

## 《八年级第五单元第 21 课-文物环境恒控系统》学生学习手册

学校名称：\_\_\_\_\_ 班级：\_\_\_\_\_ 姓名：\_\_\_\_\_

### 模块一：课程导入

#### 真实情境

想象一下，你是一位小小策展人，正在为一个新开设的“多功能特展厅”做准备。这个展厅非常繁忙，本周要展出珍贵的《宋代山水画》，下周就要换成《商周青铜器》。

但你很快发现了一个大难题：

- ✧ **字画**最怕强光，光照强了会褪色，对温度和湿度倒不那么敏感。
- ✧ **青铜器**不怕光，但最怕高温和潮湿，环境太潮湿或温度过高容易生锈。

馆长不希望每次更换展品都把整个展柜和里面的设备全部换掉。那么，我们能否设计一个“智能环境控制系统”，让它足够聪明，能够根据展品的不同，自动切换保护模式呢？

#### 核心问题

我们能否利用物联网技术，打造一个可切换模式的文物微环境恒控系统，实现对不同文物(如字画、青铜器)的精准、自动化保护？

#### 学习目标

通过本课程的学习，你将能够：

1. 理解物联网在文物保护领域的应用价值，体验“感知-判断-反馈”的自动化控制逻辑。

2. 掌握传感器的基本工作原理，并学会使用它们采集环境数据。
3. 理解“阈值”和“条件判断”在自动化控制中的核心作用。
4. 学会通过按键实现多模式切换的交互设计。
5. 设计并实现一个具备数据采集、模式切换、自动判断和声光报警功能的物联网应用。

## 模块二：概念学习

### 1. 核心元件: 传感器 (Sensor)

**想一想：**你家里的空调是如何知道房间温度，并自动调节冷风的？你的手机是如何知道屏幕应该横着还是竖着显示的。

这都归功于**传感器**！它就像我们身体的“感官”。

1. **眼睛**能感知**光线**的明暗。
2. **皮肤**能感知**温度**的冷热。
3. **耳朵**能感知**声音**的大小。

在物联网世界里，传感器就是设备的“五官”，帮助它感知周围的环境。在这个项目中，我们将用到：

1. **光照传感器**：测量光线强度，就像设备的“眼睛”。
2. **温度传感器**：测量环境温度，就像设备的“皮肤”。



图1：传感器类比图

## 2. 核心逻辑: 阈值 (Threshold) 与条件判断

想一想：你早上设置了 7:00 的闹钟，闹钟是如何知道该在 7:00 准时响铃，而不是 6:59 或 7:01 呢？

因为 **7:00** 就是你设定的一个“警戒线”，我们称之为**阈值**。当时间达到这个阈值时，闹钟就会执行“响铃”这个动作。

在文物保护中，阈值就是文物安全的环境标准：

1. 字画的保护阈值是：光照强度  $< 100$
2. 青铜器的保护阈值是：温度  $< 30^{\circ}\text{C}$

在编程中，我们使用 **如果-执行** 模块 来检查是否突破了这条“警戒线”。

## 3. 核心思想：“感知-判断-反馈”闭环

想一想：在一个炎热的夏天，你在太阳下跑步，身体会发生什么变化？

1. **感知 (Perceive):** 你的皮肤(传感器)感觉到环境很**热**，身体开始大量出汗。
2. **判断 (Judge):** 你的大脑(处理器)判断：“太热了，体温可能过高，需要降温!”。
3. **反馈 (Feedback):** 你会立即行动，比如跑到阴凉处休息，或者喝水降温(执行器动作)。

这就是一个完整的自动化控制闭环。我们的文物保护系统也是完全一样的逻辑：

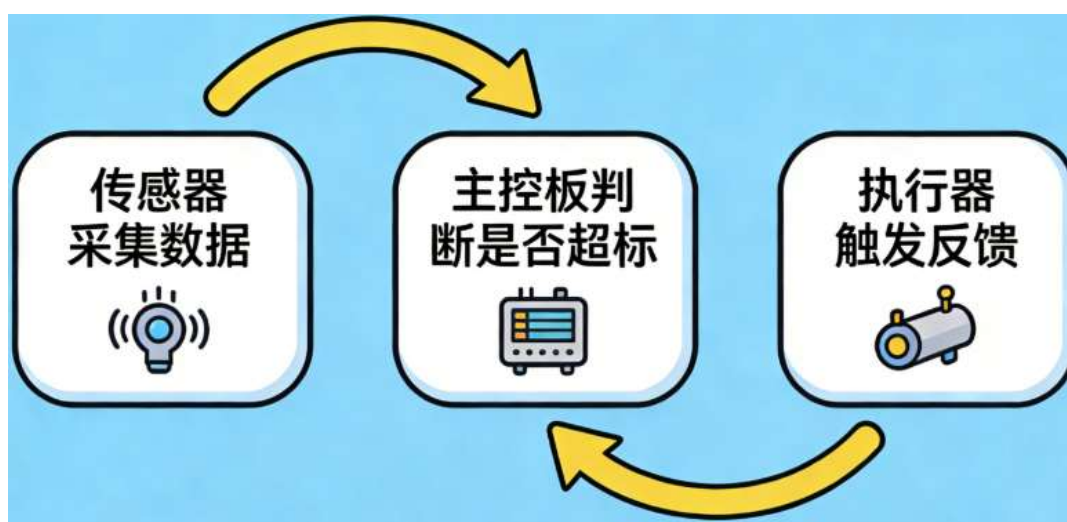


图 2: 控制闭环图

### 模块三：逻辑分析

#### 1. 系统功能拆解

一个完整的系统可以分为输入、处理、输出三个部分。让我们一起来分析一下我们的“文物环境恒控系统”吧！

系统部分	具体功能	实现方式
输入 (Input)	1. 感知环境数据	1. 使用光 <b>传感器</b> 和 <b>温度传感器</b>
	2. 接收用户切换模式的指令	2. 检测 <b>B1/B2 按键</b> 是否被按下
处理 (Process)	核心控制逻辑	1. 根据按键指令，切换 <b>工作模式</b> (字画/青铜)
		2. 在当前模式下，判断对应环境数据 <b>是否超过阈值</b> 3. 将采集到的数据和报警状态通过 <b>MQTT</b> 上传到云平台
输出 (Output)	向用户和云端反馈状态	1. 在点阵屏上显示当前模式图标
		2. 在 OLED 屏上实时显示环境数据 3. 当超标时，通过 LED 灯和蜂鸣器发出声光报警 4. 在物联网平台上查看历史数据和设备状态

2. 逻辑可视化：数据流图与状态图

画一画：请根据上面的功能拆解，将数据流动的路径用箭头连接起来。

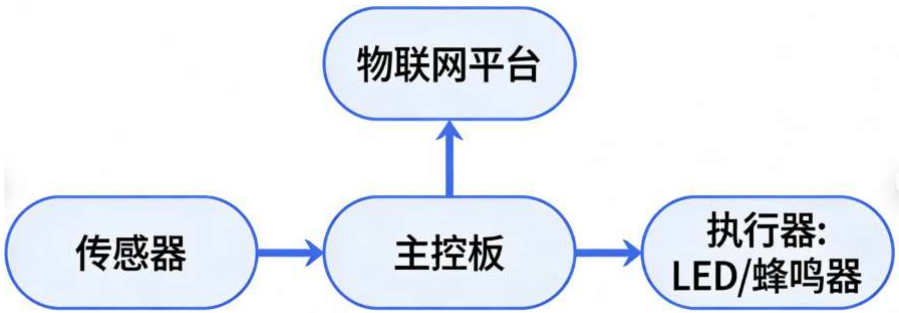


图 3: 数据流动路径图

你的数据流图：

系统几种“状态”？

我们的系统有两个主要的工作模式，每个模式下又有两种状态。我们可以用状态图来清晰地描述它们之间是如何切换的。

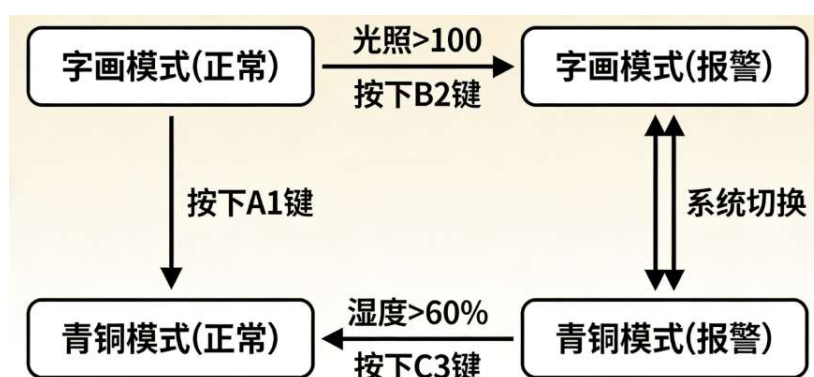


图4：系统工作状态图

看图说话：

1. 智系统的初始状态是什么？
2. 当什么事件发生时，系统会从“字画模式”切换到“青铜模式”？
3. 在“青铜模式”下，发生什么情况会进入“报警”状态？
4. 如何从“报警”状态恢复到“正常”状态？

## 模块四：算法设计

让我们用伪代码来梳理整个程序的完整逻辑。

**读一读，理一理：**下面的伪代码描述了整个流程，请重点理解程序是如何实现模式切换和条件判断的。

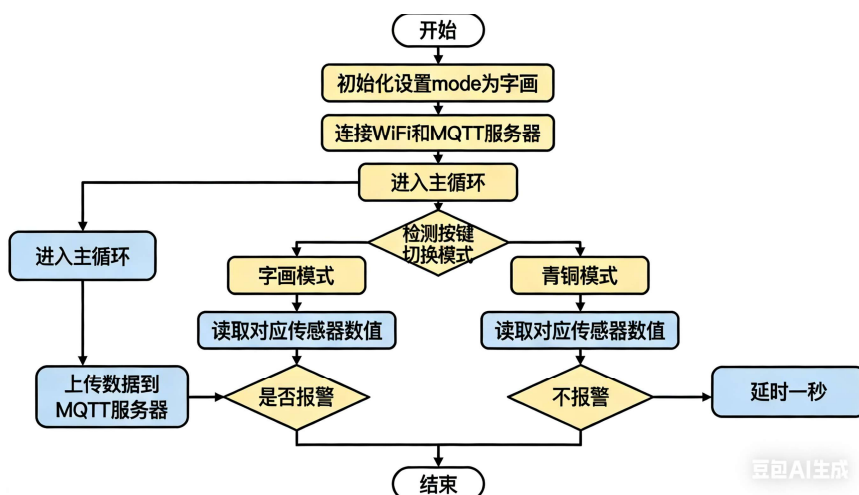


图3：代码实现流程图

## 模块五：代码实现

我们将通过四个核心任务，一步步构建出功能强大的“文物环境恒控系统”。

### 任务 1：连接网络与云平台

**目标：**让你的开发板成功连接到 Wi-Fi 和 MQTT 服务器。



**动手试一试：**

1. 将你的 Wi-Fi 和 MQTT 账户信息填入代码中。
2. 修改并运行代码，使得 OLED 屏幕上依次显示 “WiFi Connected!” 和 “MQTT Connected!”。
3. 如果任何一步连接失败，请仔细检查你填写的配置信息是否正确。

## 任务 2：采集环境数据

**目标：**学会使用光传感器和温度传感器，并将读取到的数据显示在 OLED 屏幕上。

**动手试一试：**

1. 在编程界面，找到**光传感器**和**温度传感器**的调用模块。
2. 编写代码，在循环中不断读取两个传感器的数值。
3. 使用 **打印模块** 将数值打印到电脑端的控制台，检查是否读取成功。
4. 在 OLED 屏幕上显示: Light: [光照值] 和 Temp: [温度值]。
5. 用手遮挡光传感器或用手捂住温度传感器，观察屏幕上的数值变化

## 任务 3：设置阈值，实现条件判断

**目标：**使用 **如果-执行** 模块，完成代码编写：当环境数据超标时，在屏幕上显示警告信息。

**注意：**不同的文物类型需要使用不同的阈值测量方式哦！那么应该如何判断呢？首先判断什么？





动手试一试：

1. 在任务 2 的代码基础上，增加 **如果-执行** 条件判断。
2. 当光照强度 > 100 时，在屏幕上额外显示一行 **光照过强!**。
3. 当温度 > 30 时，在屏幕上额外显示一行 **温度过高!**。
4. 用手电筒照射光传感器，观察屏幕是否按预期显示警告。

#### 任务 4：实现模式切换

**目标：**通过**按键 B1 和 B2**，切换“字画”和“青铜”两种模式，并在点阵屏上显示不同图标。



动手试一试：

1. 在程序开头增加变量：light、temp、mode，并设置初始值。
2. 在主循环中，检测 B1 和 B2 按键是否被按下。
3. 如果 B1 被按下，将 mode 变量的值改为 '字画'，并在点阵屏上显示文字。
4. 如果 B2 被按下，将 mode 变量的值改为 '青铜'，并在点阵屏上显示文字。

## 模块六：调试指导

程序出错了？别担心，这是每个程序员的必经之路！学会调试，是比学会写代码更重要的能力。

技巧 1：“分段测试”与“最小化原则”

千万不要一次性写完几百行代码再运行!这就像盖楼，没人会把所有砖头都堆好了才开始砌墙。

- 1. 先测传感器: 只写读取和显示传感器的代码，确保数据无误。
- 2. 再测判断: 加入条件判断模块，用手电筒和手捂热模拟超标，看程序反应对不对。
- 3. 然后测按键: 只写按键切换模式和显示文字的代码，看交互是否正常。
- 4. 最后测反馈: 加入 LED 和蜂鸣器，看报警是否按预期触发。

当发现问题时，你就知道问题一定出在你刚刚添加的那一小段代码里，排查范围大大缩小!

技巧 2：利用好电脑端的“眼睛”——打印模块

硬件上的屏幕显示信息有限，但电脑端的控制台是你最强大的“监视器”。在代码的关键位置加入 打印 模块，可以帮你洞察程序的每一步动态。

技巧 3：常见问题排查清单

错误现象	可能原因	解决方法
程序卡死或重启	1. 内存不足	1. 简化代码，减少全局变量
	2. 循环中存在死循环或严重错误	2. 使用“分段测试”法，定位到导致错误的代码块
传感器数值不变或异常	1. 传感器接线错误	1. 仔细检查传感器是否正常连接
	2. 初始化失败	2. 确认传感器初始化代码是否正确

错误现象	可能原因	解决方法
按键没有反应	<ol style="list-style-type: none"> <li>1. 按键检测代码逻辑错误</li> <li>2. 循环中有延时过长</li> </ol>	<ol style="list-style-type: none"> <li>1. 使用打印模块确认程序是否进入按键检测</li> <li>2. 检查等待 是否时间太长，导致无法及时响应按键</li> </ol>
LED 灯或蜂鸣器不工作	<ol style="list-style-type: none"> <li>1. 驱动代码错误</li> </ol>	<ol style="list-style-type: none"> <li>1. 单独写一小段代码测试该硬件，确保其本身是好的</li> </ol>

## 模块七：拓展挑战

我们的“文物环境恒控系统”已经初具雏形，但它的潜力远不止于此！你能为它增加更酷的功能吗？

1. **挑战 1: 增加湿度监测** 青铜器不仅怕热，还怕潮湿。你能否加入湿度传感器，在青铜模式下，当湿度超过 70% 时也进行报警？
2. **挑战 2: 引入“缓冲”机制** 有时传感器数值可能只是瞬间跳变，我们不希望它立即报警。你能否设计一个算法，要求“连续 3 次检测到超标”才真正触发报警？
3. **挑战 3: 远程控制模式** 除了本地按键，我们能否通过物联网平台下发指令，远程将系统切换到“字画”或“青铜”模式？

## 模块八：学习反思

恭喜你完成了本次项目！让我们一起回顾和总结这次探索之旅的收获吧。

## 1. 我的知识清单

请用自己的话，简要描述你对以下概念的理解：

1. 传感器：\_\_\_\_\_
2. 阈值：\_\_\_\_\_
3. 条件判断：\_\_\_\_\_
4. “感知-判断-反馈”循环：\_\_\_\_\_

## 2. 我的“拦路虎”日志

---

我遇到的问题 我是如何解决的（或仍未解决） 我从中学到了什么

---

## 3. 我的收获与思考

在这个项目中，你认为最有挑战性的部分是什么？你是如何克服的？

如果让你向一位没学过编程的家人解释这个“文物环境恒控系统”的工作原理，你会怎么说？

除了文物保护，生活中还有哪些地方应用了这种“感知-判断-反馈”的自动化控制思想？（至少举出 2 个例子）

从使用者到创造者，你已经迈出了最重要的一步。继续探索，用科技让学习变得更美好！