

**Digital Thermometer & Thermostat  
with RS-232 Interface  
Model 232DTT**

**Document No. 232DTT1795**

**B&B Electronics Mfg. Co. Inc.**

P.O. Box 1040 -- Ottawa, IL 61350  
PH (815) 433-5100 -- FAX (815) 434-7094

**Internet:**

<http://www.bb-elec.com>  
[sales@bb-elec.com](mailto:sales@bb-elec.com)  
[techsupt@bb.elec.com](mailto:techsupt@bb.elec.com)

© B&B Electronics -- April 1995

---

# TABLE OF CONTENTS

<b>CHAPTER 1. GENERAL INFORMATION.....</b>	<b>1</b>
INTRODUCTION.....	1
FEATURES .....	1
PACKING LIST .....	1
SPECIFICATIONS .....	1
RS-232 CONNECTIONS.....	2
<b>CHAPTER 2. 232DTT FUNCTIONS.....</b>	<b>3</b>
DIGITAL THERMOMETER .....	3
PROGRAMMABLE THERMOSTAT .....	3
<b>CHAPTER 3. 232DTT DATA FORMAT.....</b>	<b>5</b>
<b>CHAPTER 4. 232DTT COMMANDS.....</b>	<b>7</b>
READ TEMPERATURE .....	7
READ STATUS.....	7
READ HIGH.....	8
READ LOW .....	8
CLEAR STATUS REGISTER.....	8
SET HIGH THERMOSTAT .....	9
SET LOW THERMOSTAT.....	9
<b>CHAPTER 5. SOFTWARE.....</b>	<b>11</b>
INSTALLATION .....	11
WINDOWS SETUP .....	11
<b>APPENDIX A. USING SIMPTERM.....</b>	<b>A-1</b>
<b>APPENDIX B. DECIMAL/ASCII TABLE.....</b>	<b>B-1</b>

# Chapter 1. General Information

## Introduction

The 232DTT is a digital thermometer and thermostat which operates through an RS-232 interface. Non-volatile registers hold programmable high and low thermostat settings. Current temperature is reported by polling the DTT, while thermostat operations continue as long as the unit is powered.

## Features

- $\pm 0.5$  °C accuracy and resolution
- High, low and hysteresis type thermostat outputs
- Port-powered from host RS-232 port (RTS & DTR high)
- Thermostat thresholds programmable in non-volatile registers
- Status Register latches if thermostat outputs have tripped

## Packing List

Examine the shipping carton and contents for physical damage. The following items should be in the shipping carton:

1. 232DTT unit
2. One 232DTT 3.5" disk
3. This instruction manual

If any of these items are damaged or missing, contact B&B Electronics immediately.

## Specifications

Communications:

Standard:	RS-232 (unit is DCE)
Baud Rate:	1200 to 9600 (automatic detection)
Format:	8 data bits, 1 stop bit, no parity
Connector:	DB-25S (female)

Thermometer Error

$\pm 0.5$ °C from 0 - 70° C ( $\pm 0.9$ °F from 32 - 158°F)
$\pm 1$ °C from -40 °C to 0 °C and 70 °C to 85 °C * ( $\pm 1.8$ °F from -40 °F to 32 °F and 158 °F to 185 °F)

\* The 232DTT is guaranteed to operate from 0-70 °C.

Thermometer Resolution

0.5 °C (0.9 °F)

Temperature Conversion Time

1 second

Open Collector Thermostat Outputs

Low (on) voltages: 1.2VDC @ 100 mA max.

High (off) voltages: 40VDC max.

Power Supply

Port powered with RTS and DTR outputs set high

External power: 12VDC @ 30mA

**NOTE: When using an external supply, the supply should be connected only to specifically labeled power inputs (power jack, terminal block, etc.). Connecting an external power supply to the handshake lines may damage the unit. Contact technical support for more information on connecting an external power supply to the handshake lines.**

## RS-232 Connections

The 232DTT communicates using pins 2, 3, and 7 (TD, RD and GND). However, power is drawn from the handshake lines RTS (pin 4) and DTR (pin 20). These connections must be present to power the unit.

**Table 1. RS-232 Pinouts**

Signal	Pin
TD	2
RD	3
RTS	4
GND	7
DTR	20

## Chapter 2. 232DTT Functions

### Digital Thermometer

The 232DTT continuously measures temperature, with a new measurement completed once per second. When a Read Temperature command is received, the DTT immediately reports back the result of the last measurement completed.

### Programmable Thermostat

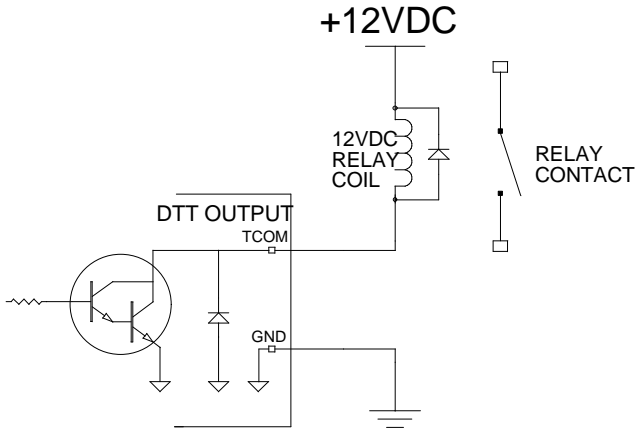
The 232DTT also functions as a thermostat. Two non-volatile registers can be programmed. The high threshold register, TH, is programmed using the !OSH command. The low threshold register, TL, is programmed using the !OSL command. A complete command description can be found in Chapter 4. Thermostat operation is a continuous background function. As long as the unit is powered, the thermostat outputs will be serviced.

Three open collector outputs provide thermostatic controls. These outputs are current sinking (low) when on, high impedance (high) when off.

1. T<sub>HIGH</sub> - Turns on when the temperature equals or exceeds the user programmed TH register.
2. T<sub>LOW</sub> - Turns on when the temperature is equal to or less than the user programmed TL register.
3. T<sub>COM</sub> - Turns on when the temperature equals or exceeds the TH register and turns off when the temperature is equal to or less than the TL register. This output allows the user to program any amount of hysteresis.

Remember, these are open collector outputs. They do not source any current (similar to a contact closure to ground). For example, to drive a relay with a 12VDC coil with the TCOM output, connect as follows:

1. Connect +12Vdc to one side of the relay coil.
2. Connect the other side of the coil to the TCOM output.
3. Connect the ground of the 12V supply to the GND terminal block.



**Figure 1. Relay Output Connection**

When connecting a mechanical relay coil, or any inductive load to the DTT outputs, a reverse bias diode must be connected across the load to protect the DTT output against large voltages caused by the inductive load.

The DTT outputs can handle a maximum voltage of 40V and a maximum current of 100mA. Do not use a relay with coil ratings exceeding these values.

## Chapter 3. 232DTT Data Format

The DTT uses 9-bit, two's complement format to represent temperature data. The ninth bit is the sign bit. If this bit is a 0, the temperature is above 0° C. If the bit is 1, the temperature is less than 0° C. Each bit represents 1/2° C.

**Table 2. Temperature/Data Relationships**

Temperature	DTT output	hex value	dec value
+125° C	0 11111010	00 FAh	250
+25° C	0 00110010	00 32h	50
1/2° C	0 00000001	00 01h	1
0° C	0 00000000	00 00 h	0
-1/2° C	1 11111111	01 FFh	511
-25° C	1 11001110	01 CEh	462
-55° C	1 10010010	01 92h	402

Two's complement format provides a way for digital devices to handle negative numbers. To convert the two's complement value to a meaningful number representing temperature, follow these steps:

1. Check bit nine. In our case this is the entire first character. If the character is decimal 0 (NUL), then the value in the second character is positive and no further complementing is necessary, go to step 6. If character is decimal 1, then continue with step 2.
2. Use the NOT function of your programming language to invert (or complement) the value in the second character.
3. We need to insure that we are only dealing with an eight bit variable. You may be able to define this in your programming language. To be sure, we can use the AND function to limit our new value to only eight bits by ANDing the value from step 2 with FFh. This masks off all but the lowest eight bits.
4. Add a 1 to the value from step 3. This completes the two's complement portion of the conversion.
5. Multiply our new value by -1. This simply adds the "-" sign to our temperature reading.

6. To convert to Celsius, divide by two.
7. To convert to Fahrenheit, multiply the Celsius value by 9; divide the result by 5; then add 32.

In the BASIC language, these steps look like this:

```
'DTT data has been stored in CHAR1$ and CHAR2$
negative = ASC(CHAR1$)      'ASCII val of sign char (1 or 0)
reading = ASC(CHAR2$)      'ASCII val of reading
```

```
IF negative = 1 THEN
    reading = NOT reading  'invert all bits (complement)
    reading = reading AND &HFF  'limit variable to 8 bits
    reading = reading + 1    'finish complement procedure
    reading = reading * -1  'multiply by -1 to add minus sign
END IF
```

```
Ctemp = reading / 2          'Celsius temperature
Ftemp = (9 * C temp / 5) + 32 'Fahrenheit temperature
```

## Chapter 4. 232DTT Commands

The DTT commands can be issued from a program, or even typed in from a simple terminal program. Two files are included on the 232DTT disk to help learn the DTT commands. A sample program with source code written in QuickBASIC 4.5 demonstrates all DTT functions and a simple terminal program is also included. In addition, a simple data logging utility is provided which allows data to be logged to a file. Appendix A demonstrates how to communicate with the DTT from the terminal program. A decimal/ASCII conversion table is included in Appendix B.

### Read Temperature

*Command:* !0RT

*Arguments:* none

*Response:* temperature reading in 9 bit two's complement format (shown in boldface)

*Dec. Example:* !0RT<0><**46**>

*ASCII Example:* !0RT<NUL>.

*Description:* The decimal 0 (NUL) followed by decimal 46, equivalent to an ASCII "." converts to a temperature of +23° C.

### Read Status

*Command:* !0RS

*Arguments:* none

*Response:* contents of 8 bit status register (shown in boldface)

*Dec. Example:* !0RS<0><**66**>

*ASCII Example:* !0RS<NUL>**B**

*Hex. Example:* !0RS<0><**42**> (since this is an eight bit register, it may be helpful to use hex representation)

*Description:* The first character (NUL) has no meaning in this response. Bit 1 is set, indicating normal operation. Bit 5 is clear indicating the low thermostat has not tripped. Bit 6 is set, indicating THIGH has tripped since the last !0SC command.

**Table 3. Status Register Description**

Bit 0	X (don't care)
Bit 1	operation
Bit 2	X
Bit 3	X
Bit 4	X
Bit 5	Low Thermostat Tripped Flag
Bit 6	High Thermostat Tripped Flag
Bit 7	X

## Read High

*Command:* !0RH

*Arguments:* none

*Response:* contents of TH register in 9 bit two's complement format (shown in boldface)

*Dec. Example:* !0RH<0><**50**>

*ASCII Example:* !0RH<NUL>**2**

*Description:* The TH register contains decimal 0 (ASCII NUL) and decimal 50 (ASCII 2). This converts to a temperature of 25 ° C.

When the current temperature is  $\geq 25^\circ\text{C}$ , the high thermostat output (THIGH) will turn on.

## Read Low

*Command:* !0RL

*Arguments:* none

*Response:* contents of TL register in 9 bit two's complement format (response is shown in boldface)

*Dec. Example:* !0RL<0><**36**>

*ASCII Example:* !0RL<NUL>**\$**

*Description:* The TL register contains decimal 0 (ASCII NUL) and decimal 36 (ASCII \$). This converts to a temperature of 18 ° C.

When the current temperature is  $\leq 18^\circ\text{C}$ , the low thermostat output (TLOW) will turn on.

## Clear Status Register

*Command:* !0SC

*Arguments:* none

*Response:* none

*Example:* !0SC

*Description:* The Clear Status command resets the DTT. Bits 5 and 6 which latch the thermostat outputs are cleared if the current temperature is within the bounds of TH and TL.

## Set High Thermostat

*Command:* !0SH

*Arguments:* 9 bit, two's complement value to set non-volatile TH register.

*Response:* none

*Dec. Example:* !0SH<0><**64**>

*ASCII Example:* !0SH<NUL>**@**

*Description:* Sending the decimal arguments 0 (ASCII NUL) and 64 (ASCII @) sets the TH register to a temperature of 32 ° C. This is a non-volatile register and will not change until this command is issued

again. After sending the Set High Thermostat command, delay for at least 10 ms to allow the DTT to program the register.

### Set Low Thermostat

*Command:* !0SL

*Arguments:* 9 bit, two's complement value to set non-volatile TL register.

*Response:* none

*Dec. Example:* !0SL<0><33>

*ASCII Example:* !0SL<NUL >!

*Description:* Sending the decimal arguments 0 (ASCII NUL) and 33 (ASCII !) sets the TL register to a temperature of 16.5 ° C. This is a non-volatile register and will not change until this command is issued again. After sending the Set Low Thermostat command, delay for at least 10 ms to allow the DTT to program the register.

**Table 4. Command Summary**

Function	Command	Argument	Return Value
read temperature	!0RT	none	9 bit, 2's comp.
read status register	!0RS	none	byte
read high thermostat	!0RH	none	9 bit, 2's comp.
read low thermostat	!0RL	none	9 bit, 2's comp.
clear status register	!0SC	none	none
set high thermostat*	!0SH	9 bit, 2's comp.	none
set low thermostat*	!0SL	9 bit, 2's comp.	none

**\*NOTE:** After “programming” the 232DTT, there is approximately 10ms of down time when the unit will not respond. Do not attempt to communicate with the DTT for at least 10ms after programming the thermostat registers.

## Chapter 5. Software

There is both a DOS program (DTT\_DOS.EXE) and a Windows program (232DTT\_W.EXE) which allow you to read the current temperature in either Fahrenheit or Celsius, as well as set the upper and lower thresholds.

### Installation

To install on your hard drive follow these steps:

- Place the disk in drive A.
- Type **A:** and press the **ENTER** key.
- Type **INSTALL** and press the **ENTER** key.

### Windows Setup

After the software is installed on your hard disk, the following files should be located in the 232DTT directory: 232DTT\_W.EXE, VBRUN300.DLL, and MSCOMM.VBX.

To set up the 232DTT under windows follow the following steps:

1. Run Windows.
2. Select the "FILE" menu under the Program Manager.
3. Select the "NEW" option.
4. From the "New Program Object" window select "Program Item" option and then select "OK".
5. From the "Program Item Properties" window:
  - a) Type in the "Description" box:  
232DTT
  - b) Type in the "Command Line" box:  
C:\232DTT\232DTT\_W.EXE
6. From the "Program Item Properties" menu select the "OK" option.

Setup of the 232DTT is now complete. The 232DTT icon should now appear in the current window. Simply double click on the icon to execute the program.

## Appendix A. Using SimpTerm

B&B Electronics shareware terminal program, SimpTerm, is included on the DTT floppy disk. SimpTerm is a powerful troubleshooting tool which will allow you to communicate through up to four serial ports at any address or interrupt. You will need to know the address and interrupt of the serial port that the DTT is connected to.

To use SimpTerm to communicate with the DTT, follow the instructions to open the port connected to the DTT.

1. Raise RTS and DTR to power the DTT, or connect an external power supply.
2. Check that the baud rate is set anywhere from 1200 - 9600 baud.
3. Type the following command, the second character is a zero, all commands must be in capital letters: **!0RT**
4. The DTT will respond with two characters indicating temperature. Room temperature is usually NUL followed by a character close in the ascii table to a comma. A NUL is indicated by **<0>**.
5. To set the TH register to 25° C, follow, type the following: **!0SH<0>2** The decimal value of the ASCII 2 is 50. This corresponds to 25° C. The NUL indicates that it is to be a positive value.\*
6. To read back the value stored in the TH register, type the following command: **!0RH**. If step 5 has been successfully completed the DTT will respond with **<0>2**.

\* The **<0>** is actually a NUL character and is not represented by a keyboard key. SimpTerm allows these characters to be transmitted using the following procedure:

Hold down the ALT key, type the decimal value (two digits) on the numeric keypad and release the ALT key. For example, the NUL is transmitted by: ALT 00 [release ALT].

## Appendix B. Decimal/ASCII Table

DECIMAL to HEX to ASCII CONVERSION TABLE												
DEC	HEX	ASCII	KEY	DEC	HEX	ASCII	DEC	HEX	ASCII	DEC	HEX	ASCII
	0	NUL	ctrl @	32	20	SP	64	40	@	96	60	'
1	1	SOH	ctrl A	33	21	!	65	41	A	97	61	a
2	2	STX	ctrl B	34	22		66	42	B	98	62	b
3	3	ETX	ctrl C	35	23	#	67	43	C	99	63	c
4	4	EOT	ctrl D	36	24	\$	68	44	D	100	64	d
5	5	ENQ	ctrl E	37	25	%	69	45	E	101	65	e
6	6	ACK	ctrl F	38	26	&	70	46	F	102	66	f
7	7	BEL	ctrl G	39	27	'	71	47	G	103	67	g
8	8	BS	ctrl H	40	28	(	72	48	H	104	68	h
9	9	HT	ctrl I	41	29	)	73	49	I	105	69	i
10	A	LF	ctrl J	42	2A	*	74	4A	J	106	6A	j
11	B	VT	ctrl K	43	2B	+	75	4B	K	107	6B	k
12	C	FF	ctrl L	44	2C	,	76	4C	L	108	6C	l
13	D	CR	ctrl M	45	2D	-	77	4D	M	109	6D	m
14	E	SO	ctrl N	46	2E	.	78	4E	N	110	6E	n
15	F	SI	ctrl O	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	ctrl P	48	30	0	80	50	P	112	70	p
17	11	DC1	ctrl Q	49	31	1	81	51	Q	113	71	q
18	12	DC2	ctrl R	50	32	2	82	52	R	114	72	r
19	13	DC3	ctrl S	51	33	3	83	53	S	115	73	s
20	14	DC4	ctrl T	52	34	4	84	54	T	116	74	t
21	15	NAK	ctrl U	53	35	5	85	55	U	117	75	u
22	16	SYN	ctrl V	54	36	6	86	56	V	118	76	v
23	17	ETB	ctrl W	55	37	7	87	57	W	119	77	w
24	18	CAN	ctrl X	56	38	8	88	58	X	120	78	x
25	19	EM	ctrl Y	57	39	9	89	59	Y	121	79	y
26	1A	SUB	ctrl Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	ctrl [	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	ctrl \	60	3C	<	92	5C	\	124	7C	
29	1D	GS	ctrl ]	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	ctrl ^	62	3E	>	94	5E	^	126	7E	~
31	1F	US	ctrl _	63	3F	?	95	5F	_	127	7F	DEL