

# How to drive a 12 bits DAC MCP4822 With a Nucleo-L476RG Part 1/2

KEIL programming environment for NUCLEO-STM32L476RG board

Jean-Christophe Toussaint  
Phelma Grenoble-INP

# Nucleo-L4 : CubeMx Configurator

Run CubeMX

New Project

MCU Selector Board Selector

Board Filter

Vendor : STMicroelectr... Type of Board : Nucleo64 MCU Series : STM32L4

Initialize all peripherals with their default Mode

Peripheral Selection

Peripherals	Nb	Max
Accelerometer	<input type="checkbox"/>	
Analog I/O	0	0
Arduino Form Factor	0	0
Audio Line In	0	0
Audio Line Out	0	0
Button	0	1
CAN	0	0
Camera	<input type="checkbox"/>	
Compass	<input type="checkbox"/>	

Boards List: 4 Items

Type	Reference	MCU
Nucleo64	NUCLEO-L476RG	STM32L476RGTx
Nucleo64	NUCLEO-L452RE	STM32L452RETx
Nucleo64	NUCLEO-L433RC-P	STM32L433RCTXP
Nucleo64	NUCLEO-L452RE-P	STM32L452RETxP

In the Board Selector  
Select STM32L4 MCU series

MCU used for this project:  
NUCLEO-STM32L476RG

New Project

Load Project

Help



Project Code Generator Advanced Settings

## Project Settings

Project Name

nucleo\_L4\_SPI\_DAC\_MCP4822\_v2

Project Location

C:\Documents and Settings\Administrateur\Mes documents

Toolchain Folder Location

C:\Documents and Settings\Administrateur\Mes documents\nucleo\_L4\_SPI\_DAC\_MCP4822\_v2

Toolchain / IDE

MDK-ARM V5



KEIL Environment

## Linker Settings

Minimum Heap Size

0x200

Minimum Stack Size

0x400

## Mcu and Firmware Package

Mcu Reference

STM32L476RGTx

Firmware Package Name and Version

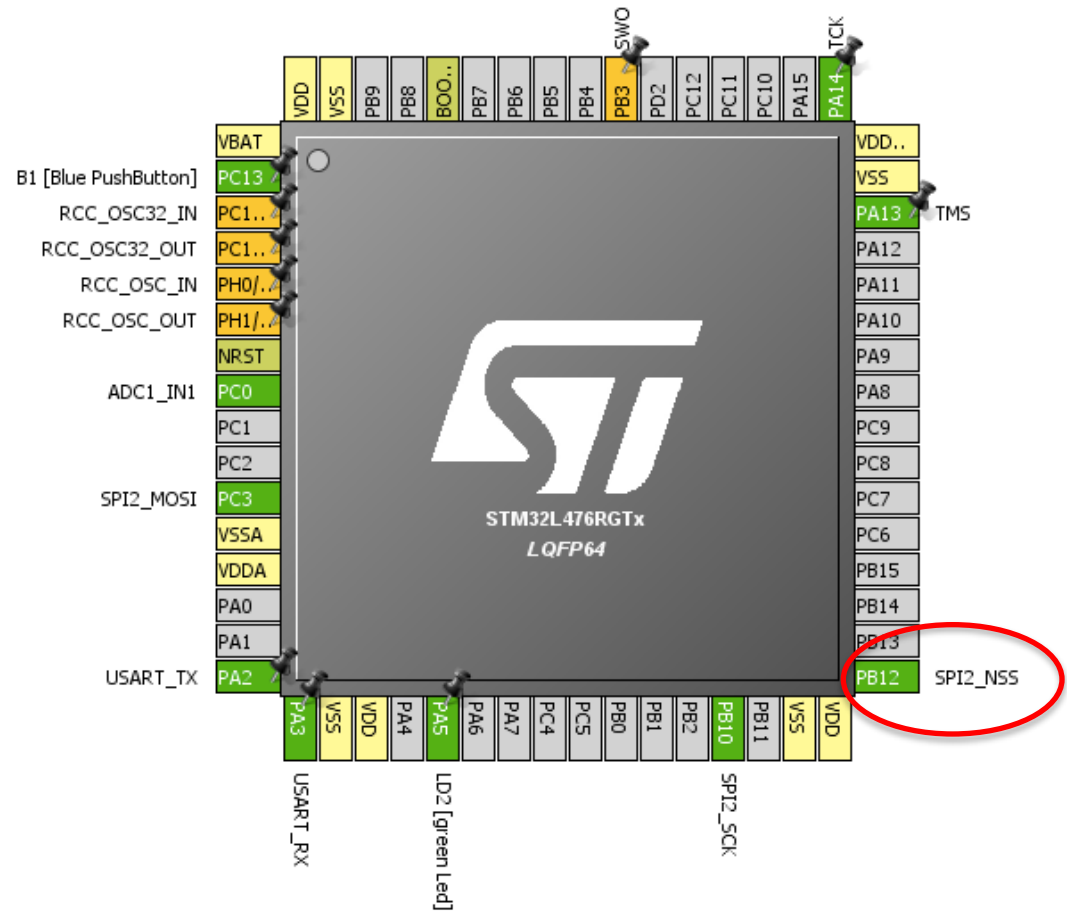
STM32Cube FW\_L4 V1.5.1



Ok

Cancel

- SAI2
- SDMMC1
- SPI1
- SPI2
  - Mode: Transmit Only Master
  - Hardware NSS Signal: Hardware NSS Output ...
- SPI3
- SWPMI1
- SYS
- TIM1
- TIM2
- TIM3
- TIM4
- TIM5
- TIM6
- TIM7
- TIM8
- TIM15
- TIM16
- TIM17
- TSC
- UART4
- UART5
- USART1
- USART2
- USART3
- USB\_OTG\_FS
- WWDG



## SPI2 Configuration

Parameter Settings ✓ User Constants ✓ NVIC Settings ✓ GPIO Settings ✓ DMA Settings ✓

Configure the below parameters :

Search :

## Basic Parameters

Frame Format	Motorola
Data Size	16 Bits
First Bit	MSB First

## Clock Parameters

Prescaler (for Baud Rate)	2
Baud Rate	40.0 Mbits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

## Advanced Parameters

CRC Calculation	Disabled
NSSP Mode	Enabled
NSS Signal Type	Output Hardware

Apply

Ok

Cancel

# Code Lines to be inserted into main.c

```
/* 10 bits Sine Sampling*/
uint16_t sine[]={
    512, 525, 537, 550, 562, 575, 587, 599, 612, 624, 636, 648, 660, 672, 684, 696,
    708, 719, 730, 742, 753, 764, 775, 785, 796, 806, 816, 826, 836, 846, 855, 864,
    873, 882, 891, 899, 907, 915, 922, 930, 937, 944, 950, 957, 963, 968, 974, 979,
    984, 989, 993, 997,1001,1004,1008,1011,1013,1015,1017,1019,1021,1022,1022,1023,
    1023,1023,1022,1022,1021,1019,1017,1015,1013,1011,1008,1004,1001, 997, 993, 989,
    984, 979, 974, 968, 963, 957, 950, 944, 937, 930, 922, 915, 907, 899, 891, 882,
    873, 864, 855, 846, 836, 826, 816, 806, 796, 785, 775, 764, 753, 742, 730, 719,
    708, 696, 684, 672, 660, 648, 636, 624, 612, 599, 587, 575, 562, 550, 537, 525,
    512, 499, 487, 474, 462, 449, 437, 425, 412, 400, 388, 376, 364, 352, 340, 328,
    316, 305, 294, 282, 271, 260, 249, 239, 228, 218, 208, 198, 188, 178, 169, 160,
    151, 142, 133, 125, 117, 109, 102, 94, 87, 80, 74, 67, 61, 56, 50, 45,
    40, 35, 31, 27, 23, 20, 16, 13, 11, 9, 7, 5, 3, 2, 2, 1,
    1, 1, 2, 2, 3, 5, 7, 9, 11, 13, 16, 20, 23, 27, 31, 35,
    40, 45, 50, 56, 61, 67, 74, 80, 87, 94, 102, 109, 117, 125, 133, 142,
    151, 160, 169, 178, 188, 198, 208, 218, 228, 239, 249, 260, 271, 282, 294, 305,
    316, 328, 340, 352, 364, 376, 388, 400, 412, 425, 437, 449, 462, 474, 487, 499};

const uint16_t NS=256;
uint16_t aTxBuffer[NS];

/* USER CODE END PV */
```

```

/* USER CODE BEGIN 2 */
for (uint16_t ns=0; ns<NS; ns++){
    uint16_t data=sine[ns]<<2;    // x4
                                // 10 bit sine x 4 --> 12 bit sine
    data |= 0x3000;
    aTxBuffer[ns]= data;
}

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{ // typecasting from (uint16_t*) to (uint8_t*)
  HAL_SPI_Transmit(&hspi2, (uint8_t*) aTxBuffer, NS, 10);
/* USER CODE END WHILE */

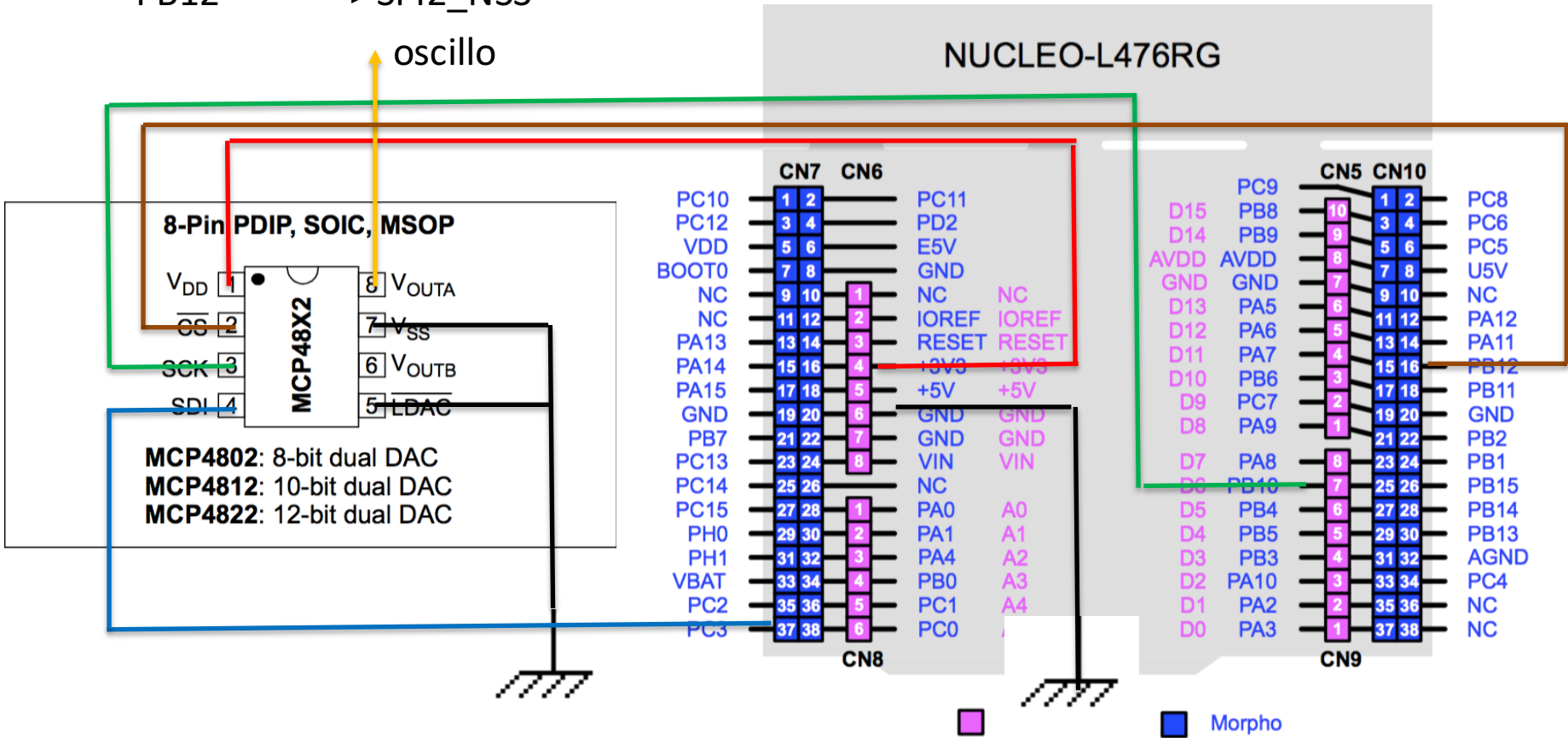
/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

```

# Wiring

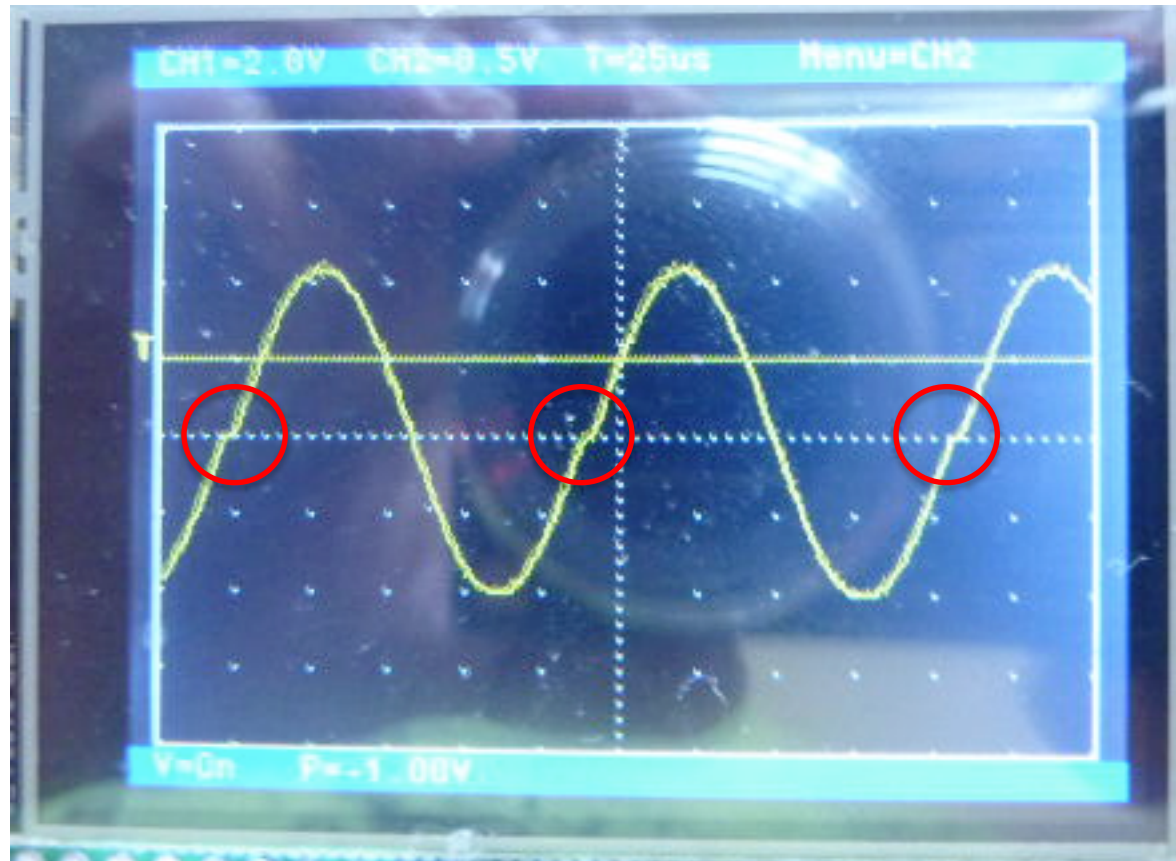
PC3 -----> SPI2\_MOSI  
 PB10 -----> SPI2\_SCK  
 PB12 -----> SPI2\_NSS

Open stm32l4xx\_hal\_msp.cc file and examine  
 void HAL\_SPI\_MspDeInit(SPI\_HandleTypeDef\* hspi)





- Compile and upload binary to MCU
- Reset MCU if necessary by pushing black button



Signal kinks observed at every call to `HAL_SPI_Transmit` !!!

# How to drive a 12 bits DAC MCP4822 Using DMA Part 2/2

KEIL programming environment for NUCLEO-STM32L476RG board

Jean-Christophe Toussaint  
Phelma Grenoble-INP

**SPI2 Configuration** ✕

✔ Parameter Settings
✔ User Constants
✔ NVIC Settings
✔ GPIO Settings
✔ DMA Settings

Configure the below parameters :

Search :  ↕ ↕ ☰

[-] Basic Parameters

Frame Format	Motorola
Data Size	16 Bits
First Bit	MSB First

[-] Clock Parameters

Prescaler (for Baud Rate)	2
Baud Rate	40.0 Mbits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

[-] Advanced Parameters

CRC Calculation	Disabled
NSSP Mode	Enabled
NSS Signal Type	Output Hardware

# SPI2 Configuration

- Parameter Settings
- User Constants
- NVIC Settings
- GPIO Settings
- DMA Settings

DMA Request	Channel	Direction	Priority
SPI2 TX	DMA1 Channel 5	Memory To Peripheral	Medium

Add Delete

### DMA Request Settings

Mode **Circular**

Increment Address

Data Width

Peripheral	Memory
<input type="checkbox"/>	<input checked="" type="checkbox"/>
Half Word	Half Word

Apply Ok Cancel

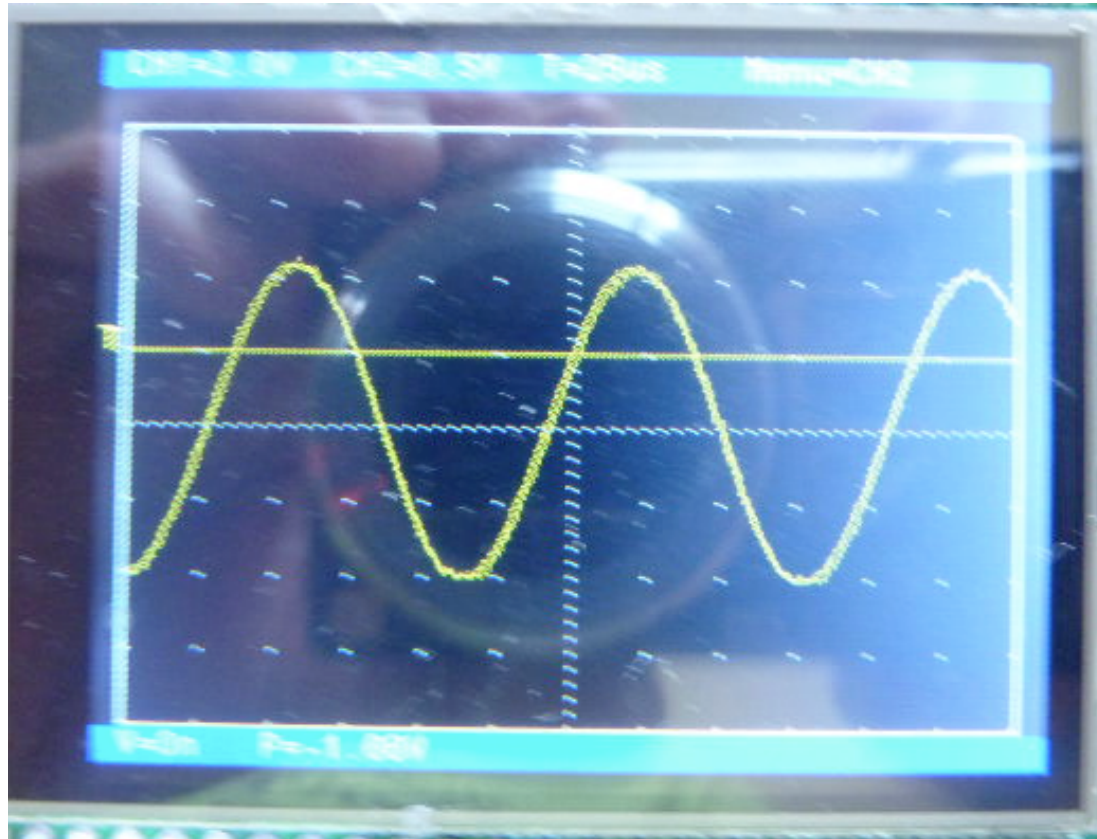
```

/* USER CODE BEGIN 2 */
for (uint16_t ns=0; ns<NS; ns++){
    uint16_t data=sine[ns]<<2;    // x4
                                   // 10 bit sine x 4 --> 12 bit sine
    data |= 0x3000;
    aTxBuffer[ns]= data;
}
// typecasting from (uint16_t*) to (uint8_t*)
HAL_SPI_Transmit_DMA(&hspi2, (uint8_t*) aTxBuffer, NS);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1){
    printf("Enter sleep mode\n");
    HAL_SuspendTick();
    HAL_PWR_EnterSLEEPMode(PWR_LOWPOWERREGULATOR_ON,PWR_SLEEPENTRY_WFI);
    printf("Enter run mode\n");
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

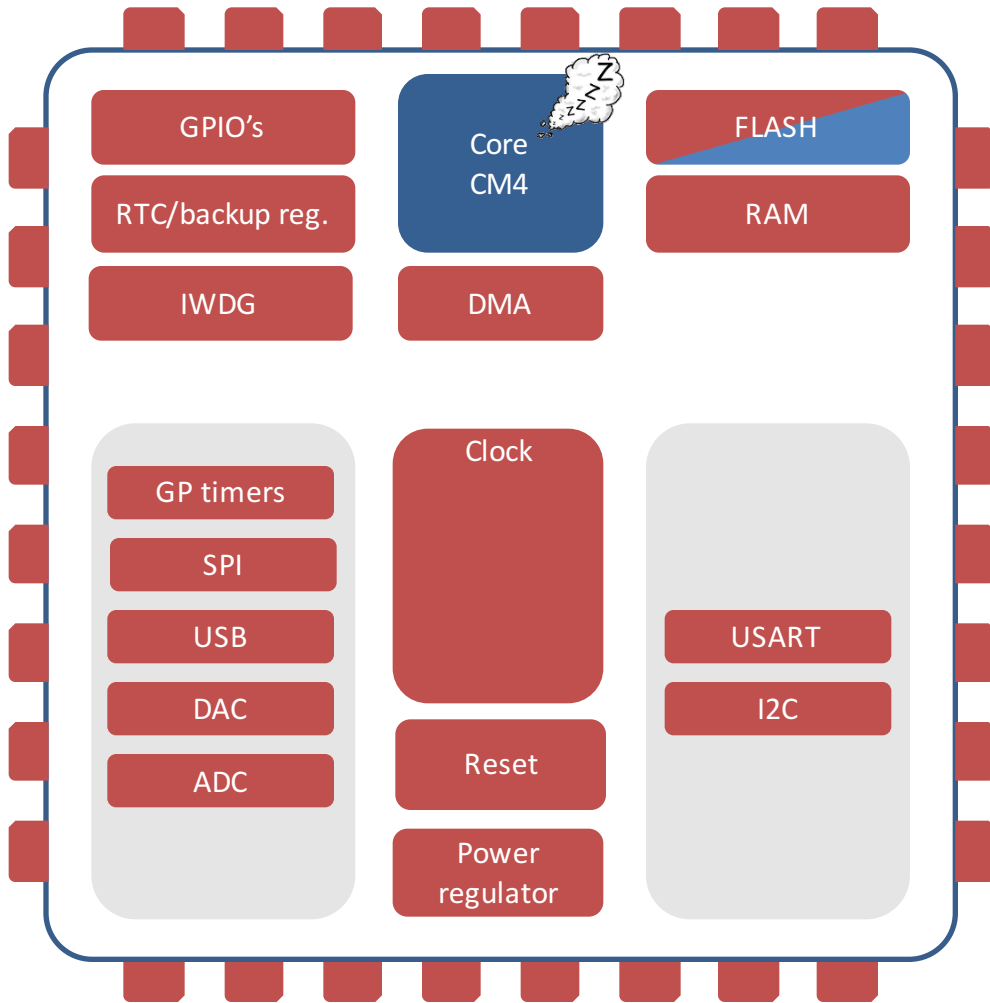
```

# Oscilloscope Output



No Signal kinks observed using DMA

# Nucleo-STM32L476 in Low Consumption : SLEEP Mode



- **Core is stopped**
- **Peripherals are running**

When exiting the Low-power sleep mode by issuing an interrupt or a wakeup event, the MCU is in Low-power Run mode.

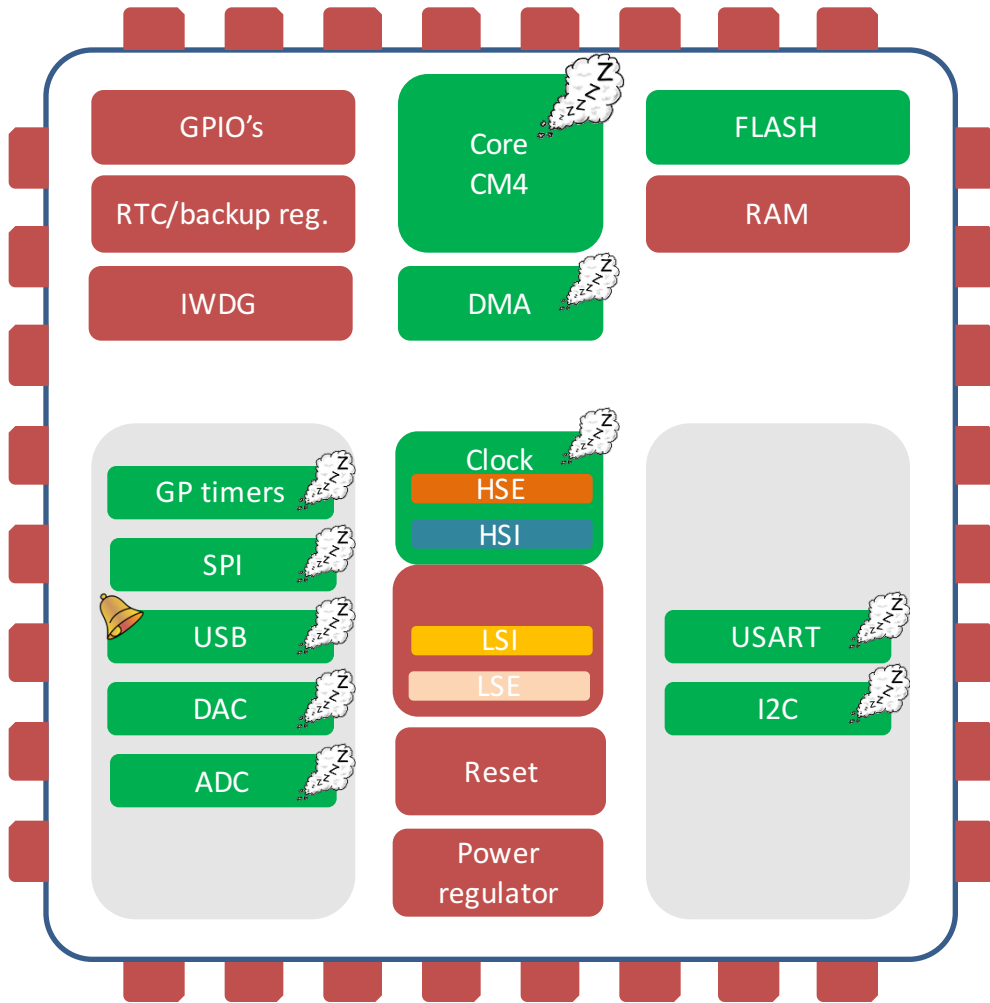
```
/* USER CODE BEGIN */
```

```
while (1) {  
    printf("Enter sleep mode\n");  
    HAL_SuspendTick();  
    HAL_PWR_EnterSLEEPMode(PWR_LOWPOWERREGULATOR_ON,PWR_SLEEPENTRY_WFI);  
    HAL_ResumeTick();  
    printf("Enter run mode\n");  
}
```

```
/* USER CODE END WHILE */
```



# Nucleo-STM32L476 in low Consumption : STOP mode



- **Core is stopped**
- **HSE, MSI clocks are OFF**
- **SRAM and registers content is preserved**
- **Peripherals with HSI, LSI, LSE clock option can be ON**
- **GPIO's keep their setup**

```
/* USER CODE BEGIN WHILE */
```

```
while (1) {
```

```
printf("Enter stop mode\n"); // not working USART disabled
```

```
HAL_PWR_EnterSTOPMode(PWR_LOWPOWERREGULATOR_ON,PWR_STOPENTRY_WFI);  
SystemClock_Config();
```

```
printf("Enter run mode\n"); // not working USART disabled  
}
```

```
/* USER CODE END WHILE */
```