Problems with WiRES-X 2.X

Category: WiRES-X
August 9, 2026
Note: Yaesu has now released version 2.01. Our recommendation remains unchanged. This version mostly addresses those few people who need the software to work with the FT-1.

Yaesu recently released version 2.x of the WiRES-X software. These are the primary things to know:

- It is the only version of WiRES-X that will work with the FT-1. (But hopefully you're using your FT-1 to do more than make a WiRES-X node!)
- IPv6 is now supported. Good news for people stuck with ISPs that only do IPv6.
- There is an IPv4 compatibility mode that can be used.
- Version 2.x IS NOT backward compatible. Version 2.x CANNOT connect to rooms running 1.5.x.
- Most Room operators are sticking with version 1.5.x because of bugs and other issues. This means that if you update to 2.x, you will NO LONGER be able to connect to almost all of the popular Rooms.
- Version 2.0 is buggy. It is a complex update to the software and it may take several releases before Yaesu gets it stable.

Note: I've been taking one for the team and have been running FusionTech on 2.01 for more than a month. These are the reasons why I don't like running a room on 2.01.

- This really is a problem! If you're running a room and need to reboot WiRES-X, you want that reboot to go as fast as possible. With a 1.5.7 reboot, nodes would rejoin the room immediately. This is because the room still exists in

the Yaesu list server. Nodes will automatically try to rejoin every 30 seconds. Thus within 30 seconds the nodes have rejoined. This is NOT TRUE with 2.0.1!!! If a 2.0.1 room is rebooted, it sends a command to the list server to DELETE THE ROOM'S ENTRY IN THE LIST SERVER. This means that a node has to propagate through the system. The result is that it can take more than 10 MINUTES for nodes to rejoin. Very bad if this happens during a net! (It has happened to me a number of times.)

- But it gets worse. I use the repeater that hosts the room to bridge to our YSF reflector. However the repeater can't rejoin the room until the Yaesu list server updates the local list of rooms. Depending upon where your room is in the list, this can take 10 to 15 minutes. So in the middle of the net, the YSF users are disconnected from WiRES-X for a very long time as is true for all users on the local repeater – which probably includes net control.
- What???? The node running on the same WiRES-X instance as the room cannot connect to that room right away? Nope. Even though both are running in the same software!
- Here's what's happening. 1.5.7 used a distributed model for providing rooms. When connecting to a room the traffic between your node and the room goes from you WiRES-X to the WiRES-X software used to host the room. Not so with 2.0.1. With 2.0.1 your WiRES-X connects to a Yaesu server in Japan. All of the rooms are hosted in Japan, not on the rooms WiRES-X software. This has advantages and disadvantages.
- Advantages: Port forwarding is not required since your WiRES-X software connects directly to a public server – like pulling up a web page. No port forwarding is needed. The efficiency of hosting on a server is much better than hosting on a consumer IP service. (Performance is mostly constant and not degraded by Netflix.) This also

integrates well when you don’t host a room and directly connect your computer to your PC via PDN. In this case Yaesu uses TCP to connect directly to the server in Japan. Japan then converts the TCP to UDP and sends the traffic to the host WiRES-X. Now both methods do the same thing.

- Disadvantages: Yaesu chose to use TCP to send audio. Nobody does this. When you listen to a podcast, music online, or watch Netflix, this is all done over UDP. Why? TCP requires a hand-shake between the server and the client. It also retries transmissions if one is lost. It guarantees that all packets are received in the correct order? Nice, eh? Until a packet or two get lost and a retry occurs. This can cause a delay of up to a second based on the ping times between the two computers. Thus the audio will have a noticeable stutter even though only 100 ms of data is lost. TCP is much more compute intensive than UDP. UDP does not have a hand-shake. Once the connection is made it just starts streaming packets to the client. The packets may arrive out of order if the Internet path between the two computers changes from one packet to the next. One would think this is terrible, but it isn’t. It means that UDP keeps on sending even if random packets get dropped. You’ll hardly notice a dropped packet (100 ms, or part of a word) versus loosing the time it takes for a retry – a second or more. You won’t notice a dropped packet on opening a web page, but you will certainly notice it when watching video or listening to audio.
- Why is Yaesu using TCP? Nobody else uses it for this purpose? I honestly don’t know. We know that UDP works fine because with 1.5.7 that’s what we’re using. Maybe they’re using TCP so that a port is open to connect to a web server? Or something? This makes no sense as they could just as easily have the UDP port open. That works.

That’s what we do with 1.5.7. TCP requires more server overhead, while UDP is very, very simple.

- Yaesu never needed users to open ports!!!! Here’s how that works. Let’s say A and B what to have a VoIP conversation using UDP. But A can’t talk to B because B doesn’t have any open ports. That where a public server, C, comes in. Both A and B connect to C and tell C who they are and what their IP address is. When A want’s to call B, A contacts C with his intentions. C responds with B’s IP address. Meanwhile the server lets B know that A is trying to contact him so B then sends a UDP packet to A. Meanwhile A is sending a UDP packet to B. These packet probably never get received but what they do is important. Sending a UDP packet causes your NAT router/firewall to open an incoming port on the same port you send the UDP packet from. Thus A and B have both opened a UDP port known only to A and B, and they can start sending packets directly to each other without involving C. This is how the older version of WiRES-X should have worked! This is what everybody else does. It’s not a secret.
- YSF works and an open port is not required. In many ways YSF is a lot nicer than WiRES-X. It only uses UDP. When a YSF user wants to connect to a reflector, they look in a Host file for the IP address and port number of the reflector. They send a UDP packet to that address, opening up an incoming port. The reflector then sends traffic to that port. YSF only requires an open UDP port if a reflector is being hosted and most people are not doing this. There are other advantages to YSF over WiRES-X which I will point out is a subsequent post.
- Single point of failure. Since ALL WiRES-X traffic is now going to through one service, failure of that service shuts down the ENTIRE WiRES-X network. This can happen when Yaesu does a reboot. No matter when they do a reboot

it will be prime time for someone on this earth. That wouldn't happen with 1.5.7. Only people that have just booted WiRES-X will notice that the list server is down. That's because everyone who has WiRES-X running already have a cache of all the nodes and rooms – they're just not getting any updates. But with 2.0.1 everything goes down. No longer can we recommend WiRES-X networking for anything related to emergency services. WiRES-X IS NOT RECOMMENDED FOR ANY EMERGENCY SERVICES ORGANIZATION. YSF is a much better choice as you have total flexibility in how the networking is done and where the server is.
- One last disadvantage is that it's impossible to connect to 1.5.7 rooms. A number of rooms that I normally connect to have been inaccessible since running 2.0.1. It is inconceivable to me why Yaesu chose to make 2.0.1 incompatible with prior versions of WiRES-X!

It is for these reasons that FusionTech recommends AGAINST updating to WiRES-X 2.x. Stay tuned to this website or the FusionTech net for any updates.

Hacking with Hackerboxes

Category: Projects
August 9, 2026
Do you miss your soldering iron? Let's face it, you know where it is, it just doesn't get much use.

What if you could get a different kit every month that was simple to build, did something interesting, and was a microcontroller learning opportunity?

Hackerboxes.com is like a magazine. You can subscribe to it and rather than something to read, you get a small kit with all the parts. The kits involve firmware which is built on the commonly available (free) Arduino platform. Instructions guide you through everything you need to do to get the code loaded. You don't have to write the code, but you can modify it to your specific wants.

These are some of the "issues" I've really enjoyed.

- An AI lab where a camera and LCD display are used to experiment with AI image recognition. The CPU includes vector instructions that accelerate AI processing.
- Do you know what the first personal computer was ever offered for sale. You're wrong! It was the KENBAK-1. A reproduction (simulated) version of the computer was provided in Kit #101.
- #102 is a USB-based oscilloscope. It comes with parts to help experiment with analog circuits.
- #105 experimented with 2.4 GHz, 433, and 915 MHz radios.
- Learn about Bluetooth with kit #109. It uses the NRF52840 and ESP32 to implement audio streaming or digital control.
- #111 experiments with LoRa communication. It has two radios. One of which has an OLED display.
- #112 experiments with Circuit Python It implements an NES emulator with HDMI outputs.
- I liked #114, WOPR. The kit was meant to emulate the WOPR computer from the movie War Games. It's the width of a 19" rack. However, I use it as a Wi-Fi based sign where I can update the text with a web browser.
- #115 is a "war driving" module. What's that? It's a set of radios, along with GPS, that scan for Wi-Fi and Bluetooth signals. The information is then downloaded to a computer. They also include a Li-Ion 18650 batter board to power it or other things.

- Want to play with RFID? You need #107.
- The MCU lab, #121, is a nice little kit consisting of a base interface board and four different microprocessors with which to experiment.
- How’s your health? #122 measures your ECG.
- #124 teaches you how to hack electronics.

A lot of the kits involve the Raspberry Pi Pico boards, my personal favorite. Many also involve versions of the ESP32 which is an absolutely amazing series of controllers. My career started with designing products using microcontrollers and microprocessors and yet I’ve learned a lot from these kits. Technological advancements in the past 15 years has resulted in impressive capabilities at very low costs.

Some of these kits are still available for $59. Subscribers pay $44 per month to receive their monthly kit.

It would be great fun if some FusionTechers joined up. That way we’d have something in common to discuss and help each other out. Just to level expectations, not all the kits have gone without issues. Some were my fault, others were not. There is a discussion board to post and answer questions. It would be even better to bring them up on FusionTech.

Check it out at HackerBoxes.com.

WXScheduler

Category: Automation,Projects,WiRES-X
August 9, 2026
Update: WXScheduler 2.9.9 is available. The 2.9.9 installer is

here.

Here’s what has changed with 2.9.9:

WXscheduler v2.9.9 updates: Several fixes related to missing automation delays that are required for properly updating Wires-X Settings.

WXscheduler v2.9.8 updates: – Now restoring Wires-X app to the Desktop only when Wires-X is currently minimized and WXscheduler is about to perform automated Wires-X Settings updates. – If the Wires-X app was minimized at the start of automated Wires-X updates, then the Wires-X app will be minimized again after the automation actions have been completed. – Validation of 5 digit “Return to RoomID” having an even first digit, has been removed.

WXscheduler v2.9.7 provides many improvements. The key ones are: – Compatible with Wires-X versions 2.010 and v1.5xx – Faster automation of the scheduled Wires-X Settings updates that are independent of CPU speed and background load. – Replaced main window’s “Scheduler list” with “View/Modify Schedule” button. – “Wait for Wires-X.exe to Restart” functionality that maintains a valid automation connection to Wires-X. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

2.9.1 fixes problems with some Windows installations not working because there no longer is a “Programs (x86)” directory.

WXScheduler is a program that allows WiRES-X nodes to automatically switch to a Room or Node at a predetermined date and time. It’s written in Python, but we now have a Windows

installer which does not require the installation of Python.

Description

WXScheduler is a program that allows your WiRES-X node to be connected to another Room or Node at a predetermined day of week, hour and minute in the specified timezone. If your browser balks downloading WXscheduler-v290-Install.exe and you cannot override the warning, then rename the xxxxx.crdownload file using File Explorer or CMD. NOTE: Recent Windows Security updates may rightly so quarantine WXscheduler-v290-Install.exe. Detecting it as a Trojan. Well that is correct only on the basis that it will install another EXE file You can override this by going into Windows Security –> Protection History and find the most recent “threat” and click the Restore link. Finally this especially occurs if you are using Windows’ OneDrive. Which I highly recommend that you do not use! WXscheduler is written in Python, converted into a Windows EXE, and delivered as an installable Windows application for Windows-7 or higher. Which means the User no longer has to install Python and library packages to run WXscheduler.pyw. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

The 2.9.0 update changes:
-Address varying CPU speeds and their impact on the automation

-Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

The old 2.6.1 installer is available here. The Github repository is here.

If you experience problems, please use the Fusion Technical Net to report them (preferred), via Github issues, or email.

Revision History:

WXScheduler was written by Bill, W9LBR. It has been packaged as an .exe with a Windows installer by Chris K9EQ. In addition, the following changes have been made.

-2025-09-15 Address varying CPU speeds and their impact on the automation

Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

– 2023-02-12 2.6.1a Clicking the ‘x’ button on the scheduler menu no longer causes the program to crash

– 2023-01-10 2.5.1 Removed radioID error messages from the log (there were too many of them and they didn’t add value)
Added an icon for the program

The installer creates a WXScheduler directory under the WIRESXA directory where the Yaesu software lives. It installs the following files:
– Readme.txt, this text
– WXScheduler.pyw, the Python source file
– WXScheduler.ico, the icon file
– WXScheduler.exe, the executable
– Uninstal.exe, the program to uninstall WXScheduler

The uninstaller removes these files and the directory. It does

not remove the .json
configuration file one directory above.

The packaged version displays the GUI without a shell console. If you want the console as well,
rename the file to *.py instead of *.pyw. You'll then need to open up a shell (command prompt)
and launch with a 32-bit Python interpreter as in: "C:\python32 WXSheduler.py".
The source code is available at https://github.com/K9EQ/WXScheduler.
Also see HamOperator.com for more information about WXScheduler.

Note that the WXScheduler is either stored in the \OneDrive\Documents\WIRESXA folder, if it exists, or in the user's Documents\WIRESXA folder. So if you can't find it in your WIRESXA folder, you may have OneDrive so look there.

WXscheduler v2.5 updates:
– Schedule settings now require a Timezone parameter.
– Compensation for Daylight Savings Time is now automatic on a per timezone basis
– Load Python Timezone package using Windows Command Prompt:
pip install pytz
– WXscheduler.cfg is now more portable and user editable
– User's HOME path is now resolved at run time and no longer saved in WXscheduler.cfg
– JSON data fields are now saved in pretty mode
– When WXscheduler-v2.5 detects an existing WXscheduler.cfg without timezone data fields,
the user will be prompted to set their local timezone value and all existing events will
be updated with the local timezone value.

– WXscheduler’s main window:
– Scheduled events are displayed in chronological order
– When the main window is moved, its new position is remembered. And all sub-windows
will be opened in the same position. This replaces PySimpleGUI’s default of centering
window within the display.
– [Add Event] and [Delete Event] buttons have been replaced with the [Scheduler] button that
when clicked provides a new window with buttons: [New] [Delete] [Update] [Cancel]

WXscheduler v2.4 updates:
– Additional exception debug information
– Changed main window title from “Wires-X Scheduler” to “WXscheduler (v…)”

WXscheduler v2.3 update:
At startup determines whether Microsoft OneDrive is active or not and locates
the Documents/WIRESXA and user/Desktop folders accordingly.

WXscheduler v2.2 updates:
At startup verifies:
– Wires-X.exe is accessable at its standard location
– /Users/????/Documents/WIRESXA folder is accessable (where WXscheduler.cfg is stored)

WXscheduler v2.1 updates:
Better displays exception information
Shows expected WIRESXA pathname to last heard data file

WXscheduler v2.0 update adds:
Display Last Heard information that the Wires-X application updates once a minute.
If available, Lat/Lon displayed as 6-character Grid Square.

Wxscheduler v1.3 updates:
Last Heard information now correctly displays the newer Yaesu models
(i.e. FT5-D, FTM-200, and FTM-500)
Desktop\Wires-X_Last_Heard.html contains hypertext encoded callsigns
that perform a QRZ.com callsign lookup.

Prerequisites (versions below are what was tested on):

Windows PC (win7 and higher)
Wires-x App (Ver-1.550)
Python 3 (32-bit) because Wires-X App uses the win32 UI
site-packages: PySimpleGUI, pywinauto, pytz

Installing Python:

Recommend going to https://www.python.org/downloads/windows/
to find a Python release that matches Windows on your PC.

After installing Python, use a CMD window to load site-packages:
– pip install pywinauto PySimpleGui pytz

Installing WXscheduler.pyw:

Copy WXscheduler.pyw to your Desktop

Double click on the WXscheduler.pyw icon to launch the program.

Hint: To see any startup or run time issues:
– Open a CMD window
– Copy WXscheduler.pyw WXscheduler.py
– WXscheduler.py

Customizing WiRES-X

Category: WiRES-X
August 9, 2026
In this post we discuss how the operation of WiRES-X can be customized through the use of the Windows Registry.

Okay, here’s a bit of a WiRES-X secret. Many of the settings which cannot normally be changed, can be changed. These include the call sign, location, etc. In fact, it is possible to use an HRI-200 with one node number with a different node number. This is necessary should one be running a room and need to switch to a different server located elsewhere.

Here’s an example that we have used. Room #21493 is FusionTech. This HRI-200 belongs to Pete, AD0MI, now located in Idaho. However the server we’re using is located near San Diego and has a number of 21811. If you check, the callsign is not AD0MI as registered with Yaesu, but K9EQ. Being able to do this is a necessity for popular rooms that cannot tolerate a room being down because an HRI-200 is being sent in for repair.

This data is stored in the Windows Registry. In the Windows search bar, type in the text “regedit”. The Registry Editor will open. This is where Windows and applications store a lot of information such as window location and size, settings, preferences, file locations, etc. Be very careful when using regedit as any changes you make are immediately effective and there is no undo. One can really mess a computer up with regedit!

In Regedit, press CNTL-F or use Edit->Find. Enter the search term “YAESUMUSEN” and deselect everything except “Keys”. Wait as it searches. The registry is very, very large and Microsoft’s

search function is not very efficient.

Expand the WIRESXA folder. These folders contain all of the operating parameters of the WiRES-X software. Click on “USER_INF”.
 In the folder you will see the key “callsign” and the value of whatever callsign you’re using.

To change an item, stop the WiRES-X software. Double click on the key whose value you wish to change. Type in the new value then click “OK”. Restart the WiRES-X software. If you changed the callsign, you’ll see that the callsign has changed.

In this folder you will also find items like the HRI-200 serial number, room number, and node number. If you change this information to match a different HRI-200, the software will then take up the identity of that HRI-200. Thus you can swap server locations or HRI-200s without changing the room number. Note that the node and room numbers appear under other folders and these must be changed as well.

Not all of the values are obvious as to what they mean but probably contain values we wish we could change. The DATA_COM folder has some interesting values in it. It would be interesting to perform experiments just to see what all of these things do.

I should note that in our long use of this product, we have identified many strange bugs. One common one is that there is no outgoing traffic even though incoming traffic is fine and everything looks like it is working. Usually restarting the software solves the problem, but not always. In another case transmissions from a YSF bridge were blocked but everything else worked. Fixing this took three months. It was necessary to uninstall the software, delete any residual configuration files, AND delete all the data in the registry. It was then necessary to reinstall the software and repeat the authentication process and re-entering all of the parameters manually before it would work again. So if you have a problem you’ve never been able to

figure out……

Yaesu’s list server contains some of the information in the registry. In some cases it’s only updated when the software is registered (callsign, node number, etc.) and some of it is updated every time the software starts (city, state, location data, etc.). In this last category the workaround is to run a script that updates the registry just before WiRES-X is started.

If you find this stuff interesting, please stop by our FusionTech net which is Monday, 7:30 PM Central in the FusionTech Room / YSF 21493. We like talking about the technical nature of our toys. We can also be found on HamOperator.com.

Favorite Rooms

Category: Popular Rooms
August 9, 2026
People often ask which rooms to listen to. Over the years, I have developed my lists of favorites and I will share them with you. If you think your room or net should be added to the list, let me know.

Dave, W6DS, provides a history for some of the rooms on his HamRadio.Solutions website. To access the YSF histories directly, click on the Network Digital Monitor link. The NDR uses YSF to acquire the audio and data. The rooms use a WiRES-X to YSF bridge so that these YSF reflectors carry the same traffic as their associated WiRES-X rooms. If you would like to have your room/reflector added to the NDR, contact Dave directly.

FusionTech 21493 WiRES-X YSF

This is naturally my favorite as I’m an original founder from all the way back in 2014. It was originally founded out of Minnesota and a lot of people do check in from the upper midwest. The geographical limits have been removed since WiRES-X provides a global community where a friend just down the street is just as close as a friend from the other side of the planet. It is not a terribly busy room so it’s a good place to let a node, repeater, or hotspot sit. We encourage those who are tuned in to apply the motto, “No call goes unanswered.” If you hear someone put out a call, please answer it even if it is just for a quick “hello.” You never know. You might have just met your next life-long friend.

The net encourages friendly conversations on subjects that are likely interesting to those listening. Technical discussions are encouraged.

Nets

Fusion Technical Net Monday 7:30 PM Central
South Dakota Net Thursday 7:30 PM Central

Website

HamOperator.com

Room audio history is FusionTech Audio History.

Texas-Nexus 21636 WiRES-X YSF

We have a lot of friends in Texas and a lot of them are on this network. The founders of Texas-Nexus got their start when we

were MNWis back in the day. They splintered off to form their own, highly successful, network. This is one of the most popular Rooms on WiRES-X. Stop in to enjoy a warm, friendly, Texas welcome!

Texas-SADRC 40324 WiRES-X and YSF

While we're on the subject of Texas, here's another really good group also headquartered out of San Antonio. The San Antonio Digital Radio Club. Also very popular.

Audio history is Texas SADRC.

The-DARC-Room 40012 Wires-X

Another "digital amateur radio club" with a rather clever name. This group is in Minnesota and was also part of MNWis way back in the day. A very friendly group who will be happy to tell you how great the weather is in Minnesota. Note: They will always try to make it sound cold and miserable except when it's hot and muggy. Their fiendish goal is to make it sound so terrible you won't want to move there! And if from that description you get the impression that they're a great bunch of people, they are!

Nets

AUS-Repeater-Net 24160 WiRES-X and YSF

There is a certain “You think that’s a big knife, THIS is a big knife” about Australian hams. A great sense of humor if you can understand what they’re saying and a welcoming sort of nature. Pete, VK5JP, is one of the many instigators behind this group. You’ll find him always eager to teach foreigners how to speak “Aussie”. Make a friend from down under and check out this room. They have a great net which occurs on North America’s Saturday, their Sunday. The time varies between 2:00 PM and 4:00 PM Pacific, depending on the time of year.

Nets

Australia Net Saturday (US) Time varies from 4:00 PM to 2:00 PM Pacific. Check the time zone for South Australia. The nets are 9:30 AM and 6:00? PM local.

CQ-Canada-VE1AO 40678 WiRES-X

The most popular English-speaking net in Canada. And you know Canadian’s. They’re always very friendly. Check them out and make some new Canadian friends.

America-KC-Wide 28054 WiRES-X YSF

This is THE most popular Room in all of WiRES-X and YSF. There is always a net going on, a conversation, or someone to answer your call.

Nets

https://www.kansascityroom-wide.com/nets/

Website

KansasCityRoom-wide.com

Audio history is KC Wide.

CALNET-Group 83196 WiRES-X YSF

I’m an active member of this group. They have repeaters covering a good of Los Angeles, Orange County, San Bernardino, and San Diego counties. A friendly technically-oriented bunch. They also operate one of the largest FM repeater networks in the state covering most of Southern California, San Francisco Bay Area and the California desert regions include Fresno, Bakersfield, and San Bernardino county. If you’re in Southern California, check them out. Give me a call, there’s a good chance I’ll be listening here, to their analog service, or on FusionTech.

Here’s where you can find them:

CalNet Repeaters
Pleasants Peak C4FM (LA, & Orange County) 449.600 – (PL for FM 151.4)
Sunset Ridge C4FM (LA, & San Bernardino County 447.020– (PL for FM 110.9)
Heaps Peak C4FM (San Bernardino, Riverside County and High Desert 445.740– (PL for FM 136.5)
Fowler Peak C4FM (San Joaquin, Stanislaus and Sacramento counties) 442.725+ (PL for FM 100.0)

Also on these non-owned repeaters:

Catalina Island Repeater 448.900 -5.00 C4FM (110.9 PL for FM only use)
Mt. Palomar (San Diego) 147.075 +.60 C4FM (107.2 PL for FM only use) I provide the WiRES-X feed for this one.

Website

Calnet.org

The CalNet audio history is CalNet.

SoCal Ham (LA Repeater) 21042 WiRES-X YSF

SoCal Ham operates a number of Fusion repeaters in and around Southern California. A current list of repeaters and other information can be found on their website.

Website

SoCalHam.com Link not currently working as of 5 Sept 2025

Audio history is SoCal Ham.

Other rooms to check out are:
Text TBA

MVARC-Idaho 43210

This is a very friendly bunch of hams with a network that covers from Boise to Idaho Falls and points beyond.

Website

MVARC

Under development

IL-Chicago-Link 82483
Nevada-USA 21676
Wolf-Den 28941 Wolf-Den
NWFG 44222
CQ-NW 86403
Ohio-Link 40557

We’re Now FusionTech

Category: Fusion
August 9, 2026
MNWis has changed its name to FusionTech. Nothing else changes. The name was changed to better reflect the nature of the organization from Minnesota and Wisconsin midwest centric to global covering interests in technology and Fusion in particular.

The network is still reachable at Room and YSF 21493. The Fusion Technical Net is the same and the people running it are the same.

The FusionTech network is part of the Bakken Amateur Radio Society (BARS). BARS was formed as a club of amateurs who worked

at Medtronic. Since that time the relationship with Medtronic employees has become less connected and subsequently most members are not from Medtronic.

The founders of FusionTech are: Chuck, K0ORK; Pete, AD0MI; and Chris, K9EQ. The net started in 2014 and is the 2nd oldest net to be held using Fusion. (Milwaukee has the honor of being number one.)

Yaesu Repeater Lockup

Category: Fusion,Troubleshooting
August 9, 2026
It is well documented that the Yaesu DR-1X and DR-2X repeaters can lock up from time to time. When they lock up, they may:

- Not respond to any transmission or control function and
  - Transmit continuously or
  - Not transmit or
  - Transmit for the duration of the time-out timer, go off the air, key up and start all over again.
- The transmission may be digital “noise” or it may be a carrier

There is another form of lock-up in which an attached HRI-200 can be used to reset the repeater.

Why do they lock up?

We don’t know for sure why this happens as we don’t have access

to the source code of the repeaters. But guesses can be made from knowing what the problem is, how microprocessor and their software works, and the common mistakes programmers make.

The firmware consists of a number of modules that do specific tasks. Data is sent to a module, the module may call other modules, and it returns with possible data. The input data may contain values that are out of range for the module. For example one value might be mode where 0=FM, 1=DN, 2=VW, and 3=VW. But what if the module receives a 4? It should just stop and return to the calling module with an error indication. Most likely the solution will be to ignore the request and stop processing. That's what should happen. But a sloppy programmer may NOT VALIDATE INCOMING DATA and allow the '4' to be processed where the results may be unpredictable since no code was written to handle this case. There is good evidence that Yaesu software and firmware DOES NOT VALIDATE INCOMING DATA resulting in an UNHANDLED EXCEPTION.

There are other types of invalid data which should not be processed. The Fusion digital data might have uncorrected errors in it, thus it's impossible to know what the correct values are. A field of data that is sent may be longer than it should be. This problem is commonly known as a BUFFER OVERFLOW. There is some evidence to support that a buffer overflow is occurring. Since the receiving buffer must be limited in size, too much data may cause the firmware to overwrite values that are past the end of the buffer. These may be critical values that are needed for the correct operation of the repeater.

The repeaters use a real-time operating system. An RTOS is event driven and will perform a function when an event occurs such as a timeout, a signal received, a clock tick, a button push, etc. There may be cases where two events are related such that one must occur before the other. For example, one event may lock

access to a hardware resource and the other event clears it. If the order were reversed the hardware resource would never be accessible and the actions of the hardware may be unpredictable or stop altogether. This is known as a RACE CONDITION. It is very likely that this error exists in the firmware and that it may be combined with another firmware error. For example, a BUFFER OVERLOW may result in a RACE CONDITION which the engineers did not see.

As a result of the above firmware problems, it is not easily predictable what the microcontroller may do next. Some systems include a Watchdog Timer. This is a hardware timer that is set for a certain period of time, perhaps 10 seconds. The microcontroller's firmware periodically resets the WDT if it is running properly. If the WDT times out due to some sort of problem, the hardware resents the controller. We have some evidence that Yaesu has implemented a WDT in the DR-2X repeater. It is therefore apparent that the WDT has been turned off, set to a crazy long value, or that the modules that reset the timer are still running even though nothing else is.

A SOFT LOCKUP may occur if a RACE CONDITION removes access to the receiver and/or transmitter. While the repeater won't receive or transmit, the other functions are working. This means the repeater can be reset without needing to cycle the power. (There is no reset button like on your PC.)

The Evidence

The most difficult problems to fix are the ones that almost never happen. These lock-up problems are in the category of infrequent but very high risk (a drive to the mountain to reset the repeater. With remotely-controlled repeaters, a lock-up is an unacceptable risk. After all, other repeaters don't seem to have this problem.

(There is a repeater bug involving the CWID on FM, that is not being discussed here.)

It is very likely that these problems occur while a user is transmitting in the digital mode. We've not been able to observe a repeater suddenly locking up without some sort of input to key the transmitter. (However, since we haven't seen it doesn't mean it doesn't happen.)

My observations are that this is more likely to happen when a user is transmitting a WiRES-X command such as changing rooms.

It appears to be more likely if the user has a weak signal into the repeater.

It is more likely to happen on two meter repeaters than on 70 cm repeater.

Notice that each of these conditions have the word "likely". That's because there are unknown additional parameters which may be involved. For example, a weak digital signal will cause more digital errors. A particular error may be involved in creating the problem.

It may also be the result of a transmission starting at the exact time the microcontroller is performing a specific function, thus resulting in a RACE CONDITION. This problem is virtually impossible to observe.

It was previously mentioned that using a two meter repeater makes this problem more likely. The issue here is the amount of noise on the two meter band. All of our electronics and networks add noise to the environment. This noise level is much stronger on two meters than on UHF. There is also some evidence that the ethernet cables can deliver RF to the ethernet devices and these devices may act like a mixer. All of this causes more noise

making stations effectively noisier. The end result is more bit errors and buffer overflows.

What to do?

The correct answer to this section is to have Yaesu fix their bugs in the microcontroller – but don't hold your breath. This could be a very costly to find the cause since it is difficult to reproduce. Another problem is that there are no Fusion repeaters in Japan. (Only D* is allowed – that's along story.) Thus the Japanese engineers working on this project would have no way to experience the problem. (The repeaters can only be operated in a shielded room – thus no noise.) I've solved problems like this before and the first step is to make the problem happen more often while monitoring the CPU. Keep trying things until the probability of a lockup increases, then try doing more of that. The business model for this is that spending a bunch of money on this is not likely to sell more radios. I also believe that Yaesu farms out certain tasks, like firmware development, to outside suppliers. In any case there is no guarantee that the firmware engineers use radios outside of the lab.

If the issue is the result of a SOFT LOCKUP, sending a command to the repeater via an attached HRI-200 may resolve the issue. This may only require switching to the repeater page in the WiRES-X software an applying the repeater's settings with the same value.

If it is a HARD LOCKUP, the only solution is to reset the microcontroller. I.e., turn the power off then on again. Ouch!

- The simplest way to address this problem is to take a timer plug, set it to turn off during a time the repeater is not likely to be in use, then turn it on again as

quickly as possible. You may have to put up with a continuous carrier for a good part of 24 hours, but at least you won't have to drive to the hill.

- If there is Internet connectivity at the repeater site, you might be able to use a Wi-Fi enabled power switch to cycle power remotely. (I do this using my home automation.) Unfortunately this requires the action of a control operator.
- Since it is unlikely Yaesu will fix this problem, the ultimate solution is to have a circuit that will automatically reset a lockup condition. Thus we have the De-lockup Project (below).

De-lockup Project

Attached to this post is a project that will automatically reset a locked-up repeater. The concept was originated by Mike, W0IH. Mike informs me that he does not have the bandwidth to provide support for this project in the long term. It is likely we will be refining the project as time goes on. One thing I'd like to do is eliminate the need for a cheap SWR meter. You can obtain a zip file of the project from here.

The project uses an inexpensive Arduino Uno microcontroller board. In addition there is a relay board plus a few extra components. The circuit does not require you to even take the cover off of your repeater and therefore will maintain the warrantee. You can learn more about Arduino at The Arduino home page.

Japan Ham Faire Pictures

Category: Activities
August 9, 2026
These photographs of the Tokyo 2024 Ham Faire are courtesy of Masa, JA1WLQ. Also thanks to W4IOD for providing the “relay”!

- 


HF/50/144/430MHz ALL MODE FIELD TRANSCEIVER
YAESU
FTX-1F

- 
YAESU
7012510
FT8
FTDX101MP

- 
The radio... YAESU
The radio... YAESU
JAIA
8J1RL

- 
144.000.000
7.084.920
HF/50/144/430MHz ALL MODE FIELD TRANSCEIVER
FTX-1F
YAESU
オートアンテナチューナー

- 
7.048.000
FT-710 Aess
FT-991A

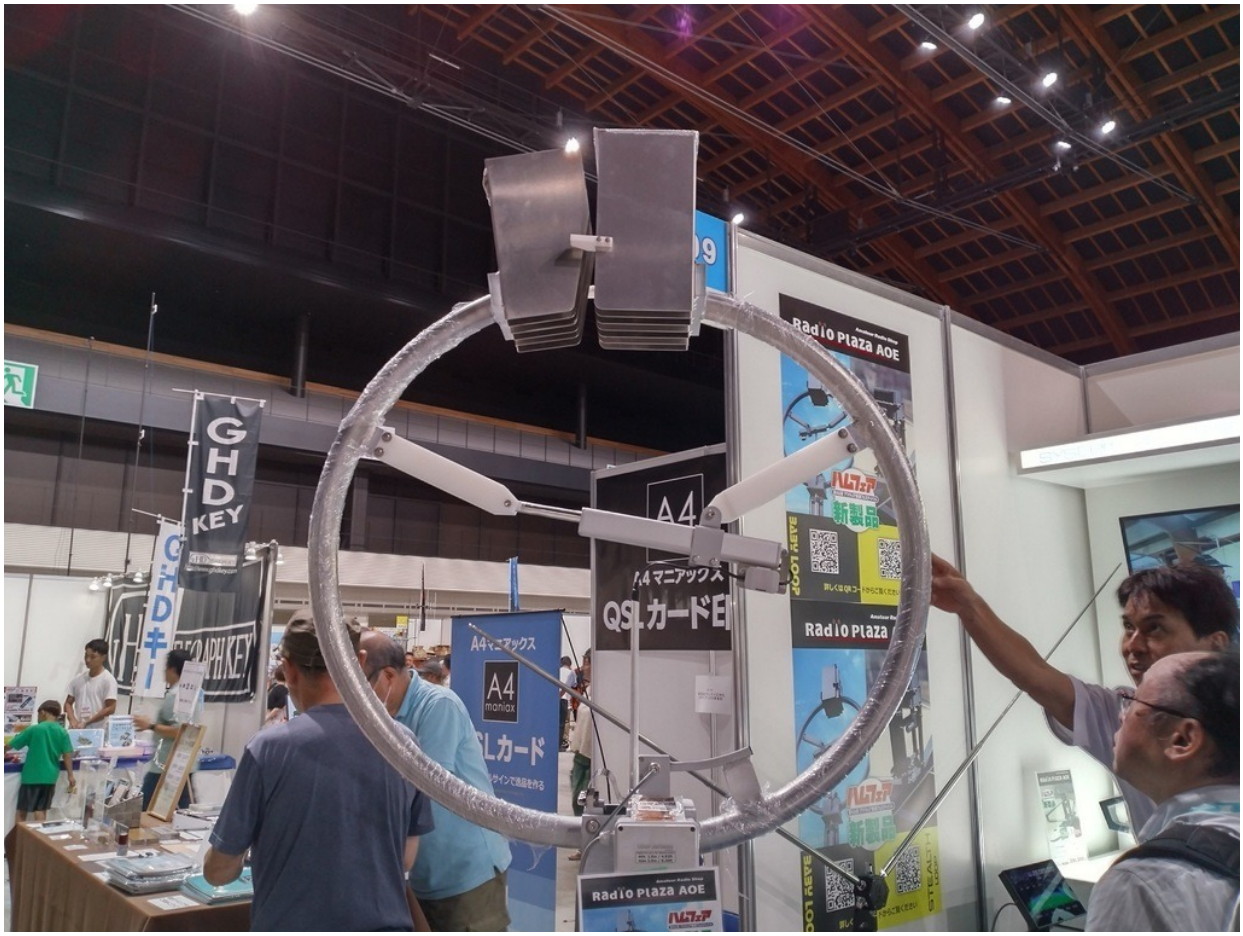

Sako Hasegawa, JA1MP
Founder of Yaesu Musen
and esteemed friend and colleague of
the global amateur radio community
TEMP METER
ALC METER
7.003.890
21.237.230
OFF
AMP1
3kHz
FAST
AUTO
MAIN
CENTER
SPAN
3DSS
MONO
MULTI
EXPAND
HOLD
M-100
FH-2
VCT-101

- 
第46回 アマチュア無線フェスティバル
ハムフェア2024
アマチュア無線
de
新たな体験
in 有明
Welcome to the new world!
会期
2024年8月
24(土)・25(日)
(10:00~17:00) (10:00~15:30)
今年は有明GYM-EX!!
ARIAKE GYM-EX
入場整理券(税込)
●一般▶2,000円
●会員▶1,000円(会員証提示)
●女性▶1,000円
●大学生(22歳未満)まで無料
JARL
JARLに入会しよう!
主催:一般社団法人日本アマチュア無線連盟
後援:総務省/文部科学省/東京都/NHK/公益財団法人日本無線協会/
一般財団法人日本アマチュア無線振興協会/日本アマチュア無線機器工業会/公益財団法人東京観光財団
協賛:電波新聞社/電波タイムズ社/CQ ham radio/子供の科学

- 
ARIAKE GYM-EX

- 
NEW
CAR TRANSCEIVER
Launch 2025
KENWOOD
VHF/UHF TRANSCEIVER
16:10
CALL
VFO
MR
MENU
145.000
433.000
H FM
PM
KEY
FUNC
VOL
SQL
MODE
TONE
REV
PF1
PF2
MIC
PANEL
micro SD
お手を触れないでください。

- 
KENWOOD
ALINCO
ICOM
QSY
QRH
QRZ
QRP
QRO

- 
GHD KEY
Radio Plaza AOE
新製品
A4 maniax
Radio Plaza

- 


TH-D75
解説集はこちら
KENWOOD
80.000
TH-D75
KENWOOD
KSC-47L

- 
ICOM
STAFF
7.020.67
7.120.33
7.010.92

- 

Hamvention 2024 Photos

Category: Activities
August 9, 2026
These photos are courtesy W4IOD. They are presented in four parts, each of which is a PDF. Click the links below for each part.

http://hamoperator.com/files/Hamvention Photos Part 1.pdf

http://hamoperator.com/files/Hamvention Photos Part 2.pdf

http://hamoperator.com/files/Hamvention Photos Part 3.pdf

http://hamoperator.com/files/Hamvention Photos Part 4.pdf

Fusion Cheats

Category: Fusion Rigs
August 9, 2026
You’re probably aware that turning your Yaesu Fusion radio on whilst holding down one or more buttons causes it to do different things. There’s a self-test on the FT1. The FTM-200 can display supply voltage. The FTM-500 can do cross-band repeat. Right now we have some handy Fusion cheats that work on the FTM-200, FTM-300, and FTM-500 with more to come.

K9EQ_BB_FusionCheats