

Dokumentation zur Erstellung des Dashboards

Inhaltsverzeichnis

Was wird benötigt?	1
Funktionsweise des Dashboards	1
Integration der Daten von TheThingsNetwork zu ThingSpeak	1
Diagramm für Dashboard in ThingSpeak erstellen	8
Kennzahlenboxen für Dashboard in ThingSpeak erstellen	10
ThingSpeak Channel als Quellen für Dashboard einsetzen	16

Was wird benötigt?

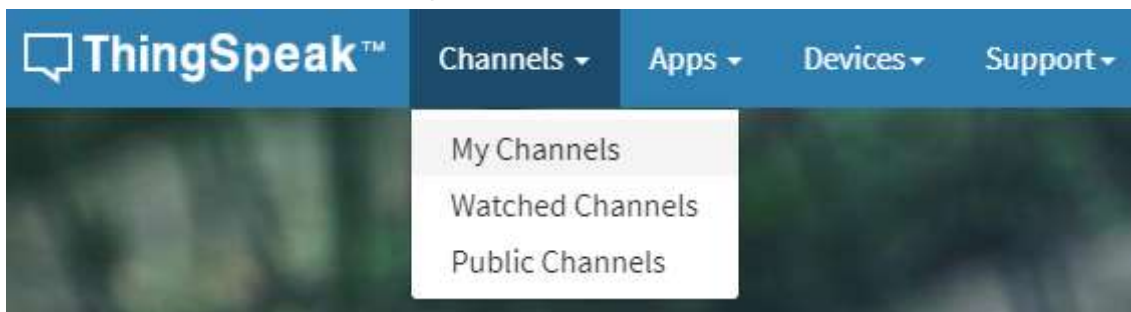
1 X MatLab-Lizenz [Kostenlose Testversion von MATLAB - MATLAB & Simulink \(mathworks.com\)](https://www.mathworks.com/products/matlab/try-now.html)

Funktionsweise des Dashboards

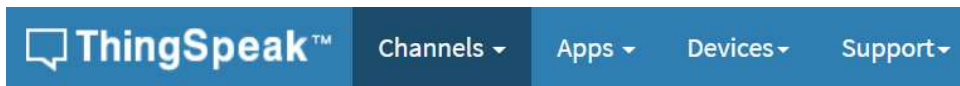
Für die Datenspeicherung und -visualisierung bietet sich die cloudbasierte ThingSpeak-Plattform an. Dort können in einem Channel die Sensordaten gespeichert und in einem weiteren näher analysiert sowie in einem Dashboard visualisiert werden. Zuvor muss allerdings eine Integration zur Applikation im TheThingsNetwork erstellt werden, die für die Datenbereitstellung sorgt.

Integration der Daten von TheThingsNetwork zu ThingSpeak

1. Kostenfrei registrieren bei [IoT Analytics - ThingSpeak Internet of Things](https://thingspeak.com/)
2. Neuen Channel für das Speichern der Daten erstellen.



3. Auf "New Channel" klicken und dann einen Namen eintragen sowie die ersten beiden Felder ausfüllen. Mit dem Klick auf "Save Channel" (unten) wird der Channel erstellt.



New Channel

Name	<input type="text" value="HTW_Lernampel_Daten"/>	
Description	<input type="text"/>	
Field 1	<input type="text" value="Lautstaerke"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Personenzahl"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text"/>	<input type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>

4. Öffnen des Tabs API Keys. Die ChannelID (oben) und der Write API Key sind so sichtbar für den nächsten Schritt.

HTW_Lernampel_Daten

Channel ID: **1488846**

Author: **mwa0000023508449**

Access: Private

[Private View](#)

[Public View](#)

[Channel Settings](#)

[Sharing](#)

[API Keys](#)

Write API Key

Key

MPC3TVKKGWIQKW8V

[Generate New Write API Key](#)

Read API Keys

Key

9UDZBOUNECDEI1SP

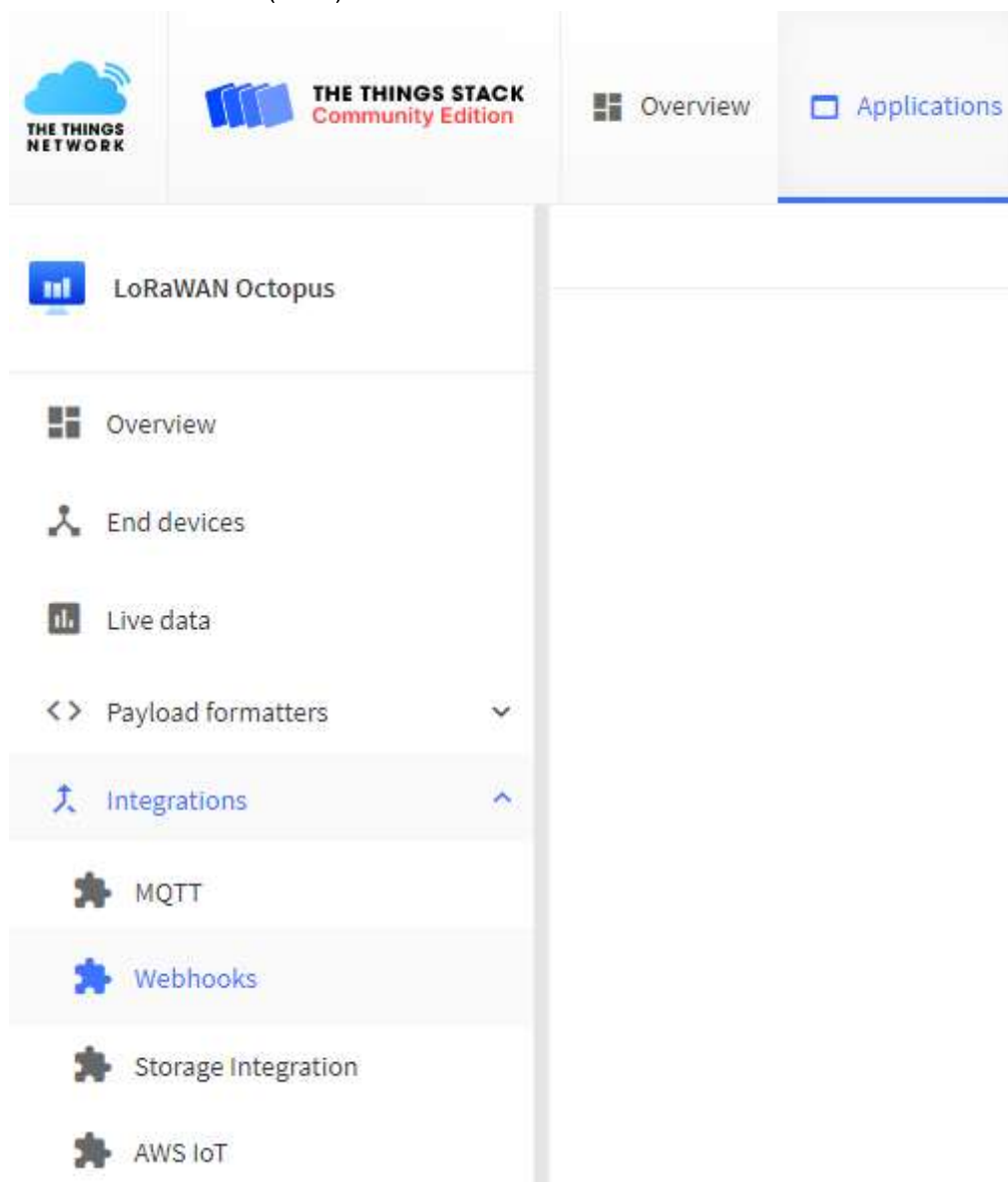
Note

[Save Note](#)

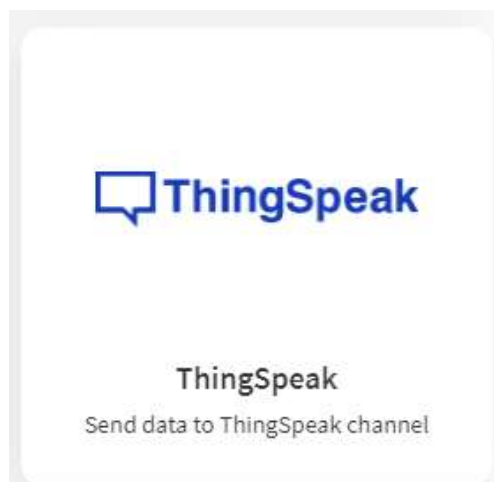
[Delete API Key](#)

[Add New Read API Key](#)

5. Öffnen der erstellten Applikation in TheThingsNetwork und navigieren zu Integrations - Webhooks (links).



6. "Add new Webhook" anklicken und ThingSpeak auswählen



7. Festlegen einer Webhook ID und übertragen der **eigenen** Channel ID und des API Keys aus der ThingSpeak Übersicht unter 4. Mit "Create thingspeak webhook" die Verbindung zur Application erstellen.

Applications > LoRaWAN Octopus > Webhooks > Add > ThingSpeak

Add custom webhook

Template information

 **ThingSpeak**

ThingSpeak
Send data to ThingSpeak channel

[About ThingSpeak](#) | [Documentation](#)

Template settings

Webhook ID *

Channel ID *

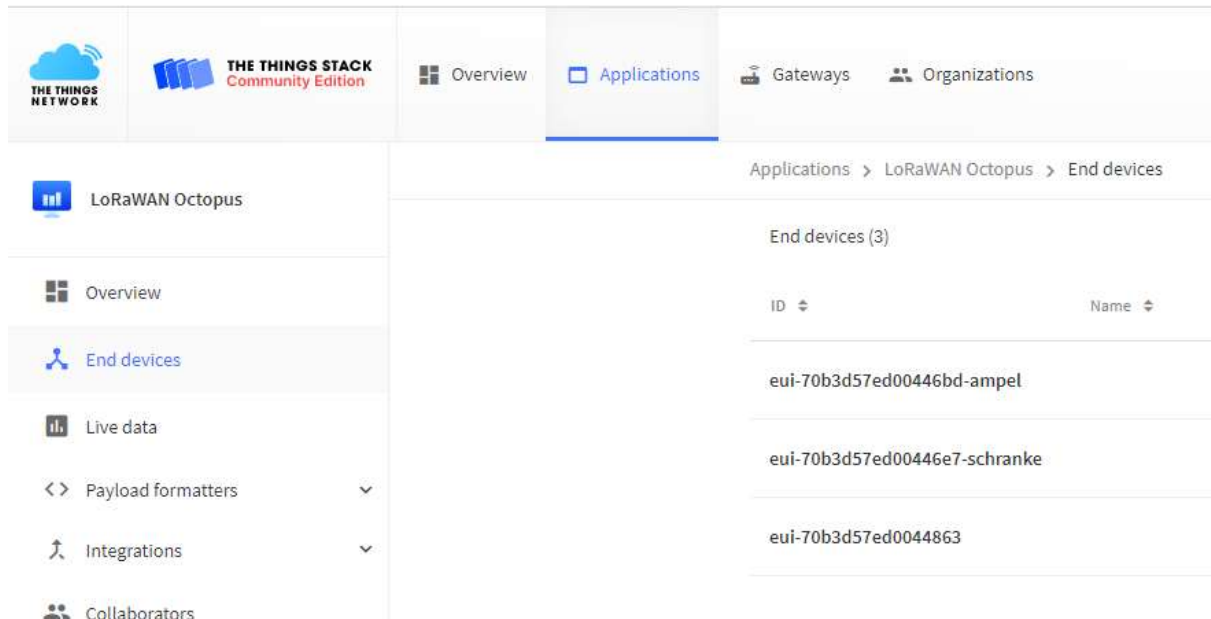
ThingSpeak Channel ID

API Key *

ThingSpeak Write API Key

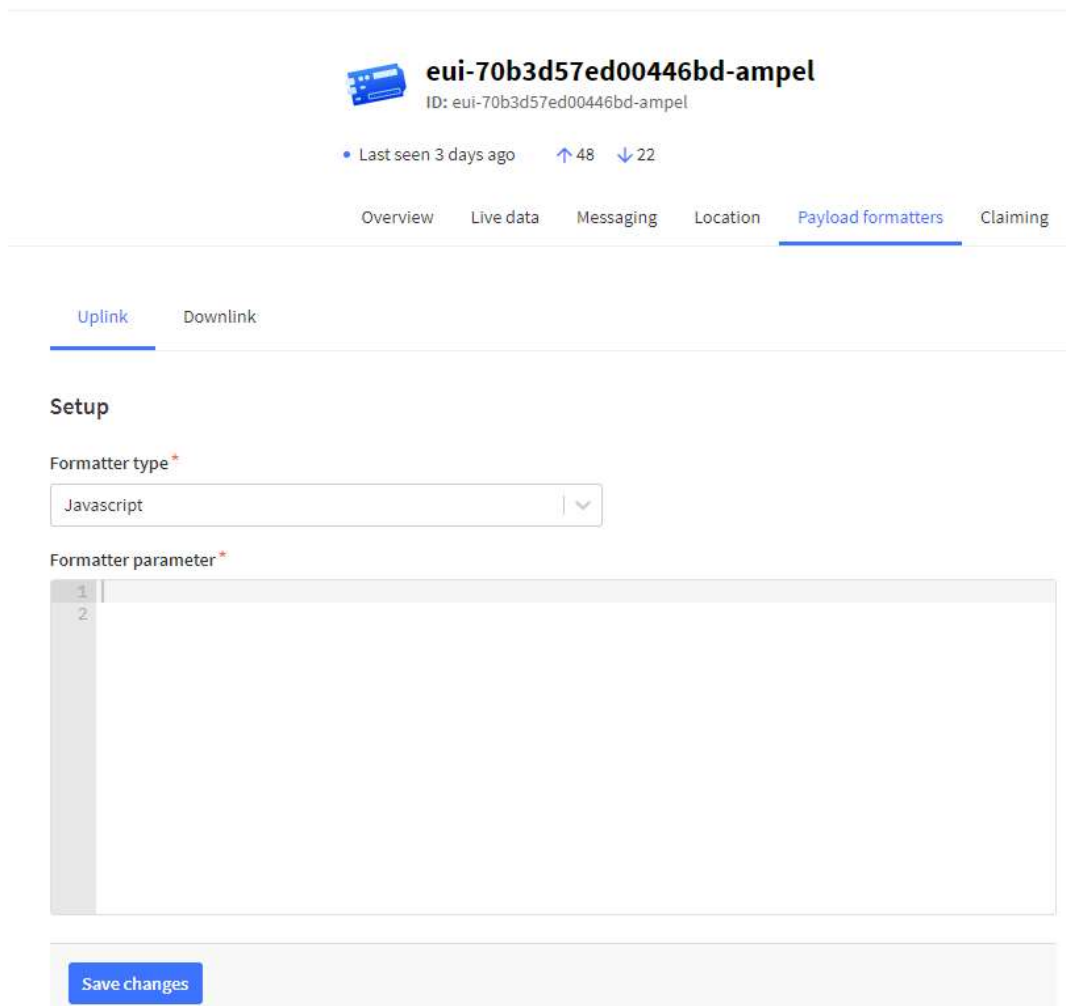
Create thingspeak webhook

8. Öffnen der erstellten Applikation in TheThingsNetwork und navigieren zu End Devices (links).



The screenshot shows the The Things Network interface. At the top, there's a navigation bar with 'THE THINGS NETWORK' logo, 'THE THINGS STACK Community Edition', and tabs for 'Overview', 'Applications' (selected), 'Gateways', and 'Organizations'. Below this, the left sidebar shows a list of application tabs: 'LoRaWAN Octopus' (selected), 'Overview', 'End devices' (selected), 'Live data', 'Payload formatters', 'Integrations', and 'Collaborators'. The main content area shows the 'End devices' tab for the 'LoRaWAN Octopus' application. It displays a list of 3 end devices with columns for 'ID' and 'Name'. The devices listed are: 'eui-70b3d57ed00446bd-ampel', 'eui-70b3d57ed00446e7-schranke', and 'eui-70b3d57ed0044863'.

9. Auswahl des Devices, das die **Lautstärke** misst und öffnen des “Payload formatters”-Tab.



The screenshot shows the 'Payload formatters' page for the device 'eui-70b3d57ed00446bd-ampel'. The page header includes the device name and ID, along with status information: 'Last seen 3 days ago', '48' upvotes, and '22' downvotes. Below this, there are tabs for 'Overview', 'Live data', 'Messaging', 'Location', 'Payload formatters' (selected), and 'Claiming'. The 'Payload formatters' tab is active, showing a 'Setup' section. Under 'Setup', there's a 'Formatter type' dropdown menu set to 'Javascript'. Below that, there's a 'Formatter parameter' section with a text area containing the number '1' on the first line and '2' on the second line. At the bottom of the page, there's a 'Save changes' button.

10. Unter "Uplink" den untenstehenden Code in die Box einfügen und mit Klick auf "Save Changes" sichern.

Wandelt die gesendeten Daten in eine Zahl um und weist sie dem ersten Feld im ThingSpeak Channel zu

```
function Decoder(b, port) {  
  var decoded = {};  
  decoded.port = port;  
  decoded.var1 = 0;  
  
  if (port === 1 ) {  
    var var1 = b[0];  
  }  
  
  return {  
    field1: var1  
  };  
}
```

11. Zurückkehren zur Übersicht der End Devices und das Device zum Zählen der Personen öffnen.

12. In den "Payload formatters"-Tab den folgenden Code in die Box einfüge und danach speichern.

Wandelt die gesendeten Daten in eine Zahl um und weist sie dem zweiten Feld im ThingSpeak Channel zu

```
function Decoder(b, port) {  
  
  var decoded = {};  
  decoded.port = port;  
  decoded.var1 = 0;  
  
  if (port === 1 ) {  
    var var1 = b[0];  
  }  
  
  return {  
    field2: var1  
  };  
}
```

13. Wenn die Applikation läuft, erscheinen die Daten im Zahlenformat zum einen in der TheThingsNetwork-Console und zum anderen im erstellten ThingSpeak-Channel.

Diagramm für Dashboard in ThingSpeak erstellen

Für die Gegenüberstellung der Anzahl an Personen in einem Raum und der gemessenen Lautstärke zu einem bestimmten Zeitpunkt bietet sich ein Liniendiagramm mit zwei Y-Achsen an (Lautstärke und Personenzahl).

1. Um das Dashboard MitschülerInnen zugänglich zu machen, sollte das Dashboard unter dem Tab "Public View" erstellt werden. Im ThinkSpeak-Channel zur Analyse der Daten muss dazu über den grünen Button "MATLAB Visualization" ein neues Feld dem Dashboard hinzugefügt werden.

The screenshot shows the ThingSpeak interface for a channel named 'HTW_Ampel_Analyse'. The channel ID is 1480622, the author is mwa0000023508449, and the access is public. The 'Public View' tab is selected. Below the tabs, there are buttons for 'Add Visualizations', 'Add Widgets', and 'Export recent data'. On the right, there are two green buttons: 'MATLAB Analysis' and 'MATLAB Visualization'. Below these buttons, the 'Channel Stats' section shows the channel was created 24 days ago, has a last entry 6 days ago, and contains 27 entries.

2. Eine neue Seite öffnet sich. Unter "Templates" soll "Custom (no starter code)" ausgewählt sein, da der nötige Code bereitgestellt wird. Anschließend auf den Button "Create" klicken.

The screenshot shows the 'Apps / MATLAB Visualizations / New' page in the ThingSpeak interface. The page has a blue header with the ThingSpeak logo and navigation links for Channels, Apps, Devices, and Support. The main content area is white and shows the breadcrumb 'Apps / MATLAB Visualizations / New'.


Templates:

- ☒ Custom (no starter code)
- ☐ Create a filled area 2-D plot
- ☐ Create a 2-D line plot
- ☐ Create 2-D line plots with y-axes on both left and right side
- ☐ Create a correlated data plot
- ☐ Create a discrete sequence data plot

Examples: Sample code to visualize data

- ☐ Use a histogram to understand variation in data
- ☐ Visualize directional data with compass plot

3. Im nächsten Schritt kann das Diagramm benannt werden als “Lautstärke und Personenzahl” und der folgende Code in das Feld für den MATLAB Code kopiert werden. Die Channel-IDs und möglicherweise die API Keys müssen wie in den Inline-Kommentaren zu erkennen ist, angepasst werden.

 Channels ▾ Apps ▾ Devices ▾ Support ▾

Apps / MATLAB Visualizations / Lautstärke und Personenzahl / Edit

Name

Lautstärke und Personenzahl

MATLAB Code

```
1 % Lautstärke - 1. Channel ID von dem Daten zur Lautstärke gelesen werden
2 readChannelID1 = 1476009;
3 fieldID1 = 1;
4
```

```
% Lautstärke - 1. Channel ID von dem Daten zur Lautstärke gelesen
werden
readChannelID1 = 1488846;
fieldID1 = 1;

% 1. Channel Read API Key
% Falls der Channel private ist, muss der read API Key zwischen ''
eingesetzt werden
readAPIKey1 = '9UDZBOUNECDEI1SP';

% Personenzahl - 2. Channel ID von dem Daten zur Personenanzahl
gelesen werden
readChannelID2 = 1488846;
fieldID2 = 2;

% 2. Channel Read API Key
% Falls der Channel private ist, muss der read API Key zwischen ''
eingesetzt werden
readAPIKey2 = '9UDZBOUNECDEI1SP';

%% Lesen der Daten %%
% Lesen der Personenzahl
[Lautstaerke, time1] = thingSpeakRead(readChannelID1, 'Field',
fieldID1, 'NumPoints', 45, 'ReadKey', readAPIKey1);

% Lesen der Lautstärke
[Personenzahl, time2] = thingSpeakRead(readChannelID2, 'Field',
fieldID2, 'NumPoints', 45, 'ReadKey', readAPIKey2);
```

```

%% Visualisierung der Daten %%
xlabel('Uhrzeit');

yyaxis left;
plot(time1, Lautstaerke)
ylabel('Lautstärke in Dezibel');

yyaxis right;
plot(time2, Personenzahl);
ylabel('Personenzahl');

```

4. Nach dem Anpassen des Codes kann über den Button “Save and Run” der Code gespeichert und ausgeführt werden. Das Ausgabefeld “MATLAB Plot Output” zeigt bei korrektem Code das gewünschte Diagramm. Falls dies nicht der Fall sein sollte, ist eine Überprüfung der Anpassungen zu empfehlen und die gezeigte Fehlermeldung zu beachten.

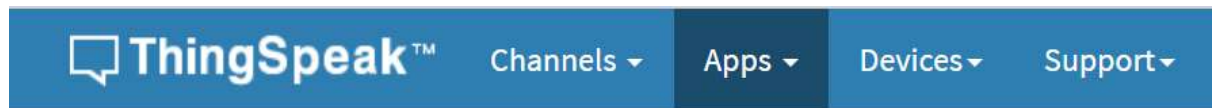
Kennzahlenboxen für Dashboard in ThingSpeak erstellen

Als zusätzliche Analyse der Lautstärke und Personenzahl können die jeweiligen Durchschnitte mittels MATLAB berechnet werden.

1. Über den grünen Button “MATLAB Analysis” kann dies umgesetzt werden.

The screenshot shows the ThingSpeak web interface for a channel named "HTW_Ampel_Analyse". At the top, there's a navigation bar with "Channels", "Apps", "Devices", and "Support" menus, along with links for "Commercial Use", "How to Buy", and a "MK" logo. Below the navigation bar, the channel name "HTW_Ampel_Analyse" is displayed, followed by its ID "1480622", author "mwa0000023508449", and access level "Public". A row of social media sharing buttons (Watch, Tweet, Gefällt mir, Weiterempfehlen) is visible. Below this, a set of tabs includes "Private View", "Public View", "Channel Settings", "Sharing", "API Keys", and "Data Import / Export". Under the "Public View" tab, there are three buttons: "Add Visualizations", "Add Widgets", and "Export recent data". To the right of these are two prominent green buttons: "MATLAB Analysis" and "MATLAB Visualization". At the bottom, the "Channel Stats" section shows "Created: 24 days ago", "Last entry: 6 days ago", and "Entries: 27".

2. Eine neue Seite öffnet sich. Unter "Templates" soll "Custom (no starter code)" ausgewählt sein, da der nötige Code bereitgestellt wird. Anschließend auf den Button "Create" klicken.



[Apps](#) / [MATLAB Analysis](#) / [New](#)

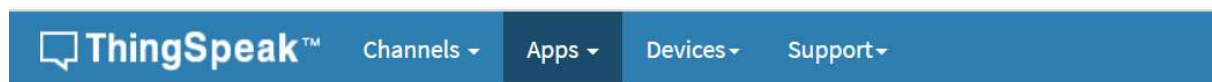
Templates:

- ☒ Custom (no starter code)
- ☐ Get data from a private channel
- ☐ Get data from a public channel
- ☐ Get data from a webpage

Examples: Sample code to analyze and transform data

- ☐ Calculate and display average humidity
- ☐ Calculate wind chill and update channel

3. Im folgenden Fenster kann nun ein Name festgelegt und der folgende Code in den Bereich für den MATLAB Code eingefügt werden. Die berechneten Durchschnitte werden dabei in den Channel für die Analyse geschrieben. Die Inline-Kommentare innerhalb des Codes beschreiben die einzelnen Abschnitte und geben an, wo möglicherweise Parameter angepasst werden müssen (z.B. der für die Durchschnittsberechnung betrachtete Zeitrahmen).



[Apps](#) / [MATLAB Analysis](#) / [Durchschnitte berechnen](#) / [Edit](#)

Name

MATLAB Code

```
1 % Liest die Personenzahl aus dem ThingSpeak channel und schreibt den Durchschnitt der letzten
2 % 45 Minuten in einen anderen ThingSpeak channel.
```

```

% Liest die Personenzahl aus dem ThingSpeak channel und schreibt den
Durchschnitt der letzten
% 45 Minuten in einen anderen ThingSpeak channel.

%% Der Read-Channel enthält die gemessene Lautstärke. %%
% Channel ID von dem Daten gelesen werden
readChannelID1 = 1488846;
% Personenanzahl ID
lautstaerkeID = 1;
% Personenanzahl ID
personenAnzahlID = 2;

% Channel Read API Key
% Falls der Channel private ist, muss der read API Key zwischen ''
eingesetzt werden
readAPIKey1 = '9UDZBOUNECDEI1SP';

% Holt die Lautstärke der letzten 45 Minuten vom Channel, der die
Lautstärkedaten speichert
% Durch Austausch der 45 nach 'NumMinutes' kann der betrachtete
Zeitraum geändert werden
lautstaerke =
thingSpeakRead(readChannelID1,'Fields',lautstaerkeID,'NumMinutes',45,
'ReadKey',readAPIKey1);

% Berechnet die durchschnittliche Lautstärke
avgLautstaerke = mean(lautstaerke);
display(avgLautstaerke,'Durchschnittliche Lautstärke');

% Holt die Personenzahl der letzten 45 Minuten vom Channel, der die
Personendaten speichert.
personenAnzahl =
thingSpeakRead(readChannelID1,'Fields',personenAnzahlID,'NumMinutes',
45,'ReadKey',readAPIKey1);

% Berechnet die durchschnittliche Personenzahl
avgPersonenAnzahl = mean(personenAnzahl);
display(avgPersonenAnzahl,'Durchschnittliche Personenzahl');

% Um die Durchschnittswerte in den Channel schreiben zu können, muss
die Channel
% ID ersetzt werden
writeChannelID = 1480622;
% Write API Key zwischen '' ersetzen:
writeAPIKey = '4XJMUAL8FB6DJTW4';

% Schreibt die Daten in den Channel für die Analyse
thingSpeakWrite(writeChannelID, [avgLautstaerke,
avgPersonenAnzahl], 'WriteKey',writeAPIKey);

```

4. Mit “Save and Run” wird der Code ausgeführt und im Output-Feld können die Ergebnisse auf ihre Richtigkeit kontrolliert werden. Rote Meldungen deuten dabei auf einen Fehler hin. Die Beschreibungen helfen bei dem Error-Handling. Im unteren Beispiel fehlt beispielsweise eine Variable oder sie ist nicht korrekt.

Save and Run

Save

Output

Durchschnittliche Lautstärke =

NaN

Unrecognized function or variable 'readAPIKey2'.

5. Um diese Analyse nicht nur einmalig durchzuführen und die aktuellen Durchschnittswerte für das Dashboard bereitstellen zu können, bietet ThingSpeak Reacts an. Diese können einen MATLAB-Code ausführen wenn ein vordefiniertes Ereignis eintritt. Für die Anwendung ist es sinnvoll, wenn neue Messwerte im Channel für die Daten geladen werden den Durchschnitt neu zu berechnen. Die Erstellung eines Reacts erfolgt mit einem Klick auf das grüne “React” im unteren Teil der Seite.

Schedule Actions

☒ Notify me via email if this MATLAB Analysis fails when triggered by TimeControl or React.

 TimeControl

Schedule your saved code to run at a specified time

 React

Name	Test Frequency
------	----------------

6. Der Name des Reacts kann frei gewählt werden. Der “Condition Type” ist “Numeric”, da es sich um eine Eingabe von Zahlen handelt und die Prüfung muss auf “On Data Insertion” stehen, sodass bei neu in den Channel geschriebenen Daten immer geprüft wird, ob das React ausgeführt werden soll. Der Channel und das zu testende Feld sind je nach Setup einzufügen. Da keine Lautstärke oder Personenzahl in dem betrachteten Szenario von 1000 erreicht werden kann, wird das React bei jeder neuen Eingabe von Daten ausgeführt und ermöglicht die Bereitstellung aktueller Daten. Mit dem Klick auf “Save React” wird alles gespeichert.

[Apps](#) / [React](#) / [React 1](#) / [Edit](#)
React Name

Condition Type
 ▾

Test Frequency
 ▾

Condition

If channel

 ▾

field

 ▾

 ▾

Action
 ▾

Code to execute

 ▾

Options

- ☐ Run action only the first time the condition is met
☒ Run action each time condition is met

7. Jetzt können die berechneten Durchschnitte über das Feld “Add Widgets” hinzugefügt werden.

The screenshot shows the ThingSpeak interface for a channel named 'HTW_Ampel_Analyse'. The channel ID is 1480622, the author is mwa0000023508449, and the access is public. There are tabs for Private View, Public View, Channel Settings, Sharing, API Keys, and Data Import / Export. Below the tabs are three buttons: '+ Add Visualizations', '+ Add Widgets', and 'Export recent data'.

8. Zur Darstellung eines Durchschnitts ist das “Numeric Display” in der Mitte geeignet.

The screenshot shows a dialog box titled 'Click on a widget to add it to the Channel'. It contains three widget options: 'Gauge' (a circular gauge), 'Numeric Display' (a digital display showing '1516.12' with a unit 'DS' below it), and 'Lamp Indicator' (a circular indicator). At the bottom right of the dialog are 'Next' and 'Cancel' buttons.

9. Im nächsten Fenster wird der Name festgelegt, das Feld je nach Setup, das Intervall für das Update und die Einheit. Das Beispiel ist für die Lautstärke ausgefüllt und da nur ganze Zahlen von dem Sensor gesendet werden und keine genaueren Werte nötig sind, ist als Datentyp “Integer” auszuwählen. Mit Klick auf “Create” wird das Widget erstellt und erscheint im Dashboard.

Configure widget parameters

Name: Durchschnitt Lautstärke

Field: Field 1

Update Interval: 15 second(s)

Units: Dezibel

Data Type: ☒ Integer ☐ Decimal 1 (# of places)

Create Cancel

10. Für den zweiten Durchschnittswert muss der Prozess ab 7. noch einmal durchlaufen werden.

11. Um die Rohdaten detaillierter auswerten zu können, besteht die Möglichkeit mittels des "Data Import/ Export"-Tabs die Daten im CSV-Format herunterzuladen.

ThingSpeak™ Channels Apps Devices Support

DatenLaermampel

Channel ID: 1488846

Author: mwa0000023508449

Access: Private

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

+ Add Visualizations + Add Widgets Export recent data

ThingSpeak Channel als Quellen für Dashboard einsetzen

Neben der gemessenen Personenzahl könnte auch die Außentemperatur einen Einfluss auf die Lautstärke haben. Diese könnte zum einen über einen weiteren Sensor erfasst werden oder von anderen ThingSpeak-Nutzern mit verwendet werden. Innerhalb von ThingSpeak besteht eine Community, die ihre Daten auch öffentlich zur Verfügung stellt. Ein Channel stellt Temperaturdaten unter anderen für Berlin bereit.

1. Für die Verwendung der Temperatur Berlins muss im Channel für die Analyse eine neue "MATLAB Visualization" hinzugefügt werden. Ein Liniendiagramm ist dafür sinnvoll.

The screenshot shows the ThingSpeak channel page for 'HTW_Ampel_Analyse'. The channel ID is 1480622, the author is mwa0000023508449, and the access is public. The page has tabs for Private View, Public View, Channel Settings, Sharing, API Keys, and Data Import / Export. Below the tabs are buttons for Add Visualizations, Add Widgets, and Export recent data. On the right, there are buttons for MATLAB Analysis and MATLAB Visualization. The Channel Stats section shows it was created 24 days ago, last entry 6 days ago, and has 27 entries.

2. Eine neue Seite öffnet sich. Unter "Templates" soll "Custom (no starter code)" ausgewählt sein, da der nötige Code bereitgestellt wird. Anschließend auf den Button "Create" klicken.

The screenshot shows the 'New' page for MATLAB Visualizations in ThingSpeak. The breadcrumb trail is Apps / MATLAB Visualizations / New. The page title is 'New'.

Templates:

- ☒ Custom (no starter code)
- ☐ Create a filled area 2-D plot
- ☐ Create a 2-D line plot
- ☐ Create 2-D line plots with y-axes on both left and right side
- ☐ Create a correlated data plot
- ☐ Create a discrete sequence data plot

Examples: Sample code to visualize data

- ☐ Use a histogram to understand variation in data
- ☐ Visualize directional data with compass plot

3. Im nächsten Schritt kann das Diagramm benannt werden als "Temperatur in Berlin in °C" und der folgende Code in das Feld für den MATLAB Code kopiert werden. Keine Anpassungen müssen an dem Code vorgenommen werden. Da die Daten von einem privaten Nutzer bereitgestellt werden, kann es sein, dass die Qualität sowie die Zuverlässigkeit der Datenbereitstellung in regelmäßigen Abständen überprüft werden sollten.

[Apps](#) / [MATLAB Visualizations](#) / [Temperatur in Berlin in °C](#) / [Edit](#)

Name

MATLAB Code

```
1 % Temperatur von einem öffentlichen Channel einlesen, um die Abhängigkeit der Lautstärke
2 % von der Temperatur beobachten zu können
3 % ChannelID
4 readChannelID = [143789];
5 % Berliner Temperatur Feld
6 fieldID1 = [5];
```

```
% Temperatur von einem öffentlichen Channel einlesen, um die
Abhängigkeit der Lautstärke
```

```
% von der Temperatur beobachten zu können
```

```
% ChannelID
```

```
readChannelID = [143789];
```

```
% Berliner Temperatur-Feld
```

```
fieldID1 = [5];
```

```
% API key zum Lesen von privaten Channels muss für die Methode unten
definiert sein, aber
```

```
% nicht eingefügt werden
```

```
readAPIKey = ''
```

```
%% Lesen der Daten %%
```

```
% Alle 15 Minuten werden die Daten von dem Sensor bereitgestellt
```

```
% 'NumPoints' (Anzahl der anzuzeigenden Datenpunkte) ist
```

```
% auf 12 eingestellt, was den letzten 3 Stunden entspricht
```

```
[data, time] = thingSpeakRead(readChannelID, 'Field', fieldID1,
```

```
'NumPoints', 24, 'ReadKey', readAPIKey);
```

```
%% Visualisieren der Daten %%
```

```
plot(time, data)
```

```
ylabel ('Temperatur in °C')
```

```
xlabel ('Uhrzeit');
```

4. Mit "Save and Run" wird der Code ausgeführt und im Output-Feld können die Ergebnisse auf ihre Richtigkeit kontrolliert werden. Das Diagramm wird