



Arduino Lab

Software in contatto con l'ambiente

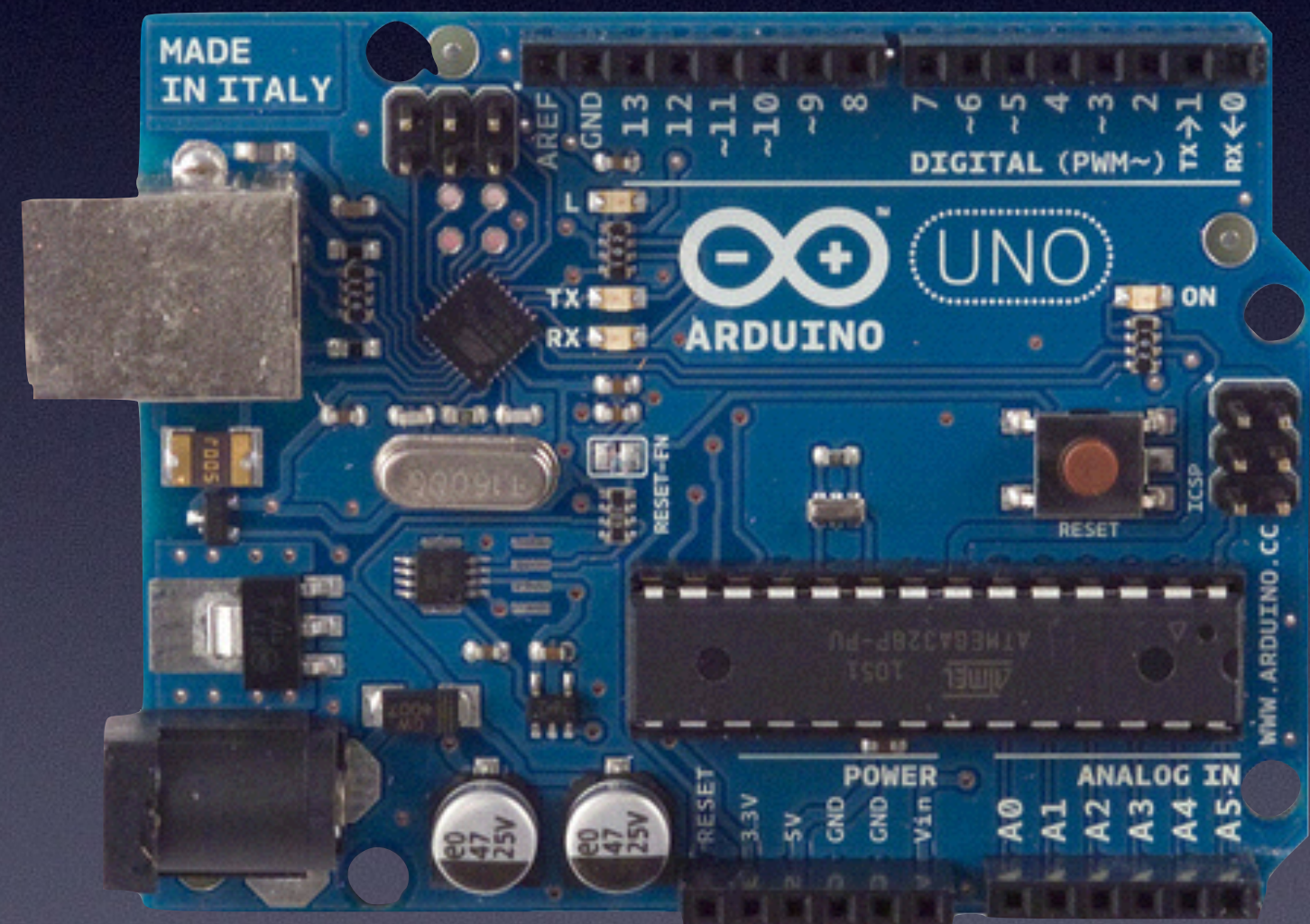
Tancredi Orlando - Mario Restuccia

Arduino

Arduino è una piattaforma di prototipazione open source

Hardware (Arduino UNO):

- Board con microcontroller (ATmega328)
- 14 pin I/O digitale
- 6 pin di input analogico
- Connessione USB usata come alimentazione, porta seriale e per caricare gli *sketch*
- Made in Italy
- Documentazione su come costruire la scheda disponibile sotto CC BY-SA 2.5



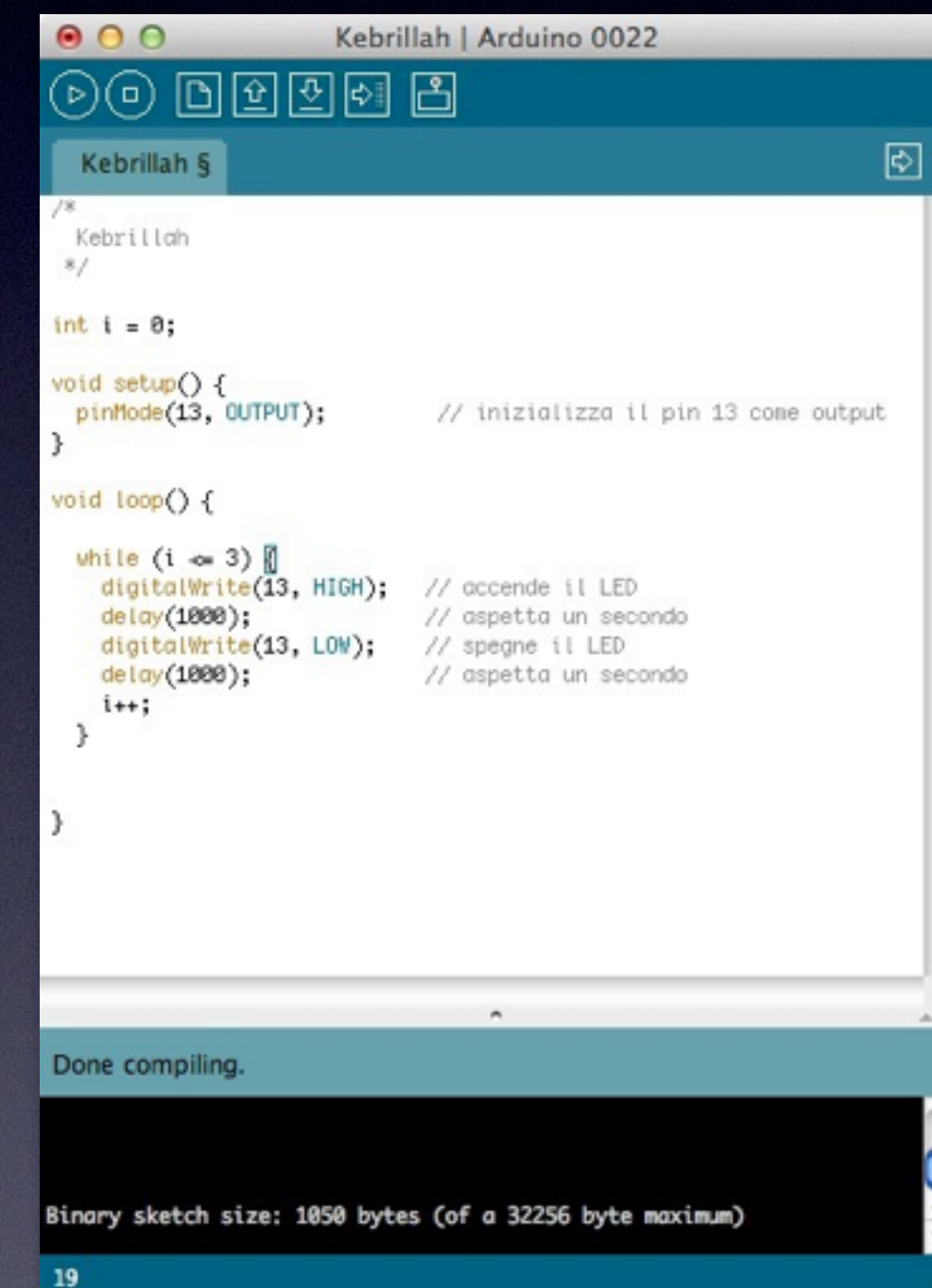
Arduino UNO, il modello da noi usato

Arduino

Arduino è un IDE per caricare gli sketch sulla scheda

Software:

- IDE che compila gli sketch e li carica sulla scheda per eseguirli
- Programmabile in Wiring, linguaggio simile a C++
- Multiplatforma
- Utilizzato per scrivere il codice e caricarlo sulla board
- Disponibile sotto licenza GPL v.2



Arduino 0022

Kebrillah

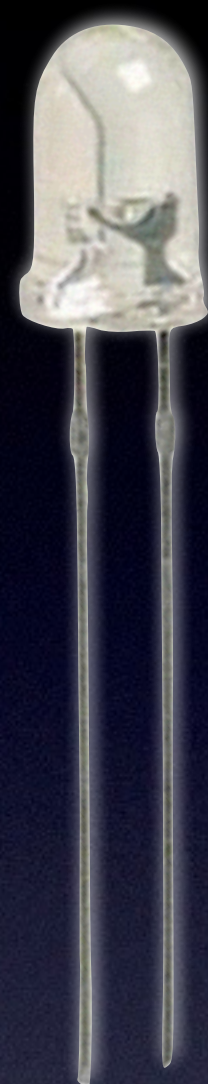
Esempio di accensione di un LED

```
1. void setup() {  
2.   pinMode(13, OUTPUT);  
3. }
```

```
5. void loop() {  
6.   digitalWrite(13, HIGH);  
7.   delay(2000);  
8.   digitalWrite(13, LOW);  
9.   delay(2000);  
10. }
```

1. Metodo “setup”, eseguito una volta sola
2. Inizializza il pin 13 come output
3. Fine del metodo “setup”

5. Metodo “loop”, eseguito in continuazione
6. Alimenta il pin 13, accendendo il LED
7. Aspetta 2 secondi
8. Spegne il LED
9. Aspetta 2 secondi
10. Fine del metodo “loop”



Pin 13
(resistenza $1\text{K}\Omega$)

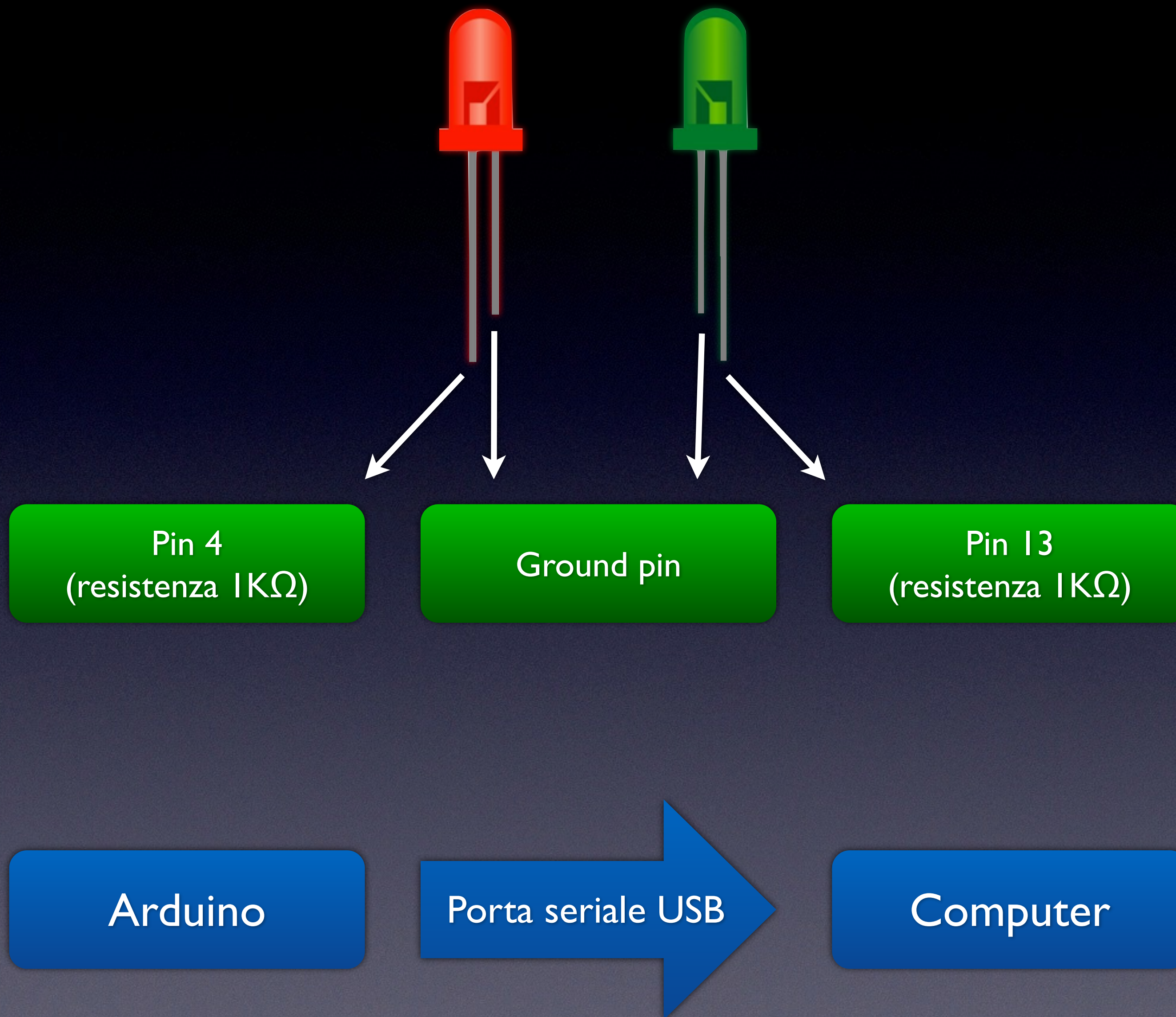
Ground pin

SerialLED

Accende dei LED e comunica via seriale

```
1. void setup() {  
2.   pinMode (13, OUTPUT);  
3.   pinMode (4, OUTPUT);  
4.   Serial.begin (9600);  
5. }  
  
7. void loop() {  
8.   Serial.println("verde");  
9.   digitalWrite(13, HIGH);  
10.  delay(1000);  
11.  digitalWrite(13, LOW);  
12.  
13.  Serial.println("rosso");  
14.  digitalWrite(4, HIGH);  
15.  delay(1000);  
16.  digitalWrite(4, LOW);  
17.  
18.  Serial.println("Hello MeLUG!");  
19.  delay(5000);  
20.}
```

```
1. Metodo setup  
2. Imposta il pin 13 (LED verde) come output  
3. Imposta il pin 4 (LED rosso) come output  
4. Numero di bit al secondo per seriale  
5. Fine metodo setup  
  
7. Metodo loop  
8. Invia via seriale il messaggio "verde"  
9. Accende il LED connesso al pin 13  
10. Aspetta 1 secondo  
11. Spegne il LED  
  
13. Invia il messaggio "rosso"  
14. Accende il LED connesso al pin 4  
15. Aspetta 1 secondo  
16. Spegne il LED  
  
18. Invia un messaggio di saluto via seriale  
19. Aspetta 5 secondi  
20. Fine metodo loop
```

PSLight

Implementazione controller PSX



Libreria PSX

1	2	3	4	5	6	7	8	9
o	o	o	o	o	o	o	o	o

Demo

Sounduino

Genera frequenze audio inviate via seriale

Codice Arduino

```
1. const int BUZZ = 13;
2. const int BAUD = 9600;
3. int note = 0;
```

```
5. void setup()
6. {
7.   pinMode(BUZZ, OUTPUT);
8.   Serial.begin(BAUD);
9. }
```

```
11. void loop() {
12.   while (Serial.available() > 0) {
13.     int rx = Serial.read();
14.     if (rx != ';') {
15.       note = note + rx;
16.     } else {
17.       if (note != 0) {
18.         tone(BUZZ, note);
19.         note = 0;
20.       } else {
21.         noTone(BUZZ);
22.       }
23.     }
24.   }
25. }
```

1. Assegna alla costante BUZZ il numero 13
2. Baud rate (per la comunicazione seriale)

7. Inizializza il pin 13 come output
8. Inizializza la comunicazione seriale

12. Ascolta i dati in arrivo via seriale
13. Assegna alla variabile rx i dati via seriale
14. Se ciò che riceve è diverso da ";"
15. Incrementa la variabile note del valore rx
16. Altrimenti
17. Se la variabile note è diversa da zero
18. La riproduce
19. E reimposta a zero la variabile note
20. Altrimenti
21. Esegue una pausa senza emettere suoni

Codice Perl

```
1. #!/usr/bin/perl
2. use strict;
3. use warnings;
4. use Device::SerialPort;
5. use Time::HiRes qw(usleep); # For sleep in ms
```

```
7. if ($#ARGV + 1 != 2) {
8.     print "Usage: $0 port filename\n";
9.     print "Example: $0 /dev/ttyASM0 money.txt\n";
10.    exit 1;
11. }
```

```
13. my $file = $ARGV[0];
14. my $dev  = $ARGV[1];
```

```
16. if (!-e $file || !-e $dev) {
17.     print "File or brain not found.\n";
18.     exit 1;
19. }
```

```
21. my $arduino = DeviceSerialPort->new($dev);
22. $arduino->baudrate(9600);
23. $arduino->databits(8);
24. $arduino->parity("none");
25. $arduino->stopbits(1);
```

```
1. #!/usr/bin/perl
2. use strict;
3. use warnings;
4. use Device::SerialPort;
5. use Time::HiRes qw(usleep); # For sleep in ms
```

```
7. if ($#ARGV + 1 != 2) {
8.     print "Usage: $0 port filename\n";
9.     print "Example: $0 /dev/ttyASM0 money.txt\n";
10.    exit 1;
11. }
```

```
13. my $file = $ARGV[0];
14. my $dev  = $ARGV[1];
```

```
16. if (!-e $file || !-e $dev) {
17.     print "File or brain not found.\n";
18.     exit 1;
19. }
```

```
21. my $arduino = DeviceSerialPort->new($dev);
22. $arduino->baudrate(9600);
23. $arduino->databits(8);
24. $arduino->parity("none");
25. $arduino->stopbits(1);
```



```
27. my %frequencies; eval { %frequencies = do "frequencies.pl"; };
28. open NOTES, "$file";

30. my ($tempo, @tone, $note, $duration);

32. while (<NOTES>) {
33.     chomp;      # No newline
34.     s/#.*//;    # No comments
35.     s/^\s+//;   # No leading white
36.     s/\s+$//;   # No trailing white
37.     next unless length;
38.     if ($_ =~ m/^TEMPO/) {
39.         $tempo = split(/\s+/, $_, -1);
40.         print "Tempo is $tempo.";
41.     } else {
42.         @tone = split(/\s+/, $_);
43.     }
44.     $note = $frequencies{$tone[0]};
45.     $duration = $tone[1]*$tempo;
46.     print "Playing $tone[0] (@$note Hz) for $tone[1] units ($duration ms).";
47.     while ($note > 255) {
48.         $arduino->write(chr(255));
49.         $note -= 255;
50.     }
51.     $arduino->write(chr($note));
52.     $arduino->write(";");
53.     usleep($duration);
54. }

55. close NOTES;
```

```
27. my %frequencies; eval { %frequenc"; };
28. open NOTES, "$file";

30. my ($tempo, @tone, $note, $duration);

32. while (<NOTES>) {
33.     chomp;      # No newline
34.     s/#.*//;    # No comments
35.     s/^\s+//;   # No leading white
36.     s/\s+$//;   # No trailing white
37.     next unless length;
38.     if ($_ =~ m/^TEMPO/) {
39.         $tempo = split(/\s+/, $_, -1);
40.         print "Tempo is $tempo.";
41.     } else {
42.         @tone = split(/\s+/, $_);
43.     }
44.     $note = $frequencies{$tone[0]};
45.     $duration = $tone[1]*$tempo;
46.     print "Playing $tone[0]ne[1]
47.     while ($note > 255) {
48.         $arduino->write(chr(255));
49.         $note -= 255;
50.     }
51.     $arduino->write(chr($note));
52.     $arduino->write(";");
53.     usleep($duration);
54. }

55. close NOTES;
```




Ground pin



Pin 13
(resistenza $1\text{K}\Omega$)

Computer

Porta seriale USB

Arduino

Demo

Fotoduino

Monitora l'illuminazione ambientale, scrive il valore sul display

Codice Arduino

```
1. #include <LiquidCrystal.h>
2. const int inPin = 0;
3. const int led = 6;
4. int fotoLuce;
5. LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

7. void setup() {
8.   pinMode(led, OUTPUT);
9.   Serial.begin(9600);
10.  lcd.begin(2,20);
11.}

13.void loop() {
14.  int fotoLuce = analogRead(inPin);
15.  Serial.print("Luce: ");
16.  Serial.print(fotoLuce);
17.  lcd.setCursor(0, 0);
18.  lcd.print("Luce: " +fotoLuce);
19.  delay(1000);
20.  if(fotoLuce < 200) {
21.    digitalWrite(led, HIGH);
22.  }
23.  else {
24.    digitalWrite(led, LOW);
25.  }
26.}
```

```
1. Include la libreria per il display LCD
2. Assegna alla costante "inPin" il valore 0
3. Assegna alla costante "led" il valore 6
4. Dichiarare il valore "fotoLuce"
5. Elenca i pin ai quali è connesso il display

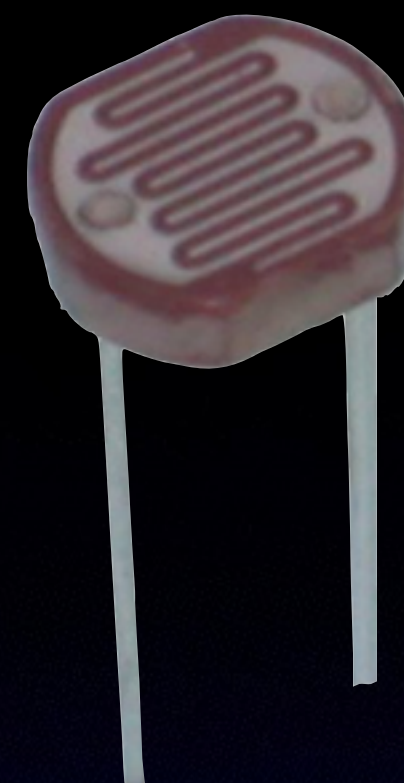
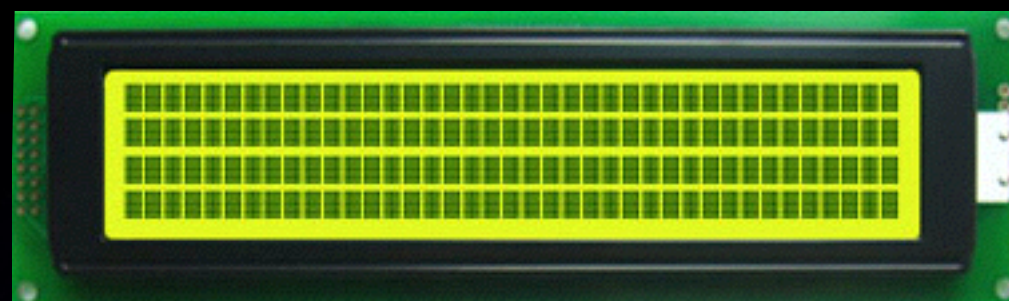
8. Dichiarare come uscita il pin 6
9. Baud rate per la comunicazione seriale
10. Numero di righe e colonne del display

14. Legge i dati dalla fotoresistenza
15. Scrive via seriale il valore della luce

17. Imposta il cursore del display all'inizio
18. Scrive sul display il valore della luce
19. Aspetta un secondo
20. Se c'è poca luce
21. Accende il led

23. Oppure
24. Spegne il led
```


Demo



Uscita digitale



Ingresso analogico

Ground - 5v

Arduino

Porta seriale USB

Computer

Grazie per l'attenzione

tancredi.orlando@gmail.com
mario.restuccia@gmail.com



Tancredi Orlando
Mario Restuccia



@turlando
@linkxvi