



Proyecto Final

LABORATORIO 1

Ing. Informática

Integrantes:

- Nores Sonzini, Tristán Agustín
- Jimenez, Facundo
- Caffa Tadey, Joan
- Olmedo, Gregorio
- Albornoz, Gabriel Angel

Profesores:

- Nano, Monica Liliana
- Salamero, Martín

Fecha de presentación: 14/11/2023

Índice:

Introducción	3
Desarrollo	4
Elementos del juego:	4
Componentes:	4
Aplicación:	5
Funcionamiento:	6
Los Autos:	7
Los Arcos:	11
Conclusión	13

Introducción

En este proyecto se realizará un juego de competencia donde dos jugadores controlan, cada uno, un auto a control remoto mediante Bluetooth. El objetivo de este juego es el de meter, utilizando solamente los autos, una pelota en el arco del jugador contrario. El jugador que logre hacer la mayor cantidad de goles en el tiempo estipulado, será el ganador del juego.

Desarrollo

Elementos del juego:

- **Autos a control remoto:** Los autos son el principal elemento del juego, ya que son el representante del jugador en la cancha. Estos autos serán controlados mediante una App para el celular que se comunicará mediante Bluetooth.
- **Arcos:** Cada jugador tendrá su arco, el cual deberá defender de los intentos del otro jugador de hacer gol. Cada arco tendrá un simple display que mostrará un contador. Ese contador corresponde a la cantidad de goles que se hicieron en ese arco. También poseerá un botón para reiniciar los contadores una vez finalizado el juego.
- **Pelota:** una simple pelota al estilo de las de metegol, el cual los jugadores deberán empujar al arco contrario utilizando solamente los autos. Este se colocará al centro de la cancha al inicio de la partida y luego de cada gol.
- **Tiempo:** Los jugadores previamente acordarán el tiempo que va a durar el partido. Una vez finalizado, el jugador con menos goles en su arco gana.

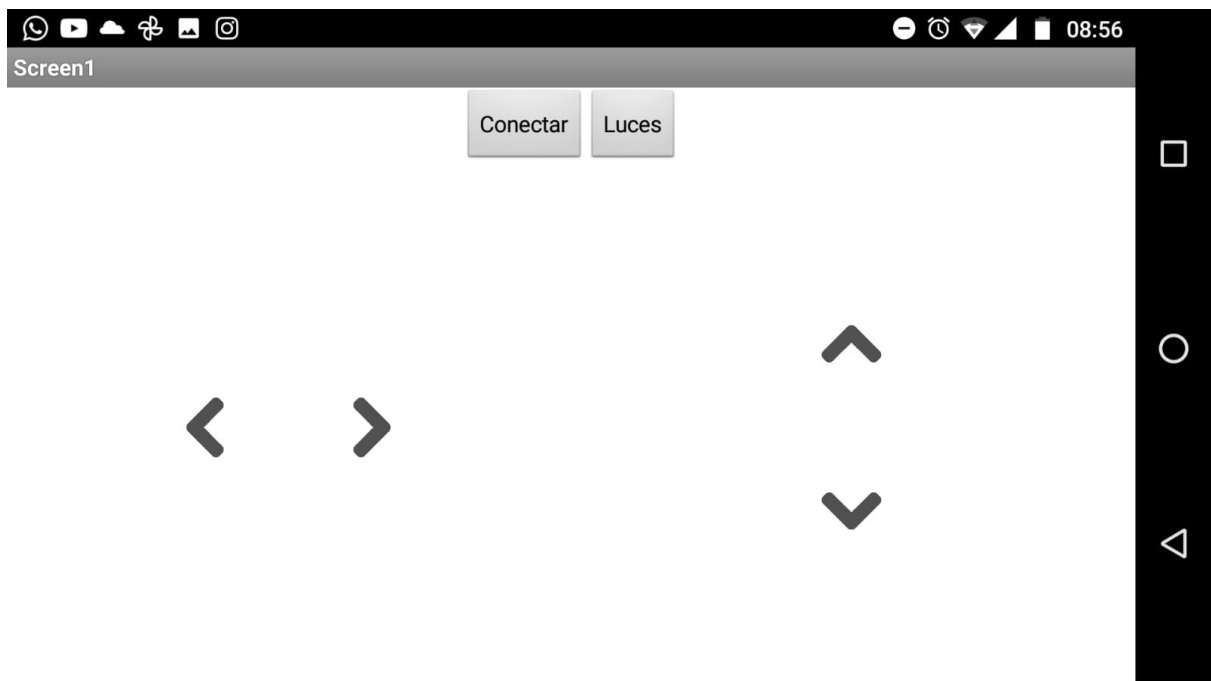
Componentes:

- Componentes de los autos (cada uno):
 - Arduino UNO.
 - Shield driver de motores L293D.
 - Modulo Bluetooth HC-06 (para comunicarse mediante Bluetooth con el Arduino).
 - Batería de 9 voltios.
 - 2 motores DC.
 - Leds y resistencias variadas.
- Componentes de los arcos (cada uno):
 - 1 display de 7 segmentos (para mostrar la cantidad de goles).
 - 2 pulsadores (uno para reiniciar el contador y otro para resetear el Latch SR).
 - Sensor de proximidad (para detectar la pelota cuando entra al arco).
 - Contador (para contar los goles).
 - Latch SR (para detectar una única vez la pelota que entra al arco, que luego se puede resetear con uno de los pulsadores mencionados anteriormente).
 - Decodificador BCD a 7 segmentos.
 - Resistencias variadas.
 - Batería de 9 voltios.

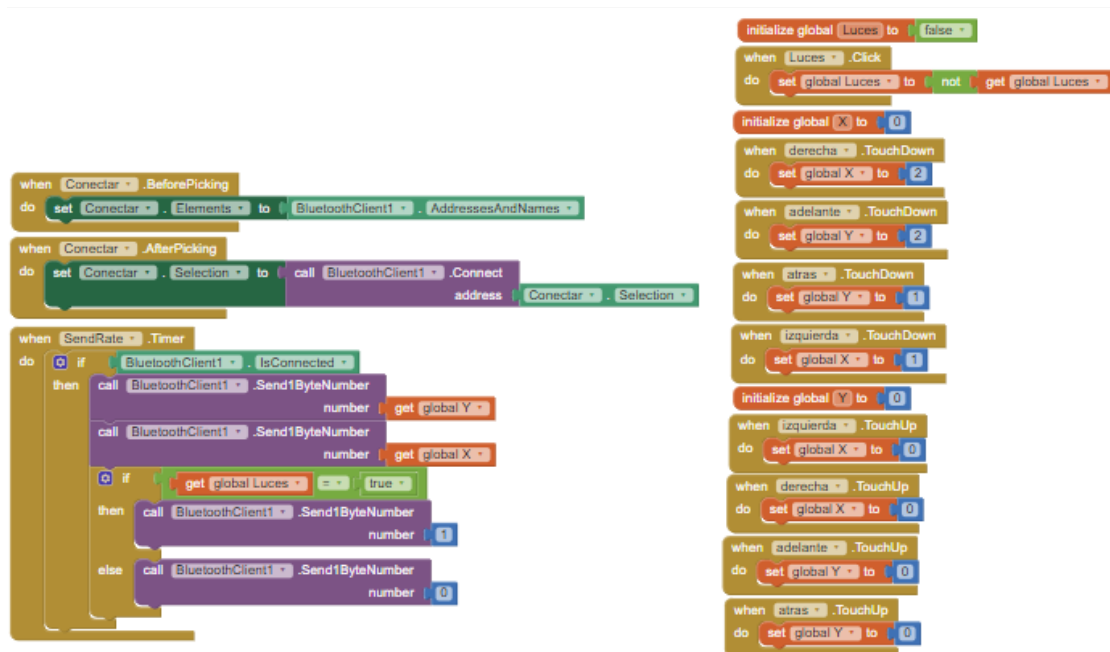
Aplicación:

El control de los autos será mediante una aplicación que se comunica con ellos mediante Bluetooth. Esta app tendrá las siguientes funcionalidades:

- Un botón que abre la lista de dispositivos Bluetooth presentes, para seleccionar el correspondiente al auto y poder comunicarse con él.
- Dos botones a la derecha apuntando uno hacia arriba y el otro hacia abajo, con la función de indicar si el auto irá hacia adelante o hacia atrás.
- Dos botones a la izquierda apuntando uno hacia la derecha y el otro hacia la izquierda, con la función de indicar si el auto girara hacia la derecha o hacia la izquierda.
- Un botón que abre la lista de dispositivos Bluetooth presentes, para seleccionar el correspondiente al auto y poder comunicarse con él.
- Un botón que enciende y apaga las luces del auto (Opcional).



El código en bloques de dicha app es el siguiente:



Funcionamiento:

Los primeros dos bloques a la izquierda son los bloques correspondientes al selector de dispositivos bluetooth disponibles.

Los bloques a la derecha son los de las variables:

- **X:** El nombre viene del eje X y almacena los valores correspondientes a girar a la derecha (un 2) y a girar a la izquierda (un 1). Si no está girando, almacena un 0.
- **Y:** El nombre viene del eje Y y almacena los valores correspondientes a avanzar (un 2) y a retroceder (un 1). Si no está moviéndose, almacena un 0.
- **Luces:** Almacena un valor booleano indicando si las luces están prendidas o no (Cambia su valor cada vez que se presiona el botón).

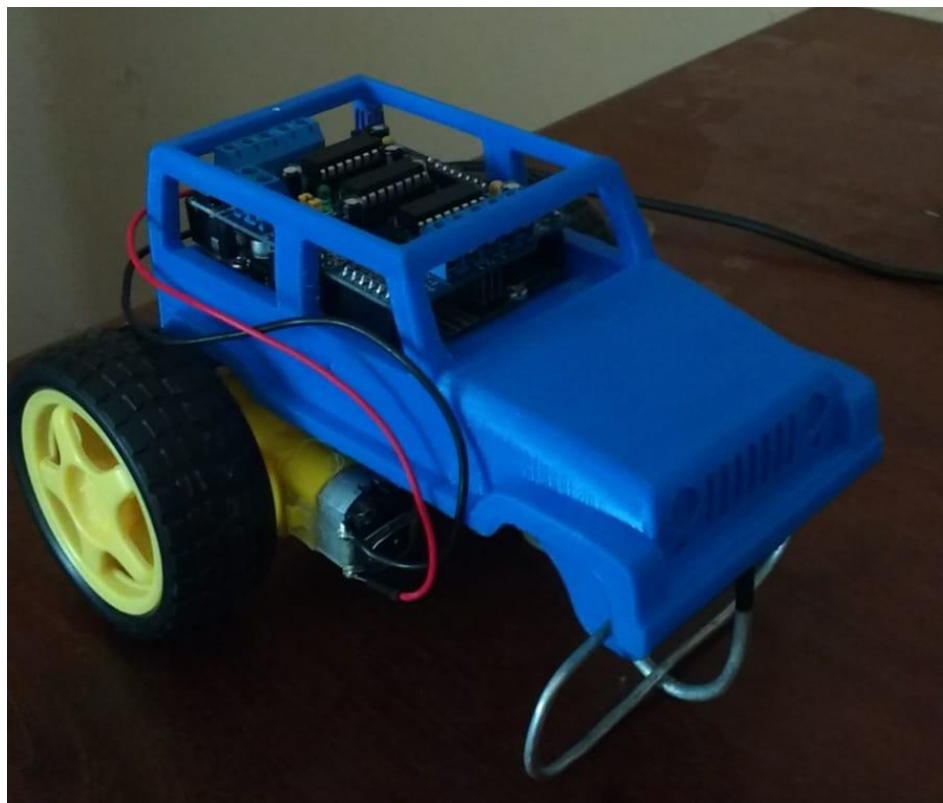
El último bloque en la esquina inferior izquierda es el encargado de mandar la información: Se usa un timer llamado “*SendRate*” que cada 100ms envía por bluetooth los valores de las variables Y, X y Luces. Esta es enviada como una cadena de 3 bytes.

Los Autos:

El diseño de los autos es el de un jeep, que se muestra a continuación:

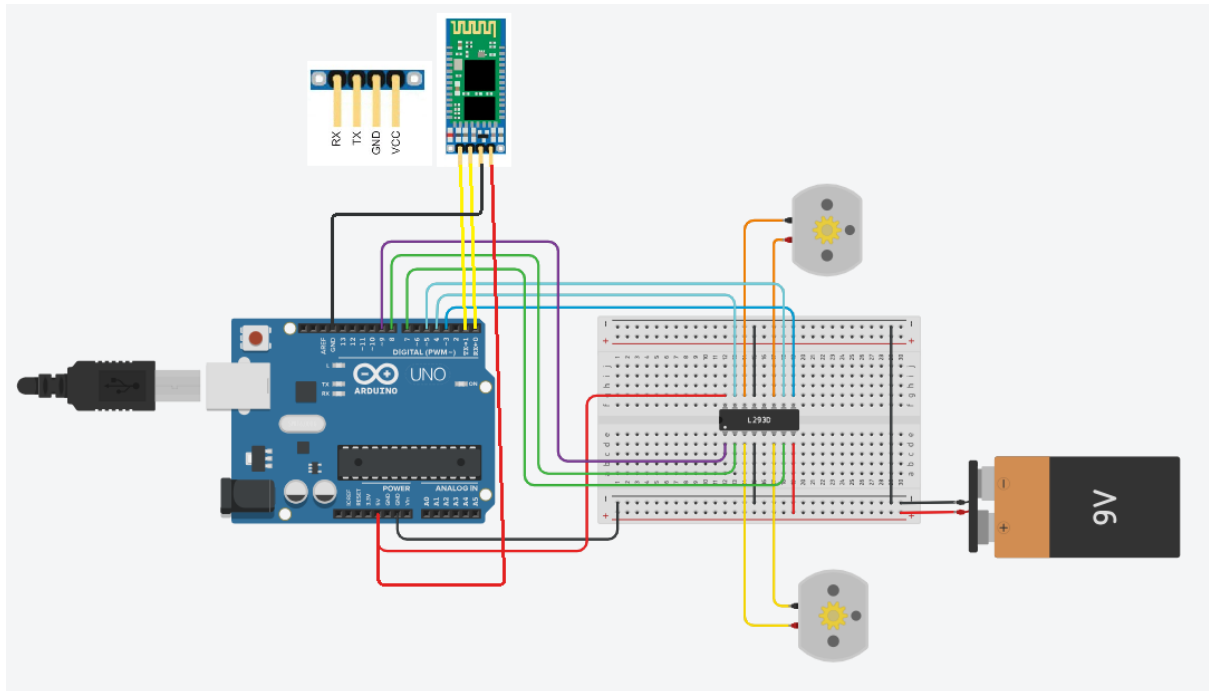


Como se ve, eso es solo la base, una vez armado con todos los componentes, quedaria asi:



Como se puede ver, se reemplazaron las ruedas del frente por alambres curvos para facilitar el giro del mismo.

El circuito basico del auto es el siguiente:



El shield driver de motores que utilizamos utiliza el integrado L293D, el mismo que esta en el circuito mostrado previamente.

Una vez armado el auto, se procede a programar al arduino con el siguiente código:

```
#include <AFMotor.h>
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);

#define MAX_VELOCIDAD 255
byte info[3];

void setup() {
  motor1.setSpeed(0);
  motor2.setSpeed(0);
  Serial.begin(9600);
}

void loop()
{
  if(Serial.available() > 0){
    Serial.readBytes(info, 3);
    //adelante
    if(info[0]==(byte)2){
      motor1.run(FORWARD);
      motor2.run(FORWARD);
      motor1.setSpeed(MAX_VELOCIDAD);
      motor2.setSpeed(MAX_VELOCIDAD);
    }
    //quieto
    if(info[0]==(byte)0){
      motor1.setSpeed(0);
      motor2.setSpeed(0);
    }
    //atras
    if(info[0]==(byte)1){
      motor1.run(BACKWARD);
      motor2.run(BACKWARD);
      motor1.setSpeed(MAX_VELOCIDAD);
      motor2.setSpeed(MAX_VELOCIDAD);
    }
    //derecha
    if(info[1]==(byte)2){
      motor1.run(BACKWARD);
      motor2.run(FORWARD);
      motor1.setSpeed(MAX_VELOCIDAD);
      motor2.setSpeed(MAX_VELOCIDAD);
    }
  }
}
```

```

//izquierda
if (info[1]==(byte)1) {
    motor1.run(FORWARD);
    motor2.run(BACKWARD);
    motor1.setSpeed(MAX_VELOCIDAD);
    motor2.setSpeed(MAX_VELOCIDAD);
}
//Luces
if (info[2]==(byte)1) {
    digitalWrite(8, HIGH);
}
else{
    digitalWrite(8, LOW);
}
}
}

```

La parte más importante del código es la lectura de la información enviada mediante bluetooth:

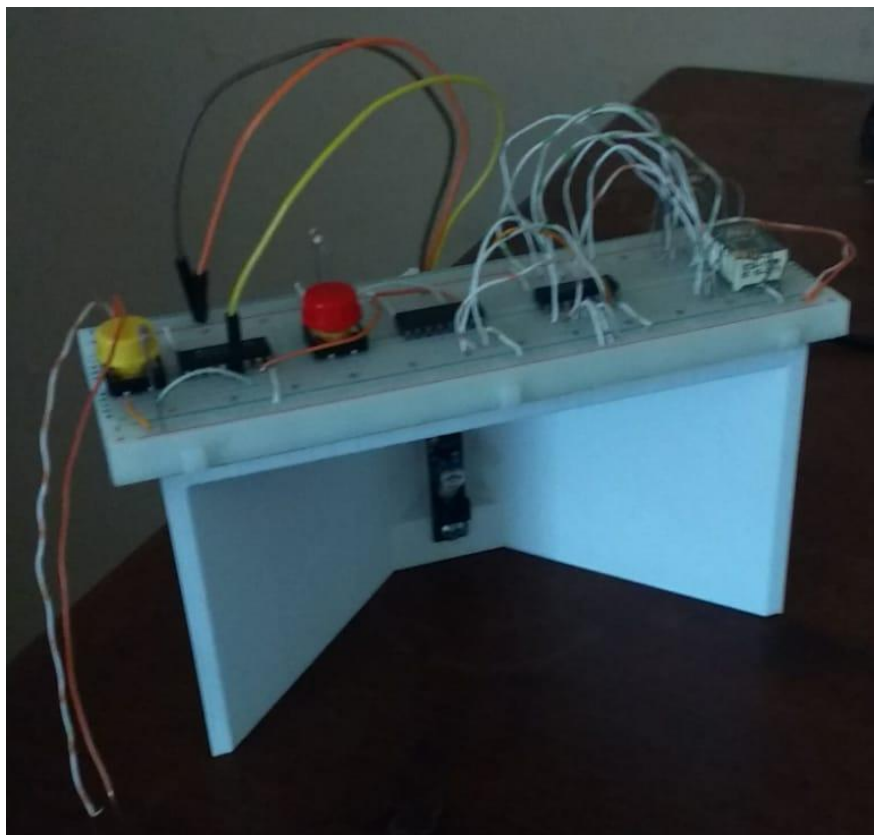
Esta información se envía como una cadena de 3 bytes, como se vio en el código de la aplicación, y se recibe como tal. Luego, con cada valor almacenado en su posición específica (info[0] tiene la información de si se desea ir para adelante, atrás o quedarse quieto; info[1] tiene la información de si se desea girar hacia la derecha, izquierda o no girar; y info[2] tiene la información de si se desea tener las luces del auto prendidas o no).

Los Arcos:

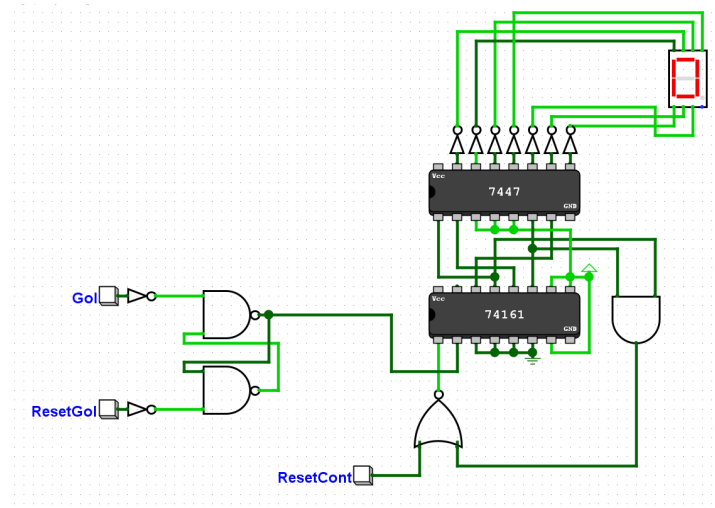
Los arcos fueron diseñados con una forma triangular para guiar la pelota al centro de el, donde esta el sensor:



Una vez armado con todos los componentes, queda así:



El circuito del arco es el siguiente:



Nota: Aca el latch Set-Reset está hecho con dos compuertas NAND, debido a que el software utilizado no tenía el integrado que se utilizó. El contador 74161 fue reemplazado por el cd4040, pero el funcionamiento es el mismo en ambos integrados (con algunas diferencias que en este proyecto no influyen). Y el 7447 es el equivalente al cd4511 que se utilizó finalmente.

El arco funciona de la siguiente manera:

Posee un latch Set-Reset (el 74ls279) al que van conectados el sensor y un pulsador. El sensor hace que la salida del latch se encienda si detecta la pelota y el pulsador resetea el latch. Esto se hizo para que el contador cuente una sola vez a la pelota.

La salida de ese latch se conecta a la entrada de un contador binario (el cd4040). Cada vez que se enciende la salida del latch, el contador suma un uno al valor anterior. El contador también posee un pulsador que resetea el valor de este.

Y las salidas del contador van a un decodificador BCD a 7 Segmentos (el cd4511) que toma en sus entradas los valores del contador y enciende sus salidas según corresponda al número que se quiera mostrar en el display de 7 segmentos.

Conclusión

Este proyecto fue una valiosa oportunidad para poner en práctica los conocimientos de programación adquiridos durante nuestra carrera. La inclusión de componentes como Arduino, sensores y displays agrega un toque técnico. En el caso de los arcos, utilizar circuitos integrados en lugar de un Arduino para la tarea simple de contar goles propuso un desafío interesante. Aunque un Arduino podría haber simplificado esta función, la decisión de optar por circuitos integrados permitió encontrar una solución más directa y eficiente. En resumen, es una combinación divertida de juegos y tecnología, ofreciendo una experiencia única de juego interactivo.