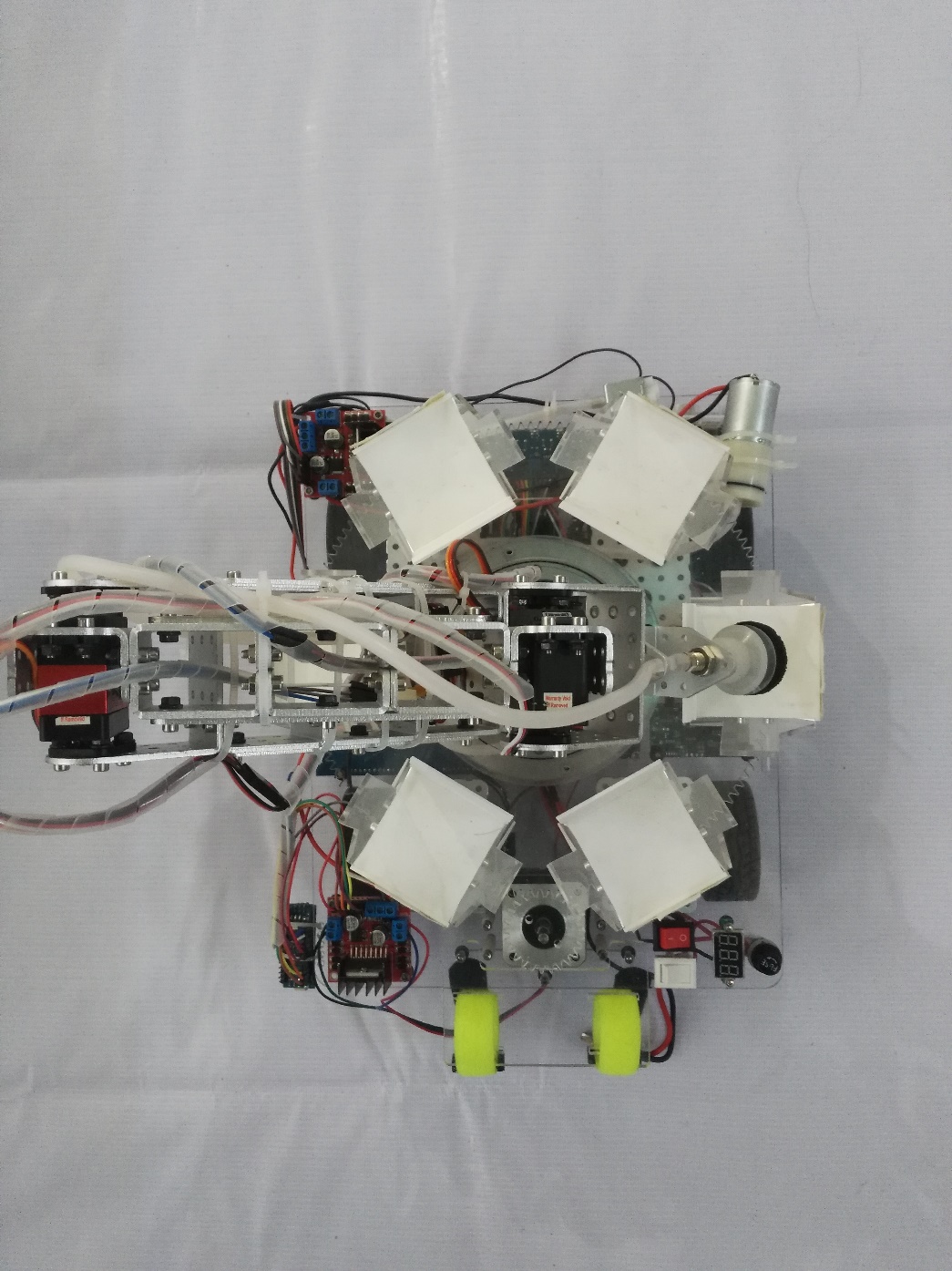
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **第四届浙江省大学生工程训练综合能力竞赛**  The 4th ZheJiang UndergraduateEngineering Training Integration Ability Competition | **电路设计方案**  Electronic Circuit Design Scheme | | 共 7 页 | 第 1 页 |
| 编 号 | （工作人员填写） |
| 1 | | **1、控制系统设计思路**  考虑到小车在比赛中需要完成巡线、上坡、扫码、装载物料、避障和放置物料等任务，我们将控制系统分为检测子系统、运算子系统、决策子系统和执行子系统，为了更好地实现功能，在一些环节中我们加入了闭环反馈控制。 我们控制方案的设计如下：  检测子系统负责读取和检测所有传感器的数值。包括读取七路巡线传感器数值、超声波距离传感器数值、扫码信息、轮子（电机）转动圈数等，这些信号是其他各个环节的基础。  运算子系统负责对一些由检测子系统检测到的信号进行处理。将信号转换成具体的数据，并对这些数据通过一定的算法进行处理。例如，对于从七路巡线传感器获得的七个数据（0或1），首先进行加权平均，再对加权平均后的数值进行PID运算，最后得到执行子系统需要的参数。  决策子系统根据检测到的信号作出相应的决策。例如，每检测到一条双侧黑线，相应的计数变量会加一，决策子系统根据这个计数变量的数值让程序进入相应的执行环节（例如，上坡、装载、避障等）。  执行子系统接受决策子系统的命令和运算子系统的数据，来控制具体器件的工作。包括控制舵机、直流电机、步进电机、气泵和电磁阀的工作。舵机由舵机驱动板驱动、直流电机由双路电机驱动板驱动、步进电机由L298N驱动、气泵和电磁阀也由L298N驱动。  整个系统的大部分控制都是开环的，也有一些闭环反馈的控制，比如巡线的PID控制、扫码时小车位置的控制等等。  我们选择 Arduino Mega2560 单片机作为主控板，Arduino Nano单片机作为下位机分担主控板的工作。主控板控制直流电机、舵机、气泵、电磁阀、七路巡线传感器、扫码器、超声波距离传感器等器件的工作以及向下位机发送命令；下位机接受主控板的命令并控制步进电机的工作。  整个系统依靠 7.5V 的锂电池直接供电，由于舵机额定电压低于 7.5V，故另通过一个降压模块实现降压功能间接给舵机供电。另采用三个电机驱动模块分别驱动直流电机、步进电机、电磁阀和气泵。电源正极串联一个 10A 保险丝防止短路。 | | |
| **2、器件选择及实施方案**   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 序号 | 元件 | 数量 | 序号 | 元件 | 数量 | 序号 | 元件 | 数量 | | 1 | 单片机（Mega） | 1 | 7 | 电机驱动 | 3 | 13 | 300度舵机 | 2 | | 2 | 单片机（Nano） | 1 | 8 | 巡线传感器 | 1 | 14 | 180度舵机 | 1 | | 3 | 超声波传感器 | 2 | 9 | 舵机控制板 | 1 | 15 | 气泵 | 1 | | 4 | 直流电机 | 4 | 10 | 稳压模块 | 1 | 16 | 电磁阀 | 3 | | 5 | 扫码模块 | 1 | 11 | 保险丝 | 1 | 17 | 电池 | 1 | | 6 | 步进电机 | 1 | 12 | 轴承 | 1 | 18 | 轮子 | 4 |   a) 单片机：由于我们对主控板的 IO 口数量要求较高，因此我们选用 ARDUINO 系列的 Mega2560 作为主控板，通过串口实现主控板和舵机控制板、扫码模块之间的通讯，通过三个数字口实现主控板和下位机Nano之间的通讯，以 C 语言作为编程语言。  b) 舵机：舵机采用两个300度大扭力舵机和一个探索者180度大舵机。  c) 步进电机：单步1.8度，精度很高。  d) HC-SR04 超声波模块：该超声波模块使用电压为 5V，静态电流小于 2mA，探测距离范围为 2~450cm，精度可达 0.3cm。  e)电机驱动：两个L298N 用来驱动步进电机和气泵，一个 WSDC2412D用于控制四个直流电机。  f)巡线传感器：七路红外传感器，可调每一路灵敏度。  g)气泵和电磁阀：工作电压都是6V，适当组合后可实现单管路吸气和放气。  h)扫码模块：主控板串口发送指令控制，扫码效率高。  **3、总结和体会**  控制部分的设计十分重要，需要达到稳、准、快三个要求。在具体实现过程中，出现过许多问题，比如器件之间的相互影响、控制板定时器的冲突等等。这些问题虽然不大，但查找起来却很麻烦，有时一个小小的问题就要花很长时间去发现和解决。不过，只要细心排查、持之以恒，问题总能得到较好的解决。  在我们遇到的问题中，除了程序上的问题，出问题最多的地方是器件间的相互影响，特别是其它器件对主控板的影响。比如，我们之前是用继电器控制电磁阀和气泵的工作，但发现气泵的启动会影响继电器与主控板相连的信号线上的电流突然增大，导致程序重启。这一点是我们在项目开始时没能预料到的。 | | |



学校名称：

装

订

线

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UZ}E0_H)D)ZBDPM3`ERCNI9 | 说 明  大箭头表示大负载的供电，小箭头表示小负载的供电和信号的传递与控制。以下是具体说明。  1.所有的传感器供电线、地线和信号线通过单片机扩展板连接在主控板 Mega2560 上，即超声波模块、七路巡线传感器、电机编码器和扫码模块。在信号线的连接上，超声波模块和七路巡线传感器的信号口与Mega板的数字口相连，电机编码器的信号口与Mega板的外部中断口相连，扫码模块与Mega板通过串口进行通讯。Nano板的供电线、地线和信号线也与Mega板直接相连，两者之间的通讯通过三个数字口进行。  2.气泵、电磁阀和步进电机需要较高的电压和较大的电流，由电源连接L298N驱动后供电。驱动气泵和电磁阀的L298N由Mega板控制；驱动步进电机的L298N由Nano板控制。  3.所有的舵机由舵机控制板供电和控制，舵机控制板则由电源经稳压模块降压、稳压后供电，并受Mega板串口命令控制。  4.四个直流电机（即电机1、2、3、4）由双H桥电机驱动模块驱动，双H桥电机驱动模块有两路驱动，一路驱动一边的两个轮子。Mega板给双H桥电机驱动提供控制信号，可实现电机的正转、反转和刹车功能。  5.7.5V电源与各器件间通过一个总开关和一个熔断器连接，若电流过大，保险丝烧断，电路断开，可防止器件烧坏。在Mega板供电支路上也串联了一个开关，这个开关的作用是可使舵机控制板单独上电，方便动作组的调节。 | | | |
|  | | 电路原理图 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第 3 页 |

装

订

线

学校名称：

|  |  |  |  |
| --- | --- | --- | --- |
| 小车程序流程路 | 说 明  小车的程序运行流程图如左图所示。具体描述如下：  小车以巡线为主体，当检测到第一条双侧黑线后，开始上坡，当检测到第一条单侧黑线后开始物料的装载，当检测到第二条双侧黑线后，开始避障或过栅格板或过减速带，当检测到第二条单侧黑线后，开始下料，当检测到全白后，前进一段距离后停止。上述流程中最主要的三个环节是巡线、定位和避障。  巡线方案：七路灰度传感器接收到灰度信号并返回0或1，通过对返回值的加权平均即可知道小车头部偏离黑线的程度。采用PID算法保证小车能够有效地调整前进方向并实现平滑转弯。  定位方案：为了使小车能够实现非常精确的定位，小车通过采集电机编码器产生的脉冲数进行定位。当小车判断到自己进入到上料区时，先让小车以巡线的方式前进一段距离，然后再退回一定的距离，使小车摆直。接下来小车每次前进一段固定的距离即可定位上料区每一列。在前进一段固定距离的过程中，小车仍保持巡线状态。  避障方案：小车在刚进避障区时会用超声波传感器测量自己与障碍的距离，若距离大于80厘米，则在避障时采用避障路线1，若距离小于80厘米且大于25厘米，则在避障时采用避障路线2，若距离小于25厘米，则在避障时采用避障路线3。小车在避障时的转弯方案是在下达左转命令后，通过采集电机编码器产生的脉冲数使左右两轮转过一定的圈数，调整好适当的圈数差，就能使小车转过固定的角度。 | | |
|  | 程序流程图 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第 4 页 |

装

订

线

学校名称：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| /\*读取七路灰度传感器数值进行加权计算、计数、设置信号标志变量\*/  void readsensor()  {  S1 = digitalRead(sensor1); S2 = digitalRead(sensor2); S3 = digitalRead(sensor3);  S4 = digitalRead(sensor4);S5 = digitalRead(sensor5);S6 = digitalRead(sensor6);  S7 = digitalRead(sensor7);  offest=((-15\*4\*S1)+(-15\*2\*S2)+(-15\*1\*S3)+(15\*1\*S5)+(15\*2\*S6)+(15\*4\*S7))/(S1+S2+S3+S4+S5+S6+S7);  if(S1==HIGH && S2==HIGH && S3==HIGH && S4==HIGH && S5==HIGH && S6==HIGH && S7==HIGH && avoidding==0){  crossroad++;  }  else if(S3==HIGH && S4==HIGH && S5==HIGH && S6==HIGH && S7==HIGH && cansingle==1 && avoidding==0)  { singleroad++;  offest=((-15\*3\*S1)+(-15\*2\*S2)+(-15\*1\*S3)+(15\*1\*S5))/(S1+S2+S3+S4+S5);  }  if(crossroad==3 && S1==LOW && S2==LOW && S3==LOW &&S4==LOW &&S5==LOW &&S6==LOW && S7==LOW && avoidding==1) {  jsd=2;  }  if(crossroad==3 && S1==HIGH && S2==HIGH && S3==HIGH &&S4==HIGH &&S5==HIGH &&S6==HIGH && S7==HIGH && avoidding==1) { jsd=1;  }  if(crossroad==-1 && S1==LOW && S2==LOW && S3==LOW &&S4==LOW &&S5==LOW &&S6==LOW && S7==LOW && avoidding==0) {  endding=1;  }  }  /\*PID\*/  float PID(float e,float kp,float ki,float kd)  {  static float e\_s=0,sum=0;//e\_s用于保存上一次的误差值，用于计算微分项。Sum用于计算累加和，计算积分项。  float r;  sum+=e;//计算积分累加和  r=kp\*e+ki\*sum+kd\*(e-e\_s);//从左至右分别是比例项、积分项、微分项  e\_s=e;//保存这一次的误差值用于下一次微分计算  return r;  }  /\*巡线\*/  void track()  {  value=PID(offest,P,I,D);  speedleft=constrain(prespeed1+value,0,255);  speedright=constrain(prespeed1-value,0,255);  go(speedleft,speedright);  } | 关键代码说明  左边是巡线的代码，采用PID算法。  函数readsensor()读取七路灰度传感器数值，加权平均后计算出偏移量offest、 对经过的双侧黑线（crossroad）和单侧黑线（singleroad）进行计数并改变一些信号标志变量的值。  函数 PID(float e,float kp,float ki,float kd)对偏移量e、P参数、I参数和D参数进行 PID运算。  函数track()将函数 PID()运算得到的数值用于左右轮差速的调节。  以下主函数即可实现巡线功能：  void loop()  {  readsensor();  track();  }  采用PID算法可以使车子在转弯过程中行驶得比较平滑。 | | | |
|  | | 关键代码说明 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第5 页 |

装

订

线

学校名称：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| /\*定位巡线\*/  void locatetrack(int defvalA){  attachInterrupt(3, encoder, CHANGE);  while(valA!=defvalA){  bmreadsensor();  if(S2==LOW && S3==HIGH )  {  foreward(60);  }else  if(S2==LOW && S3==LOW )  {  go(100,20);  }else  if(S2==HIGH && S3==HIGH )  {  go(20,100);  }  }  detachInterrupt(3);  valA=0;  return;  }  /\*编码器后退\*/  void bmback(int defvalA){  attachInterrupt(3, encoder, CHANGE);  while(valA!=defvalA){  backward(80,80);  }  detachInterrupt(3);  valA=0;  return;  }  /\*中断函数——编码器\*/  void encoder(){  valA++;  } | 关键代码说明  左边是定位的代码。通过主控板Mega板上的外部中断对电机编码器的脉冲信号进行计数，从而得到车子前进的距离。在这里，因为车子直线行驶，所以只检测一边电机编码器产生的中断信号。  函数 encoder()是中断函数，当编码器产生电平变化的脉冲时，程序进入中断函数， 计数变量valA自增一。  在函数locatetrack(int defvalA)中，while循环内的代码是为了让车子在前进的过程中能够矫正位置，这样保证车身不产生较大的歪斜。当valA增大到 defvalA时，循环结束，这样就能实现车子向前行驶一定的圈数。  函数bmback(int defvalA)实现车子向后退一定的圈数，原理与 locatetrack(int defvalA)相同，但没有车身矫直的功能。 | | | |
|  | | 关键代码说明 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第6 页 |

装

订

线

学校名称：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| char outputTable[8] = {0x08, 0x0a, 0x02, 0x06, 0x04, 0x05, 0x01, 0x09};  unsigned int stepTable[4] = {0, 406, 812, 1217};  char table[6] = {0, 1, 2, 3, -2, -1};  unsigned char i = 0;  unsigned int step = 0;  unsigned int targetStep = 0;  char position = 1;  char move = 0;  char dir = 0;  unsigned int speed = 2000; //目标速度900°/s  unsigned int OCR1A\_temp;  void setup()  {  pinMode(5, OUTPUT);  pinMode(6, OUTPUT); //5和6用于输出PWM  pinMode(2, INPUT);  pinMode(3, INPUT);  pinMode(4, INPUT); //2、3、4用于通信  DDRB |= 0x0f; //配置9、10、11、12用于输出步进电机信号  TCCR1A = 0x00;  TCCR1B = 0b00001010; //定时器1：8分频，CTC模式  TCCR2A = 0x00;  TCCR2B = 0b00000111; //定时器2：128分频，普通模式  OCR1A = 40000; //初始速度为45°/s  TIMSK2 |= 0b00000001; //启动定时器2溢出中断  TIMSK1 |= 0b00000010; //启动定时器1比较匹配中断  PORTB = outputTable[0];  }  void loop()  {  // put your main code here, to run repeatedly:  char target = 0;  analogWrite(5, 150);  analogWrite(6, 150); //维持转盘上力矩  if (target = GetTarget())  {move = table[(target - position + 6) % 6];  if (move)  dir = move > 0 ? 1 : -1; //判断方向 | else  return;  targetStep = stepTable[move > 0 ? move : -move]; //计算步数  analogWrite(5, 255);  analogWrite(6, 255);  OCR1A = 40000; //初始速度为45°/s  OCR1A\_temp = 40000;  TIMSK1 |= 0b00000010; //启动中断  TIMSK2 |= 0b00000001;  while (TIMSK1 & 0b00000010) ; //等待此次旋转结束  position = target;  delay(500); //等待转盘停下来  }  }  ISR(TIMER1\_COMPA\_vect)  { PORTB = outputTable[(i += dir) % 8];  if (++step >= targetStep) //目标步数达到后  {  step = 0; //步数清零  dir = 0; //方向回正  TIMSK1 &= 0b11111101; //停止中断  }  OCR1A = OCR1A\_temp;  }  ISR(TIMER2\_OVF\_vect)  { OCR1A\_temp -= (OCR1A\_temp - speed) >> 3;  if (!((OCR1A\_temp - speed) >> 3))  TIMSK2 &= 0b11111110;  }  unsigned char GetTarget()  {int a = digitalRead(2);  int b = digitalRead(3);  int c = digitalRead(4);  int a1 = digitalRead(2);  int b1 = digitalRead(3);  int c1 = digitalRead(4);  if (a == a1 && b == b1 && c == c1)  return a \* 1 + b \* 2 + c \* 4;  else  return 0;} | 关键代码说明  该程序使用了单片机Nano中的计时器中断1和2。定时中断2为普通模式，匀速执行，用于计算步进电机的速度，从而实现S型加速以免失步。定时中断1使用CTC模式，通过不断改写OCR1A改变其执行频率。每当定时器1发生中断时向步进电机发送一个脉冲，以此驱动其运行。该程序使用八拍的控制方式驱动步进电机，每个脉冲对应0.9°。经过齿轮减速后精度更高。  Nano板使用3个IO口与上位机进行简单的并行通信以获取需要旋转至的位置。当收到需要旋转至的位置时，Nano板便会计算其相对当前所在位置的距离并算出应该顺时针还是逆时针转动，选择最短的路径驱动步进电机将转盘转至需要的地方。当步进电机未旋转时仍向其通以50%占空比的电流，保持转盘上的扭矩，防止转盘因为外力而转动。  setup()函数用于初始化输入输出引脚和定时器配置。  loop()函数用于循环读取上位机发送的数据。当收到有效数据时便计算出方向和步数，设置好相应变量后启动两个中断，之后便等待中断关闭——步进电机转动完毕——才进行下一次读取。  ISR(TIMER1\_COMPA\_vect)函数为定时器1比较匹配中断函数，用于向步进电机发送脉冲。每次发送后都会检查步数是否已到达目标值，并且更新OCR1A以调整中断发生频率——即步进电机转速。  ISR(TIMER2\_OVF\_vect)函数为定时器2溢出中断函数，用于计算步进电机的加速曲线。通过缓冲值temp改变OCR1A的值以改变电机转速。  getTarget()函数用于从上位机读取数据，并设置了一定防抖措施以增加容错度。 | | | |
|  | | | 关键代码说明 | 比 例 |  |
| **第四届浙江省大学生工程训练综合能力竞赛** | 共 页 | 第7 页 |

装

订

线

学校名称：