

MSP

ABSTRACT

This document introduces the low-power mode of MSPM0, takes measurement of system and peripherals power consumption data based on LP-MSPM0G3507, and gives the guidance about entering and exiting the low-power mode from different peripherals.

Table of Contents

1 MSPM0 Low Power Mode Introduction	3
1.1 MSPM0 Power Domain Introduction.....	3
1.2 MSPM0 Low-Power Mode Introduction.....	4
1.3 Power Consumption Measurement Based on EnergyTrace™.....	6
2 MSPM0 System Power Consumption Measurement	8
3 MSPM0 Peripheral Power Consumption Measurement	9
3.1 GPIO.....	9
3.2 RTC.....	10
3.3 WWDT.....	11
3.4 TIMER.....	11
3.5 UART.....	13
3.6 I2C.....	14
3.7 SPI.....	15
3.8 MCAN.....	15
3.9 SYSPLL.....	16
3.10 VREF.....	17
3.11 ADC.....	18
4 Low-Power Mode Guidance	19

List of Figures

Figure 1-1. MSPM0G3507 System Diagram.....	3
Figure 1-2. MSPM0G Series Operation Modes.....	4
Figure 1-3. Operation Mode Policy Bit Configuration.....	5
Figure 1-4. EnergyTrace Configuration.....	6
Figure 1-5. EnergyTrace Configuration of XDS110.....	6
Figure 1-6. EnergyTrace Tool Introduction in CCS.....	7
Figure 3-1. GPIO Unused Pins Configuration in Sysconfig.....	10
Figure 3-2. RTC Power Consumption Test Under STANDBY1 Mode.....	10
Figure 3-3. WWDT Power Consumption Test Under STANDBY0 Mode.....	11
Figure 3-4. TIMG0 Power Consumption Test Under STANDBY1 Mode.....	11
Figure 3-5. TIMA0 Power Consumption Test Under RUN0 Mode.....	12
Figure 3-6. UART0 Power Consumption Test Under STANDBY1 Mode.....	13
Figure 3-7. I2C Power Consumption Test Under STANDBY1 Mode.....	14
Figure 3-8. SPI Power Consumption Test Under STANDBY1 Mode.....	15
Figure 3-9. SYSPLL Configuration.....	16
Figure 3-10. VREF Power Consumption Test under STANDBY1 Mode.....	17
Figure 3-11. ADC Power Consumption Test Under STANDBY1 Mode.....	18

List of Tables

Table 2-1. Low-Power Mode Power Consumption Test Table.....	8
Table 3-1. Peripheral Power Consumption Table.....	9

Table 3-2. GPIO Power Consumption Test Table.....	9
Table 3-3. RTC Power Consumption Test Table.....	10
Table 3-4. RTC Power Consumption Test Table.....	11
Table 3-5. TIMG0 on LFCLK Power Consumption Test Table.....	11
Table 3-6. TIMA0 Power Consumption Test Table.....	12
Table 3-7. UART0 Power Consumption Test Table.....	13
Table 3-8. I2C Power Consumption Test Table.....	14
Table 3-9. SPI Power Consumption Test Table.....	15
Table 3-10. MCAN Power Consumption Test Table.....	15
Table 3-11. SYSPLL Power Consumption Test Table.....	16
Table 3-12. VREF Power Consumption Test Table.....	17
Table 3-13. ADC Power Consumption Test Table.....	18
Table 4-1. Peripheral Enter/Exit Low-Power Mode Guidance.....	19

Trademarks

EnergyTrace™ and Code Composer Studio™ are trademarks of Texas Instruments.
 All trademarks are the property of their respective owners.

1 MSPM0 Low Power Mode Introduction

1.1 MSPM0 Power Domain Introduction

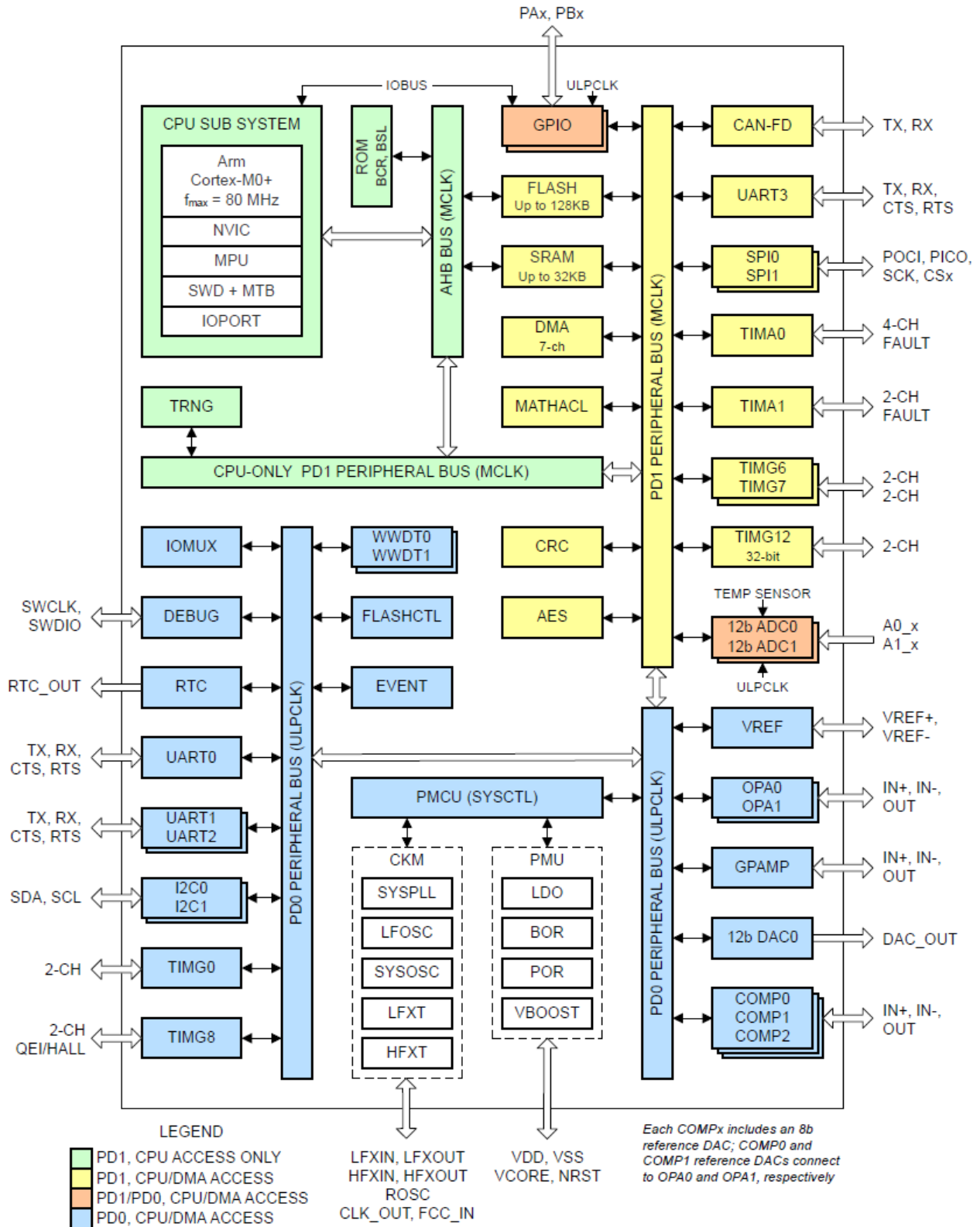


Figure 1-1. MSPM0G3507 System Diagram

There are two core power domains in MSPM0 devices: PD1 and PD0.

As shown in [Figure 1-1](#), the PD1 domain includes the CPU subsystem, the SRAM memory, PD1 peripherals, and the PD1 peripheral bus, which runs from MCLK (including the DMA) with a maximum frequency of 80MHz. The peripherals on PD1 peripheral bus are commonly high-speed peripherals, and usually consume high power while working. The PD0 domain includes the PD0 peripherals and PD0 bus segment, which runs from ULPCLK and is connected to low-speed peripherals, such as universal asynchronous receiver/transmitter (UART), inter-integrated circuit (I2C), real-time clock (RTC).

MSPM0 MCUs implement a policy-based power and clock management scheme. And in certain low-power mode, PD1 can be disabled to minimize power consumption and the peripherals connected are also disabled.

1.2 MSPM0 Low-Power Mode Introduction

There are five power modes in MSPM0 devices: RUN, SLEEP, STOP, STANDBY and SHUTDOWN. [Figure 1-2](#) shows the interaction between the modes.

PD1 is only enabled in RUN and SLEEP mode. While PD1 is disabled in STOP and STANDBY mode, the CPU registers, SRAM, and peripheral MMR configuration registers are maintained in retention such that these modules are available to resume operation immediately when STOP or STANDBY modes are exited.

PD0 is powered in all modes except SHUTDOWN mode and can be thought of as an "always-on" domain. PD0 runs from ULPCLK with a max frequency of 40MHz in RUN and SLEEP mode, 4MHz in STOP mode, and 32kHz in STANDBY mode.

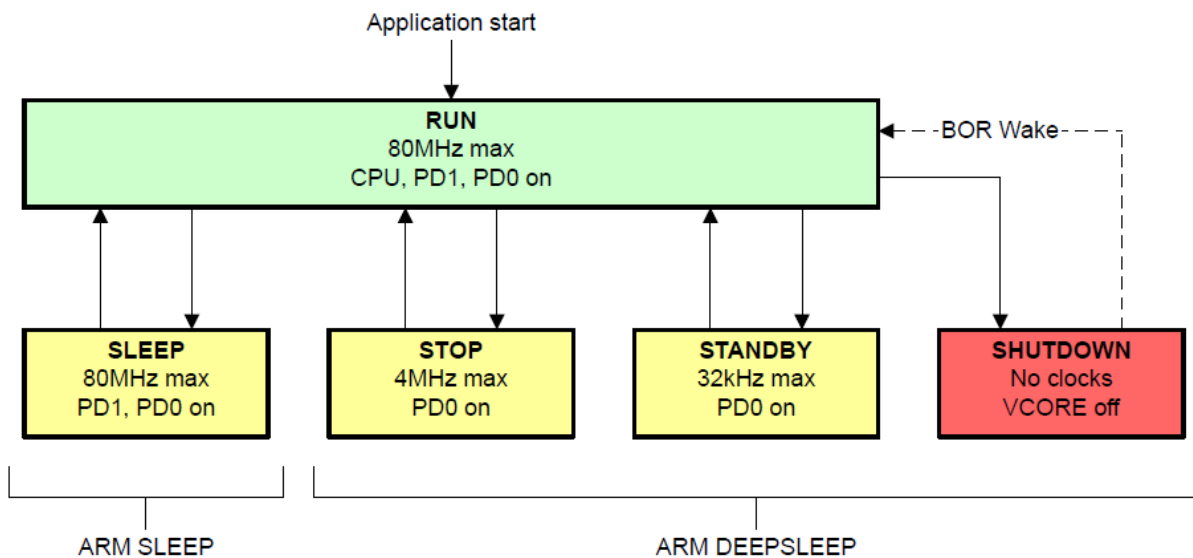


Figure 1-2. MSPM0G Series Operation Modes

[Figure 1-3](#) defines how to configure the relevant policy bits for each operating mode. All values are indicated in binary format. A dash (-) indicates that the particular policy bit is a don't care for the specified operating mode. For the detailed description of these low power modes, please refer to [2.1 PMCU Overview of MSPM0 G-Series 80MHz Microcontrollers Technical Reference Manual \(Rev. A\) \(ti.com\)](#).

1.3 Power Consumption Measurement Based on EnergyTrace™

1.3.1 EnergyTrace Introduction

The XDS110 debug probe has on-board circuitry that can be used for measuring the target's energy consumption. The hardware circuitry provides high-accuracy energy consumption with low bandwidth current and power profile. The energy profiling range covers 1µA to 100mA current draw, above which the tool can display an overcurrent message and shutdown. This tool is designed for characterizing energy consumption, but not for capturing short current spikes, because sampling occurs over large time windows (about 500µs).

For more details of EnergyTrace introduction, see 3.6 *Energy Trace* of [XDS110 Debug Probe \(ti.com.cn\)](#).

1.3.2 EnergyTrace Configuration With CCS

To start the EnergyTrace tool automatically on launching debug session, follow the steps of “Window - Preferences - Code Composer Studio™ - EnergyTrace Technology” in CCS. Enable the checkbox for “Enable Auto-launch on target connect”.

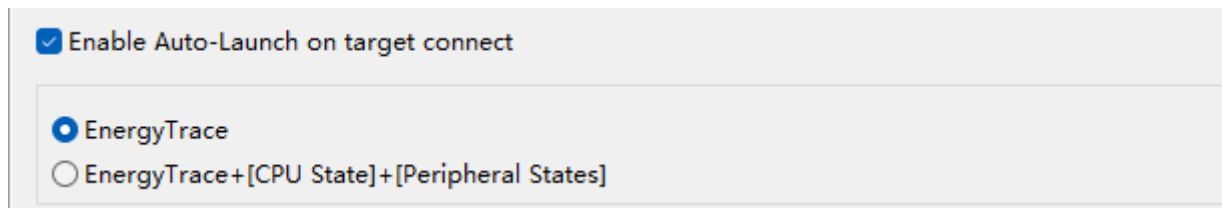


Figure 1-4. EnergyTrace Configuration

Alternatively, if the EnergyTrace tool must be started after establishing a debug connection, then when the core is connected, set the target connection as XDS110.

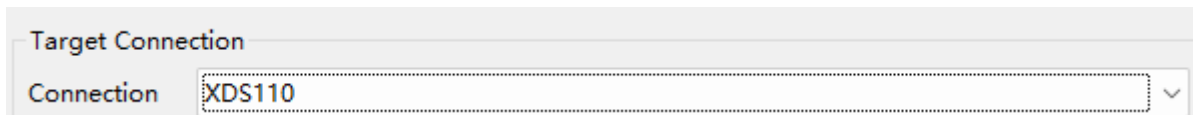


Figure 1-5. EnergyTrace Configuration of XDS110

And, [Figure 1-6](#) shows the fast launch method of EnergyTrace tool embedded in CCS. In this way, the current that XDS110 debug probe supplies to MCU system board can be measured and shown in “Current” windows. The data can be also exported, and analyzed external.

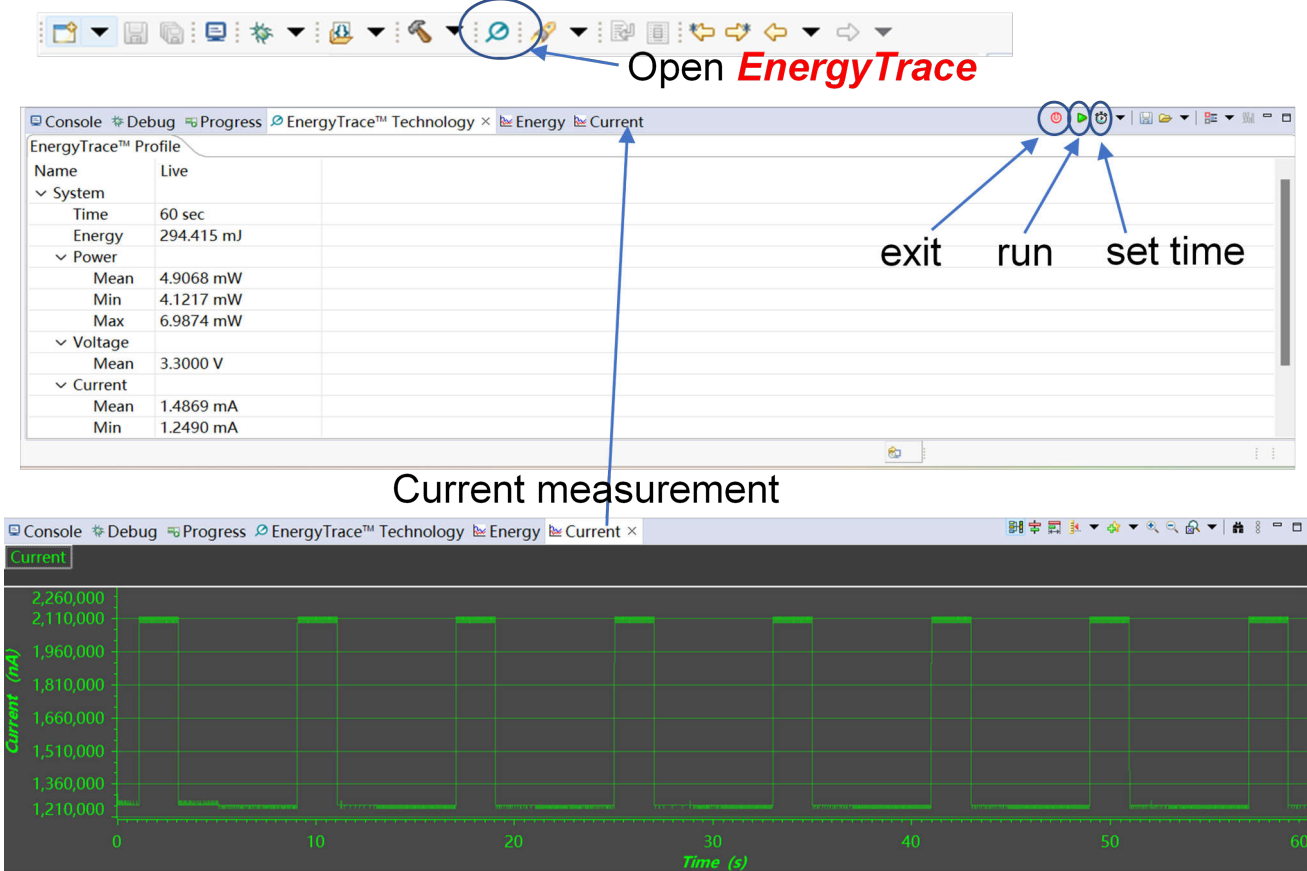


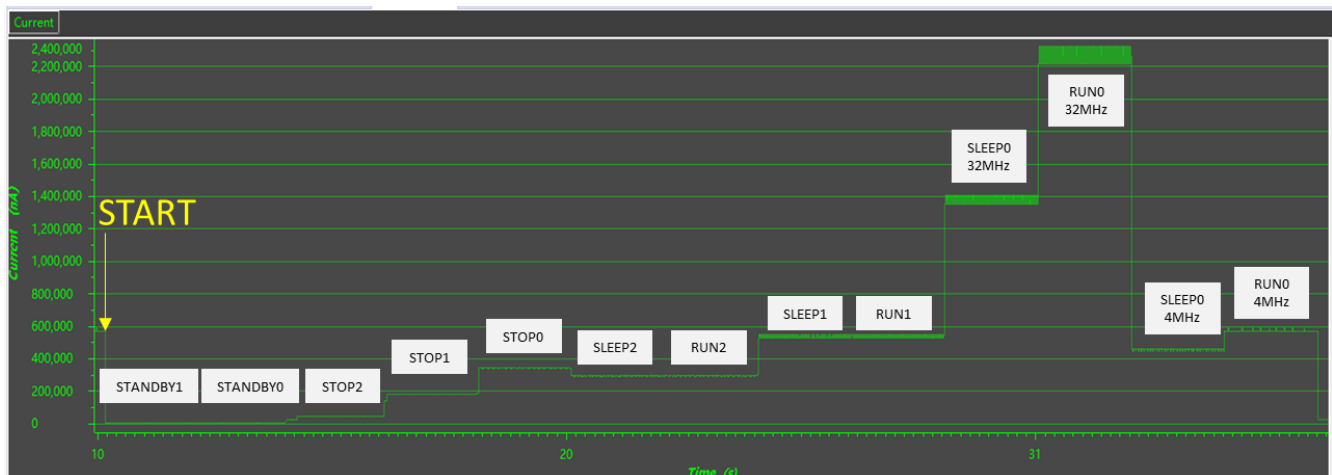
Figure 1-6. EnergyTrace Tool Introduction in CCS

2 MSPM0 System Power Consumption Measurement

Table 2-1 shows the hardware/software configuration and test results of MSPM0G350x system power consumption measurement under different low-power mode. The power consumption of MSPM0 without any peripheral enabled is strongly related to the clock configuration. For reference, SYSOSC, MCLK, ULPClk is listed under a different low-power policy.

Table 2-1. Low-Power Mode Power Consumption Test Table

Low-Power Mode	Configuration				Current
	Low-Power Policy	SYSOSC	MCLK	ULPClk	
Hardware	<ul style="list-style-type: none"> LP-MSPM0G3507 Keep J101 VDD, GND, SWCLK, SWIO connected Disconnect all other Jumpers in LP 				
Software	<ul style="list-style-type: none"> Disable all unused peripherals Disable external oscillator and clock output TIMG0 enters interrupt every 2s Low-power mode is switched after TIMG0 interrupt SYSOSC is initially configured as 32MHz 				
Low Power Policy	Standby1 + WFI	DIS	DIS	32kHz	1.5uA
	Standby0 + WFI	DIS	DIS	32kHz	1.9uA
	Stop2 + WFI	DIS	DIS	32kHz	46uA
	Stop1 + WFI	4MHz	DIS	4MHz	182uA
	Stop0 + WFI	32MHz	DIS	4MHz	346uA
	RUN2SLEEP2 + WFI	DIS	32kHz	32kHz	293uA
	RUN2SLEEP2 + while (1) {}	DIS	32kHz	32kHz	294uA
	RUN1SLEEP1 + WFI	32MHz	32kHz	32kHz	535uA
	RUN1SLEEP1+ while (1) {}	32MHz	32kHz	32kHz	536uA
	RUN0SLEEP0 + WFI (SYSOSC=32MHz)	32MHz	32MHz	32MHz	1.36mA
	RUN0SLEEP0 + while (1) {} (SYSOSC=32MHz)	32MHz	32MHz	32MHz	2.22mA
	RUN0SLEEP0 + WFI (SYSOSC=4MHz)	4MHz	4MHz	4MHz	456uA
	RUN0SLEEP0 + while (1) {} (SYSOSC=4MHz)	4MHz	4MHz	4MHz	569uA



There are several conclusions according to the power consumption test results above:

- The RUN mode and SLEEP mode configurations are valid without *WFI/WFE* instruction;
- When CPU runs at frequency of 32kHz, the RUN1/RUN2 mode has close power consumption with SLEEP1/ SLEEP2 mode because of low running clock frequency.

3 MSPM0 Peripheral Power Consumption Measurement

Table 3-1 summarizes key peripheral power consumption data of MSPM0G350x. For details of each peripheral power test performance, see the following sections.

The following provides an explanation of the two-power value in the table:

- LPM Power: The power consumption current with different peripheral enabled in low-power mode (STANDBY0/1 mode).
- Static Power in RUN0: The power consumption current difference after disabling a peripheral power or module in RUN0 mode.

Table 3-1. Peripheral Power Consumption Table

Peripherals		Clock	Power Consumption	
		SYSOSC = 32MHz	LPM Power	Static Power in RUN0
PD0	UART	ULPCLK=32MHz	3.5uA	80uA
	I2C	ULPCLK=32MHz	2.5uA	230uA
	TIMG0	LFCLK=32kHz	<0.1uA (Running in STOP/STANDBY mode)	
	RTC/WWDT	LFCLK=32kHz		
PD1	TIMA0	MCLK=32MHz	30uA (Static Power in RUN0 mode)	
	SPI	MCLK=32MHz	4.1uA	170uA
	MCAN	HFCLK=40MHz	10.6uA	480uA
Analog	SYSPLL	SYSPLL2x =32MHz	38uA (LPM Power in STANDBY1 mode)	
	VREF	MCLK=32MHz	9.0uA	80uA
	ADC	MCLK=32MHz	2.5uA	60uA

The following sectors give the power consumption test results of different peripherals. The common hardware and software configuration are shown as below and the following sectors only point the specific configuration of corresponding peripheral:

- Hardware
 - Based on LP-MSPM0G3507
 - Keep all jumpers uninstalled except J101 GND, 3V3, SWDIO, SWDCLK
 - No pullup/pulldown resistor connected to GPIO pins on the launchpad
- Software
 - Disable all peripherals unused
 - LFCLK = LFSOC, MCLK = SYSOSC (32MHz)
 - Enable TIMG0(clock from LFCLK) to switch running condition every 2s

3.1 GPIO

Table 3-2. GPIO Power Consumption Test Table

GPIO	Configuration	Current
Hardware	• The resistor connected to GND in PA21 is removed	
Test condition	Not configuring unused pins	1.8uA
	Configuring unused pins as output low	1.8uA
	Configuring unused pins as input with pulldown resistor	1.8uA
	Configuring unused pins as input without internal resistor	4.0mA

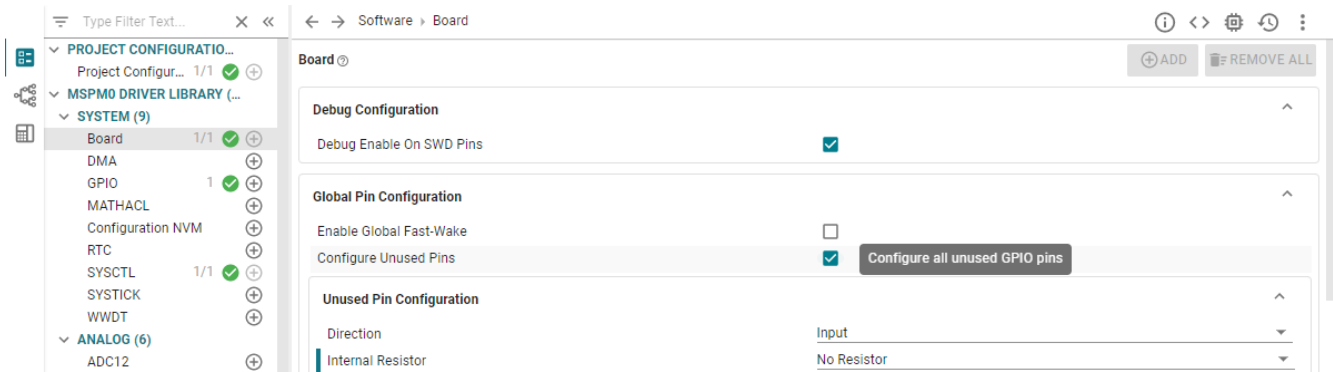


Figure 3-1. GPIO Unused Pins Configuration in Sysconfig

According to the table, when configuring pins as input without internal resistor, there is large power consumption unexpected. So for the pins unused, the key point to decrease the power waste is not to configure these pins as floating input, so it is a valid way to make the input GPIO function to be pull down internal or external.

When pins are used in peripherals, especially as input pins, the floating of an input pins brings a lot unexpected power consumption or even make the peripheral active in low-power mode. It is also recommended to configured the peripheral input pin to pullup or pulldown according.

For any pin in MSPM0, a valid way to disconnect the pin to other parts is configuring the pin as analog input by `DL_GPIO_initPeripheralAnalogFunction()`. In this case, the pin is not configured as input or output, and only leaves the connecting path to analog peripherals.

3.2 RTC

Figure 3-2 shows RTC consumes nearly no power under STANDBY1 mode (same as STANDBY0 mode).

Table 3-3. RTC Power Consumption Test Table

RTC	Configuration	Current
Software	• RTCCLK = LFCLK = 32kHz	
Test condition	Enable RTC, WFI (STANDBY1)	2.0uA
	Disable RTC, WFI (STANDBY1)	2.0uA

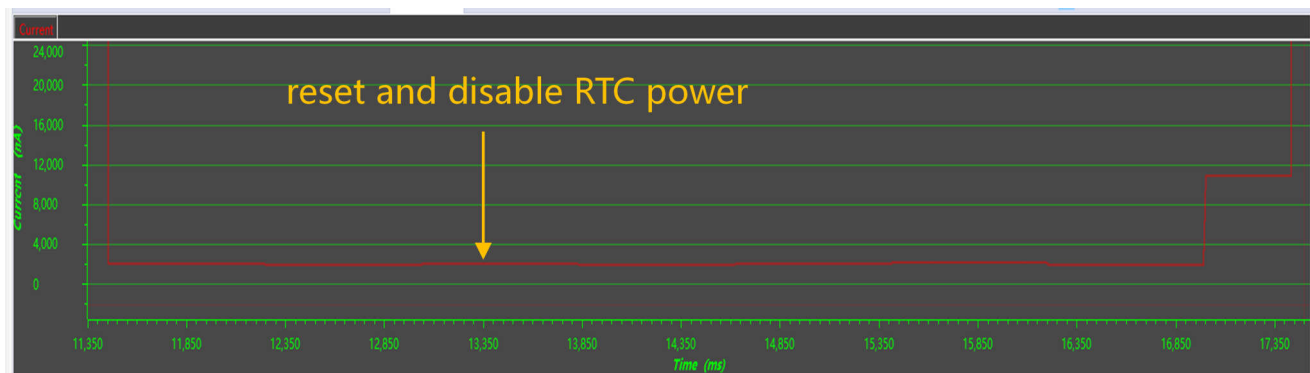


Figure 3-2. RTC Power Consumption Test Under STANDBY1 Mode

3.3 WWDT

Figure 3-3 shows WWDT consumes nearly no power (<0.1uA) under STANDBY0 mode, as STANDBY0 is the lowest power mode for WWDT working.

Table 3-4. RTC Power Consumption Test Table

WWDT	Configuration	Current
Software	<ul style="list-style-type: none"> WWDT period is configured as 500ms; Enable TIMG0(clock from LFCLK) to restart WWDT every 200ms; 	
Test condition	Enable WWDT, WFI (STANDBY0)	2.7uA
	Disable WWDT, WFI (STANDBY0)	2.6uA

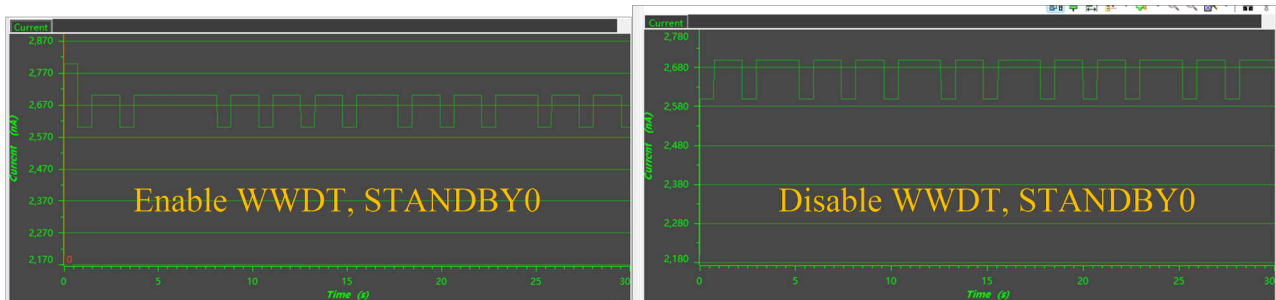


Figure 3-3. WWDT Power Consumption Test Under STANDBY0 Mode

3.4 TIMER

3.4.1 TIMG0

Figure 3-4 shows TIMG0/8 sourced by LFCLK consumes nearly no power under STANDBY1 mode. Combining the results of RTC and WWDT power test, the peripherals driven by LFCLK consume little power when running. And it is also the reason why TIMG0 is used on LFCLK as the timer to switch power policy in other tests.

Table 3-5. TIMG0 on LFCLK Power Consumption Test Table

TIMG0 on LFCLK	Configuration	Current
Software	<ul style="list-style-type: none"> TIMG0(clock from LFCLK=32kHz) period is configured as 2s; 	
Test condition	Enable TIMG0, WFI (STANDBY1)	1.7uA
	Disable TIMG0, WFI (STANDBY1)	1.7uA

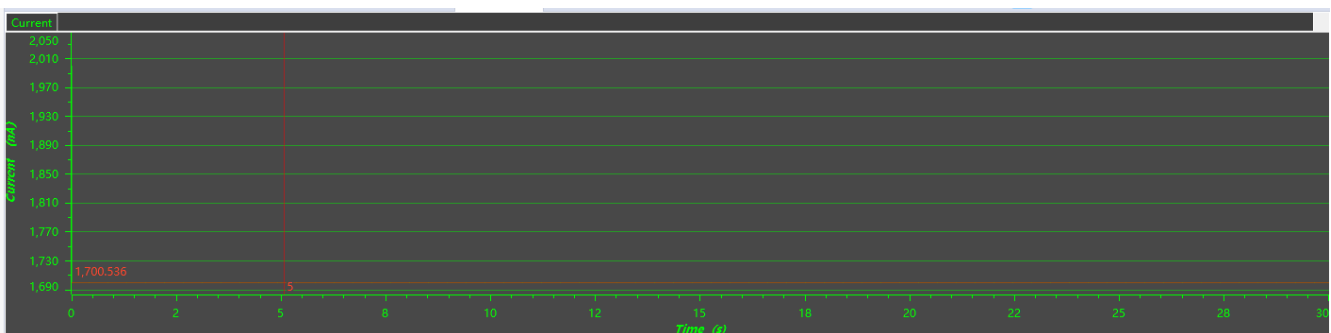


Figure 3-4. TIMG0 Power Consumption Test Under STANDBY1 Mode

3.4.2 TIMA0

The TIMA0 is enabled/disabled every 2s in TIMG0 interrupt, thus you can get the module-power consumption, and the work-power consumption current in RUN0 mode is about 30uA.

Table 3-6. TIMA0 Power Consumption Test Table

TIMA0	Configuration	Current
Software	<ul style="list-style-type: none"> TIMA0(clock from MCLK=32MHz) period is configured as 2s The test is operated in RUN0 mode 	
Test condition	Enable TIMA0	2.239mA
	Disable TIMA0	2.209mA

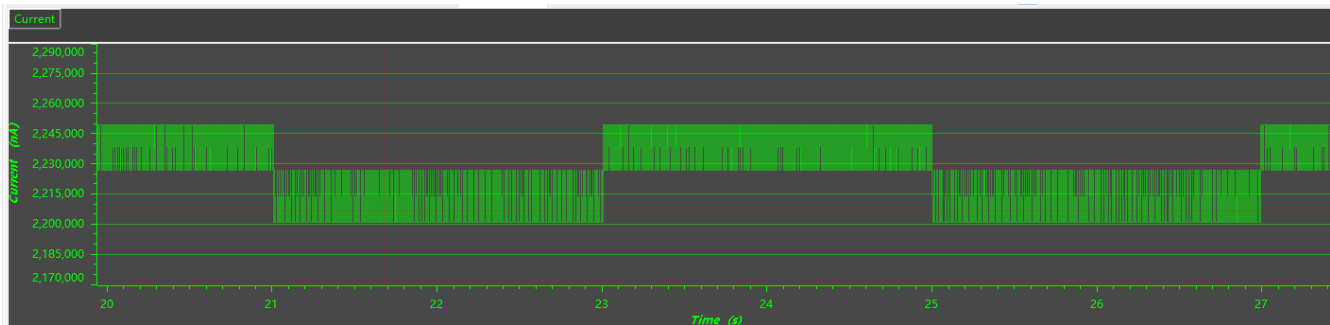


Figure 3-5. TIMA0 Power Consumption Test Under RUN0 Mode

TIMA0, as well as other timer on PD1, is forced to a disable state when system goes to STOP or STANDBY mode. To restore a timer from low-power mode, a re-configuration is necessary for timer to restart counting. Use the steps below to restart a timer:

1. Enable timer power by `DL_TimerG_enablePower();`
2. Re-configure timer x: `SYSCFG_DL_TIMER_x_init();`
3. Start the timer: `DL_TimerG_startCounter();`

3.5 UART

Table 3-7. UART0 Power Consumption Test Table

UART0	Configuration	Current
Hardware	<ul style="list-style-type: none"> Connect the UART0 pins to PC terminal by J101 RXD and TXD pins RX and TX pin internal pullup 	
Software	<ul style="list-style-type: none"> UART clock source from MCLK, and baud rate is 9600bps; 	
Test condition	Enable UART power and initial, WFI (STANDBY1)	3.5uA
	UART continuously transfer data (9600bps)	3.28mA
	RUN0 mode, while (1) {}	2.33mA
	Close UART power, while (1) {}	2.25mA
	UART disabled, WFI (STANDBY1)	1.9uA

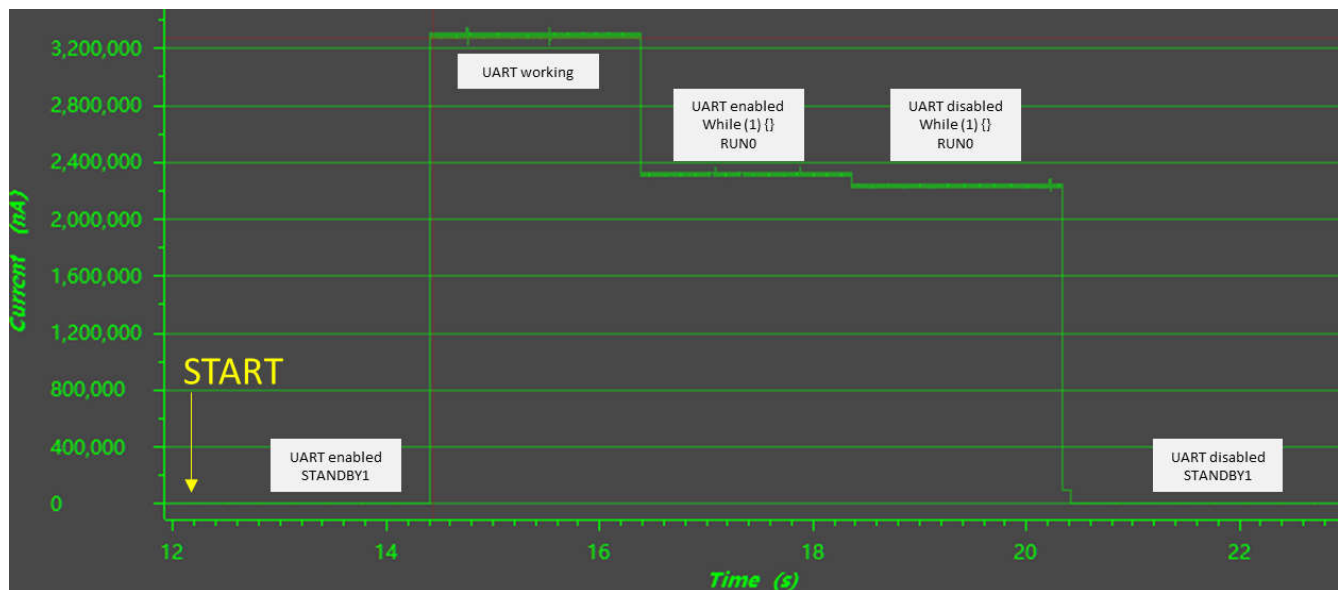


Figure 3-6. UART0 Power Consumption Test Under STANDBY1 Mode

There are several tips about UART0 in low-power mode:

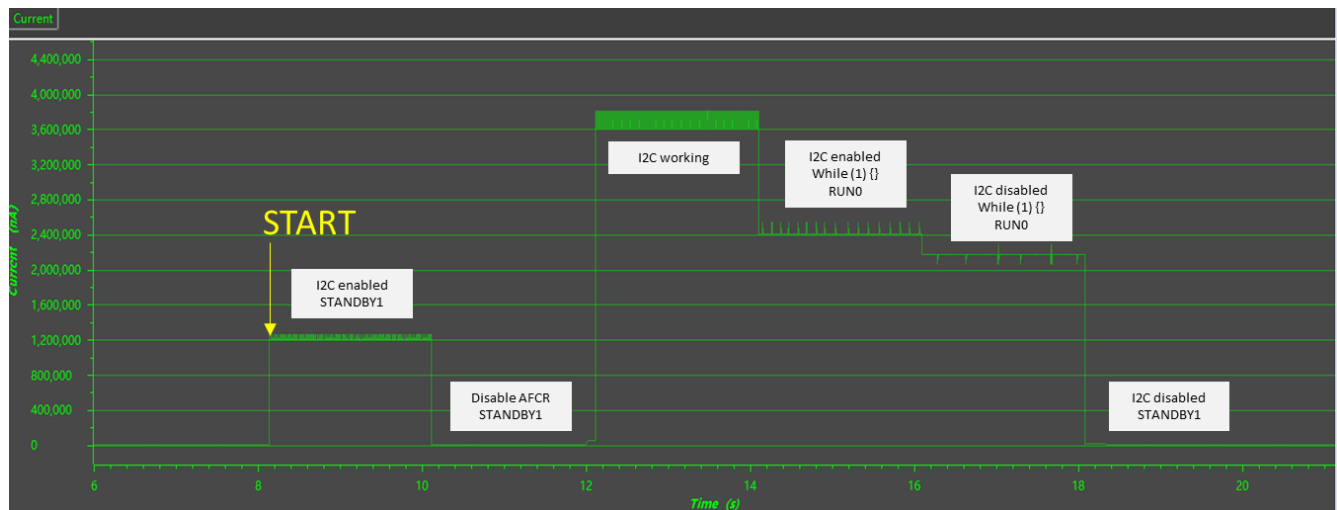
- UART module enable power in RUN0 mode is about 80uA according to the current change after close UART power
- In STANDBY mode, an enabled UART0 causes very little power consumption, less than 2uA.
- Make sure UART RX pin is pullup. Otherwise, there is unexpected current consumption after going into low-power mode cause by Asynchronous Fast Clock Request.
- Reset UART or disable power can both close UART power
- A floating or pulldown state of UART RX pin makes UART active, which causes an Asynchronous Fast Clock Request.

If UART0/1/2 are not reset before entering STOP or STANDBY mode, the configuration restores after exiting of low-power mode. And UART can work normally by `DL_UART_Main_enablePower()` after exit of low-power mode.

3.6 I2C

Table 3-8. I2C Power Consumption Test Table

I2C	Configuration	Current
Hardware	<ul style="list-style-type: none"> Not connect the target I2C device 2.2k pullup resistor in SCL and SDA pin 	
Software	<ul style="list-style-type: none"> I2C work as controller, I2C clock source from MCLK, and work frequency is 400kHz; 	
Test condition	Enable I2C power and initial, WFI (STANDBY1)	1.20mA
	Disable I2C Async Fast Clock Request, WFI (STANDBY1)	2.5uA
	I2C continuously transfer data(400kHz)	3.72mA
	RUN0 mode, while (1) {}	2.41mA
	Close I2C power, while (1) {}	2.18mA
	Enter low-power mode, WFI (STANDBY1)	2.0uA


Figure 3-7. I2C Power Consumption Test Under STANDBY1 Mode

According to the test result above, there are several conclusions and tips about I2C in low-power mode:

- I2C module enable power in this case at RUN0 mode is about 230uA
- Reset I2C or disable power can both close I2C power
- When I2C power is enabled and MCU goes into STOP or STANDBY mode, Asynchronous Fast Clock Request occurs, resulting in over 1mA current power consumption. A way to completely disable the Asynchronous Fast Clock Request function of I2C is by adding instruction `I2C_INST->GPRCM.CLKCFG = (1<<8) | (0xA9 << 24)`, where I2C_INST is I2C register base address.

If I2C is not reset before entering STOP or STANDBY mode, the configuration restores after exit of low-power mode. And I2C can work normally by enable power.

3.7 SPI

Table 3-9. SPI Power Consumption Test Table

SPI	Configuration	Current
Software	<ul style="list-style-type: none"> SPI works as controller and sourced from MCLK, and working frequency is 500kHz; 	
Test condition	Enable SPI power and initial, WFI (STANDBY1)	4.1uA
	SPI continuously transfer data(500kHz)	3.02mA
	RUN0 mode, while (1) {}	2.13mA
	Close SPI power and reset, while (1) {}	1.96mA
	Enter low-power mode, WFI (STANDBY1)	1.9uA

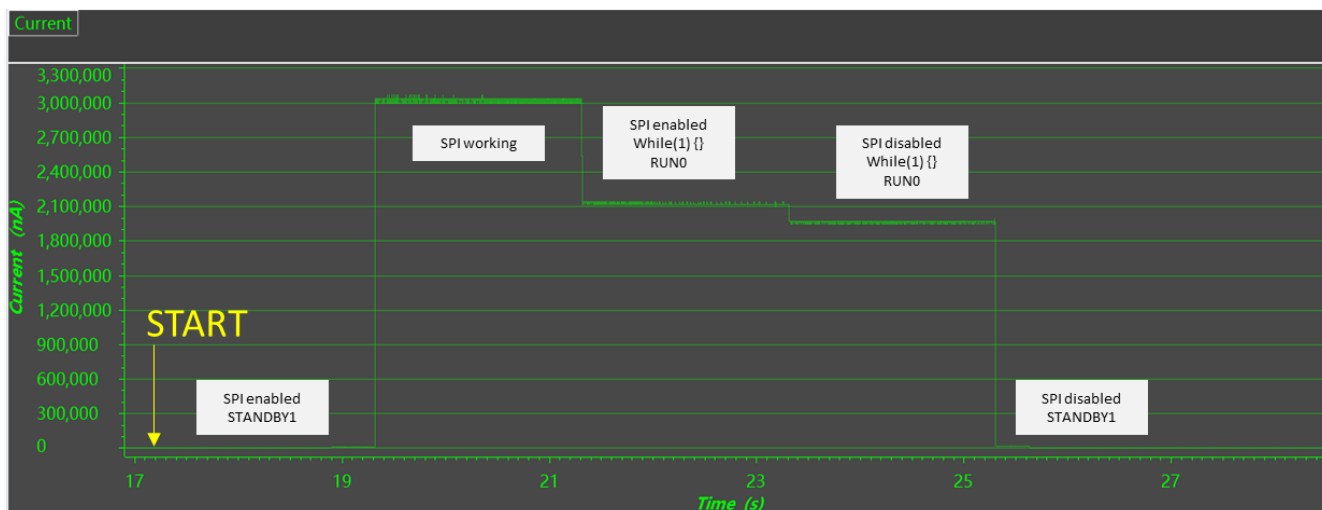


Figure 3-8. SPI Power Consumption Test Under STANDBY1 Mode

According to the test result above, there are several conclusions and tips about SPI in low-power mode:

- SPI module enable power in RUN0 mode is about 170uA
- As SPI module is located in PD1, an entry into STOP or STANDBY low-power mode forces the SPI module to be temporarily disabled. Reset SPI or disable power can also close SPI power.

If SPI are not reset before entering STOP or STANDBY mode, the configuration restores after exit of low-power mode. And SPI can work normally by enable power after exit of low-power mode.

3.8 MCAN

Table 3-10. MCAN Power Consumption Test Table

MCAN	Configuration	Current
Hardware	<ul style="list-style-type: none"> Connect another CAN device by TCAN1046 TCAN1046 is not power by the measured MCU launchpad 	
Software	<ul style="list-style-type: none"> MCAN clock from external 40MHz oscillator MCAN: 250kHz arbitration bit rate, 2MHz data bit rate 	
Test condition	Enable MCAN power, WFI (STANDBY1)	10.6uA
	Initial CAN, CAN continuously transfer data	3.38mA
	RUN0 mode, while (1) {}	3.28mA
	Close MCAN power, while (1) {}	2.80mA
	Enter low-power mode, WFI (STANDBY1)	10.0uA

According to the test result above, there are several conclusions and tips about MCAN in low-power mode:

- MCAN module enable power in RUN0 mode is about 480uA
- As MCAN module is located in PD1, an entry into STOP or STANDBY low-power mode forces the MCAN module to be temporarily disabled. Reset MCAN or disable power can also close MCAN power.
- As external oscillator is used in this case, the current in STANDBY1 is larger than description in data sheet.

For restoring MCAN configuration from low-power mode, a re-configuration is necessary for MCAN to work normally. Use the following steps below:

1. Enable timer power by `DL_MCAN_enablePower()`;
2. Re-configure MCAN x: `SYSCFG_DL_MCANx_init()`;

3.9 SYSPLL

Table 3-11. SYSPLL Power Consumption Test Table

SYSPLL	Configuration	Current
Software	• LFCLK = LFSOC, MCLK = SYSPLL2x (32MHz);	
Test condition	Enter low-power mode, WFI (STANDBY1)	38uA
	Disable PLL, WFI (STANDBY1)	1.9uA

In MSPM0, entering low-power mode is not automatically disable SYSPLL module, although MCLK and ULPCCLK are not sourced from SYSPLL anymore. As a result, SYSPLL remains enabled in low-power mode and brings unexpected power waste. The way to avoid this is manually disable SYSPLL to reduce power consumption by `DL_SYSCCTL_disableSYSPLL()` API function.

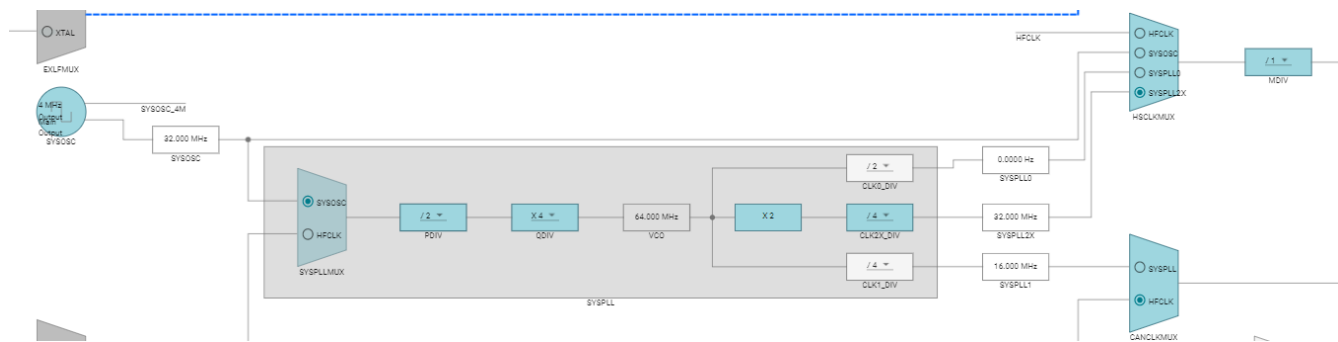


Figure 3-9. SYSPLL Configuration

3.10 VREF

Table 3-12. VREF Power Consumption Test Table

VREF	Configuration	Current
Software	<ul style="list-style-type: none"> Internal VRFE = 2.5V; 	
Test condition	Enable VREF power, enable VREF module, WFI (STANDBY1)	195uA
	RUN0 mode, while (1) {}	2.32mA
	Disable VREF module, RUN0 mode, while (1) {}	2.24mA
	Disable VREF power, RUN0 mode, while (1) {}	2.19mA
	Enter low-power mode, WFI (STANDBY1)	9.0uA

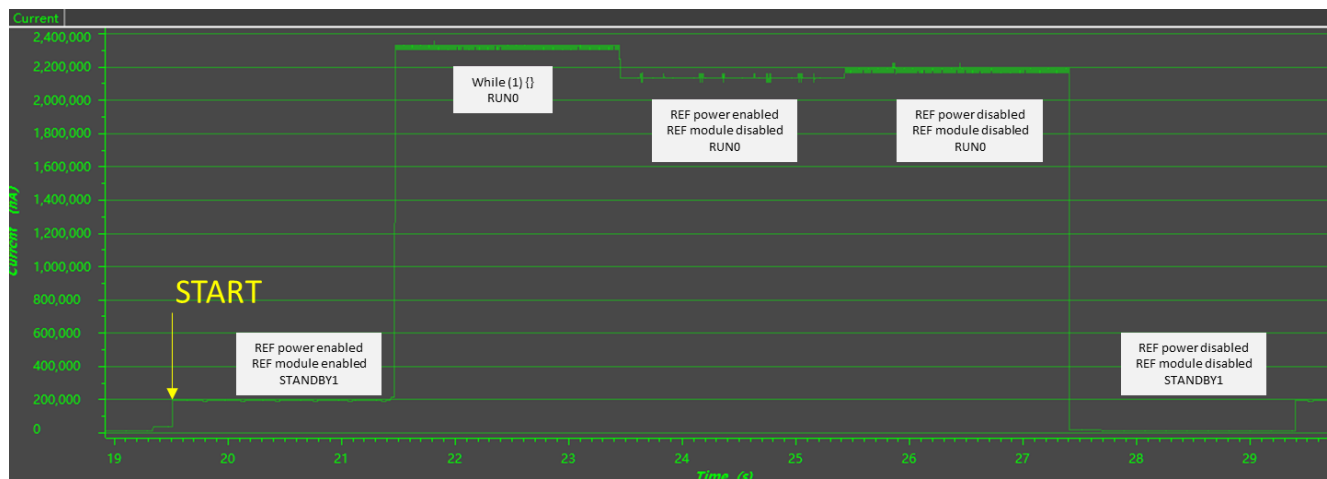


Figure 3-10. VREF Power Consumption Test under STANDBY1 Mode

According to the test result above, there are several conclusions and tips about VREF in low power mode:

- In this case, there is 9uA current left of VREF after entering STANDBY1 mode.
- The VREF power in RUN0 mode is about 80uA according to the current change after disable VREF module.

Pay attention that the power consumption of VREF can not be closed by disable VREF power only and it is necessary to disable whole VREF module. The key flow of completely close VREF power is:

1. Disable VREF module by `DL_VREF_disableInternalRef(VREF);`
2. Disable VREF power by `DL_VREF_disablePower(VREF);`
3. Enter low-power mode by `WFI/WFE` instruction.

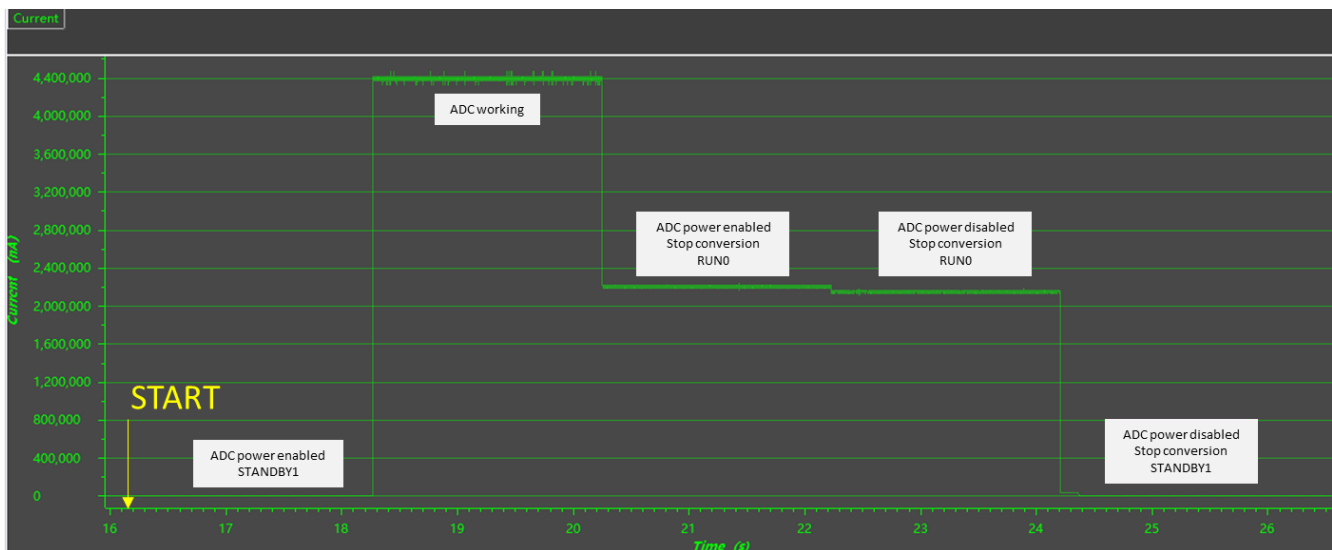
And the progress of enable VREF from low-power mode is:

1. Enable VREF power by `DL_VREF_enablePower(VREF);`
2. Enable VREF module by `DL_VREF_enableInternalRef(VREF);`

3.11 ADC

Table 3-13. ADC Power Consumption Test Table

ADC	Configuration	Current
Hardware	<ul style="list-style-type: none"> ADC input pin is PA27; 	
Software	<ul style="list-style-type: none"> ADC: <ul style="list-style-type: none"> Single repeat mode Sampling time: 1us Reference: VDDA Resolution: 12bit 	
Test condition	Enable ADC power, WFI (STANDBY1)	2.5uA
	Enable ADC conversion and start conversion, ADC continuously samples data	4.40mA
	Disable conversion, RUN0 mode, while (1) {}	2.21mA
	Close ADC power, while (1) {}	2.15mA
	Enter low-power mode, WFI (STANDBY1)	1.8uA


Figure 3-11. ADC Power Consumption Test Under STANDBY1 Mode

According to the test result above, there are several conclusions and tips about ADC in low-power mode:

- ADC module enable power in RUN0 mode is about 60uA according to the current change after close ADC power;
- ADC conversion consumes much power and it is necessary to disable conversion when ADC is not in use.

It is invalid to enter low-power mode just by disable power from a running ADC, and the a valid way to completely close ADC power is:

1. Disable REF power (If using internal reference).
2. Disable ADC conversion by `DL_ADC12_disableConversions()`.
3. Disable ADC power by `DL_ADC12_disablePower();`.
4. Enter low-power mode by `WFI/WFE` instruction.

And the steps to start an ADC is:

1. Enable REF power (If using internal reference).
2. Enable ADC power by `DL_ADC12_enablePower();`.
3. Enable ADC conversion by `DL_ADC12_enableConversions()`.
4. Start ADC conversion by `DL_ADC12_startConversion()` (if using software trigger).

4 Low-Power Mode Guidance

Table 4-1 gives the guidance of how to completely close/restore a peripheral power when entering/exiting low-power mode of MSPM0. The API function of enable or disable a peripheral can be found in the last chapter, and the final step for any peripherals to enter STANDBY or STOP mode is WFI or WFE instruction. Pay attention that the order of steps about entering and exiting low-power mode is also important.

Table 4-1. Peripheral Enter/Exit Low-Power Mode Guidance

Peripheral		Enter Low Power Mode (STANDBY)	Exit Low Power Mode (STANDBY)
PD0	UART	<ul style="list-style-type: none"> • Make sure RX pin is pullup; • Disable power. 	<ul style="list-style-type: none"> • Enable power.
	I2C	<ul style="list-style-type: none"> • Disable I2C Asynchronous Fast Clock Request; • Disable power. 	<ul style="list-style-type: none"> • Enable power.
PD1	TIMA	<ul style="list-style-type: none"> • Disable power (can be ignore for PD1 peripheral). 	<ul style="list-style-type: none"> • Enable power; • Re-configure TIMA (clock and register); • Start timer.
	SPI	<ul style="list-style-type: none"> • Disable power (can be ignore for PD1 peripheral). 	<ul style="list-style-type: none"> • Enable power.
	MCAN	<ul style="list-style-type: none"> • Disable power (can be ignore for PD1 peripheral). 	<ul style="list-style-type: none"> • Enable power; • Re-configure MCAN.
	GPIO	<ul style="list-style-type: none"> • Avoid configuring as floating input. 	<ul style="list-style-type: none"> • Configure as needed function pin.
Analog	SYSPLL	<ul style="list-style-type: none"> • Disable SYSPLL module before entering LPM. 	<ul style="list-style-type: none"> • Enable SYSPLL module if it is used as clock of any peripherals.
	VREF	<ul style="list-style-type: none"> • Disable VREF module; • Disable VREF power. 	<ul style="list-style-type: none"> • Enable VREF power; • Enable VREF module.
	ADC	<ul style="list-style-type: none"> • Disable VREF (if internal reference used); • Stop ADC conversion; • Disable ADC power. 	<ul style="list-style-type: none"> • Enable VREF (if internal reference used); • Enable ADC power; • Enable ADC conversion; • Start ADC conversion.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2024, Texas Instruments Incorporated