

ESP-HMI

【Designed by SoTWild】 My Website 当代研究所 Licence GPL Platform Github Version 1.0

Language C/C++

[[English](#)] [[简体中文](#)]



成品图

目录：

- [概述](#)
 - [FAQ](#)
- [更新日志](#)
- [硬件组成](#)
- [软件组成](#)
 - [应用](#)
- [注意事项](#)
 - [硬件](#)
 - [软件](#)
- [搭建方法](#)

• 其他

- [特别感谢](#)
 - [结语](#)
-

概述：

ESP-HMI 是 Link 设备链中的一个，是整个项目最难开发的部分，主要提供远程直接开关设备、数据汇总（设备监控）。

我还开发了一些针对不同人群的功能：文本阅读、编辑、图片查看、播放MJPEG视频、运行小程序等。如果你是开发者，你也可以对它进行二次开发。（普通人可以把它看做一个功能极其简单的小电脑）

为了高效率运行程序，我移植了 FreeRTOS 操作系统，使得主控芯片可以“同时”运行多个程序。

本文档很长，见谅。

项目开发人员：

SoTWild



FAQ：

1) 整个设备的造价大约是多少？

< 100¥

2) 复刻的难度如何？

硬件上有一两个元件最小封装为0201，软件已经开源，综合来看很容易复刻。

3) 续航时间？

充电芯片采用TP4056，可以通过配置PROG引脚控制充电电流，我这里使用了500mA充电电流，电池为4000mAh聚合物锂电池，理论上可以供电超过26小时，充电时间约为8小时。（运行电流约为150mA）

4) 成品的体积?

全面屏, 3.5寸TFT屏幕, 长8.5cm, 宽5.5cm, 厚12mm (不加电池)。

5) 支持的SD卡容量?

目前测试最大为32G, 再大应该不行。

6) 主控芯片是什么?

ESP32-D0WD-V3 双核, 模组为ESP32-WROVER。

7) 开发环境?

Visual Studio Code + Platform IO

Change Log:

2021.09.05: 开始构思

2021.10.22: 成功播放 Mjpeg 视频

2021.10.23: 在 Mjpeg 基础上, 添加了 pcm 背景音乐

2021.12.19: 成功驱动 XPT2046 触摸芯片并运行了 LVGL demo

2021.12.21: 优化 Mjpeg 240*320 像素, 帧率最高 8.9FPS

2021.12.25: 解码 jpg 文件; 完成 ESP-IED Ver 1 PCB 的绘制

2021.12.26: 编写画图程序

2021.12.30: 编写 MP3 程序

2022.01.01: ESP-HMI Ver 1 直插板成功搭建

2022.01.02: 开始编辑源代码

2022.01.02: 添加 Media Player 中 Mjpeg 播放功能

2022.01.03: 添加 Media Player 中 Jpeg 播放功能 | 添加 Media Player 中 MP3 播放功能

2022.01.09: 添加 Media Player 中 PCM 播放功能

2022.01.09: 添加 Media Player 中 TXT 播放功能

2022.01.12: 添加 Draw 功能

2022.01.12: 添加 THMini 功能

2022.03.28: 添加 Album 功能

2022.03.28: 添加 Ebook 功能

2022.03.29: 代码优化

2022.04.01: 蓝牙音频传输 (接收)

2022.04.05: 开始编写 Sounder

2022.04.10: 完成编写 Sounder

2022.04.13: 开始编写 Vision

2022.04.14: 完成编写 Vision

2022.04.17: 完善 Ebook

2022.04.18: 开始并完成编写 Settings

2022.04.22: 绘制 Ver 1 原理图

2022.05.02: 用户登陆系统

2022.07.05: 计算器 (普通)

2022.07.08: 温度计

2022.07.12: 文件上传

2022.07.14: WiFi

2022.07.30: Ver 1 PCB完成

2022.08.12: Ver 1 PCB调试完成

2022.08.13: 添加Task Manager

2022.08.14: 完成Album功能; 编写文档

2022.08.15: 编写文档

2022.08.16: 修改Task Manager | Debug Vision

2022.08.19: 完善Album

2022.08.21: 开始编写 東方微型行

2022.08.29: 修复了用户名长度的问题 | 添加bat快捷转换器

ESP-HMI 硬件

主要硬件组成:

· ESP32-WROVER

[ESP32-WROVER](#) 系列模组基于 **ESP32-D0WD 双核** 芯片设计，其强大的双核性能适用于对内存需求大的应用场景，例如多样的 **AIoT** 应用和**网关**应用。



ESP32-WROVER 模组

· 3.5 寸 TFT_LCD

驱动芯片: ILI9488

通讯接口: SPI

触摸芯片: XPT2046



屏幕模块

· 存储卡

最大 **32G**，推荐 4 或 8G。



32G TF卡

· SHT30：

SHT30能够提供极高的**可靠性和出色的长期稳定性**，具有**功耗低、反应快、抗干扰能力强**等优点。

轻松实现城市环境监控、智能楼宇、工业自动化、智能家居等物联网应用场景的温湿度传感。

- 主芯片（传感器）：Sensirion SHT30
- 供电电压(VCC)：3.3V ~ 5.5V
- 通信接口：Gravity Analog (PH2.0-3P, 模拟电压输出0.3-2.7V)
- 工作电流：< 0.5 mA
- 产品尺寸：30×22 mm
- 重量：3 g

温度测量性能：

- 量程：-40 ~ 125 °C
- 分辨率：0.01 °C, 14bit
- 精度：±0.2°C@10~55°C (典型值)，±1.5°C@-40 ~ 125 °C (典型值)
- 响应速度：> 2s

湿度测量性能：

- 量程：0~100 %RH
- 分辨率：0.006 %, 14bit
- 精度：±3 %RH@10~90 %RH (典型值)，±8 %RH@0~100 %RH (典型值)
- 响应速度：> 8s



SHT30

· PCF8563

PCF8563 是 PHILIPS 公司推出的一款**工业级内含I2C 总线接口功能的具有极低功耗的多功能时钟/日历芯片**。是一款性价比极高的时钟芯片，它已被广泛用于电表、水表、气表、电话、传真机、便携式仪器以及电池供电的仪器仪表等产品领域。



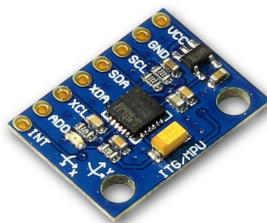
PCF8563 模块

· MPU6050:

MPU-60X0 是全球首例 ⑨ 轴运动处理传感器。它集成了3 轴MEMS陀螺仪，3 轴MEMS加速度计，以及一个可扩展的数字运动处理器DMP，可用I2C接口连接一个**第三方**的数字传感器，比如磁力计。扩展之后就可以通过其**I2C 或 SPI** 接口输出一个⑨轴的信号（SPI 接口仅在MPU-6000 可用）。

MPU-60X0 对陀螺仪和加速度计分别用了三个**16位的ADC**（0~65535），将其测量的模拟量转化为可输出的数字量。传感器的测量范围都是用户可控的，陀螺仪可测范围为 **±250, ±500, ±1000, ±2000°/秒 (dps)**，加速度计可测范围为 **±2, ±4, ±8, ±16g**。

芯片尺寸**4×4×0.9mm**，采用**QFN**封装，可承受最大**10000g**的冲击，并有可编程的**低通滤波器**。



MPU6050模块

· CH340C:

USB转串口芯片，支持通讯波特率50bps ~ 2Mbps。

此型号内置时钟，无需外部晶振。

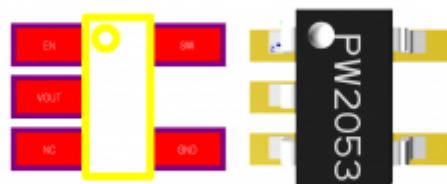


搭载了CH340C的串口下载模块

· PW2053:

PW2053是一款高效单片同步降压调节器。

[点我下载DataSheet](#)



没找到实物图.....

· TP4056:

TP4056 是一款完整的**单节锂离子电池**采用恒定电流/恒定电压**线性充电器**。



著名的充电模块

· WT8302:

WT8302系列是一款**超低EM.3.0W,单声道,D类音频功率放大器**。在5V电源下,能够向**4Ω负载提供3.0W**的输出功率,并具有高达**90%**的效率。



没找到实物图.....

PCB:

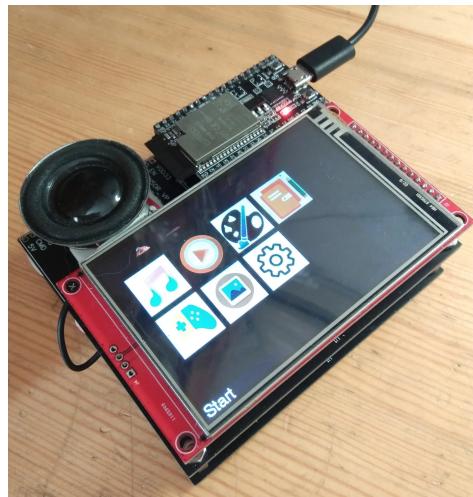
验证板:

只有**最基础硬件** (包括SHT30) , 注意 **GPIO2** 在烧录程序时需断开, 如果你要频繁烧录程序, 最好 PCB上飞线一个开关, 或者拔下开发板 (如果你愿意的话) 。

Bug:

- (1) T_CLK应连接IO0而不是IO25，已经在最新的PCB中**改正**。
- (2) 不要在意有两个ESP32开发板的插槽

[Gerber PCB ESP32开发直插板.zip](#)



验证板

最终板:

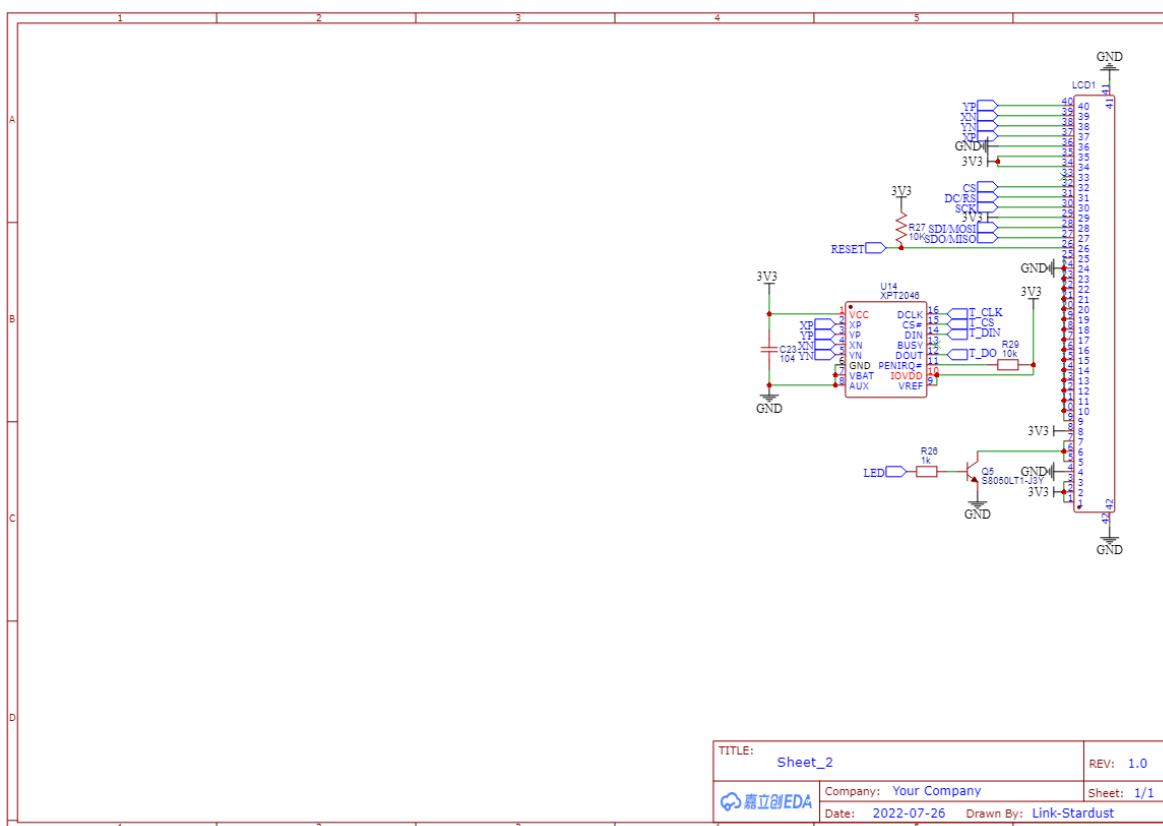
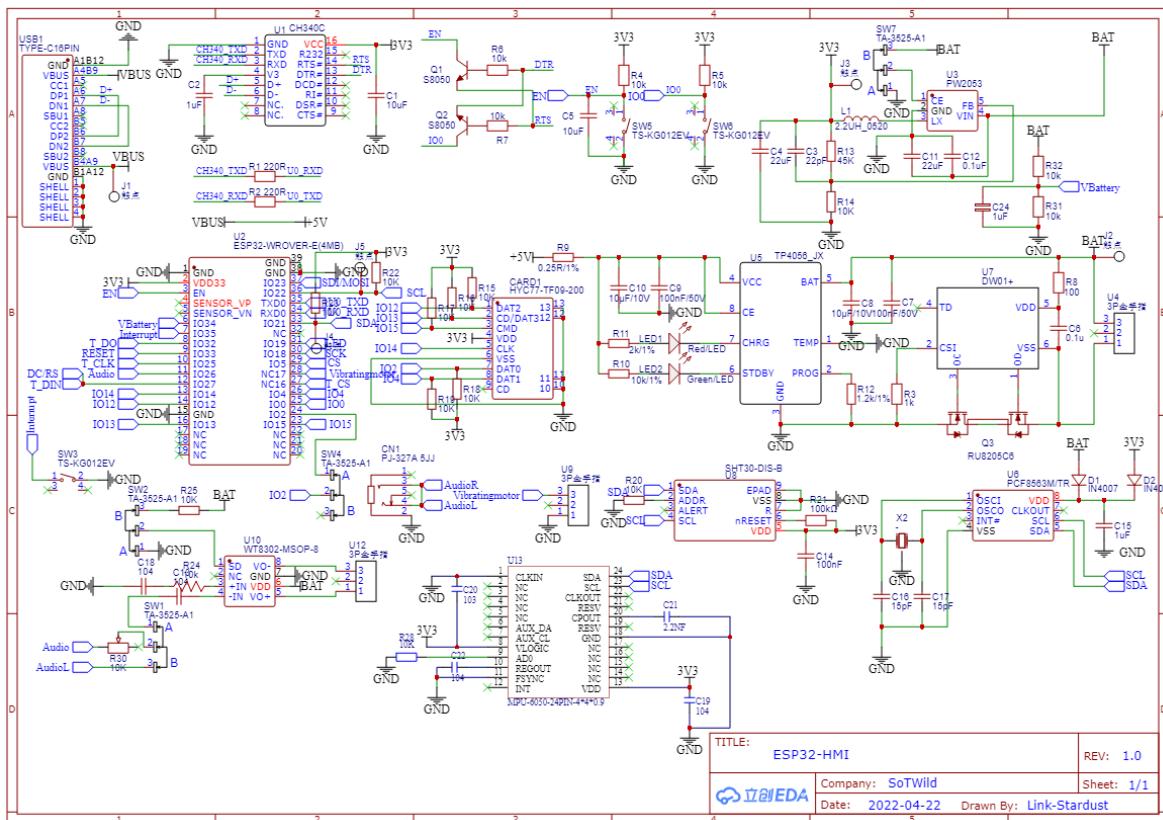
硬件齐全，自动烧录，配备电池。

Bug:

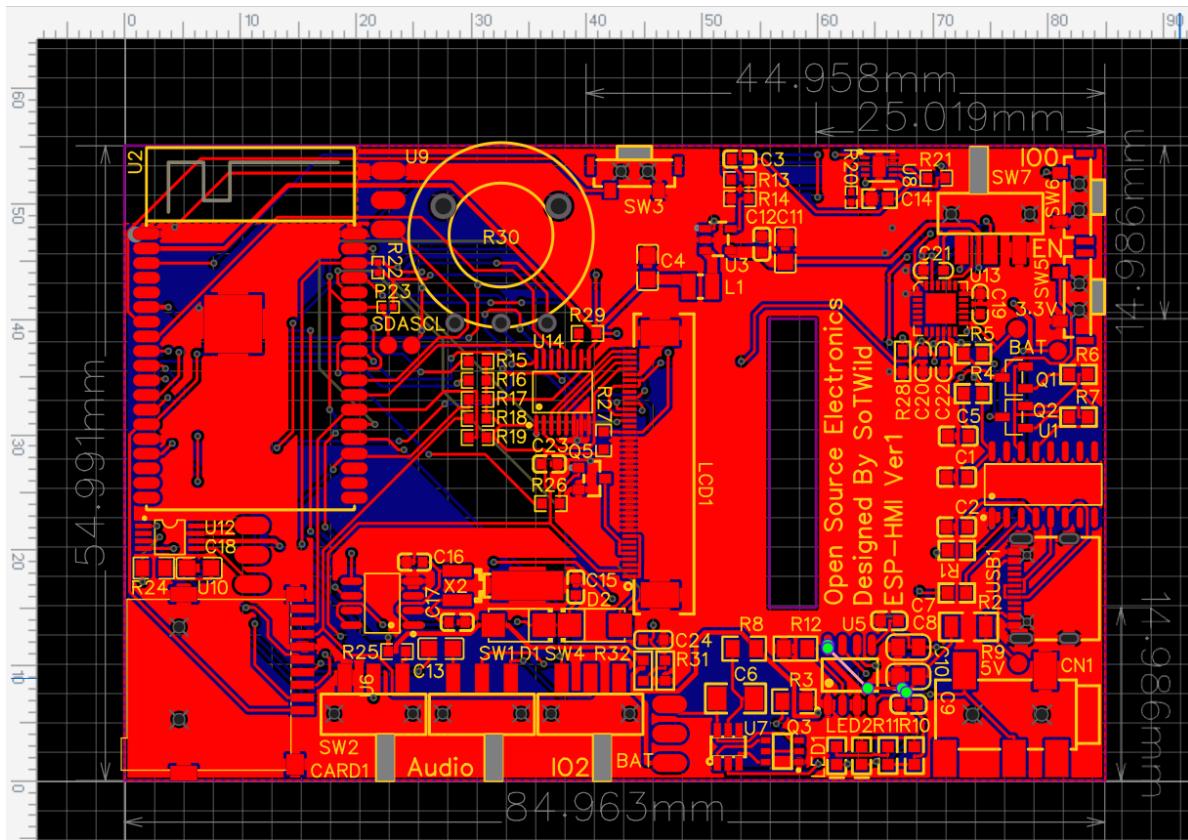
- (1) DC-DC电源部分电感封装大小错误。解决办法：可以勉强焊接
- (2) R15连接Strapping引脚导致无法启动。解决方法：不焊接R15
- (3) PCF8563设计缺陷，晶振波形错误。解决方法：重新设计
- (4) T_CLK应连接IO0而不是IO25，原理图已改正，PCB未改正。解决办法：飞线
- (5) WT8302设计缺陷，杂音。解决方法：重新设计

[Gerber文件下载](#)

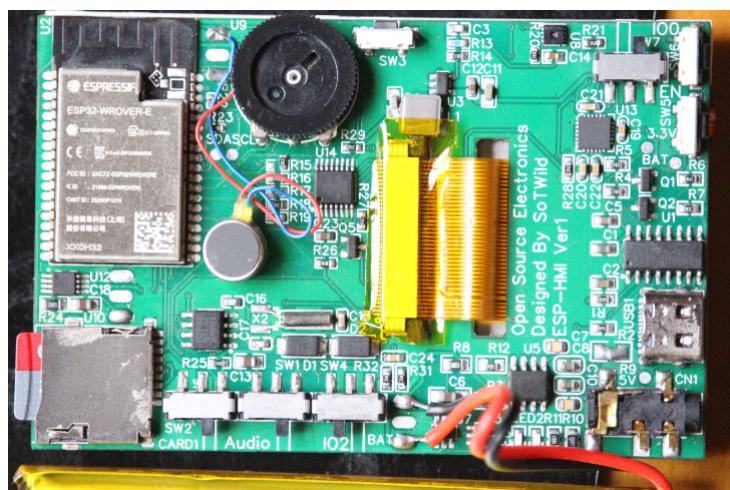
原理图:



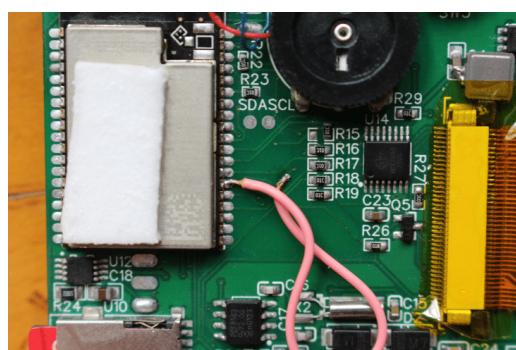
PCB图:



实物图：



未更正



飞线后

其他一些解读：

ESP-HMI配有极为先进的**USB Type-C**接口，作为供电和通信。

SW7为DC-DC芯片的使能开关，实现小开关控制大功率开关。

U9连接的为震动马达。

SW3连接**IO35**，可以自由配置为**中断引脚**或其他功能。

板载**5V、3.3V、BAT、SDA、SCL**触点，方便调试。

BOM表：

ID	Name	Designator	Footprint	Price
1	10uF	C1,C5	C0603	
2	1uF	C2	C0603	0.0391
3	22pF	C3	C0603	0.0573
4	0.1uF	C12	C0603	3.29
5	1uF	C15	C0603	0.0351
6	15pF	C16,C17	C0603	0.0883
7	22uF	C4,C11	C0805	3.29
8	0.1u	C6	1206	
9	100nF/50V	C7,C9	0603_C_JX	
10	10μF/10V	C8,C10	0805_C_JX	
11	104	C13,C18	C0805	
12	100nF	C14	603	0.0082
13	104	C19,C22	C 0603	
14	103	C20	C 0603	
15	2.2NF	C21	C 0603	
16	104	C23	C0603	
17	1uF	C24	C0603	0.032
18	HYC77-TF09-200	CARD1	TF-SMD_HYC77-TF09-200	0.9401
19	PJ-327A5JJ	CN1	AUDIO-SMD_PJ-327A5JJ	0.5698
20	IN4007	D1,D2	SMA_L4.3-W2.6-LS5.1-RD	
21	触点	5V,BAT,3.3V,SDA,SCL	触点	
22	2.2UH_0520	L1	IND-SMD_L2.5-W2.0	
23	AFC07-S40ECA-00	LCD1	FPC-SMD_P0.50-40P_LCS-SJ-H2.0	1.1266
24	Red/LED	LED1	0603_D_JX	
25	Green/LED	LED2	0603_D_JX	
26	S8050	Q1,Q2	SOT-23-3_L2.9-W1.3-P1.90-LS2.4-BR	0.0702
27	RU8205C6	Q3	SOT-23-6	0.3144
28	S8050LT1-J3Y	Q5	SOT-23_L2.9-W1.3-P1.90-LS2.4-BR	0.1207
29	220R	R1,R2	R0603	
30	10k	R4,R5,R6,R7,R29	R0603	
31	1k	R3	805	
32	100	R8	805	
33	0.25R/1%	R9	1206_R_JX	
34	10k/1%	R10	0603_R_JX	
35	2k/1%	R11	0603_R_JX	
36	1.2k/1%	R12	0805-R	
37	45K	R13	R0603	0.0135
38	10K	R14,R15,R16,R17,R18,R19,R25	R0603	0.0132
39	100kΩ	R21	R0603	0.0075
40	10K	R20,R22,R23	R0402	0.0038
41	10k	R24	R0805	
42	1k	R26	R0603	
43	10K	R27	R0603	
44	10K	R28	R 0603	
45	10K	R30	RES-ADJ-TH_R1001B	0.504
46	10k	R31,R32	R0603	0.0066
47	TA-3525-A1	SW1,SW2,SW4,SW7	SW-SMD_3P-L9.1-W3.5-P2.50-EH6.8	0.8578
48	TS-KG012EV	SW3,SW5,SW6	SW-SMD_TS-KG012EV	0.3602
49	CH340C	U1	SOP-16_L10.0-W3.9-P1.27-LS6.0-BL	2.66
50	ESP32-WROVER-E(4MB)	U2	WIFIM-SMD_39P-L31.4-W18.0-P1.27	33.11
51	PW2053	U3	SOT-23-5_L2.9-W1.6-P0.95-LS2.8-BL	
52	3P金手指	BAT,U9,U12	3P金手指	
53	TP4056_JX	U5	SOP8_150MIL_JX	
54	PCF8563M/TR	U6	SOP-8_L4.9-W3.9-P1.27-LS6.0-BL	1.1374
55	DW01+	U7	SOT-23-6-L	
56	SHT30-DIS-B	U8	DFN-8_L2.5-W2.5-P0.50-BL-EP	6.48
57	WT8302-MSOP-8	U10	MSOP-8_L3.0-W3.0-P0.65-LS4.9-BL	

ID	Name	Designator	Footprint	Price
58	MPU-6050-24PIN-440.9	U13	QFN24	
59	XPT2046	U14	TSSOP-16_L5.0-W4.4-P0.65-LS6.4-BL	3.31
60	TYPE-C16PIN	USB1	USB-C-SMD_TYPE-C16PIN	1.5593
61	-	X2	OSC-SMD_BD2.0-P2.54_TMXLI-206F32.768KHZ12.5PF20PPM	1.2412

ESP-HMI 软件

软件：

本人一切嵌入式皆为**自学**, 原谅我写成屎山或执行效率不高

主要函数：

Media Player.h 中:

```
1 | void Mjpeg_start(const char *MJPEG_FILENAME, const char *AUDIO_FILENAME);
```

这是进行 **Mjpeg** 视频播放, 源代码由[Play Video With ESP32](#)修改而来, 详情见[MjpegClass.h](#)。

```
1 | void drawsdjpeg(const char *filename, int xpos, int ypos);
```

这是绘制 **.jpg** 格式图片, 源代码由[Bodmer/JPEGDecoder](#)修改而来。

```
1 | void MP3_start(const char *filename);
```

这是进行 **.mp3** 格式音频的播放, 源代码由[ESP8266Audio](#)修改而来。

```
1 | void PCM_start(const char *AUDIOFILENAME);
```

这是进行 **.pcm** 格式音频的播放, 源代码由[Play Video With ESP32](#)修改而来。

```
1 | String readFileLine(const char* path, int num);
```

读取 **.txt** 文本的某一行, 源代码来自[peng-zhihui/HoloCubic](#)。

```
1 | void BleAudio();
```

蓝牙音频接收, 来自[ESP32-A2DP](#)。

```
1 | void CreateScrollWords(int IWIDTH, int IHEIGHT, int WAIT, int X, int Y, int  
size, int font, uint16_t TextColor, String msg);
```

创建滚动文字，修改自[TFT_eSPI](#)。

```
1 | void GetFromMometer();
```

从SHT30获得温湿度数据，修改自[SHT3x: Arduino library for Sensorion SHT3x](#)。

```
1 | void getNetTime();
```

获取网络时间，修改自[NTPClient](#)。

```
1 | void WiFiInit(string Mode);
```

初始化网络，可选STA/AP模式。

```
1 | void File_transfer_init();
```

无线文件传输功能，修改自[教你做一个ESP32-SD卡服务器哔哩哔哩bilibili](#)

[Main.h](#) 这是应用：

Sounder 音乐播放器

Vision 视频播放器

画图

Ebook 电子书阅读器

Game 小游戏

Album 相册

Settings 设置

Calculator 计算器、函数绘图器

Thermometer 温湿度计

文件上传

网络

任务管理器（资源查看器）

[MjpegClass.h](#) 这是mjpeg解码库。

[main.cpp](#) 开机时执行的程序，包括：

- 1) 加载串口
- 2) 屏幕初始化
- 3) 触摸初始化
- 4) 挂载SD卡
- 5) 连接 Wi-Fi
- 6) 读取设置文件
- 7) 用户登录
- 8) 运行 MainPage(); 主页程序

文件系统：

主要有两个文件夹：

System: 包含系统配置信息、应用图标、默认图标、控件图标等

User: 各用户文件

应用：

Sounder 音乐播放器：

需要两个文件夹，分别存放音乐JPG封面图片（可选）和MP3音乐文件。

注：封面图片大小需 $\leq 240 \times 240$ 。

- 1 .jpg的路径: /User/登录的用户名/Data/Music/MusicCover/xxx.jpg
- 2 .mp3的路径: /User/登录的用户名/Data/Music/MusicData/xxx.mp3

特点：只要添加（合适）名称的.mp3文件和同名的封面文件，系统即可**自动识别**音乐文件和封面文件。

核心实现方法：依靠以下函数每次**开机时读取指定位置的文件并将文件路径写入对应的配置文件**中。

然后系统调用 `readFileLine()`; 函数读取配置文件，得到目标文件位置。

```
1 | void listDir(fs::FS &fs, const char * dirname, uint8_t levels);
```

```
1 | void appendDir(fs::FS &fs, const char * dirname, String filename, uint8_t levels);
```

配置文件路径，分别是封面路径和音乐路径：

```
1 | /User/登录的用户名/Config/MusicCoverConfig.txt
2 | /User/登录的用户名/Config/MusicConfig.txt
```

注：系统首先通过截取字符串从读取到的文件路径中获取文件名，用户选中文件名后再填补出文件路径。

取字符串的前*i*个字符：`str=str.substring(0,i);`

去掉字符串的前*i*个字符：`str=str.substring(i);`

从右边开始取*i*个字符：`str=str.substring(str.length()-i);`

`str=str.substring(str.length()-i,str.length());`

从右边开始去掉*i*个字符：`str=str.substring(0,str.Length-i);`

常用的字符串截取方法

`sizeof(str)` 获取字符串长度。

`(str).c_str()` String转char。

Vision 视频播放器：

和Sounder的实现方法**大同小异**，添加了显示封面图片的功能。

必须要3个文件，分别是封面、**mjpeg**、**pcm**。

原因：系统读取 `videoCover` 文件夹内文件，根据封面文件名截取字符串得出文件名，后添加后缀得出**其余文件名**。

注：封面大小需 **≤ 180 x 100**。

- 1 .jpg的路径：`/User/登录的用户名/Data/Video/videoCover/xxx.jpg`
- 2 .mjpeg的路径：`/User/登录的用户名/Data/Video/videoData/xxx.mjpeg`
- 3 .pcm的路径：`/User/登录的用户名/Data/Video/videoData/xxx.pcm`

.mp4转.mjpeg的方法：下载[ffmpeg](#)，[Play Video With ESP32](#)，使用指令：

- ```
1 | ffmpeg -i input.mp4 -vf "fps=30" output.mjpeg
```

注：可以用ffmpeg调整视频序列设置或者提前准备合适的视频。

这是**.mp4转.pcm**：

- ```
1 | ffmpeg -i input.mp4 -f u16be -acodec pcm_u16le -ar 44100 -ac 1 output.pcm
```

画图：

修改自TFT_Touch库中的Draw实例。

Ebook 电子书阅读器：

和Sounder的实现方法**大同小异**，添加了显示封面图片的功能。

读取文件函数为 `String readFileLine(const char* path, int num);`

必须要两个文件:

注：封面大小需 **≤ 140 x 200**。

- 1 .jpg的路径: /User/登录的用户名/Data/Ebook/EbookCover/xxx.jpg
- 2 .mjpeg的路径: /User/登录的用户名/Data/Ebook/EbookData/xxx.txt

Game 小游戏:

移植了同人作品《玩弄蕾米莉亚的红色恶魔晚餐》中“咲夜时停搬物”游戏。

相当简单

文件路径:

- 1 /User/登录的用户名/Data/Game/TouHou/

Album 相册:

和Sounder的实现方法**大同小异**。

文件路径:

注：图片大小需 **≤ 480 x 320**。

- 1 /User/登录的用户名/Data/Image/

Settings 设置:

待修改完善。

Calculator计算器、函数绘图器:

核心函数 `string::toFloat(void)` 触摸键盘添加字符串，再将字符串转换为可运算的数据类型。

某些计算会有误差

函数绘图器未编写完成。

Thermometer温湿度计:

单纯刷屏

文件上传:

修改自[教你做一个ESP32-SD卡服务器](#)，添加了**在线查看功能**、**不同用户分区功能**、**用户名密码登录功能**。

使用**AP模式直连ESP32**时文件下载速度最高可达**350KB/s**，稳定在**330 ~ 350KB/s**。

使用**STA模式将ESP-HMI连接至路由器**时下载速度最高**327KB/s**，稳定在**200KB/s**（路由器旁）。
远离路由器时只有**18KB/s**，因此**推荐使用AP模式**。

文件保存路径：

1 | /User/登录的用户名/Data/upload/

网络：

可以选择AP模式或STA模式。

任务管理器（资源查看器）：

显示内容：

CPU：频率、运行时间

RAM：芯片RAM、SPI RAM

ROM：Flash大小、Flash速度、SD卡大小

Battery：电池电压、剩余百分比、预计使用时间

WiFi：当前模式、IP地址、SSID、MAC地址

Temperature：温度（不是CPU温度）

后续会添加真正意义上的“任务管理”功能。

ESP-HMI 注意事项：

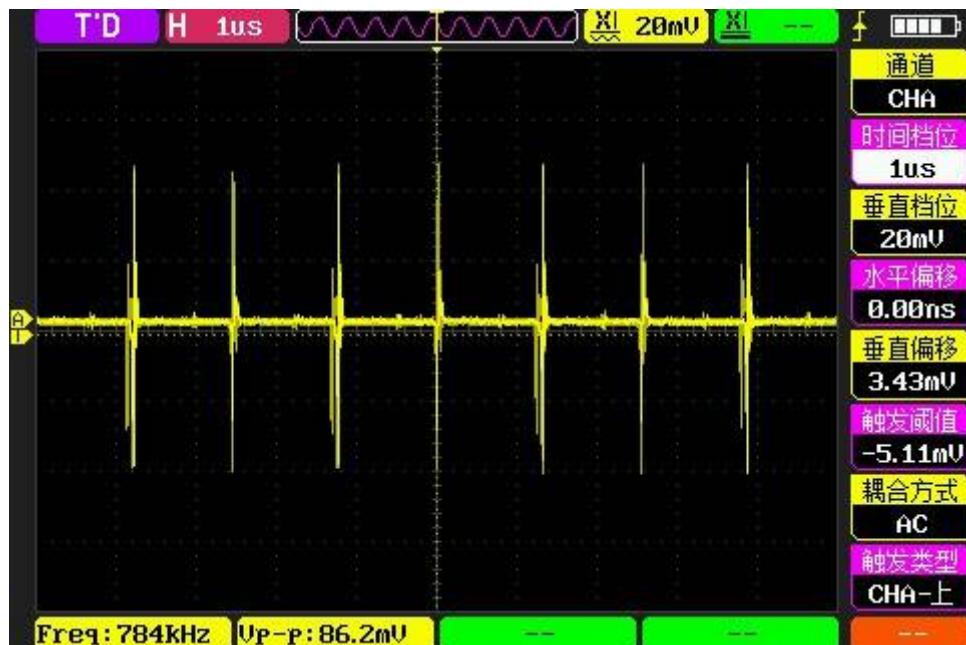
- 硬件：

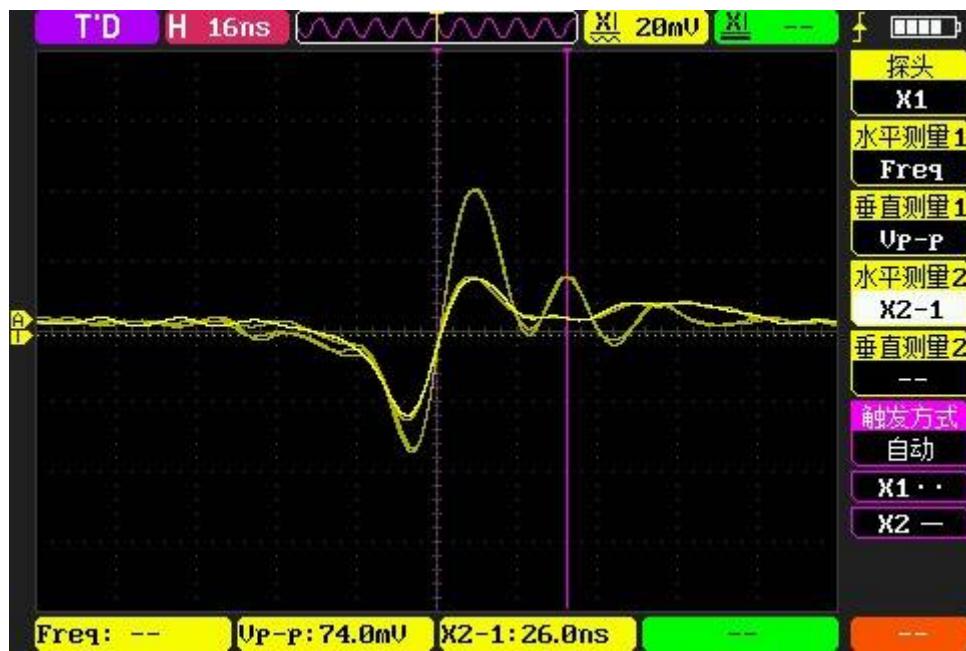
(1) PCB上要将IO25与XPT2046的T_CLK断开，将T_CLK和IO0连接。

原因：IO25和IO26为ESP32 DAC 的两个通道，音频输出时与触摸屏时钟信号冲突，会导致音频播放时（后）无法触摸。

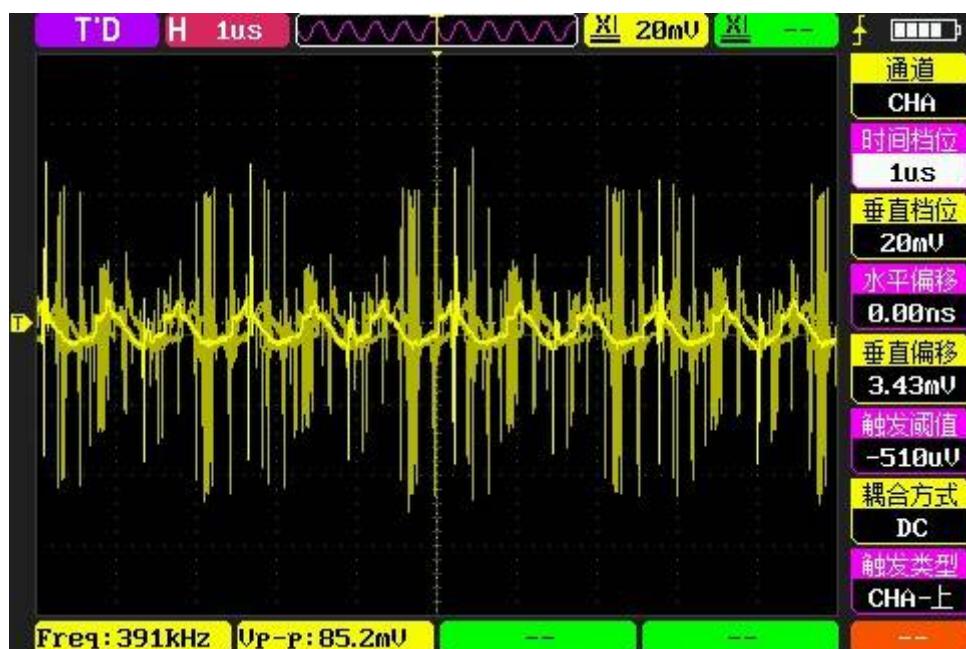
(2) 如果追求完美，建议重新设计WT8302和PCF8563的电路，并且将DC-DC设计在偏僻位置。

原因：原设计有缺陷，根据信号测量可以判定失败。





电源纹波



晶振波形，已经失控了

(3) 音频质量有限，但还是能接受。

原因：ESP32 DAC 精度为 **8-bit**，理论只可以输出**256**个值。



放大后的音频波形

(4) 在 (1) 中, 断开 IO25 后, 可以将其连接上耳机插座, 实现双声道。

原因: ESP32 中, IO25 为 **DAC Channel 1**, 是**左声道**; IO26 为 **DAC Channel 2**, 是**右声道**。
原设计中耳机的**两声道并联**, 由 IO26 提供**右声道**信号。

(5) 注意屏幕插座要买**插接上接式40p插座**, 而不是翻盖式。

原因: 翻盖式虽然好用, 但一开始设计是插接式, 没注意到, 是我踩过的坑。

- 软件:

(1) 使用了两个TFT库 (TFT_eSPI、Arduino GFX), 分别负责**绘制GUI**和**播放Mjpeg视频**。

原因: TFT_eSPI用得比较熟悉.....🐱

(2) 不同的触摸模块需要配置 (校准、设置旋转方向)

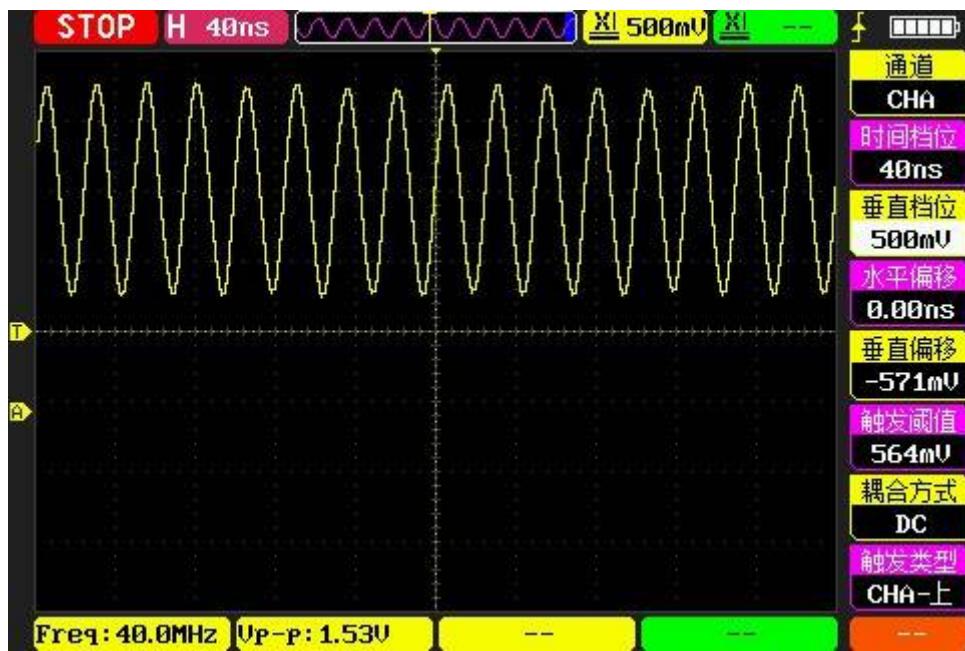
```

1 void setCal(uint16_t xmin, uint16_t xmax, uint16_t ymin, uint16_t ymax,
  uint16_t xres, uint16_t yres, boolean axis);
2 void setRotation(byte rotation);
```

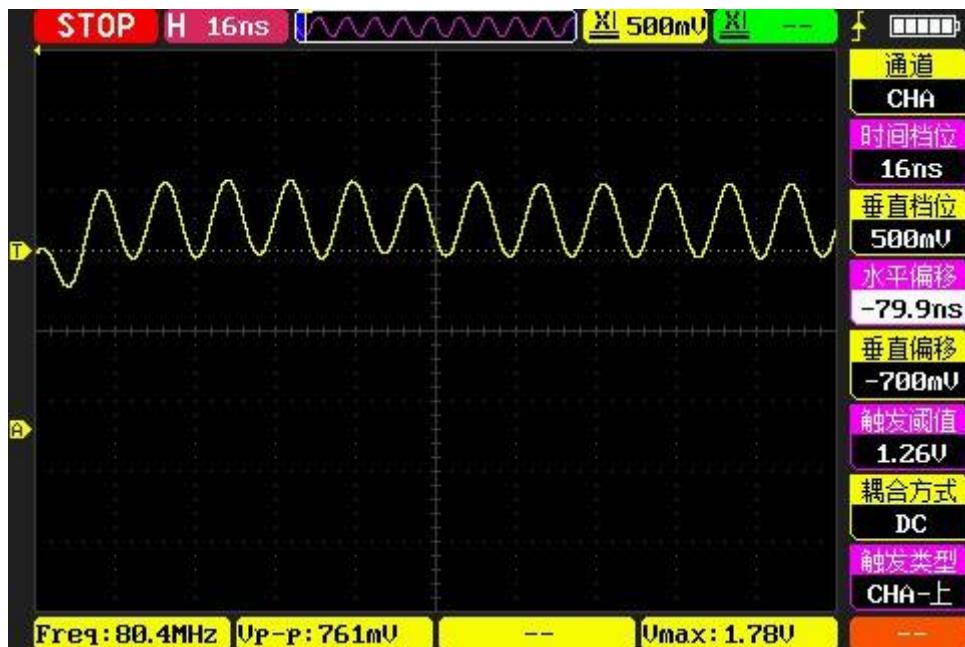
注: 代码中 `setRotation` 有多处需要修改。

(3) 在TFT_eSPI库 /User_Setups/Setup21_ILI9488.h 中, 设置了 `SPI_FREQUENCY` 为**40MHz**, 80MHz会出现**花屏**现象。

注: 若将屏幕更换为**ST7796**驱动芯片的类型再**优化PCB布线**可能会解决花屏问题 (待测试)



修改前的CLK信号：40MHz



修改后的CLK信号：80MHz

(4) 本项目文件夹中 SD 文件夹内名为 NewUserDemo 的文件夹用途是创建新用户时复制粘贴到 User 文件夹内以生成文件系统用的。

(5) **电池电压和百分比**的计算方法比较粗糙，百分比算法是：

$$\frac{\text{采样值} - \text{关机电压}}{\text{满电电压} - \text{关机电压}} \times 100\%$$

这里设置关机电压为3.5V。则代码算法是：

$$BatteryLevel = \frac{GetBatteryVol() - 3.5}{0.7(4.2V - 3.5V)} \times 100\%$$

如果要遵循电池放电曲线，提高精度，需要使用**积分**计算。

计算剩余使用时间的方法是：

$$\frac{\text{电池百分比} \times \text{电池容量}}{\text{系统耗电电流}} (\text{小时})$$

如果要计算到分钟，只要再 $\times 60$ 即可。

代码算法是：

$$GetBatteryRemainingTime = \frac{GetBatteryLevel() \times 4000}{150}$$

这里设置电池容量为4000mAh，耗电电流为150mA。

(6) 烧录完成后ESP32无限重启：重新烧录。

原因：未知。

搭建方法：

(1) 到[SoTwild/ESP-HMI: 基于 ESP32-WROVER 的人机界面 \(github.com\)](https://github.com/soTwild/ESP-HMI)这个页面选择 Code -> Download ZIP 下载压缩包。

如果出现**404**错误，可以在**本地**一个文件夹按**Shift** 再右键打开**Power Shell**窗口使用

```
1 | git clone https://github.com/soTwild/ESP-HMI.git
```

这个命令克隆存储库（前提是安装了git）。

压缩包文件：

ESP32-HMI：右键“**通过 Code 打开**”可以修改.ini配置文件后直接加载Platform IO编译上传。或者将 `include` 和 `src` 文件夹**合并**，将 `lib` 文件夹中的库导入，使用 **Arduino IDE** 打开。

Images：文档使用的图片

PCB：两版设计，包括原理图、PCB图和嘉立创EDA文件

SD：里面文件是TF卡内应装载的

LICENCE：GNU 协议

README.md：文档

README-EN.md：英文文档（还没有添加）

(2) 下载Gerber文件，送板厂打样

注：由于是4层PCB，若要黑色阻焊层，得加钱。

(3) 根据BOM表购买元件焊接

(4) 将SD文件夹中文件复制到准备好的TF卡内

(5) 上传程序，装配TF卡

(6) 调试

(7) 完成

其他：

Special Thanks:

[moononournation/Arduino_GFX: Arduino GFX developing for various color displays and various data bus interfaces \(github.com\)](#)

[Bodmer/JPEGDecoder: A JPEG decoder library \(github.com\)](#)

[earlephilhower/ESP8266Audio: Arduino library to play MOD, WAV, FLAC, MIDI, RTTTL, MP3, and AAC files on I2S DACs or with a software emulated delta-sigma DAC on the ESP8266 and ESP32 \(github.com\)](#)

[peng-zhihui/HoloCubic: 带网络功能的伪全息透明显示桌面站 \(github.com\)](#)

[pschattmann/ESP32-A2DP: A Simple ESP32 Bluetooth A2DP Library \(to implement a Music Receiver or Sender\) that supports Arduino, PlatformIO and Espressif IDF \(github.com\)](#)

[Bodmer/TFT_eSPI: Arduino and PlatformIO IDE compatible TFT library optimised for the Raspberry Pi Pico \(RP2040\), STM32, ESP8266 and ESP32 that supports different driver chips \(github.com\)](#)

[Risele/SHT3x: Arduino library for Sensorion SHT3x humidity and temperature sensors \(SHT30, SHT31, SHT35\). \(github.com\)](#)

[arduino-libraries/NTPClient: Connect to a NTP server \(github.com\)](#)

[教你做一个ESP32-SD卡服务器哔哩哔哩bilibili](#)

另外有些出处找不到了，可以的话请告知我。

结语：

该项目从**2021年09月05日**开始构思，目前为止基本要完成了。它几乎代表了我的**最高技术力**，希望该项目能为开源事业添砖加瓦。

在开发过程中，我也吸取到不少经验和教训，这里尽量分享出来，希望能帮助到各位。

