PeakTech 2025

Communication protocol

Connection settings:

USB	
Supported:	**Yes**
Vendor ID:	0x2571
Product ID:	0x4100
Mode:	RAW HID
Frame size:	8 bytes
Comments:	Dependent on USB implementation you may receive 9 bytes. First byte (always 0x00) needs to be ignored in this case.

RS232	
Supported:	**Partial** *(see comments)*
Baud:	2400
Parity:	None
Data / Stop:	8 / 1 Bit(s)
Frame size:	14 Bytes
Comments:	Device uses RS232 internal with a different frame format. There are 2 board revisions, one operates as USB HID device (driverless), one as virtual COM port. The data packet format differs for each revision.

Frame content *(USB HID)*:

Byte 0	Byte 1..2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Sign+Dec	7 seg. display *(as BCD)*	SB1	SB2	SB3	SB4	Bar graph

Frame content *(RS232)*:

Byte 0	Byte 1..4	Byte 5
Sign	Digit 4..0	0x20

...

...

Byte 6	Byte 7..10	Byte 11	Byte 12..13
Point	SB1...SB4 (as above)	Bar graph	End mark (CrLf)

0x0D 0x0A

Byte 0: *(USB HID - Sign & Decimal position)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3..0
1	Negative	Positive	1	Decimal position (0..4)

0=xxxx, 1=x.xxx, 2=xx.xx, 3+4=xxx.x

Byte 0: *(RS232 - Sign)*

0x2B for positive, 0x2D for negative values

Byte 1..2: *(USB HID - 7 segment display numbers)*

Bit 7..4	Bit 3..0
Thousands (Byte 1) / Tenths (Byte 2) *(as BCD)*	Hundreds (Byte 1) / Ones (Byte 2) *(as BCD)*

Byte 1..4: *(RS232 - 7 segment display numbers)*

Digits, transferred as ASCII, so 0x30=0, 0x31=1, … one digit each byte.

Byte 6 *(RS232 only)*: *(Decimal position)*

0=xxxx, 1=x.xxx, 2=xx.xx, 3+4=xxx.x

Byte 3 *(USB HID)* / Byte 7 *(RS232)*: *(Status Byte 1)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	AUTO	DC	AC	REL	HOLD	BPN

Byte 4 *(USB HID)* **/ Byte 8** *(RS232)* **:** *(Status Byte 2)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 *(Z1)*	0 *(Z2)*	MAX	MIN	APO	BATT	n	0 *(Z3)*

Byte 5 *(USB HID)* **/ Byte 9** *(RS232)* **:** *(Status Byte 3)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
µ	m	k	M	Continuity	Diode	%	0 *(Z4)*

Byte 6 *(USB HID)* **/ Byte 10** *(RS232)* **:** *(Status Byte 4)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
V	A	Ω	hFE	Hz	F	°C	°F

Byte 7 *(USB HID)* **/ Byte 11** *(RS232)* **:** *(Bar graph)*

Bit 7	Bit 6..0
Negative	Bargraph value (0..60, >60=OL sign)

PeakTech 3315

Communication protocol

Connection settings:

USB	
Supported:	**Yes**
Vendor ID:	0x1A86
Product ID:	0xE008
Mode:	HID
Frame size:	11 Bytes*
Comments:	see USB notes

RS232	
Supported:	**Yes**
Baud:	2400
Parity:	Odd
Data / Stop:	7 / 1 Bit(s)
Frame size:	11 Bytes
Comments:	Except for the end bytes (CrLf) each byte starts with the upper nibble 0x_011. Each frame will be send twice.

USB Notes:

The USB implementation is a bit complicated.
First of all the USB cable needs to be initialized. For this you need to send this 6 bytes to the device:
0x00 (= Report ID), 0x60, 0x09, 0x00, 0x00, 0x03.
This will initialize the cable with 2400,8,N,1, which is incorrect, but the only way it works. The software needs to clear the MSB of each received byte (it's the parity bit).
After that you can receive data, but it is not the data as you receive using RS232. To get this data we need to extract it out of a USB data frame. After that you got the data as with RS232 (also twice).
Dependent on USB implementation you may receive 9 bytes. First byte (always 0x00) needs to be ignored in this case. Same for sending the init command where you may have to remove the first byte.

USB data frame:

Byte 0	Byte 1	Byte 2..7
Action	Data	Unknown / not used

Action = 0xF0 --> Dummy packet / keep alive. Do nothing.
Action = 0xF1 --> 1 byte received (Data). Remember to set MSB to 0!

Frame content:

Byte 0	Byte 1..4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9..10
Range	Digit 3..0	Function	Status	Option 1	Option 2	End mark (CrLf)

0x0D 0x0A

Byte 0: *(Range)*

Value	V	mA	µA	Ω	Frequency	RPM
0110000 (0x30)	xxx.x mV	xx.xx mA	xxx.x µA	xxx.x Ω	x.xxx kHz	xx.xx kRPM
0110001 (0x31)	x.xxx V	xxx.x mA	xxxx µA	x.xxx kΩ	xx.xx kHz	xxx.x kRPM
0110010 (0x32)	xx.xx V			xx.xx kΩ	xxx.x kHz	x.xxx MRPM
0110011 (0x33)	xxx.x V			xxx.x kΩ	x.xxx MHz	xx.xx MRPM
0110100 (0x34)	xxxx V			x.xxx MΩ	xx.xx MHz	xxx.x MRPM
0110101 (0x35)				xx.xx MΩ		

Continuity, Diode and Ampere modes are fixed, so Range Value is always 0110000 (0x30).

Byte 1..4: *(Digits)*

Upper nibble is always 0x_011, lower nibble is the value (0..9).
Note that this multimeter does not display left-handed zeros.

Byte 5: *(Function)*

Value	Function
0110001 (0x31)	Diode
0110010 (0x32)	Frequency / RPM *
0110011 (0x33)	Ω
0110100 (0x34)	Temperature *
0110101 (0x35)	Continuity
0111000 (0x38)	ADP2
0111001 (0x39)	Current (mA)

Value	Function
0111010 (0x3A)	ADP3
0111011 (0x3B)	Voltage
0111100 (0x3C)	ADP1
0111101 (0x3D)	Current (µA)
0111110 (0x3E)	ADP0
0111111 (0x3F)	Current (A)

*) Check Judge bit in Status byte to determinate Frequency / RPM and °C / °F.

Byte 6: *(Status)*

7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	Judge	Sign	Batt	OL

Byte 7: *(Option 1)*

7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	0	0	0	V A Hz

V A Hz: See Range byte to determinate the current mode

Byte 8: *(Option 2)*

7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	DC	AC	Auto	APO

PeakTech 3415

Communication protocol

Connection settings:

USB	
Supported:	**No** *(uses virtual com port)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	2400
Parity:	None
Data / Stop:	8 / 1
Frame size:	15 Bytes
Comments:	Sometimes this device interrupts data frames on user action (turns the rotary switch, press hold, …), so you always need to check the upper nibble! Also the device stops sending data frames if in hold state.

Frame content:

The upper nibble is the byte number (beginning with 0x1_) and is thus ignored in this documentation. It should be used to detect the start of the data frame and for missing bytes. The real data is located in the lower 4 bytes. The transferred data is basically the VRAM of the multimeter.

Byte 0	Byte 1..8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
AC DC Auto	Digits	Diode K n µ	Cont. M % m	Hold Rel Ω F	Batt Hz V A	ADP °C °F	Max Min APO

Byte 0: *(AC DC Auto)*

Bit 3	Bit 2	Bit 1	Bit 0
RS232	Auto	DC	AC

Byte 1..8: *(Digits)*

To get the digit 2 Bytes are required. Byte 1 and 2 for thousends, 3 and 4 for hundreds, …
7 bits represent the 7 digits (a at the top, b the top right, … clockwise. g for the middle. The remaining bit is for the decimal point, except for the thousends digit where it represents the negative sign.
If Byte 1 is 0x20 and Byte 2 is 0x30 (all 4 data bits zero) the multimeter displays "OL".

	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1, 3, 5, 7	a	f	e	Negative / DP
Byte 2, 4, 6, 8	b	g	c	d

Byte 9: *(Diode, K, n, µ)*

Bit 3	Bit 2	Bit 1	Bit 0
Diode	K	n	µ

Byte 10: *(Continuity, M, %, m)*

Bit 3	Bit 2	Bit 1	Bit 0
Continuity	M	%	m

Byte 11: *(Hold, Relative, Ohm, F)*

Bit 3	Bit 2	Bit 1	Bit 0
Hold	Relative	Ohm	F

Byte 12: *(Battery, Hz, V, A)*

Bit 3	Bit 2	Bit 1	Bit 0
Battery	Hz	V	A

Byte 13: *(MAX, °C, MIN)*

Bit 3	Bit 2	Bit 1	Bit 0
ADP2 (always 0)	ADP1 (always 0)	°C	°F

Byte 14: *(Max Min APO)*

Bit 3	Bit 2	Bit 1	Bit 0
MAX	Substract (MINMAX)	MIN	APO

PeakTech 3430

Communication protocol

Connection settings:

USB	
Supported:	**No** *(uses virtual COM port)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	19200
Parity:	Odd
Data / Stop:	7 / 1
Frame size:	14 Bytes
Comments:	---

Frame content:

Byte 0	Byte 1..5	Byte 6
Range	Digit 4..0	Function

...

...

Byte 7	Byte 8..11	Byte 12..13
Status	Option 1..4	End mark (CrLf)
		0x0D *0x0A*

Byte 0: *(Range)*

Value	V	mA	µA	A	Ω	Frequency	Capacitor
0110000	x.xxxx V	xx.xxx mA	xxx.xx µA	xx.xxx A	xxx.xx Ω	xx.xxx Hz	xx.xxx nF
0110001	xx.xxx V	xxx.xx mA	xxxx.x µA		x.xxxx kΩ	xxx.xx Hz	xxx.xx nF
0110010	xxx.xx V				xx.xxx kΩ	x.xxxx kHz	x.xxxx µF
0110011	xxxx.x V				xxx.xx kΩ	xx.xxx kHz	xx.xxx µF
0110100	xxx.xx mV				x.xxxx MΩ	xxx.xx kHz	xxx.xx µF
0110101					xx.xxx MΩ	x.xxxx MHz	x.xxxx mF
0110110					xxx.xx MΩ	xx.xxx MHz	xx.xxx mF
0110111						xxx.xx MHz	xxx.xx mF

Continuity mode is fixed (xxx.xx Ω)
Diode mode is fixed (x.xxxx V)

Bytes 1..5 *(Digits)*

The digits are transferred as BCD encoded number. Upper nibble is always 0b/011////, lower nibble is the value itself (0 = 0b////0000, 1 = 0b////0001, …)

Byte 6: *(Function)*

Value	Function
0110000 (0x30)	Current A (auto)
0110001 (0x31)	Diode
0110010 (0x32)	Frequency / Duty [1]
0110011 (0x33)	Ohm
0110101 (0x35)	Continuity

Value	Function
0110110 (0x36)	Capacitance
0111001 (0x39)	Current A (manual)
0111011 (0x3B)	Voltage
0111101 (0x3D)	Auto µA current [2]
0111111 (0x3F)	Auto mA current [2]

[1]) Check judge bit. If 0 --> frequency, if 1 --> duty.

[2]) Check VBAR bit.

Byte 7: *(Status)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	Judge	Sign	BATT	OL

Byte 8: *(Option 1)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	MAX	MIN	REL	0

Byte 9: *(Option 2)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	UL	PMAX	PMIN	0

Byte 10: *(Option 3)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	DC	AC	AUTO	VAHz

Byte 11: *(Option 4)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	0	0	HOLD	LPF

PeakTech 3440

Communication protocol

Connection settings:

USB	
Supported:	**No** *(uses virtual COM port)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	19230
Parity:	None
Data / Stop:	8 / 1
	variable

Warning!
Incomplete data!

Frame content:

Byte 0	Byte 1	Byte 2..n	Byte n+1
Start Byte (0xA0)	Content Identifier	Content data (variable width)	End byte (0xA1)

Content Data *(if ID=0x80)*

Byte	Description
0	Range & Unit. See below.
1	Value as float (single)
2	
3	
4	
5	OL value as float (single)
6	
7	
8	
9	? 0x40 decimal position (for display)
10	Display info (range, unit, AC (~) or DC (=) as fixed 4 bytes string
11	
12	
13	
14	? 0x01 (maybe date display format)
15	Date (YYMMDD) in BCD format
16	
17	
18	Time (HHMMSS) in BCD format
19	
20	
21	? 0x03
22	?
23	?

PeakTech 4000

Communication protocol

Connection settings:

USB	
Supported:	**No** *(USB adapter included)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	2400
Parity:	Even
Data / Stop:	8 / 1
Frame size:	14 Bytes
Comments:	Device sometimes sends invalid data for a short time if the user switches the mode of measurement. Validation of data strongly recommended.

Frame content:

Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4..8	Bytes 9..13
Option 1	Option 2	Option 3	Option 4	Primary (Pri.) digits	Secundary (Sec.) digits

Byte 0: *(Option 1)*

Upper nibble is always 1010, indicating start of a data frame, lower nibble tells the range:

Value	0000	0001	0010	0011	0100	0101	0110
V AC	x.xxxx V	xx.xxx V	xxx.xx V	xxx.x V			
dBm	xxx.xx dBm						
V DC	x.xxxx V	xx.xxx V	xxx.xx V	xxx.x V			
V DC + AC	x.xxxx V	xx.xxx V	xxx.xx V	xxx.x V			
mV DC	xx.xxx mV	xxx.xx mV					
mV AC	xx.xxx mV	xxx.xx mV					
mV DC + AC	xx.xxx mV	xxx.xx mV					
Hz	xx.xxx Hz	xxx.xx Hz	x.xxxx kHz	xx.xxx kHz	xxx.xx kHz	x.xxxx MHz	xx.xxx MHz
V Diode	x.xxxx V						
Ohm	xxx.xx Ω	x.xxxx kΩ	xx.xxx kΩ	xxx.xx kΩ	x.xxxx MΩ	xx.xxx MΩ	
Continuity	xxx.xx Ω						
Capacitance	xx.xx nF	xxx.x nF	x.xxx µF	xx.xx µF	x.xxx µF	xxxx µF	
µA DC	xxx.xx µA	xxxx.x µA					
µA AC	xxx.xx µA	xxxx.x µA					
µA DC + AC	xxx.xx µA	xxxx.x µA					
mA DC	xxx.xx mA	xxxx.x mA					
mA AC	xxx.xx mA	xxxx.x mA					
mA DC+AC	xxx.xx mA	xxxx.x mA					
A DC	x.xxxx A	xx.xxx A					
A AC	x.xxxx A	xx.xxx A					
A DC + AC	x.xxxx A	xx.xxx A					

Byte 1 *(Option 2)*

Bit 7	Bit 6	Bit 5	Bits 4..0
0	HOLD	Prim. OV	Operation mode (see below)

Value	Function
00000 (0x00)	Volt AC (V)
00001 (0x01)	Volt DC (V)
00010 (0x02)	Volt DC + AC (V)
00011 (0x03)	Millivolt DC (mV)
00100 (0x04)	Millivolt AC (mV)
00101 (0x05)	Millivolt DC + AC (mV)
00110 (0x06)	Frequency (Hz)
00111 (0x07)	Diode Volt (V)
01000 (0x08)	Resistance (Ω)
01001 (0x09)	Continuity (Ω)

Value	Function
01010 (0x0A)	Capacitance (F)
01011 (0x0B)	Microampere DC (µA)
01100 (0x0C)	Microampere AC (µA)
01101 (0x0D)	Microampere DC+AC (µA)
01110 (0x0E)	Milliampere DC (mA)
01111 (0x0F)	Milliampere AC (mA)
10000 (0x10)	Milliampere DC + AC (mA)
10001 (0x11)	Ampere DC (A)
10010 (0x12)	Ampere AC (A)
10011 (0x13)	Ampere DC + AC (A)

Byte 2: *(Option 3)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Sec. Neg.	Pri. Neg.	Manual	Sec. OL	REL	Sec. dBm	Sec. Hz

Byte 3: *(Option 4)*

Bit 7	Bits 6..4	Bit 3	Bit 2	Bit 1	Bit 0
0	Secondary range (see below)	Sec. Duty	AVG	MIN	MAX

Secondary display is shown if either Sec. Duty, Sec. dBm or Sec. Hz bits are set, or if the multimeter is in Min, Max, Max-Min or Avg mode. Otherwise the data from the second display needs to be ignored (might be malformed because never set). Secondary range depends on above called Sec. bits, or if none set the primary mode if Min, Max, Max-Min or AVG mode is set. To get the range for the second display use the table below.

Range bits	Sec. Duty	Sec. dBm + Hz	Pri. V + A	Pri. Ω
000	xx.xxx	xx.xxx	x.xxxx	xxx.xx
001		xxx.xx	xx.xxx	x.xxxx k
010		x.xxxx k	xxx.xx	xx.xxx k
011		xx.xxx k		xxx.xx k
100		xxx.xx k		x.xxxx M
101				xx.xxx M

Note for V + A: If set to mV / mA the decimal position is 1 digit to the right, for µA 2 digits.

Bytes 4..8: *(Primary digits)*
The digits, beginning with the most significant one. Digits are transferred in binary form, meaning 0x00 = 0, 0x09 = 9. Values > 0x09 are invalid.

Bytes 9..13: *(Secondary digits)*
Same as above

PeakTech 4090

Communication protocol

Connection settings:

USB	
Supported:	**No** *(uses virtual COM port)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	19230
Parity:	Odd
Data / Stop:	7 / 1
Frame size:	14 Bytes
Comments:	---

Frame content:

Byte 0	Byte 1..5	Byte 6	
Range	Digit 4..0	Function	...

	Byte 7	Byte 8..11	Byte 12..13
...	Status	Option 1..4	End mark (CrLf)

0x0D *0x0A*

Byte 0: *(Range)*

Value	V	2range A	Manual A	ADP	Ω	Frequency	Capacitor
0110000	x.xxxx V	Low (IVSL)	x.xxxx A	ADP4	xxx.xx Ω	xx.xxx Hz	xx.xxx nF
0110001	xx.xxx V	High (IVSH)	xx.xxx A	ADP3	x.xxxx kΩ	xxx.xx Hz	xxx.xx nF
0110010	xxx.xx V		xxx.xx A	ADP2	xx.xxx kΩ	x.xxxx kHz	x.xxxx µF
0110011	xxxx.x V		xxxx.x A	ADP1	xxx.xx kΩ	xx.xxx kHz	xx.xxx µF
0110100	xxx.xx mV		xxxxx A	ADP0	x.xxxx MΩ	xxx.xx kHz	xxx.xx µF
0110101					xx.xxx MΩ	x.xxxx MHz	x.xxxx mF
0110110					xxx.xx MΩ	xx.xxx MHz	xx.xxx mF
0110111						xxx.xx MHz	xxx.xx mF

22 A mode is fixed (xx.xxx A)
Continuity mode is fixed (xxx.xx Ω)
Diode mode is fixed (x.xxxx V)

Bytes 1..5 *(Digits)*

The digits are transferred as BCD encoded number. Upper nibble is always 0b/011////, lower nibble is the value itself (0 = 0b////0000, 1 = 0b////0001, …)

Byte 6: *(Function)*

Value	Function
0110000 (0x30)	22A current
0110001 (0x31)	Diode
0110010 (0x32)	Frequency / Duty [1]
0110011 (0x33)	Ohm
0110100 (0x34)	Temperature °C / °F [2]
0110101 (0x35)	Continuity

Value	Function
0110110 (0x36)	Capacitance
0111001 (0x39)	Manual A current
0111011 (0x3B)	Voltage
0111101 (0x3D)	Auto µA current [3]
0111110 (0x3E)	ADP
0111111 (0x3F)	Auto mA current [3]

[1]) Check judge bit. If 0 --> frequency, if 1 --> duty.
[2]) Check judge bit. If 0 --> °C, if 1 --> °F. **Note:** The digits will represent °C in any case! °F = °C*1.8+32
[3]) Check VBAR bit.

Byte 7: *(Status)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	Judge	Sign	BATT	OL

Byte 8: *(Option 1)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	MAX	MIN	REL	RMR

Byte 9: *(Option 2)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	UL	0	0	0

Byte 10: *(Option 3)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	DC	AC	AUTO	VAHz

Byte 11: *(Option 4)*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/A	0	1	1	0	VBAR	HOLD	LPF

PeakTech 4390

Communication protocol

Connection settings:

USB	
Supported:	**No** *(uses virtual COM port)*
Vendor ID:	---
Product ID:	---
Mode:	---
Frame size:	---
Comments:	---

RS232	
Supported:	**Yes**
Baud:	4800
Parity:	None
Data / Stop:	8 / 1 Bit(s)
Frame size:	10 Bytes
Comments:	You have to read the data as binary, not as text! Otherwise you will ran into problems with bytes 6..9 which may be 0x00, which is the end-of-string marker for several programming languages!

Warning!
Not validated data!

Frame content:

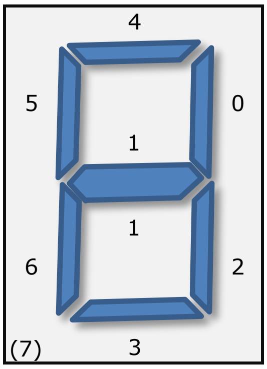
		Bit 7 / 3	Bit 6 / 2	Bit 5 / 1	Bit 4 / 0
Byte 0..1		Start identifier 0xA5 0xA5			
Byte 2	U	Thousands digit, upper nibble			
	L	AC	DC	Auto	1 (USB)
Byte 3	U	Hundreds digit, upper nibble			
	L	Thousands digit, lower nibble			
Byte 4	U	Tenths digit, upper nibble			
	L	Hundreds digit, lower nibble			
Byte 5	U	Ones digit, upper nibble			
	L	Tenths digit, lower nibble			
Byte 6	U	F	Ω	Rel	Hold
	L	Ones digit, lower nibble			
Byte 7	U	m	%	M	Continuity
	L	µ	n	k	Diode
Byte 8	U	°F	°C	0 *(unused)*	0 *(unused)*
	L	A	V	Hz	Batt
Byte 9	U	0 *(unused)*	0 *(unused)*	0 *(unused)*	0 *(unused)*
	L	0 *(unused)*	Min	Max	0 *(unused)*

Create digits:

First of all you need to build the digit value from the upper and lower nibble.
For example:
Thousands = (Byte2 AND 0xF0) OR (Byte3 AND 0x0F)

Now you have the segments that are set. With this you need to recreate the number itself. For example: 0x05 = 0b00000101 = digit 1

The MSB determinates the decimal point, except for the thousands digit. If the MSB in thousands digit is set it indicates a negative value.



(Template)

Communication protocol

Connection settings:

USB
Supported:
Vendor ID:
Product ID:
Mode:
Frame size:
Comments:

RS232
Supported:
Baud:
Parity:
Data / Stop:
Frame size:
Comments:

Frame content: