

Descrizione di alcune istruzioni di Arduino

A cura del prof. Giuseppe Spalierno luglio 2012

Un pin può essere configurato di ingresso con pinMode:

```
pinMode(9, INPUT);
```

Può essere reso alto con digitalWrite:

```
digitalWrite(9, HIGH);
```

Ciò abilita la resistenza interna di pull-up da 20Kohm.

La lettura di tale pin fornisce un livello logico HIGH. Per leggere un livello logico basso si deve applicare a tale linea un livello basso da una circuiteria esterna.

Un pin può essere configurato di uscita con pinMode:

```
pinMode(9, OUTPUT);
```

Un livello alto a tale linea scritto con digitalWrite:

```
digitalWrite(9, HIGH);
```

fornisce +5V e può pilotare un LED con resistenza in serie verso massa.

Può essere connesso ad un altro pin di uscita configurato LOW.

Il livello logico LOW è considerato tale se applicato ad un pin di ingresso con valore inferiore a 2V. Se LOW è scritto su un pin di uscita il valore è 0V e può pilotare un LED avente in serie una resistenza collegata a +5V oppure ad un pin di uscita impostata a HIGH.

Un pin di ingresso presenta una resistenza interna di 100Mohm: si può connettere un sensore ma non un LED.

Un pin di output presenta una resistenza di uscita bassa. Può erogare o assorbire fino a 40mA. Ciò consente di pilotare un LED ma non un motore e relè. Collegando ad un pin di uscita direttamente 5V si rischia di danneggiare o distruggere il pin.

Costanti numeriche:

decimali: 123

binarie : B1100

ottali : 0123

esadecimali: 0x7B

costante senza segno: 33u oppure 33U (unsegnd)

costante lunga: 387546L oppure 387546l (long)

costante lunga senza segno 535647ul

Per definire una costante:

```
#define ledPin 3 (senza punto e virgola e senza uguale. Si può omettere il  
cancellotto. E' preferibile il comando const anzichè #define)  
const float pi = 3.14;  
pi è costante in virgola mobile di valore 3.14.
```

**La scrittura analogica di un pin avviene alla frequenza di 490 Hertz.
I pin sui quali prelevare l'uscita analogica sono: 3, 5, 6, 9, 10 e 11.**

Esempio:

```
analogWrite(3, 225);  
il secondo valore è compreso tra 0 (sempre basso) e 255 (sempre alto: 100%)  
I pin 5 e 6 forniscono un PWM maggiore rispetto a quello atteso perché  
condividono, con le funzioni delay() e millis(), lo stesso timer interno.
```

Esempio:

```
int ledPin = 9;          // LED connected to digital pin 9  
int analogPin = 3;       // potentiometer connected to analog pin 3  
int val = 0;             // variable to store the read value  
  
void setup()  
{  
  pinMode(ledPin, OUTPUT); // sets the pin as output  
}  
  
void loop()  
{  
  val = analogRead(analogPin); // read the input pin  
  analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023,  
  analogWrite values from 0 to 255  
}
```

millis(); fornisce il numero di millisecondi dall'avvio del programma.

Esempio:

```
unsigned long time;  
  
void setup(){  
  Serial.begin(9600);  
}  
void loop(){  
  Serial.print("Time: ");  
  time = millis();  
  //prints time since program started  
  Serial.println(time);  
  // wait a second so as not to send massive amounts of data  
  delay(1000);  
}
```

micros(); fornisce il numero di microsecondi dall'avvio del programma. La risoluzione è 4microsecondi.