

<https://www.halvorsen.blog>



Arduino UNO R4 WiFi

ThingSpeak – IoT Cloud Platform for
Datalogging and Monitoring

Hans-Petter Halvorsen

Contents

- Introduction to Arduino UNO R4 WiFi
- Arduino IDE
- WiFi Communication using Arduino UNO R4 WiFi
- ThingSpeak with Examples
 - ThingSpeak is an IoT Cloud Platform for Datalogging and Monitoring
 - Example 1 – Using the premade Arduino Library from MathWorks
 - Example 2 – Using HTTP and the available REST API



Introduction to Arduino UNO R4 WiFi

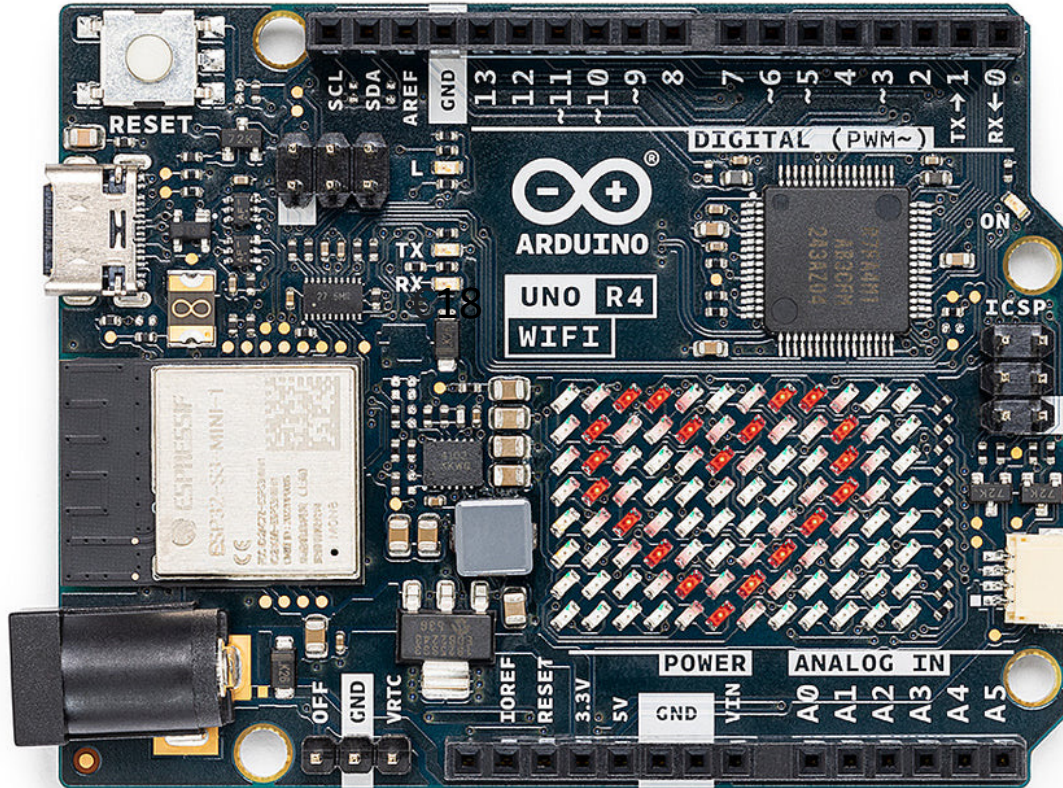
Hans-Petter Halvorsen

[Table of Contents](#)

Arduino UNO R4

- In mid 2023 a new version of the popular Arduino UNO R3 was released
- Arduino UNO R4 comes in 2 different versions:
 - Arduino UNO R4 Minima
 - Arduino UNO R4 WiFi

Arduino UNO R4 WiFi



<https://docs.arduino.cc/hardware/uno-r4-wifi>

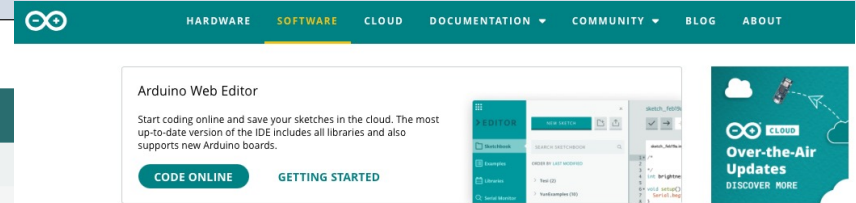
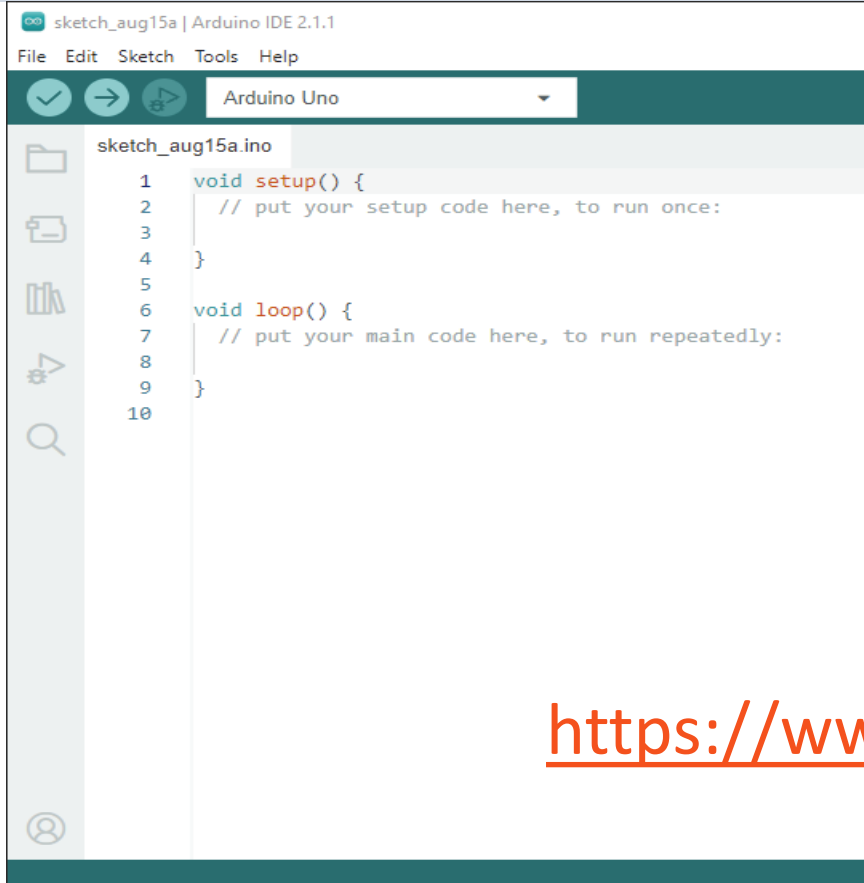


Arduino IDE

Hans-Petter Halvorsen

[Table of Contents](#)

Arduino IDE



Downloads



Arduino IDE 2.1.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

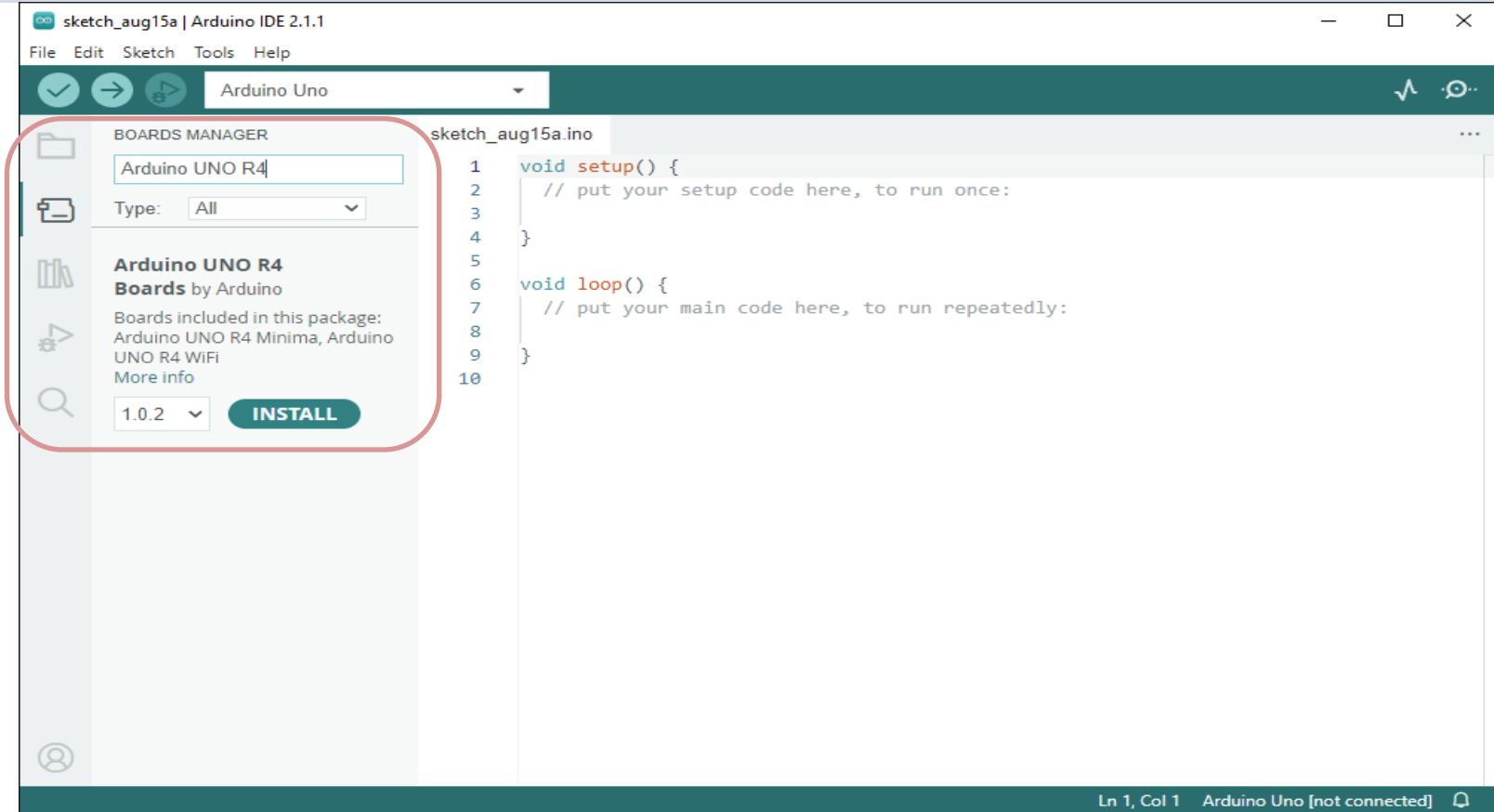
DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file
Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)
macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

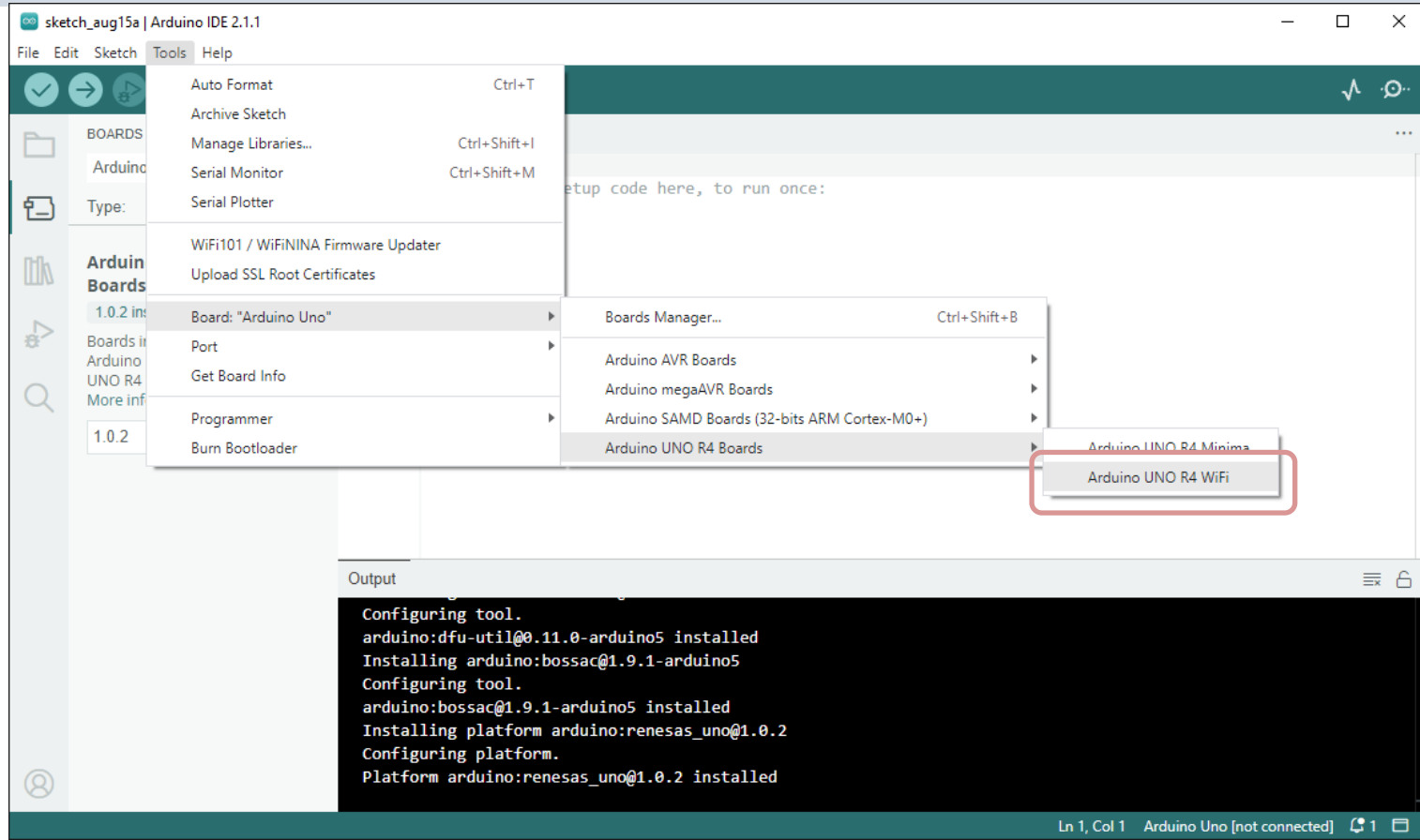
[Release Notes](#)

<https://www.arduino.cc/en/software>

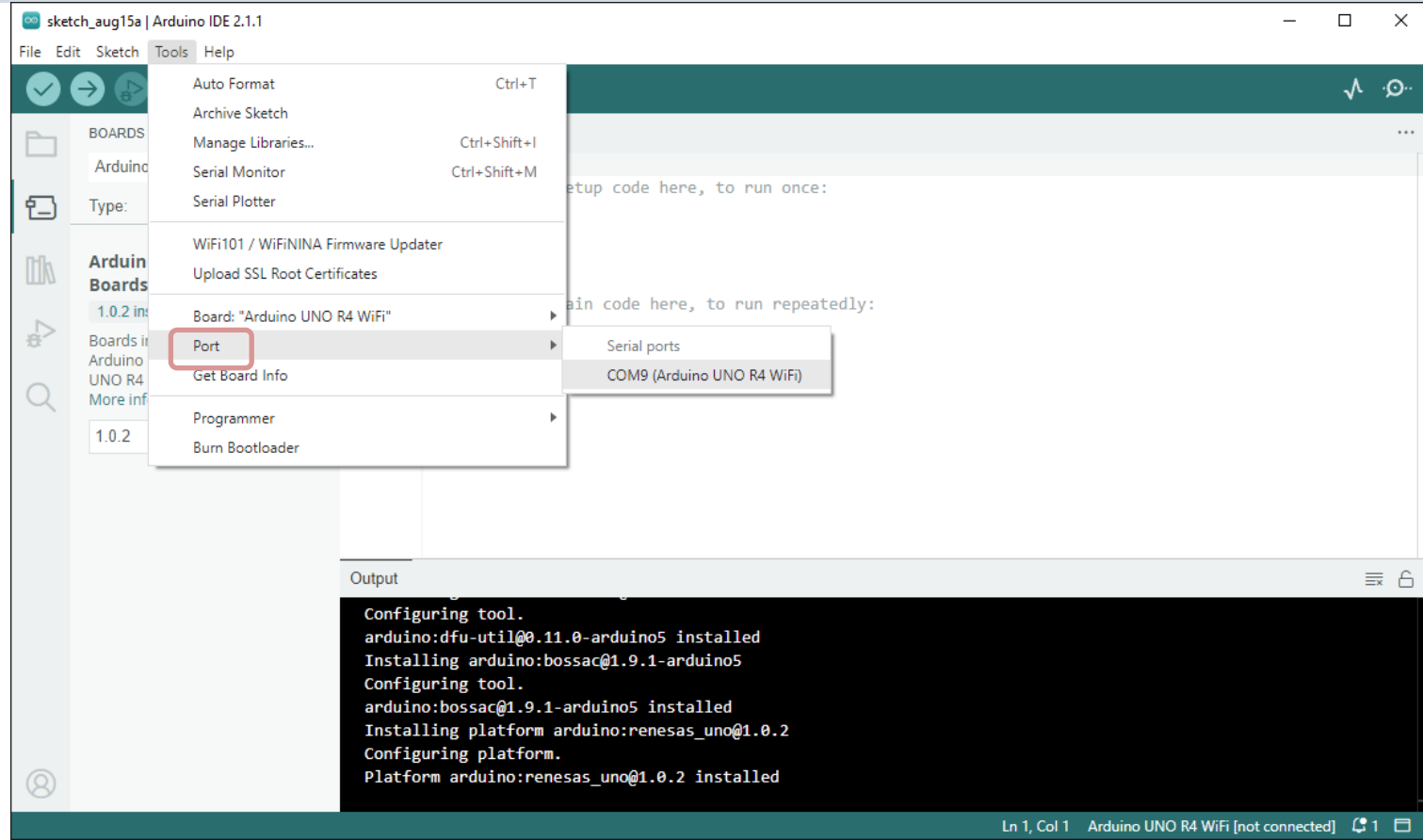
Install Arduino UNO R4 Board



Select Arduino UNO R4 Board



Select Port





WiFi

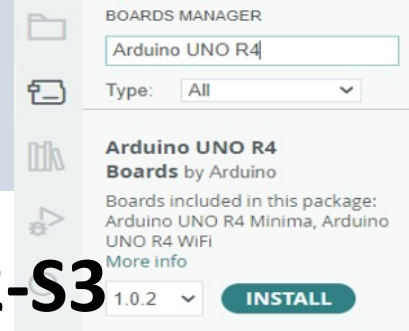
Connection Arduino UNO R4 WiFi to WiFi/Internet

Hans-Petter Halvorsen

[Table of Contents](#)

WiFi

- **Arduino UNO R4 WiFi** has a built in **ESP32-S3** module that enables you to connect to Wi-Fi networks and perform network operations.
- Wi-Fi support is enabled via the built-in **WiFiS3 library** that is shipped with the Arduino UNO R4 Core.
- Installing the Arduino UNO R4 Core automatically installs the WiFiS3 library.



```
#include <WiFiS3.h>
#include "secrets.h"

char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int status = WL_IDLE_STATUS;
```

```
void setup()
{
  Serial.begin(9600);
  ConnectWiFi();
}
```

```
void loop() {
  delay(10000);
  PrintNetwork();
}
```

Basic Arduino UNO
R4 WiFi Example
that connects to a
given WiFi Network

```
void PrintNetwork()
{
  Serial.print("WiFi Status: ");
  Serial.println(WiFi.status());

  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```

```
void ConnectWiFi()
{
  // check for the WiFi module:
  if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    while (true);
  }

  String fv = WiFi.firmwareVersion();
  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
  }

  // Attempt to connect to WiFi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, pass);

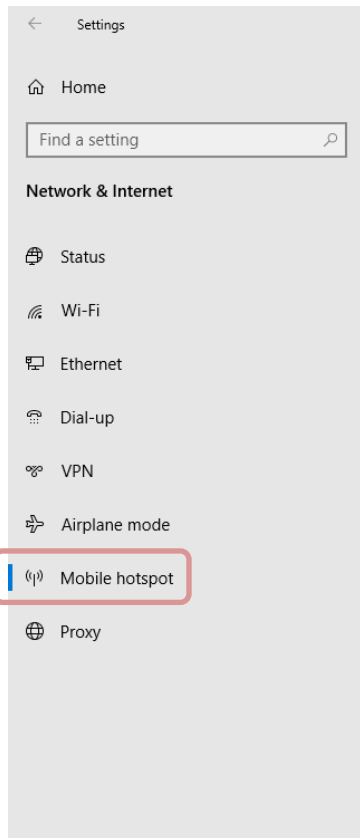
    // wait 10 seconds for connection:
    delay(10000);
  }

  // You're connected now, so print out the data:
  Serial.println("You're connected to Wifi");
  PrintNetwork();
}
```

“secrets.h”:

```
#define SECRET_SSID "xxx"
#define SECRET_PASS "xxx"
```

Setting up a Mobile Hotspot WiFi Network



Mobile hotspot

Share my Internet connection with other devices

☐ Off

Share my Internet connection from

Wi-Fi

Share my Internet connection over

☒ Wi-Fi

☐ Bluetooth

Network name:

Network password:

Network band: 2.4 GHz

Edit

Related settings

[Change adapter options](#)

[Network and Sharing Center](#)

[Windows Firewall](#)

Help from the web

[Setting up mobile hotspot](#)

[Get help](#)

You cannot connect to your Eduroam Network from Arduino.

You can easily configure a Mobile Hotspot WiFi Network in Windows 10/11 or on your smartphone.



ThingSpeak

IoT Cloud Platform for Datalogging and Monitoring

Hans-Petter Halvorsen

[Table of Contents](#)

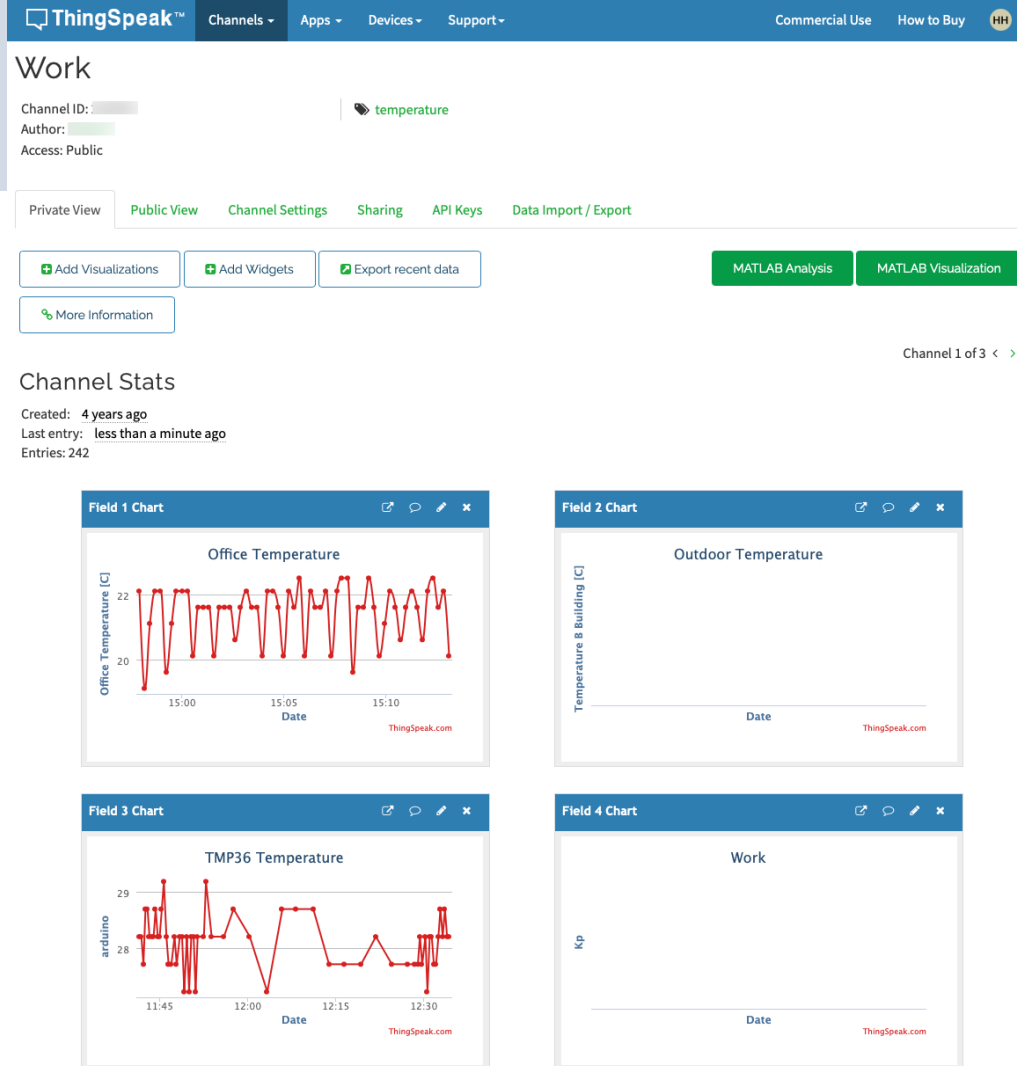
ThingSpeak

- ThingSpeak is an IoT service that lets you collect and store sensor data in the cloud and develop Internet of Things applications.
- ThingSpeak is free for small non-commercial projects
- In addition, they offer different types of licenses where you pay a monthly fee
- ThingSpeak is owned by MathWorks, the same vendor that develop the MATLAB software
- <https://thingspeak.com>

ThingSpeak

Here you see an example of how Data can be presented in the ThingSpeak Web page

<https://thingspeak.com>



ThingSpeak

- It works with Arduino, Raspberry Pi and MATLAB
 - Premade Libraries and APIs exists
- But it should work with all kind of Programming Languages, since it uses a **REST API** and **HTTP**.

ThingSpeak – Channel Settings

Channel ID: [redacted]
Author: hansha
Access: Public

temperature

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Channel Settings

Percentage complete 65%

Channel ID

Name Work

Description

Field 1 Office Temperature [C] ☒

Field 2 Temperature B Buildin ☒

Field 3 Tout ☒

Field 4 Kp ☒

Field 5 Ti ☒

Field 6 SP ☒

Field 7 Field7 ☒

Field 8 Field8 ☒

Help

Channels store all the eight fields that can have status data. Once you visualize it.

Channel Settings

- **Percentage complete:** Enter the percentage of data that is complete for the channel.
- **Channel Name:** Enter the name of the channel.
- **Description:** Enter the description of the channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
 - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - **Elevation:** Specify the elevation position in meters. For example, the elevation of the city of London is 35.052.

ThingSpeak

- You can set up different Channels in ThingSpeak.
- Each Channel can have up to 8 Fields.
- You can have up to 4 different Channels for the Free License.
- For the free license each channel can only be updated every 15 seconds.



ThingSpeak Example 1

Using the ThingSpeak Library by MathWorks

Hans-Petter Halvorsen

[Table of Contents](#)

ThingSpeak Library by MathWorks



Start by installing the ThingSpeak
Arduino Library made by MathWorks

<https://www.arduino.cc/reference/en/libraries/thingspeak/>

ThingSpeak Functions

```
#include "ThingSpeak.h"
```

The ThingSpeak Arduino Library has different functions that we can use, e.g.,:

- **WriteSingleField**: Writing a value to a single field on ThingSpeak.
- **WriteMultipleFields**: Writing values to multiple fields and status in one transaction with ThingSpeak.
- **ReadField**: Reading from a public channel and a private channel on ThingSpeak.
- **ReadMultipleFields**: Reading values from multiple fields, status, location, created-at timestamp from a public channel on ThingSpeak.

<https://github.com/mathworks/thingspeak-arduino>

ThingSpeak Functions

```
int writeField(channelNumber, field, value, writeAPIKey)
```

We will only focus on sending/writing one value at the time to ThingSpeak and therefore we will only focus on the **writeField** function in this Tutorial. See GitHub Documentation for more information about the different functions available.

<https://github.com/mathworks/thingspeak-arduino>

Basic Code Structure

```
#include <WiFiS3.h>
#include "secrets.h"
#include "ThingSpeak.h"
```

```
WiFiClient client;
```

```
..
```

```
void setup() {
    ConnectWiFi();
    ThingSpeak.begin(client);
}
```

```
void loop() {
    ThingSpeak.writeField(channelID, channelField, channelValue, writeAPIKey);
    ..
}
..
```

secrets.h

```
#define SECRET_SSID "xxxxx"
#define SECRET_PASS "xxxxx"

#define SECRET_CH_ID xxxxxx
#define SECRET_WRITE_APIKEY "xxxxxxxxxxx"
```

It is recommended to put WiFi and ThingSpeak credentials in a separate file called, e.g., secrets.h

ThingSpeak – Write API Key

The screenshot shows the ThingSpeak 'Write API Key' page. At the top, channel information is displayed: Channel ID, Author, and Access (Public). Navigation tabs include Private View, Public View, Channel Settings, Sharing, API Keys, and Data. The 'API Keys' tab is active, showing the 'Write API Key' section with a key input field and a 'Generate New Write API Key' button. Below this is the 'Read API Keys' section with a key input field, a note text area, and 'Save Note' and 'Delete API Key' buttons. A 'Help' section on the right explains the 'Write API Key' and 'Read API Keys' and provides API request examples. Two red arrows point from text boxes to the key input fields: one to the 'Write API Key' field and one to the 'Read API Key' field.

Channel ID: | temperature

Author:

Access: Public

Private View Public View Channel Settings Sharing API Keys Data

Write API Key

Key

Generate New Write API Key

Read API Keys

Key

Note

Save Note Delete API Key

Add New Read API Key

Help

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key= &field=
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels//feeds.json?results=2
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels//fields/1.json?results=2
```

Key needed to Write Data to the Channel

Key needed to Read Data from the Channel

Basic Code Structure - Explained

Arduino UNO R4 WiFi uses **WiFiS3.h** WiFi Library for WiFi Communication. For other Arduino boards you need to use other WiFi Libraries

```
#include <WiFiS3.h>
#include "secrets.h"
#include "ThingSpeak.h"
```

```
WiFiClient client;
```

```
..
```

```
void setup() {
  ConnectWiFi();
  ThingSpeak.begin(client);
}
```

```
void loop() {
  ThingSpeak.writeField(channelID, channelField, channelValue, writeAPIKey);
  ..
}
```

Put your WiFi Password, your ThingSpeak ChannelID and WriteAPIKey in **secrets.h**

Number between 1 and 8. Each Channel in ThingSpeak has 8 Fields



Here is the **writeField()** Function used

You find ChannelID and the WriteAPIKey on your ThingSpeak website

Main Program

secrets.h

```
#define SECRET_SSID "xxxxx"
#define SECRET_PASS "xxxxx"

#define SECRET_CH_ID xxxxxx
#define SECRET_WRITE_APIKEY "xxxxxxxxxxx"
```

```
#include <WiFiS3.h>
#include "secrets.h"
#include "ThingSpeak.h"
```

```
WiFiClient client;
```

```
char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int status = WL_IDLE_STATUS;
```

```
void setup() {
    Serial.begin(9600);
    ConnectWiFi();
    ThingSpeak.begin(client);
}
```

```
void loop()
{
    float temperature;
    temperature = random(2000,3000)/100.0;
    Serial.println(temperature);
    ThingSpeakWrite(temperature);
    delay(20000);
}
```

```

void ConnectWiFi()
{
    // check for the WiFi module:
    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println("Communication with WiFi module failed!");
        while (true);
    }

    String fv = WiFi.firmwareVersion();
    if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
        Serial.println("Please upgrade the firmware");
    }

    // Attempt to connect to WiFi network:
    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to WPA SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network:
        status = WiFi.begin(ssid, pass);

        // wait 10 seconds for connection:
        delay(10000);
    }

    // You're connected now, so print out the data:
    Serial.println("You're connected to Wifi");
    PrintNetwork();
}

```

Functions for connection to WiFi

```

void PrintNetwork()
{
    Serial.print("WiFi Status: ");
    Serial.println(WiFi.status());

    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
}

```

Function for Writing Data to ThingSpeak

Function for writing Data to ThingSpeak

```
void ThingSpeakWrite(float channelValue)
{
    unsigned long myChannelNumber = SECRET_CH_ID;
    const char * myWriteAPIKey = SECRET_WRITE_APIKEY;
    int channelField = 1;
    int x = ThingSpeak.writeField(myChannelNumber, channelField, channelValue, myWriteAPIKey);
    if(x == 0){
        Serial.println("Channel updated successfully.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
}
```

The Function **writeField()** is part of the Arduino ThingSpeak library which we have downloaded and are using by including ThingSpeak.h on top of our Arduino code

Channel ID: [REDACTED]

Author: [REDACTED]

Access: Private

Final Results

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

MATLAB Analysis

Export recent data

MATLAB Visualization

More Information

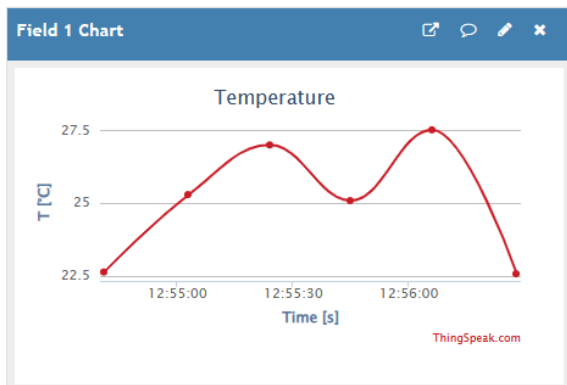
Channel 1 of 3 < >

Channel Stats

Created: 6 years ago

Last entry: less than a minute ago

Entries: 6



```
thingspeak_ex.ino secrets.h
1 #include <WiFi3.h>
2 #include "secrets.h"
3 #include "ThingSpeak.h"
4
5 WiFiClient client;
6
7 char ssid[] = SECRET_SSID;
8 char pass[] = SECRET_PASS;
9 int status = WL_IDLE_STATUS;
10
11 void setup()
12 {
13   Serial.begin(9600);
14   ConnectWiFi();
15   ThingSpeak.begin(client);
16 }
17
18 void loop()
19 {
20   float temperature;
21   temperature = random(2000,3000)/100.0;
22   Serial.println(temperature);
23   ThingSpeakWrite(temperature);
24   delay(20000);
25 }
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino UNO R4 WiFi' on '...') New Line 9600 baud

Channel updated successfully.

25.29

Channel updated successfully.

27.00

Channel updated successfully.

25.08

Channel updated successfully.



ThingSpeak

Example 2

Using HTTP and the available REST API

Hans-Petter Halvorsen

[Table of Contents](#)

ThingSpeak - REST API

The screenshot shows the ThingSpeak API Keys management interface. At the top, channel information is displayed: Channel ID, Author, and Access (Public). Navigation tabs include Private View, Public View, Channel Settings, Sharing, API Keys, and Data. The 'API Keys' tab is active, showing sections for 'Write API Key' and 'Read API Keys'. The 'Write API Key' section has a text input for the key and a 'Generate New Write API Key' button. The 'Read API Keys' section has a text input for the key, a 'Note' text area, and buttons for 'Save Note' and 'Delete API Key'. A 'Help' section on the right provides instructions on using the keys. Two red arrows point from text boxes to the key input fields: one from 'Key needed to Write Data to the Channel' to the 'Write API Key' field, and another from 'Key needed to Read Data from the Channel' to the 'Read API Keys' field. A third red arrow points from the 'API Keys' tab to the 'Key needed to Write Data to the Channel' box. A fourth red arrow points from the 'API Keys Settings' section to the 'Key needed to Read Data from the Channel' box.

Channel ID: | temperature

Author:

Access: Public

Private View Public View Channel Settings Sharing API Keys Data

Write API Key

Key

[Generate New Write API Key](#)

Read API Keys

Key

Note

[Save Note](#) [Delete API Key](#)

[Add New Read API Key](#)

Help

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key= &field=
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels//feeds.json?results=2
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels//fields/1.json?results=2
```

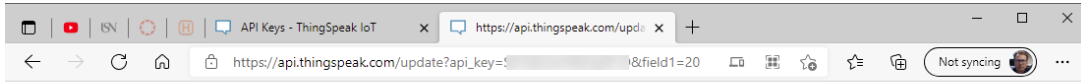
Key needed to Write Data to the Channel

Key needed to Read Data from the Channel

REST API – Write Data

Use your standard Web Browser (e.g., Microsoft Edge, or Google Chrome) and enter the following:

`https://api.thingspeak.com/update?api_key=XXXXXXXXXXXXXXXXXXXX&field1=20`



19

Your **Write API Key**

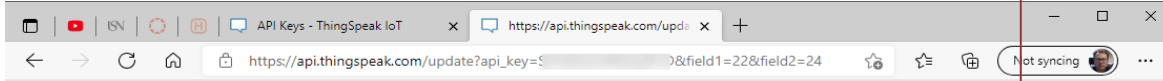
Field Number 1-8

Value

REST API – Write Multiple Fields

Use your standard Web Browser (e.g., Microsoft Edge, or Google Chrome) and enter the following:

`https://api.thingspeak.com/update?api_key=XXXXXXXXXXXXXXXXXXXX&field1=21&field2=24`



21

Your **Write API Key**

Field + Value

Field + Value Etc.

REST API – Read Data

Use your standard Web Browser (e.g., Microsoft Edge, or Google Chrome) and enter the following:

`https://api.thingspeak.com/channels/xxxxxx/fields/1.json?results=10`

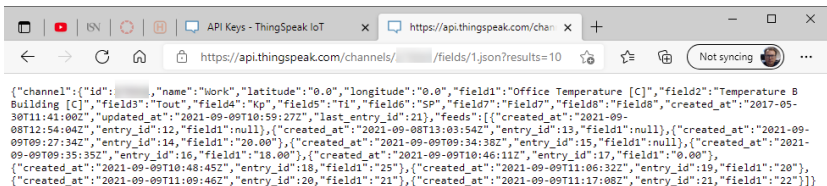
Data Format (JSON or XML)

Your Channel ID

Field Number

Number of Data Points, e.g., 1 for only the last value, 10 for the last 10 values, etc.

Resulting JSON String with Data



Main Program

```
#include <WiFiS3.h>
#include "secrets.h"

WiFiClient client;

char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int status = WL_IDLE_STATUS;

void setup()
{
    Serial.begin(9600);
    ConnectWiFi();
}

void loop()
{
    float temperature;
    temperature = random(2000,3000)/100.0;
    Serial.println(temperature);
    ThingSpeakWrite(temperature);
    delay(20000);
}
```

```

void ConnectWiFi()
{
    // check for the WiFi module:
    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println("Communication with WiFi module failed!");
        while (true);
    }

    String fv = WiFi.firmwareVersion();
    if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
        Serial.println("Please upgrade the firmware");
    }

    // Attempt to connect to WiFi network:
    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to WPA SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network:
        status = WiFi.begin(ssid, pass);

        // wait 10 seconds for connection:
        delay(10000);
    }

    // You're connected now, so print out the data:
    Serial.println("You're connected to Wifi");
    PrintNetwork();
}

```

Functions for connection to WiFi

```

void PrintNetwork()
{
    Serial.print("WiFi Status: ");
    Serial.println(WiFi.status());

    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
}

```

```

void ThingSpeakWrite(float fieldValue)
{
    char server[] = "api.thingspeak.com";
    unsigned long channelNumber = SECRET_CH_ID;
    String writeAPIKey = SECRET_WRITE_APIKEY;
    int channelField = 1;

    if (client.connect(server, 80))
    {
        String postData= "api_key=" + writeAPIKey + "&field" + String(channelField) + "=" + String(fieldValue);

        client.println( "POST /update HTTP/1.1" );
        client.println( "Host: api.thingspeak.com" );
        client.println( "Connection: close" );
        client.println( "Content-Type: application/x-www-form-urlencoded" );
        client.println( "Content-Length: " + String( postData.length() ) );
        client.println();
        client.println( postData );
    }
    else
    {
        Serial.println ( "Connection Failed" );
    }
}

```

Function for writing Data to ThingSpeak

https://api.thingspeak.com/update?api_key=xxxxxx&field1=20

Channel ID: Author:

Access: Private

Final Results

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

MATLAB Analysis

Export recent data

MATLAB Visualization

More Information

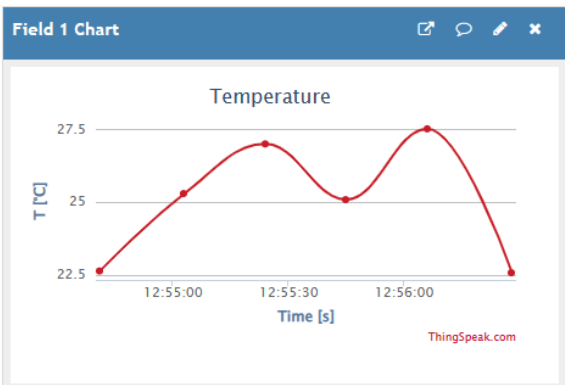
Channel 1 of 3 < >

Channel Stats

Created: 6 years ago

Last entry: less than a minute ago

Entries: 6



Arduino UNO R4 WiFi

```
thingspeak_ex2.ino secrets.h
1 #include <WiFi3.h>
2 #include "secrets.h"
3
4 WiFiClient client;
5
6 char ssid[] = SECRET_SSID;
7 char pass[] = SECRET_PASS;
8 int status = WL_IDLE_STATUS;
9
10 void setup()
11 {
12     Serial.begin(9600);
13     ConnectWiFi();
14 }
15
16 void loop()
17 {
18     float temperature;
19     temperature = random(2000,3000)/100.0;
20     Serial.println(temperature);
21     ThingSpeakWrite(temperature);
22     delay(20000);
23 }
24
25 void ConnectWiFi()
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino UNO R4 WiFi on '... New Line 9600 baud

Attempting to connect to WPA SSID: Windows10HPH
You're connected to Wifi
WiFi Status: 3
SSID:
IP Address: 192.168.137.57
29.33
27.43
22.62

Ln 5, Col 1 Arduino UNO R4 WiFi on COM9 2

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

