

Bauanleitung für den "PaxCounter"

Stand: 18.9.2018 Sebastian Müller / sbamueller.de

Was ist das? Über das Projekt

Nein, wir zählen nicht die Anzahl der Waffenstillstände im Autoverkehr. **"Pax" steht hier für "Personenanzahl": Wir versuchen ohne Einsatz von Kameras abzuschätzen, wieviele Personen ungefähr zu einem bestimmten Zeitpunkt an einem Ort sind**, an dem wir unseren "PaxCounter" aufstellen.

Wem nützt das?

Zum Beispiel Nahverkehrsunternehmen, Messeveranstaltern, Wahlkampfstandveranstaltern, Demos die damit stets ein ganz aktuelles Bild bekommen könnte an welchen Haltestellen (auch Bus) oder Orten der Bedarf gerade hoch ist. Das ist besonders interessant, wenn man diese Zahlen über längere Zeiträume erfasst.

Die einzige andere Art diese Zahlen zu erfahren sind Verkehrszählungen, die sehr personalintensiv und damit teuer sind. Sie könnten daher nie über längere Zeit durchgeführt werden.



Wie funktioniert das?

Sehr viele Menschen (wenn auch natürlich nicht alle) haben heute ein Smartphone in der Tasche. Solange diese nicht im "Flugmodus" sind, versuchen sie in der Regel alle paar Sekunden nachzusehen ob es rundherum WLAN-Stationen gibt oder sich zu einem bekannten BluetoothGerät zu verbinden (wie der Freisprecheinrichtung im Auto zum Beispiel).

Dabei senden sie kleine Datenpakete. Diese fängt ein Pax-Counter auf. Anhand der zu gleicher Zeit funkenden, verschiedenen Geräte kann abgeschätzt werden, wieviele Smartphones, oder Notebookträger gerade in der Nähe (ca. 10 m) sind.

Ist das der "Große Bruder", ist das Überwachung?

Nein. Zwar haben die meisten Handys eine weltweit eindeutige Adresse die schon immer übermittelt (sogenannte IMEI¹) wird an das Handy Netz um das Geräte zu identifizieren. Geräte die darüber hinaus WLAN oder Bluetooth nutzen haben auch eine MAC Adresse². Allerdings geben iPhones und Android die MAC Adresse nur in einer zufälligen Version heraus, diese kann von jedermann aufgefangen werden kann. Es ist kein Rückschluß auf den Träger des Handys möglich, weil diese Adressen im PaxCounter mit einem eigenen Zufallswert verbunden und vorhast wird und auch nie länger als ein paar Minuten im PaxCounter gespeichert werden, noch werden diese in irgendeiner Form weitergeleitet. Der PaxCounter mißt jede Minute die Anzahl voneinander unterscheidbaren Adressen, fertig. Er überträgt nur die Summe der Messung.

¹ https://de.wikipedia.org/wiki/International_Mobile_Equipment_Identity

² <https://de.wikipedia.org/wiki/MAC-Adresse>

Ist das datenschutzkonform?

Da personenbezogenen Daten weder erhoben, noch dauerhaft gespeichert oder übertragen werden, sondern nur anonyme Summen, ja!

Personen ohne oder mit abgeschaltetem Smartphone werden NICHT gezählt.

Ist das gefährlich?

Obwohl im Moment trotz lang anhaltenden Forschung keine Hinweise auf negative Einflüsse dieser Funksignale vorliegen, "hört" der PaxCounter nur in eine ohnehin stattfindenden Kommunikation. Der Elektrosmog wird also um Null Prozent erhöht.

Über diese Bauanleitung

Wir versuchen in dieser Bauanleitung alle drei Aspekte eines Internet of Things Gerätes abzubilden: Das eigentliche Gerät mit der Software drauf, das Zusammenspiel mit TTN und die Integration in einen Dienst, der die Daten darstellt und auslesbar macht.

Der Code ist aus diesem Grunde open source und von jedermann/-frau einsehbar. Er wird laufend optimiert. <https://github.com/cyberman54/ESP32-Paxcounter>

Dabei ist das Verhältnis: "Gerät ./ Person", je nach Ort der Messung nicht festgeschrieben. Beispielsweise wären im "Chaos Computer Club" andere Verhältnisse, als etwa an einer Straßenbahnhaltestelle oder einem Seniorenheim zu erwarten. D.h. es kommen individuelle Umrechnungsfaktoren zum Ansatz, zB Kinder, überwiegend ältere Menschen ohne WLAN/Bluetooth Ausstattung, etc..

Es handelt sich derzeit noch um einen Experimentalansatz. Einsatz und Betrieb erfolgen auf eigene Gefahr und Verantwortung. Bitte beachten sie auch die Sicherheitsanweisungen.

Grundsätzlich zählen wir mit dem Gerät die MAC Adressen von Geräten, übermitteln diese per TTN und geben die aggregierten Daten anschliessend im Dashboard von Taigo. aus.

Danke

- an alle die an der Bauanleitung mitgewirkt haben
- die Initiatoren und Programmierer des PaxCounter Projekts
- die Leute die den Bastelabend möglich gemacht haben

Was wir benötigen:

- Hardware
- Stromversorgung
- Gehäuse

Zum Bauen:

- Software zum installieren auf dem Laptop
 - Atom oder (Platformio hat die Unterstützung von Atom abgekündigt, die Erweiterung ist nur noch für VSC Code verfügbar) - <https://atom.io>
 - VS Code - <https://code.visualstudio.com>
 - und Platformio - <https://platformio.org/platformio-ide>
- Software für die Hardware - <https://github.com/cyberman54/ESP32-Paxcounter>

Accounts

- Einen The Things Network Account - <https://www.thethingsnetwork.org>
- Einen Tagio Account - <https://tagio.io>

Anleitung

Das eigentliche zusammenbauen und konfigurieren ist nicht schwierig, solange man sich an die Anleitung hier hält. An einigen Stellen sind auch alternative Lösungen oder Konfiguration möglich, der hier beschriebene Weg, ist einer von vielen Möglichen.

Hardware

Für unser Basteln haben wir mit dem diesem Board gearbeitet, die Erklärungen beziehen sich darauf.

<https://de.aliexpress.com/item/TTGO-ESP32-Paxcounter-LoRa32-V2-1-1-6-version-433-868-915MHZ-LoRa-ESP-32-OLED/32915894264.html?spm=a2g0s.9042311.0.0.3da24c4djf7RIR>

Beim Bestellen bitte die **868 MhZ Version für Europa auswählen**.

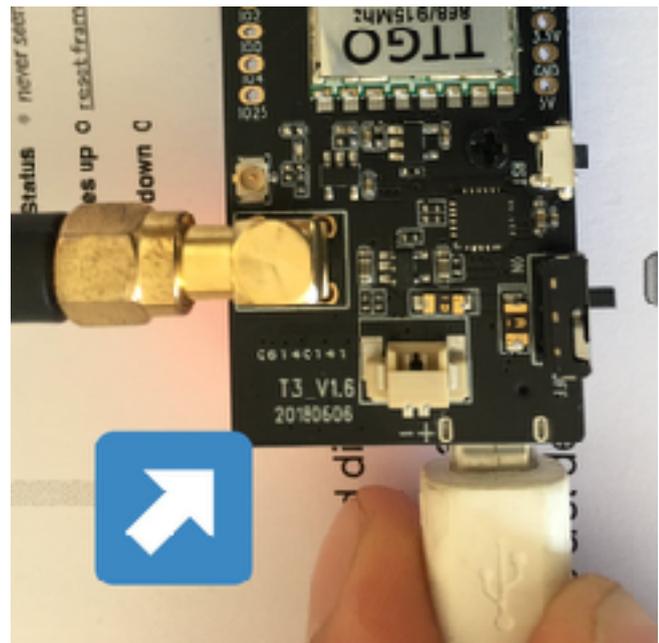
AliExpress liefert die Dinge aus China in ca. 6 Wochen. Es hat sich als Zuverlässig erwiesen.

Vor dem Basteln sollte man besonders bei den TTGO Boards prüfen um welche Version es sich handelt. Im Zweifel Anhand der Beschriftung und Versionsangabe auf der Rückseite prüfen. Einfach diese mit TTGO googlen.

Um Stromversorgung und Gehäuse muss sich jeder selber kümmern.

Vorsicht:

Allerdings hat die Platine wohl ein Design Problem: Sie sollte nicht verwendet werden um eine Batterie zu laden. Da wohl anstellen einer Diode eine Sicherung verbaut wurde.



<https://www.thethingsnetwork.org/forum/t/big-esp32-sx127x-topic-part-2/11973/822>

TTGO Versionen

Der TTGO V.2.1 wird in unterschiedlichen Versionen verkauft.
im Verzeichnis src/hal/ findet sich eine Datei ttgov21.h. In dieser ist eine Anleitung:

```
/ - without label          -> use settings (2)
/ - labeled V1.5 on pcb    -> use settings (2)
/ - labeled V1.6 on pcb -> use settings (1)
```

Wir sollten V1.6 auf der Platine haben.

Accounts

Wenn sie noch keinen haben:

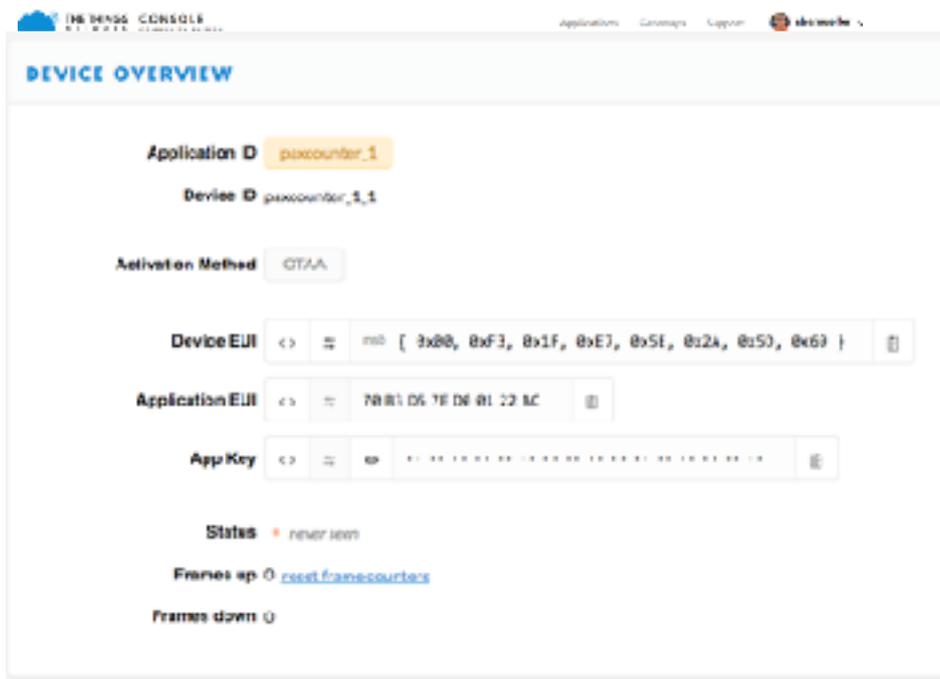
Melden sie sich bei „The Things Network“ <https://www.thethingsnetwork.org/> an und melden sie sich bei <https://tago.io/> an.

1. Things Network

Wenn sie bei <https://www.thethingsnetwork.org/> angemeldet sind klicken sie auf das Foto mit ihrem Profilbild und öffnen sie die Konsole (<https://console.thethingsnetwork.org/>) klicken sie in dieser auf „Applications“ (<https://console.thethingsnetwork.org/applications>).

Klicken sie auf „add application“.

Geben sie unter „**Application ID**“ ihrem PaxCounter einen Namen. Sollte alles klein geschrieben sein und zusammen. Etwa: „vorname_nachname_paxcounter_1“ - The Things Network will für jede ID einen individuellen Namen haben. Am einfachsten geht das wenn man etwa die eigenen Initialen nimmt und dann paxcounter_1. Es muß auch alles klein geschrieben sein und Underscores _ enthalten.



Die restlichen Felder lassen sie so wie sie sind.

Sie kommen dann in die „**Application Overview**“. Auf dem Feld „Devices“ klicken sie auf „register device“

Sie bekommen eine **Übersicht: „Register Device“**

Legen sie hier eine Device ID (etwa: sm_paxcounter_1-1) fest. Die anderen Felder füllt TTN für sie aus. Klicken sie register.

Bei Device Overview wird ihnen angezeigt:

Hier sehen sie die Device EUI und Application EUI. Die brauchen wir für die Software und zwar im msb Format. Das ist das mit diesen beiden geschwiegten Klammern { 0x00 und Zahlen die mit 0x anfangen. Das erhalten sie indem sie auf das Feld mit < > klicken.

Die activation Method lassen wir bei OTAA.

Dieses Fenster können sie offen lassen oder später zurückkehren. Wir brauchen die Device EUI und Application EUI später, wenn wir sie in mit der Software auf unsere Platine kopieren.

Jetzt müssen wir dem The Things Network noch die **Dekoderfunktion beibringen**.

Klicken sie oben auf den Namen der Anwendung, also wahrscheinlich „paxcounter_1“. So habe ich meine hier im Beispiel genannt.

Sie sehen dann eine Reiter „Payload Formats“. Da drauf klicken und in das Feld „decoder“ die Payload Funktion rüberkopieren.

Diese finden sie bei: https://github.com/cyberman54/ESP32-Paxcounter/blob/master/src/TTN/plain_decoder.js

Dann speichern (unten rechts save)

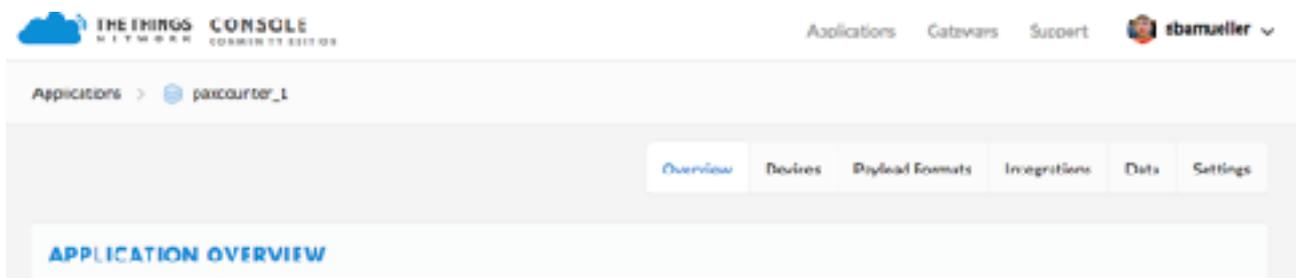
Jetzt haben wir dem TTN beigebracht wie wir die Daten die ankommen werden verarbeiten.

Wir können auch noch die Converter Funktion einrichten. Dazu klicken wir bei „Payload Formats“ auf

Wir empfehlen jedes Gerät in einer eigenen Integration laufen zu lassen. Daher für jedes Device eine eigene Integration.

Integration

Wir wollen dann die Daten die wir bekommen nicht nur in der TTN Konsole speichern, sondern auch schön in einem Dashboard darstellen.



Dazu gehen wir in der gleichen Übersicht auf „Integrations“. Und klicken wieder auf „add integration“.

Die einfache Integration, die einfach für sieben Tage die Daten speichert und aus der man sie rauskopieren kann ist „Data Storage“. Diese lässt sich mit ein paar Klicks einschalten, ist aber auch nicht besonders Übersichtlich.

Wie sie Daten mit Hilfe von IFTT etwa auf google Drive, Dropbox oder in einen Telegramm Channel schieben wollen finden sie eine Anleitung auf meinem Blog: <https://sbamueller.wordpress.com/2018/06/24/build-your-own-bike-tracker-with-ttn/>

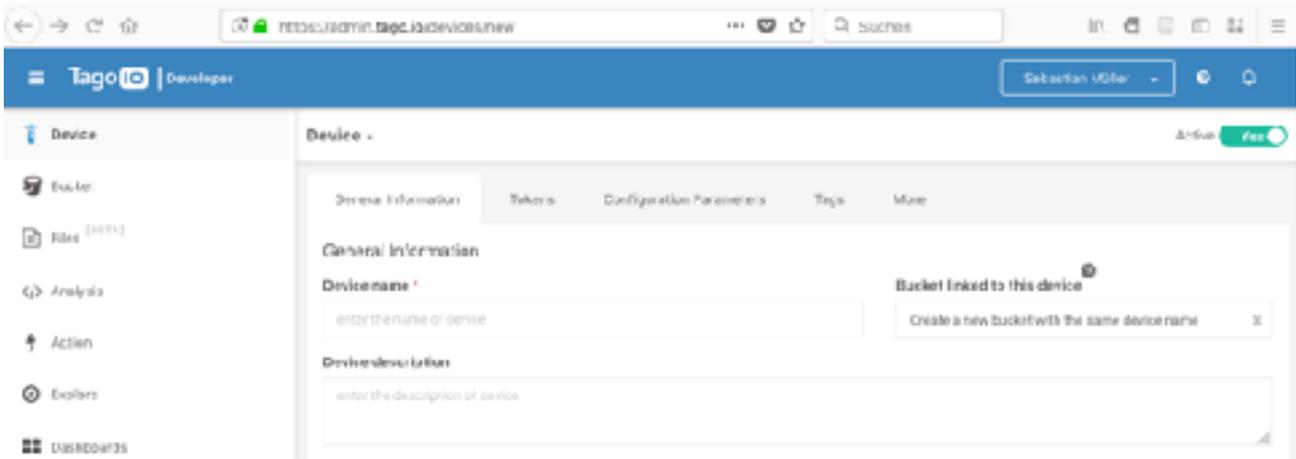
Wir wollen die Daten aber ins <https://tago.io/> kopieren, weil die ein schönes Dashboard bieten. Für bis zu 5 Geräte es auch umsonst.

Wir machen uns bei <https://tago.io/> einen Account.

Dann gehen wir oben in der blauen Zeile auf unseren Namen und klicken „my account“ und wählen beim Reiter „General Information“ den Hacken bzw. Punkt bei: „I am a **Developer** - I will build applications on Tago“. Wir klicken speichern.

Jetzt gehen wir auf „visualization preferences“ und wählen im Dropdown Menü „Admin visualization mode“: „Developer (full view with all tools...)“ aus.

Beides brauchen wir um eigene Sensoren anzubinden, selbst das Dashboard zu gestalten.

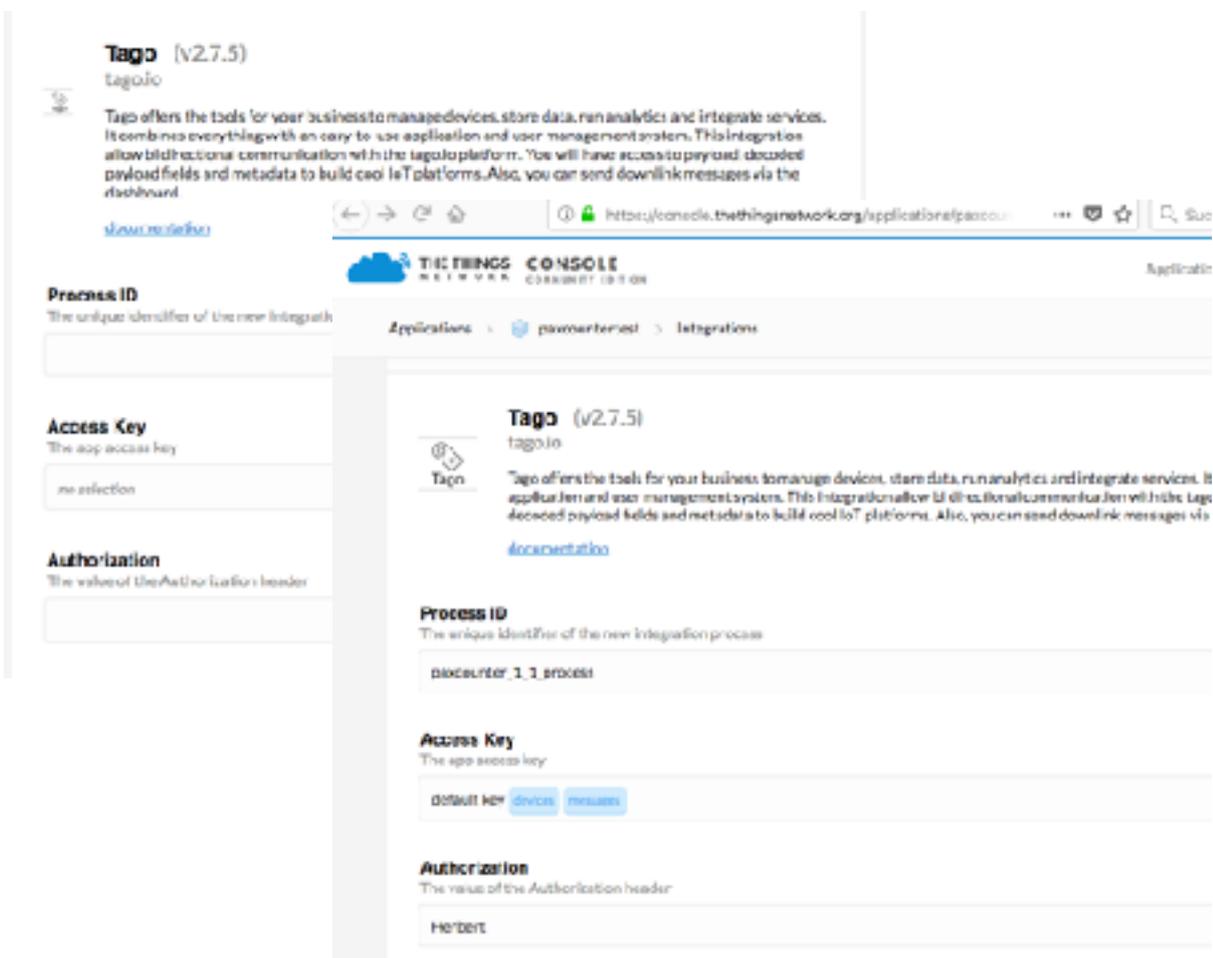


Nur dann kann man die Dinge alle machen, die wir machen machen wollen. Es gibt unter: <https://www.youtube.com/watch?v=ppZW9wfggpE&feature=youtu.be> ein sehr schönes (englisches) YouTube Video das uns alles erläutert.

Um den Paxcounter einzurichten gehen wir dann auf „Device“ (links oben) und klicken „add device“.

Sie bekommen dann eine Übersicht "General Information" bitte geben sie hier einen Device Name an. Am besten geben sie hier den gleichen Namen ein, dem sie auch dem PaxCounter im TTN gegeben haben. Etwa „paxcounter_1_1“.

Die Daten werden in einem „Bucket“ (= Eimer) gespeichert, den uns Tagoio automatisch anlegt.



Klicken sie dann auf „Tokens“ bei „Serial Number“ geben sie die Device EUI aus der Device EUI (als Hexadezimal Zahl: die sieht dann etwa so aus 123CV3235ABC, also eine normale lesbare Zahl) im The Things Network ein und bei Token einen Text den sie sich merken sollten, bzw. vorhalten. Dann „Generate“ drücken. Bei Verifikation können sie einen Zufälligen Schlüssel oder eine Art Passwort eintragen, das gleiche Passwort müßen sie dann in der TTN Konsole auch eintragen (wird gleich beschrieben, ich sag mal „Herbert“).

Jetzt gehen wir wieder in die TTN Konsole. Wählen sie dort ihre Application für den Paxcounter aus und klicken auf Integrations. Dann klicken sie bitte auf „add integration“ und wählen „Tago“ aus.

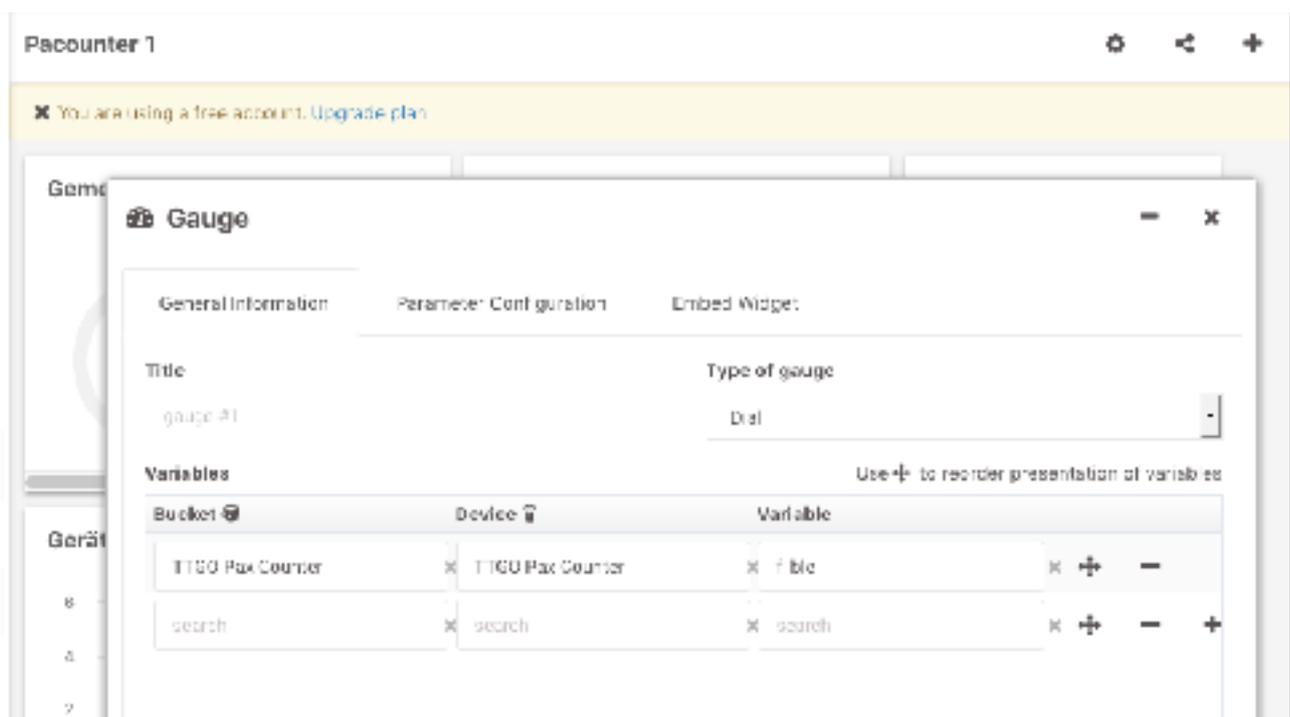
Geben sie dem ganzen eine Process ID, etwa „paxcounter_1_1_process“, bei „Access Key“ wählen sie die „default key“ aus und bei Autorisation geben sie das Passwort ein also bei uns „Herbert“) Software runterladen

Für weitere Informationen, siehe auch <https://tago.elevio.help/en/articles/102-ttn-the-things-network> als Anleitung.

Sobald das Device Daten liefert können wir nun im tago.io auf Dashboards klicken, ein neues Dashboard mit „Add Dashboard“ aufmachen. In diesem klicken wir dann auf das „+“ (Add Widget) und wähle etwa „Gauge“ aus. Geben diesem einen Namen.

Wir wählen dann den „Bucket“ in unserem Fall etwa „paxcounter_1_1_process“, er wird dann gleich den Device Namen einfügen und dann bei „Variable“: „f-ble“ für Bluetooth Geräte und „f_wifi“ für die WIFI Geräte.

Bild auf der nächsten Seite.



Mit dem  lassen sich die Eigenschaften des Dashboard einstellen, etwa den Link zu diesem teilen oder über den  „Actions“ definieren, daher etwa bei der Überschreitung eines bestimmten Wertes soll eine Email geschrieben werden.

Material

Hier findet sich die Anleitung von TTN auf Englisch: <https://www.thethingsnetwork.org/docs/applications/tago/>

Die Hardware

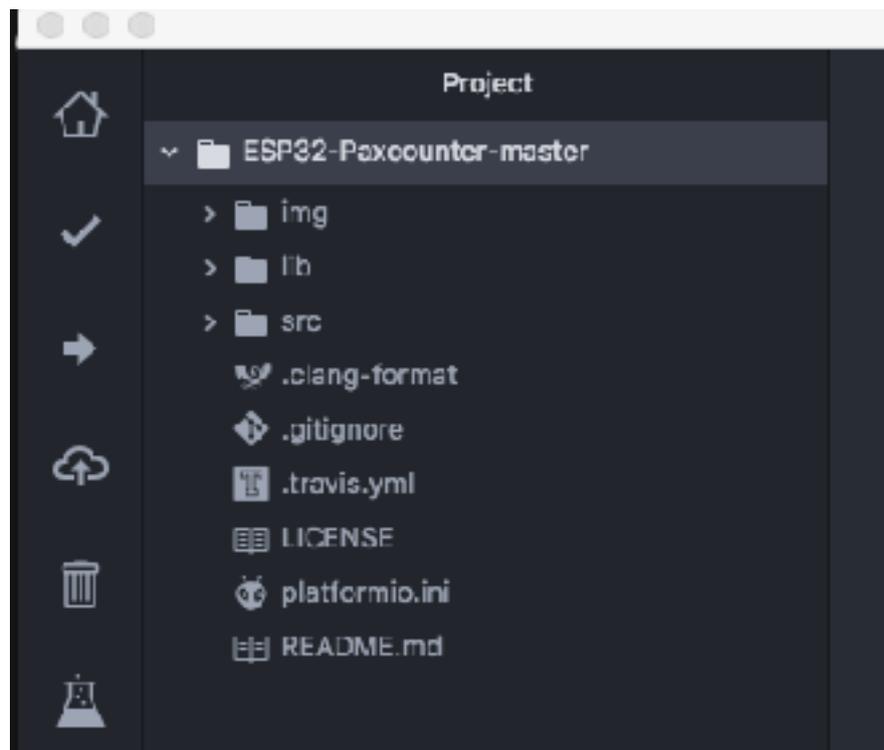
Laden sie den Editor Atom <http://atom.io> herunter und installieren diesen.
Laden sie dann PlatformIO IDE <https://platformio.org/install/ide?install=vscode>

Die Beispiele und screenshots hier sind aus diesem.

Gehen sie auf <https://github.com/cyberman54/ESP32-Paxcounter> und laden sie mit „Clone or download“ das Projekt herunter (Download as ZIP).

Entpacken sie diesen Ordner

Öffnen sie Atom und Nutzen Sie Datei/Öffnen um den Ordner aufzumachen. Sie sollten in der Leiste links im Atom die Übersicht über die Projektstruktur sehen.



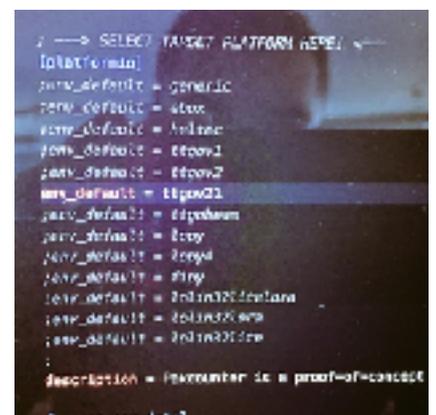
Konfiguration

klicken sie dann auf platformio.ini (mit dem kleinen Alien Gesicht drauf).

Sie sehen Zeilen mit Code. Hier wählen sie den passenden Code für ihr Board aus. Wir haben in Zeile 17 `;env_default = ttgov1`
Im Atom Editor werden die aktiven Code Zeilen rot dargestellt und der Code der aaskommentiert ist oder nur Erklärung als grau. Daher sollten sie das das Semikolon vor der Zeile mit dem ttgov1 entfernen. Dann wird dieser Code rot und vor der default Zeile eines setzen, damit diese Codezeile deaktiviert wird.

TTGO V2.1 Versionen

Standardmäßig ist die Software auf die TTGO Versionen V1.5 ausgelegt. Siehe dazu Anleitung. Wenn man die neuere TTGOo Version V1.6. verwendet müßen



wir Änderungen in der Datei „ttgo21.h“ vornehmen. Diese findet sich im Verzeichnis:/src/hal/ttgo21.h.

Daher wir müssen die unter (1) als Setting angegeben Dinge einschalten. Dazu löschen wir oberhalb von Zeile 10 */ und unterhalb von Zeile 38 */ .

Dann kopieren wir um die (2) Setting, also vor Beginn in Zeile 41 und nach dem Ende in Zeile 71 jeweils diese Zeichen. Die Einfärbung der Zeilen im Editor sollte sich ändern.

```
/*
// (2) settings for boards without label on pcb, or labeled v1.5
on pcb
```

```
***
/*
```

Weitere Aktionen

Öffnen sie im Verzeichnis „src“ die Datei **paxcounter.conf** (da sind so kleine Schieberegler davor)

ggf. sollte man unter Zeile 61

```
// Some hardware settings
#define RGBLUMINOSITY 30 // RGB LED luminosity
[default = 30%]
```

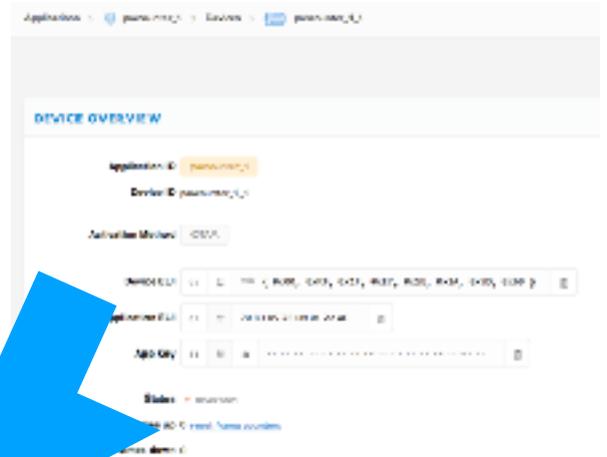
Die RGB Lumiosity auf 10% setzen. Die Teile sind schon sehr hell.

In Zeile 71 ist das europäische Frequenzband schon gesetzt.

Gehen sie zur Datei mit dem großen C „loracnf.sample.h“

Hier geben sie in die schön orange angeleuchteten Teile bitte die DEVEUI, den APPEU und den APPKEY aus der TTN Console ein.

{Achten sie darauf, dass die geschweiften Klammern am Anfang und Ende nur jeweils einmal vorhanden sind.}



```
6 static const u1_t DEVEUI[8] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
7
8 static const u1_t APPEUI[8] = {0x70, 0x83, 0xD5, 0x00, 0x00, 0x00, 0x00, 0x00};
9
10 static const u1_t APPKEY[16] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
11                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
12
```

Diese sind im MSB format. Das können sie in der Konsole unter Application unter „Device Overview“ sehen. Da einfach auf die Schaltfläche mit den „< >“ klicken und der msb wird angezeigt. Kopieren und im Atom einfügen.

Speichern und die Datei umbenennen in **loracnf.h**

Jetzt das ganze auf das Board flashen.

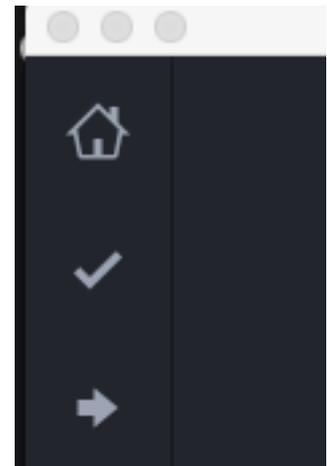
Ganz links erst den ✓ drücken, dann wird der Code kompiliert.

Dann ebenfalls links auf ➡ drücken, der Code wird auf das Board geladen (es sollte per USB angeschlossen sein)

Das richtige USB Gerät wählt man so **????** aus.

Bitte beachten und Geduld (!):

Das erste Datenpaket wird erst nach 240 Sekunden übertragen.

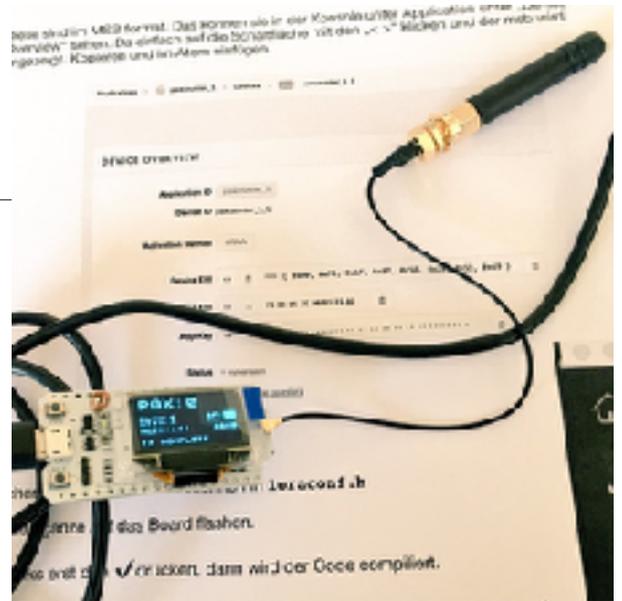


Hinweise:

Datenschutzaspekte

Die Software des PaxCounter ist in der Lage MAC Adressen zu zählen und die Anzahl der Adressen via LoRa zu übermitteln. iOS und Android(?) geben nur noch zufällig generierte MAC Adressen heraus. Diese werden nach einem bestimmten Zeitintervall geändert. Ein Tracking von Nutzern ist dadurch nicht möglich bzw. sehr erschwert.

Es erfolgt lediglich ein Zählen der Anzahl der Geräte (Summenwertbildung).



Anzahl der Personen = Anzahl der Geräte?

Nein. Nicht ganz. Ob und wie sich aus einer Geräte Anzahl, dann auf eine Personenanzahl schließen lässt, dürfte je nach usecase bzw. sozialer Situation sehr unterschiedlich sein. Das muß von den Anwendern jeweils für den eigenen usecase validiert werden, etwa durch eigene Zählung von Referenzsituationen/-werten.

Konfiguration

WIFI FILTER

in wifiscan.cpp Zeile 32 „wifi_snifer_init“ auf = 255 setzen, dann werden nicht nur Management Frames gezählt. Das sollte dazu führen, das auch alle Geräte gezählt werden, auch die die in ein WLAN eingeloggt sind.

VENDOR FILTER

in der paxcounter.conf die Zeile 15 auskommentieren

```
„#define VENDORFILTER 1 // comment out if you want to count things, not
people“
daher vor die Zeile //
```

Fernsteuerung

TTN Geräte können auch ohne direkten physischen Zugriff auf das eigentliche Gerät, aus der TTN Konsole ferngesteuert werden.

Wie es geht weiß ich aber noch nicht