# 大连海事大学

单片机应用实践报告

实验内容：自平衡两轮小车

专业班级：电子信息工程2班

姓 名：江一诺

学 号：2220220821

## 摘要

随着人工智能和机器人技术的发展，自平衡两轮小车作为一种具有典型不稳定系统特征的控制对象，已成为控制理论研究和实践教学的重要平台。本项目基于STM32F103C8T6单片机设计并实现了一款自平衡两轮小车，通过MPU6050陀螺仪和加速度计传感器实现姿态检测，采用分层PID控制策略实现车体平衡，并结合超声波测距实现避障功能。该系统通过无线串口模块实现远程控制，并利用OLED显示屏实时展示系统状态信息。本文详细介绍了系统的硬件设计、软件算法以及功能实现，重点阐述了基于PID控制的平衡策略和避障算法。实验证明，本设计方案能够有效地实现小车的自平衡功能，同时拥有良好的移动控制性能和环境适应能力，为进一步的智能机器人研究奠定了基础。

## 一、设计的任务和要求

### 需求分析

1. 自平衡控制：小车能够在无外力干扰的情况下保持垂直平衡状态
2. 远程遥控：通过无线通信方式实现对小车前进、后退、转向的远程控制
3. 避障功能：自动检测前方障碍物并做出相应反应，防止碰撞
4. 状态显示：实时显示小车的工作状态、参数和传感器数据
5. 安全保护：当小车倾角过大时能自动切断电机电源，防止损坏

### 硬件设计要求

1. 选择合适的STM32系列芯片，确定必要的外设接口和性能参数
2. 设计姿态检测模块，获取小车的实时倾角和角速度信息
3. 设计驱动电路，控制两个带编码器的直流减速电机
4. 设计超声波测距模块电路，用于障碍物检测
5. 设计OLED显示电路，提供直观的系统状态显示
6. 设计无线通信模块，实现远程控制功能

### 软件设计要求

1. 实现MPU6050数据采集和处理算法，获取准确的姿态角度
2. 设计和实现多级PID控制算法，包括直立环、速度环和转向环
3. 实现超声波测距及避障算法
4. 设计电机控制和编码器解码算法
5. 实现OLED显示功能
6. 设计完善的失控保护机制

### 测试与优化要求

1. 进行系统稳定性测试，确保小车在各种条件下都能保持平衡
2. 测试遥控系统的响应速度和可靠性
3. 测试避障功能的检测精度和反应速度
4. 优化PID控制参数，提高系统性能

## 二、研究的目的和意义

### 研究目的

本项目旨在设计和实现一个基于STM32单片机的自平衡两轮小车系统，应用现代控制理论和嵌入式系统技术解决典型的倒立摆控制问题。通过该项目，深入研究传感器数据采集与融合、实时控制算法实现及嵌入式系统开发等技术，培养综合运用所学知识解决实际工程问题的能力。

### 研究意义

1. **理论意义**：自平衡两轮小车是典型的不稳定系统，其控制算法研究对理解和掌握现代控制理论有重要帮助，能加深对PID控制、状态反馈控制等控制方法的理解。
2. **实践意义**：该项目集成了传感器数据采集、控制算法实现、驱动电路设计等多个方面，是一个综合性强的嵌入式系统应用，有助于提升实践动手能力和工程设计水平。
3. **教学意义**：自平衡两轮小车是单片机与嵌入式系统课程的经典实验项目，通过本项目的研究，可以加深对多传感器融合、实时控制系统设计等知识的理解和掌握。
4. **应用意义**：自平衡两轮小车是移动机器人的一种重要形式，其研究有助于推动智能移动机器人技术的发展，在家庭服务、工业巡检、智能交通等领域有广阔的应用前景。

## 三、系统方案的论证和选择

### 方案一：基于51单片机的自平衡小车

采用MCS-51系列单片机实现自平衡小车的控制功能。51单片机具有8位CPU，片上资源包括振荡器与时钟电路、32根IO线、外部存储器ROM和RAM寻址范围各64KB、2个16位定时器/计数器、5个中断源和2个中断优先级、全双工串行口等。通过控制I/O口的输出电平控制电机驱动电路，使用定时器实现精确的控制周期。

优点：

* 结构简单，开发环境成熟
* 成本低廉，易于获取
* 学习资料丰富，入门门槛低

缺点：

* 运算能力有限，难以执行复杂的控制算法
* I/O口数量少，难以满足多传感器和多执行机构的需求
* 缺乏硬件乘法器，浮点运算效率低
* 中断资源有限，实时性不够理想

### 方案二：基于STM32F103C8T6的自平衡小车

采用STM32F103C8T6微控制器实现自平衡小车的控制功能。STM32F103C8T6是一款基于ARM Cortex-M3内核的32位微控制器，最高工作频率72MHz，具有丰富的外设资源，包括多个定时器、I2C、SPI、USART等通信接口，以及大量的GPIO引脚。

优点：

* 32位ARM内核，运算能力强，可轻松处理复杂控制算法
* 丰富的外设资源，多个定时器支持PWM输出和编码器接口
* 多种通信接口，便于连接各类传感器和模块
* 低功耗特性，适合电池供电的移动平台
* 强大的中断系统，保证系统实时性

缺点：

* 开发难度略高，需要掌握更多专业知识
* 成本略高于51单片机

### 方案选择

综合比较两种方案，选择基于STM32F103C8T6的自平衡小车方案。理由如下：

1. 自平衡小车的控制算法较为复杂，需要进行大量的数据采集和处理，STM32F103的72MHz主频和32位处理能力可以满足这一需求。
2. 自平衡小车需要连接多种外设（MPU6050、超声波传感器、OLED显示屏、电机驱动等），STM32F103C8T6的丰富外设接口可以满足连接需求。
3. 电机控制需要高精度的PWM输出和编码器接口，STM32F103C8T6的定时器功能可以轻松实现。
4. 自平衡控制需要较高的实时性，STM32的中断系统和DMA功能可以保证数据采集和控制算法的及时执行。

## 四、系统硬件设计

### 硬件总体框架

自平衡两轮小车的硬件系统主要由以下几个部分组成：

1. 控制核心：STM32F103C8T6微控制器
2. 姿态检测：MPU6050六轴陀螺仪加速度计
3. 驱动系统：电机驱动模块和带编码器的直流减速电机
4. 测距模块：HC-SR04超声波传感器
5. 显示系统：0.96寸OLED显示屏
6. 通信模块：无线串口模块
7. 电源系统：锂电池及电源管理电路

### 元器件清单

| 序号 | 名称 | 型号 | 数量 | 功能描述 |
| --- | --- | --- | --- | --- |
| 1 | 微控制器 | STM32F103C8T6 | 1 | 控制核心 |
| 2 | 姿态传感器 | MPU6050 | 1 | 测量角度和角速度 |
| 3 | 超声波传感器 | HC-SR04 | 1 | 测量前方障碍物距离 |
| 4 | 直流减速电机 | 带编码器减速电机 | 2 | 驱动轮 |
| 5 | 电机驱动模块 | L298N | 1 | 控制电机正反转 |
| 6 | OLED显示屏 | 0.96寸IIC | 1 | 显示系统状态 |
| 7 | 无线串口模块 | HC-05/HC-06 | 1 | 无线遥控 |
| 8 | 锂电池 | 11.1V 2200mAh | 1 | 提供系统电源 |
| 9 | 车架结构件 | 亚克力板材 | 1套 | 车身支架 |
| 10 | 轮子 | 橡胶轮 | 2 | 提供摩擦力 |

### 各模块电路设计

#### 1. MPU6050姿态传感器连接

MPU6050采用I2C通信协议与STM32F103连接，引脚连接如下：

* MPU6050 VCC → STM32 3.3V
* MPU6050 GND → STM32 GND
* MPU6050 SCL → STM32 PB6 (I2C1\_SCL)
* MPU6050 SDA → STM32 PB7 (I2C1\_SDA)
* MPU6050 INT → STM32 PA5 (外部中断)

通过外部中断引脚，可实现MPU6050测量完成自动通知STM32读取数据，提高系统实时性。

#### 2. 电机驱动电路

采用L298N双H桥驱动模块驱动两个直流减速电机，连接如下：

* 电机A正端 → L298N 输出A+
* 电机A负端 → L298N 输出A-
* 电机B正端 → L298N 输出B+
* 电机B负端 → L298N 输出B-
* L298N ENA → STM32 PA1 (TIM1\_CH1)
* L298N ENB → STM32 PA4 (TIM1\_CH4)
* L298N IN1 → STM32 PB12
* L298N IN2 → STM32 PB13
* L298N IN3 → STM32 PB14
* L298N IN4 → STM32 PB15

通过定时器PWM输出控制电机速度，通过GPIO控制电机方向。

#### 3. 编码器接口

电机编码器采用STM32的定时器编码器模式读取，连接如下：

* 编码器A相位A → STM32 PA0 (TIM2\_CH1)
* 编码器A相位B → STM32 PA1 (TIM2\_CH2)
* 编码器B相位A → STM32 PB6 (TIM4\_CH1)
* 编码器B相位B → STM32 PB7 (TIM4\_CH2)

#### 4. 超声波传感器接口

HC-SR04超声波传感器连接：

* HC-SR04 VCC → STM32 5V
* HC-SR04 GND → STM32 GND
* HC-SR04 Trig → STM32 PA3
* HC-SR04 Echo → STM32 PA2 (外部中断)

通过测量回波信号的持续时间计算障碍物距离。

#### 5. OLED显示屏

OLED显示屏采用I2C通信协议连接：

* OLED VCC → STM32 3.3V
* OLED GND → STM32 GND
* OLED SCL → STM32 PB8 (I2C1\_SCL)
* OLED SDA → STM32 PB9 (I2C1\_SDA)

#### 6. 无线串口模块

蓝牙串口模块连接：

* HC-05 VCC → STM32 5V
* HC-05 GND → STM32 GND
* HC-05 TXD → STM32 PB11 (USART3\_RX)
* HC-05 RXD → STM32 PB10 (USART3\_TX)

### 电源设计

系统采用11.1V锂电池供电，通过稳压电路为各模块提供电源：

* 5V电源：通过LM7805稳压器，为HC-SR04、HC-05等模块供电
* 3.3V电源：通过LM1117-3.3稳压器，为STM32、MPU6050、OLED等模块供电
* 电机电源：直接由锂电池供电，保证足够的驱动能力

## 五、系统软件设计

### 软件架构

自平衡小车的软件系统采用模块化设计，主要包括以下模块：

1. 初始化模块：负责系统外设和参数的初始化
2. 姿态解算模块：获取MPU6050数据并计算姿态角度
3. 平衡控制模块：实现多级PID控制算法
4. 电机驱动模块：控制电机的转速和方向
5. 超声波测距模块：检测前方障碍物距离
6. 显示模块：在OLED上显示系统状态
7. 通信模块：处理无线控制命令

### 代码流程图

1. **主程序流程图**：
   * 系统初始化
   * 外设初始化（MPU6050、OLED、电机、超声波等）
   * 进入主循环
     + 显示系统状态
     + 测量障碍物距离
     + 等待中断触发平衡控制
2. **平衡控制流程图**：
   * 读取编码器和MPU6050数据
   * 检查平衡状态
   * 判断是否执行控制算法
   * 计算速度环输出
   * 计算直立环输出
   * 计算转向环输出
   * 合成最终电机控制量
   * 限幅和输出到电机

### 关键算法实现

#### 1. MPU6050 DMP姿态解算

使用MPU6050内置的DMP（Digital Motion Processor）处理器，通过I2C通信获取Roll、Pitch、Yaw角度，无需在STM32上进行复杂的四元数计算，降低主控制器负担。

// 初始化MPU6050和DMP  
MPU\_Init();  
mpu\_dmp\_init();  
  
// 获取姿态数据  
mpu\_dmp\_get\_data(&pitch, &roll, &yaw);  
MPU\_Get\_Gyroscope(&gyrox, &gyroy, &gyroz);  
MPU\_Get\_Accelerometer(&aacx, &aacy, &aacz);

#### 2. 平衡状态检测

通过比较当前角度与平衡角度（机械中值）的差值，判断小车是否处于平衡状态。

// 检查平衡车是否处于平衡状态  
uint8\_t Is\_Balance(float angle)  
{  
 // 检查角度是否在平衡范围内  
 if(fabs(angle - Med\_Angle) > BALANCE\_ANGLE\_THRESHOLD)  
 {  
 return 0; // 失去平衡  
 }  
 return 1; // 平衡状态  
}

#### 3. 直立环PD控制器

直立环采用PD控制，根据角度偏差和角速度计算输出。

//直立环PD控制器  
//输入：期望角度、真实角度、角速度  
int Vertical(float Med, float Angle, float gyro\_Y)  
{  
 int temp;  
 temp = Vertical\_Kp \* (Angle - Med) + Vertical\_Kd \* gyro\_Y;  
 return temp;  
}

#### 4. 速度环PI控制器

速度环采用PI控制，带低通滤波和积分限幅。

//速度环PI控制器  
//输入：期望速度、左编码器、右编码器  
int Velocity(int Target, int encoder\_L, int encoder\_R)  
{  
 static int Err\_LowOut\_last, Encoder\_S;  
 static float a = 0.7;  
 int Err, Err\_LowOut, temp;  
   
 //1、计算偏差值  
 Err = (encoder\_L + encoder\_R) - Target;  
 //2、低通滤波  
 Err\_LowOut = (1-a) \* Err + a \* Err\_LowOut\_last;  
 Err\_LowOut\_last = Err\_LowOut;  
 //3、积分  
 Encoder\_S += Err\_LowOut;  
 //4、积分限幅  
 Encoder\_S = Encoder\_S > 20000 ? 20000 : (Encoder\_S < (-20000) ? (-20000) : Encoder\_S);  
   
 //5、速度环计算  
 temp = Velocity\_Kp \* Err\_LowOut + Velocity\_Ki \* Encoder\_S;  
 return temp;  
}

#### 5. 转向环PD控制器

转向环采用PD控制，根据目标转向和陀螺仪Z轴角速度计算输出。

//转向环PD控制器  
//输入：角速度、角度值  
int Turn(float gyro\_Z, int Target\_turn)  
{  
 int temp;  
 temp = Turn\_Kp \* Target\_turn + Turn\_Kd \* gyro\_Z;  
 return temp;  
}

#### 6. 电机控制

根据PID控制器输出计算左右电机的PWM值和方向。

void Load(int moto1, int moto2)  
{  
 if(moto1 > 0)  
 {  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_12, GPIO\_PIN\_RESET);  
 }  
 else  
 {  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_12, GPIO\_PIN\_SET);  
 }  
 \_\_HAL\_TIM\_SetCompare(&htim1, TIM\_CHANNEL\_4, abs(moto1));  
   
 if(moto2 > 0)  
 {  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_14, GPIO\_PIN\_SET);  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  
 }  
 else  
 {  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  
 HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_SET);  
 }  
 \_\_HAL\_TIM\_SetCompare(&htim1, TIM\_CHANNEL\_1, abs(moto2));  
}

#### 7. 超声波测距

通过定时器和外部中断测量超声波回波时间，计算距离。

void GET\_Distance(void)  
{  
 HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_3, GPIO\_PIN\_SET);  
 RCCdelay\_us(12);  
 HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  
}  
  
void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  
{  
 if(GPIO\_Pin == GPIO\_PIN\_2)  
 {  
 if(HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_2) == GPIO\_PIN\_SET)  
 {  
 \_\_HAL\_TIM\_SetCounter(&htim3, 0);  
 HAL\_TIM\_Base\_Start(&htim3);  
 }  
 else  
 {  
 HAL\_TIM\_Base\_Stop(&htim3);  
 count = \_\_HAL\_TIM\_GetCounter(&htim3);  
 distance = count \* 0.017;  
 }  
 }  
}

### 控制策略

自平衡小车采用三环PID控制策略：

1. **速度环**：最外环，根据期望速度和实际速度计算期望角度
2. **直立环**：中间环，根据期望角度和实际角度计算平衡控制量
3. **转向环**：辅助环，根据转向指令和陀螺仪数据计算左右轮差速

控制周期为10ms，通过外部中断定时触发，保证实时性。

### 安全保护机制

当小车倾角超过阈值时，自动切断电机电源，防止小车倒下时电机持续工作导致损坏。

// 如果处于不平衡状态，强制速度为0  
if(balance\_state == 0)  
{  
 Target\_Speed = 0;  
 Target\_turn = 0;  
 MOTO1 = 0;  
 MOTO2 = 0;  
 Load(0, 0); // 直接停止电机  
 return; // 不平衡状态下不执行后续控制逻辑  
}

## 六、系统功能测试

### 基础平衡测试

1. **测试目的**：验证小车的基础平衡能力
2. **测试方法**：在平坦地面上启动小车，观察其保持平衡的能力
3. **测试结果**：小车能够稳定保持平衡，偏离平衡位置后能自动回归平衡状态
4. **结果分析**：基础平衡控制算法工作正常，参数调整合理

### 抗干扰能力测试

1. **测试目的**：验证小车抵抗外部干扰的能力
2. **测试方法**：对小车施加轻微外力，观察其恢复平衡的速度和稳定性
3. **测试结果**：小车能够在受到适度干扰后迅速恢复平衡
4. **结果分析**：PID参数调整合理，系统具有良好的抗干扰能力

### 遥控功能测试

1. **测试目的**：验证小车遥控系统的可靠性和响应性
2. **测试方法**：通过无线串口发送前进、后退、左转、右转等指令，观察小车反应
3. **测试结果**：小车能准确响应遥控指令，运动平稳
4. **结果分析**：遥控系统工作正常，与平衡控制系统配合良好

### 避障功能测试

1. **测试目的**：验证小车的障碍物检测和避障能力
2. **测试方法**：在小车前方放置障碍物，观察小车的反应
3. **测试结果**：当障碍物距离小于50cm时，小车会自动减速
4. **结果分析**：超声波测距系统工作正常，避障逻辑有效

### 失控保护测试

1. **测试目的**：验证小车的失控保护机制
2. **测试方法**：人为使小车倾角过大，模拟失控状态
3. **测试结果**：当倾角超过40度时，电机自动停止，防止继续运动
4. **结果分析**：安全保护机制有效，能防止意外损坏

### 电池续航测试

1. **测试目的**：测试小车的电池续航时间
2. **测试方法**：在满电状态下持续运行小车，记录可工作时间
3. **测试结果**：在平衡状态下可持续工作约2小时
4. **结果分析**：电源管理系统工作正常，续航时间满足需求

## 七、结论与展望

### 结论

通过本次自平衡两轮小车的设计和实现，成功完成了以下目标：

1. 基于STM32F103C8T6微控制器设计并实现了自平衡两轮小车的硬件系统
2. 采用MPU6050 DMP技术实现了精确的姿态检测
3. 设计并实现了多级PID控制算法，使小车能够稳定平衡
4. 实现了超声波避障功能，增强了小车的环境感知能力
5. 通过无线通信模块实现了远程控制功能
6. 设计了安全保护机制，提高了系统的可靠性

系统测试结果表明，所设计的自平衡两轮小车能够满足预期的功能要求，具有良好的平衡性能和操控性能。

### 存在的问题

1. PID参数调整较为复杂，需要多次试验才能获得最佳参数
2. 系统对电池电量变化较为敏感，电量不足时平衡性能下降
3. 超声波传感器在某些特殊表面上测距不准确
4. 小车在不平坦地面上的平衡能力有限

### 改进展望

1. 考虑采用更先进的控制算法，如模糊PID或卡尔曼滤波，提高控制精度
2. 增加电池电量检测和提示功能，优化低电量下的控制策略
3. 增加多传感器融合，如增加红外传感器辅助超声波，提高障碍物检测能力
4. 设计更智能的控制APP，增加参数调整和数据显示功能
5. 考虑加入机器视觉系统，实现更高级的环境感知和自主导航功能
6. 优化机械结构设计，提高系统抗干扰能力和适应复杂地形能力

## 八、学习体会

通过本次自平衡两轮小