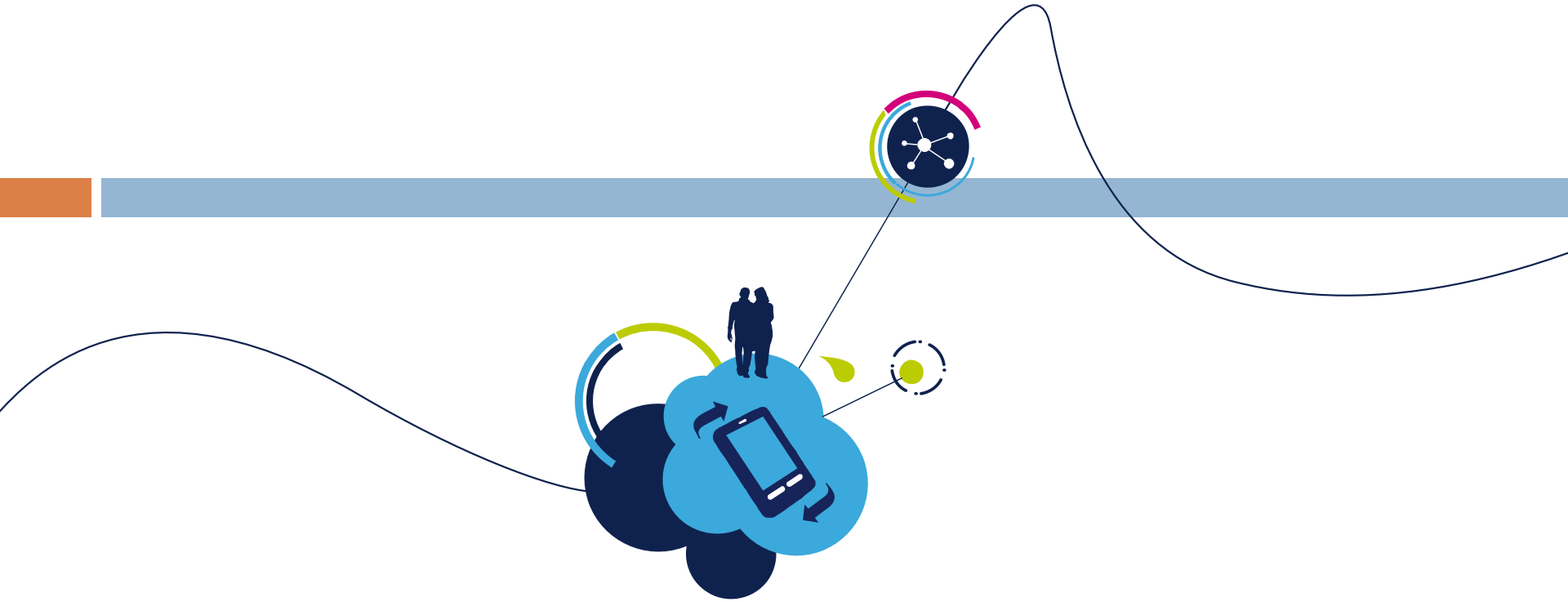
LABORATORIO:
RETI PER AUTOMAZIONE INDUSTRIALE
STM32 NUCLEO

PERIPHERALS

antonino,raucea@dieei.unict.it

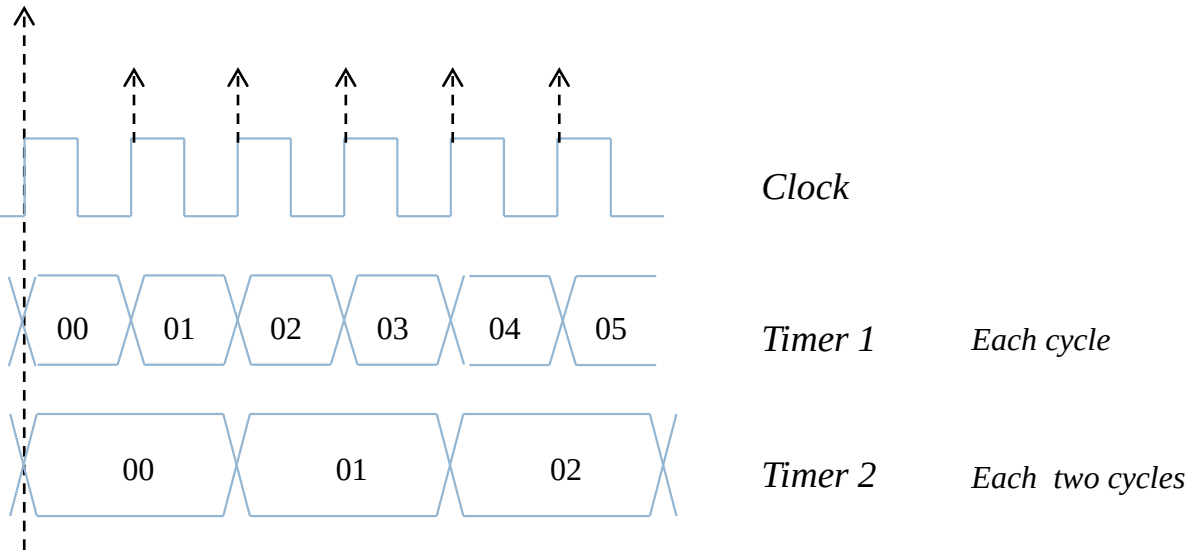
Ing. Antonino Raucea



Timers

Timer

- A Timer is a peripheral that "*counts*"
- It is used to measure time intervals, to generate delays, to timing events, to generate waveforms, etc…
- A clock must be provided
- By means of configuration register it is possible to set the number of clock cycles necessary for the increment of one unit in counting

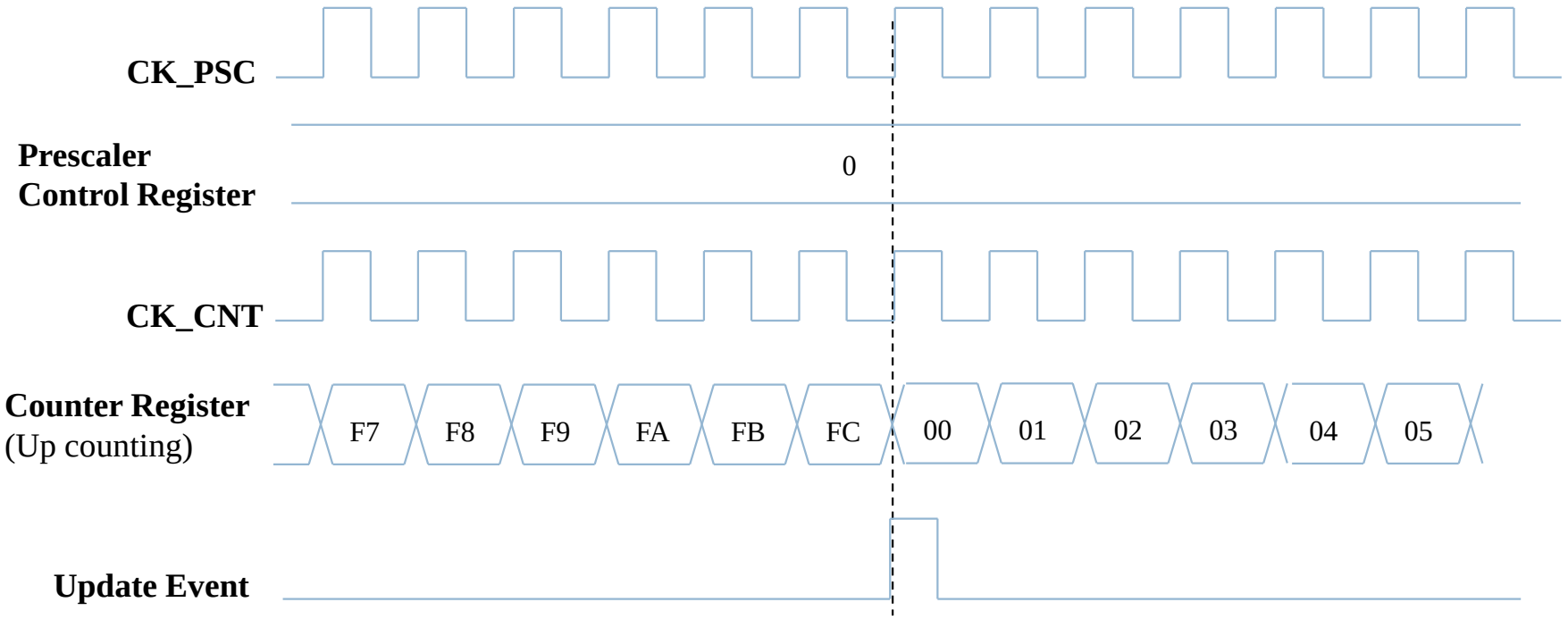


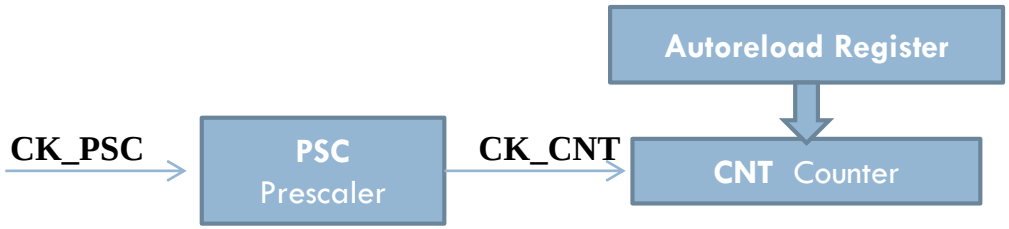
Clock Selection

The counter clock can be provided by the following clock sources:

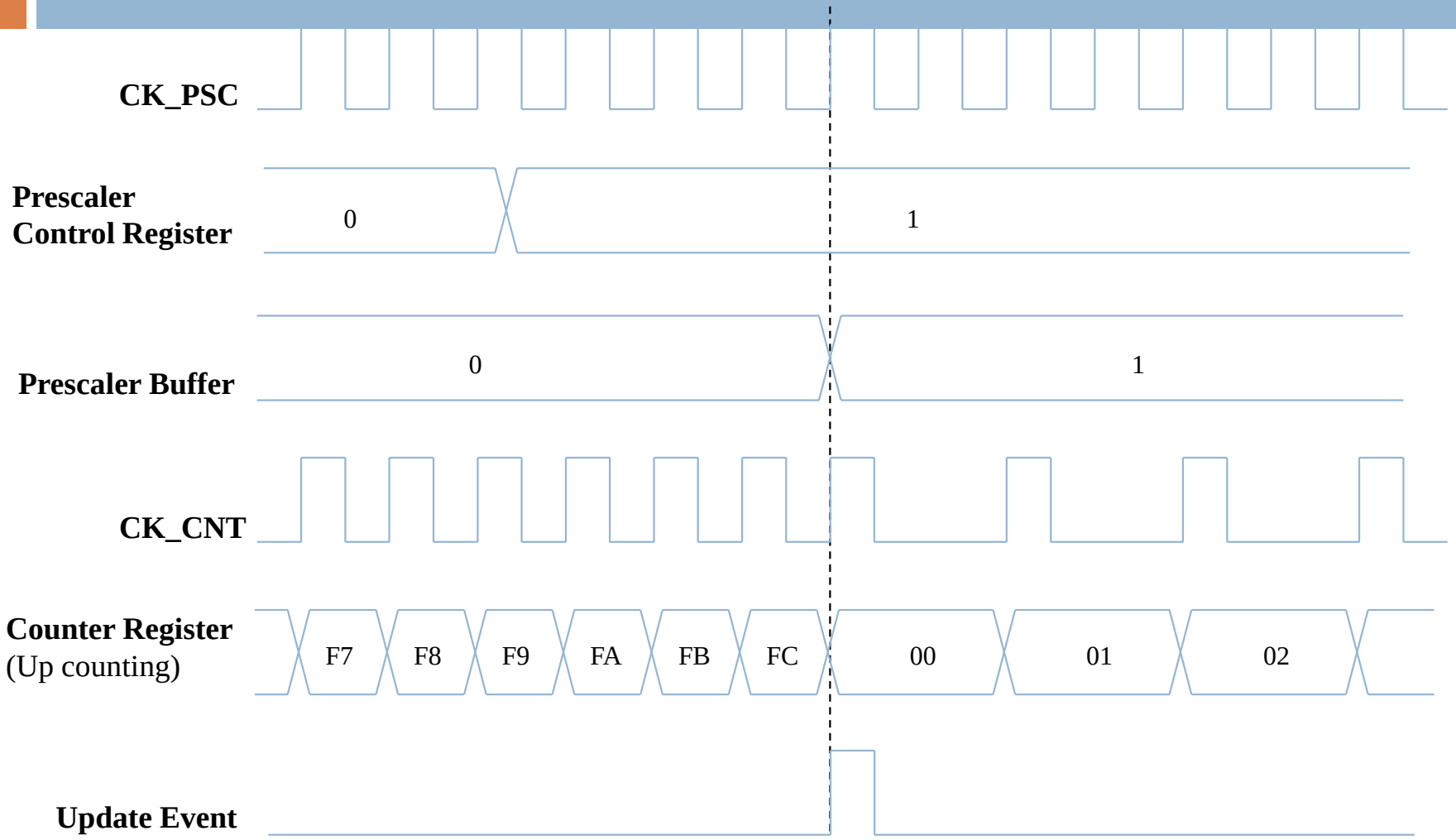
- Internal clock (CK_INT)
- External clock mode1: external input pin (TIx)
- External clock mode2: external trigger input (ETR)
- Internal trigger inputs (ITRx): using one timer as prescaler for another timer.

Counter Timing Diagram (1/2)



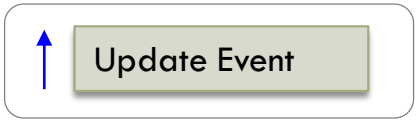


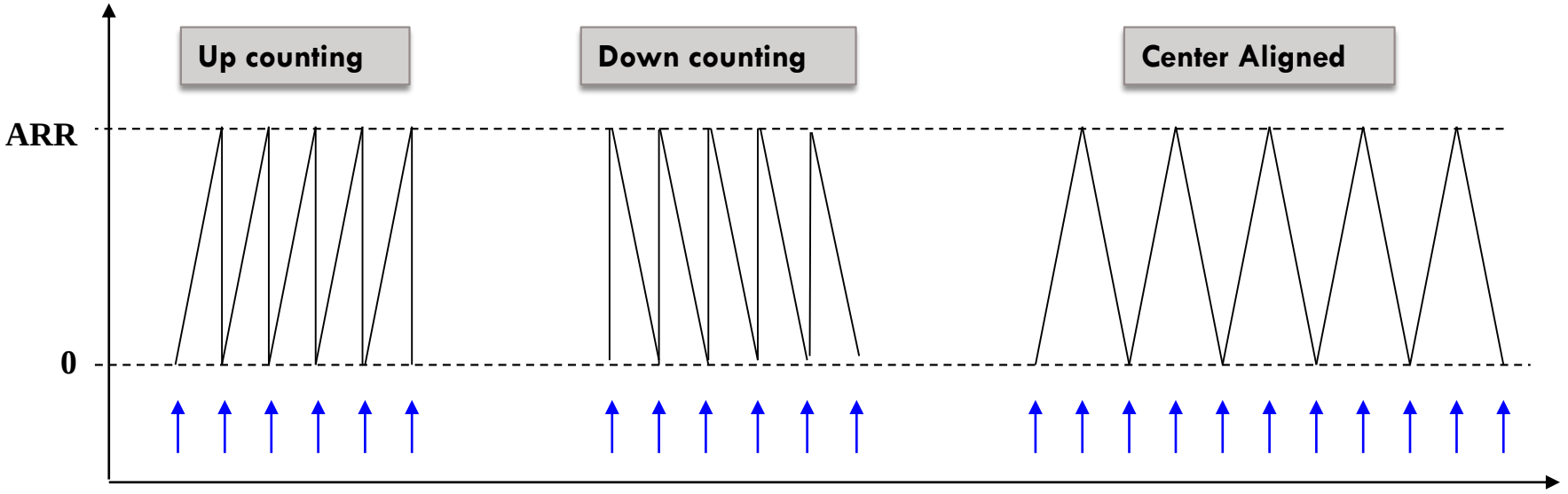
Counter Timing Diagram (2/2)



Counting Modes

- STM32 time-base unit includes:
  - Counter Register (TIMx_CNT)
  - Prescaler Register (TIMx_PSC):
  - Auto-Reload Register (TIMx_ARR)
- There are three counter modes:
  - Up counting mode
  - Down counting mode
  - Center-aligned mode

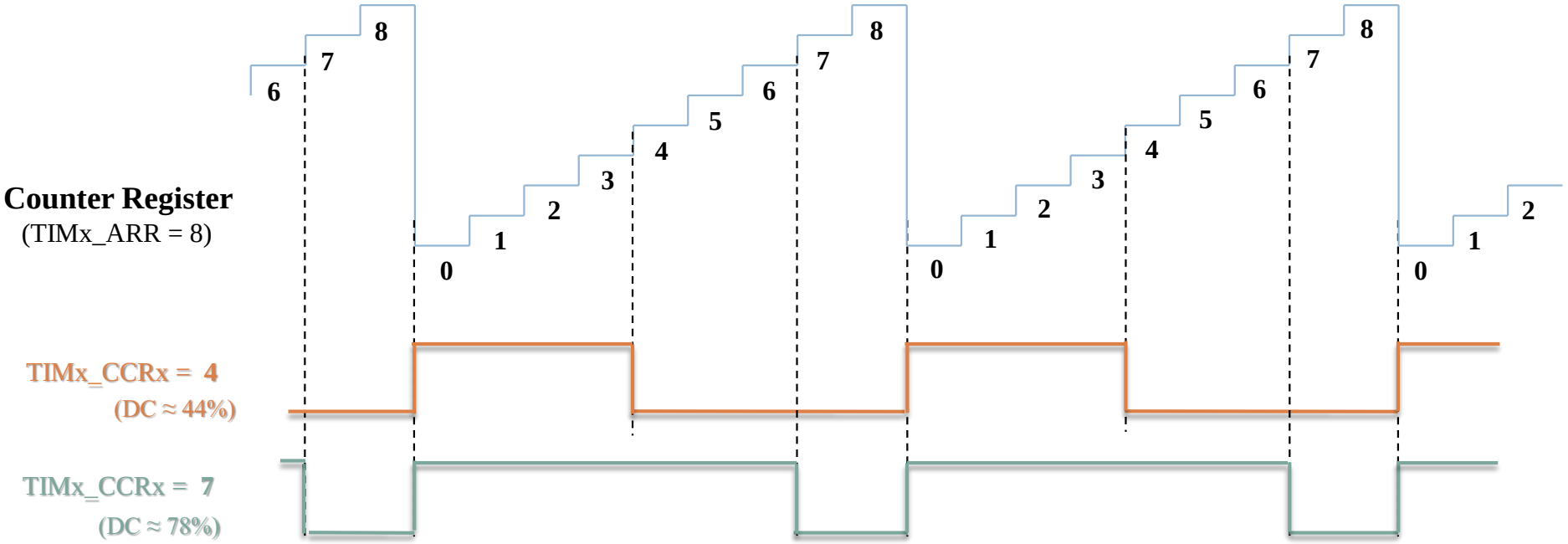




Pulse Width Modulation

Edge Aligned Mode (Standard Mode)

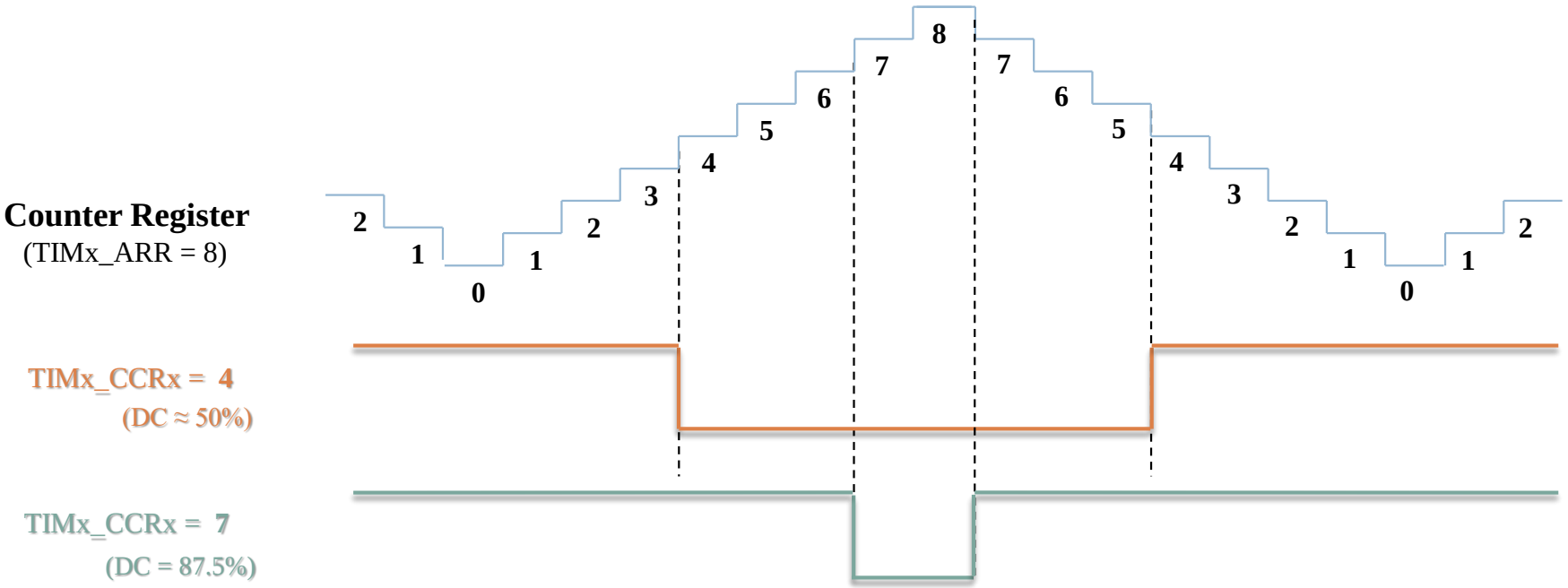
- Pulse width modulation mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.
- The PWM mode can be selected independently on each channel



Pulse Width Modulation

Center Aligned Mode

- Center-aligned mode is active when the CMS bits in TIMx_CR1 register are different from 00

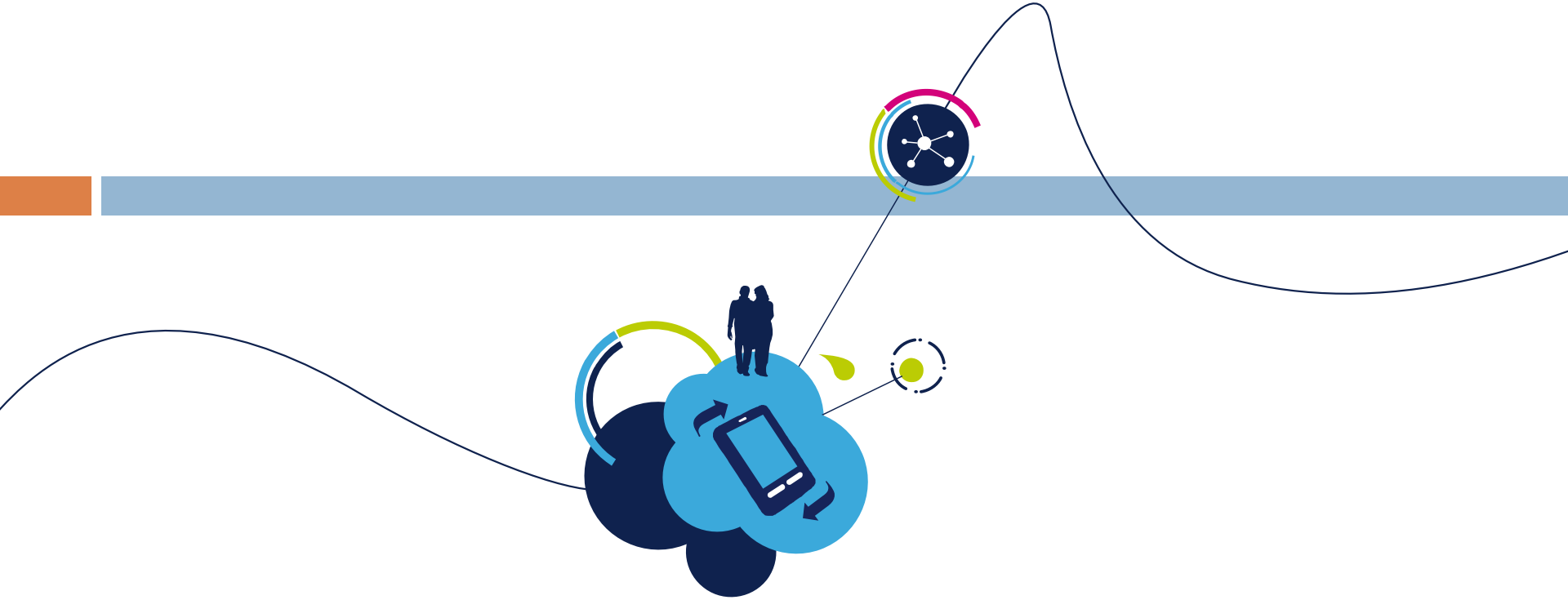


Timers Available

- Advanced-control timers (TIM1/TIM8/TIM20)
  - **16-bit auto-reload counter** driven by a programmable prescaler
  - **complementary PWM** with dead-time insertion
- General-purpose timers (TIM2/TIM3/TIM4)
  - 16-bit or 32-bit auto-reload counter driven by a programmable prescaler
- General-purpose timers (TIM15/TIM16/TIM17)
  - 16-bit auto-reload counter driven by a programmable prescaler
- Basic timers (TIM6/TIM7)
  - 16-bit auto-reload counter driven by a programmable prescaler.

They may be used for a variety of purposes, including measuring the pulse lengths of input signals (**input capture**) or generating output waveforms (**output compare, PWM**)

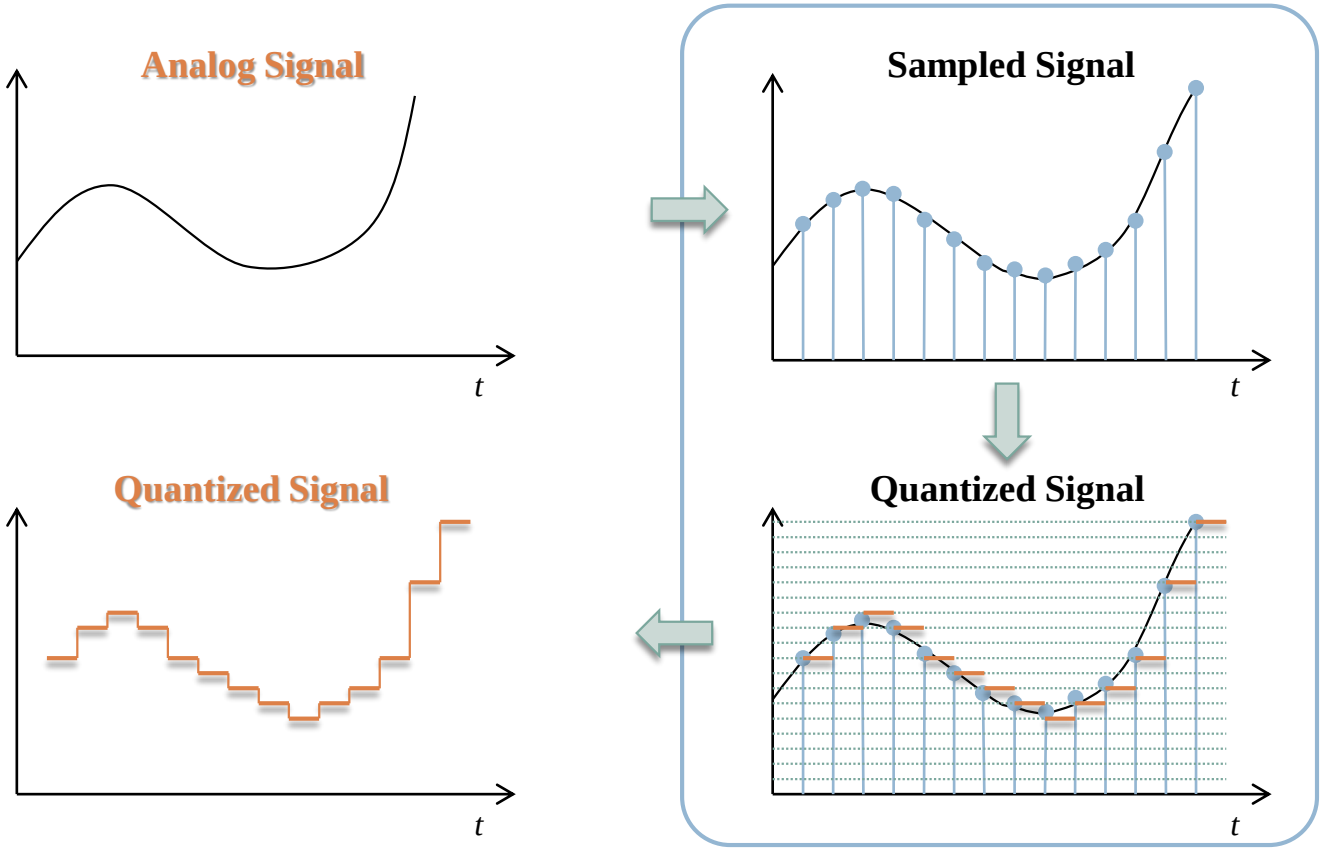
Pulse lengths and waveform periods can be modulated from a few µs to several ms.



ADC

ADC General (1/2)

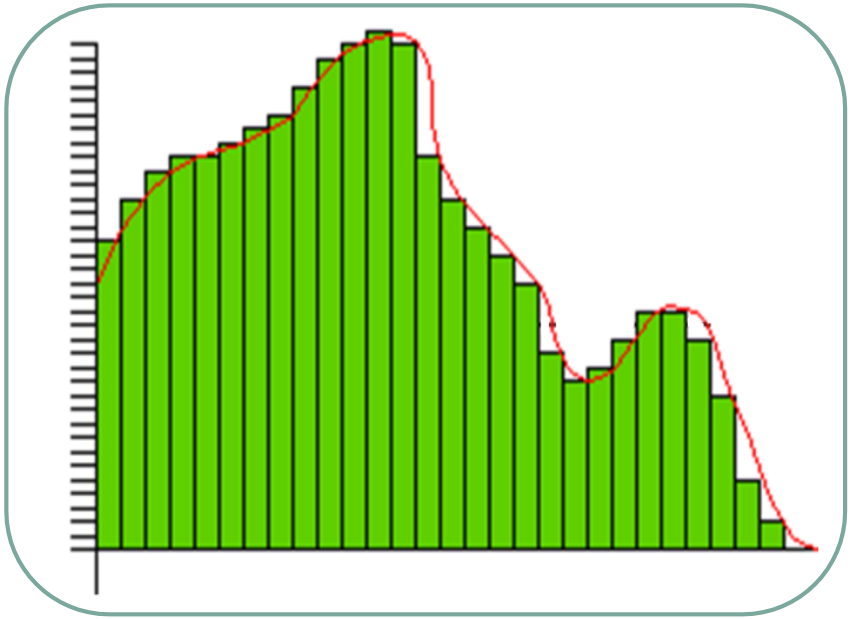
- An analog-to-digital converter (**ADC**) is a device that converts a **continuous** physical quantity (usually voltage) to a **digital** number that represents the quantity's amplitude. The conversion involves **quantization** of the input, so it necessarily introduces a small amount of **error.**



ADC General (2/2)

- The analog signal is continuous in time and it is necessary to convert this to a flow of digital values. It is therefore required to define the rate at which new digital values are sampled from the analog signal. The rate of new values is called the **sampling rate** or **sampling frequency** of the converter

- Since a practical ADC cannot make an instantaneous conversion, the input value must necessarily be held constant during the time that the converter performs a conversion (called the **conversion time**).



ADC Features (1/2)

- Up to 4 ADCs
- Programmable Conversion resolution : 12, 10, 8 or 6 bit
- External Analog Input Channels for each of the 4 ADCs:
  - 5 fast channels from dedicated GPIOs pads
  - Up to 11 slow channels from dedicated GPIOs pads
- ADC conversion time:
  - Fast channels : up to 5.1Ms/s with 12 bit resolution in single mode
  - Slow channels: up to 4,8Ms/s with 12 bit resolution in single mode
- Self-calibration
- Configurable regular and injected channels
- Hardware assistant to prepare the context of the injected channels to allow fast context switching
- Can manage Single-ended or differential inputs

ADC Features (2/2)

- 3 internal channels connected to :
  - Temperature sensor Vsense connected to ADC1
  - Internal voltage reference VREFINT connected to all ADCs
  - VBAT/2 power supply connected to ADC1
- Programmable sampling time
- Single, continuous and discontinuous conversion modes
- Dual ADC mode
- Left or right Data alignment with inbuilt data coherency
- Software or Hardware start of conversion
- 3 Analog Watchdog per ADC
- DMA capability
- Auto Delay insertion between conversions
- Interrupt generation

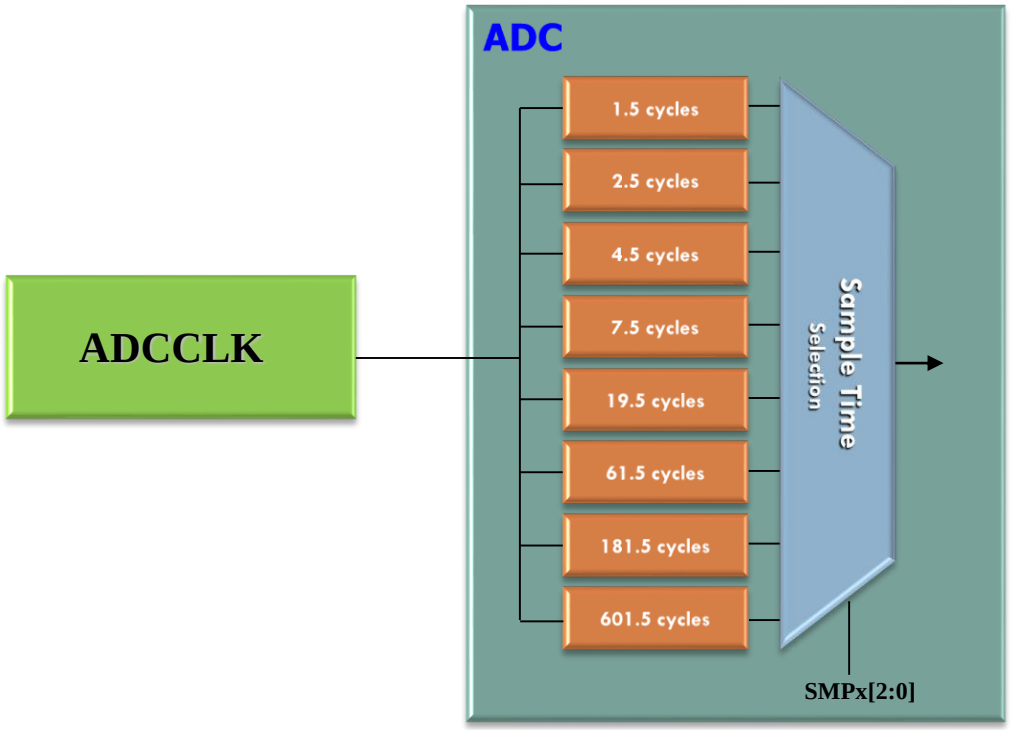
ADC Pins

Name	Signal Type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the ADC, $1.8\ V \le V_{REF+} \le V_{DDA}$
V_{DDA}	Input, analog supply	Analog power supply equal to V_{DD} and $1.8\ V \le V_{DDA} \le V_{DD}$ (3.6 V)
V_{REF-}	Input, analog reference negative	The lower/negative reference voltage for the ADC, $V_{REF-} = V_{SSA}$
V_{SSA}	Input, analog supply ground	Ground for analog power supply equal to V_{SS}
$V_{INP}[18:1]$	Positive input analog channels for each ADC	Connected either to external channels: ADC_INi or internal channels.
$V_{INN}[18:1]$	Negative input analog channels for each ADC	Connected to V_{REF-} or external channels: ADC_INi-1
ADCx_IN16:1	External analog input signals	Up to 16 analog input channels (x=ADC number = 1,2,3 or 4): • 5 fast channels • 11 slow channels

ADC Sampling Time ($T_{Sampling}$)

- Three bits programmable sampling time channel by channel programmable:

  - 1.5 cycles
  - 2.5 cycles
  - 4.5 cycles
  - 7.5 cycles
  - 19.5 cycles
  - 61.5 cycles
  - 181.5 cycles
  - 601.5 cycles



Total Conversion Time

- Total conversion Time = $T_{Sampling}$ + $T_{Conversion}$

Resolution	$T_{Conversion}$
12 bits	12,5 Cycles
10 bits	10,5 Cycles
8 bits	8,5 Cycles
6 bits	6,5 Cycles

Resolution	Total conversion Time (When FADC = 72MHz)	
12 bits	12,5 + 1,5 = 14cycles	19.4 us
10 bits	10,5 + 1,5 = 12 cycles	16,6 us
8 bits	8,5 + 1,5 = 10 cycles	13,8 us
6 bits	6,5 + 1,5 = 8 cycles	11,1 us

End of sampling

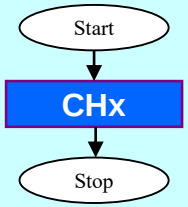
- The ADC indicates the end of sampling phase by setting the EOSMP flag only for regular conversion.
- The EOSMP flag is cleared by software by writing1 to it.
- An interrupt can be generated if the EOSMPIE bit is set in the ADC_IER register.

➔As soon as the sampling is completed it is possible to prepare next conversion (for instance switching I/Os) during the conversion phase.

ADC conversion modes

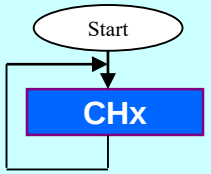
Single channel

single conversion mode



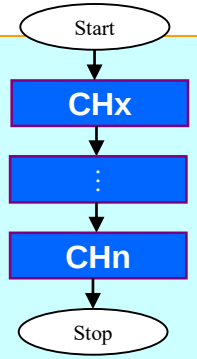
Single channel

Continuous conversion mode



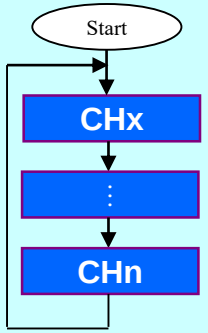
Multi-channels (Scan)

single conversion mode



Multi-channels (Scan)

continuous conversion mode



Discontinuons conversion mode