**2014年5月20日星期二**

1. 进行了初步的代码分析，实现了红外传感、灰度、串口、电机的驱动和测试，待车身搭建结束后进行下一步测试；
2. 蓝牙串口调试未成功，

可能问题：

* 1. TTL转RS232连线错误 已解决
  2. 蓝牙波特率 已解决

1. 车身的搭建：

a) 看往届视频的时候发现，当两个车子的铲子交叠，双方都触碰不到彼此的轮子，两者会出现长时间对峙的情况。设想，假如说我方的铲子能够抬起，使敌方的前轮脱离地面，我方就占据了很大优势，我们的战车甚至有机会将敌人铲翻。

b) 当前正在搭建：

搭建可活动的铲子，为保证铲子有足够大的力气，用两个舵机做铲子的关节，再加一个杠杠的结构，用电机用绑带牵引力臂。

c) 可能出现的问题：

1、杠杠、铲子硬度不够；

2、电机的力怎样输出到杠杠，用绑带会不会卡顿；

3、车体尺寸是否会超标？

**2014年5月29日星期四**

进行了电机驱动板的测试，测试结果显示只有4块电机驱动板是健全的，而按照最初的想法需要用到8块驱动板，机身搭建停滞，等待驱动板送修。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类型 | 可用PPM调速，参数可调，有状态返回（可连上MotionTerminal） | 可用PPM调速，参数可调（串口助手），状态无返回 | 可用PPM调速， 参数不可调 | PPM调节异常，参数不可调 | 总数 |
| 数量（块） | 4（均在去年的小车上） | 1 | 2 | 8 | 15 |

注：1、现在健全的只有4块，有3块勉强能用，无法调节状态，调试会很困难；

2、2辆无差别+1辆仿人组，需要8块驱动板。

**2014年5月30日星期五**

进行了电机的测试，电机刚好够用，修好了5个电机（均带测速码盘），有些摩擦力有点大，可以考虑买一瓶润滑油。

**2014年6月1日星期日**

新购进的材料清单到货，清单如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 灰度传感器 | 红外传感器 | 测距传感器 |
| 数量 | 15 | 36 | 6 |

**2014年6月2日星期一**

完成舵机、测距传感器驱动测试，性能优良。

测试阶段结束。

总结：

1. 对各种部件的连线、驱动原理、驱动方法认识很全面了，硬件软件排错能力都很强；
2. 未解决问题：驱动板。

下一阶段任务：

1. 灰度传感器AD值与舞台位置的关系；
2. 距离传感器AD值与目标距离的关系；
3. 红外传感器检测范围调节；

预计用时：2天。（结束日期2014年6月5日）

-------------------------------------------------------------------------------------------------------------------------------。

**2014年6月5日星期四**

发现PPM控制电机非常不稳定：

1. 各PPM输出口的设定值互相影响

（**例如***void SetMotor(INT8U id,INT16S sp)*中设置0~7 id口

sp数值设置为0~1000递增时，设置为0的PPM输出能够使电机静止；

而当sp数值设置为-1000~0递增时，设置为0的PPM输出使电机有一定的速 度）；

1. PPM输出使用定时器刷新，每次刷新的结果都有一定误差，造成电机速度不稳定，会有卡顿的噪声；
2. 对于驱动板来说只需要一次PPM信号便可实现速度控制，不需要一直刷新，但单次PPM输出结果误差可能太大，造成左右轮差速难以调节；
3. 上传*void SetMotor(INT8U id,INT16S sp)* 中sp大小与电机速度的关系，结果显示sp与电机速度呈线性相关。

**2014年6月7日星期六**

1. 考虑使用串口代替PPM信号控制电机，

串口控制电机的优点：

1. 速度控制精确稳定，不用考虑差速问题；
2. 可查询当前电机状态（如目标速度、当前速度、电机角度、温度等）。

缺点：占用了串口，可能给调试造成不便。

1. 上传串口控制电机驱动案例，已测试成功。

**2014年6月8日星期日**

1. 电机串口驱动程序初步编写，测试显示，给定速度之后，小车左右轮基本无差速；
2. 小车转弯，使用驱动板的位置模式控制，理论上可以精确控制小车旋转角度，运动的位置变量待测试。

**2014年6月9日星期一**

1. 发现位置模式实现小车转弯有一下几个问题：
   1. 小车顺时针、逆时针旋转同一角度，其位置变量的取值不一致；
   2. 小车转弯角度受舞台粗糙程度影响，舞台各位置旋转角度有差异；
   3. 小车位置模式旋转初始速度为0，旋转太慢；
2. 蓝牙通讯成功；

**2014年6月10日星期二**

1. 发现蓝牙与驱动板通讯成功率极低，可能是TTL转RS232的过程产生干扰；
2. ZigBee通讯成功，将使用ZigBee代替蓝牙；

3、小车旋转问题，发现准备开始旋转时，将小车速度提至最高，并提高加速度、减速度，小车旋转会变精准。

**2014年6月11日星期三**

1. ZigBee与驱动板通讯成功，成功率良好；
2. 后轮改变为两个轮子的结构；
3. 发现左前轮在转弯时有差速，速度较小时不启动，速度较大时速度较其他轮慢，可能是电机摩擦力太大，考虑换一个电机；（其实是加速度的问题，加速度大动力就大 ——8月26日）
4. 用ZigBee之后可以连上MotionTerminal，不知道怎么用串口助手调试可以直接用这个软件调试。

**2014年6月12日星期四**

发现一个比较好用的串口助手（2014/资料/软件/XCOM），数字键绑定指令绑定，预先编写指令绑定+ZigBee就变成了遥控车。

**2014年6月13日星期五**

车体改造，舵机处加支撑结构。

**2014年6月15日星期日**

1、发现新购进的红外传感器参数参差不齐，普遍距离达不到预期距离；（传感器末端有写参数，30nk的是30厘米的，很短，80nk的比较理想 ——8月26日）

2、发现驱动板位置模式控制小车转弯，没有速度模式下控制精确，因为位置模式下两个轮子会存在差速，不会同时停下来。

**2014年6月16日星期一**

1. 测量速度模式下小车转弯角度与时延的关系；
2. 完成定时器代码框架；
3. 完善串口控制电机驱动；
4. 完成小车边缘检测外部中断框架（外部中断没有高电平触发，只能使用上升沿触发，这可能导致一次危机处理后，小车没有从边缘抽离，但此时无法触发第二次危机处理）
5. 仿人组上身结构设计，使用电机作为腰部，可能超重

**2014年6月17日星期二**

发现小车转弯驱动出现错误，重新测量转弯角度与时延的关系；（控制板换过之后就发现这个问题了，可能不同控制板的参数都是不同的，可见用延时转角度不可靠 ——8月26日）

**2014年6月18日星期三**

1. 仿人组机器人舵机初步测试；
2. 烧写器只有一个是好的，两个人用的时候有点麻烦。

**2014年6月20日~7月1日**

考试期间：

1. 驱动板跟烧写器送修，修好了6块驱动板，烧写器修不好；
2. 出现烧写器不能烧写的情况，有时跟USB接口有关，换USB接口或重启PC可以解决问题。

**2014年7月3日星期四**

1. 每次灰度传感器的测试都会有不同的结果，一直都没有得到满意的校正结果；（实践证明，比较差值求校正方法的精度对编码算法来说已足够，不用纠结 ——8月26日）
2. 发现ZigBee发送信息控制驱动板，会有延时，没有电线反应快。

**2014年7月4日星期五**

1、完善了定时器的驱动，将4个定时器的定时都设定在1ms，用定时器0重写延时函数delayms（int ms），用它来控制小车旋转指定角度更方便（手动控制小车旋转固定角度，记录定时数count，写入程序）；（事实证明控制小车旋转角度参数不现实，用延时是投机取巧，不够智能；定时器的延时函数容易出错，已删除 ——8月26日）

1. 增加了读写EEPROM的驱动，可以将固定的环境参数（如擂台最大、最小灰度值）写入EEPROM，程序启动时再读取，避免换了场地人工修改程序的麻烦。

**2014年7月5日星期六**

**编码：**

1. 朝向编码：根据灰度传感器的读数，**分析边缘检测红外传感器的触发情况**和**比较三个灰度传感器的大小情况**，进行编码，确定小车朝向；

vehicle.dir

位0 位3（右前边缘检测）

位1 位2

1. 轨道编码：**确定三个灰度传感器分别处于哪个轨道**，再将各种可能出现的情况进行编码，确定小车的轨道信息。

vehicle.track2(8位)

00(无效) - -(front) - -(left) **- -**(right)

track0: 00 track1: 01 track2: 10

**2014年7月7日星期一**

使用编码信息，完成漫游代码（没有检测到敌人时回到擂台中心，旋转寻找敌人）， 测试成功，说明编码成功。

**2014年7月8日星期二**

漫游代码优化，成功识别中心武字。

**2014年7月9日星期三**

写出攻击代码框架（只完成了背对中心/面对中心，敌人在前方情况），其他情况下 的攻击策略待讨论之后决定。

**2014年7月22日星期二**

实现**灰度辅助采样**：按照设定的顺序对不同区域进行灰度采样，avr自动读取保存模拟值，采样完毕控制板自动计算灰度偏置以及轨道信息保存在eeprom中，对战程序启动时从eeprom中读取参数。（避免了人工记录，人工计算的繁琐，效率大提高）

**2014年7月24日星期四**

新增menu()函数，提供各种小车参数接口，使用超级终端可进行设置，调试时无需再反复烧写程序。

**2014年7月30日星期三**

增加**记忆周期机制**：当处于传感器盲区时，小车在一个记忆周期内会保存上次检测到敌人的传感器位置，记忆周期内，任意红外开关检测到敌人，则记忆周期中断，根据小车的记忆设定特定动作，即可解决任意两个传感器之间盲区的衔接问题。

**2014年8月4日星期一**

1、增加**角度权值参数**，角落权值越小，角落轨道占的比重越大，小车行动范围越窄， 掉下擂台的可能性就越低，但威胁度也下降。

2、优化灰度采样程序（用相邻轨道间的最大灰度差作为白色到红色的突变值），优化 goToMiddle()：走向中心武字函数，在中心区域稳定性提高。

**2014年8月6日星期三**

1. 仿人组资格认证调试自动找柱子算法（后来发现资格认证只要抱起柱子就行）；
2. 增加walkInTrack1()：在轨道2以内游走算法；

3、出发比赛。

**预选赛：**

**2014年8月8日星期五**

1. 到现场调试，忘记下载超级终端，用串口助手凑合着用，菜单程序降低了工作量；
2. 无差别组其中一个铲效果不好，回去之后改造车体。

**2014年8月9日星期六（比赛日）**：

上午灰度采样，优化跑上擂台算法，左右轮差速1000，加速度20000，延时500ms，可以直接冲向中心，下午比赛。

**比赛中发现的问题：**

无差别组：

1. 车体超规格，底板横放的情况下加两个轮宽度超标；
2. 推完棋子不能拿走车，只有第一场对战前，1分钟准备时间有机会切换算法，算法切换必须要灵活可靠；
3. 2v2不能只上一辆，没有做敌我识别，自己冲撞自己失了很多分；

仿人组：

1、仿人组可以装边缘检测；

2、仿人组也要推棋子，传感器装太高无法识别棋子；

3、走上擂台效果受台阶环境影响，可能是轮子打滑。

**比赛结果：**

无差别组 一等奖 第三名

仿人组 一等奖 第四名

**10月份合肥RobCup公开赛准备：**

**1、改进方向：**

无差别组：

1. 车体改造：前后都双轮，而且不会超标；
2. 敌我识别：两辆车用ZigBee通讯，实现状态共享，根据位置关系进行敌我识别 （考虑使用指南针模块）。

仿人组：

1. 加边缘检测；
2. 攻击算法优化，更强势。
3. **材料清单：**待确定。

**10月份合肥RobCup公开赛比赛结果：**

**无差别组： 冠军（总决赛对决中山大学）**

**仿人组： 第四名**

-------------------------------------------------------------------------------------------------------------------------------。

**项目结束**