

Extrait du PoBot

<http://www.pobot.org>

xPo en Lego (2005)

- L'association - Nos robots - Réalisations avec Lego Mindstorms -



Date de mise en ligne : lundi 24 octobre 2005

PoBot

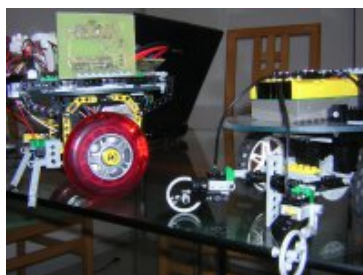
Le principe ici est d'établir un parallèle entre un PoBot sur la base d'un RCX, et un PoBot sur la base d'un MR-163 (une carte électronique équipée d'un microcontrôleur Atmel AVR ATmega163). Les deux robots sont construits avec un châssis Lego.

Voici les deux robots côte à côte :



PoBotRcx vs PoBotµC

Vue particulière du système anti-chute utilisant des capteurs qui détectent le bord de la table :



Capteurs "bord de table"

Vidéo des deux robots en action :

[Robots hybrides Lego / Pobot](http://www.dailymotion.com/video/x7hyv6_robots-hybrides-lego-pobot_tech) [http://www.dailymotion.com/video/x7hyv6_robots-hybrides-lego-pobot_tech]
envoyé par [JulienPobot](http://www.dailymotion.com/JulienPobot) [http://www.dailymotion.com/JulienPobot]

Le code de la brique RCX (langage NQC) :

```
// déclaration des moteurs #define moteur_left OUT_A #define moteur_right OUT_C // déclaration des capteurs
#define contact1 SENSOR_1 #define contact3 SENSOR_3 task main() { SetSensor (contact1,
SENSOR_TOUCH); SetSensor (contact3, SENSOR_TOUCH); start action; } task action() { while (true)
{ OnRev (moteur_left); OnRev (moteur_right); if (contact3 != 1)
{ OnFwd (moteur_right); OnFwd (moteur_left); Wait(100);
Off(moteur_right); OnRev (moteur_left); Wait(100);
} if (contact1 != 1) { OnFwd (moteur_right); OnFwd
(moteur_left); Wait(100); Off(moteur_left); OnRev
(moteur_right); Wait(100); } } }
```

Le code de la carte électronique Atmel (langage C) :

```
#include <avr/io.h> #include <stdio.h> #include <iom163.h> void port_init(void) {   DDRD = 0x00 ; // entrée des
switch   DDRC = 0xFF ; // sortie (moteur)   PORTC = 0x00 ;   DDRA = 0xFF ; // touche enable   PORTA = 0x00
; } void DelayC(void) {   unsigned char a ;   int i ;   for (i=0;i<5;i++) // délai court   for (a = 1; a++;) } char
testbit(char reg, char n) {   char m=1 < < n; // teste la valeur haute ou basse du port entrée   return (reg & m) ; } int
main(void) {   int j ; // compteur de pas moteur   port_init() ;   PORTA = 0xFF ; // autorisation enable   i = 0 ;   while
(1) { // les deux roues avancent   PORTC=0b00010001;   DelayC();   PORTC=0b01000100;   DelayC();
   PORTC=0b00100010;   DelayC();   PORTC=0b10001000;   DelayC();   if (testbit(PIND, 7)) { // un des
capteurs est déclenché   for(j = 1;j <= 100;j++) { // les deux roues reculent   PORTC=0b10001000;
   DelayC();   PORTC=0b00100010;   DelayC();   PORTC=0b01000100;   DelayC();   PORTC=0b00010001;
   DelayC();   }   for(j = 1;j <= 100;j++) { // une seule roue tourne (celle du coté où le capteur est enclenché)
   PORTC=0b00010000;   DelayC();   PORTC=0b01000000;   DelayC();   PORTC=0b00100000;   DelayC();
   PORTC=0b10000000;   DelayC();   } }   if (testbit (PIND, 4)) { // le capteur opposé est enclenché   for(j =
1;j <= 100;j++) {   PORTC=0b10001000;   DelayC();   PORTC=0b00100010;   DelayC();
   PORTC=0b01000100;   DelayC();   PORTC=0b00010001;   DelayC();   }   for(j = 1;j <= 100;j++) {
   PORTC=0b00000001;   DelayC();   PORTC=0b00000010;   DelayC();   PORTC=0b00000010;   DelayC();
   PORTC=0b00001000;   DelayC();   } }   } } // ne pas tenir compte de la dernière ligne
```