#include <bacarMotor.h>

#include <bacarComm.h>

/\* DÃ©finit les objets liÃ©s aux moteurs de la voiture

\* Les paramÃ¨tres dÃ©finissent les pattes auxquelles sont

\* raccordÃ©es le H-bridge.

\* Le 1er paramÃ¨tres dÃ©finit la patte PWM qui doit Ãªtre

\* la patte D9 ou D10.

\* Les deux autres paramÃ¨tres dÃ©finissent les pattes IN1 et IN2,

\* ce peut Ãªtre n'importe quelle patte digitale de l'Arduino \*/

BacarMotor motorA(9, 7, 8);

BacarMotor motorB(10, 11, 12);

// Cree l'objet bacarComm utilisÃ©s pour permettre la com entre arduino et orange pi

BacarComm comm;

//Cree un entier qui servira a enregistrÃ© une manoeuvre sur plusieurs tout de loop et non juste une

int x\_save ;

void setup() {

// Configure la LED pour pouvoir l'utiliser dans loop()

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(A3,INPUT); // capteurs de bords 15 et 16

pinMode(A1,INPUT);

pinMode(A2,INPUT); // capteur de proxi attention pin D2

// Initialise les objets moteurs et com

motorA.begin();

motorB.begin();

comm.begin();

}

void loop() {

int32\_t x, y;

float u, v;

// VÃ©rifie si un nouveau message de l'Orange PI a Ã©tÃ© reÃ§u

if (comm.newMessage() == true) {

digitalWrite(LED\_BUILTIN, HIGH);

// On lit les 4 valeurs contenues dans le message

x = comm.xRead(); // sera les manips a effectues envoyÃ©s par la state machine

y = comm.yRead(); // sera un entirer representant un etat global des capteurs de la voiture et envoyÃ© a la state machine

u = comm.uRead(); // sera la tension du moteur A

v = comm.vRead(); // sera la tension du moteur B

if(x != 0)

{

switch(x) //on effectue les manips

{

case 1 :

x\_save = 0 ;

manoeuvre\_M1() ;

break ;

case 2 :

x\_save = 0 ;

manoeuvre\_M2() ;

break ;

case 3 :

x\_save = 3 ;

break ;

case 4 :

x\_save = 4 ;

break ;

}

}

// si x\_save est pas a 0 on l'update

else if( (x == 0) and (x\_save != 0) )

{

x\_save = 0 ;

}

// si il n'y a pas de manip a effectuÃ©

else

{

if(u != motorA.getPwmValue()){ motorA.actuate(u); }

if(v != motorB.getPwmValue()){ motorB.actuate(v); }

}

}

// On verifie si nous sommes en train d executer une manip necessitant uniquement l'arduino

switch(x\_save)

{

case 3 :

manoeuvre\_M3() ;

break ;

case 4 :

manoeuvre\_M4() ;

break ;

}

}

//---------------------------------------------------------------------------------------------

void manoeuvre\_M1() // Faire demi-tour

{

motorA.actuate(0.7);

motorB.halt();

delay(3000);

}

//---------------------------------------------------------------------------------------------

void manoeuvre\_M2() // carre

{

int i;

for(i=0; i < 3; ++i)

{

droit\_30cm();

angle\_droit() ;

}

droit\_30cm();

car\_stop() ;

}

void angle\_droit(){

motorA.actuate(0.8);

motorB.halt();

delay(1350);

}

void droit\_30cm(){

car\_drive\_straight() ;

delay(900);

}

//---------------------------------------------------------------------------------------------

void manoeuvre\_M3() // capteurs de bord

{

// roule

car\_drive\_straight() ;

if( digitalRead(A3) == LOW ) // digitRead(pin) renvoie True si la valeur renvoye par la pin est High mais nous nous detectons un objet quand la valeur est low

{

//Faire attention si la valeur detec n est que du bruit donc double verif avec un delay

delay(15) ; //choisis en fct des tests

if( digitalRead(A3) == LOW)

{

car\_stop() ;

delay(10);

motorA.actuate(0.0) ;

motorB.actuate(-0.8) ;

while(digitalRead(A3)== LOW) // Tant que nous detecton un obstacle nous restons a -0.3 pour moteur B

{

delay(200) ;

}

car\_stop() ;

delay(100);

}

}

else if( digitalRead(A1) == LOW ) // digitRead(pin) renvoie True si la valeur renvoye par la pin est High mais nous nous detectons un objet quand la valeur est low

{

//Faire attention si la valeur detec n est que du bruit donc double verif avec un delay

delay(15) ; // choisis avec les tests

if( digitalRead(A1) == LOW)

{

car\_stop() ;

delay(10);

motorA.actuate(-0.8) ;

motorB.actuate(0.0) ;

while(digitalRead(A1)== LOW) // Tant que nous detecton un obstacle nous restons a -0.3 pour moteur A

{

delay(200);

}

car\_stop() ;

delay(100);

}

}

}

//---------------------------------------------------------------------------------------------

void manoeuvre\_M4() // capteur de proximitÃ©

{

// roule

car\_drive\_straight() ;

if( not digitalRead(A2) ) // digitRead(pin) renvoie True si la valeur renvoye par la pin est High mais nous nous detectons un objet quand la valeur est low

{

//Faire attention si la valeur detec n est que du bruit donc double verif avec un delay

delay(5) ; // choisis en fct des tests

if( not digitalRead(A2))

{

car\_stop() ;

delay(1000); // delay pour ne rien executer d'autre apres la detection

x\_save = 0 ; // remet la cst de manoeuvre a 0

}

}

}

//---------------------------------------------------------------------------------------------

// fct pour tous le programme

void car\_drive\_straight()

{

motorA.actuate(0.5);

motorB.actuate(0.5);

}

void car\_stop()

{

motorA.halt() ;

motorB.halt() ;

}