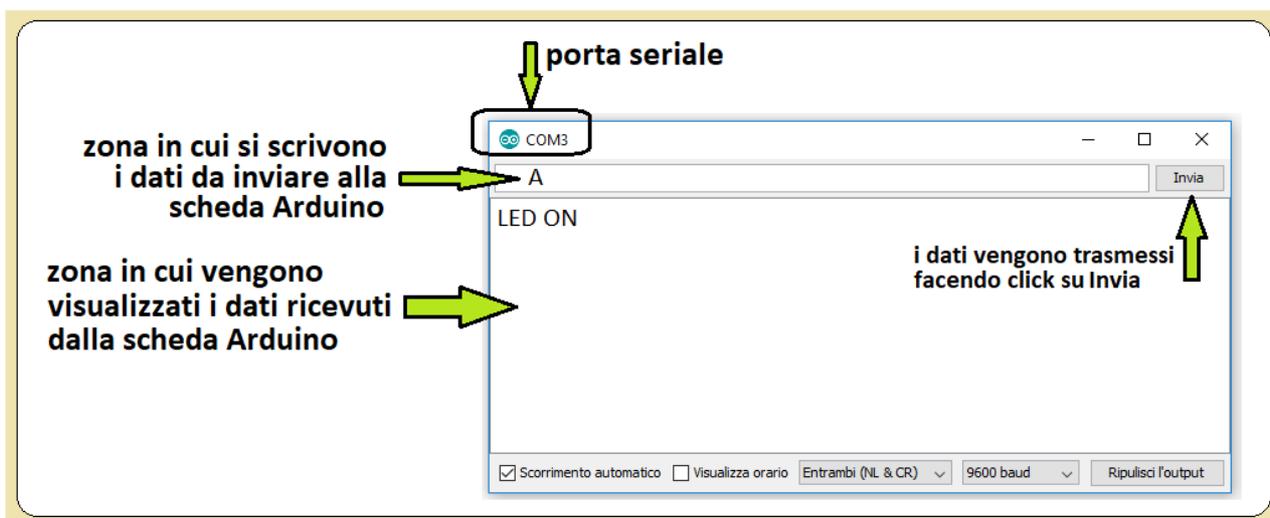


Aggiornamento

Trasmissione dati tramite monitor seriale

Il monitor seriale può essere utilizzato per trasmettere dati dal PC alla scheda Arduino e viceversa.

Sul monitor seriale sono presenti due zone appositamente dedicate a tale scopo (vedere figura).



Un primo esempio pratico molto semplice può essere il seguente:

- digitando il carattere A si accende un LED, digitando il carattere S il LED si spegne;
- nella zona sottostante viene visualizzato lo stato del LED (ON o OFF).

ESERCITAZIONE

Scopo dell'esercitazione è l'invio di caratteri alla scheda Arduino dal monitor seriale.

Con la ricezione dei caratteri la scheda si comporta in modo diverso a seconda del carattere ricevuto; in particolare:

- un LED (pin 13) si accende per 3 s se dal monitor seriale viene inviato il carattere 1;
- un LED (pin 13) si accende per 8 s se dal monitor seriale viene inviato il carattere 2;
- un LED (pin 13) si accende a intermittenza per 10 volte (blink) e poi rimane spento se dal monitor seriale viene inviato il carattere 3;
- un LED (pin 13) si accende a intermittenza per 10 volte (blink) e poi rimane acceso se dal monitor seriale viene inviato il carattere 4.

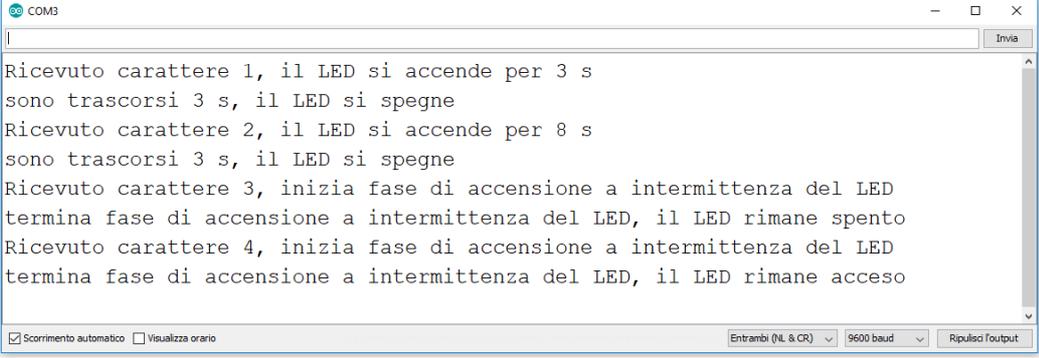
In relazione al tipo di risposta, nello spazio di monitor dedicato, viene visualizzato in tempo reale lo stato del LED (accesso/spento/blink).

Il listato del programma viene di seguito proposto.

Si evidenziano in particolare le istruzioni tipiche che consentono la trasmissione dei dati ovvero *Serial.begin(9600)*, *Serial.available()*, *Serial.read()* e *Serial.println*. L'istruzione *Serial.available()* restituisce il numero di byte arrivati dal PC sulla porta seriale e disponibili per la lettura.

```
char car;
int l;
void setup() {
  Serial.begin(9600); // velocità di comunicazione
  pinMode(13,OUTPUT); // il pin 13 viene definito come uscita
}
void loop() {
  if(Serial.available()) { // se il dato è disponibile per la lettura
    car=Serial.read(); // viene letto il carattere
    if(car=='1') {
      Serial.println("Ricevuto carattere 1, il LED si accende per 3 s");
      digitalWrite(13,HIGH);
      delay(3000);
      digitalWrite(13,LOW);
      Serial.println("sono trascorsi 3 s, il LED si spegne");
    }
    if(car=='2') {
      Serial.println("Ricevuto carattere 2, il LED si accende per 8 s");
      digitalWrite(13,HIGH);
      delay(8000);
      digitalWrite(13,LOW);
      Serial.println("sono trascorsi 3 s, il LED si spegne");
    }
    if(car=='3') {
      Serial.println("Ricevuto carattere 3, inizia fase di accensione a intermittenza del LED");
      for(l=0;l<10;l++) {
        digitalWrite(13,HIGH);
        delay(500);
        digitalWrite(13,LOW);
        delay(500);
      }
      Serial.println("termina fase di accensione a intermittenza del LED, il LED rimane spento");
    }
    if(car=='4') {
      Serial.println("Ricevuto carattere 4, inizia fase di accensione a intermittenza del LED");
      for(l=0;l<10;l++) {
        digitalWrite(13,HIGH);
        delay(500);
        digitalWrite(13,LOW);
        delay(500);
      }
      digitalWrite(13,HIGH);
      Serial.println("termina fase di accensione a intermittenza del LED, il LED rimane acceso");
    }
  }
}
```

Nella figura sottostante viene visualizzato lo stato del LED nei quattro casi previsti.



```
COM3
Ricevuto carattere 1, il LED si accende per 3 s
sono trascorsi 3 s, il LED si spegne
Ricevuto carattere 2, il LED si accende per 8 s
sono trascorsi 3 s, il LED si spegne
Ricevuto carattere 3, inizia fase di accensione a intermittenza del LED
termina fase di accensione a intermittenza del LED, il LED rimane spento
Ricevuto carattere 4, inizia fase di accensione a intermittenza del LED
termina fase di accensione a intermittenza del LED, il LED rimane acceso
```

Scorrimiento automatico Visualizza orario Entrambi (NL & CR) 9600 baud Ripulisci l'output

Una proposta ulteriore potrebbe essere quella di integrare l'esercitazione visualizzando il trascorrere dei secondi.