MCBSTM32C Tutorial

Claus Holmgaard

1 References

Mercantec MCBSTM32C
LIS302DL Datasheet
Keil CMSIS I2C Driver

2 New Project

Create a new project in Keil.

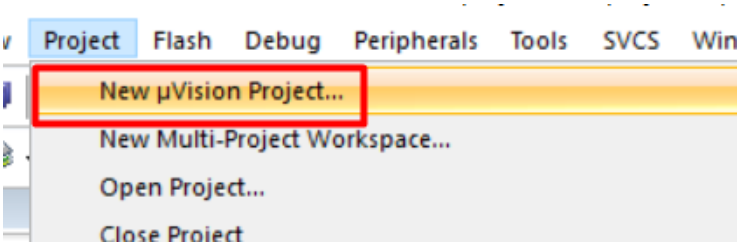


Figure 1: New Project.

Choose a place to save it.
Then select the device *STM32F107VC*.

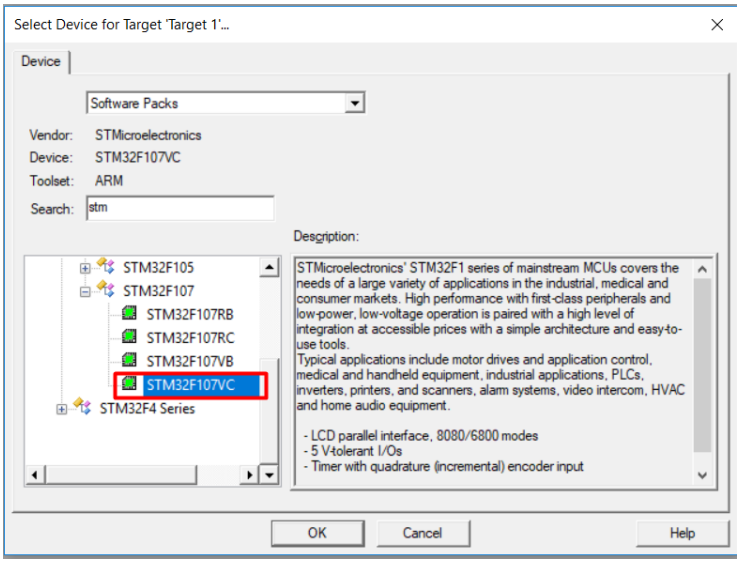


Figure 2: Select Device.

The Manage Run-Time Environment should open automatically. If it does not, open it, and make sure the board variant is *MCBSTM32C*. It defaults to the E variant.

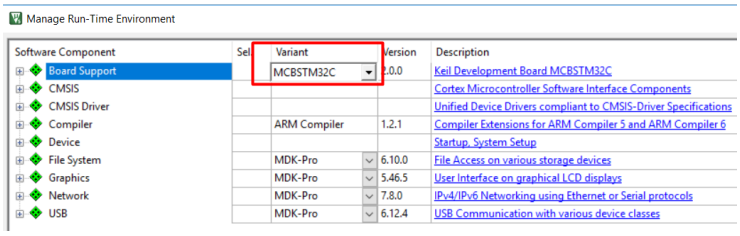


Figure 3: Select Device.

Also make sure CMSIS→Core, RTOS→Keil RTX and Device→Startup is selected.

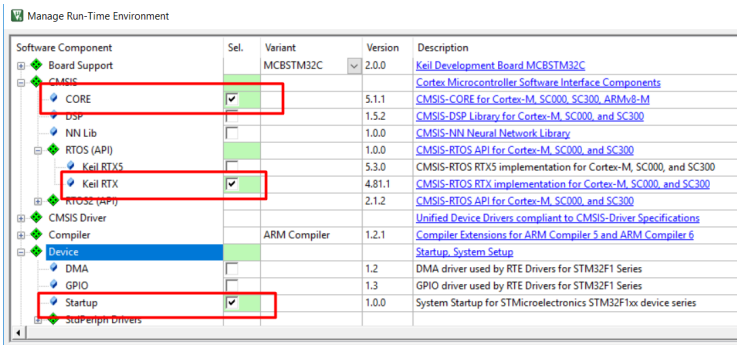
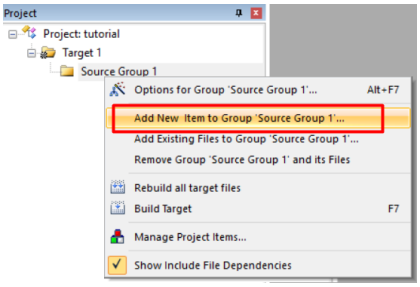
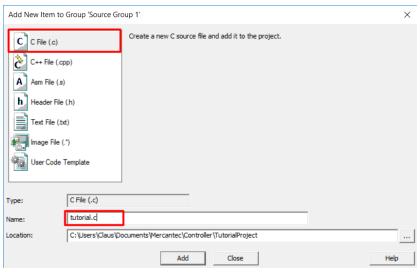


Figure 4: Run time selection.

Now we want our main .c file, I have called it *tutorial.c*. In the project pane, expand project, expand the target, then right click the source group, and select 'Add New Item'. Select a C filetype, and type in the name.



(a) Add new file.



(b) Type and name.

Our project now looks like this, where tutorial.c is empty.

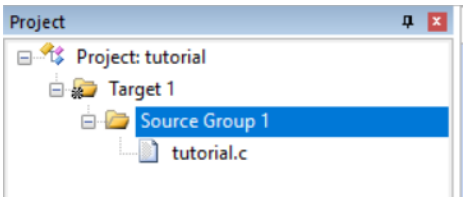
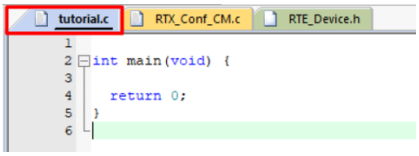
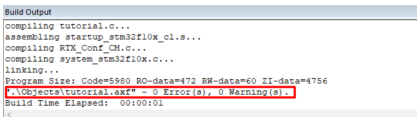


Figure 6: Project Structure.

We add the most basic code to tutorial.c, and try to compile it.



(a) tutorial.c.



(b) Compile output.

Open Target Options

Figure 8: Target Options.

Open the debug settings, and enable reset and run

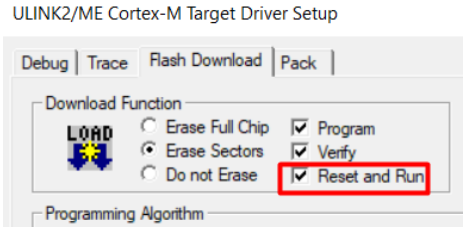
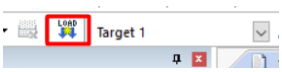
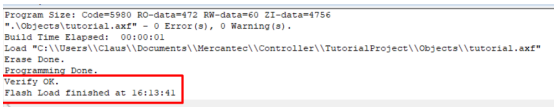


Figure 9: Reset and run.

You can load the program on to the chip, though it will do nothing at this point.



(a) Load to device.

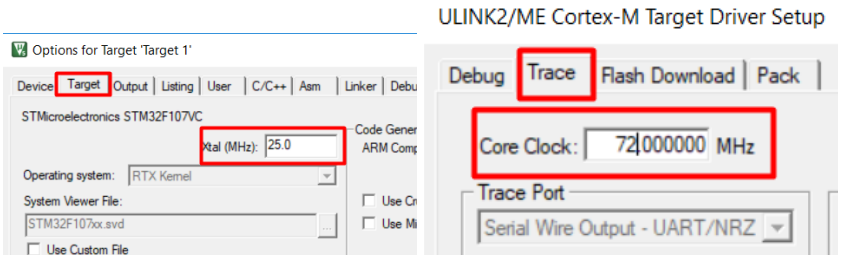


(b) Load output.

At this point, the very basic Structure for a project is there, and we're ready to move on. As someone who's not entirely sure what we're doing, we want debugging output next.

3 Debugging using printf

First, there's some setup to do.
Like before, open the target options. Set the Xtal frequency to 25MHz. Then, also like before, open the debug settings and set the core clock to 72MHz.



(a) Xtal. (b) Core clock.

Open the Run-Time Environment Manager, and make sure *STDERR*, *STDIN* and *STDOUT* in Compiler→I/O is set to *ITM*.

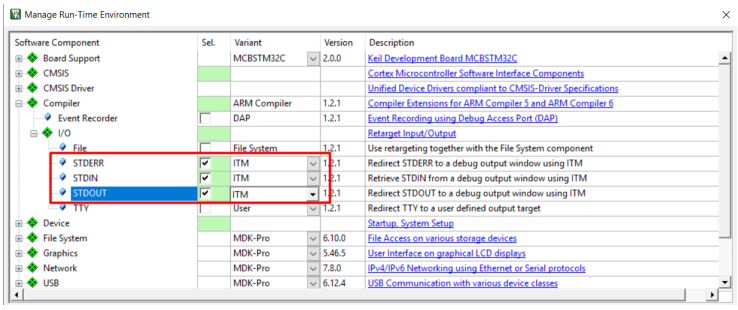


Figure 12: Run time debug settings.

In Target Options→Debug→Settings enable Trace, disable timestamps(I found them to generate quite a bit of traffic) and set the ITM Stimulus Ports.

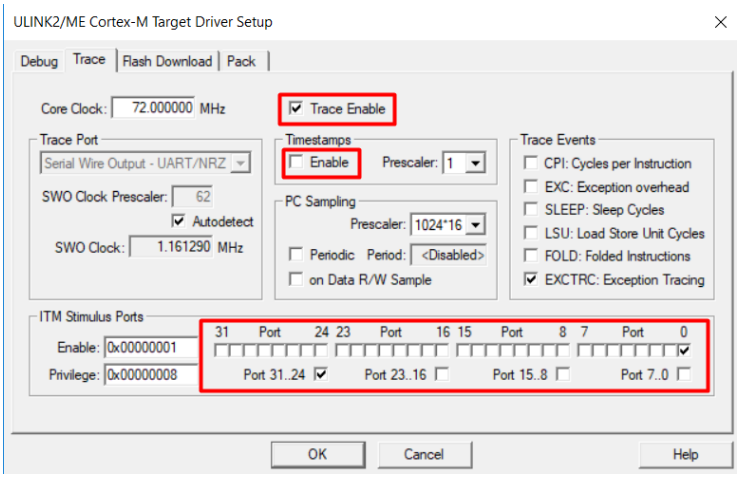


Figure 13: Trace settings.

Apparently printf statements use quite a bit of memory, so we need to increase the stacks size available. Open the file *CMSIS/RTX_Conf_CM.c*, and select the configuration wizard tab. Here, increase the stacks size.

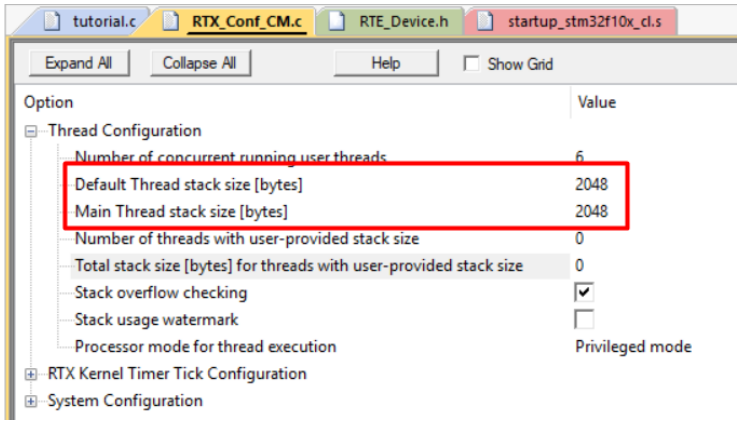


Figure 14: Stacks size.

Add a bit more code to *tutorial.c*.
We add *stdio.h* to get access to *printf*, and *cmsis_os.h* to get access to *osDelay*.

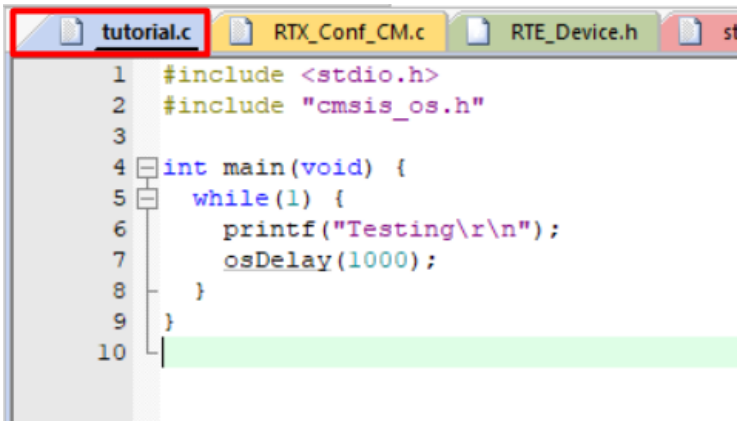


Figure 15: More code.

Compile the code, and start the debugger.

Figure 16: Start the debugger.

Open the printf debug viewer.

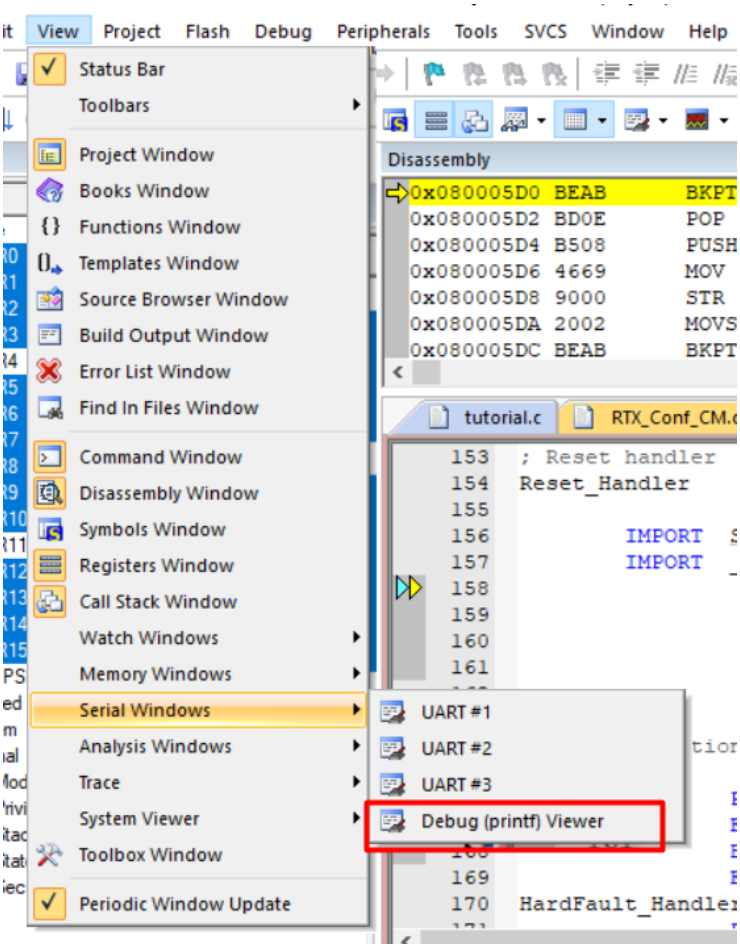


Figure 17: Debug window.

Start the debug session.

Figure 18: Start debug.

And the output from the printf statement should appear in the debug window.

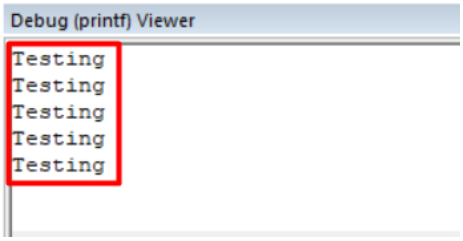


Figure 19: Debug output.

4 LCD

5 I2C & Accelerometer