Transferring project from <Disk>:\ChibiStudio to another working directory

Setting ChibiStudio to disk different than C:

Why? :-)

Somebodies are fine with placing everything on drive C: and having a big mess on the system drive. I always try to separate system disk from my programs like IDEs, CADs, editors, etc and from my data. In this case if system crashes I have bigger chance to keep my beloved programs and data not crashed with the system along. Besides, today there are many cases when on workstations system located on quite small SSD. Placing everything on it makes it smaller and less reliable because of often writing data. If you don't care about the mess, SSD size, and reliability you can pass this part. ;-)

Here, what should be done for moving ChibiStudio to another drive

1. Change paths in <Drive>:\ChibiStudio*.bat files

   Instead of

   set PATH=C:\ChibiStudio\tools\gnutools\bin;%PATH%

   put the following

   set PATH=%~dp0\tools\gnutools\bin;%PATH%

   %~dp0 will insert your current ChibiStudio directory in place.

2. Links to the bats in ChibiStudio directory. Change in their parameters C: to <Drive>: as well. You might want to change their icons also, get them from <Drive>:\ChibiStudio\eclipse

3. In file <Drive>:\ChibiStudio\eclipse\configuration\.settings\org.eclipse.ui.ide.prefs change paths in the following string
   RECENT_WORKSPACES=C\:\\ChibiStudio\\workspace161\nC\:\\ChibiStudio\\workspace26\nC\:\\ChibiStudio\\workspace_trunk\nC\:\\ChibiStudio\\workspace30
   to
   RECENT_WORKSPACES=<Drive>\:\\ChibiStudio\\workspace161\n<Drive>\:\\ChibiStudio\\workspace26\n<Drive>\:\\ChibiStudio\\workspace_trunk\n<Drive>\:\\ChibiStudio\\workspace30

4. in OpenOCD on ST-Link V2-1 (prompts for .cfg target configuration) and others in this external tools configuration need to change arguments from
   -c "telnet_port 4444" -s "C:\ChibiStudio\tools\openocd\scripts" -f "interface/stlink-v2-1.cfg" -f

"${file_prompt}"
to
-c "telnet_port 4444" -s "<Drive>:\ChibiStudio\tools\openocd\scripts" -f "interface/stlink-v2-1.cfg" -f "${file_prompt}"

5. Create two batch files for debugging in your project directory:
    1. predebug.bat (you can give it any name)

```
SET OPENOCD=E:\ChibiStudio\tools\openocd
%OPENOCD%\bin\openocd.exe -s %OPENOCD%\scripts -f %OPENOCD\scripts\board\st_nucleo_l1.cfg
-f rtos.cfg
```

    (second and third line here are actually one line)

    2. rtos.cfg

```
$_TARGETNAME configure -rtos ChibiOS
```

6. You have to run this batch file before starting the debugging. Only once when you start the work.

After that the whole ChibiStudio located in correct place and when you start it shows all available projects. However, to open any particular one you have to delete it from the tree and import again because the projects still “remember” their parent directory C:\ChibiStudio….etc

Substitute a new drive letter in Windows 10

Substitution is useful if you have more than one “workstations” and one of them does not have a drive with the same letter. While looking for a reminder how to use command SUBST, I found that this command has problem with Windows 10. There are some solutions described on different forums and discussions. Finally found psubst

https://github.com/ildar-shaimordanov/psubst

The author says it permanent substitution that keeps working even after restart computer. Did not restart my current computer, but, at least, I don’t see a problem with finding substituted drive in my system.

Transferring the project

1. Export project as archive file to the desired location.
2. Unpack the project. Note, that all files for the chibios (os directory), board config and test (if you transfer original chibios demo projects) are in the same directory with the project.
3. Import the project from the new directory to ChibiStudio. You need to delete original project created in ChibiStudio directory from the tree of projects. (it won’t be deleted physically until you request so)

4. Open Makefile and change string

```
# Imported source files and paths
CHIBIOS = ../../..
to
CHIBIOS = .
```

   You can also change project name in string

```
# Define project name here
PROJECT = ch
```

5. Open the project Properties → Resource → Linked Resources → CHIBIOS to your current project directory
6. In the project delete links to boards, os, test. When you delete the links, they will be changed to the real directories in your working project directory.
7. Now you got compact located project with it's own copy of ChibiOS, board configuration that you can commit to your favorite version control system. Don't forget to clean project before first committing that you would not put a bunch of object and listing files to the server.