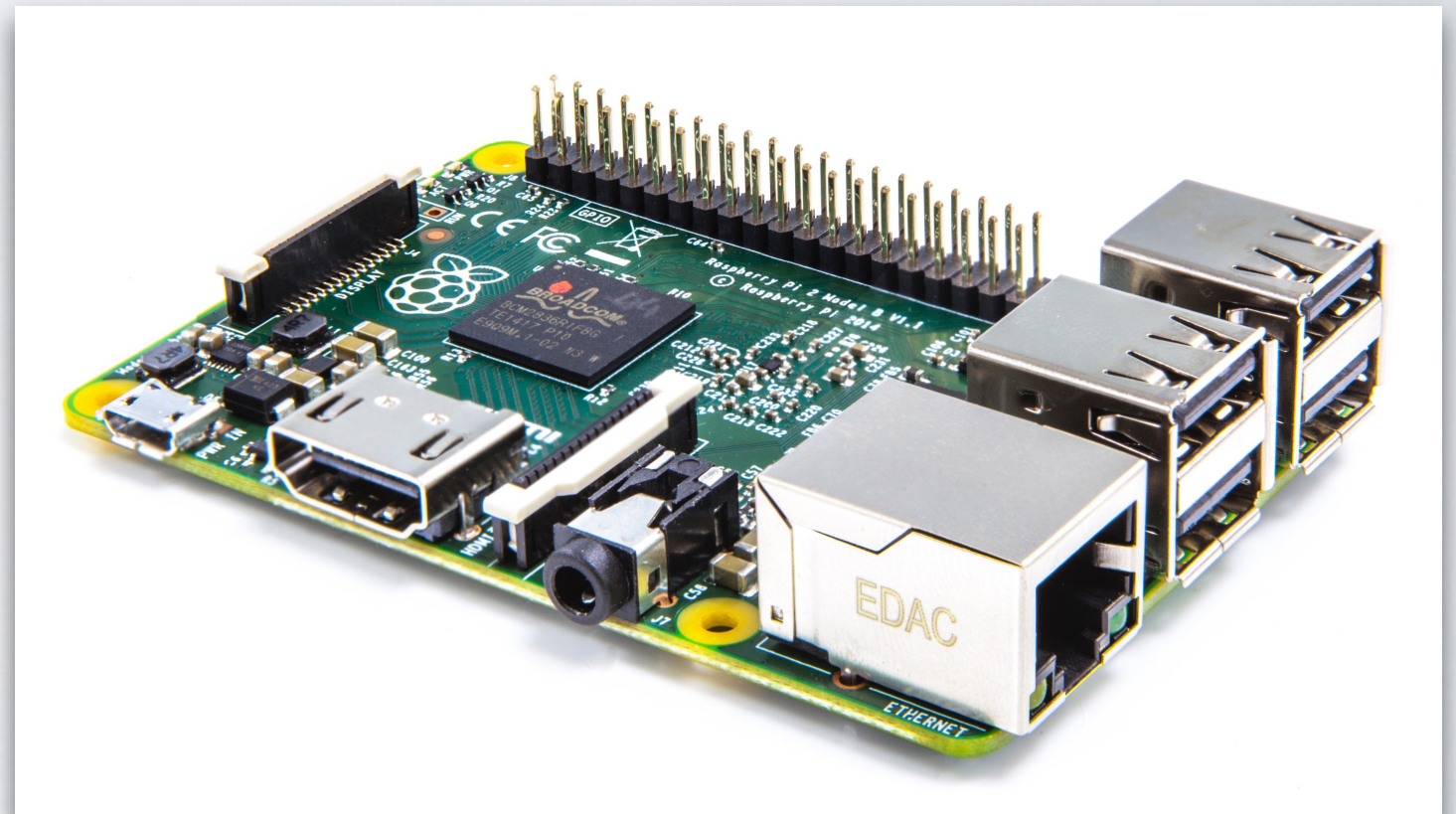


DOMOTIC ROOM

Raspberry & Sensors

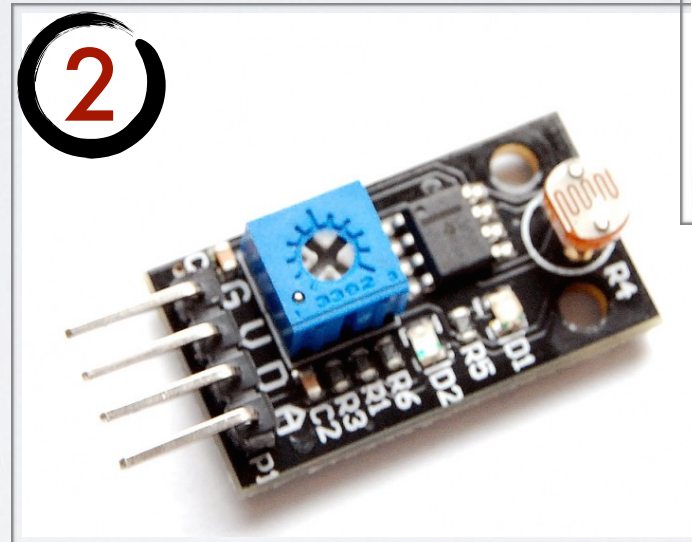
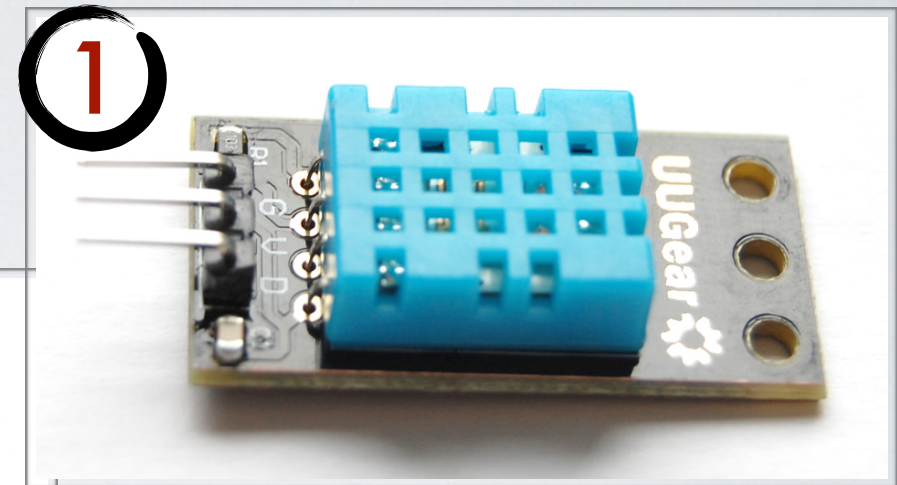
IL RASPBERRY

- Abbiamo utilizzato un raspberry 2



I SENSORI

1. DHT11: Temperatura e Umidità
2. Light: Sensore di luce
3. MQ-2: Sensore di gas
4. PIR: Sensore di movimento



I SENSORI: DETTAGLI 1/2

I sensori utilizzati sono composti del sensore vero e proprio (es: fotoresistore) e di tutte le resistenze e condensatori necessari al suo funzionamento. Grazie a questo e alle 4 uscite di alimentazione del raspberry, è stato possibile evitare l'uso della breadboard.

I SENSORI: DETTAGLI 2/2

Il DHT11, cioè il sensore di temperatura e umidità, invia valori numerici.

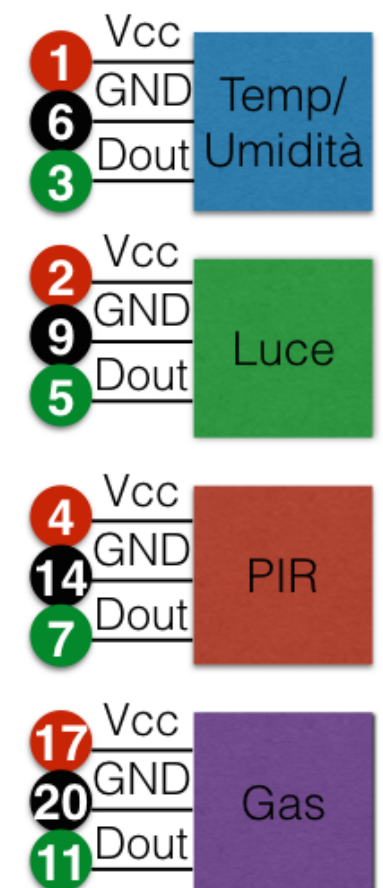
Tutti gli altri sensori sono booleani, cioè possono rappresentare lo stato acceso/spento.

I COLLEGAMENTI

- La Gpio del raspberry mette a disposizione 4 uscite di alimentazione.
- Nel codice è stata utilizzata la logica “board” (fisica)

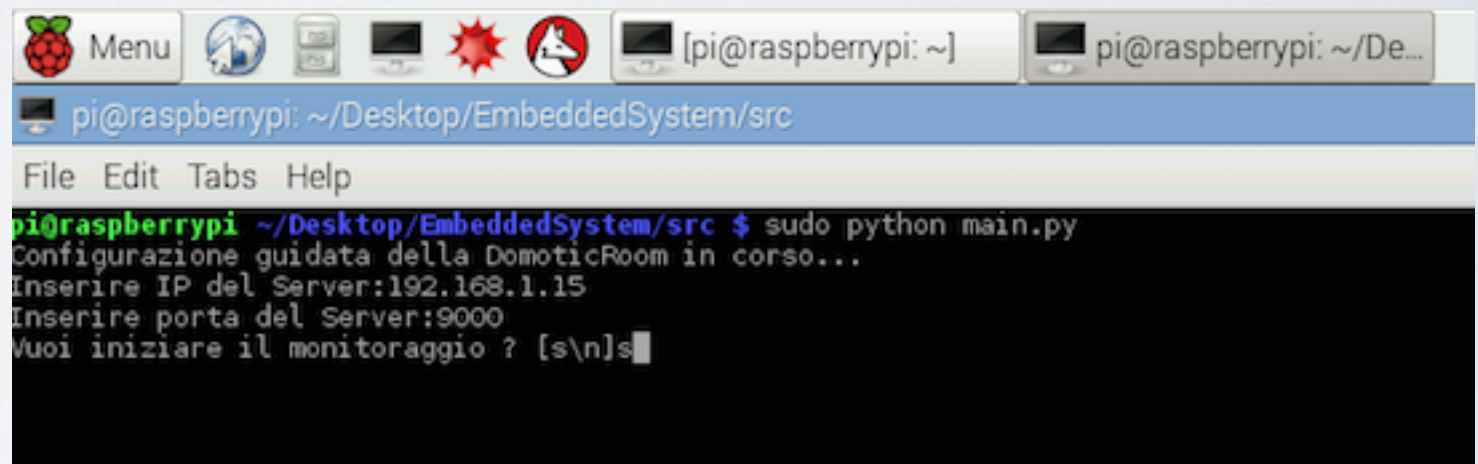
Raspberry Pi B+
B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21



AVVIO DEI SENSORI

1. Load del file .py
2. Inserimento ip del server in ascolto
3. Porta del server
4. Start del processo

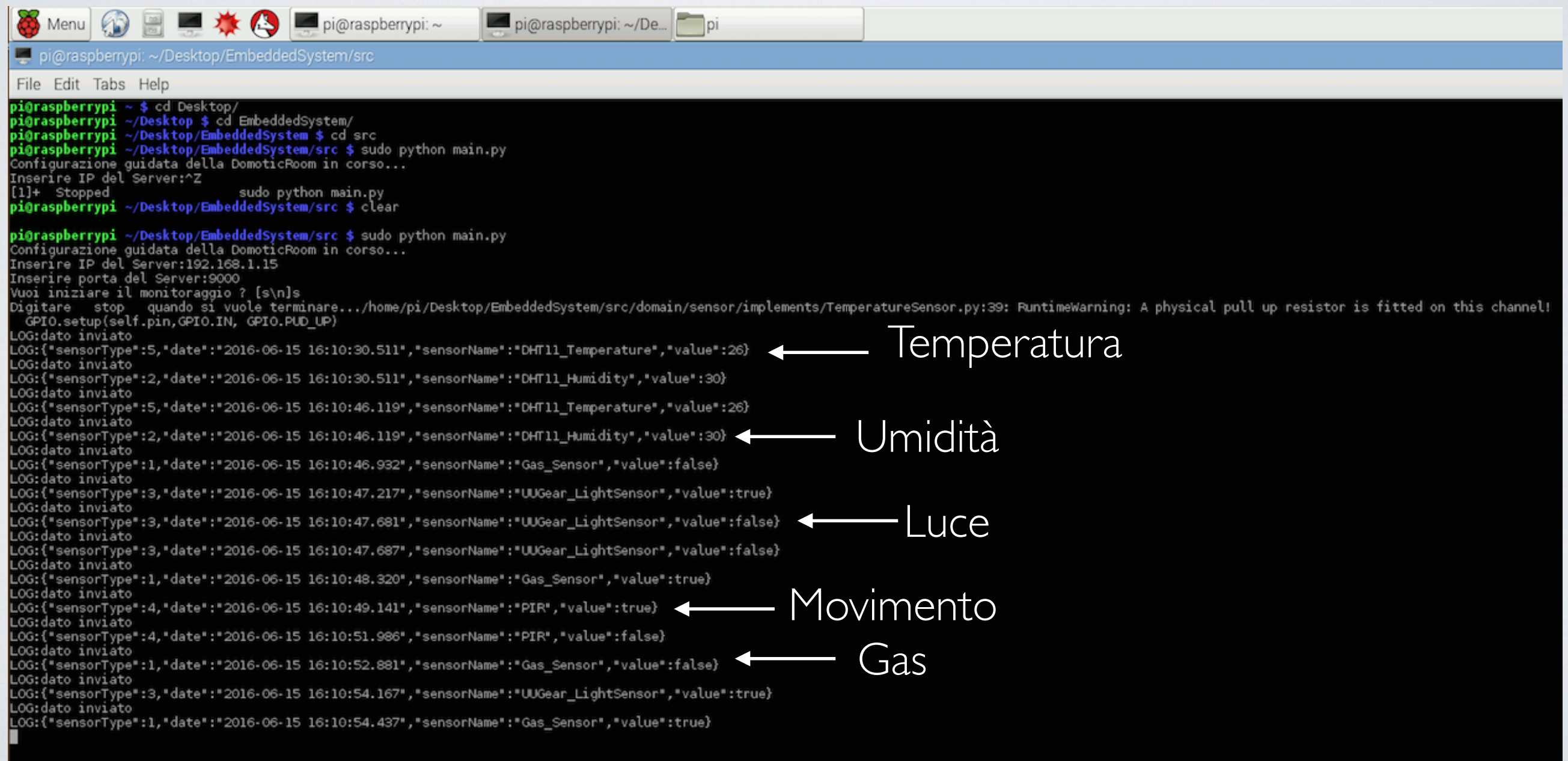


The screenshot shows a terminal window on a Raspberry Pi. The title bar includes a Raspberry Pi logo, a 'Menu' button, and several system icons. The terminal prompt is 'pi@raspberrypi: ~/Desktop/EmbeddedSystem/src'. The user has entered the command 'sudo python main.py'. The script output is in Italian, asking for a guided configuration of the DomoticRoom. It prompts for the server IP (192.168.1.15) and the server port (9000). It then asks if the user wants to start monitoring, with a default 's' (si/sì) response.

```
pi@raspberrypi ~/Desktop/EmbeddedSystem/src $ sudo python main.py
Configurazione guidata della DomoticRoom in corso...
Inserire IP del Server:192.168.1.15
Inserire porta del Server:9000
Vuoi iniziare il monitoraggio ? [s\n]s
```

AVVIO DEI SENSORI

Il programma inizia ad inviare i dati raccolti dai vari sensori



```
pi@raspberrypi: ~/Desktop/EmbeddedSystem/src
File Edit Tabs Help
pi@raspberrypi ~ $ cd Desktop/
pi@raspberrypi ~/Desktop $ cd EmbeddedSystem/
pi@raspberrypi ~/Desktop/EmbeddedSystem $ cd src
pi@raspberrypi ~/Desktop/EmbeddedSystem/src $ sudo python main.py
Configurazione guidata della DomoticRoom in corso...
Inserire IP del Server:^Z
[1]+  Stopped                  sudo python main.py
pi@raspberrypi ~/Desktop/EmbeddedSystem/src $ clear

pi@raspberrypi ~/Desktop/EmbeddedSystem/src $ sudo python main.py
Configurazione guidata della DomoticRoom in corso...
Inserire IP del Server:192.168.1.15
Inserire porta del Server:9000
Vuoi iniziare il monitoraggio ? [s\n]s
Digitare stop quando si vuole terminare.../home/pi/Desktop/EmbeddedSystem/src/domain/sensor/implements/TemperatureSensor.py:39: RuntimeWarning: A physical pull up resistor is fitted on this channel!
GPIO.setup(self.pin,GPIO.IN, GPIO.PUD_UP)
LOG:dato inviato
LOG:{"sensorType":5,"date":"2016-06-15 16:10:30.511","sensorName":"DHT11_Temperature","value":26} ← Temperatura
LOG:dato inviato
LOG:{"sensorType":2,"date":"2016-06-15 16:10:30.511","sensorName":"DHT11_Humidity","value":30} ← Umidità
LOG:dato inviato
LOG:{"sensorType":5,"date":"2016-06-15 16:10:46.119","sensorName":"DHT11_Temperature","value":26}
LOG:dato inviato
LOG:{"sensorType":2,"date":"2016-06-15 16:10:46.119","sensorName":"DHT11_Humidity","value":30} ← Luce
LOG:dato inviato
LOG:{"sensorType":1,"date":"2016-06-15 16:10:46.932","sensorName":"Gas_Sensor","value":false}
LOG:dato inviato
LOG:{"sensorType":3,"date":"2016-06-15 16:10:47.217","sensorName":"UUGear_LightSensor","value":true} ← Movimento
LOG:dato inviato
LOG:{"sensorType":3,"date":"2016-06-15 16:10:47.681","sensorName":"UUGear_LightSensor","value":false} ← Gas
LOG:dato inviato
LOG:{"sensorType":3,"date":"2016-06-15 16:10:47.687","sensorName":"UUGear_LightSensor","value":false}
LOG:dato inviato
LOG:{"sensorType":1,"date":"2016-06-15 16:10:48.320","sensorName":"Gas_Sensor","value":true}
LOG:dato inviato
LOG:{"sensorType":4,"date":"2016-06-15 16:10:49.141","sensorName":"PIR","value":true}
LOG:dato inviato
LOG:{"sensorType":4,"date":"2016-06-15 16:10:51.986","sensorName":"PIR","value":false}
LOG:dato inviato
LOG:{"sensorType":1,"date":"2016-06-15 16:10:52.881","sensorName":"Gas_Sensor","value":false}
LOG:dato inviato
LOG:{"sensorType":3,"date":"2016-06-15 16:10:54.167","sensorName":"UUGear_LightSensor","value":true}
LOG:dato inviato
LOG:{"sensorType":1,"date":"2016-06-15 16:10:54.437","sensorName":"Gas_Sensor","value":true}
```

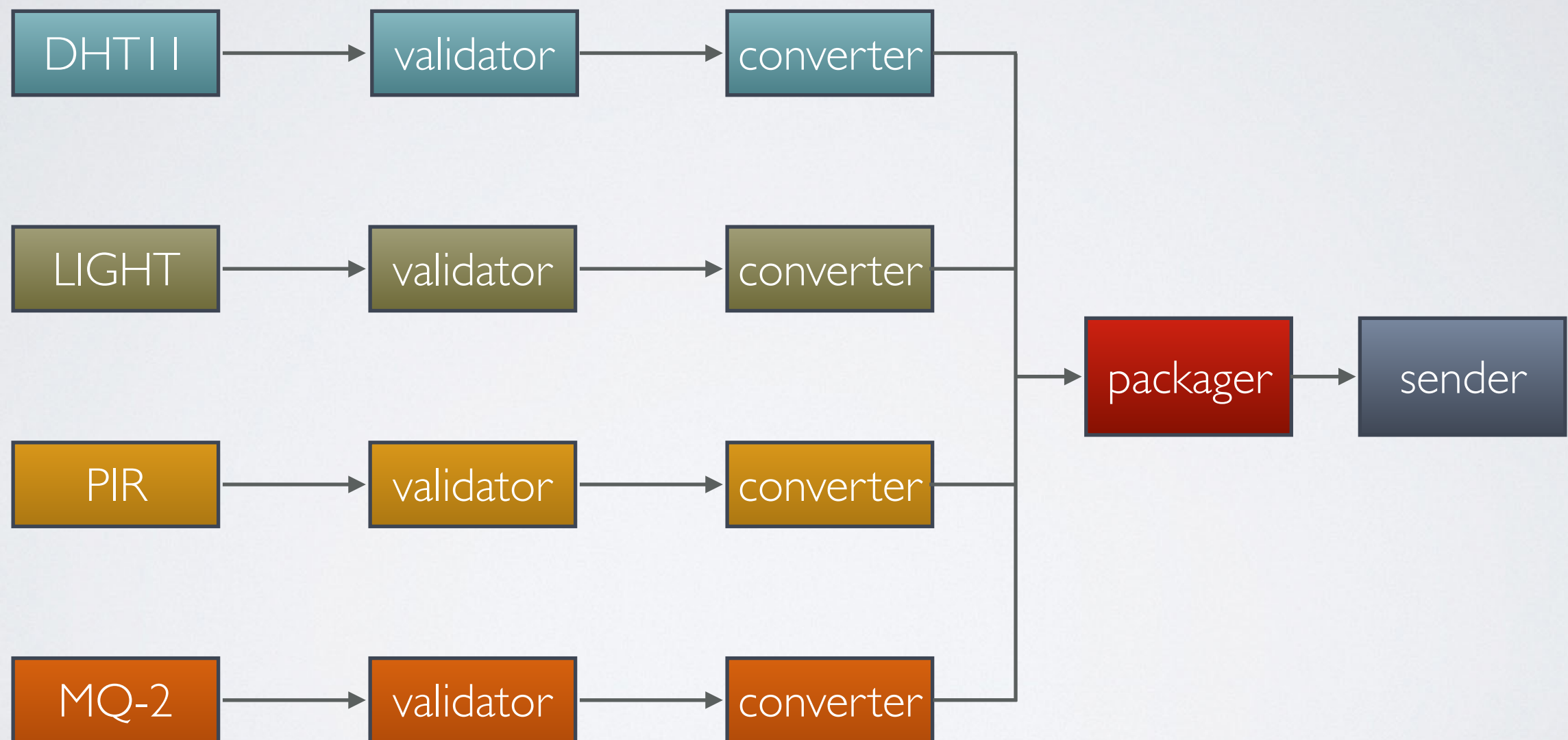

IL PROGRAMMA 1/2

- Scritto in Python
- Utilizzo delle librerie ufficiali per raspberry
Rpi.GPIO
- Utilizzo del paradigma reactive (ogni sensore corrisponde ad uno stream di dati)

IL PROGRAMMA 2/2

- Tutti i sensori, ad eccezione del dht11 funzionano ad interrupt, cioè “si svegliano” solo quando il loro stato cambia. (Il dht11 è da interrogare ogni volta)
- La comunicazione con il server avviene attraverso un pacchetto json, la cui struttura è stata concordata con il server

IL FLUSSO DEI DATI



FINE