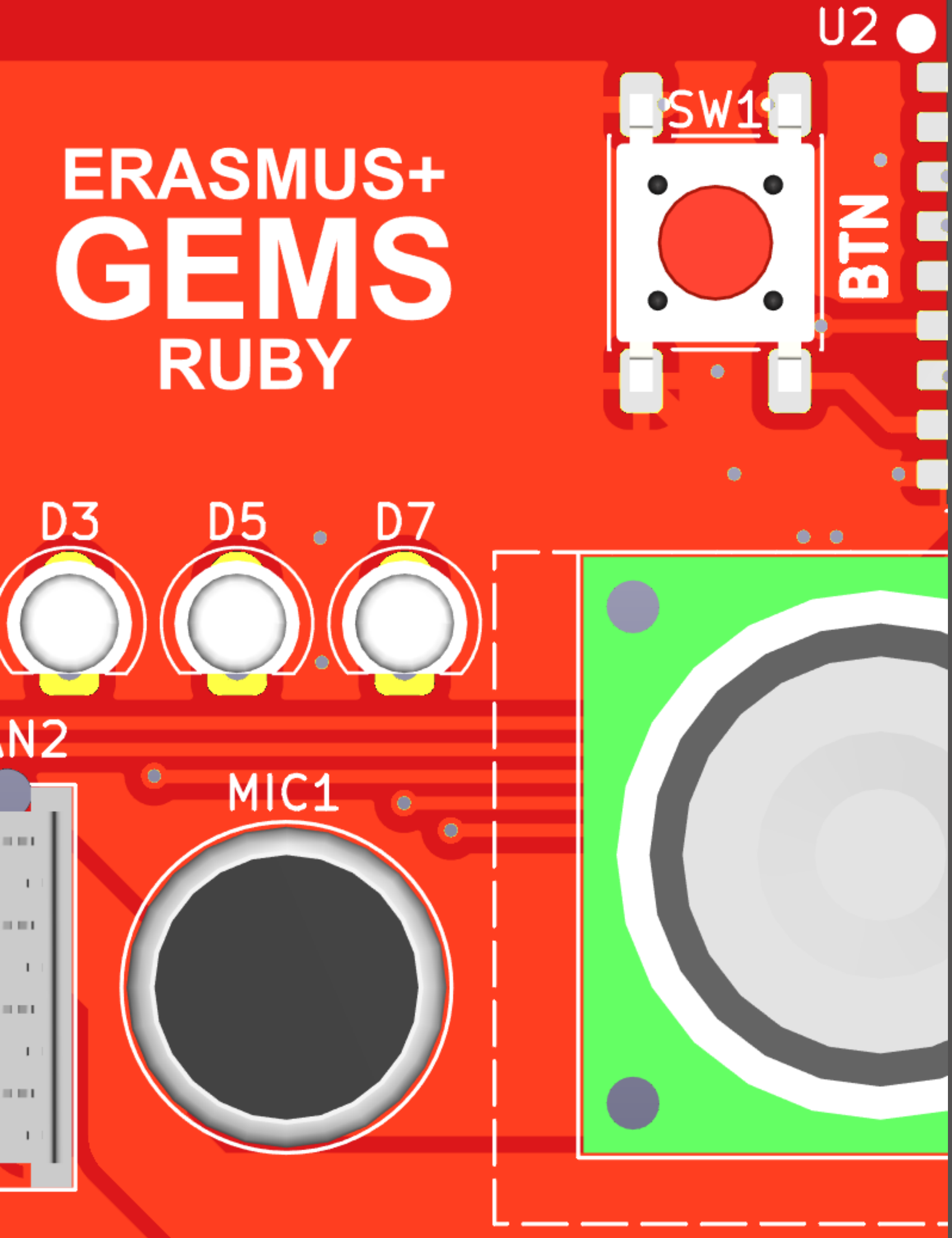


Data acquisition

Generating digital data from sensors

Cristina Losada-Gutiérrez (University of Alcala)

Index

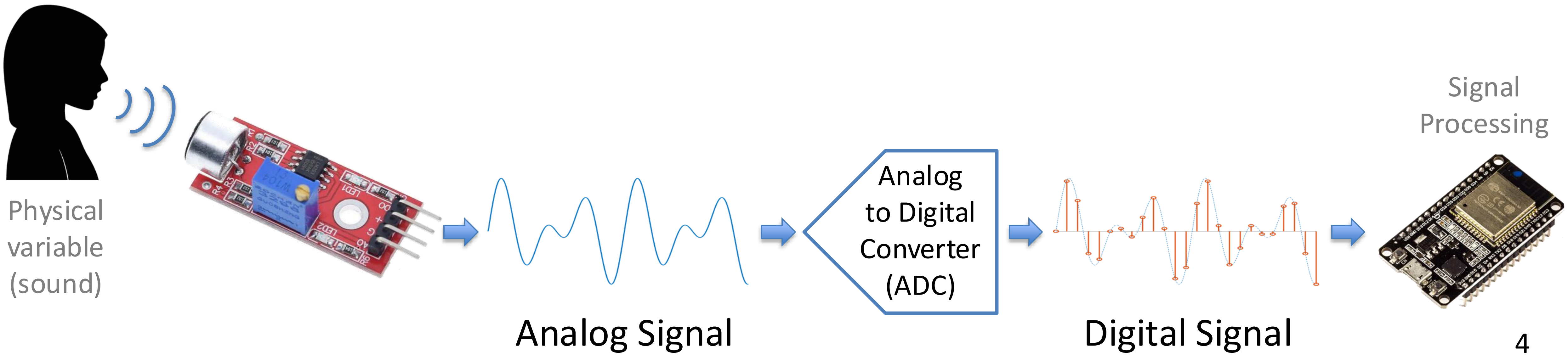
Index

TU Delft

Data acquisition (DAQ)

"**Sampling** signals that measure a **physical phenomena** and **convert** them into a **digital** form, for being processed in a digital device" **[1]**.

[1] Austerlitz, H., **Data Acquisition Techniques Using PCs** (2nd Edition), Academic Press, 2003.



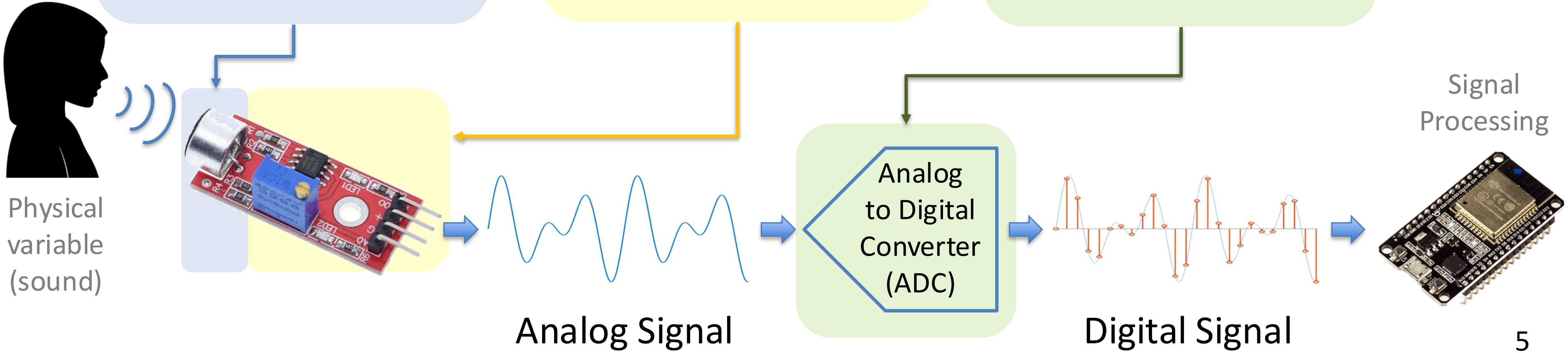
Data acquisition (DAQ)

"**Sampling** signals that measure a **physical phenomena** and **convert** them into a **digital** form, for being processed in a digital device" **[1]**.

Sensor/transducer: converts a physical variable into an electrical signal.

Signal Conditioning: filters to reduce noise and amplifiers to enhance signals.

Data conversión: converts an analog signal into a digital one.



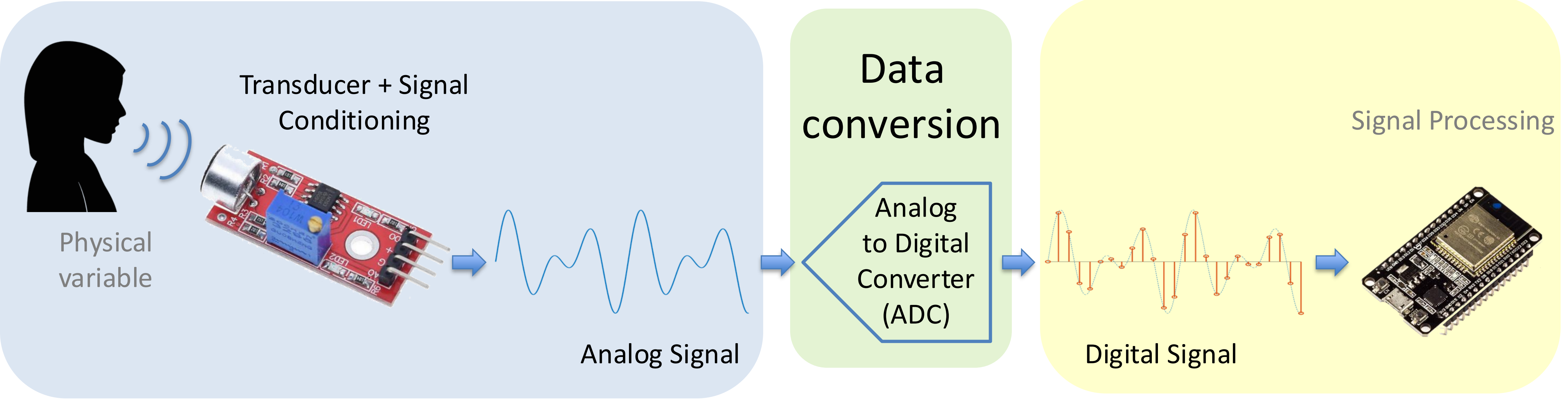
Index

Analog vs Digital signals (I)

In a typical sensory system, analog processes and signals co-exists with digital ones.

- Sensors provide **analog** signals
- But the processing is carried out in a **digital** device

Data conversion is required



Analog vs Digital signals (II)

Analog signals

- Infinite number of values within a continuous range.

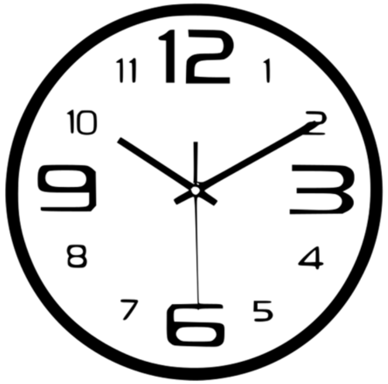


Image by [Clker-Free-Vector-Images](), [CC BY-SA]()

Digital signals

- Value from a finite set of possible values at a given time.

Image by [anaterate](), [CC BY-SA]()

Image by [OpenClipart-Vectors](), [CC BY-SA]()

Analog vs Digital signals (III)

Analog signals

- Infinite number of values within a continuous range.

☺ Values more precise.

☹ More susceptible to noise

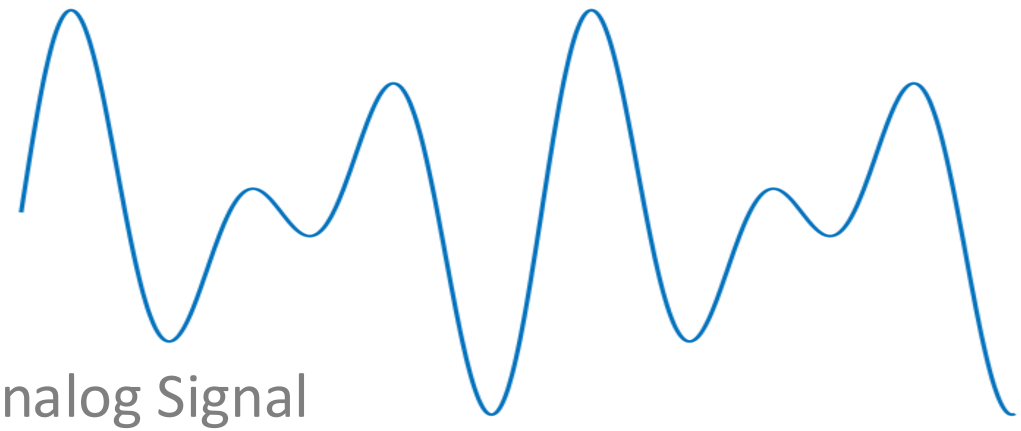
Digital signals

- Value from a finite set of possible values at a given time.

☺ Less sensible to noise, more reliable.

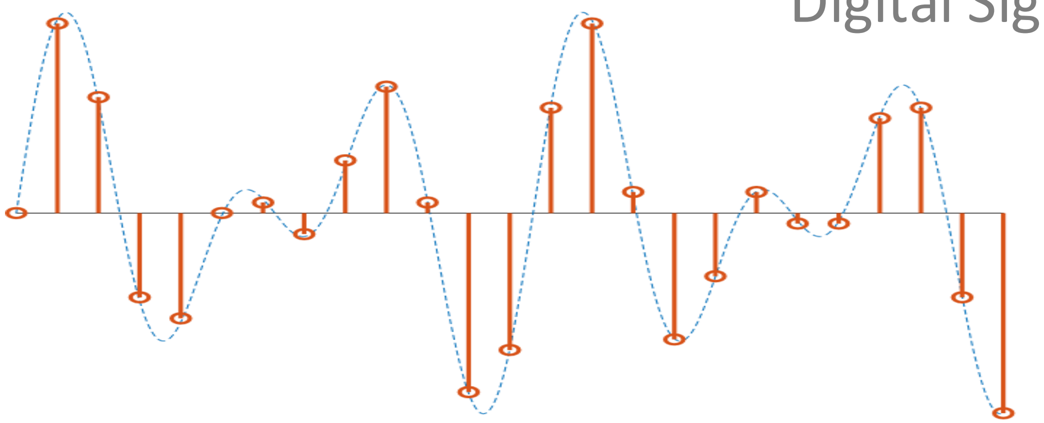
☺ Suited for Computing and digital electronics.

☹ Possible quantization and round-off errors.



Analog Signal

Digital Signal



Index

Introduction (I)

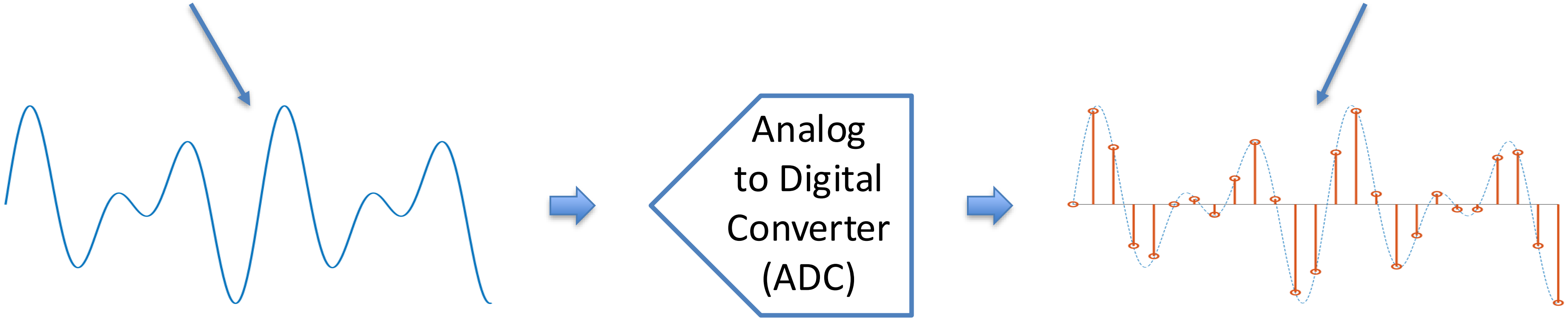
[2] Lai, E., **Practical Digital Signal Processing,** Newnes, 2003, Pages 14-49.

Converts an analog voltage to a digital value (code) representing its magnitude [2],

thereby enabling the use of electronics to interface with the analogue world around us.

Input: analog (continuous) signal

Output: digital code (with *n* bits)

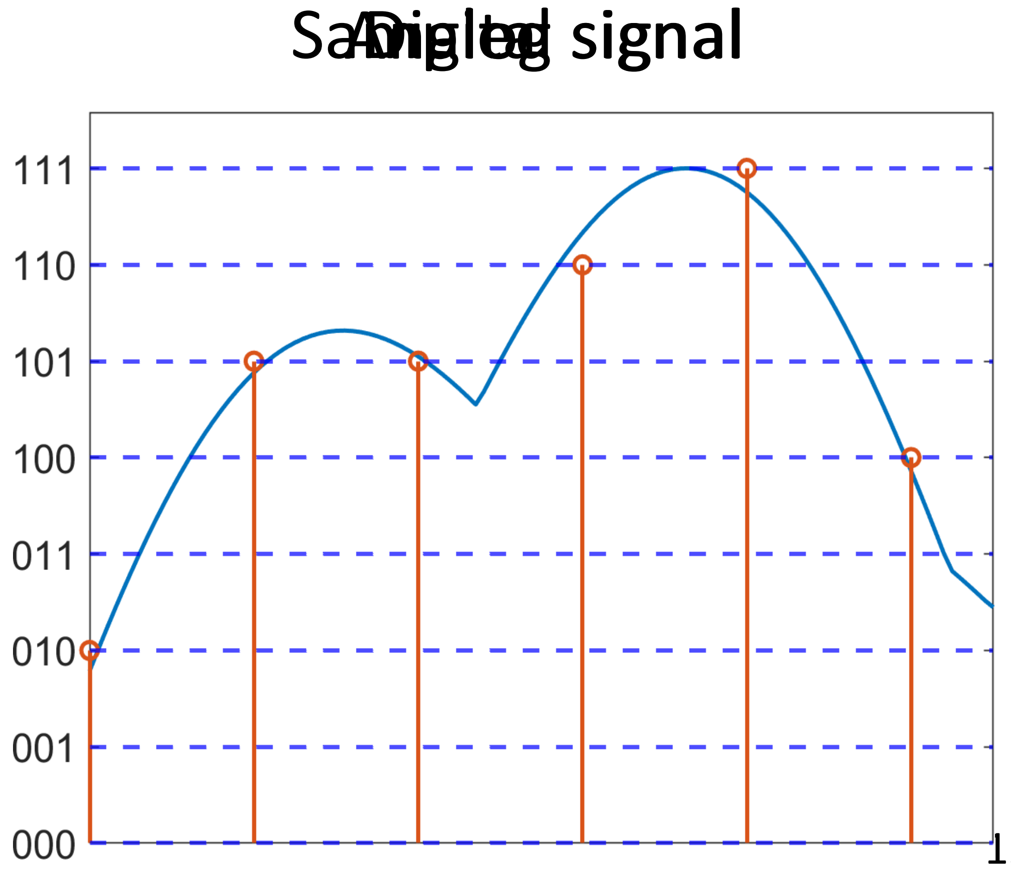


Introduction (II)

Converts an analog voltage to a digital value (code) representing its magnitude [2], thereby enabling the use of electronics to interface with the analogue world around us.

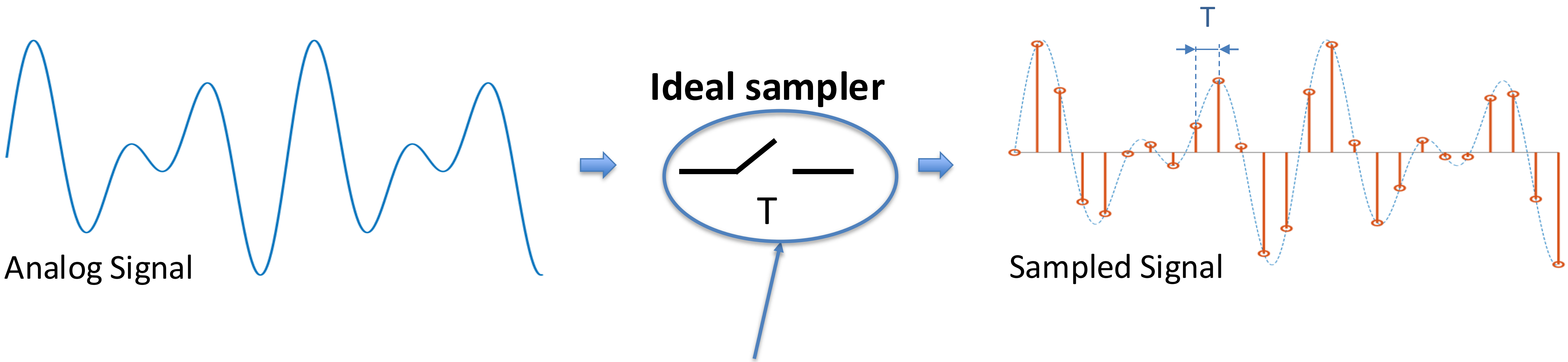
- **Sampling**: the **digital output code** is generated periodically each **sample time** (T)

- **Quantization:** replaces each real number with an approximation from a finite set of discrete values (with *n* bits).



Sampling (I)

- **Measuring** the amplitude of a continuous-time signal **at discrete instants**.
- Converts the continuous signal into a **sampled** one.



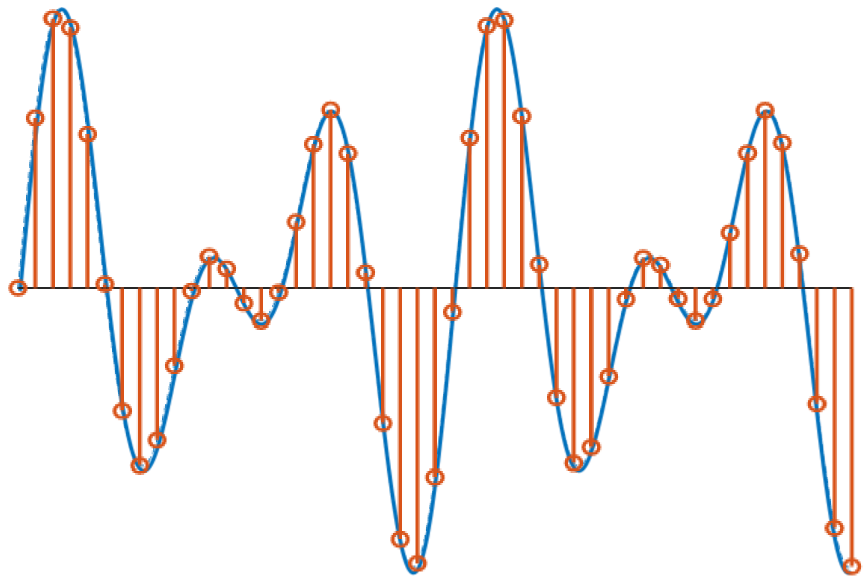
Sample time: $\boldsymbol{T}$ (time between consecutive measurements)

Sampling frequency: $\boldsymbol{f_s = \frac{1}{T}}$ (average number of samples obtained in one second)

Sampling (II)

How to chose the sampling frequency?

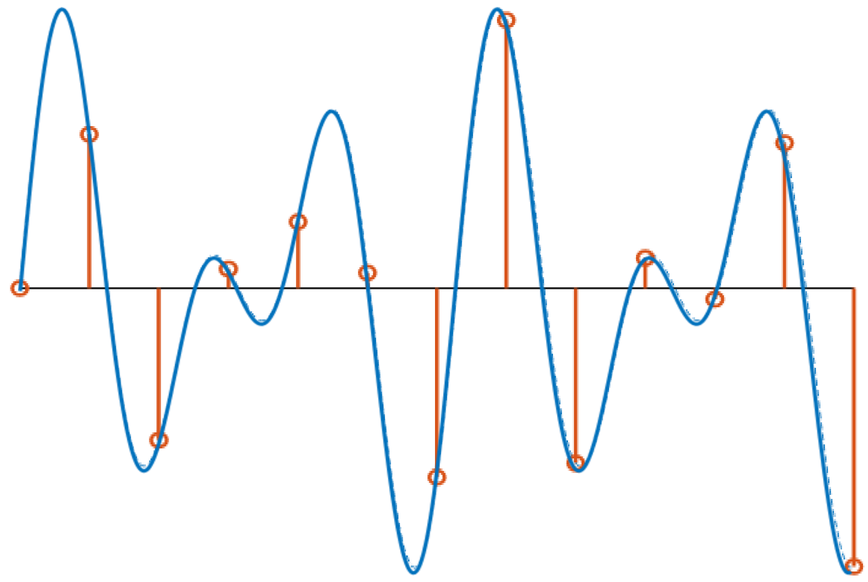
- **If $f_s \uparrow$ $(T_s \downarrow)$**
  - ☺ The sampled signal is more similar to the continuous one
  - ☹ …but this comes at an increased computational cost.



Sampling (II)

How to chose the sampling frequency?

- **If $f_s \uparrow$ ($T_s \downarrow$)**
  - ☺ The sampled signal is more similar to the continuous one
  - ☹ …but this comes at an increased computational cost.
- **If $f_s \downarrow$ ($T_s \uparrow$)**
  - ☺ The computational cost is reduced
  - ☹ … but the quality of the signal worsens.

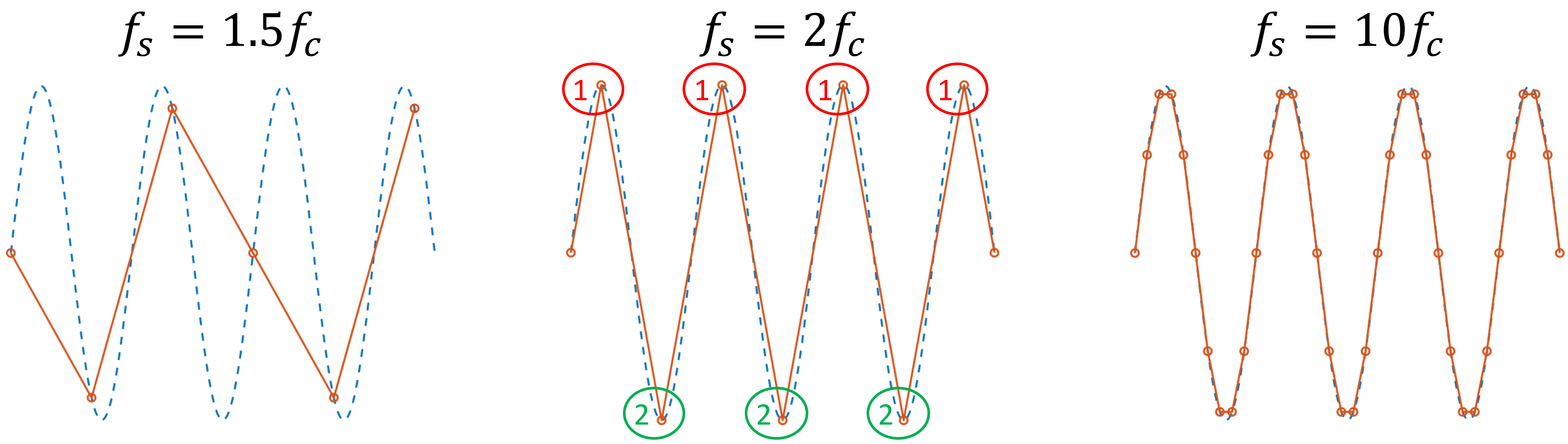


☹ If fs is too small, **it may be impossible to reconstruct** the continuous signal!

Sampling (III)

How to chose the sampling frequency?

The **Nyquist sampling theorem** states that the sample frequency must be **at least twice the highest frequency component** of the signal to avoid **aliasing** (a type of distortion) [1].



Quantization (I)

Quantization: replaces each real number with an approximation from a finite set of discrete values.

- **Resolution:** number of **bits** (binary digits) used to represent the digital code.

  With *n* bits: $\boxed{\# \, levels = 2^n}$

- **Maximum quantization error:** equals the digital resolution.

n	# of levels
8	
12	
16	

If $\boldsymbol{n}$ ↑, the number of levels ↑

and the **quantization error** ↓

Quantization (II)

Relating ADC Value to Voltage:

- the ADC assigns the maximum value $(2^n - 1)$ to the system voltage V_{cc},
- the voltage values under V_{cc} are obtained from the ratio between V_{cc} and the number of levels.

$$\frac{2^n - 1}{V_{cc}} = \frac{ADC_{output\ code}}{ADC_{input\ voltage}}$$

- The ***digitalized* voltage** can also be computed:

$$ADC_{digit.\,voltage} = \frac{V_{cc}}{2^n} ADC_{output\ code}$$

Quantization (III)

Relating ADC Value to Voltage: $ADC_{output\ code} = \frac{2^n - 1}{V_{cc}} ADC_{input\ voltage}$

<u>**Example**</u>. What is the output code if: $\boldsymbol{V_{cc} = 5V, n = 10\ bits}$ and the input voltage is $\boldsymbol{V_{in_{ADC}} = 2.18V}$**?** And what is the maximum quantization error?

$$ADC_{output\ code} = \frac{2^{10} - 1}{5} 2.18 = \boxed{\cancel{446.0280}}$$

The value is rounded to the **nearest** integer.

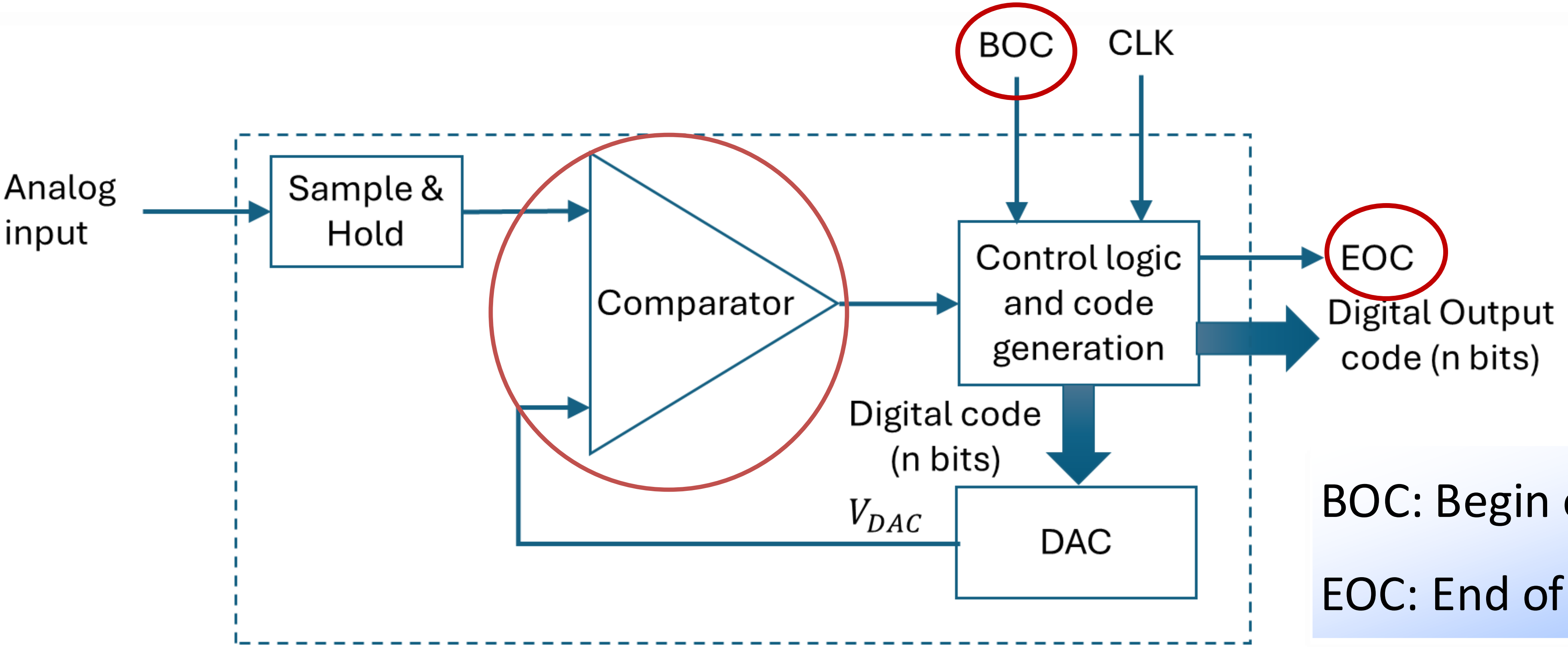
$$\boldsymbol{ADC_{output\ code} = 446_d = 01\ 1011\ 1110_b = 1BE_h}$$

$$error_{max} = \frac{5V}{2^{10}\ levels} = 4.883mV$$

Index

General Working Principle (I)

Compares the analog input with different voltage values obtained after converting a digital code to an analog signal with a Digital to Analog Converter (DAC).

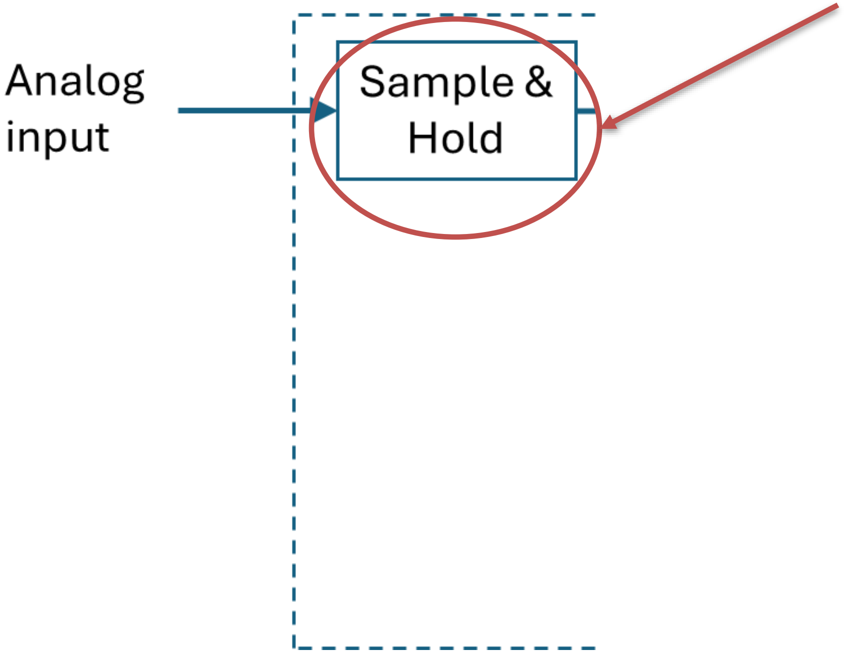


BOC: Begin of Conversion

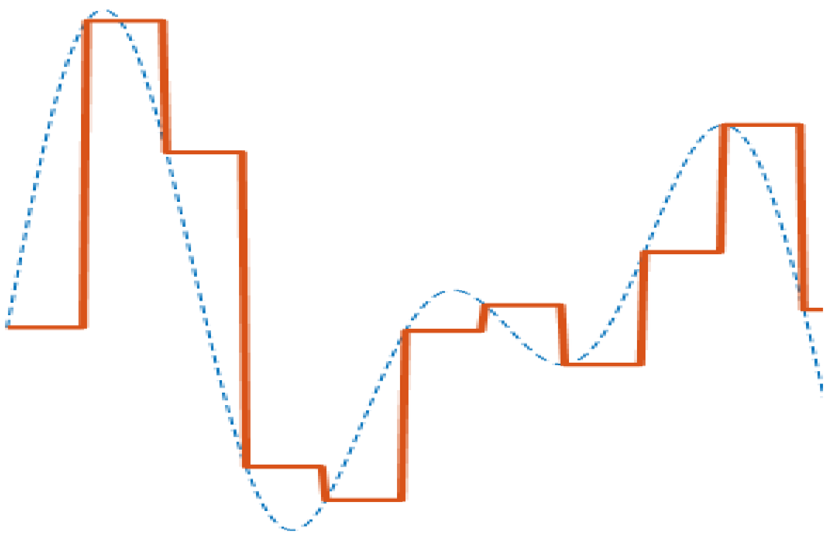
EOC: End of Conversion

General Working Principle (II)

Compares the analog input with different voltage values obtained after converting a digital code to an analog signal with a Digital to Analog Converter (DAC).



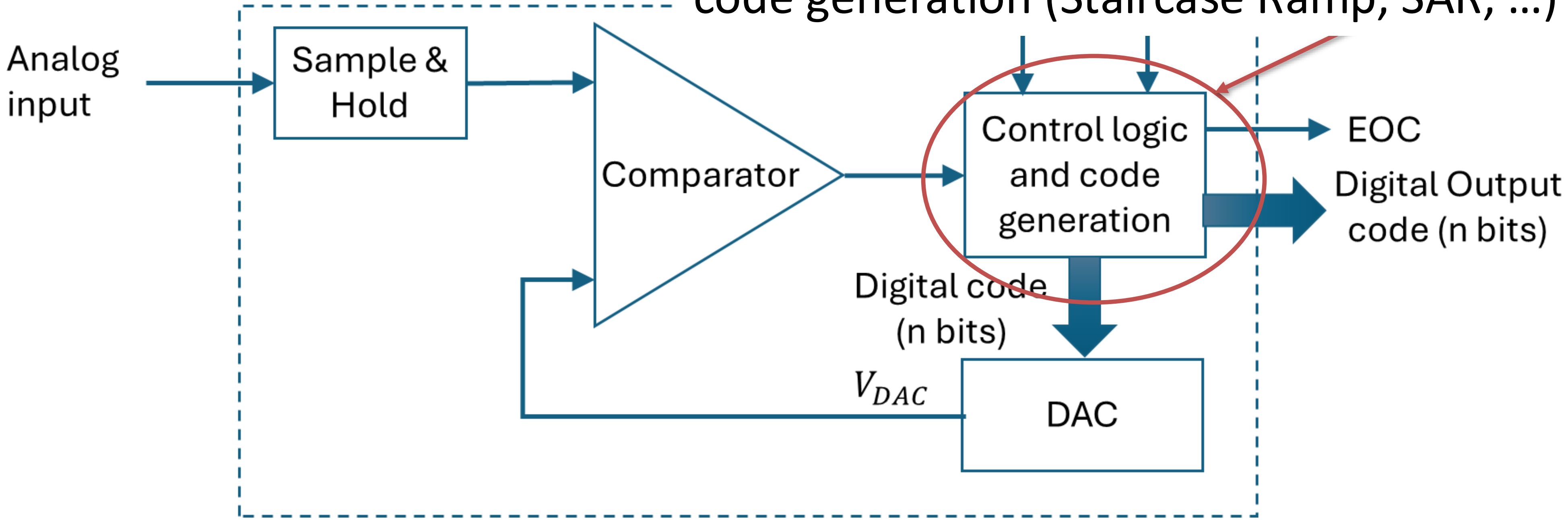
The Sample & Hold is responsible of sampling the input analog signal, maintaining its value after the following sample arrives.



General Working Principle (III)

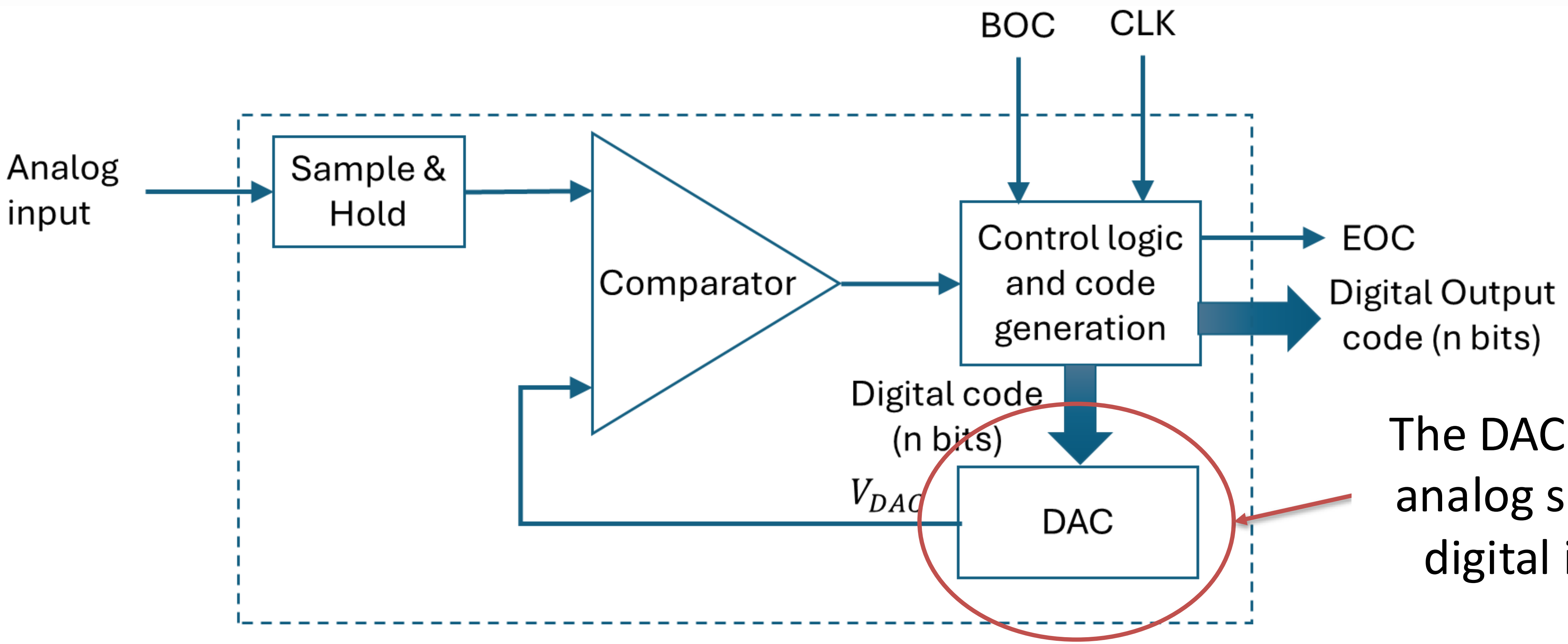
Compares the analog input with different voltage values obtained after converting a digital code to an analog signal with a Digital to Analog Converter (DAC).

There exist different alternatives for the digital code generation (Staircase Ramp, SAR, …)



General Working Principle (IV)

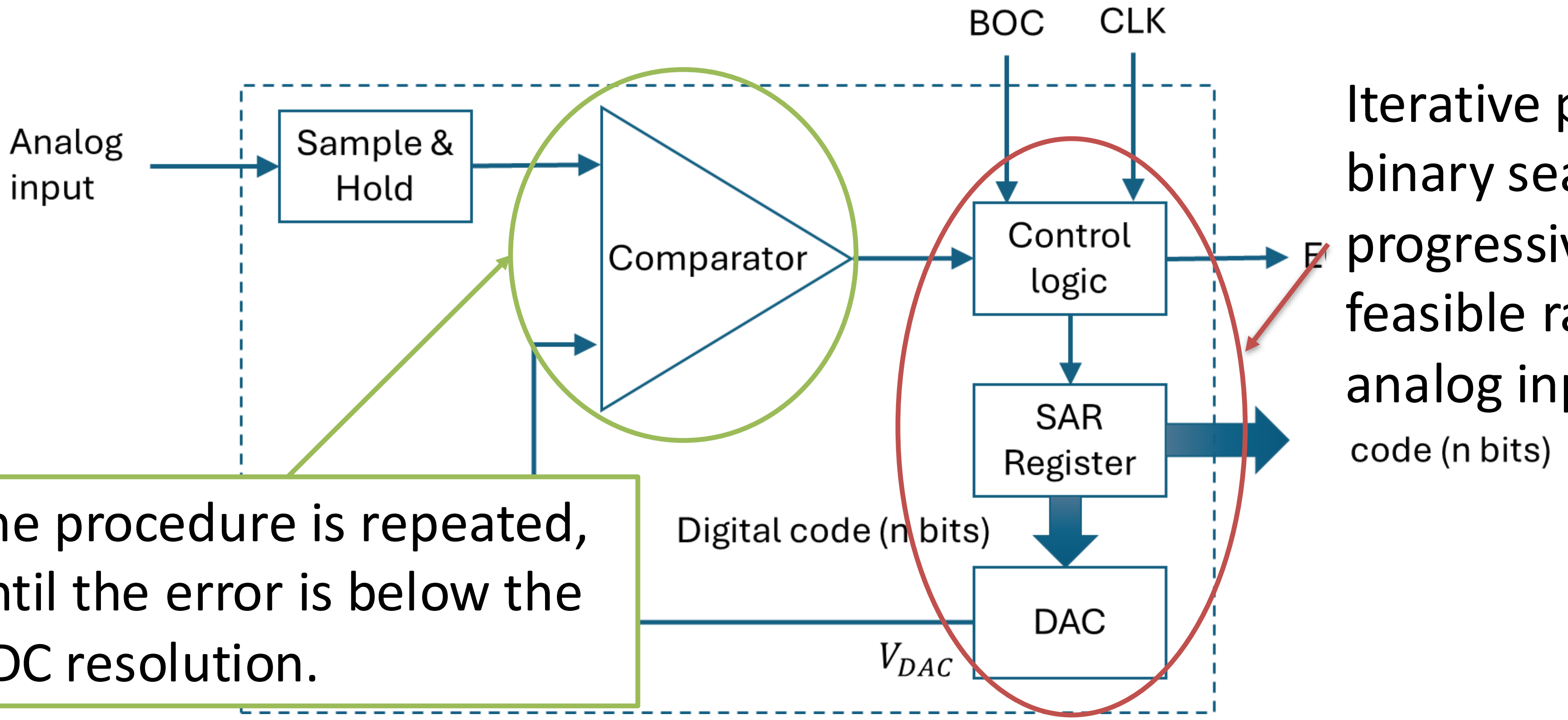
Compares the analog input with different voltage values obtained after converting a digital code to an analog signal with a Digital to Analog Converter (DAC).



The DAC provides an analog signal from a digital input code

Successive Approximation Register (SAR) ADC (I)

Compares the analog input with different voltage values obtained after converting a digital code to an analog signal with a Digital to Analog Converter (DAC).



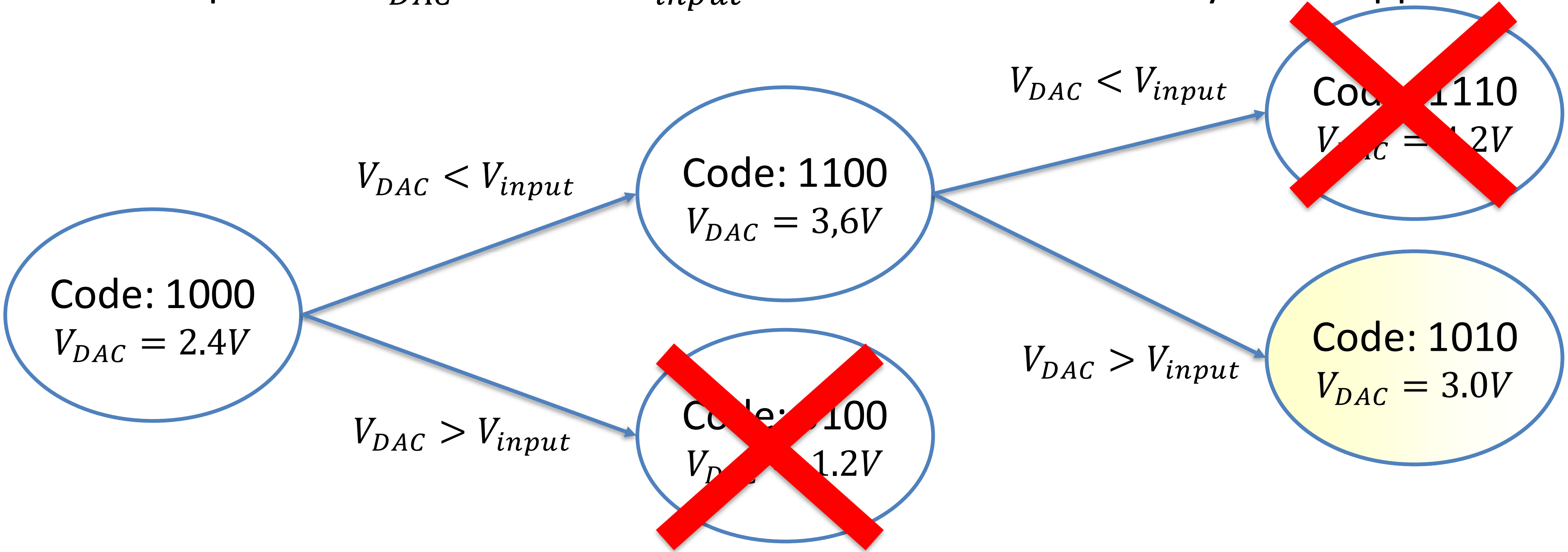
Iterative procedure based on the binary search algorithm, to progressively narrow down the feasible range within which the analog input value resides

The procedure is repeated, until the error is below the ADC resolution.

Successive Approximation Register (SAR) ADC (II)

Example: $n = 4\ bits, V_{input} = 3.0V, V_{cc} = 4.5V$ (Resolution: $\frac{4.5V}{2^4-1 levels} = 0.3\ V/level$)

1. We begin with the central value
2. Comparison: $V_{DAC} = 2.4 < V_{input} = 3.0V$ → we search only in the upper half…

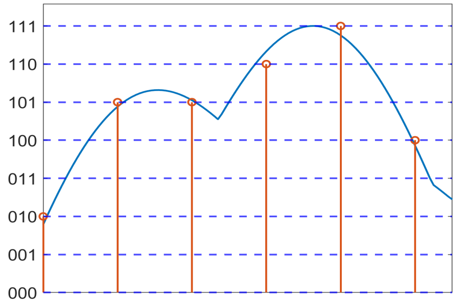


Index

Conclusions

- In a typical sensory system, analog processes and signals co-exist with digital ones.

- Analog to Digital Converters (ADCs) converts an analog voltage to a digital code representing its magnitude.



- ADCs include two different stages: **sampling** and **quantization**
  - ✓ The sampling frequency must satisfy, at least, the Nyquist theorem.
  - ✓ The number of bits determines the precision of the conversion.

- The ESP32-C3 in the GEMS prototype integrates 2 12-bit SAR ADCs, supporting a total of 6 measurement channels, with a maximum allowable voltage 3.3V, and configurable resolution and channel range.

References

[1] Austerlitz, H., **Data Acquisition Techniques Using PCs** (Second Edition), Academic Press, 2003, ISBN 9780120683772. https://doi.org/10.1016/B978-012068377-2/50004-8.

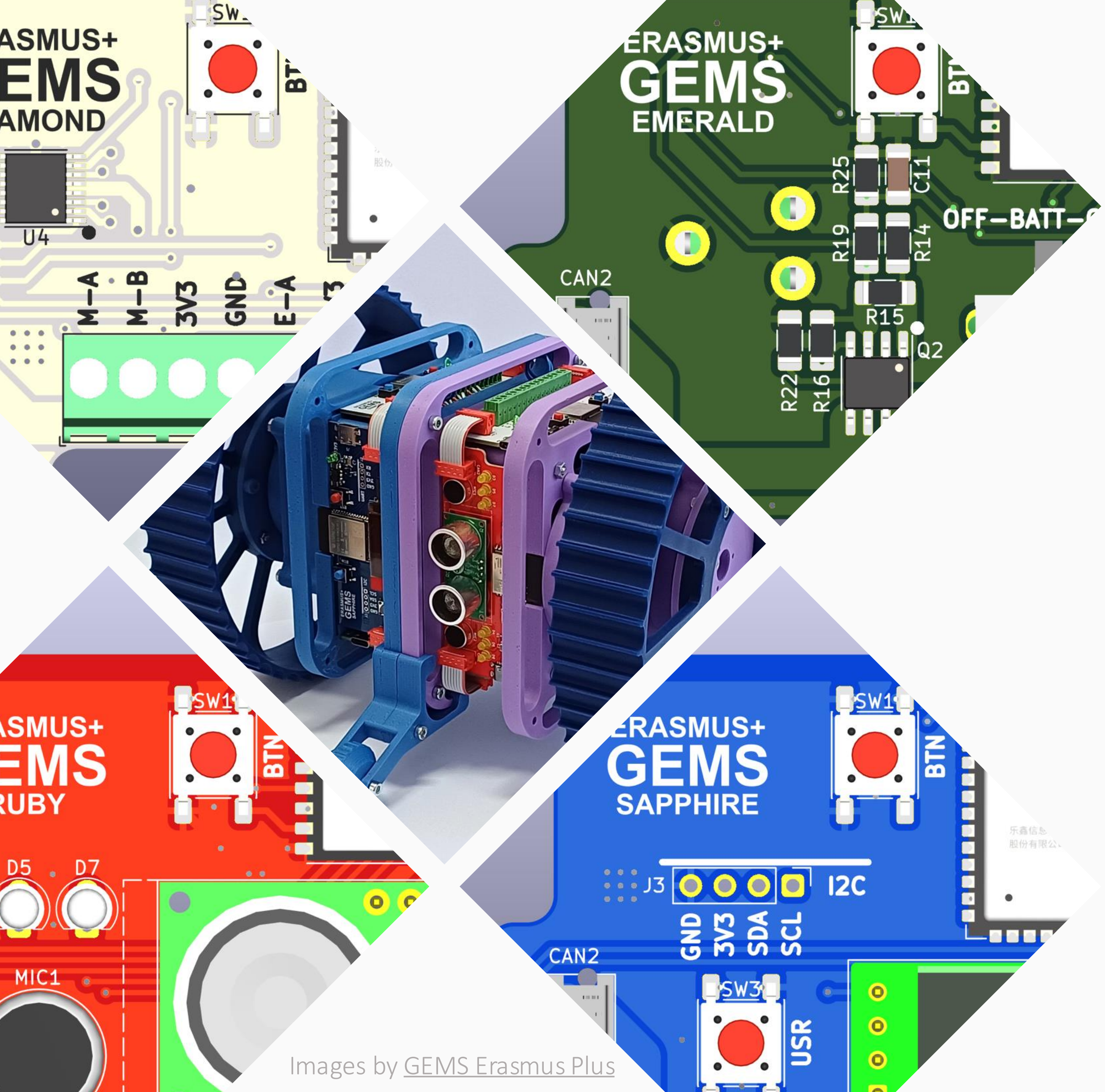
[2] Lai, E., **Practical Digital Signal Processing,** Newnes, 2003, Pages 14-49, ISBN: 9780750657983, https://doi.org/10.1016/B978-075065798-3/50002-3.

[3] Jamuna, S.; Dinesha, P.; Shashikala, K. P. A **brief review on types and design methods of ADC**. J. Eng. Res. Appl., 2018, vol. 8, no 6, p. 85-91.

[4] Espressif Systems, ESP32-C3 ADC documentation https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32c3/api-reference/peripherals/adc.html [Last accessed: 28/08/2024]

[5] Espressif Systems, **Arduino-ESP32 ADC API** https://docs.espressif.com/projects/arduino-esp32/en/latest/api/adc.html [Last accessed: 28/08/2024]

[6] Llamas, L., **How to use the ADC analog inputs in an ESP32**. https://www.luisllamas.es/en/esp32-adc/ [Last accessed: 28/08/2024].



Images by GEMS Erasmus Plus

Thank you for watching!

Co-funded by the European Union