**R179 HIDAC BTE Software Requirements**

**Author**: Dave Smail

**Date**: 11/26/2018

**Revision**: 1

**Acronyms**:

RAM – Random Access Memory

NVRAM – Non Volatile Random Access Memory

RTC – Real Time Clock

**Requirements**:

1. The software shall blink the yellow and green LED alternately so that the user knows that the R179 HIDAC BTE software is executing
2. The software shall report the software version via the serial port when acknowledging a command from the PC.
3. After power on or reset, all memory accesses are by default disabled
4. The software shall have the capability to:
   1. Starting at FLASH address 0x110000, 1 or more locations from FLASH memory that contain a known set of data values (see Table 1 below) are to be verified when the FLASH offset is enabled via the serial port.
   2. At only RAM address 0x210000, 1 or more table data shall be written, read and verified (see Table 2 below).
   3. At only NVRAM address 0x810000, 1 or more table data shall be written, read and verified (see Table 3 below).
   4. At only RTC address 0x301000, 1 or more table data shall be written, read and verified (see Table 4 below).
5. For requirement 4), for any failure, the software shall report the failure via the serial port. This shall include the expected value and the actual value.
6. For requirement 4), when a test passes, only the expected value shall be reported
7. For requirement 4), all tests are “1 shots”; the PC will have to reenable any desired test(s).
8. The software shall have the capability to peek and poke and memory location via the serial port
9. FLASH
   1. Via a serial port command, the software shall be able to enable any specific read
   2. Via a serial port command, the software shall be able to enable all reads
10. RAM
    1. Via a serial port command, the software shall be able to enable any specific write/read data value
    2. Via a serial port command, the software shall be able to enable all write/read data values
11. NVRAM
    1. Via a serial port command, the software shall be able to enable any specific write/read data value
    2. Via a serial port command, the software shall be able to enable all write/read data values.
12. RTC
    1. Via a serial port command, the software shall be able to enable any specific write/read data value
    2. Via a serial port command, the software shall be able to enable all write/read data values

**NOTES:**

1. Since all code executes out of FLASH, when reading known FLASH known values from table A, there will be other FLASH accesses while reading this data.
2. Every effort will be made when writing/reading known values to RAM to utilize internal C167 memory so as not to have other RAM accesses occurring while writing/reading known values to RAM.

**Supplemental**

Table 1 lists the FLASH location and data stored at that location

|  |  |  |
| --- | --- | --- |
| **Index** | **Address** | **Data** |
| 00 | 0x110000 | 0x0000 |
| 01 | 0x110002 | 0x0001 |
| 02 | 0x110004 | 0x0002 |
| 03 | 0x110006 | 0x0004 |
| 04 | 0x110008 | 0x0008 |
| 05 | 0x11000A | 0x0010 |
| 06 | 0x11000C | 0x0020 |
| 07 | 0x11000E | 0x0040 |
| 08 | 0x110010 | 0x0080 |
| 09 | 0x110012 | 0x0100 |
| 0A | 0x110014 | 0x0200 |
| 0B | 0x110016 | 0x0400 |
| 0C | 0x110018 | 0x0800 |
| 0D | 0x11001A | 0x1000 |
| 0E | 0x11001C | 0x2000 |
| 0F | 0x11001E | 0x4000 |
| 10 | 0x110020 | 0x8000 |
| 11 | 0x110022 | 0xFFFF |
| 12 | 0x110024 | 0xFFFE |
| 13 | 0x110026 | 0xFFFD |
| 14 | 0x110028 | 0xFFFB |
| 15 | 0x11002A | 0xFFF7 |
| 16 | 0x11002C | 0xFFEF |
| 17 | 0x11002E | 0xFFDF |
| 18 | 0x110030 | 0xFFBF |
| 19 | 0x110032 | 0xFF7F |
| 1A | 0x110034 | 0xFEFF |
| 1B | 0x110036 | 0xFDFF |
| 1C | 0x110038 | 0xFBFF |
| 1D | 0x11003A | 0xF7FF |
| 1E | 0x11003C | 0xEFFF |
| 1F | 0x11003E | 0xDFFF |
| 20 | 0x110040 | 0xBFFF |
| 21 | 0x110042 | 0x7FFF |

**Table 1**

Table 2 lists the data written to and read from RAM at address 0xTBD

|  |  |
| --- | --- |
| **Index** | **Data** |
| 00 | 0x0000 |
| 01 | 0x0001 |
| 02 | 0x0002 |
| 03 | 0x0004 |
| 04 | 0x0008 |
| 05 | 0x0010 |
| 06 | 0x0020 |
| 07 | 0x0040 |
| 08 | 0x0080 |
| 09 | 0x0100 |
| 0A | 0x0200 |
| 0B | 0x0400 |
| 0C | 0x0800 |
| 0D | 0x1000 |
| 0E | 0x2000 |
| 0F | 0x4000 |
| 10 | 0x8000 |
| 11 | 0xFFFF |
| 12 | 0xFFFE |
| 13 | 0xFFFD |
| 14 | 0xFFFB |
| 15 | 0xFFF7 |
| 16 | 0xFFEF |
| 17 | 0xFFDF |
| 18 | 0xFFBF |
| 19 | 0xFF7F |
| 1A | 0xFEFF |
| 1B | 0xFDFF |
| 1C | 0xFBFF |
| 1D | 0xF7FF |
| 1E | 0xEFFF |
| 1F | 0xDFFF |
| 20 | 0xBFFF |
| 21 | 0x7FFF |

**Table 2**

Table 3 lists the data written to and read from NVRAM at address 0xTBD

|  |  |
| --- | --- |
| **Index** | **Data** |
| 00 | 0x0000 |
| 01 | 0x0001 |
| 02 | 0x0002 |
| 03 | 0x0004 |
| 04 | 0x0008 |
| 05 | 0x0010 |
| 06 | 0x0020 |
| 07 | 0x0040 |
| 08 | 0x0080 |
| 09 | 0x0100 |
| 0A | 0x0200 |
| 0B | 0x0400 |
| 0C | 0x0800 |
| 0D | 0x1000 |
| 0E | 0x2000 |
| 0F | 0x4000 |
| 10 | 0x8000 |
| 11 | 0xFFFF |
| 12 | 0xFFFE |
| 13 | 0xFFFD |
| 14 | 0xFFFB |
| 15 | 0xFFF7 |
| 16 | 0xFFEF |
| 17 | 0xFFDF |
| 18 | 0xFFBF |
| 19 | 0xFF7F |
| 1A | 0xFEFF |
| 1B | 0xFDFF |
| 1C | 0xFBFF |
| 1D | 0xF7FF |
| 1E | 0xEFFF |
| 1F | 0xDFFF |
| 20 | 0xBFFF |
| 21 | 0x7FFF |

**Table 3**

Table 4 lists the data written to and read from RTC at address 0xTBD

|  |  |
| --- | --- |
| **Index** | **Data** |
| 00 | 0x00 |
| 01 | 0x01 |
| 02 | 0x02 |
| 03 | 0x04 |
| 04 | 0x08 |
| 05 | 0x10 |
| 06 | 0x20 |
| 07 | 0x40 |
| 08 | 0x80 |
| 09 | 0xFF |
| 0A | 0xFE |
| 0B | 0xFD |
| 0C | 0xFB |
| 0D | 0xF7 |
| 0E | 0xEF |
| 0F | 0xDF |
| 10 | 0xBF |
| 11 | 0x7F |

**Table 4**

**Serial Communication Specification / Protocol**

1. All serial communication to and from the software shall use the following parameters:
   1. 19200 baud rate, 8 bits, 1 start, 1 stop, no parity
2. All characters will be ASCII
3. The HiDAC software will echo back each character received
4. After an entire valid command is received, the following will be sent from the HiDAC software:
   1. **<OK,n.n>** where “n.n” is the software version
5. After an entire invalid command is received, the following will be sent from the HiDAC software:
   1. **<INV,n.n>** where “n.n” is the software version
6. To enable all FLASH reads, the following shall be sent to the HiDAC
   1. **<FLA,FF>**
7. To enable a specific FLASH read, the following shall be sent to the HiDAC
   1. **<FLA,nn>** where “nn” is the index number of the FLASH location found in Table 1
8. To enable all RAM reads and writes, the following shall be sent to the HiDAC
   1. **<RAM,FF>**
9. To enable a RAM read and write of specific data , the following shall be sent to the HiDAC
   1. **<RAM,nn>** where “nn” is the index found in Table 2
10. To enable all NVRAM reads and writes, the following shall be sent to the HiDAC
    1. **<NVR,FF>**
11. To enable a RAM read and write of specific data , the following shall be sent to the HiDAC
    1. **<NVR,nn>** where “nn” is the index found in Table 3
12. To enable all RTC reads and writes, the following shall be sent to the HiDAC
    1. **<RTC,FF>**
13. To enable a RTC read and write of specific data , the following shall be sent to the HiDAC
    1. **<RTC,nn>** where “nn” is the index found in Table 4
14. For requirements 6 – 13, when a test is enabled, it executes 1 time only regardless of pass or fail condition.
15. For requirements 6 – 13, when a test is enabled and passes, the following is sent back to the PC via the serial port
    1. **<FLA,OK,eeee>** where eeee is the expected value.
    2. **<RAM,OK,eeee>** where eeee is the expected value.
    3. **<NVR,OK,eeee>** where eeee is the expected value.
    4. **<RTC,OK,00ee>** where eeee is the expected value. NOTE: RTC accesses are 8 bits, thus the 2 leading 0’s.
16. For requirements 6 – 13, when a test is enabled and fails, the following is sent back to the PC via the serial port
    1. **<FLA,FAIL,eeee,aaaa>** where eeee is the expected value hex and aaaa is the actual value (hex).
    2. **<RAM,FAIL,eeee,aaaa>** where eeee is the expected value hex and aaaa is the actual value (hex).
    3. **<NVR,FAIL,eeee,aaaa>** where eeee is the expected value hex and aaaa is the actual value (hex).
    4. **<RTC,FAIL,nn,00ee,00aa>** where eeee is the expected value hex and aaaa is the actual value (hex). NOTE: RTC accesses are 8 bits, thus the 2 leading 0’s.
17. The software shall give the ability to write (poke) any memory location via the following serial command from the PC
    1. **<POS,aaaaaa,dddd>** where aaaaaaa is a hexadecimal 24 bit address and dddd is a 16 bit hexadecimal data.
    2. The software shall ignore pokes to FLASH memory,
    3. The software shall disable pokes to RAM locations that are used by the software
    4. The following is a list of valid poke locations
       1. 0x210000 – 0x21FFFF (RAM)
       2. 0x300000 – 0x30\_TBD (RTC)
       3. 0x800000 – 0x80\_TBD (NVRAM)
    5. If the address is valid, the software shall respond with **<OK,n.n>** where n.n is the software version
    6. If the address is NOT valid, the software shall respond with **<INV,n.n>** where n.n is the software version.

NOTE: For RTC accesses, the upper 2 nibbles of dddd are ignored

1. The software shall give the ability to read (peek) any memory location via the following serial command from the PC
   1. **<PES,aaaaaa>** where aaaaaaa is a hexadecimal 24 bit address.
   2. The following is a list of valid peek locations
      1. 0x100000 – 0x13FFFF (FLASH)
      2. 0x200000 – 0x21FFFF (RAM)
      3. 0x300000 – 0x30\_TBD (RTC)
      4. 0x800000 – 0x80\_TBD (NVRAM)
   3. If the address is valid, the software shall respond with **<OK,n.n>** where n.n is the software version followed by **<PES,aaaaaa,dddd>** where aaaaaaa is a hexadecimal 24 bit address and dddd is a 16 bit hexadecimal data. NOTE: For RTC accesses, the value is 8 bits and thus will always contain 2 leading 0’s.
   4. If the address is NOT valid, the software shall respond with **<INV,n.n>** where n.n is the software version.
2. The software shall give the ability to write (poke) continuously any memory location via the following serial command from the PC
   1. **<POC,aaaaaa,dddd>** where aaaaaaa is a hexadecimal 24 bit address and dddd is a 16 bit hexadecimal data.
   2. The software shall ignore pokes to FLASH memory,
   3. The software shall disable pokes to RAM locations that are used by the software
   4. The following is a list of valid poke locations
      1. 0x210000 – 0x21FFFF (RAM)
      2. 0x300000 – 0x30\_TBD (RTC)
      3. 0x800000 – 0x80\_TBD (NVRAM)
   5. If the address is valid, the software shall respond with **<OK,n.n>** where n.n is the software version
   6. If the address is NOT valid, the software shall respond with **<INV,n.n>** where n.n is the software version.

NOTE: For RTC accesses, the upper 2 nibbles of dddd are ignored

1. The software shall give the ability to read (peek) continuously any memory location via the following serial command from the PC
   1. <PEC,aaaaaa> where aaaaaaa is a hexadecimal 24 bit address.
   2. The following is a list of valid peek locations
      1. 0x100000 – 0x13FFFF (FLASH)
      2. 0x200000 – 0x21FFFF (RAM)
      3. 0x300000 – 0x30\_TBD (RTC)
      4. 0x800000 – 0x80\_TBD (NVRAM)
   3. If the address is valid, the software shall respond with **<OK,n.n>** where n.n is the software version. All subsequent continuous peeks will issue **<PE,aaaaaa,dddd>** where aaaaaaa is a hexadecimal 24 bit address and dddd is a 16 bit hexadecimal data to the PC. NOTE: For RTC accesses, the value is 8 bits and thus will always contain 2 leading 0’s.
   4. If the address is NOT valid, the software shall respond with **<INV,n.n>** where n.n is the software version.
2. The software shall give the ability to stop continuous reads (peeks) via the following serial command from the PC
   1. **<PEK>**
   2. The software shall respond with **<OK,n.n>** where n.n is the software version
3. The software shall give the ability to stop continuous writes (pokes) via the following serial command from the PC
   1. **<POK>**
   2. The software shall respond with **<OK,n.n>** where n.n is the software version