Application Note

N32A455 Series Low Power Application Note

Introduction

In the embedded product development process, battery scenarios may arise where a longer battery life is essential. In such cases, implementing low-power settings becomes necessary. This document focuses on the application scenarios of the NATIONS MCU series products and provides guidance on how to utilize the MCU to manage battery power consumption effectively by leveraging the PWR module to access various low-power modes.

The N32A455 series integrates the latest generation of the embedded ARM Cortex™-M4F processor, which enhances computing power based on the Cortex™-M3 core. It features a new floating-point unit (FPU) and supports DSP and parallel computing instructions, delivering an impressive performance of 1.25 DMIPS/MHz. This series combines efficient signal processing capabilities with the low power consumption, cost-effectiveness, and user-friendly features of the Cortex-M family of processors, making it suitable for applications requiring a mix of control and signal processing.

The N32A455 series offers five low-power operation modes: SLEEP mode, STOP0 mode, STOP2 mode, STANDBY mode, and VBAT mode. Users should select the most appropriate low-power mode based on considerations of power consumption, startup time, and available wake-up sources.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Contents

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

1 Low Power Operation Mode

1.1 SLEEP Mode

In SLEEP mode, only the CPU stops and all peripherals are active and can wake up the CPU in the event of an interrupt/event.

1.1.1 Entering SLEEP Mode

Enter SLEEP mode by executing the WFI (wait interrupt) or WFE (wait event) command and SLEEPDEEP = 0. Based on the value of the SLEEPONEXIT bit in the Cortex®-M4 system control register, there are two options for selecting the SLEEP mode entry mechanism:

- Sleep-now: If SLEEPONEXIT is cleared, the WFI or WFE command will execute immediately and the system will enter SLEEP mode immediately.
- Sleep-on-exit: If SLEEPONEXIT is set, the system enters SLEEP mode as soon as it exits from the lowest-priority interrupt handler.

In SLEEP mode, all I/O pins remain in the same state/function as in run mode.

1.1.2 Exiting SLEEP mode

If you enter SLEEP mode using the WFI command, then any peripheral interrupts responded to by the nested vector interrupt controller (NVIC) can wake the device from SLEEP mode.

If you enter SLEEP mode using the WFE command, the device will exit SLEEP mode as soon as the event occurs. Wake up events can be generated in the following ways:

- Enable an interrupt in the peripheral control register, not in NVIC, and the SEVONPEND bit in the Cortex®-M4 system control register. When MCU recovers from WFE, the peripheral interrupt suspend bit and the peripheral NVIC interrupt channel suspend bit (in the NVIC Interrupt clear suspend register) must be cleared.
- Configure an external or internal EXTI event mode so that when the CPU recovers from WFE, the peripheral interrupt suspend bit and the peripheral NVIC interrupt channel suspend bit (in the NVIC Interrupt clear suspend register) need not be cleared because the suspend bit corresponding to the event line has not been set. This mode provides the shortest wake up time since no time is lost on interrupted entry or exit.

1.2 STOP0 mode

STOP0 mode is based on the Cortex®-M4 deep sleep mode, combined with the peripheral clock control mechanism. The voltage regulator can be configured for normal or low power mode. In STOP0 mode, most clock sources in the core domain, such as PLL, HSI, and HSE, are disabled. But SRAM, R-SRAM, and all register contents are saved. In STOP0 mode, all I/O pins remain in the same state as in run mode.

1.2.1 Entering STOP0 mode

When entering STOP0 mode, the main difference is to set SLEEPDEEP= 1, PDS=0.Another difference is that MR can be run in normal mode or low power mode by configuring the register PWR_CTRL.LPS bit. When LPS = 1, MR runs in low power mode. When LPS = 0, MR runs in normal mode. In STOP0 mode, all I/O pins remain in the same state and function as in run mode. If FLASH operations are in progress, the entry into STOP0 mode will be delayed until memory access is complete. If access to the APB area is in progress, entering STOP0 mode will be delayed until the APB access is complete. In STOP0 mode, the following features can be selected by programming the individual control bits:

- Independent watchdog (IWDG): Independent watchdog is started when software writes or hardware operations are performed on its associated registers. Once started, it works until a reset message is generated
- RTC: this can be enabled by the RTCEN bit in register RCC_BDCTRL
- Internal RC oscillator (LSI RC): can be turned on by the LSIEN bit in register RCC_CTRLSTS
- External 32.768khz crystal oscillator (LSE OSC): Can be turned on by the LSEEN bit in register RCC_BDCTRL
- ADC or DAC can also consume power in STOP0 mode. ADC and DAC can be disabled before entering STOP0 mode.

Note: If your application needs to disable the external clock before entering stop mode, you must first disable the HSEEN bit and then switch the system clock to HSI. Otherwise, if the HSEEN bit remains enabled when entering stop mode and the external clock (external oscillator) is removed, the Clock Safety System (CSS) feature must be enabled to detect any external oscillator failure and avoid failure behavior when entering stop mode.

1.2.2 Exiting STOP0 mode

When an interrupt or wake up event exits STOP0 mode, the HSI RC oscillator is selected as the system clock.

When the voltage regulator is operating in low power mode, there is an additional startup delay when awakened from STOP0 mode. In STOP0 mode, if the internal regulator is in normal mode, which can reduce the startup time, but the corresponding power consumption will increase.

1.3 STOP2 mode

STOP2 mode is based on the Cortex-M4 deep sleep mode, and all the core digital logic areas are powered off. Main voltage regulator (MR) is off, HSE/HSI/PLL is off. CPU registers are maintained, LSE/LSI can be configured, GPIO is remained, and peripheral I/O multiplexing is not remained.16K bytes of R-SRAM remain, and other SRAM and register data are lost.84-byte backup register hold. GPIO, IOM and EXTI are enabled.

1.3.1 Entering STOP2 mode

When entering STOP2 mode. The main difference is to set SLEEPDEEP= 1, PWR_CTRL2.STOP2S =1, PWR_CTRL bit PDS=0, LPS=0.

In STOP2 mode, if an operation is being performed on FLAH, the time to enter STOP2 mode is delayed until memory access is complete.

If access to the APB area is in progress, entering STOP2 mode will be delayed until the APB access is complete.

In STOP2 mode, the following features can be selected by programming the individual control bits:

- Independent watchdog: Independent watchdog is started during software writing or hardware operation of its associated register, and once started will work until a reset message is generated
- RTC: this can be enabled by the RTCEN bit in register RCC_BDCTRL
- Internal RC oscillator (LSI RC): can be turned on by the LSIEN bit in register RCC_CTRLSTS
- External 32.768khz crystal oscillator (LSE OSC): Can be turned on by the LSEEN bit in register RCC_BDCTRL

Note: If you want to keep data (global variables, stacks, etc.) in STOP2, you should put it in R-SRAM.

1.3.2 Exiting STOP2 mode

When exiting STOP2 mode by issuing interrupt or wake up events, select the HSI RC oscillator as the system clock. When STOP2 exits, the code resumes execution from where it left off.

1.4 STANDBY mode

The STANDBY mode achieves lower power consumption and is based on the Cortex®-M4 deep sleep mode. The core area is completely shut down and the backup area is turned on to power the VDD and BKR.

1.4.1 Entering STANDBY mode.

Enter STANDBY mode. The main difference is that the Settings SLEEPDEEP= 1, PDS=1.

In STANDBY mode, all I/O pins remain in the high resistance state except NRST, PA0_WKUP, PC13_TAMPER, PC14, and PC15.

If you are operating on FLAH, the time to enter STANDBY mode will be delayed until the memory access is complete.

If access to the APB area is in progress, entering STANDBY mode will be delayed until the APB access is complete.

In STANDBY mode, you can program each control bit to select the following features:

- Independent watchdog: Independent watchdog is activated during software writing or hardware operation of its related registers, and once activated, it always works
- Until a reset message is generated
- RTC: this can be enabled by the RTCEN bit in register RCC_BDCTRL
- Internal RC oscillator (LSI RC): can be turned on by the LSIEN bit in register RCC_CTRLSTS
- External 32.768khz crystal oscillator (LSE OSC): Can be turned on by the LSEEN bit in register RCC_BDCTRL
- R-SRAM data retention can be enabled by the SR2STBRET bit in register PWR_CTRL2

1.4.2 Exiting STANDBY mode

The device exits STANDBY mode when an external reset (NRST pin), IWDG reset, WKUP pin rise edge, or RTC alarm event rise edge occurs. All registers except the power control status register (PWR_CTRLSTS) are reset upon awakening from the STANDBY state.

After waking up from STANDBY mode, code execution is the same as after a reset (boot pins are fired, reset vectors are read, and so on).The SBF status flag in the power control status register (PWR_CTRLSTS) indicates that the MCU exits from standby mode.

1.5 VBAT mode

In VBAT mode the CPU is off, all peripherals are off, the main voltage regulator is off, LSE/LSI can be configured, HSE/HSI/PLL is off. Except NRST/ PC13-TAMper /PC14-OSC32_IN/PC15-OSC32_OUT, most I/O ports are in high resistance state.

In VBAT mode, you can use the following features based on the configuration before the VDD power failure:

- RTC: this can be enabled by the RTCEN bit in register RCC_BDCTRL
- Internal RC oscillator (LSI RC): can be turned on by the LSIEN bit in register RCC_CTRLSTS
- External 32.768khz crystal oscillator (LSE OSC): Can be turned on by the LSEEN bit in register RCC_BDCTRL
- R-SRAM data retention can be enabled by the SR2VBRET bit in register PWR_CTRL2

1.5.1 Entering VBAT mode

When the VDD is powered off, it enters VBAT mode at any time.

1.5.2 Exiting VBAT mode

When the VDD is restored to the power-on reset threshold, the device exits the VBAT mode. After the VDD is restored, the device core area will be powered on in sequence. After waking up from VBAT mode, code execution is equivalent to reset execution. The VBATF status flag in the power control status register (PWR_CTRLSTS) indicates that the MCU exits from VBAT mode.

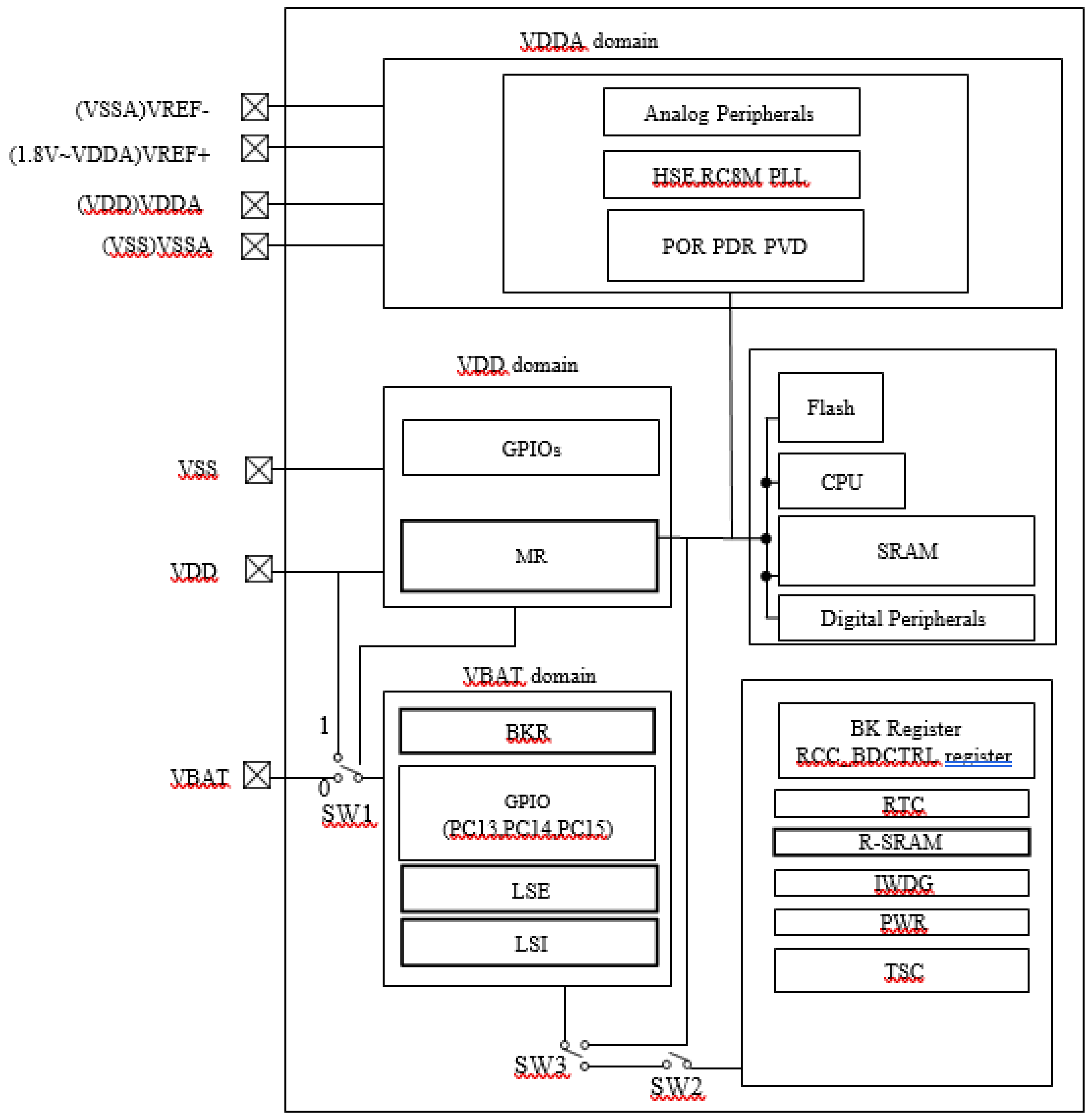
2 Power control (PWR)

2.1 Power System Introduction

N32A455 working voltage (VDD) is 1.8V~3.6V.It mainly has 3 analog/digital power regions (VDD, VBAT, VDDA).See Figure 2-1 power block diagram for details.

As the power control module of the whole device, the main function of the PWR is to control the device into different power modes and can be woken up by other events or interruptions. The N32A455 support RUN, SLEEP, STOP0, STOP2, STANDBY, and VBAT modes.

Figure 2-1 Power block



NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

2.1.1 Power Supply

In order to illustrate the functions of different power regions, some power regions are described below. The digital parts of the power regions are described in later chapters of this document , and the analog part is referred to the analog related design document.

- VDD domain: voltage input range is 1.8V~3.6V, mainly for MR power input, and for CPU, AHB, APB, SRAM, FLASH and most of the digital peripheral interface power.
- VBAT domain: The input voltage ranges from 1.8V to 3.6V. It supplies power to BKR and some special IO ports (PC13, PC14, PC15).When the VDD is powered off, the switch switches the power supply system VDD to VBAT.
- VDDA domain: Input voltage range 1.8V~3.6V, mainly for the clock and reset system, most analog peripherals power supply.

2.1.1.1 Digital module power supply system

The VDD and VBAT input voltages of N32A455 range from 1.8V to 3.6V. BKR and MR are internal voltage regulators that can supply power to the digital module power supply system. VDD and VBAT are powered by external power supply. VBAT is powered by batteries to keep the contents of the backup area, while VDD is powered by other external power supply systems. In addition, if the battery is not needed, the VBAT must be directly connected to the VDD.

- MR(RUN,SLEEP,STOP0)

MR is the internal main power controller, mainly used in RUN mode, SLEEP mode and STOP0 mode.MR has two modes, normal mode and low power mode. Low power mode is used for STOP0 to further reduce power consumption.

When the MR enters low power mode, the CPU goes into a deep sleep. The PWR_CTRL register should be set to PDS bit 0 and LPS bit 1.When MR enters normal mode, you need to set the PDS bit of the PWR_CTRL register to 0 and the LPS bit to 0.

- BKR(STOP2,STANDBY,VBAT)

The BKR is the internal backup power domain power controller and is used in STOP2, STANDBY, and VBAT modes. In STOP2 mode, the CPU state is maintained, and the digital backup area, GPIO, IOM and EXTI are additionally powered. When the CPU enters a deep sleep, set the PWR_CTRL2 register STOP2S bit to 1.

The main modules in the digital backup area include PWR, IO (PA0_WAKUP, PC13_TAMPER, PC14, PC15), R-SRAM, TSC, RTC, BKR, and RCC_BDCTRL registers. When SW3 is turned on, the CPU enters a deep sleep. When SW1 switches the power supply system to VBAT, VDD is powered off.

2.1.2 Backup Power Domain

During the reset, SW1 switches the power supply system to the VDD power supply area. In STOP2, STANDBY and VBAT modes, the internal voltage regulator BKR will supply power to the digital backup area.

Note:

- The switch between VBAT and VDD remains connected to the VBAT region during the VDD ascent phase or when PDR is detected.
- During the startup phase, if the VDD is quickly established and VDD > VBAT + 0.6V, current can be injected into the VBAT via an internal diode connection. If the power supply or battery connected to VBAT does not support this current injection. It is strongly recommended to place a low voltage diode between the power supply and the VBAT pin.

If there is no external battery in the application. It is recommended that the VBAT pins be connected to the VDD with a 100nF ceramic capacitor. In RUN, SLEEP, and STOP0 modes, the backup area is powered by the VDD (SW1 is connected to the VDD). The following functions are available:

- PC14 and PC15 can be used for normal IO ports or LSE pins
- PC13 can be used with common IO ports, TAMPER pins, RTC parity clock pins, RTC alarm clocks and second outputs

Note: Due to the fact that the current flowing through SW1 and SW2 is limited, the maximum is 3mA.So PA0_WAKUP, PC13 to PC15 IO output mode is limited, in the external 30PF capacitor, the maximum output speed is 2MHz.In addition, these IO cannot be driven by current, such as LED. The current of SW2 will be maintained at 3mA or lower because GPIO, IOM and EXTI work together to consume current. When VBAT supplies power for the backup area, the following functions can be used:

- PC14 and PC15 can only be used for LSE pins
- PC13 is used for TAMPER pins, RTC alarm clocks, or second outputs

2.2 Supply Current Characteristic

2.2.1 General Operating Conditions

Table 2-1 General Operating Conditions

Symbol	Parameter	Condition	Min	Max	Unit
f_{HCLK}	Internal AHB clock frequency	-	0	144	MHz
f_{PCLK1}	Internal APB1 clock frequency	-	0	36	
f_{PCLK2}	Internal APB2 clock frequency	-	0	72	
V_{DD}	Standard operating voltage	-	1.8	3.6	V
V_{DDA}	Analog operating voltage	Must be the same as V_{DD} (1)	1.8	3.6	V
V_{BAT}	Backup operating voltage	-	1.8	3.6	V
T_A	Ambient temperature (Temperature label 7)	Maximum power consumption	- 40	105	℃
	Ambient temperature (Temperature label 8)	Low power consumption(3)	- 40	125	
T_J	Junction temperature range	Temperature label 7	- 40	125	℃
		Temperature label 7	- 40	125	℃

1. It is recommended to use the same power supply for V_{DD} and V_{DDA} Power supply, during power up and normal operation, V_{DD} and V_{DDA} maximum of 300mV difference is allowed between.

Current consumption is a comprehensive index of a variety of parameters and factors. These parameters and factors include operating voltage, ambient temperature, I/O pin load, product software configuration, operating frequency, I/O pin flip rate, and program in memory The location in and the executed code, etc.

Current consumption measurement method description, see 2.2.2.

All of the current consumption measurements given in this section are performed in a simplified set of code that is equivalent to the Dhrystone 2.1 code.

2.2.2 Maximum Current Consumption

The microcontroller is under the following conditions:

- All I/O pins are in input mode and connected to a static level V_{DD} or V_{SS}(No load).
- All peripherals are off unless otherwise noted.
- Flash memory access time adjusted to F_{HCLK}(0~32MHz: 0 waiting periods, 32~64MHz: 1 waiting period, 64~96 MHz: 2 waiting periods, 96~128MHz: 3 waiting periods, 128~144MHz: 4 waiting periods).
- Instruction prefetch is enabled (note: this parameter must be set before setting the clock and bus divider).
- When the peripheral is turned on: $f_{PCLK1} = f_{HCLK}/4$, $f_{PCLK2} = f_{HCLK}/2$.

The parameters given in Table 2-2 and Table 2-3 are based on tests at ambient temperature and VDD supply voltage listed in Table 2-1.

Table 2-2 Maximum current consumption in RUN mode, data processing code is run from internal flash

Symbol	Parameter	Condition	f_{HCLK}	Typ [(1)]		Unit
				T_A = 105 ℃	T_A = 125 ℃	
I_{DD}	Supply current in RUN mode	External clock[(2)]. Enable all peripherals	144MHz	30.5	31.8	mA
			72MHz	17.3	18.4	
			36MHz	10.7	11.5	
		External clock[(2)]. Disable all peripherals	144MHz	15.8	17.9	
			72MHz	9.9	11.3	
			36MHz	7.1	8.1	

1. Derived from comprehensive evaluation and not tested in production.
2. The external clock is 8MHz, PLL is enabled when f_{HCLK} >8MHz.

Table 2-3 Maximum current consumption in SLEEP mode, code running in Flash or RAM

Symbol	Parameter	Condition	f_{HCLK}	Typ [(1)]		Unit
				T_A = 105 ℃	T_A = 125 ℃	
I_{DD}	Supply current in SLEEP mode	External clock[(2)]. Enable all peripherals	144MHz	24.6	24.8	mA
			72MHz	14.3	14.4	
			36MHz	9.3	9.5	
		External clock[(2)]. Disable all peripherals	144MHz	9.4	10	
			72MHz	6.8	7.2	
			36MHz	5.5	6.2	

1. Derived from comprehensive evaluation, V_{DDmax} and $f_{HCLKmax}$ enable peripheral for conditional test.
2. The external clock is 8MHz, PLL is enabled when f_{HCLK} >8MHz.

Table 2-4 Typical and Maximum Current Consumption in Shutdown and Standby Mode

Symbol	Parameter	Condition	Typical[1]			Unit
			T_A = 25℃	T_A = 105℃	T_A = 125℃	
I_{DD}	Supply current in STOP0 mode	The regulator is in operation mode with low and high speed internal RC oscillators and high speed oscillators off (no independent watchdog)	300	1670	3000	uA
		The regulator is in low power mode with low and high speed internal RC oscillators and high speed oscillators off (no independent watchdog)	150	1160	2800	

Notes:

	Supply current in STOP2 mode	The external low speed clock is on, RTC is running, R-SRAM is on, all I/O states are on, and the independent watchdog is off	10	100	180	
	Supply current in STANDBY mode	Low speed internal RC oscillator and independent watchdog are on	3	40	65	
		The low speed internal RC oscillator is on and the independent watchdog is off	2.9	40	65	
		The low speed internal RC oscillator and independent watchdog are closed, and the low speed oscillator and RTC are closed	2.7	35	60	
I_{DD_VBAT}	Backup domain(VBAT) supply current	The low speed oscillator and RTC are on	2	18	40	

Notes:

1. *The typical value is tested at VDD/VBAT= 3.3V.*
2. *Derived from comprehensive evaluation and not tested in production.*

2.2.3 Typical current consumption

MCU is under the following conditions:

- All I/O pins are in input mode and connected to a static level -- V_{DD} or V_{SS}(No load).
- All peripherals are off unless otherwise noted.
- Flash memory access time adjusted to F_{HCLK}(0~32MHz: 0 waiting periods, 32~64MHz: 1 waiting period, 64~96MHZ: 2 waiting periods, 96~128MHz: 3 waiting periods, 128~144MHz: 4 waiting periods).
- Ambient temperature and V_{DD} the supply voltage conditions are listed in Table 2-1.
- Instruction prefetch is enabled (note: this parameter must be set before setting the clock and bus divider).When the peripheral is turned on: $f_{PCLK1}= f_{HCLK}/4$, $f_{PCLK2} = f_{HCLK}/2$, $f_{ADCCLK} = f_{PCLK2}/4$.

Table 2-5 Typical current consumption in RUN mode, data processing code is run from internal Flash

Symbol	**Parameter**	**Conditions**	**f_{HCLK}**	**Typical values[1]**		**Unit**
				Enable all peripherals[2]	**Disable all peripherals**	
I_{DD}	Supply current in RUN mode	External clock[3]	144MHz	30.3	14.2	mA
			72MHz	17	8.1	
			36MHz	9.3	5.3	
		Run on high speed internal RC oscillator (HSI), using AHB predivision to reduce frequency	128MHz	29.3	12.7	mA
			72MHz	16.5	7.2	
			36MHz	8.8	3.9	

Notes:

1. *The typical value is at TA= 25 ℃, VDD=3.3V.*
2. *An additional 0.8mA of current consumption is added to each analog portion of the ADC. In the application environment, this part of the current is increased only when the ADC is turned ON (setting the ON bit of the ADC_CTRL2 register).*
3. *The external clock is 8MHz, PLL is enabled when fHCLK >8MHz.*

Table 2-6 Typical current consumption in SLEEP mode, data processing code is run from internal Flash or RAM

Symbol	Parameter	Conditions	f_{HCLK}	Typical values[1]		Unit
				Enable all peripherals[2]	Close all peripherals	
I_{DD}	Supply current in SLEEP mode	External clock[3]	144MHz	25.5	8	mA
			72MHz	12.2	5.3	
			36MHz	7.2	3.6	
		Run on high speed internal RC oscillator (HSI), using AHB predivision to reduce frequency	128MHz	21.6	6.1	mA
			72MHz	11.3	3.5	
			36MHz	6.8	2.2	

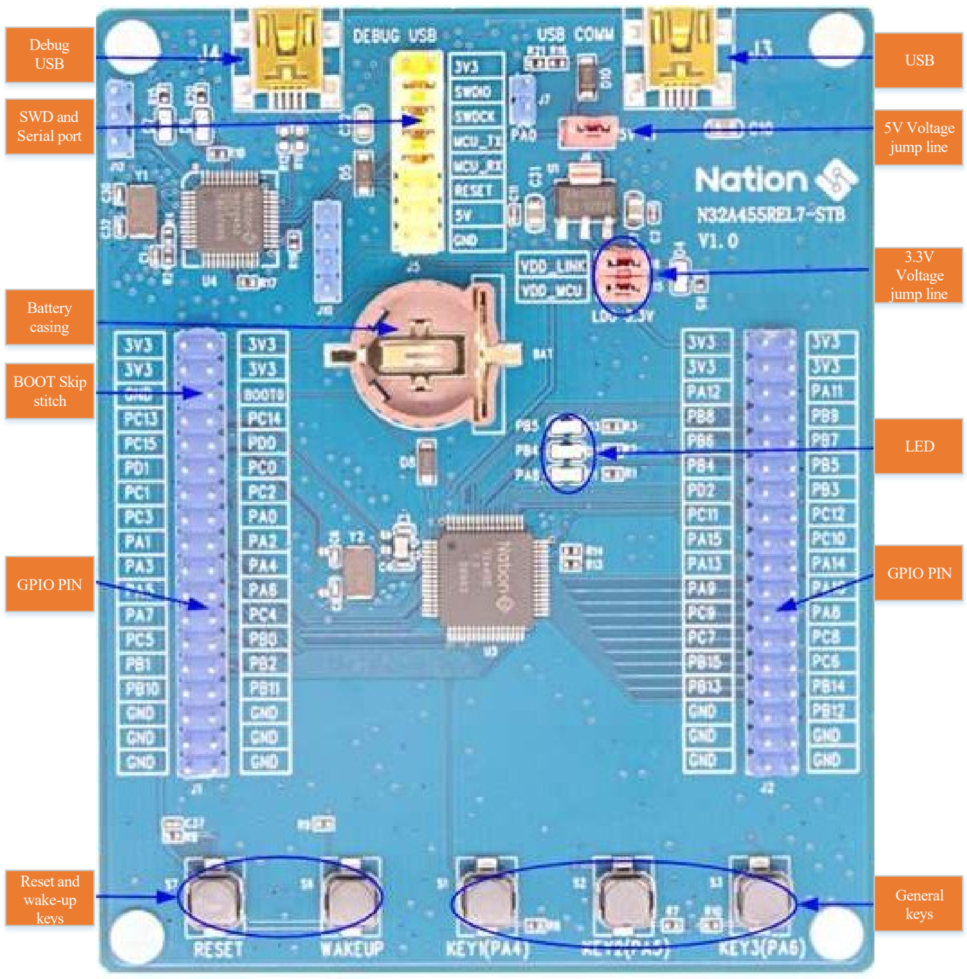
Notes:

1. *The typical value is at TA= 25 ℃, VDD=3.3V.*
2. *An additional 0.8mA of current consumption is added to each analog portion of the ADC. In the application environment, this part of the current is increased only when the ADC is turned ON (setting the ON bit of the ADC_CTRL2 register).*
3. *The external clock is 8MHz, PLL is enabled when fHCLK >8MHz.*

3 Hardware environment

3.1 Development board layout

Figure 3-1 Development Board Layout



1) Power supply for development board

The development board can be powered by USB interface (J3) and Debug USB (J4), connected to 3.3V LDO input port through J6 jumper.

2) USB port (J3)

Mini USB interface (J3), connected to MCU DP DM, can be used for USB interface communication.

3) Debug USB (J4)

The MCU can be used to download programs through the Debug USB or as a serial port.

4) SWD Interface (J5)

SWD interface can also be used for program download debugging, using ULINK2 or JLINK to download programs to the chip. You can also use the jumper to short SWDIO and SWDCK, and use the Debug USB to download programs.

5) Reset and wake up buttons (S7, S6)

S7 and S6 are reset buttons and wake buttons respectively, which are connected to NRST pins and PA0-WKUP pins of the chip respectively for chip reset and wake functions.

6) Universal keys (S1, S2, S3)

S1, S2 and S3 are respectively connected to PA4, PA5 and PA6 pins of the chip.

7) BOOT (J9,J11)

J9 and J11 are BOOT0 and BOOT1, respectively.

8) Battery holder (BAT)

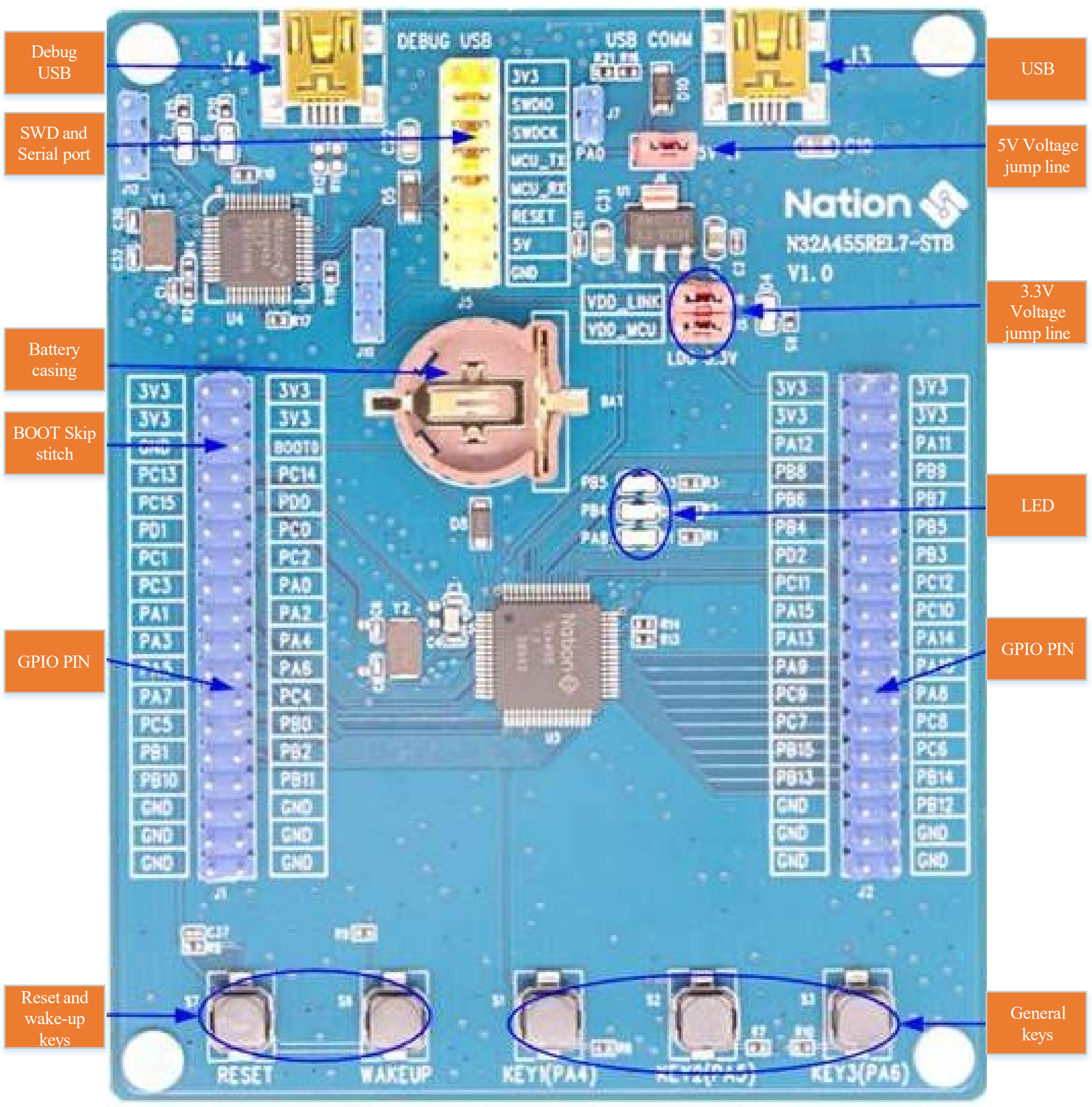
A CR1220 battery can be placed in the battery holder, which is connected to the chip VBAT pin to provide power.

9) GPIO port (J1, J2)

All the GPIO interfaces of the chip are elicited, and 3.3V voltage and GND pins are reserved on the pins for easy testing. See《CN_DS_N32A455 Series Datasheet V0.9.0 (125°C)》for the specific definition of the interface.

3.2 Development board jumper instructions

Figure 3-2 Development board jumper description



Jumper descriptions are as follows:

No.	Jump line item number	Jump line function	Directions
1	J6	5V voltage jumper	The J6 jumper connects the J3 and J4 USB ports and supplies power to the LDO3.3V input port.
2	J8 and J15	3.3V power supply jumper	J8: Supply 3.3V power to NS-LINK MCU chip. J15: Supply 3.3V power to main MCU chip.
3	J5	SWD jumper, Serial jumper	Use NS-LINK to download program to MCU through USB Debug port, need to short SWDIO and SWDCK pin.

			When using NS-LINK as a serial port through the USB Debug port, the two pins MCU_TX and MCU_RX need to be short-circuited.
4	J1	BOOT jumper	J1：BOOT0

3.3 Development board schematic diagram

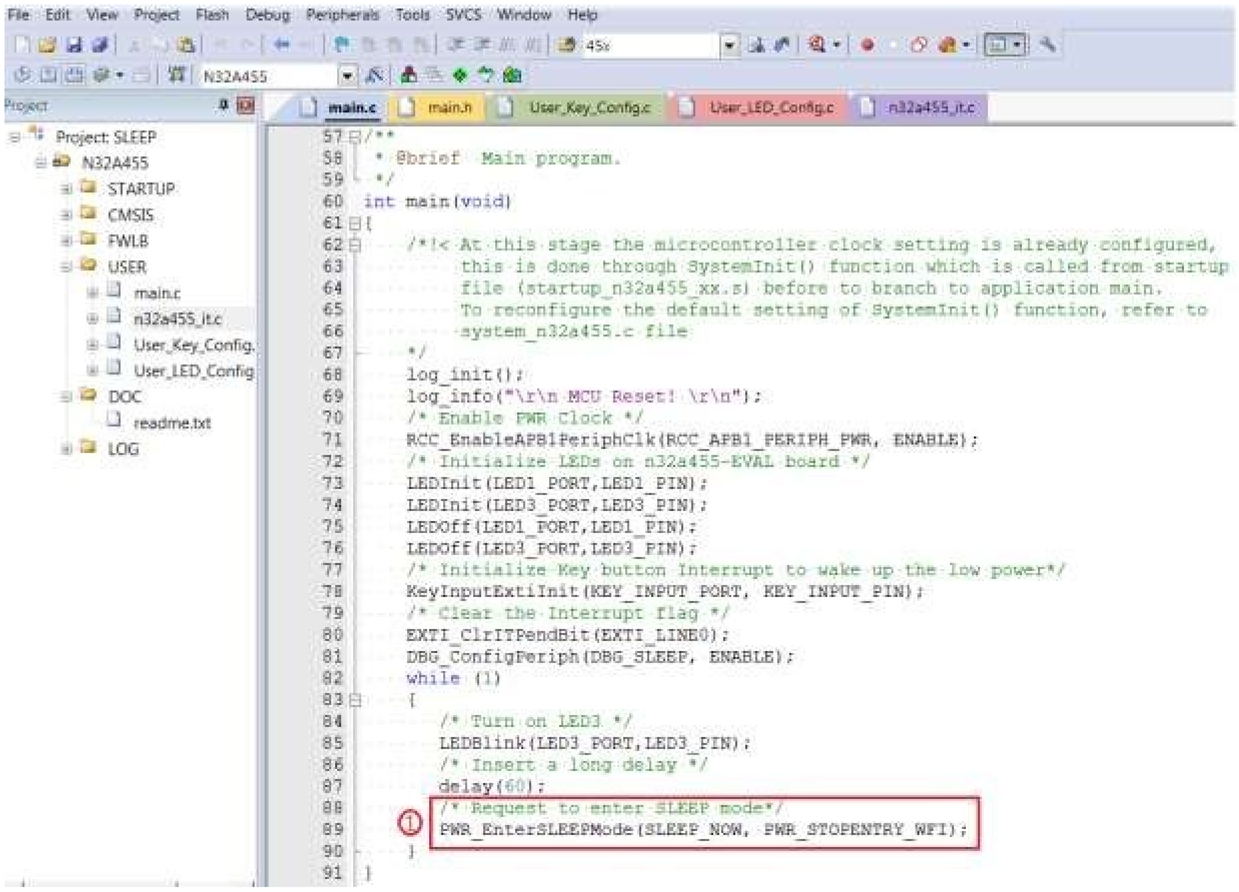
N32A455REL7-STB Development board schematic see《N32A455REL7-STB_V1.0》

4 Programming Instructions

4.1 Set SLEEP Mode

Open the SLEEP project in SDK. The part in circle ① in Figure 4-1 is the API function to enter SLEEP mode. After compiling, download it to the development board.

Figure 4-1 SLEEP entering Settings



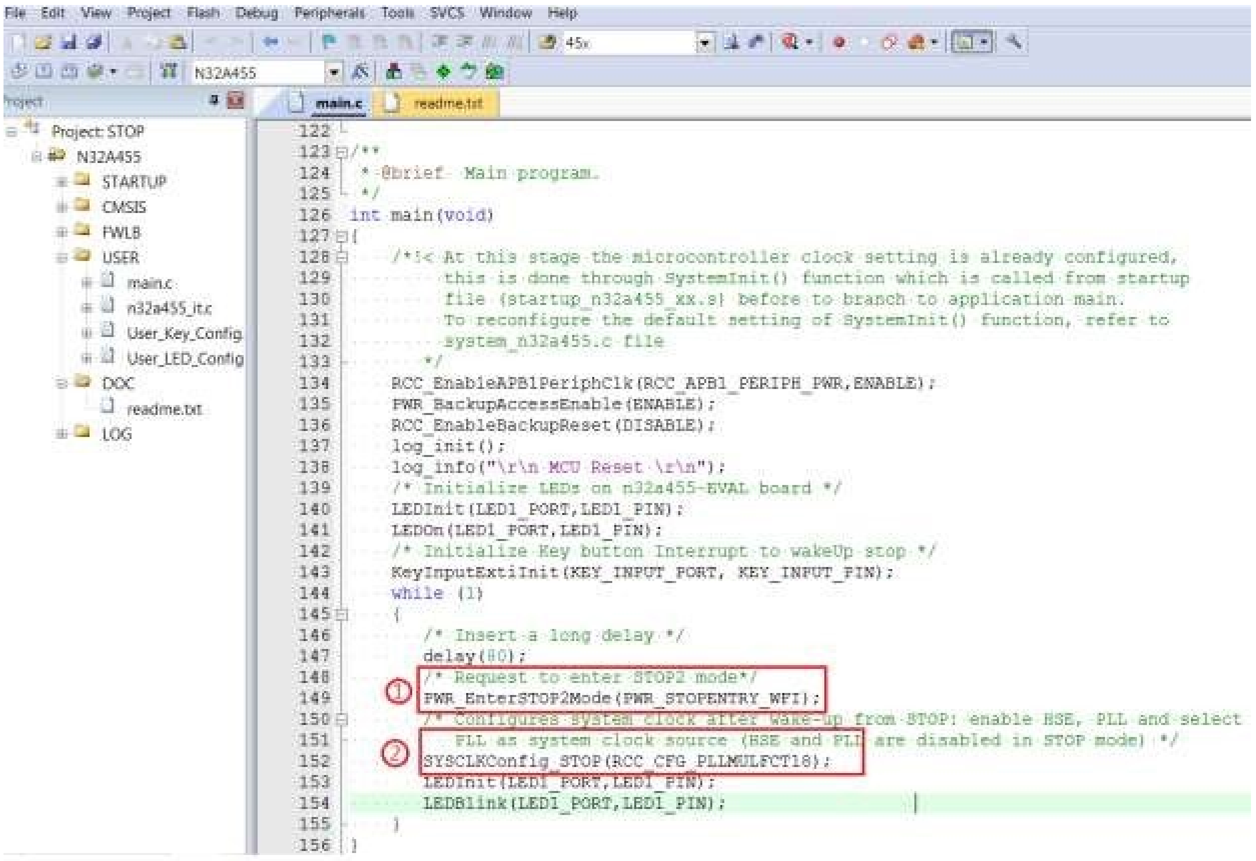
```
/**
 * @brief  Main program.
 */
int main(void)
{
    /*!< At this stage the microcontroller clock setting is already configured,
         this is done through SystemInit() function which is called from startup
         file (startup_n32a455_xx.s) before to branch to application main.
         To reconfigure the default setting of SystemInit() function, refer to
         system_n32a455.c file
    */
    log_init();
    log_info("\r\n MCU Reset! \r\n");
    /* Enable PWR Clock */
    RCC_EnableAPB1PeriphClk(RCC_APB1_PERIPH_PWR, ENABLE);
    /* Initialize LEDs on n32a455-EVAL board */
    LEDInit(LED1_PORT,LED1_PIN);
    LEDInit(LED3_PORT,LED3_PIN);
    LEDOff(LED1_PORT,LED1_PIN);
    LEDOff(LED3_PORT,LED3_PIN);
    /* Initialize Key button Interrupt to wake up the low power*/
    KeyInputExtiInit(KEY_INPUT_PORT, KEY_INPUT_PIN);
    /* Clear the Interrupt flag */
    EXTI_ClrITPendBit(EXTI_LINE0);
    DBG_ConfigPeriph(DBG_SLEEP, ENABLE);
    while (1)
    {
        /* Turn on LED3 */
        LEDBlink(LED3_PORT,LED3_PIN);
        /* Insert a long delay */
        delay(60);
        /* Request to enter SLEEP mode*/
        PWR_EnterSLEEPMode(SLEEP_NOW, PWR_STOPENTRY_WFI);
    }
}
```

4.2 Set STOP2 Mode

STOP2 mode is based on the Cortex-M4 deep sleep mode, and all the core digital logic areas are powered off. Main voltage regulator (MR) is off, HSE/HSI/PLL is off. CPU registers are maintained, LSE/LSI can be configured, GPIO is maintained, and peripheral I/O multiplexing is not maintained.16K bytes of R-SRAM remain, and other SRAM and register data are lost.84-byte backup register hold. GPIO , IOM and EXTI are enabled.

Open the STOP2 project in the SDK, the part circled in Figure 4-2 is the API function to enter STOP2, this function will set the interrupt to enter STOP2, the part ② in the circle in Figure 4-2 is to switch the system clock back to the system when exiting the STOP2 mode high-speed clock. This mode needs to pay attention to the change of the system clock, so the peripherals need to be reconfigured according to the actual clock source.

Figure 4-2 STOP2 enter Settings



```
/**
 * @brief  Main program.
 */
int main(void)
{
    /*!< At this stage the microcontroller clock setting is already configured,
         this is done through SystemInit() function which is called from startup
         file (startup_n32a455_xx.s) before to branch to application main.
         To reconfigure the default setting of SystemInit() function, refer to
         system_n32a455.c file
       */
    RCC_EnableAPB1PeriphClk(RCC_APB1_PERIPH_PWR,ENABLE);
    PWR_BackupAccessEnable(ENABLE);
    RCC_EnableBackupReset(DISABLE);
    log_init();
    log_info("\r\n MCU Reset \r\n");
    /* Initialize LEDs on n32a455-EVAL board */
    LEDInit(LED1_PORT,LED1_PIN);
    LEDOn(LED1_PORT,LED1_PIN);
    /* Initialize Key button Interrupt to wakeUp stop */
    KeyInputExtiInit(KEY_INPUT_PORT, KEY_INPUT_PIN);
    while (1)
    {
        /* Insert a long delay */
        delay(80);
        /* Request to enter STOP2 mode*/
        PWR_EnterSTOP2Mode(PWR_STOPENTRY_WFI);
        /* Configures system clock after wake-up from STOP: enable HSE, PLL and select
           PLL as system clock source (HSE and PLL are disabled in STOP mode) */
        SYSCLKConfig_STOP(RCC_CFG_PLLMULFCT18);
        LEDInit(LED1_PORT,LED1_PIN);
        LEDBlink(LED1_PORT,LED1_PIN);
    }
}
```

4.3 Set STOP0 Mode

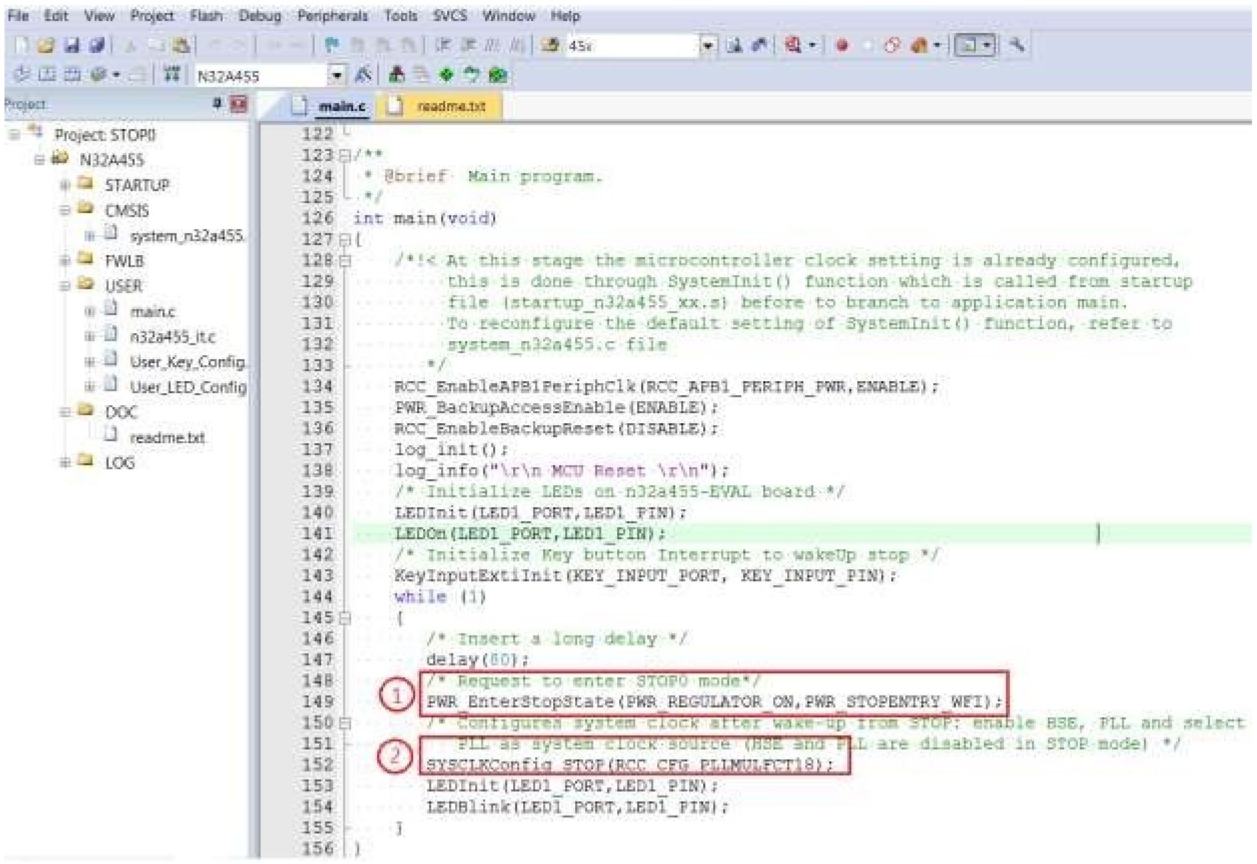
STOP0 mode is based on the Cortex®-M4 deep sleep mode, combined with the peripheral clock control mechanism. The voltage regulator can be configured for normal or low power mode. In STOP0 mode, most clock sources in the core domain, such as PLL, HSI, and HSE, are disabled. But SRAM, R-SRAM, and all register contents are saved.

In STOP0 mode, all I/O pins remain in the same state as in run mode.

Open the STOP0 project in the SDK, the part circled in Figure 4-3 is the API function to enter STOP0, this function will set the interrupt to enter STOP0, the part ② in the circle in Figure 4-3 is to switch the system clock back to the system when exiting the STOP0 mode high-speed clock. This mode needs to pay attention to the change of the system clock, so the peripherals need to be reconfigured according to the actual clock source.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Figure 4-3 STOP0 entering Settings



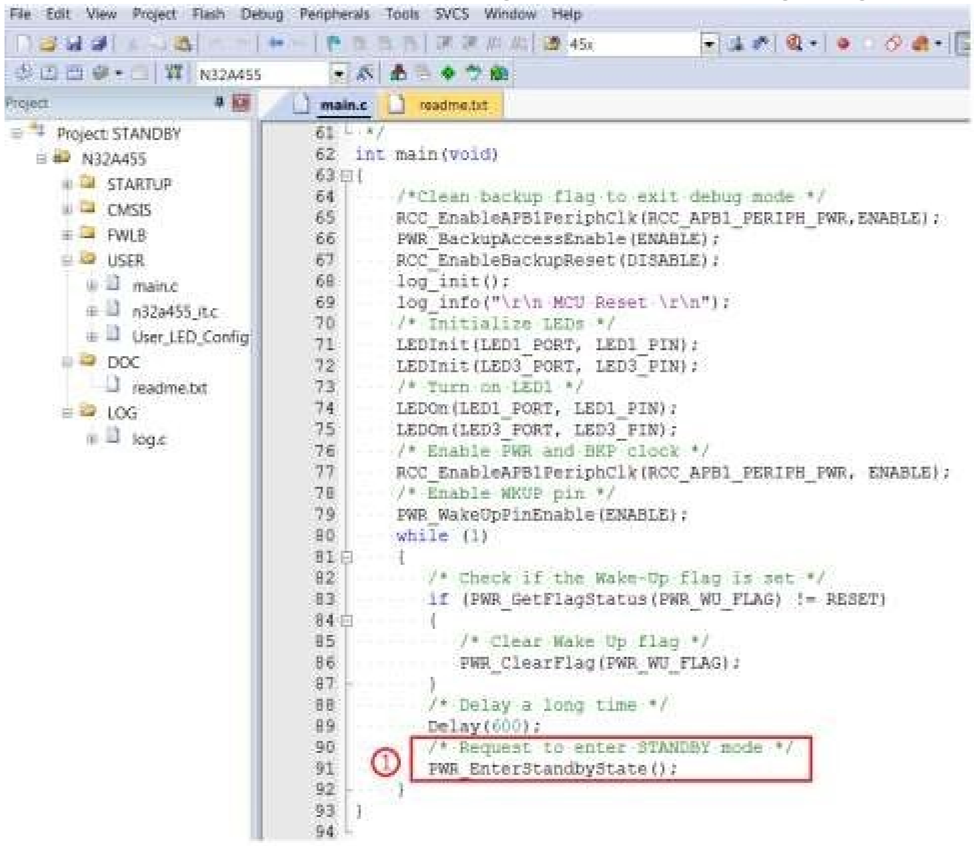
```c

/**
 * @brief  Main program.
 */
int main(void)
{
    /*!< At this stage the microcontroller clock setting is already configured,
         this is done through SystemInit() function which is called from startup
         file (startup_n32a455_xx.s) before to branch to application main.
         To reconfigure the default setting of SystemInit() function, refer to
         system_n32a455.c file
       */
    RCC_EnableAPB1PeriphClk(RCC_APB1_PERIPH_PWR,ENABLE);
    PWR_BackupAccessEnable(ENABLE);
    RCC_EnableBackupReset(DISABLE);
    log_init();
    log_info("\r\n MCU Reset \r\n");
    /* Initialize LEDs on n32a455-EVAL board */
    LEDInit(LED1_PORT,LED1_PIN);
    LEDOn(LED1_PORT,LED1_PIN);
    /* Initialize Key button Interrupt to wakeUp stop */
    KeyInputExtiInit(KEY_INPUT_PORT, KEY_INPUT_PIN);
    while (1)
    {
        /* Insert a long delay */
        delay(60);
        /* Request to enter STOP0 mode*/
        PWR_EnterStopState(PWR_REGULATOR_ON,PWR_STOPENTRY_WFI);
        /* Configures system clock after wake-up from STOP: enable HSE, PLL and select
           PLL as system clock source (HSE and PLL are disabled in STOP mode) */
        SYSCLKConfig_STOP(RCC_CFG_PLLMULFCT18);
        LEDInit(LED1_PORT,LED1_PIN);
        LEDBlink(LED1_PORT,LED1_PIN);
    }
}
```

4.4 Set STANDBY Mode

STANDBY mode can achieve lower power consumption, it is based on Cortex®-M4 deep sleep mode, the core domain is completely turned off, and the backup power domain is turned on to supply power to VDD and BKR.

Open the STANDBY project in the SDK, the part circled in Figure 4-4 is the API function to enter STANDBY, this function will set the interrupt to enter STANDBY.

Figure 4-4 STANDBY entering Settings



```c
 */
int main(void)
{
    /*Clean backup flag to exit debug mode */
    RCC_EnableAPB1PeriphClk(RCC_APB1_PERIPH_PWR,ENABLE);
    PWR_BackupAccessEnable(ENABLE);
    RCC_EnableBackupReset(DISABLE);
    log_init();
    log_info("\r\n MCU Reset \r\n");
    /* Initialize LEDs */
    LEDInit(LED1_PORT, LED1_PIN);
    LEDInit(LED3_PORT, LED3_PIN);
    /* Turn on LED1 */
    LEDOn(LED1_PORT, LED1_PIN);
    LEDOn(LED3_PORT, LED3_PIN);
    /* Enable PWR and BKP clock */
    RCC_EnableAPB1PeriphClk(RCC_APB1_PERIPH_PWR, ENABLE);
    /* Enable WKUP pin */
    PWR_WakeUpPinEnable(ENABLE);
    while (1)
    {
        /* Check if the Wake-Up flag is set */
        if (PWR_GetFlagStatus(PWR_WU_FLAG) != RESET)
        {
            /* Clear Wake Up flag */
            PWR_ClearFlag(PWR_WU_FLAG);
        }
        /* Delay a long time */
        Delay(600);
        /* Request to enter STANDBY mode */
        PWR_EnterStandbyState();
    }
}
```

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

5 Version History

Version	Date	Changes
V1.0	2022-6-20	Initial version

6 Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD.(Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to Nations Technologies Inc. and Nations Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.