

**Project Objectives**

The purpose of this system/device is to be able to recognize and store license plate information and location of any vehicle with ease anywhere at any time. This system can even be adapted to any device which support the necessary hardware and software capabilities. With a Raspberry Pi 4, a Raspberry Pi Camera module and a GPS module can easily geo tag the information of any license plate using an API.

The use of a **Raspberry Pi 4** along with a modular **micro camera** and a **GPS** module to take the picture and track the location of the entries. Once the picture is taken the information is relayed to an **API** to read and return the information of the plate. The **API** will return many parameters like car make, model, color, etc. out only interest would be the license plate number and state, so this would be filter using python to only output the desired information.

**Components Used**

* Raspberry Pi 4 (4GB RAM)
* Raspberry Pi Camera V2.1 (8 Megapixel)
* GPS Module (GY-GPS6MV1) from DIYmall
* API from OpenALPR (CarCheck API)

**Picture or Schematic of the System**

|  |  |
| --- | --- |
| Getting started with the Camera Module - Introduction | Raspberry ... |  |
|  | |

**Computer Program**

|  |
| --- |
| #Imports need to access all the parameter required for the code.  import requests  import base64  import json  from datetime import datetime  import serial  import pynmea2  from picamera import PiCamera  #Declare the time and date variable.  dateTimeObj = datetime.now()  #declare the camera variable.  camera = PiCamera()  # serial port in which the GPS is connected  port = "/dev/serial0"    #define and format the information from GPS to print.  def parseGPS(str):  if str.find('GGA') > 0:  msg = pynmea2.parse(str)  print ("Lat: %s %s -- Lon: %s %s -- Altitude: %s %s -- Satellites: %s" % (msg.lat,msg.lat\_dir,msg.lon,msg.lon\_dir,msg.altitude,msg.altitude\_units,msg.num\_sats))      #all the information imported, port ID and delay  serialPort = serial.Serial(port, baudrate = 9600, timeout = 0.5)  print(dateTimeObj) #print the date and time tag  #delay to wait for the GPS information.  for i in range(5):  str = serialPort.readline()  parseGPS(str)    # take and store the information from RPi camera.  camera.capture('/home/pi/Project/Image.jpg')  IMAGE\_PATH='/home/pi/Project/Image.jpg' #path in where the picture take was store.    SECRET\_KEY='sk\_d478fff1a6027dcf8dc6a305' #key to access the API  #convert the information to base 64 to transmit to API  with open(IMAGE\_PATH, 'rb') as image\_file:  img\_base64 = base64.b64encode(image\_file.read())  #Request the API using the key url='https://api.openalpr.com/v2/recognize\_bytes?recognize\_vehicle=1&country=us&secret\_key=%s' % (SECRET\_KEY)  r = requests.post(url, data=img\_base64) #store the information request    #look for information of a license plate else…  try:  print("License Plate: " + r.json()['results'][0]['plate'] + " State: " + r.json()['results'][0]['region']) #print just the plate and state    #information not found.  except:  print("No Plate Information Found") |

**Troubleshooting**

The GPS take time to connect and relay the coordinates from the satellite, in order to fix this problem a “For” loop was created to allow for the GPS to revive the information required to print the geo-location tag.

**Recommendations and Conclusions**

The biggest issue this system could encounter is its modular form, with the use of a Raspberry Pi makes it extremely easy to set up almost any module, but this is a drawback as well. The idea behind the system is mobility and ease of accessibility to a camera and GPS geo-tag system for a license plate scanner, the Raspberry Pi does this by incorporating more piece of hardware to it which makes it bulky and hard to create a mobile unit, if the code where to be implemented into an already Camera/GPS ready device like a phone or an APP for a device with the same capabilities the utilization would stay the same while proving easier or faster avenues to improve the system.

In conclusion this was a simple straight forward project that used modules to show how three things that not usually though of together can come together and from a useful system, whether for security or personal use the implementation of this system can have a wide range of possibilities and utility for many individual or groups.