ECEN202 2019 Lab Session 6

Assembler, timers, polling and counting

Part I: Introduction

During this first session, the idea is to become acquainted with the fundamentals of the A51- KEIL uVision5™ development system which you will use to write, compile, and download programs to the AT89C51AC3 chip development board. We will also learn how to write "assembly" language programs. The task is to write some code to implement a digital clock. Here we will be using one of the hardware timers and polling methods to control the timing of the program execution.

To run KEIL, click the `Windows Icon -> All Programs -> UV5 -> UV5` or select the Windows Icon (In ALL Programs towards the top of the quick start icon is Keil uVision5) Alternatively, use the search function on the taskbar and use the search string, "uVision."

Now it is time to create your first project. From the menu, click `Project -> New uVision Project`. Name this project `Clock1`, and set the Project location to a folder where you want to store the project.
KEIL will then ask you to select your target Device.

- Device – Atmel – AT89C51AC3
- Click OK.

Next it will ask you if you want to copy a version of 8051 Startup code. Select **"No"**
KEIL should now be divided into 3 windows. On the left is the **Project Area**, which displays the various files that your project will be making use of. Expand the folders to view the files. In addition to the Target folder, there should be a folder called Source Group that contains the 8051 source files. To add more files to your project, right-click on the corresponding folder in the Project Area, and click "Add files to group" or "Add group". Add "**clock1.a51**" to the group and double click the file to

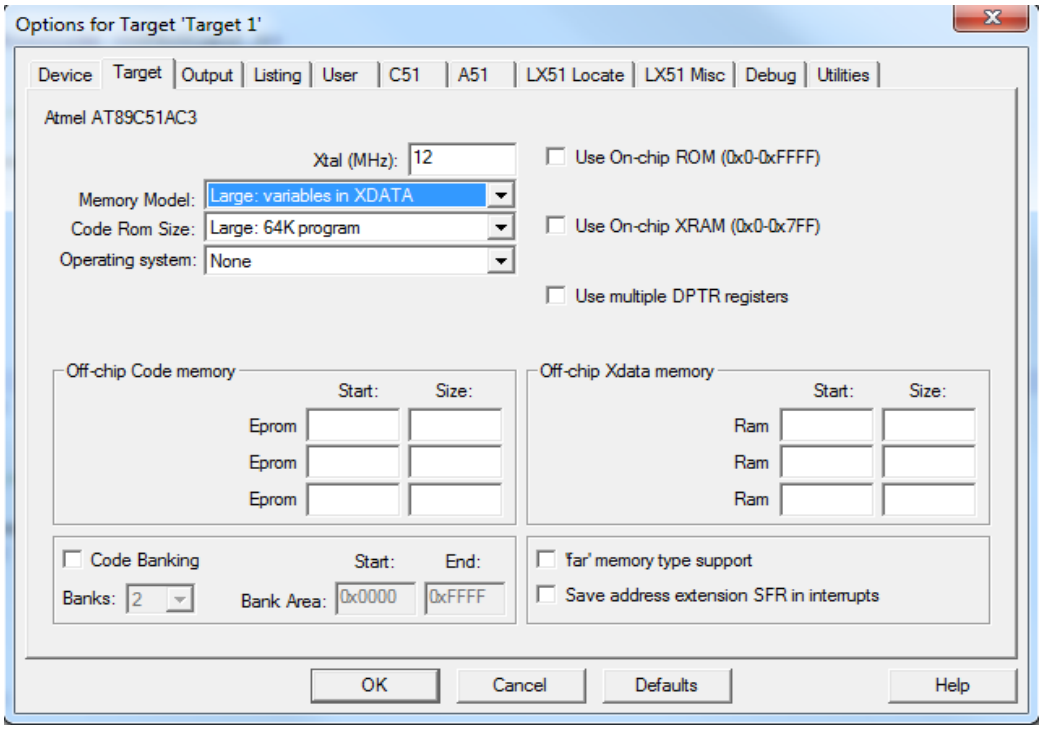
open it for editing. The bulk of the window should be occupied by a text editor, where you will write and edit your code.

You can have multiple files open for editing in the main window, and each will have a tab on the top of the window. You can close the files by clicking on the 'x' box at its top right corner (Not the 'x' for the KEIL program!). The bottom window will display **Build Output**, showing you memory statistics about your code once you compile it, and any compiler errors that come up in the process. The size of all of these windows can be adjusted by clicking and dragging their borders.

Now open file "hello.a51" and copy the text into "clock1.a51". Close "hello.a51". Save "clock1.a51", this is the code of the lab1 where we displayed the message "HELLO". We will now compile (assemble) the code and run it to check that your hardware is still working.

Before compiling your project, you must configure some important flash tools. From the toolbar, select `Flash → Configure Flash Tools,` and make the following changes in the following tabs:

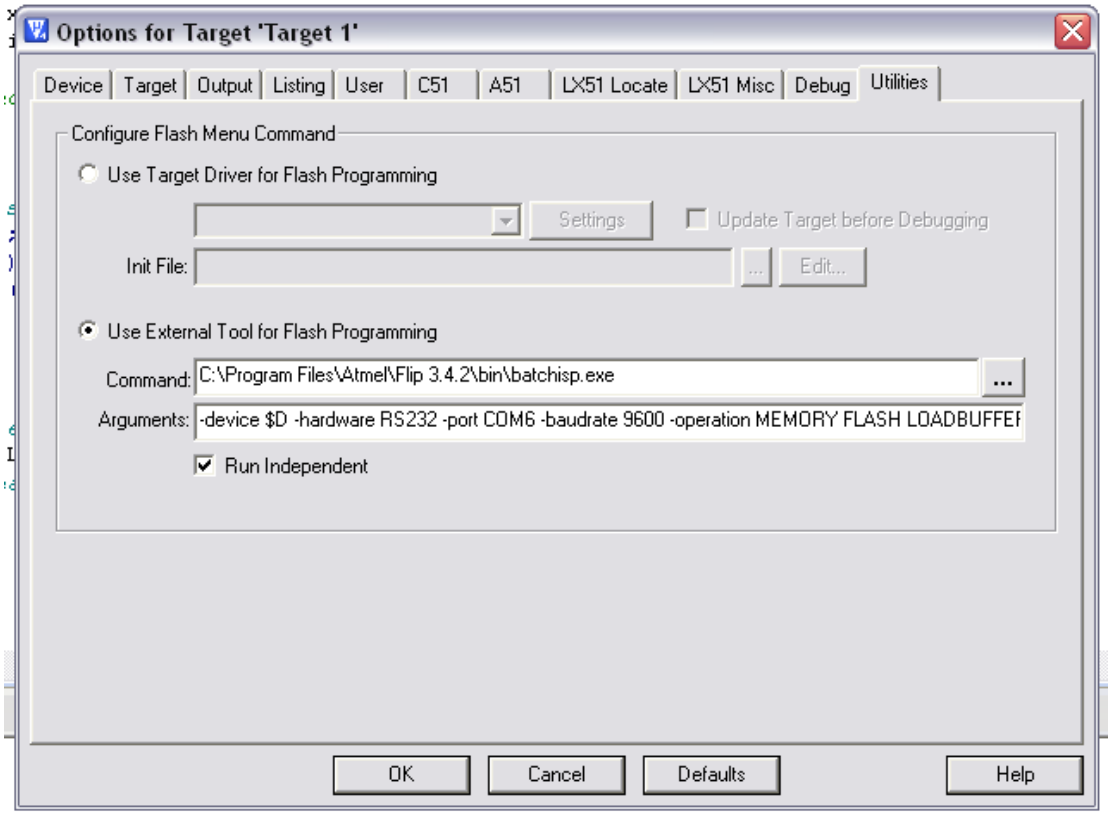
- **Output** - Tick the "**Create HEX File**" box.
- **Target** – Memory Model: Change to Large variables in XDATA.
  - Change Xtal (MHz) to 12.



Configuring the flash tools

Flash Tools

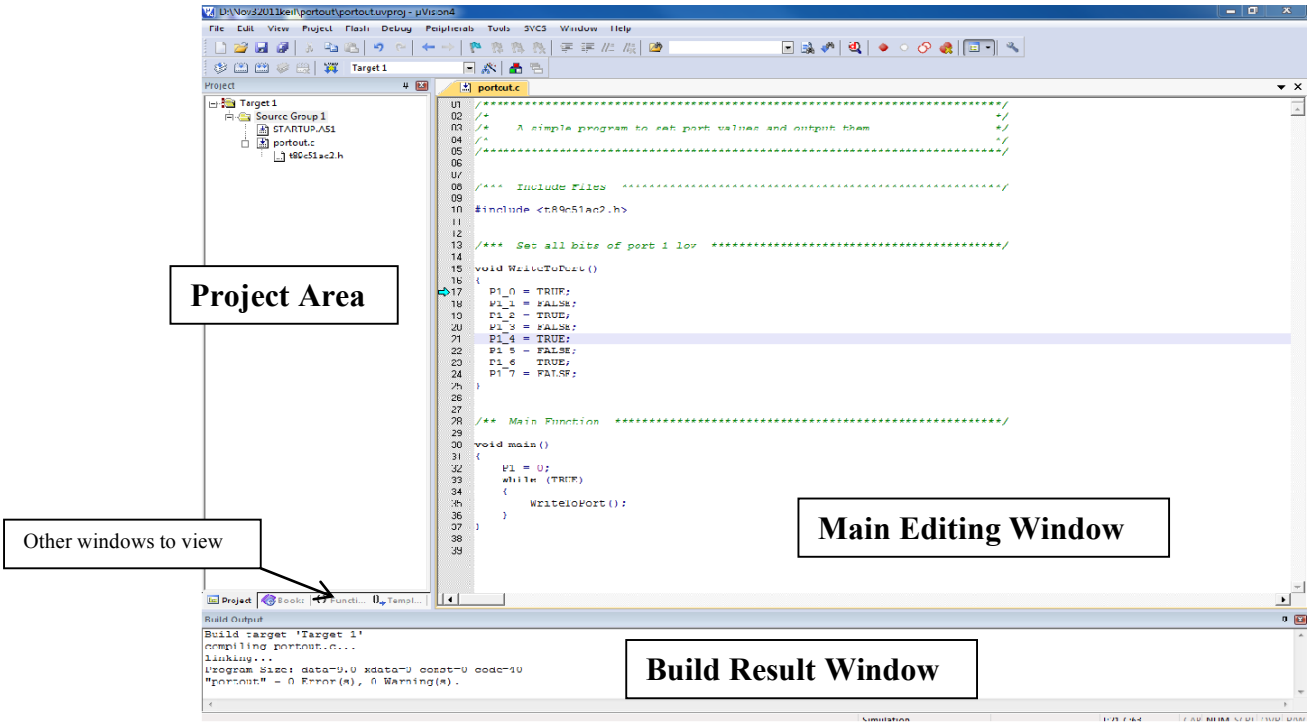
- **Utilities –** Select external tool for programming.
    - Click on the button with 3 full stops.
    - Search for the executable under c:/Program Files(x86)/atmel/Flip 3.4.7/Bin **(NOTE: the version of FLIP may have been changed since writing. Use the version that is available.)**
    - Select BatchISP.exe and click OK.
    - Under Arguments change the COM port to the COM port the micro is plugged into, and the baud rate to 9600. *(Left click on the windows icon and go into device manager. It will not let you change settings as you are not an administrator, so press ok if it prompts you. The window will show up, and you can check the COM ports in the listing.)*
    - *Tick the "**Run Independent**" box*
    - In the arguments text box, where you see "%H", change this to ".\objects\<NameOfFile>.HEX" **(In this string, do not add the quotation marks, and <NameOfFile> should be the name of your project instead. No brackets here: ".\objects\example.HEX")**
    - In the **Debug** tab, check the "use" box on the top right.
    - Click OK.



Utilities under the Flash Tools window

To build/compile your project, from the menu click `Project → Build Target` or press F7. If the compile was successful, your Build Results should tell you there were no errors. If you did get errors, congratulations, you've made your first mistake. Ask your demonstrator to help you fix it.

Note: when coding, you should check the compiler output very carefully. If the output of the compilation results in "**Target not created,"** this is a sign that something has gone wrong. It is useful to scroll up the build result window to see where things have gone wrong.



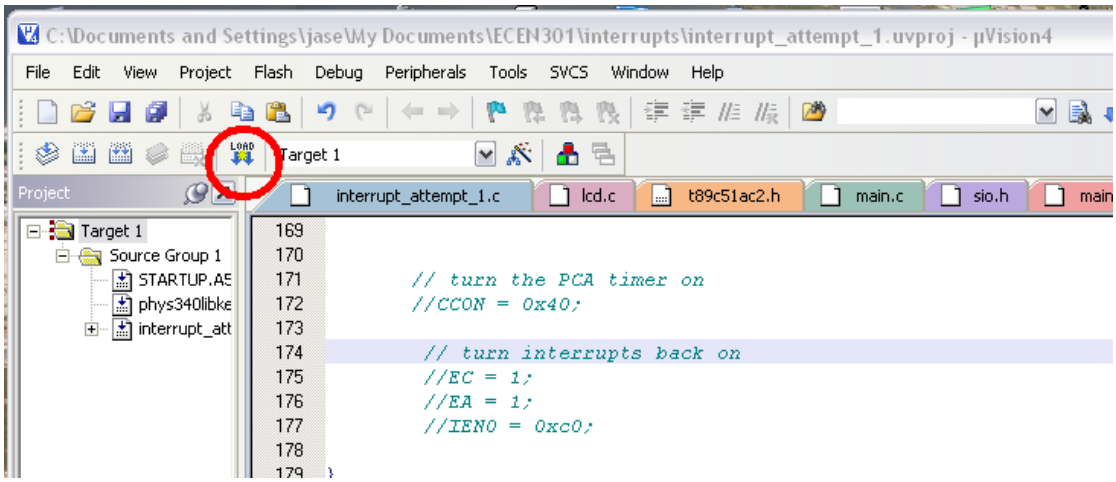
KEIL windows, with successful compilation

If you open a file explorer and look in the folder that you have created the project in, you will notice that a number of new files have been created. The end result is the .hex file, which contains the bit stream (binary file) that will be downloaded to the board.

The next step is to see your program running, (it is rather useless otherwise!). To do this, we need to download the compiled program to the microprocessor board. The steps for preparing the micro board for programming are:

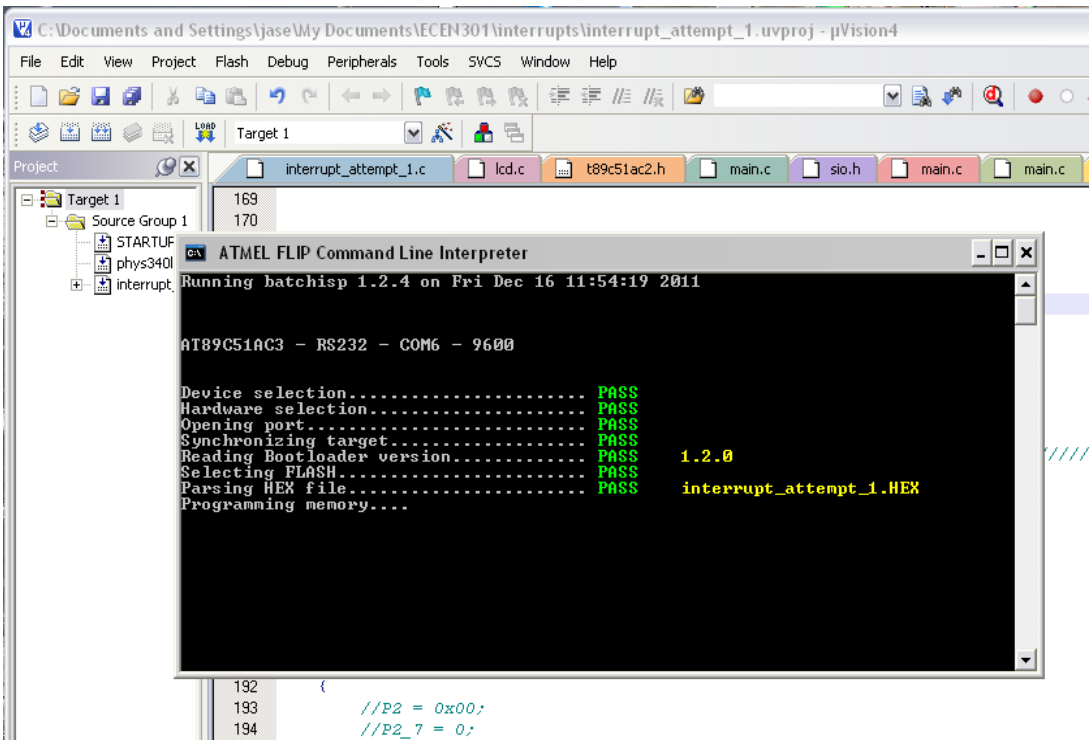
- Make sure your board has power, and the USB cable is plugged in.
- Make sure the UART is switched on.
- Press and hold the Program button.
- While holding the Program button, press and release the Reset button.
- Release the Program button.

To download the program, go to the Flash menu and select Download or alternatively hit the Load button as shown in the figure below.



Download button in the toolbar

If everything went well you should notice a small command prompt window pop up and beginning to cycle through the download process. If the synchronizing target fails, this is a sign that a timeout has occurred. Check that the UART switch on your micro board is turned on, and try repeating the button presses to put it into programming mode.



Auto-Downloader window in the process of successfully loading the program.

After a few seconds, your program should be finished downloading.

Press the RESET button to run the program. **You must press reset after each new upload to run your program.** Verify that you see the “HELLO” message.

So we have now been into an edit environment, compiled a program, and "downloaded" it onto another microprocessor, from which our program was run.

Part II: Writing the code to count and display time.

You are now going to create your own program to perform the specific task of implementing a digital clock. Delete the code within “clock1.a51” and copy the code from “Lab6 Clock.a51”. This code is a template where the timer and some additional functions have been implemented for you. Your task is to write the code that updates and displays the count.

Additional tasks:

1. How can we speed up the process of verifying that the clock is counting correctly?
2. If you place an oscilloscope probe on Port2-bit3, you will see a square wave. You will notice that the widths are not exactly 1 second. How can we correct this?
3. Try and change the code to make the clock start at a different time.