

PROGRAMACION

ROBOT CS-113



@ Fco. Larrosa (2000-2022)
Universidad Laboral Albacete

DESCRIPCIÓN DEL ROBOT CS-113

El robot -CS 113 -es un sistema inteligente controlado por microprocesador, capaz de moverse articuladamente en seis ejes. Está diseñado para simular el comportamiento de un robot industrial .

El control del robot puede realizarse desde cualquier ordenador por medio de simples instrucciones en "BASIC" o sus equivalentes en Lenguaje C.

DESCRIPCION MECANICA :

El robot es del tipo "articulación coordinada " , llamado así por la semejanza de sus movimientos con los del cuerpo humano. El movimiento del **primer eje** corresponde con el de la **cintura**. El **segundo eje** se corresponde con el movimiento **vertical del hombro**. El **tercer eje** corresponde con el **giro del codo** también en sentido vertical. La combinación del **cuarto y quinto eje** produce los movimientos de **giro e inclinación de la muñeca** y el **sexto eje** efectúa los movimientos de **abrir y cerrar la mano**.

ESPECIFICACIONES ELECTRICAS

Los elementos principales del circuito electrónico de control son:

- El microprocesador (Z-80A) como unidad principal de control
- La memoria ROM que contiene el programa del sistema
- La memoria RAM que contiene la zona de datos del sistema
- Los circuitos de accionamiento de los motores

Las especificaciones detalladas de las características generales como de los márgenes de movimiento de los motores o interconexión con distintos ordenadores se pueden consultar en las tablas 1 a 3 que se muestran a continuación :

TABLA 1.A	CARACTERÍSTICAS GENERALES
APLICACION	EDUCACION, INVESTIGACION, HOBBY
CONSTRUCCION	ESTRUCTURA DE PLANCHA METALICA
PRECISION DEL POSICIONADO	+/- 0.9 mm.
CAPACIDAD DE CARGA	500 g. (max.)
APERTURA DE PINZA	45 mm.
VELOCIDAD DE MOVIMIENTO	300 mm. /se9. (max.) <i>variable</i>
PESO DEL ROBOT	8 kg. (aprox.)
GRADOS DE LIBERTAD	5

TABLA 1.B	CS 113 MARGEN DE GIRO MOTORES
CUERPO	240° (+/- 120)
HOMBRO	140° (+/- 70)
CODO	100° (+/- 50)
INCLINACIÓN DE MUÑECA	180°
ROTACION DE LA MUÑECA	360°

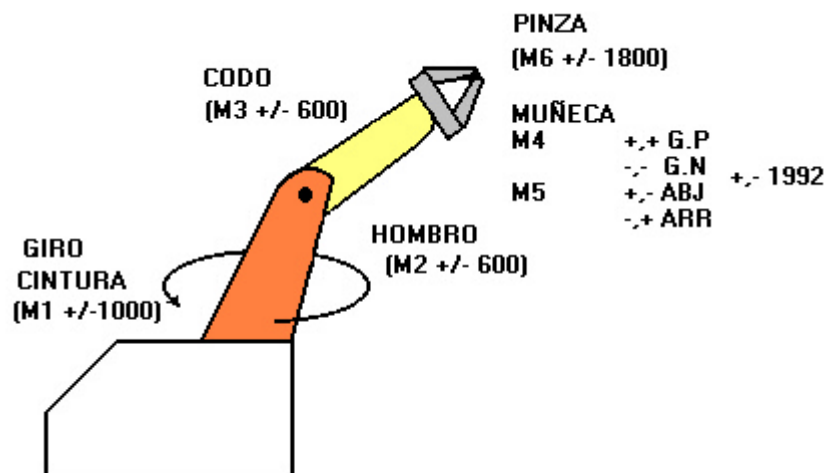
TABLA 1.C	CS 113 ALIMENTACION ELECTRICA Y CONTROL
MOTORES	6 MOTORES PASO A PASO
CONTROL DE VELOCIDAD	CONTROL POR ONDA CUADRADA
INTERFACES	CENTRONICS
LENGUAJES	BASIC, ASSEMBLER
PROGRAMACION	COMANDOS ESPECIALES
ALIMENTACION	AC 110/220V. 50/60 Hz.

TABLA 2

CS 113 ESPECIFICACIONES DEL MOVIMIENTO

MOTOR	EJE	PASO	RANGO MAXIMO EN PASOS	
			DIRECC. (+)	DIRECC. (-)
M1	CINTURA	0.12°	1000 (GIRO +)	1000 (GIRO -)
M2	HOMBRO	0.12°	600 (ARRIBA)	600 (ABAJO)
M3	CODO	0.08°	600 (ARRIBA)	600 (ABAJO)
M4	MUÑECA	0.05°	+,+ GIRO NEGATIVO	
			-, - GIRO POSITIVO → max. 1992	
M5			+,- ABAJO	
			-,+ ARRIBA	
M6	MANO	0.1°	+ 1800 → cerrar	-1800 → abrir

NOTA: Si la entrada correspondiente a M4 y M5 es p.ej. : (996,-996) la muñeca se moverá hacia abajo 90°. , si la entrada fuera (-996,-996) giraría en el sentido positivo (según las agujas del reloj) 90°. Si M4>M5 o M4<M5, se producen simultáneamente los movimientos de inclinación y giro.



MODOS DE OPERACIÓN

Observaciones previas :

- 1) Asegúrese de que el selector de tensión de red esté en la posición adecuada (110 o 220v.)
- 2) **No ajustar la posición de los ejes manualmente.** Cuando esté en movimiento, no debe permitirse que ningún objeto situado en su trayectoria pueda bloquear su movimiento. El bloqueo del movimiento del robot cambia la posición de referencia programada y puede ocasionar desperfectos en el sistema mecánico
- 3) No manipular en las partes móviles del robot para evitar posibles lesiones.
- 4) No debe programarse el robot para realizar movimientos fuera de los límites de su campo de acción.
- 5) Después de conectar la fuente de alimentación del robot, pulsar una vez el botón de reset .En funcionamiento bajo el control del programa, si se enciende el indicador de error en el robot o bien si no responde a las señales de control, accionar el pulsador de reset.
- 6) Comprobar que el cable de conexión entre el robot y el ordenador (CENTRONIC) esté bien sujeto.
- 7) Consultar la tabla nº 3 la forma de conexión adecuada entre el robot y el microcomputador que se utilice.
- 8) No debe tratarse de desensamblar ni tirar del cable de conexión que enlaza al robot a el microcomputador.
- 9) Posicionar al robot horizontalmente y evitar posibles golpes al robot.
- 10) Evitar la operación del robot sometido a vibraciones importantes, polvo, humedad, excesivo ruido eléctrico, etc. No obstruir la salida de refrigeración situada en la parte inferior de la carcasa.
- 11) Tener en cuenta que el robot NO a sido diseñado para operaciones industriales.

CONEXIÓN Y VERIFICACION :

1) Conectar el robot a la tensión de red adecuada. Accionar el interruptor de puesta en marcha y accionar una vez el pulsador de reset para liberar al robot del control del microcomputador. En este momento el robot está preparado para realizar un test de verificación. Hay dos posibles test de verificación.

2) El indicador verde de la parte frontal del robot se enciende al conectarlo a la fuente de alimentación. Si la pinza está abierta en el instante de pulsar el botón reset, se cierra automáticamente hasta alcanzar su posición máxima de cerrado.

3) En el caso de que el motor M6 gire mas allá de su límite sin que se llegue a accionar el microinterruptor que detecta el cierre de la pinza, se encenderá la lámpara de error (ROJA). En estas condiciones el robot solo obedece al pulsador de reset.

TEST DE FUNCIONAMIENTO

• **Primer modo de test** :este test debe efectuarse con el robot **sin carga**(desconectado del puerto). Su propósito es garantizar la posibilidad de movimientos de todas las uniones. También puede emplearse **para situar al robot en la posición de referencia** como paso previo a cualquier acción exterior de control de la máquina. La posición inicial o de referencia corresponde a la posición en que todas sus flechas indicadoras están alineadas ($\rightarrow \leftarrow \dots$), en esta posición los pasos de los motores son el de **CERO** según la tabla dada de los límites de cada motor. El método a seguir es el siguiente :

1) Pulsar el botón reset y medio segundo después el de test. Mantener ambos botones pulsados y soltar el botón de reset. Continuar pulsando el botón test durante unos **3 segundos** y soltarlo.

En el caso en que la pinza del robot estuviera cerrada (si no es así la cerrará) el robot realizará **una acción cada vez que se pulse el botón test**. El movimiento correspondiente no cesará mientras se tenga el botón pulsado.

La secuencia de movimientos que se irá desarrollando a medida que se vaya accionando el pulsador de test será la siguiente :

M1 (cintura) → giro negativo

M1 (cintura) → giro positivo

M2 (hombro) → elevación

M2 (hombro) → descenso

M3 (codo) → descenso

M3 (codo) → elevación

Giro negativo de pinza

Giro positivo de pinza

Descenso de la pinza

Elevación de la pinza

Abrir pinza

Cerrar pinza

A la siguiente pulsación de la tecla test se repite el proceso accionándose el motor M1. Al realizar este, se debe procurar NO sobrepasar los límites de movimiento de los motores, para no causar trastornos en los motores del robot.

• **Segundo modo de test** : Los movimientos realizados en esta segunda prueba de verificación están programados en ROM del-CS 113-. Este test está pensado para probar la **sincronización de los movimientos** de los distintos motores, la repetitividad de los movimientos y los ángulos. Para su realización, observese el siguiente método de operación :

1) Alinear adecuadamente el robot (utilizando el test 1) dejándolo en posición CERO. Pulsar el botón de reset para abandonar el primer modo de test. **Es importante asegurarse de que el robot quede lo más exactamente posible en posición CERO antes de empezar este nuevo test.**

2) Pulsar y mantener apretado el botón de reset en primer lugar y seguidamente el de test, sin soltar el de reset. Soltar el pulsador de reset y mantener pulsado el de test durante unos **5 segundos**.

3) Al cabo de este tiempo el robot empieza a realizar una serie de movimientos sincronizados bajo el control de su propio programa en ROM. En este instante debe soltarse el botón de test.

4) La velocidad de los movimientos varía entre los 5 posibles niveles siguiendo una secuencia 1-3-5 de forma indefinida.

5) Para **finalizar** este procedimiento de test, debe mantenerse pulsado el botón de **test durante un segundo**. El robot se detendrá en la **posición CERO** después de completar el ciclo a velocidad 5. Pulsar el botón de reset para recuperar el control de la máquina.

DESCRIPCION DE LAS INSTRUCCIONES (**INSTRUCCIONES ORIGINALES EN BASIC**)

ZERO : Esta instrucción indica al robot que la posición actual que tiene este es la de referencia o posición cero (HOME). Todos los movimientos ordenados se realizan tomando como posición de referencia la posición que físicamente tenía el robot al recibir este comando. **Este comando es obligatorio** porque marca las referencias de los movimientos.

- Formato: **Z**
- Ejemplo: **LPRINT "Z"**
- Ascii: **5A + <cr>** **ascii 0D**

MOVE : Ordena un desplazamiento relativo a la posición actual, indicando el número de pasos a realizar por cada motor y su signo.

- Formato: **Ma1,a2,a3,a4,a5,a6** a1...a6 → Motores

- Ejemplo: **LPRINT "M500,0,-150,0,0,0"**

- Ascii: **4D0D**

.....→ Para codificar M500,0,150,0 sería:

4D 35 30 30 2C 30 2C 31 35 30 2C 30

M 5 0 0 , 0 , 1 5 0 , 0 →Son los Ascii de los
numeros que intervienen

HERE : Mediante esta instrucción se memoriza en RAM la posición actual de cada motor del robot. Podemos almacenar hasta 100 posiciones

- Formato: **Hn** n→Posición

- Ejemplo: **LPRINT "H2"**

- Ascii: **48n 0D** n→Ascii de n

POSICION : Memoriza una posición dando los incrementos relativos de cada motor respecto a la posición de referencia. Al igual que en el caso anterior, el número de puntos que se pueden guardar por el procedimiento de RAM estándar del robot es de 100.

- Formato: **Pa1,a2,a3,a4,a5,a6**

- Ejemplo: **LPRINT "P8,-500,500,-200,0,0,0"**

- Ascii: **50 0D**

GOTO : Indica al robot que se posicione en uno de los puntos memorizados mediante las instrucciones "HERE" o "POSICION"

-Formato: **Gn**

- Ejemplo: **LPRINT "G8"** n corresponde a la posición en RAM

- Ascii: **47 n 0D**

HOME: Mediante esta instrucción se ordena al robot volver a la posición de referencia marcada por las instrucción ZERO.

- Formato: **N**
- Ejemplo: **LPRINT "N"**
- Ascii: **4E 0D**

GRIP CLOSE : Cierra la pinza hasta su límite máximo.

- Formato: **C**
- Ejemplo: **LPRINT "C"**
- Ascii: **43 0D**

GRIP OPEN : Abre la pinza hasta su límite máximo.

- Formato: **O**
- Ejemplo: **LPRINT "O"**
- Ascii: **4F 0D**

SPEED : Selecciona la velocidad de los motores del robot entre los 5 valores posibles.

- Formato: **Sn**
- Ejemplo: **LPRINT "S3"**
- Ascii: **53 n 0D**

DELAY: Detiene el movimiento del robot durante un periodo de tiempo en segundos,pero los valores están acotados entre 1 y 9

- Formato: **Dn**
- Ejemplo: **LPRINT "D3"**
- Ascii: **44 n 0D**

LIMIT : Delimita los límites máximos y mínimos de los motores del robot, si en caso de introducir un movimiento el cual es mayor que los límites establecidos para los motores del robot, se parará el robot y se encenderá el led de ERROR. El robot solo obedecerá a un reseteado o a la instrucción NEST. **v** puede valer 1 ó 0, Si vale 1 se activa el modo de vigilancia de límites y si vale = se desactivará esta protección. Por defecto está desactivada.

- Formato: **Lv**
- Ejemplo: **LPRINT "L1"**
- Ascii: **4C v 0D**

RESUMEN DE LA LISTA DE INSTRUCCIONES

"Z"	Indica la posición de referencia.
"M1 , 2, 3, 4, 5, 6"	Desplazamiento de cada uno de los motores.
"H2"	Memoriza la posición actual de cada motor .
"PB, 1, 2, 3, 4, 5, 6"	Memoriza la posición con los pasos de cada motor .
"G8"	Indica que se sitúe en una posición memorizada.
"N"	Ordena ir a la posición HOME (ZERO)
"C"	Cierra la pinza hasta su límite máximo
"O"	Abre la pinza hasta su límite máximo.
"S3"	Velocidad de los motores.
"D9"	Detiene el movimiento durante un tiempo determinado.
"L1"	Activa la comprobación de los límites de los motores.

RUTINA DE PROGRAMACIÓN CENTRONIC

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <dos.h>

void Envia (char);

void main (void)
{
    char m,i,x[35];

    strcpy (x,"Z\n");
    m=strlen(x);
    for (i=0;i<m;i++) Envia (x[i]);

    strcpy (x,"M-500,0,0,0,750,735\n");
    m=strlen(x);
    for (i=0;i<m;i++)Envia (x[i]);

    strcpy (x,"N\n");
    m=strlen(x);
    for (i=0;i<m;i++)Envia (x[i]);
}

void Envia (char dato)
{
    outp (0x37A,1);                //STB=1
    while (!(inp(0x379)& 0X80));    //BUSY=1?
    outp (0X378,dato);
    outp (0x37A,0);                //STB=0
    delay(10);
    outp (0x37A,1);                //STB=1
}

```

(INSTRUCCIONES ORIGINALES EN BASIC)

EJEMPLO PROGRAMA ROBOT

```

0 LPRINT I                ; Programa nº I.
5 LPRINT "L0"             ; Límites desactivados.
10 LPRINT "H1 "           ; Memoriza posición 1.
15 LPRINT "S5"            ; Velocidad motores máxima.
20 LPRINT "M-500,0,0,0,0" ; Giro cintura izquierda.
30 LPRINT "M0,-400,0,600,-600,-1490" ; Baja hombro, baja muñeca y abre pinza.
35 LPRINT "S3"            ; Velocidad motor intermedia.
40 LPRINT "M0, 75,0,0,0,0" ; Baja hombro.
45 LPRINT "M0,0,0,0,0,800" ; Cierra pinza. (Coge cubo)
55 LPRINT "M1000,600,0,0,0,0" ; Giro cintura derecha, sube hombro.
56 LPRINT "H3 "           ; Memoriza posición 3.

```

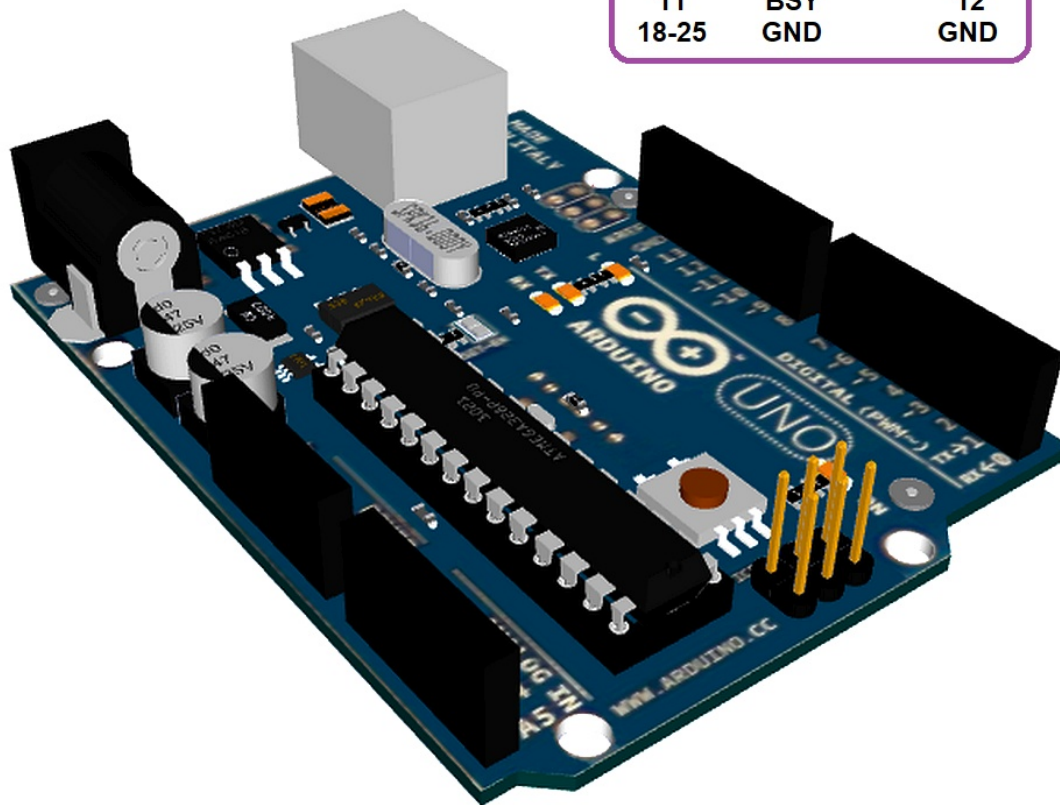
60 LPRINT "M0,-600,0,0,0,0"	; Baja hombro.
65 LPRINT "M0,0,0,0,0,700"	; Abre pinza. (Deja cubo)
70 LPRINT "S5"	; Velocidad motores máxima.
75 LPRINT "M0,600,0,0,0,500"	; Sube hombro, cierra pinza.
80 LPRINT "M-1200,0,0,0,0,0"	; Giro de cintura izquierda.
85 LPRINT "H2"	; Memoriza posición 2.
90 LPRINT "M0,-450,0,0,0,0"	; Baja hombro.
95 LPRINT "M0,0,-250,0,0,-200"	; Baja codo, abre pinza.
100 LPRINT "M0,-60,0,0,0,0"	; Baja hombro.
105 LPRINT "M0,0,0,0,0,500"	; Cierra pinza. (Coge cubo)
110 LPRINT "G2"	; Vete a la posición 2.
115 LPRINT "M1280,0,0,996,996,0"	; Giro cintura derecha, giro 90° muñeca.
116 LPRINT "H4"	; Memoriza posición 4.
120 LPRINT "M-5,-625,70,60,-60,20"	; Giro cintura izquierda, baja hombro, sube codo, baja muñeca.
125 LPRINT "M0,0,0,0,0,750"	; Abre pinza. (Deja cubo)
130 LPRINT "M0,500,0,0,0,0"	; Sube hombro.
135 LPRINT "G2"	; Vete a la posición 2.
140 LPRINT "M0,-580,200,-70,70,-300"	; Baja hombro, sube codo y muñeca, abre pinza.
150 LPRINT "M0,-150,0,0,0,-200"	; Baja hombro, abre pinza.
155 LPRINT "M0,-30,0,-90,90,0"	; Baja hombro, sube muñeca.
165 LPRINT "M0,0,0,0,0,700"	; Cierra pinza. (Coge cubo)
168 LPRINT "M0,20,0,0,0,0"	; Sube hombro.
170 LPRINT "G3"	; Vete a la posición 3.
175 LPRINT "M-8,-480,-250,0,0,0"	; Giro cintura izquierdo, baja hombro y codo.
180 LPRINT "M0,0,0,0,0,-300"	; Abre pinza. (Deja cubo)
185 LPRINT "M0,150,0,0,0,0"	; Sube hombro.
190 LPRINT "G2"	; Vete a posición 2.
200 LPRINT "M-200,-450,0,0,0,-300"	; Giro cintura izquierda, baja hombro, abre pinza.
205 LPRINT "M0, 70,0,0,0,0"	; Baja hombro.
210 LPRINT "M0,0,0,0,0,600"	; Cierra pinza.(Coge cubo)
214 LPRINT "M0,20,0,0,0,0"	; Sube hombro.
215 LPRINT "G4"	; Vete a posición 4.
216 LPRINT "S5"	; Velocidad motores máxima.
217 LPRINT "M0,0,0,920,920,0"	; Gira muñeca.
220 LPRINT "M12,-465,-190,0,0,200"	; Giro cintura derecha, baja hombro y codo.
225 LPRINT "M0,0,0,0,0,-250"	; Abre pinza. (Deja cubo)
230 LPRINT "M0,150,0,0,0,0"	; Sube hombro
240 LPRINT "N"	; Vete a posición de referencia.
245 LPRINT "M-90,-300,-20,60,-60,-15"	; Giro cint. Izq.,Baja hombro y codo, abre pinza.
255 LPRINT "M0,0,0,0,0,1000"	; Cierra pinza. (Coge cubo)
250 LPRINT "M0, 70,0,0,0,0"	; Baja hombro.
258 LPRINT "M0,20,0,0,0,0"	; Sube hombro.
260 LPRINT "G4"	; Vete posición 4.
275 LPRINT "M5,-455,-100,0,0,300"	; Giro derecho de cintura, baja hombro y codo.

280 LPRINT "M0,0,0,0,0,-800" ; Abre pinza. (Deja cubo)
 285 LPRINT "M300,300,300,0,0,0" ; Giro cintura derecho, sube hombro y codo
 290 LPRINT "M0,-420,-100,0,0,0" ; Baja hombro y codo.
 295 LPRINT "M0,0,0,0,0,1200" ; Cierra pinza. (Coge cubo)
 300 LPRINT "G3" ; Vete a posición 3.
 305 LPRINT "M0,-435,-100,-900,-900,0" ; Baja hombro y codo, giro de muñeca.
 310 LPRINT "M0,0,0,0,0,-700" ; Abre pinza. (Deja cubo)
 315 LPRINT "M0,400,0,0,0,0" ; Sube hombro.
 320 LPRINT "M200,-570,270,0,0,-140" ; Giro cintura derecho, baja hombro, sube
 codo, abre pinza.
 325 LPRINT "M0,0,0,0,0,800" ; Cierra pinza. (Coge cubo)
 330 LPRINT "M0,300,0,0,0,0" ; Sube hombro.
 335 LPRINT "M-160,-60,-270,0,0,0" ; Giro cintura izquierdo, baja hombro y codo
 340 LPRINT "M0,-5,0,0,0,0" ; Baja hombro.
 345 LPRINT "M0,0,0,0,0,-800" ; Abre pinza. (Deja cubo)
 348 LPRINT "M0,150,0,0,0,0" ; Sube hombro.
 350 LPRINT "N" ; Vete a posición de referencia.
 355 LPRINT "D3" ; Tiempo retardo de tres segundos.
 360 LPRINT "M380,-400,-18,30,-30,0" ; Gir. cint. derch, baja hombro y codo, baja
 muñeca.
 365 LPRINT "M0,0,0,996,996,0" ; Gira 90° muñeca.
 370 LPRINT "M300,0,0,0,0,0" ; Giro derecha de cintura.
 375 LPRINT "S5" ; Velocidad de Motores máxima.
 380 LPRINT "N" ; Vete a posición de referencia

ADAPTACION ROBOT CS-113 A ARDUINO

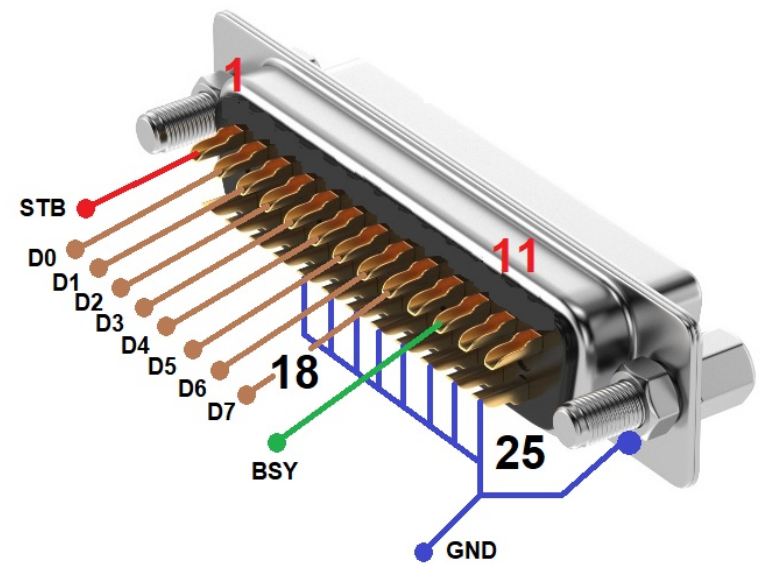
Fco. Larrosa 2022 Universidad Laboral Albacete

ARDUINO



DB25	NOMBRE	ARDUINO
1	STB	2
2	D0	3
3	D1	4
4	D2	5
5	D3	6
6	D4	7
7	D5	8
8	D6	9
9	D7	10
11	BSY	12
18-25	GND	GND

DB29 HEMBRA



```
int m, i;
char x[35];
int STB = 2;
int BUSY = 12;
// Asignamos los pines de datos [Pin-D0=3...Pin-D7=10]

void setup()
{
  Serial.begin(9600);
  for (i = 2; i <=10; i++) pinMode (i,OUTPUT);
  pinMode (BUSY,INPUT);
  digitalWrite (13,0);

  // Primero debemos situar los motores del robot
  // en la posicion de origen (Modo Test)

  // Establecemos la posicion HOME
  strcpy (x, "Z\n");
  m = strlen(x);
  for (i = 0; i < m; i++) Envia (x[i]); //Mandamos el comando

  //Movemos algunos motores de prueba
  strcpy (x, "M-500,0,0,0,750,735\n");
  m = strlen(x);
  for (i = 0; i < m; i++)Envia (x[i]);

  //Movemos algunos motores de prueba
  strcpy (x, "M0,100,0,0,0,\n");
  m = strlen(x);
  for (i = 0; i < m; i++)Envia (x[i]);
  strcpy (x, "M0,-100,0,0,0,\n");
  m = strlen(x);
  for (i = 0; i < m; i++)Envia (x[i]);
}
```

```
//Mandamos al robot a la posicion HOME
strcpy (x, "N\n");
m = strlen(x);
for (i = 0; i < m; i++)Envia (x[i]);
}

void loop()
{
  // put your main code here, to run repeatedly:

  // Protocolo de envio de datos @F.Larrosa
void Envia (char dato)
{
  digitalWrite (STB, 1); //STB=1
  while (digitalRead(BUSY)); //BUSY=1?
  OutDato(dato); //Activamos los pines del Arduino
  digitalWrite (STB, 0); //STB=0
  delay(1);
  digitalWrite (STB, 1); //STB=1
  delay(1);
}

void OutDato (char DATO)
{
  //Ponemos el dato [PinD0=3...PinD7=10]
  int j;
  int peso = 1;
  for (j = 3; j <=10; j++)
  { if ((DATO & peso) > 0) digitalWrite (j, 1);
    else digitalWrite (j, 0);
    peso = peso * 2;
  }
  delay(5);
}
```