



CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

# CarLoRA

Système embarqué - Pycom LoPy et ELM327

Paul Carretero, Trung Nguyen

Université Grenoble Alpes - IM<sup>2</sup>AG

9th April 2018



# Introduction

## Table des Matières

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

- 1 Introduction
  - Table des matières
  - Présentation du Projet
- 2 **Projet CarLoRA**
  - Architecture théorique
  - Interface Jyse
- 3 **Spécifications Techniques**
  - ELM327 OBD2
- 4 **Réalisation du projet**
  - Carte Embarqué LoPy
  - Mise en place
  - Problèmes rencontrés
- 5 **Blockchain Ethereum**
  - Principes
  - Mise en place
- 6 **Conclusion**



# Introduction

## Projet CarLoRA

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

But

- Récupération des données de conduite.
- Analyse et sauvegarde.
- Visualisation des données.

Techonologies

- Carte Lopy
- ELM327 OBD2
- LoRA
- Blockchain Ethereum

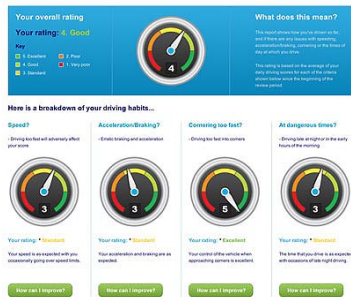


Figure – Exemple de dashboard 'Pay How You Drive'



# Projet CarLoRA

## Architecture Théorique

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

### Partie embarquée

- ELM327 sur le port ODB2
- LoPy en Bluetooth
- Envoi en LoRA

### Partie Back-End

- Broker MQTT en LoRA
- Jyse
- Back-end Java
- Sauvegarde sur une blockchain



Figure – Schéma d'une architecture LoRA IOT



# Projet CarLoRA

## Interface Jyse

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

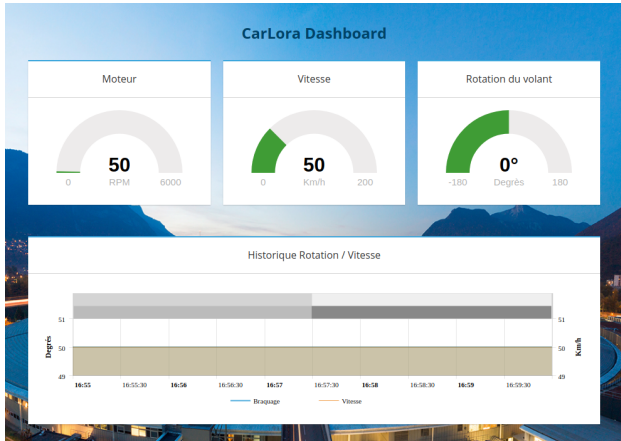


Figure – Interface Jyse

# Spécifications Techniques

## ELM327 et OBD2



CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

### OBD2

- Norme Diagnostique auto
- Différents modes/protocoles
- Code d'erreur
- Valeurs Véhicule

### ELM327

- Se branche sur un port ODB2.
- transmet les informations en Bluetooth, Wifi etc.
- Apps Smartphone.

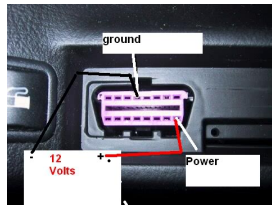


Figure – un port OBD2 et un module ELM327

# Spécifications Techniques

## Carte Embarqué LoPy



CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

**Spécifications  
Techniques**

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

### Caractéristiques

- RAM : 512 KB
- Flash: 4MB
- 3,3 à 5,5 Vcc
- 24 broches GPIO
- WiFi, BLE, LoRA

### Développement

- MicroPython
- plugin Pymakr VSCode
- Bibliothèques BETA



Figure – Carte LoPy et sa board  
PyTrack



# Réalisation du projet

Mise en place

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

## 3 variables de conduite

- La vitesse
- RPM Moteur
- Rotation volant
- => 3 sous-sujet MQTT

## Back-End serveur

- Java
- MQTT Eclipse Paho.
- Mosquitto (dev).
- S'abonne aux sujets.
- Sauv. dans la blockchain.

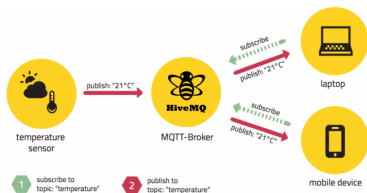


Figure – Exemple de fonctionnement MQTT





# Réalisation du projet

## Problèmes rencontrés

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

### Analyse du problème

- LoPy : BLE
- ELM327: Bluetooth Classique
- Variations des spécifications
- Incompatibilité

### Solution choisie

- Simulation des données
- Publish MQTT
- Contrainte de temps

| Specifications                               | Bluetooth   | BLE(Bluetooth Low Energy)  |
|--|---|--|
| Network/Topology                             | Scatternet  | Star Bus   |
| Power consumption                            | Low (less than 30 mA)                                       | Very Low (less than 15 mA)   |
| Speed  | 700 Kbps  | 1 Mbps   |
| Range  | <30 m   | 50 meters( 150 meters in open field)   |
| RF Frequency band                            | 2400 MHz  | 2400 MHz   |
| Frequency Channels                           | 79 channels from 2.400 GHz to 2.4835 GHz with 1 MHz spacing | 40 channels from 2402MHz to 2480 MHz (includes 3 advertising and 37 data channels) |
| Modulation                                   | GFSK (modulation index 0.35) , π/4 DQPSK, 8DPSK             | GFSK (modulation index 0.5)  |
| Latency in data transfer between two devices | Approx. 100 ms  | Approx. 3 ms   |
| Spreading                                    | FHSS (1MHz channel)   | FHSS (2MHz channel)  |
| Link layer                                   | TDMA  | TDMA   |
| message size(bytes)                          | 358 (Max)   | 8 to 47  |
| Error detector/correction                    | 8 bit CRC(header), 16 bit CRC, 2/3 FEC(payload), ACKs       | 24 bit CRC, ACKs   |
| Security                                     | 64b/128b, user defined application layer                    | 128 bits AES, user defined application layer                                       |
| Application throughput                       | 0.7 to 2.1 Mbps   | less than 0.3 Mbps   |
| Nodes/Active Slaves                          | 7   | Unlimited  |

Figure – Bluetooth vs BLE



# Blockchain Ethereum

## Principes

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

### Généralités

- Créé en 2014, par Vitalik Buterin
- Datastore Distribué P2P
- Ethereum Virtual Machine (« EVM »).

### Smart Contracts

- Usages multiples
- Contrats auto-exécutant
- Fiable et sécurisé
- Solidity

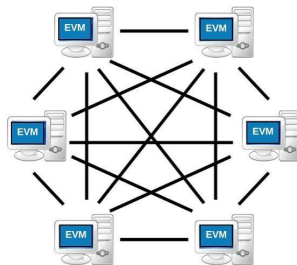


Figure – Réseau Ethereum privé



# Blockchain Ethereum

Mise en place: Docker

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

## Technologies

- 2 Containers Docker
- Réseaux virtuel
- Geth (CLI Ethereum)
- Web3J pour le back-End

## Smart Contract

- 1 Simple Smart Contract
- Données de conduite
- Timestamp
- Immutable

```
1 pragma solidity ^0.4.0;
2 contract Record {
3
4     uint timestamp;
5     uint rpm;
6     uint rot;
7     uint spd;
8
9     function Record(uint _timestamp, uint _rpm, uint _rot, uint _spd)
10        public {
11         timestamp = _timestamp;
12         rpm = _rpm;
13         rot = _rot;
14         spd = _spd;
15     }
16
17     function getTimestamp() public constant returns (uint _timestamp)
18     {
19         _timestamp = timestamp;
20     }
21
22     function getRpm() public constant returns (uint _rpm) {
23         _rpm = rpm;
24     }
25
26     function getRot() public constant returns (uint _rot) {
27         _rot = rot;
28     }
29
30     function getSpd() public constant returns (uint _spd) {
31         _spd = spd;
32     }
33 }
```

Figure – Smart Contract utilisé (Solidity)



# Conclusion

CarLoRA

Carretero,  
Nguyen

Introduction

Projet  
CarLoRA

Spécifications  
Techniques

Réalisation  
du projet

Blockchain  
Ethereum

Conclusion

## Technologies mises en places

- MQTT
- MicroPython
- Ethereum
- Jyse

## Limites

- Ethereum très simple.
- Peu de données traitée.
- Projet avorté sur le plan technique.