ESP32 MicroPython 中文 OLED 跑馬燈顯示系統

Show Image

Show Image

Show Image

一個創新的解決方案，讓 ESP32 配合 SSD1306 OLED 顯示器能夠顯示中文字元。透過 Flask API 動態生成中文字型點陣圖，支援靜態顯示和跑馬燈滾動效果。

✨ 特色功能

- 🇹🇼 **完整中文支援** - 突破 SSD1306 OLED 原生不支援中文的限制
- 🎯 **跑馬燈效果** - 平滑的文字滾動動畫
- 🌐 **網路架構** - 前後端分離設計，透過 RESTful API 通訊
- 📝 **彈性字體** - 可調整字體大小
- 🔄 **自動處理** - 智慧處理空白字元和 URL 編碼
- 💾 **記憶體優化** - 內建垃圾回收機制

🎬 應用場景

- 中文提示看板
- IoT 中文顯示器
- 即時通知裝置
- 智能家居資訊顯示

🛠️ 硬體需求

元件	規格	說明
ESP32 開發板	DevKitC 或相容板	主控制器
SSD1306 OLED	0.96吋 128x64 I2C	顯示模組
杜邦線	4條	I2C 連接線材
USB 線	Type-C/Micro-USB	供電與燒錄

接線圖

```
ESP32      →    SSD1306 OLED
GPIO 21    →    SDA
GPIO 22    →    SCL
3.3V       →    VCC
GND        →    GND
```

📋 軟體需求

PC 端 (Flask 伺服器)

- Python 3.7+
- Flask 2.0+
- Pillow 8.0+

ESP32 端

- MicroPython 1.18+
- urequests 庫
- ssd1306 庫

🚀 快速開始

1. 環境準備

```bash
# 安裝 Python 依賴
pip install flask pillow

# 下載中文字型檔案
# 將 NotoSansTC-Regular.ttf 放入專案目錄
```

2. 啟動 Flask 伺服器

```bash
python font_server.py
```

伺服器將在 `http://0.0.0.0:5000` 啟動

3. 設定 ESP32

1. 燒錄 MicroPython 韌體到 ESP32
2. 上傳以下檔案到 ESP32：
   - `oled_display.py`
   - `main.py`
   - `ssd1306.py` (OLED 驅動)

4. 修改設定

編輯 `main.py` 中的設定：

```python
WIFI_SSID = "你的WiFi名稱"
WIFI_PASSWORD = "你的WiFi密碼"
FONT_API_URL = "http://你的伺服器IP:5000/api/font?text="
```

5. 執行程式

重啟 ESP32，系統將自動：

1. 連接 Wi-Fi
2. 顯示連線狀態
3. 開始顯示中文文字

📁 專案結構

```
ESP32_Chinese_OLED/
├── font_server.py          # Flask 字型 API 伺服器
├── oled_display.py         # ESP32 OLED 顯示類別
├── main.py                 # ESP32 主程式
├── NotoSansTC-Regular.ttf  # 中文字型檔案
├── README.md               # 專案說明
└── docs/                   # 文檔目錄
    └── 專案報告.pdf         # 詳細技術報告
```

🔧 API 使用說明

取得字型點陣圖

```http
GET /api/font?text={encoded_text}
```

回應格式：

```json
{
  "bitmap": [[0,1,0,1], [1,0,1,0], ...],
  "width": 48,
  "height": 24,
  "success": true
}
```

測試範例：

```bash
# 基本中文
curl "http://localhost:5000/api/font?text=你好"

# 中英混合
curl "http://localhost:5000/api/font?text=Hello世界"

# 包含空白
curl "http://localhost:5000/api/font?text=你好%20世界"
```

💻 程式使用範例

基本使用

python

```python
from oled_display import OledChineseDisplay

# 初始化顯示器
display = OledChineseDisplay(
    scl_pin=22,
    sda_pin=21,
    font_api_url="http://192.168.1.100:5000/api/font?text=",
    scroll_mode=True
)

# 連接 Wi-Fi
display.connect_wifi("WiFi名稱", "WiFi密碼")

# 顯示文字
display.display(["你好世界", "Hello World", "跑馬燈測試"])
```

進階設定

python

```python
# 靜態顯示模式
display = OledChineseDisplay(
    scl_pin=22,
    sda_pin=21,
    font_api_url="http://192.168.1.100:5000/api/font?text=",
    scroll_mode=False  # 關閉跑馬燈
)

# 自訂顯示間隔
display.display(["文字1", "文字2"], delay_between_texts=3)

# 顯示簡單英文訊息
display.show_message("System Ready", "IP: 192.168.1.123")
```

🐛 故障排除

常見問題

問題	可能原因	解決方案
OLED 無反應	接線錯誤	檢查 I2C 連線和電源
顯示亂碼	字型檔案問題	確認字型檔案路徑正確
Wi-Fi 連線失敗	密碼錯誤	檢查 SSID 和密碼
API 連線失敗	伺服器未啟動	確認 Flask 伺服器運行狀態
記憶體不足	文字過長	縮短顯示文字或增加垃圾回收

除錯技巧

1. 檢查 I2C 設備

python

```python
import machine
i2c = machine.I2C(0, scl=machine.Pin(22), sda=machine.Pin(21))
print(i2c.scan())  # 應該顯示 [60] (0x3C)
```

2. 測試 API 連線

bash

```bash
curl "http://你的伺服器IP:5000/test"
```

3. 監控記憶體使用

python

```python
import gc
print(f"可用記憶體: {gc.mem_free()} bytes")
gc.collect()
```

📊 性能表現

- **字型轉換速度**: 200-500ms (依文字長度)
- **網路延遲**: 100-300ms (區域網路)
- **滾動流暢度**: 80ms/幀
- **記憶體使用**: 15-25KB (依文字長度)
- **最大支援文字**: 約 11 個中文字元

🔮 未來規劃

- [ ] 雲端 API 整合
- [ ] 多語言支援擴展
- [ ] 圖形和圖標顯示
- [ ] 觸控互動功能
- [ ] Web 管理介面
- [ ] OTA 更新支援

🤝 貢獻指南

歡迎提交 Issue 和 Pull Request！

1. Fork 此專案
2. 建立功能分支 (`git checkout -b feature/AmazingFeature`)
3. 提交變更 (`git commit -m 'Add some AmazingFeature'`)
4. 推送到分支 (`git push origin feature/AmazingFeature`)
5. 開啟 Pull Request

📄 授權條款

本專案採用 MIT 授權條款 - 詳見 LICENSE 檔案

👨‍💻 作者

- **專案開發者** - 您的姓名

🙏 致謝

- MicroPython 社群
- Flask 框架開發團隊
- SSD1306 驅動程式庫貢獻者
- 中文字型提供者

📞 聯絡方式

如有任何問題或建議，歡迎透過以下方式聯繫：

- GitHub Issues: 提交問題
- Email: your.email@example.com

⭐ 如果這個專案對你有幫助，請給個星星支持！