ROAD SAFETY ENHANCER

SYSTEM FOR A BETTER RIDING EXPERIENCE FOR MOTORCYCLISTS

This guide shows how to build a motorcyclist's blind spot detection device in an accessible way, with alternatives for adapting it to each motorcycle and user.

Components ≈ 53€

- ESP32 Devkit ≈ 12€
- Adafruit Neopixel strip ≈ 16€ (min. 6 LEDs)
- Powerbank ≈ 12€
- 3x HC-SR04 sensors ≈ 6€
- 3x 1kΩ resistors
- 3x resistors 2.2kΩ
- 1x 470Ω resistor
- 1x capacitor 1000uF 6V

  } ≈ 2€
- Jumper wires ≈ 3€
- 1x Breadboard ≈ 2€

Tools

- Soldering iron
- Small star key
- 3D printer

Materials

To solder the components:

- Solder wire
- Perforated plate
- Soldering paste
- Desoldering mesh
- Screws
- Heat shrink sleeve

Alternatives

- The HC-SR04 sensors can be replaced with waterproof JSN-SR04T sensors.
- The powerbank can be replaced with batteries, as long as a voltage regulator is added to protect the components.
- The device can be powered by the motorcycle's 12V battery, if the necessary changes are made so that the system is stable with voltages of this magnitude.

All the files needed to build the device are available in this **repository**.

1) Circuit

It is recommended that the circuit is first assembled using a breadboard and jumper cables to facilitate prototyping.
The online prototype of the circuit and its schematic can be found **here**. Due to Tinkercad's limitations, this online prototype is using an Arduino Uno as an example, but this should be replaced by an ESP32 due to the need for bluetooth functionality.
In short, the VCC pin of each sensor is connected to the ESP32's Vin, the GND pins to the ESP32's GND, the TRIG pins directly to the corresponding digital pin on the ESP32 and the ECHO pins must be connected to a 1kΩ resistor connected to the corresponding pin on the ESP32 and a 2.2kΩ resistor connected to GND.
The DIN pin of the LED strip should be connected to a 470Ω resistor, connected to the corresponding pin on the ESP32.
The LED's 5V pin must be connected to the capacitor's anode (positive), connected to the ESP32's Vin, while the GND pin must be connected to the capacitor's cathode, connected to the ESP's GND.

2) Programming

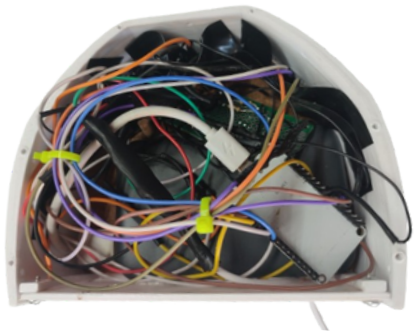
The project's .ino file is available in the “code” folder of the **repository**. All the code is explained in comments within the file. It is recommended to use the ”**Arduino IDE**" to upload the code to the micro controller, given that this software installs all the necessary libraries automatically. You may only need to **install the ESP32 drivers** for the board to be read by the computer.

3) Printing

All the project's .obj files are available in the **repository**'s “3d printing” folder. They can be printed on various materials without affecting the structural rigidity of the box.

4) Assembly

It is recommended that most of the components be soldered to prevent disconnection while driving. It is also essential that all welded points are insulated with a heat-shrink sleeve.
It is optional, but recommended, to add high-temperature and vibration-resistant material to the inside of the case to protect the components while driving. Insulating tape can also be added inside the housing, for example behind the sensors, to prevent short circuits due to vibrations.
All the screw slots in the design of the box are designed for screws with a diameter of 3mm and a length of 8mm. There are also oval spaces at the top of the box for the optional addition of wire or thread to reinforce the top of the box.
On the back, there is an oval hole to access the powerbank button to turn the device on and off. If you opt for waterproof sensors, this hole can be covered with tape to prevent water from entering without interrupting access to the button. There is also a small hole for the LED strip cable at the bottom of the back. Any cable can be used to connect to the LED strip, but for testing the device a traditional audio cable was used, which was cut out and soldered to the LED strip.
This is what the device should look like once assembled:



App (only needed for sound signals)

In the **repository** you'll find the app's .apk file in the “android app” folder for installation on android devices. You can also find the .aia file, so that you can make changes to the application code using the **Mit App Inventor**, where you can also find the **project in the gallery**.