## Car库函数

|  |  |
| --- | --- |
| **函数名** | **描述** |
| Car | 构造函数 |
| Mode | 设置引脚模式 |
| Infer | 设置机器人前进时IN1和IN2的电平 |
| Move | 控制机器人的基本行动 |

### Car

原形：Car(int a, int b, int c, int d);

前2个参数分别为IN1-IN2的引脚。IN1控制左电机转向，IN2控制右电机转向。后2个参数分别为控制左右电机转速的PWM引脚。同时默认机器人前进时IN1和IN3均为高电平。

### Mode

原形：void Mode();

设置所有引脚为输出模式。

### Move

原形：void Move(int a, int b, int c, unsigned long d = 1);

控制机器人的基本行动。前2个参数分别为左右电机的转速，第3个参数用于控制机器人的行动：2，4，6，8分别表示后退，左转，右转，前进。第4个参数为延时，单位毫秒，默认为1毫秒。

### Infer

原形：void Infer(int a, int b);

设置机器人前进时IN1和IN2的电平。默认均为高电平。若与实际情况不符，可通过此函数修改。使用方法示例：1.调用Move(140，140，2，5000)；2.观察电机转向。左轮后退，则把第1个参数改为0；右轮后退，则把第2个参数改为0。

### 例程

#include "Car.h"

Car mycar(10,11,12,13);

void setup(){

mycar.Mode();

mycar.Infer(1,1);

}

void loop(){

mycar.Move(140,140,2,1000);

}

## Track库函数

说明：记从左到右的3个红外为1-3号

|  |  |
| --- | --- |
| **函数名** | **描述** |
| Track | 构造函数 |
| Mode | 设置引脚模式 |
| Set | 设置阈值 |
| Test | 读取红外检测的模拟值 |
| Digital\_in | 读取红外的检测结果 |

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **描述** |
| value | int[] | 长度为3的数组。用于存放红外检测的模拟值。标号0-2对应1-3号红外 |

### Track

原形：Track(int a, int b, int c,int f=true);

3个参数分别对应1-3号红外的引脚。最后一个参数为true时，若循迹模块的输出模拟量高于阈值则被判定为白色；为false时。则被判定为黑色。默认值为true。

### Mode

原形：void Mode();

设置所有引脚为输出模式。

### Set

原形：void Set(int a, int b, int c);

设置各个红外的阈值，用于区分黑白。3个参数分别对应1-3号红外。

### Test

原形：void Test();

读取红外的模拟值。并将结果存放在value数组中。

### Digital\_in

原形：int Digital\_in(int a = 3);

读取红外的检测结果。返回值为由0和1组成的整数。从高位到低位分别对应1-3号红外。其中0表示检测到黑色，1表示检测到白色。参数可用来控制读取某个红外的检测结果：0-2表示1-3号红外，3或以上的数表示读取所有红外的检测结果。默认值为3。注意：高位的0会被忽略。如检测结果为黑黑白（001），则返回值为1；又如检测结果为黑白白（011），则返回值为11；此外，调用此函数前应调用Set函数设定阈值。

### 例程

#include "Track.h"

Track mytrack(A0,A1,A2, true);

void setup(){

Serial.begin(9600);

mytrack.Mode();

}

void loop(){

mytrack.Test();

for(inti=0;i<3;i++){

Serial.println(mytrack.value[i]);

}

Serial.println("END");

delay(1000);

}

## Function库函数

|  |  |
| --- | --- |
| **函数名** | **描述** |
| ServoMove | 用于控制舵机转动 |
| Modify | 自动调整寻线传感器的阈值 |
| Tracking | 控制机器人寻线，当寻线传感器检测到全黑的时候会停止行进 |
| TurnL | 原地左转90度 |
| TurnR | 原地右转90度 |
| TurnC | 原地转180度 |
| pickup() | 夹取时舵机摆动的顺序 |
| take() | 红外传感器检测到圆柱体采取夹取动作 |

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **描述** |
| SPEED1 | int | 左电机转速 |
| SPEED2 | int | 右电机转速 |
| TURN | int | 转弯时速度 |
| DELAY | int | 转弯延时 |
| BACK | int | 后退延时 |
| mycar | Car | mycar(10,11,12,13) |
| mytrack | Track | mytrack(A0，A1，A2) |

### ServoMove

原形：voidServoMove(int a, int b, int c);

控制舵机转动。前2个参数分别指定舵机从角度a转到角度b，第3个参数指定舵机号。

舵机从上往下编号为1,2,3,4

### Modify

原形：void Modify();

自动调整寻线传感器的阈值。使用操作如下：打开开关，蜂鸣器声音响起，提示将寻线传感器的红外全部对准白色区域；蜂鸣器声音变化时，提示将红外全部对准黑线；蜂鸣器声音再度变化时，表示完成调节。

### Tracking

原形：void Tracking(int a, int b, int temp = 101);

控制机器人寻线，当寻线传感器检测到全黑的时候会停止行进。前2个分别参数表示左右电机的转速，第3个参数一般情况下使用默认值即可。注意：使用此函数前需设定寻线传感器阈值。

### TurnLTurnRTurnC

原形：voidTurnL(); void TurnR(); void TurnC();

控制机器人转弯的函数。转弯角度的判定依赖于地图上的黑线，故需要先设置阈值。