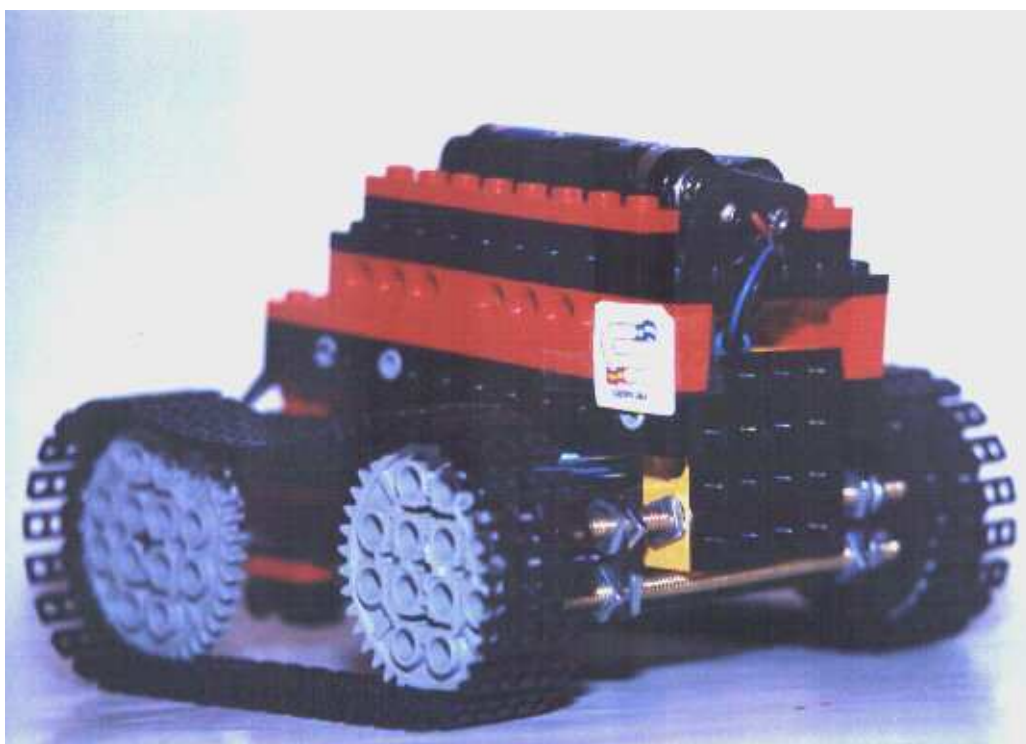


TARJETA CT293+

MANUAL DE USUARIO



INDICE

1.- INTRODUCCIÓN A LA CT293+

2.- DESCRIPCIÓN DE LOS ELEMENTOS DE LA CT293+

- 2.1.- Disposición y explicación de los componentes de la CT293+
- 2.2.- Listado de componentes
- 2.3.- Descripción de los buses de control

3.- INSTALACIÓN DE LOS SENSORES Y MOTORES

- 3.1.- Conexión de los motores de continua a la CT293+
- 3.2.- Conexión de un motor paso a paso a la CT293+
- 3.3.- Conexión de los sensores de infrarrojos a la CT293+
- 3.4.- Conexión con las entradas digitales
- 3.5.- Conexión con las entradas analógicas

4.- PROGRAMACIÓN DE LA TARJETA CT293+

- 4.1.- Programación del Bloque A: Motores y sensores de infrarrojos
 - 4.1.1.- Los sensores de infrarrojos
 - 4.1.2.- Los motores de continua
 - 4.1.3.- Un Microbot que sigue la línea negra
 - 4.1.4.- Programación avanzada de los sensores de infrarrojos
 - 4.1.5.- Programación de motores paso a paso
- 4.2.- Programación del Bloque E: Entradas digitales y analógicas
 - 4.2.1.- Ejemplo de utilización en modo digital
 - 4.2.2.- Ejemplo de utilización en modo analógico

5.- CONEXIÓN DE OTROS ELEMENTOS

APÉNDICE A: EL CHIP L293B

- A.1.- Sistema con mínimos recursos
- A.2.- Sistema intuitivo con recursos máximos

APÉNDICE B: EL SENSOR CNY70

- B.1.- Polarización en continua
- B.2.- Polarización modulada

APÉNDICE C: LOS MOTORES PASO A PASO

- C.1.- Características de los motores paso a paso
- C.2.- Control de motores paso a paso
- C.3.- Esquema de conexión de un motor paso a paso bipolar con el chip L293B
- C.4.- Esquema de conexión de un motor paso a paso unipolar con el chip SAA1027

APÉNDICE D: LISTADO DE LOS NEMONICOS DEL 68HC11

1. INTRODUCCIÓN A LA CT293+

La CT293+ es una nueva tarjeta que proporciona al sistema Tower la posibilidad de controlar motores y sensores. Esta diseñada para adaptarse perfectamente a la CT6811 y poder controlarla sin ninguna variación tanto en Bootstrap como en Single Chip. La tarjeta también puede controlarse desde otro sistema por ejemplo el puerto paralelo de un PC. Esta tarjeta es la versión mejorada de la CT293, pero se ha respetado la compatibilidad. Los programas de su antecesora valen también para la nueva, lo único que hay que verificar es la situación de los sensores y motores.

La CT293+ es el soporte principal de los dos primeros niveles de la torre BOT, con ella se proporciona el movimiento a los sistemas de control y la capacidad de analizar el entorno. Con esta placa y con la CT6811 se puede obtener la plataforma del microbot. Luego hay que programarlo para ir subiendo de nivel, hasta llegar al nivel INTELIGENTE, pues para el siguiente se necesita tener varios microbots.

Los prototipos Quark y Monumental desarrollados en enero de 1997 por el Grupo J&J implementan correctamente lo dicho arriba. El prototipo Quark es capaz de seguir una línea negra, pero también puede realizar recorridos que previamente grabados. Esto último se consigue gracias al sistema de encoders que lleva instalado. Monumental fue el ganador del primer concurso de microbots de sumo celebrado en la Universidad de Informática en la UAM.

Las características de la tarjeta CT293+ se pueden resumir en:

1. Posibilidad de control de dos motores de continua o uno paso_paso.
2. Capacidad para leer cuatro sensores de infrarrojos, pudiendo ser estos optoacopladores.
3. Disponibilidad de 8 entradas digitales de propósito general con la posibilidad de usarlas como entradas analógicas.
4. Conexión de los elementos externos con clemas.
5. Alimentación de los motores externa o interna.

En el resto de la memoria se describen los conocimientos que se consideran necesarios para sacar el máximo partido a esta tarjeta. También se introducen algunos conceptos sobre control de motores y polarización de sensores. Todo ello con programas de ejemplo debidamente comprobados.

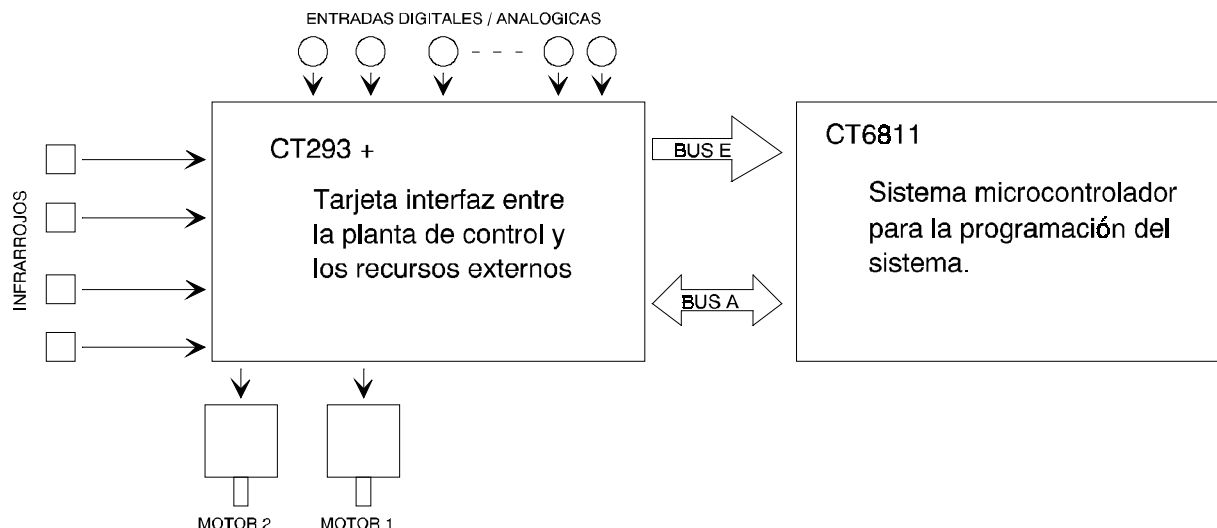


Figura 1. Utilización de la tarjeta CT293+ como soporte de un Microbot.

2. DESCRIPCIÓN DE LOS ELEMENTOS DE LA CT293+

2.1. DISPOSICIÓN Y EXPLICACIÓN DE LOS COMPONENTES DE LA CT293+.

En la figura 2.1 se puede apreciar la disposición de los componentes en la placa. Básicamente tiene dos bloques independientes. El primero de ellos (BLOQUE A) se encarga de los motores y de los sensores de infrarrojos, mientras que el segundo (BLOQUE E) controla las entradas digitales/analógicas. Hay un bus de control para cada bloque, llamados PUERTO A y PUERTO E. El usuario que este familiarizado con la CT6811 se dará cuenta del motivo de esta nomenclatura. El bus A (Puerto A) es el que maneja el bloque A, es decir motores e infrarrojos, el bus E (Puerto E) maneja el bloque E, que tiene las entradas analógico/digitales.

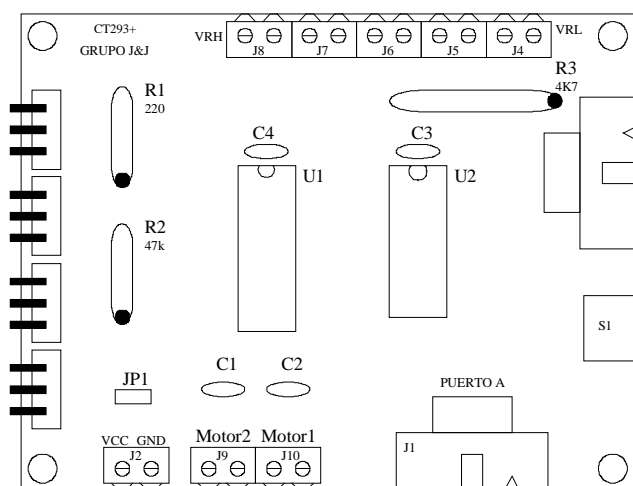


Figura 2.1. Situación de los componentes de la CT293+

El bloque A esta formado por el chip L293B¹ que gestiona la potencia de los motores y el chip 40106 que tiene dos funciones. La primera consiste en realizar una pequeña lógica para facilitar el uso de los motores, la segunda en realizar una conversión de niveles en la lectura de los infrarrojos. A este bloque también pertenecen los arrays de resistencias de 220 ohmios y 47 Kohmios que se usan para polarizar los infrarrojos.

El bloque E esta formado por un array de ocho resistencias de valor 4K7. Sirve de pull-up para las entradas digitales. Más adelante se explica esto con más detalle.

Descripción de los elementos de la CT293+:

- C1,C2 : Condensadores en paralelo con los motores. Sirven para eliminar ruido. El valor del condensador dependerá del motor utilizado. No poner condensadores electrolíticos. (por ejemplo 47nF cerámico)
- C3,C4 : Condensadores de eliminación de ruido para las pastillas integradas. (100 nF poliester)
- S1 : Array de cuatro switches. Acodado. Sirve para anular las entradas de los sensores.
- U2 : Pastilla 40106 c-mos inversora. Adapta niveles de entrada de los sensores.
- U1: driver de potencia L293B¹ para control de motores.
- PUERTO A, E: Conectores tipo bus acodado (5+5 líneas, Macho). Conexión al sistema de control.
- R1: Array de resistencias de 4+1, (valor 220 W), polarización de infrarrojos.
- R2: Array de resistencias de 4+1, (valor 47 KW), polarización de infrarrojos
- R3: Array de resistencias de 8+1 (valor 4K7 ohmios), pull_up de las entradas digitales.
- J2 : Clema doble para la alimentación motores.
- J4-J8 : Clemas dobles para las entradas digitales.
- motor1, motor2 : Clemas dobles donde se conectan los motores.
- JP1: *Jumper* para conexión interna de los motores.
- sensor1 - sensor4: Conexiones para los sensores de infrarrojos CNY70.

¹También se puede poner el L293D que es totalmente compatible.

2.2. LISTADO DE COMPONENTES.

Se indica aquí la referencia de cada componente de manera que si se deteriora alguno pueda ser sustituido por otro. Estos componentes se pueden adquirir en las tiendas de electrónica .

- C1, C2 : condensadores cerámicos de 47nF
- C3, C4 : condensadores de 100nF.
- Array de cuatro switches acodados. (Rojos con el ON abajo)
- pastilla 40106
- driver de potencia L293B¹
- Array de resistencias de 4+1 (valor 220 ohmios)
- Array de resistencias de 4+1 (valor 47K ohmios)
- Array de resistencias de 8+1 (valor 4K7 ohmios)
- 4 sensores de infrarrojos CNY70
- J2, J4-J8, motor1, motor2 : 8 Clemas dobles de alimentación para placa circuito impreso.
- zócalo pin torneado de 7+7 pines
- zócalo pin torneado de 8 +8 pines
- un *jumper*
- Dos conectores macho acodados de 5x2 pines para cable tipo bus.
- tira de pines macho-macho acodada
- cable plano tipo bus, de 10 hilos

2.3. DESCRIPCIÓN DE LOS BUSES DE CONTROL.

Como ya se ha dicho la tarjeta tiene dos bloques independientes, cada uno de ellos se conecta a la CT6811 mediante un bus de expansión. Hay veces en las cuales el usuario desea saber la disposición de los pines del bus ya sea por curiosidad, necesidad o simplemente porque quiera conectar la tarjeta a otro sistema de control. En este apartado se describen dichos buses.

También se indica en la figura 2.3. como construir el cable de unión entre la CT293+ y la CT6811 para que la asignación de pines entre los buses sea correcta. Esto es importante ya que si se construye mal el cable se puede derivar la alimentación a otros pines diferentes pudiendo incluso dañar algún elemento de la CT293+. La muesca que aparece en el conector hembra del bus coincide con la que hay en el conector macho. Al construir el cable de bus hay que situar los conectores hembra con las muescas apuntando en la misma dirección, (viendo el cable verticalmente) . Si se ve en horizontal que es el caso de la figura , se observa que las muescas se sitúan inversamente, si una mira hacia arriba la otra lo hace hacia abajo.

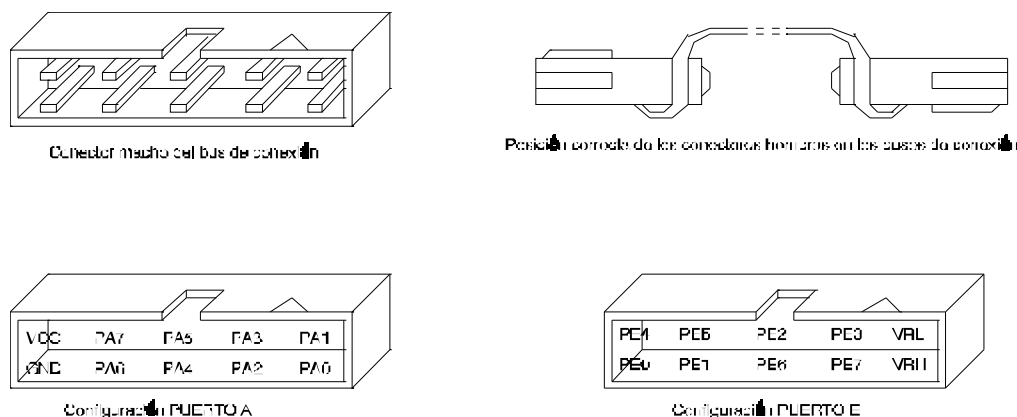


Figura 2.3.1 Descripción de los buses de control de la CT293+.

PUERTO A (BUS A)

PA0: estado sensor 1
 PA1: estado sensor 2
 PA2: estado sensor 4
 PA3: Motor 1 ON/OFF
 PA4: Motor 2 ON/OFF
 PA5: sentido de giro Motor 1
 PA6: sentido de giro Motor 2
 PA7: estado sensor 3
 GND: Masa de la CT293+
 VCC: Alimentación TTL (5v)

PUERTO E (BUS E)

PE0: entrada digital/analógica 1
 PE1: entrada digital/analógica 2
 PE2: entrada digital/analógica 3
 PE3: entrada digital/analógica 4
 PE4: entrada digital/analógica 5
 PE5: entrada digital/analógica 6
 PE6: entrada digital/analógica 7
 PE7: entrada digital/analógica 8
 VRL: Configuración nivel bajo del conversor.
 VRH: Configuración nivel alto del conversor.

A la izquierda de la tarjeta CT293+ hay unos contactos metálicos agrupados de tres en tres y con un marco que dice sensor X. Estos contactos no forman parte del bus de control o expansión, su misión es la de servir de conector para enganchar los sensores de infrarrojos. En el capítulo 3 se explica como realizar esta unión, ahora se indica su significado.

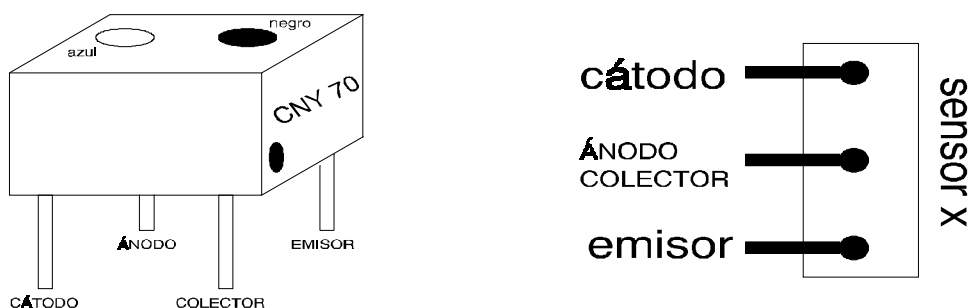


figura 2.3.2. Significado de los pines de conexión del sensor de infrarrojos.

3. INSTALACIÓN DE LOS SENSORES Y MOTORES .

La CT293+ puede incorporar bastantes tipos de sensores, también puede manejar motores paso a paso y motores de continua. En este capítulo se pretenden dar algunos consejos y guías prácticas para evitar conexiones erróneas.

3.1. CONEXIÓN DE LOS MOTORES DE CONTINUA A LA CT293+.

La tarjeta esta preparada para poder controlar dos motores de continua simultáneamente. Se realiza por medio del circuito integrado L293B que contiene dos circuitos internos con el ‘puente en H’ necesario para controlar motores. Además es un integrado que se adapta perfectamente ofreciendo una etapa de control y de potencia en un mínimo espacio. Para obtener el patillaje y el circuito de polarización de la CT293+ se recomienda mirar el apéndice A.

La conexión de los motores a la tarjeta se realiza a través de las clemas doubles llamadas ‘motor_1’ y ‘motor_2’. Estas clemas están situadas en el centro de la parte inferior de la tarjeta. Ver figura 2.1. Los motores se pueden situar lejos de la placa, aunque es aconsejable que el cable de unión no supere los 20cm, por cuestiones de ruido eléctrico, y pérdida de potencia.

Generalmente los motores tienen un sentido de giro preferente. Al girar en ese sentido las escobillas resbalan sobre el colector produciendo un desgaste pequeño. Los fabricantes de motores suelen indicar este sentido colocando un signo positivo en alguno de los conectores del motor. Si se introduce la tensión de acuerdo con ese criterio el giro obtenido será el correcto. Esto no quiere decir que el motor se vaya a romper si se introduce mal, lo único que refleja es que si la aplicación va a mover el motor en un solo sentido conviene que este coincida con el preferente para aumentar la vida útil del mismo. Cuando el motor se va a utilizar en aplicaciones que requieran ambos giros, lo anterior no afectará mucho, el desgaste de las escobillas se producirá sobre todo por los cambios de sentido y no por el rozamiento inverso que será despreciable. Se va a utilizar esta característica para definir una forma normalizada de conectar los motores, de forma que cualquier microbot construido siguiendo este patrón será compatible a nivel de software.

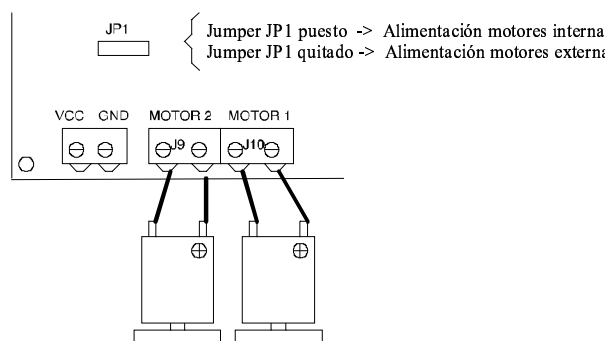


Figura 3.1.1. Esquema de la conexión de los motores

Si siguiendo la forma de conexión anterior y mirando los motores de frente se puede sacar esta tabla. Cuando el lector construya su microbot puede optar por colocar los motores al azar y luego construirse una tabla como esta o intentar situar los motores siguiendo lo anterior para obtener una tabla igual. Cualquiera de las dos opciones sirve aunque se aconseja la segunda para garantizar una compatibilidad.

DIRECCIÓN	MOTOR 1		MOTOR 2	
	BIT 5	ON (1) - DERECHA OFF (0) - IZQUIERDA	BIT 6	ON (1) - IZQUIERDA OFF (0) - DERECHA
ESTADO	BIT 3	ON (1) - motor ON OFF (0) - motor OFF	BIT 4	ON (1) - motor ON OFF (0) - motor OFF

Tabla de control de los motores

Se explica ahora la función del *jumper* JP1. Este es el encargado de seleccionar entre la alimentación interna o externa de los motores. Cuando se conecta el *jumper*, la tensión TTL (+5v) se conecta con la entrada de alimentación de los motores. La ventaja es que con una sola fuente de tensión se puede dar energía a todo el sistema. El inconveniente está en el ruido que los motores introducen en el sistema. Sobre todo aparecen caídas bruscas de tensión que pueden desprogramar la EEPROM interna. Para evitar esto está el *jumper* JP6 en la CT6811, este *jumper* protege a la eeprom haciendo un *reset* de la placa siempre que la tensión de alimentación esté por debajo de 4.5v. Esto supone un compromiso. Si se protege la eeprom el sistema puede desconectarse automáticamente en las caídas bruscas de tensión, si se quita el *jumper* JP6 de la CT6811 no se produce esa desconexión (salvo caídas muy grandes <2.5) pero se corre el riesgo de que se desprogramme la eeprom. (En los microbots del Grupo J&J el LVI (JP6) suele estar desconectado y aún así rara vez se ha desprogramado la eeprom).

Cuando el *jumper* JP1 de la CT293+ está quitado significa que se ha seleccionado la alimentación de motores externa. En este caso para que los motores se muevan se tiene que introducir la alimentación por la clema situada junto a las clemas de los motores. (Ver figura 3.1.1.). La desventaja es la necesidad de utilizar dos fuentes de alimentación, mientras que la ventaja es el poder poner más tensión a los motores. Por ejemplo se puede utilizar 12v, 9v, 1v, ... sin interferir con la alimentación TTL. Otra ventaja es que no se produce tanto ruido en la línea y el LVI puede ser compatible con los motores.

El L293B permite alimentaciones para los motores de hasta +36v y una corriente de salida de 1Amperio. Cuando se use en condiciones extremas probablemente sea necesario ponerle un disipador. En el apéndice A se describe el patillaje y la composición de este integrado.

3.2. CONEXIÓN DE UN MOTOR PASO_PASO A LA CT293+.

La CT293+ puede controlar un motor paso_paso. Para aquellas personas que no estén familiarizadas con este tipo de motores se recomienda que se lean el apéndice C donde se describen sus características y funcionamiento. En este apartado se muestra como conectarlos a la tarjeta pero no su programación que se ve en el capítulo siguiente.

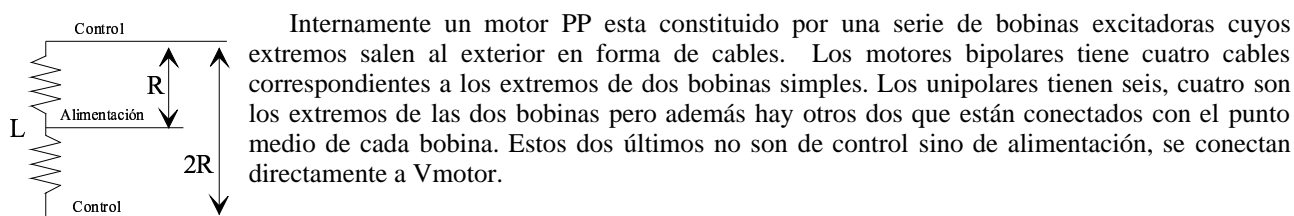


Figura 3.2.1.

Distinguir el tipo de motor no es difícil, todo se reduce a contar el número de cables que hay. Saber que cables pertenecen a la misma bobina es algo más difícil, aunque con un polímetro funcionando como óhmetro se puede averiguar. Para ello se tiene que medir la resistencia entre cables, si la lectura es de muy pocos ohmios se puede suponer que los cables pertenecen a la misma bobina. Para buscar el cable del centro en motores unipolares se utiliza el esquema de la figura 3.2.1, viendo que la resistencia mínima se produce entre uno de los extremos y el cable central.

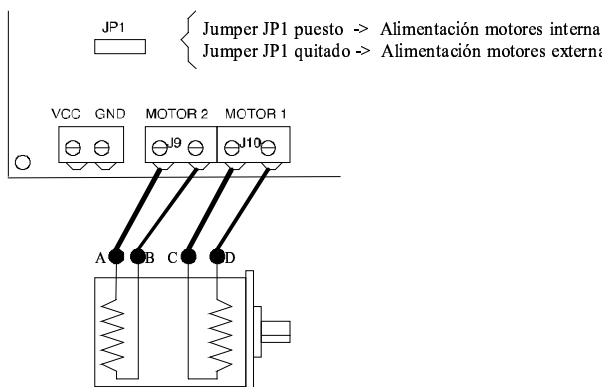


Figura 3.2.2. Conexión de un motor PP bipolar.

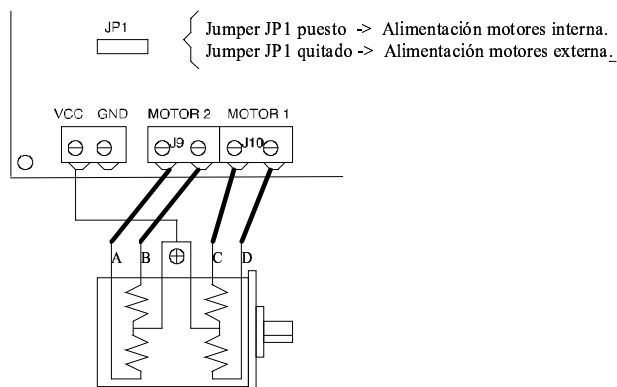


Figura 3.2.3. Conexión de un motor PP unipolar.

3.3. CONEXIÓN DE LOS SENSORES DE INFRARROJOS A LA CT293+.

Otro de los elementos que maneja la CT293+ son los infrarrojos CNY70. En concreto esta tarjeta tiene capacidad para controlar directamente cuatro. Si se recurre a las entradas digitales extras se puede llegar a controlar hasta un número de doce. Cada CNY70 incorpora un emisor y un receptor en el mismo encapsulado. Ver figura 3.3.1.

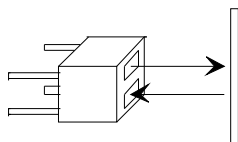


Figura 3.3.1 CNY70

En el apéndice B se comenta brevemente la utilidad de los infrarrojos y las distintas formas de usarlos. Para seguir con la explicación el lector tiene que saber que se utiliza la polarización en continua frente a la modulada ya que se adapta muy bien al objetivo buscado (seguir una línea) y además el circuito es más fácil.

La polarización en continua del emisor hace que estos infrarrojos sean de corta distancia, son capaces de distinguir el negro de otro color mientras la distancia del infrarrojo a la superficie no supere unos pocos centímetros. Lo más normal es situarlos a medio centímetro del suelo para evitar interferencias con otras fuentes de luz, ya sea solar o artificial.

A la izquierda de la tarjeta CT293+ están situados cuatro conectores acodados de tres pines donde se deben conectar los sensores de infrarrojos. Viendo ahora en el sensor CNY70 (ver figura 3.3.2) se aprecia que este tiene cuatro patas, por lo tanto hay que pasar de cuatro a tres para poder enlazar con los conectores de la CT293+. Para explicar eso se recurre al circuito de polarización (figura 3.3.2) del sensor donde se ve como las patas 'A' y 'C' están las dos conectadas a VCC. Antes de nada se cortocircuitarán dichas patas. Una vez hecho esto se conectan las tres patas (una es doble) con sus pines correspondientes en los conectores. Con ver la figura 3.3.3 bastará, tan solo se tiene que lanzar un cable desde la pata X del sensor hasta el pin X del conector.

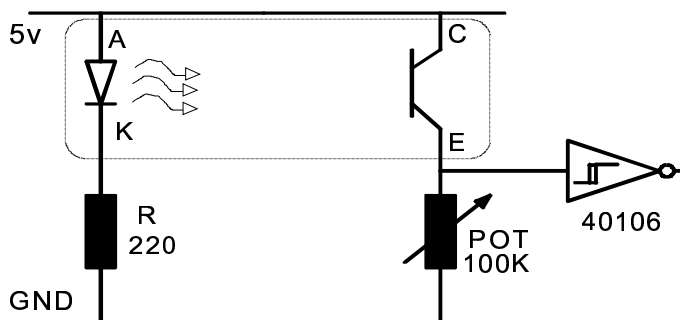
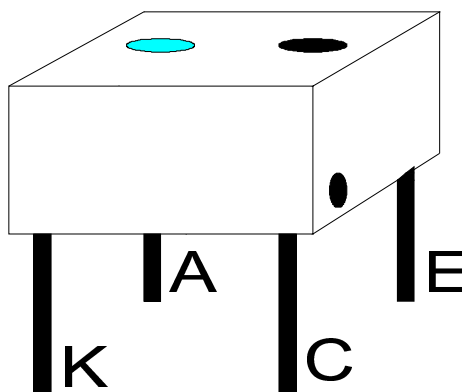
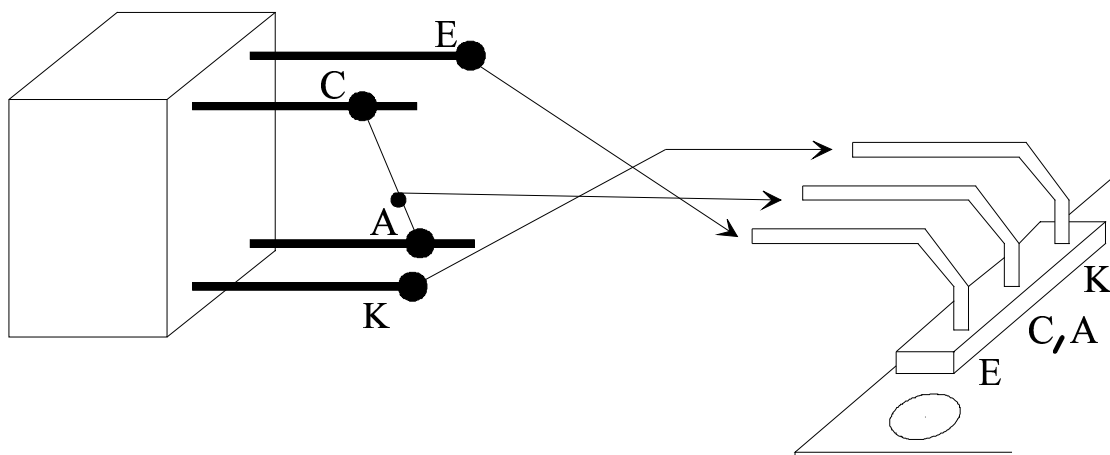


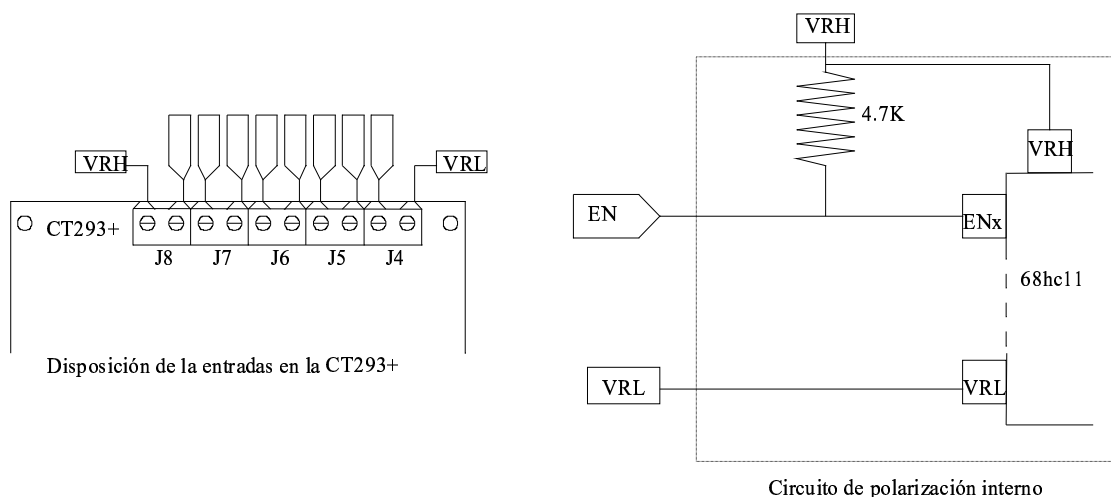
figura 3.3.2. Esquema del sensor y su circuito de polarización

En el circuito de polarización aparece un potenciómetro de 100kW pero en la tarjeta no hay ninguno, se ha sustituido por una resistencia de 47kW. Se ha perdido así la posibilidad de regular el funcionamiento de los sensores, pero tampoco hay que preocuparse pues esa regulación era útil en condiciones extremas pero en condiciones normales no.

**Figura 3.3.3. Esquema de montaje del sensor CNY70.**

3.4. CONEXIÓN CON LAS ENTRADAS DIGITALES.

Las entradas digitales se introducen por las clemas de conexión situadas en la parte superior de la tarjeta. Hay un total de 10 entradas pero sólo 8 son entradas, las otras dos se usan para establecer los niveles de referencia VRH y VRL. En modo digital estos niveles serán VRL=0 y VRH=5. Si se trabaja con la CT6811 y se tienen los *jumers* JP1 y JP2 puestos los niveles se sitúan automáticamente en VRL=GND='0' y VRH=VCC='5'.

**Figura 3.4.1. Distribución de las entradas y circuito de polarización de la CT293+.**

En la figura 3.4.1. se presenta el circuito de polarización desarrollado en la tarjeta CT293+ y la disposición de las entradas. Según el circuito de polarización la lectura que se obtiene cuando no se conecta nada en las entradas, estando los niveles de referencia a VCC y GND, es prácticamente VRH, es decir VCC. Lo que significa nivel alto '1'. Este comportamiento se debe a la resistencia de 4.7k, que se denomina resistencia de *pull up*.

Como ejemplo de conexión de dispositivos a estas entradas se va a conectar un *bumper*. En la figura 3.4.2. se describe su estructura interna y en la figura 3.4.3 se analizan las distintas posibilidades de conexión y su realización práctica en la CT293+.

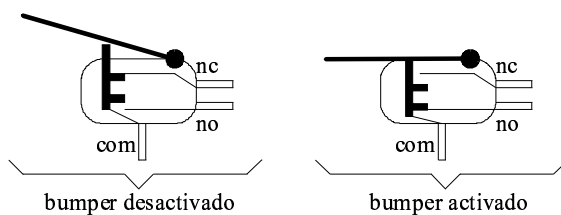


Figura 3.4.2. Estructura de un bumper.

El *bumper* tiene tres patas. La pata *com* es la común y por ella se obtiene la salida. Esta salida depende de si el *bumper* está o no activado. La pata *nc* se une con la pata *com* cuando está desactivado. Por el contrario la pata *no* se conecta a *com* cuando se active. (Se presiona sobre la palanca).

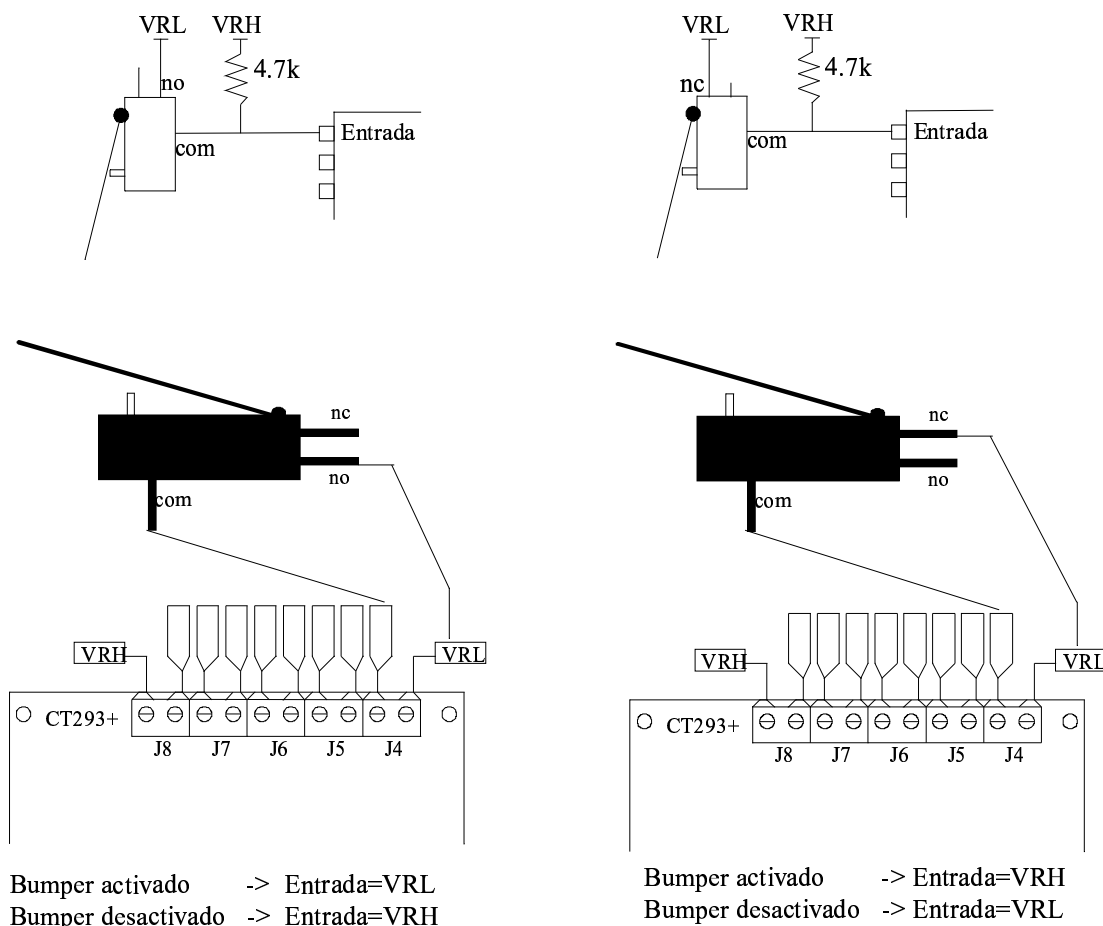


Figura 3.4.3. Conexión de un bumper a la CT293+.

En la figura se puede apreciar que se ha conectado el *bumper* a la entrada digital número uno. La utilización y programación una vez realizada la conexión se explica en el capítulo 4. Los dos esquemas anteriores se diferencian en la lectura obtenida según este activado o desactivado el *bumper*.

A estas entradas se pueden conectar cualquier otro dispositivo que sea compatible con la resistencia de pull up. Por ejemplo interruptores, pulsadores, puertas lógicas, y todo tipo de circuito que a uno se le ocurra.

3.5. CONEXIÓN CON LAS ENTRADAS ANALÓGICAS.

Las entradas digitales que se explican en el apartado anterior pueden configurarse desde la CT6811 como entradas analógicas. Este procedimiento y su utilización se explican en el capítulo 4. Ahora se indica como utilizar correctamente las entradas y su circuito de polarización. Para utilizar este modo es necesario disponer de la CT6811 y además configurarla para la utilización de los conversores analógico-digiales.

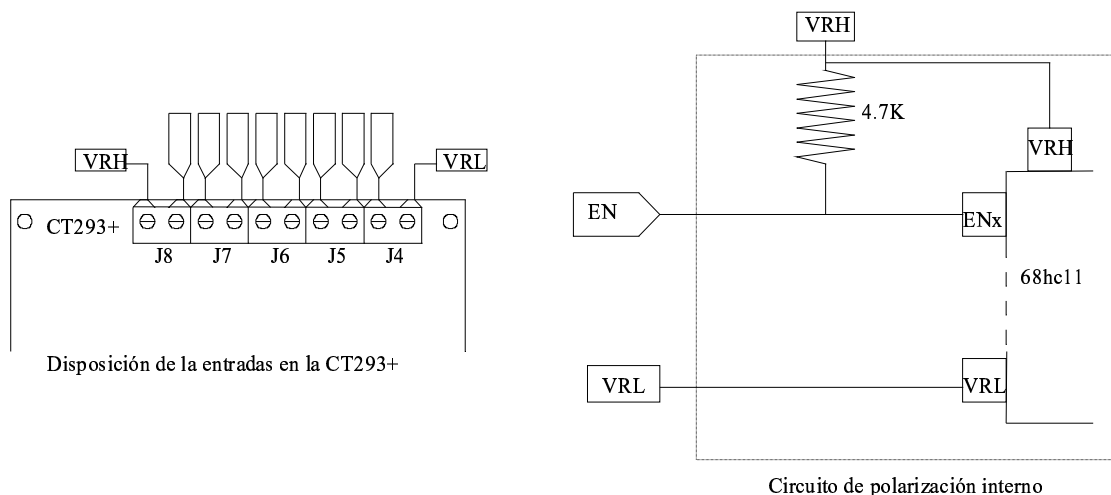


Figura 3.5.1. Distribución de las entradas y circuito de polarización de la CT293+.

El Puerto E del 68hc11 puede funcionar de dos formas distintas. La primera se corresponde con el funcionamiento en modo digital explicado los apartados 3.4 y 4.2.1. La segunda es el modo analógico que se explica aquí y en el apartado 4.2.2. Estos modos de funcionamiento no son compatibles entre si, hay que elegir entre uno u otro pero no ambos a la vez. Los dos utilizan el mismo circuito de conexión para sus entradas y lo que era una ventaja en el modo digital ahora se convierte en un inconveniente. La resistencia de pull up que antes facilitaba la conexión de ciertos dispositivos, ahora hace que la señal analógica sufra una distorsión no deseable.

Las entradas VRH y VRL son los niveles de referencia que necesita el conversor analógico digital del 68hc11 para su correcto funcionamiento. Lo normal es utilizar VRH=VCC y VRL=GND, por eso esta opción se encuentra disponible en la CT6811 con tan solo conectar los *jumpers* JP1 y JP2. (situados en la esquina superior derecha). Si se desconectan hay que introducir los niveles por las entradas correspondientes. El valor de estos debe estar comprendido entre (0v - 6v). Si se superan los 6v las conversiones pueden ser erróneas, y si se introduce una tensión por debajo de cero, negativa, pudiendo dañar permanentemente la entrada de nivel analógica del *microcontrolador* 68hc11. Además por las entradas analógicas no hay que introducir nunca tensiones negativas, pues probablemente se rompería dicha entrada.

Otros datos importantes son que no se debe poner una resistencia serie con la entrada superior a 10k Ω y que la corriente máxima de entrada no puede exceder los 25mA.

Se aconseja al lector que antes de utilizar los conversores analice el circuito de entrada al conversor para ver si alguna de las condiciones anteriores no se cumple. Además no olvidarse nunca de la resistencia de pull up que la tarjeta CT293+ introduce en las entradas.

Los mayores errores cometidos en aplicaciones con los conversores se deben:

- 1) *Impedancia de carga a la entrada analógica superior a 10k Ω , produce una medida errónea de la conversión. Se evita no superando los 10k Ω .*
- 2) *Impedancia de carga a la entrada demasiado pequeña que produce una corriente de entrada superior a los 25mA. Puede dañar permanentemente la entrada utilizada del 68hc11. Se evita colocando una resistencia que haga que la corriente de entrada sea inferior a los 25mA.*
- 3) *Introducir una tensión negativa por la entrada cuando hay una impedancia de carga de bajo valor conectada a ella. Puede dañar permanentemente la entrada utilizada del 68hc11. La mejor forma de evitar esto es no introducir nunca tensiones negativas. Tampoco en VRL.*

Se propone un ejemplo de conexión de dos potenciómetros uno de 4.7kW y otro de 100kW a las entradas analógicas de la CT293+. Se representa el circuito y los rangos de tensiones que se obtienen al realizar la conversión (a 8 bits).

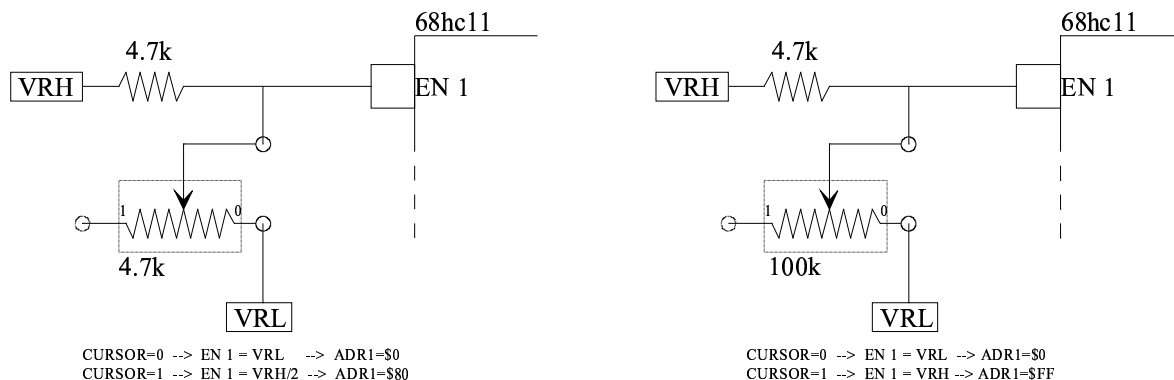


Figura 3.5.2. Circuito de entrada propuesto y valores obtenidos teóricos.

En la figura 3.5.3. se ve como se realiza la conexión física del circuito de la figura 3.5.2. Las aplicaciones analógicas son muchas y cada una tiene características distintas. Sobre todo en lo que respecta a su circuito. Se recomienda tener cuidado con no respetar algunas de las normas anteriores.

La programación de los conversores es un poco más difícil que en el caso digital pero no debe de asustar. Disponer de este tipo de recursos analógicos hace que las aplicaciones sean más potentes. Se pueden realizar controladores de temperatura, microbots que sigan la luz, posicionamiento de motores de continua, etc. En el capítulo cuatro se explica como usar y configurar los conversores.

Para aquellas personas que quieran conocer como funciona el conversor por dentro y cuales son sus limitaciones respecto a exactitud, precisión, etc se recomienda leer el capítulo 12 del M68HC11 REFERENCE MANUAL (de Motorola).

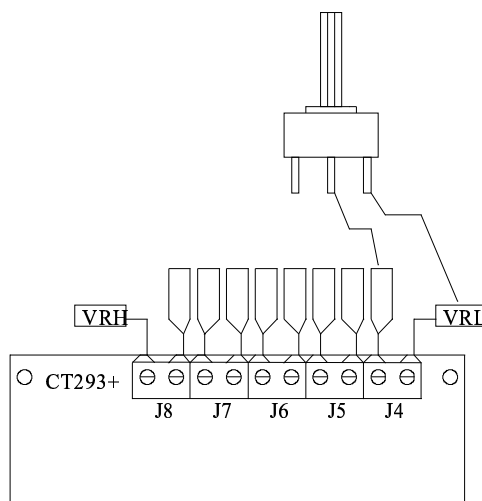


Figura 3.5.3. Conexión del potenciómetro.

4. PROGRAMACIÓN DE LA TARJETA CT293+

La tarjeta esta pensada para conectarse a la tarjeta CT6811. Aunque se puede conectar a otros tipos de controladores. El rendimiento máximo de la tarjeta se saca cuando se conecta al puerto A y al puerto E de la CT6811. El control de la CT293+ se realiza a través de dos bytes. Un byte controla los motores y sensores de infrarrojos y el otro las entradas digitales/analógicas.

Se divide este capítulo en dos partes, en la primera se explica como controlar el bloque A de la CT293+ (Motores e infrarrojos) y en la segunda se explica como usar el bloque E (entradas digitales/analógicas).

4.1. PROGRAMACIÓN DEL BLOQUE A. MOTORES Y SENSORES DE INFRARROJOS.

Lo primero es unir el Bloque A de la CT293+ con la CT6811 utilizando el bus A., para ello conectar un cable de bus entre el Puerto A de la CT293+ (conector J1) y el Puerto A de la CT6811 (conector J1). Aquellas personas que no poseen la CT6811 tienen que conectar el Puerto A de la CT293+ con el puerto correspondiente en su sistema de control.

Automáticamente al colocar dicho cable se proporcionará la alimentación TTL a la CT293+. Ahora hay que elegir una de las dos opciones siguientes. La primera consiste en utilizar esa alimentación TTL para alimentar los motores, para ello hay que colocar en su sitio el *jumper* JP1. La segunda consiste en alimentar los motores con otra fuente de alimentación, para ello hay que desconectar el *jumper* JP1 e introducir la alimentación externa por la clema J2. Tener cuidado con la polaridad.

Al utilizar la alimentación externa tener mucho cuidado con no olvidar desconectar el jumper JP1 y con la polaridad de la tensión de alimentación de los motores. Si se utiliza alimentación única antes de colocar el jumper JP1 desconectar la alimentación externa de los motores (clema J2).

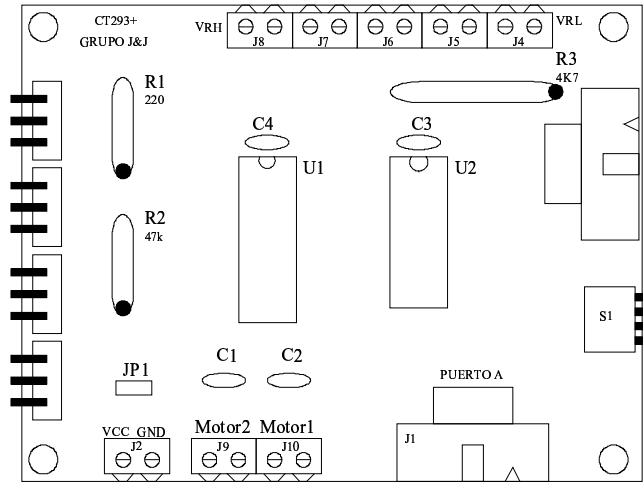


Figura 4.1.1. Componentes de la CT293+

Una vez realizado lo anterior el control de la CT293+ es muy sencillo. Tan solo se utiliza un byte de control, que se corresponde con el Puerto A de la CT6811, posición de memoria \$1000 del *microcontrolador* 68hc11. El significado de los bits de dicho byte se detalla a continuación.

SWITCH 2	MOTOR 2	MOTOR 1	MOTOR 2	MOTOR 1	SWITCH 1	SWITCH 3	SWITCH 4	Puerto A \$1000
SENSOR 3	Dirección	Dirección	ON / OFF	ON / OFF	SENSOR 4	SENSOR 2	SENSOR 1	
bit 8 - in	bit 7 - out	bit 6 - out	bit 5 - out	bit 4 - out	bit 3 - in	bit 2 - in	bit 1 - in	

4.1.1. Los Sensores De Infrarrojos.

Los bits en blanco significan que son de entrada, y proporcionan el estado de los infrarrojos. Para que la lectura de estos bits sea correcta es necesario activar las salidas de los sensores de infrarrojo correspondientes. Eso se hace con los interruptores que hay en la placa. La correspondencia entre la localización física del infrarrojo y su interruptor se representa en la siguiente figura.

Sensor 1	>>	switch 4	>>	BIT 1
Sensor 2	>>	switch 3	>>	BIT 2
Sensor 3	>>	switch 2	>>	BIT 8
Sensor 4	>>	switch 1	>>	BIT 3

Figura 4.1.3. Tabla resumen.

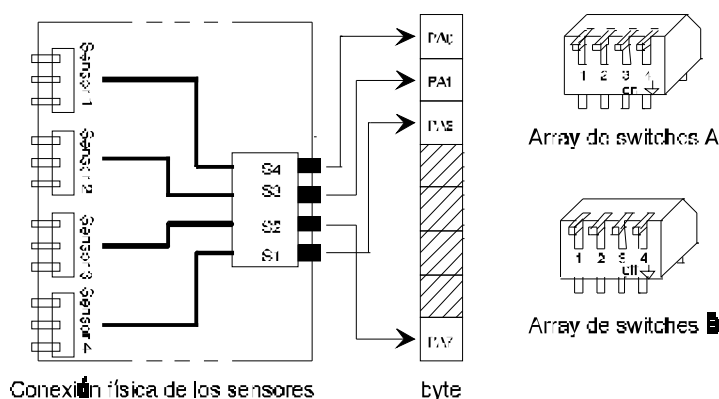


Figura 4.1.2. Asignación de los sensores.

Los interruptores que se han puesto en la placa tienen la utilidad de anular y dejar libres los bits que emplean los infrarrojos. Si el sistema sólo va a manejar motores se pueden utilizar los bits restantes como entradas sin entrar en conflicto con los sensores. Esto es importante cuando se trabaja con el Puerto A de la CT6811 ya que esos bits son muy útiles y no conviene malgastarlos. Como norma básica se aconseja que los interruptores estén a ON cuando se vayan a manejar los sensores, en caso contrario ponerlos a OFF. En la figura 3.1.2. hay dibujados dos tipos de microinterruptores, el tipo A se caracteriza porque para cerrar el circuito (poner ON) hay que situar la palanca hacia abajo. El tipo B se caracteriza por lo contrario, para poner ON hay que situar la palanca hacia arriba.

La **interpretación** del bit de un sensor es la siguiente : bit a **1** significa detectado **Negro**
bit a **0** significa detectado **Blanco (No negro)**

Se propone un ejercicio. Realizar un programa que lea el sensor 4, si está detectando negro que no encienda el LED de la CT6811 y si está detectando blanco que lo encienda.

Para realizar este programa se activará el microinterruptor S1 pues se corresponde con el sensor4. Luego se procederá a crear el software. Hay que leer el byte \$1000 de la CT6811, es decir el Puerto A. De este byte se ignorarán todos los bits menos el bit 3 (PA2) que es el que corresponde al sensor 4. Según el valor leído se actuará en consecuencia con lo propuesto.

```
; programa de ejemplo de la lectura de un sensor
; si el sensor 4 ve negro se apaga el LED, si ve blanco se enciende el LED.
```

```
      ORG $0
inicio
      LDAA $1000 ; se carga en el Acumulador A el valor del puerto A.
      ANDA #$04  ; nos quedamos con el bit-3 que es el que nos interesa
      CMPA #$04  ; bit_3 es uno ?
      BNE  cero  ; no -> salta a 'cero'
      CLRA      ; si -> apago el LED
      STAA $1000 ;
      BRA  inicio ; vuelve a empezar
cero
      LDAA #$40  ; enciendo el LED
      STAA $1000 ;
      BRA  inicio ; vuelve a empezar
      END        ; fin del programa
```

El programa anterior no esta optimizado, se propone al lector que lo simplifique. Como orientación se indica que puede pasar de ocupar 22 bytes (el de arriba) a ocupar 16 bytes. Se recomienda mirar el juego de instrucciones y reordenar el código. De todas formas en el apartado 4.1.2. se indica la solución.

SWITCH 2	MOTOR 2	MOTOR 1	MOTOR 2	MOTOR 1	SWITCH 1	SWITCH 3	SWITCH 4	Puerto A
SENSOR 3	Dirección	Dirección	ON / OFF	ON / OFF	SENSOR 4	SENSOR 2	SENSOR 1	\$1000
bit 8- in	bit 7 -out	bit 6 - out	bit 5 - out	bit 4 - out	bit 3 -in	bit 2 - in	bit 1 - in	

4.1.2. Los Motores De Continua.

Al bloque A pertenece la gestión de los motores. Los bits sombreados (del byte de control) son salidas y se encargan del funcionamiento de los motores. Los bits marcados con DIR seleccionan el sentido de giro. Los bits marcados con CE indican si se conecta o no el motor. El control de los motores se realiza con un circuito que tiene como parte principal el chip L293B que es capaz de suministrar hasta un Amperio a cada uno de los motores. A continuación se proporciona una tabla con la programación de los bits. Esta tabla hace referencia al modelo Quark del GrupoJ&J. En otros modelos el concepto giro a derechas o a izquierdas de un motor puede verse intercambiado, pues depende de como se mire al motor y de como se conecte a la placa. Para sacar esta tabla se han conectado los motores como se dijo en el capítulo tres y cada motor se ha mirado de frente. (Es decir la rueda situada entre los ojos y el motor). Se recomienda que cada uno conecte los motores como quiera y que luego se construya una tabla como la siguiente. Otra opción es probar diferentes conexiones hasta encontrar una similar a la propuesta aquí. Esto no es complicado, pensar que todo consiste en invertir los cables que unen el motor con la placa.

	MOTOR 1		MOTOR 2	
DIRECCIÓN	BIT 5	ON (1) - DERECHA OFF (0) - IZQUIERDA	BIT 6	ON (1) - IZQUIERDA OFF (0) - DERECHA
ESTADO	BIT 3	ON (1) - motor ON OFF (0) - motor OFF	BIT 4	ON (1) - motor ON OFF (0) - motor OFF

Tabla de control de los motores

Al igual que antes se propone un ejercicio. Se quiere leer el sensor 1, si esta sobre negro el motor 1 empezará a girar. Si esta sobre blanco el motor se parará.

Para resolver el problema se puede utilizar el programa del ejercicio anterior pero cambiando los valores de los bits. Ahora el sensor 1 se corresponde con el microinterruptor 4 y con el bit-1 (PA0). El bit que activa el motor 1 es el bit-3, el sentido en este ejercicio no importa.

```
; programa de ejemplo de control de motores
; si el sensor 1 ve negro encendemos el motor 1 , si ve blanco lo apagamos.
```

```
PORTA equ $0
```

```

      ORG  $0
      LDX  #$1000
inicio
      STAA $1000
      BRCLR PORTA,X $01 cero
      LDAA #$08    ; pongo un 1 en el bit 3 para encender el motor
      BRA  inicio  ; vuelve a empezar
      cero CLRA     ; apago el motor
      BRA  inicio  ; vuelve a empezar
      END          ; fin del programa
```

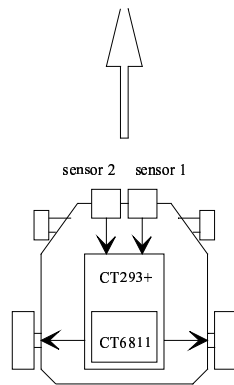
Realmente este programa es igual que el anterior, un poco modificado pero en lugar de encender el LED se enciende un motor. Una diferencia curiosa es que este programa ocupa 16 bytes....

4.1.3. Un Microbot Que Sigue La Línea Negra.

Finalmente se propone programar algo más complejo. El lector tiene que realizar un programa que permita a un microbot seguir una línea negra. Se considera que el microbot tiene dos sensores de infrarrojos (sensor 1 y sensor 2) y que dispone de dos motores independientes.

Antes de empezar a programar se tiene que analizar el problema. Se supone que los dos sensores van a ir por el interior de la línea negra. De tal forma que cuando uno se salga de ella se tiene que corregir la dirección para volver a llevarlo dentro. Este algoritmo hace que el microbot siga dicha línea pero puede dar problemas cuando en el circuito existan rotondas, (el microbot se queda dando vueltas sin salir de ella). Para evitar esto se puede diseñar un algoritmo que siga el borde de la línea.

Siguiendo con la teoría cuando se salga de la línea el sensor izquierdo hay que hacer que el microbot gire a la derecha, cuando se salga el sensor derecho el microbot tiene que girar hacia la izquierda.



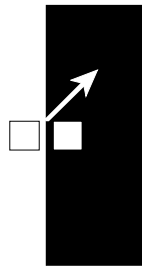
Esquema del microbot sobre el que se desarrolla la solución propuesta al ejercicio anterior. Vemos como se sitúan los sensores de infrarrojos en la parte delantera del microbot. Los motores proporcionan tracción trasera independiente.

Prestar atención a la localización de los sensores y motores, sobre todo que motor y sensor son los que se sitúan en la derecha, y cuales lo hacen en la izquierda.



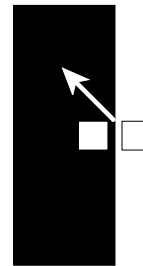
Sensor_1 -> 1
Sensor_2 -> 1
Motor_1 -> gira izquierda
Motor_2 -> gira derecha

AVANZAMOS



Sensor_1 -> 1
Sensor_2 -> 0
Motor_1 -> gira izquierda
Motor_2 -> gira izquierda

GIRAMOS DERECHA



Sensor_1 -> 0
Sensor_2 -> 1
Motor_1 -> gira derecha
Motor_2 -> gira derecha

GIRAMOS IZQUIERDA

Figura 4.1.4. Detalle del microbot y del algoritmo de programación.

En la figura 4.1.4 se explica de modo gráfico el comportamiento del microbot. También se detalla la situación de los motores y sensores. En concreto el sensor_1 y el motor_2 se sitúan en el lado derecho. Mientras que el sensor_2 y el motor_1 lo hacen en el izquierdo. El programa que se propone lee el estado de los sensores y bifurca a la acción pertinente. Si el estado de los dos sensores es nivel alto, significa que están dentro de la línea negra y entonces se tiene que ir recto.

En la siguiente tabla se indica la misma información que en la figura pero ampliada, se pone el valor que deben de tener los motores para producir giros a derechas e izquierdas.

SENSOR 1 (PA0)	SENSOR 2 (PA1)	ACCIÓN	ESTADO M1 (PA3)	DIREC M1 (PA5)	ESTADO M2 (PA4)	DIREC. M2 (PA6)	BYTE control
1	1	AVANZAR	1	0	1	0	\$18
0	1	IZQUIERDA	1	1	1	0	\$38
1	0	DERECHA	1	0	1	1	\$58

```

; Programa de control para hacer que un microbot
; siga una línea negra pintada sobre fondo blanco.

ORG $0                      ; programa para la RAM interna

LDX #$1000
inicio
STAA $0,X                  ; escribimos en el puerto A
BRSET $0,X $03 avanza      ; Si los sensores a 1 -> avanza
BRSET $0,X $01 derecha     ; Si sensor_1 a 1 -> derecha
BRSET $0,X $02 izquierda   ; Si sensor_2 a 1 -> izquierda

derecha
LDAA #$58                  ; configuro motores para girar
BRA inicio                 ; a la derecha

izquierda
LDAA #$38                  ; configuro motores para girar
BRA inicio                 ; a la izquierda

avanza
LDAA #$18                  ; configuro motores para avanzar
BRA inicio

END

```

Como puede apreciar es muy fácil cambiar el programa para que el microbot siga el borde de la línea negra. Tan solo hay que cambiar las condiciones de bifurcación. (Instrucción BRSET).

4.1.4. Programación Avanzada De Los Sensores De Infrarrojo.

El programa anterior lee los sensores directamente a través de un puerto. Este procedimiento no siempre es eficaz, cuando se están usando los sensores de infrarrojos como encoders se puede perder la cuenta de uno de los pasos. Se recuerda que un encoder es un dispositivo que es capaz de contar el ángulo de giro de un motor. Generalmente se basan en colocar en el eje del motor un disco con ranuras estrechas, este disco además se sitúa entre un emisor y un receptor de infrarrojos. De tal forma que al girar el motor las ranuras del disco dejan pasar o interrumpen alternativamente el haz infrarrojo. A la salida del receptor de infrarrojos se tiene una señal cuadrada, contando las crestas se puede saber cuantas ranuras han pasado y en consecuencia que ángulo se ha girado.

Una forma práctica de simular un encoder con un CNY70 es pegar a la rueda un disco con rayas blancas y negras pintadas. Si ahora se pone el sensor CNY enfrente con el disco, según se sitúe la raya negra o la blanca delante de él se tendrá un '1' o un '0', la misma señal cuadrada de antes. Contando el número de variaciones se obtiene el ángulo girado. Las rayas pintadas tienen que ser grandes para que funcione correctamente, mirar la figura 4.1.5.

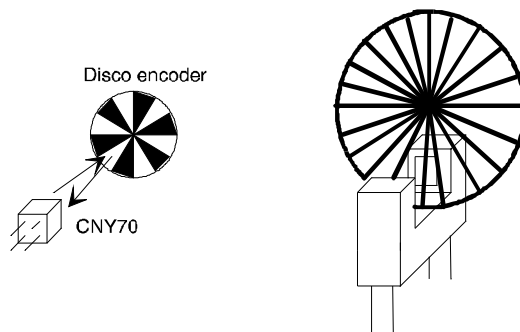


Figura 4.1.5. Dos modelos de encoders.

Como ya se ha mencionado el problema de leer los sensores de infrarrojo cuando actúan como encoders es que si se pierde la cuenta de un paso se lee un ángulo erróneo. Por eso es mucho más eficaz utilizar los capturadores de entrada del 68hc11, sobretodo porque permiten leer por medio de interrupciones, forma en la que no se pierde la lectura de ningún paso. Los capturadores están situados en el Puerto A del *microcontrolador* 68hc11, esa es la razón por la que el Bloque A de la CT293+ se controla desde este puerto.

Aquellas aplicaciones que requieran lecturas críticas en el tiempo deben ser canalizadas a través de los sensores conectados a los capturadores. Por ejemplo en el microbot GOLIAT se implementan los encoders con los sensores 1 y 2, utilizando los capturadores 3 y 2 para saber cuanto ha girado cada motor. Los otros dos sensores se encargan de controlar la línea negra, pero sin utilizar interrupciones.

En la siguiente tabla se muestra la correspondencia entre la función del puerto A con respecto a la CT293+ y los recursos que proporciona el 68hc11 para cada bit de dicho puerto.

Número de PIN del PUERTO A	Función del PIN	Sensor correspondiente
PA 0	entrada: capturador 3 / pa0	Sensor 1
PA 1	entrada: capturador 2 / pa1	Sensor 2
PA 2	entrada: capturador 1 / pa2	Sensor 4
PA 3	salida: comparadores 5,1 / pa3	On / Off motor 1
PA 4	salida: comparadores 4,1 / pa4	On / Off motor 2
PA 5	salida: comparadores 3,1 / pa5	Dirección motor 1
PA 6	salida: comparadores 2,1 / pa6	Dirección motor 2
PA 7	bidireccional: comparador 1 / pa7	Sensor 3

Tabla descriptiva de los recursos del Puerto A del 68hc11 y su correspondencia con la CT293+.

4.1.5. Programación De Motores Paso_Paso.

El funcionamiento de estos motores no tiene nada que ver con el de los motores de corriente continua. En estos motores hay cuatro cables de control (A,B,C,D) por donde se introducen las tensiones adecuadas para polarizar correctamente las bobinas internas del motor, permitiendo que el eje del motor gire un ángulo determinado, más conocido como dar un paso. Si se quiere dar otro paso se tiene que introducir otro grupo de tensiones diferentes al anterior. En el apéndice C se describen dos circuitos de polarización que permiten generar las tensiones necesarias a partir de unas señales de control TTL, (CA, CB, CC, y CD). Al final se obtiene una secuencia de control válida para cualquier tipo de motor paso_paso. Ver tabla de la derecha.

	CA	CB	CC	CD
1	1	0	1	0
2	0	1	1	0
3	0	1	0	1
4	1	0	0	1

Para continuar se supone que el circuito de conexión del motor paso_paso se corresponde con el propuesto en el apartado 3.2 para un motor bipolar. En este circuito los cables de control del motor se han conectado directamente a las salidas 'motor 1' y 'motor 2'. En concreto la bobina 1 formada por A,B se ha unido con motor_2 y la bobina 2 formada por C,D se ha unido con motor_1.

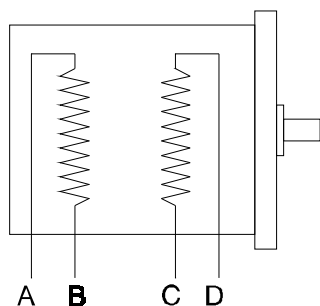


Figura. Motor PP bipolar.

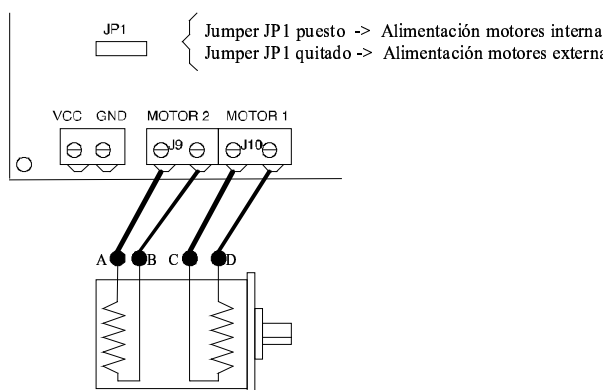


Figura. Conexión de un motor PP bipolar.

El problema que se plantea ahora es que el circuito de polarización que se tiene no se corresponde con ninguno de los propuestos en el apéndice C. Esto hace que las señales de control que se deben de utilizar no sean iguales a CA, CB, CC y CD por lo que la tabla anterior no es válida. Se tiene que buscar una nueva.

	A	B	C	D
1	Vcc	Gnd	Vcc	Gnd
2	Gnd	Vcc	Vcc	Gnd
3	Gnd	Vcc	Gnd	Vcc
4	Vcc	Gnd	Gnd	Vcc

En la tabla de la izquierda se indica la secuencia de tensiones que deben introducirse en cables de control A,B,C,D del motor paso_paso para permitir un movimiento correcto. De la observación de esta tabla tiene que salir una nueva secuencia de control.

Viendo el byte de control del bloque A de la CT293+ se tiene que los únicos bits que se pueden utilizar para controlar el motor paso_paso son el bit_4, bit_5, bit_6 y el bit_7, ya que son los que controlan las salidas 'motor_1' y 'motor_2'. De estos bits el bit_4 y el bit_5 no sirven pues producen salidas indeseadas. Por ejemplo si el bit_5 se pone a nivel bajo '0', las salidas del 'motor_2' se ponen a nivel bajo también. Es decir A=GND y B=GND, y si ahora se mira la tabla de tensiones se ve que esta configuración no aparece, por lo tanto no es válida.

SWITCH 2	MOTOR 2	MOTOR 1	MOTOR 2	MOTOR 1	SWITCH 1	SWITCH 3	SWITCH 4	Puerto A
SENSOR 3	Dirección	Dirección	ON / OFF	ON / OFF	SENSOR 4	SENSOR 2	SENSOR 1	\$1000
bit 8- in	bit 7 -out	bit 6 - out	bit 5 - out	bit 4 - out	bit 3 -in	bit 2 - in	bit 1 - in	

Los otros dos bits que quedan son los encargados de seleccionar el sentido de giro de los motores de continua. Para conseguir esto invierten la polaridad de sus salidas respectivas ('motor_1, motor_2'). Esta característica es suficiente para poder generar la tabla de tensiones..

	A	B	C	D	bit-7	bit-6	bit - 5	bit - 4
1	Vcc	Gnd	Vcc	Gnd	1	0	1	1
2	Gnd	Vcc	Vcc	Gnd	0	0	1	1
3	Gnd	Vcc	Gnd	Vcc	0	1	1	1
4	Vcc	Gnd	Gnd	Vcc	1	1	1	1

Esta tabla es la que se usa para manejar los motores paso_paso. Además, aunque se ha calculado partiendo de un motor bipolar también sirve para uno unipolar.

Si el motor es bipolar y se pone el bit_5 y el bit_4 a nivel bajo se desactiva el motor paso_paso pues se introduce GND en todas sus entradas. Si el motor es unipolar no se puede aplicar lo anterior salvo que el punto común en lugar de derivarlo a VCC se lleve a GND. Al hacer esto el sentido de giro para la misma secuencia se invierte, pero esto se puede solucionar mediante software o intercambiando las conexiones de las bobinas (B,A, D, C). La utilidad es que ahora el bit_5 y el bit_4 pueden desconectar el motor por la misma razón de antes, todas las entradas de las bobinas se pondrán a GND. Al desconectar el motor se ahorra energía pero se pierde la fuerza estática del motor. Fuerza que impedía al eje abandonar su posición hasta que no llegase una nueva secuencia.

El programa siguiente supone que hay un motor PP unipolar conectado a la CT293+. La forma de conexión se supone igual a la de la figura 3.2.3. Aunque todavía no se ha visto su programación también se supone que hay conectado a la entrada digital 8 un *pulsador o bumper*. (En el apartado 4.2. se describe este elemento).

El programa hace que el motor PP de un paso cuando se pulse el *bumper*. Para no complicar el programa no hay posibilidad ni de cambiar el sentido de giro ni de desactivar el motor. Por eso tener cuidado con el consumo del mismo. Ya que al estar siempre activo **se puede poner en peligro el L293B cuando la corriente requerida sea del orden del Amperio.**

```

; Ejemplo de control de un motor paso_paso.
; Cada vez que se pulse el bumper se dará un paso
; El motor ser unipolar y conectado a la CT293+
;
; La secuencia que hay que mandar es
;      bit7  bit6  bit5  bit4
;  1-   0     0     1     1
;  2-   0     1     1     1
;  3-   1     1     1     1
;  4-   1     0     1     1

      ORG $0

PORTA  equ $0
PORTE  equ $A

      LDX  #$1000
      LDY  #UNO
inicio LDA  0,Y
      BRSET PORTE,X $80 inicio ; esperamos hasta pulsar bumper
      STAA 0,X                ; mandamos el nuevo código
      BSR   PAUSA              ; realizamos una pausa
      INY                      ; leemos código siguiente
      CPY   #FIN               ; verificamos retorno principio
      BNE   inicio
      LDY   #UNO
      BRA   inicio

PAUSA  PSHX                    ; Subrutina que ejecuta una pausa
      LDX  #$FFFF
bucle  DEX
      CPX  #$0
      BNE  bucle
      PULX
      RTS

UNO    DB $18                  ; tabla de datos
DOS    DB $38
TRES   DB $78
CUATRO DB $58
FIN    DB $0

      END

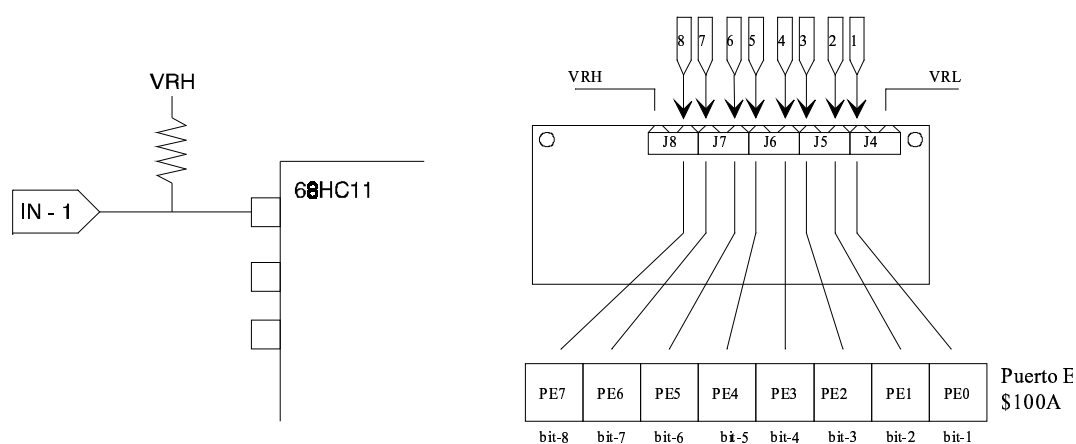
```


4.2. PROGRAMACIÓN DEL BLOQUE E. ENTRADAS DIGITALES / ANALÓGICAS.

Lo primero que se tiene que hacer es unir el Bloque E de la CT293+ con la CT6811 utilizando el bus E. Se conecta un cable de bus entre el Puerto E de la CT293+ (conector J3) y el Puerto E de la CT6811 (conector J3). El byte de control de este bloque está situado en la posición de memoria \$100A, que se corresponde con el puerto E. Para aquellas personas que no tengan la CT6811 tienen que conectar el Puerto E de la CT293+ con el puerto correspondiente en su sistema de control.

Este cable de bus tiene diez hilos, ocho de ellos se corresponden con bits de entrada y los otros dos se utilizan cuando se trabaja con las entradas analógicas. La principal función del bloque E es la de proporcionar ocho entradas con pull-up. Si no hay nada conectado en ellas se lee en los bits correspondientes un nivel alto '1'. Cuando se conecte alguna señal en la entrada dependiendo de su valor se obtiene un nivel alto o bajo. Tener cuidado al conectar las entradas con el **PULL-UP** interno que proporciona la CT293+, sólo se pueden conectar aquellos dispositivos que no sean incompatibles con él. Un dispositivo típico para conectar es un *bumper* o interruptor de fin de carrera.

Cuando se utiliza la CT6811 el Puerto E se puede utilizar de dos formas diferentes. La primera se corresponde con lo anterior, utilizar el puerto E como entradas digitales con pull-up. La segunda es configurar dicho puerto para leer canales analógicos. Aunque el bloque E no se pensó para usarlo de forma analógica las pruebas realizadas con diferentes sensores de este tipo dieron buenos resultados. Por eso se documenta esta posibilidad aunque se advierte que debido a las resistencias de pull-up hay una distorsión en la medida. Distorsión por otro lado conocida y por tanto corregible por hardware o por software.



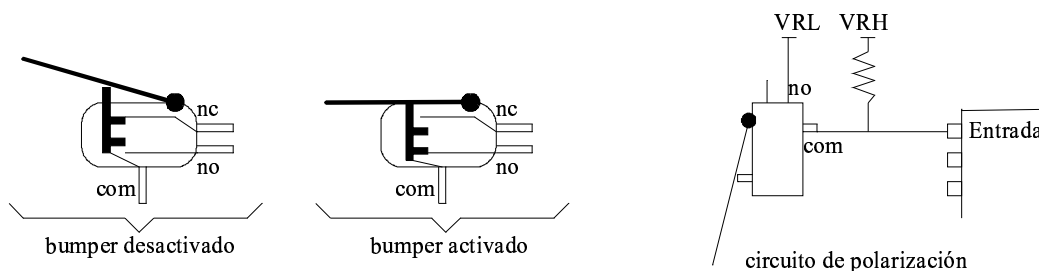
Circuito de entrada para las señales del bloque E.

Correspondencia entre las entradas y los bits del byte de control

4.2.1 Ejemplo De Utilización En Modo Digital.

Este bloque E está especialmente diseñado para poder conectar pulsadores sin ningún esfuerzo en la tarjeta. Un tipo de pulsador puede ser un *'bumper'* o interruptor de fin de carrera. En la figura (siguiente página) se recuerda su estructura interna y el circuito de polarización al conectarse a la CT293+. Viendo dicho circuito se aprecia que cuando no se activa el *bumper*, la lectura que se obtiene en la entrada digital es nivel alto '1'. Cuando se activa el *bumper* la entrada se cortocircuita con VRL, lo que implica que ahora la lectura sea nivel bajo '0'. Se recuerda también que los niveles de referencia VRL y VRH se tienen que definir salvo que se este utilizando la CT6811 con los *jumpers* JP1 y JP2 conectados. En este caso por defecto se tiene VRH=VCC y VRL=GND.

Si no se conectan los *jumpers* se tienen que establecer los niveles externamente. El rango de tensiones admisible es de 0v a 5v. Utilizando las entradas en modo digital estos valores son VRL=0v y VRH=5v. En el modo analógico es cuando más se suelen usar otras referencias distintas.



En el circuito de polarización propuesto, cuando el *bumper* esta inactivo, se lee nivel alto a la entrada debido a la resistencia de pull up. Cuando se activa se lee nivel bajo pues la entrada se ha cortocircuitado con VRL dentro del *bumper*. Este circuito no solo es válido para *bumpers* sino que también se puede conectar cualquier circuito exterior al cual no le afecte tener una resistencia de pull_up a la salida.

El programa que se propone tiene que leer el estado de un *bumper* conectado a la entrada digital 8 y encender un LED cuando se active, cuando se apriete la palanca. Se supone que la conexión es la misma que la figura de arriba con los niveles VRH=VCC y VRL=GND.

La entrada 8 se corresponde con el bit ocho del byte de control E situado en la posición de memoria \$100A del *microcontrolador* 68hc11. Al apretar la palanca en la entrada se lee nivel bajo '0', si no se aprieta se lee nivel alto '1'.

```
; Programa de ejemplo de las entradas digitales
; Conectamos un bumper en la entrada 8, cada vez que lo
; pulsemos el LED se enciende.
```

```
ORG $0

PORTA equ $0
PORTE equ $A

LDX #$1000
inicio STAA PORTA,X
BRSET PORTE,X $80 apaga
LDAA #$40
BRA inicio
apaga CLRA
BRA inicio
END
```

4.2.2 Ejemplo De Utilización En Modo Analógico.

La tarjeta CT293+ tiene un modo de funcionamiento analógico. La selección de este modo se realiza desde la CT6811 pues es realmente el *microcontrolador* 68hc11 quien ofrece este servicio. Al configurarlo para funcionar en este modo el Puerto E que antes era digital deja de ser válido y ahora sus entradas se conectan con las entradas analógicas. Por eso en la CT293+ las entradas analógicas y digitales son las mismas. El lector debe saber que ambos modos son incompatibles, si elige el digital anula el analógico y viceversa. También es verdad que utilizando el modo analógico se pueden leer los valores digitales como si se tratasen de señales analógicas. Pero al revés no se puede.

En el apartado 3.5. se describe el circuito de entrada que proporciona la CT293+ para las entradas analógicas, también se pone un ejemplo de conexión de un potenciómetro. En este apartado se explica como configurar y programar el conversor. Debido a que este modo es totalmente dependiente del 68hc11 y que es un poco más complicado que el caso digital se recurre al capítulo 4 del libro ‘ MICROCONTROLADOR 68HC11: FUNDAMENTOS, RECURSOS Y PROGRAMACIÓN’ del Grupo J&J.

Introducción

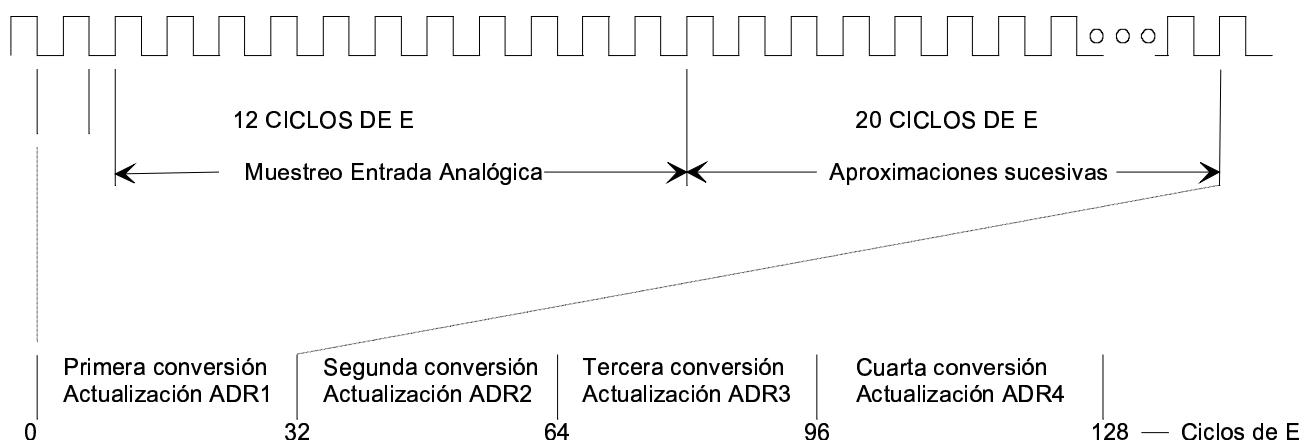
El *microcontrolador* 68HC11 tiene una serie de conversores analógico digitales que son bastante útiles y que le han dotado de su gran popularidad. En este apartado se va a indicar cómo usarlos , describiendo los distintos registros de control, y al finalmente se propone un ejemplo.

Una característica del 68HC11 es que proporciona dos entradas (VRL, VRH) de referencia para las conversiones. VRH se corresponde con el valor máximo y VRL con el valor mínimo. Las tensiones de referencia deberán ser fijadas por el usuario y estar en el rango (0v,6v) . **¡¡ cuidado con los límites pues aunque el *microcontrolador* permite superar Vcc es sumamente crítico con las tensiones negativas, es decir evitar introducir tensiones que estén por debajo de GND, tanto en el nivel de referencia VRL, como en la entrada del conversor. !!**

La placa CT6811 proporciona dos *jumpers* que permiten establecer los niveles de referencia a GND y VCC de tal manera que el usuario no tendrá que preocuparse por realizar los circuitos de referencia salvo que quiera tener otros niveles. El 68HC11 con formato de 48 pines sólo dispone de cuatro canales analógicos digitales, mientras que el de 52 pines dispone de 8 canales. La CT6811 proporciona los 8 canales.

El tiempo que tarda en realizarse una conversión es de 32 ciclos de reloj. Dicho reloj será tomado como el E cuando la frecuencia de éste sea mayor de 750K o será tomado de una señal interna generada por un circuito RC cuando la frecuencia de E sea menor de 750khz. El oscilador RC se activa situando a nivel alto el bit CSEL del registro OPTION.

Para tener información detallada sobre cómo se realiza la conversión interna y cuales son los circuitos equivalentes de entrada mirar el libro de referencia de motorola (capítulo 12).

SEÑAL DE RELOJ E**Figura 1: Secuencia de conversión****Registros del convertor A/D****A) OPTION (\$1039) :**

En este registro el bit más significativo es el **ADPU**. Este bit indica si se activa o no el convertor. Al ponerlo a nivel alto el *microcontrolador* entiende que se va a usar el convertor y empieza a cargar los condensadores utilizados para la conversión. Hay que esperar un tiempo pequeño hasta poder utilizar correctamente el convertor. En este registro hay otro bit (el número seis **CSEL**) que se encarga de seleccionar el reloj del convertor. Cuando la frecuencia del bus (reloj E) sea menor que 750Khz es recomendable utilizar otra frecuencia de reloj para la conversión. Esto se indica poniendo a nivel alto el bit **CSEL**. Al seleccionar esta opción el *microcontrolador* genera automáticamente una frecuencia propia desde un oscilador RC interno.(Esa frecuencia suele ser de 2Mhz).

Si el bit **CSEL** se deja a nivel bajo el *microcontrolador* trabaja con la frecuencia del bus. Este modo tiene dos ventajas sobre el anterior. La primera de ellas se refiere al ruido en la conversión, que es menor que con el oscilador interno. La segunda de ellas se refiere a la velocidad. Al conectar el circuito RC interno, el *microcontrolador* pierde un cierto tiempo en la sincronización de las lecturas y escrituras de la conversión.

7	6	5	4	3	2	1	0
ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0

Registro Option (\$1039)

B) ADCTL (\$1030) :

Al contrario que el registro anterior este es exclusivo del sistema convertor. De sus ocho bits sólo siete son válidos pues el bit número seis no tiene ninguna función.

7	6	5	4	3	2	1	0
CCF	0	SCAN	MULT	CD	CC	CB	CA

Registro ADCTL (\$1030)

CCF: Bandera de conversión completa.

Este es un bit de sólo lectura. Cuando se ha producido una conversión completa los valores de esta se guardan en una serie de registros. Cuando los datos han sido guardados correctamente en los registros, este bit se pone a uno indicando una conversión completa. Para ponerlo a cero hay que escribir sobre el registro **ADCTL**, y una vez hecho esto empieza inmediatamente otra conversión.

Cuando el conversor esta funcionando en modo continuo, la puesta a nivel bajo de este bit hace que se abandone la conversión que se realiza en ese momento para empezar otra nueva. Por el contrario, en modo discreto, es necesario poner a nivel bajo el bit para que se realice otra conversión.

SCAN: Control de conversión continua.

Cuando este bit está a nivel bajo, las cuatro conversiones pedidas se realizan una sola vez. Los resultados se graban en los registros y hasta que no se indique, no vuelve a hacer otra conversión. (Se indica poniendo a nivel bajo el bit CCF).

Cuando este bit está a uno las conversiones se realizan continuamente. Los registros son actualizados cada vez que llegan los nuevos datos.

MULT: Canal múltiple / Único canal.

Cuando este bit está a cero el sistema realiza cuatro conversiones consecutivas sobre el mismo canal, guardando los resultados en los registros **ADR1-ADR4**. El canal sobre el que se realiza la conversión se especifica mediante los bits CD-CA del registro **ADCTL**.

Cuando el bit está a nivel alto las conversiones se realizan una en cada canal y los resultados se guardan en los correspondientes registros. En esta modalidad los bits CD y CC se encargan de elegir el grupo de cuatro canales que será muestreado. Esta opción no esta disponible en el formato del *microcontrolador* de 48 pines, ya que sólo tiene cuatro canales analógicos en lugar de ocho.

CD,CC,CB,CA: Selección de canales.

Estos bits seleccionan los canales que van a ser operativos en las conversiones. Cuando se esta en modo múltiple los bits CD y CC son los únicos que tienen efecto. Se encargan de seleccionar el grupo de cuatro canales que se quiere muestrear. Esta opción no tiene ninguna validez en la versión del 68HC11 de 48 pines pues sólo existe el primer grupo de canales.

En modo único canal hay que seleccionar un canal de entre los disponibles. (Cuatro u ocho según las versiones). Para ello mirar la tabla que se adjunta a continuación. Dicho canal se selecciona con los bits CD,CC,CB y CA.

Como consejo práctico para leer la tabla se puede dejar siempre a nivel bajo el bit CD, cuando esta puesto a nivel alto se entra en los modos de funcionamiento especiales del *microcontrolador*, que no interesan. (Uno de esos modos es el factory testing del cual se puede encontrar más información en el manual de referencia de Motorola).

CD	CC	CB	CA	Señal del canal	Resultado en ADRx si MULT = 1
0	0	0	0	AD0 por bit 0 puerto E	ADR1
0	0	0	1	AD1 por bit 1 puerto E	ADR2
0	0	1	0	AD2 por bit 2 puerto E	ADR3
0	0	1	1	AD3 por bit 3 puerto E	ADR4
0	1	0	0	AD4 por bit 4 puerto E *	ADR1
0	1	0	1	AD5 por bit 5 puerto E *	ADR2
0	1	1	0	AD6 por bit 6 puerto E *	ADR3
0	1	1	1	AD7 por bit 7 puerto E *	ADR4
1	0	0	0	Reservado	ADR1
1	0	0	1	Reservado	ADR2
1	0	1	0	Reservado	ADR3
1	0	1	1	Reservado	ADR4
1	1	0	0	Vref hi**	ADR1
1	1	0	1	Vref low**	ADR2
1	1	1	0	Vref hi / 2**	ADR3
1	1	1	1	test**	ADR4

* : No disponible en la versión del microcontrolador de 48 pines.

** : Selección para el 'factory testing'

C) ADR4-ADR1 (\$1034-\$1031) :

En estos registros se guardan los resultados de ocho bits de la conversión. Los registros son de lectura única y una vez rellenados con los datos válidos se activa la bandera CCF para indicar conversión finalizada. En el modo continuo esta bandera no tiene efecto, aunque la puesta a cero de este bit interrumpiría la conversión en curso para comenzar una nueva.

7	6	5	4	3	2	1	0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Registro ADRx (\$103x)

Los registros **ADR1**, **ADR2**, **ADR3** y **ADR4** guardan los resultados; el bit más significativo del registro coincide con el bit más significativo de la conversión. La situación en el mapa de memoria de cada registro es respectivamente \$1031, \$1032, \$1033, y \$1034. Aunque hay ocho canales solo hay cuatro registros para guardar los datos, esto hay que tenerlo muy en cuenta, pues no se pueden muestrear ocho canales a la vez. Esta característica hace que la diferencia entre el formato de 48 pines y el de 52 pines no sea tan grande.

La lógica de control de los circuitos del sistema hace que no se produzcan interferencias entre las lecturas por parte de los usuarios y las escrituras del *microcontrolador*.

Programa ejemplo de manejo del conversor A/D

En este apartado se muestra un programa que sirve de ejemplo para la programación del conversor. Se supone que se ha conectado un potenciómetro como se indicó en el apartado 3.5. El programa lee el valor que registra la entrada analógica uno y la compara con 2.5 voltios. Si es mayor enciende el LED de la placa, en caso contrario lo apaga.

Se supone que se trabaja con la CT6811 que ya lleva incorporado el LED, y que sitúa los niveles de referencia a VCC y GND cuando están los *jumper* JP1 y JP2 conectados.

El canal analógico por el cual se introduce la señal del potenciómetro es el primero, que internamente se corresponde con el **PE0** (Primera entrada analógica del *microcontrolador*), el resultado de la conversión se obtendrá en **ADR1**. Se realiza una conversión continua de manera que cuando finaliza una conversión empieza inmediatamente la siguiente.

```

; +-----+
; | PAD.ASM.   (C) GRUPO J&J. MARZO 1997 |
; +-----+
; | Programa ejemplo para ser ejecutado en la tarjeta CT6811. |
; | Este programa se debe cargar en la RAM interna del 6811. |
; |
; | Ejemplo de utilización del canal 0 del conversor A/D. Cuando la |
; | tensión supera los 2.5 voltios se enciende el led de la placa. |
; +-----+

; Usaremos como entrada analógica el canal 1, es decir la
; entrada PE0. Los niveles de referencia (VRH, VRL) serán (Vcc,
; GND) respectivamente. Es decir los jumpers JP1 y JP2 de la
; CT6811 deben estar conectados.

        ORG $0000

OPTION  EQU $39
ADCTL   EQU $30
PORTA   EQU $00
ADR1    EQU $31

INICIO
        LDX  #$1000
        LDAA #$80
        STAA OPTION,X ; encender el conversor
        LDAA #$20      ; configuración conversor:
        STAA ADCTL,X   ; SCAN -> activo
                        ; MULT -> inactivo
                        ; ADR1 -> seleccionar primer canal

sigue   BRCLR ADCTL,X $80 sigue ; espera a que termine conversión
        LDAA  ADR1,X          ; leer el resultado de la conversión
        CMPA  #$7F            ; comparar con la mitad (127 en decimal)
                                ; que corresponde a 2.5v de entrada.
        BLO   menor           ; si es menor apagar el led
        LDAA  #$40            ; No--> encender el led
        STAA  PORTA,X
        BRA   sigue           ; Realizar la siguiente conversión

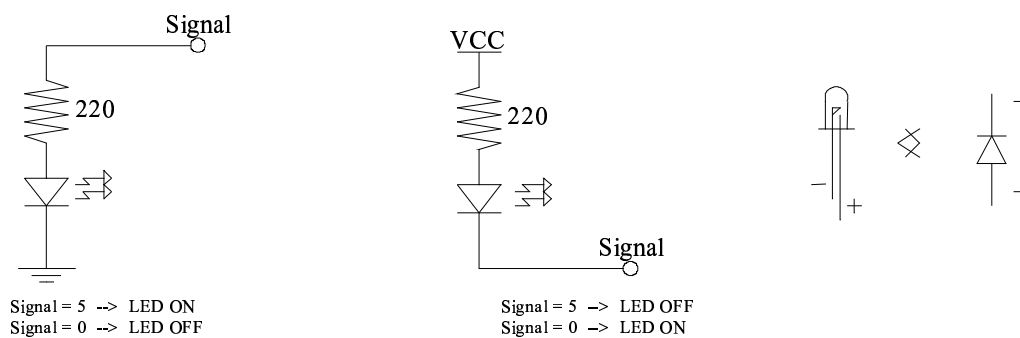
menor   CLRA
        STAA  PORTA,X          ; Apagar el led.
        BRA   sigue           ; Realizar la siguiente conversión

        END

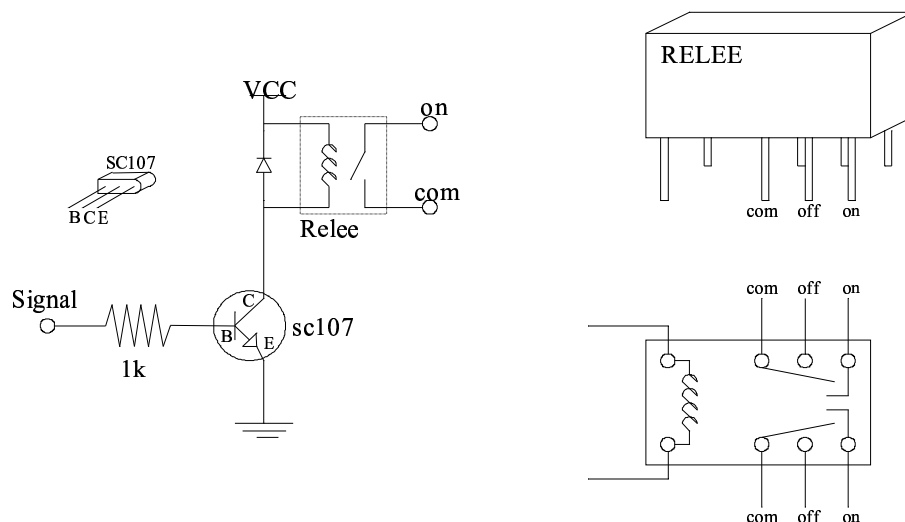
```

6. CONEXIÓN DE OTROS ELEMENTOS.

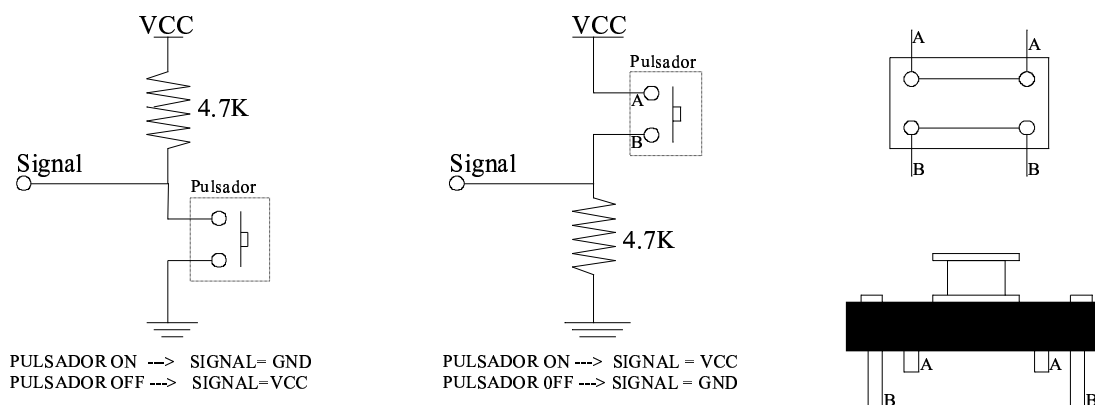
CONEXIÓN DE UN LED A UNA SALIDA DIGITAL



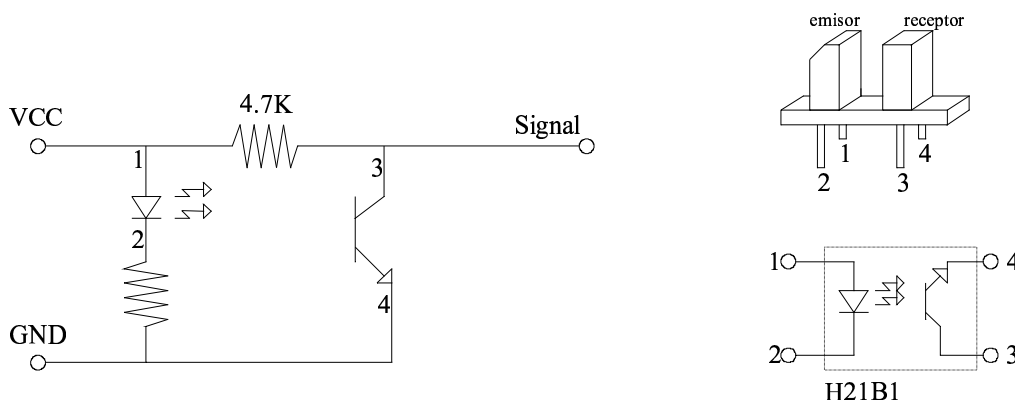
CONEXIÓN DE UN RELEE A UNA SALIDA DIGITAL



CONEXIÓN DE UN PULSADOR A UNA ENTRADA DIGITAL



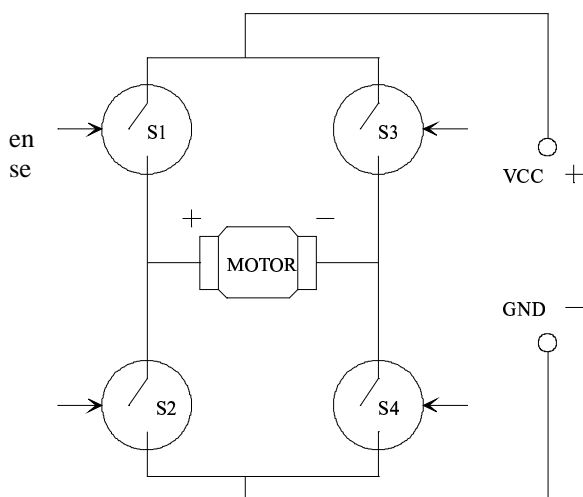
CONEXIÓN DE UN OPTOACOPLADOR



APÉNDICE A: EL CHIP L293B

El chip L293B es un driver de potencia utilizado mucho en sistemas con motores paso a paso. En la CT293+ se ha utilizado como driver de potencia para motores de continua. La gran característica de este chip es la posibilidad de implementar el puente en H necesario para manejar un motor que tiene que ser capaz de girar en los dos sentidos. Otra característica es la separación de la tensión de alimentación de la pastilla propia (TTL) y la tensión de alimentación destinada a los motores. Es una forma de controlar con tensiones TTL motores que están alimentados a mayor tensión, por ejemplo 12v.

El 293 permite alimentaciones para los motores de hasta +36v y una corriente de salida de 1Amperio. Cuando se use en condiciones extremas probablemente sea necesario ponerle un disipador.



En la figura A.1 se ve el puente en H que se ha mencionado antes. Se comprueba que según se activen los *Interruptores* se consigue que el motor gire en uno u otro sentido. Este circuito parece sencillo de hacer pero cuando se pretende implementar la realidad, empiezan a aparecer pequeños problemas. Por eso recomienda usar el chip L293B. Además se consigue ahorrar espacio, cosa que nunca viene mal.

Figura A.1 Puente en H

El patillaje del L293B se describe en la figura A.2. Se aprecia que hay cuatro pines con la indicación GND. Estos pines se recomienda que se cortocircuiten entre si para tener una masa única en el circuito.

El chip tiene dos alimentaciones distintas, como ya se mencionó antes. Una de ellas proporciona la alimentación TTL y la otra proporciona la alimentación de los motores. En el caso de querer utilizar una única alimentación para el sistema se tienen que cortocircuitar las dos entradas de alimentación. Los microbots implementados por el Grupo J&J tienen esta característica, una sola fuente de alimentación, y su funcionamiento ha sido excelente. Cuando se realice lo anterior tener presente lo siguiente:

Al utilizar una sola alimentación tener en cuenta :

1) Si los motores introducen caídas bruscas de tensión por debajo de los 4.5v conviene desconectar el LVI para que no se 'resetea' la placa automáticamente. Si se está utilizando la CT6811 para hacer lo anterior hay que quitar el jumper JP6. Cuando se quita el LVI la EEPROM queda desprotegida ante las caídas de tensión, es decir puede darse el caso de que se des programe la EEPROM interna.

2) Cuando se utiliza la CT293+, para unir las alimentaciones habrá que colocar el jumper JP1 en su sitio. Cuando se realice esta operación verificar que la alimentación externa de los motores este desconectada, ya que si no se producirá un cortocircuito entre esta y la alimentación TTL.

SWITCH S1	SWITCH S3	SWITCH S2	SWITCH S4	MOTOR
ON	OFF	OFF	ON	IZQUIERDA
OFF	ON	ON	OFF	DERECHA
ON	ON	OFF	OFF	PARADO
OFF	OFF	ON	ON	PARADO
OFF	OFF	OFF	OFF	PARADO

Tabla explicativa del puente en H

El chip se maneja muy fácilmente. Hay un conjunto de cuatro drivers de potencia dos en cada lado del chip. Cada pareja tiene su CE (chip enable) correspondiente, de tal manera que se puede activar o desactivar cada pareja por

separado. Cuando se introduce un uno lógico a la entrada del driver a su salida se obtiene un valor de tensión correspondiente a la VccMotor, es decir el valor de tensión de la alimentación de los motores.

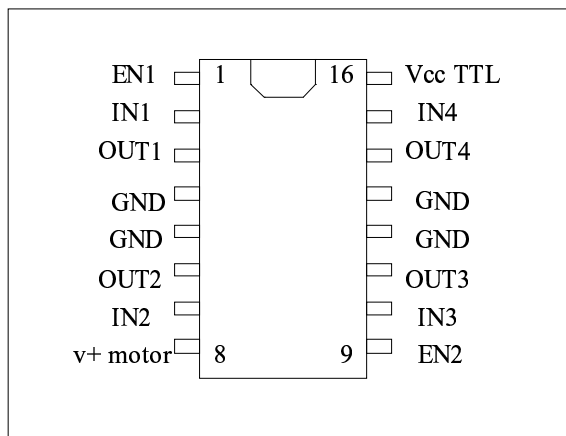


Figura A.2. Patillaje de L293B

Dependiendo de la aplicación se necesitan uno o dos drivers para controlar un motor. Se utiliza un solo driver cuando el control sea hacer girar un motor en un sentido fijo o dejarlo parado. En este caso una de las patas del motor se deja fija a Vcc o a GND y se actúa con el driver sobre la otra pata. Se utilizan dos drivers cuando se quiera controlar el estado del motor (apagado/encendido) y el sentido de giro (izquierda/derecha). En este caso cada pata del motor se conecta a un driver y según sean los valores de entrada de los drivers se tiene un comportamiento u otro.

Se supone que se quiere controlar el estado y sentido de giro de un motor, por lo tanto se utilizan dos drivers por motor, por lo que una pastilla L293B es capaz de controlar dos motores. Ahora bien se pueden implementar varias formas de realizar ese control, se proponen dos de ellas. La primera configuración tiene la ventaja de no necesitar ninguna lógica adicional, la segunda de ellas es más intuitiva y cómoda a la hora de programar pero necesita de lógica adicional.

A.1. Sistema con mínimos recursos

El sistema con mínimos recursos sólo necesita el L293B para realizar el interfaz con los motores, el esquema de conexión se proporciona en la figura A.3. y la forma de controlar el circuito en la tabla siguiente. Este circuito tiene la ventaja de ahorrar electrónica, pero no es el más indicado para realizar un buen control.

Una propiedad interesante del esquema es su transportabilidad a otros sistemas de motores, se puede usar para manejar motores paso a paso, pues en realidad cada par entrada-salida es independiente de sus compañeras, como se aprecia en la figura.

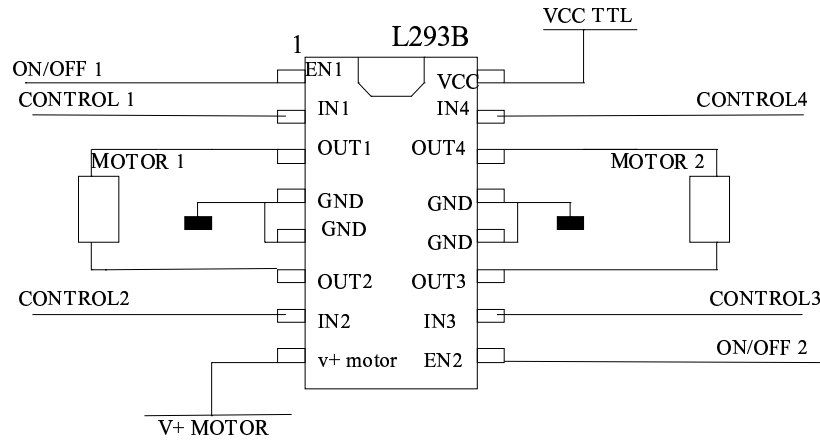


Figura A.3. Esquema reducido de la interfaz con los motores

Para simplificar el circuito se dejan las entradas ON/OFF de los motores puestas a ON, de tal manera que para apagar un motor se introducen por sus dos drivers correspondientes valores de entrada iguales, de esa forma no cae tensión en el motor y este no anda. Se ahorra dos cables pero en contra se malgasta energía cuando los motores no tengan que girar.

CONTROL 1	CONTROL 2	ON / OFF 1	ESTADO MOTOR1
ON	ON	ON	PARADO
ON	OFF	ON	DERECHA
OFF	ON	ON	IZQUIERDA
OFF	OFF	ON	PARADO
X	X	OFF	PARADO

Tabla de control del Motor 1. (para el motor 2 es análoga a esta)

A.2. Sistema intuitivo o con recursos máximos.

Esta forma se llama intuitiva porque facilita mucho los sistemas de control de motores, permite implementar un sistema de PWM para el control de velocidad. Otras ventajas son el ahorro de energía al utilizar los chip entable, y que la conexión con el controlador se sigue realizando con cuatro hilos.

El inconveniente mayor es la necesidad de introducir un poco más de lógica TTL. A pesar de esto, la CT293+ lleva este tipo de circuito. La figura A.4 pretende ilustrar el esquema (donde los triángulos significan puertas inversoras), y en la tabla siguiente un modelo de control para un motor.

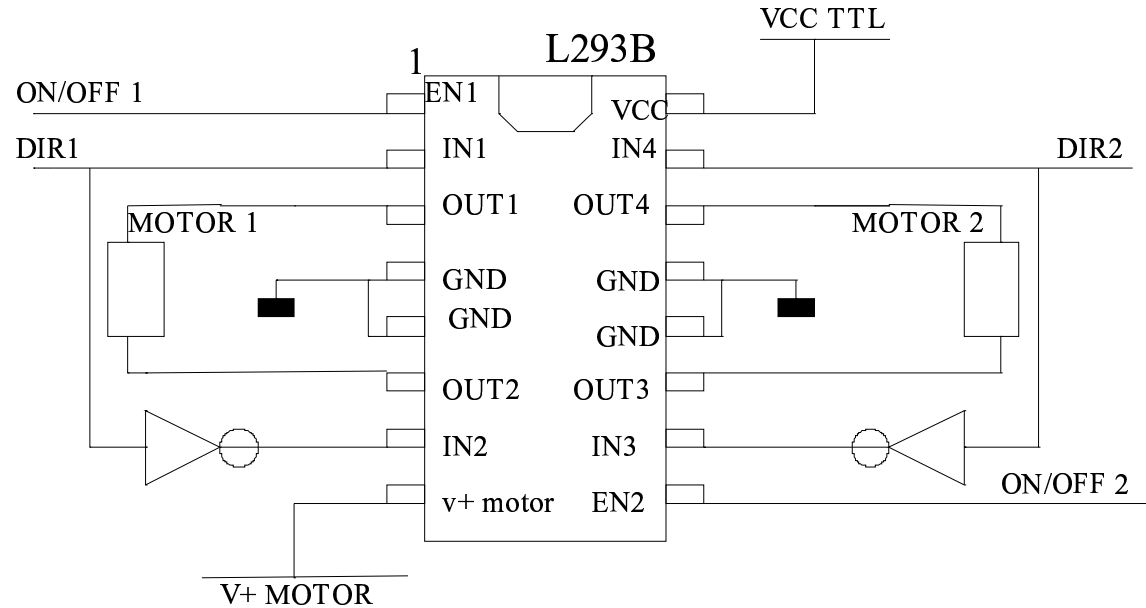


Figura A.4. Esquema de la interfaz máxima o intuitiva.

DIRECCIÓN 1	ON / OFF 1	ESTADO MOTOR 1
OFF	ON	IZQUIERDA
ON	ON	DERECHA
X	OFF	PARADO

Tabla de control del motor 1 en el modo máximo o intuitivo (para el motor 2 es análoga)

Se ha mencionado más arriba que con este modelo se puede implementar fácilmente el PWM. Se va a indicar brevemente como hacer esto. Para simplificar se analiza el motor 1, sabiendo que para el dos se hace igual. Se supone que se tiene un controlador externo al cual se le han conectado las entradas ‘dirección’ y ‘ON/OFF’ de la interfaz del motor . Con un bit se controla el sentido de giro y con el otro bit se controla la activación del motor. Es este bit el que hace el PWM . Dejando activo el bit el motor gira a la velocidad máxima para una tensión de alimentación dada, desactivando el bit el motor no gira, tiene entonces la mínima velocidad. El PWM consiste en grosso modo en introducir una señal cuadrada por esta entrada, activando y desactivando periódicamente el bit ON/OFF. Con esto se consigue que la velocidad del motor se sitúe entre la mínima y la máxima. La razón se encuentra en que los motores de continua responden (faltando un poco al rigor) al valor medio de la señal que se introduce entre sus conectores de alimentación. La velocidad depende por tanto del ciclo de trabajo de la señal. Según sea el valor de dicho ciclo, se tiene mayor o menor valor medio de tensión y por tanto el motor ve también mayor o menor tensión a su entrada. Lo que se corresponde con una mayor o menor velocidad.

APÉNDICE B: EL SENSOR CNY70.

Generalmente los sensores de infrarrojos se usan en sistemas de transmisión de información (control remoto) y en sistemas de detección de señales (alarmas, barreras,...). Se encuentran ejemplos de su uso en la mayoría de los electrodomésticos. Por ejemplo en los mandos a distancia, en los cascos inalámbricos, en células en las puertas automáticas, en los videos,...

Según las necesidades de la aplicación el rayo infrarrojo generado por el emisor tiene que recorrer más o menos distancia hasta llegar al receptor. Esto determina cómo se va a polarizar al sensor. Ocurre que cuando se quiere enviar información a larga distancia se necesita realizar una modulación, lo cuál termina siendo una tarea complicada. En otros casos esto no ocurre , el sensor esta situado a unos pocos milímetros del objeto a detectar y no necesita de una polarización complicada. Por ejemplo se pueden utilizar para distinguir una línea negra situada sobre un fondo blanco. Para esta aplicación de corta distancia basta con realizar una sencilla polarización en continua.

El CNY70 es un sensor que lleva incorporados en su encapsulado un emisor y un receptor de infrarrojos. Ambas partes están situadas en paralelo, si se quiere recibir el rayo enviado por el emisor se debe producir una reflexión en algún objeto.

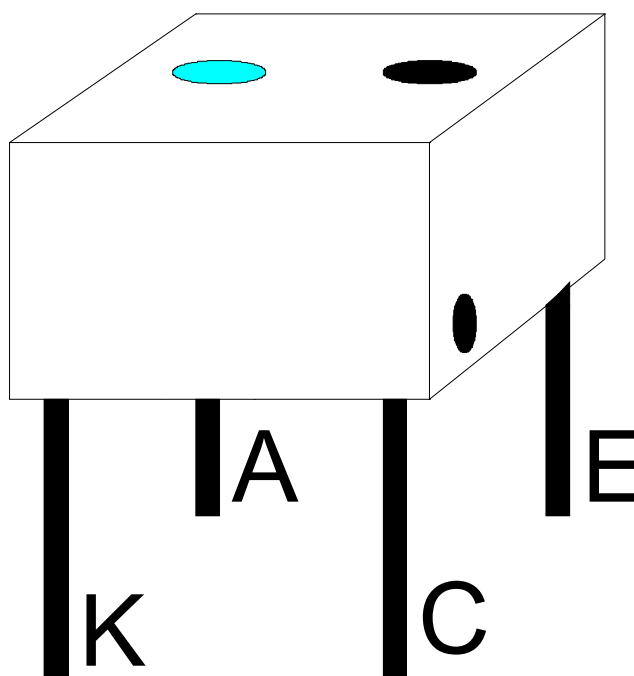


Figura B.1. Esquema del sensor CNY70.

- A -> Ánodo del emisor**
- K-> Cátodo del emisor**
- E -> Emisor del receptor**
- C-> Colector del receptor**

El emisor compuesto por ánodo y cátodo es un diodo led de infrarrojo, mientras el receptor compuesto por el emisor y el colector es un fototransistor de luz infrarroja.

A continuación se proponen dos tipos de circuitos para polarizar en continua o con modulación. En el primero (P. Continua) si se usa el CNY70, pero en el segundo (P. Modulada) se utiliza un emisor y un receptor independientes pues el CNY70 no es el más indicado.

B.1. POLARIZACIÓN EN CONTINUA

Al polarizarse en continua su distancia efectiva se reduce bastante, sólo se pueden usar para distinguir objetos muy próximos al sensor, por ejemplo una línea negra sobre un fondo blanco. La principal ventaja es la facilidad de uso y simplicidad del circuito.

La CT293+ utiliza este tipo de polarización, los circuitos que se proporcionan sólo sirven para aplicaciones de corta distancia, como por ejemplo encoders, seguimiento de líneas negras y detección de objeto inmediato.

El circuito de polarización se presenta en la figura B.2. El potenciómetro sirve para regular la distancia y el funcionamiento, pero por simplificar la placa se ha sustituido por una resistencia de 47K ohmios. El chip 40106 es un inversor c-mos, que sirve para adaptar la lectura del sensor al controlador digital, haciendo función de discretizador. De

todo el rango de tensiones que ofrece el sensor al aproximarse al objeto a detectar, solamente se obtienen dos valores: Esta cerca o esta lejos. La resistencia de 220 ohmios se usa para polarizar adecuadamente el LED emisor de infrarrojo.

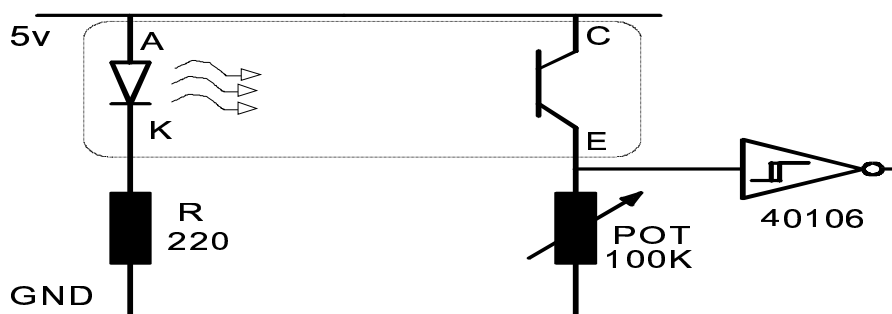


Figura B.2. Esquema de polarización en continua del sensor CNY70

B.2. POLARIZACIÓN MODULADA.

De esta forma se consigue que el sensor detecte obstáculos lejanos pero se necesita añadir bastante circuitería adicional. La forma de polarización es distinta a la anterior en todo, para empezar el emisor no está conectado a Vcc sino que lo está a una línea de control. Esta línea de control, cuando se indique, emite una señal cuadrada de unos 40KHZ. El receptor es como el de antes pero a la salida de la pata E, hay que colocar un filtro para quedarse con las señales de 40khz, este filtro puede ser óptico, un amplificador operacional como filtro paso alto o un circuito resonante L-C. Después del filtro casi seguro que se tiene que poner un amplificador para obtener un nivel adecuado de tensión.

El funcionamiento de este tipo de circuitos consiste en mandar una ráfaga de pulsos, y luego esperar a recibirlos. Si en esta espera se cuenta el tiempo transcurrido se puede saber la distancia de separación con el objeto detectado.

Realmente para realizar este tipo de sistemas ya no se usa el CNY70 sino que es preferible usar emisores y receptores separados. El emisor suele ser barato y fácil de implementar mientras que el receptor puede ser un circuito específico que incorpore las funciones de filtrado y amplificación.

Un esquema ofrecido por el MIT se ofrece a continuación, aunque los resultados obtenidos por los autores sólo sirven para unos pocos centímetros, rara vez se llega al medio metro con efectividad alta. Este esquema utiliza como emisor un LED de Siemens (SFH484) y como receptor el G1U52X de Sharp.

El circuito propuesto utiliza dos emisores independientes modulados con un circuito común que puede ser ajustado con un potenciómetro de 5K ohmios. El circuito receptor se ha simplificado mucho al utilizarse un dispositivo especial de Sharp. Este dispositivo integra el receptor de infrarrojo, los filtros y los amplificadores necesarios para obtener una buena señal.

Para aquellas personas que estén interesadas en la localización de objetos a larga distancia, utilidad práctica de este circuito, se advierte que los sistemas de infrarrojo tienen un escaso ángulo de captación de señal, motivo por el cual el receptor y el foco de la reflexión deben estar prácticamente enfrentados. Esto se puede evitar colocando lentes, pero puede ser bastante complicado. Se aconseja no olvidar los sistemas de ultrasonidos que aunque tienen el inconveniente de necesitar más energía para emitir la señal luego facilitan la recepción de esta.

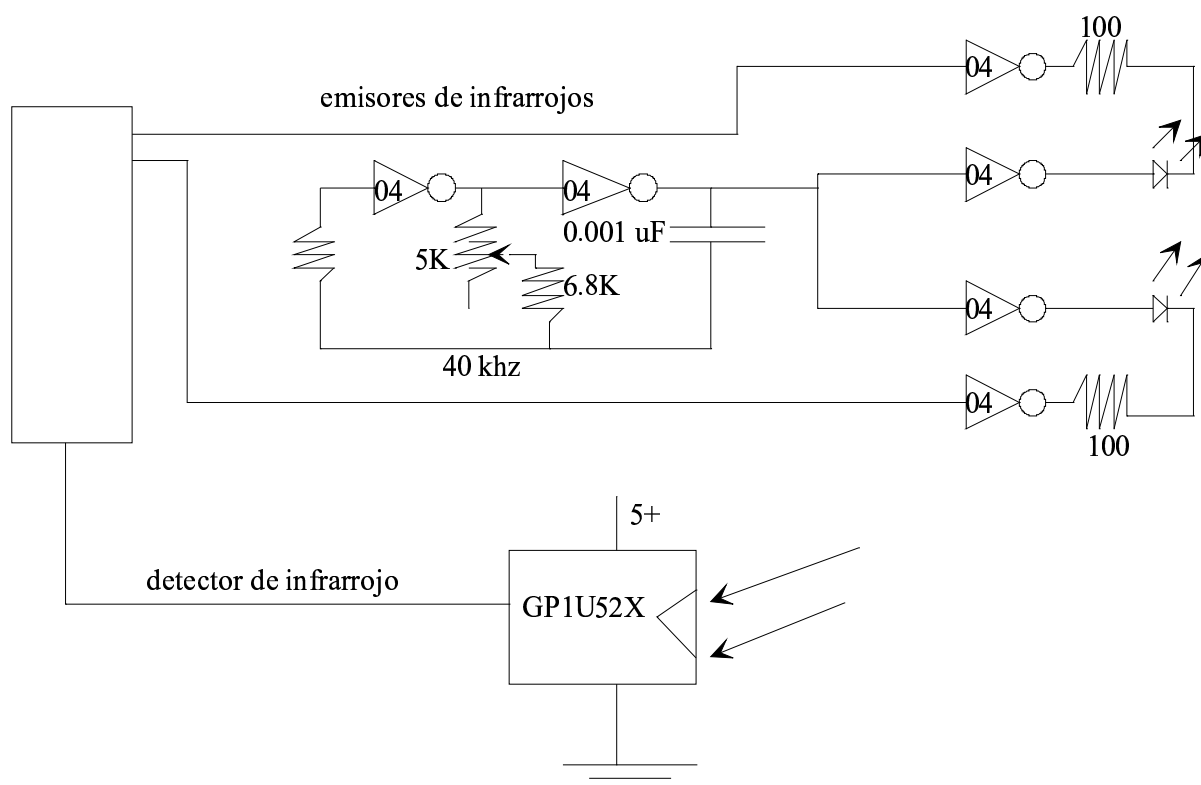


Figura B.3. Esquema proporcionado por el MIT para sensores larga distancia

APÉNDICE C: LOS MOTORES PASO A PASO.

C.1. CARACTERÍSTICAS DE LOS MOTORES PASO A PASO.

Los motores paso_paso o también llamados 'stepper motor' se utilizan en aplicaciones en las que se requiera girar y detener en ciertas posiciones el eje de un motor con una precisión de una centésima de milímetro. En un motor de corriente continua normal (motor CC) para obtener esta condición es necesario tener un circuito de realimentación más o menos complicado. Aplicaciones donde se suelen usar este tipo de motores son impresoras, plotters, robots, unidades de disco, etc.

En general el eje de un motor CC se puede detener en cualquier ángulo de giro, mientras que en uno PP (paso_paso) sólo puede situarse en unos determinados. En concreto en ' $q = q^{\circ} + n \cdot Dq$ ', donde 'n' se corresponde al número de pasos o 'steps' dados desde la posición de partida inicial ' q° ' y ' Dq ' es el incremento de ángulo que se produce en un paso. En estos motores PP el eje se detiene y se mantiene bloqueado electromagnéticamente hasta que no se de una nueva orden para hacerlo girar, se puede decir que realiza un giro discreto en comparación del que realizan los motores de corriente continua.

Un estándar es encontrar motores PP con un $Dq=7.5^{\circ}$, en ellos para completar una vuelta se tienen que dar $360^{\circ} / 7.5^{\circ} = 48$ pasos. Otros tienen $Dq=1.8^{\circ}$ por lo que tienen que dar 200 pasos. Estos últimos tienen más resolución que los primeros.

Los mayores inconvenientes de estos motores PP son la necesidad de utilizar un circuito especial para gobernarlos, su consumo, peso, vibraciones, y que generalmente tienen menor velocidad que uno de continua. La ventaja, como ya se ha dicho, está en el ahorro del circuito de reglamentación necesario para aplicaciones de control donde se quiera posesionar el eje de un motor de una manera rápida y estable. El par de salida además suele ser más alto que el de un motor de continua sin reductora.

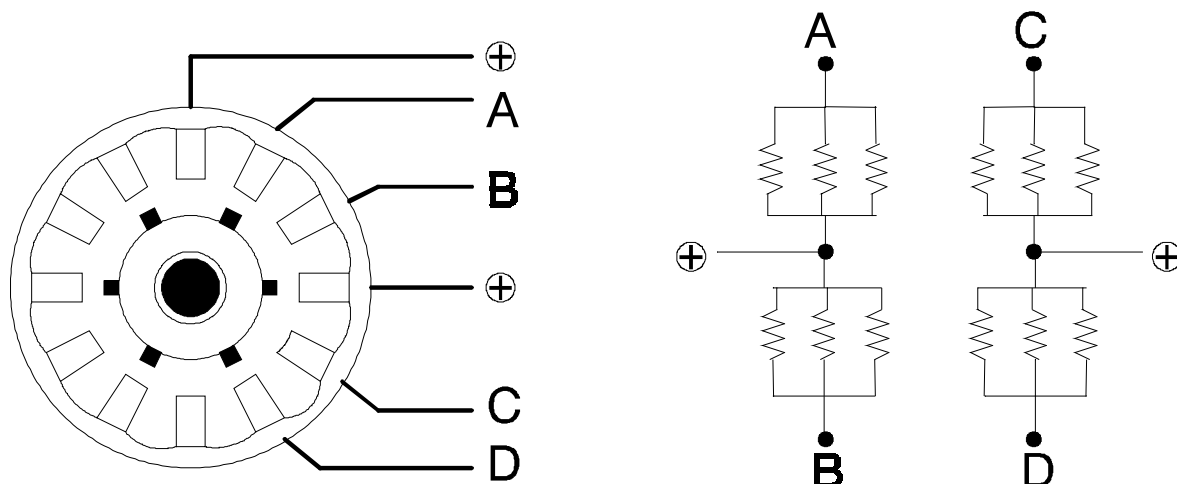


Figura C.1.1. Composición interna de un motor paso_paso.

Internamente estos motores están constituidos por un rotor (parte móvil) sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator (parte fija de un motor).

Al exterior salen unos cables que se corresponden con los extremos de las bobinas excitadoras. Según el tipo de motor si tienen 6 o 4 cables en el exterior. Un motor PP con seis cables se llama UNIPOLAR y uno con cuatro BIPOLAR. Los más comunes son los unipolares debido a que el circuito de control se simplifica mucho. Los bipolares se utilizan en aplicaciones especiales cumpliéndose lo siguiente: generalmente a igualdad de tamaño, par de salida, y número de pasos el bipolar consumirá la mitad. Si se iguala par de salida, consumo y número de pasos entonces el unipolar será el doble de grande, por consiguiente pesará el doble.

A pesar de estas diferencias los cables de control son siempre cuatro (A,B,C,D) y la secuencia de códigos que hay que introducir en estos cables es igual.

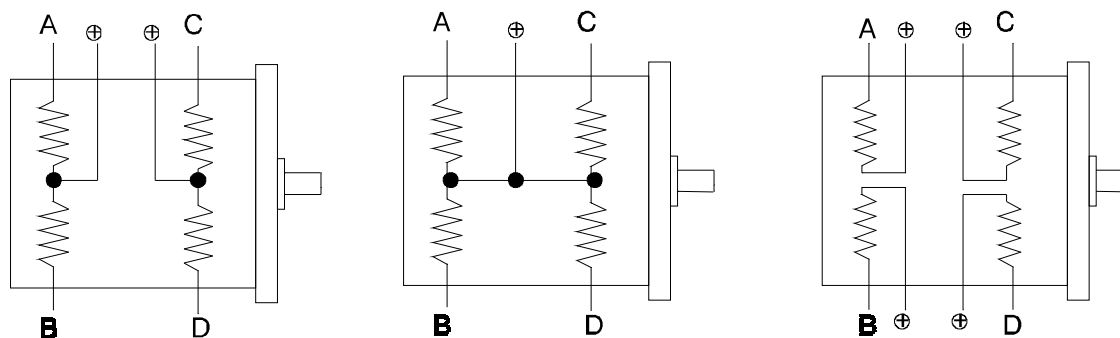


Figura C.1.2. Diferentes modelos de motores unipolares.

La figura C.1.2. muestra diferentes tipos de motores PP unipolares. La característica común es la realización de un contacto interno en el medio de las bobinas excitadoras. Lo más usual es unir todos estos cables y derivarlos a VCC. En la figura se indica con el símbolo '+'. Uniéndolos a GND el motor gira en sentido contrario, aunque esto último también se puede hacer cambiando el orden de excitación de las bobinas. Los demás cables (A,B,C,D) son de control. A este tipo de motor se le llama unipolar porque en los cables de control solamente hace falta introducir un tipo de alimentación, en el caso de la figura C.1.2. se introducirá GND ya que el punto común está a VCC.

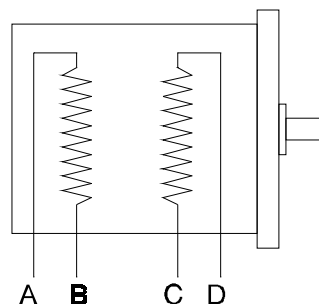


Figura C.1.3. Motor bipolar

La figura C.1.3 indica como es la estructura interna de un motor PP bipolar. En este tipo no hay cable intermedio, tan solo los de control, esta característica hace que se complique su circuito de control. Será necesario que los cables de control (A,B,C,D) puedan dar VCC y GND, es decir las dos polaridades de alimentación.

El fabricante suele especificar en alguna parte del motor los datos más importantes para su manejo. Generalmente se indica la tensión de alimentación y los grados de rotación por cada paso. 'Dq'. Si se quiere saber la corriente que circula por cada bobina se puede recurrir a la siguiente regla práctica. Medir la resistencia óhmica de una bobina y sabiendo la tensión de alimentación aplicar la ley de Ohm. ($I = V / R$). Una vez conocido este dato buscar unos circuitos excitadores de las bobinas capaces de suministrar una corriente superior a la calculada.

C.2. CONTROL DE MOTORES PASO A PASO.

Se indica de forma gráfica los circuitos necesarios para controlar un motor PP de tipo unipolar y bipolar. En el caso unipolar los cables comunes se conectan a VCC, que es lo más corriente. Si se quieren conectar a GND se tiene que cambiar el circuito.

En los diagramas aparece el símbolo de la figura C.2.1. Este representa un interruptor controlable desde la entrada de control 'C'. Cuando por ella se introduce un '1' el interruptor se cierra permitiendo el paso de corriente de un extremo al otro. Cuando se introduce un '0' se abre y no permite la circulación de corriente.

Este dispositivo ideal se puede implementar con transistores o con relees, aunque realmente se suelen usar circuitos integrados que ocupan menos espacio y que incorporan otros circuitos para facilitar la utilización de los motores. Ejemplo de esto son los chips L293B y SAA1027.

Figura C.2.1.

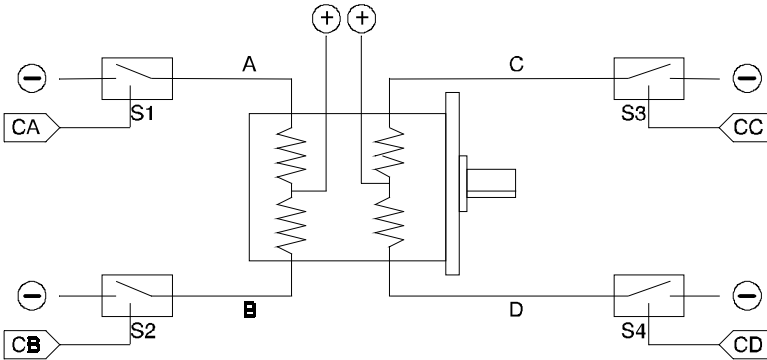
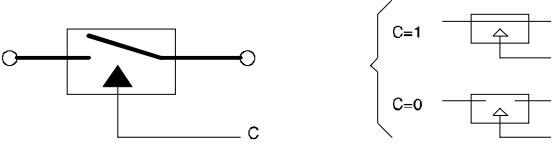


Figura C.2.2. Conexión de un motor unipolar.

En el motor unipolar solamente hacen falta cuatro interruptores. En la siguiente tabla se aprecia cuál es la secuencia que hay que seguir para hacer girar al motor. Si se recorre la tabla según 1,2,3,4,1,2,... el motor gira en un sentido, si se recorre según 4,3,2,1,4,3,... lo hace en el otro. Esta secuencia tiene su explicación pero aquí no se entra en ella. Confórmese el lector con saber que para mover el eje se van activando unas bobinas y desactivando otras, de forma que los imanes permanentes que hay en el rotor tienden a situarse en una posición estable. Al poner el código del siguiente paso las bobinas cambian de estado y otra vez el rotor tiende a buscar su posición estable. Si no se respeta el orden y se saltan pasos el movimiento es deficiente. En cada repetición de una secuencia se producen cuatro pasos.

	A	B	C	D	S1	S2	S3	S4	CA	CB	CC	CD
1	-		-		ON	OFF	ON	OFF	1	0	1	0
2		-	-		OFF	ON	ON	OFF	0	1	1	0
3		-		-	OFF	ON	OFF	ON	0	1	0	1
4	-			-	ON	OFF	OFF	ON	1	0	0	1

En esta tabla primero se indica que señal se debe meter a cada cable A,B,C,D. Una casilla en blanco indica que esa entrada debe estar desconectada. Por eso no se pone el signo de la alimentación. Luego se muestra cual debe ser el estado de cada interruptor para producir la situación anterior. Finalmente se indica el valor de las señales de control que manejan los interruptores. Se ve como en los motores unipolares solamente hace falta introducir tensiones negativas por los cables A,B,C,D. Este circuito se implementa con cuatro transistores de potencia o simplemente con cuatro relees.

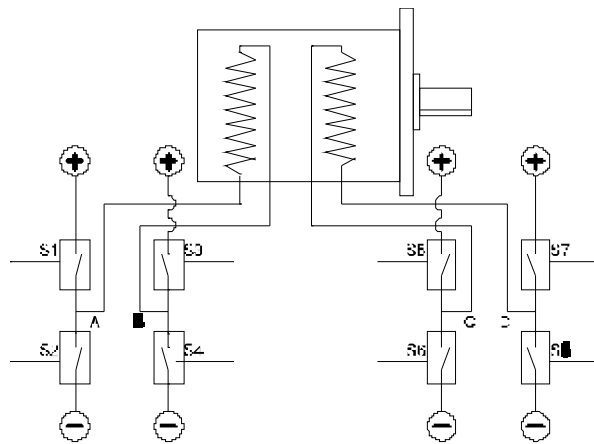


Figura C.2.3. Conexión de un motor bipolar.

En el motor bipolar hacen falta ocho interruptores, por eso se dice que su circuito de control es bastante más complicado. la interpretación de la tabla que aparece a continuación es la misma que para los motores unipolares.

	A	B	C	D	S1	S2	S3	S4	S5	S6	S7	S8	CA	CB	CC	CD
1	-	+	-	+	OFF	ON	ON	OFF	OFF	ON	ON	OFF	1	0	1	0
2	+	-	-	+	ON	OFF	OFF	ON	OFF	ON	ON	OFF	0	1	1	0
3	+	-	+	-	ON	OFF	OFF	ON	ON	OFF	OFF	ON	0	1	0	1
4	-	+	+	-	OFF	ON	ON	OFF	ON	OFF	OFF	ON	1	0	0	1

En la tabla primero se indica que señal se debe introducir en cada cable A,B,C,D. Luego se muestra cual debe ser el estado de cada interruptor para producir la situación anterior. Finalmente se ven las cuatro señales de control que gobiernan los interruptores. Ahora el lector puede preguntarse porqué sólo hay cuatro señales de control en lugar de ocho. Analizando con más cuidado la tabla se ve que las parejas (S1,S2), (S3,S4), (S5,S6), (S7,S8), siempre son inversas, introduciendo cuatro inversores se evitan cuatro señales de control.(Ver figura C.2.4.). Al final se tiene la misma secuencia que la utilizada con los motores unipolares pero a costa de complicar el circuito de control.

También se aprecia como en los motores bipolares hace falta introducir tanto tensiones negativas como positivas por los cables de control, esa es la razón de su nombre.

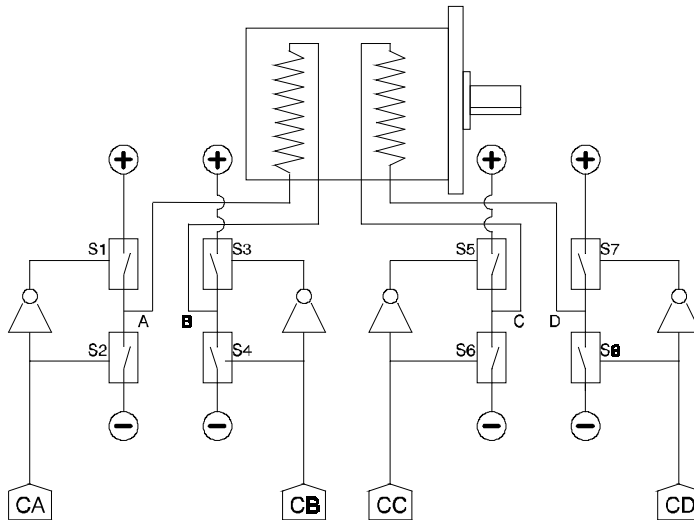


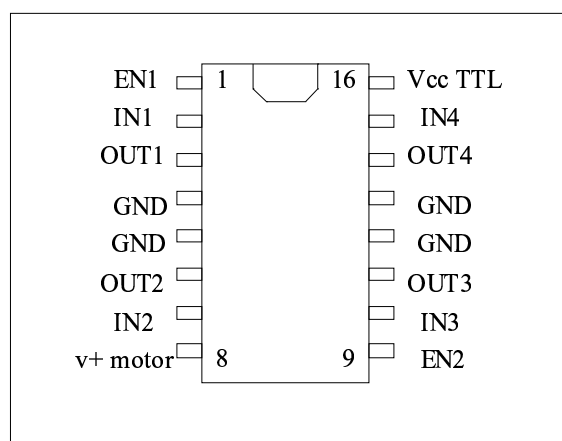
Figura C.2.4. Circuito final de polarización Motor Bipolar.

Por suerte ya no es necesario hacerse estos circuitos con elementos discretos como son los transistores, relees, etc, existen en el mercado numerosos dispositivos integrados que contienen estos circuitos y hacen transparente al diseñador y al programador todo lo anterior.

Se proponen a continuación dos esquemas de montaje con dos circuitos integrados diferentes. Se conectan los cables A,B,C y D del motor PP a las salidas correspondientes del circuito integrado, independientemente de que se este controlando un motor PP unipolar o uno bipolar. Las señales CA, CB, CC y CD se conectan a las entradas correspondientes del chip y se busca que el control respete la secuencia de la tabla siguiente.

	CA	CB	CC	CD
1	1	0	1	0
2	0	1	1	0
3	0	1	0	1
4	1	0	0	1

C.3. ESQUEMA DE CONEXIÓN DE UN MOTOR PASO_PASO BIPOLAR CON EL CHIP L293B.



El circuito integrado L293B que se ha descrito en el apéndice A se va a utilizar como driver de potencia para un motor paso_paso bipolar. También se puede utilizar para un unipolar conectando los cables de alimentación VCC del motor a la tensión V+ motor y dejar el resto del circuito igual.

El circuito se presenta simplificado, se han anulado los pines de ENABLE dejándolos siempre activos. En algunas ocasiones es necesario proteger los devanados del motor con una red de diodos o con resistencias limitadoras de corriente. En este circuito se han obviado estos detalles, pero no por ello deja de funcionar.

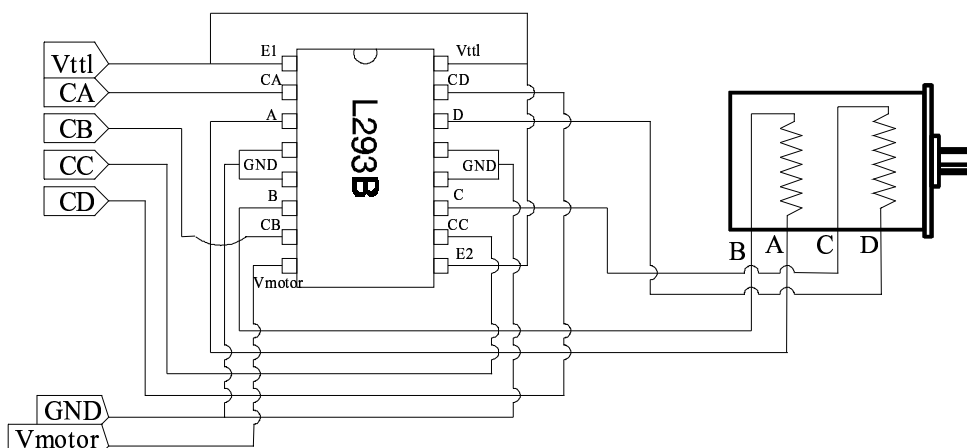
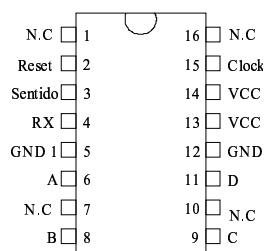


Figura C.3.1. Motor bipolar polarizado con el L293B

	CA	CB	CC	CD
1	1	0	1	0
2	0	1	1	0
3	0	1	0	1
4	1	0	0	1

Recordatorio de la secuencia de control

C.4. ESQUEMA DE CONEXIÓN DE UN MOTOR PASO_PASO UNIPOLAR CON EL CHIP SAA1027.



El circuito integrado SAA1027 de Philips genera automáticamente las señales de control. Se le indica por el pin 3 cual es el sentido de giro deseado y por el pin 15 cuando se quiere dar un paso. El automáticamente genera la secuencia necesaria para dar dicho paso. El pin 2 es de *reset*, si a su entrada se introduce un nivel bajo el circuito se bloquea y no responde a ninguna entrada.

La tensión de alimentación está comprendida entre 9.5v y 18 voltios, generalmente se maneja a 12v. Los transistores que hay en su interior son capaces de suministrar una corriente máxima de unos 0.5 Amperios.

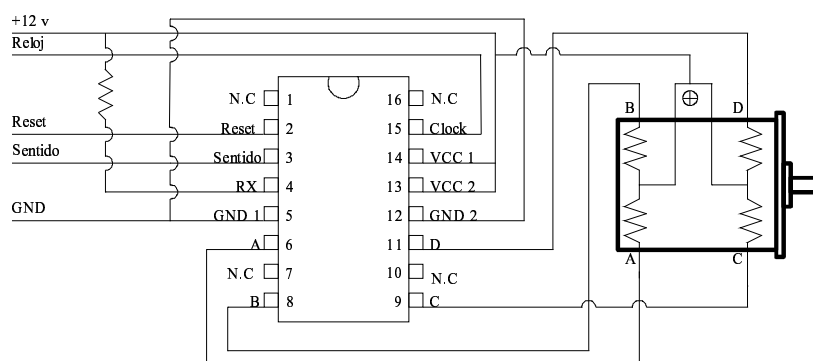


Figura C.4.1. Motor unipolar controlado con el SAA1027.

La entrada 'Reloj' es la que ordena dar un paso al motor. Esto ocurre siempre que 'reset' este inactivo es decir a nivel alto. El sentido de giro lo determina la entrada 'Sentido'. La entrada 'Reloj' se activa por flanco, cuando la señal cambia su estado de nivel bajo a nivel alto es cuando se ordena dar el paso al motor. Para volver a dar otro es necesario que se vuelva a introducir un flanco de subida.

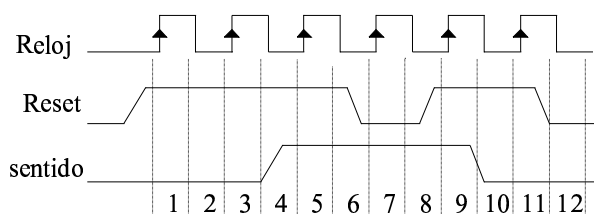


Figura C.4.2. Cronograma de control del SAA1027.

	1	2	3	4	5	6	7	8	9	10	11	12
Reset	off	off	off	off	off	on	on	off	off	off	off	on
Sentido	-->	-->	-->	<--	<--	<--	<--	<--	<--	-->	-->	-->
Reloj	up	down	up	down	up	down	up	down	up	down	up	down
Motor	-->	parado	-->	parado	<--	parado	parado	parado	<--	parado	-->	parado

APÉNDICE D: LISTADO DE LOS NEMONICOS DEL 68HC11

ABA : Añadir el contenido del acumulador B al acumulador A
ABX: Añadir el contenido del acumulador B (sin signo) al contenido del registro X
ABY: Añadir el contenido del acumulador B (sin signo) al contenido del registro Y
ADCA: Añadir al acumulador A un dato y el bit de acarreo.
ADCB: Añadir al acumulador B un dato y el bit de acarreo.
ADDA: Añadir un dato al registro A
ADDB: Añadir un dato al registro B
ADDD: Añadir un dato de 16 bits al registro D
ANDA: Realizar una operación lógica AND entre un dato y el acumulador A. Resultado en A
ANDB: Realizar una operación lógica AND entre un dato y el acumulador B. Resultado en B
ASLA: Desplazar un bit a la izquierda el acumulador A.
ASLB: Desplazar un bit a la izquierda el acumulador B
ASLD: Desplazar un bit a la izquierda el acumulador D
ASRA: Desplazar un bit a la derecha el acumulador A
ASRB: Desplazar un bit a la derecha el acumulador B
BCC: Saltar si no hay acarreo
BCLR: Poner a cero bits de la memoria
BCS: Saltar si hay acarreo
BEQ: Saltar si igual
BGE: Saltar si mayor que o igual a cero
BGT: Saltar si mayor que cero
BHI: Saltar si es mayor
BHS: Saltar si mayor o igual
BITA: Comprobar el bit especificado del acumulador A
BITB: Comprobar el bit especificado del acumulador B
BLE: Saltar si menor que o igual a cero
BLO: Saltar si menor (Mismo que BCS)
BLS: Saltar si menor o igual
BLT: Saltar si menor que cero
BMI: Saltar si negativo
BNE: Saltar si no es igual
BPL: Saltar si es positivo
BRA: Salto incondicional
BRCLR: Saltar si los bits especificados están a cero
BRN: No saltar nunca (Equivalente a una operación NOP de 2 bytes)
BRSET: Saltar si los bits especificados están a uno
BSET: Poner los bits especificados a uno
BSR: Saltar a una subrutina
BVC: Saltar si no ha habido overflow
BVS: Saltar si ha habido overflow
CBA: Comparar el acumulador A con el B
CLC: Poner a cero bit de acarreo
CLI: Permitir las interrupciones
CLR: Poner a cero el contenido de memoria especificado
CLRA: Poner a cero el acumulador A
CLRB: Poner a cero el acumulador B
CLV: Poner a cero el bit de overflow
CPMA: Comparar el acumulador A con un dato
CMPB: Comparar el acumulador B con un dato
COMA: Complementar a uno el acumulador A
COMB: Complementar a uno el acumulador B
COM: Complementar a uno el contenido de memoria especificado
CPD: Comparar el registro D con un dato

CPX: Comparar el registro X con un dato
CPY: Comparar el registro Y con un dato
DAA: Ajuste decimal
DEC: Decrementar una posición de memoria especificada
DECA: Decrementar el acumulador A
DECB: Decrementar el acumulador B
DES: Decrementar el puntero de pila SP
DEX: Decrementar el registro X
DEY: Decrementar el registro Y
EORA: Operación XOR entre un dato y el acumulador A
EORB: Operación XOR entre un dato y el acumulador B
FDIV: División
IDIV: División entera
INC: Incrementar el contenido de una posición de memoria
INCA: Incrementar el acumulador A
INCB: Incrementar el acumulador B
INS: Incrementar el puntero de pila SP
INX: Incrementar el registro X
INY: Incrementar el registro Y
JMP: Salto incondicional
JSR: Salto a una subrutina
LDAA: Cargar un dato en el acumulador A
LDAB: Cargar un dato en el acumulador B
LDD: Cargar un dato de 16 bits en el registro D
LDS: Cargar un dato de 16 bits en el puntero de pila SP
LDX: Cargar un dato de 16 bits en el registro X
LDY: Cargar un dato de 16 bits en el registro Y
LSL: Desplazamiento de un bit hacia la izquierda del contenido de una posición de memoria
LSLA: Desplazar el acumulador A un bit hacia la izquierda
LSLB: Desplazar el acumulador B un bit hacia la izquierda
LSLD: Desplazar el acumulador D un bit hacia la izquierda
LSR: Desplazar el contenido de una posición de memoria un bit hacia la derecha
LSRA: Desplazar el acumulador A un bit hacia la derecha
LSRB: Desplazar el acumulador B un bit hacia la derecha
LSRD: Desplazar el registro D un bit hacia la derecha
MUL: Multiplicación sin signo
NEG: Complementar a dos el contenido de una posición de memoria
NEGA: Complementar a dos el contenido del acumulador A
NEGB: Complementar a dos el contenido del acumulador B
NOP: No operación
ORAA: Realizar la operación lógica OR entre un dato y el acumulador A
ORAB: Realizar la operación lógica OR entre un dato y el acumulador B
PSHA: Meter el acumulador A en la pila
PSHB: Meter el acumulador B en la pila
PSHX: Meter el registro X en la pila
PSHY: Meter el registro Y en la pila
PULA: Sacar el acumulador A de la pila
PULB: Sacar el acumulador B de la pila
PULX: Sacar el registro X de la pila
PULY: Sacar el registro Y de la pila
ROL: Rotación a la izquierda del contenido de una posición de memoria
ROLA: Rotar a la izquierda el acumulador A
ROLB: Rotar a la izquierda el acumulador B
ROR: Rotar a la derecha el contenido de una posición de memoria
RORA: Rotar a la derecha el contenido del acumulador A
RORB: Rotar a la derecha el contenido del acumulador B
RTI: Retorno de interrupción
RTS: Retorno de subrutina
SBA: Restar un dato al acumulador A
SBCA: Restar un dato y el bit de acarreo al acumulador A

SBCB: Restar un dato y el bit de acarreo al acumulador B
SEC: Poner a uno el bit de acarreo
SEI: Inhibir las interrupciones
SEV: Poner a uno el bit de overflow
STAA: Almacenar el acumulador A en una posición de memoria
STAB: Almacenar el acumulador B en una posición de memoria
STD: Almacenar el registro D
STOP: Para el reloj del sistema
STS: Almacenar el puntero de pila SP
STX: Almacenar el registro X
STY: Almacenar el registro Y
SUBA: Restar un dato al acumulador A
SUBB: Restar un dato al acumulador B
SUBD: Restar un dato al registro D
SWI: Provocar una interrupción software
TAB: Transferir el acumulador A al acumulador B
TAP: Transferir el acumulador A al registro CCR
TBA: Transferir el acumulador B al acumulador A
TEST: Instrucción de test. Sólo se puede ejecutar en el modo de test
TPA: Transferir el registro CCR al acumulador A
TST: Comprobar si una posición de memoria está a cero
TSTA: Comprobar si el acumulador A está a cero
TSTB: Comprobar si el acumulador B está a cero
TSX: Transferir el puntero de pila al registro X
TSY: Transferir el puntero de pila al registro Y
TXS: Transferir el registro X al puntero de pila
TYS: Transferir el registro Y al puntero de pila
WAI: Esperar a que se produzca una interrupción
XGDX: Intercambiar los valores de los registros X y D
XGDY: Intercambiar los valores de los registros Y y D