中国工程机器人大赛暨国际公开赛（RoboWork）

工程创新项目工程创新设计赛

**（机器人赛车项目对抗赛）**

技术报告

参赛学校： 北华大学

队伍名称： Mr.轮子

参赛队员： 侯小虎 王梦龙 邹鹏涛

带队教师： 董旭斌（18543275212）

填写日期： 2019 年 4 月

关于技术报告使用授权的说明

本人完全了解2019中国工程机器人大赛暨国际公开赛（RoboWork）关于保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人和比赛组委会共同所有，比赛组委会可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛机器人的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：

带队教师签名：

日 期：

# 目 录

目录

[目 录 3](#_Toc6673108)

[0　引言/综述 3](#_Toc6673109)

[1 系统整体设计 3](#_Toc6673110)

[2 机械结构设计 4](#_Toc6673111)

[2.1 车模选型 4](#_Toc6673112)

[2.2 器件的安装 4](#_Toc6673113)

[2.2.1 摄像头的安装 4](#_Toc6673114)

[2.2.2 编码器的安装 5](#_Toc6673115)

[2.2.3 超声波的安装 5](#_Toc6673116)

[2.2.4 红外对管模块的安装 6](#_Toc6673117)

[2.3 前轮倾角的调节 6](#_Toc6673118)

[2.4 整车的形状 7](#_Toc6673119)

[3 硬件设计 7](#_Toc6673120)

[3.1 控制部分 7](#_Toc6673121)

[3.2 检测部分 8](#_Toc6673122)

[3.2.1 摄像头 8](#_Toc6673123)

[3.2.2 编码器 8](#_Toc6673124)

[3.2.3 超声波 8](#_Toc6673125)

[3.3 执行部分 8](#_Toc6673126)

[3.3.1 电机 8](#_Toc6673127)

[3.3.2 舵机 8](#_Toc6673128)

[3.3.3 指示灯 8](#_Toc6673129)

[3.4 电源部分 9](#_Toc6673130)

[3.4.1、电源部分 9](#_Toc6673131)

[3.4.2 5V电源部分 9](#_Toc6673132)

[3.4.3 3V3电源部分 9](#_Toc6673133)

[3.4.4 其余部分 9](#_Toc6673134)

[4 软件设计 9](#_Toc6673135)

[4.1 主程序框架 9](#_Toc6673136)

[4.2 道路的识别 10](#_Toc6673137)

[4.2.1 道路的提取 10](#_Toc6673138)

[4.2.2 斑马线与菱形的识别 10](#_Toc6673139)

[4.3 速度与转向控制 11](#_Toc6673140)

[4.3.1 速度的控制 11](#_Toc6673141)

[4.3.2舵机的控制 11](#_Toc6673142)

[4.3.3 换道的实现 11](#_Toc6673143)

[4.3 核心代码 12](#_Toc6673144)

[5 系统开发与调试 15](#_Toc6673145)

[5.1 中线的识别 15](#_Toc6673146)

[5.2 转向的实现 16](#_Toc6673147)

[5.3 速度的设定 16](#_Toc6673148)

[5.4 信号灯的实现 16](#_Toc6673149)

[6 结论 16](#_Toc6673150)

[参考文献 18](#_Toc6673151)

[附录 19](#_Toc6673152)

**北华大学Mr.轮子队技术报告[[1]](#footnote-1)**

**侯小虎1，王梦龙2，邹鹏涛3**

(1.北华大学，吉林省吉林市1322013; 2.北华大学，吉林省吉林市 132013)

**摘要**

该机器人赛车是以MK60DN512芯片为主控单元的能在行驶过程中脱离人为干预的自主巡线赛车。该赛车通过灰度摄像头来检测赛道情况，并通过求取最大类间方差的方法来求出动态阈值，并以此阈值来进行图像的二值化，从而提取出赛道的基本信息。通过对赛道信息的提取来控制舵机进而控制转向。通过编码器反馈出实时速度与行驶的距离。使用的PID算法调节电机的转速，再加上提取到的赛道信息能够控制该机器人赛车在不同赛道区域的速度，形成闭环控制，能够使其达到各个区域的最快速度并保持稳定性。利用红外与超声波模块在换道时进行环境检测，并做出相应的动作从而避免交通事故的发生。在按键调节、OLED显示、上位机实时检测等情况下进行了大量的测试，验证了硬件电路与控制系统的可靠性。证明了该机器人赛车的制作方案可行。

**关键词：**最大类间方差；动态阈值；PID；环境检测；

关键词是反映技术报告主题内容的名词，是供检索使用的。中英文摘要均要有关键词，关键词一般为3-5个，注意词汇专业性，使用代表性的专业词汇，各个词间用“；”隔开。关键词排在摘要正文部分下方。中文摘要及关键词采用小四号宋体。

Technical Report of Mr. Roller Team of Beihua University

HOU Xiaohu1，WANG Menglong2，ZOU Pengtao3

*（*1. Beihua University, Jilin City, Jilin Province 2.Beihua University, Jilin City, Jilin Province*)*

**Abstract：**

The robot racing car is an autonomous circuit racing car with MK60DN512 chip as the main control unit, which can break away from human intervention in the course of driving. The racing car detects the track condition by gray camera, and obtains the dynamic threshold by calculating the maximum variance between classes, and binarizes the image with this threshold, so as to extract the basic information of the track. The steering gear is controlled by extracting track information. Real-time speed and driving distance are fed back by encoder. The speed of the motor is adjusted by using the PID algorithm, and the extracted track information can control the speed of the robot racing car in different track areas, forming a closed-loop control, which can make it achieve the fastest speed in each area and maintain stability. The infrared and ultrasonic modules are used to detect the environment when changing lanes, and corresponding actions are taken to avoid traffic accidents. In the case of key adjustment, OLED display and real-time detection of PC, a lot of tests have been carried out to verify the reliability of hardware circuit and control system. It proves that the design of the robot racing car is feasible.

**Key words：**Maximum inter-class variance；Dynamic threshold；PID；Environmental detection

# 0　引言/综述

“中国工程机器人大赛暨国际公开赛”作为一个公益性、非营利赛事，目前由教育部高等学校创新方法教学指导委员会、国际工程机器人联盟等机构共同主办，从2011年发起设立，经过几年的发展，已经形成搬运工程、竞技工程、竞速工程、生物医学工程等面向工程应用、突出创新实践、在国内有一定影响力的机器人科技竞赛活动。

本次参与的机器人赛车项目是模拟日常行车，其行驶的场地与规则都与非常贴近现实生活。该项目也紧随着时代的发展，科技的进步。以比赛的形式让我们接触到无人驾驶（自动驾驶），即通过相关控制系统实现无人驾驶的智能汽车，能够根据自身的相关检测达到不同的驾驶等级并持续发展到能够达到完全自动驾驶（SAE的定义，包含在2016年更新的SAE J3016 - 201609规范中，其将自动驾驶分为0 - 5 共6个等级；NHTSA的定义，包含在其2013年发布的《Preliminary Statement of Policy Concerning Automated Vehicles》中，将自动驾驶分为0 - 4 共五个等级）。在技术成熟时无人驾驶的错误率要远远低于人类的驾驶。但就目前来看自动驾驶还远远不能达到完全自动驾驶，其安全性也低于现在的人类驾驶，也不能处理一些突发事故。

综上所述，此次比赛能将真实道路上的模拟到赛道上，同时确保安全性是第一位，比赛规定也有明确要求是按照现行交通法规执行计分。其安全性是第一位，功能性次之，速度则是最后一位（行驶距离→扣分情况→所用时间）。在此看来此次比赛不单单是竞速类的比赛。更是对自动驾驶安全性，稳定性的考验。对这制作者安全意识的培养。

# 系统整体设计

机器人赛车系统主要由检测系统、控制系统、执行系统组成。其检测系统是由总钻风MT9V032摄像头、超声波模块、红外对管和编码器组成；控制系统MK60DN512，电机驱动组成；执行系统由电机、舵机以及灯光指示组成。由检测系统来采集道路信息和周围的车辆信息并反馈给以MK60DN512芯片为核心的控制系统，由控制系统对采集到的信息进行汇总处理并将处理得到的信息传递给执行系统；执行系统由电机来控制速度，由舵机来控制转向以及换道，由灯光来显示机器人赛车的动作；系统整体框图如下：



系统整体框图

# 机械结构设计

机械结构设计的好坏直接影响到整体以及后期系统的调试，硬件决定车模的速度上限，因此搭建一个好的车模的机械结构至关重要。

## 2.1 车模选型

在经过了几次的讨论分析之后我们从自主设计、模块搭建与车模改装中选择了车模改装。相对于前两种车模改装能够选择合适的符合赛事要求的车模并在其车模上添加相关的硬件电路。最终决定使用大学生智能汽车竞赛B类车模，符合所使用的舵机的规格、安装。另外此车模的一体性更强其误差更小，防撞效果更强。车模上的硬件结构部分大多都是标准件易于改装、维护、修理和更改器件。

## 2.2 器件的安装

每个器件的安装位置都会影响到其整体的信息采集，所以在每个模块的安装时都不能影响到其他检测模块对赛道信息的检测，每个器件的安装都要流出一定的余量，一方面便于管理和检测，另一方面便于更换与维修。

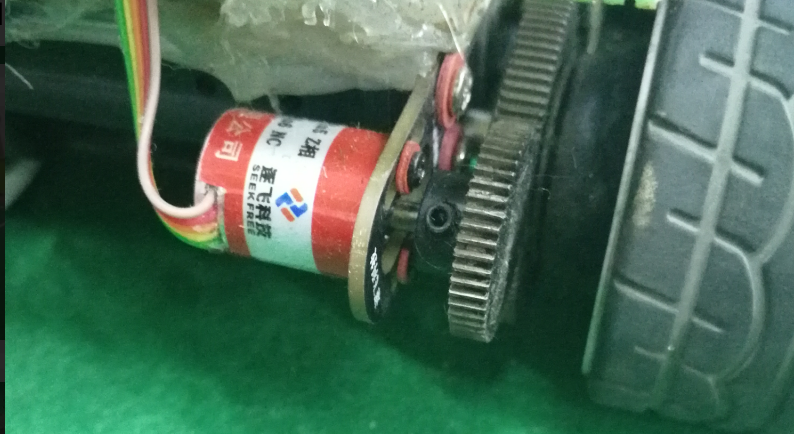
### 2.2.1 摄像头的安装

摄像头的安装位置会直接影响到对道路信息的采集，前瞻会更远，高度越高采集到的信息就越多，但相对而言就会加大摄像头的扫描起始位置，会有一定区域扫描不到且会增加一定的赛道之外的干扰信息，对赛道阈值的处理会造成很大的影响。在同样的位置摄像头的倾角越小但采集到的数据就越准确，同时看到的赛道距离就越小，无法的到更多的信息，倾角越大，采集到的数据就更多，但信息就会有所失真。综上原因所述，在实际中测试各种高度的效果，将摄像头安装在高的位置，同时增大摄像头的倾角。将摄像头安装在车模的后方，得到更多的赛道信息，尽量减小盲区。如下图所示。



### 2.2.2 编码器的安装

编码器采用逐飞科技专业为智能车比赛提供的三线编码器，即A相、B相、Z相，即可以提供速度信息，又可以得知电机速度的转向，其编码器转一圈可输出1024个脉冲，使得车模不会出现丢步的现象，使所测量出的车模速度与路程信息更加的精准。其产品硬件特点，体积小、重量轻、精度高、抗干扰能力强、可在多灰尘和潮湿环境下工作、有着稳定的工作性能。其轴承部件具有超高扭矩、超高转速、超顺滑、超静音、质量高、精度高、不发烫、超长寿命等等。比赛中，我们只是用了A、B相，就可以提取车模的路程和速度信息，同时得知，将编码器安装在车模后方的电机支架上，相当于直接安装在车模上这样就提高了编码器的稳定性，并且通过标准件齿轮的啮合可以直接与后轮的齿轮轴相连，这样就能准确记录速度与路程，如下图所示。



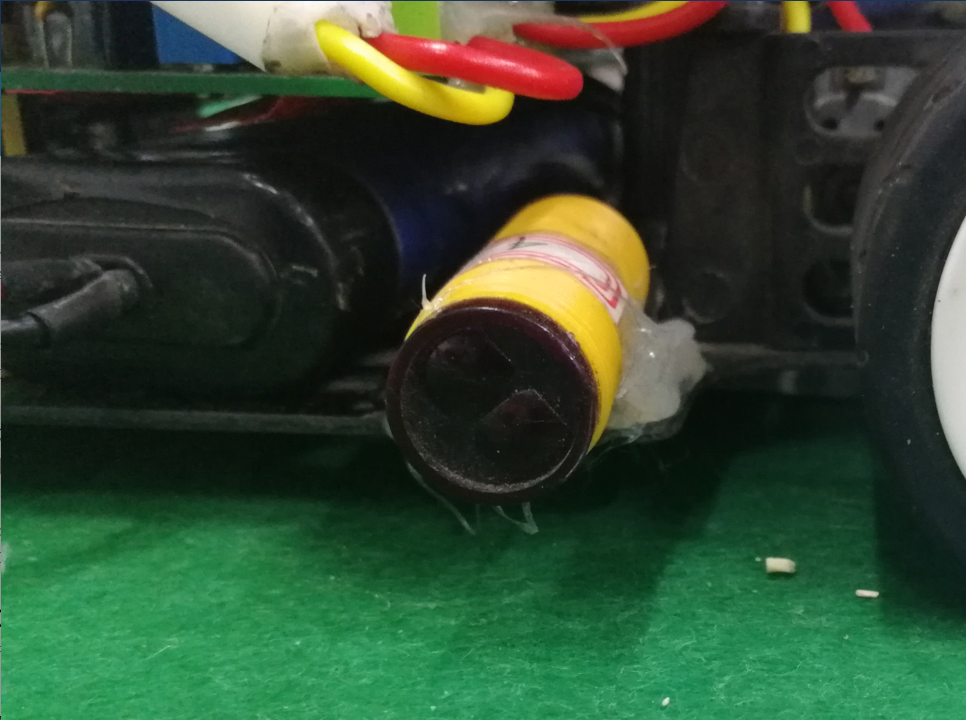
### 2.2.3 超声波的安装

超声波采用Y401型号的，其优点为所测出的距离在稳定情况下达到2m，已经能够达到比赛所需使用的距离，且其前后两次检测所需发送指令间隔时间很短，能够及时的反馈得出赛道上其它车辆以及障碍物的信息，将编码器安装在车头部位并且将超声波上扬一定的角度，使其与波束角相抵消，消除超声波检测到地面的可能。将超声波放置在车头部能够检测前面车辆防止撞车。并且能够检测较远距离的车辆，更早的得知赛道的信息并且更迅速的作出相应的判断。如下图所示。



### 2.2.4 红外对管模块的安装

红外对管安装在车底盘的四周，与底盘平行且与车的正前方向有一定的角度进行安装，可以有效的检测周围的车辆。尤其是在另一车道的时候与车头的超声波检测共同工作。如下图所示。

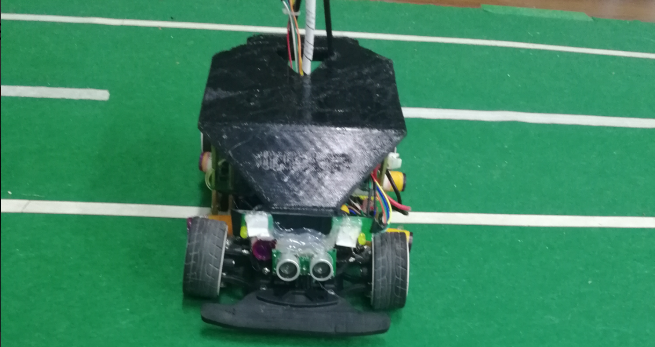


## 2.3 前轮倾角的调节

为了尽可能降低转向舵机负载，对前轮的安装角度，即前轮定位进行了调整。前轮定位的作用是保障汽车直线行驶的稳定性，转向轻便和减少轮胎的磨损。前轮是转向轮，它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束等 4 个因素决定，反映了转向轮、主销和前轴等三者在车架上的位置关系。所以我们选择前轮内倾角的结构。如下图所示。



## 2.4 整车的形状

# 硬件设计

## 3.1 控制部分

机器人赛车的主控采用的是MK60DN512,32位的单片机。主频晶振为50MHZ，内核频率为100MHZ，且能稳定的超频到180MHZ。在处理速度方面够足够。还有过压保护

## 3.2 检测部分

### 3.2.1 摄像头

摄像头采用的是逐飞科技的MT9V032的CMOS摄像头；此款摄像头采集宽度足够，能够设定采集的范围。每帧的图像采集只需3.3ms能够保障及时的更新赛道信息。工作在3.3V且功率小于250mw。

### 3.2.2 编码器

编码器采用的也是逐飞科技的1024线双相增量式mini编码器，能工作在3.3V~5V。能够工作在10000转的转速下不会丢步；其重量只有12克左右对电机的功率影响很小。

### 3.2.3 超声波

超声波选择的是US-100超声波测距模块。有一定波束角，能够检测正前方一个范围内的物体，且有串口模式方便检测。

# 3.3 执行部分

### 3.3.1 电机

电机的选择是RS-540马达，工作在5.4V~9V的电压范围内，其额定电压为7.2V；空载转速为15000r±10%，最大转速为17300r±10%。空载电流2.4A,最大电流为11.6A。最大扭矩为24.8mN.M，最大输出功率为51.49W。在较小的电压下其转矩不能使小车启动，引此要大于一定得电压值才能使机器人赛车启动行驶。

### 3.3.2 舵机

舵机选择的是JX6221数码舵机，舵机的转角为120°，扭矩为20KG。最大脉宽为900~2100us；在7.4V的工作电压下速度为0.1sec/60°，扭力为28Kg.cm正常工作电流为700mA，堵转电流为2.8A。死区为1u，反应速度为0.1s，输出安全稳定；在电压允许的条件下舵机越大其反应速度越快。

### 3.3.3 指示灯

指示灯选择的时LED小灯，因芯片的带载能力有限所以将灯直接连接在5V的电源上，将引脚口接在mosfet上，用以控制小灯的亮灭。

## 3.4 电源部分

由于电源选择的是7.4V的额定电源，但相关的主控芯片、摄像头、编码器、电机、舵机等的工作电压都是不尽相同的，整体上是7.2V、5V、3.3V同时存在并使用。因此对电源的管理是比较复杂的。

### 3.4.1电源部分

电源选择的是镍镉电池，电池的额定电压是7.2V、2000MA；充满电的情况下是8.5V。

### 3.4.2 5V电源部分

5V电源部分原采用7805芯片对7.2V的电源来进行5V的降压与稳压但在测试过程中发现7805的电流最大为1.5A，能够稳定工作的电流最大为800mA左右，虽能够供整个系统工作但其发热量大。因此改成DCDC电源模块；

### 3.4.3 3V3电源部分

3V3的电源部分原本采用1117稳压模块；主要供给处理器和摄像头但是和7805一样发热量很大，因此也改成了DCDC电源模块；能够稳定供电且发热量小。

### 3.4.4 其余部分

电机直接经过驱动板与电源相连，舵机的也直接与电源相连，控制信号由主控芯片控制。

给出作品的各部分器件的（如控制器，电源，电机，传感器）技术参数、选型、硬件电路设计及各部分功能。要着重体现创新性，突出自己对作品各部分硬件电路及各部分器件之间硬件电路的设计和改良。

# 软件设计

## 主程序框架

程序框架如图所示



## 道路的识别

道路的识别至关重要，也是相对复杂的，既要准确的识别到道路的信息去除相关噪点，又要从采集到的图像上提取出赛道中线，识别菱形、斑马线、弯道、直道、内道、外道、换道区与非换道区等，同时还需要检测周围的车辆。根据检测到的信息使小车做出相应的动作。

### 4.2.1 道路的提取

在道路识别中由于摄像头采集到的是灰度值，所以要对采集到的图像进行二值化，使图像显示成黑和白，使存储图像的二维数组中的数变为0x00或0xff此时的图像就是黑白的图像其判断也简单许多，在此时就需要用到“最大类间方差法（大津法）”来求取阈值进行图像的黑白分割。在处理到的图像中除了白线外都是黑色，能够简化赛道的信息。但对阈值的计算消耗时间太长因此要对此算法进行简化，经简化后使其每一场图像计算阈值所花费时间小于10ms。由于使用的摄像头每3ms可完成一次图像的采集，所以摄像头前后两场图像不会发生大的突变，遵循不突变的原则，从每场图像的固定行数开始从下往上进行扫描左右白色边界线，取扫描的边界点的坐标值作为上一行的基准值，进行左右两边边界扫描，在根据左右边界值提取赛道的中线。这样扫描可在很大程度上减小摄像头对道路的误判。

### 4.2.2 斑马线与菱形的识别

在处理后的图像中只有黑白像素点，而且只有道路是白色的且有一定的宽度因此只需提取出道路边线的特点即可，在识别到当前的道路后从左边界开始向右边界扫描，由于赛道内部只有菱形是可以扫描到两条白线的所以在线路内部中扫描到两次跳变沿时便可认为是菱形。

同理，在斑马线时的白线是最多的，因此检测到的黑白跳变沿次数大于正常道路的黑白跳变沿次数时即可认为时斑马线。

## 4.3 速度与转向控制

### 4.3.1 速度的控制

速度的控制选择使用的是棒棒法和PID算法结合进行优化控制电机的转速，棒棒法在直道上的应用使得车模的运动速度在直道以及启动的路段得到了很大的提高，简述棒棒法控制思想：当车模速度低于给定的目标值时，车模以最大的速度行驶，当高于车模的目标值时，则用PID算法来进行速度闭环控制。根据在道路中识别到的相关信息不同路段的提示来设置不同速度的目标值，通过PID算法来实现对PWM的控制，进而控制电机的转速即运动速度。其PID的控制原理图如下：



经测试发现同样的速度在弯道与直道是完全不一样的因此不能采用开环调整速度值。在测试的过程中发现入弯时如果速度过快就会因转向的原因发生甩尾的动作。所以采取电机闭环调速的办法。对于直道、内道弯道、外道弯道所采取的速度方案不同，直道相应的是以电机的最大转速尽可能快的跑，两个弯道则是测试出最大的入弯速度，即保证车模在入弯以及弯道内不发生甩尾、打滑、冲出赛道的现象。合理的安排不同路段的电机的速度控制，可以在一部分程度提高车模的行驶速度。

### 4.3.2舵机的控制

舵机的控制也是用PID控制PWM占空比大小进行舵机所打的固定角度控制车模的运动方向，其中偏差值是依据图像处理算法得到的赛道的实际中线与理想中线的差值，经过PID计算来对舵机的角度进行控制。

### 4.3.3 换道的实现

换道的依据，正常生活中的驾驶经验，去观看所换入的车道是否有车辆行驶，在进行相应的减速，同时打方向盘，等换到成功后进行方向盘打正，使得换道成功。则在车模上的运用体现，先是根据摄像头采集进来的图像信息，经过图像算法的处理，车模识别进入换道区，同时也根据信息得知使车模向右，还是向左换道。在根据摄像头提取出来的中线信息，使得舵机的打角的目标值相应进行一定程度的偏移，实现车模的换道，在转向的同时进行边界线的识别，待识别到左右白色边界线时候，舵机开始根据白色边界线进行纠正车模的方向，使得车身调直，进入正常行驶模式。换道区如果遇见换道区的后方、前方有车辆在行驶，则车模会根据红外模块的信息来对车模实现相应的减速，待等到其它车辆过后，再开始进行换道，进入正常换道模式，待到换道成功，进入正常行驶模式。

## 核心代码

把赛道路段分为直行、弯道、换道区路段，不同路段实行不同的速度进行控制，起点直道采用棒棒法和PID算法进行结合控棒棒法控制，其中棒棒法可以对速度进行很大优化，使得车模的提速更快。弯道内使用PID对车模速度进行控制，使速度在保证车模不出现打滑的同时，速度尽可能快的驶过弯道，根据内道以及外道的曲率计算和测试得出弯道的最大和稳定行驶速度。舵机的运用PID算法根据各部分提取出来的赛道信息同时使用加权平均法计算出赛道中线作为舵机的反馈进行车模方向的控制。全部行程舵机都采用PD控制，消除舵机的滞后现象同时控制车模的摇头。

if(Straight\_Road\_Select\_Flag ==0 && Zhi\_Flag\_1 ==1) //正常直道区域

{

Zhi\_Xing\_Over =0;

for(i =75;i >15;i-=5) //查看并提取8行元素 到19

{

for(j =50;j <90;j++)

{

if( (image[i][j] >=image\_threshold) && (image[i][j+1] >=image\_threshold) )

{Find\_White\_Range\_Flag ++; break;}

}

if(Find\_White\_Range\_Flag >=1) break; //可注释

}

if(Find\_White\_Range\_Flag >=1)

{Straight\_Road\_Select\_Flag =0; } //中间有白区

else

{

if(Zebra\_Crossing\_Flag\_Count ==0) //第一圈

{Straight\_Road\_Select\_Flag =1 ;Road\_Turn\_Right\_Flag =1; Road\_Turn\_Left\_Flag =1;}

else if(Zebra\_Crossing\_Flag\_Count ==1) //第二圈

{

if(Road\_Justice\_Flag ==0) //处于内道

{Straight\_Road\_Select\_Flag =1 ;Road\_Turn\_Right\_Flag =3; Road\_Turn\_Left\_Flag =2;}

else //处于外道

{Straight\_Road\_Select\_Flag =1 ;Road\_Turn\_Right\_Flag =2; Road\_Turn\_Left\_Flag =3;}

}

else

{Straight\_Road\_Select\_Flag =1 ;Road\_Turn\_Right\_Flag ++; Road\_Turn\_Left\_Flag ++;}

}

White\_View =Find\_White\_Range\_Flag;

Find\_White\_Range\_Flag =0; //标志位清零

}

//增加直道判断

if(Straight\_Road\_Select\_Flag ==1 )

{

for(j =50;j <90;j++)

{

if((image[79][j] >=image\_threshold) && (image[79][j+3] >=image\_threshold)) //应隔2个像素点判断

{

Find\_White\_Range\_Flag ++;

break;

}

}

if(Find\_White\_Range\_Flag ==1)

{ //{Zhi\_Xing\_Over =1;}

count1 ++;

if(count1 ==4) {Zhi\_Xing\_Over =1;} //防止突变点 大约4场图像时间

if(count1 >20) count1 =20;

}

else

{count1 =0;}

// if((Huan\_Dao\_Ok ==0)) //可增加huandao\_ok标志位 判断换道完成之后是否加速

// {Zhi\_Xing\_Over =0;}

White\_View =Find\_White\_Range\_Flag;

Find\_White\_Range\_Flag =0;

}

//起点直道

if( (Straight\_Road\_Select\_Flag ==1) && (Huan\_Dao\_Ok ==1) && (Zhi\_Xing\_Over ==1) ) //&& (Road\_Inside\_Flag ==1)

{

//外侧直道

if( (Road\_Outside\_Flag ==1) && (Road\_Inside\_Flag ==0) ) //？？？？？？？？？？？？？？？？problem Not eually one

{

for(i =70;i >30;i-=5) //查看并提取4行元素 太少？？？

{

for(j =50;j >1;j--)

{

if( (image[i][j] >=image\_threshold) && (image[i][j-2] >=image\_threshold) )

{Find\_White\_Range\_Flag\_Out ++; break;}

}

}

if(Find\_White\_Range\_Flag\_Out >=7)

{

if(Zebra\_Crossing\_Flag\_Count ==0)

{Straight\_Road\_Select\_Flag =0;Road\_Outside\_Flag =0;Road\_Turn\_Left\_Flag =2;Road\_Turn\_Right\_Flag =1;Stop\_All\_Flag =1;}

else if(Zebra\_Crossing\_Flag\_Count ==1)

{Straight\_Road\_Select\_Flag =0;Road\_Outside\_Flag =0;Road\_Turn\_Left\_Flag =3;Stop\_All\_Flag ++; }

else

{Straight\_Road\_Select\_Flag =0;Road\_Outside\_Flag =0;Road\_Turn\_Left\_Flag ++;Stop\_All\_Flag ++; }

}

// {Straight\_Road\_Select\_Flag =0;Road\_Outside\_Flag =0;Road\_Turn\_Left\_Flag ++;Stop\_All\_Flag ++; } //中间有白区

else

{Straight\_Road\_Select\_Flag =1; }

White\_view\_Out =Find\_White\_Range\_Flag\_Out;

Find\_White\_Range\_Flag\_Out =0; //标志位清零

}

//内侧直道

else if( (Road\_Inside\_Flag ==1) && (Road\_Outside\_Flag ==0))

{

for(i =70;i >30;i-=5) //查看并提取4行元素

{

for(j =100;j <148;j++)

{

if( (image[i][j] >=image\_threshold) && (image[i][j+2] >=image\_threshold) )

{Find\_White\_Range\_Flag\_In ++; break;}

}

}

if(Find\_White\_Range\_Flag\_In >=7)

{

if(Zebra\_Crossing\_Flag\_Count ==0)

{Straight\_Road\_Select\_Flag =0;Road\_Inside\_Flag =0;Road\_Turn\_Right\_Flag =2;Road\_Turn\_Left\_Flag =1;Stop\_All\_Flag =1 ;}

else if(Zebra\_Crossing\_Flag\_Count ==1)

{Straight\_Road\_Select\_Flag =0;Road\_Inside\_Flag =0;Road\_Turn\_Right\_Flag =3;Stop\_All\_Flag ++; }

else

{Straight\_Road\_Select\_Flag =0;Road\_Inside\_Flag =0;Road\_Turn\_Right\_Flag ++;Stop\_All\_Flag ++; }

}

// {Straight\_Road\_Select\_Flag =0;Road\_Inside\_Flag =0; Road\_Turn\_Right\_Flag ++;Stop\_All\_Flag ++;} //中间有白区

else

{Straight\_Road\_Select\_Flag =1; }

White\_view\_In =Find\_White\_Range\_Flag\_In;

Find\_White\_Range\_Flag\_In =0; //标志位清零

}

//判断出错

else

{Straight\_Road\_Select\_Flag =1;}

}

# 系统开发与调试

整个系统的编写与相关功能代码的添加都是在主体框架完成的情况下进行添加的。代码的调试是在赛道上仿真完成的。

代码的编译使用的是IAR Embedded Workbench其编译效率高，并且能够实时编译。

在调试的时候我们除了用IAR的仿真功能外还使用了按键、OLED显示、蓝牙以及匿名的上位机来进行数据波形的显示，采集。在速度的检测中我们直接使用匿名的上位机接收蓝牙发送的速度信息，实时监视速度的变化。查看速度的变化。以此来更改速度值。

## 中线的识别

中线识别程序是整个程序的主体部分，在确定好中线识别程序之后就开始对其进行仿真，用匿名上位机来查看摄像头采集的图像并显示出处理之后的中线的位置，并打开IAR的仿真功能同步查看相关标志位与相关的数据，一步步完善中线的识别，即使是在有干扰的情况下，例如菱形、斑马线等的干扰。也要保证中线的识别是正确的，只有完全保证中线识别是正确的之后的才能进行接下来的相关功能的实现。

## 转向的实现

在中线识别持续都完成的情况下才能进行转向。中线识别提取出来的理论中线数据是转向的原始数据，之后用PID算法解算出输出数据然后供给数字舵机来进行工作转向。再转向完成后就进行实验在相关的赛道上用手推的方法一步一步的来看舵机的反应，在出现问题是就进行仿真，同时打开逐飞上位机来查看相关的道路信息。在测试过程中发现如果舵机的转向不准确就会直接影响接下来的中线识别程序，进而可能早成中线的出粗，形成一个恶性循环。

## 速度的设定

在比赛中速度也是一个非常重要的一个因素，因此在保证稳定的情况下尽可能的提升速度，但也要注意的是弯道的速度不能和直线速度一样，弯道速度是有一个最大值的，超过了这个值就会发生“漂移”的现象，对整体的速度是非常有影响的，由于“漂移”的情况下。后轮是有打滑的，但是速度是有一个设定值得一旦打滑的现象发生会增大电机的负荷，这种情况也是非常危险的。而且外弯道与内部弯道的曲率是不同的，因此外道和内道的速度也是不同的。我们没在测试时找到打滑的速度再此速度上在此减少电机的输出，就能减少打滑现象的发生，因此整体的现象是直道加速，弯道减速。在不同的道路情况下设定不同的速度。

## 信号灯的实现

信号灯的实现是与标志位的识别与速度的检测以及换道转向是息息相关的只有全部都能识别到才能准确实现信号灯的工作。速度的检测是由编码器来实现的，当编码其检测道其速度降低将其信号传送给尾部的信号灯使其工作，在检测道转向时，无论是换道转向还是正常的弯道的转向。都会将其转向的信号传送给左车灯或右车灯，根据转向信号来分别控制左车灯或右车灯的工作。

# 结论

自从报名参加了“中国工程机器人暨国际公开赛”以来，队员一直努力奋斗，不断地讨论摄像头的巡线方案，不断地又再次改进，解决一个有一个的问题，更换一个又一个的解决方案。从零开始，不断地学习，不断的调试。同时遇见自己解决不了的问题就不断地学习去查阅资料。同时也不断地和同学，老师互相交流。在几个月的小车的制作的时间中，一步步的去发现问题，解决问题，遇见困难，克服困难。与此同时也非常感谢，学校和老师对我们比赛以来的重视以及亲切的指导，还有生活上的照料，比赛的过程也记录了我们参赛队员的成长。

尽管小车制作完成了。但我们的小车任然有很多可以改进的地方。

1. 由于小车的部分是由模块化的东西搭建的很多的线都是由杜邦线连接的，因此有些线是连接不紧密的。容易出现问题，此处要改进。
2. 由于自身的知识储备不足。小车的重量与速度之间的关系不能完全联系起来浪费很多的时间在调整速度上。
3. 摄像头的安装不稳定，会常常因为车身的晃动造成采集的图像不稳定给中线的识别造成了很大的影响。
4. 摄像头图像的采集受光线的影响特别大，即使有动态阈值的处理也会受光线的影响。因此阈值的处理还要在准确一点。
5. 对弯道以及直道的识别不能完全依靠图像，最好再增加陀螺仪来辅助判断道路的情况。
6. 对PID算法的理解还不够透彻。

# 参考文献

[1] 贾秀江.摄像头黑线识别算法和赛车行驶控制策略[J].电子产品世界.2007(05):146-147.

[2] 卓晴，黄开胜，邵贝贝。学做智能车.北京-北京航空航天大学出版社.2007.

[3] 刘进，齐晓慧，李永科.基于摄像头的智能小车设计与实现[J].传感器世界.2008(02):34-37.

[4] 杨欣，王玉凤，刘湘黔.电子设计从零开始.清华大学出版社.

# 附录

uint8 otsuThreshold(uint8 \*image, uint16 col, uint16 row) //大津法求阈值

{

#define GrayScale 256

uint16 width = col;

uint16 height = row;

int pixelCount[GrayScale];

float pixelPro[GrayScale];

int i, j, pixelSum = (60 \*150 /4);//((width -40) \* (height -25));

// int k ,t,pixelSum\_wan =(30 \*120 + 20 \*150);

static uint8 threshold = 0;

uint8\* data = image; //指向像素数据的指针

uint8 pixelmin =0xff;

uint8 pixelmax =0x00;

static uint8 Finish\_Ok =0;

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

{

pixelCount[i] = 0;

pixelPro[i] = 0;

}

//统计灰度级中每个像素在整幅图像中的个数

// if(Zhi\_Flag\_1 ==0 && Straight\_Road\_Select\_Flag ==0)

// {

// for(i = 30; i <60; i++)

// {

// for (j = 0; j < 120; j++)

// {

// pixelCount[(int)data[i \* width + j]]++; //将像素值作为计数数组的下标

// if(data[i \* width + j] >= pixelmax) pixelmax =data[i \* width + j];

// else if( data[i \* width + j] <=pixelmin) pixelmin =data[i \* width + j];

// }

// }

// for(i =60; i<height; i++)

// {

// for(j =0; j <width; j++)

// {

// pixelCount[(int)data[i \* width + j]]++; //将像素值作为计数数组的下标

// if(data[i \* width + j] >= pixelmax) pixelmax =data[i \* width + j];

// else if( data[i \* width + j] <=pixelmin) pixelmin =data[i \* width + j];

// }

// }

// //计算每个像素在整幅图像中的比例

// for (i = pixelmin; i < pixelmax; i++)

// {

// pixelPro[i] = (float) pixelCount[i] / pixelSum\_wan ;

// }

// }

//统计灰度级中每个像素在整幅图像中的个数

// else

// {

// for (i = 60; i >20; i--)

// {

// for (j = 0; j < 150; j++)

// {

// pixelCount[(int)data[i \* width + j]]++; //将像素值作为计数数组的下标

// if(data[i \* width + j] >= pixelmax) pixelmax =data[i \* width + j];

// else if( data[i \* width + j] <=pixelmin) pixelmin =data[i \* width + j];

// }

// }

for(i =20; i<height; i+=2) //前30行不扫描

{

for(j =0; j <width; j+=2)

{

pixelCount[(int)data[i \* width + j]]++; //将像素值作为计数数组的下标

if(data[i \* width + j] >= pixelmax) pixelmax =data[i \* width + j];

else if( data[i \* width + j] <=pixelmin) pixelmin =data[i \* width + j];

}

}

//计算每个像素在整幅图像中的比例

for (i = pixelmin; i < pixelmax; i++)

{

pixelPro[i] = (float) pixelCount[i] / pixelSum ;

}

// }

// start\_x =LIMIT( start\_x,0,110 );

// end\_x =LIMIT( end\_x,40,149 );

// for(i =50; i>=10;i-=10) //相隔10行改变扫描范围

// {

// if( i ==50) k=0;

// else if(i ==40) k =1;

// else if(i ==30) k =2;

// else if(i ==20) k =3;

// else if(i ==10) k =4;

//// else if(i ==0) k =5;

// for(t =0;t <10;t++)

// {

// if( start\_x <3 && Straight\_Road\_Select\_Flag ==0 )

// {

// for(j =0; j<(end\_x -k\*10);j+=2)

// {

// pixelCount[(int)data[(i+t) \* width + j]]++; //将像素值作为计数数组的下标

// if(data[(i+t) \* width + j] >= pixelmax) pixelmax =data[(i+t) \* width + j];

// else if( data[(i+t) \* width + j] <=pixelmin) pixelmin =data[(i+t) \* width + j];

// }

// }

// else

// {

// for(j =(start\_x +k\*5); j<(end\_x -k\*5);j+=2)

// {

// pixelCount[(int)data[(i+t) \* width + j]]++; //将像素值作为计数数组的下标

// if(data[(i+t) \* width + j] >= pixelmax) pixelmax =data[(i+t) \* width + j];

// else if( data[(i+t) \* width + j] <=pixelmin) pixelmin =data[(i+t) \* width + j];

// }

// }

// }

// }

//遍历灰度级[0,255]

float w0, w1, u0tmp, u1tmp, u0, u1, u, deltaTmp, deltaMax = 0;

if(Finish\_Ok ==0)

{

Finish\_Ok =1;

for (i = 0; i < 255; i++) // i作为阈值 GrayScale

{

w0 = w1 = u0tmp = u1tmp = u0 = u1 = u = deltaTmp = 0;

for (j = 0; j < GrayScale; j++)

{

if (j <= i) //背景部分

{

w0 += pixelPro[j];

u0tmp += j \* pixelPro[j];

}

else //前景部分

{

w1 += pixelPro[j];

u1tmp += j \* pixelPro[j];

}

}

u0 = u0tmp / w0;

u1 = u1tmp / w1;

u = u0tmp + u1tmp;

deltaTmp = w0 \* pow((u0 - u), 2) + w1 \* pow((u1 - u), 2);

if (deltaTmp > deltaMax)

{

deltaMax = deltaTmp;

threshold = i;

}

}

}

if( Finish\_Ok ==1)

{

for (i = threshold -10; i < threshold +10; i++) // i作为阈值 GrayScale

{

w0 = w1 = u0tmp = u1tmp = u0 = u1 = u = deltaTmp = 0;

for (j = 0; j < GrayScale; j++)

{

if (j <= i) //背景部分

{

w0 += pixelPro[j];

u0tmp += j \* pixelPro[j];

}

else //前景部分

{

w1 += pixelPro[j];

u1tmp += j \* pixelPro[j];

}

}

u0 = u0tmp / w0;

u1 = u1tmp / w1;

u = u0tmp + u1tmp;

deltaTmp = w0 \* pow((u0 - u), 2) + w1 \* pow((u1 - u), 2);

if (deltaTmp > deltaMax)

{

deltaMax = deltaTmp;

threshold = i;

}

}

}

return threshold;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数功能：舵机控制车模方向

入口参数：无

返回值 ：

函数说明：用一组PID进行计算

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Road\_Control(void)

{

//计算实际偏差

int8 i;

int16 Stero\_pwm;

float Target =0;

static int Target\_Last =75;

// Array\_Rank(Fit\_Middle,3);

// Array\_Filter(Fit\_Middle,3); //进行数组滤波处理

for(i =3;i >0;i--) // 相应计算3行元素

{

Target += i \*Fit\_Middle[i+52];

}

Target /=6; //使用加权平均

// if(Zhi\_Flag\_1 ==0)

// Target =Fit\_Middle[55];

if( (ABS(Target -Target\_Last)) > 30) Target =Target\_Last;

Stero\_pwm =PID\_Position\_Calcualte(75,(float)Target,&Steer\_arg,&Steer\_val); //直道控制

Stero\_pwm +=1510; //理论值 1520

//以下用于观测变量值

Stero\_pwm\_view =Stero\_pwm;

Target\_view =(int)Target;

Stero\_pwm =LIMIT(Stero\_pwm,1200,1800); //用于限制舵机转角范围

Target\_Last =(int)Target;

Set\_Steer\_Pwm(Stero\_pwm); //舵机

// Pwm\_Test(); //电机测试

}

int Target =0,Target\_Last =0;

extern uint8 Huan\_Dao\_Ok;

uint8 Stop\_Flag =1;

uint8 Decelerate\_Flag =0;

uint8 Detect\_Ok =0; //红外检测 Ok标志位

uint8 Voice\_Detect\_Ok =0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数功能：电机速度控制 (稳定速度) （需现场修改编码器的圈数）？？？？？？？？

入口参数：无 内弯道 40R 外弯道 50R

返回值 ：

函数说明：各阶段赛道采取不同速度控制

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Motor\_Control(void)

{

// static uint8 Decelerate\_Falg =0;

static int pwm =0;

static uint16 count =0;

// static uint8 Stop\_Flag =1;

//内道

if(Road\_Justice\_Flag ==0)

{

//正常区

if(Decelerate\_Flag ==0 )

{

//起点----终点直道

if(Straight\_Road\_Select\_Flag ==1)

{ //第一圈

if(Road\_Turn\_Right\_Flag ==1 && Road\_Turn\_Left\_Flag ==1)

Target =850; //原值 900

//第二圈 (换道速度)

else if( (Road\_Turn\_Right\_Flag ==3 && Road\_Turn\_Left\_Flag ==2) || (Road\_Turn\_Right\_Flag ==2 && Road\_Turn\_Left\_Flag ==3))

{

if(Huan\_Dao\_Ok ==1) Target =1100;

else Target =500;

}

//第三圈

else

Target =1200;

}

//普通直道

else if( (Zhi\_Flag\_1 ==1) && (Wan\_Flag\_1 ==0) && (Straight\_Road\_Select\_Flag ==0))

{

Target =850; //原值 850

}

//弯道

else if( (Zhi\_Flag\_1 ==0) && (Wan\_Flag\_1 ==1) && (Straight\_Road\_Select\_Flag ==0))

{

Target =750; //原值 700

}

}

Target\_Last =Target; //记录上次目标值

//红外检测、超声波检测

if(Straight\_Road\_Select\_Flag ==1 && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count ==1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//减速区

if(Stop\_Flag ==1)

{

//菱形 //先用固定值代替

if(Diamond\_Shape\_Flag ==1 && Total\_Rad >400 && Total\_Rad <460 ) //原值 360-420

{ Decelerate\_Flag =1;}

else if(Diamond\_Shape\_Flag >=2 && Total\_Rad >400 && Total\_Rad <460 ) {Decelerate\_Flag =2;}

if(Decelerate\_Flag ==1) Target =600;

else if(Decelerate\_Flag ==2) Target =200;

if(Zebra\_Crossing\_Flag\_Count ==1) //仅限刹车时间，检测

{

Target\_Last =Target; //记录上次目标值

//红外检测、超声波检测

if(Straight\_Road\_Select\_Flag ==1 && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count ==1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

}

//斑马线停车

if(Zebra\_Crossing\_Flag ==1 && Total\_Rad >410 )

{

Target =0;

if(Velocity <100)

{

count ++;

Decelerate\_Flag =0;

if(count >1000)

{

count =0;

Target =1000;

Stop\_Flag =0;

}

}

}

}

if( Stop\_All\_Flag >=3 && (Total\_Rad >600)) //停车标志位 （后面加弯道判断标志）

{

Target =0;

}

}

//外道

else

{

//正常区

if(Decelerate\_Flag ==0 )

{

//起点----终点直道

if(Straight\_Road\_Select\_Flag ==1)

{ //第一圈

if(Road\_Turn\_Right\_Flag ==1 && Road\_Turn\_Left\_Flag ==1)

Target =1000;

//第二圈 (换道速度)

else if( (Road\_Turn\_Right\_Flag ==3 && Road\_Turn\_Left\_Flag ==2) || (Road\_Turn\_Right\_Flag ==2 && Road\_Turn\_Left\_Flag ==3))

{

if(Huan\_Dao\_Ok ==1) Target =1100;

else Target =500;

}

//第三圈

else

Target =1200;

}

//普通直道

else if( (Zhi\_Flag\_1 ==1) && (Wan\_Flag\_1 ==0) && (Straight\_Road\_Select\_Flag ==0))

{

Target =900;

}

//弯道

else if( (Zhi\_Flag\_1 ==0) && (Wan\_Flag\_1 ==1) && (Straight\_Road\_Select\_Flag ==0))

{

Target =800;

}

}

//红外检测、超声波检测

Target\_Last =Target; //记录上次目标值 (专为减道区 使用检测装置)

if( (Straight\_Road\_Select\_Flag ==1 || Zhi\_Flag\_1 ==0) && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count >=1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//减速区

if(Stop\_Flag ==1)

{

//菱形 //先用固定值代替

if(Diamond\_Shape\_Flag ==1 && Total\_Rad >380 && Total\_Rad <440 ) //测试 原值 400-460

{Decelerate\_Flag =1;}

else if(Diamond\_Shape\_Flag ==2 && Total\_Rad >380 && Total\_Rad <440 ) {Decelerate\_Flag =2;}

if(Decelerate\_Flag ==1) Target =600;

else if(Decelerate\_Flag ==2) Target =200;

if(Zebra\_Crossing\_Flag\_Count ==1) //仅限刹车时间

{

Target\_Last =Target; //记录上次目标值

//红外检测、超声波检测

if(Straight\_Road\_Select\_Flag ==1 && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count ==1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

}

//斑马线停车

if(Zebra\_Crossing\_Flag ==1 && Total\_Rad >400 )

{

Target =0;

if(Velocity <100)

{

count ++;

Decelerate\_Flag =0;

if(count >850)

{

count =0;

Target =1000;

Stop\_Flag =0;

}

}

}

}

if( Stop\_All\_Flag >=3 && (Total\_Rad >580)) //停车标志位 （后面加弯道判断标志）

{

Target =0;

}

}

if( (Target -Velocity >= 400) && (Zhi\_Flag\_1 ==1) && (Detect\_Ok ==0) &&(Voice\_Detect\_Ok ==0)) pwm =800;

else

{

if(Velocity <10) {Straight\_val.integral =0; } //速度零，停车，积分项清零

pwm = -( PID\_Position\_Calcualte(Velocity,Target,&Straight\_arg,&Straight\_val) );

}

// Target\_Last =Target; //记录上次目标值

if(pwm >=0)

{

pwm =LIMIT(pwm,0,800);

Set\_Pwm(pwm,0);

}

if(pwm <0)

{

pwm =ABS(pwm);

pwm =LIMIT(pwm,0,800);

Set\_Pwm(0,pwm);

}

pwm\_view =pwm;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数功能：电机速度控制 (提速)

入口参数：无

返回值 ：

函数说明：各阶段赛道采取不同速度控制

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Motor\_Control\_Fast(void)

{

// static uint8 Decelerate\_Falg =0;

static int pwm =0;

static uint16 count =0;

// static uint8 Stop\_Flag =1;

//内道

if(Road\_Justice\_Flag ==0)

{

//正常区

if(Decelerate\_Flag ==0 )

{

//起点----终点直道

if(Straight\_Road\_Select\_Flag ==1)

{ //第一圈

if(Road\_Turn\_Right\_Flag ==1 && Road\_Turn\_Left\_Flag ==1)

Target =1000; //原值 900

//第二圈 (换道速度)

else if( (Road\_Turn\_Right\_Flag ==3 && Road\_Turn\_Left\_Flag ==2) || (Road\_Turn\_Right\_Flag ==2 && Road\_Turn\_Left\_Flag ==3))

{

if(Huan\_Dao\_Ok ==1) Target =1100;

else Target =500;

}

//第三圈

else

Target =1300;

}

//普通直道

else if( (Zhi\_Flag\_1 ==1) && (Wan\_Flag\_1 ==0) && (Straight\_Road\_Select\_Flag ==0))

{

Target =900; //原值 850

}

//弯道

else if( (Zhi\_Flag\_1 ==0) && (Wan\_Flag\_1 ==1) && (Straight\_Road\_Select\_Flag ==0))

{

Target =800; //原值 700

}

}

Target\_Last =Target; //记录上次目标值

//红外检测、超声波检测

if( (Straight\_Road\_Select\_Flag ==1 || Zhi\_Flag\_1 ==0)&& Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count >=1)

Hongwai\_Control(); //限制第二圈 换道区 弯道区 第二圈、第三圈都要检测

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//减速区

if(Stop\_Flag ==1)

{

//菱形 //先用固定值代替

if(Diamond\_Shape\_Flag ==1 && Total\_Rad >400 && Total\_Rad <460 ) //原值 360-420

{ Decelerate\_Flag =1;}

else if(Diamond\_Shape\_Flag >=2 && Total\_Rad >400 && Total\_Rad <460 ) {Decelerate\_Flag =2;}

if(Decelerate\_Flag ==1) Target =600;

else if(Decelerate\_Flag ==2) Target =200;

//红外检测、超声波检测

Target\_Last =Target; //记录上次目标值 (专为减道区 使用检测装置)

if( (Straight\_Road\_Select\_Flag ==1 || Zhi\_Flag\_1 ==0) && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count >=1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//斑马线停车

if(Zebra\_Crossing\_Flag ==1 && Total\_Rad >410 )

{

Target =0;

if(Velocity <100)

{

count ++;

Decelerate\_Flag =0;

if(count >840)

{

count =0;

Target =1000;

Stop\_Flag =0;

}

}

}

}

if( Stop\_All\_Flag >=3 && (Total\_Rad >600)) //停车标志位 （后面加弯道判断标志）

{

Target =0;

}

}

//外道

else

{

//正常区

if(Decelerate\_Flag ==0 )

{

//起点----终点直道

if(Straight\_Road\_Select\_Flag ==1)

{ //第一圈

if(Road\_Turn\_Right\_Flag ==1 && Road\_Turn\_Left\_Flag ==1)

Target =1100;

//第二圈 (换道速度)

else if( (Road\_Turn\_Right\_Flag ==3 && Road\_Turn\_Left\_Flag ==2) || (Road\_Turn\_Right\_Flag ==2 && Road\_Turn\_Left\_Flag ==3))

{

if(Huan\_Dao\_Ok ==1) Target =1100;

else Target =500;

}

//第三圈

else

Target =1300;

}

//普通直道

else if( (Zhi\_Flag\_1 ==1) && (Wan\_Flag\_1 ==0) && (Straight\_Road\_Select\_Flag ==0))

{

Target =1000;

}

//弯道

else if( (Zhi\_Flag\_1 ==0) && (Wan\_Flag\_1 ==1) && (Straight\_Road\_Select\_Flag ==0))

{

Target =900;

}

}

Target\_Last =Target; //记录上次目标值

// Hongwai\_Control();

//红外检测、超声波检测

if( (Straight\_Road\_Select\_Flag ==1 || Zhi\_Flag\_1 ==0) && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count >=1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//减速区

if(Stop\_Flag ==1)

{

//菱形 //先用固定值代替

if(Diamond\_Shape\_Flag ==1 && Total\_Rad >380 && Total\_Rad <440 ) //测试 原值 400-460

{Decelerate\_Flag =1;}

else if(Diamond\_Shape\_Flag ==2 && Total\_Rad >380 && Total\_Rad <440 ) {Decelerate\_Flag =2;}

if(Decelerate\_Flag ==1) Target =600;

else if(Decelerate\_Flag ==2) Target =200;

//红外检测、超声波检测 (专为减道区 使用检测装置)

Target\_Last =Target; //记录上次目标值

if( (Straight\_Road\_Select\_Flag ==1 || Zhi\_Flag\_1 ==0) && Total\_Rad >200 && Zebra\_Crossing\_Flag\_Count >=1)

Hongwai\_Control(); //限制第二圈 换道区

if(Zhi\_Flag\_1 ==1 && Total\_Rad >240 && Detect\_Ok ==0)

KS103\_Disatance\_Control();

//斑马线停车

if(Zebra\_Crossing\_Flag ==1 && Total\_Rad >400 )

{

Target =0;

if(Velocity <100)

{

count ++;

Decelerate\_Flag =0;

if(count >1000)

{

count =0;

Target =1000;

Stop\_Flag =0;

}

}

}

}

if( Stop\_All\_Flag >=3 && (Total\_Rad >580)) //停车标志位 （后面加弯道判断标志）

{

Target =0;

}

}

if(Target -Velocity >= 450 && Zhi\_Flag\_1 ==1) pwm =800;

else

{

if(Velocity <10) {Straight\_val.integral =0; } //速度零，停车，积分项清零

pwm = -( PID\_Position\_Calcualte(Velocity,Target,&Straight\_arg,&Straight\_val) );

}

// Target\_Last =Target; //记录上次目标值

if(pwm >=0)

{

pwm =LIMIT(pwm,50,900);

Set\_Pwm(pwm,0);

}

if(pwm <0)

{

pwm =ABS(pwm);

pwm =LIMIT(pwm,50,900);

Set\_Pwm(0,pwm);

}

pwm\_view =pwm;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数功能：光电管检测控制

入口参数：无

返回值 ：无

函数调用：执行时放在 超声波检测后面。在换道区使用光电管。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Hongwai\_Control(void)

{

switch (Road\_Justice\_Flag)

{

case 0:

{

switch (Zhi\_Flag\_1)

{

case 0:

{

if( Hongwai\_Status(Hongwai\_2) ==0 )

{Target =50;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_3) +Hongwai\_Status(Hongwai\_4) <2)

// {Target =100;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_5) +Hongwai\_Status(Hongwai\_6) <2)

// {Target =100;Detect\_Ok =1 ;}

else

{Target =Target\_Last;Detect\_Ok =0;}

break;

}

case 1:

{

if( Hongwai\_Status(Hongwai\_1) +Hongwai\_Status(Hongwai\_2) <2 )

{Target =0;Detect\_Ok =1 ;}

else if( Hongwai\_Status(Hongwai\_3) +Hongwai\_Status(Hongwai\_4) <2)

{Target =100;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_5) +Hongwai\_Status(Hongwai\_6) <2)

// {Target =100;Detect\_Ok =1 ;}

else

{Target =Target\_Last;Detect\_Ok =0;}

break;

}

}

break;

}

case 1:

{

switch (Zhi\_Flag\_1)

{

case 0:

{

if( Hongwai\_Status(Hongwai\_2) ==0 )

{Target =0;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_3) +Hongwai\_Status(Hongwai\_4) <2)

// {Target =100;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_5) +Hongwai\_Status(Hongwai\_6) <2)

// {Target =100;Detect\_Ok =1 ;}

else

{Target =Target\_Last;Detect\_Ok =0;}

break;

}

case 1:

{

if( Hongwai\_Status(Hongwai\_1) +Hongwai\_Status(Hongwai\_2) <2 )

{Target =0;Detect\_Ok =1 ;}

else if( Hongwai\_Status(Hongwai\_3) +Hongwai\_Status(Hongwai\_4) <2)

{Target =100;Detect\_Ok =1 ;}

// else if( Hongwai\_Status(Hongwai\_5) +Hongwai\_Status(Hongwai\_6) <2)

// {Target =100;Detect\_Ok =1 ;}

else

{Target =Target\_Last;Detect\_Ok =0;}

break;

}

}

break;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数功能：超声波检测距离控制

入口参数：无

返回值 ：检测距离1 m以内

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void KS103\_Disatance\_Control(void)

{

static uint8 Brake\_Rank =0;

//switch版本减速

if(KS103\_Distance <=150 && KS103\_Distance >120) {Brake\_Rank =1;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=120 && KS103\_Distance >90){Brake\_Rank =2;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=90 && KS103\_Distance >60) {Brake\_Rank =3;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=60 && KS103\_Distance >30) {Brake\_Rank =4;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=30 && KS103\_Distance >20) {Brake\_Rank =5;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=20 && KS103\_Distance >10) {Brake\_Rank =6;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <=10 && KS103\_Distance >5) {Brake\_Rank =7;Voice\_Detect\_Ok =1 ;}

else if(KS103\_Distance <5) {Brake\_Rank =8;Voice\_Detect\_Ok =1 ;}

else {Brake\_Rank =0; Voice\_Detect\_Ok =0 ;}

switch (Brake\_Rank)

{

case 0:Target =Target\_Last;break;

case 1:Target =(int)1.5 \*KS103\_Distance; break;

case 2:Target =(int)1.5 \*KS103\_Distance; break;

case 3:Target =(int)1.5 \*KS103\_Distance; break;

case 4:Target =50; break;

case 5:Target =30; break;

case 6:Target =20; break;

case 7:Target =10; break;

case 8:Target =0; break;

default: break;

}

}

//位置式PID

int16 PID\_Position\_Calcualte(float feedback,

float expect,

PID\_arg\_t \*pid\_arg,

PID\_val\_t \*pid\_val)

{

static int16 out;

pid\_val->bias =feedback -expect; //求偏差

pid\_val->bias =0.8f \*pid\_val->bias; //进行简单过滤处理

pid\_val->bias +=0.2f \*pid\_val->last\_bias;

pid\_val->integral +=pid\_val->bias;

pid\_val->integral =LIMIT(pid\_val->integral,-10000,10000);

pid\_val->differential =pid\_val->bias -pid\_val->last\_bias; //求微分项

D\_view =pid\_val->differential;

out =(int)(pid\_arg->kp \* pid\_val->bias + pid\_arg->ki \* pid\_val->integral + pid\_arg->kd \*pid\_val->differential);

pid\_val->last\_bias =pid\_val->bias;

return out;

}

//选择法进行排序

void Array\_Rank(uint8 str[],uint8 len) //len长度数组从大到小进行排序

{

uint8 i,j,k,temp;

for(i =0;i <(len-1);i++)

{

k =i;

for(j =(i+1);j <len;j++)

if(str[j] >str[k])

k =j;

if(k !=i)

{temp =str[i];str[i] =str[k];str[k] =temp;}

}

}

1. 队伍名称：Mr.轮子；参赛队员：侯小虎，王梦龙，邹鹏涛；指导老师：董绪斌；具体联系人：侯小虎，联系方式：13166970981 [↑](#footnote-ref-1)