



Tecnológico de Monterrey

Alfonso Morales Gutiérrez - A01329634

Amilcar Ozuna Cruz - A01350665

Luis Daniel Pérez Michel - A01206071

Laboratorio Instrumentación Mecatrónica

Proyecto 2 Parcial

Carro a control remoto via Bluetooth

23 Marzo 2018

Introducción

En este punto del semestre hemos hecho diversas prácticas en las que descubrimos nuevos componentes electrónicos utilizados frecuentemente en un laboratorio de instrumentación mecatrónica. Para este proyecto retomamos algunos de esos elementos y en conjunto de nuevos elementos y conocimientos trataremos de armar un carro a contro remoto via bluetooth el cual mediante un código pre programado podamos darle instrucciones mediante un smartphone.

Marco Teórico

Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.).

MDF

Un tablero DM es un aglomerado elaborado con fibras de madera (que previamente se han desfibrado y eliminado la lignina que poseían) aglutinadas con resinas sintéticas mediante fuerte presión y calor, en seco, hasta alcanzar una densidad media.

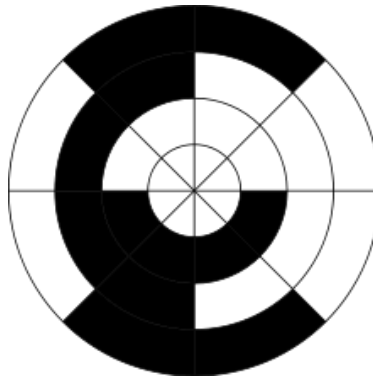
También se le llama DM (densidad media), mdf (sigla en inglés de *Medium Density Fibreboard*), Trupan (nombre común en Chile derivado de la marca comercial) o Fibrofácil (denominado así en Argentina, también derivado de una marca).

Presenta una estructura uniforme y homogénea y una textura fina que permite que sus caras y cantos tengan un acabado perfecto.



Encoder

Un encoder es un dispositivo de detección que proporciona una respuesta. Los Encoders convierten el movimiento en una señal eléctrica que puede ser leída por algún tipo de dispositivo de control en un sistema de control de movimiento, tal como un mostrador o PLC. El encoder envía una señal de respuesta que puede ser utilizado para determinar la posición, contar, velocidad o dirección. Un dispositivo de control puede usar esta información para enviar un comando para una función particular.



Módulo Bluetooth HC05

Descripción:

El módulo de bluetooth HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo Maestro-Esclavo, quiere decir que además de recibir conexiones desde una PC o tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto nos permite por ejemplo, conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos:

- Solicitudes
- Dispositivos informáticos y periféricos
- Receptor GPS
- Control Industrial
- Proyectos MCU

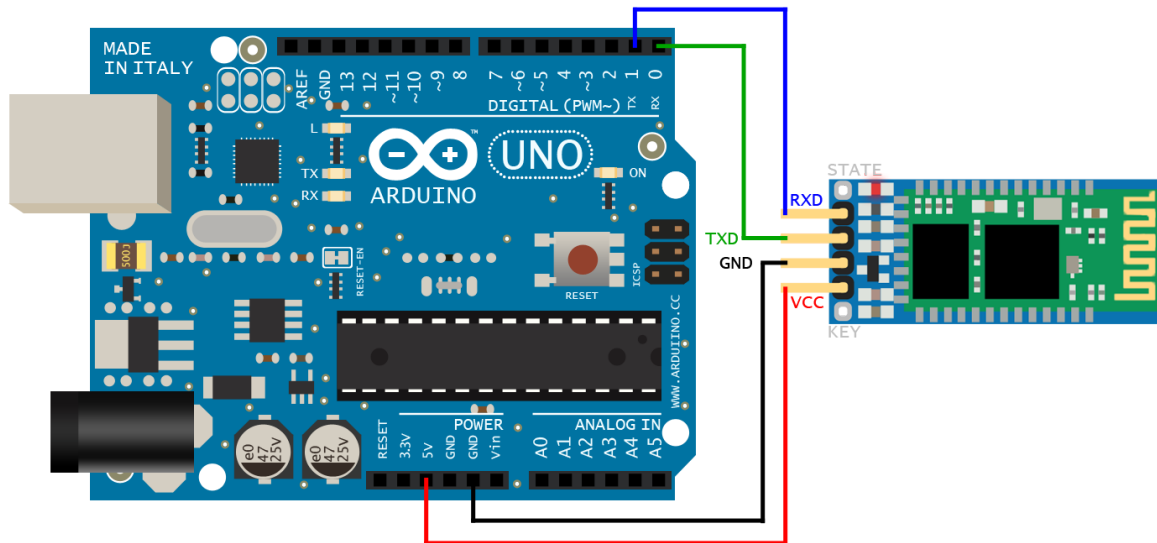
Especificaciones:

- Compatible con el protocolo Bluetooth V2.0.
- Voltaje de alimentación: 3.3VDC – 6VDC.
- Voltaje de operación: 3.3VDC.
- Baud rate ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Baud rate por defecto: 9600
- Tamaño: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
- Corriente de operación: < 40 mA
- Corriente modo sleep: < 1mA

La tarjeta incluye un adaptador con 6 pines de fácil acceso para uso en protoboard.

Los pines de la board correspondientes son:

- EN
- VCC
- GND
- TX
- RX
- STATE



Sensor infrarrojo para Encoder

Características:

Los codificadores ópticos o encoders incrementales constan, en su forma más simple, de un disco transparente con una serie de marcas opacas colocadas radialmente y equidistantes entre sí, de un sistema de iluminación en el que la luz es colimada de forma adecuada, y de un elemento fotorreceptor. El eje cuya posición se quiere medir va acoplado al disco transparente. Con esta disposición, a medida que el eje gire se irán generando pulsos en el receptor cada vez que la luz atraviese cada marca, y llevando una cuenta de estos pulsos es posible conocer la posición del eje.



Motorreductor

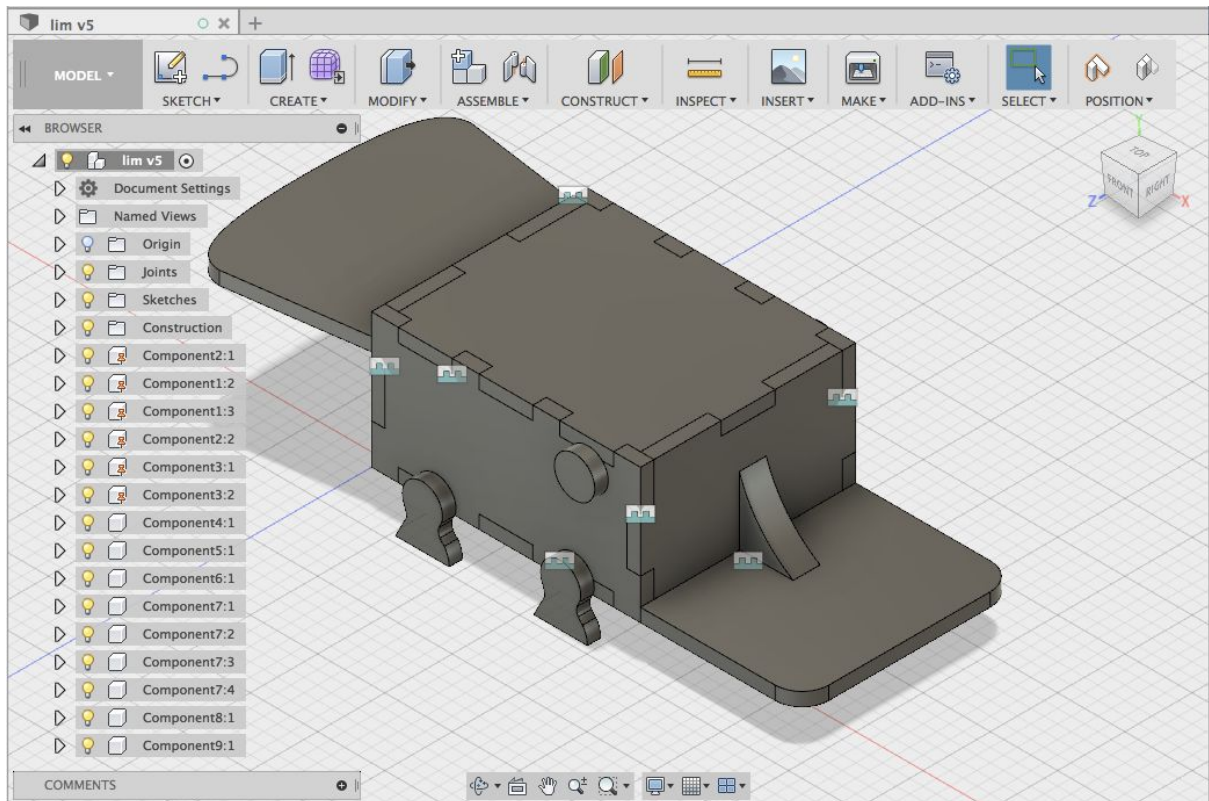
Un motorreductor es un reductor de velocidad con un motor directamente conectado. Un motorreductor no utiliza un adaptador campana y acople para montar el motor, el motor es de hecho parte del reductor de velocidad. Una primera ventaja del motorreductor comparado con otra combinación motor y reductor, es la longitud. Moto reductores son compactos en tamaño.



Desarrollo experimental

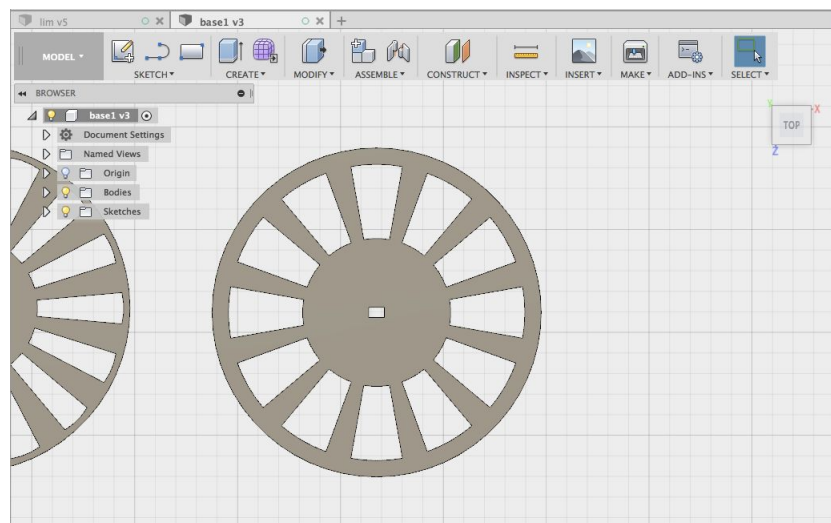
El presente proyecto fue un reto integrador que nos exigió plasmar en forma de un carrito controlado vía celular la mayoría de lo que hemos visto hasta ahora, siendo el módulo Bluetooth HC-05 el componente más importante para la realización del modelo además de la utilización de un puente H.

- Como primer paso se realizaron diversos bocetos para la estructura y decidimos optar por una opción fácil y divertida de hacer, por lo que realizamos un carrito con forma de perry el ornitorrinco.
- Una vez definida la forma de nuestra estructura decidimos dibujarla en computadora para posteriormente cortarla en la máquina laser del DICI.



Modelado en 3D de la carcasa en forma del personaje de Disney, Perry el ornitorrinco, realizado con el software Autodesk Fusion 360.

- Se cortaron las piezas en MDF de 3mm y se armó la estructura.
- El encoder se cortó al tamaño de una llanta (8 centímetros de diámetro) por lo que decidimos utilizarlo para ambas funciones.



Diseño del encoder con el software Autodesk Fusion 360.

- Una vez con el encoder listo decidimos realizar un circuito sencillo para recibir la información arrojada por el encoder. Para ello se siguió el siguiente diagrama.

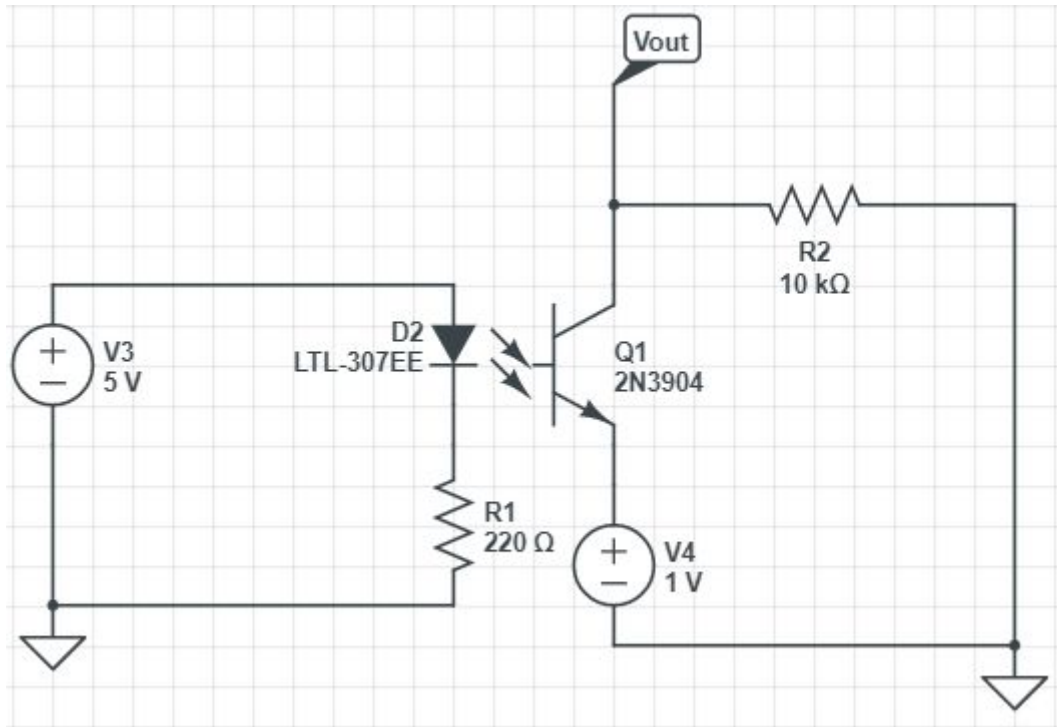


Diagrama de conexión para el Encoder

- Ya que cableamos el encoder nos dimos a la tarea de obtener la distancia que recorre nuestro encoder al avanzar. Para ello se realizaron los siguientes cálculos.

Primero determinamos la separación en grados entre cada hueco del encoder

$$n = \frac{360}{12} = 30$$

Buscamos el desplazamiento en radianes

$$2\pi \rightarrow 360$$

$$x \rightarrow 30$$

$$x = \frac{(30)(2\pi)}{360} = \frac{1}{6}\pi = 0.5235987756$$

Y obtenemos la longitud de desplazamiento para un hueco del encoder

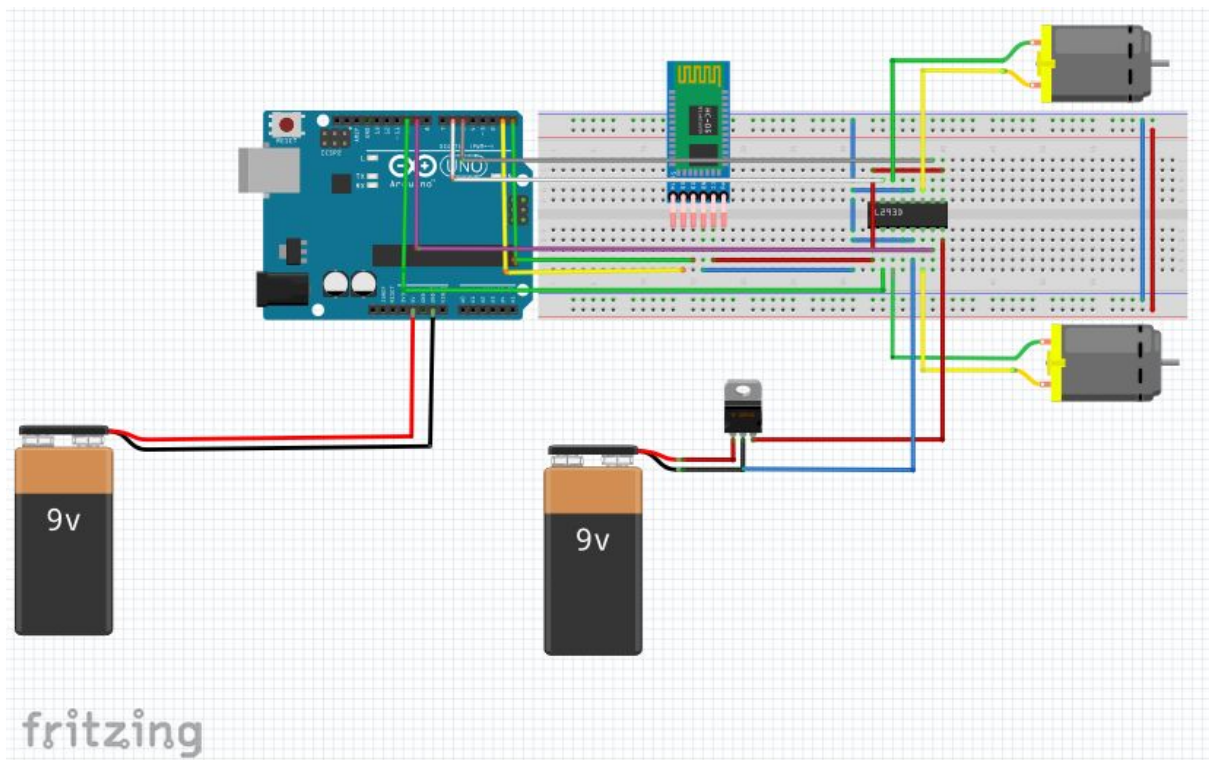
$$L = (0.5235987756)\left(\frac{8}{2}\right)(\# \text{ de veces que pasa por el hueco del encoder})$$

- Con la expresión anterior pudimos hacer que el carrito se moviera hasta cierta distancia deseada.

- Se realizó el código de Arduino para que cumpliera con las siguientes funciones.

Algoritmo

1. Crear un canal de comunicación mediante los pines TX y RX.
2. Crear una serie de instrucciones para los cuales el carrito opere de forma distinta, es decir:
 - a. Definir e implementar un comando para que los motores giren hacia adelante a una determinada distancia.
 - b. Definir e implementar un comando para que los motores giren hacia atrás a una determinada distancia..
 - c. Definir e implementar un comando para que los motores hagan girar 90 grados hacia la derecha el carrito sobre su propio eje.
 - d. Definir e implementar un comando para que los motores hagan girar 90 grados hacia la izquierda el carrito sobre su propio eje.
 - e. Definir e implementar un comando para que los motores se detengan.
3. Leer una cadena de comandos designados a letras específicas para cada operación.
4. Almacenar la cadena de instrucciones que el usuario decida
5. Leer cada instrucción individualmente y ejecutar.
6. Hacer lo mismo que en el paso 5 hasta que se haya realizado la última instrucción mandada por el usuario



Esquema del circuito que controla el carrito

Código de Arduino utilizado, y explicación:

```
#include <SoftwareSerial.h> // Incluimos la libreria SoftwareSerial

int gate, j, i; //Variables utilizadas internamente por el programa.
String comando="";
char x;
bool state, estadoult;
int izqA = 5;
int izqB = 6;
int derA = 9;
int derB = 10;
int encoderA = 2;
int encoderALast = LOW;
int encoderPos = 0;
int n= LOW;
int a=0;
int estado=1;

void setup()
{
  Serial.begin(9600); // Inicializamos el puerto serie
  pinMode(derA, OUTPUT); //Declaramos los pines como entrada o salida de datos
  pinMode(derB, OUTPUT);
  pinMode(izqA, OUTPUT);
  pinMode(izqB, OUTPUT);
  pinMode(encoderA, INPUT);
}
```

Figura 1: Inclusión de bibliotecas, declaración de variables, velocidad del puerto serial, y especificación del modo de trabajo de los pines.

En la figura 1 podemos observar todas las variables que se utilizaron para el código además de la especificación del modo de trabajo para cada una de los pines. Al haber introducido la serie de instrucciones para definir los pines en el void setup () nos aseguramos de que sólo se iba a realizar una vez dicha acción.

```
void loop()
{
  gate=0;
  comando="";
  estadoult=LOW;

  do
  {
    if(Serial.available()>0) //Si el arduino detecta un envío de señal
    {
      x=Serial.read();      // Se almacena la señal recibida y se agrega al string de comandos
      delay(10);
      Serial.print(x);
      comando.concat(x);
    }

    if(x=='S' || x=='s')
    {
      gate=1;              //cuando se envíe la señal de start ("S"), se sale de la recepción de datos.
    }
  }
  while (gate!=1);
```

Figura 2: Loop del programa y lectura de datos

El código esencialmente es una compilación de diversos códigos que ya habíamos realizado. Por ejemplo utilizamos parte del código de la práctica 12 en dónde utilizamos por primera vez el puente H (HL293), sólo que para esta ocasión no modificamos las velocidades de los motores mediante los pulsos PWM.

Por otra parte agregamos a nuestro código la posibilidad de recibir una cadena de comandos para ejecutar una vez que se mandaba el carácter "S" o "s".

En la figura 2 se puede observar que se utilizan dos sentencias principales que son el if y el do while. El do while nos permite introducir una cadena de caracteres mientras no se cumpla la condición del gate igual a 1, cabe destacar que todo esto es realizado mediante comunicación Bluetooth® que funciona gracias al ciclo if que detecta los envíos de señal mediante un Smartphone. Finalmente todas las instrucciones son guardadas en la cadena de caracteres llamada comando.

Proyecto - 2 Parcial

```
for(j=0; j<comando.length(); j++) //Ciclo que asegura leer cada una de las instrucciones almacenadas.
{
    switch(comando.charAt(j)) //Se pasa el string por cada uno de sus caracteres con la funcion char at y el indice j.
    {

        case 'a': //Movimiento hacia adelante
        do{
            n = digitalRead(encoderA);
            if ((a>=0) && (a<=100)) // Restricción de movimiento a 100 cm
            {
                digitalWrite(derB, LOW);
                digitalWrite(izqB, LOW);
                digitalWrite(derA, HIGH);
                digitalWrite(izqA, HIGH);

                if ((encoderALast == LOW) && (n == HIGH)) // Lectura del desplazamiento realizado por las llantas
                {
                    encoderPos++;
                    a=encoderPos*0.52359877559*(8/2);
                    Serial.println(encoderPos);
                    Serial.println(encoderPos*30);
                    Serial.println((encoderPos*0.52359877559)*(8/2));
                }

                encoderALast = n;
            }

        }

    }
    else
    {
        digitalWrite(derB, LOW);
        digitalWrite(izqB, LOW);
        digitalWrite(derA, LOW);
        digitalWrite(izqA, LOW);
        estado=0;

    }
}while(estado!=0);
if (encoderPos!=100 && a!=0) //Reinicio de variables una vez que se alcanza la distancia de 100 cm
{
    a=0;
    encoderPos=0;
}
delay(2000);
estado=1;
break;
```

Figura 3: Lectura individual de las instrucciones y comando avanzar

Una vez que se ha llenado la cadena de caracteres el ciclo for nos permite realizar una lectura individual de cada carácter para ejecutar la instrucción ligada a dicho carácter seleccionado.

```
case 'r': //Movimiento hacia atras
digitalWrite(derA,LOW);
digitalWrite(izqA,LOW);
digitalWrite(derB,HIGH);
digitalWrite(izqB,HIGH);
delay(2000); //Tiempo de cambio de una instrucción a otra
break;

case 'd': //Movimiento hacia atras
digitalWrite(derA,LOW);
digitalWrite(izqB,LOW);
digitalWrite(izqA,HIGH);
digitalWrite(derB,HIGH);
delay(1000); //Tiempo de cambio de una instrucción a otra
break;

case 'i': //Movimiento hacia la izquierda
digitalWrite(derB,LOW);
digitalWrite(izqA,LOW);
digitalWrite(derA,HIGH);
digitalWrite(izqB,HIGH);
delay(1000); //Tiempo de cambio de una instrucción a otra
break;

case 'p': //Detener ambas ruedas
digitalWrite(derB,LOW);
digitalWrite(izqA,LOW);
digitalWrite(derA,LOW);
digitalWrite(izqB,LOW);
delay(1000); //Tiempo de cambio de una instrucción a otra
break;

}
}
}
```

Figura 4: “Menú” de Instrucciones a ejecutar

Para esto se utilizó la función de switch para mostrar todas las opciones disponibles de nuestro programa. Teniendo una estructura parecida la que se muestra a continuación:

switch (comando(lectura del n caracter))

- Caso de 'a':
 - Mueve el carrito 100 cm hacia adelante.
 - (Activa Motores derecho e izquierdo con giro positivo mientras realiza una lectura del desplazamiento que se compara con la distancia establecida por el usuario)
- Caso de 'r':

- Mueve el carrito hacia atrás.
 - (Activa Motores derecho e izquierdo con giro negativo)
- Caso de 'i':
 - Mueve el carrito hacia la izquierda.
 - (Activa Motor derecho con giro positivo e izquierdo con giro negativo)
- Caso de 'd':
 - Mueve el carrito hacia la derecha.
 - (Activa Motor derecho con giro negativo e izquierdo con giro positivo)
- Caso de 'p':
 - Detiene el carrito
 - (Desactiva ambos motores)

El programa se encargó de leer cada instrucción y ejecutar una a la vez.

A continuación se presentan algunas fotos del carrito y los circuitos utilizados:

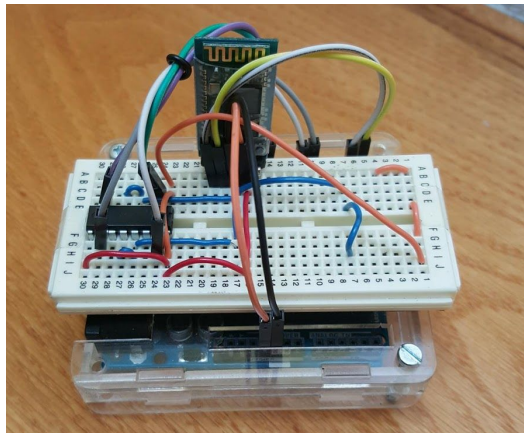


Figura 5: Circuito compacto montado al arduino

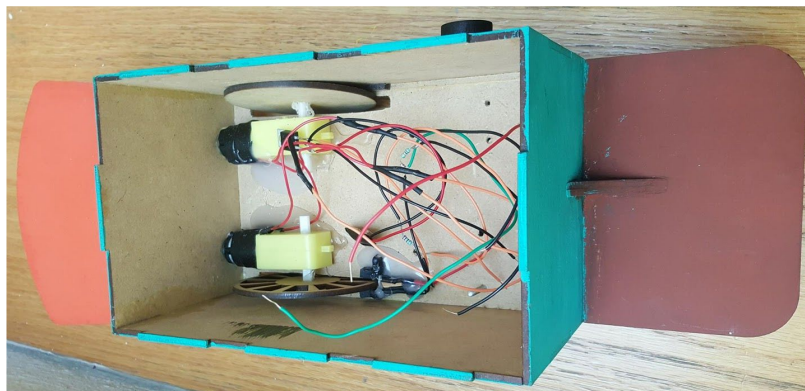


Figura 6: Encoder y motores instalados en la estructura

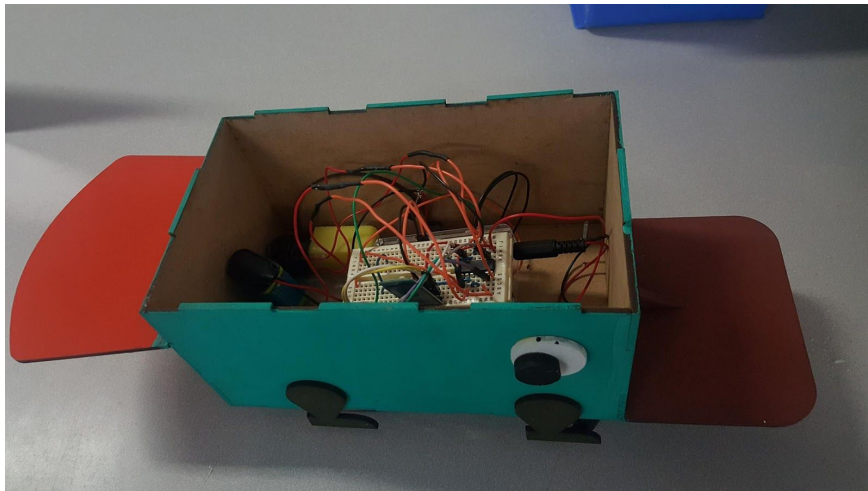


Figura 7: Circuito completo montado en la estructura

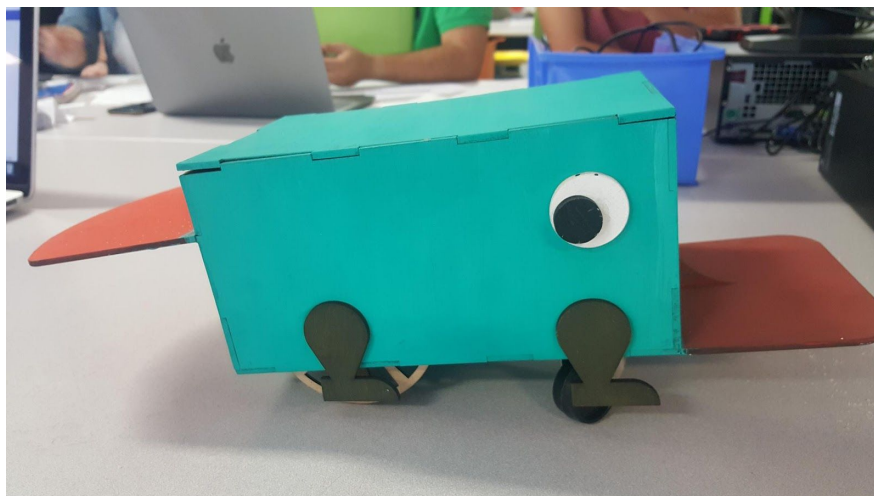


Figura 8: Producto terminado- Carrito con forma de "Perry el Ornitorrinco"

En el siguiente link puede encontrar un breve video que muestra el funcionamiento del carrito controlado por bluetooth:

<https://www.youtube.com/watch?v=Qvkp2GofpBQ&feature=youtu.be>

Análisis de resultados y conclusión.

Para el desarrollo de este proyecto se presentaron diversas complicaciones, algunas más difíciles de resolver que otras, en primer lugar lograr que con ayuda del codificador y el encoder, que a su vez funcionaba como llanta, se avanzara una determinada distancia de acuerdo al diámetro del encoder, una vez resuelto esto surgieron pequeños problemas en el momento de programar en lenguaje de Arduino las acciones que debería de realizar el dispositivo al ingresarse comandos específicos, por ejemplo cuando se introducía el comando para que avanzara cierta distancia y se deseaba introducirlo de nuevo no era posible que se moviera de nuevo, ya que ciertas variables que limitaban el movimiento para la primera

introducción del comando necesitaban ser borradas (reiniciadas a cero), lo cual suena sencillo pero que al estar hablando de una función switch case se complicó un poco encontrar la forma de hacerlo correctamente, sin embargo al final fue posible solucionar dichas problemáticas y obtener el producto final que cumpliera con los requisitos propuestos. Fue de gran utilidad realizar este proyecto ya que aprendimos a manipular una cadena de caracteres con funciones que no habíamos utilizado, y que sin importar el orden en que fueran introducidos los comandos, el software, en conjunto con el microcontrolador reconocieran que era lo que se debía de realizar.

Bibliografías

¿Qué es Arduino? ~ Arduino.cl. (n.d.). Retrieved March 24, 2018, from <http://arduino.cl/que-es-arduino/>

¿Qué es un Encoder? (n.d.). Retrieved March 24, 2018, from <http://encoder.com/blog/encoder-basics/que-es-un-encoder/>

Tablero de fibra de densidad media. (2018, March 16). Retrieved March 24, 2018, from https://es.wikipedia.org/wiki/Tablero_de_fibra_de_densidad_media

HC-05 Módulo Bluetooth Serial. (n.d.). Retrieved March 24, 2018, from <http://teslabem.com/modulo-bluetooth-serial-hc-05.html>

¿Qué es un Motorreductor? (n.d.). Retrieved March 24, 2018, from <https://www.energiacontrolada.com/fag/Que-es-un-motorreductor>