
Minitel

Instructions on how to use a Minitel as a Linux terminal.

Felicitas Pojtinger

2021-12-11

Contents

1	Introduction	2
1.1	Contributing	2
1.2	License	2
2	Compatible Minitels	3
3	Minitel DIN-5 to USB/RS232/Serial Adapter	3
4	Minitel Shortcuts	6
5	Testing the Adapter	7
6	Setting up the Keymap	7
7	Setting up agetty	8
8	Setting up tmux	10

1 Introduction

1.1 Contributing

Found an error or have a suggestion? Please open an issue on GitHub (github.com/pojntfx/minitel):



Figure 1: QR-Code to the source code on GitHub

1.2 License

This document and included source code is Free Culture/Free Software.



Figure 2: Badge of the AGPL-3.0 license

Minitel (c) 2021 Felicitas Pojtinger

SPDX-License-Identifier: AGPL-3.0

2 Compatible Minitels

Your Minitel needs to have a Funz or Fnct key and the DIN-5 port at the back side. This includes the following Minitels:

- Minitel 1B
- Minitel 2
- Alcatel ADF 258

3 Minitel DIN-5 to USB/RS232/Serial Adapter

To build the adapter to connect the Minitel to a PC, you need the following (cheap) components:

- 220 k Ω resistor
- 22 k Ω resistor
- 10 k Ω resistor
- 2N2222 transistor
- Male DIN-5 plug
- PL2303HX USB to UART TTL converter

You will need to check the pinout of the DIN-5 plug/cable; in my case, the following mapping is present:

The diagram shows a 6x5 grid with handwritten text. The title 'DIN-5 (Minitel)' is in the top-left cell. The mapping is as follows:

Port	Color	Function
0	Red	(RX)
1	White	
2	Yellow	(GND)
3	Blue	(TX)
4	Green	
5		

Figure 3: DIN-5 port mapping

Connect like so:

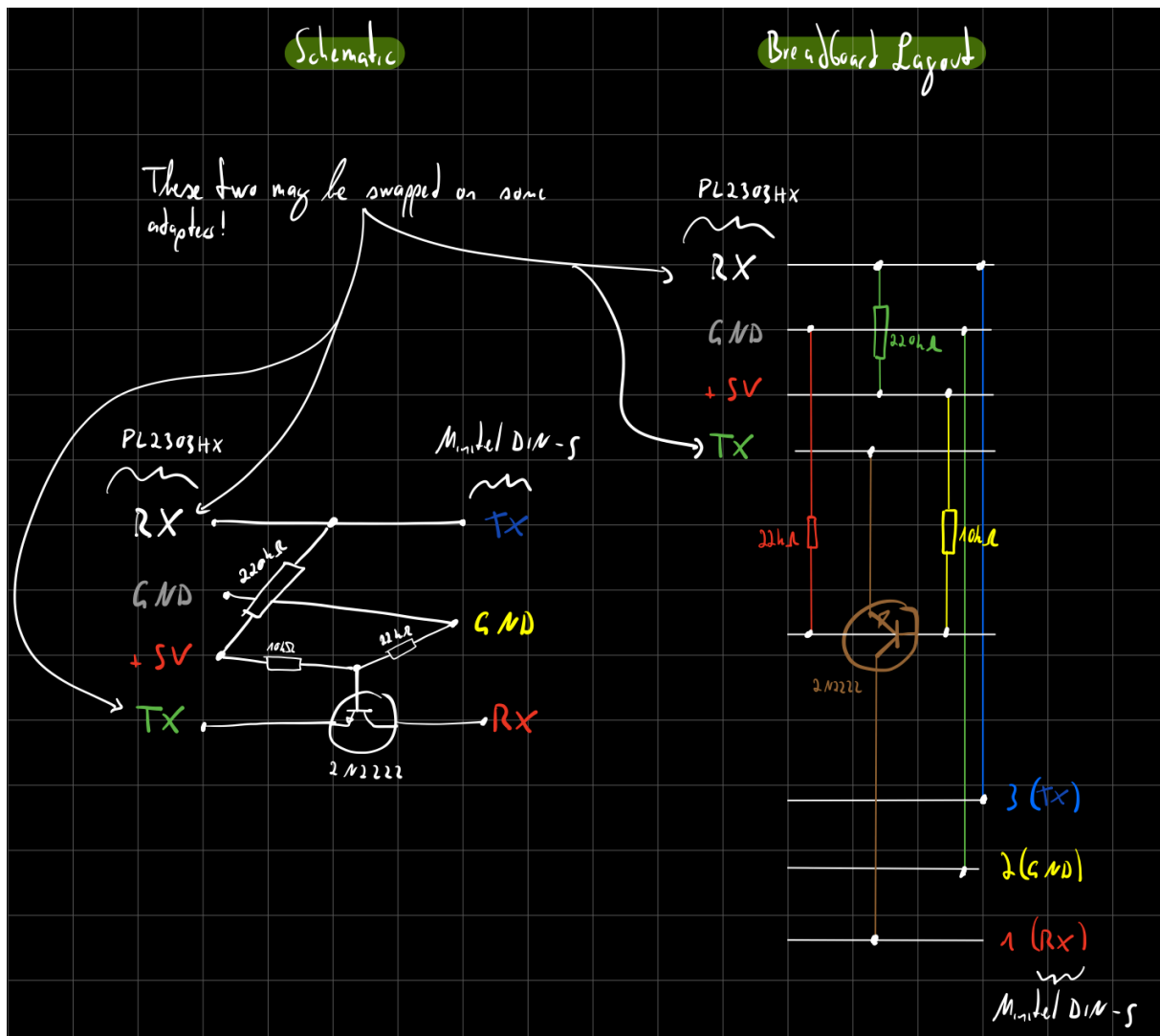


Figure 4: Schematic and breadboard layout

In my case, the breadboard prototype ended up looking like this:

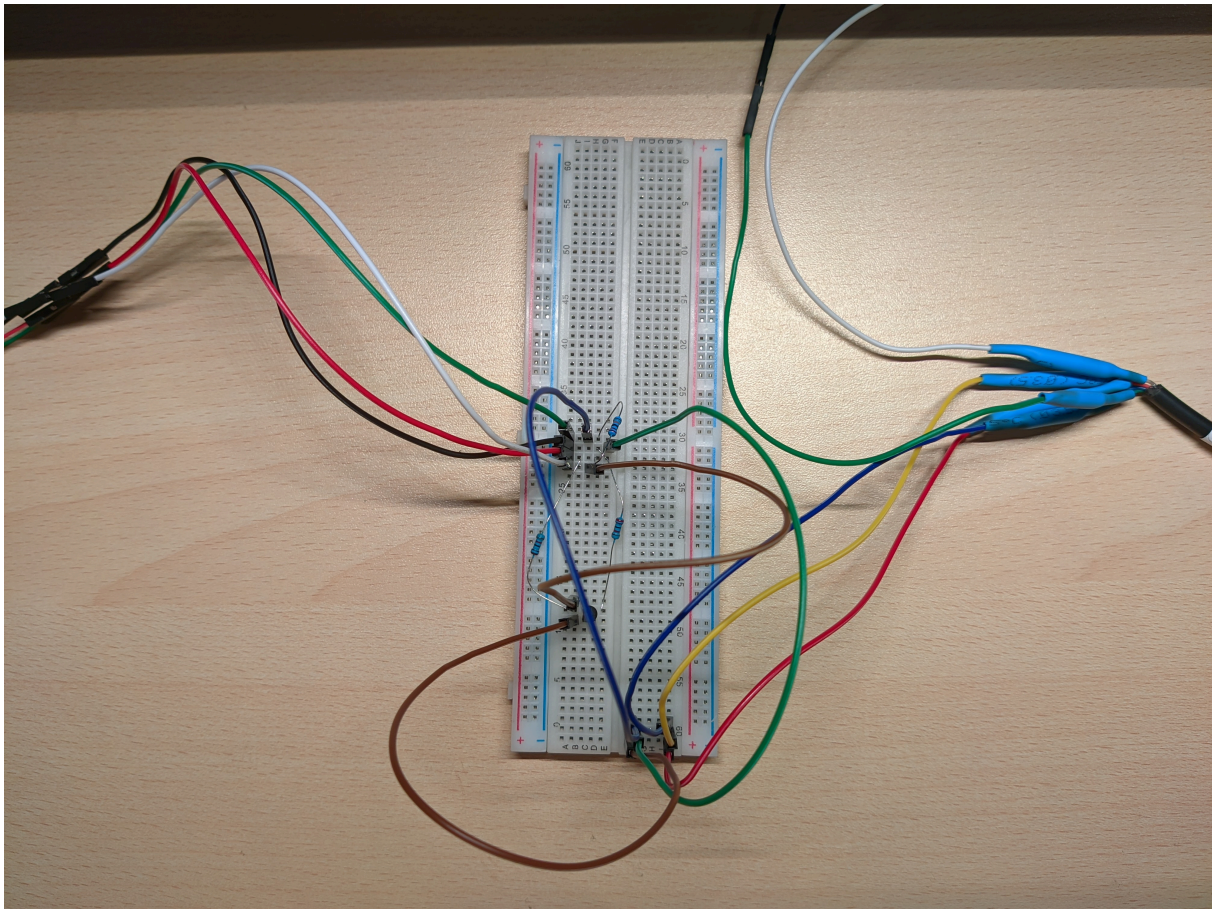


Figure 5: Breadboard prototype

I got this layout from [Pila's blog](#).

4 Minitel Shortcuts

Minitel terminals show the integrated phonebook by default; for them to be usable serial terminals, use the following shortcuts:

French Minitel 1B/2:

1. Fnc + T A: Enables ASCII mode
2. Fnc + T E: Disables local echo
3. Fnc + P 4: Sets baud rate to 4800 Baud (the maximum)

More info can be found on [Pila's blog](#).

Italian Minitel (Alcatel ADF 258):

1. Funz + Mem: Switches to terminal mode
2. Funz + M A: Enables ASCII mode
3. Funz + M E: Disables local echo
4. Funz + B 4: Sets baud rate to 4800 Baud (the maximum)

More info can be found on [Retronomicon](#).

5 Testing the Adapter

First, plug the PL2303HX into a USB port on your PC, then run the following:

```
1 $ sudo stty -F /dev/ttyUSB0 4800 istrip cs7 parenb -parodd brkint
   ignpar icrnl ixon ixany opost onlcr cread hupcl isig icanon echo
   echoe echok
```

This will initialize the terminal. Now, set up the Minitel using the shortcuts, and try to display something on it:

```
1 $ echo 'Hello, Minitel!_' | sudo tee /dev/ttyUSB0
```

If the `_` did not print correctly, run the following and try again:

```
1 $ echo 'ă' | sudo tee /dev/ttyUSB0 # Fixes # and _ etc.
```

You may use [Minicom](#) for further debugging: Start it using `sudo minicom -s -D /dev/ttyUSB0` and use 4800 Baud, 7 data bits, even parity bit, 1 stop bit and disable hardware flow control.

6 Setting up the Keymap

[Alexandre Montaron](#) has worked on improved support for the Minitel on Linux by providing a `terminfo` file; to get and use it, run the following:

```
1 $ curl -L -o /tmp/mntl.ti http://canal.chez.com/mntl.ti
2 $ tic /tmp/mntl.ti -o /etc/terminfo
```

You can also find a mirror [on GitHub Gist](#).

7 Setting up agetty

Using `getty`, or `agetty` in our case, it is possible to log into your PC using the Minitel. Exact setup instructions depend on your distribution, but for [Fedora 35](#) the following works; be sure to set up the Minitel using the shortcuts beforehand:

```
1 $ sudo tee /usr/lib/systemd/system/minitel-getty@.service <<'EOT'
2 # SPDX-License-Identifier: LGPL-2.1-or-later
3 #
4 # This file is part of systemd.
5 #
6 # systemd is free software; you can redistribute it and/or modify it
7 # under the terms of the GNU Lesser General Public License as
8 # published by
9 # the Free Software Foundation; either version 2.1 of the License, or
10 # (at your option) any later version.
11
12 [Unit]
13 Description=Serial Getty on %I
14 Documentation=man:agetty(8) man:systemd-getty-generator(8)
15 Documentation=http://0pointer.de/blog/projects/serial-console.html
16 BindsTo=dev-%i.device
17 After=dev-%i.device systemd-user-sessions.service plymouth-quit-wait.
18     service getty-pre.target
19 After=rc-local.service
20
21 # If additional gettys are spawned during boot then we should make
22 # sure that this is synchronized before getty.target, even though
23 # getty.target didn't actually pull it in.
24 Before=getty.target
25 IgnoreOnIsolate=yes
26
27 # IgnoreOnIsolate causes issues with sulogin, if someone isolates
28 # rescue.target or starts rescue.service from multi-user.target or
29 # graphical.target.
30 Conflicts=rescue.service
31 Before=rescue.service
32
33 [Service]
34 # The '-o' option value tells agetty to replace 'login' arguments with
35 # an
36 # option to preserve environment (-p), followed by '--' for safety, and
37 # then
38 # the entered username.
39 ExecStart=/usr/bin/sh -c "chcon -t tty_device_t /dev/%I && sudo stty -F
40     /dev/%I 4800 istrip cs7 parenb -parodd brkint ignpar icrnl ixon
41     ixany opost onlcr cread hupcl isig icanon echo echoe echok && /sbin/
42     agetty -o '-p -- \\u' -c %I 4800 m1b-x80 $TERM"
43 Type=idle
44 Restart=always
```

```
38 UtmpIdentifier=%I
39 TTYPath=/dev/%I
40 TTYReset=yes
41 TTYVHangup=yes
42 IgnoreSIGPIPE=no
43 SendSIGHUP=yes
44
45 [Install]
46 WantedBy=getty.target
47 EOT
48 $ sudo systemctl daemon-reload
49 $ sudo systemctl enable --now minitel-getty@ttyUSB0
```

You should now get a login prompt (Note the # where there should be a _):



Figure 6: Minitel showing a login prompt

You can show the login prompt again at a later time using the following:

```
1 $ sudo systemctl restart minitel-getty@ttyUSB0
```

After logging in, you should get a fully-featured shell:



Figure 7: Minitel showing the shell

We'll fix the #/_ characters next.

8 Setting up tmux

`tmux` makes using the Minitel much more enjoyable by providing support for panes and much more. You can use it by running:

```
1 $ tmux
```

It should look like this:



Figure 8: Minitel showing tmux

To get started, I recommend taking a look at the [Tmux cheatsheet](#).

To fix the #/_ characters and enable easy resetting when turning the Minitel on/off, run the following:

```
1 $ echo "bind-key r run-shell \"echo ä'; reset; echo 'Terminal has been
   reset, press q to close\"" >> ~/.tmux.conf
```

This will add a new command, Ctrl + b r, which will reset the terminal and fix the character set:



Figure 9: Minitel showing tmux after entering Ctrl + b r



Figure 10: Minitel showing `tmux` with the working charset

This now allows running complex software, like Vim, Links, Lynx and `cmus`:



Figure 11: Minitel showing Vim



Figure 12: Minitel showing DuckDuckGo on Links



Figure 13: Minitel showing Hacker News on L i n k s



Figure 14: Minitel showing this page on L i n k s



Figure 15: Minitel showing DuckDuckGo on Lynx



Figure 16: Minitel showing cmus