Lab 3 – EECS3216

In this lab, our goal is to design and implement a **functional calculator** capable of performing basic arithmetic operations, including **addition, subtraction, and multiplication** of integers. The calculator will be developed using the **HCS12 microcontroller** and will interact with users through a **keypad and a 7-segment LED display**. Here are the key features we want to have:

1. **User Input via Keypad**
    - The calculator should accept numerical inputs and arithmetic operations via the **keypad module**.
    - Users should be able to enter multi-digit numbers sequentially.
    - The calculator should be able to recognize and process valid key presses.
2. **Processing Arithmetic Operations**
    - The program should correctly perform **integer addition, subtraction, and multiplication** based on user input.
    - It must handle both **positive and negative integers** as well as **zero**.
    - The calculator should follow the correct sequence of operations (left-to-right execution).
3. **Displaying Results**
    - The computed result should be displayed on the **7-segment LED display** in a readable format.
    - In case of overflow (result exceeding the display limit), an error message or indicator should be shown.
4. **Handling User Interaction**
    - Implement **de-bouncing** to ensure that key presses are registered correctly without unintended multiple inputs.
    - Provide a mechanism to clear input (e.g., a "C" key to reset the current operation).
    - Ensure that invalid inputs (such as pressing an operator first) are handled gracefully.
5. **Execution on the Dragon Board**
    - The program should be developed using **Eclipse for HCS12** and loaded onto the **Dragon Board**.
    - It should be tested using **TeraTerm** for serial communication and debugging.
    - The program will be executed using the **D-Bug12 monitor** and controlled via serial commands.

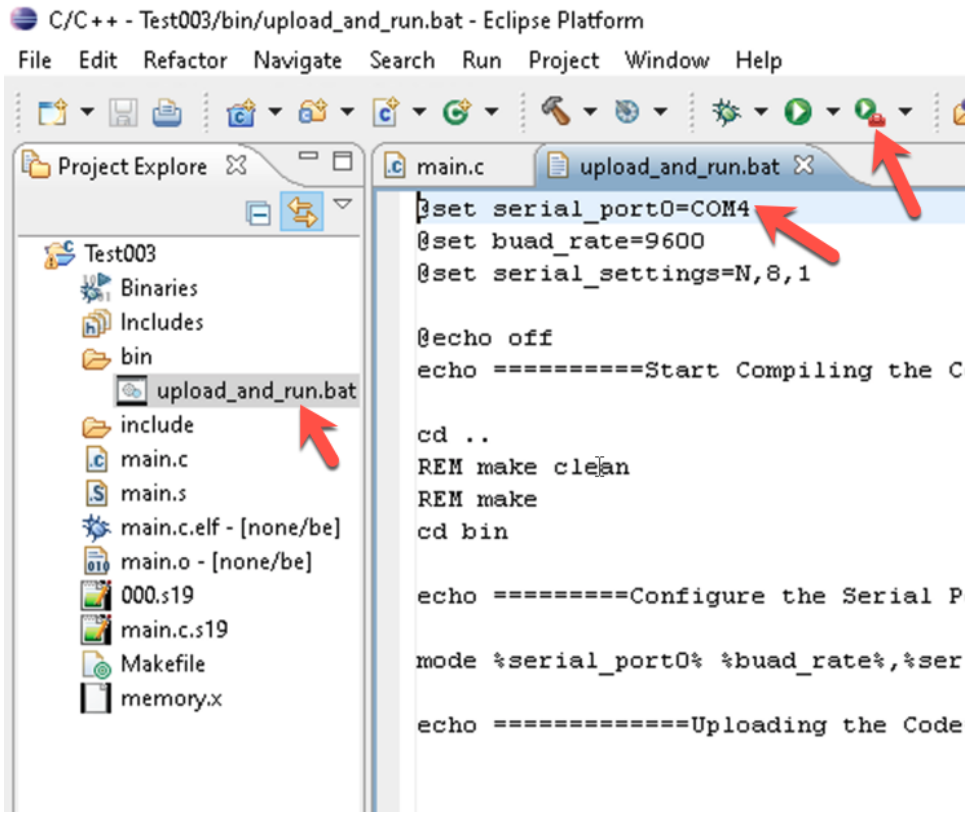
Important note: You can also use alternative MCUs if you have them available, such as Arduino boards, Raspberry Pi Pico, ST Microcontrollers (including Nucleo boards), or TI TIVA boards. However, it is essential to thoroughly understand and explain all the underlying libraries and internal mechanisms. Simply relying on API calls is insufficient for this lab; a comprehensive grasp of the low-level details is required.

Pre-lab

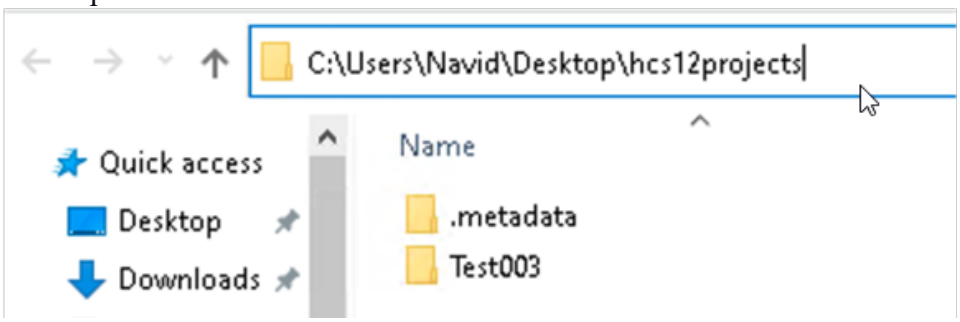
It is essential to review the necessary foundational concepts and resources. Below are the key areas you should focus on:

1. **Review EECS3216 Lectures**

    - Refresh your understanding of **HCS12 architecture**, **C language**, and **peripheral programming**.
    - Pay close attention to topics such as **I/O port control, DDR, and memory mappings**.
2. **Access HCS12 Documentation & User Manuals from e-class.**
    - All essential reference materials, including **technical documentation** and **user manuals** for the **HCS12 MCU** and **Dragon12 board**, can be found in the **e-Class HCS12 folder**.
    - These resources will provide in-depth information about **registers, instruction sets, and hardware configurations** required for development.
3. **Prerequisite Knowledge from EECS2031**
    - Ensure you have a strong grasp of **C programming**, as it is the primary language used for HCS12 development.
    - Review concepts such as **pointers, memory management, bitwise operations, and low-level programming techniques**.
4. **C Programming Considerations for HCS12**
    - Unlike standard desktop environments, **HCS12 does not support the standard C library (libc)**.
    - This means that **fundamental functions** such as atoi(), strlen(), and printf() are **not available by default**, and you may need to **implement them manually**.
    - Focus on writing **efficient, memory-conscious code** tailored for a tiny **embedded system environment**.
5. Watch the demo video on how to use GCC-HCS12 Eclipse IDE: https://www.eecs.yorku.ca/course_archive/2024-25/W/3216/demo/demo-eclipse-hcs12.mp4
6. As we covered in class:
    - The Keypad module of the Dragon board uses HCS12 port A, and the 7-segment LEDs display utilizes port B.
    - Make sure you know how to de-bounce button/key push events. Make sure you can use Eclipse HCS12 Hello World program demonstrated in the class.
    - Remember that you can use TeraTerm serial terminal to connect to Dragon boards running D-Bug12.
        - 9600 baud, eight data bits, one stop bit, XOn/XOff handshaking, and no parity.
        - Issue command "load" and transfer your S-Record file (000.s19). Issue command "g 2000" to run your code. If you configure the IDE with the correct serial port, you can automate this as demonstrated in the class.
        - Of course, you need to adjust your USB to Serial COM Port (look under Windows Device Manager to see what ports you have):



- If you need the installer for the Eclipse-based HCS12 IDE, it is located in e-class material under name gcc-hcs12-ide.exe. It is basically a script that benefit from Eclipse IDE, and it creates a workspace on Desktop under a directory called hcs12projects.
- If you need the workspace and sample stationary files only, it is located under course material under name hcs12projects.zip which needs to be placed on Desktop:



** Read the lab instructions and **create a flowchart of the program you want to write (TAs will collect this at the beginning of the lab)**. Make sure you use modular design and use functions.

Objective

Let's start with a very elementary calculator to calculate one-digit multiplication and additions using the keypad and 7-segment LEDs. Assume Keypad 'A' key is used for addition, 'B' key for multiplication, and 'D' as equal/enter (you can use 'C' for clear/reset). An example would be 2+3=5:

- Press 2 on the keypad, and 7-segs will show "2",
- Press A on the keypad,
- Press 3 on the keypad, and 7-segs will show "3",
- Finally, press D on the keypad, and 7-segs will show "5".

Once you can do this, make sure you can handle 3x4=12:

- Press 3 on the keypad, and 7-segs will show "3",
- Press B on the keypad,
- Press 4 on the keypad, and 7-segs will show "4",
- Finally, press D on the keypad, and 7-segs will show "12".
    - Remember that we covered how you can show multiple digits on 7-segs displays in the class.

Once this is done, proceed to incorporate the LCD as well. This means the results will be shown on LCD as well.

Lab Demonstration & Report Submission

1. **Lab Demonstration**
    - Present your **working implementation** to your **TA** during the designated lab session.
    - Ensure that your program functions correctly, meets all specified requirements, and handles both expected and edge cases properly.
    - Be prepared to **explain your approach**, including the logic behind your code, how you interfaced with the hardware, and any challenges you encountered.
2. **Lab Report Requirements**
    - Provide a brief overview of the lab objectives and the purpose of your implementation.
    - **Implementation Details**:
        - Explain the **step-by-step development process**, including key design decisions.
        - Describe how you handled **input processing, computation, and output display**, debouncing, 7-segs, and LCD at minimum.
        - Include **diagrams, flowcharts, or pseudocode** to enhance clarity.
    - **Code Explanation**:
        - Document and explain all major sections of your **source code**.
        - Highlight **important functions, algorithms, and optimizations** used in your implementation.
    - **Hardware & Software Configuration**:
        - Specify the **tools, development environment, and configurations** used (e.g., **Eclipse HCS12, Dragon12 board, TeraTerm**).

- **Testing & Debugging**:
  - Describe how you tested your implementation, including **test cases and expected outputs**.
  - Mention any **bugs encountered and how they were resolved**.
- **Conclusion & Reflection**:
  - Summarize the key learnings from the lab.
  - Discuss potential improvements or additional features that could be implemented in future iterations.
- **Report Submission (at the beginning of the next lab)**
  - Ensure that your report is **well-structured, clear, and properly formatted**.
  - Include **screenshots or terminal outputs** to support your explanations.
  - Submit the final version according to the **lab submission guidelines**.