

GIFTCAR

简介

这是上海交通大学溥渊未来技术学院新生杯智能车竞赛的示例代码
也适用于智能车竞赛-标准竞速组
使用了思岚C1雷达+ESP32S3开发板
使用了多线程技术防止雷达卡死并极大地降低雷达延迟(毫秒级!!!)

接线参考

- 马达绿线¹¹
- 马达紫线¹²
- 马达蓝线¹³
- 舵机橙线¹⁴
- 雷达黄线¹⁹
- 雷达绿线²⁰

文件说明

rpLidarTypes.cpp & rpLidar.cpp

雷达库的核心代码,不要轻易改动

goldenglowLink.cpp

学长给你们留下的小礼物,一键调用雷达,舵机和马达
→专心搞算法吧孩子,硬件上的苦学长帮你们吃完了
→名字来源于澄闪,一只源石技艺是电的可爱折耳猫 | 夹带私货这一块

```
void initialize();
```

初始化小车,请务必在程序的初始化中调用一次

```
void lidarLoop();
```

持续获取雷达数据的死循环,千万不要主动调用,谨慎修改

```
void getLidarData(float* buffer);
```

一键调用雷达

将雷达获取的数据复制到`buffer`数组里,内容为从当前时间~一圈前的有效数据(刚好一圈)

- `buffer` 数组应该在调用本函数之前自己创建,类型是 `float[360]`
- `buffer[i]` 表示雷达在 i° 测得的距离(单位 `mm`),无效数据为 `0`
- 留意 `0` 数据

```
void operateServo(float steering);
```

一键调用舵机

- `steering` 为舵机角度,取值范围 $[0^\circ, 180^\circ]$
- 面向舵机的轴观察, 0° 右, 90° 上, 180° 左
- 一般取值不要超过 $[60^\circ, 120^\circ]$
- 谨慎使用极端角度,可能会损坏舵机(平均一个组烧两个舵机)

```
void operateMotor(float speed);
```

一键调用马达

- `speed` 为马达相对输出功率,取值范围 $[-1, 1]$
- 负后退正前进
- 一般 `abs(speed) >= 0.6`,否则车可能动不起来

debug.cpp

```
void debugReadDistance();
```

测试雷达

预期输出:前后左右四个方向的距离(可能有 `0`,说明这个点漏了,问题不大)和一圈的有效样本点个数(这个值通常在 $[200, 300]$)

```
void debugOperateServo();
```

测试舵机

预期输出:舵机在 80° 和 100° 之间来回转动

```
void debugOperateMotor();
```

测试马达

预期输出:马达在后退和前进之间来回切换

core.cpp

主程序,请把你的代码写在这里

```
void initCore();
```

初始化,会在程序启动时调用一次

```
void loopCore();
```

主循环,会重复执行

```
void scan();
```

获取雷达数据并存储在data[]数组中

```
void decide();
```

根据读取到的雷达数据进行决策

出于一些考量,我们不能提供参加国赛的代码,你看到的这份代码有很大的优化空间,请自由发挥

```
void move();
```

根据决策结果控制舵机和马达的输出

出于一些考量,我们不能提供参加国赛的代码,你看到的这份代码有很大的优化空间,请自由发挥