

RS232 VT100 Terminal V2.0 for Commodore Plus/4 by GMOLN

Introduction

This program is a **partially** VT100 compatible serial terminal, using the built in serial port of the Commodore Plus/4. Not all VT100 commands implemented yet. There are several commands which can't be implemented or can be implemented only partially, because of the limitations of the Commodore Plus/4. For example screen size, available foreground and background colour combinations in text mode, etc. However, the implemented command set provides good compatibility and makes the program usable in practice. **Further VT100 commands will be implemented in future releases.**

The program was written in **C** and compiled with **cc65**. The custom **6551A** driver with full-blown hardware flow control capabilities was written in assembly and complied with **ca65**. The whole program is compressed with Time Crunch V5.0.

Tests were done with serial console **agetty** from **util-linux 2.29.2** running on Raspberry Pi 2 / Raspbian Jessie.

Software

RS232 settings

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- CTS/RTS Hardware Flow Control

These settings are not changeable and **the serial console of the host computer must use the same settings**. Please note that, because of the speed limitation of the Commodore Plus/4, **CTS/RTS hardware flow control is mandatory**. Please refer to the Hardware section about the required user port / RS232 cable.

Some advice about the host computer settings

As it was mentioned above, the serial console of the host computer must use **CTS/RTS hardware flow control** and must run in **VT100** mode. In case of **agetty**, the following parameters can be used:

```
agetty -h 9600 %I vt100
```

Option **-h** will enable hardware flow control.

On Raspberry Pi and in case of Raspbian, the default serial console on the built in serial port needs to be disabled first with **raspi-config** utility. Then the **GPIO** pins need to be reconfigured with **rpi-serial** utility to provide **RTS/CTS** hardware flow control lines before starting the serial console with **agetty**.

If **stty** is available on the host, then command **stty rows 25 cols 40** can resize the terminal and correct the line breaks.

ASCII/PETSCII capabilities

The program automatically converts received ASCII characters to PETSCII and entered PETSCII characters to ASCII with some limitations. Currently only the alphanumeric characters are converted and some special characters:

- Received "~" shown as **PETSCII 168**
- Received "_" shown as **PETSCII 228**
- Received **ASCII 7** (bell) will flash the border of the screen with red colour
- Buttons "**Shift**" + "0" send **ASCII 94** as "^"
- Buttons "**Shift**" + "-" send **ASCII 95** as "_"
- Buttons "**Shift**" + "*" send **ASCII 126** as "~"
- **Clear** button types "**clear**" + **ASCII 13**
- **Home** button types "**cd ~**" + **ASCII 13**
- Buttons "**Shift**" or "**C=**" + "**RunStop**" exits from the program and return to Basic
- **UTF8** characters can be recognized by the program, but not displaying them. In case of received UTF8 character, the program will display "?"

Implemented ASCII Control characters

In general, the "**Control**" + "**A...Z**" keys send the appropriate **ASCII control character** according to the standard with the following exceptions:

- Instead of "**Control**" + "@" the "**C=**" + "@" sends **ASCII 0** as null
- "**RunStop**" also sends **ASCII 3** as end of text
- Instead of "**Control**" + "S" the "**C=**" + "S" sends **ASCII 19** as device control 3
- Instead of "**Control**" + "\" the "**Control**" + "£" sends **ASCII 28** as file separator
- Instead of "**Control**" + "^" the "**Control**" + "6" sends **ASCII 30** as record separator
- Instead of "**Control**" + "_" the "**Control**" + "7" sends **ASCII 31** as unit separator

Implemented VT100 commands

<ESC>[6n_____ Report Cursor Position, sends <ESC>[{ROW};;{COLUMN}]R

<ESC>[xD_____ Cursor Backward

<ESC>[xC_____ Cursor Forward

<ESC>[xB_____ Cursor Down

<ESC>[xA_____ Cursor Up

<ESC>[xJ_____ Clear Down, Up, Screen

<ESC>[xK_____ Erase Line, Start or End of Line

<ESC>[{ROW};;{COLUMN}]H_____ Set cursor position

<ESC>[H_____ Cursor Home

<ESC>[{attr1};...;{attrn}]m_____ Set Display Attributes

attr1: __0_____ Reset all attributes

1_____ Bright

2_____ Dim

5_____ Blink

7_____ Reverse

8_____ Hidden

30_____ Black

31_____ Red

32_____ Green

33_____ Yellow

34_____ Blue

35_____ Purple

36_____ Cyan

37_____ White

attr2: __30_____ Black

31_____ Red

32_____ Green

33_____ Yellow

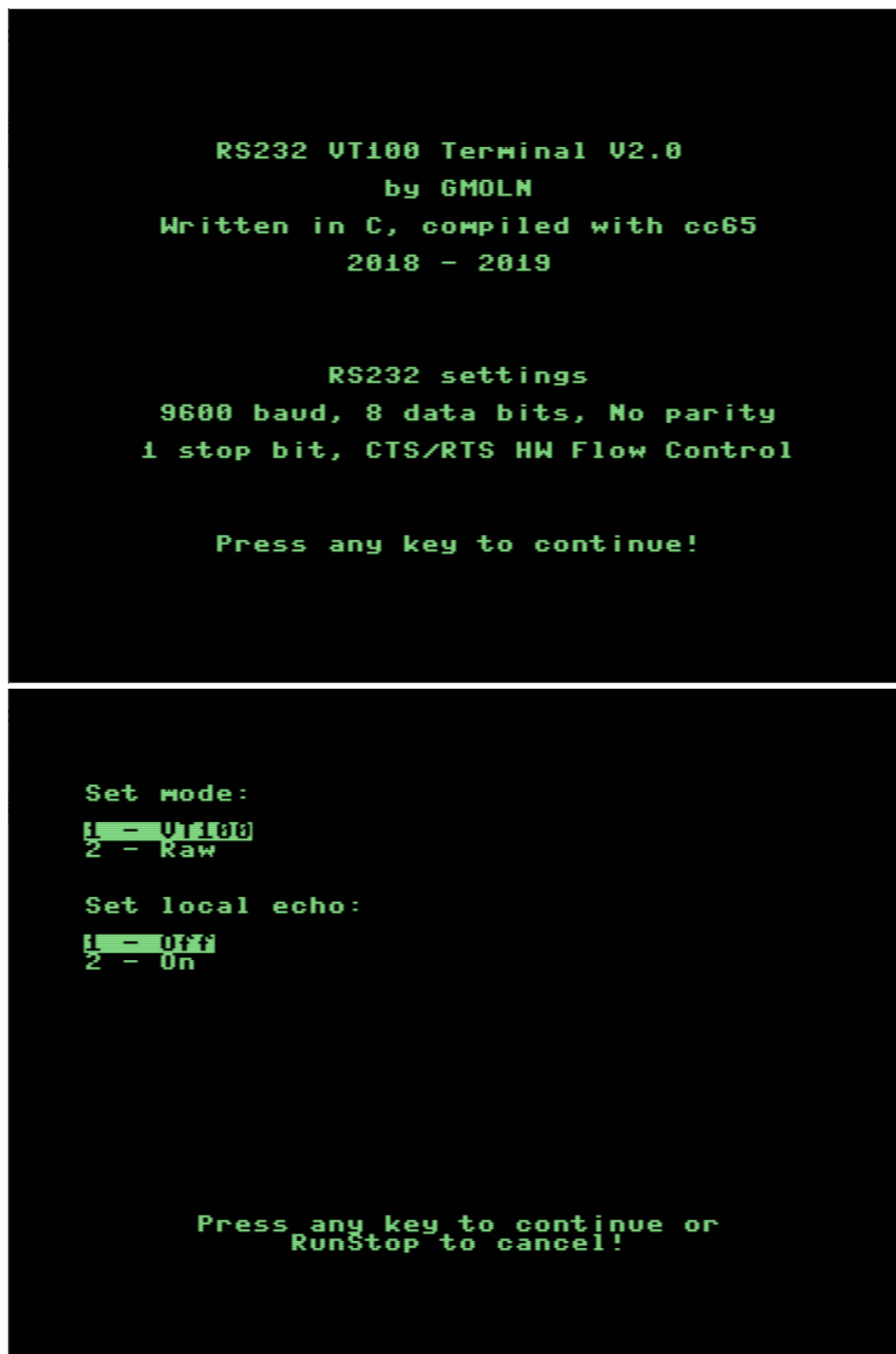
34_____ Blue

35_____ Purple

36_____ Cyan

37_____ White

Program settings



Set mode will select between **VT100** and **Raw** modes. In **Raw** mode, the program will not process the VT100 commands, however all other functionalities will still work like ASCII / PETSCII conversion, ASCII commands, etc.

Set local echo will switch local echo on or off. Usually, the serial console of the host provides echo, but there are cases when it is not provided by the host.

Known issues / limitations

It is strongly recommended to use command **stty rows 25 cols 40** after logon, if **stty** is available on the host. It can adjust the terminal screen size and correct the line breaks.

Colours can be displayed but with some limitations for **ls** command.

Command **watch** works perfectly.

Cursor up / down can display command history correctly even in case of long commands spread over multiple lines.

The **mc**, **top** and **htop** will not be displayed correctly.

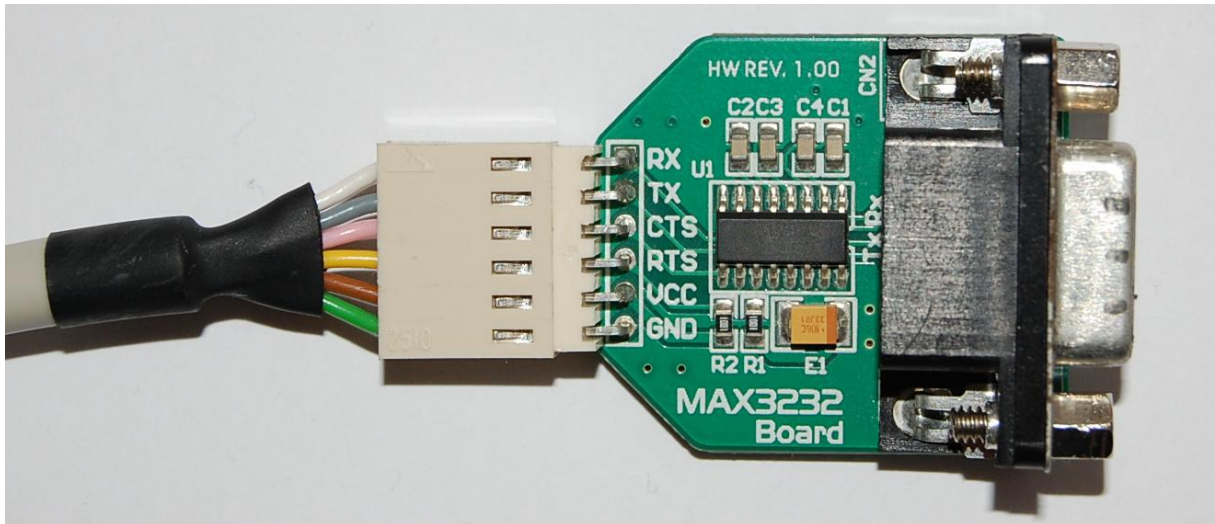
Function keys **F1...F7** & **Help** will not send anything, they are not implemented yet.

Hardware

The required user port / RS232 cable is as simple as possible.



It is based on a 24 pin Commodore User Port connector which can be purchased from various sources on eBay. Because the **6551A ACIA** in the Commodore Plus/4 is working with **TTL** levels, a further RS232 level shifter is required.



In this example, a **MAX3232** board is used from **Mikroelektronika**, but any other RS232 level shifter with **hardware flow control** capabilities can be used with **MAX (3)232** chips. The connections between the **User Port** connector and the **MAX3232** board should be the following:

User Port pin	MAX3232 board pin
2	VCC (+5V)
A	GND
C	RX
D	RTS
L	CTS
M	TX

User port pin out can be found below:

