STM32MP15 stm32cubemx DDR suite hands-on training

Bossen WU

Hands-on objectives

- At Hands-on completion, trainee will be able to
  - Create a DDR configuration based on DDR datasheet
  - Generate Device Tree with DDR content and build boot files for basic mode
  - Build U-Boot SPL file tied with Device Tree
  - Connect to target using boot files generated and ‘DDR interactive’ protocol
  - Perform DDR tuning and propagate back the tuning parameters to DDR configuration
  - Execute DDR tests

Hands-on pre-requisites

- Equipment
  - STM32MP157C-DK2
  - micro USB cable + USB Type C cable
  - Linux computer
- Tools
  - STM32CubeMX / STM32CubeIDE
  - STM32CubeProgrammer
- SW image
  - Starter Packages
- Documentation
  - DDR3L Micron datasheet: “MT41K256M16TW-107-P-V00H_4Gb_DDR3L.pdf”

Hands-on sequence details

- Step1: Create the DDR configuration of STM32MP157C-DK2 using STM32CCubeMX and the Micron DDR3 datasheet
- Step2: Check the DDR node generated in Device Tree
- Step3: Build U-Boot SPL file tied with Device tree
- Step4: Connect the STM32CubeMX to the target
  - DDR interactive with U-Boot SPL (basic boot) loaded in SYSRAM
  - DDR interactive with U-Boot SPL (basic boot) as a SD-Card image
- Step5: Perform DDR tuning and propagate back the tuning parameters to configuration made in Step1
- Step6: Perform DDR tests on target programmed with configuration made in Step1

Let’s start DDR suite hands-on !

life.augmented

Step1 - create the DDR configuration

45 min

Step1 – create MB1263 DDR configuration

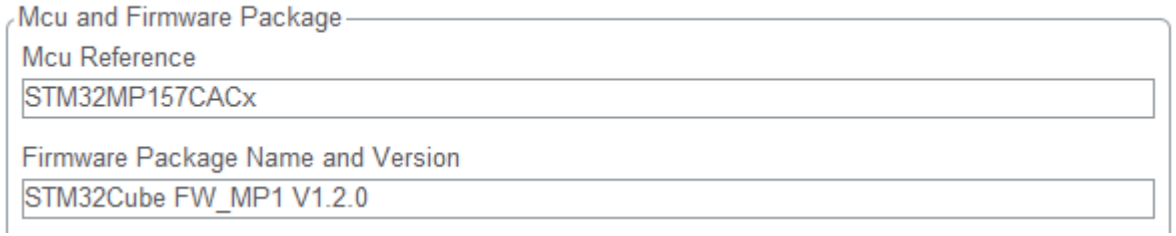
- On STM32CubeMX, start a new STM32MP15 project for STM32MP157AAAx chip
- In Pinout tab, for DDR IP, indicate
  - DDR Type = DDR3L
  - Width = 16bits
  - Density = 4Gits
- In Clock Configuration tab, indicate 533MHz for the DDRC clock
- In Configuration tab, for the DDR IP
  - Enter following geometry parameters: Row-Bank-Column address mapping / No Relaxed Timings
  - Complete the configuration with JEDEC timings you will find in the Micron datasheet:
    - Make sure to select parameters from the speed bin “DDR3(L)–1066 / 8-8-8”. This is the column name
    - Note: use backup slides in case you want to save time looking for timings in datasheet (not recommended for first time users)

Step2 - check the configuration

20 min

Step2 – check the configuration from device tree file (1)

- On STM32CubeMX, go to ‘Project Manager’
  - Enter a project name (and a project location if required)
  - You should have Mcu and firmware package field as below:



  - Choose OK
  - The Device tree will be generated

20 min

Step2 – check the configuration from device tree file (2)

- Choose to open the target folder once device tree generation is over
  - Edit stm32mp15<name-Of-Project>-mx.dts file with a text editor
    - You should have following node indicating 4Gb configuration (0x20000000)

```
memory@c0000000 {
        reg = <0xc0000000 0x20000000>;

        /* USER CODE BEGIN memory */
        /* USER CODE END memory */
};
```

  - Edit stm32mp15-mx.dtsi file with a text editor and compare it to arch/arm/dts/stm32mp15-ddr3-1x4Gb-1066-binG.dtsi in U-boot
    Except the header, rest of the file should be the same

life.augmented

Step3 – generate u-boot SPL

30 min

Step3 – build u-boot SPL file tied with device tree (1)

- Follow https://wiki.st.com/stm32mpu/wiki/How_to_create_your_own_machine
- To be noticed that the boot scheme has to be set to ‘basic’ for U-Boot SPL, see §3: ‘3.2 Edit the new machine config file’: stm32mp1-<ProjectName>-config.inc
- Generate the device tree, see §2: ‘Generate devicetree’
- Launch the Yocto build for U-Boot, see §5:Compile your image with the yocto build process.
- U-Boot SPL binary is available in <buildfolder>/images/stm32mp1-<ProjectName> Filename is u-boot-spl-<ProjectName>-basic.stm32 for STM32MP157C-DK2
- Copy u-boot-spl-<ProjectName>-basic.stm32 file in a safe place, it will be used for Step4

30 min

Step3 – build u-boot SPL file tied with device tree (2)

- Copy of U-Boot SPL file to SD-Card:
  - Make sure you have GPT partitions created on your SD-Card
  - And follow:https://wiki.st.com/stm32mpu/wiki/How_to_populate_the_SD_card_with_dd_command
  - (Or https://wiki.st.com/stm32mpu/wiki/How_to_update_U-Boot_on_an_SD_card)
- Keep the SD-Card in a safe place, it will be used for Step4

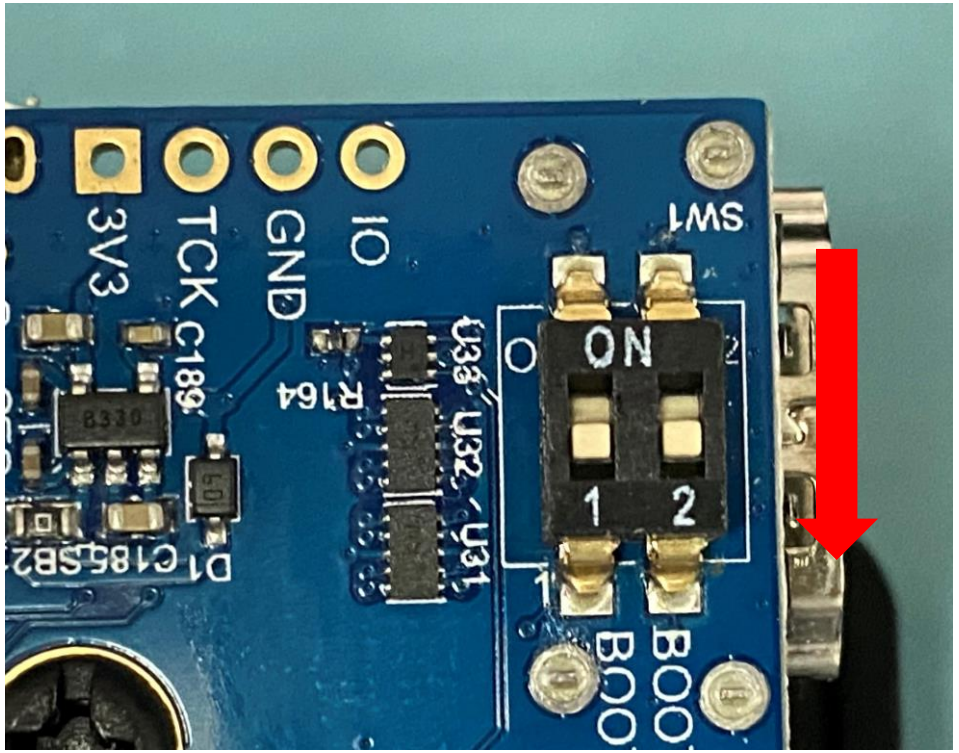
life.augmented

Step4 – connect to the board

30 min

Step4 – connection to the target

- Using u-boot-spl-<ProjectName>-basic.stm32 file (STM32CubeProgrammer)

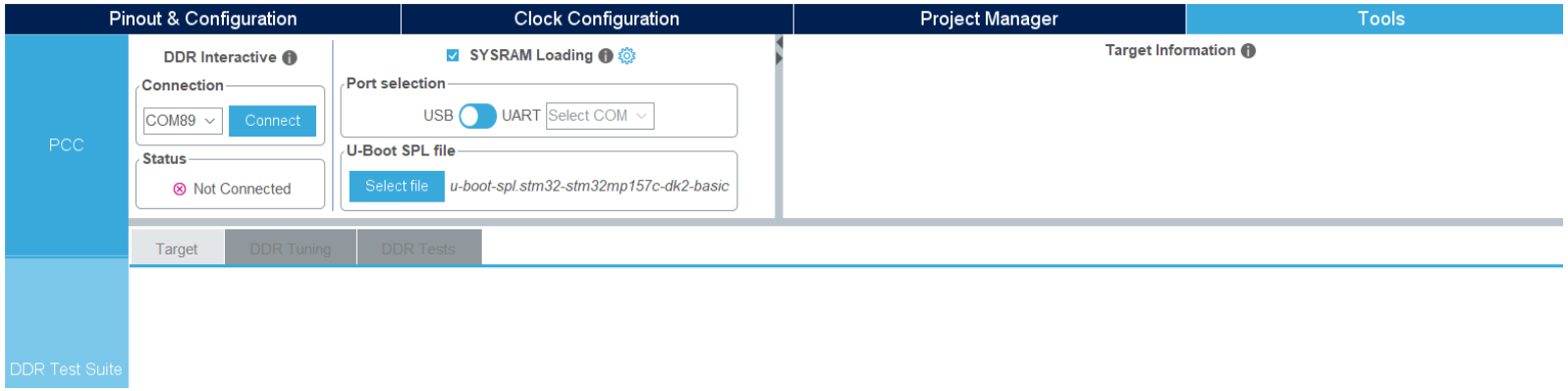


life.augmented

30 min

Step4 – connection to the target

- In STM32CubeMX Configure the DDR Test Suite connection banner according to connection setup chosen in previous slides



- The STM32CubeProgrammer for MPU should be installed. If STM32CubeMX does not detect it automatically, browse to the install folder in Connection settings menu (⚙)
- Reset the board after pressing CONNECT button

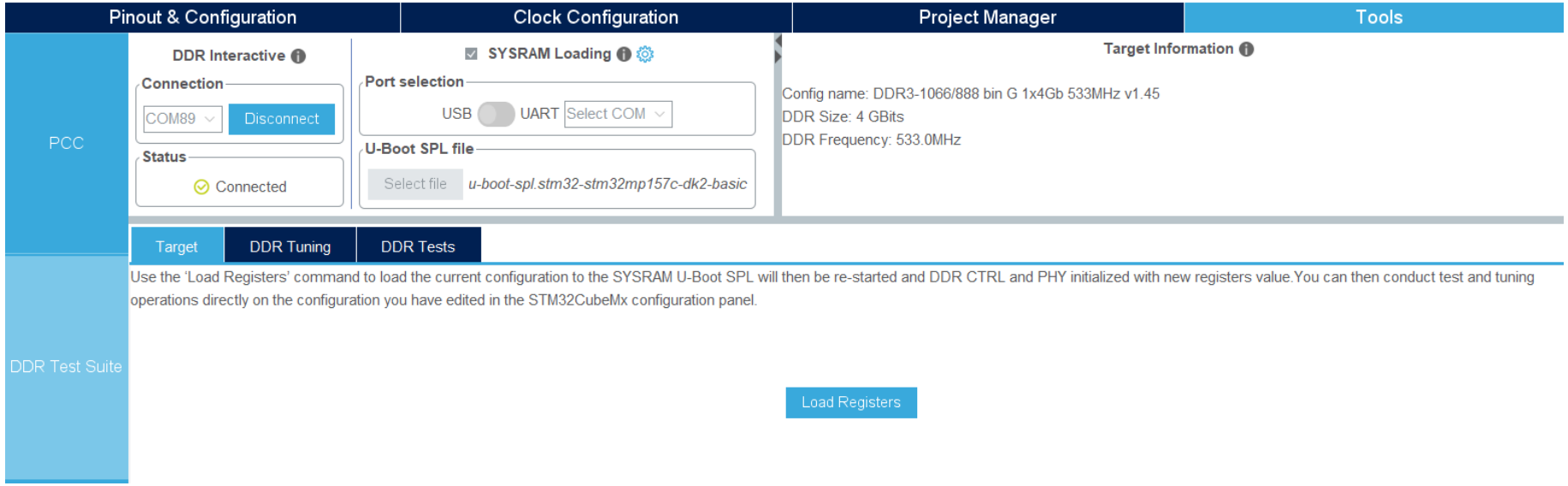
life.augmented

Step5 – perform DDR tuning

20 min

Step5 – perform tuning on the target (1)

- With Step3 and Step4, you have a DDR configuration via U-Boot SPL file that is used to boot the STM32MP1 board, and the board is connected. You can skip the load registers step
- If you skip Step3 and use a ‘vanilla’ U-Boot SPL file, you can still load your DDR configuration (from Step1) to memory via the ‘Load registers’ feature available in Target tab

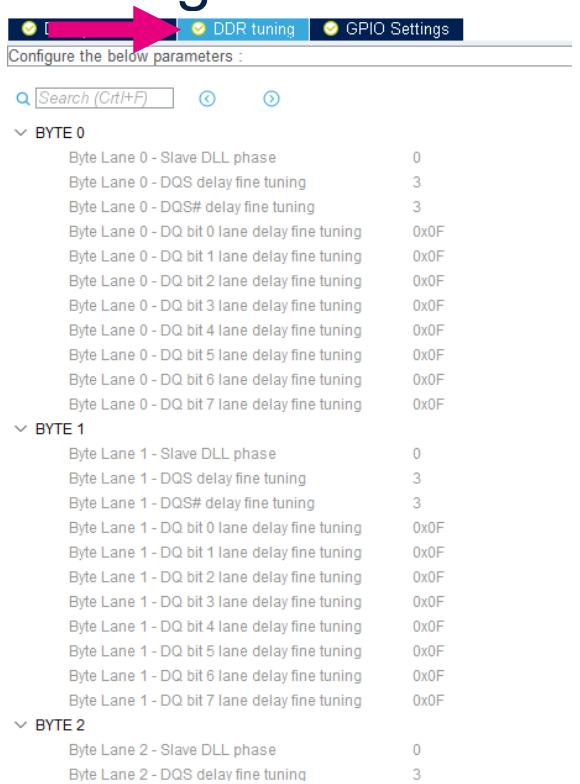


life.augmented

20 min

Step5 – perform tuning on the target (1)

- Before your start Tuning operation, have a look to the ‘DDR tuning’ tab in the DDR IP configuration panel
- Execute then the tuning operation
- The 3 steps should run without errors
- Choose then to save the tuning parameters to the configuration
- Check now the ‘DDR tuning’ tab in the DDR IP configuration panel. Note the differences
- You can now save the STM32CubeMX project to save tuning parameters and redo step2 to ensure the device tree is generated with Tuning parameters



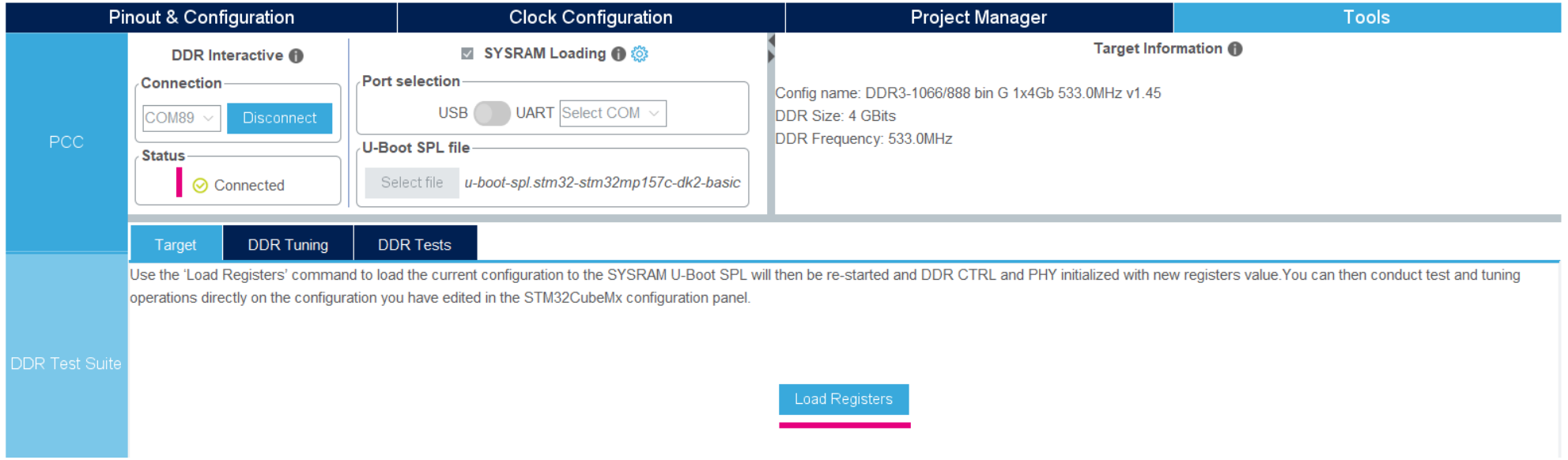
life.augmented

Step6 – perform DDR tests

20 min

Step6 – perform test on the target (1)

- With Step3 and Step4, you have a DDR configuration via U-Boot SPL file that is used to boot the STM32MP1 board, and the board is connected. You can skip the load registers step.
- If you skip Step3 and use a ‘vanilla’ U-Boot SPL file, you can still load your DDR configuration (from Step1) to memory via the ‘Load registers’ feature available in Target tab



20 min

Step6 – perform test on the target (2)

- Execute tests on the defined configuration.
- All test should be passed
- You can modify the DDR configuration with funky parameters, load registers (or generate new Device Tree and new U-Boot SPL) and check test are now failing.
- Some consideration on tests:
  - Base address 0xC0000000, impossible to trig test with test start address below this one
  - Tests with Size parameters required either Size to be 2^n, either to be a multiple of 4
  - Test start address + Test size should be below the maximum DDR address (i.e 0xC0000000 + 0x20000000 for 4Gb DDR size)

Backup

life.augmented

Parameter collection from DRAM datasheet

- Reference used on STM32MP157C-DK2 board
  - Micron DDR3L – 4Gb
  - MT41K256M16TW-107-P-V00H
  - Speed bin = 1066/8-8-8

Table 9: Timing Parameters Used for I_{DD} Meas

I_{DD} Parameter		DDR3L -800		DDR3L -1066		
		-25E	-25	-187E	-187	
		5-5-5	6-6-6	7-7-7	8-8-8	
^{t}CK (MIN) I_{DD}		2.5		1.875		
CL I_{DD}		5	6	7	8	
^{t}RCD (MIN) I_{DD}		5	6	7	8	
^{t}RC (MIN) I_{DD}		20	21	27	28	
^{t}RAS (MIN) I_{DD}		15	15	20	20	
^{t}RP (MIN)		5	6	7	8	
^{t}FAW	x4, x8	16	16	20	20	
	x16	20	20	27	27	
^{t}RRD I_{DD}	x4, x8	4	4	4	4	
	x16	4	4	6	6	
^{t}RFC	1Gb	44	44	59	59	
	2Gb	64	64	86	86	
	4Gb	104	104	139	139	
	8Gb	140	140	187	187	

Parameter collection from DRAM datasheet

- DDR3/DDR3L parameters collected from Micron datasheet (1066/8-8-8):
  - Write recovery time (tWR) – 15ns
  - Refresh Interval Time (tREFI) – 7,8us
  - Refresh Cycle Time (tRFC) – 260ns
  - Row Address to column Address Delay (tRCD) – 15ns
  - Row Precharge Time (tRP) – 15ns
  - Row Cycle Time (tRC) – 52,5ns
  - Row Active Time (tRAS) – 37,5ns
  - Max between ACT and PRE to same bank (tRASmax) – 9*tREFI
  - READ-to-PRECHARGE time (tRTP) – 4 clocks (7,57ns)
  - Write recovery time (tWTR) – 4 clocks (7,57ns)
  - MODE REGISTER SET command cycle time (tMRD) – 4 clocks
  - MODE REGISTER SET command update delay (tMOD) - 12 clocks (22,72ns)
  - CAS#-to-CAS# command delay (tCCD) - 4 clocks
  - ACTIVATE-to-ACTIVATE minimum command period (tRRD) - 7,57ns
  - Four ACTIVATE windows (tFAW) – 50ns
  - Exit self refresh to commands requiring a locked DLL (tXSDLL) – 512 clocks
  - Min clocks after self refresh entry (tCKSRE) – 10ns
  - Min clocks before self refresh exit (tCKSRX) – 10ns
  - Power-Down Exit (tXP) – 6 ns
  - Min number of CKE high/low during Self Refresh and Power D (tCKE) – 3 clocks (5,68ns)
  - Power-Down Exit (tXPDLL) – 24ns

Page size is 2KB

Tck is 1.875 ns