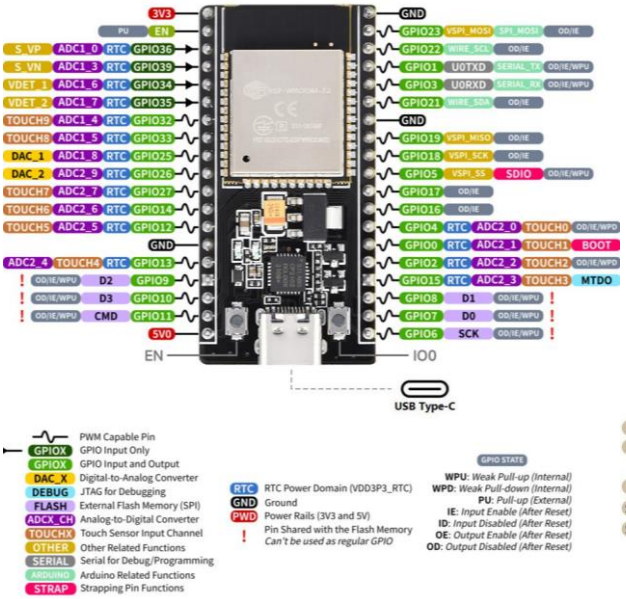
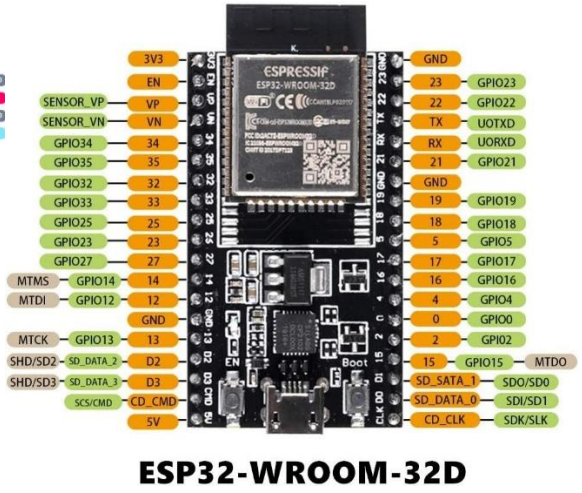


(usb c variant)



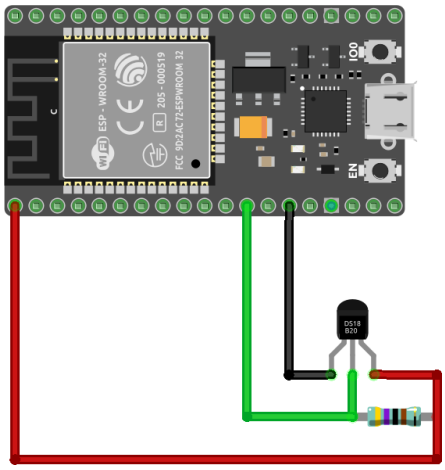
(micro USB variant)

Prepare the Arduino IDE for use with the ESP32 microcontroller board

1. Open the Arduino Preferences (Arduino → Preferences)
2. Look for "Additional Boards Manager URLs" input box and paste https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json in it.
3. Click OK.
4. Then under Tools → Boards: , click on Boards Manager.
5. Install the ESP32 board by Espressif Systems.
6. Use board NodeMCU-32s

DS18B20 Temperature sensor

We're going to use the Dallas DS18b20 sensor to measure temperature.
You'll need the "dallastemperature" and "onewire" libraries.
The DS18B20 sensor needs a pullup resistor on it's data pin.



Write a sketch that reads the temperature from the sensor and prints the temperature in the console. Lots of simple examples can be found online.

Add the InfluxDB library from the library manager

1. Under Sketch → Include Libraries, click on Manage Libraries.
2. Search for 'influxdb' in the search box.
3. Install "ESP8266 Influxdb by Tobias Schürg" library.
4. Copy the following code to the Arduino IDE

```
#if defined(ESP32)
#include <WiFiMulti.h>
WiFiMulti wifiMulti;
#define DEVICE "ESP32"
#endif

#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>

// WiFi AP SSID
#define WIFI_SSID "YOUR_WIFI_SSID"
// WiFi password
#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"

#define INFLUXDB_URL "https://sensin.xablu.com"
#define INFLUXDB_TOKEN "YOUR_TOKEN"
#define INFLUXDB_ORG "YOUR_ORGANISATION_ID"
#define INFLUXDB_BUCKET "YOUR_BUCKET"

// Time zone info
#define TZ_INFO "UTC2"

// Declare InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN,
InfluxDbCloud2CACert);
```

```
// Declare Data point
Point sensor("wifi_status");

void setup() {
    Serial.begin(115200);

    // Setup wifi
    WiFi.mode(WIFI_STA);
    wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

    Serial.print("Connecting to wifi");
    while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(100);
}
Serial.println();

// Accurate time is necessary for certificate validation and writing in batches
// We use the NTP servers in your area as provided by: https://www.pool.ntp.org/zone/
// Syncing progress and the time will be printed to Serial.
timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");


// Check server connection
if (client.validateConnection()) {
      Serial.print("Connected to InfluxDB: ");
      Serial.println(client.getServerUrl());
  } else {
      Serial.print("InfluxDB connection failed: ");
      Serial.println(client.getLastErrorMessage());
  }
}
void loop() {}
```

Set Wifi and InfluxDB credentials and test your connection

For Wifi you can use the provided AccessPoint credentials. Fill in the WiFi SSID and password in the code.
SSID: Xablu IoT
Password: WelkomBijXablu!

For InfluxDB you need an API key and an organisation ID, login to the https://sensin.xablu.com webportal and create a new API key via the “API tokens” options in the left menu bar.
You can find your organization in the “ABOUT” section of your profile.
Copy the organization ID to your esp32 code at INFLUXDB_ORG
Copy the generated API token to your esp32 code at INFLUXDB_TOKEN

Change the bucket name in the code to the bucket name in your InfluxDB account. You can find your bucket name via “Buckets” in the left menu bar.

Write data

Assign a name to your datapoint in `Point sensor(String)` method that needs a string

Before you can start writing data to the database, you have to assign tags to the sensor point to structure your data.
(eg: `sensor.addTag("Xablu", "Distance Sensor"))`

Now we have to extend the loop function, so it will send actual dataFields to the database. The goals is to write the temperature measured by the DS18b20 sensor to the database. Include the following elements in your loop function to make it work properly:

```
Make sure that the esp32 is connected to the wifi
    if (wifiMulti.run() != WL_CONNECTED) {
    Serial.println("Wifi connection lost");
}

Clear fields for reusing the point. Tags will remain the same as set above.
sensor.clearFields();

Create a data field for the database (eg: int distance = 5;
sensor.addField("Measured distance", distance))

sensor.addField("variable Name", [Data]);

Get the data that is will be send to the database with the method
sensor.toLineProtocol(), and print it to the console

Write the point to the database
client.writePoint(sensor)

client.writePoint() Is a Boolean and returns true if the writing was
successful and false when the writing has failed. Check if the data is
correctly written to the database. Otherwise, print the error message to the
console
client.getLastErrorMessage()
```

Write the collected sensor data to the database every 10(!) seconds

Create a data graph

To display you data that is send to the database in a graph, go to "Data Explorer" in InfluxDB. Select "Switch to old Data Explorer". Select your bucket, tags and fields that you created and want to display in your graph. Press submit to show your data in the graph.
Change the graph type in the left top corner to see different visualization of the same data you send to the database. You can also customize your graph type to your own preferences.