

La mia passione per l'automazione

L'automazione, il controllo remoto, la gestione delle immagini e la comunicazione dei dati attraverso l'etere sono argomenti che da sempre mi affascinano; così tre anni fa ho deciso di seguire un corso extrascolastico di programmazione Arduino tenuto nei laboratori di fisica della Scuola, con l'obiettivo di imparare ad utilizzare la board Arduino.

Le prime realizzazioni:



Partecipazione progetto "TU SEI"



*Menzione speciale all' Istituto
Tecnico Tecnologico M. Buonarroti
per il lavoro svolto*



La nostra realizzazione:



Obiettivo

- Gestione accessi di veicoli in area riservata ad alta affluenza.

Soluzioni tradizionali

- Chiave elettronica a contatto, codice di accesso.

Innovazione proposta nel progetto

- Riconoscimento automatico veicoli autorizzati, tramite lettura targa.

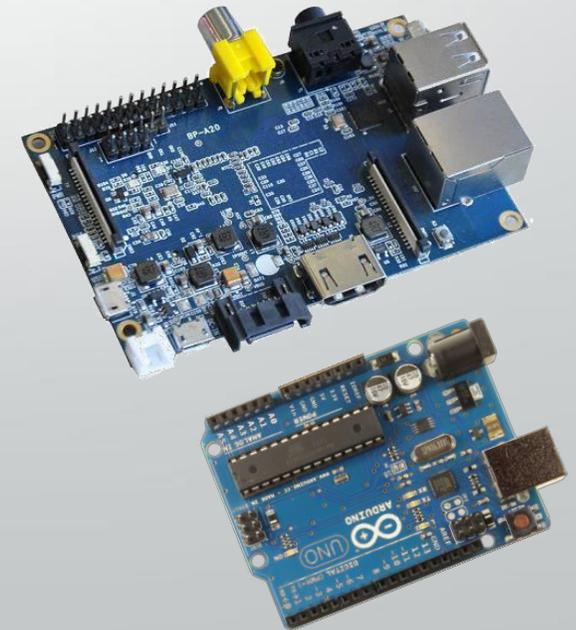


Scelte progettuali

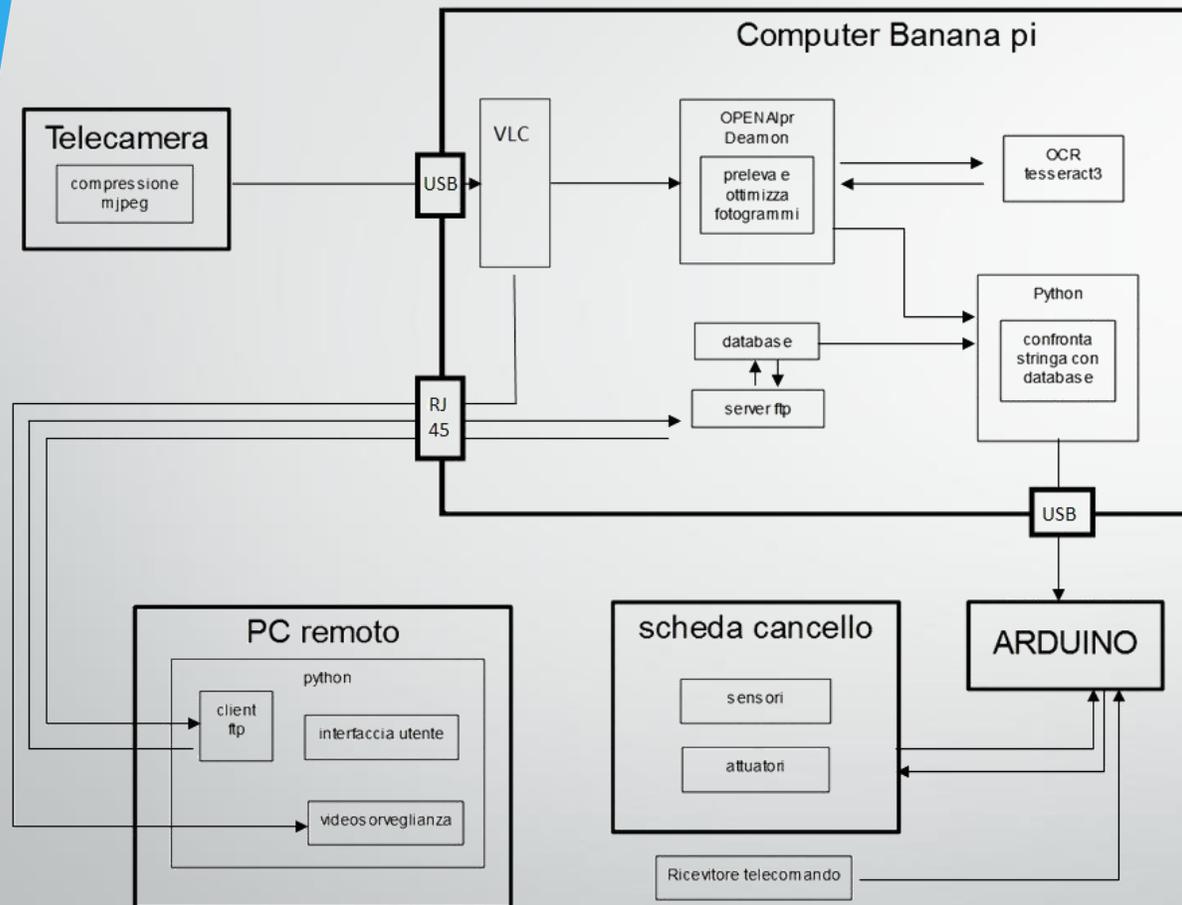
- Funzionamento base del cancello affidato ad **Arduino**;
- Parte di riconoscimento targhe, server e streaming video svolta da **Raspberry Pi**;
- Videosorveglianza e gestione database da **remoto**;

Perché Arduino e Raspberry pi

- Costo ridotto (meno di 50 €)
- Completamente Open Source
- Non dipende da costose licenze di Software Proprietari



Struttura del sistema



- 1. Telecamera:** acquisisce il flusso video
- 2. Banana pi:** (computer) compatto ed economico gestisce il riconoscimento;
- 3. Computer remoto:** gestione del database e per la visualizzazione del flusso video
- 4. Arduino:** (microcontrollore) piccolo ed efficace per il funzionamento del cancello;
- 5. Scheda cancello:** componentistica elettronica necessaria al funzionamento

Linux e il software libero nel nostro sistema

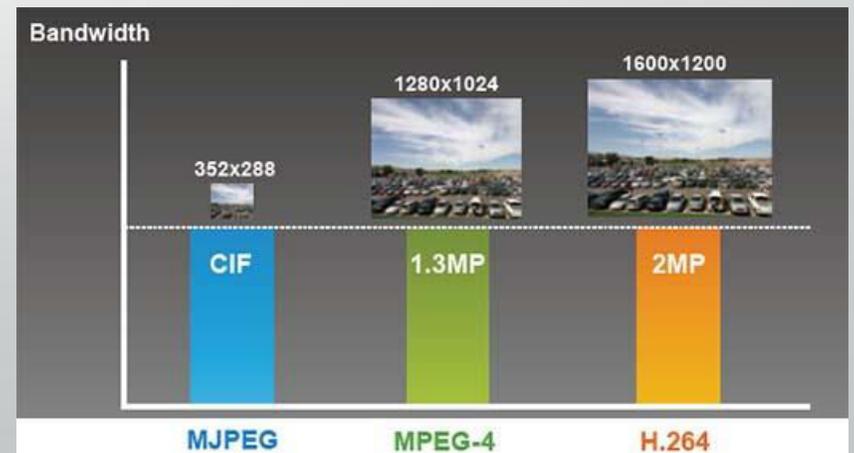
- Abbiamo utilizzato il sistema operativo linux che è distribuito con una licenza che ne permette non solo l'utilizzo da parte di chiunque ed in qualsiasi circostanza, ma anche la modifica, la copia e l'analisi (licenza GPL v3)
- Abbiamo installato e settato su linux il software VLC per creare lo streaming video ed il software OCR per la decodifica del video al fine di trasformare i pixel dell'immagine digitale in testo ASCII della targa.
- Abbiamo creato, utilizzando il linguaggio di programmazione python, il nostro software in grado di gestire i dati provenienti dall'ocr, confrontarli con il database e gestire l'apertura del cancello.
- Abbiamo creato un ulteriore software scritto in Python per gestire da computer remoto, con adeguata interfaccia grafica, il database contenente le targhe e il video in streaming (proveniente dall'hardware Banana Pi)



Telecamera e compressione video MJPEG

Il sistema basato sul protocollo MJPEG acquisisce dal sensore ottico singole foto, le codifica nel formato JPEG e poi le manda in sequenza al client, che visualizza il flusso video come una sequenza di foto.

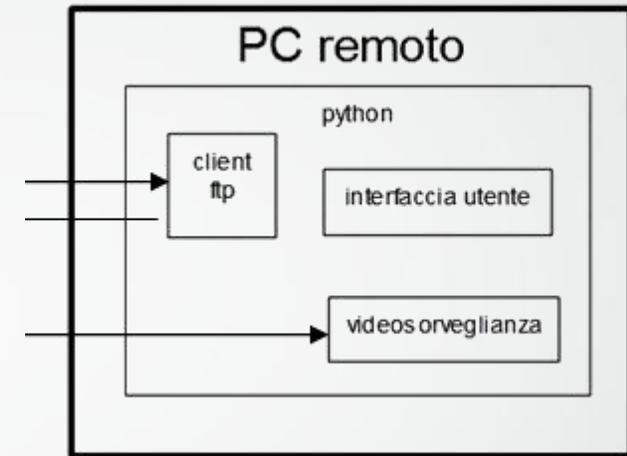
- Vantaggio: grazie alla sua struttura "grezza" il formato MJPEG non richiede processori di alto livello per comporre il flusso e poi per scomporlo nuovamente in singoli fotogrammi per l'analisi.
- Svantaggio: a parità di qualità di immagine, la rete viene utilizzata anche 20 volte di più rispetto ai protocolli più avanzati che creano una compressione intraframe (MPEG2, H264).



Computer remoto

Obiettivo:

Fornire un interfaccia con l'utente rendendo possibile la gestione del database e la visualizzazione del flusso video della telecamera



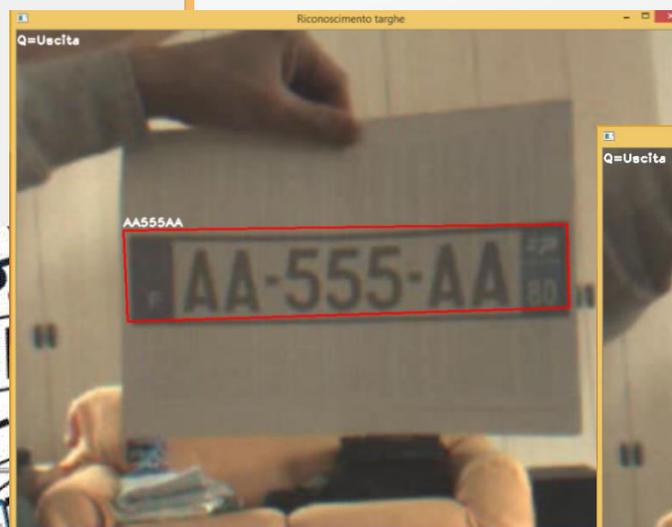
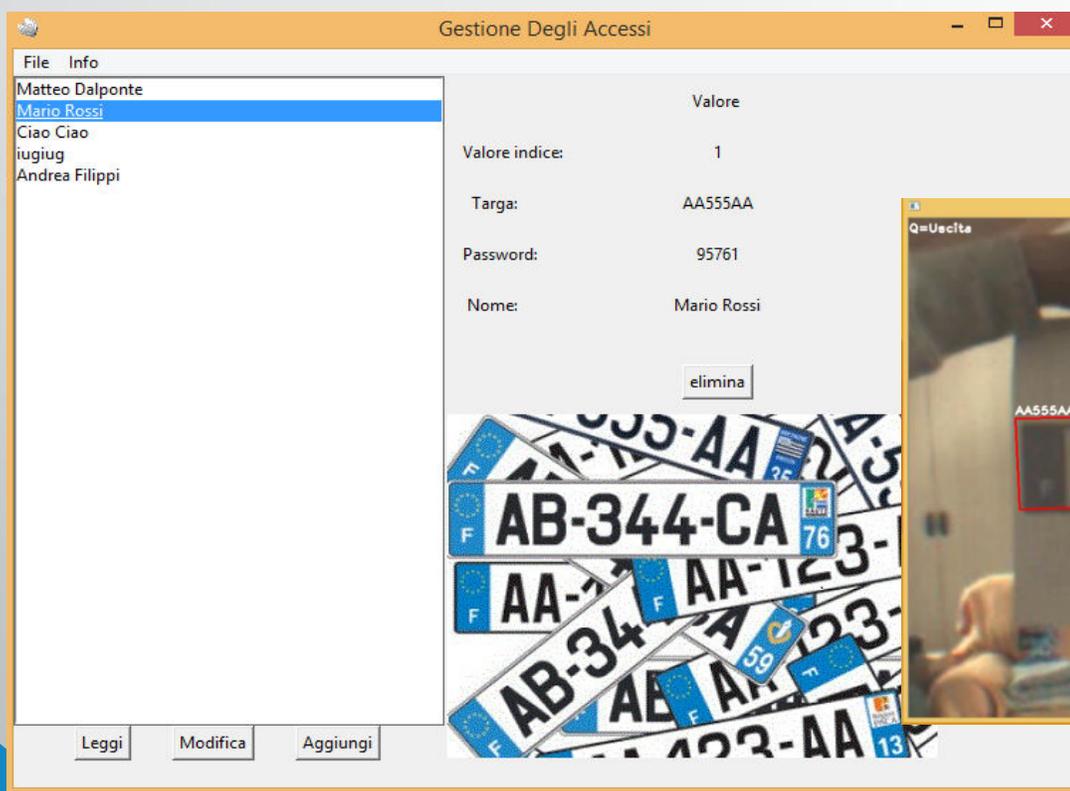
Funzionamento

- Lettura/scrittura database attraverso una connessione di Rete (LAN o Wireless) con protocollo FTP
- Software Python per l'interfaccia utente
- Software Python per la lettura dello streaming MJPEG e la visualizzazione del video



L'interfaccia grafica per la gestione da computer

Tramite queste due interfacce grafiche (GUI) è possibile aggiungere, eliminare o modificare le targhe presenti nel database remoto e visualizzare il video per la sorveglianza del cancello



Un estratto esemplificativo della nostra programmazione

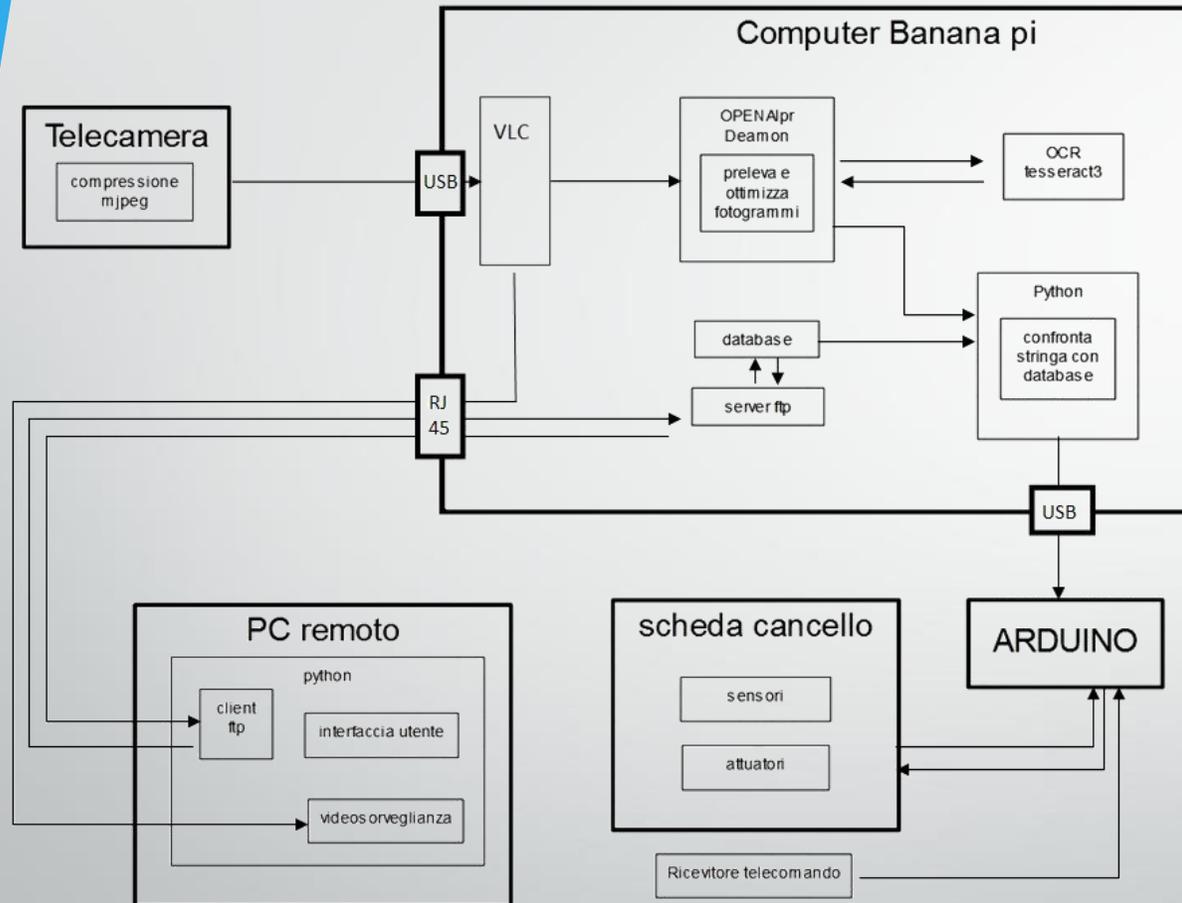


In questa parte del programma, scelta a titolo di esempio, i 40 tentativi di riconoscimento prodotti dall' OCR vengono confrontati con tutte le targhe contenute nel database. Nel caso in cui vi sia corrispondenza viene inviato il comando di apertura ad Arduino

```
1.  for num in range(0,40) :
2.      try:
3.          try:
4.              number lines = len(dati)
5.              for i in range(0, number lines):
6.                  diz1 = eval(dati[i])
7.                  targa diz = diz1['targa']
8.                  if (str(d['results'][0]['candidates'][num]['plate']) ==
targa diz):
9.                      try:
10.                          ser.write("1")
11.                      except:
12.                          pass
13.                      valstanga = 1
14.                      nome = diz1['nome']
15.                      print "          Consento l'accesso!      ", time, " ms ",
nome
16.
17.
18.          except:
19.              pass
20.
21.      except:
22.          pass
```



Struttura del sistema



- 1. Telecamera:** acquisisce il flusso video
- 2. Banana pi:** (computer) compatto ed economico gestisce il riconoscimento;
- 3. Computer remoto:** gestione del database e per la visualizzazione del flusso video
- 4. Arduino:** (microcontrollore) piccolo ed efficace per il funzionamento del cancello;
- 5. Scheda cancello:** componentistica elettronica necessaria al funzionamento

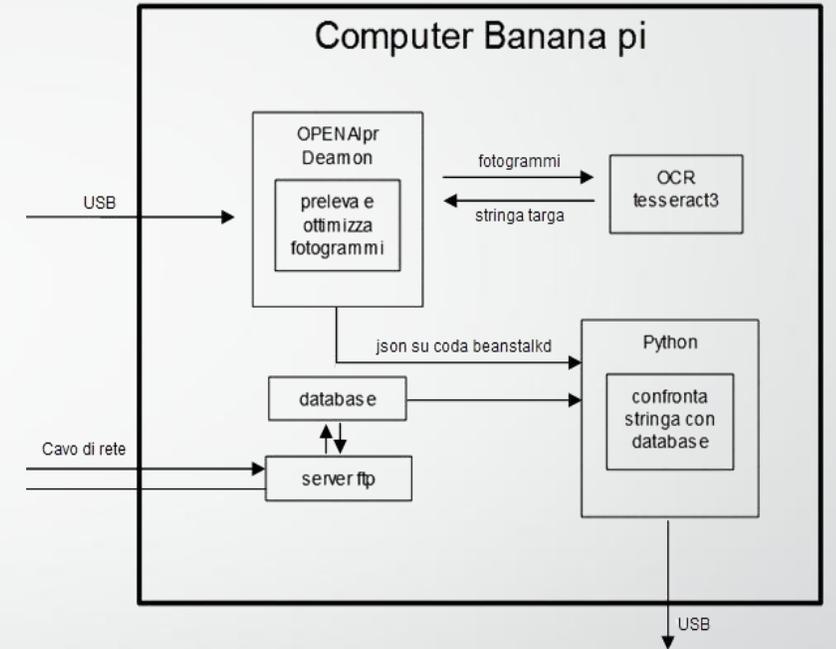
Banana Pi

Obiettivo

Elaborazione fotogrammi, confronto testo con database e comando apertura

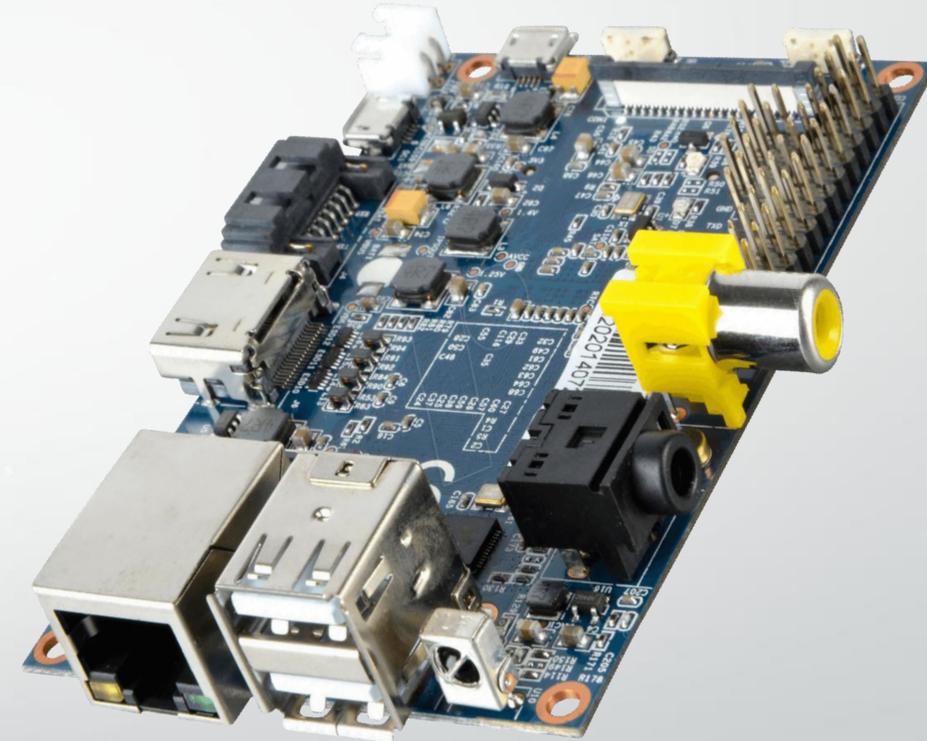
Funzionamento

- Software che preleva i fotogrammi dal flusso video e li ottimizza (**alprDeamon**)
- Software di riconoscimento ottico dei caratteri OCR (**TesseractOcr**)
- File **database** condiviso con il computer remoto in cui sono presenti le targhe consentite
- **Software Python** che confronta le targhe con il database e comanda l'apertura se consentita.



Cos' è Banana Pi

- E' un "**single-board**" computer
- Supporta il sistema operativo Lubuntu (distro Linux)
- Integra una porta di rete LAN permettendo la creazione di un sistema in grado di collegarsi alla rete



a
andrea andrea andrea

OpenAlpr

Obiettivo

Prelevare fotogrammi dal flusso video, ottimizzarne contrasto, luminosità, dimensione e colori per l'invio al software di riconoscimento ottico.

Creazione di una coda di lavoro su cui vengono salvati i dati in struttura json pronti per essere letti dal nostro software

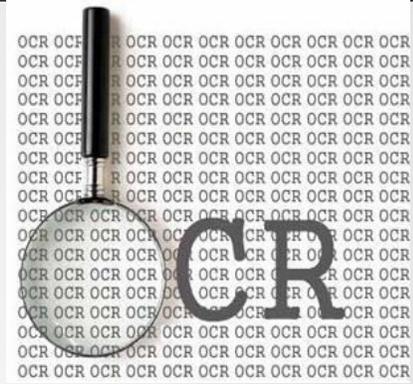
Cos'è

OpenALPR è una libreria finalizzata alla creazione di un Software adibito al riconoscimento delle targhe dei veicoli

La coda di lavoro Beanstalk e la struttura dati json

- **Beanstalk** ha l'obiettivo di trasferire informazioni in modo sicuro fra due applicazioni distinte
- **Json** rappresenta una struttura standard per la memorizzazione dei dati

TesseractOcr



Obiettivo

Analizzare i fotogrammi ricevuti da openAlpr e restituire un dato digitalizzato del testo

Funzionamento

Un software di riconoscimento ottico si basa sull' apprendimento, fornendo degli esempi tipici di targhe e il corrispettivo contenuto, il Software crea delle possibili relazioni che in seguito consulterà.

Il riconoscimento avviene per *step*:

- Pulizia immagine da disturbi
- Individuazione aree di testo
- Isolamento singoli caratteri e tentativo di ricostruzione di caratteri rovinati
- Processo di riconoscimento con risultato probabilistico



File Database

Obiettivo

Memorizzare in modo efficace i dati, permettere una facile futura implementazione con nuovi attributi relativi ad ogni targa

Struttura

Esempio di dato memorizzato:

```
{'targa':'INA1043', 'nome':'Andrea Filippi', 'password':741852}
```

Funzione per ricavare il dato riferito alla chiave targa:

```
diz['targa']
```

Software Python

Obiettivo

Lettura dei dati ricevuti dall' OCR, confronto con database e comunicazione seriale con arduino

Cosa fa

- Avvia il processo Alpr e l' OCR
- Apre la comunicazione seriale con arduino
- Avvia la lettura nella coda di lavoro Beanstalk e decodifica i dati
- Confronta le molteplici targhe ricevute in ordine di probabilità e se una è presente nel database comanda l' apertura



Arduino



Obiettivo

Gestire il funzionamento ordinario del cancello, garantendone l' esercizio autonomo anche in assenza della parte di riconoscimento targhe.

Cos' è Arduino

Piattaforma Open Source composta da un microcontrollore Atmel programmabile.

E' completamente Italiana, piccola ed economica.

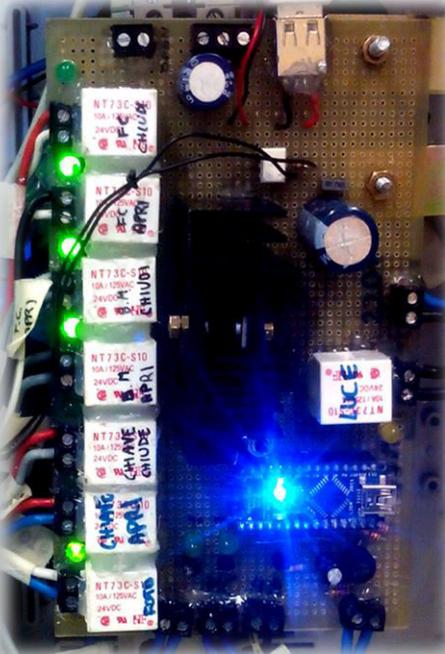
Funzionamento

- Ingressi con indice di priorità al fine di garantire la **sicurezza** del cancello
- Programma eseguito a **ciclo continuo** per evitare "momenti ciechi"
- Motore spento per mezzo secondo nelle inversioni di marcia per evitare sovracorrenti nel motore

Scheda cancello

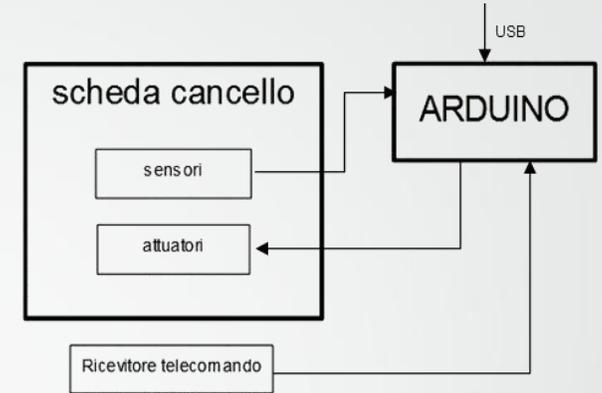
Obiettivo

- *Interfacciare* Il microcontrollore Arduino con la sensoristica
- Azionare *parti di potenza* come il motore e la luce di attenzione
- Fornire *l'alimentazione* a tutta l'elettronica presente.



Funzionamento

- I sensori presenti eccitano le bobine dei relè, Arduino effettua la lettura sui rispettivi contatti.
- Per azionare la parte di potenza (motore monofase) si utilizzano dei transistor che vanno ad eccitare le bobine dei relè
- L'alimentazione è ottenuta raddrizzando 18V in corrente alternata ricevuti da un trasformatore attraverso un ponte di Graetz e successivamente stabilizzati a 5V in due *step* con un 7812 e un 7805, grazie a tre condensatori si riduce il ripple.



Ulteriori punti di forza del progetto

- Offre potenziali **sviluppi applicativi**;
 - Riconoscimento ottico oggetti
 - Gestione remota automazioni industriali
 - Gestione e trasmissione dati
- È completamente **libero**;
- Accessibile a **tutti**;
- Sviluppato su **piattaforme dinamiche** al passo con la tecnologia
- È un'iniziativa **multidisciplinare**;

Partecipazione progetto "TU SEI"



*Menzione speciale all' Istituto
Tecnico Tecnologico M. Buonarroti
per il lavoro svolto*



Trento, 22 maggio 2015