MANUAL TÉCNICO

PROYECTO 1
ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES

Grupo 4

Sección A

ÍNDICE

INTRODUCCIÓN

App Inventor

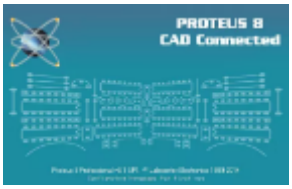
App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación.

Arduino

Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real.

Proteus

Proteus es un potente software comercial de captura de esquemas, simulación y auto-ruteado, capaz de simular en tiempo real sistemas completos basados en microcontrolador.



Arduino IDE

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware.

Los programas de arduino están compuestos por un solo fichero con extensión “ino”, aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

FUNCIONAMIENTO DEL PROGRAMA

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

El funcionamiento incluye la estructura básica del sketch con los métodos setup() y loop().

setup() es llamado cuando empieza la ejecución Se usa para incializar variables, pin modes, librerías, etc. Este solo se ejecutará una vez, cada que se reinicie o encienda la placa de Arduino.

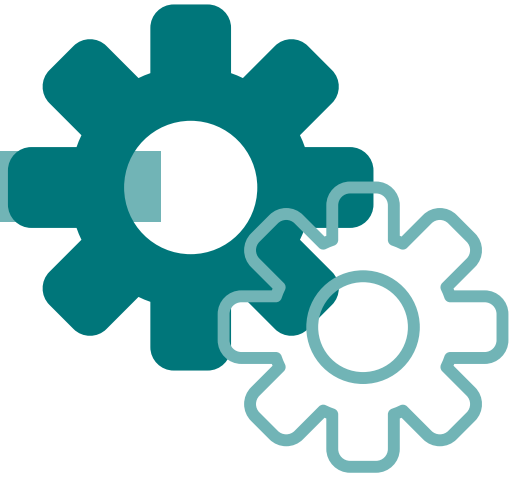
loop() se ejecuta después de setup() y repite en ciclo el código que contenga permitiendo programar cambios y respuestas .

setup() y loop() son parte de la estructura del sketch.

VARIABLES Y OBJETOS

```
long duracion;
long distancia; // long distancia para obtener decimales
int motor11=2;
int motor12=3;
int motor21=4;
int motor22=5;
int motoratras1=6;
int motoratras2=7;
int stoplight=8;
int derecha=9;
int izquierda=10;
int detenerse=11;
int echo=12;
int trig=13;
int right=23;
int left=25;
int alta = 256;
int baja = 50;
int actual=150;
```

SETUP()



```
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    setMotores();
}
```

Para el método setup se inicializan dos puertos seriales del Arduino a 9600 baudios, para que esto pueda hacerse debe ser un modelo de Arduino que tenga varias salidas de comunicación serial. Luego de esto se llama al método setMotores() que establece todos los pines necesarios como salida.

LOOP()

```
void loop() {
    VerificarMeta();
    delay(500);
}
void VerificarMeta(){
  int distancia = sensorUS(trig,echo);
  if(distancia<3){
    stopcar();
    Serial.println("Llego a la meta!!!");
    Serial1.println("Llego a la meta!!!");
  }else{
    //opciones();
    opciones1(distancia);
    Serial1.print("Distancia: ");
    Serial1.print(distancia);
    Serial1.println(" cm");
  }
}
```

El loop solo llama al metodo VerificarMeta y luego coloca un retraso de medio segundo.

El método VerificarMeta declara e inicializa la variable distancia y le asigna el valor que retorna la función sensorUS este por medio de calculos y los pulsos del sensor calcula la distancia. Dependiendo de la distancia este realiza la serie de opciones disponibles para el dispositivo o lo detiene.

Se escriben mensajes en terminal y salida serial.

OTRASFUNCIONES()

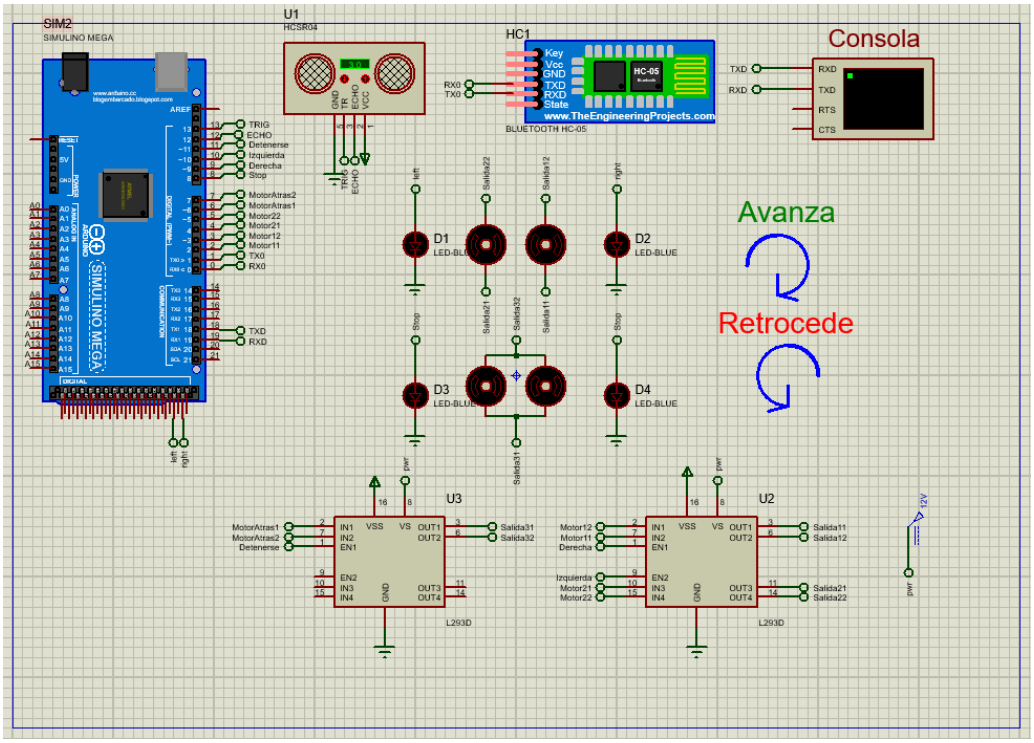
```
void opciones(){
   String algo="";
   while(Serial.available()>0){
      algo +=(char)Serial.read();
   }
   if(algo!=""){
      Serial1.println(algo);
      if(algo=="1"){//adelante Las llantas giran
         adelante();
      }else if(algo=="2"){//reversa-> ajustar Giro con todas la llantas
         reversa();
      }else if(algo=="3"){//izquierda-> parar llanta derecha
         irIzquierda();
      }else if(algo=="4"){//derecha-> detener llanta izquierda
         irDerecha();
      }else if(algo=="5"){//alta velocidad--> aumento en velocidad
         vAlta();
      }else if(algo=="6"){//velocidad Baja--> bajar velocidad
         vBaja();
      }else if(algo=="7"){//Detenerse--> digitalWrite(pin,LOW);
         stopcar();
      }
  }
  delay(50);
}
```

El método opciones define una cadena de caracteres mientras lee la entrada al pin serial y agrega ese dato a la cadena.
Si la cadena no es vacía selecciona entre las opciones disponibles. que configuran los movimientos de los motores.

```
void adelante(){
  analogWrite(derecha, actual);
  analogWrite(izquierda, actual);
  analogWrite(detenerse, actual);
  digitalWrite(stoplight, HIGH);
  digitalWrite(left,HIGH);
  digitalWrite(right,HIGH);
  digitalWrite(motor11,HIGH);
  digitalWrite(motor12,LOW);
  digitalWrite(motor21,HIGH);
  digitalWrite(motor22,LOW);
  digitalWrite(motoratras1,HIGH);
  digitalWrite(motoratras2,LOW);
  Serial1.print("Avanzando velocidad
actual:  ");
  Serial1.print(actual);
  Serial1.println(" pwm");
}
```

El método adelante y los otros que se llaman desde las opciones son combinaciones de digitalWrite para los pines de salida alternando estados altos y bajos y analogWrite con valores diferentes para modular la velocidad de los motores.

CIRCUITO



El circuito simulado en Proteus utiliza la librería Simulino para tener acceso al microcontrolador y que permita carga el binario generado a partir del código programado.

Para las conexiones se usan los terminales por defecto con etiquetas que actúan como puentes directos hacia sus pares.

A continuación se listan los componentes que vemos en el diagrama.

Simulino Mega

L293D

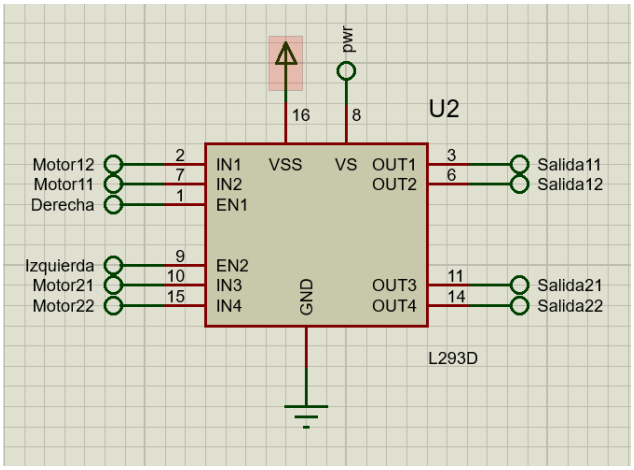
Motores DC

Led azul

Módulo bluetooth HC-05

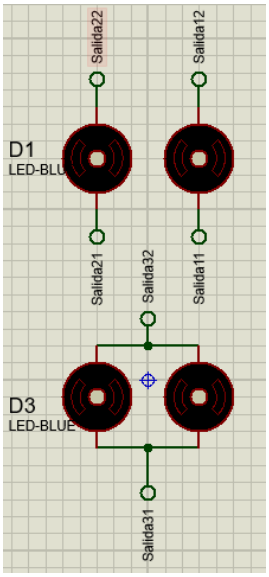
Sensor Ultrasónico

Consola

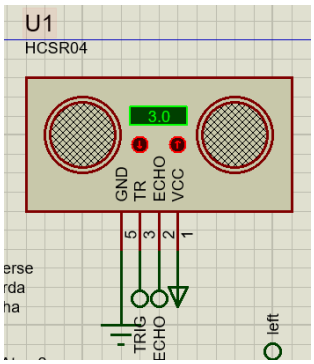


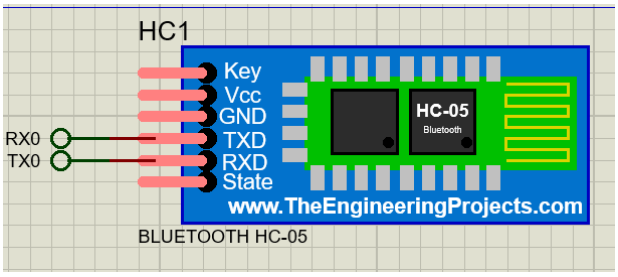
Puente H L293D permite controlar el giro y la velocidad de 2 motores DC, esto con dos entradas digitales del arduino una alimentacion de 12 v para los motores y una de 5 para la circuitería interna. El pin enable es el que nos permite la modulacion por ancho de pulso. Se utilizan dos, uno para dos motores delanteros y otro para los dos traseros.

Las salidas de los L293D van hacia los motores que controlan el giro



Sensor ultrasónico, tiene un emisor y un receptor que por medio de señales ultrasónicas mide el tiempo en el que viaja de ida y vuelta. Para lo cual tiene el pin de trigger o gatillo y el echo. El tiempo en que el echo se encuentre en estado alto es el tiempo en que la señal viaja.

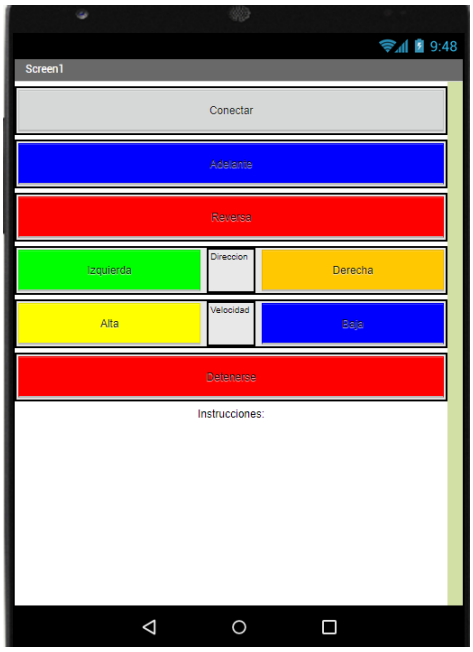


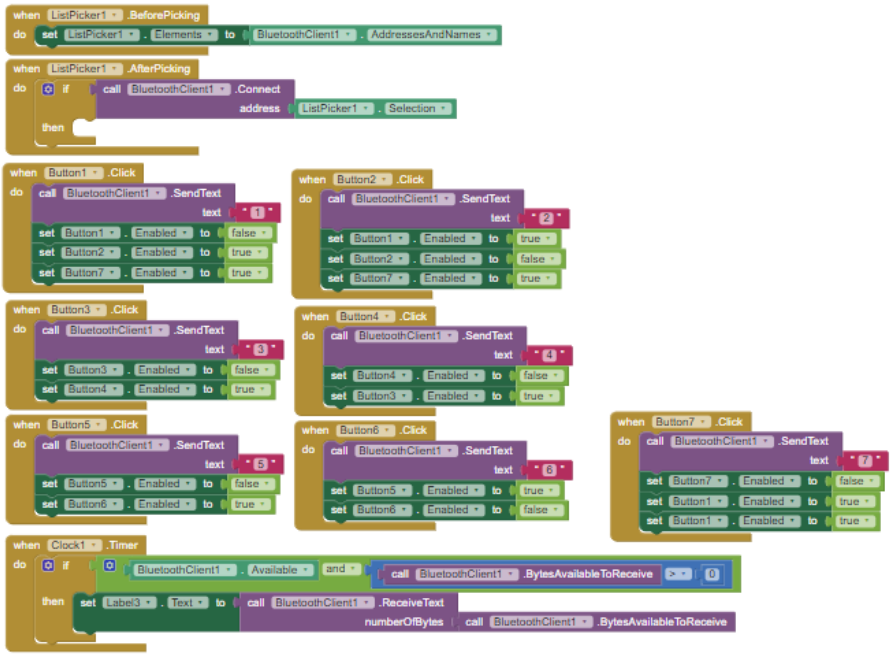


Modulo HC-05 permite la comunicación a través de señales electrómagneticas en el rango de las microondas codificadas para comunicarse con otros dispositivos. Envía las señales seriales que definamos y recibe aquellas con las que esté sincronizado.

APLICACION ANDROID

Con appInventor se desarrolló esta sencilla aplicación que cuenta con un listpicker, 7 botones, 3 etiquetas, un cliente bluetooth y un reloj.





Con una programación a través de bloques visuales de código, se sigue una lógica de eventos que nos permite conectarnos, enviar y recibir los mensajes entre el celular y la placa arduino. En este caso la emulación en computadora.

El listpicker nos permite seleccionar entre las direcciones bluetooth en nuestro telefono y luego se asigna.

Una vez conectado podemos, dependiendo del botón pulsado, enviar un número hacia la placa que hará cambios en los motores y led.

Dependiendo de la distancia en el sensor ultrasónico recibiremos un mensaje desde el dispositivo Arduino si se ha llegado o no a la meta.