Compiling MicroC-OS II code with Visual Studio

Operating system coming with Labrosse's book is done for old Borland compiler. These systems are no longer used. This installation can also be modified for newer versions of Microsoft Visual Studio.
Programming uCOS is almost same in this simulated environment as it would be in real HW. This kind of simulated approach is also made in business life, mainly because :

- Real hardware is not yet available, because our company is also developing a the same time the real HW, but it's not ready yet.
- SW is developed to be easily ported to different HW platforms, so it is easier to create a portable SW is in simulated environment

More information about this Win32 ported environment is found from http://www.it.fht-esslingen.de/~zimmerma/software/index_uk.html.

Preparations

1) Unzip the ZIPed Windows port files (whole directory tree of Ports) into directory C:\SOFTWARE\UCOS-II
2) Unzip the general part of uCOS into C:\SOFTWARE\UCOS-II\SOURCE
3) Download and unzip the first exercise RTOS_1 in to directory C:\SOFTWARE\uCOS-II\Ports\80x86\WIN32\VC\exercises\
4) Be carefull to use a local drive (C:\), Do not use network drive, it will give you problems
5) In this exercise directory there is ready made makefile (MAKEFILE.vc) for compiling this exercise
6) Make modifications to makefile if nessesary, so the path is pointing to exact location of your files
7) Check that filename in variable EXAMPLE hac value rtos_1 also check that UCOS_PORT_EX point to the same directory that this rtos_1.c is located

Compile from command line

This method is easier when you have problems with paths in Visual Studio Project definitions

- From Windows menu Open Visual Studio Command Prompt from All Programs/Visual Studio 20xx/Visual Studio Tools
- Move to the working directory to place where you have the makefile•

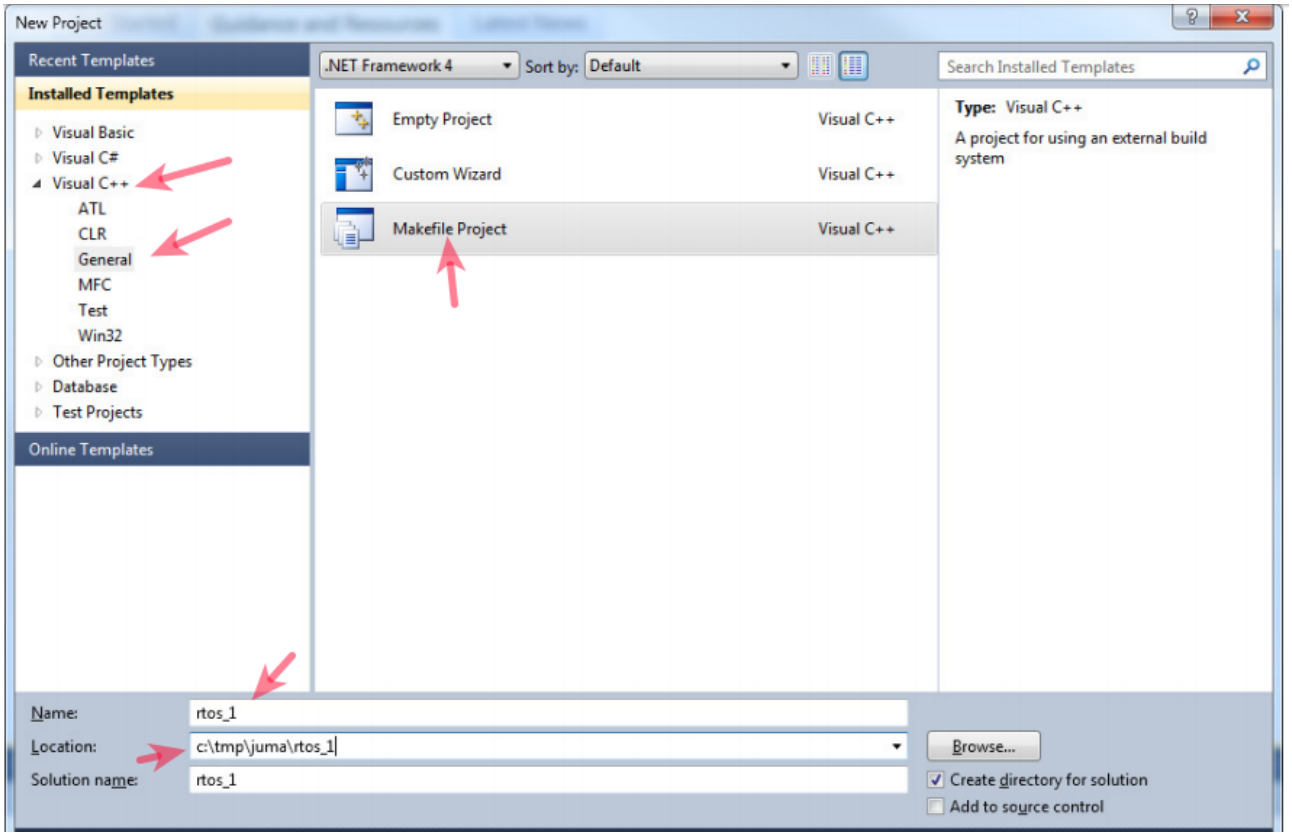
Give command

nmake /f makefile.vc all

Compiling with Visual Studio

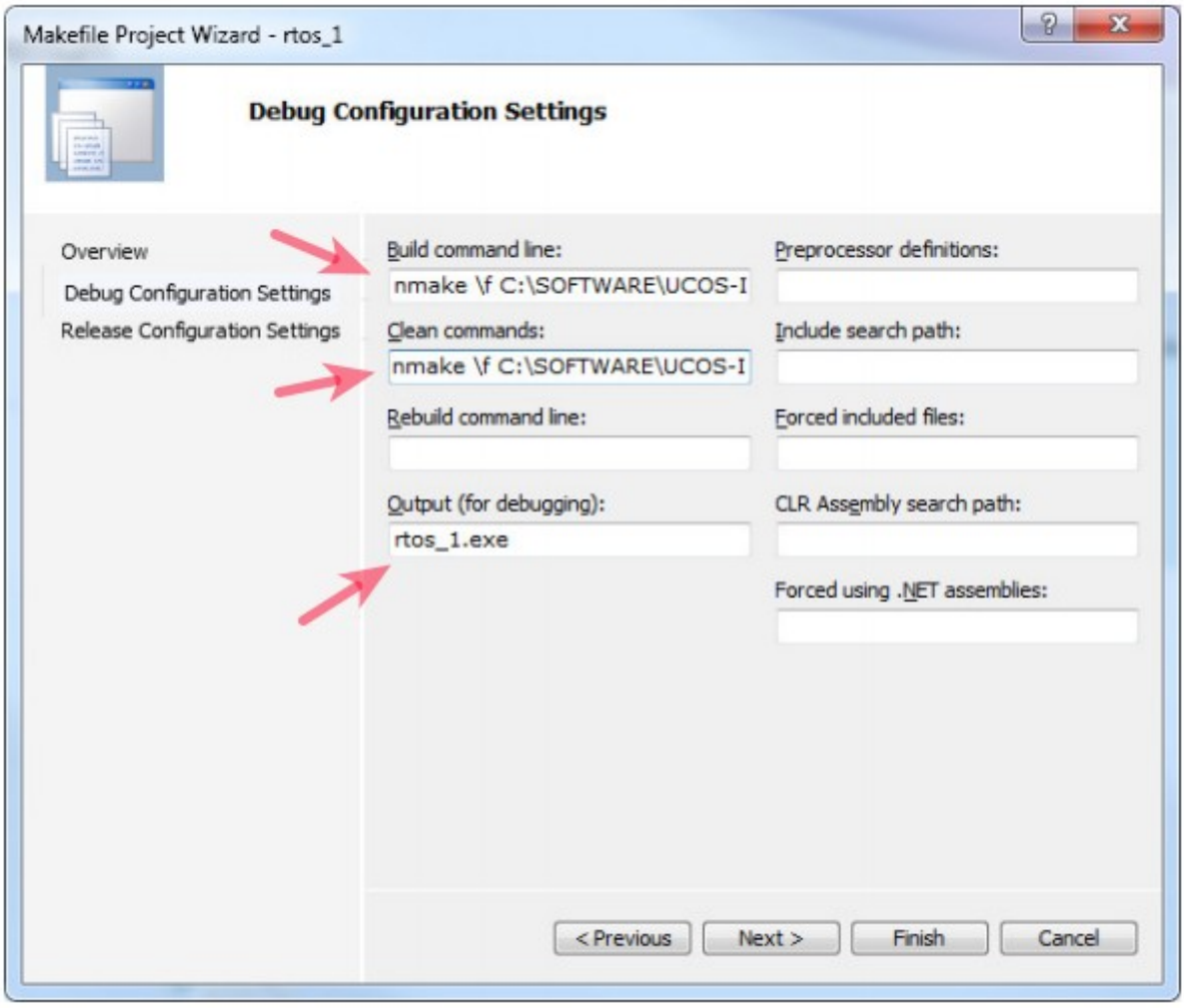
• Start Visual Studio 20xx -

• Create new **Makefile project** (File => New => (General) Makefile project):



- Project name have to be same as you have in variable EXAMPLE in your makefile
  - Name project as rtos_1 in this first case
- Change the location of project files (files needed by Visual Studio, not C-files) to **somewhere else than your profile (C:\users\...) and not to U: drive** C:\temp would be fine
- For make file project you have to define where your makefile.vc exist and create a compilation command

For make file project you have to define where your makefile.vc exist and create a compilation command:



Be VERY careful of the syntax of the command and directory path especially !!!!
Use the following as example

nmake /f C:\SOFTWARE\uCOS-II\Ports\80x86\WIN32\VC\exercises\RTOS_1\makefile.vc all

nmake /f C:\SOFTWARE\uCOS-II\Ports\80x86\WIN32\VC\exercises\RTOS_1\RTOS_1\makefile.vc clean

If you made mistakes here these can be corrected later from project **properties** submenu **nmake**

Some remarks to make your working easier

Command line tools

Create a shortcut from Visual Studio Native x86 command to desktop. Modify a parameter Start In from shortcut Properties to point to tour exercise directory (RTOS_1)

Modifying uCOS C-files with Visual Studio

Add all your project C- and h files (from firectory **RTOS_1**) (Optionaly also from , **Ports/src** ja **SOURCE**) to your Makefile project (Add … Existing files)

Automatic code completition can be activated by adding all three directory places to toolchain search path

Proect Properties - VC++ Directories - Include Search Path