

# 基于机械控制、点阵与位姿感应的 树莓派智能电子功德箱系统

2 0 2 3 微 机 原 理 与 微 系 统

汇报人：李稚凌

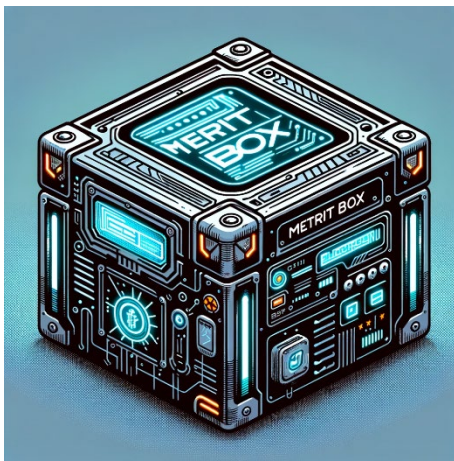
# 目录

1 / 项目背景

2 / 系统框架

3 / 模块封装

4 / 项目总结

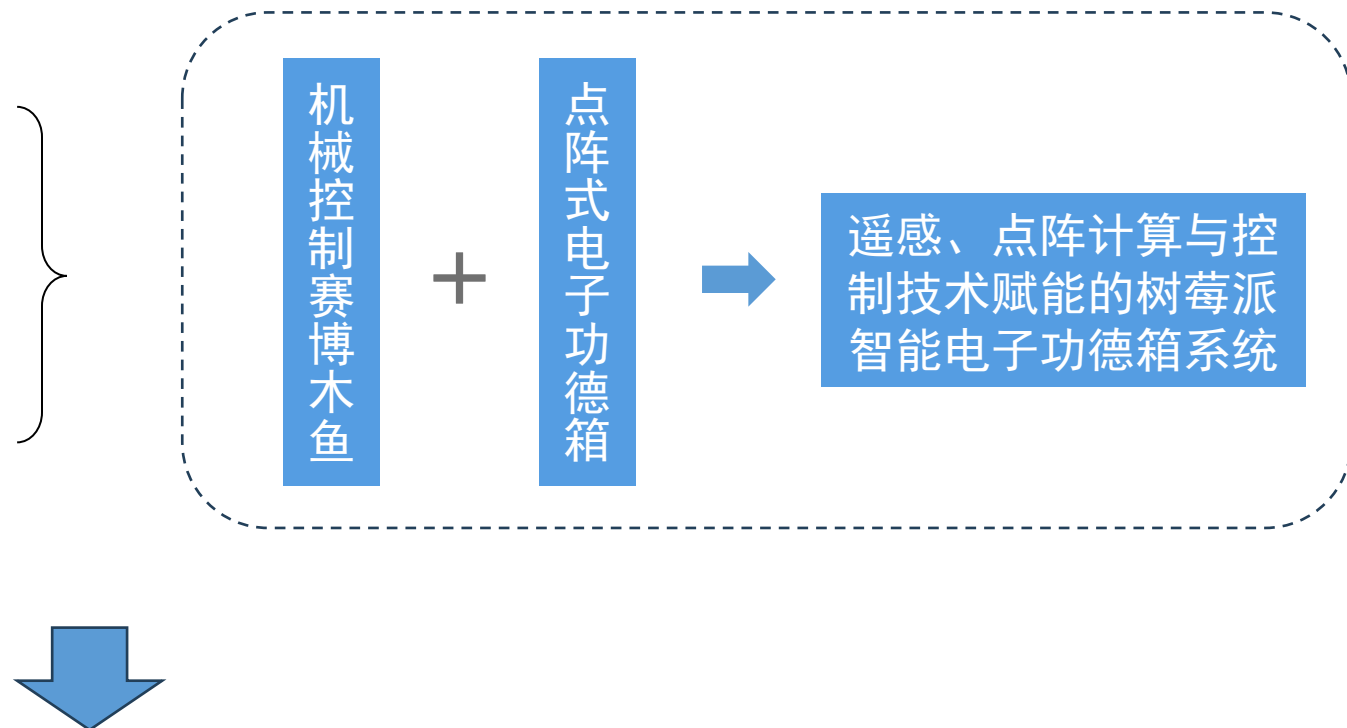


# 01

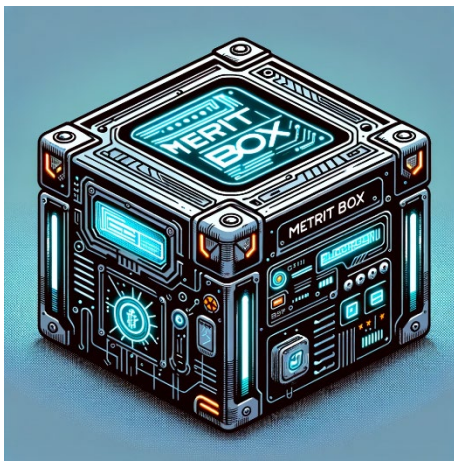
## 项目背景

## Part 1/ 项目背景

- 社会上对于更加便捷的功德箱数字化解决方案的广泛需求
- 年轻人群体中日渐复苏的传统文化和玄学思想
- 传统实体操作数字化转型需求的推动



通过将传统的功德箱概念与现代技术结合，促进传统内容的现代化，为社会提供新颖的传统文化活动参与方式



# 02

## 系统框架

& 功能详细介绍

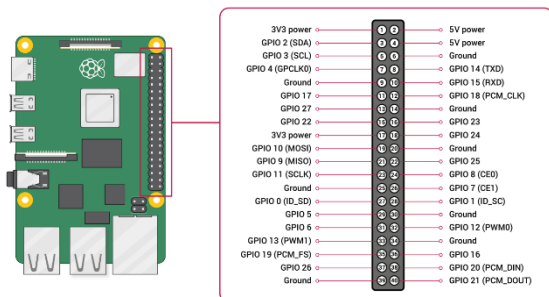


GY-25 (固定功德箱后侧)

# / 系统框架

UART通讯

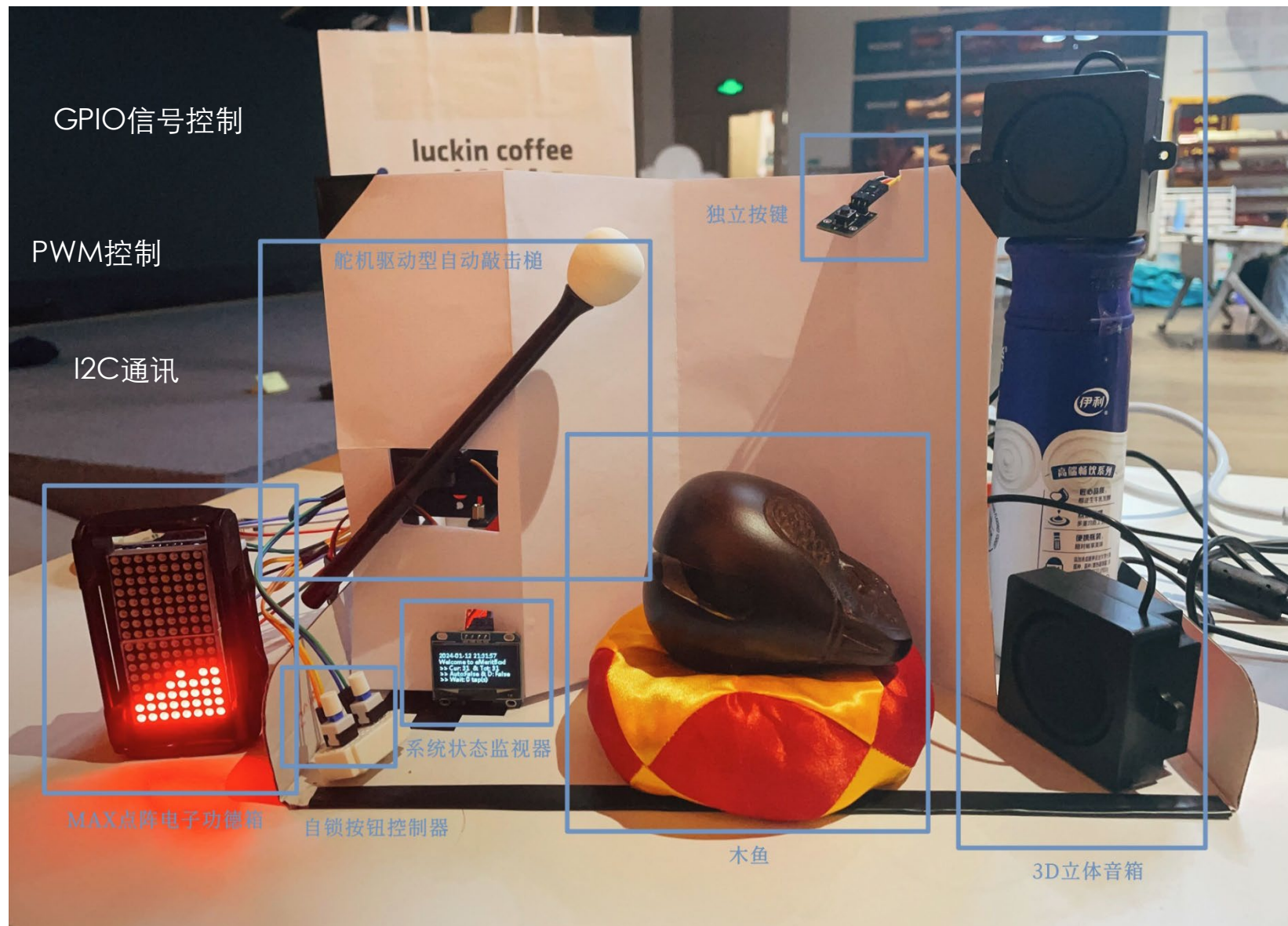
主控板：树莓派 (在屏风背后)



SPI级联通讯

GPIO信号  
模式控制

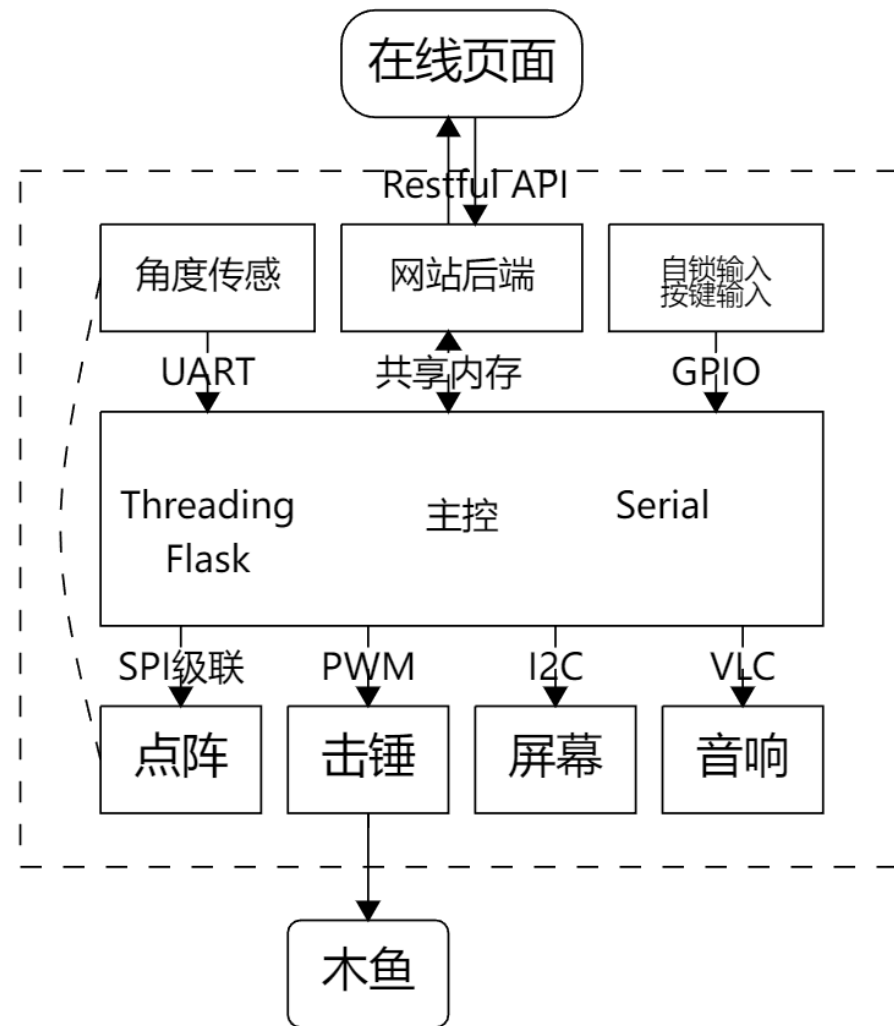
VLC控制  
音频输出



# Module 1 / 树莓派主机

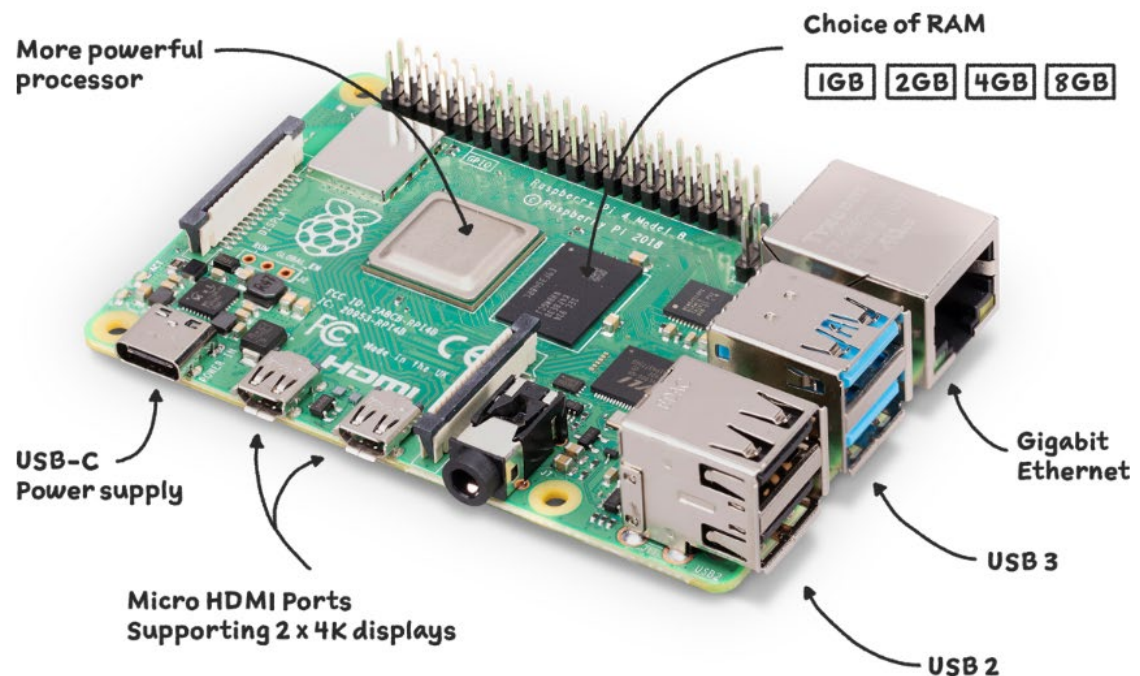
## 控制内容：

- 敲击槌PWM控制
- 功德箱MAX点阵控制
- GY-25数据处理
- OLED显示控制
- 实时按键扫描
- 音乐播放控制
- 木鱼网页服务器



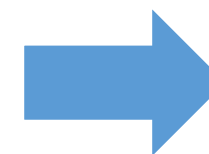


# Module 1 / 树莓派主机



## 功能丰富

- 核心处理器
- GPIO引脚
- 无线网络连接
- 音视频输出
- ...



承担系统  
主要控制工作



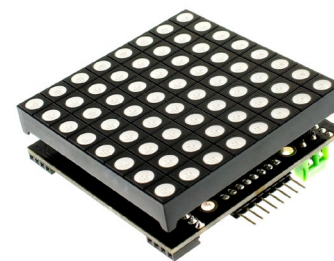
# Module 1/ 树莓派主机

## 控制内容：

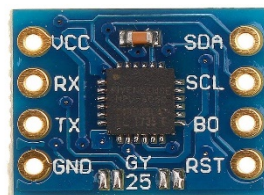
- 敲击槌PWM控制
- 功德箱MAX点阵控制
- GY-25数据处理
- OLED显示控制
- 实时按键扫描
- 音乐播放控制
- 木鱼网页服务器



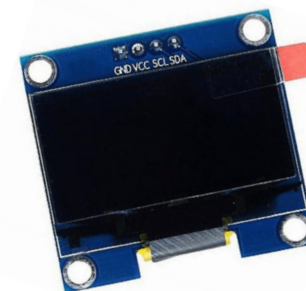
→ 敲击槌



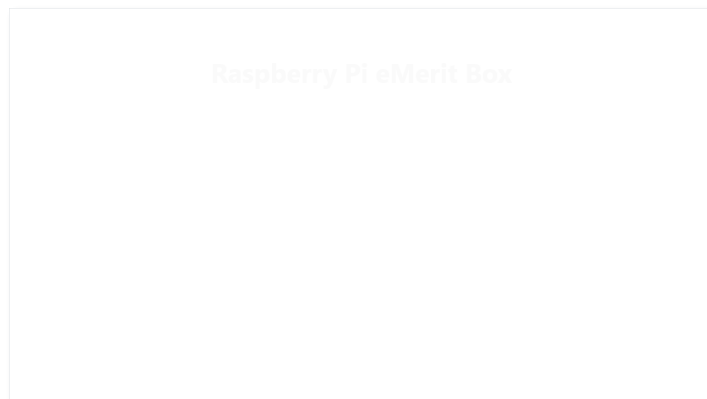
→ MAX点阵功德箱



→ GY-25角度传感



→ OLED状态显示



→ 木鱼网页服务



→ 音乐播放器



## Part 1/ 基础功能：木鱼敲击与功德球积攒

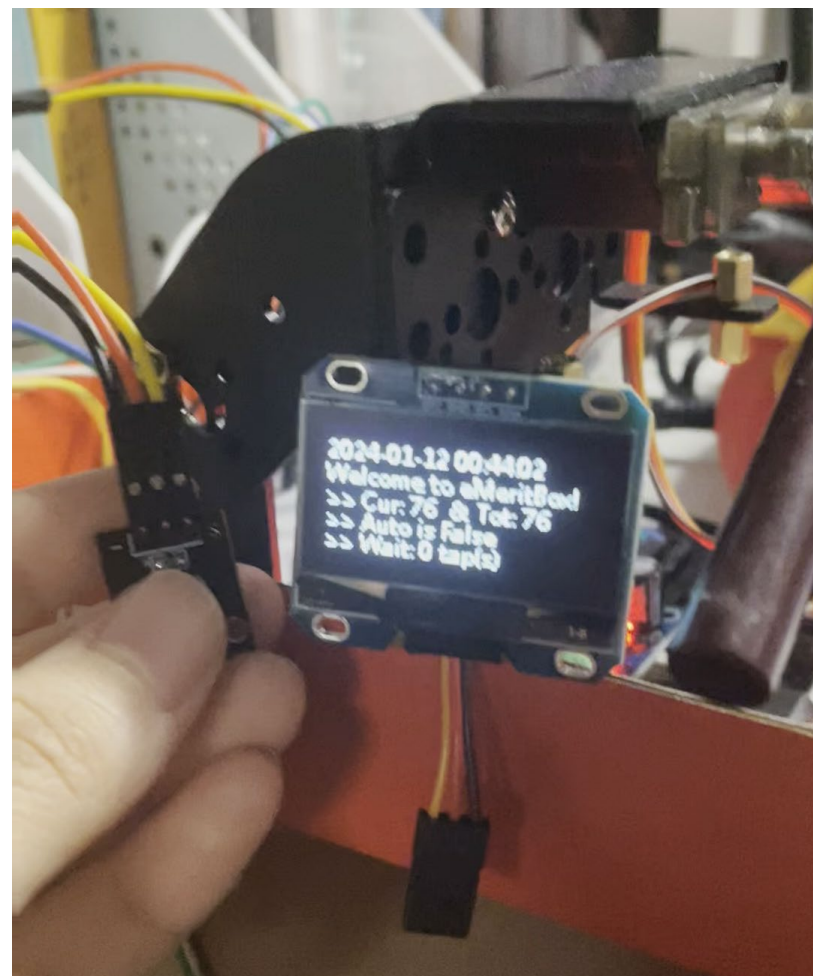
系统实现了电子木鱼的自动敲击，每次敲击后，都会积攒一颗功德球进入到功德箱中，从顶部随机一个位置刷新。



(同时会有佛经颂唱)

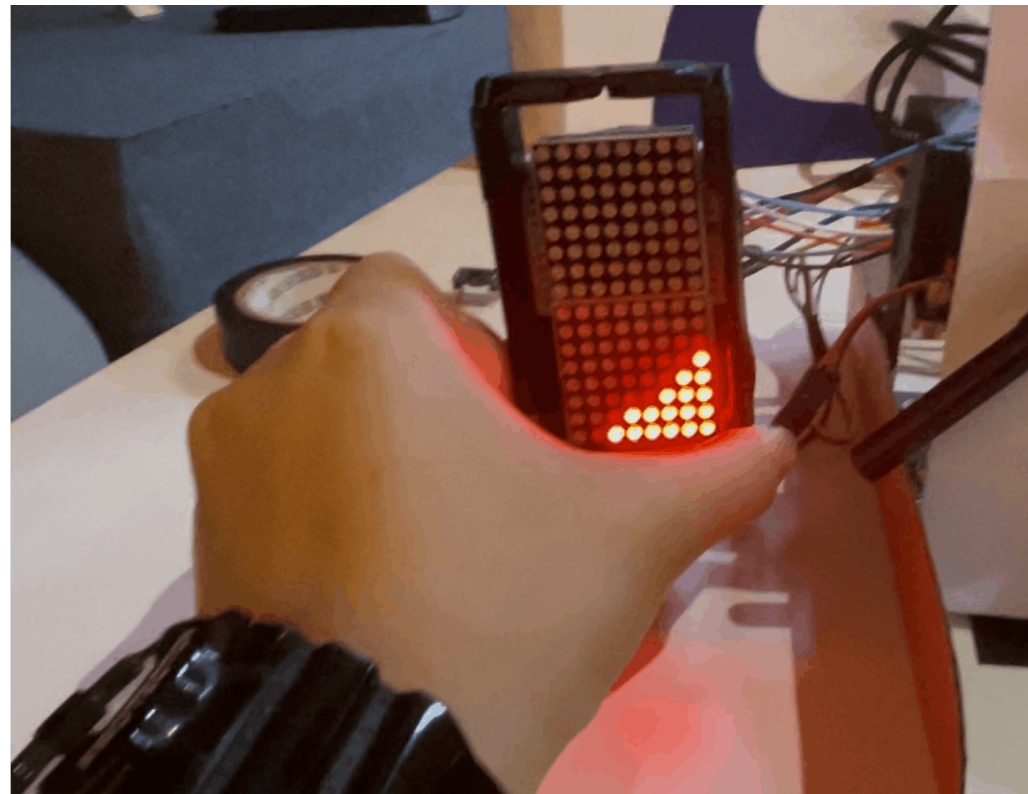
## Part 2/ 功德箱模式一：非Auto状态与按键连击器

在右侧按钮控制模式为 Auto: False 的情况下，单击按键将会增加一次待敲击次数，持续按住将会“连击”，在自动模式时待敲击次数不消耗，非自动模式下消耗并敲击木鱼。



## Part 3/ 功能升级：功德箱重力感应

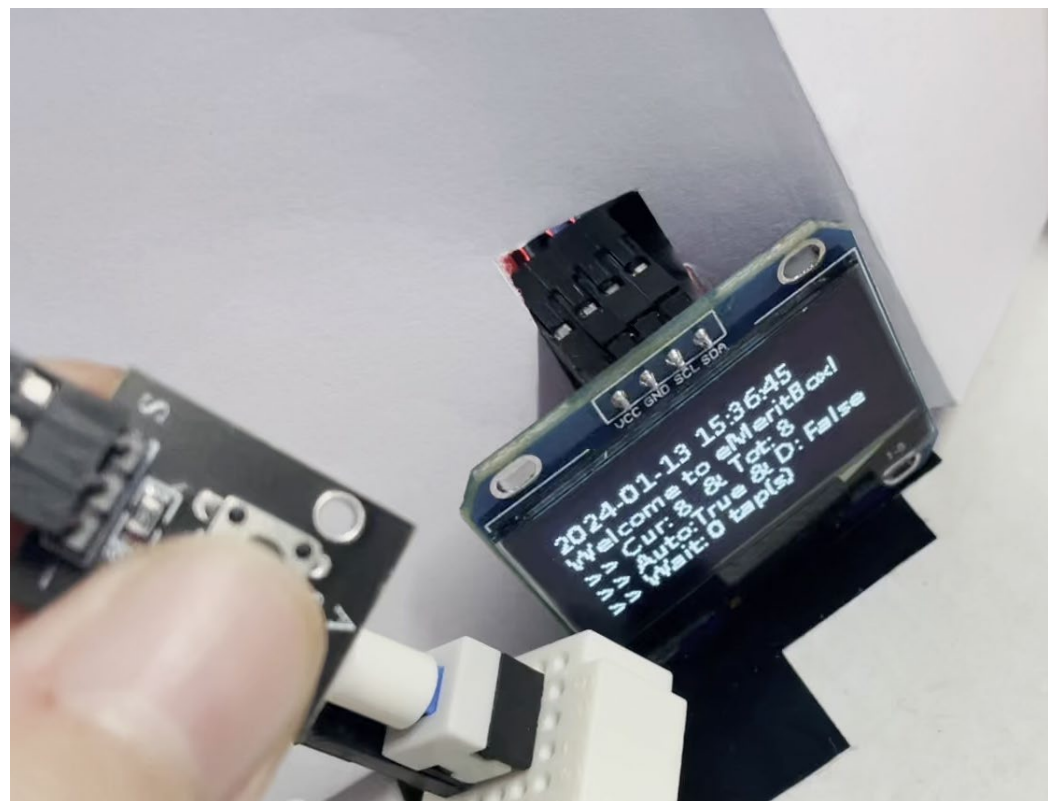
当功德箱中积攒有功德球时，功德箱可以使用遥感控制，实现功德球能够根据重力方向缓缓移动。





## Part 4/ 功德箱模式二：Auto状态与捐功德状态切换器

在右侧按钮控制模式为 Auto: True 的情况下，单击按键将会切换功德球捐赠模式 Donate: On/Off，进而控制箱子是否为打开状态。



## Part 5/ 功能升级：功德箱捐献状态切换

当功德箱捐赠模式开启时，功德箱顶部被打开，由于重力感应，通过倾倒使得功德球到达箱子顶部时，功德球会掉出功德箱，完成捐献。

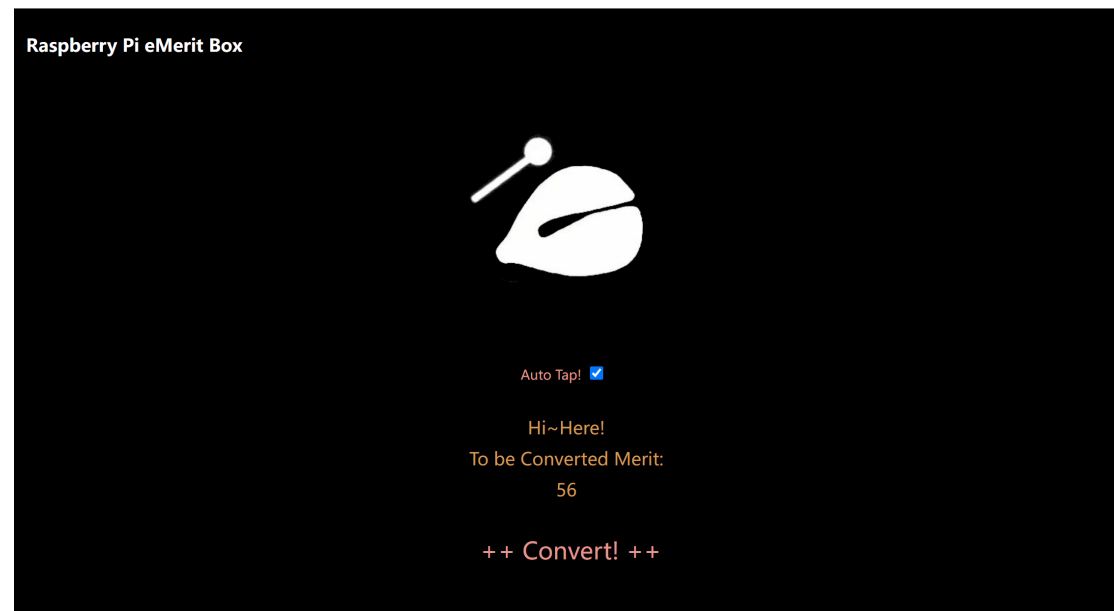


## Part 6/ 功德箱模式三：网络客户端控制敲击补充器

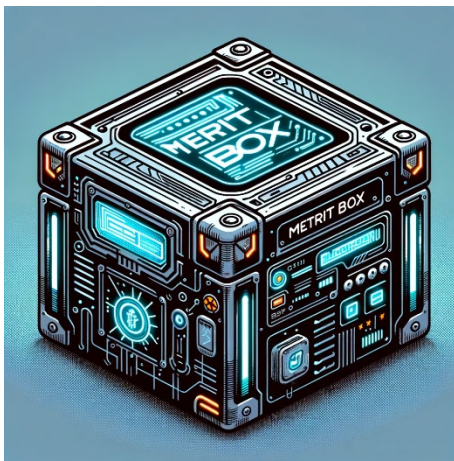
除此之外，在树莓派内部还使用 Flask 搭建了木鱼服务器，能够使用浏览器网页访问木鱼应用程序，通过点击 Convert!，将浏览器中存储的敲击量转化为实际木鱼敲击，实现功德一站式快速获得！

- >> 前后端分离
- >> 完全使用树莓派实现

此处是使用电脑浏览器访问树莓派 ip，进入树莓派主页，并点击进入木鱼应用。







# 03

## 模块封装

& 技术难点与解决方案

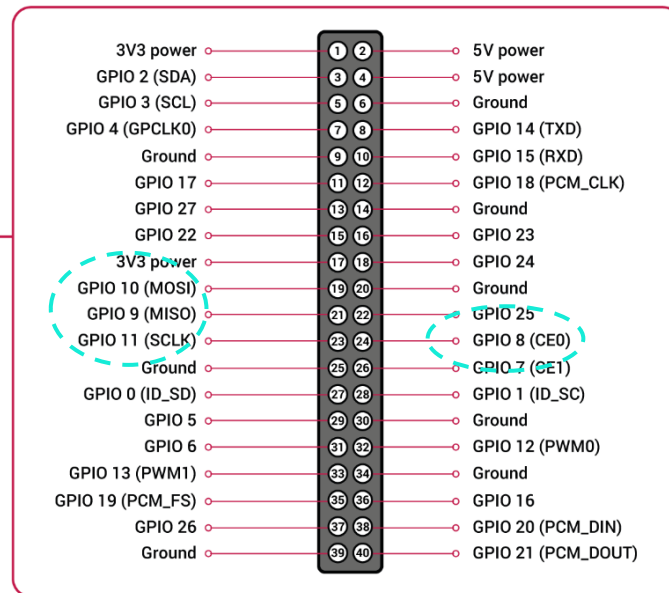
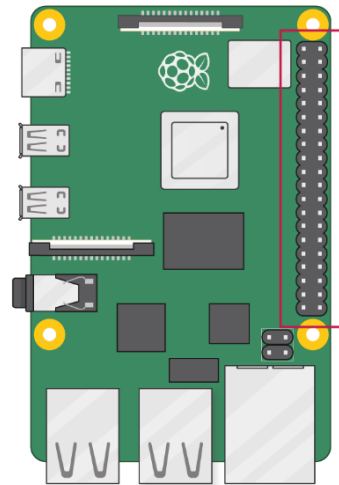
# Module 2/ Max7219点阵功德箱

## 模块简介:

Max7219LED 点阵控制模组;

SPI 通讯协议双模块级联通讯;

附加芯片负责控制每块4\*4点阵;



X 2

## 技术难点:

点阵驱动适配和功德球显示控制;

类封装: SnowBox与MatrixCtrl;

-> 协作完成功德球遥感控制显示;

```
> class SnowBox: ...
> class LEDMatrix: ...
```

## 解决方案:

配置环境调用luma matrix库;

手动编写点阵映射绘制函数;

进行两个类的封装, 手动编写了复杂的多个方法, 协同完成;

```
def add_snow(self, n):
```

```
def update(self):
```

```
def check_and_move(self, idx):
```

```
def map_and_draw(self, snowbox)
```

# Module 3/ GY-25遥感角度传感

## 模块简介:

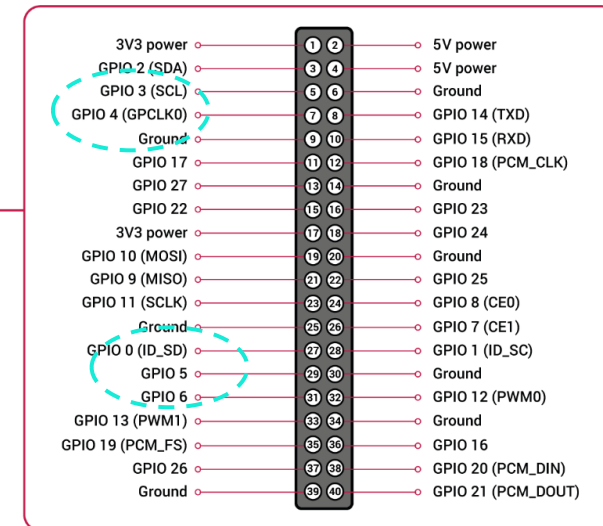
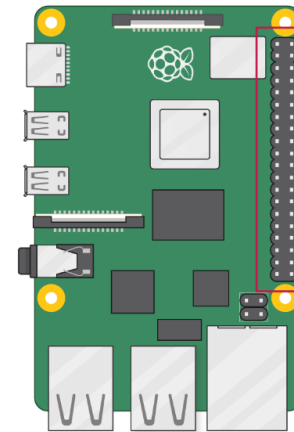
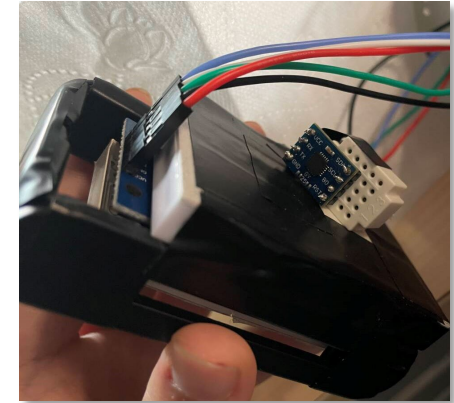
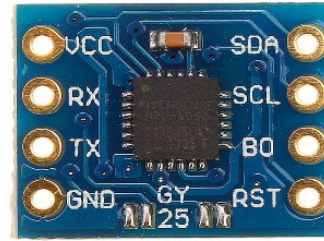
GY-25 传感器，是 MPU6050 陀螺仪模块的高级封装模块，能高精度地传输当前模块位姿。

## 技术难点:

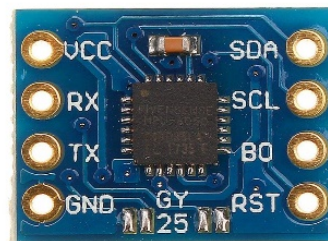
UART 通讯与数据获取；  
类封装：GY-25Ctrl；  
树莓派 UART 端口被蓝牙占用；

## 解决方案:

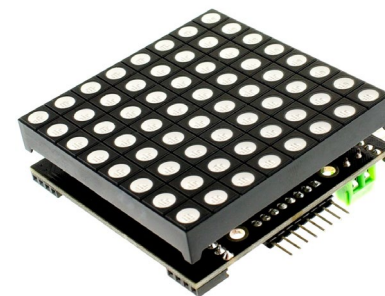
将目前 GY-25 仅有的C语言解算方式使用Python实现，处理字节码大小端问题；  
进行 GY-25 类的封装，方便主控调用相关功能；  
处理树莓派 UART 占用 \*



## Module 2+3/ GY-25 + Max7219



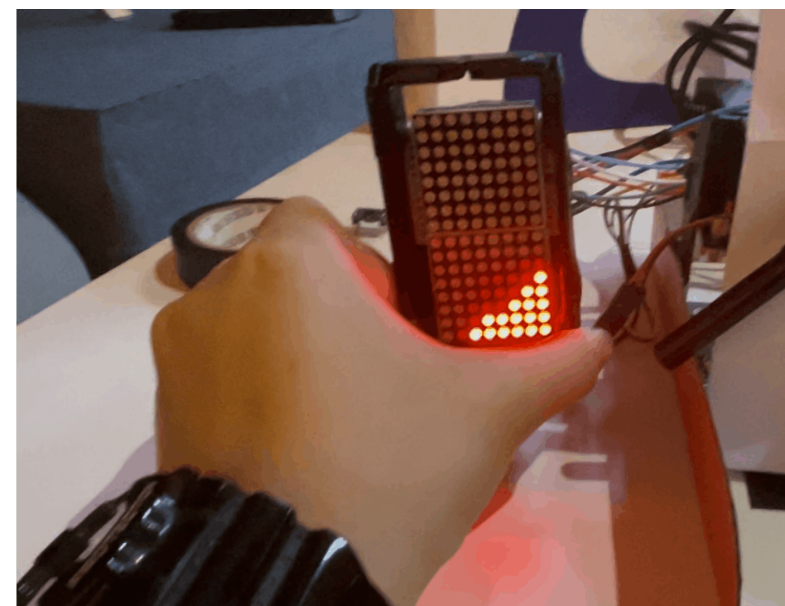
X



X 2

### 运动追踪 · 功德箱实现方案：

利用子线程高频率实时获取陀螺仪数据，根据当前位姿，调用“def change\_g”改变SnowBox对象的重力属性，进而自动地在定时更新点阵位置（def update）时使用新的重力方向进行更新。





# Module 4/ MG-90舵机木鱼敲击槌

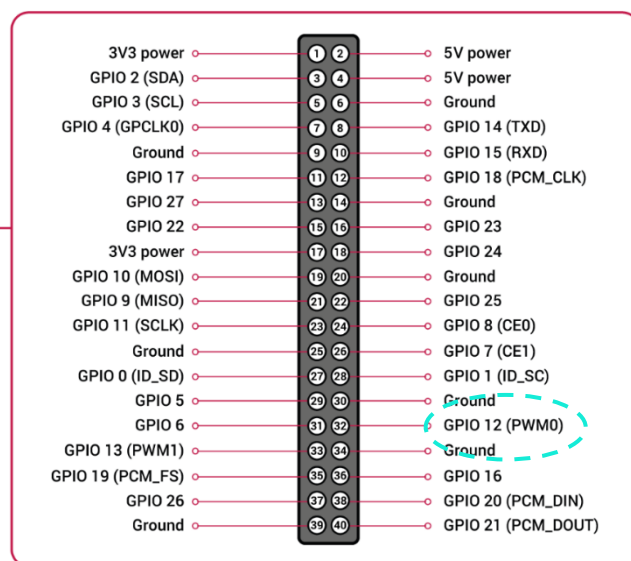
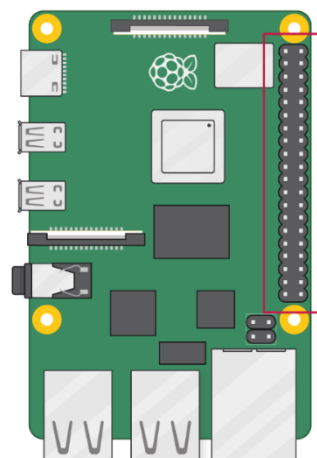
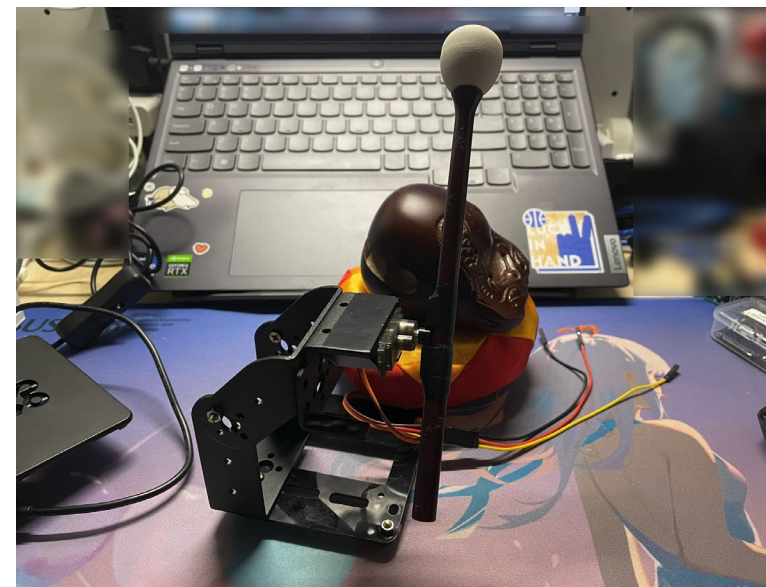
## 模块简介:

MG-90，是一款金属舵机。

通过一定的机械方法完成了木鱼敲击槌的组装，为敲击木鱼动作的实现奠定了物质基础。

## 技术难点:

PWM控制舵机角度；  
敲击动作的参数调节；  
敲击模式、次数统计、待敲次数记录等变量管理；



## 解决方案:

反复进行测试，调节控制参数；  
进行 ServoCtrl 类的封装，方便主控调用与读取，同时便于对接自身的服务器功能；

# Module 5/ OLED状态监控

当前状态总监视器一站式浏览

## 模块简介:

OLED显示模组，用于文字或图案的清晰显示，效果好，对比度高，刷新速度快。

## 技术难点:

树莓派IIC.0通道默认禁用;

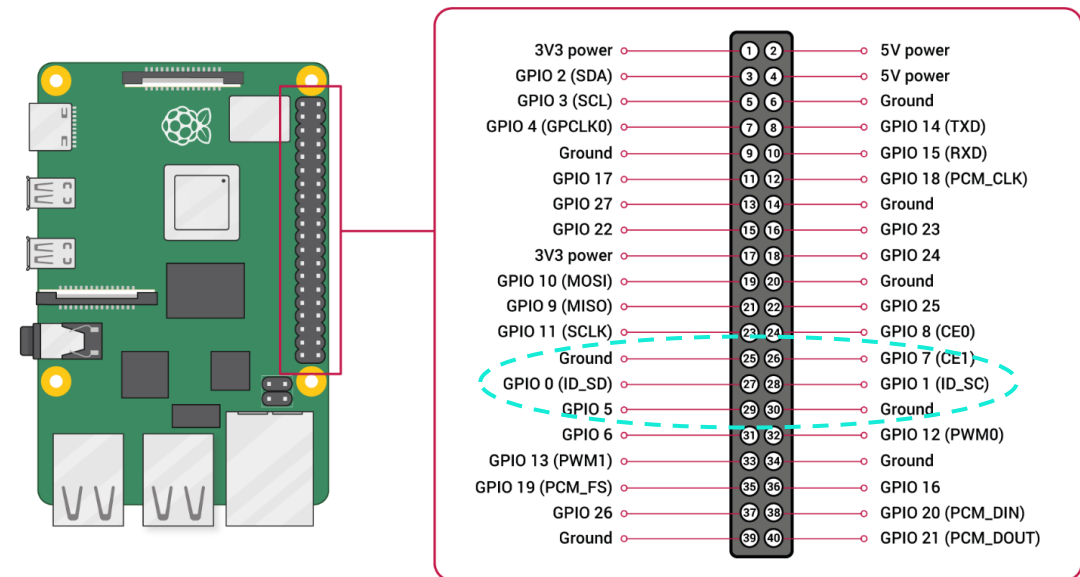
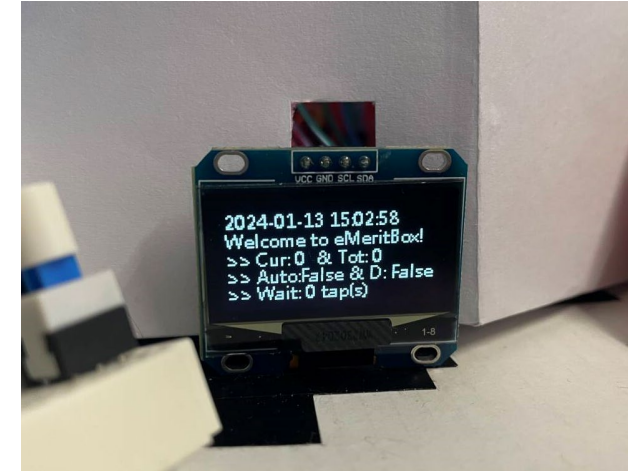
整体显示优化调节;

## 解决方案:

反复测试，调节显示参数如字体字号等;

封装为OLEDCtrl类方便调用;

开通树莓派IIC.0通道\*



# Module 6/ 3D立体音箱与三按键联合交互

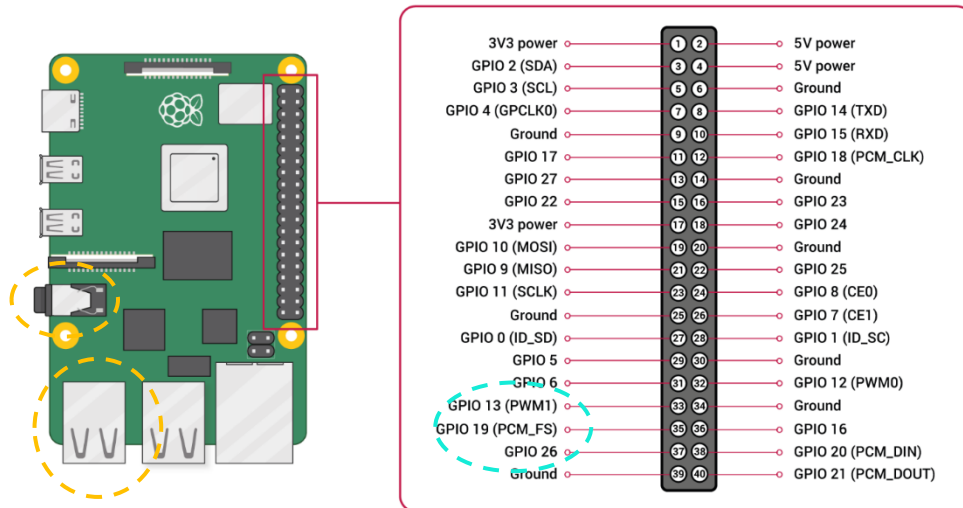
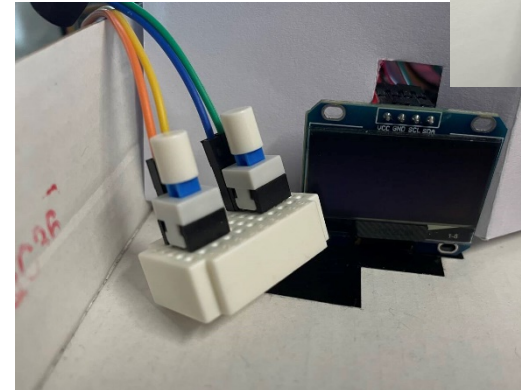
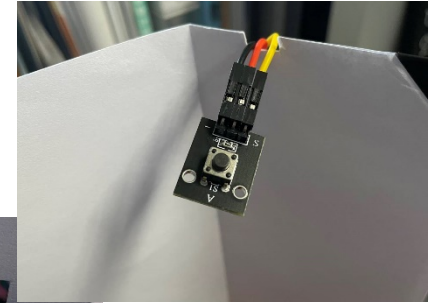
背景音乐极简控制沉浸式体验 & 多线程按键动作响应

## 模块简介:

USB音箱，使用VLC控制音乐播放，封装为BGMPPlayer类；  
双自锁开关 + 单按键开关，灵活控制音乐、模式和多种功能。

音乐控制  
左

模式控制  
右





# / 相关问题研讨

## 1. 硬件串口与mini串口默认映射对换

**ttyAMA0 <-> ttyS0**

## 2. 多串口配置

**+ttyAMA1, 2, ...**

```
Shell v
pi@raspberrypi:~ $ dtoverlay -h uart2
Name:  uart2
Info:  Enable uart 2 on GPIOs 0-3
Usage: dtoverlay=uart2,<param>
Params: ctsrts      Enable CTS/RTS on GPIOs 2-3 (default off)
```

UART 硬件资源占用与调度

树莓派双 IIC 通道占用与冲突

模块调度过多导致端口冲突

事件嗅探的多线程解决方案

# / 相关问题研讨

1. 关闭 camera 功能
2. 添加 `dtparam=i2c_vc=on`

## 3.3: Board-specific labels and parameters

Raspberry Pi boards have two I2C interfaces. These are nominally split: one for the ARM, and one for VideoCore (the "GPU"). On almost all models, `i2c1` belongs to the ARM and `i2c0` to VC, where it is used to control the camera and read the HAT EEPROM. However, there are two early revisions of the Model B that have those roles reversed.

To make it possible to use one set of overlays and parameters with all Pis, the firmware creates some board-specific DT parameters. These are:

```
i2c/i2c_arm
i2c_vc
i2c_baudrate/i2c_arm_baudrate
i2c_vc_baudrate
```

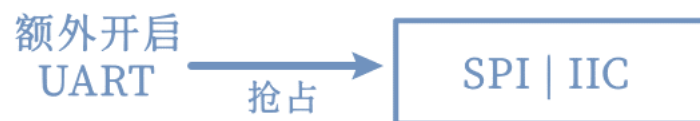
UART 硬件资源占用与调度

树莓派双 IIC 通道占用与冲突

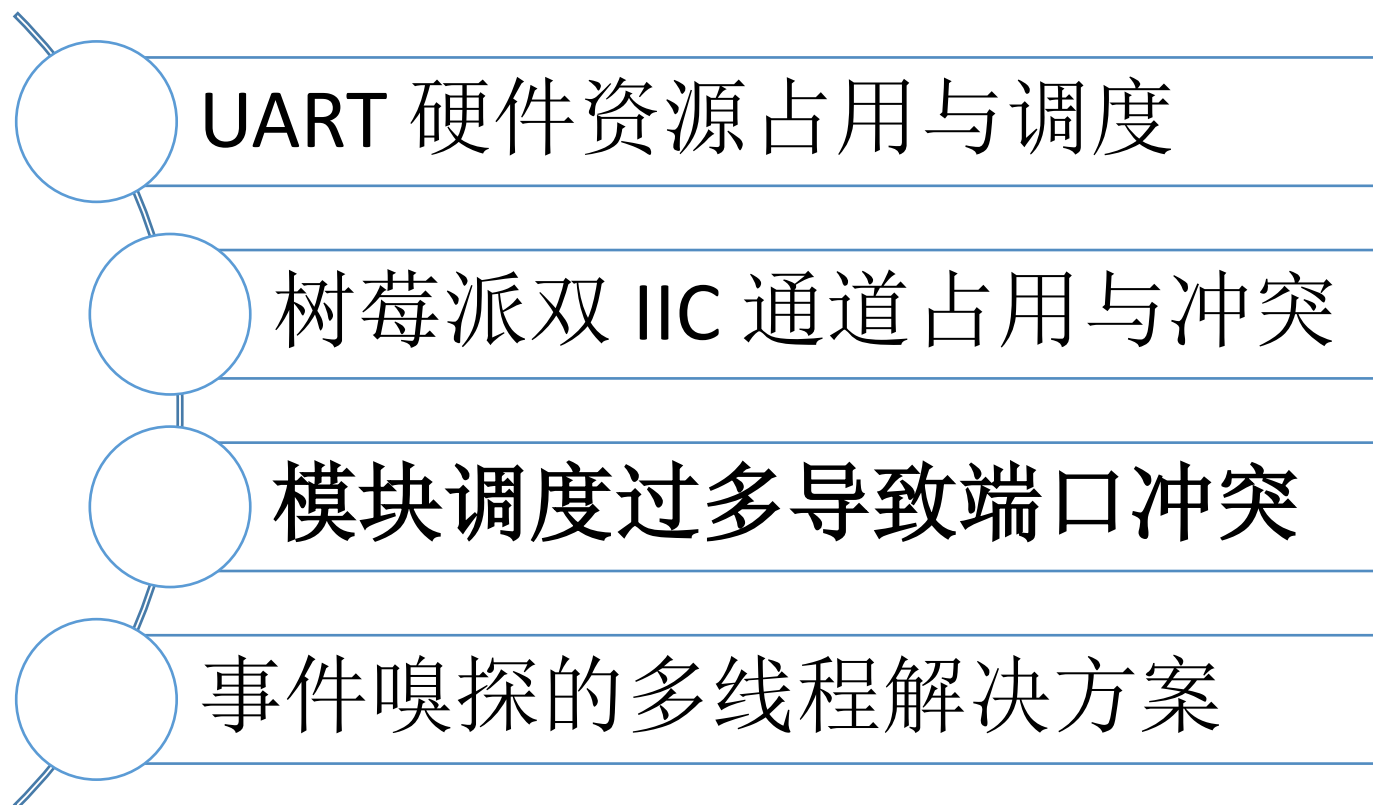
模块调度过多导致端口冲突

事件嗅探的多线程解决方案

## / 相关问题研讨



模块数量不断增加时，  
技术栈难度非线性！



# / 相关问题研讨

## 触发事件列表:

1. 自锁按钮状态改变
2. 多功能单按键被触发
3. 实时遥感数据读取
4. 服务器端口请求响应
5. ...

在长时间程序占用（如舵机执行）期间，  
如何保证触发事件的瞬间响应？

```
try:  
    _thread.start_new_thread(check_btn, ())  
    _thread.start_new_thread(check_ypr, ())  
    _thread.start_new_thread(serve, ())
```

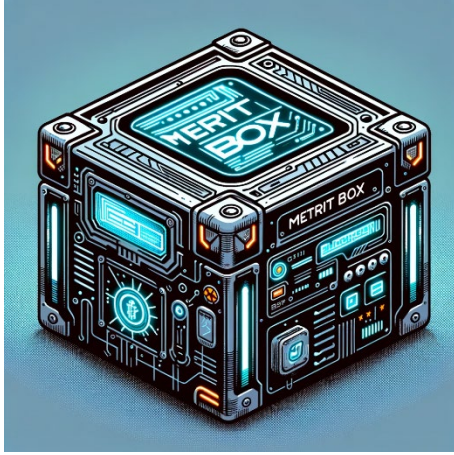
在Python运行文件内执行任务多线程处理

UART 硬件资源占用与调度

树莓派双 IIC 通道占用与冲突

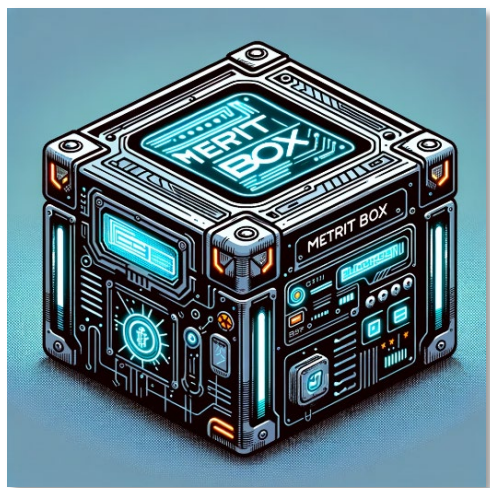
模块调度过多导致端口冲突

事件嗅探的多线程解决方案



# 04

## 项目总结



## eMeritBox

# 敲电子木鱼，得赛博功德

实现目标电子功德箱系统的完整构建；  
融合多种技术和模块协作；  
解决所遇到的技术问题；  
完整封装可供使用的MAX点阵重力  
控制和显示类，可迁移。

eMeritBox > src > main\_ctrl.py > ...

```
27 > class BGMPlayer: ...
44
45
46 > class OLEDCtrl: ...
72
73
74 > class SnowBox: ...
192
193
194 > class LEDMatrix: ...
240
241
242 > class ServoController: ...
331
332
333 > class GY25: ...
366
367
368 # global
369 servo_controller = None
370 box = None
371 gy25 = None
372 inversed = None
373
374 app = flask.Flask(__name__)
```

The background is a vibrant blue with a complex, organic pattern of lighter and darker shades, creating a textured, almost marbled effect. Two thin, white horizontal lines extend from the left and right sides of the word 'Thanks!', framing it.

**Thanks!**