# Development of Product

### Library imported

Logotipo

Descripción generada automáticamente

This library allows the Arduino board to control Liquid Crystal displays (LCDs), it is used only when a I2C module is connected to the LED. The functions I will be using are: init( ), backlight( ), setCursor( ) and print ( ).

### LCD initialization



To use this library, the size proportion of the screen is needed, therefore, the function above creates an object from the class LiquidCrystal\_I2C that has parameters to set the LCD address to 0x27 for a 16 chars and 2 line display.

Texto

Descripción generada automáticamente

The objective of these functions is:

|  |  |
| --- | --- |
| **init()** | Initialize the I2C module connected to the LCD |
| **backlight()** | Turns on the background light of the LCD |
| **print()** | Writes text in the LCD, it is similar to a Serial.print function |

### Class Sensor

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

A class named Sensor is created because there are 2 gas sensors and they have similarities that can be shared using a class, also, using this class lead to a more organized code.

The functions created in the class are private because they are going to be used in an overloaded constructor inside the class, so the functions are not needed to Be public.

The objective of these functions is:

|  |  |
| --- | --- |
| **sensorPin()** | Hold the pin the sensor is connected to |
| **ledPin()** | Holds the pin the LCD is connected to |
| **sensorCurrentValue()** | State the current value of the sensor |
| **sensorLimit()** | Define the threshold which the sensors are subjected to |

Texto

Descripción generada automáticamente

An overloaded constructor is created, this overloaded constructor will have *sensorPin, ledPin* and *sensorLimit* as parameters.

The overloaded constructor will help creating the sensors in an easier manner. I decided to use an overloaded constructor instead of a default constructor because I needed to use parameters and a default constructor doesn’t deal with parameters.

When assigning the values for each gas sensor it will look like this:

Texto

Descripción generada automáticamente

### Monitor()

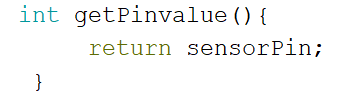
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteA function named Monitor is created in the class, this function has a boolean as the return type and will read the sensor current value and compare it with the sensor limit, when the sensor limit is lower than the current value, otherwise, it returns false.

### Accesors

Texto

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### sensoritos[ ]



An array named sensoritos is created, the main advantage of creating this array is that when iterating the sensors, I don’t need to put them individually, leading to a more simplified code.

Sensoritos is used because there is very little difference between the sensors. Why write the exact same code twice when you can iterate over a data structure and call the same function for each element?

Texto

Descripción generada automáticamente

This array is used for initializing the leds in a for loop instead of initializing the leds separately, the loop iterates the items store in the array and uses an accessor from the class called getLedPin (returns ledPin) to configures the specified pin to behave as an output using the pinMode() function.

## Loop for toggling on the leds and the buzzer

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

The objective of this loop is to toggle the leds and the buzzer when the threshold of the sensor value is surpassed.

It accomplishes this by using the tone function, this function will make the buzzer work when receiving the pin, the frequency, and the duration. After the buzzer starts making sound, the led will turn on and off using the digitalWrite function, enabling and disabling the getLedPin accessor.

### Display the sensors values and limits

For the LCD to display the sensor concentrations and thresholds, it will use the setCursor function to state the position in which the first and second concentration will be displayed. To display the concentrations, two accessors from the class will be used, the readData accessor return the sensor pin in which the corresponding gas sensor is connected and the getSensorLimit accessor returns the sensor limit.

Texto, Carta

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente