



# Light Traffic

AUCLAIR William, LENOIR Adrien, 1G2TP5



# Plan

- Contexte
- Objectifs
- Principe de fonctionnement
- Ressources
- Contraintes
- Algorithmes
- Implémentation & Maquette
- Echechs
- Améliorations
- Conclusion



# Contexte

Assurer la sécurité des usagers (piétons, automobilistes) en régulant la circulation à proximité d'une intersection en T entre deux routes principale et secondaire à l'aide de feux tricolores.

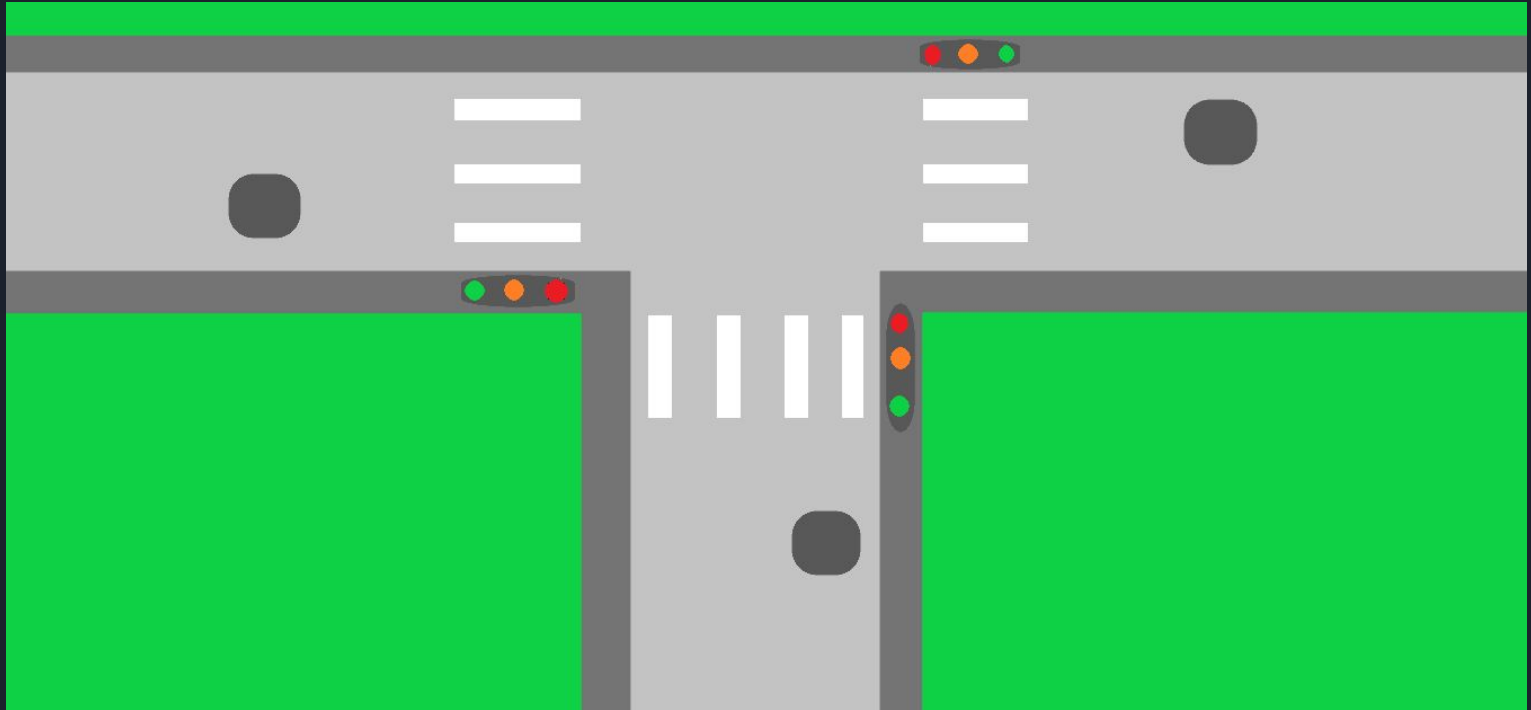


# Objectifs

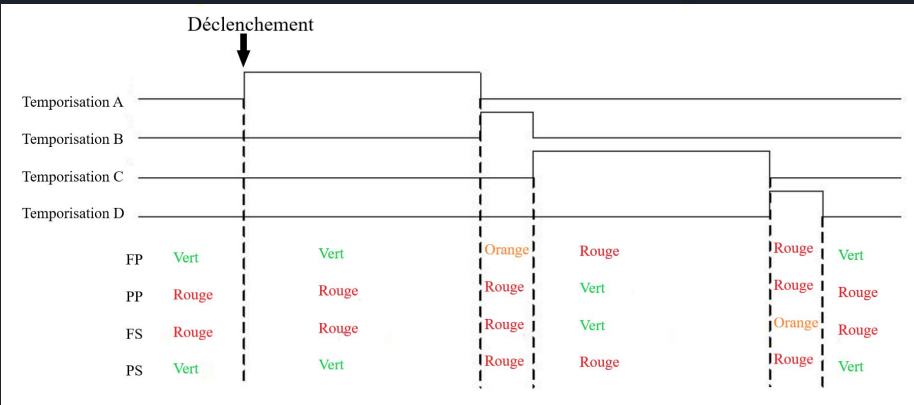
Contrôler la circulation à l'aide de feux tricolores munis de :

- Capteurs de présence afin de détecter les usagers ;
- De chronomètres afin d'afficher le temps d'attente ;

# Principe de fonctionnement



# Principe de fonctionnement



Présence (P)	Présence (S)	Feux (P)	Feux (S)
OUI + était Vert	OUI, NON + était Rouge	Vert	Rouge + Tempo (30s)
OUI + était Rouge	OUI, NON + était Vert	Rouge + Tempo (5s)	Jaune + Tempo (5s)
OUI + était Rouge	OUI, NON + était Jaune	Vert	Rouge + Tempo (30s)
NON + était Vert	OUI + était Rouge	Jaune + Tempo (5s)	Rouge + Tempo (5s)
NON + était Rouge	OUI + était Vert	Rouge + Tempo (30s)	Vert
NON + était Jaune	OUI + était Rouge	Rouge + Tempo (30s)	Vert
NON + était Vert	NON + était Rouge	Vert	Rouge + Tempo (30s)
NON + était Rouge	NON + était Vert	Vert	Rouge + Tempo (5s)
NON + était Rouge	NON + était Jaune	Vert	Rouge + Tempo (30s)



# Ressources

## Langages / Logiciels :

- ASM
- C++

## Matériel :

- Afficheurs 7 segments (x6)
- Carte STM32F407
- Détecteurs de présence (x3)
- LEDs (x9)

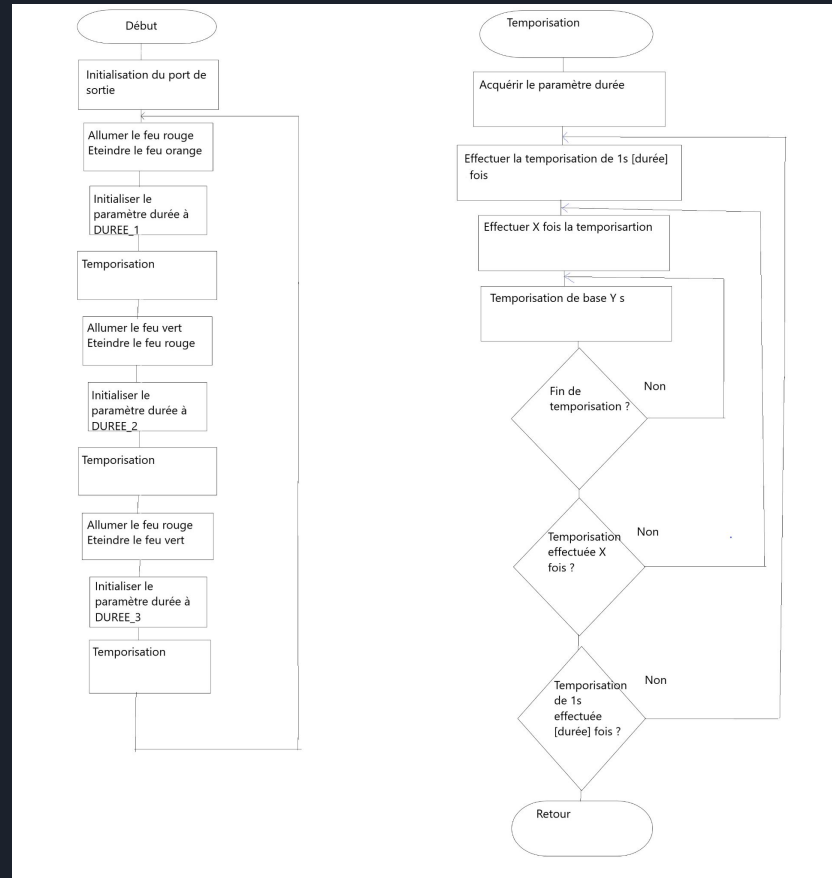


# Contraintes

- Seuil de détection < 3 m avant le feu
- Temps d'attente < 5 OU 30 s
- Respect du budget imparti (40€)
- Respect de la deadline



# Architecture





# Main et initialisation de la carte

main :

[main.c](#)

initialisation :

[405 DISC.c](#)

```
typedef struct feu{
    int id;
    int cpt_d;
    int cpt_u;
    int vert;           //Contient le booleen correspondant à l'allumage du feu vert.
    int jaune;          //De même pour le feu jaune.
    int rouge;          //De même pour le feu rouge.
}FEU;
```

# Initialisation des ports, lecture/écriture

Fichier C :

[Initialisations.c](#)

Entête :

[Initialisations.h](#)

Pins utilisés	Utilisation	Pins utilisés	Utilisation	Pins utilisés	Utilisation
PA0	Anode 0 Diz	PB0	Cathode Diz 1	PC0	Capteur 1 (P1)
PA1	Anode 1 Diz	PB1	Cathode Diz 2	PC1	Capteur 2 (P2)
PA2	Anode 2 Diz	PB2	Cathode Diz 3	PC2	Capteur 3 (S)
PA3	Anode 3 Diz	PB3	Cathode Unit 1	PC3	
PA4	Anode 4 Diz	PB4	Cathode Unit 2	PC4	
PA5	Anode 5 Diz	PB5	Cathode Unit 3	PC5	
PA6	Anode 6 Diz	PB6	LED verte 1	PC6	
PA7	Anode 0 Unit	PB7	LED Jaune 1	PC7	
PA8	Anode 1 Unit	PB8	LED Rouge 1	PC8	
PA9	Anode 2 Unit	PB9	LED verte 2	PC9	
PA10	Anode 3 Unit	PB10	LED Jaune 2	PC10	
PA11	Anode 4 Unit	PB11	LED Rouge 2	PC11	
PA12	Anode 5 Unit	PB12	LED verte 3	PC12	
PA13	Anode 6 Unit	PB13	LED Jaune 3	PC13	
PA14		PB14	LED Rouge 3	PC14	
PA15		PB15		PC15	

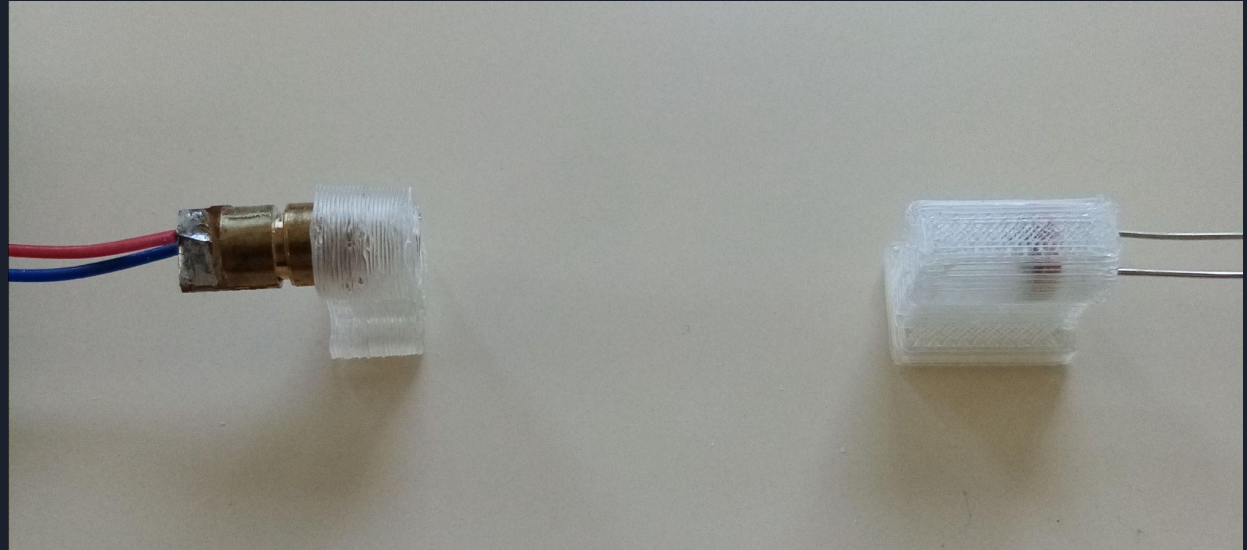
# Détection de présence

Fichier C :

[detection.c](#)

Entête :

[detection.h](#)





# Contrôle des LEDs

Fichier C :

[LEDs.c](#)

Entête :

[LEDs.h](#)

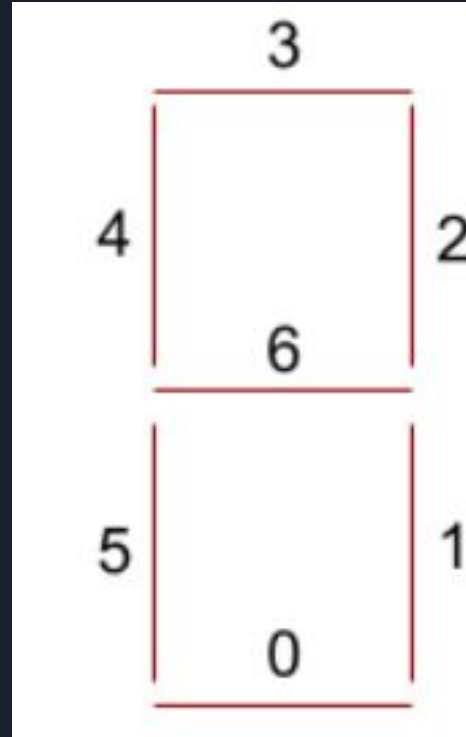
# Affichage du temps d'attente

Fichier C :

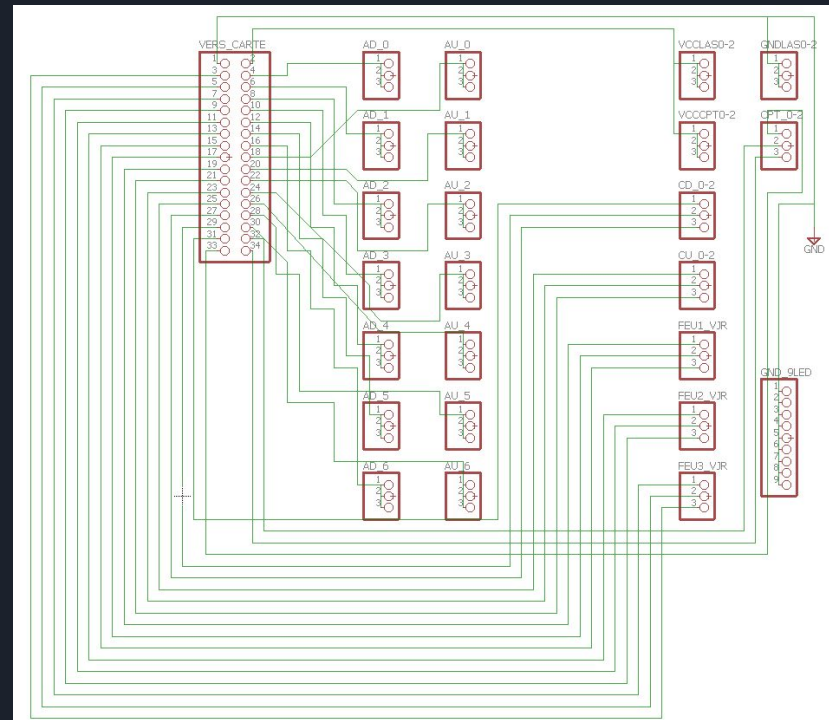
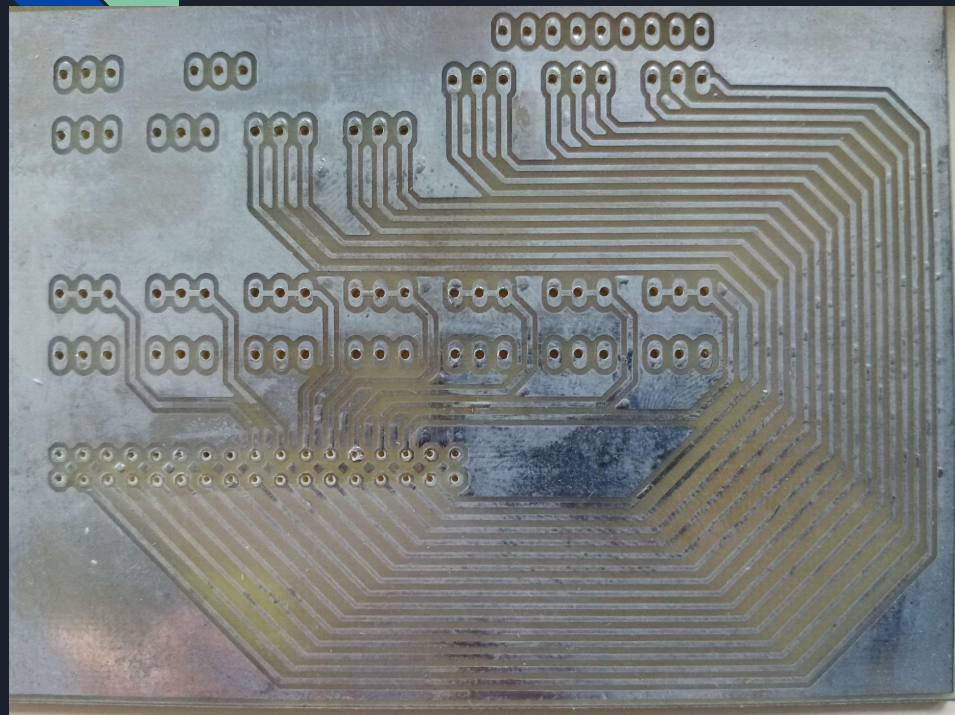
[Decompte.c](#)

Entête :

[Decompte.h](#)



# Circuit imprimé

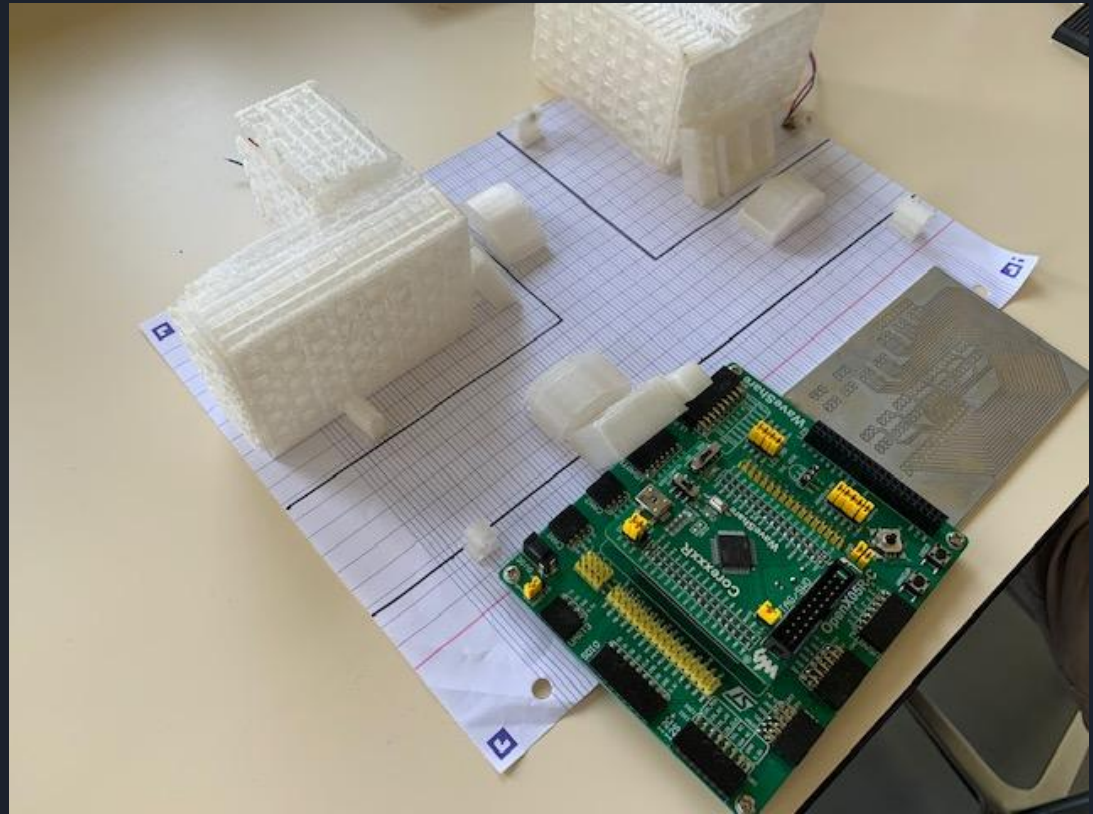


# Maquette

Dimensions :

Plateau : 25x25 cm

Largeur de la route : 5 cm







# Echecs & difficultés

- Le projet n'a pas pu aboutir (La maquette n'ayant été que partiellement achevée)
- Problèmes d'impression 3D
- Obtention de la liste des ports GPIOs disponibles



# Améliorations

- Alimenter le réseau à l'aide de panneaux solaires ;
- Concevoir un modèle adapté à tous les usagers (piétons, ...) ;



# Conclusion

- Expérience stimulante & accessible
- Meilleure gestion du temps
- Cependant approfondissement de l'apprentissage des langages ASM et C et amélioration de la prise en main de Eagle et Keil