Department of ACES, SHU

Embedded Systems Design

Semester 2 academic year 2017/18

**Reference Manual**

**The purpose of the manual is to present information that is relevant to various stages of the coursework development.**

**In particular,**

**appendix 0 - Setting up your virtual machine - p.2**

**appendix A - how to generate a Keil MDK-ARM v5 project from the STM32CubeMX (with a special note for LabA2) - p.6**

**appendix B - how to use an STMCubeMX-generated project in the Keil MDK ARM v5 - p.9**

**appendix C - how to use printf with the Cortex M embedded trace macrocell and Keil MDK-ARM v5 - p.11**

**appendix D - how to debug code in Keil MDK-ARM v5 - p.14**

**appendix E - how to archive a completed project for future reference and/or cloning - p.15**

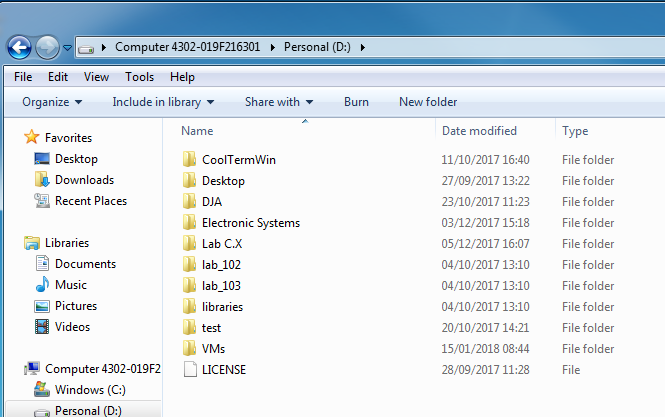
**appendix F - how to clone a complete STMCubeMX+Keil MDK-ARM project - p.16**

**appendix G - how to connect the on board USB CDC device to a host computer using a virtual COM port - p.17**

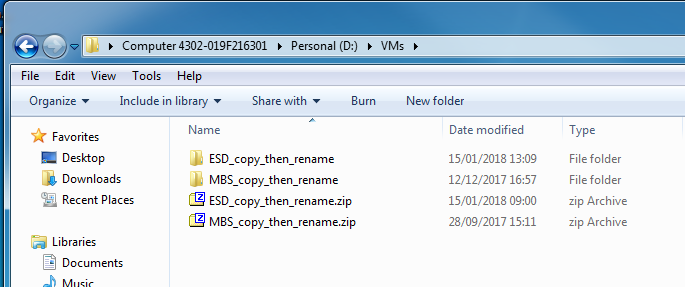
**appendix H - how to connect the on board USB host to an external mass storage device - p.21**

**Appendix 0**

**Running the virtual machine at the first time**

Navigate to the D: drive (you can see a different directory structure at your computer) 

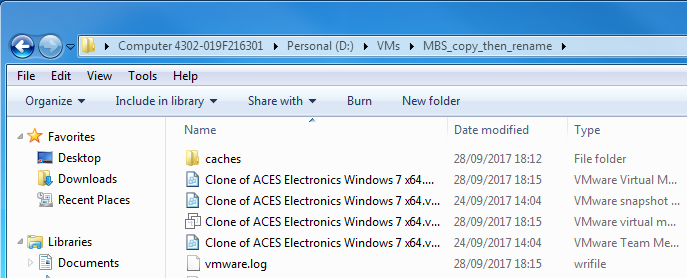
and into the directory VM as shown below



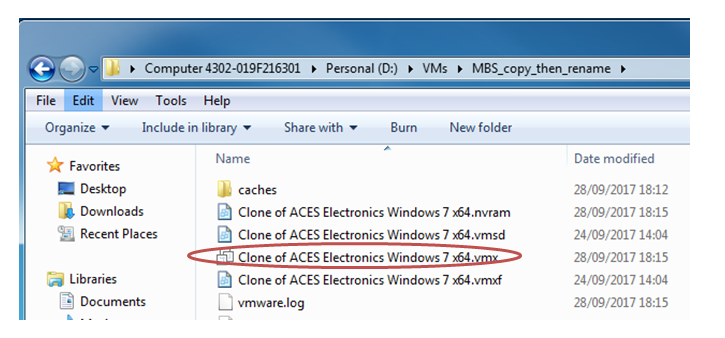
You can see the archive and unpacked virtual machine's files.

***Please copy the unpacked directory and rename it to your initials or student's ID number.***

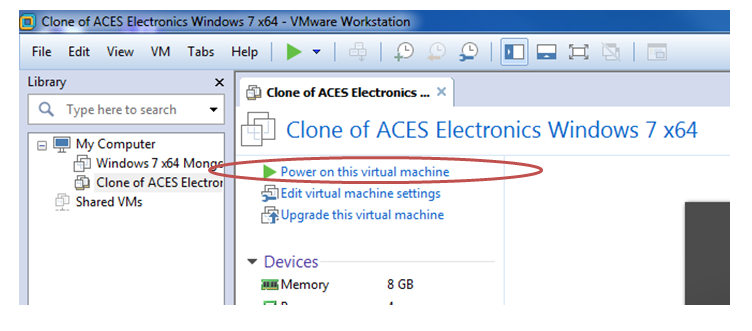
Navigate inside your copied and renamed directory:

****

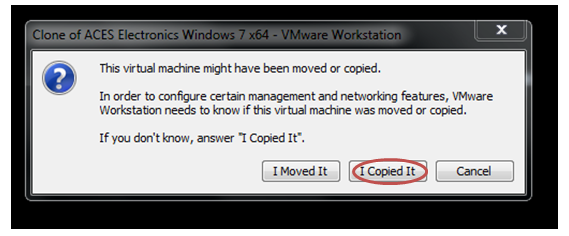
You may want to have clearer view to start the VM. Expand the width of the "Name" column then double click on the file shown below

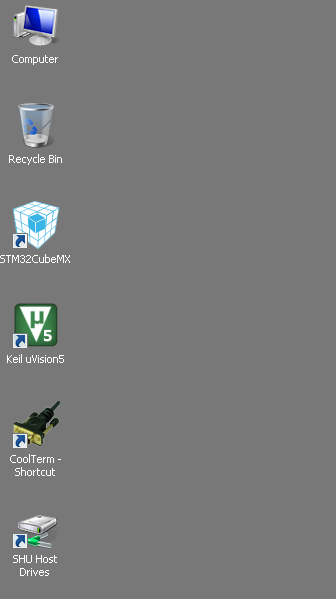


When the hypervisor (VM manager) loads, you will see the following screen where you need to double click the shown entry to start the VM



(Some of you may be presented with a dialog window where you need to click on "I copied it" as shown

 )

When the VM operates you will see the conventional Windows 7 environment. The icons you will use most are located in the top left corner.

They are (from the top to the bottom) -

file manager of the virtual machine;

recycle bin;

shortcut for the STM32 CubeMX;

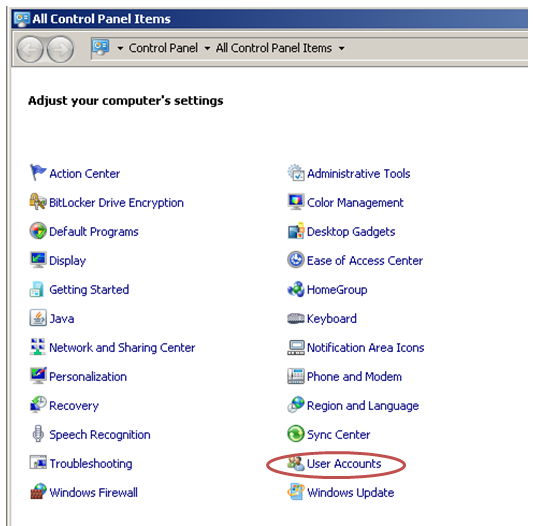
shortcut for the Keil MDK (uVision 5);

terminal software to print from the board to the PC (Lab C and mini project);

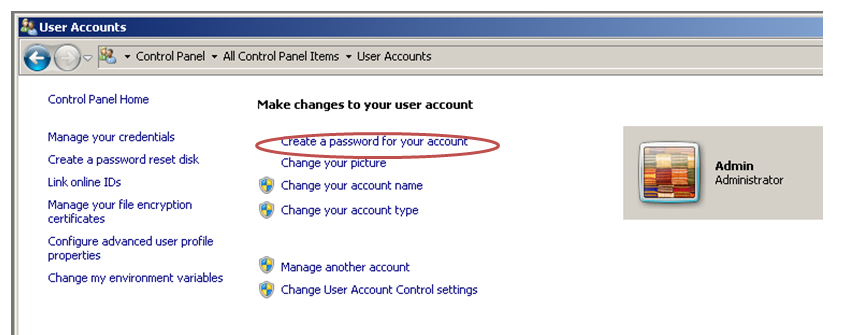
the gateway to the drive D: of the actual physical machine and your network drive F:

***Please set the password to access the virtual machine***

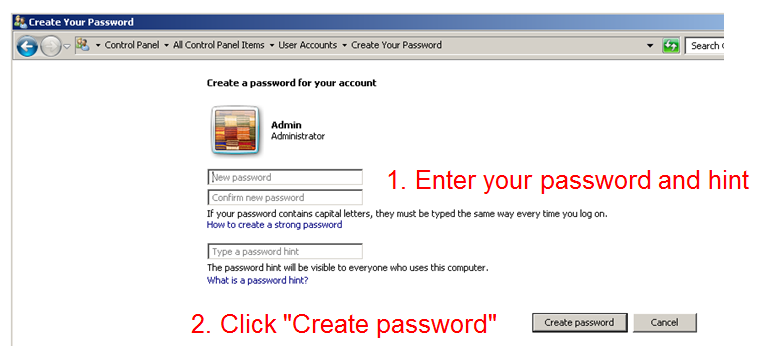
Click "Start" then "Control panel". Select "User accounts" as shown



Click "Create password for your account"



Select your password



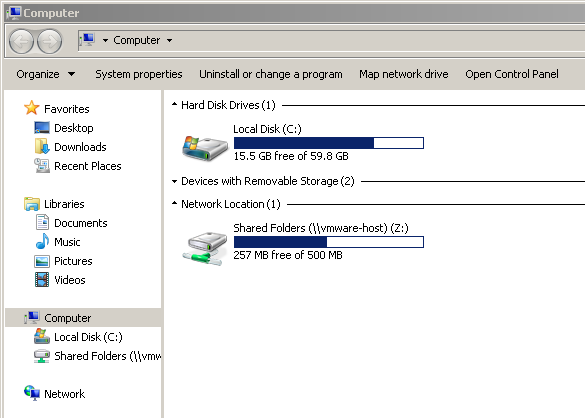
Please be advised:

1) if the password is forgotten, your work is not recoverable;

2) if you do not set the password and someone copies your files, triggering a plagiarism case, all of the students involved will get zero credit.

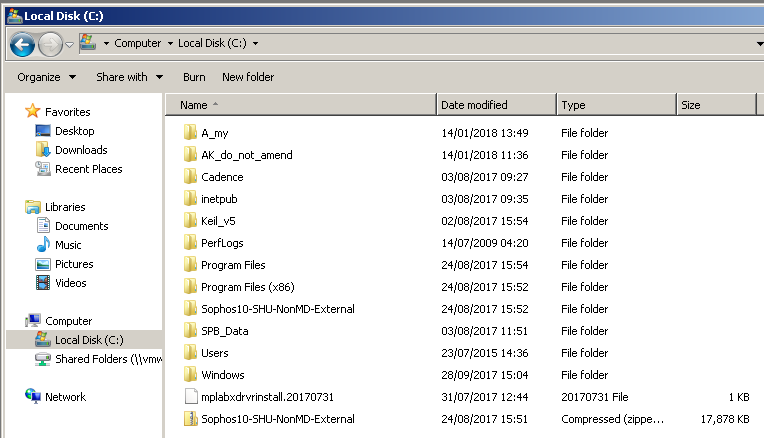
**Navigating your virtual machine**

"My computer" view

****

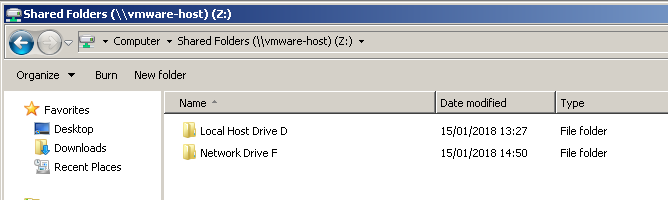
The C: is the virtual drive that you cannot see from the host computer.

It contains



Directory ***A\_my*** at the very top is for all of your stuff. Please do not forget to backup your stuff (this directory) at the end of every session to a USB memory stick or your network drive (just in case someone deleted your virtual machine by mistake).

You can also access the D: drive of the host computer and your network drive via "Shared folders":

****

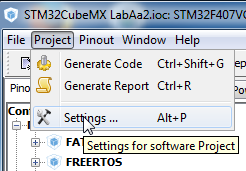
**! Do not forget to copy your *A\_my* folder to your network drive (or to the D: drive of the host computer then to your memory stick) !**

**! DO NOT FORGET TO SHUT DOWN YOUR VIRTUAL MACHINE AFDTER COMPLETING YOUR WORK !**

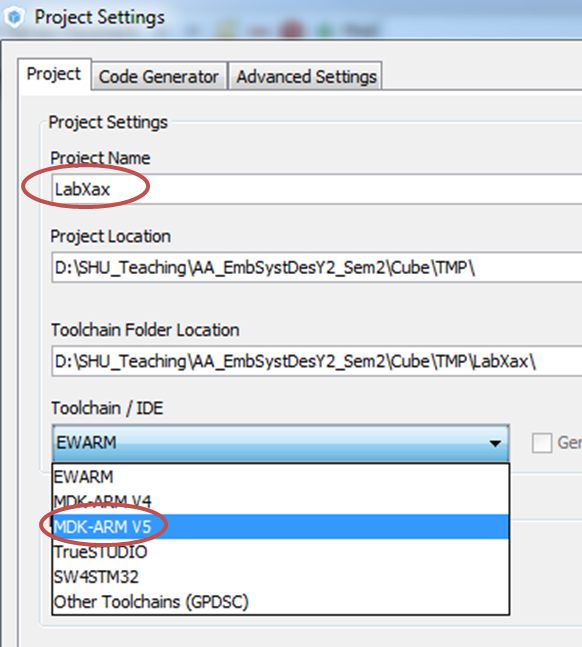
**Appendix A**

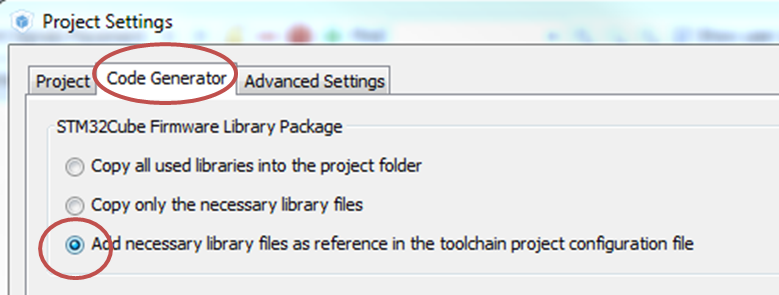
**How to generate a Keil MDK-ARM v5 project from the STM32CubeMX (with a special note for LabA2)**

One can set the required options from Project/Settings…:

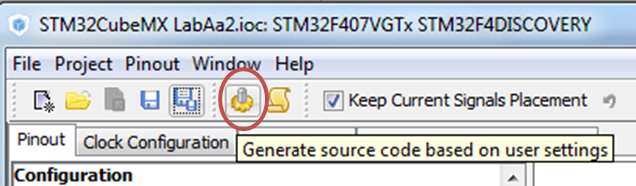


Please check that the options are selected as follows (circled are options that need to be changed if a new project was created from scratch; use the default values for all the other options):



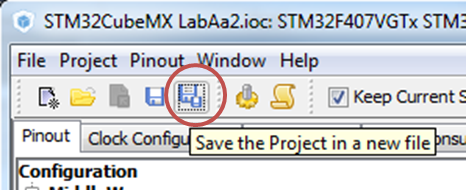


Click ***Generate source code based on user settings***

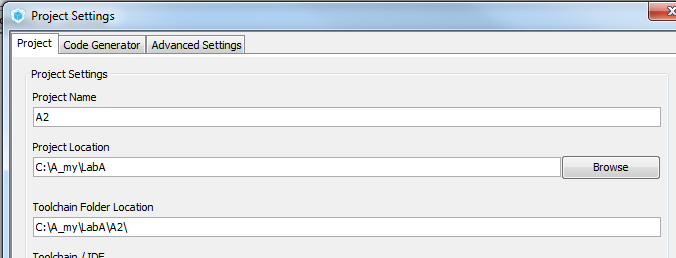
******

***Special note for the LabAa2:***

after opening the provided template (***A2.ioc*** file), please first "***Save the project in a new file***" (that is required because the file was prepared at a different computer)



The ***Project settings*** dialog (like shown above) will come up.

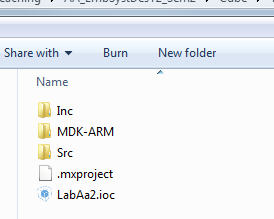
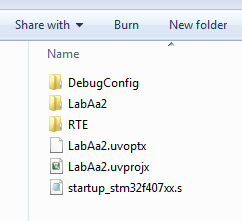


Put the mouse cursor in the field ***Project Name*** (after **A2**), hit ***Enter*** then click OK - all done.

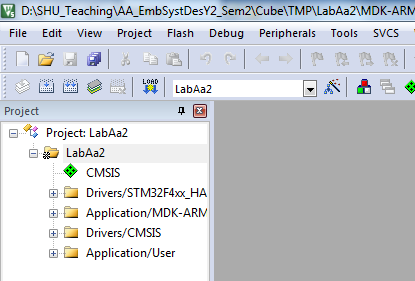
**Appendix B**

**How to use an STMCubeMX-generated project   
in the Keil MDK ARM v5**

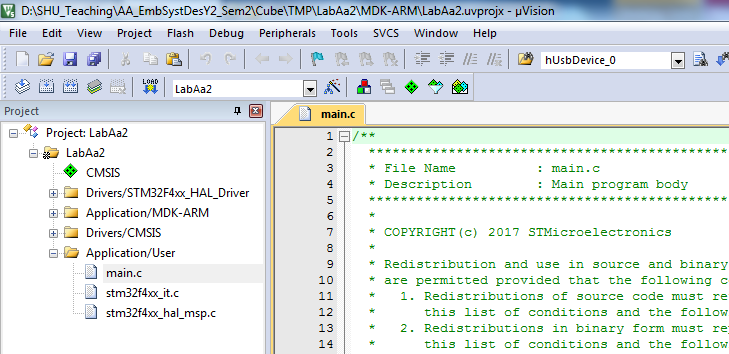
After generation of the MDK-ARM project, the relevant project folder will have the following content (on the left) with the following content of the folder MDK-ARM (on the right)

The file with extension ***uvprojx*** above on the right is the project file; double click on it brings about the MDK-ARM IDE.



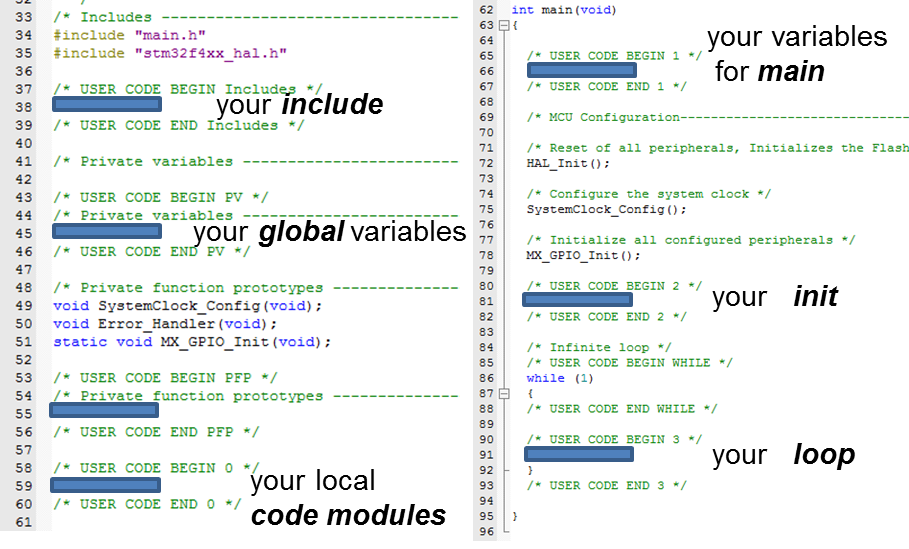
You may want to explore content of all the folders but you will only work with the bottom one that contains the following editable files



**REMEMBER AND FOLLOW**

if you want to be able to amend your CubeMX parameters then re-generate the MDK-ARM project, ***STRICTLY*** observe the placeholders in the CubeMX generated code.

Blank template generated by CubeMX with the suggested use of placeholders



Why is it important? Because you may want to re-generate the MDK-ARM project later when you amend your CubeMX project. Anything that is located outside of the placeholders will be deleted without any mercy and/or warning.

Lab A2 includes an exercise in migrating the developed C code from the Lab A1 MDK-ARM project. You need to go through all the placeholders and copy any added statements into the Lab A2 code. During the exercise you will find how I used various placeholders to add custom statements.

**Appendix C**

**How to use printf with the Cortex M embedded trace macrocell and Keil MDK-ARM v5**

The MCU we use is equipped with an embedded trace macrocell (ETM) which is capable of sending/receiving information to a host computer at the run time (some cheaper STM32F microcontrollers do not provide such an opportunity). To get this information displayed on a PC, however, one needs an additional hardware bridge (e.g., ST Link MCU on the Discovery Board) and specific software (e.g., included into the MDK-ARM).

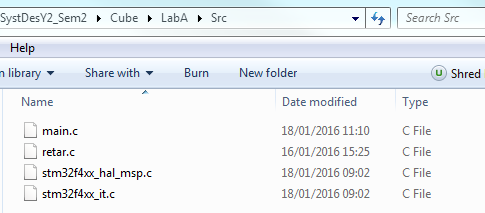
The following actions are required in order to utilise the ETM for ***printf***:

1. retargeting the standard C output ***stdout*** to the embedded macrocell (write a retargeting subroutine);

2. enabling the ETM in MDK ARM settings before compiling and uploading the code;

3. opening a window in MDK ARM to observe the output.

1. In order to use ***printf*** function, it is necessary to copy file ***retar.c*** into the source files folder of the project generated by the STM32CubeMX: please locate this file in the Src folder of the Lab A1 1 and copy it over to the Src folder of a new project



Additionally, the ***main.c*** file should include the following line shown along with the placeholders to reference this file:

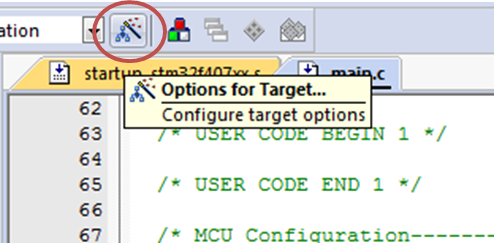
**/\* USER CODE BEGIN Includes \*/**

**#include "retar.c"**

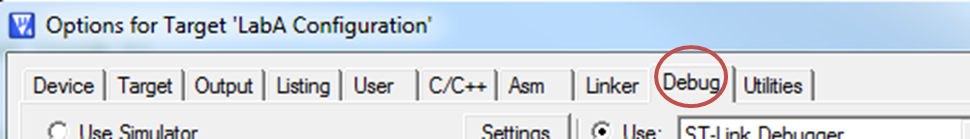
**/\* USER CODE END Includes \*/**

2. Use of ***printf*** additionally requires enabling single wire debug hardware on the STM32F4 microcontroller, and telling the on-board programmer what frequency it will be running at. Please follow these steps:

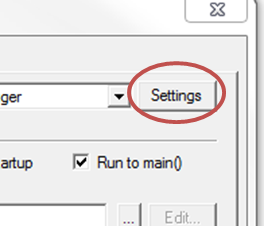
click on "Options for target.." button



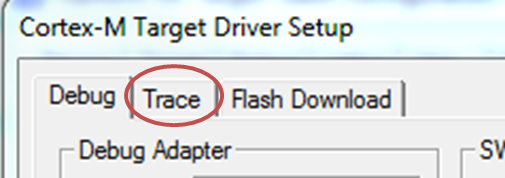
then click on the "Debug" tab



then click on "Settings" button

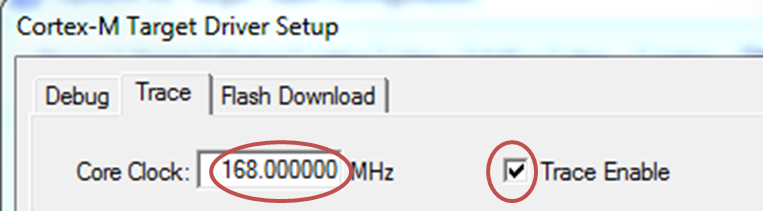


Select the "Trace" tab

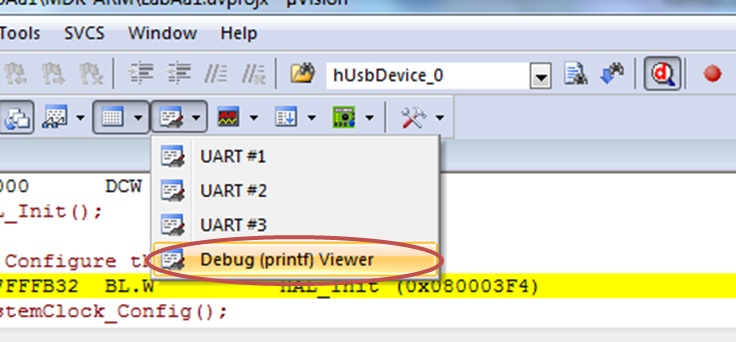


Check the "Trace Enable" checkbox and enter the correct core clock frequency

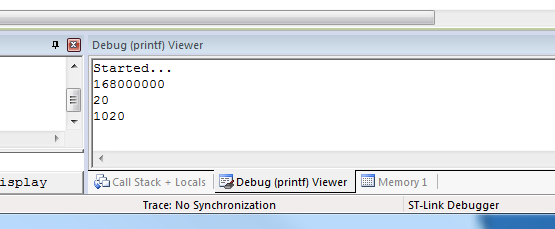
(the frequency that you achieved in the STM32CubeMX; below the largest available 168MHz clock frequency is set; **but YOU need to set the FCLK/HCLK Cortex clock frequency from YOUR STM32CubeMX project** )



3. The window for ***printf*** can be opened in the ***Debug*** mode:



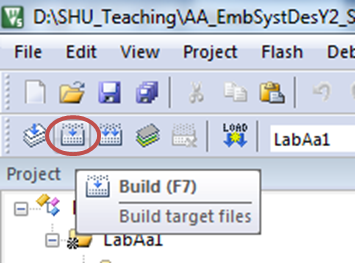
then the output becomes visible in the bottom right corner



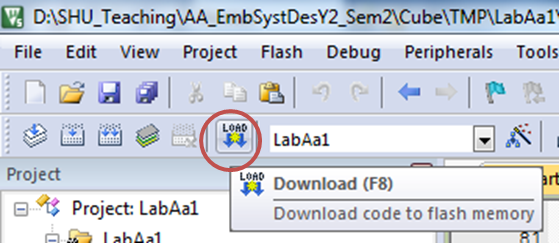
**Appendix D**

**How to debug code in Keil MDK-ARM v5**

Getting the C code compiled into the executable code

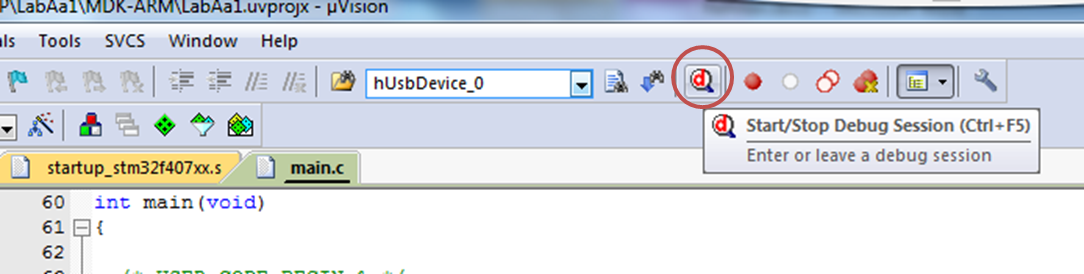


Uploading the executable code to the MCU

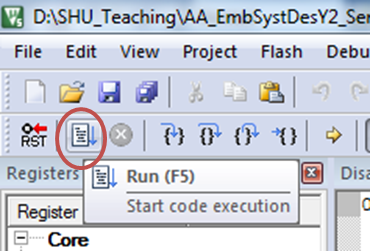


Running the code on the MCU  
( ! please do not press the black pushbutton on the board as you are used to   
as you will be unable to ***printf*** and control the code execution ! )

clcik Debug



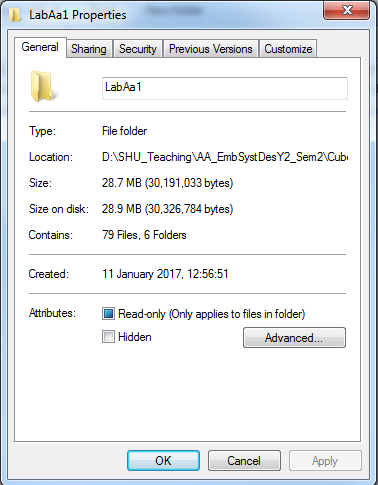
then start the code:



**Appendix E**

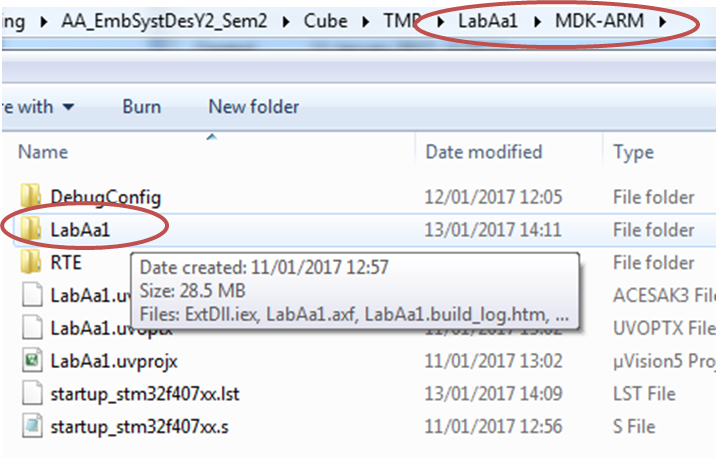
**How to archive a completed project for future reference and/or cloning**

When the project is first compiled its size grows to tens of MBs (to 28.7 MB below)



This size can quickly overfill your network disk quota; moreover most of the files generated by the compiler can be easily restored by re-compiling the code.

For this reason after completing an assignment, please feel free to delete the folder with the code generated by the compiler as shown below (here it occupies 28.5 MB)

****

**Appendix F**

**How to clone a complete STMCubeMX+Keil MDK-ARM project**

***Abstract -*** *using Cube is convenient and time saving but one needs to make sure that the Cube and MDK files are fully coordinated. In this case, say, adding or removing a peripheral is easy and straightforward. Below is a detailed procedure of how to achieve this coordination when moving from one completed assignment to another. (Think of PCB design - you always have one file for schematic diagram, and another file for the board. If they are not in sync, the design is wrong.)*

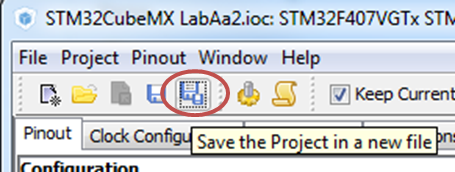
**Introduction**: On quite a few occasions developers want to clone (make a complete copy) of a particular project; most commonly to create something akin to the "restore point" on the computer from which the development can resume later if the newly added features completely destroy the application. In this coursework cloning is used to get credit for a completed assignment and carry on further developments.

**Step-by-step procedure to follow:**

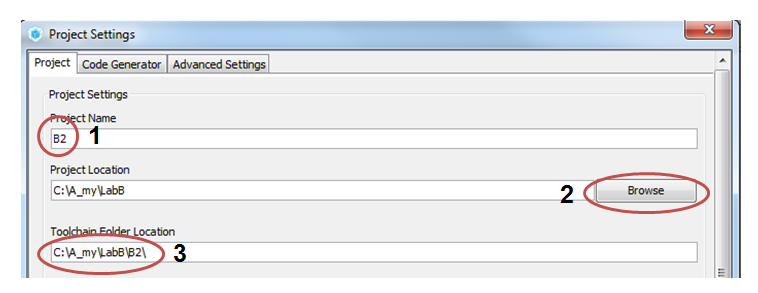
The easiest way to accomplish cloning for a CubeMX+ARM-MDK combination seems to be the following one:

- open the CubeMX project with the extension .ioc

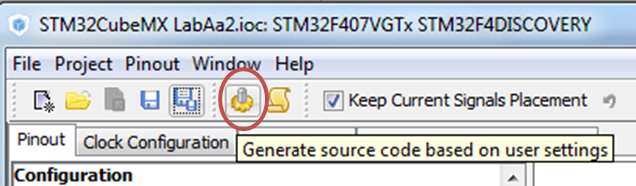
- Click ***Save the project in a new file***



- type in the **project name** to the required one (like B2, C1 etc - look for the item **1** below), and navigate to the directory where you want the new **project folder** to be located (like LabB, LabC - look for item **2** below); within the selected project folder new folder will be added **by Cube** (as shown by item **3** below) then click OK at the left bottom part of the window



- generate the MDK-ARM project from the CubeMX

******

(at this point you have a valid MDK project but it lacks the code you already developed for the previous assignment; ***do not open*** this project yet).

- copy the developed code (full content of the ***Src*** folder of the completed assignment into the ***Src*** folder of the newly generated folder)

- from Cube generate the code ***again***, and now ***open*** the MDK project when the code has been generated.

From now on any changes in the Cube project can be inserted into the code without breaking things. ***Make sure you add your code only within the Cube placeholders.***

**Appendix G**

**How to connect the on board USB CDC device to a host computer using a virtual COM port without using MDK-ARM**

***- If you want to do it from home you first need to install the VCP driver as follows***

Download the driver from the STMicroelectronics web site:

**STSW-STM32102** STM32 Virtual COM Port Driver

<http://www.st.com/web/en/catalog/tools/PF257938>

by clicking button "Download" in the bottom right corner of the page.

I have downloaded file ***stsw-stm3102\_VCP.zip*** .

Follow the instructions available in the text file within the archive; here are these from my downloaded archive:

1- Unpack the attached zip file

2- Run your "VCP\_V1.4.0\_Setup.exe"

3- Go to Your installation directory - for example,   
C:\Program Files (x86)\STMicroelectronics\Software\Virtual comport driver

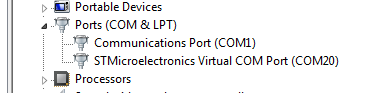
4- Go to Your OS version directory ([Win7] or [Win8])  
+ Then :

- Double click on dpinst\_x86.exe if you are running a 32-bits OS version

- Double click on dpinst\_amd64.exe if you are running a 64-bits OS version

       + Follow the onscreen instructions.

To check whether the installation was successful, connect the micro USB socket of the STM32F4Discovery board to the computer and open the ***Device Manager***. It should show an STMicroelectronics VCP like below (the number of the VCP on your computer can be different)

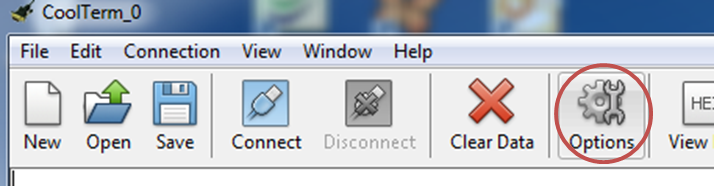


***- there is a need to use some terminal software on the PC side do display the received messages instead of the MDK-ARM***

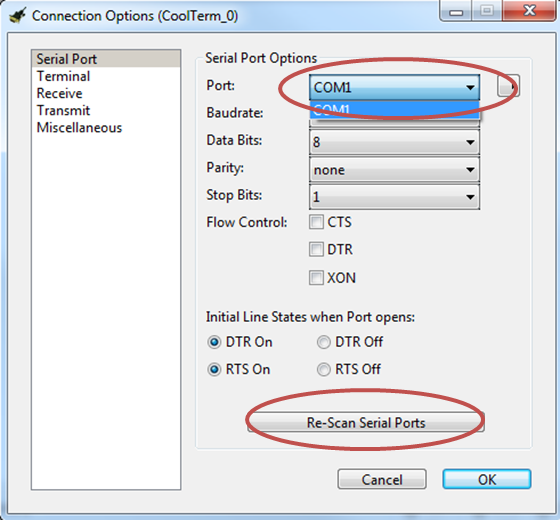
***I strongly recommend to download CoolTerm*** from here <http://freeware.the-meiers.org/CoolTerm_Win.zip> (or from the BlackBoard where I placed a copy for convenience) and use it. It does not require administrator's rights and allow re-scanning the USB ports if something goes wrong (cable disconnects, buggy firmware etc).

After unpacking the archive start the ***CoolTerm.exe***

First click on the options

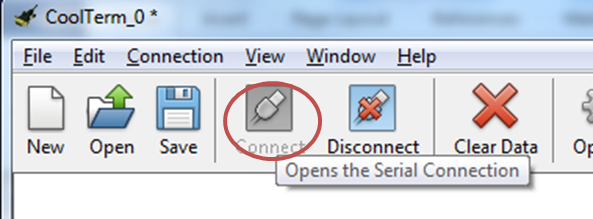


then select and appropriate virtual com port from the drop-down list

******

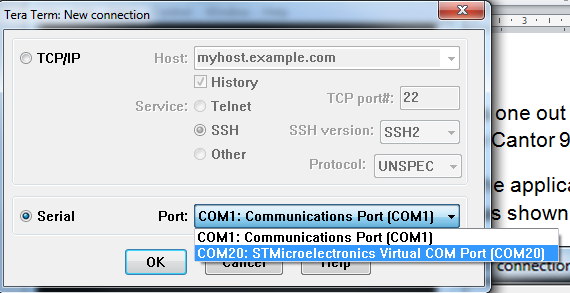
As you can see from the picture, you can re-scan serial ports anytime without the need to restart the application. If your board is not visible from the menu then something is wrong with it or the USB code.

Click OK then Connect and you are set

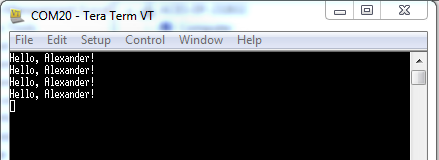


***Another very respected and reliable piece of software, installed on all of the lab's computers, is called TeraTerm.***

First open the application by e.g. typing TeraTerm in the start box. On the welcome screen select serial connection and a STMicroelectronics virtual COM port to connect to as shown below:



If the CDC device code is running and you have retargeted the *printf* output to the CDC device, you will instantly see picture like below with the addition of timestamps.



The downside of the TeraTerm is that it is great for robust applications and not for their development as it cannot check the availability of COM devices after it has first started. You may need to restart the TeraTerm or/and re-plug the cable and/or restart the PC to get connected again after any failure.

**Appendix H**

**How to connect the on board USB host to an external mass storage device**

One can use for the MSC host either a USB flash drive or an SD card with a USB card reader (such a reader costs ~ £1 including delivery on eBay if you happened not to have one already). ***Make sure you backup all your files on your flash drive/SD card for the case if the development goes wrong. The flash drive should be formatted using FAT32 file system (most drives with the capacity below or equal to 32 GB are formatted this way at the factory.)*** Thephotograph below shows possible options

You will likely need another piece of hardware to connect the USB A male connector of the USB flash drive/card reader to the micro USB female connector on the STM32F4Discovery board. This can be either USB On-The-Go (OTG) cable, or USB OTG external adaptor or USB OTG internal adaptor that is placed inside the USB A male connector

You will be very unfortunate if you need to pay for any of these items more than ~£1 excluding postage and packing.

However there was a little hurdle that took a lot of time from one student for which I am sorry.

He used a 64 GB flash drive, and it did not work at all. However the same Cube/MDK project worked without a problem with a smaller drive.

Neither the STM application note ( **tinyurl.com/hf2sdu6** ) nor the application note from the FatFS developer ( <http://elm-chan.org/fsw/ff/en/appnote.html> ) mentioned any volume size restrictions.

It turned out that Microsoft restricted the size of FAT32-formatted flash drives (FatFS works with this format) to 32 GB in order to promote the NTFS file system (<http://www.makeuseof.com/tag/format-large-hard-drive-fat-fat32/>). For this reason, if you got a 64 GB or higher capacity flash drive, it is most likely to be formatted using NTFS. You can re-format your drive but you might lose data in process; I think it is to be more convenient if you simply use a smaller capacity drive (the smallest one I have tested was a now rudimentary 128 MB flash drive and it worked fine).