Serial
PC Watchdog™
Command Addendum

BERKSHIRE PRODUCTS, INC.

Phone: 770-271-0088

http://www.berkprod.com/

Rev: 1.00

Table of Contents

1. NOTES **1**

2. FIRMWARE NOTES **2**

2.1 Ten (10) Byte Command Format **2**
2.2 Change Serial Speed to 14.4K Baud **2**
2.2.1 Sample Code for Baud Change 3
2.3 Method to Determine New Serial Type **4**

3. LOW-LEVEL ACTIVE MODE OPERATION **5**

3.1 Command Summary **5**
3.2 Command: 0x02 – Read Temp - Re-trigger Watchdog **7**
3.3 Command: 0x04 - Get Status Information **8**
3.4 Command: 0x08 - Get Firmware Version Number **9**
3.5 Command: 0x0C - Get Dip Switch Settings **10**
3.6 Commands: 0x10 & 0x11 - Read / Write Current POD Time **11**
3.7 Commands: 0x14 & 0x15 - Read / Save Non-Volatile POD Time **12**
3.8 Commands: 0x18 & 0x19 - Read / Write Current Watchdog Time **13**
3.9 Command: 0x1A - Read Holding Register Watchdog Time **14**
3.10 Commands: 0x1C & 0x1D - Read / Save Non-Volatile Watchdog Time **15**
3.11 Command: 0x20 - Read Re-trigger Count **16**
3.12 Command: 0x30 – Enable / Disable Watchdog **17**
3.13 Command: 0x40 – Power Module (External Only) **18**
3.14 Commands: 0x50 & 0x51 - Read / Write Relay Control (Internal Only) **19**
3.15 Commands: 0x58 & 0x59 - Read / Write Temperature Offset **20**
3.16 Commands: 0x5E Buzzer Control (Internal Only) **21**
3.17 Commands: 0x5F NV Buzzer Control (Internal Only) **22**
3.18 Command: 0x80 - Reset PC **23**
3.19 Command: 0x84 - Get / Clear Reset Count **24**
3.20 Commands: 0x88 & 0x89 - Read / Write Reset Relay Pulse **25**
3.21 Commands: 0xC0 & 0xC1 - Read / Write User ID Data **26**
3.22 Invalid Commands **27**
3.23 Special Diagnostics - 0xE0 to 0xFF (Do Not Use) **27**

1. Notes

This manual covers the new low level commands that have been added to the Internal and External Serial PC Watchdogs. This only covers the new commands. The original ones are described in the existing manuals. The new firmware on either board will be 3.xx or higher.

The latest versions of all manuals and sample code can be found on our site at:

http://www.berkprod.com/

If you have any questions, corrections, or feedback about this manual please contact us at:

http://www.berkprod.com/Other_Pages/Contact_Us.aspx

2. Firmware Notes

This section covers the new 10 byte command format and the method of switching the baud rate on the boards to 14.4K Baud for those boards with firmware equal or higher than 3.xx.

2.1 Ten (10) Byte Command Format

The new boards (External Rev-C and Internal Rev-D) now support a 10 byte format to allow them to send and receive data packets that in most cases are identical to the low level USB commands. The boards still accept their current 5 & 6 byte command packets and in some cases these new 10 bytes command will overlap and extend the old commands.

The data is sent as ASCII/Hex where each hex byte is represented as two ASCII bytes. The new data format has three bytes: a one byte command and a two byte data portion: the new send format is:

0x01 - 0x57 - 0x88 - CMDh - CMDl - MSBh - MSBl - LSBh - LSBl - Csum

The first three bytes are absolute constant values and the last byte is the same simple checksum. As an example consider sending a command 0x19 to set the new watchdog timer to 500 (0x01f4) seconds. The data packet would look like:

0x01 – 0x57 – 0x88 – 0x31 – 0x39 – 0x30 – 0x31 – 0x46 – 0x34 – 0xdb

NOTE: Use 0x41 – 0x46 for ASCII A-F

The response packet is always 9 bytes. It always ends with CR (0x0d) and does not have the leading 0x01 or a checksum:

0x57 – 0x88 – CMDh - CMDl - MSBh - MSBl - LSBh - LSBl – 0x0d

2.2 Change Serial Speed to 14.4K Baud

The new boards (External Rev-C and Internal Rev-D) now support a method to put them into 14.4K Baud. They always power up at 1200 Baud and they can be forced back to 1200 Baud by putting the TxD line in a break condition for 20 milli-seconds or longer.

Normally when your software starts it should use the break condition to ensure you are at the correct Baud. If a transmit packet fails then send a break condition and retry the Baud switch again just in case the power was lost on the Serial PC Watchdog. Once the watchdog is at 14.4K Baud it will not switch back, even after it has timed out.

2.2.1 Sample Code for Baud Change

The following sample Windows code shows how the switch is done:

```
#define ENQ 0x05
#define ESC 0x1b
#define DC1 0x11
#define DC2 0x12

    j = 0 ;
    for(;;)
    {
        // Update the DCB: 1200 - 2 stop bit (8 bits, no parity)
        DcbSerCom.BaudRate = CBR_1200 ;
        DcbSerCom.StopBits = TWOSTOPBITs ;
        // write back updated DCB info
        i = SetCommState(hSerCom, &DcbSerCom);

        SendBreak() ;       // TxD in break for 20mS - force 1200
        if(++j == 3)
        {
            return false ;  // exit and leave at 1200 baud
        }

        // Special chars for Baud switch -  these must
        // be sent right after break at 1200 Baud
        buff[0] = ESC ;
        buff[1] = DC1 ;
        i = WriteFile(hSerCom, (LPCVOID)buff, 2, &dwComSent, NULL) ;
        // Allow time for data to xmit at 1200 Baud
        // 1/1200 * 11 * 2 = 18.33ms to send the data
        Sleep(30) ;

        // Update the DCB: 14400 - 1 stop bit (8 bits, no parity)
        DcbSerCom.BaudRate = CBR_14400 ;
        DcbSerCom.StopBits = ONESTOPBIT ;
        // write back updated DCB info
        i = SetCommState(hSerCom, &DcbSerCom);

        // now send an ENQ at 14.4K - see if Wdog switched
        i = TransmitCommChar(hSerCom, ENQ) ;
        i = ReadFile(hSerCom, (LPVOID)buff, 1, &dwComRecv, NULL) ;
        // if Wdog answered with DC2 then it worked.
        if( i && (buff[0] == DC2))
            break ;
   }
   return true ;
```

The ESC/DC1 and the ENQ must be done right after the TxD break or the watchdog will still stay at 1200 Baud.

2.3 Method to Determine New Serial Type

One method to determine which board type you have is to do an initial command send using the old 5 byte format at 1200 Baud using the CMD_REV_MAJOR (0x30). If the value returned is 3 or higher then you have a board that will support these new features.

3. Low-Level Active Mode Operation

These low level commands here are not recommended for new Windows users. Use the new DLL on the CD.

3.1 Command Summary

This tables summarizes the commands that can be sent:

Command	Description	NV
0x02	Read Temp – Re-trigger Watchdog	No
0x04	Get Status Information	No
0x08	Get Firmware Version	No
0x0C	Get Dip Switch Settings	No
0x10	Read Current ARM Time	No
0x11	Write Current ARM Time	No
0x14	Read Non-Volatile Arm Time	No
0x15	Write Non-Volatile Arm Time	Yes
0x18	Read Current Watchdog Time	No
0x19	Write Current Watchdog Time	No
0x1A	Read Hold Register Watchdog Time	No
0x1C	Read Non-Volatile Watchdog Time	No
0x1D	Write Non-Volatile Watchdog Time	Yes
0x20	Read Re-trigger Count	No
0x30	Enable / Disable Watchdog	No
0x40	Power Module Control	No
0x50	Read Relay Control	No
0x51	Write Relay Control	Yes
0x58	Read Temperature Offset	No
0x59	Write Temperature Offset	Yes
0x5E	Buzzer Control	No
0x5F	NV Buzzer Control	Yes
0x80	Reset PC	No
0x84	Get/Clear Reset Count	No
0x88	Read Reset Relay Pulse Width	No
0x89	Write Reset Relay Pulse Width	Yes
0xC0	Read User ID Data	No
0xC1	Write User ID Data	Yes

The NV column tells you if the command will write data to the non-volatile memory and thus will take longer to execute. Allow up to 100mS for a non-volatile memory write command.

The Serial PC Watchdogs uses a 3 byte report format. The report format in both directions is formatted as:

Byte 0	Byte 1	Byte 2
CMD	Data MSB	Data LSB

*Do not use commands higher than **0xE0**. They are for testing purposes only and will produce undesirable results.*

3.2 Command: 0x02 – Read Temp - Re-trigger Watchdog

This command will return the internal temperature of the PC and it also serves to re-trigger ("tickle") the watchdog to reset the internal countdown timer.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x02	Not Used	Not Used

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x02	Not Used	Temp

Notes:

Returns the temperature data in degrees Celsius. The temperature readings are updated once per second. This is a signed value that can report temperatures less than zero.

3.3 Command: 0x04 - Get Status Information

This command will return the status of the board. It is recommended that this command be used every once in a while. The non-volatile memory chip employs hardware circuits and software features to prevent inadvertent data corruption. However no method is always perfect.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x04	Not Used	Not Used

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x04	Not Used	Bits

Notes:

The response data byte is a bit field defined as:

- **D0:** Set to one (1) if non-volatile memory OK. If zero (0) then all data was cleared.
- **D1:** Set to one (1) if temperature sensor is OK.
- **D2:** Not used - will read as zero.
- **D3:** Set if EEPROM was corrupted
- **D2:** Set if EEPROM write failed
- **D5:** Set to one means the board is disabled by the disable command.
- **D6:** Set to one (1) means the board has POD Extend switch on. The board has finished the POD time but is waiting for the first Temp read to “tickle” the watchdog.
- **D7:** Set to one (1) when the board has completed its power-on delay and is armed.

3.4 Command: 0x08 - Get Firmware Version Number

This command will return the numeric version of the firmware on the board.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x08	Not Used	Not Used

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x08	Whole#	Fraction

Notes:

The response data will be two bytes. The MSB is the whole number portion of the version and the LSB is the fraction.
EX: 0x01 0x09 = Version 1.09

3.5 Command: 0x0C - Get Dip Switch Settings

This command will return the setting of the Dip switch. The returned data is a byte (8 bits) for each switch with D0 representing SW8 through D7 for SW1. A switch that is ON will read as a one (1).

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x0C	Not Used	Not Used

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x0C	Not Used	Bits

3.6 Commands: 0x10 & 0x11 - Read / Write Current POD Time

When you use the Watchdog for the first time it will wait 2.5 minutes, or it may use a programmed value, after a power-up or a reset before it arms itself. This allows the PC additional time to complete a reboot sequence. These commands allow you to check and shorten / lengthen the current delay. See the next section for storing a new wait time.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x10	Not Used	Not Used
0x11	Delay Time	Delay Time

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x10	Delay Time	Delay Time
0x11	Not Used	Flag

Notes:

The delay time sent can range from 0x0000 to 0xFFFF seconds. If the time is less than the current remaining wait time then the arming will occur sooner. The board will return a flag byte after a write operation. The flag will be zero if the board is already passed the Power-On-Delay (**POD**) and is armed. This command is only active for the current session and the board will revert to its programmed delay at the next reboot.

3.7 Commands: 0x14 & 0x15 - Read / Save Non-Volatile POD Time

This command allows you to shorten or lengthen the Power-On-Delay (**POD**) after every reboot by saving the value in non-volatile memory.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x14	Not Used	Not Used
0x15	Delay Time	Delay Time

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x14	Delay Time	Delay Time
0x15	Not Used	Not Used

Notes:

If the time read is 0x0000 using command 0x14 then there is not a new Arm time in the non-volatile memory. The new Arm time can range from 0x0001 to 0xFFFF seconds. If the value is set to zero (0x0000) on command 0x15 (write) then any Arm time in non-volatile memory will be erased and the board will go back to using the 2.5-minute delay. A non-zero value will be stored and then used as the Arm time on future resets and power-on. *Be careful not to make it too short.*

3.8 Commands: 0x18 & 0x19 - Read / Write Current Watchdog Time

When you use the Watchdog for the first time it will use a countdown time set by the DIP switches (section 4.4) or it will use a stored value if programmed. This command allows you override the switch or stored setting to shorten or lengthen the delay. See the next section for storing a new delay time.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x18	Not Used	Not Used
0x19	Delay Time	Delay Time

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x18	Delay Time	Delay Time
0x19	Not Used	Not Used

Notes:

The delay time can range from 0x0001 to 0xFFFF seconds. A value of zero (0x0000) allows you to return to the time set on the DIP switch or the stored value in non-volatile memory if it exists. The new time will be activated at the next re-trigger of the Watchdog. A read will return 0xFFFF if the board is not yet armed.

Be careful using low times (1 or 2 seconds) if your application runs as a low priority task.

3.9 Command: 0x1A - Read Holding Register Watchdog Time

The Serial PC Watchdog uses an internal register to store the watchdog time that will be reloaded into the countdown timer every time the board it "tickled"

Typically this value will be determined by the DIP Switch setting. It can be replaced by using the Write Current Watchdog Time (0x19 in the prior section) or it can be replaced with a value from the non-volatile memory which overrides the DIP Switch values at power up.

Send Command:

Port 6 CMD	**Port 5** MSB	**Port 4** LSB
0x1A	Not Used	Not Used

WDog Response:

Port 6 RESP	**Port 5** MSB	**Port 4** LSB
0x1A	Hold Time	Hold Time

3.10 Commands: 0x1C & 0x1D - Read / Save Non-Volatile Watchdog Time

These commands allow you to check and shorten / lengthen the watchdog time for every reboot by saving the value in non-volatile memory.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x1C	Not Used	Not Used
0x1D	Delay Time	Delay Time

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x1C	Delay Time	Delay Time
0x1D	Not Used	Not Used

Notes:

If the time read is 0x0000 using command 0x1C then there is not a new watchdog time in the non-volatile memory. The new watchdog time can range from 0x0001 to 0xFFFF seconds. If the value is zero (0x0000) on command 0x1D then any watchdog time in non-volatile memory will be erased and the board will go back to using the DIP switch settings. A non-zero value will be stored and then used as the watchdog time on future resets and power-ups. The new time will also be activated next time the watchdog board is re-triggered.

3.11 Command: 0x20 - Read Re-trigger Count

Every time the watchdog gets re-triggered by the re-trigger command (0x02) it will increment a 16 bit counter. This command allows you to read the counter. If you set the flags in the LSB to a non-zero value then the counter will be cleared after the value has been returned.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x20	Not Used	Flag

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x20	Count MSB	Count LSB

3.12 Command: 0x30 – Enable / Disable Watchdog

When this command is received it will disable the watchdog if the MSB is 0xA5 and the LSB ix 0xC3. If you send any other values then the watchdog will be enabled. The flag returned will be non-zero if the watchdog is enabled.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x30	0xA5	0xC3

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x30	Not Used	Flag

3.13 Command: 0x40 – Power Module (External Only)

This command allows you to turn the power module on and off.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x40	Not Used	Bits

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x40	Not Used	Bits

Notes:

The send LSB data byte is a bit field defined as:

D0: Set to one – turn power module on
D1: Set to one – turn power module off
D2-7: Not used – set to zero.

The response data byte is a bit field defined as:

D0: Set to one if the module was on before new bits applied
D1: Set to one if the module was off before new bits applied
D2-7: Not used - will read as zero.

3.14 Commands: 0x50 & 0x51 - Read / Write Relay Control (Internal Only)

This command provides a method to control the auxiliary relay on the board. You can also invert the operation of the relay at power-up and save the setting. When the relay is inverted it will be turned on within a few hundred milli-seconds of the PC being powered on.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x50	Not Used	Not Used
0x51	Not Used	Bits

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x50	Not Used	Bits
0x51	Not Used	Bits

Notes:

The data byte in the send command (0x51) is a bit field defined as:

D0: Set to one (1) to turn relay on.
D1: If set to one the auxiliary relay will pulse for the same length of time as the reset relay (see command 0x88). This option will be stored in non-volatile memory.
D2: If set to one the auxiliary relay will latch on after the first watchdog reset and stay on. You can turn off the relay with bit D0. This option will be stored in non-volatile memory.
D3-6: Not used - write zeros for future compatibility.
D7: Set to one (1) to store relay inversion setting in non-volatile memory. Relay will turn on a few hundred milli-seconds after the PC powers up. This is only done at power-up. You can turn the relay off with bit D0. Setting bit D1 will also turn the relay off after the first watchdog reset.

The data byte in the command response (0x50 or 0x51) is also a bit field defined as:

D0: Set to one (1) means auxiliary relay is on.
D1: Auxiliary relay set for pulse at watchdog reset.
D2: Auxiliary relay set for latch on at watchdog reset.
D3-6: Not used - read zeros for future compatibility.
D7: Set to one (1) means non-volatile inversion setting is active.

3.15 Commands: 0x58 & 0x59 - Read / Write Temperature Offset

This command allows you to increase the temperature trip points on the watchdog board. These commands allow you to set the offsets up to 31 (0x1f). If the offset is used then it will be active every time the PC powers up.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x58	Not Used	Not Used
0x59	Not Used	Offset

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x58	Not Used	Offset
0x59	Not Used	Not Used

Notes:

The offset here is in degrees C and can range from 0x00 to 0x1F. Values higher will be corrected modulo 32. A non-zero value is stored in the non-volatile memory.

3.16 Commands: 0x5E Buzzer Control (Internal Only)

This command provides a method to control the buzzer on the board. The settings in this command are only in effect while the board is powered. See the next command for non-volatile buzzer options.

Send Commands:

Port 6 CMD	**Port 5** MSB	**Port 4** LSB
0x5E	Buzz Time	Bits

WDog Responses:

Port 6 RESP	**Port 5** MSB	**Port 4** LSB
0x5E	Buzz Time	Bits

On a command send the Buzz Time value along with the Enable bit: D0 is used to start the buzzer. The time is in 10mS tics so a value of 100 results in a 1 second buzzer. A value of zero turns off the buzzer. The value 255 (0xff) is special and it results in a continuous buzzer that must be turned off with another command.

On a command return the Buzz Time will be non-zero if the buzzer is still on.

Notes:

The LSB data byte in the send command is a bit field defined as:

D0: Set to one to Enable the Buzz Time value.
D1: Set to one to disable, zero to enable all further buzzer sound. See D7.
D4-6: Not used - write zeros for future compatibility.
D7: This bit acts as en Enable for D1. If D7 is clear then D1 is ignored.

The LSB data byte in the command response is also a bit field defined as:

D0: Set to one means buzzer is on.
D1: Set to one means buzzer is disabled.
D2-7: Not used - read zeros for future compatibility.

3.17 Commands: 0x5F NV Buzzer Control (Internal Only)

This command provides a method to control the buzzer on the board. The settings in this command are saved in non-volatile (NV) memory and are used every time the boards powers up.

Send Commands:

Port 6 CMD	**Port 5** MSB	**Port 4** LSB
0x5F	Buzz Time	Bits

WDog Responses:

Port 6 RESP	**Port 5** MSB	**Port 4** LSB
0x5F	Buzz Time	Bits

On a command send the Buzz Time value along with the Enable bit: D0 is used to save a new buzzer time to used each time the boards powers up or reboots the PC. The time is in 10mS tics so a value of 100 results in a 1 second buzzer. A value of zero resets the board back to its default value of 0.6 seconds. The value 255 (0xff) is special and it results in a continuous buzzer that must be turned off with a 0x5E Buzzer command.

On a command return the Buzz Time will be non-zero if there is a value in NV memory.

Notes:

The LSB data byte in the send command is a bit field defined as:

D0: Set to one to Enable the Buzz Time value to be stored.
D1: Set to one to disable, zero to enable all further buzzer sound. See D7.
D2-6: Not used - write zeros for future compatibility.
D7: This bit acts as en Enable for D1. If D7 is clear then D1 is ignored.

The LSB data byte in the command response is also a bit field defined as:

D0: Set to one means the NV value is non-zero.
D1: Set to one means buzzer is disabled always.
D2-7: Not used - read zeros for future compatibility.

3.18 Command: 0x80 - Reset PC

Use this command carefully. It will reset your PC.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x80	Flag	Delay Time

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x80	Delay Time	Bits

Notes:

The Flag can be one of 3 values:

- 0xA5 – Reset the board using the delay time in the LSB. If the delay time is set to zero the reset will occur immediately. The delay time can be as high as 255 (0xFF) seconds.
- 0xC3 – Cancel the pending reset if there was a prior delay time used.
- 0x39 – Read Back the current delay time left in MSB.

The LSB data byte in the command response is a bit field defined as:

D0: Set to one means Reset Accepted. Delay Time will be echoed in MSB
D1: Set to one means the Cancel Reset was accepted.
D2: Set to one means a reset was not pending if you tried a Read Back time.
D3-6: Not used - read zeros for future compatibility.
D7: Set to one means board is still in POD. Commands ignored.

The board must be ARMed to accept this command. If the bit D7 is set then POD is still active. Either wait for POD to end or shorten the current POD time.

Even if the Watchdog board has been disabled via command 0x30, this command will still work!

3.19 Command: 0x84 - Get / Clear Reset Count

When the watchdog board powers up it will set a counter to zero. Every time it resets the PC this count will increase by one. Use this command to get the count. If the Clear Flag is non-zero then the board will clear the count after it has been returned. If the count is cleared to zero then the bottom LED will also be turned off.

Send Command:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x84	Not Used	Clear Flag

WDog Response:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x84	Not Used	Count

Notes:

The count ranges from 0x00 to 0x80 (128). When the counter hits the upper limit it will stop rather than rollover.

3.20 Commands: 0x88 & 0x89 - Read / Write Reset Relay Pulse

The default mode for the Watchdog is to pulse a reset relay (which simulates pushing the reset button) for 2.5 seconds. This value works for most computers, however some machines may require longer or shorter values. This command allows you to tailor the pulse time by selecting a delay in 50mS tics.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0x88	Not Used	Not Used
0x89	Not Used	Delay Tics

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0x88	Not Used	Delay Tics
0x89	Not Used	Not Used

Notes:

The delay time can range from 0x01 to 0xFF tics. A value of zero (0x00) allows you to return to the default 2.5 second pulse. The new time will be activated at the next re-boot of the PC and it will be stored in non-volatile memory.

Be careful using low times (1 or 2 tics) since these may be too short to generate valid resets on the PC.

3.21 Commands: 0xC0 & 0xC1 - Read / Write User ID Data

Eight (8) bytes of the non-volatile memory have been reserved for the user to store their own ID data. These bytes could have many uses with one of them being a software security mechanism. A user could make their software refuse to run unless there was a watchdog board in the system with the correct code in the memory.

Send Commands:

BYTE 0 CMD	**BYTE 1** MSB	**Byte 2** LSB
0xC0	Offset	Not Used
0xC1	Offset	User Data

WDog Responses:

BYTE 0 RESP	**BYTE 1** MSB	**Byte 2** LSB
0xC0	Not Used	NV Data
0xC1	Not Used	Not Used

Notes:

The offset should be a number between 0 and 7. Higher values will be corrected to modulo 8. Data can be any value from 0x00 to 0xFF.

3.22 Invalid Commands

If you send a command is not implemented in the firmware on the board then the board will respond with zeros in all three response bytes.

3.23 Special Diagnostics - 0xE0 to 0xFF (Do Not Use)

These commands are used for internal testing purposes.

*Do not use commands higher than **0xE0**. They are for testing purposes only and will very likely produce undesirable results.*