

Beacon Software Script

C. W. Andreasen – K7CWA

One thing some Amateur Radio operators do is to put a low power beacon transmitter on the air and other stations listen to figure out if a band is open to various locations. One example is on Six Meters where a band opening can be missed because everybody is listening and nobody is talking. Beacons reside grouped near each other near the bottom of the band, and the location and frequency is published and available on the Internet.

With a small investment one can build a beacon but a number of things are needed; a low power transmitter and antenna, a control unit, and a power supply.

This article covers only the control, and expects the Ham to have basic skills with tools and computers and have the technical ability to do the rest.

The code I am presenting here runs on any model of Raspberry Pi... I have run it on the Pi4, Pi3, and Pi-zero. It is written as a bash script, which is universal in the Linux/Raspberry pi world. The Pi operating system is Linux Debian. We do not use the desktop (GUI) and all is done in the console (Terminal) mode.

Included with the download is a file named READ_ME and that gives a lot of detail, but the bottom line is once you put in the message (any length), it will continuously loop repeating the message in CW keying of the transmitter.

The program is flexible and can key the relay, or not, and can also send the code to the screen, or not, in the form of “(A) dit dah”, for testing. Of course when deployed there will be no keyboard or monitor so the display is disabled.

Another cool thing is the Pi's have built-in WiFi so one can use ssh to connect over the WiFi and use a remote computer as keyboard and screen, if WiFi is available.

What is required is the ssh interface must be enabled in the Pi and this can be done with the console command “sudo raspi-config” then using the menus enable ssh. When back in the console mode, do the command “hostname -I” to get your IP address (like ‘192.168.1.29’), then from the computer you want to work from, in console mode, enter ‘ssh [logon name] @ [IP address]’ and the first time it will want to store your fingerprint (computers), say “yes” and it won't ask again. It will ask you for your password each time you connect.

If not set up to auto-start, like for testing, start the script with “bash blink.sh”.

To view/download the files, use the following links:

https://qso.com/k7cwa/blink/READ_ME

https://qso.com/k7cwa/blink/blink_text

<https://qso.com/k7cwa/blink/blink.sh>

Note: the apparent blanks before ME and text, are not blanks, but underscore (_)