

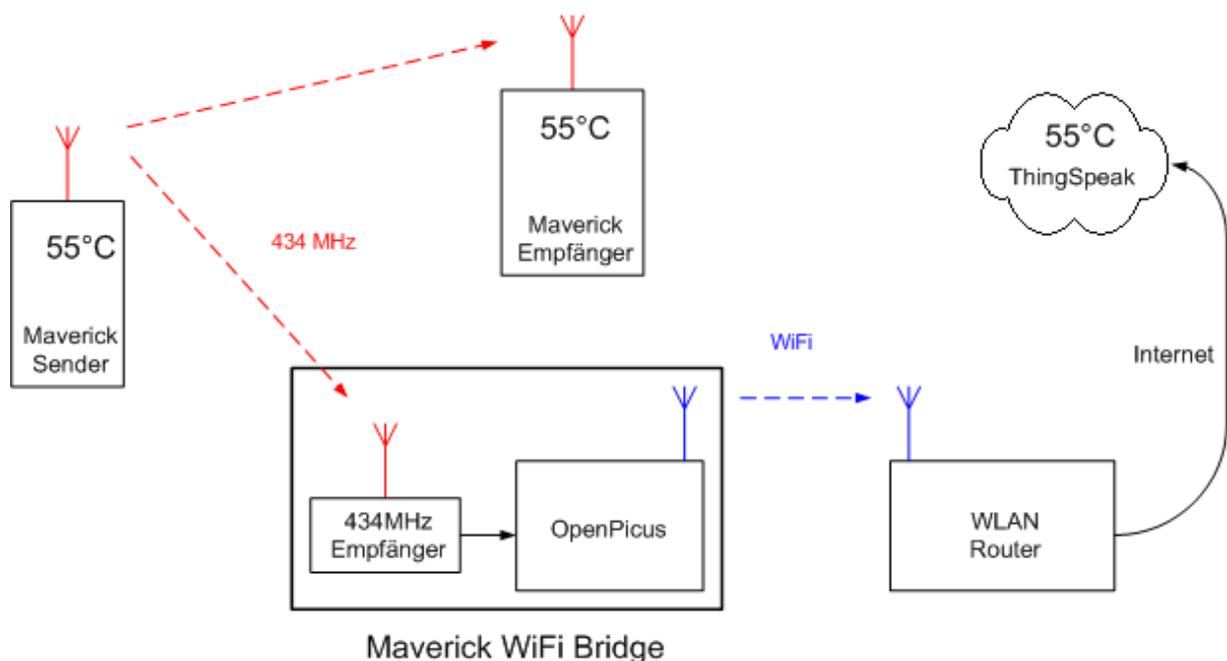
Maverick ET-732 WiFi Bridge – v1.03

1. Einleitung

Diese Anleitung beschreibt wie man die Funkdaten des BBQ-Thermometers Maverick ET-732 dekodieren und zum Cloudservice www.thingspeak.com senden kann. Die Abfrage der Temperaturdaten ist dadurch live und weltweit möglich sobald man über Internet und einen Webbrowser verfügt (Windows, MacOS, Android, iOS, ...).

Das Maverick Thermometer muss dabei nicht modifiziert werden und kann somit ganz normal benutzt werden. Lediglich das Funkprotokoll wird „angezapft“.

Bitte zunächst die KOMPLETTE Anleitung durchlesen!!! Thingspeak ist nicht unbedingt notwendig (Kapitel 7).



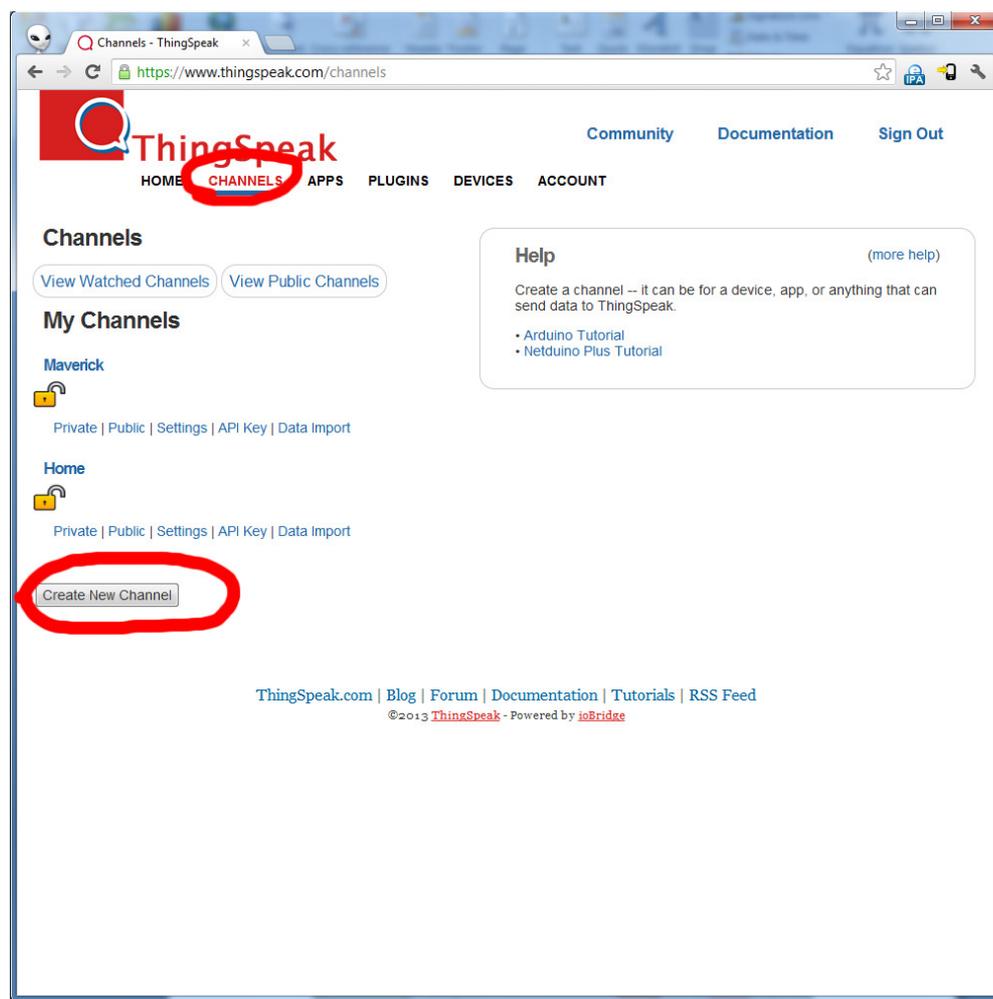
2. Bauteilliste

Folgende Bauteile werden benötigt:

- OpenPicus WiFi Board ca. 60€
<http://store.openpicus.com/openpicus/prodotti.aspx?cprod=015365>
<http://tigel.de/product/2309>
- Aurel MID RX 3V, 434MHz Receiver ca 12€
<http://de.farnell.com/aurel/rx-mid-3v/modul-receiver-434mhz-am-3v/dp/1699472>

3. Einrichtung Thingspeak-Account

Zunächst muss ein Account bei www.thingspeak.com eingerichtet werden.
Danach folgender Ablauf:



Channel 5779 - ThingSpeak x
https://www.thingSpeak.com/channels/5779#channelsettings

ThingSpeak Community Documentation Sign Out
HOME CHANNELS APPS PLUGINS DEVICES ACCOUNT

Channels » Channel 5779 Watch Tweet 0 Gefällt mir +1 0

Private View Public View **Channel Settings** API Keys Data Import

Percentage Complete

Name

Description

Tags

Latitude

Longitude

Elevation

Make Public?

URL

Video ID YouTube Vimeo

Field 1 remove field

Field 2 add field

Field 3 add field

Field 4 add field

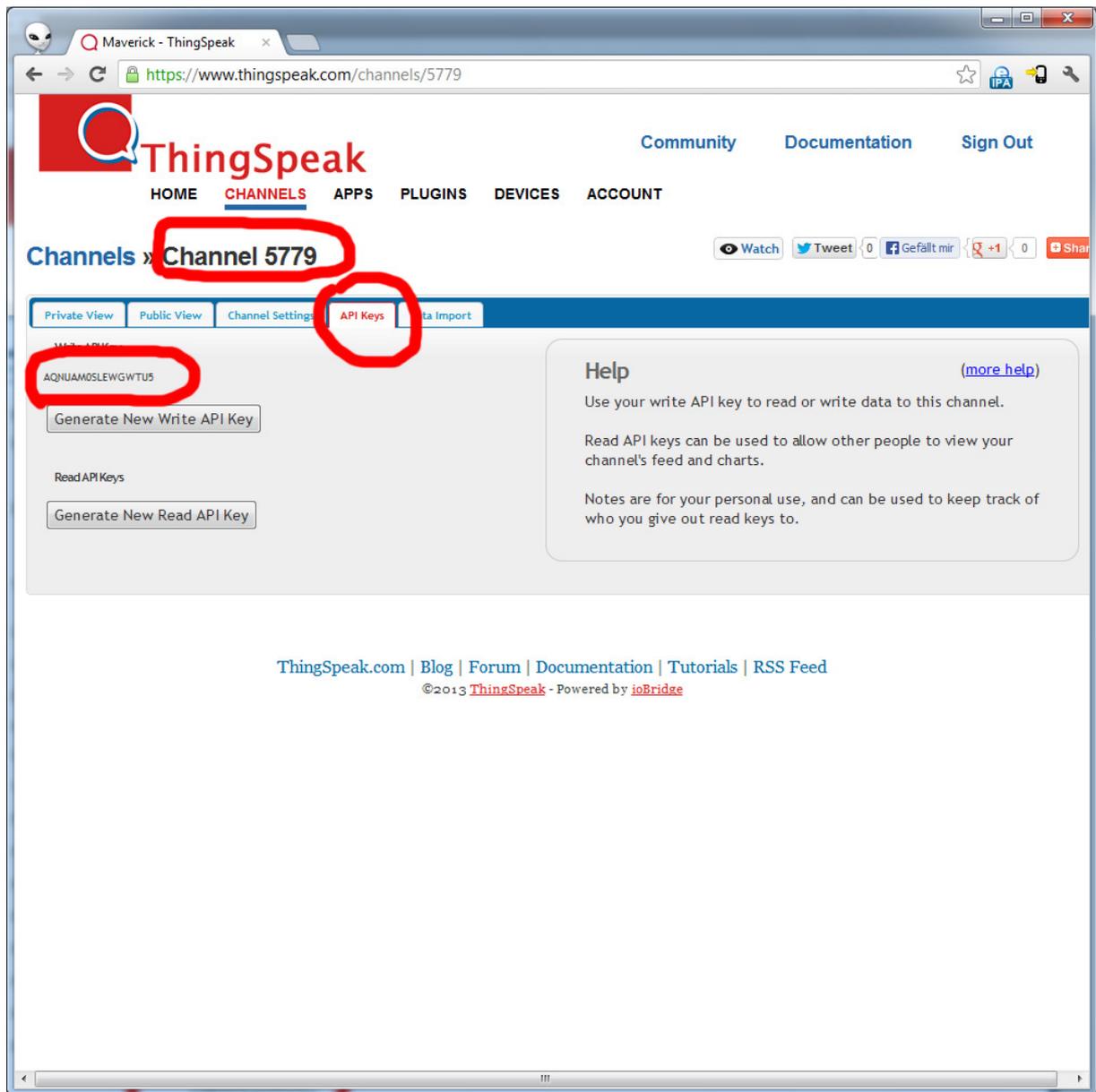
Field 5 add field

Field 6 add field

Field 7 add field

Field 8 add field

Help
Make this channel public to allow anyone to view its feed and charts without using API keys.
By using the URL field, you can provide a link to more information for your public channel, such as a blog post or webpage about the channel.
Add a video to the public view of your channel by providing a YouTube or Vimeo ID, which is found in the URL of your video.
Add up to 8 fields that can be tracked. A field must be added before it can store data.
Clicking on the "Clear Channel" button will delete ALL feed data associated with this channel, but will leave the channel's info intact.
Completeness will help your channel appear higher in the list of public channels.



Die Channel Nummer und den Write-API-Key notieren!!!

Nicht auf einen der „Generate New xxx API Key“ Buttons drücken.

4. Installation der OpenPicus Entwicklungsumgebung

Für die Installation der Entwicklungsumgebung wird Windows XP oder Windows 7 benötigt (Windows 8 funktioniert vermutlich auch, habe ich allerdings nicht getestet).

Zunächst muss der Microchip Compiler heruntergeladen und installiert werden:

http://ww1.microchip.com/downloads/en/DeviceDoc/MPLABC30Combo_v3_24Lite.zip

Danach muss die IDE installiert werden:

<https://code.google.com/p/openpicus/downloads/detail?name=FlyPort%20IDE-2.3-Setup.exe&can=2&q=>

Nun noch den virtuellen COM-Port Treiber installieren:

<http://www.ftdichip.com/Drivers/VCP.htm>

Nun auf die blaue „Wizard“ Fahne klicken und so oft NEXT klicken bis man diesen Dialog erhält:

The screenshot shows the 'TCP/IP Configuration' dialog box with the 'Wireless Configuration' tab selected. The title bar reads 'TCP/IP Configuration' and the window contains the 'openPICUS' logo and the slogan 'Let's fly!'. The 'Default SSID Name' field is set to 'meineSSID'. The 'Default Network Type' is set to 'Infrastructure' and the 'Default PS Poll' is set to 'Disabled'. The 'Scan settings' section shows 'Default Scan Type' set to 'Passive Scan'. Under 'Scan Channels', channels ch.1 through ch.11 are checked, while ch.12 is unchecked. At the bottom, the device name 'Maverick_WiFi' is shown, along with '< Back', 'Next >', and 'Cancel' buttons.

Statt „meineSSID“ die SSID des eigenen WiFi Netzwerks eintragen.

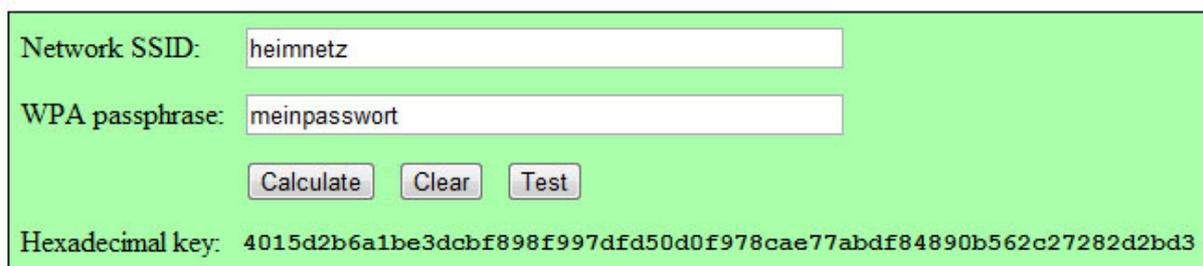
Nochmal NEXT klicken um diesen Dialog zu erhalten:

The screenshot shows the 'TCP/IP Configuration' dialog box with the 'Wireless Security' tab selected. The title bar reads 'TCP/IP Configuration' and the window contains the 'openPICUS' logo and the slogan '...Fly safe... fasten your seat belt!'. The 'Default Security' dropdown is set to 'WPA or WPA2 auto select'. There is an unchecked checkbox for 'Flyport will calculate Key from Passphrase'. The 'Key Data' section includes a note: 'This must be created in conjunction with the Pass Phrase and SSID.' The 'PSK key' field contains the hexadecimal string '683fa30e3faa8bb5a076edd58165f50d4daf4d1ced33949a88d18fcd0496ad4'. The 'PSK Phrase' field is empty. At the bottom, the device name 'Maverick_WiFi' is shown, along with '< Back', 'Next >', and 'Cancel' buttons.

Unter PSK key muss man nun den Key des eigenen WiFi Netzwerks eingeben. Aber ACHTUNG!!! Das ist nicht das WiFi Passwort dass man sonst verwendet. Das Passwort ist eigentlich die PSK Phrase und aus der PSK Phrase wird der PSK key mit einem recht aufwendigem Algorithmus berechnet. Das ist eine ziemliche Herausforderung für den kleinen Prozessor auf dem Flyport, darum greifen wir dem Prozessor etwas unter die Arme.

Auf dieser Homepage kann man die eigenen SSID und das Passwort eingeben und sich den PSK key berechnen lassen: <http://jorisvr.nl/wpapsk.html>

Ein kleines Bsp für die Berechnung:



Network SSID: heimnetz

WPA passphrase: meinpasswort

Calculate Clear Test

Hexadecimal key: 4015d2b6a1be3dcbf898f997dfd50d0f978cae77abdf84890b562c27282d2bd3

Den „Hexadecimal key“ als PSK key eintragen und so oft NEXT drücken bis der Wizard abgeschlossen ist.

Den Wizzard muss man nur einaml ausführen, es sei denn die SSID oder das Passwort ändert sich irgendwann.

Nun auf den grünen „Compile Project“ Pfeil drücken.

Das dauert nun etwas... warten bis im Fenster ganz unten die Erfolgsmeldung kommt:

```
----- End Project Compilation -----  
-----  
----- Started at 30.01.2013 23:05:38  
----- Completed at 30.01.2013 23:05:52  
----- Duration (hh.mm.ss) 00.00.13  
-----  
-----
```

Nun das Flyport-Modul in das USB-Nest stecken und über USB mit dem PC verbinden. Einige Zeit warten bis der Treiber installiert wurde.



Im IDE Programm auf „Download Firmware“ rechts oben klicken.

Unter Serial Port die richtige COM Schnittstelle auswählen. Ist man sich nicht sicher kann man auch einfach alle verfügbaren COM Schnittstellen ausprobieren.

Danach auf „Download“ klicken um die Firmware dauerhaft in das Flyport Modul zu programmieren. Ist der Vorgang beendet kann das USB Kabel abgesteckt, das IDE Programm geschlossen und das Flyport Modul aus dem USB-Nest entnommen werden.

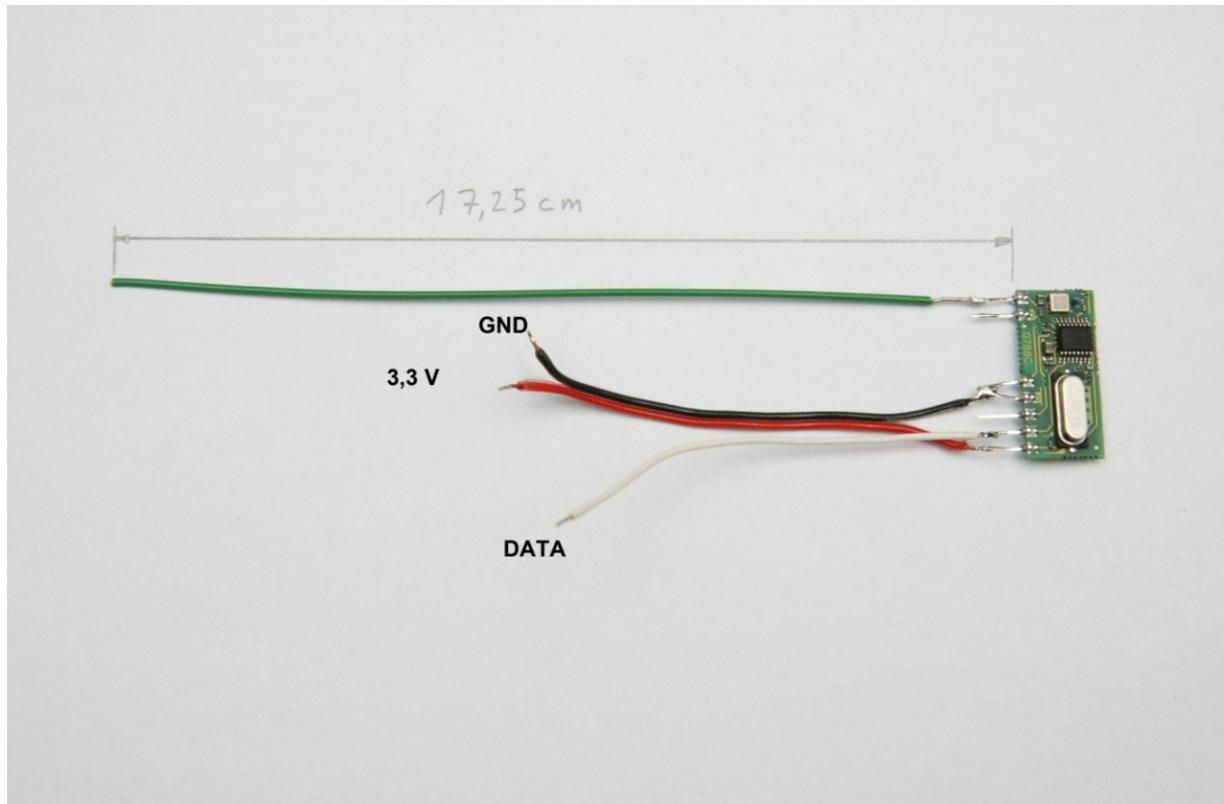
6. Löten

Nun muss nur noch der Empfänger mit dem Flyport verbunden werden.

So sehen die 2 Teile aus:

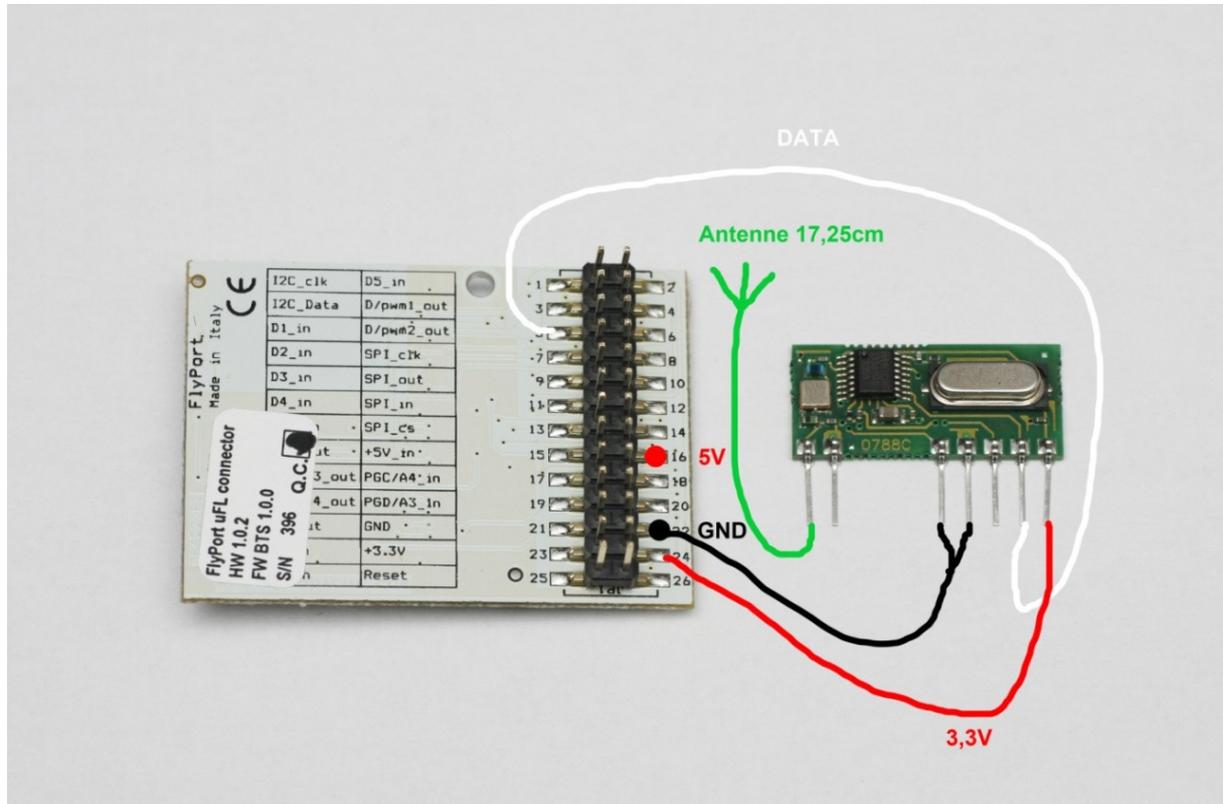


Zunächst folgende Drähte an den Empfänger anlöten:
(der grüne Draht ist die Antenne und muss ab der Printplattenkante 17,25cm lang sein)

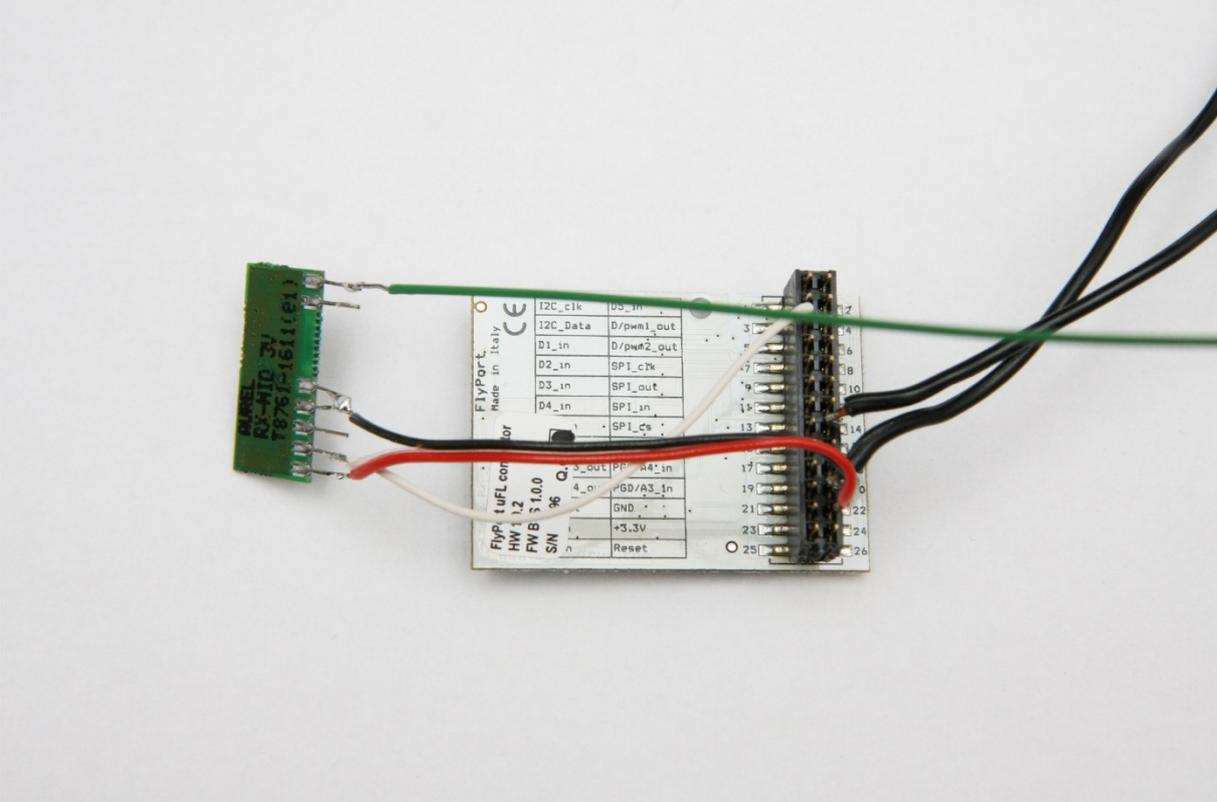
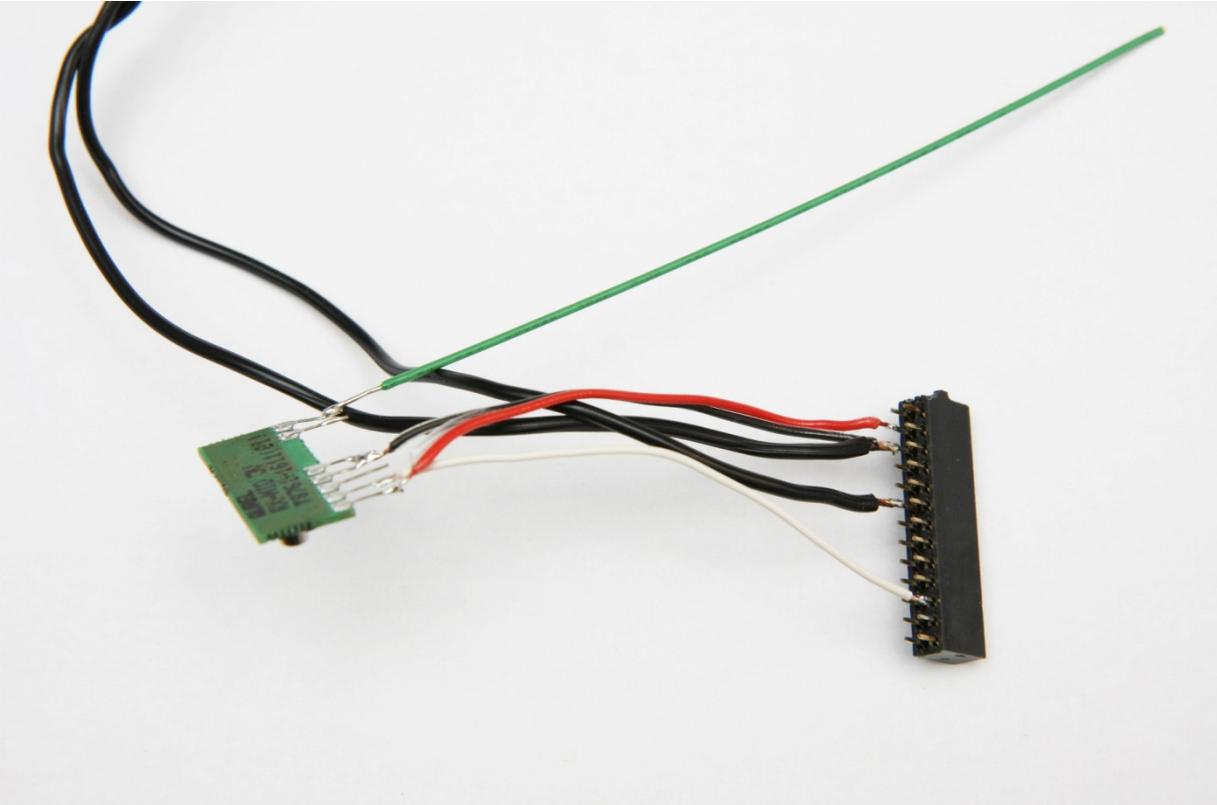


Und so müssen die beiden Module verbunden werden:

An den schwarzen und roten Punkt wird die 5V Spannungsversorgung angeschlossen. Ein einfaches Netzteil mit ca. 1A reicht vollkommen aus.

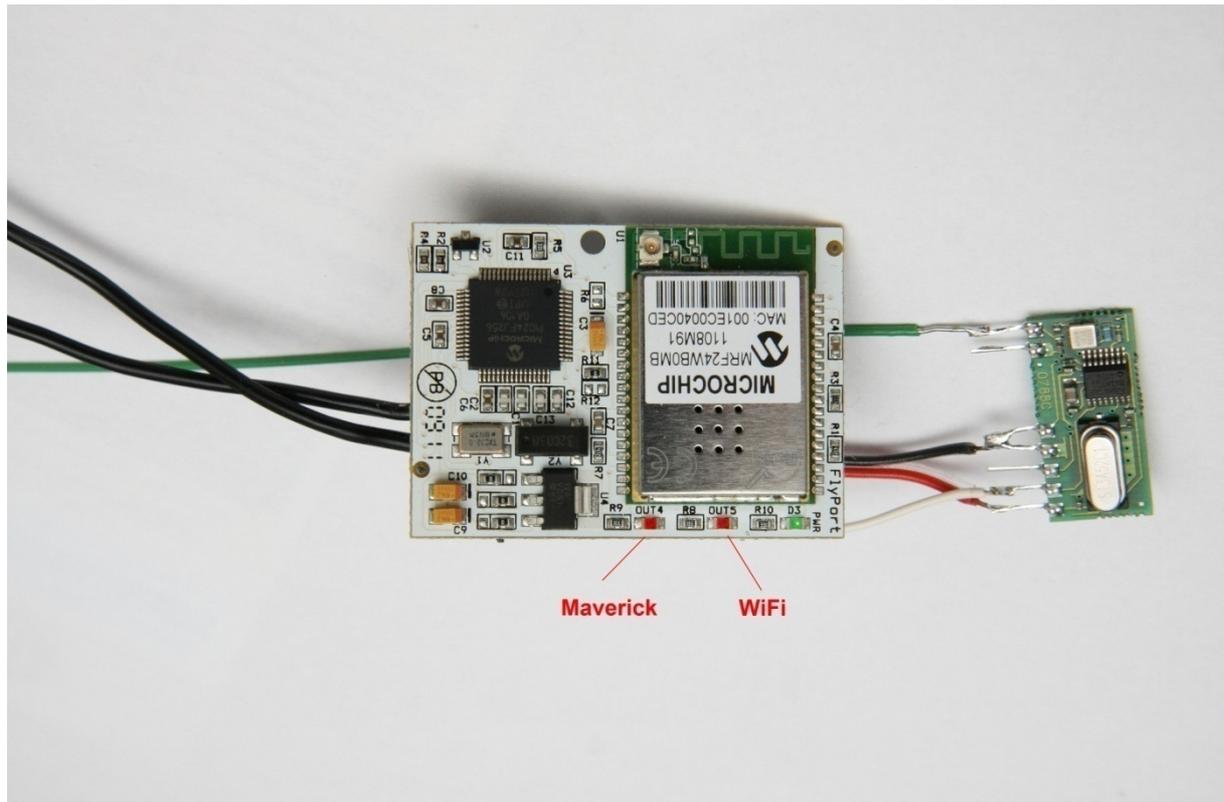


Ich habe beide nicht direkt miteinander verbunden. Das Empfangs-Modul habe ich mit einer Buchsenleiste verbunden und das Flyport-Modul wird dann in diese Buchsenleiste gesteckt:



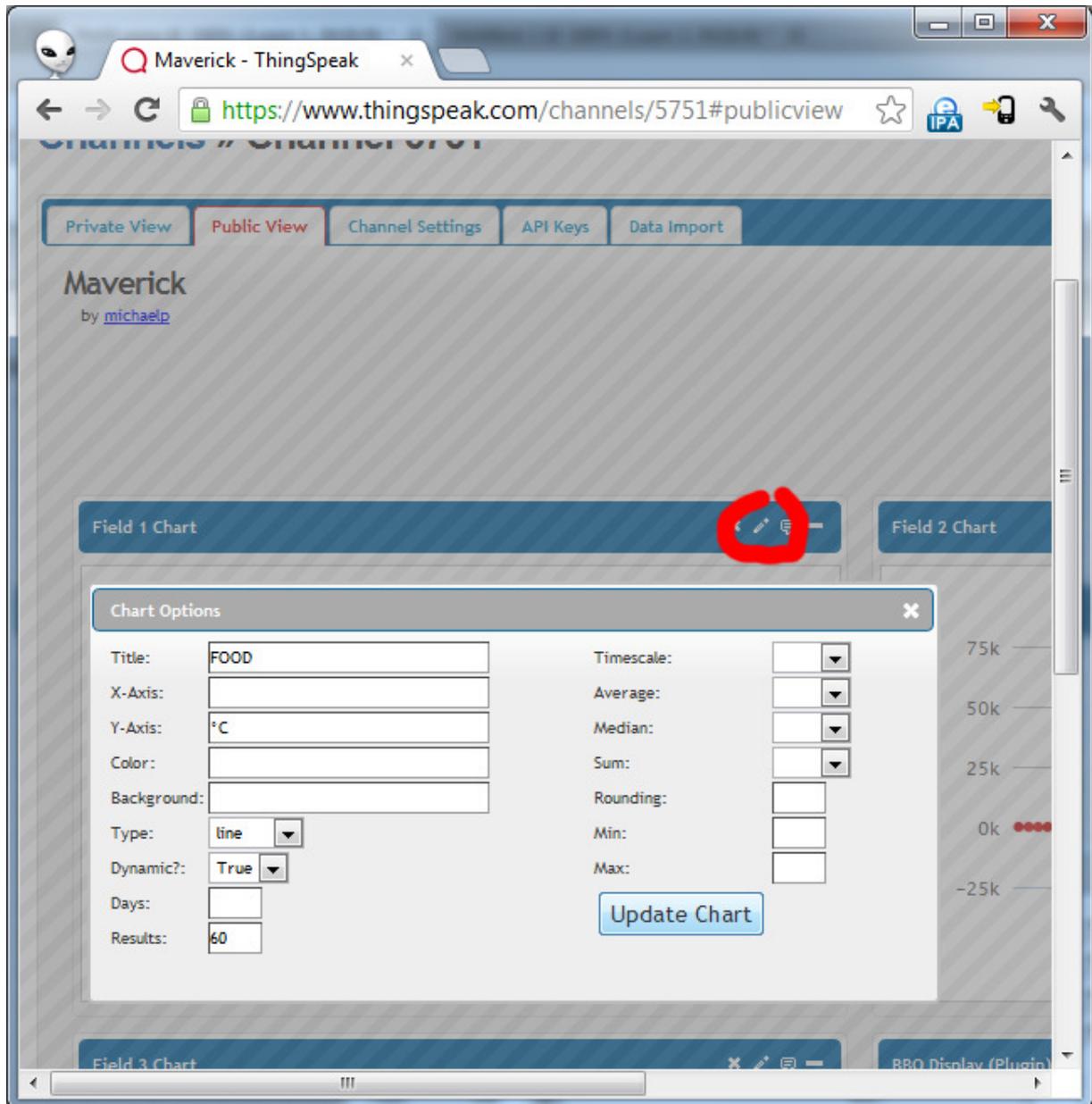
Nun nur noch die 5V anlegen, und zumindest den Maverick Sender einschalten. Den Maverick Empfänger kann man einschalten, muss man aber nicht.

Die LEDs auf dem Flyport-Modul leuchten einerseits wenn gerade ein Datenpaket vom Maverick Sender empfangen wurden, bzw. Wenn ein Datenpaket über WiFi zu www.thingspeak.com gesendet wird.



7. Erweiterte Darstellungsmöglichkeiten

Thingspeak bietet einige nette Features die recht hilfreich sind.
Jedes Diagramm-Fenster kann beliebig platziert werden.



Durch klicken auf den kleinen Bleistift kann jedes Diagramm konfiguriert werden:

Ein „Titel“ sowohl Achsenbeschriftungen können eingegeben werden. Unter „Results“ kann definieren wieviele Messpunkte dargestellt werden. Lässt man „Results“ leer, so kann man mit „Timescale“ definieren dass nur alle 10, 15, 20, usw. Minuten ein Messpunkt angezeigt wird.

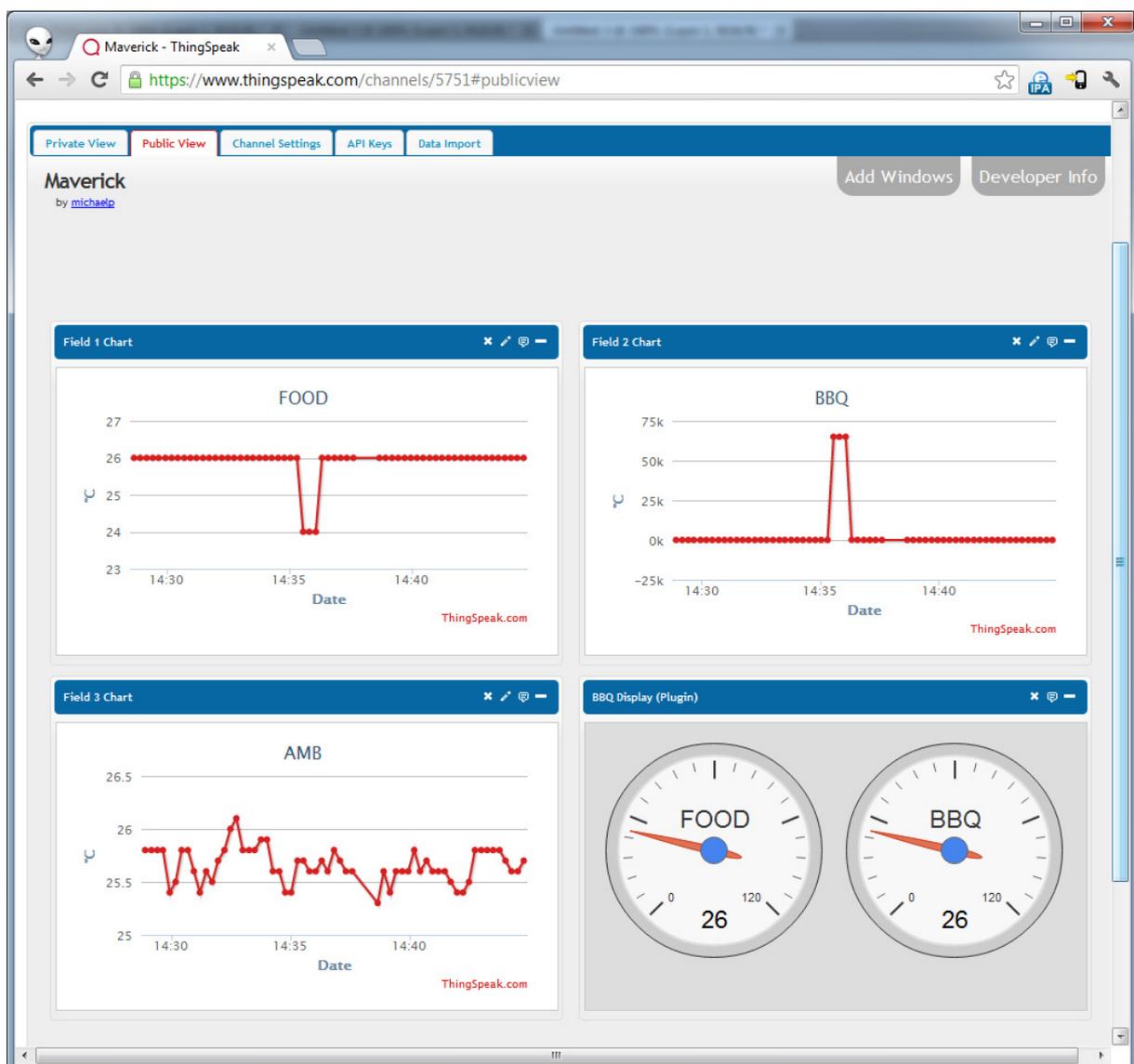
Sind die Messdaten zu unruhig könnte man die Anzeige gemittelt darstellen lassen.

zB.: Average=10, würde über 10 Minuten alle Messwerte mitteln und nur diesen Mittelwert anzeigen.

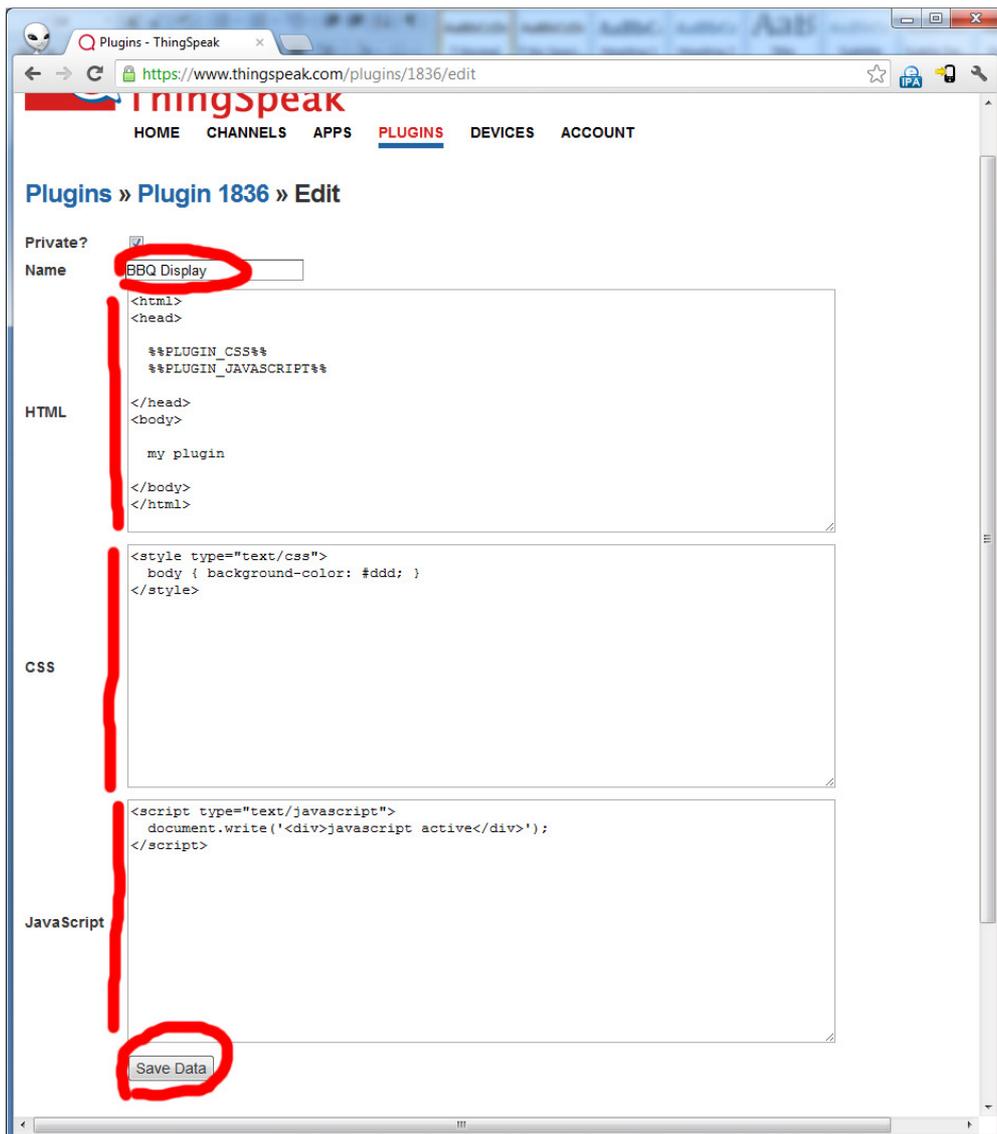
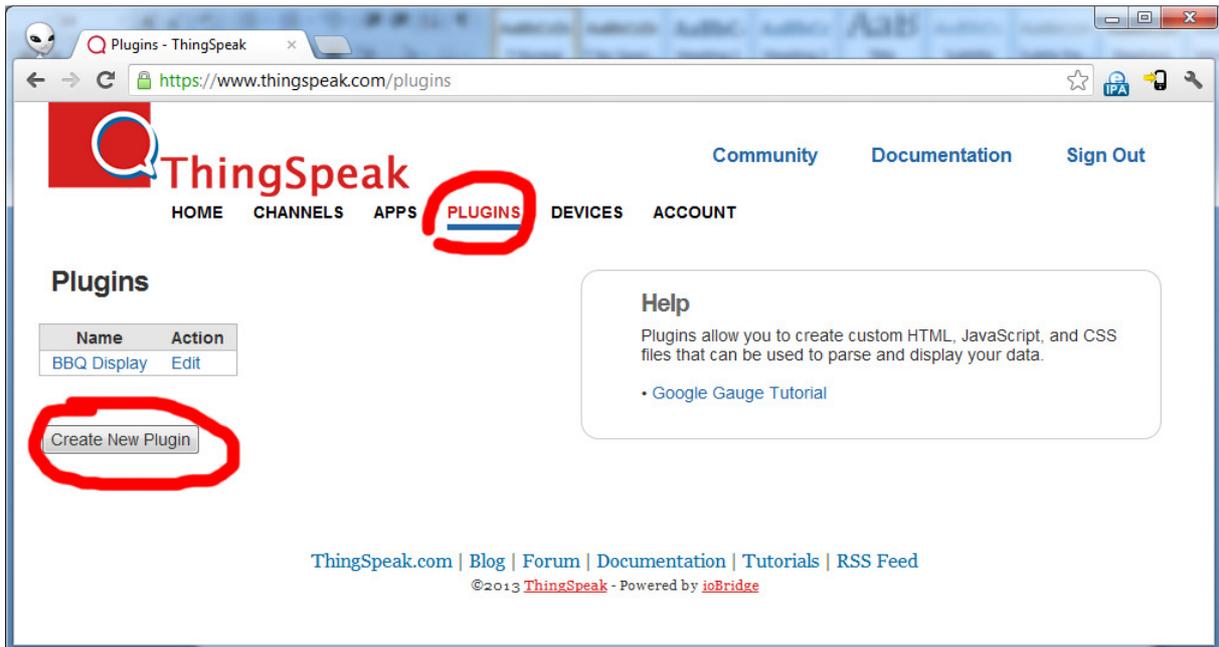
Probieren geht hier über studieren 😊

Ein Wichtiges Feature ist das Min/Max-Feld. Da ja die Erkennung das Maverick-Signals nicht immer einwandfrei ist, könnte man hier zB Min=0 und Max=200 einstellen. Im Diagramm würde man somit keine Ausreißer unter 0 bzw. über 200°C sehen.

Ein sehr nettes Feature ist die Einbindung sogenannter Plugins. Damit läßt sich eine analoge Anzeige wie in diesem Screenshot rechts unten realisieren:



Dazu auf der Plugin-Seite auf „Create New Plugin“ klicken.



Unter „Name“ einen sprechenden Namen zB „BBQ Display“ eintragen.

Den Inhalt vom HTML Feld löschen und durch diesen ersetzen:

```
<html>
<head>

<title>Google Gauge - ThingSpeak</title>

%%PLUGIN_CSS%%
%%PLUGIN_JAVASCRIPT%%

</head>

<body>
  <div id='chart_div'></div>
</body>
</html>
```

Den Inhalt vom CSS Feld löschen und durch diesen ersetzen:

```
<style type="text/css">
  body { background-color: #ddd; }
</style>
```

Den Inhalt vom JavaScript Feld löschen und durch diesen ersetzen:

```
<script type='text/javascript' src='https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js'></script>
<script type='text/javascript' src='https://www.google.com/jsapi'></script>
<script type='text/javascript'>
```

```
    var chart;
    var charts;
    var data;
    var temp_max=120;    // set maximum display temperature
    var temp_min=0;     // set minimum display temperature
    var green_start=0;  // set start of green display area
    var green_end=0;    // set end of green display area
    var yellow_start=0; // set start of yellow display area
    var yellow_end=0;   // set end of yellow display area
    var red_start=0;    // set start of red display area
    var red_end=0;      // set end of red display area
```

```
    google.load('visualization', '1', {packages:['gauge']});
    google.setOnLoadCallback(initChart);
```

```
    function displayData(point1, point2) {
```

```
        // display only FOOD
        //data.setValue(0, 0, 'FOOD');
        //data.setValue(0, 1, point1);
```

```
        // display only BBQ
        //data.setValue(0, 0, 'BBQ');
```

```

//data.setValue(0, 1, point2);

// display FOOD and BBQ
var data = google.visualization.arrayToDataTable([
    ['Label', 'Value'],
    ['FOOD', point1],
    ['BBQ', point2]
]);

    chart.draw(data, options);
}

function loadData() {

    // variable for the data point
    var p1,p2;

    $.getJSON('https://api.thingspeak.com/channels/XXXX/feed/last.json?callback=?',
function(data) {

    // get the data point
    p1 = data.field1;
    p2 = data.field2;

    p1 = Math.round(p1);
    p2 = Math.round(p2);

    if (p1>temp_max)
        p1=temp_max;
    if (p2>temp_max)
        p2=temp_max;

    displayData(p1, p2);

    });

}

function initChart() {

    data = new google.visualization.DataTable();
    data.addColumn('string', 'Label');
    data.addColumn('number', 'Value');
    data.addRows(1);

    chart = new google.visualization.Gauge(document.getElementById('chart_div'));

    options = {width: 440, height: 220, minorTicks: 5, min:temp_min,
        max:temp_max,
        greenFrom: green_start, greenTo: green_end,
        yellowFrom: yellow_start, yellowTo: yellow_end,
        redFrom: red_start, redTo: red_end,};

    loadData();

    setInterval('loadData()', 1000);
}

```

```
    }  
</script>
```

Die **XXXX** müssen durch die eigene Channel Nummer ersetzt werden!!!!

Die Channel Nummer haben wir zuvor schon notiert.

Der **rot markierte Teil** im JavaScript kann an die eigenen Bedürfnisse angepaßt werden. Die Werten gelten aber für beide Anzeigen, also für FOOD und BBQ!

```
var temp_max=120;    // set maximum display temperature  
var temp_min=0;     // set minimum display temperature  
var green_start=0;  // set start of green display area  
var green_end=0;    // set end of green display area  
var yellow_start=0; // set start of yellow display area  
var yellow_end=0;   // set end of yellow display area  
var red_start=0;    // set start of red display area  
var red_end=0;      // set end of red display area
```

temp_max=??? das Maximum der analogen Anzeige

temp_min=??? das Minimum der analogen Anzeige

Mit den green/yellow/red_start/end-Werten kann man einen Bereich im Anzeigegerät grün, gelb bzw. rot darstellen lassen.

zB könnte man den Bereich von 85-95°C grün darstellen lassen.

```
green_start=85;
```

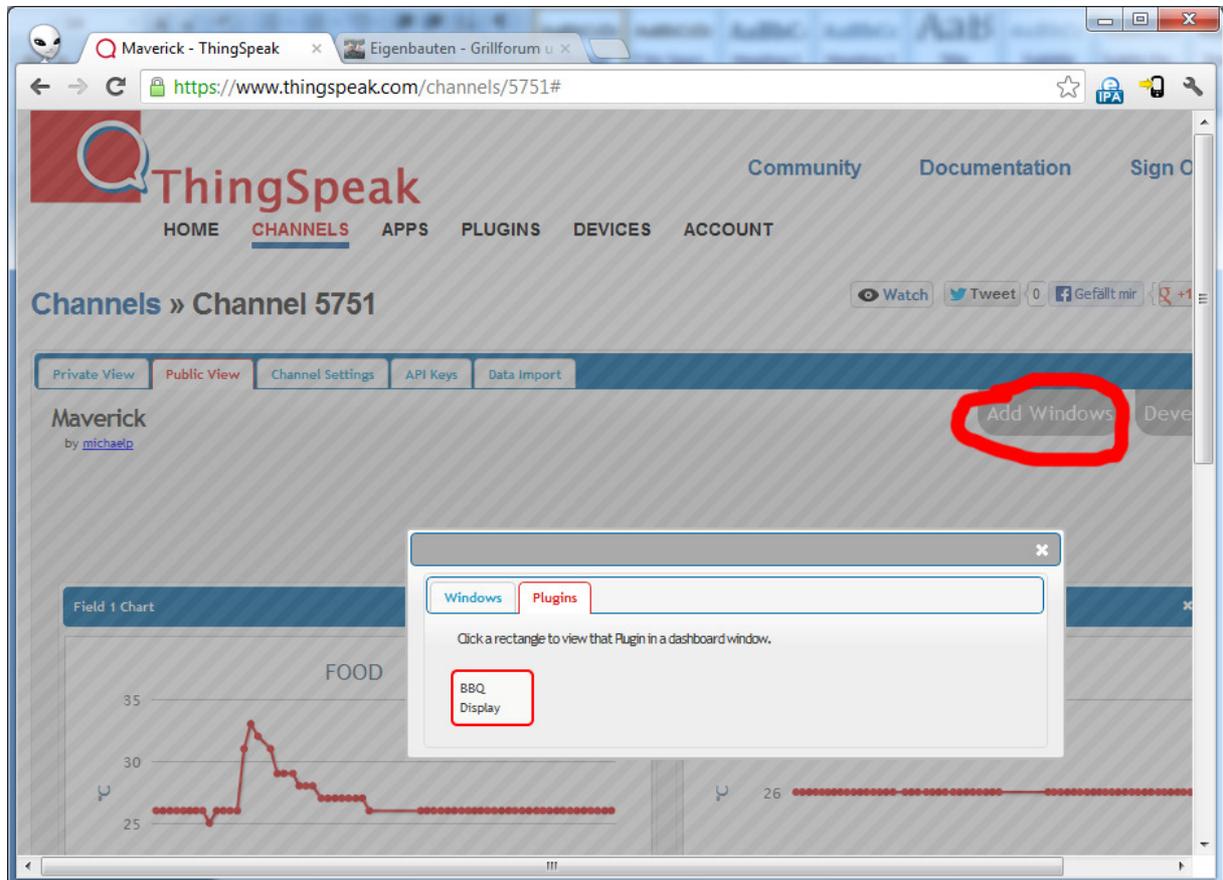
```
green_end=95;
```

Möchte man allerdings unterschiedlich konfigurierte Anzeigen für FOOD und BBQ haben (zB.: FOOD sollte den roten Bereich von 56-120°C und BBQ den roten Bereich von 110-120°C haben) dann müssen 2 Plugins erstellt werden.

Der blaue Bereich im Java-Script muss angepaßt werden, je nachdem was im Plugin angezeigt werden soll. Ich denke das ist selbsterklärend.

Auf jeden Fall muss das Plugin mit „Save“ abgespeichert werden.

Nun muss das Plugin auch angezeigt werden. In der Channel-Ansicht auf „Add Windows“ klicken und dort unter Plugins das zuvor erstellte Plugin auswählen.

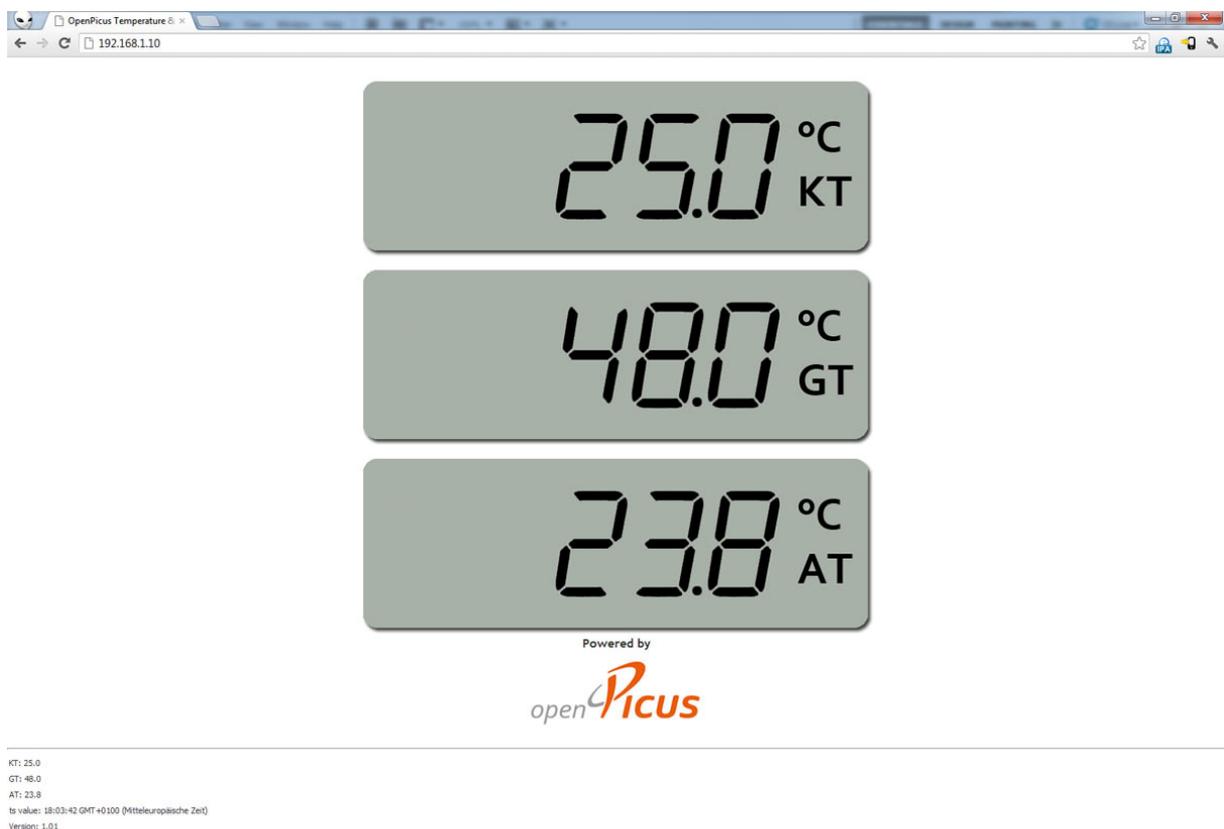


8. Änderungen v1.01

Auf dem Flyport läuft ein kleiner Webserver. Die Homepage des Flyports zeigt nun die aktuellen Temperaturdaten mit autorefresh an. Diese interne Homepage ist natürlich nur im eigenen Netzwerk sichtbar. Befindet man sich nicht im eigenen Netzwerk so muss man mit Diensten wie zB DynDNS (nicht kostenlos) die IP des Flyports in die „Außenwelt transportieren“. Das ist aber nicht notwendig da es ja eben den Gratisdienst ThingSpeak gibt. Zu Hause hat man also die Wahl zwischen der internen Homepage oder ThingSpeak. Außerhalb des Heimnetzwerks kann man immer auf ThingSpeak zurückgreifen. Um die interne Homepage des Flyports anzuzeigen muss man die IP-Adresse des Flyports im Browser eingeben. Um die IP-Adresse des Flyports herauszufinden muss man den eigenen Router befragen. Da dies von Router zu Router unterschiedlich ist kann ich dazu keine weiteren Tips geben.

Falls die Temperaturüberwachung

Interne Homepage (Außentemperatursensor angeschlossen):



OpenPicus Temperature & x

192.168.1.10

25.0 °C
KT

48.0 °C
GT

23.8 °C
AT

Powered by
openPicus

KT: 25.0
GT: 48.0
AT: 23.8
ts value: 18:03:42 GMT+0100 (Mitteleuropäische Zeit)
Version: 1.01

COSM und interne Homepage über Smartphone : (ohne Außentempersensur)



Falls also die Kontrolle außerhalb des eigenen Netzwerks nicht notwendig ist (also COSM), man die netten Diagramme von COSM nicht benötigt und man in der Lage ist über den Router die IP-Adresse des Flyports herauszufinden so ist ein COSM Zugang nicht unbedingt notwendig. In diesem Fall einfach keine Feed-ID und/oder keinen Key im Sourcecode eintragen.

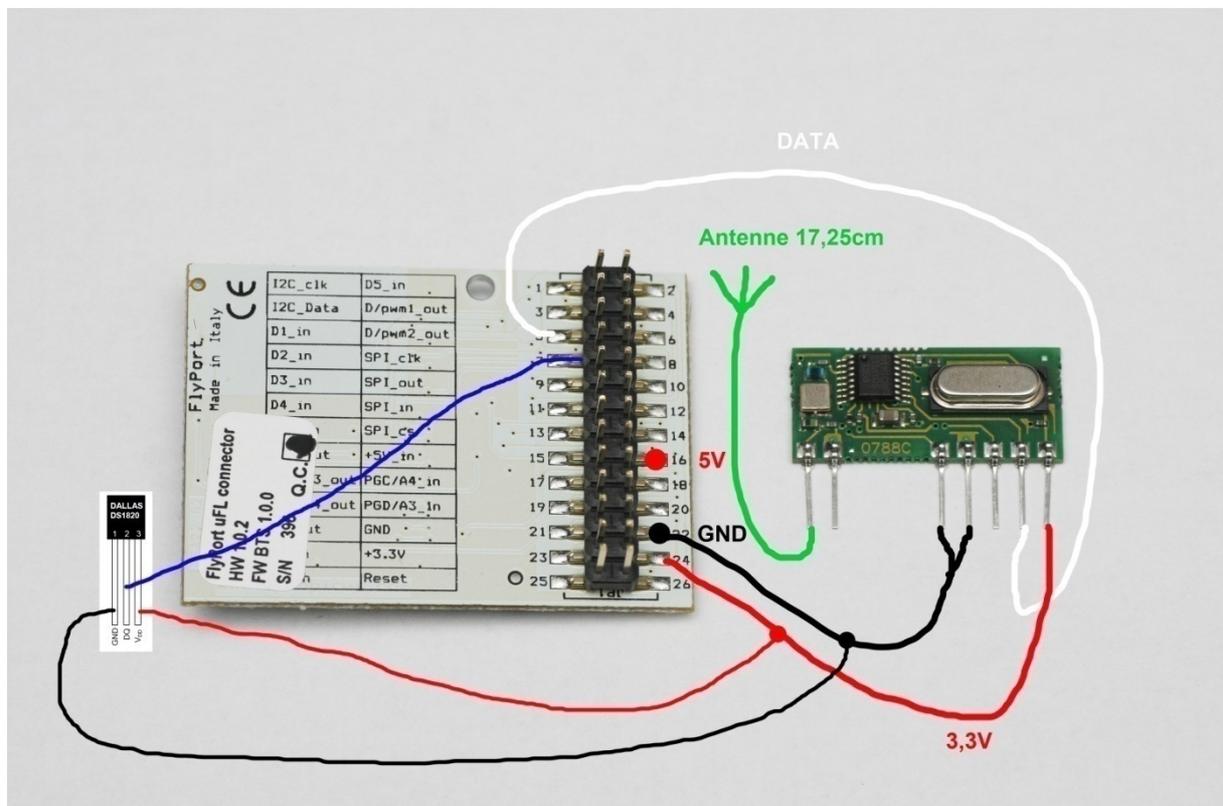
Optional kann nun ein digitaler Temperatursensor DS1820 bzw. DS18S20 (der Nachfolger des obsoleten DS1820) als Außentempersensur angeschlossen werden. Man könnte auch nur den DS18S20 anschließen und keinen Maverick betreiben um zB das SousVide Becken zu überwachen. Dazu würde sich zB dieser DS18S20 im Edelstahlgehäuse empfehlen. ACHTUNG!!! Der ist nicht wasserdicht, also nur das Edelstahlgehäuse ins Wasser tauchen.

<http://www.conrad.de/ce/de/product/184037/Temperatur-Sensor-Hygrosens-Temp-Fuehler-DS-1820-BT-Kabel-2-m-55-125-C-Gehaeuseart-Edelstahlgehaeuse>

Schließt man keinen DS18S20 an, so ändert sich nichts bei COSM. Sobald ein DS18S20 angeschlossen wird, erscheint ein dritter Stream Namens „AT“ im eigenen Feed.

Bitte unbedingt einen DS18S20 auswählen der sich für 3,3V Versorgungspg. eignet. zB hier: http://www.produktinfo.conrad.com/datenblaetter/175000-199999/176168-da-01-en-DS_1820_1820S.pdf

Anschlußschema Außentemperatursensor:



9. Änderungen v1.02

Als experimentelles Feature kann nun ein Lüfter an den Pin4 des Flyports (Beschriftung D/pwm1_out) angeschlossen werden. Es handelt sich dabei um eine simple 2-Punkt-Regelung. Über die Eingabefelder *Temperature* und *Hysteresis* können die Parameter eingegeben und schließlich über den Button *Set Fan Values* bestätigt werden.

Ist die BBQ-Temperatur höher als $Temperature + Hysteresis$ wird der Pin4 auf LOW gezogen.

Fällt die BBQ-Temperatur wieder unter *Temperature-Hysteresis* wird Pin4 wieder auf HIGH gesetzt.

Da immer auf die Antwort des ThingSpeak-Servers gewartet werden muss, gibt es eine leichte Verzögerung zwischen der Anzeige des Maverick-Empfängers und der Anzeige auf der Homepage.

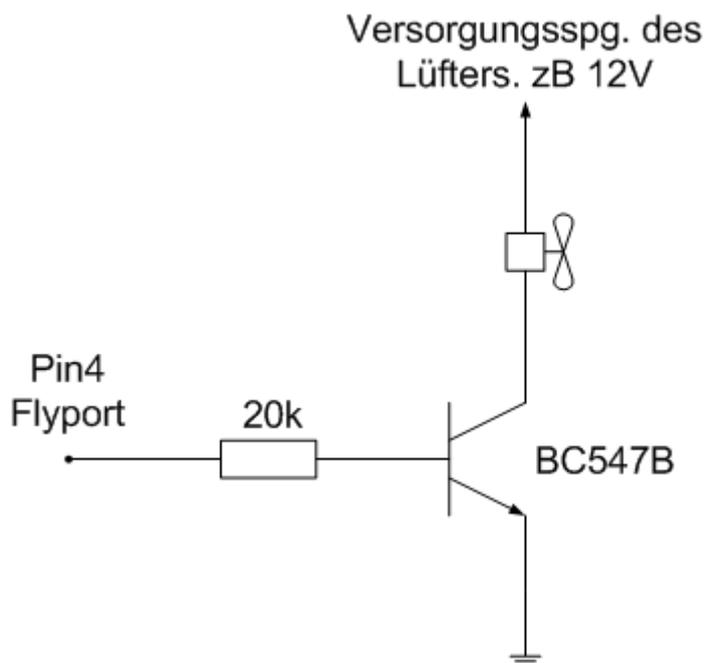
ACHTUNG!!!!

NIEMALS den Lüfter direkt an den Pin4 des Flyport anschließen, es muss immer irgendeine Art von Treiber zwischen Lüfter und Flyport sein. Eine einfache Transistorschaltung mit dem Lüfter im Kollektorzweig bietet sich an.

Der Ausgang (Pin4) liefert 0V (Lüfter aus) bzw 3,3V (Lüfter ein).

Beispielschaltung:

Ein Basiswiderstand von 20k ist ein guter Anfangswert. Je nach Lüfter bzw. Transistor muss dieser allerdings angepaßt werden.



10. Änderungen v1.03

Da mittlerweile aus COSM das bei weitem unübersichtlichere, kompliziertere und unpotentere Xively geworden ist, wird aber dieser Version das CloudService www.thingspeak.com verwendet.

11. Kleingedrucktes

Ich übernehme keinerlei Garantie dass diese Anleitung auch tatsächlich funktioniert. Es könnten zB Probleme bei der Netzwerkkonfiguration auftreten (Firewall etc...) für die ich keinen Support geben kann.

Weiters übernehme ich keine Verantwortung falls beim Zusammenbau oder späterem Betrieb Schaden an diesem oder anderem Equipment entsteht oder die eigene Gesundheit bzw. Die Gesundheit Anderer gefährdet wird.

Nicht falsch verstehen, bei mir funktioniert alles und ich habe schon anderen Personen bei der Inbetriebnahme solcher Schaltungen geholfen. Trotzdem können unerwartete Probleme auftreten für die ich einfach keine Haftung übernehmen kann.