

Debugging with CodeWarrior and USBDM

Debugging with CodeWarrior

To download and debug applications using CodeWarrior you need the ELZET80 USBDM adapter and have the driver software installed. You can get the newest driver revisions from here: <http://usbdm.sourceforge.net/>

Debugging without bootloaders

If you want to debug your application and don't need the bootloaders you can just press the „Start Debugging“-button in CodeWarrior. If you haven't already done it, select the configuration for your hardware and debugging should start automatically.

Debugging with bootloaders

If you want to debug your applications with bootloaders in place you need to download them first with the „ARM Programmer“, a tool which is installed with the USBDM drivers. After you start the program and it has detected your USBDM adapter, switch over to the „Target“-tab and press „Load Hex Files“. Choose the compiled loader binary (e.g. uTaskerBM_KBED.afx.S19) from the „KineticCodeWarrior“-directory in the applications folder. Select „Unsecure“ as security option. This is very important otherwise debugging with bootloaders will NOT work.

After everything is set you can press the „Program Flash“-button.

Do this with the „uTasker Bare Minimum“-loader and with the „Ethernet Bootloader“ and you're ready to go!

In CodeWarrior select the debug configuration for your hardware with „BM“ in the name and start debugging.

Note that this approach just works for one session. If you want to restart debugging you need to erase the device either with the ARM Programmer and the „EraseMass“ option and reflash the loaders or you need to execute the „dc6“ command in bootloader-mode to erase the application.

You can use ELZET80's Device Discovery application to force the device to bootloader mode. After that you can connect to the serial console and execute the „dc6“ command. Output should look like in the picture below. If the code was erased successfully you should be able to start debugging again.

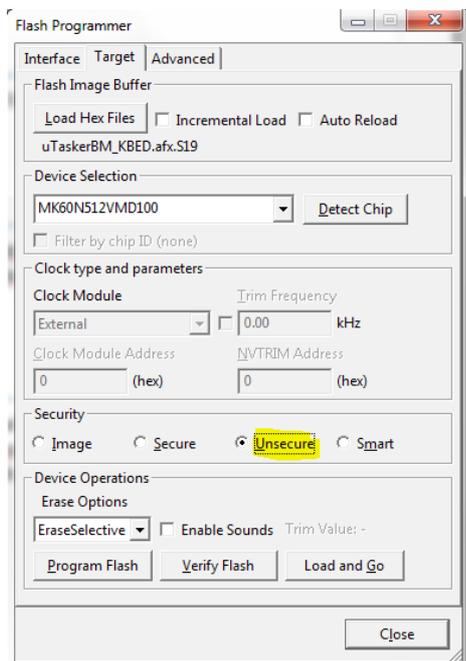


Figure 1: ARM Programmer Target-tab

NET-kBed Serial Loader

```

=====
[0x0000c000/0x0002ffff]
me = mass erase
bc = blank check
dc[1..6] = delete code (64k/96k/128k/160k/192k/224k)
ld = start load
go = start application
> dc6
Delete code [y/n] ? >

```

Deleting code: 0x00038000...successful

Figure 2: NET-kBED Serial Loader