#include "mbed.h"

Serial PC(USBTX, USBRX); // Détecteur

Serial M1(A0, A1); // Moteur 1: axe x (axe optique)

Serial M2(A4, A5); // Moteur 2: axe y (perpendiculaire à l'axe optique.

Serial det(A3, A2);// (a finir) pour vers ordinateur

void position1(float p0, float pas, int n);

// fonction qui renvoie la position du moteur M1 au moteur et à Matlab

// p0 correspond à la position innitiale (commande recue par Matlab)

// pas correspond au pas indiqué par Matlab

// n correspond à l'anvancement , la n-ieme mesure

void position2(float p0, float pas, int n);

// fonction qui renvoie la position du moteur M2 au moteur et à Matlab

// p0 correspond à la position innitiale (commande recue par Matlab)

// pas correspond au pas indiqué par Matlab

// n correspond à l'anvancement , la n-ieme mesure

void lire(void);

// Fonction qui lit la position de M1 et M2 et la mesure du détecteur et qui l'envoie à Matlab

void automatique(void);

// Fonction qui envoie automatiquement les mesures à Matlab en fonction de la position des moteurs.

// Si la fonction recoit n1=n2=0 elle ne fait qu'ue mesure, cette fonction

// permet donc de faire une commande manuelle.

void talk(void);

// Fonction qui creer une chaine de caractère à partir de 3 données :

// la position de M, la position de M2 et l'intensité mesurée avec le détecteur.

void Matlab(void);

// Fontion d'interruption correspondant à l'arrivée d'une information de Matlab

// et qui modifie les m1 à m7 avec les 7 valeurs correspondant à notre convention

// et rendant lecture 1 si le message est correct et lance la boucle "active"

double pas1, pas2, p01, p02, p1, p2, n1, n2, mesure, tol;

// pas1 => pas du moteur 1,donnée provenant de matlab

// pas2 => pas du moteur 2

// p01 => Position innitiale de M1

// p02 => Position innitiale de M2

// n1 => nombre de décallage de pas pour M1

// n2 => nombre de décallage de pas pour M2

// p1 => Position de M1 en temps réel on renvoie à Matlab

// p2 => Position de M2 en temps réel on renvoie à Matlab

// mesure => Mesure envoyé à Matlab pour faire les graphes

// tol=> Tolerancement donné par l'utilisateur

double m1, m2, m3, m4, m5, m6, m7;

// correspond à notre parsemage de message

int lecture = 0; // variable qui valide une lecture

int main() {

 PC.attach(&Matlab,Serial::RxIrq);

 while(1){

 if ( lecture == 1){

 if ( m1 == 0){ // lire positions initiales

 lire();

 }

 if ( m1 == 1){ // mode manuel déplace le moteur (donnée par m2) à la position (donnée par m3)

 if (m2 == 1){ // moteur 1

 p01 = m3;

 position1(p01,0,1);

 tol=m4;

 }

 if (m2 == 2){ // moteur 2

 p02 = m3;

 position2(p02,0,1);

 tol=m4;

 }

 lire();

 }

 if ( m1 == 2){ // mode automatique

 pas1 = m2;

 p01 = m3;

 n1 = m4;

 pas2 = m5;

 p02 = m6;

 n2 = m7;

 automatique();

 }

 lecture = 0; // remet à zero la lecture

 }

 }

}

void position1(float p0, float pas, int n){

 double p = p0+n\*pas;

 M1.putc(p/100); // distance max 0-100

 // prendre en compte le tolerancement dans cette fonction

 }

void position2(float p0, float pas, int n){

 double p = p0+n\*pas;

 M2.putc(p/100); // distance max 0-100

 M1.putc(p/100); // distance max 0-100

 // prendre en compte le tolerancement dans cette fonction

 }

void lire(void){

 if (M1.readable()) //des que M1 est prête

 {

 p1 = scanf("%s",M1); // on mesure la donnée envoyée par le moteur 1

 p1 = p1\*100;

 }

 if (M2.readable()) //des que M2 est prête

 {

 p2 = scanf("%s",M2); // on mesure la donnée envoyée par le moteur 2

 p2 = p2\*100;

 }

 if (det.readable()) //des que det est prête

 {

 mesure = det.scanf("%s", mesure ); // on mesure la donnée envoyée par le moteur 1

 }

 talk();

 }

void automatique(void){

 // Il faut s'assurer que les valeurs p01,p02, n1, n2, pas1 et pas2 ont bien été récupérées.

 int i, j;

 for (i=0 ; i<n1 ; i++)

 {

 position1(p01,pas1,i);

 for (j=0 ; j<n2 ; j++)

 {

 position2(p02,pas2,j);

 lire();

 }

 }

}

void talk(void){ // écrit un message à matlab avec la convention S1,2,3E

 // ramener si nécessaire p1,p2,m dans la gamme entier positif (si impossible voir typage de chaine

 char chaine[8];

 chaine[1] = 'S';

 chaine[2] = (char)p1;

 chaine[3] = ',';

 chaine[4] = (char)p2;

 chaine[5] = ',';

 chaine[6] = (char)mesure;

 chaine[7] = 'E';

 PC.printf(chaine);

 }

void Matlab(void) { // fonction d'intérruption permettant de lire avec notre

// convention S1,2,3,4,5,6,7E

 if( PC.getc() == 'S' ){

 m1 = PC.getc(); // m correspondant au mode d'utilisation

 PC.getc();

 m2 = PC.getc();

 PC.getc();

 m3 = PC.getc();

 PC.getc();

 m4 = PC.getc();

 PC.getc();

 m5 = PC.getc();

 PC.getc();

 m6 = PC.getc();

 PC.getc();

 m7 = PC.getc();

 }

 if (PC.getc() != 'E') { // Envoyer un signal erreur mais pour plus tard

 lecture = 1 ;

 }

}