Laboratory 4
Commenting slides

Jacopo Sini
Politecnico di Torino
Dip. Automatica e Informatica
jacopo.sini@polito.it

Example

For this laboratory it is provided an example based on a tank level.

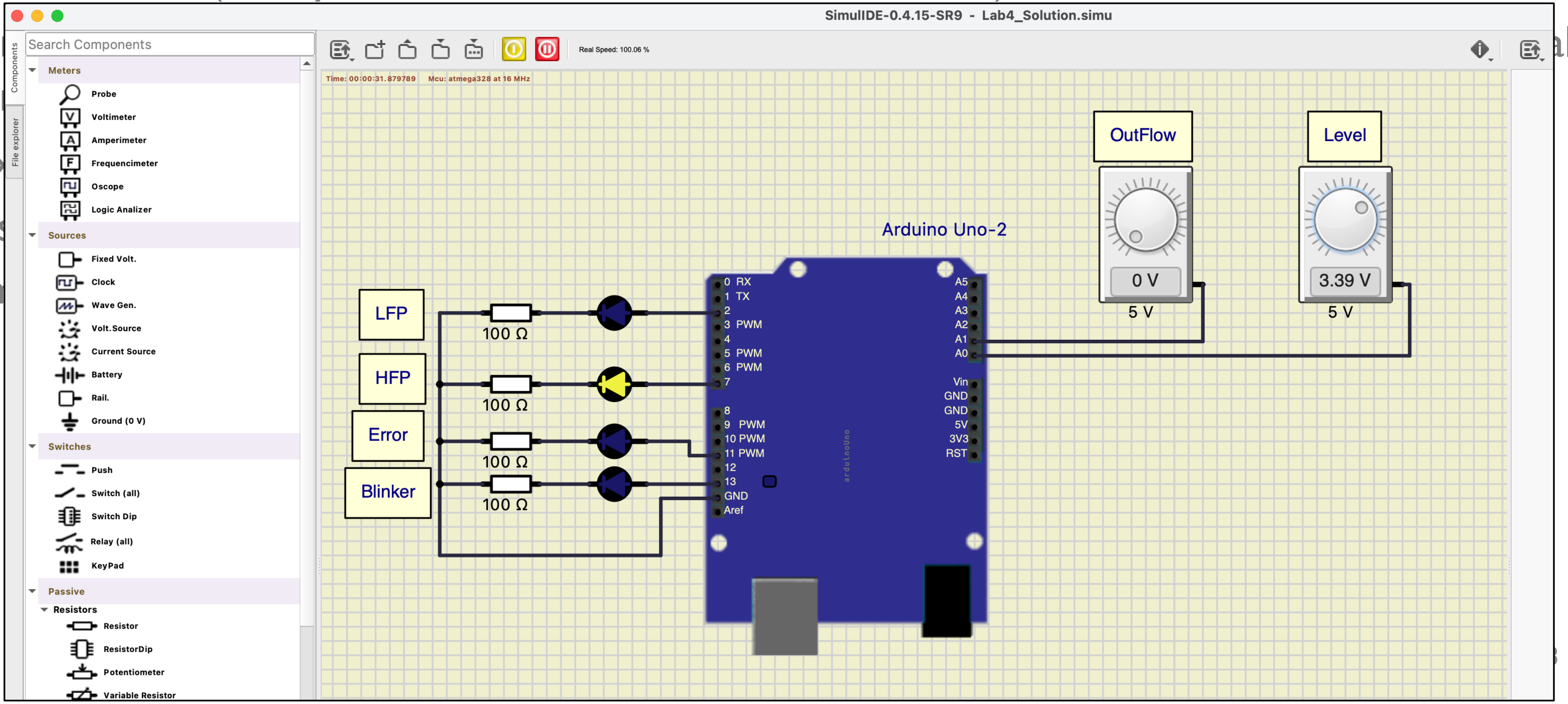
- It is composed by 6 files:
  - Lab4_Solution.sim1 (the SimulIDE file to run the controller on the Arduino)
  - controller_arduino.hex (binary file of the firmware for the Arduino)
  - controller_arduino.slx (model containing the I/O functions for the Arduino Peripherals alongside all the needed calibration and casting blocks).
  - controller.slx (the controller of the Lab 2, with some modification on error management)
  - harness.slx (same as Lab 2)
  - plant.slx (same as Lab 2)

Example

For this laboratory it is provided an example based on a tank level.

It is composed by 6 files:

- Lab4_Solution.sim1 (the SimulIDE file to run the controller on the Arduino)
- controller_arduino.hex (binary file of the firmware for the Arduino)
- controller_a ... ll the needed calib
- controller.slx
- harness.slx (s
- plant.slx (san

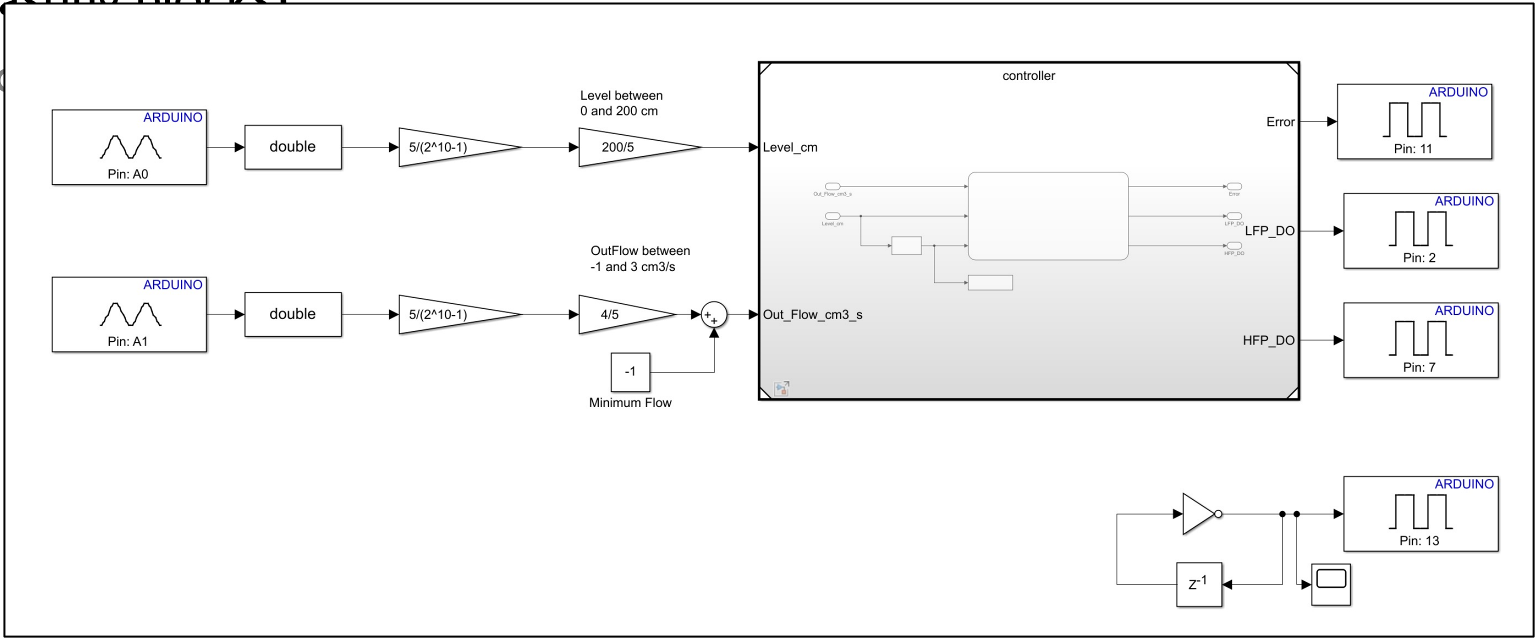


Example

For this laboratory it is provided an example based on a tank level.

It is composed by 6 files:

- Lab4_Solution.sim1 (the SimulIDE file to run the controller on the Arduino)
- controller_arduino.hex (binary file of the firmware for the Arduino)
- controller_arduino.slx (model containing the I/O functions for the Arduino Peripherals alongside all the needed calibration and casting blocks)
- controller.slx (the contr
- harness.slx (same as Lab
- plant.slx (same as Lab 2)



Hardware-In-the-Loop (HIL)

- The Plant is co-simulated with the code of the controller
  - Validate the software implementation of the controller on the target system when running in real time
  - The co-simulation is executed part on a rapid prototyping hw, part on the target system

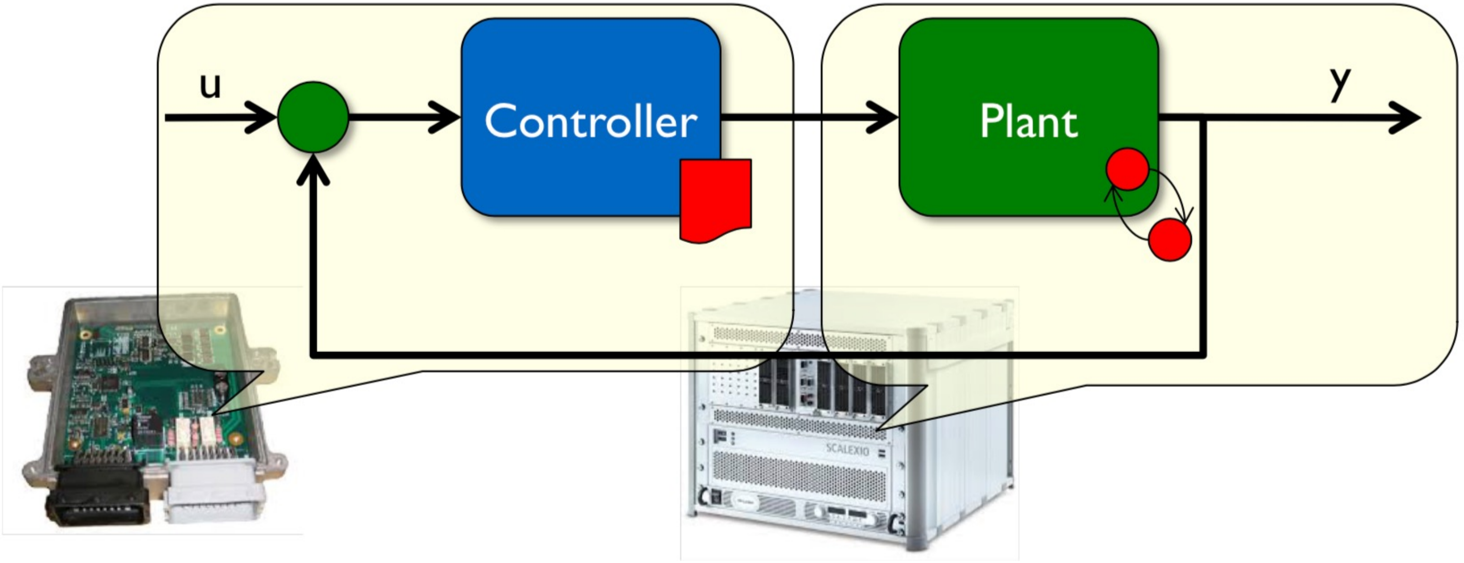


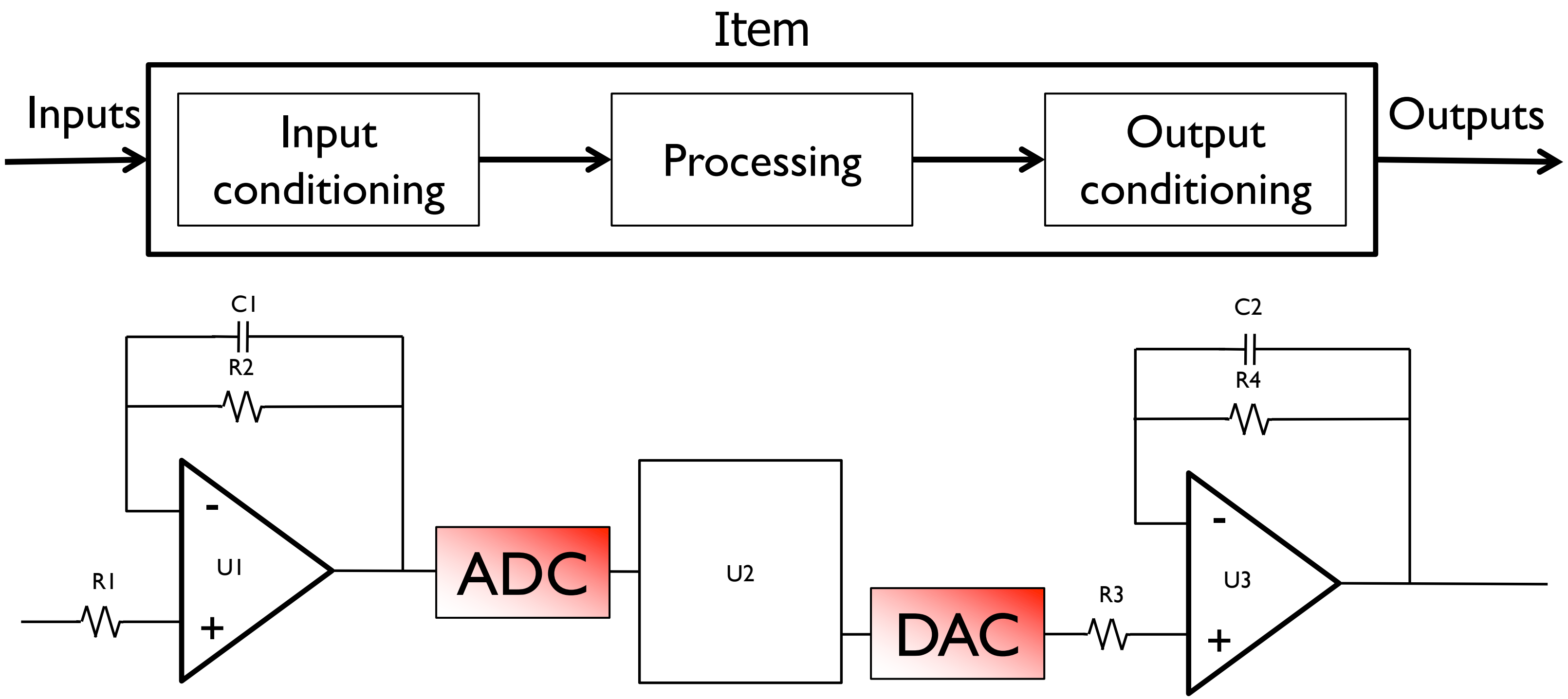
Figure from slide 47 – `05-sw-test.pdf`

Why test the software in HIL?

- The model, by definition, is an incomplete description of the real world
  - Real ADC/DAC have not infinite resolution and are not perfectly linear
  - Presence of noise added to the sensors readout
  - The real sensors are not linear and may produce artifacts
  - While in the MIL simulation environment we can image that the software execution time is negligible, in the actual system the software has to run in real time!

  - In this case we do not have real hardware, so we implement it inside a simulator capable to run the software on a virtual Arduino.

Digital control system architecture



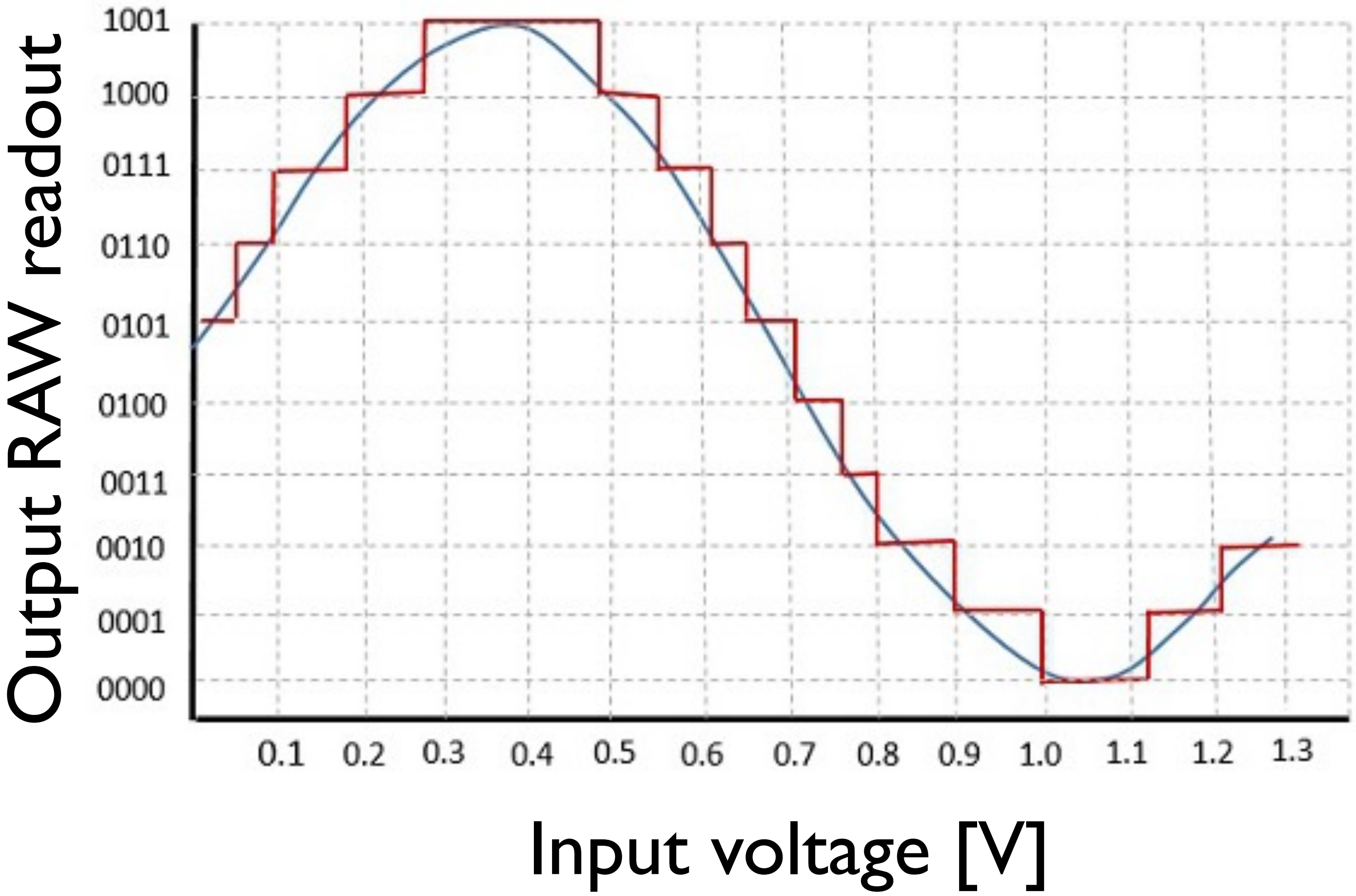
ADC and DAC models

- The Analog to Digital Converter (ADC) is an electronic device that is able to convert a continuously varying voltage into a discrete integer number
- Contrarily, the Digital to Analog Converter is a device able to convert a discrete integer number into a voltage

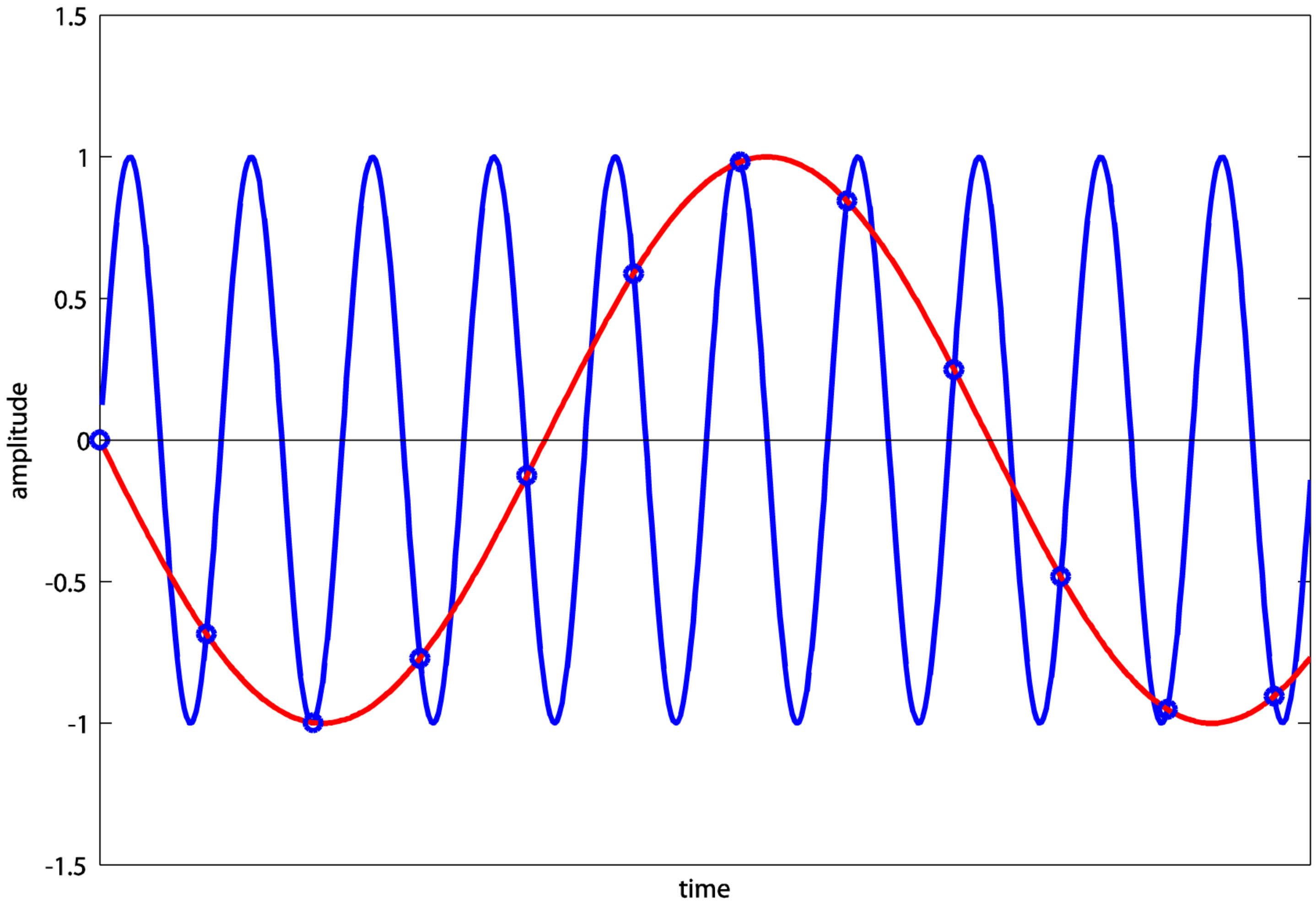
- Most important specification of these devices are:
  - Number of bits (N), also called resolution -> it leads to quantization error
  - Sample frequency (f_S) -> it leads to aliasing

Quantization error

ADC response curve

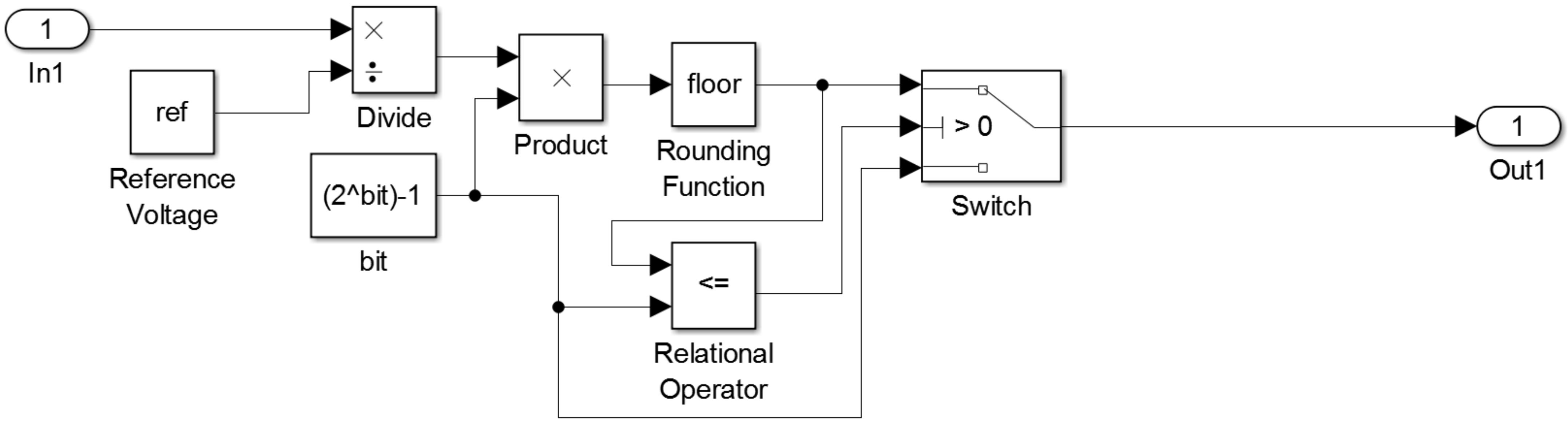


Aliasing



Ideal ADC

- An ideal ADC can be modelled by Simulink as follows:



Ideal ADC

- In the firmware, in order to obtain the readout in Volts from the RAW value measured from the ADC we have to apply the formula

$$V_{\text{in}} = RAW_{\text{in}} \cdot \frac{V_{\text{ref}}}{2^N - 1}$$

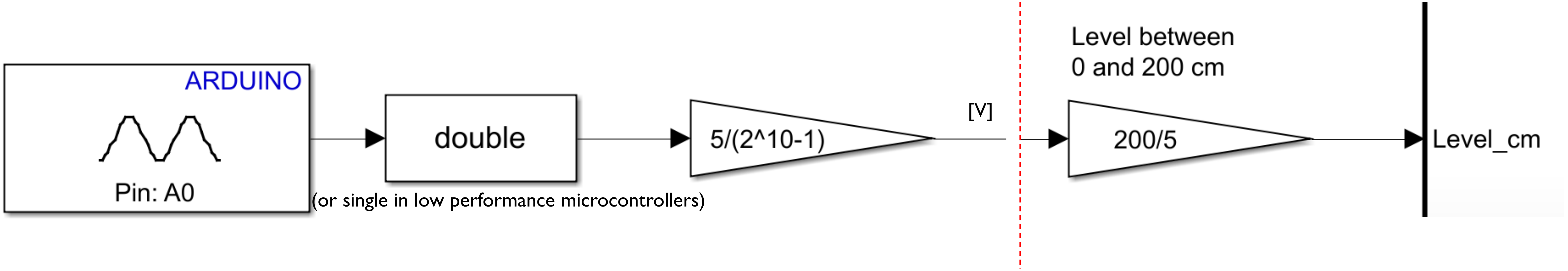
where V_{in} is the input value in Volts, RAW_{in} is the readout from the ADC, N is the number of bits and V_{ref} is the maximum voltage the device can measure.

The readout in a Nyquist-rate ADC is available after a time $T_s = 1/f_s$ (called sample time).

Due to *aliasing* phenomenon, an ADC cannot measure signals with frequency higher than $\frac{f_s}{2}$ so in real input conditioning systems is better to include a low-pass filter.

Read from a real ADC

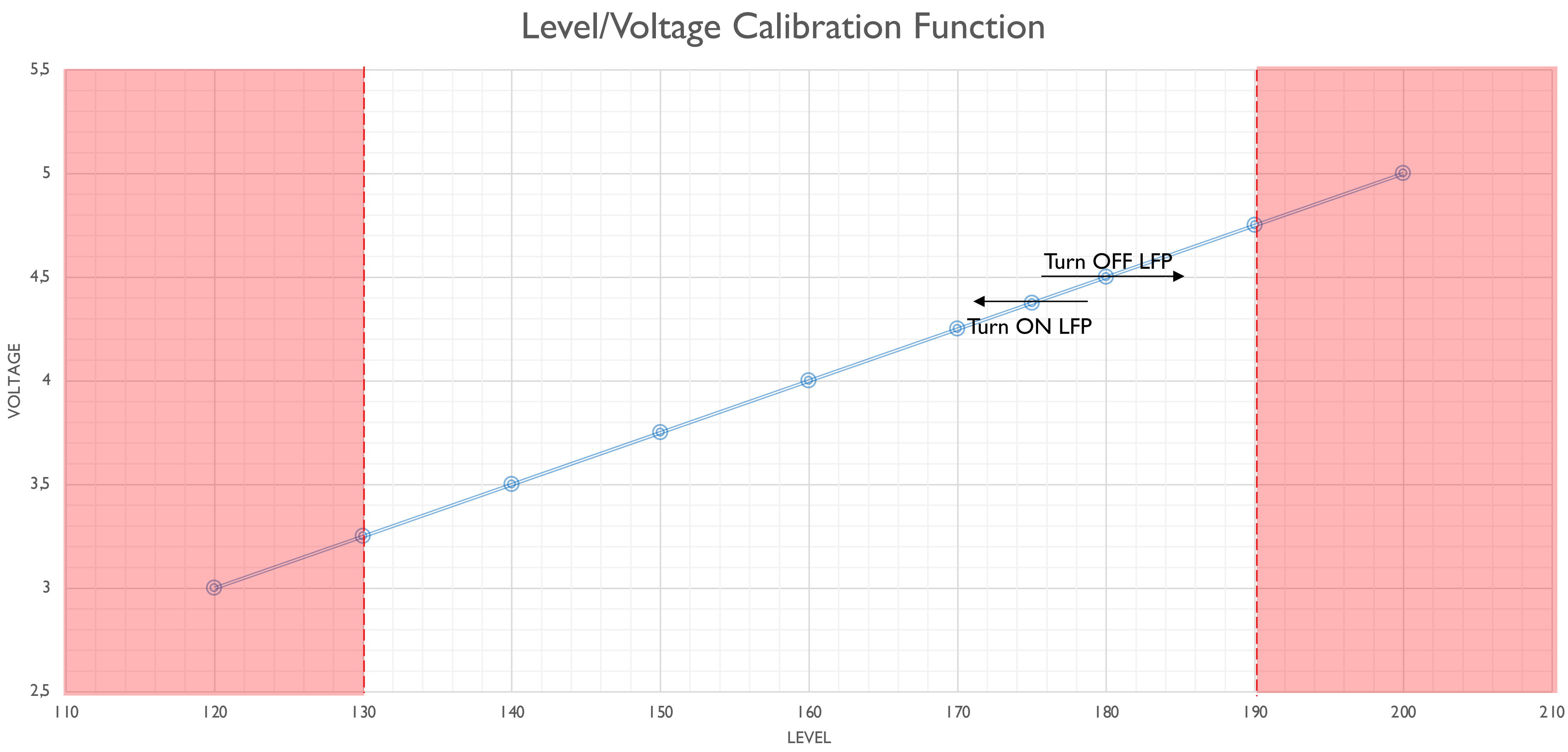
- In a Simulink model implemented by the Simulink , the formula of the previous slide is:



This separation can be useful for those cases where the ADC device driver provides a floating point voltage

In this case, since we are reading from an ADC of the Arduino Uno, we have N =10 and Vref = 4.9951 V

Calibration for Level



Error is defined outside 130 < Level < 190 range