基于Arduino的四轮小车控制系统。

以下是对代码的总体概述：

硬件组成：

1.四个带编码器的直流电机

2.MPU6050陀螺仪加速度计

3.超声波传感器

4.舵机（用于控制超声波传感器的方向）

主要功能：

1.电机速度控制（使用PID算法）

2.障碍物检测和避障

3.位置和姿态估计

4.自主导航

代码结构：

1.初始化函数（setup()）：设置硬件、初始化传感器和PID控制器

2.主循环（loop()）：实现小车的主要行为逻辑

3.中断处理函数：用于定时执行PID计算和电机速度读取

4.各种辅助函数：实现具体的功能，如移动、转向、障碍物检测等

关键算法：

1.PID控制：用于精确控制电机速度

2.卡尔曼滤波：用于融合陀螺仪和加速度计数据，提高姿态估计精度

3.障碍物检测算法：使用超声波传感器检测周围环境

运行流程：

1.小车默认向前移动

2.通过超声波传感器持续检测前方障碍物

3.当检测到障碍物或达到距离限制时，停止并进行环境扫描

4.根据扫描结果决定转向方向，然后继续前进

特殊功能：

1.使用MPU6050进行姿态估计和位置计算

2.通过舵机控制超声波传感器进行360度环境扫描

这个项目综合了多种传感器和控制算法，实现了一个具有自主避障能力的智能小车系统。

以下是对代码的详细解释：

初始化函数

setup()函数是Arduino的初始化函数，用于设置电机控制引脚、PID控制器参数、串口通信、MPU6050传感器和超声波传感器。它还启动了一个定时器中断，每40毫秒触发一次，用于定期执行PID计算和电机速度读取。

主循环

loop()函数是Arduino的主循环函数，根据前方是否有障碍物和行驶距离是否达到上限来决定机器人的行为。它调用了不同的函数来实现前进、转向和障碍物检测等功能。

移动和转向

moveForward()函数用于控制机器人向前移动。它通过PID控制器调整四个电机的速度，以保持直线行驶。turn(float turnAngle)函数用于控制机器人转向，直到达到目标角度。stop()函数用于停止所有电机。

定时器中断处理

interruptHandler()函数是定时器中断处理函数，根据当前状态机状态执行不同的操作，如PID计算和电机速度读取。

传感器初始化

mpuInit()函数初始化MPU6050传感器，设置加速度计和陀螺仪的量程，并进行传感器校准。ultrasonicInit()函数初始化超声波传感器和舵机。

数据处理

moveCaculate()函数获取MPU6050传感器数据，计算机器人的位移和角度，并更新机器人的坐标。kalmanFilter()函数应用卡尔曼滤波器来平滑传感器数据。

障碍物检测

obstacleDetectMoving()和obstacleDetectStand()函数分别在移动和静止时进行障碍物检测。getDistance()函数获取超声波传感器测量的距离。obstacleDistanceCheck()函数判断是否有障碍物。

行驶距离检测

distanceLimitDetect()函数计算机器人行驶的距离，当距离超过设定值时，设置距离上限标志位。

超声波扫描

readUltrasonicStand()函数在机器人静止时进行超声波扫描，获取不同角度的距离数据。

PID控制

PID\_Cal\_Computer\_Out()函数定期计算PID输出并更新电机PWM值。Motor\_PWM\_Set()函数根据PID计算结果设置电机的PWM输出，以控制电机转速和方向。

电机初始化

Motor\_Init()函数初始化电机控制引脚和编码器引脚，并确保所有电机停止。

通过这些函数的协同工作，这个四轮机器人能够实现自动行驶、障碍物检测和避障等功能。