

Projeto-Internet das Coisas

♥ **Erika Maschio Dos Santos**
maschioerika1@gmail.com

♥ **Giovana Bernardes da Silva**
bgiovana622@gmail.com



CardioTech

Cardiotech–Leitor de Frequência Cardíaca

A **CardioTech** visa não apenas criar um produto inovador, mas também impactar positivamente a vida das pessoas ao oferecer uma solução acessível e confiável para o monitoramento da saúde cardiovascular.

Nosso leitor não se limita a simplesmente registrar a frequência cardíaca; ele representa um avanço na integração entre tecnologia e praticidade do uso diário.



Utilizando a robustez do Arduino Uno e uma tecnologia de sensoriamento, nosso dispositivo proporcionará aos usuários a capacidade de monitorar sua saúde cardiovascular de forma contínua e precisa. Isso não só permite que eles estejam conscientes de seu estado de saúde em tempo real, mas também facilita a detecção precoce de possíveis irregularidades no ritmo cardíaco, proporcionando uma janela de oportunidade para intervenções preventivas.

Ao oferecer um dispositivo portátil, versátil e de fácil uso, a **CardioTech** está comprometida não apenas com a qualidade do produto, mas também com o bem-estar e a segurança dos seus usuários. Estamos determinados a estabelecer novos padrões de excelência no mercado de dispositivos de saúde pessoal, tornando o monitoramento da frequência cardíaca uma parte integrante e acessível da rotina diária de cuidados com a saúde.

Benefícios do Produto:

01

Precisão e Confiabilidade:

Nosso leitor oferecerá resultados precisos para que os usuários possam confiar nas informações obtidas.

02

Portabilidade e Versatilidade:

Design compacto e leve que facilita o transporte e permite o monitoramento conveniente em qualquer lugar.

03

Promoção da Saúde Pessoal:

Ao facilitar o monitoramento regular da frequência cardíaca, o produto incentivará hábitos saudáveis e permitirá que os usuários tomem medidas preventivas para manter sua saúde cardiovascular.

Materiais utilizados:

Arduino Uno, tela OLED 128×32, sensor MAX30102, campainha, tinkercad

Concorrentes:

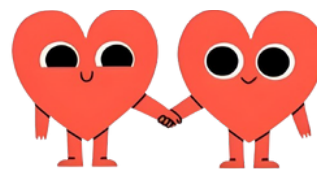
Empresas de tecnologia: Apple, Samsung e Garmin

Startups especializadas em saúde e bem-estar: Fitbit, Withings, Whoop, entre outras.

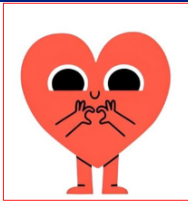
Empresas de dispositivos médicos: Medtronic, Philips e Abbott.

Respostas da pesquisa

Link do formulário: <https://forms.gle/E73UzTV8N7Rx7kgY7>



Nome	Qual é o seu interesse em leitores de frequência cardíaca?	Quais funcionalidades você considera mais importantes ?	Você estaria disposto a investir em um leitor de frequência cardíaca ?
Daniela Maschio	Muito interessado	Precisão nas medições	Sim
Isabela Maschio	Interessado	Análise de dados em tempo real	Sim
Vilma Madrid	Muito interessado	Precisão nas medições	Sim
Celso Maschio	Interessado	Precisão nas medições	Sim
Israel Vieira	Interessado	Precisão nas medições	Talvez
Alessandro Lima	Muito interessado	Precisão nas medições	Sim
Maiara Monteiro	Interessado	Precisão nas medições	Talvez
Andrey Mottola	Interessado	Precisão nas medições	Sim
Grasi Vieira	Interessado	Análise de dados em tempo real	Não
Luan Ramos	Pouco interessado	Facilidade de uso	Talvez
Celo Duarte	Interessado	Conectividade com aplicativos móveis	Talvez
Vivian Da Silva	Pouco interessado	Análise de dados em tempo real	Não
Lorenzo Camaciel	Interessado	Análise de dados em tempo real	Não
Mateus	Pouco interessado	Análise de dados em tempo real	Sim
Vitor Gabriel	Pouco interessado	Facilidade de uso	Sim
Francisco Lopes	Interessado	Precisão nas medições	Sim
Eduardo Pinto	Pouco interessado	Análise de dados em tempo real	Talvez
Rebeca Bueno	Muito interessado	Análise de dados em tempo real	Sim
Biel Reupert	Interessado	Precisão nas medições	Talvez
Guilherme Vaz	Interessado	Precisão nas medições	Talvez
Adam Kisa	Interessado	Facilidade de uso	Talvez
Nicolas heberle	Muito interessado	Portabilidade	Sim
Camile Schicht	Pouco interessado	Conectividade com aplicativos móveis	Não
Henrique mds	Não interessado	Análise de dados em tempo real	Não
Nathalia Flores	Pouco interessado	Precisão nas medições	Sim
Weslleys Lemos	Não interessado	Precisão nas medições	Não



· **Erika Maschio Dos Santos**

maschioerika1@gmail.com

· **Giovana Bernardes da Silva**

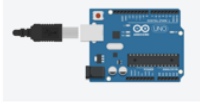
bgiovana622@gmail.com

O nosso projeto consiste na criação de um protótipo de um sensor de batimentos cardíacos, onde serão utilizadas ferramentas como o Arduino Uno, uma placa de ensaio, um sensor ultrassônico, um sensor PIR, um micro servo, um led, dois Lcd's 16x2, um gerador de funções e um osciloscópio.

O aparelho tem como objetivo proporcionar aos usuários a possibilidade de monitorar sua saúde cardiovascular de forma simples e precisa, tornando o monitoramento da frequência cardíaca acessível de forma descomplicada e objetiva.



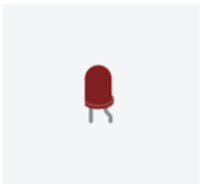
Arduino Uno



Placa de ensaio

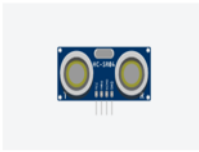


Led





Sensor ultrassônico



4 / 7

Sensor PIR

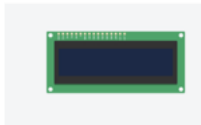


Micro servo





Lcd 16x2



Gerador de funções





Osciloscópio



7 / 7

Circuito pronto



Projeto Internet das coisas

Giovana Bernardes da Silva - bgiovana622@gmail.com

Erika Maschio Dos Santos - maschioerika1@gmail.com



Código do projeto:

O sistema consiste em analisar os dados da frequências cardíacas de um pessoa em um certo intervalo de tempo.

E, após isso, retornar o status dos batimentos medidos que serão divididos em quatro status;

Status: Normal (de 1hz a 1.2 hz)

Quando a media dos batimentos cardíacos esta normal.

Status: Bom (entre 1.2hz a 1.5hz)

Quando a media dos batimentos cardíacas esta boa.

Status: Alerta (entre 1.5 hz e 2.2 hz)

Quando a media dos batimentos cardíacos esta alta.

Status: Perigo (2.2 hz ou mais)

Quando a media dos batimentos cardíacos esta muito alta

Componentes:

1 Servo motor;

1 Sensor ultrassônico;

1 Sensor PIR;

1 Osciloscópio;

1 Gerador de funções;

2 Lcds 16x2;

1 Led Vermelho;

*/

```
#include <LiquidCrystal.h>
```

```
#include <Servo.h>
```

```
#define trig 5
```

```
#define echo 4
```

```
LiquidCrystal lcd1(6,7,10,11,12,13);
```

```
LiquidCrystal lcd2(8,9,10,11,12,13);
```

```
Servo motor;
```

```
// Recebera a leitura do gerador de funções;
```

```
int  leitura;
```

```
// Variável usada para controle do servo motor
```

```
int pos = 0;
```

```
/*
```

Variáveis time e dist serão responsáveis por armazenar, respectivamente, o tempo que a onda sonora emitida pelo sensor demorou para chegar à pessoa até o sensor, e a distância que este está do sensor.

Onde a dist é a metade da distância percorrida pela onda, que será calculada usando a fórmula $\Rightarrow \Delta X = v \cdot \Delta T$, onde v é a velocidade do som no ar, à temperatura ambiente, $v \sim 340,29 \text{ m/s}$;

```
*/
```

```
int time, dist;
```

```
// Acumula a diferença entre os tempos (em milissegundos) do ligar e apagar do led, para  
// checagem da frequência média dos batimentos cardíacos
```

```
float mediaDif, ciclosCounter;
```

```
// Variáveis responsáveis por armazenar, respectivamente, o tempo (em milissegundos)  
// do acender do led, do apagar, e para o cálculo da diferença entre esses tempos
```

```
float timeStart, timeEnd, timeDif;
```

```
// Variável responsável por indicar se pessoa foi detectada pelo sensor pir para ativar o  
// sistema
```

```
bool presenca = false;
```

```
// Variáveis responsáveis por indicar o início e fim de leitura do gerador de funções
```

```
bool startLeitura, leituraCompleta;
```

```
// Variáveis responsáveis por setar a aparição da dica correta no lcd uma única vez após a leitura ser completada
```

```
bool dica_1, dica_2, dica_3, dica_4, ciclo;
```

```
// Variáveis que representa, respectivamente, o limite de contagem dos ciclos e o contador de ciclos
```

```
const byte counterLimit = 50;
```

```
byte counter;
```

```
void setup()
```

```
{
```

```
    // Inicializa os displays
```

```
    Serial.begin(9600);
```

```
    lcd1.begin(16,2);
```

```
    lcd2.begin(16,2);
```

```
    // Inicializa as variáveis globais
```

```
    varsStart();
```

```
    // Inicializa os pinos para os leds e os componente do sistema
```

```
    pinMode(2, OUTPUT); // Servo motor
```

```
    // Pinos para os leds
```

```
    pinMode(6, OUTPUT);
```

```
    pinMode(7, OUTPUT);
```

```
    pinMode(8, OUTPUT);
```

```
pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

pinMode(A3, OUTPUT); // Pino para o led vermelho

pinMode(A2, INPUT); // Pino para gerador de funções

pinMode(trig, OUTPUT); // Sensor ultrassônico

pinMode(echo, INPUT); // Sensor ultrassônico


// Inicializa o servo motor

motor.attach(2);

motor.write(0);


// Pino de interrupção para o sensor pir

pinMode(3, INPUT);

attachInterrupt(digitalPinToInterrupt(3), acionar, RISING);

}


void loop()

{

    if (presenca) // O sistema será ativado assim que for detectado alguma movimento pelo
    sensor pir

    {

        lcd1.display();

    }

}
```

```
lcd2.display();
```

```
proxySensor();
```

```
servoMotor();
```

```
statusCheck();
```

```
/*
```

Se a distancia do dedo da pessoa ao sensor for menor
que 0.18 m e a leitura não estiver completa, a função
freqCardica será chamada.

```
*/
```

```
if (dist <= 180 && !leituraCompleta)
```

```
    freqCardiaca();
```

```
}
```

```
else
```

```
{
```

```
    analogWrite(A3, 0);
```

```
    lcd1.noDisplay();
```

```
    lcd2.noDisplay();
```

```
}
```

```
}
```

```
/*
```

Função freqCardiaca(): será executada sempre que a leitura
não estiver completa, e o contador não chegar a seu limite

máximo devido anteriormente (50 ciclos).

A lógica implementada na função segue a seguinte ideia:

se o dado lido pelo gerador de funções estiver entre $]-400, 400[$

o led vermelho será aceso, caso contraio, será desligado.

Apos acionar o led, será armazenado o instante em que ele foi aceso, e, após desligar, será armazenado o tempo que foi desligado.

Se a diferença entre os tempos for muito pequena, significa que o led acendeu e apagou rápida, seguindo a lógica dada pela formula $f = 1/T$. Logo, terá uma frequências alta.

E, se caso a diferença entre os tempos for alta, a frequências será baixa.

Para uma melhor estimativa dos dados da frequências cardíaca, será feito um calculo onde o tempo total dos batimentos no ciclo definido no projeto (50 ciclos no loop do arduino) será dado pelo período medio. Logo a frequências calculada será a frequências media dos batimentos cardíacos nos 50 ciclos definidos no código.

*/

```
void freqCardiaca()
```

```
{
```

```
  if (counter != counterLimit)
```

```
  {
```



```
leitura = analogRead(A2);
```

```
if (leitura > 400 || leitura < -400)
```

```
{
```

```
    delay(20);
```

```
    analogWrite(A3, 255);
```

```
    delay(20);
```

```
    timeStart = millis();
```

```
}
```

```
else
```

```
{
```

```
    delay(20);
```

```
    analogWrite(A3, 0);
```

```
    delay(20);
```

```
    timeEnd = millis();
```

```
    ciclo = true;
```

```
}
```

```
delay(130);
```

```
if (ciclo)
```

```
{
```

```
    timeDif = timeEnd - timeStart;
```

```
if (timeDif < 0)
    timeDif = -timeDif;
```

```
if (timeDif < 1000)
{
    mediaDif += timeDif;
    ciclosCounter++;
}
ciclo = false;
}
```

```
counter++;
```

```
}
```

```
else
```

```
{
    leituraCompleta = true;
    analogWrite(A3, 0);
    mediaDif /= ciclosCounter;
}
```

```
}
```

```
void proxySensor()
```

```
{
```

```
// Emite um pulso sonoro de 40 kHz
```

```
digitalWrite(trig, 1);
```

```
delayMicroseconds(10);
```

```
digitalWrite(trig, 0);
```

```
// Retorna a duração do pulso em microsegundos
```

```
time = pulseIn(echo, 1);
```

```
/*
```

$v \sim 340,29 \text{ m/s} = 34029 \text{ cm/s} = 34029 \cdot 10^{-6} \text{ cm/us}$

$v \sim 0.034029 \text{ cm/us}$

como a distancia até o dedo da pessoa é a metade

da distância percorrido pelo som, então $d = dx/2 = v \cdot dt/2$

$d = dt \cdot 0.034029/2 \Rightarrow d = dt \cdot 0.0170145;$

```
*/
```

```
dist = time*0.0170145;
```

```
delay(10);
```

```
}
```

```
void servoMotor()
```

```
{
```

```
if (dist <= 180 && pos <= 0) servoOpen();
```

```
if (dist > 180 && pos >= 90) servoClose();  
}
```

```
void servoOpen()  
{  
  for (pos = 0; pos <= 90; pos += 3)  
    motor.write(pos), delay(20);  
}
```

```
void servoClose()  
{  
  for (pos = 90; pos >= 0; pos -= 3)  
    motor.write(pos), delay(20);  
}
```

```
void varsStart()  
{  
  analogWrite(A3, 0);  
  leituraCompleta = false;  
  startLeitura = false;  
  dica_1 = dica_2 = dica_3 = dica_4 = ciclo = false;  
  mediaDif = ciclosCounter = timeStart = timeEnd = timeDif = 0;  
  counter = 0;  
  lcd1.clear();  
}
```

```
    lcd2.clear();  
}
```

```
void statusCheck()  
{  
    if (dist > 180 && leituraCompleta)  
        varsStart();  
  
    else if (dist > 180)  
        exibirMsg(7);  
  
    else if (dist <= 180 && counter < counterLimit && !startLeitura)  
        exibirMsg(6);  
  
    else if (dist <= 180 && counter >= counterLimit && leituraCompleta)  
        exibirMsg(8);  
}
```

```
int exibirDica()  
{  
    int dica;  
  
    if (mediaDif < 330 && mediaDif > 280) //Normal (1 hz à 1.2 hz)  
        dica = 1;
```

```
else if (mediaDif <= 280 && mediaDif > 255) //Bom (1.2 hz à 1.5 hz)
```

```
    dica = 2;
```

```
else if (mediaDif <= 255 && mediaDif > 206) //Alerta (1.5 hz à 2.2 hz)
```

```
    dica = 3;
```

```
else if (mediaDif <= 206) //Perigo (2.2 hz ou mais)
```

```
    dica = 4;
```

```
exibirMsg(dica);
```

```
return dica;
```

```
}
```

```
void printLcd1Status()
```

```
{
```

```
    lcd1.setCursor(0,1);
```

```
    switch (exibirDica())
```

```
    {
```

```
        case 1:
```

```
            lcd1.print("Status: Normal");
```

```
            break;
```

```
        case 2:
```

```
            lcd1.print("Status: Bom");
```

```
break;

case 3:

    lcd1.print("Status: Alerta!");

break;

default:

    lcd1.print("Status: Perigo!");

break;

}

}

// Função responsável por exibir as mensagens nos leds;

void exibirMsg(int index)

{

    switch (index)

    {

        case 1:

            dica1();

            break;

        case 2:

            dica2();

            break;

        case 3:

            dica3();

            break;
```

```
case 4:

    dica4();

break;

case 5:

    msgDados();

break;

case 6:

    msgAnalise();

break;

case 7:

    msgAproximar();

break;

default:

    msgReaproximar();

break;

}

}
```

```
void msgDados()

{

    if ((dica_1 == dica_2) && (dica_3 == dica_4))

    {

        lcd2.clear();

        lcd1.clear();

        lcd2.setCursor(0,0);
```



```
    lcd2.print("Dados analisados");  
  
    delay(1500);  
  
    lcd2.clear();  
  
}
```

```
    lcd1.clear();  
  
    lcd1.setCursor(0,0);  
  
    lcd1.print("Analisado!");  
  
    printLcd1Status();  
  
}
```

```
void msgAnalise()  
{  
  
    startLeitura = true;  
  
    analogWrite(A3, 0);  
  
    lcd2.clear();  
  
    lcd1.clear();  
  
    lcd1.setCursor(0,0);  
  
    lcd1.print("Analisando...");  
  
  
  
    lcd2.setCursor(0,0);  
  
    lcd2.print("Aguarde...");  
  
    delay(1000);  
  
    lcd2.clear();  
  
    lcd2.setCursor(0,1);
```

```

    lcd2.print("Coletando dados!");

    delay(1000);

    lcd2.clear();
}

void msgAproximar()
{
    lcd2.clear();

    lcd2.setCursor(0,0);

    lcd2.print("Ola! ");

    delay(1000);

    lcd2.clear();

    lcd2.setCursor(0,0);

    lcd2.print("Aproxime seu dedo do sensor!");

    delay(1000);

    for (int i = 0; i < 12; i++)

        lcd2.scrollDisplayLeft(), delay(106);

    delay(200);

    lcd2.clear();
}

void msgReaproximar()
{
    startLeitura = false;

```

```
msgDados();

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("Para uma nova leitura, reaproxime");

delay(500);

for (int i = 0; i < 17; i++)

    lcd2.scrollDisplayLeft(), delay(106);

delay(500);

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("o dedo do sensor");

delay(1000);

lcd2.clear();

}
```

```
// Dicas;
```

```
void dica1()

{

    if (dica_1 == false)

    {

        dica_1 = true;

        lcd2.clear();

        lcd2.setCursor(0,0);
```

```
lcd2.print("Status: Normal!");

delay(1000);

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("Para manter esse status, mantenha uma");

delay(500);

for (int i = 0; i < 21; i++)

    lcd2.scrollDisplayLeft(), delay(106);

delay(800);

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("Alimentacao Saudavel!");

delay(300);

for (int i = 0; i < 6; i++)

    lcd2.scrollDisplayLeft(), delay(106);

delay(1500);

}

}
```

```
void dica2()

{

    if (dica_2 == false)

    {

        dica_2 = true;

        lcd2.clear();
```

```

    lcd2.setCursor(0,0);

    lcd2.print("Status: Bom!");

    delay(1000);

    lcd2.clear();

    lcd2.setCursor(0,1);

    lcd2.print("Para melhorar o status,");

    delay(700);

    for (int i = 0; i < 7; i++)

        lcd2.scrollDisplayLeft(), delay(106);

    delay(800);

    lcd2.clear();

    lcd2.setCursor(0,1);

    lcd2.print("Adote uma Alimentacao Saudavel!");

    delay(200);

    for (int i = 0; i < 14; i++)

        lcd2.scrollDisplayLeft(), delay(106);

    delay(1500);

}

}

void dica3()

{

    if (dica_3 == false)

    {

        dica_3 = true;

```

```

    lcd2.clear();

    lcd2.setCursor(0,0);

    lcd2.print("Status: Alerta!");

    delay(1000);

    lcd2.clear();

    lcd2.setCursor(0,1);

    lcd2.print("Pode ser nervosismo ou algo grave.");

    delay(500);

    for (int i = 0; i < 19; i++)

        lcd2.scrollDisplayLeft(), delay(106);

    delay(800);

    lcd2.clear();

    lcd2.setCursor(0,1);

    lcd2.print("Recomendamos que procure um medico!");

    delay(200);

    for (int i = 0; i < 20; i++)

        lcd2.scrollDisplayLeft(), delay(106);

    delay(1000);

}

}

void dica4()

{

    if (dica_4 == false)

    {

```

```

dica_4 = true;

lcd2.clear();

lcd2.setCursor(0,0);

lcd2.print("Status: Ruim!");

delay(1000);

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("Nada bom, recomendamos");

delay(500);

for (int i = 0; i < 6; i++)

    lcd2.scrollDisplayLeft(), delay(106);

delay(500);

lcd2.clear();

lcd2.setCursor(0,1);

lcd2.print("que procure um medico!");

delay(300);

for (int i = 0; i < 6; i++)

    lcd2.scrollDisplayLeft(), delay(106);

delay(1500);

}

}

// Se a função for chamada, resetara todas as variáveis e desligara o sistema

void acionar() {

    presenca = !presenca;

```

```
varsStart();  
  
lcd1.noDisplay();  
  
lcd2.noDisplay();  
  
if (pos >= 90) servoClose();  
}
```