K47 NEWS

10.05.2020

Was ist COTA?

Wer als Funkamateur auf Kurzwelle aktiv ist, wird schon mal der ein oder andere Station, z.B. aus DL begegnet sein, die nicht unbedingt als seltene DX-Station zu bezeichnen war, aber trotzdem ein PileUp generiert hat, wo man sich fragt… … warum? Und dann hat diese Station mit dem alltäglichen Rufzeichen komische Abkürzungen verwendet, wie Aktivität von DL-04404, oder Referenznummer RPB 399, oder Vy 11…

COTA steht für Castles On The Air und gehört zum Programm WCA = World Castle Activity bzw. World Castle Award. COTA ist damit quasi der deutsche Ableger des weltweiten Burgen-programms.

COTA: http://logsearch.cotagroup.org/index.php

WCA: http://wcagroup.org/

Dabei geht es darum, Funkaktivitäten direkt von Burgen oder Schlössern oder aus der näheren Umgebung durchzuführen und dabei diese Burgen und Schlösser zu aktivieren/in die Luft zu bringen, wie man als Funkamateur halt sagt.

Bedingung für so eine Aktivierung ist, dass

a) man sich entweder auf dieser Burg befindet oder die Aktivität (fest oder portabel) aus einer maximalen Entfernung von 1000 m um diese Burg herum stattfindet,
b) die Burg oder das Schloss eine offizielle, weltweit einmalige Referenznummer hat.

Viele deutsche Burgen und Schlösser sind bereits registriert. Dies kann man entweder bei COTA: http://logsearch.cotagroup.org/index.php nachschlagen oder sich komfortabler über http://www.cotagroup.org/cotagroup/map/?show=wca (Kartenmaterial) informieren. Hier kann man dann auch sehen, welche Burg oder welches Schloss noch nie aktiviert wurde.

Hier sucht man sich das Bundesland heraus und schon füllt sich die Karte.

Dabei ist z.B. DL-00899 die Referenznummer der Burg Eltz und RPB-220 die Referenznummer in Rheinland-Pfalz. DL steht für Deutschland, RPB für Rheinland-Pfalz.

OV-Treff
Kath. Familienbildungsstätte Mayen e.V.
Bäckerstr. 12, 56727 Mayen

OVV
Telefon. 02605/1808

Email: DD5WW@DARC.DE
Redaktion: DD8VX@darc.de

Das sind also die seltsamen Nummern, die man austauscht. Und der Abschiedsgruß 73 ist ja bekannt und die COTA-Akteure sagen dann noch „… und die 11“ (Gruß der COTA-Akteure).

Man unterscheidet dabei zwei Arten von Aktivisten, die Aktivierer und die Hunter (die Sammler). Die Aktivierer verteilen die Punkte/Referenznummern, die Hunter sammeln diese Nummern.

Hat man eine bestimmte Anzahl Referenznummern gesammelt, getrennt nach Aktivierer oder Hunter erhält man dann auf Antrag entsprechende Diplome.

Beispiel:

Weiterhin gibt es feste Veranstaltungen z.B. am 1. Mai in Deutschland oder auch Mitte Juni im Rahmen der weltweiten Aktivitäten.

Also Diplomsammler, Wanderfreunde oder Portabelfreunde… auf geht’s...

73 & 11, de Matthias, DD8VX

CQ DL 5/2020 – Flexible UTC-UHR – Exklusive Fortsetzung für Mitglieder von K47

Dem ein oder anderen ist vielleicht mein Beitrag in der letzten CQ-DL aufgefallen.

SOFT & HARD

Basis: 7"-TFT-LCD-Car-Monitor

Flexible UTC-Uhr

Matthias Follmann, DD8VX

Vorgestellt wird ein Selbstbauprojekt, das nicht nur Datum und Uhrzeit, sondern auch weitere Informationen (Frequenz des Transceivers, Temperaturen etc.) anzeigen kann.

TFT color monitor
UTC: 18:09:08
MESZ Temp Shack : 23.5 °C Temp Aussen : 13.2 °C
Datum: 20.10.2019
FT-857D: 7.087 MHz
DL-04404
RPB 399
DD8VX

Vor ein paar Jahren habe ich, durch das Raspberry-Pi-Fieber angesteckt, einige Projekte mit diesem Rechner realisiert. Auch mit den ESP8266- und Arduino-Modulen habe ich einige Erfahrungen gesammelt und inzwischen mein Haus, sowie meinen Garten „automatisiert" und vernetzt.

Uhr fürs Shack

Kürzlich kam mir die Idee, für mein Shack auch eine UTC-Uhr zu realisieren. Bei der Suche in meinem Vorrat nach Bauteilen ist mir dann wieder der 7"-TFT-LCD-Car-Monitor mit Farb-Display in die Hände gefallen. Das sollte das Display meiner neuen UTC-Uhr werden.

Da als zentrale Komponente ein Raspberry bei mir bereits mit einem Apache Webserver läuft (Einrichtung z.B. unter [1]), sollte es ein php-Skript werden, das, regelmäßig aktualisiert, nicht nur Datum und Uhrzeit, sondern auch weitere Informationen (Frequenz des Transceivers, Temperaturen etc.) anzeigen kann. Wie man mit dem Raspberry Pi den Trx anzapft (CAT-Schnittstelle) und die Frequenz ausliest, kann man in der CQ DL 6/2013 [2] in dem Artikel „Web-basierte Steuerung des Transceivers" nachlesen. Dies habe ich mir damals zunutze gemacht und als ersten Part die CAT-Schnittstelle realisiert. Die Daten daraus wurden für APRS aufbereitet [3]. Auch habe ich am Raspberry Pi bereits verschiedene Temperatursensoren angeschlossen. Es handelt sich hierbei um die 1-Wire-Schaltung mit dem DS18B20. Hierzu gibt es im Internet jede Menge Anleitungen. Somit war bis auf ein kleines Skript alles vorhanden.

Das Programm

Ich setze natürlich entsprechende Programmiersprachen für den Webserver sowie html und php voraus und beschränke mich auf die wichtigsten Funktionen.

Das Skript läuft auf dem Raspberry, der als Webserver und mySQL-Datenbank dient und somit sind ja dort alle benötigten Programmteile, bzw. Daten/Messwerte vorhanden.

Hier nun die Skript-Bestandteile für die Darstellung der Uhr (**Bild 1**):

```
<?php
```

Zuerst das Aussehen (Hintergrundfarbe und ein positioniertes Bild meiner QSL-Karte).

```
echo "<body style="background-color:#0000FF">";
echo "<img src="/Images/dd8vx.jpg
style="width:200px; height:140px;
position:absolute; left:440px; top:200px;
">";
```

Dann wird im folgenden Abschnitt das aktuelle Datum und die Greenwich Mean Time erzeugt.

```
# Datum und Uhrzeit
$timestamp = time();
$Datum = date("d.m.Y",$timestamp);
$Uhrzeit = gmdate("H:i:s",$timestamp);
```

Danach ermittelt man noch die aktuelle Zeitzone.

```
$Zeitzone = date("_T");
```

Die Zeitzone soll auch noch in MESZ oder MEZ „übersetzt" werden.
Dazu dient dann die if.. then.. else.. Funktion.

```
if ($Zeitzone == "CEST") {
    $Zone = "MESZ";
} else {
    $Zone = „MEZ";
}
```

Nun holen wir die verteilten Daten und Messwerte, die an verschiedenen Orten auf dem Raspberry Pi bereits vorliegen und von anderen Skripten regelmäßig aktualisiert werden.
Die folgende Zeile startet ein Shell-Skript, welches die Trx-Frequenz im Verzeichnis /home/pi/APRS/ausgelesene_Freq.txt ablegt und zwar in der Datei ausgelesene_Freq.txt.

30 CQ DL 5-2020

SOFT & HARD

Zur Person
Matthias Follmann, DD8VX
Jahrgang 1966
Amateurfunkgenehmigung seit 1995
Anschrift: matthias.follmann@t-online.de

```
exec(. /home/pi/APRS/Frequenz_auslesen.sh.);
```

Das Skript selbst habe ich hier nicht beschrieben, es ähnelt sich aber aus [3]. Da unser php-Skript ja in kurzen Abständen aktualisiert werden soll, erhalten wir so in kurzen Abständen die jeweilige aktuelle Transceiver-Frequenz.
Die folgende Zeile liest dann die erzeugte Datei aus und in der Variablen *$Frequenz* haben wir so immer die aktuelle QRG.

```
$Frequenz = file_get_contents("/home/pi/APRS/ausgelesene_Freq.txt");
```

Nun holen wir uns analog die weiteren Daten, z.B. die Temperaturwerte.

```
# Temperaturen von den Sensoren holen
$Aussen_temp = file_get_contents("/home/pi/temp5.txt");
```

usw.
./home/pi/temp5.txt ist natürlich durch den Pfad und die Datei auf dem eigenen Raspberry zu ersetzen. Nun muss das Ganze nur noch dargestellt werden. Hier empfiehlt es sich, mit php eine Tabelle zu erzeugen, in die man die Texte und Variablen einbettet und somit sehr leicht formatieren, positionieren und gestalten kann. Den Umgang mit html-/php-Tabellen sollte man kennen oder im Internet nachlesen. Hier nur ein Grundgerüst:

```
echo „<table border='0' style='nach Belieben'>";
```

„nach Belieben" ist auszugestalten, z.B.:

```
width:720px; position:absolute; left:15px;
top:15px; font-size:76px; color:#FFFFFF
```

oder ähnlich, bitte ggf. im Internet recherchieren.

```
echo „<tr><td>UTC: $Uhrzeit</td></tr>";
echo „<tr><td>$Zone Temp Shack :
$Shack_temp_neu &deg;C Temp Aussen :
$Aussen_temp &deg;C</td></tr>";

echo „<tr><td>Datum:</td></tr>"
...
echo „</table>";
```

Damit ist das Design abgeschlossen.
Mit dem letzten Block wird nun der automatische Aufruf unserer Seite (bei mir *UTC-Uhr.php*) realisiert. In der Variablen *$sec* können auch andere Zeiträume in Sekunden eingetragen werden. Das geht dann halt schneller oder etwas langsamer, bis die neuen Daten vorliegen. Ein zu kurzer Abstand macht aber keinen Sinn.

```
$page = $_SERVER['PHP_SELF'];
$sec = „10";
header("Refresh: $sec; url=$page");
```

... und php beenden

```
?>
```

Auf meinem Raspberry Pi läuft der Midori-Browser im Kiosk-Modus [4]. Somit startet der Browser bei jedem Neustart des Raspberry automatisch und zeigt im angeschlossenen TFT-Bildschirm im Vollbildmodus die UTC-Uhr. Das **Aufmacherfoto** zeigt das Endergebnis.
Ich erhebe keinen Anspruch auf Vollständigkeit und Fehlerfreiheit, hoffe aber, einige Anreize geschaffen und das ein oder andere erfolgreiche Projekt initiiert zu haben. Insbesondere ist das Ganze dann eine günstige und flexible Lösung, wenn man schon einen Raspberry Pi in Betrieb hat.

CQ DL

Literatur und Bezugsquellen
[1] https://tutorials-raspberrypi.de/webserver-installation-apache2
[2] CQ DL 6/2013, www.darc.de
[3] CQ DL 5/2016, www.darc.de
[4] https://maker-tutorials.com/autostart-midori-browser-vollbild-kiosk-mode-via-konsole-ohne-desktop

```
<?php
echo "<body style="background-color:#0000FF">";
echo "<img src="/Images/dd8vx.jpg style="width:200px; height:140px; position:absolute; left:440px; top:200px; ">";

# Datum und Uhrzeit
$timestamp = time();
$Datum = date("d.m.Y",$timestamp);
$Uhrzeit = gmdate("H:i:s",$timestamp);

# Zeitzone ermitteln
$Zeitzone = date("_T");
if ($Zeitzone == "CEST") {
    $Zone = MESZ;
} else {
    $Zone = MEZ;
}

# Aus den Transceiver die aktuelle Frequenz auslesen
exec(. /home/pi/APRS/Frequenz_auslesen.sh.);
$Frequenz = file_get_contents("/home/pi/APRS/ausgelesene_Freq.txt");

# Temperaturen von den Sensoren holen
$Shack_temp_neu = file_get_contents("/home/pi/temp2.txt");
$Aussen_temp = file_get_contents("/home/pi/temp5.txt");
```

Bild 1: Skript-Bestandteile für die Darstellung der Uhr

CQ DL 5-2020 31

Eine Fortsetzung wird es bereits in der nächsten CQ-DL geben.

Aber, wie das so mit den Projekten ist, sie stehen niemals still. Inzwischen gibt es bereits weitere Ideen für Anwendungszwecke bzw. Inhalte.

Exklusive zeige ich hier die aktuelle Version mit ein paar Highlights, die ich auch noch näher beschreiben möchte.

Neben den bekannten Funktionalitäten aus CQ DL 05/200 und 06/2020 (Tuner-Daten), habe ich auch ein kleines Hilfsmittel gebaut, welches mir den Betrieb über den QO-100 erleichtert.

Da ich bekannterweise aus dem 70cm-Band sende, welches vom UpConverter dann auf 2,4 Ghz umgesetzt wird, wollte ich ein einfaches Hilfsmittel, dass mir dann direkt beim Einstellen der Sendefrequenz, die Down Link Frequenz anzeigt. Das erleichtert auch das Drehen auf Frequenzen von CQ rufenden Stationen.

Hierzu habe ich das Skript etwas ergänzt.

Die auf dem Trx eingestellte QRG wird unter 1 angezeigt und in die Down Link Frequenz des Satelliten umgesetzt und erscheint dann unter 2.

Damit ich nicht vergesse das Netzteil des Converters ein bzw. auszuschalten gibt es noch den Punkt 3 . Der grüne Punkt erscheint, wenn das Netzteil an ist, ein roter Punkt, wenn das Netzteil aus ist.

Ach ja, auch nur, wenn ich Satellitenbetrieb mache, bei normalen 70cm Betrieb erscheint das alles nicht….