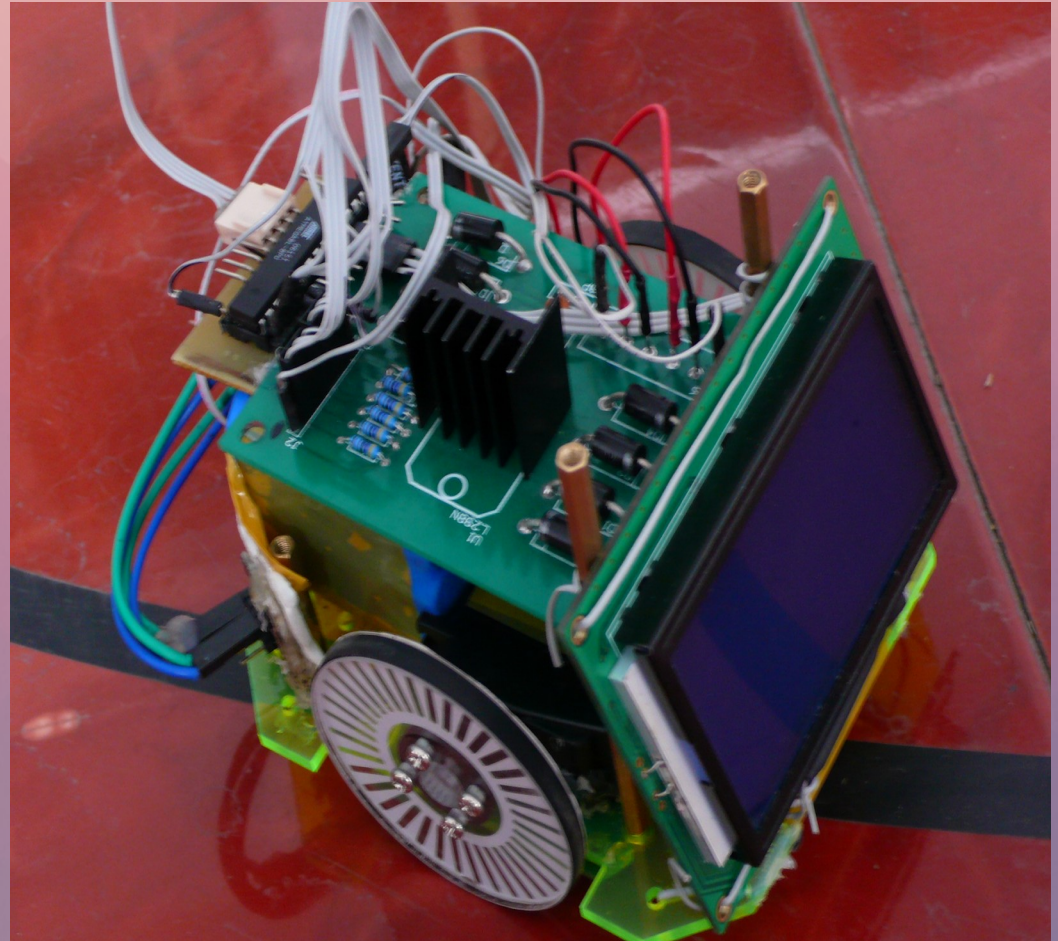
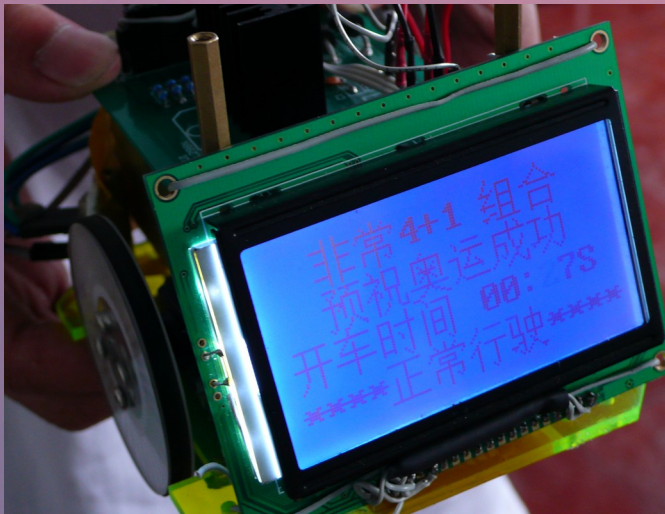
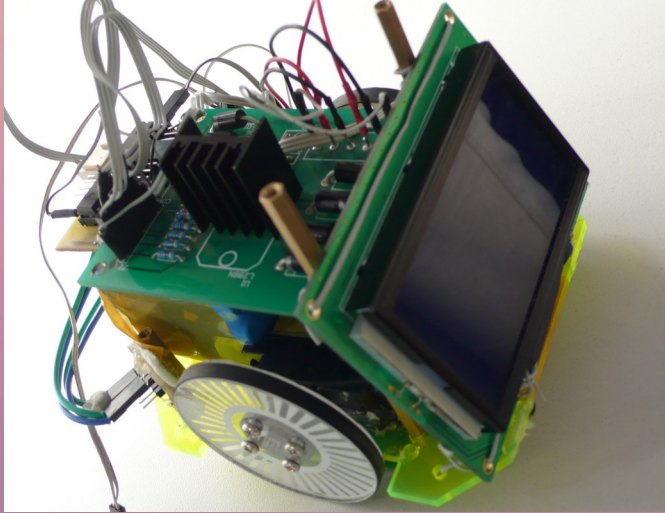


单片机课程设计

智能小
车

智能小车



智能小车

◆ **摘要：**本课题组设计制作了一款具有智能判断功能的小车，功能强大。小车具有以下几个功能：寻迹功能（按路面的黑色轨道行驶）；计算并显示行走的时间，并可掉头返回出发点。作品以两电动机为主驱动，通过各类传感器件来采集各类信息，送入主控单元 AVR 单片机（ ATmega8 ），处理数据后完成相应动作，以达到自身控制。电机驱动电路采用四通道驱动集成芯片 L293D ，采用 PWM 脉冲调制驱动电动机。黑带检测采用反射式光电对管完成；并通过 IO 口传送给操作系统，由控制单元处理数据后完成相应动作，实现了无人控制即可完成一系列动作，相当于简易机器人。

◆ **关键字：** 智能判断 寻迹行驶 自动入库 PWM 简易机器人

目 录

- ◆ 1. 引言
- ◆ 2. 系统设计
- ◆ 3. 模块设计
- ◆ 4. 系统原理及理论分析
- ◆ 5. 测试与分析

引言

- ◆ 课程设计的目的是提高我们大学生的专业技能，增强我们的动手能力。更重要的是让我们掌握更扎实的专业基础知识，学习更多的现代的最新科研技术，与当代社会飞快发展的科技共同前进。我们选择这个题目的原因是因为现在自动控制领域越来越向机器人、智能方向发展。

系 统 设 计

(1) 设计任务

设计自动寻迹小车系统，车体在以白色为背景的黑色实线上行驶。

(2) 设计要求

- ◆ 小车检测到黑线时，能沿着黑线前进。
- ◆ 小车能实时记录行驶时间。
- ◆ 小车遇到障碍物时，做出蔽障动作掉头返回。

系 统 设 计

总体方案论证

根据题目的要求，系统可以划分为控制部分和信号检测部分。其中信号检测

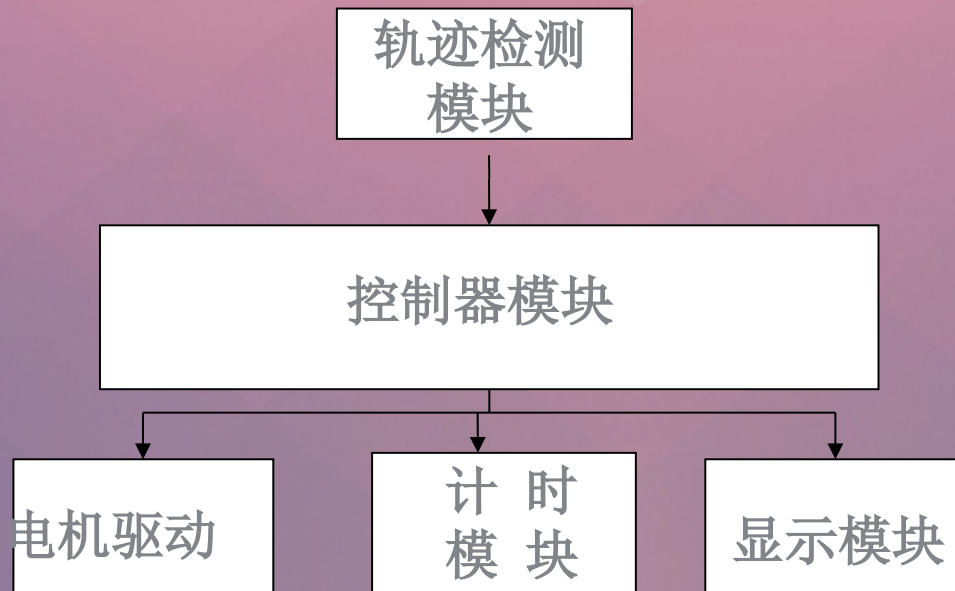


图 1-1 小车的基本模块方框图

系 统 设 计

- ◆ 部分包括：路面检测模块。
- ◆ 控制部分包括：电机驱动模块，显示模块，控制器模块，计时模块。
- ◆ 模块框图如图 1-1 所示。

模块设计

◆ 控制器模块

根基题目的要求，控制器主要用于各个传感器信号的接受和辨认、控制小车的电机的动作、控制显示小车运动的时间及路程。对于控制器的选择有以下方案。

◆ 方案：采用 ATMEL 公司的 AVR 系列 ATmega8 单片机作控制器。

◆ AVR 系列单片机是采用低功率、非挥发 CMOS 工艺制造的 Harvard 结构的新型单片机，具有 1MIP/MHz 的高速处理能力。ATmega8 系列单片机是 AVR 中高档产品具有丰富的片内资源，而且价格低廉性能稳定。并具有在线编程（ISP）的特点。给调试带来了极大的方便。

◆ 在本设计中基于多种比较我们使用 ATmega8 单片机。

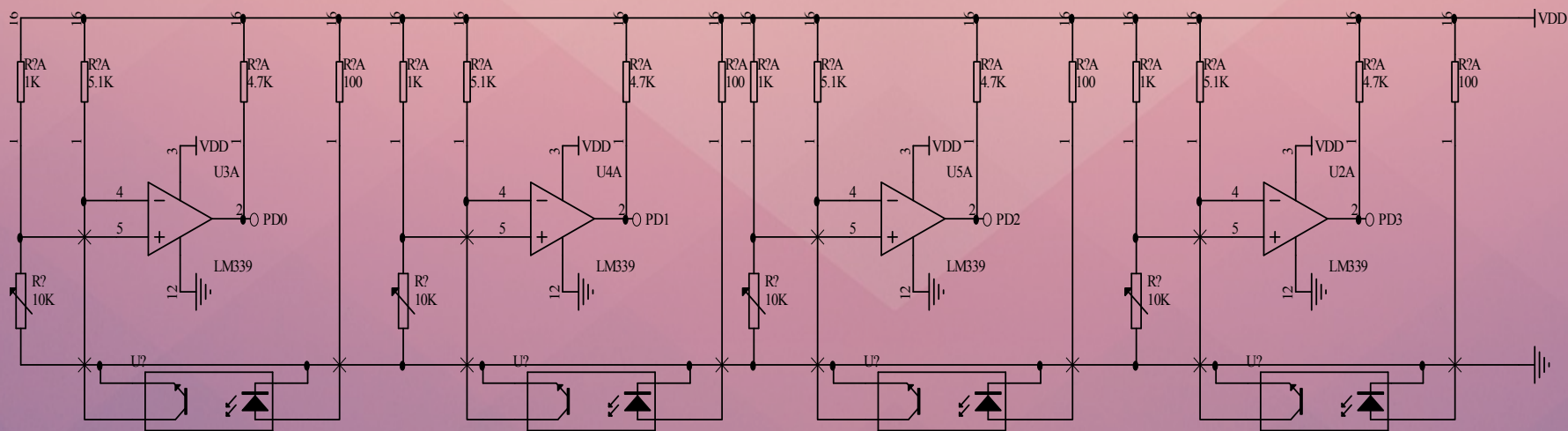
模块设计

◆ 寻迹模块

◆ 黑带检测方案:

- ◆ 由于黑白物体对光的反射和吸收程度不同，因此本系电路采用的是反射式光电管 TCRT5000，通过调整合适的分压阻值使输出信号可以达到在白纸上小于 0.7V、在黑纸上大于 4.5V 的高低电平。用四对 TCRT5000 组成检测黑线的光电传感电路，分别控制 4 个方向，可以预测轨迹，使物体能够沿线行走，本方案也易于实现，比较可靠，因此采用方案二。
- ◆ 输入信号进入 LM339 比较器，稳定性能得到提升。当小车低部的某边反射式光电对管遇到黑带时输入电平为高电平，反之为低电平。结合中断查询方式，通过程序控制小车往哪个方向行走。电路中的可调电阻可调节灵敏度，以满足小车在不同光度的环境光中能够寻迹。黑带检测电路图如图 2 所示。

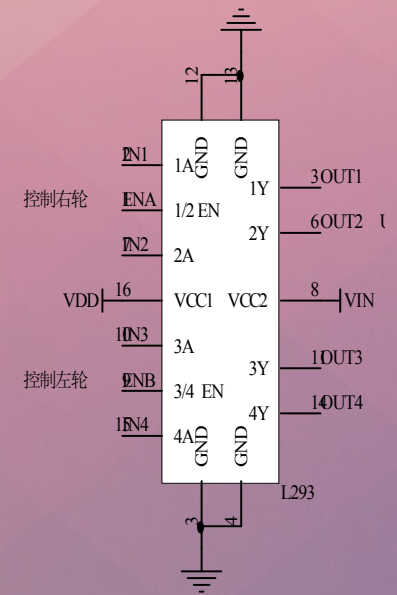
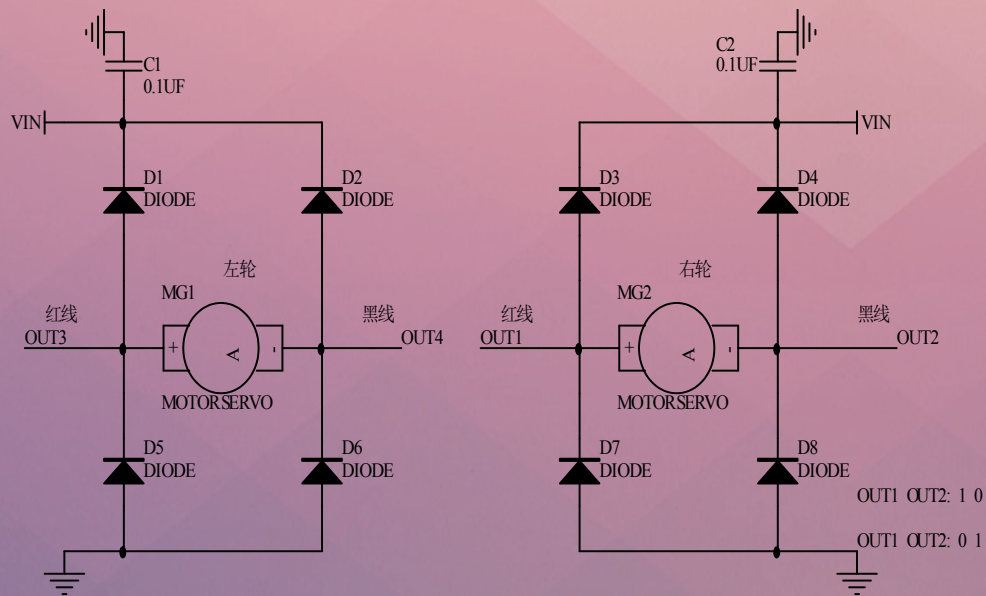
黑带检测电路图



参考电压: $U_{REF} = R2 / (R2 + R1)$

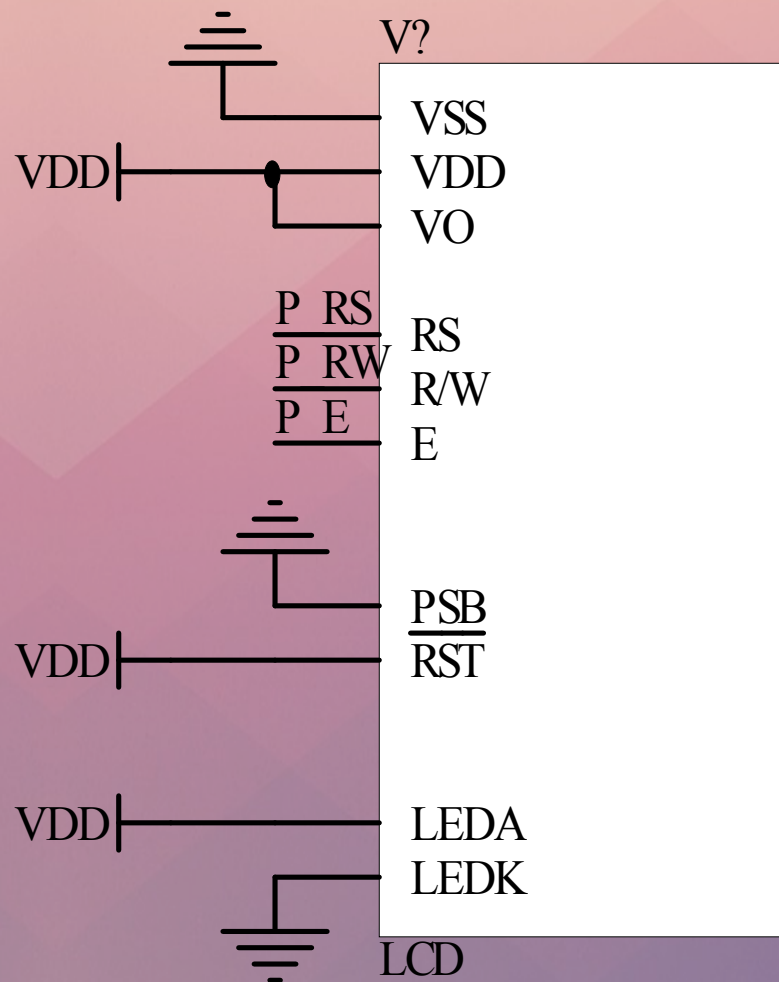
这里我们取 $R1$ 为 1K，则 U_{REF} 为 2.5V。当光电对管输出电压大于 2.5V，输出端口 2 为高电平 1，否则为低电平 0。

智能车驱动电路

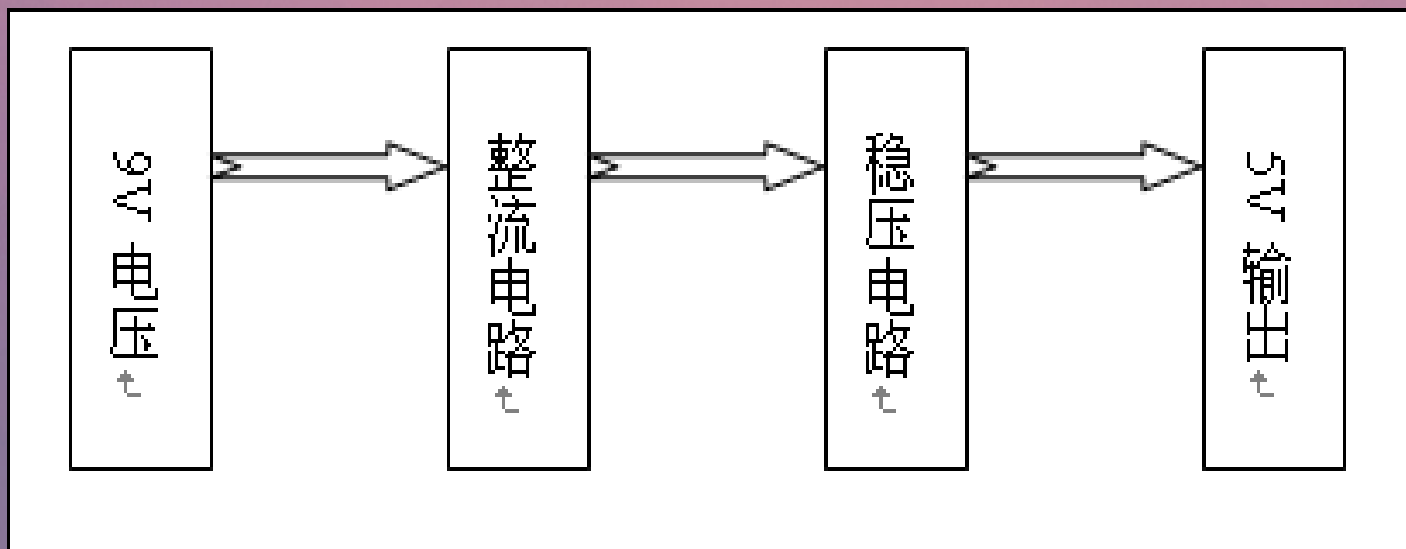


显示模块

- ◆ 采用 LCD 显示，用单片机可实现显示数据，并且可以实现单片机 I/O 口的并用，显示效果直观，明亮，调试不难，还省电。故采用 LCD 液晶屏显示。显示亮度和字体大小在演示时更具灵活，十分美观，令人赏心悦目。显示电路如图 6 所示。



◆ 由于电机在正常工作时对电源的干扰很大，如果只用一组电源时会影响单片机的正常工作，所以我们选用双电源供电。一组 5V 给单片机和控制电路供电， 另外一组 9V 给电机供电。以免影响控制部分电源的品质。考虑到如果 9V 电源直接驱动，那么使小车的速度太快，转弯不灵敏，造成小车来不及转弯就冲出跑道，则在电源处加了一个稳压管（7805），使电机在稳定的 5V 电压下工作。电路框图如图 7 所示，



硬件抗干扰的实现

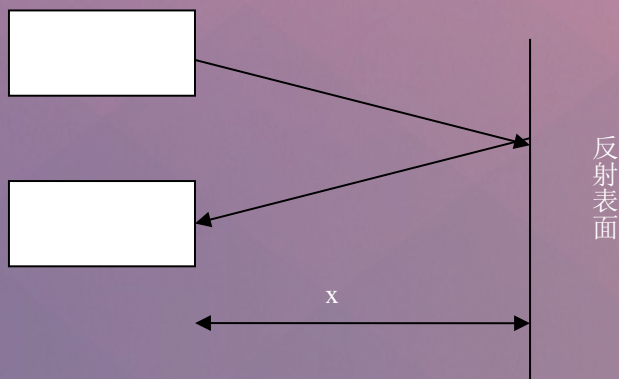
- ◆ 为了使系统硬件稳定可靠，我们采用了一系列的措施来减少干扰：
 - ◆ 1、 采用单片机与电机分离供电的方案；
 - ◆ 2、 采用地线、电源线加粗，做成 PCB 板的形式；
 - ◆ 3、 给电机加滤波电路；对于单片机和其它 IC 的闲置端在不改变系统逻辑的情况下接地或接电源；
 - ◆ 4、 印制板图布线时尽量减少回路环的面积，降低感应噪。

各功能模块的实现

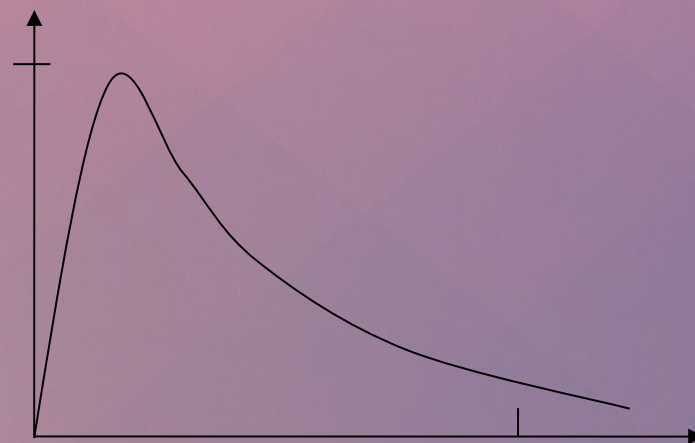
◆ 寻迹实现

◆ （1）黑带检测原理

- ◆ 利用光的反射原理，当光线照射在白纸上，反射量比较大，反之，照在黑色物体上，由于黑色对光的吸收，反射回去的量比较少，这样就可以判断黑带轨道的走向。由于各路传感器会对单片机产生一定的干扰，使信号发生错误。因此，采用一级射极输出方式对信号进行隔离，这样系统对信号的判断就比较准确。



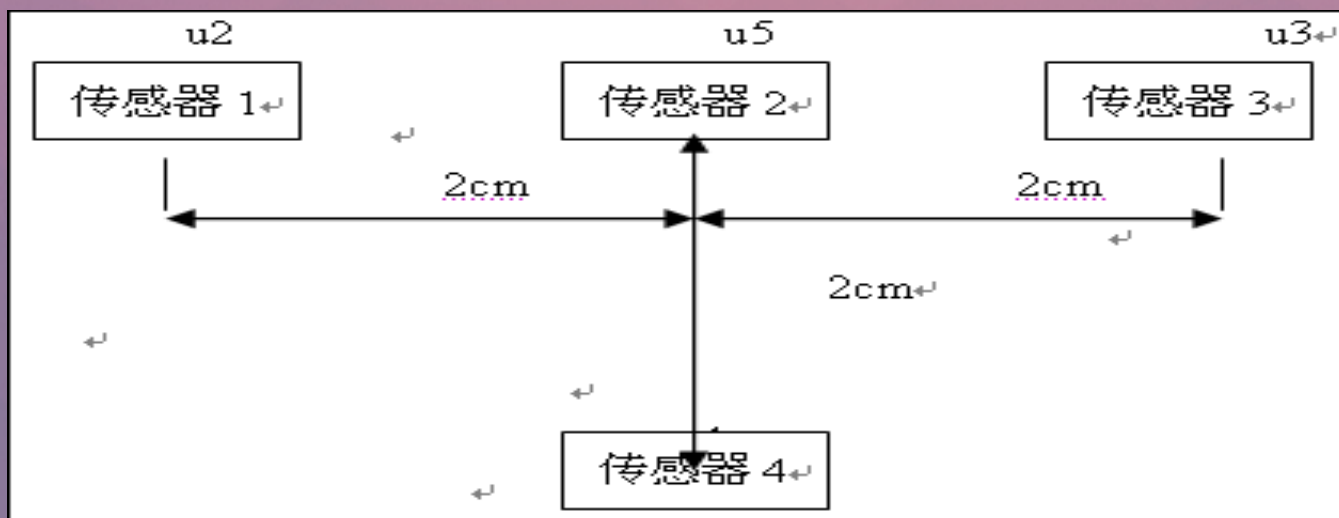
对管发射接收原理



光强度相应曲线 10

各功能模块的实现

- 传感器的使用数量应该尽量少以减少单片机的信号处理量，寻迹小车底部一共安装有 4 对反射式光电对管传感器，选用运算放大器 LM339，光电传感器检测到的信号经放大器放大整形送微处理器判断、运算、控制。1、3 传感器骑在黑线两侧用于检测是否跑偏，2、4 传感器在黑线范围内用于主要检测，以确保小车一直在黑线上运行，传感器 5 放在 3 的旁边，距它 2cm，与 1、2、3 并排，用于检测小车在返回时停在出发点。传感器状态决定下一步的运行方式。



主程序流程图

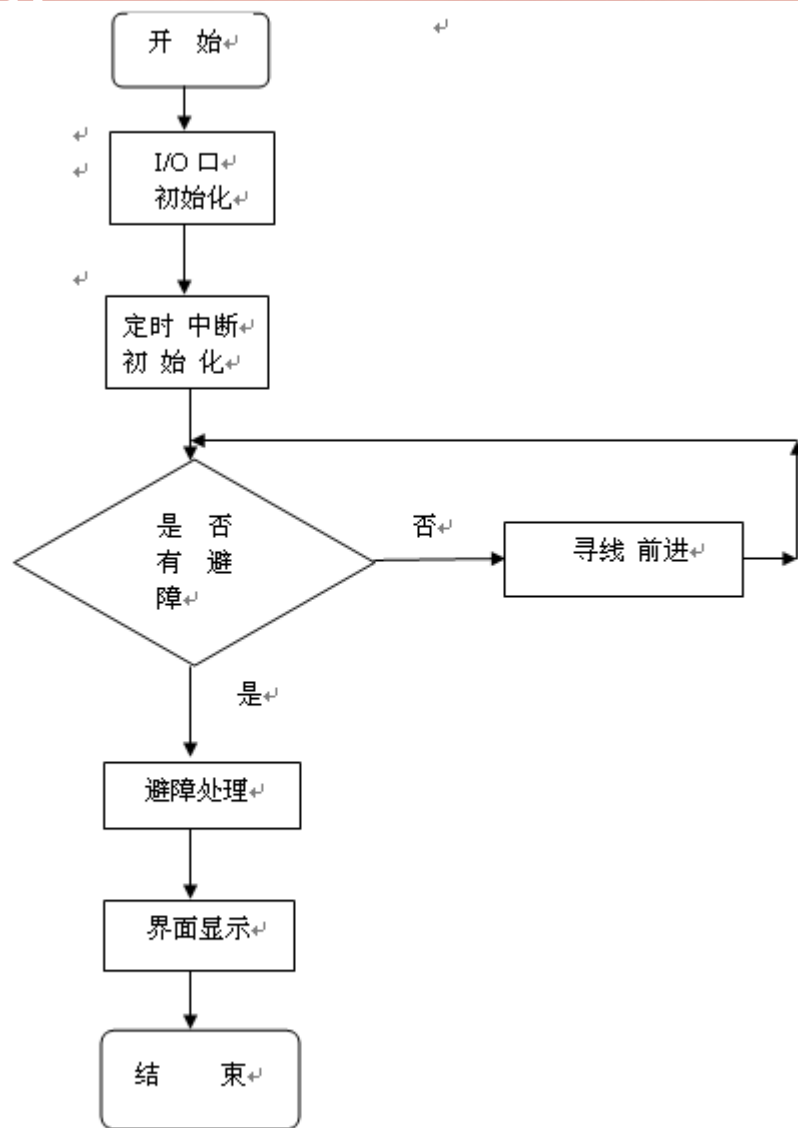


图9 主程序流程图

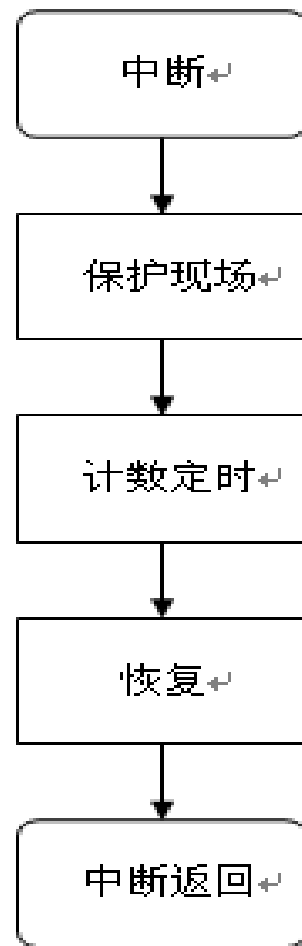


图 11 T0 定时子程序

部分程序

```
#include <avr/io.h>

#include <avr/interrupt.h>

#include <avr/signal.h>

#include <avrbit.h>

#include <avr/wdt.h>

typedef unsigned char uchar;

typedef unsigned int uint;

/*****对电机的一些参数设置*****/

#define ONWARD 0x1B // 前进

#define IN1 PB0 // 右轮 OC1A

#define IN2 PB5

#define IN3 PB3 // 左轮 OC1B

#define IN4 PB4

#define MaxSpeed 0x0000 // 速度参数选择

#define MidSpeed_10 0x0399 // 慢

#define MidSpeed_20 0x0333

#define MidSpeed_30 0x02cc

#define MidSpeed_40 0x0266

#define MidSpeed_50 0x01ff

#define MidSpeed_60 0x0199
```

```

◆ #define MidSpeed_70 0x0133
◆ #define MidSpeed_80 0x00cc
◆ #define MidSpeed_90 0x0066 // 快
◆ #define StopSpeed 0x03ff
◆ #define TurnSpeed_QQ MidSpeed_60 // 转弯快的 速度
◆ #define TurnSpeed_SS MidSpeed_30
◆ #define TurnSpeed_Q MidSpeed_60 // 转弯快的 速度
◆ #define TurnSpeed_S MidSpeed_30
◆ #define LeftAA TurnSpeed_QQ // 向左 大转弯
◆ #define LeftBB TurnSpeed_SS
◆ #define LeftA TurnSpeed_Q // 向左 小转弯
◆ #define LeftB TurnSpeed_S
◆ #define RightAA TurnSpeed_SS // 向右 大转弯
◆ #define RightBB TurnSpeed_QQ
◆ #define RightA TurnSpeed_S // 向右 小转弯
◆ #define RightB TurnSpeed_Q
◆ uint FlagBack = 0;

```

```

*   /*#define P_ENA PD_4
*
*   #define P_ENB PD_1*/
*
*   /***** 液晶参数设置 *****/
*
*   #define P_RS PC_0
*
*   #define P_RW PC_1
*
*   #define P_E PC_2
*
*   #define clear 0x01 //清除显示
*
*   #define home 0x02 //位址归位
*
*   #define dis_on 0x0f //打开显示
*
*   #define in_mode 0x06 //输入方式,地址增1,光标右移
*
*   uchar str2[6];
*
*   /***** 定时T0参数定义 *****/
*
*   #define TIME0 4000 //0.25ms * 4000 = 1s
*
*   #define INTNUM0 6 //定时TCNT0初始值
*
*   #define SECONDS 60 //T0 秒数 60秒
*
*   #define MINS 60 // 60分
*
*   uint Count0 = 0; //T0 1s 计数变量
*
*   uint Second = 0; //T0 秒 计数变量
*
*   uint Min = 0; // 分变量
*
*   uint Time[4]; // 秒 分
*
*   /***** 长延时函数 *****/
*
*   void DELAY( uint delaytime)
*   {
*       uint i,j;
*
*       for(j=0;i<=delaytime;i++)
*       {
*           for(j=0;j<=0x5ffe;j++);
*
*       }
*   }
*
*   }

```

```

*   /***** 液晶模块 *****/
*
*   void bit8_serial_input(unsigned int ix) //
*   {
*       unsigned int i,j,nc;
*
*       BitPort ab;
*
*       P_E=0;
*
*       ab.W=ix;
*
*       for(i=0;i<8;i++)
*       {
*           P_RW=ab.B.bit7;
*
*           ab.W<=1;
*
*           P_E=1;
*
*           for(j=0;j<5;j++)
*
*           nc=0;
*
*           P_E=0;
*       }
*   }
*
*   void instruct(unsigned int dat)
*   {
*       unsigned int ch;
*
*       P_RS=1;
*
*       ch = 0xf8;
*
*       bit8_serial_input(ch);
*
*       ch=dat&0xf0;
*
*       bit8_serial_input(ch);
*
*       ch=dat<<4;
*
*       bit8_serial_input(ch);
*
*       P_RS=0;
*   }
*
*   }

```

```

* void write_abyte(unsigned int dat)
* {
*     unsigned int ch;
*
*     P_RS=1;
*
*     ch=0xfa;
*
*     bit8_serial_input(ch);
*
*     ch=dat&0xf0;
*
*     bit8_serial_input(ch);
*
*     ch=dat<<4;
*
*     bit8_serial_input(ch);
*
*     P_RS=0;
*
* }
*
* void write_word(unsigned int w)
* {
*     unsigned int ch,cl;
*
*     ch=w>>8;
*
*     cl=w;
*
*     write_abyte(ch);
*
*     write_abyte(cl);
*
* }
*
* void dis_hz_str(unsigned int x, unsigned int y, unsigned char *p_hz)
* {
*
*     unsigned int loc;
*
*     unsigned char *p;
*
*     unsigned int xi;
*
*     unsigned int xline[5]={0,1,3,2,4};
*
*     x=xline[x];
*
*     instruct(home);
*
*     loc=((x<<3)+y-9) | 0x80;
*
*     instruct(loc);
*
*     p=p_hz;
*
*     xi=*p++;
*
*     xi<=8;

```

```

* while(xi)
* {
*     write_word(xi);
*
*     xi=*p++;
*
*     if(xi==0)
*
*         break;
*
*     xi<=8;
*
*     xi=*p++;
*
*     }
* }
*
* void dis_str(unsigned int x, unsigned int y, unsigned char *str)
* {
*
*     unsigned int loc;
*
*     unsigned char *p;
*
*     unsigned int xline[5]={0,1,3,2,4};
*
*     p=str;
*
*     x=xline[x];
*
*     instruct(home);
*
*     loc=(16*(x-1)+y-1);
*
*     if(loc&0x01)
*
*     {
*
*         loc/=2;
*
*         instruct(loc | 0x80);
*
*         write_abyte(' ');
*
*     }
*
*     else
*
*     {
*
*         loc/=2;
*
*         instruct(loc | 0x80);
*
*     }
*
*     while(*p)
*
*     {
*
*         write_abyte(*p++);
*
*     }
*
* }

```

```
♦ /*****液晶界面初始化
*****/
```

```
♦ void InitLcd ()
```

```
♦ {
```

```
♦     initial_screen();           // 初始化屏幕
```

```
♦     instruct(home);
```

```
♦     dis_hz_str(1,2,"非常");
```

```
♦     dis_hz_str(1,6,"组合");
```

```
♦     dis_str(1,7,"4+1");
```

```
♦
```

```
♦     dis_hz_str(2,2,"预祝奥运成功");
```

```
♦     dis_str(2,10,"2008.8.8");
```

```
♦     dis_hz_str(3,1,"开车时间");
```

```
♦     dis_str(3,10,"00: 00");
```

```
♦     dis_str(3,15,"S");
```

```
♦
```

```
♦     dis_str(3,15,"S");
```

```
♦     dis_hz_str(4,2,"简易寻迹蔽障小车");
```

```
♦     dis_str(4,10,"2008.8.8");
```

```
♦     dis_hz_str(5,1,"结束");
```

```
♦     dis_str(5,10,"00: 00");
```

```
♦     dis_hz_str(6,1,"");
```

```
♦     dis_str(6,10,"00: 00");
```

```
♦     dis_hz_str(7,1,"");
```

```
♦     dis_str(7,10,"00: 00");
```

```
♦     dis_hz_str(8,1,"");
```

```
♦     dis_str(8,10,"00: 00");
```

```
♦     dis_hz_str(9,1,"");
```

```
♦     dis_str(9,10,"00: 00");
```

```
♦ }
```

```
♦ /*****数据处理*****/
```

```
♦ void DisposeData()
```

```
♦ {
```

```
♦     Time[3] = Min / 10;
```

```
♦     Time[2] = Min % 10;           //分
```

```
♦     Time[1] = Second / 10;
```

```
♦     Time[0] = Second % 10;        //秒
```

```
♦ }
```

```
♦
```

```
♦
```

```
♦ /*****液晶显示数据*****/
```

```
♦ void DisTime ()
```

```
♦ {
```

```
♦
```

```
♦     itoa(Time[0]);
```

```
♦     dis_str(3,14,str2);
```

```
♦
```

```
♦
```

```
♦     itoa(Time[1]);
```

```
♦     dis_str(3,13,str2);
```

```
♦
```

```
♦
```

```
♦     itoa(Time[2]);
```

```
♦     dis_str(3,11,str2);
```

```
♦
```

```
♦
```

```
♦     itoa(Time[3]);
```

```
♦     dis_str(3,10,str2);
```

```
♦
```

```
♦
```

```
♦ }
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
♦
```

```
// 定时 TO

// 功能：秒表
//

//
*****
****//

void initial_T0()    // 初始化

{

    uint sreg;

    sreg = SREG;

    cli();

    TCCR0 = (1<<CS00);    // 选择系统时钟

    TCNT0 = INTNUM0;    // 计数 250 次 即 0.25ms 产生一次中断

    //TIMSK = (1<<TOIE0) | TIMSK;

    SREG = sreg;

}

/***** 定时器 1 初始化 *****/

void Timer1Init( )

{

    uint sreg;

    sreg = SREG;

    cli();

    TCCR1A = (1<<WGM11) | (1<<WGM10) | ( 1<<COM1A1) | ( 1<<COM1A0) | ( 1<<COM1B1) |
( 1<<COM1B0);

    TCCR1B = (1<<CS10);

    // 10 位 PWM,相位可调，向上计数匹配置位 1M / 2046 = 488 HZ

    SREG = sreg;

}
```

```
/* *****写 OCR1A 寄存器***** */

* void SetOutputComReg1A(uint tempocr)    //控制右轮

* {

*     uint sreg;

*     sreg = SREG;

*     cli();

*

*     OCR1A = tempocr;

*

*     SREG = sreg;

* }

* /***** 写 OCR1B 寄存器 *****/

* void SetOutputComReg1B(uint tempocr)    //控制左轮

* {

*     uint sreg;

*     sreg = SREG;

*     cli();

*

*     OCR1B = tempocr;

*

*     SREG = sreg;

* }

* /***** 向左大转弯 *****/

* void TurnLeftAA()

* {

*     SetOutputComReg1A(LeftAA);    //左轮减速，右轮加速，向左大转弯

*     SetOutputComReg1B(LeftBB);

*

*     // DELAY(1);

*

* }
```

```

♦ /***** 主函数开始 *****/
♦ int main(void)
♦ {

♦         OSCCAL=0xB2;           //1M 时钟源

♦
♦         DDRB = 0xff;
♦         DDRC = 0x0f;
♦         DDRD = 0x00;           // 低四位接对管传感器

♦         InitLcd ();

♦
♦         PORTB = (1<<IN1) | (0<<IN2) | (1<<IN3) | (0<<IN4);
♦         //PORTD = 0xff;

♦         initial_T0();           // 定时 T0 初始化
♦         Timer1Init();           // 定时 T1 初始化

♦         // SetOutputComReg1A( 0x0399);
♦         // SetOutputComReg1B(0x0066);           // 控制左轮
♦         //      TurnRightAA();
♦         //      TurnRightA();
♦         //      TurnLeftAA();
♦         //      TurnLeftA();
♦

```

```
♦ RunFor();

♦          TIMSK = (1<<TOIE0);
♦ sei();

♦          while(1)
♦          {
♦ if(FlagBack == 0)
♦          {

♦          PORTB = (1<<IN1) | (0<<IN2) | (1<<IN3) | (0<<IN4);
♦          Forward();
♦
♦
♦          }

♦          ZhangAi();
♦ DisposeData();
♦ DisTime ();
♦
♦          }
♦ return (0);
♦ }
```

```

//*****//
//          中断开始          //
//*****//
SIGNAL(SIG_OVERFLOW)    //SIGNAL 必须大写才行
{
    TCNT0 = INTNUM0;

    Count0++;

    if(Count0 == TIME0)
    {
        Count0 = 0;

        Second++;

        if(Second == SECONDS)    //60 秒
        {
            Second = 0;
            Min++;
            if(Min == MINS)    //60 分
            {
                Min = 0;

            }

        }

    }

}
}
```