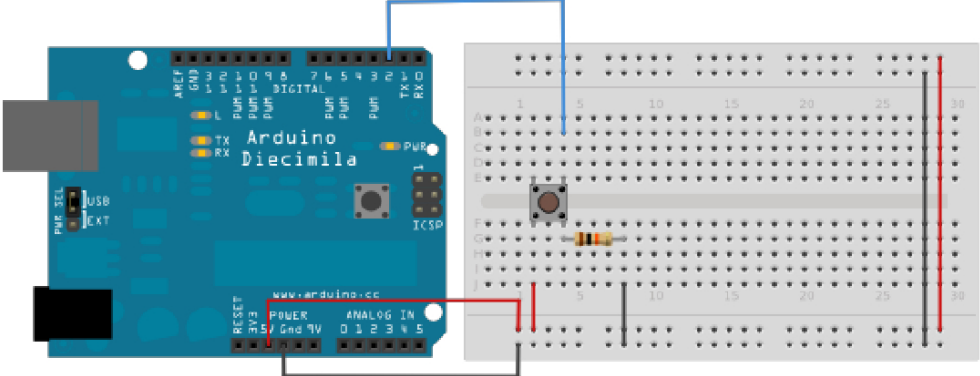
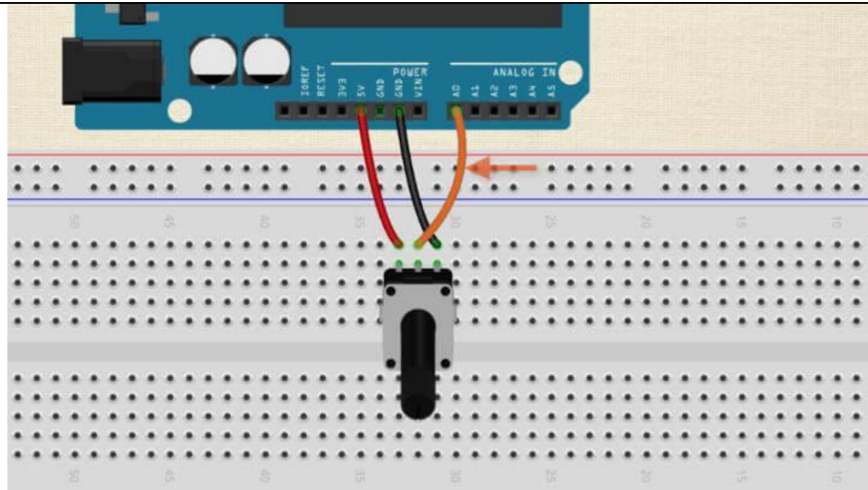
	<b>School of Science and Technology</b> <b>Computer Science</b> <b>Embedded Systems Architecture</b> <i>Prof. Lorenzo Morresi</i>	MSc in Computer Science (LM-18) A.A. 2019-2020
Project Title	<b>SIGNAL READING AND ADC</b>	Sheet ARDUINO n_3
<u>Description</u>	This activity concern the reading of both analogical and digital signals by ARDUINO UNO board and the analog to digital conversion by ADC.	
Necessary materials	ARDUINO UNO board 1 push button 1 potenziometro or trimmer - 10 KΩ 1 resistor - 10 KΩ 1 red LED 1 resistor - 220 Ω	
<u>Sketch</u>  <b>DIGITAL SIGNAL READING</b>	<pre> /* DigitalReadSerial Reads a digital input on pin 2, prints the result to the serial monitor This example code is in the public domain. */  void setup() {   Serial.begin(9600);   pinMode(2, INPUT); }  void loop() {   int sensorValue = digitalRead(2);   Serial.println(sensorValue, DEC); } </pre>	
<u>Pictorial /Schematic</u>		
<u>Note</u>	<p>In normal condition, button not pressed, we will see a sequence of 0 scroll on the serial monitor, when the button is pressed we will see the printed value varying from 0 to 1, releasing the button the value varies from 1 to 0.</p> <p>The configuration shown is called pull-down. Follow the attached sheet and create a pull-up configuration. Which kind on differences you have in the digital signal reading?</p>	
<u>Sketch</u>  <b>A/D CONVERSION</b>	<pre> //Arduino - Analog to Digital Conversion  void setup() {   Serial.begin(9600);    // apro comunicazione seriale } </pre>	

```

void loop() {
int adcValue = analogRead(A0); // leggo valore su A0
Serial.print("Valore adc: "); // invio testo al monitor seriale
Serial.print(adcValue);      // invio valore al monitor seriale
}

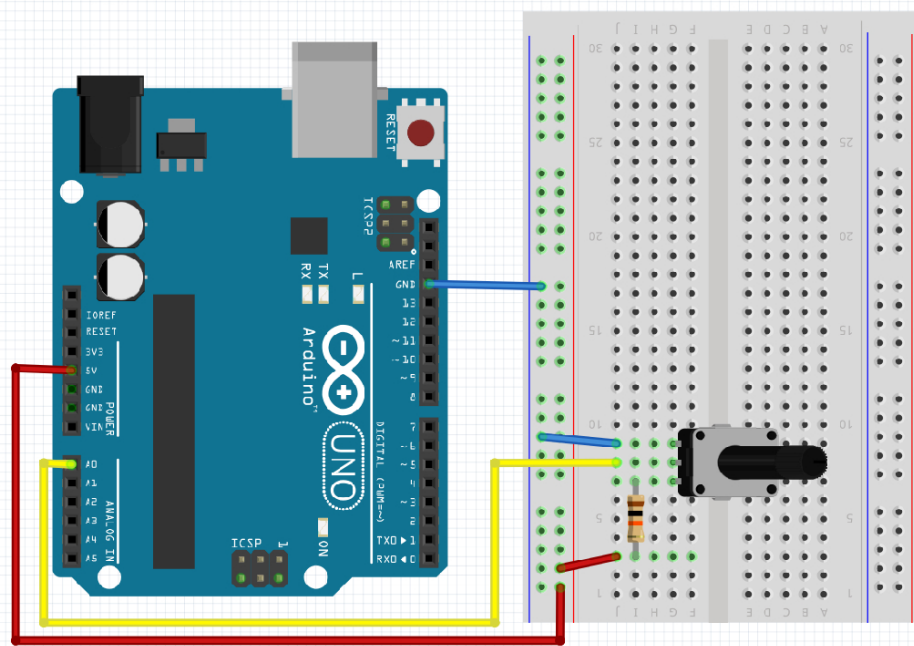
```

**Pictorial /Schematic**



**Note /Try to**

Detailed analysis of the ADC with Arduino is shown in the attached sheet.  
 Change the circuit configuration, adding a 10 kΩ resistor as shown in the diagram.



Are there variations on the signal reading compared to the previous case? Which?

**Sketch**  
  
**ANALOGIC SIGNAL READING**

```

//Arduino - Analog Read

void setup() {
  Serial.begin(9600);          // apro comunicazione seriale
}

void loop() {
  int adcValue = analogRead(A0); // leggo valore su A0
  Serial.print("Valore adc: "); // invio testo al monitor seriale
}

```

```

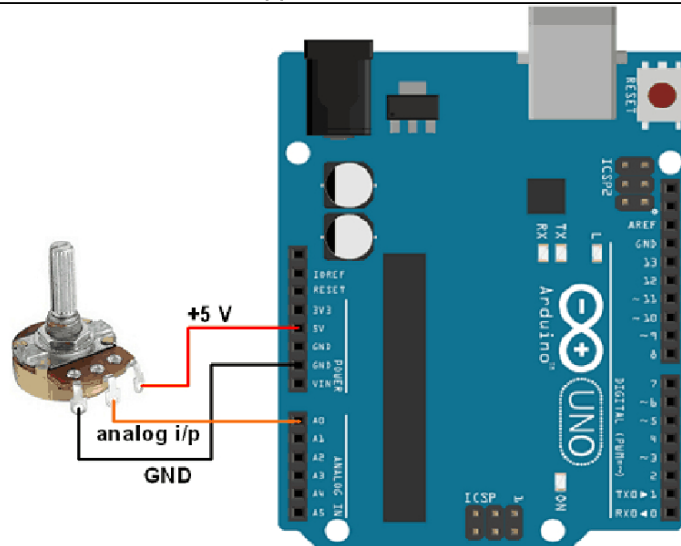
Serial.print(adcValue);           // invio valore al monitor seriale

float tensione = (adcValue*5.00)/1023.00; //calcolo tensione
Serial.print(" Valore Tensione: "); // invio testo al monitor seriale
Serial.println(tensione);        // invio valore al monitor seriale

delay(10);                       //piccolo ritardo per non leggere
}                                  // ADC troppo velocemente ed avere errori

```

Pictorial  
/Schematic



Try to Set a threshold in ADC such that if the threshold is exceeded a LED lights up.

# Resistenza di pull-up / pull-down

## Introduzione

Le resistenze di **pull-up** e di **pull-down** sono molto comuni quando si utilizzano i microcontrollori o qualsiasi altro dispositivo digitale.

I pin di ingresso dei circuiti digitali presentano normalmente un'impedenza abbastanza alta, e questo significa che basta una corrente minima per poter cambiarne lo stato. Ma significa anche che sono soggetti a captare qualsiasi disturbo elettromagnetico o elettrostatico e quindi a commutare le loro uscite in modo del tutto imprevisto ed incontrollato. Un pin di ingresso che si trova in questo stato (cioè a cui non è collegato nulla) si definisce in stato di **floating** (fluttuante).

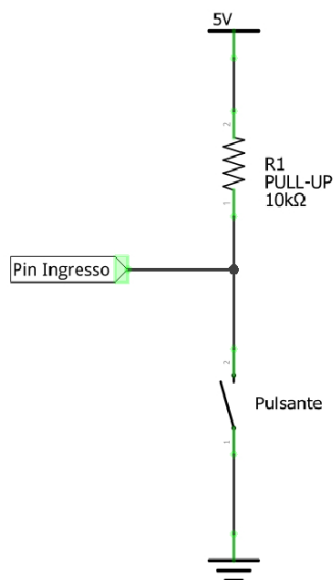
Lasciare un circuito in questo stato è un errore molto grave, in quanto è possibile (e senza dubbio lo sarà) che campi elettrici e magnetici esterni indurranno sul pin in questione delle tensioni (anche piccole) tali da far commutare il circuito interno.

Da qui deriva la necessità di impostare il suddetto pin in uno stato perfettamente conosciuto, che non possa portare a stati di incertezza sulle letture effettuate.

## Cosa è una resistenza di pull-up

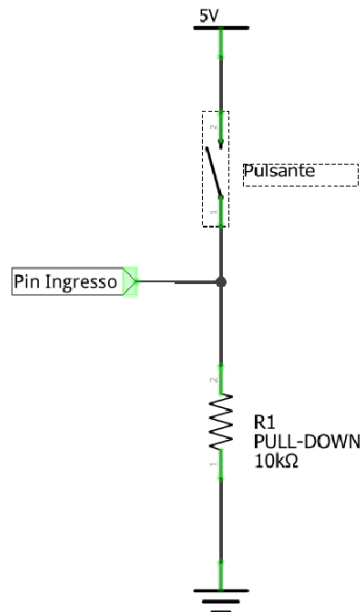
Ecco quindi che viene introdotto il concetto di resistenza di **pull-up** e di **pull-down**.

Si inserisce una piccola resistenza tra il pin di ingresso e la tensione di riferimento, per fare in modo che all'ingresso sia sempre presente un segnale certo. Se la resistenza viene inserita tra il pin e la **V<sub>dd</sub>** si parlerà di resistenza di pull-up, se invece viene inserita tra il pin e la **V<sub>ss</sub>** (normalmente GND o 0V) si parlerà di resistenza di pull-down. Vediamo degli esempi concreti.



Con la resistenza R1, il potenziale del pin è impostato alla tensione di riferimento **V<sub>cc</sub>** (in questo caso 5V). Quando l'interruttore viene chiuso l'ingresso viene portato a massa.

Quindi gli stati possibili per il pin vengono ben definiti dalle tensioni presenti ai due capi della connessione, senza possibili disturbi esterni. Adesso invertiamo la logica e colleghiamo la resistenza tra il pin e la massa, e l'interruttore tra il pin e la  $V_{cc}$ .



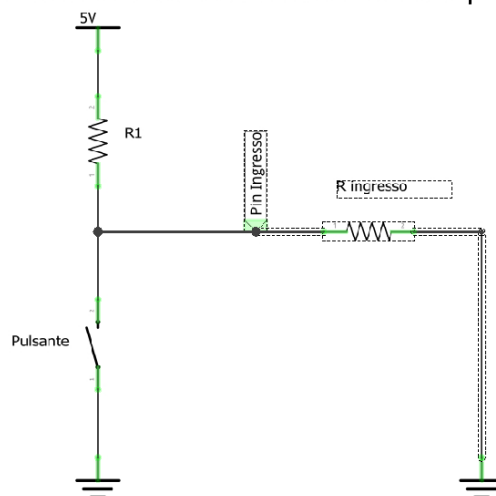
Con la resistenza R1, il potenziale del pin è impostato alla tensione di riferimento  $V_{ss}$  (in questo caso 0V). Quando l'interruttore viene chiuso l'ingresso viene portato al valore di  $V_{cc}$ .

Anche in questo caso, gli stati possibili per il pin sono ben definiti dalle tensioni presenti ai due capi della connessione.

### Calcolare il valore della resistenza di pull-up

A questo punto ci si pone quindi il dubbio su quali sono i valori accettabili per una resistenza di **pull-up** o di **pull-down** da collegare ad un porta di ingresso logico?

La resistenza deve essere dimensionata per poter svolgere correttamente il suo lavoro, cioè quello di portare a livello alto la tensione sul pin di ingresso.



Bisogna considerare che il circuito equivalente del pin di ingresso può essere sintetizzato come una resistenza, che chiamiamo  $R_{ingresso}$ .

Quando il pulsante è aperto, il pin di ingresso si trova nello stato **HIGH** e la corrente che fluisce nel circuito è bassa per via della resistenza, ma grazie all'alta impedenza sarà sufficiente a mantenerne alto lo stato.

Quando il pulsante viene premuto la corrente fluisce a massa e la tensione è prossima a 0V, portando il pin nello stato **LOW**.

Quindi il dimensionamento della resistenza di **pull-up** dipende molto dall'impedenza del pin di ingresso, che si può conoscere leggendo il datasheet del microcontrollore. Proviamo a fare dei calcoli per vedere come cambiano i valori di corrente e di tensione al pin di ingresso al variare della **R<sub>1</sub>**

Fissiamo i seguenti valori:

**V<sub>in</sub>** = tensione di alimentazione = 5 V

**R<sub>ingresso</sub>** = 1 MΩ

R <sub>1</sub>	R <sub>ingresso</sub>	R <sub>eq</sub> (Ω)	V <sub>pin</sub>	I (A)	Potenza (W)
1 Ω	1 MΩ	0,99999900	4,99999500	5	25
10 Ω	1 MΩ	0,99999000	4,99995000	0,5	2,5
100 Ω	1 MΩ	0,99990001	4,99950005	0,05	0,25
1 kΩ	1 MΩ	0,99900100	4,99500500	0,005	0,025
10 kΩ	1 MΩ	0,99009901	4,95049505	0,0005	0,0025
100 kΩ	1 MΩ	0,90909091	4,54545455	0,00005	0,00025
1 MΩ	1 MΩ	0,50000000	2,50000000	0,000005	0,000025
10 MΩ	1 MΩ	0,09090909	0,45454545	0,0000005	0,0000025
100 MΩ	1 MΩ	0,00990099	0,04950495	0,00000005	0,00000025

Possiamo notare 3 cose:

per valori bassi di **R<sub>1</sub>**, la caduta di tensione è irrilevante, ma la corrente che entra nel pin di ingresso è molto alta, con relativi alti consumi

per valori compresi tra  $1/100 R_{ingresso} < R_1 < 1/10 R_{ingresso}$  la caduta di tensione è irrilevante ed i consumi sono contenuti

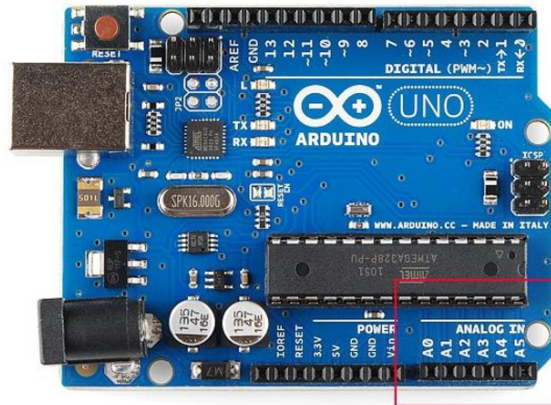
per valori di **R<sub>1</sub>** dello stesso ordine di grandezza di **R<sub>ingresso</sub>** o superiori, la caduta di tensione è notevole e potrebbe non essere sufficiente a portare il pin a livello **HIGH**

Nella pratica si può quindi applicare la regola empiricamente corretta di utilizzare una resistenza di **pull-up** (o di pull-down) pari ad 1/10 della **R<sub>ingresso</sub>**.

Nei progetti con **Arduino** possiamo tranquillamente utilizzare una resistenza di **pull-up** di circa 10 kΩ.

# Analog to Digital Conversion

An Analog to Digital Converter (ADC) is a very useful feature that converts an analog voltage on a pin to a digital number. By converting from the analog world to the digital world, we can begin to use electronics to interface to the analog world around us.



Not every pin on a microcontroller has the ability to do analog to digital conversions. On the Arduino board, these pins have an 'A' in front of their label (A0 through A5) to indicate these pins can read analog voltages.

ADCs can vary greatly between microcontroller. The ADC on the Arduino is a 10-bit ADC meaning it has the ability to detect 1.024 ( $2^{10}$ ) discrete analog levels (from 0 to 1023). Some microcontrollers have 8-bit ADCs ( $2^8 = 256$  discrete levels) and some have 16-bit ADCs ( $2^{16} = 65.536$  discrete levels).

The way an ADC works is fairly complex. There are a few different ways to achieve this feat (see Wikipedia for a list), but one of the most common technique uses the analog voltage to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete.

## Relating ADC Value to Voltage

The ADC reports a *ratiometric value*. This means that the ADC assumes 0V is 0, 5V is 1023 and anything less than 5V will be a ratio between 5V and 1023.

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

Analog to digital conversions are dependant on the system voltage. Because we predominantly use the 10-bit ADC of the Arduino on a 5V system, we can simplify this equation slightly:

$$\frac{1023}{5} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

If your system is 3,3V, you simply change 5V out with 3,3V in the equation. If your system is 3,3V and your ADC is reporting 512, what is the voltage measured? It is approximately 1,65V.

If the analog voltage is 2,12V what will the ADC report as a value?

$$\frac{1023}{5,00\ V} = \frac{ADC\ Reading}{2,12\ V}$$

Rearrange things a bit and we get:

$$ADC\ Reading = \frac{2,12\ V}{5,00\ V} \times 1023 = 434$$