# 硬件设计

电控部分是控制机器人完成所需动作的核心通过软件进行行为控制编程达到对舵机的角度控制。

电控设计包括控制系统设计、 电源设计和舵机的工作设计。

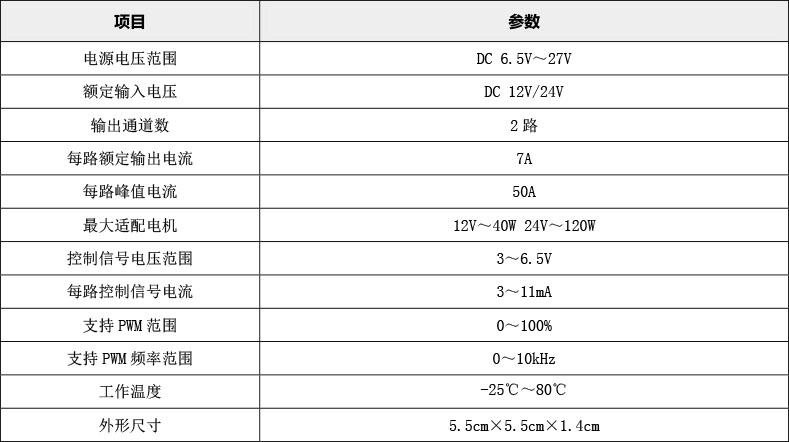
1. 控制系统设计
2. 控制板以STM32F103RE芯片

STM32F407VET6是一款高性能基础ARM Cortex-M4 32位微控制器, 100引脚LQFP封装. 该微控制器包含168MHz运行频率32位RISC核心. Cortex-M4核心具有浮点运算单元 (FPU)单精准度, 支持所有ARM单精准度数据处理指令与数据类型. 它还可以运行全套DSP指令, 以及包含1个用于增强应用程序安全性的内存保护单元 (MPU). STM32F407VET6包含高速嵌入式存储器, 以及多种增强型I/O与外设连接至2条APB总线, 3条AHB总线与32位多AHB总线阵列. 该MCU具有12位模数转换器, 数模转换器, 低功率RTC, 通用16位计时器 (包括PWM计时器), 通用32位计时器, 标准与高级通讯接口. 多种省电模式允许设计低功耗应用.

* 512K字节闪存
* 196K字节SRAM, 包括64K字节CCM (核心耦合存储器)数据RAM
* 3个12位, 2.4MSPS 模数转换器与2个12位数模转换器
* 运行电压范围: 1.8V至3.6V
* 16流DMA控制器, 带FIFO与突发支持
* 10个通用与2个高级控制计时器
* 82个I/O端口
* 以太网, 3个SPI, 2个I2S, 3个I2C, 4个USART, 2个UART, 2个CAN, SDIO, USB接口
* 8位至14位并行相机接口, 最高可达54Mbytes/s
* 环境温度范围: -40°C至85°C

1. 大功率H桥 直流电机驱动模块

此模块使用门电路、与MOS管组合方式实现电机正反转、制动及调速控制。既有较大的输出电流又有类似L298灵活的控制信号逻辑。



1. 电源设计

根据控制板和电机驱动模块的电压要求准备3s航模锂电池和LM2596S DC-DC直流可调降压稳压电源模块板，通过四根导线将电池分别与LM2596S DC-DC直流可调降压稳压电源模块板和直流电机驱动模块相连，再用LM2596S DC-DC直流可调降压稳压电源模块以合适电压与控制板相连

1. 舵机设计

机器人采用**SG90s舵机，DS3120 20kg舵机和DS3230 30kg定制360°可控舵机，分别控制圆盘连杆，圆筒升降和圆盘旋转。工作方式是通过控制板输出pwm信号控制舵机转动角度**

1. 灰度设计

灰度采用单路数字灰度，用于前后循迹和机器人定位，

# 软件设计

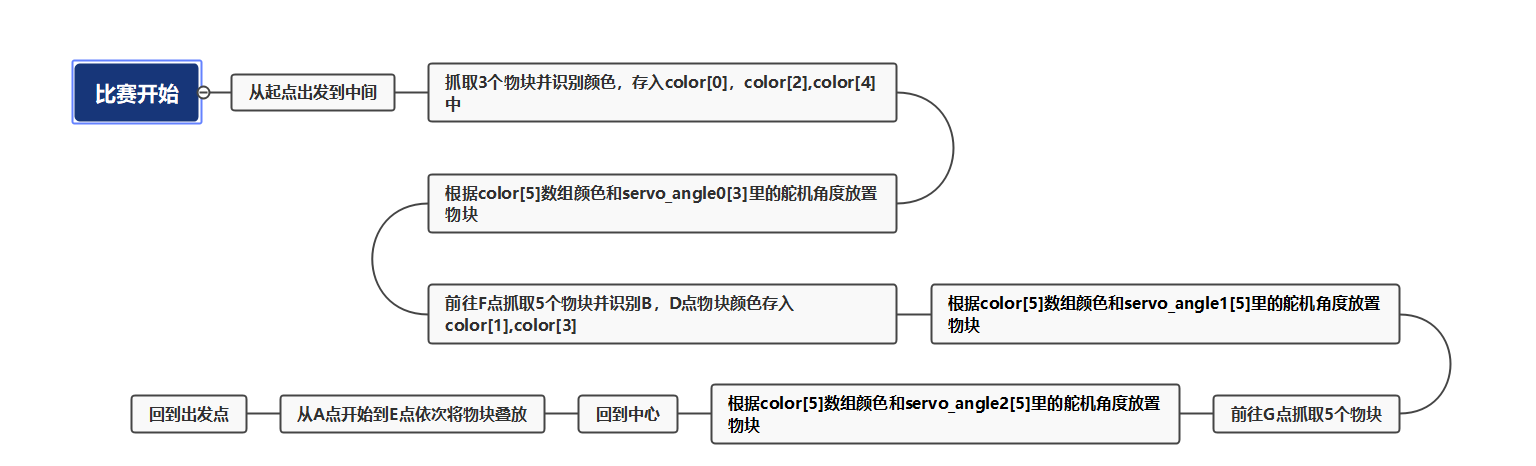
算法策略：

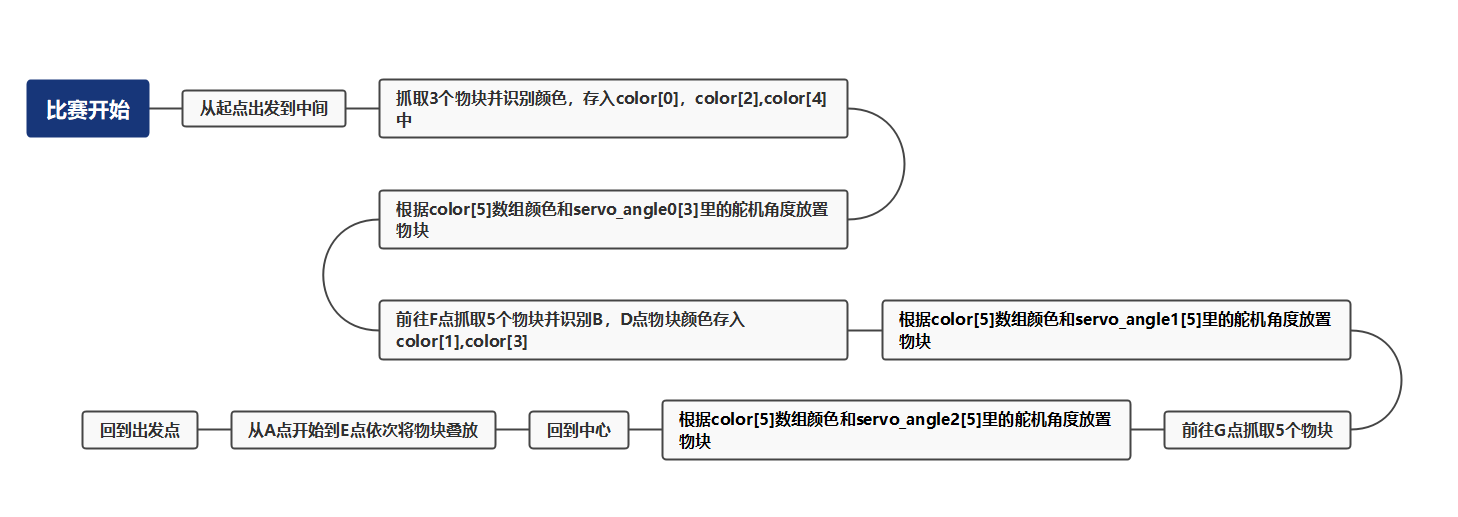
我们的车使用的编程环境是keil5:keil5是一个实用、灵活、用户友好的开源电子原型平台

该方案的编写方式:正如搬运小车光电工程组的规则所表明的那样，有许多类型的任务。一开始我们也很苦恼，我们会写很多很多的程序来处理这种可能性吗?根据老师的建议，以及我们对小车在完成任务时的运动和路径的观察和分析，我们为每种情况设计了一个无所不在的程序。程序分析:小车从圆的中心开始，然后转一个相应的角度，到达一个点，转到中心，再转一个角度，转到一个方向，再转回来。这是一个接一个的重复。我们设置了一些特定的角度变量，整个任务被分成三个小的模块，模块由三个小的重复动作组成，然后是小的子模块，也就是子模块，等等，这些都是次要的过程，所以我们的过程变得简单和模块化。然后可以在我们的主要课程中使用这些模块，并执行相应的功能。

对比赛规则的分析表明，参赛队伍通过抽签，从五种颜色中确定了A、B、C、D、E对应方块的颜色顺序。.第三环节将G,F区物料堆的另五个颜色物料推到对应颜色区域,小环ABCDE五个区标记,我们得到,写一个算法输入通道,5块的颜色,可以产生旋转绘画其整个长度上的核心,使推车推到区对应颜色的块。

分析之后,显然存在有重复行为的关系之间的旋转角度，角度通过时间和灰度控制。car\_angle[i]-=angle;

if(car\_angle[i]>2.0 car\_angle[i]=car\_angle[i]-4.0;if(car\_angle[i]<=-2.0) car\_angle[i]=car\_angle[i]+4.0;



#include "stm32f4xx.h"

#include "usart.h"

#include "delay.h"

#include "motor.h"

#include "pid.h"

#include "tcs3200.h"

#include "servo.h"

#include "math.h"

#include "truck.h"

#include "exit.h"

//左轮正转

#define left\_f GPIO\_SetBits(GPIOE,GPIO\_Pin\_9);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_10)

//左轮反转

#define left\_b GPIO\_SetBits(GPIOE,GPIO\_Pin\_10);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_9)

//右轮正转

#define right\_f GPIO\_SetBits(GPIOE,GPIO\_Pin\_11);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_12)

//右轮反转

#define right\_b GPIO\_SetBits(GPIOE,GPIO\_Pin\_12);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_11)

//刹车

#define stop GPIO\_ResetBits(GPIOE,GPIO\_Pin\_9);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_10);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_12);\

GPIO\_ResetBits(GPIOE,GPIO\_Pin\_11)

//读取灰度

#define huidu hfm=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_6);\

hfr=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_7);\

hfl=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_8);\

hbm=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_9);\

hbr=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_10);\

hbl=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_11);\

hd1=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_1);\

hd2=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_2);\

hd3=GPIO\_ReadInputDataBit(GPIOC,GPIO\_Pin\_3)

//前进

#define forward huidu;\

left\_f;\

right\_f;\

if(hfl==1&&hfm==0&&hfr==0)\

huidu\_flag=0;\

if(hfl==1&&hfm==1&&hfr==0)\

huidu\_flag=1;\

if(hfl==0&&hfm==1&&hfr==0)\

huidu\_flag=2;\

if(hfl==0&&hfm==1&&hfr==1)\

huidu\_flag=3;\

if(hfl==0&&hfm==0&&hfr==1)\

huidu\_flag=4;\

if(hfl==1&&hfm==1&&hfr==1)\

huidu\_flag=2;\

xunji\_f(huidu\_flag)

//后退

#define back huidu;\

left\_b;\

right\_b;\

if(hbl==1&&hbm==0&&hbr==0)\

huidu\_flag=0;\

if(hbl==1&&hbm==1&&hbr==0)\

huidu\_flag=1;\

if(hbl==0&&hbm==1&&hbr==0)\

huidu\_flag=2;\

if(hbl==0&&hbm==1&&hbr==1)\

huidu\_flag=3;\

if(hbl==0&&hbm==0&&hbr==1)\

huidu\_flag=4;\

if(hbl==1&&hbm==1&&hbr==1)\

huidu\_flag=2;\

xunji\_b(huidu\_flag)

//右转

#define turn\_right huidu;\

left\_f;\

right\_b;\

TIM\_SetCompare1(TIM5,800);\

TIM\_SetCompare2(TIM2,800)

//左转

#define turn\_left huidu;\

left\_b;\

right\_f;\

TIM\_SetCompare1(TIM5,800);\

TIM\_SetCompare2(TIM2,800)

//圆筒舵机角度

#define up 2000

#define down 500

//360圆盘舵机角度

#define A1 700

#define A2 1600

#define B1 2300

#define B2 1000

#define C1 1900

#define C2 1700

#define D1 1500

#define D2 2200

#define E1 1100

#define E2 810

//舵机角度输出

#define servo\_360(angle) TIM\_SetCompare1(TIM12,angle)

#define servo\_back1(angle) TIM\_SetCompare1(TIM9,angle)

#define servo\_back2(angle) TIM\_SetCompare1(TIM11,angle)

#define servo\_A(angle) TIM\_SetCompare2(TIM3,angle)

#define servo\_B(angle) TIM\_SetCompare1(TIM4,angle)

#define servo\_C(angle) TIM\_SetCompare2(TIM4,angle)

#define servo\_D(angle) TIM\_SetCompare3(TIM4,angle)

#define servo\_E(angle) TIM\_SetCompare4(TIM4,angle)

//时间

extern double tim;

//灰度数值储存

int hfm,hfr,hfl;

int hbm,hbr,hbl;

int hd1,hd2,hd3,hd4;

int huidu\_flag,stop\_flag=0;

int i,j,angle;

enum c{star,f5,green,white,red,black,blue,g5};

enum e{a1,b1,c1,d1,e1,a2,b2,c2,d2,e2};

//旋转角度时间

double car\_angle[8]={2.0,1.5,1.0,0.5,0,(-0.5),(-1.0),(-1.5)};

//物块颜色储存

int color[5];

//360圆盘角度顺序

int servo\_angle0[3]={2,4,0};

int servo\_angle1[5]={2,1,4,3,0};

int servo\_angle2[5]={4,0,2,3,1};

//前进速度控制

void xunji\_f(int w)

{

switch(w)

{

case 0: motor\_set(700,1400); break; //111 1 011

case 1: motor\_set(1000,1300); break; //111 0 011

case 2: motor\_set(1200,1200); break; //111 0 111 //正中间位置

case 3: motor\_set(1300,1000); break; //110 0 111

case 4: motor\_set(1400,700); break; //110 1 111

default: motor\_set(1200,1200); break;

}

}

//后退速度控制

void xunji\_b(int u)

{

switch(u)

{

case 0: motor\_set(1200,700); break; //111 1 011

case 1: motor\_set(1100,900); break; //111 0 011

case 2: motor\_set(1000,1000); break; //111 0 111 //正中间位置

case 3: motor\_set(900,1200); break; //110 0 111

case 4: motor\_set(700,1100); break; //110 1 111

default: motor\_set(1000,1000); break;

}

}

//旋转角度时间更改

void car\_angle\_change(double angle)

{

int i;

for(i=0;i<8;i++)

{

car\_angle[i]-=angle;

if(car\_angle[i]>2.0) car\_angle[i]=car\_angle[i]-4.0;

if(car\_angle[i]<=-2.0) car\_angle[i]=car\_angle[i]+4.0;

}

}

//转向函数

void turn(double angle)

{

tim=0;

double ang=angle;

if(angle>0)

{

while(1)

{

turn\_left;

if(tim>angle) if(hfm==1) break;

}

}

else if(angle<0)

{

while(1)

{

turn\_right;

if(tim>-angle) if(hfm==1) break;

}

}

tim=0;

while(1){stop;if(tim>0.5) break;}

car\_angle\_change(ang);

}

//各个舵机控制函数

void servo(int a,int c)

{

switch(a)

{

case 0: servo\_360(A1); break;

case 1: servo\_360(B1); break;

case 2: servo\_360(C1); break;

case 3: servo\_360(D1); break;

case 4: servo\_360(E1); break;

case 5: servo\_360(A2); break;

case 6: servo\_360(B2); break;

case 7: servo\_360(C2); break;

case 8: servo\_360(D2); break;

case 9: servo\_360(E2); break;

default:break;

}

switch(a)

{

case 0: servo\_A(c); break;

case 1: servo\_B(c); break;

case 2: servo\_C(c); break;

case 3: servo\_D(c); break;

case 4: servo\_E(c); break;

default:break;

}

}

int main(void)

{

//外设初始化

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

delay\_init(84);

EXTIX\_Init();

Motor\_GPIO\_init();

TRUCK\_GPIO();

tcs3200\_init(1,1);//n=clour\_disguish();

MotorL\_PWM1\_Init(5000-1,1);

MotorR\_PWM2\_Init(5000-1,1);

SERVO\_TIM12\_Init(20000-1,84-1);

SERVO\_TIM9\_Init(20000-1,168-1);

SERVO\_TIM11\_Init(20000-1,168-1);

SERVO\_TIM3\_Init(20000-1,84-1);

SERVO\_TIM4\_Init(20000-1,84-1);

TIM7\_Int\_Init(10000-1,84-1);

servo\_360(C1);

servo\_A(up);

servo\_B(up);

servo\_C(up);

servo\_D(up);

servo\_E(up);

servo\_back2(1000);

servo\_back1(1500);//1380

delay\_ms(2000);

while(1){

//////////////////////////////////////////////////////////////////////第一次前进

tim=0;

while(1)

{

forward;

servo(c1,up);

if(tim>3) {if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.1) break;}

servo(c1,down);

delay\_ms(500);

servo(c2,down);

delay\_ms(500);

color[0]=clour\_disguish();

delay\_ms(100);

////////////////////////////////////////////////////////////////后退

tim=0;

while(1)

{

back; //1.5

if(tim>0.2) {if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

/////////////////////////////////////////////////////////////////////////右转向

turn(car\_angle[blue]);

///////////////////////////////////////////////////////////////////////////前进

tim=0;

while(1)

{

forward;

servo(e1,up);

if(tim>0.2) {if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo(e1,down);

delay\_ms(500);

servo(e2,down);

delay\_ms(500);

color[2]=clour\_disguish();

delay\_ms(100);

///////////////////////////////////////////////////////////////////////////////////后退

tim=0;

while(1)

{

back;

if(tim>0.2){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

/////////////////////////////////////////////////////////////////////////////转

turn(car\_angle[green]);

/////////////////////////////////////////////////////////////////////////////前进

tim=0;

while(1)

{

forward;

servo(a1,up);

if(tim>0.2) {if(hd3==1) {delay\_ms(1); huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.1) break;}

servo(a1,down);

delay\_ms(500);

servo(a2,down);

delay\_ms(500);

color[4]=clour\_disguish();

delay\_ms(100);

///////////////////////////////////////////////////////////////////////////////////后退

tim=0;

while(1)

{

back;

if(tim>0.2){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

///////////////////////////////////////////////////////

for(i=0,j=0;i<3;i++,j=j+2)

{

turn(car\_angle[color[j]]);

tim=0;

while(1)

{

forward;

servo(servo\_angle0[i],down);

if(tim>1.5) {if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.1) break;}

servo(servo\_angle0[i],up);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>1.5){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.1) break;}

}

////////////////////////////////////////////////////////////////

//

//

//

//

//

//////////////////////////////////////////////////////////////////

turn(car\_angle[f5]);

/////////////////////////////////////////////////////

tim=0;

while(1)

{

forward;

servo(d1,up);

if(tim>1) {if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

tim=0;

while(1)

{

forward;

if(tim>0.55) break;

}

tim=0;

while(1){stop;if(tim>0.5) break;}

servo\_A(down);

servo\_B(down);

servo\_C(down);

servo\_D(down);

servo\_E(down);

delay\_ms(1000);

delay\_ms(500);

servo(b2,down);

delay\_ms(500);

color[1]=clour\_disguish();

tim=0;

while(1)

{

back;

if(tim>2){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

////////////////////////////////////////////////

for(i=0;i<5;i++)

{

turn(car\_angle[color[i]]);

tim=0;

while(1)

{

forward;

servo(servo\_angle1[i],down);

if(tim>1.1) break;

}

tim=0;

while(1){stop;if(tim>0.1) break;}

servo(servo\_angle1[i],up);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>1.2){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.1) break;}

}

//////////////////////////////////////////////////////////////

//

//

//

//

////////////////////////////////////////////////////////////////

turn(car\_angle[g5]);

/////////////////////////////////////////////////////

tim=0;

while(1)

{

forward;

servo(d1,up);

if(tim>1.2) {if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

tim=0;

while(1)

{

forward;

if(tim>0.55) break;

}

tim=0;

while(1){stop;if(tim>0.2) break;}

servo\_A(down);

servo\_B(down);

servo\_C(down);

servo\_D(down);

servo\_E(down);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>2){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.2) break;}

////////////////////////////////////////////////

for(i=0;i<5;i++)

{

turn(car\_angle[color[i]]);

tim=0;

while(1)

{

forward;

servo(servo\_angle2[i],down);

if(tim>0.2){if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1){stop;if(tim>1) break;}

servo(servo\_angle2[i],up);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>0.5){if(hd1==1){delay\_ms(1);huidu;if(hd1==1)break;}}

}

tim=0;

while(1){stop;if(tim>0.5) break;}

}

////////////////////////////////////////////////////

for(i=0;i<5;i++)

{

delay\_ms(500);

turn(-0.5);

if(i==1||i==2||i==3)

{

tim=0;

while(1)

{

back;

servo\_back1(1450);

servo\_back2(1500);

if(tim>0.5){if(hd2==1) {delay\_ms(1);huidu;if(hd2==1) break;}}

}

tim=0;

while(1)

{

back;

servo\_back1(1450);

servo\_back2(1500);

if(tim>0.1) break;

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo\_back2(500);

delay\_ms(2000);

servo\_back1(1650);

delay\_ms(2000);

tim=0;

while(1)

{

back;

if(tim>0.5){if(hd3==1) {delay\_ms(1);huidu;if(hd3==1) break;}}

}

tim=0;

while(1)

{

back;

if(tim>0.1)break;

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo\_back2(1500);

delay\_ms(1000);

servo\_back1(1450);

delay\_ms(1000);

servo\_back2(500);

delay\_ms(1000);

servo\_back1(1650);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>0.5){if(hd2==1) {delay\_ms(1);huidu;if(hd2==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo\_back2(1500);

delay\_ms(1000);

servo\_back1(1450);

delay\_ms(1000);

servo\_back2(500);

delay\_ms(1000);

servo\_back2(1500);

delay\_ms(1000);

tim=0;

while(1)

{

forward;

if(tim>0.5){if(hd1==1&&hd2==1) break;}

}

while(1){stop;if(tim>0.3) break;}

}

else

{

tim=0;

while(1)

{

back;

servo\_back1(1450);

servo\_back2(1500);

if(tim>1.0){if(hd2==1) {delay\_ms(1);huidu;if(hd2==1) break;}}

}

tim=0;

while(1)

{

back;

if(tim>0.2)break;

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo\_back2(1500);

delay\_ms(1000);

servo\_back1(1450);

delay\_ms(1000);

servo\_back2(500);

delay\_ms(1000);

servo\_back1(1650);

delay\_ms(1000);

tim=0;

while(1)

{

back;

if(tim>0.5){if(hd2==1) {delay\_ms(1);huidu;if(hd2==1) break;}}

}

tim=0;

while(1){stop;if(tim>0.3) break;}

servo\_back2(1500);

delay\_ms(1000);

servo\_back1(1450);

delay\_ms(1000);

servo\_back2(500);

delay\_ms(1000);

servo\_back2(1500);

delay\_ms(1000);

tim=0;

while(1)

{

huidu;

left\_f;

right\_f;

if(hfl==1&&hfm==0&&hfr==0)

huidu\_flag=0;

if(hfl==1&&hfm==1&&hfr==0)

huidu\_flag=1;

if(hfl==0&&hfm==1&&hfr==0)

huidu\_flag=2;

if(hfl==0&&hfm==1&&hfr==1)

huidu\_flag=3;

if(hfl==0&&hfm==0&&hfr==1)

huidu\_flag=4;

if(hfl==1&&hfm==1&&hfr==1)

huidu\_flag=2;

switch(huidu\_flag)

{

case 0: motor\_set(500,1200); break; //111 1 011

case 1: motor\_set(800,1100); break; //111 0 011

case 2: motor\_set(1000,1000); break; //111 0 111 //正中间位置

case 3: motor\_set(1100,800); break; //110 0 111

case 4: motor\_set(1200,500); break; //110 1 111

default: motor\_set(1100,1000); break;

}

if(tim>2.5){if(hd2==1){delay\_ms(1);huidu;if(hd2==1)break;}}

}

while(1){stop;if(tim>0.3) break;}

}

}

delay\_ms(500);

turn(car\_angle[star]);

tim=0;

while(1)

{

forward;

if(tim>2){if(hd2==1) {delay\_ms(1);huidu;if(hd2==1) break;}}

}

while(1){stop; }

}

}

# 系统开发与调试

循迹调试

舵机调试

颜色传感器调试

转向调试