

# ROBOT SEGUIDOR DE LINEA CON ORUGAS

ORGANIZACIÓN DE PROYECTOS FINAL DE CARRERA



# TABLA DE CONTENIDO

TABLA DE CONTENIDO .....	2
MEMORIA .....	3
OBJETIVOS Y ALCANCES .....	4
1.1 ASPECTOS METODOLÓGICOS .....	5
1.2 TAREAS ANTECEDENTES .....	5
1.3 NORMAS Y REFERENCIAS .....	6
1.4 PROGRAMAS INFORMÁTICOS EMPLEADOS .....	6
1.5 REQUISITOS DEL DISEÑO .....	7
1.6 SOLUCIONES ADOPTADAS .....	9
1.6.1 PLATAFORMA DE DESARROLLO ARDUINO .....	9
1.6.2 SOFTWARE DE DISEÑO. DIPTRACE .....	18
1.6.3 PLACA DE CONEXIONES .....	23
1.6.4 CHASIS Y TRANSMISIÓN .....	26
1.6.5 DRIVER MOTORES .....	28
1.6.6 SENSOR CNY70 .....	32
1.6.7 PLACA SENSORES DE LÍNEA .....	36
1.6.8 ENCODERS .....	39
1.6.9 PLACA DE COMUNICACIÓN BLUETOOTH .....	42
1.6.10 CARGADOR DE BATERÍAS .....	44
ANEXOS .....	48
ARDUINO UNO REV3 SCHEMATIC .....	49
PLANOS .....	50
PLANO DEL COCHE SIGUELÍNEAS .....	51
PLANO MOTORES Y PIEZAS DE SUJECCIÓN .....	52
PLANO PCB CONEXIONES .....	53
PLANO PCB DRIVER DE MOTORES .....	54
PLANO PCB SENSORES DE LÍNEA .....	55
PLANO PCB ENCODERS .....	56
PLANO PCB BLUETOOTH BREAKOUT .....	57
PLANO PCB BLUETOOTH EXPANSIÓN .....	58
ESTADO DE MEDICIONES .....	59
ESTADO DE MEDICIONES .....	60
PRESUPUESTO .....	61
CUADRO DE PRECIOS UNITARIOS DE MATERIALES .....	62
CUADRO DE PRECIOS UNITARIOS DE MANO DE OBRA .....	63
CUADRO DE PRECIOS UNITARIOS TOTALES .....	64
CUADRO DE PRECIOS UNITARIOS DESCOMPUESTOS .....	65
PRESUPUESTO .....	68
RESUMEN DEL PRESUPUESTO .....	69

**MEMORIA**

## OBJETIVOS Y ALCANCES

El objetivo de este proyecto es la realización de un robot partiendo desde cero, diseñando el chasis de metacrilato, la colocación de los motores y placas mediante tornillería.

El coche oruga será capaz de detectar una línea y seguirla, o la programación de un trayecto preestablecido para que sea capaz de realizarlo posteriormente.

La comunicación bluetooth cada vez está más extendida, por lo que se decidió que el robot podrá comunicarse mediante bluetooth con otros dispositivos, logrando seleccionar el modo de funcionamiento a distancia: sigue líneas o trayectoria.

En el modo sigue líneas, los 6 sensores de infrarrojos situados en la parte delantera del robo, lo más cerca posible del suelo, detectarán la línea. Del mismo modo se emplean estos sensores en las ruedas como encoders relativos para conocer en todo momento la posición del robot.

En todo momento estará controlado mediante la plataforma de desarrollo Arduino, siendo la base del funcionamiento del mismo.

El sistema de tracción delantera está fijado a las ruedas locas traseras mediante una oruga.

También dispondrá de un cargador de baterías para no depender de la disponibilidad de una fuente de alimentación externa y no ver limitado su recorrido por el cable de alimentación, aun así también podrá utilizarse alimentándose de dicha fuente.

El diseño del robot señalado cumple los requisitos y especificaciones definidos en la asignatura de Organización de Proyecto Final de Carrera.

## 1.1 ASPECTOS METODOLÓGICOS

El equipo de trabajo de la asignatura de Organización de Proyecto de Fin de Carrera está formado por:

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

Para la realización del proyecto del robot sigue líneas nos hemos basado en el estudio de otros robots construidos anteriormente en esta misma asignatura por alumnos de la Universidad de La Rioja. Algunos de los componentes empleados en nuestro proyecto (como los motores) han sido reciclados de los robots de años anteriores.

El profesor [REDACTED] ha sido el principal colaborador en la asignatura de Organización de Proyectos de Fin de Carrera.

## 1.2 TAREAS ANTECEDENTES

La tarea previa a la realización del proyecto en sí constó de varios apartados:

- Elección de la familia de microcontrolador a utilizar, comparando el PIC 18F2550 con la arquitectura Arduino UNO que será la que utilizaremos.
- El estudio previo del sistema de traslación del robot.
- La idea inicial era un robot que empleara patas articuladas para desplazarse, pero se llegó a la conclusión de que el proyecto se encarecería por la necesidad de usar como mínimo el doble de motores.
- A continuación se debatió la idea de emplear un sistema de dos ruedas tractoras con una tercera rueda “loca” que serviría de apoyo del chasis, pero la idea fue desechada ya que ya se había realizado en años anteriores.

- Finalmente surgió la idea de usar orugas para transmitir el par de las ruedas tractoras delanteras a las traseras, siendo esta nuestra elección.
- El diseño del chasis, posicionamiento y sujeción tanto de los motores como de los ejes traseros así como de la transmisión de la potencia de los motores.
- El diseño de la circuitería que más adelante haría funcionar los distintos elementos del coche. Se tuvieron que diseñar los circuitos de la placa controladora de los motores, la placa de comunicaciones bluetooth, los encoders relativos para el control de posición, la placa con los sensores infrarrojos de detección de línea y la placa base que conecta el Arduino con todas las demás placas anteriormente mencionadas.

### 1.3 NORMAS Y REFERENCIAS

Para la realización de este proyecto con fecha de entrega y presentación el viernes, 18 de mayo de 2012, se ha tenido en cuenta la siguiente norma:

**UNE 157001:2002** que trata de los criterios generales de la realización de proyectos.

### 1.4 PROGRAMAS INFORMÁTICOS EMPLEADOS

A lo largo de la realización del trabajo ha sido necesaria la intervención de varios programas informáticos. Entre ellos están:

- **AutoCad 2009** para la realización de los planos del proyecto.
- **Dip Trace** para el diseño de los circuitos y el PCB.
- **Orcad** para la correcta impresión de los PCB.
- **Arduino** para la programación del mismo Arduino.

## 1.5 REQUISITOS DEL DISEÑO

Recorrido del vehículo siguiendo una línea:

- Anchura línea de 3cm aprox.( $\pm 2\text{mm}$ )
- Curvas de radio medio no inferior a  $20\text{cm} \pm 1\text{cm}$
- Posibilidad circulo de dicho radio ( $20\text{cm} \pm 1\text{cm}$ ).
- Recorrido formando ángulos rectos  $\pm 15^\circ$
- Cruce de dos trayectorias perpendiculares  $\pm 15^\circ$
- Cruces en forma de T prioridad al giro a la derecha.
- Pendientes de subida y bajada de  $30^\circ \pm 5\%$

Movimiento del vehículo siguiendo una trayectoria fijada:

- Giros a  $90^\circ$
- Realización de trayectorias ortogonales.
- Determinación de posición (ayuda sensores).

Movimiento del vehículo de forma manual:

- Comunicaciones para seguimiento de trayectoria.
- Control de movimientos en tiempo real.

Carga:

- Capacidad de transportar peso no superior a  $300\text{gr} \pm 5\%$ .
- Volumen cámara video o fotográfica con captura de video.( $10 \times 15 \times 10\text{cm}$ )

Control:

- Detección obstáculo en circuito. Realizar función de paro.
- Iniciar la marcha con indicación código radiofrecuencia, (a determinar por la propiedad).
- Parar la marcha con indicación código radiofrecuencia, (a determinar por la propiedad).



## 1.6 SOLUCIONES ADOPTADAS

### 1.6.1 PLATAFORMA DE DESARROLLO ARDUINO

#### *Descripción:*

Para la realización de este proyecto, se ha estipulado desde un inicio que el sistema de control base se había de realizar con la plataforma Arduino.



Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea, controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP).

Las placas pueden ser hechas a mano o compradas, montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues eres libre de adaptarlos a tus necesidades.

Esto nos permite una aproximación a este tipo de microcontrolador, tan popular tanto entre los aficionados como en la industria, que cada día se va sofisticando más y más (comunicación por USB, Ethernet, etc...).

En nuestro caso concreto, se ha utilizado la versión Arduino UNO R3:



Arduino Uno R3 Front

Arduino Uno R3 Back

La principal dificultad en el manejo de este dispositivo es que no se ha tomado contacto con el mismo previamente en la titulación, ni con este ni con ningún otro de la misma familia.

### *Características:*

El Arduino UNO es una placa electrónica basada en el ATmega328.



Cuenta con 14 entradas/salidas digitales (de los cuales, 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, una de 16 MHz del oscilador de cristal, una conexión USB, un conector de alimentación, una cabecera de ICSP, y un botón de reset. Contiene todo lo necesario para que el micro funcione, solamente se tiene que conectar a un PC mediante USB, o alimentarlo mediante un adaptador AC-DC, o una batería.

El Uno difiere de todas las placas anteriores en que no utilizan el chip controlador FTDI USB-serie. Por el contrario, cuenta con la Atmega16U2

(Atmega8U2 hasta la versión R2) programado como un convertidor de USB a serie.

La revisión 3 de la placa, tiene las siguientes nuevas características:

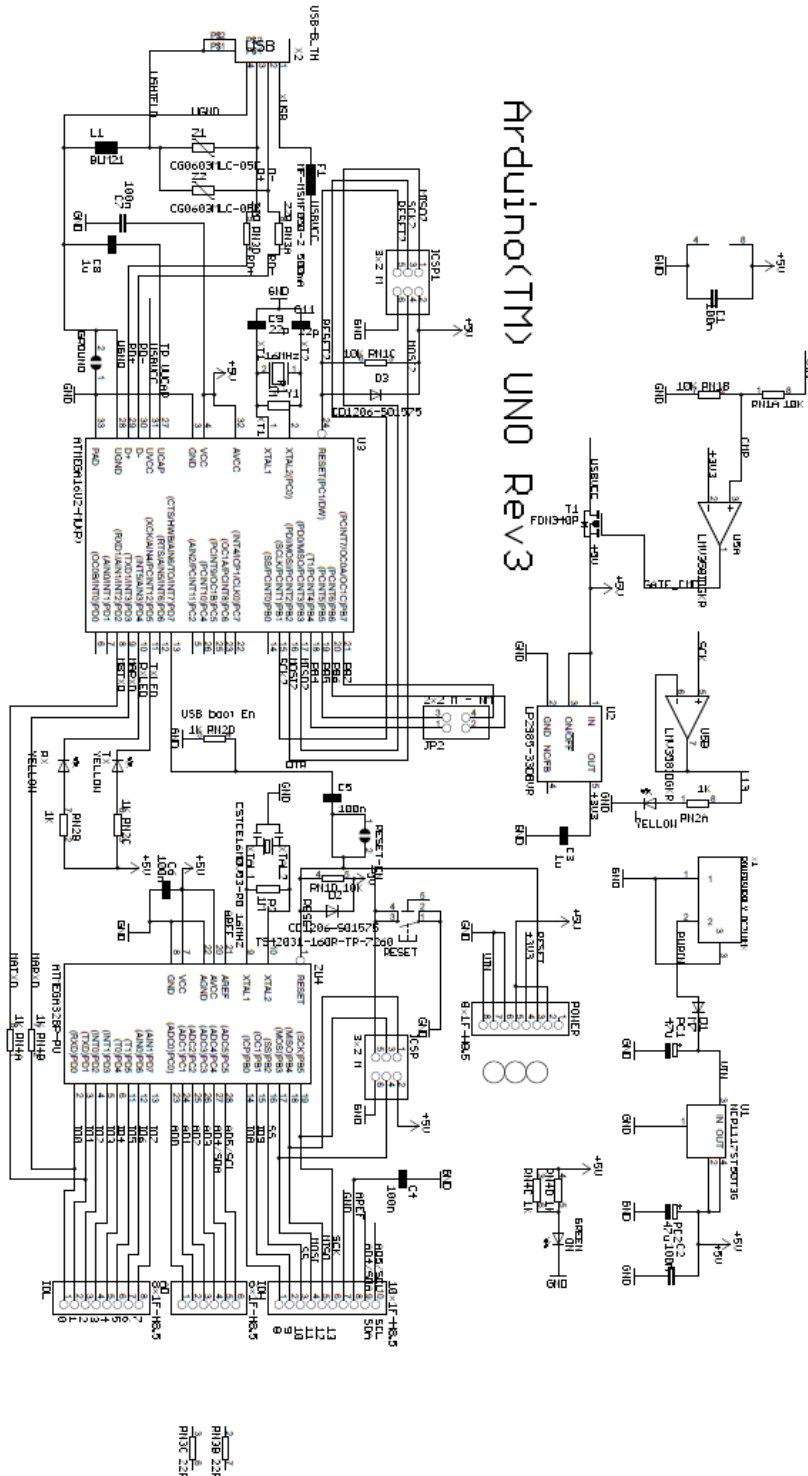
- 1,0 pinout: añadidos pines SDA y SCL que se encuentran cerca del pin AREF y otros dos pin nuevos que están cerca del pin de RESET, el IOREF que permite a las shields adaptarse a la tensión proporcionada por la placa. En el futuro, las shields serán compatibles tanto con la tarjeta que utiliza el AVR, que opera con 5V y con el Arduino Due, que funciona con 3.3V. El segundo es un pin no está conectado, que se reserva para usos futuros.
- Circuito de RESET más fuerte.
- El Atmega16U2 reemplaza al 8U2.

### *Resúmen:*

<b>Microcontrolador</b>	ATmega328
<b>Voltaje de funcionamiento</b>	5V
<b>Voltaje de Entrada (recomendado)</b>	7-12V
<b>Voltaje de entrada (límites)</b>	6-20V
<b>Pines digitales I/O</b>	14 (6 de las cuales proporcionan una salida PWM)
<b>Pines de entrada analógica</b>	6
<b>Corriente continua de los pines I/O</b>	40 mA
<b>Corriente continua del pin de 3,3 V</b>	50 mA
<b>Memoria Flash</b>	32 KB ( ATmega328 ) de los cuales 0,5 KB utilizado por gestor de arranque
<b>SRAM</b>	2 KB ( ATmega328 )
<b>EEPROM</b>	1 KB ( ATmega328 )
<b>Velocidad de reloj</b>	16 MHz

### *Diseño esquemático de referencia:*

En la página oficial de Arduino se ofrece un diseño esquemático de la placa que se usa para alojar el microcontrolador:



[http://arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf)

### **Alimentación:**

El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La alimentación externa (no USB), puede venir de un adaptador AC-DC o batería. El adaptador se puede conectar mediante un conector jack de 2.1mm de centro positivo. Los cables que vengan de la batería se pueden insertar en los pines de GND y Vin del conector de alimentación.

La placa puede operar bajo un suministro externo de 6 a 20 voltios. Si se suministran menos de 7V, sin embargo, el pin de 5V puede ser que no de los 5V y la placa sea inestable. Si se utiliza más de 12V, el regulador de voltaje puede llegar a sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN:** El voltaje de entrada de Arduino cuando está usando una fuente externa (a diferencia de 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o si el suministro de tensión es mediante la conexión jack, se puede acceder a ella mediante este pin.
- **5V:** Este pin genera 5V regulados por el regulador de la placa. La placa puede ser alimentada mediante la entrada de alimentación (7 – 12V), el conector USB (5V) o el pin VIN (7 -12V). El suministro de tensión a través de los pines 5V o 3.3V no pasa por el regulador, y puede dañar la placa. No es aconsejable.
- **3.3V:** Suministro de 3.3V generada por el regulador. El consumo de corriente máxima es de 50mA.
- **GND:** Pines de tierra.

### **Memoria:**

El ATmega328 tiene 32MB (con 0,5KB utilizados para el gestor de arranque). También dispone de 2 KB de SRAM y 1 KB de memoria EEPROM (que se puede leer y escribir con la librería EEPROM ).

### ***Entrada y salida:***

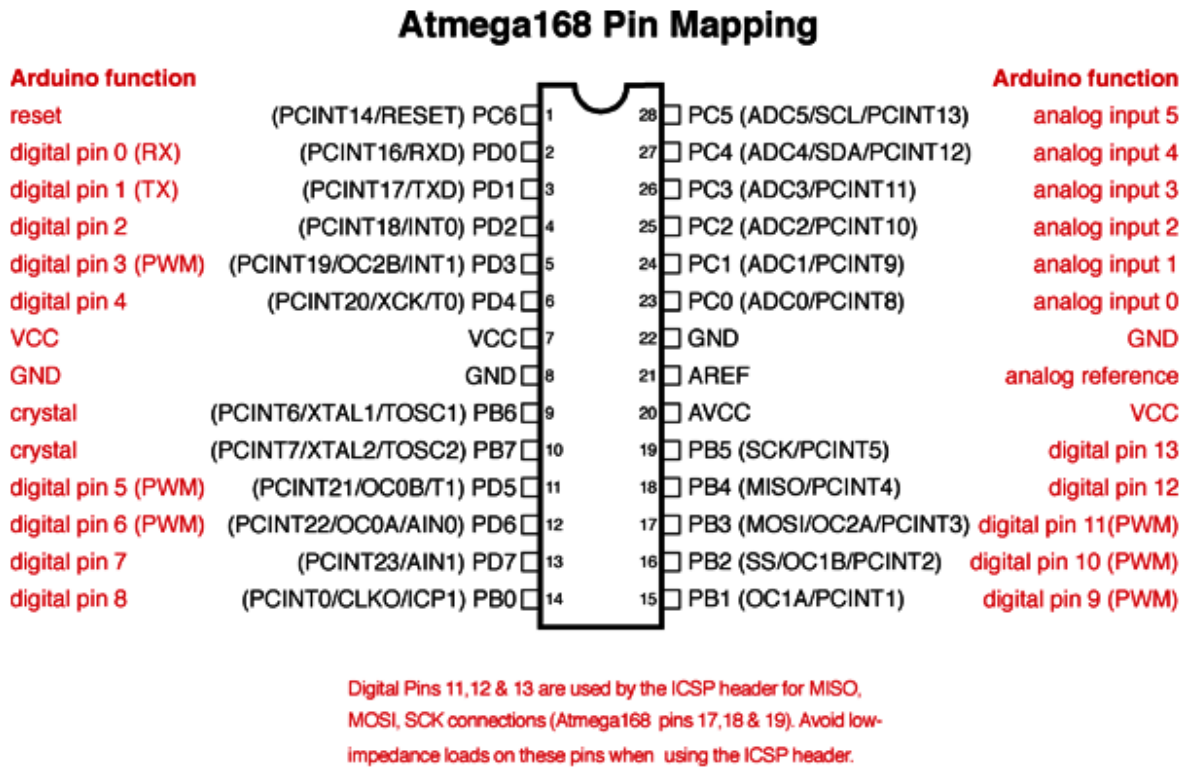
Cada uno de los 14 pines digitales en el Uno se puede utilizar como entrada o salida, usando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Operan a 5V. Cada pin puede proporcionar o recibir un máximo de 40mA y tiene una resistencia pull-up interna (desconectada por defecto) de 20-50 kOhmios. Además, tiene algunos pines con funciones especiales:

- **Serie, o 0(RX) y 1(TX):** Se utiliza para recibir (RX) y transmitir (TX) datos serie TTL. Estos se encuentran conectados a los pines correspondientes del chip USB-TTL ATmega8U2.
- **Interrupciones externas: 2 y 3.** Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver la función `attachInterrupt()` para más detalles.
- **PWM: 3, 5, 6, 9, 10 y 11.** Proporcionan 8 bits de salida PWM con la función `analogWrite()`.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Estos pines son el soporte de la comunicación SPI, usando la librería SPI.
- **LED: 13.** Hay un LED conectado al pin 13. Cuando el pin está en valor HIGH, el LED está encendido, y cuando está en LOW, el LED está apagado.

El Uno tiene 6 entradas analógicas, etiquetadas como A0 a A5, cada una de las cuales proporciona 10 bits de resolución (1024 valores diferentes posibles). Por defecto miden desde masa a 5 voltios, aunque es posible cambiar el extremo superior de su rango con el pin AREF y la función `analogReference()`.

- **TWI: pines A4 o SDA y A5 o SCL.** Comunicación TWI, usando la librería Wire library.
- Hay otro par de pines en la placa:
- **AREF:** Voltaje de referencia para las entradas analógicas. Se usa con la función `analogReference()`.
- **RESET:** Para resetear el microcontrolador se pone esta línea a LOW. Se suele usar para añadir un botón de reset a las shield que se adjuntan a la placa.

El mapa de pines del ATmega8, 168 y 368 es idéntico:



### Comunicación:

El Arduino Uno se puede conectar fácilmente con un ordenador, otro Arduino, u otros microcontroladores. El ATmega328 ofrece una UART TTL (5V) de comunicación serie, que esta disponible en los pines 0(RX) y 1(TX). Un ATmega16U2 en la placa canaliza esta comunicación serie a través de un puerto USB que será reconocido por el software del ordenador como un puerto COM virtual. El firmware '16U2 utiliza el estándar de los controladores USB COM, y no necesita de controladores externos. Sin embargo, en Windows se requiere un archivo .inf. El software de Arduino incluye un monitor serie que permite enviar y recibir datos de texto simple en la placa Arduino. Los LED de RX y TX en la placa parpadean cuando se están transmitiendo datos a través del chip USB-Serie y la conexión USB-ordenador (pero no para la conexión serie con los pines 0 y 1).

Una librería SoftwareSerial permite la comunicación serie en cualquiera de los pines de la placa Uno.

El ATmega 328 también es compatible con la comunicación I2C (TWI) y SPI. El software de Arduino incluye una librería Wire para simplificar el uso del bus I2C. Para la comunicación Spi, hay que utilizar la librería SPI .



### ***Programación:***

El Arduino Uno puede ser programado con el software de Arduino. Hay que seleccionar “Arduino Uno” en *Herramientas>Board menú* (de acuerdo con el microcontrolador en la placa).

El ATmega 328 en el Arduino Uno, viene pregrabado con un gestor de arranque que permite cargar nuevo código en el sin usar un hardware programador externo. Se comunica usando el protocolo STK500 original.

También se puede pasar por alto el gestor de arranque y programar el microcontrolador usando la cabecera ICSP (In-Circuit Serial Programming).

El código fuente del firmware del ATmega16U2 (o 8U2 en las placas de la revisión 1 y revisión 2) está disponible. El ATmega16U2 /8U2 está cargado con un gestor de carga DFU, que puede ser activado por:

- En las placas de la rev.1: conectando el puente de soldadura en la parte de atrás de la placa (cerca del mapa de Italia) y después reseteando el 8U2.
- En las placas de la rev.2 o posteriores: hay una resistencia con la que tirando las líneas HWB del 8U2/16U2 a tierra, hace más fácil ponerlo en modo DFU.

Se puede entonces usar el Atmel's FLIP software (Windows) o el DFU programmer (Mac y Linux) para cargar un nuevo firmware. O se puede usar la cabecera ISP con un programador externo (sobreescribiendo el gestor de carga DFU).

### ***Reset automático (software):***

En lugar de requerir una presión física del botón de reset antes de la carga, el Arduino Uno está diseñado de manera que permite ser reseteado por el software que se ejecuta desde un ordenador conectado. Una de las líneas de control de flujo hardware (DTR) del ATmega8U2/16U2 está conectada a la línea de reset del ATmega 328 a través de condensador de 100nF. Cuando esta línea es activada (nivel bajo), la línea de reset cae lo suficiente como para hacer el reset del chip. El software de Arduino utiliza esta capacidad que le permite cargar el código con solo pulsar el botón de “Upload” en el entorno de programación. Esto significa que el gestor de carga puede tener un tiempo de espera más corto, en la medida en que la bajada de DTR pueda ser coordinada con el comienzo de la carga.

Esta configuración tiene otras implicaciones. Cuando el Uno se conecta a un ordenador con sistema operativo Mac o Linux, se resetea cada vez que se realiza una conexión a través del USB desde software. Para el siguiente medio segundo más o



menos, el gestor de arranque se está ejecutando en el Uno. Mientras está programado para ignorar datos incorrectos (es decir, nada excepto una carga de código nuevo), se interceptarán los primeros bytes de datos enviados a la placa después de que la conexión se abre. Si un esquema que está ejecutando en la placa recibe una configuración en tiempo u otros datos cuando comienza por primera vez, hay que asegurarse de que el software con el que se comunica espera un segundo después de abrir la conexión y antes de enviar esos datos.

El Uno contiene una ruta que puede ser cortada para deshabilitar el autoreset. Los pads a cada lado de la ruta se pueden volver a soldar para volverla a habilitar. Esta etiquetada como “RESET-ES”. También se puede desactivar el autoreset conectando una resistencia de 110 Ohmios desde 5V a la línea de reset.

#### ***Protección USB contra sobrecorriente:***

El Arduino Uno tiene un polifusible reajutable que protege los puertos USB del ordenador de cortocircuitos y sobrecorrientes. Aunque la mayoría de los ordenadores ofrecen su propia protección interna, el polifusible añade una capa adicional de protección. Si hay más de 500mA aplicándose al puerto USB, el fusible automáticamente corta la conexión hasta que el cortocircuito o la sobrecorriente se eliminan.

#### ***Características físicas:***

La longitud y anchura máximas de la PCB de Uno son 2,7 y 2,1 pulgadas respectivamente; con el conector USB y conector de alimentación se extiende más allá de dichas dimensiones. Hay cuatro orificios para tornillos que permiten sujetar la placa a una superficie o caja. Nótese que la distancia entre los pines digitales 7 y 8 es de 160 milésimas de pulgada (0,16”), no un múltiplo de la separación de 100 milésimas de pulgada que tienen los otros pines.

### 1.6.2 SOFTWARE DE DISEÑO. DIPTRACE

#### *Descripción:*

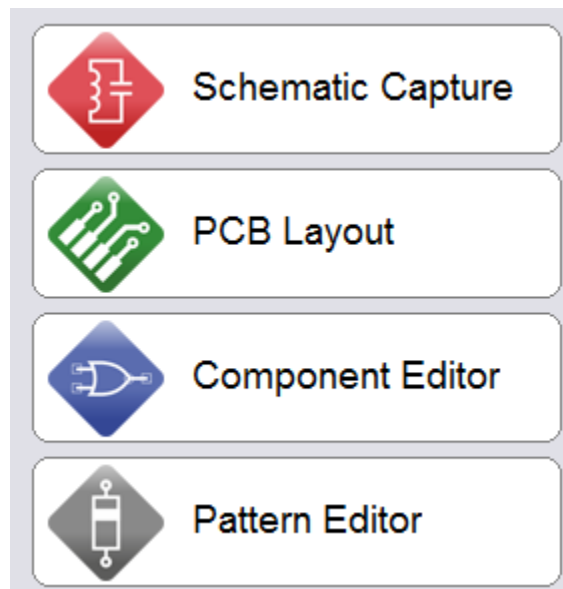
Para la realización de los diseños esquemáticos, y la posterior creación de las PCB para las diferentes funciones dentro de este proyecto, se ha utilizado el software *DIPTRACE*.



El que hemos usado para la realización de las placas de circuito impreso en el caso particular de este proyecto, es una versión gratuita que permite colocar hasta 300 pines, que son más que suficientes en el caso de nuestras necesidades.

#### *Características:*

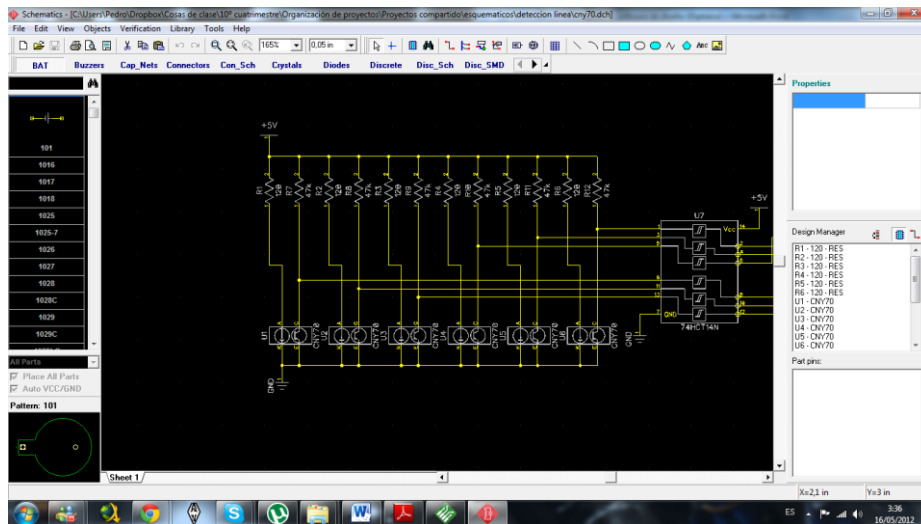
DIPTRACE contiene cuatro programas, y sirve de gestor para la creación de circuitos electrónicos.



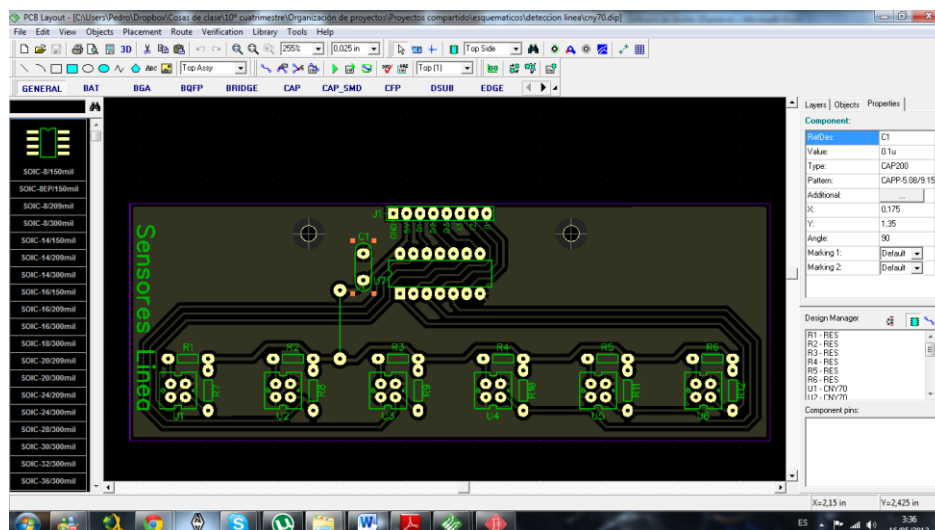
Menú de arranque de *DIPTRACE*

Los programas de los que se compone son los siguientes:

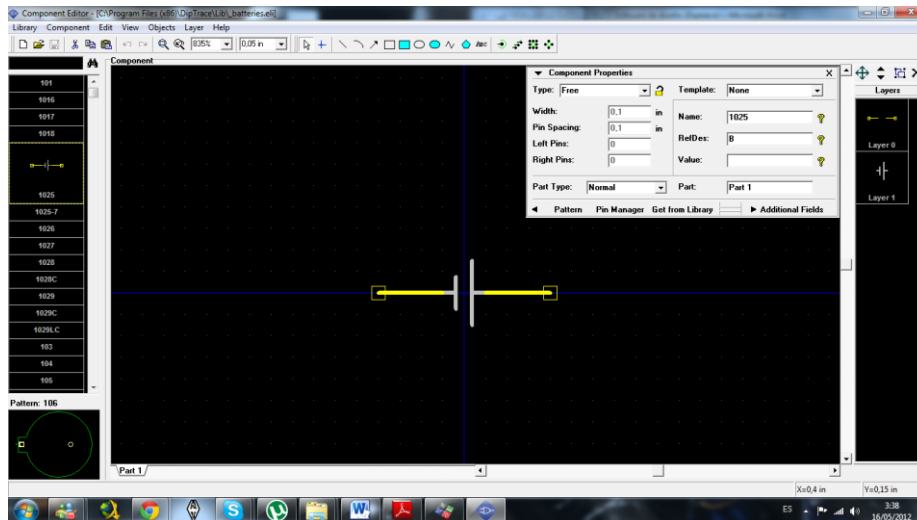
- **Schematic Capture.** Programa de creación de circuitos esquemáticos. Con este programa podremos representar el circuito esquemático con el que trabajaremos. Aquí se realizarán todas las conexiones entre los diferentes dispositivos, para poder definir totalmente el circuito de cada placa utilizada en el proyecto.



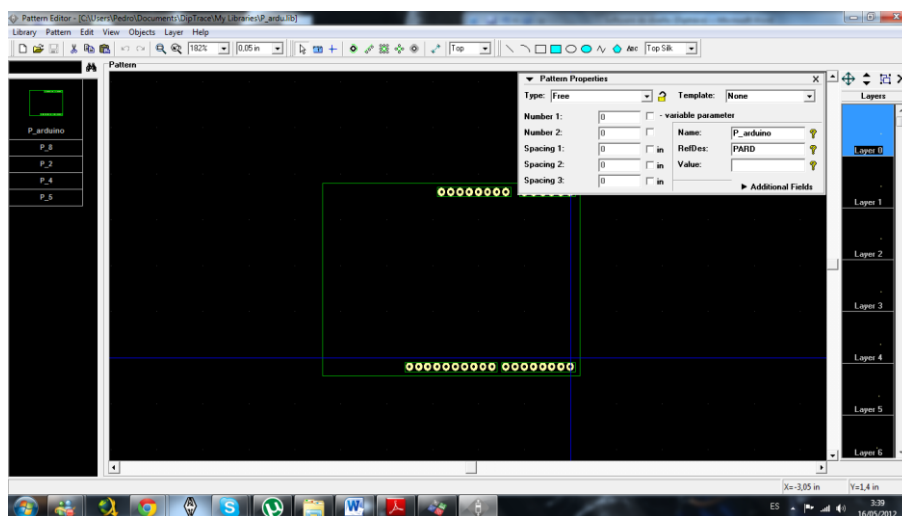
- **PCB Layout.** Programa que usamos para la creación de las PCB que se imprimen y se colocan en el coche. Se toman los esquemáticos creados en el anterior, y se realizan los cálculos de las pistas y las rutas para una correcta conexión, dentro de las dimensiones reales de cada placa. Cuenta con la posibilidad de exportar los archivos de PCB a otros formatos manejables desde otros sistemas.



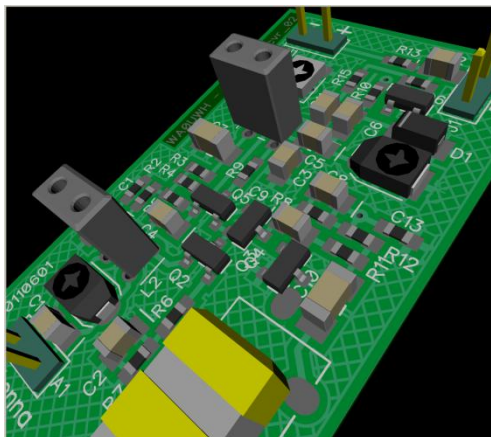
- **Component Editor.** Editor de librerías de componentes. Da a DIPTRACE la posibilidad de editar y crear librerías de componentes que podrán ser usados dentro de los esquemáticos que se desarrollan en *Schematic Capture*.



- **Pattern Editor.** Programa para creación de librerías de componentes para PCB's. Permite crear y editar componentes y librerías de componentes que son usadas para la creación de las placas de circuito impreso en *PCB Layout*. Estas librerías y componentes se relacionan con las librerías de componentes para esquemáticos, haciendo posible así, la relación directa para la creación de una PCB a partir del esquema correspondiente diseñado previamente.



Cuenta con otras prestaciones, como son la posibilidad de visualizar las PCB en 3D. Esto habilita el poder ver volumétricamente la placa de circuito impreso que se vaya a realizar.



Vista en 3D de una placa en DIPTRACE

Se pueden aplicar a los diseños ajustes de Spice, o adjuntar modelos. Se puede comprobar cómo funciona exportando en formato .cir el circuito y abrirlo desde cualquier software que interprete este formato.

También se puede exportar en otros formatos, para poder ser tratados desde otras plataformas. Esto también vale para formatos de imagen, como las .dxf que interpreta AutoCad. Todo esto añade una tratabilidad extra del formato que queramos imprimir.

Contiene desde su instalación una gran variedad de librerías estándar, que contienen más de 100000 componentes de diferentes fabricantes, además de la posibilidad de la creación y edición de otras librerías, como se ha señalado anteriormente.

### ***Análisis de la solución:***

Es un entorno muy intuitivo y fácil de aprender, además de tener tutoriales oficiales de manejo básico muy detallados.

En nuestro caso ha sido una solución muy acertada, ya que a la hora de realizar los diseños, es rápido y no necesita una parametrización muy complicada, además de ofrecer una perfecta interrelación entre los programas de la plataforma.

También nos ha servido por la posibilidad de exportar las PCB a otros formatos que se manejan desde otros entornos, como puede ser el Orcad. Esto ha resultado útil porque usamos el formato de éste último para imprimir a escala real.

En definitiva ha sido una herramienta que nos ha ayudado a reducir esfuerzos en la creación de las placas de circuito impreso. Además, al contar con la versión utilizada, es una solución que resulta muy económica.

### 1.6.3 PLACA DE CONEXIONES

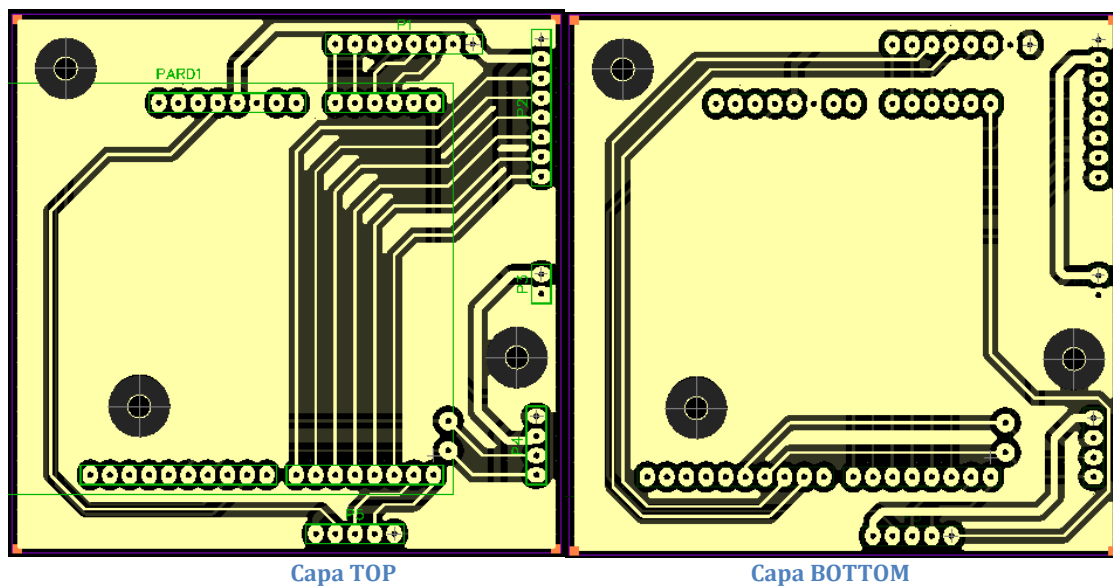
#### *Descripción:*

Para la interconexión de Arduino con las demás placas periféricas se ha realizado una PCB que sirve de unión entre el microcontrolador y dichas placas.

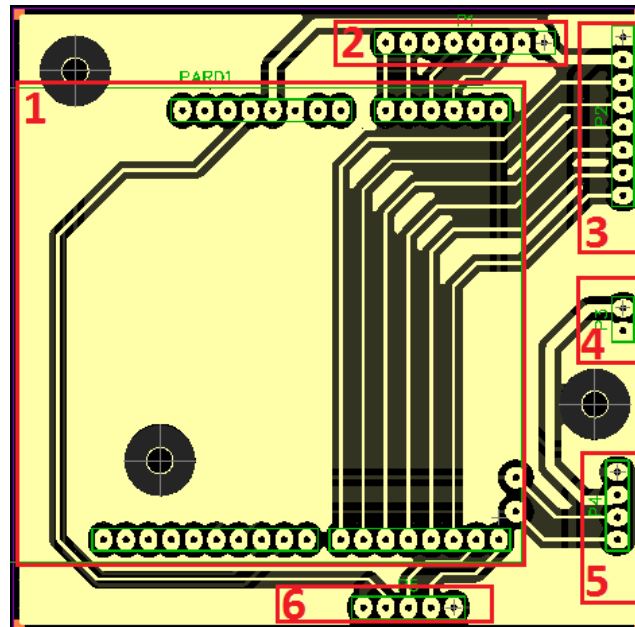
Se ha realizado en el *PCB Layout* de *DIPTRACE*, impreso en acetato y revelado.

Nos hemos asegurado de que las conexiones queden en una distribución cómoda para la posterior orientación de las placas conectadas.

#### *Placa PCB*



### Descripción de las partes:



1. Situación de la placa Arduino.
2. Conexión a la placa de drivers de los motores.
3. Conexión a la placa de sensores de línea delanteros.
4. Conexión a la alimentación.
5. Conexión a placa de lectura de encoders.
6. Conexión a la placa bluetooth.

### Características:

Se han realizado las pistas de manera que las de alimentación y masas son de 0,04 pulgadas, y el resto de 0,016 pulgadas, por tener que soportar menos corriente.

Como particularidades, para los drivers de motores, se han utilizado dos de las salidas PWM del Arduino, y para la placa bluetooth era necesaria la conexión con la patilla de 3,3V, en vez de la de 5V, por funcionar en ese rango.

El resto de patillaje de conexión, se han usado entradas y salidas digitales típicas. Se ha reservado una patilla de entrada analógica por una posible ampliación.



### *Análisis de la solución*

Hemos realizado este diseño de conexión para poder conectar el resto de placas de manera modular, facilitando el cambio de una parte en caso de avería. Por eso elegimos esta opción frente a la otra que consiste en una sola placa que incluyese todos los elementos electrónicos necesarios para el proyecto, dicha placa hubiese sido más complicada y susceptible a errores y en caso de avería a su remplazo integro.

Es una placa simple que ayuda a una mejor distribución de los elementos en el robot.

#### 1.6.4 CHASIS Y TRANSMISIÓN

##### *Descripción:*

El chasis del robot seguidor de línea está pensado de una manera compacta y sencilla, en pos de poder cambiar alguna de las piezas en el caso en que diera problemas. Esta construido a partir de materiales ligeros, con un soporte para las placas electrónicas que vienen colocadas encima.

##### *Antecedentes:*

El modelo está inspirado en los modelos de robots velocistas ya que el circuito es de similares características, y tienen sistemas de transmisión similares. En lo que pueden diferir en este caso, es que en vez de simples ruedas para la dirección y tracción del robot siguelíneas, se ha usado un sistemas de orugas, que no suelen ser frecuentes en este tipo de diseños, y que presentan una serie de ventajas e inconvenientes.

##### *Justificación:*

###### *Ventajas e inconvenientes de las ruedas frente a las orugas:*

Las orugas se suelen usar en vehículos pesados, normalmente. Consiste en un conjunto de eslabones modulares que permiten un desplazamiento estable aun en terrenos irregulares.

La mayoría de las orugas forman parte de un cinturón flexible con un conjunto de eslabones rígidos unidos a otros fuertemente. En nuestro caso, para esta aplicación, al ser a pequeña escala, se trata de cinturones flexibles de goma, continuos (no formados por eslabones).

La oruga ayuda al vehículo a distribuir el peso en una superficie mayor que la que hubiera tenido con el empleo de ruedas, y esto hace que pueda moverse por un número mayor de superficies sin hundirse debido a su propio peso. La presión que se ejerce con un robot oruga es menor que en un robot con ruedas, debido a que con la oruga, la superficie entre la que se reparte el peso es mayor.

Sin embargo, hay que tener en cuenta ciertos inconvenientes a la hora de realizar este proyecto en relación con la oruga. Estos inconvenientes pueden provocar que se necesite más par en los motores, y que, al sujetar cada dos ruedas con una misma oruga, esto tenga como consecuencia rozaduras en la sujeción de los ejes de las ruedas, o desviaciones por un mal ajuste.

## ***Materiales***

Para garantizar la mayor eficiencia energética en este aspecto se deben emplear los materiales más ligeros posibles, y suficientemente resistentes como para componer una estructura sólida.

Desde un principio se propuso un montaje con base en una placa de metacrilato, siendo un diseño sencillo y simple, con la posibilidad de cambiar casi modularmente cualquier parte que de problemas.

Para la sujeción de las ruedas no tractoras se han usado otras piezas de metacrilato que van perpendicularmente a la base general, y por las que atraviesan los ejes de dichas ruedas. Los motores tienen de serie una base que se atornilla a la base general y quedan fijados por la parte de debajo. Las placas han sido diseñadas con orificios para tornillería que también se fijarán a la base por encima, con la excepción de la placa de los sensores de línea (CNY70) que irá sujeta por debajo y en la parte delantera del robot para tener un control anticipado a la ruta.

## ***Construcción y funcionamiento:***

El sistema realiza tracción delantera a dos ruedas, transmitiendo energía cinética a dichas ruedas, que comunican con las traseras mediante orugas.

El robot no alcanza mucha altura, disminuyendo todo lo posible el que se desestabilice en marcha y en las curvas. Con las orugas se aporta estabilidad al repartir el peso de una mejor manera.

El sistema de transmisión está pensado para que pueda subir una rampa de 30º, que es la especificación que se demanda como uno de los objetivos del proyecto.

## ***Análisis de la solución:***

La oruga hace que se reparta mejor el peso y aporta una mayor estabilidad en el trazado junto al diseño general del chasis, que, como ya se ha señalado, no alcanza mucha altura con este fin.

Por otro lado, los motores pueden demandar un consumo elevado debido a que se requiere un par motor mayor al usar orugas, pero en cualquier caso disponemos de alimentación suficiente.

### 1.6.5 DRIVER MOTORES

#### *Descripción:*

Los motores que usamos para la tracción del robot están elegidos por las propias especificaciones de diseño. Se trata del motor reductor C-6065, de Fadisel.



C-6065 de Fadisel

Estos motores tienen un consumo máximo de 0,85A a 3V. Esto le permite girar a 92 rpm.

El driver que nos permite manejar tal cantidad es el L298N, además de ser de fácil adquisición y montaje cómodo. Este driver lo usaremos con sus correspondientes diodos para protección.



Driver para motores L298N

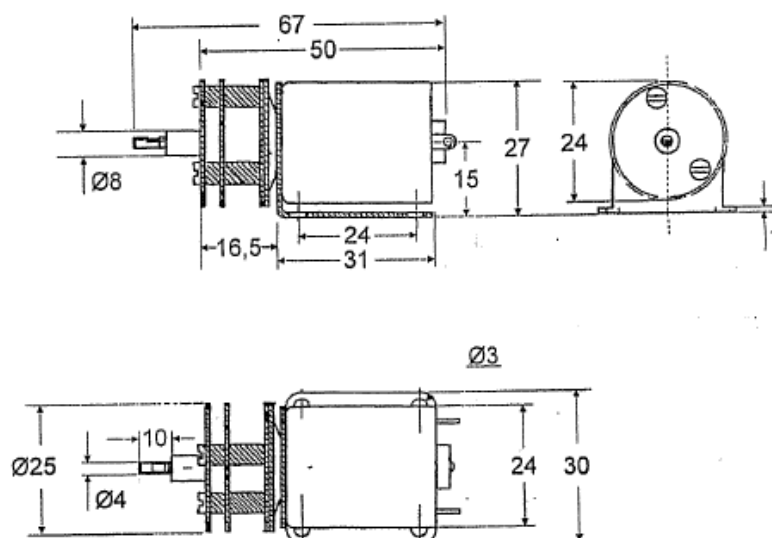
A partir del driver hemos construido la correspondiente placa de conexiones con los motores de tracción delantera.

### *Características:*

El motor C-6065 tiene las siguientes propiedades técnicas:

- Velocidad a 1,5V: 46 rpm.
- Corriente sin carga: 0,12A (1,5V).
- Corriente a máxima eficiencia: 0,53A (1,5V).
- Par de giro a 1,5V: 6200 g.cm.
- Velocidad a 3V: 92 rpm.
- Corriente sin carga: 0,16A (3V).
- Corriente a máxima eficiencia: 0,85A (3V).
- Par de giro a 1,5V: 1300 g.cm.
- Eje de salida: diámetro de 4mm.
- Fijación: soporte integrado con cuatro taladros de diámetro igual a 3mm.

### *Dimensiones del motor:*



Dimensiones del motor C-6065

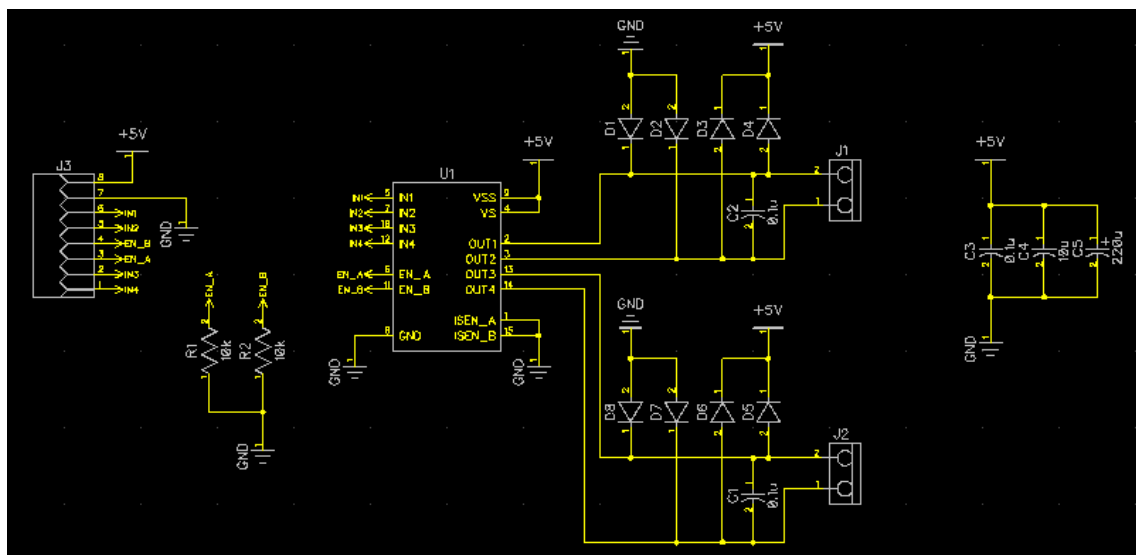
## *Diseño de la PCB*

El diseño de la placa del driver de los motores ha seguido los siguientes requisitos de diseño:

- Basado en el driver L298N.
- Diodos de 1A para protección del driver.
- Condensador de almacenaje para picos de consumo de 220 $\mu$ F.
- Condensadores de filtrado de ruido de 0.1 $\mu$ F y 10 $\mu$ F.
- Condensadores de filtrado (0.1 $\mu$ F) lo más cerca posible de las regletas de conexión de motores.
- Dos regletas de 2 pines para fácil conexión de motores.
- Conector de 8 pines: 2 para selección de sentido y 1 de velocidad por canal más la alimentación.

## *Esquemático:*

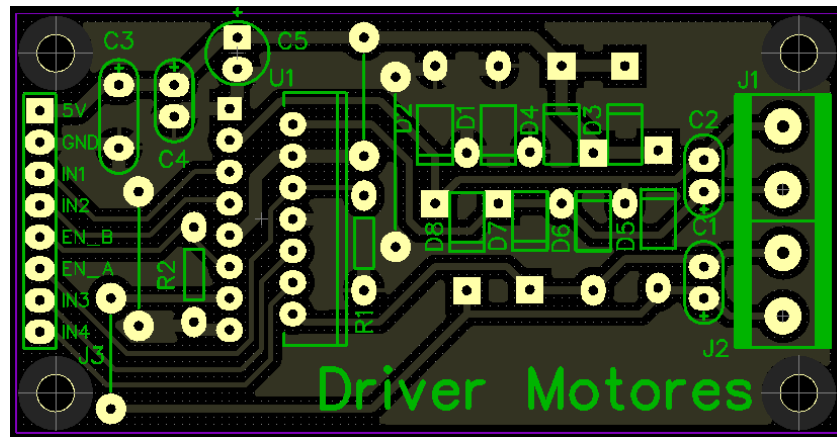
Aquí podemos ver lo que se ha señalado arriba, en forma de imagen esquemática del circuito:



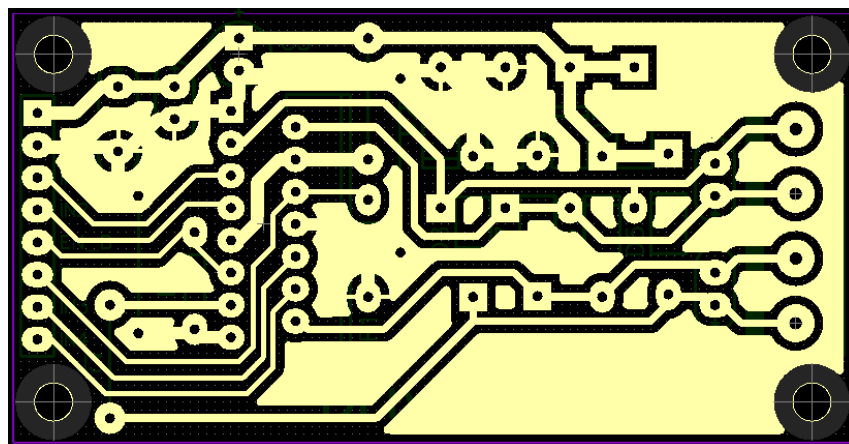
Archivo esquemático del circuito de driver de motores

### *Placa PCB:*

Es una placa de circuito impreso a una sola cara, pero mostraremos las dos caras, para una mejor visualización y comprensión de la distribución de los dispositivos en la placa:



Capa TOP



Capa BOTTOM

### *Análisis de la solución:*

La solución adoptada se ha realizado de esta manera porque el motor venía dado por las especificaciones iniciales antes de realizar el proyecto. Además, el driver se ha especificado de esta manera para poder realizar la función de control de la mejor manera posible, y para poderlo conectar al sistema central Arduino de una manera inmediata y sencilla.

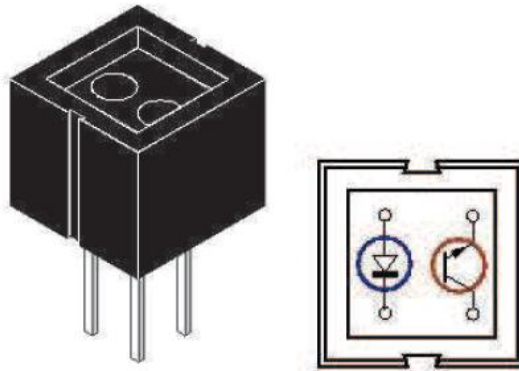
### 1.6.6 SENSOR CNY70

#### *Descripción:*

El sensor de percepción para la lectura de línea y de encoders que se ha empleado es el CNY70, el cual fue impuesto desde un principio como parte de los requerimientos.

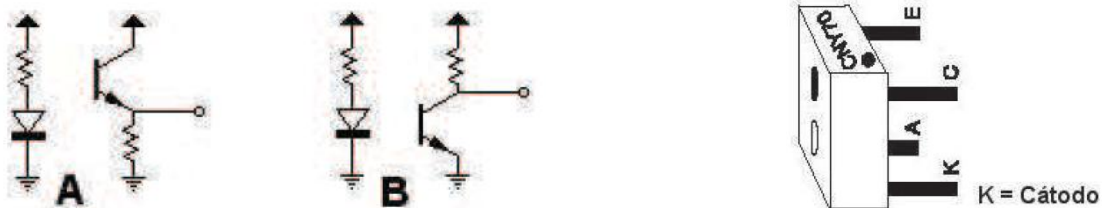
Es un sensor de corto alcance basado en un emisor y receptor de luz, cuyo funcionamiento se basa en la capacidad de reflexión del objeto a leer y la detección del rayo luminoso que se refleja en el.

#### *Esquema:*



El CNY70 tiene cuatro pines de conexión. Dos de ellos se corresponden con el ánodo y cátodo del emisor, y las otras dos se corresponden con el colector y el emisor del receptor. Los valores de las resistencias son típicamente 47K ohmios para el receptor y 120 ohmios para el emisor.





### *Funcionamiento y disposición:*

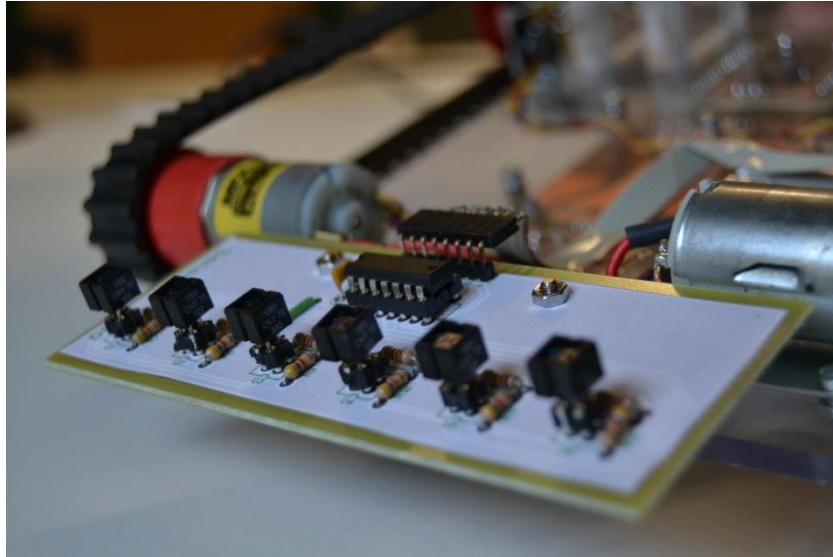
El CNY70 devuelve por la patilla de salida correspondiente, según el montaje, un voltaje relacionado con la cantidad de luz reflejada por el objeto. Para el montaje A, se leerá del emisor un '1' cuando se refleje luz y un '0' cuando no se refleje. Para el montaje B, los valores se leen del colector, por lo que son inversos a los del A.

Si conectamos la salida a una entrada digital del microcontrolador, entonces obtendremos un '1' o un '0' en función del nivel al que el microcontrolador establece la distinción entre ambos niveles lógicos.

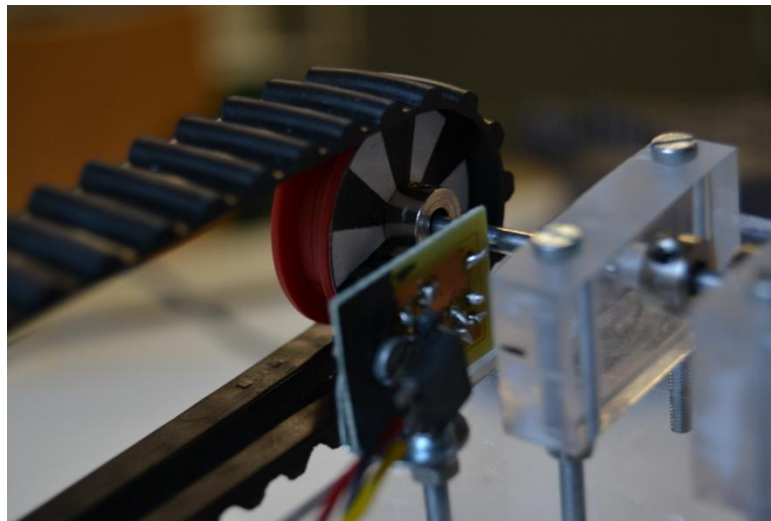
Otra posibilidad es conectar la salida a una entrada analógica. De este modo, mediante un conversor A/D se pueden obtener distintos valores. Esto permite la detección dinámica de blanco y negro (muy útil cuando el recorrido presenta alteraciones en la iluminación aunque en este caso no contamos con esa posibilidad). Esta aplicación se ha descartado debido a su mayor complejidad, prefiriendo un método puramente digital.

Se ha elegido una distribución de 6 sensores, para mayor estabilidad en el ciclo de control que por ejemplo, con uno que solo lleve 4.

Se ha elegido una disposición en línea recta porque se toma como referencia la disposición de un robot velocista, y no rastreador. En este tipo de robots, dicha distribución es simple, se trata de una línea recta en la que, como máximo, la línea será percibida por dos sensores, nunca por más. Los sensores son equidistantes unos de otros, siendo dicha distancia la mitad de la anchura de la pista que se va a leer.



Sensores de lectura de línea en la parte delantera inferior del coche.



Sensores CNY70 para la lectura de encoders relativos de las ruedas no tractoras.

### *Simulación:*

En este caso aunque sí existen recursos software para la simulación de este tipo de sensores (aunque no vienen integrados en los programas de base), las pruebas con este dispositivo han sido realizadas en el laboratorio.

### *Análisis de a solución:*

El único inconveniente del sensor es la necesidad de tener que situarlo muy próximo al objeto para detectar correctamente la reflexión. Por lo demás, es una solución muy buena y barata para la detección de línea negra o blanca.

### 1.6.7 PLACA SENSORES DE LÍNEA

#### *Descripción:*

Para la detección de la línea que ha de describir el robot, contamos con los sensores CNY70. Para la correcta colocación de dichos sensores, se ha realizado en DIPTRACE una placa de circuito impreso que se colocará enfrentada a la ruta en el suelo y a una distancia mínima para conseguir leerla.

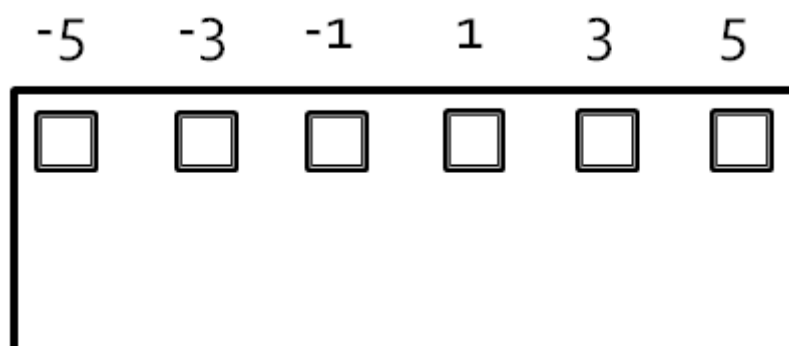
La línea a seguir es de 3cm de ancho y siendo éste un robot de tipo velocista, se ha decido realizar una matriz de 6 sensores dispuestos en recta, de forma transversal a la línea, siendo la separación entre ellos la mitad del ancho de la línea, es decir, 1,5cm.

La distancia al suelo debe ser como máximo 5mm para evitar fallos en la detección por la iluminación externa, que puede mejorarse tapando los sensores.

#### *Características:*

En la placa de sensores se sitúan los diferentes sensores de línea en disposición recta. Se disponen de tal forma que como máximo, se leerá la pista con dos de los sensores.

De cara al control, realizaremos un control proporcional en función de los sensores que se detectan en cada momento. El peso para el control proporcional de cada sensor en la placa será el siguiente:



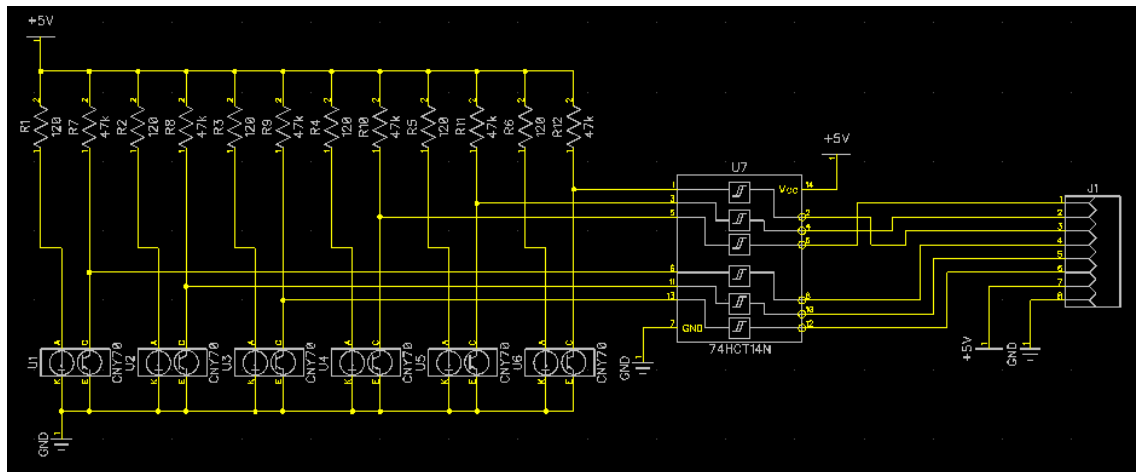
**Distribución del peso para el control proporcional en los sensores**

Para el control, tenemos en cuenta el error proporcional que se tiene en cada situación a lo largo de una ruta cuando se esta leyendo con los sensores. Este error es el resultado de sumar el peso de los sensores que están activos, y luego dividir entre el número de sensores activos (evitamos el uso de coma flotante):

$$Error = \frac{\sum_{i=1}^n W_{sensor}}{n}$$

### Esquemático:

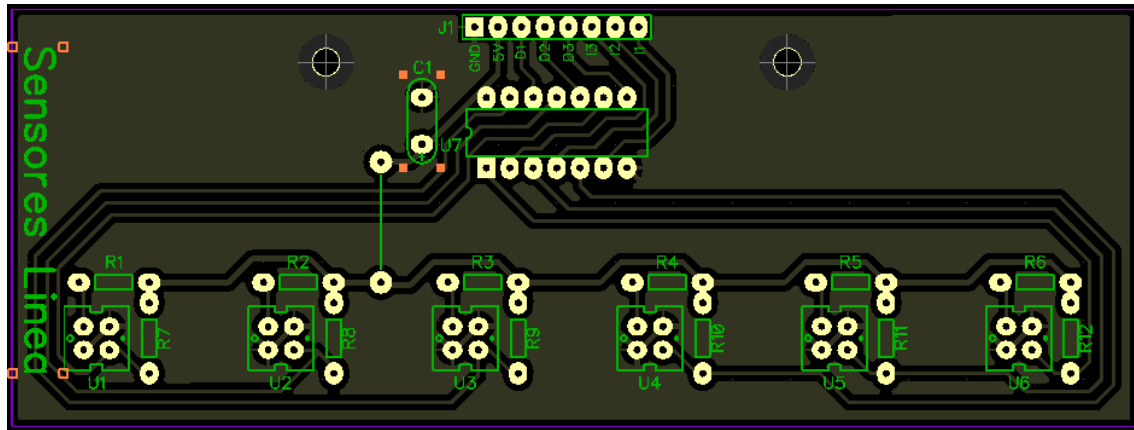
El archivo esquemático de diseño de la placa de sensores de línea es el siguiente:



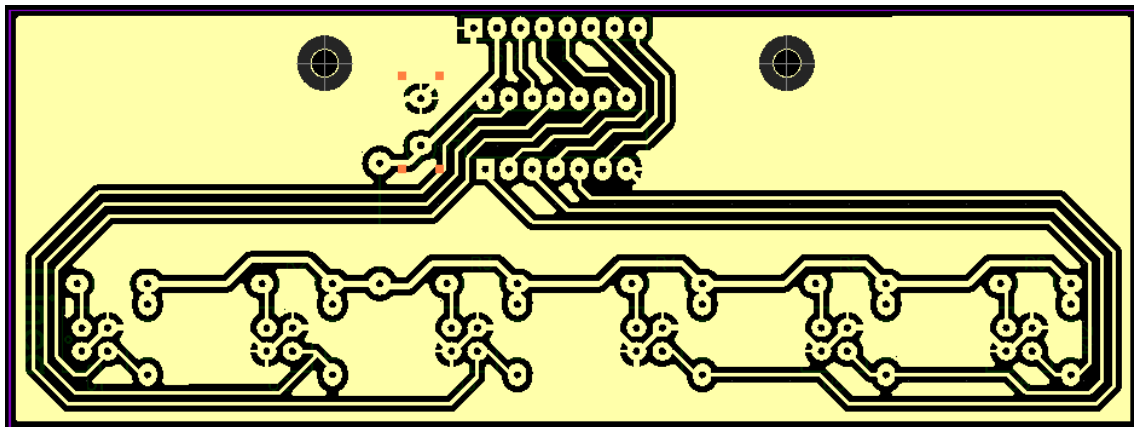
Se puede ver que hemos colocado las resistencias indicadas en el apartado del CNY70 por cada sensor, y se han conectado a disparadores Schmitt para crear la señal cuadrada que recibiremos para saber si están leyendo línea o no.

### ***Placa PCB:***

Aquí vemos la distribución real de los sensores con respecto al disparador y a la tira de pines de la conexión externa a la placa central:



Capa TOP



Capa BOTTOM

### ***Análisis de la solución:***

Los CNY70 estaban establecidos en los requisitos iniciales del diseño.

Se ha decidido colocar la placa de esta manera, por proporcionarnos una capacidad de control proporcional por programación cómoda. Se ha colocado la placa en la parte delantera del coche y a escasos milímetros del suelo con tornillería para una lectura fiable.

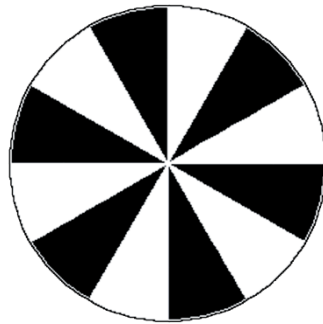
### 1.6.8 ENCODERS

#### *Descripción:*

Para la medición del recorrido que el robot siguelíneas irá haciendo a lo largo de cada ruta, se han utilizado encoders relativos. Estos encoders se han situado en las ruedas no tractoras (traseras) por la parte interior.

#### *Características:*

Cada encoder está dividido en 12 partes, que alternan entre blanco y negro cada una. Esto sirve para poder leer los pasos del encoder con uno de los sensores CNY70.



Encoder de 12 divisiones

La lectura de los encoders se realiza por medio de sensores CNY70 que a su vez están conectados a inversores schmitt para obtener una señal cuadrada, la cual es más fácil de leer mediante nuestro micro.

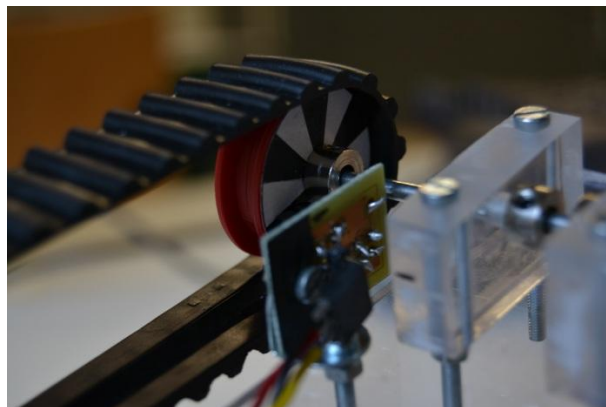
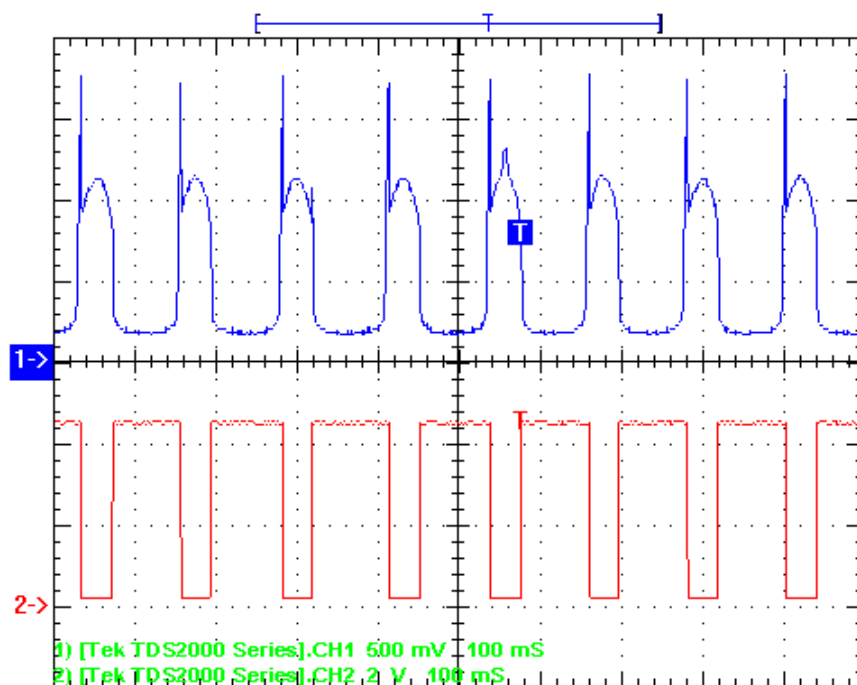


Imagen del encoder colocado en la rueda, con la placa de detección.

### Capturas con el osciloscopio:

Alimentando el motor a 3 voltios y midiendo la entrada y la salida de la señal introducida en el encoder obtenemos la siguiente señal:



En azul, la salida de los sensores CNY70

En rojo, señal cuadrada proporcionada por los inversores schmitt.

Las medidas tomadas a la entrada y a la salida son las siguientes:

Measurement Method	[Tek TDS2000 Series].Data.Waveforms.C		[Tek TDS2000 Series].Data.Waveforms.C	
Measurement	Value	Units	Value	Units
Frequency	8.1026	Hz	8.1697	Hz
Pos. Pulse Width	3.9454m	S	84.400m	S
Neg. Pulse Width	119.47m	S	38.004m	S
Rise Time	5.6700m	S	326.04u	S
Fall Time	39.821m	S	320.00u	S
Pos. Duty Cycle	31.968m	%	689.52m	%
Neg. Duty Cycle	968.03m	%	310.48m	%
Pos. Overshoot	0.0000	%	18.519m	%
Neg. Overshoot	0.0000	%	0.0000	%
Peak to Peak	1.6200	V	4.4000	V
Amplitude	1.6200	V	4.3200	V
High	1.7600	V	4.4800	V
Low	140.00m	V	160.00m	V
Maximum	1.7600	V	4.5600	V
Minimum	140.00m	V	160.00m	V
Mean	477.56m	V	3.1955	V
Cycle Mean	486.70m	V	3.1260	V
RMS	636.32m	V	3.7494	V
BurstWidth	903.86m	S	905.20m	S
Period	123.42m	S	122.40m	S
Energy	404.74m		14.052	
CEnergy	51.256m		1.6807	
ACRMS	420.42m	V	1.9602	V
CRMS	644.44m	V	3.7055	V



### *Análisis de la solución:*

La solución tomada es simple, fácil de montar y además usamos el mismo sensor que en el caso de la lectura de la pista, lo que nos facilita el circuito de acondicionamiento, y lo más importante, funciona.

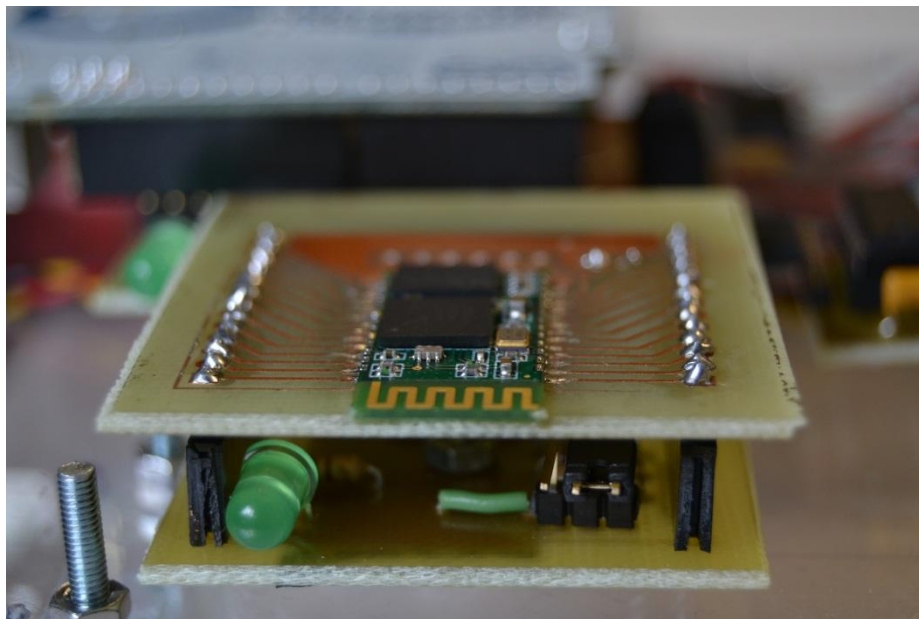
### 1.6.9 PLACA DE COMUNICACIÓN BLUETOOTH

#### *Descripción:*

Uno de los requisitos especificados para la asignatura era la comunicación del robot oruga de forma remota con otros dispositivos. El método elegido fue la comunicación por bluetooth.

Una vez diseñada la placa para poder acoplar el chip integrado de bluetooth se diseñó la placa “inferior” para poder conectarla a la placa de conexiones periféricas del Arduino.

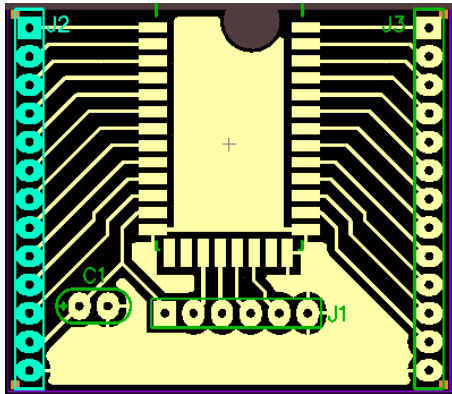
Se le colocó un diodo LED para saber en qué momento está conectado con otro dispositivo. Cuando el LED está parpadeando, no hay ningún dispositivo conectado. En el momento en que está encendido sin parpadeo significará que está conectado con un dispositivo.



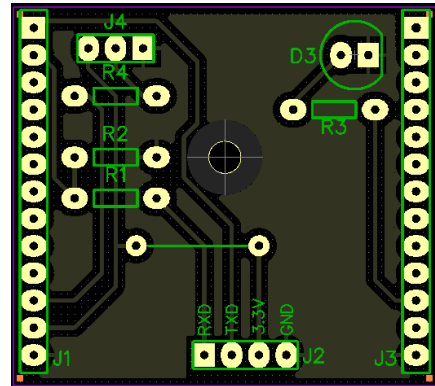
Placa de comunicación bluetooth con el LED de conexión.

### *Características:*

La comunicación bluetooth consta de dos PCB's:



Placa breakout, para situación  
del módulo bluetooth



Placa de expansión  
para conexión con Arduino

### *Análisis de la solución*

Fue necesaria la construcción de dos placas inicialmente independientes. La primera y superior vista en la fotografía es la que conecta el chip integrado bluetooth con nuestra PCB para poder conectarla luego a otra de mayor tamaño y así poder tener acceso a la misma. La soldadura tuvo que ser minuciosa puesto que la separación entre los pines era muy pequeña.

La segunda placa está conectada a la placa bluetooth mencionada anteriormente y a su vez a la placa de conexiones periféricas del Arduino. En esta placa se añadió el LED de conocimiento de conexión de comunicación.

## 1.6.10 CARGADOR DE BATERÍAS

### *Antecedentes*

Un cargador de baterías es un dispositivo que se utiliza para almacenar energía en un acumulador secundario, o recargable, mediante la conexión a un sistema productor o transmisor de electricidad. Los cargadores actuales presentan diferentes diseños y cubren distintas funciones de acuerdo con la corriente de carga de entrada o del formato de la batería que se quiere reutilizar.

Los cargadores de baterías que actualmente se pueden encontrar en el mercado son alrededor de una decena de tipos, que cubren las necesidades de las aplicaciones domésticas e industriales, y, cuyo desarrollo se orienta hacia la mejora de las capacidades de carga, la prolongación de su vida útil y la seguridad y sencillez de empleo.

Cargadores de baterías podemos encontrarlos simples, de acción 'de goteo' o de proceso lento, con temporizador de carga, los formatos llamados inteligentes, los denominados rápidos, los de pulso, los inductivos, los basados en conexiones USB o los solares que cargan por acción de la energía que se puede recoger de paneles fotovoltaicos.

El cargador de baterías simple es un sencillo dispositivo que no tiene en cuenta ni la carga de la pila que se ha de volver a regenerar, ni el tiempo que se precisa para completar la acción. Normalmente mantienen tensiones y corrientes constantes. Se trata, por tanto, de un elemento accesorio sencillo y barato. Los cargadores sencillos tardan más tiempo en recargar las baterías para evitar los efectos del exceso de carga que puede llegar a estropear la misma pila.

Los cargadores de baterías lentos, por su parte, son cargadores sencillos con sistemas de carga que realimentan la batería recargable muy despacio. Con estas baterías, nunca hay problemas de sobrecarga. Por ejemplo, las baterías se pueden dejar indefinidamente en los recargadores sin riesgo alguno. Es su sistema de seguridad y una garantía de funcionamiento eficiente.

Los cargadores con temporizadores son unos dispositivos que desde el comienzo de la década de los noventa se popularizaron para su empleo en pequeños equipos portátiles. Con el tiempo, y con el aumento de la capacidad de carga de las pilas recargables, los sistemas de estos recargadores se han convertido paradójicamente en fuente de problemas. En su día, se vendían en paquetes, cargadores y pilas juntos.

Así, los cargadores sólo podían operar con las pilas que acompañaba el fabricante, cuando el nivel de carga de las pilas fue aumentando, los cargadores se volvieron insuficientes para alcanzar los niveles de cargas adecuados para las nuevas pilas que salían al mercado. Este es un defecto que los consumidores reprocharon a las industrias suministradoras de las primeras pilas de níquel-cadmio.

Los cargadores inteligentes son, en su género, los más efectivos. Porque un cargador inteligente, puede controlar la tensión de batería, la temperatura y el tiempo de carga para determinar el nivel óptimo de corriente que se necesita para la recarga. De este modo, incluso en las baterías del tipo Ni-Cd y las del formato NiMH, el voltaje introducido en las baterías aumenta lentamente durante el proceso de carga hasta que la batería está completamente cargada. Después, la tensión disminuye y el cargador deja de cargar porque está completada la carga de energía.

Estos cargadores reciben la denominación de 'delta-V' porque su referencia de recarga es el voltaje. Un problema común con estos cargadores es que la cuantificación de los 'delta-V' puede llegar a ser muy variable, incidiendo sobre la eficiencia misma del sistema de recarga. Así, se llegan a dar casos en cargadores inteligentes que continúan cargando después de haber completado toda la carga de la batería. La solución de algunos fabricantes para estos casos anómalos ha sido el diseño de diferentes modos de corte de energía para evitar sobrecargas.

Un típico cargador rápido e inteligente carga una batería hasta un 85% de su capacidad máxima en menos de una hora, luego el sistema cambia a carga lenta, e invierte varias horas en alcanzar el límite máximo de la carga de la batería.

Los cargadores rápidos, por su parte, cargan con mayor velocidad pero también presentan circuitos de control para no dañar los elementos sensibles de las pilas. Por ejemplo, algunos modelos añaden sistemas de refrigeración para mantener la temperatura de las baterías dentro de unos márgenes de seguridad.

Los llamados cargadores de pulso controlan la intensidad de la carga, la frecuencia de la carga y, por supuesto, su voltaje. Esta tecnología opera con cualquier tipo de baterías, con todos los voltajes y con las capacidades y la química de las baterías actualmente en uso. Una de las ventajas de los cargadores de pulso es que se evitan los riesgos de sobrecalentamiento de las baterías.

Los cargadores basados en los sistemas USB son otros de los modelos más corrientes emplean los cables standard usados en los puertos de contacto de ordenadores portátiles y teléfonos móviles y los aprovechan para generar cargas de bajo voltaje para pilas que se utilizan en pequeños equipos.

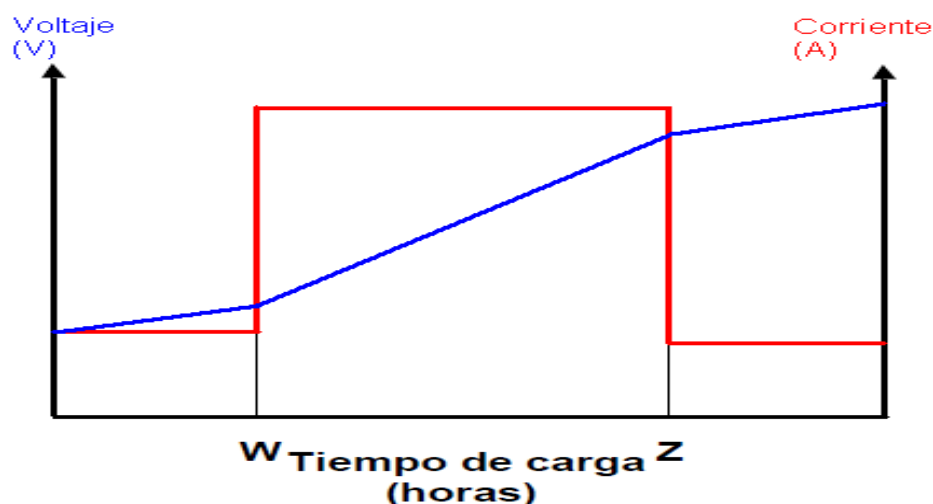
Los cargadores de baterías solares aprovechan la energía captada por paneles fotovoltaicos que convierten la radiación solar en energía eléctrica. Esta energía eléctrica va directamente a las baterías que se recargan. Normalmente, se usan para empleos muy limitados como la alimentación de las baterías pequeñas para caravanas, casas de campo, electrónica de gran consumo o usos domésticos.

### *Solución adoptada*

Para nuestro cargador de batería se a optado por realizar un sistema inteligente, que nos cargara las baterías en tres etapas distintas, la primera realizara una carga lenta por 15 minutos para estabilizar la carga de las baterías y su temperatura, corriente constante del 10% de la carga total de la batería, después realizaremos una carga rápida a una corriente constante de 1/3 de la corriente de las baterías, en esta etapa vigilarémos la temperatura de las baterías constantemente, si esta excede de un valor establecido dependiendo del tipo de estas, se bajara la corriente de carga y si aun así esta temperatura excede del limite establecido se volverá a la corriente del 10%.

En la ultima etapa una vez alcanzada una tensión determinada, se pasara a carga flotante con una corriente establecida de 10%, para cargar las baterías completamente sin peligro de sobrecargas.

También se ha dispuesto un sistema temporizador que nos permite parar la carga de las baterías si el tiempo excede de lo habitual, para que no exista problemas de sobrecargas en baterías con poca carga.



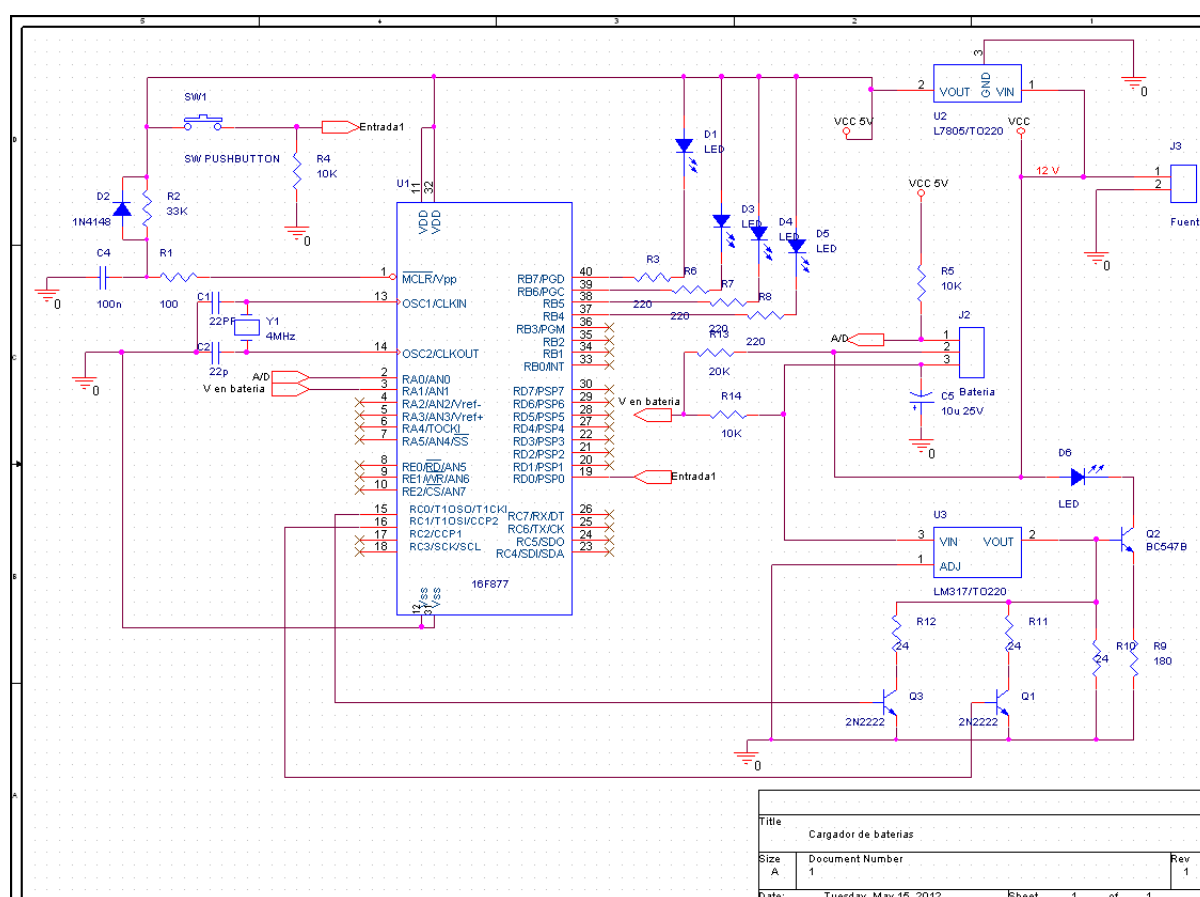
## Análisis

El circuito está basado en un PIC16F877A, este dispone de ocho conversores A/D internos, que nos permitirán calcular la temperatura de las baterías y el voltaje de la carga.

Para la regulación de la intensidad nos serviremos del LM317, que nos permite regular la intensidad por medio de la resistencia presente entre su terminal VOUT y ADJ, esta resistencia será controlada por el PIC, que por medio de los transistores 2n2222 nos ajustara la resistencia total deseada en cada etapa de la carga.

También disponemos de un botón para comenzar el circuito de carga de nuevo.

Y tres leds indicadores del valor de la corriente actual de carga, y otro el D6 que varia dependiendo de la carga actual de la batería.



**ANEXOS**



## ARDUINO UNO REV3 SCHEMATIC

# PLANOS

## PLANO DEL COCHE SIGUE LÍNEAS

## PLANO MOTORES Y PIEZAS DE SUJECCIÓN

## PLANO PCB CONEXIONES

## PLANO PCB DRIVER DE MOTORES

## PLANO PCB SENSORES DE LÍNEA

## PLANO PCB ENCODERS



## PLANO PCB BLUETOOTH BREAKOUT

## PLANO PCB BLUETOOTH EXPANSIÓN

# **ESTADO DE MEDICIONES**

**ESTADO DE MEDICIONES**

# **PRESUPUESTO**

**CUADRO DE PRECIOS UNITARIOS DE MATERIALES**

**CUADRO DE PRECIOS UNITARIOS DE MANO DE OBRA**

**CUADRO DE PRECIOS UNITARIOS TOTALES**



**CUADRO DE PRECIOS UNITARIOS DESCOMPUESTOS**





**PRESUPUESTO**

**RESUMEN DEL PRESUPUESTO**