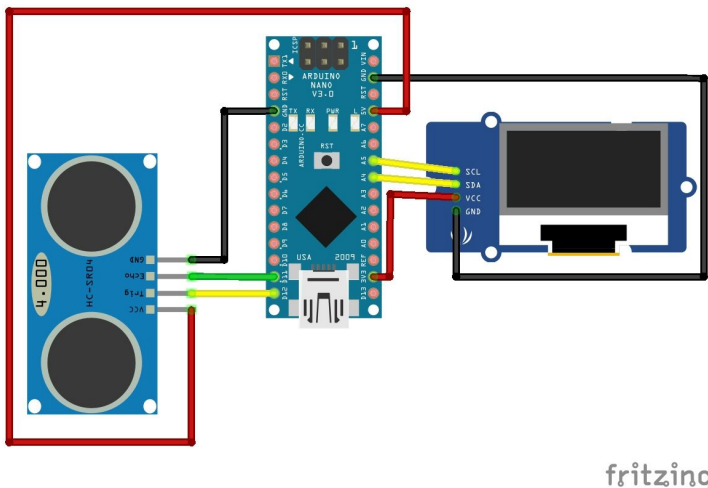


Per l'ultimo tutorial ho deciso di proporvi alcune diverse idee per migliorare il casco.
Un progetto sfrutta arduino, mentre l'altro è stato fatto utilizzando il raspberry.

Per il progetto arduino sono necessari 2 componenti un SR04 ed un Display OLED.
Il componente SR04 è un misuratore di distanze che sfrutta gli ultrasuoni. Emettendo degli impulsi sonori è possibile calcolare il tempo che impiegano a tornare dopo essere rimbalzati sull'ostacolo.
Per poter visualizzare questo dato ho pensato di utilizzare un display OLED, così che possa poi essere installato all'interno del casco per poter avere le informazioni sugli ostacoli nei punti ciechi del casco.
Le connessioni sono state fatte come da schema:



SR04 richiede un'alimentazione di 5V, e 2 pin digitali. uno di input ed uno di output.

Il display OLED funziona grazie al protocollo I2C, e lo controlliamo con i pin SDA ed SCL.

Il codice allegato permette di indicare la distanza tra il sensore ad ultrasuoni ed il primo ostacolo che incontra di fronte a se.

Come sempre sono partito da dei codici di esempio, e li ho combinati assieme per ottenere il risultato voluto.

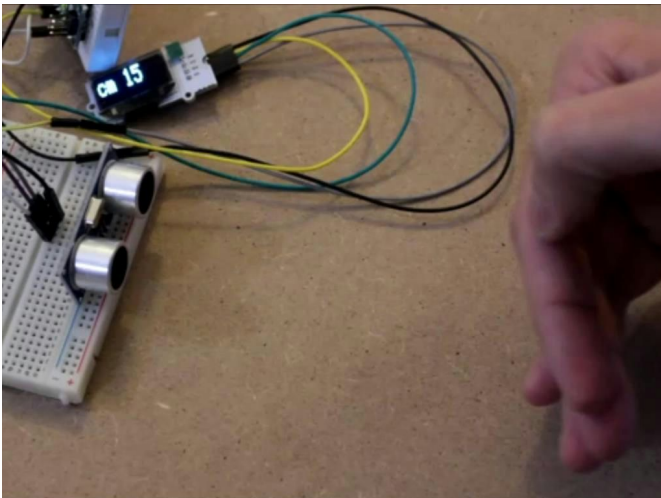
Per la parte del sensore SR04, ho utilizzato la libreria newPing presente in arduino playground:

<http://playground.arduino.cc/Code/NewPing>

Mentre per il display mi sono ispirato agli ottimi esempi e librerie di Adafruit:

<https://learn.adafruit.com/monochrome-oled-breakouts/arduino-library-and-examples>

Il codice risultante è molto semplice grazie al supporto delle librerie. una volta compiute le dovute inizializzazioni si tratta semplicemente di misurare il ping con il sensore ad ultrasuoni, pulire i display e resettare la posizione del cursore, infine scrivere il valore riportato in centimetri.



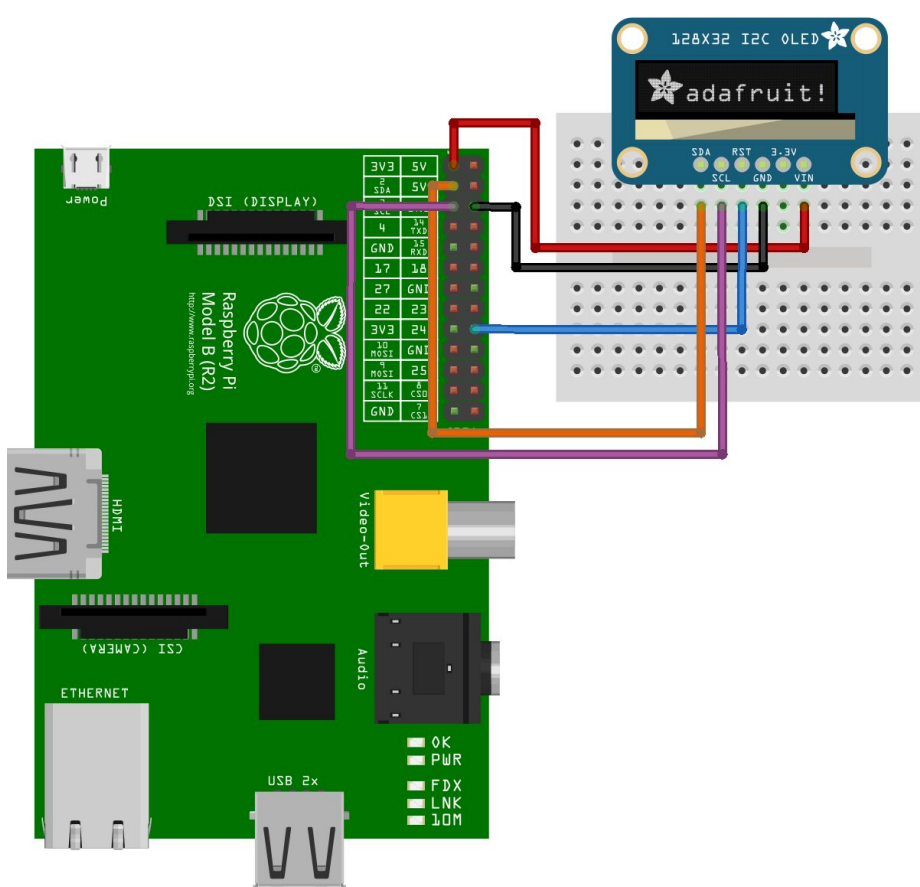
[link al codice arduino]

Ho quindi deciso di creare una versione più avanzata del progetto, ma per fare ciò avevo una maggior potenza di calcolo e di funzionalità. L'idea era quella di fare il riconoscimento del numero delle persone che si avvicinano ed indicarne il numero sempre su un display OLED.

Il miglior candidato a tale funzione è Rasperri pi 2. Si tratta di un vero e proprio mini pc che ci permette così di avere la potenza necessaria di fare il riconoscimento dei volti.

Per la programmazione ho puntato su Python, che è già pre-installato con il sistema operativo raspbian. Anche in questo caso la parte grossa del lavoro le fanno le librerie, soprattutto per quanto riguarda la parte del riconoscimento dei volti, grazie alle librerie OpenCV.

Ho creato il circuito seguendo gli esempi adafruit:



fritzing

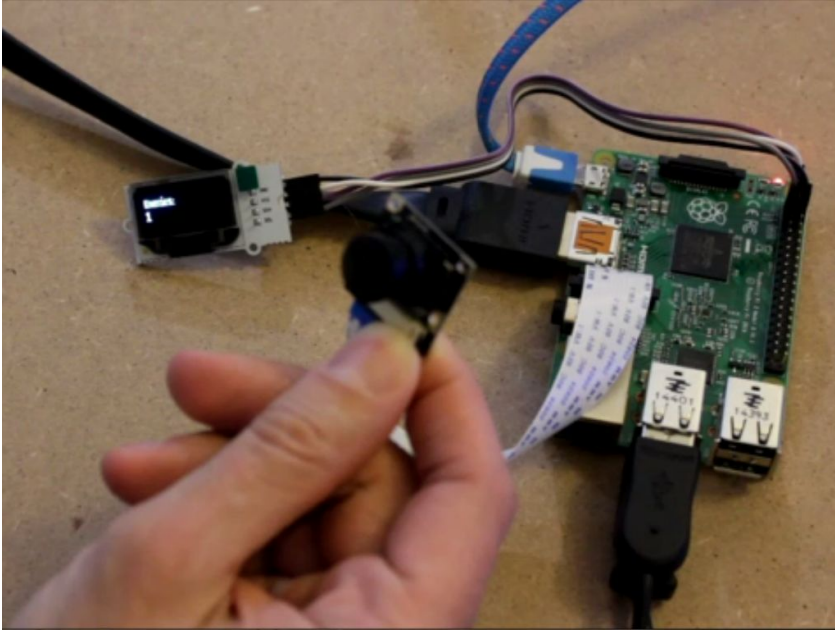
Il mio OLED non presenta il pin di reset, che ho semplicemente ommesso.

Per la parte di acquisizione video ho utilizzato una videocamera che si connette tramite flat cable nell'apposita porta a fianco di quella HDMI.

Una volta inizializzata la nostra SDCard micro con il sistema operativo Raspbian (<https://www.raspberrypi.org/downloads/raspbian/>) mi è bastato seguire le procedure di Adafruit per preparare tutto il necessario per l'utilizzo del display: <https://learn.adafruit.com/adafruit-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>

Come sempre, dopo aver googlato un po' di esempi di openCV, ho fatto il mix per le mie necessità. La parte OLED si gestisce in modo molto simile a quello di Arduino, Mentre il concetto per il riconoscimento dei volti è quello di catturare degli screenshot a cadenza regolare, e li si danno in pasto alle openCV che ci restituiscono il dato desiderato.

[link codice raspberry]



Per concludere Ho apportato una miglioria al casco.

Attualmente uno dei difetti maggiori era dal costante rumore dei servomotori quando il casco viene alimentato.

Questo è dovuto al fatto che i servomotore restano sempre in tensione per mantenere la posizione che gli viene impostata da codice, questo perché altrimenti se sottoposti ad una forza non potrebbero opporre resistenza. Purtroppo questo comporta oltre al rumore fastidioso, un dispendio di energia, che è un grosso problema perchè utilizziamo un' alimentazione a batterie.

Nel caso del nostro caso però abbiamo la fortuna che nelle posizioni di completa apertura e chiusura la gravità non è sufficiente a far muovere la parte frontale.

La modifica è molto semplice, la libreria servo espone il metodo detach che interrompe la richiesta di impostazione ad una posizione. <https://www.arduino.cc/en/Reference/ServoDetach>

Trovate il codice finale in allegato.

[Codice finale arduino]