

# The Ramylizer

*Not Exactly* a Summer Camp Activity (but you get the idea)

Dr. Samuel Bechara  
Assistant Professor of Practice  
Mechanical Engineering  
Colorado State University

## Objective

This guide will walk you through the steps necessary to construct your own breathalyzer and is modelled after activities that students do during Summer Camps that we have created. The arduino is already pre-loaded with the software necessary but usually, participating students in the summer camps would program the microcontrollers themselves.

## Disclaimer

You guys are smart. Clearly this isn't a super accurate device. Don't sue us if you use it, blow  $< 0.08$  and then get a DUI. It *should* be moderately accurate but not accurate enough to bet your drivers license on.

## Interested in More?

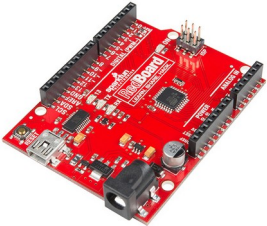


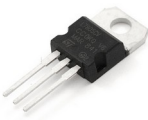
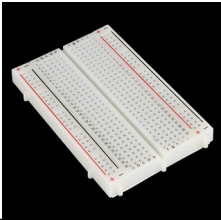
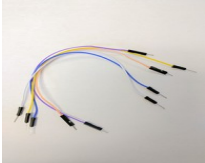
Interested in learning more about this project? Want to make it with your friends? I have provided all of the code, diagrams, and order list for you online. You can find all the materials at my github page:

<https://github.com/sbechara/ramylizer>

The parts list can be found here:

[https://www.sparkfun.com/wish\\_lists/154135](https://www.sparkfun.com/wish_lists/154135)

## Materials

Sparkfun Redboard (Arduino Microcontroller)		This is the “brain” of the whole project and is a very versatile and powerful computer! It can receive and/or send data in a number of different ways. In this activity we are going to concentrate on the analog output from the gas sensor but the capabilities of this are only limited by your imagination!
MQ-3 Alcohol Gas Sensor		This is the main sensor that we will use to detect alcohol. It has a high sensitivity and a fast response time. Sensor provides an analog resistive output based on alcohol concentration. The drive circuit is very simple, all it needs is one resistor. A simple interface could be a 0-3.3V ADC.
4-Digit 7-Segment Display		These types of displays are neat but usually VERY difficult to program. Well, this one solves that problem by being a little “smarter” and having a microprocessor on board! That means you can simply connect this to an arduino using three wires and get nice pretty numbers! We will use it to display the estimated BAC value.
Voltage Regulator <i>*Optional kind of</i>		This is a basic voltage regulator. It will take our battery output and make sure that our sensors see as close to 5V as possible. This will help with the accuracy of our calculation of BAC.
Breadboard		This is a very common device used as a prototyping helper. It allows us to make electrical connections easily.
Male-to-male jumper wires		These are just helpful wires to help us make electrical connections from all of the devices.  The re

## Part 1: Background – Basic Electronics

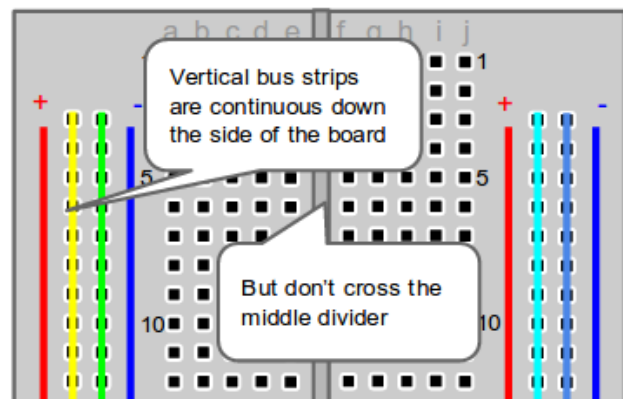
Although it is out of the scope of this project, it is a good idea for you to get a basic knowledge of how electricity works and the physical properties that govern the movement of charge through wires. <http://www.instructables.com/id/Basic-Electronics/> is a great website that can help you get started.

The three main concepts that you need for today are to understand that:

### 1) In order to work, circuits need to be powered

Just like water won't move unless it is on a hill, charge won't move unless there is a potential difference. Our arduino has the ability to power devices that require both 3.3V and 5V sources. To do this, we will use the vertical bus strips on the breadboard to easily connect to our power source. Since our gas sensor requires a 5V source, we will use one jumper wire to connect the 5V power source on the arduino to the + vertical bus and another jumper wire to connect ground from the arduino (GND) to the – vertical bus. This will provide the power we need for our circuit.

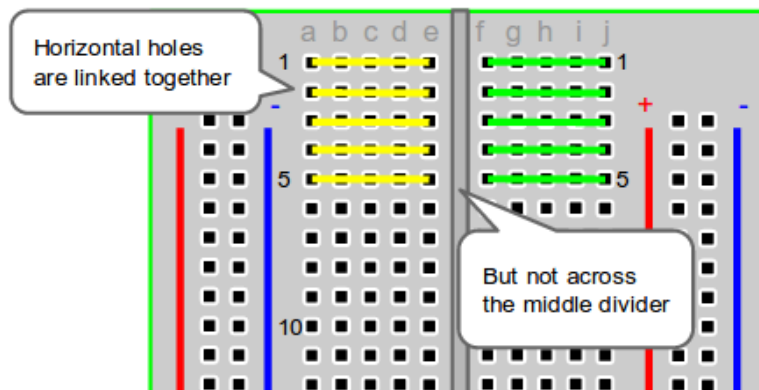
In the example we will do today, this is technically unnecessary, we could connect directly to the arduino to the sensor with one jumper wire. But it is good form if you decide to add more sensors in the future!



## 2) In order to work, circuits need to be closed

The jumper wires and the breadboard allow us to close circuits. The breadboard has plates that run along the numbered rows but do not cross the middle divider. This allows us to use jumper wires to make connections from the arduino, to the gas sensor, without *directly* connecting a wire to each.

*If you are having a problem, it is likely that you have an open circuit.*



## 3) Do not create a short circuit

DON'T DO THIS, but if you connect a wire directly from the positive to the negative side of a power supply, you'll create what is called a short circuit. This is a very bad idea. If in doubt, ask if the connection you are making is a short circuit.

## Part 2: Wiring

Now that we understand how a breadboard works it is time to wire up your breathalyzer! We will utilize the breadboard to make connections. Normally, you would be free to wire up the sensor to *any* of the analog pins and wire the LCD display to *any* of the digital pins BUT since you do not have the ability to program the Arduino (unless you have a spare laptop on you) you **must follow the diagram exactly...**

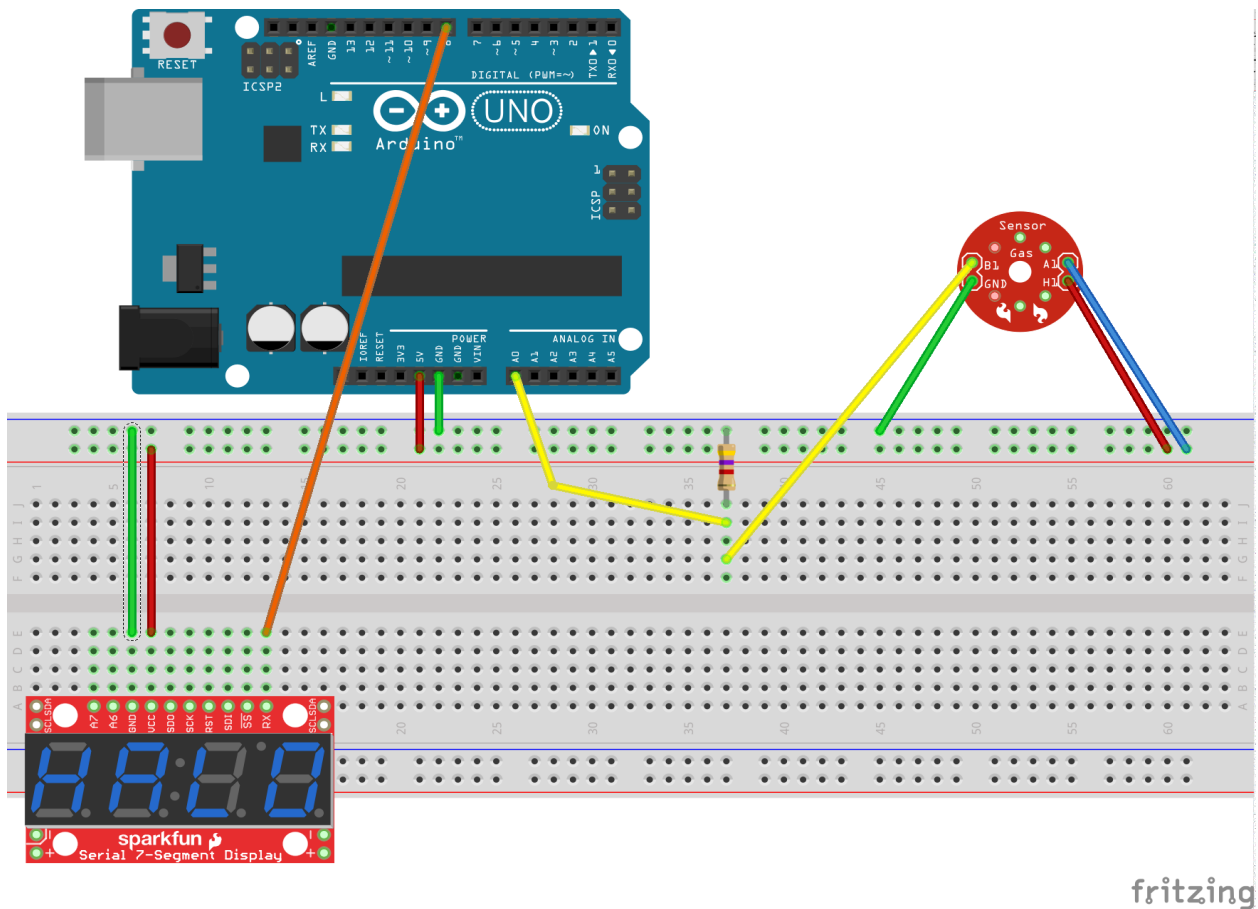


Figure 1: Fritzing diagram of arduino setup

If you find it difficult to understand the diagram, you can use this table to help as well.

Arduino Pin	Breadboard	Gas Sensor	LCD
+5V	+ Power Rail		
GND	- Power Rail		
Digital 8			RX
	+ Power Rail		VCC
	- Power Rail		GND
	+ Power Rail	A1	
	+ Power Rail	H1	
	- Power Rail	GND	
Analog 0	- Power Rail via the 4.7k $\Omega$ resistor	B1	

Note: You should notice that we are not using the 5V voltage regulator included in your kit. I ordered 5 (because I am short sighted) and during my testing found that 1 of them didn't work. I am also dumb and didn't mark which one didn't work before I threw them in the kits. Finally, I didn't have enough time to order more. However, I found that the Arduino was able to power the gas sensor reasonably well without the regulator so it should be ok...

## Part 3: Programming

If this was a normal summer camp activity we would talk about programming concepts and how the students need to program the breathalyzer. Since we do not have time or computers on hand, I'll just show you the code that I wrote to program these devices. The code is already pre-loaded onto each Arduino.

```
#include <SoftwareSerial.h>

// ramylizer_withLCD.ino
// Created by: Samuel Bechara, PhD
// Created on: 21 October 2019

// These are the Arduino pins required to create a software serial
// instance. We'll actually only use the TX pin.
const int softwareTx = 8;
const int softwareRx = 7;

// Pins and Initialization
const int analogPin = 0;
float sensorReading = 0;
float V_rl = 0;
float ppm = 0;
float bac = 0;

SoftwareSerial s7s(softwareRx, softwareTx);

void setup() {
  Serial.begin(9600);
  s7s.begin(9600);
  s7s.write(0x76);

  // Subfunction to set decimal place on LCD
  setDecimals(0b00000001);
}

void loop() {
  // Get sensor value
  sensorReading = analogRead(analogPin);

  // Convert to sensor voltage
  V_rl = sensorReading * 5 / 1024;

  // Calculate PPM based off of curve on spec sheet
  ppm = 4.62986*exp(1.05313*V_rl);

  // Calculate BAC
  bac = ppm/2600;

  // Sensor floats around a little bit...this is a cheat
```

```
// Only register BAC if > 0.01
if (bac<=0.01){
    bac = 0.00;
}

s7s.print(bac);
// Serial print commands for testing purposes
// Serial.println(sensorReading);
// Serial.println(V_rl);
// Serial.println(ppm);
// Serial.println(bac);
delay(500);
}

// Turn on any, none, or all of the decimals.
// The six lowest bits in the decimals parameter sets a decimal
// (or colon, or apostrophe) on or off. A 1 indicates on, 0 off.
// [MSB] (X)(X)(Apos)(Colon)(Digit 4)(Digit 3)(Digit2)(Digit1)
void setDecimals(byte decimals)
{
    s7s.write(0x77);
    s7s.write(decimals);
}
```

## Part 4: Powering and Testing (*Prost!*)

Once you have double checked all your connections you are ready to power up and test your breathalyzer! All you have to do is insert a 9V battery into the battery connector, then insert the barrel plug adapter into the Arduino. The LCD screen should light up. You are ready to test your device!

### Troubleshooting / Hints / Miscellaneous:

- If you do not notice the LCD screen light up right away, you have a wiring problem. Call me over and I'll help get things settled.
- I also provided conical vials that you can put the gas sensor into. However, during my testing (yes it was fun), I noticed too much condensation and it kind of grossed me out. It should suffice to blow onto the sensors directly.
- The batteries will only last about 20 minutes before burning out. I recommend, plugging the battery in only while measuring BAC and unplugging as soon as you are done.