Introduction

The µTasker "Bare-Minimum" Boot-loader allows software uploading via Ethernet or the Internet to be performed with only a small boot loader, occupying 7 x 256byte FLASH sectors (1,75k) on the SAM7X.

The Bare-Minimum Boot-loader is detailed in the document "uTaskerBoot.doc".

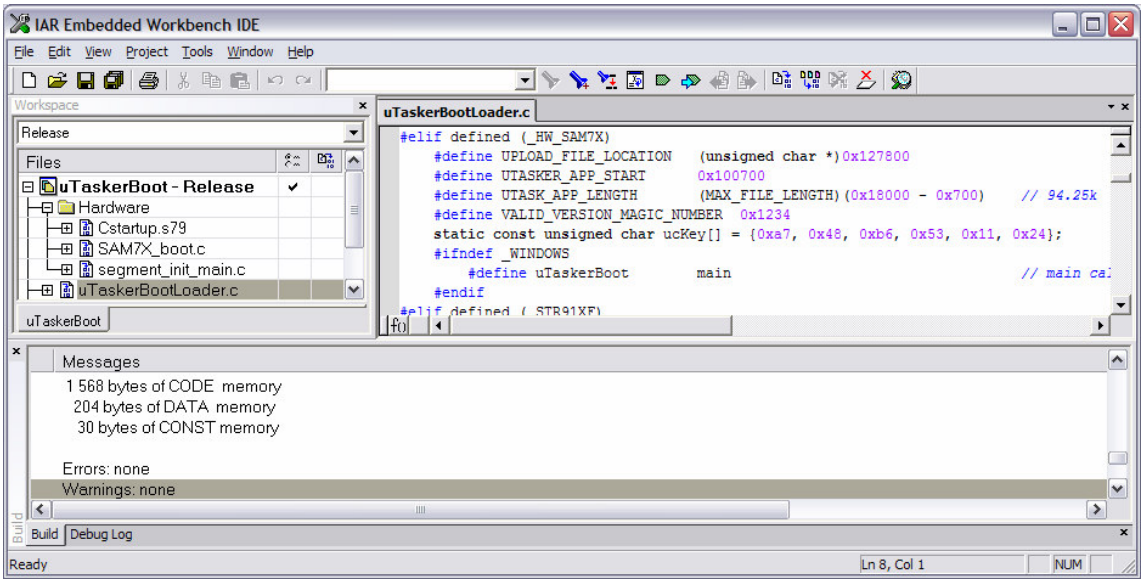
This document explains the method for using and programming the project for the SAM7X and the IAR project. It is assumed that the SAM7X256 is available (for example on the ATMEL demo board) since it allows loading software updates up to 94,25k in size, where as the SAM7X128 would require other settings and could only load file of half the size.

Projects and Targets

There are two IAR projects to be used to achieve the goals of programming the project with "Bare-Minimum" loading support. These projects are delivered in the µTasker demo project and so can be used immediately for first tests and also as a basis for further projects.

Both projects can be opened in two IAR Embedded Workbench IDEs at the same time.

The boot project is shown in the project window below. It should be compiled with the target RELEASE and compiles and links to take over the start-up sequence of the target.



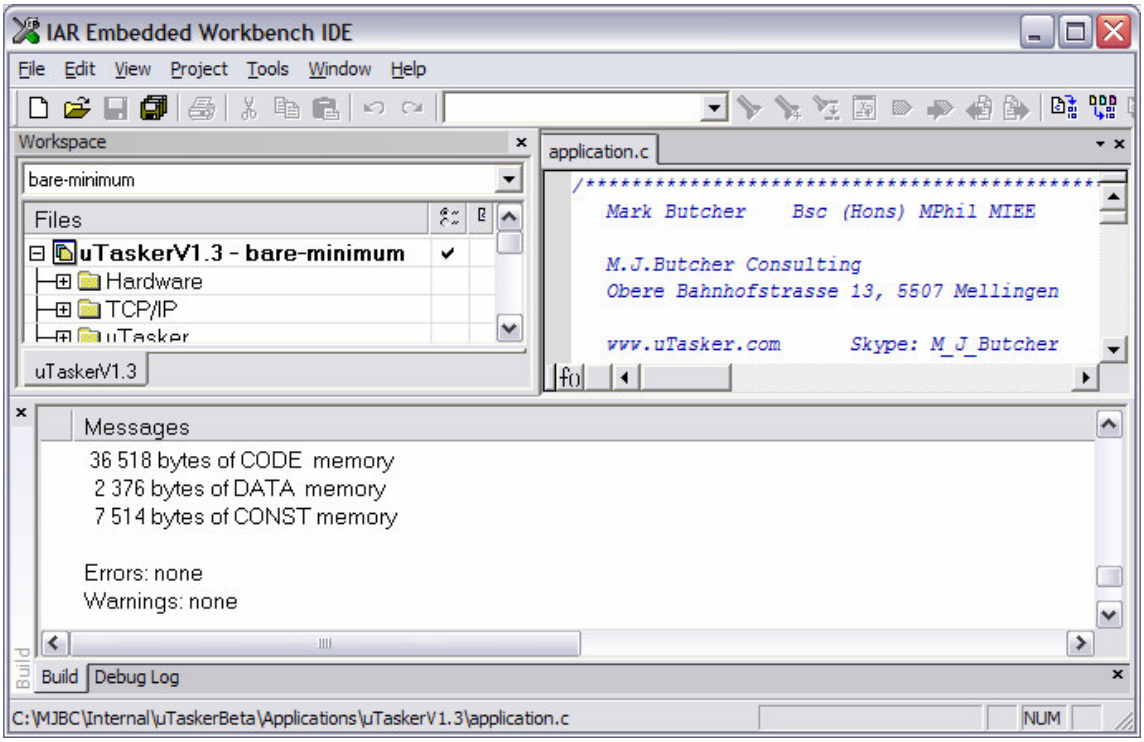
The file uTaskerBootLoader.c contains various settings which can be customised according to the details given further below.

Once this project has been compiled, it can be loaded to the target together with an initial application, where it occupies the first 1,75k of FLASH.

The target generates the boot loader binary file

`\uTaskerBoot\IAR_SAM7X\Release\Exe\uTaskerBoot.bin`

Details for loading the code are given after the application project has been discussed.



This project window shows the µTasker demo project which has three target configurations: Debug, Release and bare_minimum.

The ***release*** project generates a stand-alone version for FLASH which doesn't require the boot loader but doesn't support software updates as possible when the boot loader is installed.

The target generates the demo project binary file

`uTaskerV1.3\IAR_SAM7X\Release\Exe\uTaskerV1.3.bin`

The ***debug*** project runs in SRAM, but is not discussed further here.

bare_minimum is used when the project is to support the bare minimum boot loader or if the code is to be uploaded to a board supporting the boot loader principle.

The code is linked to start at the address 0x00000700. (The 8th FLASH sector in the SAM7X).

The target generates the demo project binary file

`uTaskerV1.3\IAR_SAM7X\bare-minimum\Exe\uTaskerV1.3.bin`

Loading the first software to the target

Once the boot loader project and the µTasker demo project have been successfully built, the binary files can be loaded to the SAM7X based board.

Note that the instructions are also valid for any application but the demo project is used here as reference.

The simplest and sleekest method for loading the code is using the ATMEL SAM-BA tool as described in the µTasker tutorial to the SAM7X – preferably using USB. The problem is that the two binary files must first be merged into one file so that this is simply possible.

Fortunately there is a tool delivered with the project which can be executed to do this. It can be found in the target file:
`uTaskerV1.3\IAR_SAM7X\bare-minimum\Exe\uTaskerMerge.exe`

It takes the two binary files as parameters and creates a single one with the third parameter name. To merge the two files in their original folders the following command can be executed:

```
uTaskerMerge.exe
..\..\..\..\uTaskerBoot\IAR_SAM7X\Release\Exe\uTaskerBoot.bin
uTaskerV1.3.bin uTaskerV1.3_boot.bin
```

We will see below that this is all set up in a bat file to make the process very simple. The bat file `BM-Convert.bat` in the target directory can be executed to generate the loadable file called `uTaskerV1.3_boot.bin`.

If you now load this file using SAM-BA to your target the µTasker demo project will run on your board. The boot loader is also now installed so the software upload feature is possible.

Be warned that the target code built to work together with eth bare-minimum boot loader will not be able to run alone because it is linked to start at the address 0x700 ad so has no reset vector of its own.

Uploading new software via the network

The idea of the bare-minimum boot loader should be clear from the document "uTaskerBoot.doc". It enables uploading new software via the Ethernet connection, which has various advantages. As well as the obvious one that it can be performed at a distance and over the Internet, it also allows updates without first deleting the entire FLASH contents, and any data or files stored in it.

In order to upload software, the application installed on the board must support the file transfer to the correct address. See "uTaskerBoot.doc" for more details. The µTasker demo project supports it via HTTP post and FTP.

The binary file generated by the ***bare_minimum*** target can not be loaded directly because it is missing an important header used for verification and authentication.

The file `BM-Convert.bat` in target directory contains in addition to the merge command discussed in the previous section also the following command:

```
uTaskerConvert.exe uTaskerV1.3.bin z_Upload.bin -0x1234 -
a748b6531124
```

This uses the conversion utility to take the generated file (`uTaskerV1.3.bin`) and add the header (the code a748b6531124 corresponds to the boot loader setting so that only files generated with the correct key will be accepted – the key can of course be changed for your own project). The utility generates the output file z_Upload.bin. (Note that this file is automatically put to the correct location in the file system).

Since the demo project supports HTTP post, simply connect to the web server using a standard web browser and go to the page with the software upload support (administration side). Select the file to be uploaded and click on the "Upload new software" button. After about 10s the page will refresh and the new software will have been programmed in the meantime.

Even if the project doesn't have a web server supplying the upload support, the file can also be copied using FTP. Simply grab the file to be uploaded and drag it to a browser connected to the FTP serving running on the target. Reset the target and the boot loader will do the rest (if you are not next to the board this can also be commanded via the web browser on the administration page).

Both methods work well and even resets or power downs during the process will not cause catastrophic failure to occur. Again see the boot loader document for more information.

File system requirements

It is important that the file location in the file system corresponds to the boot loaders settings. This also depends to some extent on the file system settings since the SAM7X has a very good FLASH granularity enabling various settings to be configured.

The demo project uses a file size resolution of 1k and the boot loader is configured in its delivered form to check for any copy new software starting at the location 0x127800. This location corresponds to the last file 'z' in the file system (see also the file system examples in `\uTaskerV1.3\WebPages\WebPagesSAM7X\FileSystem\` for an overview of this setting). The demo project is also configured to be able to work with file system sizes and files or larger than 64k as also described there.

In any case, the demo project is set up to show the boot loader in action and serves as a reference before any personal changes are made.

Summary

This document has introduced the µTasker "Bare-Minimum" Boot-loader on the SAM7X with IAR. It has described how the boot loader and the application can be loaded to the FLASH memory to perform a platform for further software upgrades via Ethernet.

Each time a new application software is created it can be compiled for the Bare-Minimum target and converted to a binary file for simple transfer to the target via HTTP post or FTP.

The next time the board is reset, the new program is automatically updated in FLASH, allowing unlimited changes to the application code including operating system, driver, TCP/IP stack and application modifications.

The process is safe in the event of resets or power cycles during the process and is also secure against malicious software upgrade attacks due to the security key which must match between the key configured in the boot loader and the encoded key in the software header.

For full technical details of the boot loader see the document "uTaskerBoot.doc".