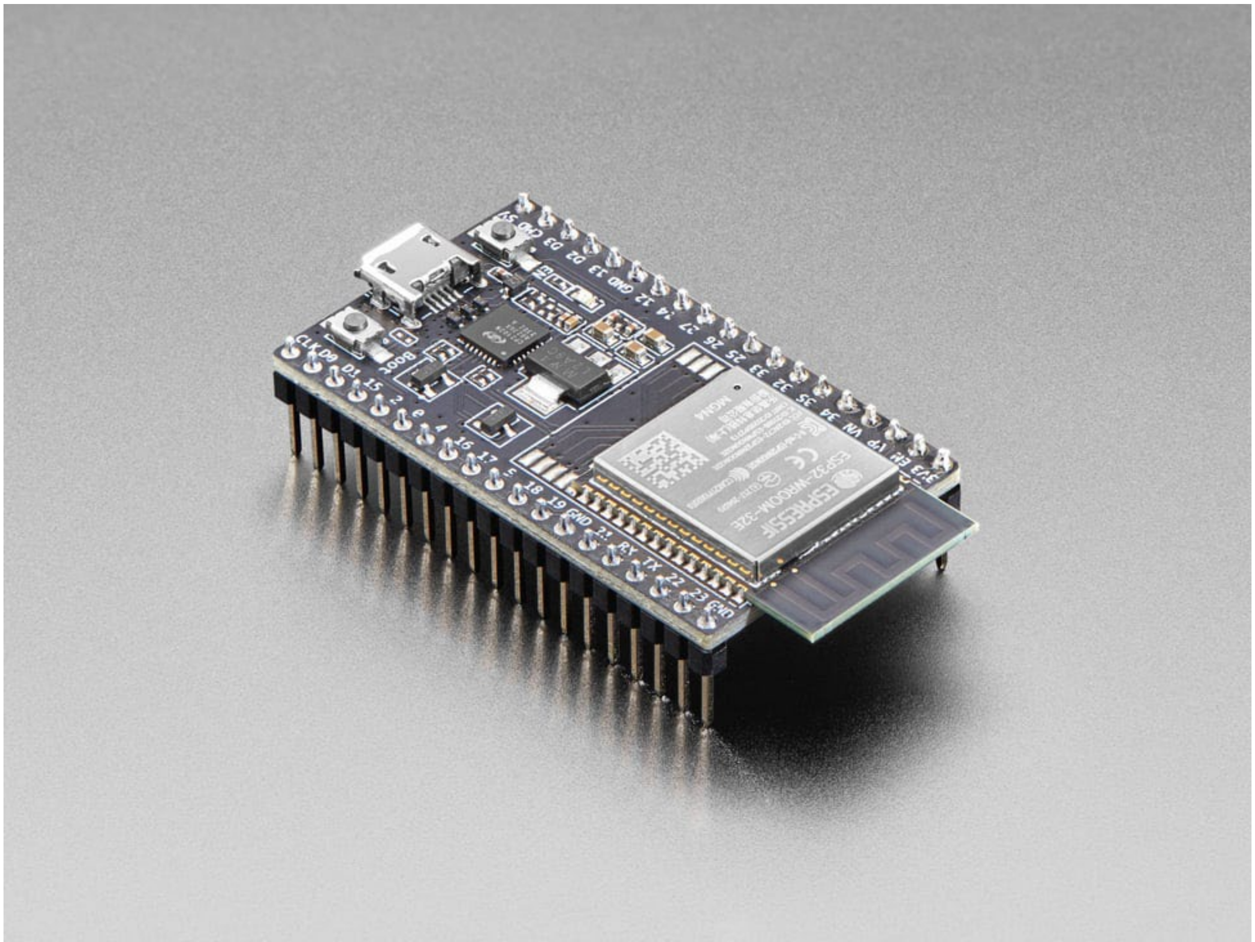
OpenWeatherOneCall

ESP32 Library

V4.0.0



The only better way to get the weather is by sticking your head out the window

Created by JESSICA HERSHEY

for

Arduino Library Manager and Platform.IO

www.github.com/jhershey69

Thank you for installing OpenWeatherOneCall v4. This upgrade has many changes from the v3.3.3 last released and we'll cover them all. The most major change is the DEPRECATION of the LEGACY CALLING function that held on to the old API call from V2***. We are no longer supporting anything before v3 now.

Also, OpenWeatherMap, who has proprietary model ownership, has made a few changes to the information they send and how it is secured. Their basic account for API 3.0 is a *Pay as You Call* type of subscription BUT you can still get 1000 calls a day for FREE. That is 1 API call every 90 seconds. Since OpenWeatherMap only updates the API every 10 minutes there is really no need to call the API more than 144 times a day.

Also, some of the information they return has changed. They no longer return **HOURLY** information with **HISTORICAL** API calls for the FREE accounts, but they do have history now from January 1, 1979, to today. No more 5-day look back restrictions. Future forecasts are still restricted to 8 days in the CURRENT weather call, and there is a single day TIMESTAMP call that allows a look ahead of 4 individual days.

Changes not associated with OpenWeatherMap include user selectable date and time formatting, a change to how historical weather is called by using human readable dates instead of number of days which was restricted to 5 days as mentioned earlier.

So, let's get into the nitty gritty of how to install, how to set up flags, and how to get the weather.

AND NOW THE PART
YOU'VE ALL BEEN WAITING FOR!

INSTALLATION

Install as you would any other library using the library manager or downloading the zip file from GitHub.

Dependencies are built into the Library Managers. If you are installing manually, you will also need to install ArduinoJSON v7 or above. ArduinoJSON versions before 7 are no longer supported in this library. However, they tell me this should still work with v6 but might throw errors that are insignificant. Why chance it?

Once you have it installed all you need do it place it in your sketch as any other library:

```
#include “openweatheronecall.h”
```

Nothing else is a required #include for your sketch for this.

OVERVIEW

OpenWeatherOneCall v4.0.0 returns the following data sets of weather information:

CURRENT weather which includes:

- **Current weather conditions**
- **Daily (8 day future forecast)**
- **Hourly (48 hour future forecast)**
- **Minutely (60 minutes of precipitation forecast)**
- **Alerts (Local NWS alerts)**

TIMESTAMP (previously known as HISTORICAL):

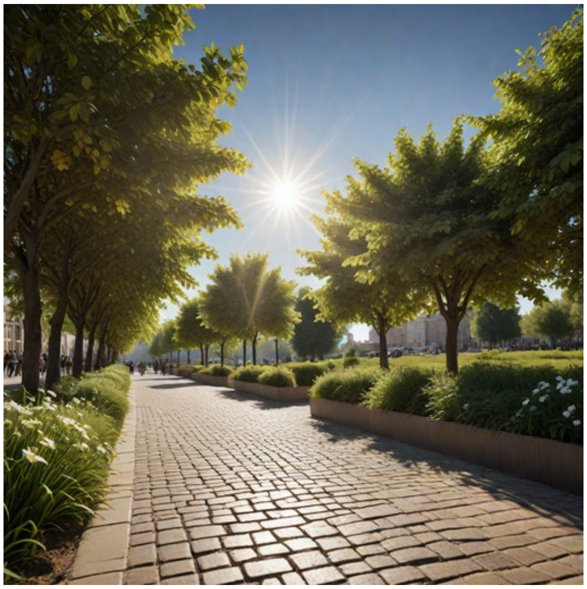
- **Any single date from 01/01/1979 to 4 days in future**

OVERVIEW (Weather Summary) for:

- **Today and tomorrow**

AIR QUALITY includes:

- **Air Quality Index**
- **CO, NO, NO_2, O_3, SO_2, NH_3, $PM_{2.5}$, PM_{10}**



SKETCH SETUP

I’ll be using the SimpleWeather example sketch to explain the basics for setting up your API calls.

You’ll need to have an internet connection of course. It matters not how you do that. If you use WIFI then you should naturally have:

```
#define HOMESSID "????????????????"
#define HOMEPW "????????????????"
```

OpenWeatherOneCall.h has an include for WiFi.h which is the usual source of WIFI information and connection in the ESP32. If you have issues or use something else, you can comment it out.

Next, you will need to indicate your OpenWeatherOneCall API ID Key from OpenWeatherMap.

```
char ONECALLKEY[] = "????????????????????????????????";
```

This is the 3.0 API key for One Call by Call subscription. This subscription gives you 1000 calls a day for free. Other keys for other subscriptions for other APIs will not work and will throw back an error.

Next, we’ll cover LOCATION MODES, what they are, and how to set them up.

LOCATION MODES

Location Modes inform the program how you will be providing **Latitude** and **Longitude**. It makes no difference where you gather this information. You can put them in manually, extract them from a GPS, use a CITY ID from OpenWeatherMap, or triangulate with nearby Wifi signals. The program is only interested in the Lat and Lon numbers and how you are getting them.

Set flag locationMode in your setup as such:

```
locationMode = 1;
```

1 for Lat/Lon, 2 for City ID, 3 for IP Address

For **manually** setting your desired location you must also do this:

```
float myLATITUDE = 39.9537;  //<-----This is Toms River, NJ
float myLONGITUDE = -74.1979;
```

For **City ID** Location MODE setting if used

```
int myCITYID = 4504476;  //<-----Toms River, NJ USA
```

For **IP Address** MODE you must use a program such as WiFiTri available from the same fine people as you got this library.

Once again, you can get the numbers from your GPS just plug them into the above **myLATITUDE** and **myLONGITUDE** variables.

Next up, about your options for weather…

WEATHER OPTIONS

Looking at the example script you will see this:

```
int myCURRENT = 1;       //<-----0 is CURRENT off, 1 is on

int myAIRQUALITY = 1;    //<-----0 is Air Quality off, 1 is on
int myTIMESTAMP = 1;     //<-----0 is TIMESTAMP off, 1 is TIMESTAMP on
char timestampDate[11];  //<-----MM/DD/YYYY right here limit is 4 days in future,
past is 01/01/1979
int myOVERVIEW = 1;      //<-----0 is OVERVIEW off, 1 is OVERVIEW on
char overviewDate[9] = "TOMORROW";   //<-----"TODAY" or "TOMORROW"
```

These are the **FLAGS** that turn on/off the available API calls as described in the OVERVIEW section of this manual. These can be called in any combination or none at all. Older versions of this library restricted the call to a single choice, no more does that matter!

Let’s have a look at each option in order of appearance:

CURRENT WEATHER

Setting myCURRENT to 1 (ON) returns the options as outlined in OVERVIEW. You can select which portions of CURRENT WEATHER you desire, and we will cover how to do that in EXCLUDES. These are again:

- **Daily (8 day future forecast)**
- **Hourly (48 hour future forecast)**
- **Minutely (60 minutes of precipitation forecast)**
- **Alerts (Local NWS alerts)**

AIR QUALITY

Setting myAIRQUALITY to 1 (ON) returns the following information about your local current air quality:

Carbon monoxide (CO), Nitrogen monoxide (NO), Nitrogen dioxide (NO_2), Ozone (O_3), Sulphur dioxide (SO_2), Ammonia (NH_3), and particulates ($PM_{2.5}$ and PM_{10}).

More than you would ever want to know about Air Quality, pollutants, and how they are scaled are available on the OpenWeatherMap Air Quality page here.

The JSON response for Air Quality is:

```
{
 "coord":[
    50,
    50
  ],
  "list":[
    {
      "dt":1605182400,
      "main":{
        "aqi":1
      },
      "components":{
        "co":201.94053649902344,
        "no":0.01877197064459324,
        "no2":0.7711350917816162,
        "o3":68.66455078125,
        "so2":0.6407499313354492,
        "pm2_5":0.5,
        "pm10":0.540438711643219,
        "nh3":0.12369127571582794
      }
    }
  ]
}
```

TIMESTAMP (previously known as HISTORICAL):

Setting myTIMESTAMP to 1 (ON) returns any single date from 01/01/1979 to 4 days in future. Using timestampDate[11]=”MM/DD/YYYY”, set the date you wish to retrieve.

The JSON response is:

```
{
"lat": 52.2297,
"lon": 21.0122,
"timezone": "Europe/Warsaw",
"timezone_offset": 3600,
"data": [
 {
   "dt": 1645888976,
   "sunrise": 1645853361,
   "sunset": 1645891727,
   "temp": 279.13,
   "feels_like": 276.44,
   "pressure": 1029,
   "humidity": 64,
   "dew_point": 272.88,
   "uvi": 0.06,
   "clouds": 0,
   "visibility": 10000,
   "wind_speed": 3.6,
   "wind_deg": 340,
   "weather": [
     {
       "id": 800,
       "main": "Clear",
          "description": "clear sky",
          "icon": "01d"
        }
      ]
    }
  ]
}
```

OVERVIEW

OVERVIEW is known as SUMMARY on OpenWeatherMap and provides a “summary” of weather conditions for **today** and **tomorrow** ONLY. This summary is AI generated.

You turn it off and on with myOVERVIEW = 1 in the script.

You indicate which day you want with:
overviewDate[9] = "TODAY";   //<-----"TODAY" or "TOMORROW"

The JSON response is:

```
{
  "lat": 51.509865,
   "lon": -0.118092,
   "tz": "+01:00",
   "date": "2024-05-13",
   "units": "metric",
   "weather_overview": "The current weather is overcast with a
temperature of 16°C and a feels-like temperature of 16°C.
The wind speed is 4 meter/sec with gusts up to 6 meter/sec
coming from the west-southwest direction.
The air pressure is at 1007 hPa with a humidity level of 79%.
The dew point is at 12°C and the visibility is 10000 meters.
The UV index is at 4, indicating moderate risk from the
sun's UV rays.
The sky is covered with overcast clouds, and there is
no precipitation expected at the moment.
Overall, it is a moderately cool and cloudy day
with light to moderate winds from the west-southwest."
}
```

The response doesn’t return the overview in a neatly word wrapped format like here. You’ll have to do that yourself.

EXCLUDES

Moving on, the CURRENT weather option has 4 subsections:

- **Daily (8 day future forecast)**
- **Hourly (48 hour future forecast)**
- **Minutely (60 minutes of precipitation forecast)**
- **Alerts (Local NWS alerts)**

You can retrieve any or all of these in a single call. To tell the API which to return or leave out we use EXCLUDES. From the example script:

```
// Excludes are EXCL_D, EXCL_H, EXCL_M, EXCL_A

// Those are DAILY, HOURLY, MINUTELY, and ALERTS
// You set them like this: myEXCLUDES = EXCL_D+EXCL_H+EXCL_M+EXCL_A
// If you leave them set to 0 as below you get a JSON file with ALL CURRENT
WEATHER measurements which is a huge file

int myEXCLUDES = 0;  //<-----0 Excludes is default
```

Consult the variables available page for information sent in the JSON.

The only other things to set up are the UNITS of measurement you wish to use, your Time Zone offset, and how you want your DATE/TIME formatted.

```
int myUNITS = IMPERIAL;  //<-----METRIC, IMPERIAL, KELVIN (IMPERIAL is default)

int myTimeZone = -4;     //<--------- GMT OFFSET only used in OVERVIEW

int myDTF = 3;      // 1 M/D/Y 24H, 2 D/M/Y 24H, 3 M/D/Y 12H, 4 D/M/Y 12H
                    // 5/6 TIME ONL10 M/D/Y ONLY, 11 D/M/Y ONLY
                    // ISO8601 options: 12 YYYY-MM-DD ONLY, 13 THH:MM:SS ONLY
                    // 14 YYYY-MM-DDTHH:MM:SSY 24H
```

And that’s it for setting up the script. Easy as pie. Unless it is Strawberry Rhubarb Pie then that is a bit more complicated.

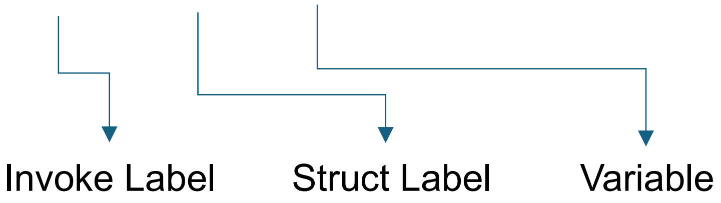
PUTTING IT ALL TOGETHER

Once you have your FLAGS and DATES set above you will need to INVOKE the library and assign it a reference label. You do it this way:

```
OpenWeatherOneCall OWOC;
```

We have used OWOC in the script but you can use whatever you like that is easier for you if OWOC doesn't fit your use. This is what you will use to call the variables from the structs created by the deserialized JSON. For example:

```
OWOC.location.CITY, OWOC.location.STATE, OWOC.location.COUNTRY
```



Invoke Label Struct Label Variable

Once you have invoked the library all you need to do to run it is:

```
OWOC.parseWeather();
```

That's it! No more passing clunky variables, the program takes care of everything from this point on.

All that is left to do is write your program to retrieve and display the variables you desire. You can display these as you like, with TFT, SERIAL, however.

Consult the VARIABLES PDF for a complete list of variables.

SAMPLE SCRIPT

We’ve covered a lot of ground here and followed along with the sample script for most of the way. If you load and run the sample script using only the WIFI info changes you will get an output to your serial monitor with information on the weather in my hometown of Toms River, NJ.

On your serial monitor the output will be like this from 08/08/2024:

```
Location: Toms River, US-NJ US

CURRENT
Temp :  74
Humidity :  87

FORECAST - Up to 8 days future forecast
Date: 08/09/2024 High Temperature:  75
Date: 08/10/2024 High Temperature:  85
Date: 08/11/2024 High Temperature:  88
Date: 08/12/2024 High Temperature:  84
Date: 08/13/2024 High Temperature:  82
Date: 08/14/2024 High Temperature:  84
Date: 08/15/2024 High Temperature:  85
Date: 08/16/2024 High Temperature:  89

HOURLY   - Up to 48 hours forecast
16:00 Actual Temp:  74
16:00 Feels Like Temp:  75
17:00 Actual Temp:  74
17:00 Feels Like Temp:  75
18:00 Actual Temp:  74
18:00 Feels Like Temp:  76
19:00 Actual Temp:  74
19:00 Feels Like Temp:  76
20:00 Actual Temp:  74
20:00 Actual Temp:  74
20:00 Feels Like Temp:  75
21:00 Actual Temp:  74
21:00 Feels Like Temp:  75

MINUTELY   - Up to 60 minutes precipitation forecast
16:36 Precipitation:  0.00
16:37 Precipitation:  0.00
16:38 Precipitation:  0.00
16:39 Precipitation:  0.00
16:40 Precipitation:  0.00
16:41 Precipitation:  0.00
16:42 Precipitation:  0.00
```

```
16:43 Precipitation:  0.00
16:44 Precipitation:  0.00
16:45 Precipitation:  0.00
16:46 Precipitation:  0.00
16:47 Precipitation:  0.00
16:48 Precipitation:  0.00
16:49 Precipitation:  0.00
16:50 Precipitation:  0.00
16:51 Precipitation:  0.00
16:52 Precipitation:  0.00
16:53 Precipitation:  0.00
16:54 Precipitation:  0.00
16:55 Precipitation:  0.00
16:56 Precipitation:  0.00
16:57 Precipitation:  0.00
16:58 Precipitation:  0.00
16:59 Precipitation:  0.00
17:00 Precipitation:  0.00
17:01 Precipitation:  0.00
17:02 Precipitation:  0.00
17:03 Precipitation:  0.00
17:04 Precipitation:  0.00
17:05 Precipitation:  0.00

TIMESTAMP - Single day weather for any date from 01-01-1979 to 4 days in
future
Fri - 03/03/2023
Temperature:  49
Feels like:  49
Humidity:  66%

OVERVIEW - Summary for TODAY or TOMORROW generated by AI

Longitude: -74.197899
Timezone: -04:00
Date: 2024-08-09
Units: imperial
Weather Overview: Tomorrow's weather will bring partly cloudy skies with
rain, and temperatures ranging from a low of 74°F to a high of 85°F. The day
will feel warmer, with a heat index of 91°F. The wind will be blowing at a
speed of 21 miles per hour, with gusts up to 36 miles per hour, coming from
the south. The humidity will be at 67%, and there is a 100% chance of rain,
with a moderate rain expected. The UV index will be at a high of 9, so it's
important to take precautions against sun exposure. Overall, it will be a day
of mixed weather conditions, with a high chance of rain and windy conditions.
Stay prepared for the changing weather throughout the day.

Air Quality and Pollution Numbers for today
Air Quality :  1
Carbon :  230.31
Nitrogen :  0.16
Ozone :  55.08

ALERT *** ALERT *** ALERT
Sender : NWS Mount Holly NJ
Event : Coastal Flood Advisory
ALERT : * WHAT...Up to one foot of inundation above ground level expected
```

```
in low- lying areas near back bays and tidal waterways.

* WHERE...Ocean.

* WHEN...From 6 AM Friday to 6 AM EDT Saturday.

* IMPACTS...Flooding of lots, parks, and roads with only
isolated road closures expected.

* ADDITIONAL DETAILS...The main impacts from minor coastal
flooding are expected along Barnegat Bay, especially near
Mantoloking, where persistent southerly winds will trap high
water from draining at high tide.
```

As you can see there is a massive amount of information you can obtain with this library from Open Weather Map. How you use it is up to you. You can design amazing graphics and interfaces that give your users information on any weather they may want to know.

Imagine designing a complete weather station!

As the creator of this library, I hope you do great things with it. In fact, I'd love to see the things you do. Hit me up with some graphics and projects at my GitHub repository page.

Thanks!

Jessica Hershey
http://www.github.com/jhershey69