Creato repository GIT contenente tutto il materiale consultato e prodotto per lo stage.

Ho creato un piccolo script in Bash(*bash.sh*) e un file chiamato *ip.txt* dove viene memorizzato l’IP del Raspberry, lo script in bash memorizza nella variabile **var** l'indirizzo IP attuale, dopodiché con un ciclo while legge il file e confronta l’IP attuale con quello precedentemente memorizzato nel file, se è diverso significa che il Raspberry ha cambiato IP(non è più quello memorizzato nel file) quindi sovrascrive l’indirizzo IP memorizzato precedentemente con quello attuale e manda una mail con l’IP attuale, altrimenti non fa niente.

Visto come funziona il protocollo I2C.

**15/12/19**

Ho collegato la fotocamera, scritto un piccolo script python dove viene scattata una foto ed eventualmente un video.

Scritto script python che invia via email l’IP nel caso in cui cambi(confrontandolo con quello memorizzato in un file ogni 5 minuti e al riavvio del raspberry)

Il file **allegato.py** contiene lo script python che invia anche un file in allegato(da usare eventualmente quando si dovrà inviare la foto alla ricezione della posta)

**24/12/19**

*La cartella AWS contiene tutto il codice che interagisce con quest’ultimo.*

Ho collegato il sensore(funziona perfettamente) e scritto lo script python “**sensor.py** ”contenuto nella cartella AWS , il quale verifica che ci sia il passaggio di un oggetto abbastanza vicino(prossimità >5), nel caso non ci sia fa una sleep di 0.5 secondi e poi controlla nuovamente(rimane nel while fino a quando passa qualcosa abbastanza vicino), nel caso positivo(nel nostro esempio è stata imbucata una lettera) viene scattata una foto con la funzione *capture***\_***image*, dopodichè viene caricata l’immagine nel bucket *S3* di AWS con la funzione *upload\_image* e infine viene inviata l’email con la funzione send\_email per notificare la ricezione della posta con in allegato la foto della persona che la imbuca.

Il file “**add\_images\_to\_s3.py**” carica n foto dal percorso specificato al bucket *S3* con il relativo nome e cognome, dopo di che con il file “**upload\_to\_dynamodb.py**” vengono caricati in *DynamoDb* una coppia chiave-valore per ogni oggetto presente in *S3* dove, la chiave contiene il codice identificativo univoco di un volto, il valore contiene il nome e il cognome della persona nella foto

Adesso rimane da scrivere il codice della funzione lambda la quale, ogni volta che arriva una nuova lettera deve “capire” che è stata caricata una nuova foto in *S3*, quindi deve caricare la coppia chiave-valore corrispondente in *DynamoDb,* e gestire il caso della ricezione della posta da parte di una persona sconosciuta (non presente nel bucket *S3*).

Successivamente bisogna modificare il codice che invia la mail inserendo il nome del mittente nella mail, il quale viene ottenuto grazie al riconoscimento facciale di AWS

Tutto il codice scritto finora è stato testato e funziona correttamente.