

---

Subject: USB I/O-Kommunikation mit DMM M-3890DT (Fa. Conrad) ?

Posted by [Bernd Furnier](#) on Mon, 03 Jul 2006 09:34:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hallo LabVIEW-Freunde,

für unser Labor möchten wir ein VISA bauen, um verschiedene, vom Voltkraft (Conrad) DMM M-3890DT erfasste, Messwerte in LabVIEW nutzen zu können.

Von Fa. Conrad habe ich eine Übersicht vom USB-Protokoll des M-3890DT bekommen, siehe P.P.S.

Da ich noch keine Kommunikationsroutinen für USB erstellt habe, weiss ich nicht, in welcher Form (Bit-, Byte-Ebenen, CRC, Start- Stop-Bit??) ich die Daten zusammenbauen muss, um mit dem M-3890DT über USB zu kommunizieren.

Toll wäre, wenn mit jemand von Euch die notwendigen Schritte zur Lösung der beschriebenen Aufgabe vermitteln könnte, oder schon das passende VI (VISA) in der Schublade hat, damit ich die Schritte nachvollziehen kann.

Die Vendor-ID meines DMM M-3890DT ist: 0x0925, die Produkt-ID ist: 0x1234

Danke schon mal im Voraus ,

Bernd

P.S.: Grundlagen zur LabVIEW-Programmierung kann man sich im - auf meiner homepage <http://www.fu-net.de> verlinkten - Tutorium ansehen.

P.P.S.: Die Bytes für das USB-Protokoll lauten gemäß der Mitteilung von Fa. Conrad:

1. Byte: Bit 0 (Vorzeichen): 0 -> plus  
1 -> minus

Bit 1, 2 (Dezimalpunkt (DP)):

0 0 -> NODP

0 1 -> LSD

1 0 -> MDP

1 1 -> MSD

Bit 5, 4, 3 (Modell):

0 0 0 -> M-3890D

Bit 6, 7 (Kanal):

0 0 -> MAIN

0 1 -> Sub1

1 0 -> Sub2

- - -> not used

2. Byte: Bit 7, 6, 5, 4 (Funktion)

0 0 0 0 -> DC V 0

0 0 0 1 -> AC V 1

0 0 1 0 -> Widerstand 2

0 0 1 1 -> DC uA 3

0 1 0 0 -> DC mA 4

0 1 0 1 -> DC A 5

0 1 1 0 -> AC uA 6

0 1 1 1 -> AC mA 7

1 0 0 0 -> AC A 8

1 0 0 1 -> Frequenz 9  
 1 0 1 0 -> Kapazität 10  
 1 0 1 1 -> Signal ausg. 11  
 - - - -  
 1 1 1 0 -> etc 14

Bit 3, 2, 1, 0 (Bereich DC V oder AC V):

0 0 0 0 -> mv 0  
 0 0 0 1 -> V 1

Bit 3, 2, 1, 0 (Bereich Widerstand):

0 0 0 0 -> Ohm 0  
 0 0 0 1 -> kOhm 1  
 0 0 1 0 -> MOhm 2

Bit 3, 2, 1, 0 (Bereich DC uA oder AC uA):

0 0 0 0 -> uA 0  
 0 0 0 1 -> mA 1

Bit 3, 2, 1, 0 (Bereich DC mA oder AC mA):

0 0 0 0 -> mA 0

Bit 3, 2, 1, 0 (Bereich DC A oder AC A):

0 0 0 0 -> A 0

Bit 3, 2, 1, 0 (Frequenz):

0 0 0 0 -> kHz 0  
 0 0 0 1 -> mHz 1

Bit 3, 2, 1, 0 (Kapazität):

0 0 0 0 -> nF 0  
 0 0 0 1 -> uF 1

Bit 3, 2, 1, 0 (etc):

0 0 0 0 -> Durchgang 0  
 0 0 0 1 -> Diode 1  
 0 0 1 0 -> hFE 2  
 0 0 1 1 -> Temperatur 3  
 0 1 0 0 -> Logik 4  
 0 1 0 1 -> EF 5  
 0 1 1 0 -> dB 6

3. Byte und 4. Byte (MAIN):

Byte 3: Bit 7, 6, 5, 4 -> first (1st) digit  
           Bit 3, 2, 1, 0 -> second (2nd) digit  
 Byte 4: Bit 7, 6, 5, 4 -> third (3rd) digit  
           Bit 3, 2, 1, 0 -> forth (4th) digit

digit 1, 2, 3, 4 (<10): 0 bis 9 <- Zahlenwert  
 digit 2, 3 = "rd" <- Logic range "rdy"  
 digit 2, 3 = "Lo" <- Logic range "Lo"

digit 2, 3 = "Hi" <- Logic range "Hi"  
digit 4 = "-" <- Logic range "-----"

#### 5. Byte (Sub1):

Bit 0: 0 -> plus  
1 -> minus

Bit 2, 1 (DP):

0 0 -> NODP  
0 1 -> LSD  
1 0 -> MDP  
1 1 -> MSD

Bit 5, 4, 3 (Model):

0 0 0 -> M3890D

Bit 7, 6 (Kanal):

0 0 -> Main  
0 1 -> Sub1  
1 0 -> Sub2  
- - not used

#### 6. Byte (Sub1):

Bit 7, 6, 5, 4 (Funktion): Display unit

0 0 0 0 -> DC V no unit  
0 0 0 1 -> AC V dB  
0 0 1 0 -> Widerstand no unit  
0 0 1 1 -> DC uA no unit  
0 1 0 0 -> DC mA no unit  
0 1 0 1 -> DC A no unit  
0 1 1 0 -> AC uA no unit  
0 1 1 1 -> AC mA no unit  
1 0 0 0 -> AC A no unit  
1 0 0 1 -> Frequenz no unit  
1 0 1 0 -> Kapazität no unit  
1 0 1 1 -> Signal ausg. no unit  
- - - - -  
1 1 1 0 -> etc no unit

#### 7. Byte und 8. Byte (Sub 1):

Byte 7: Bit 7, 6, 5, 4 -> first (1st) digit

Bit 3, 2, 1, 0 -> second (2nd) digit

Byte 8: Bit 7, 6, 5, 4 -> third (3rd) digit

Bit 3, 2, 1, 0 -> forth (4th) digit

digit 1, 2, 3, 4 (<10): 0 bis 9 <- Zahlenwert

digit 1, 2, 3, 4 (<10)= " ": no display

digit 1, 2, 3, 4 (<10)= "OL": Overflow

#### 9. Byte (Sub 2):

Bit 0: 0 -> plus

1 -> minus

Bit 1, 2 (Dezimalpunkt (DP)):

0 0 -> NODP

0 1 -> LSD

1 0 -> MDP

1 1 -> MSD

Bit 5, 4, 3 (Modell):

0 0 0 -> M-3890D

Bit 6, 7 (Kanal):

0 0 -> MAIN

0 1 -> Sub1

1 0 -> Sub2

- - -> not used

10. Byte (Sub 2):

Bit 7, 6, 5, 4 (Funktion): Display unit

0 0 0 0 -> DC V no unit

0 0 0 1 -> AC V dB

0 0 1 0 -> Widerstand no unit

0 0 1 1 -> DC uA no unit

0 1 0 0 -> DC mA no unit

0 1 0 1 -> DC A no unit

0 1 1 0 -> AC uA no unit

0 1 1 1 -> AC mA no unit

1 0 0 0 -> AC A no unit

1 0 0 1 -> Frequenz no unit

1 0 1 0 -> Kapazität no unit

1 0 1 1 -> Signal ausg. no unit

- - - - -

1 1 1 0 -> etc no unit

Bit 3, 2, 1, 0 (Range): Display Unit

0 0 1 1 -> Temperatur F

0 1 0 0 -> Logik V

1 0 1 1 -> Signal Out V

---

Subject: Re: USB I/O-Kommunikation mit DMM M-3890DT (Fa. Conrad) ?

Posted by [Brand](#) on Tue, 04 Jul 2006 07:29:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hallo Bernd,

bei NI gibt es eine Anleitung, wie man an die Entwicklung herangeht:

<http://zone.ni.com/devzone/conceptd.nsf/webmain/ADF3152837E2B4A486256B5600642AC7>

Ich hatte es selbst vor 1,5 Jahren mal probiert, bin aber gescheitert, weil der Instrumentenhersteller die Details des Kommunikationsprotokolls nicht herausgerückt hat.

Wenn ich meine ersten Schritte wiederfinde, werde ich Sie Dir hier einstellen.

Es gibt weiter Hinweise in diesem Forum. Benutze doch die Such-Funktion, um sie zu finden.

Gruß Holger

---

---

Subject: Re: USB I/O-Kommunikation mit DMM M-3890DT (Fa. Conrad) ?

Posted by [Manfred Leffler](#) on Tue, 04 Jul 2006 13:33:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Bernd,

Holger hat schon einen guten Link gepostet. Eventuell ist folgender Link auch hilfreich

<http://zone.ni.com/devzone/conceptd.nsf/webmain/6792BAB18242082786256DD7>

006B6416?opendocument

Gruss

Manfred

---