

MMDVM .96" OLED Now Works!

It has really been bugging me. I've never been able to get a 0.96" OLED to work on my Pi-star hotspots. The 1.3" works okay. Why not the 0.96". Here's why:

The 1.3" screen uses the SSD1306 controller. Adafruit uses this controller on all their OLED screens. The (cheap) Chinese OLED screens use an older SSD1106 controller.

Raspberry Pi uses the Adafruit OLED driver software, so Pi-star does also. Naturally Adafruit has no desire to support the SSD1106 as they don't sell anything that uses it. In other words, the standard Pi-star distribution OLED driver doesn't work with the SSD1106 .96" displays.

What's different? The SSD1106 has a slightly smaller refresh RAM array than the SSD1306. That causes each line of pixels to be off by 2 hence small valid text ends up looking like garbage. The graphics still look okay.

The solution is to replace the Adafruit driver with a modified version that supports the SSD1106. The driver is located at `/usr/local/lib/ArduiPi_OLED.so.1`. I have built a replacement driver with the SSD1106 support.

1. SSH into your Pi-Star (can be done from the Expert screen).
2. Switch to root with "sudo su".
3. Make the disk RW with the command "rpi-rw"
4. Move to the directory where the driver is installed with "cd /usr/local/lib".
5. Copy the driver file from HamOperator with the command:
"wget
http://HamOperator.com/files/libArduiPi_OLED.so.1.0".
6. Change the permissions on the driver with the command:
"chmod 777 /usr/local/lib/ArduiPi_OLED.so.1.0".

(I don't know why '777' '755' should work but the original was '777'.

7. Reboot

8. The OLED type needs to be '3' for the 0.96 display and '6' for the 1.3" display (set in Expert->MMDVM Host). The OLED must also be enabled from the configuration page

Thanks to [Charles](#) for doing the driver work!

If you want to download the file directly, you can do that here: [libArduiPi_OLED.so.1.0](#)

YSF Reflector Software

The HamOperator YSF Reflector is currently supporting MNWis, US Kentucky, and America-Link. A test reflector runs at US K9EQ.

A project has been going on at the HamOperator to improve the YSF Reflector software. Why do this?

The original software is very basic and offers very little capability and control. It also creates a HUGE problem when trying to bridge YSF hotspots to WiRES-X rooms because it just passes the data along with no filtering. That means whatever garbage gets sent to the reflector, it doesn't even need to be Fusion, that same data is sent to every connection including the bridged WiRES-X node. When the WiRES-X node gets this data it can misbehave or crash. Hence a project to fix these problems.

And while we're fixing problems, why not make it better by giving it more capabilities?

Current differences between the standard YSF Reflector software and the K9EQ version:

- Decodes all meta data including FICH and all data fields for every mode.
- Extensive logging allows selected logging of parameters needed to determine "what's going on?".
- Filters to drop packets if:
 - The FICH is invalid
 - The FICH does not pass sanity checks
 - Data Wide packets
 - Wires-X control packets
- Outputs a text file of connected nodes to simplify dashboards.
- Performance reporting
- Options can be changed in a *.ini file
- Ability to host multiple YSF Reflectors on a single server
- Internal documentation of the program (comments)
- Several builds are available:
 - Windows
 - Linux
 - Debian server (MNWis and US Kentucky are running in the "cloud")

Future enhancements will include:

- Drop problematic WiRES-X packets with invalid room/node number in the CSD1 field
 - Future improvement: Edit the packets to remove the problematic information
- Ability to black list callsigns and IP addresses.
- Kerchunk filter to drop short key-ups
- Remote control and status reporting of the reflector via Fusion text messages

- Integrated dashboard removes need for php and greatly reduces CPU overhead and network bandwidth
- Programmed reporting of audio levels informing users if their mic level is too high or too low
- Integration with IMRS
- New, engineered and reviewed API to more tightly integrate hotspots with the reflector
- Ability to programmatically send messages
- Ability to programmatically edit data fields on the fly (i.e., change GPS data, call sign, etc.)
- Improve internal program documentation
- Complete program re-write in Python. This will provide the ultimate in portability

The reflector will not support non-Fusion communications such as DMR or D*. The reason for this is that non-Fusion systems do not include all of the data that Fusion provides. This would force the reflector to become the lowest common denominator resulting in technology and feature restrictions.

The intent is for our version of YSF Reflector to be open sourced. It will be developed and enhanced with feedback from the Fusion community. By using good engineering practices, our hope is to provide a high quality service that equals or exceeds the existing quality of the Fusion/WiRES-X system. This is an enhancement to WiRES-X, not a replacement.

YSF WiRES-X Unintended Node Switching

A problem exists with YSF/MMDVM hotspots bridging to WiRES-X

rooms. Here's what happens.

A user has a hotspot and accesses it using a Fusion HT. They use the WiRES-X control mode to change the hotspot to different YSF Reflectors. Let's say the user wants to access MNWis (Room #21,493):

- They enter the number of the YSF Reflector, "US MNWis 21493", which is 37,624. The hotspot switches to US MNWis 21493 which is bridged to WiRES-X Room # 21,493.
- The user leaves their radio in the WiRES-X control mode. They hear stations, but when they talk, nobody answers and nobody is talking.

Here's what happened: When they keyed up to transmit, their radio sent a command to switch to Room #37,624. The Wires-X node that was previously connected to #21,493 now connects to #37,624 which belongs to a very nice gentleman in France.

How does this happen?

The Fusion transmitted data includes callsign fields CSD1, CSD2, and others. We're interested in CSD1. This 10-byte field contains two items: The room number of the connection and; The TxID or DP-ID of the radio. A typical CSD1 looks like 21493F0yxh where 21493 is the room number and F0yxh is the DP-ID. The '21493' in this field will cause a Wires-X node to switch to that room.

When the radio IS NOT in Wires-X control mode, the CSD1 field will be: '*****F0yxh' and no switch will be made.

Bottom line:

DON'T USE WiRES-X control mode to talk through a hotspot or MMDVM!

Workarounds and fixes: (Updated 23-Aug-2019)

Node operators who are experiencing this problem should use

the WiRES-X block feature to prevent the node from switching to the room that has the YSF reflector number.

For example: Assume that our node connects to MNWis, Room # 21,493. Our node also provides bridging to "YSF MNWis 21493" which is # 37,624.

- In the View->Node-Info(N) window press "Add".
- In the "Input ID" dialog box enter the YSF reflector number. In our example this is "37624".
- Press "OK" then "Close".
- When a station using the YSF bridge still has WiRES-X control mode enabled accesses the node, the command to switch rooms will be rejected by the WiRES-X software. The software will indicate a rejected attempt to switch rooms.

As an alternative, set the node so that connection changes are not allowed.

- File->Settings->Call Settings->Uncheck "Round Room Connection".
- Un check "Accept calls while in Round Room QSO.
- Check "Return to Room"
- Fill in the WiRES-X room number. Example: 21493.

YSF server side solutions:

I maintain a version of YSF Reflector that has filters to prevent hotspots causing problems with WiRES-X nodes. My first step is to drop all packets that contain an incorrect WiRES-X room number. Eventually I plan to replace the first 5 bytes of CSD1 with '*****'. This last step is difficult to do because it requires the reflector to decode the data, modify it, then recompute the CRC. This also involves working with the interleaving and forward error correction. It is obviously easier to just drop the packets, but that may confuse people on the YSF side since nobody on the WiRES-X side will hear them.

If you are interested in running the enhanced YSF Reflector software that will fix this problem at some point, please contact me privately.

73,

Chris, K9EQ

Old Stuff – Computers

How can computers be “old stuff”? We’ll see....

The Home First Computers

This week we take a look at three pioneers in the computer business:

- From [1975 MITS' catalog](#) featuring the Altair 8800 computer. (0.6 MB)
- From the [1977 Southwest Technical Products catalog](#) with their 6800-based computer. (5 MB)
- From [1978 The Commodore PET catalog](#) (0.5 MB)
- From [1979 The Texas Instruments TI-99](#) (1.2 MB)
- And from [1980 Ohio Scientific catalog](#) with their products. (7 MB)

How far we have come! Be sure to check out memory size and prices!

Old Stuff – Reference

Techniques and Projects

Here is a variety of new old stuff that focuses on electronics and radio techniques and projects.

- 1958 From Popular Electronics magazine, [The Experimenter's Handbook](#) (15 MB)
- 1965 Again from Popular Electronics, [The Experimenter's Handbook](#) (8 MB)
- 1973 [Electronics Theory Handbook](#) (4 MB)
- 1973 [Understanding Electronic Circuits](#) (7 MB)

Handbook, Dictionary, and FixIt

For the New Old Stuff for this week, we present some reference materials.

- The [1936 ARRL Handbook 14th Edition](#) (54 MB)
- The [1944 Allied Dictionary of Radio Terms](#) (4 MB)
- From Popular Science in 1946, the [Third Radio Make It, Fix It Annual](#) (6 MB)

Tubes, Tubes, tubes....

- From 1951, a manual from Eimac, [The Care and Feeding of Power Tetrodes](#) (3 MB) You are feeding your tetrodes, aren't you?
- 1960 offers a document on [Getting the Most out of Vacuum Tubes](#) (13 MB)
- From 1968, the [RCA Tube Manual](#) (28 MB)

Old Stuff – Broadcasting

All About FM

This week we have a couple of documents from the early 1940's explaining FM. At this time FM was very, very new. It had the benefit of reducing the noise one would hear on AM. A lot of work went into convincing the world that FM was the way of the future.

[1941-All about FM](#) (3 MB)

[1941-GE FM Primer](#) (4 MB)

AM Broadcast Transmitters

Some classic (and one BIG) broadcast transmitters.

[Raytheon RA-250](#) Popular with 250 watt small town, low budget stations (1.2 MB)

[Bauer 707](#) Popular with the 1KW day / 250 night stations. Offered as a kit! (2 MB)

[Continental 105C](#) A 1,000 KW transmitter! (8 MB)

The Story of Stereo 1960

From 1960 the story of stereo from having two ears to stereo LP's and FM!

[The Story of Stereo 1960](#) (8 MB)

Old Stuff – Catalogs

Allied Radio Catalogs

[PDF of the 1969 Allied Radio Catalog](#) (50 MB)

Radio Shack Catalogs

So what exactly is a Meissner Signal Shifter? What did it cost? How about a 2 and one half meter transceiver? And where could you buy one in 1946? Guess what? Radio Shack. THEY SOLD RADIOS! And they had a radio shack.

[Radio Shack Catalog 1939](#) (6 MB)

[Radio Shack Catalog 1946](#) (11 MB)

[1986 Radio Shack](#) (20 MB)

Lafayette

[1968 Lafayette Radio Electronics](#) (16 MB)

Heathkit

[1976 Heathkit](#) (71 MB)

WiRES-X Automation

Yaesu does not provide a mechanism that allows the WiRES-X software to be controlled by another program, i.e., having another program switch to a certain Room when a net starts.

Windows does, however, permit another program to send events to a program. Each window, menu item, and dialog in a Windows program has a unique identifier. It is possible to use these identifiers to send "message" to the WiRES-X software.

The WiRES-X Automation Project's purpose is to bring together people who are interested in developing this technology and sharing their results.

To get things going, here are two mechanisms for automating WiRES-X:

1. AutoIT: <http://www.autoitscript.com>
2. Python – an excellent programming language found at python.org

Dave, N9TOW, provided the following information:

Packages I have installed on my WiresX system.

```
C:\Users\WiresX>pip list
```

```
comtypes (1.1.3)
```

```
pip (9.0.1)
```

```
pywin32 (220)
```

```
pywinauto (0.6.3)
```

```
setuptools (28.8.0)
```

```
six (1.10.0)
```

```
https://github.com/pywinauto/pywinauto
```

To install

```
pip install -U pywinauto
```

Script that executes changing channels on WiresX app

```
import time
from pywinauto import Application
app = Application().connect(path="C:\Program Files
(x86)\YAESUMUSEN\WIRES-X\wires-X.exe")
app.WiresX.menu_select("Connect(C)->Connect To(T)")
```

```
time.sleep(1.5)
app.InputID.Edit.set_edit_text("21493")
time.sleep(.5)
app.InputID.OK.click()
time.sleep(4)
dialogs = app.windows()
##app.Dialog.CloseButton.click()
```

DV4mini Fusion Monitor Program

This program uses a DV4mini to monitor over-the-air Fusion signals and display the meta data (does not display voice, pictures, messages, etc.)

This version of the program will output a record when a station keys up and then another record when the station unkeys. I've done this so that the user doesn't get overwhelmed with all of the meta data that is produced.

You'll need to review the [Yaesu Digital Communication Standards](#) document to understand what the different fields mean.

Computer Requirements: Windows 7 or later

Program Installation

1. Download the program from this link: [DV4mini YSFMonitor](#).
2. Create a directory on your Windows PC and unzip this folder into that directory.
3. Download and install the Microsoft Visual Studio 2017

redistributable for your machine

Running the Program

1. Open a command box (type cmd) and change to the directory in which you installed the software.
 2. Check the program: Type VSFRX.exe -v. It should respond with the version number.
 3. User Device Manager to determine the com port of your DV4mini.
 4. Enter frequency and com port as follows: VSFRX.exe com7 444525000
- Note that the frequency must be entered as 9 digits.
5. A log file will be created in the same directory as the program.

Output Example from the 30-Apr-2018 Net

```
M: 2018-05-01 01:01:30.338 DV4mini version: V01.77
M: 2018-05-01 01:01:30.338 VSFRX-20180430 K9EQ starting
M: 2018-05-01 01:01:33.646
M: 2018-05-01 01:01:33.646 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 6 SQ: 0 SC: 0
M: 2018-05-01 01:01:33.646 Terminator, CSD1
M: 2018-05-01 01:01:33.646 0000: 2A 2A 2A 2A 2A 48 35 35 37 56
57 38 4F 4A 2D 20 20 20 20 20 *****H557VW80J- *
M: 2018-05-01 01:01:33.646 Terminator, CSD2
M: 2018-05-01 01:01:33.646 0000: 4B 39 45 51 20 20 20 20 20 20
57 38 4F 4A 20 20 20 20 20 20 *K9EQ W80J *
M: 2018-05-01 01:01:36.748
M: 2018-05-01 01:01:36.748 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:01:36.748 Header, CSD1
M: 2018-05-01 01:01:36.748 0000: 2A 2A 2A 2A 2A 46 30 58 49 4B
4B 44 38 47 52 4E 20 20 20 20 *****F0XIKKD8GRN *
M: 2018-05-01 01:01:36.748 Header, CSD2
M: 2018-05-01 01:01:36.748 0000: 4B 39 45 51 20 20 20 20 20 20
4B 44 38 46 4A 48 20 20 20 20 *K9EQ KD8FJH *
M: 2018-05-01 01:01:43.742
M: 2018-05-01 01:01:43.742 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:01:43.743 Terminator, CSD1
```

M: 2018-05-01 01:01:43.744 0000: 2A 2A 2A 2A 2A 46 30 58 49 4B
4B 44 38 47 52 4E 20 20 20 20 *****F0XIKKD8GRN *
M: 2018-05-01 01:01:43.745 Terminator, CSD2
M: 2018-05-01 01:01:43.745 0000: 4B 39 45 51 20 20 20 20 20 20
4B 44 38 46 4A 48 20 20 20 20 *K9EQ KD8FJH *
M: 2018-05-01 01:01:48.921
M: 2018-05-01 01:01:48.921 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:01:48.921 Header, CSD1
M: 2018-05-01 01:01:48.921 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
57 36 5A 44 52 20 20 20 20 20 *****W6ZDR *
M: 2018-05-01 01:01:48.921 Header, CSD2
M: 2018-05-01 01:01:48.921 0000: 4B 39 45 51 20 20 20 20 20 20
44 55 31 5A 44 52 20 20 20 20 *K9EQ DU1ZDR *
M: 2018-05-01 01:01:59.513
M: 2018-05-01 01:01:59.513 FICH: FI: HC, DT: VD1, BN: 0, BT:
1, FN: 2, FT: 6 SQ: 0 SC: 0
M: 2018-05-01 01:02:00.317
M: 2018-05-01 01:02:00.317 FICH: FI: TC, DT: VD2, BN: 3, BT:
1, FN: 3, FT: 3 SQ: 0 SC: 27
M: 2018-05-01 01:02:01.717
M: 2018-05-01 01:02:01.717 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:01.717 Terminator, CSD1
M: 2018-05-01 01:02:01.717 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
57 36 5A 44 52 20 20 20 20 20 *****W6ZDR *
M: 2018-05-01 01:02:01.717 Terminator, CSD2
M: 2018-05-01 01:02:01.717 0000: 4B 39 45 51 20 20 20 20 20 20
44 55 31 5A 44 52 20 20 20 20 *K9EQ DU1ZDR *
M: 2018-05-01 01:02:02.483
M: 2018-05-01 01:02:02.483 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:04.203
M: 2018-05-01 01:02:04.203 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:04.203 Terminator, CSD1
M: 2018-05-01 01:02:04.203 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
4B 44 38 41 47 4F 4A 4F 48 4E *****KD8AGOJOHN*
M: 2018-05-01 01:02:04.203 Terminator, CSD2
M: 2018-05-01 01:02:04.203 0000: 4B 39 45 51 20 20 20 20 20 20
41 44 30 4D 49 20 20 20 20 20 *K9EQ AD0MI *

M: 2018-05-01 01:02:09.834
M: 2018-05-01 01:02:09.834 FICH: FI: HC, DT: VD2, BN: 0, BT: 0, FN: 0, FT: 6 SQ: 0 SC: 0
M: 2018-05-01 01:02:09.834 Header, CSD1
M: 2018-05-01 01:02:09.834 0000: 32 31 34 39 33 47 30 32 7A 42 41 42 38 52 4C 2D 54 4F 4D 20 *21493G02zBAB8RL-TOM *
M: 2018-05-01 01:02:09.834 Header, CSD2
M: 2018-05-01 01:02:09.834 0000: 4B 39 45 51 20 20 20 20 20 20 41 42 38 52 4C 20 20 20 20 20 *K9EQ AB8RL *
M: 2018-05-01 01:02:19.424
M: 2018-05-01 01:02:19.424 FICH: FI: TC, DT: VD2, BN: 0, BT: 0, FN: 0, FT: 6 SQ: 0 SC: 0
M: 2018-05-01 01:02:19.424 Terminator, CSD1
M: 2018-05-01 01:02:19.424 0000: 32 31 34 39 33 47 30 32 7A 42 41 42 38 52 4C 2D 54 4F 4D 20 *21493G02zBAB8RL-TOM *
M: 2018-05-01 01:02:19.424 Terminator, CSD2
M: 2018-05-01 01:02:19.424 0000: 4B 39 45 51 20 20 20 20 20 20 41 42 38 52 4C 20 20 20 20 20 *K9EQ AB8RL *
M: 2018-05-01 01:02:23.581
M: 2018-05-01 01:02:23.581 FICH: FI: HC, DT: VD2, BN: 0, BT: 0, FN: 0, FT: 6 SQ: 0 SC: 0
M: 2018-05-01 01:02:23.583 Header, CSD1
M: 2018-05-01 01:02:23.584 0000: 32 31 34 39 33 45 30 50 71 61 4B 46 38 50 4D 2F 41 4C 41 4E *21493E0PqaKF8PM/ALAN*
M: 2018-05-01 01:02:23.584 Header, CSD2
M: 2018-05-01 01:02:23.584 0000: 4B 39 45 51 20 20 20 20 20 20 4B 46 38 50 4D 20 20 20 20 20 *K9EQ KF8PM *
M: 2018-05-01 01:02:37.296
M: 2018-05-01 01:02:37.296 FICH: FI: TC, DT: VD2, BN: 0, BT: 0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:37.296 Terminator, CSD1
M: 2018-05-01 01:02:37.298 0000: 32 31 34 39 33 45 30 50 71 61 4B 46 38 50 4D 2F 41 4C 41 4E *21493E0PqaKF8PM/ALAN*
M: 2018-05-01 01:02:37.298 Terminator, CSD2
M: 2018-05-01 01:02:37.299 0000: 4B 39 45 51 20 20 20 20 20 20 4B 46 38 50 4D 20 20 20 20 20 *K9EQ KF8PM *
M: 2018-05-01 01:02:51.164
M: 2018-05-01 01:02:51.165 FICH: FI: HC, DT: VD2, BN: 0, BT: 0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:51.166 Header, CSD1
M: 2018-05-01 01:02:51.167 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A

57 35 4C 4E 41 2D 4B 45 56 4E *****W5LNA-KEVN*
M: 2018-05-01 01:02:51.168 Header, CSD2
M: 2018-05-01 01:02:51.169 0000: 4B 39 45 51 20 20 20 20 20 20
57 30 4D 44 54 20 20 20 20 20 *K9EQ W0MDT *
M: 2018-05-01 01:02:57.665
M: 2018-05-01 01:02:57.665 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:02:57.667 Terminator, CSD1
M: 2018-05-01 01:02:57.668 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
57 35 4C 4E 41 2D 4B 45 56 4E *****W5LNA-KEVN*
M: 2018-05-01 01:02:57.668 Terminator, CSD2
M: 2018-05-01 01:02:57.669 0000: 4B 39 45 51 20 20 20 20 20 20
57 30 4D 44 54 20 20 20 20 20 *K9EQ W0MDT *
M: 2018-05-01 01:03:00.908
M: 2018-05-01 01:03:00.908 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:03:00.908 Header, CSD1
M: 2018-05-01 01:03:00.908 0000: 2A 2A 2A 2A 2A 45 35 67 47 79
4B 31 4B 43 2D 50 4F 52 54 32 *****E5gGyK1KC-PORT2*
M: 2018-05-01 01:03:00.908 Header, CSD2
M: 2018-05-01 01:03:00.908 0000: 4B 39 45 51 20 20 20 20 20 20
4B 31 4B 43 20 20 20 20 20 20 *K9EQ K1KC *
M: 2018-05-01 01:03:07.414
M: 2018-05-01 01:03:07.414 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:03:07.414 Terminator, CSD1
M: 2018-05-01 01:03:07.414 0000: 2A 2A 2A 2A 2A 45 35 67 47 79
4B 31 4B 43 2D 50 4F 52 54 32 *****E5gGyK1KC-PORT2*
M: 2018-05-01 01:03:07.414 Terminator, CSD2
M: 2018-05-01 01:03:07.414 0000: 4B 39 45 51 20 20 20 20 20 20
4B 31 4B 43 20 20 20 20 20 20 *K9EQ K1KC *
M: 2018-05-01 01:03:29.710
M: 2018-05-01 01:03:29.710 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:03:29.710 Header, CSD1
M: 2018-05-01 01:03:29.710 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
4B 30 4F 52 4B 2D 43 48 55 4B *****K0ORK-CHUK*
M: 2018-05-01 01:03:29.710 Header, CSD2
M: 2018-05-01 01:03:29.710 0000: 4B 39 45 51 20 20 20 20 20 20
57 30 4D 44 54 20 20 20 20 20 *K9EQ W0MDT *
M: 2018-05-01 01:03:37.764 TIMEOUT

M: 2018-05-01 01:04:24.355 TIMEOUT
M: 2018-05-01 01:04:44.738 TIMEOUT
M: 2018-05-01 01:04:53.392
M: 2018-05-01 01:04:53.392 FICH: FI: TC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:04:53.394 Terminator, CSD1
M: 2018-05-01 01:04:53.394 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
4B 30 4F 52 4B 2D 43 48 55 4B *****K00RK-CHUK*
M: 2018-05-01 01:04:53.395 Terminator, CSD2
M: 2018-05-01 01:04:53.395 0000: 4B 39 45 51 20 20 20 20 20 20
57 30 4D 44 54 20 20 20 20 20 *K9EQ W0MDT *
M: 2018-05-01 01:04:55.917
M: 2018-05-01 01:04:55.917 FICH: FI: HC, DT: VD2, BN: 0, BT:
0, FN: 0, FT: 7 SQ: 0 SC: 0
M: 2018-05-01 01:04:55.917 Header, CSD1
M: 2018-05-01 01:04:55.917 0000: 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
4B 47 34 53 42 47 2D 44 45 4E *****KG4SBG-DEN*
M: 2018-05-01 01:04:55.917 Header, CSD2
M: 2018-05-01 01:04:55.917 0000: 4B 39 45 51 20 20 20 20 20 20
57 38 4F 4A 20 20 20 20 20 20 *K9EQ W80J *

Listen to MNWis Online

You can now listen to MNWIS online. Just follow this link with your favorite browser:

[MNWIS Online](#)

As an alternative, you can listen to MNWis using the Broadcastify app on your smart device. Download Broadcastify and search for "MNWIS".

[Apple Download](#)

[Android Download](#)