

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

Introduction

µTasker is an operating system designed especially for embedded applications where a tight control over resources is desired along with a high level of user comfort to produce efficient and highly deterministic code.

The operating system is integrated with TCP/IP stack and important embedded Internet services alongside device drivers and system specific project resources.

This document informs how to convert a DEMO9S12NE64* (the low cost demo board from Freescale™) into a quality BDM (Background Debug Module) programming tool. It requires a simple cable with three wires to connect to the BDM of another MC9S12NE64 and a µTasker based software which allows controlling various operations from a Browser via Ethernet (or even by Internet). The project demonstrates various inbuilt features in the µTasker.

It is a useful tool for anyone just finished with initial evaluation and moving to his/her own MC9S12NE64 based PCBs. The DEMO board has a serial monitor pre-installed which allows comfortable software downloads and debugging. Fresh devices do not have such a monitor program loaded and require a BDM to either load this serial monitor or an application program. This simple conversion turns the DEMO hardware into a quality Ethernet based BDM programming tool allowing such tasks to be reliably and efficiently accomplished.

**Any other hardware will also do the trick – it just needs the appropriate three GPIOs connected appropriately.*

This is the first Beta version of this programming tool which specifically enables programming of the Freescale™ serial monitor since this is the most important first task needed to be performed when starting with a new board. It supports programming of this code from the file system (whereby the serial monitor binary file is loaded once to the DEMO9S12NE64 and then programmed by clicking on a button on the programming page) or by upload from the user's PC from the Browser. The correct operation using the POST method has been verified using IE6.0 and Firefox 5.0.

Various other useful features such as monitoring registers and memory, halting and stepping code are also operational which make it already a useful tool for various purposes.

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

DEMO9S12NE64

The DEMO9S12NE64 is a low cost demo board but is often adequate for complete project development. Using its I/O extension port it is possible to connect external circuits and perform most tasks which will be required in a final system. In this document you will learn how to connect just three lines of the I/O extension port to the BDM port of further devices in order to communicate with them using the BDM protocol and perform such tasks as halting, resetting, starting, reading memory and registers, deleting FLASH and programming the serial monitor software.

It is assumed that you are already familiar with the use of the DEMO9S12NE64 and can load software to it over the serial port using one of various available programs.

This project is delivered as a zip file (BDM_ne64_B1.0.zip) including three file types:

- BDM_ne64.s19 The software to be programmed to the DEMO9S12NE64
- Various *.htm files The web pages to be loaded to the DEMO9S12NE64 via FTP
- N_FreescaleSerialMonitor.bin A binary image of the Freescale™ serial monitor which can also be loaded via FTP or programmed using the POST method each time that it is required

To get started, simply load the software BDM_ne64.s19 using the normal programming method. This can also be performed by using the free programming tool NE64_Deb which is delivered as part of the µTasker NE64 kit and available for free download from the following address: (http://www.utasker.com/software/ne64/NE64_Deb.zip). Instructions to its use can be found in the NE64 tutorial at <http://www.utasker.com/docs/NE64/uTaskerV1.2-Tutorial.PDF>. This requires that the serial monitor (as delivered installed in the DEMO9S12NE64) is still programmed.

Once the software has been loaded, the device will be reachable via Browser or FTP at the address 192.168.0.3. [<http://192.168.0.3> and <ftp://192.168.0.3>]. It is next necessary to load the web pages to the device using an FTP client with passive FTP support – Internet Explorer 6.0 is fine – just make sure that the passive FTP option is disabled in the extended options and again see the NE64 tutorial for a complete guide.

Default network settings are IP 192.168.0.3, Sub-net Mask 255.255.254.0, Default gateway 192.168.0.1, MAC 00-00-00-00-00-00, auto-negotiation. All settings can then be modified later from your Browser.

Copy the *.htm files delivered in the BDM_ne64.zip to the integrated FTP server (using drag-and-drop is easiest – see also the NE64 tutorial for more details). Optionally the serial monitor binary N_FreescaleSerialMonitor.bin can also be transferred.

Once the web pages have been copied you can then browser the device using a normal web browser and set an alternative IP address and a MAC address for your device (as long as it is in a local network its value is not that important, as long as it doesn't clash with another local device's MAC address. If your device is to be accesses from the Internet it will be necessary to have a MAC address belonging to you programmed or else put the device behind a firewall in a demilitarised zone).

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

Connecting the new NE64 BDM Programmer to another device

The software uses two ports pins on the port H as follows.

- Reset line PH5 – I/O pin 11
- BKGD line PH4 – I/O pin 9

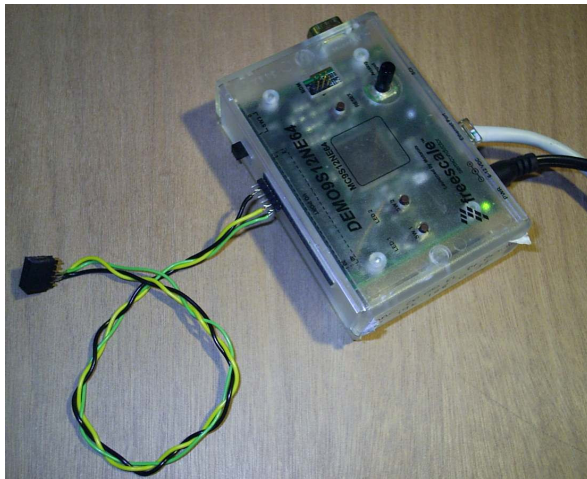
The ground connection is taken from I/O extension connector pin 3

The BDM connector (assuming that it has the same layout as the one on the DEMO9S12NE64) is connected to these lines as follows:

- Reset line BDM Pin 4
- BKGD line BDM pin 1

The ground connection between the two modules is taken from BDM pin 2

Here is a photo of a simple three wire connector used by us. It has a 10 way socket on the BDM end simply to aid in positioning it on another DEMO9S12NE64. Only the three described connections are actually connected and required.



DEMO I/O Pin 3 ↔ BDM 2 (0V)
 DEMO I/O Pin 9 ↔ BDM 1 (BKGD)
 DEMO I/O Pin 11 ↔ BDM 4 (RESET)

Demo board on the right being used to delete and re-program the serial monitor program on the demo board on the left.

Here the cable length is about 20cm and this has worked very reliably. Longer cable lengths are possible but there will (somewhere) be a limit due to the cable capacitance not allowing the signal timing accuracy to be achieved. Therefore we recommend also the same approximate length for your own use.

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

Using your new BDM programming tool

Once the software and the web pages have been loaded to the DEMO9S12NE64 to be used as a BDM programmer the rest is controlled from a standard Browser.

The start page looks like this (a target with application has been halted in this screen-shot):

MC9S12NE64 LAN BDM Programmer - Powered by µTasker

Register	Contents	Actions	
SP	0x3fee	Single step	<div style="border: 1px solid black; padding: 5px;"> Software Version: Beta1.0 IP Address: 192.168.0.6 MAC Address: 00-11-22-33-44-57 Device ID: Mark's BDM <input type="button" value="Save New Device ID"/> </div> <ul style="list-style-type: none"> o Configure LAN interface o Open a Memory Window o Go to Programming Side o www.uTasker.com
PC	0x4353 [0x32ee]	RUN	
IY	0x3fee	0000 RUN to breakpoint	
IX	0x3fee	HALT	
D	0x0000	Reset Target	
CCR	S X H I N Z V C <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	FORCE DEBUG MODE	
<input type="button" value="Modify registers"/>		Status = Halted	
<input type="button" value="Refresh"/>			

With the BDM cable connected to the target to be programmed, click on **“Reset Target”** to reset it. If it already has a program running it may not respond to a reset without first being forced into the BDM mode, which is performed by clicking on **“FORCE DEBUG MODE”**, after which the target will respond to the reset.

You will see that register contents are displayed and that it is also possible to modify them as well as the CCR (Condition Code Register) flags. It is also possible to step through code, but this will not be described any further here because we will concentrate on programming the serial monitor program to a new device.

Note that the software version is displayed along with the device’s IP settings. You can even personalise the device with its own name in the **“Device ID”** field.

μTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

If you need to set up the Ethernet LAN to suit your (normal) network, this can be performed by clicking on “**Configure LAN interface**”. The following side appears:

μTasker - LAN configuration

Ethernet Settings

Setting	Value	Modified
MAC address	<input type="text" value="00-11-22-33-44-57"/>	
IP address	<input type="text" value="192.168.0.6"/>	<input type="checkbox"/>
Subnet mask	<input type="text" value="255.255.254.0"/>	<input type="checkbox"/>
Gateway IP address	<input type="text" value="192.168.0.1"/>	<input type="checkbox"/>
Ethernet speed	100M <input type="radio"/> 10M <input type="radio"/> Auto-negotiate <input checked="" type="radio"/>	<input type="checkbox"/>
Configure using DHCP server	<input type="checkbox"/> (set IP to 0.0.0.0 if no preferred setting)	<input type="checkbox"/>
Settings validated	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Saving of new settings cause an immediate reset and must be validated within a period of 3 minutes otherwise the original settings will be returned.
This ensures that invalid settings do not render a device unreachable.*

[Go back to menu page](#)

Here is it possible to program the MAC address (if not already set) as well as the device's IP and Ethernet configurations. Please see the NE64 tutorial for a more detailed discussion of the use of this side.

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

From the start page you can also open memory windows, which open up in their own Browser windows. Several memory windows can be used at the same time to monitor various sections of the target devices memory. The following shows a typical memory display, where the address to be displayed can be entered and refreshed. Using “*Previous block*” and “*Next block*” you can then step forwards and backwards through memory.

MC9S12NE64 Memory Display

Address	Memory content (Hex)	ASCII
0x0000	0200 0000 0000 0000 1f00 1080 9000 0100
0x0010	3900 010d 000f 0010 0000 8201 0380 40f2	9.....@.
0x0020	0000 93f9 0000 0000 0000 0000 0000 0000
0x0030	0000 ff00 0000 007c 8080 f17c 4700 8000 G...
0x0040	0000 0000 0000 0000 0000 0000 0000 0000
0x0050	0000 0000 0000 0000 0000 0000 0000 0000
0x0060	0000 0000 0000 0000 0000 0000 0000 0000
0x0070	0000 0000 0000 0000 0000 0000 0000 0000

Loading the serial monitor program

Using the button “**Go to Programming Side**” on the start side, the Upload page will open.

Upload page

Program	Empty <input checked="" type="checkbox"/>	Delete	First delete the area(s) to be loaded (if not blank). Then select the Freescale Serial Monitor binary file to be programmed.
File System	Empty <input checked="" type="checkbox"/>	Delete	
Entire FLASH	Empty <input type="checkbox"/>	Delete	

[Go back to main menu](#)

Two methods are supported to upload the Freescale™ serial monitor to the new device. However in both cases it may first be necessary to delete the entire FLASH before commencing the upload itself.

If there is already code programmed to the serial monitor space (from 0xf800 to 0xffff in FLASH) the Empty check box next to “**Entire FLASH**” will not be set. This indicates that it is first necessary to delete the entire FLASH, which also deletes and FLASH protection which will often be present in this FLASH range. Simply click on the “**Delete**” button next to “**Entire FLASH**” and the Empty check box will then show that the area has been deleted.

In order to use the method “**Program Serial Monitor from file system**” the serial monitor binary file must have been copied to the file system. This is performed using FTP (as for copying the web pages) and details to the method can be found in the NE64 tutorial. When this button is clicked, the contents are copied from the file system to the target device. After the copy procedure a page will be displayed informing of the programming success or else of a programming error. The programming error will occur should the binary file not be found in the file system.

The second programming method is to use the Upload button. This also demonstrates the µTasker POST feature in operation. Select the file [N_FreescaleSerialMonitor.bin] to be uploaded from your PC using the search button (here displayed as “**Durchsuchen**” since the screen shot was made using a German version of Windows...) – *this is, for example, how web-mail programs allow users to send email attachments*. Then click on “**Upload**”.

Your Browser performs an upload of the data using the POST method and the BDM programmer copies the received data to the position of the serial monitor in FLASH (0xf800..0xffff). Again there will be a new page automatically opened to indicate programming success. In each case it takes about 2 or 3 seconds to program the serial monitor to the FLASH.

If you want you can check the contents of the address range beginning at 0xf800 in order to see the start of the serial monitor code in the target device. Then go to the start page and command a reset. You will see that the PC is set to 0xf800, the start address of the serial monitor code, and by pressing “**Single Step**” a few times the first few instruction executions are seen. Pressing “**Run**” allows the target to execute the freshly loaded serial monitor, after

µTasker – Guide to converting a DEMO9S12NE64 into a LAN based BDM programming tool for the MC9S12NE64

which the device can be contacted using the normal serial programming methods as does the DEMO9S12NE64.

- If you would like to learn more about the serial monitor program from Freescale™ there is an application note **AN2548**, including source code, on the Freescale™ web site.
- If you would like to learn more about actually controlling devices using the BDM then see section 17 (**Background Debug Module BDMV4**) of the NE64 data sheet, which gives enough details to implement the control.

Conclusion

The document has shown how a DEMO9S12NE64 can simply be transformed into a useful Ethernet based BDM tool to program the Freescale™ serial monitor to fresh devices.

The program `BDM_ne64.s19` was loaded to the DEMO9S12NE64, followed by its web pages via FTP. By connecting the DEMO9S12NE64 to a target device using a simple three wire BDM connector it was possible to load the binary serial monitor program either directly from the file system or via Browser upload. Register and memory contents of the target processor could be inspected and the new program could be started via the BDM.

The `BDM_ne64.s19` version Beta 1.0 is primarily intended for programming the serial monitor but it can be seen that various additional features already exist and that future version will allow uploading application code and deleting specific parts of FLASH memory.

We hope that you enjoy converting your own DEMO9S12NE64 to a useful programming tool and that you find the BDM programmer of value to your own work.

This software and documentation are offered free of charge in an “as is” condition. It is supported by email, again free of charge, but comes with no guaranty and is used entirely at your own risk – see the uTasker general licensing terms at the following address:

<http://www.utasker.com/Licensing/License.html>

When using the supplied software and following the guide, you accept these licensing terms which exclude us, as developers, from any liability claims.

This said, we have used the basic tool for over a year now with great success, so don't get worried...