

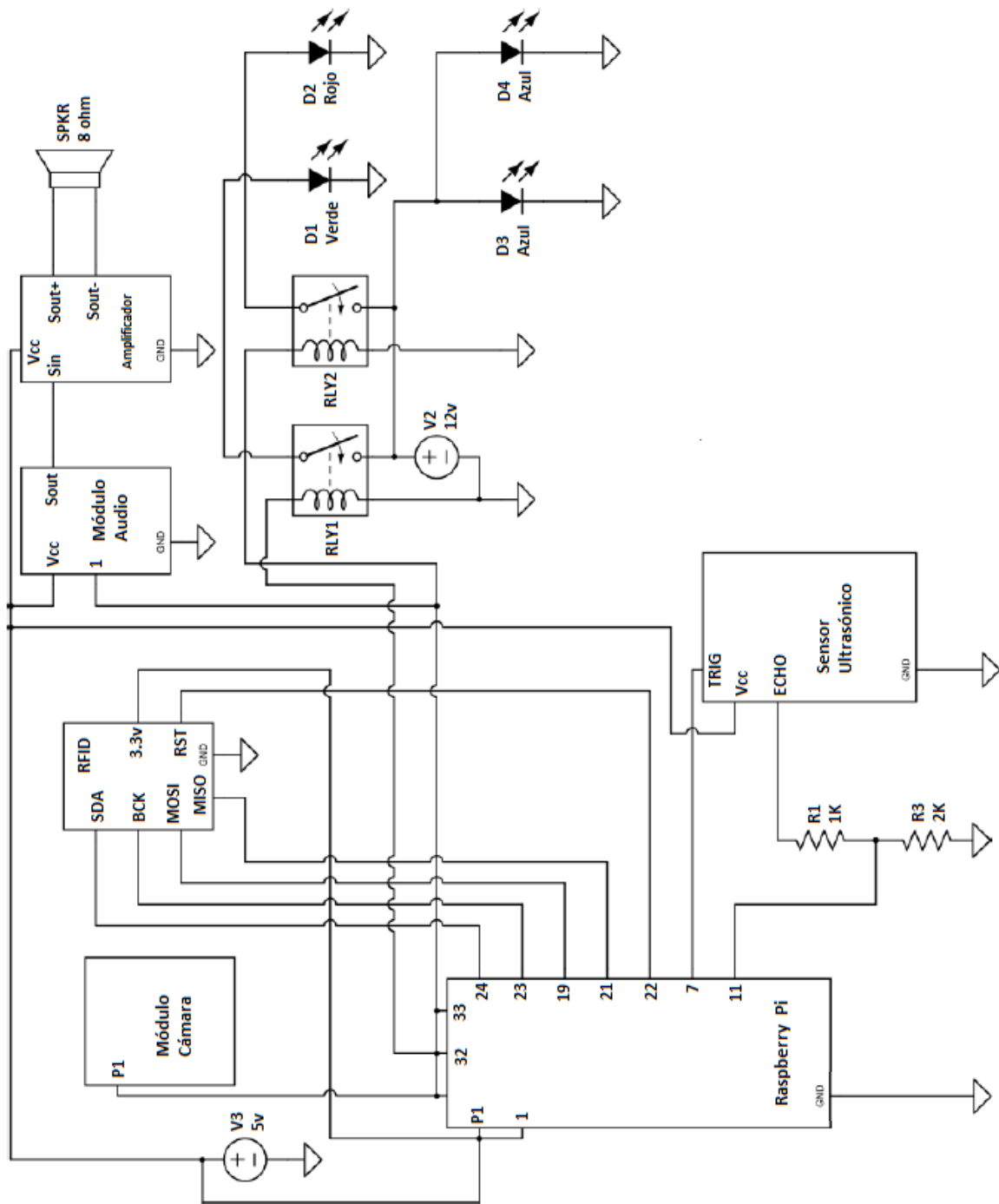
Este proyecto se encuentra bajo licencia GPL3, vease este link:

<https://www.gnu.org/licenses/gpl-3.0.txt>

This project is under GPL3 license, refer to the link shown below:

<https://www.gnu.org/licenses/gpl-3.0.txt>

Diagrama del circuito



Código del programa

```
from openalpr import Alpr
from picamera import PiCamera
from time import sleep
from datetime import datetime
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
import time
import json
from Adafruit_IO import Client, Feed

#Inicializa RFID reader
reader = SimpleMFRC522()

## Adafruit IO Inicia
# key code
ADAFRUIT_IO_KEY = 'f1d4657ca7b242d99429d253b445fe79'

# Usuario
ADAFRUIT_IO_USERNAME = 'salazarabj'

# Creamos la instancia del REST client.
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

# creamos los feed de Adafruit IO
placa_feed = aio.feeds('placa')
alerta_feed = aio.feeds('alerta')
#confidence_feed = aio.feeds('confidence')
```

```
## Adafruit IO Termina
```

```
## Lector de placas Inicia
```

```
alpr = Alpr("us", "/etc/openalpr/openalpr.conf",  
            "/usr/share/openalpr/runtime_data")
```

```
camera = PiCamera()
```

```
placa = "
```

```
##Setup del sensor de distancia
```

```
PIN_TRIGGER = 7
```

```
PIN_ECHO = 11
```

```
GPIO.setup(PIN_TRIGGER, GPIO.OUT)
```

```
GPIO.setup(PIN_ECHO, GPIO.IN)
```

```
##Setup de Relays
```

```
ALARMA = 31
```

```
VERDE = 32
```

```
ROJO = 33
```

```
GPIO.setup(ALARMA, GPIO.OUT)
```

```
GPIO.setup(ROJO, GPIO.OUT)
```

```
GPIO.setup(VERDE, GPIO.OUT)
```

```
#Constantes para el loop de control y verificacion de datos
```

```
DISTANCIA = 20 #Distancia para iniciar la lectura de distancia
```

```
placas = ['BBH322', '139940', '418008']
```

```
tarjetas = [991666632722, 40168296170, 40124848487]
```

try:

while True:

 #Estado por defecto de las alertas

 GPIO.output(ALARMA, GPIO.LOW)

 GPIO.output(ROJO, GPIO.LOW)

 GPIO.output(VERDE, GPIO.HIGH)

 #Sensor distancia

 GPIO.output(PIN_TRIGGER, GPIO.LOW)

 print "Esperando al sensor"

 time.sleep(0.1)

 print "Calculando distancia"

 GPIO.output(PIN_TRIGGER, GPIO.HIGH)

 time.sleep(0.00001)

 GPIO.output(PIN_TRIGGER, GPIO.LOW)

 while GPIO.input(PIN_ECHO)==0:

 pulse_start_time = time.time()

 while GPIO.input(PIN_ECHO)==1:

 pulse_end_time = time.time()

 pulse_duration = pulse_end_time - pulse_start_time

 distance = round(pulse_duration * 17150, 2)

```

print "Distancia:",distance,"cm"

#Fin sensor de distancia


if distance < DISTANCIA:

    ##Leer placa: Leemos placa por que esta un auto presente
    print('Leer placa')
    #Nombre con hora y fecha para cada foto
    now = datetime.now()
    _datetime = now.strftime("%m-%d-%Y-%H-%M-%S")
    photoName = '/home/pi/psys/f_placas/'+_datetime+'.jpg'

    # Take a photo
    print('Taking a photo')
    camera.capture(photoName)

    # Ask OpenALPR what it thinks
    analysis = alpr.recognize_file(photoName)

    # If no results, no car!
    if len(analysis['results']) == 0:
        print('No number plate detected')

    else:
        placa = analysis['results'][0]['plate']
        print('Numero de placa detectado: ' + placa)

    #print(json.dumps(analysis, indent=4))
    if placa in placas:
        #Relay verde, auto permitido

```

```
print('Se Enciende relay verde')
GPIO.output(VERDE, GPIO.HIGH)
GPIO.output(ROJO, GPIO.LOW)
print('***** Se encontro placa *****')
```

```
alerta = 0
#Enviamos datos al servidor
print('Enviando datos al servidor')
aio.send(placa_feed.key, str(placa))
aio.send(alerta_feed.key, str(alerta))
# aio.send(confidence_feed.key, str(confidence_placa))
print('Datos enviados')
```

else:

```
#Relay rojo, auto no permitido, activa alarma
GPIO.output(ROJO, GPIO.HIGH)
GPIO.output(VERDE, GPIO.LOW)
GPIO.output(ALARMA, GPIO.HIGH)
print("")
print("")
print('Se Enciende relay rojo')
print('***** No se encontro placa *****')
print("")
print("")
```

```
alerta = 1
#Enviamos datos al servidor
print('Enviando datos al servidor')
aio.send(placa_feed.key, str(placa))
aio.send(alerta_feed.key, str(alerta))
```

```

# aio.send(confidence_feed.key, str(confidence_placa))
print('Datos enviados')

#Esperamos por tarjeta RFID para detener la alarma
print('Esperando RFID')
tarjeta, text = reader.read()
print('Tarjeta Id:')
print(tarjeta)
#print(text)
if tarjeta in tarjetas:
    #Relay verde, auto permitido
    print('Se encontro tarjeta')
    GPIO.output(ROJO, GPIO.LOW)
    GPIO.output(VERDE, GPIO.HIGH)
    GPIO.output(ALARMA, GPIO.LOW)
    #Apagamos relay rojo
    #Agrega placa a la lista de placas
    placas.append(placa)
    print("***** Lista de placas *****")
    print placas
    alerta = 0
    #Enviamos datos al servidor
    #Esperamos para asegurar el envio
    time.sleep(1)
    print('Enviando datos al servidor')
    aio.send(placa_feed.key, str(placa))
    aio.send(alerta_feed.key, str(alerta))
    # aio.send(confidence_feed.key, str(confidence_placa))
    print('Datos enviados')

```



```
        else:
            print('Tarjeta invalida')
            ##Termina lectura de placas
            #Fin de leer placa
    else:
        print "No leer placa"

except KeyboardInterrupt:
    GPIO.output(ROJO, GPIO.LOW)
    GPIO.output(VERDE, GPIO.LOW)
    GPIO.output(ALARMA, GPIO.LOW)
    print('Shutting down')
    #alpr.unload()
```