

## Ein Raspi als Beacon ...

... sendet lokal begrenzt, aber "frei im Raum schwebende" Information: Ein mögliches Element von "angereicherter Realität" (Augmented Reality)

Wer einen Raspberry Pi 4 oder Pi Zero W (Raspi) zur Verfügung hat, kann diesen mit wenigen Befehlen in ein Beacon (engl. "Leuchtfieber") verwandeln. Beide Bauformen besitzen eingebaute Bluetooth-Module, die geeignet sind, im BLE-Modus (als besonders energiesparende Bluetooth-Variante) zu arbeiten. BLE-Module sind Voraussetzung für die Programmierung von Beacons. Letztere gibt es in Apple- (iBeacons) und Android-Versionen (Eddystone) mit jeweils unterschiedlicher Programmierung. Wir legen hier das besondere Augenmerk auf Eddystone-Beacons.

Das notwendige Vorgehen, wie du einen Raspi zum Beacon machen kannst und die Erklärung, was durch die einzelnen Schritte im Detail bewirkt wird, ist z.B. auf <https://hackaday.io/project/10314-raspberry-pi-3-as-an-eddystone-url-beacon> ausführlicher beschrieben. Die Zusammenhänge sind kompliziert und

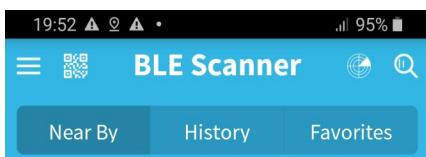
```
pi@raspberrypi:~ 
Datei Bearbeiten Reiter Hilfe
pi@raspberrypi:~ $ sudo hciconfig hci0 up
pi@raspberrypi:~ $ sudo hciconfig hci0 leadv 3
LE set advertise enable on hci0 turned status 12
pi@raspberrypi:~ $ sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe
0f 16 aa fe 10 00 03 73 70 65 79 65 72 2e 62 65
< HCI Command: ogf 0x08, ocf 0x0008, plen 24
 17 02 01 06 03 03 AA FE 0F 16 AA FE 10 00 03 73 70 65 79 65
 72 2E 62 65
> HCI Event: 0xe0 plen 4
 01 08 20 00
pi@raspberrypi:~ $
```

vielgestaltig. Wir versuchen hier einen ersten Einblick zu geben, der leider viele Fragen auch offen lassen muss.

Voraussetzung, um der folgenden kompakte Anleitung folgen zu können: Du musst wissen, wie du Befehle an einen Raspi über das Konsolenprogramm (Bild links) gibst.

Hier die drei notwendigen Befehle für einen ersten "Raspi als Beacon"-Test. Sie müssen jeweils ohne Zeilenumbruch eingegeben werden:

```
sudo hciconfig hci0 up
sudo hciconfig hci0 leadv 3
sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe 0f 16 aa fe 10 00 03 73 70 65 79 65 72 2e 64 65
```



Im Konsolenfenster des Raspis müsste sich dann das oben gezeigte Bild ergeben.  
Hinweis: "sudo hciconfig hci0 up" wird nicht mit einer Rückmeldung kommentiert, wenn dieser Befehl richtig ausgeführt wurde.



Wenn der Raspi die drei Befehle ausgeführt hat, müsste jetzt die (im Playstore angebotene) BLE-Scanner-App die Beacon-Information so wie auf dem Bild links anzeigen. Der Raspi war ca. 16 cm vom Smartphone entfernt und dieses reagierte auf das Tippen auf "OPEN URL" mit dem Laden der Seite "[www.Speyer.de](http://www.Speyer.de)" in den Smartphone-Browser.

Warum sich diese Anzeige und diese Funktion als Link ergeben hat, ist in dem dritten Befehl verschlüsselt:

```
sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe 0f 16 aa fe 10 00 03 73 70 65 79 65 72 2e 64 65
```

Die technischen Details sind auf der oben genannten Seite erklärt.

Hier nur kurz dazu, wo dort "[speyer.de](http://speyer.de)" steht. Das ist die hexadezimale Darstellung der rot hervorgehobenen ASCII-Zeichen:

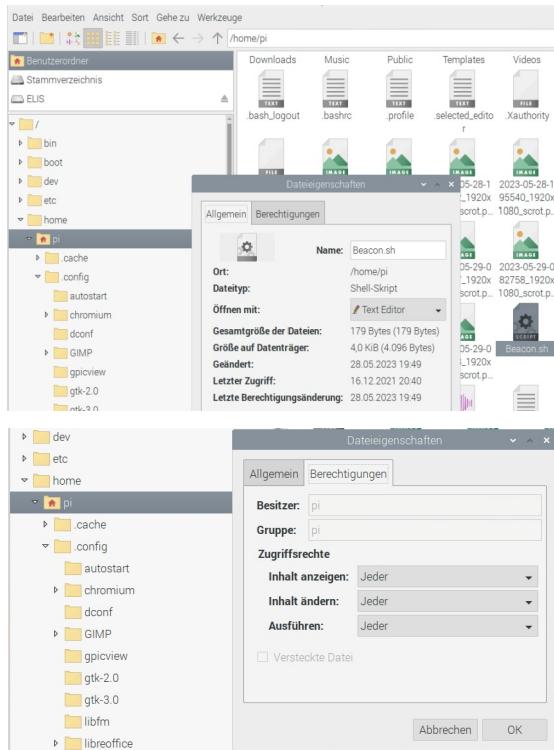
73 steht für ein kleines s, 70 für ein p ..... 2e für den Punkt etc.

Im Folgenden zeigen wir noch, wie der Raspi so eingerichtet werden kann, dass er beim Hochfahren automatisch zum oben beschriebene Eddystone-Beacon. Diese Seiten zu lesen lohnt sich aber nur, wenn du schon tiefer in die Raspi-Welt eingestiegen bist oder es für die Zukunft anstrebst.

## Autostart der Befehlsfolge für das Raspi-Eddystone-Beacon

Im Verzeichnis /home/pi/ wird mit dem normalen Raspi-Editor eine Datei angelegt mit den Zeilen:

```
#!/bin/bash
sudo hciconfig hci0 up
sudo hciconfig hci0 leadv 3
sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe 0f 16 aa fe 10 00 03 73 70 65 79 65 72 2e 64 65
exit 0
```



Diese Datei wird "Beacon.sh" genannt. Dadurch wird dem Raspi mitgeteilt, dass es sich um eine sogenannte "Bash"- oder "Shell"-Datei handelt. Das ist eine Datei, die eine Befehlsfolge für das Betriebssystem enthält.

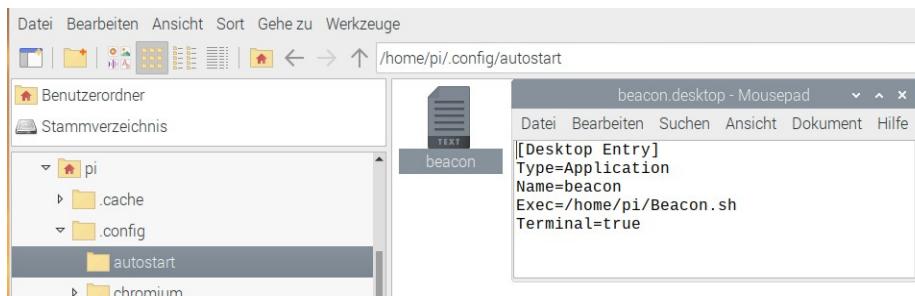
Bei Windows heißen solche Dateien "Batch-Dateien".

Wenn du diese Datei erstellt hast, musst du zusätzlich noch mit den Rechten ausstatten, dass sie vom Rechner ausgeführt werden kann. Dazu gehst du in den Dateiordner, klickst mit der rechten Maustaste auf die Datei und wählst "Dateieigenschaften". Du siehst die Reiter "Allgemein" und "Berechtigungen". Schalte auf "Berechtigungen", setze für diesen Zweck alle drei Zugriffsrechte auf "Jeder".

Nachdem du das Dateieigenschaften-Fenster geschlossen hast, klickst du wieder mit der rechten Maustaste auf die Datei, wählst diesmal aber "Öffnen" und dann "Im Terminal ausführen". Nun sollte die Befehlsfolge ablaufen und der Raspi zum Beacon werden.

**Wenn das funktioniert, musst du zusätzlich die eigentliche Autostart-Datei angelegen, die schließlich bei jedem Hochfahren des Raspis "Beacon.sh" startet.** Das ist noch ein

Stück komplizierter. Im Verzeichnis /home/pi/.config/autostart/ muss die Datei beacon.desktop vorhanden sein. Die Endung ".desktop" erscheint allerdings erst, wenn sie geöffnet wird. Achtung: Bei der Auswahl von /.config durch den Dateimangers im Ordner "pi" muss die Option "Versteckte Dateien anzeigen" aktiviert sein, damit der Ordner ".config" überhaupt gefunden werden kann.



Diese Datei lässt sich meines

Wissens nach nur über ein Konsolenkommando erzeugen:

```
pi@raspberrypi:~ $ sudo nano /home/pi/.config/autostart/beacon.desktop
```

Zum Kopieren: **sudo nano /home/pi/.config/autostart/beacon.desktop**.

Durch diesen Befehl wird die noch leere Datei auch gleich mit dem nano-Editor geöffnet. Sie muss dann mit den gezeigten Einträgen gefüllt werden.

(Die Eingabe, das Speichern und das Schließen von Dateien mit dem Nano-Editor ist gewöhnungsbedürftig. Beispielsweise wird das Speichern mit dem gleichzeitigen Drücken von "Strg" und "o" ausgeführt und das Schließen der Datei anschließend mit "Strg" und "x".)

Wenn diese Datei dann existiert, muss sie ebenfalls die Rechte bekommen, dass sie ausgeführt wird. Das gelingt mit der Konsoleneingabe:

Mit eRechtsklick auf die Datei im Fenster des Dateimangers kommst du nicht weiter, weil die Dateien in "./config" für den normalen Zugriff gesperrt sind.

```
pi@raspberrypi:~ $ sudo chmod 777 /home/pi/.config/autostart/beacon.desktop
```

Hier zum Kopieren: **sudo chmod -R 777 /home/pi/.config/autostart/beacon.desktop**

Nun sollte sich "beacon.desktop" mit einem Rechtsklick auf die Datei und den Befehl "Im Terminal ausführen" starten lassen. Und sie sollte vor allem ohne weitere Eingriffe deinerseits den Raspi zum Beacon machen, sobald der Erstgenannte hochgefahren wird.

## **Den gleichen Raspi zeitgesteuert in verschiedene Beacons verwandeln**

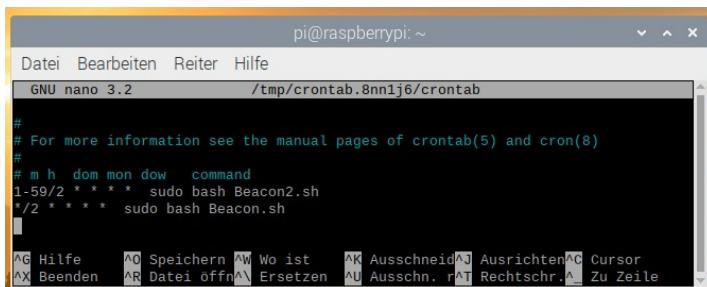
Denkbar ist es, dass die vom Beacon beworbene Website im Laufe des Tages geändert werden soll. Beispielsweise könnte von 6 bis 17 Uhr "leifiphysik.de" für die Arbeit angeboten werden und anschließend "speyer.de" für die Stadt mit ihren Freizeitangeboten.

Für solche Änderung lässt sich der Raspi programmieren. Der Befehl für die das Einstellen von "speyer.de" wurde bereits vorgestellt. Er wird durch das Ausführen der Shell-Datei "Beacon.sh" im Ordner /home/pi erledigt. Für Leifiphysik erstellen wir die Shell-Datei "Beacon2.sh" mit den entsprechenden Inhalten und Berechtigungen ebenfalls in /home/pi.

Die einzige Änderungen gegenüber Beacon.sh ist - außer dem neuen Dateinamen - der Inhalt der dritten Zeile:

```
sudo hcitool -i hci0 cmd 0x08 0x0008 1c 02 01 06 03 03 aa fe 14 16 aa fe 10 00 03 6c 65 69 66 69 70 68 79 73 69 6b 2e 64 65
```

Jetzt können wir mit dem Konsolen-Befehl: **sudo crontab -e** eine Tabelle mit Zeiten und Befehlen, "crontab" genannt, anlegen. Die Regeln für den Aufbau von crontabs findet man im Netz ausführlich beschrieben. Wir zeigen hier nur eine, die jede Minute den Raspi als Beacon zwischen "speyer.de" und "leifiphysik.de" umprogrammiert.



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window contains a terminal session where the user is editing a crontab file using the nano text editor. The file path shown is "/tmp/crontab.8nn1j6/crontab". The content of the file is as follows:

```
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
1-59/2 * * * * sudo bash Beacon2.sh  
*/2 * * * * sudo bash Beacon.sh
```

At the bottom of the terminal window, there is a menu bar with options like Datei, Bearbeiten, Reiter, Hilfe, and a toolbar with icons for Help, Save, Find, Cut, Paste, Undo, Redo, and others. The status bar at the bottom shows keyboard shortcuts for various functions.

Die drei einzigen nicht auskommentierten Zeilen lauten:

```
1-59/2 * * * * sudo bash Beacon2.sh
```

```
*/2 * * * * sudo bash Beacon.sh
```

Dabei darf die dritte Zeile als Leerzeile nicht vergessen werden!

[View Gallery](#)

31.2k

[18](#)

[2.6k](#)

[22](#)

## Team (1)

-  [Yamir Encarnacion](#)

[Join this project's team](#)

- 
- [eddyStone-URL-calculator](#)
- [Raspberry Pi 3 as an Eddystone URL beacon](#)

[completed project](#)

[Software](#)

[raspberry pi BLUETOOTH LOW POWER beacon EDDYSTONE URL BEACON](#)

## This project is submitted for

- [Hackaday.com Tip Line](#)

This project was created on 03/16/2016 and last updated 3 years ago.

## Description

Detailed instructions of how to turn your Raspberry Pi 3 into a Eddystone-URL beacon can be found at  
<http://webgazer.org/update/tutorial/2016/03/16/raspberrypi-eddyStone-url.html>

## Details

Raspberry Pi 3 as an Eddystone URL beacon

Mar 16, 2016

This tutorial will show you how you can take your Raspberry Pi 3 and turn it into an Eddystone URL beacon.

## What you will need:

- A Raspberry Pi 3
- Power supply for the Raspberry Pi
- SD card for the raspberry pi

# What is an Eddystone beacon:

Eddystone is a protocol specification by Google that allows a Bluetooth low energy device to broadcast one way messages. See <https://github.com/google/eddyStone>. Currently the specification defines three types of messages that can be broadcast: a UID, a URL, or telemetry.

The magic of the Eddystone beacon is on the app side where your phone listens for these broadcast messages and either displays an alert when it detects something or performs some kind of action.

In this tutorial I will show you how to setup your Raspberry Pi 3 to broadcast a URL.

## Setting up the Pi

1. Download Raspbian from <https://www.raspberrypi.org/downloads/>
2. Install the Raspbian Image
  - See <https://www.raspberrypi.org/documentation/installation/installing-images/>
3. Find the ip of the raspberry pi with from your laptop/desktop
  - You can try the Raspberry Pi Finder.  
See <http://lifehacker.com/the-raspberry-pi-finder-easily-locates-your-pis-ip-addr-1702081021>  
or With nmap (you can install nmap with macports on OSX)  
\$ sudo nmap -sP 192.168.2.0/24  
or  
\$ arp -a | grep "b8:27"
4. Log into the Pi (password is raspberry)
  - \$ ssh pi@<the\_ip\_of\_your\_pi>
5. Look at the help of the hciconfig command
  - \$hciconfig -h
6. Enable the Bluetooth device
  - pi@raspberrypi:~ \$ sudo hciconfig hci0 up
7. Set the Bluetooth device to “advertise and not-connectable”
  - pi@raspberrypi:~ \$ sudo hciconfig hci0 leadv 3
8. Enter the Beacon Advertising Data
  - pi@raspberrypi:~ \$ sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe 0f 16 aa fe 10 00 03 77 65 62 67 61 7a 65 72 08 00 00 00 00 00 00 00 00
  - 73 s
  - 70 p
  - 65 e
  - 79 y
  - 65 e
  - 72 r
  - 2e .
  - 64 d

- 65 e

```
sudo hcitool -i hci0 cmd 0x08 0x0008 17 02 01 06 03 03 aa fe 0f 16 aa fe 10 00 03 73 70 65 79 65 72 2e 64 65
```

Here is a breakdown of the payload

<b>Option</b>	<b>Description</b>
0x08	#OGF = Operation Group Field = Bluetooth Command Group = 0x08
0x0008	#OCF = Operation Command Field = HCI_LE_Set_Advertising_Data = 0x0008
17	Length. The hexadecimal 17 converts to 23 decimal which is the number of bytes that follow
02	Length
01	Flags data type value
06	Flags data
03	Length
03	Complete list of 16-bit Service UUIDs data type value
aa	16-bit Eddystone UUID
fe	16-bit Eddystone UUID
0f	Length. The hexadecimal 0f converts to 15 decimal which is the number of bytes that follow
16	Service Data data type value
aa	16-bit Eddystone UUID
fe	16-bit Eddystone UUID
10	Frame Type = URL
00	TX Power (this should be calibrated)
02 changed to 03	URL Scheme (http:// = 0x02)(https:// = 0x03). See 2016-13-09 edit comments.
77	'w' in hexadecimal
65	'e' in hexadecimal
62	'b' in hexadecimal
67	'g' in hexadecimal
61	'a' in hexadecimal
7a	'z' in hexadecimal
65	'e' in hexadecimal
72	'r' in hexadecimal
08	.org (.org = 0x08)
00	
00	
00	
00	
00	
00	

The command above broadcasts my blog's URL <https://webgazer.org>.

If you want to advertize a different URL enter the URL of the link that you want to advertize in the link below.

<http://yencarnacion.github.io/eddystone-url-calculator/>

To detect the Raspberry Pi beacon with an iPhone follow the steps in the video below which outlines how to enable Chrome's Physical Web extension on iOS [https://www.youtube.com/watch?v=gxPcPXSE\\_O0](https://www.youtube.com/watch?v=gxPcPXSE_O0)

On Android, your phone should detect the URL if you have Android 4.3.2 or higher with bluetooth turned on, location turned on, and Chrome location runtime permission turned on.

See <https://support.google.com/chrome/answer/6239299?hl=en>. However, I had to install the Physical Web App from [https://play.google.com/store/apps/details?id=physical\\_web.org.physicalweb&hl=en](https://play.google.com/store/apps/details?id=physical_web.org.physicalweb&hl=en) to make it work.

---

Edited 2016-13-09

As pointed out in the comments, Google made a change so as to require https in order for chrome to detect the URL Beacon. So in order to fix a change was made in the URL Scheme from http:// = 0x02 to https:// = 0x03.

<b>Plus Sign</b>	+	Shift =	43	00101011	2B
<b>Comma</b>	,	,	44	00101100	2C
<b>Hyphen / Minus Sign</b>	-	-	45	00101101	2D
<b>Period</b>	.	.	46	00101110	2E
<b>Forward Slash</b>	/	/	47	00101111	2F
<b>Zero Digit</b>	0	0	48	00110000	30
<b>One Digit</b>	1	1	49	00110001	31
<b>Two Digit</b>	2	2	50	00110010	32
<b>Three Digit</b>	3	3	51	00110011	33
<b>Four Digit</b>	4	4	52	00110100	34
<b>Five Digit</b>	5	5	53	00110101	35
<b>Six Digit</b>	6	6	54	00110110	36
<b>Seven Digit</b>	7	7	55	00110111	37
<b>Eight Digit</b>	8	8	56	00111000	38
<b>Nine Digit</b>	9	9	57	00111001	39

<b>Colon</b>	:	Shift ;	58	00111010	3A
<b>Semicolon</b>	;	;	59	00111011	3B
<b>Less-Than Sign</b>	<	Shift ,	60	00111100	3C
<b>Equals Sign</b>	=	=	61	00111101	3D
<b>Greater-Than Sign</b>	>	Shift .	62	00111110	3E
<b>Question Mark</b>	?	Shift /	63	00111111	3F
<b>At Sign</b>	@	Shift 2	64	01000000	40
<b>Capital A</b>	A	Shift A	65	01000001	41
<b>Capital B</b>	B	Shift B	66	01000010	42
<b>Capital C</b>	C	Shift C	67	01000011	43
<b>Capital D</b>	D	Shift D	68	01000100	44
<b>Capital E</b>	E	Shift E	69	01000101	45
<b>Capital F</b>	F	Shift F	70	01000110	46
<b>Capital G</b>	G	Shift G	71	01000111	47
<b>Capital H</b>	H	Shift H	72	01001000	48
<b>Capital I</b>	I	Shift I	73	01001001	49
<b>Capital J</b>	J	Shift J	74	01001010	4A
<b>Capital K</b>	K	Shift K	75	01001011	4B
<b>Capital L</b>	L	Shift L	76	01001100	4C
<b>Capital M</b>	M	Shift M	77	01001101	4D
<b>Capital N</b>	N	Shift N	78	01001110	4E
<b>Capital O</b>	O	Shift O	79	01001111	4F
<b>Capital P</b>	P	Shift P	80	01010000	50
<b>Capital Q</b>	Q	Shift Q	81	01010001	51

<b>Capital R</b>	R	Shift R	82	01010010	52
<b>Capital S</b>	S	Shift S	83	01010011	53
<b>Capital T</b>	T	Shift T	84	01010100	54
<b>Capital U</b>	U	Shift U	85	01010101	55
<b>Capital V</b>	V	Shift V	86	01010110	56
<b>Capital W</b>	W	Shift W	87	01010111	57
<b>Capital X</b>	X	Shift X	88	01011000	58
<b>Capital Y</b>	Y	Shift Y	89	01011001	59
<b>Capital Z</b>	Z	Shift Z	90	01011010	5A
<b>Left Bracket</b>	[	[	91	01011011	5B
<b>Backward Slash</b>	\	\	92	01011100	5C
<b>Right Bracket</b>	]	]	93	01011101	5D
<b>Caret</b>	^	Shift 6	94	01011110	5E
<b>Underscore</b>	_	Shift -	95	01011111	5F
<b>Back Quote</b>	`	`	96	01100000	60
<b>Lower-case A</b>	a	A	97	01100001	61
<b>Lower-case B</b>	b	B	98	01100010	62
<b>Lower-case C</b>	c	C	99	01100011	63
<b>Lower-case D</b>	d	D	100	01100100	64
<b>Lower-case E</b>	e	E	101	01100101	65
<b>Lower-case F</b>	f	F	102	01100110	66
<b>Lower-case G</b>	g	G	103	01100111	67
<b>Lower-case H</b>	h	H	104	01101000	68
<b>Lower-case I</b>	I	I	105	01101001	69

<b>Lower-case J</b>	j	J	106	01101010	6A
<b>Lower-case K</b>	k	K	107	01101011	6B
<b>Lower-case L</b>	l	L	108	01101100	6C
<b>Lower-case M</b>	m	M	109	01101101	6D
<b>Lower-case N</b>	n	N	110	01101110	6E
<b>Lower-case O</b>	o	O	111	01101111	6F
<b>Lower-case P</b>	p	P	112	01110000	70
<b>Lower-case Q</b>	q	Q	113	01110001	71
<b>Lower-case R</b>	r	R	114	01110010	72
<b>Lower-case S</b>	s	S	115	01110011	73
<b>Lower-case T</b>	t	T	116	01110100	74
<b>Lower-case U</b>	u	U	117	01110101	75
<b>Lower-case V</b>	v	V	118	01110110	76
<b>Lower-case W</b>	w	W	119	01110111	77
<b>Lower-case X</b>	x	X	120	01111000	78
<b>Lower-case Y</b>	y	Y	121	01111001	79
<b>Lower-case Z</b>	z	Z	122	01111010	7A