Fundamentos y conceptos básicos para robot sigue línea.

Instituto Tecnológico de Mexicali.

Felix Nicasio, Veronica Quintero Rosas, Arnoldo Diaz Ramirez, Claudia Martinez Castillo

Índice

Introducción.

El objetivo fundamental del siguiente manual es ayudar al lector a comprender y entender los principios básicos para la fabricación de un robot seguidor de línea, lo cual puede estar dado de dos formas, ya sea fondo blanco y line negra o viceversa sin embargo cualquiera que sea el caso la lógica y fundamentos son los mismos.

Se pretende dar a conocer conceptos básicos de electrónica para el caso de quienes no estén familiarizados con estos, o en caso de tener dudas poder facilitar su comprensión, en lo siguiente se tocaran temas que van desde aspectos a considerar para el diseño físico, los componentes electrónicos necesarios así como la programación del mismo y las pruebas finales. Además se retomaran conceptos previos con fines de retroalimentación y recordatorio.

Comenzaremos con la diferenciación de los dos tipos de señales fundamentales, señales analógicas y digitales.

Tanto las señales analógicas como las digitales son utilizadas para transmitir información, generalmente a través de impulsos eléctricos.

En lo siguiente se da una pequeña descripción da cada una para facilitar su comprensión y diferenciarlas entre sí.

Analógica

La señal analógica es aquella que presenta una variación continua con el tiempo, es decir, que a una variación suficientemente significativa del tiempo le corresponderá una variación igualmente significativa del valor de la señal (la señal es continua y generalmente es representada por ondas senoidales).

Toda señal variable en el tiempo, por complicada que ésta sea, se representa en el ámbito de sus valores (espectro) de frecuencia. De este modo, cualquier señal es susceptible de ser representada descompuesta en su frecuencia fundamental y sus armónicos.

Básicamente esto se refiere a que una señal analógica, a lo largo que transcurre el tiempo nunca deja de tener un valor y su variación es progresiva, es decir ya sea que esta aumente o disminuya los cambios comprenden todos los posibles valores entre su máximo y su mínimo, por ejemplo supongamos un botón de volumen de perilla(potenciómetro) cuyo valor inicial es 0, y comenzamos a subir el volumen, esto en lenguaje electrónico es una variación en el voltaje que alimenta el circuito amplificador de audio, ahora este va a ir incrementando acorde al giro del botón, pero no tendrá brincos de 0v a 1v sino que incrementara poco a poco en toda la cantidad de valores que pudieran existir en este rango (0, 0.1000…., 0.1200…., 1, 1.2, 5) hasta llegar a su valor máximo.

Digital.

Por el contrario las señales digitales, llevan la información a través de código binario (**0** o **1**); donde cada bit es una representación de dos amplitudes distintas. Las señales digitales no son continuas, sino discretas y generadas por modulación digital.

En este caso las ondas no son senoidales, sino cuadradas, esto significa que para el caso de las señales digitales los valores tienen solo dos estados, los cuales son conocidos como estados lógicos, **0** y **1**, sin embargo, nuevamente a nivel electrónico estos valores son realmente **0** y **5** volts respectivamente. Aquí se aprecia su principal diferencia, retomando el ejemplo anterior si este fuera el caso del volumen este solo tendría dos estados uno donde no se escucharía nada y otro donde sería su valor máximo, sin pasar por los valores intermedios.

Una vez teniendo esto claro retomaremos el punto original, el cual es la lectura y escritura de datos análogos, para mostrar esto se utilizaran pequeños bloques de código fuente de programas de Arduino, orientados a la correcta implementación de los mismos a la hora de la fabricación de nuestro robot.

Lectura y escritura de datos analógicos.

La lectura de datos o valores análogos es sumamente sencilla para el caso de Arduino, ya que nativamente este cuenta con una instrucción que facilita la lectura de los mismos. Como bien sabemos los distintos modelos de placas para desarrollo de Arduino tienen un conjunto de puertos digitales y analógicos (los cuales podemos apreciar en la figura 1), por ser el más común utilizaremos Arduino uno.

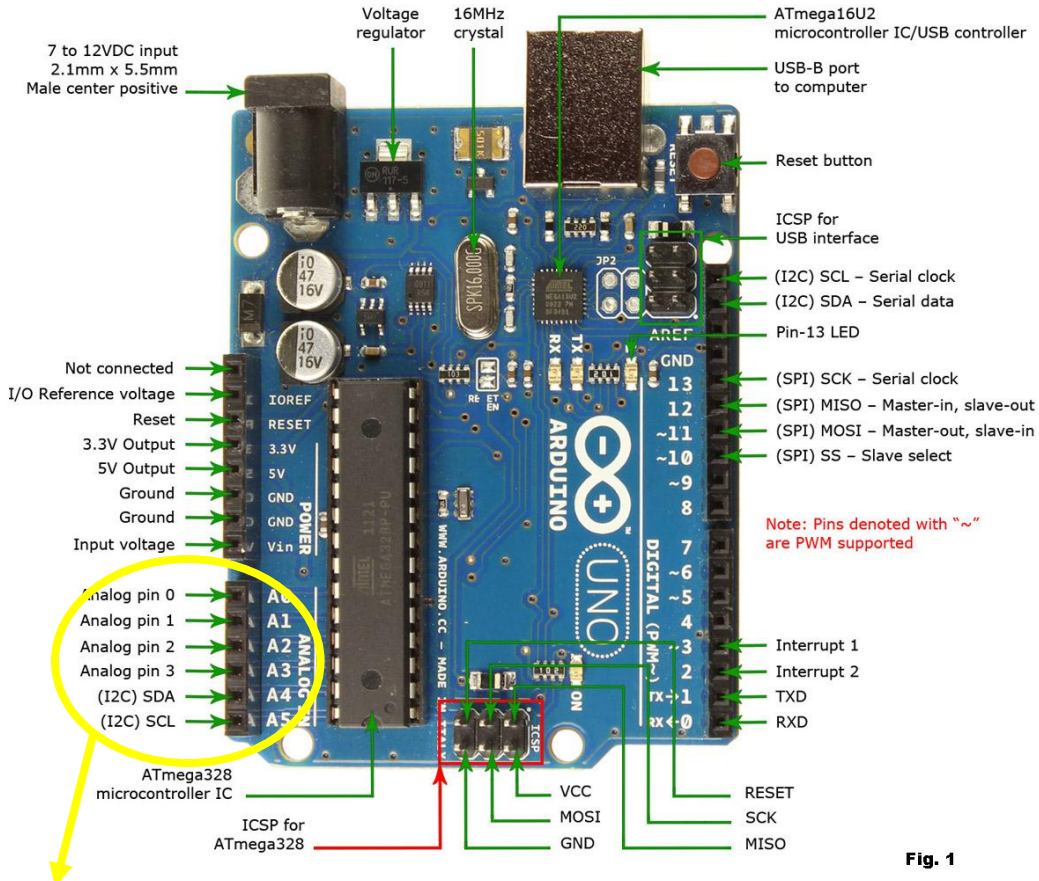


Fig. 1

Puertos usados para lectura y escritura de señales analógicas.

En esta figura podemos apreciar que se especifica la función de todos los pines o puertos de la placa Arduino uno, para no generar confusión se profundizara solo lo necesario, es decir solo mencionaremos los pines que serán utilizados. Si deseas más información en específico puedes acceder a la página oficial: ***https://www.arduino.cc/***

Una vez identificados los pines que vamos a necesitar, ahora un pequeño ejemplo, realizaremos lo mencionado al inicio, es decir la lectura de la variación de un potenciómetro, en la figura 1.2 se aprecia la forma de conexión.

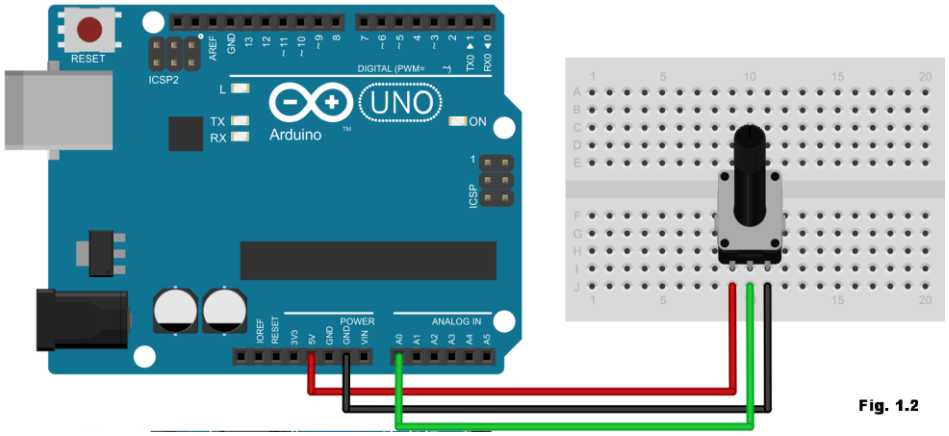


Fig. 1.2

Ya teniendo nuestro pequeño circuito conectado procederemos a hacer uso de la función que antes se mencionaba, **analogRead ()**;

```
/*
 * ********************************************************************************************
 * ************************ Instituto Tecnologico de Mexicali,************************
 * ****************** primer concurso interno de seguidores de linea ******************
 * ********************************************************************************************
 * Ejemplo que muestra la implementacion de analogRead()
 * para interpretacion de señal analogica.
 */
 //Lo primero sera especificar el pin analogico a utilizar como entrada de datos.
 int entradaDatos = A0;
 //una variable que obtendra las lecturas de A0.
 int valorDatos = 0;

 void setup() {
  // se establece la velocidad para transmitir datos en serie a 9600 baudios, esto es
  //para visualizar las lecturas en el monitor serial de arduino.
  Serial.begin(9600);
}

void loop() {
  // leer valor presente en la entrada analogica A0 y se le asigna a la variable valorDatos
  valorDatos = analogRead(entradaDatos);
  // se imprime el valor leido en el monitor serial para poder visualizarlo
  Serial.println(valorDatos);
}
```

Ya teniendo este valor podemos por ejemplo establecer un rango y definir que accion tomar cuando entre o salga del rango de valores la lectura obtenida.

Prueba de sensores CNY70.

El componente conocido como CNY70 es un sensor infrarrojo, de corto alcance (menos de 5 cm) que se utiliza para detectar colores de objetos y superficies. Contiene un emisor de radiación infrarroja (fotodiodo) y un receptor (fototransistor).

El fotodiodo emite un haz de radiación infrarroja, el fototransistor recibe ese haz de luz cuando se refleja sobre alguna superficie u objeto. Dependiendo de la cantidad de luz recibida por el fototransistor el dispositivo envía una señal de salida, la puede ser interpretada por Arduino.

Este sensor puede utilizarse como entrada digital o analógica. Cuando el sensor está orientado hacia una superficie u objeto de color negro éste absorbe gran parte de la luz emitida por el diodo. Entonces el sensor enviará un valor alto (HIGH – 1). A su vez cuando el sensor se sitúa sobre una superficie u objeto de color blanco gran parte de la luz emitida por el diodo será reflejada al fototransistor lo que provoca que el sensor envié un valor bajo (LOW - 0).

Para identificar las terminales y realizar correctamente las conexiones del sensor, tienes que colocar el sensor con la parte del fotodiodo y del fototransistor hacia arriba y los terminales hacia abajo. También debes asegurarte que la cara del sensor que está serigrafiada con el nombre del dispositivo quede situada a tu derecha, tal y como se muestra en la siguiente imagen. Una vez identificados los terminales hay que realizar las conexiones al protoboard y a Arduino.

En la figura 2 se puede apreciar la forma en la que se ha realizado la conexión.

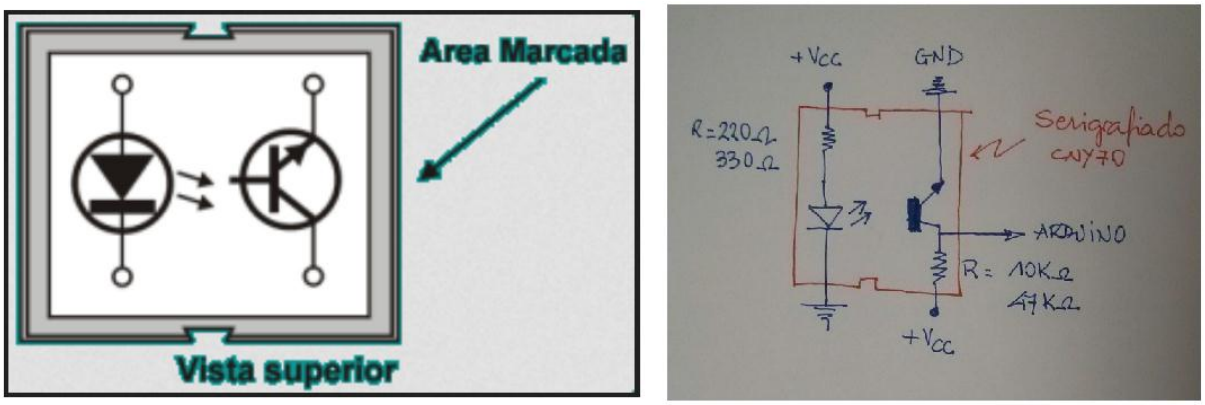


Fig. 2

Una vez terminada la conexión procedemos a realizar las primeras pruebas, utilizando nuestro primer programa, solo con pequeñísimas modificaciones para hacer más atendible lo que está sucediendo.

```
/*
 * ********************************************************************************
 * ************************ Instituto Tecnologico de Mexicali,************************
 * ***************** primer concurso interno de seguidores de linea ******************
 * ********************************************************************************
 * leyendo datos desde el sensor CNY70
 */
 //especificar el pin analogico a utilizar como entrada de datos.
 int entradaSensor = A0;
 //una variable que obtendra las lecturas de A0.
 int valorSensor = 0;

 void setup() {
  // se establece la velocidad para transmitir datos en serie a 9600 baudios.
  Serial.begin(9600);
}

void loop() {
  // se lee el valor que envia el sensor.
  valorSensor = analogRead(entradaSensor);
  // se imprime el valor leido en el monitor serial para poder visualizarlo
  Serial.println(valorSensor);
  //establecemos un pequeño retardo en cada muestreo de datos.
  delay(500);
}
```

Una forma de corroborar por lo menos que esta encendido nuestro sensor es enfocándolo con la cámara de nuestro celular, debido a que emite luz infrarroja con la cámara podremos apreciar que esta encendido o no, Posteriormente podemos colocar el sensor apuntando a superficies de distinto color, para nuestro propósito negro y blanco, y después observar los valores que nos arroja en el monitor serie de Arduino.

En caso de no poder notar que el sensor esta encendido revisar conexiones y volver a intentar la prueba en un lugar menos iluminado.

¿Qué es el PWM?

Modulación por ancho de pulso (por sus siglas en ingles pulse-width modulation), es una técnica que consiste en producir un pulso rectangular con un ciclo de trabajo determinado, este ciclo de trabajo puede variar de 0 a 100%. Ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de voltaje circulante por un circuito.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{\tau}{T}$$

D, es el ciclo de trabajo
τ, es el tiempo en que la función es positiva (ancho del pulso)
T, es el período de la función

Intentando hacer esto un poco más entendible veámoslo de forma gráfica, se muestra un pulso con un ciclo de trabajo del 50%. Es decir Ton/T = 0.5

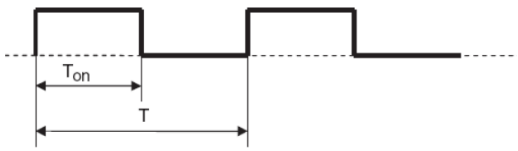


fig. 3

Tenemos tres ejemplos diferentes en los cuales se muestra lo que sucede con el voltaje cuando se le aplican distintos PWM, se genera un valor promedio acorde al ancho de pulso que se aplica, esto lo usaremos nosotros para realizar el control de la velocidad de nuestro vehículo.

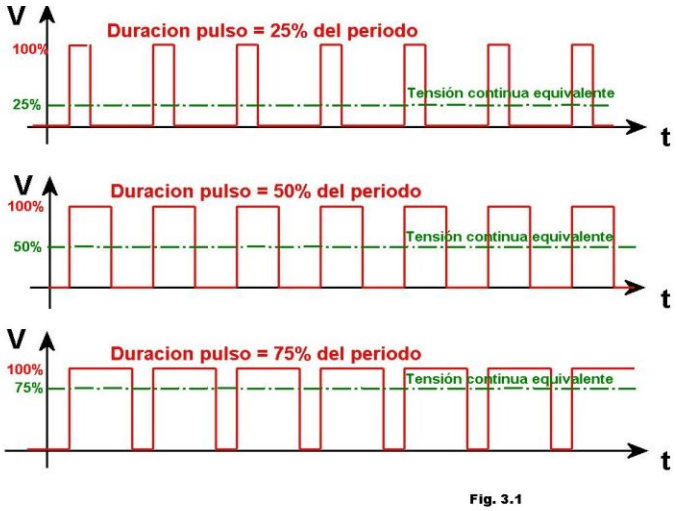


Fig. 3.1

Lo que podemos inferir de la figura 3.1 es que si tenemos un voltaje máximo de 10 v, en el primer caso al aplicar un PWM del 25% se tiene un voltaje de ese mismo valor es decir 2.5 v, de esta forma regulamos el voltaje enviado a los motores, y por ende regular la velocidad.

Fade led con PWM.

Ahora el código de ejemplo correspondiente, tenemos un LED al cual se le aplicara un PWM con ciclos de trabajo desde 0% hasta 100%, lo cual provocara que la intensidad de su brillo aumente progresivamente acorde al porcentaje del ciclo de trabajo y luego disminuye de la misma forma, a esto se le conoce como fede-in y fade-out, finalmente lo que logramos es controlar la intensidad del brillo del LED.

```
 * ********************************************************************************
 * ************************ Instituto Tecnologico de Mexicali,************************
 * ***************** primer concurso interno de seguidores de linea ******************
 * ********************************************************************************
 * fade-in y fade-out de un led con pwm
 */
// conectaremos un led al pin 9 de arduino, notese que es un pin con pwm.
int ledPin = 9;

void setup() {
//para este caso no sera necesario definir nada.
}

void loop() {
  // fade in desde el valor minimo hasta el valor maximo, con incremento de 5 unidades.
  for (int fadeVal = 0 ; fadeVal <= 255; fadeVal += 5) {
    //se manda el valor del fade al pin del led.
    analogWrite(ledPin, fadeVal);
    // pause de 30 milisegundos para el cambio de intencidad.
    delay(30);
  }
  // fade out desde el valor maximo hasta el minimo con decremento de 5 unidades
  for (int fadeVal = 255 ; fadeVal >= 0; fadeVal -= 5) {
    // se establece el vaor del fade al pin de nuestro led
    analogWrite(ledPin, fadeVal);
    // pause de 30 milisegundos para el cambio de intencidad.
    delay(30);
  }
}
```

El valor de 255 que se utiliza como maximo en el ciclo **for()** es por que el generador de PWM de arduino es de 8 bits por lo que tine 256 valores distintos de codificacion. Como observacion adicional podemos agregar que para fines de un mejor entendimiento podemos cambiar el nombre de la vaiable fadeVal por pwmVal para recordar que ese valor representa el porcentaje de ciclo de trabjao del pwm..

Transistor como switch.

Un transistor puede ser configurado para funcionar como interruptor, los estados que se manejan hablando de transistores son en corte o saturación.

No profundizaremos mucho en estos conceptos, basta con saber que cuando el transistor se encuentra en corte el interruptor está abierto y en estado de saturación es un interruptor cerrado, conociendo esto podemos utilizar un transistor como el **2n2222A** el cual es de los llamados de propósito general, para realizar un pequeño ejemplo de implementación, con un circuito sumamente básico.

Para esta demostración no será necesario programar nada así que podemos dejar Arduino de lado por esta ocasión, sin embargo será necesario conocer el diagrama de funcionamiento de nuestro transistor a utilizar. Así que será necesario tener su hoja de datos, para ello podemos utilizar la siguiente página: ***http://www.alldatasheet.com/*** en la cual podremos encontrar información de este y muchos más componentes de electrónica.

PIN	DESCRIPTION
1	emitter
2	base
3	collector, connected to case

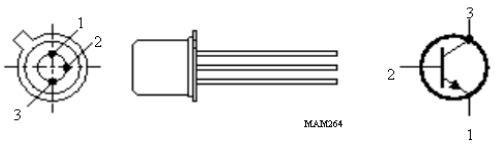


Fig.1 Simplified outline (TO-18) and symbol.

aquí podemos ver como esta conformado internamente este componente, lo que sera necesario conocer para su correcta conexión.

En la **figura 5** podemos apreciar el circuito que necesitamos, observando detenidamente podremos notar que es muy sencillo y lo que haremos será encender el led.

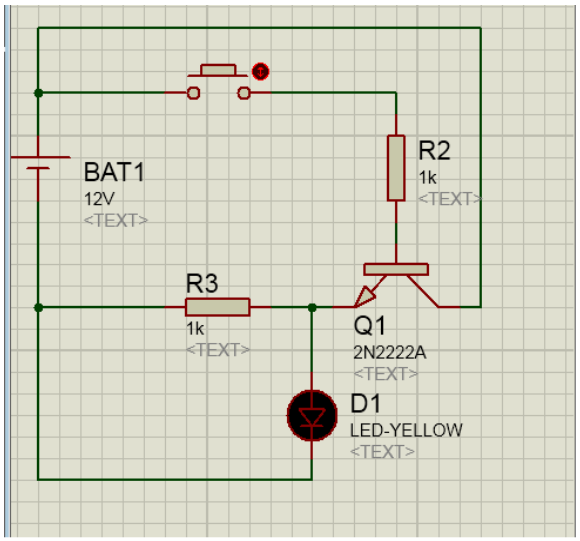


Fig. 5

Todo comienza presionando el botón el cual energiza o excita lo que conocemos como la base del transistor y esto a su vez permite el paso de la energía entre colector y emisor, para finalmente encender el led, en la siguiente figura podemos apreciarlo.

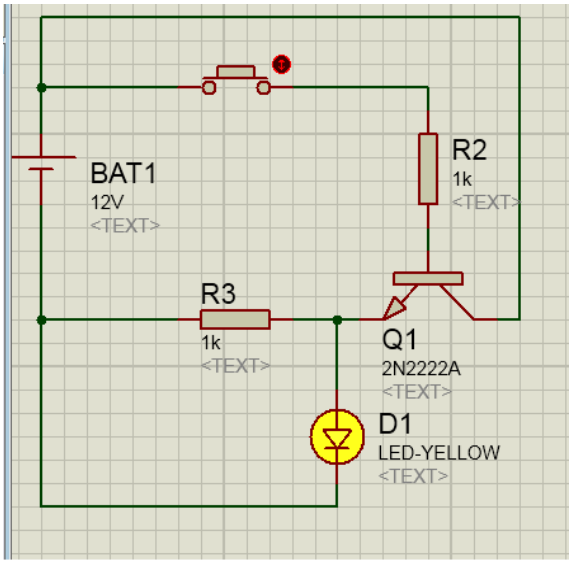


Fig. 5.1

Manejo de motores.

Modificando un poco el ejemplo anterior se mostrara como podemos realizar el control de motores además de incluir un poco de programación para fines demostrativos.

En esta ocasión utilizaremos el diagrama mostrado en la **figura 6** así que asegúrese de realizar la conexión correcta y conectar el cable indicado al pin 9 de Arduino o a cualquiera con capacidad de utilizar **PWM,** al igual que anteriormente controlamos la intensidad de un led, esta vez controlaremos la velocidad de nuestro motor.

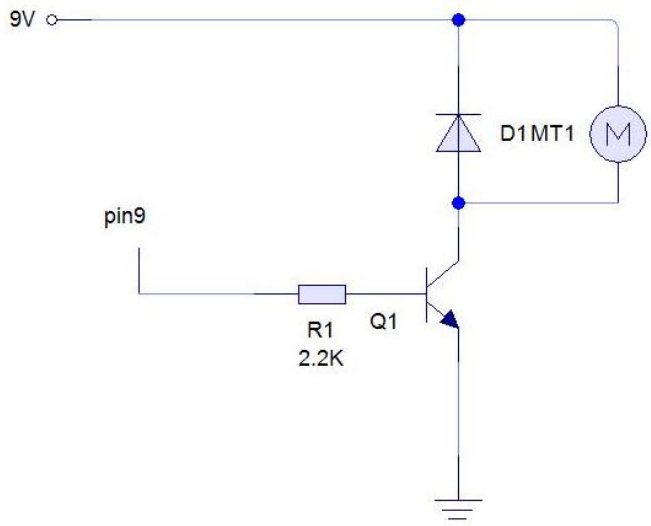


Fig. 6

Ahora nuestro código de ejemplo, como base utilizaremos el código del tema Fade led con **PWM**.

```
/*
 * ********************************************************************************************
 * ************************ Instituto Tecnologico de Mexicali,************************
 * ****************** primer concurso interno de seguidores de linea ******************
 * ********************************************************************************************
 * control de motores *
 */
// utilizamos el pin 9 para conectuar nuestro circuito que controlara el motor.
int motorPin = 9;

void setup() {
//establecemos el modo de funcionamiento de nuestro pin  a utilizar
pinMode (motorPin, OUTPUT);
}
void loop() {
  // aceleramos el motor.
  for (int fadeVal = 0 ; fadeVal <= 255; fadeVal += 5) {
    analogWrite(motorPin, fadeVal);
    delay(30);
  }
  // desaceleramos el motor.
  for (int fadeVal = 255 ; fadeVal >= 0; fadeVal -= 5) {
    analogWrite(motorPin, fadeVal);
    delay(30);
  }
}
```

Puente H

Un **Puente en H** es un circuito electrónico cuya función es controlar el giro de los motores, específicamente controla el sentido en el que el motor gira.

Generalmente son muy usados en robótica por lo tanto no podía faltar para fines de este curso. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes independientes, generalmente cuatro transistores, también pueden encontrarse circuitos pre-armados para control de motores de cierta potencia en el mercado, solo que es recomendable conocer la potencia que nuestros motores consumirán, debido a que los circuitos comúnmente no soportan mucho amperaje y tienden a sobrecalentarse, lo que provoca corto circuito.

Sin embargo factores como el peso del vehículo, la superficie en la que se moverá, el tiempo que tardara en movimiento, pueden provocar que el consumo de amperaje aumente, por esta razón debemos considerar un margen para prever algún problema de este tipo.

Se muestra la estructura básica de un puente H en la figura 7.

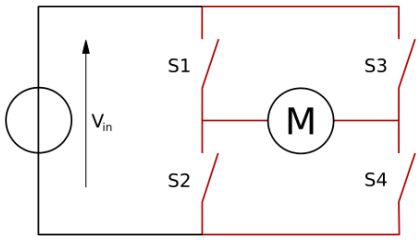


Fig. 7

Y aquí denotamos como es que el giro del motor cambia de acuerdo al flujo del voltaje.

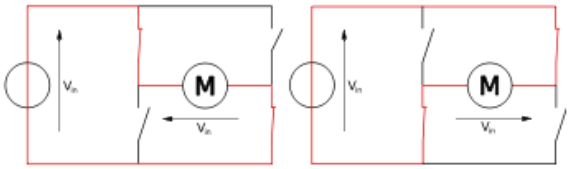


Fig. 7.1

Cuando los interruptores S1 y S4 (ver primera figuras 7 y 7.1) están cerrados (y S2 y S3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

Con la nomenclatura que estamos usando, los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque esto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4.

Como se menciona al inicio de este capítulo, estos circuitos pueden ser encontrados ya diseñados en el mercado por ejemplo el l298 que se muestra a continuación.

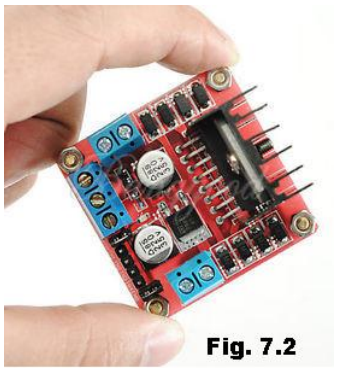


Fig. 7.2

Así como este hay varios más en el mercado de diferentes proveedores y diseños sin embargo el funcionamiento es el mismo. En adelante utilizaremos este circuito para la explicación de su funcionamiento por ser de los más comunes.

Aquí la configuración de sus pines.

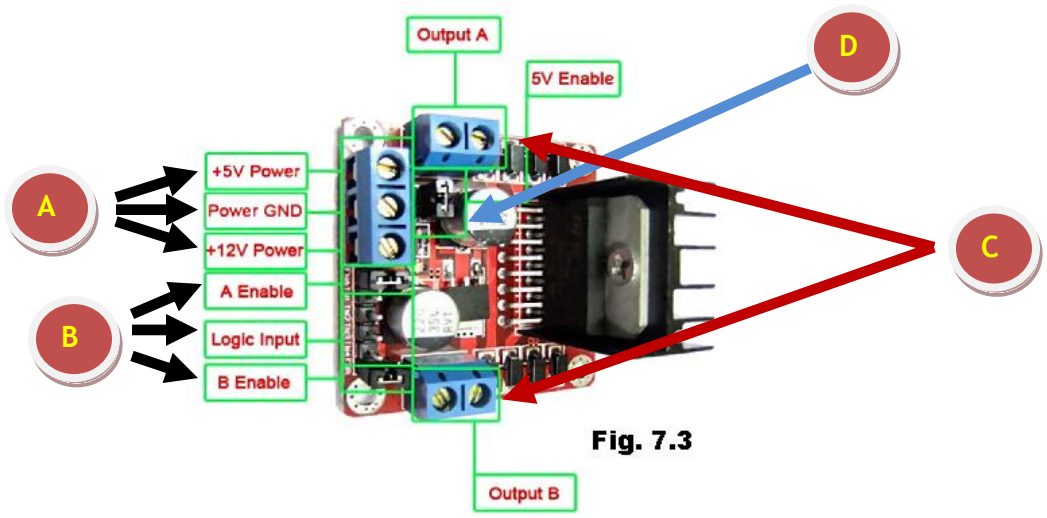


Fig. 7.3

A El circuito puede ser alimentado desde 5v hasta 12v.

B (Logic input) Las instrucciones son en binario, cada motor responde a un par de bits, y se puede habilitar el control con PWM (Enable) de manera independiente para cada motor.

C Las conexiones para cada motor (Motor izquierdo y derecho).

D Habilita el regulador interno del circuito para trabajar con una solo fuente y además utilizar el pin 5v para alimentar algo adicional sin superar los 500mA.

Se muestra un pequeño codigo para la utilizacion del puente H en el cual lo unico que hace es enviar instrucciones para girar el motor a la derecha, luego a la izquierda y finalmente se detiene, implementando funciones.

```
 * ********************************************************************************
 * ************************ Instituto Tecnologico de Mexicali,************************
 * ***************** primer concurso interno de seguidores de linea ******************
 * ********************************************************************************
 * control de motores con puente H*
 */
int PinIN1 = 7;
int PinIN2 = 6;
void setup() {
  // inicializar la comunicación serial a 9600 bits por segundo:
  Serial.begin(9600);
  // configuramos los pines como salida
  pinMode(PinIN1, OUTPUT);
  pinMode(PinIN2, OUTPUT);
}
void loop() {
  //se invocan las funciones respectivas.
  derecha();
  Serial.println("Giro a la derecha");
  delay(1000);
  izquierda();
  Serial.println("Giro a la izquierda");
  delay(1000);
  detenido();
  Serial.println("Motor Detenido");
  delay(1000);
}
```

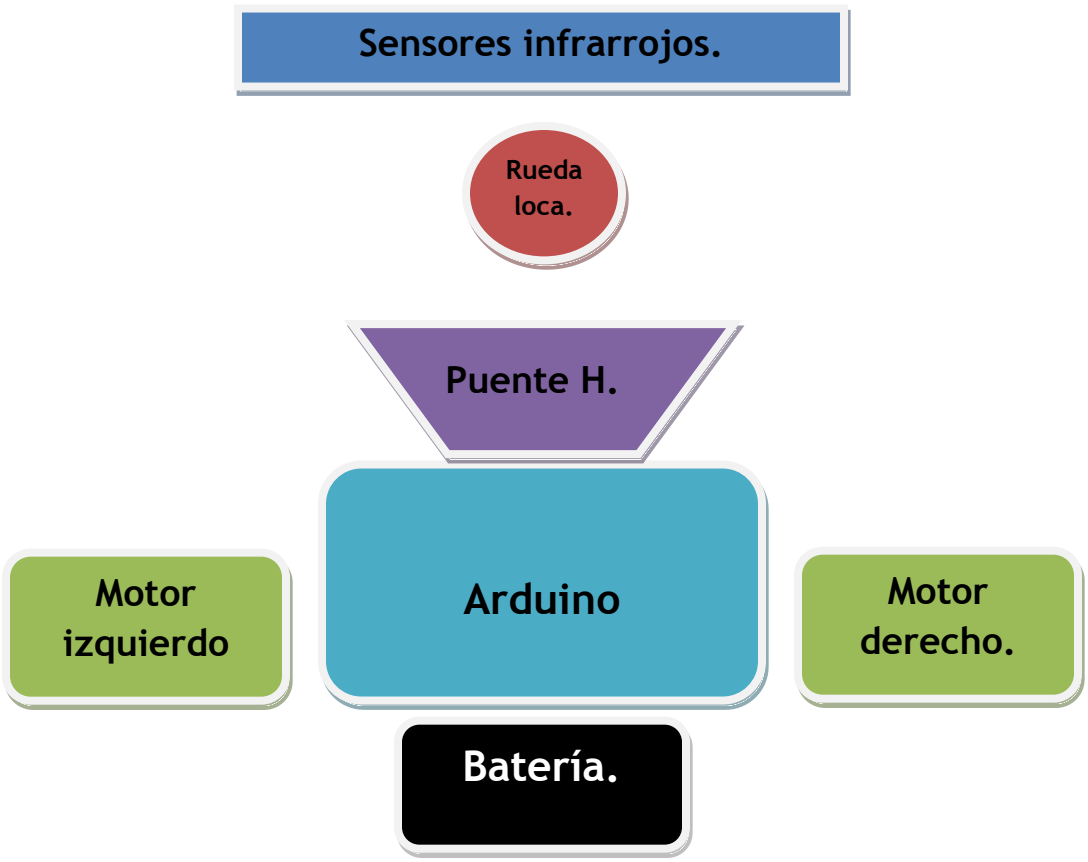
Aquí estan declaradas nuestras funciones.

```
void derecha()//funcion para realizar el giro a la derecha mandando un 1 y 0 a los pines 1 y 2 del puente h.
{
  digitalWrite (PinIN1, HIGH);
  digitalWrite (PinIN2, LOW);
}
void izquierda()//funcion para realizar el giro a la izquierda mandando un 0 y 1 a los pines 1 y 2 del puente h.
{
  digitalWrite (PinIN1, LOW);
  digitalWrite (PinIN2, HIGH);
}
void detenido()//funcion para detener el motor enviando 0's a ambos pines.
{
  digitalWrite (PinIN1, LOW);
  digitalWrite (PinIN2, LOW);
}
```

Armado del robot.

Ya en este punto se comenzara con el ensamble o unificación de los elementos con los que estará conformado nuestro robot, esta forma puede variar ya que no todos tendremos al alcance los mismos elementos sin embargo los básicos serán los mismos.

En primera instancia mostraremos un esquema general de los componentes de nuestro robot para tener la idea del ensamble y funcionamiento del mismo.



En este se pueden apreciar los elementos básicos por los que está conformado nuestro robot.

Batería; La encargada de proveer la alimentación de corriente necesaria para el funcionamiento de los circuitos y motores del robot. Se sugiere que sean dos baterías independientes una para motores y otra para el circuito controlador.

Arduino; Prácticamente la pieza principal de nuestro robot, ya que es el encargado del control del mismo así como del procesamiento de los datos enviados por los sensores.

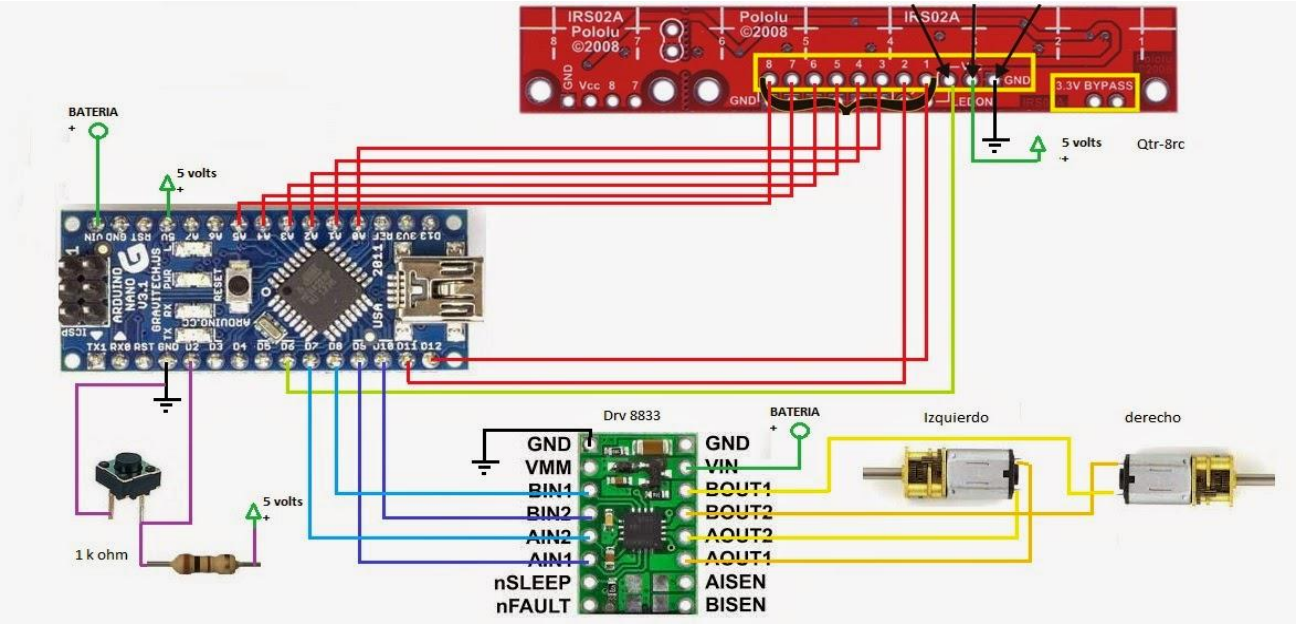
Puente H (Driver); Controlador de potencia para motores, recibe órdenes mediante código binario para controlar cosas como el giro y la velocidad de los mismos.

Sensores infrarrojos; Elementos encargados del censado y detección de la línea negra en fondo blanco o viceversa.

Rueda loca; Solo utilizada como tercer apoyo, sin embargo juega un papel muy importante ya que reduce considerablemente la fricción del robot contra la superficie.

Motores; elementos encargados de la tracción del robot, es importante considerar el tipo de ruedas para asegurar un mayor agarre.

Aquí el diagrama electrónico de conexión.



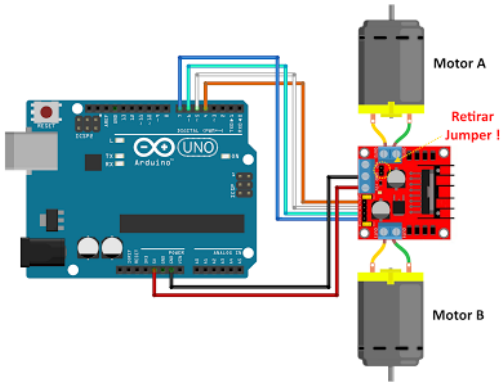
En esta figura se muestra la conexión realizada entre los diferentes elementos del robot, desde los sensores infrarrojos, que para este caso se utiliza una barra de sensores pololu de 8 sensores infrarrojos, los cuales nos aseguran una mejor precisión a la hora de censar la línea.

Como se puede apreciar estos sensores se conectan a los respectivos pines de Arduino (ya sea Arduino nano, Arduino uno o Arduino mega).

Este a su vez acorde a los datos recibidos por la barra de sensores envía la ordena al controlador de los motores (puente H) en forma de código binario, lo cual determina la dirección hacia donde girara nuestro robot y la velocidad que tendrá este desplazamiento.

Finalmente los motores reciben la orden y giran en e sentido que les sea indicado, como dato adicional tenemos que cada motor cuenta con un pequeño moto reductor (transmisión) el cual determina la relación de la que es nuestro motor, aquí debemos tener cuidado a la hora de escoger los motores, ya que dependiendo de su relación es que son para velocidad o para fuerza de tracción.

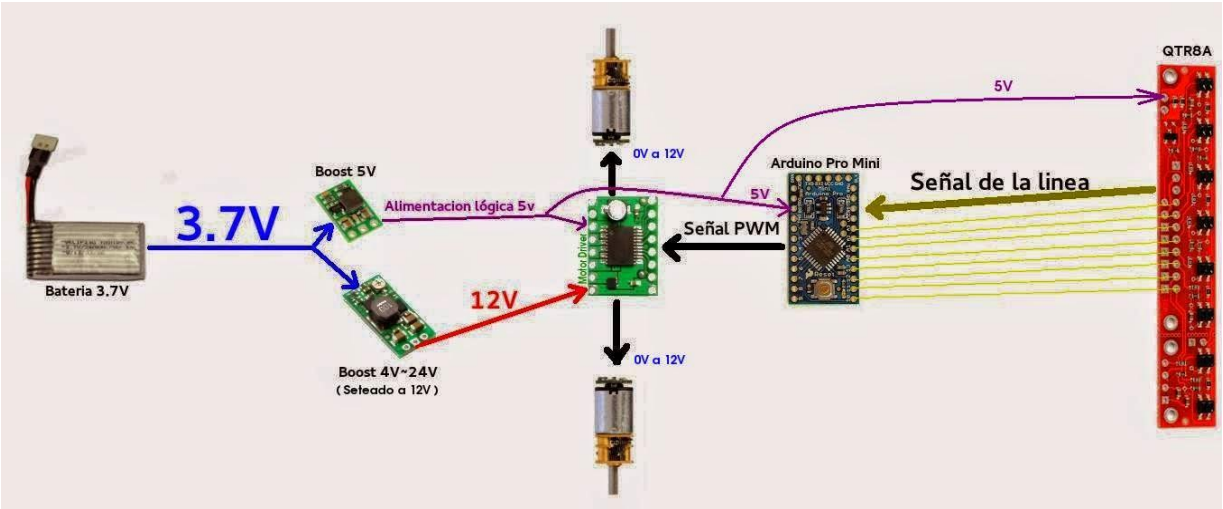
Se muestran otras conexiones posibles



Dependiendo del equipo con el que se cuente se pueden presentar distintos tipos de armado en el circuito, sin embargo el funcionamiento esperado y código del mismo es igual probablemente con diferencias mínimas.

Si se desea se puede optar por la utilización de un potenciador de batería (Boost) el cual nos ayuda para amplificar el voltaje que nuestra batería nos provee.

En la siguiente figura se puede apreciar que se están utilizando dos potenciadores de gama distinta, se utiliza uno para elevar el voltaje de 3.7 v que entrega la batería hasta 5 v, para la alimentación de los componentes y otro de 3.7 v a 12 v para la alimentación de los motores, lo que nos permite mayor potencia y facilita alcanzar velocidades mayores.

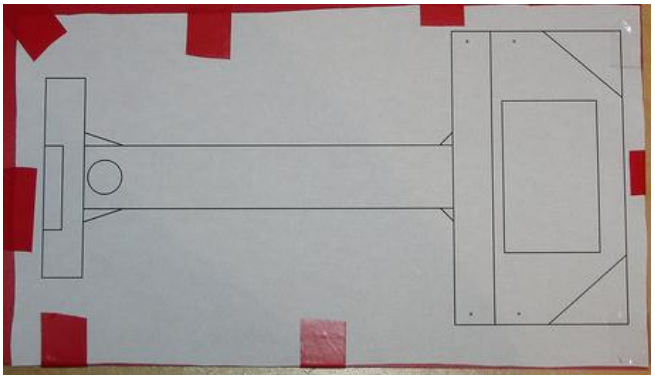


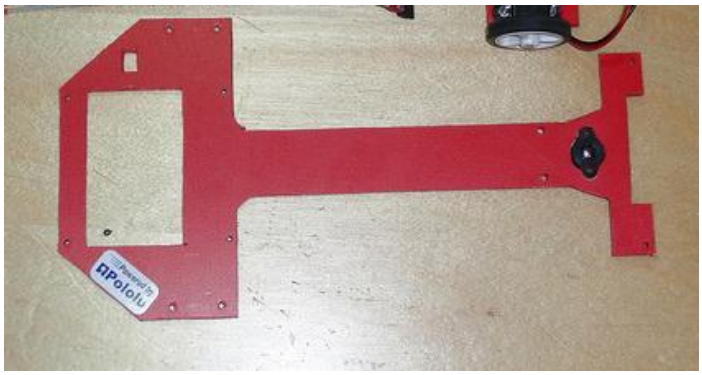
Chasis.

La selección del material del cual estará conformado el chasis del robot, es también un factor importante a considerar ya que la idea es que el peso del robot no sea excesivo, es decir se busca minimizar el peso del mismo.

Por esta razón la selección del material para el chasis, así como su diseño debe ser cuidadosa, ya que intervienen otros factores como pudiera ser el aerodinamismo del vehículo y su tamaño.

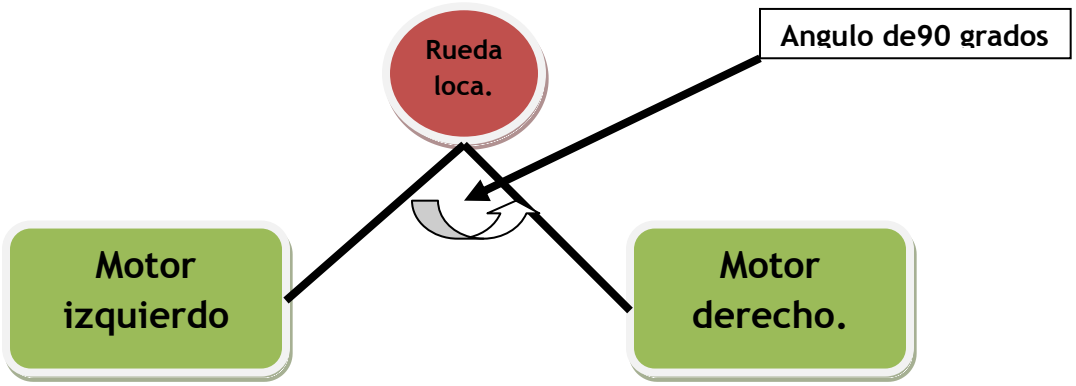
Aquí un ejemplo con la forma clásica y representativa para este tipo de vehículos, sin embargo su forma y material queda abierto a las posibilidades y gustos de cada quien.





Se deben considerar aspectos un poco más técnicos, el primero de ellos, la longitud entre la llanta loca y los sensores, se sugiere que sea de aproximadamente 2.5 centímetros debido a que estando más cerca provocaría lecturas erróneas por la inestabilidad generada por la fricción.

Otro punto importante es que además se debe generar un ángulo de 90 grados entre las dos ruedas de tracción y la rueda loca como se muestra a continuación, sin embargo estos detalles también quedan a consideración del diseñador del vehículo.



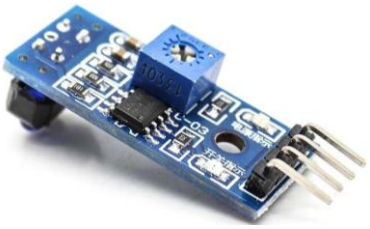
Sensores.

Como ya se mencionó anteriormente los sensores utilizados para la construcción del vehículo, son infrarrojos, estos sensores poseen multitud de aplicaciones como por ejemplo, vigilancia de objetos y personas, medida de temperaturas remotas en aplicaciones industriales, detección de fugas de calor, monitorización y detección de incendios, y diferentes aplicaciones científicas y médicas.

El sensor infrarrojo se basa en la emisión de cuerpo negro ideal, es decir, un cuerpo que absorbe y re-emite toda la radiación incidente, independientemente de la longitud de onda que sea. Ya que esta situación es ideal, se trabaja con una **aproximación** de cuerpo negro, donde la radiación incidente no sólo se absorbe, también se refleja y se transmite.

Por ser el más comúnmente utilizado el sensor cny70 será el que igualmente utilizaremos para la explicación e implementación en nuestro seguidor, este **es un sensor óptico infrarrojo**, de un rango de corto alcance (menos de 5 cm) que se utiliza para detectar colores de objetos y superficies. Contiene un emisor de radiación infrarroja -fotodiodo- y un receptor -fototransistor-. El fotodiodo emite un haz de radiación infrarroja, el fototransistor recibe ese haz de luz cuando se refleja sobre alguna superficie u objeto.

Este es el principio básico y de esta misma forma funciona toda la gama de sensores de este tipo, sin embargo no son exactamente iguales ya que la sensibilidad o potencia varían dependiendo el modelo del mismo y del circuito que los controle.



Motores.

Son quizás de los elementos más importantes de nuestro vehículo, ya que de ellos dependerá en gran medida la velocidad y precisión con la que nuestro robot se desplazara, para la selección de estos motores, conocidos en el mercado como moto reductores, se recomienda tomar en cuenta principalmente el valor de relación que tienen, este valor determina la velocidad y torque de cada motor, el hecho de hablar de relación hace referencia a la cantidad de vueltas que serán necesarias por parte del motor, para lograr que la rueda o llanta logre una completa, en la siguiente figura se puede apreciar de forma un poco más visual este hecho.

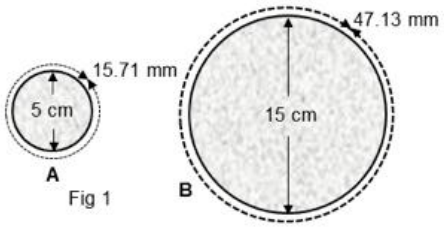


Fig 1

Para ello se muestran dos figuras, las cuales tomaremos respectivamente como el motor y la llanta del vehiculo, aunque realmente un motorreductor esta conformado por una gran cantidad de engranes, esto varia dependiendo la minima velocidad que se desee alcanzar, o la maxima en su defecto, permitiendo con esto motorreductores con relaciones hasta de 1000 : 1, significando que por cada mil vueltas del motor, la llanta o el engrane final cumplira solo una.

Aqui se ejemplifica dicha situacion conciderando a la circunferencia **A** como el motor y la circunferencia **B** como la etapa final de traccion, se puede apreciar por los datos que se tienen. Que el diametro correspondiente al motor es una tercera parte de lo que seria la llanta. Es decir esto nos da una relacion de 3 : 1, por lo que podemos afirmar lo antes mencionado, tambien de acuerdo a esta relacion sabemos que por cada 3 vueltas del motor se tendra una de la llanta, no obstante si se desea aumentar o disminuir esta velocidad, se deberan agregar engranes intermedios al motor y la llanta, logrando con esto que se tenga un incremento en la velocidad con la misma tencion de corriente o que se tenga una velocidad minima con una tencion que se encuentre dentro de un rango de operacion aceptable.

Los rangos de operacion en cuanto a tenciones e intensidades, se especifican en los detalles tecnicos de cada motor, entre estos datos se pueden encontrar las dimensiones, la realcion de cada del mismo, la tension de voltaje minima y maxima soportada entre otros.

Como dato importante toda la gama de motores de este tipo tienen el mismo diseño y las mismas dimensiones, exeptuando el mayor de ellos cuya relacion es de 1000 : 1, asi que si fuera necesarrio cambiar un motor por otro de mayor o menor relacion, esto no debe suponer ningun problema en cuanto a cuestiones de siseño del vehiculo.

Generalmente esto diminutos motorreductiores estan destinado para uso a 6V, aunque en general, este tipo de motores pueden funcionar a tenciones por encima y por debajo de ese voltajenominal, por lo que su rango de operacion va desde los 3 V hasta los 9 V (la rotacion de los motores comienza desde una tencion minima de 0,5 V) valores menores de este rango pueden no ser apropiados para el funcionamiento adecuado y mayores niveles de voltraje pueden afectar en forma negativa la vida util del motor.

(continuar…)

Baterías.

La fuente de energia del vehiculo, las baterias o bateria seran las encargadas de brindar el voltaje necesario para alimentar a todos los componentes del vehiculo, desde los sensores hasta el controlador de los motores, en base a las pruebas realizadas anteriormente se obtienen datos interesantes en cuanto a este punto, es decir se ha observaio que es altamente recomendable la utilizacion de cuando menos dos baterias indepoendientes, una sera la encargada de alimentar exclusivamente el circuito controlador, arduino en este caso y a los sensores del vehiculo, y de forma individual otra alimentara el controlador de los motores.

Se ha observado esta necesidad por la siguiente razon, dependiendo del disñeo del vehiculo este puede llegar a alacanzar un peso mayor a lo esperado, como consecuencia esto genera un consumo de corriente elevado cada vez que el vehiculo intente desplazarse, particularmente cada vez que los motores arrancan al estar completamente detenidos es cuando se presenta un consumo mayor, si pudieramos ver esto reflejado en una grafica la representacion seri similar a la siguinete.

“Imagen con caida o variacion de tenciones.”

Este fenomeno es conocido comunmente como caidas o picos de voltaje respectivamente, por lo tanto cada una de estas variaciones significa una diferencia considerale Este consumo deriva en un consumo acelerado de la carga de la bateria, particularmetne este genera que cada vez que los motores intenten desplazar el vehiculo se realice un consuo digamos excevio de corriente po

(continuar…)

Controlador.