The Latest

Beacon FM-CW

Up

Posted by AG6QV Frank
Tags:

In the previous post I described how to make a HAM Radio beakon that could work with both CW and FM modulation. As I was testing the results of that I noticed an issue with the FM modulation as it would also modulate the carrier. That was not the intention, but it did function as an FM beacon. In this updated design I have utilized two Selector Blocks to switch the morse code signal either directly to the HackRF One or to go through the FM modulation stages to provide a constant tone.

I also updated the GUI to have a slider to set the CW speed between 5 and 20 words per minute with a default of 12 and added an input box to change the message used to generate the CW code. The screenshot below was created from running the Python code after manually adding the additional frequencies.

The two Python files are available for download: Beacon.py and Beacon_epy_block_0.py as well as the flow graph: Beacon.grc.

Link to this Post

HackRF One - Beacon update

Up

Posted by AG6QV Frank
Tags:

I have made 3 different versions of the HAM Radio beacon project using GNU Radio and my HackRF One SDR hardware. The first version, described in the previous previous post and covered 5 HAM bands and used a QT GUI to select the beacon frequency. The second version eliminated the GUI and automatically switched between the 5 frequencies in the VHF and UHF bands. Since there was one additional HAM band I edited the Python file directly to add the 6th band. The 3rd version was similar to the second version but cover the 5 bands from 23cm to 3cm. Since 3cm or 10GHz is above the max frequency the HackRF One can operate on this would have to rely on a harmonic of a frequency below 6GHz.

Over the weekend the PNW Microwave group met to discuss SDR projects and work on other beacons and I got a few suggestion for improvements, especially to the version with a GUI as that will allow for more interactive control of the beacon.

The first suggestion was to increase the output power. The HackRF One is controlled by a "Soapy" GNU Radio block that provides input for center frequency, bandwidth, CGA gain and to turn the output amplifier on/off. In my original version I had left the amplifier off. Turning it on would make it easier to receive the harmonics on 10 GHz. A final version would need some filtering to avoid transmitting on the base frequency, that would be outside of the HAM bands. It will also be necessary to add a bit more power to make the beacon useful, but more about that in a future post.

Another suggestion was to allow the user to switch between CW and FM modulation. The beacon is design to send morse code and when listening to the beacon on an FM receiver it is difficult to hear the actual morse code if the beacon is transmitting a carrier. Implementing a system to generate a FM modulated signal and keep the carrier on while turning the modulation on and off would make it possible to use a FM receiver.

Today I started to implement the additional functionality. There was a few experiments to find the correct blocks to use and to ensure the selection of modulation would make the system work for both CW and FM. The flow graph is shown on the image below.

You can download the GNU Companion flowgraph file here.

The GUI provides fixed variables for audio, RF sampling rate, drive for the CW mode and the message that will be translated into morse code. There are currently no QUI elements to changes these interactively. The GUI does provide elements for changing the frequency, switching between CW and FM and to change the frequency used for the tone in FM modulation. The basic functionality uses a custom Python block named "Beacon Block". This is the code that can transform a string of characters into morse code. The output of this block is fed into two multipliers. Since the output is switching between 0 and 1 the effect will be to have the multiplier turning the signals on and off. The output of the multipliers are then connected to the selector, controlled by the selected modulation. The selected signal will then be fed into the Soapy HackRF sink and also to a FFT Frequency display to visually show the output.

The image below is taken from the version where I edited the Python file to provide beacon frequencies for all the HAM bands from 60 MHz to 10 GHz. The frequencies used is .080 on 6m and 33cm, .280 on all other bands to 23cm and .380 on all bands above 23cm. That way the frequencies all fall within the allocated beacon sub bands according to ARRL's band plan.

Finally the two Python files that were generated by GNU Radio Companion, and edited to allow more than frequencies can be downloaded from these links: Beacon.py and Beacon_epy_block_0.py. Beacon.py is the main file that includes the code to the custom block to handle the translation of characters to morse code. The two files should be located in the same directory.

Link to this Post

HackRF One - 5 band beacon

Up

Posted by AG6QV Frank
Tags: 2m | GNU Radio | HackRF One

Radio Beacons are used in HAM radio to test propagation. If you are able to receive a beacon on a given frequency you might also be able to make contacts with stations in the same area as the beacon, and on frequencies close by. Beacons can also be used to test antenna and receiver performances. Setting up test beacons for one or many HAM radio bands can be a large task. With HackRF One and Software Defined Radio (SDR) it can be a much easier task.

I have created a simple flowgraph that can select between 5 preconfigured frequencies on 6m, 2m, 70cm, 33cm, and 23cm. The flowgraph also includes a variable to define the beacon message. The message is a series of characters that will be translated into morse code when the program is launched.

Translation into morse code required the use of a custom block in GNU Radio Companion. Custom blocks can be written in Python and are relatively simple to get custom functionality into the flowgraph.

The image below shows the flowgraph with variables on the top and the logic at the bottom. I included a frequency sink to show where the signal is when the different frequencies are selected and to visualize the CW code.

The next image is a screenshot of the application running. The maximum output of HackRF One is 10-15dbm or 10-25mW so a small amplifier and an external antenna might help. In addition a valid HAM radio license is required to transmit any signal in the HAM bands.

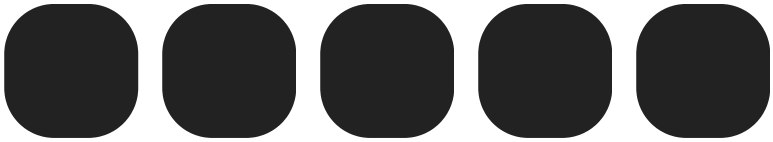
Download the flowchart.

Update:

I made a new version of the flowchart for the multiband beacon. It now operates on 6 bands: 6m, 2m, 1.25m, 70cm, 33cm, 23cm. It can work on any number of bands/frequencies where each frequency is defined as an element in an array within the custom block that generates the CW code. It will automatically switch to the next

band after completing a full transmission of the CW message.

Link to this Post



Previous 3Get Next 3

Get RSS feed

Get notified via email when new posts are published.

Sign Up

Recent Blog Posts

Blog Archives

Tags

10 GHz {2}

2m {3}

GNU Radio {5}

HackRF One {4}

HAM {7}

HF {2}

PNW Microwave {1}

X-Band {1}

Calendar

April 2025

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			