

## Getting Started and Blinky Example

This **Quick Start Guide** shows how to install the Keil software development tools and run a variety of example projects on the MCBTWRK60 board containing the Freescale **MK60N512VMD100** device.

### Installing the Software

The MCBTWRK60 works with the Keil **MDK-ARM** (Microcontroller Development Kit) which features the industry-standard ARM® compiler, the µVision4 IDE/Debugger and analysis tools.

All examples shown in this quick start guide will work with the MDK-Lite Edition:

- Download the latest version of MDK-ARM from:
  - [www.keil.com/demo/eval/arm.htm](http://www.keil.com/demo/eval/arm.htm)
- Double click the downloaded file and follow the instructions to install MDK-ARM into your root directory
- The evaluation version does not require a license key

**Note:** MDK-ARM will be installed in your root directory by default. This ensures all included examples work correctly.

### Connecting to the MCBTWRK60

MDK-ARM supports ULINK2, ULINKpro and OS-JTAG downloading and debugging for the MCBTWRK60 board.

To connect via ULINK2 or ULINKpro to your PC you need:

- USB A-miniB cable to power the board.
- ULINK2 or ULINKpro Debug Adapter with USB A-B cable

To connect via OS-JTAG to your PC you need:

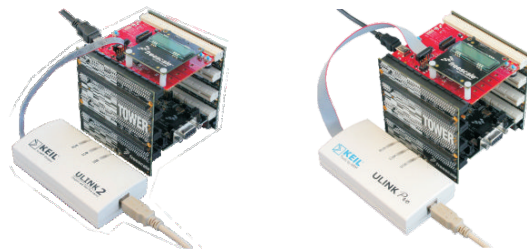
- USB A-miniB cable (provides power & debug connection).

### Using a ULINK Adapter or OS-JTAG

The ULINK family of debug adapters connect your PC's USB port to the Cortex Debug connector on your target board allowing you to download and analyze embedded programs running on your target hardware.

ULINK2 supports JTAG, Serial Wire Debug (SWD), and Serial Wire Viewer (SWV) for on-the-fly debug and analysis. ULINKpro supports these interfaces plus ETM Instruction Trace.

The **ULINK2** and **ULINKpro** adapters connect to the MCBTWRK60 using the Cortex Debug connector.



The **OS-JTAG** JTAG interface from P&E Micro is included on the MCBTWRK60 board. The USB cable provides the power and the debug connection from the PC to the MCBTWRK60.






### Blinky Example


This example demonstrates the ease of downloading and debugging an application on a target board.

It can be found at:







**C:\Keil\ARM\Boards\Freescale\TWR-K60N512\Blinky**

To use this example

-  Start µVision
- Open the Blinky.uv2 project file  
**Project - Open**
-  Compile and link the Blinky application  
**Project - Build**
-  Program the application into on-chip Flash ROM  
**Flash - Download**

- The LEDs on the target are controlled by the potentiometer setting. As the setting increases, the LEDs scroll faster.
-  Start debug mode

Using the debugger commands, you may:

-  Single step through code
-  Set breakpoints
-  Run the application
-  Review variables in the watch window
-  Reset the device to re-run the application
-  Use the yellow arrow (program counter) to view the current assembler or C statement

## Blinky example with RTX

This example implements a step-motor driver application using the RTX real-time operating system. Four LEDs blink to simulate the activation of the four output driver stages.

It can be found at:

**C:\Keil\ARM\Boards\Freescale\TWR-60N512\RTX\_Blinky**

To use this example:

- Start  $\mu$ Vision
- Open the Blinky project file
  - Project - Open**
- Compile and link the application, then program it into on-chip Flash ROM
  - Project - Build Target**
  - Flash - Download**
  - Press RESET button**

The LEDs on the target blink from left to right, and back again. Turn the potentiometer dial to vary the speed of blinking. You may debug the example project using the  $\mu$ Vision debugger.

- Start debug mode
- Set breakpoints  
A breakpoint can be a simple halt at an instruction, or a complex combination of boolean conditions
- Run the application  
The code runs to the end, or to the next breakpoint

Detailed information about the RTX tasks and configuration can be seen in the **Kernel Aware** dialog windows.

### Debug - OS Support

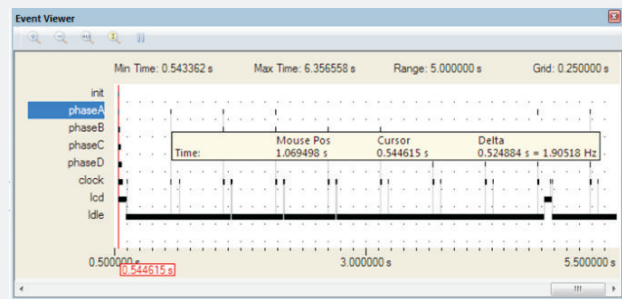
The **RTX Tasks and System** window displays system resource and active task information.

Property	Value
Timer Number:	0
Tick Timer:	10.000 mSec
Round Robin Timeout:	50.000 mSec
Stack Size:	200
Tasks with User-provided Stack:	0
Stack Overflow Check:	Yes
Task Usage:	Available: 7, Used: 5
User Timers:	Available: 0, Used: 0

ID	Name	Priority	State	Delay	Event Value	Event Mask	Stack Load
255	os_idle_demon	0	Running				0%
6	clock	1	Wait_AND		0x0000	0x0100	32%
5	phaseD	1	Wait_AND		0x0000	0x0001	32%
4	phaseC	1	Wait_DLY	30	0x0000	0x0001	36%
3	phaseB	1	Wait_AND		0x0000	0x0001	32%
2	phaseA	1	Wait_AND		0x0000	0x0001	32%

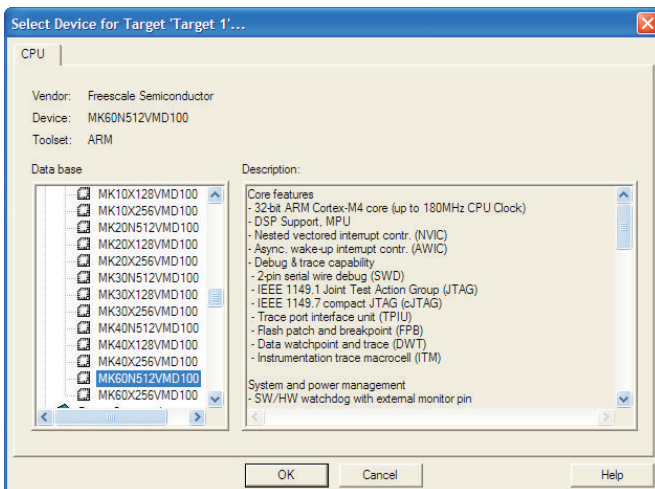
The **Event Viewer** displays detailed task switching information.



## Creating a New Project

MDK makes it easy to create new programs for the K60N512 by providing the Device Database, Startup Code and Configuration Wizard to help you select your target, and configure the tools.

- Start a new project
    - Project - New  $\mu$ Vision Project**
  - Assign a project file name and save to your preferred location
- $\mu$ Vision opens up the Device Database for you to select your targeted device.
- Scroll down to vendor Freescale, and select a device



$\mu$ Vision can automatically import the device's startup code into the project workspace, to save time, and get you started quickly.

To add files to your new project, expand the folder **Target 1** in the **Project** window and right click on **Source Group 1** to open the context menu. Then use **Add Files to Group** to add existing source files to the project.

New source files may be created for this project using **File - New**. Compile and debug following the steps shown for the example above. Ensure that the debug driver is selected and configured.

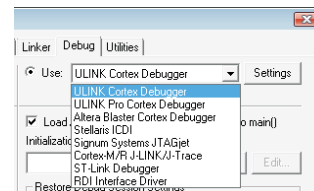
To select and configure the ULINK adapter or OS-JTAG:

Click on the **Options for Target** icon

- On the **Debug** tab,

For Debug Adapter...	Select...
Keil ULINK2	ULINK Cortex Debugger
Keil ULINKpro	ULINK Pro Cortex Debugger
P&E Micro OS-JTAG	Pemicro OSJTAG

then click **Settings** to configure the debug session



- Do the same under the **Utilities** tab to configure Flash download