

1. TCP/IP Empfangsbuffer in der SPS

Zeichen in separaten Empfangsbuffer

Bei TCP/IP können zusammenhängende ASCII Sequenzen in verschiedene Telegramm-Pakete aufgeteilt werden. Falls nun direkt auf das Prompt „>“ Zeichen gewartet wird und dann das Telegramm gelöscht wird, können Zeichen von der nächsten ASCII Sequenz mitgelöscht werden und das Protokoll gerät durcheinander. Deshalb müssen die empfangenen Zeichen in einen separaten Empfangsbuffer abgelegt werden. Wird in diesem Empfangsbuffer das Prompt „>“ erkannt so kann die Information gesichert und der Buffer bis und mit dem Prompt Zeichen gelöscht werden. Allfällige Zeichen nach dem Prompt dürfen nicht gelöscht werden und bleiben im Empfangsbuffer. So bleibt der Telegrammfluss gewährleistet.

Beispiel Telegramme

TCP/IP	Socket receive			Receive buffer			ASCII command	
	String	length		String	size		String	
<i>telegram</i> <i>n</i>	'@S1\$R\$N>'	6	-->	'@S1\$R\$N>'	6	-->	@S1	
<i>n+1</i>	'TP\$R\$N-1\$R\$N>'	9	-->	'TP\$R\$N-1\$R\$N>'	9	-->	TP-1	
<i>n+2</i>	'@S1\$R\$N>TM'	8	-->	'@S1\$R\$N>TM'	8	-->	@S1	
<i>n+3</i>	'C\$R\$N-301\$R\$N>'	9	-->	'TMC\$R\$N-301\$R\$N>'	11	-->	TMC-301	
<i>n+4</i>	'@S1\$R\$N>T'	10	-->	'@S1\$R\$N>T'	10	-->	@S1	
<i>n+5</i>	'P\$R\$N-1\$R\$N>'	11	-->	'TP\$R\$N-1\$R\$N>'	12	-->	TP-1	
<i>n+6</i>	'TMC\$R\$N-312\$R\$N>'	12	-->	'TMC\$R\$N-312\$R\$N>'	12	-->	TMC-312	



exceptional motion controls

Prompt „>“ abwarten vor nächstem Befehl

Bevor ein neuer Befehl abgesetzt wird muss das Promptzeichen „>“ abgewartet werden. Dann testen, dass kein „?“ im Empfang String vorhanden ist, sonst wurde der Befehl nicht erkannt. Bei >TP ist die z.B. das Sendetelegramm viel kürzer als das Empfangstelegramm, weil dieses den Positionswert enthält. Es darf kein TP Befehl gesendet werden bevor die Antwort vom vorgängigen TP vollständig eingetroffen ist mit dem Promptzeichen „>“ als Abschluss.

Kommunikation via Ethernet TCP/IP Socket mit automatischen Event Mitteilungen

Beispiel

Open Socket	(<i>IP_Address Port=10001</i>)
Senden	<CR> Carriage Return
Wiederholen <CR>	bis prompt „>“ kommt
Event Mitteilungen auf ON	evt1 --> <i>@s0 kommt sofort als Bestätigung, dass Event Modus aktiv ist, unabhängig vom Status</i>
Status Antworten	@s0 (Power OFF) @s1 (Achse Stillstand, in Position) @s2 (Achse in Bewegung) @s9 (Error, Power OFF) @h (Achse referenziert)
Referenz fahren	<i>ASCII Kommandos</i> >ref
Beispiel Antwort-Sequenz	@s2 (Achse in Bewegung, Referenz)* @s1 (Achse Stillstand, mit Umkehr)* @s2 (Achse in Bewegung, Referenzmarken) @s1 (Achse Stillstand, in Position) @h (Achse referenziert) <i>)* kommen nur falls Fahrt auf mechanische Begrenzung oder auf Endanschlag. Dann wird die Fahrrichtung automatisch umgekehrt und die Absolutposition in Gegenrichtung gesucht. Im Applikations- programm @h abwarten.</i>
Fahren auf Absolutposition	>g12000 (Auf Position 12000Increment)
Autom. Antworten	@s2 (Achse in Bewegung) @s1 (Achse Stillstand, in Position)



exceptional motion controls

2. TCP/IP Erfassung über Wireshark

Senden eines korrekten
Kommandos z.B. "TE"

```
112 13.928681 195.186.17.36 192.168.2.72 TCP 60 https > 54199 [ACK] Seq=4339 Ack=1504 win=31711 Len=0
113 13.978659 192.168.2.72 192.168.2.142 TCP 57 54196 > scp-config [PSH, ACK] Seq=23 Ack=58 Win=64343 Len=3
114 13.981035 192.168.2.142 192.168.2.72 TCP 62 scp-config > 54196 [PSH, ACK] Seq=58 Ack=26 Win=2047 Len=8
[+] Frame 113: 57 bytes on wire (456 bits), 57 bytes captured (456 bits)
[+] Ethernet II, Src: HewlettP_3a:9e:db (64:31:50:3a:9e:db), Dst: Pronet_e0:8d:fe (00:20:4a:e0:8d:fe)
[+] Internet Protocol Version 4, Src: 192.168.2.72 (192.168.2.72), Dst: 192.168.2.142 (192.168.2.142)
[+] Transmission Control Protocol, Src Port: 54196 (54196), Dst Port: scp-config (10001), Seq: 23, Ack: 58, Len: 3
[+] Data (3 bytes)
0000 00 20 4a e0 8d fe 64 31 50 3a 9e db 08 00 45 00 . J...dl P:....E.
0010 00 2b 72 fd 40 00 80 06 00 00 c0 a8 02 48 c0 a8 .+P.@. . . . .H.
0020 02 8e d3 b4 27 11 5e 7d e5 66 00 48 39 5a 50 18 . . . . } .f.H9ZP.
0030 fb 57 86 44 00 00 54 45 0d .w.D..TE .
```

Address
Command TE (Ascii Nr. T = 54 E= 45)
Carriages return (Ascii Nr. 0d)

Nach dem Senden von <CR> **keine weiteren
Charakter senden** bis das Prompt
">" erscheint.

Receive

```
112 13.928681 195.186.17.36 192.168.2.72 TCP 60 https > 54199 [ACK] Seq=4339 Ack=1504 win=31711 Len=0
113 13.978659 192.168.2.72 192.168.2.142 TCP 57 54196 > scp-config [PSH, ACK] Seq=23 Ack=58 Win=64343 Len=3
114 13.981035 192.168.2.142 192.168.2.72 TCP 62 scp-config > 54196 [PSH, ACK] Seq=58 Ack=26 Win=2047 Len=8
[+] Frame 114: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
[+] Ethernet II, Src: Pronet_e0:8d:fe (00:20:4a:e0:8d:fe), Dst: HewlettP_3a:9e:db (64:31:50:3a:9e:db)
[+] Internet Protocol Version 4, Src: 192.168.2.142 (192.168.2.142), Dst: 192.168.2.72 (192.168.2.72)
[+] Transmission Control Protocol, Src Port: scp-config (10001), Dst Port: 54196 (54196), Seq: 58, Ack: 26, Len: 8
[+] Data (8 bytes)
0000 64 31 50 3a 9e db 00 20 4a e0 8d fe 08 00 45 00 dIP:... J....E.
0010 00 30 aa b3 40 00 40 06 09 ee c0 a8 02 8e c0 a8 .0..@.@. ....
0020 02 48 27 11 d3 b4 00 48 39 5a 5e 7d e5 69 50 18 .H'...H 9Z^}.iP.
0030 07 ff 0d b5 00 00 54 45 0d 0a 30 0d 0a 3e .....TE ..0..>
```

Address
Command TE (Ascii Nr. T = 54 E= 45)
Carriages return (Ascii Nr. 0d)
Line feed (Ascii Nr. 0a)
Line feed (Ascii Nr. 0a)
Prompt ">" (Ascii Nr. 3e)



exceptional motion controls

Senden eines inkorrekten Kommandos z.B. "TX"

```
120 15.702770 192.168.2.72 192.168.2.142 TCP 57 54196 > scp-config [PSH, ACK] Seq=26 Ack=66 win=64335 Len=3
121 15.704880 192.168.2.142 192.168.2.72 TCP 62 scp-config > 54196 [PSH, ACK] Seq=66 Ack=29 win=2047 Len=8
122 15.902544 192.168.2.72 192.168.2.142 TCP 54 54196 > scp-config [ACK] Seq=29 Ack=74 win=64327 Len=0

Frame 120: 57 bytes on wire (456 bits), 57 bytes captured (456 bits)
Ethernet II, Src: HewlettP_3a:9e:db (64:31:50:3a:9e:db), Dst: Pronet_e0:8d:fe (00:20:4a:e0:8d:fe)
Internet Protocol Version 4, Src: 192.168.2.72 (192.168.2.72), Dst: 192.168.2.142 (192.168.2.142)
Transmission Control Protocol, Src Port: 54196 (54196), Dst Port: scp-config (10001), Seq: 26, Ack: 66, Len: 3
Data (3 bytes)
0000 00 20 4a e0 8d fe 64 31 50 3a 9e db 08 00 45 00 . J...d1 P:....E.
0010 00 2b 73 00 40 00 80 06 00 00 c0 a8 02 48 c0 a8 .+S.@... ..H..
0020 02 8e d3 b4 27 11 5e 7d e5 69 00 48 39 62 50 18 .H'....H 9b^}.i.H9bP.
0030 fb 4f 86 44 00 00 54 58 0d .O.D..TX .
```

Address
Command TX (Ascii Nr. T = 54 X = 58)
Carriages return (Ascii Nr. 0d)

Receive

```
120 15.702770 192.168.2.72 192.168.2.142 TCP 57 54196 > scp-config [PSH, ACK] Seq=26 Ack=66 win=64335 Len=3
121 15.704880 192.168.2.142 192.168.2.72 TCP 62 scp-config > 54196 [PSH, ACK] Seq=66 Ack=29 win=2047 Len=8
122 15.902544 192.168.2.72 192.168.2.142 TCP 54 54196 > scp-config [ACK] Seq=29 Ack=74 win=64327 Len=0

Frame 121: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: Pronet_e0:8d:fe (00:20:4a:e0:8d:fe), Dst: HewlettP_3a:9e:db (64:31:50:3a:9e:db)
Internet Protocol Version 4, Src: 192.168.2.142 (192.168.2.142), Dst: 192.168.2.72 (192.168.2.72)
Transmission Control Protocol, Src Port: scp-config (10001), Dst Port: 54196 (54196), Seq: 66, Ack: 29, Len: 8
Data (8 bytes)
0000 64 31 50 3a 9e db 00 20 4a e0 8d fe 08 00 45 00 d1P:... J....E.
0010 00 30 aa b4 40 00 40 06 09 ed c0 a8 02 8e c0 a8 .0..@.@. ....
0020 02 48 27 11 d3 b4 00 48 39 62 5e 7d e5 6c 50 18 .H'....H 9b^}.iP.
0030 07 ff fe 96 00 00 54 58 0d 0a 3f 0d 0a 3e .....TX ..?..>
```

Address
Command TX (Ascii Nr. T = 54 X = 58)
Carriages return (Ascii Nr. 0d)
Line feed (Ascii Nr. 0a)
Answer "?" (Ascii Nr. 3f)
Prompt ">" (Ascii Nr. 3e)

Jenny Science AG
Sandblatthe 7a
CH-6026 Rain, Schweiz

Tel +41 (0) 41 455 44 55
Fax +41 (0) 41 455 44 50

www.jennyscience.ch
info@jennyscience.ch