



life.augmented



DDRFW-UTIL for DDR sanity check

Agenda

- 1 DDRFW-UTIL overview
- 2 Hands-on build DDRFW-UTILS & run DDR tests
- 3 Adapt DDRFW-UTIL to your board

DDRFW-UTIL Overview

DDRFW-UTIL overview

- **DDRFW-UTIL**, a firmware for DDR initialization and DDR tests
- **When these tests should be run ?**
 - At bring-up phase for **board sanity-check** of DDR bus for confidence prior to production
 - Default test parameters meant to catch low-margin timings of the PCB over the STM32MP1 temperature range
 - DDR errors, to pin-point the root cause

but

Does not happen when PCB designers respects “**AN5692** STM32MP13x lines DDR memory routing guidelines”

Tests

- **Basic tests:** These simple and running fast tests are intended to capture the major configuration or hardware issues showing off immediately.
- **Intensive tests:** These tests use extensive coverage of data and address patterns for noise and high SSO conditions, high throughput traffic or interleaved read/write.
- **Stress tests:** These tests are intensive and executed with stretched conditions

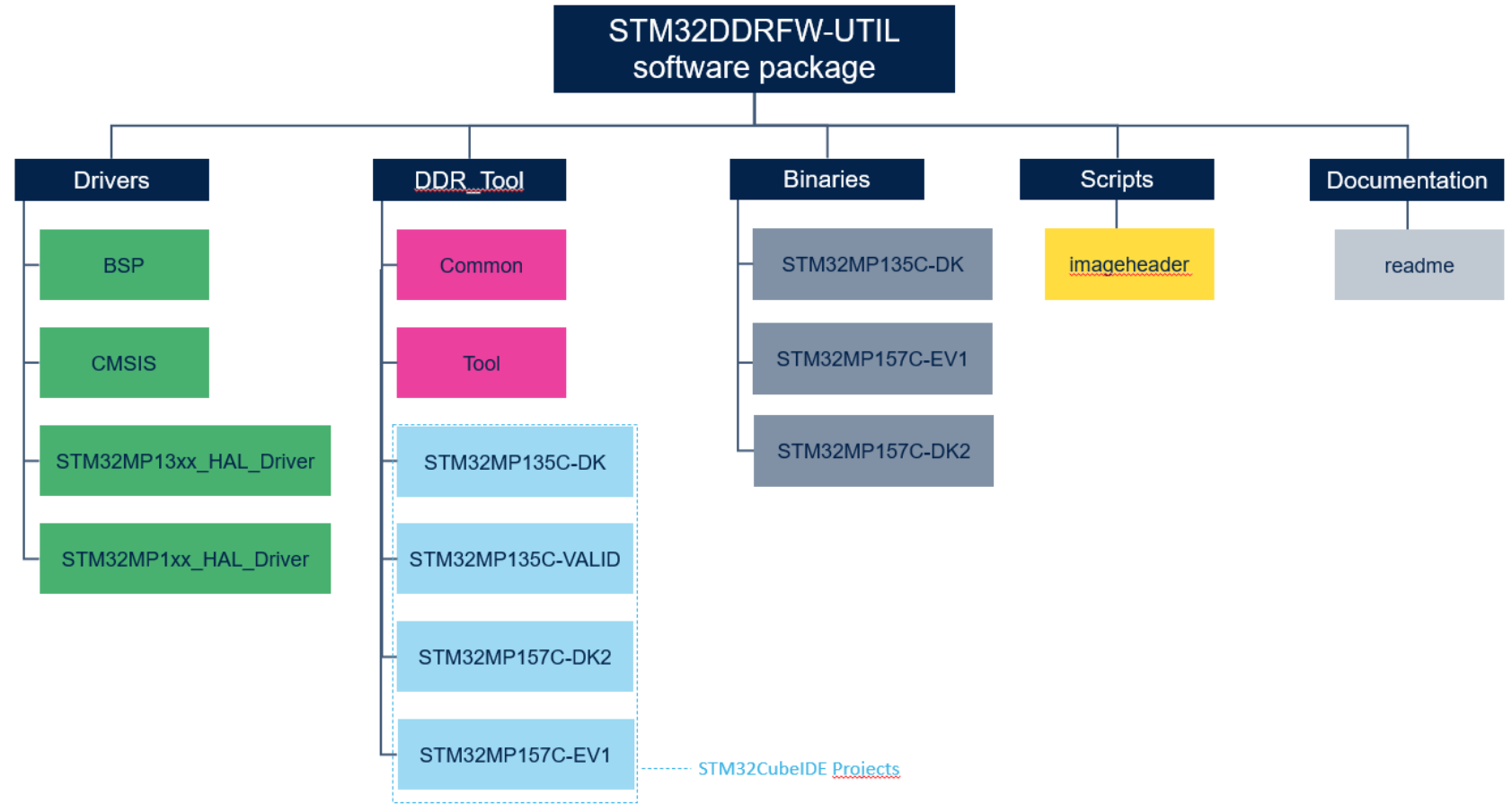
DDRFW-UTIL overview

TEST	TEST NAME	TYPE	DESCRIPTION
1	Simple Databus	Basic	Verifies each data bus signal can be driven high at the given address.
2	Databus Walking 0	Basic	Verifies each data bus signal can be driven low.
3	Databus Walking 1	Basic	Verifies each data bus signal can be driven high.
4	Address Bus	Basic	Verifies each address bus line in a memory region by performing a walking 1 test on the relevant address bits and checking for aliasing.
5	Mem Device	Intensive	Performs read/write over an entire memory region. Each data bit is written and read back with 0 and 1 values.
6	Simultaneous Switching Output	Intensive	Stresses the data bus over an address range by doing simultaneous switching output. Writes a pseudo-random value and reads it back.
7	Noise	Intensive	Verifies read/write while forcing switching of all data bus lines.
8	Noise Burst	Intensive	Verifies read/write while forcing switching of all data bus lines (test based on 8-word bursts).
9	Random	Intensive	Verifies read/write with a pseudo-random value on one region.
10	Frequency Selective Pattern	Intensive with stress	Stresses data bus by performing successive write 8-word burst operations using mostly zero/one patterns and frequency divider patterns (F/1, F/2, F/4) for 16 and 32 data bus width.
11	Block sequential	Intensive	Well known user-space memory tester adapted.
12	Checkerboard		
13	Bit spread		
14	Bit flip		
15	Walking ones		
16	Walking zeroes		
17	Infinite read	Basic	Performs an infinite read for a specific pattern (for debug and lab usage only, not visible).
18	Infinite write	Basic	Performs an infinite write access to DDR (for debug and lab usage only, not visible).

DDRFW-UTIL overview

Project architecture

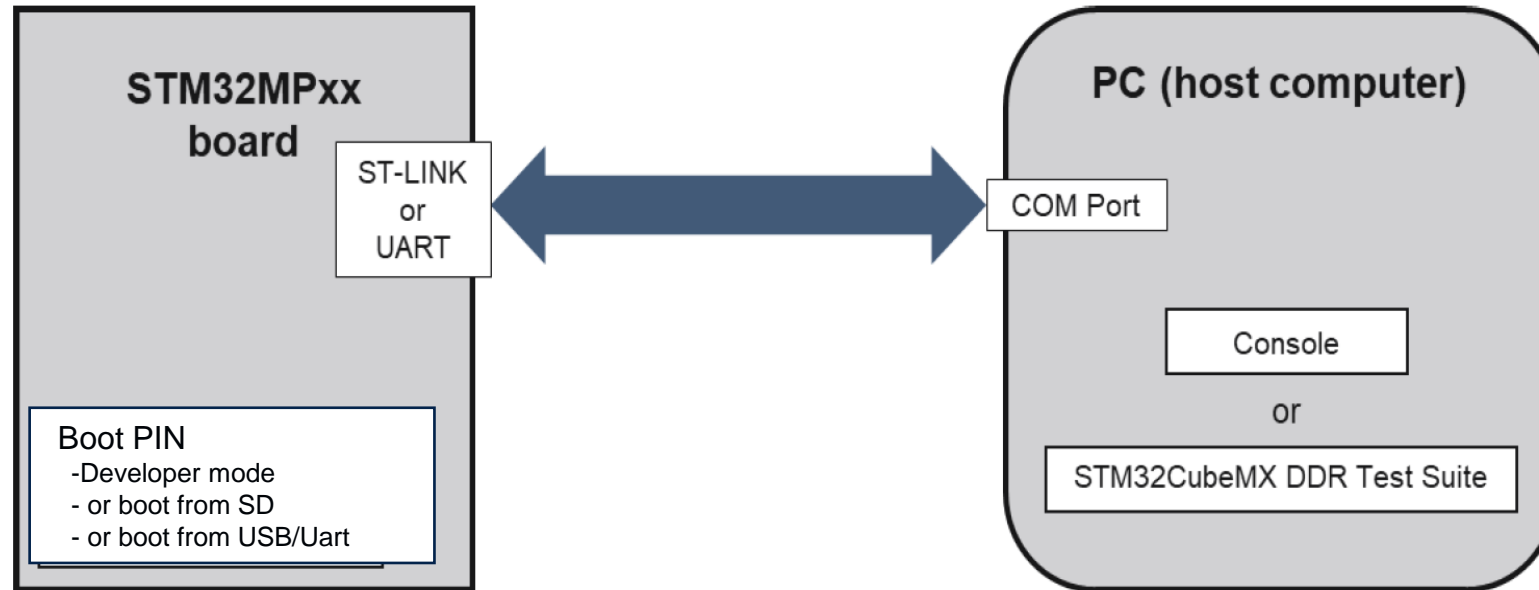
- Firmware delivered as a STM32CubeIDE project, can be generated under Windows OS.
- Source code can be adapted to PCB
- Use same drivers as Azure-RTOS STM32MP13 bare metal



DDRFW-UTIL overview

HW connection for test control in DDRFW-UTIL console

- UART communication is handled either by the ST-LINK VCP, or by a dedicated port.
- By default, UART4 with Bps=115200 Bit=8 Parity=None Stop=1, UART can be changed in DDRFW-UTIL sources



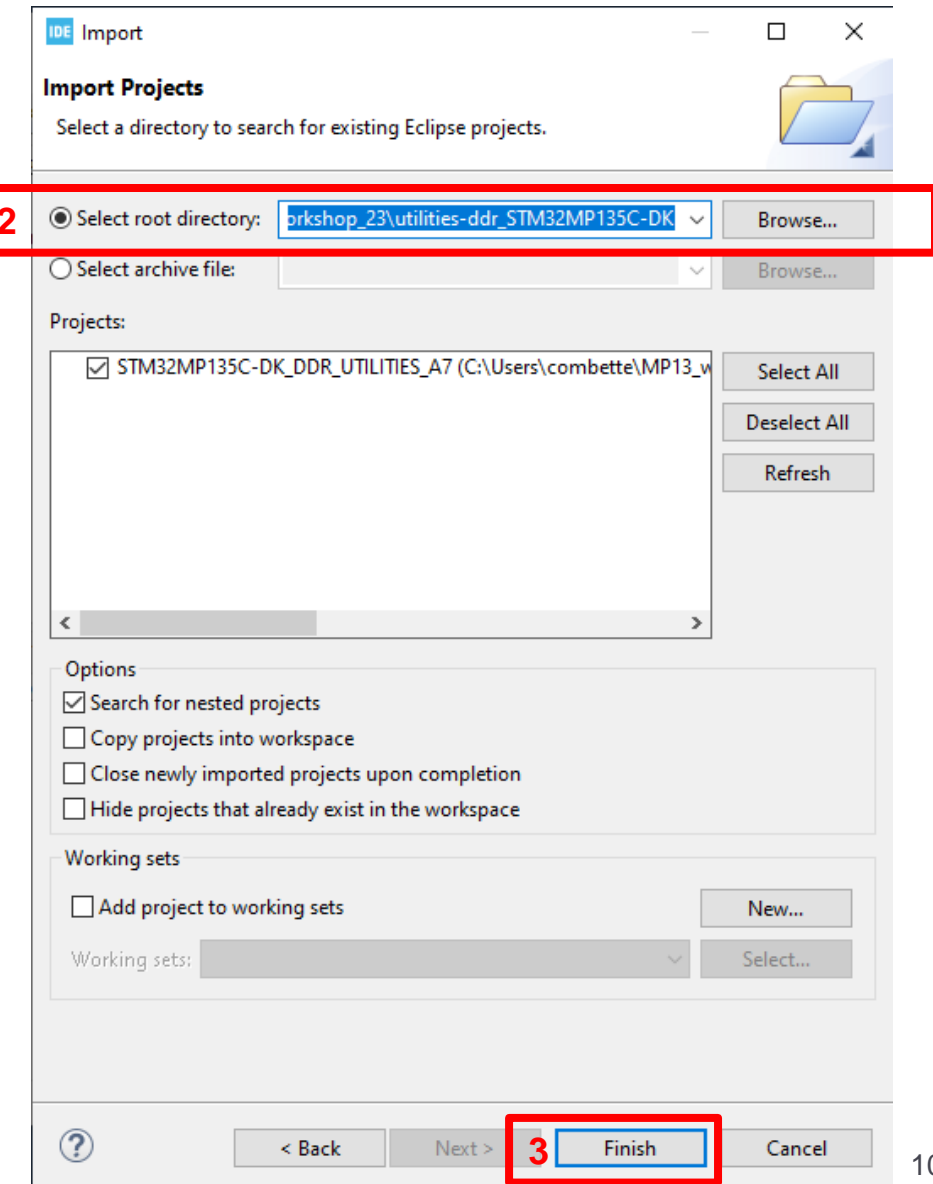
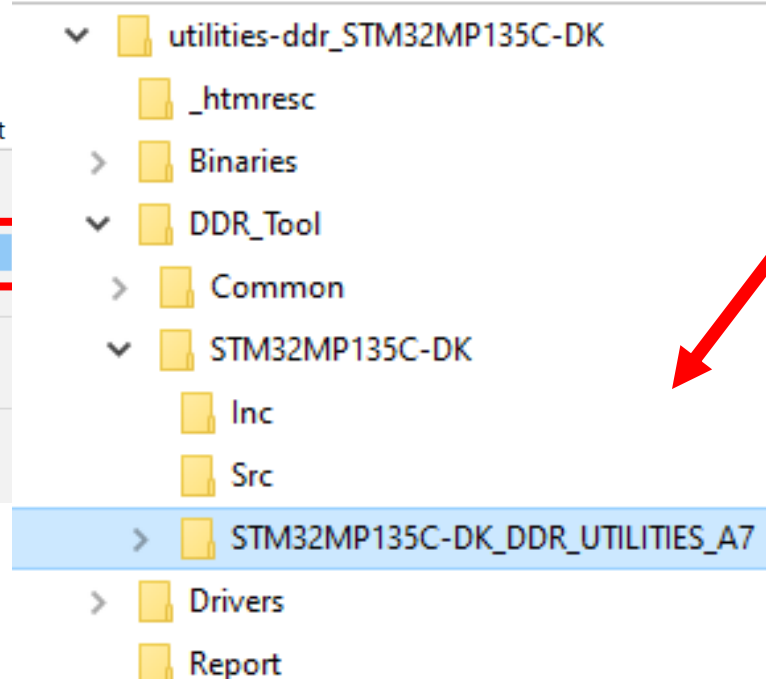
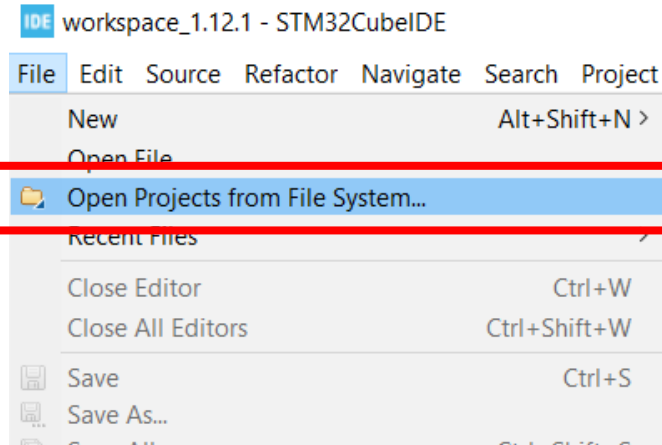
Hands-on Build DDRFW-UTIL and run DDR tests

Objectives

- Run the DDR tests on STM32MP135F-DK board

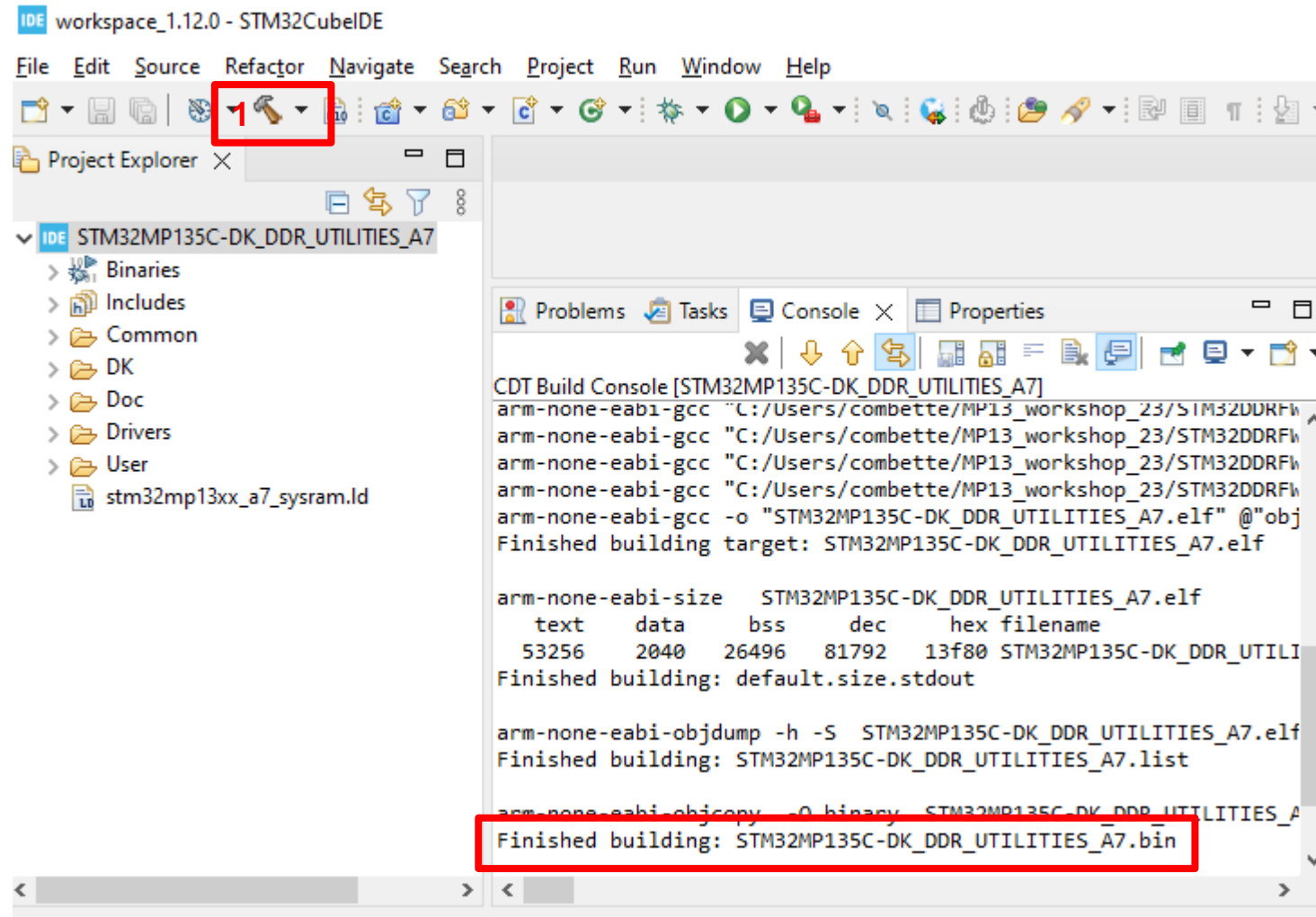
- Extract C:\ST_Workshop\STM32DDRFW-UTIL_for_STM32MP135C-DK.zip in C:\ST_Workshop\
 - Open STM32MP135C-DK_DDR_UTILITIES STM32CubeIDE project

Build the firmware



Build the firmware

Note: at that stage, DDRFW-UTIL source code must be adapted to customer PCB for settings for UART, STPMIC I2C, PLL frequency of DDR controller



workspace_1.12.0 - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- IDE STM32MP135C-DK_DDR_UTILITIES_A7
 - Binaries
 - Includes
 - Common
 - DK
 - Doc
 - Drivers
 - User
 - stm32mp13xx_a7_sysram.ld

Problems Tasks Console Properties

CDT Build Console [STM32MP135C-DK_DDR_UTILITIES_A7]

```
arm-none-eabi-gcc "C:/Users/combette/MP13_workshop_23/STM32DDRFW-UTIL/Source/Drivers/STM32MP135C-DK/STM32MP135C-DK_DDR_UTILITIES_A7.c" -c -o "STM32MP135C-DK_DDR_UTILITIES_A7.o"
arm-none-eabi-gcc "C:/Users/combette/MP13_workshop_23/STM32DDRFW-UTIL/Source/Drivers/STM32MP135C-DK/STM32MP135C-DK_DDR_UTILITIES_A7.o" -o "STM32MP135C-DK_DDR_UTILITIES_A7.elf" @obj
Finished building target: STM32MP135C-DK_DDR_UTILITIES_A7.elf

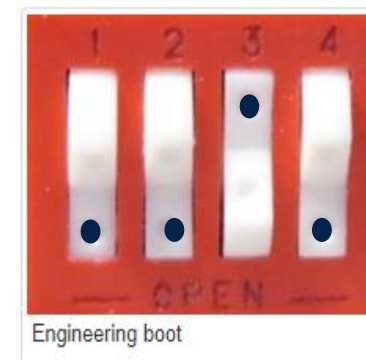
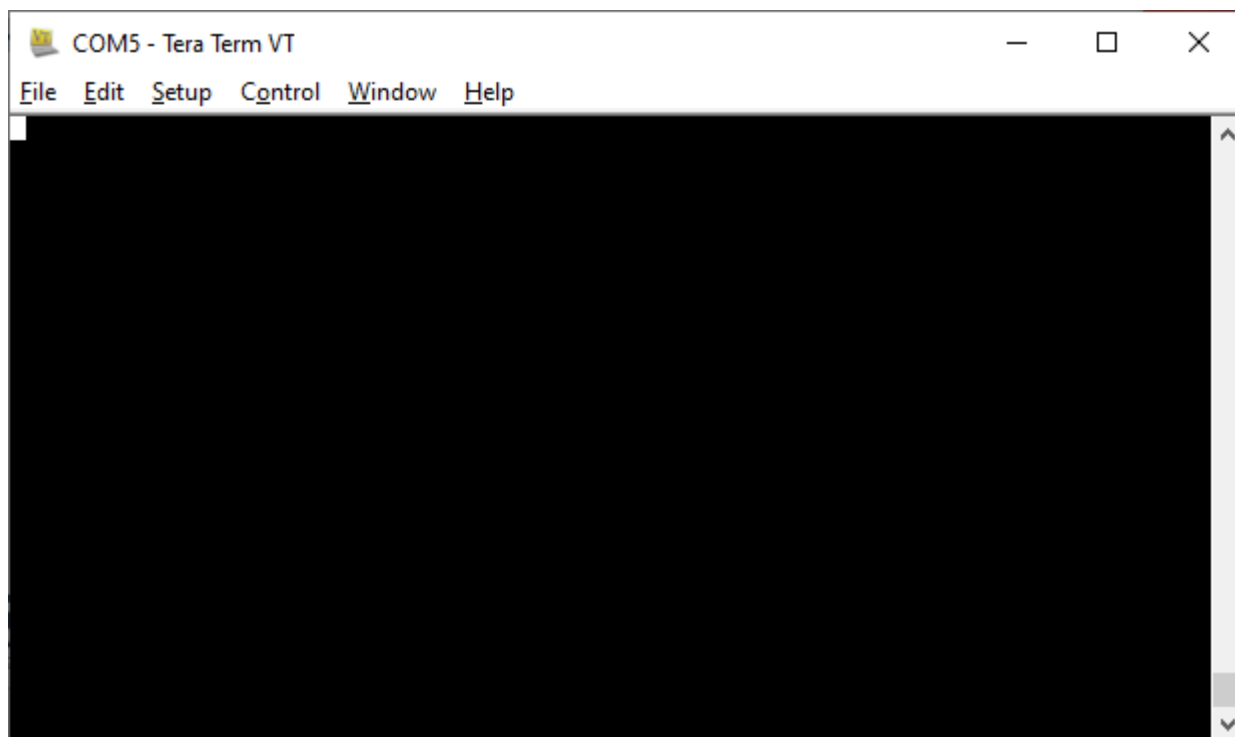
arm-none-eabi-size STM32MP135C-DK_DDR_UTILITIES_A7.elf
text data bss dec hex filename
53256 2040 26496 81792 13f80 STM32MP135C-DK_DDR_UTILITIES_A7.elf
Finished building: default.size.stdout

arm-none-eabi-objdump -h -S STM32MP135C-DK_DDR_UTILITIES_A7.elf
Finished building: STM32MP135C-DK_DDR_UTILITIES_A7.list

arm-none-eabi-objcopy -O binary STM32MP135C-DK_DDR_UTILITIES_A7.elf STM32MP135C-DK_DDR_UTILITIES_A7.bin
Finished building: STM32MP135C-DK_DDR_UTILITIES_A7.bin
```

Setup

- Set boot pin in developer mode
- Keep console opened
- Reset the board

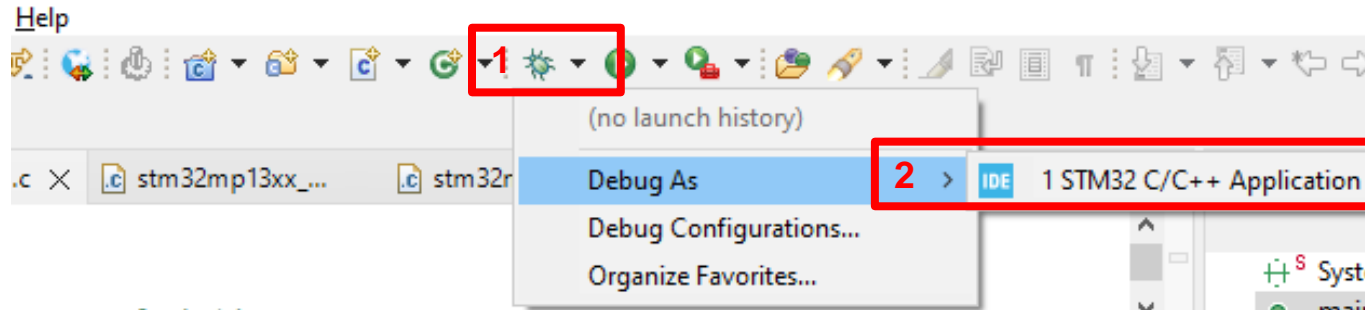


Red LED &
screen is black



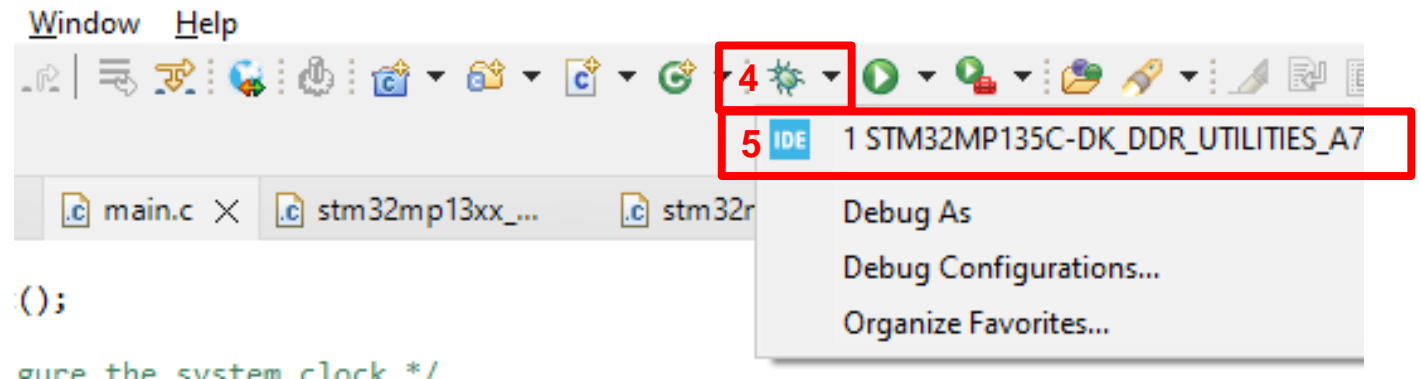
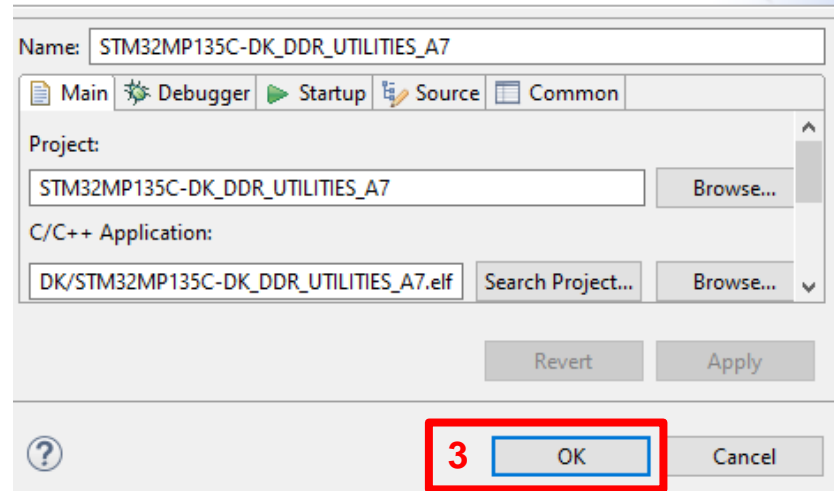
Run DDRFW-UTIL

- Create & start a debug session



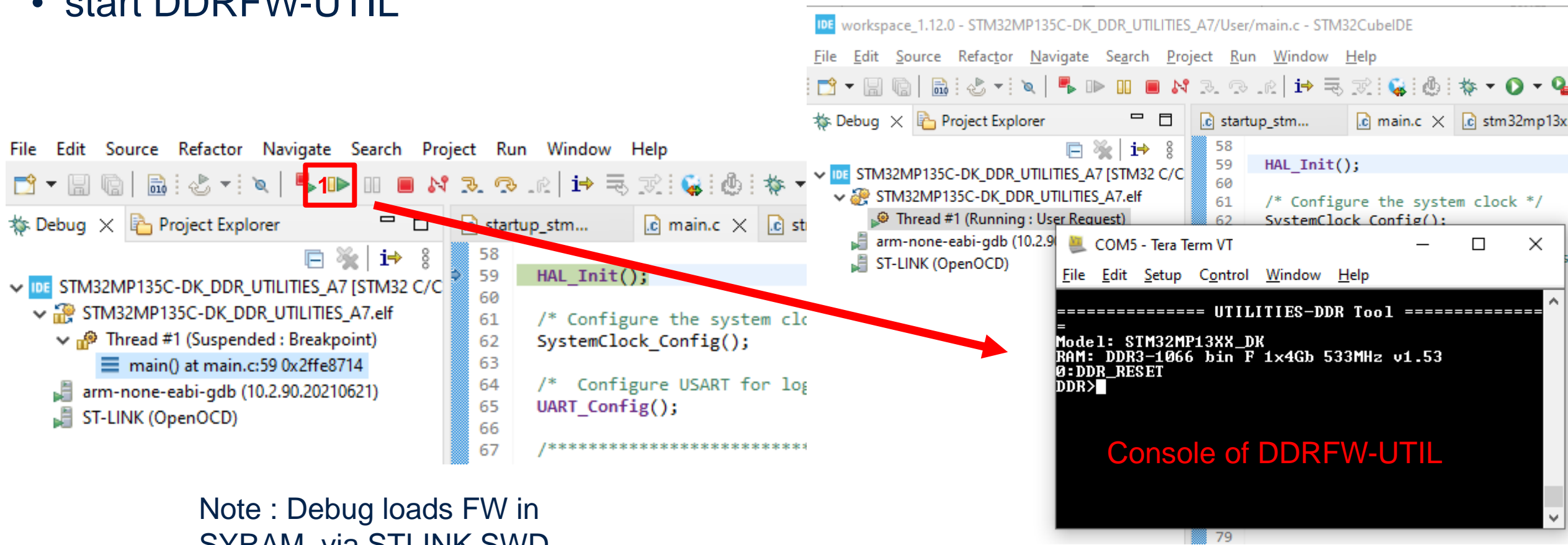
IDE Edit Configuration

Edit launch configuration properties



Run DDRFW-UTIL

- start DDRFW-UTIL



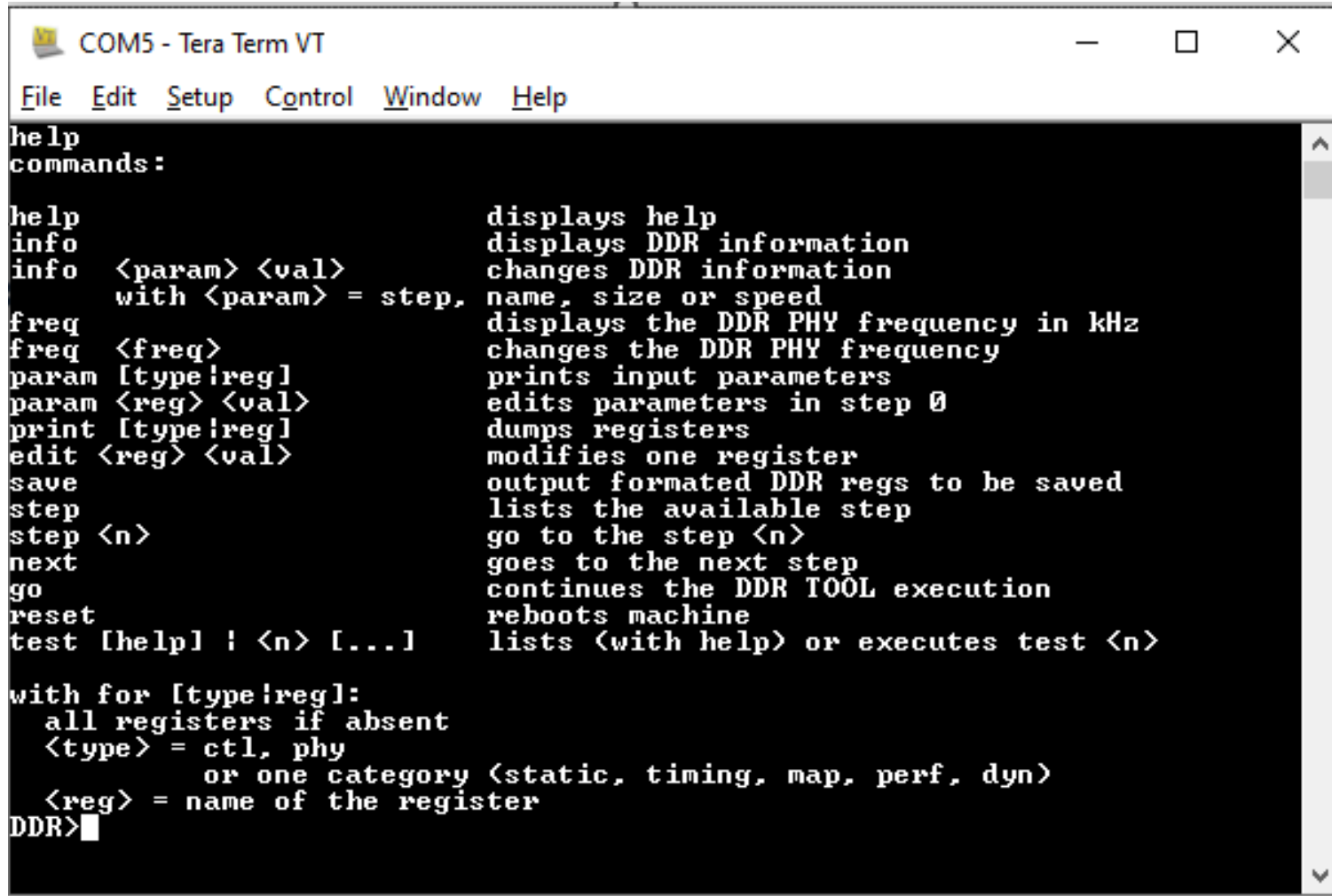
The screenshot displays the STM32CubeIDE interface. The Project Explorer on the left shows the project structure, with 'Thread #1 (Suspended: Breakpoint)' at 'main()' in 'main.c'. The main editor window shows the 'main.c' file with the 'Run' button (a green play icon) highlighted by a red box. A red arrow points from this button to a separate window titled 'COM5 - Tera Term VT'. This window shows the output of the 'UTILITIES-DDR Tool', including model and RAM information, and a prompt 'DDR>'.

```
===== UTILITIES-DDR Tool =====  
Model: STM32MP13XX_DK  
RAM: DDR3-1066 bin F 1x4Gb 533MHz v1.53  
0:DDR_RESET  
DDR>
```

Note : Debug loads FW in
SYRAM via STLINK SWD
Fw is broken at main()

DDRFW-UTIL Commands

- Commands of DDRFW-UTIL



```
COM5 - Tera Term VT
File Edit Setup Control Window Help
help
commands:

help          displays help
info          displays DDR information
info <param> <val> changes DDR information
               with <param> = step, name, size or speed
freq          displays the DDR PHY frequency in kHz
freq <freq>   changes the DDR PHY frequency
param [type|reg] prints input parameters
param <reg> <val> edits parameters in step 0
print [type|reg] dumps registers
edit <reg> <val> modifies one register
save          output formatted DDR regs to be saved
step          lists the available step
step <n>       go to the step <n>
next          goes to the next step
go            continues the DDR TOOL execution
reset         reboots machine
test [help] ! <n> [...] lists <(with help)> or executes test <n>

with for [type|reg]:
    all registers if absent
    <type> = ctl, phy
             or one category (static, timing, map, perf, dyn)
    <reg> = name of the register
DDR>
```

DDRFW-UTIL commands

- Other DDR Ctrl commands for debug

DDR>print
see register value

```
Tera Term - [disconnected] VT
File Edit Setup Control Window Help
DDR>print
==ctl.static==
mstr= 0x00040401
mrctrl0= 0x00000010
mrctrl1= 0x00000000
derateen= 0x00000000
derateint= 0x00800000
pwrctl= 0x00000000
pwrimg= 0x00400010
hwlctl= 0x00000000
rfsctl0= 0x00210000
rfsctl3= 0x00000000
crparctl0= 0x00000000
zqctl0= 0xc2000040
dfitmg0= 0x02050105
dfitmg1= 0x00000202
dfilpcf0= 0x07000000
dfiupd0= 0xc0400003
dfiupd1= 0x00000000
dfiupd2= 0x00000000
```

The "print" and "edit" directly access the CTRL and PHY registers

DDR>save
To save in xxx-template.h format

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
DDR>save
/* DDR REG VALUES TO BE SAUED */
#define DDR_MEM_NAME "DDR3-1066 bin F 1x4Gb 533MHz v1.53"
#define DDR_MEM_SPEED 533000
#define DDR_MEM_SIZE 0x20000000

/* ctl.static */
#define DDR_MSTR 0x00040001
#define DDR_MRCTRL0 0x00000010
#define DDR_MRCTRL1 0x00000000
#define DDR_DERATEEN 0x00000000
#define DDR_DERATEINT 0x00800000
#define DDR_PWRCTL 0x00000000
#define DDR_PWRIMG 0x00402010
#define DDR_HWLCTL 0x00000003
#define DDR_RFSCTL0 0x00210000
#define DDR_RFSCTL3 0x00000000
#define DDR_CRPCARCTL0 0x00000000
#define DDR_ZQCTL0 0xc2000040
#define DDR_DFITMG0 0x07020002
#define DDR_DFITMG1 0x00000404
#define DDR_DFILPCFG0 0x07000000
#define DDR_DFIUPD0 0x00400003
```

DDR>param
To change value before initialization

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
param ptr0 0
ptr0= 0x00000000
DDR>
```


Run DDR test

- sanity check of customer PCB

DDR>step 3

First initialize DDR controller

DDR> test 0

Run all the tests

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
info
step = 0 : DDR_RESET
name = DDR3-1066 bin F 1x4Gb 533MHz v1.53
size = 0x200000000
speed = 533000 kHz
DDR>
DDR>
DDR>
DDR>step
0:DDR_RESET
1:DDR_CTRL_INIT_DONE
2:DDR_PHY_INIT_DONE
3:DDR_READY
4:RUN
DDR>
DDR>
DDR>step 3
step to 3:DDR_READY
1:DDR_CTRL_INIT_DONE
2:DDR_PHY_INIT_DONE
3:DDR_READY
DDR>
```

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
test 0
invalid step 0:DDR_RESET expecting 3:DDR_READY
DDR>step 3
step to 3:DDR_READY
1:DDR_CTRL_INIT_DONE
2:DDR_PHY_INIT_DONE
3:DDR_READY
DDR>test 0
result 1:Test Simple DataBus = Passed
result 2:Test DataBusWalking0 = Passed
result 3:Test DataBusWalking1 = Passed
result 4:Test AddressBus = Passed
result 5:Test MemDevice = Passed
result 6:Test SimultaneousSwitchingOutput = Passed
result 7:Test Noise = Passed
result 8:Test NoiseBurst = Passed
result 9:Test Random = Passed
result 10:Test FrequencySelectivePattern = Passed
result 11:Test BlockSequential = Passed
result 12:Test Checkerboard = Passed
result 13:Test BitSpread = Passed
result 14:Test BitFlip = Passed
result 15:Test WalkingZeroes = Passed
result 16:Test WalkingOnes = Passed
Result: Pass [Test All]
DDR>
```

Run DDR test

- Check the memory integrity, the bus

- Test the integrity of a physical memory

Check each bit of the memory of first 100Mbytes (set 0 and 1)

DDR> test 5 0x6400000 0xC0000000



- Test the Simultaneous Switching Output.

Verifies successive reads and writes to the same memory word, holding one bit constant while toggling all other data bits simultaneously

Over the first 512 Kbytes

DDR> test 6 0x80000 0xC0000000

- Test the data bus

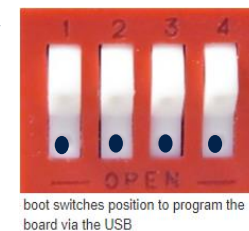
Verifies each data line by walking 1 on fixed address.

Test the data bus wiring in a memory region by performing a walking 1's test at a fixed address within that region.

DDR> test 1 0xC0000000

DDRFW-UTIL in standalone

- DDRFW-UTIL started in SYSRAM by USB DFU ROM code
 - Set boot pin to Boot from USB , connect CN7 USB OTG cable to PC
 - Use the .stm32 with STM32CubeProgrammer
- `cd C:"\Program Files"\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin`
- `STM32_Programmer_CLI.exe -c port=usb1 -w C:\ST_Workshop\utilities-ddr_STM32MP135C-DK\DDR_Tool\STM32MP135C-DK\STM32MP135C-DK_DDR_UTILITIES_A7\DK\STM32MP135C-DK_DDR_UTILITIES_A7.stm32 0x01 --start 0x01`



```
Administrator: Command Prompt - STM32_Programmer_CLI.exe -c port=usb1 -w C:\Users\combette\MP13_workshop_23\STM32DDRFW-UTIL_for_STM...
Board : --
Device ID : 0x0501
Device name :
Device type : MPU
Device CPU : Cortex-A7
Start operation achieved successfully

C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin>STM32_Programmer_CLI.exe -c port=usb1 -w C:\Users\combette\MP13_workshop_23\STM32DDRFW-UTIL_for_STM32MP135C-DK\utilities-ddr\Binaries\STM32MP135C-DK_DDR_UTILITIES_A7\STM32MP135C-DK_DDR_UTILITIES_A7_signed.stm32 0x01 --start 0x01

-----
STM32CubeProgrammer v2.12.0
-----

USB speed : High Speed (480MBit/s)
Manuf. ID : STMicroelectronics
Product ID : DFU in HS Mode @Device ID /0x501, @Revision ID /0x1003
SN : 002C000C3232511538303631
DFU protocol: 1.1
Board : --
Device ID : 0x0501
Device name : STM32MP13xx
Device type : MPU
Revision ID : --
Device CPU : Cortex-A7

Memory Programming ...
Opening and parsing file: STM32MP135C-DK_DDR_UTILITIES_A7_signed.stm32
File : STM32MP135C-DK_DDR_UTILITIES_A7_signed.stm32
Size : 54.50 KB
Partition ID : 0x01

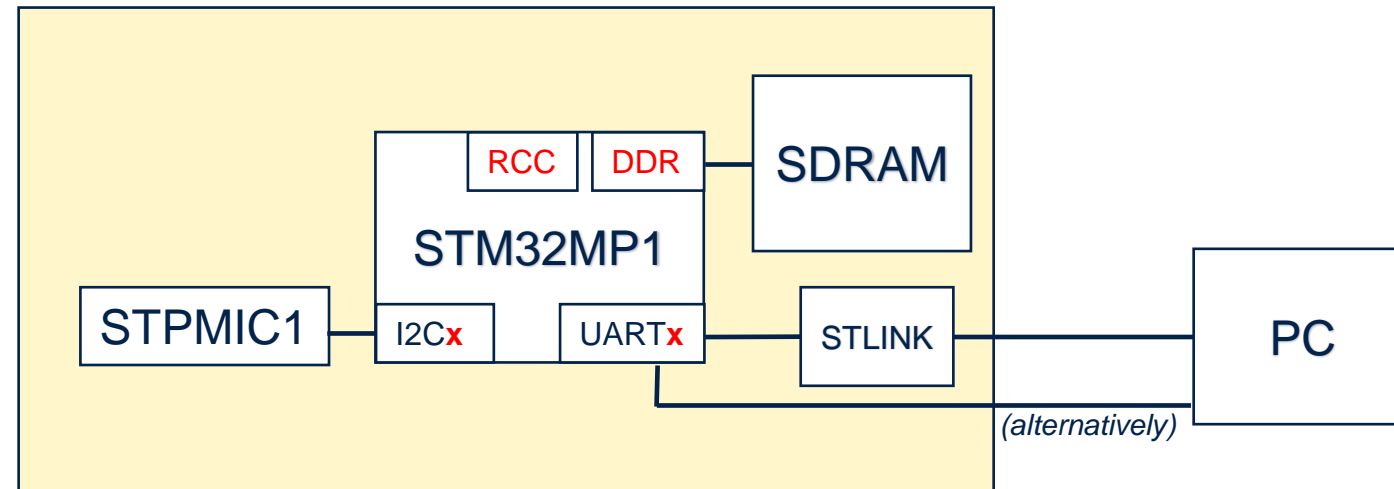
Download in Progress:
38%
```

```
COM9 - Tera Term VT
File Edit Setup Control Window Help

===== UTILITIES-DDR Tool =====
Model: STM32MP13XX_DK
RAM: DDR3-1066 bin F 1x4Gb 533MHz v1.53
0:DDR_RESET
DDR>
```

Adapt DDRFW-UTIL to your board

Possible DDRFW-UTIL modifications



Project

```
<utilities-ddr_STM32MP135C-DK>
|_ README.md
|_ <Binaries>
|   |_ <STM32MP135C-DK_DDR_UTILITIES_A7>
|       |_ STM32MP135C-DK_DDR_UTILITIES_A7_signed.stm32 (to be loaded on SDCARD or via USB DFU)
|_ <DDR_Tool>
|   |_ <Common>
|       |_ <Inc>
|           |_ stm32mp_util_def.h
|           |_ system_time.h
|           |_ log.h
|       |_ <Src>
|           |_ system_time.c
|           |_ syscalls.c
|   |_ <STM32MP135C-DK>
|       |_ <Inc>
|           |_ ddr_tests.h
|           |_ ddr_tool.h
|           |_ ddr_tool_util.h
|           |_ main.h
|           |_ RTE_Components.h
|           |_ stm32_device_hal.h
|           |_ stm32mp13xx-util-ddr3-4Gb-template.h
|           |_ stm32mp13xx_disco_conf.h
|           |_ stm32mp13xx_hal_conf.h
|           |_ stm32mp13xx_it.h
|           |_ stm32mp13xx-util-lpddr2-4Gb-template.h
|           |_ stm32mp13xx-util-lpddr3-4Gb-template.h
|           |_ stm32mp_util_conf.h
|           |_ stm32mp_util_ddr_conf.h
|       |_ <Src>
|           |_ ddr_tests.c
|           |_ ddr_tool.c (common file that implements HAL DDR Interactive function with all available tool commands)
|           |_ ddr_tool_util.c
|           |_ main.c
|           |_ stm32mp13xx_hal_msp.c
|           |_ stm32mp13xx_it.c
|   |_ <STM32MP135C-DK_DDR_UTILITIES_A7>
|       |_ stm32mp13xx_a7_sysram.ld
|       |_ project
|       |_ cproject
|_ <Drivers>
|   |_ <BSP>
|       |_ <STM32MP13xx_DISCO>
|   |_ <CMSIS>
|       |_ <MP1>
|           |_ ...
|       |_ <Device>
|           |_ <ST>
|               |_ <STM32MP13xx> (board specific file that handles stpmic regulators and i2C, leds)
|               |_ <STM32MP13xx_HAL_Driver>
```

(DDR registers setting given as example)

(given as example)

(given as example)

(to be adapted for customize board)

(select DDR according to board memory)

(common file that implements all DDR tests)

(board specific file that handles uart communication)

Customize to PCB

DDR PHY reg values depending on DDR

settings for UART, PMIC I2C
PLL frequency of DDR controller

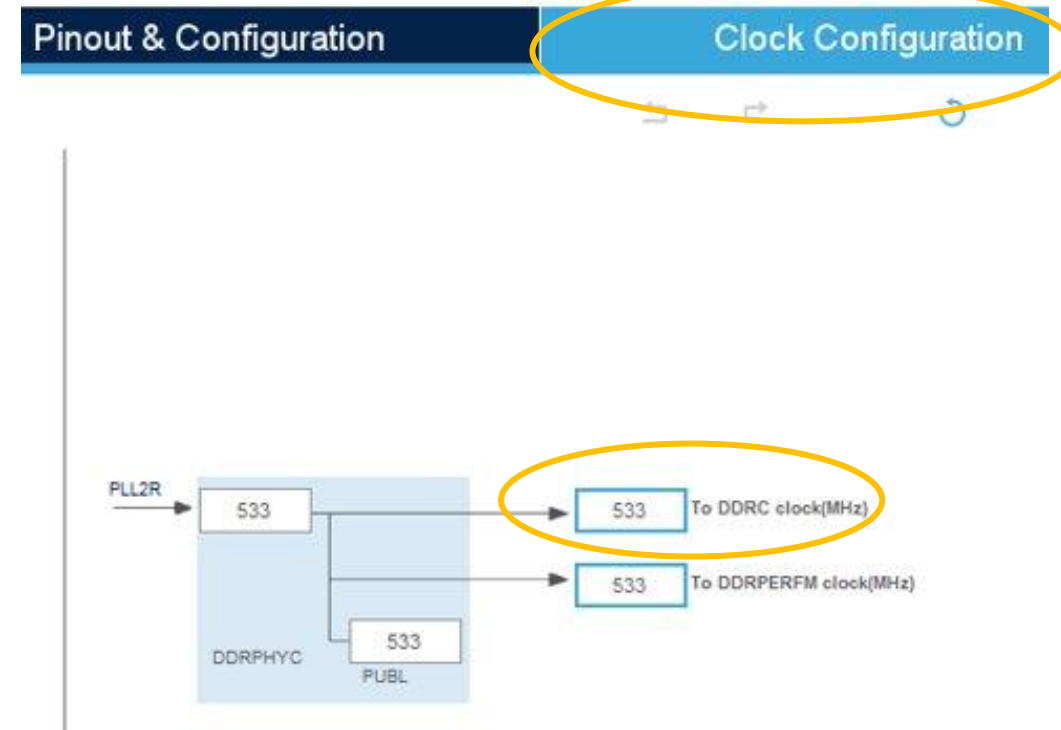
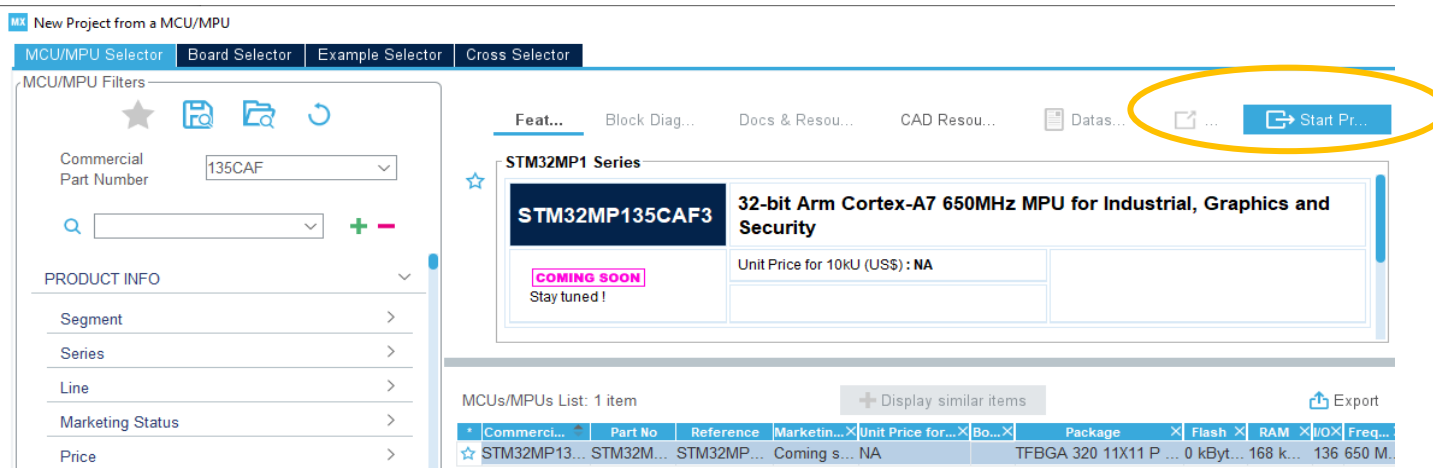
```
stm32mp_util... X stm32mp13xx... stm32mp13xx...
31 #define UTIL_PLL2_P 2
32 #define UTIL_PLL2_Q 2
33 #define UTIL_PLL2_R 1
34 #define UTIL_PLL2_FRACV 0x1400
35 #define UTIL_PLL2_MODE RCC_PLL_FRACTIONAL
36 #define UTIL_PLL2_DIVN_MIN 24
37 #define UTIL_PLL2_DIVN_MAX 199
38
39 /* UART related configuration */
40 #define UTIL_UART_INSTANCE UTIL_UART4
41 #define UTIL_UART_TX_PIN GPIO_PIN_6
42 #define UTIL_UART_TX_GPIO_PORT UTIL_GPIOD
43 #define UTIL_UART_TX_AF GPIO_AF8_UART4
44 #define UTIL_UART_RX_PIN GPIO_PIN_8
45 #define UTIL_UART_RX_GPIO_PORT UTIL_GPIOD
46 #define UTIL_UART_RX_AF GPIO_AF8_UART4
47 #define UTIL_UART_BAUDRATE 115200
48 #define UTIL_UART_WORDLENGTH UART_WORDLENGTH_8B
49 #define UTIL_UART_STOPBITS UART_STOPBITS_1
50 #define UTIL_UART_PARITY UART_PARITY_NONE
51 #define UTIL_UART_HWFLOWCTL UART_HWCONTROL_NONE
52 #define UTIL_UART_MODE UART_MODE_TX_RX
53 #define UTIL_UART_OVERSAMPLING UART_OVERSAMPLING_16
54
55 /* PMIC related configuration */
56 #define UTIL_USE_PMIC 1
57 #define UTIL_PMIC_I2C_PORT UTIL_I2C4
58 #define UTIL_PMIC_I2C_SCL_PIN GPIO_PIN_15
59 #define UTIL_PMIC_I2C_SCL_GPIO_PORT UTIL_GPIOE
```

Customize to PCB

For new DDR memory type / frequency : create a new xxx-template.h with the help of STM32CubeMX

1 Open new project with STM32CubeMx

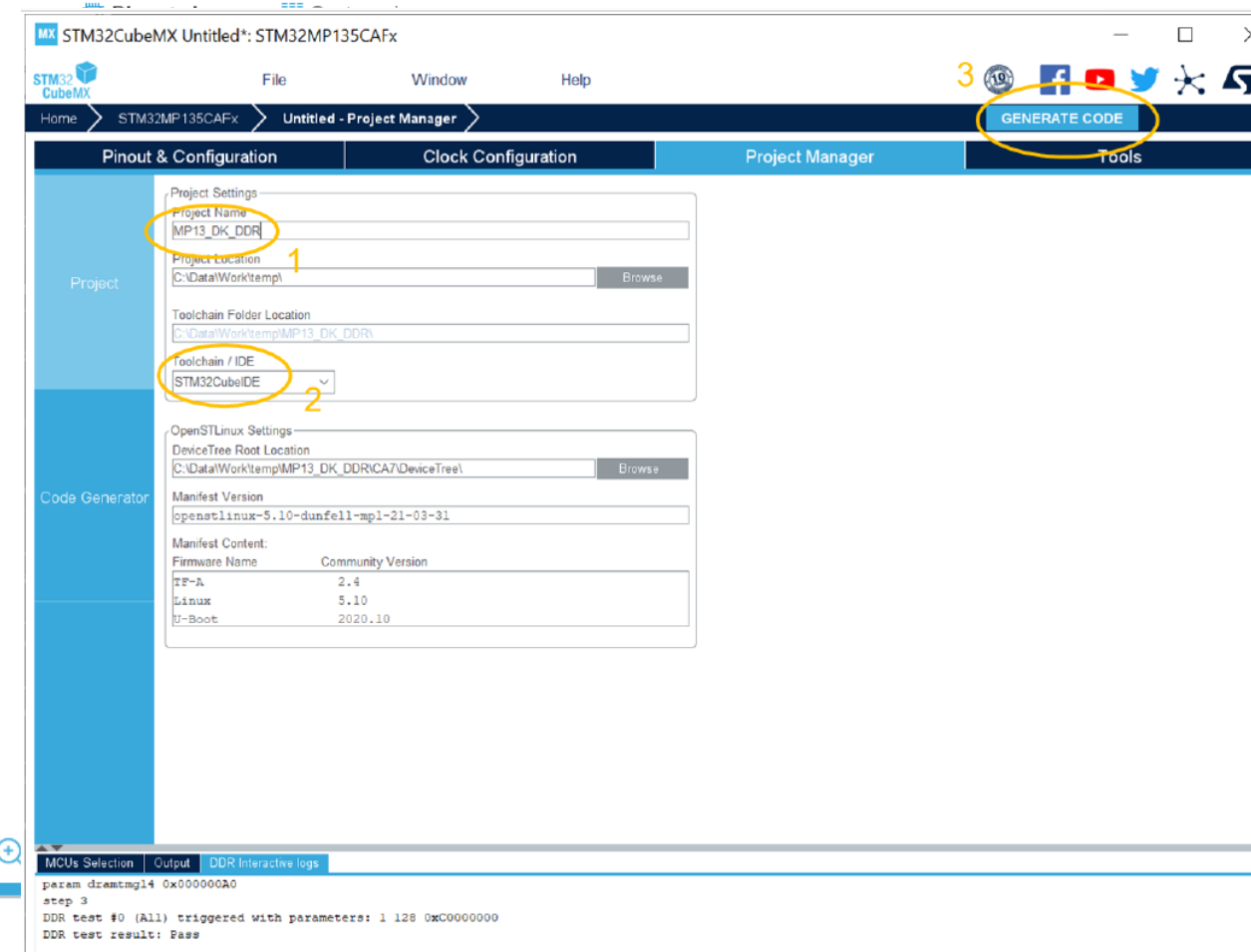
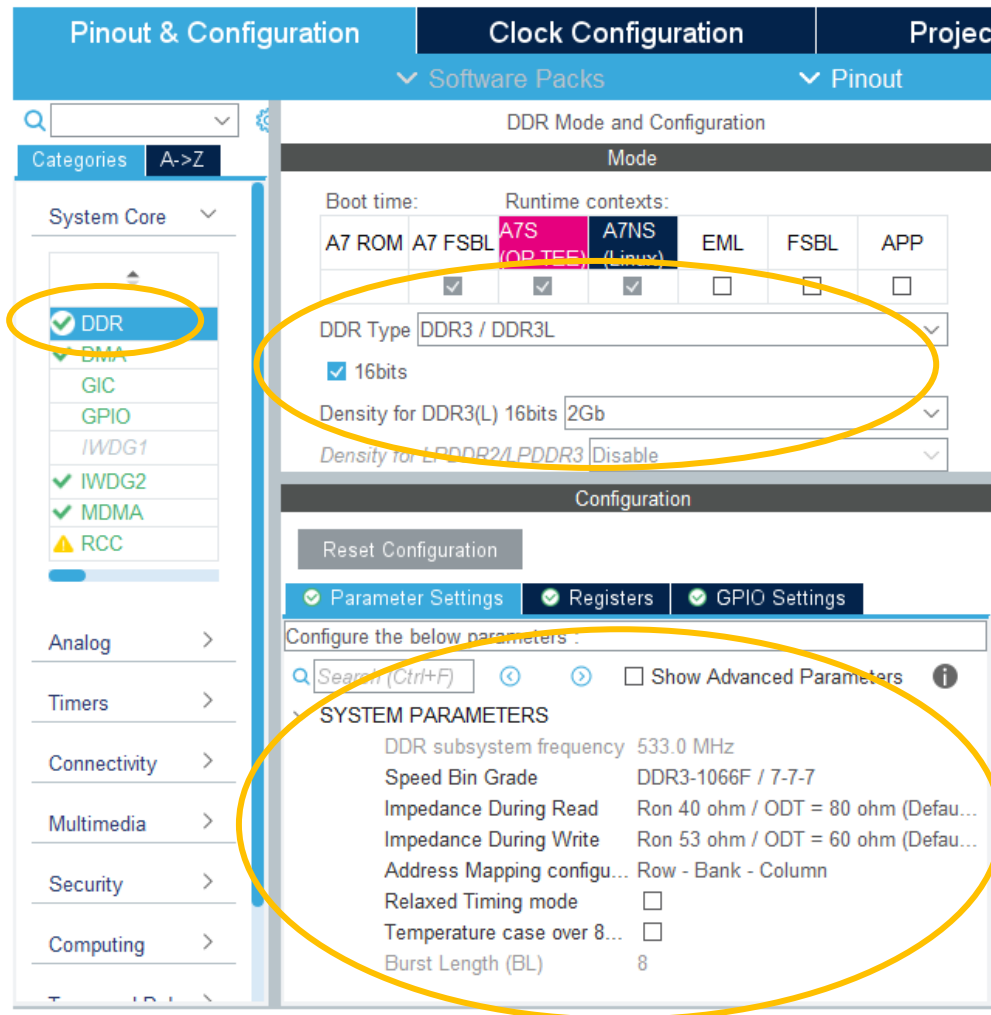
2 Set DDR frequency



Customize to PCB

3 set DDR type, size, seed/grade, impedances

4 Generate device tree to compute the new DDR timings



Customize to PCB

5 Copy DDR Setup .dtsi file -> new xxx-template.h file

6 report the PLL2 setting -> in stm32_util_conf.h

The screenshot shows a file explorer on the left with a tree view containing folders like 'Quick access', 'OneDrive', 'STMicroelectronics', 'This PC', 'Network', '4.0-DK2-Default', 'CA7', 'DeviceTree', '4.0-DK2-Default', 'kernel', 'optee-os', 'tf-a' (highlighted), 'u-boot', 'CM4', 'Common', 'Drivers', 'EWARM', and 'Middleware'. The main window shows a table of files with columns 'Name', 'Status', and 'Date modified'. The file 'stm32mp15-mx_ini.dtsi' is highlighted. Below the table, the contents of this file are shown in a Notepad window. The file is generated by STMicroelectronics STM32CubeMX DDR Tool and contains various defines for DDR configuration.

Name	Status	Date modified
stm32mp15-mx_ini.dtsi	✓	8/1/2022 1:07
stm32mp157c-4.0-dk2-default-mx.dts	✓	8/1/2022 1:07
stm32mp157c-4.0-dk2-default-mx-fw-config.dts	✓	8/1/2022 1:07

```
* File generated by STMicroelectronics STM32CubeMX DDR Tool
* DDR type: DDR3 / DDR3L
* DDR width: 16bits
* DDR density: 4Gb
* System frequency: 533000kHz
* Relaxed Timing Mode: false
* Address mapping type: RBC
*
* Save Date: 2022.08.01, save Time: 13:07:43
*/

#define DDR_MEM_NAME      "DDR3-DDR3L 16bits 533000kHz"
#define DDR_MEM_SPEED     533000
#define DDR_MEM_SIZE      0x20000000

#define DDR_MSTR 0x00041401
#define DDR_MRCTRL0 0x00000010
#define DDR_MRCTRL1 0x00000000
#define DDR_DERATEEN 0x00000000
#define DDR_DERATEINT 0x00080000
#define DDR_PWRCTL 0x00000000
#define DDR_PWRTMG 0x00400010
#define DDR_HWLPCTL 0x00000000
```

