RS232 mouse adapter for Commodore PLUS/4 8 bit computer

V2, 2016.12.24

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 developped 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 got low response. 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.

THIS is JUST the hardware part.

This is version 2.00 (DRAFT version) of this document (2016.12.24)

Possible methods to get working Mouse on Plus/4

We should not to forget the simpliest 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 C64with GEOS, Amica 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.

2) Levente Hársfalvi's (TLC) PC mouse could NOT be converted to fit to Plus/4
It is a great project, but still expecting POTX, POTY lines on the joystick ports.
http://c64.rulez.org/capac/64eger/64egerdoku.htm

This projects's supported mouse types:

	Supported?
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 (TLC), 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.

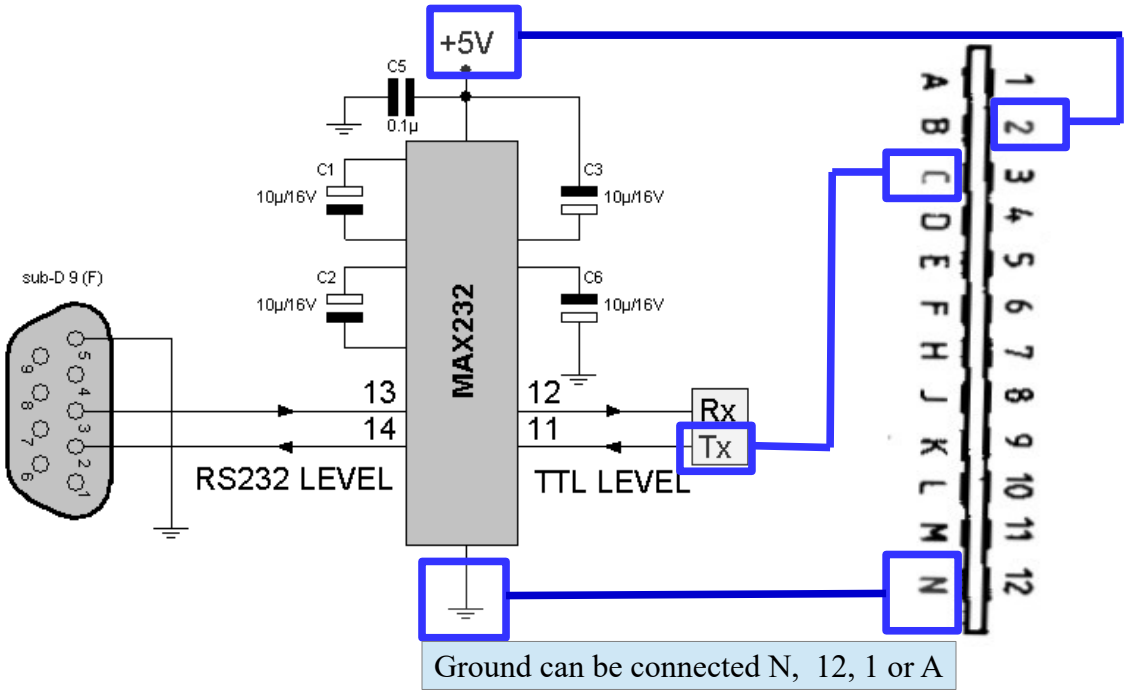
There is only one small problem: the RS232 has different voltage level than the TTL ones. And here comes into the picture the MAX232 IC, that makes this voltage level conversion.

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



Ground can be connected N, 12, 1 or A

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 RxD
4 P2	D RTS
5 P3	E DTR
6 P4	F P7
7 P5	H DCD
8 RxC	J P6
9 ATN	K P1
10 9 V AC	L DSR
11 9 V AC	M TxD
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 easiest approach with READY components:

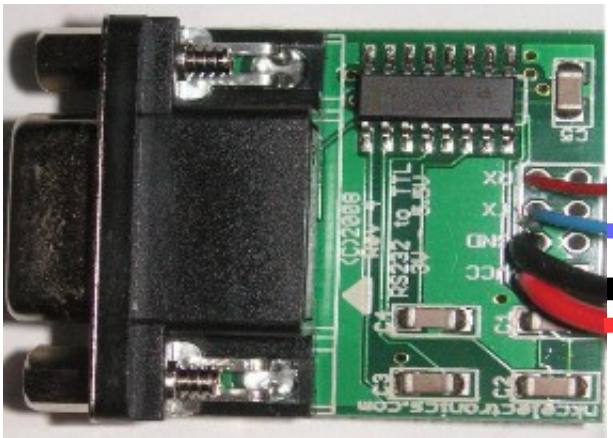
-Buy a Max232 board cheap from Internet. Price: 2 - 4 USD / piece
-Buy a User port (c64 User port is physically the same) Price: 2 - 8 USD / piece
-Buy an RS232 mouse Price: 2 - 15 USD
(Serial/DB9/RS232 type required, and NOT a PS/2 mouse with an RS232 converter!)
-Buy a DB9 gender changer, to able to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

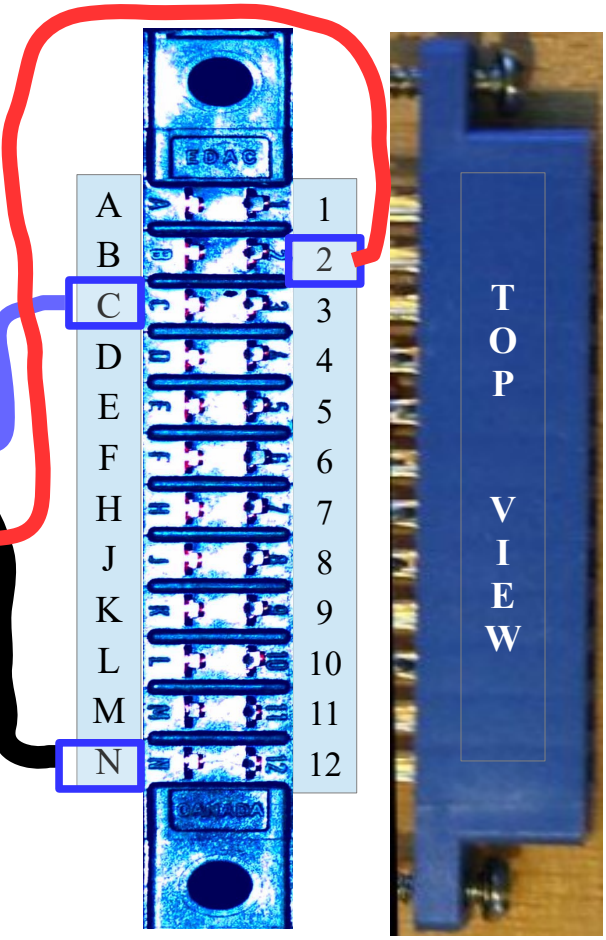
So, in ideal case from 8-10 USD you may have the HW of a working mouse on Plussy!
In reality it will finish at 12-25USD (especilly working RS232 mouses in nice condition are rather rare nowadays, and User ports cannot be called cheap too).
Locally in Hungary I may find untested RS232 mouses for <2 USD
This is still much cheaper, that 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 planel	On User Port
Rx	Not connected (M)
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 mouses 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;
}
```

Alternative methods:

The Max232 ready and cheap boards represent the simpliest form of RS232 mouse connection.
Need very little soldering, with DuPont cable easy to connect to the board.
No change in the RS232 mouse required.
But still no housing, and soldering are sensitive to external force, and does not look professional.
I could hardly found in my flat a good lokking plastic box would fit to my User Port connector.

But there are other options too.

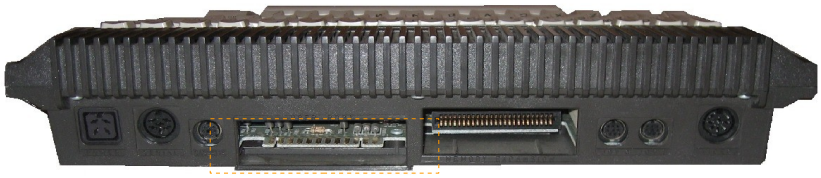
1) For VIC-20/C64 there are ready RS232 User Port cartrides made even by Commodore.
One example is the Commodore **VIC1011a.**

(on the last page you can find the schematic drawing of the VIC-1011A)

As stated in Plus/4 User Manual on Page209: "*RS-232 on the Plus/4 is standard RS-232 format, but the voltages are TTL levels (0 to 5V) rather than the normal RS-232 -12 to 12 volt range. The cable between the Commodore Plus/4 and the RS-232 device — should accommodate the necessary voltage conversions. (The Commodore RS-232 interface cartridge handles this properly.)"*

Despite a lot of discussion on the User Port differences (C64-Pluss/4), according to my latest investigation of differerent C64 User Port projects, my conclusion is that the C64 and Plus/4 User ports are MUCH more similar than we thought based on the different naming of the pins. At LEAST for serial (RS232 and Modem) protocol.
As the GND, +5V VDC, +9V VDC and RESET lines are on the same pins, you cannot do any harm by connecting a C64 User port cartride or Modem to a Plus/4.
Even more, if the 8 bit paralell data lines not used (these lines are really screwed up on +4), just purely the RS232 related read/write lines, it is **highly probable, that those will WORK with a Plus/4 without any modification.** I still need to buy one, not available in Huingary at all. :-(

Only two problem foreseen:
a) Probably you cannot install eg.SID card or even datasette to the Plussy, too wide.



b) Price: it is close to 50 USD+ Postage from US
So the landing price (75USD) will be close to the SID card + 1351 mouse version.

2) Recently I found a **similar project posted for a C64, just** after my project documentation was almost ready. :-(

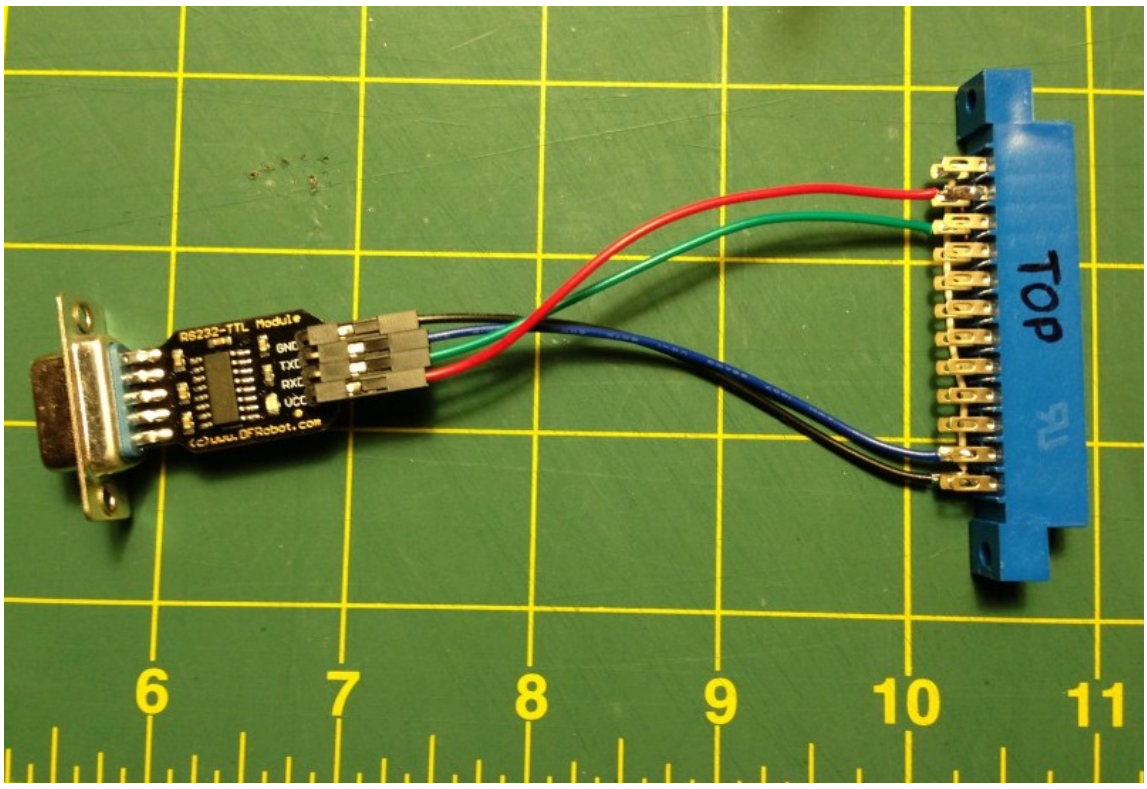
You can find it with google as "DIY RS-232 Interface for Commodore C64 for under $15" at http://biosrhythm.com/?p=1136

Please check below the ready adapter, look almost the same as ours. But with a big difference.
In that case the guy connected the TXD line of of the TTL adapter card to the User port M, but I think it could be wrong.
His suggestions:

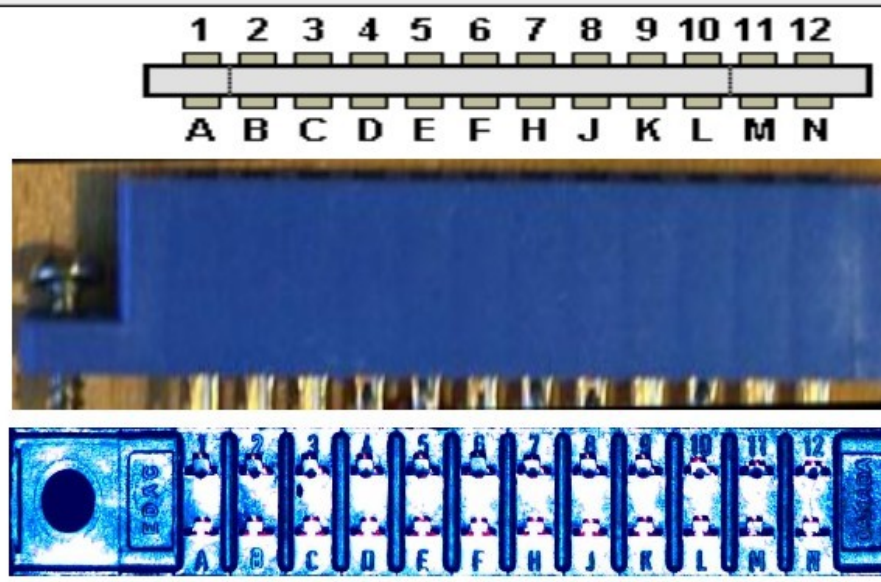
RS232-TTL Module	C64 User Port	
GND	A & N	
TXD	M	(PA2, CIA 2 bit2)
RXD	B & C	(Flag2 and Data 0)
VCC	2	

In this C64 case the TTL converter module the TXD is a sending (transmitting) signal (mouse 3 byte data), and the M pin of the C64 is PA2 ((PortA of CIA 2, Bit 2)), and it also a output line from C64.
As I see, an output line need to be connected to a receiving line, and not to an other output line.

So I think he swapped TXD and RXD. Or maybe I am wrong :-)
(if I am right, the same will work on Plus/4 too, just by swapping TXD and RXD to User port)



Commodore C64 and Plus/4 User port comparison
(based on Datasheets available on the net)



USER PORT PIN LAYOUT

Pin #		C64	Plus/4	
1	GND	GROUND	GND	GROUND
2	+5V	+5 VDC (100 mA max)	+5V	+5 VDC
3	/RESET	Reset, will force a cold start.	RESET	Reset
4	CNT1	Counter 1, from CIA #1	PB2	Data 2
5	SP1	Serial Port 1, from CIA #1	PB3	Data 3
6	CNT2	Counter 2, from CIA #2	PB4	Data 4
7	SP2	Serial Port 2, from CIA #2	PB5	Data 5
8	/PC2	Handshaking line, from CIA #2	RxC	Receive Clock
9	ATN	Serial attention in	ATN	Serial attention in
10	+9V AC	+9 VAC (+ phase) (100 mA max)	+9V AC	+9 VAC (+ phase)
11	+9V AC	+9 VAC (- phase) (100 mA max)	+9V AC	+9 VAC (- phase)
12	GND	GROUND	GND	GROUND
A	GND	GROUND	GND	GROUND
B	/FLAG2	FLAG2	PB0	Data 0
C	PB0	Data 0	RxD	Receive Data
D	PB1	Data 1	RTS	Request To Send
E	PB2	Data 2	DTR	Data Terminal Ready
F	PB3	Data 3	PB7	Data 7
H	PB4	Data 4	DCD	Data Carrier Detect
J	PB5	Data 5	PB6	Data 6
K	PB6	Data 6	PB1	Data 1
L	PB7	Data 7	DSR	Data Set Ready
M	PA2	PA2	TxD	Transmit data
N	GND	GROUND	GND	GROUND

VIC1011A Schematic diagram (not a MAX232)