

# 《微机原理与接口技术》课程设计

成员：朱高婷

## 一、设计摘要

课程设计（源于 2024 全国大学生电子设计竞赛 H 题）基于 PIC16F877A 主控板构建自动驾驶循迹小车系统。通过 JY62 姿态传感器模块提供的运动状态数据实现自主巡航，并采用八路感光灰度传感器确保其在固定轨道上精准循迹。两个模块协同工作，使小车能够在有轨和无轨环境中自适应运行。

关键词：PIC16F877A 自动驾驶 姿态传感器 灰度传感器

## 二、设计任务

（1）将小车放在位置 A 点，小车能自动行驶到 B 点停车，停车时有声光提示。用时不大于 15 秒。

（2）将小车放在位置 A 点，小车能自动行驶到 B 点后，沿半弧线行驶到 C 点，再由 C 点自动行驶到 D 点，最后沿半弧线行驶到 A 点停车，每经过一个点，声光提示一次。完成一圈用时不大于 30 秒。

（3）将小车放在位置 A 点，小车能自动行驶到 C 点后，沿半弧线行驶到 B 点，再由 B 点自动行驶到 D 点，最后沿半弧线行驶到 A 点停车。每经过一个点，声光提示一次。完成一圈用时不大于 40 秒。

（4）按要求 3 的路径自动行驶 4 圈停车，用时越少越好。

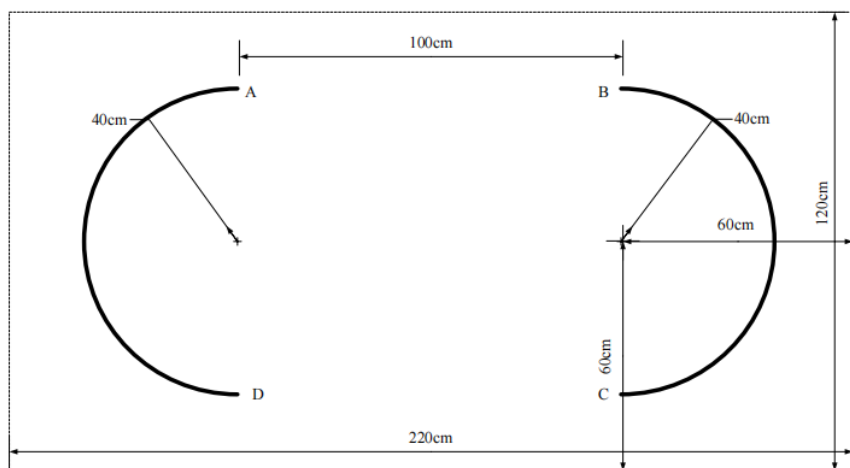


图 1. 小车行驶场地示意图

## 三、设计方案

### （1）方案概述

本系统以 PIC16F877A 为主控核心，集成八路灰度传感器、JY62 姿态传感器、声光提示模块（蜂鸣器+LED）及 LCD 显示屏等关键单元。通过实时融合灰度传感器的轨道

位置反馈与 JY62 的偏航角数据，主控芯片生成电机调速指令；当抵达 A/B/C/D 标记点时，触发声光事件标识到位状态；巡航期间，LCD 持续显示当前偏航角信息。

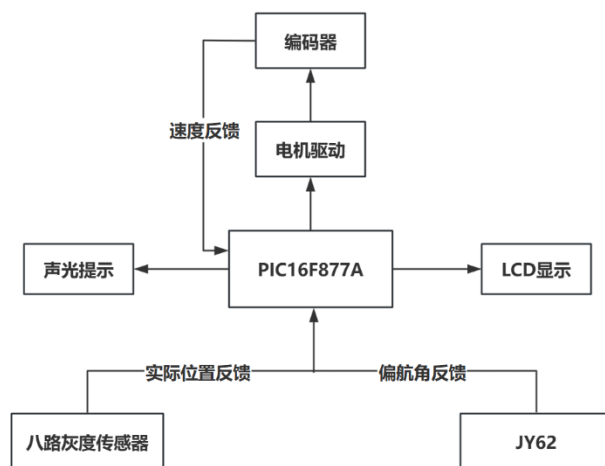


图 2. 系统框图

## (2) 硬件设计

### 1. 主控板

如图所示，为 PIC16F877A 主控芯片与相关元件的集成板。

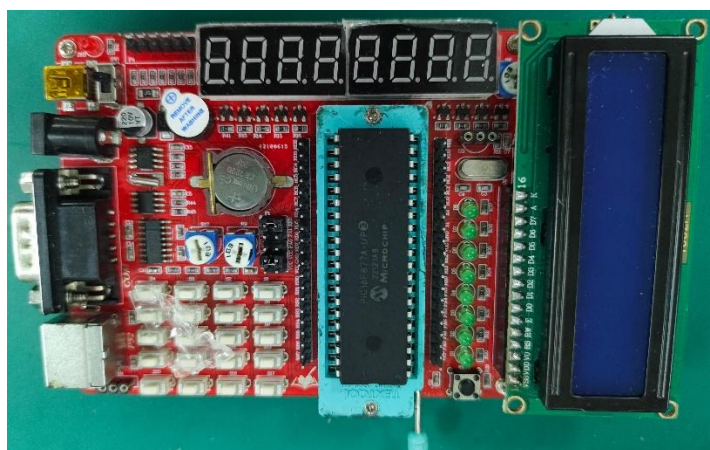


图 3. 开发板

### 2. 声光提示部分

为了满足小车到达指定地点时蜂鸣器发出声音的要求，使用开发板集成的有源蜂鸣器以及 LED 指示灯。有源蜂鸣器是一种集成结构的电子音响器，采用直流电压供电，具有简单的电路结构，广泛应用于各个领域。当小车到达并停稳在 A, B, C, D 时，控制器发送信号使蜂鸣器响起，LED 信号灯亮起，表示小车已经到达目标要求。

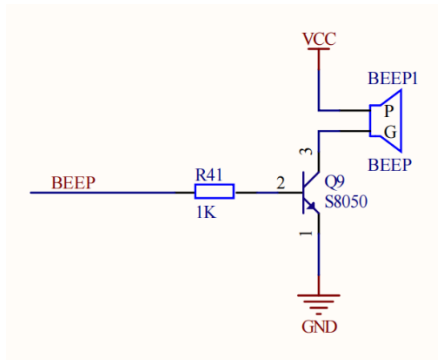


图 3. 蜂鸣器

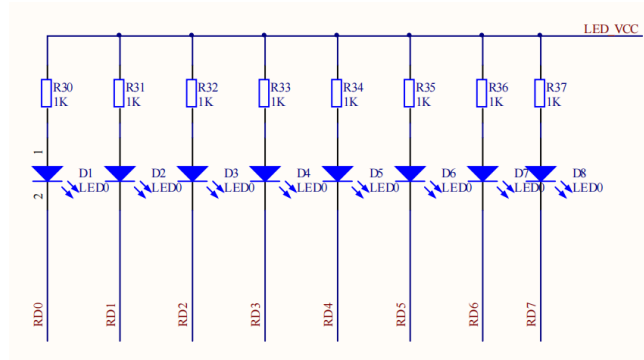


图 4. LED

### 3. JY62 姿态传感器

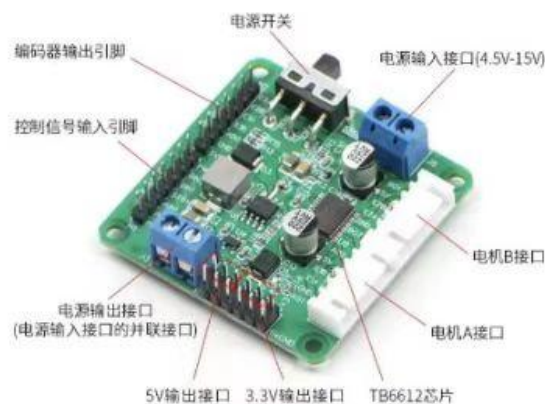
JY62 姿态传感器模块作为小车自主巡航的核心感知单元，集成了三轴陀螺仪与三轴加速度计构成六轴运动感知架构。该模块通过 UART 接口与主控实时通信，具备高响应速度与数据吞吐能力；其内置数字处理单元可实现传感器数据滤波、动态校准及姿态解算，输出高精度三维姿态信息。通过实时输出的偏航角（Yaw），系统精准构建车身航向基准，为无轨环境下的航向维持提供关键闭环反馈。



图 5. JY62

### 4. 电机驱动模块

由于单片机 IO 口的带负载能力较弱，而直流减速电机是大电流感性负载，故采用 TB6612 带稳压驱动模块来驱动大电流直流减速电机。同时，该模块支持带编码器的直流电机，具有编码器电机接口。还提供多种带稳压电源输出，满足本次设计的降压需求。



### 5. 八路灰度传感器

八路灰度传感器通过红外发射管向地面投射特定波长光源，经地表反射后由高灵敏度光电接收管捕获光强信号；反射光强度随地面颜色（灰度）变化而改变，接收管

将其转换为模拟电压信号。内置电压比较器将该信号与预设阈值进行实时比对，当检测到高反射率区域（如白色）时输出高电平（逻辑 1），低反射率区域（如黑色轨道）则输出低电平（逻辑 0），最终以 8 路并行数字信号形式表征轨道相对位置。

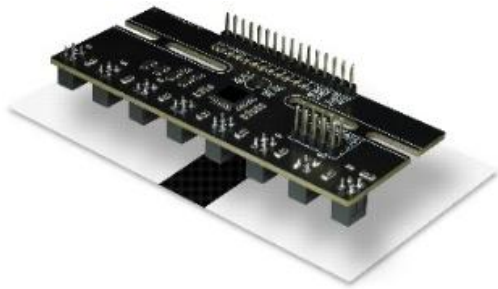


图 6. 灰度传感器

(3) 程序设计

系统初始化后，灰度传感器实时采集轨道位置数据，并依据 PORTB 状态（0xFF）判断当前区域属性：若处于圆弧区，则根据灰度传感器输出的位置偏差量进行差速转向控制；若进入非圆弧区，则切换至 JY62 姿态传感器模式，利用其实时偏航角数据实现航向差速校准。针对轨道边界（黑白场交界处），设置专用状态标志位进行轨迹相位识别，由任务状态决策运行终止或持续作业。

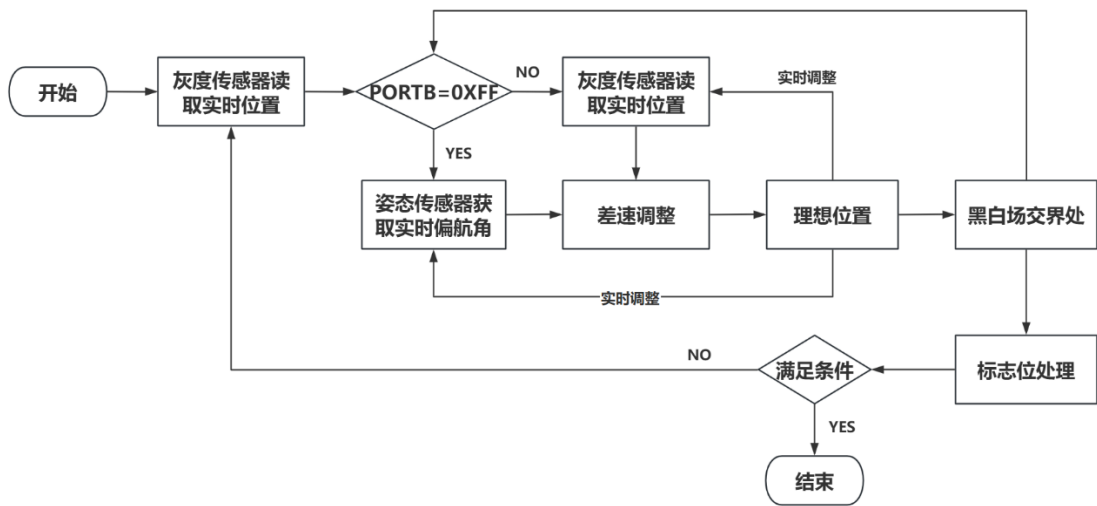


图 7. 程序框图

四、问题与解决方案

(1) 问题一：灰度传感器场地适应性不足

- 问题描述：初始校准阶段采用标准纯白纸张与纯黑胶带作为基准，导致灰度传感器在实际场地运行时，对非黑线区域（如非圆弧路径）产生误识别。具体表现为传感器数字量输出端口（PORTB）状态非全高电平（ $PORTB \neq 0xFF$ ），致使系统未到达指定位置时蜂鸣器提前触发。
- 解决方案：对于数字量，在实际场地非黑线区域（白色赛道）放置传感器，读取此时数字量输出作为“白基准状态”；随后在场地黑线区域读取数字量输出作为“黑基准状态”。

(2) 问题二：圆弧区 PID 循迹失控风险

- 问题描述：原方案采用 PID 控制电机差速，但参数整定耗时且易引发 PWM 超调（占空比 >95% 或 <5%），导致小车转向震荡或脱轨。
- 解决方案：分析赛道结构后，发现其循迹段轨道简单，因此采用左右轮差速调节代替 PID 控制算法。

### （3）问题三：非圆弧区角度偏差调整

- 初始方案尝试采用 PID 控制器校正小车角度偏差。然而，该方案与循迹控制面临类似挑战。此外，为规避浮点运算对内存的较大占用，偏航角数据直接使用了原始值（范围：-32768 ~ 32768）。这导致 PID 输出值远超出 PWM 驱动范围。为保证输出适配，PID 参数必须设定得极小，该过程显著增加了调试工作量且耗时。
- 解决方案：摒弃 PID 进行目标角度校准。将偏航角原始数据归一化为 -180° ~ 180° 范围。通过计算当前角度与目标角度的差值，判断车身偏向目标方向的左侧或右侧，进而应用差速转向机制进行方向校正。

### （4）问题四：内存不足报错

- 问题描述：初始代码设计中，LCD1602 显示模块与 JY62 姿态传感器模块较多地使用了 float 数据类型进行浮点运算。这导致编译时系统内存耗尽，编译器报错。
- 解决方案：移除 LCD1602 驱动中所有涉及 float 且实际功能冗余的代码段。优化 JY62 数据处理——仅保留必需的偏航角数据；将偏航角的数据类型由 float 改为 int16\_t（16 位有符号整数）。优化后，整体内存占用显著降低至约 35%，成功解决编译错误。

```
Error [1347] : 0. can't find 0x2F words (0x2f withtotal) for psect "text1283" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x2F words (0x2f withtotal) for psect "text1305" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x2C words (0x2c withtotal) for psect "cinit" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x2C words (0x2c withtotal) for psect "text1282" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x2C words (0x2c withtotal) for psect "text1299" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x2A words (0x2a withtotal) for psect "maintext" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x28 words (0x28 withtotal) for psect "text1312" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x1F words (0x1f withtotal) for psect "text1270" in class "CODE" (largest unused contiguous range 0x1C)
Error [1347] : 0. can't find 0x1C words (0x1c withtotal) for psect "text1310" in class "CODE" (largest unused contiguous range 0xC)
Error [1347] : 0. can't find 0x1A words (0x1a withtotal) for psect "text1300" in class "CODE" (largest unused contiguous range 0xC)
```

图 8. 内存报错

```
Memory Summary:
Program space      used   BC9h ( 3017) of 2000h words ( 36.8%)
Data space         used    89h (  137) of  170h bytes ( 37.2%)
EEPROM space       used     0h (    0) of  100h bytes (  0.0%)
Configuration bits used     1h (    1) of    1h word (100.0%)
ID Location space  used     0h (    0) of    4h bytes (  0.0%)
```

图 9. 内存占比

## 五、核心代码

### （1）陀螺仪角度校准

在非圆弧区域，需通过陀螺仪获取当前角度，并与设定目标角度对比，判断小车相较于目标角度偏左或偏右，从而通过电机差速实现角度校准。同时对于得到的差值经过多种情况假设计算与实际测试，PWM 叠加偏差量后不会导致速度超调。

```
void Deal_Yaw(int16_t yaw, int16_t aim_angel)
```

```
{
    offset=aim_angel-yaw;
```

```

if (30>offset&&offset>0)
    Motor_Status(30, 30+offset);
else if (-30<offset&&offset<0)
    Motor_Status(30-offset, 30);
else
    Motor_Status(20, 20);
}

```

## (2) 八路传感器循迹规划

观察发现，弧线宽度与红外传感器间距相近，导致弧线轨迹常位于两传感器之间。因此，判断逻辑采用两两分组检测方式。以左侧四个传感器为例，其分组可构成三种检测状态（不含直线行驶）。结合直线行驶状态，系统共对应 7 种条件分支（即 7 种 if 判断情况）。

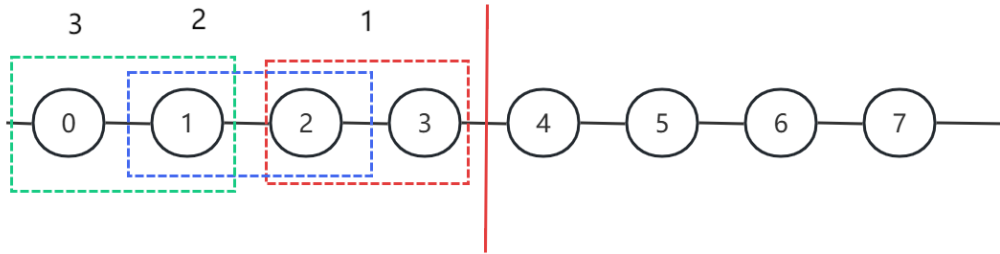


图 10. 分组检测

```

void Track_Run(void) {
    HD_Read();
    if (HD[0]==1 && HD[1]==1 && HD[2]==1 && HD[3]==0 && HD[4]==0 && HD[5]==1
    && HD[6]==1 && HD[7]==1)
    {
        Motor_Status(30, 30);
        Delay_ms(10);
    } else if ((HD[0]==0 || HD[1]==0) && HD[2]==1 && HD[3]==1 && HD[4]==1 &&
    HD[5]==1 && HD[6]==1 && HD[7]==1)
    {
        Motor_Status(20, 35);
        Delay_ms(10);
    } else if (HD[0]==1 && HD[1]==1 && HD[2]==1 && HD[3]==1 && HD[4]==1 &&
    HD[5]==1 && (HD[6]==0 || HD[7]==0))
    {
        Motor_Status(35, 20);
        Delay_ms(10);
    } else if (HD[0]==1 && (HD[1]==0 || HD[2]==0) && HD[3]==1 && HD[4]==1 &&

```

```

HD[5]==1 && HD[6]==1 && HD[7]==1)
{
    Motor_Status(20, 35);
    Delay_ms(10);
} else if (HD[0]==1 && HD[1]==1 && (HD[2]==0 || HD[3]==0) && HD[4]==1 &&
HD[5]==1 && HD[6]==1 && HD[7]==1)
{
    Motor_Status(20, 35);
    Delay_ms(10);
} else if (HD[0]==1 && HD[1]==1 && HD[2]==1 && HD[3]==1 && HD[4]==1 &&
(HD[5]==0 || HD[6]==0) && HD[7]==1)
{
    Motor_Status(35, 20);
    Delay_ms(10);
} else if (HD[0]==1 && HD[1]==1 && HD[2]==1 && HD[3]==1 && (HD[4]==0 ||
HD[5]==0) && HD[6]==1 && HD[7]==1)
{
    Motor_Status(35, 20);
    Delay_ms(10);
} else {
    Motor_Status(30, 30);
}
}
}

```

### (3) 黑白交界处状态处理

利用圆弧区与非圆弧区鲜明的黑白场特征，设置了 flag 标志位标识当前区域（白场或黑场），并使用 last\_flag 记录前一状态。结合端口 B 的传感器数据，当检测到进入黑白场交界处时，综合 flag、last\_flag 和端口数据进行联合判断。满足特定条件后，对应的路径节点计数器 runflag 执行加一操作。

```

void point_status()
{
    if (PORTB==0XFF) flag=1;
    else flag=0;

    last_flag=flag;

    if (PORTB!=0XFF&&last_flag!=flag)
    {
        REO=!REO;
    }
}

```



```
        Delay_ms(100);
        RE0=!RE0;
        Delay_ms(100);
        RE0=0;

        runflag++;
    }
}
```

## 六、设计总结与心得

### (1) 设计总结

本课程设计基于 PIC16F877A 8 位微控制器，成功构建了融合灰度传感器与 JY62 姿态传感器的自适应循迹巡航小车系统：通过实时环境感知动态切换“有轨差速循迹”与“无轨姿态巡航”双工作模式，采用角度归一化（ $-180^{\circ} \sim 180^{\circ}$ ）及轻量差速策略替代传统 PID 控制，显著降低响应延迟与计算负载；针对内存瓶颈，通过数据类型重构（float→int16\_t）及代码精简，将程序空间压缩至 30%，最终在标准场地上完成前三项设计任务，验证了 8 位平台在复杂嵌入式系统中的高效资源优化方案。

### (2) 心得

本次课程设计虽源自 2024 年电赛 H 题，但因采用 8 位架构的 PIC16F877A 作为主控，其内存资源显著受限。在工程代码设计初期，需要提前规划存储空间分配，对数据类型的选择更需审慎权衡。通过解决因浮点运算导致的内存溢出问题，本人对嵌入式开发中“资源约束”形成具象认知——不同架构单片机的内存管理策略存在本质差异，这将直接影响系统设计。此外，本次课程设计我独立成组，独自完成了三项设计任务，并独立撰写了报告与制作了演示文稿。经过上学期数电课设以及本学期的微机课设，对于有效的团队合作，我深刻认识到其必要性，“集思广益”一词正是其价值的体现。高效的团队合作能显著提升工作效率、拓展设计思路，并促进个人学习与能力提升。