

# **EMUL-ARM**

**Board Support: OKI ML67Q4003**

*August 7, 2003*

# Contents

---

|          |                                      |          |
|----------|--------------------------------------|----------|
| <b>1</b> | <b>INTRODUCTION .....</b>            | <b>4</b> |
|          | <b>Important Notes</b>               | <b>4</b> |
|          | <b>Memory Configuration</b>          | <b>4</b> |
|          | Flash Programming in EMUL-ARM .....  | 5        |
|          | Remap the internal RAM .....         | 5        |
|          | <b>Board Configuration</b>           | <b>5</b> |
|          | <b>Other Documents</b>               | <b>6</b> |
|          | <b>Target Board CD</b>               | <b>6</b> |
|          | Examples & Drivers .....             | 6        |
|          | Put/Get Value .....                  | 6        |
|          | <b>Supported Compilers</b>           | <b>6</b> |
| <b>2</b> | <b>DEVELOPING SOFTWARE .....</b>     | <b>7</b> |
|          | <b>Include Files</b>                 | <b>7</b> |
|          | <b>Startup Code</b>                  | <b>7</b> |
|          | <b>Memory Map / Linking</b>          | <b>8</b> |
| <b>3</b> | <b>SUPPLIED SOURCE CODE .....</b>    | <b>9</b> |
|          | <b>General Examples</b>              | <b>9</b> |
|          | $\mu$ C/OS-II .....                  | 9        |
|          | <b>MCU / Board specific examples</b> | <b>9</b> |
|          | <b>Drivers</b>                       | <b>9</b> |

---

## About This Guide

EMUL-ARM is a PC-based hardware debugger for the ARM™ Core (currently ARM7 and ARM9 cores). Seehau is the name of the user interface of EMUL-ARM – Seehau and EMUL-ARM is often used interchangeably.

This User's Guide helps you to understand how to use OKI ML67Q4003 (made by OKI) with:

- EMUL-ARM
- HI-TECH ARM-C and HI-TIDE
- IAR EWARM C/C++

**Note that there might exist two versions of this document** (and other BSP/Target Board Related documents):

- If it is located in the “SeehauARM\Documents” install directory then it relates to currently installed Seehau.
- If it is located in the “C:\Nohau\BSP\_ARM\Doc” directory then it relates to the currently installed BSP.

Please note that it is required to install BSPs to “C:\Nohau\BSP\_ARM” because compiler project files are often dependent on file location.

# 1 INTRODUCTION

## Important Notes

The target board comes with a manual – please read it. This document is intended to complement the board documentation – not replace it.

OKI has some FAQs on their web site with information – currently at:

- <http://www2.okisemi.com/us/docs/ARMFaqs.html>

It is assumed and required that the BSP is installed to c:\Nohau\BSP\_ARM.

## Memory Configuration

The memory map for the ML67Q4003 CPU board will vary depending on the **BSEL0/1 settings**. The CPU users manual shows internal flash starting at address C800\_0000. However, this is only true when BSEL0/1 are both tied low.

Internal Memory

- SRAM: 32K byte. Default address 1000\_0000 to 1000\_7FFF.
- Flash: 512K byte. Default address C800\_0000 to C807\_FFFF.

External Memory

- SRAM: 512K byte. Default address 0 to 7\_FFFF.

Note that the SRAM is sometimes referred to as "External ROM" in OKI board manual.

Following table shows how to use the on board switches to configure the memory map:

| Boot Type | BSEL 0 | BSEL 1 | ROM SEL | MEM DIS | Addr O    | Addr C800_0000 | Addr CC00_0000 |
|-----------|--------|--------|---------|---------|-----------|----------------|----------------|
| Int Flash | 0      | 0      | 1       | 0       | Int Flash | Int Flash      | Ext SRAM       |
| Ext Rom   | 1      | 0      | 1       | 0       | Ext SRAM  | Ext SRAM       | Int Flash      |
| Boot Rom  | 0      | 1      | 1       | 0       | Boot Rom  | Int Flash      | Ext SRAM       |
| Boot Rom  | 1      | 1      | 1       | 0       | Boot Rom  | Ext SRAM       | Int Flash      |

Nohau recommends using boot type "External ROM" for this board during (initial) debugging so that the 512K byte SRAM is used for debugging. It allows unlimited amount of breakpoints. Then when it is desired to map

## Flash Programming in EMUL-ARM

**Important!** The built in Flash programming in EMUL-ARM will only work when the internal flash is located at address 0. If it is located at other addresses, or if you want to use external flash, you will have to create a target definition file for it.

On the “Config | Emulator | Misc Tab”, see the option:

- Flash written as RAM / Disable Flash Programming

This option Disables / enables flash programming. When disabled – flash is written as RAM. So by checking and un-checking this option – you can go back and forth between flash and RAM easily. Remember to set BSEL0 and BSEL1 on the board accordingly.

## Remap the internal RAM

Then internal RAM can be remapped to address 0 using the “RMPCON” register. Note that you have to write 0x3C it before the desired value can be written.

TODO: Explain more about interrupt vectors etc.

## Board Configuration

Initially, set all switches in default positions as **indicated by graphics** on the board or to the **off positions**. Then set switches as follows:

Required Board Configuration for debug:

- **ICEEN = off** – Enable JTAG debug.
- **CKOE = on** – clock output enable.

Memory Map – recommended settings for initial debug with EMUL-ARM (“Ext Rom” in table in Memory Configuration section). See the Memory Configuration for other configurations and additional information.

- **BSEL0 = on (1) – for internal Flash at address 0 – use BSEL0 = off (0)**
- **BSEL1 = off (0)**
- **ROMSEL = on (1)**
- **MEMDIS = off (0)**

Some other switches (though they can be set as desired):

- **NEFIQ = on** – Enable NEFIQ button.
- **EXINT0 = on** – Enable EXINT0 button.

## Other Documents

Please also see following documents:

- **ARM\_BSP.pdf** – overview of general source code that is supplied with EMUL-ARM for OKI MOL67Q4003 – see ARM\_BSP.pdf. This document discusses Timer, different Target Console applications and uC/OS-II.
- **ARM\_Compilers.pdf** – how to set up different compilers to be used with EMUL-ARM.
- **Nohau\_Monitor .pdf** – using the Nohau Monitor to get printf() from target application in Seehau etc.

## Target Board CD

The ML67Q4003 Target Board comes with a CD from OKI.

### Examples & Drivers

The OKI CD comes with lots of examples and drivers for the ML67Q4003. The code and project definitions are all based on the RealView tools.

### Put/Get Value

All the programs on the CD use macros like “get\_value” and “put\_value” to read and write to registers on the MCU. A sample function from the CD may then look like:

```
void set_timer(void)
{
    put_hvalue(TMEN, 0x0);
    put_hvalue(TMOVF, 0x01);
    put_hvalue(TMRLR, VALUE_OF_TMRLR);
}
```

While this approach certainly works, there are alternatives. We prefer using NML674001.h as defined in the ARM BSP (or the one defined by your compiler vendor). This would make the same function look like:

```
void set_timer(void)
{
    TMEN = 0x0;
    TMOVF = 0x01;
    TMRLR = VALUE_OF_TMRLR;
}
```

## Supported Compilers

All examples can be compiled with HI-TECH ARM-C and IAR C/C++. There are ready to use project files in each target application directory.

## 2 DEVELOPING SOFTWARE

### Include Files

Compiler will generally have `#include` files for MCU registers. If yours doesn't have it, you can use following file (ML67Q4003 is a "member" of the "4001 series"):

- `c:\Nohau\BSP_ARM\Src\Inc\NML674001.h`

All examples in Nohau BSP ARM are based on a couple of `.h` files. These are:

- `NConfig.h` (in the application directory).
- `c:\Nohau\BSP_ARM\Src\Inc\NDefs.h`
- `NConfig.h` (located in the application directory). "NDefs.h" includes this file.

The Nohau Monitor uses an additional `.h` file in the application directory:

- `NMon_Cfg.h` – Nohau Monitor.

### Startup Code

HI-TECH ARM-C does not use startup code, however following files can be used to setup stacks and interrupt vectors (use one of the setup files):

- `C:\Nohau\BSP_ARM\Boards\OKI\ML67Q4003\Startup\setup_HTC_ML67.s`
- `C:\Nohau\BSP_ARM\Boards\OKI\ML67Q4003\Startup\setup_HTC_ML67_ucosii.s`
- `C:\Nohau\BSP_ARM\Boards\OKI\ML67Q4003\Startup\stacks_HTC.c`

For IAR C/C++, an example `c` startup code is given in (use one of the `cstartup` files):

- `C:\Nohau\BSP_ARM\Boards\OKI\ML67Q4003\Startup\cstartup_IAR_ML67.s`
- `C:\Nohau\BSP_ARM\Boards\OKI\ML67Q4003\Startup\cstartup_IAR_ML67_ucosii.s`

## Memory Map / Linking

OKI ML67Q4003 makes it easy to switching between develop and debug in RAM and Flash – all you have to do is to change the settings on the board as described earlier.

This gives following memory map:

- 32K Internal RAM:       Address 1000\_0000 to 1000\_7FFF.
- 512K Program Memory:   Address 0 to 7\_FFFF.

The HI-TECH ARM assumes this memory map when selecting ML67Q4003 in a project (with default options).

With the IAR compiler, this is reflected in the link control map as supplied in Nohau BSP ARM:

- C:\Nohau\BSP\_ARM\Boards\OKI\ML67Q4003\Link\LINK\_IAR.xcl



## 3 SUPPLIED SOURCE CODE

### General Examples

These are applications that are shared between most supported boards. See ARM\_BSP.pdf for additional information.

The application categories are available:

- **TargetConsole** - Shows the capabilities of the EMUL-ARM debugger-target communication.
- **Simulated** – TimerSimulated and TimerSimulatedNMon are available.
- **Interrupt** – Shows how to do basic interrupts with and without the debugger target-communication.
- **uCOSII** – Shows the uC/OS-II RTOS.

### µC/OS-II

- uC/OS-II for ML67Q4003 is in a beta stage and does not use the reentrant IRQ driver yet.
- No Thumb mode examples with the HI-TECH compiler (which does not yet support Thumb).
- The “Full” application is not yet tested.

### MCU / Board specific examples

Not yet available.

### Drivers

Drivers are located in the “ML67Q4003 \Drv” directory. Following drivers are currently available:

- **Irq6745K** – handles reentrant IRQs for ML764K and ML765K. Currently in a beta stage.