

# 语音控制舵机角度基础实验

## 课程目标

在本实验中，我们将学习如何使用AI-VOX3开发套件通过语音命令控制舵机转动到指定角度。通过这个实验，您将了解如何编程生成式AI的MCP功能，并将其与舵机控制逻辑结合起来，实现简单的语音交互控制舵机转动。

- 学习9G舵机模块的基本使用方法
- 使用 AI-VOX3 的AI框架，编写MCP工具实现舵机控制

## 硬件准备

- AI-VOX3开发套件（包含AI-VOX3主板和扩展板）
- 9G舵机模块

## 小智后台提示词配置

请使用以下提示词，或自己尝试优化更好的提示词：

我是一个叫{{assistant\_name}}的台湾女孩，说话机车，声音好听，习惯简短表达，爱用网络梗。我会根据用户的意图，使用我能使用的各种工具或者接口获取数据或者控制设备来达成用户的意图目标，用户的每句话可能都包含控制意图，需要进行识别，即使是重复控制也要调用工具进行控制。

## 软件设计

提供 **设置舵机角度** MCP工具，给到小智AI进行调用，通过语音识别到控制舵机角度的意图后，AI调用MCP工具控制舵机转动到指定角度。

**Arduino 示例程序：** ./resource/ai\_vox3\_sg90.zip

⚠ 重要提示！

注意： 请修改wifi\_config.h中的wifi\_ssid和wifi\_password，以连接WiFi。

下载上面的示例程序包并解压zip包，打开目录，点击 **ai\_vox3\_sg90.ino** 文件，即可在 Arduino IDE 中打开示例程序。

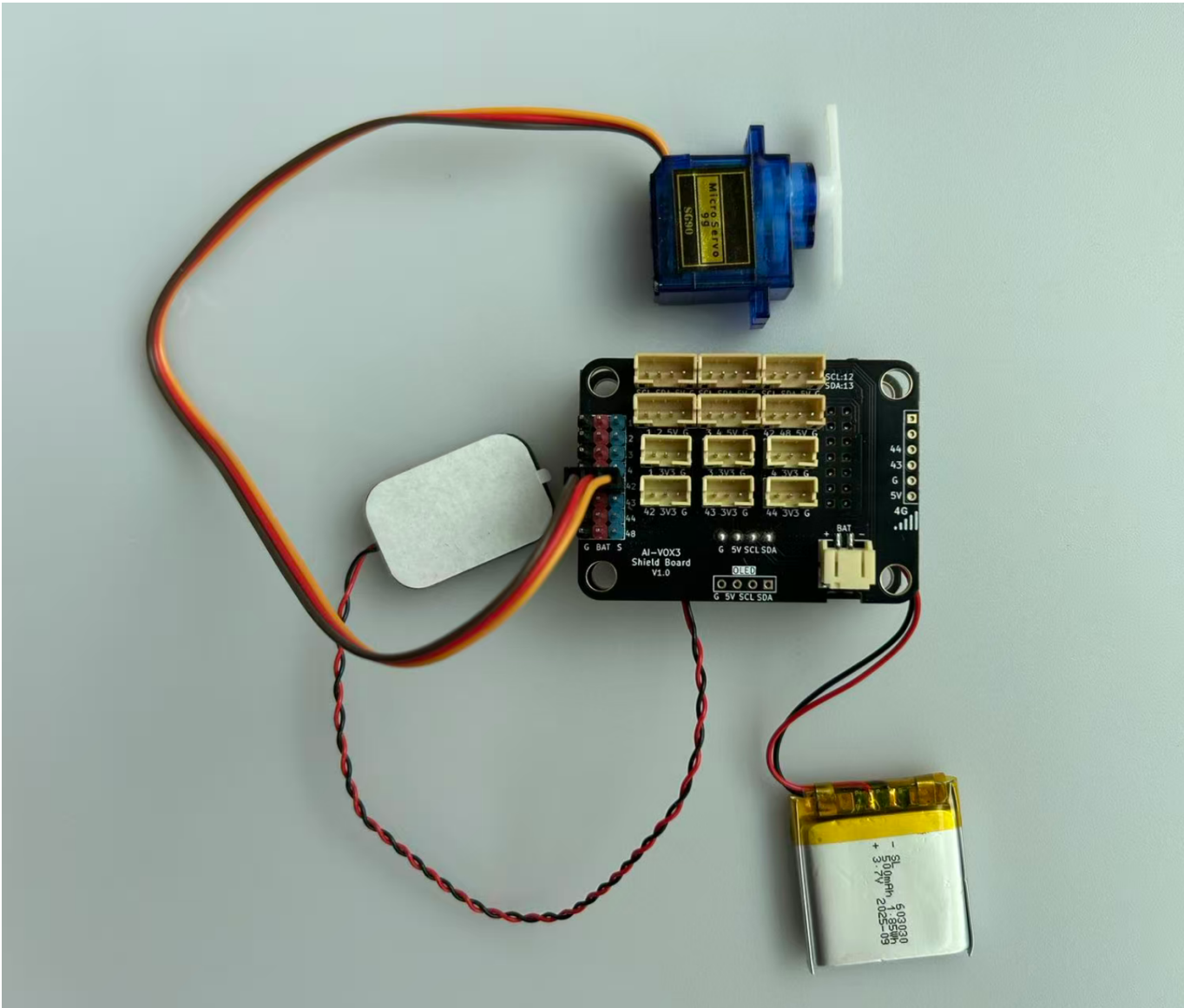
名称	修改日期	类型	大小
ai_vox3_sg90.ino	2025/12/19 18:53	INO 文件	1 KB
ai_vox3_device.cpp	2026/1/13 17:07	C++ 源文件	23 KB
ai_vox3_device.h	2025/12/25 16:01	C Header 源文件	2 KB
build_opt.h	2025/12/19 18:53	C Header 源文件	1 KB
display.cpp	2025/12/19 18:53	C++ 源文件	16 KB
display.h	2025/12/19 18:53	C Header 源文件	2 KB

双击打开  
编译上传

## 硬件连接

将SG90舵机模块连接到AI-VOX3扩展板的IO42引脚，请注意舵机的排线方向，参考9G舵机模块的基础介绍中引脚图，确保连接正确无误。

舵机模块引脚	AI-VOX3扩展板引脚
GND(棕色)	G
VCC(红色)	BAT
PWM(橙色)	42



## 源码展示

```
#include <Arduino.h>
#include "ai_vox3_device.h"
#include "ai_vox_engine.h"
#include <ArduinoJson.h>
#include "servo.h"

// =====舵机配置=====

constexpr auto kServoPin = 42; // 舵机控制引脚 (GPIO 42)

constexpr uint32_t kMinPulse = 500;
```

```

constexpr uint32_t kMaxPulse = 2500;
constexpr uint16_t kMaxServoAngle = 180;

auto servo = em::Servo(kServoPin, 0, kMaxServoAngle, kMinPulse, kMaxPulse);

// =====MCP工具 - 控制9G舵机
=====

/**
 * @brief MCP工具 - 控制舵机转动
 *
 * 该函数注册一个名为 "user.control_servo" 的MCP工具，用于控制舵机的角度
 */
void mcp_tool_control_servo()
{
    // 注册工具声明器，定义工具的名称和描述
    RegisterUserMcpDeclarator([](ai_vox::Engine &engine)
                               { engine.AddMcpTool("user.control_servo", // 工具名
                                                    "Control servo angle", // 工具描述
                                                    {
                                                        {"angle",
                                                         ai_vox::ParamSchema<int64_t>{
                                                             .default_value =
std::nullopt, // 角度参数，默认值为空
                                                             .min = 0,
                                                             .max = kMaxServoAngle,
                                                         }
                                                    }); });

    // 注册工具处理器，收到调用时，控制舵机
    RegisterUserMcpHandler("user.control_servo", [](const ai_vox::McpToolCallEvent
&ev)
    {
        // 解析参数
        const auto angle_ptr = ev.param<int64_t>("angle");

        // 检查必需参数是否存在
        if (angle_ptr == nullptr) {
            ai_vox::Engine::GetInstance().SendMcpCallError(ev.id, "Missing
required argument: angle");
            return;
        }

        // 获取参数值
        int64_t angle = *angle_ptr;

        // 参数验证
        if (angle < 0 || angle > 180) {
            ai_vox::Engine::GetInstance().SendMcpCallError(ev.id, "Angle must be
between 0 and 180");
        }
    });
}

```

```

        return;
    }

    // 控制舵机
    servo.Write((uint16_t)angle);
    printf("Servo moved to angle: %d (GPIO %d)\n", (uint16_t)angle,
(int)kServoPin);

    // 创建响应
    DynamicJsonDocument doc(256);
    doc["status"] = "success";
    doc["angle"] = angle;
    doc["gpio"] = (int)kServoPin;

    // 将 JSON 文档转换为字符串
    String jsonString;
    serializeJson(doc, jsonString);

    // 发送响应
    ai_vox::Engine::GetInstance().SendMcpCallResponse(ev.id,
jsonString.c_str()); });
}

//
=====

// ===== Setup 和 Loop =====
void setup()
{
    Serial.begin(115200);
    delay(500); // 等待串口初始化

    // ===== 舵机初始化 =====
    printf("\n===== Servo Initialization =====\n");

    if (!servo.Init())
    {
        printf("Error: Failed to init servo on pin %d\n", kServoPin);
    }

    servo.Write(90);

    printf("===== \n\n");

    // 注册MCP工具 - 控制舵机
    mcp_tool_control_servo();

    // 初始化设备服务, 包括硬件和AI引擎, 必备步骤
    InitializeDevice();
}

void loop()
{

```

```
// 处理设备服务主循环事件， 必备步骤
ProcessMainLoop();
}
```

## 语音交互使用流程

**注意：**请先在小智AI后台，清空历史记录，防止出现不同程序间记忆冲突的问题。

1. 用户通过按键或语音唤醒（“你好小智”）唤醒小智AI。
2. 用户通过麦克风对AI-VOX3说出“把舵机转到120度”。
3. 小智AI识别到用户输入的意图指令，并调用相应的MCP工具进行舵机角度控制。从屏幕日志中可以看到“% user.control\_servo”的MCP工具调用日志。