



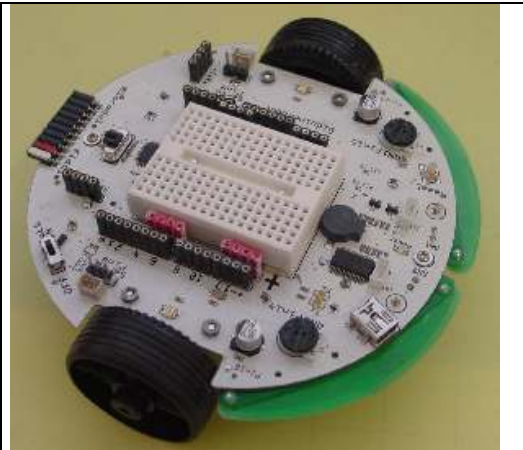
Eviter les obstacles – approche C

2e page: Eviter les obstacles – Arduino

Le DdR a deux moustaches en entrée et 2 moteurs bidirectionnels en sortie.

L'approche C raisonne avec les ports et bits sur lesquels les périphériques sont câblés. On définit des masques utilisés avec des ET et OU logiques pour modifier les bits concernés.

PORTC le port sur lequel sont câblés les moustaches
bMousD le numéro du bit pour la moustache droite
mMousD le masque 8 bits pour filtrer le bit moustache
1<<bMousD est une écriture préférée pour écrire un masque dans lequel un seul bit est actif.



```
//Moustaches3.ino
```

```
// On essaie deux corrections en cas d'obstacle à droite: tourner; à gauche: reculer puis tourner
```

```
#define bMousD 2 // PORTC bit 2 (pin 16)
#define bMousG 3
#define mMousDroite 1<<bMousD // masque
#define mMousGauche 1<<bMousG
#define MousDTouche !(PINC & mMousDroite) // ou !(PINC&(1<<bMousD))
#define MousGTouche !(PINC & mMousGauche)
```

```
#define Avance PORTD &= 0b00001111 ; PORTD |= 0b01100000
#define Recule PORTD &= 0b00001111 ; PORTD |= 0b10010000
#define TourneD PORTD &= 0b00001111 ; PORTD |= 0b10000000
#define TourneG PORTD &= 0b00001111 ; PORTD |= 0b00010000
#define Stop PORTD &= 0b00001111 ;
```

```
void setup ()
```

```
{
  DDRD = 0b11111111;
  PORTD = 0b00000000;
  DDRC = 0b00000000;
  PORTC = 0b11111111; // pull-ups
}
```

```
// ces définitions seront cachées dans in #include après les premiers tests, pour se concentrer
// sur le programme principal (revenir si on n'a pas rencontré d'obstacle, etc.)
```

```
void loop()
```

```
{
  Avance ;
  if (MousDTouche)
  {
    TourneD ; delay (300) ;
  }
  if (MousGTouche)
  {
    Recule ; delay (300);
    TourneG ; delay (300) ;
  }
}
```

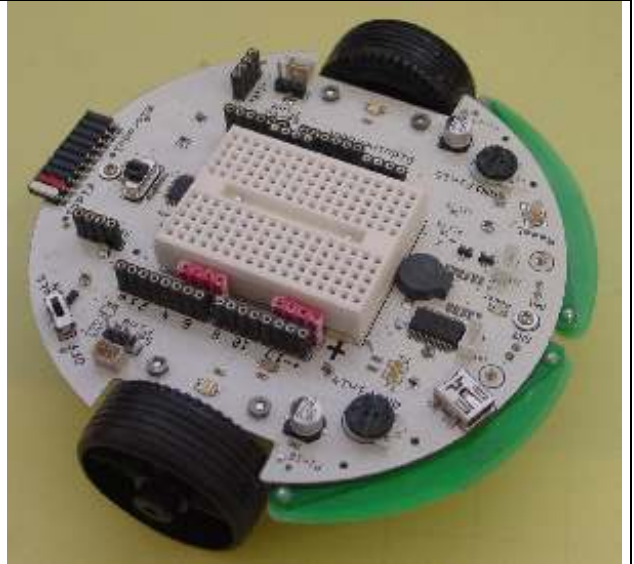
En écrivant `TourneD ; delay (300) ;` sur la même ligne, on met en évidence la fonctionnalité testée: reculer, mais pendant un certain temps

Eviter les obstacles – Arduino

Le DdR a deux moustaches en entrée et 2 moteurs bidirectionnels en sortie.

Arduino simplifie la vision du processeur en numérotant les entrées-sorties (pins).

La première étape est de dire sur quelles pins sont câblés les moteurs et les moustaches, et quelles sont les polarités de ces signaux (0 = LOW ou 1 = HIGH) lorsque le pin est active.



La 2e colonne montre comment préparer le #include, qui doit se trouver dans le même sketch.

```
//Moustaches4.ino
#define MousD 16
#define MousG 17
// notations suivantes définie
//dans http://mediawiki.e-apprendre.net
#define M1 7
#define E1 6
#define E2 5
#define M2 4
#define MousD 16
#define MousG 17

void Avance () {
    digitalWrite (E1,1) ;
    digitalWrite (M1,0) ;
    digitalWrite (E2,1) ;
    digitalWrite (M2,0) ;
}
// etc pour les autres déplacements

void setup ()
{
    pinMode (MousD, INPUT) ;
    pinMode (MousD, INPUT) ;
    pinMode (M1, OUTPUT) ;
    pinMode (E1, OUTPUT) ;
    pinMode (E2, OUTPUT) ;
    pinMode (M2, OUTPUT) ;
}

void loop()
{
    Avance () ;
    if (digitalRead (MousD) == 0)
    {
        TourneD () ; delay (300) ;
    }
    if (digitalRead (MousG) ==0)
    {
        Recule () ; delay (300) ;
        TourneG () ; delay (300) ;
    }
}
```

```
//Moustaches.h
#include Arduino.h
#define MousD 16
#define MousG 17

#define M1 7
#define E1 6
#define E2 5
#define M2 4
#define MousD 16
#define MousG 17

void Avance () {
    digitalWrite (E1,1) ;
    digitalWrite (M1,0) ;
    digitalWrite (E2,1) ;
    digitalWrite (M2,0) ;
}
// etc pour les autres déplacements
void setup ()
{
    pinMode (MousD, INPUT) ;
    pinMode (MousD, INPUT) ;
    pinMode (M1, OUTPUT) ;
    pinMode (E1, OUTPUT) ;
    pinMode (E2, OUTPUT) ;
    pinMode (M2, OUTPUT) ;
}
-----
//Moustaches1.ino
#include "Moustaches.h"
void loop()
{
    Avance () ;
    if (digitalRead (MousD) == 0)
    {
        TourneD () ; delay (300) ;
    }
    if (digitalRead (MousG) ==0)
    {
        Recule () ; delay (300) ;
        TourneG () ; delay (300) ;
    }
}
```