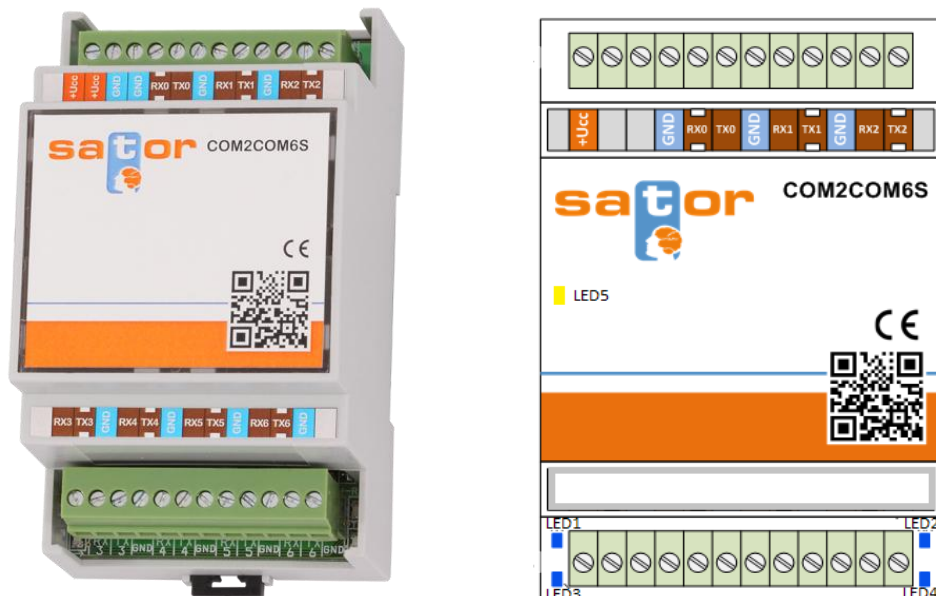


Per-RS232 translator (hw COM2COM6) [CZE]

Interface RS232-RS232 pro ovládání „nového“ zařízení „starými“ RS232 kódy – překládá kódy mezi starým a novým zařízením.



Interface RS232-RS232 pro ovládání „nového“ zařízení „starými“ RS232 kódy; využívá překlad pomocí uživatelsky zadávané tabulky RS232 kódů pro **původní zařízení (PORT0)** i RS232 kódů pro **nové zařízení (PORT1)**, včetně nastavení parametrů (baudrate/parity) obou portů. Uživatel může pro každý port zadat až 8 různých kontrolních kódů o délce max. 55 znaků, jak pro směr Rx (příjem control commands), tak i pro Tx (vyslání control commands příp. potvrzovacích odpovědí).

Interface lze používat i jako Baudrate Converter.

Pro nastavení interface a pro otestování komunikace s připojenými zařízeními slouží **systémový port (PORT2)**, které se provádí pomocí běžných textových příkazů a zpráv vyslaných z hyperterminálu nebo podobné aplikace s příkazovým řádkem. Příchozí, odchozí i systémová komunikace je indikována LED diodami. Data jsou uložena v EEPROM.

Typickým příkladem použití interface je např. výměna projektoru napojeného přes RS232 k (neznámému) řídicímu systému, za nový projektor s jinými řídicí kódy – bez nutnosti modifikovat původní program či nastavení řídicího systému.

Řídicí systém (HostPC) tedy i po obnově projektoru vysílá příkazy pro původní zařízení, interface tyto příkazy zachytí, a vyšle do nového projektoru příslušné jiné příkazy uložené v překladové tabulce (CtrlTable). Interface také umožňuje zachytit a vyslat zpět do řídicího systému potvrzovací odpovědi, pokud je původní řídicí systém vyžaduje, případně tyto potvrzovací odpovědi simulovat (režim Autoresponse). Totéž platí i směrem k novému zařízení, včetně simulace odpovědi které zařízení může vyžadovat, ale řídicí systém je v dané konfiguraci nevyšílá.

Režim Passthru (ver. 1.4>) určuje zda commands příchozí na port, jenž nejsou definovány v CtrlTable, budou zahozeny (pass =0, defaultní nastavení), nebo zda budou přeposlány nezměněny druhému zařízení (pass =1). Příchozí cmds které se shodují s uloženými cmds v CtrlTable jsou vždy přeloženy a poslány do druhého zařízení. Nastavení Passthru je společné pro oba směry komunikace. Tento režim může být vhodný např. pro inovaci projektoru stejného výrobce, který má stejné příkazy jako původní, jen např. s doplněnými novými vstupy. Pak stačí aktivovat režim Passthru (pass =1) a do CtrlTable uložit jen cmds pro nové vstupy (typicky HDMI1 místo VGA1). Ostatní příkazy (ON, OFF, MENU, ASPECT,...) do nového zařízení v tomto režimu projdou beze změny.

Z praktického hlediska, je-li prázdná CtrlTable (lze smazat příkazem init) a je aktivní režim Passthru (pass =1), interface se chová jako běžný Baudrate-Converter.

Specifikace/HW:

PCB: COM2COM6S V1 R003

Rozměry: 105x53x60mm, v provedení na lištu DIN, šířka 3MOD.

Napájení: 12-24VDC / typ. 50mA, svorky +Ucc, GND

RS232-PORT0 (ControlSystem / HostPC): Rx0,Tx0,GND (nepodporuje RTS/CTS); baudrate 9600/19200/38400/57600/115200, parity No/Even/Odd, 8-dataBit, 1-stopBit, (defaultně 9600,N,8,1), timeout 200ms

RS232-PORT1 (new controlled DEVICE): Rx1,Tx1,GND (nepodporuje RTS/CTS); baudrate 9600/19200/38400/57600/115200, parity No/Even/Odd, 8-dataBit, 1-stopBit, (defaultně 9600,N,8,1), timeout 200ms

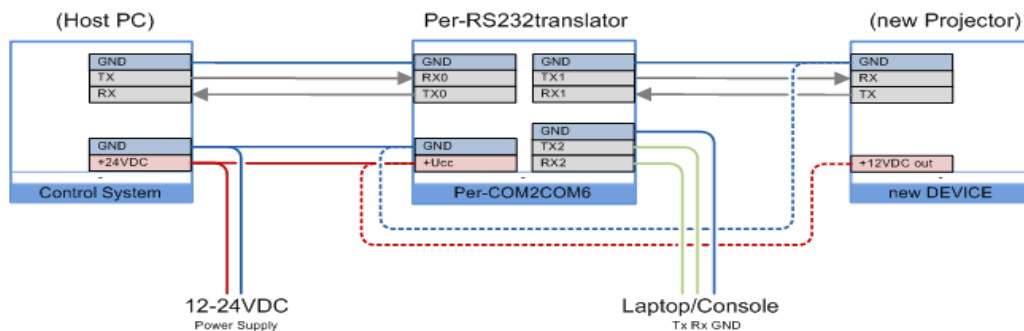
RS232-PORT2 (system settings PORT): Rx2,Tx2,GND (nepodporuje RTS/CTS); parametry vždy 9600,N,8,1, timeout 4s

LED1-4: aktivita Rx0 (LED1), aktivita Tx1 (LED2), aktivita Rx1 (LED3), aktivita Tx0 (LED4)

LED-indikace: krátké bliknutí: ctrl command se shoduje s daty v CtrlTable (příchozí string na Rx), příp. je vyslán přeložený command (odchozí string na Tx); delší bliknutí: ctrl string se neshoduje s daty v CtrlTable, zakončení příjmu na timeout LED5: svit - zadávání dat nebo příkazů na syst. portu (PORT2), zhasne po ukončení zadání nebo na timeout

Typické zapojení HW:

HostPC nebo ControlSystem je typicky zapojen na PORT0, ovládané zařízení (Device) na PORT1, viz obr.:



Signál z TX HostPC (původní ctrl commands) vede do Rx0; vstup RX HostPC přijímá signál (původní potvrzovací zprávy) z Tx0. Vstup RX DEVICE přijímá signál (nové ctrl commands) z Tx1, signál z TX DEVICE (nové potvrzovací zprávy) vede do Rx1. Servisní PC pro nastavení interface se připojuje na PORT2.

Funkcionalita a systémová nastavení:

Nastavení a komunikace se provádí textovými příkazy, zadávání je možné např. z PC aplikace Hyperterminál, HerculesSETUP apod. na PORT2, který je trvale nastaven s parametry 9600,N,8,1 (no RTS/CTS:-)

Po zapnutí interface nebo restartu se na 1sec rozsvítí všechny LED, a je-li připojen ServisPC, v okně hyperterminálu se zobrazí úvodní hláška nastavení portů, kontrola načtení dat z EEPROM do CtrlTable (...) a prompt (>):

```
port0: 19200,E,8,1
port1: 9600,N,8,1
....
>help
>
```

Ihned poté interface přijímá a vyhodnocuje příchozí příkazy (ctrl strings) na Rx0 resp. Rx1, a příp. vysílá přeložené odpovědi (reply strings) na Tx1 resp. Tx0.

Interface odděluje příchozí příkazy na Rx0 / Rx1 po uplynutí 200ms (timeout), tj. HostPC by neměl příkazy posílat rychleji za sebou. Pokud jsou ale příkazy platné (načtené v CtrlTable), pak jsou přeloženy a vyslány okamžitě, tedy provede se odeslání i vícekrát během 200ms.

Vyhodnocení ctrl string se provádí s načtenými příkazy v CtrlTable na shodu znaků zleva – pokud např. v CtrlTable je příkaz „HELLO“, pak nastane shoda (a překlad s případným vysláním určeného stringu do DEVICE) pokud HostPC vyšle příkazy „HELLO“ i např. „HELLO123“.

Zobrazením znaku '>' (prompt) je interface připraven na zadávání systémových příkazů ze ServisPC. Všechny příkazy na servisním portu se ukončují stiskem klávesy <Ent> (tj. dvojnázak \x0D\x0A, příp. stačí jen znak \x0D). Zadávání se ukončí i pokud je-li prodleva mezi znaky větší jak 4s.

Systémové příkazy:

Nastavení interface se provádí pomocí systémových příkazů zadaných z příkazového řádku ze ServisPC připojeného k portu PORT2 (komunikační parametry 9600,N,8,1); během zadávání svítí indikační LED5:

help<Ent>

zobrazí seznam příkazů interface.

ver<Ent>

zobrazí verzi firmware interface.

reboot<Ent>

provede reboot interface. Zobrazí aktuální nastavení pro PORT0 a PORT1, a dojde k načtení všech dat z EEPROM.

!init<Ent>

provede vymazání všech uživatelských dat v CtrlTable a EEPROM, nastaví Port0 a Port1 do default stavu a proběhne reboot.

!pass<Ent>

vypíše aktuální stav režimu Passthru.

!pass<Ent> ... vypíše aktuální stav režimu Passthru (0= neaktivní, 1= aktivní)

!pass <data><Ent>

načte a uloží hodnotu pro stav režimu Passthru (0 nebo 1).do EEPROM.

!pass 1<Ent> ...aktivuje režim Passthru.

!portN<Ent>

vypíše parametry komunikace pro daný port; N je číslo portu (0 nebo 1).

!port0<Ent> ... vypíše parametry komunikace pro PORT0

!portN <data><Ent>

načte a uloží parametry komunikace pro daný port; N je číslo portu (0 nebo 1) a zobrazí aktuální nastavení portu.

zadané parametry v poli <data> musí být jedna z podporovaných rychlostí (baudrate) a parita zadaná jako N, E nebo O; další parametry nejsou podporovány a jsou trvale nastaveny na 8-dataBit, 1-stopBit.

!port0 19200,E,8,1<Ent> ... nastaví parametry pro PORT0 na 19200,E,8,1

!port1 19200,O,7,2 <Ent> ... nastaví parametry pro PORT1 na 19200,O,8,1

!port0 38400,N <Ent> ... nastaví parametry pro PORT0 na 38400,N,8,1

!port0 9600 <Ent> ... změní parametr baudrate pro PORT0 na 9600, ostatní ponechá bez změny

!port0 0,N <Ent> ... změní parametr parity pro PORT0 na N, ostatní ponechá bez změny

?rxN<Ent>

vypíše seznam všech ctrl string uložených v CtrlTable pro směr Rx daného portu; N je číslo portu (0 nebo 1).

?rx0<Ent> ... seznam ctrl string pro Rx0 (PORT0)

?txN<Ent>

vypíše seznam všech ctrl string uložených v CtrlTable pro směr Tx daného portu; N je číslo portu (0 nebo 1).

?tx1<Ent> ... seznam ctrl string pro Tx1 (PORT1)

!rxN,M<Ent>

vypíše ctrl string načtený a uložený v CtrlTable; N je číslo portu (0 nebo 1), M je index načteného ctrl stringu (0 až 7).

!rx0,0<Ent> ... ctrl string pro Rx0 (PORT0), pozice 0 v CtrlTable

!rx1,7<Ent> ... ctrl string pro Rx1 (PORT1), pozice 7 v CtrlTable

!rxN,M <data><Ent>

načte a uloží do CtrlTable ctrl string zadaný v poli <data>; N je číslo portu (0 nebo 1), M je index načteného ctrl stringu (0 až 7); zadaný (user) string může mít až 55 znaků, musí být zadán jako soubor ASCII znaků bez mezer (whitechars) a může obsahovat jen tisknutelné znaky; pro netisknutelné znaky, mezery a kontrolní znaky se musí použít ASCII „upercase hex \xNN notatoin“.

!rx0,0 Hello123<Ent> ... ctrl string pro Rx0 (PORT0), pozice 0, bude obsahovat: Hello123

!rx0,1 Hello123\x0D\x0A<Ent> ... ctrl string pro Rx0 (PORT0), pozice 1, bude obsahovat: Hello123<0Dh><0Ah>

!rx0,2 \x01\x30\x31\x00\x0D<Ent> ... ctrl string pro Rx0 (PORT0), pozice 2, bude obsahovat: <01h>01<00h><0Dh>

!rx1,7 Hello\x20Hello\x20123<Ent> ... ctrl string pro Rx1 (PORT1), pozice 7, bude obsahovat: Hello Hello 123

!txN,M<Ent>

viz *!rxN,M<Ent>* ... platné pro Tx.

!txN,M <data><Ent>

viz *!rxN,M <data><Ent>* ... platné pro Tx.

!respN <data><Ent>

nastaví a vypíše stav režimu Autoreponse pro uložené ctrl stringy v CtrlTable pro daný port; N je číslo portu (0 nebo 1). hodnota v poli <data> je buď 1 nebo 0 (stav yes/no); pokud je režim Autoreponse např. pro PORT0 aktivní, pak při příjmu ctrl stringu načteného v CtrlTable např. jako Rx0,7 je vyslán zpět relevantní string (se stejným indexem) načtený zde jako Tx0,7 (případně prázdný string / žádná odpověď na Tx0 není-li žádný string na dané pozici načten).

!resp0 1<Ent> ... aktivace režimu Autoreponse pro PORT0

!resp1 0<Ent> ... deaktivace režimu Autoreponse pro PORT1

#rxN,M<Ent>

vypíše ctrl string načtený a uložený v CtrlTable pro daný port a vyšle relevantní (se stejným indexem) ctrl string na Tx pin opačného portu, pro možnost otestování reakce připojeného zařízení resp. pro kontrolu správnosti syntaxe ctrl stringu; pokud je pro daný port aktivní režim Autoreponse, vyšle se i uložený relevantní string na Tx pin stejného portu. Při zadání tohoto příkazu se tedy interface chová jako kdyby obdržel na daný Rx pin platný (načtený) ctrl string z připojeného zařízení. N je číslo portu (0 nebo 1), M je index načteného ctrl stringu (0 až 7).

#rx0,2<Ent> ... ctrl string pro Rx0 (PORT0), pozice 2 v CtrlTable je přeložen a jako tx1,2 vyslán na pin Tx1 (PORT1); je-li aktivní Autoreponse portu0, pak se vyšle i string tx0,2 na Tx0 (PORT0)

#txN,M<Ent>

vypíše ctrl string načtený a uložený v CtrlTable pro daný port a vyšle ho na Tx pin, pro možnost otestování reakce připojeného zařízení resp. pro kontrolu správnosti syntaxe ctrl stringu. N je číslo portu (0 nebo 1), M je index načteného ctrl stringu (0 až 7).

#tx1,2<Ent> ... ctrl string pro Tx1 (PORT1), pozice 2 v CtrlTable je vyslán na pin Tx1 (PORT1)

#txN,M <data><Ent>

umožňuje pro daný port zadat uživatelský (user) string a vyšle ho jako ctrl string na Tx pin daného portu, pro možnost otestování reakce připojeného zařízení resp. pro kontrolu správnosti syntaxe ctrl stringu. N je číslo portu (0 nebo 1), M je index a v tomto příkazu je irrelevantní. Zadaný user string v poli <data> může mít až 55 znaků, musí být zadán jako soubor ASCII znaků bez mezer (whitechars) a může obsahovat jen tisknutelné znaky; pro netisknutelné znaky, mezery a kontrolní znaky se musí použít ASCII „upercase hex \xNN notatoin“.

#tx1,0 Hello\x20Hello\x0D<Ent> ... ctrl string vyslaný na Tx1 (PORT1) bude obsahovat: Hello Hello<0Dh>

Pro jiné speciální požadavky než zde uvedené konfigurace interface, nebo přednastavení a otestování ovládacích kódů do CtrlTable lze po dohodě provést na zakázku při výrobě.

Náhled systémové komunikace viz obr. níže.

5/2019

www.perys.cz

```

Received/Sent data
port0: 19200,N,8,1
port1: 9600,N,8,1
....
>help
>

help
Look port0-rx/tx as OLD_cmd/reply and port1-tx/rx as NEW_cmd/reply
  use port number N=0..1; stored string index M=0..7
  use all strings without whitechars (space, comma, tab)
  use uppercase \xNN hex notation for nonprint/whitechars
help<Ent> ..print this help
ver<Ent> ..print fw version
reboot<Ent> ..perform device reboot
?rxN<Ent> ..list stored Rx ctrl strings of port(N)
?txN<Ent> ..list stored Tx ctrl strings of port(N)
#rxN,M<Ent> ..test port(N) stored ctrl string Rx[M]: #rx0,2
#txN,M<Ent> ..test port(N) stored ctrl string Tx[M]: #tx1,2
#rxN,M <data><Ent> ..test port(N) user string Tx[M]: #tx0,0 Hello\x0D
!rxN,M <data><Ent> ..set port(N) ctrl string Rx[M]: !rx0,2 OPER=001\x0D\x0A
!txN,M <data><Ent> ..set port(N) ctrl string Tx[M]: !tx1,2 PWR\x20is\x20off
!portN <data><Ent> ..set port(N) baudrate / parity: !port0 115200,N,8,1
!respN <data><Ent> ..set port(N) autoresp. (0=no/1=yes): !resp0 1
!pass <data><Ent> ..set passthru cmd mode (0=no/1=yes): !pass 1
!init<Ent> ..erase all stored data and set ports to default
>

ver
Per-RS232 translator v.1.4 [www.perys.cz]
>

?rx0

rx0,0: POF
rx0,1: PON
rx0,2: ADZZ:IIS:DVI
rx0,3: ADZZ:IIS:HD1
rx0,4: ADZZ:IIS:RG1
rx0,5: QPW
rx0,6: QSH
rx0,7:
>

>?tx1

tx1,0: PWR OFF
tx1,1: PWR ON
tx1,2: SOURCE A0
tx1,3: SOURCE 30
tx1,4: SOURCE 1F
tx1,5:

```

Modem lines

CD RI DSR CTS DTR RTS

Send

HEX

HEX

HEX

Serial

Name: COM8

Baud: 9600

Data size: 8

Parity: none

Handshake: OFF

Mode: Free

