

Application note

How to build a LPBAM application on STM32U5 MCUs using STM32CubeMX

Introduction

This application note shows how to build an LPBAM (low-power background autonomous mode) application using the new STM32CubeMX LPBAM feature on STM32U5 MCUs. It details all the necessary steps and instructions to follow in STM32CubeMX, and the APIs that must be added to the STM32CubeMX (version 6.9.0) generated code.

This document does not provide details about LPBAM operating mode, nor the LPBAM utility, nor about the STM32CubeMX LPBAM feature. It includes only the steps necessary to build the application.





1 General information

LPBAM is an operating mode that allows peripherals to function autonomously, independently of power modes, and without running any software. A hardware subsystem embedded in the STM32 microcontroller implements LPBAM.

To create an LPBAM application using the STM32CubeMX tool, the user needs the STM32CubeMX standard view. This sets up the main application and code generation, and the LPBAM view to build the LPBAM applications.



Figure 1. SM32CubeMX tool scope

This document applies to all STM32U5 devices. All these products are Arm[®]-based microcontrollers.

Note: For more information on LPBAM, refer to the application note STM32U5 series power optimization using LPBAM (AN5645).

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

arm



2 LPBAM TempSense application description

2.1 Principle

The purpose of this application is to run periodic ADC conversions in STOP2 mode, with converted value threshold detection. Once threshold is reached, the conversion period is modified staying in STOP2 mode. The threshold value is also modified, and its interrupt is enabled to wake up from Stop mode when the ADC converted data reaches this second threshold value

After reset, the main application configures the system clock to the maximum device frequency. This increases the application performance. It then configures the system power supply on the SMPS regulator to reduce the device power consumption.

The main application then configures the resources required for the main application (ICACHE, GPIO, DMA, LEDs).

When all of the resources needed are configured and ready, the main application calls the LPBAM APIs to initialize, build, link, and start the scenario.

The application scenario description is as follows:

- The ADC peripheral is configured to convert the internal temperature sensor mapped to its temperature sensor channel. The conversion starts with a period of 100 ms. For this application, LPTIM PWM is the trigger signal. The transfer of converted data from ADC peripheral to SRAM peripheral is done by a DMA channel. The transferred data is stored in a buffer (Threshold1_Data_Buffer). The ADC analog watchdog monitors the internal temperature signal to be converted. At this point, the analog watchdog 1 threshold is configured to the threshold_1 value. The whole system then enters STOP2 mode.
- When the internal temperature reaches the analog watchdog 1 threshold_1 value, the ADC analog watchdog 1 generates its trigger event. It does not use an interrupt, in order to avoid the system wake-up. This signal propagates to another DMA channel that updates the ADC conversion period to 10 ms (LPTIM PWM signal period), changes the buffer conversion to a new buffer (Threshold2_Data_Buffer), and configures the analog watchdog 1 to the threshold_2 value.
- When the internal temperature reaches the analog watchdog 1 threshold_2 value, the ADC analog watchdog 1 generates an analog watchdog interrupt. This wakes up the whole system from STOP2 mode.
- After the wake-up from STOP2 mode, the main application calls the LPBAM APIs to stop, unlink, and deinitialize the scenario.

2.2 Implementation

The following resources are needed in order to implement the scenario:

- an ADC peripheral with analog watchdog threshold-detection capability
- an LPTIM peripheral with PWM generation capability. Linked internally to ADC peripheral trigger input.
- a first DMA channel, to ensure the storage of converted data
- a second DMA channel to ensure:
 - starting of the first DMA channel
 - reconfiguration of the ADC analog watchdog 1 threshold
 - update of LPTIM period, and its reconfiguration
 - restart of the first DMA channel after detection of threshold 1.

To implement this scenario, the user must:

- 1. Configure the ADC peripheral (using the HAL), to perform conversions on the internal temperature sensor channel.
- 2. Configure the ADC analog watchdog 1 (using the HAL), to detect the temperature sensor threshold 1 value in silent mode (with no interrupt, to avoid system wake-up).
- Configure the LPTIM peripheral (using the HAL) to generate a PWM signal with 100 ms period, and 50% duty cycle. This is used as a trigger source for the ADC.
- 4. Build an LPBAM queue 1, named Threshold1_Conversion_Q (with no DMA trigger condition). This allows:
 - ADC conversion start (with no configuration)
 - storage of converted data below threshold 1 in the SRAM Threshold1_Data_Buffer buffer.



- 5. Build an LPBAM queue 2 named Threshold2_Conversion_Q (with no DMA trigger condition). This allows:
 - ADC conversion start (with no configuration)
 - storage of converted data below threshold 2 in the SRAM Threshold2_Data_Buffer buffer.
- Build an LPBAM queue 3 named Thresholdx_Config_Q (with analog watchdog 1 DMA trigger condition). This allows:
 - queue 1 execution start, at the first step
 - ADC conversion stop
 - LPTIM PWM period update
 - reconfiguration of the analog watchdog threshold 2 (with interrupt generation capability)
 - queue 2 execution start.



Figure 2. Block diagram

2.3 STM32CubeMX LPBAM TempSense application building

Open STM32CubeMX, version 6.9.0 is used to develop this application, and choose your board in the board selector menu.

New Project from a Board MCU/MPU Selector Example Selector Cross Selector R C 3 * Features STM32U5 Series NUCLEO-U575ZI-Q Part Numbe ☆ NUCLEO Q + ACTIVE Product is in mas PRODUCT INFO Туре Supplier MCU / MPU Serie 5 Marketing Status Price MEMORY Ext. Flash = 0 (MBit) Boards List: 1 item

Figure 3. NUCLEO-U575ZI-Q board selection

DT73276V1



Then choose start project with no peripheral initialization pushing on "NO" button. (To avoid generation of useless project code, the required peripherals are initialized later.)

	Figure 4. Peripheral Initialization	
MX Board P	roject Options: NUCLEO-U575ZI-Q	×
? Init	ialize all peripherals with their defaul	t Mode ?
	Yes No	
To build an LPBAM application when T 1. Choose "without TrustZone activate 2. Click on OK	FrustZone [®] is deactivated: ed".	
	Figure 5. TrustZone [®] deactivated	
TrustZ	Figure 5. TrustZone [®] deactivated	×
TrustZ	Figure 5. TrustZone [®] deactivated cone feature available Do you want to create a new proje	× ect :
TrustZ	Figure 5. TrustZone [®] deactivated cone feature available Do you want to create a new proje without TrustZone activated ?	× ect :
TrustZ	Figure 5. TrustZone [®] deactivated Cone feature available Do you want to create a new proje without TrustZone activated ? with TrustZone activated ?	× ect :

The STM32CubeMX tool entry point is always the standard view. Opening the project manager and saving the main project is recommended.

For this application:

- 1. Click on project manager panel.
- 2. Name the project LPBAM ADC TempSense
- 3. For the "Project Location", save the project with the other examples in the firmware package.

Figure 6. Project settings

Home	STM32U575ZITXQ - NUC	LEO-U575ZI-Q	PBAM_ADC_TempSense.ioc - Project Manager > LPBAM Scenario & Co	infiguration-TempSens GENERATE CODE
	Pinout & Confi	guration	Clock Configuration	Project Manager
	Project	Project Settings Project Name	LPBAM_ADC_TempSense	
	20078/000	Project Location	2Cube_FW_U5_V1.0.2\STM32Cube_FW_U5_V1.0.0\ProjectsWUCLEC	-U575ZI-QIApplications/LPBAM Browse

As this application is under the STM32Cube firmware file tree, there is no need to generate LPBAM utility files. The user must then set up the code generator settings according to the targeted generated location:

- 1. Go to "Code Generator".
- 2. Choose "Add necessary library files as reference in the toolchain's project configuration file".

Figure 7. STM32Cube MCU packages

M)	STM32CubeN	MX Untitled*: STM32U575ZITxQ N	UCLEO-U575ZI-Q							
	CubeMX	F	ile	Window	Help					
	Home >	STM32U575ZITxQ - NUC	CLEO-U575ZI-Q	> Untitled	- Project Manager	LPBAM Scenario	& Configuration	\rightarrow	GENERATE CODE	
1		Pinout & Configura	tion		Clock Configu	ration		Projec	ct Manager	
		Project								
			STM32Cube MCU p Copy all used lib Copy only the ne Add necessary lip	ackages and en raries into the p ecessary library ibrary files as re	nbedded software pack roject folder files ference in the toolchain	s project configuration file				

At this point, the project is configured and the ./OC file is saved under the selected path.

The user then opens the "Pinout & Configuration" view.

From this view, the user can configure all the main application resources. For this project, only the system resources need to be configured.

To increase the system performance, enabling the ICACHE peripheral in one-way configuration is recommended. To do this, click on the "System Core" menu, then on the "ICACHE" peripheral, and change "Mode" to "1-way (direct mapped cache)".

Figure 8. ICACHE activation



To reach the highest performance, configure the system clock to the highest value.

For this project, the user chooses PLL 1 as the system clock source. The PLL source is the MSI oscillator. (MSI range four is recommended to provide a 4 MHz clock signal).

The PLL 1 configuration is as follows:

- PLLM = 1
- PLLN = 80
- PLLP = 2
- PLLQ = 2
- PLLR = 2

Figure 9. Clock configuration



For this project, the user chooses SysTick as the application timebase.

Figure 10. Changing the timebase source

Pinout & Configuration		Clock Configuration		Project Manager
		✓ Software Packs	✓ Pinout	
Q ✓ Ø Categories A->Z	_	SYS Mode and Configuration Mode		Pinou
System Core CORTEX_M33 DCACHE1 GPDMA1 GPIO ICACHE IWDG LPDMA1 NVIC RAMCFG A RCC SYS	Timebase Source SysT	ick ₽	~	

At this point, the main project system is configured. On the next step, the user needs to build the LPBAM TempSense application.

Click on the "LPBAM Scenario & Configuration" panel.

Figure 11. LPBAM scenario and configuration selection

KTM32Cube	IX LPBAM_ADC_TempSense.ioc*: STM32U575ZITxQ NUC	LEO-U575ZI-Q			
STM32 CubeMX	File	Window	Help		
Home >	STM32U575ZITxQ - NUCLEO-U575ZI-Q	> LPBAM_ADC	C_TempSense.ioc - Pinou	t & Configuration	 LPBAM Scenario & Configuration
	Pinout & Configuration	С	lock Configuration		Project Manager
		~	Software Packs	✓ Pinout	

To add an LPBAM application, click on the "+ Add Application" option. This is situated on the left, under the LPBAM manager.

When adding an LPBAM application, the STM32CubeMX tool shows the LBPAM view. As for the standard view, it contains "LPBAM Scenario & Configuration", "Pinout & Configuration" and "Clock Configuration" panels.

The naming chosen in the project should be reused in code-generated APIs and variables. This ensures consistency between STM32CubeMX LPBAM tool views and LPBAM generated application code. Carefully chosen user application naming is therefore recommended for clear and readable generated code. To do this:



- 1. Change the name of the application from LpbamAp1 to TempSens.
- 2. Change the name of the "Scenario" to "MultiThres".

Figure 12. Application and scenario naming

Home 🔰 STM32U575ZITxQ - NUCLEO-U575ZI-Q	LPBAM_	ADC_TempSense	.ioc - LPBAM Scenario & Configurati	on 🔰 LPBAM Scen
LPBAM Scenario & Configuratio	on		Pinout & Configura	tion
LPBAM Management ✓ ILPBAM Manager ✓ © TempSens ✓ © MultiThres I Queue1 I Add Queue I Add Application	~	Queue1		+ -

After renaming of the LPBAM application and scenario, the user needs to configure the system power. To reach the lowest power consumption regarding the hardware target, the user shuts off all unused resources through the PWR peripheral. To do this:

- Click on the "Pinout & Configuration", then on "Power and Thermal", then on "PWR".
- Under "Low Power", select "Power saving mode", then enable the power-down for all the SRAMs, except SRAM4 and ICACHE.



Figure 13. PWR configuration

DT73278V1

At this point, the system is optimized in terms of consumption. It is ready to host the LPBAM application operation. The user then needs to configure (using HAL drivers) the LPBAM scenario peripherals listed in Section 2.2: Implementation. To do this:

- 1. Click on "Analog", then on "ADC4".
- 2. An "ADC4 Mode and Configuration" window appears. Under "Mode", scroll down and select "Temperature Sensor Channel" (the only channel used in this application).

As any error occurring while in low-power mode stops the LPBAM mechanism running, it is recommended to enable any error interrupts.

Under the "Configuration" menu, click on "Advanced Settings for LPBAM" and enable "Overrun interrupt".

		LPBAM Scenario & Configuration	Pinout & Configuration
			✓ Pinout
Q	~ 《	ADC4 Mode and Configurat	lion
Categories A->	Z	Mode	
System Core	>	Z Temperature Sensor Channel	
Analog	~	Vrefint Channel	·
•		Configuration	
COMP1		Reset Configuration	
COMP2		Parameter Settings Advanced Settings for LPBAM User Con	istants 😔 NVIC Settings
DAC1		Configure the below parameters :	
OPAMP1 OPAMP2 VREFBUF		Q Search (Ctri+F) ③ ③	\$
		Overrun interrupt Enable	

Figure 14. "ADC4 Mode" and "Advanced Settings for LPBAM" configuration

The ADC configuration sequence for this scenario is as follows:

- 1. Click on "Parameter Settings" and change the "Sequencer" to "Sequencer set to not fully configurable". (Use only internal ADC channel.)
- 2. Change the "Scan Conversion Mode" to "Forward".
- 3. Enable the "DMA Continuous Requests" (ADC conversions are performed in an infinite loop via DMA mode).
- 4. Change the "Sampling Time Common 1" to 19.5 cycles.
- 5. Change the "Sampling Time Common 2" to 814.5 cycles.
- 6. Change "External Trigger Conversion Source" to "LPTIM 1 CH1 event".
- 7. Enable "Analog WatchDog1".
- 8. Configure the "High Threshold" to 1005, which is equivalent to 0.809 V (1005 / 4095) * 3.3 V).

Figure 15. ADC4 parameter settings



DT73279V1

The NVIC is used to detect ADC error interrupts, then wake the system from low-power mode. It is mandatory to enable this configuration via the NVIC settings.

Click on "NVIC Settings", and enable the "ADC4 global interrupt".

Figure 16. ADC4 NVIC settings

		LPBAM Scenario & Configuration			Р	inout & Configu	iration
					🗸 Pinou	ıt	
٩	~ 6	ADC4 Mode	and Configuration				
Categories A->	Z	Cor	nfiguration				
System Core	>	Reset Configuration					
Analog	~	Parameter Settings Settings for LPBAM	User Constants		C Settings		
		NVIC Interrupt Table	Enable	1	Preemption Priority	Sub Priority	
ADC4		ADC4 (12bits) global interrupt		0		0	

At this point, the ADC is configured and ready to start conversion, with analog watchdog monitoring enabled. The LPTIM peripheral must then be configured to generate a PWM signal. It is set to 100 ms period, and 50% duty cycle:

- 1. Click on "Timers" then on "LPTIM1". The "LPTIM1 Mode and Configuration" window appears.
- 2. In "Mode", change the mode from "disable" to "Count internal clock events". Then select "Channel_1_Active".
- 3. Under "Configuration", in the "Parameter Settings":
 - a. Click on 🤗 , and select no check. Then change the "Period" to "LPTIM1_PWM_PERIOD1".
 - b. Under "channel 1", change "capture-Compare Selection" from "Capture" to "Compare", and "pulse value" to "LPTIM1_PWM_PULSE1".

		LPBAM Scenario & Configuration		Pinout & Configuration
				✓ Pinout
	~ 0	U	PTIM1 Mode and Configuration	
ategories A-	>Z		Mode	
System Core	>	Mode Counts internal clock events		~
oystem oure		External Trigger		
Analog	>	Channel 1 Active		
		CH1 IO usage no IO used : only internal insut/oute	d connections	
Timers	~	CHTTO dsage no to dsed , only internal input/outp	di comecions	*
+			Configuration	
LPTIM1	-	Reset Configuration		
LPTIM4		Decemptor Cattings Advanced Settings for LD	RAM Dillor Constants DN/C Satt	
RTC		Advanced Settings for LPT	SAM Ser constants Stavic Sett	iigs
		Configure the below parameters :		
		Q Search (Otri+F) () ()		0
Connectivity	>	V Clock	Drawnin D. d	
,		Clock Prescaler	Prescaler Divi	
Power and The	r	Period	LPTIM1 PWM PERIOD1	6
		Repetition counter	0	~
		✓ Preload		
		Update Mode	Update Immediate	
		✓ Trigger		
		Trigger Source	Software Trigger	
		✓ channel 1		
		capture-Compare Selection	Compare	
		capture-Compare Selection pulse value	LPTIM1_PWM_PULSE1	

Figure 17. LPTIM1 activation and parameter settings

On application start up, the LPTIM period must be 100 ms and the LPTIM clock frequency is 32 kHz. The auto-reload and counter compare values are calculated as follows:

 $Auto_reload = (counter_clock_frequency \cdot period) - 1 = (32.10^3 \cdot 0.1) - 1$ (1) = 3199

$$Counter_compare = (counter_clock_frequency \cdot pulse) - 1 = (32.10^3 \cdot 0.05)$$
(2)
- 1 = 1599

Where frequency is measured in Hz, and period and pulse are in seconds. In the "User Constants" panel, click on "add "and create three "Constants":

- LPTIM1_PWM_PERIOD1 = 3199U
- LPTIM1_PWM_REPETITION = 0U



LPTIM1_PWM_PULSE1 = 1599U

Home 🔰 STI	M32U575	SZITXQ - NUCLEO-U575ZI-Q 🔰 LPBAM_ADC_Temp:	pSense ioc - Clock Configuration 🔰 LPBAM Scenario & Configuration-TempSens >
		LPBAM Scenario & Configuration	Pinout & Configuration
			✓ Pinout
2	~ 0	LPTIM1	1 Mode and Configuration
Categories A->	Z	E.	Configuration
System Core	>	Reset Configuration	
Analog	>	Parameter Settings Advanced Settings for LPBAM	User Constants O NVIC Settings
Timers	~	Search Constants	
		Search (Ctrl+F)	add remove:
C LPTIM1		Constant Name	Constant Value
LPTIM4		LPTIM1_PWM_PERIOD1	3199U
RTC		LPTIM1_PWM_REPETITION	UO
		LPTIM1_PWM_PULSE1	1599U

Figure 18. User constants adding

At this point, the ADC and LPTIM only need the LPTIM clock configuration before they can generate a PWM signal. The user must also configure the wake-up and peripheral kernel clocks. These allow the LPBAM application peripheral to operate in low-power mode.

For optimum power consumption, the wake-up clock frequency and kernel clocks sources should be chosen carefully.

When using the MSIK as kernel clock, use of the same scale (range 4 for this application) as an MSIK system clock is recommended:

- 1. Choose MSIK as ADC bus clock.
- 2. Choose LSI as LPTIM kernel clock .



Figure 19. LPTIM1 and ADC4 clock configuration

The next step is to configure the DMA channel that ensures the transfer of converted data from ADC to SRAM threshold buffers.

In LPBAM mode, the DMA channel must be configured in linked-list modes.



When DMA transfer is done in an infinite loop, the DMA channel execution mode should be configured in circular mode:

- 1. Left-click on "System core", then on "LPDMA1". Enable the "Linked-List Mode" for "CH0".
- 2. Click on "CH0". Then change the "Execution Mode" to "Circular" so that ADC conversion tasks are run in an infinite loop.

	l	_PBAM Scenario & Configuration		Pinout & Configuration
				✓ Pinout
	~ 🔕	LPDMA1 M	ode and Configuration	1
ategories A->	Z	с	onfiguration	
System Core	\sim	Reset Configuration		
÷		All Channels CH0 Advanced Settings for LPBAM	User Constants Since NVIC Set	ttings
GPIO		Configure the below parameters :		
NVIC	_	Q Search (Ctrl+F) ③ ④		0
RCC		Channel Configuration for Linked List		
		Priority	Low	
Analog	>	Execution Mode (circular/linear) of the Linked List	Circular	
Analog		Linked List Execution Mode	The List is fully executed	and database in the ball and
Timers	>	Transfer Event Generation	The TC (and the HT) event is gen	erated at the (respectively half) end
Connectivity	>			

Figure 20. "LPDMA1 Mode" and "CH0" configuration

Click on the "Advanced Settings for LPBAM". Then enable:

- "Data transfer Error"
- "Update Link Error"
- "User Setting Error"



lome > STM	32U575Z	ITxQ - NUCLEO-U575ZI-Q	LPBAM_ADC_TempSens	e.ioc - Pinout & Configurati	on 🔰 LPBAM S	cenario & Configuration-TempSer
		LPBAM Scenario & Con	figuration		Pinou	t & Configuration
					✓ Pinout	
۲	~ (0)		LPDMA1 M	lode and Configuration		
Categories A->Z	z		C	onfiguration		
System Core	~	Reset Configuration				
¢		All Channels OCHO	Advanced Settings for LPBAM	🥥 User Constants 🛛 📀 N	MC Settings	
C LPDMA1		Configure the below parameters				
NVIC		Q Search (Ctrl+F)	0			0
RCC		V IP Internal Interrupt for CH0				
		Data Transfer Error		Enable		
		Update Link Error		Enable		
Analog	>	User Setting Error		Enable		
Timers	>	Trigger Overrun Error		Disable		

Next, go to "NVIC Settings" and enable "LPDMA1 SmartRun Channel 0 global interrupt".

DT73281V1

Note:

Figure 22. LPDMA1 NVIC settings



DT73282V1

All LPBAM peripherals are now configured and ready for scenario queue building. It is mandatory to ensure that all used resources are configured before starting to build the scenarios.

2.4 Scenario queue building

As for the LPBAM application and scenario, the naming chosen for queues in the project is reused in codegenerated APIs and variables.

The chosen queue name is automatically suffixed with "_Q" in the generated code. Now, go to the "LPBAM Scenario & Configuration" tab. Change the name of "Queue1" to "Threshold1_Conversion".

Figure 23. Queue naming

Home 🔰 STM32U575ZITxQ - NUCLEO-U575ZI-Q	LPBAM_ADC_TempSense.ioc - LPBAM Scenario & Configuration			LPBAM Scenario & Configuration-Tem	pSens >
LPBAM Scenario & Conf	iguration	_		Pinout & Configuration	
LPBAM Management	~	Threshold1_Conversio	'n	· •	Queue Ci
✓					Circul

The user must then build the Threshold1_Conversion queue functionalities by using the LPBAM function toolbox library.

The function toolbox provides functionalities that perform diverse LPBAM scenarios for each supported peripheral.

As for LPBAM queues, each LPBAM peripheral function can be renamed. By default, the STM32CubeMX LPBAM tool provides a default name. The function name is used in code-generated variable names.

The Threshold1_Conversion queue permits the ADC peripheral conversion and analog watchdog 1 monitoring to start. The converted data is stored in the Threshold1_Data_Buffer buffer area in SRAM. (The user does not need to reconfigure the ADC peripheral, as it is previously configured using the HAL.)

The ADC configuration for this queue is as follows:

- 1. Go to the left, to the LPBAM function toolbox. Click on ADC4, then on the plus in front of the "Conversion data". An ADC4: "Conversion_data" box appears in the middle. Then go to the right, and change the function name to "ConvData_Threshold1".
- 2. Enable DMA continuous requests.
- 3. Change the data buffer name to "Threshold1_Data_Buffer".
- 4. Change the value of "Data Buffer Offset" to 0, and "Number of Data" to 512.



Figure 24. Queue1 building

Home STM32U575ZITxQ - NUCLEO-U575ZI-Q STM32U575ZITxQ - NUCLEO-U575ZI-Q	I_ADC_TempSense.ioc - Pinout & Configuration	LPBAM Scenario & Configuration-TempSense	GENERATE CODE	CHECK LPBAM DESIGN
LPBAM Scenario & Configuratio		Pinout & Configuration	Clo	ock Configuration
LPBAM Management >	Threshold1_Conversion	те	empSense/MultiThres/Threshold1_	Conversion/ADC4:ConvData_Threshold1
TempSense: Settings 🗸 🗸	ADC4:Co	mvData_Threshold1 ⁸⁸ Co	nter the Function Name Com/Data_Threshold1	~
LPBAM Scenario Settings LPBAM Scenario uses resources from Smart Run Domain LPRAM Scenario is hosted by LPDMA1		DN	MA Continuous Requests Enabled	~]
		Co	onversion Data - Data Configuration	~
LPBAM function Toolbox		Dat	ata Buffer Name Threshold1_Data_Buffer	1
ADC4 ~		Dat	ata Buffer Offset	1
Conversion Config		Nu	umber of Data 512	
Conversion data		Trig	igger Configuration	~

After adding all necessary functions to the active queue (respecting the order of function execution - first function, first execution), it is mandatory to configure the queue parameter.

For this queue, the ADC data conversion task must be done in an infinite loop. Also, the DMA channel that hosts this queue generates error interrupts to wake up the system from stop mode:

- 1. Go to the LPBAM manager and click on the "Threshold1_Conversion" queue.
- 2. Check "Circular Mode".
- 3. Enable: "Data Transfer Error Interrupt", "Update link Error interrupt", and "User Setting Error Interrupt".
- 4. Drag the arrow to face "Data" instead of "Conf", as only data conversion need to be repeated continuously.

Figure 25. Queue1 parameter configuration

DC_TempSense ioc - LPBAM Scenario & Configuration 💦 LPBAM Scenario & Configuration-1	GENERATE	CODE CHECK LPBAM DESIGN	
Pinout & Configuration		Clock Configuration	
Threshold1_Conversion +	TempSen	s/MultiThres/Threshold1_Conversion	
	Queue Circularity Configuration		Y
Conf Data ADC4: ComData_Threshold1	Circular Mode		
	DMA Channel Configuration		~
	Priority	Low	~
	DMA Channel Interrupt Configuration		~
	Data Transfer Error Interrupt	Enable	~
	Update Link Error Interrupt	Enable	\sim
	User Setting Error Interrupt	Enable	~
	Transfer Complete Interrupt	Disable	~
	Trigger Overrun Interrupt	Disable	~

As for the Threshold1_Conversion queue, the Threshold2_Conversion queue allows:

- ADC peripheral conversion start
- analog watchdog 1 monitoring start
- data storage in the Threshold2_Data_Buffer buffer, and writing the data to SRAM

The user does not need to reconfigure the ADC peripheral, as it is reconfigured with the Thresholdx_Config queue.

Click on "+Add Queue" to create a second queue, and name it "Threshold2_Conversion".

Figure 26. Queue2 naming

Home STM32U575ZITxQ - NUCLEO-U575ZI-Q	> LPBAM_4	DC_TempSense.ioc - LPB/	AM Scenario & Configuration	> LPBAM Scenario & Configura	ation-Temp
LPBAM Scenario & Co	onfiguration		P	inout & Configuration	
LPBAM Management	~	Threshold1_Conversion	Threshold2_Conversion		1 (t) (t)

Note:



The ADC configuration for this queue is as follows:

- 1. Go to the left, to the LPBAM function toolbox. Click on ADC4, then on the plus in front of "Conversion data".
- 2. Go to the right, and change the function name to ConvData_Threshold2.
- 3. Enable "DMA Continuous Requests".
- 4. Change "Data Buffer Name" to Threshold2_Data_Buffer.
- 5. Change the value of "Data Buffer Offset" to '0', and "Number of Data" to '512'.

Figure 27. Queue2 building

Home 🔪 STM32U575ZITxQ - NUCLEO-U575ZI-Q 🔪 LPBAM_	ADC_TempSense ioc - Pino	out & Configuration 🔰 LPBAM Scenario & Configuration-TempSense 🔰	GENERATE CODE CHECK LPBAM DESIGN
LPBAM Scenario & Configuration		Pinout & Configuration	Clock Configuration
LPBAM Management >	Threshold1_Conversion	Threshold2_Conversion	TempSense/MultiThres/Threshold2_Conversion/ADC4:ConvData_Threshold2 Enter the Function Name ComOata_Threshold2
TempSense: Settings LPBAM Scenario Settings LPBAM Scenario uses resources from Smart Run Domain DPBAM Scenario is hosted by LPDMA1		ADC4 CemData_Threshold2 ¹⁰	Conversion Data
LPBAM function Toolbox			Conversion Data - Data Configuration v Data Buffer Name Threshold2, Data_Buffer

For this queue, the ADC data conversion task must be done in an infinite loop. Also, the DMA channel that hosts this queue generates error interrupts to wake up the system from stop mode in case an error has occurred.

- Go to the LPBAM manager and click on the "Threshold2_Conversion" queue.
- Check "Circular Mode".
- Enable "Data Transfer Error Interrupt", "Update Link Error Interrupt", and "User Setting Error Interrupt".
- Drag the arrow to face "Data" instead of "Conf", as only data conversion needs to be repeated continuously.

Figure 28. Queue2 parameter configuration

DC_TempSense.ioc - LPBA	M Scenario & Configuration	on-TempSens GENE	ERATE CODE	CHECK LPBAM DESIGN			
N	Pinout & Configuration		C	ock Configuration			
Threshold1 Conversion	Threshold2 Conversion	Ten	npSens/MultiThre	es/Threshold2_Conversion			
		Queue Circularity Configurat	tion		~		
	Conf Data ADC4:Com/Data_Threshold2	🖾 Circular Mode					
		DMA Channel Configuration	DMA Channel Configuration				
		Priority		Low	~		
		DMA Channel Interrupt Cont	figuration		~		
		Data Transfer Error Interrupt		Enable	~		
		Update Link Error Interrupt		Enable	~		
		User Setting Error Interrupt		Enable	~		
		Transfer Complete Interrupt		Disable	~		
		Trigger Overrun Interrupt		Disable	~		

As for the Threshold1_Conversion and Threshold2_Conversion queues, the user must add, then rename the next queue to Thresholdx_Config.



Add a third queue, and name it "Thresholdx_Config".

Home 🔪 STM32U575ZITxQ - NUCLEO-U575ZI-Q	> LPBAM_A	DC_TempSense.ioc - LPBA	M Scenario & Configuration	🔪 LPBAM Scenario & C	onfiguration-Temp
LPBAM Scenario & Conf	figuration			Pinout & Configuration	
LPBAM Management	~	Threshold1_Conversion	Threshold2_Conversion	Thresholdx_Config	- +
✓ IPBAM Manager ✓ ③ TempSens ✓ ④ MultiThres — Threshold1_Conversion — Threshold2_Conversion — Threshold2_Conversion — Had Queue → Add Application					

Figure 29. Queue3 naming

The purpose of the Thresholdx_Config queue is to start the DMA channel execution of the Threshold1_Conversion queue when starting the LPBAM application execution.

Detection of the analog watchdog 1 threshold 1 signal conditions the execution of the subsequent functionalities. When detected, the Thresholdx_Config queue should update the LPTIM PWM period value to 10 ms to increase ADC conversion cadence. Then reconfigure the analog watchdog threshold 2 value, and start the execution of the DMA channel that hosts the Threshold2_Conversion queue.

Unlike the other queues, the third queue has four nodes.

The LPDMA1 configuration sequence for the first node is as follows:

- 1. Go to the LPDMA1 and click on the "+" in front of start, then change the function name to "Threshold1_Q_Start".
- 2. Select the available DMA channel to execute the "Threshold1_Conversion" queue.
- 3. Change "Queue Name" from "No Selection" to "Threshold1_Conversion".

Home > STM32U	175ZITXQ - NUCLEO-U575ZI-Q 💙 LPBAN	_ADC_TempSense ioc - Pinout & Configu	ration	LPBAM Scenario & Configuration-TempSense	GENERATE	E CODE CHECK I	PBAM DESIGN
	LPBAM Scenario & Configuration			Pinout & Configuration		Clock Configura	ation
LPBAM Management	>	Threshold1_Conversion Thresh	old2_Conversio	n Thresholdx_Config + -	TempSense/Multi	Thres/Thresholdx_Config/LP	DMA1:Threshold1_Q_Start
TempSense: Settings	>				Enter the Function Name	hreshold1_Q_Start	
			LPDMA1:Th	reshold1_Q_Start®	Start Parameters - Queue Co	infiguration	~
LPBAM function Tools	ox Ý				DMA channel to be started	LPDMA1 Channel0	~
ADC4	v				Queue to start	Designed inside LPBAM tool	Ŷ
Conversion Config	1 A.				Queue Name	Threshold1_Conversion	~
Conversion data							
Conversion	1. 4 . 1				Start Parameters - WakeUp I	interrupts Configuration	~
Analog Watchdog					Transfer Complete	Disable	×
001101					Half Transfer Complete	Disable	v .
COMPT							
COMP2	>				Trigger Configuration		× .
DAC1	>				The Function execution is	not conditionned by a Trigger	v
12C3	>						
LPDMA1	>						

Figure 30. First node configuration

The LPTIM1 configuration sequence for the second node is as follows:

- 1. Go to LPTIM1 and click on the "+" in front of PWM.
- 2. Change the function name to "Period_10ms".
- 3. Enable the period update state, and insert 319 as the period value.
- 4. Enable the pulse update state, and insert 159 as the pulse value.
- Under "Trigger Configuration", change "The Function execution is" from "Not conditioned by a trigger" to "Triggered on the Rising edge of the hardware Signal". This conditions the LPTIM update execution by analog watchdog 1 threshold 1 detection.
- 6. Change "Trigger hardware Signal is" from "EXTI line 0" to "ADC4 AWD1".



Figure 31. Second node configuration

Home STM32U575ZITxQ - NUCL	.EO-U575ZI-Q 💙 LPBAM_A	DC_TempSense.ioc - Pinou	t & Configuration	LPBAM Scenario & Configuration-	TempSense >	GENERAT	ECODE	CHECK LPBAM DESIGN	
LPBAM So	enario & Configuration			Pinout & Configuration			C	Clock Configuration	
LPBAM Management	>	Threshold1_Conversion	Threshold2_Conversion	Thresholdx_Config	14 N.	TempSense/	MultiThres/Thre	sholdx_Config/LPTIM1:Peri	od_10ms
TempSense: Settings	5				E	Enter the Function Name	Period_10ms		
LPBAM function Toolbox	~		LPDMA1:Thre	shold1_Q_Start	P	NWM			~
Sector (C	hannel	Channel 1		~
ADC4	~		LPTIM1:P	'eriod_10ms ⁸⁰	P	eriod update state	Enabled		~
Conversion Config					P	eriod value	319		
Conversion data	· ••				P	ulse update state	Enabled		~
Conversion	1. A. C.				: P	ulse value	159]
Analog Watchdog	· + · ·				re	epetition update state	Disabled		~
COMP1	\$				т	rigger Configuration			ů.
COMP2	>				т	he Function execution is	trigged on the Rising	Edge of the Hardware Signal	~
DAC1	>				т	rigger Hardware Signal is	ADC4 AWD1		~
12C3	5								
LPDMA1	5								
LPGPI01	>								
					1				

The ADC4 configuration for the third node is as follows:

- Click on ADC4, then on the "+" in front of the conversion config
- Change the name of the function to "Conversion_Config"
- Change "Sequencer" to "Sequencer set to not fully configurable"
- On the right, under the "Conversion_Config" disable the "Discontinuous Conversion Mode"
- Change "External Trigger Conversion edge" from "none trigger detection" to "Trigger detection on the rising edge"
- Enable "DMA Continuous Requests"
- Change "1st Channel" to "Temperature Sensor"

Figure 32. Third node conversion configuration

DC_TempSense.ioc - Pinout	t & Configuration 🔰 LPBAM Scenario & Configuration-TempSense 🔪	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration		Clock Configuration
Threshold1_Conversion	Threshold2_Conversion + LPDMA1:Threshold1_O_Stan [®]	TempSense/MultiThre Enter the Function Name Conversion Conversion Config	s/Thresholdx_Config/ADC4:Conversion_Config _Config
		Sequencer	Sequencer set to not fully configurable
	LPTIM1:Period_10ms	Scan Conversion Mode	Forward
		Continuous Conversion Mode	Disabled ~
	ADDA Comming Conf. P	Discontinuous Conversion Mode	Disabled ~
	ADC4.Conversion_Coning	External Trigger Conversion Edge	Trigger detection on the rising edge
		External Trigger Conversion Source	LPTIM 1 CH1 event
		Trigger Frequency	Low frequency
		Low Power Auto Wait	Enabled ~
		DMA Continuous Requests	Enabled V
		Number Of Channel can be converted	[1 v]
		1st Channel	Channel Temperature Sensor v

The analog watchdog configuration is as follows:

- 1. Enable "Analog WatchDog1"
- 2. Change "Analog WatchDog Channel" to "Channel Temperature Sensor"
- 3. Configure "High Threshold" to 1100
- 4. Enable "Interrupt Mode" to wake up the system after the detection of the analog watchdog 1 threshold 2 value.



Figure 33. Third node "Analog Watchdog1" configuration

Analog WatchDog 1		~
Enable Analog WatchDog1 Mode	Enabled	~
Watchdog Mode	Single regular channel	~
Analog WatchDog Channel	Channel Temperature Sensor	~
High Threshold	1100	
Low Threshold	0	
Interrupt Mode	Enabled	~

The LPDMA1 configuration for the fourth node is as follows:

- 1. Click on LPDMA1, then on the plus in front of start and change the function name to "Threshold2_Q_Start".
- 2. Select the available DMA channel to execute the Threshold2_Conversion queue.
- Use of the same DMA channel is recommended in this case. This reduces power consumption, as the Threshold1_Conversion and Threshold2_Conversion queues are never executed simultaneously.
 - 3. Change the queue name to "Threshold2_Conversion".

Figure 34. Fourth node configuration

	Pinout & Configuration		Clock Configuration			
Threshold1 Conversion	Threshold2 Conversion Thresholdx Config +	TempSens/MultiThres/Thresholdx_Config/LPDMA1:Threshold2_Q_Start				
		Enter the Function Name	Threshold2_Q_Start			
	LPDMA1:Threshold1_Q_Start	Start Parameters - Queue C	onfiguration ~			
		DMA channel to be started	DPDMA1 Channel0 ~			
	LPTIM1:Period_10ms	Queue to start	Designed inside LPBAM tool			
		Queue Name	Threshold2_Conversion ~			
	ADC4: Conversion_Config ¹⁰	Start Parameters - WakeUp	Interrupts Configuration			
	· · · · · · · · · · · · · · · · · · ·	Transfer Complete	Disable ~			
	LPDMA1:Threshold2_Q_Start [®]	Half Transfer Complete	Disable			

In this queue also, the user needs the DMA channel to generate error interrupts. These wake the system up from stop mode in case an error has occurred.

The third queue parameter configuration is as follows:

- Go to the left, and click on "Thresholdx_Config" under "Threshold2_Conversion"
- Enable "Data Transfer Error Interrupt", "Update Link Error Interrupt", and "User Setting Error Interrupt"

Note:



	ie.	Pinout & Configuration		Clock Configuration		
Threshold1 Conversion	Threshold? Conversion	Thresholdy Config	TempSens/MultiThres/Thresholdx_Config			
The should be sh	The should _ conversion	Threatenax coming	Queue Circularity Configuration		×	
	LPDMA1:Thresho	ld1_Q_Start ⁸⁸	Circular Mode			
	L DTIMI Durin	4.10	DMA Channel Configuration		Ŷ	
	LP HWI Pelo	u_toms	Priority	Low	~	
	ADC4:Conversion	on_Config [®]	DMA Channel Interrupt Configuration		×	
			Data Transfer Error Interrupt	Enable	~	
	LPDMA1:Thresho	ld2_Q_Start	Update Link Error Interrupt	Enable	~	
			User Setting Error Interrupt	Enable	~	
			Transfer Complete Interrupt	Disable	~	
			Trigger Overrun Interrupt	Disable	~	
			DMA Channel NVIC Configuration		Ŷ	
			Preemption Priority	0	~	
			Sub Priority	0	~	

Figure 35. Queue3 parameter configuration

DT73283V1

At this point, the "LPBAM TempSense" application is built in the STM32CubeMX LPBAM tool. It is then recommended to check the LPBAM design via the "Check LPBAM design" button. The STM32CubeMX tool checks, in the background, the consistency of the built application. It returns detected issues in the LPBAM log.

Figure 36. LPBAM design check

STM32Cub	eMX LPBAM_ADC_TempSense.ioc*: STM32U575ZITxQ NUCL	EO+U575ZI+Q							– ø ×
STM32 CubeMX	File	Window	Help					😥 🖡 P 💟	* 57
Home >	STM32U575ZIT×Q - NUCLEO-U575ZI-Q	> LPBAM_A	DC_TempSense.ioc - Cl	ock Configuration 🛛 🔪 LF	PBAM Scenario & Configuration-TempSens	\rangle	GENERATE CODE	CHECK LPBAM DESIGN	
LPBAM Scenario & Configuration				Pinout & Configuration		Cli	ock Configuration		
LPBAM	Management	~	Threshold1_Conversion	Threshold2_Conversion	Thresholdx_Config		TempSens/MultiTh	res/Thresholdx_Config	

The LPBAM log returns general information to help in LPBAM scenario customization. This information is in accordance with the scenario-build information.

The check LPBAM design step is optional, but recommended.

If the LPBAM log is clean (no warning "Chestnut messages" or error "Pink messages"), the user can return to the STM32CubeMX standard view to generate code.



Home > STM32U575ZTxQ - NUCLEO-U575Z-Q > LPBAM_A	DC_TempSens.ioc - Pinout & Configuration	LPBAM Scenario & Configuration-TempSens	GENERATE CODE	CHECK LPBAM DESIGN	
LPBAM Scenario & Configuration		Pinout & Configuration		Clock Configuration	
LPBAM Management	Threshold1_Conversion Threshold2_Convers	sion Thresholdx_Config < > +	TempSens/M	lultiThres/Thresholdx_Config	
🗢 🚾 LPBAM Manager			Queue Circularity Configuration	×	
✓ ⊘ TempSens ✓ ⊗ MultiThres ✓ Threshold1_Conversion	LPDMA1:T	Threshold1_Q_Start	Circular Mode		
MCUs Selection Output LPBAM Output Log					
		Show Attribute Warning Messages			
Tespes explicit. MultiPles Security 0 Peripheral Configuration (In Trapper Revealers Signals used in this scenario should be configured in this separation 'Jacon's Configuration' panel 0 Peripheral Configuration (In Trapper Revealers Signals used in this scenario configured in this separation 'Jacon's Configuration' panel 0 Peripheral Configuration (In Trapper Revealers Signals used in this scenario configuration in processing Configuration in the scenario configuration in the scenario configuration' panel 0 Concern Configuration (In Trapper Revealers Signals Used in this scenario (In Trapper Signals Configuration) in the scenario configuration in the scenario configuration in the scenario configuration (In Signals Configuration) in the scenario configuration in the scena					

DT73284V1





Note:

Note: This is not the only way to make an LPBAM application. Using other steps can also work, but this is not recommended.

Click on "pinout" and "configuration" to return to the STM32CubeMX standard view.

Any LPBAM application made by the STM32CubeMX LPBAM tool is DMA channel agnostic. Therefore, the LPBAM application configures the DMA channel chosen to host the queue. It is up to the main application to allow the necessary DMA channels to host the LPBAM application.

For this application, a DMA channel hosts the LPBAM queue.

As DMA channel 0 is used to host the Threshold1_Conversion and Threshold2_Conversion queues, the user must choose any other DMA channel.

The LPBAM application configures the DMA channel chosen to host the queue. Hence, through the STM32CubeMX standard the user only needs to enable the DMA channel and its interrupt line, as follows:

- 1. Click on LPDMA1, and choose any mode for channel 1. (The LPBAM application configures the mode and parameters.)
- 2. Go to "NVIC Settings" and enable the interrupt line.

The LPDMA is chosen because it is functional down to Stop 2 mode.

Figure 38. "LPDMA1 Mode" and "NVIC Settings" configuration

Pino	out & Con	figuration	Clock	Configuration		
				 Software Pac 		🗸 Pinou
Q ~	0		LPDMA1 Mode and Config	guration		
Categories A->Z			Mode			
System Core	~	Channel 0 - No Internal FIFO Dis	able			~
		Channel 1 - No Internal FIFO Link	ed-List Mode			~
CORTEX_M33			Configuration			
GPDMA1		Reset Configuration				
GPIO VICACHE		All Channels SECURITY	🔮 CH1 😔 User Constants	Solution Setting Solution Setting Solution Setting		
IWDG		NVIC Interrup	ot Table	Enabled	Preemption Priority	/ Sub Prio
V LPDMA1		LPDMA1 SmartRun Channel 1 global	interrupt	V	0	0

The user must configure the system power supply to achieve the lowest power consumption for the hardware target. The SMPS must be enabled through the PWR peripheral. To do this:

- Click on the "Pinout & Configuration", then on "Power and Thermal", then on "PWR".
- Under System power supply , select "SMPS" as power regulator.

Figure 39. Enable SMPS in main application

		Pinout & Configuration	Clock Configuration	
		and a second	✓ Software Packs	
1	~	PWR Mode	and Configuration	8
ateornes .	∿>Z		Aode	
		Dead Datters Sizeals deabled		
System Core	2	Crac Calley Signer County		
	10	Power saving mode		
Analog		Security/Privilege attributes		_
Timera		> isitationWilda		
		Cont	iguration	
Connectivity	>	The set of		
		And the second second		
Multimedia	Σ_{-}	Power Saving O PWR Privilege O User Constants		
		Configure the below parameters :		
Security	2	G Search (Ctri+F) 0 0		0
Complete		 System power supply 		
company		Power Regulator	SMPS	
Mddeware	3	 SRAM power down in Run mode 		
		SRAM1 power down in Run mode	Disable	
Trace and D	. >	SRAM2 power down in Run mode	Disable	
		SRAM3 power down in Run mode	Disable	
Power and T	¥.	SRAM4 power down in Run mode	Disable	
		SRAM power down in Stop mode	1000000	
S PWR		SRAM1 Page1 power down in Stop (0, 1, 2, 3) mode	Disable	
0		SKAW1 Pagez power down in Stop (0, 1, 2, 3) mode	Disable	
		SPAM9 Pages power down in Stop (0, 1, 2, 3) mode SPAM9 Pages power down in Stop (0, 1, 2) mode	Deable	
Ublibes	2	SRAM2 Pana2 newsr down in Stop (0, 1, 2) mode	Disable	
		SRAM3 Page1 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page2 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page3 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page4 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page5 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page6 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page7 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM3 Page8 power down in Stop (0, 1, 2, 3) mode	Disable	
		SRAM4 power down in Stop (0, 1, 2, 3) mode	Disable	
		RUALINE power down in Stop (0, 1, 2, 3) mode	Usable	
		DCACHE1 power down in Stop (0, 1, 2, 3) mode	Chisable	
		DKA22 PIAM power down in bcop (0, 1, 2, 3) mode	Disable	
		PFRIPH RAM power down in Stop (0, 1, 2, 3) mode	Disable	
		· · · · · · · · · · · · · · · · · · ·	Enditro vel	

DT73285V1



At this point, all the LPBAM application needs are configured.

As the LPBAM targets the lowest possible power consumption, it is recommended that all unused pins are set to analog mode:

- 1. Click on "Pinout" and choose "Set unused GPIOs".
- 2. Set the highest number of GPIOs available (111 in this application).
- 3. Change "GPIO Type" from "Input" to "Analog".

Figure 40. Pinout configuration

Pin	out & Configuration	t & Configuration Clock Configuration			Project Manager		
		~	Software Packs	▲ Pinout			
Q ~	0	Ν		Undo Mode and pinout	Ctrl-Z		
Categories A->Z		43		Redo Mode and pinout	Ctrl-Y		
				□ Keep Current Signals Placement	Ctrl-K		
System Core	>			Show User Label			
				Disable All Modes	Ctrl-D		
Analog	>			Clear Pinouts	Ctrl-P		
Timers	>		8 8 8 8 8 8 8	Clear Single Mapped Signals	Ctrl-M		
THICIS				Pins/Signals Options	Ctrl-O		
Connectivity	>		PES	List Pinout Compatible MCUs	Alt-L		
			P ES	Export pinout with Alt. Functions			
Multimedia	>		P 66 VBAT	Export pinout without Alt. Functions	Ctrl-U		
Security			UDER_BUTTON POID	Reset used GPIOs	Alt-G		
occurry			RCC_00032_00T PC15	Set unused GPIOs	Ctrl-G		
Computing	>		PFD	Pinout View Colors			
			P12	Layout reset			

Figure 41. GPIO configuration

MX Set unused GPIOs			×
Number of GPIOs	0	111	
GPIO Type	Analog 🗸	Ok	Cancel

At this point, the main and the LPBAM applications are ready to be generated. Click on "GENERATE CODE".

Figure 42. Code generation

STM32U575ZITXQ - NUCLEO-U575ZI-Q	LPBAM_A	ADC_TempSense.ioc - Pinout & Configuration $>$ LPBAM Scen	ario & Configuration	\rangle	GENERATE CODE
Pinout & Configuration		Clock Configuration		Project Manag	ger
		✓ Software Packs	✓ Pinout		

For the LPBAM application, the generated files are:

- Ipbam_tempsens.h
- Ipbam_tempsens_config.c
- Ipbam_tempsens_multithres_build.c
- Ipbam_tempsens_multithres_config.c

They contain all the APIs needed by the main application to manage any LPBAM application correctly. After successful generation of the project, click on "Open Project".

Figure 43. Code successfully generated



Note: To use the BSP drivers, the user needs to add the necessary files, which are taken into account during the compilation.

2.5 How to call APIs

According to the project needs, the user has to call the necessary generated APIs.

This application uses the BSP library, so the user must include board resources in the main application. Go to the main.h and add *«#include "stm32u5xx_nucleo.h"»* in the user section named "USER CODE BEGIN Includes".

/* Define to prevent recursive inclusion*/
#ifndef MAIN H
#define MAIN_H
#ifdef cplusplus
extern "C" {
#endif
/* Includes*/
#include "stm32u5xx_hal.h"
/* Private includes*/
/* USER CODE BEGIN Includes */
#include "stm32u5xx_nucleo.h"
/* USER CODE END Includes */
/* Exported types*/

The main application can call the generated LPBAM application from any application file. The LPBAM application is included in the main.c file.

In the main.c file, add «#include "lpbam_tempsens.h"» in the "USER CODE BEGIN Includes" user section.

```
/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include "lpbam_tempsens.h"
/* USER CODE END Includes *
```

The main application must declare any buffers used in the LPBAM application. These are then exported to the LPBAM application files.

Add the buffer declarations to the "USER CODE BEGIN P" user section in main.c.

```
LPTIM_HandleTypeDef hlptim1;
/* USER CODE BEGIN PV */
/* Buffers declaration */
uint16_t Threshold1_Data_Buffer[512] = {0U};
uint16_t Threshold2_Data_Buffer[512] = {0U};
/* USER CODE END PV */
/* Private function prototypes -----*/
```

Any additional service can be added under user codes in the main.c file. For this application, an application API is needed that:

- Enters Stop2 mode and checks whether or not the system was in stop mode
- Checks the content of the ADC buffers



These functionalities are added to the "USER CODE BEGIN 4" user section.

```
/* @brief Enter Stop2 mode and checks whether the system was in Stop2 or not.
 * @param None
* @retval None
*/
static void Enter Stop2 Mode (void)
  /* Enter the system to STOP2 mode */
   HAL RCC PWR CLK ENABLE();
  HAL PWREX EnterSTOP2Mode (PWR STOPENTRY WFI);
  /* Check that the system was resumed from stop 2 */
  if ( HAL PWR GET FLAG(PWR FLAG STOPF) == 0U)
  {
    Error Handler();
  /* Clear stop flag */
   HAL PWR CLEAR FLAG(PWR FLAG STOPF);
  /* Check that stop flag is cleared */
  if ( HAL PWR GET FLAG(PWR FLAG STOPF) != 0U)
   Error Handler();
  }
/**
 \star @brief Verify whether the buffer has a value greater than the threshold or not.
  * @param Buffer
                   Pointer to the buffer that contains the data to be checked.
  * @param BufferSize Size of the buffer to be checked.
 * @param Threshold The Threshold value.
 * @retval Comparaison status.
 */
static uint32 t ADC Buffer Check(void* Buffer, uint32 t BufferSize, uint16 t Threshold)
  uint32 t count = 0;
  /* Repeat for all buffer size */
  while (count < BufferSize)
  {
    /* Check if the buffer value is greater than threshold ^{\star/}
   if ( *(uint16 t*)((uint16 t*)Buffer + count) > Threshold )
   {
      return 1;
    }
    count++;
  return 0;
}
```

Add the functions declarations in the main.c under the user section named "USER CODE BEGIN PFP".

```
/* USER CODE BEGIN PFP */
static void Enter_Stop2_Mode(void);
static uint32_t ADC_Buffer_Check(void* Buffer, uint32_t BufferSize, uint16_t ThresholdValue);
/* USER CODE END PFP */
```

To reach the lowest power consumption, all pins unused by the LPBAM scenario are configured in analog mode, including the debug pins. The user can use LEDs to observe the project runtime status.

In the core of the main() API, the LEDs must be initialized using the BSP layer inside the "USER CODE BEGIN 2" user section.

```
/* USER CODE BEGIN 2 */
/* Initialize LED1 and LED3 : GREEN and RED leds */
BSP_LED_Init(LED1);
BSP_LED_Init(LED3);
```

After initializing the LEDs, the user calls the LPBAM generated APIs to initialize, build, link, and start the LPBAM application within the "USER CODE BEGIN 2" user section.

The call sequence of the LPBAM application-generated APIs is unique. It is mandatory to follow this to make any LPBAM application run.



```
/* LPBAM TempSens application init */
MX_TempSens_Init();
/* LPBAM TempSens application MultiThres scenario init */
MX_TempSens_MultiThres_Init();
/* LPBAM TempSens application MultiThres scenario build */
MX_TempSens_MultiThres_Build();
/* LPBAM TempSens application MultiThres scenario link */
MX_TempSens_MultiThres_Link(&handle_LPDMA1_Channel1);
/* LPBAM TempSens application MultiThres scenario start */
MX_TempSens_MultiThres_Start(&handle_LPDMA1_Channel1);
```

At this point, the LPBAM application is operating. The lowest-power mode must then be entered to ensure the functional aspects.

For this application and device, Stop2 is the lowest-power mode.

/* Enter Stop2 mode */
Enter Stop2 Mode();

At this point, the LPBAM application is still functional, while reaching the lowest power consumption. The power consumption is measured during this step. See detail in Section 3: Power consumption measurement .

After the detection of the low thresholds, the ADC analog watchdog generates its interrupt to allow the system to exit Stop2 mode. The LPBAM application is still running, as it is not affected by entering and exiting low power mode.

For this application, the user calls the following API in the main application's "USER CODE BEGIN 2" user section. This stops, unlinks and de-initializes the LPBAM application.

```
/* LPBAM TempSens application MultiThres scenario stop */
MX_TempSens_MultiThres_Stop(&handle_LPDMA1_Channel1);
/* LPBAM TempSens application MultiThres scenario unlink */
MX_TempSens_MultiThres_UnLink(&handle_LPDMA1_Channel1);
/* LPBAM TempSens application MultiThres scenario de-init */
MX_TempSens_MultiThres_DeInit();
```

At this point, all used resources are deinitialized and can be reused cleanly in the main application.

To ensure that the ADC analog watchdog thresholds are changed correctly, a check of buffer data converted is implemented for each threshold buffer. If any error is detected during this step, the applicative code calls the error handler API.

```
/* Check that the ADC_Buffer contains a value greater than the threshold1 */
if (ADC_Buffer_Check(Threshold1_Data_Buffer, 512, 1005) == 0U)
{
    Error_Handler();
}
/* Check that the ADC_Buffer contains a value greater than the threshold2 */
if (ADC_Buffer_Check(Threshold2_Data_Buffer, 512, 1100) == 0U)
{
    Error_Handler();
}
```

Inside the error handler, the application turns on the red LED. This is implemented under the "USER CODE BEGIN Error_Handler_Debug" user section.

```
/ * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    ______disable_irq();
    /* Turn LED3 on */
    BSP_LED_On(LED3);
    while (1)
```

} /* USER CODE END Error_Handler_Debug */

At this point, the main application is ready to manage the LPBAM application and its execution state (fail or success).

It is up to the user to export the main application user buffers, in the lpbam_tempsens_multithres_build.c file. For this, the buffer declaration is added under the "USER CODE BEGIN EV" user section.

The trigger signal management depends on the user application. The user must therefore add starting and stopping trigger signal APIs.

For this application, the ADCs LPTIM PWM signal is the conversion trigger.

Starting PWM signal generation is done by a call from the lpbam_tempsens_multithres_config.c file, under the "USER CODE BEGIN TempSens_MultiThres_Start" user section.

```
/* USER CODE BEGIN TempSens_MultiThres_Start */
/* LPBAM LPTIM1 start PWM generation */
if (HAL_LPTIM_PWM_Start(&hlptim1, LPTIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE END TempSens MultiThres Start */
```

Stopping PWM signal generation is done by a call from the lpbam_tempsens_multithres_config.c file, under the "USER CODE BEGIN TempSens_MultiThres_Stop" user section.

```
/* USER CODE BEGIN TempSens_MultiThres_Stop */
/* LPBAM LPTIM1 stop PWM generation */
if (HAL_LPTIM_PWM_Stop(&hlptim1, LPTIM_CHANNEL_1) != HAL_OK)
{
Error_Handler();
}
/* USER CODE END TempSens_MultiThres_Stop */
```

The ADC calibration depends on the device and application needs.

The user may add the calibration call (if needed) under the "USER CODE BEGIN ADC4_Init Calibration" user section.

```
/* USER CODE BEGIN ADC4_Init Calibration */
/*
* ADC4 Calibration
*/
if (HAL_ADCEx_Calibration_Start(&hadc4, ADC_CALIB_OFFSET, ADC_SINGLE_ENDED) != HAL_OK)
{
Error_Handler();
}
/* USER CODE END ADC4 Init Calibration */
```

For the ADC peripheral used by this LPBAM application, it is mandatory to add the enable of V_{DDA} power supply under the "USER CODE BEGIN ADC4_MspInit 0" user section.

```
/* USER CODE BEGIN ADC4_MspInit 0 */
/* Enable VDDA supply for ADC */
HAL_PWREx_EnableVddA();
/* USER CODE END ADC4_MspInit 0 */
```

For each interrupt callback generated in the lpbam_tempsens_multithres_config.c file, it is advisable to introduce code to report the main-application status. This code should be added under the callback user section.

For this application, the error (red) LED is added under the "USER CODE BEGIN Thresholdx_Config_DMA_Error_Callback" user section.



```
/**
 * @brief Thresholdx_Config queue dma error callback
 * @retval None
 */
static void MX_Thresholdx_Config_Q_DMA_Error_Callback(DMA_HandleTypeDef *hdma)
{
    /* USER CODE BEGIN Thresholdx_Config_DMA_Error_Callback */
    /* Turn on LED3 */
    BSP_LED_On(LED3);
    /* USER CODE END Thresholdx_Config_DMA_Error_Callback */
}
```

At this point, the project is ready and finalized. It can run safely with reporting any issues during run and low power modes.

As the DMA access depends on device SOC integration, it is mandatory to check the accessibility of the DMA channel instance. This impacts the operating aspects of the LPBAM application.

For this application, the DMA channel used can access only SRAM4 in the same power domain.

According to the preferred IDE supported by this application, in the IAR project under files:

- 1. Right click on the LPBAM_ADC_TempSense
- 2. Click on options \rightarrow "Linker" \rightarrow "Memory Regions"
- Change the start address of the RAM to 0x28000000 and the end address to 0x28003FFF (range of address accessible by the DMA channel)

ptions for node	"LPBAM_ADC	_TempSense"					
Category: General Options						Factory S	Settings
Static Analysis Runtime Checking C/C++ Compilei		#define Diag	nostics	Checksum	Encodings	Extra	Options
Assembler Output Convert	er III	Config Library	Input	Optimizations	Advanced	Output	List
Custom Build Build Actions	Linker config	guration file editor	tion file -		;	<	
Linker	Vector Table	Memory Regions	Stack/	Heap Sizes			
Simulator		Start:		End:			
CADI CMEIS DAD	ROM	0x080	00000	0x081F	FFFF		
GDB Server I-jet	RAM	0×280	00000	0x2800	3FFF		^
J-Link/J-Trace							
Nu-Link					Save		

Figure 44. "Memory Regions" modification in IAR

In the Keil® project:

- 1. Open "Options" \rightarrow "Target"
- 2. Change the start of the IRAM1 to 0X28000000 and the size to 0x4000

Figure 45. "Memory Regions" modification in Keil®

🔣 C:\Users\tayssir zardi\Downloads\LPBAM_ADC_TempSense\MDK-ARM\LPBAM_ADC_TempSense.uvprojx - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools S	VCS Window Help
19 29 19 (4) 2 (2) 2	18] 詳 詳 //E //E 19 RCC_ClkinitStruct.SYSCL 🔍 🗟 🥐 🔍 🔹 🌢 🔿 🔗 🚷 🚽 🔝 🔦
🧼 🕮 🕮 🔹 - 🗮 🛛 🗱 LPBAM_ADC_TempSense 🖂 💦 🛔	1 🗄 🔶 🗇 🏟
Project 📮 🛛	Options for Target 'LPBAM_ADC_TempSense' X
Project: LPBAM_ADC_TempSense	
😑 🚂 LPBAM_ADC_TempSense	Device Target Output Lusting User (C/C++ (ACb) Asm Linker Debug Utilities
Application/MDK-ARM	STMicroelectronics STM32U575ZITx Code Generation
🕀 🧰 Application/User/Core	ARM Compiler: Use default compiler version 6
Application/User/LPBAM/TempSens	Software Model: Non-Secure Mode
🕀 🛅 Doc	Operating system: None
Drivers/BSP/STM32U5xx_Nucleo	System Viewer File: Use MicroLIB Big Endian
Drivers/STM32U5xx_HAL_Driver	STM32U5xx.svd Floating Point Hardware: Single Precision
Drivers/CMSIS	Use Custom File
🗉 🛄 Utilities	
CMSIS	Read/Only Memory Areas
	default off-chip Start Size Startup default off-chip Start Size NoInit
	C ROM1: C RAM1: C
	C RAM2:
	C RAM3:
	on-chip on-chip
	IROM1: 0x8000000 0x200000
	OK Cancel Defaults Help

 In the STM32CubeIDE go to the STM32U575ZITXQ_FLASH.Id and configure the RAM ORIGIN = 0x28000000 and the LENGTH = 16 Kbytes

Figure 46. "Memory Regions" modification in the STM32CubeIDE

workspace_1.9.0 - LPBAM_ADC_TempSense/STM32U575ZITXQ_FLASH.Id - STM32CubeIDE File Edit Source Refactor Navigate Search Project Run Window Help 🖻 🗖 😭 STM32U575ZITXQ_FLASH.Id 🗵 Project Explorer × 🖻 🕏 🍞 🖇 🛛 37 38/* Highest address of the user mode stack */ LPBAM_ADC_TempSense (in STM32Cu) 39_estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */ > 🔊 Includes 40 Application 41_Min_Heap_Size = 0x200 ; /* required amount of heap */ V 🕞 User 42_Min_Stack_Size = 0x400 ; /* required amount of stack */ > 🗁 Core 43 > 🕞 LPBAM 44 /* Memories definition */ 45 MEMORY > 🗁 Startup > 🗁 Doc 46 { RAM (xrw) : ORIGIN = 0x28000000, LENGTH = 16K FLASH (rx) : ORIGIN = 0x08000000, LENGTH = 2048K 47 > 🗁 Drivers 48 FLASH (rx) > 🗁 Utilities 49 } LPBAM ADC TempSense.ioc 50 STM32U575ZITXQ_FLASH.Id 51/* Sections */ 52 SECTIONS

Placing the "RAM" section in the SRAM accessible by the DMA channel, is not the only way to make the LPBAM application functional. It is, however, the simplest method.



3

Power consumption measurement

To measure the power consumption of the application, the STM32 Power Shield application is used to supply the Nucleo board.

PowerShield is a plug-and-play solution intended to ease power consumption measurements.



Figure 47. Current consumption signal during project execution





Revision history

Date	Version	Changes
03-Oct-2022	1	Initial release.
14-Dec-2022	2	Updated: Introduction Section 1 General information Section 2.3 STM32CubeMX LPBAM TempSense application building
23-Aug-2023	3	Updated: Introduction STM32CubeMX LPBAM TempSense application building Scenario queue building How to call APIs
07-Feb-2024	4	Title updated

Table 1. Document revision history



Contents

1	Gen	eral information	2
2	LPB	AM TempSense application description	3
	2.1	Principle	3
	2.2	Implementation	3
	2.3	STM32CubeMX LPBAM TempSense application building	4
	2.4	Scenario queue building	13
	2.5	How to call APIs	22
3	Pow	er consumption measurement	28
Rev	ision	history	29



List of tables

Table 1. D	Document revision history	29
------------	---------------------------	----



List of figures

Figure 1.	SM32CubeMX tool scope	. 2
Figure 2.	Block diagram	. 4
Figure 3.	NUCLEO-U575ZI-Q board selection	. 4
Figure 4.	Peripheral initialization.	5
Figure 5.	TrustZone [®] deactivated	. 5
Figure 6.	Project settings	. 5
Figure 7.	STM32Cube MCU packages	. 6
Figure 8.		. 6
Figure 9.	Clock configuration	. 7
Figure 10.	Changing the timebase source	. 7
Figure 11.	LPBAM scenario and configuration selection	. 7
Figure 12.	Application and scenario naming.	. 8
Figure 13.	PWR configuration	. 8
Figure 14.	"ADC4 Mode" and "Advanced Settings for LPBAM" configuration	. 9
Figure 15.	ADC4 parameter settings	. 9
Figure 16.	ADC4 NVIC settings	10
Figure 17.	LPTIM1 activation and parameter settings	10
Figure 18.	User constants adding	11
Figure 19.	I PTIM1 and ADC4 clock configuration	11
Figure 20.	"I PDMA1 Mode" and "CH0" configuration	12
Figure 21.	"I PDMA1 Advanced Settings" for I PBAM configuration	12
Figure 22.	I PDMA1 NVIC settings	13
Figure 23.	Queue naming	13
Figure 24.	Queue1 building	14
Figure 25		14
Figure 26.	Queue2 naming	14
Figure 27	Queue2 huilding	15
Figure 28		15
Figure 29		16
Figure 30	First node configuration	16
Figure 31		17
Figure 32		17
Figure 33	Third node "Analog Watchdog1" configuration	18
Figure 34		10
Figure 34.		10
Figure 35.		19
Figure 36.		19
Figure 37.		19
Figure 38.		20
Figure 39.		20
Figure 40.		21
Figure 41.		21
Figure 42.		21
Figure 43.		22
Figure 44.	"Internery Regions" modification in IAR.	26
Figure 45.		21
Figure 46.		21
Figure 47.		28
Figure 48.	Current consumption signal during stop 2 mode	28

IMPORTANT NOTICE - READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2024 STMicroelectronics – All rights reserved