

<https://www.halvorsen.blog>



# ThingSpeak

Change Values from a HTML Webpage

Hans-Petter Halvorsen

# Contents

- [Introduction](#)
- [ThingSpeak](#)
- [HTML Webpage](#)
- [Improvements](#)

## PID Controller Management

Here you can change the settings for the Process Control System.

### PID Parameters

Kp:

Ti [s]:

Td [s]:

### Process Settings

Setpoint [°C]:



# Introduction

# Introduction

- In this Tutorial we will see how we can update values in ThingSpeak from a Webpage
- ThingSpeak is an IoT service that lets you collect and store sensor data in the cloud

# Introduction

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾ Commercial Use How to Buy HH

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

+ Add Visualizations + Add Widgets

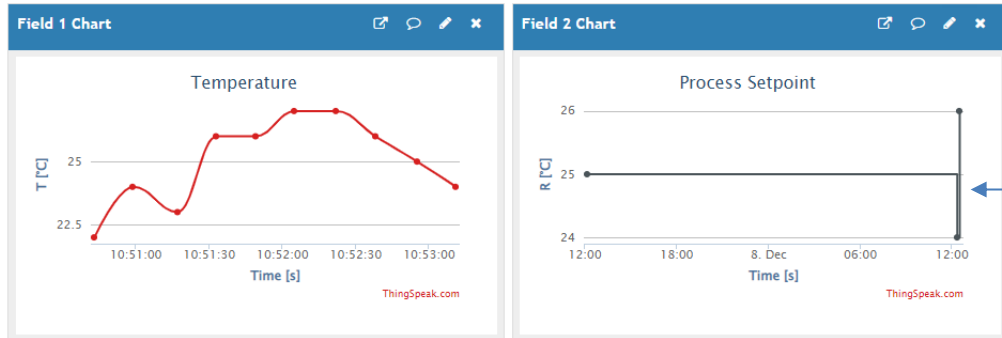
Export recent data More Information

MATLAB Analysis MATLAB Visualization

Channel 1 of 3 < >

## Channel Stats

Created: 5 years ago  
Last entry: 3 minutes ago  
Entries: 15



## Change Setpoint

Here you can change the setpoint for your process.

Setpoint [°C]:

Submit

# Final Solution

We create the Webpage step by step and end with the following:

In addition to the Webpage, we can, e.g., have a PID Controller running on an Arduino or Raspberry Pi (will not be shown in this Tutorial).

This Application then Controls a given Process and get updated PID Settings from ThingSpeak.

## PID Controller Management

Here you can change the settings for the Process Control System.

### PID Parameters

Kp:

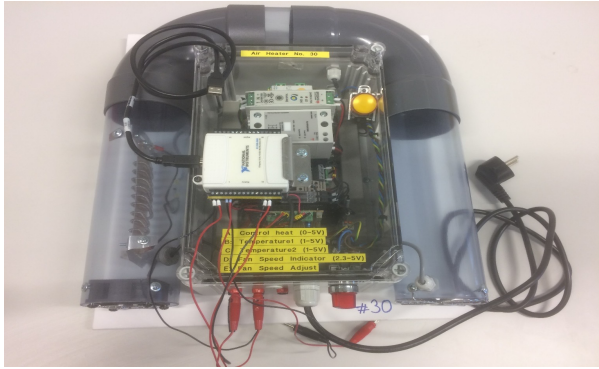
Ti [s]:

Td [s]:

### Process Settings

Setpoint [°C]:

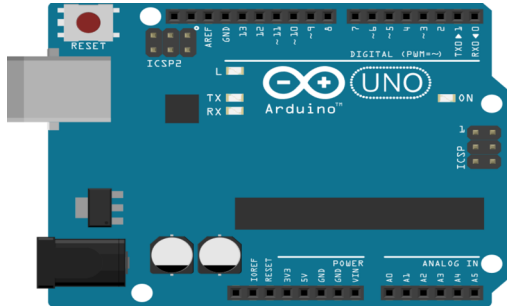
# Example of Final Solution



Process



Arduino



Process Value and Control Value



Get updated Settings



Webpage

## PID Controller Management

Here you can change the settings for the Process Control System.

### PID Parameters

Kp:

Ti [s]:

Td [s]:

### Process Settings

Setpoint [°C]:



ThingSpeak

Embedded PID Controller with no GUI



# ThingSpeak



# 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
- <https://thingspeak.com>

# ThingSpeak

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

<https://thingspeak.com>

The screenshot displays the ThingSpeak web interface for a channel named "temperature". The top navigation bar includes "Channels", "Apps", "Devices", and "Support". The channel information shows "Channel ID: [redacted]", "Author: [redacted]", and "Access: Public". Navigation tabs include "Private View", "Public View", "Channel Settings", "Sharing", "API Keys", and "Data Import / Export". Action buttons for "Add Visualizations", "Add Widgets", "Export recent data", and "More Information" are visible, along with "MATLAB Analysis" and "MATLAB Visualization" buttons.

**Channel Stats**  
Created: 4 years ago  
Last entry: less than a minute ago  
Entries: 242

**Field 1 Chart: Office Temperature**  
A line chart showing "Office Temperature [C]" on the y-axis (ranging from 20 to 22) against "Date" on the x-axis (ranging from 15:00 to 15:10). The data shows a fluctuating red line between approximately 19.5°C and 22.5°C.

**Field 2 Chart: Outdoor Temperature**  
A line chart showing "Outdoor Temperature [C]" on the y-axis against "Date" on the x-axis. The chart area is currently blank, indicating no data is displayed.

**Field 3 Chart: TMP36 Temperature**  
A line chart showing "TMP36 Temperature" on the y-axis (ranging from 28 to 29) against "Date" on the x-axis (ranging from 11:45 to 12:30). The data shows a red line fluctuating between approximately 27.5 and 29.5.

**Field 4 Chart: Work**  
A line chart showing "Work" on the y-axis against "Date" on the x-axis. The chart area is currently blank, indicating no data is displayed.

# 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**.
- **LabVIEW** has built-in **HTTP Client** functions that you can use for this purpose
- **MQTT** API also available

# 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 [redacted]

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 data that a ThingSpeak app collects. Each channel has up to eight fields that can hold any type of data, plus the status data. Once you collect data in a channel, you can visualize it.

## Channel Settings

- **Percentage complete:** Calculated based on the number of fields that are complete. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak 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 meters. For example, the elevation of the city of London is 35.052.

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.

# ThingSpeak - REST API

The screenshot shows the ThingSpeak REST API interface for a channel named 'temperature'. The channel ID is [redacted], the author is [redacted], and the access is public. The interface includes tabs for Private View, Public View, Channel Settings, Sharing, API Keys, and Data. The 'API Keys' tab is active, showing two sections: 'Write API Key' and 'Read API Keys'. The 'Write API Key' section has a 'Key' input field and a 'Generate New Write API Key' button. The 'Read API Keys' section has a 'Key' input field, a 'Note' text area, and 'Save Note' and 'Delete API Key' buttons. There is also an 'Add New Read API Key' button at the bottom. On the right side, there is a 'Help' section with 'API Keys Settings' and 'API Requests'.

Channel ID: [redacted] | temperature

Author: [redacted]

Access: Public

Private View Public View Channel Settings Sharing API Keys Data

Write API Key

Key [redacted]

Generate New Write API Key

Read API Keys

Key [redacted]

Note [redacted]

Save Note Delete API Key

Add New Read API Key

Key needed to Write Data to the Channel

Key needed to Read Data from the Channel

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=[redacted]&field=[redacted]
```

Read a Channel Feed

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

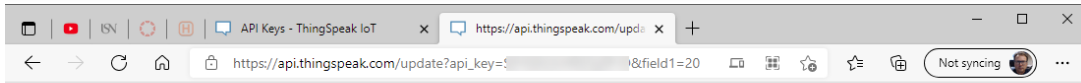
Read a Channel Field

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

# 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`



Your **Write API Key**

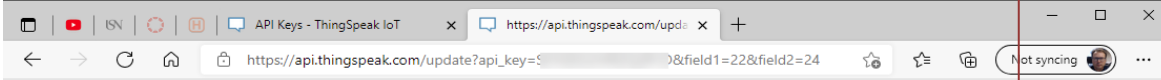
Field Number 1-8

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

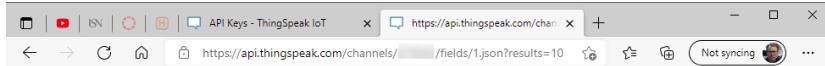
Field + Value Etc.

# REST API – Read Data

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

Data Format (JSON or XML)

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



```
{ "channel": { "id": "1", "name": "Work", "latitude": "0.0", "longitude": "0.0", "field1": "Office Temperature [C]", "field2": "Temperature B  
Building [C]", "field3": "Tou", "field4": "Kp", "field5": "T1", "field6": "Sp", "field7": "Field7", "field8": "field8", "created_at": "2017-05-  
30T11:41:00Z", "updated_at": "2021-09-09T10:59:27Z", "last_entry_id": 21, "feeds": [{"created_at": "2021-09-  
08T12:54:04Z", "entry_id": 12, "field1": null}, {"created_at": "2021-09-08T13:03:54Z", "entry_id": 13, "field1": null}, {"created_at": "2021-09-  
09T09:27:34Z", "entry_id": 14, "field1": "20.00"}, {"created_at": "2021-09-09T09:34:38Z", "entry_id": 15, "field1": null}, {"created_at": "2021-  
09-09T09:35:35Z", "entry_id": 16, "field1": "18.00"}, {"created_at": "2021-09-09T10:46:11Z", "entry_id": 17, "field1": "0.00"},  
{"created_at": "2021-09-09T10:48:45Z", "entry_id": 18, "field1": "21"}, {"created_at": "2021-09-09T11:06:32Z", "entry_id": 19, "field1": "20"},  
{"created_at": "2021-09-09T11:09:46Z", "entry_id": 20, "field1": "21"}, {"created_at": "2021-09-09T11:17:08Z", "entry_id": 21, "field1": "22"}]}
```

Your Channel ID

Field Number

Resulting JSON String with Data

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





# HTML Webpage

# Webpage

## Change Setpoint

Here you can change the setpoint for your process.

Setpoint:

°C

# HTML Code

```
<!DOCTYPE html>
<html lang="en">

<script>
function updateSetpoint()
{
  var sp = document.getElementById("setpoint").value;
  var website = "https://api.thingspeak.com/update?api_key=IEQ6WFMNE99LNLIS&field2=" + sp;
  window.location.href = website;
}
</script>

<head>
  <meta charset="UTF-8">
  <title>Change Setpoint</title>
</head>

<body>

  <div>

    <h1>Change Setpoint</h1>
    <p>Here you can change the setpoint for your process.</p>

    <form action="javascript: updateSetpoint()">
      <label for="setpoint">Setpoint:</label><br>
      <input type="text" id="setpoint" name="setpoint" value="25">°C
      <br>

      <input type="submit" value="Submit">
    </form>

  </div>

</body>
</html>
```

# Results

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

MATLAB Analysis

MATLAB Visualization

Export recent data

More Information

Channel 1 of 3 < >

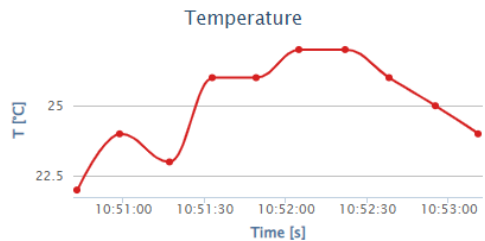
## Channel Stats

Created: [5 years ago](#)

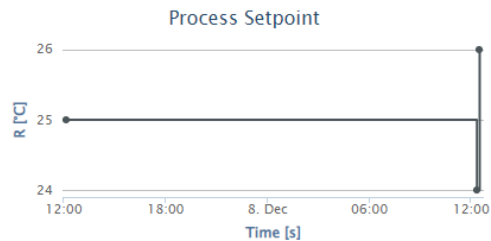
Last entry: [3 minutes ago](#)

Entries: 15

Field 1 Chart



Field 2 Chart



## Change Setpoint

Here you can change the setpoint for your process.

Setpoint:

°C



# Improvements

# Improvements

- Add Stylesheet
- Add Bootstrap
- Add type="Number"
- Update PID Settings + Setpoint
- ..

# Stylesheet

```
..  
  
<head>  
  <meta charset="UTF-8">  
  <title>Change Setpoint</title>  
  <link rel="stylesheet" href="stylesheet.css">  
</head>  
  
..
```

## Change Setpoint

Here you can change the setpoint for your process.

Setpoint:

 °C

# stylesheet.css

```
body {  
  font-family: "Sans-serif", Verdana;  
  color: #153744;  
  text-align: left;  
}  
  
h1 {  
  font-size: 40px;  
  font-family: "Sans-serif", Verdana;  
  color: #153744;  
  text-align: left;  
}
```



# Bootstrap

- Bootstrap is a popular HTML, CSS, and JavaScript framework
- Bootstrap is free to download and use
- Resources
  - <https://getbootstrap.com>
  - <https://www.w3schools.com/bootstrap5/index.php>

```
<!DOCTYPE html>
<html lang="en">

<script>
function updateSetpoint()
{
  var sp = document.getElementById("setpoint").value;
  var website = "https://api.thingspeak.com/update?api_key=IEQ6WFMNE99LNLIS&field2=" + sp;
  window.location.href = website;
}
</script>

<head>
  <title>Change Setpoint</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
</head>

<body>

<div class="container-fluid">

  <h1>Change Setpoint</h1>
  <p>Here you can change the setpoint for your process.</p>

  <form action="javascript: updateSetpoint()">
    <label for="setpoint">Setpoint [°C]:</label><br>
    <input type="number" id="setpoint" name="setpoint" class="form-control" value="25" min="0" max="50">

    <input type="submit" value="Submit" class="btn btn-danger">
  </form>

</div>

</body>
</html>
```

# Type="Number"

```
<input type="number" id="setpoint" name="setpoint" value="25" min="0" max="50">
```

# Webpage

## Change Setpoint

Here you can change the setpoint for your process.

Setpoint [°C]:

# Update PID Settings

## PID Controller Management

Here you can change the settings for the Process Control System.

### PID Parameters

Kp:

Ti [s]:

Td [s]:

### Process Settings

Setpoint [°C]:

```
<!DOCTYPE html>
<html lang="en">

<script>
function updatePid()
{
  var Kp = document.getElementById("Kp").value;
  var Ti = document.getElementById("Ti").value;
  var Td = document.getElementById("Td").value;
  var sp = document.getElementById("setpoint").value;
  var website = "https://api.thingspeak.com/update?api_key=IEQ6WFMNE99LNLIS&field2=" + sp + "&field3=" + Kp + "&field4=" + Ti + "&field5=" + Td;
  window.location.href = website;
}
</script>

<head>
<title>PID Controller Management</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RwuH61DGLwZJEdK2KadqF9CUG65" crossorigin="anonymous">
</head>

<body>

<div class="container-fluid">

  <h1>PID Controller Management</h1>
  <p>Here you can change the settings for the Process Control System.</p>

  <form action="javascript: updatePid()">

    <h2>PID Parameters</h2>

    <label for="Kp">Kp:</label><br>
    <input type="number" id="Kp" name="Kp" value="1" min="0" max="50" class="form-control">

    <label for="Ti">Ti [s]:</label><br>
    <input type="number" id="Ti" name="Ti" value="10" min="0" max="1000" class="form-control">

    <label for="Td">Td [s]:</label><br>
    <input type="number" id="Td" name="Td" value="0" min="0" max="100" class="form-control">

    <br/>
    <h2>Process Settings</h2>

    <label for="setpoint">Setpoint [°C]:</label><br>
    <input type="number" id="setpoint" name="setpoint" value="25" min="0" max="50" class="form-control">

    <br/>
    <input type="submit" value="Submit" class="btn btn-danger">
  </form>

</div>

</body>
</html>
```

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

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

