PS 2000 B Programmierhandbuch

PS 2000 B Programming Guide

Jump to english version

Stand: 06.06.2016

Inhaltverzeichnis

1. Allgemeines ... 3
1.1 Einleitung ... 3
1.2 Treiberinstallation ... 3
1.2.1 Probleme bei der Treiberinstallation beheben ... 4
1.3 Begriffe ... 5
2. Das Kommunikationsprotokoll ... 5
2.1 Aufbau der Kommunikation ... 5
2.2 Serielle Übertragungsparameter ... 5
2.3 Sollwerte und Istwerte umrechnen ... 6
2.3.1 Istwerte ... 6
2.3.2 Sollwerte ... 6
2.4 Telegrammaufbau ... 6
2.4.1 Beispiel für ein Telegramm ... 7
3. Kommunikation mit dem Gerät ... 8
3.1 Vorgehensweise ... 8
3.2 Startdelimiter erzeugen ... 8
3.3 Die Maske für Objekte mit Steuerbyte ... 9
3.4 Beispiele ... 9
3.4.1 Gerät in Fernsteuerbetrieb umschalten ... 9
3.4.2 Status abfragen ... 9
3.5 Mögliche Probleme beim Setzen von Zuständen ... 10
3.6 Rückmeldungen ... 10
3.7 Fehlerbehandlung ... 11
3.8 Objektliste ... 11

1. Allgemeines

1.1 Einleitung

Diese Anleitung dient zur Erläuterung der binären Kommunikation mit einem Gerät der Serie PS 2000 B über dessen USB-Anschluß. Der USB-Treiber erzeugt auf der PC-Seite für das Gerät einen virtuellen COM-Port (VCP), über den die Kommunikation stattfindet. Die Verwendung des COM-Ports vereinfacht die Ansteuerung des Ports auf ein Minimum an Konfiguration und Code. Der COM-Port muß lediglich geöffnet und nach dem Öffnen einmal konfiguriert werden. Die Daten werden dann immer an den geöffneten Port geschickt.

Hinweis: diese Anleitung erläutert keine Methoden, wie in den diversen Programmiersprachen Hardware angesprochen, benutzt und konfiguriert wird. Informationen dazu sind in einschlägigen Quellen, wie Büchern oder dem Internet, zu finden.

1.2 Treiberinstallation

Verbinden Sie das Gerät mittels des mitgelieferten Kabels mit dem PC. Achten Sie darauf, den Mini-USB-Stecker vorn am Gerät richtig einzustecken (erfordert möglicherweise etwas Kraftaufwand). Falls das Kabel für den Standort des Gerätes zu kurz sein sollte, kann es mittels eines USB-Hubs oder eines USB-Verlängerungskabels verlängert werden (max. 5m insgesamt). Windows sollte daraufhin einen Treiberinstallationsvorgang starten, welcher nach einem Treiber fragt:

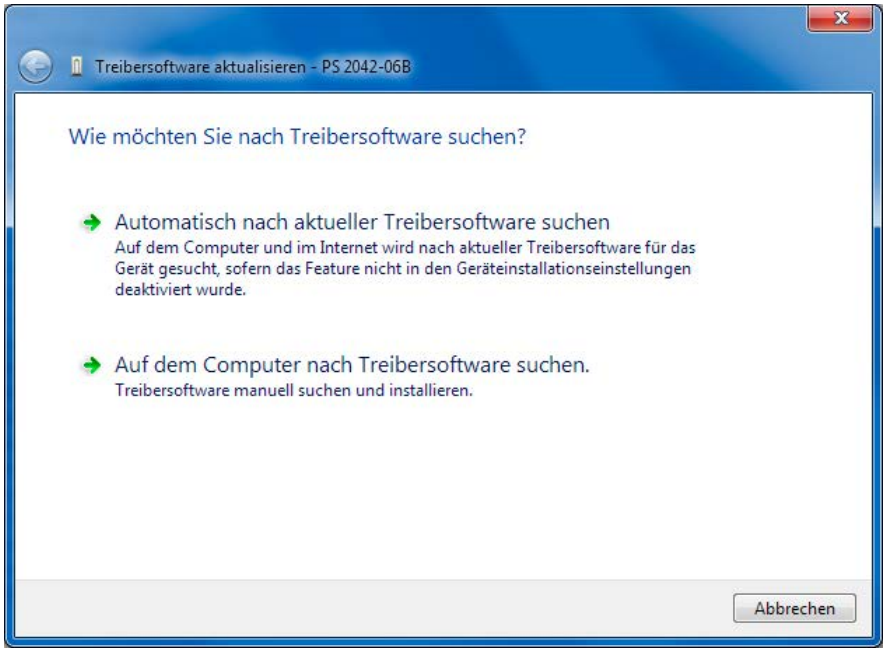


Bild 1

Wählen Sie hier die unterste Option (Windows XP/Vista/7). In den folgenden Fenstern verweisen Sie den Dialog dann auf den Treiber, der im Ordner driver\ auf der mitgelieferten CD zu finden ist. Es gibt jeweils eine Datei für Single-Modelle (1x DC-Ausgang) und Triple-Modelle (3x DC-Ausgang). Windows installiert für jedes neue Gerät der Treiber einmal. Ist der Treiber auf dem System noch nicht vorhanden, so muß er nur beim ersten Gerät auf Festplatte installiert werden. Nach erfolgreicher Installation geschieht nichts weiter. Man kann die Installation überprüfen, indem man den Windows Geräte-Manager aufruft. Dort sollte dann ein

 PS 2000 Single (COM8) oder PS 2000 Triple (COM8), je nach Modell,

zu finden sein. Der COM-Port COM8 ist hier nur ein Beispiel.

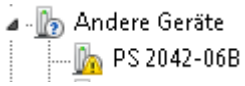
1.2.1 Probleme bei der Treiberinstallation beheben

Windows sollte nach dem Treiber fragen, sofern dieser noch nicht auf dem PC installiert ist. Geschieht dies nicht, öffnen Sie den Windows Gerätemanager:

XP: Startmenü (oder Desktop) → Rechtsklick auf „Arbeitsplatz“ → Eigenschaften → Tab „Hardware“ → Knopf „Geräte-Manager“

Vista / Windows 7: Startmenü (oder Desktop) → Rechtsklick auf „Computer“ → Eigenschaften → Link „Geräte-Manager“ oder in Suchleiste eintippen

Dort finden Sie in „Andere Geräte“ das zu installierende Gerät, wie z. B.:



Dann Rechtsklick auf das Gerät und „Treiber(software) aktualisieren“. Es erscheint der bekannte Windows-Treiber-Dialog. Wählen Sie den untersten Punkt (Beispiel von Windows 7, Vista und XP sind ähnlich):

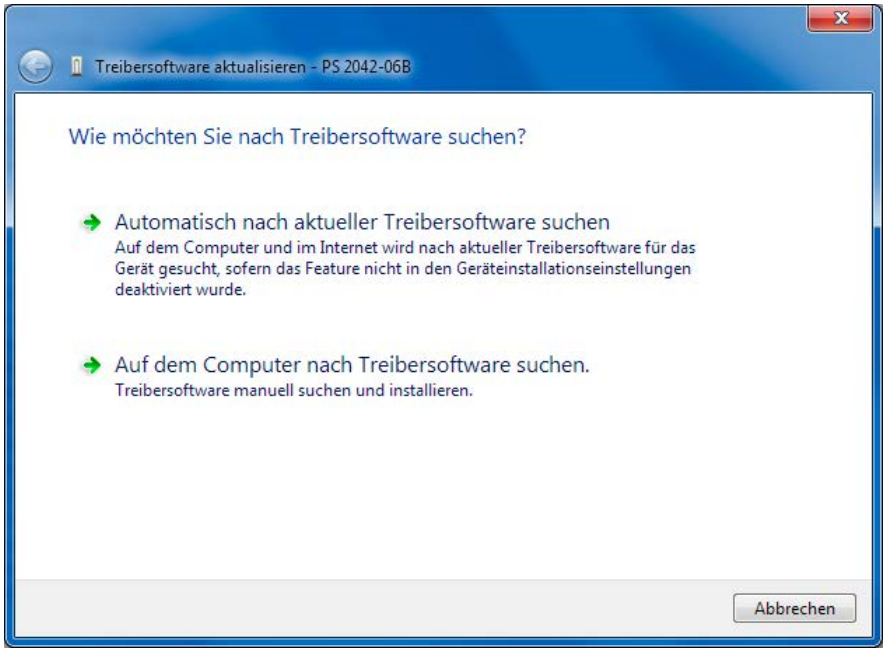


Bild 2

Im folgenden Fenster den Pfad zum Treiber auf der mitgelieferten CD bzw. Ordner \driver aus der entpackten ZIP-Datei angeben:

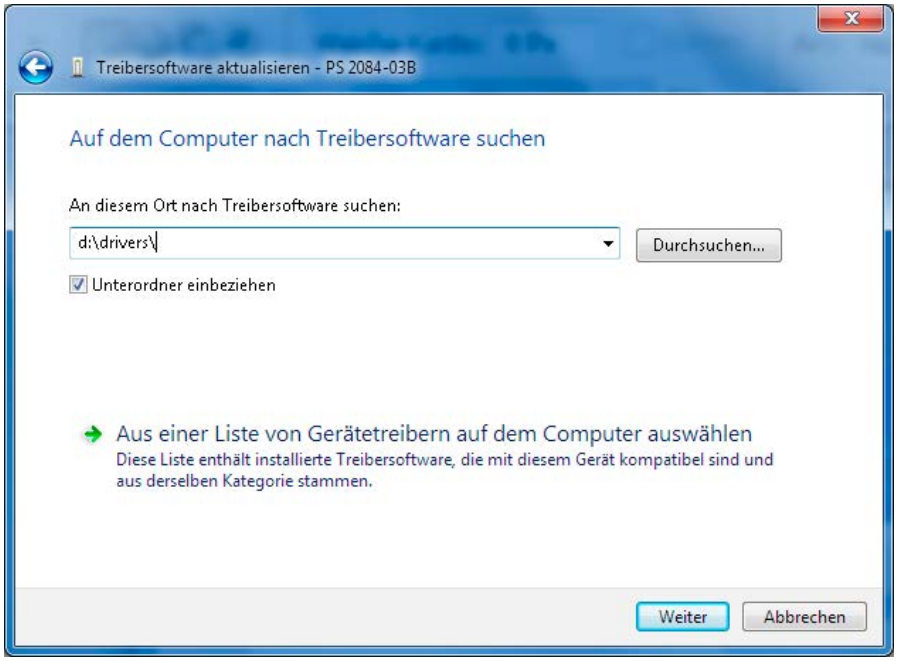
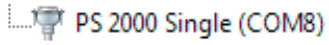


Bild 3

Windows installiert nun das Gerät. Nach erfolgreicher Installation ist das Gerät mit der Bezeichnung „PS2000 Single“ oder „Triple“ im Gerätemanager unter „Anschlüsse (COM & LPT)“ zu finden.



Hinweis: Die COM-Port-Nummer hier ist ein Beispiel. Windows vergibt für jedes dieser Geräte einen neuen COM-Port, die es sich „merkt“.

Hinweis: Falls mehrere Geräte am PC angeschlossen sind, erscheinen auch mehrere „PS2000 Single“ im Gerätemanager.

1.3 Begriffe

Telegramm = Kette von hexadezimalen Bytes, mit unterschiedlicher Länge. Wird entweder zum Gerät gesendet oder vom Gerät empfangen. Ein Teil des Telegramms (Datenbereich) repräsentiert entweder hexadezimale Werte oder ASCII-Strings.

2. Das Kommunikationsprotokoll

2.1 Aufbau der Kommunikation

Die Kommunikation mit dem zu steuernden Gerät basiert auf diesen zwei Telegrammformen:

a) Sendung: es wird ein Objekt gesendet, das einen Wert, z.B. Spannung, setzen soll. Sofern dies im momentanen Betriebszustand des Gerätes zulässig ist, wird das Objekt akzeptiert und ausgeführt. Das Gerät sendet dann als Antwort eine Fehlermeldung mit Fehlercode 0. Falls die Ausführung des Befehls nicht zulässig oder das Telegramm fehlerhaft ist, kommt eine Fehlermeldung mit Fehlercode 1 oder höher.

b) Anfrage: es wird mittels eines Objekts eine Anfrage an das Gerät gesendet, worauf man eine Antwort erwartet. Ist die Anfrage für den momentanen Betriebszustand des Gerätes zulässig, wird sie ausgeführt und die Antwort, nach einer gewissen Verarbeitungszeit, sofort gesendet. Die Antwort enthält als Inhalt die angefragten Daten. Falls nicht zulässig, wird als Antwort eine Fehlermeldung gesendet.

2.2 Serielle Übertragungsparameter

Die Übertragung von Daten erfolgt trotz USB-Schnittstelle im Stil eines seriellen COM-Ports, also byteweise. Bei der Übertragung eines Bytes werden folgende Bits übertragen:

Startbit + 8 Datenbits + Paritätsbit + Stoppbit

Das Parität wird auf ungerade (engl.=odd) geprüft.

Es sind zur Konfiguration des COM-Ports auf der PC-Seite folgende Parameter zu setzen:

Baudrate: 115200 Bd

Parität: ungerade (odd)

Stoppbits: 1

Am Gerät kann diesbezüglich nichts eingestellt werden.

2.3 Sollwerte und Istwerte umrechnen

Sollwerte und Istwerte werden als Prozentwerte übertragen. Ein Wert von 0x6400 entspricht 100,00%. Wenn also ein Gerät eine Nennausgangsspannung von 42V hat, dann würde der übertragene Spannungsistwert beim Wert 0x3200 = 50,00% der Spannung, also 21V entsprechen.

Das Highbyte ist die Prozentzahl (0x64 = dezimal 100) und das Lowbyte die Nachkommastellen der Prozentzahl, wobei 0x500 dann 5% ergeben, jedoch 0x550 nicht 5,5%. Man muß die eingehenden Istwerte sowie die ausgehenden Sollwerte daher umrechnen.

2.3.1 Istwerte

Istwerte werden aus dem Gerät gelesen und liegen nach Erhalt der Antwort als hexadezimaler 16-Bit-Wert vor, der einen Prozentwert repräsentiert.

$$\text{Realer Istwert} = \frac{\text{Nennwert des Gerätes} * \text{Prozent-Istwert}}{25600}$$

Beispiel: Nennwert des Gerätes ist 42V, der prozentuale Istwert kam als 0x2454 = 9300. Nach der Formel ergibt sich der Istwert = 42 * 9300 / 25600 = 15,26V.

2.3.2 Sollwerte

Sollwerte müssen als hexadezimaler 16-Bit-Prozentwert gesendet werden und sind daher vor dem Schicken umzurechnen. Beim Auslesen verhält es sich wie bei Istwerten.

$$\text{Prozentualer Sollwert} = \frac{25600 * \text{Realer Sollwert}}{\text{Nennwert des Gerätes}}$$

Beispiel: der Sollwert soll 25,5V sein, der Nennwert d. Gerätes ist 42V. Nach der Formel ergibt sich ein Prozent-Sollwert = 25600 * 25,5 / 42 = 15543 = 0x3CB7. Nachdem der Sollwert 0x3CB7 an das Gerät gesendet wurde, sollte ein 42V-Modell die gewünschten 25,5V am Ausgang einstellen.

2.4 Telegrammaufbau

Das binäre Telegramm hat den Aufbau

Byte 0	Byte 1	Byte 2	Variable Bytes	2 Bytes
SD	**DN**	**OBJ**	**DATEN**	**CS**

und setzt sich aus diesen Bytes zusammen:

Byte 0: **SD** (start delimiter)

Der Startdelimiter kennzeichnet den Beginn des Telegramms und enthält die Länge der Daten, die Telegrammrichtung und den Telegrammtyp.

Bits 0-3: Datenlänge -1 des Datenbereichs im Telegramm (Bytes 3-18)

Die Datenlänge beim Senden bzw. die zu erwartende Datenlänge bei einer Anfrage müssen hier angegeben werden. Die zu erwartende Datenlänge ist aus der Objektliste, Spalte 6 zu entnehmen.

Bit 4: Richtung

0 = Nachricht vom Gerät an den PC

1 = Nachricht vom PC an das Gerät

Bit 5:

1 = Cast type, für Senden oder Anfragen immer auf 1 setzen, ist 0 bei Antworten

Bits 6+7: Sendungstyp

00= reserviert

01= Anfrage von Daten

10= Antwort auf eine Anfrage

11 = Senden von Daten

Byte 1: **DN** (device node)

Hier ist zu unterscheiden, ob ein Single- oder Triple-Modell angesprochen werden soll. Bei einem Single-Modell hat der DN keine Relevanz und kann irgendeinen Wert enthalten, ss wird jedoch empfohlen, dieses Byte immer auf 0x00 zu setzen. Bei einem Triple-Modell gilt folgendes:

- ein Triple-Modell PS 2000 B hat immer die zwei Device Nodes 0 und 1

- Device node 0 gehört zu Ausgang 1, Device node 1 gehört zu Ausgang 2

Das heißt, beim Setzen bzw. Abfragen von Sollwerten und Zuständen wird hier durch den Device node der Ausgang ausgewählt.

Byte 2: **OBJ** (object)

Die Kommunikationsobjekte eines Gerätes werden über die hier angegebene Zahl adressiert. In der Objektliste werden die weitere Funktion(en) oder Eigenschaften der Objekte beschrieben. Siehe Abschnitt 3.8.

Byte 3 - 18: **Daten**

Der Datenbereich kann 0-16 Bytes lang sein, die Länge des Telegramms variiert also. Bei einer Anfrage (PC -> Gerät) werden keine Daten übermittelt, der Datenbereich entfällt dann und ab Byte 3 folgt direkt die Checksumme. Nur bei einer Antwort (Netzgerät -> PC) oder beim Senden (PC -> Netzgerät) werden Daten übermittelt.

Letzte 2 Bytes: **CS** (check sum)

Die Position der Prüfsumme (check sum) ist stets am Ende des Telegramms. Die Prüfsumme wird über die einfache Addition aller vorherigen Bytes des Telegramms gebildet. Sie ist zwei Bytes lang. Das Highbyte wird vor dem Lowbyte gesendet.

2.4.1 Beispiel für ein Telegramm

Es sollen Istwerte abgefragt werden (Objekt 71). Das Abfragetelegramm müßte dann so aussehen (Hexwerte):

75 00 47 00 BC (Single-Modell bzw. Ausgang 1 eines Triple-Modells)

75 01 47 00 BD (Ausgang 2 eines Triple-Modells)

Die Antwort (Single-Modell oder Ausgang 1 eines Triple-Modells) könnte dann so aussehen:

85 00 47 01 01 64 00 1E 00 01 50 [1]

Das ergibt 42V (grün = Istwert Spannung) und 1,8A (blau = Istwert Strom) bei einem PS 2042-06B mit 42V Nennspannung und 6A Nennstrom. Der Status wird mit 0x0101 (pink) angegeben.

1 Siehe beiliegende Objektliste für die Zuordnung der Datenbytes in der Antwort

3. Kommunikation mit dem Gerät

3.1 Vorgehensweise

Generell gilt:

- Überwachung (Monitoring), also nur Abfrage von Istwerten und Status, ist immer möglich. Das Gerät benötigt dazu keinen Fernsteuerbetrieb
- Setzen von Zuständen und Sollwerten (Controlling) erfordert die vorherige Aktivierung des Fernsteuerbetriebes (remote)
- Bei Triple-Modellen sind die Hauptausgänge 1 und 2 fernsteuerbar und müssen, wie bei manueller Bedienung, separat behandelt werden

Um ein Gerät zu **steuern**, sprich z. B. einen Sollwert zu senden und zu setzen, müssen Sie

1. den Fernsteuerbetrieb aktivieren (Objekt 54)

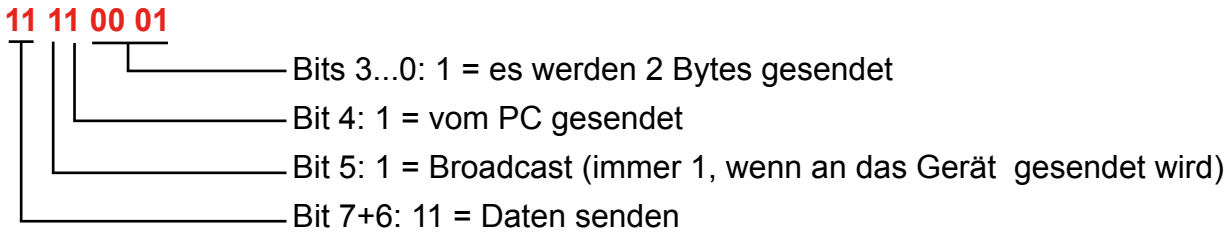
2. den Sollwert senden

3. den Ausgang einschalten (wenn nicht bereits „on“)

Der Fernsteuerbetrieb sollte verlassen werden, wenn er nicht mehr benötigt wird. Solange er aber aktiviert ist, kann das Gerät bzw. der jeweilige Ausgang, wenn es sich um ein Triple-Modelle handelt, nicht oder nur bedingt manuell bedient werden. Der Modus wird stets auf der Front des Gerätes angezeigt.

3.2 Startdelimiter erzeugen

Gemäß des Kommunikationsprotokolls (siehe oben) ist das erste Byte der Startdelimiter, der von der Richtung des Telegramms und dem Anfragetyp abhängig ist. Für dieses Beispiel nehmen wir einen SD von 0xF1. In Bits zerlegt sieht der SD dann so aus:



Es wird also etwas an das Gerät gesendet. Das Objekt und die Daten im Gerät bestimmen, was gesendet wird bzw. was das Gerät daraufhin macht.

Alternativ zum bitweisen Zusammensetzen des Startdelimiters kann man sich das vereinfachen, indem man Hexwerte addiert:

SD = Sendungstyp + Cast type + Richtung + Datenlänge

wobei Sendungstyp entweder

0xC0 Daten senden oder

0x40 Anfrage an das Gerät

0x80 Antwort vom Gerät

und Cast type

0x20 Broadcast (Senden/Anfrage an das Gerät)

0x00 Antwort vom Gerät

und Richtung entweder

0x10 vom PC ans Gerät oder

0x00 vom Gerät an den PC

und die Datenlänge - 1 von

0x00...0x0F bis zu 16 Bytes am Stück

Vom obigen Beispiel ausgehend ergibt sich der SD mit 0xF1 aus 0xC0 + 0x20 + 0x10 + 0x01.

3.3 Die Maske für Objekte mit Steuerbyte

Ein oder mehrere Objekte erfordern eine Maske, die immer zusammen mit dem eigentlichen Steuerbyte gesendet wird. Diese Maske ist in Spalte 6 der Objektliste angegeben (siehe seperate PDF-Datei oder „3.8. Objektliste“). Im Steuerbyte haben die einzelnen Bits unterschiedliche Funktionen, daher muß bestimmt werden, welches Bit geändert werden sollen. Dies bestimmt die Maske mit einer 1 für das zu ändernde Bit.

Beispiel: man möchte Bit 0 des Steuerbytes verändern, also 0 oder 1 setzen, dann wäre das Steuerbyte 0x00 oder 0x01 und die Maske 0x01, entsprechend der Wertigkeit der Bits. Siehe Objektliste, z. B. Objekt 54.

3.4 Beispiele

3.4.1 Gerät in Fernsteuerbetrieb umschalten

Das zu benutzende Objekt ist 54 (als Hexwert 0x36), die Maske für den Fernsteuerbetrieb (siehe auch Objektliste) ist 0x10 und das Steuerbyte für „Fernsteuerung ein“ ist 0x10. Somit ergibt sich dieses Telegramm:

F1 00 36 10 10 01 47 bzw. **F1 01 36 10 10 01 48**

Zum Umkehren des Ganzen, also der Deaktivierung, ist dann z. B. **F1 00 36 10 00 01 37** zu senden. Die Maske bleibt natürlich gleich, nur das Steuerbyte ändert sich.

Hinweis: bei den Triple-Modellen müssen die Ausgänge 1 und 2 separat in Fernsteuerung geschaltet werden!

3.4.2 Status abfragen

Der Status kann entweder mit Objekt 71 oder 72 abgefragt werden, wenn man die zus. mitgelieferten Ist- oder Sollwerte ignorieren möchte. Das Resultat des Status ist bei beiden Objekten gleich. Man sendet zuerst eine Anfrage (Objekt 71):

75 00 47 00 BC bzw. **75 01 47 00 BD**

und erhält dann eine Antwort, die so aussehen könnte:

65 00 47 00 05 17 36 64 00 01 62

Das erste Statusbyte 0x00 gibt an, daß das Gerät nicht in Fernsteuerbetrieb ist und das zweite gibt mit 0x05 laut Bitzuordnung an , daß der Ausgang ein und Strombegrenzung (CC) aktiv ist.

3.5 Mögliche Probleme beim Setzen von Zuständen

Mit dem Objekt 54 wird der Fernsteuerbetrieb (Remote) aktiviert/deaktiviert oder der Ausgang des Gerätes ein- bzw. ausgeschaltet. Die Verwendung des Objektes und der Maske lassen es zwar zu, daß im Steuerbyte beides gleichzeitig gesetzt/zurückgesetzt werden kann, dies ist aber nicht zu empfehlen!

Setzen des Eingangs/Ausgangs erfordert, daß bereits Remote aktiv ist und sollte daher nach der Aktivierung von Remote durch erneutes Senden des Objektes 54 geschehen. Natürlich nur mit dem entsprechenden Bit. Beim Beenden des Fernsteuerbetriebes umgekehrt genauso.

Wenn Sie beide Bits gleichzeitig setzen, kann es vorkommen, daß das Gerät zuerst den Ausgang/Eingang setzen will, bevor Remote aktiv ist und das wird mit einer Fehlermeldung quittiert. Daher ist es vielleicht sinnvoll, nach dem Setzen des Ausgangs/Eingangs dessen Status durch das Objekt 70 zurückzulesen.

3.6 Rückmeldungen

Beim Senden von Werten oder Zuständen schickt das Gerät eine Bestätigung (positve acknowledge), wenn der Befehl fehlerfrei ankam, in Ordnung war und ausgeführt wurde. Die Antwort kommt als Fehlermeldung (Objektnummer: 0xFF) und mit Fehlercode **0x00**. Siehe auch Fehlercodeliste unten.

Bei einem Telegrammfehler (Anwender hat falsches Telegramm gesendet) oder in Abhängigkeit vom Zustand des Gerätes kann bzw. wird eine Fehlermeldung zurückgegeben, die einen Code aus der Tabelle enthält:

Fehlercode			
Hex.	**Dez.**	**Beschreibung**	
00	0	Kein Fehler	
03	3	Prüfsumme nicht korrekt	
04	4	Startdelimiter falsch	
05	5	Falsche Adresse für Ausgang	
07	7	Objekt nicht definiert	
08	8	Objektlänge nicht korrekt	
09	9	Schreib-Leserechte verletzt, kein Zugriff	
0F	15	Gerät ist in "Lock" Modus	
30	48	Obere Grenze des Objektes überschritten	
31	49	Untere Grenze des Objektes unterschritten	

Legende
Kommunikationsfehler
Userfehler

Beispiel: wenn man z. B. mit Objekt 50 bei einem Gerät die Spannung setzen will und das Gerät nicht im Remote-Modus ist, dann würde sich das Fehlertelegramm **80 00 FF 0F 01 8E** ergeben. Der Fehlercode 0x0F besagt, daß das Gerät im „Local-Modus“ ist, also nicht in „Remote“.

Erläuterungen zu einigen Fehlercodes:

Code 0x7: die Objektnummer im Telegramm an das Gerät ist dem Gerät unbekannt.

Code 0x8: die Länge des Datenfeldes im Telegramm ist in der Objektliste definiert. Dieser Fehler kommt z. B., wenn ein Sollwert (immer 2 Bytes bei Typ „Int“) gesendet werden soll, das Datenfeld aber nur ein Byte enthielt. Selbst wenn der Startdelimiter die richtige Telegrammlänge enthält, dies dient zusätzlich zum Schutz, daß falsche Werte gesetzt werden.

Code 0x9: es wurde versucht, ein Objekt zu schreiben (=setzen), das laut Objektliste nur lesbar ist (Typ: read only (ro)).

3.7 Fehlerbehandlung

Problem: Es wurden mehrere Anfragen gestellt, aber nicht alle wurden beantwortet

Ursache: Die Anfragen wurden womöglich zu schnell nacheinander gestellt. Es ist zwischen zwei Befehlen eine gewisse Zeit zu warten. Für ein Gerät der Serie PS 2000 B ist ein zeitlicher Minimumabstand zwischen zwei Telegrammen von **50ms** festgelegt.

Problem: Sollwerte oder Status werden nicht gesetzt

Mögliche Ursachen:

- Das angesprochene Gerät befindet sich nicht im Fernsteuerbetrieb
- Die gesendeten Werte sind falsch, d.h. zu groß. Es wird dann eine Fehlermeldung gesendet. Oder der Wert wird akzeptiert, kann aber für den jeweiligen Zustand nicht umgesetzt werden (Spannungssollwert gesendet, während Gerät in Strombegrenzung ist)

3.8 Objektliste

PS 2000 B

1	2	3	4	5	6	7	8
Objekt / Object	Beschreibung / Description	Zugriff / Access	Datentyp / Data type	Datenlänge / Data length in Bytes	Maske bei Typ 'char' / Mask for type 'char'	Daten / Data	Beispiel oder weitere Erkärung / Example or further description
0	Gerätetyp / Device type	ro	string	16			PS2042-06B + EOL (EOL= End of Line 0x00)
1	Geräteseriennummer / Device serial no.	ro	string	16			1034440002 + EOL
2	Nennspannung / Nominal voltage	ro	float	4			Unenn / Unom = 42.0 (Fließkommazahl / Floating point number IEEE754 Standard)
3	Nennstrom / Nominal current	ro	float	4			Inenn / Inom = 6.0 (Fließkommazahl / Floating point number IEEE754 Standard)
4	Nennleistung / Nominal power	ro	float	4			Pnenn / Pnom = 100.0 (Fließkommazahl / Floating point number IEEE754 Standard)
6	Geräteartikelnummer / Device article no.	ro	string	16			39200112 + EOL
8	Hersteller / Manufacturer	ro	string	16			Herstellername / Manufacturer's name + EOL
9	Softwareversion / Software version	ro	string	16			V2.01 09.08.06 + EOL
19	Geräteklasse / Device class	ro	int	2			0x0010 = PS 2000 B Single, 0x0018 = PS 2000 B Triple
38	OVP Grenze / OVP threshold	rw	int	2			Überspannungssollwert (% von 1,1 * Unenn * 256) / Overvoltage set value (% of 1.1 * Unom * 256)
39	OCP Grenze / OCP threshold	rw	int	2			Überstromsollwert (% von 1,1 * Inenn * 256) / Overcurrent set value (% of 1.1 * Inom * 256)
50	Sollwert U / Set value U	rw	int	2			Spannungssollwert (% von Unenn * 256) / Set value of voltage (% of Unom * 256)
51	Sollwert I / Set value I	rw	int	2			Stromsollwert (% von Inenn * 256) / Set value of current (% of Inom * 256)
54	Steuerung des Netzteils / Power supply control	rw	char	2	0x01 0x01 0x0A 0x10 0x10 0xF0 0xF0	0x01 0x00 0x0A 0x10 0x00 0xF0 0xE0	Leistungsausgang ein / Switch power output on Leistungsausgang aus / Switch power output off Alarme quittieren / Acknowledge alarms Umschalten in Fernsteuerbetrieb / Switch to remote control Umschalten in Handbedienung / Switch to manual control Tracking ein / Tracking on Tracking aus / Tracking off
71	Status + Istwerte / Status + Actual values	ro	int	6		Byte 0: Bits 1+0: Byte 1: Bit 0: Bits 2+1: Bit 3: Bit 4: Bit 5: Bit 6: Bit 7: Word 1: Word 2:	Gerätezustand abfragen / Query device state 00 = freier Zugriff / free access; 01=Remote 1 = Ausgang eingeschaltet / Output on Reglerstatus / Controller state: 00=CV; 10= CC 1 = Tracking aktiv / Tracking active ** 1 = OVP aktiv / OVP active 1 = OCP aktiv / OCP active 1 = OPP aktiv / OPP active 1 = OTP aktiv / OTP active Spannungsistwert (% von Unenn * 256) / Actual voltage (% of Unom * 256) Stromistwert (% von Inenn * 256) / Actual current (% of Inom * 256)
72	Status + Aktuelle Sollwerte / Status + Momentary set values	ro	int	6		Byte 0: Bits 1+0: Byte 1: Bit 0: Bits 2+1: Bit 3: Bit 4: Bit 5: Bit 6: Bit 7: Word 1: Word 2:	Gerätezustand abfragen / Query device state 00 = freier Zugriff / free access; 01=Remote 1 = Ausgang eingeschaltet / Output on Reglerstatus / Controller state: 00=CV; 10= CC 1 = Tracking aktiv / Tracking active ** 1 = OVP aktiv / OVP active 1 = OCP aktiv / OCP active 1 = OPP aktiv / OPP active 1 = OTP aktiv / OTP active Spannungssollwert (% von Unenn * 256) / Set value of voltage (% of Unom * 256) Stromsollwert (% von Inenn * 256) / Set value of current (% of Inom * 256)

Legende / Legend:
ro = Nur lesen / Read only
rw = Schreiben und Lesen / Read and write
int = 16 bit Wert / value
char = 8 bit Wert / value
float = 32 bit Fließkommazahl / Floating point number
string = Zeichenkette mit 0x00 am Ende / String with 0x00 at the end

** Nur PS 2000 B Triple / PS 2000 B Triple only

PS 2000 B Programming Guide

Date: 06-06-2016

Table of contents

1. Preamble 15
1.1 Introduction 15
1.2 Driver installation 15
1.2.1 Trouble-shooting driver installation problems 16
1.3 Terms 17
2. The communication protocol 17
2.1 Structure of the communication 17
2.2 Serial communication parameters 17
2.3 Translating set values and actual values 18
2.3.1 Actual values 18
2.3.2 Set values 18
2.4 Telegram structure 18
2.4.1 Example telegram 19
3. Communicating with the device 20
3.1 General 20
3.2 Creating a start delimiter 20
3.3 The mask for objects with a control byte 21
3.4 Examples 21
3.4.1 Switch device to remote mode 21
3.4.2 Query status 21
3.5 Possible problems when setting device conditions 22
3.6 Acknowledging and error messages 22
3.7 Trouble-shooting 23
3.8 Object list 23

1. Preamble

1.1 Introduction

This guide was written to explain the binary communication protocol for devices of series PS 2000 B to the user. The device is accessed via the built-in USB port. The USB driver will create a virtual COM port (VCP) for any new unit of this device type. Using the COM port will reduce programming of the communication port and its access to a minimum, because a COM port is normally just opened, configured, used and closed afterwards. Data is always sent to the open port.

Note: this guide does not instruct about programming and access of real or virtual hardware ports in the various programming languages. Information about this can be found in books or the internet.

1.2 Driver installation

Connect the unit to the PC via the included USB cable. Take care of fitting the mini USB plug correctly and tightly into the socket on the front (might require some force to push it in completely). If the cable is too short for the final placement of the unit, it can be extended by another USB cable (max. 5m) or an USB hub. Windows installs the driver once for every PS 2000 B that is new to the system. In case the driver is not already present on the system, it has to be installed once.

Windows should try to find and install the driver like this:

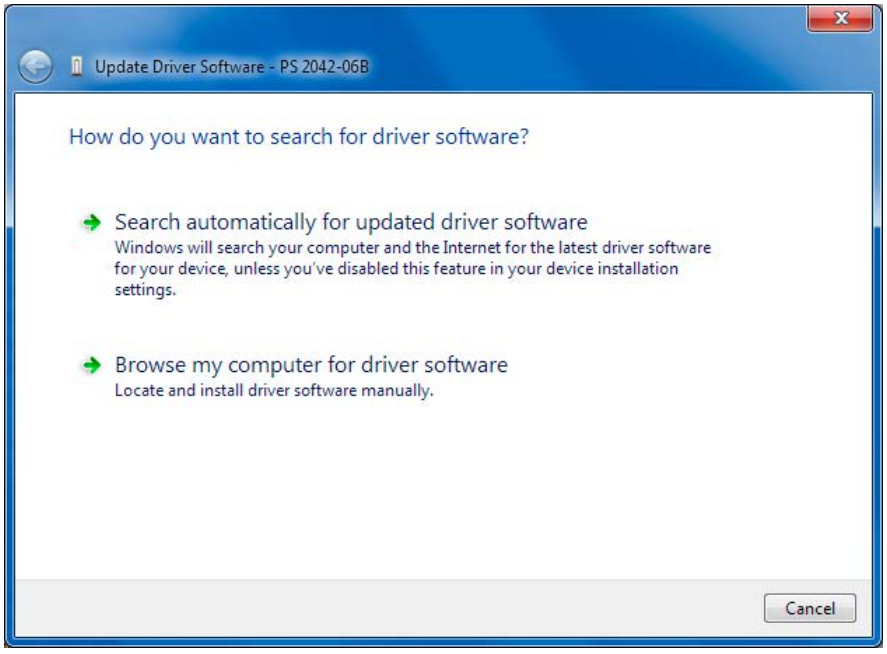
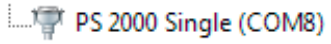


Figure 1

Always select the lowermost option here (Windows XP/Vista/7). In the next dialogue windows you are requested to point to the driver, which is located on the included CD in folder \driver. The correct installation of driver and device can be verified by opening the device manager and looking for



or PS 2000 Triple (COM8), depending on the model,

in section „Ports (COM & LPT)“. COM8 is just an example. If multiple units are connected to the PC, there should be multiple entries here.

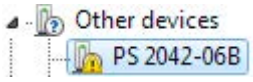
1.2.1 Trouble-shooting driver installation problems

When connecting a new device to the PC and if the driver is not already installed on the system, Windows should ask for a driver. In case it does not, the failed driver installation is reported by Windows. In this case you need to open the Windows device manager:

XP: Start (or desktop) → Right-click „Computer“ → Properties → Tab „Hardware“ → Button „Device manager“

Vista / Windows 7: Start (or desktop) → Right-click „Computer“ → Properties → Link „Device manager“ or type „device manager“ in the search bar and select it from the result list

In the section „Other devices“ you should see the device to install, like for example:



Then right-click the entry and select „Update Driver Software“. The well-known Windows driver installation dialogue will appear. Select the lowermost option (example is from Windows 7, Vista and XP are similar):

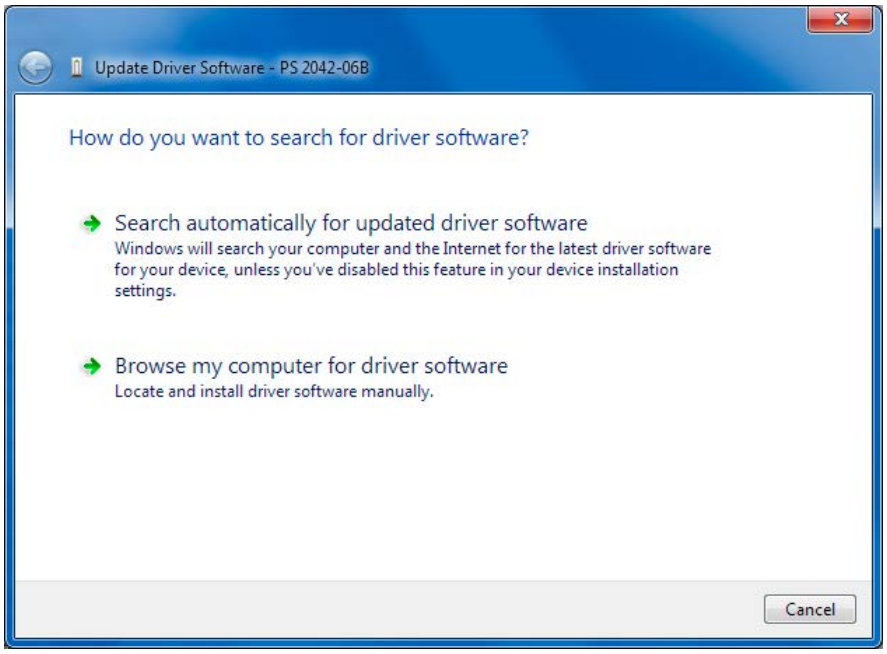
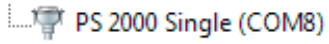


Figure 2

In the next window select the path to the driver on the EasyPS2000 CD:

Figure 3

Windows will then install the device. After a successful installation the device can be found as „PS2000 Single“ or „Triple“ in the device manager, in section „Ports (COM & LPT)“.



Note: The COM port number 8 above is only an example. Windows assigns a new COM port for every new device of this type that is installed in the system. The port is remembered and used again if the device is connected the next time.

Note: In case there are multiple units connected to the PC, they will all be listed as single devices in the device manager.

1.3 Terms

Telegram = Chain of hexadecimal bytes, with varying length. It is either sent to a device or received from it. A part of the telegram (data field) represents hexadecimal values or ASCII strings.

2. The communication protocol

2.1 Structure of the communication

The communication with the unit is based on these telegram types:

a) Message: an object is sent which shall, for instance, set the output voltage. As long as this action is permitted by the current state of the device, the object is accepted and executed. The device will send an answer in form of an error message, containing the error code 0. If the execution of the command is not possible due to the current state of the device or the telegram is wrong, an error message with error code other than 0 is returned.

b) Query: a query is sent to the device and an answer is expected. If the query is permitted for the current state of the device it is executed and answered. The answer contains the requested data. If not permitted it will send an error message as answer.

2.2 Serial communication parameters

Data transfer is done via a virtual COM port (VCP), which is generated by the USB driver.

With the serial transmission of one byte following bits are sent:

Start bit + 8 Data bits + Parity bit + Stop bit

The parity is checked for „odd“.

Set these parameters for the particular port on the PC:

Baud rate:	115200 Bd
Parity:	odd
Stop bits:	1

On the device, serial settings can not be altered.

2.3 Translating set values and actual values

Set values and actual values are transmitted as percentage values. A value of 0x6400 corresponds to 100.00%. If a device has a nominal voltage of 42V and the received actual value is 0x3200 = 50.00%, then it corresponds to 21V output voltage.

The high byte is the percentage number (0x64 = decimal 100) and the low byte is the decimal place of it, whereas 0x500 would be 5%, for example, but 0x550 would not be 5.5%. You need to translate the outgoing set values and the incoming actual values before they can be used.

2.3.1 Actual values

Actual values are queried and read from the device and will be returned as hexadecimal 16 bit values, representing a per cent value.

$$\text{Real actual value} = \frac{\text{Nom. value} * \text{Per cent act. value}}{25600}$$

Example: The nominal value of the device is 42V, the percentage actual value came in as 0x2454 = 9300. It results in: Actual value = 42 * 9300 / 25600 = 15.26V.

2.3.2 Set values

Set values have to be translated into 16 bit per cent values before transmission. Reading set values back from the device requires to translate them vice versa.

$$\text{Percentage set value} = \frac{25600 * \text{Real set value}}{\text{Nom. value of the device}}$$

Example: the set value for voltage shall be 25.5V, the nom. value of the device is 42V. With the formula it results in: Percentage set value = 25600 * 25.5 / 42 = 15543 = 0x3CB7. After sending 0x3CB7 to the 42V model, it should set 25.5V output voltage.

2.4 Telegram structure

The telegram is structured like this:

Byte 0	Byte 1	Byte 2	Variable bytes	2 bytes
SD	**DN**	**OBJ**	**DATA**	**CS**

and is built of these bytes:

Byte 0: **SD** (start delimiter)

The start delimiter determines how to handle the telegram furthermore. Meaning of the bits:

Bits 0-3: Data length -1 of the data in the data field of the telegram (bytes 3-18)

The data length - 1 of the data in the telegram when sending. At a query, the data length of the expected data is given here.

Bit 4: Direction

0= Telegram from device to control unit

1= Telegram from control unit to device

Bit 5

1= Cast type, for sending/querying it must be set to 1, in answers it will be 0

Bits 6+7: Transmission type

00 = Reserved

01 = Query data

10 = Answer to a query

11 = Send data

Byte 1: **DN** (device node)

Here you need to distinguish whether a Single or Triple model of PS 2000 B series is going to be controlled. With a Single model (one DC output), the DN is ignored and can be any value, but recommended to be left zero. With a Triple model (three DC outputs), following applies:

- a Triple model of PS 2000 B always has two device nodes (DN): 0 and 1

- device node 0 is assigned to DC output 1 and device node 1 is assigned to DC output 2

The control of both output regarding the objects is identical, they are only distinguished by this byte.

Byte 2: **OBJ**

The communication objects for a device are addressed by this byte. In the communication object lists, the objects and their function(s) are explained in detail. See below in section 3.8.

Byte 3 - 18: **Data field**

The data field can be 0-16 bytes long, hence the length of the telegram varies. If a query is sent (PC -> device) and no data is sent, the data field is not used and the checksum of the telegram follows directly after byte 2. Data are only transmitted when sending something to the device or when receiving an answer from it.

Word x: **CS** (check sum)

The check sum is always located at the end of the telegram. It is built by the simple addition of all preceding bytes of the telegram. It is two bytes long. The high byte is placed before the low byte.

2.4.1 Example telegram

Actual values shall be queried from the device. According to the object list, this can be done with object 71. The telegram to query the actual values has to look like this:

75 00 47 00 BC (Single model or output 1 of a triple model)

75 01 47 00 BD (Output 2 of a triple model)

The answer from the device could be like this (single model or output 1 of a triple model):

85 00 47 01 01 64 00 1E 00 01 50

This will translate to 42V (green = actual voltage) and 1.8A (blue = actual current) with a PS 2042-06B with 42V nominal voltage and 6A nominal current. The status (see description of object 71) is returned as 0x0101 (pink).

3. Communicating with the device

3.1 General

It applies:

- Monitoring, i.e. only querying actual values and status, is always possible. The device doesn‘t require to be in remote mode in this case
- Setting of status and set values (controlling) requires the activation of the remote control mode
- With the triple models, outputs 1 and 2 are remotely controllable and have to be operated seperately

In order to start controlling a device, for example by sending a set value, you need to

1. activate the remote mode (object 54)

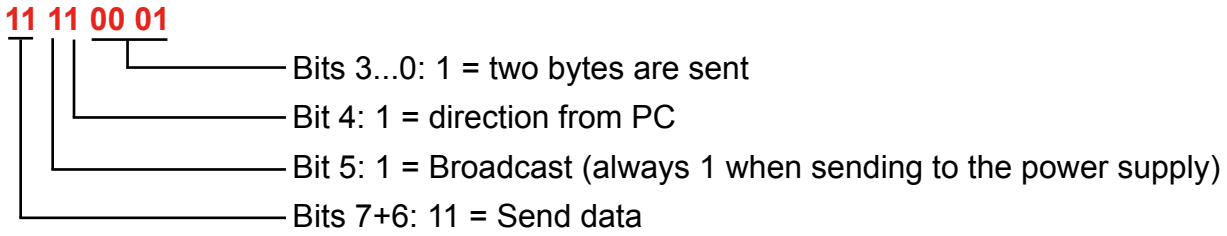
2. send the set value

3. set the output/input to on (if not already on)

Remote control should be left again, if not used any further. As long as it is active, the device can not be operated manually. The mode is indicated on the front display.

3.2 Creating a start delimiter

According to the telegram format (see above), the first byte of a telegram is the start delimiter, which depends on the type and direction of the telegram. For example, the SD can be 0xF1 and looks like this in single bits:



The SD determines that data are sent to the device. The content of the data and the object define what is sent and what the device will do in reaction.

Alternatively to the bitwise assembly, this can be simplified by adding hex values:

SD = Message type + Cast type + Direction + Length

whereas the message type is either

0xC0 Send data or

0x40 Query data

0x80 Answer from device

and the cast type is either

0x00 Answer from device (single cast) or

0x20 Broadcast (sending/querying data)

and the direction is either

0x10 from PC to the device or

0x00 from device to the PC

and the data length - 1 can be

0x01...0x0F up to 16 bytes of data

By the above example, the SD of 0xF1 is built from 0xC0 + 0x20 + 0x10 + 0x01.

3.3 The mask for objects with a control byte

Some objects require a mask, which is sent together with the control byte. The possible mask value is given in column 6 of the object list (see seperate PDF file or section „3.8. Object list“). The bits of the control byte have various functions, so it has to be determined which bit has to be changed. This is determined by the mask with a 1 for the corresponding bit of the control byte.

Example: bit 0 of the control byte shall be changed to 0 or 1. This will result in the control byte being either 0x00 or 0x01 and the mask being 0x01. Also see the object list, for example object 54.

3.4 Examples

3.4.1 Switch device to remote mode

The object to use is 54 (in hexadecimal 0x36), the mask for the remote mode (also see object lists) is 0x10 and the control byte for remote mode is 0x10. The telegram results as this:

F1 00 36 10 10 01 47 resp. **F1 01 36 10 10 01 48**

In order to reverse this command, deactivation of the remote mode that is, you need to send

F1 00 36 10 00 01 37. The mask stays the same, only the control byte changes.

Note: with the triple models, output 1 and 2 have to be seperately set into remote control!

3.4.2 Query status

The status can either be queried with object 71 or object 72, if the other returned values (actual values or set values) can be ignored. The returned status is identical with both objects. First, you need to send the query (object 71):

75 00 47 00 BC resp. **75 01 47 00 BD**

and the device will return an answer like this:

65 00 47 00 05 17 36 64 00 01 62

The first status byte 0x00 states, that the device is not in remote mode, i.e. local mode, and the second status byte states with 0x05, that the output is on and constant current mode (CC) active.

3.5 Possible problems when setting device conditions

The object 54 is used to either activate/deactive the remote control operation or the input/output of a device. The object can be used to activate both at once, but it is strongly not recommended to do so, because setting the input/output requires the remote control operation already being active and else would generate an error message. Best way is to activate remote control first with the corresponding bit set in the control byte and then control the input/output by sending object 54 a second time with a different control byte. When deactivating remote control it simply goes vice versa.

It is also useful to read back the state of the device with object 70, in order to check if object 54 has been set correctly.

3.6 Acknowledging and error messages

When sending values or condition, the device will return an acknowledging message in form of an error message (object: 0xFF) containing error code **0**. This indicates that the last command was received and executed correctly. Otherwise, if the device cannot execute the last command by some reason or the telegram was wrong, an error message containing an error code other than 0 is returned. See following table:

Error code			
Hex.	**Dec.**	**Description**	
00	0	No error	
03	3	Check sum incorrect	
04	4	Start delimiter incorrect	
05	5	Wrong address for output	
07	7	Object not defined	
08	8	Object length incorrect	
09	9	Read/Write permissions violated, no access	
0F	15	Device is in "Lock" state	
30	48	Upper limit of object exceeded	
31	49	Lower limit of object exceeded	

Legend

Communication error

User error

Example: if you try to set the output voltage of the device while it is not in remote control, the device returns the error message **80 00 FF 0F 01 8E**. The error code 0x0F indicates, that the device is in „local mode“, means not in remote control and won‘t thus accept the set value.

Explanation of some error codes:

Code 0x7: the object number used in the telegram is unknown to the device.

Code 0x8: the length of the data field in the telegram is defined in the object list. This error code will be returned if a set value, which is always 2 bytes because of type „int“, should have been sent but the data field only contained one byte. Even if the start delimiter contained the correct telegram length. This is a protection against setting wrong values.

Code 0x9: the user tried to write to an object which is, according to the object list, read only (ro).

3.7 Trouble-shooting

Problem: Multiple queries were sent, but not all of them were answered

Cause: The queries have been sent too fast. Depending on the execution time of the device, you need to include a certain latency between two transmissions. The minimum time between the transmission of two commands is defined at **50ms** for a device of PS 2000 B series.

Problem: Set values and status are not set

Possible causes

- The contacted device is not in remote control mode
- The sent values are wrong (too high, too low). An error message is returned. Or the value is accepted, but can not be transferred to the output because of the current device condition, like when sending a voltage set value while the device is in current limitation.

3.8 Object list

See external PDF file named „object_list_ps2000b_de_en.pdf“ (german/english) or „object_list_ps2000b_cn_en.pdf“ (chinese/english) for the most recent list of available communication objects. These two PDFs should always be along with this document.