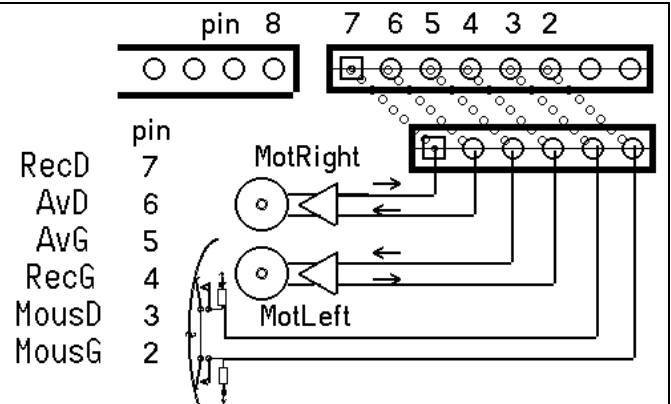




Définitions, fonctions et exemples pour le xBot

1ere partie Moteurs et moustaches

Rappelons le câblage standard et les noms des signaux .
 Rappelons que les pins 2 à 7 sont sur les bits 2 à 7 port D du processeur Avr368.
 Les bits 2 et 3 sont en entrée pour lire les moustaches. Les bits 4 à 7 commandent les moteurs, avec les bits 5 et 6 qui permettent le PWM Arduino, ils sont donc utilisés comme signaux pour faire avancer.



Fichiers de définition Arduino et C

```
//XbotDef.h Définitions moustach et moteur
#include <Arduino.h>
#define RecG 4
#define AvG 5
#define AvD 6
#define RecD 7
#define MousD 3
#define MousG 2
#define ObsG !digitalRead(MousG)
#define ObsD !digitalRead(MousD)
#define LedOn digitalWrite (13,HIGH)
#define LedOff digitalWrite (13,LOW)

void MotMousSetup () {
  pinMode(RecG,OUTPUT);
  pinMode(AvG,OUTPUT);
  pinMode(AvD,OUTPUT);
  pinMode(RecD,OUTPUT);
  // moteurs arretes
  digitalWrite(RecG, LOW);
  digitalWrite(RecD, LOW);
  analogWrite(AvG,0);
  analogWrite(AvD,0);

  pinMode(MousD,INPUT);
  pinMode(MousG,INPUT);
}

void Avance () {
  digitalWrite(AvG, HIGH);
  digitalWrite(RecG, LOW);
  digitalWrite(AvD, HIGH);
  digitalWrite(RecD, LOW);
}
etc
```

```
// XbotCDef.h definitions et fonctions
#define bRecG 4 // PORTD actifs à 1
#define bAvG 5
#define bAvD 6
#define bRecD 7
#define maskMot
#define bMousD 3 // PORTD actifs à 0
#define bLedD 3
#define bMousG 2
#define bLedG 2

#define ObsG !(PIND & 1<<bMousG)
#define ObsD !(PIND & 1<<bMousD)

void MotMousSetup () {
  DDRD |= (1<<bMousD | 1<<bMousG)
  DDRD &= ~maskMot
}

void Avance () {
  PORTD &= ~maskMot ; // clear all
  PORTD |= 1<<bAvD | 1<< bAvG ;
}
idem Recule TourneD TourneG

par interruption Speed (vitD,vitG)
```

Programme 1 - Evite les obstacles

```
// EvitObstacle.ino
#include "XbotDef.h"
void setup() {
  MotMousSetup ();
}
void loop() {
  Avance ();
  if (ObsG) { { //on recule et tourne
    Recule ; delay (300) ;
    TourneG ; delay (200) ;
  }
```

```
// EvitObstacleC.ino
#include "XbotDef.h"
int main() {
  MotMousSetup ();
  while (1) {
    Avance ();
    if (ObsG) { { //on recule et tourne
      Recule ; delay (300) ;
      TourneG ; delay (200) ;
    }
```

<pre> } if (ObsG) { //on recule et tourne Recule ; delay (300) ; TourneD ; delay (100) ; } } </pre>	<pre> if (ObsG) { //on recule et tourne Recule ; delay (300) ; TourneD ; delay (100) ; } } </pre>
---	---

A noter que les fichiers inclus peuvent très bien être écrits avec les fonctions Arduino. Ce qui compte, ce n'est que le programme principal ne dépende pas du matériel et du type de compilateur. Notre programme ne sait rien du robot si ce n'est qu'il sait avancer et tourner. Mais nous avons fait une grave erreur en disant dans le programme qu'il recule de 0.3 secondes. C'est pas valable pour un autre robot! Il aurait fallu écrire DelRecul et déclarer au début du programmes, avec les déclarations matérielles

```
#define DelRecul 300 OU const int DelRecul = 300 ;
```

C'est en écrivant des programmes plus riches, en changeant de robot, de compilateur, que l'on comprend bien cette nécessité d'une bonne structuration des programmes. On essaie tout de suite de bien faire?

Programme 2- Ne pas se coincer contre l'obstacle

On remarque que face à un obstacle, le robot alterne à l'infini. Un amélioration est de changer la durée de la rotation. S'il tourne plus longtemps à droite qu'à gauche le robot va se décaler et a plus de chance de quitter l'obstacle.

Plus efficace, c'est de compter les obstacles et tous le 5 par exemple faire une pirouette.

<p>Première chose, toujours, toujours quand on continue un programme qui marche, donner un nouveau nom au fichier, aussi explicite que possible. Décrire le comportement voulu, l'algorithme, la date, l'auteur.</p>	
--	--

Bon, maintenant, facile de déclarer un compteur et l'incrémenter dans chaque "if". Mais ou tester? Dans la boucle loop avec un if (programme complet sous EvitObstacleCompte.ino) On pourrait aussi dire "tant que le compteur est inférieur à x" on fait rien de spécial.

<pre> byte compteObst = 0 ; #define nbObst 5 #define DelTouComptObst 800 void loop() { Avance ; if (ObsD) { //on recule et tourne Recule ; delay (300) ; TourneG ; delay (200) ; compteObst++; } if (ObsG) { //on recule et tourne Recule ; delay (300) ; TourneD ; delay (100) ; compteObst++; } if (compteObst > nbObst) { compteObst = 0 ; TourneD ; delay (DelTouComptObst) ; } // end if } // end loop </pre>	<pre> // insérer avant le programme les fonctions EviterObsD();EviterObsG(); byte compteObst = 0 ; #define nbObst 5 #define DelTouComptObst 800 void loop() { compteObst = 0 ; while (compteObst < nbObst) { Avance ; if (ObsD) { EviterObsG () ; compteObst++; } if (ObsG) { EviterObsD () ; compteObst++; } } // end while TourneD ; delay (DelTouComptObst) ; } // end loop </pre>
---	--

On remarque dans la 2e colonne que l'on a défini des fonctions pour faciliter la lecture du programme principal. Dans le programme, on veut éviter les obstacle. Comment, c'est un

autre niveau de réflexion qui se définit dans des fonctions. Une fois les fonctions bien définies et adaptées, on les mets dans un #include.

Une fonction est une action, son nom commence par un verbe.

<pre>void EviterObsD () { //on recule et tourne Recule ; delay (300) ; TourneG ; delay (200) ; }</pre>	<pre>void EviterObsG () { //on recule et tourne Recule ; delay (300) ; TourneD ; delay (200) ; }</pre>
--	--

Programme 6 - Agir au hasard

Ce serait mieux de corriger le comportement au hasard. Arduino a la fonction random (min,max) qui génère des nombres entre min et max-1.

Dans le programme précédent, il suffit de changer dans le if nbObsst par random (min,max) .

Renommez le fichier et essayez (seulement si vous n'y arrivez pas: EvitObstacleRandom.ino)

Autres idées

Dessiner un polygone

Programme 8 – Suivre un mur

Contourner l'extrémité d'une paroi

Modifier la vitesse

Corriger pour une ligne droite

Tracer une spirale

Ajoutez des affichage et capteurs

Il y a quantité d'informations sur le web. Les documents Didel sont nombreux, pour les trouver taper par exemple "Didel distance" et vous trouvez notre doc sur les capteurs de distance en meilleure position.

Une liste classée de nos documents se trouve sous <http://www.didel.com/Liens.pdf>