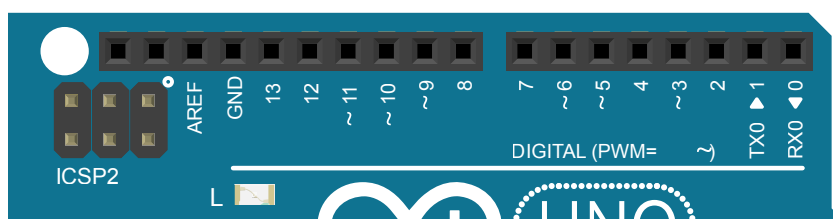


DIGITAL INPUTS

How the Arduino board detects a 5 V signal

A digital input is a voltmeter that can only read two values LOW (\sim GND, 0 V), and the value HIGH (\sim 5 V).



The Arduino board UNO has 14 digital ports (called DIGITAL on the board), numbered from 0 to 13. Each of the ports can be declared as being an input in the microcontroller program. The ports 0 and 1 are reserved for serial communication, you should not use them. The GND port is the ground of the board (0 V).

When a digital port is declared as input, the state of the port can be read by the Arduino board (HIGH or LOW) and that value can be used in the program to launch this or that action.

To determine the state of a digital input (HIGH or LOW), the board measures the value of the voltage applied to that port. The voltages that are inferior to around 3 V will be read as LOW, the voltages that are superior will be read as HIGH.



You must avoid intermediary voltages, which risk giving random results.

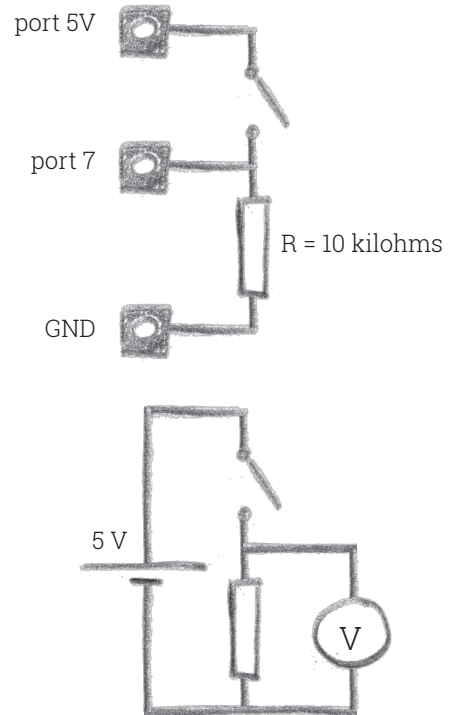
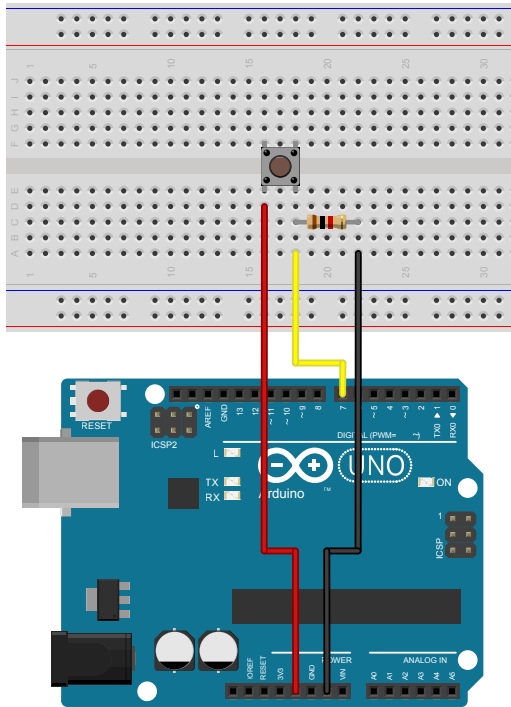
A non-connected digital entry will float (with no affected voltage its state will remain undetermined). The reading of this port can give almost any result.

A voltage superior to 5.5 V can destroy the Arduino board.

KNOWLEDGE – DIGITAL INPUTS

Usage examples

Here is a simple setup to read the state of a switch (open or closed) as well as two equivalent electrical diagrams: the port 7 is equivalent to a voltmeter that can only measure 0 volts or 5 volts.



When the switch is open, port 7 and the ground are connected and are at the same electrical potential because no current is circulating: the board measures 0 V (LOW). When the switch is closed, the potential of the input 7 rises to 5 V (HIGH) and a current of 0.5 mA is circulating between the 5V port and the GND port, which poses no problem.

The resistor ensures that the measured voltage on port 7 is indeed that of the GND port when the switch is open. Without this resistance (called pull-down), the port 7 would be floating and its measurement would be indeterminate.



Be careful: if port 7 is declared by error to be the output instead of input in the program, there is a risk of a short circuit when the switch is closed

Programming

```
pinMode(7, INPUT) ;           // defines digital port 7 as input
                               // put this instruction into the setup()

buttonstate = digitalRead(7) ; // returns the state of input 7 (true or false,
                               // 0 or 1, HIGH or LOW) and attributes it to the buttonState variable
                               // (used here as an example).
```