



(RTTY) Contesting on GNU/Linux with QARTest

Serge Stroobandt, ON4AA

Copyright 2017–2021, licensed under [Creative Commons BY-NC-SA](#)

QARTest, even though written in Italian, is interesting because it supports many European RTTY contests and the software is kept maintained. Running **QARTest** with **PlayOnLinux** on GNU/Linux looks promising. However, I have not yet managed to run it fully stable **as I did with Win-Test**. Hence, the information contained on this page should be considered as experimental. I will update this page as I make progress. **Drop me a line** when you managed to get **QARTest** running in a stable fashion on GNU/Linux.

Introduction

RTTY is a great mode for contesting. There is no need to type a lot, nor do you need to shout your voice hoarse. What is more, you even do not need to listen to RTTY stations in order to work them. These characteristics render late-night RTTY contesting especially gentle towards family members and nearby neighbours. The required signal-to-noise ratio lies between a moderately low -5 to -9 dB @ 2500 Hz; in any case, much lower than the +10 dB for SSB!

However, getting set for RTTY contesting under GNU/Linux is kind of challenging.

Previously, I have been using **Win-Test** on **Xubuntu LTS**.



QARTest TODO

I prefer the use of **PlayOnLinux** — a front-end for Wine — over using bare metal Wine. **PlayOnLinux** offers the benefit of installing unrelated Windows™ programs on separate virtual drives, also called «bottles» in **CrossOver**, a commercial offering similar to the free (as in beer) **PlayOnLinux**. The use of virtual

PlayOnLinux drives keeps things uncluttered and limits the damage when any Windows™ program goes awry.

microHAM Digi Keyer

If you own a modern transceiver with a USB port which presents audio as a sound device, you may skip this part.

Many transceiver interfaces work with GNU/Linux. However, as with all hardware under GNU/Linux, always check for Linux support on the web before acquiring a new interface! Because they are quite popular and I had one lying around, this article will focus on the excellent **microHAM** interface products.



Figure 1: Front panel of the **microHAM** Digi Keyer

The good news is **Matthias (Max) Moeller, DJ5QV**, was so kind to code and share the **mhuxd microHAM Unix daemon**. In doing so, he could count with the help of **Joe Subich, W4TV** of **microHAM** for providing protocol specifications. The daemon should support all **microHAM** interface products.

On Ubuntu-derived systems, it is best to install **mhuxd** by adding its **PPA repository**. This will ensure updates are installed automatically when they become available. Type the following commands (without the \$ prompt) in a terminal window:

```
$ sudo add-apt-repository ppa:dj5qv/mhuxd-0.5
$ sudo apt-get update
$ sudo apt-get install mhuxd
```

The **mhuxd** daemon should immediately start running in the background. You can check that and perform the initial configuration by opening the following URL in your browser: <http://localhost:5052> Bookmark this URL as you may need it again occasionally. Note that **microHAM** interfaces are powered by the transceiver; not the computer.

The screenshot shows the 'mhuxd Device Router' web interface. At the top, there are navigation tabs for 'Home', 'Daemon', and 'DIGI KEYER'. The 'DIGI KEYER' tab is active. Below the navigation, there is a breadcrumb trail 'Home » Summary'. The main content area is titled 'Summary' and contains two sections: 'Summary' and 'Keyer List'.

The 'Summary' section displays the following information:

| | |
|-------------|--------------------------|
| Version: | mhuxd 0.50rc3 |
| Hostname: | e8500 |
| Process ID: | 1937 |
| Log File: | /var/log/mhuxd/mhuxd.log |
| Loglevel: | INFO |

The 'Keyer List' section displays a table with the following data:

| Name | Serial | Firmware | Status |
|------------|----------|----------|--------|
| DIGI KEYER | DKQB2QDR | 5.4 | ONLINE |

Figure 2: Summary screen of the mhuxd daemon available at <http://localhost:5052>, showing the keyer's firmware version.

Caveat: Upgrade to the latest firmware!

Once mhuxd is installed, first, **make absolutely sure your microHAM keyer is running the latest firmware!** Failing to do so, guarantees problems with MMTTY PTT keying over the FSK port —as I experienced first-hand.

The firmware for the micro/Digi/CW keyers (MK/CK/DK) goes under the name of mmk. Hidden deep within its change file `mmk_change_log.txt`, the November 22, 2007 entry reads:

«Some changes in keyer protocol and settings format was introduced to be compatible with router v5.1.0. Hence firmware **MUST** be upgraded and power-up settings **MUST** be again stored to EEPROM of keyer.»

Installing new keyer firmware is the one and only instance where you are probably quicker off borrowing a Windows™ computer. In less than 10 minutes, you get the job done. Just install the Windows™ **microHAM USB Device Router** and let this program upgrade your keyer to the latest version of the appropriate firmware. Keep your transceiver and cabling at hand too, because —remember— the keyer gets its power from the transceiver.

microHAM port routing

In this step, the CAT, PTT and FSK channels of the microHAM keyer will be assigned to a Linux device virtual serial port. Also make sure to tick off RTS-PTT for each of these virtual serial ports.

Here is why: MMTTY is only able to employ a single COM port for both PTT T/R and FSK tone keying. Instead of using the dedicated PTT1 port, MMTTY will need to key PTT with a RTS (ready to send) signal on the FSK1 port. (See [below](#).) It is a good idea to also configure RTS-PTT for the other virtual serial ports for use with other programs.

Table 1: Digi Keyer port routing

| channel | RTS-PTT | virtual serial port |
|---------|-------------------------------------|---------------------|
| CAT1 | <input checked="" type="checkbox"/> | /dev/mhuxd/cat1 |
| PTT1 | <input checked="" type="checkbox"/> | /dev/mhuxd/ptt1 |
| FSK1 | <input checked="" type="checkbox"/> | /dev/mhuxd/fsk1 |

The screenshot shows the 'mhuxd Device Router' web interface. The top navigation bar includes 'Home', 'Daemon', and 'DIGI KEYER'. The main content area is titled 'Daemon Ports Configuration and Routing'. It features a 'Port List' table with the following data:

| ID | Type | Port / Device | Status | Remote Access | RTS-PTT | DTR-PTT | Destination Name | Destination Serial | Channel |
|----------------------------|------|---------------|--------|---------------|-------------------------------------|--------------------------|------------------|--------------------|---------|
| <input type="checkbox"/> 1 | VSP | cat1 | ok | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | DIGI KEYER | DKQB2QDR | CAT1 |
| <input type="checkbox"/> 2 | VSP | ptt1 | ok | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | DIGI KEYER | DKQB2QDR | PTT1 |
| <input type="checkbox"/> 3 | VSP | fsk1 | ok | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | DIGI KEYER | DKQB2QDR | FSK1 |

Below the table is a 'Remove' button. Underneath is an 'Add Port' section with two radio buttons for 'Port Type': 'VSP Virtual Serial Port' (selected) and 'TCP Network Port'. Below that is a 'Destination Keyer' section with a radio button for 'DIGI KEYER (DKQB2QDR)' (selected). A 'Continue' button is at the bottom of the 'Add Port' section.

Figure 3: Configuration and routing of the `mhuxd` daemon virtual serial ports for the `microHAM Digi Keyer`

To make the `mhuxd` device ports accessible from your account, add your username to the `mhuxd` group. Log out for this change to take effect.

```
$ sudo adduser $USER mhuxd
```

In the rare event of a `mhuxd` daemon crash, there is no need to restart your computer. Typing the following command in a terminal window should do as long as your system is not using `systemd`.

```
$ sudo service mhuxd restart
```

Specifying the CAT protocol

The CAT1 port on the microHAM interface needs to be informed about the serial protocol parameters which are specific to your transceiver brand and model. Consult your transceiver manual for the proper settings. This CAT port will be used by QARTest —not MMTTY— to read and control your transceiver.

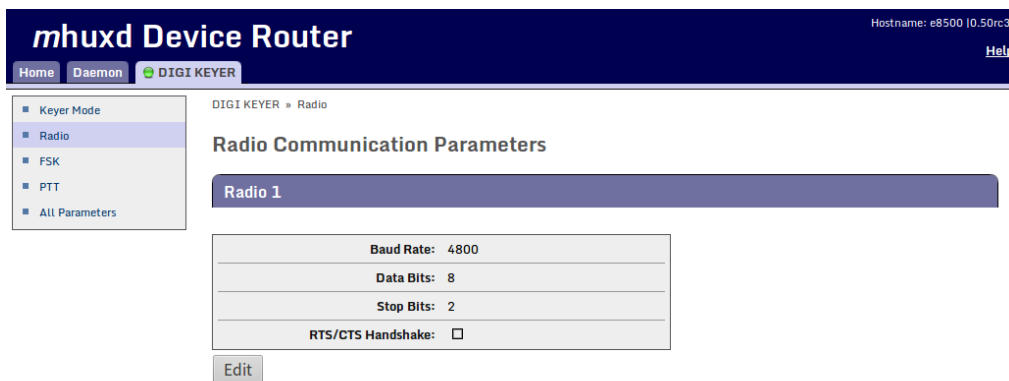


Figure 4: Computer-aided transceiver protocol parameters for interfacing with the Yaesu FT-990. Consult your transceiver manual for the proper settings.

FSK parameters

Standard FSK parameters for amateur radio TTY service are shown below. Other services (press, diplomatic, military) may require different parameters.

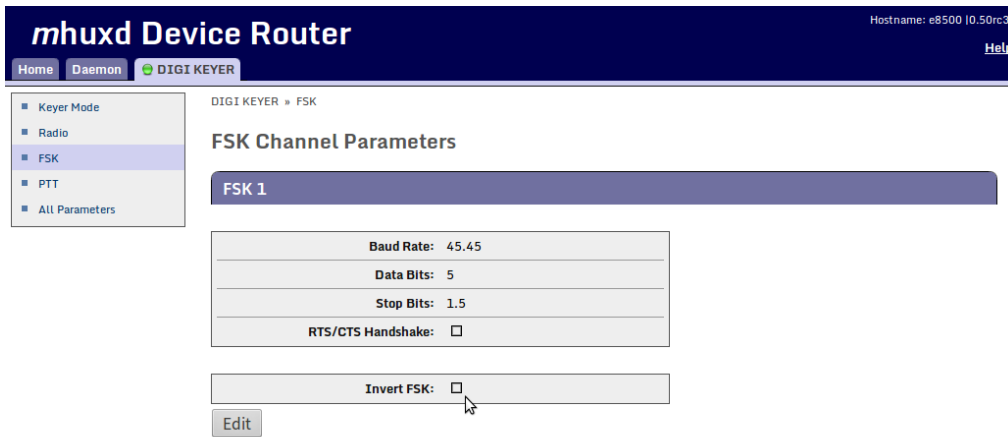


Figure 5: FSK parameter settings for RTTY, with an option to inverse the FSK stream on transmit.

Setting the keyer mode

Keyer mode is the only setting which requires changing each time one switches between operating modes (e.g. when switching from real FSK for MMTTY to DIGITAL AFSK for use with **Fldigi**). On newer microHAM interfaces (DK2, MK2 and MK2R+), it is more practical to opt for the Keyer Mode Follows RIG automation. This will eliminate once and for all the need to revisit this mhuxd configuration page.

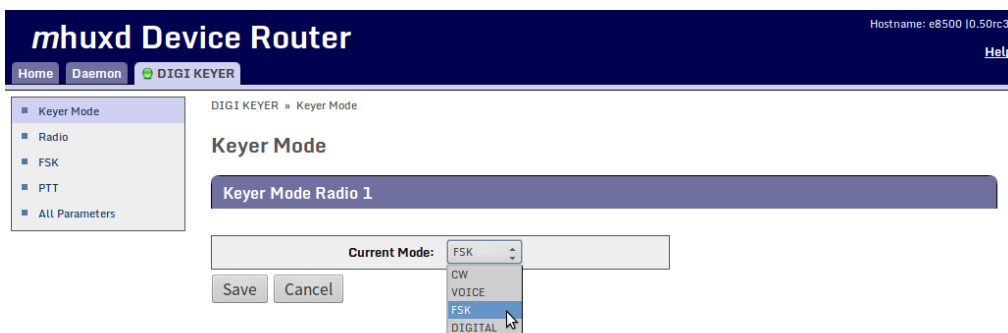


Figure 6: Setting the keyer mode for real FSK keying with MMTTY.

Transceiver settings

- Whenever possible, always use truly keyed FSK on transmit. Because of confusing FT-990 CAT messaging, I am required to always operate my Yaesu FT-990 in RTTY-USB mode.
- To protect the transceiver PA from overheating, reduce the RF PWR to 50%.
- If one of the tones falls outside the (250 Hz) bandpass filter, use the transceiver's SHIFT knob to correct this.
- For fine tuning, employ RX CLAR and TX CLAR simultaneously on the FT-990.
- I prefer to listen to lower pitched "European" RTTY tones instead of the US default. On the FT-990, this is set using DIP switches.

Table 2: Mark and shift settings

| style | MARK | shift | SPACE |
|-------------------|---------|--------|---------|
| US default | 2125 Hz | 170 Hz | 2295 Hz |
| EU easy listening | 1275 Hz | 170 Hz | 1445 Hz |

Caveat: Reversed FSK on RX implies inverting FSK on TX!

My Yaesu FT-990 insists on designating RTTY-LSB as reversed or RTTYR in its CAT communication with my computer. This is cumbersome for logging QSOs on LoTW which does not know how to handle RTTYR as a mode. Hence, I operate this transceiver always in its RTTY-USB mode. This means Rev. on reception is always on in MMTTY **and** either my keyer or my transceiver is set to invert the FSK transmission stream. I chose to do the latter by flipping a DIP switch on the FT-990. Remember: If nobody answers your calls, chances are you are sending in reverse!

PlayOnLinux



As mentioned earlier, PlayOnLinux is a neat front-end for Wine, offering the possibility to install unrelated Windows™ programs on segregated virtual drives. Being included with the Xubuntu LTS repositories, installing PlayOnLinux is straightforward:

```
$ sudo apt-get install playonlinux
```

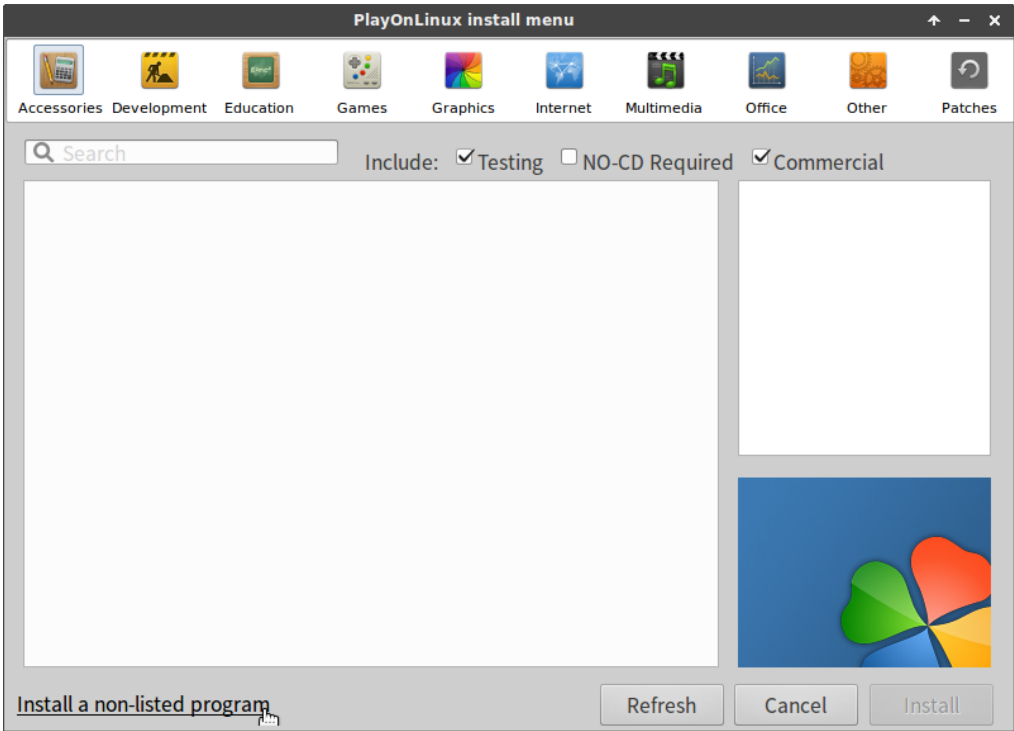


Figure 7: After clicking the `+ Install` button, click on «Install a non-listed program» at the bottom of the window.

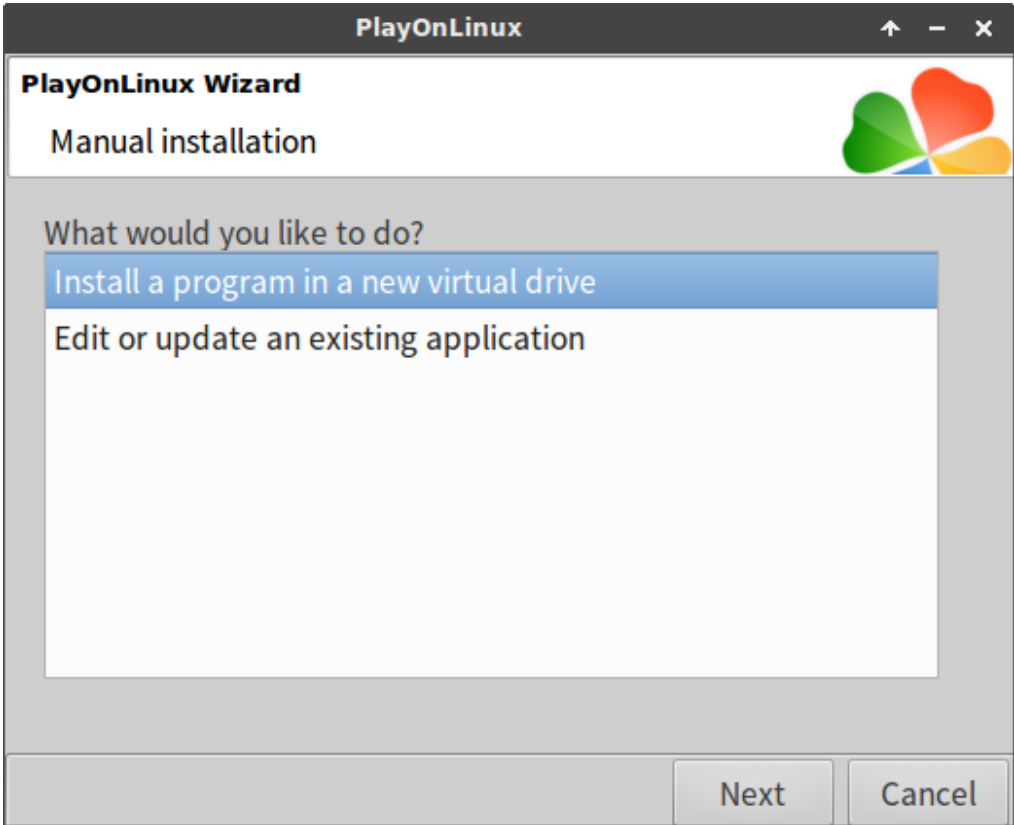


Figure 8: Choose «Install a program in a new virtual drive» and click on `Next`.

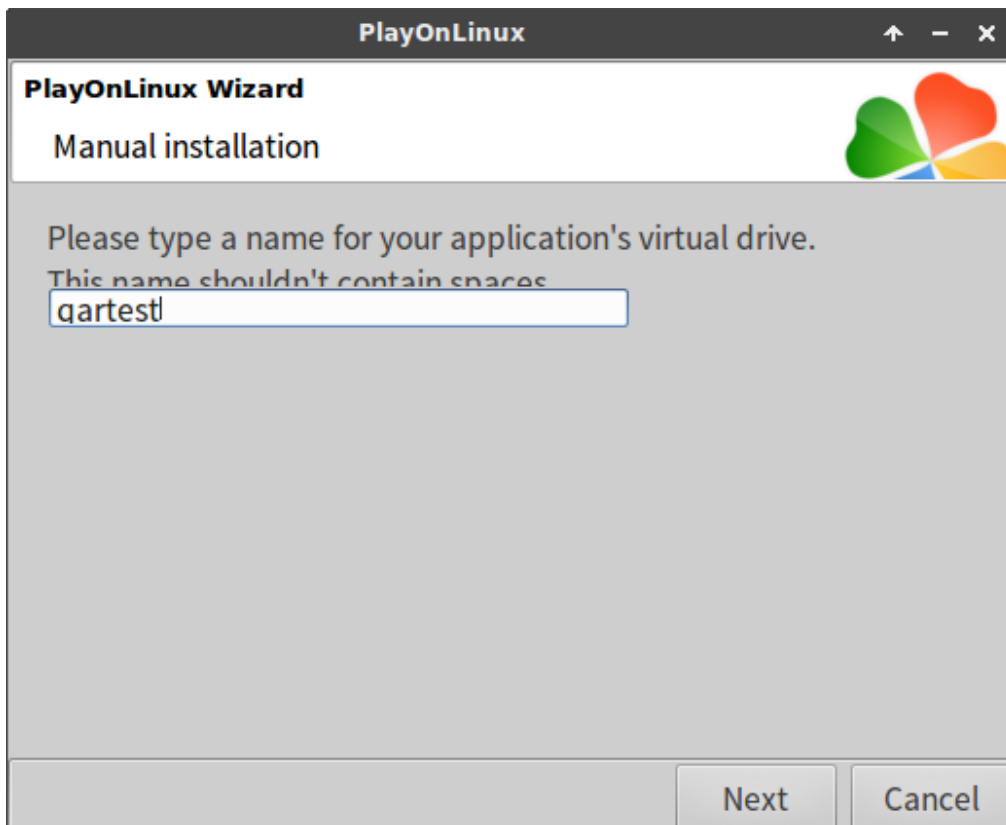


Figure 9: Type qartest and click on .

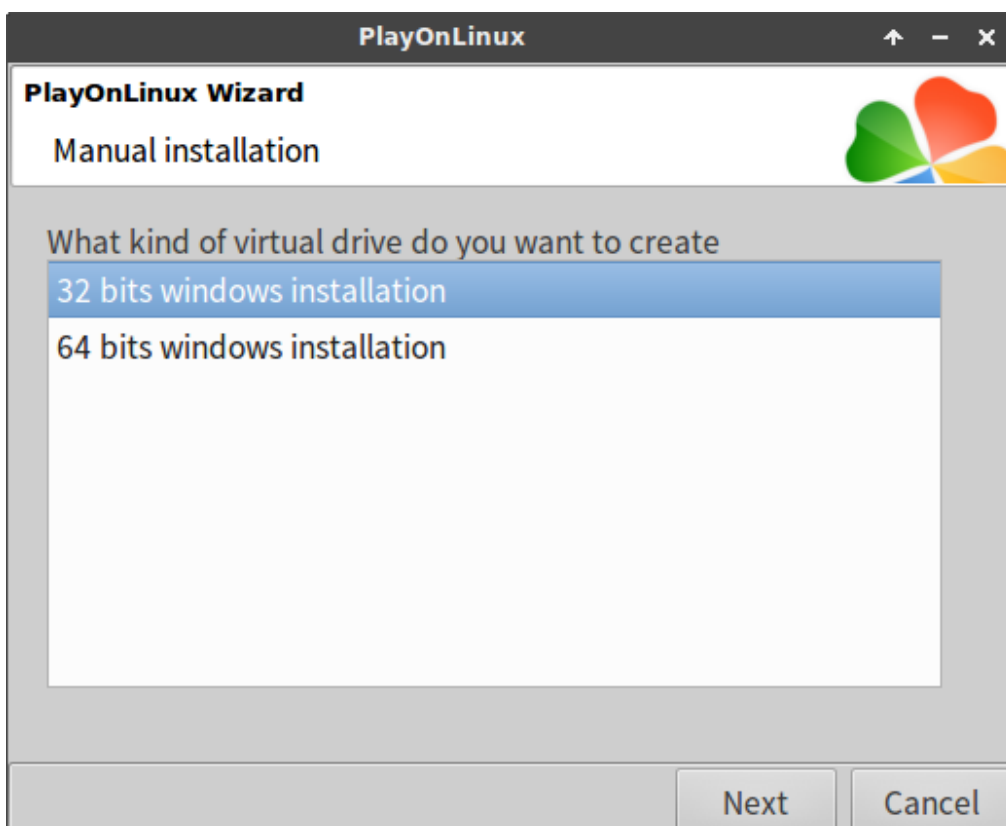


Figure 10: Choose «32 bits windows installation» and click on .

MMTTY installation



MMTTY is installed first, on a PlayOnLinux virtual drive called `qartest`. **MMTTY** and **QARTest** will share this virtual drive because both programs are related and need to be aware of one another. **MMTTY** allows for **hard FSK keying**, which in a contest environment is **less prone to errors** than modulating audio AFSK.

Caveat: Do not install EXTFSK!

On the web, you may find conflicting advice about installing the **EXTFSK** **MMTTY** extension. Here is **the definitive answer** by Joe Subich, W4TV, of **microHAM**:

«***NEVER*** use **EXTFSK** with **microHAM** **microKEYER**, **DigiKeyer**, **microKEYER II**, **DigiKeyer II**, **MK2R+** or **micro2R**. All of those interfaces provide a **UART compatible, virtual port for FSK** that handles 45, 50, 75 and 100 baud **RTTY (baudot)** directly. **EXTFSK** is not compatible with the **FSK output of the microHAM “keyer” devices.**»

Wine COM port mapping

Both **MMTTY** and **Win-Test** carry along their own drivers to communicate with the **CAT** interface of most common transceivers. These drivers function perfectly with **PlayOnLinux**, provided the Linux **mhuxd** device ports appear as **COM** ports to the **Windows™** programs on the **win-test** virtual drive.

I reiterate that as a **GNU/Linux** user, **one needs to make oneself member of the mhuxd group** in order to use these virtual ports.

Old wine versions, prior to version 2.8

Up to **wine** version 2.8, symbolic links from the Linux device ports to the typical **Windows™** **COM** ports can be created directly in the directory `~/.PlayOnLinux/wineprefix/qartest/dosdevices` on the **qartest** virtual drive:

```

$ cd ~/.PlayOnLinux/wineprefix/qartest/dosdevices
$ ln -s /dev/mhuxd/cat1 com1
$ ln -s /dev/mhuxd/ptt1 com2
$ ln -s /dev/mhuxd/fsk1 com3

```

New wine versions, since version 2.8

Since the advent of wine version 2.8, above solution no longer works because wine overwrites any symbolic link. However, the CAT1, PTT1 and FSK1 ports can be mapped to respectively COM1, COM2 and COM3 by **adding new string entries to the HKEY_LOCAL_MACHINE\Software\Wine\Ports key** of the wine registry on the qartest virtual drive.

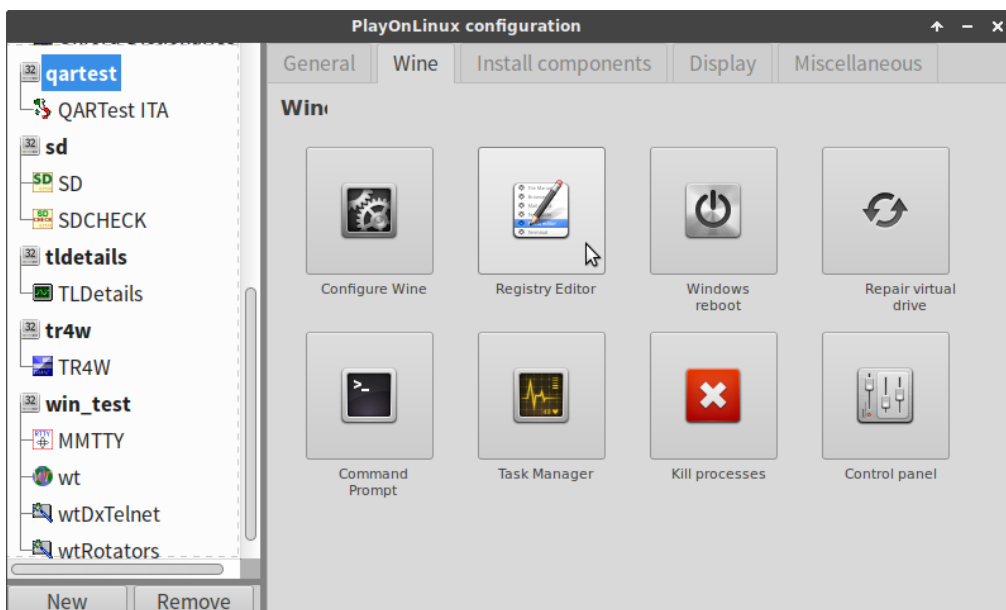


Figure 11: The registry editor can be opened through: PlayOnLinux → Configure → qartest virtual drive → Wine tab → Registry Editor.

Table 3: String values to be added

| name | type | data |
|------|--------|-----------------|
| COM1 | REG_SZ | /dev/mhuxd/cat1 |
| COM2 | REG_SZ | /dev/mhuxd/ptt1 |
| COM3 | REG_SZ | /dev/mhuxd/fsk1 |

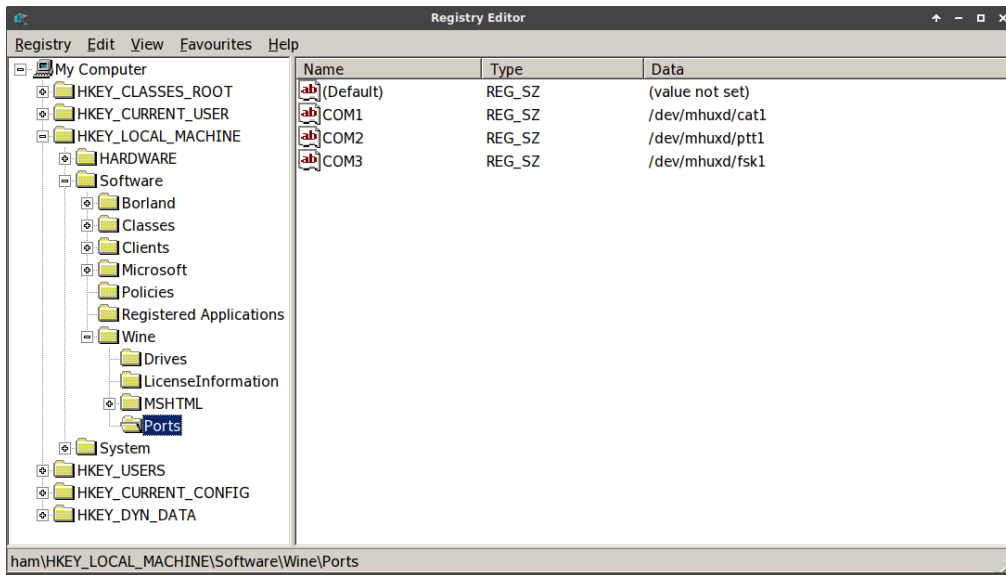


Figure 12: The HKEY_LOCAL_MACHINE\Software\Wine\Ports key with three new string values added to remap COM1, COM2 and COM3 to the mhuxd device ports.

Wine DPI setting

When you are used to GNU/Linux or for people with poor eyesight, the fonts on the Windows™ screens may appear particularly small. Luckily, the dots per inch (DPI) screen resolution can easily be set for each wine virtual drive individually.

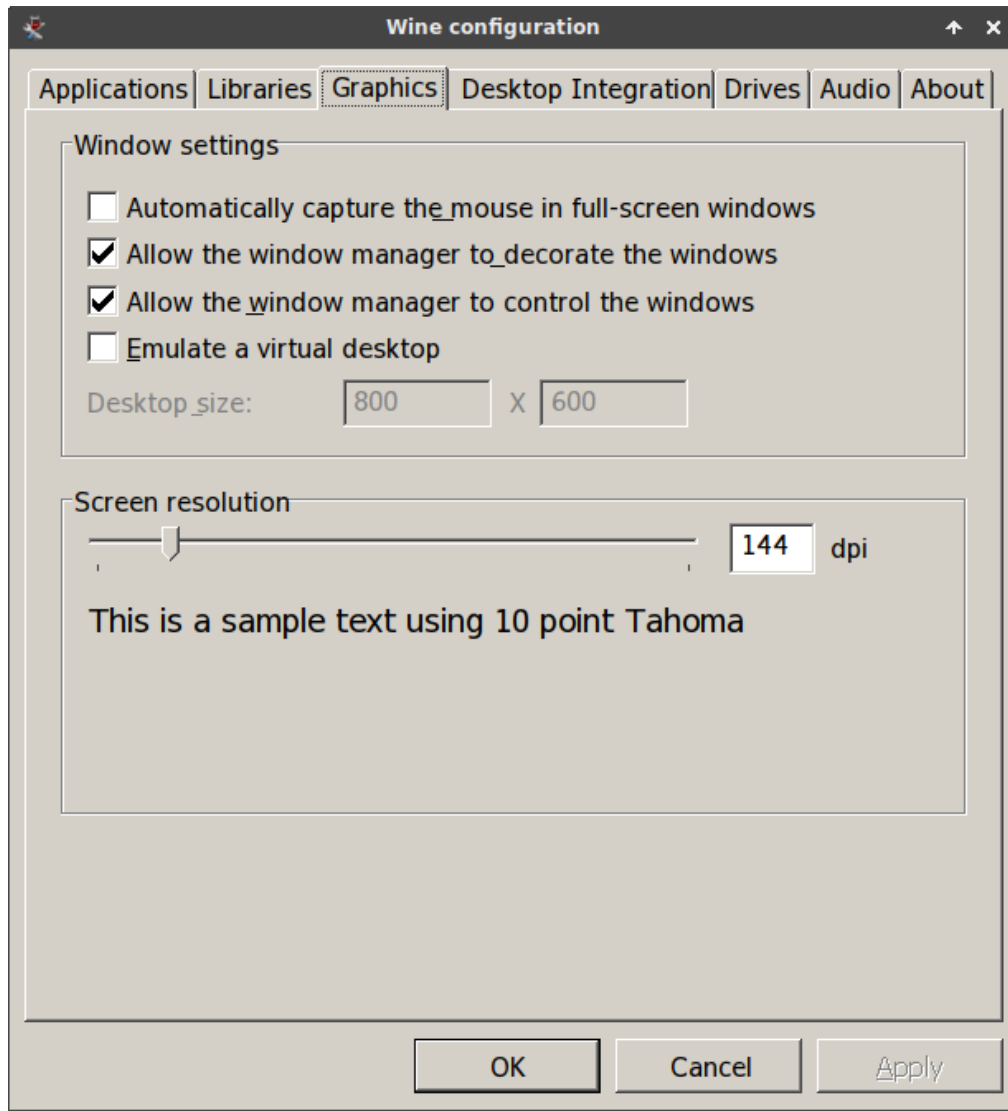


Figure 13: The dots per inch (DPI) screen resolution can easily be increased through: PlayOnLinux → Configure → qartest virtual drive → Wine tab → Configure Wine → Graphics.

Pulseaudio settings

The microHAM Digi Keyer contains its very own low-noise **PCM2902 stereo USB1.1 codec** chip. To use it, open pulseaudio volume control while MMTTY is running and **select the PCM2902 chip as recording source for MMTTY**. Newer microHAM interfaces use a differently named chip, as do modern transceivers with a built-in sound device and USB port.

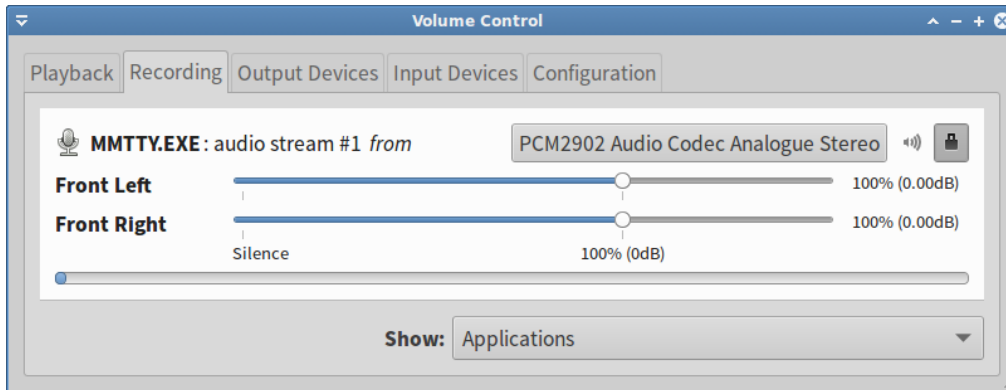


Figure 14: Binding MMTTY recording to the PCM2902 codec using pulseaudio volume control in **Xubuntu LTS**

MMTTY configuration

MMTTY is best configured and tested running standalone, i.e. independently of QARTest. All MMTTY configuration settings will remain in place under QARTest, except for the CAT settings and casual QSO message macros which apply only to MMTTY running standalone.

COM port for keying

I am a fervent advocate of **hard FSK keying**, which in a contest environment is **less prone to errors** than modulating audio AFSK. To do so, select COM3 in MMTTY's TX menu for FSK (and PTT) keying.

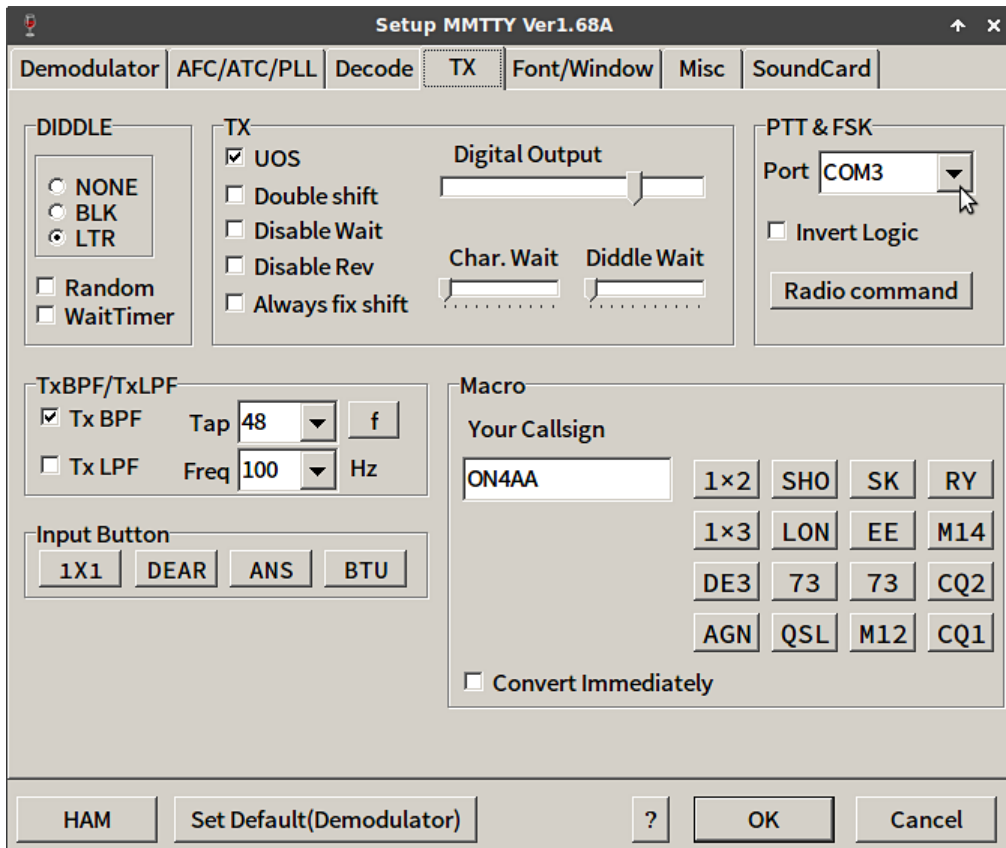


Figure 15: Select COM3 for FSK (and PTT) keying in the MMTTY TX menu

MMTTY is only able to use a single COM port for both FSK and PTT keying. Hence, PTT keying will be performed with a RTS (ready to send) signal on the very same COM3 port. For this to happen, place the appropriate check mark in MMTTY's Radio command menu, underneath the TX menu. This setup will only work when RTS-PTT was previously selected on the [microHAM port routing](#) screen.

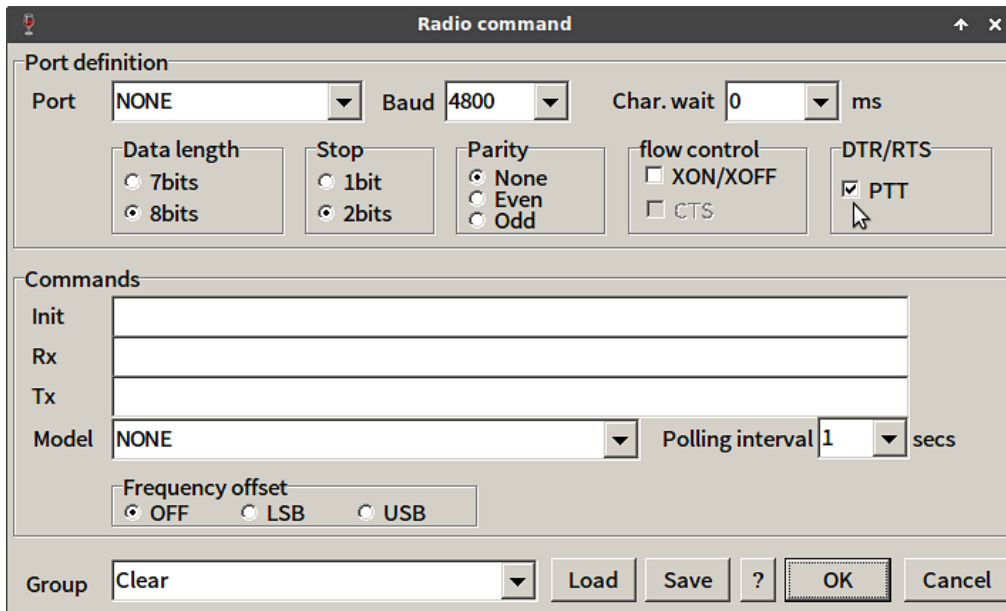


Figure 16: Select RTS for PTT keying in the MMTTY Radio command menu

MMTTY's all important main settings are next up for discussion.

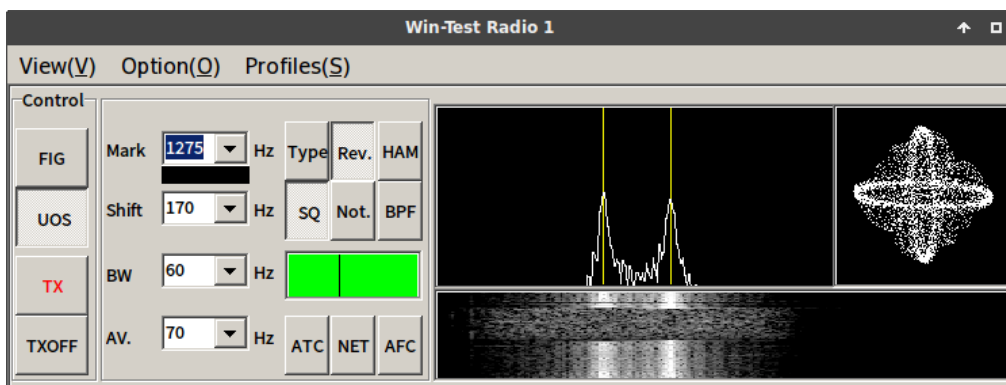


Figure 17: MMTTY on Linux!

Rev.

Because of **confusing FT-990 CAT messaging**, I need to operate my Yaesu FT-990 in RTTY-USB mode. Consequently, MMTTY's *Reverse* Rev. needs to be always on in my case. Test this setting by entering in an RTTY QSO before the contest! Very occasionally, when restarting under QARTest, MMTTY might be a little confused showing Rev. as on whilst not decoding in reverse. Should this happen, simply click Rev. twice, and all should be again in good order.

I have noticed that, **upon starting MMTTY V1.68A**, one needs to click the **Rev.** button **twice** in order to put MMTTY effectively in reverse decoding!

UOS on

It is recommended to *Unshift On Space* **UOS**, on both transmit and receive, whilst employing space delimiters (instead of using 599-001-001 hyphens). Doing so, increases the noise immunity for letters (which are more common), by always forcing the LTRS code page (i.e. unshifting) after a space.¹ By using unshifting spaces as delimiters between number sequences, the FIGS shift will also be repeated more often.

ATC

Automatic Threshold Control **ATC** adjusts the input level to the comparator in accordance with the strength of the input signal. If propagation conditions are good, it is better left off. It is only useful when there is an echo on the received signal.²



MMTTY's decoding threshold can also be set manually by moving the slider along the green bar. Set this according to your local noise level and propagation conditions.

NET

Activating **NET** makes your **transmit frequency** follow your receive frequency when using AFSK. This feature **should never be used during a contest**. It is of no effect when employing truly keyed FSK.

AFC

Automatic Frequency Control **AFC** lets MMTTY **track the frequency of the received signal** within the receive passband. **Leave **AFC** off when operating in S&P mode**; In RUN mode, having **AFC** switched on may be useful for working «lid operators». However, **remember to reset your MARK tone using the **HAM** button, when switching it back off!** Otherwise, you yourself become the «lid operator»!!!

Binary code 00000 is the null character, used for idling.

**Table 5: 5-bit ITA2 USTTY
code pages**

| binary | hex | LTRS | FIGS |
|--------|-----|------|------|
| 00011 | 03 | A | - |
| 11001 | 19 | B | ? |
| 01110 | 0E | C | : |
| 01001 | 09 | D | \$ |
| 00001 | 01 | E | 3 |
| 01101 | 0D | F | ! |
| 11010 | 1A | G | & |
| 10100 | 14 | H | # |
| 00110 | 06 | I | 8 |
| 01011 | 0B | J | BELL |
| 01111 | 0F | K | (|
| 10010 | 12 | L |) |
| 11100 | 1C | M | . |
| 01100 | 0C | N | , |
| 11000 | 18 | O | 9 |
| 10110 | 16 | P | 0 |
| 10111 | 17 | Q | 1 |
| 01010 | 0A | R | 4 |
| 00101 | 05 | S | ' |
| 10000 | 10 | T | 5 |
| 00111 | 07 | U | 7 |
| 11110 | 1E | V | ; |
| 10011 | 13 | W | 2 |
| 11101 | 1D | X | / |
| 10101 | 15 | Y | 6 |
| 10001 | 11 | Z | " |
| 01000 | 08 | CR | CR |
| 00010 | 02 | LF | LF |
| 00100 | 04 | SP | SP |
| 11111 | 1F | LTRS | LTRS |
| 11011 | 1B | FIGS | FIGS |

QARTest installation



QARTest TODO

Install Win-Test on the same `qartest` virtual drive where MMTTY was installed previously.

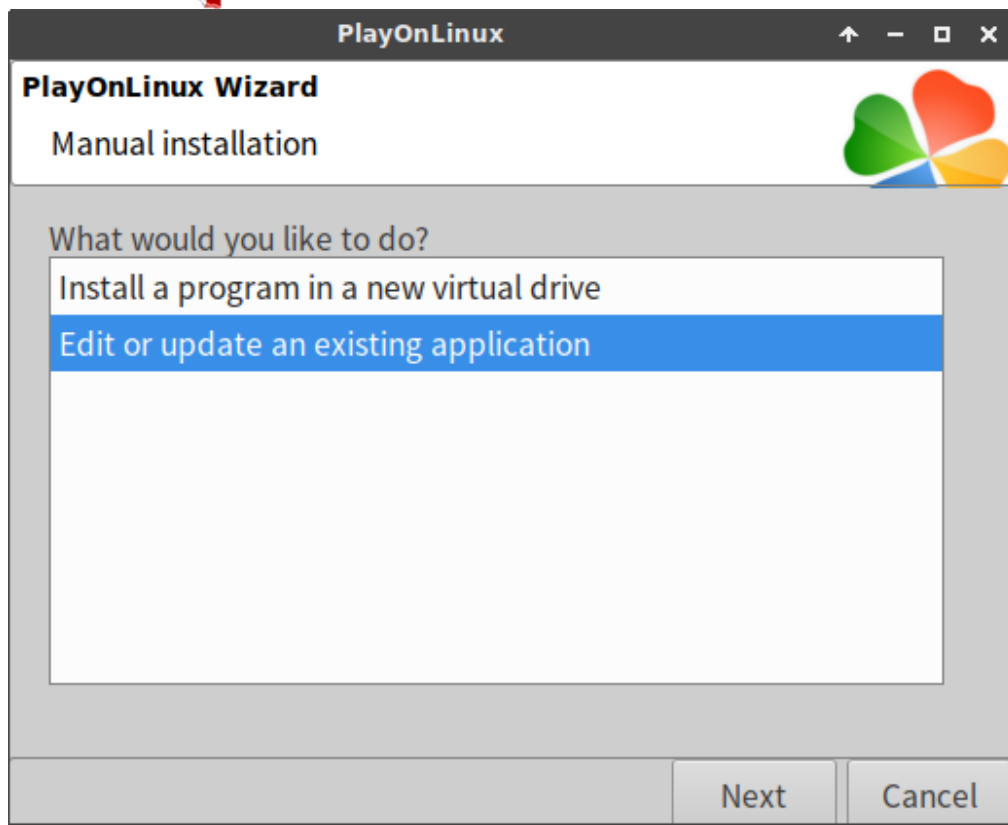


Figure 18: Choose «Edit or update an existing application» and click on Next.

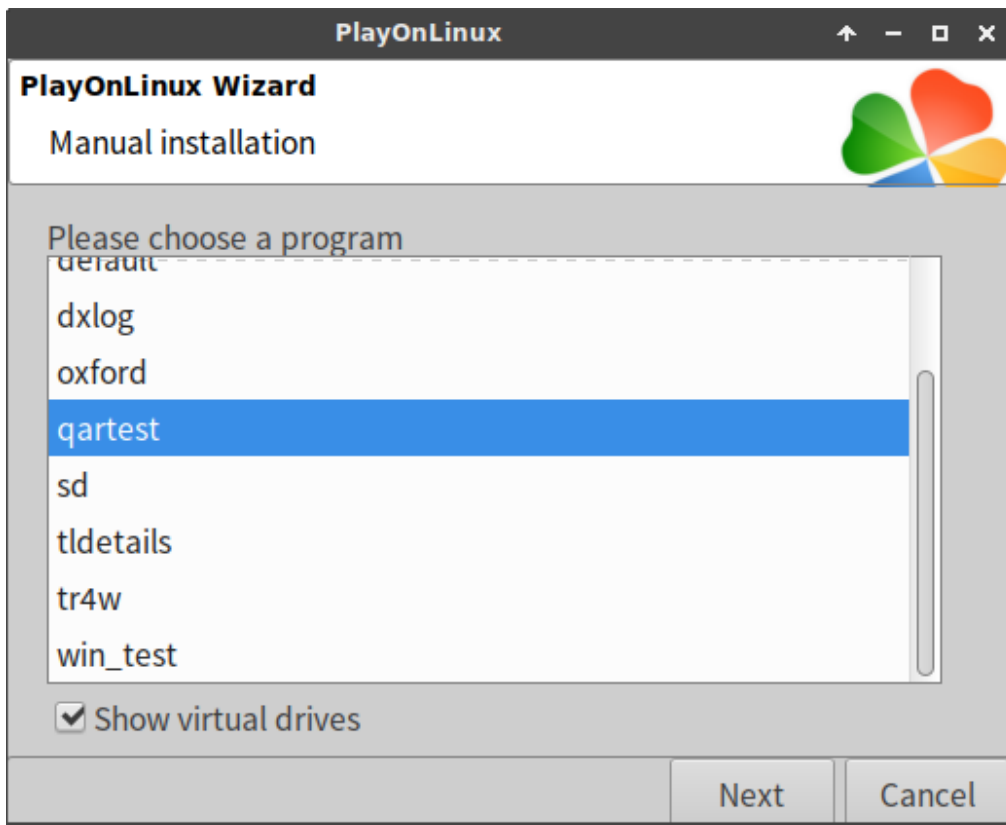


Figure 19: Select «Show virtual drives» and choose the qartest virtual drive. Continue by clicking on Next.

QARTest port configuration

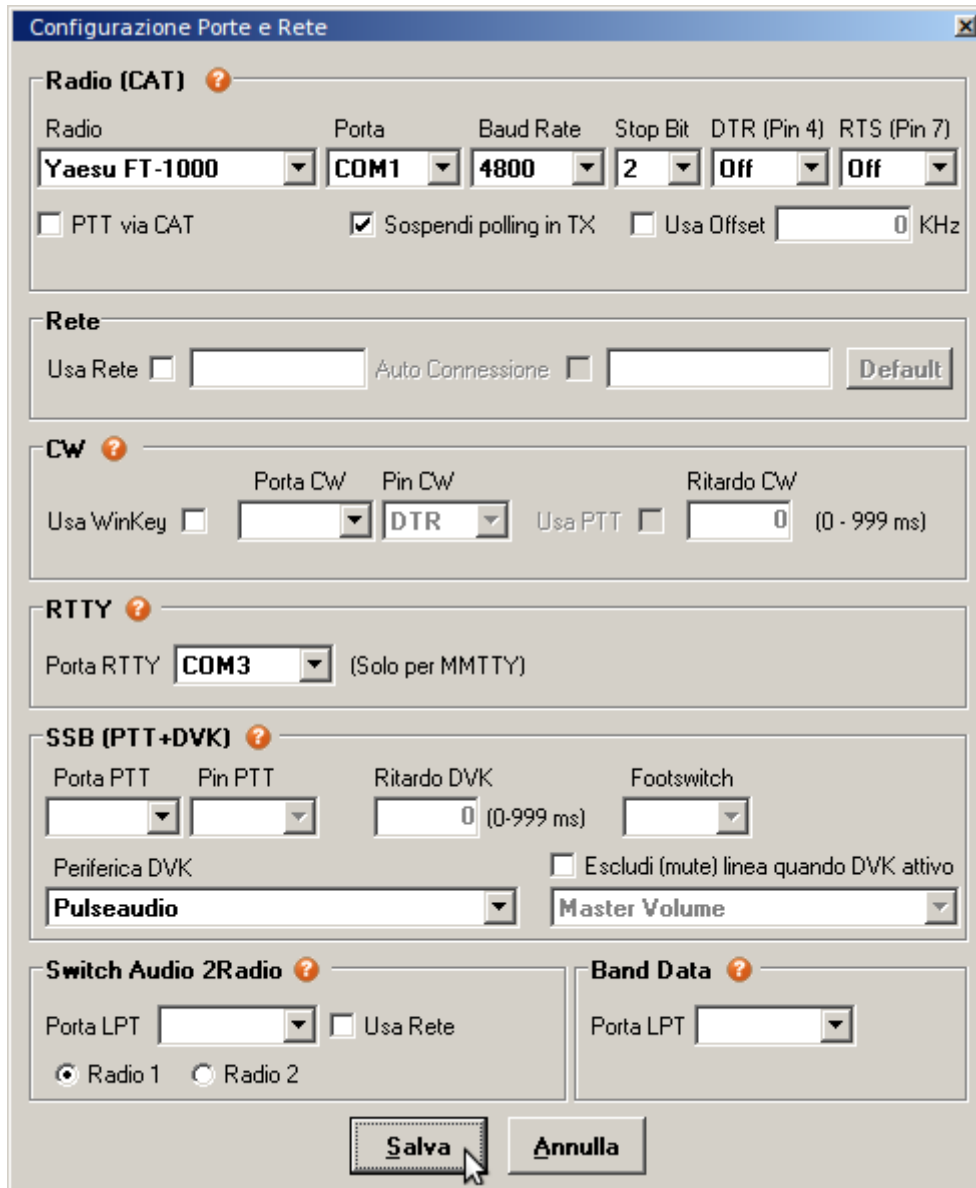


Figure 20: QARTest port configuration: The transceiver make and model are selected for CAT on COM1. Note that DTR and RTS are switched Off to avoid unwanted PTT switching. COM3 will be used by MMTTY for both FSK and PTT keying. Salva means Save.

RTTY contest messages

QARTest's default RTTY contest messages can be improved upon. [Donald A. Hill, AA5AU](#), of [RTTY contesting](#) fame offers a great breakdown of optimal [RTTY contest messages](#). Some of this great information has been condensed in the following set of rules³ and the tables below. On his fabulous website, Don also offers a number of [contest-specific examples](#).

1. Start all primary messages with a carriage return (CR/LF).
2. End all messages with a single space character.
3. Do not use outdated characters or character strings such as ~~DE~~, ~~K~~, ~~BK~~, ~~SK~~ or ~~UR~~ in any of your messages.
4. Send RST always as 599 and send it only once if required. If not required, do not send RST at all.
5. Always employ space delimiters (instead of 599-001-001 hyphens).¹
6. Always end your CQ message with CQ.
7. Always end your run confirmation message with either CQ or QRZ.

If you are completely new to RTTY, you will certainly find the [RTTY DXing presentation](#)¹ of Ed Muns, WØYK/P49X, very helpful in getting you started.

RTTY messages are defined under the Log → Messaggi menu.

During a contest, you can work stations in two ways:

- either *running*: i.e. calling CQ TEST,
- or *search & pounce*: i.e. working stations that are calling CQ.

QARTest does not feature any specific means to switch contest mode.

Table 6: QARTest macros

| key | use | message |
|-----|---------|------------------------------------|
| F1 | run | {CR}TEST {MY} {MY} CQ{SP} |
| F2 | run | {CR}{HIS} 599 {#} {#} 14{SP} |
| F3 | run | {CR}{HIS}{LOGQSO} TU {MY} CQ{SP} |
| F4 | repeat | {CR}AGN? AGN?{SP} |
| F5 | S&P | {CR}{MY} {MY} {MY}{SP} |
| F6 | S&P | {CR}599 {#} {#} 14 {MY}{SP} |
| F7 | S&P | {CR}QSL {MY}{SP} |
| F8 | repeat | {CR}{#} {#} {#} {#}{SP} |
| + | pile-up | {CR}{HIS}{LOGQSO} TU NOW {POP}{F2} |

References

1. Ed Muns, WØYK/P49X. RTTY DXing. Published online 2012.
<https://www.ncdxc.org/presentations/2012/W0YK-RTTY-DXing-06212012.pdf>

2. Jan Ditzian, KX2A. MMTTY help. Published 2003. https://hamwaves.com/linux.rtty/doc/MMTTY_Help.v166G.pdf
3. Donald A. Hill, AA5AU. RTTY contest messages. <https://www.rttycontesting.com/lagniappe/rtty-messages/>



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Other licensing available on request.

Unattended [CSS](https://www.w3.org/Style/CSS/Prince/) typesetting with . The Prince logo features the word "Prince" in a stylized purple font with "Print with CSS" written in smaller text above it.

This work is published at <https://hamwaves.com/qartest/en/>.

Last update: Monday, March 1, 2021.