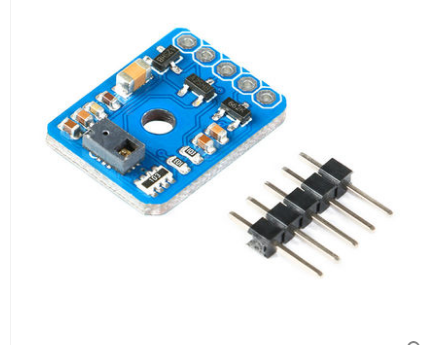
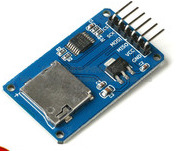
ESP32网络相册

**一、系统硬件：**

一块ESP32开发板、一块TFT显示屏（驱动ILI9341、不用触摸、尺寸自选，我用的是240\*320，2.8寸）、Micro SD卡读取模块、micro SD卡（我测试的时候2G的挂载成功但是无法读取，最后用的128M的）、PAJ7620手势传感器

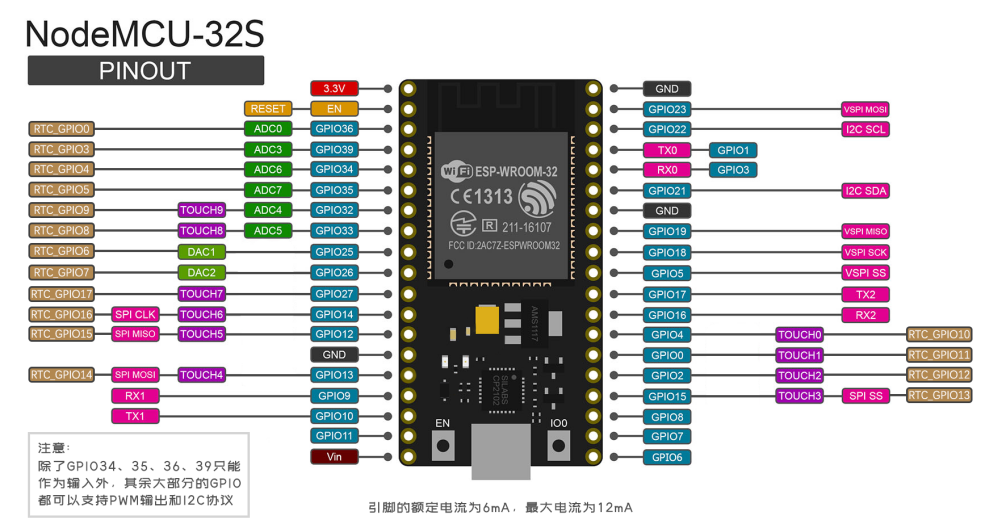
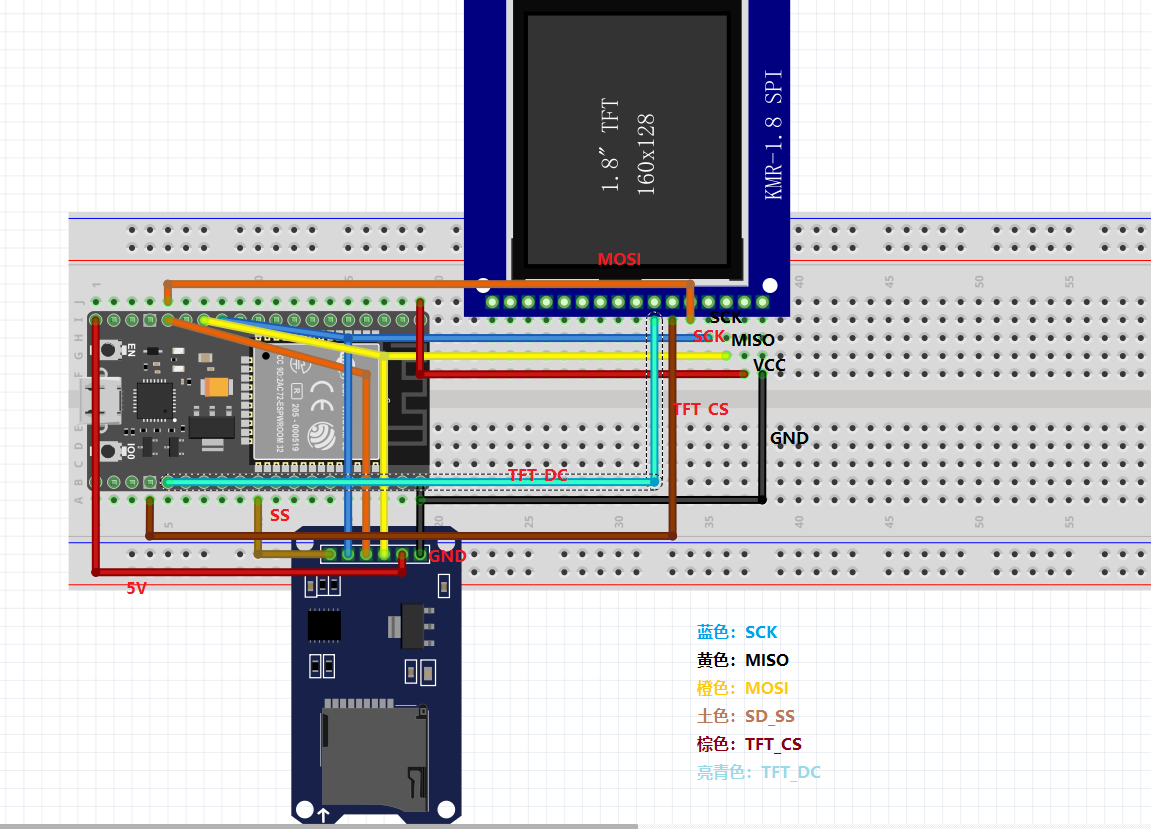




ESP32模块作为主控，实现天气信息获取和将SD卡内图片推送到TFT屏幕上。

**二、接线：**

因为TFT屏幕与SD卡共用一套SPI，所以他们的一些引脚是接在一起的，只需要控制片选引脚即可，PAJ7620使用的是IIC总线(PAJ7620未在图中绘出)，在源码里我注释了接线的引脚。



**三、简单的网页知识：**

首先，假设我们想访问达尔闻首页网站，我们就要在浏览器地址栏输入达尔闻的网站地址: <https://darwinlearns.com>。当您输入完达尔闻首页地址并按下回车以后，您所用的浏览器会通过DNS服务查到达尔闻网站服务器的IP地址。我们假设达尔闻服务器地址为12.34.56.78。接下来浏览器就会向IP地址12.34.56.78的服务器发送http请求。当网站服务器收到了请求后，会把被请求的网页信息传输给浏览器，然后浏览器就会把收到的网页信息转换成网页显示在浏览器中。一般网页首先会请求网页的根网页，也就是在网页最后加一个“/”。

我们所建立的文件上传网页也是这样来通信的，如果您还会一点点前端知识，那么这个上传文件的网页您完全可以写的更加漂亮。

**四、ESP32与手机通信代码：**

**1.首先需要调用以下3个库：**

#include <WiFi.h>

#include <WiFiClient.h>

#include <WebServer.h>

**2.创建监听对象：**

WebServer server(80);//监听80服务端口

**3.setup( )函数中需要的内容：**

//SERVER INIT 客户端服务初始化

server.on("/", HTTP\_GET, handleRoot);//当访问根网页时跳转到handleRoot函数

server.on("/", HTTP\_POST,[](){uplaodFinish();}, handleFileUpload);//有文件上传时由handleFileUpload()函数处理，上传完成后由uplaodFinish()函数处理

server.onNotFound([](){if(!handleFileRead(server.uri()))server.send(404, "text/plain","FileNotFound");}); //访问的url没有找到回复404页面

server.begin();//网络服务开启

**4.loop( )函数中调用：**

server.handleClient();//处理网页请求函数，需不断调用

**5.handleRoot()函数处理访问根网页请求：**

void handleRoot()

{

server.send(200, "text/html", mainPageString);//将html网页发送到网页请求端，mainPageString数组内存放的是html的内容

server.client().stop();//停止处理网页请求，防止同时多个访问

}

```

**6.handleFileUpload()函数将上传的文件下载到SD卡内：**

void handleFileUpload(){//网络文件下载服务处理函数

if(server.uri() != "/") return;//若访问根网页就返回

HTTPUpload& upload = server.upload();//获取上传的文件

String filename;

char \*file\_sd;

if(upload.status == UPLOAD\_FILE\_START){//开启下载上传的文件

filename = upload.filename;//字符串filename等于上传的文件名

if(!filename.startsWith("/"))

{

filename = "/loge"+String(tft\_num)+".jpg";//如果文件开头没有/则添加/ 并且对文件名中添加计数字符 tft\_num

tft\_num++;//文件标记数+1

EEPROM.write(20,tft\_num);//将文件（照片）数量数据保存到内存，方便开机调用

EEPROM.commit();//写入内存

}

Serial.print("handleFileUpload Name: "); Serial.println(filename);//打印文件名

SD.remove(filename);//删除SD卡内文件名与字符串filename重名的文件

fsUploadFile = SD.open(filename, "w");//创建名为字符串filename的文件

filename = String();//字符串重新赋空

} else if(upload.status == UPLOAD\_FILE\_WRITE){//文件处于写入状态

if(fsUploadFile)

fsUploadFile.write(upload.buf, upload.currentSize);//将上传文件写入SD卡

} else if(upload.status == UPLOAD\_FILE\_END){//文件上传处于完成状态

if(fsUploadFile)

fsUploadFile.close();//关闭文件

}

}

**7.uplaodFinish()函数处理文件下载完成，发送完成网页**

void uplaodFinish() {

server.send(200, "text/html", uploadString);//发送上传完成网页

}

1. handleFileRead()函数负责处理资源访问，当网页请求图片等文件时此函数处理

bool handleFileRead(String path){

int upload\_ = tft\_num - 1;

if(upload\_<0)

upload\_ = 0;

path = "/loge"+String(upload\_)+".jpg";//定向请求的路径

Serial.println("handleFileRead: " + path);//串口打印 请求文件的路径

if(path.endsWith("/")) path += "index.htm";//对文件类型更改

String contentType = getContentType(path);

String pathWithGz = path + ".gz";

if(SD.exists(pathWithGz) || SD.exists(path)){

if(SD.exists(pathWithGz))

path += ".gz";//将文件后缀以压缩文件发送会比较快

File file = SD.open(path, "r");//打开请求的网页

size\_t sent = server.streamFile(file, contentType);//发送文件到请求的网页

file.close();//关闭文件

return true;//网页请求的文件找到

}

return false;//网页请求的文件没有找到

}

**五、ESP32心知天气信息获取解析：**

**1.首先需要调用以下2个库：**

#include <WiFi.h>

#include <ArduinoJson.h>//使用JSON-v5版的库

**2.向心知天气服务器发送问询代码：**

if(client.connect("api.seniverse.com",80)==1)//连接服务器并判断是否连接成功，若成功就发送GET 请求数据下发

{//换成你自己在心知天气申请的私钥//改成你所在城市的拼音

//client.print("GET /v3/weather/now.json?key=\*\*\*\*\*\*\*\*&location=beijing&language=zh-Hans&unit=c HTTP/1.1\r\n"); //心知天气当天的URL格式\*\*\*\*处为个人申请密匙

client.print("GET /v3/weather/daily.json?key=\*\*\*\*\*\*\*\*&location=beijing&language=zh-Hans&unit=c HTTP/1.1\r\n"); //心知天气的三天天气URL格式\*\*\*\*处为个人申请密匙

client.print("Host:api.seniverse.com\r\n");

client.print("Accept-Language:zh-cn\r\n");

client.print("Connection:close\r\n\r\n"); //向心知天气的服务器发送请求。

String status\_code = client.readStringUntil('\r'); //读取GET数据，服务器返回的状态码，若成功则返回状态码200

if(client.find("{")==1)//跳过返回的数据头，直接读取后面的JSON数据

{

String json\_from\_server=client.readStringUntil(']'); //读取返回的JSON数据

json\_from\_server = "{"+json\_from\_server+"]}]}";//对数据进行补全

Serial.println(json\_from\_server);//打印得到的数据信息

parseUserData(json\_from\_server);//将读取的JSON数据，传送到JSON解析函数中进行显示。

}

else

{

Serial.println("Not find.");

}

}

else

{

Serial.println("connection failed this time");

delay(5000); //请求失败等5秒

}

client.stop(); //关闭HTTP客户端，采用HTTP短链接，数据请求完毕后要客户端要主动断开

delay(50000);

1. **Json数据解析并串口打印：**

int weather\_num[3];

DynamicJsonBuffer jsonBuffer(1660);//申请解析数据的大小

JsonObject& root = jsonBuffer.parseObject(content);

JsonObject& results\_0 = root["results"][0];

JsonObject& results\_daily0 = results\_0["daily"][0];

JsonObject& results\_daily1 = results\_0["daily"][1];

JsonObject& results\_daily2 = results\_0["daily"][2];

const char\* results\_0\_now\_data = results\_daily0["date"];//天气时间

const char\* results\_0\_now\_text = results\_daily0["text\_day"];//天气时间

const char\* results\_0\_now\_code = results\_daily0["code\_day"];//天气现象数值

const char\* results\_1\_now\_code = results\_daily1["code\_day"];//天气现象数值

const char\* results\_2\_now\_code = results\_daily2["code\_day"];//天气现象数值

String high\_tem\_0= results\_daily0["high"];//天气温度最高数值

String high\_tem\_1= results\_daily1["high"];//天气温度最高数值

String high\_tem\_2= results\_daily2["high"];//天气温度最高数值

String low\_tem\_0= results\_daily0["low"];//天气温度最低数值

String low\_tem\_1= results\_daily1["low"];//天气温度最低数值

String low\_tem\_2= results\_daily2["low"];//天气温度最低数值

const char\* hum= results\_daily0["humidity"];//天气湿度数值

const char\* wind\_speed0 = results\_daily0["wind\_speed"];//天气风速数值

const char\* wind\_speed1 = results\_daily1["wind\_speed"];//天气风速数值

const char\* wind\_speed2 = results\_daily2["wind\_speed"];//天气风速数值

const char\* rain= results\_daily0["rainfall"];//天气降雨量数值

//atoi()函数将字符转换为数字

weather\_num[0] = atoi(results\_0\_now\_code);

weather\_num[1] = atoi(results\_1\_now\_code);

weather\_num[2] = atoi(results\_2\_now\_code);

Serial.println(results\_0\_now\_code); //通过串口打印出需要的信息

Serial.println(results\_1\_now\_code);

Serial.println(results\_2\_now\_code);

Serial.print("\r\n");

drawSdJpeg(0,0,0,3);//加载天气主页面

tft\_showstring(30,145,2,results\_0\_now\_data);//显示今日日期

drawSdJpeg(weather\_num[0],25,25,1);//加载今天天气图片

drawSdJpeg(weather\_num[1],45,188,4);//加载明天天气图片

drawSdJpeg(weather\_num[2],45,245,4);//加载后天天气图片

tft\_showstring(180,34,2,low\_tem\_0+"-"+high\_tem\_0);//显示今日温度字符

tft\_showstring(180,63,2,wind\_speed0);//显示今日风速字符

tft\_showstring(180,90,2,hum);//显示今日湿度字符

tft\_showstring(190,119,2,rain);//显示今日降雨量字符

tft\_showstring(180,189,2,low\_tem\_1+"-"+high\_tem\_1);//显示明日温度字符

tft\_showstring(180,210,2,wind\_speed1);//显示明日风速字符

tft\_showstring(180,248,2,low\_tem\_2+"-"+high\_tem\_2);//显示后天温度字符

tft\_showstring(180,270,2,wind\_speed2);//显示后天风速字符