# Hacking with Hackerboxes

Category: Projects
May 3, 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
May 3, 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

---

# Crimp Vs. Solder

Category: Projects
May 3, 2026
What’s best, crimping or soldering. Should you use crimping? How is a good crimp done.

Below are references that address these questions.

Wikipedia Entry

Crimping vs Soldering Advantages and Disadvantages

Good and Bad RF Crimp Connections

ARRL Instructions for Crimping RF Connectors

How to Make a Quality Crimp

More on Making a Quality Crimp (technical)

Molex Crimping Manual (technical)

TE Systems Crimp Chart

Notes on crimping:

It is always best to have the right tools. Make sure you use the right connector for the cable and the right crimp die for the connector. With the right tools, RF connectors can be crimped very quickly and are very reliable. One nice advantage, is that it is easy to reuse the connector. The center pin is generally soldered (which is easy to unsolder) and the shield is crimped with a short metal “tube”. Some companies sell bags of replacement “tubes” so the connectors can be reused. Follow the connector’s instructions EXACTLY and you’ll have a solid connection that will last forever.

If you are making outdoor connections, I recommend using N connectors as they are hermetically sealed. (PL-259s are not) For power connections, I recommend using the same connectors that are used to connect solar panels together. They’re much cheaper than Anderson Power Poles and they form a hermetic seal. For signal connectors, go to your hardware store and look in the irrigation section. They have special twist-on connectors that are meant to be underground or sit in the rain. They’re pretty cheap if you buy a big bag.

Check back again in about a week and I’ll have more links and references.

---

# WiRES-X Automation

Category: Automation,Projects,WiRES-X
May 3, 2026

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)
pypiwin32 (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()
```

# Python Program

Bill, W9LBR, developed a Python program that runs on the WiRES-X computer. The software has been updated as of 12-Feb-2024. See more here.

This is from his description:

WXscheduler.pyw is a Python 3 program that automates certain aspects of
Wires-X Room/Node connections according to a userrdefined schedule.
To that end WXscheduler.pyw must be run on the same Windows PC that hosts the Wires-X application controlling either an HRI-200 or
a USB connected radio that supports PDN operation.

Schedulable operations:
– Connect

– Disconnect
– SetUnlimitedTOT
– SetTimeoutTOT
– Restart Wires-X App

Manual operation: Button that causes a Force Disconnect

Bonus operation: Displays Last Heard information that the Wires-X app
updates once a minute. Converting Lat/Lon into Grid Square if available.

A zip file containing instructions and the Python program is available at WXScheduler.

Update History

2 June 2023 v1.3 updates: – Last Heard now displays FT5-D, FTM-200, and FTM-500 correctly – Desktop\Wires-X_Last_Heard.html now generated to be viewed with PC browser

12 Feb 2022 updates: – Corrected grid square rendering of GPS coordinates as displayed in the Last Heard frame – version (v1.2) now displayed in main window title

5 Feb 2022 minor updates: – Fixed Last Heard display of recently manufactured FT70-D radios – WXscheduler.txt is now in DOS format, making it easier to read with Notepad 2022 version of WXscheduler.pyw – A scheduled event now performs multiple actions – Added File->Settings->Call settings actions – Fixed issue when Wires-X App is minimized – Fixed issue when Wires-X Settings window left open

# DV4mini Fusion Monitor Program

Category: Hotspots,Projects
May 3, 2026
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: YSFRX.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 YSFRX-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 ******H557VW8OJ- *
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 W8OJ *
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 **********K0ORK-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 W8OJ *