

百工谱——AIGC物品展示框

Artisan's Tome

- ▼ 百工谱——AIGC物品展示框
 - Artisan's Tome
 - 项目简介
 - 项目功能
 - 开源链接
 - 视频链接
 - ▼ 使用说明
 - 注册API
 - 网络设置
 - 结构设计
 - ▼ 硬件设计
 - 电源电路
 - 主控电路
 - 墨水屏驱动
 - ▼ 嵌入式软件
 - 网络连接、存储与配网
 - 休眠与唤醒
 - 网页访问
 - 墨水屏显示
 - ▼ 图像生成与处理
 - 运行平台与环境
 - Prompt设计
 - API调用与接口设计
 - 图像处理
 - 存储与下载
 - BOM表与费用
 - 参考资料

项目简介

一款基于ESP32的智能桌面艺术装置。

项目功能

本设备主控为ESP32 Pico-D4，基于预设库中时代、地域、职业等元素与文化符号，通过Kolors图像生成模型，创造出一件符合该时代与人物特征、独一无二的虚拟物品，通过墨水屏向用户展示，让每一次交互都成为在历史长河中的一次随机发现。

开源链接

todo

视频链接

todo

使用说明

注册API

设备使用前需注册API，并传入相应参数。

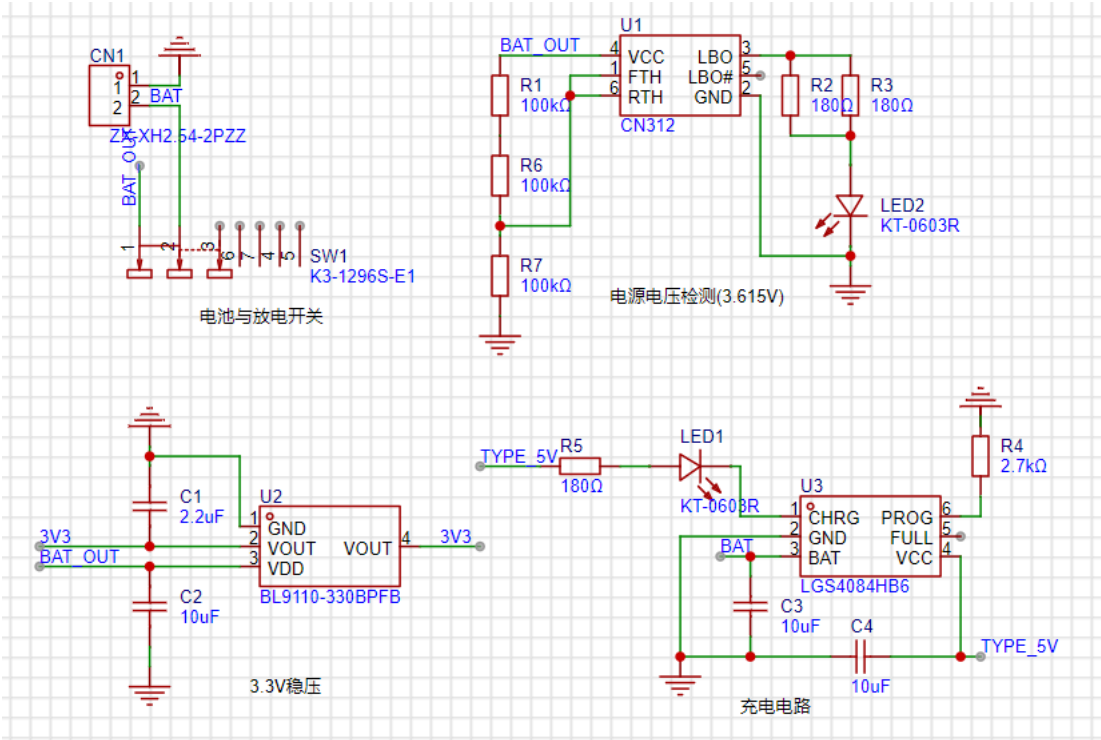
网络设置

结构设计

todo

硬件设计

电源电路



本设备采用3.7V锂电池供电，经线性稳压器BL9110-330BPFB输出3.3V，参考前作^[1]。

电源电压检测采用CN312，其内部主要包括电压比较器与基准电压源。简而言之，RTH引脚电压高于内部基准电压（1.205V）时LBO输出低电平，FTH低于基准电压时LBO输出高电平，LBO#引脚电平相反。方便起见，设置基准电压为3.605V，且FTH与RTH电压相等，则电源电压低于3.605V时LED亮起，反之熄灭。

嵌入式软件

网络连接、存储与配网

本项目支持WiFi连接与配网。
ESP32提供的Preferences库利用芯片内置的NVS，实现了数据的断电保留，可用于记录连接过的WiFi信息，实现自动连接。
设备每次成功连接后，记录WiFi的SSID与密码，在下次启动时将依次尝试连接所有记录的网络，直到成功或连接超时。
配网模式下，设备工作在AP模式，并维护一个HTTP服务器（页面使用html编写，储存在char*变量中）。用户连接ESP32的WiFi后，通过浏览器访问设置的固定地址 192.168.4.1:3000，即可进入配网页面。用户输入WiFi的SSID与密码后，点击提交按钮，设备将尝试连接该WiFi，若成功则自动重启，并保存输入的表单内容。

休眠与唤醒

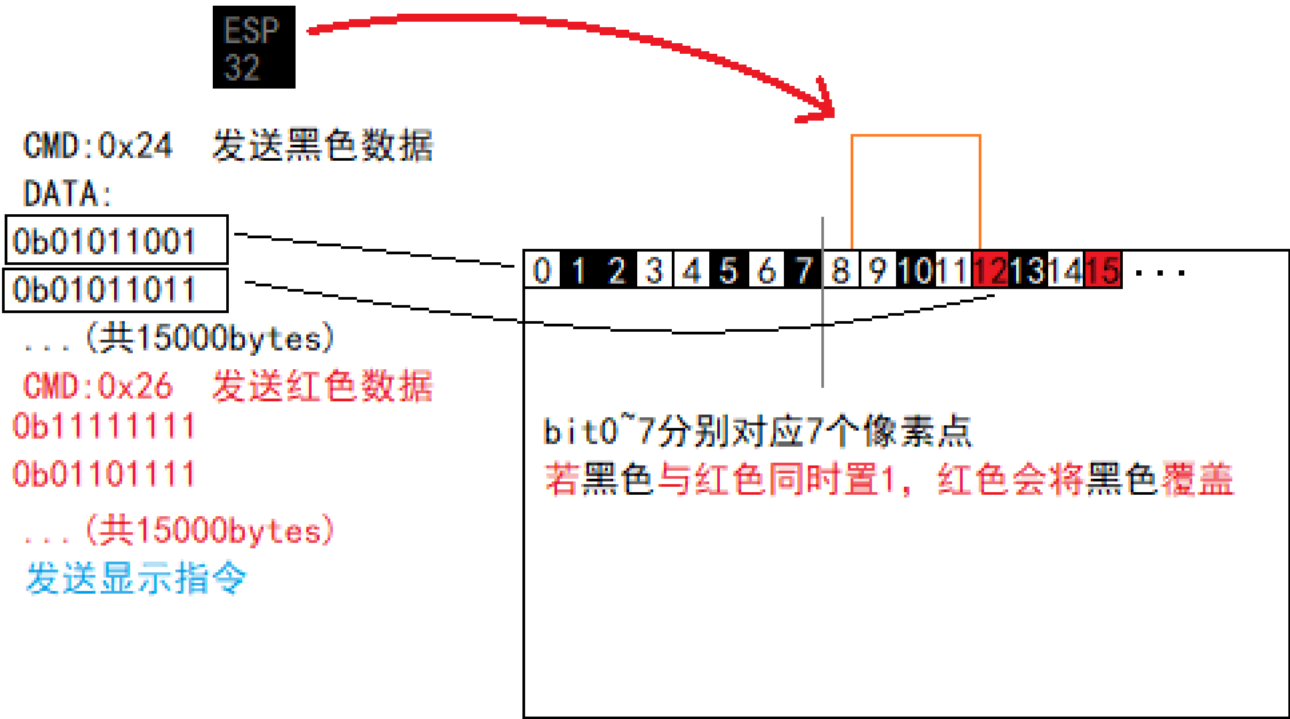
为保证设备在常开情况下能够长时间工作，在不需要生成图像时ESP32处于深度休眠状态，此时除RTC、ULP协处理器等外设几乎全部关闭，电流可降低到uA级别^[4]。
休眠后，可通过RTC GPIO唤醒，或通过RTC定时器唤醒，本项目中两方式均采用。
具体地，esp_sleep_enable_timer_wakeup(us)可配置设备在指定时间后自动唤醒，而 esp_sleep_enable_ext1_wakeup(pins, mode) 用于指定GPIO唤醒的引脚与模式（电平或边沿等），并在 esp_deep_sleep_start() 执行后进入休眠状态。
唤醒后 esp_sleep_get_wakeup_cause() 可获取唤醒原因，用于执行不同的操作。

网页访问

此部分参考^[1]即可。

墨水屏显示

项目采用汉硕WFH0420CZ35墨水屏，驱动程序可完全参考WaveShare 4.2inch e-Paper Module (B)^[5]，使用SPI接口进行通信，本项目使用GPIO模拟SPI时序。
本项目屏幕尺寸为400*300，若仅使用黑色显示，则图片所需字节为400*300/8=15000，在ESP32 Pico-D4中具有足够内存空间。
显示逻辑如下：
墨水屏为红黑白三色，因此需要将图片分为黑色、红色两部分。以黑色为例，每一位分别控制一个像素点（低电平为墨水显示），黑色会被红色覆盖。从左上角开始，15000字节依次对应从上到下、从左到右的像素点，低位在左。完成数据发送后，需向墨水屏发送全局刷新指令，更新图像。



若换用其他型号墨水屏，驱动程序需自行修改，但接口与代码框架大体类似。

图像生成与处理

运行平台与环境

图像生成与处理均在华为云FunctionGraph上运行^[1]。由于华为云的免费APIG服务已停止，但先前申请的APIG仍可使用，因此所有设备均调用已部署的API，传入用户自行配置的AI Token。
也可自己搭建服务器或使用类似服务，此时需要修改ESP32中的URL。

Prompt设计

使用DeepSeek大语言模型的目的是，获取一段可用于生图的Prompt。由于完全使用AI生成会使得内容趋于同质化，因此使用**随机数**产生可列举的元素。
为使AI生成的图像模型符合以下特征，Prompt采用**循序渐进**的方式，先根据职业生成人物设定，再根据设定生成物品，最后根据生成图Prompt与物品描述，且严格限制输出格式以便读取。
(该Prompt在后续可能会更新)

"你是一位游戏中的平民阶层角色，年龄" + age + "，背景为" + culture + "文化，" + era + "'时代。
请根据以下约束生成内容：
role: 你的具体职业（具体而简短），该职业类型属于'" + role + "'
item: '" + typ + "'（8字以内，不要带括号），
该场景下参考物品品质：草帽5/酒30/铁锄50/米10/绢200/牛1500，该物品品质为'" + price + "'

description: 一段描述性文字，涉及其特征、功能、来历、故事等，不使用第一人称
- 长度：保证在60字符以上、75字符以下（计算标点）
- 语言风格：'" + style + "'
- 不要带有emoji

prompt: 用于文生图的提示词
- 必须包含：物品材质+形态+颜色+细节特征，白色背景
- 禁止出现：拼接碎片、透视变形
- 需描述物品形态、材质、颜色、典型特征等
- 描述物品时语言需简洁而准确，不要出现歧义，物品名称可适当换成便于文生图理解的描述

严格按照以下示例输出json：
{
"role":string
"item": string
"description": string
"prompt": string
}
...

其中，随机元素已在代码中列举：

```

age = str(np.random.randint(12, 70))
role = np.random.choice(["农业/种植类", "工匠/工业/技术类", "养殖/畜牧类", "公共事务/法律类", ...
                           ])
if(np.random.random() < 0.8):
    era_selection = ["西周", "秦代", "汉代", ...]
    style_selection = [
        "铭文风格，参考先秦时期甲骨文和金文，简约古拙，多使用“唯”、“其”等古语，庄重肃穆。",
        "诗经风格，类似诗经的四言诗句，质朴口语化，带有比兴手法，多提及自然景物，生活气息浓厚。",
        ...
    ]
    era_index = np.random.randint(0, len(era_selection))
    era = era_selection[era_index]
    style = style_selection[era_index * 2 + np.random.randint(0, 2)]
else:
    era_selection = ["史前", "魔法时代", ...]
    style_selection = [
        "神话风格，来自史前，语言充满对自然力量的敬畏，将万物拟人化、神化，描述如创世史诗般宏大而神秘。",
        "岩画风格，极其简练、具象，如同刻在岩壁上的符号，只描述原始社会下动作、猎物和基本需求，原始粗犷。",
        ...
    ]
    era_index = np.random.randint(0, len(era_selection))
    era = era_selection[era_index]
    style = style_selection[era_index * 2 + np.random.randint(0, 2)]

culture = np.random.choice(["江南", "岭南", "巴蜀", "中原", ...])
price = str(int(10 ** (np.random.random() * 4)))
if(np.random.random() < 0.7):
    typ = "生成一项与你的职业特征强相关的工具，原料，产品或物品"
else:
    typ = "生成一项与你的职业弱相关或无关，但你可能会携带或使用的日常生活用品或个人配饰"

```

返回示例如下：

```

{
  'role': '风水罗盘匠',
  'item': '铜制风水盘',
  'prompt': '3D建模参考图，白色背景，等距视角，写实风格，铜制圆盘表面氧化泛青，中央太极阴阳鱼浮雕，外圈镌刻天干地支篆文，边缘有磨损包浆',
  'description': '祖传三代的黄铜罗盘，指针永远指向东南'}

```

Propmt用于图像生成，且传入Base64形式的以下参考图像以统一风格：



API调用与接口设计

为提高设备对不同AI的兼容性，以便在AI推陈出新时用户可选择更好的模型，同时避免多设备使用相同API导致短时间用量达到上限，本项目允许用户自定义API的URL与模型名称。

选用的大语言模型需采用OpenAI形式API，以火山引擎的豆包为例，POST 请求格式如下：

```
curl https://ark.cn-beijing.volces.com/api/v3/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $ARK_API_KEY" \
-d '{
  "model": "doubao-1-5-pro-32k-250115",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful assistant."
    },
    {
      "role": "user",
      "content": "Hello!"
    }
  ],
  "stream": false
}'
```

其中，请求头的 `Authorization` 字段为API密钥， `model` 字段为模型名称，与URL需作为访问FunctionGraph时的 `POST` 请求体参数传入。 `stream` 流式输出为 `false`，若部分API默认为`true`则需显式指定为`false`。

对于图像生成，各API的功能、接口存在差异，且此项目需要用到参考图像功能，因此固定采用硅基流动的 `Koolors` 模型（**免费**且每天可生成400张图像），仅需将API密钥传入即可。

在FunctionGraph中，提供的参数会被解析，并在调用API时相应传入。

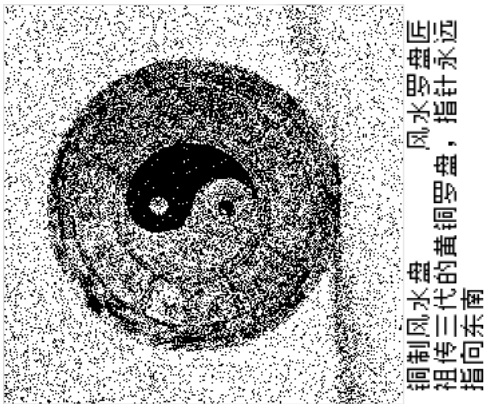
图像处理

为使彩色图像在黑白屏幕显示，需要将图像转为灰度图，并将尺寸缩小到300*300（原图为1:1）。此后对图片进行随机抖动处理：

设像素点的灰度值为 $g \in [0, 1]$ ，则转换后灰度值为 $h = g > rand() ? 1 : 0$ ，其中 $rand()$ 服从 $[0, 1]$ 上的均匀分布。

完成后，在图像上添加文字，包括物品名称、拥有者职业与物品描述，字体采用Zpix。完整图像将进行旋转以适应屏幕方向，编码为15000字节的文件，对应墨水屏驱动时依次传递的120000像素（仅黑色）。

示例如下：



存储与下载

文件上传至阿里云对象存储（OSS）指定路径中（“`a.img`”）。若用户自行使用更大的显示屏，可将内容分为多个小文件，ESP32多次调用OSS下载URL分别下载，或探索其他方法。

由于FunctionGraph试图导入阿里云SDK-oss2没成功，代码中直接使用requests库进行上传，需要手动构建签名信息。

在名为 `bucket_name` 的Bucket中创建路径为 `path` 的文件，即使用 `POST` 方法访问 `https://{bucket_name}.{endpoint}/{path}`，则请求头需满足以下形式：

```
headers = {
    'Authorization': f'OSS {access_key_id}:{signature}',
    'Date': date_str, # 类似Tue, 30 Sep 2025 02:58:16 GMT
    'Content-Type': content_type
}
```

`signature` 通过以下形式构建：

```
bucket_name = 'your_bucket_name'
endpoint = 'your_endpoint' # 如oss-cn-shenzhen.aliyuncs.com
method = 'PUT'
content_type = 'application/octet-stream' # 二进制文件
date_str = email.utils.formatdate(usegmt=True)
access_key_id = str(os.getenv('OSS_ACCESS_KEY_ID'))
access_key_key = str(os.getenv('OSS_ACCESS_KEY_SECRET'))
canonicalized_oss_headers = ''

canonicalized_resource = f'/{bucket_name}/{path}'
string_to_sign = f"{method}\n\n{content_type}\n{date_str}\n{canonicalized_oss_headers}{canonicalized_resource}"
h = hmac.new(
    access_key_key.encode('utf-8'),
    string_to_sign.encode('utf-8'),
    hashlib.sha1
)
signature = base64.b64encode(h.digest()).strip().decode('utf-8')
```

OSS_ACCESS_KEY_ID 与 OSS_ACCESS_KEY_SECRET 为阿里云OSS的Access Key ID与Access Key Secret，可在阿里云控制台获取：

账号 >

退出登录

账号 ID:

主账号 个人认证

权限与安全

安全管控 访问控制 AccessKey

费用与成本 >

可用额度

充值汇款

不建议使用云账号 AccessKey

云账号 AccessKey 具有账号的完全权限，且不能做权限控制。因保管和使用不当，导致云账号 AccessKey 被他人利用，将会对账号资源造成巨大的安全威胁。

使用临时凭证

建议使用 STS Token 临时凭证方案进行程序访问，减少 AccessKey 泄漏风险，无需定期轮换，同时支持更灵活和精细的云资源授权。 [了解更多](#)

使用 RAM 用户 AccessKey

必须使用 AccessKey 的情况，建议使用 RAM 用户 AccessKey，并进行最小化授权。 [了解更多](#)

☒ 我确认知晓云账号 AccessKey 安全风险

使用 RAM 用户 AccessKey

继续使用云账号 AccessKey

身份管理

用户

用户组

角色

权限管理

授权

RAM 用户是 RAM 中的一种实体身份，代表需要访问云的人员或应用。如您对于 RAM 用户的使用情况做检测和治理，可移步身份权限治理中心。

创建用户

筛选登录名、显示名、用户 ID 或 AccessKey ID

☐ 用户登录名称/显示名称

备注

创建用户

用户账号信息

* 登录名称

+ 添加用户

访问方式

☐ 控制台访问 用户使用账号密码访问云控制台

☒ 使用永久 AccessKey 访问 创建 AccessKey ID 和 AccessKey Secret

确定

返回

创建用户

若开通 OpenAPI 调用访问，请及时保存 AccessKey 信息，页面关闭后将无法再次获取信息。

用户信息

下载 CSV 文件

<input type="checkbox"/> 用户登录名称	状态	启用控制台登录	登录密码	AccessKey ID	AccessKey Secret	操作
<input type="checkbox"/> 	创建用户: 成功 开启 OpenAPI 调用访问: 成功	否	无	<input checked="" type="checkbox"/> 复制	<input checked="" type="checkbox"/> 复制	<input checked="" type="checkbox"/> 复制
<input type="checkbox"/> 						

☐ 添加到用户组

☐ 添加权限

☐

手动创建

☐

添加到用户组

添加权限

删除

新增授权

☒ 账号级别 ☐ 资源组级别

授权主体

已选择授权主体

权限策略

oss

所有策略类型

<input checked="" type="checkbox"/> 策略名称	策略类型	描述
<input checked="" type="checkbox"/> AliyunOSSFullAccess	系统策略	管理对象存储服务 (OSS) 权限
<input type="checkbox"/> AliyunOSSReadOnlyAccess	系统策略	只读访问对象存储服务 (OSS) ...
<input type="checkbox"/> AliyunYundunNewBGPAntiDDoS	系统策略	管理云盾DDoS高防 (中国内地...)
<input type="checkbox"/> AliyunYundunNewBGPAntiDDoS	系统策略	只读访问云盾DDoS高防 (中国...)

已选择权限策略

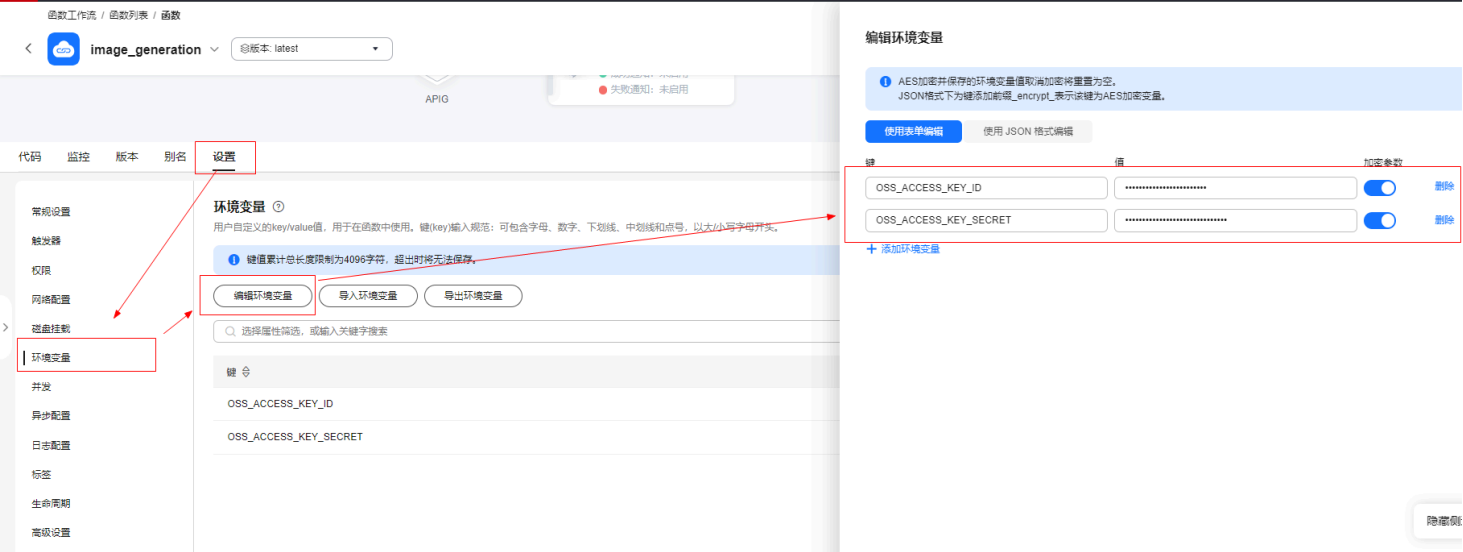
系统策略 (1)

AliyunOSSFullAccess

确认新增授权

取消

注意：以上方式获取Access Key时，关闭第⑤步的页面后将无法再次查看！
完成步骤后，将其配置在FunctionGraph的环境变量中。



由于Bucket权限设置为公共读，因此可直接在ESP32 GET 访问 `https://{bucket_name}.{endpoint}/{path}` 获取图片。云端通过向下位机发送生成成功信息（`status=0`）后提示下载。

BOM表与费用

以下部分元件立创商城与其它途径相差较大，在 `Supplier` 一栏中未写 `LCSC` 则为其它途径价格。

No.	Quantity	Comment	Designator	Footprint	Value	Manufacturer Part	Manufacturer
1	1	2.2uF	C1	C0402	2.2uF	HGC0402R5225K160NTEJ	Chinocera(华瓷)
2	1	10uF	C2	C0402	10uF	CGA0402X5R106M100GT	HRE(芯声)
3	4	4.7uF	C3,C4,C6,C11	C0402	4.7uF	CL05A475MP5NRNC	SAMSUNG(三星)
4	10	10uF	C5,C8,C9,C12,C14,C15,C16,C17,C18,C19	C0402	10uF	CL05A106MQ5NUNC	SAMSUNG(三星)
5	4	100nF	C7,C10,C13,C20	C0402	100nF	CL05B104KO5NNNC	SAMSUNG(三星)
6	1	ZX-XH2.54-2PZZ	CN1	CONN-TH_2P-P2.50_HX25003-2A		ZX-XH2.54-2PZZ	Megastar(兆星)
7	3	SS36	D1,D2,D3	DO-214AC_L5.0-W2.5-LS5.0-FD		SS36	R+O(宏嘉诚)
8	1	U.FL-R-SMT-1(80)	JP1	RF-SMD_FRF05002-JSS103M		U.FL-R-SMT-1(80)	HRS(广瀚)
9	1	1uH	L1	IND-SMD_L2.0-W1.6-B	1uH	FTC201610S1R0MBCA	cjiang(长江微电)
10	2	KT-0603R	LED1,LED2	LED0603-RD		KT-0603R	KENTO
11	1	NH-B1010RGBT-HF	LED3	LED-ARRAY-SMD_4P-L1.0-W1.0-TR_NH-B1010RGBT		NH-B1010RGBT-HF	国星光电

No.	Quantity	Comment	Designator	Footprint	Value	Manufacturer Part	Manufacturer
12	1	HL2300	Q1	SOT-23-3_L2.9-W1.3-P1.90-LS2.4-BR		HL2300	R+O(宏嘉诚)
13	2	S8050	Q2,Q3	SOT-23-3_L2.9-W1.6-P1.90-LS2.8-BR		S8050	Hottech(合科泰)
14	9	10kΩ	R1,R6,R7,R8,R9,R13,R15,R16,R17	R0402	10kΩ	0402WGF1002TCE	UNI-ROYAL(厚声)
15	3	180Ω	R2,R3,R5	R0402	180Ω	0402WGF1800TCE	UNI-ROYAL(厚声)
16	1	2.7kΩ	R4	R0402	2.7kΩ	0402WGF2701TCE	UNI-ROYAL(厚声)
17	1	470mΩ	R10	R0402	470mΩ	0402WGF470LTCE	UNI-ROYAL(厚声)
18	3	500Ω	R11,R12,R14	R0402	500Ω	RT0402FRE07500RL	YAGEO(国巨)
19	1	K3-1296S-E1	SW1	SW-TH_K3-1296S-E1		K3-1296S-E1	韩国韩荣
20	2	TS-1088-AR02016	SW2,SW3	SW-SMD_L3.9-W3.0-P4.45		TS-1088-AR02016	XUNPU(讯普)
21	1	CN312	U1	SOT-23-6_L2.9-W1.6-P0.95-LS2.8-BL		CN312	CONSONANCE(上海如暗)
22	1	BL9110-330BPFB	U2	SOT-223_L6.5-W3.5-P2.30-LS7.0-BR		BL9110-330BPFB	BL(上海贝岭)
23	1	LGS4084HB6	U3	SOT-23-6_L2.9-W1.6-P0.95-LS2.8-BL		LGS4084HB6	Legend-Si(棱晶半导体)
24	1	FPC 0.5-24P HYH2.0	U4	FPC-SMD_24P-P0.50_FPC-0.5-24P-HYH2.0		FPC 0.5-24P HYH2.0	SHOU HAN(首韩)
25	1	ESP32-PICO-D4	U5	ESP32-PICO-D4_lengthen	-	ESP32-PICO-D4	ESPRESSIF(乐鑫)
26	1	CH340K	U6	ESOP-10_L4.9-W3.9-P1.00-LS6.0-BL-EP		CH340K	WCH(南京沁恒)
27	1	TYPE-C 16PIN 2MD(073)	USB1	USB-C-SMD_TYPE-C-6PIN-2MD-073		TYPE-C 16PIN 2MD(073)	SHOU HAN(首韩)

以上共16.73元（不计运费）。

墨水屏建议选用二手（淘宝上很多，但尽量选没有老化的），400*300价格在20~60都有，当然也可以选择不同尺寸的（代码需自行适配）。

PCB为33元（可使用嘉立创免费打样），外壳3D打印为（待定）元。

墨水屏按30元计算，不计PCB，共计成本（待定）元。

参考资料

- [1] 【星火计划】骑行导航辅助显示 https://oshwhub.com/ljh345/cycling_assistance。
- [2] 墨水屏驱动板 有示例 <https://e.tb.cn/h.SVfh0rEaqfkrB5y?tk=m9hz4DtTtNR>。

- [3] 4.2inch e-Paper Module (B) Manual https://www.waveshare.net/w/upload/9/97/4.2inch_e-Paper_Schematic.pdf.
- [4] Arduino-ESP32深度睡眠模式：超低功耗设计与唤醒策略 https://blog.csdn.net/gitblog_00419/article/details/151563516.
- [5] [other] 两款汉朔4.2寸墨水屏电子价签拆解与点亮，型号都是Stellar-X <https://www.mydigit.cn/forum.php?mod=viewthread&tid=274369&page=1&authorid=1055519>.