

# Atividade de desenvolvimento - Arduíno

Lucas Alves Da Costa.

<sup>1</sup>LucasAlves16.LA@gmail.com

<sup>2</sup>Instituto federal de educação, ciência e tecnologia-Goiás  
(IFG)

**Matricula: 20151070130183.**

## 1. Atividade

Utilizar-se de um Arduino para montar a seguinte estrutura:

- A. Detectar a presença de uma pessoa usando o sensor ultrassom. (Você também poderá utilizar um sensor de presença PIR que, apesar de não ter sido explicado em nossa apostila, é muito simples de usar.
- B. Detectar um incêndio através da temperatura muito alta da casa usando um LM35
- C. Detectar se está de noite através de um sensor de luz LDR;
- D. Possuir um botão para ligar e desligar um alarme;
- E. Acender 1 LED de alta intensidade, caso seja noite;
- F. Um alarme, usando o buzzer, deverá tocar, caso a temperatura do LM35 seja muito alta;
- G. Esse alarme também deve tocar, caso acionado e uma pessoa seja detectada;
- H. Medir umidade através do DHT11;
- I. Ter um display LCD que mostrará os valores lidos em cada sensor, se o alarme está ligado ou desligado e o motivo do alarme, caso dispare.

**O Arduino montado ficou da seguinte forma:**

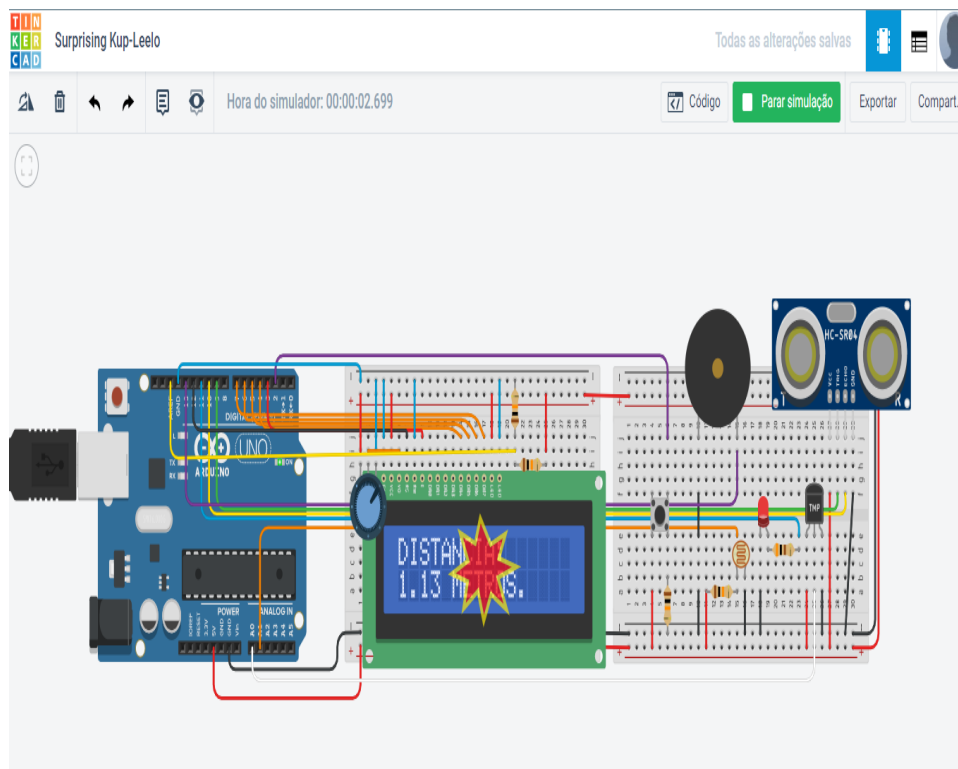


Figura 1- Arduíno montado com todos os componentes exigidos  
**Implementação:** Para cada critério estabelecido será mostrado um trecho de código utilizado na implementação.

### 1.1. Detectar a presença de uma pessoa usando o sensor ultrassom:

```
1 // Espera 0,5s (500ms) entre medicoes.
2 if (millis() - print_timer > 500) {
3     print_timer = millis();
4
5     // Pulso de 5V por pelo menos 10us para iniciar medição.
6     digitalWrite(trig_pin, HIGH);
7     delayMicroseconds(5);
8     digitalWrite(trig_pin, LOW);
9
10    /* Mede quanto tempo o pino de echo ficou no estado alto, ou
11       seja,
12       o tempo de propagação da onda. */
13    uint32_t pulse_time = pulseIn(echo_pin, HIGH);
14
15    /* A distância entre o sensor ultrassom e o objeto será
16       proporcional a velocidade
17       do som no meio e a metade do tempo de propagação. Para o
18       ar na
19       temperatura ambiente  $V_{som} = 0,0343 \text{ cm/us}$ . */
20    double distance = 0.01715 * pulse_time;
21
22    // Imprimimos o valor na porta serial;
23    Serial.print(distance);
24    Serial.println(" cm");
25    double metros = distance/100; // convertendo a distancia em
26    metros
27
28    if(distance < 120.0){ // Verifica se a distancia é menor que
29        1 metro e 20 centímetros OR 120 cm
30        Serial.println("Presenca detectada...");
31        lcd.clear();
32        lcd.print("Presenca ");
33        lcd.setCursor(0,2);
34        lcd.print("detectada!"); // escreve o alerta de presenca
35        na tela de cristal
36        delay(200);
37        lcd.clear();
38        lcd.print("DISTANCIA: ");
39        lcd.setCursor(0,2);
40        lcd.print(metros);
41        lcd.print(" METROS."); // escreve na tela de cristal a
42        distancia detectada pelo sensor
43        delay(200);
44        alarmeDistance(); //Aciona o alarme de distancia
```

38

}

## 1.2. Detectar um incêndio através da temperatura muito alta da casa usando um LM35

```

1  temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
2  float temp = temperatura/1024*1562.5/10; // Converte a
    temperatura em graus celcius
3
4  Serial.print("Temperatura: ");
5  Serial.println(temperatura);
6  lcd.clear(); //Limpa o display
7  //Imprime a temperatura no display
8  lcd.print("Temperatura: ");
9  lcd.setCursor(0,2);
10 lcd.print(temp);
11 lcd.print(" Graus C ");
12 delay(200);
13
14
15 if(temp > 20.0){// Verifica se a temperatura superior a 20
    C
16   Serial.println("incendio detectado");
17   lcd.clear();
18   lcd.print("Alert: INCENDIO");
19   lcd.setCursor(0,2);
20   lcd.print("DETECTADO!"); // escreve o alerta na tela de
    cristal
21   alarme(); // Aciona o alarme
22
23 }
24
25 delay(100);

```

## 1.3. Detectar se está de noite através de um sensor de luz LDR; Acender um LED de alta intensidade, caso seja noite;

```

1  //ler o valor do LDR
2  ldrValor = analogRead(ldrPin); //O valor lido ser entre 0 e
    1023
3  int button2 =(int)*controle; // recebo o estado do bot o
4  //se o valor lido for maior ou igual 800, liga o led
5  if (ldrValor>= 800){
6    digitalWrite(ledPin,HIGH);
7    lcd.clear();
8    lcd.print("Boa noite...");
9    if(button2 != 0){
10     lcd.setCursor(0,2);
11     lcd.print("Alarme: ON"); // escrevo na tela de cristal o
        estado do alarme
12   }
13   delay(200);

```

```

14 }
15 // sen o, apaga o led
16 else{
17     lcd.clear();
18     lcd.print("Bom Dia..");
19     delay(200);
20     if(button2 != 0){
21         lcd.setCursor(0,2);
22         lcd.print("Alarme: ON");//estado do alarme
23     }
24 }

```

#### 1.4. Possuir um botão para ligar e desligar um alarme:

```

1 void interrupcao(){
2     if(*controle != 1) controle = 0;// muda o valor da variavel
        quando detectada uma interrupção
3     if(*controle == 0){
4         buttonState = 1;
5         controle = &buttonState;// mudo o estado do botão para alto
6     }
7     if(*controle == 1){
8         buttonState = 0;
9         controle = &buttonState;// mudo o estado do botão para
            baixo
10    }
11    Serial.println("BUTTON PRESS!!");// console serial: sempre que
        o botão for pressionado
12    lcd.setCursor(15,2);
13    lcd.print("#");// Escreve um '#' no canto inferior direito
        simbolizando que o botão foi acionado.
14    delay(100);
15 }

```

Esta função é iniciada sempre que o botão for pressionado, o alarme já estará ativado ao iniciar o Arduino, quando o botão for pressionado o estado do botão mudará de alto(1) para baixo(0), caso pressionado novamente o estado se inverte de baixo para alto.

#### 1.5. Um alarme, usando o buzzer, deverá tocar, caso a temperatura do LM35 seja muito alta:

```

1 void alarme(){
2     int button =(int)*controle;
3     Serial.print("ALARME state-> ");
4     Serial.println(*controle); // imprime o valor da variavel no
        serial console
5
6     if(button == 0){// Caso o botão mude de estado (1/0) escrevo
        na tela de cristal
7         lcd.clear();
8         lcd.print("ALERT: Alarme");

```

```

9     lcd.setCursor(0,2);
10    lcd.print("Desativado!");
11    delay(200);
12    }
13    while(button != 0){// enquanto o estado do bot o for alto o
        alarme toca
14    for(int x=0;x<180;x++){
15        //converte graus para radiando e depois obt m o valor do
            seno
16        seno=(sin(x*3.1416/180));
17        //gera uma frequencia a partir do valor do seno
            frequencia = 2000+(int(seno*1000));
18        tone(13,frequencia);
19        delay(2);
20    }
21
22    Serial.print("O ALARME state-> ");
23    Serial.println(*controle);
24    // passando o estado atual do bot o para a variav l de
        controle do la o
25    button = (int)*controle;
26    }
27
28    noTone(13);// caso o estado do bot o mude finalizo o alarme
        no pino 13
29    Serial.println("Alarme finalizado!");
30    }

```

A função *alarme()* é responsável por emitir uma frequência sonora no buzzer conectado ao Arduino, sempre que for executada essa função verificará o estado do botão caso seja alto é considerado que o alarme está ativado, caso seja baixo considera-se o alarme desativado, enquanto o botão estiver em estado alto o alarme irá tocar, caso seja pressionado seu estado se reverte de alto para baixo e o alarme é encerrado. O alarme só irá disparar caso esteja acionado, caso contrario será exibida uma mensagem informando que o mesmo está desativado.

#### 1.6. Esse alarme também deve tocar, caso acionado e uma pessoa seja detectada:

```

1 //Alarme do sensor ultrassonico utilizado para detectar
    presen a
2 void alarmeDistance(){
3     int frequencia = 1500;
4     // Emite uma leve frequencia de X intera es , cada uma sendo
        mais que a ultima
5     for(int x=0;x<20;x++){
6         tone(13,frequencia);
7         frequencia -= 15;
8         delay(2);
9     }
10    tone(13,0);// for ando a parada do alarme
11    noTone(13);// interrompendo o alarme no pino 13.
12 }

```

Esta função emite um pequeno som no buzzer conectado ao Arduino sempre que algo for detectado pelo sensor ultrassônico ha uma distancia mínima. Fiz essa função separada pois não seria viável disparar um alarme sempre que uma pessoa for detectada, ao invés disso é emitido uma leve frequência de alerta.

### 1.7. Medir umidade através do DHT11:

O ambiente de desenvolvimento utilizado para construir esta atividade não suporta a biblioteca utilizada pelo sensor DHT11, por isso não o utilizei.

### 1.8. Ter um display LCD que mostrará os valores lidos em cada sensor, se o alarme está ligado ou desligado e o motivo do alarme, caso dispare:

Para finalizar a tarefa veja a baixo o código completo da atividade desenvolvida, contendo a implementação da tela de cristal liquido e os respectivos valores nela impressos.

## 2. Todo o código fonte está comentado para melhor entendimento.

```
1 #include <LiquidCrystal.h> //Inclui biblioteca no programa
2 /*
3 Cria objeto lcd da classe LiquidCrystal
4 RS 2
5 Enable 3
6 DB4 4
7 DB5 5
8 DB6 6
9 DB7 7
10 */
11 LiquidCrystal lcd(12,3,4,5,6,7);
12
13 // Leitura HC-SR04
14 const uint8_t trig_pin = 9;
15 const uint8_t echo_pin = 10;
16 //Sensor de temperatura usando o LM35
17 const int LM35 = A0; // Define o pino que lera a sa da do LM35
18 float temperatura; // Vari vel que armazenara a temperatura
19   medida
19 uint32_t print_timer;
20 //Sensor de luz
21 int ledPin = 11; //Led no pino 11
22 int ldrPin = 1; //LDR no pino anal gico A1
23 int ldrValor = 0; //Valor lido do LDR
24 //AlaRme
25 float seno;
26 int frequencia;
27 int buttonState;
28 const int buttonPin = 2; // Numero do pino do botao de pressao
29 const int alrmPin = 13; // Numero do pino do alarme
30 // Vari veis
31 // Vari vel para leitura do estado do botao
```

```

32 int* controle; //Ponteiro usado para mudar o estado do botao em
    tempo de execucao
33 // Função que dispara o alarme de incendio quando chamada
34 void alarme(){
35     int button =(int)*controle;
36     Serial.print("ALARME state-> ");
37     Serial.println(*controle); // imprime o valor da variavel no
        serial console
38
39     if(button == 0){// Caso o botao mude de estado (1/0) escrevo
        na tela de cristal
40         lcd.clear();
41         lcd.print("ALERT: Alarme");
42         lcd.setCursor(0,2);
43         lcd.print("Desativado!");
44         delay(200);
45     }
46     while(button != 0){// enquanto o estado do botao for alto o
        alarme toca
47         for(int x=0;x<180;x++){
48             //converte graus para radiando e depois obtem o valor do
                seno
49             seno=(sin(x*3.1416/180));
50             //gera uma frequencia a partir do valor do seno
51             frequencia = 2000+(int(seno*1000));
52             tone(13,frequencia);
53             delay(2);
54         }
55
56         Serial.print("O ALARME state-> ");
57         Serial.println(*controle);
58         // passando o estado atual do botao para a variavel de
            controle do laço
59         button = (int)*controle;
60     }
61     noTone(13);// caso o estado do botao mude finalizo o alarme
        no pino 13
62     Serial.println("Alarme finalizado!");
63 }
64 //Alarme do sensor ultrassonico utilizado para detectar presenca
65 void alarmeDistance(){
66     int frequencia = 1500;
67     // Emite uma leve frequencia de X intera es , cada uma sendo
        mais que a ultima
68     for(int x=0;x<20;x++){
69         tone(13,frequencia);
70         frequencia -= 15;
71         delay(2);
72     }

```

```

73 tone(13,0); // forçando a parada do alarme
74 noTone(13); // interrompendo o alarme no pino 13.
75 }
76 //Funcao responsavel por mudar o valor da variavel (controle)
   utilizada em outras funcoes
77 // Isso feito a partir de uma interrupcao pr -definida.
78 void interrupcao(){
79     if(*controle != 1) controle = 0; // muda o valor da variavel
   quando detectada uma interrupcao
80     if(*controle == 0){
81         buttonState = 1;
82         controle = &buttonState; // mudo o estado do botao para alto
83     }
84     if(*controle == 1){
85         buttonState = 0;
86         controle = &buttonState; // mudo o estado do botao para baixo
87     }
88     Serial.println("BUTTON PRESS!!"); // console serial: sempre que
   o botao for pressionado
89     lcd.setCursor(15,2);
90     lcd.print("#"); // Escreve um '#' no canto inferior direito
   simbolizando que o bot o foi acionado.
91     delay(100);
92 }
93 //Executa primeiramente e uma s vez quando o Arduino liga
94 void setup() {
95     attachInterrupt(0,interrupcao,RISING); // Configura a
   interrup o
96     Serial.begin(9600); // Habilita Comunica o Serial a uma
   taxa de 9600 bauds.
97     pinMode(ledPin,OUTPUT); //define a porta 11 como sa da
98     analogReference(EXTERNAL); // Muda a referencia para a
   tens o no pino AREF
99     lcd.begin(16,2); //Inicializa display de 2 linhas x 16 colunas
100
101     // Configura o do estado inicial dos pinos Trig e Echo.
102     pinMode(trig_pin, OUTPUT);
103     pinMode(echo_pin, INPUT);
104     digitalWrite(trig_pin, LOW);
105     // saida do alarme
106     pinMode(13,OUTPUT);
107     pinMode(alrmPin, OUTPUT); //Inicializa o pino do alarme como
   saida (OUTPUT)
108     pinMode(buttonPin, INPUT); // Inicializa o pin do botao como
   entrada (INPUT)
109
110 }
111
112 void loop() {

```



```

113 //ler o valor do LDR
114 ldrValor = analogRead(ldrPin); //O valor lido sera entre 0 e
    1023
115     int button2 =(int)*controle;// recebo o estado do botao
116 //se o valor lido for maior ou igual      800, liga o led
117 if (ldrValor>= 800){
118     digitalWrite(ledPin,HIGH);
119     lcd.clear();
120     lcd.print("Boa noite...");
121     if(button2 != 0){
122         lcd.setCursor(0,2);
123         lcd.print("Alarme: ON");// escrevo na tela de cristal o
            estado do alarme
124     }
125     delay(200);
126 }
127 // sen o , apaga o led
128 else{
129     lcd.clear();
130     lcd.print("Bom Dia..");
131     delay(200);
132     if(button2 != 0){
133         lcd.setCursor(0,2);
134         lcd.print("Alarme: ON");//estado do alarme
135     }
136 }
137 // Espera 0,5s (500ms) entre medicoes.
138 if (millis() - print_timer > 500) {
139     print_timer = millis();
140
141     // Pulso de 5V por pelo menos 10us para iniciar medicao.
142     digitalWrite(trig_pin, HIGH);
143     delayMicroseconds(5);
144     digitalWrite(trig_pin, LOW);
145
146     /* Mede quanto tempo o pino de echo ficou no estado alto, ou
        seja,
147     o tempo de propaga o da onda. */
148     uint32_t pulse_time = pulseIn(echo_pin, HIGH);
149
150     /* A distancia entre o sensor ultrassom e o objeto ser
        proporcional a velocidade
151     do som no meio e a metade do tempo de propaga o. Para o
        ar na
152     temperatura ambiente Vsom = 0,0343 cm/us. */
153     double distance = 0.01715 * pulse_time;
154
155     // Imprimimos o valor na porta serial;
156     Serial.print(distance);

```

```

157     Serial.println(" cm");
158     double metros = distance/100;// convertendo a distancia em
        metros
159
160     if(distance < 120.0){// Verifica se a distancia      menor que
        1 metro e 20 centimetros OR 120 cm
161         Serial.println("Presenca detectada...");
162         lcd.clear();
163         lcd.print("Presenca ");
164         lcd.setCursor(0,2);
165         lcd.print("detectada!");// escreve o alerta de presenca na
        tela de cristal
166         delay(200);
167         lcd.clear();
168         lcd.print("DISTANCIA: ");
169         lcd.setCursor(0,2);
170         lcd.print(metros);
171         lcd.print(" METROS.");// escreve na tela de cristal a
        distancia detectada pelo sensor
172         delay(200);
173         alarmeDistance();//Aciona o alarme de distancia
174     }
175
176 }
177 /*-----*/
178 temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
179 float temp = temperatura/1024*1562.5/10;// Converte a
        temperatura em graus celcius
180
181     Serial.print("Temperatura: ");
182     Serial.println(temperatura);
183     lcd.clear(); //Limpa o display
184     //Imprime a temperatura no display
185     lcd.print("Temperatura: ");
186     lcd.setCursor(0,2);
187     lcd.print(temp);
188     lcd.print(" Graus C ");
189     delay(200);
190
191
192     if(temp > 20.0){// Verifica se a temperatura      superior a 20
        C
193         Serial.println("incendio detectado");
194         lcd.clear();
195         lcd.print("Alert: INCENDIO");
196         lcd.setCursor(0,2);
197         lcd.print("DETECTADO!");// escreve o alerta na tela de
        cristal
198         alarme();// Aciona o alarme

```

```

199     }
200
201
202     delay(100);
203 }

```

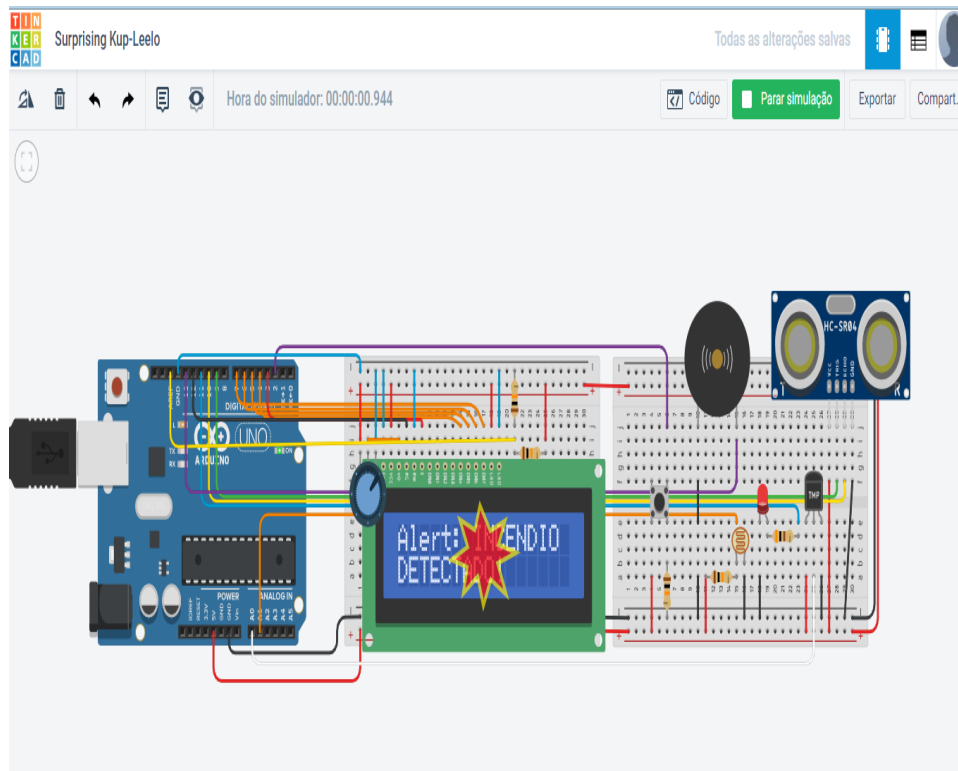


Figura 2- Projeto em funcionamento