



# Projekt: TCP/IP Datenserver

Projekttyp: Software

Dokumentationstyp: Kundeninformation

Ziele: Versendung der aufgezeichneten Rohdaten und / oder korrigierten/berechneten Daten im NEURO PRAX ® / THERA PRAX ® System über eine Ethernet-Schnittstelle mittels TCP/IP – Protokoll

Versendung zusätzlicher Informationen, z.B. Name des Patienten, Datum, Zeit der Messung, gewählte Kanäle und Einheiten, Impedanzwerte, usw.

# Inhaltsverzeichnis

1 Allgemeines.....	3
1.1 Historie.....	3
1.2 Begriffe.....	3
2 Datenservice.....	4
2.1 Überblick.....	4
2.2 Datenservice für Rohdaten im Netzwerk.....	4
2.3 Datenservice für korrigierte Daten im Netzwerk.....	5
2.4 Serviceerkennung durch Clients.....	6
2.4.1 Suche nach dem Datenservice im Netzwerk (service discovery).....	6
2.4.2 Auflösung des Datenservice im Netzwerk (resolve service).....	7
3 Übertragungsprotokolle.....	9
3.1 Definition der einzelnen Übertragungsprotokolle.....	9
3.2 General Information Protocol (Version 1).....	10
3.3 Marker Name Protocol (Version 1).....	11
3.4 Buffer Overflow Protocol (Version 1).....	11
3.5 Impedance Status Protocol (Version 1).....	12
3.6 Data Protocol (Version 1).....	13
4 Beispielprogramm für Client.....	14
5 Beispielprotokolle.....	15
6 Referenzen und Literatur.....	17

# 1 Allgemeines

## 1.1 Historie

V1                      Erstfassung

## 1.2 Begriffe

dns-sd	domain name service – service discovery
IP	Internet protocol
NPSPL	NEUO PRAX ® Signal Processing Library
proprietär	unfrei; geistiges Eigentum der neuroConn GmbH
TCP	Transmission Control Protocol

## 2 Datenservice

### 2.1 Überblick

Das NEURO PRAX ® System soll Datendienste bereitstellen, die an andere PC-Systeme die aufgenommenen Rohdaten und / oder verarbeiteten Daten versendet. Das Versenden geschieht über eine Ethernet-Schnittstelle und unter Nutzung des TCP/IP-Protokolls. Das NEURO PRAX ® System fungiert dabei als Datenserver. Andere PC-Systeme (Clients) können sich mit dem Datenserver zu einem Client-Server-System verbinden und die Daten empfangen. Im ersten Entwicklungsansatz ist nur das Versenden der Daten vorgesehen, eine Beeinflussung (Steuerung) durch Kommandos des Clients jedoch nicht.

In der (beliebigen) Netzwerkkonfiguration soll der Datenserver des NEURO PRAX ® von den Clients jederzeit ohne spezielle Netzwerkkonfiguration gefunden werden. Dazu wird eine von Apple Software entwickelte Bonjour Technologie – auch bekannt als „Zero Configuration Networking“ – genutzt [1]. Diese Technologie ist für viele Betriebssysteme frei verfügbar. Für das Verständnis der nachfolgenden Darstellungen wird Grundlagenwissen für folgende Bereiche vorausgesetzt:

- Bonjour-Technologie (z.B. Bonjour Printer Service [2])
- Umsetzung von „Zero Configuration Networking“ in verschiedenen Betriebssystemen / Programmiersprachen [1]
- kommandozeilen orientierte Abfragen von Netzwerkdiensten mit „dns-sd“ [1].

### 2.2 Datenservice für Rohdaten im Netzwerk

Das NEURO PRAX ® Systems soll nach Softwarestart automatisch einen Datenservice im Netzwerk anbieten:

Bezeichnung: neuroConn Data Service

Verantwortlich: neuroConn GmbH

Kontakt: [info@neuroconn.de](mailto:info@neuroconn.de)

Protokoll: proprietär

Transportprotokoll: TCP

Liste der TXT records: proprietär

Jedes NEURO PRAX ® Gerät, das diesen Service anbietet, registriert den Service als:

**<hostname>.\_neuroconn.\_tcp.local**

Mit Hilfe der Zeroconf-Technologie können alle im Netzwerk verfügbaren Services abgefragt werden. Zu jedem der Dienste kann die dazugehörige IP-Adresse und der Port, auf welchem der Service angeboten wird, ermittelt werden. Der wesentliche Vorteil besteht darin, dass sich über einen längeren Zeitraum die IP-Adresse und die Netzwerkkonfiguration für das NEURO PRAX ® System ändern kann. Der vom NEURO PRAX ® angebotene Service bleibt aber bestehen und kann damit jederzeit unabhängig von der IP-Adresse und der Netzwerkkonfiguration gefunden werden.

Über die „TXT record keys“ des Service werden zusätzliche Informationen zum Service übermittelt:

**Tabelle 1: TXT record keys für den Datenservice der Rohdaten.**

Key	Beschreibung	Wert
productID	Die Produktbezeichnung für den Service.	DataServerTCP
type	Eine genauere Bezeichnung des Servicetyps.	rawData
vendorID	Die Identifizierung des Herstellers, der diesen Datenservice anbietet.	neuroConn GmbH
softwareVersion	Die Softwareversion, die diesen Service anbietet. Dieser Wert kann benutzt werden, um anhand der Softwareversion bestimmte Funktionsmerkmale des Service zu unterscheiden.	1

## **2.3 Datenservice für korrigierte Daten im Netzwerk**

Die Algorithmen in der NPSPL korrigieren bestimmte Signale der Rohdaten oder erzeugen sogar neue Kanäle als zusätzliche Berechnungsergebnisse. Bearbeitete Rohdaten oder zusätzliche Signale werden in diesem Dokument als „korrigierte Daten“ bezeichnet. Für die korrigierten Daten wird ein eigener Datenservice vom NEURO PRAX ® System angeboten. Da mehrere Servicedienste von einem NEURO PRAX ® System angeboten werden können, sollten die Clients anhand des TXT-Eintrages „type“ die Unterscheidung nach Rohdaten oder korrigierten Daten vornehmen:

**Tabelle 2: Unterschiede in den TXT record keys für den Datenservice der korrigierten Daten.**

Key	Beschreibung	Wert
type	Eine genauere Bezeichnung des Servicetyps.	corrData

Die Datenservices werden über verschiedene Ports angeboten. Auch hier muss der Client die notwendigen Informationen zum Service vorher abfragen.

## 2.4 Serviceerkennung durch Clients

Es gibt zwei Wege, um Servicedienste im Netzwerk anzubieten, zu suchen oder aufzulösen:

- mit dem Kommandotool „dns-sd“ (Bonjour Printer Service)
- mit API-Funktionen in verschiedenen Programmiersprachen [1].

In diesem Dokument werden nur Beispiele mit dem Kommandotool „dns-sd“ gegeben. In [1] werden verschiedene API-Funktionen für die Programmiersprachen C, Java, CFNetwork, Cocoa, Ruby, Python behandelt.

### 2.4.1 Suche nach dem Datenservice im Netzwerk (service discovery)

Kommandotool:      `dns-sd -B <servicetype> <domain>`

Mit dem Kommandotool **dns-sd -B \_neuroconn.\_tcp** werden alle Instanzen (PCs und Geräte) angezeigt, die den Service „\_neuroconn.\_tcp“ in der lokalen Domäne anbieten. In Abbildung 1 ist ein Anwendungsbeispiel gegeben, bei dem von einem PC (Sydney 2) zwei dieser Dienste angeboten werden und diese Dienste aktiv sind (Kennzeichnung „ADD“).

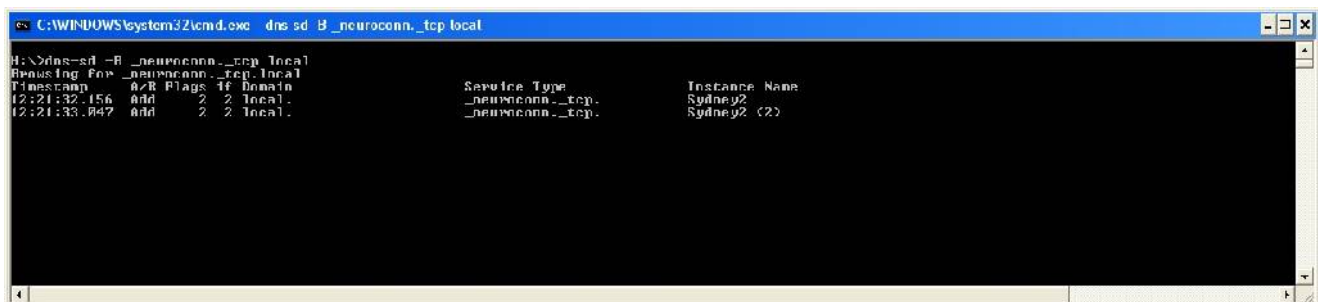


Abbildung 1: Suche nach einem Datenservice mit dem Kommandotool „dns-sd“.

Mit diesem dns-sd Befehl kann man nur herausfinden, ob \_neuroconn.\_tcp Dienste im Netzwerk angeboten werden. Was genau angeboten wird und wo diese Services zu erreichen sind, muss durch Auflösung des Service herausgefunden werden. Durch die Kennzeichnung „ADD“ wird angezeigt, dass die Services aktiv sind. Sobald ein Service nicht mehr angeboten wird, wird dieser mit „RMV“ gekennzeichnet.

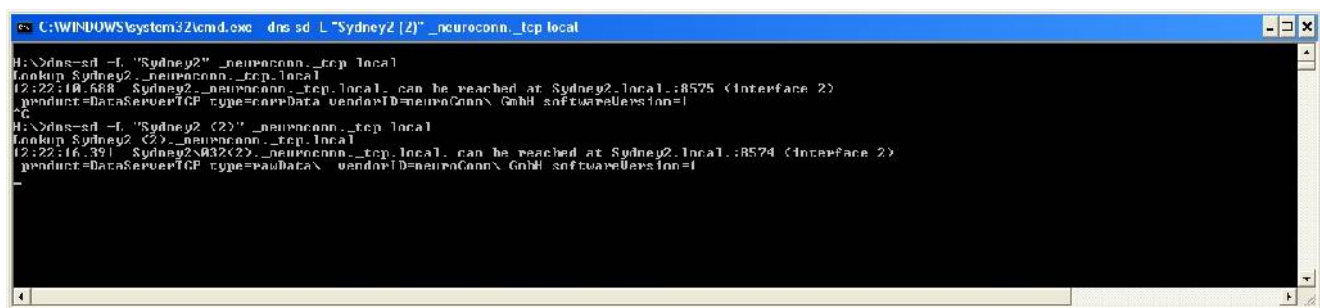
## 2.4.2 Auflösung des Datenservice im Netzwerk (resolve service)

Kommandotool: `dns-sd -L <instance name> <servicetyp> <domain>`

Zu jeder Instanz eines aktiven Service können abgefragt/aufgelöst werden:

- die Adresse und der Port, unter dem der Service aktuell erreichbar ist
- das genutzte Interface des Service (bei LAN und WLAN können unterschiedliche Services und Interfaces angeboten werden)
- die Inhalte der TXT-Record-Keys

In Abbildung 2 sind zwei Beispiele für die Auflösungen des Service „\_neuroconn.\_tcp“ für die Instanzen „Sydney2“ und „Sydney2 (2)“ gegeben. Der Service \_neuroconn.\_tcp ist für die Instanz „Sydney2“ erreichbar unter der Adresse „Sydney2.local“ mit dem Port 8575 auf Interface 2. Aus den TXT-Record-Keys geht hervor, dass dieser Dienst ein DataServerprodukt über TCP bereitstellt und dass es sich um korrigierte Daten handelt. Der Anbieter ist die Firma „neuroConn GmbH“ und die Softwareversion ist 1. Die Instanz „Sydney2 (2)“ kann genauso aufgelöst werden.

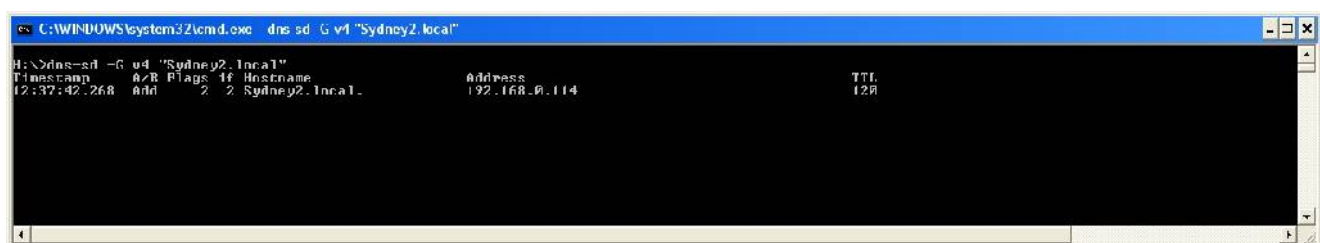


```
es C:\WINDOWS\system32\cmd.exe dns-sd -L "Sydney2 (2)" _neuroconn._tcp local
H:\>dns-sd -L "Sydney2" _neuroconn._tcp local
Lookup Sydney2._neuroconn._tcp.local
12:22:10.688 Sydney2._neuroconn._tcp.local: can be reached at Sydney2.local:8575 (Interface 2)
product=DataServerTCP type=capData vendorID=neuroConn GmbH softwareVersion=1
H:\>dns-sd -L "Sydney2 (2)" _neuroconn._tcp local
Lookup Sydney2 (2)._neuroconn._tcp.local
12:22:16.391 Sydney2 (2)._neuroconn._tcp.local: can be reached at Sydney2.local:8574 (Interface 2)
product=DataServerTCP type=capData vendorID=neuroConn GmbH softwareVersion=1
```

Abbildung 2: Auflösung mehrerer Datenservices mit dem Kommandotool „dns-sd“.

Soll die exakte IP-Adresse für eine Instanz herausgefunden werden, dann wird hierfür das Kommando:

`dns-sd -G v4 <Instanz>.<Domäne>` genutzt (vgl. Abbildung 3).



```
es C:\WINDOWS\system32\cmd.exe dns-sd -G v4 "Sydney2.local"
H:\>dns-sd -G v4 "Sydney2.local"
Timestamp 0x0 Flags 1f Hostname Address TTL
12:37:42.268 Add 2 2 Sydney2.local 192.168.0.114 120
```

Abbildung 3: Aktuelle IP-Adresse für die Instanz „Sydney2.local“.

Durch unterschiedliche Netzkonfigurationen kann sich die IP-Adresse jederzeit ändern. Der angebotene Service und die Instanz (=Hostname) ändern sich dagegen nicht. Für Clients, die den \_neuroconn.\_tcp Service nutzen wollen, wird deshalb folgender Weg empfohlen:

- Suche nach allen aktiven Services vom Typ \_neuroconn.\_tcp

- Auflösung der Instanzen
- Auswahl der Instanzen für Client-Server-Verbindungen
- Ermittlung der IP-Adresse und des Ports für die Client-Server-Verbindung
- Verbindung zum Client herstellen



## 3 Übertragungsprotokolle

### 3.1 Definition der einzelnen Übertragungsprotokolle

#### DataServerTCP – Protocols : Overview (Version 1)

protocol type	protocol name	event for transmission to TCP/IP client
1	general information protocol (GIP)	The user has to set up the EEG machine for recording (number of electrodes, sampling frequency, patient's name, etc). After selecting a new session (record) the customer switches to the measuring mode (selects the tap "Measuring"). Every time the user switches to the measuring mode, the NEURO PRAX software sends out a general information protocol. Additionally, at the end of the recording an updated general information protocol will be sent.
2	marker name protocol (MNP)	This protocol contains the available marker names and values. It will be sent out to the TCP/IP client after switching from the tap "Patient" to the tap "Measuring" for a new session/recording.
3	impedance status protocol (ISP)	After preparation of electrode-to-skin-contact the user can check the impedances with the NEURO PRAX system. Every time the impedances has been checked, the NEURO PRAX system sends out the impedance status protocol with the updated impedance results.
4	data protocol (DP)	This protocol contains the recorded and/or processed data, the current sample index, a.s.o.
5	buffer overflow protocol (BOP)	Notification to the clients (buffer overflow at server).

**Similarities:**

character encoding:	Latin-1
field delimiter:	\$
protocol fields:	fixed size, each field contains one string
string alignment:	Left-aligned for text string filled with blanks Right-aligned for value strings starting with blanks
no. of fields:	not fixed, depends on the number of EEG channels and on the protocol type
EEG/EXG data format:	binary data, float precision (4 Bytes), little-endian byte ordering
protocol start:	neuroConn\$<protocolType>\$<protocolName>\$<protocolVersion>\$.....
protocol end:	end\$
channel / sample indexing:	starts with 0

## 3.2 General Information Protocol (Version 1)

parameter number	parameter name	parameter alias	size(bytes)	string	example
			incl. Delimiter \$	alignment	
1	neuroConn protocol	ProtocolStart	10	left	neuroConn
2	protocol type	ProtocolType	4	right	1
3	protocol name	ProtocolName	18	left	DataServerTCP-GIP
4	protocol version	ProtocolVersion	4	right	1
5	recording: file name	recFileName	19	left	20100329151422.EEG
6	recording: path name	recPathName	255	left	d:\neuroprax\datafiles
7	database: patient's name	DB_PatName	255	left	Mr. Public
8	database: patient's first name	DB_PatFirstName	255	left	John Q.
9	database: patient's birthday	DB_PatBirthday	11	left	2010-04-08
10	database: patient's identification number	DB_PatINR	255	left	101
11	name of the electrode setup	elecSetup	255	left	EEG-27-EP
12	sampling frequency	smpFreq	6	right	4000
13	name of selected algorithm	selAlgorithm	255	left	TMS
14	total number of channels	numChannels	5	right	4
15	number of EXG channels	numEXGchannels	5	right	4
16	channel names	ChannelNames	9 x numChannels	left	Fp1,Fp2,F7,F3
17	channel types	ChannelTypes	9 x numChannels	left	EEG,EEG,EEG,EEG
18	channel units	ChannelUnits	9 x numChannels	left	uV,uV,uV,uV
19	channel references	ChannelReference	9 x numChannels	left	GND,GND,GND,GND
20	protocol end	ProtocolEnd	4	left	end

\* birthday format:  
YYYY-MM-DD  
(similar to ISO 8601)

protocol size:           fixed:                           1616  
                          variable:               36 \* numChannels

### 3.3 Marker Name Protocol (Version 1)

parameter number	parameter name	parameter alias	size(bytes)	string	example
			incl. Delimiter \$	alignment	
1	neuroConn protocol	ProtocolStart	10	left	neuroConn
2	protocol type	ProtocolType	4	right	2
3	protocol name	ProtocolName	18	left	DataSetServerTCP-MNP
4	protocol version	ProtocolVersion	4	right	1
5	marker count	mC	4	right	10
6	marker index 1	markerIndex1	7	right	16384
7	marker name 1	markerName1	33	left	Start Record
...	...	...	...	...	...
...	marker index mC	markerIndex-mC	7	right	16384
...	marker name mC	markerName-mC	33	left	
...	protocol end	ProtocolEnd	4	left	end

protocol size:      fixed:                      44  
                          variable:                40 \* mC

### 3.4 Buffer Overflow Protocol (Version 1)

parameter number	parameter name	parameter alias	size(bytes)	string	example
			incl. Delimiter \$	alignment	
1	neuroConn protocol	ProtocolStart	10	left	neuroConn
2	protocol type	ProtocolType	4	right	5
3	protocol name	ProtocolName	18	left	DataSetServerTCP-BOP
4	protocol version	ProtocolVersion	4	right	1
5	protocol end	ProtocolEnd	4	left	end

protocol size:      fixed:                      40                      Bytes

### 3.5 Impedance Status Protocol (Version 1)

parameter number	parameter name	parameter alias	size(bytes) incl. Delimiter \$	string alignment	example
1	neuroConn protocol	ProtocolStart	10	left	neuroConn
2	protocol type	ProtocolType	4	right	3
3	protocol name	ProtocolName	18	left	DataSetTCP-ISP
4	protocol version	ProtocolVersion	4	right	1
5	number of EXG channels	numEXGchannels	5	right	4
6	channel name 0	chName0	9	left	Fp1
7	impedance status 0	impStatus0	3	right	0*
8	channel name 1	chName1	9	left	Fp2
9	impedance status 1	impStatus1	3	right	-1*
10	channel name 2	chName2	9	left	F7
11	impedance status 2	impStatus2	3	right	-2*
...					
...					
...					
(K-1)*2+6	channel name (K-1) **	chName(K-1)	9	left	O2
(K-1)*2+7	impedance status (K-1) **	impStatus(K-1)	3	right	-2*
(K-1)*2+8	protocol end	ProtocolEnd	4	left	end

\* impedance status

\*\*

-2 = poor contact, inacceptable (color: red) <as shown in the NEURO PRAX>

K denotes the number of EXG channels

-1 = worthy of improvement (color: yellow) <not shown in the NEURO PRAX>

0 = good contact (color: green) <as shown in the NEURO PRAX>

protocol size: fixed: 45  
variable: 12 \* numEXGchannels

### 3.6 Data Protocol (Version 1)

parameter number	parameter name	parameter alias	size(bytes) incl. Delimiter \$	string alignment	example
1	neuroConn protocol	ProtocolStart	10	left	neuroConn
2	protocol type	ProtocolType	4	right	4
3	protocol name	ProtocolName	18	left	DataServerTCP-DP
4	protocol version	ProtocolVersion	4	right	1
5	sample index	sampleIndex	12	right	2147483647
6	number of samples (=N)	numSmp	12	right	2147483647
7	number of EXG channels (=K)	numChannels	12	right	2147483647
8	data	data	<see below>	<binary>	
9	protocol end	ProtocolEnd	4	left	end

\*\*

The time stamp is measured in samples  
after start record.

#### data:

matrix data (=x) with K channels (K=numChannels) and N sample s(=numSmp)  
precision: float (4 Bytes)  
machine format: IEEE floating point with little-endian byte ordering

$x[0][0] \ x[1][0] \ x[2][0] \ ... \ x[K-1][0]$

$x[0][1] \ x[1][1] \ x[2][1] \ ... \ x[K-1][1]$

$x[0][2] \ x[1][2] \ x[2][2] \ ... \ x[K-1][2]$

...

$x[0][N-1] \ x[1][N-1] \ x[2][N-1] \ ... \ x[K-1][N-1]$

protocol size:

76

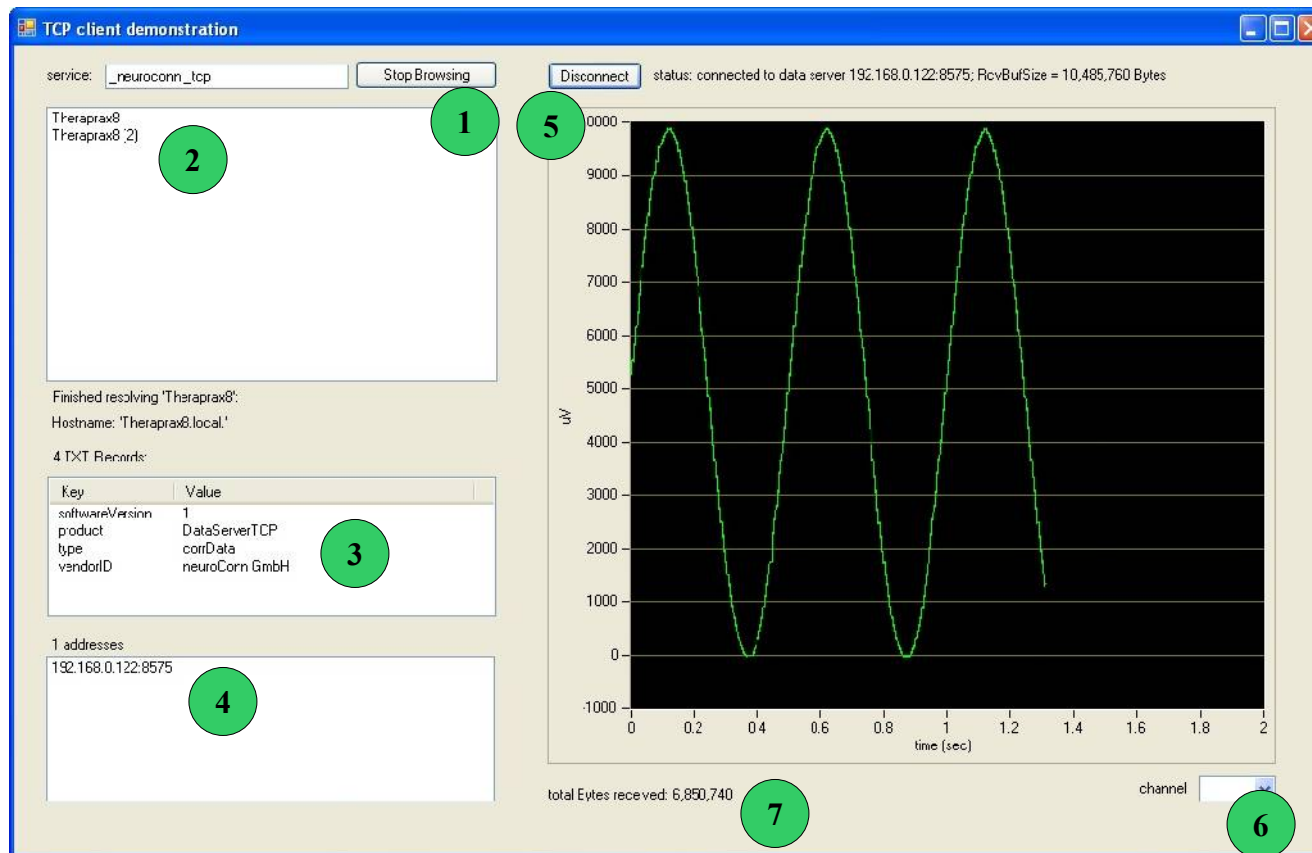
Bytes

$4 * \text{noChannels} * \text{noSamples}$

Bytes

## 4 Beispielprogramm für Client

Im Rahmen der Entwicklung und für Testzwecke wurde ein Beispielprogramm eines Datenclients in C# entwickelt. Dieses Programm ermöglicht das Suchen von Datendiensten ①, das Auflösen der Datendienste ②, das Anzeigen der TXT-Record-Einträge des Datendienstes ③, das Auflösen der IP-Adresse und des Ports des Datenservers ④, das Verbinden mit dem Datenserver ⑤, die Online-Anzeige der Daten mit Kanalauswahl ⑥ und die Registrierung der Anzahl empfangener Bytes ⑦.



## 5 Beispielprotokolle

### General Information Protocol:

```
neuroConn$ 1$DataServerTCP-GIP$ 1$20101001094539.EEG$D:\Neuroprax\Datafiles\  
$-  
$-  
$3000-01-01$-1  
$-  
$ 4000$  
$ 30$ 27$Fp1 $Fp2 $F7 $F3 $Fz $F4 $F8 $T3 $C3 $Cz $C4 $T4 $Cp5 $Cp1 $Cp2 $Cp6  
$T5 $P3 $Pz $P4 $T6 $O1 $O2 $VEOG_I $VEOG_II $ECG1 $ECG2 $REF_EEG $DTRIG $MRK $EEG  
$EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG $EEG  
$EEG $EEG $EEG $EEG $EEG $EEG $EEG $EOG $EOG $ECG $ECG $REF_EEG $DIG $MRK $μV  
$μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV  
$μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $μV $ $GND $GND $GND $GND  
$GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND  
$GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $GND $end$
```

### Marker Name Protocol:

```
neuroConn$ 2$DataServerTCP-MNP$ 1$ 36$ 3$FB+ / EP1 $ 8$FBQuote+ / EP1_END $ 4$FB- / EP2  
$ 9$FBQuote- / EP2_END $ 1$TR+ / EP3 $ 6$TRQuote+ / EP3_END $ 2$TR- / EP4  
$ 7$TRQuote- / EP4_END $ 67$Pause $ 68$PauseEnd $ 16384$StartRecord $  
16513$ImpCheckEnd $ 100$Eyes closed $ 101$Eyes open $ 102$Calculate $  
103$Calculate Stop $ 104$Light on $ 105$Light off $ 106$Hypervent on $  
107$Hypervent off $ 108$Move head $ 109$Look right $ 110$Look left $ 111$Leg  
left $ 112$Leg right $ 113$Arm left $ 114$Arm right $ 115$Baseline  
$ 116$Read $ 117$Listen $ 118$Paint $ 119$Sleep 1 $ 120$Sleep 2  
$ 121$Sleep 3 $ 122$Sleep 4 $ 123$Sleep 5 $end$
```

### **Impedance Status Protocol:**

neuroConn\$ 3\$DataServerTCP-ISP\$ 1\$ 27\$Fp1 \$-2\$Fp2 \$-2\$F7 \$-2\$F3 \$-2\$Fz \$-2\$F4 \$-2\$F8 \$-2\$T3 \$-2\$C3  
\$-2\$Cz \$-2\$C4 \$-2\$T4 \$-2\$Cp5 \$-2\$Cp1 \$-2\$Cp2 \$-2\$Cp6 \$-2\$T5 \$-2\$P3 \$-2\$Pz \$-2\$P4 \$-2\$T6 \$-  
2\$O1 \$-2\$O2 \$-2\$VEOG\_I \$-2\$VEOG\_II \$-2\$ECG1 \$-2\$ECG2 \$-2\$end\$

### **Data Protocol:**

neuroConn\$ 4\$DataServerTCP-DP \$ 1\$ 0\$ 63\$ 30\$Ð%—Å”á BdŸBá žB£“œB

...

end\$



## 6 Referenzen und Literatur

- [1] Cheshire, S., Steinberg, D.H. Zero Configuration Networking. O'Reilly Media, Inc. 2006. ISBN 0-596-10100-7
- [2] Bonjour for windows:  
<http://www.apple.com/de/support/bonjour/>  
  
Bonjour Services for Windows:  
<http://support.apple.com/kb/DL999>
- [3] Donahoo, Michael J., Calvert, Kenneth L. TCP/IP Sockets in C: practical guide for programmers. 2<sup>nd</sup> edition. Amsterdam: Morgan Kaufmann Publishers 2009. ISBN 978-0-12-374540-8
- [4] <http://www.win32developer.com/tutorial.shtml>
- [5] Testprogramm Advanced TCP/IP Data Logger:  
  
<http://www.aggsoft.com/>
- [6] Makofske, David B., Donahoo, Michael J., Calvert, Kenneth L. TCP/IP sockets in C#: practical guide for programmers. Amsterdam: Morgan Kaufmann Publishers 2004. ISBN 0-12-466051-7