

SOFTWARE LICENSE AGREEMENT FOR APPA WinDMM100 95 / 98

IMPOTANT :

Please read carefully before using the Software.

This APPA WinDMM Software License Agreement ("SLA") is a legal agreement between you (either an individual or a single entity) and APPA TECHNOLOGY CORP. for the APPA WinDMM SOFTWARE PRODUCT AND DOCUMENTATION identified above, which includes RS-232 Protocol.

The SOFTWARE PRODUCT AND DOCUMENTATION provided to you by APPATECHNOLOGY CORP. is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

As you receive this SOFTWARE PRODUCT AND DOCUMENTATION, you agree to become bound by the terms of this agreement.

The license allows you to use the SOFTWARE PRODUCT AND DOCUMENTATION for internal purposes only on a single computer.

You may not market, distribute or transfer copies of SOFTWARE PRODUCT AND DOCUMENTATION to others or electronically transfer the SOFTWARE PRODUCT AND DOCUMENTATION from one computer to another over a network, you may not decompile, reverse engineer, disassemble or otherwise reduce the code of the SOFTWARE PRODUCT to a human perceivable form.

The RS-232 protocol is one part of the APPA WinDMM and is exactly and completely the same as the APPA WinDMM, but no other warranties for the end user to write a driver using the RS-232 protocol.

The RS-232 protocol :

1. Communication Type : RS-232C
2. Communication protocol :
 - A : Baud Rate : 9600 bps
 - B : Data length : 8 bits
 - C : Parity check : None
 - D : Stop bit : 1 bit
3. Data format : The data format is HEX code.

A. PC sends a command to DMM for requesting to read as follow :

0x55	0x55	0x00	0x00	0xAA
------	------	------	------	------

After sending the command, wait for receiving data format from DMM, the time out setting must be bigger than 450ms.

B. When DMM receives the command from PC, will send the data format to PC as follow :

0x55	0x55	0x00	0x0E	Rotor Code	Blue Code	Key Code	Range Code	Main Reading (5 Bytes)
------	------	------	------	---------------	--------------	-------------	---------------	---------------------------

	Sub-reading (5 Bytes)	Checksum
		sum

4.

a. Rotor code

00H	01H	02H	03H	04H	05H	06H	07H	08H	09H
OFF	V	mV	Ohm	Diode	mA	A	Cap.	Hz	Temp.

b. Blue code

	V	mV	Ohm	Diode	mA	A	Cap.	Hz	Temp.
00H	AC	AC	Ohm	Diode	AC	AC	Cap	Hz	deg.C
01H	DC	DC	Low Ohm	Beeper	DC	DC	-----	Duty Factor	deg.F
02H	AC+ DC	AC+ DC	-----	-----	AC+ DC	AC+ DC	-----	-----	-----

c. Key code Ignore

d. Range code

	DC V	AC V	(AC+DC) V	DC mV	AC mV	(AC+DC) mV
00H (auto)	2V	2V	2V	20mV	20mV	20mV
01H (auto)	20V	20V	20V	200mV	200mV	200mV
02H (auto)	200V	200V	200V			
03H (auto)	1000V	750V	750V			
80H (manual)	2V	2V	2V	20mV	20mV	20mV
81H (manual)	20V	20V	20V	200mV	200mV	200mV
82H (manual)	200V	200V	200V			
83H (manual)	1000V	750V	750V			

	DC mA	AC mA	(AC+DC) mA	DC A	AC A	(AC+DC) A
00H (auto)	20mA	20mA	20mA	2A	2A	2A
01H (auto)	200mA	200mA	200mA	10A	10A	10A
80H (manual)	20mA	20mA	20mA	2A	2A	2A
81H (manual)	200mA	400mA	200mA	10A	10A	10A

	Ohm	Low Ohm	Cap	Hz	□	□
00H (auto)	200Ohm	2kOhm	4nF	20Hz	400□	400□
01H (auto)	2kOhm	20kOhm	40nF	200Hz	1200□	2192□
02H (auto)	20kOhm	200kOhm	400nF	2kHz		
03H (auto)	200kOhm	2MOhm	4μF	20kHz		
04H (auto)	2Mohm	20MOhm	40μF	200kHz		
05H (auto)	20Mohm		400μF	1MHz		
06H (auto)			4mF			
07H (auto)			40mF			
80H (manual)	200Ohm	2kOhm	4nF	20Hz	□	□
81H (manual)	2kOhm	20kOhm	40nF	200Hz	400□	400□
82H (manual)	20kOhm	200kOhm	400nF	2kHz	1200□	2192□
83H (manual)	200kOhm	2MOhm	4μF	20kHz		
84H (manual)	2Mohm	20MOhm	40μF	200kHz		
85H (manual)	20Mohm	200Mohm	400μF	1MHz		
86H (manual)	200Mohm	2Gohm	4mF			
87H (manual)	2Gohm		40mF			

e. Main Reading

Data LSB	Data 2SB	Data HSB	Status	Function
24 bits signed number			Refer Point code unit table	Refer Function table

f. Sub-reading

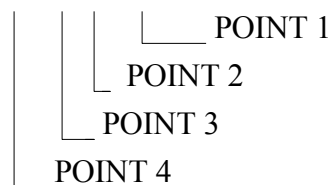
Data LSB	Data 2SB	Data HSB	Status	Function
24 bits signed number			Refer Point code unit table	Refer Function table

g : State: Bit7 to Bit3 are for Unit ; Bit2 to Bit0 are for Decimal Point.

1.POINT CODE :

Bit2	Bit1	Bit0	Point
0	0	0	None
0	0	1	Point1
0	1	0	Point2
0	1	1	Point3
1	0	0	Point4

Remark : 2 . 0 . 0 . 0 . 0



2.UNIT CODE :

Bit7	Bit6	Bit5	Bit4	Bit3	Unit	Bit7	Bit6	Bit5	Bit4	Bit3	Unit
0	0	0	0	0	None	0	1	1	1	1	Hz
0	0	0	0	1	V	1	0	0	0	0	KHz
0	0	0	1	0	mV	1	0	0	0	1	MHz
0	0	0	1	1	A	1	0	0	1	0	□
0	0	1	0	0	mA	1	0	0	1	1	□
0	0	1	0	1	dB	1	0	1	0	0	s
0	0	1	1	0	dBm	1	0	1	0	1	ms
0	0	1	1	1	nF	1	0	1	1	0	ns
0	1	0	0	0	uF	1	0	1	1	1	V
0	1	0	0	1	mF	1	1	0	0	0	mV
0	1	0	1	0	Ω	1	1	0	0	1	A
0	1	0	1	1	KΩ	1	1	0	1	0	mA
0	1	1	0	0	MΩ	1	1	0	1	1	Ω
0	1	1	0	1	GΩ	1	1	1	0	0	KΩ
0	1	1	1	0	%	1	1	1	0	1	MΩ

h : FUNCTION :

CODE		DATA
00H	0	NONE
01H	1	Input Reading
02H	2	Freq
03H	3	Period
04H	4	Duty Factor
08H	8	Stamp (Store, Recall,Login,Logout)
09H	9	Store
0AH	10	Recall
0CH	12	Auto Hold
0DH	13	Max
0EH	14	Min
10H	16	Peak Hold Max
11H	17	Peak Hold Min
17H	23	Δ
19H	25	Ref
1AH	26	dBm
1BH	27	dB
25H	37	Avg
26H	38	Probe “character”
27H	39	Er “character”
28H	40	FUSE “character”
29H	41	PAUS “character”
2AH	42	Logout Max data
2BH	43	Logout Min data
2CH	44	Logout Max Turning Point
2DH	45	Logout Min Turning Point
2EH	46	Logout data
2FH	47	Period Time
30H	48	FULL “character”
31H	49	EPEr “character”
32H	50	EEPROM □□□□ “character”
33H	51	Login Stamp

i : Checksum:

The last byte is the SUM of every byte in the received data format except Checksum itself, you can use the Checksum (using the last two digits) to check the data receiving from DMM is correctly or not.

Example for Checksum :

$$0x55+0x55+0x00+0x0E+0x01+0x01+0x00+0x00+0x64+0x01+0x00+0x0c+x01+0x0e+0x00+0x0e+0x00+0x00 = 0x12C$$

$$\text{Checksum} = 0x2C$$