



Universidad  
Nacional  
de Quilmes

Julio - 2019  
Departamento de Ciencia y Tecnología

# **Detector de Alcholemia**

CONTI AUGUSTO  
VIGO FACUNDO

Introducción a la programación de  
microcontroladores con Arduino

Profesor Jose Luis Di Biase

UNIVERSIDAD NACIONAL DE QUILMES

## Descripción

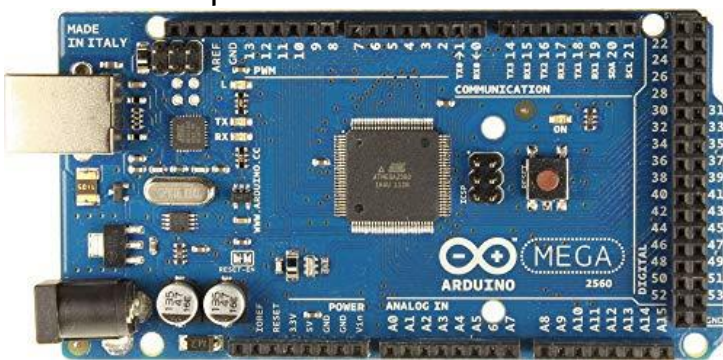
En el proyecto fuente que utilizamos, solo muestra la conexión con el sensor y como obtener los datos para mostrarlos en pantalla. Decidimos ir un poco mas allá, e implementar un botón que de inicio al análisis, mientras suena un pitido, para luego mostrar el resultado en pantalla con un mensaje personalizado, tratando de hacerlo mas parecido a un detector de verdad.

## Disclaimer

Este detector de alcoholemia no es lo suficientemente preciso, y no debe ser utilizado para medir el nivel de intoxicación real. ¡Si tomó, no maneje!

## Materiales

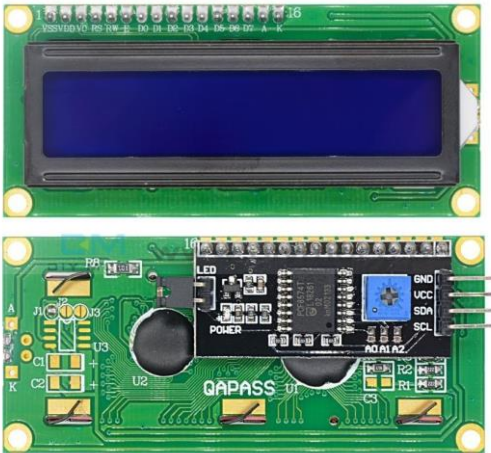
- Arduino Mega \$0  
Propiedad de Facundo.



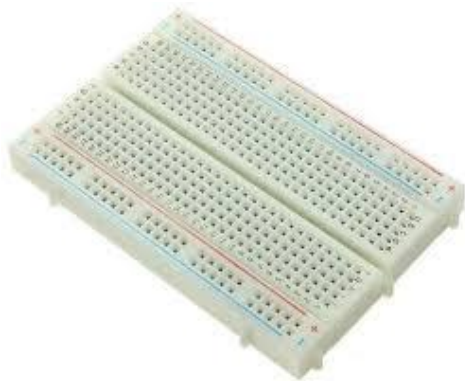
- MQ-3 (Sensor de alcohol) \$180  
Comprado en Sysport



- LCD display 16x2 \$265  
Comprado en Sysport



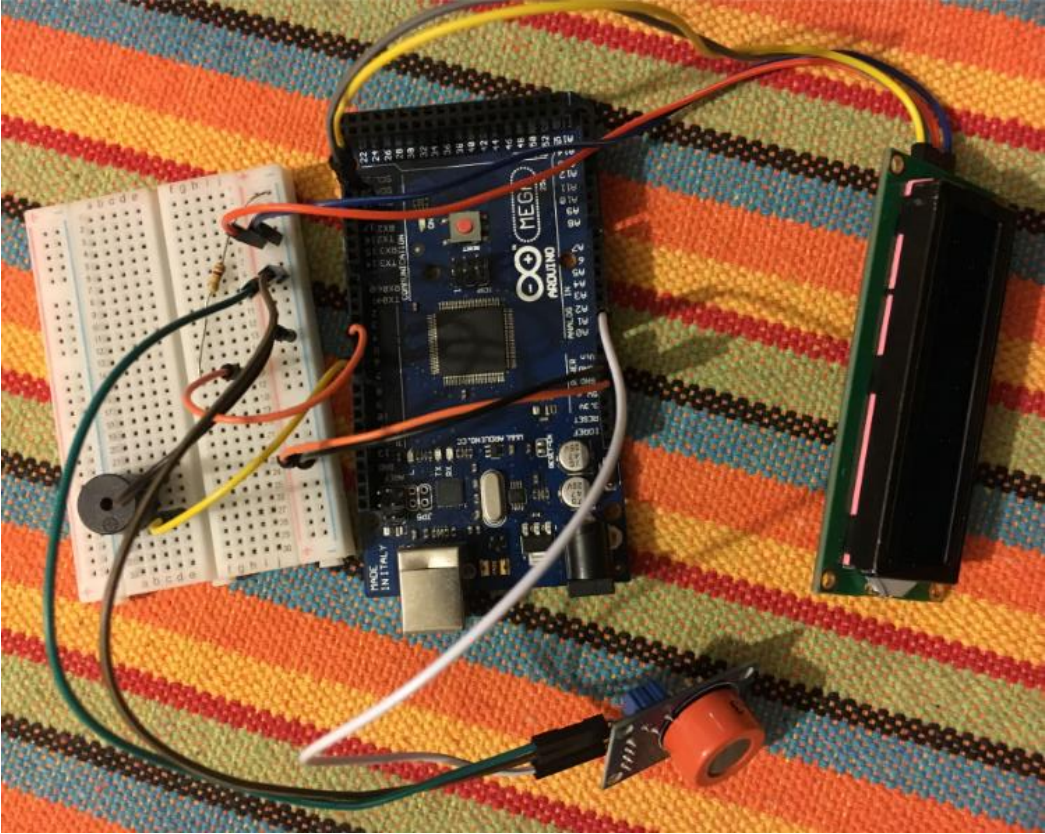
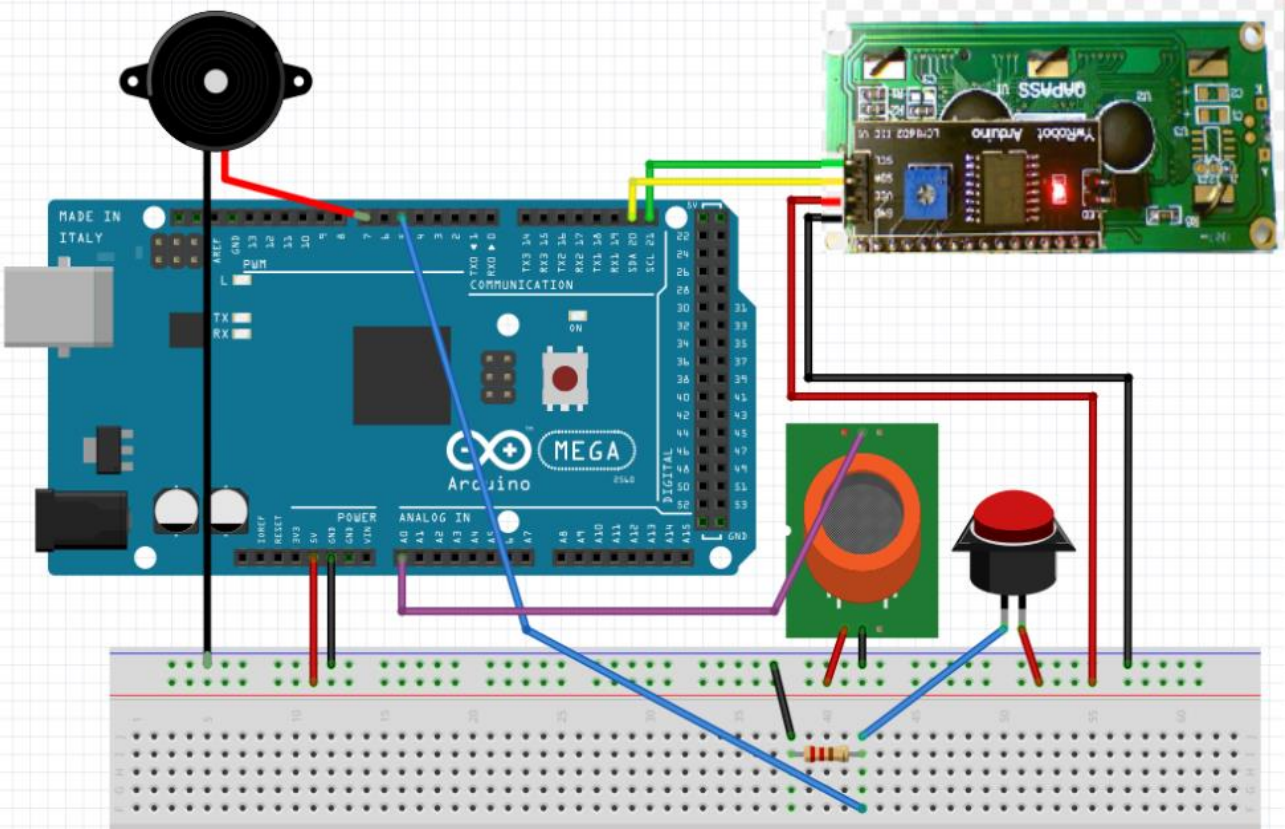
- Protoboard \$120  
Comprado en Sysport



- Pulsador
- Buzzer
- Resistencia 10kOhm
- Cables  
\$120  
Comprado en Electrónica RC



Circuito



## Resultado final



## Código fuente

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int TIME_UNTIL_WARMUP = 20;
unsigned long time;

int TIME_UNTIL_MEASURE = 5;
unsigned long measurement_start;

int analogPin = A0;
int val = 0;

const int buzzerPin = 7;
const int buttonPin = 5;

bool measurement_mode = false;
bool measurement_done = false;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  Serial.begin(115200);
```

```

    lcd.init();
    lcd.backlight();
}

void loop() {
    int button_state = digitalRead(buttonPin);

    if (button_state && !measurement_mode) {
        measurement_mode = true;
        measurement_start = millis()/1000;
        measurement_done = false;
    }

    delay(100);
    time = millis()/1000;
    if(time<=TIME_UNTIL_WARMUP)
    {
        int progress_time = map(time, 0, TIME_UNTIL_WARMUP, 0, 100);
        printWarming(progress_time);
    }
    else
    {
        if (!measurement_mode && !measurement_done) {
            printPress();
        }
        if (measurement_mode && !measurement_done) {
            printMeasure();
            tone(buzzerPin, 1000);
            val = readAlcohol();
        }
        if (measurement_mode && !measurement_done && ((time - measurement_start)>
TIME_UNTIL_MEASURE)){
            noTone(buzzerPin);
            measurement_mode = false;
            measurement_done = true;
            lcd.clear();
        }
        if(measurement_done) {
            printAlcohol(val);
            printAlcoholLevel(val);
        }
    }
}

void printWarming(int progress)
{
    lcd.setCursor(0,0);
    lcd.print("Calentando: ");
    lcd.print(progress);
    lcd.print("%");
}

void printPress()
{
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("Presiona");
    lcd.setCursor(2,1);
    lcd.print("para iniciar");
}

```

```

}

void printMeasure()
{
  lcd.setCursor(0,0);
  lcd.print("Sopla hasta que");
  lcd.setCursor(0,1);
  lcd.print("pare el sonido");
}

void printAlcohol(int value)
{
  lcd.setCursor(0,0);
  lcd.print("Sensor: ");
  lcd.print(val);
}

void printAlcoholLevel(int value)
{
  lcd.setCursor(0,1);
  if(value<200)
  {
    lcd.print("SuperSobrio");
  }
  if (value>=200 && value<280)
  {
    lcd.print("Una birrita");
  }
  if (value>=280 && value<350)
  {
    lcd.print("Un par de birras");
  }
  if (value>=350 && value <450)
  {
    lcd.print("Nivel P.Bulrich!");
  }
  if(value>450)
  {
    lcd.print("MODO CHANO!");
  }
}

int readAlcohol() {
  int value = 0;
  int val1;
  int val2;
  int val3;

  val1 = analogRead(analogPin);
  delay(10);
  val2 = analogRead(analogPin);
  delay(10);
  val3 = analogRead(analogPin);

  value = (val1+val2+val3)/3;
  return value;
}

```

## Problemas

Perdimos mucho tiempo al tratar de hacer funcional el display 16x2, instalamos mal la librería LiquidCrystal I2C, hasta que logramos hacerlo funcionar con la librería de “Frank de Brabander” v1.1.2

LiquidCrystal I2C by Frank de Brabander Version 1.1.2 **INSTALLED**

**A library for I2C LCD displays.** The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.

[More info](#)

Select version

## Fuentes – links

<http://www.electronics-lab.com/project/arduino-breathalyzer-using-mq3-gas-sensor-oled-display/>

<https://sysport.mercadoshops.com.ar/>

<http://electronicarc.net/blog/>