Time Profiling Support Tool

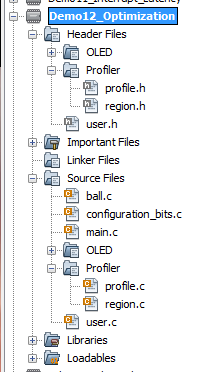
The time profiling tool uses script files and a GNU binary utility program called xc32-nm to reads the program’s executable image (elf) file and extract function name, address and size information. With this information it can create a C file (region.c) with profiling support:

* A constant table (RegionTable) containing name, starting and ending addresses for each function’s code.
* A table (RegionCount) to track the access count for each region.
* A table (SortedRegions) to hold a list of region numbers sorted by access count.
* An initialized constant variable NumProfileRegions.

Details of how the tool works are provided in Module 12A (Performance Analysis).

# Installation

* Copy the Scripts folder into your project folder.
* Copy the contents of the src folder into your project’s source code folder. This will create a subfolder called Profiler which holds region.c, region.h, profile.c and profile.h.
* Add these .c and .h files to the project file tree in MPLAB X as shown in the figure below.



# Time Profiling Procedure

Use the following build procedure:

* Build your program in MPLAB X.
* Run **update\_regions.bat** (in the Scripts folder) to create a new region.c file (which is placed in the src\Profiler folder). This file will have the correct number of entries but the addresses may be wrong (this is ok – a later pass will fix this).
* Rebuild your program in MPLAB X to generate an executable with the correct size region table. **Note that this table still uses the old function addresses, which will be wrong if the previous region table had a different number of entries.**
* Run **update\_regions.bat** to create a new region.c file with the correct function addresses and the correct number of entries.
* Rebuild your program in MPLAB X to generate an executable with the correct region addresses.
* Now you can download and execute your program on the target hardware. This will populate the RegionCount table to indicate the execution time profile.

In the future this process will likely be automated and integrated into the build process.

# Warnings and Troubleshooting

* You must regenerate region.c and rebuild your executable every time you change your program in a way that might change functions sizes or locations.
* There is a chicken-and-egg problem: in order to create region.c, we need the executable file. But the linker will not create the executable file if region.c does not exist. So the solution is to copy a dummy region file (dummyregion.c) to region.c. Then the linker will be able to build the executable, which getregions can analyze to create the correct region.c.
* If MPLAB X complains about not being able to reload region.c, try closing and reopening the region.c window.
* The executable file location is hardcoded in update\_regions.ps1 as **..\dist\PIC32MZ\_EC\debug\\*.elf**. If you change the name of the target configuration (e.g. processor type) or build to debug on your target board, you will need to change this path accordingly. In the future this will be likely be automated.