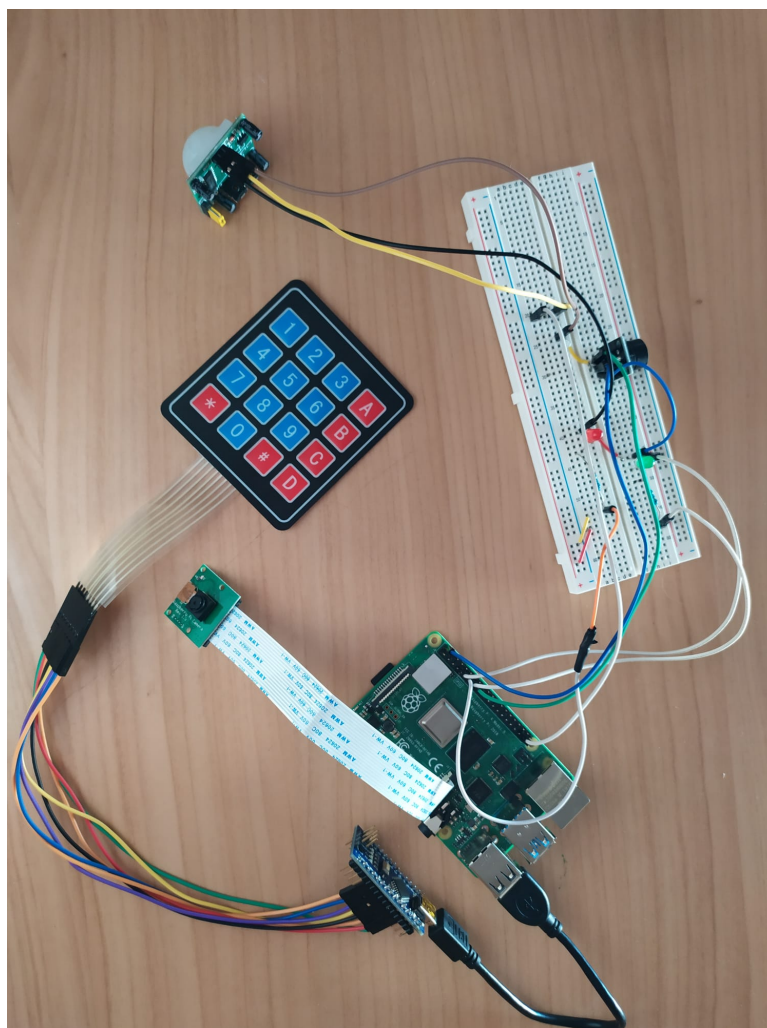


Un implementazione di un sistema antifurto con Arduino e Raspberry Pi

Lorenzo Mustich

21 luglio 2020



Indice

1	Introduzione	3
2	Componenti	4
2.1	HC-SR501 Sensore PIR	4
3	Architettura	5
4	Circuito	6
5	Librerie	6
6	Conversione video	7

1 Introduzione

Il progetto in questione tratta di un'antifurto da appartamento che utilizza un Raspberry Pi4 come unità centrale di elaborazione, un Arduino come controllo dell'input da utente e due bot Telegram per l'attivazione e disattivazione e per l'invio dei video. L'antifurto può essere attivato e disattivato in due modi:

- utilizzando un bot scritto appositamente: **AntiTheftOnOff** (*@ATOnOffBot*);
- nel caso in cui non si abbia la suddetta applicazione, digitando una serie di caratteri alfanumerici per mezzo del tastierino a membrana.

Una volta attivato, un sensore PIR, appositamente tarato, rileverà eventuali movimenti all'interno del suo campo visivo innescando l'accensione di una spia e di un allarme prodotto da un buzzer attivo; successivamente, una videocamera produrrà un filmato di pochi secondi dell'area. Un secondo bot Telegram (**ATVideoBot**, *@ATVideoBot*) è adibito all'invio del video al proprietario della casa.

2 Componenti

Di seguito è mostrata la lista dei componenti:

- Raspberry Pi4;
- Arduino;
- Sensori:
 - HC-SR501, sensore PIR di movimento;
 - Buzzer attivo;
 - PiCamera;
 - Tastiera a membrana 4x4 16 tasti;
- Led rosso/verde;
- Resistenze da ohm e da ohm



Figura 1: Componenti del sistema

2.1 HC-SR501 Sensore PIR

Si tratta di un sensore composto da due slot composte da un materiale sensibile agli infrarossi. Quando è in stato di *idle*, i due slot captano la stessa quantità di raggi; nel momento in cui un corpo caldo entra nella zona d'azione del sensore intercetterà il primo dei due slot creando una differenza di potenziale all'interno

del sensore. Quando il corpo lascia il campo visivo, esso intercetterà il secondo slot portando la differenza di potenziale ad un valore negativo. Il sensore è in grado di rilevare questo cambiamento. Il **sensore HC-SR501** ha un raggio d'azione di 110 gradi conici con una distanza che va dai 3 ai 7 metri. È possibile impostare per quanto tempo l'output del PIR deve essere tenuto alto dopo la rilevazione del movimento: questo intervallo va dai 3 secondi ai 5 minuti.

3 Architettura

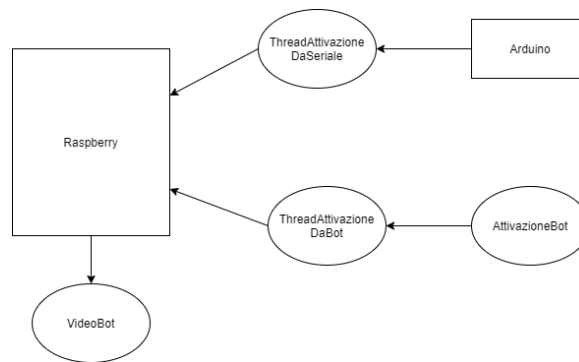


Figura 2: Architettura del sistema

Raspberry ed Arduino comunicano tramite un cavo USB: alla pressione dei tasti corrispondenti alla password viene scritto un carattere su seriale che viene recepito dal Raspberry, avviando l'antifurto. Come spiegato precedentemente, l'attivazione può avvenire in due modi, utilizzando un bot apposito o il tastierino a disposizione. Questi due metodi sono implementati da due Thread che si occupano indipendentemente delle due modalità:

- nel primo caso, il microcontrollore confronta l'input da utente con la password scelta;
- nel secondo caso, l'utente invia dei comandi di attivazione e disattivazione tramite bot.

Per notificare l'avvenuta accensione e spegnimento del sistema è stato posto un led rosso lampeggiante.

4 Circuito

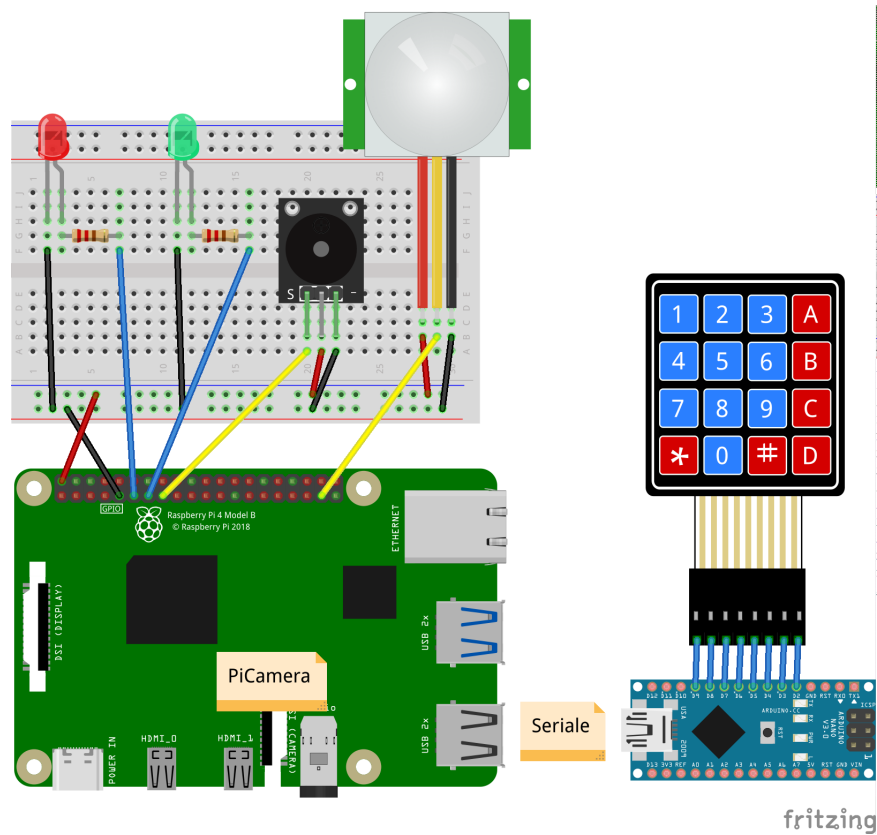


Figura 3: Circuito

5 Librerie

Ai fini del corretto funzionamento del sistema è necessario avere a disposizione le seguenti librerie:

- Python (Raspberry Pi):
 - modulo per bot Telegram: *pip3 install python-telegram-bot*;
 - modulo per la gestione della camera: *pip3 install picamera*;
 - modulo per la lettura da seriale: *pip3 install pyserial*;
 - modulo per la conversione dei video: *pip3 install ffmpeg-python*;
- Arduino:
 - libreria Arduino per la gestione della tastiera: *"Keypad.h"*

6 Conversione video

La PiCamera produce video in soli due formati, .h264 e .mjpeg, che non permettono la loro fruizione da Telegram nella sua versione per smartphone. È stato necessario introdurre un passaggio di conversione nel codice che permettesse l'invio di file.mp4.

```
ffmpeg -y -r 15 -i /home/pi/video.h264 -an -c:v copy /home/pi/video.mp4  
> /dev/null > 2&1
```

Di seguito la spiegazione dei vari parametri:

- -y: sovrascrive i file di output senza chiedere all'utente;
- -r: numero di fps;
- -i: percorso del file di input;
- -an: esclude dal flusso l'audio;
- -c:v copy: non effettua un nuovo encoding sul flusso video in uscita;
- -o: percorso del file di output