

Projet fin d'année

DRIVING LEARNING MODULE

Présenté par :

Arthur BORG
Félix GOUBIN
Alexandre LEPEERS
Talla TOURE



Apprentis SEE5

Le : 14/12/2021

Plan

Introduction

I/ Traitement des flux vidéo

II/ Gestion des alertes

III/ Implémentation piCar & control moteur

IV/ Affichage sur Dashboard

Conclusion

Introduction

Projet de fin d'année

Présentation du DLM

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Introduction

Drinving Learning Module:

- Algorithme d'assistance à l'apprentissage de la conduite
- Prévention de risques
- Informations de parcours -> écoconduite
- Sauvegarde du parcours -> subventions potentielles par les assurances auto
- Assistance visuel aux malentendants
- Réduction d'agressions sur les moniteurs d'auto-école/permis (triplé depuis 2017)

I/ Traitement des flux vidéo

Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

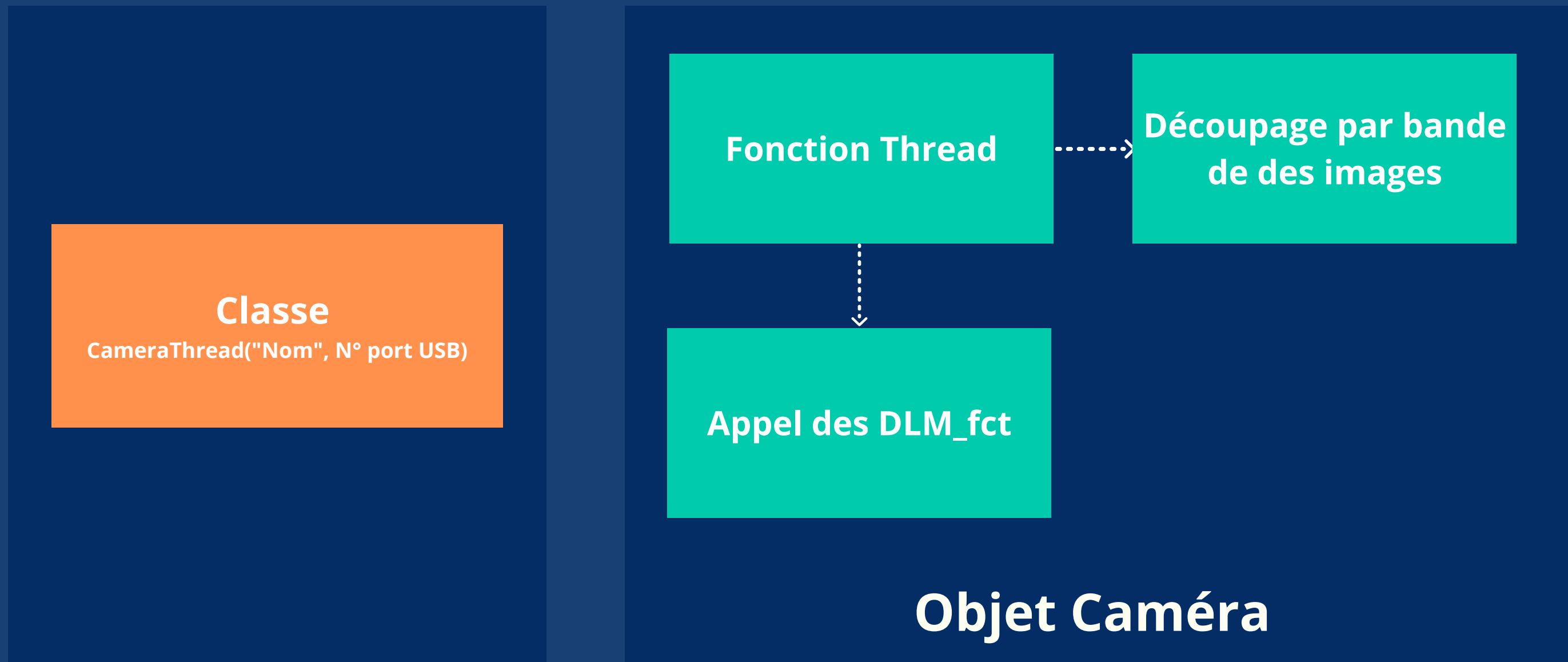
IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Initialisation du flux vidéo



Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Initialisation du flux vidéo

```
class camThread(threading.Thread):
    def __init__(self, previewName, camID):
        threading.Thread.__init__(self)
        self.previewName = previewName
        self.camID = camID
    def run(self):
        print("Starting " + self.previewName)
        camPreview(self.previewName, self.camID)
```

Classe

```
def camPreview(previewName, camID):
    # cv2.namedWindow(previewName)
    cam = cv2.VideoCapture(camID)
    if cam.isOpened(): # try to get the first frame
        rval, frame = cam.read()
        classe, colors, net = Object_Detection.init_model()
    else:
        rval = False

    while rval:
        # cv2.imshow(previewName, frame)
        rval, frame = cam.read()
        # --- Split frame --- #
        height = frame.shape[1]
        width = frame.shape[0]

        frame2scan_left = frame[0:width, 0:int(height * .3)]
        frame2scan_middle = frame[0:width, int(height * .3):int(height * .7)]
        frame2scan_right = frame[0:width, int(height * .7):height]

        # --- Operations on the Frame --- #
        frame2scan_left, frame2scan_right = Line_Detection.line_detection(frame2scan_left, frame2scan_right)
        frame2scan_middle = Object_Detection.object_detection(frame2scan_middle, camID, classe, colors, net)

        # --- Regroupement de la frame pour le résultat --- #
        result_frame = np.concatenate((frame2scan_left, frame2scan_middle, frame2scan_right), 1)
        result_frame = Traffic_Light_Detection.traffic_light_detection(result_frame)
        # affichage du flux vidéo dans une fenêtre
        cv2.namedWindow("Result_" + str(camID))
        cv2.imshow("Result_" + str(camID), result_frame)
        # --- Exit the Video --- #
        key = cv2.waitKey(20)
        if key == 27: # exit on ESC
            break
    cv2.destroyWindow(previewName)
```

Fonction thread

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Fonction de signalisation N°1

Détection de
feux tricolores

DLM1

Recherche de rectangles
au fond noir

Find_rect()

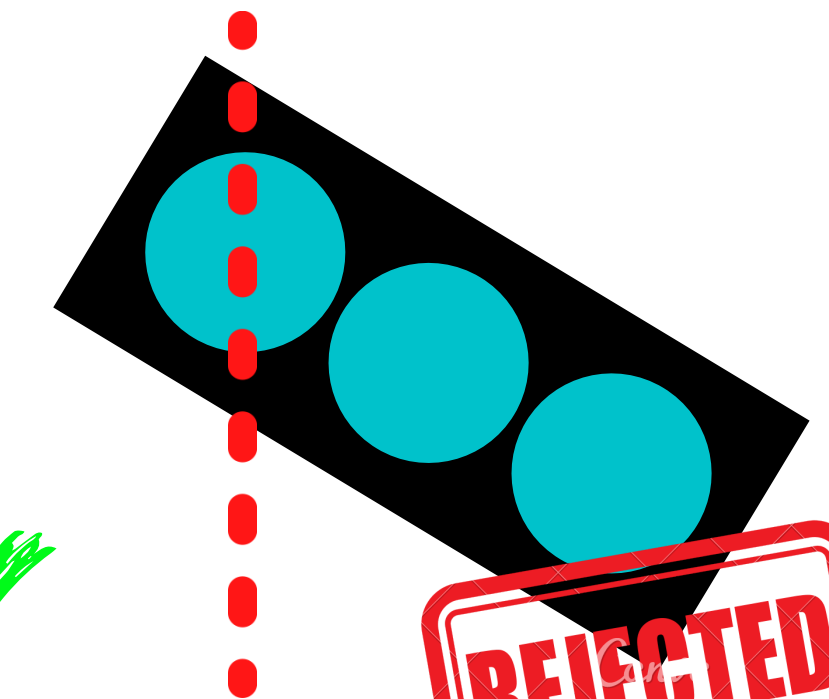
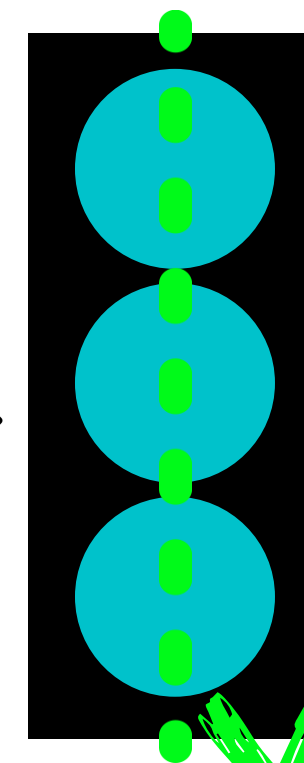
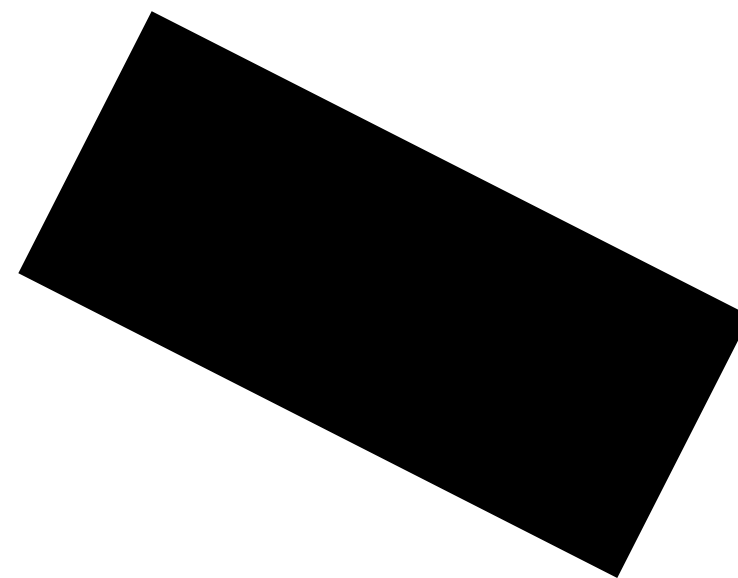
Recherche 3 cercles dans le
rectangle.

Test alignement vertical

Find_circle()



Find_rect()



Find_circle()

REJECTED

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Fonction de signalisation N°1

Recherche de rectangles
au fond noir

Find rect()

Séquencement par
étape

```
def find_rect(frame2scan_top):  
    # --- Recherche de noir dans l'espace colorimétrique HSV --- #  
    hsv_frame = cv2.cvtColor(frame2scan_top, cv2.COLOR_BGR2HSV)  
    h, s, v = cv2.split(hsv_frame)  
    v[v <= 0] = 0  
    v[v > 50] = 0  
  
    # --- Pre-traitement de l'image pour faire ressortir les contours --- #  
    normed = cv2.normalize(v, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8UC1)  
    kernel = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(3, 3))  
    opened = cv2.morphologyEx(normed, cv2.MORPH_OPEN, kernel)  
  
    # --- Detections des contours --- #  
    (contours, _) = cv2.findContours(opened, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)  
  
    bboxes = []  
    cnts = []  
    dst = frame2scan_top.copy()  
    i = 0  
  
    # --- Test de la forme des contours à la recherche d'un rectangle noir --- #  
    for cnt in contours:  
        i = i + 50  
        ## Récupère le rectangle droit de la forme  
        bbox = cv2.boundingRect(cnt)  
        x, y, w, h = bbox  
        if w < 150 and h < 2 * w: # La hauteur ne peut pas dépasser plus de 2 fois la largeur  
            continue  
  
        circle, nb_circle = find_circle(dst[y: y + h, x: x + w]) # Appel de la fonction pour rechercher de  
        if nb_circle == 3: # Si l'on a 3 cercles alignés, dessine le rectangle qui forme le feu tricolore  
            cv2.rectangle(dst, (x, y), (x + w, y + h), (0, 0, 255), 2, 16) # Dessine rect
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Fonction de signalisation N°1

Recherche 3 cercles dans le rectangle.

Test alignement vertical

Find_Circles()

Séquencement par étape

```
def find_circle(rect_frame):  
    gray = cv2.cvtColor(rect_frame, cv2.COLOR_BGR2GRAY)  
    gauss = cv2.GaussianBlur(gray, (9, 9), 2, 2)  
    circles = cv2.HoughCircles(gauss, cv2.HOUGH_GRADIENT, 2.0, 20, 50, 30, 100)  
    count_circle = 0  
    # ensure at least some circles were found  
    if circles is not None:  
        # convert the (x, y) coordinates and radius of the circles to integers  
        circles = np.round(circles[0, :]).astype("int")
```

```
c_cercle_x = {'x1': 0, 'x2': 0, 'x3': 0}  
c_cercle_y = {'y1': 0, 'y2': 0, 'y3': 0}  
c_radius = list()  
if len(circles) == 3:  
    try:  
        for (x, y, r) in circles:  
            count_circle += 1  
            c_cercle_x['x' + str(count_circle)] = x  
            c_cercle_y['y' + str(count_circle)] = y  
            c_radius.append(r)  
    except TypeError:  
        pass
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉO

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Traitement des flux vidéo

Fonction de signalisation N°1

Recherche 3 cercles dans le rectangle.

Test alignement vertical

Find Circles()

Séquencement par étape

```
try:
    if(c_cercle_x["x1"] and c_cercle_x["x2"]) == c_cercle_x["x3"]: # Test l'alignement des cercles
        # --- Dessine les cercles --- #
        for i in range(count_circle):
            cv2.circle(rect_frame, (c_cercle_x["x" + str(i+1)], c_cercle_y["y" + str(i+1)]), c_radius[i],
                        (0, 255, 0), 4)

            cercle_rouge = min(c_cercle_y,
                               key=c_cercle_y.get) # Récupère l'index du cercle avec la position la plus haute
            cercle_vert = max(c_cercle_y,
                              key=c_cercle_y.get) # Récupère l'index du cercle avec la position la plus basse

            _, _, R = rect_frame[c_cercle_y[cercle_rouge], c_cercle_x['x1']] # Récupère la composante de Rouge
            _, G, _ = rect_frame[c_cercle_y[cercle_vert], c_cercle_x['x1']] # Récupère la composante de Vert de

            if (R >= 130 and G < R): # Si le cercle le plus haut à suffisamment de rouge, alors il detecte un :
                cv2.putText(rect_frame, "R", (c_cercle_x['x1'], c_cercle_y[cercle_rouge]),
                            cv2.FONT_HERSHEY_SIMPLEX,
                            0.5, (255, 0, 0))
            elif (G >= 130 and R < G): # Si le cercle du bas est suffisamment vert, alors feu vert
                cv2.putText(rect_frame, "V", (c_cercle_x['x1'], c_cercle_y[cercle_vert]),
                            cv2.FONT_HERSHEY_SIMPLEX,
                            0.5, (255, 0, 0))
            else:
                count_circle = 0
        else:
            count_circle = 0 # si pas d'alignement, rejete l'hypothèse du feu tricolor et annule le dessin de
except IndexError:
    pass
```

II/ Gestion des alertes



Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Problématique

Organiser la gestion des alertes

Optimiser la conduite :

- - coûteuse
- - polluante
- + confortable
- + responsable

Analyser les données calculées par l'application

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Solutions

Protocole de communication

- Format d'une trame CCTTDDDD
- Broker MQTT

Architecture de sauvegarde des alertes

- locale
- externe - MongoDB

Interface de développement Node-Red

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Architecture

Traitement image



**Gestion
des alertes**

Display



Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Format d'une trame : CCTTDDDD

CC :

- 00 avant
- 01 arriere

TT :

- 00 info
- 01 warning
- 10 critical

DDDD :

- | | |
|----------------|----------------------|
| • 0000 pieton | • 0110 feu_V |
| • 0001 voiture | • 0111 vitesse |
| • 0010 camion | • 1000 plaque |
| • 0011 velo | • 1001 stop |
| • 0100 ligne | • 1010 ceder_passage |
| • 0101 feu_R | • 1011 sens_interdit |

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Serveur MQTT local

Installation

```
sudo apt update  
sudo apt install mosquitto mosquitto-clients
```

Enable/disable mosquitto broker when device is turned on :

```
sudo systemctl (enable|disable) mosquitto
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Serveur MQTT local

```
1639467760: New connection from 127.0.0.1 on port 1883.  
1639467760: New client connected from 127.0.0.1 as mosqsub|15492-DESKTOP-K (c1, k60).  
1639467760: Sending CONNACK to mosqsub|15492-DESKTOP-K (0, 0)  
1639467760: Received SUBSCRIBE from mosqsub|15492-DESKTOP-K  
1639467760:      dlm/display/ (QoS 0)  
1639467760: mosqsub|15492-DESKTOP-K 0 dlm/display/  
1639467760: Sending SUBACK to mosqsub|15492-DESKTOP-K  
1639467789: Received UNSUBSCRIBE from nodered_46d934b0efdb4833  
1639467789:      /dlm/logs
```

Broker - port 1883

```
nshx@DESKTOP-KDLADB2:~$ mosquitto_sub -t "/dlm/display/"  
feu_R  
velo  
vitesse  
_
```

Subscriber - Display

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Base de données MongoDB

DLM.logs

COLLECTION SIZE: 1.91KB TOTAL DOCUMENTS: 8 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

FILTER { field: 'value' }

QUERY RESULTS 1-8 OF 8

```
_id: ObjectId("61b792d3f66874d815f99478")
payload: "01000010"
time: "2021-12-13 19:36:10"
carID: "12345"
topic: "/dlm/logs"
camera: "front"
type: "info"
data: "camion"
_msgid: "dcaba2755385c0ac"
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Base de données MongoDB

1. Create project: DLM
2. Create user | pw
3. Create cluster
4. Create collection: logs
5. Configure Network Access
6. Get remote URL: Connect()

```
const uri = "mongodb+srv://adminfgo:  
<password>@cluster0.t22mb.mongodb.net/myFirstDatabase?retryWrites=true&w=majority";
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Interface Node-Red

Installation

From Ubuntu Shell :

```
sudo apt install build-essential git curl  
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered
```

1. NodeJS

2. npm - node packet manager

>> node-red - https://localhost:1880

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Interface Node-Red

```
nshx@DESKTOP-KDLADB2:~$ cd ~/.node-red/
nshx@DESKTOP-KDLADB2:~/.node-red$ ls -al
total 144
drwxr-xr-x 1 nshx nshx  4096 Dec 14 08:45 .
drwxr-xr-x 1 nshx nshx  4096 Dec 14 03:24 ..
-rw-r--r-- 1 nshx nshx   131 Dec 14 03:29 .config.modules.json
-rw-r--r-- 1 nshx nshx 19187 Dec 14 03:29 .config.nodes.json
-rw-r--r-- 1 nshx nshx 18645 Dec 14 03:29 .config.nodes.json.backup
-rw-r--r-- 1 nshx nshx    95 Dec 14 03:27 .config.runtime.json
-rw-r--r-- 1 nshx nshx   455 Dec 14 03:28 .config.users.json
-rw-r--r-- 1 nshx nshx   453 Dec 14 03:28 .config.users.json.backup
-rw-r--r-- 1 nshx nshx 16015 Dec 14 08:45 .flows.json.backup
-rw-r--r-- 1 nshx nshx    51 Dec 14 03:35 .flows_cred.json.backup
drwxr-xr-x 1 nshx nshx  4096 Dec 14 03:32 DLM
-rw-r--r-- 1 nshx nshx 16015 Dec 14 08:45 flows.json
-rw-r--r-- 1 nshx nshx   131 Dec 14 03:35 flows_cred.json
drwxr-xr-x 1 nshx nshx  4096 Dec 14 03:27 lib
drwxr-xr-x 1 nshx nshx  4096 Dec 14 03:29 node_modules
-rw-r--r-- 1 nshx nshx 46975 Dec 14 03:29 package-lock.json
-rw-r--r-- 1 nshx nshx   503 Dec 14 03:29 package.json
-rw-r--r-- 1 nshx nshx 20290 Dec 14 03:27 settings.js
```

Arborescence node-red

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Dépendance mongodb

```
const { MongoClient } =  
    require('mongodb');
```

From Ubuntu Shell:

```
cd ~/.node-red  
npm install mongodb  
vi setting.js
```

```
/** The following property can be used to set predefined values in Global Context.  
 * This allows extra node modules to be made available with in Function node.  
 * For example, the following:  
 *   functionGlobalContext: { os:require('os') }  
 * will allow the `os` module to be accessed in a Function node using:  
 *   global.get("os")  
 */  
functionGlobalContext: {  
    // os:require('os').  
    mongodb:require('mongodb')  
},
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

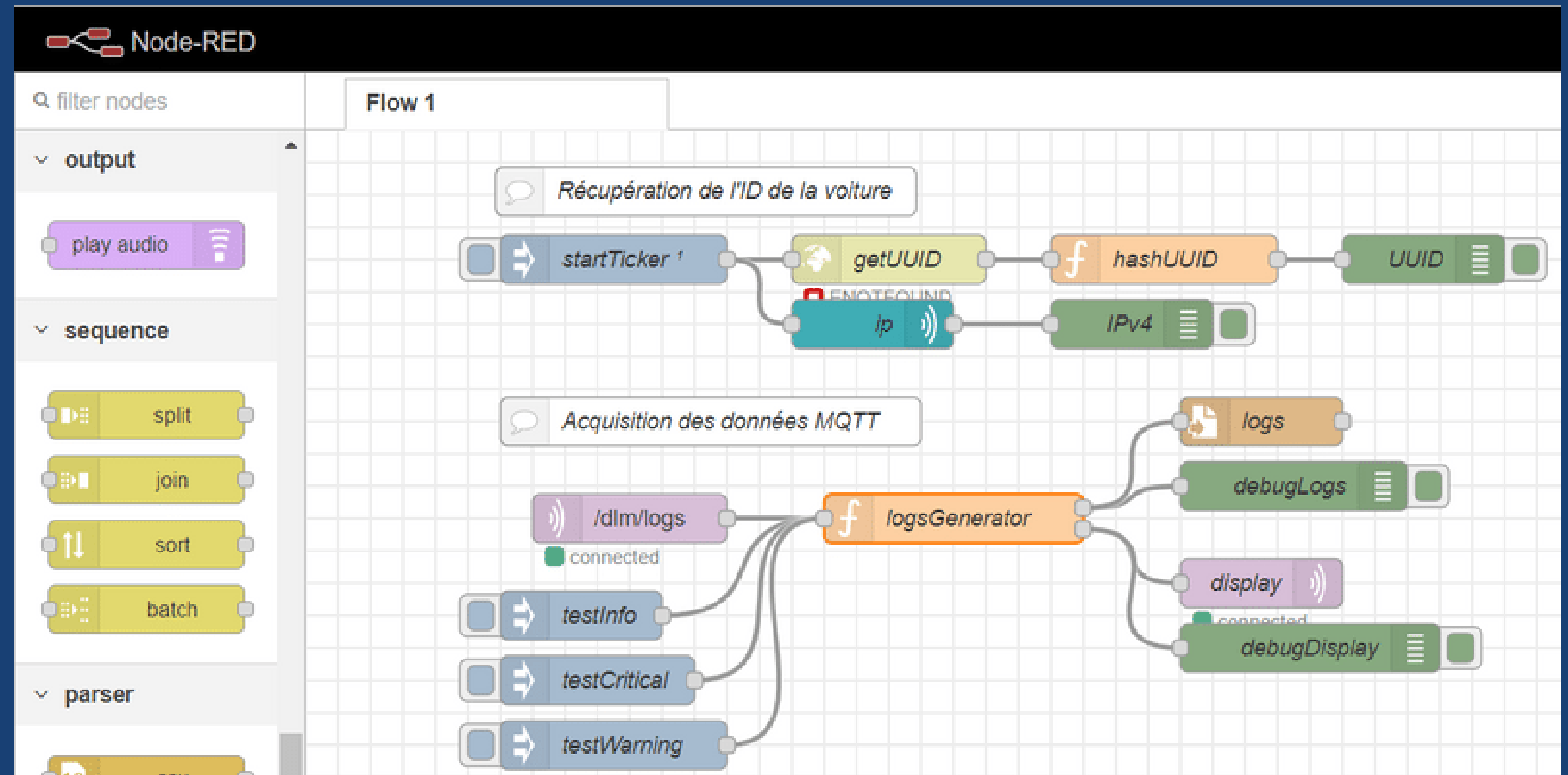
IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Flow node-red



Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Flow node-red

node : logsGenerator

- getPayload()
- buildJSON()
- parsePayload()
- camera()
- type()
- data()

```
9      + ('00' + date.getMinutes()).slice(-2) + ':'
10     + ('00' + date.getSeconds()).slice(-2)
11   );
12
13   var m_json = {
14     "payload": {
15       "carID" : global.get("carID"),
16       "topic" : msg.topic,
17       "rawData" : msg.payload.toString(),
18       "time" : date,
19       "camera" : CC(msg.payload.substring(0,2)),
20       "type" : TT(msg.payload.substring(2,4)),
21       "data" : DDDD(msg.payload.substring(4,8))
22     }
23   };
24
25   return [m_json, {"payload" : DDDD(msg.payload.substring(4,8))}];
26 } catch (Exception) {
27   return "trame inconnue"
28 }
29
30 function CC(cc){
31   try{
32     if ( cc == "00" ) { return "front"; }
33     if ( cc == "01" ) { return "back"; }
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

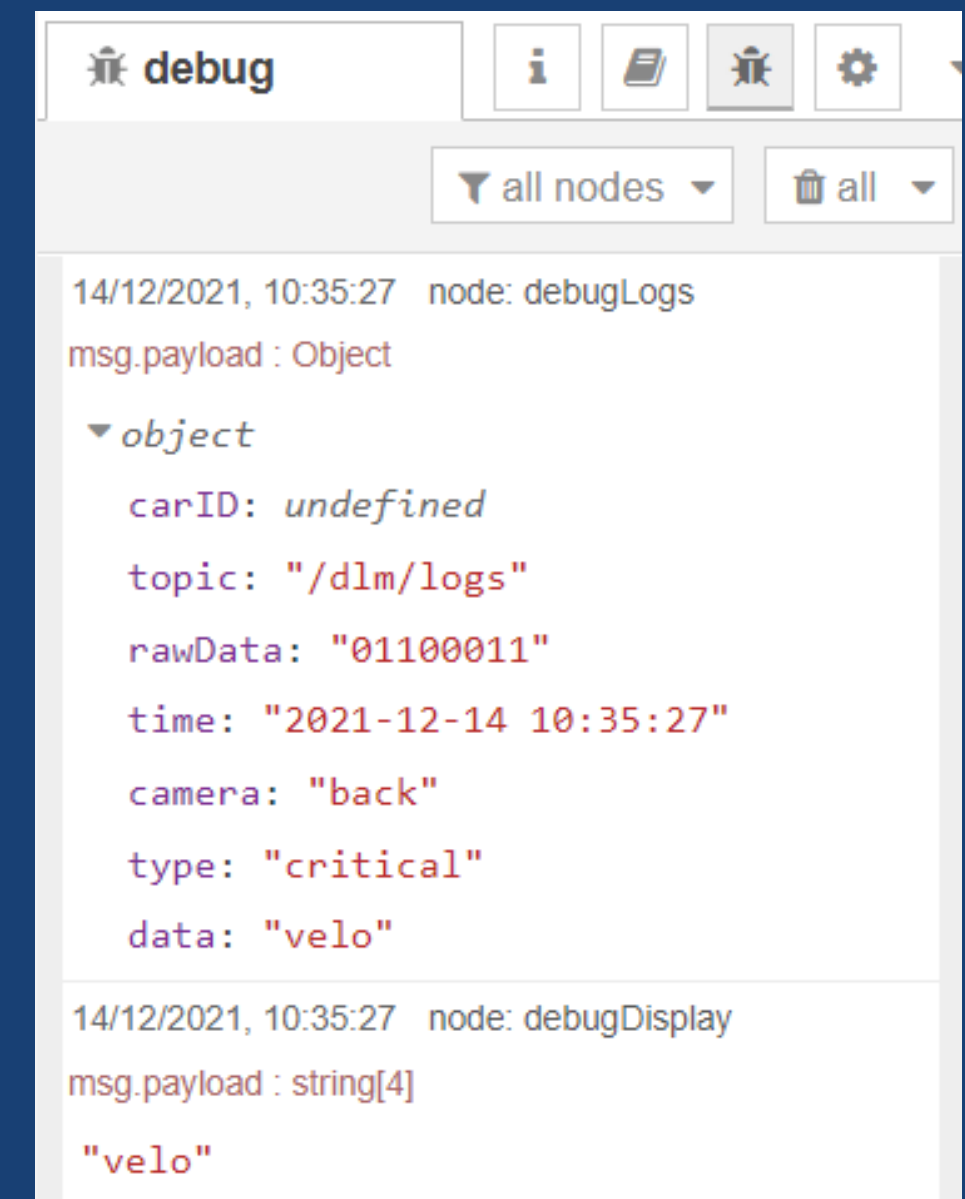
SOURCES

Gestion des alertes

Flow node-red

node : logsGenerator

- getPayload()
- buildJSON()
- parsePayload()
- camera()
- type()
- data()



Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

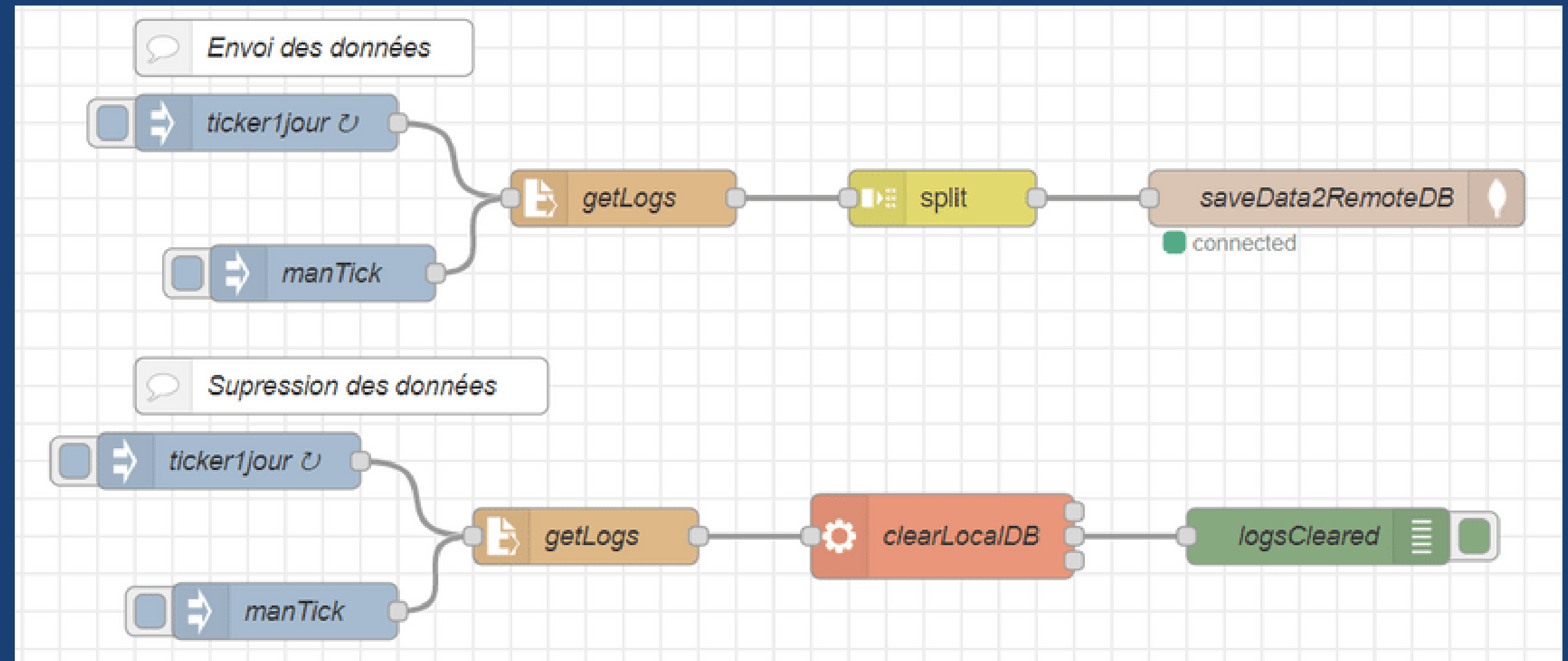
IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Flow node-red



Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

Flow node-red : Connection à mongoDB

Edit mongodb out node > **Edit mongodb node**

Delete Cancel Update

Properties

Host cluster0.t22mb.mongodb.net

Connection topology DNS Cluster (mongodb+srv://

Connect options ?retryWrites=true&w=majority

Database DLM

Username adminfgo

Password

Name DLM

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes

```
nshx@DESKTOP-KDLAD82: ~/node-red/DLM/logs
-rw-r--r-- 1 nshx nshx 453 Dec 14 03:28 .config.users.json.backup
-rw-r--r-- 1 nshx nshx 16015 Dec 14 08:45 .flows.json.backup
-rw-r--r-- 1 nshx nshx 51 Dec 14 03:35 .flows_cred.json.backup
drwxr-xr-x 1 nshx nshx 4096 Dec 14 03:32 DLM
-rw-r--r-- 1 nshx nshx 16015 Dec 14 08:45 flows.json
-rw-r--r-- 1 nshx nshx 131 Dec 14 03:35 flows_cred.json
drwxr-xr-x 1 nshx nshx 4096 Dec 14 03:27 lib
drwxr-xr-x 1 nshx nshx 4096 Dec 14 03:29 node_modules
-rw-r--r-- 1 nshx nshx 46975 Dec 14 03:29 package-lock.json
-rw-r--r-- 1 nshx nshx 503 Dec 14 03:29 package.json
-rw-r--r-- 1 nshx nshx 20290 Dec 14 03:27 settings.js
nshx@DESKTOP-KDLAD82:~/node-red$ cd DLM/logs/
nshx@DESKTOP-KDLAD82:~/node-red/DLM/logs$ cat logs.json
{"topic":"/dln/logs","rowData":"01100011","time":"2021-12-14 10:35:27","camera":"back","type":"critical","data":"velo"}
nshx@DESKTOP-KDLAD82:~/node-red/DLM/logs$ cat logs.json
{"topic":"/dln/logs","rowData":"01100011","time":"2021-12-14 10:35:27","camera":"back","type":"critical","data":"velo"}
{"topic":"/dln/logs","rowData":"01100011","time":"2021-12-14 10:38:26","camera":"back","type":"critical","data":"velo"}
{"topic":"/dln/logs","rowData":"00010111","time":"2021-12-14 10:38:27","camera":"front","type":"warning","data":"vitesse"}
nshx@DESKTOP-KDLAD82:~/node-red/DLM/logs$

nshx@DESKTOP-KDLAD82: ~
1639474518: Sending PINGRESP to nodered_31b6d9b069ab08bd
1639474527: Received PUBLISH from nodered_31b6d9b069ab08bd (d0, q0, r0, m0, '/dln/display/', ... (4 bytes))
1639474527: Sending PUBLISH to mosqsub|15494-DESKTOP-K (d0, q0, r0, m0, '/dln/display/', ... (4 bytes))
1639474550: Received PINGREQ from mosqsub|15494-DESKTOP-K
1639474550: Sending PINGRESP to mosqsub|15494-DESKTOP-K
1639474587: Received PINGREQ from nodered_31b6d9b069ab08bd
1639474587: Sending PINGRESP to nodered_31b6d9b069ab08bd
1639474610: Received PINGREQ from mosqsub|15494-DESKTOP-K
1639474610: Sending PINGRESP to mosqsub|15494-DESKTOP-K
1639474647: Received PINGREQ from nodered_31b6d9b069ab08bd
1639474647: Sending PINGRESP to nodered_31b6d9b069ab08bd
1639474669: Received PINGREQ from mosqsub|15494-DESKTOP-K
1639474669: Sending PINGRESP to mosqsub|15494-DESKTOP-K
1639474706: Received PUBLISH from nodered_31b6d9b069ab08bd (d0, q0, r0, m0, '/dln/display/', ... (4 bytes))
1639474706: Sending PUBLISH to mosqsub|15494-DESKTOP-K (d0, q0, r0, m0, '/dln/display/', ... (4 bytes))
1639474707: Received PUBLISH from nodered_31b6d9b069ab08bd (d0, q0, r0, m0, '/dln/display/', ... (7 bytes))
1639474707: Sending PUBLISH to mosqsub|15494-DESKTOP-K (d0, q0, r0, m0, '/dln/display/', ... (7 bytes))
1639474729: Received PINGREQ from mosqsub|15494-DESKTOP-K
1639474729: Sending PINGRESP to mosqsub|15494-DESKTOP-K

nshx@DESKTOP-KDLAD82: ~/node-red
14 Dec 08:45:12 - [info] Started flows
14 Dec 08:45:12 - [info] [mqtt-broker:9587a71d.b08928] Connected to broker: mqtt://localhost:1883
14 Dec 08:45:13 - [error] [http request:getUID] RequestError: getaddrinfo ENOTFOUND sys
14 Dec 09:23:50 - [info] [mqtt-broker:9587a71d.b08928] Disconnected from broker: mqtt://localhost:1883
14 Dec 09:25:18 - [info] [mqtt-broker:9587a71d.b08928] Connected to broker: mqtt://localhost:1883

nshx@DESKTOP-KDLAD82: ~
nshx@DESKTOP-KDLAD82:~$ mosquitto_sub -t "/dln/display/"
feu_R
velo
vitesse
velo
velo
vitesse
```

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

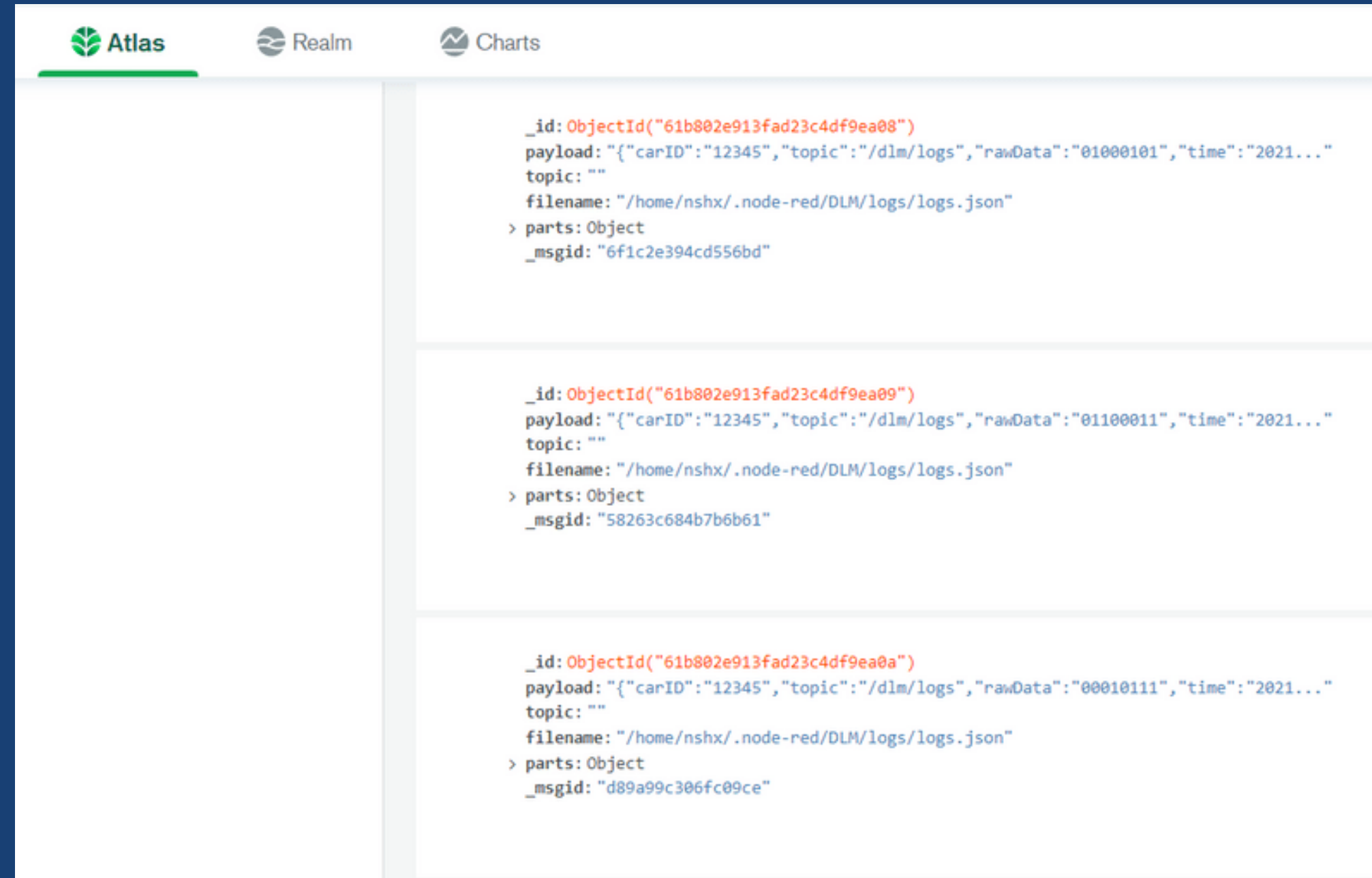
III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Gestion des alertes



III/ Contrôle moteur et implémentation Picar

Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Libraries de controle

Objectif

Controle moteur de la picar4WD et implémentation de l'algorithme de vision

Environnement

- Picar-4wd
- Raspberry PI4
- Python3

- Communication : SSH et SCP

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

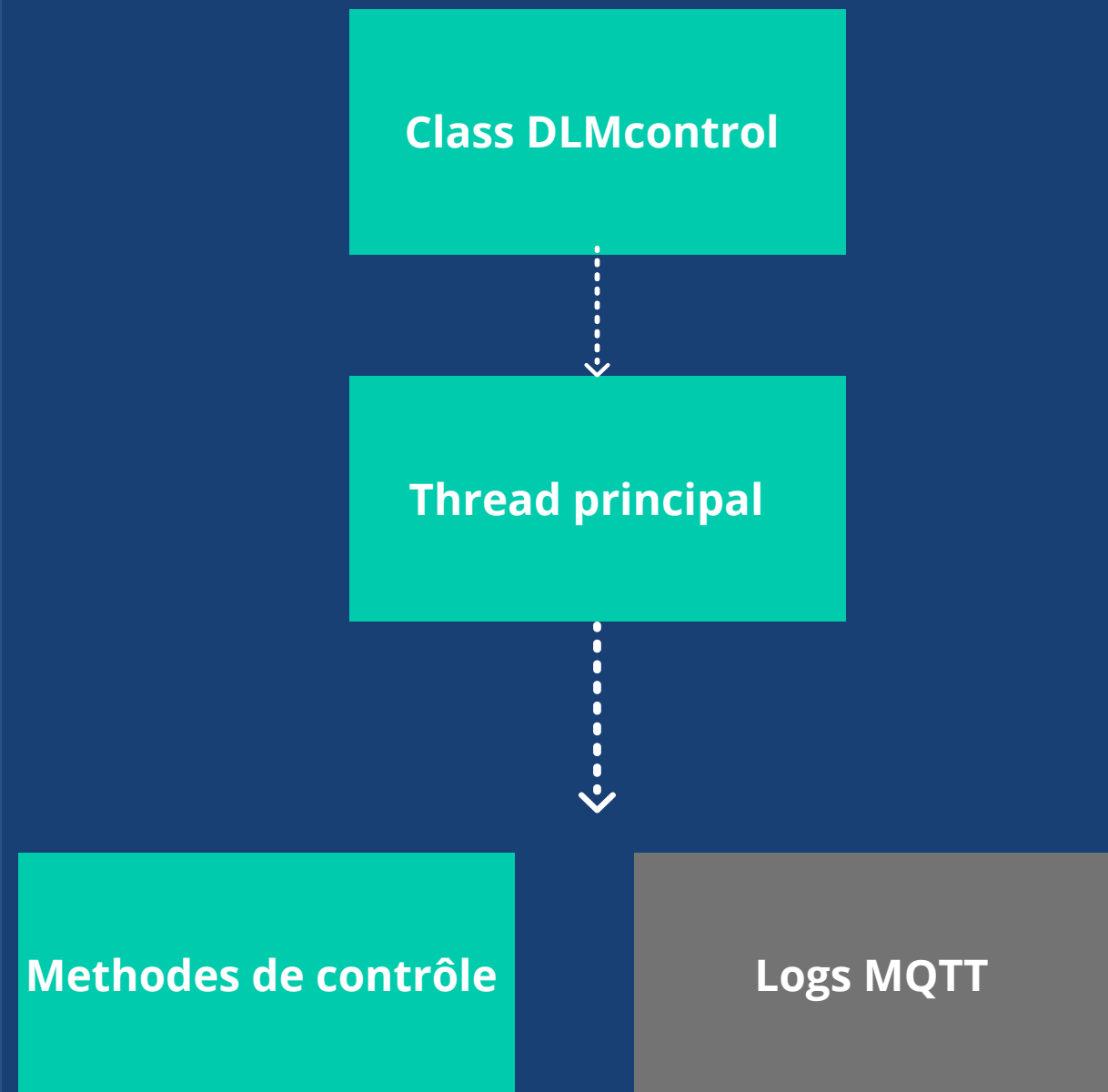
IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Libraries de controle

Architecture



Méthodes de Classe

- def sigListen(self):
- def sigSend(self):

- def start(self)
- def stop(self)
- def forwardspeed(self, speed)
- def turn(self, direction)

- def EcoMode(self):

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Libraries de controle

Exemple de code & Démo

```
def sigListen(self):  
  
    while True :  
        warning = self.sig_warning  
        if (warning == 0) :  
            self.speed = 15  
            #self.stop()  
            print("LOG : current speed")  
        elif (warning == 1):  
            #self.forwardspeed(5)  
            self.speed = 10  
            print("LOG : Speed reduce speed", self.speed)  
        elif (warning == 2):  
            self.speed = 5  
            #self.forwardspeed(5)  
            print("LOG : DANGER STOP ", self.speed)  
        elif (warning == 3):  
            self.speed = 0  
            #self.forwardspeed(5)  
            print("LOG : DANGER STOP ", self.speed)  
        time.sleep(2)
```

```
def sigSend(self, sig_warning, payload):  
  
    self.payload = payload  
    self.sig_warning = sig_warning  
    print("LOG : Signal sent level",sig_warning, payload)
```

Méthode d'envoi

Thread d'écoute

IV/ Affichage pour Dashboard



Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Affichage pour Dashboard

Problématique

Afficher les informations reçues/traitées à un conducteur?

Solutions

- Un système HUD (Head-up Display)
- Affichage de la dernière alerte
- Icones pour simplifier l'affichage

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Affichage pour Dashboard

Solution Sélectionnée

Application mobile sous android studio en Java

Objectifs

- Afficher
- Interfacier
- Faciliter
- Anticiper

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Affichage pour Dashboard

Application mobile

Création d'un nouveau client MQTT

```
client = new MqttAndroidClient(this.getApplicationContext(), MQTTHOST, clientId);
```

```
IMqttToken token = client.connect(options);
```

Tentative de connexion au broker

```
static String MQTTHOST = "tcp://localhost:1883";  
String topicStr = "/d1m/display/";
```

Paramètres de connexion

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Affichage pour Dashboard

Application mobile

```
public void messageArrived(String topic, MqttMessage message) throws Exception {  
    subText.setText(new String(message.getPayload()));  
}
```

Remplacement du texte de l'IHM

Cependant :

- Application non adapté dans le cadre de notre projet
- Implémentation impossible sur un serveur local
- **Solution** : changement sur un serveur Web

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Affichage pour Dashboard

Application WebServer

Création d'un serveur en javascript et HTML.

```
app.listen(8080, function(req, res)
```

```
function onConnect()
```

```
function onMessageArrived(res, topic, message)
```


Conclusion

Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

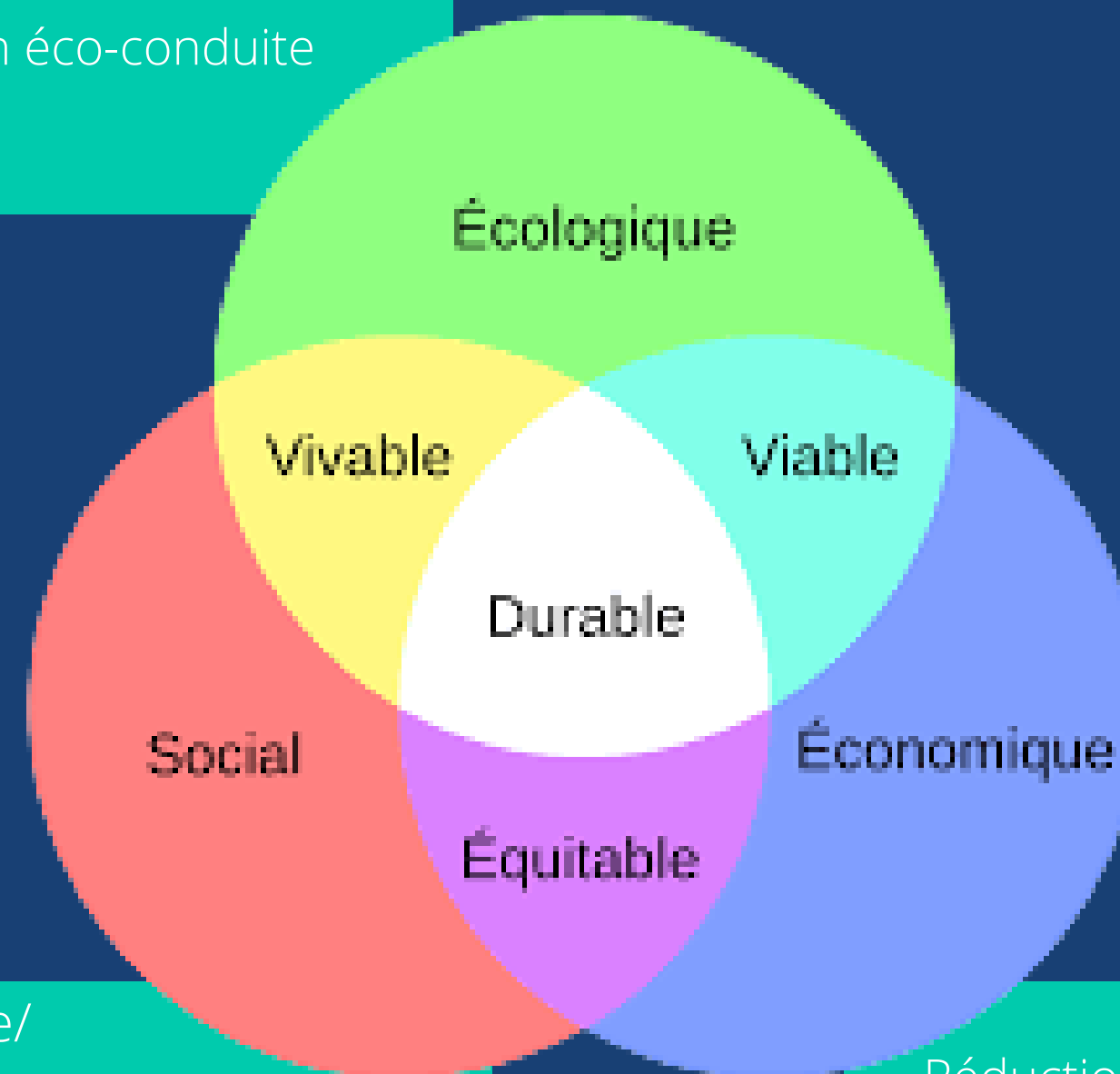
IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Conclusion

Optimisation d'énergie
- mode éco
Réduction de l'usure batterie
Prévention éco-conduite



Conduite plus responsable/
confortable
Aide de passage au permis malentendant
Protection des moniteurs (auto-école)
Réduction des accidents de la route

Réduction prévisionnelle de l'usure de
la batterie, des pneus et de l'essence.
Moins de déchets dû aux accidents

SOURCES

Projet de fin d'année

Plan

INTRODUCTION

I/ TRAITEMENT VIDÉOS

II/ GESTION DES ALERTES

III/ PICAR & CMD MOTEUR

IV/ AFF. DASHBOARD

CONCLUSION

SOURCES

Sources

- <https://joshtronic.com/2018/05/08/how-to-install-nodejs-10-on-ubuntu-1804-lts/>
- <https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/>
- <https://nodered.org/docs/user-guide/writing-functions>