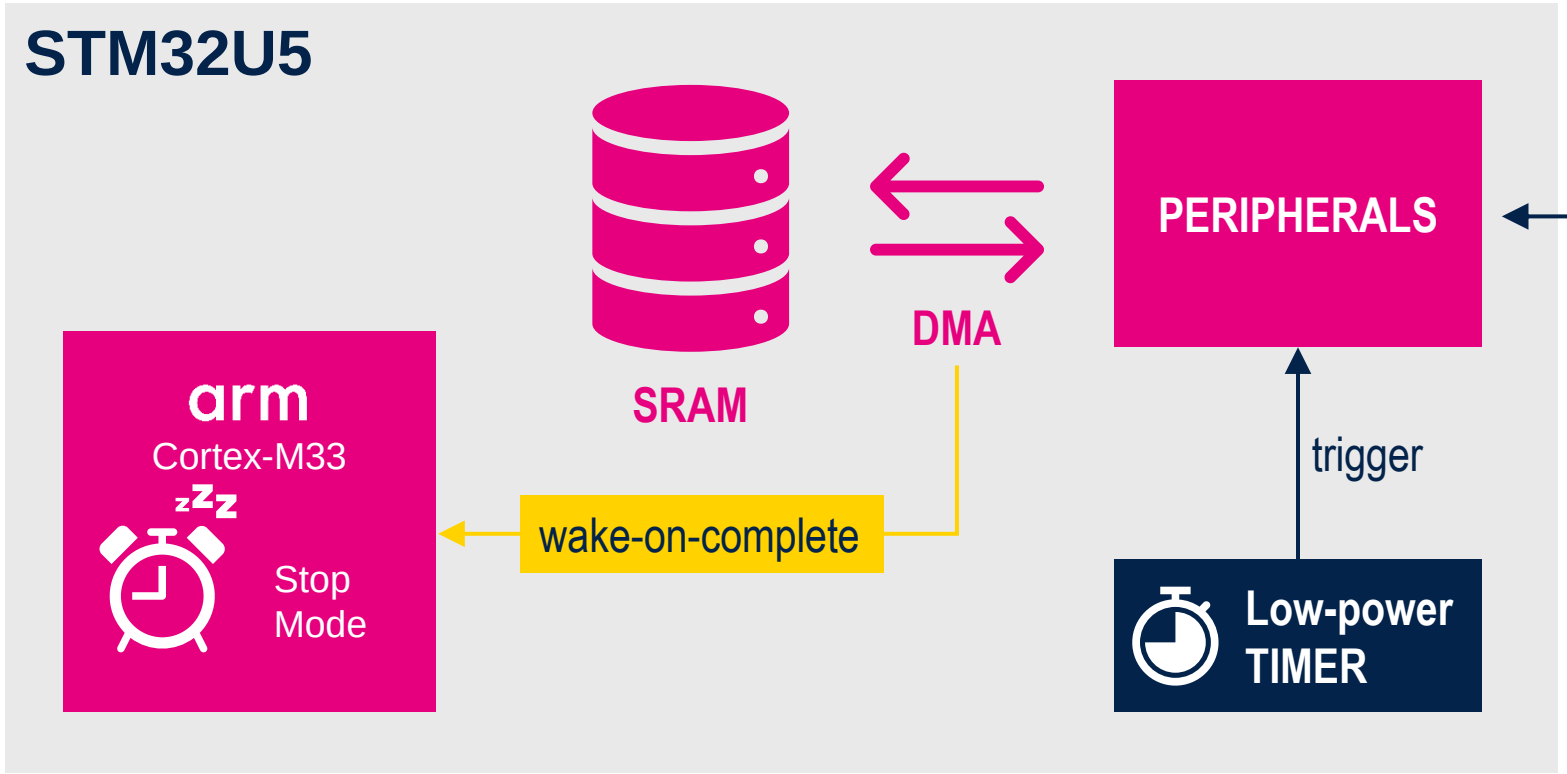
LPBAM – Low Power Background Autonomous Mode

Manuel Marcias – South EMEA Field Application Engineer – Microntrontroller Products

LPBAM Theory

Cut MCU power consumption by 10

Low Power Background Autonomous Mode (LPBAM)



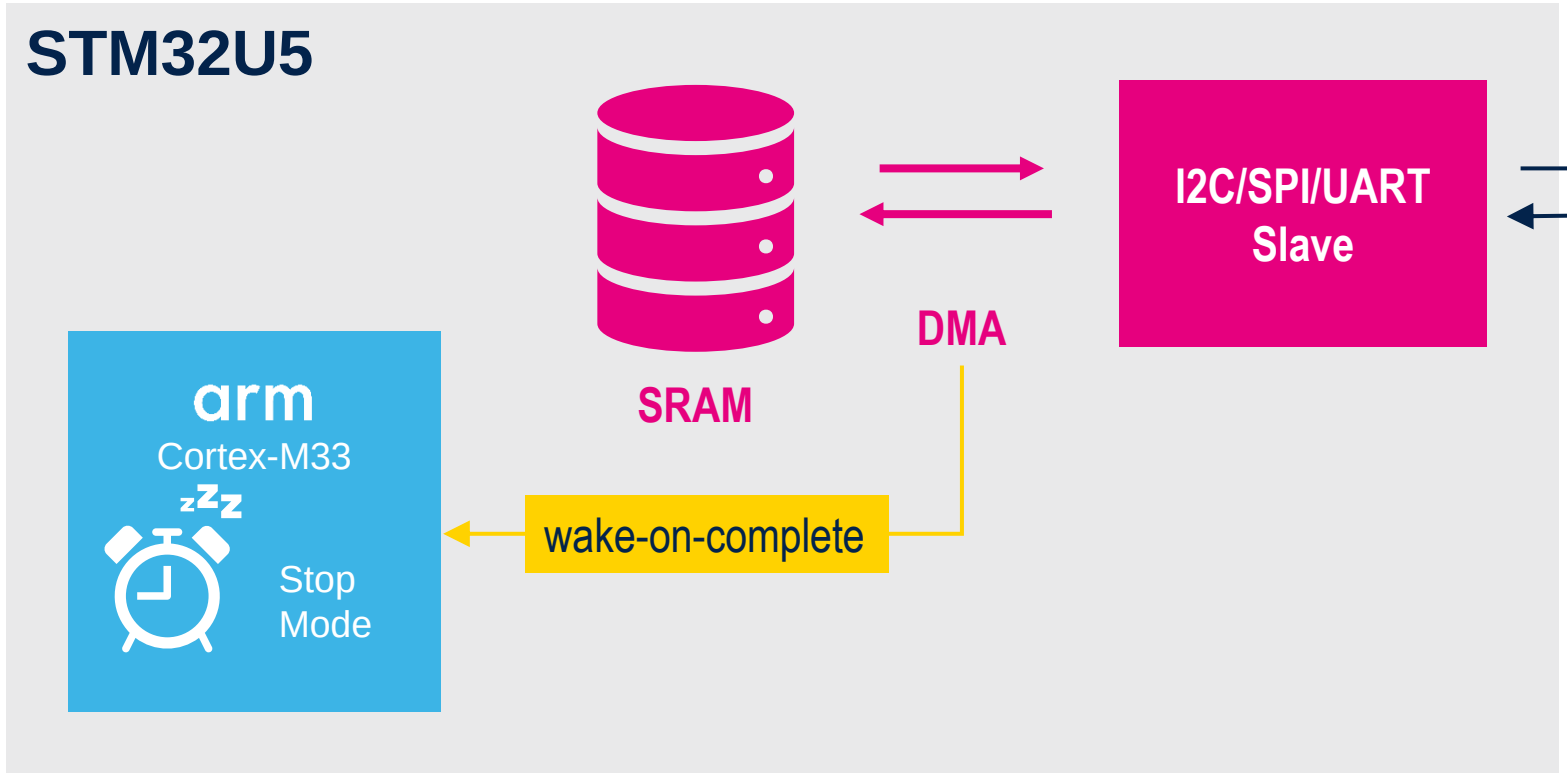
Typical Scenario (w/o LPBAM): Switch between STOP and ACTIVE mode to save power

LPBAM:

- Allows peripherals to be **functional and autonomous** in STOP0/1/2 modes
- **Peripherals works in STOP** mode thanks to **their own independent clock** request capability
- The **peripheral bus clock is automatically switched ON/OFF** by the peripheral

LPBAM: Use Cases

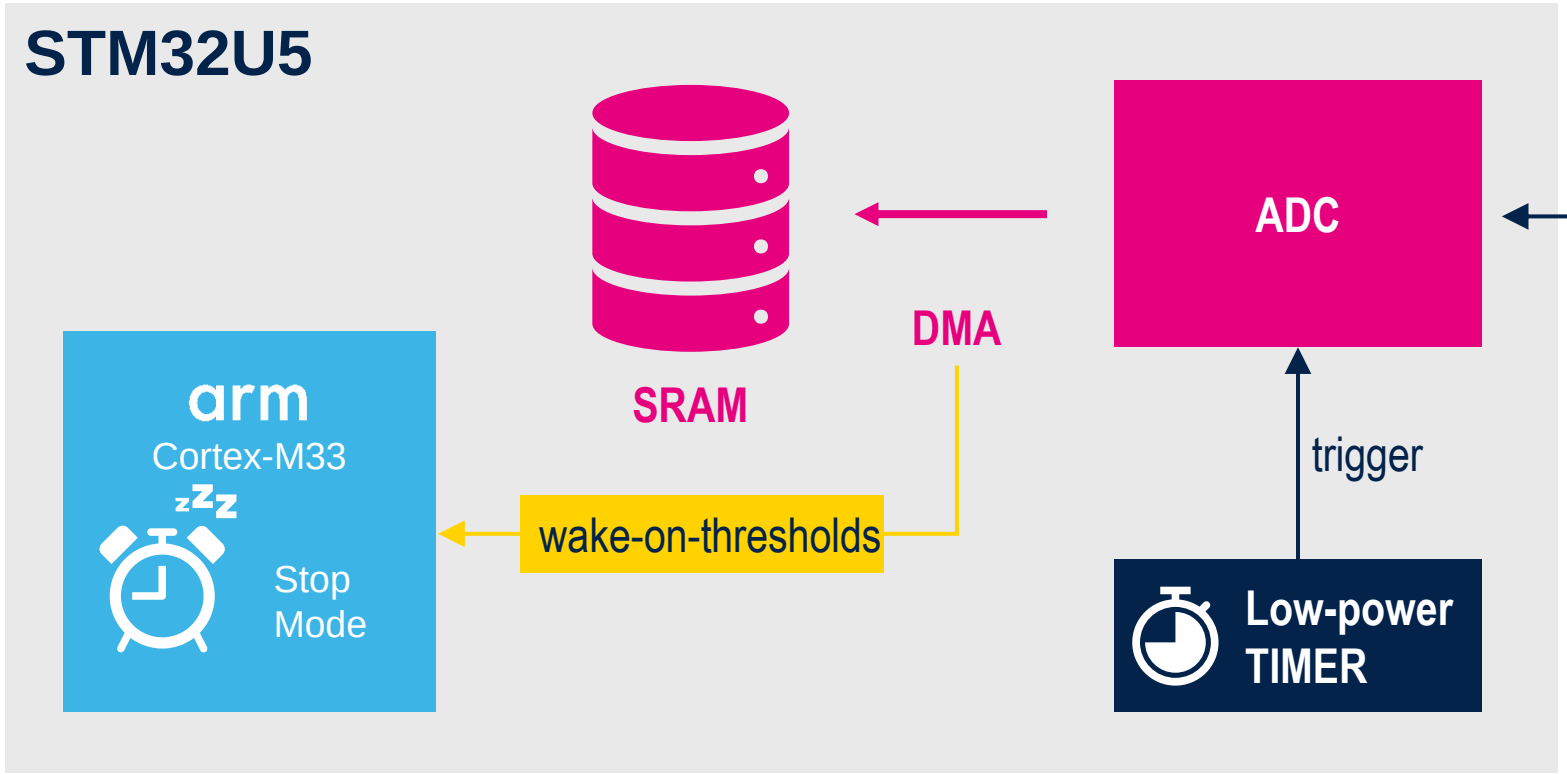
I2C Slave/Master transfer - SPI or UART RX/TX



Wakeup can happen at the end of the transfer, at DMA half buffer event etc.

LPBAM: Use Cases

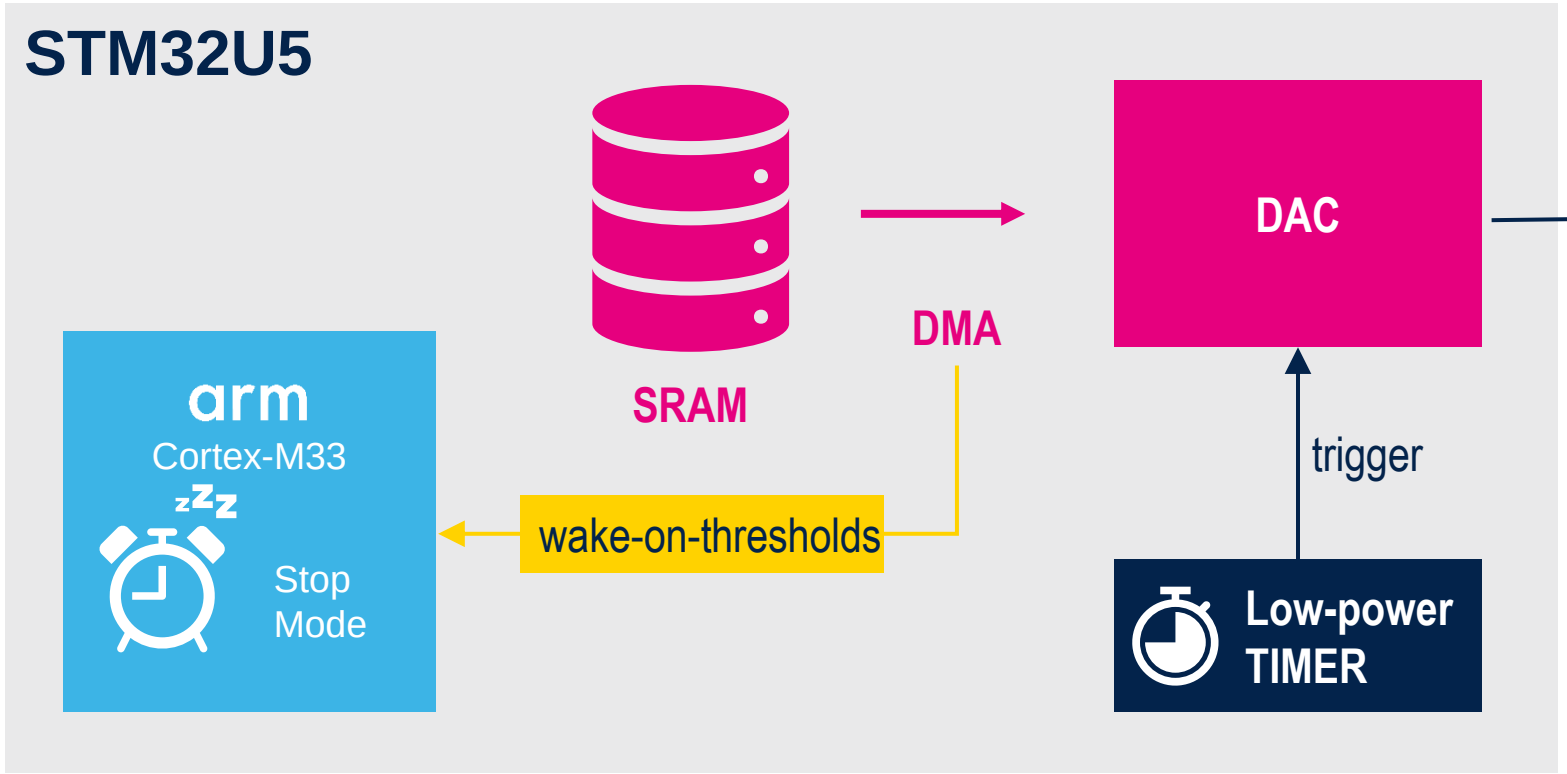
ADC Conversion triggered by a Low Power Timer



Wakeup from Stop Mode on analog watchdog if the ADC conversion is out of the programmed thresholds

LPBAM: Use Cases

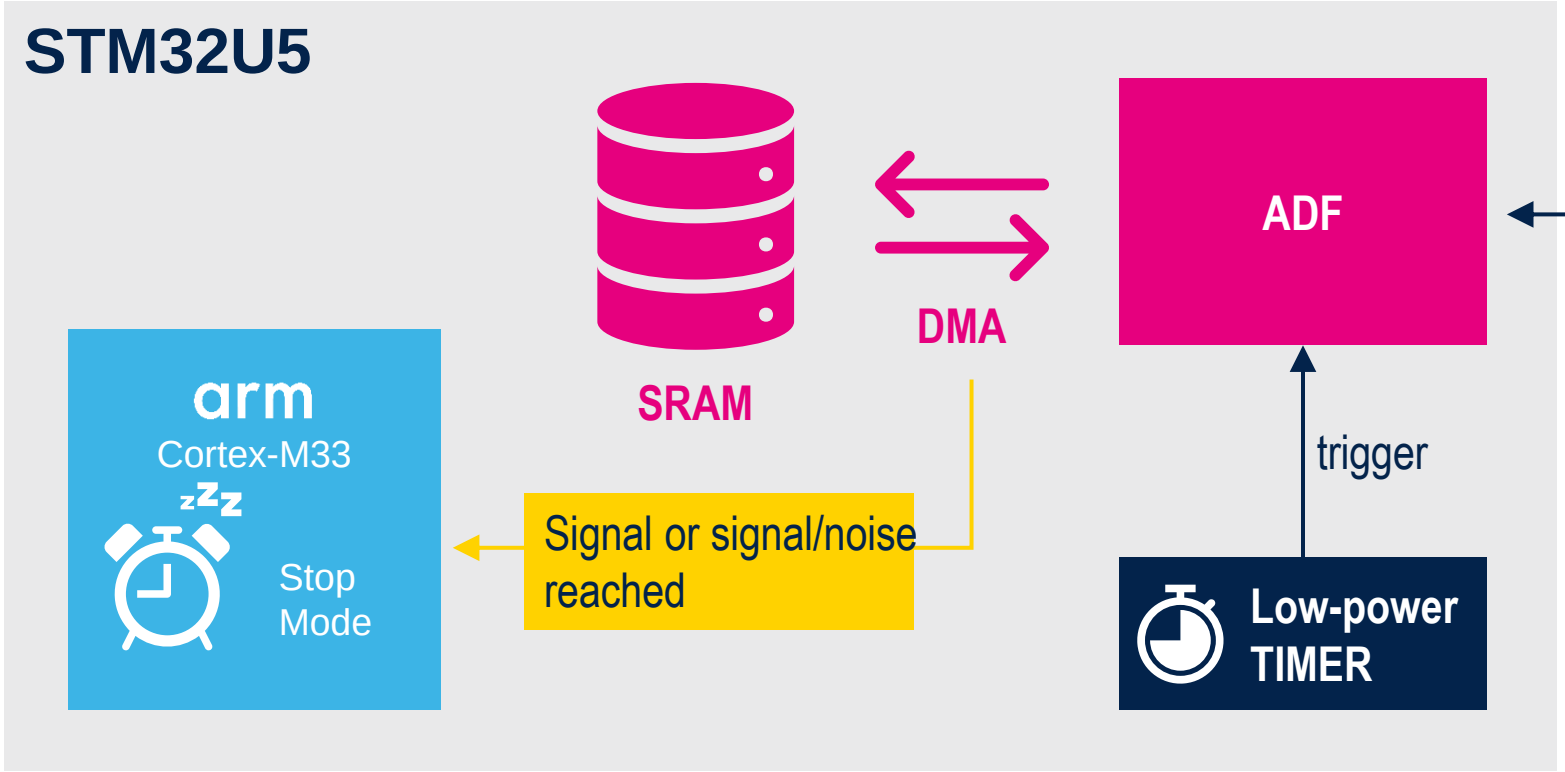
DAC Conversion in Sample&Hold



DAC can be used in S&H so its value should be periodically reloaded

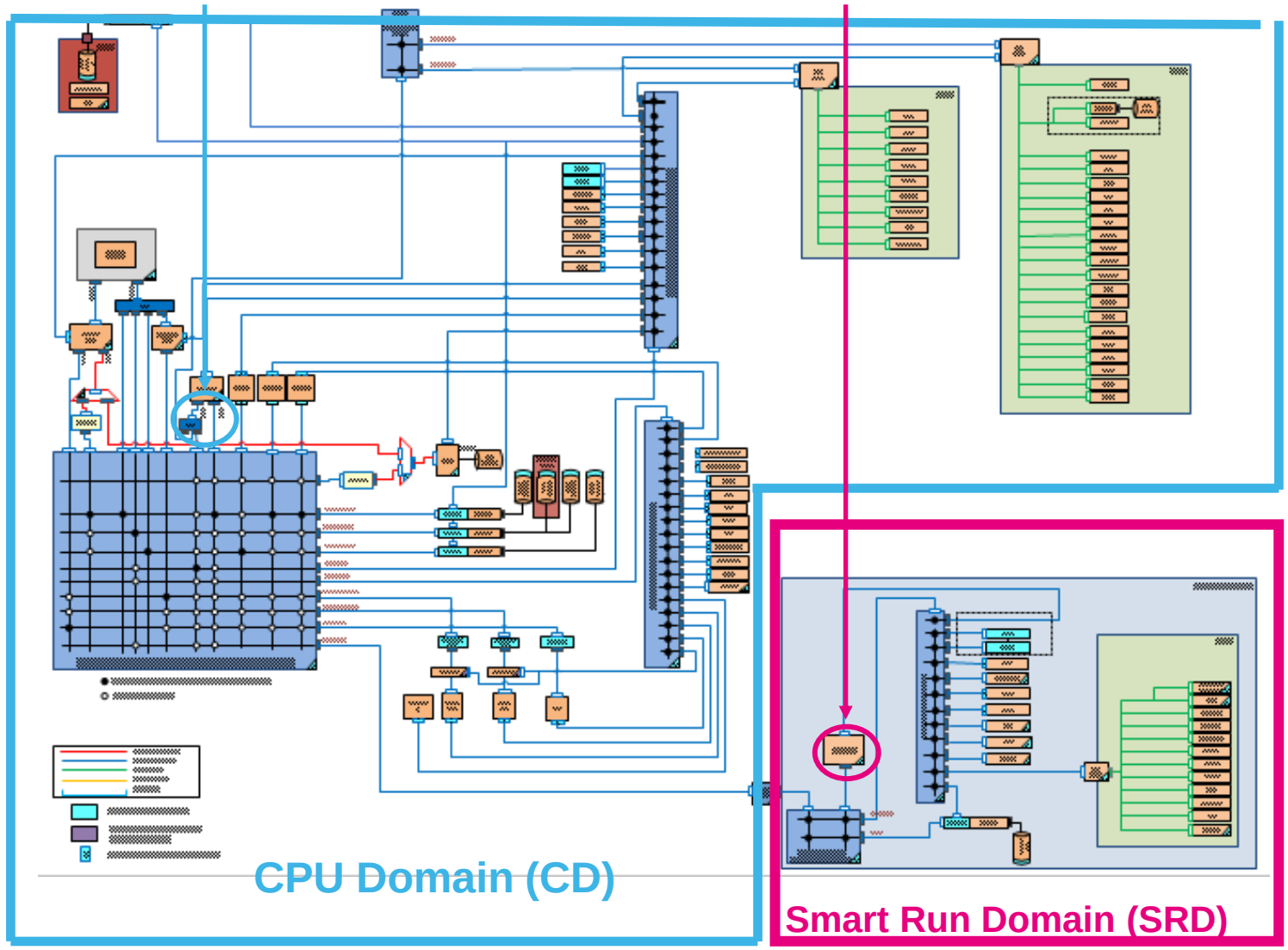
LPBAM: Use Cases

Voice Activity Detection with Audio Digital Filter



Audio Digital Filter (ADF) can detect sound level or SNR. Wake up from stop on sound activity detection

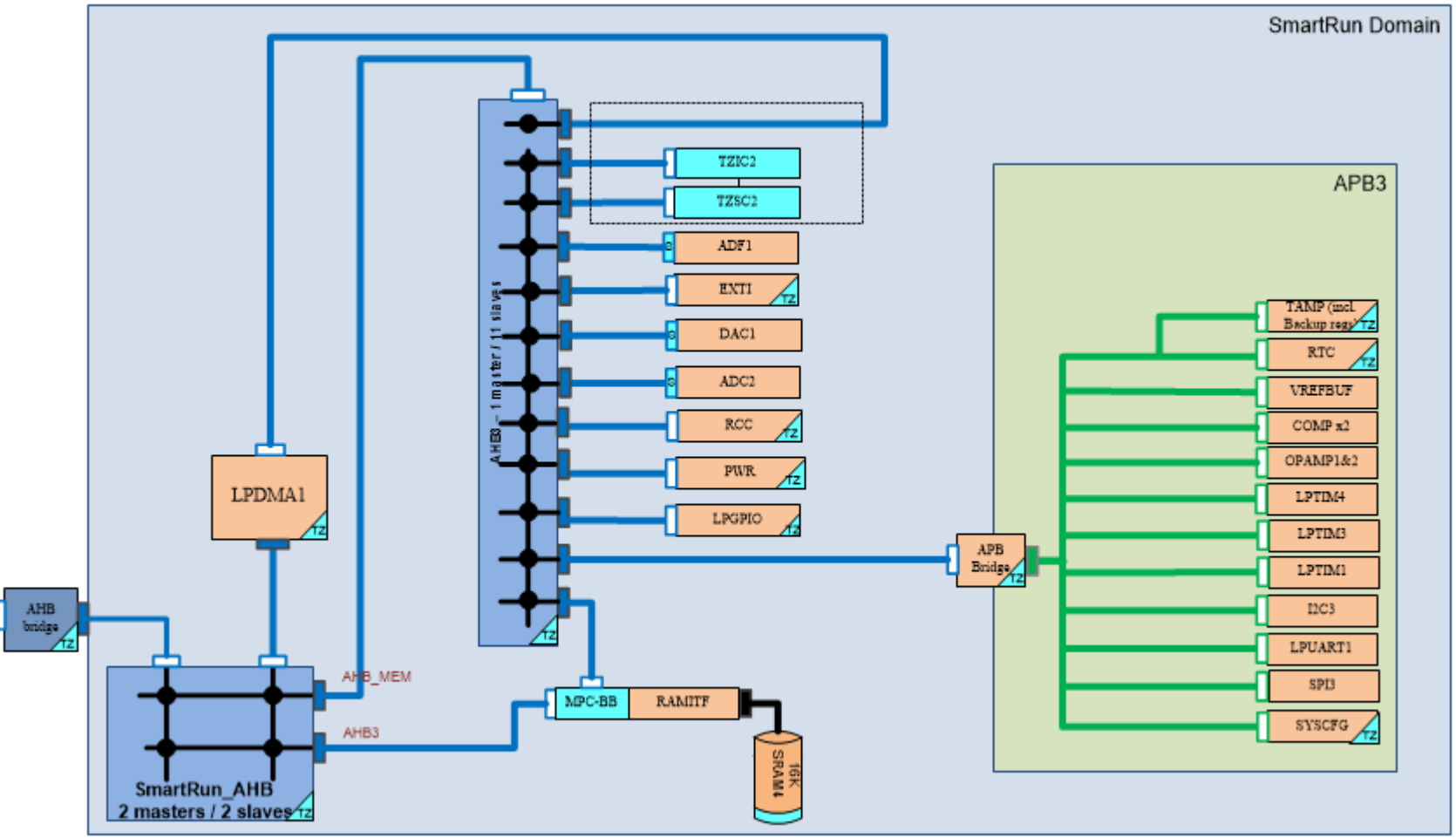
GPDMA & LPDMA



A new DMA IP:

- Single DMA driver
- 2 HW instances
  - LPDMA
  - GPDMA
- Integrated DMAMUX features
- Improved autonomy
  - Own clock request management
  - Flexible intra-channel and inter-channel input/output control

Smart Run Domain



SmartRun Domain (SRD) relies on DMA allowing autonomous operation during low power modes down to STOP2 with LPDMA.

2xMasters:

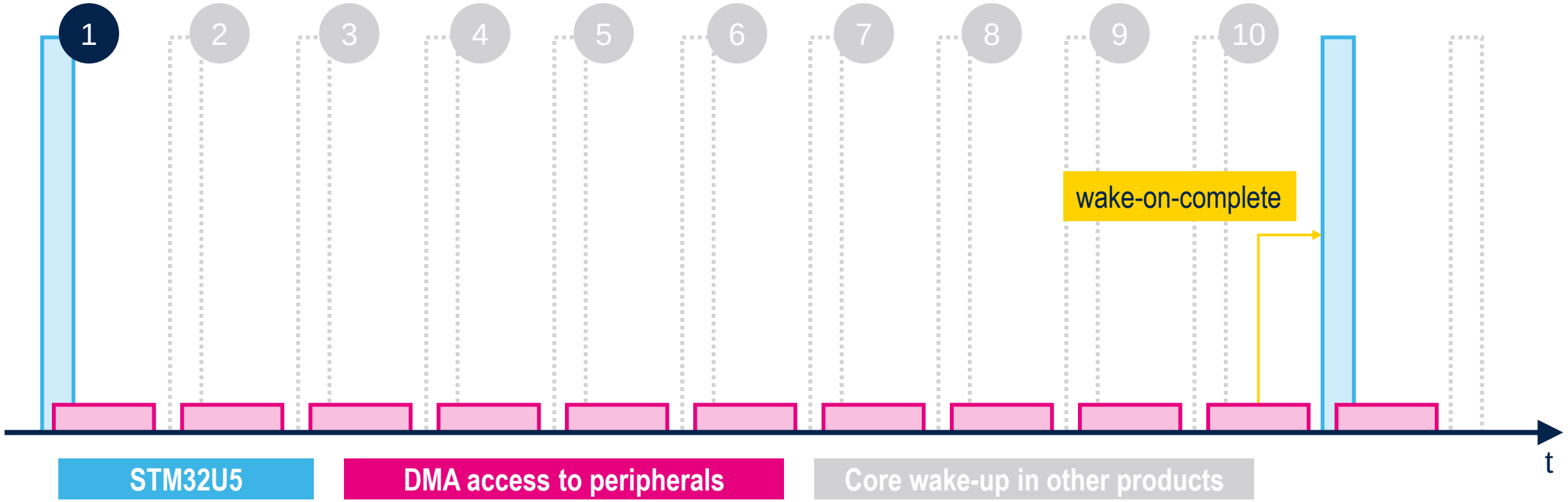
- AHB Bus Matrix
- LPDMA1 (low-power DMA)

2xSlaves

- AHB3 & APB3 peripherals
- SRAM4 (16KB)

Clock Distribution in SRD

The LPBAM avoids waking up the Cortex-M33 core every time we collect data from the peripherals, resulting in energy savings



Peripherals supporting LPBAM

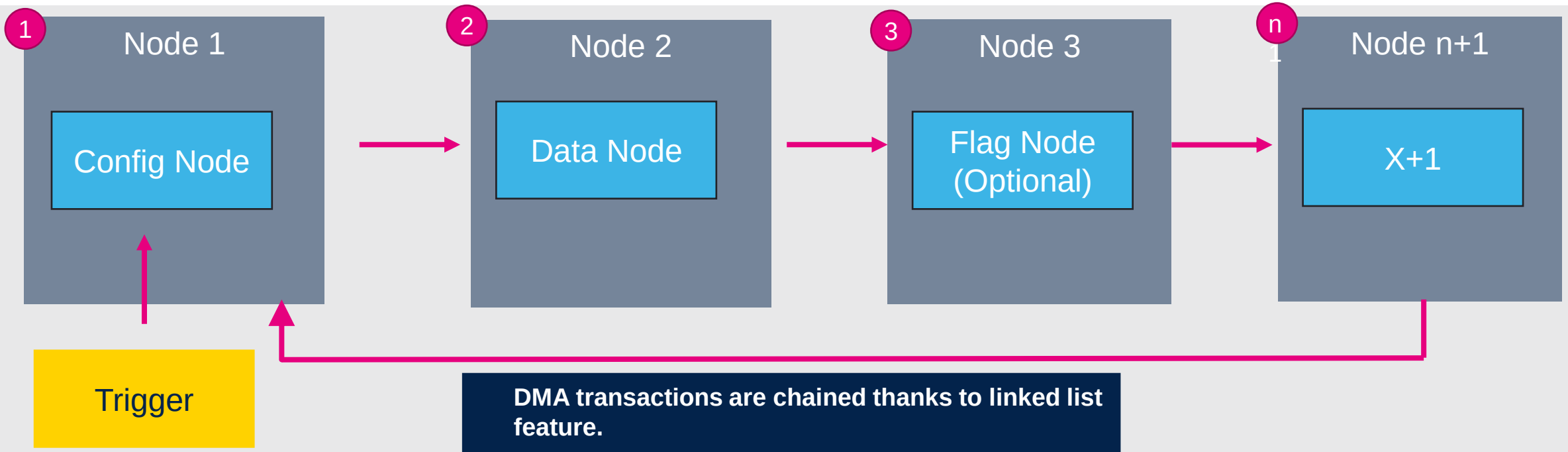
- **Autonomous Peripherals** = they have clock request capability and support DMA transfer in STOP
- **Passive Peripherals**= Can only be reconfigured or used as trigger. Do not support DMA nor clock request

Category	LPBAM Category	Functional down to STOP0/1	Functional Down to STOP2
SRAMs	Passive	SRAM1/2/3/4+BKPSRAM	SRAM4
DMA	Autonomous	GPDMA1,LPDMA1	LPDMA1
Communication peripherals	Autonomous	I2C1/2/3/4 SPI1/2/3 USART1-5 , LPUART1	I2C3 SPI3 LPUART1
Timers	Autonomous Passive	LPTIM1/2/3/4 RTC	LPTIM1/3 RTC
Application peripherals	Autonomous	MDF1,ADF1	ADF1
Analog peripherals	Autonomous Passive	ADC4,DAC1 Comparator,OPAMP,Vrefbus	ADC4,DAC1 Comparator,OPAMP,Vrefbus
I/Os	Passive	GPIO/LPGPIO	LPGPIO

ISR generated by Autonomous Peripheral or Passive Peripherals can wake up STM32U5 from STOP0/1/2 mode

Linked List Recap

LPBAM Bases



LPDMA1 ch0

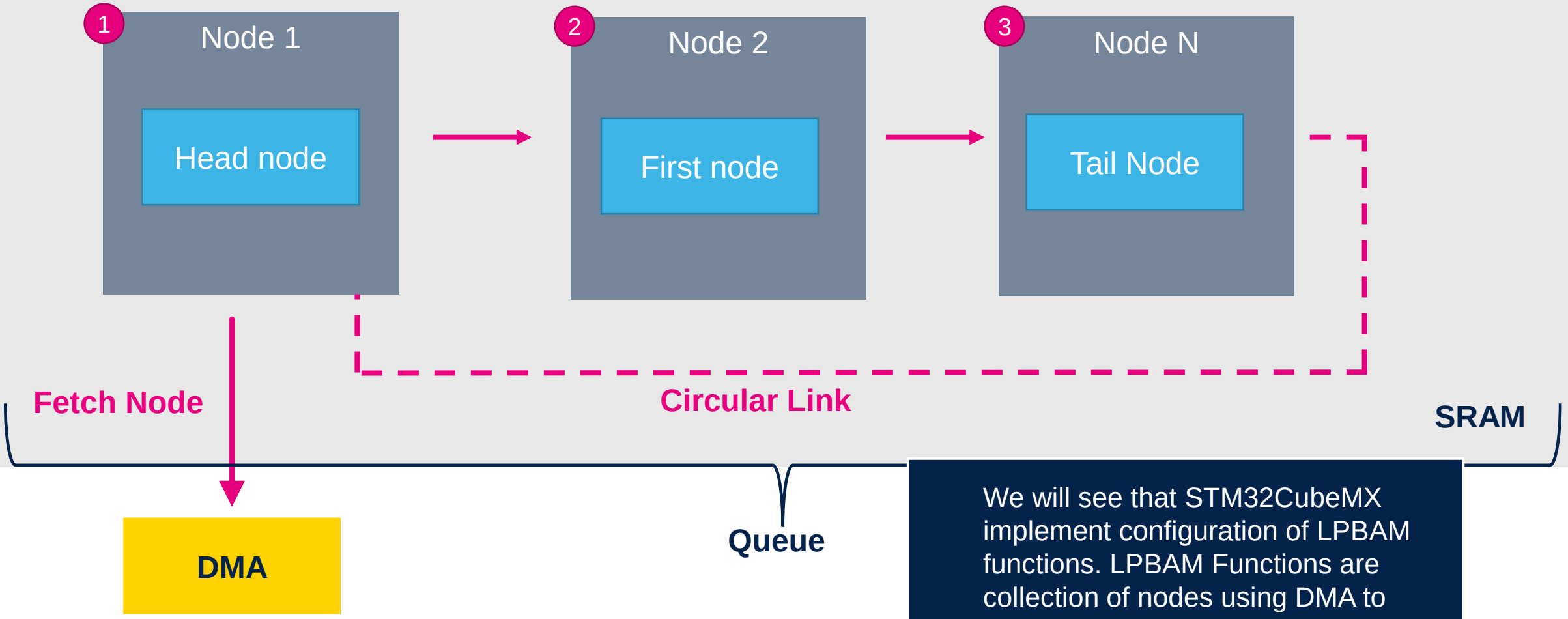
DMA transactions are chained thanks to linked list feature.

LPBAM task is a sequence of DMA transactions conditioned by peripheral request/trigger.

DMA transactions can be a combination of configure peripherals, transmit data or clear flags.

LPBAM task can be executed in an infinite loop (If no error is occurred).

Linked list mode

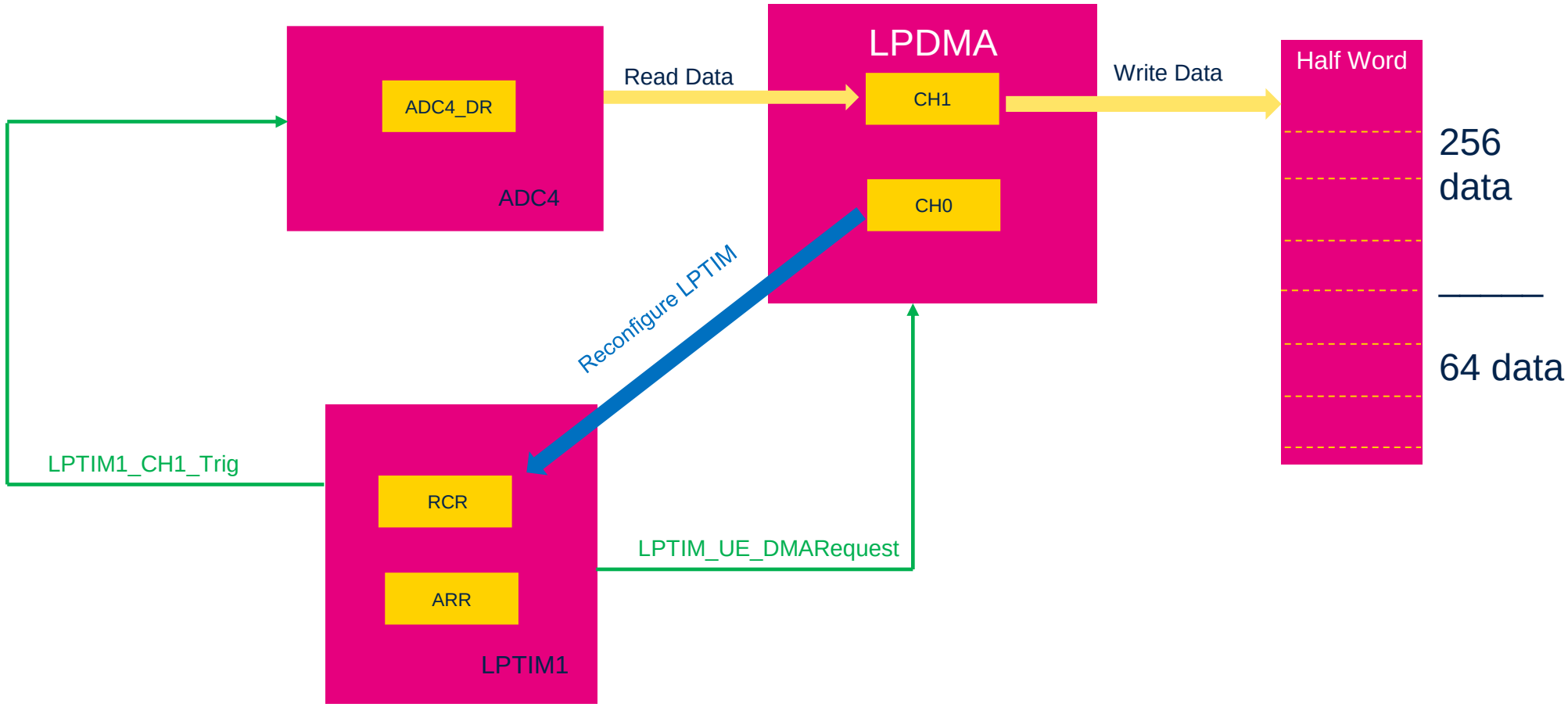


We will see that STM32CubeMX implement configuration of LPBAM functions. LPBAM Functions are collection of nodes using DMA to read/write registers

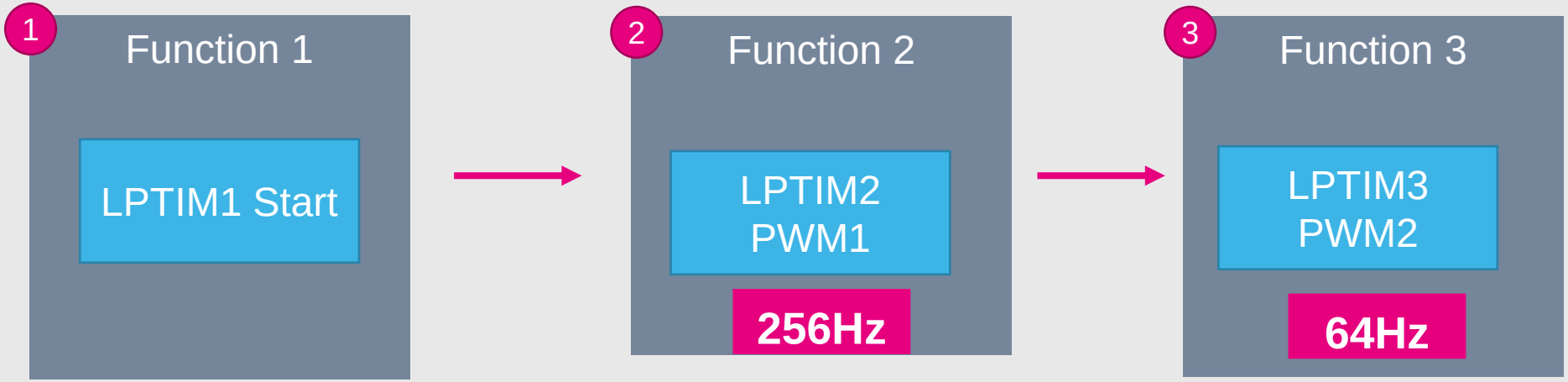
Handson Overview

life.augmented

LPBAM Handson Application



Queue 1



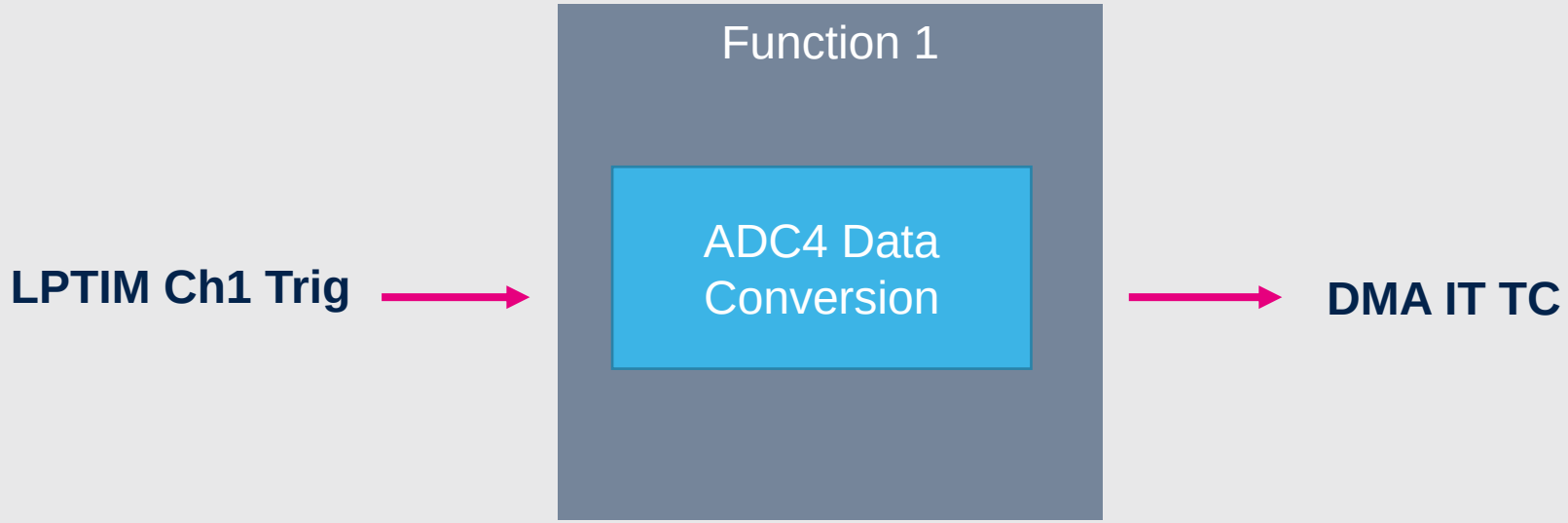
LPDMA1 ch0

One channel One Queue with three functions

LPTIM1 used as a trigger for ADC channel

PWM duty cycle changed by function 2 and 3

Queue 2

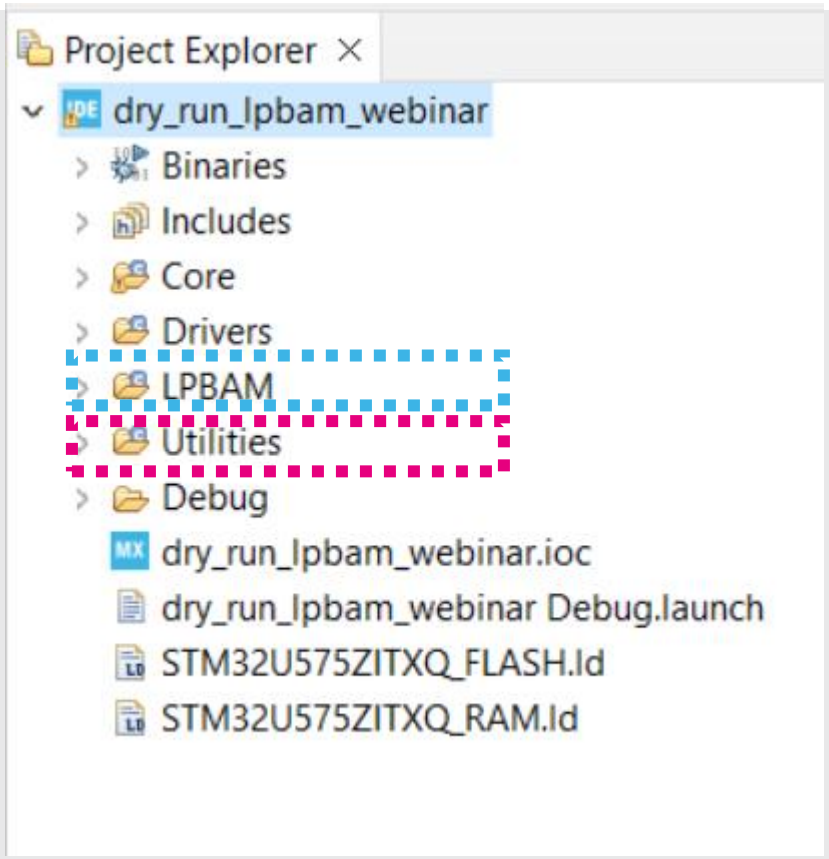


LPDMA1 ch1

- **One channel One function only**
- **ADC4 will start conversion of 320 samples**
- **MCU will be awaken by DMA IT TC**

Generated Code

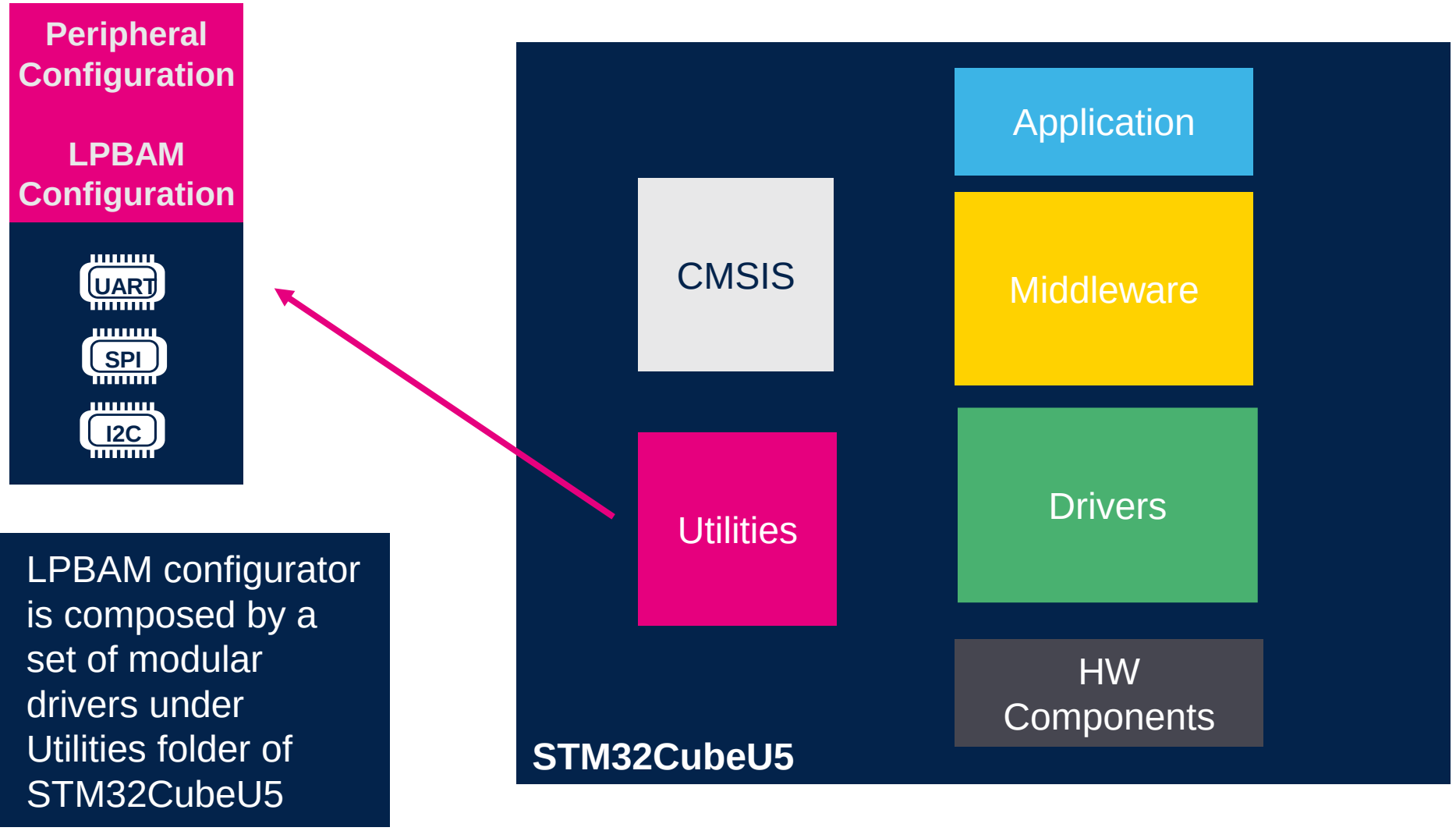
LPBAM Application Project tree



LPBAM is the folder containing LPBAM applications files

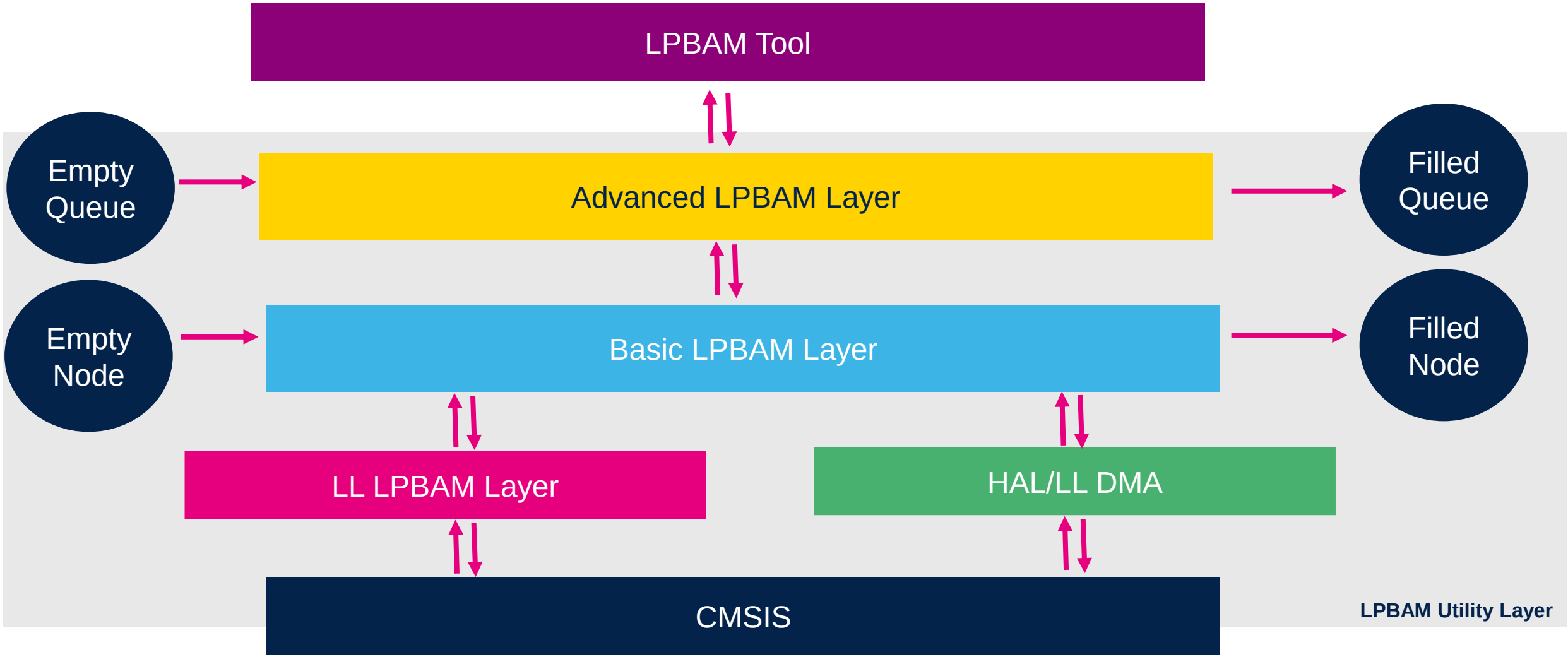
Utility folder contains the LL, Basic and Advanced Driver

STM32CubeU5

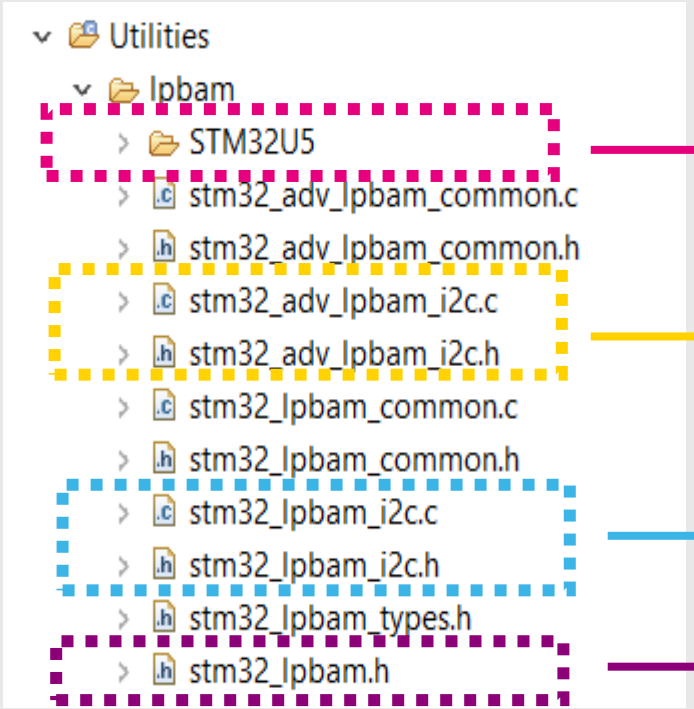


LPBAM configurator is composed by a set of modular drivers under Utilities folder of STM32CubeU5

LPBAM Utility Layer



Utilities Folder



LL Layer Module

Advanced Layer Module

Basic Layer Module

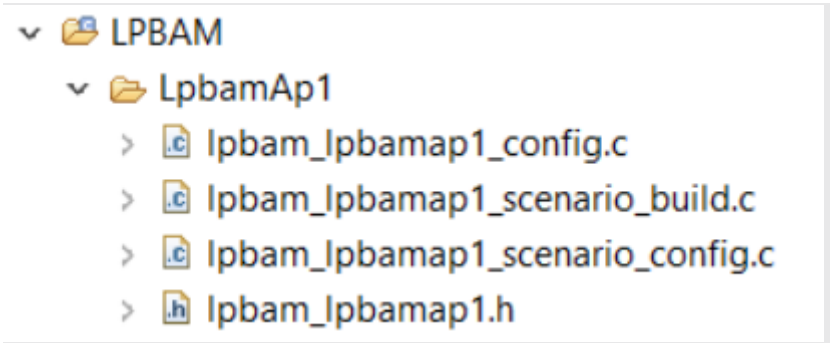
Single Entry Point

LPBAM Utility contains a set of indipendent modules and each module is a pair of basic and advanced files

Each module manages configurability and data trasnfer for a given peripheral

LPBAM application has a single entry points which includes all basic, advanced and common functionalities

LPBAM Application folder



Lpbam_lpbam1.h is the entry point for LPBAM application layer. It includes ADV Layer and must be included in main.c

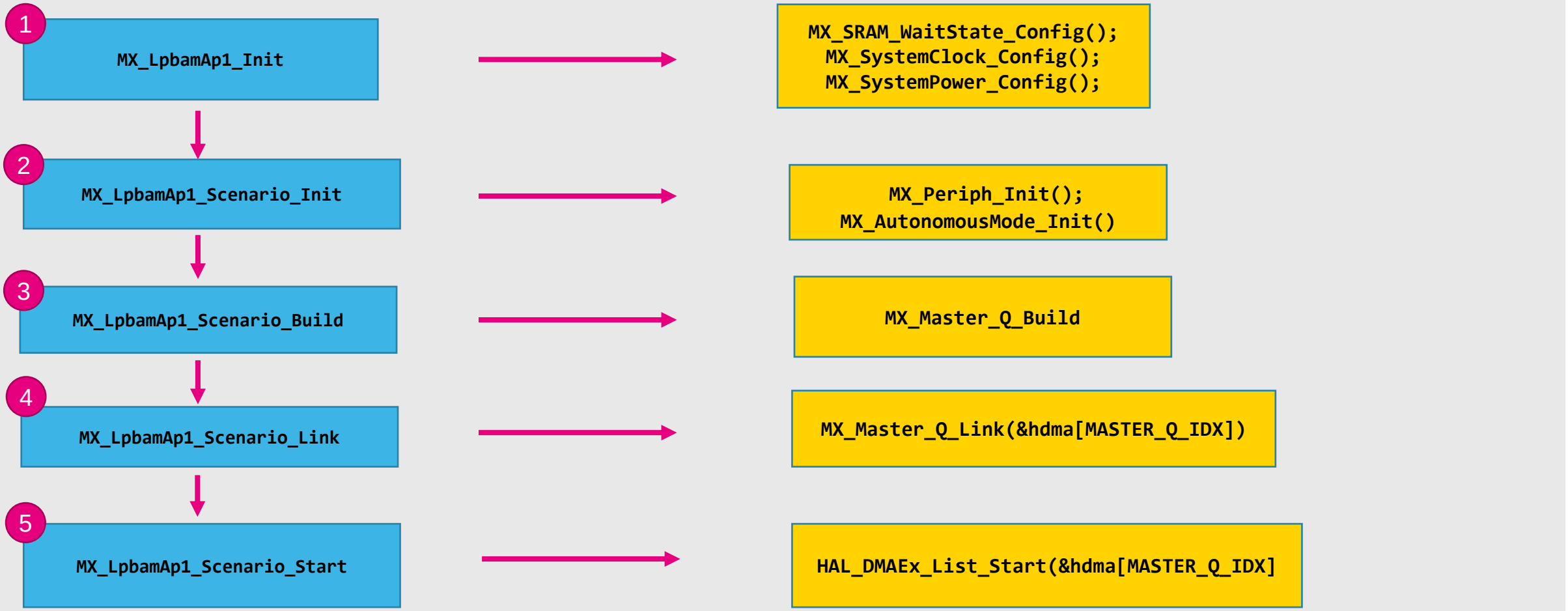
`Lpbam_lpbamp1_config.c`	`MX_LpbamAp1_Init`
`Lpbam_lpbamp1_scenario_build.c`	`MX_LpbamAp1_Scenario_Build`

`Lpbam_lpbamp1_scenario_config.c`	`MX_LpbamAp1_Scenario_Init`
	`MX_LpbamAp1_Scenario_Link`
	`MX_LpbamAp1_Scenario_Start`

`Lpbam_lpbamp1.h`	`#include "stm32_lpbam.h"`

LPBAM User Sequence

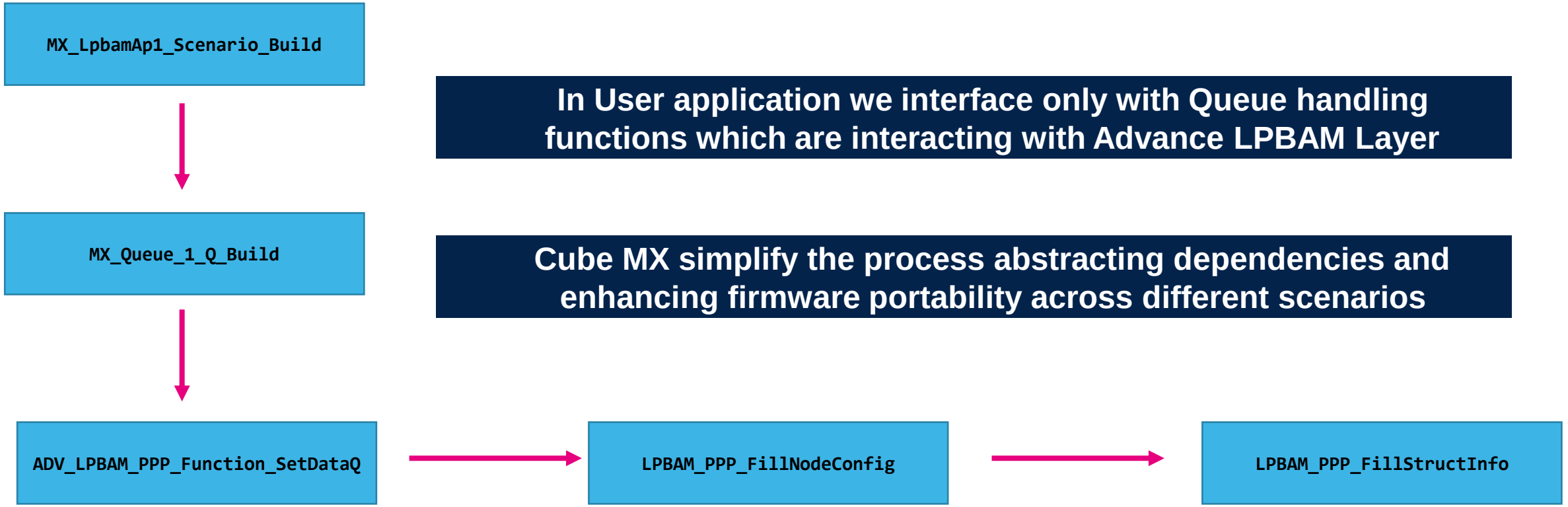
This is typical order of function calls in main.c to start LPBAM Application



life.augmented

LL LPBAM Layer

This is how we move from application to the LL Layers via Advanced and Basic Layers:



LPBAM Configurator

How to start LPBAM Application?

Peripheral Configuration

LPBAM Configuration

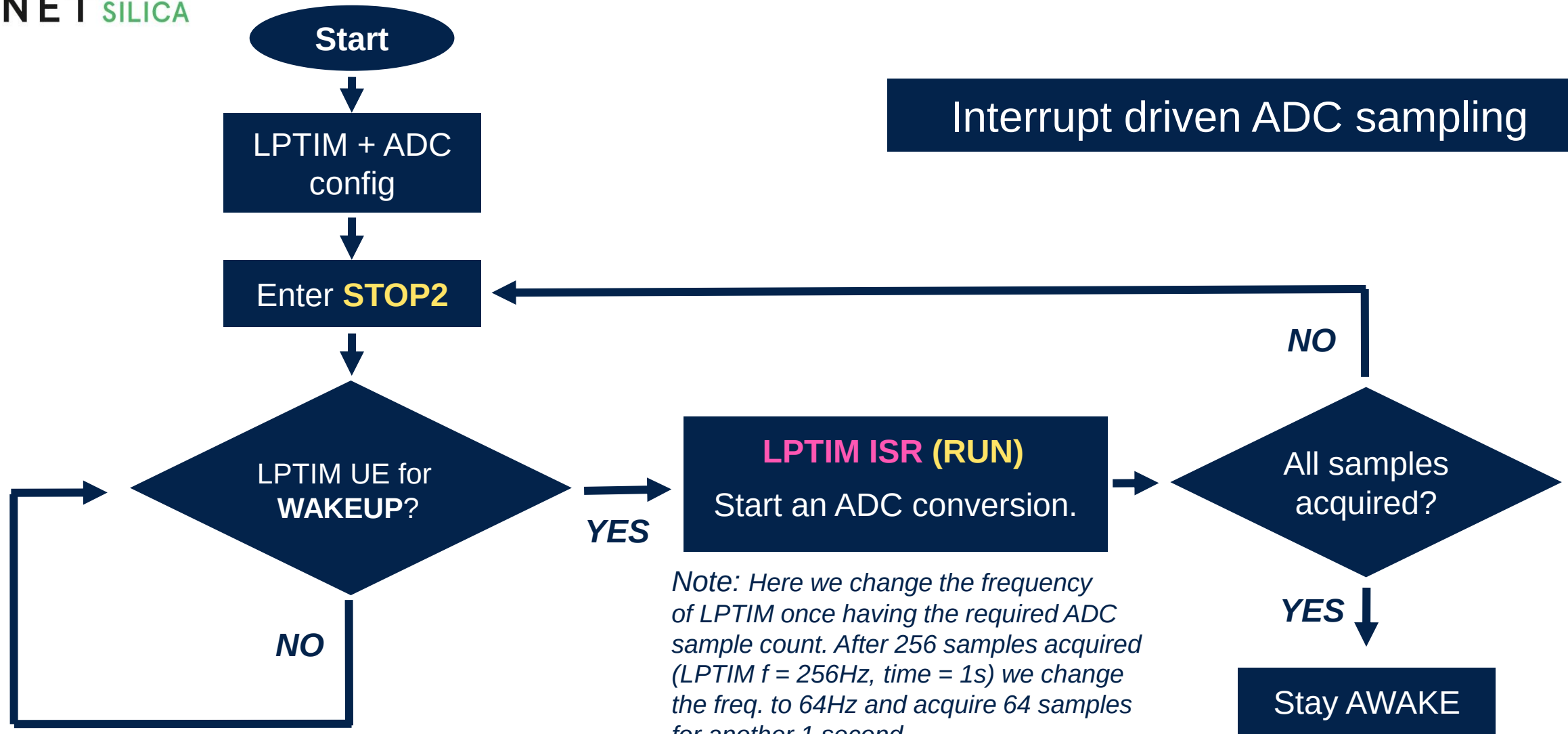
1. Define scenario and Select peripheral instances
2. Configure Queue, DMA channel ISR, NVIC and Attributes
3. Enter Basic peripheral parameters,data configurations, trigger configurations
4. Initialize peripherals and low Power mode in Pinout&Configuration
5. Select clock configuration
6. Check LPBAM Design and Generate Code

STM32CubeMX abstracts linked list complexity with user friendly GUI for LPBAM Scenario Configuration

life.augmented

Benchmark

Interrupt driven ADC sampling

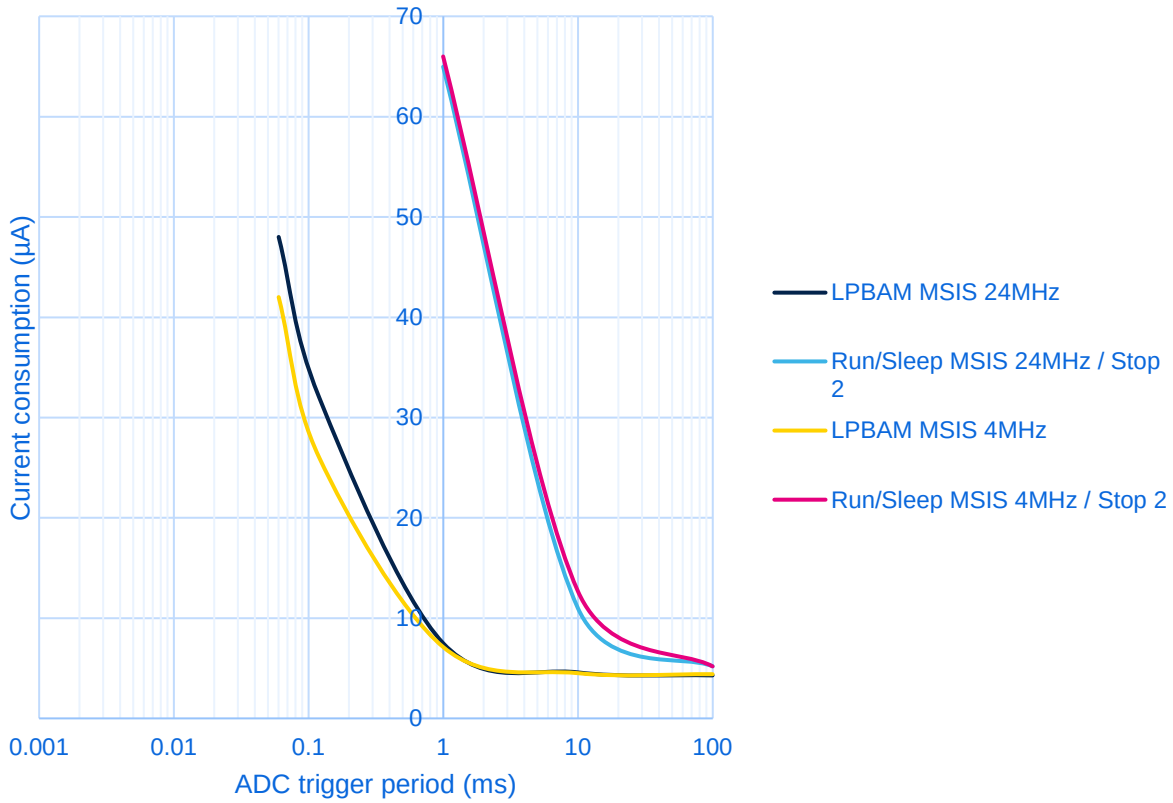


Note: Here we change the frequency of LPTIM once having the required ADC sample count. After 256 samples acquired (LPTIM f = 256Hz, time = 1s) we change the freq. to 64Hz and acquire 64 samples for another 1 second.

Results discussion

Power consumption Estimation

Curves show the benefit of LPBAM in Stop2 vs waking up from Stop2 mode at each update Event ISR



Test Scenario:

- ADC4 clocked by MSIK triggered by LPTIM clocked by LSI
- LPTIM changes it's PWM frequency two times
- ADC4 sampling time is 1.5clock cycles

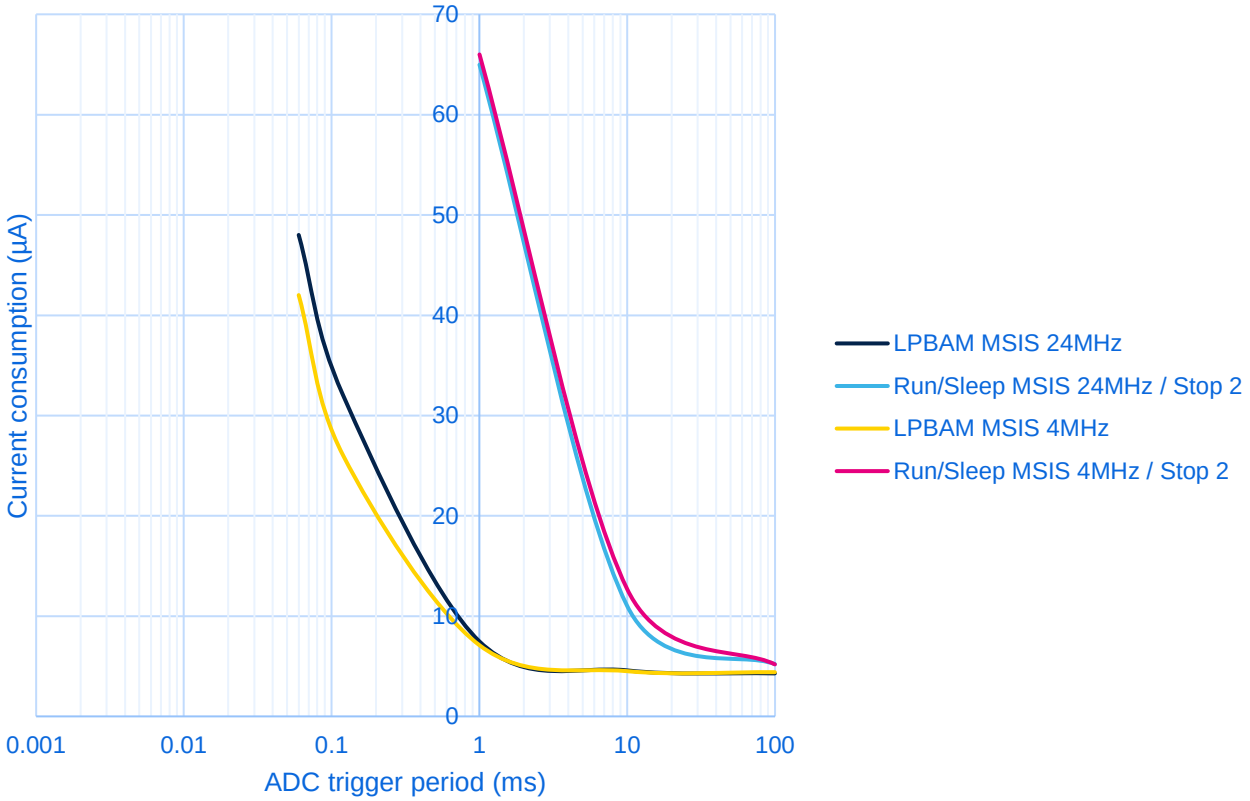
Sampling Freq	Run/Stop @ 4MHz	LPBAM @4Mhz
256Hz/64Hz	~20uA	~4.5uA
2.56KHz/640Hz	~100uA	~10uA

LPBAM@ 4MHz	LPBAM @24Mhz
~4.1uA	~4.3uA

LPBAM Benefit is much larger when increasing the sampling frequency and consumptions are linear with clk freq

Power consumption Estimation

Curves show the benefit of LPBAM in Stop2 vs waking up from Stop2 mode at each update Event ISR



Test Scenario:

- ADC4 clocked by MSIK triggered by LPTIM clocked by LSI
- LPTIM changes it's PWM frequency two times
- ADC4 sampling time is 1.5clock cycles

Sampling Freq	LPBAM Consumption saving vs Run/Stop @ 24MHz
1Khz	90%
100Hz	60%
10Hz	20%

LPBAM Benefit is much larger when increasing the sampling frequency and consumptions are linear with clk freq

Important findings

1. MSIK frequency has a limited impact on average consumption as the phase in which Smart Run Domain is active varies linearly with kernel clock frequency

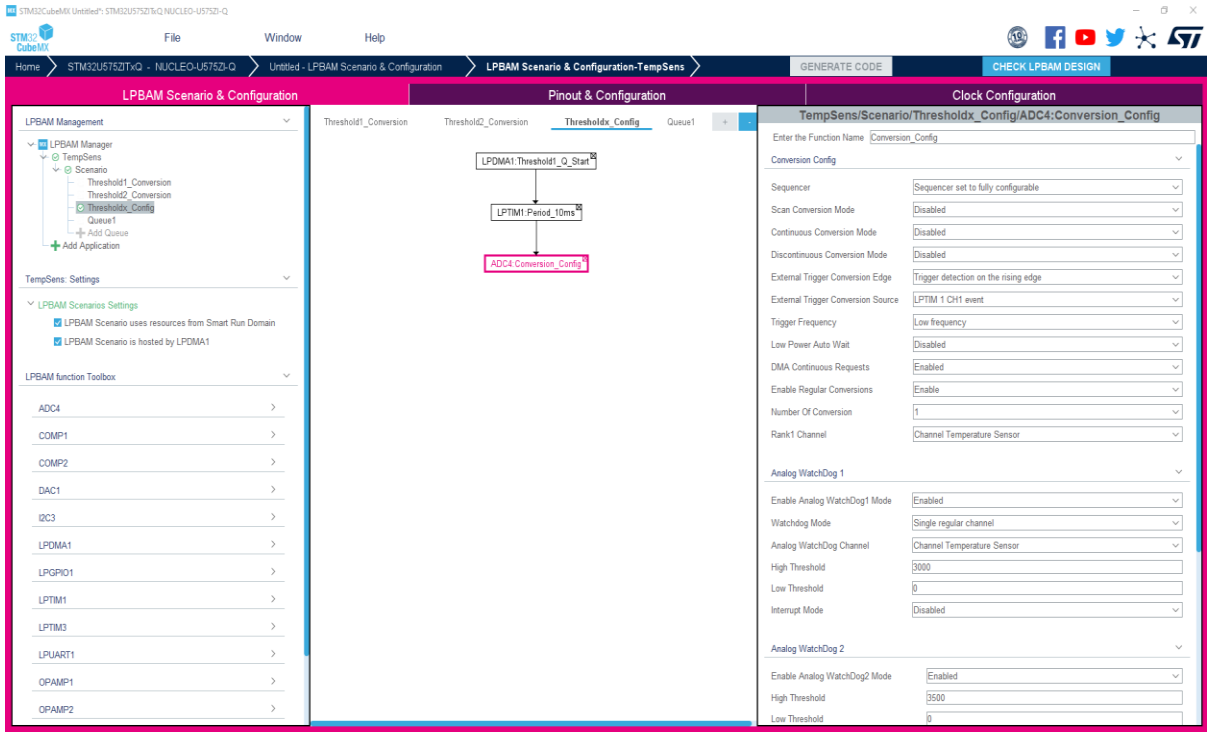
2. LPBAM Benefit is much larger when increasing the sampling frequency

3. LPBAM in Stop 2 proves large consumption benefits which grow significantly in the 50 Hz to 10 kHz sampling range

Wrap Up

LPBAM Tool

- Abstracts linked list complexity
- User friendly interface
- Reduces LPBAM applications development time / effort.
- Provides granularity configuration
- Hardware agnostic firmware.
- Portability inter family/devices.



Thanks to LPBAM Configurator user can easily implement LPBAM scenario with reduced Time to Market

Main Steps handled by STM32CubeMx

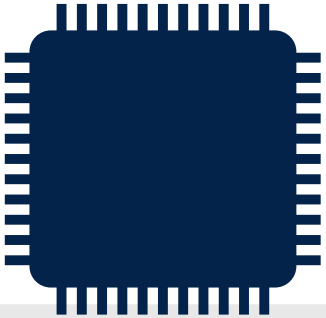
- Configure ICACHE in 1-way mode, enable flash prefetch
- Power down unused flash banks
- Power down unused SRAM
- Switch off bus clock is peripherals are not used
- Select oscillators and analog peripherals in the lowest possible power mode allowed by the application
- Select wakeup from Stop oscillators depending on the application consumption and wakeup time

LPBAM Clock strategies

In LPBAM dynamic consumption is linear with clock frequency and DRUN duration linearly decreases with sys clock frequency therefore there is a clear benefit vs Stop/Run approach which increases significantly with sampling frequency.

LSI and LSE are recommended as kernel clocks to get lowest possible power consumption

Low power background autonomous mode (LPBAM)



No need for CPU – all is based on DMA (LPDMA & GPDMA)

Peripheral's configuration & activity can be chained thanks to **DMA linked-list**, down to Stop 2 mode.

Power gain:

Peripherals can operate in STOP2 mode without any need to wake up

Bus and Kernel clocks are **provided to IP only when necessary** in Stop mode

Analog peripherals/oscillators are powered-on only when needed

Reference Links

life.augmented

Reference Links

- [1] Reference manual STM32U575xx and STM32U585xx advanced Arm-based 32-bit MCUs RM0456
- [2] Datasheets for STM32U575xx (DS13737) and STM32U585xx DS13086
- [3] Application note STM32 microcontroller GPIO configuration for hardware settings and low-power consumption AN4899
- [4] Application note STM32U575/585 power optimization AN5652
- [5] Application note STM32U5 Series power optimization using LPBAM AN5645
- [6] On-demand webinar New STM32CubeMX feature for Low-Power Background Autonomous Mode configuration with STM32U5 Link

life.augmented

LPBAM Glossary

- **LPBAM = Low Power Background Autonomous Mode**
- **LPBAM Node** = Elementary entity of DMA transfer. Node configuration is stored in SRAM
- **LPBAM Queue** = A set of DMA transfer nodes linked to each other.
- **LPBAM Linked List**= DMA modality of usage which allows the DMA peripheral to chain a list of DMA transfers.
- **LPBAM Function** = A set of DMA nodes chained together to perform a specific task

Our technology starts with You

life.augmented