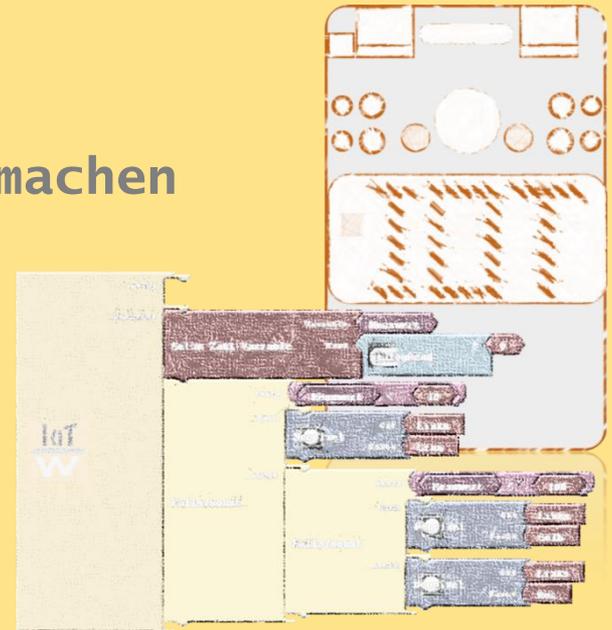


# IoT-Werkstatt

Das Internet der Dinge anfassbar machen



Internet  
of  
Things



Umwelt-Campus  
Birkenfeld

H O C H  
S C H U L E  
T R I E R

Klaus-Uwe Gollmer  
und Guido Burger  
5.2BSEC



IoT  
W

# WLAN

## IoT:Internet Zugang

Der Octopus hat ein eingebaute WLAN-Schnittstelle im 2.4 GHz Bereich. Damit kann sich unser Kit mit jedem Smartphone oder dem Internet verbinden. Wir unterscheiden zwei Betriebsmodi:



### Accesspoint

Der Octopus spannt ein eigenes WLAN auf, bei dem sich andere Teilnehmer als Client anmelden können. (WPA2 verschlüsselt)



#### Netzname (SSID)

Ist beim Client (Smartphone, PC) in der Netzwerk-Auswahlliste sichtbar

**Passwort** bei der Anmeldung (am Client-PC, Smartphone, anzugeben,

- **8 Zeichen** für Anmeldung ohne Passwort einfach Stringelement aus dem Block entfernen

### Internet-Zugang über bestehendes Netz

Der Octopus verbindet sich mit einem bestehenden WLAN-Netz und ermöglicht so ggf. den Internetzugang (WAP2 verschlüsselt)



#### Netzname (SSID)

Name bestehendes WLAN

**Passwort** bestehendes WLAN



# GSM-Modem

Ein externes seriell Modem ermöglicht den mobilen Betrieb.

Voraussetzung: Externes Adafruit-Fona GSM Modem

## Internet-Zugang über Mobilfunk

Der Octopus verbindet sich mit einem Mobilfunknetz und ermöglicht so den mobilen Internetzugang

**APN (Access Point Name)**

Netzzugangspunkt  
Mobilfunkprovider



**User / Passwort**  
optional

## Pinbelegung

Fona	IoT-Kit (Feather-Wing)
GND	GND
Vio	VCC (3V)
RX	GPIO12
TX	GPIO14
Key	GND



# HTTP-Server

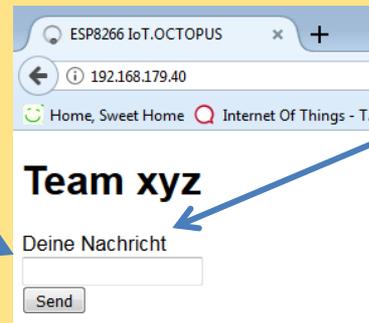
## IoT: HTTP-Protokoll

Mit einem Baustein können wir unseren eigenen Web-Server aufbauen. Die IP-Adresse wird bei der Initialisierung angezeigt. Diese ist nur intern im jeweiligen WLAN-Netz erreichbar. Im eigenen Netz (WLAN-Baustein Accesspoint) besitzt der Server immer die feste Adresse 192.168.4.1

### Ohne Sensoren

#### Deine Nachricht

Wird an den Octopus übertragen und dort ggf. auf einem vorhandenen CharlieWing LED-Matrix Modul angezeigt. (Flaschenpost)



#### Überschrift

Wird als Titel der Homepage angezeigt



### Mit Sensoren

#### Aktuelle Informationen

Messdaten werden auf der Homepage angezeigt und zyklisch aktualisiert

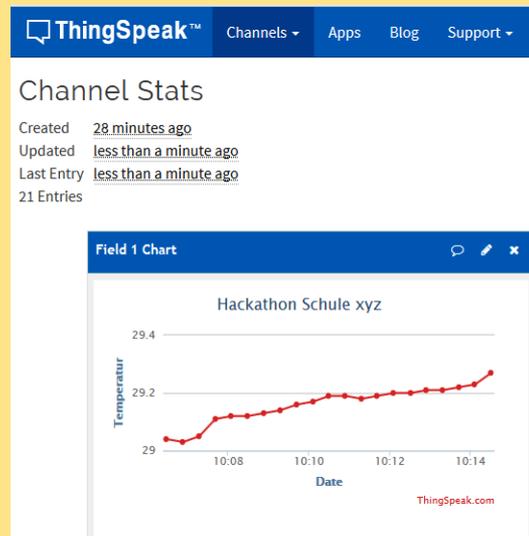


# Thingspeak

Thingspeak.com ist eine Cloud-Anwendung. Dort können Messdaten gespeichert und über eine Web-Schnittstelle visualisiert werden. Voraussetzung ist ein kostenloser Account bei Thingspeak, oder die Nutzung des quelloffenen Servers.

## Schreibzugriff

Messdaten werden per HTTP-Protokoll über die REST-API an den Thingspeak-Server gesendet.



**Alternativ eigener Server:**  
thingspeak.raspi.iot  
thingspeak.umwelt-campus.de

**Send Thingspeak**

Server	api.thingspeak.com
API-Key	2H71DFR7CWVZHCUY
Feld 1	
Feld 2	<b>API-Key</b>
Feld 3	Per Cut&Paste vom
Feld 4	Thingspeak-Server
Feld 5	kopieren
Feld 6	
Feld 7	
Feld 8	

**Ask Thingspeak**

Server	api.thingspeak.com
Kanal	meine Kanalnummer
Feldindex	1

## Lesezugriff

Der aktuelle Messwert für die angegebene Kanalnummer und den angegebenen Feldindex wird beim Thingspeak-Server angefragt

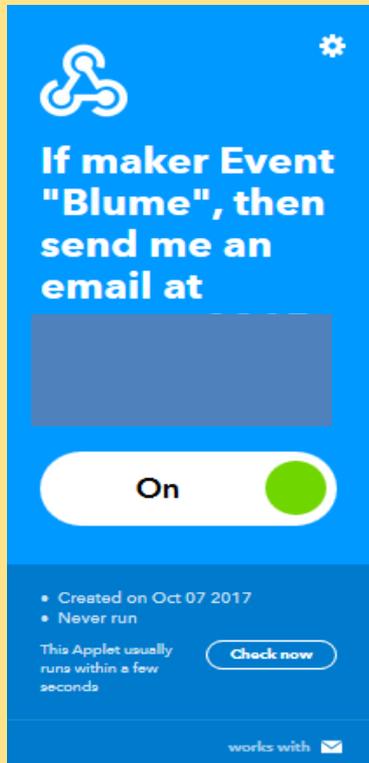


# If This Then That (IFTTT)

IFTTT.com ist eine Cloud-Anwendung, mit der einfache Regeln abgebildet werden können. In unserem Fall können wir ein Ereignis auf dem Webhooks-Kanal triggern. Dieses Ereignis kann dann konfigurierte Aktivitäten auslösen. Voraussetzung ist ein kostenloser Account bei IFTTT.

## Trigger Event auslösen

Der angegeben Event wird gefeuert, Autorisierung durch den Key



### IFTTT-Applet

Achtung: wenn der Trigger mehrmals gefeuert wird, gibt es entsprechend viele Mails. Also im Ardublock aufpassen (nur einen Trigger auslösen).

### Ereignis

Im Maker-Channel des FTTT-Server konfigurieren



### Key

Per Cut&Paste vom IFTTT-Server kopieren



yahoo.com ist eine Cloud-Anwendung, mit der über ein REST-API u. a. die aktuellen Wetterdaten für einen Standort abgefragt werden können.

### Winddaten abfragen

Abfrage der Windgeschwindigkeit per REST-API.  
Yahoo antwortet mit einem XML- oder JSON-Text  
Dieser muss nach dem interessierenden  
Schlüsselwort durchsucht werden. Der Baustein ist  
experimentell, ggf. muss die Auswertung der  
Suchinformation manuell angepasst werden.  
Rückgabe Windgeschwindigkeit getestet.

#### Suchbegriff

API-Kommandos unter  
<https://developer.yahoo.com/weather/>



#### Where on earth identification

Standortangabe über Suchmaschine  
<http://woeid.rosselliot.co.nzl>

Beispiel [https://query.yahooapis.com/v1/public/yql?q=select \\* from weather.forecast where woeid=690631 and u=%27c%27&format=xml](https://query.yahooapis.com/v1/public/yql?q=select * from weather.forecast where woeid=690631 and u=%27c%27&format=xml)

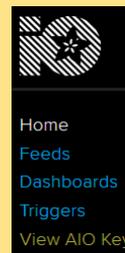
```
- <results>
- <channel>
  <yweather:units distance="km" pressure="mb" speed="km/h" temperature="C"/>
  <title>Yahoo! Weather - Saarbruecken, SL, DE</title>
- <link>
  http://us.rd.yahoo.com/dailynews/rss/weather/Country__Country/*https://weather.yahoo.com/country/state/city-690631/
</link>
<description>Yahoo! Weather for Saarbruecken, SL, DE</description>
<language>en-us</language>
<lastBuildDate>Sat, 07 Oct 2017 12:51 PM CEST</lastBuildDate>
<ttl>60</ttl>
<yweather:location city="Saarbruecken" country="Germany" region=" SL"/>
<yweather:wind chill="45" direction="235" speed="22.53"/>
<yweather:atmosphere humidity="85" pressure="33525.26" rising="0" visibility="25.91"/>
<yweather:astronomy sunrise="7:41 am" sunset="6:57 pm"/>
```



MQTT (Message Queue Telemetry Transport) ist ein IoT-Protokoll zur Sensor-Aktor Kommunikation. Es basiert auf einem publish/subscribe-Modell. Ein hochverfügbarer Broker dient als Vermittler zwischen mobilen Clients.

### Verbindung zum Broker

Einmalig z.B. im setup-Teil. Hier ist ggf. die Autorisierung beim Broker notwendig. Jeder Client benötigt einen Namen.



Freier Broker (ohne Anmeldung)  
broker.mqtt-dashboard.com



User, Passwort. Adafruit: AIO Key

### Nachrichten senden

Sendet eine Nachricht an das angegebene Topic



### Nachrichten empfangen

Abonniert das angegebene Topic beim Broker  
Broker sendet eine Nachricht, wenn sich der Inhalt ändert. Im ArduBlock wird dann automatisch in eine callback-Funktion verzweigt und der dort angegebene Befehlsblock ausgeführt. Die Nachricht steht dort im String MQTT\_Rx\_Payload zur Verfügung.



# The Things Network (TTN)

TTN ist eine Cloud-Infrastruktur zur Kommunikation über LoRaWAN.

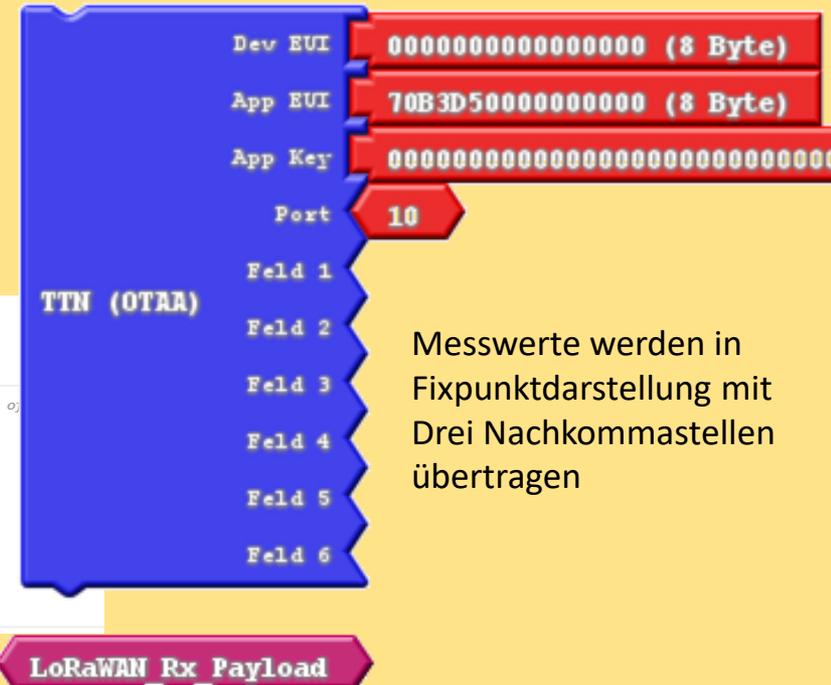
Voraussetzung ist ein Account bei TTN und ein LoRa-Kommunikationsmodul

### OTAA Verbindung

Hier erfolgt die Autorisierung über die Angabe der EUIs. Details finden sich auf der TTN-Webseite. Die Messwerte im Feld 1-6 werden als 24-bit Integer Übertragen. Auf TTN-Seite ist ein Decoder nötig.

```
decoder converter validator encoder

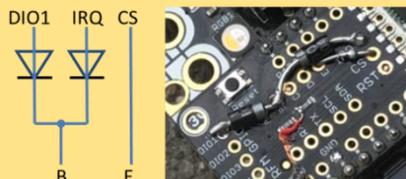
1 function Decoder(bytes, port) { // Decode an uplink message from a buffer (array) of bytes to an object of
2 var decoded = {}; //
3 decoded.port = port;
4 decoded.humidity = 0; decoded.temp = 0; decoded.pressure=0;
5 if (port === 10) { // Port selection
6 decoded.humidity = (bytes[0] << 16 | bytes[1]<<8 | bytes[2] | (bytes[0] & 0x80 ? 0xFF<<24 : 0))/1000;
7 decoded.temp = (bytes[3] << 16 | bytes[4]<<8 | bytes[5] | (bytes[3] & 0x80 ? 0xFF<<24 : 0))/1000;
8 decoded.pressure = (bytes[6] << 16 | bytes[7]<<8 | bytes[8] | (bytes[6] & 0x80 ? 0xFF<<24 : 0))/1000;
9 //decoded.feld4 = (parseFloat((bytes[6] << 8 | bytes[7]).toFixed(2)/10.);
10 }
11 return decoded;
12 }
```



Messwerte werden in Fixpunktdarstellung mit Drei Nachkommastellen übertragen

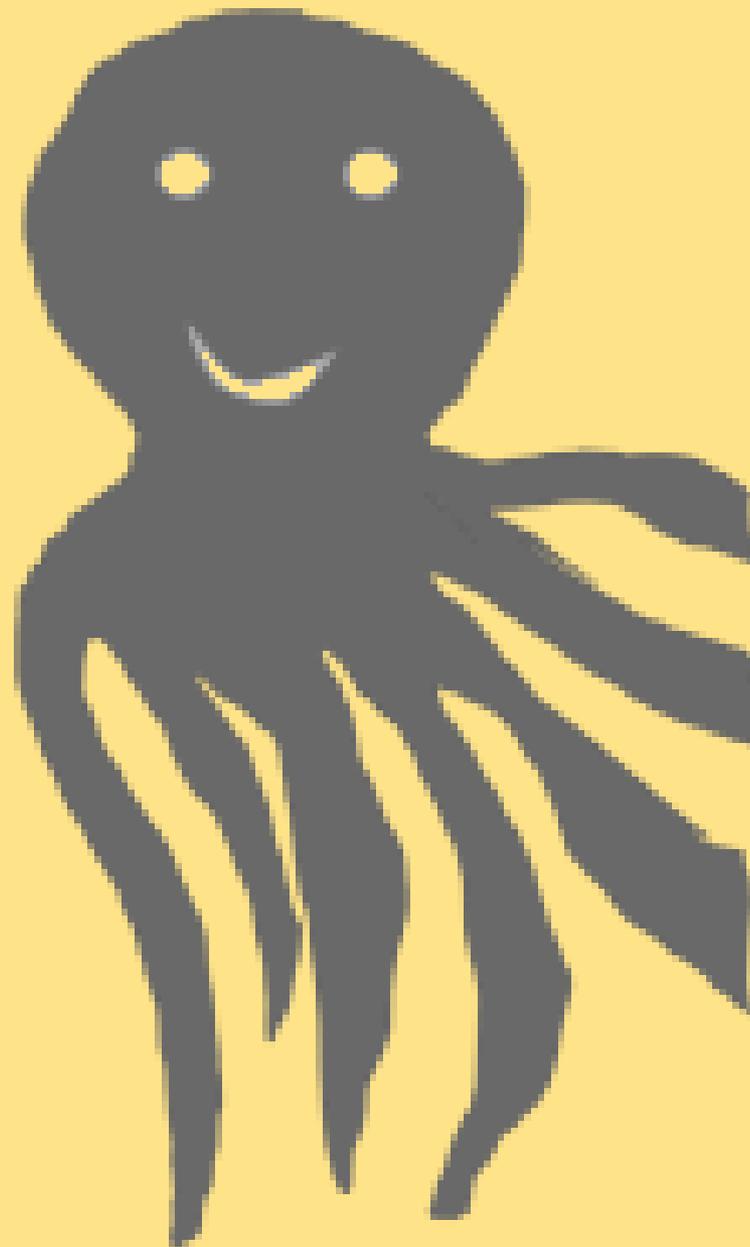
Ein ggf. im RX-Slot empfangenes Payload kann ausgewertet werden

### Adafruit LoRaWing



<https://www.adafruit.com/product/3231>





Klaus-Uwe Gollmer  
und Guido Burger

