ALLEGATI

SOFTWARE ARDUINO

```
////////// seriale  //////////
int serial = 0;
////////// input  //////////
int chiaveA = A1;
int chiaveC = 2;
int fcA = 3;
int fcC = 4;
int bmA = 5;
int bmC = 6;
int foto = 7;
int stopE = 8;
int plateRecognition = 9;
////////// output  //////////
int mA = 12;
int mC = 11;
int luce = 10;

int ma[5] = {0,0,0,0};
int mc[5] = {0,0,0,0};
int lum[5] = {0,0,0,0,0};
//timer inversione marcia cw/ccw
long tim00=0;
long tim01=0;
int stopTime = 500;
//timer autochiusura
long tim02=0;
int autoCloseTime = 2000;
//blocco d'emergenza
boolean stopEm =1;

void setup() {
Serial.begin(9600);

////////// input  //////////
pinMode(chiaveA,INPUT_PULLUP);
pinMode(chiaveC,INPUT_PULLUP);
pinMode(fcA,INPUT_PULLUP);
pinMode(fcC,INPUT_PULLUP);
pinMode(bmA,INPUT_PULLUP);
pinMode(bmC,INPUT_PULLUP);
pinMode(foto,INPUT_PULLUP);
pinMode(stopE,INPUT_PULLUP);
pinMode(plateRecognition,INPUT_PULLUP);
////////// output  //////////
pinMode(mA,OUTPUT);
pinMode(mC,OUTPUT);
pinMode(luce,OUTPUT);
pinMode(13,OUTPUT);

delay(1000);
}

void loop() {
//////////  SERIALE  //////////
serial = Serial.read();
if(serial==49){
  Serial.flush();
  digitalWrite(13,HIGH);
  ma[0] = 1;
  mc[0] = 0;
```

```
}

if (digitalRead(plateRecognition)==0){  //comando apertura con riconoscimento
targhe
  ma[0] = 1;
  mc[0] = 0;
}

if (digitalRead(chiaveA) == 0){ //comando apertura con chiavi
  ma[0] =1;
  mc[0] = 0;
}

if (digitalRead(chiaveC) == 0){ //comando chiusura con chiavi
  mc[0] = 1;
  ma[0] = 0;
}
if(digitalRead(foto)==1 && mc[0]==1){ //fotocellula solo sulla chiusura
 ma[0] = 1;
 mc[0] = 0;
}

if(digitalRead(bmC) == 1 && mc[0]==1){  //barriera mobile sulla chiusura
 mc[0]=0;
 ma[0]=1;
}

if(digitalRead(bmA) == 1 && ma[0]==1){  //barriera mobile sull'apertura
 ma[0]=0;
 mc[0]=1;
}

if(digitalRead(fcA) == 0){  //fotocellula fine apertura che avvia timer
autochiusura
if(ma[0]==1){
  tim02 = millis();
  }
   ma[0] =0;
}

if((millis()-tim02)>autoCloseTime && (millis()-tim02)<(autoCloseTime+100) &&
millis()>(autoCloseTime+100)){  //autochiusura dopo tot millisecondi
  tim02 = tim02 - 1000;  //tolgo tempo per evitare che Arduino rientri più volte
  mc[0] = 1;
}

if(digitalRead(fcC) == 0){  //fotocellula fine chiusura
  mc[0] =0;
}

if(digitalRead(stopE)==1){  //pulsante d'emergenza
  mc[0] = 0;
  ma[0] = 0;
  lum[0] = 0;
}

if(ma[2]==1 || mc[2]==1 ){
  lum[0]=1;  //accendiamo la luce lampeggiante insieme ai motori
  stopEm = 1;  //se un motore è viene acceso dopo lo stop di emergenza
riabilitiamo il sistema
}else{
  lum[0]=0;
}
```

```
////////// RIAPRE  //////////
if (ma[0] > ma[1] && mc[0] < mc[2]){ //è stato invertito il senso di marcia dopo
tot tempo riAPRE
tim00 = millis();
//delay(1);
}
if((millis()-tim00)<stopTime){  //prima di invertire la marcia spegni tutto per
un tot di tempo
if(digitalRead(stopE)==1){ //rimani pronto se fosse premuto il pulsante di stop
d'emergenza
stopEm =0;
}
ma[0] = 0;
mc[0] = 0;
lum[0] = 1;
}else if ((millis()-tim00)<(stopTime+100) && (millis()-tim00)>stopTime && stopEm
==1){  //fai riaprire la stanga
ma[0] = 1;
tim00 = tim00 - 101;
}
////////// RICHIUDE  //////////
if (mc[0] > mc[1] && ma[0] < ma[2]){ //è stato invertito il senso di marcia dopo
tot tempo richiude
tim01 = millis();
//delay(1);
}
if((millis()-tim01)<stopTime){
if(digitalRead(stopE)==1){
stopEm =0;
}
ma[0] = 0;
mc[0] = 0;
lum[0] = 1;
}else if ((millis()-tim01)<(stopTime+100) && (millis()-tim01)>stopTime && stopEm
==1){
mc[0] = 1;
tim01 = tim01 - 101;
}

////////// SHIFT  //////////
for(int i=0; i<4; i++){
  ma[(4-i)] = ma[(3-i)];
  mc[(4-i)] = mc[(3-i)];
  lum[(4-i)] = lum[(3-i)];
  }

//////////  SCRITTURA USCITE  /////////
if (ma[0] == mc[0]) { //evitiamo corti piuttosto spegniamo tutto
  ma[0] =0;
  mc[0] =0;
}

if (ma[0] == ma[4]){ //per qualche ciclo le variabili possono essere oscillanti
se per 5 cicli rimangono costanti allora possiamo accendere il relè!
digitalWrite(mA,ma[0]);
}
if (mc[0] == mc[4]){
digitalWrite(mC,mc[0]);
}
if (lum[0] == lum[4]){
digitalWrite(luce,lum[0]);
}
}
```

SOFTWARE BANANA PI

```
import serial
import beanstalkc
import json
import os

immx=0
immy=0
buffer=0
valstanga = 0

open_alprd = "/home/bananapi/openalpr/src/build/alprd --config /etc/openalpr"
os.system("pkill alprd")
os.system(open_alprd)

try:
        ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
except:
        try:
                ser = serial.Serial('/dev/ttyUSB1', 9600, timeout=1)
        except:
                try:
                        ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
                except:
                        pass

beanstalk = beanstalkc.Connection(host='localhost', port=11300)
beanstalk.watch('alprd')

while True:

        f = open('/home/bananapi/Desktop/Database.conf','r')
        s=str(f.readlines())
        dati=eval(s)

        job = beanstalk.reserve()
        jobb = job.body
        job.delete()
        d = json.loads(jobb)

        time = str(d["processing_time_ms"])
        targa = str(d["results"][0]["plate"])
        x1 = str(d['results'][0]['coordinates'][0]['x'])
        y1 = str(d['results'][0]['coordinates'][0]['y'])
        x2 = str(d['results'][0]['coordinates'][1]['x'])
        y2 = str(d['results'][0]['coordinates'][1]['y'])
        x3 = str(d['results'][0]['coordinates'][2]['x'])
        y3 = str(d['results'][0]['coordinates'][2]['y'])
        x4 = str(d['results'][0]['coordinates'][3]['x'])
        y4 = str(d['results'][0]['coordinates'][3]['y'])
        immx = str(d['img_width'])
        immy = str(d['img_height'])


        for num in range(0,40):
                try:
```

```
                        try:
                                number_lines = len(dati)
                                for i in range(0, number_lines):
                                        diz1 = eval(dati[i])
                                        targa_diz = diz1['targa']
                                        if (str(d['results'][0]['candidates'][num]['plate']) == targa_diz):
                                                try:
                                                        ser.write("1")
                                                except:
                                                        pass
                                                valstanga = 1
                                                nome = diz1['nome']
                                                print "                    Consento l'accesso!     ",
time," mS    ", nome



                        except:
                                pass

                except:
                        pass



        trasm = dict()
        trasm["targa"]=nome
        trasm["tempo"]=time
        trasm["valstanga"]=valstanga
        trasm["x1"]=x1
        trasm["y1"]=y1
        trasm["x2"]=x2
        trasm["y2"]=y2
        trasm["x3"]=x3
        trasm["y3"]=y3
        trasm["x4"]=x4
        trasm["y4"]=y4
        trasm["immx"]=immx
        trasm["immy"]=immy

        f1 = open('/home/bananapi/Desktop/trasmissione.conf','w')
        f1.write (str(trasm))
```

PROGRAMMA PER LA VISUALIZZAZIONE REMOTA DEL VIDEO

```
import cv2
import urllib
import numpy as np
import ctypes
from ftplib import FTP
import time

x1=0
x2=0
x3=0
x4=0
y1=0
y2=8
y3=0
y4=0
immx=0
immy=0
immx1=500
immy1=500
tempopresente=0
tempopassato=0
valstanga=0



stream=urllib.urlopen('http://192.168.0.44:8080/video')
ftp = FTP('192.168.0.44', 'bananapi', 'bananapi', timeout=2)


bytes=''
while True:
    tempopresente=time.time()
    if (tempopresente-tempopassato)>0.2:
        tempopassato=tempopresente

        try:
            temp=open('temp.conf','w')
            ftp.retrbinary('RETR /home/bananapi/Desktop/trasmissione.conf', temp.writelines)
            temp.close()
            f = open('temp.conf','r')
            trasm=eval(f.read())

            user32 = ctypes.windll.user32
            Yval=int(user32.GetSystemMetrics(1)*0.66)
            rapp_trasf=((user32.GetSystemMetrics(1)*0.66)/(eval(trasm['immy'])))
            Xval=int((eval(trasm['immx']))*rapp_trasf)

            targa=trasm['targa']
            x1=int((eval(trasm['x1']))*rapp_trasf)
            y1=int((eval(trasm['y1']))*rapp_trasf)
            x2=int((eval(trasm['x2']))*rapp_trasf)
            y2=int((eval(trasm['y2']))*rapp_trasf)
            x3=int((eval(trasm['x3']))*rapp_trasf)
            y3=int((eval(trasm['y3']))*rapp_trasf)
            x4=int((eval(trasm['x4']))*rapp_trasf)
            y4=int((eval(trasm['y4']))*rapp_trasf)
            immx=int((eval(trasm['immx']))*rapp_trasf)
            immy=int((eval(trasm['immy']))*rapp_trasf)
            valstanga=(trasm['valstanga'])

        except:
```

```
            pass


        bytes=stream.read(1024)
        a = bytes.find('\xff\xd8')
        b = bytes.find('\xff\xd9')
        if a!=-1 and b!=-1:
            jpg = bytes[a:b+2]
            bytes= bytes[b+2:]
            i = cv2.imdecode(np.fromstring(jpg, dtype=np.uint8),cv2.CV_LOAD_IMAGE_COLOR)

            ridimensiona=cv2.resize(i, (Xval,Yval), fx=5, fy=5)

            pts = np.array([[x1,y1],[x2,y2],[x3,y3],[x4,y4]], np.int32)

            if valstanga==0:
                cv2.polylines(ridimensiona,[pts],True,(0,0,255),2)
            else:
                cv2.polylines(ridimensiona,[pts],True,(0,255,0),2)

            cv2.putText(ridimensiona,targa, (x1,y1-5), cv2.FONT_HERSHEY_PLAIN, 1.2,(255,255,255),2)
            cv2.putText(ridimensiona,"Q=Uscita", (0,20), cv2.FONT_HERSHEY_PLAIN, 1.2,(255,255,255),2)


            cv2.imshow("Riconoscimento targhe",ridimensiona)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break


    cv2.destroyAllWindows()
```

PROGRAMMA PER LA GESTIONE REMOTA DEL DATABASE

```
from Tkinter import *
import tkMessageBox
import os
from ftplib import FTP
from StringIO import StringIO
import subprocess

def aggiornolista():
    global lista

    temp=open('temp.conf','w')
    ftp.retrbinary('RETR /home/bananapi/Desktop/Database.conf', temp.writelines)
    temp.close()
    f = open('temp.conf','r')
    s=f.readlines()
    number_lines = len(s)
    lista= Listbox(finestra, width=50, height=30)

    try:
        for i in range(0, number_lines):
            diz = eval(s[i])
            nome = diz['nome']
            lista.insert (i, nome)
    except:
        print "errore generale lettura database"
        pass
    lista.grid(row=0,column=0,columnspan=5, rowspan=10,)


def task():

    finestra.after(500,task)
    temp=open('temp.conf','w')
    ftp.retrbinary('RETR /home/bananapi/Desktop/Database.conf', temp.writelines)
    temp.close()
    f = open('temp.conf','r')
    s=f.readlines()
    number_lines = len(s)


    try:
        valore= eval(indice.get())
        diz = eval(s[valore])
        targa = diz['targa']
        password = diz['password']
        nome = diz['nome']
        str_password.set (str(password))
        str_targa.set(str(targa))
        str_nome.set(str(nome))

    except:
        pass
```

```
def lettura():

    global label5
    global label8
    global label9
    global label10
    global label11
    global bottone3

    try: #caso in cui la situazione precedente fosse una modifica di un qualche altro parametro
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass

    try:  #caso in cui la situazione precedente fosse una lettura di qualche valore
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        bottone3.destroy()
    except:
        pass

    try: #caso in cui la situaz. prec. fosse una aggiunta
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass



    try:
        posizione=lista.curselection()
        indice.set (str(posizione[0]))
        label5 = Label(finestra, text="                Valore                ")
        label5.grid(row=0,column=6)

        label8 = Label(finestra, textvariable=indice)
        label8.grid(row=1,column=6)
        label9 = Label(finestra, textvariable = str_targa)
        label9.grid(row=2,column=6)
        label10 = Label(finestra, textvariable = str_password)
        label10.grid(row=3,column=6)
        label11 = Label(finestra, textvariable = str_nome)
        label11.grid(row=4,column=6)
```

```
        bottone3=Button(text="elimina", command=elimina)
        bottone3.grid(row=6,column=6)

    except:
        tkMessageBox.showerror(title="Leggi", message="Selezionere un valore dalla entry box")



def modifica():

    global label5
    global label8
    global label9
    global label10
    global label11
    global label6
    global entry1
    global entry2
    global entry3
    global bottone5
    global save

    save=1
    try: #caso in cui la situazione precedente fosse una modifica di un qualche altro parametro
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass

    try:  #caso in cui la situazione precedente fosse una lettura di qualche valore
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        bottone3.destroy()
    except:
        pass

    try: #caso in cui la situaz. prec. fosse una aggiunta
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass

    try:
        posizione=lista.curselection()
```

```
        indice.set (str(posizione[0]))

        label5 = Label(finestra, text="                Valore                ")
        label5.grid(row=0,column=6)

        label8 = Label(finestra, textvariable=indice)
        label8.grid(row=1,column=6)
        label9 = Label(finestra, textvariable = str_targa)
        label9.grid(row=2,column=6)
        label10 = Label(finestra, textvariable = str_password)
        label10.grid(row=3,column=6)
        label11 = Label(finestra, textvariable = str_nome)
        label11.grid(row=4,column=6)

        label6 = Label(finestra, text="        Modifica        ")
        label6.grid(row=0,column=7)

        entry1 = Entry(finestra, textvariable = ins_targa)
        entry1.grid(row=2,column=7)
        entry2 = Entry(finestra, textvariable = ins_password)
        entry2.grid(row=3,column=7)
        entry3 = Entry(finestra, textvariable = ins_nome)
        entry3.grid(row=4,column=7)

        bottone5=Button(text="Salva", command=salva)
        bottone5.grid(row=5,column=7)

    except:
        tkMessageBox.showerror(title="Modifica", message="Selezionere un valore dalla entry box")



def aggiungi():

    global label6
    global entry1
    global entry2
    global entry3
    global bottone3
    global bottone5
    global save

    save=0

    try: #caso in cui la situazione precedente fosse una modifica di un qualche altro parametro
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass
```

```
    try:  #caso in cui la situazione precedente fosse una lettura di qualche valore
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
        label11.destroy()
        bottone3.destroy()
    except:
        pass

    try: #caso in cui la situaz. prec. fosse una aggiunta
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass


    label6 = Label(finestra, text="        Aggiungi:        ")
    label6.grid(row=0,column=7)

    entry1 = Entry(finestra, textvariable = ins_targa)
    entry1.grid(row=2,column=7)
    entry2 = Entry(finestra, textvariable = ins_password)
    entry2.grid(row=3,column=7)
    entry3 = Entry(finestra, textvariable = ins_nome)
    entry3.grid(row=4,column=7)
    bottone5=Button(text="Salva", command=salva)
    bottone5.grid(row=5,column=7)



def elimina():
    print "ELIMINO DAL DATABASE"
    print "posizione:", lista.curselection()

    temp=open('temp.conf','w')
    ftp.retrbinary('RETR /home/bananapi/Desktop/Database.conf', temp.writelines)
    temp.close()
    f = open('temp.conf','r')
    s=f.readlines()
    valore= eval(indice.get())
    print valore
    temp=open('temp.conf','w')
    diz =(s[valore])
    s.remove(diz)
    print s
    temp.writelines(s)
    temp.close()
    ftp.storbinary('STOR /home/bananapi/Desktop/Database.conf', open('temp.conf', 'r'))
    lista.destroy()
    aggiornolista()


    label5.destroy()
    label5.destroy()
```

```
    label8.destroy()
    label9.destroy()
    label10.destroy()
    label11.destroy()
    bottone3.destroy()


def salva():


    temp=open('temp.conf','w')
    ftp.retrbinary('RETR /home/bananapi/Desktop/Database.conf', temp.writelines)
    temp.close()
    f = open('temp.conf','r')




    if save:
        nl = "\n"
        s=f.readlines()
        temp=open('temp.conf','w')
        valore= eval(indice.get())
        diz = str({'targa':ins_targa.get(), 'nome':ins_nome.get(), 'password':ins_password.get(),  })+nl
        s[valore]=diz
        temp.writelines(s)
        temp.close()
        ftp.storbinary('STOR /home/bananapi/Desktop/Database.conf', open('temp.conf', 'r'))
    else:
        nl = "\n"
        s=f.readlines()
        temp=open('temp.conf','w')
        diz = str({'targa':ins_targa.get(), 'nome':ins_nome.get(), 'password':ins_password.get(),  })+nl
        s.append(diz)
        print s
        temp.writelines(s)
        temp.close()
        ftp.storbinary('STOR /home/bananapi/Desktop/Database.conf', open('temp.conf', 'r'))



    print "MODIFICHE DATABASE:"
    if save:
        print "posizione:", lista.curselection()
    print "targa:    ", ins_targa.get()
    print "password: ", ins_password.get()
    print "nome:     ", ins_nome.get()
    entry1.delete(0,END)
    entry2.delete(0,END)
    entry3.delete(0,END)
    lista.destroy()
    aggiornolista()

    try: #caso in cui la situazione precedente fosse una modifica di un qualche altro parametro
        label5.destroy()
        label8.destroy()
        label9.destroy()
        label10.destroy()
```

```
        label11.destroy()
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass

    try: #caso in cui la situaz. prec. fosse una aggiunta
        label6.destroy()
        entry1.destroy()
        entry2.destroy()
        entry3.destroy()
        bottone5.destroy()
    except:
        pass



def start_streaming():

    tkMessageBox.showinfo(title="Streaming", message="per uscire dalla prossima finestra premere q sulla tastiera")

    #os.system("dist\plate.exe")
    si = subprocess.STARTUPINFO()
    si.dwFlags |= subprocess.STARTF_USESHOWWINDOW
    subprocess.call('dist\plate.exe', startupinfo=si)

def informazioni():
    tkMessageBox.showinfo(title="Info", message="Versione del Software 0.01 Matteo Dalponte, Andrea Filippi")


try:
    ftp = FTP('192.168.0.44', 'bananapi', 'bananapi', timeout=2)
    finestra = Tk()
    finestra.geometry("750x550")
    finestra.title("Gestione Degli Accessi")
    #finestra.iconbitmap(r'c:\Python27\DLLs\py.ico')
    finestra.iconbitmap(r'icona.ico')

    indice=StringVar()
    str_targa=StringVar()
    str_password=StringVar()
    str_nome=StringVar()
    str_targaEntry=StringVar()
    ins_targa=StringVar()
    ins_nome=StringVar()
    ins_password=StringVar()

    barra_menu=Menu(finestra)
    menu_file=Menu(barra_menu)
    barra_menu.add_cascade(label="File", menu=menu_file)
    menu_file.add_command(label="Streaming", command=start_streaming)
    menu_info=Menu(barra_menu)
    barra_menu.add_cascade(label="Info", menu=menu_info)
    menu_info.add_command(label="Informazioni", command=informazioni)
```

```
    finestra.config(menu=barra_menu)

    aggiornolista()


    bottone1=Button(text="Leggi", command=lettura)
    bottone1.grid(row=11,column=1)
    bottone2=Button(text="Modifica", command=modifica)
    bottone2.grid(row=11,column=2)
    bottone4=Button(text="Aggiungi", command=aggiungi)
    bottone4.grid(row=11,column=3)


    label1 = Label(finestra, text="Valore indice: ")
    label1.grid(row=1,column=5)
    label2 = Label(finestra, text="Targa:        ")
    label2.grid(row=2,column=5)
    label3 = Label(finestra, text="Password:     ")
    label3.grid(row=3,column=5)
    label4 = Label(finestra, text="Nome:         ")
    label4.grid(row=4,column=5)

    logo = PhotoImage(file="targhe1.gif")
    w1 = Label(finestra, image=logo).grid(row=7,column=5,columnspan=5, rowspan=10,sticky=W)



    finestra.after(500,task)
    finestra.mainloop()

except:
    error=Tk()
    labelerror = Label(error, text="Il server FTP non ha potuto aprire la connessione con il database").pack()
    labelerror = Label(error, text="Si prega di controllare che il server sia acceso e sia presente una risposta ping verso
lo stesso.").pack()
    tkMessageBox.showerror(title="Errore connessione", message="controllare se il server è in funzione e verificarne la
connessione alla rete lan")
    error.destroy()
```

RELAZIONE TECNICA DELL' AZIENDA ALGORAB

Creatività e spirito di intraprendenza

Algorab, in qualità di azienda che ha raggiunto una posizione di primo piano nell'ambito del telecontrollo, non può che riconoscere che il progetto legato al riconoscimento delle targhe risulta essere interessante e ben ideato.

In una realtà imprenditoriale qualità come l'intraprendenza, l'approccio al risultato, il problem solving, l'autonomia, sono da sempre apprezzate. Gli studenti Andrea Filippi e Matteo Dalponte che, con i docenti Giuseppe Russo e Claudio Casotti, hanno contribuito alla realizzazione del progetto si sono contraddistinti per il possesso di tutte queste caratteristiche, che Algorab ha notato fin dal primo incontro.

La preparazione, la capacità e la motivazione dimostrata dal personale e dagli studenti dell'Istituto hanno permesso di creare un progetto dall'indiscutibile potenziale, suscettibile di applicazione in vari ambiti. Le utilità di controllo degli accessi di veicoli in un parcheggio e la registrazione dei passaggi, specie se implementate con hardware e software più performanti, sono sicuramente suscettibili di attuazione in diversi settori e scenari.

Imprenditorialità

Uno degli aspetti più meritevoli del sistema è sicuramente costituito dalla estrema economicità di hardware e software.

Andrea Filippi e Matteo Dalponte hanno infatti portato a compimento il lavoro sfruttando delle librerie software open source sulle quali hanno poi sviluppato il proprio applicativo. Ne è conseguito un pacchetto software di sistema assolutamente economico e libero dai costi e vincoli che gli applicativi proprietari impongono. Riguardo all'architettura software scelta va inoltre considerato che il sistema risulta anche essere portabile sfruttando diversi tipi di hardware, sia lato elaborazione (la scheda PC), sia lato telecamera.

Un medesimo impegno teso all'economicità è stato svolto anche con riferimento alla componente hardware per la realizzazione del prototipo. Certamente, in caso di utilizzo del sistema in uno scenario reale, sarà opportuno ricorrere ad alcune componenti hardware più costose, come ad esempio una telecamera adatta all'uso esterno. Resta tuttavia degna di nota la semplicità, linearità ed economicità del progetto, che Algorab ha avuto modo di apprezzare fin dal primo incontro.

Conclusioni

Il sistema software sviluppato, allo stato attuale, potrebbe essere utilizzato fin da subito per regolare l'accesso a parcheggi privati in cui l'entrata è consentita esclusivamente a determinate vetture, la cui targa è nota e inserita a sistema. Per applicare il sistema a scenari più complessi (quali parcheggi a pagamento, controllo degli accessi, registrazione dei dati storici) sarebbe necessario affinare il software di controllo delle autorizzazioni.

Ad ogni modo, quel che rileva è che il progetto per il sistema di riconoscimento delle targhe colpisce per la sua efficacia e, al contempo, per la semplicità ed economicità. Se, come scriveva Sydney J. Harris, "lo scopo della scuola è quello di trasformare gli specchi in finestre" si può senz'altro sostenere che l'approccio pratico ed intrapredente dimostrato dagli studenti Andrea Filippi e Matteo Dalponte e dai docenti dell'Istituto Buonarroti ha senz'altro colto nel segno.

ALGORAB S.r.l.
Via L. Negrelli 21/13 - tel. 0461 242482
38015 LAVIS (Trento)
Partita IVA 01367190228
Reg. Imprese Trento n. 01367190228

algorab