

Alarma de Mochila Antirrobo

1. Introducción

Este proyecto se puede encontrar en:

https://create.arduino.cc/projecthub/tctree333/backpack-alarm-ffcd5b?ref=tag&ref_id=alarm&offset=1



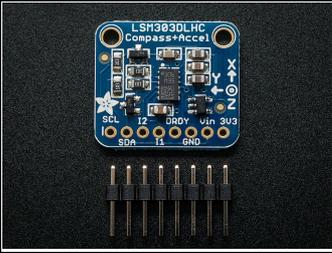
Se trata de una alarma de mochila antirrobo. Su dueño activa la alarma si va a dejar la mochila desatendida. Si un intruso mueve la mochila, suena un timbre bastante llamativo que lo ahuyenta.

Se basa en el compás/acelerómetro LSM303, que es un dispositivo que utiliza una brújula para controlar la orientación. Cuando hay un movimiento de más de 10 grados se hace saltar la alarma.

2. Componentes y materiales



Arduino UNO & Genuino UNO

	<p>Acelerómetro de triple eje Adafruit + Magnetómetro (Brújula) - LSM303</p>
	<p>Alarma piezo grande parkFun - 3kHz</p>
	<p>batería y/o pila genérica (3,7 V o 5 V dependiendo del modelo de arduino usado)</p>

3. Instalación, código y montaje.

3.1.Instalar las bibliotecas

Descargue la biblioteca Adafruit_LSM303DLHC y la biblioteca Adafruit_Sensor como archivos zip.
Abre el IDE de Arduino. Si no lo tienes, descárgalo en *Sketch>Include Library> Add.ZIP Library ...*

3.2. Código del programa en Arduino

```

/* Incluimos las librerías necesarias */
#include<Wire.h>
#include <Adafruit_LSM303_U.h>
#include <Adafruit_LSM303.h>
#include <Adafruit_Sensor.h>

/* Ajustamos la sensibilidad del sensor */
const int sens = 10;

/* Definimos las variables y pines de entrada/salida del arduino */
const int switchstate = 0;
const int buzzer = 13;
const int pulsador = 8;

/* Asignamos un único identificador ID único a este sensor al mismo tiempo */
Adafruit_LSM303_Mag_Unified mag = Adafruit_LSM303_Mag_Unified(12345);

void setup(void)

```

```

{

/* JComienzo del programa */
Serial.begin(9600);
Serial.println("Starting...");

/* Definimos pines de salida y entrada */
pinMode(buzzer, OUTPUT);
pinMode(pulsador, INPUT);

/* Inicializamos el sensor */
if(!mag.begin())
{
/* Existe un problema al detectar el sensor LSM303 ... comprobar las conexiones*/
tone(buzzer,4000);
delay(500);
noTone(buzzer);
delay(500);
tone(buzzer,4000);
delay(500);
noTone(buzzer);
delay(500);

while(1);
}

/* Esperamos 5 segundos */
delay(5000);

/* Alerta cuando la alarma se activa */
tone(buzzer,4000);
delay(500);
noTone(buzzer);

}

/* Función para obtener el valor del sensor*/
intgetDeg(void){

/* Obtenemos un evento del sensor*/
sensors_event_t event;
mag.getEvent(&event);
float Pi = 3.14159;

/* Calculamos el ángulo del sensor en el vector x,y */
float heading = int((atan2(event.magnetic.y,event.magnetic.x) * 180) / Pi);
/* Normalize to 0-360 */
if (heading < 0)
{
heading = 360 + heading;
}
returnheading;
}

/* Nos metemos en el loop del programa*/
voidloop(void)
{

/* Recogemos el valor del pulsador para apagar la alarma si está encendida*/
switchstate= digitalRead(pulsador);

/* Tomamos el valor actual del sensor */
intoldDeg = getDeg();
delay(1000);
intnewDeg = getDeg();
}
}

```

```

/* Si los ángulos son diferentes al anterior estado del sensor pita porque la mochila se ha movido*/
if (newDeg< (oldDeg-sens) &&oldDeg != 0 &&newDeg != 0) {
  /* Suenan la alarma */
  //digitalWrite(13, HIGH);
  tone(buzzer,4000);
  Serial.println("Triggered");
  Serial.println("");
}else if (newDeg> (oldDeg+sens) &&oldDeg!= 0 &&newDeg != 0) {
  /* suena la alarma*/
  tone(buzzer,4000);
  //digitalWrite(13, HIGH);
  Serial.println("Triggered");
  Serial.println("");
}

/* apagamos la alarma si pulsamos el pulsador */
if (switchstate==HIGH)
{

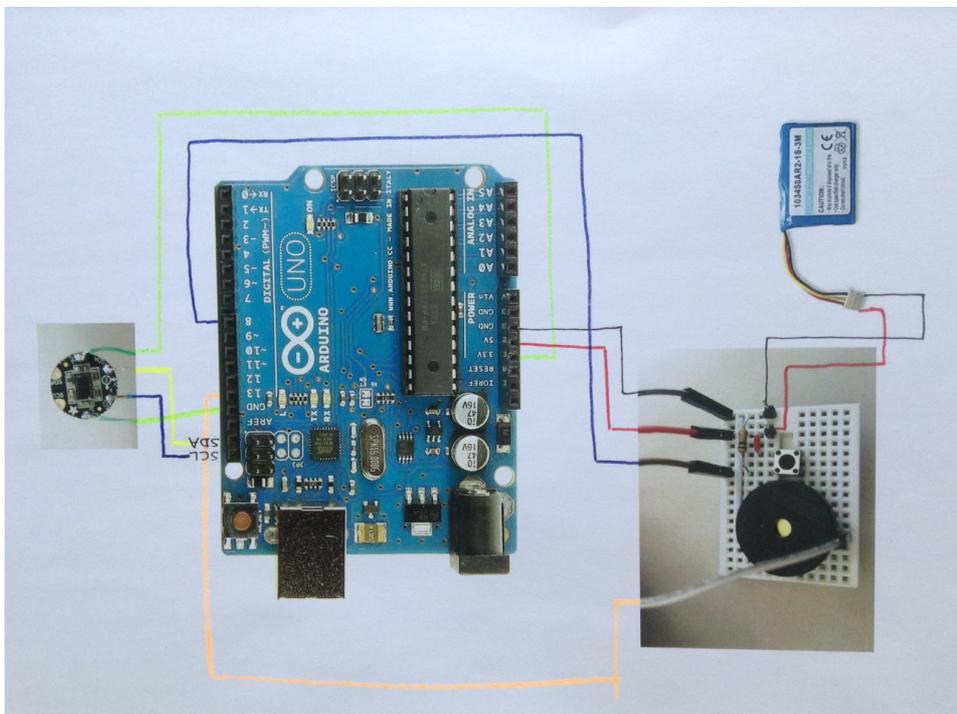
Serial.println("Apagamos la alarma");
noTone(buzzer);
switchstate==LOW;
}

}

```

3.3. Implementación del montaje

Las conexiones finales entre el arduino y la placa queda de la siguiente manera:



4. Funcionamiento del prototipo final

Dejamos desconectada la pila para que no se descargue. Para empezar la demostración conectamos la pila. Después de unos instantes de silencio se oye un pitido, lo que indica que la alarma está funcionando. Si a partir de ese momento la alarma se gira o se mueve se escuchará la alarma, ya que se activará a registrarse un movimiento.

Para desactivarla hay que presionar durante un segundo aproximadamente el pulsador negro.

Si se quiere se puede ajustar la variable 'sens' en el código: si el sensor es demasiado sensible, incremente la variable 'sens' o si se quiere más sensibilidad, disminuya la variable 'sens'.

5. Mejoras añadiendo un componente de Bluetooth

Podemos integrar un módulo de Bluetooth al sistema con la finalidad de que la alarma sea por ejemplo controlada por un teléfono. El componente, denominado HC-05 o HC-06 tiene un aspecto como este:



Por otro lado, para que la alarma pueda ser controlada usando este componente Bluetooth usando un Smartphone Android, tenemos que bajarnos algún programa del market que sea compatible, tal como el "Arduino Bluetooth Controller" que lo podrán encontrar en el siguiente enlace:

<https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor&hl=es&gl=US>

A continuación, siga los siguientes pasos:

1. Abra la aplicación cuando finalice la descarga y busque el módulo en la lista de dispositivos Bluetooth (debe tener el módulo de bluetooth de su dispositivo móvil encendido para que lo detecte y pueda conectarse).
2. Toque HC-05 (módulo Bluetooth Arduino) y si solicita la contraseña o el código pin, escriba 0000 o 1234
3. Seleccione el "Modo de cambio".
4. Vaya a las configuraciones y toque el icono "On" para configurar el "mensaje" que se enviará como "1" (ver imagen).
5. Configure el icono "Apagado" en "2" y toque el icono de la esquina superior izquierda para salir de la configuración.

Finalmente, se muestra el código desarrollado para nuestro prototipo de alarma controlada por bluetooth desde un teléfono móvil Android:

```
/* Incluimos las librerías necesarias */  
#include <Wire.h>  
#include <Adafruit_LSM303_U.h>  
#include <Adafruit_LSM303.h>  
#include <Adafruit_Sensor.h>
```

```

/* Ajustamos la sensibilidad del sensor */
const int sens = 10;

/* Definimos las variables y pines de entrada/salida del arduino */
int switchstate = 0;
const int buzzer = 13;
const int pulsador = 8;

int state = 0;

/* Asignamos un único indentificador ID único a este sensor al mismo tiempo */
Adafruit_LSM303_Mag_Unified mag = Adafruit_LSM303_Mag_Unified(12345);

void setup(void)
{

/* JComienzo del programa */
Serial.begin(9600);
Serial.println("Starting...");

/* Definimos pines de salida y entrada */
pinMode(buzzer, OUTPUT);
pinMode(pulsador, INPUT);

/* Inicializamos el sensor */
if(!mag.begin())
{
/* Existe un problema al detectar el sensor LSM303 ... comprobar las conexiones*/
tone(buzzer,4000);
delay(500);
noTone(buzzer);
delay(500);
tone(buzzer,4000);
delay(500);
noTone(buzzer);
delay(500);

while(1);
}

/* Función para obtener el valor del sensor*/
int getDeg(void){

/* Obtenemos un evento del sensor*/
sensors_event_t event;
mag.getEvent(&event);
float Pi = 3.14159;

/* Calculamos el ángulo del sensor en el vector x,y */
float heading = int((atan2(event.magnetic.y,event.magnetic.x) * 180) / Pi);
/* Normalize to 0-360 */
if (heading < 0)
{
heading = 360 + heading;
}
return heading;
}

/* Nos metemos en el loop del programa*/
void loop(void)
{

```

```

/* Recogemos el valor del pulsador para apagar la alarma si está encendida*/
//switchstate= digitalRead(pulsador);

//añadido ahora para BLUETHOOTH
if(Serial.available()){
    state = Serial.read();
    Serial.println(state);

} // esta parte del código es para solo 1 Carácter o Unidad.

//si la alarma está apaga
if (state == '0') {

    //Serial.println("Alarma apagada");
    noTone(buzzer);
}
// de lo contrario si el estado es 1 ese sería Encendido "ON"
else
if (state == '1') {

    //Serial.println("Alarma encendida");
    /* Tomamos el valor actual del sensor */
    int oldDeg = getDeg();

    //inicialmente el delay estaba a 1000 y ahora lo pongo a 500
    delay(500);

    int newDeg = getDeg();
    //Serial.print("newDeg:");
    //Serial.println(newDeg);
    //Serial.print("oldDeg:");
    //Serial.println(oldDeg);
    /* Si los ángulos son diferentes al anterior estado del sensor pita porque la mochila se ha movido*/
    if (newDeg < (oldDeg-sens) && oldDeg != 0 && newDeg != 0) {
        /* Suena la alarma */

        tone(buzzer,4000);
        //Serial.println("Triggered");
        Serial.println("");
    }else if (newDeg > (oldDeg+sens) && oldDeg!= 0 && newDeg != 0) {
        /* suena la alarma*/
        tone(buzzer,4000);
        //Serial.println("Triggered");
        //Serial.println("");
    }

}

} //final del if

}

```