

# Projektarbeit Pager

Salih Can Arioglu

Erstes Ausbildungsjahr 2020/21

Betreut von Uwe Bastian

# Inhaltsverzeichnis

1 Einleitung.....	1
1.1 Was ist ein Pager? .....	1
1.2 Motivation .....	1
1.3 Funktionsweise .....	1
2 Entwurf .....	2
2.1 Realisierung .....	2
2.2 Verwendete Bauteile .....	3
2.2.1 CubeCell© Boards .....	3
2.2.2 LiPo Akku 1000mAh .....	5
2.2.3 Leuchtdiode (LED) .....	5
2.2.4 Taster .....	6
2.2.5 Buzzer Piepser Signalgeber .....	6
2.2.6 Gehäuse .....	7
3. Software .....	8
3.1 Pager_OS.....	8

# 1 Einleitung

## 1.1 Was ist ein Pager?

Ein Pager – auch Funkmeldeempfänger (FME) oder Funkrufempfänger – ist ein handlicher mobiler Funkempfänger, der zu Alarmierungszwecken sowie zu Nachrichtenübermittlung eingesetzt wird (Wikipedia).

## 1.2 Motivation

In der heutigen Zeit werden rund um die Uhr große Mengen an Daten über den gesamten Globus übertragen. Um dies zu gewährleisten werden stabile Stromnetzwerke benötigt, die die Server und den Funkmasten sicher mit Energie versorgen. Kommt es mal zu einem Zwischenfall (Stromausfall), so wäre ein großer Bereich vom Stromnetz getrennt. Das wäre noch nicht alles, die jeweiligen Funkmasten und Server, die nicht Notstrom versorgt werden, wäre ebenfalls offline. Damit wäre die Datenübertragung ebenfalls in diesem Bereich lahmgelegt. Die Fachleute, die das Problem beheben sollen, könnte man nicht oder nur nach langer Zeit erreichen. Hier findet der Pager sein Anwendungsbereich. Bei einem vorliegenden Fehler wäre es möglich über ein Notstrom versorgtes Netzwerk (LoRaWAN, LWL/Lanstrecke) eine Nachricht an die jeweiligen Fachleute zu übermitteln. Somit wäre eine konstante Kommunikation sichergestellt.

## 1.3 Funktionsweise

Der Pager wird für die Datenübertragung auf „The Things Network“ (kurz TTN) registriert. Sobald dieser angemeldet ist, überträgt er zunächst seine Batteriespannung und wiederholt dies alle 20 min. Mit der „Schedule downlink“-Funktion von TTN, kann ein gewünschter Payload im Hex-Code an den Pager gesendet werden. Der Pager läuft im C-Betrieb, damit die Nachricht sofort empfangen werden kann. Die empfangene Nachricht wird dann in ASCII-Zeichen umgewandelt und auf dem Display ausgegeben. Zusätzlich leuchtet eine LED auf. Dieser Zustand bleibt erhalten, bis eine neue Nachricht empfangen wird, der Taster oder der Reset-Knopf betätigt wird. Die empfangene Nachricht (Payload) wird nach der Umwandlung auch nach einem bestimmten Zeichen oder Zeichenfolge durchsucht, standardmäßig das „@“-Zeichen. Liegt dieses vor so handelt es sich um eine dringende Nachricht. Die LED fängt an zu blinken und ein Mini-Signalgeber gibt im gleichen Takt ein akustisches Signal von sich. Diese Nachricht kann nicht von einer neu empfangenen Nachricht überschrieben werden. Es muss der Taster oder der „Reset“-Knopf am Pager betätigt werden. Damit ist sichergestellt, dass die Nachricht auch gelesen wird.

## 2 Entwurf

### 2.1 Realisierung

Für die Realisierung des Pagers wird das „Heltec© CubeCell® Dev-Board Plus“ oder der „CubeCell® GPS-6502 HTCC-AB02S“ verwendet. Diese Boards unterstützen die Arduinoumgebung und LoRaWAN 1.0.2. Über das schon vormontierte Display werden die empfangenen Nachrichten ausgegeben. Den benötigten Strom liefert ein 1000mAh Lithiumakku und kann direkt an das integrierte Lademodul angeschlossen werden.

Die PINS werden wie folgt angeschlossen:

- GPIO5 der Mini-Piepser (Signalgeber),
- GPIO6 die LED mit Vorwiderstand (800Ohm),
- GPIO7 der Taster mit einem Pulldown-Widerstand (10kOhm),
- VDD für die Spannungsversorgung (3,7V).

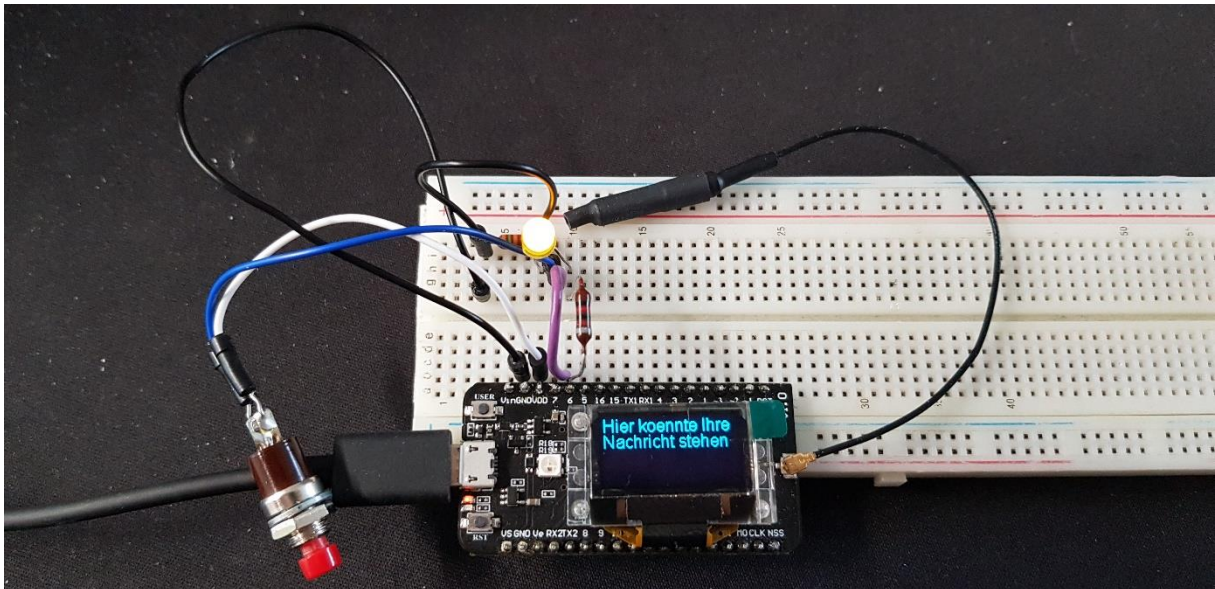


Abbildung 1: Testaufbau (ohne Signalgeber an GPIO5)

## 2.2 Verwendete Bauteile

### 2.2.1 CubeCell© Boards

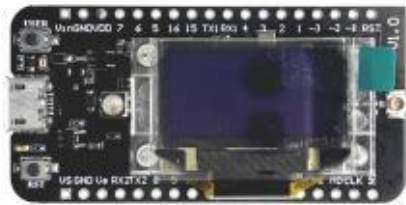


Abbildung 3: CubeCell© Lora GPS-6502



Abbildung 2: CubeCell© Dev-Board Plus

Die CubeCell© Boards mit Oled Display sind Preiswerte LoRaWAN Boards, die sich perfekt für Lora Node Applikationen eignet. Diese Boards sind Arduino© kompatibel und können mit einer detaillierten Anleitung von Heltec© einfach auf die Arduino IDE integriert und programmiert werden. Zudem besitzen die Boards genügend digitale Pins (GPIO). Durch das Ultra Low Power Design haben die Boards im Deep Sleep Modus einen Stromverbrauch von ungefähr 56uA. Die Akkuspannung kann über den Analogeingang vom Mikrokontroller einfach abgefragt werden.

#### *Technische Daten:*

- Mikroprozessor: ASR6501 (48 MHz ARM® Cortex® M0+ MCU, 128KB FLASH, 16KB SRAM)
- Oled Farbe: Weiss
- Oled Auflösung: 128x64
- LoRa-Chip SX1262
- LoRa-Arbeitsband: 868MHz EU\_863\_870
- LoRa maximale Ausgangsleistung  $22 \pm 1$  dB
- 6 x GPIO
- FLASH: 128KB intern
- Schnittstelle: Mikro USB und LoRa (IPEX)
- Akku-Anschluss: 3.7V Lithium SH1,25 2pin
- LoRa 14dBm Output: 90mA
- Schlafmodus Deep Sleep (VBAT/Batteriebetrieb): 11uA

Electrical Features	Condition	Minimum	Typica	Maximum
Power Supply	USB powered ( $\geq 500\text{mA}$ )	4.7V	5V	6V
	Lithium powered ( $\geq 250\text{mA}$ )	3.3V	3.7V	4.2V
	3.3V (pin) powered ( $\geq 150\text{mA}$ )	2.7V	3.3V	3.5V
	5V (pin) powered ( $\geq 500\text{mA}$ )	4.7V	5V	6V
Power Consumption(mA)	LoRa Rx Mode		10mA	
	LoRa 10dB output		70mA	
	LoRa 14dB output		90mA	
	LoRa 17dB output		100mA	
	LoRa 20dB output		105mA	
	Sleep Mode (USB powered)		9.6mA	
	Sleep Mode (VBAT/battery powered)		21 $\mu\text{A}$	
Output	3.3V pin output			500mA
	5V pin output (USB powered only)		Equal to the input current	
	External device power control (Vext 3.3V)			350mA

Abbildung 4: elektrische Charakteristik

*Download und Links:*

- CubeCell® Dev-Board Plus (Heltec®)
  - <https://Heltec.org/project/htcc-ab02/>
- CubeCell® Dev-Board Plus (bastelgarage)
  - <https://www.bastelgarage.ch/CubeCell-dev-board-plus-868mhz-lora-node-htcc-ab02>
- CubeCell® GPS-6502 (Heltec®)
  - <https://Heltec.org/project/htcc-ab02s/>
- CubeCell® GPS-6502 (bastelgarage)
  - <https://www.bastelgarage.ch/CubeCell-gps-6502-868mhz-lora-node-htcc-ab02s?search=cubecel>
- CubeCell® GPS-6502 (Amazon)
  - [https://smile.amazon.de/Seamuing-LoRaWAN-Development-ASR6502-intelligente/dp/B08B1G5THB/ref=sr\\_1\\_6?\\_mk\\_de\\_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=sx1262&qid=1623043393&sr=8-6](https://smile.amazon.de/Seamuing-LoRaWAN-Development-ASR6502-intelligente/dp/B08B1G5THB/ref=sr_1_6?_mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=sx1262&qid=1623043393&sr=8-6)
- Arduino Installation Anleitung
  - [https://Heltec-automation-docs.readthedocs.io/en/latest/CubeCell/quick\\_start.html#use-arduino-board-manager](https://Heltec-automation-docs.readthedocs.io/en/latest/CubeCell/quick_start.html#use-arduino-board-manager)

### 2.2.2 LiPo Akku 1000mAh



Abbildung 5: LiPo Akku

Dieser Akku ist für die beiden verwendeten Boards ideal ausgelegt und besitzt einen 2Pin Stecker, der in die Buchse auf der Unterseite der Boards passt. Zudem hat er zusätzlich eine Schutzelektronik eingebaut, die eine Tiefenentladung und Überladung verhindert. Der Akku hat eine Kapazität von 1000mAh, dadurch kann der Pager ca. drei Tage ohne nachladen betrieben werden kann.

#### Technische Daten:

- Typ: 102050
- Spannung: 3.7V
- Kapazität: 1000mAh
- Betriebstemperatur max.: -25°C bis max +60°C!
- Kabellänge: 35mm
- Stecker: JST-PH 2Pin 1.25mm
- Grösse: 52 x 20 x 9mm
- Gewicht: 17g

#### Download und Links:

- LiPo Akku (Amazon)
  - [https://www.amazon.de/EREMIT-Batterie-1000mAh-803040-Stecker-Gelb-Wei%C3%9F/dp/B0842536CD/ref=sr\\_1\\_6?\\_mk\\_de\\_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=LiPo+Akku+1000mAh&qid=1622873294&sr=8-6](https://www.amazon.de/EREMIT-Batterie-1000mAh-803040-Stecker-Gelb-Wei%C3%9F/dp/B0842536CD/ref=sr_1_6?_mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=LiPo+Akku+1000mAh&qid=1622873294&sr=8-6)
- LiPo Akku (bastelgarage)
  - <https://www.bastelgarage.ch/lipo-akku-1000mah-jst-1-25-lithium-ion-polymer-fur-lora-ttgo>

### 2.2.3 Leuchtdiode (LED)



Abbildung 6: LED 3mm Rot

Es wird eine handelsübliche LED mit einem Durchmesser von 3mm verwendet. Diese wird mit einem Vorwiderstand (80Ohm) an den digitalen Pin (GPIO6) vom Board angeschlossen, da an den Pins 3,3V anliegen.

## 2.2.4 Taster



Abbildung 7: Mini-Taster Schließer

Bei dem Taster handelt es sich um einen Schließer. Dieser wird zwischen dem Anschluss „VDD“ und dem digitalen Pin (GPIO7) angeschlossen. Zusätzlich wird an dem Pin (GPIO7) ein 10kOhm Pulldown-Widerstand auf Masse gelegt. Immer wenn der Taster betätigt wird, liegt am GPIO7 die Spannung VDD an.

## 2.2.5 Buzzer Piepser Signalgeber



Abbildung 8: Mini-Piepser

Der Signalgeber wird an den digitalen Pin (GPIO5) angeschlossen und sobald eine Spannung anliegt, wird ein akustisches Signal erzeugt.

### Technische Daten:

- Lautstärke: 80 dB
- Tonart: Dauerton
- Versorgungsspannung: 3,3V

### Download und Links:

- Signalgeber 6V
  - <https://www.ebay.de/itm/111246283031>
- Signalgeber 3,3 – 5 V
  - <https://www.ebay.de/itm/174640633804?hash=item28a96497cc:g:3cMAAOSw4oBgK-2C>



## 2.2.6 Gehäuse



Abbildung 9: Gehäuse mit Deckel

### 2.2.6.1 Pager Gehäuse und Deckel

Das Gehäuse ist eine Eigenentwicklung. Zunächst wurde ein öffentlich zur Verfügung stehendes Gehäuse genommen und anschließend angepasst. Die STL-Datei für das Gehäuse und dem entsprechenden Deckel stehen zur freien Verfügung (GitHub). Dieses Gehäuse ist für den CubeCell® GPS-6502 konstruiert. Der Akku wird unter dem Board verstaut und die Antenne hat ihre eigene geschlossene Kammer. In diesem Gehäuse ist kein Platz für den Mini-Taster, hier wird über den Reset-Knopf die empfangene Nachricht gelöscht.

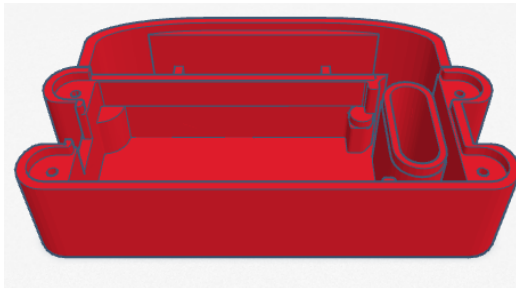


Abbildung 11: 3D-Modell Gehäuse

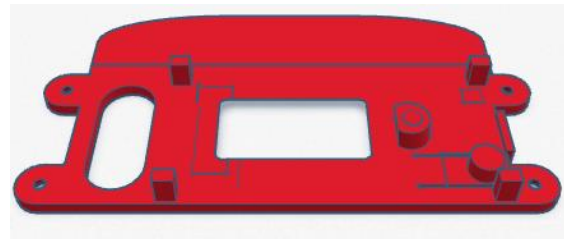


Abbildung 10: 3D-Modell Deckel

### 2.2.6.2 Pager Komplett

Dieses Gehäuse bietet genug Platz für alle Bauteile und auch für ein deutlich größeren Akku. Zusätzlich gibt es auch eine Öffnung für ein externes Lademodul. Die Halterungen für das Board am Deckel haben nicht die richtige Maße, daher muss viel mit Heißkleber gearbeitet werden.

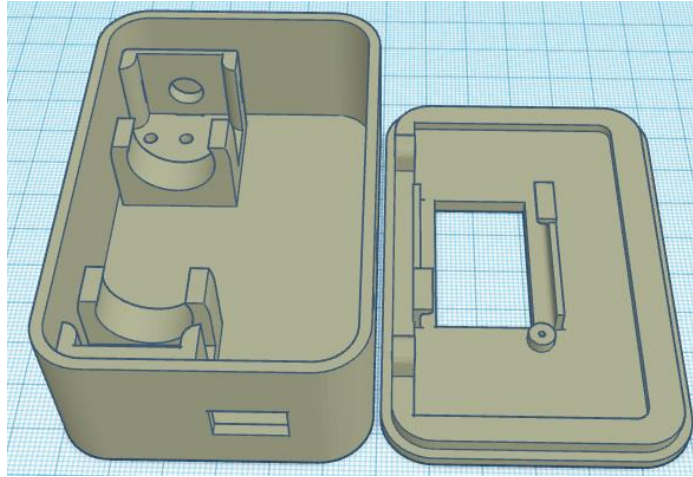


Abbildung 12: 3D-Modell Komplett

#### Download und Links:

- Gehäuse
  - <https://github.com/BadUwe/Pager/tree/main/3D%20Druck>
- Laderegler 5V
  - <https://www.ebay.de/itm/133145660842?hash=item1f00198daa:g:OkoAAOSwhQtav1c8>

## 3. Software

### 3.1 Pager\_OS

Das Pager Programm ist eine Eigenentwicklung und steht zur freien Verfügung. Wir übernehmen keine Garantie für die Funktionalität des Programms und übernehmen keine Haftung oder Verantwortung für das Auftreten von Problemen und/oder entstandenen Schäden, die bei der Anwendung auftreten können. Auf die Funktionsweise des Programms, wird hier nicht weiter eingegangen, diese sind im Programm selbst ausführlich kommentiert.

#### Download und Links:

- Pager\_OS
  -