



Robotica avanzata: Raspberry Pi e Python

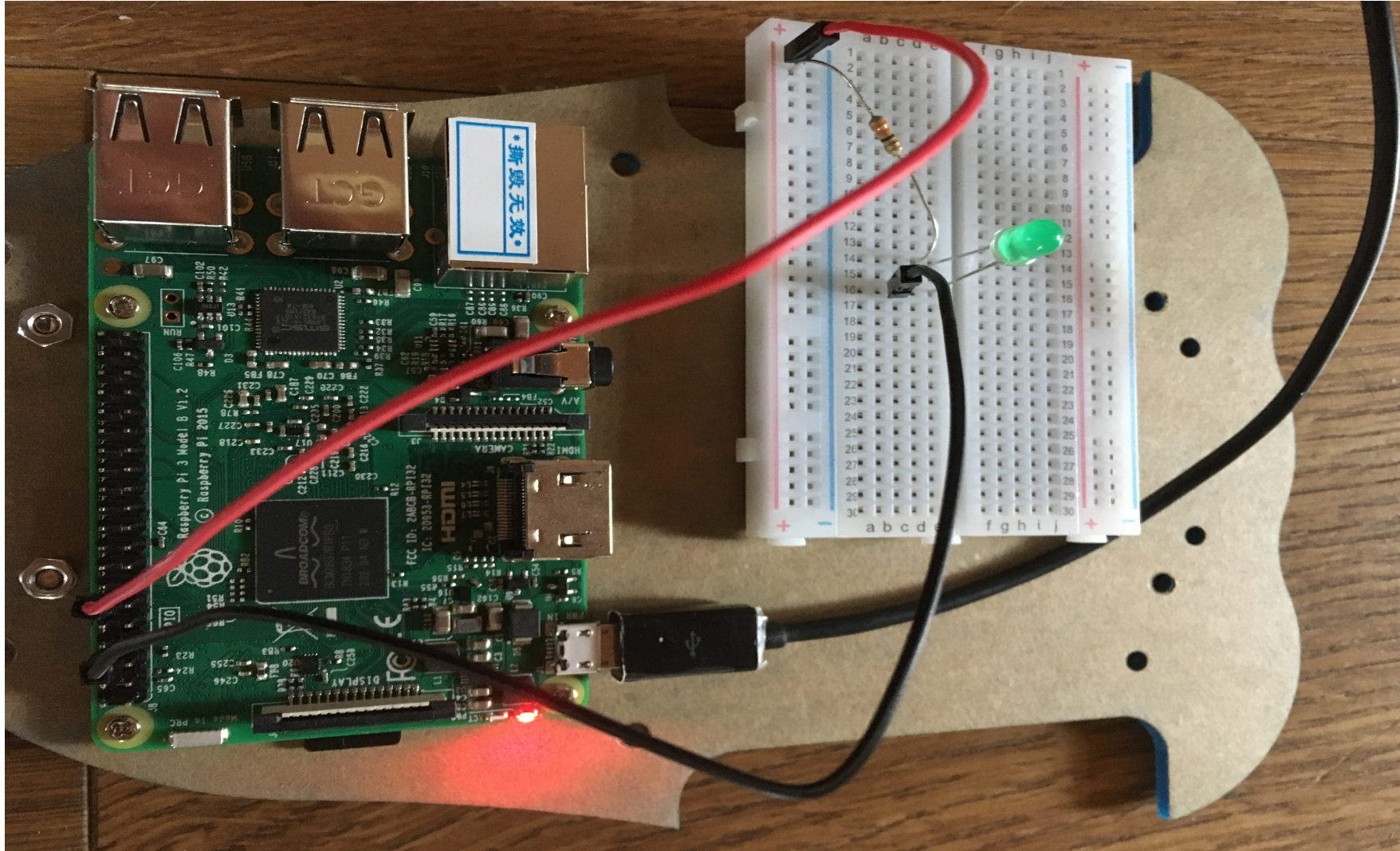
Lezione 2: un semplice programma Python (e anche in Scratch...)

RaspberryPi

II software: codice Python

```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BCM)
5  GPIO.setwarnings(False)
6  GPIO.setup(18, GPIO.OUT)
7
8  print "LED on"
9  GPIO.output(18, GPIO.HIGH)
10
11 time.sleep(1)
12
13 print "LED off"
14 GPIO.output(18, GPIO.LOW)
```

L'hardware



```
1  import RPi.GPIO as GPIO
2  import time
```

Queste prime due righe “includono” due librerie: RPi.GPIO e time. Le librerie estendono le funzioni native del linguaggio Python introducendone di nuove.

- Rpi.GPIO contiene le funzioni per l'utilizzo dei GPIO
- time contiene funzioni relative al RTC e alla misura di tempi / applicazione di delay

RaspberryPi

```
4    GPIO.setmode(GPIO.BCM)
5    GPIO.setwarnings(False)
6    GPIO.setup(18,GPIO.OUT)
```

Queste istruzioni configurano i GPIO:

- riga 4: definisce la modalità di nomenclatura dei pin a BCM. Vedi: <https://pinout.xyz/>
- riga 5: disabilita la segnalazione di warning
- riga 6: definisce che il pin 18 è un output

RaspberryPi


```
8  print "LED on"
9  GPIO.output(18,GPIO.HIGH)
```

Qui inizia il programma vero e proprio!

- riga 8: stampa sullo standard-output la stringa "LED on"
- riga 9: pone ad HIGH logico il pin 18, e il LED si accende!

RaspberryPi

```
11  time.sleep(1)
```

- riga 11: si utilizza la funzione sleep della libreria time. La funzione sleep produce una pausa nell'esecuzione del programma di durata pari all'argomento espresso in secondi.

RaspberryPi

```
13  print "LED off"  
14  GPIO.output(18, GPIO.LOW)
```

Qui il programma finisce

- riga 13: stampa sullo standard-output la stringa “LED off”
- riga 14: pone ad LOW logico il pin 18, e il LED si spegne!

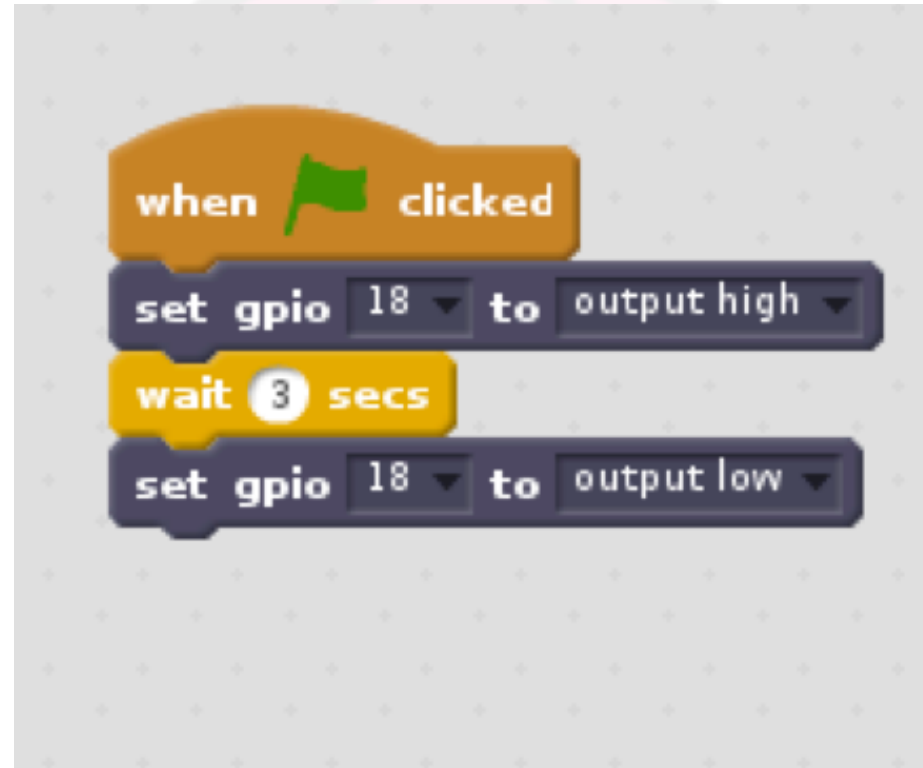
RaspberryPi

E in Scratch 1?



<https://www.raspberrypi.org/documentation/usage/gpio/scratch1/README.md>

E in Scratch 2?



Raspberry Pi

E in C?

```
#include <wiringPi.h>
#include <stdio.h>

#define LedPin 1

int main(void) {
    if(wiringPiSetup() == -1) { //when initialize wiringPi failed, print message to screen
        printf("setup wiringPi failed !\n");
        return -1;
    }

    pinMode(LedPin, OUTPUT);
    digitalWrite(LedPin, HIGH); //led on
    printf("led on\n");
    delay(3000); //wait 1 sec
    digitalWrite(LedPin, LOW); //led off
    printf("led off\n");

    return 0;
}
```