👨‍💻

Répertoire de codes

Groupe 3 : Thibault Benahmed, Louis Jacomino, Samuel Menay-Marigny, Victor Minot

Code Envoi

```
#include "mbed.h"
#include <cstdint>
#include <string.h>

UnbufferedSerial my_pc(USBTX,USBRX);
UnbufferedSerial my_rf(A0,A1);

// Pins Joystick
AnalogIn pot_X(A3);
AnalogIn pot_Y(A4);

//int a;
uint8_t a;
char message[128];

int main() {
    // Mesure X/Y
    my_pc.baud(9600);
    my_rf.baud(19200);
    sprintf(message, "Bienvenue a l'IOGS - Emet\r\n");
    my_pc.write(message, strlen(message));
    while(1){

        int mesX,mesY;
        double X,Y,Val_X,Val_Y,X0 = 1.65,Y0 = 1.65;
        double theta, R, spd = 0.0;

        mesX = pot_X.read_u16();
        mesY = pot_Y.read_u16();

        Val_X = (mesX / 65520.0) * 3.3;
        Val_Y = (mesY / 65520.0) * 3.3;

        X=Val_X-X0;
        Y=Val_Y-Y0;

        // Conversion en polaire

        R=(Y/abs(Y))*sqrt(X*X+Y*Y);
        theta = atan(abs(Y)/X)*(180/3.14) - (X/abs(X))*90;
        if (abs(R) < 0.1 || abs(theta) < 0.5) theta = 0 ;


        spd= R / 1.65;
        if (abs(spd)<0.05) spd = 0 ;
        // Securité moteur

        if (spd>1) spd = 0.0 ;
        // Gestion 2 moteurs

        double spd_D=spd,spd_G=spd;

        if ((theta)>80) theta = 80 ;
        if ((theta)<-80) theta = -80 ;
        if (theta>5) {
            if (theta>10 && theta<35) spd_D = 0.35 , spd_G = 0.65;
            if (theta>35 && theta<65)  spd_D = 0.25 , spd_G = 0.75;
            if (theta>65) spd_D = 0 , spd_G = 1;
        }
        if (theta<-5) {
            if (theta>-35 && theta<-10) spd_D = 0.65 , spd_G = 0.35;
            if (theta<-35 && theta>-65)  spd_D = 0.75 , spd_G = 0.25;
            if (theta<-65) spd_D = 1 , spd_G = 0;
        }
        //sprintf(message, "Valeur de a = %d\r\n", a);
        uint8_t VD,VG;
        VD=(uint8_t)(abs(spd_D)*15) & 0x0F ;
        VG=(uint8_t)(abs(spd_G)*15) & 0x0F ;
        a = VD + (VG<<4);
        my_rf.write(&a, 1);
        sprintf(message, "Send %f %f / %x %x = %x %f %f\r\n", abs(spd_D), abs(spd_G), VD, VG, a, theta, spd);
        my_pc.write(message, strlen(message));
        thread_sleep_for(300);
    }
}
```

Code Réception

```
#include "mbed.h"
#include <cstdint>
#include <string.h>

UnbufferedSerial my_pc(USBTX,USBRX);
UnbufferedSerial my_rf(A0,A1);

uint8_t data;
char message[128];

DigitalOut directionG(D8);
DigitalOut directionD(D7);
PwmOut vitesseG(D10);
PwmOut vitesseD(D11);


int main() {
    my_pc.baud(9600);
    my_rf.baud(19200);
    double val_D, val_G;
    directionG = 0 ;
    directionD = 1 ;
    sprintf(message, "Bienvenue a l'IOGS - Recept\r\n");
    my_pc.write(message, strlen(message));
    while(1){
        if (my_rf.readable()) {
            my_rf.read(&data, 1);
            val_D = (1.0/15)*(uint8_t)(data & 0x0F);
            val_G = (1.0/15)*(uint8_t)((data>>4) & 0x0F);
            printf("Val_D = %f \t Val_G = %f \n",val_D,val_G);
        // Activation moteur
        vitesseG.period_us(1000);
        vitesseD.period_us(1000);
        vitesseG.write(val_G*0.7);
        vitesseD.write(val_D*0.7);
        }
    }
}
```

Code Etat capteur de ligne

```
#include "mbed.h"
#define S 0.8
/*seuil en volt*/

/*les capeturs sont numérotés de gauche à droite côté imprimé, IOGS à l endroit*/

AnalogIn portL1(A1);
AnalogIn portL2(A2);
AnalogIn portL3(A3);

int valintL1;
int valintL2;
int valintL3;

double valdoubleL1;
double valdoubleL2;
double valdoubleL3;

int etatL1=0;
int etatL2=0;
int etatL3=0;

int etat=0;

Ticker ticker_etat;

void toggle_etat(void);




int main()
{
    ticker_etat.attach(&toggle_etat, 1s);
    while (1)
    {

    }
}

void toggle_etat(){
    valintL1 = portL1.read_u16();
    valintL2 = portL2.read_u16();
    valintL3 = portL3.read_u16();
    valdoubleL1 = (valintL1 / 65520.0) * 3.3;
    valdoubleL2 = (valintL2 / 65520.0) * 3.3;
    valdoubleL3 = (valintL3 / 65520.0) * 3.3;
    if(valdoubleL1<S) etatL1=1;
    else etatL1=0;
    if(valdoubleL2<S) etatL2=1;
    else etatL2=0;
    if(valdoubleL3<S) etatL3=1;
    else etatL3=0;

    if(etatL1==0){
        if(etatL2==0){
            if(etatL3==0){
                etat=0;
            }
            else{
                etat=1;
            }
        }
        else{
            if(etatL3==0){
                etat=2;
            }
            else{
                etat=3;
            }
        }
    }
    else{
        if(etatL2==0){
            if(etatL3==0){
                etat=4;
            }
            else{
                etat=5;
            }
        }
        else{
            if(etatL3==0){
                etat=6;
            }
            else{
                etat=7;
            }
        }
    }
    printf("état = %d \n", etat);
}
```

Code Caractéristique capteur de distance

```
#include "mbed.h"
AnalogIn in(A4);

double tension = 0;

int main(){
    while(1){
        wait_us(1000000);
        tension=in.read();
        printf("Vin=%lf V\n",tension*3.3);
    }
}
```

Code Réception avec Détection d’obstacles + Buzzer (non fonctionnel)

```
#include "mbed.h"
#include <cstdint>
#include <string.h>

UnbufferedSerial my_pc(USBTX,USBRX);
UnbufferedSerial my_rf(A0,A1);

uint8_t data;
char message[128];

DigitalOut directionG(D8);
DigitalOut directionD(D7);
PwmOut vitesseG(D10);
PwmOut vitesseD(D11);
```

```
AnalogIn in(A4);
DigitalOut buzzer(D12);
double tension_distance = 0;


Ticker ticker_distance;

int obstacle = 0;

void toggle_distance(void);

int main() {
    my_pc.baud(9600);
    my_rf.baud(19200);
    double val_D, val_G;
    directionG = 0 ;
    directionD = 1 ;
    sprintf(message, "Bienvenue a l'IOGS - Recept\r\n");
    my_pc.write(message, strlen(message));
    buzzer = 0;
    ticker_distance.attach(&toggle_distance, 100ms);
    while(1){
        if(obstacle==0 && my_rf.readable()){
            my_rf.read(&data, 1);
            val_D = (1.0/15)*(uint8_t)(data & 0x0F);
            val_G = (1.0/15)*(uint8_t)((data>>4) & 0x0F);
            printf("Val_D = %f \t Val_G = %f \n",val_D,val_G);}
        else{
            val_D = 0;
            val_G = 0;
        }
        // Activation moteur
        vitesseG.period_us(1000);
        vitesseD.period_us(1000);
        vitesseG.write(val_G*0.7);
        vitesseD.write(val_D*0.7);
        }

}

void toggle_distance(){
    tension_distance=(in.read())*3.3;
    if (tension_distance>1){
        obstacle=1;
        buzzer=1;
    }
    else{
    obstacle=0;
    buzzer=0;}

}
```