

# Detector de concentração de álcool com base em Arduino

Lisandro Almeida, Luan Viana e Maycon Borba

Instituto Federal Goiano - Campus Rio Verde

24 de junho de 2017

# Sumário I

- 1 Introdução
- 2 Objetivo
- 3 Fundamentação teórica
  - Lei N° 11.705
  - Arduino IDE
  - Hardware Arduino
- 4 Git
  - O que é Git?
  - Funcionalidade
- 5 GitHub
- 6 Metodologia
- 7 Drive Safe Detector(DSDetector)



## Sumário II

- Esquema do sistema
- Diagrama UML
- Funcionamento

### 8 Resultados

### 9 Referências Bibliograficas



## Introdução

- Junto com o desenvolvimento da economia, mais pessoas têm carros e por isso, mais carros aparecem nas estradas. Só no ano de 2016 foram 216.262 acidentes dos quais 7.619 foram por ingestão de álcool.

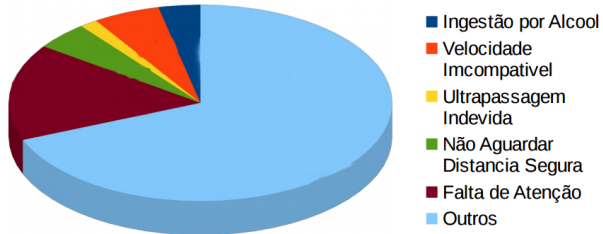


Figura: Total de Acidentes em 2016

## Introdução

- Muitos motoristas ignoram o perigo de dirigir depois de beber. 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 do exame de sangue, respiração, saliva e urina.

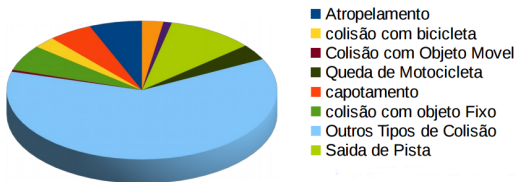


Figura: Tipos dos Acidentes Causados Por Ingestão de Alcool.

# Objetivo

- O objetivo deste trabalho é elaborar um sistema embarcado para detecção de concentração de álcool baseado no Arduino, que futuramente 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.



# Fundamentação teórica

## Fundamentação teórica



## Lei Nº 11.705 (Lei Seca)

- Está 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.





## Punição

- De acordo com a nova legislação, o motorista que for flagrado com nível de álcool acima do permitido (0,1 mg/l de sangue) terá que pagar uma multa de R\$ 2.934,70 e terá a carteira de habilitação suspensa pelo prazo de 12 meses.
- O pior é para quem estiver embriagado (níveis acima de 0,3 mg/l): o motorista corre o risco de ser preso, e a detenção é de 6 meses a 1 ano.



## Tempo para poder dirigir

- Tempo em média o álcool leva para desaparecer do organismo:

Bebida	Tempo em minutos
Um copo de cerveja (350 ml)	60
Uma dose de vinho (150 ml)	85
Uma dose de uísque, tequila ou pinga (50 ml)	75

- A margem de erro do bafômetro, segundo o Inmetro, é de apenas 1 %.



# Arduino IDE



Verifica o programa



Grava o programa



Novo programa



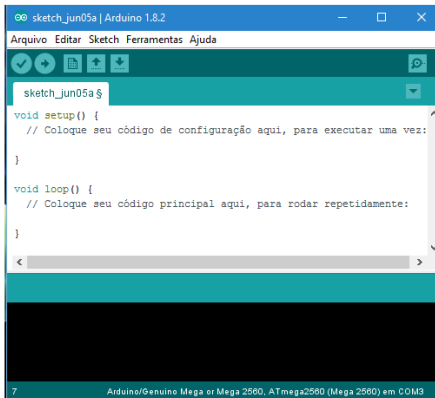
Abre programa



Salva programa



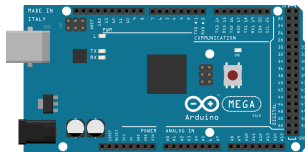
Monitor Serial



# Hardware Arduino

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: Arduino.cc



# Git



## O que é Git?

- Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.



## Funcionalidade

- Permitir que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes. Por mais complexo que isso seja, ele tenta manter tudo em ordem para evitar problemas para nós desenvolvedores.



# GitHub

- O Github é um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Além disso, quase todos os projetos/ frameworks/ bibliotecas sobre desenvolvimento open source estão no github.



<https://github.com/mrafaelcb/DSDetector>



## 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;

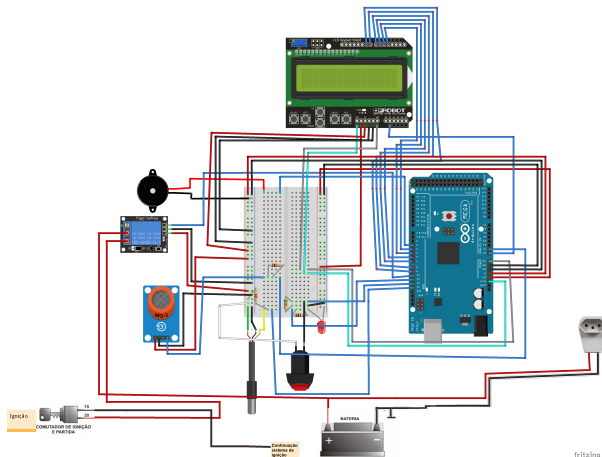


# Drive Safe Detector

- 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.

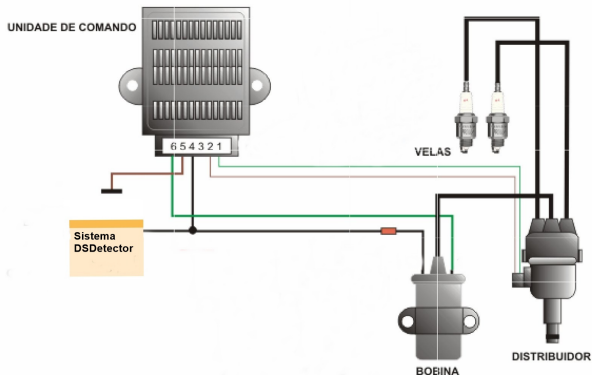


## Esquema Parte 1

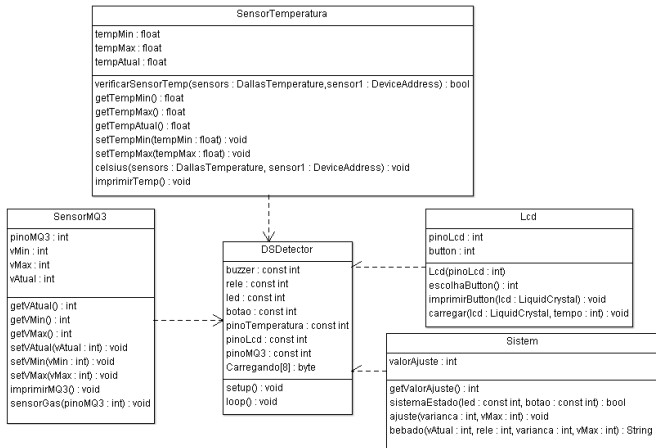


## Esquema Parte 2

### ESQUEMA COMPLETO DO SISTEMA DE IGNIÇÃO ELETRÔNICA TRANSISTORIZADA



# UML



## Funcionamento Parte 1

- 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 acionara o Buzzer Beep.



## Funcionamento Parte 1

```
}else{//Se o sistema estiver desligado  
    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);//reexecutar a cada 2 segundo  
}
```



## Funcionamento Parte 2

- 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.



## Funcionamento Parte 2

```
//-----SETUP-----  
void setup() {  
  Serial.begin(9600); //iniciando porta serial na velocidade 9600 padrão  
  sensors.begin(); //iniciando a biblioteca DallasTemperature  
  lcd.createChar(0, Carregando);  
  lcd.begin(16, 2);  
  sensorTemp = new SensorTemperatura();  
  sistemaMetodos = new DSDetector();  
  sensorMQ = new SensorMQ3();  
  tela = new Lcd(pinoLcd);  
  pinMode(led, OUTPUT); //definindo pino led como saída  
  pinMode(buzzer, OUTPUT);  
  pinMode(rele, OUTPUT);  
  pinMode(botao, INPUT); //definindo pino botao como entrada  
  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); //2min
```



## Funcionamento Parte 3

- 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 alcool na respiração do motorista e a temperatura ambiente.



## Funcionamento Parte 3

```
}else if(sensorTemp->getTempAtual()>19 and sensorTemp->getTempAtual()<=27){
    tela->carregar(lcd,4375);//1min 10s
}else{
    tela->carregar(lcd,100);//2min30s 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();//imprimir resultados da temperatura
    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);//
}
}
```

## Funcionamento Parte 4

- 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.

```
String DSDetector::bebado(int vAtual,int rele,int varianca,int vMax){  
    ajuste(varianca,vMax);  
    if(vAtual <= getValorAjuste()){  
        digitalWrite(rele, LOW);  
        return "Voce nao bebeu...";  
    }else{  
        digitalWrite(rele, HIGH); // desliga o veiculo  
        return "Voce esta bebado";  
    }  
}
```

## Funcionamento Parte 5

- 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.



## Funcionamento Parte 5

```
//-----LOOP-----  
void loop(){  
  if(sistemaMetodos->sistemaEstado(led,botao)){//Verificar se o sistema está ligado  
    lcd.clear();//limpar display lcd  
    sensorTemp->celsius(sensors,sensor1);//função pegar temperatura  
    sensorMQ->sensorGas(pinoMQ3);//função pegar valor ppm  
    lcd.setCursor(0,0);//escolher mostrar na parte superior do display  
    lcd.print("Temperatura:");  
    lcd.print((int)sensorTemp->getTempAtual());//pegar valor da temperatura  
    lcd.print(" C");  
    lcd.setCursor(0,1);//escolher mostrar na parte inferior do display  
    lcd.print("MQ3:");  
    lcd.print(sensorMQ->getVAtual());//pegar valor do mq3  
    delay(3000);//esperar 3 segundos  
    lcd.clear();//limpar display lcd  
    lcd.setCursor(0,0);//escolher mostrar na parte superior do display  
    lcd.print("Resultado:");  
    lcd.setCursor(0,1);//escolher mostrar na parte inferior do display  
    lcd.print("//chamada da função para verificar se está bebado  
sistemaMetodos->bebado(sensorMQ->getVAtual(), rele, (int)sensorTemp->getTempAtual(), sensorMQ->getVMax());  
    if((int)sensorTemp->getTempAtual()<=19){//se a temperatura for inferior a 19°C  
      delay(150000);//esperar 2 minutos e 30 segundos  
    }else if((int)sensorTemp->getTempAtual()>19 and (int)sensorTemp->getTempAtual()<27){  
      delay(120000);//esperar 2 minutos  
    }else{//se a temperatura for superior ou igual a 27°C  
      delay(145000);//esperar 2 minutos e 25 segundos  
    }  
  }  
}
```

## Metodologia dos Testes

- Foram feitas coletas de dados de acordo com os seguintes intervalos de temperatura, temperatura menor ou igual a  $19^{\circ}\text{C}$ , maior que  $19^{\circ}\text{C}$  e menor que  $27^{\circ}\text{C}$  e maior ou igual a  $27^{\circ}\text{C}$ , intervalos os quais foram verificados mudanças na coleta do sensor MQ-3.

## Fórmula

- $BAC(\text{em mg/L}) = BrAc(\text{em mg/L}) * FATOR.$
- BAC = é a concentração de álcool no sangue.
- BrAc = é a concentração de álcool na respiração.
- FATOR = varia de país para país no caso do Brasil é 0,3.





## Tabela 1

- Valores do sensor MQ-3 de acordo com cada faixa de temperatura, logo após o aquecimento do sensor MQ-3.

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

## Tabela 2

- Depois da detecção de álcool o sensor MQ-3 demora um certo tempo para restabelecer seus valores normais, o ppm abaixa rapidamente até um certo ponto depois vai abaixando de forma despadronizada com tempo indefinido para restabelecer o valor inicial.

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






## Tabela 3

- 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

## Referências Bibliograficas

-  Maxin Integrated, *Programmable Resolution 1-Wire Digital Thermometer*. Disponível em:  
<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
-  Paul Brook, *Repositories GitHub, Arduino One-Wire library*. Disponível em: <https://github.com/pbrook/arduino-onewire>.  
16 de outubro de 2012.
-  Miles Burton, *Arduino Library for Maxim Temperature Integrated Circuits*. Disponível em:  
<https://github.com/milesburton/Arduino-Temperature-Control-Library>.  
11 de abril de 2017.

## Referências Bibliográficas



Antony García González, *Sensor MQ-3, un Detector de Alcohol*. Disponível em:  
<http://panamahitek.com/sensor-mq-3/>. 1 de fevereiro de 2014.



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.



## Referências Bibliográficas




-  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.
-  ARDUINO e GENUINO PRODUCTS, *Arduino MEGA 2560 e Genuino MEGA 2560*. Disponível em:  
<https://www.arduino.cc/en/main/arduinoBoardMega2560>.
-  Saulo Anderson Bibiano Jardim, *Como Funciona o Sistema de Ignição*. Disponível em:  
<http://www.ebah.com.br/content/ABAAAA8pgAC/como-funciona-sistema-ignicao>.

## Referências Bibliograficas

-  Git, *Git - Controle de Versões Rápidas*. Disponível em: <https://git-scm.com/doc>.
-  Tigris.org Open Source Software Engineering Tools, *ArgoUML Documentation*. Disponível em: <http://argouml.tigris.org/documentation/>. 29 de janeiro de 2011.
-  GitHub Inc, *The World's Leading Software Sevelopment Platform*. Disponível em: <https://github.com/>.



## Referências Bibliográficas

-  Lei Seca, *Lei Nº 11.705, de 19 de junho de 2008*. Disponível em: <http://www.planalto.gov.br/ccivil03/ato2007-2010/2008/lei/l11705.htm>.  
16 de junho de 2008.
-  Inmetro, *Etilômetro (Bafômetro) Controle pelo Inmetro*. Disponível em: <http://www.inmetro.rs.gov.br/etilometro.html>.
-  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.



## Referências Bibliograficas



Arduino IDE 1.8.2, *Language Reference*. Disponível em:  
<https://www.arduino.cc/en/Reference/HomePage>.



Fritzing Eletronics Made Easy, *Fritzing References*. Disponível em: <http://fritzing.org/learning/fullreference>.

