

INTRODUZIONE ALLA ROBOTICA CON ARDUINO

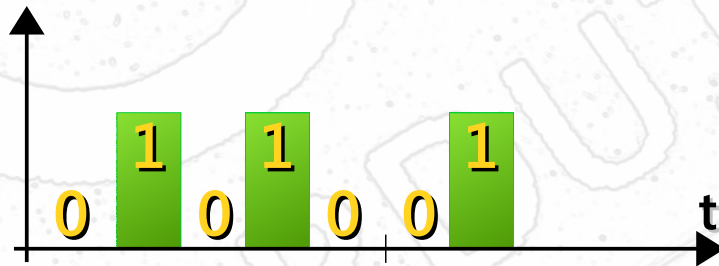
D.A.M. Bros Robotics - www.dambrosrobotics.it

Scheda Riassuntiva Lezione 3 - Input digitali

Il termine digitale deriva dal termine inglese digit (cifra): i calcolatori digitali operano elaborando quantità numeriche elementari che prendono il nome di cifre binarie. Una cifra (digit) può assumere solo due valori logici:

Zero (livello logico basso) - Uno (livello logico alto) che prendono il nome di bit.

Una sequenza di bit rappresenta un segnale digitale, che è di tipo discreto cioè può assumere solo due i due valori alto (0) e basso (1).



La scheda Arduino Uno alimentata a 5V interpreta i segnali digitali nel seguente modo :

Zero logico: Tensione compresa tra 0V e 1.5V

Uno logico: Tensione compresa tra 3V e 5V

Tutti i valori di tensione compresi tra 1.5V e 3V possono essere interpretati come 0 oppure 1 in modo casuale e quindi generano uno stato non definito.

Arduino dispone di 14 pin digitali configurabili come Ingresso o uscita e numerati da 0 a 13. **Gli ingressi Analogici possono essere utilizzati come IO digitali e per essi si può utilizzare la numerazione da 14 a 19 oppure da A0 ad A5.** Di default un pin digitale è configurato come ingresso, in ogni caso l'istruzione da utilizzare per la configurazione è la seguente :

`pinMode(pin,mode);`

Dove al posto di pin si deve inserire il numero del pin digitale che si utilizza, mentre il campo mode può assumere una delle seguenti espressioni

INPUT se il pin è di ingresso

OUTPUT se il pin è di uscita

INPUT_PULLUP attiva il resistore di pullup integrato

Ad esempio per utilizzare il pin numero 5 come ingresso si dovrà **scrivere nel setup**

```
void setup{  
pinMode(5,INPUT);  
}
```

Volendo leggere lo stato presente su di un pin digitale si deve invece utilizzare l'istruzione

digitalRead(pin)

Dove al posto di pin si deve inserire il numero del pin digitale di cui si vuole conoscere lo stato. Trattandosi di un ingresso digitale la funzione digitalRead restituisce :

0 (LOW) se sul pin è presente una tensione compresa tra 0V e 1.5V (tra 0 e circa 1V per le schede a 3.3V)

1 (HIGH) se sul pin è presente una tensione compresa tra 3V e 5V (tra 2V e 3.3V per le schede a 3.3V)

Quindi per leggere lo stato del pin digitale 5 si dovrà scrivere :

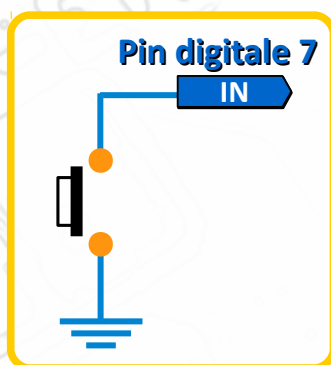
```
int stato;  
stato=digitalRead(5);
```

La variabile stato assume il valore 0 o 1 a seconda del valore di tensione presente sul pin digitale 5.

Non applicare ai pin tensioni superiori a 5V o negative altrimenti si distrugge il pin e/o il microcontrollore.

Come applicare un segnale digitale alla scheda Arduino

Per generare un segnale digitale che non vari casualmente si può utilizzare un pulsante normalmente aperto (NA) ma non basta collegarlo da solo tra l'ingresso digitale la massa. Infatti in questo modo quando si preme il pulsante si legge sicuramente il valore zero logico, ma quando il pulsante non è premuto la lettura torna è casuale.

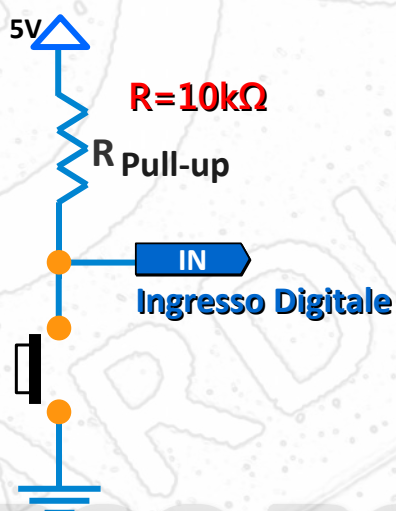


Per risolvere questo problema si devono usare dei resistori di pull-up o pull-down che consentono di vincolare il potenziale dell'ingresso digitale al valore di 0V se si vuole leggere uno zero logico, oppure a 5V se si vuole leggere un uno logico.

Rete di pullup con pulsante Normalmente aperto

Se il **pulsante è premuto** il pin è collegato a massa 0V e si legge lo **0 logico**.

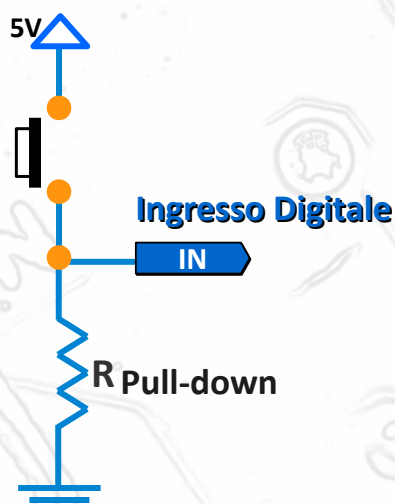
Se il **pulsante non è premuto** il pin è collegato all'alimentazione +5V e si legge l' **1 logico**.



Rete di pulldown con pulsante Normalmente aperto

Se il **pulsante è premuto** il pin è collegato all'alimentazione +5V e si legge l' **1 logico**.

Se il **pulsante non è premuto** il pin è collegato a massa 0V e si legge lo **0 logico**.



Il valore del resistore R in entrambi i casi è scelto in base a considerazioni sulla potenza e rumore

$$R=10k\Omega \quad I=V/R=5/10k\Omega=500\mu A$$

$$R=100k\Omega \quad I=50\mu A$$

$$R=1\Omega \quad I=5A \quad \text{Potenza alta } 25W !!$$

$$R=1M\Omega \quad I=5\mu A \quad \text{Rumore !!}$$

Anche il pulsante ha una sua portata nominale (ad esempio 50mA)

Valori troppo bassi del resistore (1Ω , 10Ω) comportano la circolazione di una corrente troppo alta e quindi un consumo eccessivo della batteria di alimentazione della scheda.

Un valore troppo alto del resistore ($1M\Omega$, $10M\Omega$) rende il segnale sensibile ai disturbi esterni di natura elettromagnetica.

Normalmente si sceglie un valore compreso tra $1k\Omega$ e $10k\Omega$ che rappresenta un buon compromesso tra i due problemi precedenti.

Resistore di pullup integrato nel microcontrollore

All'interno del microcontrollore ATmega328 in **corrispondenza di ogni pin digitale di IO e dei pin analogici** è presente un resistore il cui valore è compreso tra 20kΩ e 50kΩ, che può essere utilizzato come resistore di pullup, in sostituzione di quello esterno.

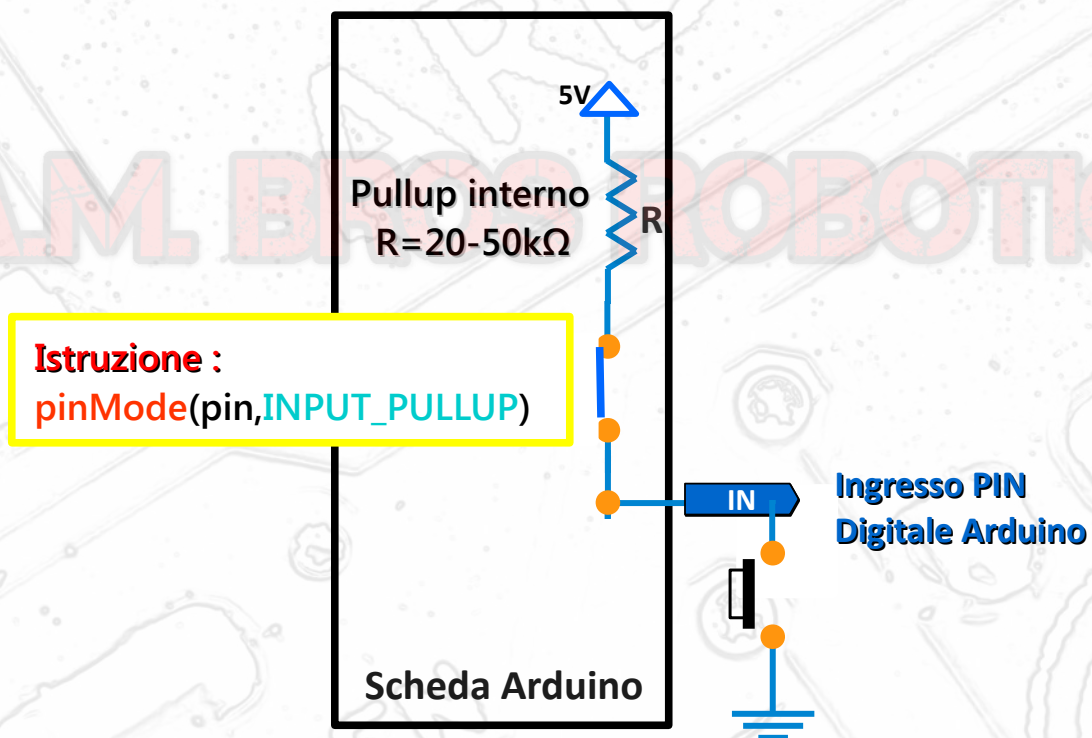
Per attivare tale resistore si deve utilizzare nel setup l'istruzione :

```
pinMode(pin,INPUT_PULLUP);
```

Utilizzando il resistore di pullup interno, al pin digitale andrà collegato soltanto il pulsante. La dichiarazione per attivare il pullup interno può essere fatta sempre nel setup anche con le due istruzioni:

```
pinMode(pin,INPUT);    digitalWrite(pin,HIGH);
```

La disattivazione del resistore di pullup è automatica, infatti un nuovo programma automaticamente pone tutti i pin IO come INPUT salvo diversa impostazione nel setup.



Programma che legge lo stato di un pin digitale

// Legge un segnale dal pin digitale

const int Digitalinput = 7; // Piedino Digitale

const int BAUD_RATE = 9600; // Velocità di comunicazione porta seriale

const int Time = 500; // Tempo di attesa tra due cicli consecutivi

void setup() {

Serial.begin(BAUD_RATE); // Inizializzazione porta seriale

pinMode(Digitalinput, **INPUT**); // Questa dichiarazione può essere omessa

}

void loop() {

int DigitalValue= **digitalRead**(Digitalinput); // Lettura dello stato logico presente sul piedino digitale

Serial.print("Stato logico sul piedino digitale = "); // COMMENTARE questa istruzione per usare il plotter
 // seriale

Serial.println(DigitalValue); // Stampa su monitor seriale lo stato logico presente sul pin digitale

delay(Time); // Attesa per visualizzazione lenta sul monitor seriale

}