ESPaper Setup Guide

Thank you for buying the Squix ESPaper Display module. I hope you will have a lot of fun with it. I'm looking forward to see your applications and extensions for it. This guide will help you to set it up and do your first steps programming it. In case of problems please head over to https://support.squix.org where you can post questions and I will try to answer them as quick as possible.

Hardware Setup

Installing the Driver

In order to save as much energy during regular usage we decided to separate the USB-to-Serial converter from the display module. This means that we need an external device to re-program the display. I am assuming here that you own the programming module included in the ESPaper Plus Kit. You can use any other USB-to-Serial converter which supports 3V3 logic levels but I cannot give you support for those here. The converter included in the kit uses a chip called CP2102 from SiLabs. Please download the driver for your operating system from this site: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Software Setup

Install Arduino IDE

As in many of my other projects will be programming the ESPaper from the Arduino IDE. Go to the Arduino page and download the latest version for your operating system. There are different installers available, pick the one that suits your needs the most and install the application.

Install the ESP8266 platform

The Arduino IDE by default only supports Arduino hardware. The ESPaper module has a ESP8266 chip at its hard so we need to install support for this device. Go to the Arduino > Preferences and copy this URL in the text field labelled "Additional Boards Manager URLs":
`http://arduino.esp8266.com/stable/package_esp8266com_index.json`

After that head over to Tools > Board > Board Manager…

And install the ESP8266 board support:

Board Manager

Now the Arduino IDE supports the ESP8266.

Connecting the programmer

It's now time to connect the programmer with your ESPaper board and see, if we can talk to the board:

Programmer attached to ESPaper

The programmer which came with the ESPaper Plus Kit provides enough power to drive the module while you are programming it. Please note: it is more efficient to use the USB port to charge the battery than to use the programmer for this purpose.

Serial Connection Settings
In the Arduino IDE go to Tools and make sure that your settings look like this:

```
Board: "Generic ESP8266 Module"
Flash Mode: "DIO"
Flash Frequency: "40MHz"
CPU Frequency: "80 MHz"
Flash Size: "2M (1M SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
Reset Method: "ck"
Upload Speed: "115200"
Port: "/dev/cu.usbserial-A50285BI"
Get Board Info
```

Arduino IDE programming settings

Now open the serial console in the Arduino IDE and see if there is output when you turn the ESPaper module on (e.g. by using the ON/OFF switch). This can look like garbage, but if you see characters your computer is capable of talking to the device.

Bringing ESP8266 into programming mode

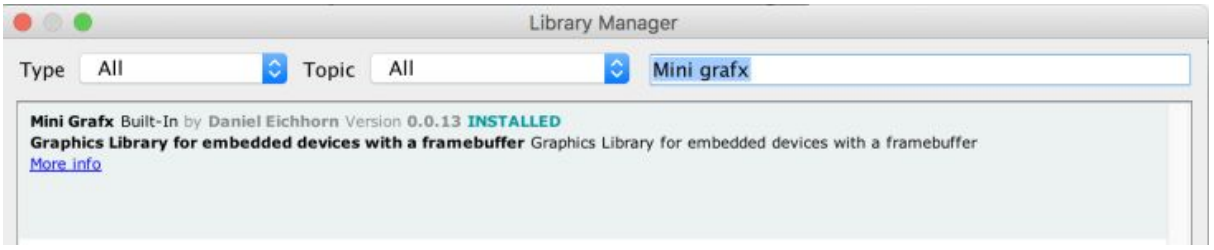
When designing the ESP8266 we gave preference of energy efficiency over convenience for programmers. This means that the ESPaper module doesn't have this nice circuit which puts the ESP8266 into programming mode when you click the upload button in the Arduino IDE.

But in order to program the chip we have to put it into this mode:

Programming mode: 1) press button S0. 2) Hold it down while pressing RESET 3) Release RESET

Installing libraries

To give you as much creative freedom as possible I've created a library to drive the display. It is called MiniGrafx and it is available on Github. You can also install it through the Arduino IDE library manager. Go to Sketch > Include Library > Manage Libraries… and enter "Mini Grafx" in the search field:



The version might be different than on this image. Just make sure that you have the latest version installed (with the highest version number). Since we're going to run a WeatherStation example we're also going to install my WeatherStation library:

For parsing potentially huge data files we need the following library:

You'll also need the simpleDSTadjust library to do proper daylight saving adjustments:

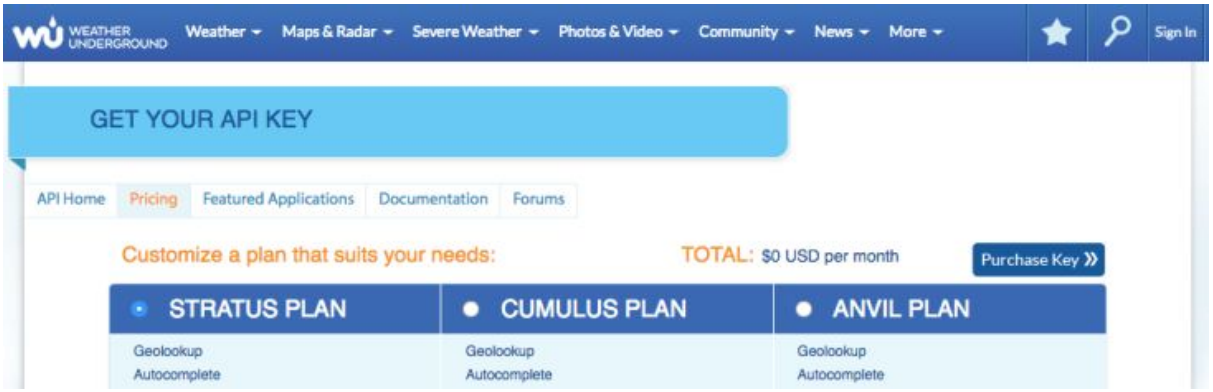
Installing the demo project

Now go to https://github.com/squix78/espaper-weatherstation and download the project by clicking on the green "Clone or Download" button. If you are familiar with git you can of

course also clone the project. Extract the zip file and open the espaper-weatherstation.ino file.

Getting a Wunderground API Key

This key is required to use the service which provides the weather information. To get the Wunderground API key go to https://www.wunderground.com/weather/api/d/pricing.html and pick the “Stratus Developer” plan. Then get your API key from this page:



Configuring the weather station code

Before we can upload the weather station to the ESPaper module we have to do a few adjustments.
Let’s start with the WiFi Settings. Replace yourssid with the name of your WiFi network and yourpassw0rd with its password. I had problems with a network that contained a dash (“-“) in the SSID. If you are having problems consider this hint… There are also nice tools like the WifiManager which avoid placing the WiFi credentials into the firmware. But they also take more time to connect to WiFi, which in turns reduces your battery life time…
Now the Wunderground settings. Set the wunderground key you got in the previous step. Also place your city. The WUNDERGROUND_COUNTRY field can be a bit tricky. In the US you usually put the state. Outside of the US you usually put the country code, for Switzerland this is “CH”. The WeatherStation library will take these values and put them into a URL. Consider the following two:

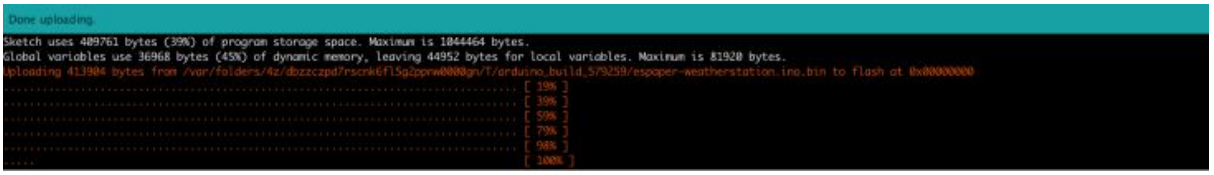
- http://api.wunderground.com/api/808ba87ed77c4511/conditions/q/CH/Zurich.json
- http://api.wunderground.com/api/808ba87ed77c4511/conditions/q/CA/SAN_FRANCISCO.json

In the last segment after the slash you can see Zurich.json and SAN_FRANCISCO.json. This is your city name. Also note that SAN_FRANCISCO has a underscore instead of a space. Zurich doesn’t have the umlaut “ü” which it would have in German. Then the segement before that: CH and CA. So for Zurich it’s the country sign, for San Francisco the

US state. You can test your settings. Take this URL and replace the values in <..> with your values:
http://api.wunderground.com/api/<WUNDERGROUND_KEY>/conditions/q/<STATE_OR_COUNTRY>/<CITY>.json
If your browser shows a nice JSON object you know that it should work in your code as well. Now let's move on to the time zone. Change the UTC_OFFSET field according to your time zone. The number after UTC_OFFSET is the number of ours your "in front" or "behind" of the UTC time zone. Next enter settings according to your timezone and when daylight saving time starts and ends. If you don't know how to do that, have a look at this page.
And as a last point you might have a look at the IS_METRIC and IS_STYLE_12HR which influence if temperature is showed in degree celsius (°C) or fahrenheit (°C).

Uploading the Code

As mentioned in an earlier paragraph connect the ESPaper to your computer and put the module into programming mode by pressing S0 (middle button), Reset (right button) while keeping the first pressed, releasing Reset and releasing S0. Then press the upload button from the menu. If everything went well you should see something like this:



From time to time this might not work and you will see a message similar to this:

In this case make sure that there is a port selected in the tools menu. Disconnect the module from the programmer and reconnect it. Make sure that they are connected correctly. Then run again through the S0, S0+Reset, S0 button sequence and try to upload the code again.

Help

In case you got stuck somewhere during this guide please go to https://support.squix.org and check if somebody else already posted a similar problem. If you can't find a similar problem please post your question there. I will try to help you as soon as possible