Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C

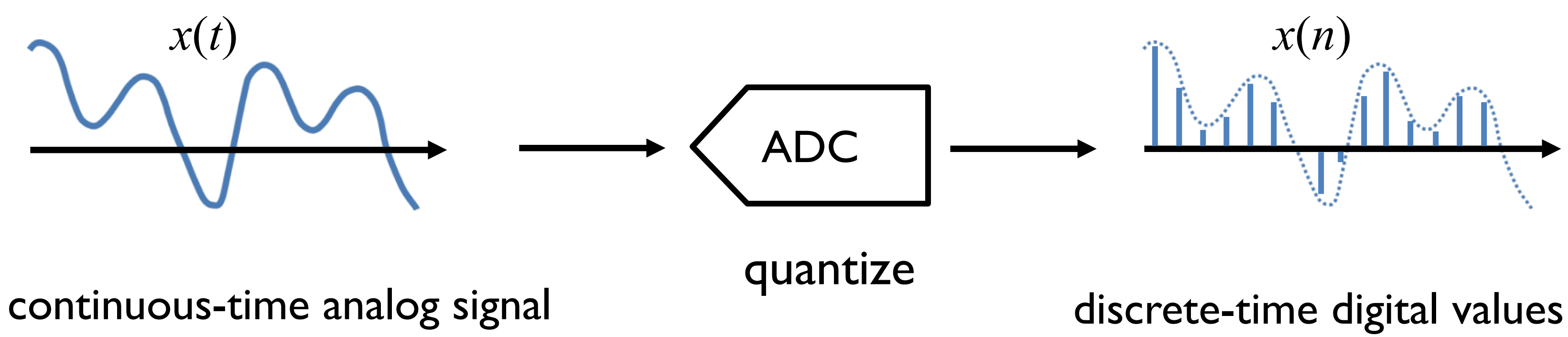
Chapter 20
Analog-to-Digital Converter (ADC)

Dr. Yifeng Zhu
Electrical and Computer Engineering
University of Maine

Spring 2015

Analog-to-Digital Converter (ADC)

- ADC is important to almost all application fields
- Converts a continuous-time voltage signal within a given range to discrete-time digital values to quantify the voltage's amplitudes

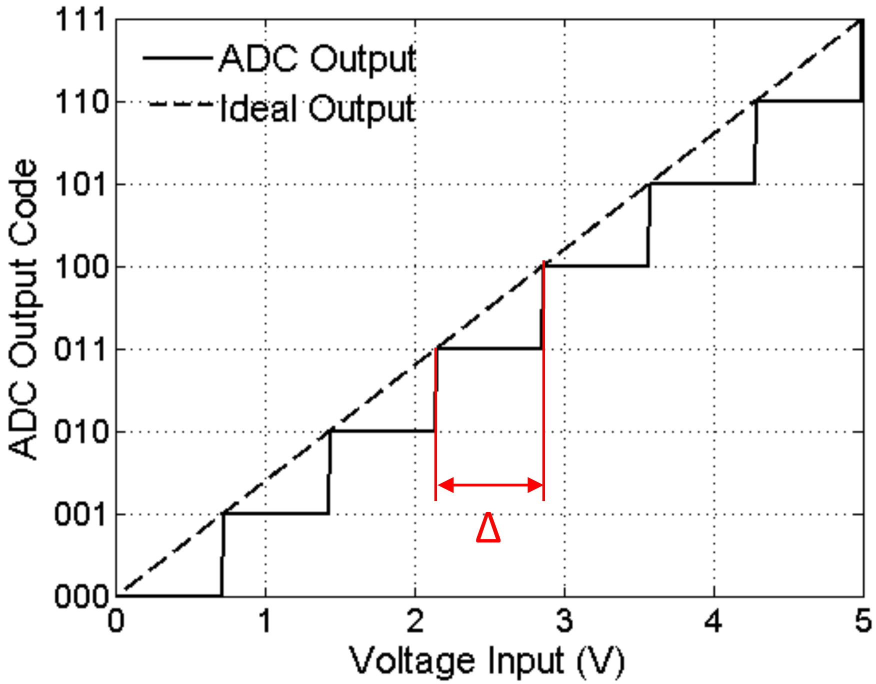


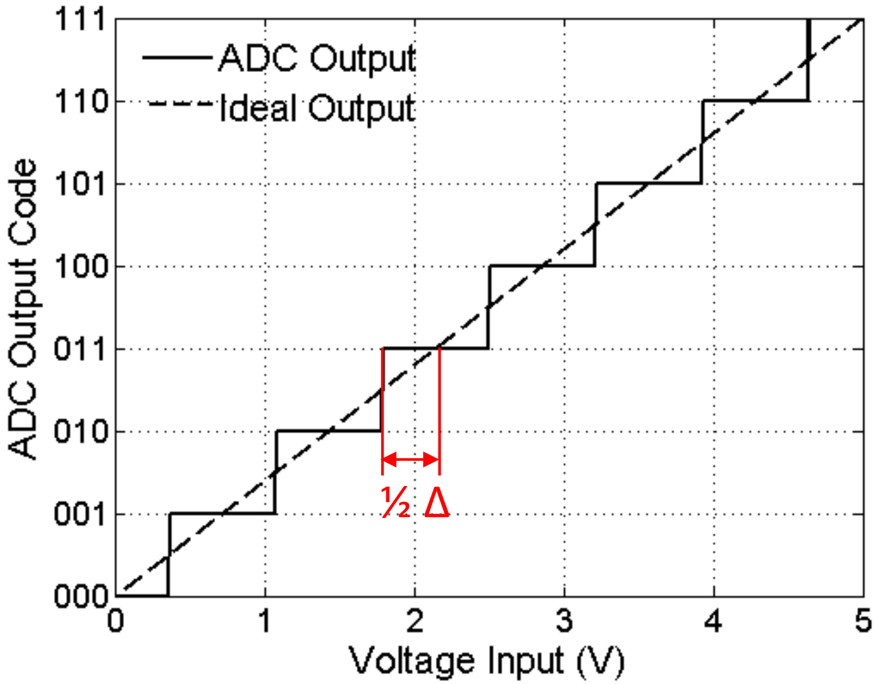
Analog-to-Digital Converter (ADC)

- Three performance parameters:
  - sampling rate: number of conversions per second (thousands to milions)
  - resolution: number of bits in ADC output (6 to 24)
  - power dissipation: power efficiency of ADC
- Many ADC implementations:
  - sigma-delta
    - for apps requiring low sampling rate (<100 kHz) but high resolution (12-24 bit)
    - e.g.: voice and audio apps in cell phones
  - successive-approximation (SAR)
    - for low-power data acquisitions w/ moderate sampling rates (<5MHz)
    - e.g.: STM32 microcontrollers
  - pipelined
    - for high-speed apps (sampling rate > 5MHz) and relatively low resolution
    - e.g.: radar communication

Resolution

- Resolution is determined by number of bits (in binary) to represent an analog input.
- Example of two quantization methods (N = 3)





$$Digital\ Result = floor\left(2^3 \times \frac{V}{V_{REF}}\right)$$

$$Digital\ Result = round\left(2^3 \times \frac{V}{V_{REF}}\right)$$

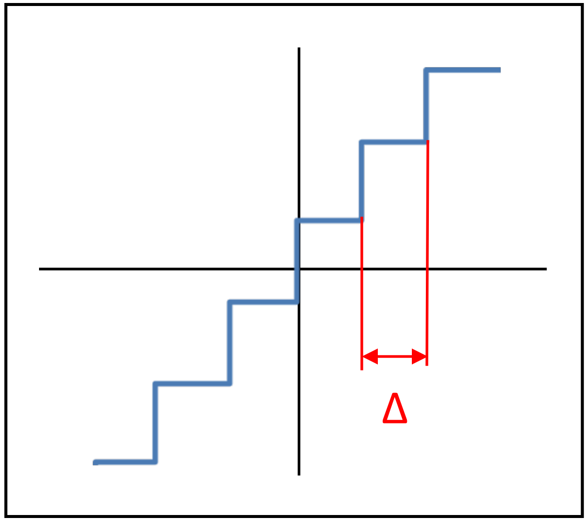
Max quantization error = Δ = $V_{REF}/2^3$

Max quantization error = ±½ Δ = ±$V_{REF}/2^4$

$$round(x) = floor(x + 0.5)$$

Quantization Error

- For N-bit ADC, it is limited to ±½Δ
- Δ = is the step size of the converter.

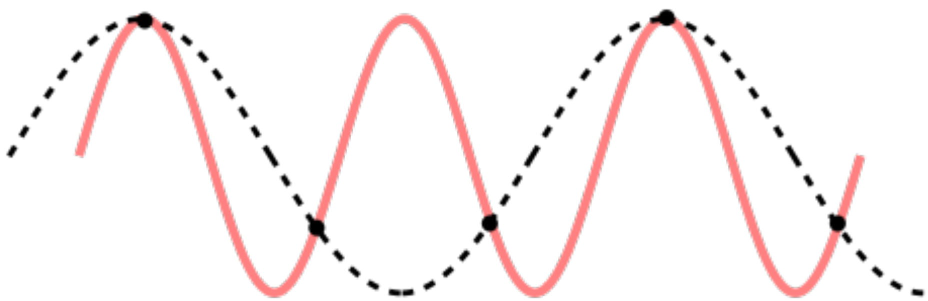


- Example: for 12-bit ADC and input voltage range [0, 3V]

$$Max\ Quantization\ Error = \frac{1}{2}\Delta = \frac{3V}{2\times 2^{12}} = 0.367mV$$

Minimum Sampling Rate: Nyquist–Shannon Sampling Theorem

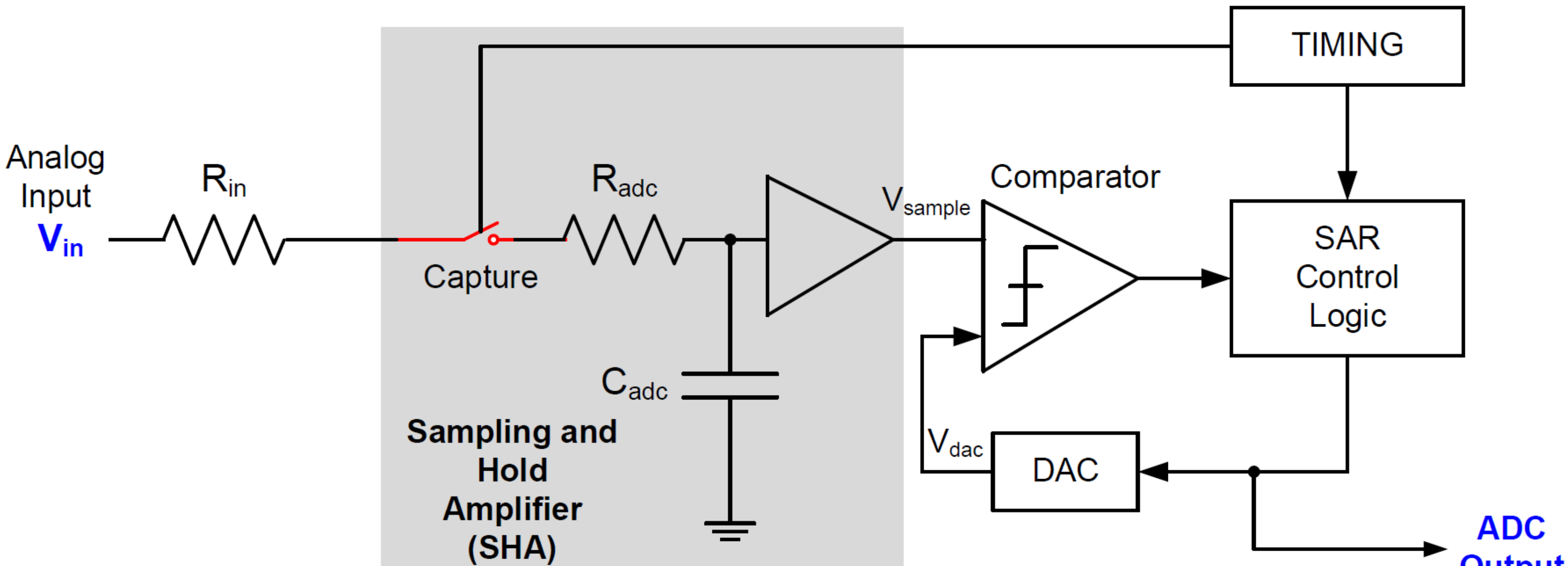
- In order to be able to reconstruct the analog input signal, the sampling rate should be at least twice the maximum frequency component contained in the input signal
- Example of two sine waves have the same sampling values. This is called aliasing.



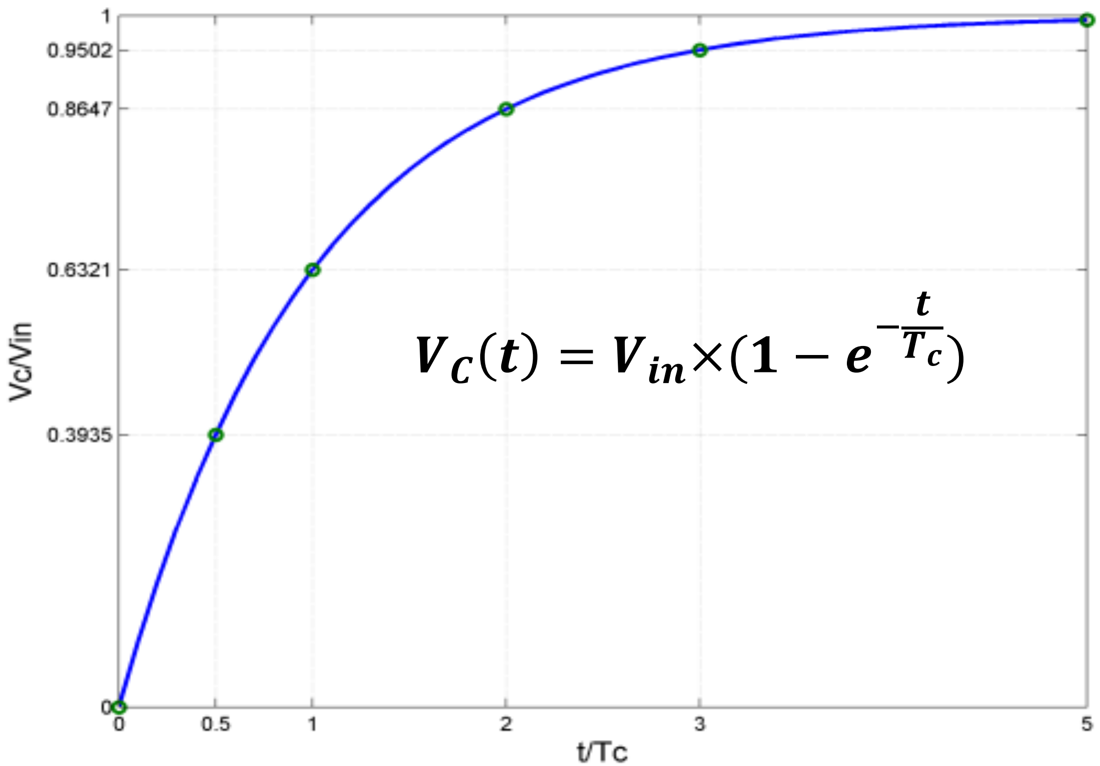
from wiki.com

- Antialiasing (beyond the scope of this course)
  - Pre-filtering: use analog hardware to filtering out high-frequency components and only sampling the low-frequency components. The high-frequency components are ignored.
  - Post-filtering: Oversample continuous signal, then use software to filter out high-frequency components

Successive-approximation (SAR) ADC



Determining Minimum Sampling Time



$$V_C(t) = V_{in} \times (1 - e^{-\frac{t}{T_c}})$$

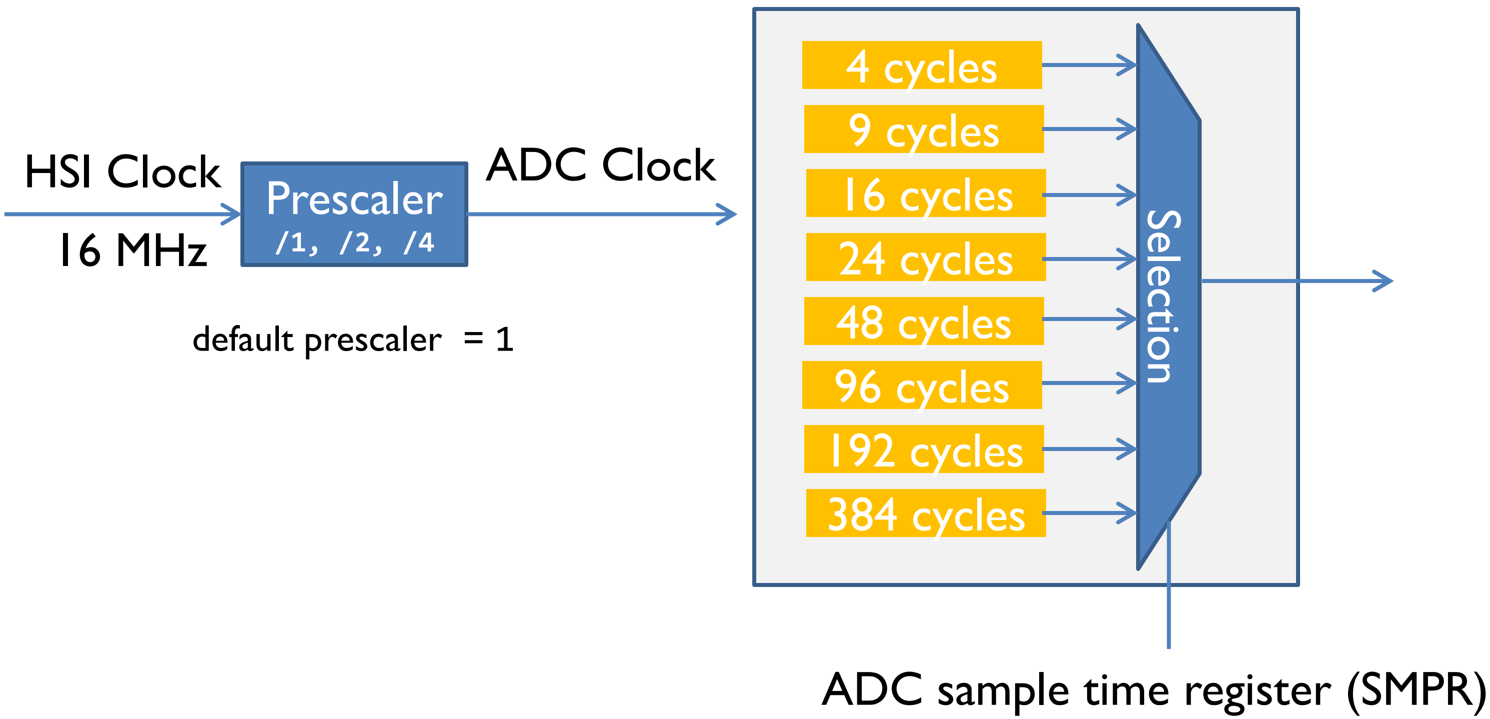
Sampling time is software programmable!

Larger sampling time → Smaller sampling error

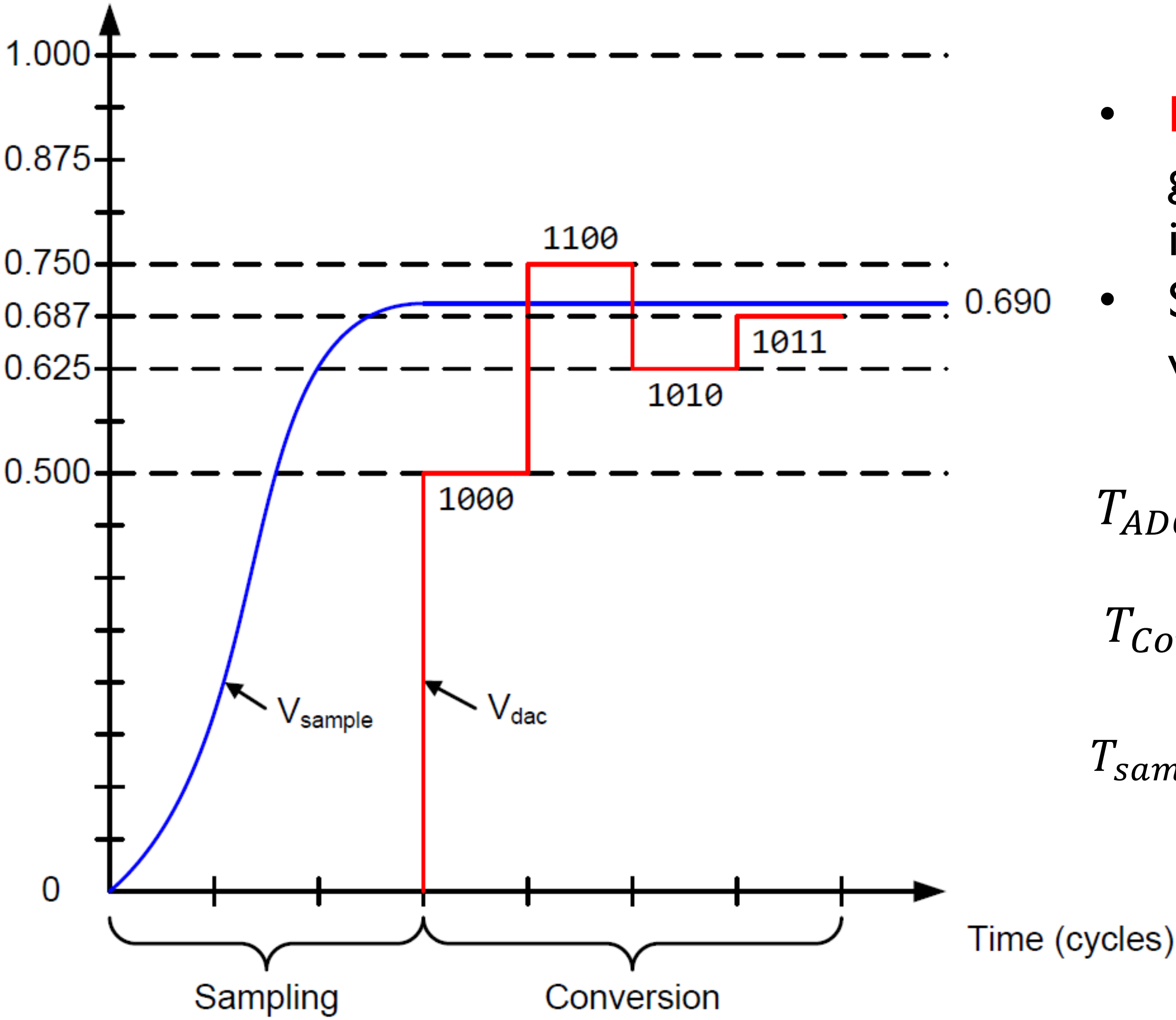
Larger sampling time → Slower ADC speed

Tradeoff

Programming ADC Sampling Time



Successive-approximation (SAR) ADC



- Binary search algorithm to gradually approaches the input voltage
- Settle into ±½ LSB bound within the time allowed

$$T_{ADC} = T_{sampling} + T_{Conversion}$$

$$T_{Conversion} = \mathrm{N} \times T_{ADC\_Clock}$$

$T_{sampling}$ is software configurable

ADC Conversion Time

$$T_{ADC} = T_{sampling} + T_{Conversion}$$

Suppose ADCCLK = 16 MHz and Sampling time = 4 cycles

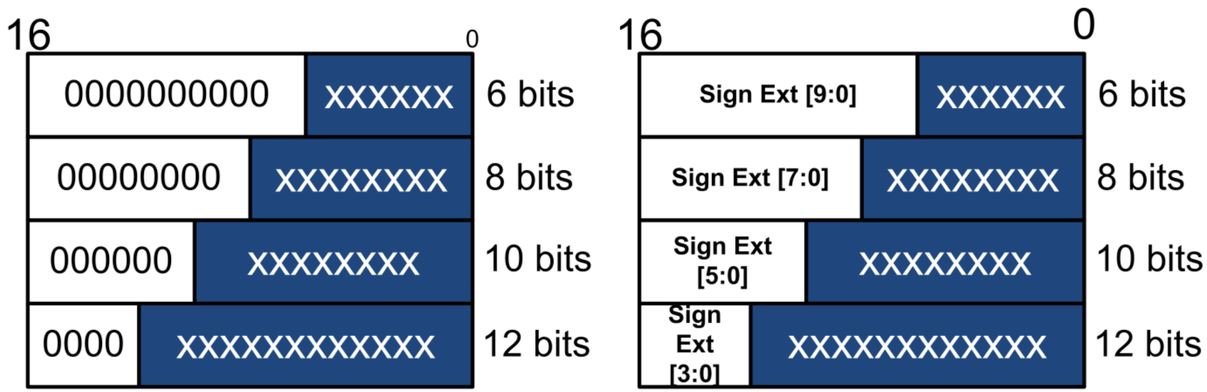
For 12-bit ADC

$$T_{ADC} = 4 + 12 = 16\ cycles = 1\mu s$$

For 6-bit ADC

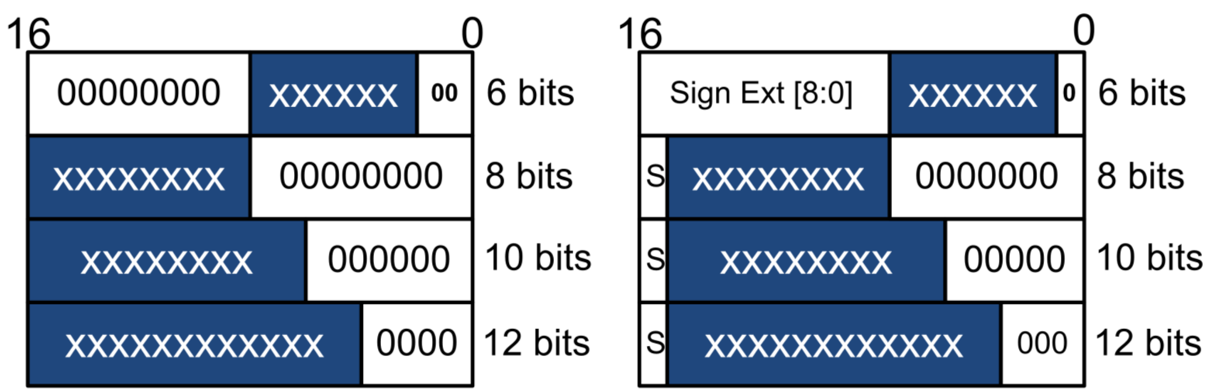
$$T_{ADC} = 4 + 6 = 10\ cycles = 625ns$$

Data Alignment



Right alignment for a regular channel

Right alignment for an injected channel

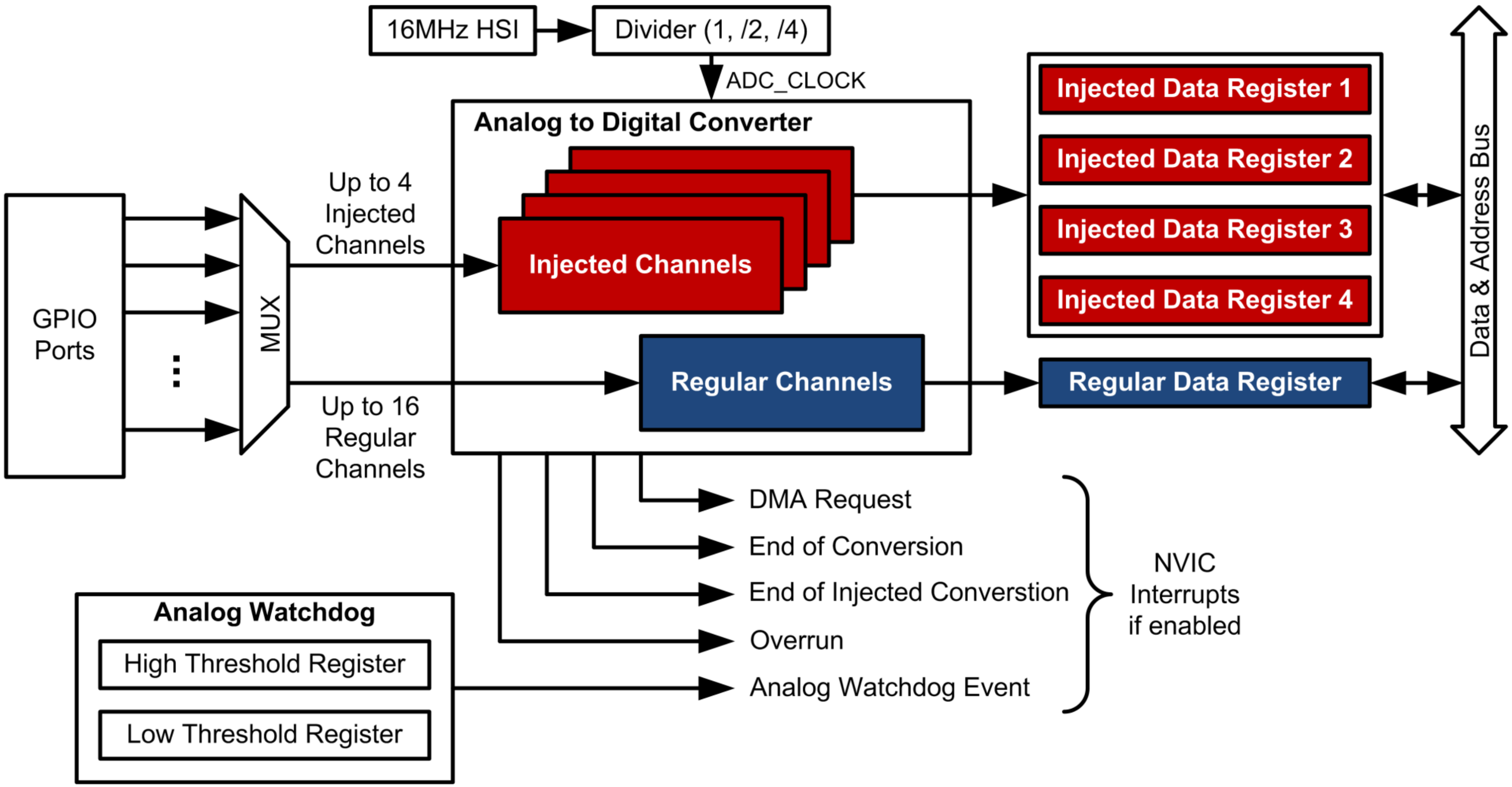


Left alignment for a regular channel

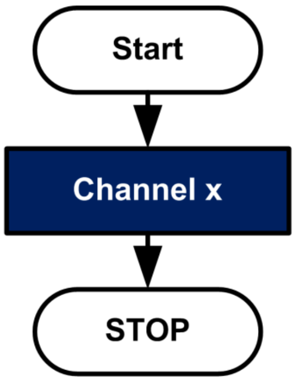
Left alignment for an injected channel

S = Sign bit

ADC: Regular vs injected

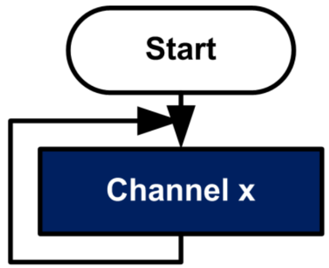


ADC Mode



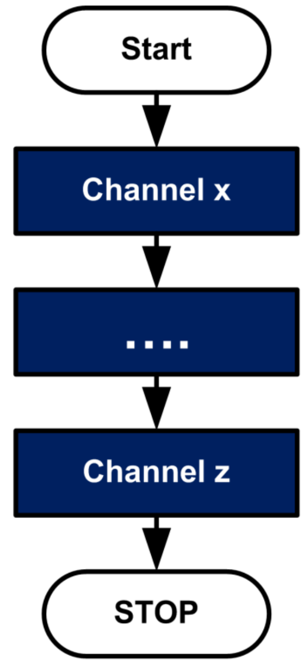
Single Channel, Single Conversion Mode

CONT in ADC_CR2 = 0
SCAN in ADC_CR2 = 0



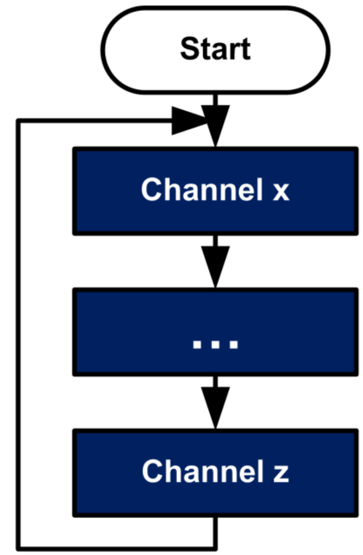
Single Channel, Continuous Conversion Mode

CONT in ADC_CR2 = 1
SCAN in ADC_CR2 = 0



Scan Mode with Single Conversion

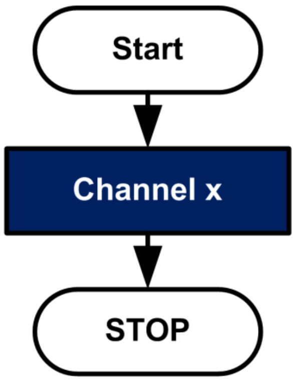
SCAN in ADC_CR2 = 1
CONT in ADC_CR2 = 0



Scan Mode with Continuous Conversion

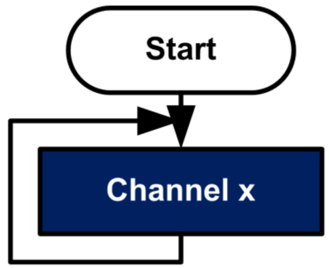
SCAN in ADC_CR2 = 1
CONT in ADC_CR2 = 1

ADC Mode



Single Channel, Single Conversion Mode

CONT in ADC_CR2 = 0
SCAN in ADC_CR2 = 0



Single Channel, Continuous Conversion Mode

CONT in ADC_CR2 = 1
SCAN in ADC_CR2 = 0

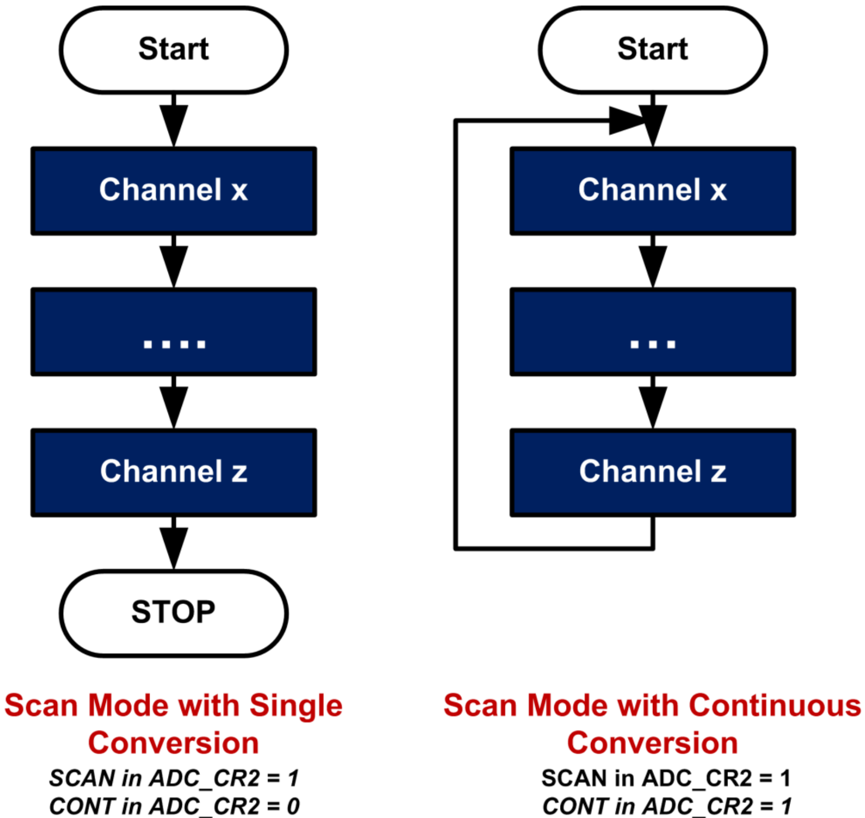
Regular channel:

1. Set SWSTART in ADC_CR2
2. The channel is selected by SQ1[4:0] in SQR5
3. Result is stored in ADC_DR
4. EOC is set after conversion
5. Interrupt is generated if EOCIE is set

Injected channel:

1. Set JSWSTART in ADC_CR2
2. The channel is selected by JSQ1[4:0] in JSQR
3. Result is stored in ADC_JDR1
4. JEOC is set after conversion
5. Interrupt is generated if JEOCIE is set

ADC Mode

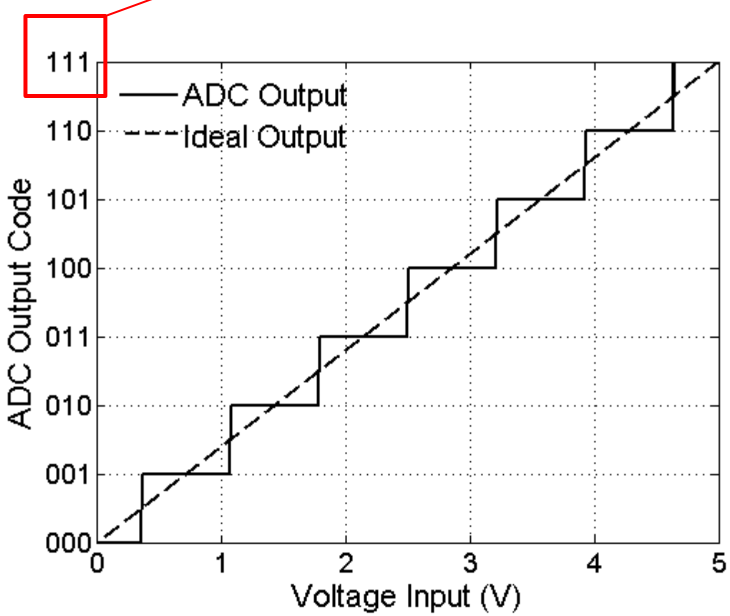


- Channels are selected by ADC_SQRx registers for regular channels, and by ADC_JSQR register for injected channel
- All channels in a regular group share the same result register ADC_DR. Make sure to read data between consecutive sampling.

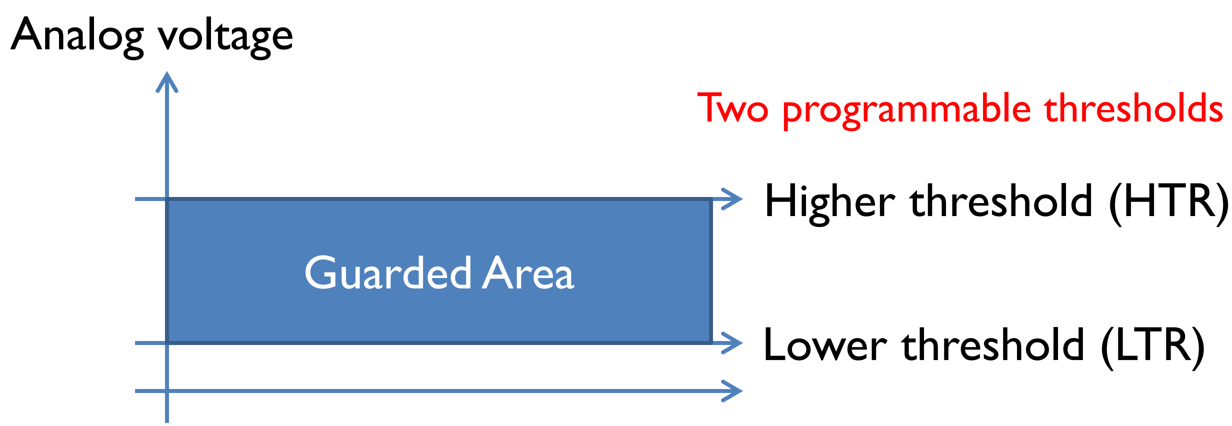
V_{REF}

- Some chips does not expose V_{REF} to a pin
  - STM32L LQFP64 does not have V_{REF} pin
  - STM32L LQFP100 does
- Infer internal V_{REF}
  - How?

Corresponds to V_{REF}

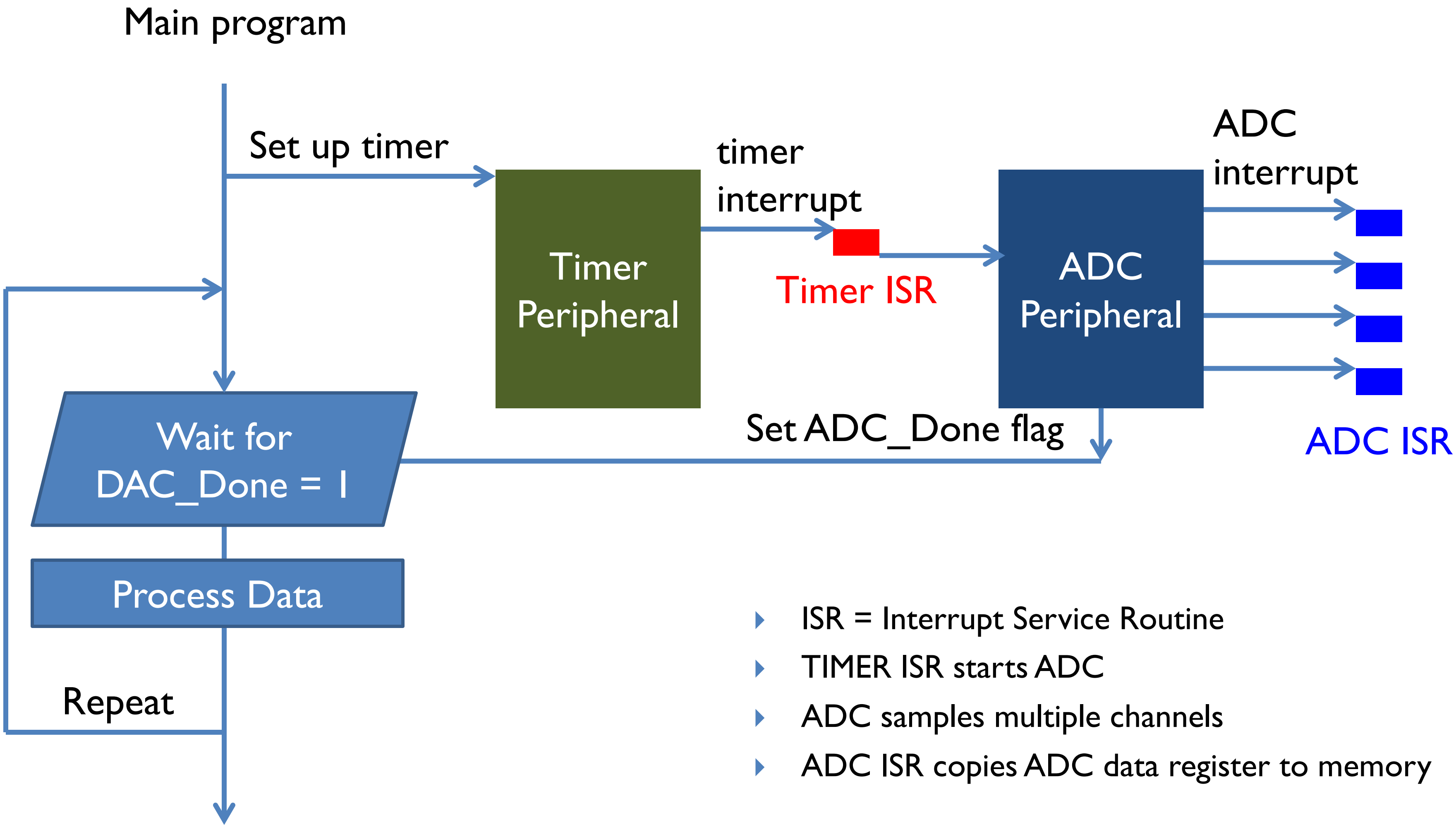


Analog Watchdog



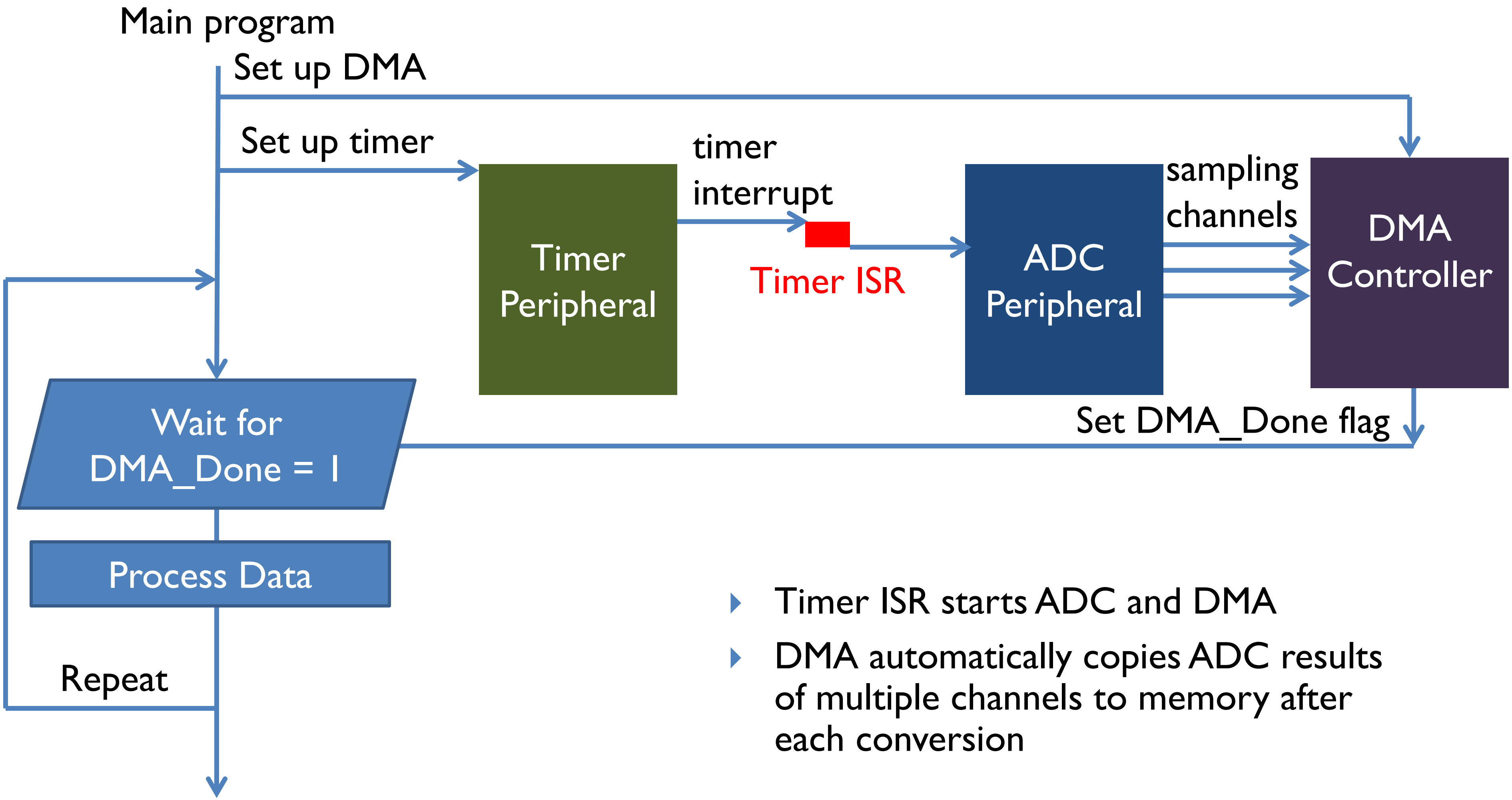
- If $V < V_{LTR}$ or $V > V_{HTR}$, the analog watchdog (AWD) flag (in ADC Status Register) is set, generating an interrupt to the processor
- The monitor is automatically performed by hardware, not software
- Convenient and efficient feature
- Help processor detect exceptions and recover from specific situations
  - For example, monitor sensor data and raise alarm on some level.

Example: ADC with Timer Interrupts



- ISR = Interrupt Service Routine
- TIMER ISR starts ADC
- ADC samples multiple channels
- ADC ISR copies ADC data register to memory

Example: ADC with Timer and DMA



- Timer ISR starts ADC and DMA
- DMA automatically copies ADC results of multiple channels to memory after each conversion