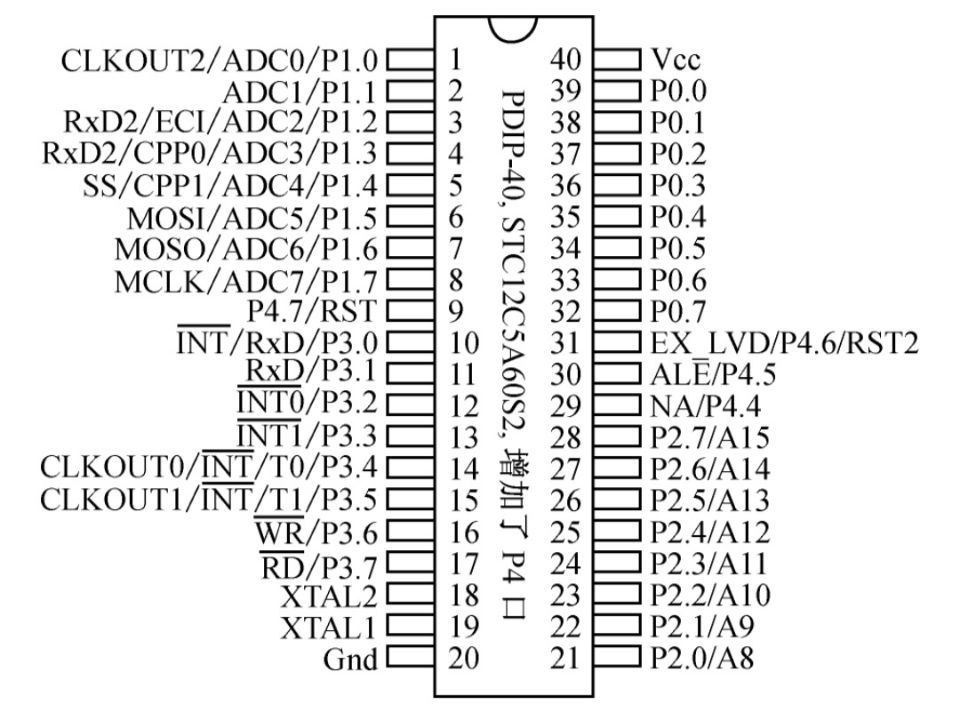
**基于STC12c5a60s2单片机循迹搬运小车装置**

**摘 要 ：**本装置基于stc12c5a60s2单片机为控制核心，stc89c52rc为辅助单片机。整体装置由单片机部分，传感器部分，电机驱动与电源组成。该小车具有循迹黑线的功能并可避免十字路口交叉干扰，能稳定做出指定转向并继续稳定行驶于黑线道路上。通过键盘扫描可在下机位设置其所需抓取的矿泉水瓶并运送至终点。

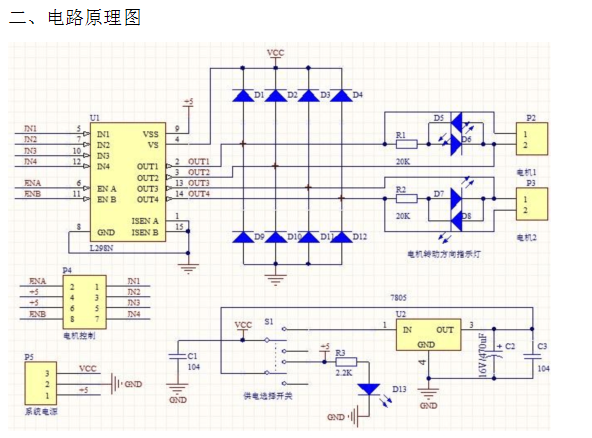
**关键词：**循迹小车；搬运小车；STC12c5a60s2；引导线识别；PWM调速控制;PWM舵机控制;LM2596DC-DC降压模块；L298N电机驱动；7805稳压模块

**一、方案设计于论证**

小车可分为电源模块、驱动模块、循迹模块、抓取模块、显示模块和按键模块。电源模块由12.6V锂电池组、LM2596DC-DC降压模块、7805稳压芯片组成。由LM2596DC-DC降压模块将12.6V电压降低并稳定到9V，供给L298N电机驱动。再由7805稳压芯片将9V电压降低并稳定在5V，供给单片机和其他部分使用。其中，L298N芯片电源由L298N板载78M05稳压芯片提供。驱动模块由L298N电机驱动和2台直流小电动机组成，由L298N电机驱动驱动2台小电机。循迹模块由5颗TCRT5000红外反射传感器组成，使用LM393比较器，由外部颜色差异返回不同信号。抓取模块由一颗TCRT5000红外反射传感器、2台SG90舵机和夹爪组成，由TCRT5000红外反射传感器测量距离，两台舵机分别控制左右夹爪，实现抓取。显示模块由2片595驱动的数码管构成，从而显示时间。按键模块由3个微动开关组成，一端连接单片机io口，一端接地，按下后将IO口置低，使单片机感受到低电平。



小车循迹前，通过按键的置低为单片机确定线路或选定模式循迹时，开始循迹后，由TCRT5000红外反射传感器的红外对管向地面发射红外信号，白色地面反射光线较多，黑色地面反射光线较少，红外对管识别不同反射强度，经处理后向单片机返回不同电平。单片机读取电平，做出反馈，通过L298N电机驱动，以控制占空比的方式控制电机电压，实现转速调整。以差速方式实现小车的方向调整，从而达到循迹目的。当小车到达需要转向路口时，通过TCRT5000红外反射传感器探测路口位置，同样通过差速方式使小车调整方向，实现转弯。当小车与矿泉水瓶距离较近时，TCRT5000O红外反射传感器接收到反射光线，给单片机返回一个高电平，单片机接收后，通过脉冲使机械爪抓紧，夹住矿泉水瓶。到达终点后，再停止脉冲输出，松开机械爪。



**二、理论分析与计算**

在电源模块中，使用LM2596DC-DC降压模块和7805稳压芯片两级降压原因：

7805的额定最大电流为1A，若将12.6V直接降为5V，压降较大，且有功率较大的电机，导致电流也偏大，使7805稳压芯片发热严重，易烧毁，且损耗较高，而电机转速却较低。因此使用LM2596DC-DC降压模块将电压降至9V，供给电机，再使用7805稳压芯片将电压降至5V。因LM2596DC-DC降压模块额定电流较大，且发热较少，且7805稳压芯片不再为电机供电，电流较小，使发热量整体减少，且电机转速可调至较快，损耗也相对减小。

**三、系统软件设计**

**3.1 系统软件**

系统软件采用C语言编写，采用模拟PWM信号来控制直流电机的两轮转速，主程序先进行系统初始化，键盘扫描读取抓取目标编号，配置所使用的寄存器，向辅助单片机发出计时指令，然后进入循迹模式。通过定时器0中断不断扫描传感器连接IO口状态，并做出左右电机调速行为，以达到引导线循迹功能。当小车进入任务区时，会根据最左右两侧传感器计数，以此按照指定规划路线前进，此路线设计与键盘扫描相结合对应生成3种不同的左转判断条件，达到指定抓取。当小车行驶到矿泉水瓶前方，触发红外传感器，此时辅助单片机做出抓取动作。当车通过计数判断抵达终点，则整车停止向辅助单片机发出指令停止计时并放开瓶子。

**3.2系统初始化函数程序**

void init(){

TMOD=0x01; //设置定时器模式为16位

AUXR=0xc1; //启动stc12c5a60s2的1T模式，以此提高模拟PWM的频率

TH1=0xff; //用于电机PWM的模拟源

TL1=0xf9;

TH0=0xdb; //用于循迹判断中各IO口的扫描

TL0=0xfc;

EA=1; //开启总中断

ET0=1;

ET1=1;

TF0=0;

TF1=0;

TR0=1; //T0与T1定时器开始工作

TR1=1;

}

**3.3 键盘扫描函数程序**

void keyscan(){ //键盘扫描检测按键以及发出GO指令

key=1; //确保各按键初始值为1

key\_go=1;

while(1){

if (key==0){

delay(40); //使用程序消除抖动

if (key==0){

nb++; //用于计数规划路线，确定其抓取ABC指定矿泉水瓶

while(!key); //等待松开按键再继续循环扫描

} }

if (key\_go==0){

delay(50);

if (key\_go==0){

while(!key\_go);

Go=0; /\*此指令将主单片机一脚设为低电平，与辅助单片机连接，拉低辅助单片机一脚电平，实现计时口令\*/

break;

}

}

}

}

**3.4 循迹逻辑判断程序**

/\*采用双传感器逻辑判断，牺牲速度，但是确保其稳定程度。当两个传感器距离设置合适时，能够将曲线不断的相切，做到以折线拟合曲线，从而确保稳定程度。\*/

void xunji(){

if ((L1 == 1) && (R1 == 0)) turn\_left1();

else if ((L1 == 0) && (R1 == 1)) turn\_right1();

else if ((L1 == 0) && (R1 == 0)) qianjin();

else if ((L1 == 1) && (R1 == 1)) qianjin();

}

**四、测试方法与结果分析**

**4.1循迹测试**

将小车置于比赛道路上，进行实地测试，多次测试调整过弯参数以及传感器放置数量位置以及角度，从而确保整体系统稳定性。测试前后结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 测试前 | 测试后 | 修改原因 |
| 循迹传感器数量 | 5个 | 2个 | 观察发现，传感器过多，实际循迹基本只使用了中部两颗传感器 |
| 传感器放置方位 | “一”字型排放 | 前后放置 | 避开不同作用传感器工作时干扰 |
| 传感器高度 | 距离地面2mm | 距离地面8mm | 因地面不平，过低导致传感器误判 |

**4.2 舵机测试**

因为使用SG90模拟舵机，所以需要使用模拟PWM作为调节方式，因为收到晶振以及程序运行干扰，导致其与理论值不服，使用示波器作多次调节以后可以做到准确抓取瓶子，有效解决测试之初舵机严重抖动问题。

**4.3 键盘扫描测试**

因为键盘扫描存在抖动，仅有两个独立键盘则使用程序消抖解决，为确保准确设定延时参数，设置了一次应该做出的反应，多次实验确定使用delay（40）与delay（50）两个参数作为消抖参数。

**五、总结分析与结论**

本次比赛以小组的形式完成，在我们这个小组中，分工明确，根据每个人的优点来明确分工扬长避短，尽可能地节省时间，提高效率。除了做好自己的事，组员之间也会互相帮助，众人拾柴火焰高。从最初实验材料的选择到小车的制作，单片机的选择，电源等的选择等我们经历了大量的讨论，最终确定了型号。在焊板子的时候，看似简单，实则只要不小心就会出错误，小车就制作失败了，经历多次的失败，终于小车做好了。到了写程序的时候，我们刚刚开始学C语言，有很多的知识还掌握的不透彻，调试的过程出现了不少问题，不断的调试，不断的发现问题，一直这样，循环往复，我已经不记得我们调试了多少次了，有那么一刻想放弃，队员之间相互鼓励，加油打气，一路磕磕绊绊走过来。进入了复赛，队员之间信心增加了不少，积极性也高了，改进程序，不断的测试。这次比赛充分体现了学和用的紧密结合，体会到了比赛的乐趣，尽管有不开心，但是快乐大于难过，这次比赛强化了大家合作的意识，提高了个人的能力。

**[参考文献]**

[1] 王盼宝.智能车制作[M].北京:清华大学出版社,2018.

[2] 孙鹏.51单片机C语言学习之道[M].北京:清华大学出版社,2018.

[3] 马忠梅，籍顺心.单片机的C语言应用程序设计[M].第4版.北京:北京航空航天大学出版社,2007.

[4]张巍，苏闯.单片机C51项目设计与开发[M].北京:北京理工大学出版社,2013.

[5]李令伟，周中孝，黄文涛，王宛增.嵌入式C语言实战教程[M].北京:电子工业出版社,2014.

**附录**

1. **STC12c5a60s2(源代码)**

**#include <12c5a60s2.h>**

**#define uchar unsigned char**

**unsigned char zkb1 = 0;//左边电机的占空比**

**unsigned char zkb2 = 0;//右边电机的占空比**

**unsigned char t = 0;//定时器中断计数器**

**unsigned char nb = 11;//用作判断是否掉头**

**unsigned char nbn = 0;//用作判断是否停车**

**unsigned char def = 1;//用作判断是否停车**

**void xunji();**

**sbit key=P2^4;**

**sbit key\_go=P2^1;**

**sbit LG = P1 ^ 4;**

**sbit LB = P1 ^ 3;**

**sbit RB = P1 ^ 2;**

**sbit RG = P1 ^ 0;**

**sbit ENR = P1 ^ 1;**

**sbit ENL = P1 ^ 5;**

**sbit R1 = P0 ^ 1;**

**sbit L1 = P0 ^ 0;**

**sbit C1 = P0 ^ 4;**

**sbit L2=P0^2;**

**sbit R2=P0^3;**

**sbit Go=P0^7;**

**void delay(unsigned int m){ //延时函数，为0.5ms的倍数**

**unsigned int j,i;**

**for(i=0;i<m;++i)**

**for(j=0;j<248;++j);**

**}**

**void keyscan(){ //键盘扫描检测按键以及发出GO指令**

**key=1;**

**key\_go=1;**

**while(1){**

**if (key==0){**

**delay(40);**

**if (key==0){**

**nb++;**

**while(!key);**

**} }**

**if (key\_go==0){**

**delay(50);**

**if (key\_go==0){**

**while(!key\_go);**

**Go=0;**

**break;**

**}**

**}**

**}**

**}**

**void Timer0() interrupt 3{**

**if (t < zkb1) ENL = 1;**

**else ENL = 0;**

**if (t < zkb2) ENR = 1;**

**else ENR = 0;**

**t++;**

**if (t > 100) t = 0;**

**TH1=0xff;**

**TL1=0xf9;**

**}**

**void Timer1() interrupt 1{**

**if(def){**

**xunji();}**

**TL0 = 0xdb; //设置定时初值**

**TH0 = 0xfc; //设置定时初值**

**}**

**void qianjing() {**

**zkb1 = 25;**

**zkb2 = 24;**

**}**

**/\*左转函数1\*/**

**void turn\_left1() {**

**zkb1 = 18;**

**zkb2 = 40;**

**}**

**/\*右转函数1\*/**

**void turn\_right1() {**

**zkb1 = 39;**

**zkb2 = 14;**

**}**

**/\*左转函数4\*/**

**void turn\_left4() {**

**LB=1;LG=0;**

**do{**

**zkb1 = 22;**

**zkb2 = 23;**

**if(L1==1)break;**

**}while(1);**

**LB=0;LG=1;**

**turn\_right1();**

**delay(29);**

**}**

**void xunji(){**

**if ((L1 == 1) && (R1 == 0)) turn\_left1();**

**else if ((L1 == 0) && (R1 == 1)) turn\_right1();**

**else if ((L1 == 0) && (R1 == 0)) qianjing();**

**else if ((L1 == 1) && (R1 == 1)) qianjing();**

**}**

**void init(){**

**TMOD=0x01;**

**AUXR=0xc1;**

**TH1=0xff;**

**TL1=0xf9;**

**TH0=0xdb;**

**TL0=0xfc;**

**EA=1;**

**ET0=1;**

**ET1=1;**

**TF0=0;**

**TF1=0;**

**TR0=1;**

**TR1=1;**

**}**

**int main () {**

**Go=1;**

**keyscan();**

**init();**

**ENL=1;**

**ENR=1;**

**RB=0;**

**LB=0;**

**LG=1;**

**RG=1;**

**zkb1=40;**

**zkb2=40;**

**while(1){**

**if (C1==1){**

**delay(15);**

**if(C1==0) nb--;}**

**if (R2==1){**

**delay(80);**

**if(R2==0)nbn++;}**

**if(nb==8){**

**def = 0;**

**turn\_left4();**

**nb=7;**

**def =1;**

**}**

**else if(nb==4||nb==2||nb==0){**

**def = 0;**

**turn\_left4();**

**nb--;**

**def =1;**

**}**

**if (nbn ==9){**

**def=0;**

**zkb1=zkb2=0;**

**Go=1;**

**}**

**}**

**}**

1. **STC89c52rc（源代码）**

**#include <reg51.h>**

**#define uchar unsigned char**

**unsigned char code fseg[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};**

**unsigned char code segbit[]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};**

**unsigned char disbuf[4]={0,0,0,0};**

**unsigned char code LED\_0F[] = {// 0 1 2 3 4 5 6 7 8 9 A b C d E F -**

**0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x8C,0xBF,0xC6,0xA1,0x86,0xFF,0xbf**

**};**

**unsigned int cnt=0;//用作叠加到1s**

**unsigned char count=0;//用作叠加到1s**

**unsigned char dj1=0;//用作叠加到1s**

**unsigned char dj2=0;//用作叠加到1s**

**int sec=0; //计时多少s；**

**sbit djl=P1^2;**

**sbit djr=P1^1;**

**sbit Go=P1^0;**

**sbit carrt\_led=P1^3;**

**sbit DIO = P2^2; //串行数据输入**

**sbit RCLK = P2^1; //时钟脉冲信号——上升沿有效**

**sbit SCLK = P2^0; //打入信号————上升沿有效**

**void LED\_OUT(uchar X){ //数显管的移位操作**

**uchar i;**

**for(i=8;i>=1;i--)**

**{**

**if (X&0x80) DIO=1; else DIO=0;**

**X<<=1;**

**SCLK = 0;**

**SCLK = 1;**

**}**

**}**

**void LED4\_Display (){ //数显管的输出显示**

**unsigned char code \*led\_table; // 查表指针**

**uchar i;**

**//显示第4位**

**led\_table = LED\_0F +sec/1000%10;**

**i = \*led\_table;**

**LED\_OUT(i);**

**LED\_OUT(0x08);**

**RCLK = 0;**

**RCLK = 1;**

**//显示第3位**

**led\_table = LED\_0F + sec/100%10;**

**i = \*led\_table;**

**LED\_OUT(i);**

**LED\_OUT(0x04);**

**RCLK = 0;**

**RCLK = 1;**

**//显示第2位**

**led\_table = LED\_0F + sec/10%10;**

**i = \*led\_table;**

**LED\_OUT(i);**

**LED\_OUT(0x02);**

**RCLK = 0;**

**RCLK = 1;**

**//显示第1位**

**led\_table = LED\_0F + sec%10;**

**i = \*led\_table;**

**LED\_OUT(i);**

**LED\_OUT(0x01);**

**RCLK = 0;**

**RCLK = 1;**

**}**

**void Timer1() interrupt 1{**

**LED4\_Display();**

**cnt++;**

**TL0 = 0xdb; //设置定时初值**

**TH0 = 0xfc; //设置定时初值**

**}**

**void Timer0() interrupt 3{**

**if(count < dj2)**

**{djl = 1;}**

**else**

**{djl = 0;}**

**if(count < dj2) //控制占空比上下**

**{djr = 1;}**

**else**

**{djr = 0;}**

**count++;**

**if (count >= 138) //T = 20ms则定时器计数变量清0**

**{count = 0;}**

**TL1 = 0x2e; //设置定时初值**

**TH1 = 0xfF; //设置定时初值**

**}**

**int main () {**

**TMOD=0x01;**

**TH0=0xdb;**

**TL0=0xfc;**

**TH1=0x2e;**

**TL1=0xff;**

**EA=1;**

**ET1=1;**

**TF1=0;**

**TR1=1;**

**ET0=1;**

**TF0=0;**

**TR0=1;**

**djl=djr=0;**

**while (1){**

**if (carrt\_led==1&&Go==0){**

**dj1=dj2=7;**

**}**

**if(cnt>500&&Go==0){**

**sec++;**

**cnt=0;**

**}**

**}**

**}**