The Latest

Grid Square Finder

Up

Posted by AG6QV Frank
Tags:

When operating microwave radios in the field it is common to exchange Maidenhead grid squares that allow us to calculate the distance between the two stations. The Maidenhead grid square is a system of letters and numbers corresponding to a specific location. The larges squares are identified with two letters from A-R. The squares are 20x10 degrees. Adding two numbers makes the grid size 2x1 degrees. It is common to use 4 characters for most contacts on FT8, but for microwave contacts both stations might be in the same 4 character grid so we add more characters to be more precise with the location.

Grid squares can be calculated from the coordinates provided by a GPS receiver, but doing the calculations in the field is not practical, and you might not know the exact location you will be operating from ahead of time so I created a small device using an Arduino Nano, a GPS module, a LCD display and a 3D printed box to bring with the radio. The system will show the 10 character grid square, number of satellites and alternating date and time (UTC).

I found a small GPS module on Amazon based on the GT-U7 chip. It came in a two pack.

I mounted the Arduino Nano on an expansion board that will allow for both 12V and USB power options. These came in a package of 4 or 6.

The enclosure was created with TinkerCadÆ

The Arduino code is a combination of functionality found on GitHub and will work with Arduino Nano, Uno, Pro Mini and many other versions. When testing the system it can take a while for the GPS unit to lock on to satellites, especially indoors. I have had the unit sitting in the window sill for a couple of days and it constantly see 5-9 satellites. It will be interesting to see how fast it can lock in the field.

A view inside the assembled unit:

The GPS antenna is attached to the side of the box qith double sided tape.

And from the outside with power and GPS signal locked.

I added a button on the side to switch through 4, 6, 8 and 10 characters in the grid square. The code defaults to 8. The small size of the squares when 10 digits are shown makes it difficult to get a stable reading. Moving the box around will cause the last two characters to jump plus/minus on grid square in both directions, but with 10 characters each grid is about 20x20 meters.

The Arduino sketch can be downloaded here and the stl file for the box here.

Link to this Post

Microwave step attenuator

Up

Posted by AG6QV Frank
Tags: 10 GHz | GNU Radio

While searching through eBay listings I came across a step attenuator that was listed aa 0-50 dB in 10 dB steps attenuator, and good for DC to 18 GHz. It was recently priced so I decided to spring for it. When it arrived I started to research the origin. It turned out to be from an old HP 8555 spectrum analyzer. Most likely a unit no longer working, but the attenuator was advertised as tested. I found a users/service manual that described the operation and the voltages needed to activate the solenoids. It looked to be simple as it was using 12V solenoids. The attenuator has three solenoids that are used to enable/disable 3 different attenuators in series. After putting the attenuator on the test bench I confirmed that all 3 was working and providing 10, 20 and 40 dB attenuation respectively. So the range is 0-70 dB and not as the specs for the HP 8555 analyzer listed (0-50 db). In the spectrum analyzer the attenuator is placed between the N connector on the front panel and the mixer and can handle +33 dBm or 2W maximum. Good for many microwave applications.

With the information about how the attenuator is operated and the knowledge that all 3 stages was working as expected I started designing the control unit. The first step was to decide how to generate the voltage needed for each of the three solenoids. In order to engage the solenoid the two terminals needed a +12V pulse for 150 ms and in order to switch it back the polarization of the pulse should change ro -12V for 150 ms. This can be achieved with an H-bridge. This is basically 4 transistors configured to have 2 inputs and two outputs. This will allow a positive voltage when the input is high and low and a negative output when the inputs are low and high. In addition there would be no voltage across the outputs if both inputs are low. H-bridges are commonly known to drive DC motors, allowing them to turn in both directions depending on the voltage being positive or negative. Amazon is selling a package of 4 H-bridge modules based on the L298N chip for just $10. It would not make sense to try to build one from scratch at those prices. The module provides 5V input logic and 12V output, exactly what's needed to use an Arduino Nano as the controller.

With the parts ordered I designed a 3D printed enclosure that could be glued on to the side of the attenuator, covering the terminals for the solenoids and providing push buttons to change the attenuation up and down and a display to show the current value. Since the solenoids are latching it's possible to connect the power, set the desired attenuation and then disconnect the power. That way there will be no noise from the Arduino board while the attenuator is in use. I also made the code to set the default attenuation to 70 dB when the unit is powered up. This is needed to set the highest attenuation but also because it's not possible to read the current value/state of the each attenuator. Bringing them to a default value makes sense.

The finished attenuator and control box all assembled.

The inside of the control unit showing the two H-bridge boards on the left and the Arduino Nano on the right. I placed the Arduino nano on an breakout board that allow for 12V power input and easy access to the pins needed to control the H-bridges and the two input buttons.

And finally the unit with power applied. The small display is a 128*32 pixel OLED display with a I2C interface. When power is applied it will show my call sign and then set the attenuator to 70 dB.

Here are links to the Arduino sketch and the 3D stl file.

Link to this Post

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

Previous 1Get Next 3

Get RSS feed

Get notified via email when new posts are published.

Sign Up

Recent Blog Posts

Blog Archives

March 2025 {1}

January 2025 {2}

October 2024 {5}

March 2024 {1}

August 2023 {1}

May 2023 {1}

April 2023 {1}

March 2023 {1}

January 2023 {2}

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		