实验：低功耗系统设计
--基于NUCLEO-L476RG

STM32 L4

life.augmented

目标

- 本动手实验的目的是学会自己构建工程，实现STM32L476 不同的低功耗模式和理解如何进入、退出的触发机制，并且对比分析实测功耗和理论推导功耗。

- 实验分成四个个部分:

  - ***介绍*** :讲解STM32L476 关键的功耗优化特性 (20分钟)

  - ***任务1***: 使用STM32CubeMX 建立一个基于NUCLEO-L476RG 板子的功耗测试工程 (15分钟)

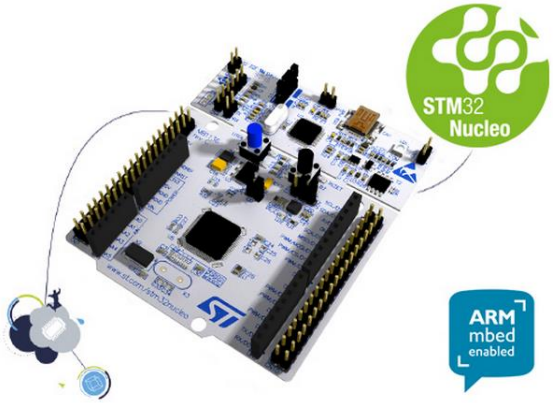
  - ***任务2***：使用CubeMX来计算NUCLEO-L476RG在不同功耗模式下的理论功耗 (15分钟)

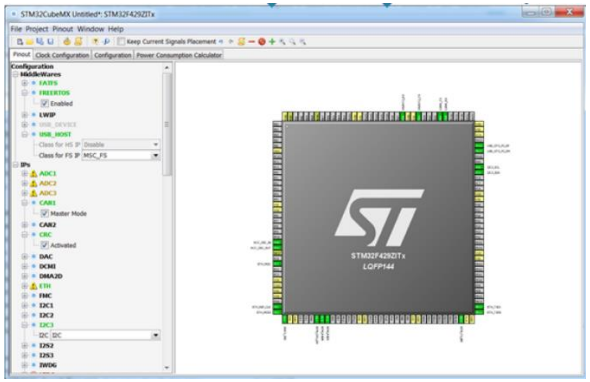
  - ***任务3***: 实际测量STM32L476 板子在不同功耗模式下的功耗值，并且和使用STM32CubeMX理论推算的功耗值进行比较分析(40 分钟)

- 总共时间: 90 分钟

实验准备

- 硬件准备
  - Nucleo-L476RG评估板
  - 可测试电流的示波器/万用表
  - Mini USB线一根
  - PC一台
- 软件准备
  - IAR V6.7.0 或者以上版本
  - STM32CubeL4 V1.6.0
  - STM32CubeMX V4.19
- 预计时间: 90分钟

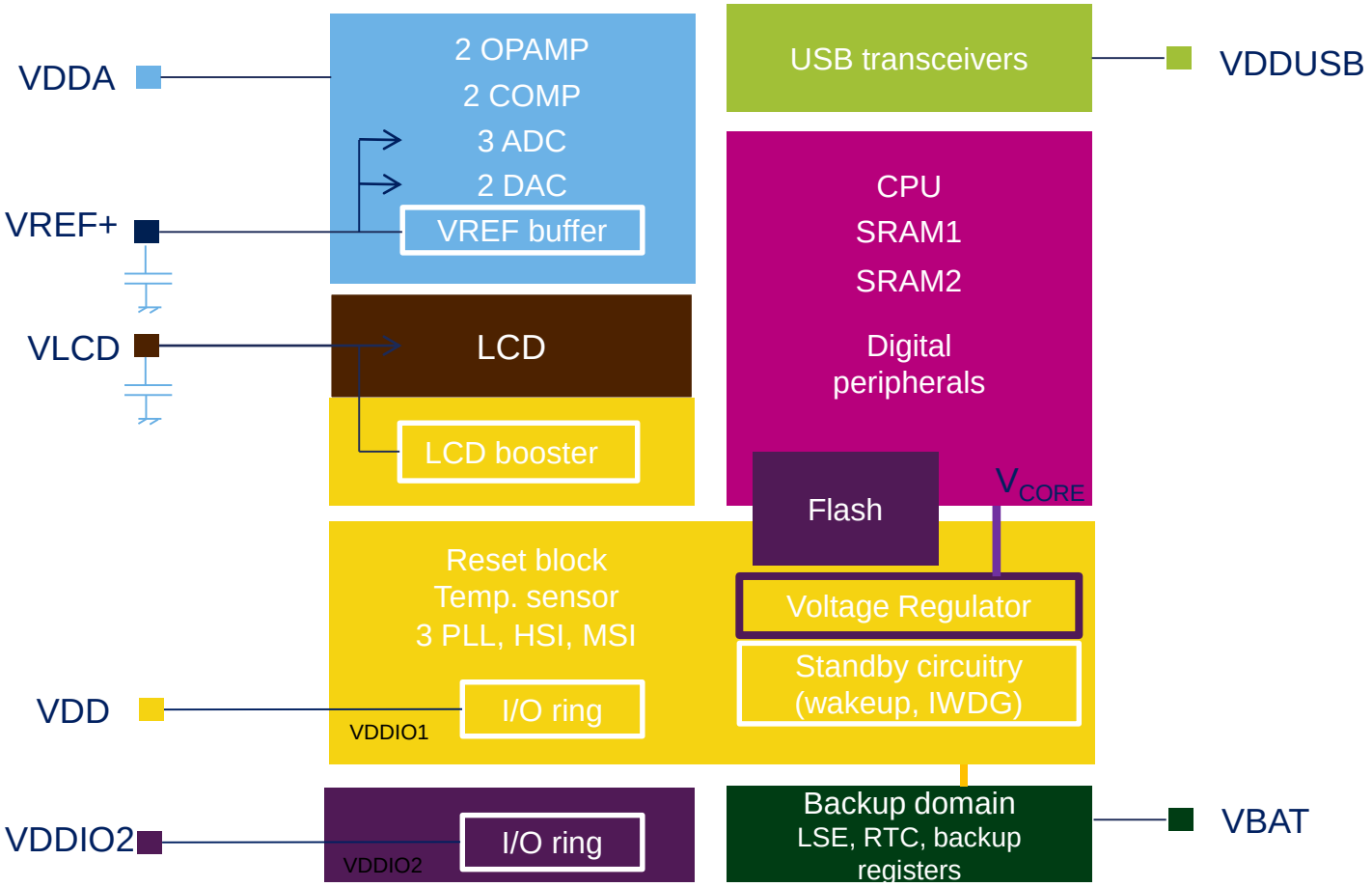




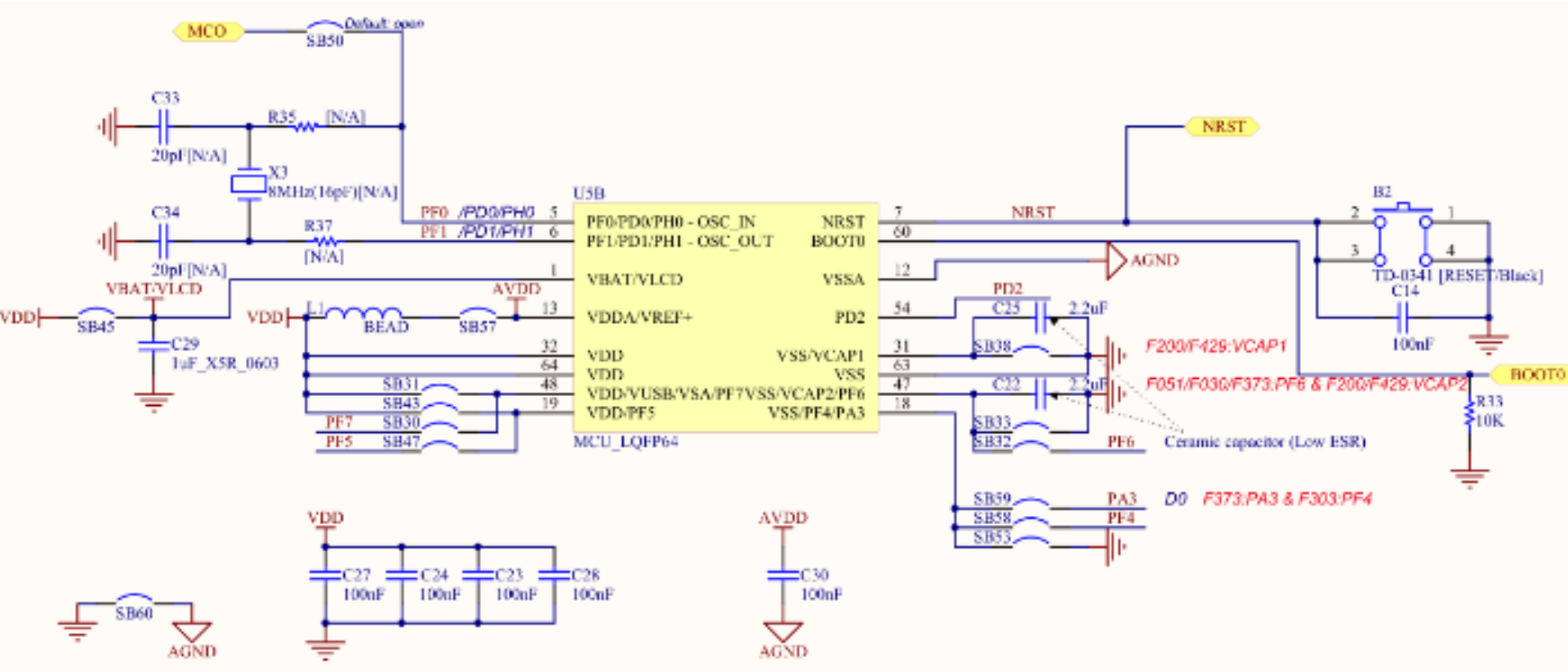
实验环节
- 介绍

life.augmented

MCU电源框架图

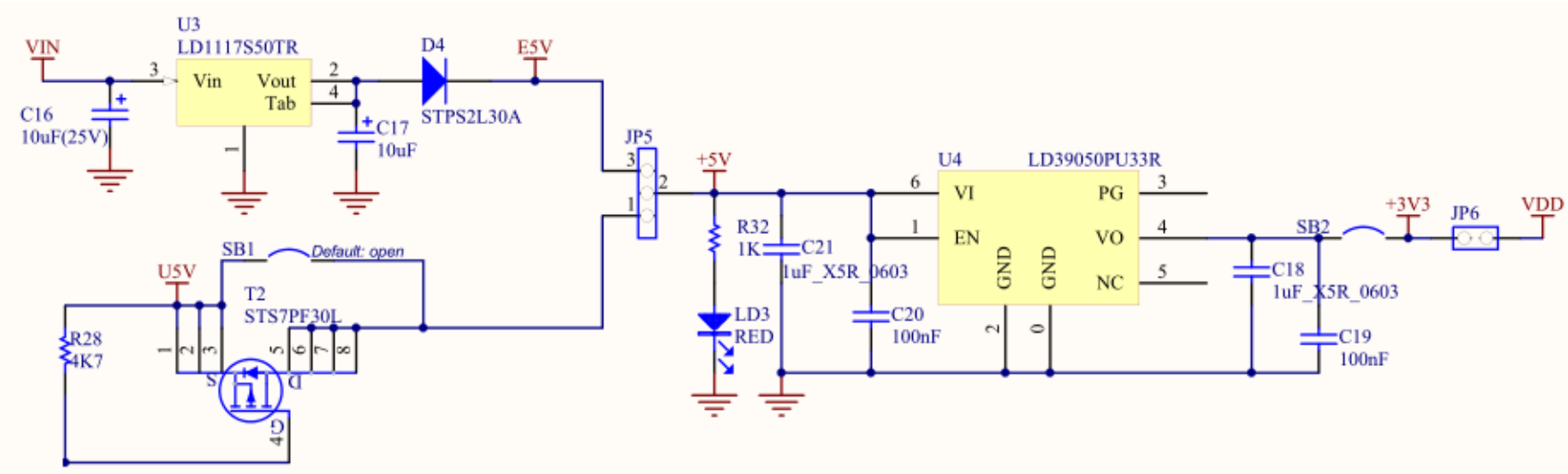

VDDA
2 OPAMP
2 COMP
3 ADC
2 DAC
VREF buffer
USB transceivers
VDDUSB
VREF+
CPU
SRAM1
SRAM2
Digital peripherals
VLCD
LCD
LCD booster
Flash
VCORE
Reset block
Temp. sensor
3 PLL, HSI, MSI
Voltage Regulator
Standby circuitry (wakeup, IWDG)
VDD
I/O ring
VDDIO1
VDDIO2
I/O ring
VDDIO2
Backup domain
LSE, RTC, backup registers
VBAT

Nucleo-L476电源框架图



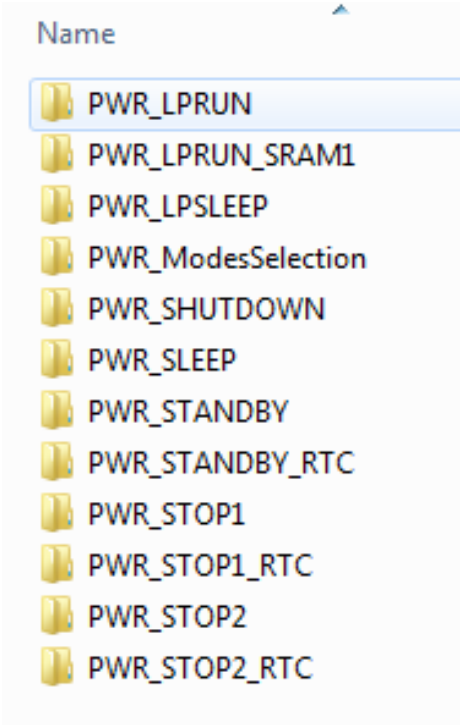
供电部分原理图

- 供电方式
  - 直接USB提供5V供电
  - 使用外部E5V来给整个系统供电
  - 通过JP5这个跳线帽来切换
- 测试点
  - 在JP6处串上万用表或者夹上电流探头



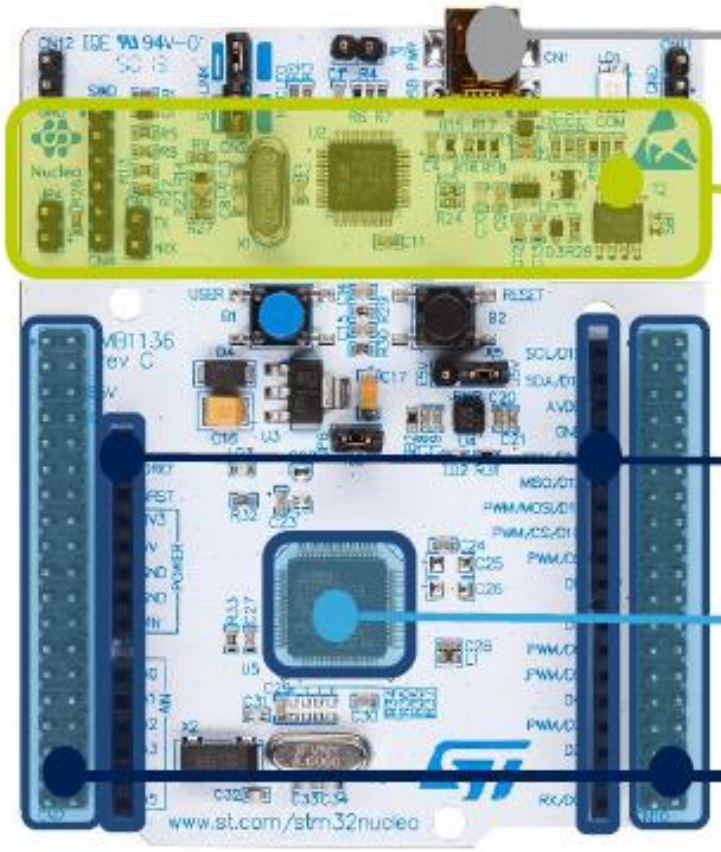
实验案例：STM32L4xx

- STM32L4xx的低功耗模式比较多：
  - Sleep模式
  - Low-power run模式
  - Low-power sleep模式
  - STOP1模式
  - STOP2模式
  - Standby with SRAM2模式
  - Standby 模式
  - Shutdown模式



- 所有的范例程序都包含在一下路径：
  - STM32Cube_FW_L4_V1.x.x\Projects\STM32L476RG-Nucleo\Examples\PWR
  - STM32Cube_FW_L4_V1.x.x\Projects\STM32L476RG-Nucleo\Examples_LL\PWR
  - STM32Cube_FW_L4_V1.x.x\Projects\STM32L476RG-Nucleo\Examples_MIX\PWR

NUCLEO-L476RG



Flexible board power supply through USB or external source

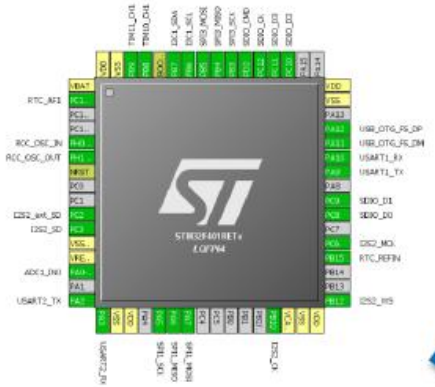
Integrated Debugging and programming probe

Arduino UNO extension connectors easy access to add-ons (*)

STM32 Microcontroller

Morpho extension headers: Direct access to all STM32 I/Os

STM32CubeMX



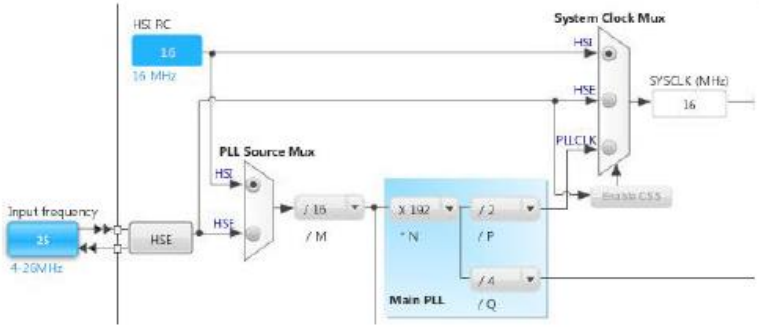
Pinout Wizard

Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

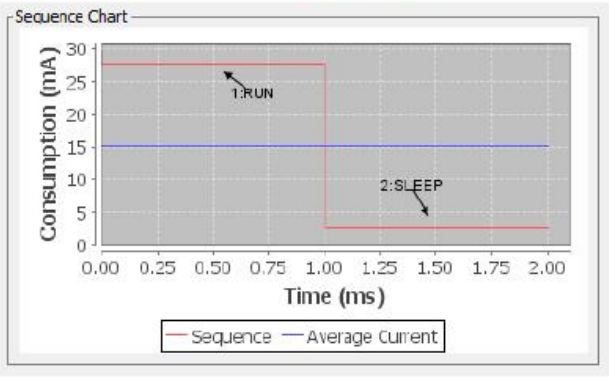
Baud Rate
BaudRate must be between **110 Bits/s** and **10.5 MBits/s**.

Peripherals & Middleware Wizard

Clock Tree wizard



Power Consumption Wizard

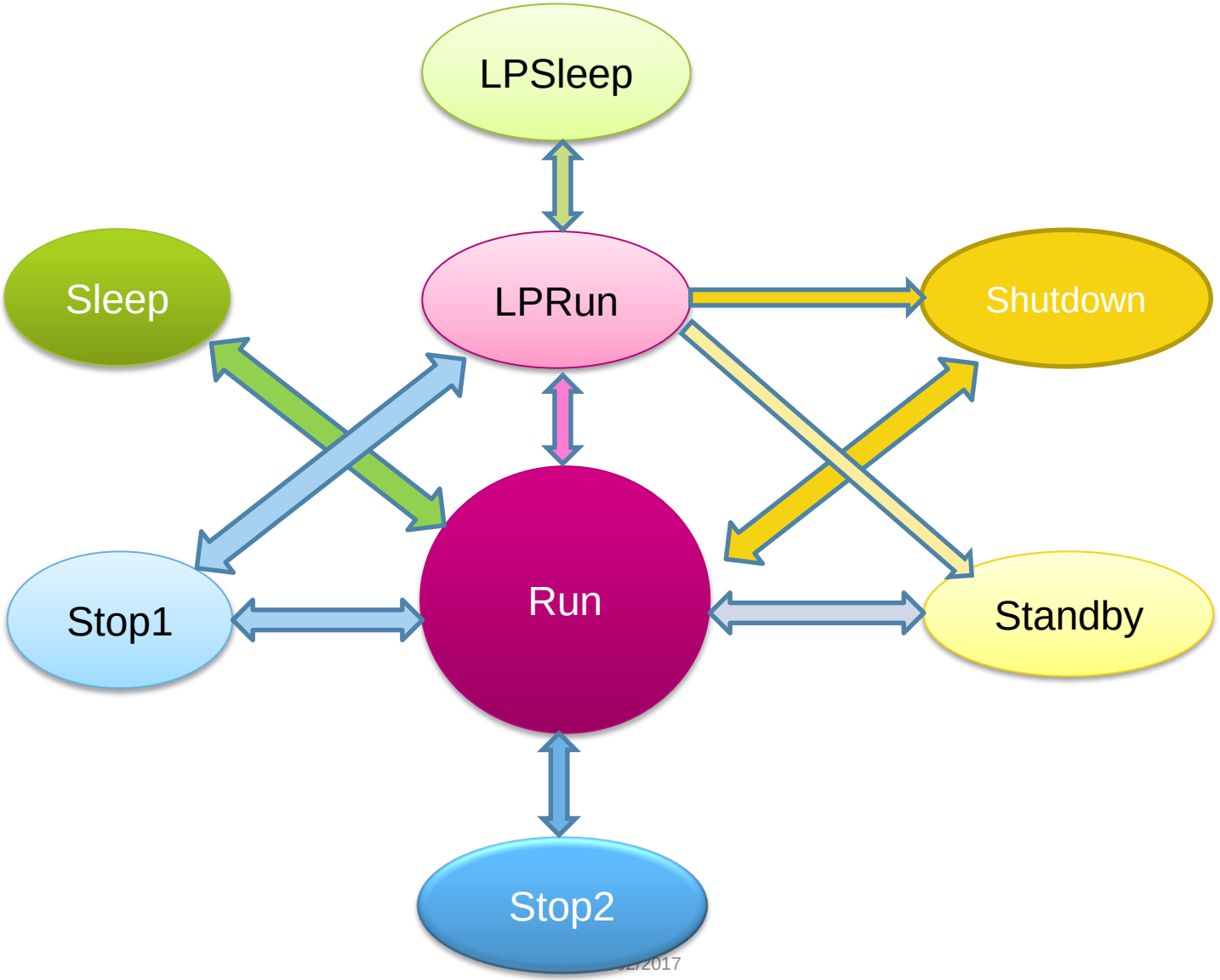


Low-power modes summary

Mode	Regulator	CPU	Flash	SRAM	Clocks	Peripherals In Bold : wakeup source	Consumption @ 1.8V	Wakeup time
Run	R1	Yes	ON(1)	ON	Any	All	127 µA/MHz	N/A
	R2					All except OTG, SDMMC, RNG	111 µA/MHz	
LPRun	LPR	Yes	ON(1)	ON	Any *except PLL*	All except OTG, SDMMC, RNG	136 µA/MHz	TBD
Sleep	R1	No	ON(1)	ON(2)	Any	All **Any IT or event**	37 µA/MHz	6 cycles
	R2						35 µA/MHz	
LPSleep	LPR	No	ON(1)	ON(2)	Any *except PLL*	All except OTG, SDMMC, RNG **Any IT or event**	40 µA/MHz	6 cycles
Stop 1	LPR	No	OFF	ON	LSE/LSI	**Reset pin, all I/Os** **BOR,PVD,PVM,RTC,LCD,IWDG,** **COMPx**,DACx,OPAMPx,**USARTx,** **LPUART,I2Cx,LPTIMx,OTG_FS,** **SWPMI**	7.3µA w/o RTC 7.6 µA w/RTC	4 µA RAM 6 µA Flash
Stop 2	LPR	No	OFF	ON	LSE/LSI	**Reset pin, all I/Os** **BOR,PVD,PVM,RTC,LCD,IWDG,** **COMPx,LPUART,I2C3,LPTIM1**	1.2 µA w/o RTC 1.4 µA w/RTC	5 µs RAM 7 µs Flash
Standby	LPR	DOWN	OFF	SRAM2 ON	LSE/LSI	**Reset pin, 5 WKUPx pins** **BOR, RTC, IWDG**	+ 235 nA	14 µs
	OFF			DOWN			128 nA w/o RTC 433 nA w/RTC	
Shutdown	OFF	DOWN	OFF	DOWN	LSE	**Reset pin, 5 WKUPx pins** **RTC**	43 nA w/o RTC 265 nA w/RTC	256 µs

1. Can be put in power-down and clock can be gated off
2. SRAM1 and SRAM2 can be gated off independently

Low-power modes transitions



STM32L4 Power Mode

Run mode

Run Mode Range 1
Ex: execution from Flash

- Cortex M4
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)
- Main regulator (MR)
  - Range 1 (up to 80MHZ)
  - Range 2 (up to 26MHZ)
- Low Power regulator (LPR) up to 2MHz

Range 1
127uA/MHz at 80 MHz
(10.2mA)

Range 2
111uA/MHz at 26 MHz
(2.9 mA)

Available Peripheral

- GPIO
- DMA
- FSMC
- QUADSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Available Clock

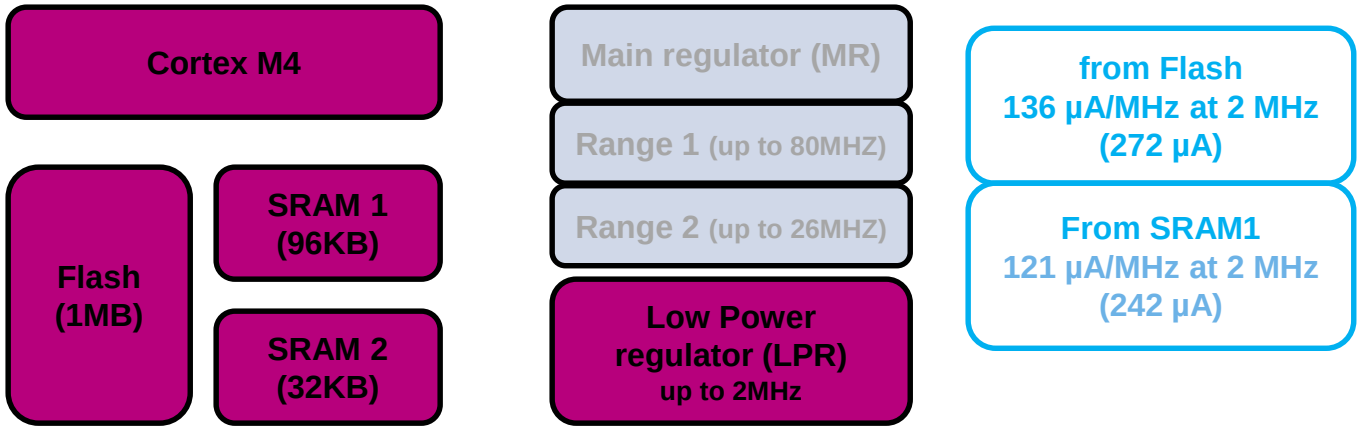
- HSI
- HSE
- LSI
- LSE
- MSI

Active cell | Frozen cell | Cell in power-down | Available Periph and clock

STM32L4 Power Mode

Low-power run mode

Low-power run mode
Ex: execution from Flash



Available Peripheral

- GPIO
- DMA
- FSMC
- QUADSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Active cell | Frozen cell | Cell in power-down | Available Periph and clock

STM32L4 Power Mode

Sleep mode

Sleep Mode Range 1
Ex: Flash ON, SRAMs ON (default)

Zzz

Cortex M4

Flash (1MB)

SRAM 1 (96KB)

SRAM 2 (32KB)

Main regulator (MR)

Range 1 (up to 80MHZ)

Range 2 (up to 26MHZ)

Low Power regulator (LPR) up to 2MHz

Range 1
37 µA/MHz at 80 MHz
(2,96 mA)

Range 2
35 µA/MHz at 26 MHz
(0,92 mA)

Available Peripheral

- GPIO
- DMA
- FSMC
- QUADSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Active cell

Frozen cell

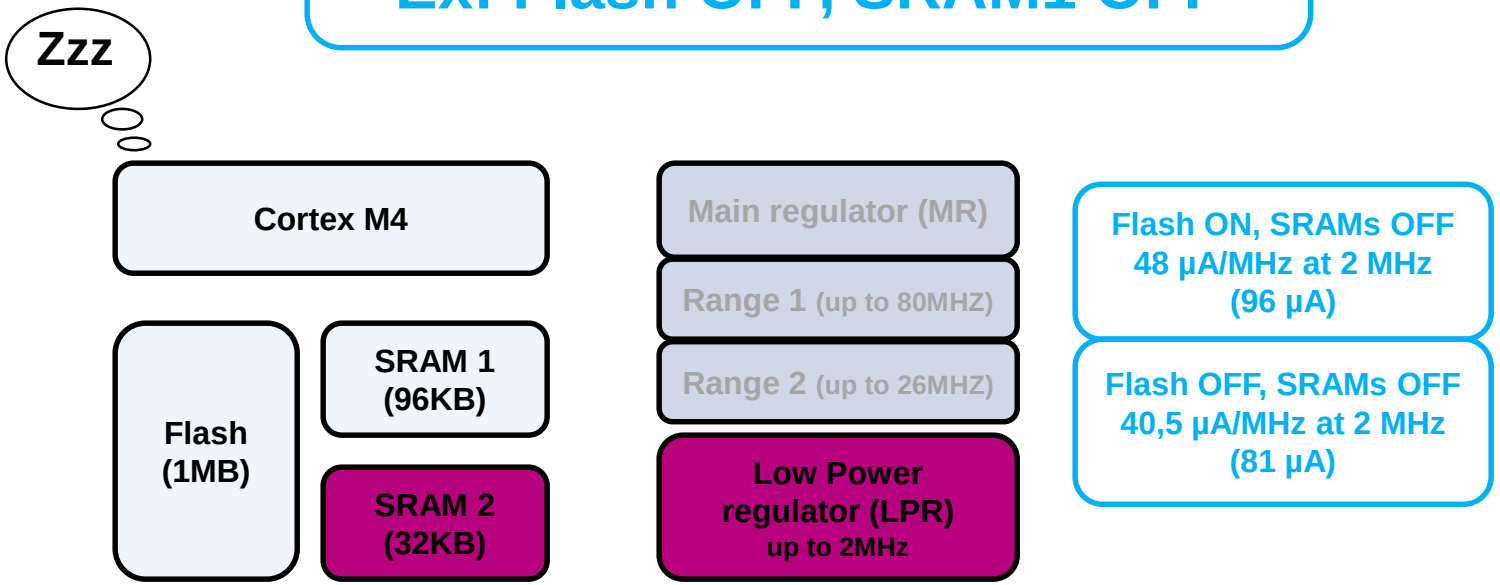
Cell in power-down

Available Periph and clock

STM32L4 Power Mode

Low-power sleep mode

Low-power sleep mode
Ex: Flash OFF, SRAM1 OFF



Available Peripheral

- GPIO
- DMA
- FSMC
- QUADSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Active cell | Frozen cell | Cell in power-down | Available Periph and clock

STM32L4 Power Mode

Stop 1 Mode

Available Peripheral

I/Os kept, and configurable

- GPIO
- DMA
- FSMC
- QSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Stop 1 w/ RTC on LSE quartz ⇨ **7,9 µA @3.0V**
7.6 µA @1.8V

- Cortex M4
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)

- Main regulator (MR)
- Low Power regulator (LPR)
- Backup domain
  - Backup Register (32x32-bits)
  - RTC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Wake-up event

- NRST
- BOR
- PVD
- PVM
- RTC + Tamper
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SWPMI
- COMP
- LPTIM 1
- LPTIM 2
- IWDG
- GPIOs

6us wake-up from Flash
4us wake-up from RAM

STM32L4 Power Mode

Stop 1 Mode

Available Peripheral

I/Os kept, and configurable

- GPIO
- DMA
- FSMC
- QSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

Stop 1 w/o RTC ⇨ **7,5 µA @ 3.0V**
7.3 µA @ 1.8V

Zzz

Cortex M4

Flash (1MB)

SRAM 1 (96KB)

SRAM 2 (32KB)

Main regulator (MR)

Low Power regulator (LPR)

Backup domain

Backup Register (32x32-bits)

RTC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Wake-up event

- NRST
- BOR
- PVD
- PVM
- RTC + Tamper
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SWPMI
- COMP
- LPTIM 1
- LPTIM 2
- IWDG
- GPIOs

6us wake-up from Flash
4us wake-up from RAM

STM32L4 Power Mode

Stop 2 Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os kept, and configurable

Stop 2 w/ RTC on LSE quartz ⇨ **1.66 µA @3.0V**
1.43 µA @1.8V

Zzz

- Cortex M4
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)

Main regulator (MR)

Low Power regulator (LPR)

- Backup domain
  - Backup Register (32x32-bits)
  - RTC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Wake-up event

- NRST
- BOR
- PVD
- PVM
- RTC + Tamper
- LCD
- LP UART
- I2C 3
- COMP
- LPTIM 1
- IWDG
- GPIOs

7us wake-up from Flash
5us wake-up from RAM

STM32L4 Power Mode

Stop 2 Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os kept, and configurable

Stop 2 w/o RTC ⇨ **1.25 µA @3.0V**
1.19 µA @1.8V

Zzz

- Cortex M4
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)

- Main regulator (MR)
- Low Power regulator (LPR)
- Backup domain
  - Backup Register (32x32-bits)
  - RTC

Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

Wake-up event

- NRST
- BOR
- PVD
- PVM
- RTC + Tamper
- LCD
- LP UART
- I2C 3
- COMP
- LPTIM 1
- IWDG
- GPIOs

7us wake-up from Flash
5us wake-up from RAM

STM32L4 Power Mode

Standby Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- **BOR**
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- **IWDG**
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os can be configured
w/ or w/o pull-up
w/ or w/o pull-down

Standby w/ SRAM2 w/o RTC

405 nA @ 3.0V
363 nA @1.8V

Zzz

Cortex M4

Main regulator (MR)

Flash (1MB)

SRAM 1 (96KB)

SRAM 2 (32KB)

Low Power regulator (LPR)

Backup domain

Backup Register (32x32-bits)

RTC

Available Clock

- HSI
- HSE
- **LSI**
- **LSE**
- MSI

Wake-up event

- **NRST**
- **BOR**
- **RTC + Tamper**
- **IWDG**
- **5 WKUP pins**

14 us wake-up

STM32L4 Power Mode

Standby Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- **BOR**
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- **IWDG**
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os can be configured
w/ or w/o pull-up
w/ or w/o pull-down

Standby w/ RTC on LSE quartz

674 nA @ 3.0V
433 nA@ 1.8V

Zzz

Cortex M4

Main regulator (MR)

Flash (1MB)

SRAM 1 (96KB)

SRAM 2 (32KB)

Low Power regulator (LPR)

Backup domain

Backup Register (32x32-bits)

RTC

Wake-up event

- **NRST**
- **BOR**
- **RTC + Tamper**
- **IWDG**
- **5 WKUP pins**

14 us wake-up

Available Clock

- HSI
- HSE
- **LSI**
- **LSE**
- MSI

STM32L4 Power Mode

Standby Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- **BOR**
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- **IWDG**
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os can be configured
w/ or w/o pull-up
w/ or w/o pull-down

Standby ⇨ **169 nA @ 3.0V**
128 nA @ 1.8V

Zzz

- Cortex M4
- Main regulator (MR)
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)
- Low Power regulator (LPR)
- **Backup domain**
  - **Backup Register (32x32-bits)**
  - **RTC**

Available Clock

- HSI
- HSE
- **LSI**
- **LSE**
- MSI

Wake-up event

- **NRST**
- **BOR**
- **RTC + Tamper**
- **IWDG**
- **5 WKUP pins**

14 us wake-up

STM32L4 Power Mode

Shutdown Mode

Available Peripheral

- GPIO
- DMA
- FSMC
- QSPI
- BOR
- PVD, PVM
- LCD
- USB OTG
- USART
- LP UART
- I2C 1 / I2C 2
- I2C 3
- SPI
- CAN
- SDMMC
- SWPMI
- SAI
- DFSDM
- ADC
- DAC
- OPAMP
- COMP
- Temp Sensor
- Timers
- LPTIM 1
- LPTIM 2
- IWDG
- WWDG
- Systick Timer
- Touch Sens
- RNG
- AES
- CRC

I/Os can be configured
w/ or w/o pull-up
w/ or w/o pull-down
But floating when exit from Shutdown

Shutdown ⇨ **77 nA @ 3.0V**
43 nA @ 1.8V

Zzz

- Cortex M4
- Flash (1MB)
- SRAM 1 (96KB)
- SRAM 2 (32KB)

- Main regulator (MR)
- Low Power regulator (LPR)
- **Backup domain**
  - **Backup Register (32x32-bits)**
  - **RTC**

Available Clock

- HSI
- HSE
- LSI
- **LSE**
- MSI

Wake-up event

- **NRST**
- **RTC + Tamper**
- **5 WKUP pins**

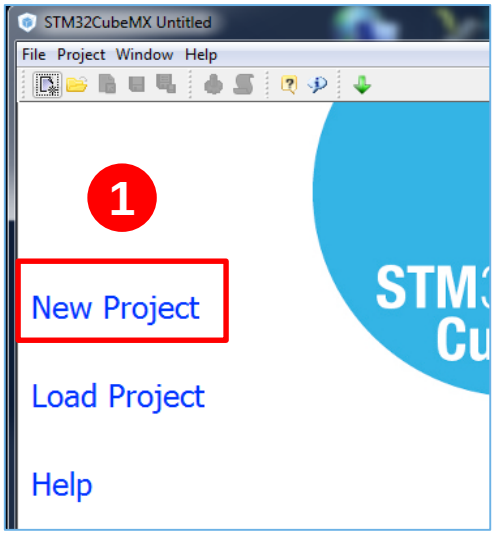
256 us wake-up

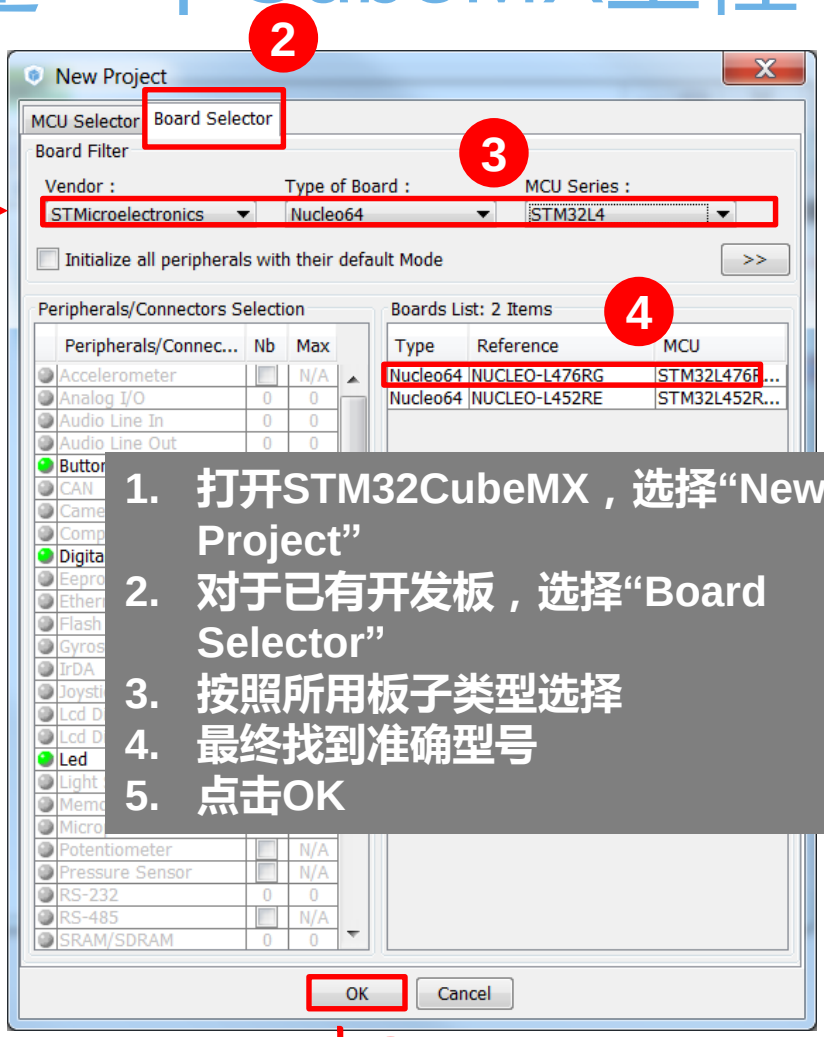
实验环节

-- ***任务1***: 使用STM32CubeMX 建立一个基于NUCLEO-L476RG 板子的功耗测试工程

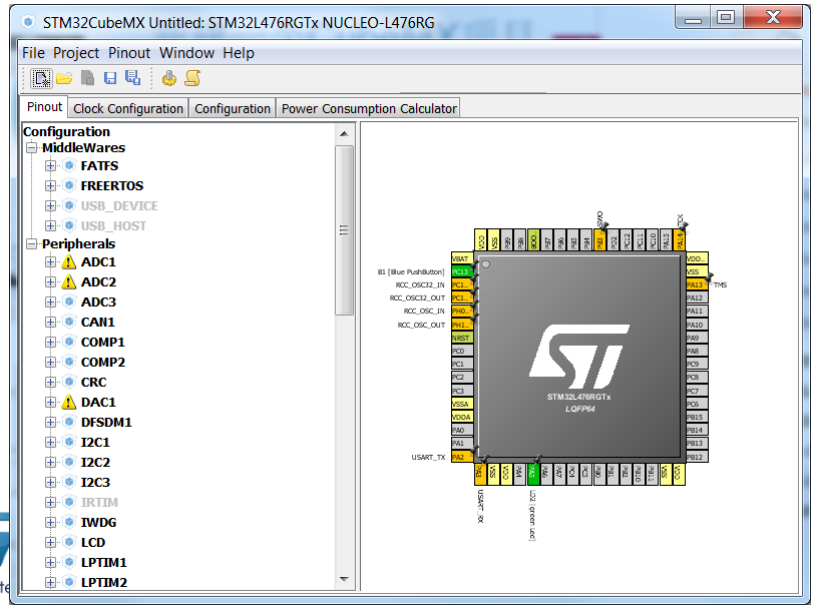
life.augmented

Step 1: 新建一个CubeMX工程



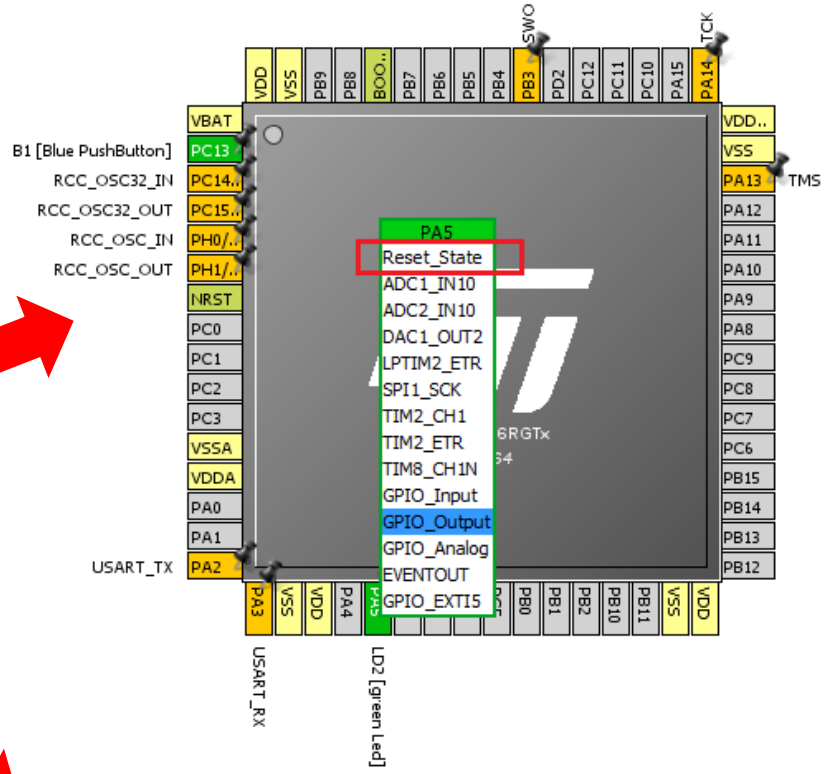


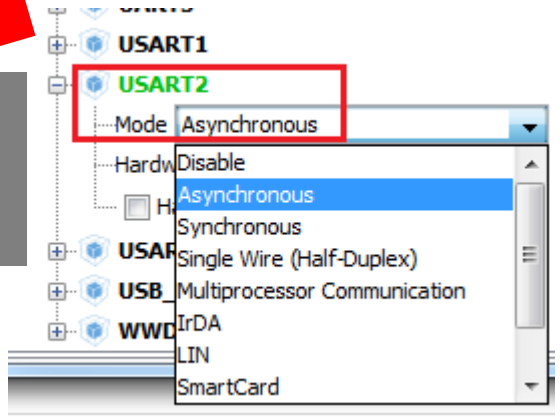
1. 打开STM32CubeMX，选择“New Project”
2. 对于已有开发板，选择“Board Selector”
3. 按照所用板子类型选择
4. 最终找到准确型号
5. 点击OK



Step 2:引脚配置

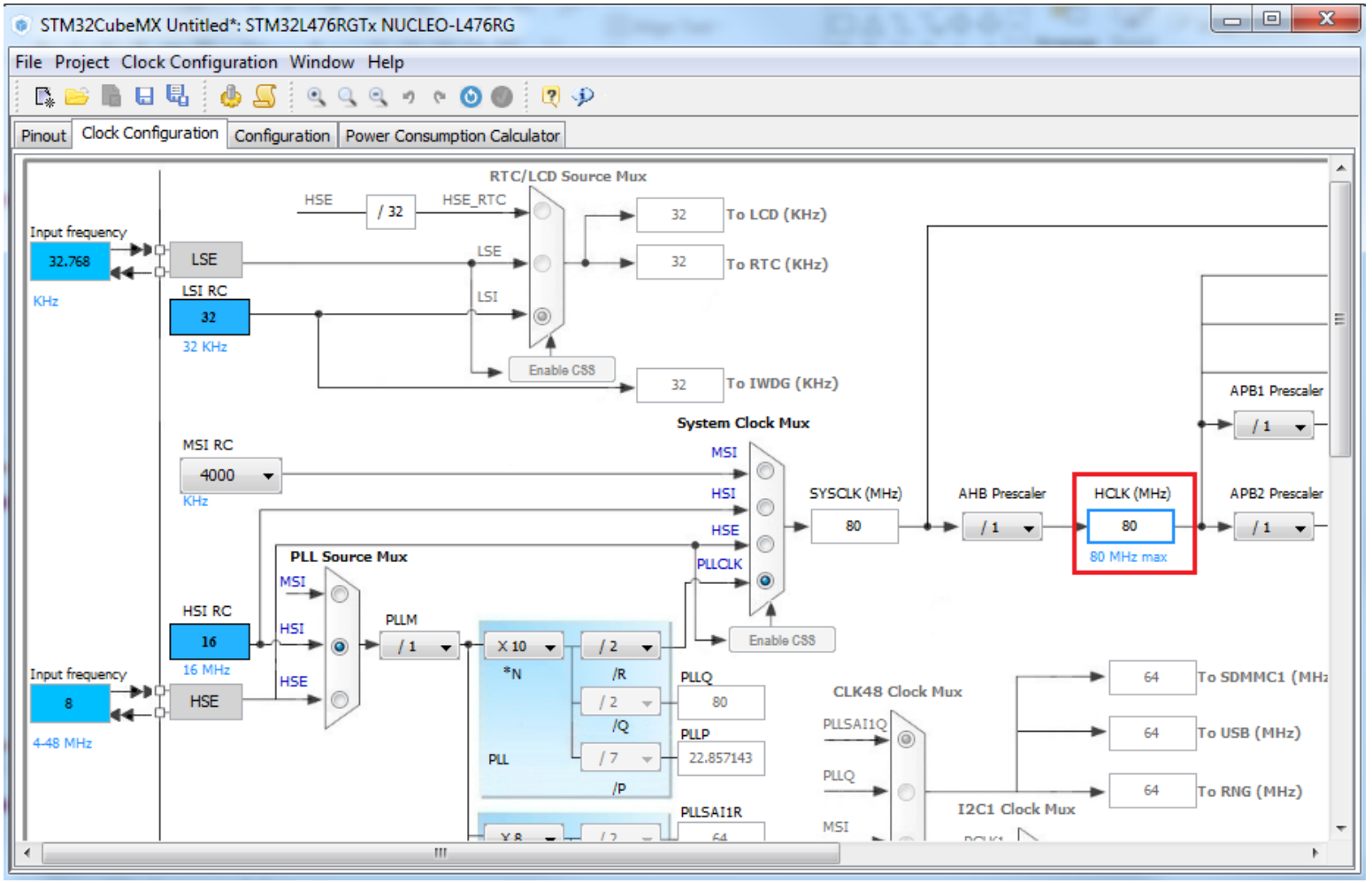
- **配置需要使用的外设引脚：**
  - **SYS**
  - **SWD Options**
  - **GPIO**
    - **PA5 Reset**
  - **USART**
    - **实现串口打印**





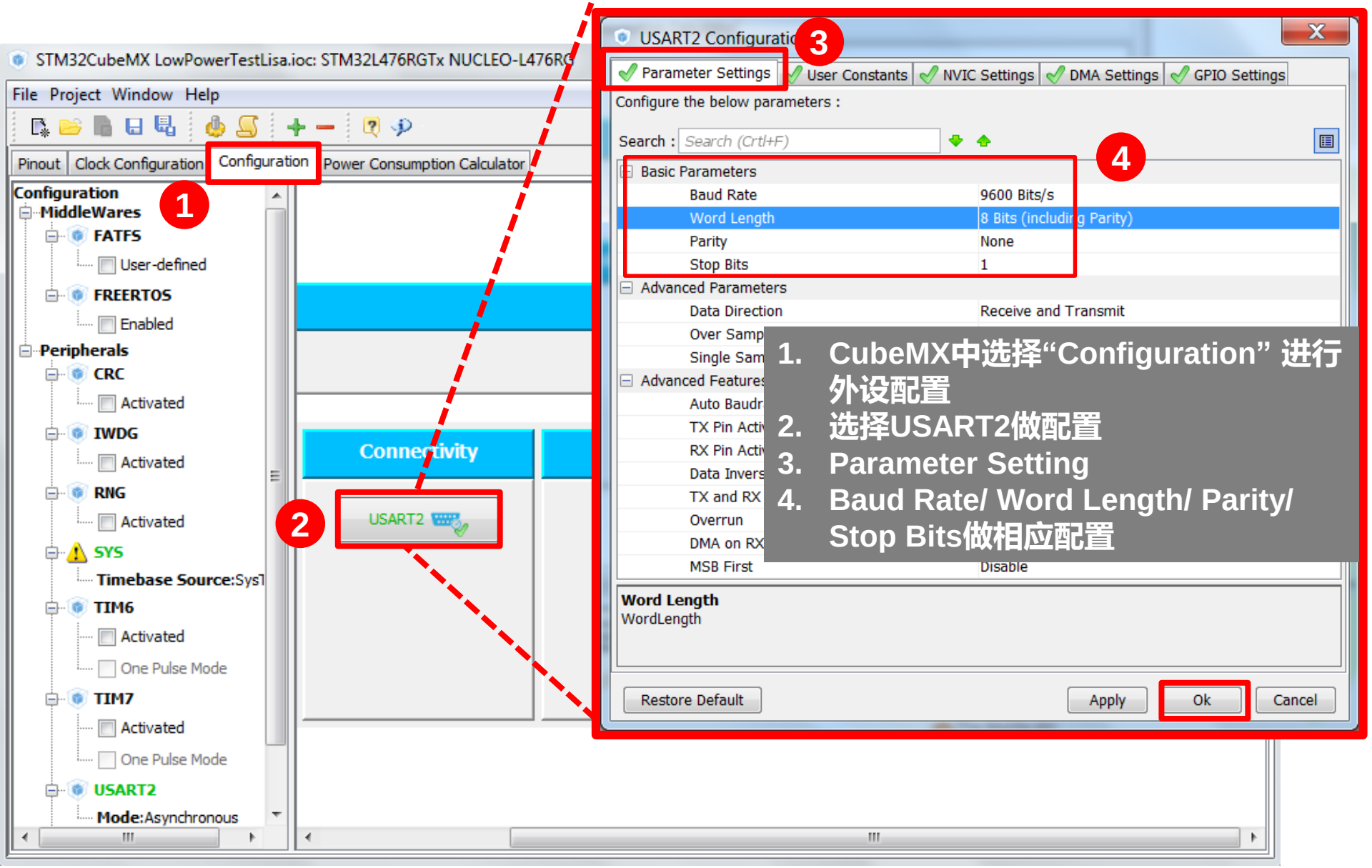
1. PA5选择“Reset_State”
2. 根据所用板子，选择实现串口打印的外设端口，配置成Asynchronous

Step 3:时钟配置



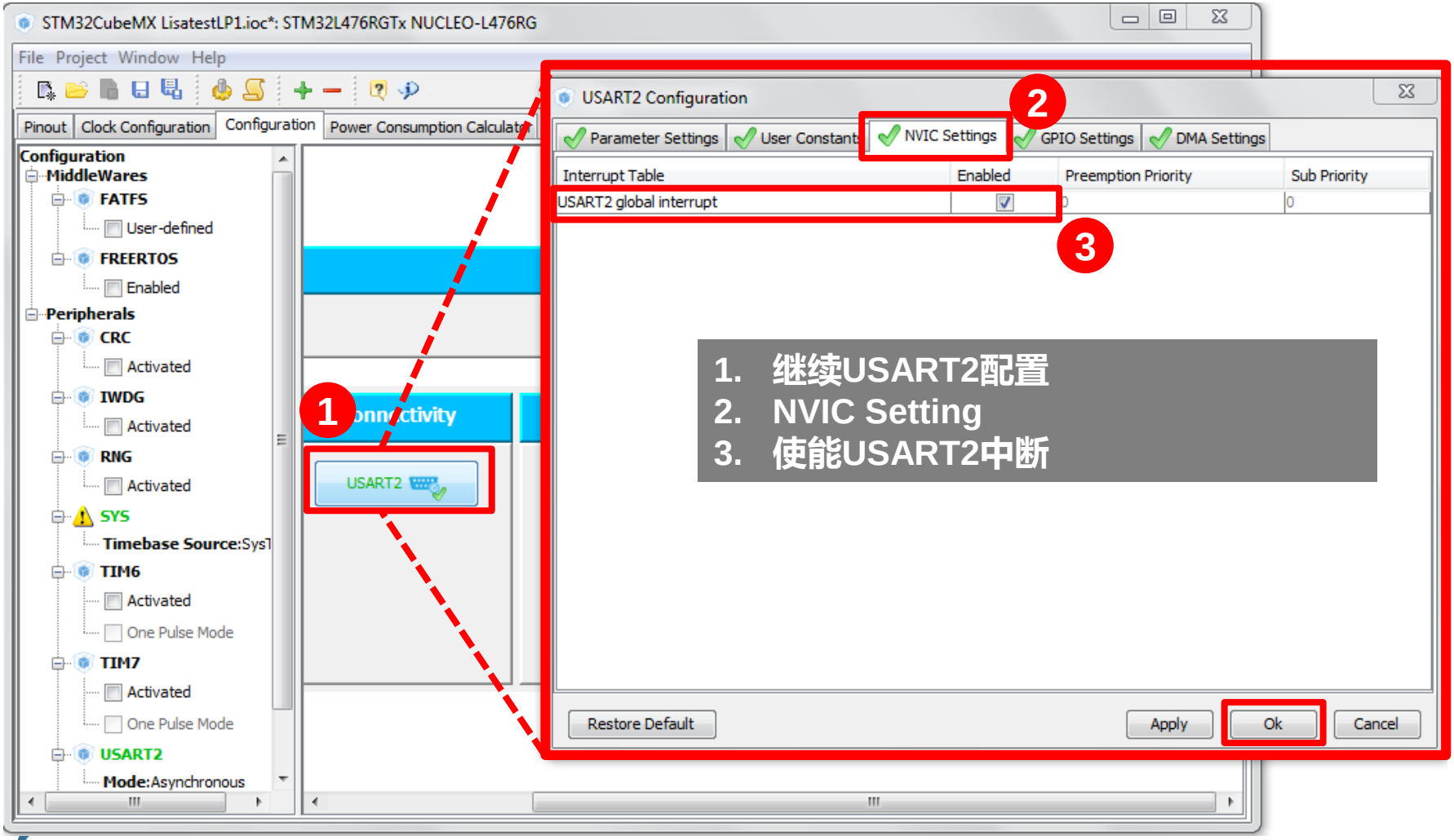
根据所用板子，HCLK和SYSCLK配置成最高频率80MHz

Step 4:外设配置

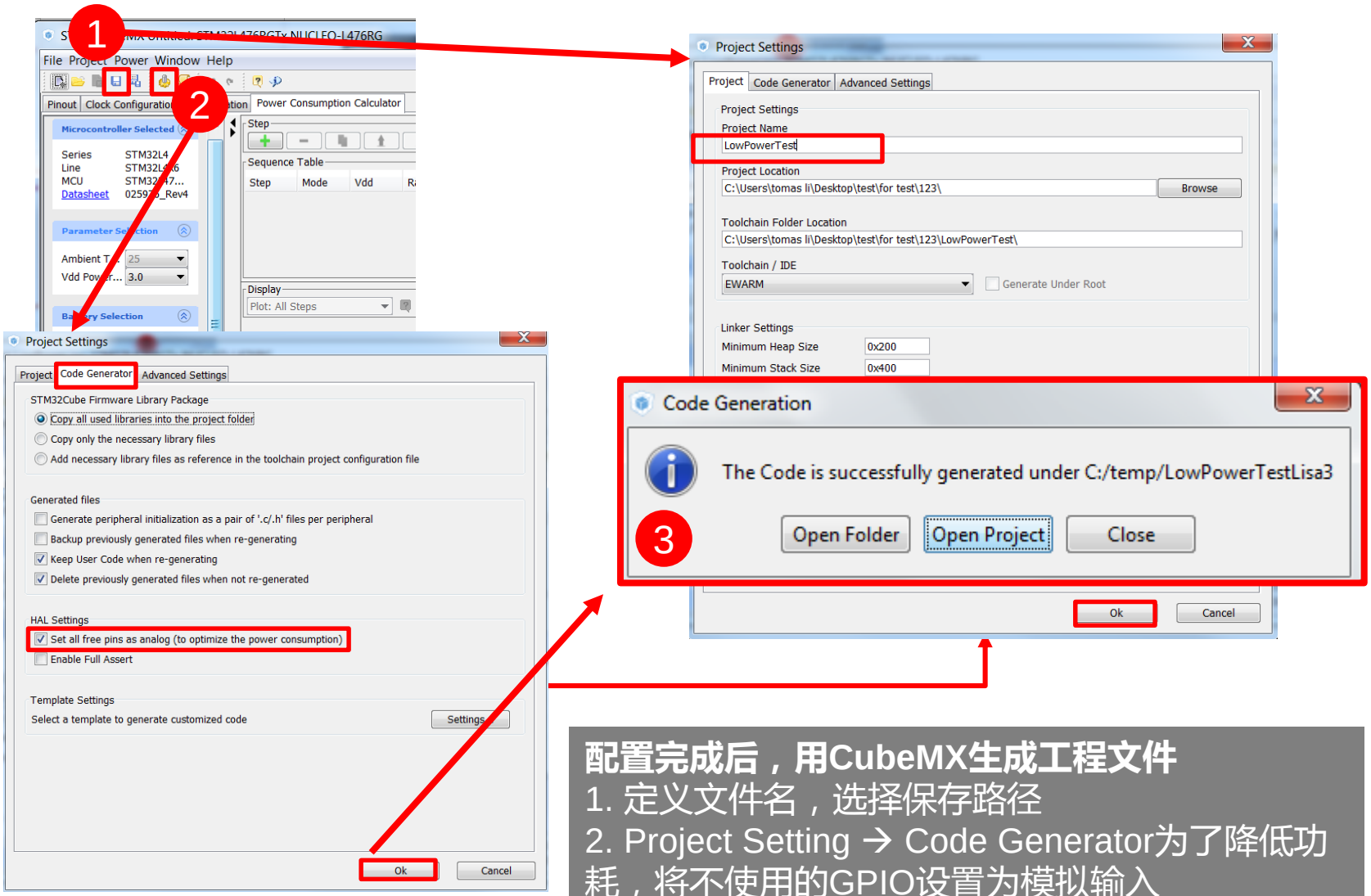


1. CubeMX中选择“Configuration” 进行外设配置
2. 选择USART2做配置
3. Parameter Setting
4. Baud Rate/ Word Length/ Parity/ Stop Bits做相应配置

Step 4:外设配置(续)

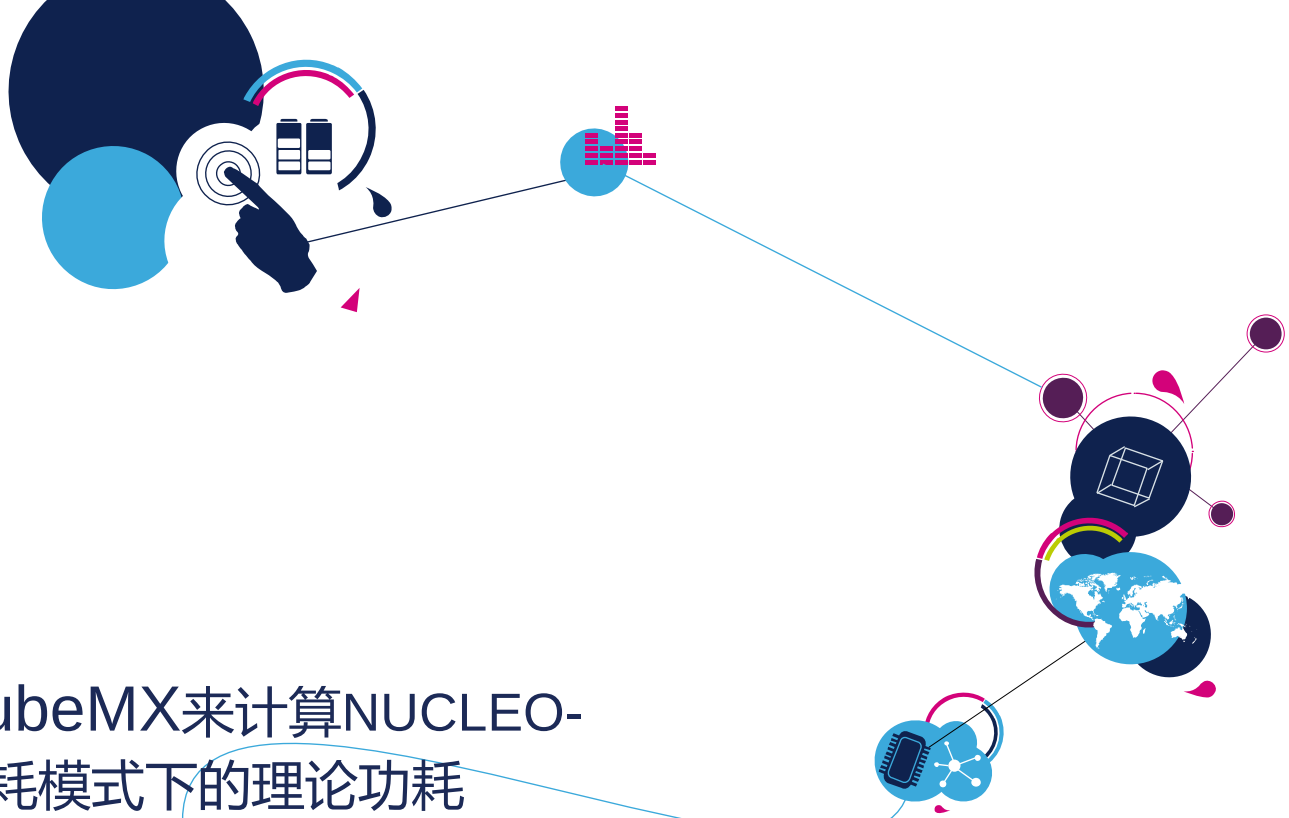

STM32CubeMX LisatestLP1.ioc*: STM32L476RGTx NUCLEO-L476RG
File Project Window Help
Pinout Clock Configuration Configuration Power Consumption Calculator
Configuration
MiddleWares
FATFS
User-defined
FREERTOS
Enabled
Peripherals
CRC
Activated
IWDG
Activated
RNG
Activated
SYS
Timebase Source:SysT
TIM6
Activated
One Pulse Mode
TIM7
Activated
One Pulse Mode
USART2
Mode:Asynchronous
USART2
1
USART2 Configuration
2
Parameter Settings
User Constants
NVIC Settings
GPIO Settings
DMA Settings
Interrupt Table
Enabled
Preemption Priority
Sub Priority
USART2 global interrupt
3
1. 继续USART2配置
2. NVIC Setting
3. 使能USART2中断
Restore Default
Apply
Ok
Cancel

Step5 生成源代码



配置完成后，用CubeMX生成工程文件

1. 定义文件名，选择保存路径
2. Project Setting → Code Generator为了降低功耗，将不使用的GPIO设置为模拟输入
3. 生成代码工程



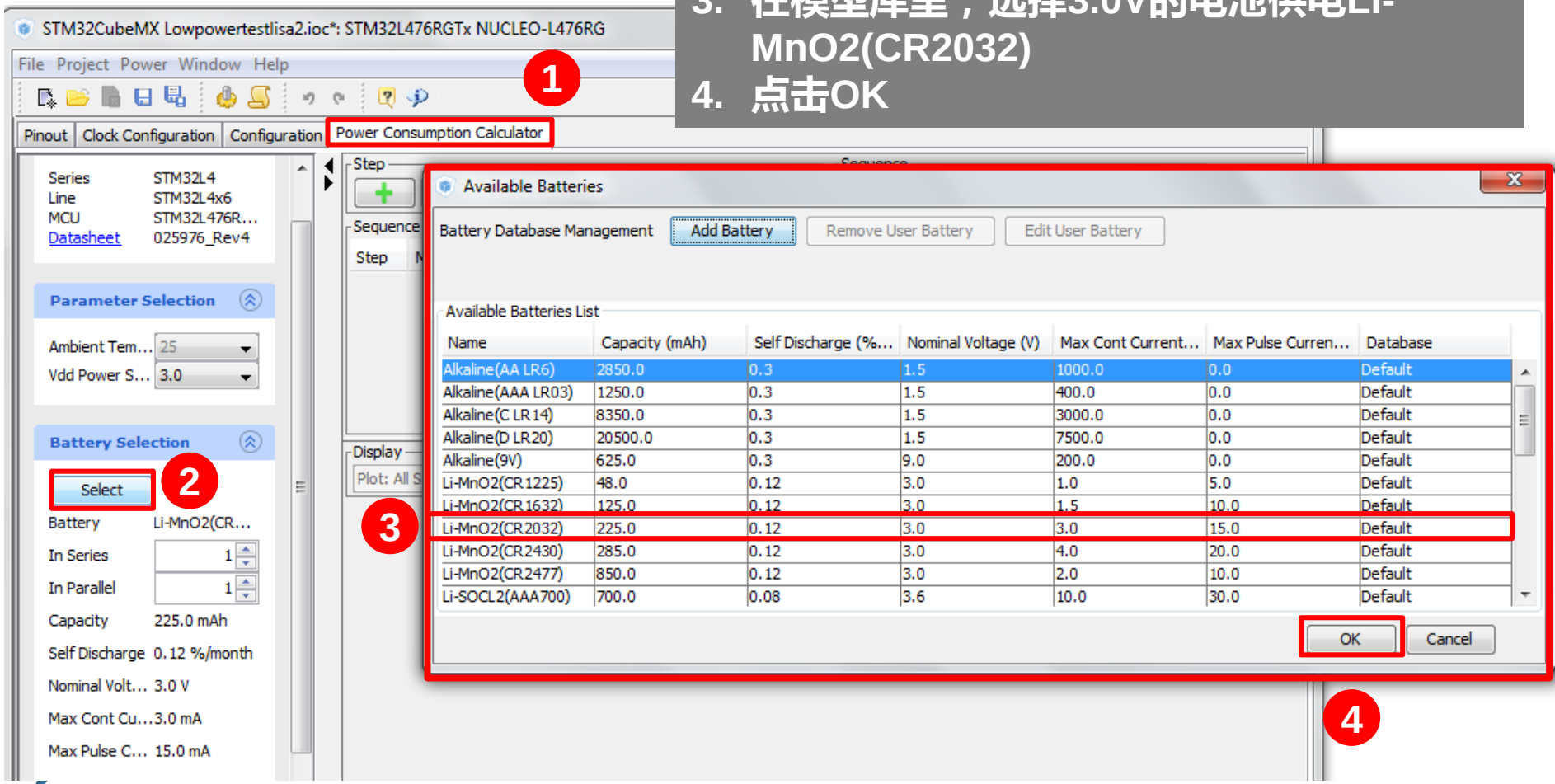
实验环节

--***任务2***：使用CubeMX来计算NUCLEO-L476RG在不同功耗模式下的理论功耗

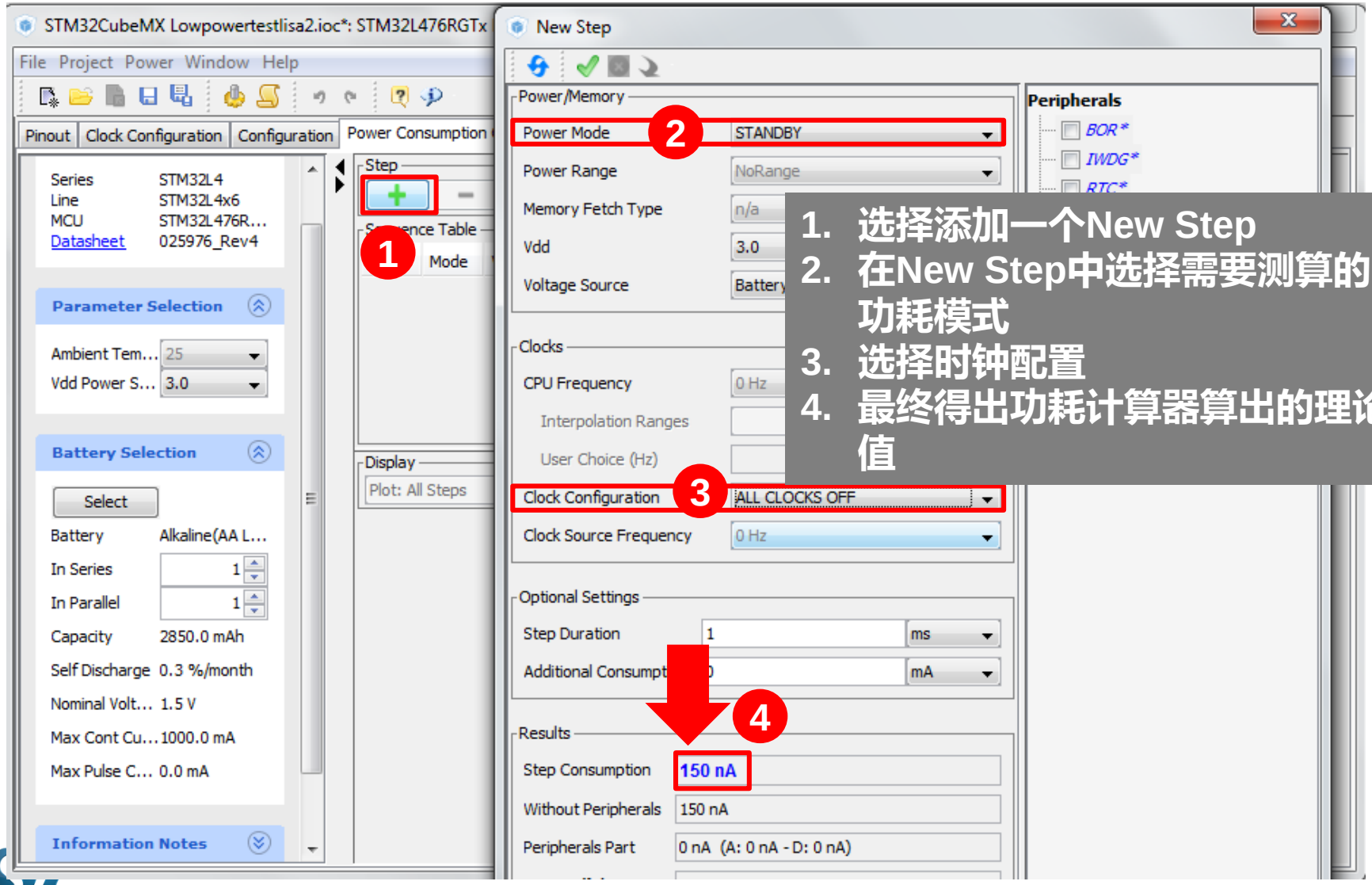
life.augmented

功耗计算器的使用

1. 选择“Power Consumption Calculator”
2. 选择合适的电池供电
3. 在模型库里，选择3.0V的电池供电Li-MnO2(CR2032)
4. 点击OK



功耗计算器的使用（续）



1. 选择添加一个New Step
2. 在New Step中选择需要测算的功耗模式
3. 选择时钟配置
4. 最终得出功耗计算器算出的理论值

理论结果

使用STM32CubeMX的Power Consumption Calculator得出的理论结果：

低功耗模式	理论计算值
SHUTDOWN	64nA
STANDBY	150nA
STOP2	1.18uA
LPSLEEP 2MHz	81uA

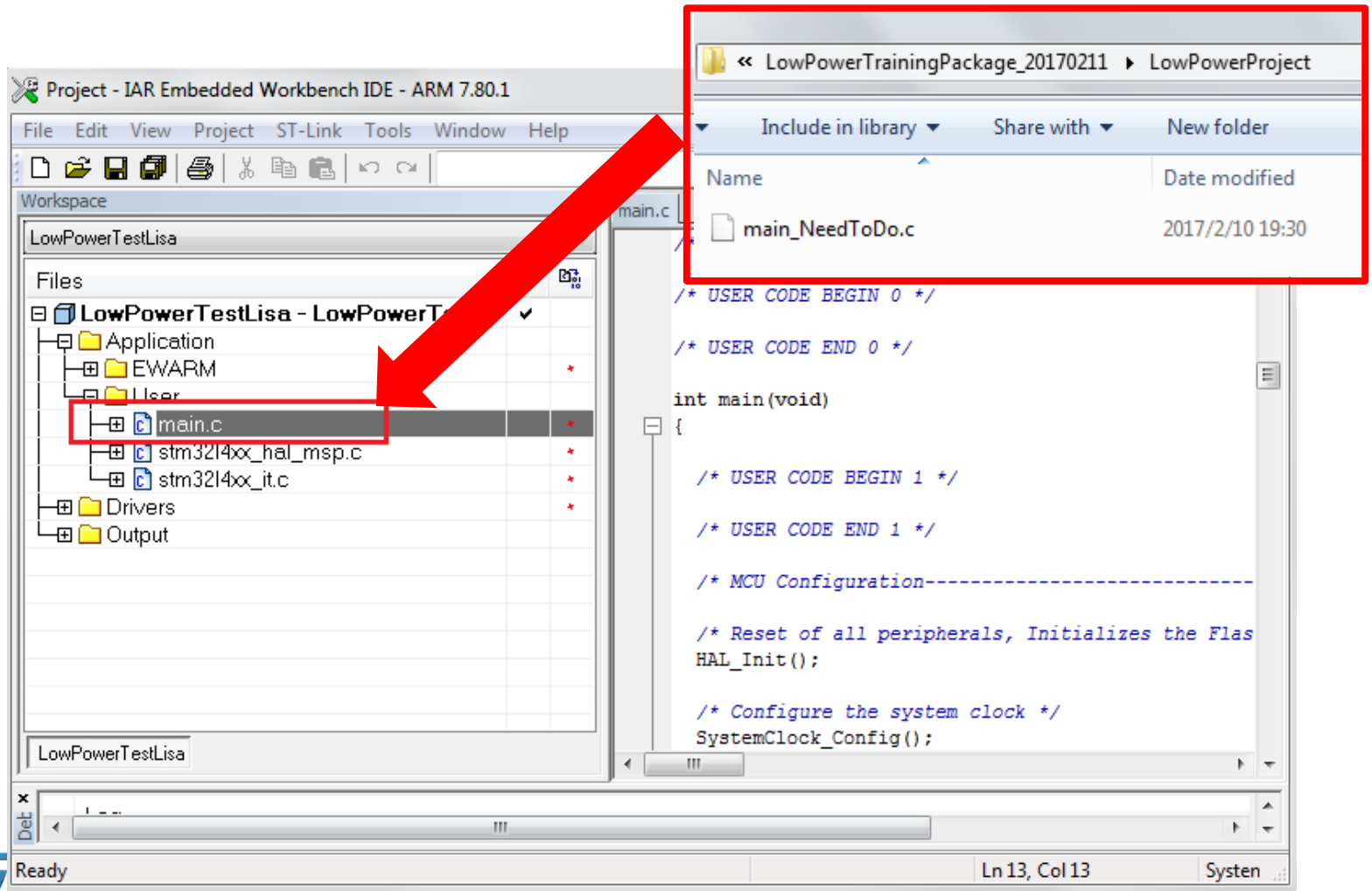
实验环节

--***任务3***: 实际测量STM32L476 板子在不同功耗模式下的功耗值，并且和使用STM32CubeMX理论推算的功耗值进行比较分析

life.augmented

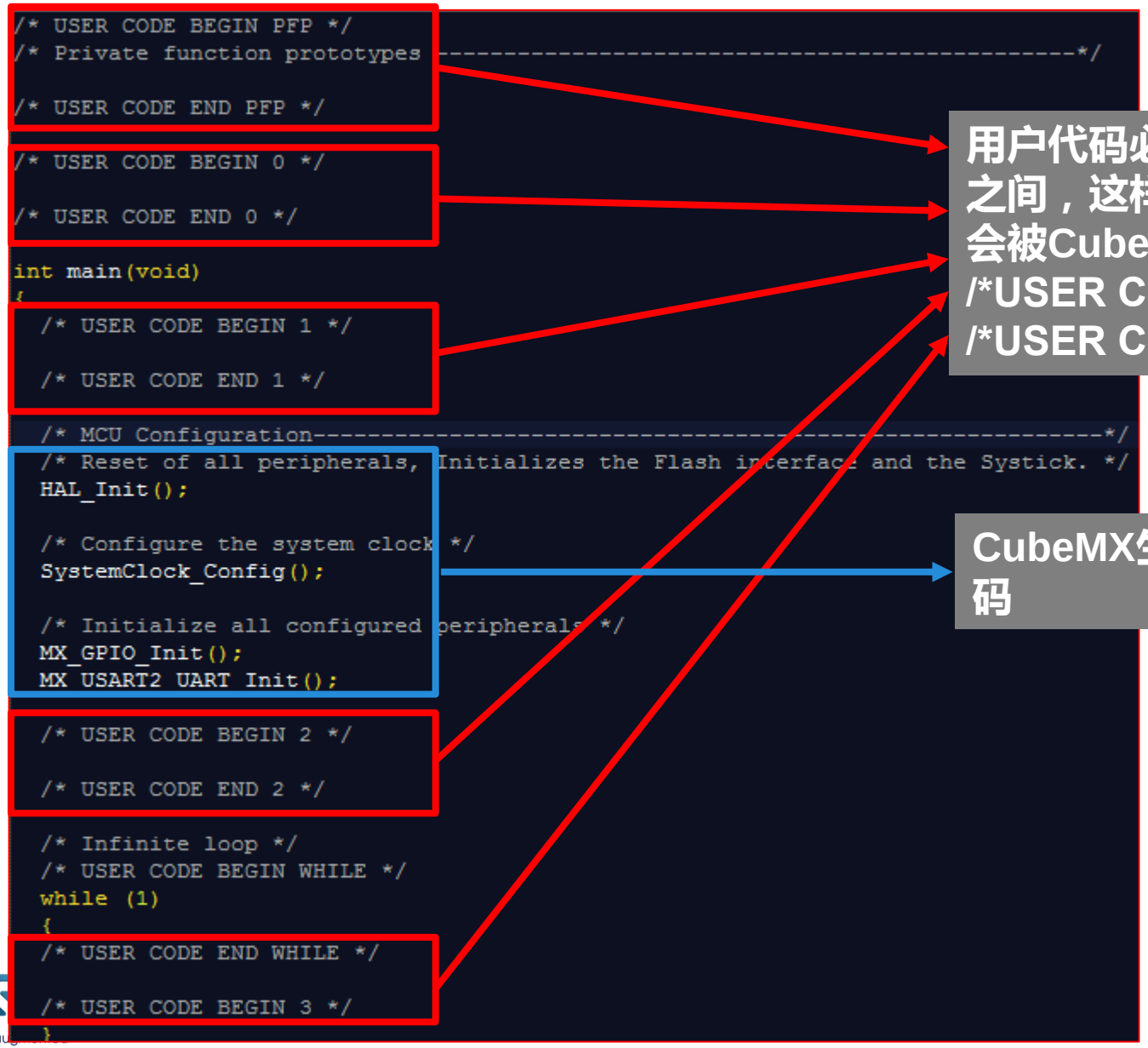
打开并替换main.c函数

用培训工程包里的 main_NeedToDo.c内容
替换目前main.c内容



life.augmented

STM32CubeMX USER CODE规则



用户代码必须放在如下的注释之间，这样加入的用户代码不会被CubeMX再生成代码改动：
/*USER CODE BEGIN…*/
/*USER CODE END…*/

CubeMX生成的MCU初始化代码

如何进入Shutdown模式


```
void test_shutdown(void)
{
  printf("\n\r Executing test (SHUTDOWN) \n\r");
  printf(" Please measure current then use Reset button to
select another test \n\r");

  /* Set all GPIO in analog state to reduce power
consumption */
  GPIO_AnalogState_Config();

  /* Enable Power Clock */
  __HAL_RCC_PWR_CLK_ENABLE();

  /* Enter the SHUTDOWN mode */
  HAL_PWREx_EnterSHUTDOWNMode();

}
```

Test_shutdown API可以直接拷贝放入main.c的USER CODE区域
/*USER CODE BEGIN…*/
/*USER CODE END…*/

Search:
// TODO 1: Enter into Shutdown mode

如何进入Standby模式


```
void test_standby(void)
  {
  printf("\n\r Executing test (STANDBY) \n\r");
  printf(" Please measure current then use Reset button to select
another test \n\r");

  /* Set all GPIO in analog state to reduce power consumption */
  GPIO_AnalogState_Config();

  /* Enable Power Clock */
  __HAL_RCC_PWR_CLK_ENABLE();

  /* Request to enter STANDBY mode */
  HAL_PWR_EnterSTANDBYMode();

}
```

Test_standby API可以直接拷贝放入main.c的USER CODE区域
/*USER CODE BEGIN…*/
/*USER CODE END…*/

Search:
// TODO 2: Enter into Standby mode

如何进入STOP模式

Test_stop API可以直接拷贝放入main.c的USER CODE区域
/*USER CODE BEGIN…*/
/*USER CODE END…*/

Search:
// TODO 3: Enter into Stop2 mode

```
void test_stop2(void)
{
  printf("\n\r Executing test (STOP2) \n\r");
  printf(" Please measure current then use Reset button to select another test \n\r");

  /* Set all GPIO in analog state to reduce power consumption */
  GPIO_AnalogState_Config();

  /* Enable Power Clock */
  __HAL_RCC_PWR_CLK_ENABLE();

  /* Enter STOP 2 mode */
  HAL_PWREx_EnterSTOP2Mode(PWR_STOPENTRY_WFE);
}
```

如何进入Low Power Sleep模式


```
void test_lpsleep_2mhz(void)
  {
  printf("\n\r Executing test (LPSLEEP 2MHz - FLASH OFF) \n\r");
  printf(" Please measure current then use Reset button to select another test \n\r");

  /* Set all GPIO in analog state to reduce power consumption */
  GPIO_AnalogState_Config();

  /* Enable Flash power down mode during Sleep mode     */
  /* (uncomment this line if power consumption figures  */
  /*  must be measured with Flash still on in Low Power */
  /*  Sleep mode)                                       */
  __HAL_FLASH_SLEEP_POWERDOWN_ENABLE();

  /* Set the System clock to 2 MHz (MSI) */
  SystemClock_2MHz();

  /* Suspend Tick increment to prevent wakeup by Systick interrupt.         */
  /* Otherwise the Systick interrupt will wake up the device within 1ms     */
  /* (HAL time base).                                                       */
  HAL_SuspendTick();

  /* Switch off all clock enable ... */
  RCC->AHB1SMENR = 0x0;
  RCC->AHB2SMENR = 0x0;
  RCC->AHB3SMENR = 0x0;
  RCC->APB1SMENR1 = 0x0;
  RCC->APB1SMENR2 = 0x0;
  RCC->APB2SMENR = 0x0;

  /* Enable Power Clock */
  __HAL_RCC_PWR_CLK_ENABLE();

  /* Enter SLEEP Mode, Main regulator is OFF */
  HAL_PWR_EnterSLEEPMode(PWR_LOWPOWERREGULATOR_ON, PWR_SLEEPENTRY_WFI);
}
```

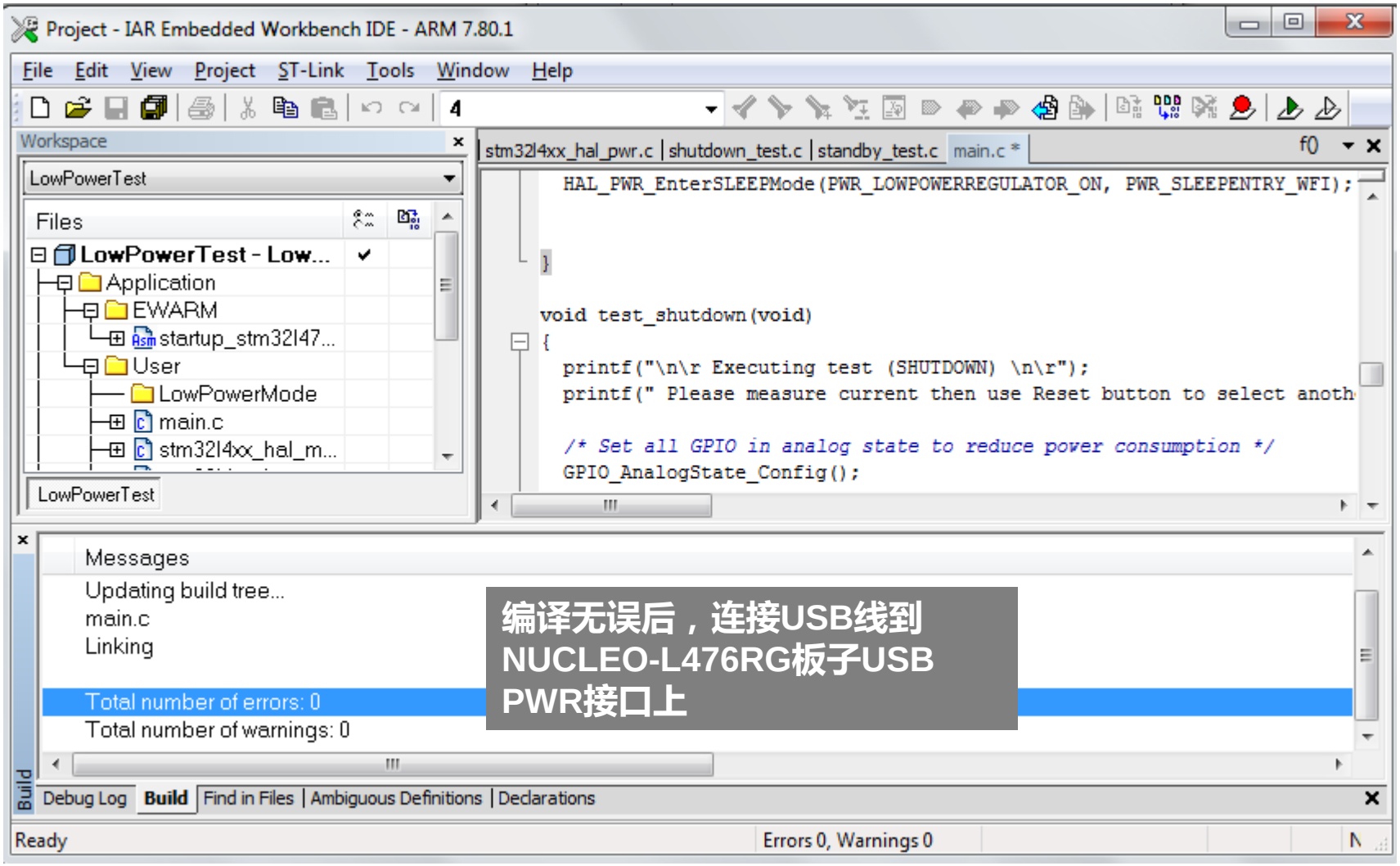
Test_lpsleep_2mhz API可以直接拷贝放入main.c的USER CODE区域
/*USER CODE BEGIN…*/
/*USER CODE END…*/

Search:
// TODO 4: Enter into Low Power Sleep mode

完成低功耗模式API调用

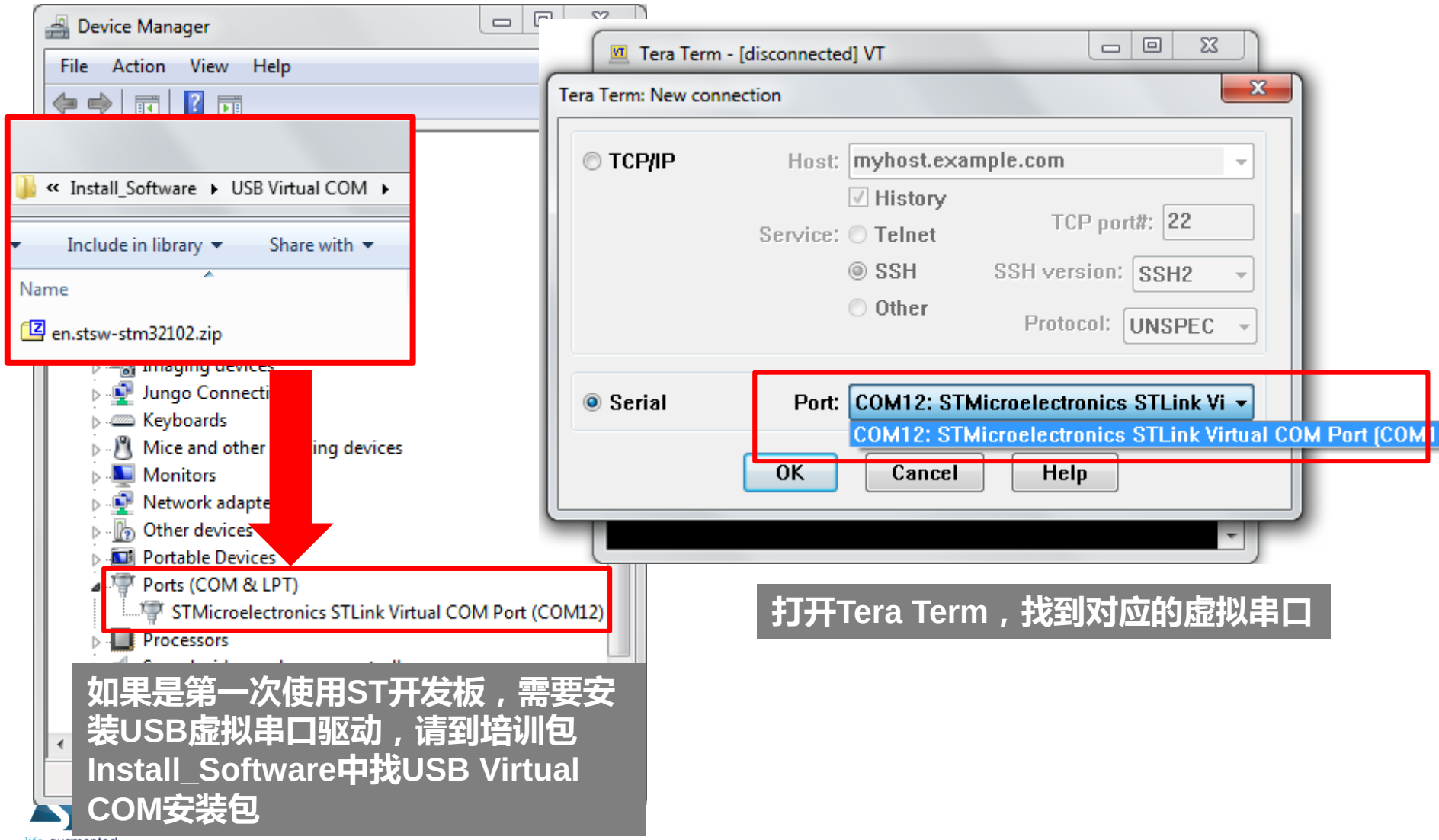

```
 // __EFF_NENW1NW2   __ATTRIBUTES int       strcmp(const
char *, const char *)
// strcmp(s, "x\r") ==0, input char is x\r, it is true

      // TODO 5: If receive char "0", enter into Shutdown mode


      // TODO 6: If receive char "1", enter into Standby mode


      // TODO 7: If receive char "2", enter into Stop2 mode


      // TODO 8: If receive char "3", enter into Low Power Sleep
```

编译下载



编译无误后，连接USB线到NUCLEO-L476RG板子USB PWR接口上

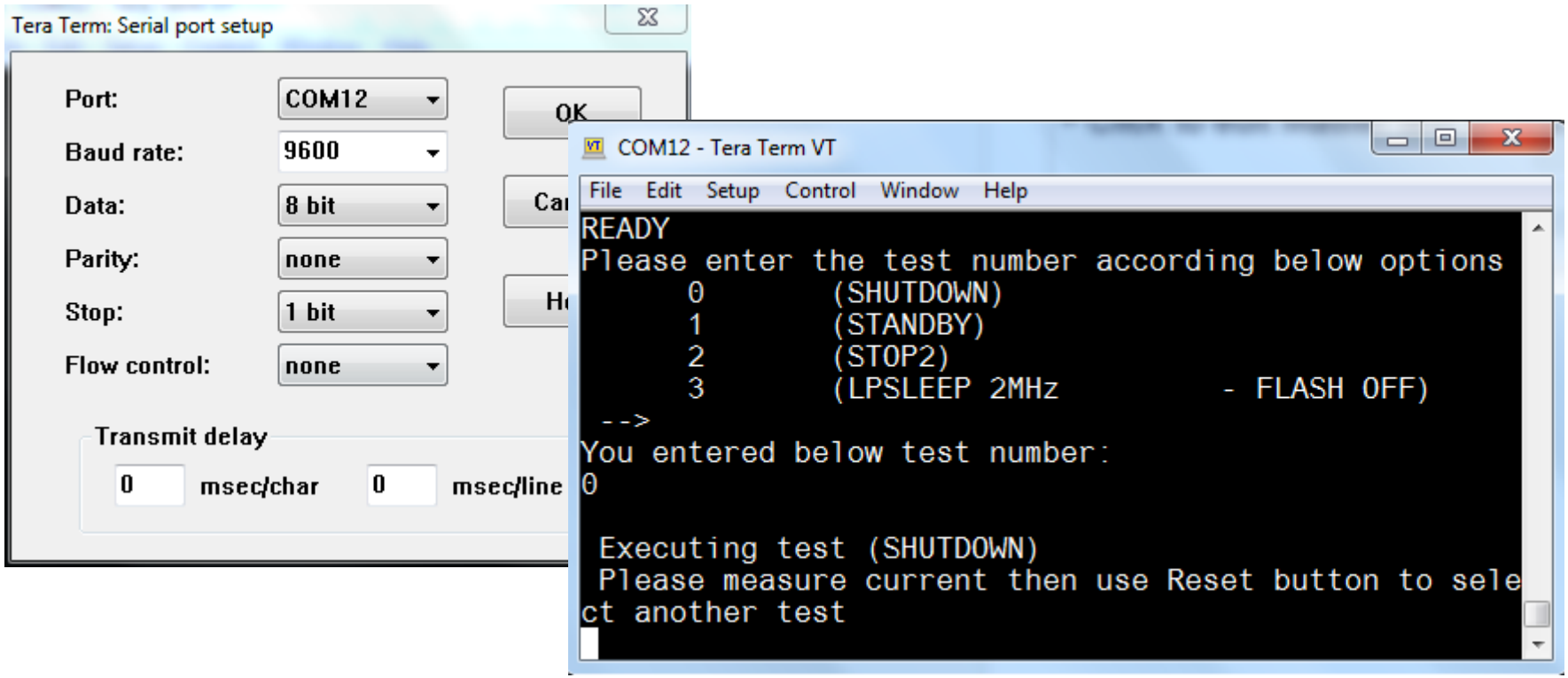
虚拟串口识别



如果是第一次使用ST开发板，需要安装USB虚拟串口驱动，请到培训包Install_Software中找USB Virtual COM安装包

打开Tera Term，找到对应的虚拟串口

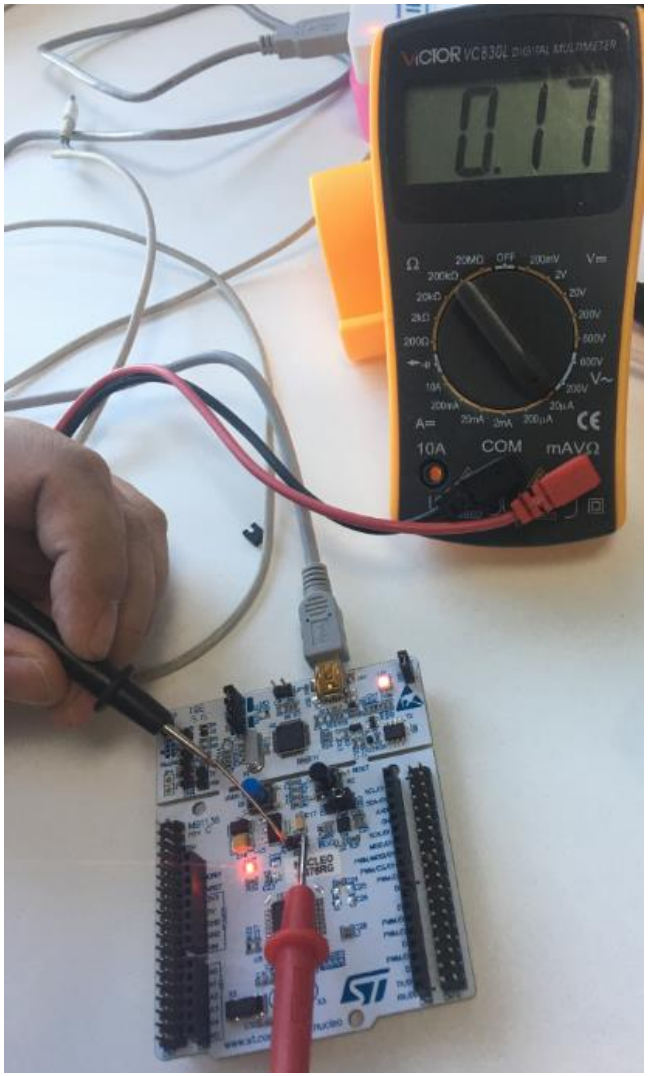
串口工具配置



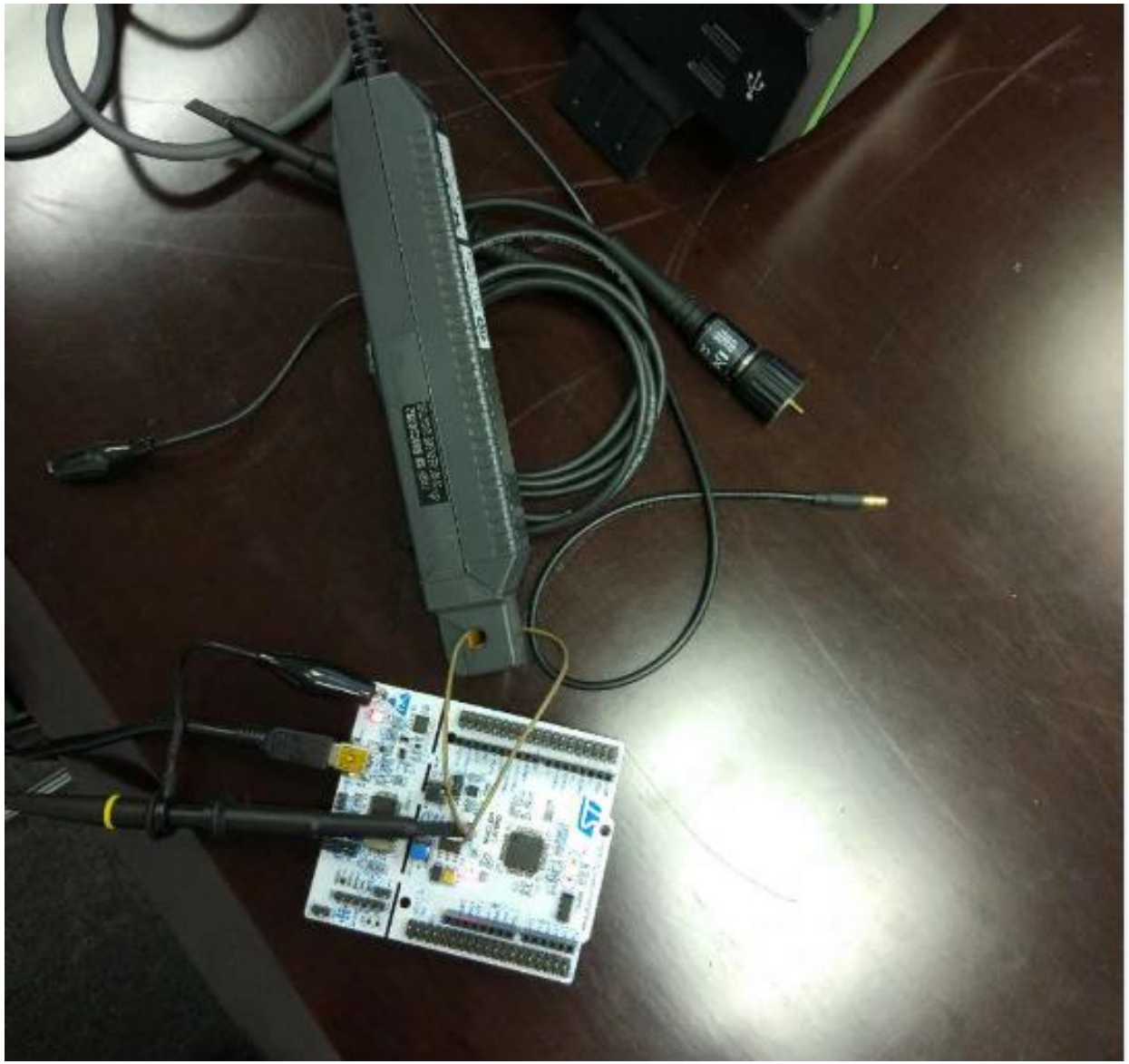
根据提示，输入不同的数字，板子进入相应的工作模式

实际测试电流

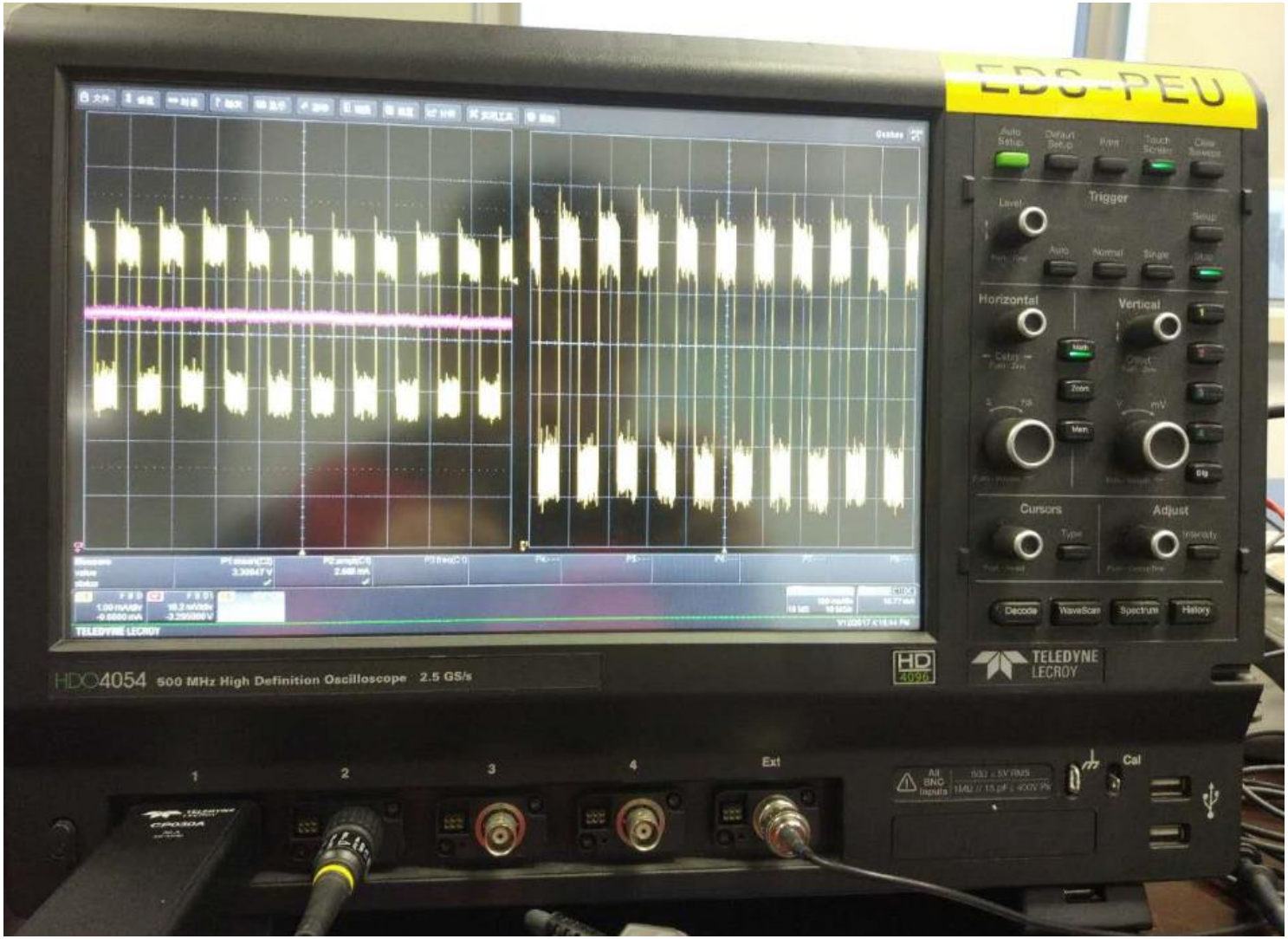
将电流表串在JP6的两端，实际测得电流



实际测量电流



实际测量电流


EDS-PEU
Trigger
Horizontal
Vertical
Cursors
Adjust
Decode
WaveScan
Spectrum
History
HDO4054 500 MHz High Definition Oscilloscope 2.5 GS/s
TELEDYNE LECROY
1
2
3
4
Ext
Cal

实践动手

- 测试一下其他低功耗模式下的电流，查看所得数据与数据手册上标称数据是否相符

- 依次测试STOP2模式、STANDBY模式、LPSLEEP模式和SHUTDOWN模式下的功耗, 填入下表，并且与自己在任务2中 功耗计算器获得的理论值做对比：

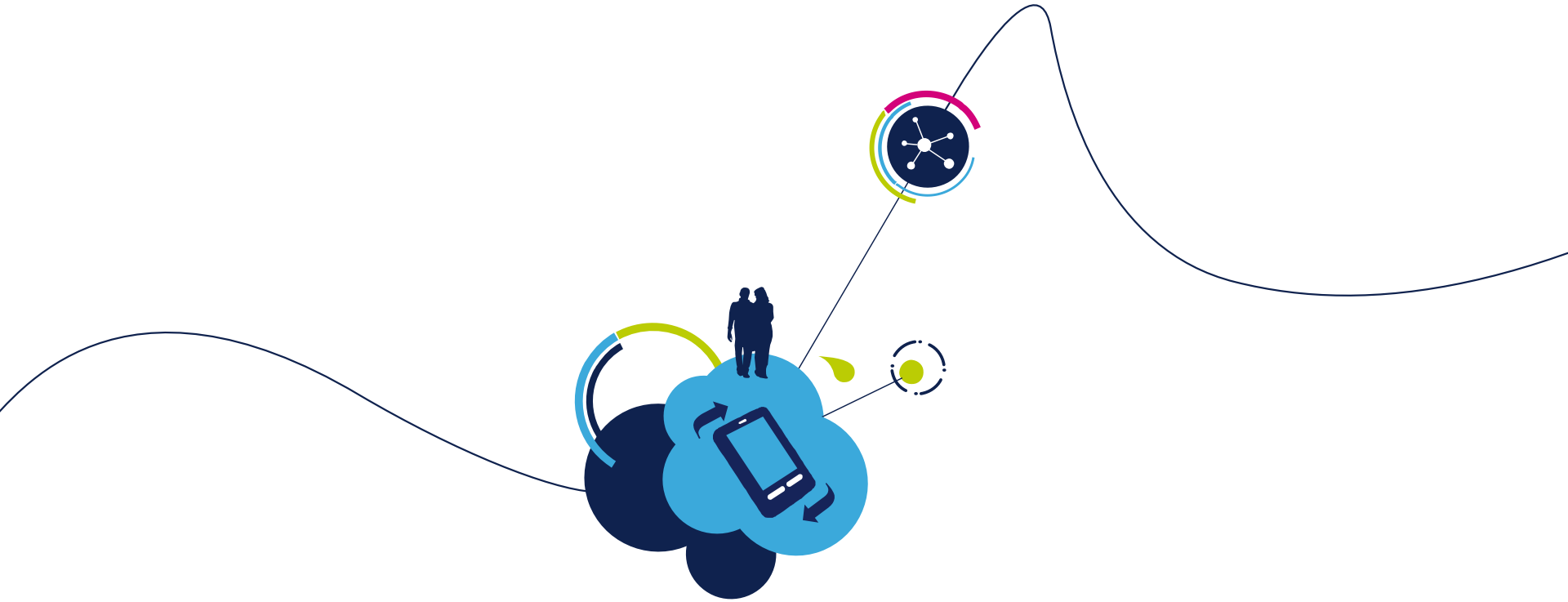
低功耗模式	实际测量值
SHUTDOWN	
STANDBY	
STOP2	
LPSLEEP 2MHz	

对比

低功耗模式	实际测量值	理论计算值
SHUTDOWN	75nA	64nA @3.0V
STANDBY	165nA	150nA@3.0V
STOP2	1.45uA	1.18uA@3.0V
LPSLEEP 2MHz	85uA	81uA@3.0V

总结

- 今天的动手实验中，我们学习和理解了STML476的低功耗特性和不同低功耗模式。
  - STML476的低功耗优化特性有了清晰的认识
  - 如何使用STM32CubeMX从头开始实现一个低功耗工程
  - 如何使用功耗计算器去理论分析不同功耗模式下的功耗值
  - 以及如何在NUCLEO板子上实际测得不同功耗值



谢谢

www.stmcu.com.cn

life.augmented