Preparation

If you do not have a teammate:

Please join the “matchmaking” with GAs to find a teammate in the first room;

If you have a teammate:

Register your group on Canvas;

Register the board from Mr Ho in the first room with your teammate
(1 board per person);

Download STM32IDE Version 1.10.1;

PCI-1: Familiarization with Development Environment

CG2028 COMPUTER ORGANIZATION

Hou Linxin (TA), Week 8-9

Contact Details

- **Course matters**
    - Teaching Assistant, Hou Linxin
    - Email: hou.lx@nus.edu.sg
    - (Course-related questions)
- **Lab matters**
    - Lab Officer Mr. Ho Fook Mun
    - Email: elehofm@nus.edu.sg
    - (Admin-related questions)

Office Hour: 9am-6pm on weekdays

Your helpful friends - the 3 GAs

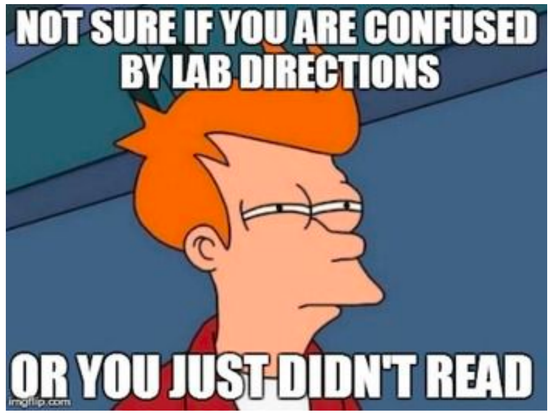
Pande Shreya
e0426333@u.nus.edu

Yuan Xun
e0919068@u.nus.edu

Thaw Tint Te Tun
e1144046@u.nus.edu

Lab Rules

- Arrive on time and bring your hardware to the lab
- No food or beverage in the lab
- **Dress code: covered shoes**
- Switch off appliances when not in use, take good care of your board.
- Use electrical equipment and apparatus in accordance with the manufacturer’s operating instructions. Read data sheet!



Lab Schedule

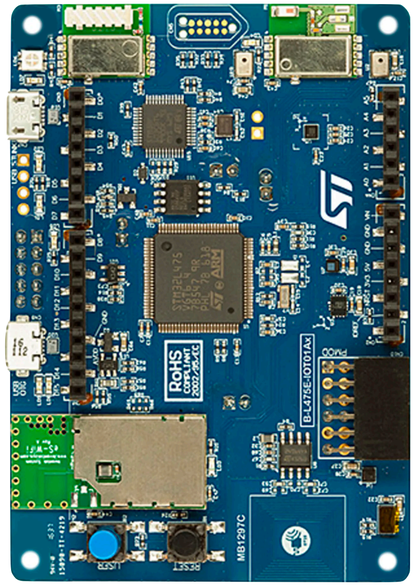
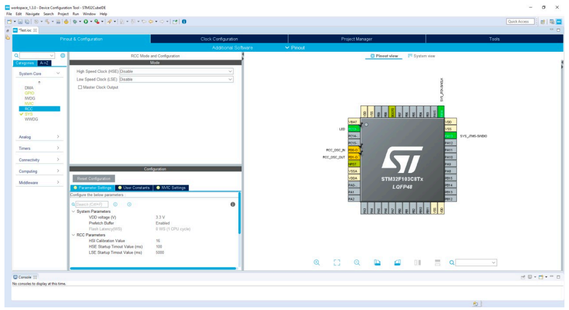
WEEK	CG2028 Laboratory
3	PCI-1: Familiarization with Development Environment
4	PCI-2: Assembly Language and C Programming
5	Lab Break :)
6	Assignment Submission

- Starts your assignment early.
- No late submission is considered.
- Try not do last-minute submission to prevent traffic congestion.

Objectives of Today

- Installation of STM32CubeIDE on PC;
- Familiarising with STM32CubeIDE, which is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors;
- Build, download and execute:
  - a simple assembly language program, and
  - a C program, on the ARM Cortex-M4 system on chip on the STM32L475VG board;
- Understand basic ARM assembly instructions and familiar with their usage.

Resources

Hardware	Software	Instruction
		CANVAS Pages Lab manuals will be uploaded to Canvas before labs. Slides will be uploaded to Canvas after labs.
STM32L475VG	STM32CubeIDE 1.10.1	

Installation of STM32CubeIDE

- STM32CubeIDE Installation: https://www.st.com/en/development-tools/stm32cubeide.html
- ST-LINK server software module: https://www.st.com/en/development-tools/st-link-server.html#get-software (for Mac user)
- For MAC user: verify the downloaded installer under “Privacy & Security”
- For Windows user: disable your antivirus software/firewall during downloading and installation
- Use the lab desktop if you did not bring you laptop/have but do testing on your own PC after lab

STM32CubeIDE Familiarisation

- To Familiarize the IDE & Test Setup
  - Decide the Workspace
  - Load the Projects
  - Build and Debug the Blink Projects
  - Test Hardware and IDE Setup
  - Pause and Terminate
  - You are NOT required to modify the codes
- Example Projects:
  - **blink_pure_asm.s**
  - **blink_plain_c.c**
- Expected observation: green LED blinking on test board

Understand STM32CubeIDE

- Familiarise with STM32CubeIDE debugging process
  - Breakpoint
  - Step-into & step-over
  - Registers and memory inspection
- Example Projects:
  - **asm_basic**
- Purpose of the code: Calculate ANSWER = A*B + C*D
- Format for instruction set can be refer to Chapter A7 of ARMv7-M Architecure Ref Manual

Discussion

- What is the difference between **LDR** and **MOV**?
  - LDR is for definition: memory to processor
  - MOV is for relocation: processor to processor
- What is the difference between **LDR** and **STR**?
  - LDR loads a register with a value from memory: memory to processor
  - STR stores a register value into memory: processor to memory

Discussion

- What is the difference between **.equ** and **.word** when declaring constants?
  - **.equ** is like **#define** in C: no variable created
  - **.word** is like **unsigned int** in C: create a constant variable
- What are the memory addresses of constant **A** and **B** respectively? And what are the memory addresses of constant **C** and **D?**
  - **=A, =B; C and D does not have addresses**
- Get memory address from register list
- Key in the memory address to get date stored at the memory address

Summary (asm_basic)

	Label	Address	Content
.word	A	=A	A
.word	B	=B	B
.equ	C	-	=C
.equ	D	-	=D
.lcomm	ANS	=ANS	-

Homework

- Modify on the **main.s** of **asm_basic,** to store all the odd numbers between 50 and 100 into the memory, starting from the address right after where **ANSWER** is.

- **Hint 1:** If the address of constant **ANSWER** is **ADDR**, the address of next memory location right **ANSWER** is **ADDR+4**.

- **Hint 2:** A loop needs to be created to recurring store values into the memory.