



# Projet d'électronique

## Robot Distanciation sociale

GROUPE 10

Encadré par : Francis Bras

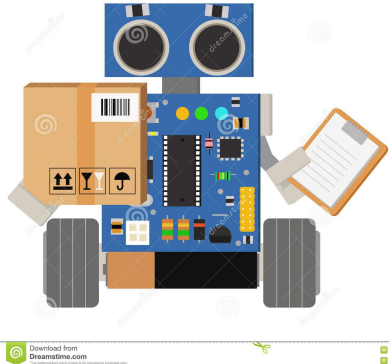


# Plan

- I. Description du projet
- II. Présentation des éléments embarqués sur le robot et leurs rôles
- III. Présentation de l'identifiant du robot:
  - Emetteur/Récepteur Infrarouge
- IV. Difficultés et problèmes rencontrés
- V. Conclusion

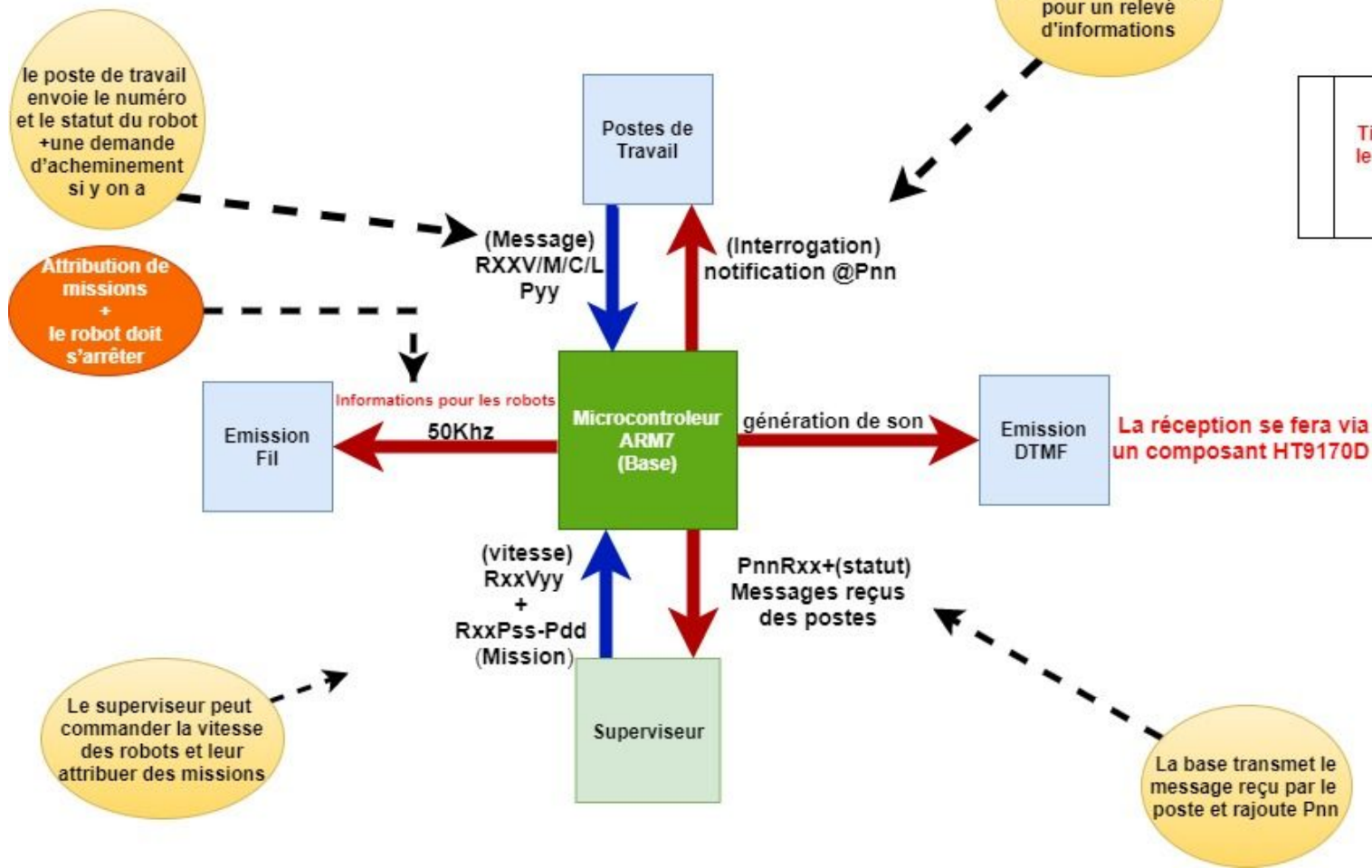
# But du projet :

Création d'un système évitant les interactions entre ouvriers lors d'une épidémie



# Fonctionnement du système:

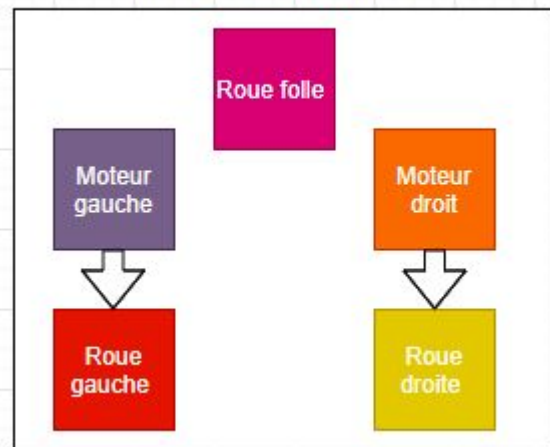
- ❑ Utilisation de robots livreurs sur un parcours en boucle défini par un fil électrique scotché au sol
- ❑ Commandes injecté dans le fil
- ❑ Composition du système: -Postes ouvriers - La base -Robot
- ❑ Modulation du courant dans le fil
- ❑ Communication par liaison série
- ❑ Gestion des différents statuts du robot :
  - Disponible
  - Missionné
  - Colis enlevé
  - En livraison
- ❑ DTMF
- ❑ Infra rouge



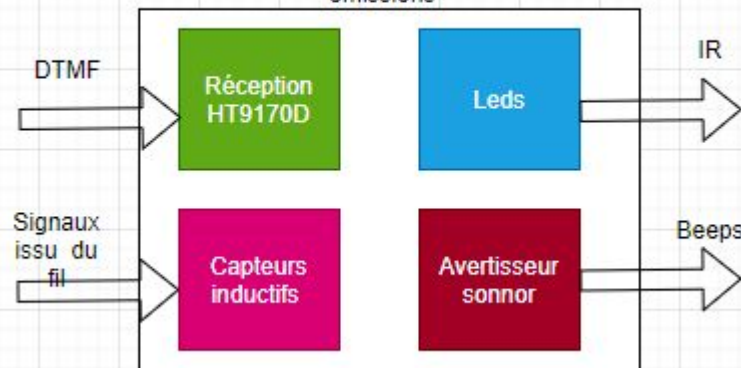
Titre:Environnement dont  
lequel circulent les robots  
Groupe 10

# Composants embarqués sur le robot avec leurs rôles

Système de déplacement



Systèmes de réception et de émissions



Micro-contrôleur

Télémètre ultrason

1) Traiter les données issu:

Du fil

D'autre composant de réception

2) Donner des consignes pour:

Codage de statut robot

Pour les moteurs

L'avertisseur sonore

Son rôle est de détecter les obstacles

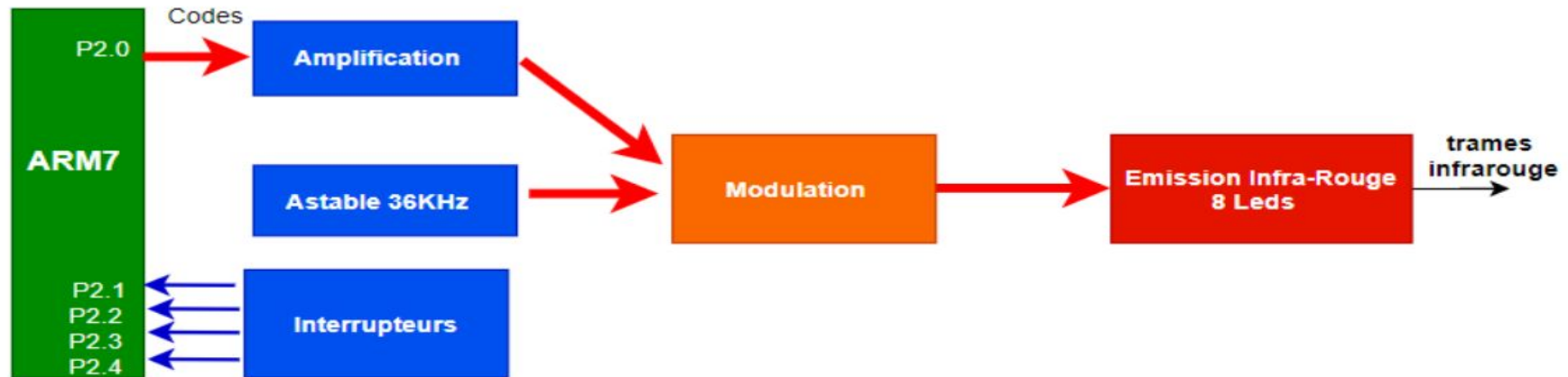
# Les différents boutons embarqués sur le





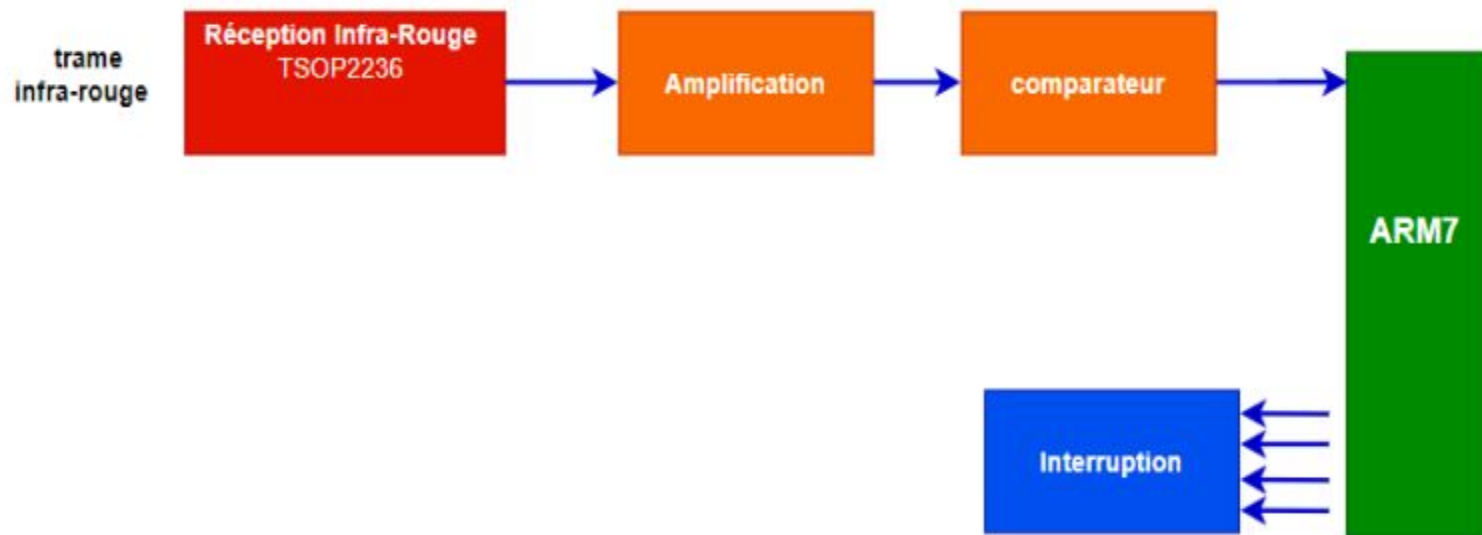
# Transmission de l'identifiant du robot

Infra-Rouge (émission)



\*

## Infra-Rouge (réception)



# Partie Programmation et simulations : IR

1. Le but est de générer une trame d'impulsions '0' et '1' logic
2. Utiliser les sorties PWM pour générer des signaux rectangulaires avec une facilité de modulation
3. utilisation du Pin P1.21 en double edge

```
31 //-----
32 void init_PWM1_2(){
33   PWM1MR2= 1; // set (front montant )
34   PWM1MR3= 24; // reset (front descendant) => 2µs niveau haut
35 }
36
37 void init_pwm() // utilisation de PWM1.3 en single edge => P1.21
38 {
39   PWM1MR0= 3360; // 280 µs => Tho
40   PWM1MR1= 333; // 28 µs
41
42   PWM1LER= 0xF; // activation de MR0, MR1, MR2 , MR3
43   PWM1TCR= 0X9; // cpt on + activation mode PWM
44   PWM1PCR= 0xE08; //mode double edge + activation
45   //PWM1MCR= 0x249; // PWM1MR0 ir , PWM1MR1 et PWM1MR2 et PWM1MR3 interruption
46   PWM1MCR= 0x24B; // PWM1MR0 ir + rst, PWM1MR1 et PWM1MR2 et PWM1MR3 interruption
47   PINSEL3= 10<<10; // mode PWM
48   FIO1PIN= 1<<21; // initialisaer à 1 la sortie P1.20
49
50   VICVectAddr8= (unsigned long) IT_PWM;
51   VICIntEnable =1<<8; // activation de canal
52 }
```

# Partie Programmation et simulations : IR

## Pour générer une trame

1. générer des impulsions par la fonction motif0() avec une période  $28\mu\text{s}$  et  $HL = 2\mu\text{s}$
2.  $T=[150\mu\text{s}, 300\mu\text{s}] \Rightarrow$  choix :  $280\mu\text{s}$  pour  $\Rightarrow$  10 impulsions avec  $T = 28\mu\text{s}$
3. utiliser le mode interruption PWM

```
7 // déclaration de fonctions :
8 void detect_front();
9 void init_PWM1_2();
10 void zero_logic();
11 void zero();
12 void un_logic();
13
14 int motif0() { // fonction qui génère les impulsions
15     entree++; // vérifier le nombre d'entree
16     if(entree==1) {FIO1PIN^=1<<21; PWM1IR=2;VICVectAddr=0;} // set
17     if(entree==2) {FIO1PIN^=1<<21; PWM1IR=8;VICVectAddr=0;} // reset 0xFFDFFFFFF;
18     if(entree==3) {init_PWM1_2();i++;}
19     if(entree==4) {entree=0;PWM1IR=1;VICVectAddr=0;}
20
21 }
22
```

```
26 void IT_PWM(void )_irq { // si PWM1PC= PWM1MR0 ou PWM1MR1 ou PWM1MR2
27     if(PULSE_IR==1) un_logic(); // fonction motif
28     else zero_logic();
29     VICVectAddr=0;
30 }
31 //-----
```

# Partie Programmation et simulations : IR

## Pour générer '0' et '1' logic

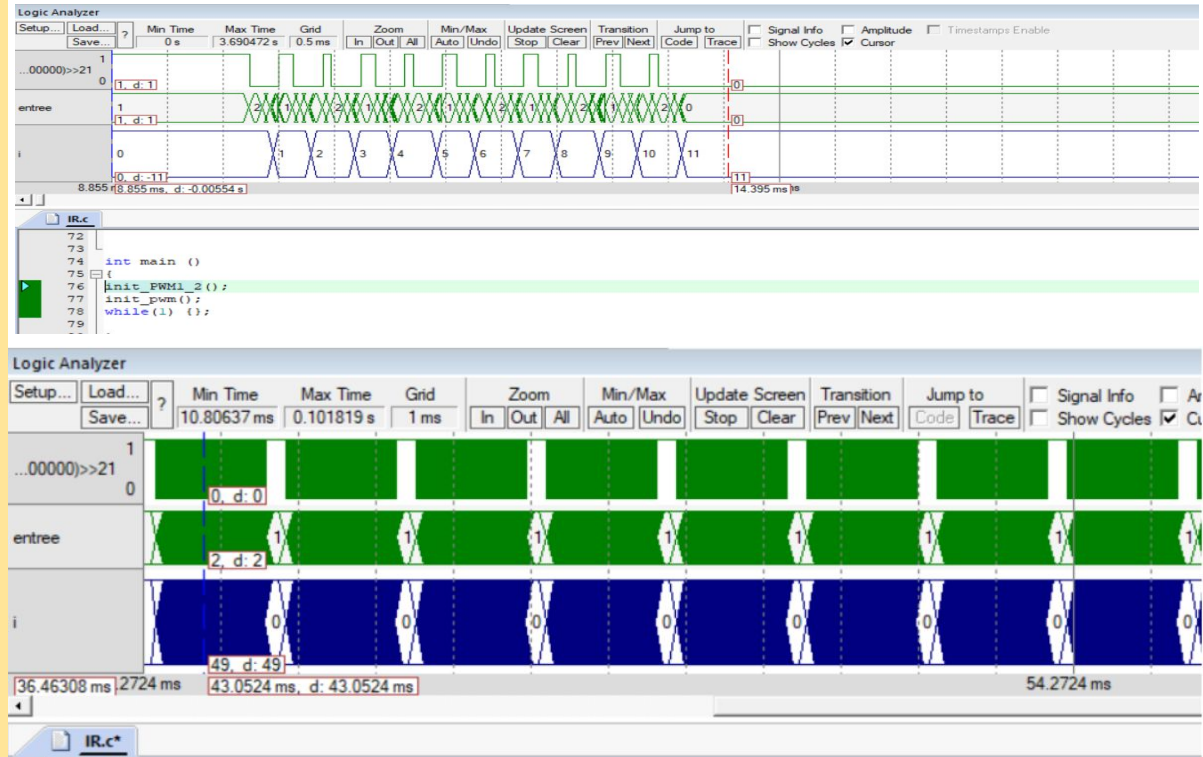
1. zéros logic '0' :  
est représenté : 100  
1 => 10 impulsions  
pendant  $\tau$   
0 => un blanc pendant  $\tau$   
0 => un blanc pendant  $\tau$
2. un logic '1' :  
est représenté : 110  
2x 1 => 20 impulsions  
pendant 2x  $\tau$   
0 => un blanc pendant  $\tau$

```
71 }
72 // fonction un logic
73 void un_logic(){
74
75 if(i<=40) {motif0();VICVectAddr=0;} // 20 impulsions 2X plus large que dans le cas de "0" logic
76 if(i>40 && i<60) {zero();if(entree==3)FIO1PIN=0<<21;entree=0;VICVectAddr=0;}
77 if(i>=60 && i<80) {motif0(); FIO1PIN=0<<21;i=0;VICVectAddr=0;}
78 }
79
80
81 // fonction zero logic 100 ( 1 dure un Th0 et 0 dure 2 Tho )
82 void zero_logic(){
83 j++;
84 if(i<=20) motif0(); // 10 impulsion
85 if(i>20 && i<60) {zero(); if(entree==3)FIO1PIN=0<<21;entree=0;VICVectAddr=0;} // un blanc 2x Large
86 if(i>=60 && i<=80) {motif0();i=0;VICVectAddr=0;} // 10 impulsions
87 }
```

# Partie Programmation et simulations : IR

## Simulation :

1. simulation de '0' logic
2. transmission d'une trame de '1' logic
3. '10' début de trame
4. 16 bits à transmettre  
=> CPT=15 décrémente  
jusqu'à 0 pour indiquer la fin  
de trame



# Difficultés et problèmes rencontrés

- Manque de temps
- Pas d'accès aux équipements électronique
- Difficulté de la communication entre les membres du groupe

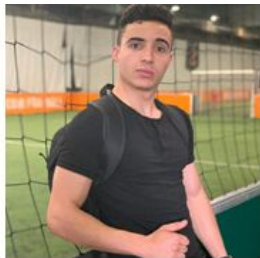
## CONCLUSION

Ce projet nous a permis de:

- Se familiariser avec le langage de programmation des microcontrôleurs.
- Compréhension et Interprétation d'un cahier de charge.
- Confronter des problèmes de réalisations et les résoudre
- Bonne interprétation des datasheets
- Gestion du stress et du temps
- Développement d'esprit d'équipe
- Un pas dans le monde professionnel







KADI Koussaila

- ✓ Ecran\_supervision
- ✓ Dessin grafcet status robot
- ✓ mae\_robot (vitesse)
- ✓ mae\_poste
- ✓ simul\_telem
- ✓ simul\_IR



MADI Ziri

- ✓ dessin\_base
- ✓ bilan\_Esbase
- ✓ simul\_boutons



CHABANE Sabrina

- ✓ Dessin\_poste\_ouvrier
- ✓ Bilan\_Esposte
- ✓ Simul\_telem



BELALOUACHE Rachid

- ✓ dessin\_robot
- ✓ bilan\_Esrobot
- ✓ schm\_robot
- ✓ mae\_robot (recp DTMF)
- ✓ simul\_boutons

TORCHOUN Fouad

- ✓ dessin\_environn emecnt\_flux
- ✓ schm\_poste
- ✓ schm\_base
- ✓ Schema\_robot
- ✓ Schema\_telem

