c) Insert user code in the main function, after initialization calls and before the while loop, to perform actual read/write from/to the SD card:

```
int main(void)
{

      ….
      MX_FATFS_Init();

        /* USER CODE BEGIN 2 */
      /*##-0- Turn all LEDs off(red, green, orange and blue) */
         HAL_GPIO_WritePin(GPIOG, (GPIO_PIN_10 | GPIO_PIN_6 | GPIO_PIN_7 |
      GPIO_PIN_12), GPIO_PIN_SET);
      /*##-1- FatFS: Link the SD disk I/O driver ##########*/
          if(retSD == 0){
             /* success: set the orange LED on */
            HAL_GPIO_WritePin(GPIOG, GPIO_PIN_7, GPIO_PIN_RESET);
      /*##-2- Register the file system object to the FatFs module ###*/
          if(f_mount(&SDFatFs, (TCHAR const*)SD_Path, 0) != FR_OK){
           /* FatFs Initialization Error : set the red LED on */
              HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
              while(1);
            }   else    {
      /*##-3- Create a FAT file system (format) on the logical drive#*/
       /* WARNING: Formatting the uSD card will delete all content on the
      device */
        if(f_mkfs((TCHAR const*)SD_Path, 0, 0) != FR_OK){
          /* FatFs Format Error : set the red LED on */
          HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
          while(1);
        } else {
      /*##-4- Create & Open a new text file object with write access#*/
        if(f_open(&MyFile, "Hello.txt", FA_CREATE_ALWAYS | FA_WRITE) !=
      FR_OK){
        /* 'Hello.txt' file Open for write Error : set the red LED on */
        HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
        while(1);
            } else {
       /*##-5- Write data to the text file ####################*/
         res = f_write(&MyFile, wtext, sizeof(wtext), (void
      *)&byteswritten);
        if((byteswritten == 0) || (res != FR_OK)){
          /* 'Hello.txt' file Write or EOF Error : set the red LED on */
          HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
          while(1);
          } else {
       /*##-6- Successful open/write : set the blue LED on */
          HAL_GPIO_WritePin(GPIOG, GPIO_PIN_12, GPIO_PIN_RESET);
          f_close(&MyFile);
       /*##-7- Open the text file object with read access #*/
          if(f_open(&MyFile, "Hello.txt", FA_READ) != FR_OK){
         /* 'Hello.txt' file Open for read Error : set the red LED on */
```

```
                HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
                while(1);
                } else {
             /*##-8- Read data from the text file #########*/
                res = f_read(&MyFile, rtext, sizeof(wtext), &bytesread);
                if((strcmp(rtext,wtext)!=0)|| (res != FR_OK)){
              /* 'Hello.txt' file Read or EOF Error : set the red LED on */
                HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET);
                while(1);
                } else {
              /* Successful read : set the green LED On */
                HAL_GPIO_WritePin(GPIOG, GPIO_PIN_6, GPIO_PIN_RESET);
             /*##-9- Close the open text file #################*/
                f_close(&MyFile);
              }}}}}}}
             /*##-10- Unlink the micro SD disk I/O driver #########*/
                FATFS_UnLinkDriver(SD_Path);

              /* USER CODE END 2 */

              /* Infinite loop */
              /* USER CODE BEGIN WHILE */
        while (1)
```

8 Tutorial 3- Using PCC to optimize the embedded application power consumption and more

8.1 Tutorial overview

This tutorial focuses on STM32CubeMX Power Consumption Calculator (PCC) feature and its benefits to evaluate the impacts of power-saving techniques on a given application sequence.

The key considerations to reduce a given application power consumption are:

- Reducing the operating voltage
- Reducing the time spent in energy consuming modes

  It is up to the developer to select a configuration that will give the best compromise between low-power consumption and performance.
- Maximizing the time spent in non-active and low-power modes
- Using the optimal clock configuration

  The core should always operate at relatively good speed, since reducing the operating frequency can increase energy consumption if the microcontroller has to remain for a long time in an active operating mode to perform a given operation.
- Enabling only the peripherals relevant for the current application state and clock-gating the others
- When relevant, using the peripherals with low-power features (e.g. waking up the microcontroller with the I2C)
- Minimizing the number of state transitions
- Optimizing memory accesses during code execution
  - Prefer code execution from RAM to Flash memory
  - When relevant, consider aligning CPU frequency with Flash memory operating frequency for zero wait states.

The following tutorial will show how STM32CubeMX PCC feature can help to tune an application to minimize its power consumption and extend the battery life.

Note: *PCC does not account for I/O dynamic current consumption and external board components that can also affect current consumption. For this purpose, an "additional consumption" field is provided for the user to specify such consumption value.*

8.2 Application example description

The application is designed using the NUCLEO-L476RG board based on a STM32L476RGTx device and supplied by a 2.4 V battery.

The main purpose of this application is to perform ADC measurements and transfer the conversion results over UART. It uses:

- Multiple low-power modes: Low-power run, Low-power sleep, Sleep, Stop and Standby
- Multiple peripherals: USART, DMA, Timer, COMP, DAC and RTC
  - The RTC is used to run a calendar and to wake up the CPU from Standby when a specified time has elapsed.
  - The DMA transfers ADC measurements from ADC to memory
  - The USART is used in conjunction with the DMA to send/receive data via the virtual COM port and to wake up the CPU from Stop mode.

The process to optimize such complex application is to start describing first a functional only sequence then to introduce, on a step by step basis, the low-power features provided by the STM32L476RG microcontroller.

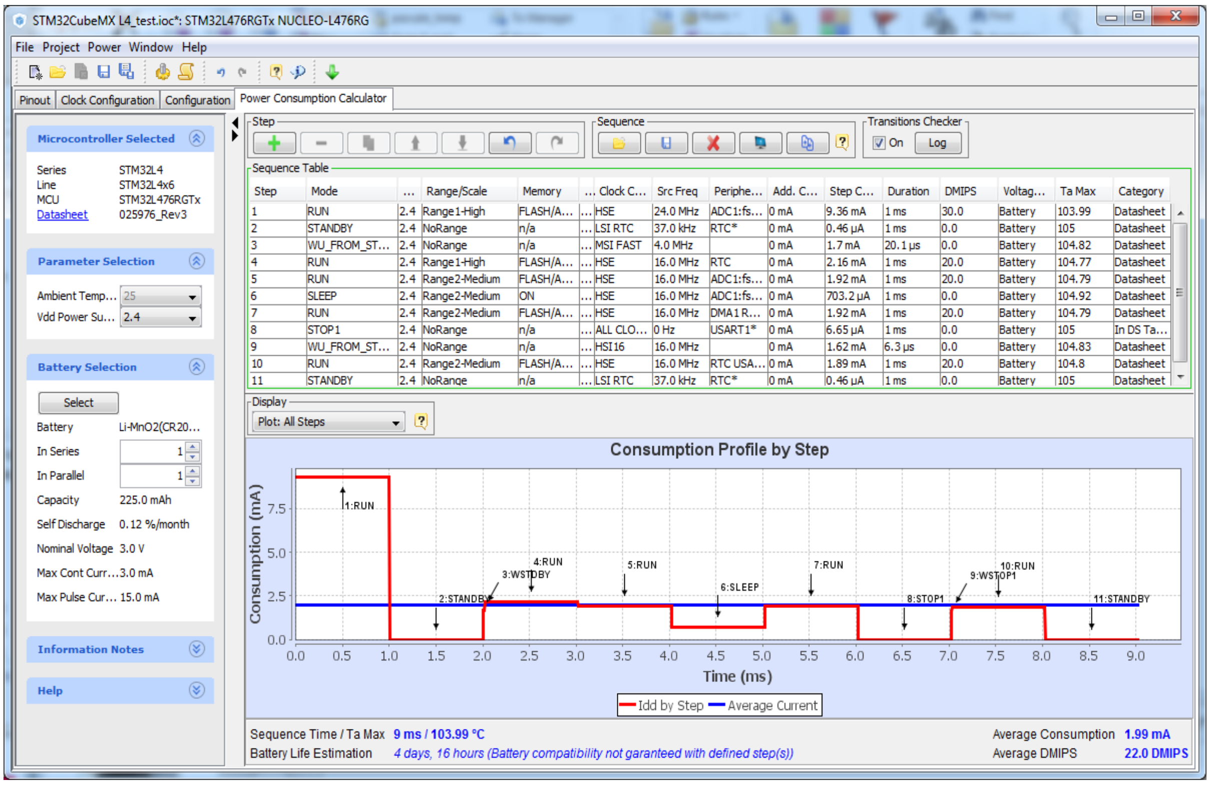
8.3 Using the Power Consumption Calculator

8.3.1 Creating a PCC sequence

Follow the steps below to open PCC and create the sequence (see *Figure 173*):

1. Launch STM32CubeMX.
2. Click **new project** and select the Nucleo-L476RG board from the **Board** tab.
3. Click the **Power Consumption Calculator** tab to select the Power Consumption Calculator view. A first sequence is then created as a reference.
4. Adapt it to minimize the overall current consumption. To do this:
   a) Select 2.4 V V_{DD} power supply. This value can be adjusted on a step by step basis (see *Figure 174*).
   b) Select the Li-MnO2 (CR2032) battery. This step is optional. The battery type can be changed later on (see *Figure 174*).

Figure 173. Power Consumption Calculation example



Microcontroller Selected

Series	STM32L4
Line	STM32L4x6
MCU	STM32L476RGTx
Datasheet	025976_Rev3

Parameter Selection

Ambient Temp...	25
Vdd Power Su...	2.4

Battery Selection

Select

Battery	Li-MnO2(CR20...
In Series	1
In Parallel	1
Capacity	225.0 mAh
Self Discharge	0.12 %/month
Nominal Voltage	3.0 V
Max Cont Curr...	3.0 mA
Max Pulse Cur...	15.0 mA

Information Notes

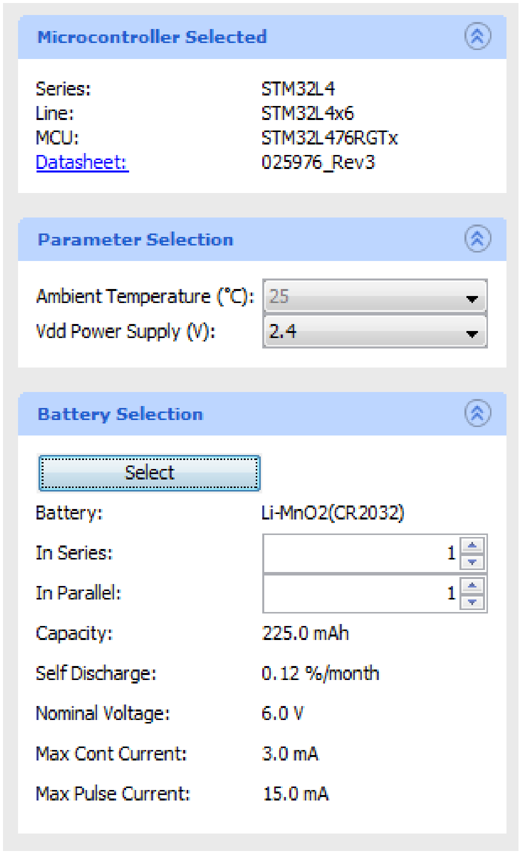
Help

Sequence Table

Step	Mode	...	Range/Scale	Memory	...	Clock C...	Src Freq	Periphe...	Add. C...	Step C...	Duration	DMIPS	Voltag...	Ta Max	Category
1	RUN	2.4	Range1-High	FLASH/A...	...	HSE	24.0 MHz	ADC1:fs...	0 mA	9.36 mA	1 ms	30.0	Battery	103.99	Datasheet
2	STANDBY	2.4	NoRange	n/a	...	LSI RTC	37.0 kHz	RTC*	0 mA	0.46 µA	1 ms	0.0	Battery	105	Datasheet
3	WU_FROM_ST...	2.4	NoRange	n/a	...	MSI FAST	4.0 MHz		0 mA	1.7 mA	20.1 µs	0.0	Battery	104.82	Datasheet
4	RUN	2.4	Range1-High	FLASH/A...	...	HSE	16.0 MHz	RTC	0 mA	2.16 mA	1 ms	20.0	Battery	104.77	Datasheet
5	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	ADC1:fs...	0 mA	1.92 mA	1 ms	20.0	Battery	104.79	Datasheet
6	SLEEP	2.4	Range2-Medium	ON	...	HSE	16.0 MHz	ADC1:fs...	0 mA	703.2 µA	1 ms	0.0	Battery	104.92	Datasheet
7	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	DMA1 R...	0 mA	1.92 mA	1 ms	20.0	Battery	104.79	Datasheet
8	STOP1	2.4	NoRange	n/a	...	ALL CLO...	0 Hz	USART1*	0 mA	6.65 µA	1 ms	0.0	Battery	105	In DS Ta...
9	WU_FROM_ST...	2.4	NoRange	n/a	...	HSI16	16.0 MHz		0 mA	1.62 mA	6.3 µs	0.0	Battery	104.83	Datasheet
10	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	RTC USA...	0 mA	1.89 mA	1 ms	20.0	Battery	104.8	Datasheet
11	STANDBY	2.4	NoRange	n/a	...	LSI RTC	37.0 kHz	RTC*	0 mA	0.46 µA	1 ms	0.0	Battery	105	Datasheet

Sequence Time / Ta Max	9 ms / 103.99 °C	Average Consumption	1.99 mA
Battery Life Estimation	*4 days, 16 hours (Battery compatibility not garanteed with defined step(s))*	Average DMIPS	22.0 DMIPS

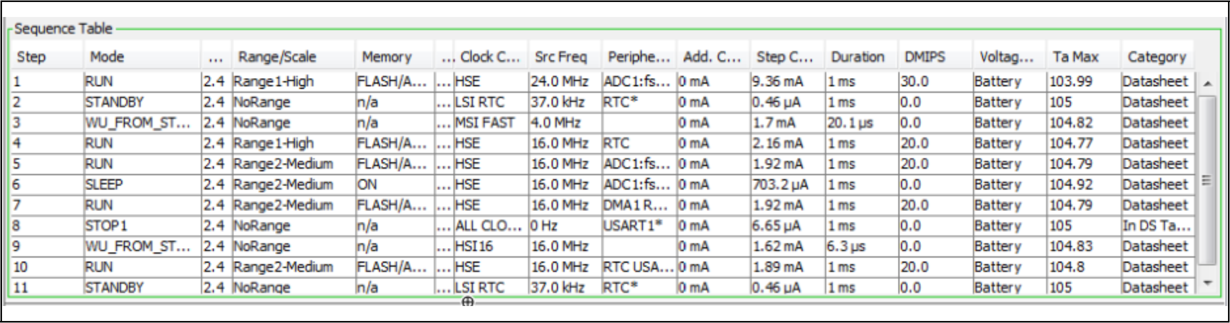
Figure 174. PCC V_{DD} and battery selection menu



Series:	STM32L4
Line:	STM32L4x6
MCU:	STM32L476RGTx
Datasheet:	025976_Rev3
Ambient Temperature (°C):	25
Vdd Power Supply (V):	2.4
Battery:	Li-MnO2(CR2032)
In Series:	1
In Parallel:	1
Capacity:	225.0 mAh
Self Discharge:	0.12 %/month
Nominal Voltage:	6.0 V
Max Cont Current:	3.0 mA
Max Pulse Current:	15.0 mA

5. Enable the **Transition checker** to ensure the sequence is valid (see *Figure 174*). This option allows verifying that the sequence respects the allowed transitions implemented within the STM32L476RG.
6. Click the **Add** button to add steps that match the sequence described in *Figure 174*.
   - By default the steps last 1 ms each, except for the wakeup transitions that are preset using the transition times specified in the product datasheet (see *Figure 175*).
   - Some peripherals for which consumption is unavailable or negligible are highlighted with '*' (see *Figure 175*).

Figure 175. PCC Sequence table

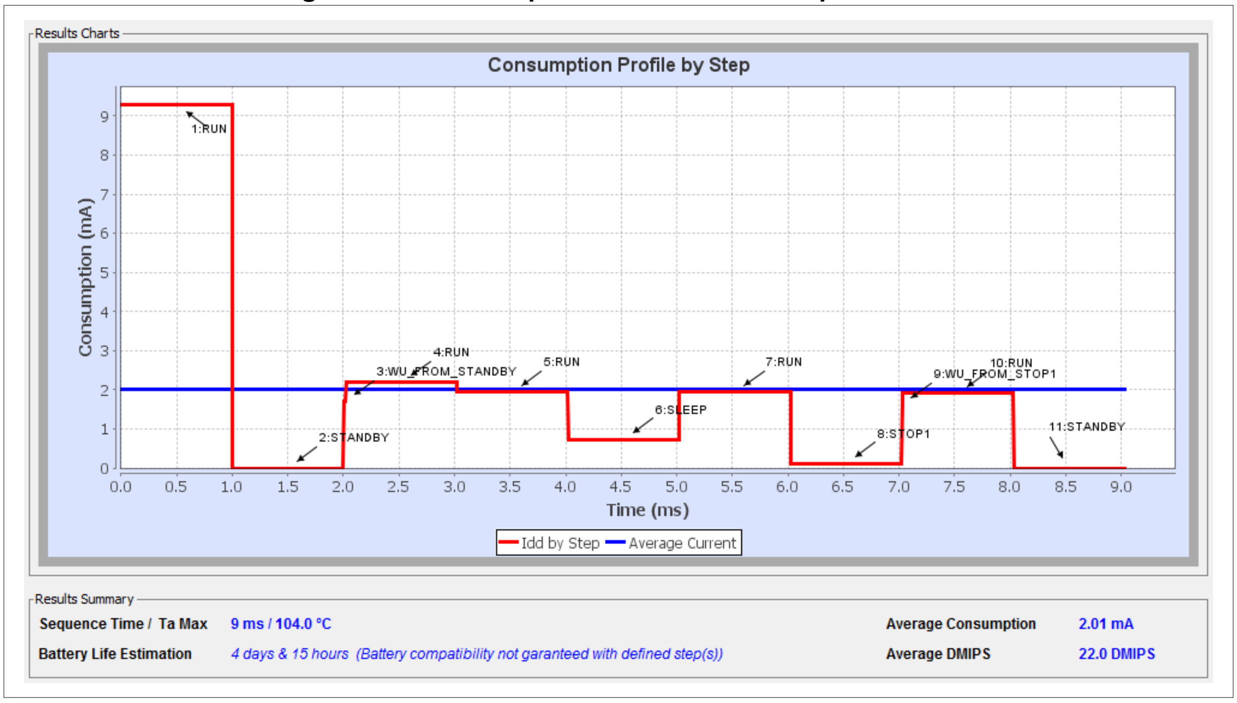


Step	Mode	...	Range/Scale	Memory	...	Clock C...	Src Freq	Periphe...	Add. C...	Step C...	Duration	DMIPS	Voltag...	Ta Max	Category
1	RUN	2.4	Range1-High	FLASH/A...	...	HSE	24.0 MHz	ADC1:fs...	0 mA	9.36 mA	1 ms	30.0	Battery	103.99	Datasheet
2	STANDBY	2.4	NoRange	n/a	...	LSI RTC	37.0 kHz	RTC*	0 mA	0.46 µA	1 ms	0.0	Battery	105	Datasheet
3	WU_FROM_ST...	2.4	NoRange	n/a	...	MSI FAST	4.0 MHz		0 mA	1.7 mA	20.1 µs	0.0	Battery	104.82	Datasheet
4	RUN	2.4	Range1-High	FLASH/A...	...	HSE	16.0 MHz	RTC	0 mA	2.16 mA	1 ms	20.0	Battery	104.77	Datasheet
5	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	ADC1:fs...	0 mA	1.92 mA	1 ms	20.0	Battery	104.79	Datasheet
6	SLEEP	2.4	Range2-Medium	ON	...	HSE	16.0 MHz	ADC1:fs...	0 mA	703.2 µA	1 ms	0.0	Battery	104.92	Datasheet
7	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	DMA1 R...	0 mA	1.92 mA	1 ms	20.0	Battery	104.79	Datasheet
8	STOP1	2.4	NoRange	n/a	...	ALL CLO...	0 Hz	USART1*	0 mA	6.65 µA	1 ms	0.0	Battery	105	In DS Ta...
9	WU_FROM_ST...	2.4	NoRange	n/a	...	HSI16	16.0 MHz		0 mA	1.62 mA	6.3 µs	0.0	Battery	104.83	Datasheet
10	RUN	2.4	Range2-Medium	FLASH/A...	...	HSE	16.0 MHz	RTC USA...	0 mA	1.89 mA	1 ms	20.0	Battery	104.8	Datasheet
11	STANDBY	2.4	NoRange	n/a	...	LSI RTC	37.0 kHz	RTC*	0 mA	0.46 µA	1 ms	0.0	Battery	105	Datasheet

7. Click the **Save** button to save the sequence as SequenceOne.

The application consumption profile is the generated. It shows that the overall sequence consumes an average of 2.01 mA for 9 ms, and the battery lifetime is only 4 days (see *Figure 176*).

Figure 176. PCC sequence results before optimization



Results Summary

Sequence Time / Ta Max	9 ms / 104.0 °C	Average Consumption	2.01 mA
Battery Life Estimation	*4 days & 15 hours (Battery compatibility not garanteed with defined step(s))*	Average DMIPS	22.0 DMIPS

8.3.2 Optimizing application power consumption

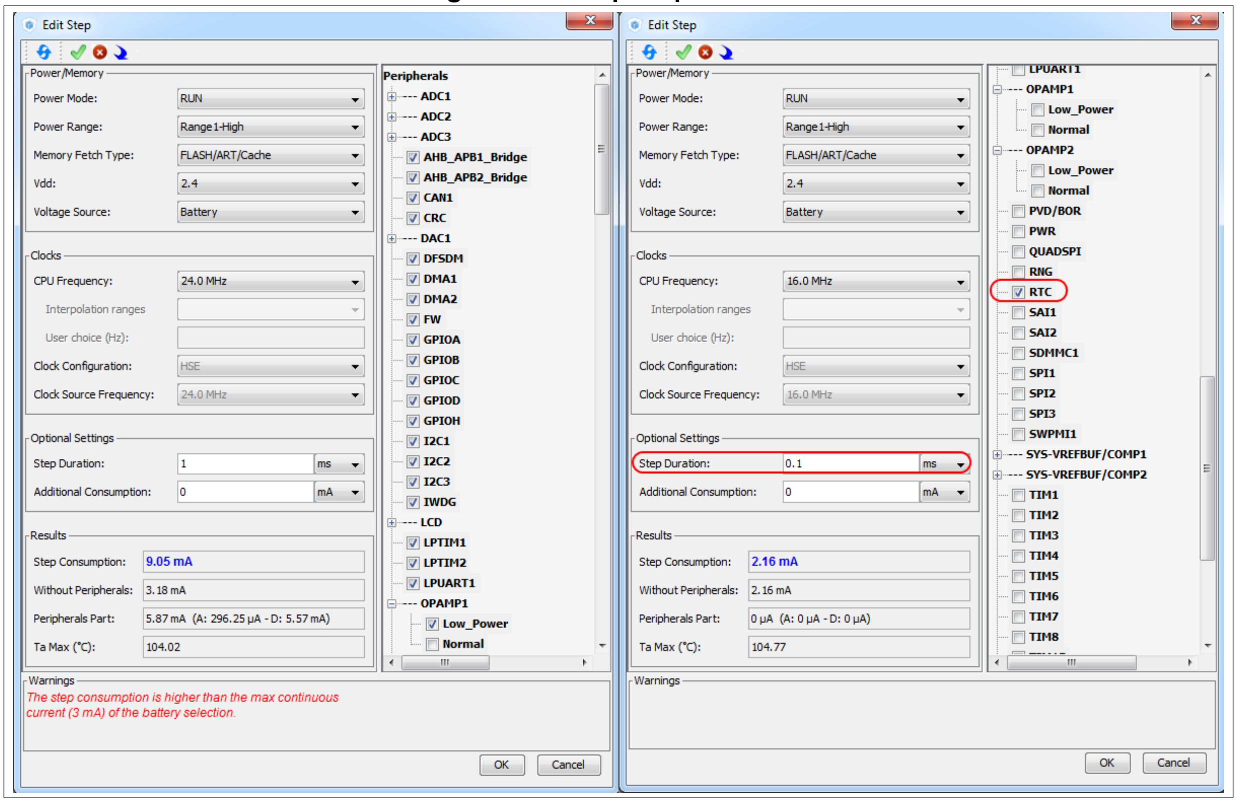
Let us now take several actions to optimize the overall consumption and the battery lifetime. These actions are performed on step 1, 4, 5, 6, 7, 8 and 10.

The next figures show on the left the original step and on the right the step updated with several optimization actions.

Step 1 (Run)

- Findings
  All peripherals are enabled although the application requires only the RTC.
- Actions
  – Lower the operating frequency.
  – Enable solely the RTC peripheral.
  – To reduce the average current consumption, reduce the time spent in this mode.
- Results
  The current is reduced from 9.05 mA to 2.16 mA (see *Figure 177*).

Figure 177. Step 1 optimization



Edit Step (left)

Power/Memory
- Power Mode: RUN
- Power Range: Range1-High
- Memory Fetch Type: FLASH/ART/Cache
- Vdd: 2.4
- Voltage Source: Battery

Clocks
- CPU Frequency: 24.0 MHz
- Interpolation ranges:
- User choice (Hz):
- Clock Configuration: HSE
- Clock Source Frequency: 24.0 MHz

Optional Settings
- Step Duration: 1 ms
- Additional Consumption: 0 mA

Results
- Step Consumption: 9.05 mA
- Without Peripherals: 3.18 mA
- Peripherals Part: 5.87 mA (A: 296.25 µA - D: 5.57 mA)
- Ta Max (°C): 104.02

Peripherals
- ADC1
- ADC2
- ADC3
- ✓ AHB_APB1_Bridge
- ✓ AHB_APB2_Bridge
- ✓ CAN1
- ✓ CRC
- DAC1
- ✓ DFSDM
- ✓ DMA1
- ✓ DMA2
- ✓ FW
- ✓ GPIOA
- ✓ GPIOB
- ✓ GPIOC
- ✓ GPIOD
- ✓ GPIOH
- ✓ I2C1
- ✓ I2C2
- ✓ I2C3
- ✓ IWDG
- LCD
- ✓ LPTIM1
- ✓ LPTIM2
- ✓ LPUART1
- OPAMP1
  - ✓ Low_Power
  - Normal

Warnings
The step consumption is higher than the max continuous current (3 mA) of the battery selection.

Edit Step (right)

Power/Memory
- Power Mode: RUN
- Power Range: Range1-High
- Memory Fetch Type: FLASH/ART/Cache
- Vdd: 2.4
- Voltage Source: Battery

Clocks
- CPU Frequency: 16.0 MHz
- Interpolation ranges:
- User choice (Hz):
- Clock Configuration: HSE
- Clock Source Frequency: 16.0 MHz

Optional Settings
- Step Duration: 0.1 ms
- Additional Consumption: 0 mA

Results
- Step Consumption: 2.16 mA
- Without Peripherals: 2.16 mA
- Peripherals Part: 0 µA (A: 0 µA - D: 0 µA)
- Ta Max (°C): 104.77

Peripherals
- LPUART1
- OPAMP1
  - Low_Power
  - Normal
- OPAMP2
  - Low_Power
  - Normal
- PVD/BOR
- PWR
- QUADSPI
- RNG
- ✓ RTC
- SAI1
- SAI2
- SDMMC1
- SPI1
- SPI2
- SPI3
- SWPMI1
- SYS-VREFBUF/COMP1
- SYS-VREFBUF/COMP2
- TIM1
- TIM2
- TIM3
- TIM4
- TIM5
- TIM6
- TIM7
- TIM8

Warnings

Step 4 (Run, RTC)

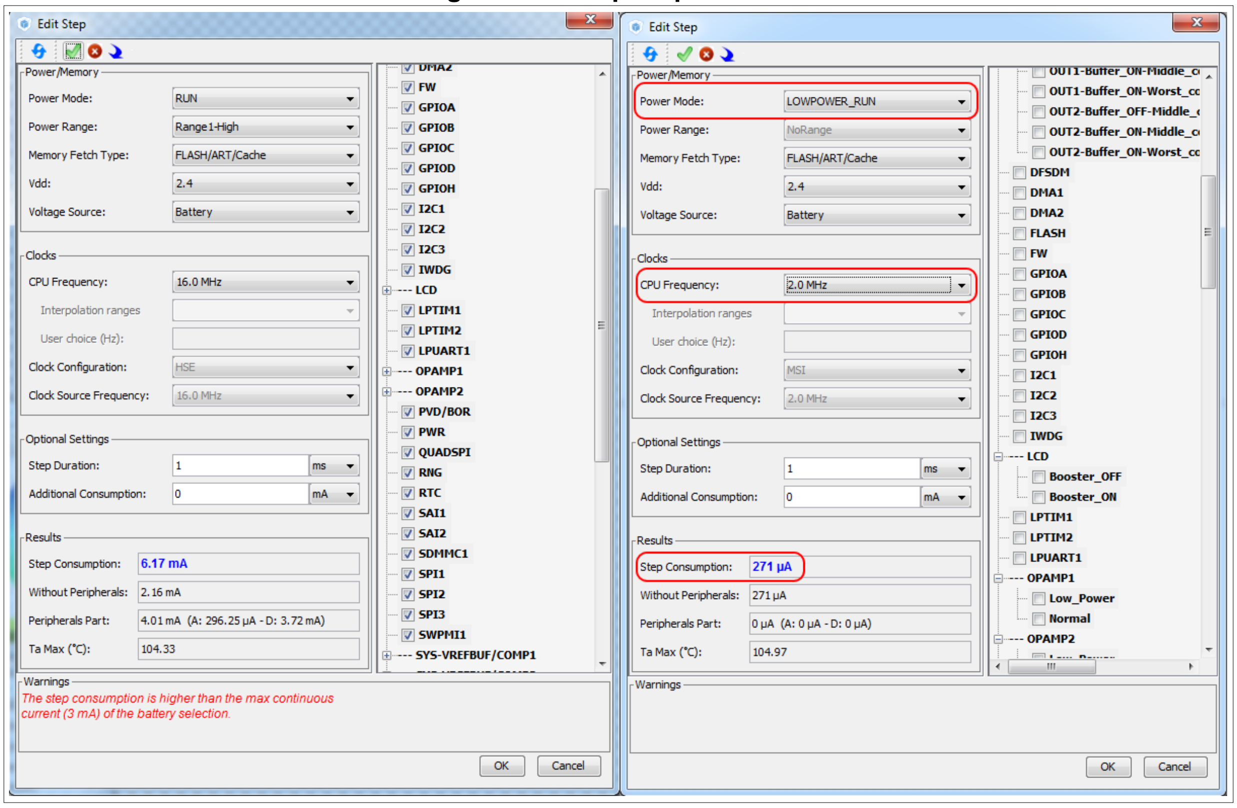
- Action:
  Reduce the time spent in this mode to 0.1 ms.

Step 5 (Run, ADC, DMA, RTC)

- Actions
  - Change to Low-power run mode.
  - Lower the operating frequency.
- Results

  The current consumption is reduced from 6.17 mA to 271 µA (see *Figure 178*).

Figure 178. Step 5 optimization



Edit Step

Power/Memory

Power Mode:	RUN
Power Range:	Range1-High
Memory Fetch Type:	FLASH/ART/Cache
Vdd:	2.4
Voltage Source:	Battery

Clocks

CPU Frequency:	16.0 MHz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	HSE
Clock Source Frequency:	16.0 MHz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	6.17 mA
Without Peripherals:	2.16 mA
Peripherals Part:	4.01 mA (A: 296.25 µA - D: 3.72 mA)
Ta Max (°C):	104.33

✓ DMA2, ✓ FW, ✓ GPIOA, ✓ GPIOB, ✓ GPIOC, ✓ GPIOD, ✓ GPIOH, ✓ I2C1, ✓ I2C2, ✓ I2C3, ✓ IWDG, --- LCD, ✓ LPTIM1, ✓ LPTIM2, ✓ LPUART1, --- OPAMP1, --- OPAMP2, ✓ PVD/BOR, ✓ PWR, ✓ QUADSPI, ✓ RNG, ✓ RTC, ✓ SAI1, ✓ SAI2, ✓ SDMMC1, ✓ SPI1, ✓ SPI2, ✓ SPI3, ✓ SWPMI1, --- SYS-VREFBUF/COMP1

Warnings

The step consumption is higher than the max continuous current (3 mA) of the battery selection.

OK Cancel

Edit Step

Power/Memory

Power Mode:	LOWPOWER_RUN
Power Range:	NoRange
Memory Fetch Type:	FLASH/ART/Cache
Vdd:	2.4
Voltage Source:	Battery

Clocks

CPU Frequency:	2.0 MHz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	MSI
Clock Source Frequency:	2.0 MHz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	271 µA
Without Peripherals:	271 µA
Peripherals Part:	0 µA (A: 0 µA - D: 0 µA)
Ta Max (°C):	104.97

OUT1-Buffer_ON-Middle_c, OUT1-Buffer_ON-Worst_cc, OUT2-Buffer_OFF-Middle_, OUT2-Buffer_ON-Middle_c, OUT2-Buffer_ON-Worst_cc, DFSDM, DMA1, DMA2, FLASH, FW, GPIOA, GPIOB, GPIOC, GPIOD, GPIOH, I2C1, I2C2, I2C3, IWDG, --- LCD, Booster_OFF, Booster_ON, LPTIM1, LPTIM2, LPUART1, --- OPAMP1, Low_Power, Normal, --- OPAMP2

Warnings

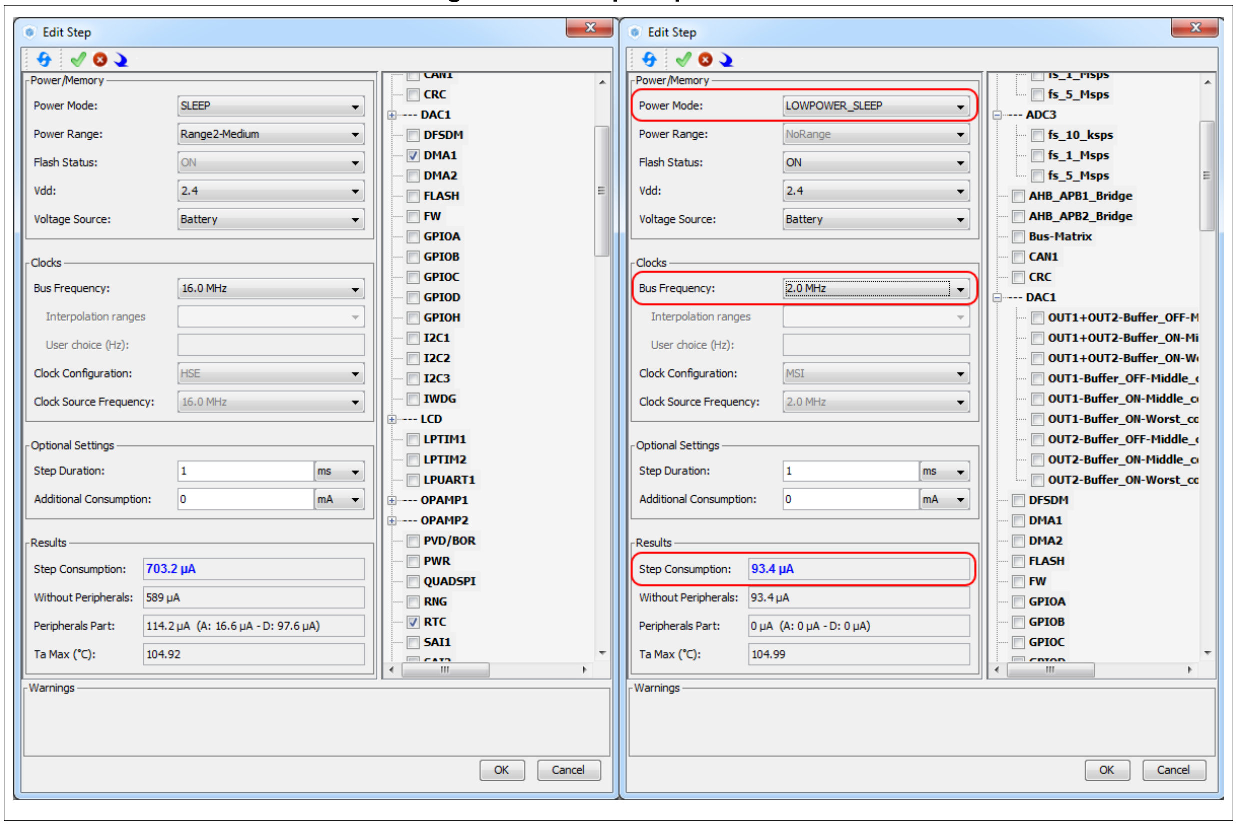
OK Cancel

Step 6 (Sleep, DMA, ADC,RTC)

- Actions
  - Switch to Lower-power sleep mode (BAM mode)
  - Reduce the operating frequency to 2 MHz.
- Results

  The current consumption is reduced from 703 µA to 93 µA (see *Figure 179*).

Figure 179. Step 6 optimization



Edit Step

Power/Memory

Power Mode:	SLEEP
Power Range:	Range2-Medium
Flash Status:	ON
Vdd:	2.4
Voltage Source:	Battery

Clocks

Bus Frequency:	16.0 MHz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	HSE
Clock Source Frequency:	16.0 MHz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	703.2 µA
Without Peripherals:	589 µA
Peripherals Part:	114.2 µA (A: 16.6 µA - D: 97.6 µA)
Ta Max (°C):	104.92

CAN1, CRC, DAC1, DFSDM, DMA1 ✓, DMA2, FLASH, FW, GPIOA, GPIOB, GPIOC, GPIOD, GPIOH, I2C1, I2C2, I2C3, IWDG, LCD, LPTIM1, LPTIM2, LPUART1, OPAMP1, OPAMP2, PVD/BOR, PWR, QUADSPI, RNG, RTC ✓, SAI1

Warnings

OK Cancel

Edit Step

Power/Memory

Power Mode:	LOWPOWER_SLEEP
Power Range:	NoRange
Flash Status:	ON
Vdd:	2.4
Voltage Source:	Battery

Clocks

Bus Frequency:	2.0 MHz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	MSI
Clock Source Frequency:	2.0 MHz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	93.4 µA
Without Peripherals:	93.4 µA
Peripherals Part:	0 µA (A: 0 µA - D: 0 µA)
Ta Max (°C):	104.99

fs_1_Msps, fs_5_Msps, ADC3, fs_10_ksps, fs_1_Msps, fs_5_Msps, AHB_APB1_Bridge, AHB_APB2_Bridge, Bus-Matrix, CAN1, CRC, DAC1, OUT1+OUT2-Buffer_OFF-M, OUT1+OUT2-Buffer_ON-Mi, OUT1+OUT2-Buffer_ON-W, OUT1-Buffer_OFF-Middle_, OUT1-Buffer_ON-Middle_c, OUT1-Buffer_ON-Worst_c, OUT2-Buffer_OFF-Middle_, OUT2-Buffer_ON-Middle_c, OUT2-Buffer_ON-Worst_c, DFSDM, DMA1, DMA2, FLASH, FW, GPIOA, GPIOB, GPIOC

Warnings

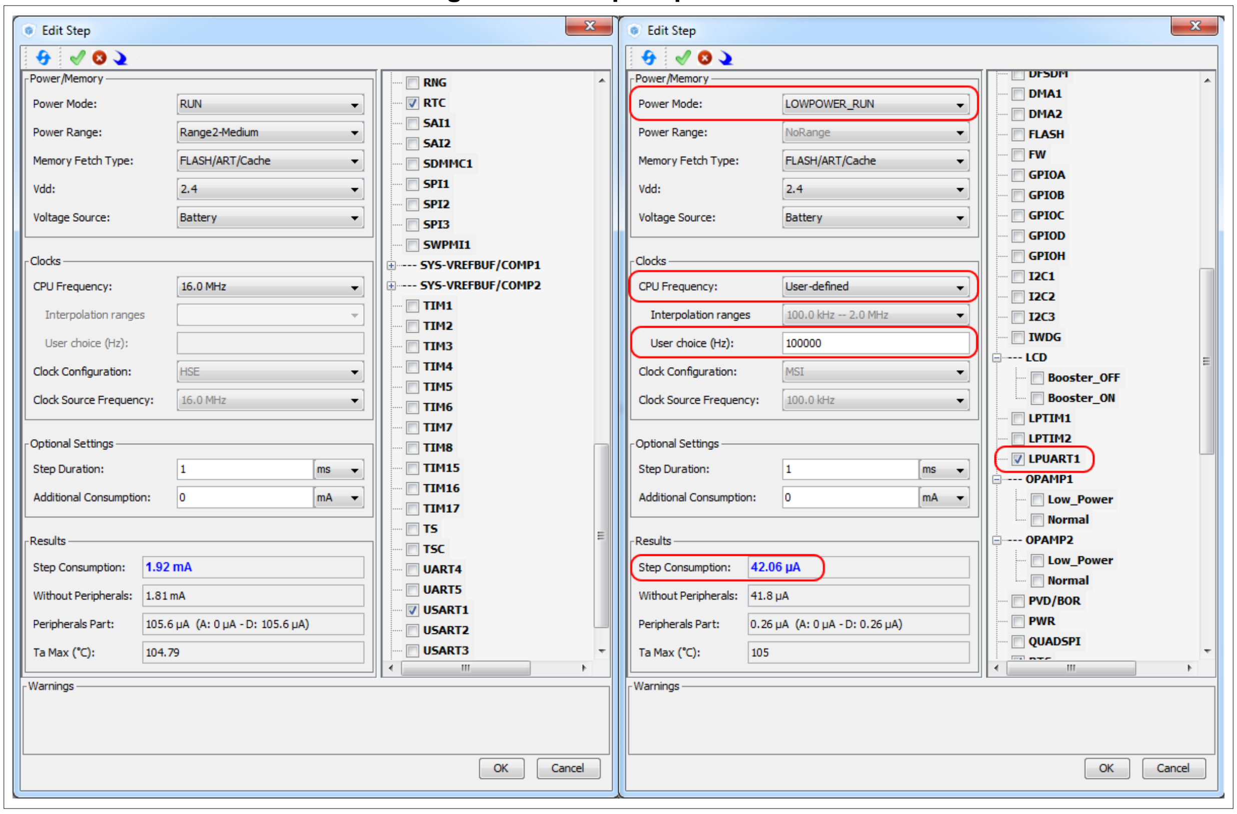
OK Cancel

Step 7 (Run, DMA, RTC, USART)

- Actions
  - Switch to Lower-power run mode.
  - Use the power-efficient LPUART peripheral.
  - Reduce the operating frequency to 1 MHz using the PCC interpolation feature.
- Results

  The current consumption is reduced from 1.92 µA to 42 µA (see *Figure 180*).

Figure 180. Step 7 optimization



Edit Step

Power/Memory
Power Mode: RUN
Power Range: Range2-Medium
Memory Fetch Type: FLASH/ART/Cache
Vdd: 2.4
Voltage Source: Battery

Clocks
CPU Frequency: 16.0 MHz
Interpolation ranges
User choice (Hz):
Clock Configuration: HSE
Clock Source Frequency: 16.0 MHz

Optional Settings
Step Duration: 1 ms
Additional Consumption: 0 mA

Results
Step Consumption: 1.92 mA
Without Peripherals: 1.81 mA
Peripherals Part: 105.6 µA (A: 0 µA - D: 105.6 µA)
Ta Max (°C): 104.79

Warnings

RNG, RTC ✓, SAI1, SAI2, SDMMC1, SPI1, SPI2, SPI3, SWPMI1, SYS-VREFBUF/COMP1, SYS-VREFBUF/COMP2, TIM1, TIM2, TIM3, TIM4, TIM5, TIM6, TIM7, TIM8, TIM15, TIM16, TIM17, TS, TSC, UART4, UART5, USART1 ✓, USART2, USART3

OK Cancel

Edit Step

Power/Memory
Power Mode: LOWPOWER_RUN
Power Range: NoRange
Memory Fetch Type: FLASH/ART/Cache
Vdd: 2.4
Voltage Source: Battery

Clocks
CPU Frequency: User-defined
Interpolation ranges: 100.0 kHz -- 2.0 MHz
User choice (Hz): 100000
Clock Configuration: MSI
Clock Source Frequency: 100.0 kHz

Optional Settings
Step Duration: 1 ms
Additional Consumption: 0 mA

Results
Step Consumption: 42.06 µA
Without Peripherals: 41.8 µA
Peripherals Part: 0.26 µA (A: 0 µA - D: 0.26 µA)
Ta Max (°C): 105

Warnings

DFSDM, DMA1, DMA2, FLASH, FW, GPIOA, GPIOB, GPIOC, GPIOD, GPIOH, I2C1, I2C2, I2C3, IWDG, LCD (Booster_OFF, Booster_ON), LPTIM1, LPTIM2, LPUART1 ✓, OPAMP1 (Low_Power, Normal), OPAMP2 (Low_Power, Normal), PVD/BOR, PWR, QUADSPI, RTC

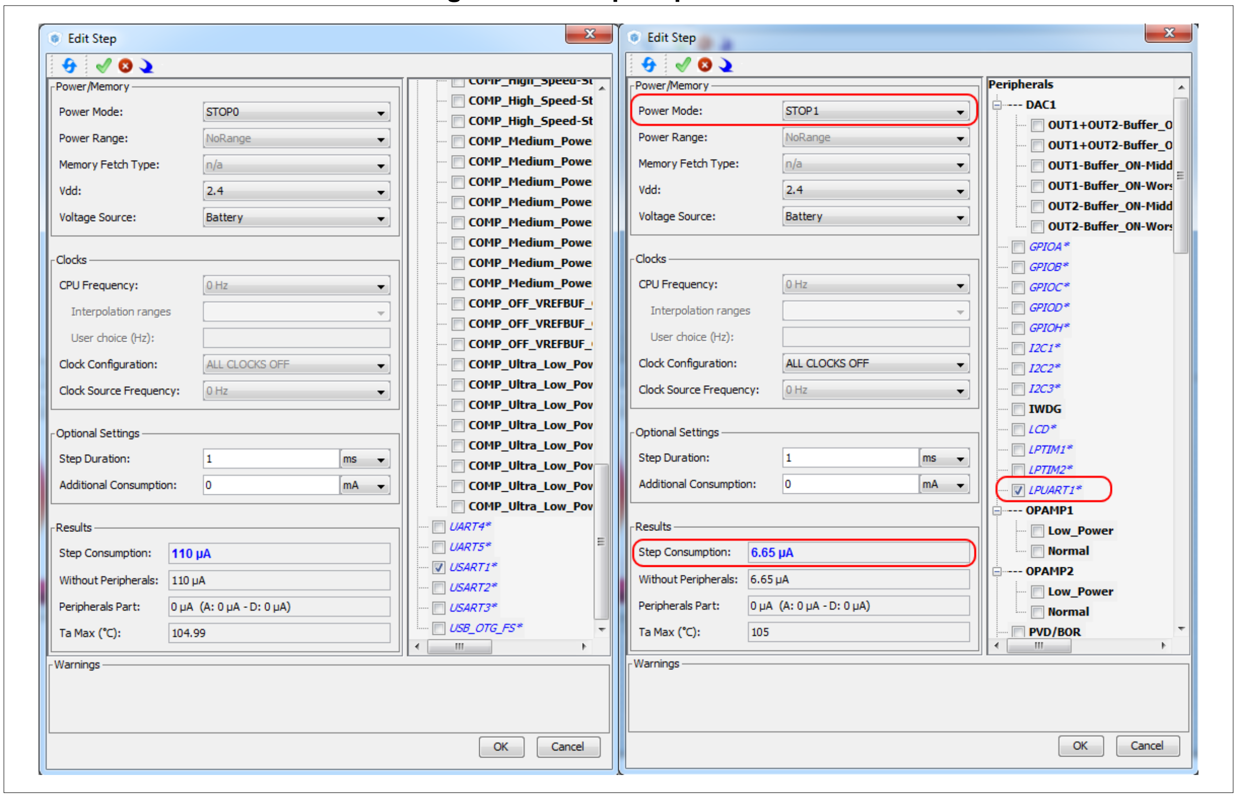
OK Cancel

Step 8 (Stop 0, USART)

- **Actions**:
  - Switch to Stop1 low-power mode.
  - Use the power-efficient LPUART peripheral.
- **Results**

  The current consumption is reduced from 110 µA to 6.65 µA (see *Figure 181*).

Figure 181. Step 8 optimization



Edit Step

Power/Memory

Power Mode:	STOP0
Power Range:	NoRange
Memory Fetch Type:	n/a
Vdd:	2.4
Voltage Source:	Battery

Clocks

CPU Frequency:	0 Hz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	ALL CLOCKS OFF
Clock Source Frequency:	0 Hz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	110 µA
Without Peripherals:	110 µA
Peripherals Part:	0 µA (A: 0 µA - D: 0 µA)
Ta Max (°C):	104.99

Warnings

COMP_High_Speed-St, COMP_High_Speed-St, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_Medium_Powe, COMP_OFF_VREFBUF_, COMP_OFF_VREFBUF_, COMP_OFF_VREFBUF_, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, COMP_Ultra_Low_Pov, UART4*, UART5*, [✓ USART1*], USART2*, USART3*, USB_OTG_FS*

Edit Step

Power/Memory

Power Mode:	STOP1
Power Range:	NoRange
Memory Fetch Type:	n/a
Vdd:	2.4
Voltage Source:	Battery

Clocks

CPU Frequency:	0 Hz
Interpolation ranges	
User choice (Hz):	
Clock Configuration:	ALL CLOCKS OFF
Clock Source Frequency:	0 Hz

Optional Settings

Step Duration:	1 ms
Additional Consumption:	0 mA

Results

Step Consumption:	6.65 µA
Without Peripherals:	6.65 µA
Peripherals Part:	0 µA (A: 0 µA - D: 0 µA)
Ta Max (°C):	105

Warnings

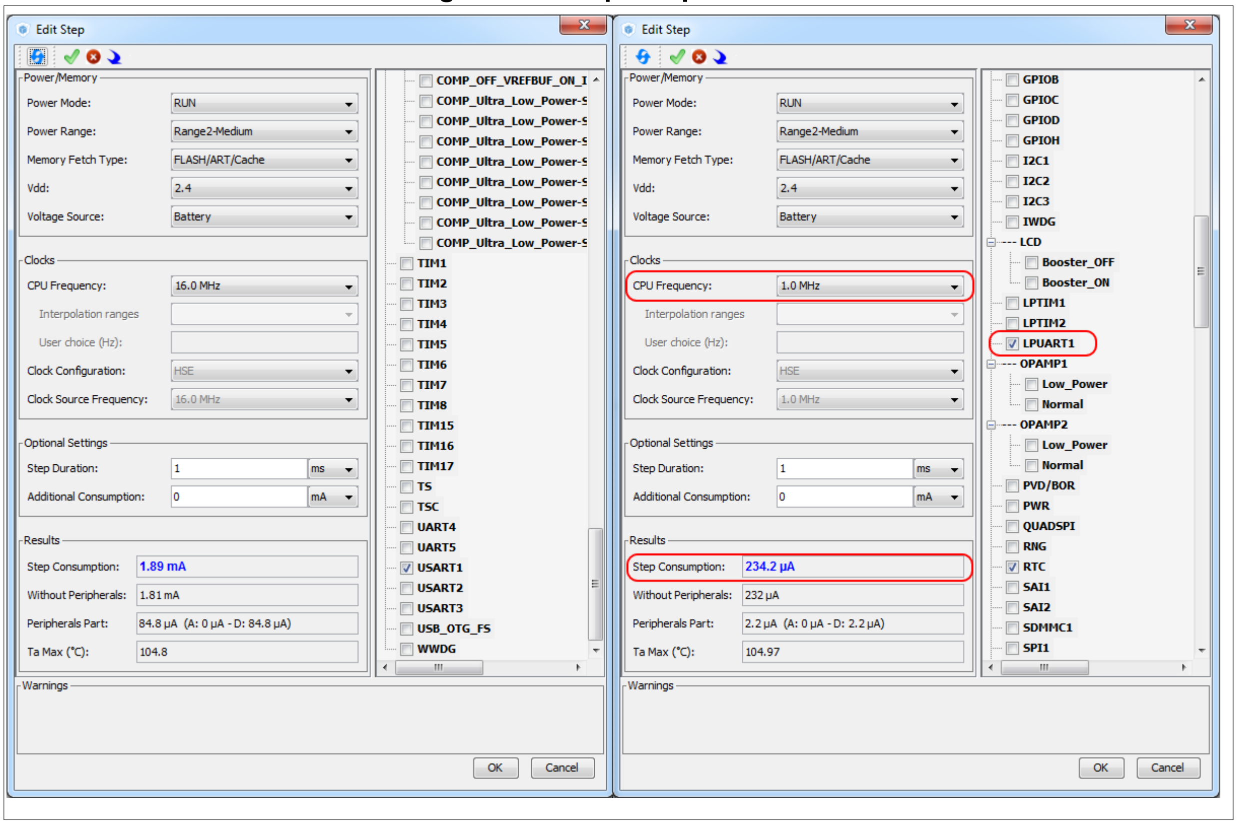
Peripherals

- DAC1
  - OUT1+OUT2-Buffer_O
  - OUT1+OUT2-Buffer_O
  - OUT1-Buffer_ON-Midd
  - OUT1-Buffer_ON-Wors
  - OUT2-Buffer_ON-Midd
  - OUT2-Buffer_ON-Wors
- GPIOA*
- GPIOB*
- GPIOC*
- GPIOD*
- GPIOH*
- I2C1*
- I2C2*
- I2C3*
- IWDG
- LCD*
- LPTIM1*
- LPTIM2*
- [✓ LPUART1*]
- OPAMP1
  - Low_Power
  - Normal
- OPAMP2
  - Low_Power
  - Normal
- PVD/BOR

Step 10 (RTC, USART)

- Actions
  – Use the power-efficient LPUART peripheral.
  – Reduce the operating frequency to 1 MHz.
- Results
  The current consumption is reduced from 1.89 mA to 234 µA (see *Figure 182*).
  The example given in *Figure 183* shows an average current consumption reduction of 155 µA.

Figure 182. Step 10 optimization



Edit Step (left)

Power/Memory
Power Mode: RUN
Power Range: Range2-Medium
Memory Fetch Type: FLASH/ART/Cache
Vdd: 2.4
Voltage Source: Battery

Clocks
CPU Frequency: 16.0 MHz
Interpolation ranges
User choice (Hz):
Clock Configuration: HSE
Clock Source Frequency: 16.0 MHz

Optional Settings
Step Duration: 1 ms
Additional Consumption: 0 mA

Results
Step Consumption: 1.89 mA
Without Peripherals: 1.81 mA
Peripherals Part: 84.8 µA (A: 0 µA - D: 84.8 µA)
Ta Max (°C): 104.8

Warnings

COMP_OFF_VREFBUF_ON_I, COMP_Ultra_Low_Power-S (×8), TIM1, TIM2, TIM3, TIM4, TIM5, TIM6, TIM7, TIM8, TIM15, TIM16, TIM17, TS, TSC, UART4, UART5, ✓ USART1, USART2, USART3, USB_OTG_FS, WWDG

OK Cancel

Edit Step (right)

Power/Memory
Power Mode: RUN
Power Range: Range2-Medium
Memory Fetch Type: FLASH/ART/Cache
Vdd: 2.4
Voltage Source: Battery

Clocks
CPU Frequency: 1.0 MHz
Interpolation ranges
User choice (Hz):
Clock Configuration: HSE
Clock Source Frequency: 1.0 MHz

Optional Settings
Step Duration: 1 ms
Additional Consumption: 0 mA

Results
Step Consumption: 234.2 µA
Without Peripherals: 232 µA
Peripherals Part: 2.2 µA (A: 0 µA - D: 2.2 µA)
Ta Max (°C): 104.97

Warnings

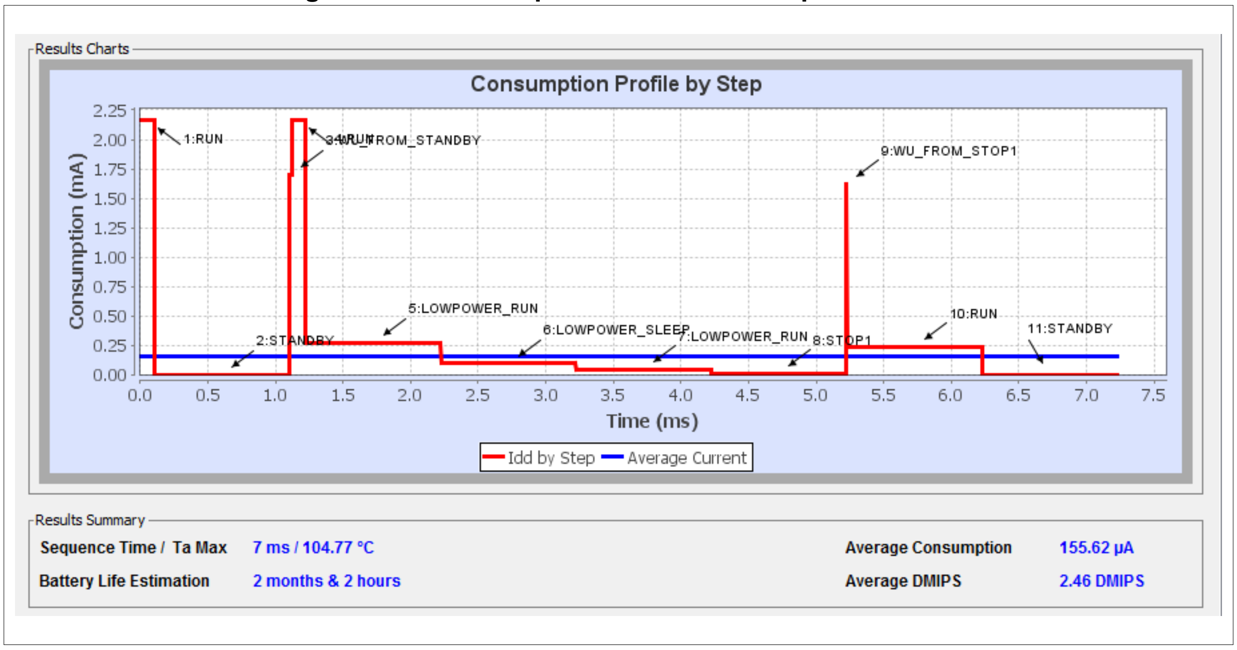
GPIOB, GPIOC, GPIOD, GPIOH, I2C1, I2C2, I2C3, IWDG, LCD (Booster_OFF, Booster_ON), LPTIM1, LPTIM2, ✓ LPUART1, OPAMP1 (Low_Power, Normal), OPAMP2 (Low_Power, Normal), PVD/BOR, PWR, QUADSPI, RNG, ✓ RTC, SAI1, SAI2, SDMMC1, SPI1

OK Cancel

See *Figure 183* for the sequence overall results: 7 ms duration, about 2 month battery life, and an average current consumption of 165.25 µA.

Use the **compare** button to compare the current results to the original ones saved as SequenceOne.pcs.

Figure 183. PCC Sequence results after optimizations



Sequence Time / Ta Max	7 ms / 104.77 °C	Average Consumption	155.62 µA
Battery Life Estimation	2 months & 2 hours	Average DMIPS	2.46 DMIPS

9 FAQ

9.1 On the Pinout configuration pane, why does STM32CubeMX move some functions when I add a new peripheral mode?

You may have unselected ☐ Keep Current Signals Placement . In this case, the tool performs an automatic remapping to optimize your placement.

9.2 How can I manually force a function remapping?

You should use the **Manual Remapping** feature.

9.3 Why are some pins highlighted in yellow or in light green in the Chip view? Why cannot I change the function of some pins (when I click some pins, nothing happens)?

These pins are specific pins (such as power supply or BOOT) which are not available as peripheral signals.

9.4 Why do I get the error “Java 7 update 45’ when installing ‘Java 7 update 45’ or a more recent version of the JRE?

The problem generally occurs on 64-bit Windows operating system, when several versions of Java are installed on your computer and the 64-bit Java installation is too old.

During STM32CubeMX installation, the computer searches for a 64-bit installation of Java.

- If one is found, the ‘Java 7 update 45’ minimum version prerequisite is checked. If the installed version is older, an error is displayed to request the upgrade.
- If no 64-bit installation is found, STM32CubeMX searches for a 32-bit installation. If one is found and the version is too old, the ‘Java 7 update 45’ error is displayed. The user must update the installation to solve the issue.

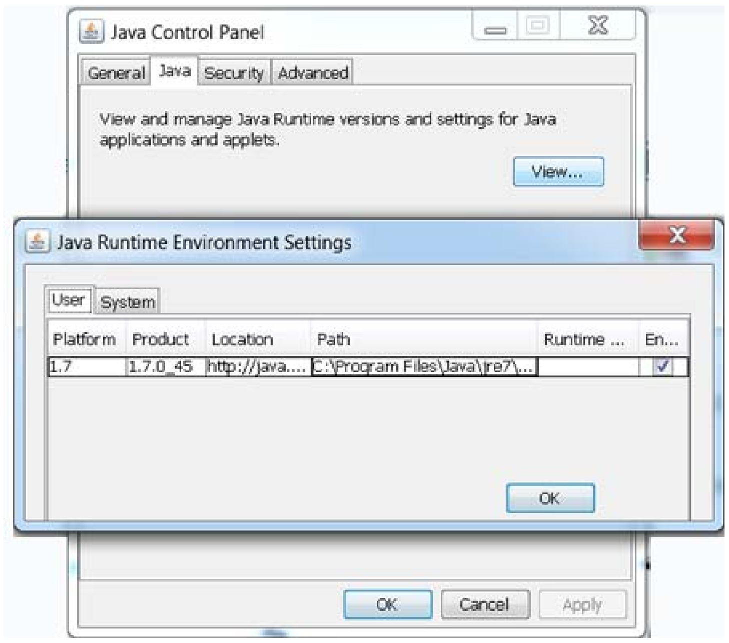
To avoid this issue from occurring, it is recommended to perform one of the following actions:

1. Remove all Java installations and reinstall only one version (32 or 64 bits) (Java 7 update 45 or more recent).
2. Keep 32-bit and 64-bit installations but make sure that the 64-bit version is at least Java 7 update 45.

Note: Some users (Java developers for example) may need to check the PC environment variables defining hard-coded Java paths (e.g. JAVA_HOME or PATH) and update them so that they point to the latest Java installation.

On Windows 7 you can check your Java installation using the Control Panel. To do this, double-click Java icon from Control Panel\All Control Panel to open the Java settings window (see *Figure 184*):

Figure 184. Java Control Panel



View and manage Java Runtime versions and settings for Java applications and applets.

Platform	Product	Location	Path	Runtime ...	En...
1.7	1.7.0_45	http://java....	C:\Program Files\Java\jre7\...		✓

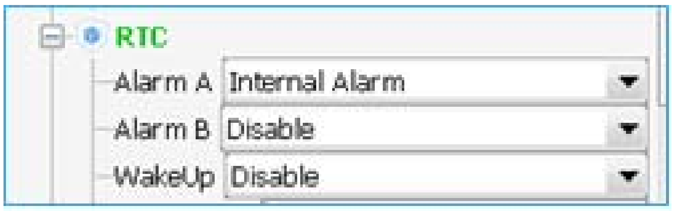
You can also enter *'java –version*' as an MS-DOS command to check the version of your latest Java installation (the Java program called here is a copy of the program installed under C:\Windows\System32):

```
java version "1.7.0_45"
Java (TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot (TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
```

9.5 Why does the RTC multiplexer remain inactive on the Clock tree view?

To enable the RTC multiplexer, the user shall enable the RTC IP in the **Pinout** view as indicated in below:

Figure 185. Pinout view - Enabling the RTC

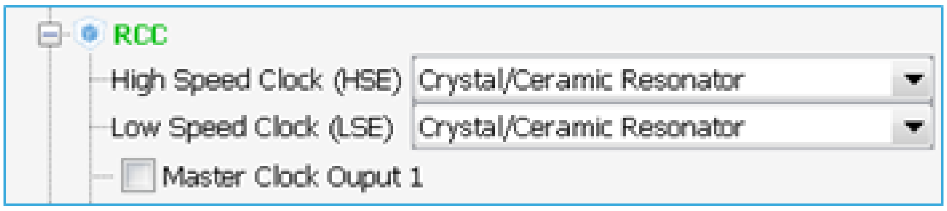


Alarm A: Internal Alarm
Alarm B: Disable
WakeUp: Disable

9.6 How can I select LSE and HSE as clock source and change the frequency?

The LSE and HSE clocks become active once the RCC is configured as such in the **Pinout** view. See *Figure 186* for an example.

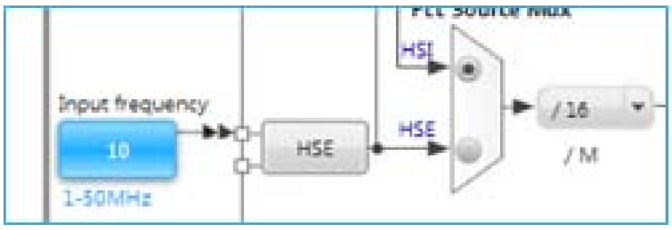
Figure 186. Pinout view - Enabling LSE and HSE clocks



RCC
High Speed Clock (HSE) Crystal/Ceramic Resonator
Low Speed Clock (LSE) Crystal/Ceramic Resonator
Master Clock Ouput 1

The clock source frequency can then be edited and the external source selected:

Figure 187. Pinout view - Setting LSE/HSE clock frequency



9.7 Why STM32CubeMX does not allow me to configure PC13, PC14, PC15 and PI8 as outputs when one of them is already configured as an output?

STM32CubeMX implements the restriction documented in the reference manuals as a footnote in table Output Voltage characteristics:

“PC13, PC14, PC15 and PI8 are supplied through the power switch. Since the switch only sinks a limited amount of current (3 mA), the use of GPIOs PC13 to PC15 and PI8 in output mode is limited: the speed should not exceed 2 MHz with a maximum load of 30 pF and these I/Os must not be used as a current source (e.g. to drive a LED).”

Appendix A STM32CubeMX pin assignment rules

The following pin assignment rules are implemented in STM32CubeMX:

- Rule 1: Block consistency
- Rule 2: Block inter-dependency
- Rule 3: One block = one peripheral mode
- Rule 4: Block remapping (only for STM32F10x)
- Rule 5: Function remapping
- Rule 6: Block shifting (only for STM32F10x)
- Rule 7: Setting or clearing a peripheral mode
- Rule 8: Mapping a function individually (if Keep Current Placement is unchecked)
- Rule 9: GPIO signals mapping

A.1 Block consistency

When setting a pin signal (provided there is no ambiguity about the corresponding peripheral mode), all the pins/signals required for this mode are mapped and pins are shown in green (otherwise the configured pin is shown in orange).

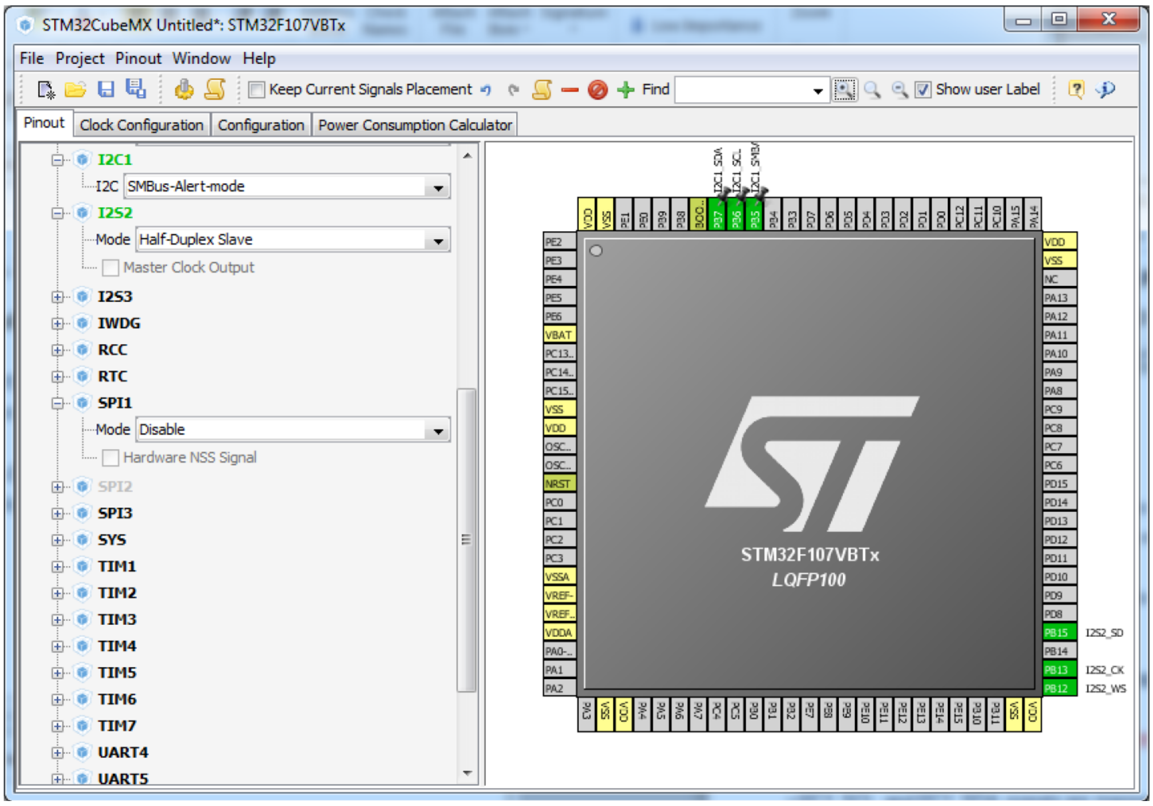
When clearing a pin signal, all the pins/signals required for this mode are unmapped simultaneously and the pins turn back to gray.

Example of block mapping with a STM32F107x MCU

If the user assigns I2C1_SMBA function to PB5, then STM32CubeMX configures pins and modes as follows:

- I2C1_SCL and I2C1_SDA signals are mapped to the PB6 and PB7 pins, respectively (see *Figure 188*).
- I2C1 peripheral mode is set to SMBus-Alert mode.

Figure 188. Block mapping



STM32CubeMX Untitled*: STM32F107VBTx

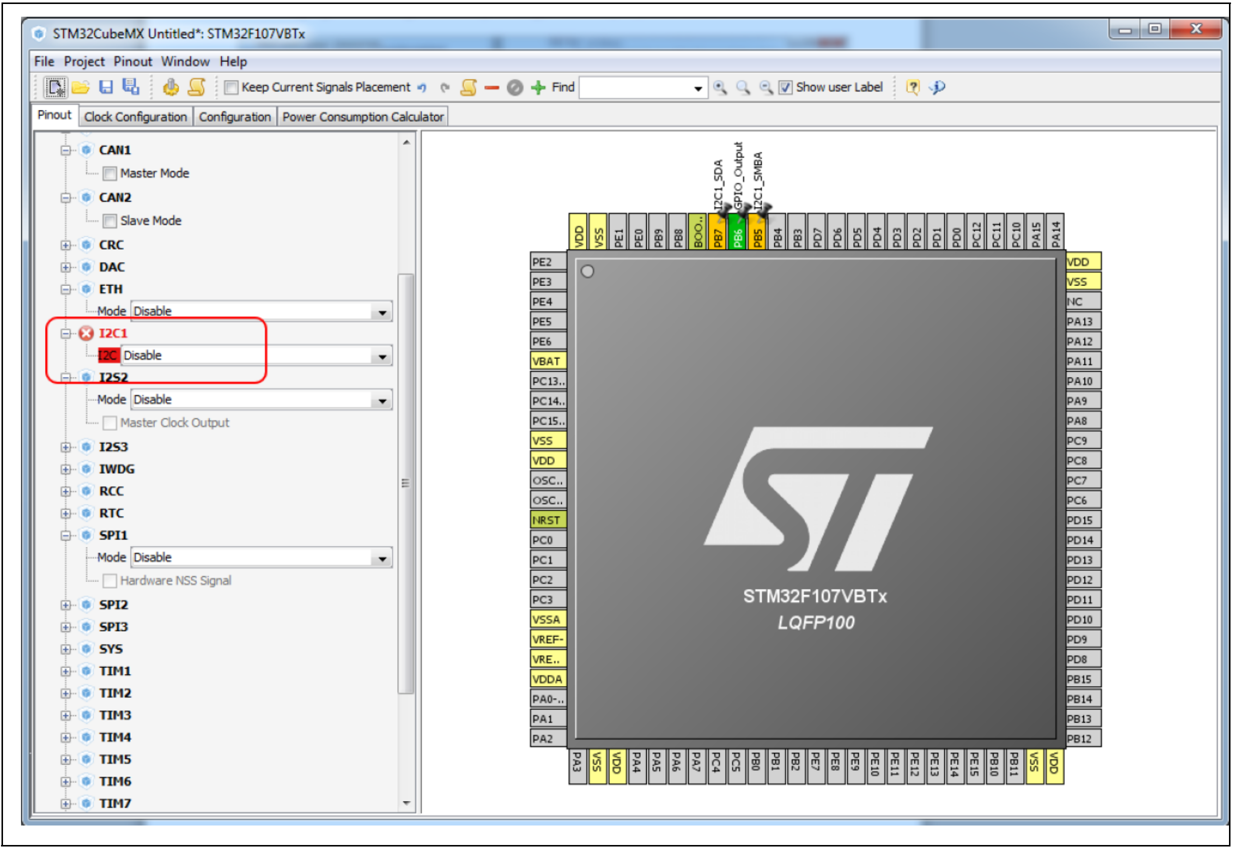
I2C1
I2C SMBus-Alert-mode
I2S2
Mode Half-Duplex Slave
Master Clock Output
SPI1
Mode Disable
Hardware NSS Signal

Example of block remapping with a STM32F107x MCU

If the user assigns GPIO_Output to PB6, STM32CubeMX automatically disables I2C1 SMBus-Alert peripheral mode from the peripheral tree view and updates the other I2C1 pins (PB5 and PB7) as follows:

- If they are unpinned, the pin configuration is reset (pin grayed out).
- If they are pinned, the peripheral signal assigned to the pins is kept and the pins are highlighted in orange since they no longer match a peripheral mode (see *Figure 189*).

Figure 189. Block remapping



For STM32CubeMX to find an alternative solution for the I2C peripheral mode, the user will need to unpin I2C1 pins and select the I2C1 mode from the peripheral tree view (see *Figure 190* and *Figure 191*).