

**RS232
mouse
adapter
for
Commodore
PLUS/4
8 bit
computer**

V1, 2016.05.19

**Szilveszter Bencsik, alias BSZGG
Zoltan Markus, alias MMS**

Opening words

The Commodore 264 series was originally planned to be a cheap ZX Spectrum competitor.

The evidence is the Commodore 116, with very small footprint and rubber keys. Looks very same as a ZX Spectrum. It was the first version, just before Mr. Bil Herd joined the project team.

The single chip design let it possible, integrating graphics, music, keyboard handling and IRQ control into into one small IC, the TED chip.

Due to the cost cautious design, several functions known in the C64 was left out. As C64 was \$299 at that time, and C116 planned to be a \$49 machine, nobody cared, as they wanted to compete with 16K and 48K ZX Spectrum. Both the graphics (320x200 with 121 colors), both the music (2 channels with volume control) was superior to ZX. So sprite handling, waveform control in music left out, User port is missing from lower positioned machines (C16, C116, 232). All these were missing from Spectrums too.

After Mr. Traimler fired from his own company, the new management was lost, what to do with the project. They did NOT want to compete with Sinclair, but wanted to make a business and starter machine. C116 was not able to fullfill any of those ideas (rubber keys are no-no for programming, 16K was not enough for serious work). But during development it turned out, how much flexible the design is, and several new machines grew out from the original concept (C16 in black C64 housing with 16KB RAM and robust keyboard (planned to be cheap VIC20 sucessor), the 232 with 32KB, Plus/4 with 64KB RAM +typewriter style keyboard, and 364 with extra numeric keyboard).

For our RS232 mouse project the **most important missing function was the POXT and POTY Paddle control inputs from the Joystick ports**. These data lines is utilized later by the 1351 mouse in C64, developed later (in 1986). Some great productivity utilities, like GEOS (1986), Amica Paint can be used much easier with 1351.

All these C64 hardwares and softwares were developed after 264project finished, and the machine was on the streets. Noone could forsee, that serious GUIs could be developed for 64K.

When GEOS +4 came out, due to my C64 experience I know, how much better it could be on Plus/4. **But because of the missing POT lines there was no way to connect an 1351 mouse to Plus/4, even if you manage to connect it physically with a joystick adapter**. My interest and forum communication grew towards the topic, without any real success.

Later on I found **bszggg**'s great homepage, listed an almost complete project for RS232 mouse: <http://bencsikszilveszter.hu/plus4/plus4/plus4.htm> .

I shared the page with others, but the response was almost zero. I was surprised a little, but finally understood: even if the board is simple to build (but there was no circuit diagram or exact HW description), there is no SW to utilize it, not to mention GEOS driver or other useful utilities.

Without proper documentation some people (like me) who has few experience in hardware stuff, is lost. It will be an interesting HW stuff, but noone will use it, or can use it properly.

So finally -somewhere in 2012- I decided to **make the adapter VERY easy to produce, and develop** together with bszggg and the Plus/4 members **a new mouse standard, with few hacked older SWs and some BASIC support** to get some ground for Plus/4 mouse.

So THIS IS IT.

This is version 1.00 (DRAFT version) of this document (2016.05.19)

Copyright

The original design and software are © 2010 bszggg, Szilveszter Bencsik.

The new „easy” hw core are © 2013 MMS, Zoltan Markus.

I am NOT a HW expert, so the it can be with FULL of mistakes :-)

This is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY

Possible methods to get working Mouse on Plus/4

We should not to forget the simplest method: to use the same HW C64 has + an 1351 mouse.

1) **NST (BSZ's) SID card** has an ATARI style JOY port, with the POTX, POTY on SID (on C64 the SID managed the JOY ports). The 1351 mouse together with the SID card can do the job, the same exact way as on a C64 with GEOS, Amiga Paint, etc.

Let's speak about the price. A SID Card is 60€ with a SID chip and a working 1351 mouse alone 30-50€ too. **WOW, 90€ for a mouse, that has no supporting SW yet.** (Plus/4 GEOS may worth a check with the 1351 driver.)

I wanted to create something anyone at any time can buy, or build without hesitation. SID+1351 is just too expensive and rare for this.

Going to the basics, you need a serial data protocol transfer to Plus/4. It can be done via the Expansion port with some special circuitry, or via the User port where the serial protocol (next to the 8 bit data lines) do exists by default. I chose the User port, because it is there on Plus/4 (sorry, C16, C116 owners!) and because mouse is only useful on a 64K machine.

This projects's supported mouse types:

	<u>Supported?</u>
Commodore 1351	No
USB mouse	No
PS/2 mouse	No
RS232 Serial mouse	YES!

USB mouse data transfer is just too high to able to manage with the built in ACIA (has some top speed of 19200 baud),

PS/2 is also too high, due to extended feaures, although have several, even slower modes.

(There is a very good artice about the PS/2 protocol by Levente Harsfalvi, who has a much higher level of knowledge on the hardware stuff, than me. It worth to be read, find with google!)

In short: handling of USB or PS/2 can be managed with some microcontrollers. But I wanted to use the Plus/4 raw power without expensive or complicated boards or ICs, programming or EPROM burning skills/tools (I as I also do not have them).

So it turned out, that an ancient (30 years old :-)) machine needs ancient protocol. The the RS232 came into the picture, and **RS232** already HAD some adapters on C64. RS232 mouses still exist on the market, and less rare (and cheaper) than the 1351 or even Amiga mouse.

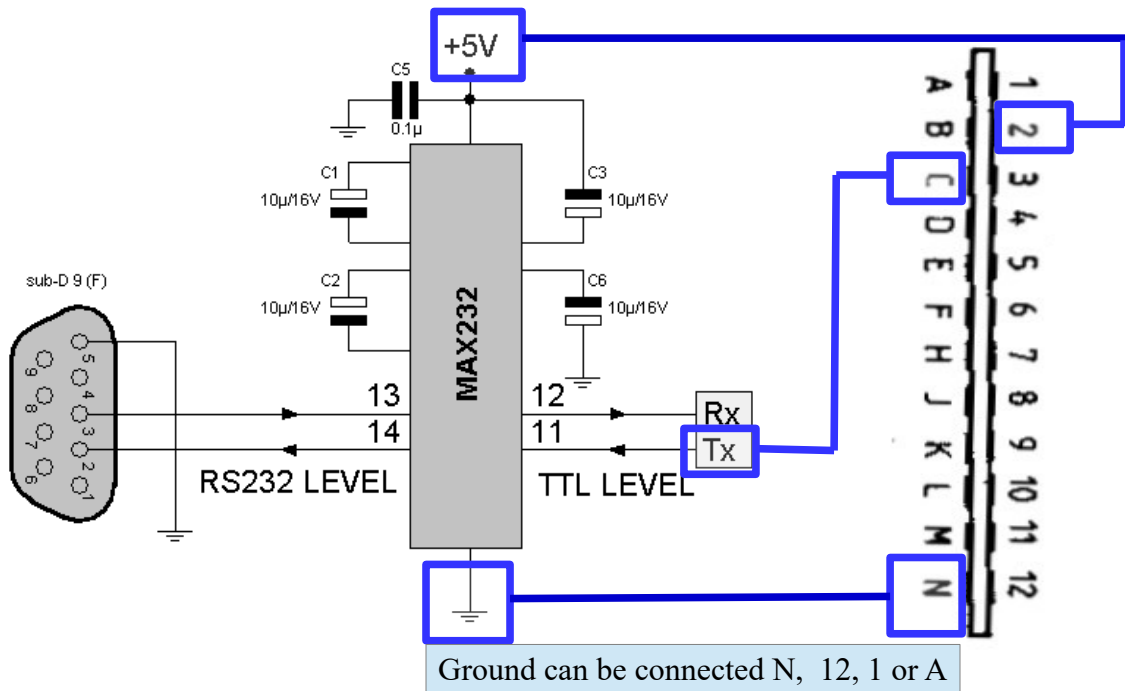
There is only one small problem: the RS232 has different voltage level than the TTL ones. And here comes into the picture the MAX232 IC, that makes this voltage level conversion.

About the RS232/TTL conversion circuitry:

You can buy a Max232 IC, and design + built you own board

MAX232 based TTL / RS232 converter board

Commodore Plus/4 User Port



D9 (DE-9) pinout

Pin For use Pin For use Pin For use

- 1 DCD (Data Carrier Detect)
- 2 RD (Receive Data)
- 3 TD (Transmit data)
- 4 DTR (Data Terminal Ready)
- 5 GROUND
- 6 DSR (Data Set Ready)
- 7 RTS (Request to Send)
- 8 CTS (Clear to Send)
- 9 RI (ringing Indicator)

Plus/4 User port pinout

- | | |
|-----------|----------|
| 1 GROUND | A GROUND |
| 2 +5V | B PO |
| 3 BRESET | C Rx/D |
| 4 P2 | D RTS |
| 5 P3 | E DTR |
| 6 P4 | F P7 |
| 7 P5 | H DCD |
| 8 Rx/C | J P6 |
| 9 ATN | K P1 |
| 10 9 V AC | L DSR |
| 11 9 V AC | M Tx/D |
| 12 GROUND | N GROUND |

The **MAX232 IC** is a dual driver/receiver from Texas Instruments that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. It converts TIA/EIA-232-F inputs (standard RS232) to 5-V TTL/CMOS level signal. Operates up to 120kb/s speed.

IC datasheet

<http://www.ti.com/lit/ds/symlink/max232.pdf>

Usually here comes the LAYOUT diagram, but we do not need it. 100% ready MAX232 circuits can be bought from eBay. Very cheap. No soldering required (yet).

So the easier approach with READY components:

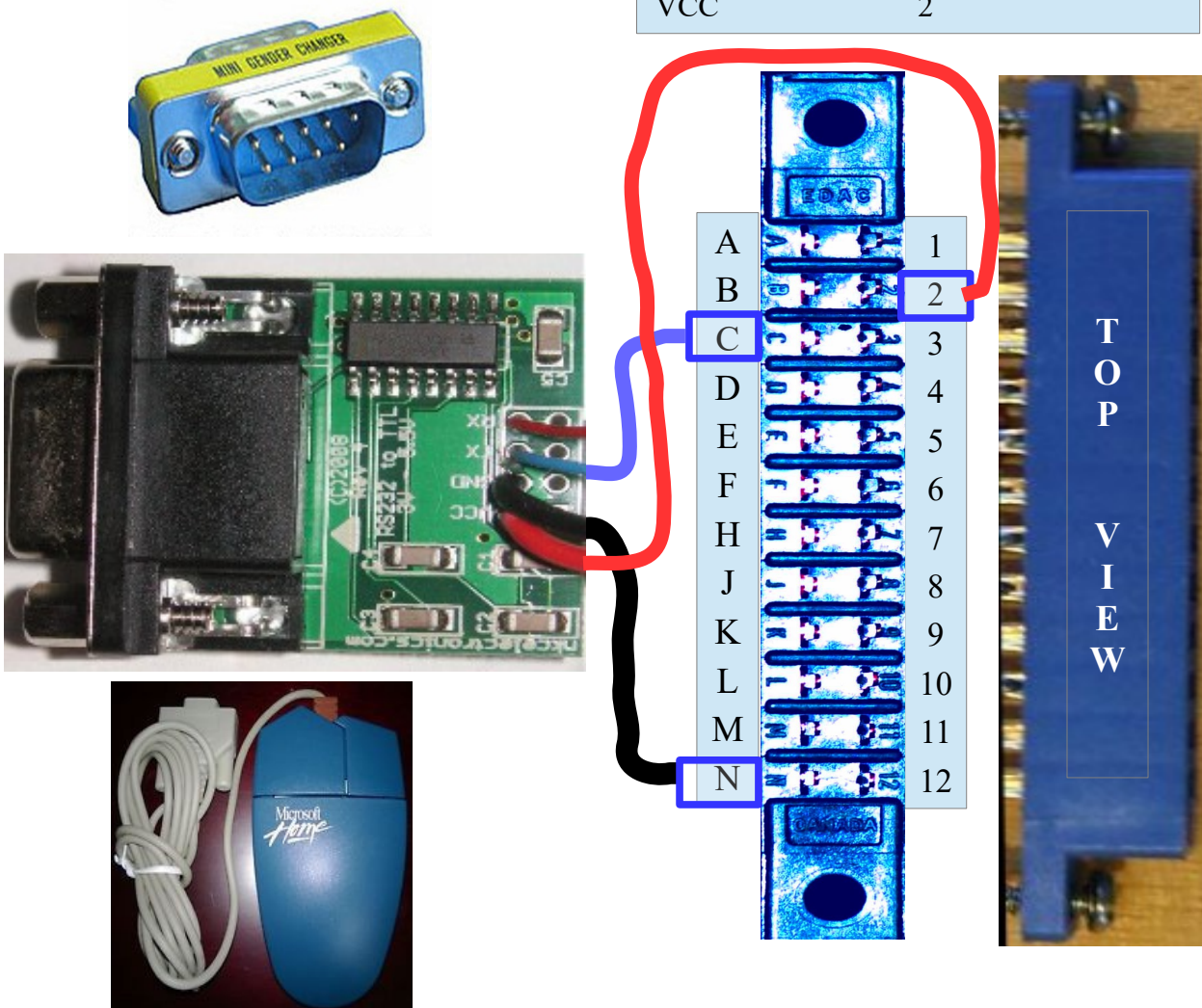
- Buy a Max232 board cheap from Internet. Price: 2 - 4 USD / piece
- Buy a User port (c64 User port is physically the same) Price: 2 - 8 USD / piece
- Buy an RS232 mouse Price: 2 - 15 USD
- (Serial/DB9/RS232 type required, and NOT a PS/2 mouse with an RS232 converter!)
- Buy a DB9 gender changer, to able to connect them Price: 2 - 3 USD
- Both Max232 and RS232 mouse have female connector

So, in ideal case from 8-10 USD you may have the HW of a working mouse on Plussy!
 In reality it will finish at 12-25USD (especially working RS232 mice in nice condition are rather rare nowadays, and User ports cannot be called cheap too).
 Locally in Hungary I may find untested RS232 mice for <2 USD
 This is still much cheaper, than the SID + 1351 solution.

In case you look for a mouse, lot of people around you may not know, that their old **DB9/serial mouse** may worth 10-15 USD, may get it free. Can pick up few on the next used HW store.

To able to connect a standard (female connector) RS232 Serial mouse to the Max232 circuit (female socket) please use a "Male to male" gender changer adapter

On panel	On User Port
Rx	Not connected
Tx	C
GND	1, 12, A or N
VCC	2



About the RS232 protocol:

After initialization of the RS232 mouse, it will send data only when any change happened in it's status: button pressed or moved. Basic RS232 mice send three bytes, the newer Wheel mouse types send 4 bytes (but we do not care).

The content of the three bytes (© <http://paulbourke.net/dataformats/serialmouse/>)

“The serial protocol for the mouse is 1200 baud, 7 bit, 1 stop bit, no parity.” Fits to our needs.

	D6	D5	D4	D3	D2	D1	D0
1st byte	1	LB	RB	Y7	Y6	X7	X6
2nd byte	0	X5	X4	X3	X2	X1	X0
3rd byte	0	Y5	Y4	Y3	Y2	Y1	Y0

Where

- LB is the state of the left button, 1 = pressed, 0 = released.
- RB is the state of the right button, 1 = pressed, 0 = released
- X0-7 is movement of the mouse in the X direction since the last packet. Positive movement is toward the right.
- Y0-7 is movement of the mouse in the Y direction since the last packet. Positive movement is back, toward the user.

Sample C code to decode three bytes from the mouse passed in "s"

The button and position (x,y) are returned.

```
/*
 s should consist of 3 bytes from the mouse
*/

void DecodeMouse(unsigned char *s,int *button,int *x,int *y)
{
 *button = 'n'; /* No button - should only happen on an error */
 if ((s[0] & 0x20) != 0)
  *button = 'l';
 else if ((s[0] & 0x10) != 0)
  *button = 'r';
 *x = (s[0] & 0x03) * 64 + (s[1] & 0x3F);
 if (*x > 127)
  *x = *x - 256;
 *y = (s[0] & 0x0C) * 16 + (s[2] & 0x3F);
 if (*y > 127)
  *y = *y - 256;
}
```