

Proyecto Carfox



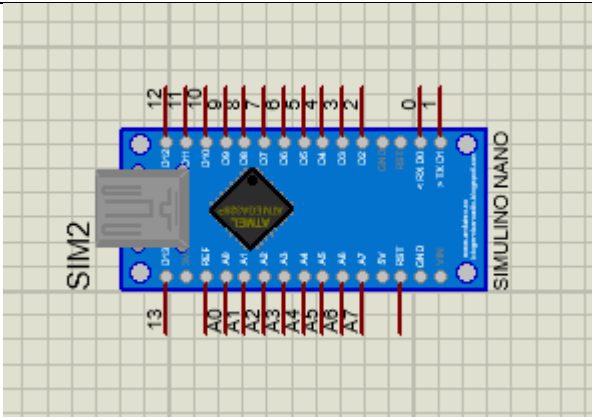
Carfox es un dispositivo diseñado para encontrar a todos los que necesiten ayuda en situaciones de emergencia, en específico su objetivo es ayudar a ubicar a personas que sufran accidentes automovilísticos y así los servicios de emergencia vayan a socorrer a los heridos del accidente, todo esto en el menor tiempo posible. Todo esto con ayuda de una placa de Arduino especializada para el internet de las cosas (Arduino MRK Fox 1200) y gracias al uso de las redes 0G, junto con el servicio de Sigfox, transmitir la información tanto del accidente como la ubicación donde tuvo lugar el accidente. Cabe mencionar que para poder realizar esto fue necesario en particular contar con dos componentes que nos ayudaría a identificar este tipo de eventos como lo son el IMU de 9 grados de libertad que nos permite medir los cambios de las aceleraciones en sus 3 ejes, además de permitirnos medir la temperatura y un flexiforce que nos permite identificar un posible choque automovilístico.

Propuesta del Grupo

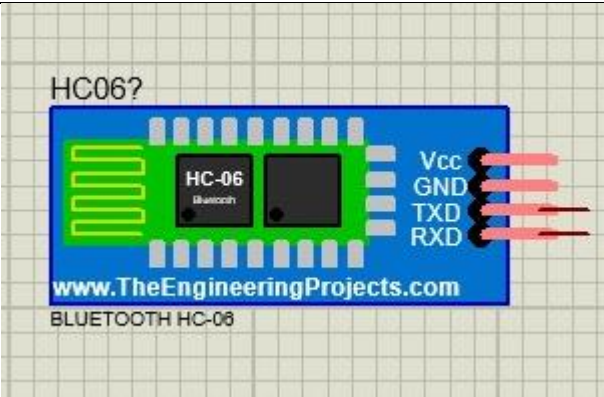
Como propuesta alternativa y debido a las limitaciones de las limitaciones, se tomaron en cuenta las siguientes consideraciones para realizar la simulación del proyecto como lo fueron el tipo de Arduino y el IMU de 9 grados de libertad en este caso eran la esencia del proyecto, efecto de esto se decidió usar otras componentes que emulen sus mediciones como por ejemplos potenciómetros y un sensor de temperatura. Así mismo, tomando en cuenta las dimensiones del dispositivo se tomó como una alternativa razonable el uso del Arduino nano para optimizar las dimensiones del diseño final. Sin embargo, emular la transmisión de información representaría una dificultad a la hora de realizar la simulación, motivo por el cual se propusieron alternativas para representar este envío de datos que en si es la esencia del proyecto Carfox.

En consecuencia, la solución mas viable resulto ser un modulo bluetooth HC-06 (un dispositivo esclavo) que nos permitiría tanto enviar como recibir información, logrando cumplir con la ide del proyecto con ayuda de una aplicación móvil que nos permita recibir la información transmitida por medio del Arduino a través del modulo bluetooth.

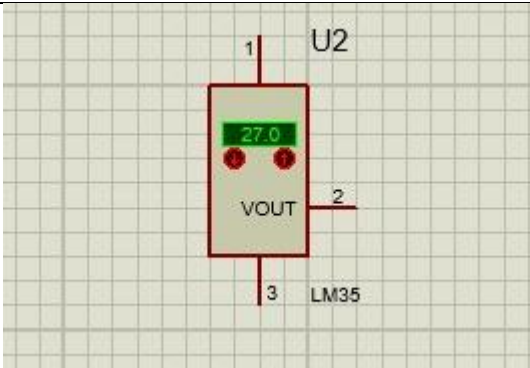
Componentes usadas en la simulación



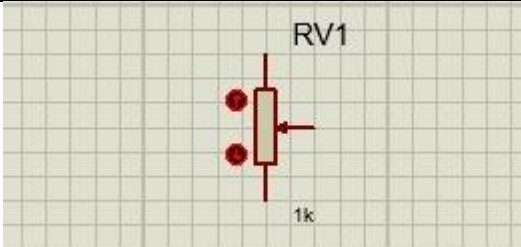
Arduino Nano



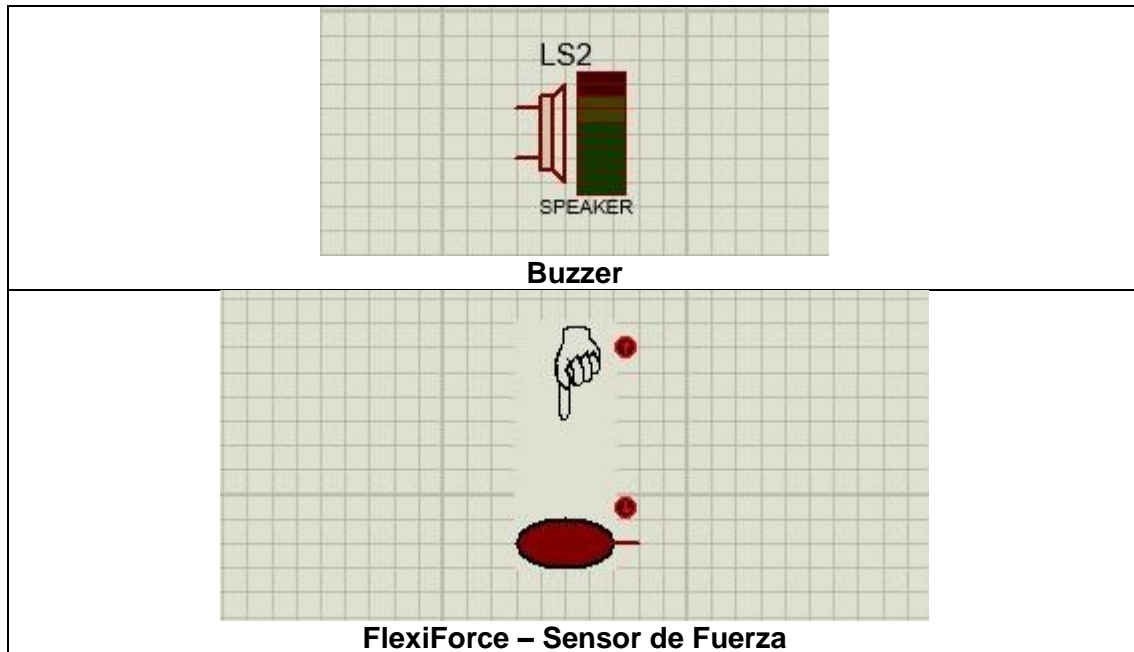
Modulo Bluetooth HC-06



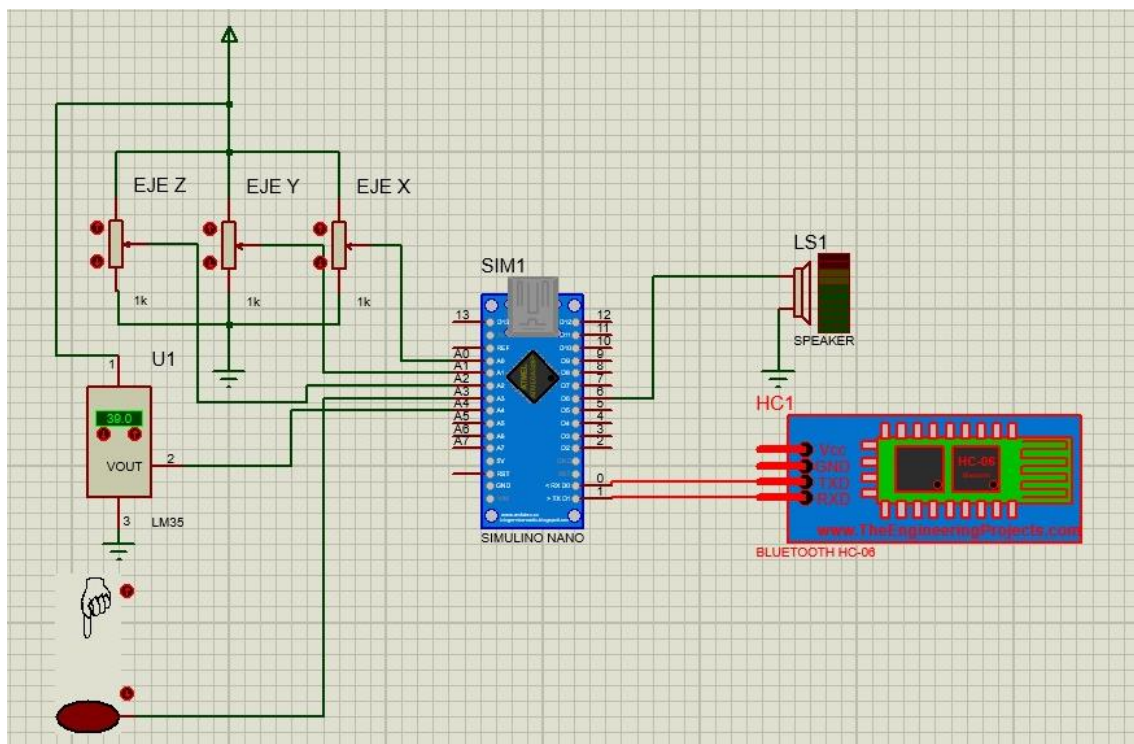
Sensor de Temperatura (LM35)



Potenciómetro



Esquema del Circuito



Programación

```
//Variables y valores definidos de las conexiones en el Arduino
#define temperatura A4
#define sensorFuerza A3
#define buzzer 6
#define ejex A0
#define ejey A1
#define ejez A2
float Ax, Ay, Az;
float temp;
int presion;
//Parametros limitantes
int aceleracionLimite = 70;
int limiteCaidaLibre = 50;
int presionLimite = 50;
int temperaturaLimite = 40;
//Mensaje de estado
String estado = "Estable";

void setup() {
  Serial.begin(9600); //Iniciamos comunicación Serial
}

void loop() {
  //RELizamos un mapeo de los valores del potenciómetro para simular las
  aceleraciones
  Ax = map(analogRead(ejex), 0, 1023, -100, 100);
  Ay = map(analogRead(ejey), 0, 1023, -100, 100);
  Az = map(analogRead(ejez), 0, 1023, -100, 100);

  if (abs(Ax) > aceleracionLimite || abs(Ay) > aceleracionLimite || abs(Az) >
  aceleracionLimite) { //Verificamos si se excede la velocidad limite
    estado = "Exceso de velocidad"; //Cambia el estado del mensaje
    //Activamos el zumbador
    analogWrite(buzzer, 100);
    delay(100);
    analogWrite(buzzer, 0);
    delay(200);
  }
  else{
    analogWrite(buzzer, 0); //EL zumbador permanece inactivo
  }

  float caidaLibre = abs(Az);
  if (caidaLibre > limiteCaidaLibre && abs(Ax) < 70 && abs(Ay) < 70) {
  //Verificamos si se encunetr en caida libre
    estado = "Caida Libre"; //Cambia el estado del mensaje
    //Activamos el zumbador
```

```

    analogWrite(buzzer,500);
    delay(100);
    analogWrite(buzzer, 0);
    delay(200);
}

else{
    analogWrite(buzzer, 0); //EL zumbador permanece inactivo
}

if (Az < 0 && (abs(Ax) > 40 || abs(Ay) > 40)) { //Detecta si ha ocurrido un
volcamiento
    estado = "Volcamiento"; //Cambia el estado del mensaje
    //Activamos el zumbador
    analogWrite(buzzer, 300);
    delay(100);
    analogWrite(buzzer, 0);
    delay(200);
}

else{
    analogWrite(buzzer, 0); //EL zumbador permanece inactivo
}

presion = map(analogRead(sensorFuerza), 0, 205, 0, 100); //Mapeo de los
rangos del sensor de fuerza
if (presion > presionLimite) {
    estado = "Choque!!!"; //Cambia el estado del mensaje
    //Activamos el zumbador
    analogWrite(buzzer, 900);
    delay(100);
    analogWrite(buzzer, 0);
    delay(200);
}

else{
    analogWrite(buzzer, 0); //EL zumbador permanece inactivo
}

temp = (analogRead(temperatura) / 10240.0) * 5000.0;
if (temp > temperaturaLimite) {
    estado = "Incendio!!!"; //Cambia el estado del mensaje
    //Activamos el zumbador
    analogWrite(buzzer, 900);
    delay(100);
    analogWrite(buzzer, 0);
    delay(200);
}

```

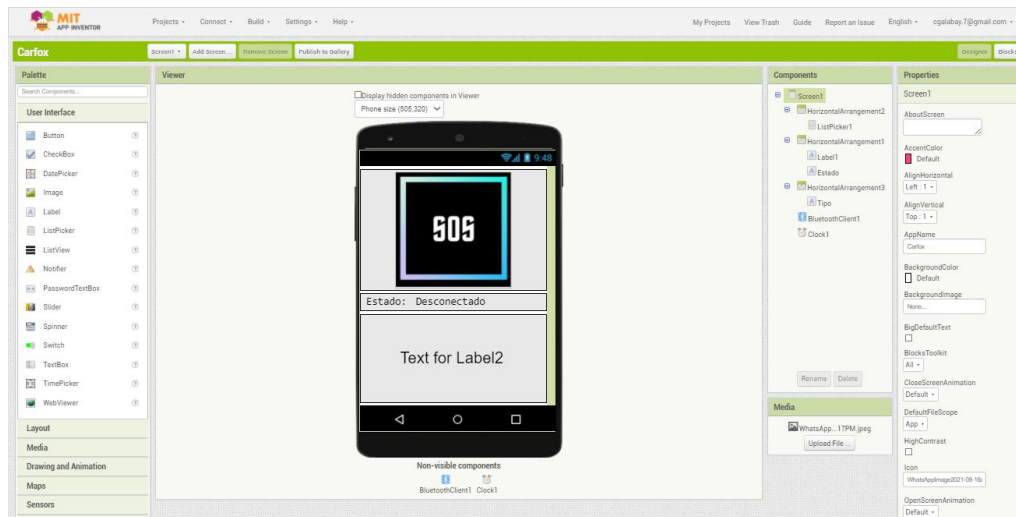
```

else{
  analogWrite(buzzer, 0); //EL zumbador permanece inactivo
}

Serial.print(estado); //Se envia la información
delay(400); //Tiempo de espera entre cada envio de información
}

```

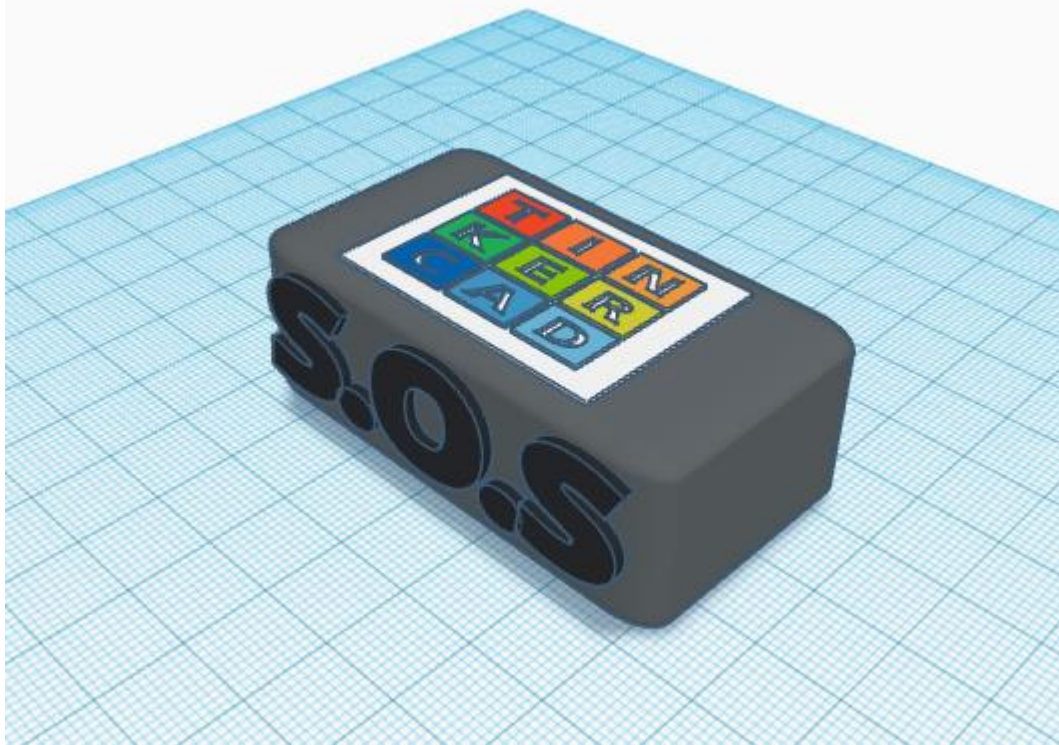
Diseño de la aplicación móvil



Programación de la aplicación

	<p>Iniciamos la aplicación móvil con la etiqueta de tipo con el texto “Estable”.</p>
	<p>Iniciamos una variable global info_recivida = “0”.</p>
	<p>Al darle clic a la imagen del logo se muestran la lista de dispositivos bluetooth disponibles.</p>
	<p>Después conecta el dispositivo bluetooth que se selecciono de la lista de dispositivos y si la conexión fue exitosa se cambia la etiqueta a “Conectado” y el color de su texto cambiará a verde.</p>
	<p>Usando el reloj de la aplicación para controlar la recepción de información, verificamos la conexión, seguido comprobamos si se ha recibido información para finalmente mostrar en la etiqueta “Tipo” y mostrar el tipo de accidente según la información recibida del Arduino.</p>

Diseño 3D



Enlace del diseño 3D en Tinkercad:

https://www.tinkercad.com/things/0LRTIfMKffp-logo-proyecto-carfox/edit?sharecode=OIHleF441rv6VEd4KDxsR_meQn8QxoxIzOHhEAG8Mkk