Amilcar Ozuna Cruz - A01350665
Alfonso Morales Gutiérrez - A01329634
Luis Daniel Pérez Michel - A01206071

Laboratorio de instrumentación mecatrónica

Práctica final

Proyecto integrador:R2-D2

Introducción

En esta práctica final quisimos como buen equipo juntar un poco de lo mejor que aprendimos a lo largo de este semestre en esta materia, mezclar los conocimientos obtenidos con algo que nos gustara y poder aplicarlo a la vida real. En este caso todos somos fans de la franquicia de Star Wars y se nos hizo buena idea revivir a uno de los personajes más entrañables de esta historia, R2D2. Todos conocemos a este pequeño amigo que es más que una navaja suiza, sobre todo en medio de las batallas contra el imperio, por eso es que quisimos buscar aplicaciones electrónicas que pudieran parecerse a lo mucho que realiza este robot. Entre ellas está el poder moverse en cualquier ángulo y a una gran velocidad. Para hacer esto posible utilizamos arduinos, sensores, ruedas y un poco de creatividad. Ojalá nuestro trabajo refleje lo que aprendimos en este curso.

Marco teórico

Arduino UNO

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

PWM

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

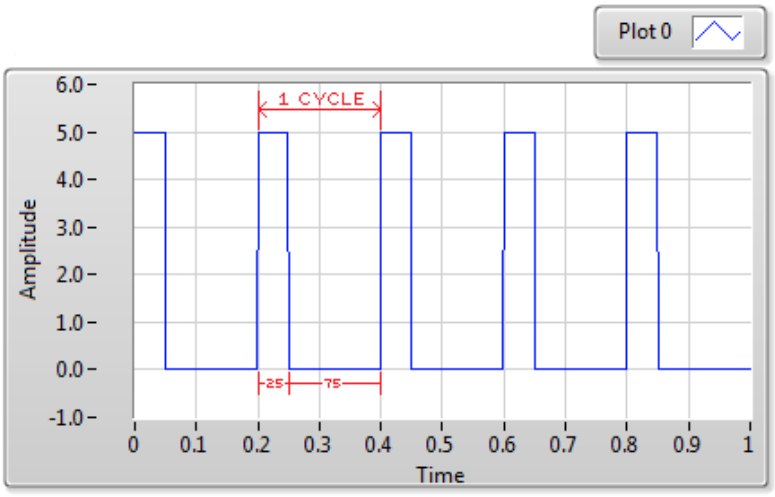
El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = T$$

D es el ciclo de trabajo

es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función



Sensor de color TCS 3220

El sensor de color TCS3200 es un convertidor de luz a frecuencia que combina fotodiodos de silicio reconfigurables y una corriente de frecuencia en un solo circuito integrado. La salida es una onda cuadrada (ciclo de trabajo 50%) con una frecuencia directamente proporcional a la intensidad de luz. Las entradas y salidas digitales permiten una interfaz directa con un microcontrolador u otro conjunto de circuitos lógicos, por esta razón el sensor TCS3200 es ideal para líneas de producción, domótica, robótica, etc.

Color

El color es la impresión producida por un tono de luz en los órganos visuales, o más exactamente, es una percepción visual que se genera en el cerebro de los humanos y otros animales al interpretar las señales nerviosas que le envían los fotorreceptores en la retina del ojo, que a su vez interpretan y distinguen las distintas longitudes de onda que captan de la parte visible del espectro electromagnético.

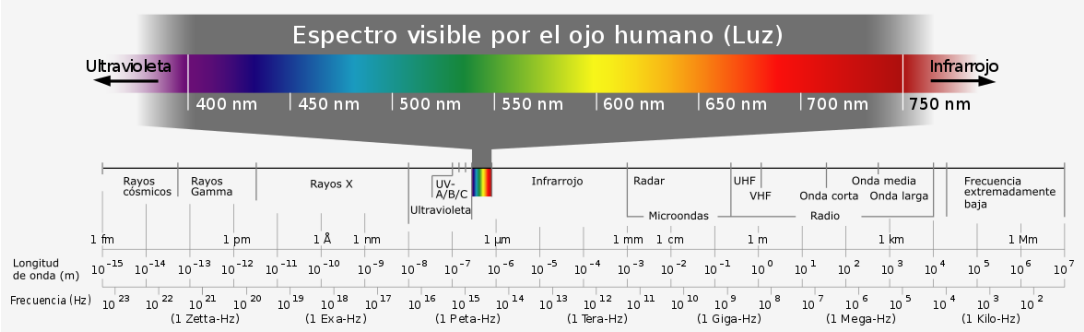
Todo cuerpo iluminado absorbe una parte de las ondas electromagnéticas y refleja las restantes. Las ondas reflejadas son captadas por el ojo e

interpretadas en el cerebro como distintos colores según las longitudes de ondas correspondientes.

Luz visible

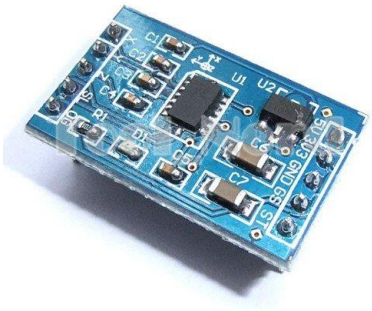
Se llama espectro visible a la región del espectro electromagnético que el ojo humano es capaz de percibir. A la radiación electromagnética en este rango de longitudes de onda se le llama luz visible o simplemente luz. No hay límites exactos en el espectro visible: el ojo humano típico responderá a longitudes de onda de 390 a 750 nm, aunque algunas personas pueden ser capaces de percibir longitudes de onda desde 380 hasta 780 nm. Los arcoíris son un ejemplo de refracción del espectro visible. El espectro visible:

Se llama a la región del que el humano es capaz de percibir. A la radiación electromagnética en este rango de longitudes de onda se le llama o simplemente luz. No hay límites exactos en el espectro visible: el ojo humano típico responderá a de 390 a 750 , aunque algunas personas pueden ser capaces de percibir longitudes de onda desde 380 hasta 780 nm. Los son un ejemplo de refracción del espectro visible. El espectro visible:



Acelerómetro

Este sensor (MMA7361) es un acelerómetro analógico de 3 ejes (x,y,z). El nivel de las medidas del acelerómetro, nos permite medir la aceleración, o la inclinación de una plataforma con respecto al eje terrestre



Sensor ultrasonico

Como su nombre lo indica, los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.

Sonido

El sonido (del latín *sonĭtus*, por analogía prosódica con *ruido, chillido, rugido*, etc.), en física, es cualquier fenómeno que involucre la propagación de ondas mecánicas (sean audibles o no), generalmente a través de un fluido (u otro medio elástico) que esté generando el movimiento vibratorio de un cuerpo.

El sonido humanamente audible consiste en ondas sonoras y ondas acústicas que se producen cuando las oscilaciones de la presión del aire, son convertidas en ondas mecánicas en el oído humano y percibidas por el cerebro. La propagación del sonido es similar en los fluidos, donde el sonido toma la forma de fluctuaciones de presión. En los cuerpos sólidos la propagación del sonido involucra variaciones del estado tensional del medio.

Velocidad del sonido

La velocidad del sonido es la dinámica de propagación de las ondas sonoras. En la atmósfera terrestre es de 343,2 m/s (a 20 °C de temperatura, con 50 % de humedad y a nivel del mar). La velocidad del sonido varía en función del medio en el que se transmite. Dado que la velocidad del sonido varía según el medio, se utiliza el número Mach 1 para indicarla. Así un cuerpo que se mueve en el aire a Mach 2 avanza a dos veces la velocidad del sonido, independientemente de la presión del aire o su temperatura.

La velocidad o dinámica de propagación de la onda sonora depende de las características del medio en el que se realiza dicha propagación y no de las características de la onda o de la fuerza que la genera. Su propagación en un medio puede servir para estudiar algunas propiedades de dicho medio de transmisión.

Condición if

Los condicionales if-else, son una estructura de control, que nos permiten tomar cierta decisión al interior de nuestro algoritmo, es decir, nos permiten determinar que acciones tomar dada o no cierta condición, por ejemplo determinar si la contraseña ingresada por el usuario es válida o no y de acuerdo a esto darle acceso al sistema o mostrar un mensaje de error.
La sintaxis de un condicional if-else, es en principio similar a la del condicional if, pero adicionando una nueva "estructura" que es el else, el cual indica la acción o conjunto de acciones a llevar a cabo, en caso de que la condición del if no se cumpla. Cabe resaltar que el else siempre se pone inmediatamente después del if, en caso de ser necesario, el else es incapaz de funcionar por sí solo, siempre debe ir acompañado por un if.

Puente H

El puente H o puente en H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder.
Los puentes H ya vienen hechos en algunos circuitos integrados, pero también se pueden construir a partir de componentes discretos.

Funciones de arduino utilizadas

Librería math.h

Las funciones trigonométricas y exponenciales de Arduino usan la biblioteca avr-libc. La biblioteca incluye una gran cantidad de útiles funciones matemáticas para manipular números de coma flotante.

Librería SoftwareSerial.h

El hardware Arduino tiene soporte integrado para comunicación serial en los pines 0 y 1 (que también va a la computadora a través de la conexión USB). El soporte serial nativo ocurre a través de una pieza de hardware (integrada en el chip) llamada UART. Este hardware permite que el chip Atmega reciba

comunicación en serie incluso mientras se trabaja en otras tareas, siempre que haya espacio en el buffer serie de 64 bytes.

Función Serial.begin

Establece la velocidad de datos en bits por segundo (baudios) para la transmisión de datos en serie. Para comunicarse con la computadora, se pueden usar cualquiera de estos valores: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200.

Función Serial.print

Imprime datos en el puerto serie como texto ASCII legible por humanos. Este comando puede tomar muchas formas.

Función serial.write

Escribe datos binarios en el puerto serie. Esta información se envía como un byte o serie de bytes; para enviar los caracteres que representan los dígitos de un número, use la función print ().
Sintaxis:Serial.write(val)

Función analogWrite

Escribe un valor analógico (onda PWM) en un pin. Se puede usar para encender un LED con diferentes niveles de brillo o para conducir un motor a diferentes velocidades. Después de una llamada a analogWrite (), el pin generará una onda cuadrada constante del ciclo de trabajo especificado hasta la próxima llamada a analogWrite ()

Función analogRead

Lee el valor del pin analógico especificado. La placa Arduino contiene un convertidor analógico a digital de 6 canales (8 canales en Mini y Nano, 16 en la Mega). Esto significa que mapeará voltajes de entrada entre 0 y 5 voltios en valores enteros entre 0 y 1023. Esto produce una resolución entre las lecturas de: 5 voltios / 1024 unidades o, .0049 voltios (4.9 mV) por unidad. El rango de entrada y la resolución se pueden cambiar usando analogReference ().
Sintaxis:analogRead(pin)

Función analogReference

Configura el voltaje de referencia utilizado para la entrada analógica (es decir, el valor utilizado como la parte superior del rango de entrada).
Sintaxis: analogReference(type)

Función pinMode

Configura el pin especificado para que se comporte como una entrada o una salida. Consulte la descripción de (pines digitales) para obtener detalles sobre la funcionalidad de los pines.
Sintaxis: pinMode(pin, modo)

Función map

Re asigna un número de un rango a otro. Es decir, un valor de Bajo se asignaría a Bajo, un valor de Alto a Alto, los valores intermedios a los valores intermedios, etc.

Función delay

Pausa el programa por la cantidad de tiempo (en milisegundos) especificado como parámetro. (Hay 1000 milisegundos en un segundo).

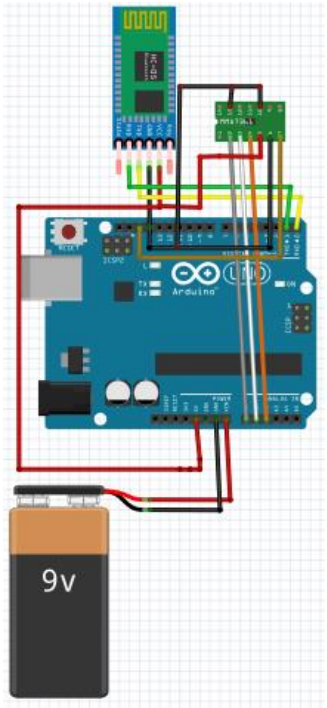
Desarrollo experimental

Como parte de la última práctica se buscó realizar un proyecto que integrara la mayor parte de los sensores vistos durante el semestre, por lo que es posible apreciar el uso de 2 placas ARDUINO, 2 módulos HC-05 (en configuración maestro-esclavo respectivamente), sensor de color y un acelerómetro.
La integración de todos estos sensores sin duda fue un reto para el equipo, sin embargo logramos culminar exitosamente con el desarrollo de nuestras ideas para la presente práctica.

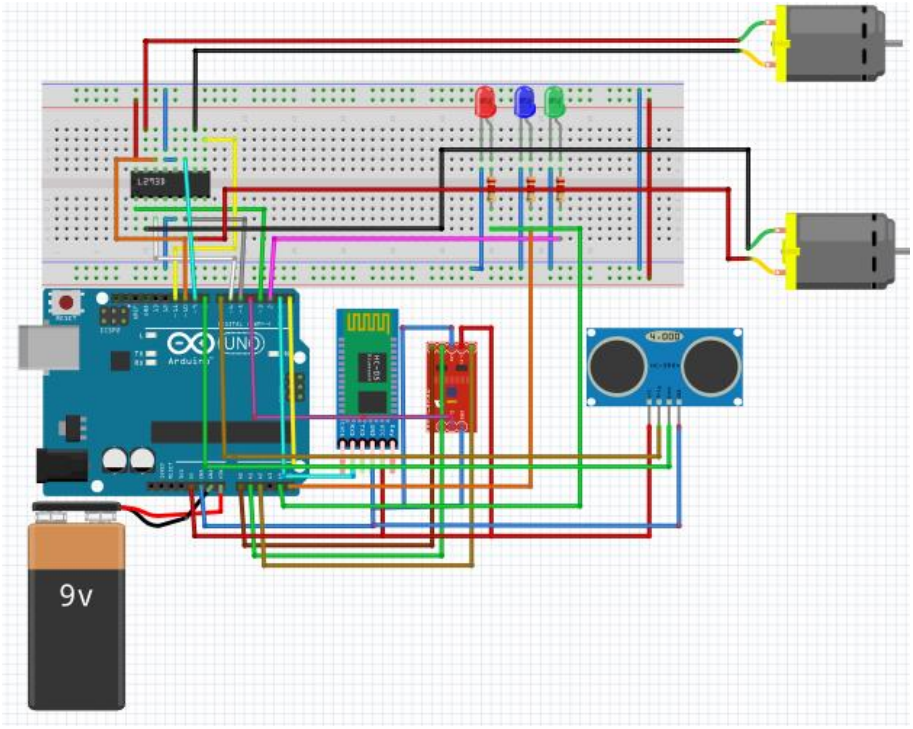
1. Como primer paso se analizó el funcionamiento de cada uno de los componentes a utilizar.
2. Una vez hecho esto decidimos comenzar a realizar las conexiones respectivas del acelerómetro y el módulo Bluetooth HC-05 en forma de esclavo.
3. Ya que se cable el circuito maestro se realizaron pruebas para ver si este funcionaba de la manera correcta.

4. Posteriormente se realizó la conexión del segundo circuito separado correspondiente al esclavo. Dicho circuito contenía un módulo HC-05, un ARDUINO, un puente H para controlar los movimientos del robot, además de un sensor de color que le permite al R2 D2 identificar 3 colores diferentes.
5. Finalmente procedimos a la subir los códigos de ARDUINO que se mostrarán a continuación y se comprobó que cada componente funcionará de forma adecuada.

Los diagramas de conexión se muestran a continuación.



Esquema del circuito tipo Maestro que incluye una placa ARDUINO, acelerometro y módulo Bluetooth Hc-05



Esquema del circuito tipo Esclavo que incluye una placa ARDUINO, sensor ultrasónico, sensor de color y módulo Bluetooth Hc-05

Como se mencionó anteriormente, como parte del presente proyecto se realizaron 2 códigos. El primero de ellos fue el código de maestro. Dentro de este código podemos observar algunas operaciones y otras funciones. A continuación se anexa el código obtenido para el maestro.

```
#include <SoftwareSerial.h>
#include <math.h>

int xPin = A3;
int yPin = A4;

int xVal = 0;
int yVal = 0;
int zVal = 0;

double angleYZ = 0;
double angleXZ = 0;

int xValor;
int yValor;

SoftwareSerial mySerial(2,3);// RX,TX
```

Figura 1: Declaración de variables, e inclusión de bibliotecas especiales.

```
void setup()
{
   analogReference(EXTERNAL)
   Serial.begin(9600);
   mySerial.begin(9600);
   pinMode(yPin, INPUT);
   pinMode(xPin, INPUT);
}
```

Figura 2: Definición del modo de uso para pines y establecimiento de la velocidad del puerto serie.

```
void loop()
{
  xVal = analogRead(0);
  yVal = analogRead(1);
  zVal = analogRead(2);

  xVal = map(xVal, 0, 1023, -500, 500);
  yVal = map(yVal, 0, 1023, -500, 500);
  zVal = map(zVal, 0, 1023, -500, 500);

  angleXZ = atan((double)xVal / (double)zVal);
  angleXZ = angleXZ*(57.2958);

  angleYZ = atan((double)yVal / (double)zVal);
  angleYZ = angleYZ*(57.2958);

    if(xVal>=0&&zVal>=0) //Cálculo del ángulo dependiendo del cuadrante en el plano XZ
      angleXZ=atan(abs((double)zVal/(double)xVal))*(57.2958);
    else
    if(xVal<0&&zVal>0)
      angleXZ=180-atan(abs((double)zVal/(double)xVal))*(57.2958);
    else
    if(xVal<0&&zVal<0)
      angleXZ=atan(abs((double)zVal/(double)xVal))*(57.2958)+180;
    else
    if(xVal>0&&zVal<0)
      angleXZ=360-atan(abs((double)zVal/(double)xVal))*(57.2958);
    if(yVal>=0&&zVal>=0) //Cálculo del ángulo dependiendo del cuadrante en el plano YZ
      angleYZ=360-atan(abs((double)zVal/(double)yVal))*(57.2958);
    else
    if(yVal<0&&zVal>0)
      angleYZ=180+atan(abs((double)zVal/(double)yVal))*(57.2958);
    else
    if(yVal<0&&zVal<0)
      angleYZ=180-atan(abs((double)zVal/(double)yVal))*(57.2958);
    else
    if(yVal>0&&zVal<0)
      angleYZ=atan(abs((double)zVal/(double)yVal))*(57.2958);
```

Figura 3: Obtención de ángulos a partir del acelerómetro

Para este segmento del código se realizaron operaciones matemáticas para obtener ángulos precisos que variarán al cambiar la posición del acelerómetro , por ello se añadió la biblioteca math.h en el inicio.

```
//ADELANTE
  if(angleYZ>=170 && angleYZ <190)
  {
    char b;
    Serial.println('b');
    mySerial.print('b');
  }

  else if(angleYZ>=190 && angleYZ <=250)
  {
    char a;
    Serial.println('a');
    mySerial.print('a');
  }
//ATRAS
  else if(angleYZ>=310 && angleYZ <345)
  {
    char c;
    Serial.println('c');
    mySerial.print('c');
  }

  else if(angleYZ>=345 && angleYZ <=360)
  {
    char d;
    Serial.println('d');
    mySerial.print('d');
  }
```

Figura 4: Movimientos en Y

En este segmento del código se puede observar que se realizaron diversos ciclos para controlar los movimientos en el eje de las Y para que los motores pudieran hacer el movimiento hacia adelante y hacia atrás una vez que se cumplían ciertas condiciones. Como es de suponer, se tienen dos ciclos para cada una de la sentencias (adelante o atrás) debido a que con esto se controló la velocidad con pulsos de PWM en el código del Esclavo explicado más adelante.

```
//MOVIMIENTOS EN X

//DERECHA
  else if(angleXZ>=110 && angleXZ <150)
  {
    char e;
    Serial.println('e');
    mySerial.print('e');
  }

  else if(angleXZ>=150 && angleXZ <=190)
  {
    char f;
    Serial.println('f');
    mySerial.print('f');
  }
//IZQUIERDA
  else if(angleXZ>=0 && angleXZ <40)
  {
    char g;
    Serial.println('g');
    mySerial.print('g');
  }

   else if(angleXZ>=40 && angleXZ <=70)
  {
    char g;
    Serial.println('h');
    mySerial.print('h');
  }


//NINGUNO
  else if ((angleYZ!=((angleYZ>=170 && angleYZ <=250)
  ||(angleYZ>=295 && angleYZ <=360)))||(angleXZ!=((angleXZ>=110 && angleXZ <=190)
  ||(angleXZ>=0 && angleXZ <=70))))
  {
    char z;
    Serial.println('z');
    mySerial.print('z');
  }
```

Figura 5: Movimientos en X

Este segmento del código es muy parecido al que se explicó anteriormente, únicamente se incluyó una sentencia para cuando ninguna de las condiciones se cumplia.

Anteriormente se explicó el código que se descargó en la placa ARDUINO correspondiente a la modalidad de maestro. A continuación se explica el código utilizado como esclavo.

```
#include <SoftwareSerial.h>

char a, b, c, d, e, f, g, h;

int V0=4,S0=14,S1=15,S2=16,S3=17;
int lg=2, lb=19, lr=18;

int enA=3;
int enB=11;

long distancia;
long tiempo;

int izqA=5;
int izqB=6;
int derA=9;
int derB=10;

int x=0;

char dist;

int var=0;

char state = 0;

float VR,VB,VG;
```

Figura 6: Declaración de variables, e inclusión de bibliotecas especiales.

En este segmento de código se muestran los nombres de las distintas variables utilizadas.

```
void setup()
{
   Serial.begin(9600);

   pinMode(S0,OUTPUT);
   pinMode(S1,OUTPUT);
   pinMode(S2,OUTPUT);
   pinMode(S3,OUTPUT);
   pinMode(V0,INPUT);

   pinMode(lg,OUTPUT);
   pinMode(lb,OUTPUT);
   pinMode(lr,OUTPUT);

   digitalWrite(lg,LOW);
   digitalWrite(lb,LOW);
   digitalWrite(lr,LOW);

   digitalWrite(S0,HIGH);
   digitalWrite(S1,HIGH);

   pinMode(derA, OUTPUT);
   pinMode(derB, OUTPUT);
   pinMode(izqA, OUTPUT);
   pinMode(izqB, OUTPUT);
   pinMode(enA, OUTPUT);
   pinMode(enB, OUTPUT);

   pinMode(7,OUTPUT);
   pinMode(8,INPUT);
}
```

Figura 7: Definición del modo de uso para pines y establecimiento de la velocidad del puerto serie.

Aquí se define cada pin en su respectiva modalidad como entrada o salida de datos.

```
void loop()
{
    digitalWrite(7,LOW); // Por cuestión de estabilización del sensor
    delayMicroseconds(5);
    digitalWrite(7, HIGH); //envío del pulso ultrasónico
    delayMicroseconds(10);
    tiempo=pulseIn(8, HIGH);
    distancia= int(0.017*tiempo); //fórmula para calcular la distancia obteniendo un valor entero

    digitalWrite(S2,LOW); //RED
    digitalWrite(S3,LOW);
    VR=pulseIn(VO,HIGH);

    digitalWrite(S2,LOW); //BLUE
    digitalWrite(S3,HIGH);
    VB=pulseIn(VO,HIGH);

    digitalWrite(S2,HIGH); //GREEN
    digitalWrite(S3,HIGH);
    VG=pulseIn(VO,HIGH);
```

Figura 8: Definición del modo de uso para pines y establecimiento de la velocidad del puerto serie.

En el presente ciclo se repiten las instrucciones en caso de que sus sentencias sean verdaderas. Por su parte las líneas de código que se observan sirven para determinar la distancia con el sensor ultrasónico y para establecer los parámetros de lectura de color mediante el sensor TCS3220.

```
if(VG<VR && VG<VB && VB>VR) //Se establecen los límites para evaluar el color
{
  //Serial.println("VERDE");
  digitalWrite(lg,HIGH);
}

if(VB<VR && VB<VG && VR>VG)
{
  //Serial.println("AZUL");
  digitalWrite(lb,HIGH);
}

if(VR<VB && VR<VG && VG>VB)
{
   //Serial.println("ROJO");
   digitalWrite(lr,HIGH);
}
```

Figura 9: Comparación de las magnitudes entre los 3 distintos colores

```
if(Serial.available() > 0)
{

  state = Serial.read();

    if (state=='a')
    {
      analogWrite(enA,150);
      analogWrite(enB,150);

      digitalWrite(derA,HIGH);
      digitalWrite(derB,LOW);
      digitalWrite(izqA,HIGH);
      digitalWrite(izqB,LOW);
    }

    else if (state=='b')
    {
      analogWrite(enA,255);
      analogWrite(enB,255);

      digitalWrite(derA,HIGH);
      digitalWrite(derB,LOW);
      digitalWrite(izqA,HIGH);
      digitalWrite(izqB,LOW);
    }

    else if (state=='c')
    {
      analogWrite(enA,150);
      analogWrite(enB,150);

      digitalWrite(derA,LOW);
      digitalWrite(derB,HIGH);
      digitalWrite(izqA,LOW);
      digitalWrite(izqB,HIGH);
    }

    else if (state=='d'||distancia<10)
    {
      analogWrite(enA,255);
      analogWrite(enB,255);

      digitalWrite(derA,LOW);
      digitalWrite(derB,HIGH);
      digitalWrite(izqA,LOW);
      digitalWrite(izqB,HIGH);
    }
```

Figura 10: Ejecución de movimientos

```
else if (state=='e')
{
  analogWrite(enA,150);
  analogWrite(enB,150);

  digitalWrite(derA,LOW);
  digitalWrite(derB,HIGH);
  digitalWrite(izqA,HIGH);
  digitalWrite(izqB,LOW);
}

else if (state=='f')
{
  analogWrite(enA,255);
  analogWrite(enB,255);

  digitalWrite(derA,LOW);
  digitalWrite(derB,HIGH);
  digitalWrite(izqA,HIGH);
  digitalWrite(izqB,LOW);
}
else if (state=='g')
{
  analogWrite(enA,150);
  analogWrite(enB,150);

  digitalWrite(derA,HIGH);
  digitalWrite(derB,LOW);
  digitalWrite(izqA,LOW);
  digitalWrite(izqB,HIGH);
}

else if (state=='h')
{
  analogWrite(enA,255);
  analogWrite(enB,255);

  digitalWrite(derA,HIGH);
  digitalWrite(derB,LOW);
  digitalWrite(izqA,LOW);
  digitalWrite(izqB,HIGH);
}
```

```
      else if (state=='z')
      {
        digitalWrite(derA,LOW);
        digitalWrite(derB,LOW);
        digitalWrite(izqA,LOW);
        digitalWrite(izqB,LOW);
      }

      else
      {
        //Serial.println("No identificado");
        digitalWrite(lg,LOW);
        digitalWrite(lb,LOW);
        digitalWrite(lr,LOW);
      }
  }

}
```

Para este segmento primero se establece la conexión entre los módulos Bluetooth. Una vez hecho esto se realiza el movimiento de los motores a su respectiva velocidad dependiendo de la variable recibida. Además el código en reversa para velocidad máxima tiene una condición en la que entra el código del sensor ultrasónico en caso de tener una distancia menor a 10 centímetros. Finalmente se observa un else que controla los cambios de color. Dicho else se encarga de apagar los diodos LED cuando no se cumple ninguna de las condiciones correspondientes al segmento del código de color.

En las imágenes que se muestran a continuación se observan los resultados físicos obtenidos

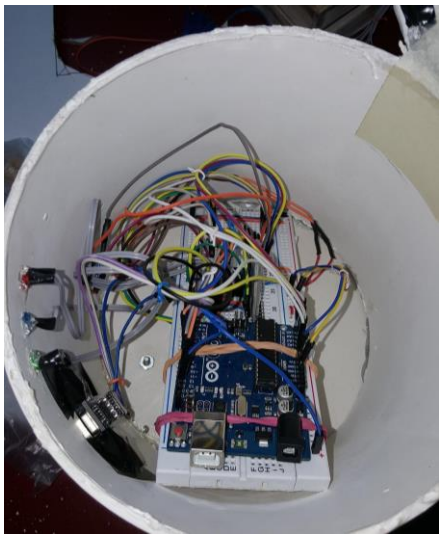


Figura 11: Circuitos internos (Esclavo)

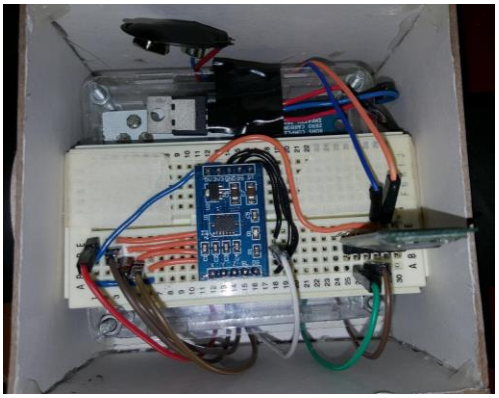


Figura 11: Circuitos internos (Maestro)

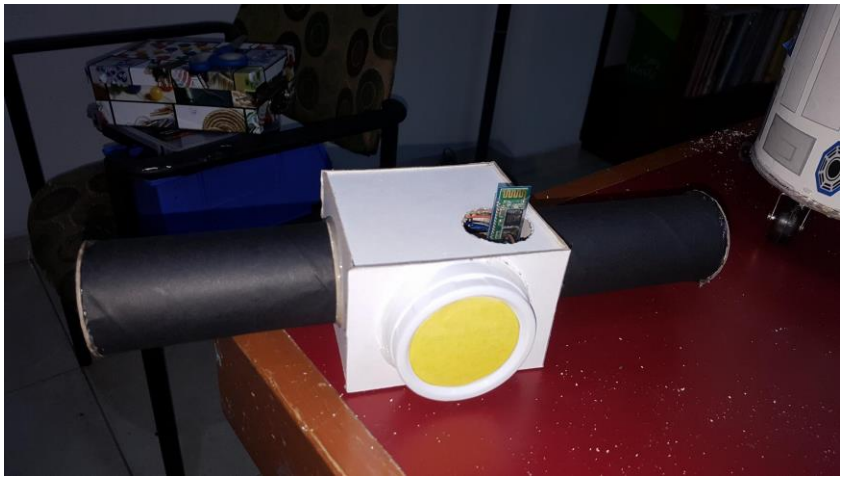


Figura 12: Estructura física del volante

Figura 13: Estructura física R2 D2

Figura 14: Póster presentado en la exposición

Análisis de resultados

Para el presente proyecto se nos presentaron diversas complicaciones a lo largo de la elaboración del mismo. Uno de los más grandes retos a los que nos enfrentamos fue el envío y recepción de datos mediante los 2 módulos bluetooth, ya que estos no se sincronizaban de la forma correcta. No obstante, se trabajó muy duro hasta que se consiguió una comunicación exitosa entre ambas placas de arduino. Por otra parte, se pensó en instalar un buzzer para emitir sonidos pero al momento de subir el código a la placa, nos dimos cuenta de que las sentencias para el movimiento de los motores se veía afectada por lo que decidimos dejar afuera ese elemento y agregar el sensor de color y ultrasonido. La implementación de estos fue sencillo ya que se reciclaron los códigos de las prácticas pasadas, por lo que únicamente nos enfocamos en que el carrito pudiera moverse de la forma adecuada al subir los códigos antes mencionados. Finalmente se realizó la presentación en la expo, y aunque la respuesta del robot R2 D2 no fue óptima, logró sacar una que otra sonrisa e impresión.

Conclusiones

La realización de este proyecto sin duda alguna fue un gran reto, no obstante dejó en su camino muchos aprendizajes nuevos. En un principio fue un poco complicado lograr configurar una placa de arduino como esclava y la otra como maestra, para que de este modo una se encargara de enviar datos a través del monitor serial a otro arduino, un circuito se encargaba de de almacenar los valores que entrega el acelerómetro de acuerdo a la posición que se le daba, y dependiendo de los rangos del valor dentro de ciertos rangos de ángulo, se enviaría un carácter específico a través del puerto serial al circuito que controlaba el sentido de giro de los motores, teniendo cuatro posibles escenarios: hacia adelante, hacia atrás, izquierda y derecha. Además se utilizaron dos sensores más: el sensor de color y el sensor ultrasónico, cuando se detectaba un color (verde, rojo o azul), se encendía un diodo LED de acuerdo a lo leído por el sensor, mientras que el ultrasónico funcionaba para detectar si había un obstáculo detuviera los motores para que el robot no chocará contra dicho objeto. Gracias a la integración de estos componentes electrónicos, la programación y la correcta conexión entre los componentes en conjunto, permitieron obtener un funcionamiento muy cercano al que teníamos planeado, además utilizamos muchos de los conocimientos adquiridos a través de este curso para poder aplicarlos en la vida real en un ambiente recreativo como lo es la creación de un robot semi autómata al que se le puede comandar inalámbricamente una serie de acciones.

Bibliografía

- Munguia, F. (2018, January 20). Sensor de color TCS3200 con Arduino. Retrieved April 17,
- 2018, from https://hetpro-store.com/TUTORIALES/sensor-de-color-tcs3200-con-arduino/
- Color. (2018, April 11). Retrieved April 17, 2018, from https://es.wikipedia.org/wiki/Color
- DigitalWrite(). (n.d.). Retrieved April 17, 2018, from
- https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/
- PulseIn(). (n.d.). Retrieved April 17, 2018, from
- https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/
- (n.d.). Retrieved April 17, 2018, from https://www.arduino.cc/en/Serial/Println