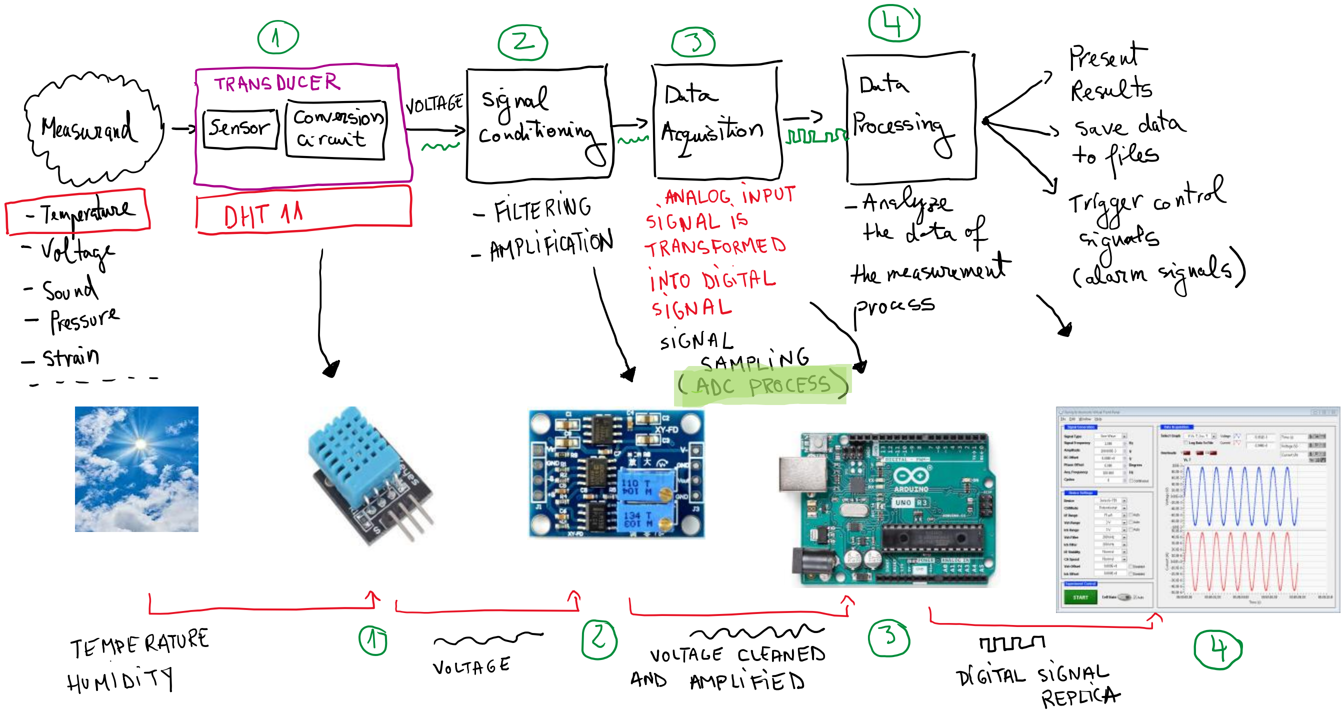
ADC PROCESS EXPLAINED

Discussion – Analog to Digital Conversion details (ADC process)



TRANSDUCER - element that converts a non-electrical measurand / quantity signal (voltage, current) that we can evaluate / measure.

Ex: microphone - converts sound -> electrical signals

speaker - converts electrical signals -> sound

SENSOR - can be used alone or included in the transducer. The sensor comes in direct contact with the measured quantity. More sensors can be included in a single transducer.

CONVERSION CIRCUIT - implements a first adjustment of the signal generated by the sensor.

Ex: current to voltage conversion, voltage division, resistance to voltage conversion.

TRANSDUCER NGM2611 - E13 and it includes the sensor TGS 2611 (used for CH4 - methane concentration measurements)

Sensor examples: thermocouple (used for temperature measurements -> generates a small current)

thermistor (— " — -> it is a variable resistance that changes with temperature)

photoresistor (used to detect illumination)

Sensors

-> [passive] (do not require additional power signals to work)

- thermocouple, piezo sensor for vibration measurements

-> [active] (require power supply signals to work) DHT11 is an active sensor because we power it at 5V.

- capacitive sensors, inductive sensors, bridge type sensors

SIGNAL CONDITIONING

→ Amplification – increase the magnitude of generated signals comming from the transducer. In this way signals of low amplitude become more increased in level, while external noise and perturbations will not affect deeply our measurement process.

– perform different mathematical operations on our input signals (addition, inversion, substraction, integration etc.)

Operational Amplifier – OA – is the main component that we will discuss.
– circuit –

→ Filtering – is used to remove unwanted frequencies from the content of a signal ⇒ we remove noise/interference or we can say that we clean up the signals.

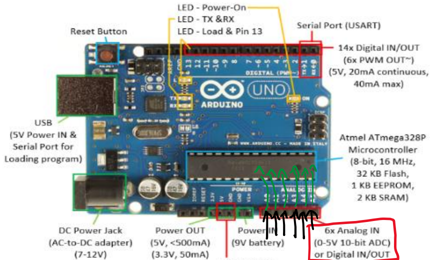
ADC – it is the process in which the analog form of a signal is converted to the digital form. The digital form is processed by computer software.

Ex: ARDUINO Board – it receives signals from a sensor and then it converts the signals to their digital form and uses a programmed micro-controller to perform signal processing. Finally it sends the results to the computer by serial connection.

Electrical signals from sensors/transducers (thermocouples, resistive-capacitive-inductive elements used for temperature measurement, strain gauges, pressure gauges) is in analogue form. Microcontrollers require a *digital form* of the signal data. This digital form is used to perform mathematical operations, implementation of different calculus algorithms etc. Digital signals are easily transmitted and stored (Example: https://pdf1.alldatasheet.com/datasheet-pdf/view/28845/TI/TLC0820ACN.html).

An analogic signal is continuously variable, it has an infinite number of values defined over a period and evolves (more or less) smoothly over a range of values. This signal is an analogue version of the measurand it represents. When a microprocessor receives this input signal, it must be initially converted into a digital form. A digital signal is a sequence of pulses, often just on/off signals. The value of the measurand in a digital signal is represented by discrete samples, finite over a period.

Analogue-to-digital conversion involves several stages. First samples of the analogue signal are recorded at fixed time intervals (supplied by a clock circuit). The result is a series of pulses with heights which vary in accord with the variation of the analogue signal. Each sampled value being held until the next pulse. This ADC circuit converts each sample into a sequence of pulses representing the value. For example, the first sample value might be represented by 101, the next sample by 011 and so on. The 1 represents an 'high' value signal, the 0 a 'low' value signal. ADC process involves a **sample and hold** circuit followed by an **ADC converter** circuit. The sample rate (or sample interval) is performed at fixed intervals of time. Errors can arise if the input is sampled below a rate, called the *Nyquist rate* (recall the discussion on signal sampling and data acquisition).

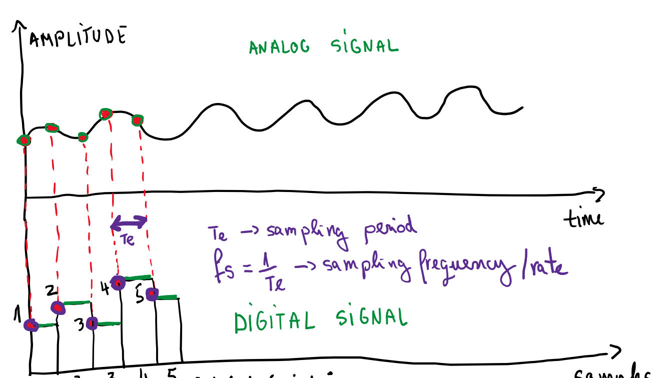


SPECIFICATIONS

6x ANALOG IN – there are 6 channels that can receive analog voltage signals from sensors.

0 – 5V – the input RANGE. This means that each of the analog signals can take values from 0V to 5V.

10 bit ADC – WORD LENGTH, it gives the performance of the conversion process.



COLLECTION OF SAMPLES FROM THE ORIGINAL ANALOG SIGNAL.

EACH ANALOG VALUE IS ASSOCIATED WITH A DIGITAL CODE.

Analog signal → ADC CIRCUIT [SAMPLE AND HOLD → CONVERSION CIRCUIT] → DIGITAL CODE FOR A RECORDED VOLTAGE

voltage representing the song of a bird 1.619 V

$\boxed{331} = \underbrace{0101001011}_{10\ bits} = $

0	1	0	1	0	0	1	0	1	1
2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	256	0	64	0	0	8	0	2	1

$256 + 64 + 8 + 2 + 1 = \boxed{331}$

MSB – MOST SIGNIFICANT BIT

LSB – LEAST SIGNIFICANT BIT

WORD LENGTH → 10 bits

ADC conversion process

- → RANGE
  - how much voltage can we apply at the analog inputs? (ARDUINO accepts 5V)
- → RATE
  - how fast can our ADC circuit make the conversion from analog to digital? This is related to the sampling period.
- → RESOLUTION
  - given by the bit word (EX. ARDUINO has ADC on 10 bits. Other devices can have ADC on 16 bits, 24 bits).
  - More bits means better resolution but also a more expensive ADC.

An ADC circuit has an input RANGE of 8V and a word length of 3 bits. Present the digital values associated with the input range.

n = 3 (3 bits of word length) ⇒ $2^3 = \boxed{8 \text{ possible combinations.}}$ → MUST COVER THE RANGE

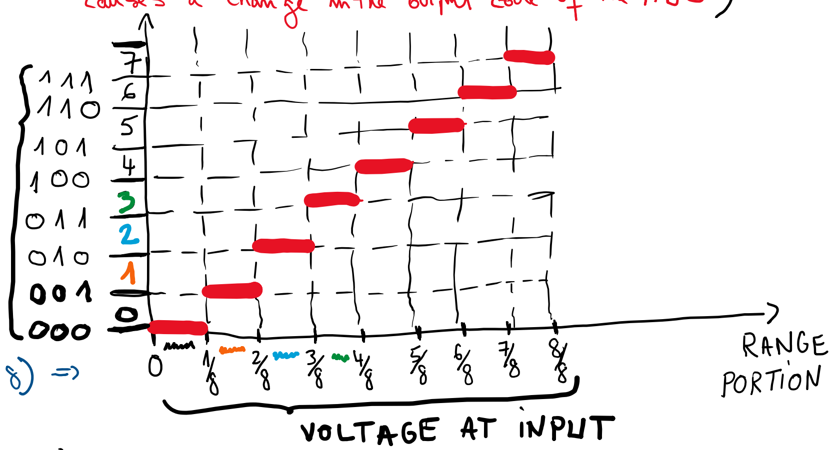
$$\boxed{\Delta = \frac{RANGE}{2^n} = \frac{8V}{2^3} = 1V}$$

(resolution is 1V. The smallest change in the analog signal at the input which causes a change in the output code of the ADC)

- if at input i get a value $[0 \div 1)$V ⇒ output we have value 000 or 0;
- if at the input we get a value $[1 \div 2)$ ⇒ output we have 001 or 1;
- if at he input we get a value $[2 \div 3)$ ⇒ output we have 010 or 2;

– – – – – – – – –

- if at the input we get a value of $[7 \div 8)$ ⇒ output we have 111, the value is 7.



$\boxed{1.34}$ → ADC (8 MAXIMAL, 0 MINIMAL) → [0:1) $\boxed{[1:2)}$

000
$\boxed{001}$ 1
010
011
100
101
110
111

COVER A RANGE FROM 0V TO 8V

8 INTERVALS

CODES FROM 0 TO 7

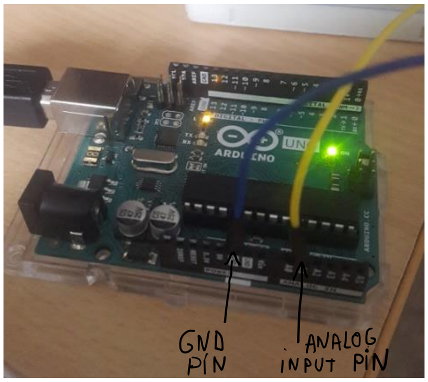
THE IMPORTANCE OF RESOLUTION FOR ADC

EXAMPLE → **5V** INPUT RANGE

- 3 BIT ADC ⇒ $\Delta = \frac{5V}{2^3} = 0.625V$ or $\boxed{625mV}$ NOT SO GOOD
- $\boxed{10\ BIT\ ADC}$ ⇒ $\Delta = \frac{5V}{2^{10}} = \frac{5V}{1024} \cong \boxed{4.88\ mV}$ OK
- 16 BIT ADC ⇒ $\Delta = \frac{5V}{2^{16}} = \frac{5V}{65536} \cong \boxed{76.3\mu V}$ BEST

THE RANGE IS DIVIDED IN 1024 INTERVALS GIVING CODES FROM 0 TO 1023

SMALLER RESOLUTION ALLOWS US TO EXPRESS THE SAME RANGE WITH BETTER ACCURACY. OR MORE EXACT.



ARDUINO UNO BOARD

TO THE ANALOG INPUT PIN I GET THE ANALOG VOLTAGE $X = 1.616\,V$

$0 - 5V \Rightarrow \Delta = \frac{5V}{2^{10}} = \frac{5V}{1024} \cong 4.88\,mV$ (RANGE)

$V_{in} = \frac{\text{DIGITAL CODE} + 0.5}{\text{NUMBER OF INTERVALS}} * \text{RANGE}$

WORD LENGTH – 10 bits

RESOLUTION – 4.88 mV

```
RUNNING...
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->328  Binary value->101001000  Voltage value->1.604
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->0  Binary value->0  Voltage value->0.002
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->330  Binary value->101001010  Voltage value->1.614
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
Digital code->329  Binary value->101001001  Voltage value->1.609
PAUSED...
```

$\text{INPUT VOLTAGE} = \frac{329 + 0.5}{2^{10}} * 5V = \frac{329.5}{1024} * 5V \cong 1.609\,V$

$\frac{330 + 0.5}{1024} * 5V = \cong 1.614\,V$

VOLTAGE OFF — 0V at INPUT

$329 \rightarrow 1.609\,V$

$330 \rightarrow 1.614\,V$

$1.614 - 1.609 \approx 5\,mV$ THE RESOLUTION

OFF — 0V at INPUT

EACH CHANGE OF DECIMAL VALUE HAPPENS WITH THE STEP OF THE RESOLUTION !!!

The output of a sensor changes with 0.5mV for each degree °C of temperature change. The output range of the sensor is 0°C - 200°C. We would like to use an ADC to convert this voltage signal and present the temperature on a display with a resolution of 0.5°C (22.0°C, 22.05°C, 23.00°C, 23.05°C etc.). What are the characteristics of an ADC that can be used for this application?

HOW LARGE SHOULD BE THE ADC INPUT RANGE?

ADC INPUT RANGE ≈ 100 mV

$\underbrace{0.5\,mV/°C}_{\text{SENSOR RESOLUTION}} * \underbrace{200\,°C}_{\text{SENSOR TEMPERATURE RANGE}} = \boxed{100\,mV}$ ← ADC INPUT RANGE

HOW DO I NEED TO PRESENT MY VALUES?

0.5 mV → 1 °C of change at the sensor output

0.25 mV → 0.5 °C of change on my display

$\Delta = \frac{0.5\,mV}{2} = \boxed{0.25\,mV}$ ← maximum required ADC resolution to present data with 0.5°C of resolution!

HOW MANY BITS DOES MY ADC REQUIRE? WORD LENGTH?

$\Delta = 0.25\,mV = \frac{RANGE}{2^n} = \frac{100\,mV}{2^n} \Rightarrow$

$\Rightarrow 2^n = \frac{100\,mV}{0.25\,mV} = 400 \Rightarrow \log_2(400) \cong \boxed{8.64}$

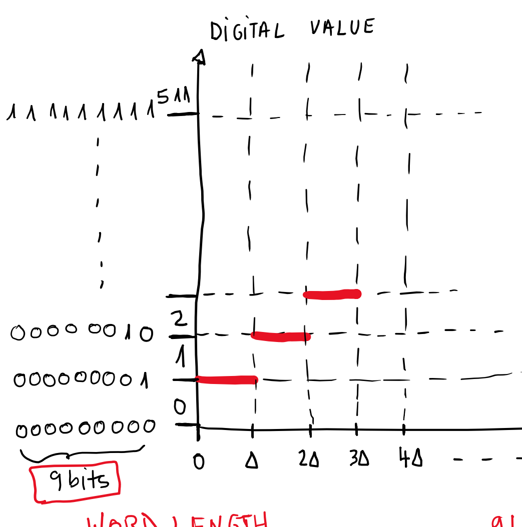
$n = 8.64$ rounded up $\Rightarrow \boxed{n = 9\text{ bits}}$ or higher

$\Delta = \frac{100\,mV}{2^9} = \frac{100\,mV}{512} \cong \boxed{0.195\,mV}$ required ✓

↳ better than 0.25 mV

ADC RESOLUTION (0.195 mV) MUST BE SMALLER THAN SENSOR OUTPUT RESOLUTION (0.5 mV/°C)!

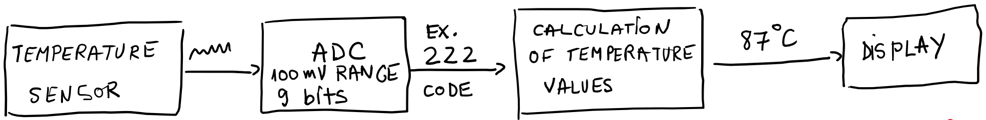
$0.195\,mV < 0.25\,mV$



INPUT	CODE
[0 – 0.195) mV	0
[0.195 – 0.39) mV	1
[0.39 – 0.78) mV	2
⋮	⋮
[99.45 – 99.645) mV	511

ADC INPUT RANGE
MAX. 100 mV

9 bits → WE DIVIDE OUR RANGE OF 100 mV in 512 INTERVALS, EACH INTERVAL OF 195 mV. ⇒ 512 * 0.195 V = 99.84 mV + ERROR COMPENSATION



$$Vin = \frac{222 + 0.5}{512} \cdot 100mV = 0.04345 = 43.45mV \cong 86.9\,°C = \underline{\underline{87.00\,°C}}$$

→ CONVERSION ERROR COMPENSATION

$$\underline{511} \rightarrow Vin = \frac{511 + 0.5}{512} * 100\,mV = 99.9\,mV \Rightarrow 199.8\,°C \Rightarrow 200\,°C$$

The maximum sensor output range

127 → - - - - - - → 50° C

49.8 °C => 50°C because our screen only prints values with 0.5°C resolution.

Rate : Speed of the conversion process. For most ADCs it is < 10µs for an 8 bit word length.
=> fs (sampling frequency) $\cong \frac{1}{10\mu s} \cong$ 100 kHz (minimal) ← more than sufficient for all our applications.

Resolution:

→ ADC resolution is defined by number of bits and input range. It is the smallest variation of the analog input signal that modifies the digital code at the ADC output.

→ If we talk about the resolution of a measurement device (Example DMM - digital multimeter) we refer to the minimal modification that we can see on the screen of that device.

EX: ADC gives code 222 => temperature we see on the screen is 87.0°C
ADC gives code 221 => ———— " ———— 86.5°C (even if from the calculations we get 86.52°C)
ADC gives code 223 => ———— " ———— 87.0°C (even if from calculations we get 87.3°C)

ADC resolution may be better than display resolution.

→ Sensor output resolution (0.5 mV/°C) must be always larger than ADC resolution.

Δ = 0.195 mV $\cong$ 0.4 °C ← better than sensor resolution