

## **TELECOMMANDE HF ROBOT**

### **Spécifications Techniques**

#### **1 Présentation**

Cet élément fonctionnel du projet, permet de télécommander à distance, par le biais d'une liaison radio HF dans la bande 433MHz, les fonctions suivantes du Robot :

Arrêt (ou veille)

Déplacement Avant Arrière sur trajectoire ligne droite (à vitesse constante)

Tourne Droite Gauche sur une trajectoire circulaire (à vitesse constante)

La télécommande du Robot sera unidirectionnelle (pas de remontée d'information du Robot vers le bloc émetteur).

La distance de fonctionnement de la télécommande en espace libre est d'environ 20 mètres au maximum.

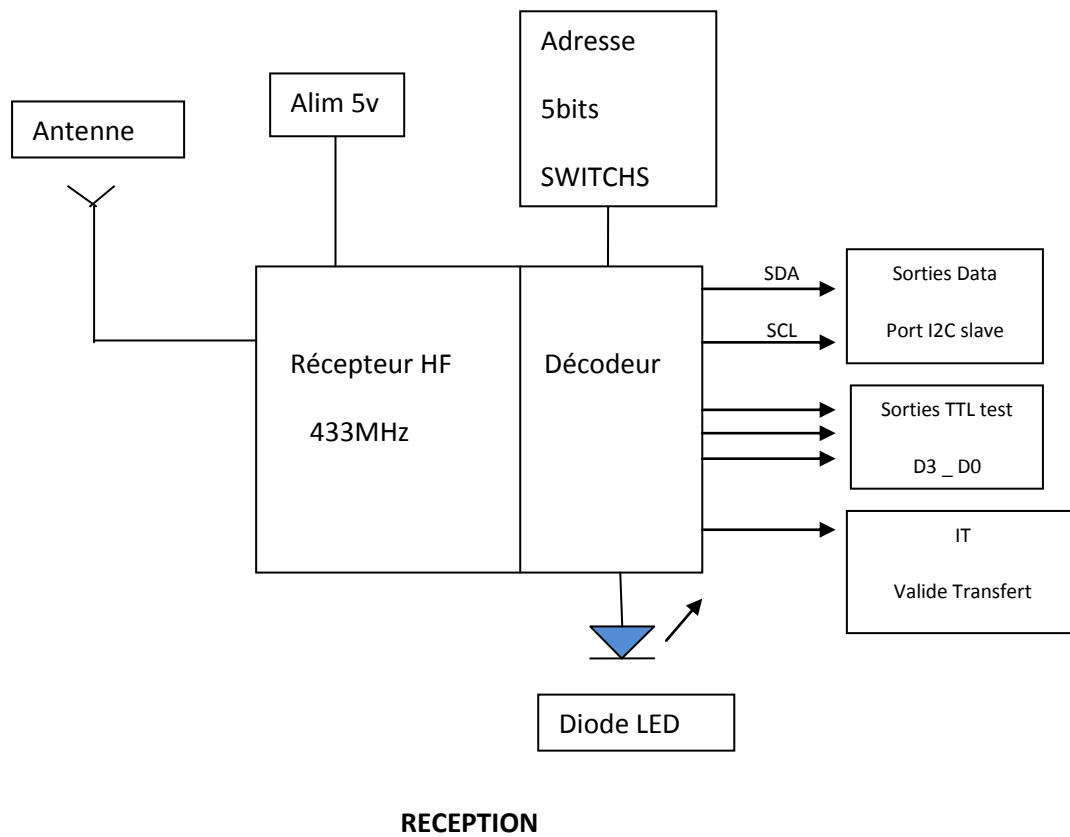
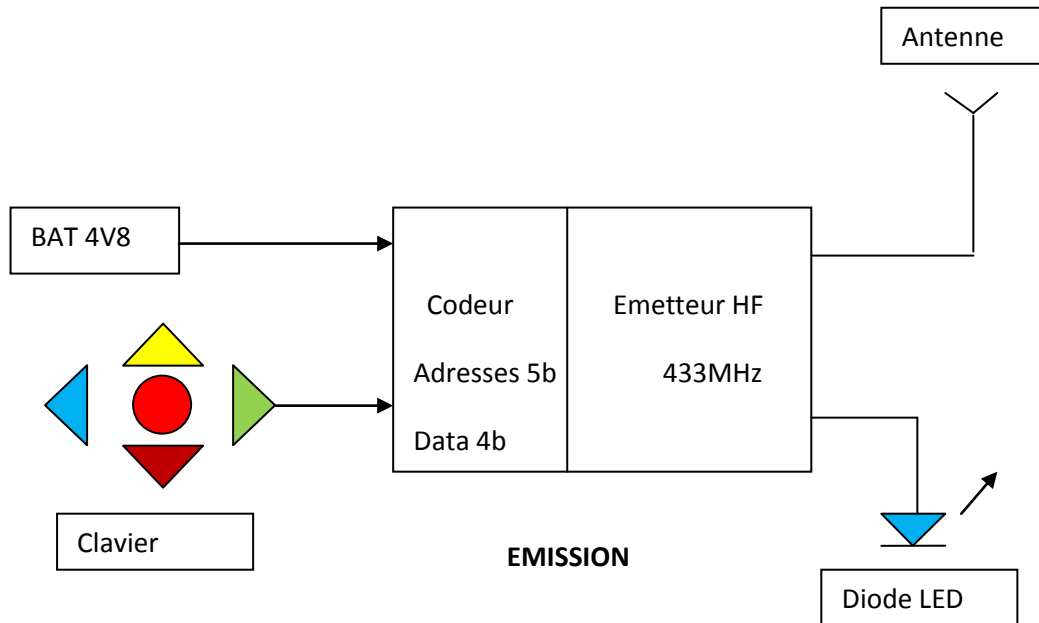
#### **2 Décomposition**

La télécommande sera composée des deux blocs fonctionnels suivants :

Un émetteur HF en boîtier de taille réduite auto alimenté par une batterie rechargeable de 4V8 150 mAH. Il possèdera 5 touches de commande, et un voyant de type LED de contrôle de la transmission des informations. Tous ces éléments sont situés en face avant. Une carte électronique de taille réduite effectuant le codage des commandes et un émetteur HF embarqué.

Une carte électronique de taille réduite comprenant un récepteur HF en entrée et un bloc décodeur, permettant de dialoguer avec l'unité centrale micro-programmée de commande et contrôle du Robot utilisant un port de données sur 4 bits et un signal de validation des data reçues.

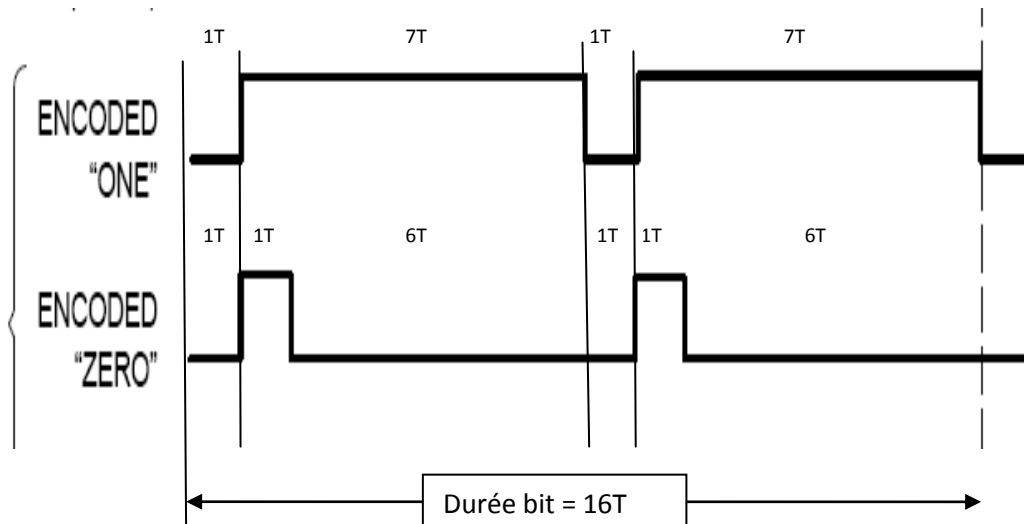
## 3 Schémas fonctionnels EMISION / RECEPTION



## 4 Description du bloc EMMISION

### 4-1 Robustesse de la transmission

Les informations numériques séries transmises en modulation OOK seront codées MICD selon le schéma prédéterminé suivant avec son horloge de base  $T = 250\mu s \pm 1\%$ .



Durée bit (Tbit) = 16T soit 4ms compatible de la bande passante du canal radio

La trame séquentielle Adresses Data est transmise MSB en tête (Big Indian), et dans cet ordre répétée une fois avec un intervalle de temps de 112T soit 14 ms.

Cette trame est répétée si la durée d'appui d'une touche est supérieur à la durée totale de la trame 400T soit 100 ms.

La combinaison d'appui de touche est interdite pas de transmission de donnée.

### 4-2 Adressage d'un récepteur parmi N (ADRESSES)

Au niveau de l'émetteur, un code adresse sur 5 bits ( $A4\_A0$ ) est imposé par programmation qui permet d'adresser dans un canal de transmission 1 récepteur ROBOT parmi 32 possibles.

Au niveau du récepteur une sélection de l'adresse du récepteur sur 5 bits par micro-switchs .

### 4-3 Codes d'informations transmises (DATA)

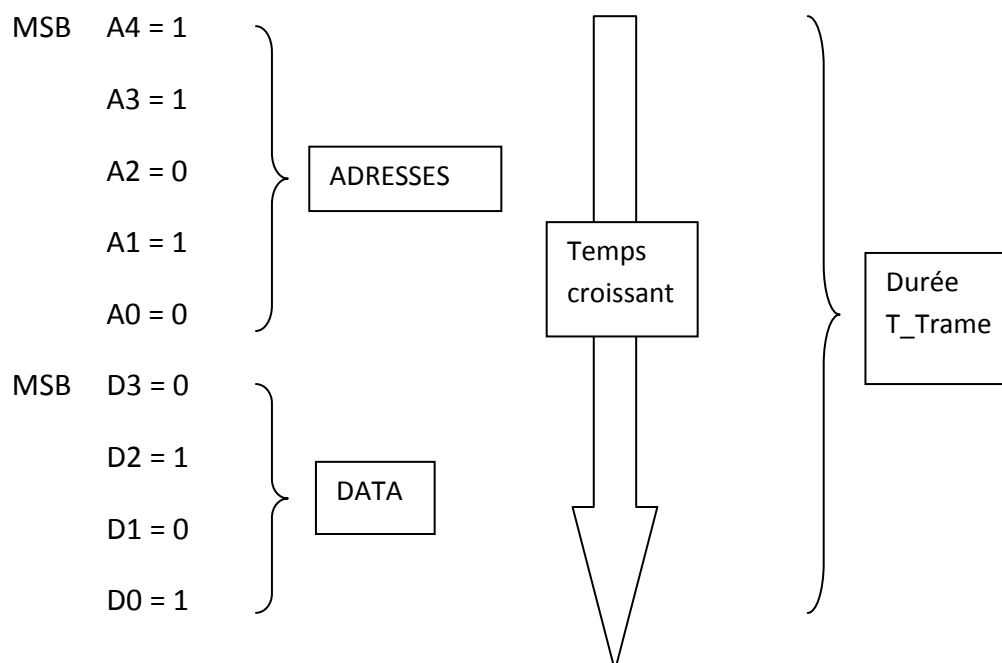
Les codes touches de la télécommande transmis tiendront sur 4 bits ( $D3\_D0$ ), leur représentation Hexa décimale est la suivante :

## Module Conception de Produit

Touche DROITE	1h
Touche AVANT	2h
Touche MARCHÉ/ARRET	3h
Touche ARRIERE	4h
Touche GAUCHE	5h

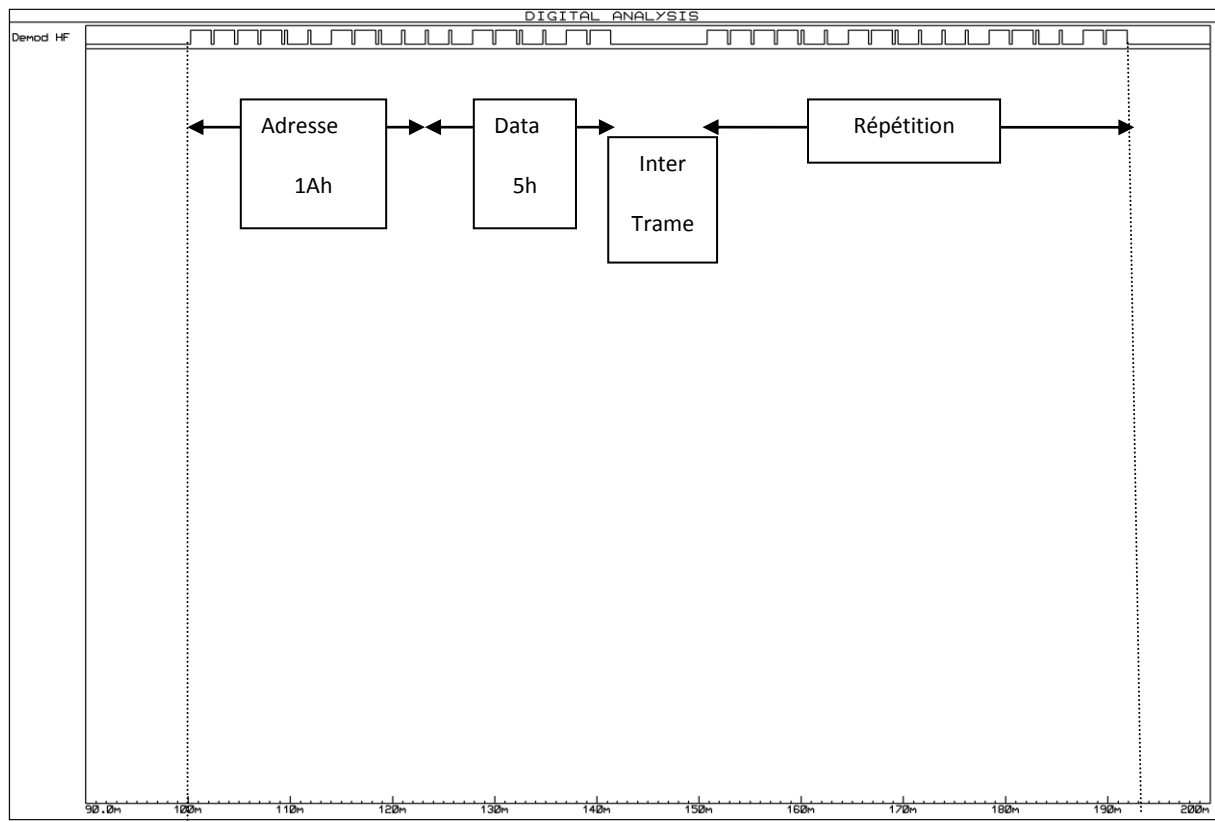
### EXEMPLE codage Adresse = 1Ah, touche GAUCHE

La trame exemple montre la transmission temporelle des données suivantes en mode Poids Forts en tête (Big Indian) :



# Module Conception de Produit

## Chronogramme codage Adresse 1Ah, touche GAUCHE



## **5 Description du bloc RECEPTION**

Le bloc réception se compose des deux éléments principaux suivants :

Un récepteur HF accordé sur la fréquence porteuse  $F_p = 443\text{MHz}$  qui fonctionne en démodulation OOK est qui restitue en sortie (Dout) la trame émise Adresse, Data au format TT 5V.

Un décodeur bâti autour d'un microcontrôleur pic 16F1824 permet de programmer par une batterie de 5 micro-switchs, l'adresse du récepteur, et de décoder le message d'adresse et des données transmises.

Si adresse et données sont décodées valides, alors le décodeur le signale en basculant IT = 0, et en mémorisant le code de la data reçue.

Une ligne de transmission au format I2C en mode slave permet de lire la data reçue.

Afin d'effectuer les tests de bon fonctionnement du récepteur trois lignes de données parallèles (D3 \_ D0) TTL 5V sont activées par défaut.

Une Led signale en permanence l'activité du récepteur à chaque basculement du signal IT.

### **5-1 Caractéristiques interface I2C slave**

Mode de fonctionnement : slave

Fréquence : 100KHz

Adresse Physique : 0xA2

### **5-2 Description Interface I2C mode Esclave**

Le bloc réception logique de la télécommande du ROBOT est bâti autour d'un microcontrôleur PIC 16F1824 accessible via une liaison série synchrone au standard I2C , qui fonctionne en mode esclave pour une fréquence d'horloge  $F_{scl} = 100\text{KHz}$  max.

Le récepteur télécommande possède deux modes de fonctionnement accessibles par deux octets de commande de mode suivant :

**Mode réception** octet de mode = 0x31,0x00 (mode par défaut à la mise sous tension)

**Mode test** 0X32 octet de mode = 0x32,0x00

Le récepteur possède une ligne d'IT niveau TTL 5v active au niveau bas qui signale la présence de données valides au niveau du récepteur.

## Module Conception de Produit

**En mode Réception** la donnée valide est celle de la touche encodée par le boîtier télécommande conformément au tableau des **Codes d'informations transmises** codées ASCII suivants :

Touche DROITE	0x31
Touche AVANT	0x32
Touche MARCHÉ/ARRÊT	0x33
Touche ARRIÈRE	0x34
Touche GAUCHE	0x35

**En mode Test** la donnée valide est constante et codée 0x55 cette donnée est répétée cycliquement à la période de 65ms accompagnée de l'IT

## 6 Description Interface mode Maître (microcontrôleur ROBOT)

### **En mode Réception**

Dès que le signal d'IT est valide le maître peut engendrer la lecture de la donnée présente.

La commande Maître revêt la forme syntaxique I2C suivante :

Transmis par le Maître    S 0XA2 0X31 0X00 Sr 0XA3 A A N P

Reçu par le maître            A A A A 0X31 0X3y 00    (y = 1 à 5 selon la touche valide)

### **En mode Test**

Dès que le signal d'IT est valide (cyclique à 65ms) le maître peut engendrer la lecture de la donnée présente.

La commande Maître revêt la forme syntaxique I2C suivante :

Transmis par le Maître    S 0XA2 0X32 0X00 Sr 0XA3 A A N P

Reçu par le maître            A A A A 0X32 0X55 00

### **Nota**

Une driver développé en langage C est disponible et permet simplement ce dialogue.