THE BEACON

West Mountain Radio

Quarter 3 2026

Articles of Interest

RIGblaster Plug & Play

Low Cost Data Port Interface

Simplest RIGblaster to plug in, built-in rig control, & fully isolated CW & FSK outputs but is only compatible with specific radio models

Getting the Most Out of the CLRdsp and CLRspkr

by: Sholto Fisher, K7TMG

Frequently I'm asked about our adaptive DSP audio products and how to use them effectively. I thought I'd share some information on what I've learned over the years in optimizing audio quality and noise reduction using them.

The first thing to remember when using any DSP filter for listening to SSB audio is to select a slow AGC setting on the radio. This helps a lot in preventing "pumping" of the signal.

AGC is a method used in nearly all radio receivers to help even-out the signal. It's a regulating circuit which tries to maintain a good signal amplitude at its output regardless of input amplitude. On many radios, you can set the AGC time constant which is the speed at which this circuit operates. Generally speaking for speech you'd want a slow AGC. A fast AGC is often used with digital modes and sometimes Morse Code but usually does not sound pleasant for SSB signals.

The next tip sounds counter-intuitive but it really works. Reduce your RF gain! This was standard advice in years past but seems to have largely been forgotten and perhaps not as important as it once was due to advances with modern radios, but it's still a good thing to remember. Quite often you can reduce your RF gain (say from 100% to 60%) and you may actually improve practical SNR by preventing nonlinear effects due to close-by signals or excessive RF gain. Used in combination with a slow AGC this often gives the best sounding audio. If you don't believe me try it!

A word about DSP functions in the radio. There is no harm trying to cascade audio filters. For instance, you could set the radio's DSP noise reduction to a minimal value and then use the variable DSP control on the ClearSpeech® products. This can often be very effective. The noise-blanker circuit (NB) in the radio can be useful for ignition based interference and can usually be left engaged all the time. As the ClearSpeech® algorithm implements an auto-notch function, you probably won't need this turned on in the radio itself but there's no harm in experimentation especially when there are multiple heterodynes in the signal.

For the most pleasant sounding audio, try using a wide filter selection on the radio. On many modern radios, you can extend the bandwidth up to 3.6KHz or even higher. Obviously, when there are a lot of close by signals you'll find a setting of 2.8KHz or even less optimal.

The filtering control on the ClearSpeech® products is continually variable. Quite often I've found the most interesting settings to be about 9 'o' clock for light filtering, 12 'o' clock for medium filtering and about 2 'o' clock for a heavy filtering "sweet spot". Going further than this can start to make the audio sound a bit "mechanical" as DSP artifacts can be triggered on large swings of the input signal. Reducing RF gain can help a lot in this case too.

I hope you'll take the time to experiment with the audio DSP products we make. Regardless of radio make and model they offer outstanding audio noise reduction for your listening pleasure!

For more information on our ClearSpeech® technology, point your browser to
https://www.westmountainradio.com/content.php?page=digital_process

ClearSpeech® is a registered trademark of West Mountain Radio.

Epic PWRgate and Network UPS Tools

by: Rod Repp, W9RNR

Shortly after obtaining my license, I started building out my base station capabilities. One radio, then another, and shortly I was building a shelving unit to hold them all. I came to the quick realization that getting power to each radio would be challenging using only the binding posts on the power supply.

On a net, someone had suggested looking up Powerpole® equipped power distribution blocks. This initially led me to the RIGrunner 4005. But I also stumbled across the Epic PWRgate, and the temptation to ensure I had radio capability through a power outage was too much to bear. Shortly, both were in my cart and heading to the shack.

In the years since, the RIGrunner 4005 was moved to the truck, and a RIGrunner 4008 sits in its place in the shack. One constant with the Epic PWRgate has been vendor firmware updates. If you have used one of the devices, you know that the update comes with the penalty of needing a Windows PC. That can be a problem when your primary PC is a Linux PC, but luckily I've been a virtualization administrator for many years and have some Windows Virtual Machines stashed away for just this purpose.

Previous to this update, I had been playing with all sorts of ESP32 and NRF devices for mesh networking. I had been connecting them to my Linux PC console and observing the output and behavior of these devices, so why not see what this Epic PWRgate would do. Well, to my surprise, I noticed data. The device was streaming data. Data that could be scraped and used within the Network UPS Tools (NUT) program that I was already monitoring my APC UPS's with and integrated with Home Assistant.

While I have written and edited programs, I would consider myself more of a systems administrator than a programmer. For some reason, I decided to give AI a try. I went through a few AI engines and pushed the task at hand to them. I finally ended up on Google Gemini and was excited to find that Gemini Flash was able to produce a C++ driver for the Epic PWRgate that would compile. I quickly configured NUT to use this new driver and created all the relevant entries and configurations needed (after many iterations back and forth with Gemini).

Soon, I had NUT picking up the status of the Epic PWRgate. I quickly tasked AI to assist in editing my UPS status card to bring in this new driver. Once Home Assistant had some data history, I did see that the code wasn't perfect. There were times when the driver was reporting no data. I described the

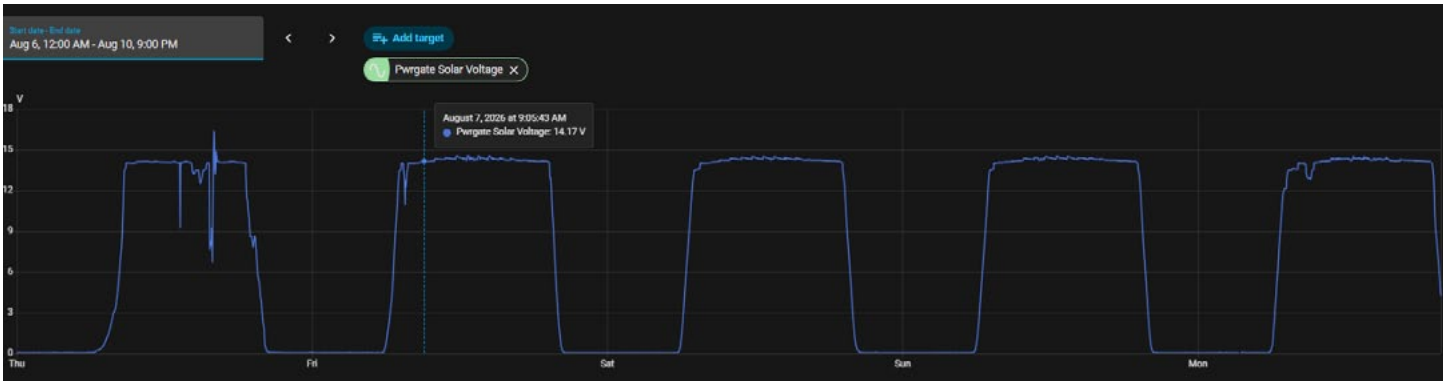
errors to Gemini and after some further troubleshooting, we had the correct string manipulation to prevent the data stalls from the text output of the Epic PWRgate.

After running this newer version for a few more days, the history was now clean and the drop outs had ceased. In writing the driver, I had tried to get every status to display on the console of the Epic PWRgate (on battery, no battery, and such forth). I had a suspicion that the device was capable of sending more than just the statuses I was seeing. I e-mailed West Mountain Radio support asking if they would inform me of all the statuses the Epic PWRgate might end up sending. I had very low expectations, but a few days later, I received a reply with the information I needed. I added the additional statuses into the code and recompiled. As usual, a few days of waiting and I still had good tracking and history.

However, there was one piece of this puzzle still missing. Those who own an Epic PWRgate are aware that there is a solar input as well as a power supply input. I have had some solar panels stored for some time that were intended to connect to this device. Finally with the right MC4 to Powerpole® connectors, I rigged a 100 watt panel into the south facing window of the shack. Yes, the panel isn't in the best orientation, but it is at least facing south.

When I added the panel, I went back to Home Assistant and my UPS card hoping to see the data being logged. It was then that I learned that I had chosen my solar input variable hastily. I checked the serial output, and the Epic PWRgate was reporting non zero values (it was late in the day and light was fading fast with a sharp decline in the voltage as I continued on). I spooled up another AI chat and described the issue. Home Assistant wasn't seeing what the Epic PWRgate was outputting. I was given a couple options: 1) I could run a command line helper on my HA instance that would gather the info (which for technical reasons wasn't workable), or 2) I could change to a built-in variable that Home Assistant would understand.

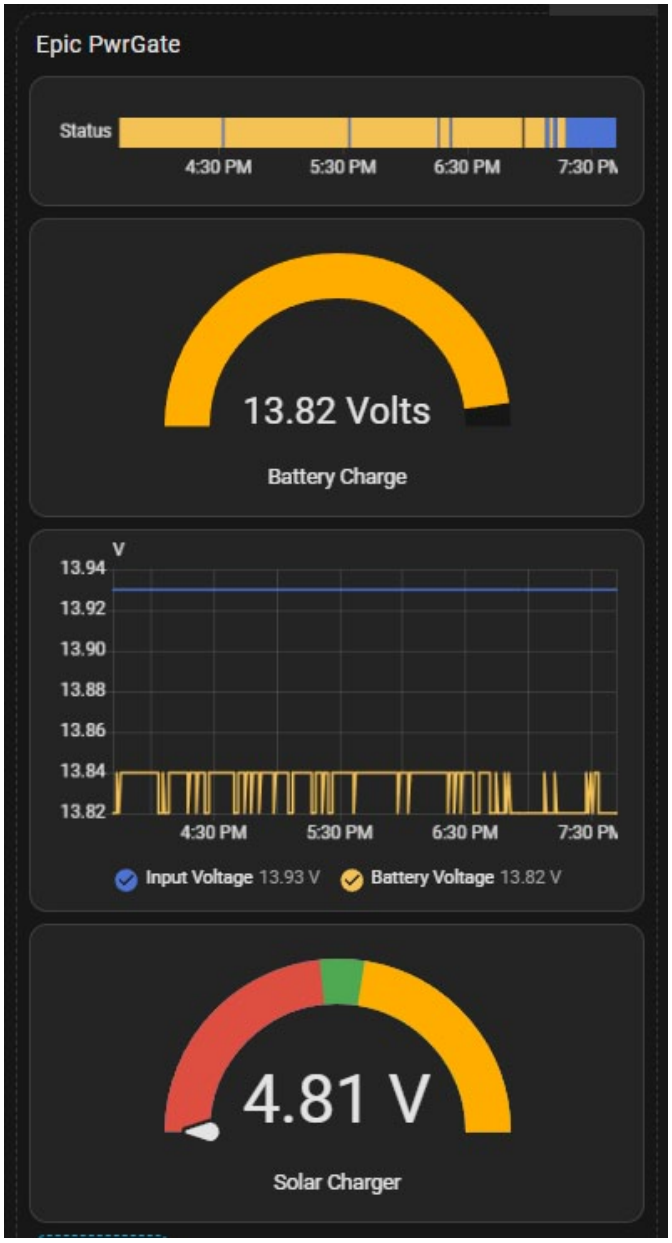
I went back to the code and used input_bypass_voltage to track the solar voltage. I know the programmers reading this are trying to avoid the eye twitch, but again, I'm a sysadmin and not a developer. I just wanted it to work. Once I did that and changed the card to reflect, I was now tracking the solar voltage capabilities of the Epic PWRgate as well.



My last task to complete will be to find the right DC barrel connector to Powerpole® connector for my shack PC to plug into the RIGrunner 4008 and have the complete Network UPS Tools suite ensuring my radio PC is protected and will safely shutdown on low battery condition. All this is thanks to some unintended use of data that West Mountain Radio built into their device.

Code on Github: https://github.com/W9RNR/nut

Powerpole® is a registered trademark of Anderson Power Products, Inc.



Programmatically Controlling Internet Capable RIGrunners

by: Mark Seigesmund, W9WMR

The usual way to control a RIGrunner 4005i or 10010i is through a web interface. Some users have an application where non-humans need to monitor or control the RIGrunner. There are several options that will allow you to do that.

1. Simple Network Management Protocol (SNMP)

The SNMP protocol allows one device to monitor or change information in another device. This protocol is available in Revision 3 and newer units. There are commands that can be sent to list what is available for monitoring and changing as well as commands to get a value or change a value. Not all values can be changed. There are many free utilities on the web you can use to play with this protocol. There are also interfaces built into most TCP/IP stacks if you are writing code. For these examples, we show the command line tools from SnmpSoft. They are free, simple to use and can be called from programs or batch scripts. There are GUI utilities as well.

To get a list of what objects are available, use snmpwalk. For example:

```
C:\>snmpwalk -r:192.168.100.108
 SnmpWalk v1.01 - Copyright (C) 2009 SnmpSoft Company
 [ More useful network tools on http://www.snmpsoft.com ]

 OID=.1.3.6.1.2.1.1.1.0, Type=OctetString, Value="West Mountain Radio
RIGrunner 4005i"
 OID=.1.3.6.1.2.1.1.2.0, Type=OID, Value=1.3.6.1.4.1.15117.1
 OID=.1.3.6.1.2.1.1.3.0, Type=TimeTicks, Value=0:10:28.60
 OID=.1.3.6.1.2.1.1.4.0, Type=OctetString, Value="West Mountain Radio"
 OID=.1.3.6.1.2.1.1.5.0, Type=OctetString, Value=4005i #1
 OID=.1.3.6.1.2.1.1.6.0, Type=OctetString, Value="A power supply near
you"
         ...
 OID=.1.3.6.1.4.1.15117.1.2.65.0, Type=OctetString, Value=Node 1
 OID=.1.3.6.1.4.1.15117.1.2.66.0, Type=Integer, Value=1
 OID=.1.3.6.1.4.1.15117.1.2.67.0, Type=OctetString, Value=Node 2
 OID=.1.3.6.1.4.1.15117.1.2.68.0, Type=Integer, Value=1
 OID=.1.3.6.1.4.1.15117.1.2.69.0, Type=OctetString, Value=Node 3
 OID=.1.3.6.1.4.1.15117.1.2.70.0, Type=Integer, Value=1
 OID=.1.3.6.1.4.1.15117.1.2.71.0, Type=OctetString, Value=Node 4
 OID=.1.3.6.1.4.1.15117.1.2.72.0, Type=Integer, Value=0
 OID=.1.3.6.1.4.1.15117.1.2.73.0, Type=OctetString, Value=Node 5
```

If the default port is used, you only need specify the IP address of the RIGrunner (-r:).

To turn node 1 off (a zero value), you need the OID (-o:) for the snmpset command with the value (-val:) and that the value type is integer (-tp:int) like this:

```
C:\>snmpset -r:192.168.100.108 -o:.1.3.6.1.4.1.15117.1.2.66.0 -val:0 -tp:int
SnmpSet v1.01 - Copyright (C) 2009 SnmpSoft Company
[ More useful network tools on http://www.snmpsoft.com ]

OK
```

To read a value, you need the OID like this example to read the RIGrunner power on time:

```
C:\>snmpget -r:192.168.100.108 -o:.1.3.6.1.2.1.1.3.0
SnmpGet v1.01 - Copyright (C) 2009 SnmpSoft Company
[ More useful network tools on http://www.snmpsoft.com ]

OID=.1.3.6.1.2.1.1.3.0
Type=TimeTicks
Value=1:20:03.30
```

You can password protect read and/or write access to SNMP. To do this, you need to enter a password for the "Community" entries under network settings via the web interface. You then use that password with the **-c:**password command line option. In SNMP, the community name is effectively the password. You can further protect against brute force attacks by using a non-standard port number for SNMP.

2. Browser Emulation

You can use HTTP to send requests directly to the RIGrunner. The method will vary depending on your programming language and library. This method works even if you have an older unit without SNMP. The following is a source code example to toggle the on/off status of port 1 at the press of a button. This code is in Delphi using the Indy library.

```
procedure TForm1.Button1Click(Sender: TObject);
var IdHTTP: TIdHTTP;
    Params : TStringList;
    result :string;
begin
  IdHTTP := TIdHTTP.Create(nil);
  Params := TStringList.Create;
  try
     Params.Add('DIGOUT_TOGGLE=1');
     IdHTTP.Request.Method :=  'POST';
     IdHTTP.Request.ContentType := 'application/x-www-form-urlen-
coded';
     result := IdHTTP.Post('http://192.168.100.108/index.htm',
params);
  except
  end;
  IdHTTP.free;
  params.free;
end;
```

To find the names like DIGOUT_TOGGLE you can use WireShark to analyze what your browser uses for specific operations or examine the HTML in your browser using a right click and VIEW PAGE SOURCE.

3. Node-Red

Node-Red is a generic web based graphical data collection interface that can be used for many purposes. For more information, visit https://nodered.org/

If you also use Node-Red to control your radio, then you can set it up to turn devices on/off based on your band or simply time of day.

One of our customers developed an interface for the RIGrunners and produced a video to show it in operation:

https://www.youtube.com/watch?v=nc6gDAvxf4Q

Getting Started with HF Digital Modes

by: Sholto Fisher, K7TMG

Our presentation for Hamvention 2026 provided an overview of digital modes in Amateur Radio explaining how they evolved, their benefits, common applications, and the equipment and software needed to use them.

Digital modes use analog radio signals (such as multiple tones, frequency shifts, or phase shifts) to transmit digitally encoded information, including Text, Images, Binary data and Control information between computers or modems.

Most amateur digital modes are designed for person-to-person communication, though some are used for networking and automated messaging.

History of Digital Modes

Digital communication has existed much longer than many people realize.

Key milestones include:

Morse Code (1840s): Often considered the first digital communication system because it uses discrete symbols.

RTTY (Radioteletype, 1920s): One of the earliest true digital radio modes, still widely used in contests.

Hellschreiber (1925): Sends graphical representations of text rather than encoded characters.

AMTOR (1978): Improved HF reliability using error correction.

Packet Radio (1980s): Enabled the first amateur wireless digital networks.

Pactor (late 1980s onward): Faster, more reliable data transfer, especially for HF email and bulletin boards.

PSK31 (late 1990s): Revolutionized amateur digital communications by allowing operation with only a computer sound card instead of expensive hardware modems.

2000s: Explosion of sound-card modes including MT63, Olivia, Contestia, MFSK, JT65, WinMOR, and digital voice.

Today (2026): The most common modes include FT8, FT4, WSPR, VARA, PSK31, RTTY, JS8Call, FreeDV and Olivia.

Read more: https://www.westmountainradio.com/kb_view_topic.php?id=OT302
Watch the presentation: https://www.youtube.com/watch?v=k9chp_I1hb0

INTERESTED IN MICROCONTROLLERS?
Click here for more info:
www.westmountainradio.com/pic_resources

READY TO PROGRAM A PIC® MICROCONTROLLER?
Click here to learn more about the CCS C Compiler:
www.westmountainradio.com/c_compiler

Visit Our Booth!

Radio Expo
September 27, 2026
Belvidere, IL

Winterfest
January 23, 2027
Collinsville, IL

Follow Us!

West Mountain Radio
1020 Spring City Dr.
Waukesha, WI 53186
www.westmountainradio.com