

Gli Interrupt.

- La tecnica dell'interruzione (interrupt) consiste nell'inviare, da parte delle periferiche, un segnale di richiesta di interruzione; il microcontrollore può gestire le interruzioni rispettando le priorità nel caso vi siano più richieste di interruzione in contemporanea; prima di interrompere il programma, il microcontrollore finisce l'istruzione in corso e poi passa alla subroutine di gestione dell'interruzione. Più semplicemente, il processore esegue normalmente il suo lavoro, ma quando succede qualcosa di importante riceve una chiamata dal gestore degli interrupt che gli dice di iniziare la Interrupt Service Routine (ISR). Aniché impiegare tutto il tempo a verificare una certa condizione all'interno del codice, viene usato un bit del microcontrollore per segnalare il verificarsi

sketch_nov17a\$

```
volatile int valoreSensore=LOW;
#define BUTTON 2;

void setup(){
  Serial.begin(9600);
  pinMode(2,INPUT_PULLUP);
  digitalWrite(2,HIGH);
  pinMode(13,OUTPUT);
  attachInterrupt(0,catturato0,CHANGE);
}

void loop(){
  int j;
  for(int i=0;i<100;i++){
    j=j+i*i;
    Serial.println("Ora attendo...");
    delay(5000);

    j=1;
  }
}

void catturato0(){
  valoreSensore=digitalRead(2);
  digitalWrite(13,!valoreSensore);
  Serial.println("INT");
  delay(100);
}
```

procedura associata all'interrupt.

di una determinata condizione. Sui chip standard Atmega sono presenti solo due ingressi di interrupt, che sono collegati agli ingressi **digitali 2 e 3**. La subroutine di interrupt può essere eseguita sul fronte di discesa (**FALLING**), di salita (**RISING**) oppure quando l'ingresso del piedino cambia di stato (**CHANGE**), **LOW** fa scattare l'evento di interrupt ogni volta che il pin (2 o 3) ha valore logico basso. L'istruzione fondamentale per l'uso degli interrupt è: **attachInterrupt()** la quale riceve tre parametri: il piedino da utilizzare, il nome della procedura da eseguire e il tipo di evento che si intende intercettare.

Procedimento:

Il programma realizzato è utile esclusivamente all'apprendimento e all'uso degli interrupt e consiste nell'accensione di un Led (collegato al pin digitale 13) nel momento in cui viene premuto il pulsante (collegato invece al pin digitale 2 di interrupt). Qualunque sia l'operazione che il microcontrollore sta eseguendo, viene avviata la procedura relativa all'interrupt e gestiti l'accensione del Led e l'invio di un messaggio tramite porta seriale che ci confermi l'effettiva esecuzione della

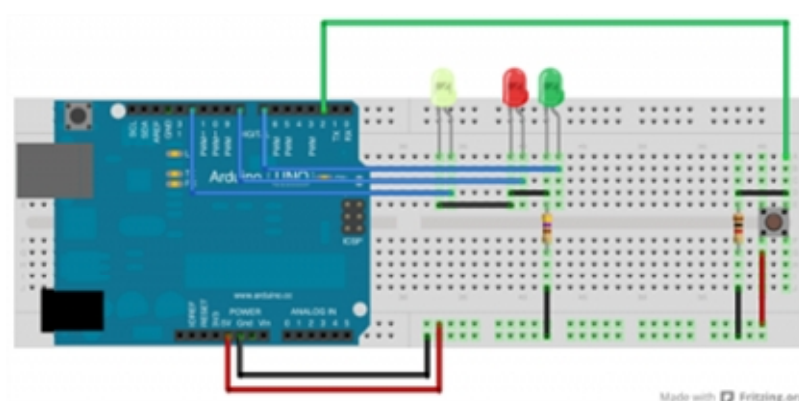
Conclusioni:

L'utilizzo degli interrupt può essere particolarmente utile nel caso in cui ci si voglia accertare che un'operazione venga eseguita istantaneamente oppure nel caso in cui ci si voglia accertare che venga eseguita un'operazione in corrispondenza di un evento asincrono esterno. Per esempio nel caso in cui si debba controllare lo stato di diversi pin di input, si potrebbe rischiare che il programma non si accorga di un eventuale cambiamento di stato di un pin; questa situazione potrebbe verificarsi se il microcontrollore sta controllando lo stato di altri piedini di input. Abilitando gli interrupt ci si può assicurare che il microcontrollore esegua immediatamente la procedura associata all'interrupt e che quindi venga gestita la priorità degli eventi asincroni. Il microcontrollore Atmega a bordo dell'Arduino Uno, prevede solo due ingressi di interrupt, mentre la Atmega2560 a bordo dell'Arduino Mega presenta quattro ingressi di interrupt.

Nel caso in cui si vogliano collegare più elementi agli ingressi di interrupt, si può far uso di multiplexer, oppure di porte logiche OR che permettano di azionare gli interrupt ogni volta che almeno uno dei dispositivi collegati al gate raggiunga un valore logico alto.

Altro esempio

Lo schema elettrico:



collega il pulsante al pin 2 che utilizzerai come interrupt, alla pressione del pulsante il valore letto sul pin 2 sarà HIGH in quanto la pressione di pulsante mette in contatto il pin 2 di arduino con il polo positivo +5v.

```
int ledRed=8;
int ledGreen=7;
int ledYellow=12;
int interrCount=0;

void setup()
{
  pinMode(ledRed, OUTPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(ledYellow, OUTPUT);

  digitalWrite(ledRed, LOW);
  digitalWrite(ledGreen, LOW);
  digitalWrite(ledYellow, LOW);

  attachInterrupt(0, interruptGiallo, RISING);
}
```

```
void loop() {
  interrCount++;

  digitalWrite(ledRed, HIGH);
  digitalWrite(ledGreen, LOW);
  delay(300);
  digitalWrite(ledRed, LOW);
  digitalWrite(ledGreen, HIGH);
  delay(300);

  if ( interrCount == 10 )
  {
    interrCount = 0;
    digitalWrite(ledYellow, LOW);
  }
}

void interruptGiallo()
{
  digitalWrite(ledYellow, HIGH);
}
```