Using a Nohau BDM Emulator to Debug an HCS12 application – Improvements Advantages and Disadvantages

By: Doron Fael, Nohau

The Nohau low cost BDM Emulator for the HC12 and HCS12 family of micro-controllers has become a powerful debugging tool for the HC12 family and the HCS12 family in particular, over the last year. Nowadays, serial debug interfaces such as the BDM have developed so much, that they offer many features that were not possible until 5 years ago. The BDM emulator offers the best price/performance – including High Level and Assembly Level support for all HC12 and HCS12 derivatives, starting from the A4 and B32, through the MC9S12Dxxx family, and MC9S12Hxxx series of micro-controllers, of micro-controllers, and to newer HCS12 families.

Nohau's main specialty and only business is designing and building emulators. This fact underlies the high quality and ease-of-use of the Nohau BDM emulator, compared to other reputable companies which design BDM emulators as a side product. Maintaining, improving and implementing support for new derivatives with the BDM emulator.

The BDM Emulator is a serial emulation device, which connects to a target board with a target HC12 or HCS12 micro-controller using a simple 6-pin connector, and communicates to it serially using a Motorola defined communication protocol. The Nohau BDM Emulator also communicates to a PC computer and the Seehau user-interface program using one of the following 3 connections: USB, LPT printer port, or a dedicated ISA card (See Figure 1).

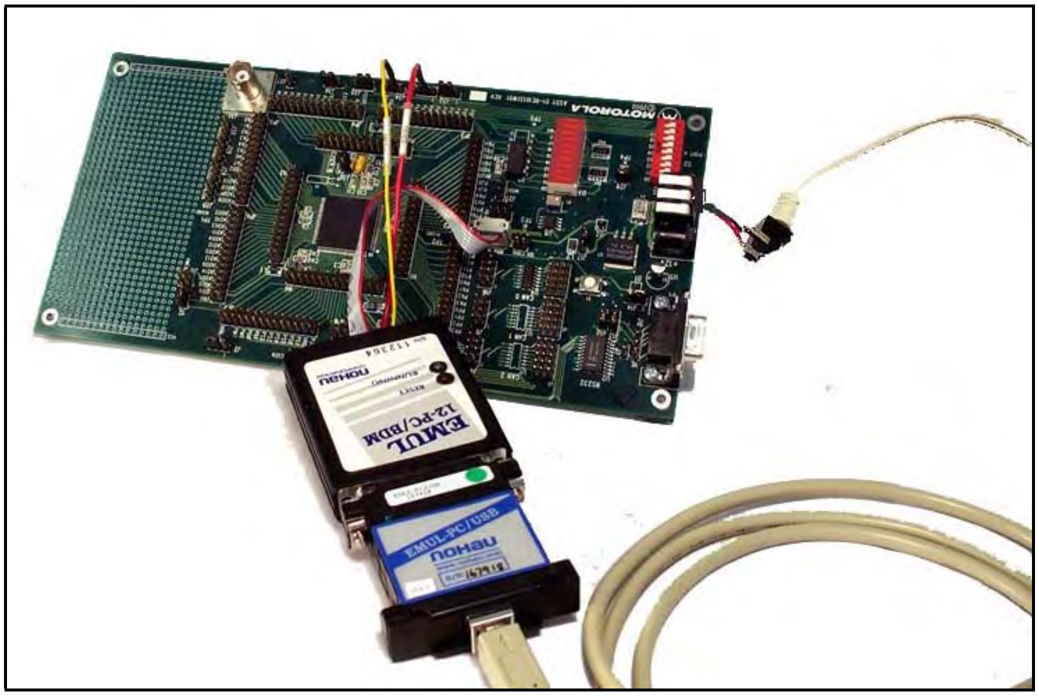
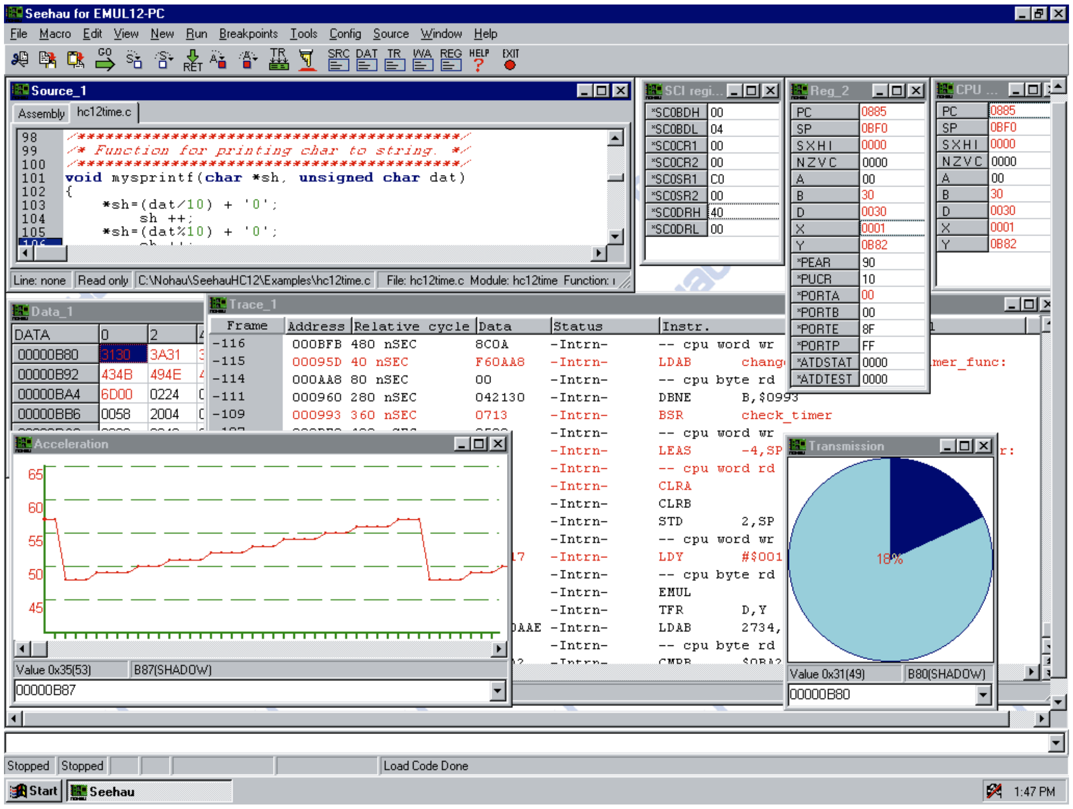


Figure 1 - the Nohau BDM emulator connected to a Motorola MC9S12DP256 Evaluation Board on one side, and to the USB communication device to the PC on the other side.

When comparing the BDM emulators available in the market, you will find there are BDM emulators that sell for less and even much less than the Nohau BDM emulator. However, a short comparison between the features of various BDM emulators available in the market will reveal the need for the special features found on the Nohau BDM. A description of these features follows.

The Nohau BDM supports both C level and Assembly level debugging. This includes displaying the C source with the line that has been reached, clearly marked, when code execution stops. Setting breakpoints in the source code is simple - by a mouse click. Evaluation of C variables and structures is also easy - by simply holding the mouse over a variable name. For more complex C expressions the Inspect and Watch windows are available too. The Seehau user-interface software is used as the front end for the BDM emulator. The same Seehau user-interface is used for the full-featured emulator as well. (Figure 2).

By contrast very low-price BDM emulators available in the market, do not support C level debugging, and don't have a powerful and user friendly user interface.



```
/*******************************************/
/* Function for printing char to string. */
/*******************************************/
void mysprintf(char *sh, unsigned char dat)
{
    *sh=(dat/10) + '0';
        sh ++;
    *sh=(dat%10) + '0';
```

Line: none | Read only | C:\Nohau\SeehauHC12\Examples\hc12time.c | File: hc12time.c Module: hc12time Function:

DATA	0	2
00000B80	3130	3A31
00000B92	434B	494E
00000BA4	6D00	0224
00000BB6	0058	2004

```
Frame  Address Relative cycle  Data     Status   Instr.
-116   000BFB  480 nSEC        8C0A     -Intrn-  -- cpu word wr
-115   00095D  40 nSEC         F60AA8   -Intrn-  LDAB     chang
-114   000AA8  80 nSEC         00       -Intrn-  -- cpu byte rd
-111   000960  280 nSEC        042130   -Intrn-  DBNE     B,$0993
-109   000993  360 nSEC        0713     -Intrn-  BSR      check_timer
                                        -Intrn-  -- cpu word wr
                                        -Intrn-  LEAS     -4,SP
                                        -Intrn-  -- cpu word rd
                                        -Intrn-  CLRA
                                        -Intrn-  CLRB
                                        -Intrn-  STD      2,SP
                                        -Intrn-  -- cpu word wr
                                        -Intrn-  LDY      #$001
                                        -Intrn-  -- cpu byte rd
                                        -Intrn-  EMUL
                                        -Intrn-  TFR      D,Y
                                        -Intrn-  LDAB     2734,
                                        -Intrn-  -- cpu byte rd
```

SCI regi...	
*SC0BDH	00
*SC0BDL	04
*SC0CR1	00
*SC0CR2	00
*SC0SR1	C0
*SC0SR2	00
*SC0DRH	40
*SC0DRL	00

Reg_2	
PC	0885
SP	0BF0
SXHI	0000
NZVC	0000
A	00
B	30
D	0030
X	0001
Y	0B82
*PEAR	90
*PUCR	10
*PORTA	00
*PORTB	00
*PORTE	8F
*PORTP	FF
*ATDSTAT	0000
*ATDTEST	0000

CPU ...	
PC	0885
SP	0BF0
SXHI	0000
NZVC	0000
A	00
B	30
D	0030
X	0001
Y	0B82

Value 0x35(53) | B87(SHADOW) | 00000B87

Value 0x31(49) | B80(SHADOW) | 00000B80

Stopped | Stopped | Load Code Done

Figure 2 – The Seehau user interface program is the front end for the Nohau BDM and Full Featured emulators. It displays the C source code in colors for easy reading (upper left). Memory locations and variables can be viewed in many formats, including graphical ones such as graphs over time, pie diagrams etc. Registers and memory locations that changed their value since the last CPU-step show in red, for easy spotting of changes that took place during the last code execution. All to make it as easy as possible to debug code.

The Nohau BDM pod (the hardware) has proven to be powerful and up-to-date. The same original BDM hardware that was released to support the 68HC812A4 five years ago, is still used today to support the most recent HCS12 derivatives. No hardware upgrades are needed. Download the latest Seehau software from www.icetech.com, install it, and all the HC12 and HCS12 devices are supported to their maximum speed. Many other BDM emulators had to be reprogrammed or exchanged to support new devices such as the HCS12 family to its 25MHz maximum bus rate.

In the lab the BDM emulator even worked with an MC9S12DP256 device running at 32MHz bus-rate. Routines are in place to support an even higher speed operation if and when Motorola release even faster HC12 devices.

The BDM pod can interface to targets powered by the VDDX range from 3V to 5.5V.

Fast Erase and Programming of the internal HC12 Flash and EEPROM is built in to the Seehau user interface. The code under debug is programmed to the internal HC12 Flash, and executes from there. Software breakpoints cannot be used when the code is in Flash, so 2 internal HC12 Hardware Breakpoints are automatically used to allow code debugging.

The BDM emulator allows the use of the internal HC12 PLL to up the speed right after Reset. This is achieved by specifying the required final PLL register values, so Seehau can implement them automatically after Reset. This ability is somewhat limited on the BDM emulator, as the communication between the target HC12 and the BDM Emulator is asynchronous according definition of the BDM communication protocol. As a result, both ends need to know the communication rate at any given time. Unfortunately, changing the target HC12 bus speed also affects the BDM communication rate. Thus, the only way to allow the use of the internal HC12 PLL, is the one described here, to allow the BDM to track the changed communication speed when it happens. The Nohau full emulator on the other hand, is more powerful in this field of the PLL utilization, and allows speed changes as many times and as often as required with no limitations and no need to specify the PLL register values.

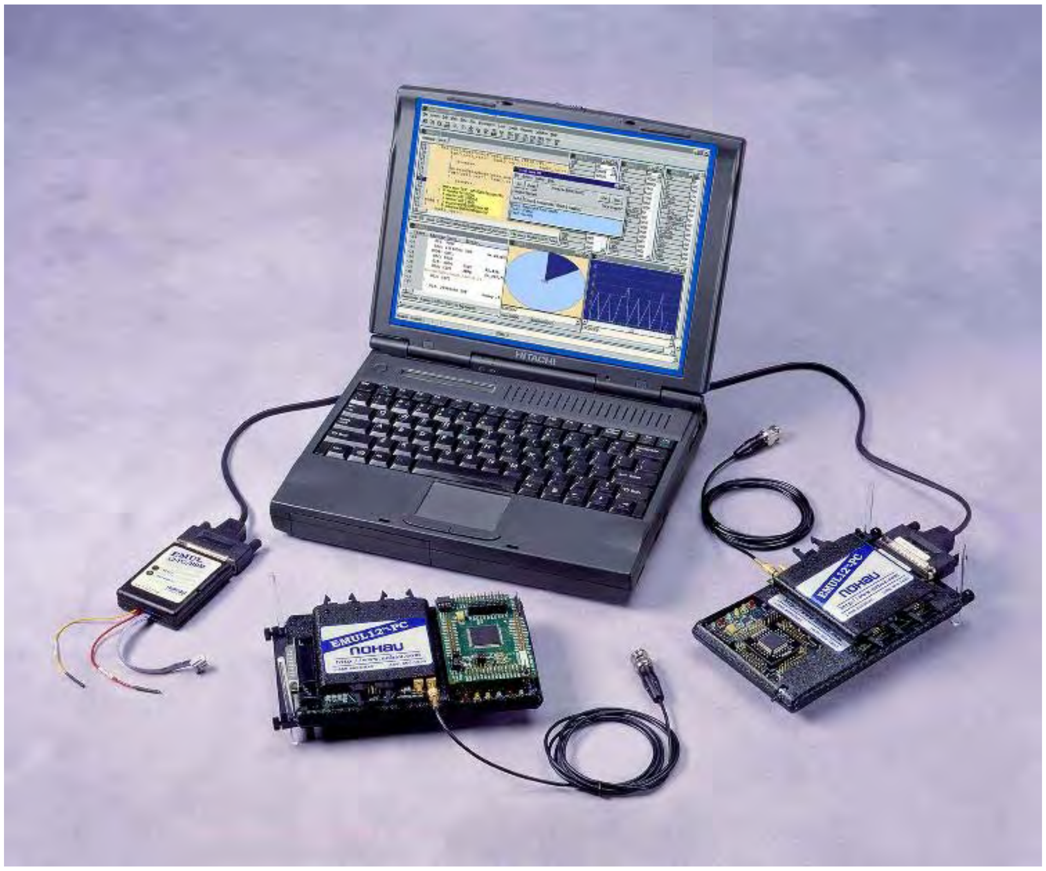


Figure 3 – various HC12 Emulators available from Nohau. On the left the BDM Emulator, on the middle the Full-Featured Emulator for the MC9S12Dxxx family, and on the right the Full-Featured emulator for the 68HC912B family. All 3 emulators are hand held, and can go anywhere your laptop can.

Another interesting ability of the BDM emulator is to view changes to memory locations, and memory-mapped registers while the HC12 code is running, and without having to stop it. This is achieved by selecting the RUN-TIME Data window, and using a powerful feature of the HC12 BDM interface that allows memory operations to be injected to the bus stream during code execution. Using this ability also allows <u>writing</u> to any memory location or memory-mapped register, during HC12 code execution.

BDM vs. Full-Featured Emulators

The above discussion brings up the following question:

If the BDM emulator is so good and powerful, why does anybody need a full-featured HC12 emulator? Can't you do all debugging using a good BDM emulator, and save the extra cost of a full-featured emulator?

The answer is that it depends.

For many simple applications, the BDM is sufficient and good to get the job done.

On the other hand, the BDM lacks some crucial features, required for many complex applications involving a lot of code (32KByte and up). These features are found only in a full-featured emulator, and can make the difference between a successful completion and an unsuccessful completion of a project.

The lack of these crucial features on the BDM is because of the limited nature of the on chip HC12 BDM module, and is not because of a limited nature of the BDM emulator hardware or software. These limitations exist on all BDM emulators.

Big multi-engineer projects often employ one full-featured emulator for every two BDM emulators. The philosophy behind this choice is that the elusive bugs will be solved with the help of the full-featured emulator, while the routine code debug can be done with either the BDM or the full-featured emulator.

For smaller one-engineer projects, the following advantages of the full-featured emulator need to be considered before choosing which emulator type to use:

1. The most powerful feature of the full-featured emulator is the trace. The trace can record the executed instruction and/or the bus activity that takes place from the time code execution is resumed until it is stopped again. Analyzing this recorded information can reveal why and how the program execution has reached unexpected code. The trace can also be used to debug applications without ever stopping the HC12 from running. This is sometimes required for sensitive applications that must always have a timer running in order not to damage some piece of hardware. The trace has a powerful set of triggers and a filter to specify what needs to be recorded, and what sequence of events should lead the trace to stop and display its recorded content for debugging.
2. Debugging of more complicated code which involves frequent speed changes, debugging of Reset cycles (internal or external triggered), or debugging of power down modes cycling, is very limited with a BDM emulator, and requires a full-featured emulator.
3. Complicated code debug using more than 2 hardware breakpoints requires the use of a full-featured emulator, which has an unlimited number of both hardware and software breakpoints.
4. A full-featured emulator has emulation RAM that can replace the internal HC12 FLASH and EEPROM for the duration of the debugging. RAM memory is more flexible than a non-volatile memory, and makes it easier to debug the code, without having to reprogram the internal FLASH every time the code is modified, and reloaded.
5. The full-featured emulator may work in stand-alone mode with no target board. This allows for code debug to start even before the target board is available or completed.

For more information go to:
http://www.icetech.com/emul12pc.html and
http://www.icetech.com/parts_lists/em12pc.pdf
Email: support@icetech.com
Or call Nohau at 650.375.0409
or USA toll free 800.686.6428