

Codigo fuente Microcontrolador de Microchip PIC16F877A

```
#include "\\Astrolaser\\py_laser.h"
```

```
#include <stdlib.h>
```

```
unsigned int32 test1;
```

```
unsigned int32 test2;
```

```
void motorx(int x){
```

```
    if(x==0){
```

```
        output_b(0b00000001);
```

```
    }
```

```
    if(x==1){
```

```
        output_b(0b00000011);
```

```
    }
```

```
    if(x==2){
```

```
        output_b(0b00000010);
```

```
    }
```

```
    if(x==3){
```

```
        output_b(0b00000110);
```

```
    }
```

```
    if(x==4){
```

```
        output_b(0b00000100);
```

```
    }
```

```
    if(x==5){
```

```
        output_b(0b00001100);
```

```
    }
```

```

if(x==6){

    output_b(0b00001000);

}

if(x==7){

    output_b(0b00001001);

}

test1=test1+1;

}

void motory(int y){

    if(y==0){

        output_b(0b00010000);

    }

    if(y==1){

        output_b(0b00110000);

    }

    if(y==2){

        output_b(0b00100000);

    }

    if(y==3){

        output_b(0b01100000);

    }

    if(y==4){

        output_b(0b01000000);

    }

    if(y==5){

        output_b(0b11000000);

    }

}

```

```

    }

    if(y==6){

        output_b(0b10000000);

    }

    if(y==7){

        output_b(0b10010000);

    }

    //test2++;

    test2=test2+1;

}

void motorx1(int x){ //pasos completos

    if(x==0){

        output_b(0b00001001);

    }

    if(x==1){

        output_b(0b00000011);

    }

    if(x==2){

        output_b(0b00000110);

    }

    if(x==3){

        output_b(0b00001100);

    }

    test1=test1+1;

}

```

```

void motory1(int y){

    if(y==0){

        output_b(0b10000000);

    }

    if(y==1){

        output_b(0b01000000);

    }

    if(y==2){

        output_b(0b00100000);

    }

    if(y==3){

        output_b(0b00010000);

    }

    test2=test2+1;

}

void main()

{

    char rs232[32];//guarda string resivido en rs232

    int32 limpia;//ayuda a limpiar la variable rs232

    signed int32 b;

    signed int32 motor_count;//para el for del motor

    signed int32 num_in;//numero combertido del string

    char acu;//variable auxiliar para guardar temporalmente la entrada del rs232

    int xacu;//cuenta la cantidad de digitos que entra el usuario

    int motorxy;//0=motor x(a) ; 1=motor y(b)

```

```

int delayX=4;

int delayY=4;

test1=0;

test2=0;

SET_TRIS_A(0);

SET_TRIS_B(0);

SET_TRIS_E(0);

output_a(0b00000000);

output_b(0b00000000);

output_e(0b00000000);

setup_adc_ports(NO_ANALOGS);

setup_adc(ADC_OFF);

setup_psp(PSP_DISABLED);

setup_spi(FALSE);

setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);

setup_timer_1(T1_DISABLED);

setup_timer_2(T2_DISABLED,0,1);

setup_comparator(NC_NC_NC_NC);

setup_vref(FALSE);

while(true){

    test1=0;

    test2=0;

    printf("ready \r\n");

    putc(7);

    xacu=0;

    acu='k';

```

```

for(limpia=0;limpia<32;limpia++){

    rs232[limpia]=NULL;

}

while((acu != 'x') && (xacu<32)){

    acu = getch();

    printf("%c",acu);

    if(acu=='a'){

        motorxy=0;

        acu='k';

    }

    if(acu=='b'){

        motorxy=1;

        acu='k';

    }

    if(acu=='l'){//prende laser

        output_high(PIN_C0);

        acu='k';

    }

    if(acu=='o'){//apaga laser

        output_low(PIN_C0);

        acu='k';

    }

    if((acu!='k') && (acu!='x') && (acu!='l') && (acu!='o')){

        rs232[xacu]=acu;

        xacu++;

    }

```

```

}

printf("\r\n");

num_in = atoi32(rs232);

if(motorxy==0){

    if(num_in>0){

        num_in=num_in*1;

        for(b=0;b<num_in;b++){

            if(motor_count<4){

                motorx1(motor_count);

                motor_count++;

                delay_ms(delayX); //4 minimo actual 5, presicion 25

            }else{

                motor_count=0;

                motorx1(motor_count);

                motor_count++;

                delay_ms(delayX); //4 minimo actual 5, presicion 25

            }

        }

    }

}

if(num_in<0){

    num_in=num_in*-1;

    for(b=0;b<num_in;b++){

        if(motor_count>=0){

            motorx1(motor_count);

            motor_count--;

            delay_ms(delayX);

```

```

    }else{

        motor_count=3;

        motorx1(motor_count);

        motor_count--;

        delay_ms(delayX);

    }

}

}

}

if(motorxy==1){

    if(num_in>0){

        num_in=num_in*1;

        for(b=0;b<num_in;b++){

            if(motor_count<4){

                motory1(motor_count);

                motor_count++;

                delay_ms(delayY);//4

            }else{

                motor_count=0;

                motory1(motor_count);

                motor_count++;

                delay_ms(delayY);//4

            }

        }

    }

}

if(num_in<0){

```



```

num_in=num_in-1;

for(b=0;b<num_in;b++){

    if(motor_count>=0){

        motory1(motor_count);

        motor_count--;

        delay_ms(delayY);

    }else{

        motor_count=3;

        motory1(motor_count);

        motor_count--;

        delay_ms(delayY);

    }

}

}

}

printf("pasos motor 1 : %Ld \r\n", test1);

printf("pasos motor 2 : %Ld \r\n", test2);

output_b(0b00000000);

}

}

```