Tutorial Fourth Session

Lecturer :
Shahab Mahmoudi Sadaghiani

Email : shahab.mahmoudisadaghiani@mail.mcgill.ca

ECSE 444 - Microprocessor -Winter 2023

McGill

Agenda:

1. Timer
2. DAC
3. External Interrupts
4. DMA (Direct Memory Access)

McGill

Agenda:

1. Timer
2. DAC
3. External Interrupts
4. DMA (Direct Memory Access)

McGill

Timers

- Advanced-control timers (TIM1/TIM8)
- General-purpose timers (TIM2/TIM3/TIM4/TIM5)
- General-purpose timers (TIM15/TIM16/TIM17)
- Basic timers (TIM6/TIM7)
- Low-power timer (LPTIM)

38.2 TIM2/TIM3/TIM4/TIM5 main features

General-purpose TIMx timer features include:

- 16-bit (TIM3, TIM4) or 32-bit (TIM2 and TIM5) up, down, up/down auto-reload counter.
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535.
- Up to 4 independent channels for:
  - Input capture
  - Output compare
  - PWM generation (Edge- and Center-aligned modes)
  - One-pulse mode output
- Synchronization circuit to control the timer with external signals and to interconnect several timers.
- Interrupt/DMA generation on the following events:
  - Update: counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
  - Input capture
  - Output compare
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes
- Trigger input for external clock or cycle-by-cycle current management

39.2 TIM15 main features

TIM15 includes the following features:

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535
- Up to 2 independent channels for:
  - Input capture
  - Output compare
  - PWM generation (edge mode)
  - One-pulse mode output
- Complementary outputs with programmable dead-time (for channel 1 only)
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Repetition counter to update the timer registers only after a given number of cycles of the counter
- Break input to put the timer’s output signals in the reset state or a known state
- Interrupt/DMA generation on the following events:
  - Update: counter overflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
  - Input capture
  - Output compare
  - Break input (interrupt request)

39.3 TIM16/TIM17 main features

The TIM16/TIM17 timers include the following features:

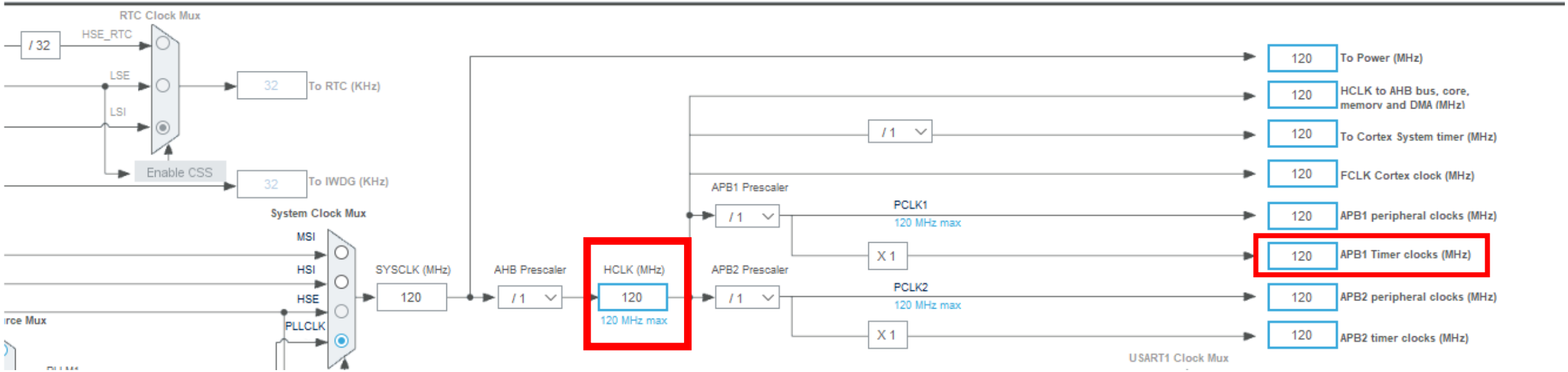
- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535
- One channel for:
  - Input capture
  - Output compare
  - PWM generation (edge-aligned mode)
  - One-pulse mode output
- Complementary outputs with programmable dead-time
- Repetition counter to update the timer registers only after a given number of cycles of the counter
- Break input to put the timer’s output signals in the reset state or a known state
- Interrupt/DMA generation on the following events:
  - Update: counter overflow
  - Input capture
  - Output compare
  - Break input

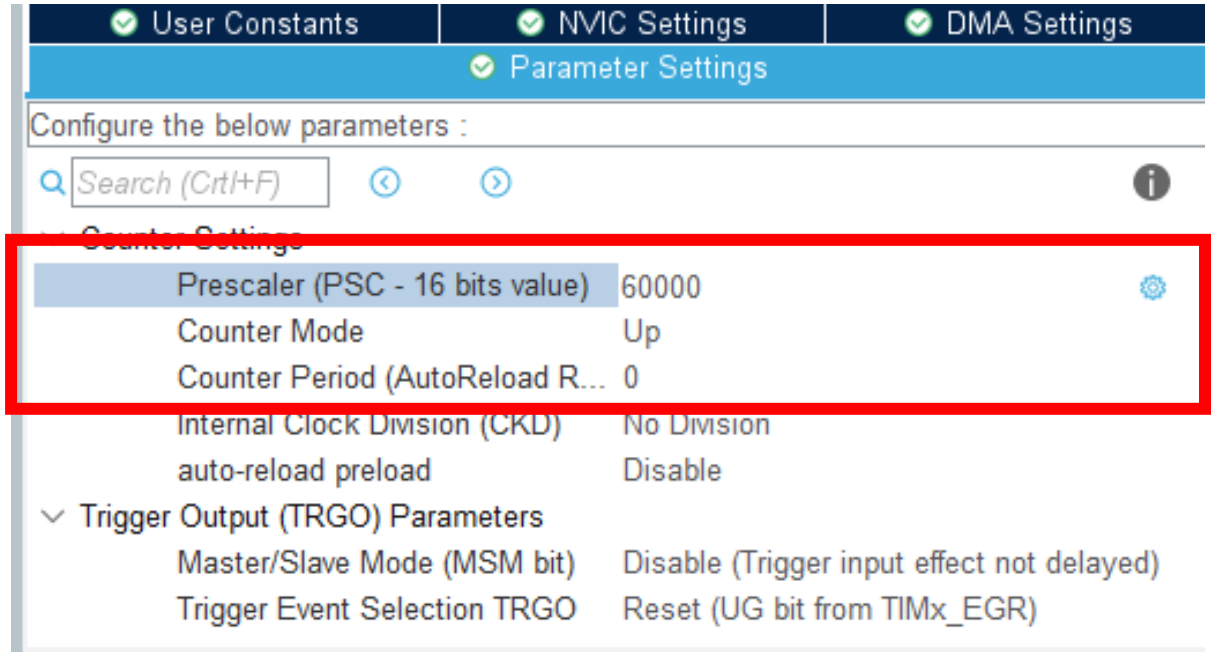
40.2 TIM6/TIM7 main features

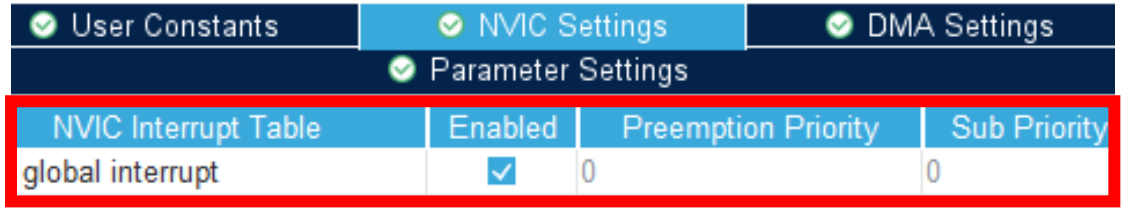
Basic timer (TIM6/TIM7) features include:

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535
- Synchronization circuit to trigger the DAC
- Interrupt/DMA generation on the update event: counter overflow

Clock Configuration




User Constants
NVIC Settings
DMA Settings
Parameter Settings
Configure the below parameters :
Search (Crtl+F)
Prescaler (PSC - 16 bits value) 60000
Counter Mode Up
Counter Period (AutoReload R... 0
Internal Clock Division (CKD) No Division
auto-reload preload Disable
Trigger Output (TRGO) Parameters
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)
Trigger Event Selection TRGO Reset (UG bit from TIMx_EGR)


User Constants
NVIC Settings
DMA Settings
Parameter Settings
NVIC Interrupt Table
Enabled
Preemption Priority
Sub Priority
global interrupt
0
0

HAL Driver

65.2.3 Time Base functions

This section provides functions allowing to:

- Initialize and configure the TIM base.
- De-initialize the TIM base.
- Start the Time Base.
- Stop the Time Base.
- Start the Time Base and enable interrupt.
- Stop the Time Base and disable interrupt.
- Start the Time Base and enable DMA transfer.
- Stop the Time Base and disable DMA transfer.

This section contains the following APIs:

- ***HAL_TIM_Base_Init()***
- ***HAL_TIM_Base_DeInit()***
- ***HAL_TIM_Base_MspInit()***
- ***HAL_TIM_Base_MspDeInit()***
- ***HAL_TIM_Base_Start()***
- ***HAL_TIM_Base_Stop()***
- ***HAL_TIM_Base_Start_IT()***
- ***HAL_TIM_Base_Stop_IT()***
- ***HAL_TIM_Base_Start_DMA()***
- ***HAL_TIM_Base_Stop_DMA()***

Interrupt Callback function:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
if (htim->Instance == TIM17) {
    HAL_IncTick();
  }

}
```

Agenda:

1. Timer
2. DAC
3. External Interrupts
4. DMA (Direct Memory Access)

McGill

22.2 DAC main features

The DAC main features are the following (see *Figure 157: Dual-channel DAC block diagram*)

- One DAC interface, maximum two output channels
- Left or right data alignment in 12-bit mode
- Synchronized update capability
- Noise-wave and Triangular-wave generation
- Dual DAC channel for independent or simultaneous conversions
- DMA capability for each channel including DMA underrun error detection
- External triggers for conversion
- DAC output channel buffered/unbuffered modes
- Buffer offset calibration
- Each DAC output can be disconnected from the DACx_OUTy output pin
- DAC output connection to on chip peripherals
- Sample and hold mode for low power operation in Stop mode
- Input voltage reference, V_{REF+}

Figure 157 shows the block diagram of a DAC channel and *Table 146* gives the pin description.

22.4.3 DAC data format

Depending on the selected configuration mode, the data have to be written into the specified register as described below:

- Single DAC channel

  There are three possibilities:

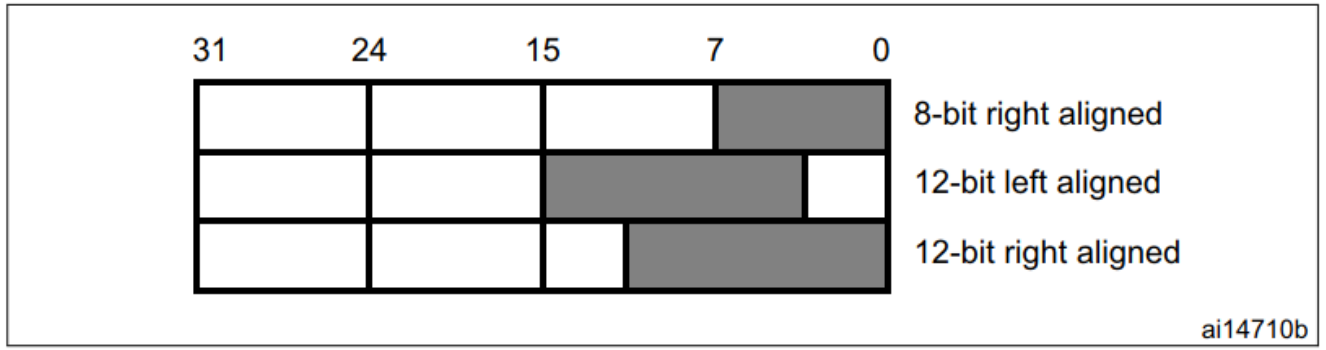
  – 8-bit right alignment: the software has to load data into the DAC_DHR8Rx[7:0] bits (stored into the DHRx[11:4] bits)

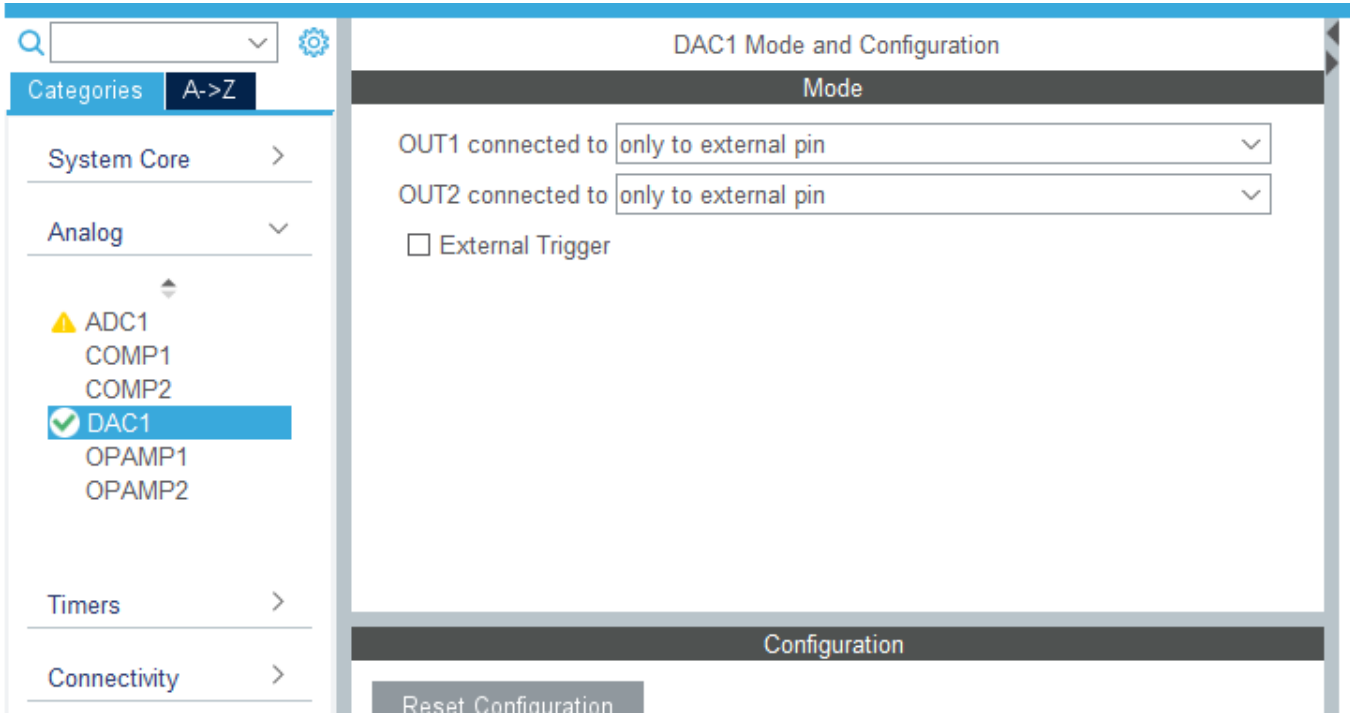
  – 12-bit left alignment: the software has to load data into the DAC_DHR12Lx [15:4] bits (stored into the DHRx[11:0] bits)

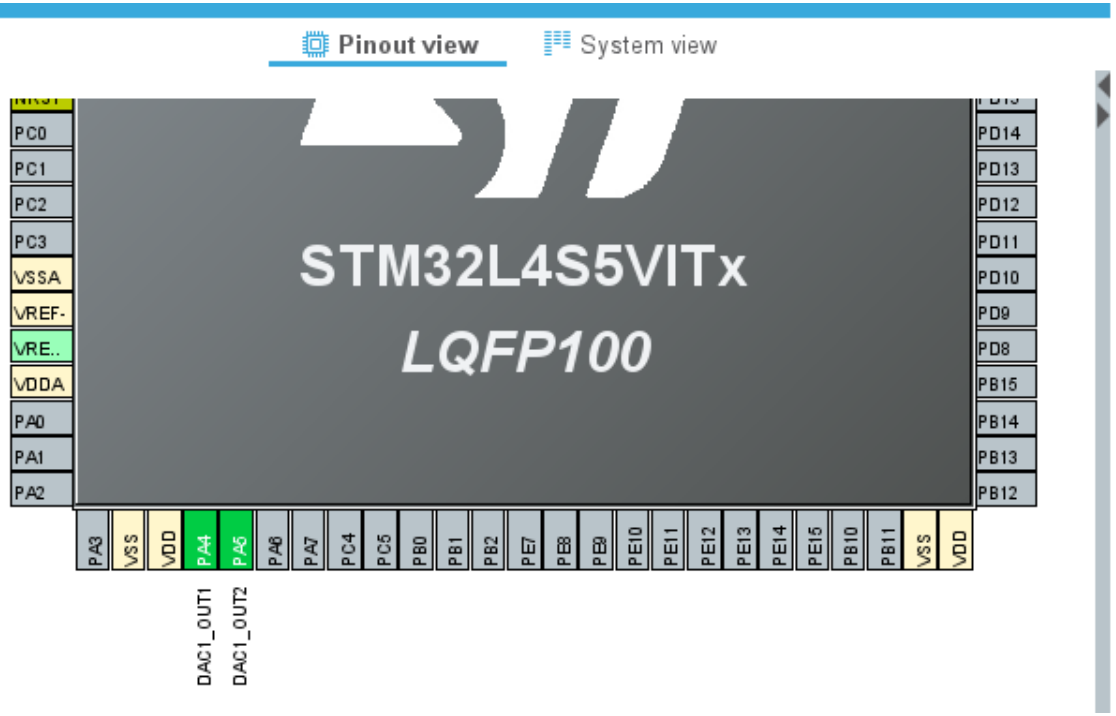
  – 12-bit right alignment: the software has to load data into the DAC_DHR12Rx [11:0] bits (stored into the DHRx[11:0] bits)

Depending on the loaded DAC_DHRyyyx register, the data written by the user is shifted and stored into the corresponding DHRx (data holding registerx, which are internal non-memory-mapped registers). The DHRx register is then loaded into the DORx register either automatically, by software trigger or by an external event trigger.

Figure 158. Data registers in single DAC channel mode







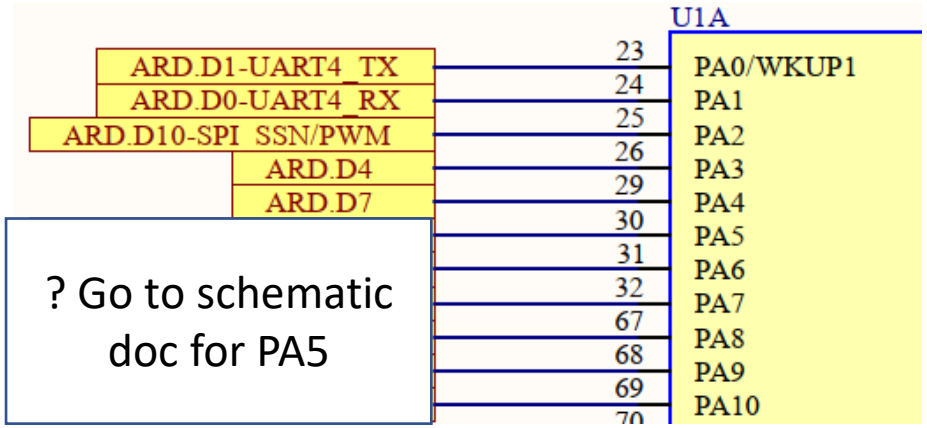


Table 145. DAC features

DAC features	DAC1
Dual channel	X
Output buffer	X
I/O connection	DAC1_OUT1 on PA4, DAC1_OUT2 on PA5
Maximum sampling time	1MSPS
Autonomous mode	-

14.2.4 IO operation functions

This section provides functions allowing to:

- Start conversion.
- Stop conversion.
- Start conversion and enable DMA transfer.
- Stop conversion and disable DMA transfer.
- Get result of conversion.

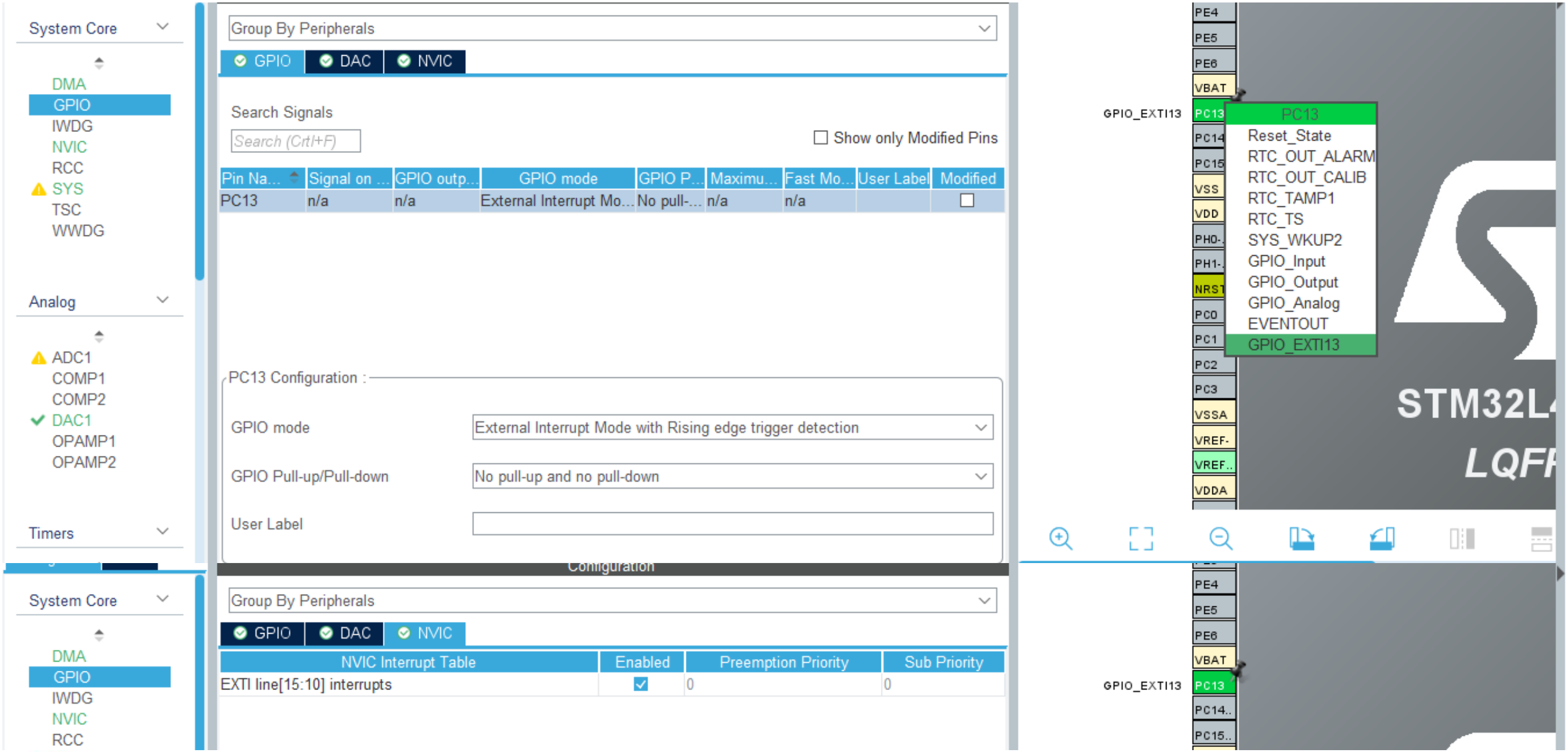
This section contains the following APIs:

- ***HAL_DAC_Start()***
- ***HAL_DAC_Stop()***
- ***HAL_DAC_Start_DMA()***
- ***HAL_DAC_Stop_DMA()***
- ***HAL_DAC_IRQHandler()***
- ***HAL_DAC_SetValue()***
- ***HAL_DAC_ConvCpltCallbackCh1()***
- ***HAL_DAC_ConvHalfCpltCallbackCh1()***
- ***HAL_DAC_ErrorCallbackCh1()***
- ***HAL_DAC_DMAUnderrunCallbackCh1()***

Agenda:

1. Timer
2. DAC
3. External Interrupts
4. DMA (Direct Memory Access)

McGill


System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
TSC
WWDG
Analog
ADC1
COMP1
COMP2
DAC1
OPAMP1
OPAMP2
Timers
Group By Peripherals
GPIO
DAC
NVIC
Search Signals
Search (Crtl+F)
Show only Modified Pins
Pin Na...
Signal on ...
GPIO outp...
GPIO mode
GPIO P...
Maximu...
Fast Mo...
User Label
Modified
PC13
n/a
n/a
External Interrupt Mo...
No pull-...
n/a
n/a
PC13 Configuration :
GPIO mode
External Interrupt Mode with Rising edge trigger detection
GPIO Pull-up/Pull-down
No pull-up and no pull-down
User Label
Configuration
NVIC Interrupt Table
Enabled
Preemption Priority
Sub Priority
EXTI line[15:10] interrupts
0
0
GPIO_EXTI13
PC13
Reset_State
RTC_OUT_ALARM
RTC_OUT_CALIB
RTC_TAMP1
RTC_TS
SYS_WKUP2
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI13
PE4
PE5
PE6
VBAT
PC14
PC15
VSS
VDD
NRST
PC0
PC1
PC2
PC3
VSSA
VREF-
VDDA
STM32L
LQF

Interrupt Callback function:

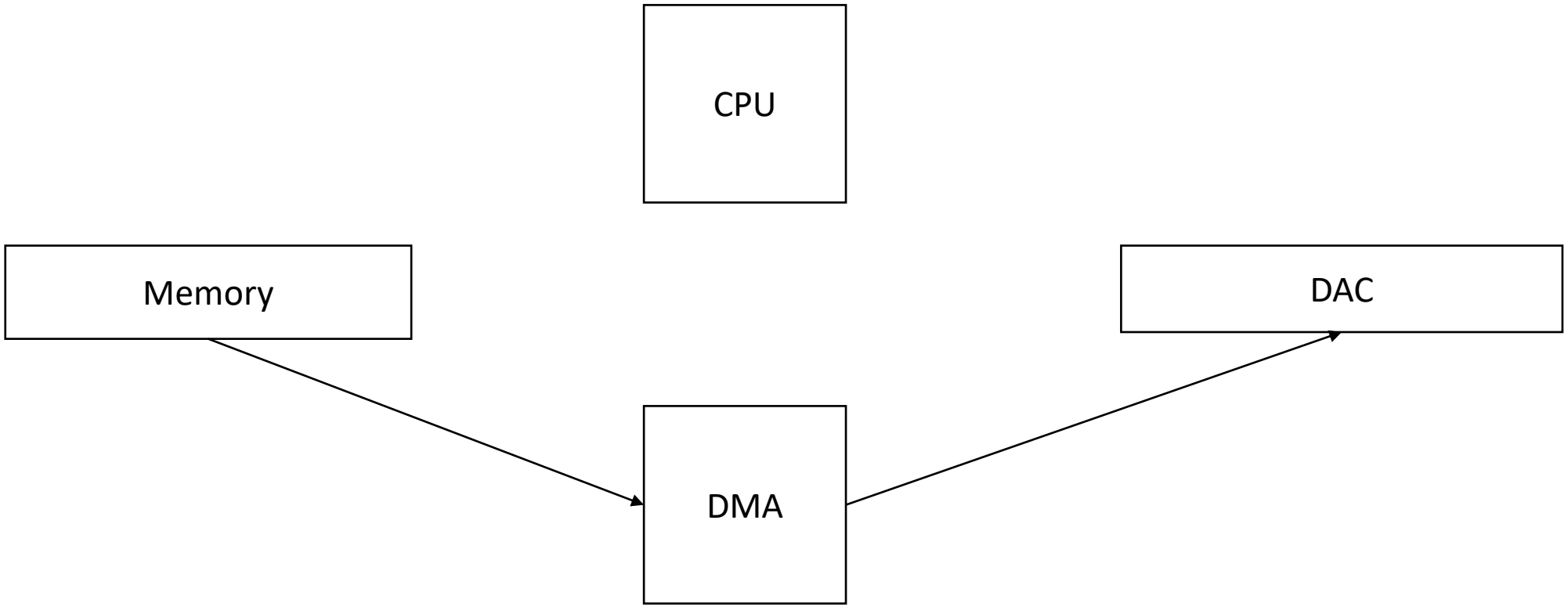
```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){



}
```

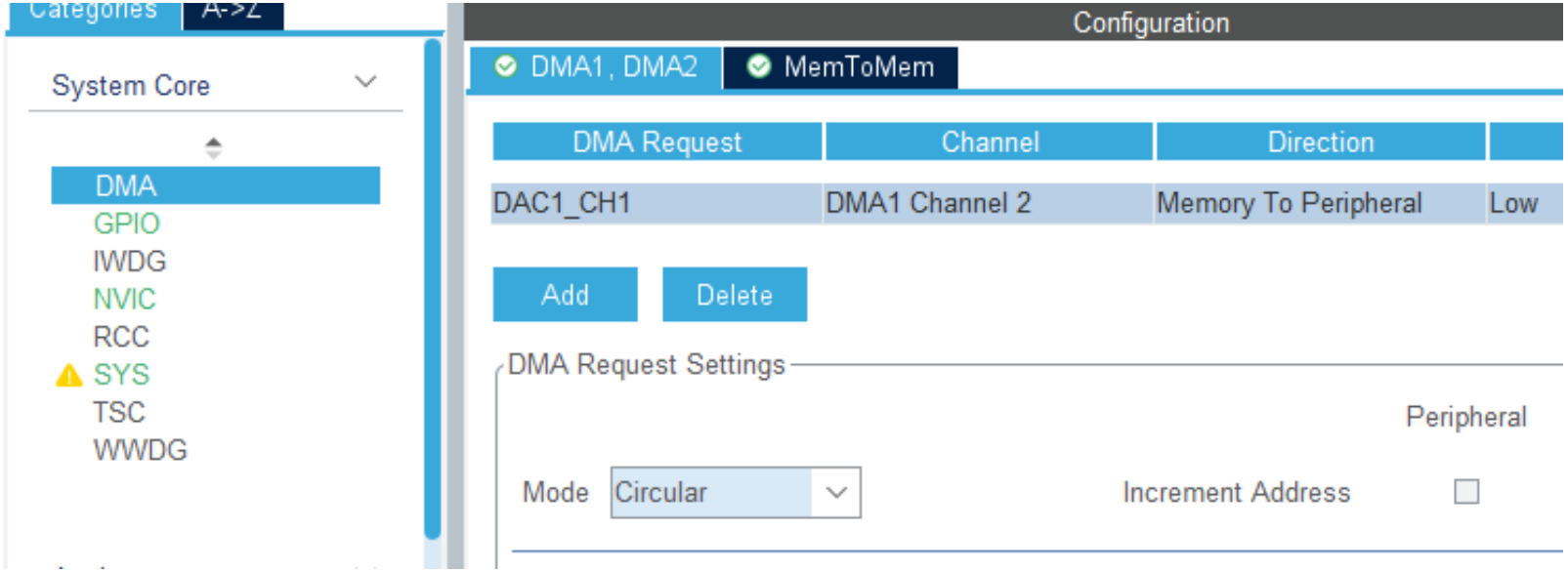
Agenda:

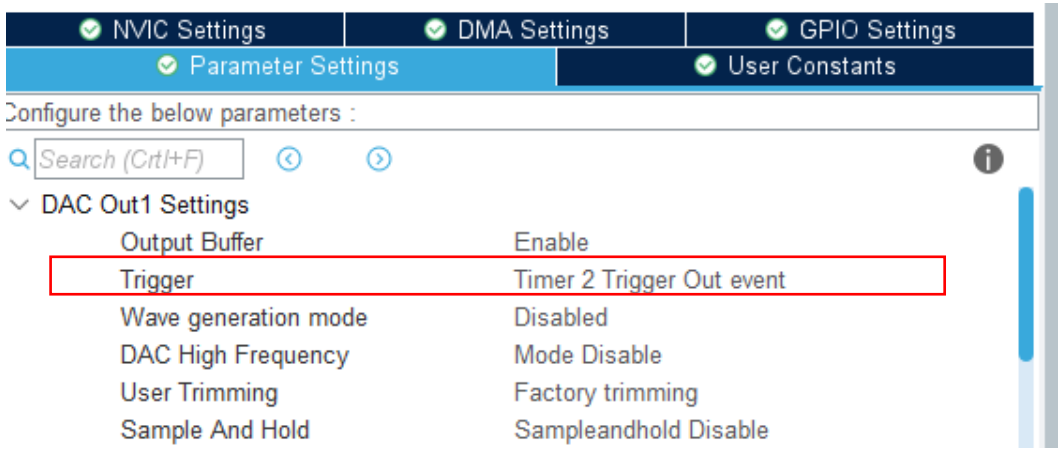
1. Timer
2. DAC
3. External Interrupts
4. DMA (Direct Memory Access)

McGill

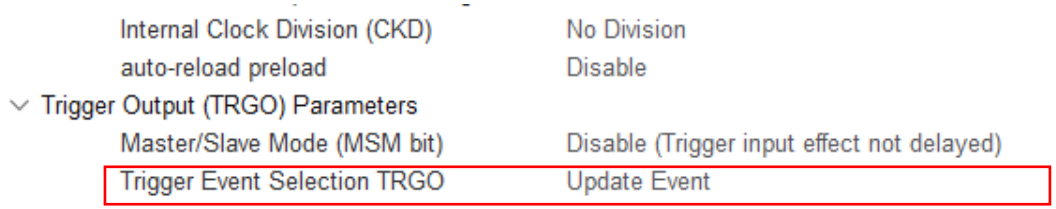
Direct Memory Access (DMA):







Timer settings



20.2.3 IO operation functions

This section provides functions allowing to:

- Configure the source, destination address and data length and Start DMA transfer
- Configure the source, destination address and data length and Start DMA transfer with interrupt
- Abort DMA transfer
- Poll for transfer complete
- Handle DMA interrupt request

This section contains the following APIs:

- ***HAL_DMA_Start()***
- ***HAL_DMA_Start_IT()***
- ***HAL_DMA_Abort()***
- ***HAL_DMA_Abort_IT()***
- ***HAL_DMA_PollForTransfer()***
- ***HAL_DMA_IRQHandler()***
- ***HAL_DMA_RegisterCallback()***
- ***HAL_DMA_UnRegisterCallback()***

Lab steps :

1. Implement signals and monitor with SWV Trave Timeline Graph -- > **In lab you need to probe by oscilloscope**.
    1. You can use look up table for generating waves or arm-math library for sine wave.
    2. In the oscilloscope, you can check the frequency and amplitude of your signal.

2. Read the push button status in the interrupt mode. --> check the push button functionality by turning on/off a LED.

3. Implement the timer callback method (void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){}) --> Again you can use a LED for debugging.

4. Test the DAC functionality by set a single value to its register. (HAL_DAC_SetValue(&hdac1, DAC1_CHANNEL_1, DAC_ALIGN_12B_R, 4095) --> You can monitor DAC output in oscilloscope.

5. Combine Timer and DAC.

6. Transfer data between Memory and DAC by DMA. (HAL_DAC_Start_DMA)

Image source : www.mtlblog.com/