

Monitoramento de Frequência Cardíaca

Daniel Rath

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília, FGA UnB
Gama - DF
daniel7rath@gmail.com

Rodrigo Sousa Santos

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília, FGA UnB
Gama - DF
rodrigo.sousa2711@gmail.com

1. RESUMO

Este projeto tem como objetivo monitorar a frequência cardíaca voltada às pessoas que praticam atividades físicas visando um treinamento mais eficaz para perda de gordura.

2. INTRODUÇÃO

A obesidade no Brasil cresce cada vez mais ao decorrer do tempo. Nos últimos dez anos, houve um aumento de 60% no número de pessoas acima do peso, sendo 1 (uma) cada 5 (cinco) pessoas [1].

Muitas pessoas acreditam que suar é sinônimo de queimar calorias, porém, a realidade é que um exercício menos intenso e duradouro pode ser tão eficiente e quanto uma série de exercícios curtos e bastante intensos [2]. Para se obter esta eficiência, a variável mais importante é a frequência cardíaca.

A frequência cardíaca ideal para queimar gordura e emagrecer durante o treino é de 60 a 75% da frequência cardíaca (FC) máxima, que varia de acordo com a idade e sexo. O treino nesta intensidade melhora o condicionamento físico, utilizando mais gordura como fonte de energia, contribuindo para perda de peso [3].

Para se conhecer a faixa de frequência cardíaca ideal, deve-se aplicar a seguinte fórmula:

Homens: $220 - (\text{idade}) * 0,6$ e $220 - (\text{idade}) * 0,75$
Mulheres: $226 - (\text{idade}) * 0,6$ e $226 - (\text{idade}) * 0,75$ [3]

A tabela abaixo apresenta a frequência cardíaca ideal para o emagrecimento e a perda de gordura, de indivíduos de idades: entre 20 e 40 anos [3].

| Idade | FC ideal para homens | FC ideal para mulheres |
|-------|----------------------|------------------------|
| 20 | 120 - 150 | 123 - 154 |
| 25 | 117 - 146 | 120 - 150 |
| 30 | 114 - 142 | 117 - 147 |
| 35 | 111 - 138 | 114 - 143 |
| 40 | 108 - 135 | 111 - 139 |

Tabela 1 – Frequência Cardíaca ideal [3].

Dessa maneira, a utilização de aparelhos para a medição de frequência cardíaca tem sido cada vez mais difundida, tendo em vista a magnitude do resultado proveniente dos tais.

2.1 OBJETIVOS

Montar uma pulseira capaz de monitorar a frequência cardíaca durante a prática de exercício físico, mostrando e atualizando esta faixa de frequência em que se encontra de acordo com os batimentos cardíacos, podendo ser visualizada pelo celular.

2.2 REQUISITOS

Consistem nos requisitos desse projeto, a utilização de um sensor de frequência cardíaca, um módulo de display (*smartphone* - conexão via *Bluetooth*), microcontrolador, além de um módulo de energia.

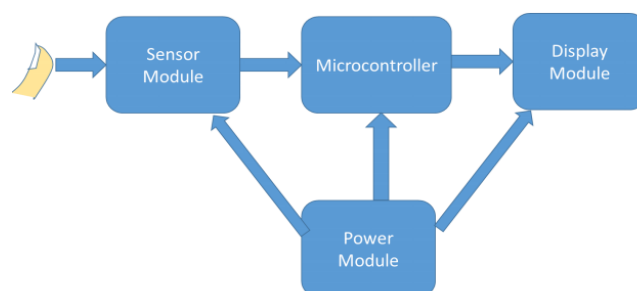


Figura 1 – Requisitos do projeto [4].

Tradução das palavras em língua inglesa - *Sensor module*: módulo do sensor; *microcontroller*: microcontrolador; *display module*: módulo de exibição (tela); *power module*: módulo de energia.

2.3 BENEFÍCIOS

Atualmente, tem sido até motivo de propaganda o fato de muitos *smartphones* (Android e iPhone), e inclusive tablets, apresentarem aplicativos de saúde e fitness, com a funcionalidades de medir os níveis de frequência cardíaca e de oxigênio no sangue. Esses aparelhos utilizam a câmera e o *flash* do aparelho para realizar tais medições[5]. Entretanto, apesar do uso crescente desses aplicativos, não se fala que os tais são recomendados ou sequer aprovados por especialistas da área.

Médicos afirmam que o uso desses aplicativos pode ser até danoso à saúde. Devido aos seus métodos de medição, eles não substituem os instrumentos de precisão, não sendo, assim, recomendados [6].

Tendo em vista a importância de medições como as tais à prática de atividades físicas, bem como a necessidade de instrumentos de precisão, propõe-se a construção de um aparelho, utilizando sensores de pulso infravermelho (IR - *infrared*).

Esse sensor apresenta, em si, mais parâmetros de regulação - em relação aos métodos de medição utilizando celulares, por exemplo -, portanto, pode apresentar uma precisão de resposta muito maior [7].

Sendo a confiabilidade dos dados apresentados um fator determinante e fundamental em um sensor de pulso, identifica-se, claramente, as vantagens desse projeto a ser desenvolvido.

3. DESENVOLVIMENTO

A solução proposta é utilizar um sensor de pulso, que detecta quando o coração bombeia o sangue através do corpo, pois a cada batida há uma onda de pulso (como uma onda de choque) que percorre todas as artérias até as extremidades do tecido capilar onde o sensor de pulso está conectado. O sangue real circula no corpo muito mais lento do que a onda de pulso viaja [15].

O desenvolvimento é dividido por aspectos de hardware e software.

3.1 ASPECTOS DE HARDWARE

Na Tabela 2 é apresentado os materiais utilizados na implementação do projeto, as quantidades de cada e uma breve descrição dos mesmos.

| Material | Qtd | Descrição/ Utilização |
|--------------------|-----|--|
| MSP-430G2553 | 1 | Microcontrolador |
| HC-05 | 1 | Módulo Bluetooth |
| Sensor de Pulso | 1 | Sinal do batimento cardíaco |
| Smartphone Android | 1 | Receber e enviar dados ao módulo Bluetooth |

Tabela 2 – Materiais utilizados na implementação do projeto.

• Módulo Bluetooth - HC-05

O periférico HC-05 consiste em um módulo Bluetooth SSP (*Serial Port Protocol* - Protocolo de Porta Serial). A tecnologia Bluetooth é muito utilizada para a comunicação sem fio entre dispositivos. No nosso caso, ela foi escolhida pela praticidade: desejava-se exibir os resultados do sensor na tela do *smartphone*; usando esse módulo Bluetooth, toda a transmissão serial de dados seria independente da conexão *wifi* do aparelho [8].

A imagem abaixo apresenta um esquemático da forma como devem ser feitas as conexões entre o módulo hc-05 e o MSP430.

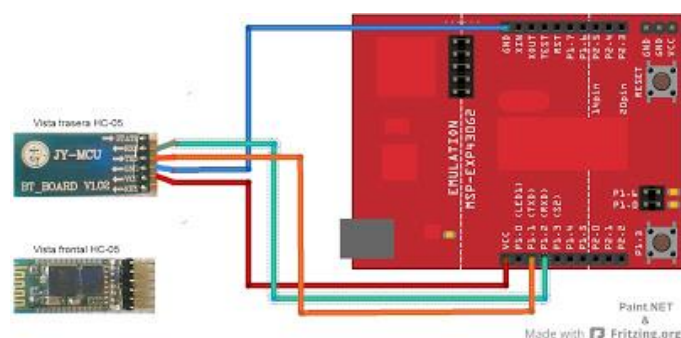


Figura 2 – Conexões hc-05e MSP430 [9].

Tradução das palavras em língua espanhola - *vista traseira*: vista traseira,

A tabela abaixo apresenta a relação das conexões entre as portas do microcontrolador e do módulo *Bluetooth* [9].

| MSP430 | HC-05 |
|--------|-------|
| VCC | VCC |
| P1.1 | TX |
| P1.2 | RX |
| GND | GND |

Tabela 3 – Conexões entre o microcontrolador e o módulo hc-05

• **Sensor de Pulso**

Este módulo possui um sensor óptico com amplificador, que faz um processo chamado fotopletismógrafo, que é um bem conhecido dispositivo médico usado para monitoramento não invasivo da frequência cardíaca. O sinal de pulso cardíaco que sai de um fotopletismógrafo é uma flutuação analógica da voltagem, e sua forma de onda é chamada de fotopletismograma, ou PPG. O sensor amplifica o sinal bruto do Sensor de Pulso anterior e normaliza a onda de pulso em torno de $V / 2$ (ponto médio na voltagem), respondendo a mudanças relativas na intensidade da luz. Se a quantidade de luz incidente no sensor permanecer constante, o valor do sinal permanecerá em (ou próximo de) 512 (ponto médio da faixa do ADC). Quanto mais luz, maior o sinal, quanto menos luz, menor o sinal. A luz do LED verde que é refletida de volta para o sensor muda durante cada pulso. [15]

Além disso, seu tamanho é bem reduzido (16 x 2,7mm), possui baixo consumo de energia (cerca de 4mA) e sua tensão de operação é de 3 a 5V. O sinal S fornece um valor analógico [10].



Figura 3 – Sensor de pulso.

O sensor possui 3 pinos, VCC, GND e S (sinal). Abaixo podemos ver a tabela e imagem mostrando as conexões entre o módulo sensor e o MSP430.

| MSP430 | Sensor de Pulso |
|-----------|-----------------|
| VCC | + |
| P1.3 (A3) | S |
| GND | - |

Tabela 4 – Conexões entre o microcontrolador e o sensor de pulso.

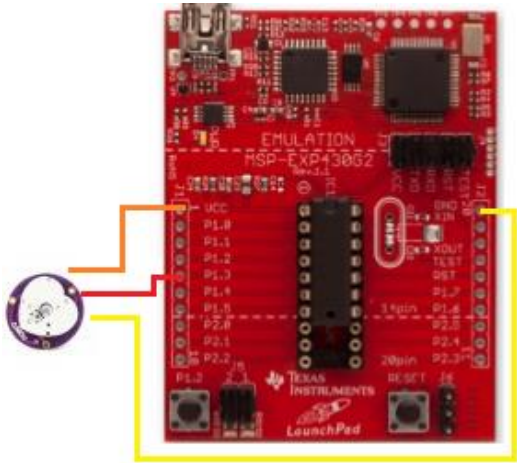


Figura 4 – Conexões Sensor de Pulso e MSP430.

3.2 ASPECTOS DE SOFTWARE

Para o sensor de pulso, foi utilizado uma porta com conversor \analógico digital, dada no pino P1.3 (A3). Foi utilizado e configurado um conversor A/D, para leitura do sensor. O conversor AD do msp430 é chamado de ADC10. O ADC10 é uma conversão analógica para 10 bits, tendo uma resolução de $2^{10} - 1$, ou seja, 1023 passos. [13].

Para calcularmos os batimentos cardíacos precisamos encontrar o momento instantâneo de batimento cardíaco. O código foi projetado para medir um intervalo de batimentos (IBI) pelo tempo entre os momentos em que o sinal cruza os 50% da amplitude da onda. Os batimentos por minuto (BPM) são calculados a cada batida de uma média de 10 tempos anteriores do IBI. Abaixo vemos a forma de onda emitida do sinal mostrando suas porcentagens de amplitude.

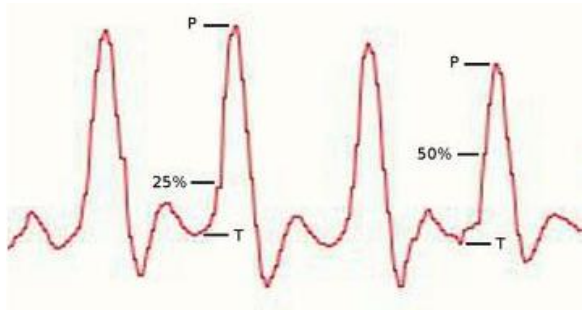


Figura 5 – Forma de onda sensor de pulso.

Para fazer a conversão AD do sinal é importante ter uma taxa de amostragem regular com resolução suficientemente alta para obter uma medição confiável do tempo entre cada batida, e com isso adotamos um

timer de 500 Hz, ou seja, 2ms. Na configuração do timer no código, utilizamos um clock SMCLK de 1 MHz, e fazendo a divisão, encontramos um valor inteiro para TACCR0 de 1000-1. Cada vez que o ciclo se completa, a flag de interrupção é ativada. [15]

A primeiro passo a se fazer quando entrar na interrupção é ativar a leitura do senso de pulso, através do conversor AD. Este conversor AD do msp430G2553 possui 10 bits, sendo chamado de ADC10, possuindo uma resolução de 1023 passos[13]. A variável que recebe a conversão é *Signal*. Após a conversão incrementamos a variável *sampleCounter* (coletor de amostras), que indica que se passou 2ms. No caso a variável *dt* corresponde ao instante de tempo atual menos o instante que ocorreu a última batida, contando o tempo que leva uma batida da outra.[15]

Para acompanhar os valores mais altos e mais baixos da onda, onde precisamos obter uma medida precisa de amplitude, utilizamos um limite inferior como forma de filtrar. Caso o valor do sinal seja menor que *tresh* (630) e se o intervalo de tempo em que não ocorreu uma batida for maior que 3/5 de IBI entra em duas condições. Caso seja menor que o *vale*, a variável *T* recebe o valor do sinal, caso o sinal seja maior que *tresh* e maior que o pico *P* (512), *P* recebe o valor do sinal. [15]

Para verificar se ocorreu um pulso ou não, temos a primeira condição, pois *dt* deve ser maior que 60, pois só após esse período uma nova batida deve ser procurada. Caso o sinal seja maior que *tresh* e o pulso seja falso, pois o intervalo de batida seja maior que *dt* de 60, então atribuímos o valor do pulso para verdadeiro e acendemos o led.

O *firstBeat* booleano é inicializado como *true* e o *secondBeat* é inicializado como *falso* na inicialização, então, na primeira vez em que encontramos uma batida, somos expulsos pelo retorno na condição do *primeiroBeat*. Isso acabará jogando fora a primeira leitura do IBI, porque é ruim. Na segunda vez, podemos confiar (mais ou menos) no IBI, e usá-lo para semear o array *rate* [] para começar com um BPM mais preciso. O BPM é derivado de uma média dos últimos 10 valores de IBI. [15]

Agora especificamente para calcularmos o BPM, primeiramente pegamos uma variável grande, *runningTotal*, para coletar os IBIs. Após isso o conteúdo da taxa [] é alterado e adicionado ao *runningTotal*. O mais antigo IBI (11 batidas atrás) cai fora da posição 0, e o IBI fresco é colocado na posição 9. Então é um

processo simples para calcular a média da matriz e calcular o BPM. [15]

No caso de não encontramos a não-batida, fazemos uma limpeza *pulsePin* e o pulso booleano. Em seguida, a amplitude da onda que acabou de passar é medida e o limiar é atualizado com a nova marca de 50%. *P* e *T* são redefinidos para o novo limite. Caso não ocorra nenhuma batida, as variáveis usadas para encontrar o batimento cardíaco serão reinicializadas para os valores de inicialização. [15]

Saindo da interrupção temos as funções do conversor AD, as configurações da comunicação serial do bluetooth, com *baud rate* de 9600, e a função que envia para o celular, enviando BPM, IBI e a leitura do conversor AD, *Signal*.

4. RESULTADOS

Para um melhor resultado do sensor, percebemos que ao colocarmos na orelha, ele converte com o pico bem mais alta que a batida falsa, deixando mais fácil de visualizar quando ocorre uma batida ou não. Comparando com um sensor de pulso original, percebemos um erro bastante pequeno quando comparamos os dois, pois estava dando uma diferença de 1 a 2 batimentos cardíacos no máximo em todos os testes feitos na orelha. Utilizando um aplicativo pelo celular percebemos que todos os valores chegaram corretamente e sem atraso pela transmissão serial do bluetooth

5. CONCLUSÃO

Neste projeto, tivemos muita dificuldade para trabalhar com este sensor de pulso, pois ele é muito sensível, alterando seu valor de pico dependendo da força aplicada sobre ele no momento de medição, além de possuir valores diferentes quando colocado em um dedo ou no outro, diferenciando também da orelha. Mesmo fazendo o tratamento adequado do sinal, ficaria um pouco inviável este modelo de sensor para monitorar frequência cardíaca durante atividades físicas, pois com sua sensibilidade, sua estabilidade pode ser afetada de acordo com os movimentos feitos, além de que com o melhor resultado obtido ser na orelha, poderia incomodar bastante os clientes que usarem este produto para esta finalidade.

Por outro lado, podemos ver a importância do microcontrolador MSP430 para qualquer tipo de

finalidade, pois ele pode cobrir uma enorme gama de áreas de projetos, possuindo tanto pinos digitais, pinos analógicos, clock's internos e pinos para externos, vários registradores, conversor AD, além de poder trabalhar em baixo consumo de energia, sendo este último podendo ser habilitado e desabilitado dentro de um mesmo código de acordo com a finalidade do projeto.

6. APÊNDICE

```
#include <msp430g2553.h>
#include <stdio.h>
#include <stdlib.h>

#define LED BIT0
#define LEDS BIT6
#define IN_AD BIT3
#define IN_AD_CH INCH_3

//Variáveis
int pulsePin = 0;
int blinkPin = 13;

// Varii
volatile int BPM;
volatile int Signal;
volatile int IBI = 60;
volatile boolean Pulse = false;
volatile boolean QS = false;

//Declaração das variáveis da interrupção
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 512;
volatile int T = 512;
volatile int thresh = 630;
volatile int amp = 0;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;

int main()
{
    mov.w #0x5480, &WDTCCTL;
    mov.b &CALBC1_1MHZ, &BCSCTL1;
    mov.b &CALDCO_1MHZ, &DCOCTL;

    serialInit();
    AD_Init();

    bis.b LED, &P1DIR;
    bis.b LEDS, &P1DIR;
    bic.b LED, &P1OUT;
    bic.b LEDS, &P1OUT;

    mov.w #0x3E7, &TACCR0;
    mov.w #0x212, &TA0CTL;
```

```
_BIS_SR(LPM0_bits + GIE);

__enable_interrupt();

return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TA0_ISR(void)
{
    //__disable_interrupt();
    ADC10CTL0 |= ENC + ADC10SC; // Manda começar
    uma conversao
    while((ADC10CTL0 & ADC10IFG)==0); // Espera a
    conversao ficar pronta
    Signal = ADC10MEM;

    sampleCounter += 2;
    int dt = sampleCounter - lastBeatTime;

    print(BPM, IBI, Signal);

    if(Signal < thresh && dt > (IBI/5)*3){
        if (Signal < T){
            T = Signal;
        }
    }

    if(Signal > thresh && Signal > P){
        P = Signal;
    }

    if (dt > 60)
    {
        if ( (Signal > thresh) && (Pulse == false) && (dt >
        (IBI/5)*3) )
        {
            Pulse = true;
            P1OUT |= LED;
            IBI = sampleCounter - lastBeatTime;
            lastBeatTime = sampleCounter;

            if(secondBeat)
            {
                secondBeat = false;
                for(int i=0; i<=9; i++)
                {
                    rate[i] = IBI;
                }
            }

            if(firstBeat)
            {
                firstBeat = false;
                secondBeat = true;
                _BIS_SR(GIE);
                return;
            }
        }

        word runningTotal = 0;
```


[2] CUNACIA, Pedro. **Descubra qual a frequência cardíaca ideal para emagrecer.** Disponível em: <<https://www.ativo.com/corrida-de-rua/iniciantes/fc-ideal-para-perder-peso/>>. Acesso em: 01 de Abril 2018 às 20:30

[3] BRUCE, Carlos. **Saiba qual é a melhor frequência cardíaca no treino para emagrecer.** Disponível em: <<https://www.tuasaude.com/frequencia-cardiaca-para-emagrecer/>>. Acesso em: 01 de Abril 2018 às 21:00

[4] COURSE WEBSITES: **Pulse Sensor.** Disponível em: <<https://courses.engr.illinois.edu/ece445/getfile.asp?id=6879>> Acesso em: 4/4/2018

[5] OLHAR DIGITAL. **App do iPhone utiliza flash e câmera para medir frequência cardíaca.** Disponível em: <<https://olhardigital.com.br/noticia/app-do-iphone-utiliza-flash-e-camera-para-medir-frequencia-cardiaca/14877>>. Acesso em: 02 de Abril 2018 às 16:52

[6] MAP OF SPORTS **Questioning the Value of Health Apps.** Disponível em: <https://well.blogs.nytimes.com/2015/03/16/health-apps-provide-pictures-if-not-proof-of-health/?ref=health&_r=0>. Acesso em: 02 de Abril 2018 às 16:33

[7] MAKEZINE. **Infrared Pulse Sensor.** Disponível em: <<https://makezine.com/projects/ir-pulse-sensor>>. Acesso em: 02 de Abril 2018 às 15:47

[8] Módulos Bluetooth HC05 e HC06 para comunicação com dispositivos móveis com Arduino. Disponível em: <<http://blog.eletrogate.com/modulos-bluetooth-hc05-e-hc06-para-comunicacao-com-dispositivos-moveis-com-arduino>>. Acesso em: 02 de Maio 2018.

[9] GALFAMA. Launchpad Bluetooth Android. Disponível em: <<https://galfama.blogspot.com.br/2013/02/control-de-leds-del-msp430-launchpad.html>> Acesso em: 02 de Maio 2018.

[10] BAÚ da Eletrônica. Sensor de Frequência Cardíaca. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-frequencia-cardiaca/#tab-description>>

[11] ENGINEERING AT ILLINOIS. **Pulse Sensor.** <<https://courses.engr.illinois.edu/ece445/getfile.asp?id=6879>>. Acesso em: 04 de Abril 2018.

[12] APP STORE. **Pulse Sensor.** <<https://itunes.apple.com/al/app/pulse-sensor/id761094899?mt=8>>. Acesso em: 02 de Maio 2018.

[13] Coder-Tronics - Tutorial MSP430 ADC. <<http://coder-tronics.com/msp430-adc-tutorial/>> Acesso em: 30 de Maio 2018.

[15] Pulse Sensor Amped
<https://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1> Acesso em: 04 de Junho 2018.

