

Detector de concentração de álcool com base em Arduino

Luan S. Viana¹, Maycon Rafael C. Borba², Lisandro Almeida³

¹IFgoiano Campus Rio Verde-GO – Bacharelado em Ciência da Computação

Abstract. *This article will cover an embedded system For detection of alcohol concentration based on Arduino, DSDetector, this system uses the paradigm of object-oriented programming, using as a motivation the C ++ programming language, made entirely in the Arduino platform. In its Hardware it consists of two input peripherals, MQ-3 Gas Sensor and DS18B20 Temperature Sensor, three output peripherals, Lcd display, Buzzer Beep, Red Led and other components. This system is due to the need to reduce the number of accidents in transit, many of them due to the ingestion of alcohol by drivers.*

Resumo. *Este artigo irá tratar sobre um sistema embarcado para detecção de concentração de álcool baseado no Arduino, o DSDetector, esse sistema utiliza o paradigma de programação orientado a objetos, usando como motivação a linguagem de programação C++, feito totalmente na plataforma Arduino. Em seu Hardware ele é composto por dois periféricos de entrada, Sensor de gás MQ-3 e sensor de Temperatura DS18B20, três periféricos de saída, display Lcd, Buzzer Beep, Led Vermelho e demais componentes. Esse sistema se dá pela necessidade de diminuir o número de acidentes em trânsito, boa parte deles por ingestão de álcool por parte dos motoristas.*

1. Introdução

Junto com o desenvolvimento da economia, mais pessoas têm carros e por isso, mais carros aparecem nas estradas. Muitos motoristas ignoram o perigo de dirigir depois de beber podendo vir a ocasionar acidentes.

No Brasil em 2008 foi empregada a Lei Seca. Esta lei altera a Lei no 9.503, de 23 de setembro de 1997, e a Lei no 9.294, de 15 de julho de 1996, nos termos do 4º do art. 220 da Constituição Federal, para inibir o consumo de bebida alcoólica por condutor de veículo automotor, e dá outras providências.

A Lei Seca impõe uma maior rigorosidade no consumo de álcool por parte de motoristas, sua idéia principal é diminuir o índice de acidentes, com um consumo de quase 0% de álcool permitido, será difícil provocar tantos acidentes fatais, que a cada dia se tornavam mais frequentes.

A forma mais precisa de determinar se o motorista ingeriu ou não bebida alcoólica é a detecção das quantidades de álcool no sangue que pode ser obtido através de exame de sangue, respiração, saliva e urina.

Utilizando um fórmula simples chegamos a concentração de álcool no sangue de pessoa:

$$\text{BAC (em mg/L)} = \text{BrAC(em mg/L)} * \text{FATOR.}$$

Onde

- BAC e a concentração de álcool no sangue.
- BrAC a concentração de álcool na respiração.

O fator, descrito na fórmula, varia de acordo com a temperatura ambiente.

2. Objetivos

O objetivo deste artigo é elaborar um sistema embarcado para detecção de concentração de álcool baseado no Arduino que possa ser implantado em um veículo para que apenas se locomova caso o motorista não esteja sobre efeito de álcool, assim evitando um possível acidente.

Esse sistema deve ser capaz de coletar a concentração de álcool no sangue do motorista através de sua respiração e do ar ambiente, realizando as compensações da temperatura ambiente, e a partir do limite permitido pela Lei Seca que é de 0,3 mg/l de álcool no sangue, definir se o motorista esta ou não hauto a dirigir.

3. Fundamentação teórica

Utilizamos a plataforma *Git*, que é um sistema de controle de versão de arquivos. Através dela podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente para o mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas. O *Github* é um serviço web que oferece diversas funcionalidades extras aplicadas ao *Git*.

Arduino, é uma plataforma de prototipagem eletrônica de hardware, projetada com um microcontrolador Atmel AVR, com suporte de entrada/saída, uma linguagem de programação padrão C/C++, ela utiliza uma interface serial ou USB, para interligar-se ao computador, que é usado para programá-la e interagi-la em tempo real, na (Figura 1) podemos ver sua interface de programação.

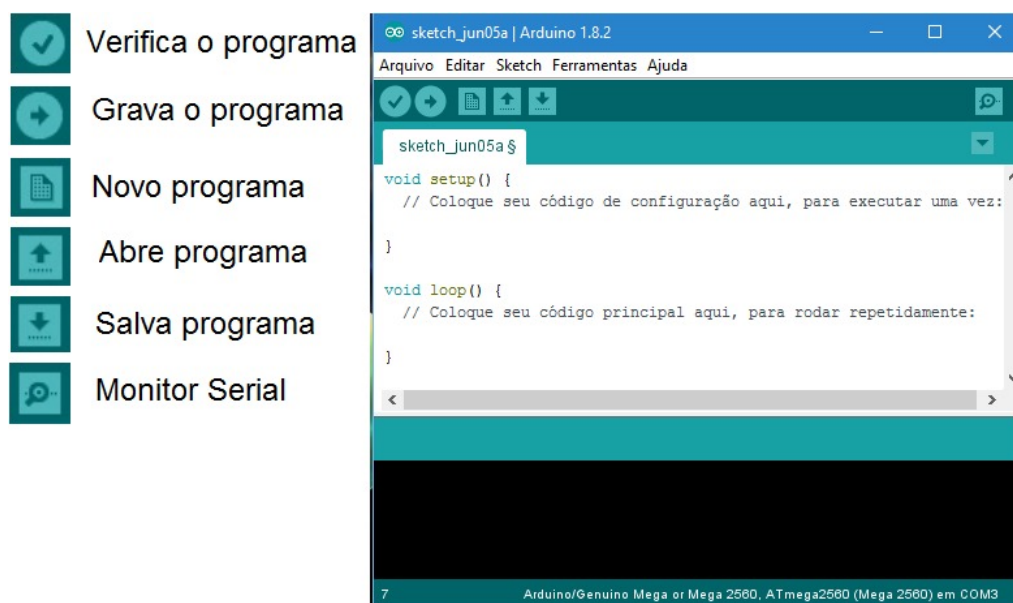


Figura 1. Interface da IDE do Arduino, Arduino IDE 1.8.2.

Na (Figura 2) a placa ATmega 2560 utilizada no projeto e na (Figura 3) suas especificações técnicas.

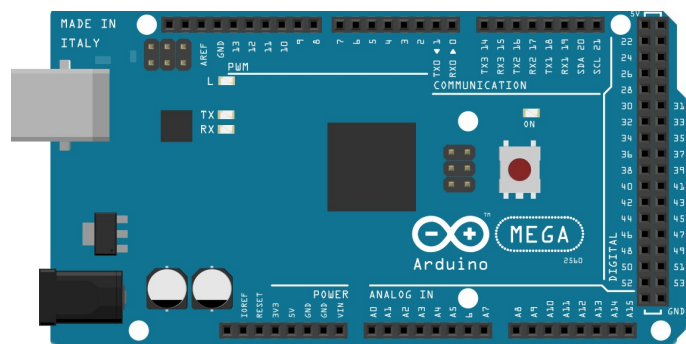


Figura 2. Placa Arduino ATmega 2560.

Especificações técnicas	
Microcontrolador	ATmega2560
Tensão de entrada	5V
Tensão de entrada (recomendada)	7-12V
Tensão de entrada (limite)	6-20V
Pinos I/O digitais	54 (15 saída PWM)
Pinos Analógicos	16
Memória Flash	256 KB
Velocidade do clock	16 MHz

Figura 3. Especificações técnicas da placa.

4. Metodologia

- Produção de um bafômetro controlado pela plataforma Arduino.
- Realização de medições de concentrações de álcool no ar com uma boa aproximação dos valores reais das concentrações.
- Exibir o valor adquirido em um Display de Cristal Líquido. o qual é a interface entre o motorista e o sistema.

5. Drive Safe Detector

O *DSDetector*, um sistema embarcado projetado em C++, utiliza o paradigma de programação orientado a objetos, feito totalmente na plataforma *Arduino*. Ele é composto por dois periféricos de entrada, Sensor de gás *MQ-3* e sensor de Temperatura, três periféricos de saída, Display *Lcd*, *Buzzer Beep* e Led Vermelho, e demais componentes.

Ele é capaz de coletar a concentração de álcool no sangue do motorista através de sua respiração e do ar ambiente, realizando as compensações da temperatura ambiente, a qual necessita a utilização de um sensor de temperatura acoplado ao sistema, e a partir do limite permitido pela Lei Seca que é de 0,3 mg/l de álcool no sangue, definir se o motorista esta ou não haptó a dirigir.

5.1. Esquemático e Diagrama UML do Sistema

Foi feito o esquema do Hardware do sistema assim como mostra a (Figura 4), onde mostra que o sistema estará acoplado à fonte de alimentação da ignição, ou seja, o sistema terá controle sobre a partida do veículo.

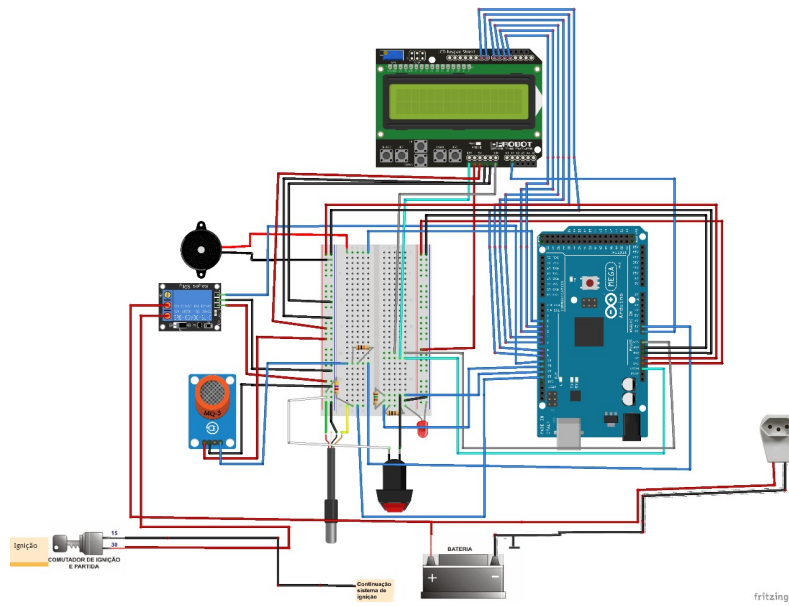


Figura 4. Esquemático do sistema feito a partir da plataforma *Fritzing*.

O diagrama UML do projeto dispõe as classes do projeto e suas relações, padrão tipicamente utilizado na programação orientada a objetos.

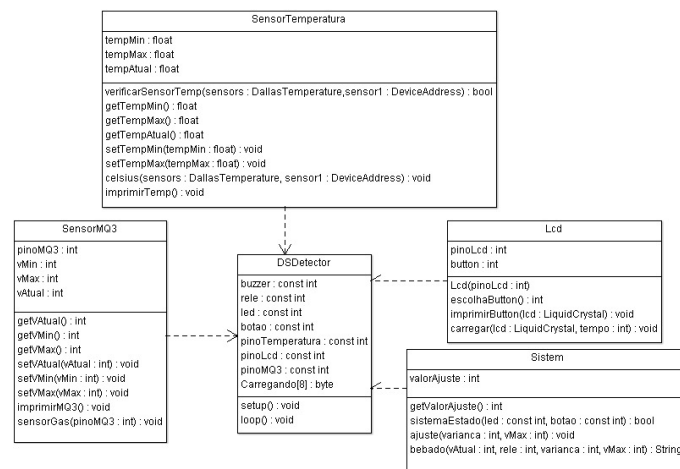


Figura 5. Diagrama UML do *DSDetector*

5.2. Como Funciona o Sistema

- O *DSDetector*, será acoplado a um veículo tendo como fonte de alimentação a bateria do veículo com tensão de 12v a qual será regulada, a partir da montagem do sistema no veículo ele será ligado, ficando em funcionamento por tempo indeterminado, caso o motorista desligue o sistema ele não poderá dar partida no veículo e o sistema acionará o *Buzzer Beep*.
- Assim que ligado, o sistema terá um tempo para entrar em operação, devido ao sensor *MQ-3* que precisa ser aquecido, tempo o qual irá variar de acordo com a temperatura ambiente, após a inicialização o sistema estará em funcionamento total.

- Logo após a inicialização do sistema vai entrar em modo de espera até que o motorista aperte o botão select, então o sistema fará a primeira leitura de concentração de álcool na respiração do motorista e a temperatura ambiente.
- O sistema irá ficar verificando se o motorista ingeriu álcool ou não, caso motorista tenha ingerido álcool o sistema não irá permitir que o motorista ligue o veículo.
- Logo após a primeira verificação o sistema irá rodar em segundo plano, verificando se à álcool no ambiente e lendo a temperatura para ajustar com a detecção de álcool no ambiente.

6. Testes e Resultados

Foram feitas coletas de dados de acordo com os seguintes intervalos de temperatura, temperatura menor ou igual a 19°C, maior que 19°C e menor que 27°C e maior ou igual a 27°C, intervalos os quais foram verificados mudanças na coleta do sensor *MQ-3*.

Podemos observar na (Figura 6) os valores do sensor *MQ-3* de acordo com cada faixa de temperatura, logo após o aquecimento do sensor, o fator irá variar de acordo com os três intervalos de temperatura, a partir dele foi estabelecido um BrAc máximo, numa temperatura abaixo de 19°C o *DSDetector* irá alterar seu código estabelecendo o fator como sendo 0,004 mg/l, ou seja, caso o sensor tenha coletado um BrAc acima de 75ppm o BAC irá ultrapassar o limite permitido pela Lei Seca.

Temperatura °C	BrAc MAX em ppm	BrAc MIN em ppm	Fator em mg/l	BAC mg/l
----- 19°C	----- 75 ppm	10 ppm -----	0,004	0,3 mg/l
19°C ----- 27°C	----- 65 ppm	45 ppm -----	0,004615385	0,3 mg/l
27°C -----	----- 90 ppm	65 ppm -----	0,333333333	0,3 mg/l

Figura 6. Tabela da primeira coleta de dados do sistema.

Depois da detecção de álcool o sensor *MQ-3* leva um certo tempo para restabelecer seus valores normais, então foi adotado um tempo padrão para restabelecimento do sensor de acordo com a temperatura, assim como mostra a (Figura 7).

Temperatura (°C)	Tempo em Segundo	BrAc (ppm)
----- 19	150	79
19 ----- 27	120	84
27 -----	145	104

Figura 7. Tempo de restabelecimento do sensor.

Depois da detecção de álcool no sensor *MQ-3* e a espera para o mesmo se restabelecer o código sofre alteração devido a forma despadronizada para a volta do valor inicial, sendo agora os valores para detecção modificados.

Temperatura °C	BrAc MAX em ppm	BrAc MIN em ppm	Fator em mg/l	BAC mg/l
----- 19°C	----- 79 ppm	10 ppm -----	0,003797468	0,3 mg/l
19°C ----- 27°C	----- 84 ppm	79 ppm -----	0,003571429	0,3 mg/l
27°C -----	----- 104 ppm	84 ppm -----	0,002884615	0,3 mg/l

Figura 8. Tabela da segunda coleta de dados do sistema.

7. Código Principal do Sistema

```
/*
// Programa : DSDetector
// Autor : Maycon Rafael Campos Borba
*/

//-----BIBLIOTECAS-----
#include "SensorTemperatura.h"
#include "SensorMQ3.h"
#include "Sistem.h"
#include "Lcd.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>
#include <NewTone.h>

//-----PINOS-----
const int buzzer = 3;
const int rele = 10;
const int led = 11;
const int botao = 12;
const int pinoTemperatura = 13;
const int pinoLcd = 0;
const int pinoMQ3 = 1;
//-----Caracteres LCD propios-----
byte Carregando[8] = {
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
};

//-----VARIAVEIS-----
int frequenciaBuzzer = 1500;
//-----PONTEIROS-----
SensorTemperatura *sensorTemp;
DSDetector *sistemaMetodos;
SensorMQ3 *sensorMQ;
Lcd *tela;
LiquidCrystal lcd(8, //d1
                  9, //d0
                  4, //d5
```

```

        5 ,//d4
        6 ,//d3
        7 );//d2
OneWire oneWire(pinoTemperatura);
DallasTemperature sensors(&oneWire);
DeviceAddress sensor1;

//-----SETUP-----
void setup(){
    Serial.begin(9600);
    sensors.begin();
    lcd.createChar(0,Carregando);
    lcd.begin(16,2);
    sensorTemp = new SensorTemperatura();
    sistemaMetodos = new DSDetector();
    sensorMQ = new SensorMQ3();
    tela = new Lcd(pinoLcd);
    pinMode(led,OUTPUT);
    pinMode(buzzer,OUTPUT);
    pinMode(rele,OUTPUT);
    pinMode(botao,INPUT);
    pinMode(pinoMQ3,INPUT);
    digitalWrite(rele, HIGH);
    lcd.setCursor(2,0);
    lcd.print("Bem-vindo ao");
    lcd.setCursor(3,1);
    lcd.print("DSDetector");
    delay(2496);
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("AGUARDE...");
    delay(2496);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Aquecendo sensor");
    sensorTemp->celsius(sensors,sensor1);
    if(sensorTemp->getTempAtual()<=19){
        tela->carregar(lcd,7500);//2 min
    }else if(sensorTemp->getTempAtual()>19
and sensorTemp->getTempAtual()<=27){
        tela->carregar(lcd,4375);//1 min 10s
    }else{
        tela->carregar(lcd,9375);
    }
    lcd.setCursor(0,0);
    lcd.print("Para assoprar");

```

```

        lcd.setCursor(0,1);
        lcd.print("botao select");
        if (tela->escolhaButton()==5){
            delay(3000);
            lcd.clear();
            sensorTemp->celsius(sensors,sensor1);
            sensorTemp->imprimirTemp();
            sensorMQ->sensorGas(pinoMQ3);
            sensorMQ->imprimirMQ3();
            lcd.setCursor(0,0);
            lcd.print(" Temperatura:");
            lcd.print((int)sensorTemp->getTempAtual());
            lcd.print(" C");
            lcd.setCursor(0,1);
            lcd.print("MQ3:");
            lcd.print(sensorMQ->getVAtual());
            delay(3000);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(" Resultado:");
            lcd.setCursor(0,1);
            lcd.print(
                sistemaMetodos->bebado(sensorMQ->getVAtual()
                    ,rele,(int)sensorTemp->getTempAtual(),
                    sensorMQ->getVMax()));
            delay(1000);//
        }
    }

//-----LOOP-----
void loop(){
    if(sistemaMetodos->sistemaEstado(led,botao)){
        lcd.clear();
        sensorTemp->celsius(sensors,sensor1);
        sensorMQ->sensorGas(pinoMQ3);
        lcd.setCursor(0,0);
        lcd.print(" Temperatura:");
        lcd.print((int)sensorTemp->getTempAtual());
        lcd.print(" C");
        lcd.setCursor(0,1);
        lcd.print("MQ3:");
        lcd.print(sensorMQ->getVAtual());
        delay(3000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Resultado:");
    }
}

```



```

        lcd.setCursor(0,1);
        lcd.print(
sistemaMetodos->bebado(sensorMQ->getVAtual(), rele,
(int) sensorTemp->getTempAtual(), sensorMQ->getVMax()));
        if((int) sensorTemp->getTempAtual() <= 19){
            delay(150000);
        } else if((int) sensorTemp->getTempAtual() > 19
            and (int) sensorTemp->getTempAtual() < 27){
            delay(120000);
        } else{
            delay(145000);
        }
    } else {
        lcd.clear();
        lcd.setCursor(4,0);
        lcd.print(" Sistema");
        lcd.setCursor(3,1);
        lcd.print(" Desligado!");
        NewTone(buzzer, frequenciaBuzzer);
        digitalWrite(rele, HIGH);
        delay(100);
        noNewTone(buzzer);
        delay(2000);
    }
}
}

```

Referências

- [1] Maxim Integrated, *Programmable Resolution 1-Wire Digital Thermometer*. Disponível em: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [2] Paul Brook, *Repositories GitHub, Arduino One-Wire library*. Disponível em: <https://github.com/pbrook/arduino-onewire>. 16 de outubro de 2012.
- [3] Miles Burton, *Arduino Library for Maxim Temperature Integrated Circuits*. Disponível em: <https://github.com/milesburton/Arduino-Temperature-Control-Library>. 11 de abril de 2017.
- [4] Antony García González, *Sensor MQ-3, un Detector de Alcohol*. Disponível em: <http://panamahitek.com/sensor-mq-3/>. 1 de fevereiro de 2014.
- [5] Arduino e Cia, *Shield LCD 16x2 com Keypad*. Disponível em: <http://www.arduinoecia.com.br/2013/08/arduino-shield-lcd-16x2-com-keypad.html>. 27 de agosto de 2013.
- [6] Guilherme Biff Zarelli, *Arduino Sensor de Gás Detector de Gás (Arduino MQ Gas Sensor)*. Disponível em: <http://helpdev.com.br/2013/07/07/arduino-sensor-de-gas-detector-de-gas-arduino-mq-gas-sensor/>. 7 de julho de 2013.

- [7] ARDUINO e GENUINO PRODUCTS, *Arduino MEGA 2560 e Genuino MEGA 2560*. Disponível em: <https://www.arduino.cc/en/main/arduinoBoardMega2560>.
- [8] Saulo Anderson Bibiano Jardim, *Como Funciona o Sistema de Ignição*. Disponível em: <http://www.ebah.com.br/content/ABAAAA8pgAC/como-funciona-sistema-ignicao>.
- [9] Git, *Git - Controle de Versões Rápidas*. Disponível em: <https://git-scm.com/doc>.
- [10] Tigris.org Open Source Software Engineering Tools, *ArgoUML Documentation*. Disponível em: <http://argouml.tigris.org/documentation/>. 29 de janeiro de 2011.
- [11] GitHub Inc, *The World's Leading Software Sevelopment Platform*. Disponível em: <https://github.com/>.
- [12] Lei Seca, *Lei Nº 11.705, de 19 de junho de 2008*. Disponível em: <http://www.planalto.gov.br/ccivil03/ato2007-2010/2008/lei/111705.htm>. 16 de junho de 2008.
- [13] Inmetro, *Etilômetro (Bafômetro) Controle pelo Inmetro*. Disponível em: <http://www.inmetro.rs.gov.br/etilometro.html>.
- [14] Blog FILIPEFLOP, *Medindo temperatura debaixo d'água com DS18B20*. Disponível em: <http://blog.filipeflop.com/sensores/sensor-de-temperatura-ds18b20-arduino.html>. 2 de junho de 2015.
- [15] Arduino IDE 1.8.2, *Language Reference*. Disponível em: <https://www.arduino.cc/en/Reference/HomePage>.
- [16] Fritzing Eletronics Made Easy, *Fritzing References*. Disponível em: <http://fritzing.org/learning/fullreference>.