STC 智能车全系产品，合影如下：



二、底层驱动库需求：

按理说，大家对 STC 应该很熟悉了，至少比对英飞凌和恩智浦熟悉很多了，而且 STC 单

片机本身也简单易用，寄存器也没多少，不像 32 位单片机那么复杂，就算自己从头写也不会太难，但很奇怪的是自从新规则发布以来，依然有很多想参加智能车竞赛的同学们希望有个底层库供他们开发使用，我想这里面有一部分是的确还没学习过单片机，还有一部分应该就是懒吧，对于还没有单片机基础的，可以参考逐飞底层驱动库的写法自己学习单片机的操作和使用，但是对于那部分懒癌同学，我也无话可说了啊，毕竟是开源的，欢迎使用，欢迎大佬提出使用建议，相互交流，共同进步。

三、逐飞 STC 智能车开源库：

本文汇总介绍逐飞发布基于上文中提到的三款单片机的底层库，分别是“**逐飞 STC8H8K 开源库**” & “**逐飞 STC8G2K 开源库**” & “**逐飞 STC8A8K 开源库**”。下面我们就通过本文来详细了解下吧。

版权声明：该开源库版权归成都逐飞智能设备有限公司所有，欢迎各位使用并传播该库程序，修改内容时务必保留逐飞科技的版权声明。未经允许不得用于商业用途或变相获利。

逐飞 STC8H8K 开源库链接：https://gitee.com/seekfree/STC8H8K64_Library

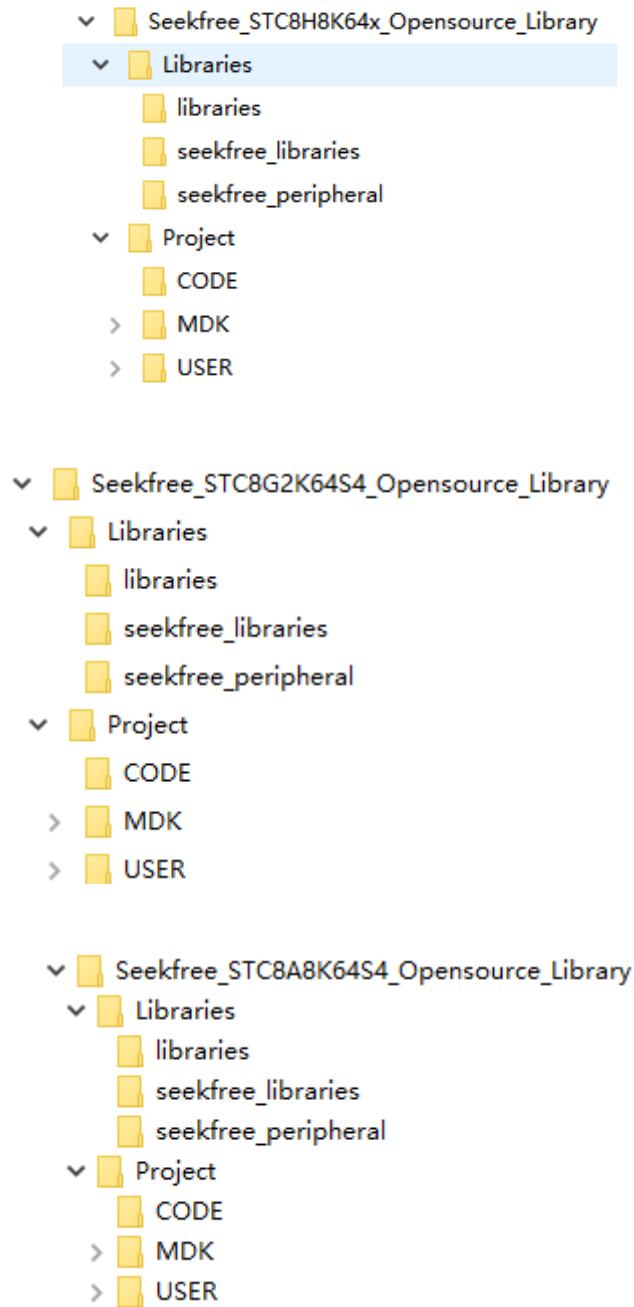
逐飞 STC8G2K 开源库链接：https://gitee.com/seekfree/STC8G2K64S4_Library

逐飞 STC8A8K 开源库链接：https://gitee.com/seekfree/STC8A8K64S4_Library

STC8H/STC8G/STC8A 底层库继续沿用逐飞科技 RT10xx 系列库的风格，使用过逐飞库的同学应该知道，简洁易懂，容易上手。

本次 STC8H/STC8G/STC8A 开源库使用 MDK 进行编写的。使用单片机寄存器作为最底层。然后将各个模块使用的寄存器进行封装提升易用性。当大家觉得库的功能不满足自己使用需求时，可以采用同样的方式自己调用寄存器来实现更多其他的功能。大家可以通过我们的库来学习官方的寄存器怎么使用，当自己学会怎么使用官方的寄存器之后也可以直接使用寄存器，这样的好处就是没有多余的代码，降低自己代码的体积，提高效率。我们并不希望用户只会用我们的库，我们更多的是希望初学者不被高门槛给拦在外面，逐飞库只是提供一个入门学习的渠道而已。

3.1、目录结构图



Libraries 文件夹下放置的是 STC 头文件，以及数据类型声明，board 等文件。

seekfree_libraries 文件夹下放置的是逐飞科技精心编写的底层驱动，底层驱动是用 STC 提供的头文件进行二次封装，以简化各个模块的使用步骤，使用更加方便。如果不是参加竞赛的用户可以使用这一层的驱动来实现 STC8H/8A/8G 的快速上手，也可以通过查看我们的代码用于寄存器如何使用。目前支持 ADC、EXTI、IIC、SPI、PIT、CTIME、UART。如果有需要其他功能亦可直接调用寄存器来实现其他功能。

`seekfree_peripheral` 文件夹下放置的是各类常用的模块驱动，当使用到这些模块时只需要调用函数即可实现相应的功能，非常的简单方便。目前实现线性 CCD、1.8 寸 TFT、ICM20602 六轴陀螺仪、IIC 通讯协议（模拟 IO）、1.14 寸 IPS 液晶屏、MPU6050 六轴陀螺仪、OLED 显示屏、虚拟示波器通讯协议、无线转串口模块。

`CODE` 文件夹下放置的是用户自己添加的代码文件，当用户自己添加文件的时候我们建议全部添加在这个文件夹内。

`USER` 文件夹下放置的是 `main.c`、`isr.h`、`isr.c` 文件。

3.2、底层驱动功能模块

目前逐飞 `STC8H/STC8G/STC8A` 库实现了芯片中 ADC、SPI、IIC、UART、GPIO、PIT、DELAY、EXTI、CTIMR 功能模块的使用，以下几个模块单独说明下：

GPIO 模块：

使用过 89C52 的同学就知道 GPIO 不需要初始化，直接就能使用。在官方提供的头文件中，所有的端口都已经被定义了。

```
sfr      P0          = 0x80;
sbit     P00        = P0^0;
sbit     P01        = P0^1;
sbit     P02        = P0^2;
sbit     P03        = P0^3;
sbit     P04        = P0^4;
sbit     P05        = P0^5;
sbit     P06        = P0^6;
sbit     P07        = P0^7;
```

我们所编写的 GPIO 模块包含了 GPIO 模式选择，上拉使能。

EXTI 模块：

外部中断模块包含了五个外部中断源触发中断，其中 INT0 和 INT1 支持边沿触发+下降沿触发，其他三个只支持下降沿触发。

PWM 模块：

STC8H8K 可以输出 8 路 16 位 PWM，STC8G2K64S4 理论上可以输出 45 路 PWM，STC8A8K64S4 可以输出 8 路 PWM，但对于节能组来说 8 路 PWM 实际上已经够用了。

CTIMR 模块：

CTIMR 模块是通过定时器外部计数，作为输入捕获。一共有五个定时器，最多实现五个 CTIMER 模块，每个模块可以实现一路编码器的脉冲采集，这样就可以采集逐飞带方向输出的编码器数据。

PCA 模块：
























通过 PCA 模块，进行周期中断。

DELAY 模块：

STC8G/STC8A 通过 PCA 模块，进行硬件延时。由于 STC8H 没有 PCA 模块，这里的延时方式是软件延时。

3.3、各外设模块的底层

逐飞 STC8H/8G/8A 库还包含各个外设的模块的，具体包含外设模块如下图所示：

 headfile.h	2020/4/18 12:34	H 文件	1 KB
 SEEKFREE_18TFT.c	2020/4/18 12:33	C 文件	17 KB
 SEEKFREE_18TFT.h	2020/4/18 12:33	H 文件	4 KB
 SEEKFREE_ABSOLUTE_ENCODER.c	2020/4/16 17:53	C 文件	16 KB
 SEEKFREE_ABSOLUTE_ENCODER.h	2020/4/16 18:24	H 文件	3 KB
 SEEKFREE_FONT.c	2020/4/7 17:02	C 文件	23 KB
 SEEKFREE_FONT.h	2020/4/9 17:36	H 文件	2 KB
 SEEKFREE_ICM20602.c	2020/4/16 18:41	C 文件	19 KB
 SEEKFREE_ICM20602.h	2020/4/16 17:49	H 文件	5 KB
 SEEKFREE_IIC.c	2020/4/16 17:53	C 文件	8 KB
 SEEKFREE_IIC.h	2020/4/13 15:39	H 文件	2 KB
 SEEKFREE_IPS114_SPI.c	2020/4/16 17:37	C 文件	19 KB
 SEEKFREE_IPS114_SPI.h	2020/4/16 17:45	H 文件	5 KB
 SEEKFREE_MPU6050.c	2020/4/16 17:28	C 文件	5 KB
 SEEKFREE_MPU6050.h	2020/4/13 15:40	H 文件	3 KB
 SEEKFREE_OLED.c	2020/4/18 12:34	C 文件	18 KB
 SEEKFREE_OLED.h	2020/4/13 16:09	H 文件	4 KB
 SEEKFREE_TSL1401.c	2020/4/15 14:11	C 文件	4 KB
 SEEKFREE_TSL1401.h	2020/4/11 14:57	H 文件	1 KB
 SEEKFREE_VIRSCO.c	2020/4/14 10:44	C 文件	4 KB
 SEEKFREE_VIRSCO.h	2020/4/14 10:45	H 文件	2 KB
 SEEKFREE_WIRELESS.c	2020/4/18 15:15	C 文件	4 KB
 SEEKFREE_WIRELESS.h	2020/4/18 15:15	H 文件	2 KB

3.4、库例程

逐飞也为各位准备了各个模块的库例程，如下图截图所示，可以让新手朋友或者没有使用过我们库的小伙伴能够快速上手，大家从截图中可以看到目录全部采用英文，因为这样可以避免出现各种奇怪的问题，所以建议写代码的小伙伴都用全英文的路径避免出问题。最下方的 excel 中有讲解例程的功能。

1-LED Blink Demo	2020/4/18 15:38	文件夹	
2-GPIO Input Demo	2020/4/18 15:38	文件夹	
3-EXTI Demo	2020/4/18 15:38	文件夹	
4-PIT Timer Interrupt Demo	2020/4/18 15:38	文件夹	
5-ADC Demo	2020/4/18 15:38	文件夹	
6-PWM Demo	2020/4/18 15:38	文件夹	
7-Decode Demo	2020/4/18 15:38	文件夹	
8-UART Demo	2020/4/18 15:38	文件夹	
9-Printf Demo	2020/4/18 15:38	文件夹	
10-External Count Demo	2020/4/18 15:38	文件夹	
Libraries	2020/4/18 15:37	文件夹	
version.xls	2020/4/14 16:41	XLS 工作表	19 KB
例程说明.xlsx	2020/4/14 16:48	XLSX 工作表	10 KB

3.5、使用示例

接下来我们通过两个外设模块的举例来演示库是如何使用的。

示例一：线性 CCD 数据采集。

根据任务需要我们查找 seekfree_peripheral 文件夹下，可以找到 SEEKFREE_TSL1401 文件，我们只需要调用相关的函数即可实现任务需求。从这个示例我们可以看到，“红孩儿”线性 CCD 又要重出江湖了，用于节能组车库识别，或者使用 STC 单片机做基础四轮组的同学也可以采用电磁导航+线性 CCD 识别车库的方案。

```
#include "headfile.h"

void main()
{
    DisableGlobalIRQ(); //关闭总中断
    board_init();      //初始化寄存器
    ccd_init();        //TSL1401线阵CCD初始化
    EnableGlobalIRQ(); //开启总中断
    while(1)
    {
        if(tsl1401_finish_flag)
        {
            //发送图像到上位机
            SendImageData();

            //采集完成标志位清除
            tsl1401_finish_flag = 0;
        }
    }
}
```

示例二：编码器数据采集。

编码器数据采集，是通过定时器作为外部输入捕获来获取编码器数据。脉冲捕获函数仅

需要填写一个参数即可获取数据。下图示例为双编码器数据采集示例。

```
#include "headfile.h"
//由于STC8A/8G没有正交解码功能，所以该例程仅支持带方向的编码器解码

//定义脉冲引脚
#define SPEEDL_PLUSE    CTIM0_P34
#define SPEEDR_PLUSE    CTIM3_P04
//定义方向引脚
#define SPEEDL_DIR      P33
#define SPEEDR_DIR      P03

int16 templ_pluse = 0;
int16 tempr_pluse = 0;
void main()
{
    DisableGlobalIRQ();           //关闭总中断
    board_init();                 //初始化寄存器
    ctimer_count_init(SPEEDL_PLUSE); //初始化定时器0作为外部计数
    ctimer_count_init(SPEEDR_PLUSE); //初始化定时器3作为外部计数
    EnableGlobalIRQ();           //开启总中断
    while(1)
    {
        //读取采集到的编码器脉冲数
        templ_pluse = ctimer_count_read(SPEEDL_PLUSE);
        tempr_pluse = ctimer_count_read(SPEEDR_PLUSE);

        //计数器清零
        ctimer_count_clean(SPEEDL_PLUSE);
        ctimer_count_clean(SPEEDR_PLUSE);

        //采集方向信息
        if(1 == SPEEDL_DIR)      templ_pluse = -templ_pluse;
        else                      templ_pluse = templ_pluse;
        if(1 == SPEEDR_DIR)      tempr_pluse = -tempr_pluse;
        else                      tempr_pluse = tempr_pluse;

        //延时100ms
        pca_delay_ms(100);
    }
}
```

3.6、注意事项：

3.6.1、串口需要占用定时器，CTIMER 外部计数也需要占用定时器。不要同时使用同一个定时器。

3.6.2、逐飞 STC8H/STC8G/STC8A 库，可以在线仿真，也可以使用 stc-isp 软件进行下载，编译完成的目标文件为 HEX 文件，对应的 HEX 文件保存在对应工程的 Seekfree_STC8xxxxxxx_Opensource_Library\Project\MDK\Out_File 文件下。

3.6.3、逐飞 STC8H/STC8G/STC8A 库，默认使用内置晶振。在开源库中 board.h 文件中 FOOSC 的值设置为 0，则程序将自动获取系统频率（即 ISP 下载时候设置的频率）。

四、结束语

本期的库介绍就到这里了，大家在使用库的过程中如果遇到什么 BUG，请及时联系我们！我们将尽快修复。PS：通过 QQ 群联系我们更方便，也方便大家讨论交流，QQ 群技术交流平台（逐飞科技_智能车③部：824575535），如果发现库有 BUG、或不足、或对库有什么建议，可以在群里提问或发表建议，也可与管理员私聊，保障该开源库有什么缺陷时可以及时得到解决。然后再分享给各位，形成良性循环。

最后，当然是希望大家多多支持逐飞科技，逐飞 STC8H/STC8G/STC8A 的核心板和两种引脚的主板都已备好现货，车库识别神器—“红孩儿线性 CCD”也已备好现货，节能组相关的硬件模块基本都备齐了，如需购买请移步至“逐飞科技”淘宝店，店内搜索 STC 即可找到相关产品，您的支持是我们开源的动力。再次感谢车友们的厚爱！

最后，分享一篇在逐飞科技微信公众号发布的一篇关于第十五届节能平衡小车的推文：
STC 节能平衡小车浅析—By “逐飞科技”

https://mp.weixin.qq.com/s?_biz=MzAxMjQxNjEyMw==&mid=2247483755&idx=1&sn=e2a3f886359d812407534e82a6f53cec&chksm=9bb3647facc4ed69f0da4d6684bfff6324c5007c377ba30b1365d557b64e38784d7aad6a242&token=1399039666&lang=zh_CN#rd

“逐飞科技”官方公众号

这是由逐飞科技官方运营的公众号，专注于分享知识，交流技术，努力构建一个嵌入式开发交流学习平台，互相促进，共同成长！感谢您关注逐飞科技，愿逐飞科技陪伴您的成长



长按二维码，关注我们

本文由“逐飞科技”编辑整理，时间仓促，水平有限，难免有遗漏和错误，大神轻拍，欢迎交流指正。愿智能车世界更美好！