**Analogue and Digital Signals**

An analog signal is a kind of signal that is continuously variable, as opposed to having a limited number of steps along its range (called [digital](https://www.allaboutcircuits.com/video-lectures/digital-concepts-terms/)). A well-known example of analog vs. digital is that of clocks: analog being the type with pointers that slowly rotate around a circular scale, and digital being the type with decimal number displays. The analog clock has no physical limit to how finely it can display the time, as its “hands” move in a smooth, pauseless fashion. The digital clock, on the other hand, cannot convey any unit of time smaller than what its display will allow for.

Analogue Signals for your project

Our world is analogue, with infinite values of for example sounds, or colour values.

With an Arduino board, you use the analogue pins to read spans of values from sensors, or the Pulse Width Modulation (PWM) pins marked with a ~(tilde) symbol to control the current flow to a connected component.

On a Raspberry Pi, there are no pure analogue input pins, and so two alternative options are made the scene. The 1st option is to use a specially designed A/D circuit chip (a Raspberry Pi “hat”). The 2nd option is to use an Arduino as an analogue sensor board, in conjunction with the Snap4Arduino software package. When need for analogue output (indeed, pseudo-analogue output), the PWM option is a feasible solution using the Raspberry Pi unit as well, in a similar way as in case of Arduino. More specifically special libraries add PWM functionality to Raspberry Pi GPIO pins. Last but not least, is the option to use an Arduino as a pseudo-analogue pin board, assisted by the Snap4Arduino software package.

**Digital signal for your project**

Digital signals have a set span of values.

When you use an Arduino, digital signals are either On or Off - ‘HIGH’ or ‘LOW’ - 0V or 5V/3.3V. On an Arduino board, there are 14 digital pins that can be used to write or read to and from electronic components.

On a Raspberry Pi, there is a considerable number of digital inputs/outputs on its 40-pin connector that comply with 3.3V logic. More specifically, 3.3V logic levels means that the Raspberry Pi will interpret anything very close to zero volts as a logic ‘0’ and anything higher than around 2V as a logic ‘1’. Applying higher voltage levels on these pins could damage the board. These pins can be programmed to intercept signals from digital input devices (sensors) or to control output devices (actuators).

A very interesting alternative is to use an Arduino board (plugged in a USB port of the Raspberry Pi unit) as an extension board for the Raspberry Pi that provides extra digital pins. Snap4Arduino tool offers the necessary visual programming command options.

When you create interactive projects, you need to take both the analogue and the digital world into account.

