# ESP32/ESP8266 Weather Forecaster

## 1 OVERVIEW

Using an ESP32 or ESP8266 to forecast weather can be achieved by reading air pressure from a sensor such as the BOSCH BMP085/180 or BME280.

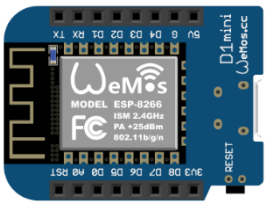
By monitoring the changes in air pressure and categorising the changes thus:

- Rising quickly
- Rising
- Rising slowly
- Steady
- Falling slowly
- Falling
- Falling quickly

Enables the weather to be forecast using a set of rules based on air pressure at your location (adjusted from sea level with a offset) and then by monitoring changes over time, in this case 3 and 1 hours leading to a prediction with a good level of certainty.

## 2 HARDWARE

The project uses an ESP32 or ESP8266 processor and the I2C bus to communicate with an OLED display of 1.3” or 0.96” size together with a BOSCH BME280 or BMP180 air pressure sensor. The choices are, but not limited to:

| CPU | | Display | | Pressure Sensor |
|---|---|---|---|---|
|  | |  | |  |
| ESP32 | | OLED 1.3” SH1106 | | BMP085/BMP180 |
|  | or |  | or |  |
| ESP8266 | | OLED 0.96” SSD1306 | | BME280 |

# 3 SOFTWARE

The source needs to be adjusted depending on the hardware configuration. The BOSCH sensors require:

| Device | Driver Library Required |
|---|---|
| BMP085 or BMP180 | Adafruit_BMP085 |
| BME280 | Adafruit_BME280 |

Both libraries do-not provide support for the Bosch devices but are easily modified to enable support and compilation.

The displays require the correct driver support, the most common screen driver device is the SSD1306 for the 0.96” OLED display or SH1106 for the 1.3” OLED variant.

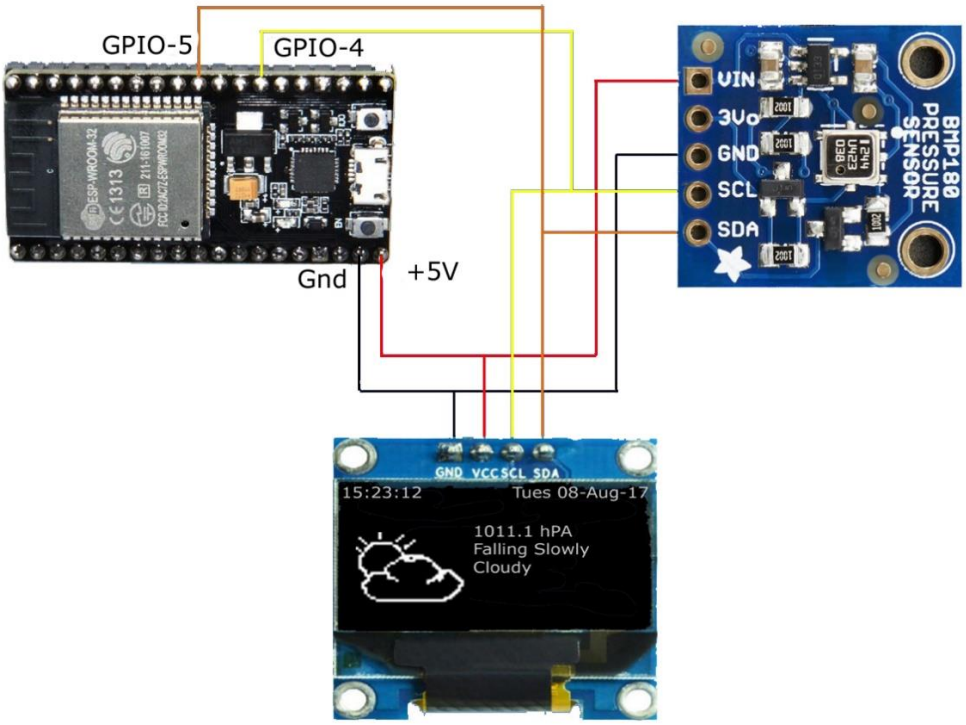
The code works equally well on an ESP8266 but the time source requires some modification, see the ESP8266 source for the differences.

# 4 SYSTEM WIRING

Connect the Esp32 or ESP8266 to the OLED and BMP/BME Sensors using the I2C bus connections. For example if you decide to use GPIO-5 for the I2C SDA function and GPIO-4 for the I2C SCL function then the source code statement that initiates the I2C bus support would be:

Wire.begin(SDA,SCL) or Wire.begin(5,4);

Generally you can use any two pins to provide the I2C bus function, but be aware that other pins are used for other functions two, so be careful not to create a conflict.



# 5 ADAFRUIT LIBRARY CHANGES REQUIRED

The Adafruit sensor BME280 has an I2C address that has been set at **0x77** and if you are using an Adafruit device then no modification of the device library address is required.

However, if you have one of the many third-party devices these nearly always use the address **0x76** and require the following changes to the library file:

BME280 Modify the address in the library file: **Adafruit_BME280.h** to read thus:

```
#define BME280_ADDRESS                (0x76)
```

Generally the BMP085 third-party devices use the address 0x77 as does the Adafruit variety.

In addition to device address changes the files Adafruit_BMP085.cpp and Adafruit_BME280.cpp need to be modified to remove a default Wire.begin() statement so that the ESP32 can have the required pins defined.

For the Adafruit_BMP085 library, edit the file Adafruit_BMP085.CPP and comment out the wire.begin statement:

```
boolean Adafruit_BMP085::begin(uint8_t mode) {
if (mode > BMP085_ULTRAHIGHRES) mode = BMP085_ULTRAHIGHRES;
oversampling = mode;
//Wire.begin(); //****************** Comment out
```

For the Adafruit_BME280 library, edit the file Adafruit_BME280.CPP and comment out the wire.begin statement:

```
bool Adafruit_BME280::begin(uint8_t addr) {
 _i2caddr = addr;  // init I2C or SPI sensor interface
if (_cs == -1) { // I2C
 //Wire.begin(); //****************** Comment out
```

# 6 DAYLIGHT SAVING AND TIME ZONE ADJUSTMENTS

The source code time statement is currently provided for UK time and Daylight Saving Time is enabled, this requires a time configuration statement thus:

configTime(1**3600, 3600, "pool.ntp.org"); // +1hour (1*60*60=3600=+1hour) ahead for DST in the UK

The format of the ‘configtime’ statement is thus:

configTime(timezone, daylightoffset, primary_timeserver_address, secondary_timeserver_address);

therefore examples are:

configTime(1**3600, 3600, "pool.ntp.org"); // UK and DST is active*

configTime(*0, 0, "pool.ntp.org");  // UK and DST is not active*