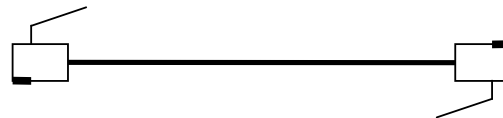


**Figyelem!!**

Az összekötő kábelnek teljesíteni kell, hogy az azonos pontok legyenek összekötve. (1. pont az 1-el, 2. pont a 2-vel stb. )!

<b>(CSR 1,2)</b>	<b>+9V/R</b>	A soros vonali illesztőegység tápellátását végzi.
<b>(CSR 3)</b>	<b>B</b>	Soros vonali jelvezeték
<b>(CSR 4)</b>	<b>A</b>	Soros vonali jelvezeték
<b>(CSR 5,6)</b>	<b>0V GND/R</b>	A soros vonal (valamint a +9V/R) referenciapontja.



A kommunikáció sebessége 38400 Baud, 8bit, paritásbit nincs, RS485 halfduplex.

A frekvenciaváltó szolgaként működik, csak akkor válaszol, ha ennek megfelelő parancsot kapott.

Távírányításra a terminál soros vonal szolgál, ilyenkor valamelyik digitális bemenetre az **“RS485táv”** funkciót kell programozni, majd aktivizálni kell a bemenetet (ilyenkor a terminál funkció megszűnik).

A soros vonalon érkező parancsra 2-3ms-on belül válaszol a frekvenciaváltó.

Egyszerre csak egy parancsot lehet küldeni, pl. START.

A soros vonali protokoll leírása.

Keretezés: STX (0x02),  
 $0x30+(U8)hossz \gg 4$ ,  
 $0x30+(U8)hossz \& 0x0f$ ,  
 $0x30+(U8)adat[0] \gg 4$ ,  
 $0x30+(U8)adat[0] \& 0x0f$ ,  
 ...,  
 ...,  
 $0x30+(U8)adat[hossz-1] \gg 4$ ,  
 $0x30+(U8)adat[hossz-1] \& 0x0f$ ,  
 $0x30+(U8)sum \gg 4$ ,  
 $0x30+(U8)sum \& 0x0f$ ,  
 ETX (0x03).

Ahol  $sum=(U8)$ (az STX karakterrel kezdődően az  $adat[hossz-1]$ -el végződően elküldött bájtok összege).

A hossz változó az üzenet típusától függően 4, 6, vagy 8 lehet.

Az adat[0] mindig a 'T' karakter.

Az adat[1] a lekérdezendő frekvenciaváltó címe a sorosvonalon: 1-16 lehet. A 0 cím mindenkinek szól, de ilyenkor senki sem válaszol.

Az adat[2] a lekérdezés szelektor. Ha 0, akkor nincs válasz, egyébként 1-25-ig a terminál funkció kijelzési képében kiválasztható mennyiségek közül választ, és válaszol a frekvenciaváltó.

Az adat[3] a bemenet szelektor.

Ha a legfelső bit (d7) 0, akkor digitális bemenet,

ahol d6=érték, d5-d0=0: funkció:

0-36: a sorkapocs digitális bemenetekre programozható parancsok. A hossz=4.

Ha a legfelső bit (d7) 1, akkor analóg bemenet, d6=0, d5-d0:

0: nincs funkció, (16bit), hossz=6,

1: vezérlés alapjel [0,1Hz], (16bit), hossz=6,

2: szabályozás alapjel [0,01%], (16bit), hossz=6,

3: pozíció alapjel 32bitesen, (32bit), hossz=8,

4: timeout idő [0,1s]. (16bit), hossz=6.

16 bites analóg érték esetén az adat[4] a felső 8 bit, az adat[5] az alsó 8 bit.

32 bites analóg érték (pozíció alapjel) esetén  $\text{adat}[4]=((U32)\text{pozíció})\gg 24,$   
 $\text{adat}[5]=((U32)\text{pozíció})\gg 16,$   
 $\text{adat}[4]=((U32)\text{pozíció})\gg 8,$   
 $\text{adat}[5]=((U32)\text{pozíció})\&0\text{xff}.$

A válasz üzenet keretezése ugyanaz, mint az előbb.

A hossz változó az üzenet tipusától függően 5, 7 lehet.

Az adat[0] mindig a 't' karakter.

Az adat[1] a lekérdezett frekvenciaváltó címe a sorosvonalon: 1-16 lehet.

Az adat[2] a lekérdezés szelektor.

16 bites analóg érték esetén az adat[3] a felső 8 bit, az adat[4] az alsó 8 bit.

32 bites analóg érték (pozíció alapjel) esetén  $\text{adat}[3]=((U32)\text{pozíció})\gg 24,$   
 $\text{adat}[4]=((U32)\text{pozíció})\gg 16,$   
 $\text{adat}[5]=((U32)\text{pozíció})\gg 8,$   
 $\text{adat}[6]=((U32)\text{pozíció})\&0\text{xff}.$

```
//-----  
// Kijelzési képből kiválasztható mennyiségek  
#define VK_KIJELZES_STATUS 0  
#define VK_KIJELZES_UZEMMOD 1  
#define VK_KIJELZES_F 2  
#define VK_KIJELZES_IMOTOR 3  
#define VK_KIJELZES_V_A_JEL 4  
#define VK_KIJELZES_SZ_A_JEL 5  
#define VK_KIJELZES_M_A_JEL 6  
#define VK_KIJELZES_ELL_JEL 7  
#define VK_KIJELZES_AN_BE1 8  
#define VK_KIJELZES_AN_BE2 9  
#define VK_KIJELZES_SZAML_1 10  
#define VK_KIJELZES_SZAML_2 11  
#define VK_KIJELZES_C_SZAML 12  
#define VK_KIJELZES_N 13  
#define VK_KIJELZES_N_IFA 14  
#define VK_KIJELZES_PFELVETT 15  
#define VK_KIJELZES_UHALOZAT 16  
#define VK_KIJELZES_UDC 17  
#define VK_KIJELZES_UMOTOR 18  
#define VK_KIJELZES_TERMELES 19  
#define VK_KIJELZES_TBORDA 20  
#define VK_KIJELZES_UZEMORA1 21  
#define VK_KIJELZES_UZEMORA2 22  
#define VK_KIJELZES_DIG_BE 23  
#define VK_KIJELZES_NYOMATEK 24  
#define VK_KIJELZES_RELEK 25  
#define VK_KIJELZES_URES_SOR 26
```

```
//-----  
// a sorkapocs digitális bemenetekre programozható parancsok  
#define VK_DIGITBE_KIVALASZT_N 0  
#define VK_DIGITBE_KIVALASZT_START_K 1  
#define VK_DIGITBE_KIVALASZT_START_NY 2  
#define VK_DIGITBE_KIVALASZT_IRANYV 3  
#define VK_DIGITBE_KIVALASZT_KUSZAS_N 4  
#define VK_DIGITBE_KIVALASZT_KUSZAS_I 5  
#define VK_DIGITBE_KIVALASZT_K_HIBA 6  
#define VK_DIGITBE_KIVALASZT_NYUGTAZ 7  
#define VK_DIGITBE_KIVALASZT_STOP 8  
#define VK_DIGITBE_KIVALASZT_KIPOERGES 9  
#define VK_DIGITBE_KIVALASZT_DC_FEK 10  
#define VK_DIGITBE_KIVALASZT_F_TARTAS 11  
#define VK_DIGITBE_KIVALASZT_M_P_FEL 12  
#define VK_DIGITBE_KIVALASZT_M_P_LE 13  
#define VK_DIGITBE_KIVALASZT_VEZ_SZAB 14  
#define VK_DIGITBE_KIVALASZT_PRG_TILT 15  
#define VK_DIGITBE_KIVALASZT_PRG_STOP 16  
#define VK_DIGITBE_KIVALASZT_PRG_LEP 17  
#define VK_DIGITBE_KIVALASZT_AN_CSERE 18  
#define VK_DIGITBE_KIVALASZT_MODOSITO 19  
#define VK_DIGITBE_KIVALASZT_SZAM1_FEL 20  
#define VK_DIGITBE_KIVALASZT_SZAM1_LE 21  
#define VK_DIGITBE_KIVALASZT_SZAM1_Z 22  
#define VK_DIGITBE_KIVALASZT_SZAM2_FEL 23  
#define VK_DIGITBE_KIVALASZT_SZAM2_LE 24  
#define VK_DIGITBE_KIVALASZT_SZAM2_Z 25  
#define VK_DIGITBE_KIVALASZT_IFAPOZ_0 26  
#define VK_DIGITBE_KIVALASZT_PRG_BE1 27  
#define VK_DIGITBE_KIVALASZT_PRG_BE2 28  
#define VK_DIGITBE_KIVALASZT_PRG_BE3 29  
#define VK_DIGITBE_KIVALASZT_PRG_BE4 30  
#define VK_DIGITBE_KIVALASZT_PRG_BE5 31  
#define VK_DIGITBE_KIVALASZT_PRG_BE6 32  
#define VK_DIGITBE_KIVALASZT_PRG_BE7 33  
#define VK_DIGITBE_KIVALASZT_PRG_BE8 34  
//-----  
// Analog alapjelek  
#define FrekiAlapjelCmd 1  
#define SzabAlapjelCmd 2  
#define Pos32AlapjelCmd 3  
#define TimeoutAlapjelCmd 4  
//-----
```