



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

ROBOT SIGUE LÍNEAS EVITA OBSTÁCULOS PARA ENSEÑANZA DOCENTE

Autor: Carlos Mena Alepuz

Titulación: Grado en Ingeniería Electrónica
Industrial y Automática

Tutor: Francisco Javier Ibañez

Valencia, Septiembre de 2017

RESUMEN

En pleno siglo XXI la electrónica y robótica avanzan a un ritmo frenético. Esto se debe al descubrimiento de nuevos materiales y nuevas tecnologías mediante las cuales se consigue facilitar mucho la vida de los seres humanos.

Los robots están surgiendo tanto para realizar tareas para las cuales el ser humano no está preparado y también para ayudarnos en tareas cotidianas y repetitivas.

Se pueden encontrar todo tipo de robots, desde los utilizados en tareas del hogar como son los robots de limpieza o de cocina, hasta los utilizados en la industria para mover cargas pesadas o trabajar de manera más rápida y eficiente.

El siguiente proyecto está pensado para iniciar a los alumnos de la Escuela Técnica Superior de Ingeniería del Diseño en el mundo de la robótica. Para ello se va a diseñar un robot el cual servirá para que los alumnos se inicien en el mundo de la programación. La finalidad es que el alumnado aplique en el laboratorio los conocimientos adquiridos en el aula trabajando en el robot, programándolo para que desarrolle una serie de funciones con la mayor precisión y brevedad posibles.

Agradecimientos

En primer lugar, quiero agradecer el apoyo recibido por todas esas personas que han estado a mi lado cuando he llevado a cabo este proyecto. Gracias Javier por tu esfuerzo y ayuda para la consecución del proyecto. Agradecer también a toda mi familia y amigos porque sin vuestro apoyo y motivación esto no habría sido posible.

Índice

1. Introducción	1
1.1 Inicios de la robótica	1
WilCat , Boston Dynamics:	2
KR AGILUS, KUKA.....	3
Roomba 980, iRobot	3
Jia Jia, Universidad de Ciencia y Tecnología de China	4
2. Planteamiento del proyecto	5
3. Microcontrolador PIC16F88	6
4. Componentes utilizados	7
5. Funciones.....	11
5.1 Sigue líneas	11
5.2 Evita obstáculos.....	14
5.2.1 Función obtener distancia desarrollada en C++	16
5.3 Control mediante infrarrojos	16
5.3.1 Codificación de datos SONY	18
5.3.2 Función para capturar datos sony en c++	20
6. Diseño	21
6.1 Esquemático completo función evita obstáculos	25
6.2 Esquemático completo función sigue líneas	26
6.3 Esquemático completo función control remoto	27
6.4 Esquemático en ARES para el diseño de PCB	28
7. Montaje	31
8. Programación	35
8.1 Código cabecera	36
8.2 Código función sigue líneas.....	37
8.3 Código función evita obstáculos	39
8.4 Código control Remoto	41
8.5 Diagramas de flujo del código	42
9. Sistemas y Subsistemas	45
9.1 Sistema mecánico	45
9.2 Sistema electrónico	45
9.3 Diagrama de sistemas.....	46
10. Conclusiones.....	47
11. Bibliografía.....	48
12. ANEXOS.....	49
10.1 Posibles mejoras.....	49
10.2 Pliego de condiciones	52
10.3 Presupuesto	56
10.4 Datasheets	59

Índice de ilustraciones

Ilustración 1 WildCat, Boston Dynamics	2
Ilustración 2 KR AGILUS , KUKA	3
Ilustración 3 Romba 980	3
Ilustración 4 Jia Jia	4
Ilustración 5 PIC 16F88.....	6
Ilustración 6 Pines PIC16F88	7
Ilustración 7 Chasis robot.....	8
Ilustración 8 Ruedas y motores robot.....	8
Ilustración 9 Acabado final del chasis según fabricante	9
Ilustración 10 Cables utilizados	9
Ilustración 11 Leds 5mm	10
Ilustración 12 Resistencias	10
Ilustración 13 Driver L298	10
Ilustración 14 Sensor Infrarrojo CNY 70	12
Ilustración 15 Esquemático Siguelíneas	12
Ilustración 16 Esquemático Siguelíneas real.....	13
Ilustración 17 Sensor de ultrasonidos HCSR04	14
Ilustración 18 Funcionamiento gráfico HCSR04.....	15
Ilustración 19 Cálculo distancia.....	15
Ilustración 20 Funcionamiento TSOP	17
Ilustración 21 Sensor Infrarrojo TSOP44xx.....	17
Ilustración 22 Esquemático TSOP 44xx	18
Ilustración 23 Tren de pulsos protocolo SIRC 12 Bits.....	18
Ilustración 24 Esquemático completo robot.....	21
Ilustración 25 Esquemático conexión LED's con PIC16F88	22
Ilustración 26 Esquemático conexión driver L298 con motor DC al microcontrolador	23
Ilustración 27 Puente H giro sentido horario	24
Ilustración 28 Puente H giro sentido antihorario.....	24
Ilustración 29 Esquemático completo función Evita obstáculos.....	25
Ilustración 30 Esquemático completo función Control remoto.....	26
Ilustración 31 Esquemático conexión módulo sigue líneas con microcontrolador	27
Ilustración 32 PCB módulo sigue líneas ARES	28
Ilustración 33 PCB Central ARES.....	29
Ilustración 34 PCB Central.....	30
Ilustración 35 PCB módulo sigue líneas.....	30
Ilustración 36 Montaje motor al chasis.....	31
Ilustración 37 Chasis con ruedas, motores y rueda libre montado	32
Ilustración 38 Chasis con módulo sigue líneas y Driver montados.	33
Ilustración 39 Chasis con todas las conexiones.....	34
Ilustración 40 Conexión Pickit.	35
Ilustración 41 Diagrama de flujo sigue líneas.....	42
Ilustración 42 Diagrama de flujo evita obstáculos.	43
Ilustración 43 Diagrama de flujo control remoto.....	44
Ilustración 44 Diagrama de Flujo de los sistemas del robot.	46

Índice de tablas

Tabla 1 Dispositivos receptores	19
Tabla 2 Código por botón.....	19
Tabla 3 Coste componentes.....	56
Tabla 4 Coste mano de obra	57
Tabla 5 Tiempo asociado a la mano de obra.....	57
Tabla 6 Tiempo asociado a la mano de obra.....	57
Tabla 7 Coste adicionales	58
Tabla 8 Coste total.....	58

1. Introducción

1.1 Inicios de la robótica

Durante siglos los seres humanos han tratado de construir máquinas que se asemejasen a partes del cuerpo humano, los egipcios ya unieron brazos mecánicos a las estatuas de sus dioses. También los griegos construyeron estatuas con sistemas hidráulicos pero los inicios de la robótica como tal puede fijarse a principios del siglo XVIII durante el inicio de la industria textil, cuando Joseph Jacquard inventa en 1801 una máquina textil programable mediante tarjetas perforadas.

¿Sabemos realmente el significado de lo que es un “robot”? Este término fue introducido por primera vez por el dramaturgo y autor checoslovaco Karel Capek, en su obra de teatro R.U.R (Robots Universales de Rossum) en 1921, la cual trataba como tema principal el posible peligro de estas máquinas ya que según el autor el ser humano creaba el robot y era el robot el que eliminaba a su creador.

Fue en el año 1939 cuando Isaac Asimov comenzó a desarrollar diferentes robots y a él se le atribuye el acuñamiento del término robótica y es cuando surgen las denominadas “tres Leyes de Robótica ” que son las siguientes:

1. Un robot no puede actuar contra un ser humano o, mediante la inacción, que un ser humano sufra daños.
2. Un robot debe de obedecer las órdenes dadas por los seres humanos, salvo que estén en conflictos con la primera ley.
3. Un robot debe proteger su propia existencia, a no ser que esté en conflicto con las dos primeras leyes.

Durante el transcurso del siglo XX se desarrollaron los primeros robots, en 1954 Devol, diseña el primer robot programable. En 1960 el primer robot Unimate se instaló en Ford Motor Company para atender una máquina de fundición de troquel.

En 1978 se introdujo el robot PUMA para tareas de montaje por Unimation, basándose en diseños obtenidos en un estudio de la empresa General Motors.

En grandes rasgos la historia de la robótica la podemos clasificar en cinco generaciones: las dos primeras, alcanzadas en los ochenta, incluían la gestión de tareas repetitivas con autonomía muy limitada. La tercera generación incluiría visión artificial, en lo cual se ha avanzado mucho en los ochenta y noventa.

La cuarta incluye movilidad avanzada en exteriores e interiores y la quinta entraría en el dominio de la inteligencia artificial en lo cual se está trabajando actualmente.

La definición más concreta de robot según:

- La **Asociación Japonesa de Robótica Industrial (JIRA)** define robot como:

"Dispositivos capaces de moverse de modo flexible análogo al que poseen los organismos vivos, con o sin funciones intelectuales, permitiendo operaciones en respuesta a las ordenes humanas"

- El **Instituto de Robótica de América (RIA)** define robot industrial como:

"Un manipulador multifuncional y programable diseñado para desplazar materiales, componentes, herramientas o dispositivos especializados por medio de movimientos programados variables con el fin de realizar tareas diversas"

Aquí podemos ver algunos ejemplos de robots del siglo XXI:

WildCat , Boston Dynamics:

Boston Dynamics, empresa especializada en robots que forma parte del gigante Google, ha desarrollado un novedoso robot que responde al nombre de WildCat. Se trata de un cuadrúpedo capaz de trotar, cambiar de dirección y galopar a una velocidad de 25 Km/h. El robot está radiocontrolado para que siga las indicaciones de los científicos.

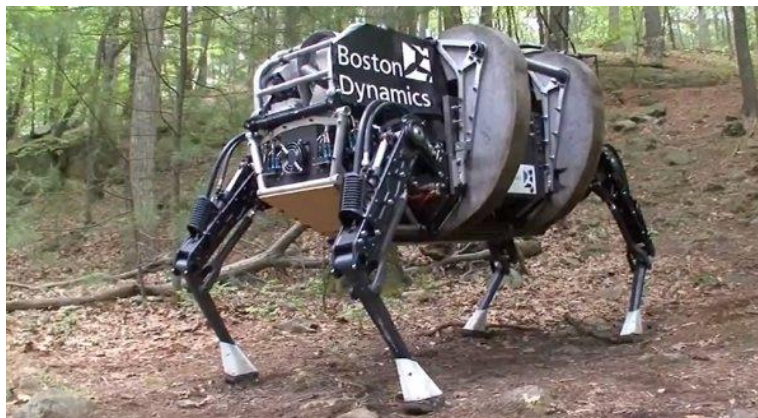


Ilustración 1: WildCat, Boston Dynamics

KR AGILUS, KUKA

Se trata de un robot automatizado para ser usado en líneas de producción. Diseñado por la empresa alemana KUKA Roboter GmbH. Este robot posee 6 ejes mediante los cuales puede desarrollar tareas de corte, empaque, paletización, manipulación y ensamblaje a altas velocidades. Tiene una capacidad máxima de carga de 6Kg y un alcance de aproximadamente 706,7 mm .



Ilustración 2 : KR AGILUS , KUKA

Roomba 980, iRobot

Se trata de un robot aspirador para el hogar el cual navega por toda su casa controla su posición y se recarga según sea necesario hasta terminar el trabajo de aspiración. Se puede configurar mediante una APP móvil. Es un claro ejemplo de la evolución de los robots para facilitar la existencia a los seres humanos.



Ilustración 3 : Roomba 980

Jia Jia, Universidad de Ciencia y Tecnología de China

Se trata del robot humanoide más avanzado hasta la fecha, es capaz de responder a las preguntas que le hagan los humanos, reaccionar mediante expresiones en su rostro y diferentes funciones de inteligencia artificial como detectar el género de la persona con la que está hablando entre otras.



Ilustración 4 : Jia Jia

2. Planteamiento del proyecto

Teniendo en cuenta lo citado anteriormente, el objetivo de este proyecto es la elaboración de un robot el cual los alumnos de la ETSID aprendan el funcionamiento de un microcontrolador mediante la programación del mismo.

El robot será diseñado pensando en que sea fácil la sustitución de cualquiera de sus componentes ya que tendrá que soportar muchas pruebas y podrá sufrir accidentes. Con esto se pretenden conseguir que el alumno sea capaz de desarrollar sus conocimientos y pueda ponerlos en práctica en los laboratorios

Las funcionalidades del robot serán las siguientes:

- Función **Sigue líneas**, el robot deberá ser capaz de seguir por sí mismo un trazado oscuro en una superficie sin salirse de este mediante el uso de 3 emisores- receptores de infrarrojos CNY70
- Función **Evita obstáculos**, el robot deberá ser capaz de recorrer un circuito evitando todos los obstáculos puestos en el recorrido. Para esto el robot dispondrá de un sensor de ultrasonidos HCSR04.
- **Control mediante infrarrojos**, para esta función el alumno dispondrá de un mando universal de TV a modo de control remoto para enviar señales de infrarrojos a un receptor TSOP48XX ubicado en el robot y así dirigirlo por un recorrido.

3. Microcontrolador PIC16F88

Un microcontrolador es un circuito integrado digital y programable que puede ser usado para diversas funcionalidades. Esta compuesto por una unidad central de proceso (CPU), memorias (ROM y RAM) y líneas de entrada y salida.

El elegido para este proyecto es el PIC16F88, distribuido por Microchip, empresa fundada en 1989 en EE.UU dedicada a la fabricación de microcontroladores, memorias y semiconductores analógicos.

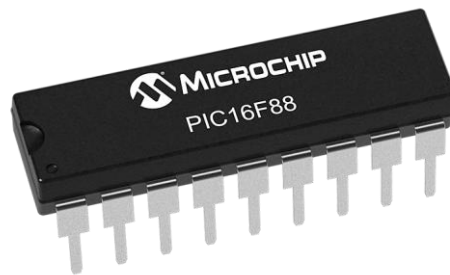


Ilustración 5 : PIC 16F88

Se trata de un microcontrolador muy comercializado debido a su sencilla arquitectura de 8 bits y 18 pines, internamente consta de:

- Memoria Flash de programa (4K x 14).
- Memoria EEPROM de datos (256 x 8).
- Memoria RAM (368 registros x 8).
- Oscilador interno de 8MHz.
- Modulaci3n por ancho de pulso (PWM) de 10Bits,
- Comunicaci3n as3ncrona mediante su m3dulo AUSART
- Comunicaci3n s3ncrona SPI 3 Hilos o I2C en 2 hilos, con el m3dulo SSP
- Conversor de Anal3gico a Digital de 7 canales
- Un Timer de 8 bits
- Un divisor de frecuencia.
- Varios puertos de entrada-salida (16 pines en dos puertos, 8 pines el puerto A y 8 pines el puerto B).

En la siguiente imagen podemos observar la disposición de los pines físicamente en el microcontrolador.

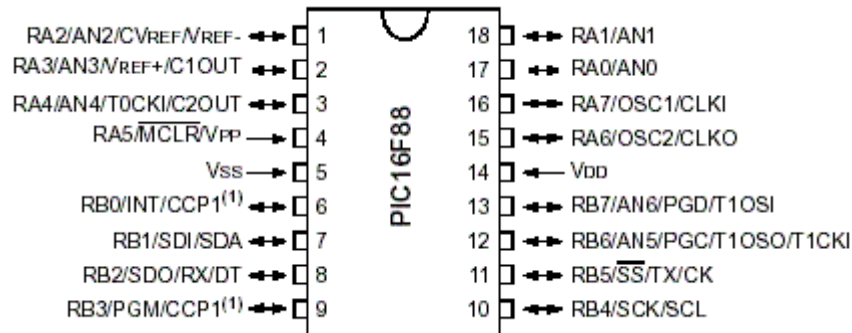


Ilustración 6 : Pines PIC16F88

4. Componentes Utilizados

Para el desarrollo del robot hemos elegido una serie de materiales basándonos en sus características y funcionalidades pero también intentando reducir el coste del proyecto al mínimo posible.

Para ello se han seleccionado los siguientes componentes:

El microcontrolador seleccionado es el presentado en el apartado anterior. Se trata del PIC16F88 ya que cumple con los requisitos de nuestro proyecto gracias a su número de entradas y salidas tal y como se puede observar en el datasheet anexo al final del proyecto.

Para la estructura del robot se ha decidido comprar un pack que incluye todo lo necesario.

En el pack comprado viene incluido:

- 1 x Chasis de vehículo
- 2 x Motor
- 2 x ruedas con gomas
- 2 x codificadores de velocidad
- 4 x Faston
- 1 x rueda libre
- 1 x interruptor
- 1 x Porta pilas
- 1 x Pack de tornillos y tuercas

El chasis está fabricado sobre metacrilato y lleva todas las perforaciones necesarias para la correcta colocación de todos los componentes que vienen en el pack.



Ilustración 7: Chasis robot

En la imagen siguiente podemos observar las dos ruedas junto a los encoders y a los motores de continua los cuales proporcionarán el movimiento a nuestro robot.



Ilustración 8: Ruedas y motores robot

El resultado final después de haber montado correctamente todas las piezas tiene que ser el siguiente a excepción del porta-pilas ya que en su lugar se va a usar una batería externa la cual será adherida al chasis mediante velcro.

Todas las instrucciones de montaje las podremos seguir en los anexos adjuntos al proyecto.

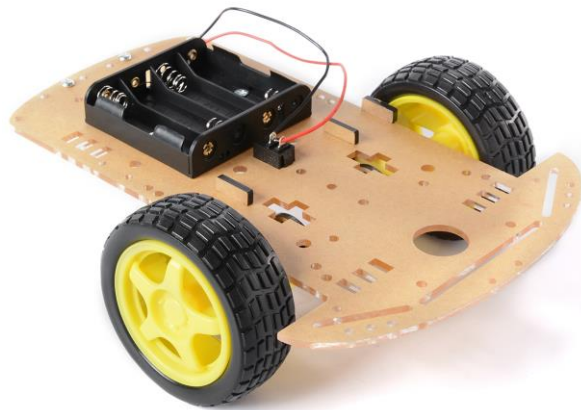


Ilustración 9: Acabado final del chasis según fabricante

Para el conexionado de todos los elementos electrónicos del sistema de seguimiento se utilizará cable de cobre estañado recubierto de PVC 1mm de diámetro, debido a su bajo coste y fácil manejo.

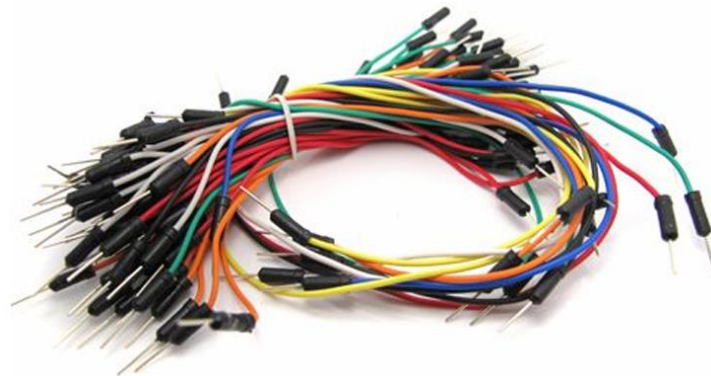


Ilustración 10: Cables utilizado

Se han utilizado 3 leds de 5mm dos de color rojo y uno de color verde para indicar los diferentes estados en los que el robot se encuentra. Para el correcto conexionado de los leds y también de los sensores CNY70 se deben usar resistencias de diferentes valores estandarizados. Las siguientes ilustraciones muestran los tipos de resistencias y leds que se van a utilizar



Ilustración 11: Leds 5mm

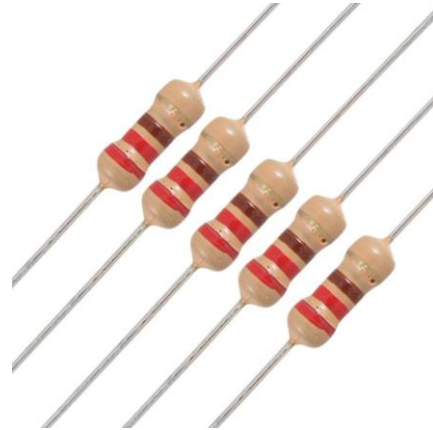


Ilustración 12: Resistencias

En lo respectivo al movimiento del robot, los encargados de conseguirlo son los dos motores de continua con reductora de velocidad situados en las ruedas del mismo. Para conseguir su correcto funcionamiento se ha utilizado un driver L298.

Se ha decidido usar este driver ya que es capaz de suministrar a los motores los 5V y 100mA que necesitan para funcionar, también porque es capaz de funcionar con una alimentación desde 6V a 35V gracias al regulador integrado LM7805 el cual le permite funcionar de dos maneras. Puede funcionar con una alimentación entre 6V y 12V cuando el jumper de selección está activo y entre 12V y 35V cuando el jumper está inactivo. Todo esto se puede obtener de los datasheets adjuntos como anexos.



Ilustración 13: Driver L298

Se trata de un driver con dos puentes-H, los cuales nos permiten controlar la velocidad y la dirección de los dos motores. Los puentes-H están formados por 4 transistores que permiten la inversión del sentido de la corriente y de esta forma invertir el giro de los motores

Sus características principales son las siguientes:

- Voltaje de alimentación, mínimo de 5 V. Posee dos entradas, una de 5V para controlar la parte lógica y otra para alimentar las salidas al motor, que pueden ser de 5V o más.
- La tarjeta tiene la opción de habilitar un regulador LM7805 integrado en ella para alimentar la parte lógica.
- Corriente máxima 2 Amperios.
- Posee 6 entradas de control (ver tabla de control)
- Admite entradas de señal PWM para el control de velocidad.
- Dimensiones: 43 mm x 23,9 mm x 43 mm.
- Salidas: para 2 motores de DC o para un motor bipolar paso a paso

5. FUNCIONES

5.1 SIGUE LÍNEAS

Para realizar la función de siguelineas el robot necesitará recibir datos de su entorno por ello se ha decidido usar 3 emisores- receptores de infrarrojos CNY70 para diseñar el módulo sigue líneas.

El modelo de sensor elegido es el CNY70, se trata de un sensor compacto donde la fuente de luz (diodo emisor) y el detector (fototransistor) se montan en la misma dirección. La detección se consigue mediante la reflexión o no del haz infrarrojo sobre la superficie dibujada.

Características del sensor:

- Longitud de onda del haz infrarrojo de 950nm.
- Intensidad del diodo emisor $I_{AK} = 50mA$
- Intensidad de colector $I_C = 50mA$
- Tensión de colector emisor $I_{CE} = 32V$
- Tensión de emisor colector $V_{EC} = 7V$.
- Consumo aproximado de 200mW
- Distancia de detección de 0.3 a 5mm

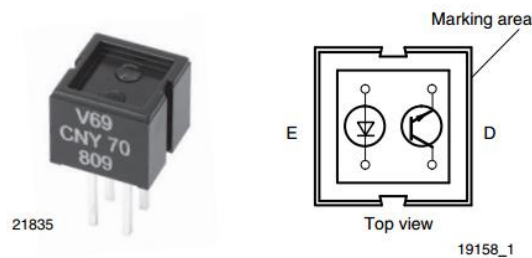


Ilustración 14: Sensor Infrarrojo CNY 70

En la siguiente imagen se puede observar el esquemático utilizado para simular los sensores CNY70 en PROTEUS. Se ha decidido utilizar pulsadores ya que no se puede simular el funcionamiento del sensor en el programa. La función de los pulsadores es simular cuando el sensor detecta la línea pulsando los diferentes pulsadores. Los sensores CNY70 están nombrados como SI (sensor izquierdo), SC (sensor central) y SD (sensor derecho).

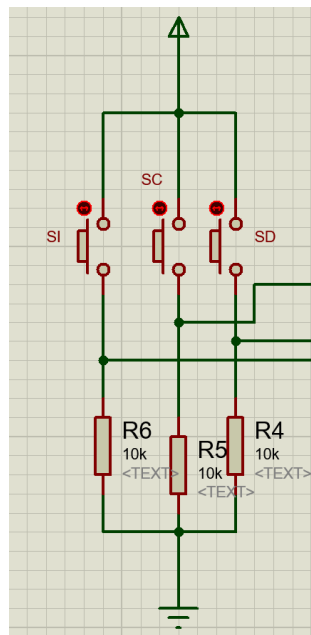


Ilustración 15: Esquemático Siguelineas

El esquemático real utilizado para el desarrollo de la PCB es el siguiente:

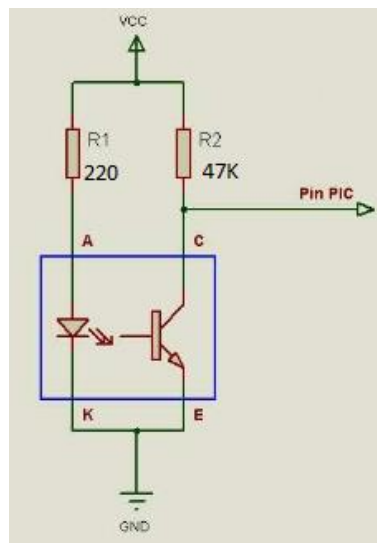


Ilustración 16: Esquemático Siguelineas real

En el esquemático de la ilustración 16 se observa el circuito de activación necesario para un único sensor CNY70. Se precisan dos resistencias cuyos valores se obtienen de la siguiente forma.

Observando en el datasheet adjunto en los anexos se obtiene que la intensidad máxima que puede tener el diodo es de 50mA y que la caída de voltaje es aproximadamente 1,5 V por tanto tenemos lo siguiente:

$$I = \frac{V - V_{led}}{R} = < 50mA = \frac{5V - 1.5V}{220\Omega} = \mathbf{16mA} < 50mA$$

La resistencia del led puede ser de 220 Ω ó superior ya que con esta ya se consigue una intensidad en el led de aproximadamente 16mA que es menor que la máxima admitida (50mA) por tanto el led funcionará correctamente.

5.2 EVITA OBSTÁCULOS

En la función de evita obstáculos se necesita que el robot detecte los obstáculos a su alrededor con la suficiente antelación y distancia para evitar colisionar con estos. Para ello se ha utilizado un sensor de ultrasonidos modelo HCSR04.

Un sensor de ultrasonidos es un dispositivo para medir distancias. Su funcionamiento se basa en enviar un pulso de alta frecuencia el cual rebota en los objetos cercanos y es reflejado hacia el sensor que dispone de un micrófono adecuado a esa frecuencia. Midiendo el tiempo transcurrido entre el envío del pulso y la recepción del mismo se puede obtener la distancia a la que se encuentra el objeto a evitar.

En el modelo HCSR04 las características principales son las siguientes:

- Corriente de reposo < 2mA
- Corriente de trabajo: 15mA
- Ángulo de medición: 30°
- Ángulo de medición efectivo : 15°
- Detección de 2cm a 400cm
- La precisión puede variar entre los 3mm o 0.3cm
- Frecuencia de trabajo 40KHz

Pines:

- VCC: Alimentación +5V (4.5V min – 5.5V max)
- TRIG: Disparo del sensor (TTL) recibe un pulso de habilitación de parte del microcontrolador, mediante el cual se le indica al módulo que comience a realizar la medición de distancia.
- ECHO: Eco del disparo del sensor (TTL) a través de este pin el sensor “muestra” al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de vuelta al módulo.
- GND

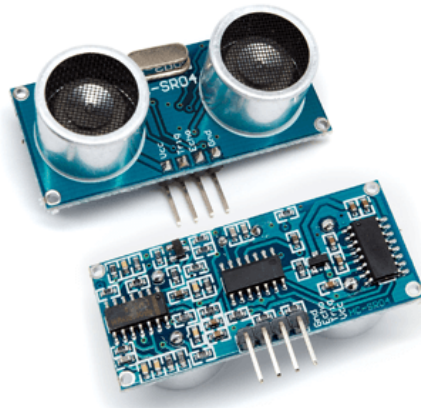


Ilustración 17: Sensor de ultrasonidos HCSR04

Funcionamiento

1. Enviar un pulso de aproximadamente 10µs por el pin Trigger
2. El sensor enviará 8 pulsos de 40KHz y coloca el pin Echo en alto, esto es necesario para activar el contador de tiempo.
3. Cuando el sensor reciba el reflejo del pulso pondrá el pin Echo a nivel bajo y así terminará el conteo de tiempo.
4. La distancia se puede obtener empleando la velocidad del sonido (340 m/s) y multiplicándola por el tiempo obtenido en µs x 0.017

Un dato a tener en cuenta es que el tiempo medido es el doble al real ya que se cuenta el tiempo que tarda la señal en ir y volver por tanto recorre dos veces la distancia que separa el sensor del objeto.

En las siguientes ilustraciones se puede observar lo explicado con mayor detalle y de forma más sencilla de entender.

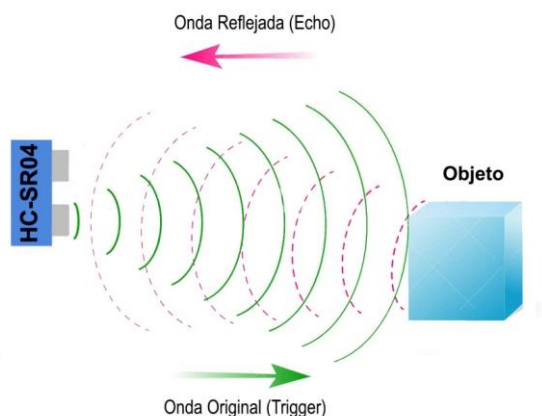


Ilustración 18: Funcionamiento gráfico HCSR04

$$\text{Velocidad} = \frac{\text{Espacio}}{\text{Tiempo}} \longrightarrow \text{Espacio} = \text{Velocidad} \times \text{Tiempo}$$

$$\text{Velocidad del sonido} = 343 \text{ m/s} = 0.0343 \text{ cm/}\mu\text{s}$$

$$\text{Espacio} = 0.0343 \times \text{Tiempo}$$

*Pero como la onda ha recorrido el camino dos veces (ida y vuelta) hay que dividir entre dos para conocer la distancia a la que se encuentra el objeto.

$$\boxed{\text{Espacio} = 0.01715 \times \text{Tiempo}}$$

Ilustración 19: Cálculo distancia

A continuación se detalla una función elaborada en código C++ para obtener datos de la distancia a la que se haya el robot de los obstáculos de su alrededor mediante el sensor de ultrasonidos.

5.2.1 FUNCIÓN OBTENER DISTANCIA DESARROLLADA EN C++

```
// FUNCIÓN DISTANCIA

#define DISP PORTAbits.RA6
#define ECO PORTAbits.RA5
float dist()
{
    float dist_cm;
    unsigned int medida;
    T1CON=0b00000101;
    medida=0;
    DISP = 1;
    _delay(10);
    DISP = 0;
    while (ECO == 0){};
    TMR1=0;
    while (ECO == 1){};
    medida = TMR1;
    dist_cm = medida/58;
    return (dist_cm);
}
```

5.3. CONTROL MEDIANTE INFRARROJOS

En este modo la finalidad es conseguir controlar con un mando universal los movimientos del robot mediante el uso de señales infrarrojas.

Para ello se ha utilizado un mando universal para la marca SONY y un receptor de infrarrojos el cual recibirá los datos de movimiento de los motores del mando y los descodificará para pasarlos al microcontrolador o arduino y conseguir el movimiento deseado.

Para el componente receptor de infrarrojos se ha seleccionado el modelo TSOP48XX de fabricante VISHAY. Se ha seleccionado este modelo debido a su bajo coste y su compatibilidad con mandos universales, se trata de un sensor basado en un fotodiodo PIN sensible a los infrarrojos. Lleva incorporado un amplificador de transimpedancia (convertor corriente-tensión) y el demodulador.

Al tratarse de un modelo con salida en el colector del transistor, en reposo proporcionará un nivel lógico alto. Al recibir un haz infrarrojo, la salida del transistor satura y generará un nivel lógico bajo.

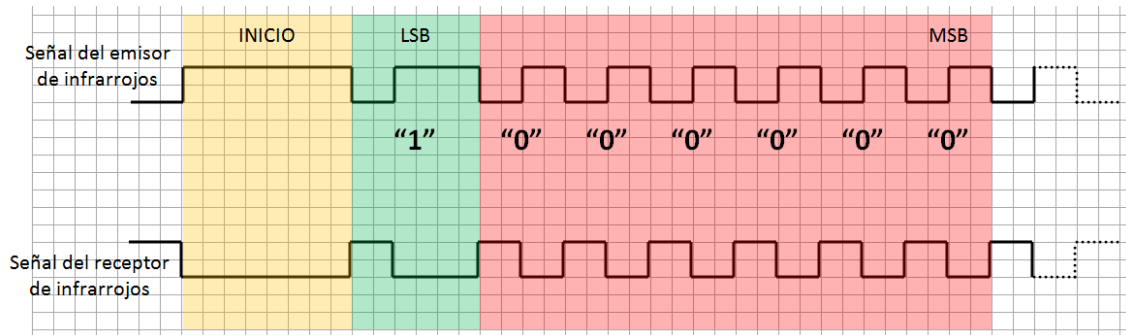
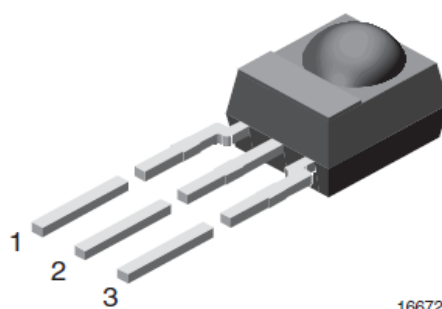


Ilustración 20: Funcionamiento TSOP

En la ilustración número 20 se puede observar como actuaría el receptor TSOP44XX al recibir una señal del mando a distancia. Este saturará al recibir un nivel lógico alto y generará un nivel lógico bajo.

En la siguiente ilustración se pueden observar la posición de los tres pines según el modelo, en nuestro caso el TSOP48



16672

MECHNICAL DATA

Pinning for TSOP44.., TSOP48..:

1 = OUT, 2 = GND, 3 = V_S

Pinning for TSOP22.., TSOP24..:

1 = OUT, 2 = V_S , 3 = GND

Ilustración 21: Sensor Infrarrojo TSOP44xx

La siguiente ilustración muestra como es el esquema interior del sensor con sus componentes.

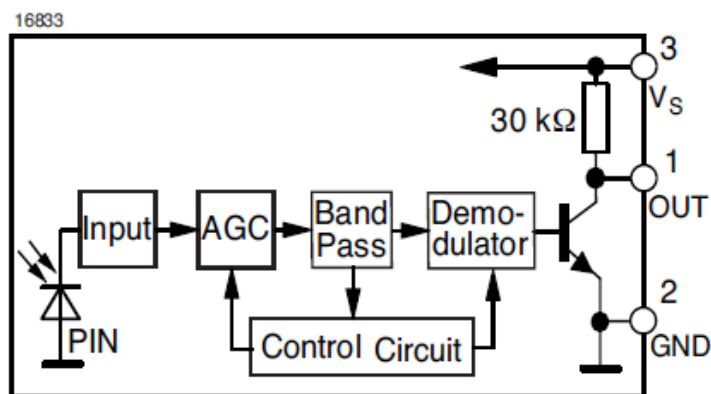


Ilustración 22: Esquemático TSOP 48xx

5.3.1 Codificación de datos SONY

El mando utilizado se trata de un mando universal compatible con la marca SONY cuyas características son las siguientes:

- Distancia de funcionamiento : 10m
- 1 Led transmisor
- Base de datos de códigos IR universales
- Nº de marcas de base datos: > de 1100
- Alcance de la frecuencia portadora: 24 – 55Khz
- Angulo de transmisión: 90 grados.
- Alimentación mediante 2 pilas AAA

La siguiente ilustración muestra un tren de pulsos típico del protocolo SIRC de 12 bits utilizado para transferir datos con un mando universal SONY. Con este protocolo el primer bit transmitido es el LSB y el tiempo básico (T) es de 600us y coincide con la duración de un nivel bajo.

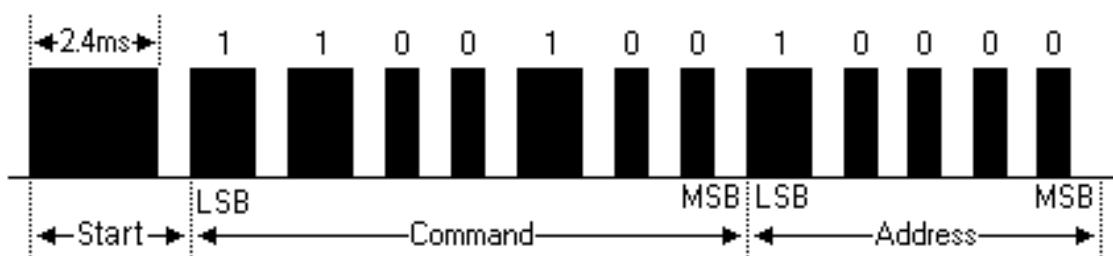


Ilustración 23: Tren de pulsos protocolo SIRC 12 Bits

La condición de inicio (START) se da al enviar un nivel alto durante 4T, lo equivalente a 2,4ms. A continuación se envían 7 bits mediante los cuales se indica la tecla pulsada (COMMAND) seguidos de 5 bits más para indicar el destino (ADDRESS).

En las siguientes tablas se pueden observar las relaciones de cada tecla del mando con el "command" que este enviará.

Address	Type
1	TV
2	VCR 1
3	VCR 2
6	Laser Disc Unit
12	Surround Sound
16	Cassette deck
17	CD Player
18	Equalizer

Tabla 1: Dispositivos receptores

Command	Function
0	Digital Key 1
1	Digital Key 2
2	Digital Key 3
3	Digital Key 4
4	Digital Key 5
5	Digital Key 6
6	Digital Key 7
7	Digital Key 8
8	Digital Key 9
9	Digital Key 0
16	Channel +
17	Channel -
18	Volume +
19	Volume -
20	Mute
21	Power
22	Reset
23	Audio Mode
24	Contrast +
25	Contrast -
26	Colour +
27	Colour -
30	Brightness +
31	Brightness -
38	Balance Left
39	Balance Right
47	Standby

Tabla 2: Código por botón

A continuación se detalla una función elaborada en código C++ para obtener datos sobre las teclas usadas en el control remoto mediante el sensor TSOP y así poder utilizarlos para con ayuda del microcontrolador mandar órdenes a los motores.

5.3.2 FUNCIÓN PARA CAPTURAR DATOS SONY EN C++

```
///FUNCIÓN CAPTURA SONY

int captura(void) {
    int K=0, valor=0;
    int dato [7];

    ETQ1:

    for (K==; K<7; K++){dato[K] = 0;};
    while(ENT==0){};
    __delay_us(900);
    if (ENT==0) {
        while(ENT==0){};
        for(K=0; K<7; K++){
            while (ENT==1){};
            __delay_us(900);
            if(ENT==0)
                dato[K]=1;
            while(ENT==0){};
            else dato[K]=0;
            valor= valor + (dato[K]<<K);
        }
        return(valor);
    }
    goto ETQ1;
}
```

6. DISEÑO

El diseño de las placas del circuito impreso ha sido elaborado con el programa de circuitos PROTEUS, mediante el cual se ha diseñado el esquemático que se puede observar a continuación.

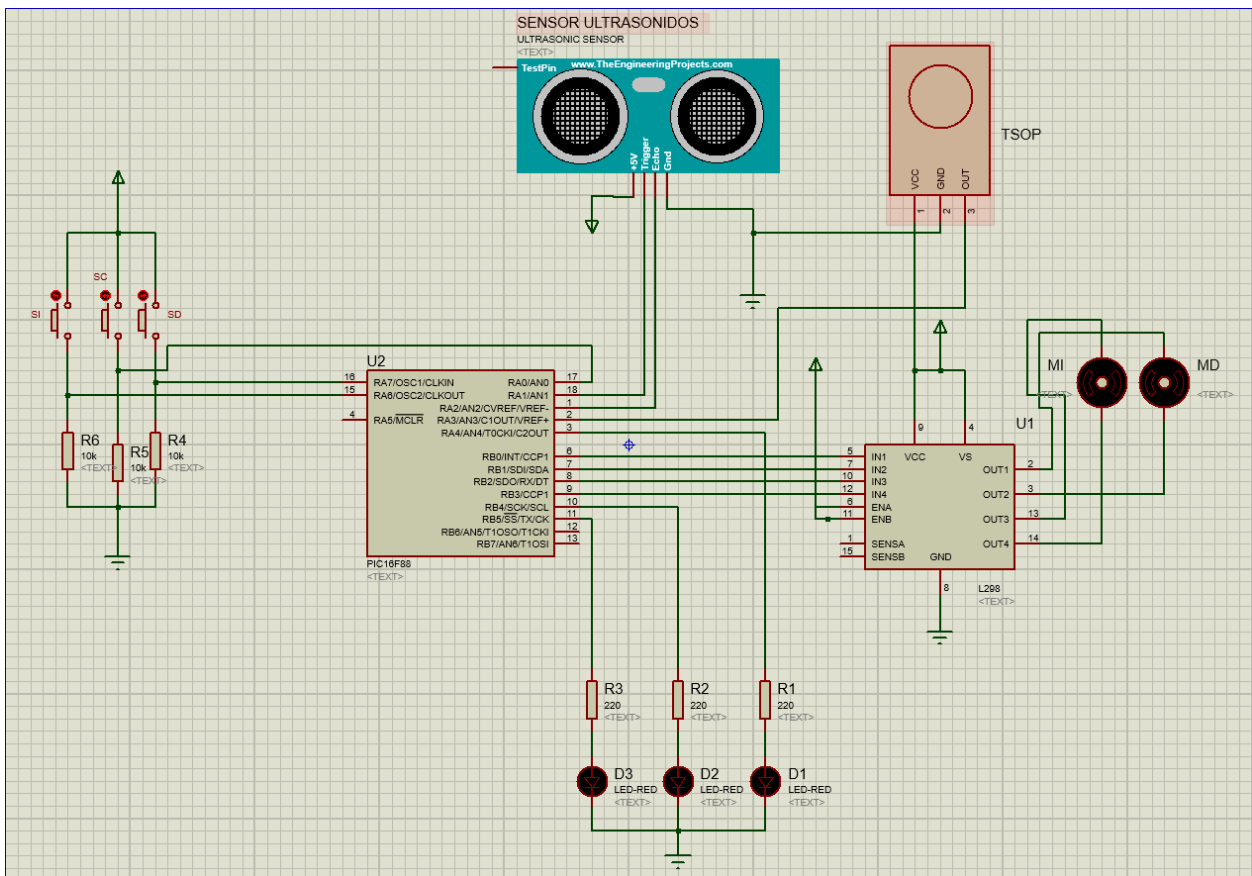


Ilustración 24: Esquemático completo robot

En este esquemático se observan todas las conexiones necesarias para el correcto funcionamiento del robot en los diferentes modos que este posee. A continuación se explicarán detalladamente cada módulo del mismo para su correcto entendimiento.

El programa empleado para el desarrollo de este ha sido PROTEUS versión 8.4. El primer paso para empezar con el diseño ha sido añadir el PIC que vamos a utilizar, en nuestro caso el PIC16F88 el cual podemos observar en la siguiente imagen junto a 3 diodos leds usados para indicar la dirección en la que gira el robot.

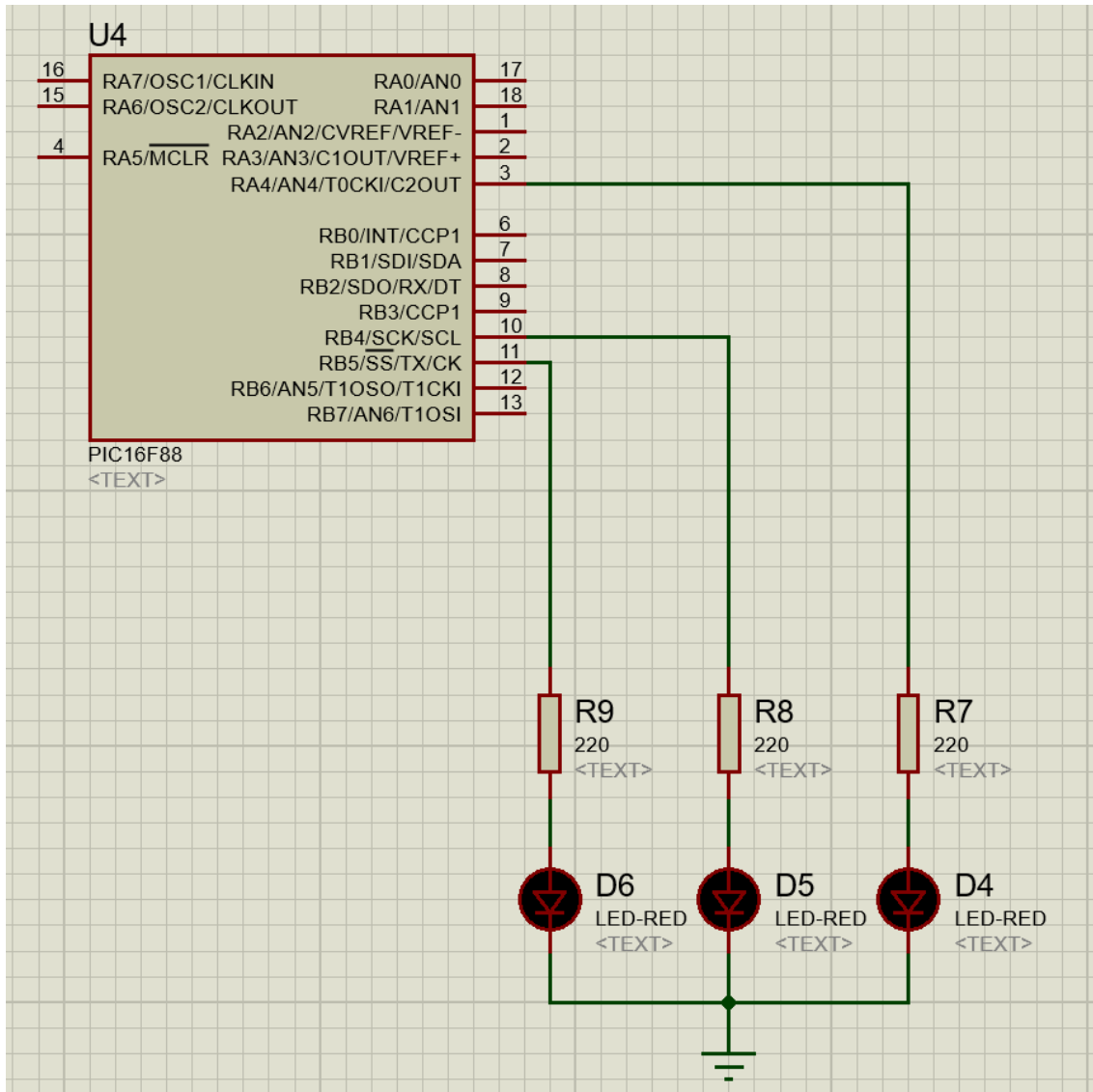


Ilustración 25: Esquemático conexión LED's con PIC16F88

Para las conexiones de los diodos se han utilizado las salidas RA4, RB4 y RB5 del PIC. A continuación se han añadido una resistencia antes de cada diodo para que estos funcionen correctamente y evitemos que se estropeen por una elevada tensión.

Estas resistencias son R9, R8 y R7, cada una de ellas de un valor de 220Ω conectadas a los diodos led D6, D5 y D4. Este valor se calcula teniendo en cuenta lo siguiente:

Se están usando diodos de 5mm y los valores para funcionamiento óptimo de estos según fabricante son 2.1V y 15mA. Teniendo en cuenta que el PIC16F88 saca de sus pines 5V y con ayuda de la ley de Ohm obtenemos lo siguiente:

$$V = V_{cc} - V_d = I_{nominal} * R$$

Por lo que el valor de la resistencia resulta:

$$R = \frac{V_{cc} - V_d}{I_{nominal}} = \frac{5V - 1.8V}{15mA} = 213\Omega$$

El valor obtenido es 213 Ω el valor inmediatamente superior normalizado serían **220 Ω** .

Aquí tendríamos el primer módulo el cual funcionará en todos los modos del robot y es independiente de las demás funciones.

En la siguiente imagen podemos observar el esquemático compuesto por los dos motores de continua y el puente en H para su óptimo funcionamiento L296D conectados al PIC16F88.

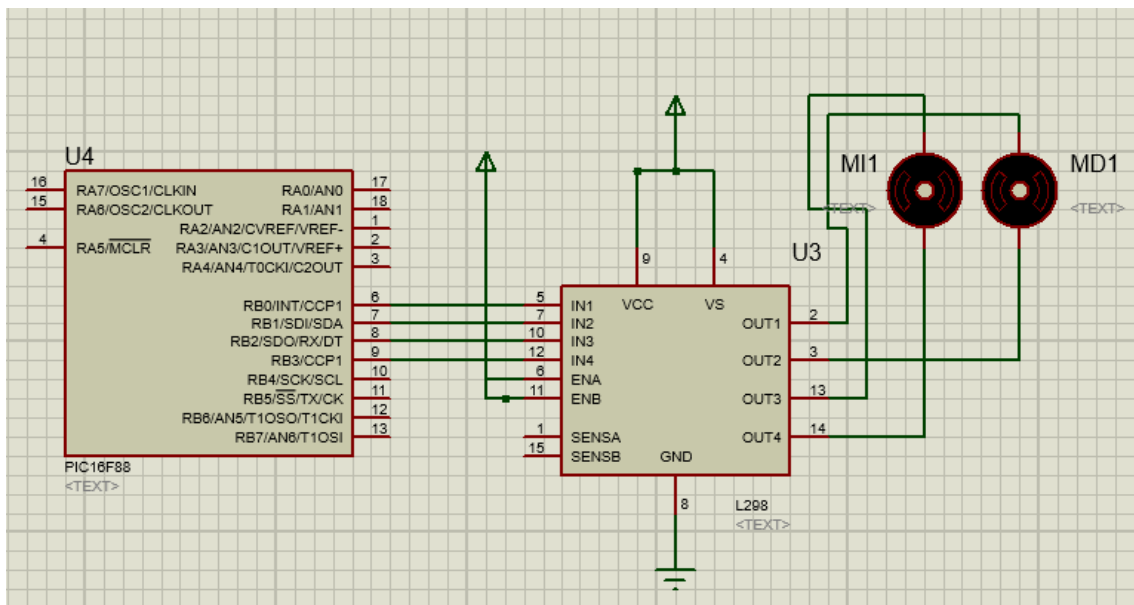


Ilustración 26: Esquemático conexión driver L298 con motor DC al microcontrolador

En la imagen podemos observar como las salidas RB0, RB1, RB2 y RB3 pertenecientes al microcontrolador están conectadas a las entradas IN1, IN2, IN3 e IN4 del puente en H. La entrada ENB del LD296 está conectada a una fuente de alimentación +5V al igual que los pines VCC y VS. El pin 8 del puente en H está conectado a tierra.

La parte más importante de este módulo es la correcta conexión de los motores de continua la cual es la siguiente, la salida OUT1 y OUT3 van conectadas al MD (motor derecho) mientras que las salidas OUT2 y OUT4 están conectadas al MI (motor izquierdo).

Estas conexiones se deben al funcionamiento del puente en H. Este está compuesto principalmente por 4 interruptores, los cuales se accionan de 2 en 2. En la ilustración número 27 activamos los interruptores 1 y 4, así el positivo le llega al motor por el borne derecho y el motor gira en sentido de las agujas del reloj.

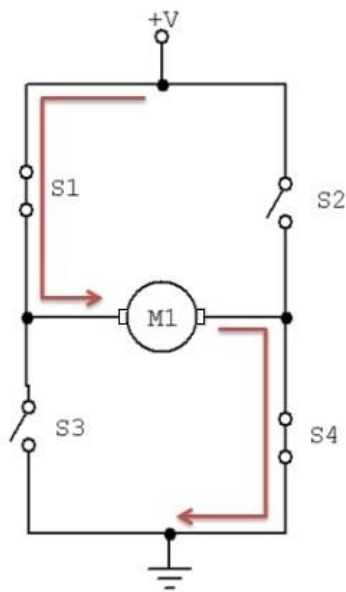


Ilustración 27: Puente H giro sentido horario

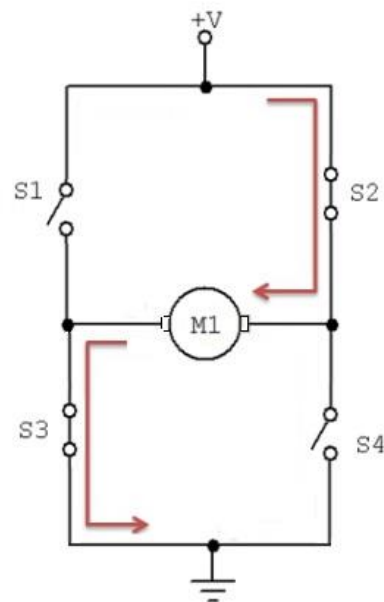


Ilustración 28: Puente H giro sentido antihorario

En la ilustración número 28, desactivamos los interruptores 1 y 4 y activamos los interruptores 2 y 3, de esta forma el positivo le llega al motor por el borne izquierdo y el motor gira en sentido anti horario.

Con esto ya podríamos poner en funcionamiento nuestro robot pero este no sabría que hacer ya que no recibiría datos de su entorno para poder realizar las funciones para las que está diseñado.

Para esto se añade el sensor de ultrasonidos (HCSR04) el cual podemos observar en la ilustración 22. Para el conexionado de este se han utilizado las salidas RA1 y RA2 del microcontrolador (PIC16F88).

La salida RA1 está conectada al pin Trigger del sensor mientras que la RA2 se conecta al pin Echo. A los demás pines del sensor se conectan al izquierdo respecto al esquemático un toma de +5V y al derecho una toma de tierra.

6.1 Esquemático completo función evita obstáculos

Con este esquemático ya se puede realizar la función evita obstáculos del robot, ya que ya dispone de todos los elementos necesarios para recibir datos de su entorno con el sensor de ultrasonidos y procesarlos en el microcontrolador. Una vez en el microcontrolador, este tomará una serie de decisiones y enviará datos al puente en H y este a su vez a los motores para girar en el sentido correcto para no colisionar con el entorno.

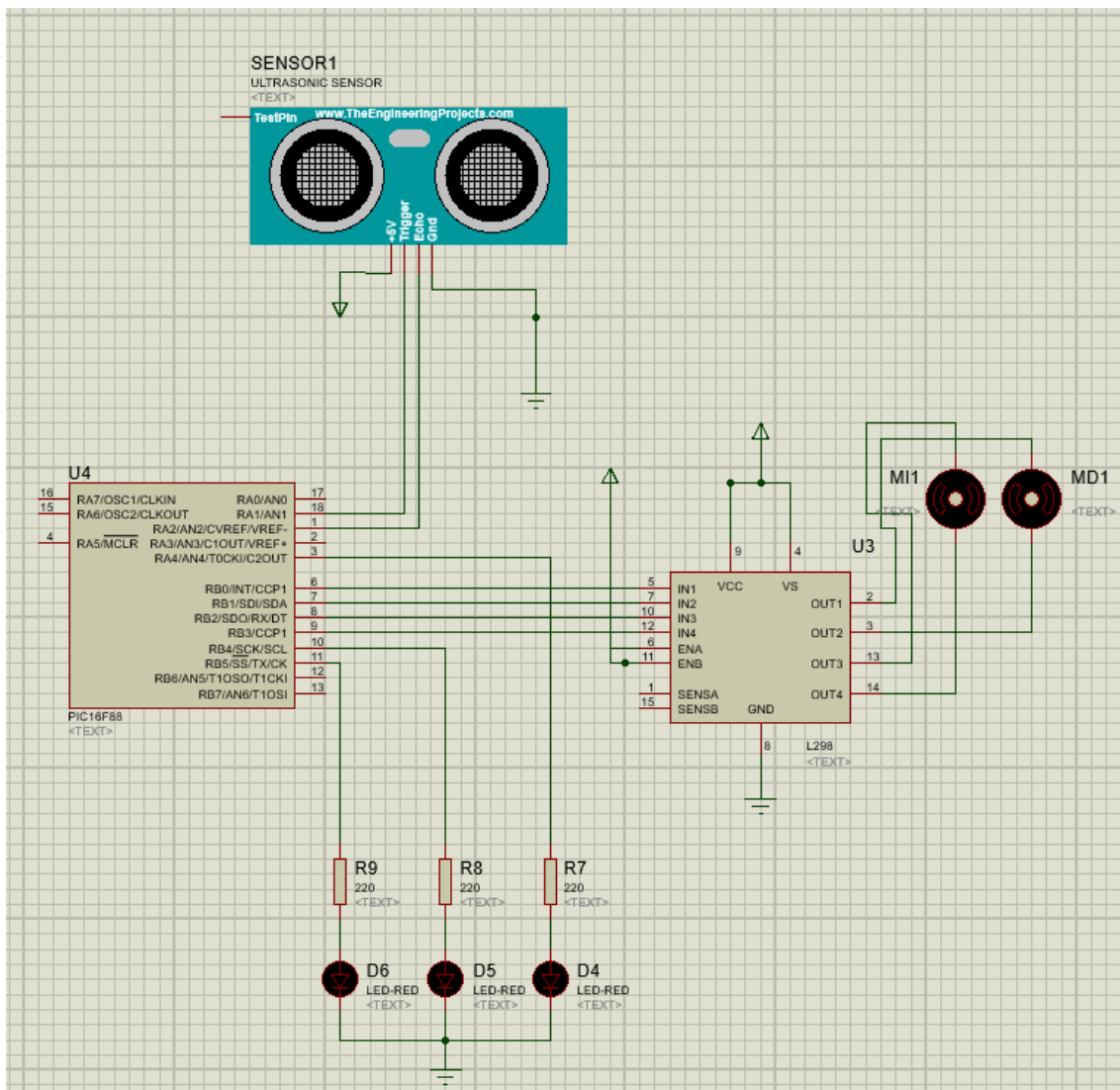


Ilustración 29: Esquemático completo función Evita obstáculos

6.2 Esquemático completo función control remoto

En el siguiente esquemático se puede observar el receptor infrarrojo necesario para el funcionamiento mediante control remoto. El conexionado del mismo es simple ya que únicamente posee tres pines, de los cuales el pin VCC va conectado a +5V y el pin GND va conectado a tierra. El pin 3 es el encargado de sacar los datos recibidos por el sensor, va conectado al pin RA3 del microcontrolador para enviar los datos recibidos del control remoto al mismo.

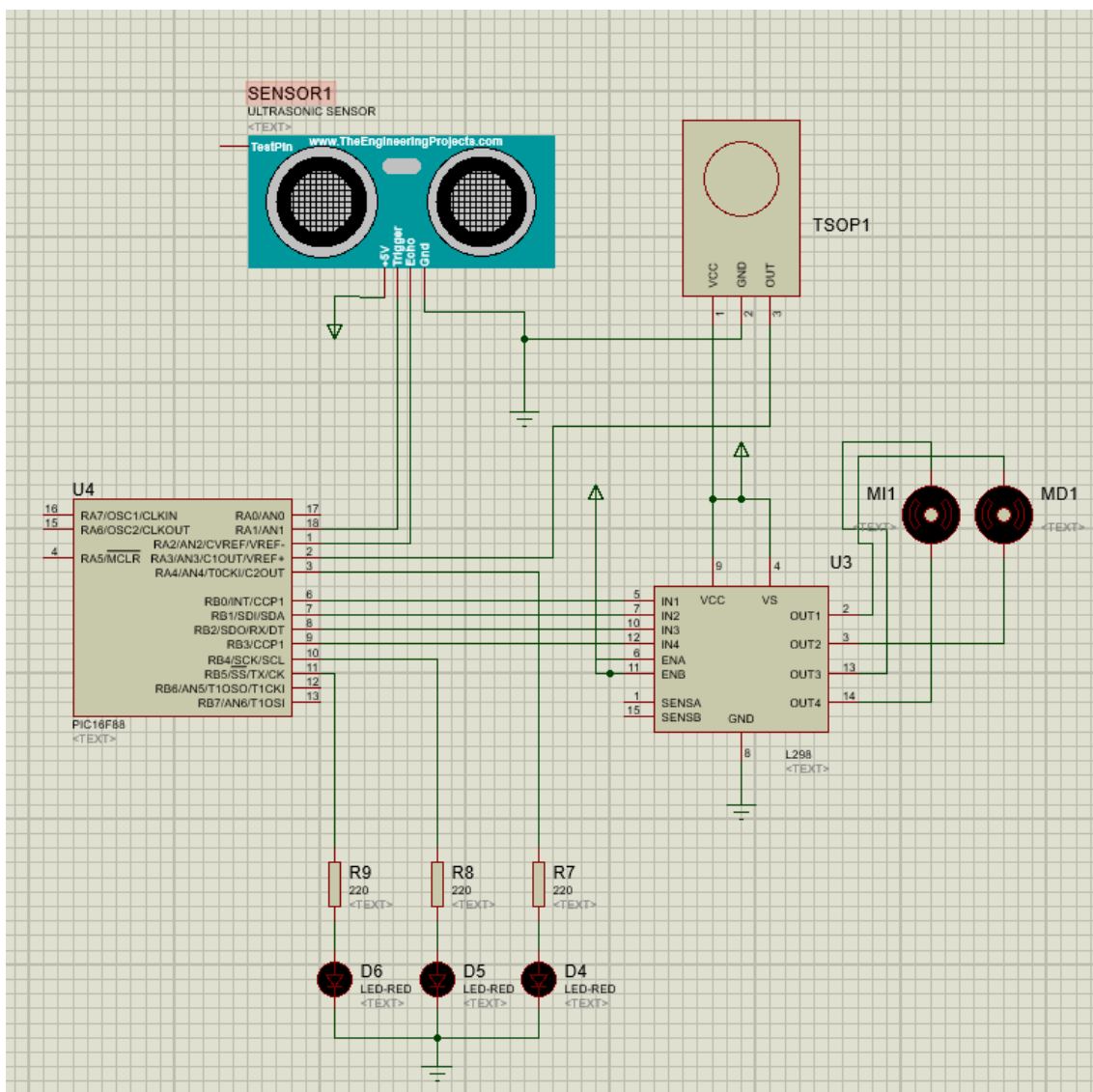


Ilustración 30: Esquemático completo función Control remoto

6.3 Esquemático función sigue líneas

En este módulo se conectan los componentes necesarios para realizar la función sigue líneas, es decir los tres sensores CNY70 cuya función es recopilar los datos necesarios para no salirse del recorrido indicado.

En el esquemático se observan los 3 sensores SI (sensor izquierdo) SC (sensor central) y SD (sensor derecho). Todos ellos van conectados a una entra de +5V por la patilla derecha del componente según la imagen y por la otra conectados a una resistencia de 10KΩ conectada a masa y al microcontrolador mediante el siguiente orden: SI conectado a la entrada RA6, SC conectado a la entrada RA7 y el SD conectado a la entrada RA.

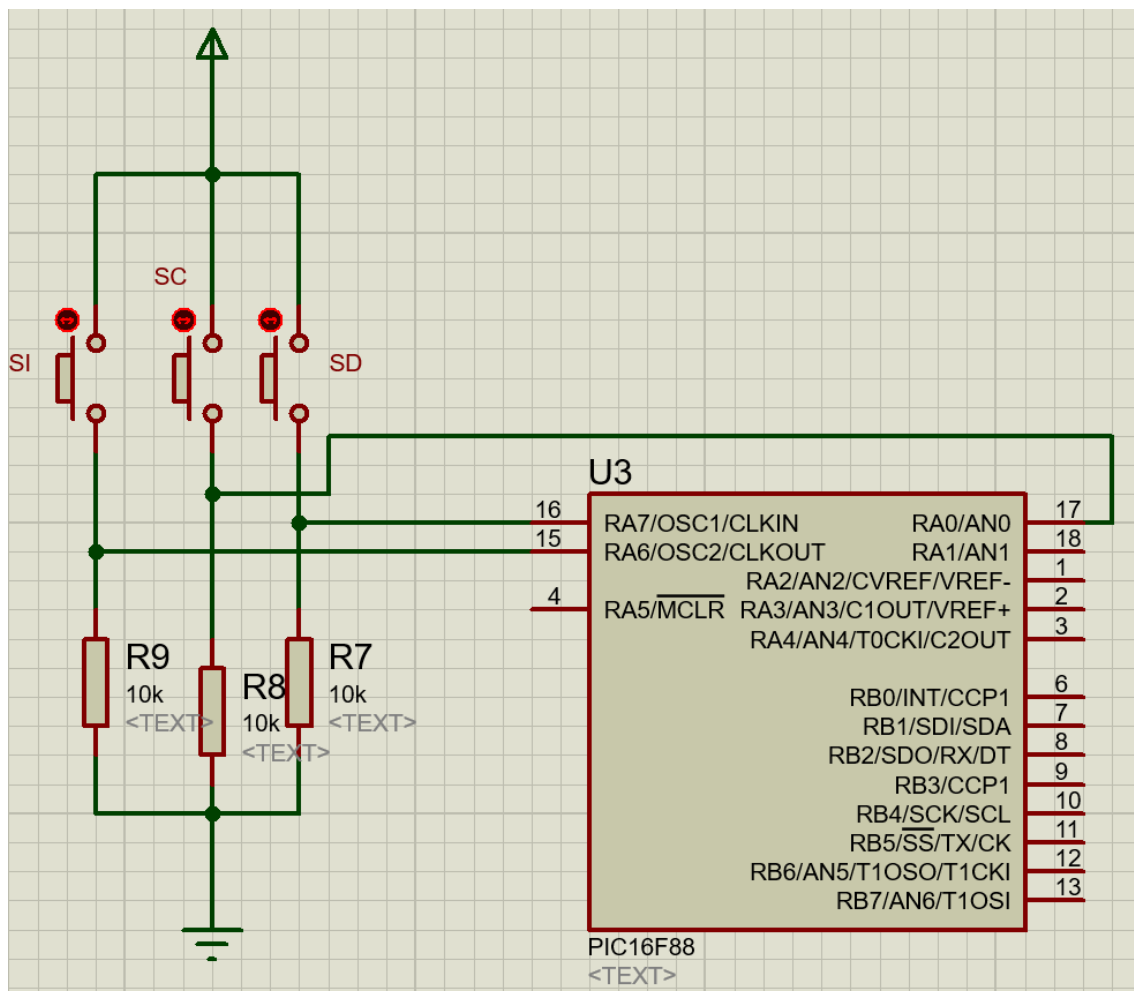


Ilustración 31: Esquemático conexión módulo sigue líneas con microcontrolador

6.4 Esquemático en ARES para el diseño de PCB

Se ha decidido dividir el circuito en dos placas de circuito impreso para evitar tener una cantidad de cables innecesarios los cuales pueden provocar malos contactos y con esto fallos.

Se ha decidido hacer una PCB para el módulo sigue líneas y otra para el resto de componentes.

Este sería el resultado de pasar el esquemático de la ilustración 24 a PCB con ayuda de la herramienta ARES incluida en el programa PROTEUS.

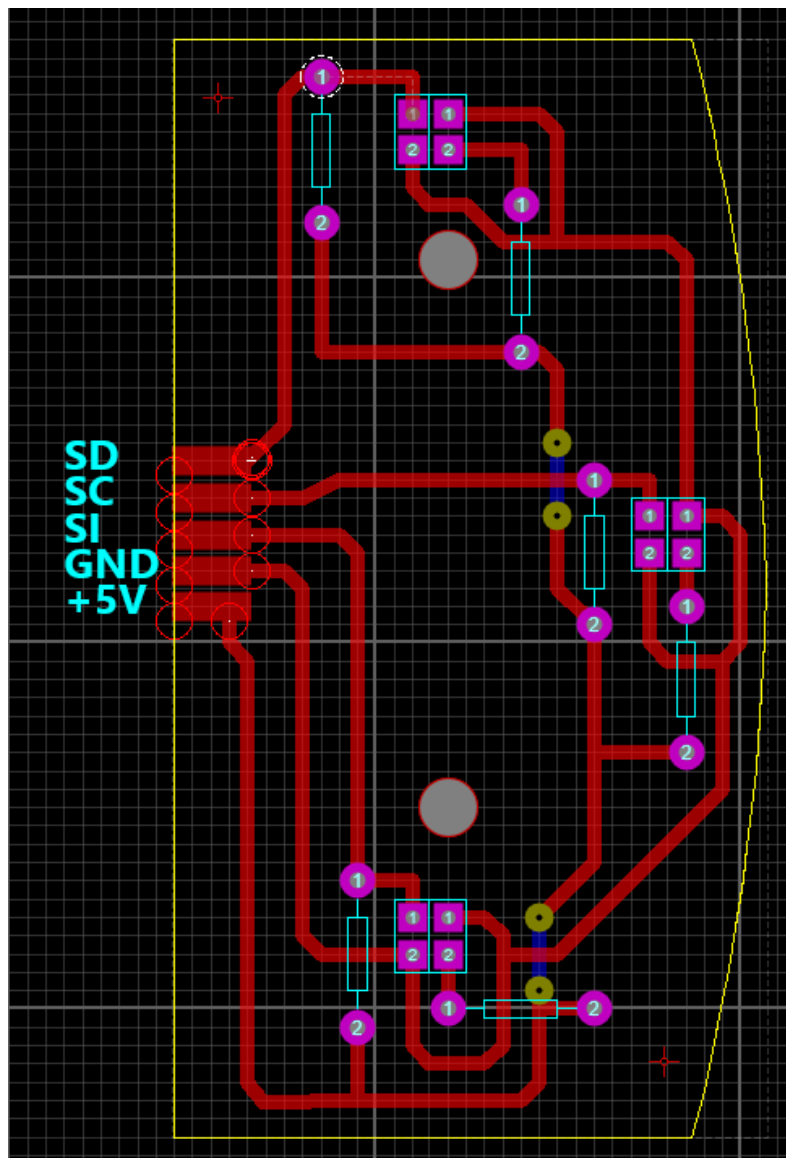


Ilustración 32: PCB módulo sigue líneas ARES

En la ilustración 26, se observa el diseño de la PCB central del robot en el cual irán conectados todos los componentes que el robot necesita tal y como se muestra en el esquemático de la ilustración 19.

Esta tarea ha sido realizada con ayuda de la herramienta ARES incluida en PROTEUS.

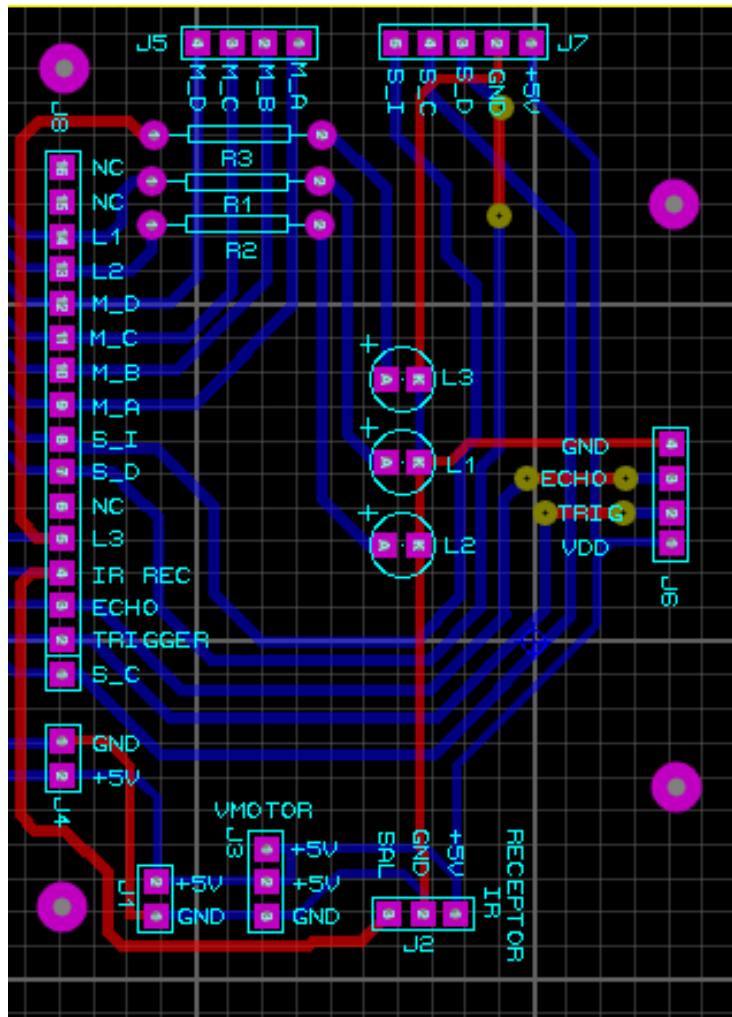


Ilustración 33: PCB Central ARES

En las siguientes imágenes se muestran los resultados de la fabricación de las PCB en placas de cobre para circuito impreso. Estas serán usadas para el conexionado de todo el circuito del robot facilitándonos mucho esta tarea.

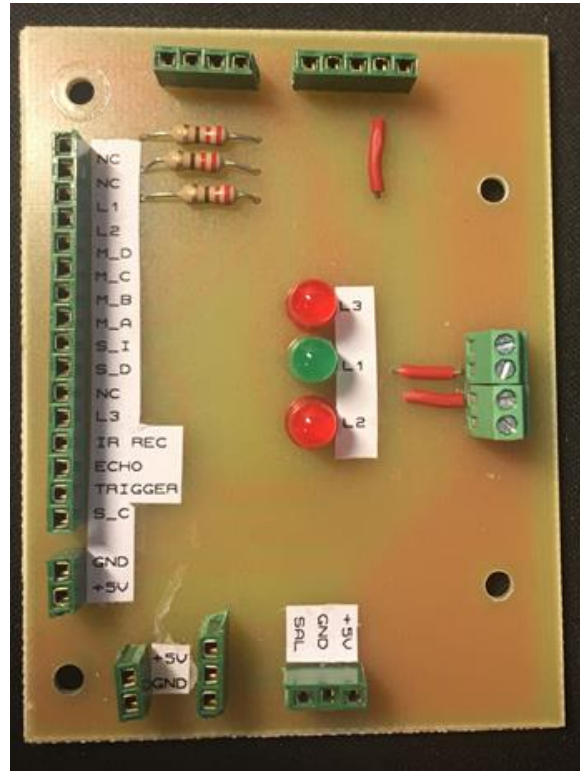


Ilustración 34: PCB Central

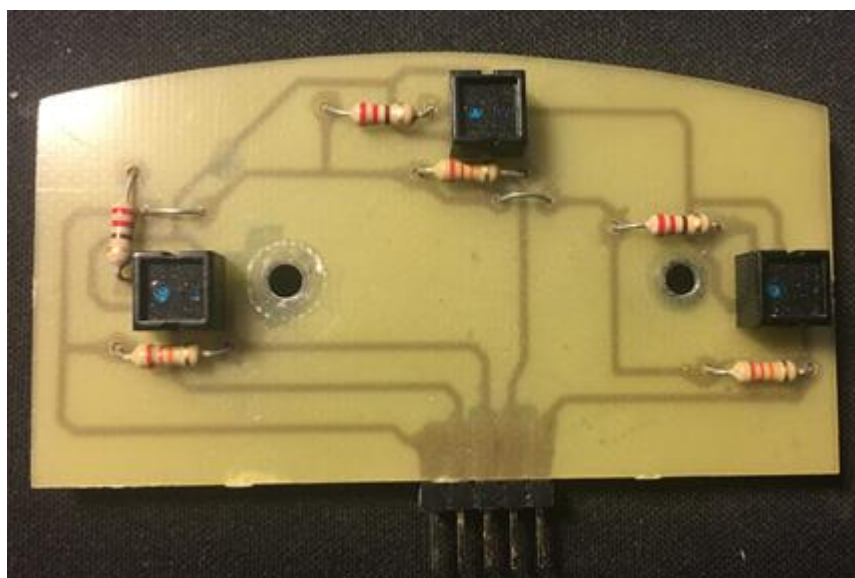


Ilustración 35: PCB módulo sigue líneas

7. Montaje

A continuación se van a detallar los pasos seguidos para el montaje de toda la estructura del robot en la cual irán montados los componentes del mismo.

El primer paso ha sido el montaje del chasis, en este paso se han montado los motores, las ruedas y la rueda libre. Lo primero a realizar es el montaje de los motores al chasis de metacrilato

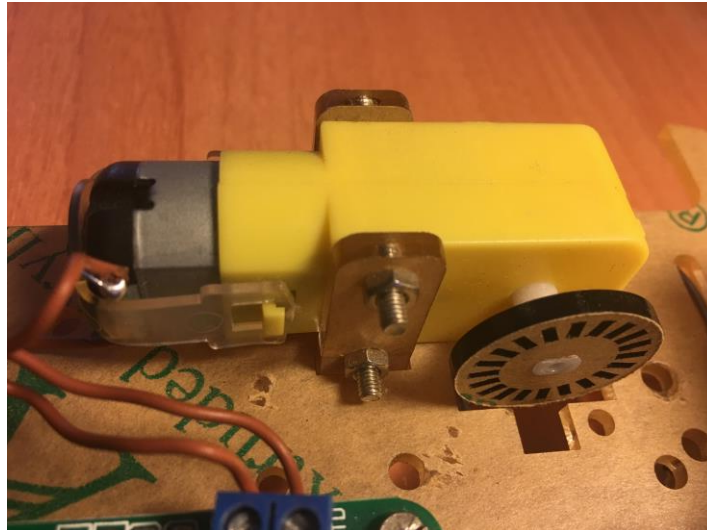


Ilustración 36: Montaje motor al chasis

Para la unión de ambos motores al chasis únicamente se necesitan las piezas que vienen en el pack comprado, se colocan dos tornillos atravesando cada motor para fijarlo y se sueldan dos cables a cada patilla del motor para conectarlos posteriormente al L298.

Una vez colocados los dos motores y soldados los respectivos cables, se procede a colocar las ruedas a cada motor. A continuación se atornilla la rueda libre a la parte trasera del chasis con cuatro tornillos y se coloca una pequeña cantidad de silicona para dejarla fija y evita con esto que se desplace en exceso dejándola como una rueda fija. Debe quedar como en la siguiente imagen.

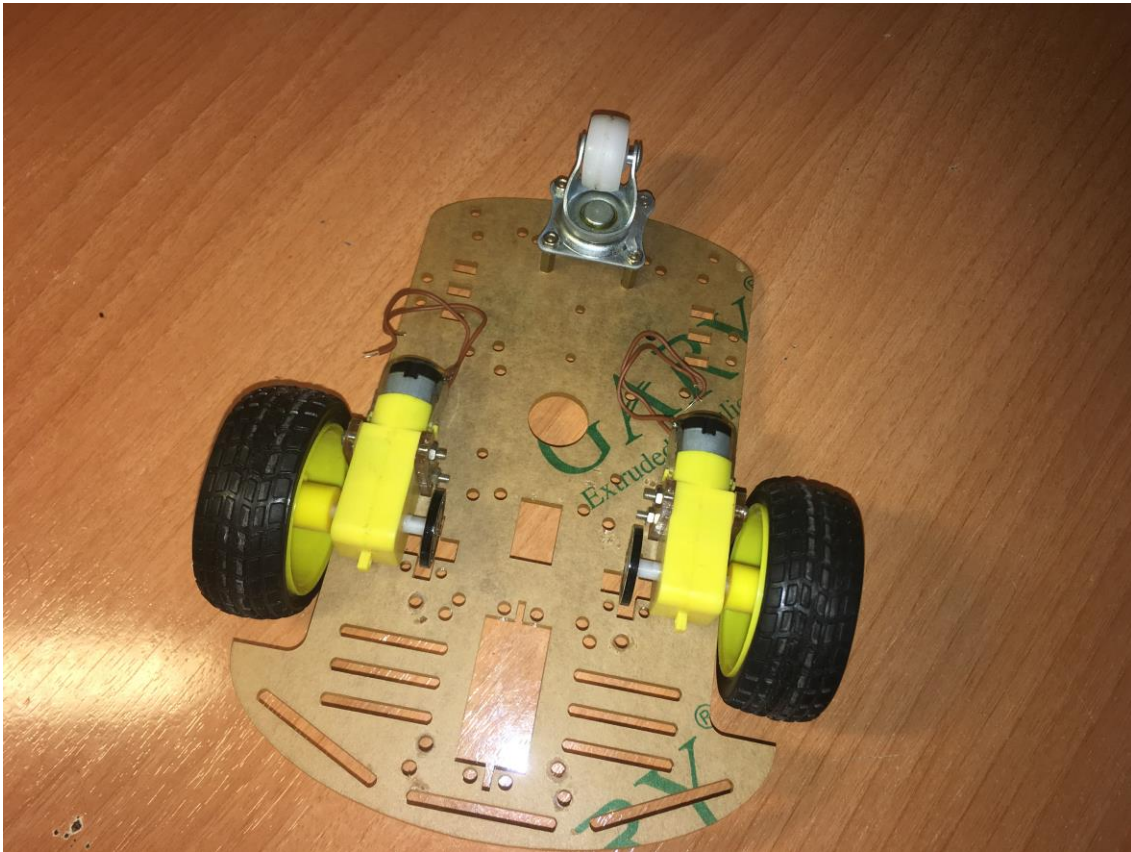


Ilustración 37: Chasis con ruedas, motores y rueda libre montado

Una vez montadas las ruedas y los motores, la siguiente pieza en colocar es el puente en H y la PCB que contiene los sensores CNY70 y se encarga de la función sigue líneas. Para fijar el puente en H se utilizan dos tornillos con sus respectivas tuercas y se fija en la parte de abajo del chasis entre ambos motores. Con esto facilitamos la conexión entre los motores y el driver.

A continuación fijamos el módulo sigue líneas en la parte de abajo del chasis lo más adelante posible. Para este paso se han utilizado unos extensores de los tornillos para poder dejar a una distancia aproximada del suelo de 1 Cm. Con esto conseguiremos un funcionamiento óptimo de los sensores.

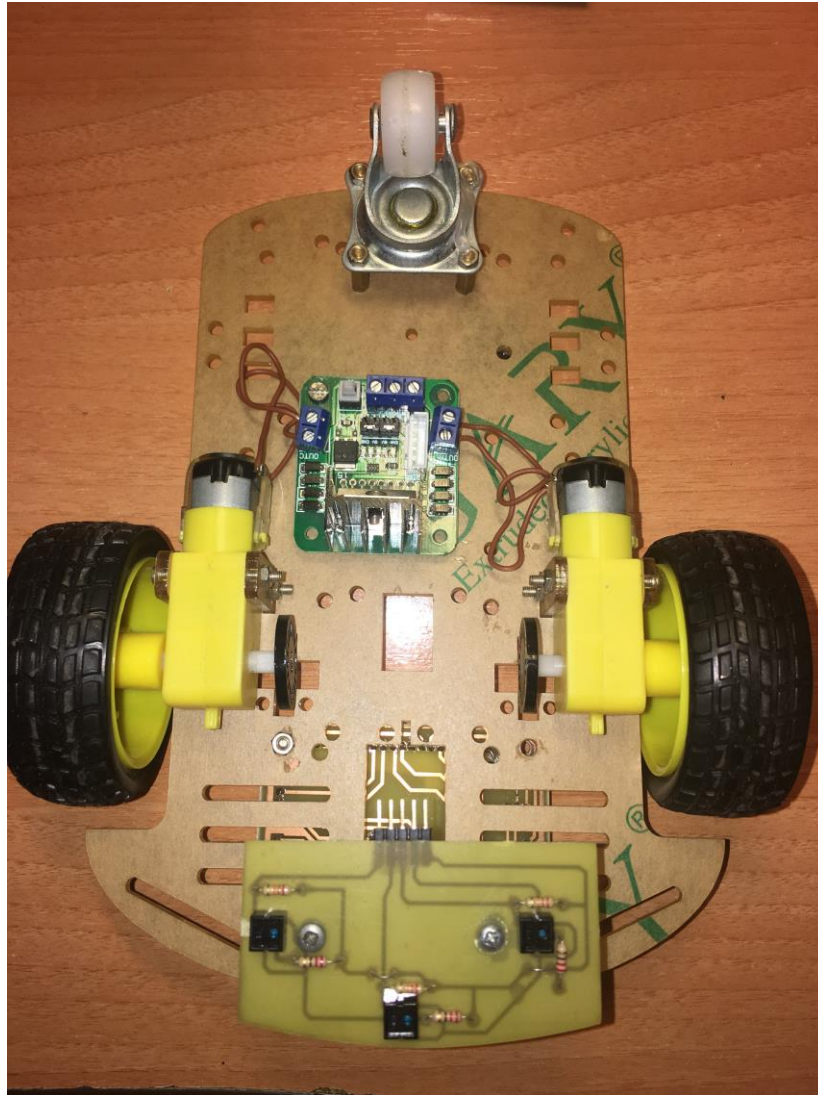


Ilustración 38: Chasis con módulo sigue líneas y Driver montados

El siguiente paso es colocar la PCB central y conectar todos los componentes mediante los cables pertinentes.

Para fijar la PCB a la parte superior se han utilizado 4 alzadores atornillados a la placa y se ha colocado en la parte delantera para facilitar la medición del sensor de ultrasonidos. Una vez fijada la PCB se procede a conectar todos los componentes entre sí. Primero conectamos los motores al driver, después se conecta el driver a la PCB para llevar los datos de la PCB al driver y de ahí a los motores mediante los pines hembra que tiene esta, también conectamos los pines de alimentación del driver a la PCB. Conectamos a continuación el sensor de ultrasonidos HCSR04 y el sensor de infrarrojos TSOP44xx a la PCB. Por último conectamos el módulo sigue líneas a la PCB, junto a la batería y el PIC16F88.

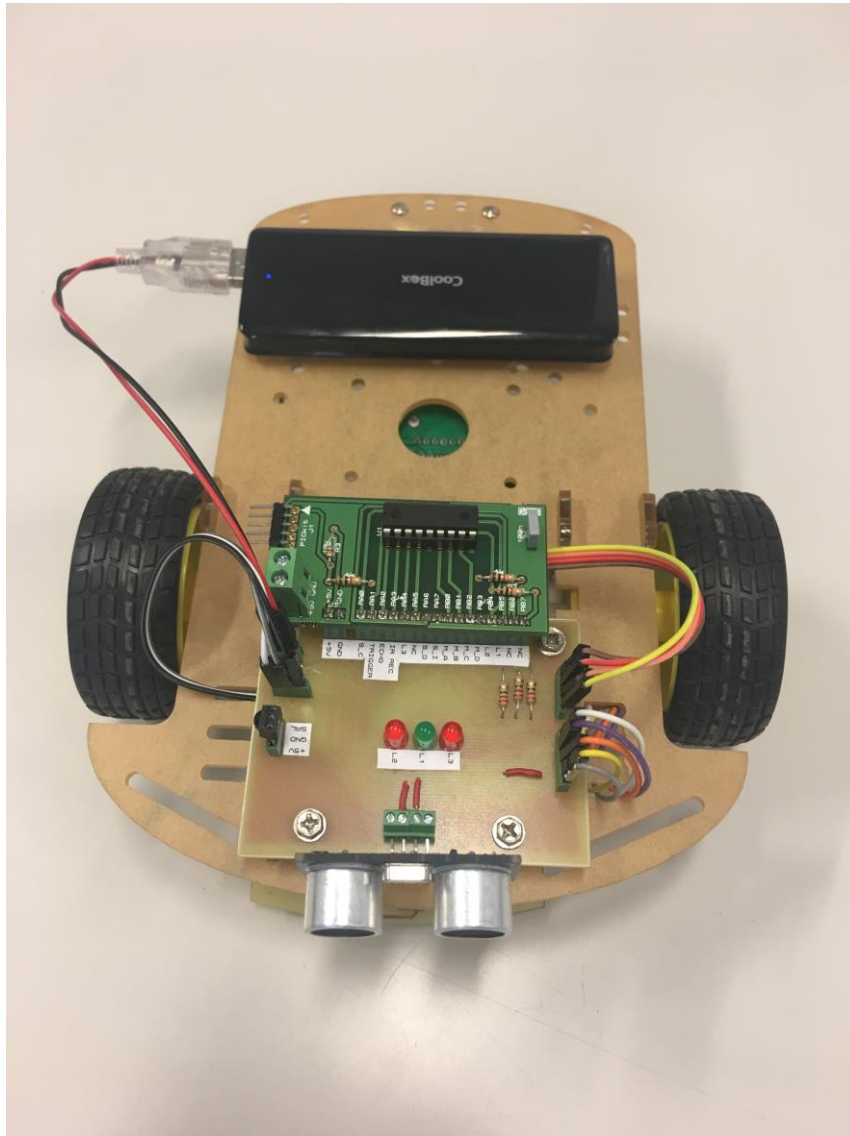


Ilustración 39: Chasis con todas las conexiones

8. PROGRAMACIÓN

En este apartado se van a detallar los pasos a seguir para poder programar el microcontrolador PIC16F88 de manera correcta.

Para poder transferir un código fuente a un microcontrolador lo primero que tenemos que tener en cuenta es si nuestro PC tiene instalado el programa MPLAB X, también es necesario el uso del kit de desarrollo del proveedor llamado PICKIT el cual proporciona Microchip para poder comunicar un ordenador con el PIC.

El programa MPLAB X se utilizará para desarrollar el código fuente y a posteriori para transferir el código al PIC mediante el PICKIT conectado al USB del PC. Una vez se tenga el código fuente correcto, conectaremos el PICKIT al PC, buscaremos el archivo .HEX del código que se quiere transferir y pulsaremos "Write". Una vez nos aparezca en verde en la pantalla procederemos a desconectar todo y ya tendremos el código en el PIC listo para conectar al robot.

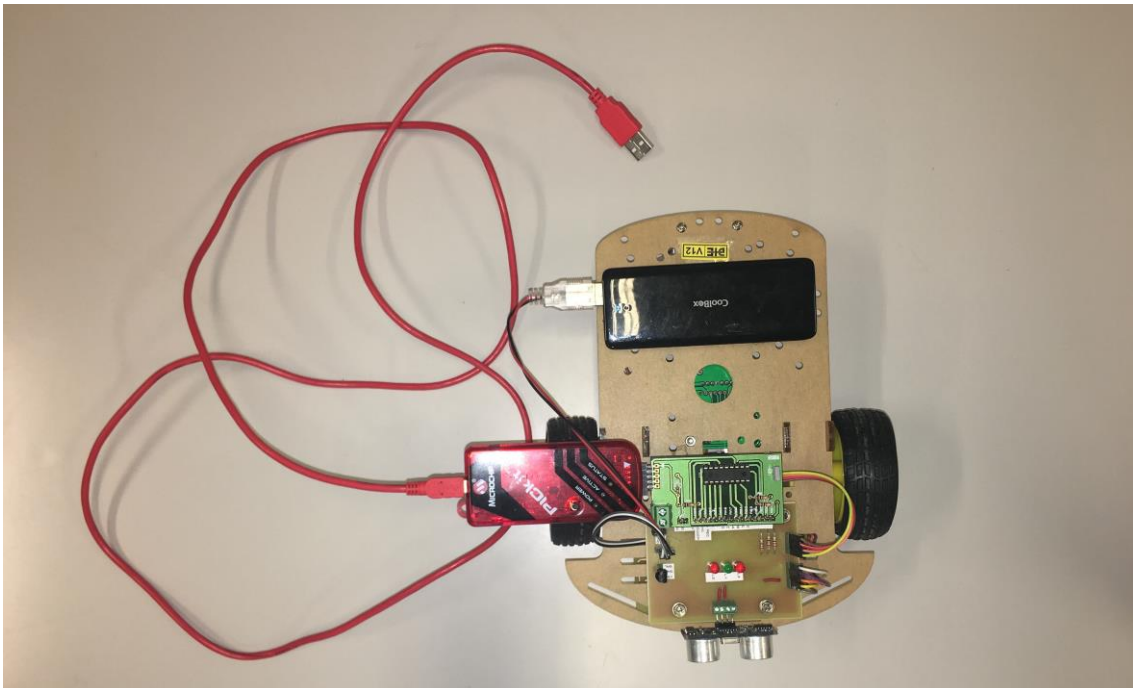


Ilustración 40: Conexión Pickit

En la ilustración número 40 se muestra el modo de conectar el Pickit al robot para así transferir el código. El PIC16F88 se conecta a la PCB por medio de otra PCB desarrollada en la universidad por tanto no se a detallado su elaboración.

Esta PCB tiene unos pines para conectar el Pickit a la misma y facilitar la transferencia de código del ordenador al robot mediante un cable USB

8.1 Código cabecera

```
#include <pic16f88.h>
#include <xc.h>

// CONFIG1
#pragma config FOSC = INTOSCIO // Oscillator Selection bits (INTRC oscillator; port I/O
function on both RA6/OSC2/CLKO pin and RA7/OSC1/CLKI pin)
#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config MCLRE = OFF // RA5/MCLR/VPP Pin Function Select bit (RA5/MCLR/VPP
pin function is MCLR)
#pragma config BOREN = OFF // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF // Low-Voltage Programming Enable bit (RB3 is digital
I/O, HV on MCLR must be used for programming)
#pragma config CPD = OFF // Data EE Memory Code Protection bit (Code protection
off)
#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write
protection off)
#pragma config CCPMX = RB0 // CCP1 Pin Selection bit (CCP1 function on RB0)
#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code
protection off)
#pragma config DEBUG = OFF
// CONFIG2
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock
Monitor disabled)
#pragma config IESO = OFF // Internal External Switchover bit (Internal External
Switchover mode disabled)

#define _XTAL_FREQ 4000000

//PERIFÉRICOS UTILIZADOS EN LA PLACA DE EXPANSIÓN

#define LED_R PORTBbits.RB1

void main(void) {
    TRISA=0b00011100; //configuración de puerto A
    TRISB=0b00000001; //Puerto B todo salidas excepto RB0
    OSCCON = 0b01101100; /*BITS 6,5,4 ESTABLECEN LA FRECUENCIA. 110 -> 4MHz*/
    ANSEL = 0b00000000; //todo digital
    OPTION_REGbits.nRBPU = 0; //Habilita resistencias de pull-up en las entradas del
puerto B
    PORTB=0;
    PORTA=0;
}
```

8.2 Código Función sigue líneas

Función sigue líneas

```
int estado=0;
int led=0;
    while(1){
switch (estado)
{case 0;
    {MI_F=1; MD_F=1;
    while((S_I==0) &&(S_C==0) &&(S_D=0)) {}
    if((S_I==0) &&(S_C==1) &&(S_D==0))
        {estado=1;
        break;}
    else if((S_I==0) &&(S_C==0) &&(S_D==1))
        {estado=2;
        break;}
    else if ((S_I==1) &&(S_C=0) &&(S_D==0))
        {estado=3;
        break;}
    else{
        estado=0;
        break;}}
case 1:
    {MI_F=1; MD_F=1;
    while(((S_I==0) &&(S_C=1) &&(S_D==0)) || ((S_I==0) &&(S_C=0)
        ) &&(S_D==0)) {}
    if((S_I==0) &&(S_C=0) &&(S_D==1))
        {estado=2;
        break;}
    else if ((S_I==1) &&(S_C=0) &&(S_D==0))
        estado=3;
        break;}
    else
        {estado=1;
        break;}}
```

case 2:

```
    {MI_F=1; MD_F=0;
while(((S_I==0)&&(S_C=0)&&(S_D==0))||((S_I==0)&&(S_C=0
    )&&(S_D==1))){}
if((S_I==0)&&(S_C=1)&&(S_D==0))
    {estado=1;
    break;}
else if ((S_I==1)&&(S_C=0)&&(S_D==0))
    estado=3;
    break;}
else
    {estado=2;
    break;}}
```

case 3:

```
    {MI_F=0; MD_F=1;
while(((S_I==1)&&(S_C=0)&&(S_D==0))||((S_I==0)&&(S_C=0
    )&&(S_D==0))){}
if((S_I==0)&&(S_C=1)&&(S_D==0))
    {estado=1;
    break;}
else if ((S_I==0)&&(S_C=0)&&(S_D==1))
    estado=2;
    break;}
else
    {estado=3;
    break;}}
if((S_C==0))
    {LV=1;}
else if ((S_I==0)&&(S_C=0)&&(S_D==0))
    {LV=0;}
else{LV=0;}
if(S_I==0)
    {LR_I=1;}
else{LR_I=0;}
if(S_D==0)
    {LR_D=1;}
else{LR_D=0;}
}
return;
```

8.2 Código función evita obstáculos

```
void avanzar()
{
    in1=1;
    in2=0
    in3=1
    in4=0
    led2=1
}
void giraD()
{
    in1=1;
    in2=0;
    in3=0;
    in4=1;
    led3=1;
}
void giraI
{
    in1=0;
    in2=1;
    in3=1;
    in4=0;
    led1=1;
}
void retroceder()
{
    in1=0;
    in2=1;
    in3=0;
    in4=1;
    led1=1;
}

void parar()
{
    controlo;
    in1=0;
    in2=0;
    in3=0;
    in4=0;
    led1=1;
    led3=1;
    controlo=dist();
    if (controlo<6){
        retroceder();
    }
}
```

```

void main (void){
    float contrololo;
    TRISA=0b11101101;
    TRISB=0b00000000;
    ASCCON=0b01101100;
    ANSEL=0b00001000;
    OPTION_REGbits.nRBPU=
    while(1){
        contrololo=dist();
        if(contrololo>30){
            avanzar();
        }
        if(contrololo<30){
            parar();
            __delay_ms(300);
            giraD();
            __delay_ms(300);
            parar();
            __delay_ms(300);
            contrololo=dist();
            if(contrololo>30){
                avanzar();}
            if(contrololo<=30){
parar();
__delay_ms(300);
                girarI();
                __delay_ms(600);
                parar();
                __delay_ms(300);
                contrololo=dist();
            if (contrololo>30){
                avanzar();}
            if(contrololo<=30){
                parar();
                __delay_ms(3600);
                giraI();
                __delay_ms(300);
                parar();
                __delay_ms(300);
            }
        }
    }
    return;}

```

8.4 Código control Remoto

```
void main (void){
    TRISA=b11101101;
    TRISB=b00000000;
    OSCCON=0b01101100;
    ANSEL=0b00000000;
    PORTB=0;
int tecla;
tecla=captura_SONY();

if (tecla==4){
    MI_F=1;
    MD_F=1
    __delay_ms(100);
    if(tecla==3){
        MI_F=0;
        MD_F=1;
        __delay_ms(100);
    }
    if(tecla==5){
        MI_F=1;
        MD_F=0;
        __delay_ms(100);
    }
    if(tecla==7){
        MI_F=1;
        MD_F=1;
        __delay_ms(100);
    }
    if(tecla==9){
        MI_F=0;
        MD_F=0;
        __delay_ms(100);
    }return;}}
```

8.5 Diagramas de flujo del código

Diagrama de flujo de la función sigue líneas

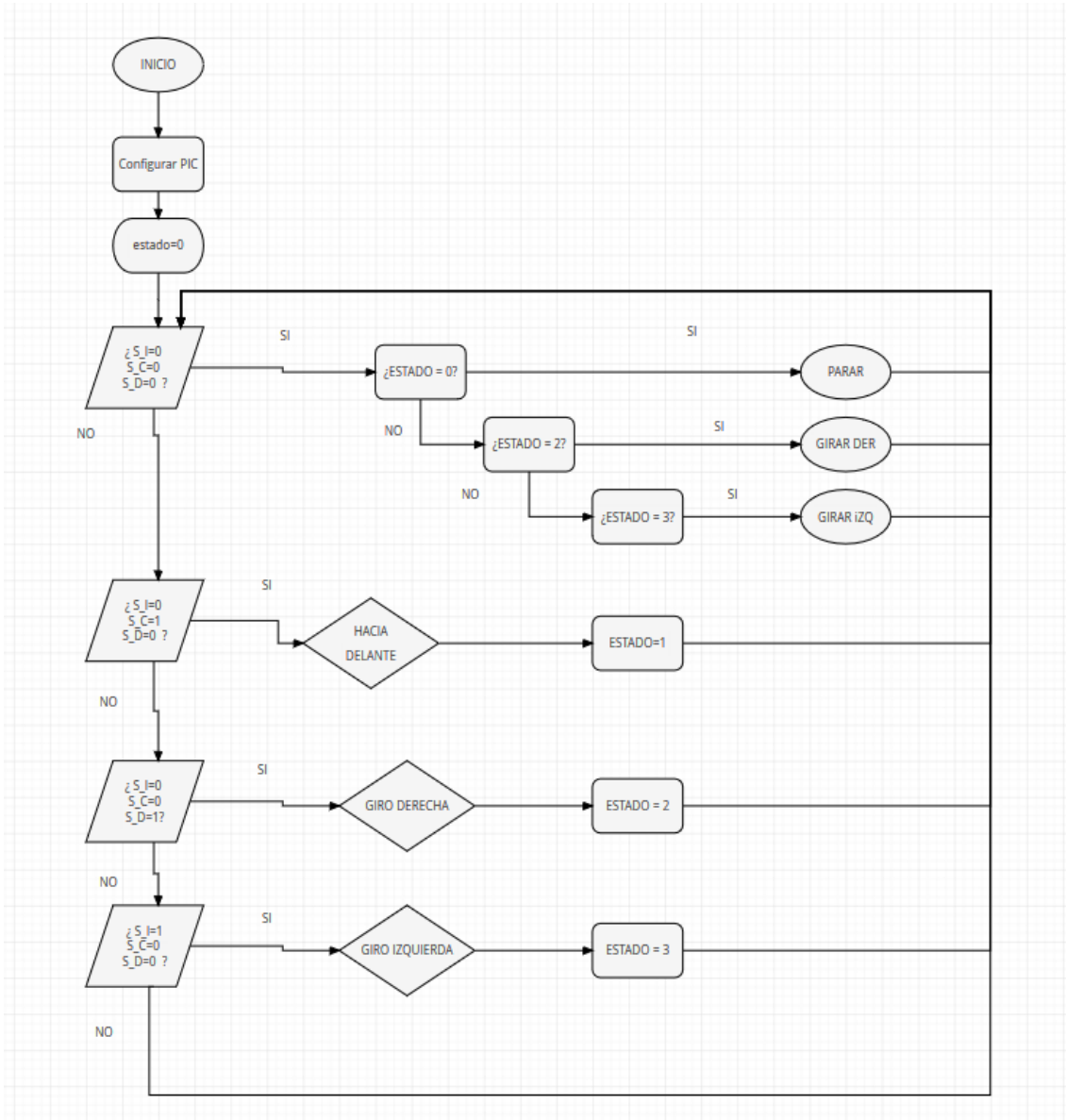


Ilustración 41: Diagrama de flujo sigue líneas

Diagrama de flujo función evita obstáculos

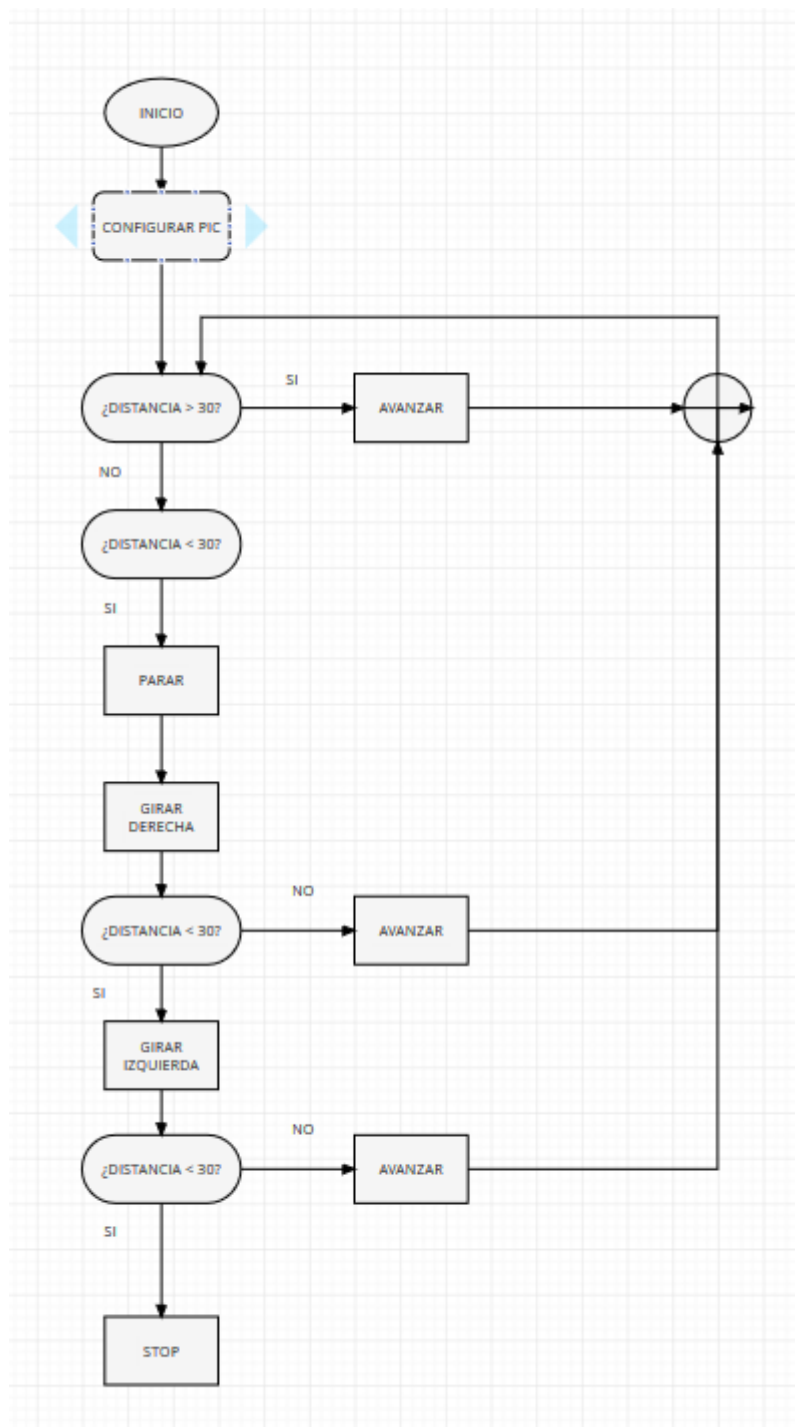


Ilustración 42: Diagrama de flujo evita obstáculos

Diagrama de flujo control remoto

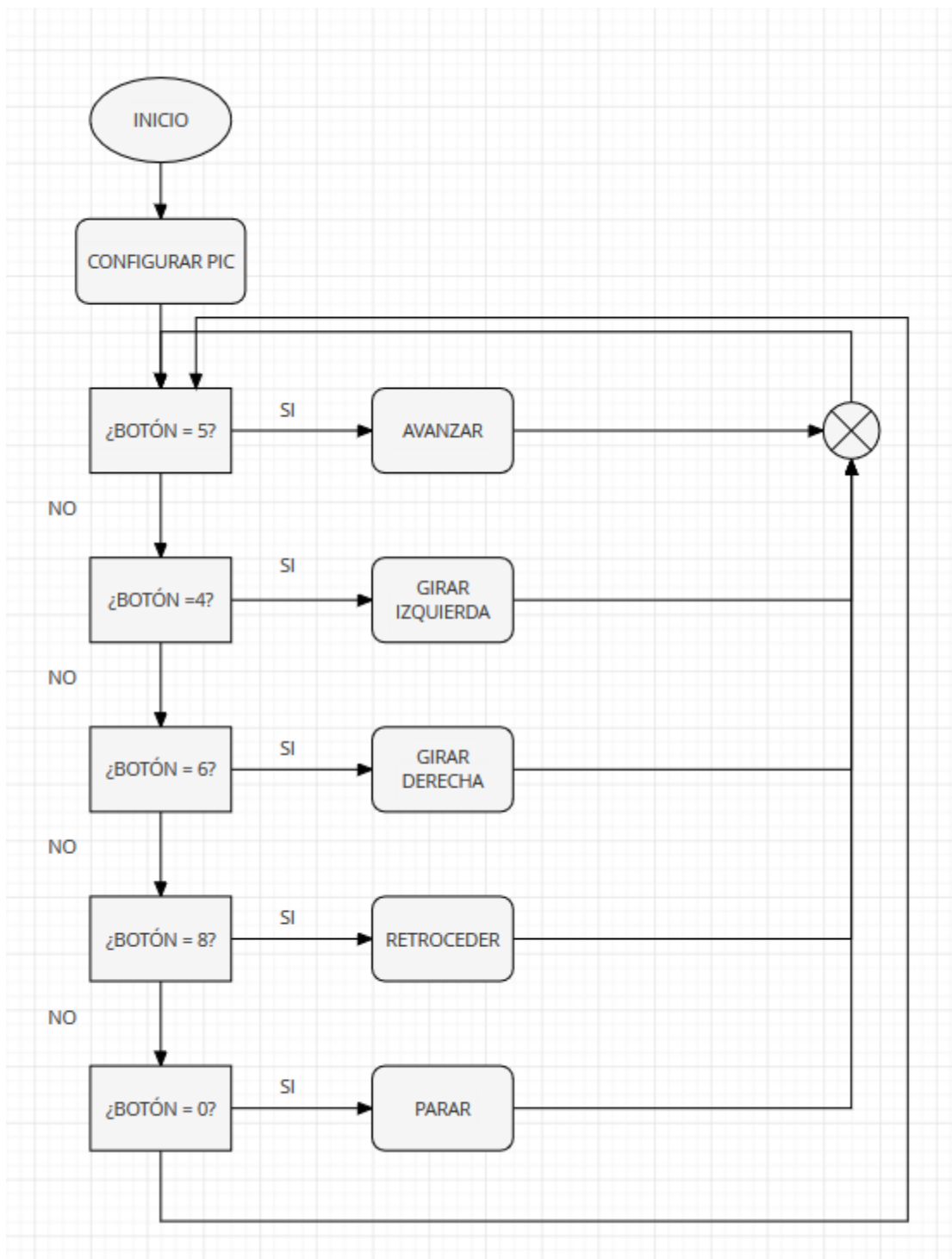


Ilustración 43: Diagrama de flujo control remoto

9. Sistemas y subsistemas

El proyecto realizado se puede dividir en dos sistemas: el mecánico y el electrónico.

9.1 Sistema mecánico

El sistema mecánico corresponde a todos los componentes que forman parte del chasis, y ayudan al movimiento del robot, como son las ruedas, motores etc. Gracias a estos el robot puede desplazarse por su entorno para facilitar la adquisición de datos del mismo

9.2 Sistema electrónico

El sistema electrónico engloba todo lo relacionado con el funcionamiento y control del robot. Este sistema puede subdividirse a su vez en tres.

- El primero está relacionado con la alimentación, es decir, la tensión y la corriente, como la batería que proporciona el voltaje necesario para el correcto funcionamiento del circuito.
- El segundo es el de adquisición de datos, y en él podemos encontrar el microcontrolador, los sensores de ultrasonidos y los sensores infrarrojos. Este subsistema engloba las diferentes partes mediante las cuales se analizan las acciones que deben llevarse a cabo. Por ejemplo, con el sensor infrarrojo se mide la cantidad de luz y según los datos recogidos, se pasa al tercer subsistema: control.
- El subsistema de control que analiza los datos obtenidos y es el encargado de realizar las diferentes acciones. El PIC o arduino junto al driver forman parte de él, uno controla la dirección de giro del robot y el otro analiza los movimientos que debe realizar el robot en función de los datos obtenidos por los sensores tras realizar las debidas lecturas. En este subsistema también se ha añadido la placa board, ya que aquí se encuentran todos los componentes conectados

9.3 Diagrama de sistemas

Para visualizar estos sistemas y subsistemas se realiza el siguiente diagrama de flujo:

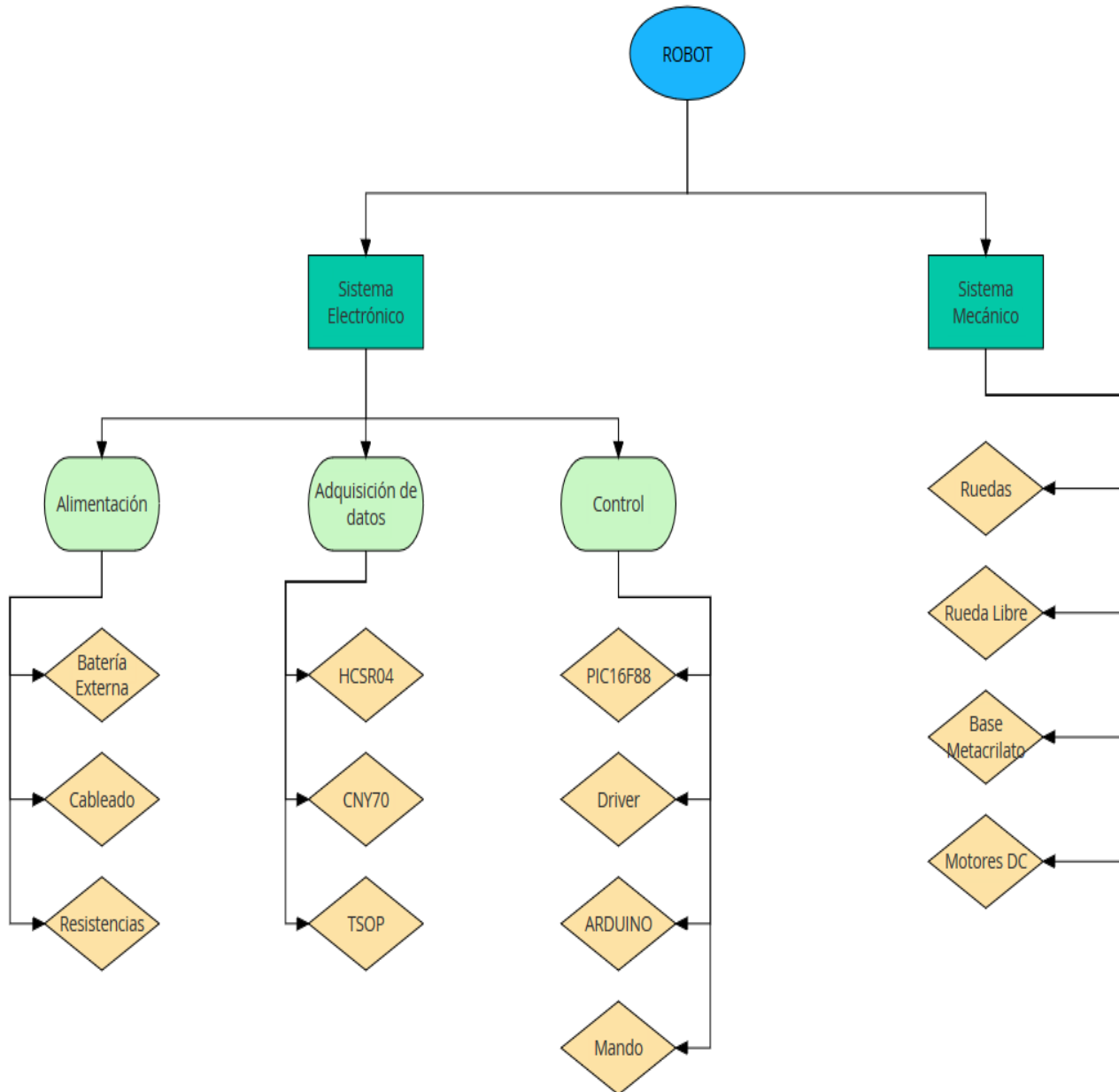


Ilustración 44: Diagrama de Flujo de los sistemas del robot.

10. Conclusiones

El objetivo de este proyecto ha sido la elaboración de un robot basado en un microcontrolador que sirviese a los alumnos de la ETSID a aplicar en el laboratorio los conocimientos teóricos aprendidos en las aulas.

Una vez concluido el proyecto se ha podido comprobar el correcto funcionamiento y por tanto cumplimiento de las exigencias establecidas por el tutor:

Las exigencias eran elaborar tres métodos de funcionamiento mediante el cual los alumnos en grupos de 3 ó 4 personas participasen en una competición en la cual habrá tres pruebas diferentes.

- La primera prueba, el robot debe ser capaz de evitar los obstáculos a su paso correctamente y sin colisionar con ayuda de un sensor de ultrasonidos.
- La segunda prueba, el robot debe seguir una línea dibujada en el suelo sin salirse del recorrido y en el menor tiempo posible.
- La tercera prueba, el robot tiene que obedecer las indicaciones del alumno que este le mandará mediante un mando universal mediante un haz de infrarrojos.

Las principales metas que se pretendían alcanzar con el desarrollo de robot y se han alcanzado son:

- Aplicar y perfeccionar los conocimientos y habilidades en el campo de la electrónica, impartidas en la carrera. Entre ellas, cabe citar: La necesidad de realizar la circuitería del robot, el cálculo de las corrientes necesarias, la selección de los componentes que fuesen acordes a nuestras necesidades.
- Mejorar el conocimiento de la programación en lenguaje C++
- Afrontar el desarrollo y documentación de un proyecto desde cero

En resumen, los conocimientos que se han adquirido ha dado lugar al desarrollo del montaje y programación del robot, tal y como se planteaba al inicio del texto.

11. Bibliografía

BIBLIOGRAFÍA

- [1] Arduino, "Arduino" [Online]. Available : <https://www.arduino.cc/>
- [2] MicroChip "PIC16F88" [Online] Available:
<http://www.microchip.com/wwwproducts/en/PIC16F88>
- [3] Vishay "Vishay" [Available] : <https://www.vishay.com/docs/83751/cny70.pdf>
- [4] ST " Datasheet L298" [Online] Available :
<http://www.st.com/content/ccc/resource/technical/document/datasheet/82/c/c/3f/39/0a/29/4d/f0/CD00000240.pdf/files/CD00000240.pdf/jcr:content/translations/en.CD00000240.pdf>
- [5] L. Llamas, " Medir distancia con arduino y sensor de ultrasonidos hc-sr04" [Online] Available: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>
- [6] Libertad electrónica "Decodificando el Infrarrojo: SIRC" [Online] Available: <https://libertadelectronica.wordpress.com/2009/12/03/decodificando-el-infrarrojo-sirc/>
- [7] MicroChip "PICkit3" [Online] Available:
<http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=PG164130>

12. Anexos

ANEXO 1: Posibles mejoras

En este apartado se van a tratar las posibles mejoras que podrían ser aplicadas al robot en caso de ser necesario una remodelación en un futuro.

- La primera mejora que se podría llevar a cabo es la adaptación de la PCB central para que el robot pueda ser controlado por un arduino en lugar del PIC16F88. Para ello sería necesario realizar un acople de arduino a las conexiones de la PCB que bien podría ser otra PCB o una shield para arduino que se conecte directamente a la PCB.
- Otra posible mejora que se puede desarrollar sería la conexión de dos ruedas extras con sus respectivos motores de corriente continua para mejorar la tracción la estabilidad y también los ángulos de giro y su precisión.
- También se podría introducir en el circuito un servomotor acoplado al sensor de ultrasonidos para que este se mueva a la vez que el robot avance y así se más eficiente la captación de los objetos que tiene el robot a su alrededor.
- Por ultimo una mejora que podría ser útil en caso de usar el robot al aire libre sería la inclusión de una placa solar para que el robot sea autónomo y no necesite de una batería para su funcionamiento habitual.

¿Qué es Arduino?

Se trata de una plataforma de prototipado de código abierto " open source" cuyos software y hardware son polivalentes y sencillos de usar. El hardware Arduino más sencillo consiste en una placa con un microcontrolador y una serie de puertos de entrada y salida. La diferencia entre los distintos tipos de arduino se encuentra en la tensión utilizada en las placas. Generalmente en los que poseen microcontroladores CortexM3 tiene un voltaje de 3,3 voltios, mientras que la gran mayoría poseen AVR los cuales utilizan una tensión de 5 voltios. Y, por otra parte, el número de conexiones, procesador utilizado, memoria y, sobre todo, el número de entradas y salidas y la posibilidad de alimentar distintos elementos desde la propia placa Arduino.

Arduino se puede utilizar tanto para desarrollar elementos autónomos como bien para conectarse a otros dispositivos o interactuar con otros programas con el hardware o el software. Puede usarse para accionar un motor de corriente continua como para transformar los datos obtenidos de un teclado para transmitirlos a un PC.

Como podemos observar en la siguiente imagen existe gran variedad de placas arduino según las exigencias del proyecto a realizar. Desde arduinos para proyectos más simples, a arduinos usados en el desarrollo de wearables o impresión en 3D como es el caso del "MATERIA 101"



Ilustración 45 : Categorías de Arudino según sus principales usos.

Si ya es numerosa la cantidad de placas de las que se dispone aún lo es más la cantidad de shields que existen para desarrollar sobre estas. Desde GPS, relojes en tiempo real, Bluetooth, pantallas LCD, ultrasonidos etc...

Para el desarrollo de este proyecto se va a utilizar el ARDUINO UNO ya que con este modelo tenemos las necesidades cubiertas en cuanto a conectividad y entradas y salidas.

El arduino uno se trata de una board basada en un microcontrolador Atmega328. Esta tiene 14 pines que pueden actuar tanto de entradas como de salidas digitales, de las cuales 4 pueden emplearse para salidas PWM. Tiene también 6 entradas analógicas, un resonador cerámico de 16 MHz, una entrada Jack para conectar una fuente de alimentación, un conector ICSP, un puerto USB hembra y un botón de reset.

Características

- **Microcontrolador:** ATmega328
- **Voltaje Operativo:** 5v
- **Voltaje de Entrada :** 7 – 12 v
- **Pines de Entradas/Salidas Digital:** 14 (De las cuales 6 son salidas PWM)
- **Pines de Entradas Analógicas:** 6
- **Memoria Flash:** 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
- **SRAM:** 2 KB (ATmega328)
- **EEPROM:** 1 KB (ATmega328)
- **Velocidad del Reloj:** 16 MHZ.

En la siguiente imagen podemos observar la disposición de los componentes físicamente en el arduino:

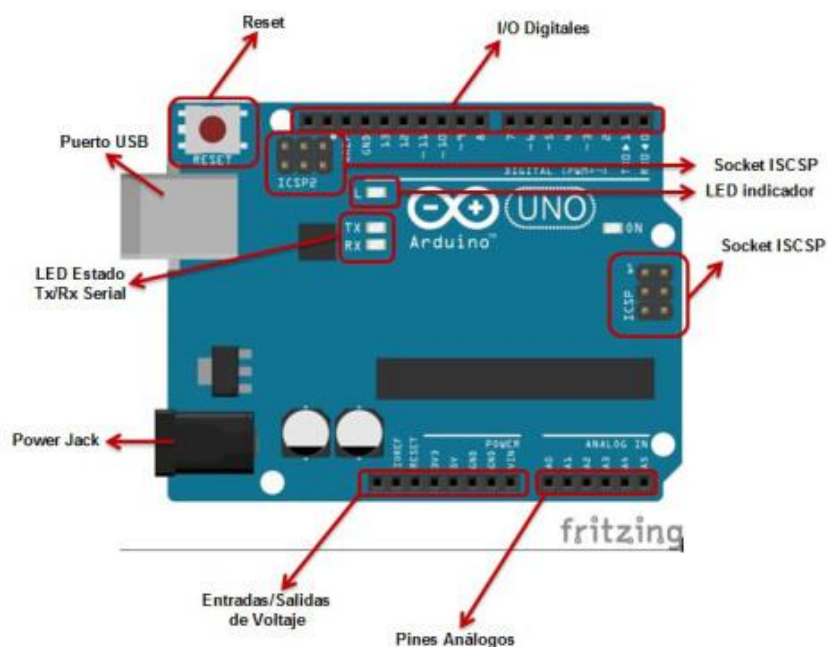


Ilustración 46: Arduino UNO

Software

Para programar Arduino se necesita comunicar el Arduino con la placa mediante un lenguaje de programación, el utilizado por Arduino se llama "Wiring". Es una combinación entre el lenguaje C/C++ y la plataforma Processing.

Para poder trabajar a nivel programación del procesador, se ha de utilizar un software específico, paquetes de librerías que permiten el uso del lenguaje de forma completa. Otra ventaja es el software utilizado por Arduino que tiene licencia libre, es decir, puede descargarse de forma fácil y gratuita desde la página oficial. La versión más reciente es el Arduino 1.8.1

ANEXO 2. PLIEGO DE CONDICIONES

1. Alcance del pliego

El objeto de este documento es fijar las condiciones técnicas mínimas que debe cumplir el robot para uso docente especificando los requisitos de durabilidad, utilidad, funcionamiento, fiabilidad y seguridad. El ámbito de aplicación de este documento se extiende a los sistemas mecánicos, y electrónicos que conforman el robot. Si se justifica adecuadamente, se podrán adoptar soluciones alternativas a las exigidas en este documento, debido al desarrollo tecnológico o a la imposición de nueva normativa, siempre que sea necesario u optimice su utilidad, y que no impliquen ni una disminución de los requisitos mínimos de calidad especificados en el mismo, ni un empeoramiento en su funcionamiento y/o utilidad.

2. Normas de calidad y seguridad

Normativa de aplicación

- Reglamento Electrotécnico para Baja Tensión (REBT) y sus Instrucciones Complementaria MI.BT, incluidas las hojas de interpretación.
- Ordenanzas de Seguridad e Higiene en el Trabajo (OSHT).
- Ley de Protección del Ambiente Atmosférico (LPAA).
- Ley número 88/67 de 8 de noviembre: *Sistema Internacional de Unidades de Medida* S.I.
- Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 1: Robots. (ISO 10218-1:2011)
- UNE-EN ISO 13482:2014 Robots y dispositivos robóticos. Requisitos de seguridad para robots no industriales. Robots de asistencia personal no médicos.(ISO 13482:2014).
- UNE-EN 62115 EC Juguetes eléctricos. Seguridad
- UNE-EN 71-8:2012 Seguridad de los juguetes. Juegos de actividad para uso doméstico.

3. Materiales

Resistencias

Descripción

Resistencias de valor fijo de película metálica con tolerancia máxima del 1% para evitar la introducción de ruido intrínseco al circuito. Además deben poseer un coeficiente de temperatura bajo (recomendación 50 ppm).

Motor

Descripción

Son motores de unas dimensiones de 6.92 cm x 2.25 cm x 1.88 cm y un peso de 35 g. Utilizan una tecnología de imán permanente y escobillas de carbón. Poseen una caja reductora 1:48 de piñones plástica. El motor se deberá alimentar de 3 a 9 V, con un voltaje de operación nominal de 3 V. Posee una velocidad sin carga de 110 rpm.

Driver

Descripción

Se utilizará un driver constituido por dos puentes en H por canal de salida, con diodos de libre circulación. Además deberá entregar mínimo una corriente de 500 mA por canal. Se recomienda la utilización del L298D de STMicroelectronics o similar.

Microcontrolador

Descripción

Se utilizará el modelo PIC16F88 de Microchip que posee 16 pines configurables con una velocidad de procesado de 32 kHz.

Sensor Ultrasonidos

Descripción

El ultrasonido tendrá como mínimo una Frecuencia de trabajo de 40 Hz con una zona muerta de trabajo de 2 cm como máximo y un rango de 4 m mínimo. Se recomienda utilizar el modelo *HC-SR04 de ElecFreaks*

PCB

Descripción

Placa de cobre con pistas pegadas y perforadas para admitir los terminales de los componentes.

Soporte metacrilato

Descripción

Consiste en una placa de metacrilato de 129x190x5mm dotada con agujeros para los tornillos de 3.5mm de diámetro.

Ruedas

Descripción

Consiste en una rueda de 30 mm de diámetro con dibujo neumático

Rueda libre

Descripción

Las medidas adecuadas para dicha rueda serán de 7 mm de radio. Además deberá poder girar con libertad.

4. Programación

Una parte importante del proceso de fabricación es la de transferir el código fuente que implementa el funcionamiento del robot al procesador. Para ello es necesario disponer de un ordenador con el programa MPLAB X, y el kit de desarrollo que proporciona Microchip (Pickit), que funciona como comunicación entre el microcontrolador y el PC. Este kit se puede encontrar en dos versiones, Pickit 2 y Pickit 3, se puede utilizar cualquiera de los dos durante el proceso. También es necesario tener instalado el programa correspondiente al kit de desarrollo que se utilice. Se abrirá el programa Pickit 2/3, se conecta el programador al ordenador mediante USB. Se abre un nuevo script en el programa, donde se escribe todo el programa, una vez desarrollado y guardado el código se busca el archivo .hex generado por el programa MPLAB X.

5. Montaje

Se procede a soldar los componentes en la PCB del módulo siguelineas con un soldador tipo lápiz y estaño. En ella van a ser soldados los sensores CNY70 y sus correspondientes resistencias.

En cada perforación de la placa, donde se ha introducido la patilla del componente, se utiliza un trozo de estaño para unir el componente con la placa. Para ello se acerca la punta del soldador al estaño, que está tocando con la placa y la patilla del componente, para darle calor, fundirlo y unir la patilla a la placa. Esta operación se repite de la misma manera para cada patilla que se quiera unir con la placa.

Una vez soldados todos los componentes, se retira la placa y se deja enfriar la soldadura a temperatura ambiente 5-10 minutos por precaución. Seguidamente, se realizan los mismos pasos pero esta vez para la otra PCB en la cual van a ir fijados los siguientes componentes:

- 3 Leds
- 3 Resistencias
- Conectores hembra

Por otro lado, se monta la base del robot con las ruedas, los motores DC y la rueda libre en el siguiente orden: primero se atornillan los dos motores a la base para después acoplar una rueda a cada eje del motor y a continuación se atornilla en la parte trasera de la base la rueda libre, utilizada básicamente para hacer de soporte y dar estabilidad.

Una vez terminada la base, en la parte delantera se atornilla la PCB a la cual van unidos todos los componentes, en la parte de abajo del robot se atornillará el puente en H L298D con ayuda de dos tornillos. Conectamos los motores mediante cables al puente en H y este a su vez a la PCB central. Colocamos en la parte trasera del robot la batería externa mediante un trozo de cinta de velcro la fijamos al chasis, a continuación la conectamos a la PCB central y colocamos el sensor TSOP y el HCSR04.

A continuación fijamos mediante tornillos en la parte inferior delantera del robot la PCB del módulo siguelineas, esta debe estar a una distancia aproximada de 1 cm del suelo, y la conectamos a la PCB central. Una vez realizado esto se puede dar por finalizado el proceso de montaje.

3. ANEXO PRESUPUESTO

En el presente apartado se va a detallar el presupuesto de fabricación y desarrollo del Proyecto, es decir, el precio total de fabricar un robot.

Dentro del mismo se divide a su vez, en el cuadro de precios elementales, el cuadro de precios descompuestos, donde se desglosa según sus diferentes módulos y procesos, y un resumen del presupuesto.

En lo referente a los precios, los de los componentes han sido sacados de la página del fabricante, y los de los operarios en base a una estimación. El tiempo de ejecución de los operarios se ha estimado en base al tiempo que se tardó en realizar las diferentes tareas, con lo que el coste de mano de obra podría variar ligeramente.

1. Coste componentes

Componente	Cantidad	Precio Unitario	Importe Total
Arduino Uno	1	26,58€	26,58€
L298D	1	3,82€	3,82€
Batería externa	1	9,99€	9,99€
Estaño	1 bobina	0,60€	0,60€
PIC16F88	1	2,95€	2,95€
Kit montaje robot	1	29,75€	29,75€
Mando a distancia universal	1	16,95€	16,95€
CNY 70	3	0,95€	2,85€
Cableado	30	2,80€	2,80€
TSOP48XX	1	1,07€	1,07€
PCB	2	3€	6€
LED rojo	1	0,15	0,15
LED verde	2	0,15	0,30
Conector hembra aéreo 2 polos (1*10)	1	1,24	1,24
Conectores hembra (1*30)	1	2,49	2,49
Resistencia de 10 kΩ con un 1% de tolerancia	9	0,044€	0,396€
TOTAL			107,936€

Tabla 3: Coste componentes.

2. Coste mano de obra

	Coste por hora
Ingeniero Graduado	35 €/h
Técnico	12 €/h

Tabla 4: Coste mano de obra.

Teniendo en cuenta los precios por hora en función de la categoría profesional, se procede a aplicarse el número de horas aproximadas realizadas por cada uno.

Tarea	Graduado en Ingeniería	Técnico
Diseño	20h	
Fabricación Partes		10 h
Montaje		4 h
Programación Microcontrolador	4h	
Pruebas del Programa	3 h	
Testeo Partes Mecánicas	1 h	
Testeo Software	3 h	
TOTAL	31h	14 h

Tabla 5: Tiempo asociado a la mano de obra

Aplicándose el coste por horas de cada operario se obtiene un valor.

Operario	Horas totales	Precio por hora	Coste Total
Graduado en Ingeniería	31h	35 €	1.085€
Técnico	14h	12 €	168€
Total			1.253€

Tabla 6: Coste total sin IVA

3. Costes adicionales

Se añaden también como costes dentro del presupuesto el uso de los aparatos utilizados tales como soldador, herramientas, ordenadores y el gasto energético utilizado.

Se estimarán ambos costes como un 3 % sobre el precio de la mano de obra, por lo que el cálculo quedará de la siguiente forma.

% Aplicado a los gastos adicionales	3%
Mano de obra	1.253€
Costes adicionales	37,59 €

Tabla 7: Coste adicionales

4. Costes totales

A la suma de todos los costes anteriores se ha de añadir el 21 % del IVA, así queda el precio final del proyecto.

Precio Total	
Coste de los componentes	107,936
Coste mano de obra	1.253€
Coste adicional	37,59€
Total sin IVA	1398,526€
Total con IVA (21 %)	1692,22€

Tabla 8: Coste total.

El coste total del proyecto será de **1692,22 €**

3. ANEXO DATASHEETS

1. DATASHEET HCSR04



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

2. DATASHEET L298



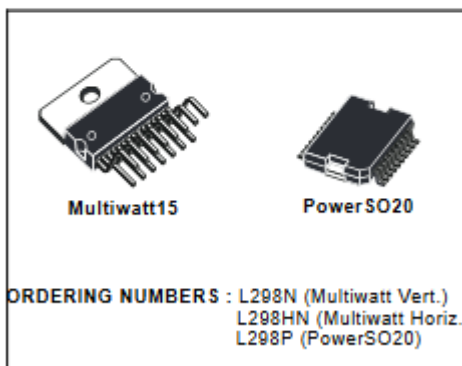
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

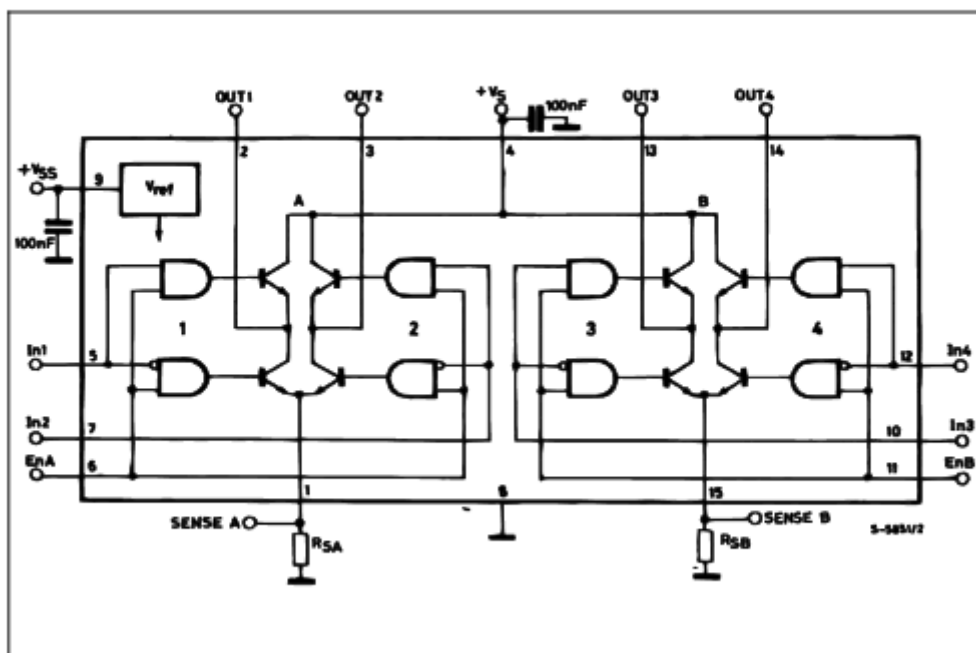
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



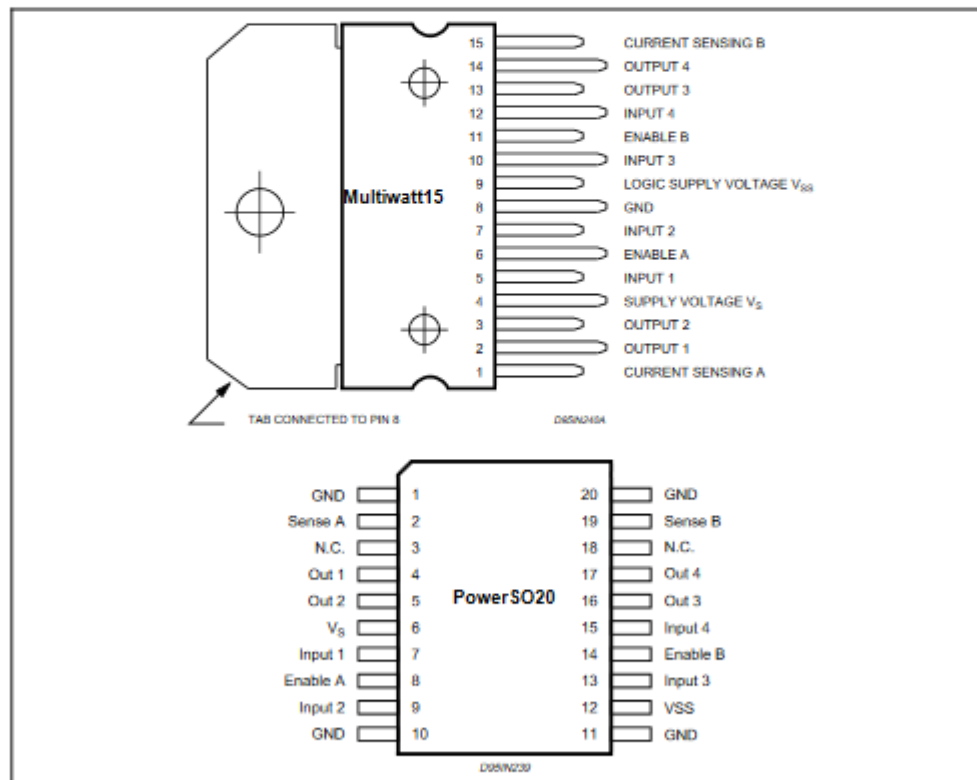
DATASHEET L298

L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-case)}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

DATASHEET L298

L298

PIN FUNCTIONS (refer to the block diagram)

MW.15	Power SO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		48	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0		13	22	mA
		V _I = L		50	70	mA
		V _I = H				mA
		V _{en} = L			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0		24	36	mA
		V _I = L		7	12	mA
		V _I = H				mA
		V _{en} = L			6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _L	Low Voltage Input Current (pins 5, 7, 10, 12)	V _I = L			-10	μA
I _H	High Voltage Input Current (pins 5, 7, 10, 12)	V _I = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat (H)}	Source Saturation Voltage	I _L = 1A	0.95	1.35	1.7	V
		I _L = 2A		2	2.7	V
V _{CEsat (L)}	Sink Saturation Voltage	I _L = 1A (5)	0.85	1.2	1.6	V
		I _L = 2A (5)		1.7	2.3	V
V _{CEsat}	Total Drop	I _L = 1A (5)	1.80		3.2	V
		I _L = 2A (5)			4.9	V
V _{sense}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

DATASHEET L298

L298

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1 (V)$	Source Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (2); (4)		1.5		μs
$T_2 (V)$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		0.2		μs
$T_3 (V)$	Source Current Turn-on Delay	$0.5 V_i$ to $0.1 I_L$ (2); (4)		2		μs
$T_4 (V)$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.7		μs
$T_5 (V)$	Sink Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		0.7		μs
$T_6 (V)$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.25		μs
$T_7 (V)$	Sink Current Turn-on Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		1.6		μs
$T_8 (V)$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.2		μs
$f_c (V)$	Commutation Frequency	$I_L = 2A$		25	40	KHz
$T_1 (V_{en})$	Source Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (2); (4)		3		μs
$T_2 (V_{en})$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		1		μs
$T_3 (V_{en})$	Source Current Turn-on Delay	$0.5 V_{en}$ to $0.1 I_L$ (2); (4)		0.3		μs
$T_4 (V_{en})$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.4		μs
$T_5 (V_{en})$	Sink Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		2.2		μs
$T_6 (V_{en})$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.35		μs
$T_7 (V_{en})$	Sink Current Turn-on Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		0.25		μs
$T_8 (V_{en})$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.1		μs

- 1) Sensing voltage can be $-1 V$ for $t \leq 50 \mu s$; in steady state $V_{sens} \text{ min} \geq -0.5 V$.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

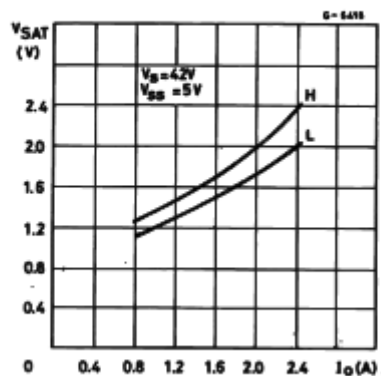
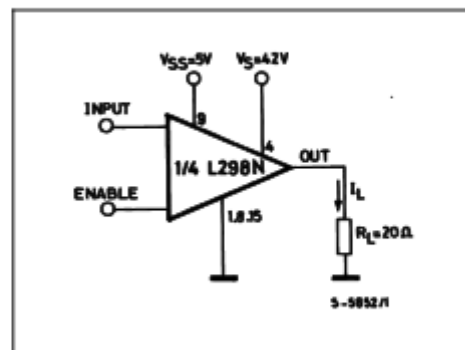


Figure 2 : Switching Times Test Circuits.



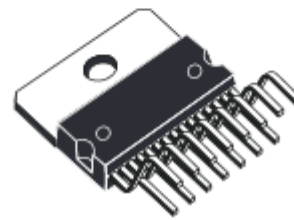
Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

DATASHEET L298

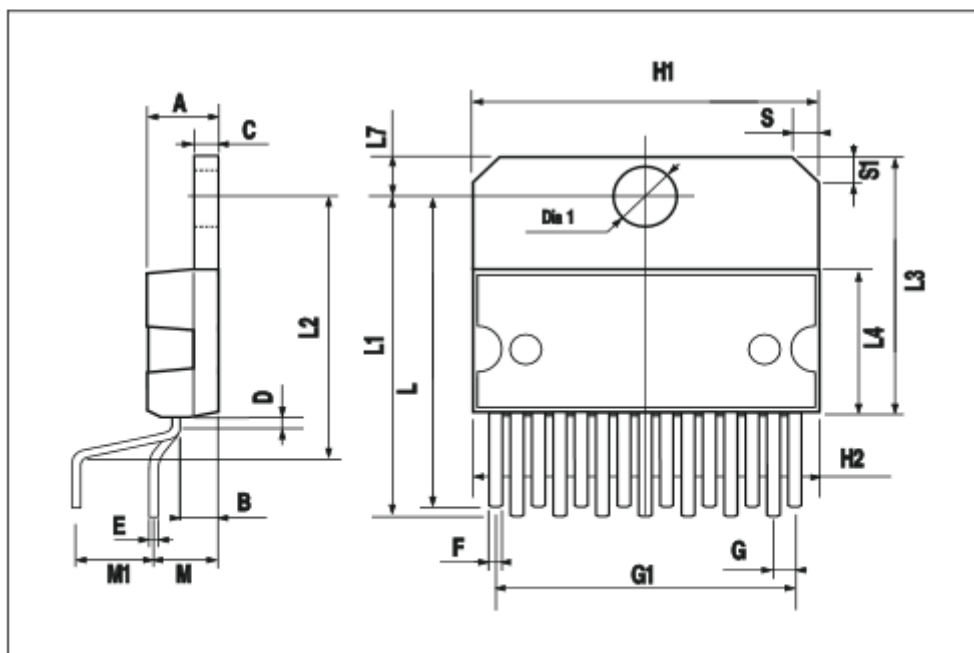
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.888
L1	21.7	22.1	22.5	0.854	0.870	0.888
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 V



3. DATASHEET PIC16F88



PIC16F87/88

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
 - Primary Run: RC oscillator, 76 μ A, 1 MHz, 2V
 - RC_RUN: 7 μ A, 31.25 kHz, 2V
 - SEC_RUN: 9 μ A, 32 kHz, 2V
 - Sleep: 0.1 μ A, 2V
- Timer1 Oscillator: 1.8 μ A, 32 kHz, 2V
- Watchdog Timer: 2.2 μ A, 2V
- Two-Speed Oscillator Start-up

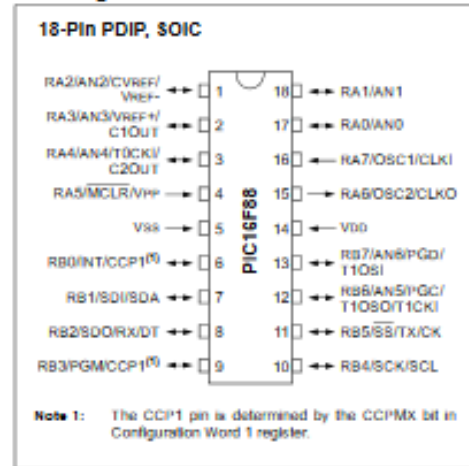
Oscillators:

- Three Crystal modes:
 - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
 - ECIO: up to 20 MHz
- Internal oscillator block:
 - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I²C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
 - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
 - Programmable on-chip voltage reference
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Pin Diagram



Special Microcontroller Features:

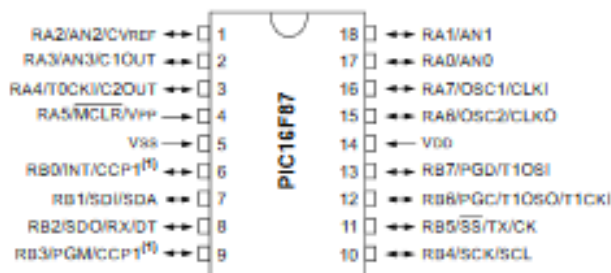
- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
 - Programmable period from 1 ms to 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	8 SP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	1	1	Y	2	Y	2/1

PIC16F87/88

Pin Diagrams

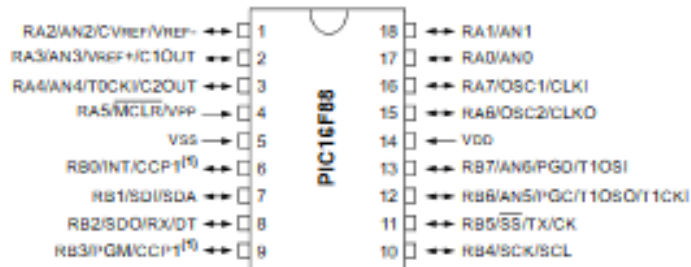
18-Pin PDIP, SOIC



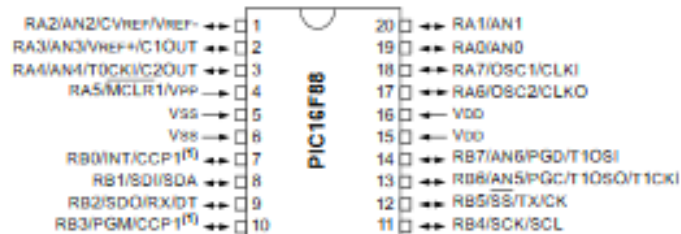
20-Pin SSOP



18-Pin PDIP, SOIC



20-Pin SSOP



Note 1: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

4. DATASHEET RESISENCIAS



Metal Film Fixed Resistors

Type LR Series

Key Features

- Superior quality metal film resistors with 1% tolerance and temperature coefficients down to 50 ppm. 3 case sizes are available in 0.25, 0.6, 0.75W. The LR1L series is a low ohmic value range from 0.1 to 0.82 ohm. Ideally suited where low resistance and small size are required.
- Metal film resistors have excellent stability under load and severe environmental conditions. They exhibit very low noise current and voltage coefficients. They are available in a wide range of resistance values and are suitable for general purpose and precision applications.



The resistive element comprises a thin film of nickel-chrome alloy evaporated onto a high thermal conductivity ceramic element. Metal end caps are force fitted to the element prior to spiralling to value. Tinned copper lead wires are welded to the end caps and the components are then coated. One coat of phenolic resin is followed by three coats of epoxy resin. All resistors are tested for value and tolerance.

Characteristics - Electrical

	LR0204		LR1L	LR1			LR2	LR100	LR200
Rated Power @ 70°C (W)	0.25		0.5	0.6			0.75	1	2
Resistance Range (Ohms) Min	1R0	10R	R10	1R0	10R	1M1	1R0	10R	10R
	Max		9R11	1M0	R82	9R11	1M0	10M	1M0
Tolerance (%)	1	1	5	0.5 1 2	0.5 1 2	0.5 1 5	1 2	1 5	1 5
Code Letter	F	F	J	D F G	D F G	D F J	F G	F J	F J
Temp. Coefficient (ppm/°C)	± 100	± 100	± 200	± 100	± 50	± 100	± 100	2550/100	2550/100
Selection Series	E24	E24	E12	E24	E24	E24	E24	E24	E24
On Request		E96			E96		E96	E96	E96
Limiting Element Voltage	200		350	350			350	500	500
Max Permitted Element Voltage	200		350	350			350	500	500
Max Overload Voltage	400		500	700			700	1000	1000
Max Intermittent Overload Voltage	500		750	750			750	1000	1000
Operating Temp. Range (°C)	-55 to +155								
Climatic Category	55/155/56								
Dielectric Strength (V)	500	700		700			700	700	700
Insulation Resistance Min Dry (Mohms)	1000								

5. DATASHEET CNY70

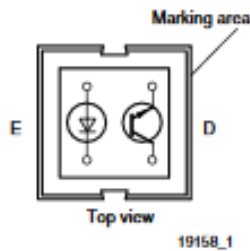


www.vishay.com

CNY70

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output



FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 7 x 7 x 6
- Peak operating distance: < 0.5 mm
- Operating range within > 20 % relative collector current: 0 mm to 5 mm
- Typical output current under test: $I_C = 1 \text{ mA}$
- Emitter wavelength: 950 nm
- Daylight blocking filter
- Lead (Pb)-free soldering released
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912



RoHS COMPLIANT

DESCRIPTION

The CNY70 is a reflective sensor that includes an infrared emitter and phototransistor in a leaded package which blocks visible light.

APPLICATIONS

- Optoelectronic scanning and switching devices i.e., index sensing, coded disk scanning etc. (optoelectronic encoder assemblies).

PRODUCT SUMMARY				
PART NUMBER	DISTANCE FOR MAXIMUM CTR_{rel} (1) (mm)	DISTANCE RANGE FOR RELATIVE $I_{out} > 20\%$ (mm)	TYPICAL OUTPUT CURRENT UNDER TEST (2) (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
CNY70	0	0 to 5	1	Yes

Notes

(1) CTR: current transference ratio, I_{out}/I_{in}

(2) Conditions like in table basic characteristics/sensors

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	VOLUME (1)	REMARKS
CNY70	Tube	MOQ: 4000 pcs, 80 pcs/tube	-

Note

(1) MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25 \text{ }^\circ\text{C}$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
COUPLER				
Total power dissipation	$T_{amb} \leq 25 \text{ }^\circ\text{C}$	P_{tot}	200	mW
Ambient temperature range		T_{amb}	- 40 to + 85	$^\circ\text{C}$
Storage temperature range		T_{stg}	- 40 to + 100	$^\circ\text{C}$
Soldering temperature	Distance to case 2 mm, $t \leq 5 \text{ s}$	T_{sd}	260	$^\circ\text{C}$
INPUT (EMITTER)				
Reverse voltage		V_R	5	V
Forward current		I_F	50	mA
Forward surge current	$t_p \leq 10 \text{ } \mu\text{s}$	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25 \text{ }^\circ\text{C}$	P_V	100	mW
Junction temperature		T_J	100	$^\circ\text{C}$

DATASHEET CNY70



www.vishay.com

CNY70

Vishay Semiconductors

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
OUTPUT (DETECTOR)				
Collector emitter voltage		V_{CEO}	32	V
Emitter collector voltage		V_{ECO}	7	V
Collector current		I_C	50	mA
Power dissipation	$T_{amb} \leq 25\text{ }^{\circ}\text{C}$	P_V	100	mW
Junction temperature		T_J	100	$^{\circ}\text{C}$

ABSOLUTE MAXIMUM RATINGS

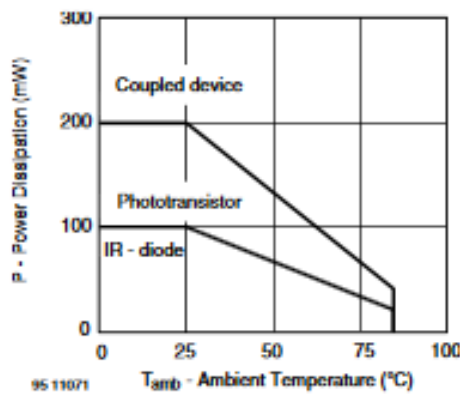


Fig. 1 - Power Dissipation vs. Ambient Temperature

BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
COUPLER						
Collector current	$V_{CE} = 5\text{ V}$, $I_F = 20\text{ mA}$, $d = 0.3\text{ mm}$ (figure 1)	I_C (1)	0.3	1.0		mA
Cross talk current	$V_{CE} = 5\text{ V}$, $I_F = 20\text{ mA}$, (figure 2)	I_{CX} (2)			600	nA
Collector emitter saturation voltage	$I_F = 20\text{ mA}$, $I_C = 0.1\text{ mA}$, $d = 0.3\text{ mm}$ (figure 1)	V_{CEsat} (2)			0.3	V
INPUT (EMITTER)						
Forward voltage	$I_F = 50\text{ mA}$	V_F		1.25	1.6	V
Radiant intensity	$I_F = 50\text{ mA}$, $t_p = 20\text{ ms}$	I_e			7.5	mW/sr
Peak wavelength	$I_F = 100\text{ mA}$	λ_p	940			nm
Virtual source diameter	Method: 63 % encircled energy	d		1.2		mm
OUTPUT (DETECTOR)						
Collector emitter voltage	$I_C = 1\text{ mA}$	V_{CEO}	32			V
Emitter collector voltage	$I_E = 100\text{ }\mu\text{A}$	V_{ECO}	5			V
Collector dark current	$V_{CE} = 20\text{ V}$, $I_F = 0\text{ A}$, $E = 0\text{ lx}$	I_{CFO}			200	nA

Notes

- (1) Measured with the "Kodak neutral test card", white side with 90 % diffuse reflectance
 (2) Measured without reflecting medium

6. DATASHEET TSOP44XX

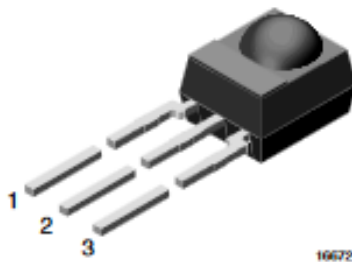


www.vishay.com

TSOP22..., TSOP24..., TSOP48..., TSOP44..

Vishay Semiconductors

IR Receiver Modules for Remote Control Systems



MECHANICAL DATA

Pinning for TSOP44..., TSOP48...:

1 = OUT, 2 = GND, 3 = V_S

Pinning for TSOP22..., TSOP24...:

1 = OUT, 2 = V_S , 3 = GND

FEATURES

- Improved immunity against HF and RF noise
- Low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against optical noise
- Insensitive to supply voltage ripple and noise
- Material categorization:
for definitions of compliance please see www.vishay.com/doc?999912



RoHS
COMPLIANT
HALOGEN
FREE
GREEN
(2-2008)

DESCRIPTION

The TSOP22..., TSOP48..., TSOP24.. and TSOP44.. series are miniaturized IR receiver modules for infrared remote control systems. A PIN diode and a preamplifier are assembled on lead frame, the epoxy package contains an IR filter.

The demodulated output signal can be directly connected to a microprocessor for decoding.

The TSOP24..., TSOP44.. series devices are optimized to suppress almost all spurious pulses from Wi-Fi and CFL sources. They may suppress some data signals if continuously transmitted.

The TSOP22..., TSOP48.. series devices are provided primarily for compatibility with old AGC2 designs. New designs should prefer the TSOP24..., TSOP44.. series containing the newer AGC4.

These components have not been qualified according to automotive specifications.

PARTS TABLE					
AGC	LEGACY, FOR LONG BURST REMOTE CONTROLS (AGC2)			RECOMMENDED FOR LONG BURST CODES (AGC4)	
Carrier frequency	30 kHz	TSOP4830	TSOP2230	TSOP4430	TSOP2430
	33 kHz	TSOP4833	TSOP2233	TSOP4433	TSOP2433
	36 kHz	TSOP4836	TSOP2236	TSOP4436 (1)(2)(3)	TSOP2436 (1)(2)(3)
	38 kHz	TSOP4838	TSOP2238	TSOP4438 (4)(5)(6)	TSOP2438 (4)(5)(6)
	40 kHz	TSOP4840	TSOP2240	TSOP4440	TSOP2440
	56 kHz	TSOP4856	TSOP2256	TSOP4456 (7)(8)	TSOP2456 (7)(8)
Package	Mold				
Pinning	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S , 3 = GND	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S , 3 = GND	
Dimensions (mm)	6.0 W x 6.95 H x 5.6 D				
Mounting	Leaded				
Application	Remote control				
Best remote control code	(1) RC-5 (2) RC-6 (3) Panasonic (4) NEC (5) Sharp (6) r-step (7) Thomson RCA				

DATASHEET TSOP44X

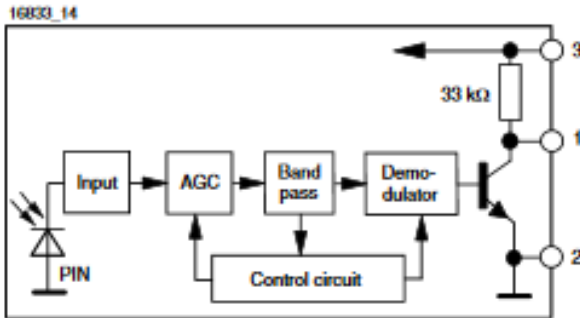


www.vishay.com

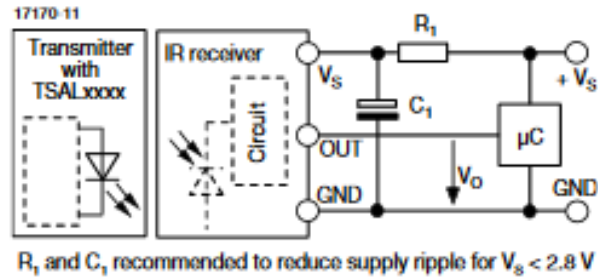
TSOP22..., TSOP24..., TSOP48..., TSOP44..

Vishay Semiconductors

BLOCK DIAGRAM



APPLICATION CIRCUIT



ABSOLUTE MAXIMUM RATINGS

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Supply voltage		V_S	-0.3 to +6	V
Supply current		I_S	5	mA
Output voltage		V_O	-0.3 to 5.5	V
Voltage at output to supply		$V_S - V_O$	-0.3 to $(V_S + 0.3)$	V
Output current		I_O	5	mA
Junction temperature		T_J	100	°C
Storage temperature range		T_{stg}	-25 to +85	°C
Operating temperature range		T_{amb}	-25 to +85	°C
Power consumption	$T_{amb} \leq 85$ °C	P_{tot}	10	mW
Soldering temperature	$t < 10$ s, 1 mm from case	T_{sd}	260	°C

Note

- Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect the device reliability.

ELECTRICAL AND OPTICAL CHARACTERISTICS ($T_{amb} = 25$ °C, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply current	$E_v = 0, V_S = 5$ V	I_{SD}	0.55	0.7	0.9	mA
	$E_v = 40$ klx, sunlight	I_{SH}	-	0.8	-	mA
Supply voltage		V_S	2.5	-	5.5	V
Transmission distance	$E_v = 0$, test signal see Fig. 1, IR diode TSAL6200, $I_F = 200$ mA	d	-	45	-	m
Output voltage low	$I_{Osl} = 0.5$ mA, $E_e = 0.7$ mW/m ² , test signal see Fig. 1	V_{Osl}	-	-	100	mV
Minimum irradiance	Pulse width tolerance: $t_{pl} - 5/f_o < t_{po} < t_{pl} + 6/f_o$, test signal see Fig. 1	$E_e \text{ min.}$	-	0.12	0.25	mW/m ²
Maximum irradiance	$t_{pl} - 5/f_o < t_{po} < t_{pl} + 6/f_o$, test signal see Fig. 1	$E_e \text{ max.}$	50	-	-	W/m ²
Directivity	Angle of half transmission distance	$\Phi_{1/2}$	-	± 45	-	deg