CARRITO SENSOR DE COLOR Y SENSOR SEGUIDOR DE LÍNEA

Abstract

Este articulo habla sobre el desarrollo que se realizó para elaborar un carrito que por medio de Arduino será capaz de encontrar puntos que se indicaran atravez de una app móvil que nos brinda Arduino en base a bluethooth.

***Palabras Clave*— Carrito, Arduino, Sensor**

I. Objetivo del Proyecto

Aplicar los conocimientos de la clase de Sistemas Digitales y Teoría Computacional en base a un carrito que tendrá que hallar puntos establecidos por medio de un patrón donde apoyándose con una aplicación móvil conectada por bluethooth al Arduino se indicara que punto buscar y se detendrá hasta que el carro encuentre el punto y espere la nueva instrucción

II. Marco teorico

El kit para armar carrito para Arduino es uno de los proyectos más populares entre estudiantes y aficionados a la robótica.

A. Antecedentes

De proyectos anteriores se encuentran el carrito seguidor de línea que trabaja con Arduino y sensores infrarrojos donde su misión es seguir una línea negra marcada en un fondo blanco. Arrojando 1 o 0 cuando detectan o no la línea negra, indicándole así a los motores que se detengan o sigan cuando los sensores les arroja tales valores.

Otro proyecto sería el carrito manejado a control remoto usando el módulo de Bluethooth de Arduino y usando la aplicación de esta misma usándolo como control remoto que le indica al Arduino que hacer, moverse a la derecha, dar vuelta, etc.

Para el sensor de color como solo compara los colores en base a los tres colores primarios (rojo, verde y azul) hallamos un proyecto donde mostrándole al sensor los colores, prendia un led de acuerdo al color detectado.

B. Base de circuitos logicos

Los materiales que se necesitaron al realizar este proyecto fueron:

- Arduino Uno
- Kit de chasis para carrito
- 2 sensores infrarrojos FC-51
- 1 Sensor de color TCS3200
- Puente H
- Modulo de Bluethooth HC-6

C. Diagramas de circuitos ,

A continuación, se presentaran los diagramas de cada uno de los componentes que se necesitaron en la realización de este proyecto:

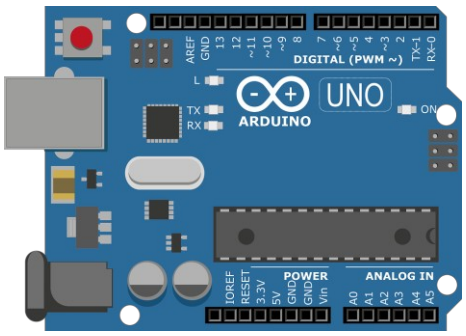


Figure 1 Complementos del Arduino UNO

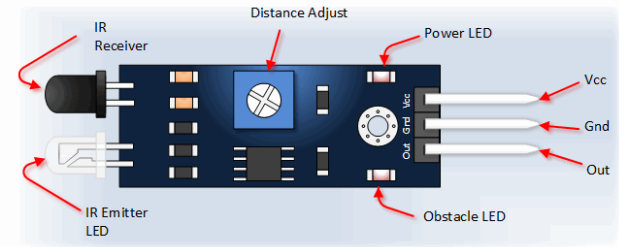


Figure 2 Diagrama de los componentes del sensor infrarrojo FC-51

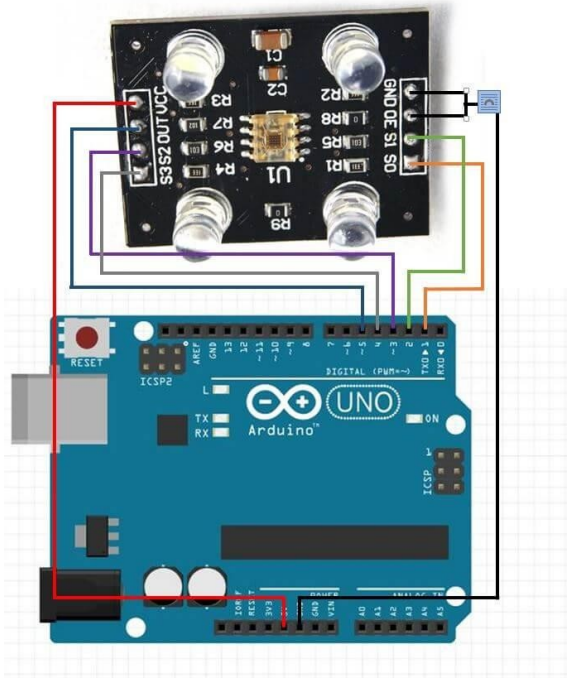


Figure 3 Diagrama que muestra como se conecta el sensor de color al arduino teniendo 5 entradas y 2 salidas a tierra y una a voltaje

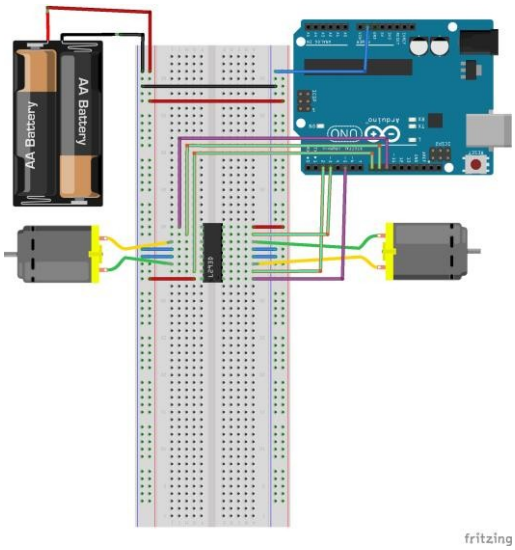


Figure 4 Diagrama que muestra la conexion del integrado l293d como puente H para los dos motores del carrito

.

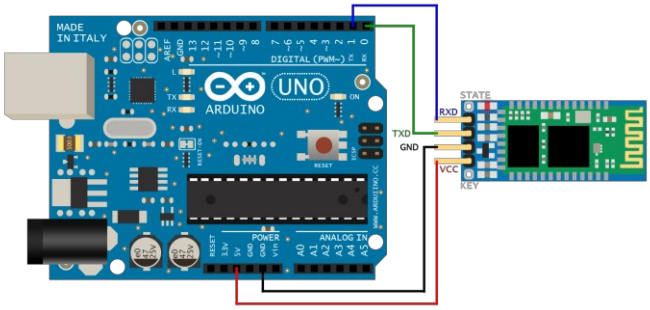


Figure 5 Modulo Bluethooth HC-6 y su diagrama de conexión

III. Marco experimental

En este apartado se explicara lo experimental que se hizo al realizar este proyecto:

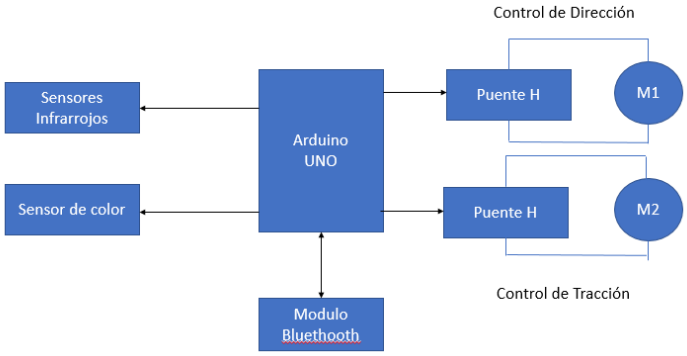


Figure 6 Diagrama de bloques

A. Descripción del funcionamiento lógico

El funcionamiento del proyecto se da en gran parte, a una placa Arduino Uno, la cual será programada para poder interpretar las señales que se presenten, en este caso será la señal de Bluetooth, la cual le dirá hacia donde ir. Esta señal el arduino deberá de interpretarla y mediante codigo y los sensores equipados, deberá realizar las acciones de moverse de un punto a otro y así poder explotar los recursos disponibles.

El carrito empieza en un estado inicial basándonos en una tela donde está dibujado el siguiente patrón:

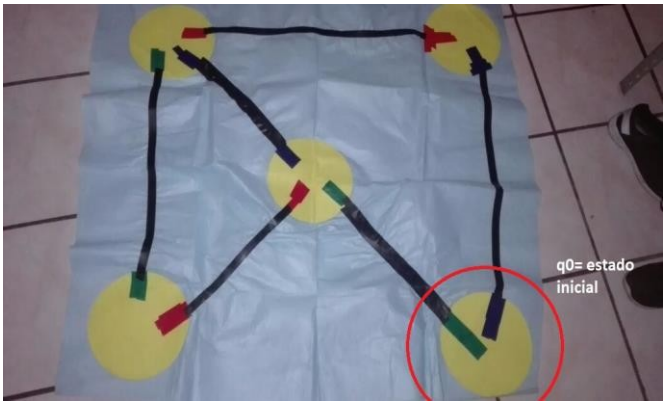


Figure 7 Patrón del carrito

Ya que se define cual será el estado inicial del carrito, en este caso será q0.

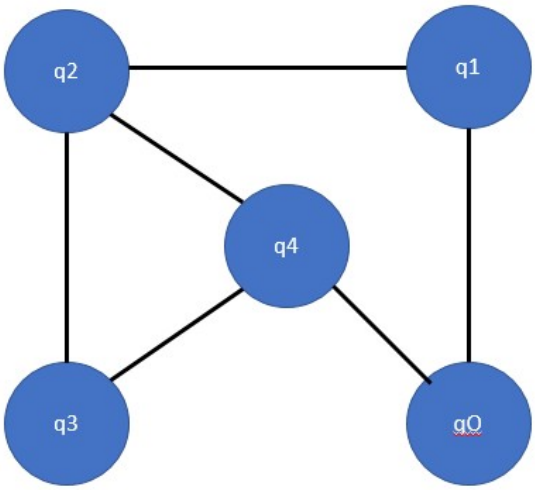


Figure 8 Diagrama que define los estados del patrón para el carrito

Por medio de la aplicación móvil en su "Terminal mode" que se conecta por el módulo bluetooth, se envía a que punto desea que se vaya el carrito y este primero valida las cadenas mandadas sobre esta terminal y después avanzara de acuerdo a los patrones ya definidos en el programa hecho en el IDE de Arduino.

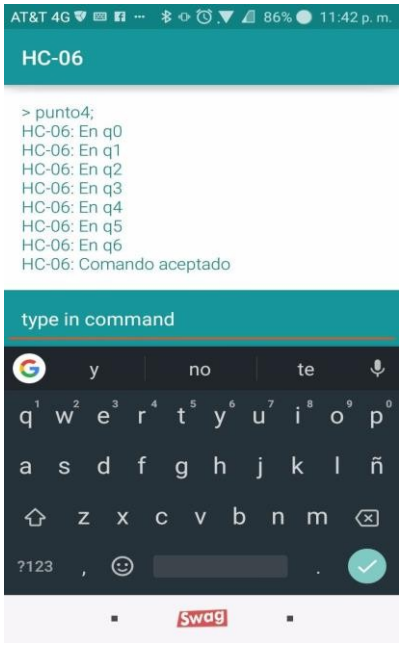
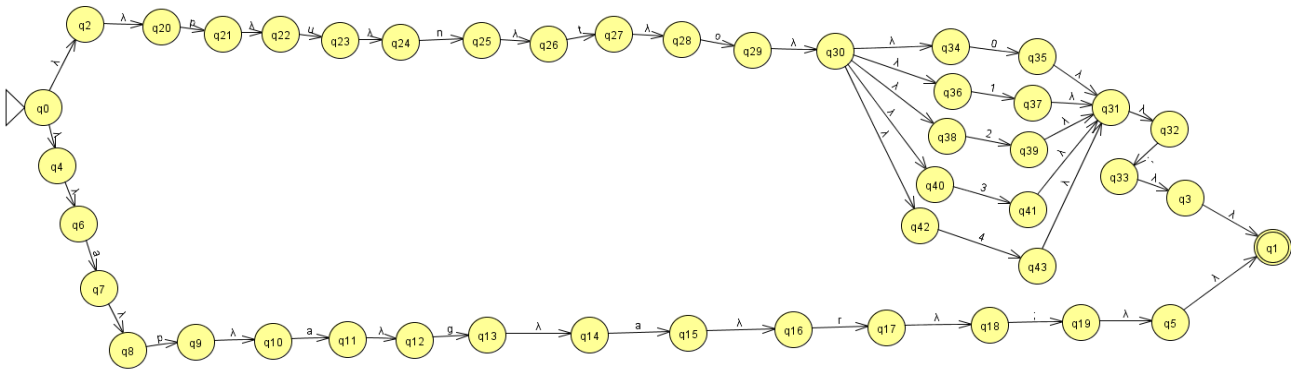


Figure 9 Terminal mode de la aplicación movil de Arduino y el modulo Bluetooth

Como se ve en la Figure 7 los puntos tienen en sus orillas colores por lo que le servirán al carrito saber que camino hacer para llegar al punto que desea,
Por ejemplo, si el carrito esta en el estado inicial (q0) y quiere ir a q4, el carrito girara sobre su propio eje para que con el sensor de color reconozca el camino de la tira color verde como se muestra en la Figure 7 y este siga con los sensores infrarrojos el camino de la línea negra hasta llegar a q4 y así será lo mismo, buscar el color del camino para que llegue a su destino.
Todos los caminos para llegar a los respectivos puntos entre si están declarados dentro del programa que se realizo para el funcionamiento del proyecto.

B. Diagrama de estados



C. Gramatica

LHS		
S	→	pA
S	→	aG
A	→	uB
B	→	nC
C	→	tD
D	→	oE
E	→	0F
F	→	;M
G	→	pH
H	→	al
I	→	gJ
J	→	aK
K	→	rL
L	→	;M
E	→	1F
E	→	2F
E	→	3F
E	→	4F
M	→	λ

D. Derivación

Input punto1;
String accepted! 9 nodes generated.

LHS		RHS
S	→	pA
S	→	aG
A	→	uB
B	→	nC
C	→	tD
D	→	oE
E	→	0F
F	→	;M
G	→	pH
H	→	al
I	→	gJ
J	→	aK
K	→	rL
L	→	;M
E	→	1F
E	→	2F
E	→	3F
E	→	4F
M	→	λ

Production	Derivation
	S
S→pA	pA
A→uB	puB
B→nC	punC
C→tD	puntD
D→oE	puntoE
E→1F	punto1F
F→;M	punto1;M
M→λ	punto1;

Input apagar;
String accepted! 9 nodes generated.

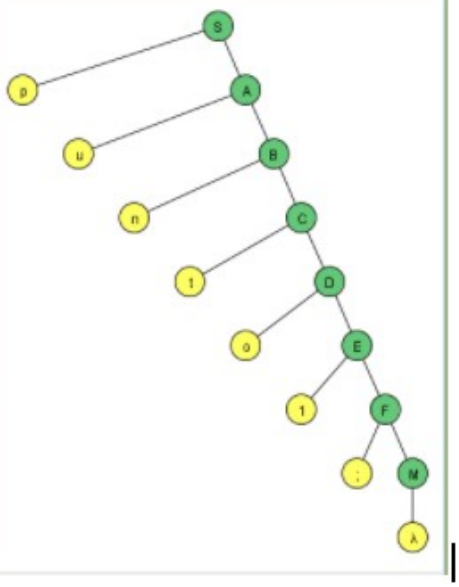
LHS		RHS
S	→	pA
S	→	aG
A	→	uB
B	→	nC
C	→	tD
D	→	oE
E	→	0F
F	→	;M
G	→	pH
H	→	al
I	→	gJ
J	→	aK
K	→	rL
L	→	;M
E	→	1F
E	→	2F
E	→	3F
E	→	4F
M	→	λ

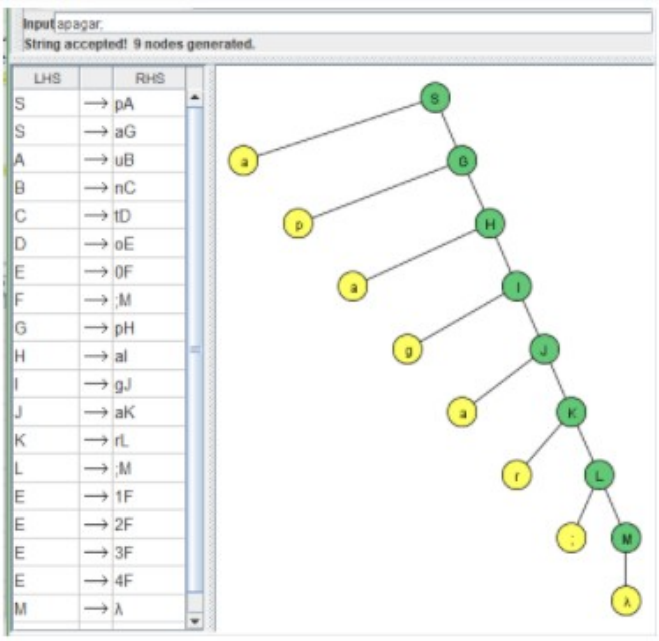
Production	Derivation
	S
S→aG	aG
G→pH	apH
H→al	apal
I→gJ	apagJ
J→aK	apagaK
K→rL	apagarL
L→;M	apagar;M
M→λ	apagar;

E. Árboles de Derivación

Input punto1;
String accepted! 9 nodes generated.

LHS		RHS
S	→	pA
S	→	aG
A	→	uB
B	→	nC
C	→	tD
D	→	oE
E	→	0F
F	→	;M
G	→	pH
H	→	al
I	→	gJ
J	→	aK
K	→	rL
L	→	;M
E	→	1F
E	→	2F
E	→	3F
E	→	4F
M	→	λ





IV. Funcionamiento del proyecto

A. Manual de Usuario

1. Para poder probar este proyecto es necesario tener un dispositivo móvil con la aplicación de Arduino Bluetooth instalada y asi poder conectar el modulo Bluetooth que tiene el carrito con el dispositivo y empezar a mandar la primera instrucción.

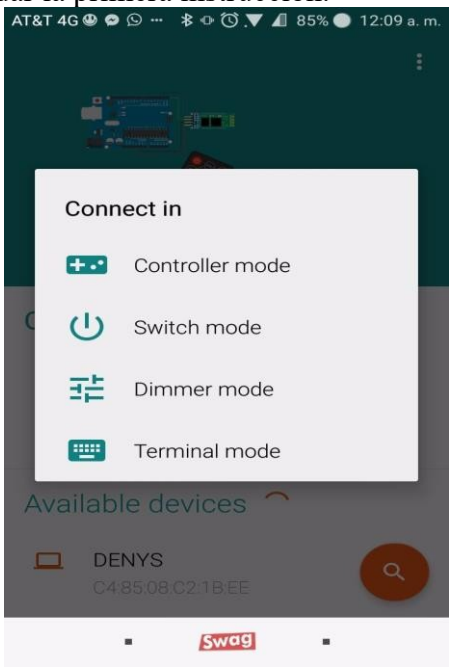
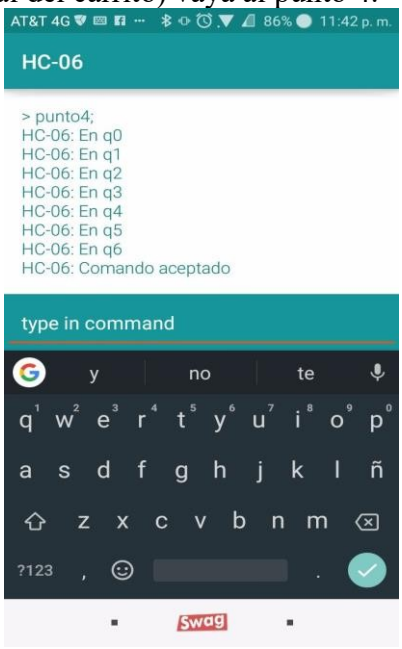


Figure 10 Pantalla donde el dispositivo y el modulo ya estan conectados

2. En la pantalla principal al conectar los dispositivos tendremos que escoger de entre las opciones que muestra la Figure 10, Terminal mode, que es donde se le escribirá a que punto desea que avance el carrito. En la siguiente imagen se tomo como ejemplo que del punto 0 (q0 ya que es el estado inicial del carrito) vaya al punto 4.



3. Al dar enter y que la consola halla aceptado el comando, el carrito empezara a girar sobre su propio eje buscando su camino a tomar para llegar al punto.

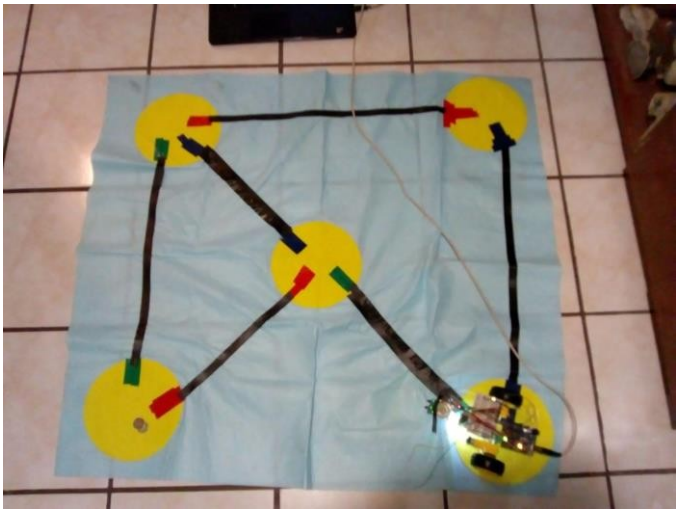


Figure 11 El carrito empieza a girar buscando el color verde que es el que lo lleva al punto 4 deacuerdo a la Figure 8

4. Despues de que el carrito haya localizado el color asignado con el sensor de color de acuerdo al camino para el punto ya dicho, el carrito procederá a avanzar atraves de la linea negra gracias a los sensores infrarrojos, hasta llegar al punto predicho.

5. Si se quiere volver a mandar una nueva instrucción de un nuevo punto a seguir, se tiene que repetir el paso 2, y el carrito volverá a buscar el color del camino para poder llegar a su punto de destino.

B. Programación detallada

En la programación usamos el IDE de Arduino para poder elaborar el programa y mandarlo a la placa de Arduino.
Se declaro primero los pines en donde se localizan la entrada y salida del módulo de Bluetooth, después se declara una clase que se usa para poder asignarle a cada punto su transición o su camino que hace para llegar a cada punto.

```
#include <SoftwareSerial.h>
SoftwareSerial BlueTooth(13,12); //Pines de entrada y s

class Transicion{
  public:
    String transicion[2]={"",""};
    String getNombre(){
      return nombre;
      }
    void setTransicion(String primera,String segunda){
      transicion[0]=primera;
      transicion[1]=segunda;
      }
  private:
    String nombre="";
  };
```

Se declaran los caminos de cada punto para llegar a otros puntos como arreglos e identificando cada camino por el color que debe seguir para llegar a ellos.

```
String q0_q1[]={"azul"};
String q0_q2[]={"verde","azul"};
String q0_q3[]={"verde","rojo"};
String q0_q4[]={"verde"};

String q1_q0={"azul"};
String q1_q2={"rojo"};
String q1_q3[]={"rojo","verde"};
String q1_q4[]={"azul","verde"};

String q2_q0[]={"azul","verde"};
String q2_q1={"rojo"};
String q2_q3={"verde"};
String q2_q4={"azul"};

String q3_q0[]={"rojo","verde"};
String q3_q1[]={"verde","rojo"};
String q3_q2={"verde"};
String q3_q4={"rojo"};
```

Se declaran las entradas y salidas de cada sensor con el pin en el que están conectados en la placa Arduino.

```
const int s0 = 7;  //color
const int s1 = 8;  //color
const int s2 = 9;  //color
const int s3 = 10;  //color
const int out = 11; //color
const int m1 = 6;   //motor1
const int m2 = 5; // motor2
const int mR = 2; // motor2 reversa
const int sl4 = 12;  //Seguidorlinea1
const int sl5  = 13;  //Segudirlinea2
int rojo = 0;
int verde = 0;
int azul = 0;
```

En la función de setup solo se dejaran declaradas que variables serán entradas y que variables serán salidas

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

  Serial.begin(9600);
  pinMode(s0,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
  pinMode(out,INPUT);
  pinMode (m1, OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(mR,OUTPUT);
  pinMode (sl4, INPUT);
  pinMode(sl5,INPUT);

  digitalWrite(s0,HIGH);
  digitalWrite(s1,HIGH);
}
```

En la siguiente imagen se muestra la función leer_cadenas que es la que se encarga de validar aquella cadena dada en la Terminal mode y si la primera verificación que la evalúa carácter por carácter, el autómata empieza

```
void loop() {
  leer_Cadenas(); //Se evalua la funcion de leer cadenas todo el tiempo
}

void leer_Cadenas(){
  char caracter='\b';
  while (BlueTooth.available()>0) {
    caracter = BlueTooth.read();//Se lee cacacter por caracter desde el bluetooth
    BlueTooth.flush();
    if(caracter==';'){
       cadena = cadena + ';'; //Primera verificacion, si la pase empieza le automata
       q0();
      }
      else{
        cadena = cadena + caracter;
      }
  }
}
```

Se declaran en el siguiente código los estados del automata

```
void q0(){//Estado inicial del automata
  if(cadena.charAt(contador)=='p'){
       contador++;
       BlueTooth.println("En q0");
       q1();
     }else if(cadena.charAt(contador)=='a'){
       contador++;
       q7();
     }else{
       qError();
     }
}
```

Estado final del autómata donde se define a que destino quiere llegar el carrito.

```
void q13(){//Estado final, si se llega aqui es por que el comando fue valido
  BlueTooth.println("Comando aceptado");
  if(cadena.equals("apagar;")){
     digitalWrite(m1,LOW);
     digitalWrite(m2,LOW);
     digitalWrite(mR,LOW);
     BlueTooth.println("Apagar");
  }else{//Si el comando aceptado es diferente a apagar, se empieza
    switch(cadena.charAt(5)){//Se evalua el nuemero existente en la cadena, que será el destino deseado
      case '1': recorrerTransiciones(numero.toInt());
                qDeseada = "q1";
        break;
      case '2': recorrerTransiciones(numero.toInt());
                qDeseada="q2";
        break;
      case '3': recorrerTransiciones(numero.toInt());
                qDeseada="q3";
        break;
      case '4': recorrerTransiciones(numero.toInt());
                qDeseada="q4";
```

Función que recorre las transiciones que debe seguir el carro dependiendo su posición final y actual

```
boolean siNumero(char numero){//Se recorel el conjunto d enumeros para verificar si se oorgó un numero vali
  for(int i=0;i<5;i++){
    if(numeros[i]==numero)
      return true;
    }
  return false;
}

void qError(){//Funcion para evaluar cadenas no validas
  BlueTooth.println("Comando introducido no valido");
  cadena="";
  contador=0;
}
```

Función que se encarga del funcionamiento del seguidor de línea la transportarse de una estado a otro, cada if esta adecuado a las condiciones que se pueden dar, y se obtendrá a configuración de los motores necesaria para que se continúe con el trayecto.

```
void SeguidorLinea(String estado){//Funcion que se encarga del funcio
  while(digitalRead(sl4)!= LOW && digitalRead(sl5)!=LOW){//Se va a re
      if(digitalRead(sl4)== HIGH && digitalRead(sl5)== HIGH)//Cada if
      {                                                      //a confi
           digitalWrite (m1, HIGH);
           digitalWrite (m2, HIGH);
      }

      if(digitalRead(sl4)== HIGH && digitalRead(sl5)== LOW)
      {
          digitalWrite (m1, LOW);
          digitalWrite (m2, HIGH);
      }
      if(digitalRead(sl4)== LOW && digitalRead(sl5)== HIGH)
      {
          digitalWrite (m1, HIGH);
          digitalWrite (m2, LOW);
      }
  }
```

Función que identifica los colores que se están evaluado en el sensor, se hará girar el carro sobre su propio eje hasta que encuentre el color que se necesita, al encontrarlo se pasa el control del carro al seguidor de línea

```
void girar(String colorl,String qActual){//Funcion que identifi
  if(colorl=="rojo"){
      do{//Se hara girar el carro sobre su propio eje hasta que
        color();//Se pasa el control del carro al seguidor de l
         digitalWrite (ml, HIGH);
         digitalWrite (mR, HIGH);//Salidas necesarias para que
      }while(!(rojo < azul && verde > azul && rojo < 25));
      delay (500);
      SeguidorLinea(qActual);
    }
  if(colorl=="azul"){
       do{
        color();
         digitalWrite (ml, HIGH);
         digitalWrite (mR, HIGH);

        }
      while(!(azul < rojo && azul < verde && verde < rojo));
      delay (500);
```

Función que evalúa los resultados obtenidos por el sensor, se asigna una y otra vez los valores arrojados por el sensor

```
void color()//Funcion que evalua los resultados obtenidos por el sen
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);//Se asigna una y otra vez los valores arroj
  rojo = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
  azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);
  verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
```

V. Resultados y Conclusiones

A lo largo del desarrollo del proyecto se nos dificulto en primer lugar conseguir todo el material, ya que por un lado el material era relativamente costoso y la mayoría se agotaba rápidamente.

El entender cómo funcionaba cada sensor y el saber como se manejaba la programación en Arduino también nos llevó mucho tiempo ya que también nos dábamos cuenta poco a poco que nos faltaba material para llevar a cabo el proyecto.

Al principio de acabar con toda la programación y probarla empezó a fallar demasiado y no funcionó como esperábamos, nosotros pensando que quizá la programación estaba mal planteada porque los motores ni siquiera se movían pero los sensores funcionaban de acuerdo a la programación, nos dimos cuenta al final que el voltaje no llegaba a los motores del carrito y es por eso que el carrito no avanzaba. Por lo que para resolver el problema se debe encontrar la manera de proporcionar a cada motor la misma cantidad de voltaje para un correcto funcionamiento, al menos 4.5.

Referencias

- https://hetpro-store.com/TUTORIALES/sensor-de-color-tcs3200-con-arduino/
- https://www.taloselectronics.com/kit-robot-seguidor-de-linea-para-arduino/
- https://jhosman.com/index.php/2013/06/04/construir-un-robot-seguidor-de-linea-con-arduino-softwarelibre-hardwarelibre/