

# ADAPTACIÓN ROBOT MENTOR *alecop* PARA ARDUINO



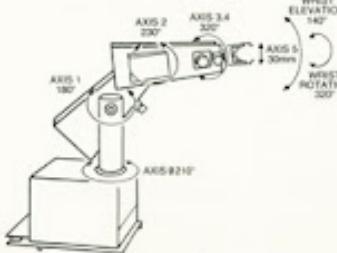
**mentor desk top robot**

- A low cost, practical introduction to robotics.

Featured in the September 1984 to January 1985 issues of PRACTICAL ELECTRONICS



A compact, electrically powered desk-top robot giving six axes of movement all simultaneously servo-controlled. "Mentor" has been specially designed to give smooth controlled performance, coupled with a high degree of built-in toughness - a combination ideal for use in educational establishments. Program length is limited only by computer memory of the user. Features such as integral control electronics and power supply, long-life bronze and nylon bearings, special circuitry for inertial compensation, optional on-board ADC and hand-held simulator as the teaching pendant, make the all-British designed and manufactured Mentor an extremely cost-effective teaching instrument.



**mentor Simulator**



Change the positions of the simulator arm and the robot will copy the movement. The simplest possible means of teaching is used. The computer is able to store all the moves and replay them whenever required.



## ÍNDICE

1. OBJETIVO GENERAL.....	1
1.1. Detalles generales del brazo.....	
2. SISTEMA DE CONTROL.....	5
2.1. Elementos.....	
3. DISEÑOS INICIALES PCB.....	7
4. ELECTRÓNICA.....	13
4.1. Fuente de alimentación.....	
4.2. Relé.....	
4.3. ULN2803.....	
4.4. Arduino Mega 2560.....	
5. CONEXIONES.....	15
5.1. Mando.....	
6. SOFTWARE.....	16
7. CONCLUSIÓN.....	22

## **1. OBJETIVO GENERAL**

El presente proyecto se enfoca al control de un robot manipulador por medio de su mando original, permitiendo monitorizar los valores desde un ordenador. Sin disponer de la información de software ni de hardware del fabricante se procederá a diseñar y elaborar los circuitos, interfaz y programas que conformaran el sistema de control mentor, así como la integración del mismo.

Para el proyecto hemos decidido usar un Arduino para construir el brazo robot, porque ofrece ventajas significativas.

Primero, su entorno de desarrollo es fácil de usar, ideal para quienes son nuevos en la programación de microcontroladores. Además, Arduino se beneficia de una comunidad de usuarios muy amplia y activa, proporcionando una gran cantidad de recursos, tutoriales y soluciones compartidas que facilitan la resolución de problemas comunes. También es compatible con una diversidad de sensores y actuadores, muchos de los cuales vienen con bibliotecas específicas para Arduino, simplificando la integración y el control en proyectos de robótica.

Desde el punto de vista económico, Arduino resulta más accesible que muchos otros microcontroladores, con una variedad de modelos y "shields" que ofrecen funcionalidades adicionales, como controladores de motor integrados, a bajo coste. Además, el soporte para múltiples plataformas operativas y su naturaleza de código abierto permiten una gran flexibilidad y personalización del proyecto.

En resumen, para un proyecto educativo o de entretenimiento, Arduino proporciona un excelente equilibrio entre funcionalidad, facilidad de uso y costo, haciéndolo una opción muy atractiva para la construcción de brazos robóticos.

El proyecto se divide en toda la parte de hardware, es decir, todos los CI, microprocesadores, sensores, actuadores y la propia construcción de la PCB y la parte de Software en la que desarrollaremos el código implementado.

## 1.1. Detalles generales del brazo mentor

Robot manipulador: Mentor(Nº de Serie: 803132), fabricado por Cybernetic Applications, bajo fines educativos.

El mismo presenta una configuración de tipo manipulador articulado y cuenta con una arquitectura poliarticulada.

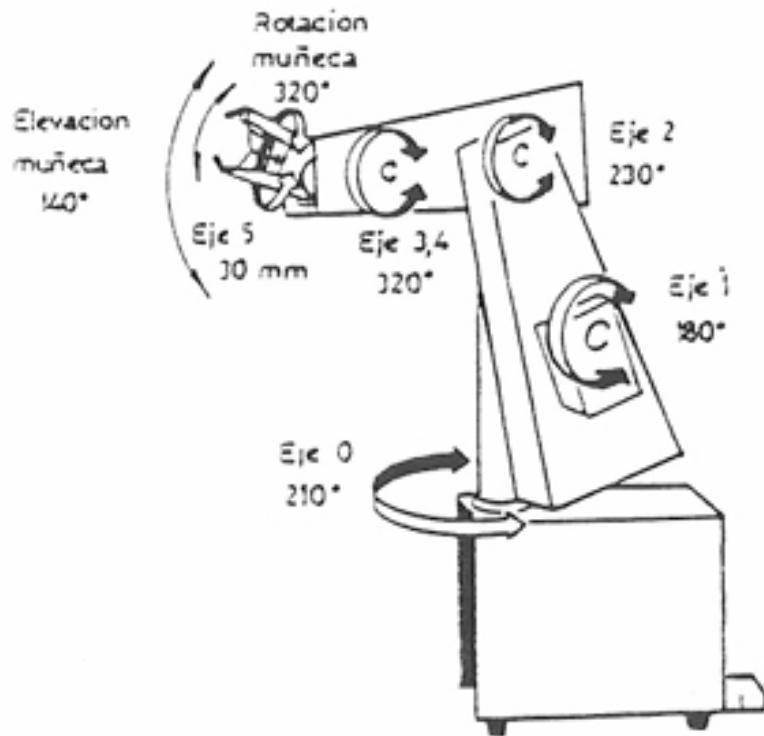
Es un robot de 5 GDL, los cuales están denotados: base, brazo 1, brazo 2, Pinza 1 y 2 y Grapper.

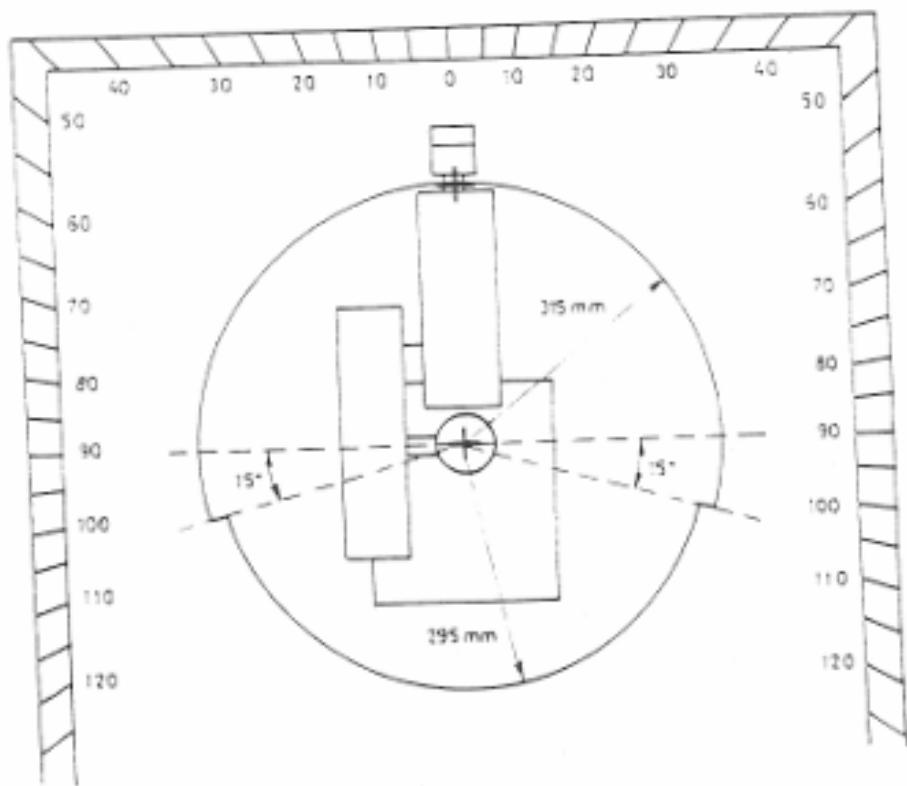
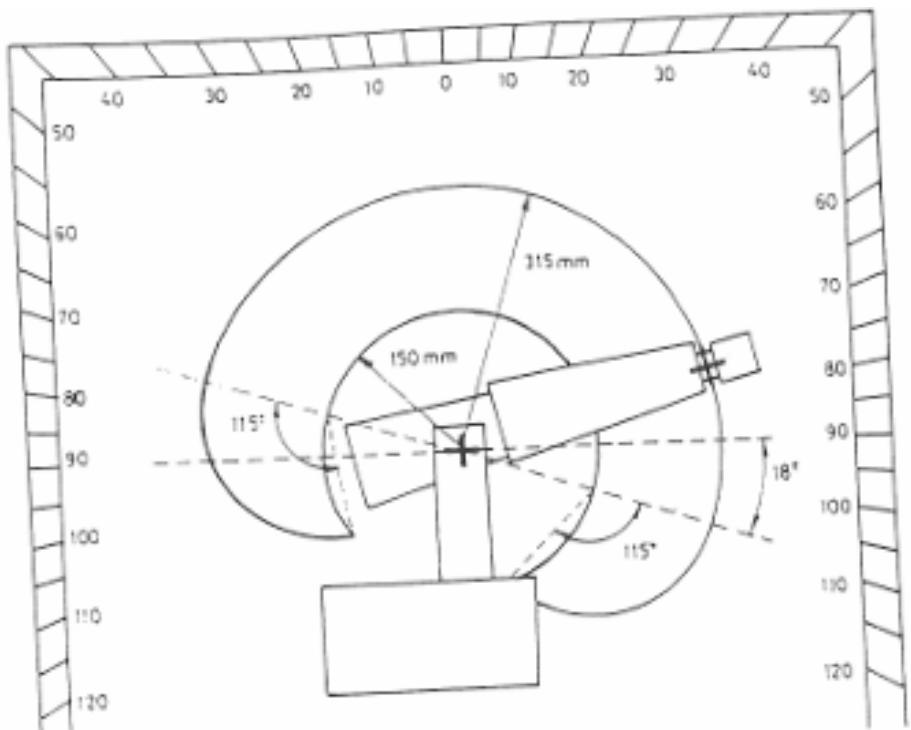
Cada articulación realiza movimientos rotatorios desarrollados por motores de 12v de corriente continua (CC), acoplados a una caja reductora y su posicionamiento se registra a través de encoders analógicos (potenciómetros).

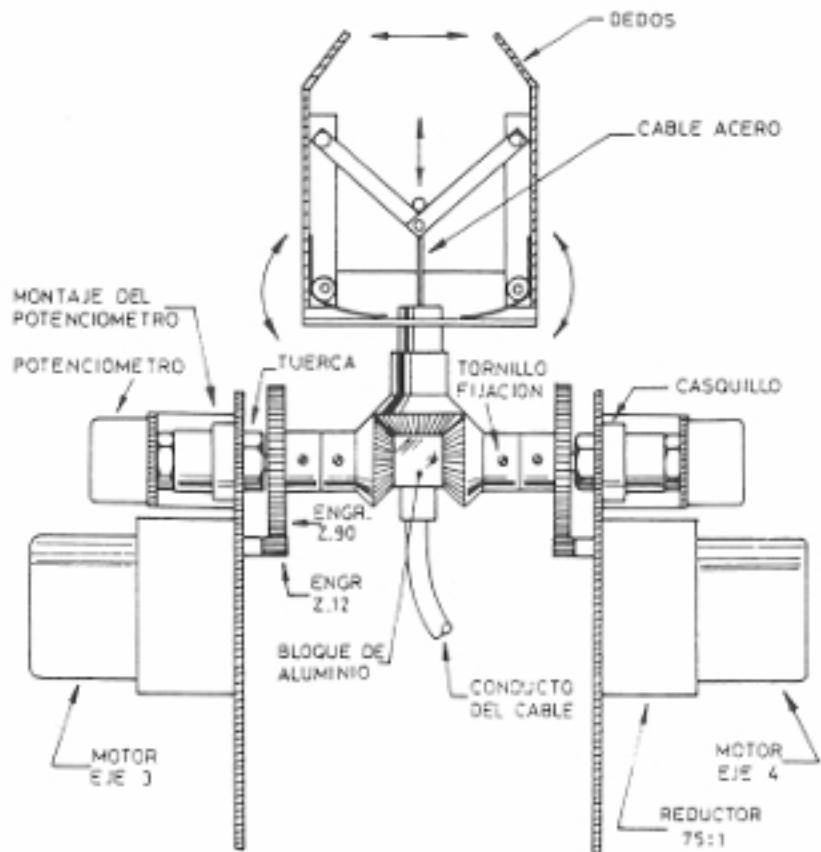
Los elementos que lo integran están construidos de acero, plástico y aluminio, siendo este ultimo el que constituye la mayor parte de su estructura.

Su capacidad máxima de carga es de 1 kg.

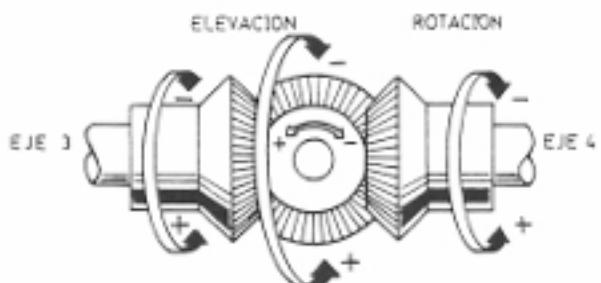
Se procede a adjuntar fotos representativas de las articulaciones:







SISTEMA DE GOBIERNO DE LA MUÑECA Y PINZA



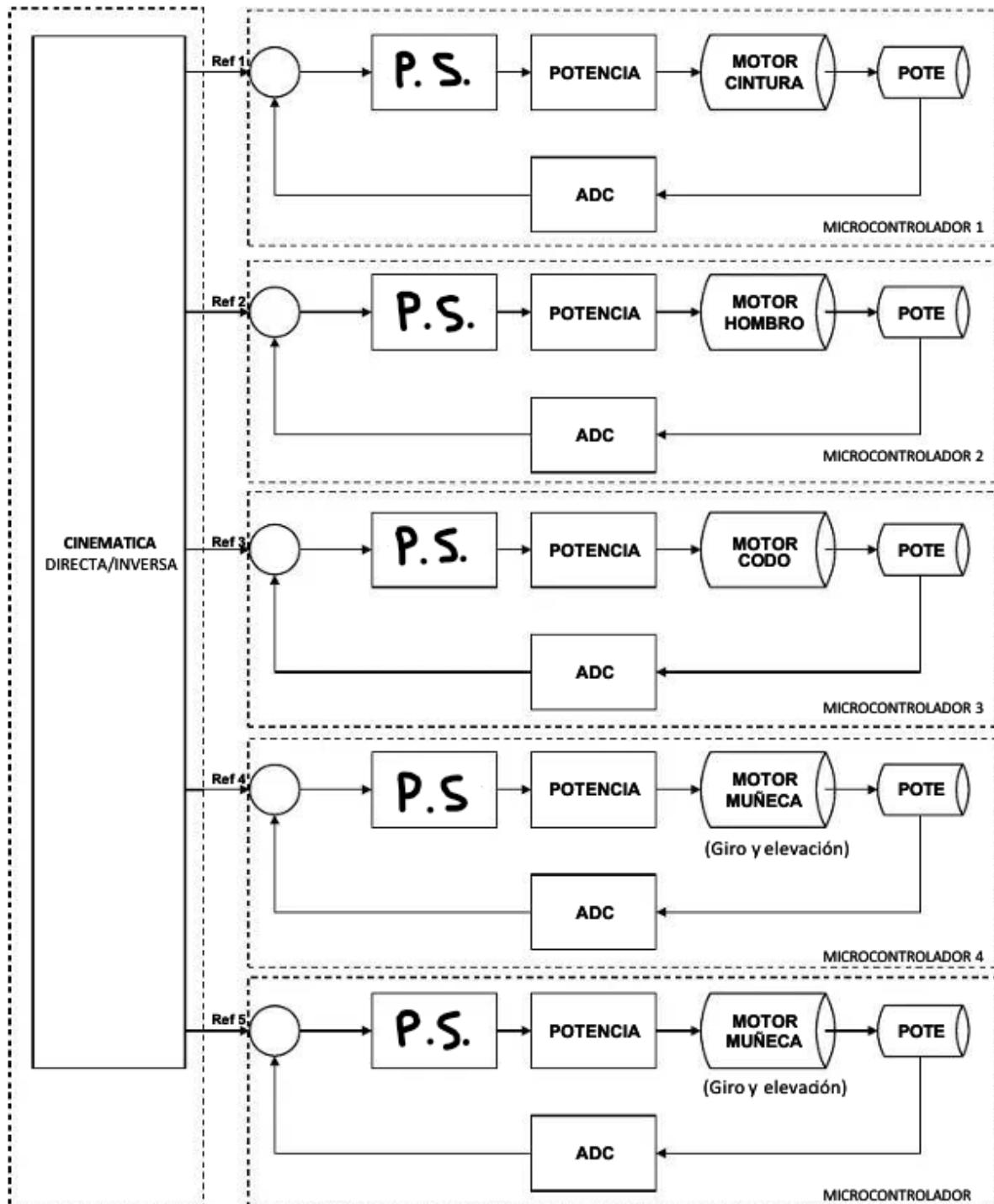
EJE 3	EJE 4	E : R
0	-1	$-\frac{1}{2}$ $+\frac{1}{2}$
0	+1	$+\frac{1}{2}$ $-\frac{1}{2}$
-1	0	$-\frac{1}{2}$ $-\frac{1}{2}$
-1	-1	-1
-1	+1	0
+1	0	$+\frac{1}{2}$ $+\frac{1}{2}$
+1	-1	0
+1	+1	1

MOVIMIENTO DE ELEVACION Y ROTACION  
DE LA MUÑECA

## 2. SISTEMA DE CONTROL

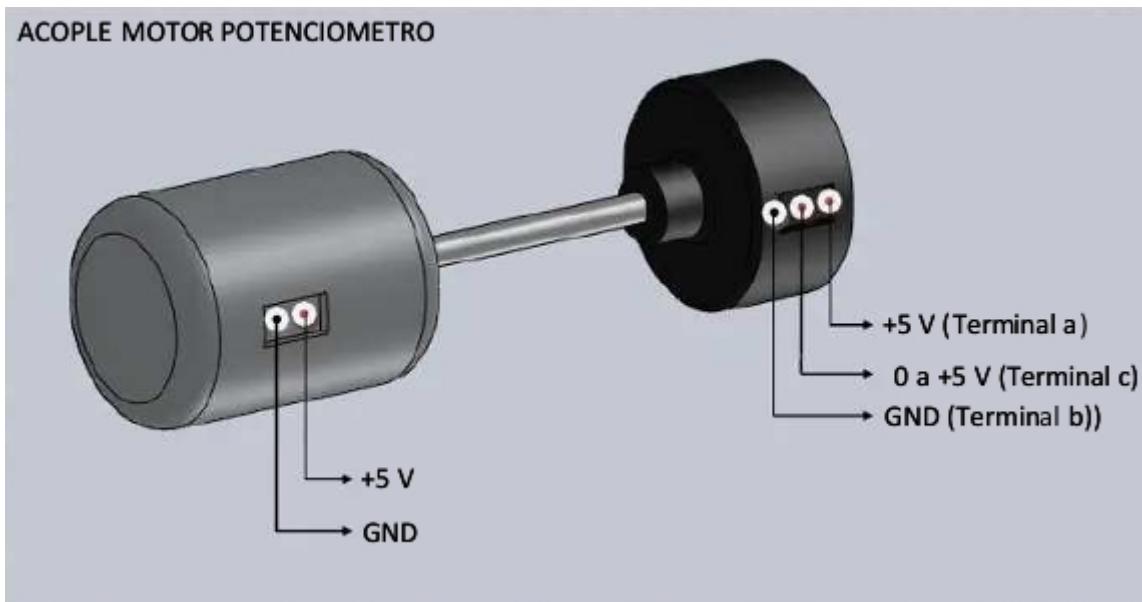
Para el control de posición de cada articulación del robot mentor, se recurre al diseño y construcción de una imitación muy simplificada de un controlador PID. No hemos recurrido a un control PID, dada la complejidad que ello conlleva.

A continuación se adjunta el diseño de diagrama en bloque general del proyecto.



## Elementos

- Motor de CC de imán permanente de 12v, 500mA y 500 r.p.m. Los motores de CC de imán permanente comerciales no giran a la misma velocidad en sentido dextrógiro que en sentido levógiro por lo que el controlador no tendrá la misma respuesta en ambos sentidos.
- Potenciómetro lineal de  $5K\Omega$ , de giro continuo, es decir, sin límites.
- Acople mecánico entre el eje del motor y el eje del potenciómetro.
- Fuente de 12v de CC para alimentar los terminales fijos del motor y fuente de 5v de CC para alimentar los terminales del potenciómetro.



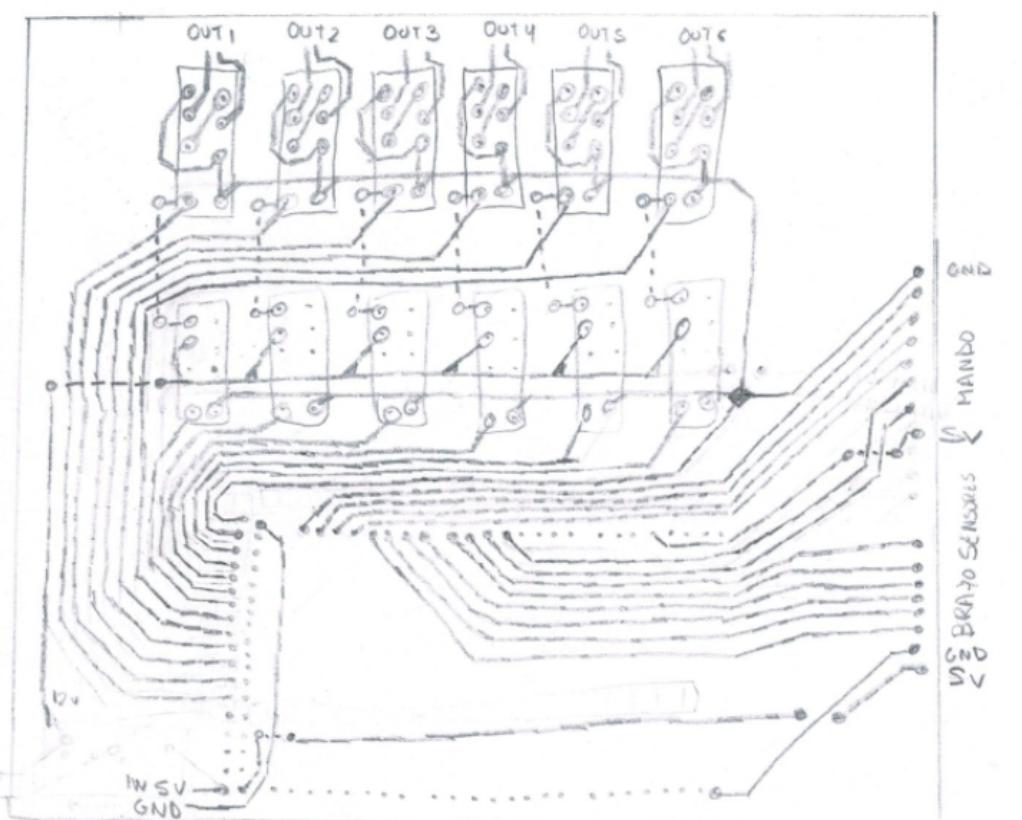
*Error en la foto, debería de aparecer, 12v de alimentación del motor*

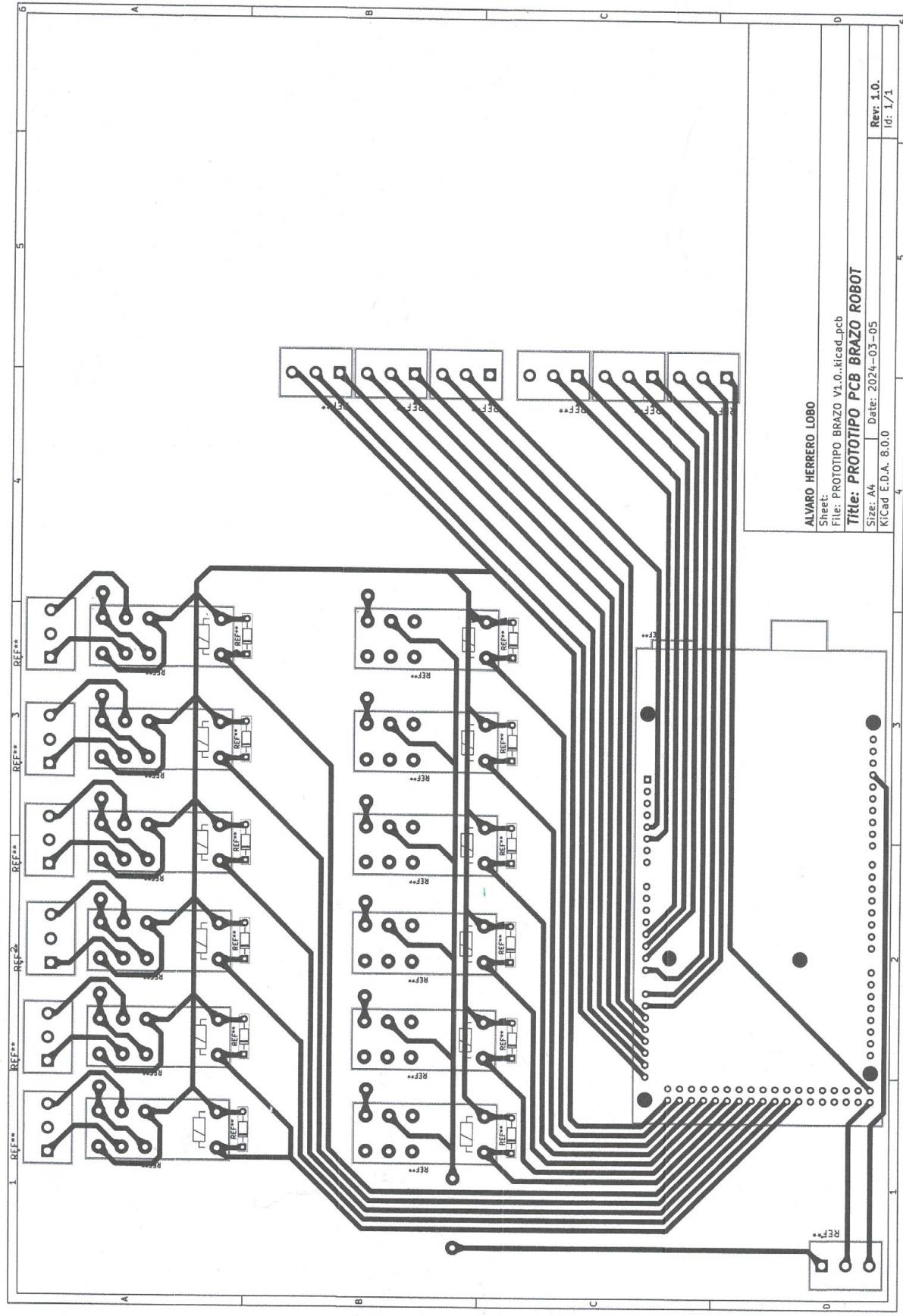
### 3. DISEÑOS INICIALES DE LA PCB

Para controlar los motores se requieren mínimo 500mA, pero el Arduino solo es capaz de alimentar 25mA, por ello se recurre a los CI ULN2803 conectado a los 12 relés de control, dos por cada motor, uno de activación/desactivación y otro de control de dirección.

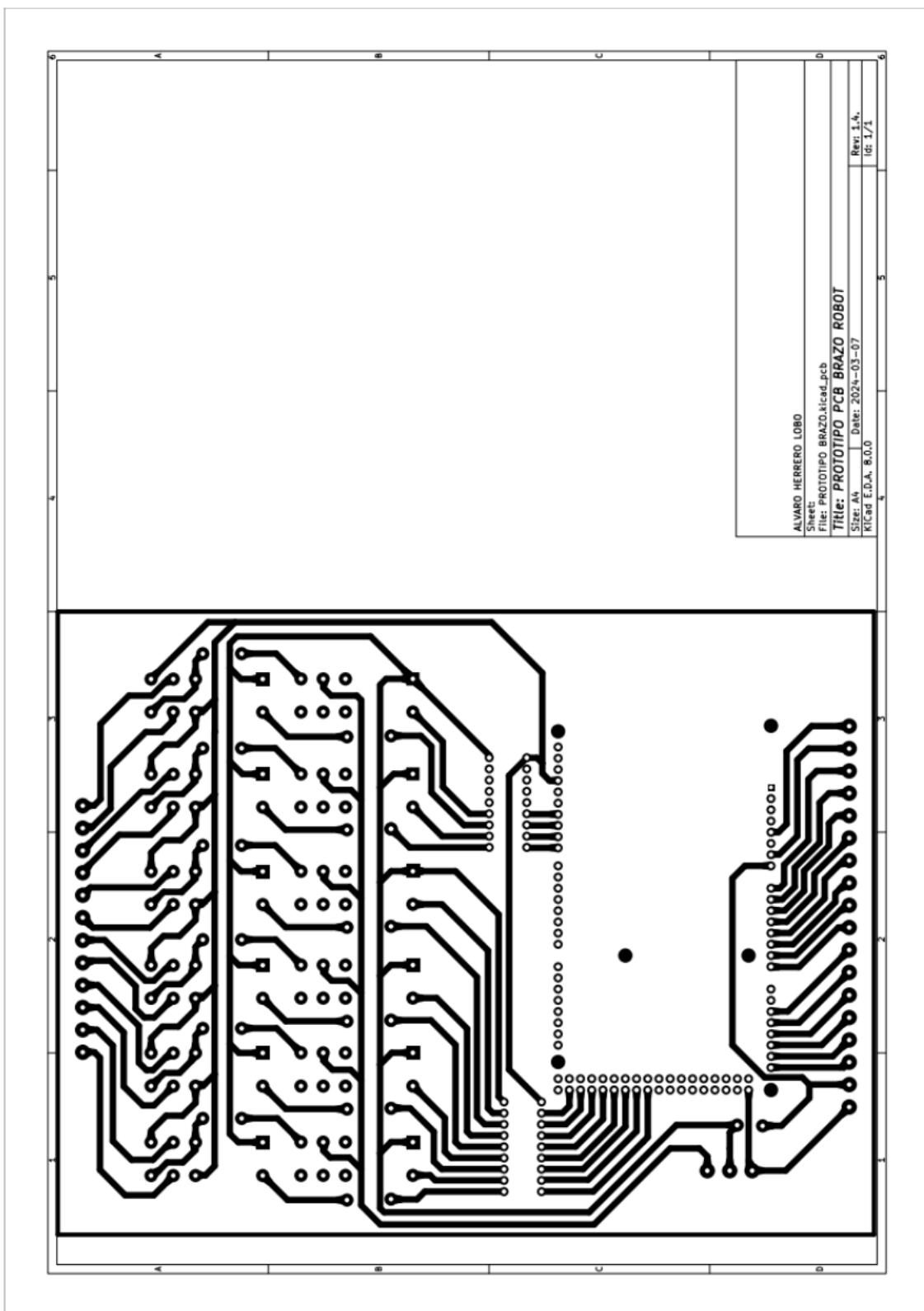
La PCB cuenta con tres bornes de alimentación, 5v/12v/GND, 6 bornes de salida para los motores y 18 bornes de entrada para los sensores.

Se adjuntan diseños previos al final:





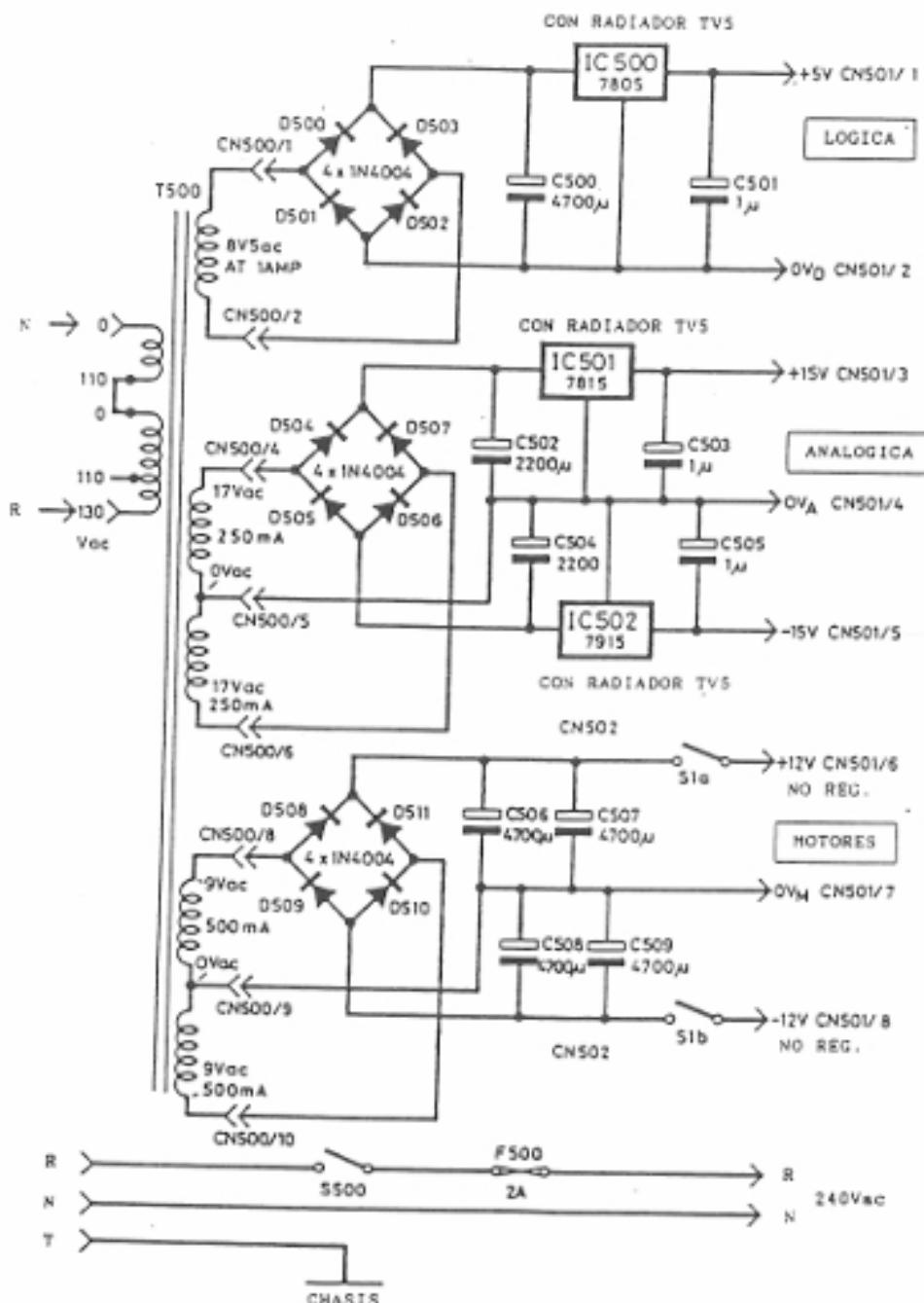
## DISEÑO FINAL PCB:



## 4. ELECTRÓNICA

### 4.1. Fuente de alimentación.

Hemos sustituido la fuente de alimentación original por una fuente de alimentación de PC, hemos utilizado solo las salidas de GND, 12v (potencia) y 5v (lógica). A falta del esquemático eléctrico de la fuente nueva, se adjunta el de la fuente original.



## 4.2. Relé

Para el control final de los motores se han usado 12 relés dobles, 2 por cada motor, uno de activación/desactivación y otro de sentido de giro. Se recomienda añadir diodos de protección en paralelo a los relés.

Se adjunta la ficha técnica del relé:



## PCB Relay

**G2R**

**A Power Relay for a Variety of Purposes with Various Models**

- Conforms to VDE0435 (VDE approval: C250 insulation grade), UL508, CSA22.2, SEV, SEMKO.
- Meets VDE0700 requirements for household products according to VDE0110.
- Clearance and creepage distance: 8 mm/8 m.
- Models with CTI250 material available.
- High-sensitivity (360 mW) and high-capacity (16 A) types available.
- Double-winding latching type available.
- Plug-in with test button and quick-connect terminals available.
- Highly functional socket available.











### Ordering Information

Classification		Enclosure ratings	Coil ratings	Contact form			
				SPST-NO	SPDT	DPST-NO	DPDT
PCB terminal	General-purpose	Flux protection	AC/DC	G2R-1A	G2R-1	G2R-2A	G2R-2
		Fully sealed		G2R-1A4	G2R-14	G2R-2A4	G2R-24
	Bifurcated contact	Flux protection	DC	G2R-1AZ	G2R-1Z	---	---
		Fully sealed		G2R-1AZ4	G2R-1Z4	---	---
	High-capacity	Flux protection	AC/DC	G2R-1A-E	G2R-1-E	---	---
	High-sensitivity	Flux protection		G2R-1A-H	G2R-1-H	G2R-2A-H	G2R-2-H
	Double-winding latching	Flux protection	Unsealed AC/DC DC AC/DC	G2RK-1A	G2RK-1	G2RK-2A	G2RK-2
	General-purpose	---		---	G2R-1-S	---	G2R-2-S
	LED indicator	---		---	G2R-1-SN	---	G2R-2-SN
	LED indicator with test button	---		---	G2R-1-SNI	---	G2R-2-SNI
	Diode	---		---	G2R-1-SD	---	G2R-2-SD
	LED indicator and diode	---		---	G2R-1-SND	---	G2R-2-SND
	LED indicator and diode with test button	---		---	G2R-1-SNDI	---	G2R-2-SNDI
	General-purpose	AC/DC		G2R-1A3-S	G2R-13-S	---	---
	LED indicator	G2R-1A3-SN		G2R-13-SN	---	---	---
	LED indicator and diode	DC		G2R-1A3-SND	G2R-13-SND	---	---

**Note:** 1. When ordering, add the rated coil voltage to the model number.

Example: G2R-1A 12 VDC

Rated coil voltage

2. OMRON has also prepared the above relays with AgSnn contacts, which are more tolerant of large inrush currents and physical movement compared with relays with standard contacts. When ordering, add "-ASI" to the model number.

Example: G2R-1A-ASI

3. Standard, NO contact type relays are TV-3 class products in accordance with the TV standards of the UL/CSA. Models with AgSnn contacts are TV-5 class products.

Example: G2R-1A-ASI

When ordering a TV-8 class model, insert "-TV8" into the model number as follows:

Example: G2R-1A-TV8-ASI

4. Models with CTI250 material are also available.

Contact your OMRON representative for more details.

## Specifications

### ■ Coil Ratings

Rated voltage		12 VAC	24 VAC	100' (110) VAC	120 VAC	200' (220) VAC	220 VAC	230 VAC	240 VAC
Rated current (H) (ref. value)	50 Hz	93 mA	46.5 mA	11 mA	9.3 mA	5.5 (4.0) mA	5.1 mA	4.7 (3.7) mA	4.7 mA
	60 Hz	75 mA	37.5 mA	9 (10.6) mA	7.5 mA	4.5 (5.3) mA	4.1 mA	3.8 (3.1) mA	3.8 mA
Coil resistance		65 Ω	260 Ω	4,800 Ω	6,500 Ω	20,200 (25,000) Ω	25,000 Ω	26,850 (30,000) Ω	30,000 Ω
Coil inductance (H) (ref. value)	Armature OFF	0.19	0.81	13.34	21	51.3	57.5	62	65.5
	Armature ON	0.39	1.55	26.84	42	102	117	124	131
Must operate voltage		80% max. of rated voltage							
Must release voltage		30% min. of rated voltage							
Max. voltage		140% of rated voltage (at 23°C)							
Power consumption		Approx. 0.9 VA at 60 Hz (approx. 0.7 VA at 60 Hz)							

- Note:
1. Rated voltage of bifurcated crossbar contact type: 100(110) VAC, 200(220) VAC, 230 VAC (Approx. 0.7 VA at 60 Hz)
  2. Depending on the type of Relay, some Relays do not have coil specifications. Contact your OMRON representative for more details.

Rated voltage		5 VDC	6 VDC	12 VDC	24 VDC	48 VDC	100 VDC		
Rated current (50/60 Hz) (see note 1.)		106 mA	88.2 mA	43.6 mA	21.8 mA	11.5 mA	5.3 mA		
Coil resistance		47 Ω	68 Ω	275 Ω	1,100 Ω	4,170 Ω	18,860 Ω		
Coil inductance (H) (ref. value)	Armature OFF	0.20	0.28	1.15	4.27	13.86	67.2		
	Armature ON	0.39	0.55	2.29	8.55	27.71	93.2		
Must operate voltage		70% max. of rated voltage							
Must release voltage		15% min. of rated voltage							
Max. voltage		170% of rated voltage (at 23°C)							
Power consumption		Approx. 0.53 W							

Note: Rated voltage of bifurcated crossbar contact type: 12 VDC, 24 VDC

### High-sensitivity Relays

Rated voltage		5 VDC	6 VDC	12 VDC	24 VDC	48 VDC			
Rated current (50/60 Hz) (see note 1.)		71.4 mA	60 mA	30 mA	15 mA	7.5 mA			
Coil resistance (see note 1.)		70 Ω	100 Ω	400 Ω	1,600 Ω	6,400 Ω			
Coil inductance (H) (ref. value)	Armature OFF	0.37	0.53	2.14	7.80	31.20			
	Armature ON	0.75	1.07	4.27	15.60	62.40			
Must operate voltage		70% max. of rated voltage							
Must release voltage		15% min. of rated voltage							
Max. voltage		170% of rated voltage (at 23°C)							
Power consumption		Approx. 0.36 W							

- Note:
1. The rated current and coil resistance are measured at a coil temperature of 23°C with a tolerance of +15%/-20% (AC rated current) or ±10% (DC coil resistance).
  2. LEDs are used for the built-in operation indicator. For models equipped with these indications, the VAC rated current must be increased by approximately 1 mA; the VDC rated current, by approximately 4 mA.
  3. Operating characteristics are measured at a coil temperature of 23°C.

### 4.3. ULN2803

El integrado ULN2803 es un array de transistores Darlington que se utiliza para controlar cargas de alta corriente y voltaje a partir de señales de salida de baja potencia, como las de un microcontrolador.

En nuestro proyecto lo hemos utilizado para aumentar la intensidad de salida del microprocesador, para poder alimentar los relés.

Se adjunta ficha técnica del integrado:


**MOTOROLA**

Order this document by ULN2803/D

<b>ULN2803</b>	<b>ULN2804</b>
----------------	----------------

**Octal High Voltage,  
High Current Darlington  
Transistor Arrays**

The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open-collector outputs and free wheeling clamp diodes for transient suppression.

The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.

**OCTAL PERIPHERAL  
DRIVER ARRAYS**

**SEMICONDUCTOR  
TECHNICAL DATA**

**MAXIMUM RATINGS** ( $T_A = 25^\circ\text{C}$  and rating apply to any one device in the package, unless otherwise noted.)

Rating	Symbol	Value	Unit
Output Voltage	$V_O$	50	V
Input Voltage (Except ULN2801)	$V_I$	30	V
Collector Current – Continuous	$I_C$	500	mA
Base Current – Continuous	$I_B$	25	mA
Operating Ambient Temperature Range	$T_A$	0 to +70	$^\circ\text{C}$
Storage Temperature Range	$T_{stg}$	-55 to +150	$^\circ\text{C}$
Junction Temperature	$T_J$	125	$^\circ\text{C}$

$R_{\theta JA} = 55^\circ\text{C}/\text{W}$   
Do not exceed maximum current limit per driver.

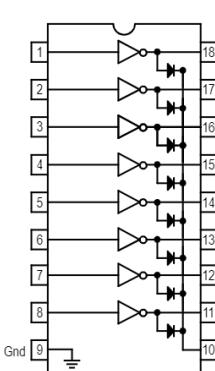


**A SUFFIX**  
PLASTIC PACKAGE  
CASE 707

**ORDERING INFORMATION**

Device	Characteristics		
	Input Compatibility	$V_{CE}(\text{Max})/I_C(\text{Max})$	Operating Temperature Range
ULN2803A	TTL, 5.0 V CMOS	50 V/500 mA	$T_A = 0 \text{ to } +70^\circ\text{C}$
ULN2804A	6 to 15 V CMOS, PMOS		

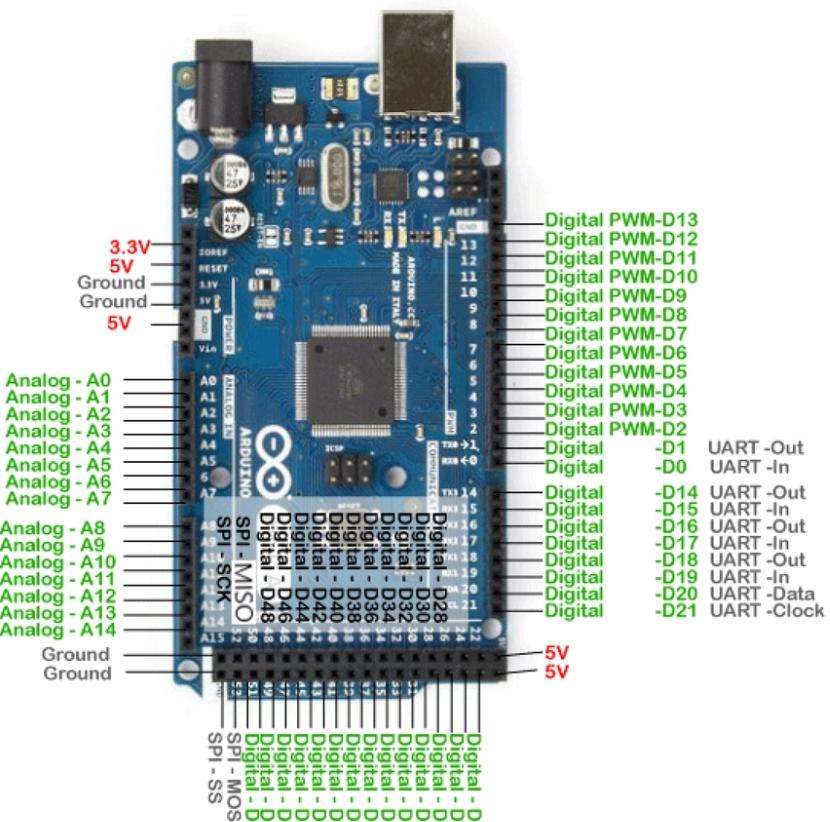
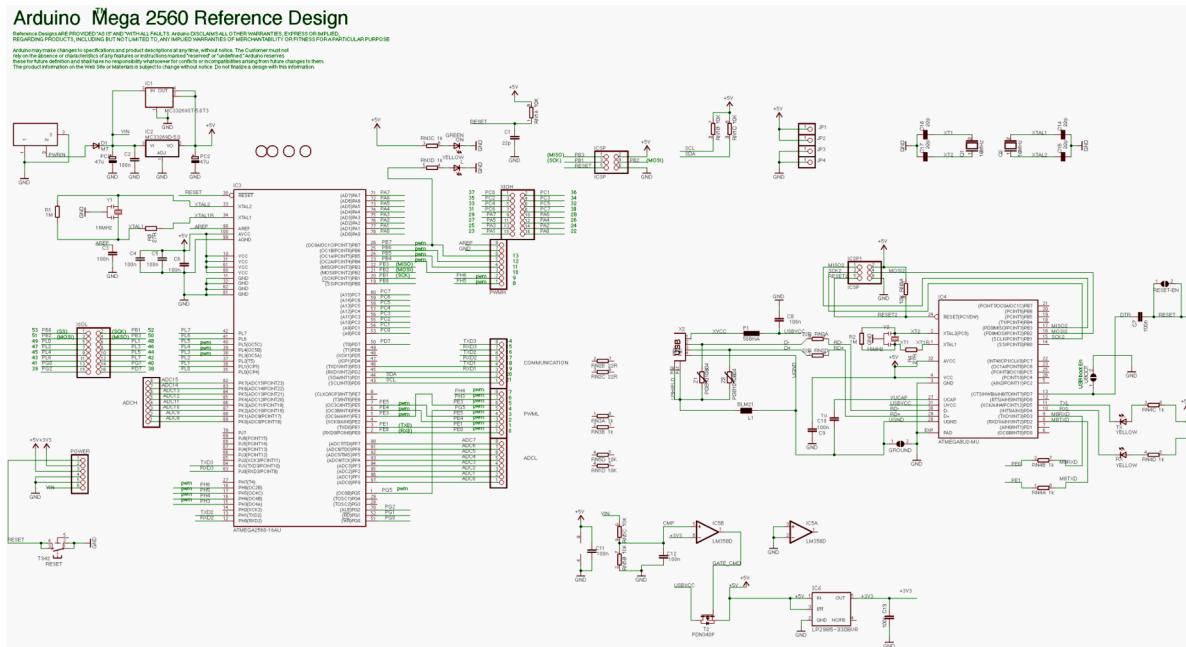
**PIN CONNECTIONS**



## 4.4. Arduino Mega 2560

El Arduino Mega es una placa de desarrollo basada en el microcontrolador ATmega2560, que ofrece un mayor número de entradas y salidas digitales y analógicas en comparación con otras placas Arduino, lo que lo hace ideal para proyectos como el nuestro.

Se adjunta esquemático:



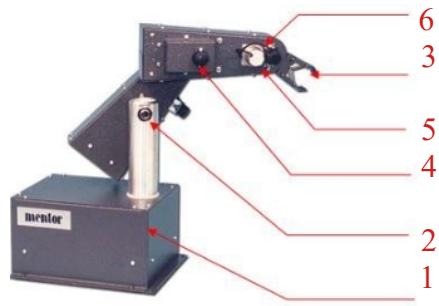
## 5. CONEXIONES

Para realizar el proyecto hemos utilizado 25 pines del microprocesador, 12 salidas digitales y 13 entradas analógicas.

### Entradas analógicas

#### **BRAZO**

- A0** - Potenciómetro 1
- A2** - Potenciómetro 3
- A3** - Potenciómetro 2
- A4** - Potenciómetro 6
- A5** - Potenciómetro 5
- A8** - Potenciómetro 4
- A15** - Interruptor general

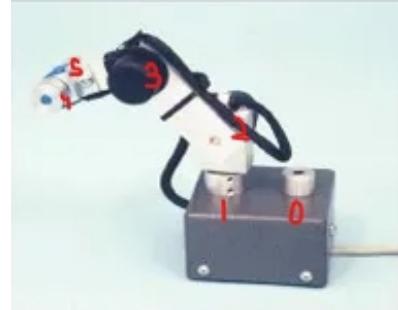


#### **MANDO**

- A6** - Potenciómetro 1
- A10** - Potenciómetro 4
- A11** - Potenciómetro 2
- A12** - Potenciómetro 3
- A13** - Potenciómetro 5

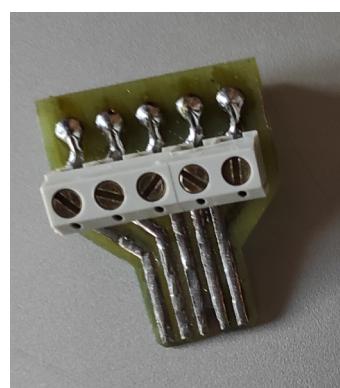
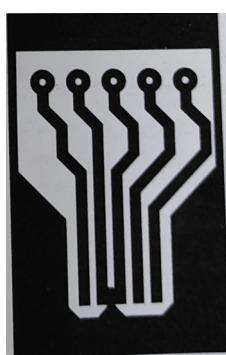
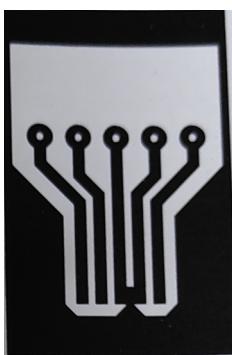
### Salidas digitales

- 8** – inversión grapper
- 9** – activación grapper
- 10** – inversión pinza 1
- 11** – activación pinza 1
- 12** – inversión base
- 23** – activación pinza 2
- 25** – inversión pinza 2
- 27** – activación brazo 1
- 29** – inversión brazo 1
- 31** – activación brazo 2
- 33** – inversión brazo 2
- 35** – activación base



### **5.1. Mando**

Para permitir la conexión entre el mando y la PCB se procede a hacer un conector de 10 pines que se une al conector original del mando.



## 6. SOFTWARE

Se adjunta el codigo en C utilizado para el control del brazo robot:

```
void setup() {
    Serial.begin(9600);
    pinMode(12, OUTPUT); //inversion giro bas
    pinMode(35, OUTPUT); //activacion base

    pinMode(8, OUTPUT); //inversion clapper
    pinMode(9, OUTPUT); //activacion clapper
    pinMode(10, OUTPUT); //inversion pinza
    pinMode(11, OUTPUT); //activacion pinza
    pinMode(23, OUTPUT); //activacion pinza
    pinMode(25, OUTPUT); //inversion pinza
    pinMode(31, OUTPUT); //activacion b2
    pinMode(33, OUTPUT); //inversion giro b2
    pinMode(27, OUTPUT); //activacion b1
    pinMode(29, OUTPUT); //inversion giro b1
}

void loop() {
    int MC = analogRead(A15);
    Serial.println("----- DATOS BRAZO ROBOT -----");
    Serial.print("MOTOR CURRENT: ");
    Serial.println(MC);

    //PROCESAMIENTO DATOS ROBOT
    int posBase = analogRead(A0);
    int GBase = 210 - map(posBase, 0, 677, 0, 210);
    Serial.print("POSICION ROBOT BASE: ");
    Serial.println(GBase);

    int posPinza1 = analogRead(A4);
    int GposPinza1 = map(posPinza1, 477, 866, 0, 140);
    Serial.print("POSICION ROBOT PINZA 1: ");
    Serial.println(GposPinza1);

    int posPinza2 = analogRead(A5);
    int GposPinza2 = map(posPinza2, 376, 777, 0, 140);
    Serial.print("POSICION ROBOT PINZA 2: ");
    Serial.println(GposPinza2);

    int posPinzaClap = analogRead(A2);
    int GposPinzaClap = map(posPinzaClap, 758, 577, 0, 100);
    Serial.print("POSICION ROBOT PINZA CLAP: ");
    Serial.println(GposPinzaClap);

    int posB2 = analogRead(A3);
    int GB2 = map(posB2, 100, 795, 0, 230);
```

```

Serial.print("POSICION ROBOT B2: ");
Serial.println(GB2);

int posB1 = analogRead(A8);
int GB1 = map(posB1, 65, 650, 0, 180);
Serial.print("POSICION ROBOT B1: ");
Serial.println(GB1);

//PROCESAMIENTO DATOS MANDO
int posMBase = analogRead(A6);
int GMBase = map(posMBase, 0, 1023, 0, 210);
Serial.print("POSICION MANDO BASE: ");
Serial.println(GMBase);

int posPinzaIncl = analogRead(A13);
int GposPinzaIncl = map(posPinzaIncl, 0, 1023, 0, 140);
Serial.print("POSICION MANDO INCL PINZA: ");
Serial.println(GposPinzaIncl);

int posPinzaClapper = analogRead(A10);
int GposPinzaClapper = map(posPinzaClapper, 0, 1023, 0, 100);
Serial.print("POSICION MANDO PINZA CLAPPER: ");
Serial.println(GposPinzaClapper);

int posMB2 = analogRead(A11);
int GMB2 = map(posMB2, 230, 1000, 0, 230) - 5; // corrección de 5 grados
Serial.print("POSICION MANDO B2: ");
Serial.println(GMB2);

int posMB1 = analogRead(A12);
int GMB1 = map(posMB1, 367, 1023, 0, 180);
Serial.print("POSICION MANDO B1: ");
Serial.println(posMB1);

//MOVIMIENTO

if (MC < 1023) {
    //MOVIMIENTO BASE
    if (GMBase > GBase && GBase < 200) {
        if((GBase + 10) > GMBase){
            digitalWrite(12, LOW);
            digitalWrite(35, HIGH);
            delay(40);
            digitalWrite(12, LOW);
            digitalWrite(35, LOW);
            delay(40);
        }
    } else{
        digitalWrite(12, LOW);
        digitalWrite(35, HIGH);
    }
}

```

```

        }
    }
else if (GMBase < GBase && GBase > 10) {
    if((GBase - 10) < GMBase){
        digitalWrite(12, HIGH);
        digitalWrite(35, HIGH);
        delay(40);
        digitalWrite(12, LOW);
        digitalWrite(35, LOW);
        delay(40);
    }
    else{
        digitalWrite(12, HIGH);
        digitalWrite(35, HIGH);
    }
}
else {
    digitalWrite(12, LOW);
    digitalWrite(35, LOW);
}

//MOVIMIENTO PINZA
if (GposPinzaIncl >= GposPinza1 && GposPinzaIncl >= GposPinza2 && GposPinza1 < 110) {
    if ((GposPinza1 + 25) > GposPinzaIncl){
        digitalWrite(10, HIGH);
        digitalWrite(11, HIGH);
        digitalWrite(23, HIGH);
        digitalWrite(25, HIGH);
        delay(40);
        digitalWrite(10, LOW);
        digitalWrite(11, LOW);
        digitalWrite(23, LOW);
        digitalWrite(25, LOW);
        delay(40);
    }
    else{
        digitalWrite(10, HIGH);
        digitalWrite(11, HIGH);
        digitalWrite(23, HIGH);
        digitalWrite(25, HIGH);
    }
}
else if (GposPinzaIncl <= GposPinza1 && GposPinzaIncl <= GposPinza2 && GposPinza2 > 25)
{
    if ((GposPinza1 - 25) < GposPinzaIncl){
        digitalWrite(10, LOW);
        digitalWrite(11, HIGH);
        digitalWrite(23, HIGH);
        digitalWrite(25, LOW);
        delay(40);
    }
}

```

```

digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(23, LOW);
digitalWrite(25, LOW);
delay(40);
}
else{
digitalWrite(10, LOW);
digitalWrite(11, HIGH);
digitalWrite(23, HIGH);
digitalWrite(25, LOW);
}
}
else {
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(23, LOW);
digitalWrite(25, LOW);
}
}

if (GposPinzaClapper >= GposPinzaClap && GposPinzaClap < 70) {
if ((GposPinzaClapper + 10) > GposPinzaClap) {
digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
delay(100);
digitalWrite(8, LOW);
digitalWrite(9, LOW);
delay(100);
}
else {
digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
}
}
}

else if (GposPinzaClapper <= GposPinzaClap && GposPinzaClap > 50) {
digitalWrite(8, LOW);
digitalWrite(9, HIGH);
}
else {
digitalWrite(8, LOW);
digitalWrite(9, LOW);
}

//MOVIMIENTO BRAZOS
if (GMB2 > GB2 && GB2 < 200) {
if ((GB2 + 20) > GMB2) {
digitalWrite(33, LOW);
digitalWrite(31, HIGH);
delay(30);
digitalWrite(33, LOW);
digitalWrite(31, LOW);
}
}

```

```

        delay(30);
    }
    else {
        digitalWrite(33, LOW);
        digitalWrite(31, HIGH);
    }
}
else if (GMB2 < GB2 && GB2 > 20) {
    if ((GB2 - 20) < GMB2) {
        digitalWrite(33, HIGH);
        digitalWrite(31, HIGH);
        delay(30);
        digitalWrite(33, LOW);
        digitalWrite(31, LOW);
        delay(30);
    }
    else {
        digitalWrite(33, HIGH);
        digitalWrite(31, HIGH);
    }
}
else {
    digitalWrite(33, LOW);
    digitalWrite(31, LOW);
}

if (GMB1 > GB1 && GB1 < 170) {
    if ((GB1 + 10) > GMB1) {
        digitalWrite(29, HIGH);
        digitalWrite(27, HIGH);
        delay(30);
        digitalWrite(29, LOW);
        digitalWrite(27, LOW);
        delay(30);
    }
    else {
        digitalWrite(29, HIGH);
        digitalWrite(27, HIGH);
    }
}
else if (GMB1 < GB1 && GB1 > 10) {
    if ((GB1 - 10) < GMB1) {
        digitalWrite(29, LOW);
        digitalWrite(27, HIGH);
        delay(30);
        digitalWrite(29, LOW);
        digitalWrite(27, LOW);
        delay(30);
    }
    else {

```

```
    digitalWrite(29, LOW);
    digitalWrite(27, HIGH);
}
}
else {
    digitalWrite(29, LOW);
    digitalWrite(27, LOW);
}
}
else {
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(23, LOW);
    digitalWrite(25, LOW);
    digitalWrite(31, LOW);
    digitalWrite(33, LOW);
    digitalWrite(27, LOW);
    digitalWrite(29, LOW);
}
}
```

## 7. CONCLUSIÓN

Este proyecto se propuso desarrollar una adaptación de un brazo robótico para poder controlarlo mediante un Arduino Mega, con el objetivo de demostrar la viabilidad de utilizar plataformas de bajo costo para la automatización de tareas simples. El brazo robótico fue diseñado para manejar objetos de hasta 1kg y alcanzar un radio de operación de 40 centímetros aproximadamente.

A lo largo del desarrollo, se lograron cumplir todas las metas establecidas inicialmente. El brazo es capaz de realizar movimientos más o menos precisos, controlados por el mando, lo que demuestra la efectividad del Arduino Mega en la gestión de múltiples motores y sensores simultáneamente. Uno de los desafíos más significativos fue calibrar correctamente los potenciómetros para asegurar la precisión de los movimientos. La solución fue implementar un sistema de retroalimentación que ajusta los ángulos de los motores en tiempo real.

De este proyecto, hemos aprendido la importancia de una planificación detallada y de las pruebas iterativas en el diseño de sistemas robóticos. La capacidad de adaptación y la resolución de problemas fueron habilidades esenciales que se fortalecieron durante la ejecución del proyecto.

Para futuras iteraciones, recomendamos explorar el uso de sensores adicionales, como sensores de proximidad o de fuerza, para expandir las capacidades del brazo robótico, ampliación bluetooth o añadir una tarjeta SD para poder guardar movimientos repetitivos, permitiendo una mayor interacción con el entorno. Asimismo, sería beneficioso investigar materiales más ligeros y resistentes para mejorar la eficiencia y la capacidad de carga del brazo.

En conclusión, este proyecto no solo ha demostrado que es posible implementar soluciones robóticas efectivas y económicas con Arduino, sino que también ha proporcionado una base sólida para futuras investigaciones y desarrollos en el campo de la robótica educativa y de hobby.

