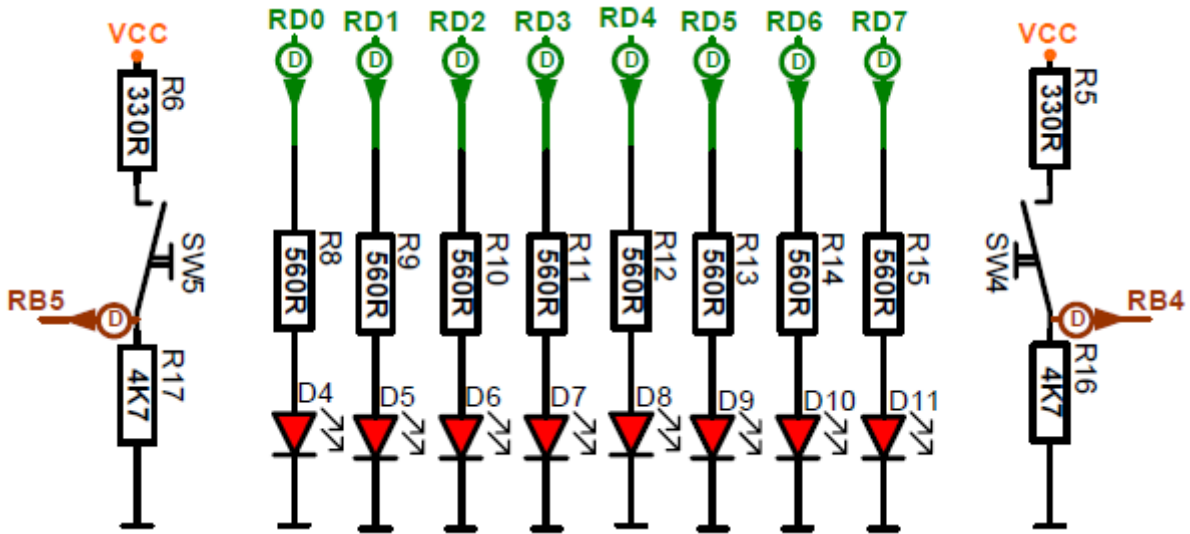


I. SCHEMA STRUCTUREL PARTIEL :



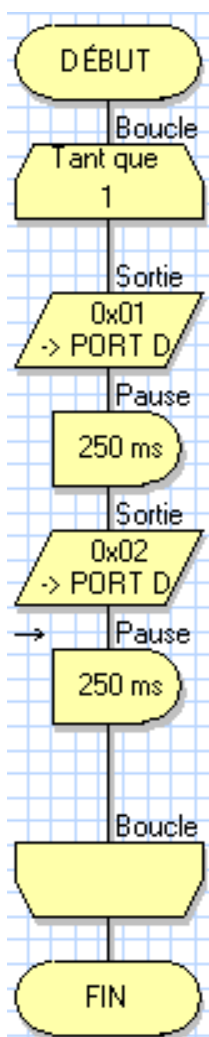
1. Faire une recherche : Qu'est-ce qu'un port d'entrée/sortie pour un microcontrôleur ?
2. En sachant que le port A s'écrit RA, que le port B s'écrit RB et que le bit 0 du port A s'écrit RA0..., sur quel port du microcontrôleur les DEL (LED en anglais) sont-elles connectées ?
3. Du point de vue du microcontrôleur, les DEL sont-elles des entrées ou des sorties ?
4. Sur quel port et sur quels bits du microcontrôleur les BP (Boutons poussoirs) ou SW (Switches en anglais) sont-ils connectés ?
5. Du point de vue du microcontrôleur, les BP sont-ils des entrées ou des sorties ?
6. Lorsqu'un BP est relâché, à quel potentiel la broche du microcontrôleur est-elle reliée (0V ou VCC) ?
7. Lorsqu'un BP est appuyé, à quel potentiel la broche du microcontrôleur est-elle reliée (0V ou VCC) ?
8. Compléter le tableau ci-dessous :


Un 0 sur BP signifie relâché ; un 1 sur BP signifie appuyé

| BP5 (SW5) | BP4 (SW4) | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | Contenu du Port B en hexadécimal |
|-----------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|----------------------------------|
| 0 | 0 | 0 | 0 | | | 0 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 0 | | | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | | | 0 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 0 | | | 0 | 0 | 0 | 0 | |

II. PROGRAMMATION :

Expérimentation 1a : Allumer une DEL puis une autre




- ◆ Choisir Fichier - Nouveau
- ◆ Puis **FORMULA FLOWCODE** comme puce à programmer.
- ◆ Cliquer sur **Misc** puis **FormulaFlowcode** pour ajouter le tableau de bord du buggy.
- ◆ Sélectionner le tableau de bord puis par un clic droit, accéder aux **Propriétés Etendues**
- ◆ Décocher « attendre appui sur BP pour exécuter le programme »
- ◆ Saisir le programme ci-contre
- ◆ Puis lancer la simulation complète  et observer ce qui se passe.

Expérimentation 1b :

Chenillard avec plusieurs DEL

Trouver toutes les valeurs hexadécimales permettant d'allumer successivement une seule DEL en partant de la LED 0 jusqu'à la LED 7 au rythme de 250ms.

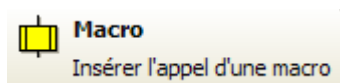
- ◆ Après avoir simulé votre programme, connecter le buggy avec le cordon USB ainsi que l'adaptateur secteur (réglé sur 6V).
- ◆ Transférer le programme en cliquant sur  (compiler vers la puce).
- ◆ Vérifier le fonctionnement et appeler le professeur.
- ◆ Comment se nomment les LED 0 à 7 sur le schéma structurel partiel ?
- ◆ Vous testerez tous les prochains programmes de la même manière.

Expérimentation 1c :

Chenillard avec plusieurs DEL

Modifier le programme pour allumer successivement une seule DEL en partant de la LED 0 jusqu'à la LED 7 au rythme de 250ms puis de la LED 7 jusqu'à la LED 0.

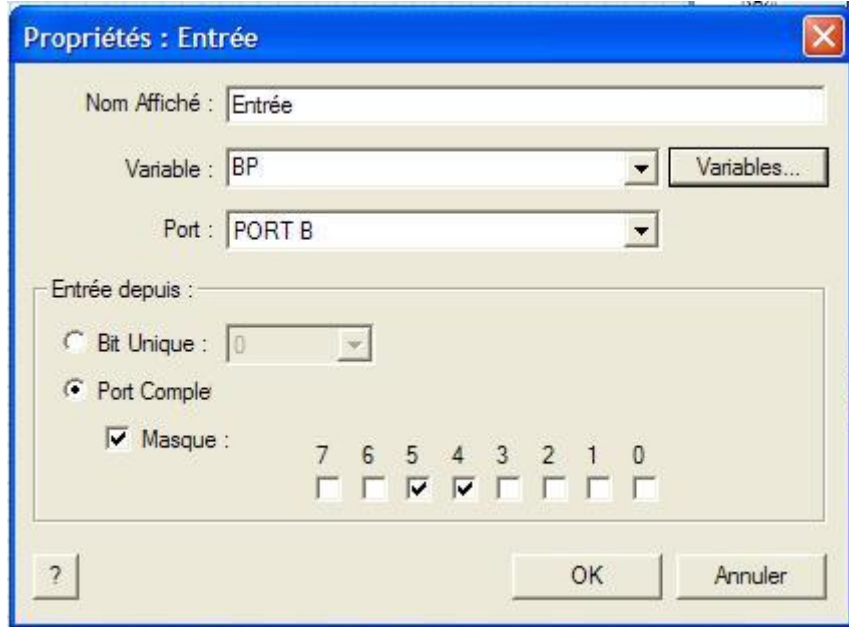
Pour simplifier le programme principal, vous pouvez utiliser la fonction « macro » (équivalente à un « sous-programme ») pour avoir une macro « sens1 » et une autre « sens2 »



- ◆ Cliquer sur **Créer Nouvelle Macro**
- ◆ Saisir sens1 puis valider OK
- ◆ Sélectionner la macro sens1 puis cliquer sur **OK et Editer Macro**
- ◆ Insérer la partie du programme qui représente la partie sens1

Expérimentation 2a : Décodage simple

Pour ne prendre en compte que les Boutons Poussoirs, il faut faire l'acquisition du contenu du port B avec la fonction « masque » sur les bits considérés soit les bits 4 et 5.



Lancer la simulation complète  et observer ce qui se passe lors d'appui sur les BP.

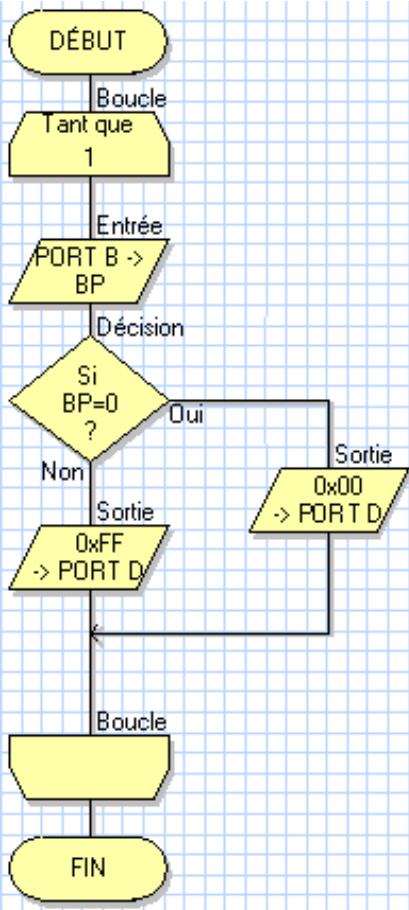
Expérimentation 2b :

Modifier le programme pour allumer les 4 DEL gauches lors d'un appui sur le BP de gauche et pour allumer les 4 DEL droites lors d'un appui sur le BP de droite.

Vous pouvez lire un bit unique si vous le souhaitez au lieu du port complet.

Expérimentation 2c :

Compléter le programme précédent pour allumer toutes les DEL lors d'un appui simultané sur les 2 BP.



Expérimentation 2d : Décodage multiple :

Les DEL sont allumées pour les 2 codes suivants :

Dans le cas contraire les DEL sont éteintes.

La condition sera testée avec une fonction "ou" (OR en anglais) dont la syntaxe suivante :

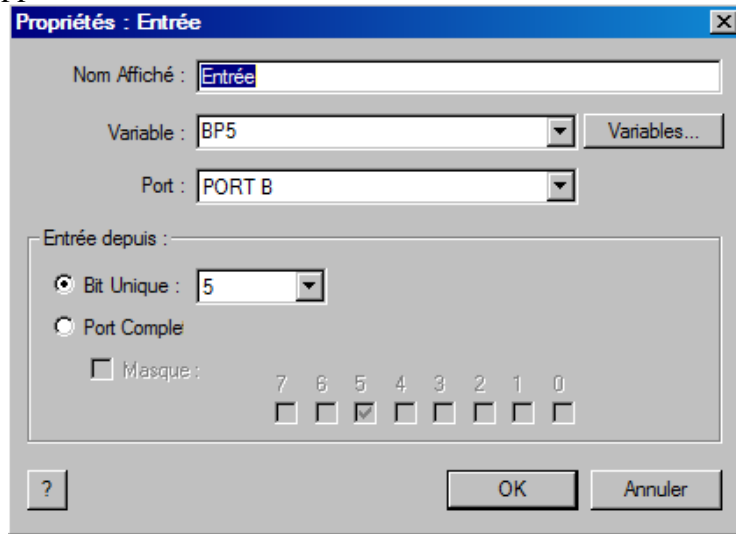
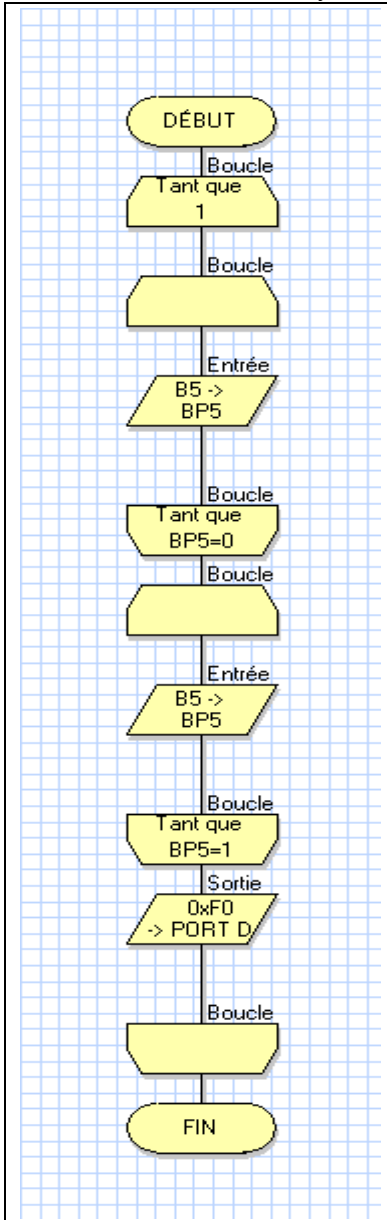
(BP = 0x...) OR (BP = 0x...)

| BP5 | BP4 |
|-------|--------|
| Fermé | Ouvert |
| Fermé | Fermé |

est la

Expérimentation 3a : Allumage des DEL si appui sur BP

Saisir ce programme et observer ce qui se passe sur le robot après un appui sur le BP5.



Quel est le problème ?

Vous pouvez aussi utiliser le mode pas-à-pas de la simulation (F8) pour vous aider.

Expérimentation 3b :

Modifier le programme pour éteindre les DEL lors d'un appui sur le BP4.

Expérimentation 3c :

Inversion de l'état des DEL par appui sur un seul BP :

Initialement les 8 DEL sont éteintes. Un appui sur un bouton poussoir inverse l'état des DEL.

On introduira une variable N de type « octet » qui contiendra la valeur à écrire sur le port qui commande les DEL.



Pour inverser le contenu binaire de la variable N, on utilisera un



opérateur d'inversion : $N = \text{NOT } N$ dans un bloc de calcul

Expérimentation 4 : Recopie

Recopier **en permanence** l'état des interrupteurs BP4 et BP5 sur les LED 4 et 5. Exemple : si BP4 fermé alors LED4 allumée.

| <i> Icône </i> | <i> Configuration à utiliser </i> |
|--|--|
| Entrée  | La valeur lue sera placée dans une variable Etat_BP de type « Octet ». |
| Sortie  | Le contenu de la variable Etat_BP sera écrit sur le port. |
| Boucle | On utilisera une boucle infinie (répétition infinie des séquences à l'intérieur de la boucle). On écrira « tant que 1 » : condition logique toujours vraie. |


RESSOURCES :


Conversion hexadécimal / binaire / décimal :


| Hex | Bin | Dec |
|-----|------|-----|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| A | 1010 | 10 |
| B | 1011 | 11 |
| C | 1100 | 12 |
| D | 1101 | 13 |
| E | 1110 | 14 |
| F | 1111 | 15 |


Exemple pour l'allumage de DEL (de 0 à 7) :

$2^7 = 128$ $2^6 = 64$ $2^5 = 32$ $2^4 = 16$ $2^3 = 8$ $2^2 = 4$ $2^1 = 2$ $2^0 = 1$

 = 0b00000111 = 0x07 = 4+2+1 = 7

 = 0b10000010 = 0x82 = 128+2 = 130

 = 0b11000011 = 0xC3 = 128+64+2+1 = 195

 = 0b01110110 = 0x76 = 64+32+16+4+2 = 118

7 6 5 4 3 2 1 0

}
}
}

binaire hexadécimal décimal