

```

#include <Wire.h>
#include <Servo.h>
#include <EEPROM.h>
#include <I2C_eeprom.h>

I2C_eeprom ee(0x50, 128);

int slave_add = 1;
int Comando = 0;
int Envio = 0;
int flagRecepcion = 0;

int ang_cad_der, ang_pie_der, ang_cad_izq, ang_pie_izq;

int giro;

int trama[30]; //5 primeros son el código de Comando. "XXXX:". trama[5] indica,
// si procede, el número de servo.
// 6,7,8 indican (3 cifras en Ascii) valor numerico para n° grados
// de servo indicado en trama 5.
// para funciones generales a todos los servos con parametros de
// grados (escribir eeprom o mover todos al mismo tiempo):
// 6, 7, 8= cadera derecha // 9, 10, 11= pie derecho // 12, 13, 14 = cadera izq
// // 15, 16, 17 = pie izquierdo
// 18 Fin carro
char data[4] = {
    0, 0, 0, 0
}; //Para enviar las posiciones home de las articulaciones al maestro

Servo cadera_der;
Servo cadera_izq;
Servo pie_der;
Servo pie_izq;

int home_pie_izq = 0;
int home_pie_der = 0;
int home_cad_izq = 0;
int home_cad_der = 0;
int LnNumSerie = 6;

//Mapa memoria EEPROM, offset de fabricacion en los servos
//Cambios a fecha 09/09/2015: reordenado
int add_off_cad_izq = 0;
int add_off_cad_der = 1;
int add_off_pie_izq = 2;
int add_off_pie_der = 3;

int add_nombre = 5;

char NS[13] = { '$', 'O', 'K', 'N', 'S', ':', 0, 0, 0, 0, 0, 0, 0, '#' };

void setup()
{
    //Wire.begin(slave_add); // join i2c bus with address #1
    //Wire.onReceive(receiveEvent); // register event
    //Recepcion datos en esclavo (escritura desde el maestro)
    //Wire.onRequest(requestEvent); // register event

```

```

    //Petición lectura desde el maestro
    Serial.begin(115200); // start serial for output
    // Serial.println("Arrancada, Hola soy ZUM");

    //Asignación pines Servos Cambios a fecha 09/09/2015
    cadera_izq.attach(2);
    cadera_der.attach(3);
    pie_izq.attach(4);
    pie_der.attach(5);

    //Escribimos nombre
    //for(int i=0;i<4;i++){
    //int AsciiToInt(Nombre[0]);
    EEPROM.write(add_nombre, 36); //escribimos '$' en Ascii
    // }

    //Leer Numero Serie y pasar RB para log en base datos.
    ee.begin();
    for (int i = 0; i < LnNumSerie; i++) {
        NS[i + 6] = ee.readByte(i);
    }

    Serial.write(NS);

    //pinMode(13,OUTPUT);
    //digitalWrite(13,HIGH);
}

void loop()
{

    if (Serial.available()) {
        receiveEvent(0);
    }

    if (flagRecepcion == 1) { //Hemos recibido algo
        Serial.println("Recibido:");
        Serial.println(Comando);
        switch (Comando) {
            case 1: //Leer offset
                home_pie_izq = EEPROM.read(add_off_pie_izq);
                home_cad_izq = EEPROM.read(add_off_cad_izq);
                home_pie_der = EEPROM.read(add_off_pie_der);
                home_cad_der = EEPROM.read(add_off_cad_der);
                Serial.println("HOME leido de EEPROM:");
                Serial.println(home_pie_izq);
                Serial.println(home_cad_izq);
                Serial.println(home_pie_der);
                Serial.println(home_cad_der);
                Envio = 1;
                break;
            case 2: //Escribir offset

                ang_cad_der = (trama[6] - '0') * 100 + (trama[7] - '0') * 10 +
                    (trama[8] - '0');
                ang_pie_der = (trama[9] - '0') * 100 + (trama[10] - '0') * 10 +
                    (trama[11] - '0');

```

```
// ang_cad_izq = (trama[12]-'0')*100 + (trama[13]-'0')*10 +
//              (trama[14]-'0');
// ang_pie_izq = (trama[15]-'0')*100 + (trama[16]-'0')*10 +
//              (trama[17]-'0');
```

```
//iteracion 4:
```

```
ang_cad_der = ang_cad_der - 90;
ang_pie_der = ang_pie_der - 90;
// ang_cad_izq = ang_cad_izq-90;
// ang_pie_izq = ang_pie_izq-90;
```

```
//EEPROM.write(add_off_pie_izq,ang_pie_izq); //3
//delay(30);
//EEPROM.write(add_off_cad_izq,ang_cad_izq); //2
//delay(30);
```

```
EEPROM.write(add_off_pie_der, ang_pie_der); //1
delay(30);
EEPROM.write(add_off_cad_der, ang_cad_der); //0
delay(30);
```

```
// Serial.println("HOME escrito en EEPROM:");
// Serial.println(ang_cad_der);
// Serial.println(ang_pie_der);
// Serial.println(ang_cad_izq);
// Serial.println(ang_pie_izq);
```

```
break;
```

```
case 22: //Escribir offset
```

```
// ang_cad_der = (trama[6]-'0')*100 + (trama[7]-'0')*10 + (trama[8]-'0');
// ang_pie_der = (trama[9]-'0')*100 + (trama[10]-'0')*10 +
// (trama[11]-'0');
```

```
ang_cad_izq = (trama[6] - '0') * 100 + (trama[7] - '0') * 10 +
              (trama[8] - '0');
ang_pie_izq = (trama[9] - '0') * 100 + (trama[10] - '0') * 10 +
              (trama[11] - '0');
```

```
//iteracion 4:
```

```
// ang_cad_der = ang_cad_der-90;
// ang_pie_der = ang_pie_der-90;
ang_cad_izq = ang_cad_izq - 90;
ang_pie_izq = ang_pie_izq - 90;
```

```
EEPROM.write(add_off_pie_izq, ang_pie_izq); //3
delay(30);
EEPROM.write(add_off_cad_izq, ang_cad_izq); //2
delay(30);
```

```
//EEPROM.write(add_off_pie_der,ang_pie_der); //1
//delay(30);
//EEPROM.write(add_off_cad_der,ang_cad_der); //0
//delay(30);
```

```
// Serial.println("HOME escrito en EEPROM:");
// Serial.println(ang_cad_der);
// Serial.println(ang_pie_der);
// Serial.println(ang_cad_izq);
// Serial.println(ang_pie_izq);
```

```
break;
```

```

case 3: //Mover a 90°
    cadera_der.write(90);
    cadera_izq.write(90);
    pie_der.write(90);
    pie_izq.write(90);
    Serial.println("A 90");
    break;
case 4: //Mover a HOME
    home_pie_izq = EEPROM.read(add_off_pie_izq);
    home_cad_izq = EEPROM.read(add_off_cad_izq);
    home_pie_der = EEPROM.read(add_off_pie_der);
    home_cad_der = EEPROM.read(add_off_cad_der);
    cadera_der.write(home_cad_der);
    cadera_izq.write(home_cad_izq);
    pie_der.write(home_pie_der);
    pie_izq.write(home_pie_izq);
    Serial.println("A HOME");
    break;
case 5: //Mover servos a posicion especifica

    ang_cad_der = (trama[6] - '0') * 100 + (trama[7] - '0') * 10 +
        (trama[8] - '0');
    ang_pie_der = (trama[9] - '0') * 100 + (trama[10] - '0') * 10 +
        (trama[11] - '0');
    ang_cad_izq = (trama[12] - '0') * 100 + (trama[13] - '0') * 10 +
        (trama[14] - '0');
    ang_pie_izq = (trama[15] - '0') * 100 + (trama[16] - '0') * 10 +
        (trama[17] - '0');

    cadera_der.write(ang_cad_der);
    pie_der.write(ang_pie_der);
    cadera_izq.write(ang_cad_izq);
    pie_izq.write(ang_pie_izq);

    Serial.println("Todos A POS dada");
    break;
case 6: //Mover servo indicado a posicion especifica

    giro = 0;
    giro = (trama[6] - '0') * 100;
    giro = giro + (trama[7] - '0') * 10;
    giro = giro + (trama[8] - '0');

    switch (trama[5] - '0') {
    case 1: //Cadera derecha
        cadera_der.write(giro);
        Serial.println("Cad_der A POS dada");
        break;
    case 2: //Pie derecho
        pie_der.write(giro);
        Serial.println("pie_der A POS dada");
        break;
    case 3: //Cadera izquierda
        cadera_izq.write(giro);
        Serial.println("Cad_izq A POS dada");
        break;
    case 4: //Pie izquierdo

```

```

        pie_izq.write(giro);
        Serial.println("pie_izq A POS dada");
        break;
    default:
        Serial.println("Seleccion servo NOK");
        break;
    }
    break;
default:
    Serial.println("Comando NOK");
    break;
}

```

```

Serial.println("Trama recibida:");
Serial.println(trama[0]);
Serial.println(trama[1]);
Serial.println(trama[2]);
Serial.println(trama[3]);
Serial.println(trama[4]);
Serial.println(trama[5]);
Serial.println(trama[6]);
Serial.println(trama[7]);
Serial.println(trama[8]);
Serial.println(trama[9]);
Serial.println(trama[10]);
Serial.println(trama[11]);
Serial.println(trama[12]);
Serial.println(trama[13]);
Serial.println(trama[14]);
Serial.println(trama[15]);
Serial.println(trama[16]);
Serial.println(trama[17]);
Serial.println(trama[18]);
Serial.println(trama[19]);
Serial.println(trama[20]);

```

```

flagRecepcion = 0;
trama[0] = 0;
trama[1] = 0;
trama[2] = 0;
trama[3] = 0;
trama[4] = 0;
trama[5] = 0;
trama[6] = 0;
trama[7] = 0;
trama[8] = 0;
trama[9] = 0;
trama[10] = 0;
trama[11] = 0;
trama[12] = 0;
trama[13] = 0;
trama[14] = 0;
trama[15] = 0;
trama[16] = 0;
trama[17] = 0;
trama[18] = 0;
trama[19] = 0;

```

```

    trama[20] = 0;
    trama[21] = 0;
    trama[22] = 0;
    trama[23] = 0;
    trama[24] = 0;
    trama[25] = 0;
    trama[26] = 0;
    trama[27] = 0;
    trama[28] = 0;
    trama[29] = 0;
}
delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany)
{
    int indice = 0;

    while (1 <= Serial.available()) // loop through all but the last
    {
        trama[indice] = Serial.read(); // receive byte as a character
        indice++;
    }

    switch (trama[0]) {
    case 82: //"R":
        //if(trama[1]=="O" && trama[2]=="F" && trama[3]=="C" &&trama[4]==":") {
        //Comando=1;};
        if (trama[1] == 79 && trama[2] == 70 && trama[3] == 67 && trama[4] == 58) {
            Comando = 1;
        };
        break;
    case 87: //"W":
        //if(trama[1]=="O" && trama[2]=="F" && trama[3]=="C" &&trama[4]==":") {
        // Comando=2;};
        if (trama[1] == 79 && trama[2] == 70 && trama[3] == 67 && trama[4] == 58) {
            if (trama[5] == 49) //Derecha(1)
            {
                Comando = 2;
            }
            else if (trama[5] == 50) //Izquierda(2)
            {
                Comando = 22;
            }
        };
        break;
    case 77: //"M":
        //if(trama[1]=="9" && trama[2]=="0" && trama[3]=="C" &&trama[4]==":") {
        // Comando=3;};
        if (trama[1] == 57 && trama[2] == 48 && trama[3] == 67 && trama[4] == 58) {
            Comando = 3;
        };
        //if(trama[1]=="H" && trama[2]=="O" && trama[3]=="C" &&trama[4]==":") {
        // Comando=4;};
        if (trama[1] == 72 && trama[2] == 79 && trama[3] == 67 && trama[4] == 58) {

```

```

        Comando = 4;
    };
    //if(trama[1]=="S" && trama[2]=="S" && trama[3]=="C" &&trama[4]==":") {
    // Comando=5;};
    if (trama[1] == 83 && trama[2] == 83 && trama[3] == 67 && trama[4] == 58) {
        Comando = 5;
    };
    if (trama[1] == 83 && trama[2] == 120 && trama[3] == 67 && trama[4] == 58) {
        Comando = 6;
    };
    break;
default:
    Comando = 0;
    break;
}

flagRecepcion = 1;
}

void requestEvent()
{ //Preparamos la trama para enviar al maestro
    switch (Envio) {
        case 1:
            data[0] = home_cad_izq;
            data[1] = home_pie_izq;
            data[2] = home_cad_der;
            data[3] = home_pie_der;
            data[4] = 0;
            Wire.write(data); // respond with message of 11 bytes
            // as expected by master
            break;
        default:
            break;
    }
    Envio = 0;
    data[0] = 0;
    data[1] = 0;
    data[2] = 0;
    data[3] = 0;
    data[4] = 0;
}

```