

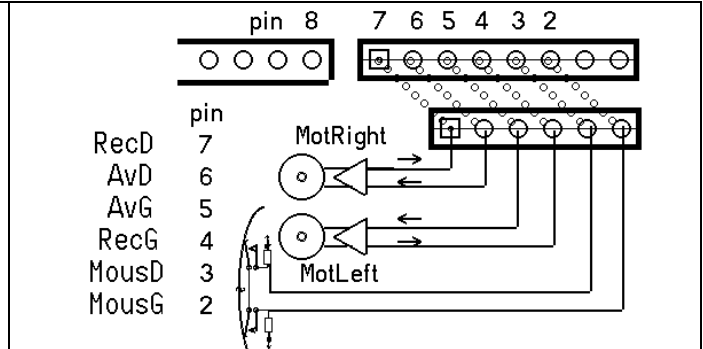


Définitions et fonctions pour le xBot

voir aussi xbot start.

1ere partie Moteurs et moustaches

Rappelons le câblage standard et les noms des signaux .
 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.



Attention. Les min/MAF sont une source d'erreur fréquente XBot (ancienne notations Xbot ou xBot)
 Attention: On utilise en tout ou rien des macros - Avance; - et pour le PWM PFM des fonctions – Vitesse (pfnG,pfmD);

Les fichiers des programmes cités se trouvent sous www.didel.com/xbot/XBotLib.zip

Fichier de définition

```
//XBotDef.h Définitions moustaches et moteur (parfois XBotCDef.pdf)
#include <Arduino.h>
#define bLed 5 // PortB pin 13 debug
#define LedOn bitSet (PORTB,bLed)
#define LedOff bitClear (PORTB,bLed)
#define LedToggle PORTB ^= 1<<bLed

// Câblage sur XBot
#define RecG 4
#define AvG 5
#define AvD 6
#define RecD 7
#define MousD 3
#define MousG 2

// Test des obstacles avec les moustaches
#define ObsG !digitalRead(MousG)
#define ObsD !digitalRead(MousD)

// macros pour les exercice simples appel Avance;
#define Avance PORTD |= 0b01100000; PORTD &= 0b01101111
#define Recule PORTD |= 0b10010000; PORTD &= 0b10011111
#define TourneD PORTD |= 0b10100000; PORTD &= 0b10101111
#define TourneG PORTD |= 0b01010000; PORTD &= 0b01011111
#define Stop PORTD &= ~0b11110000

// macros pour le Pfm
#define AvD bitClear (PORTD,bRecD); bitSet (PORTD,bAvD)
#define RecD bitSet (PORTD,bRecD); bitClear (PORTD,bAvD)
#define StopD bitClear (PORTD,bRecD); bitClear (PORTD,bAvD)
#define AvG bitClear (PORTD,bRecG); bitSet (PORTD,bAvG)
#define RecG bitSet (PORTD,bRecG); bitClear (PORTD,bAvG)
#define StopG bitClear (PORTD,bRecG); bitClear (PORTD,bAvG)

// A appeler dans le setup
void XBotSetup () {
  DDRD |= 0b11110000 ; // force les 1 out
  DDRD &= 0b11110011 ; // force les 0 in
  PORTD = 0x0F ; // --> 0 Mot stop
  DDRB |= 1<<bLed ; // Led13
}
```