

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. JUSTIFICATIVA

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: $220 - (\text{idade}) * 0,60$ e $220 - (\text{idade}) * 0,75$
Mulheres: $226 - (\text{idade}) * 0,60$ 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. 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.

3. 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.

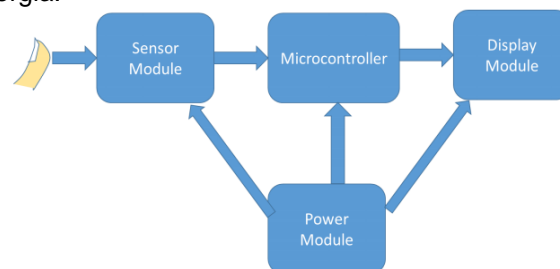


Figura 1 – Requisitos do projeto [4].

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

4. 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

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.

Segue abaixo uma tabela, que apresenta os materiais utilizados na implementação do projeto, as quantidades utilizadas de cada e uma breve descrição dos tais.

Tabela 2 – Materiais utilizados na implementação do projeto.

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].

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

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

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.

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

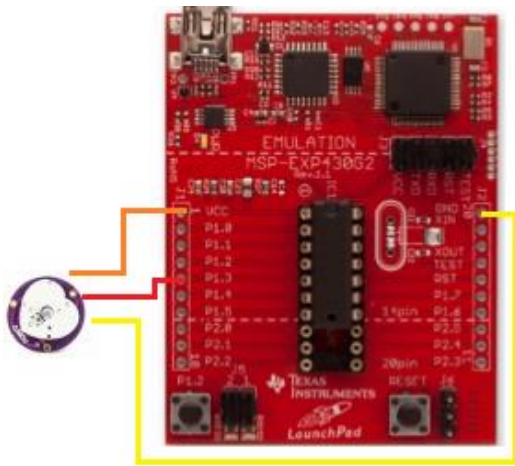


Figura 4 – Conexões Sensor de Pulso e MSP430.

6. ASPECTOS DE SOFTWARE

• Módulo Bluetooth - HC-05

Para esse protótipo funcional correspondente ao ponto de controle 3, foram testadas as funcionalidades de transmissão e recebimento de dados com o código em C, bem como o processamento de comandos desse módulo. Para isso, foi feito um código com o intuito de primeiramente enviar uma palavra por Bluetooth para um dispositivo móvel, de forma a testar se a comunicação entre eles esteja funcionando. O código se encontra comentado em anexo, e ele consegue se comunicar com o celular através de um aplicativo chamado “Bluetooth Terminal HC-05”, que se encontra de forma gratuita na Play Store para dispositivos Android.

Como foi compreendida a forma que a comunicação por meio desse módulo é utilizada e verificado o pleno funcionamento, os passos a seguir são: a realização do processamento e transmissão dos dados.

• Módulo Sensor de Pulso

Para este módulo, 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].

• Lógica do Código

Para fazer a conversão AD do sinal, foi utilizado um timer de 2ms, que é o suficiente para poder amostrar o sinal analógico de entrada. A cada 2ms é ativada a interrupção, onde é feito tanto a conversão AD quanto os cálculos dos batimentos por minuto.

Para ser feitos os cálculos, temos que pensar em uma forma para conseguirmos chegar nos batimentos por minuto. Por isso a ideia é pegar dois

picos, ou seja, duas batidas, calcular o tempo de duração dos dois picos por completo e fazer a divisão de 2 minutos pelo tempo de duas batidas, obtendo por fim os batimentos por minuto. Foram feitos 2 while, um para cada pulso. Ao inicializar a variável pulso com valor 1, ela já entrará automaticamente no primeiro while, onde pode ser encontrado um if e else, para somar o tempo em que decorre o primeiro pulso. Ao final a variável pulso recebe o valor 2, indo para o segundo pulso. Após passar por um if e else para somar os milissegundos, a variável pulso recebe o valor 0 para que não entre em nenhum dos while, seguindo o procedimento de conta.

No caso 1 minuto são 60 segundos, então 2 minutos são 120 segundos, que é equivalente a 120000 milissegundos, valor utilizado para realização da conta. Com isso os batimentos cardíacos são os 120000 milissegundos dividido pelo tempo das duas batidas, guardada na variável milissegundos. Após o cálculo a variável milissegundos é zerada para repetição da interrupção.

Como o sensor é muito sensível, foi feito um if para que não entrasse nenhum número de bpm absurdo na média, minimizando o erro ao mostrar o resultado para o indivíduo. Após este valor estar dentro do normal, é feita a distribuição dos valores para serem feitas as médias, onde são armazenados seis valores, indo de A e F, e depois sendo o mesmo dividido por 6. Com 6 médias de 2 batimentos cardíacos temos um erro bem pequeno associado à resposta. Foi acrescentada uma parte do código em assembly, como visto no anexo.

7. ANEXOS

• Código:

```
#include <msp430g2553.h>
#include <intrinsics.h>
#include <legacymsp430.h>
```

```
//Declaração geral
#define sensor BIT3;
unsigned int saida;
#define LEDS BIT6;
```

```
//Declaração do bluetooth
int j,i,w =0;
unsigned int nadc=0;
unsigned char b[4], c[5];
unsigned char a=2;
```

```
//Declaração para cálculo dos BPM
int Pulso = 1;
int ligado = 1;
float soma_media = 0;
float media = 0;
float batimentos = 0;
float bpm = 0;
float milissegundos= 0;
```

```

float media_A = 0;
float media_B = 0;
float media_C = 0;
float media_D = 0;
float media_E = 0;
float media_F = 0;

//Função Bluetooth
void serialInit()
{
    P1SEL= BIT1 + BIT2; //P1.1 = RXD P1.2=TXD
    P1SEL2= BIT1 +BIT2; // P1.1=RXD & P1.2=TXD
    UCA0CTL1|= UCSSEL_2; // SMCLK
    UCA0BR0=104; // BAUDRATE AT 1 MHz 9600
    UCA0BR1=0;//1MHz 9600
    UCA0MCTL= UCBRS0; // MODULATION UCBRSx=1
    UCA0CTL1&=~UCSWRST; // ** INITIALIZE USCI
STATE MACHINE
    IE2|= UCA0RXIE; // ENABLE VSCI_A0 RX
INTERRUPT
}

unsigned char serialRead()
{
    while(!(IFG2&UCA0RXIFG)); //USCI_A0 RX buffer
ready ?
    return UCA0RXBUF;
}

void serialWrite(unsigned char c)
{
    while(!(IFG2&UCA0TXIFG)); // USCI_A0 TX buffer
ready ?
    UCA0TXBUF=c; // TX
}

void serialwriteString(const char *str)
{
    while(*str)
        serialWrite(*str++);
}

void serial_write_int(unsigned int temp)
{
    for( i=0;i<4;i++)
    {
        b[i]=temp%10;
        //serialWrite(b[i]);
        temp=temp/10;
    }

    for(j=3;j>=0;j--)
    {
        serialWrite(b[j] + 48);
    }

    serialWrite(' ');
    serialWrite('\n');
}

serial_write_float(float v)

```

```

{
    w = v*100;

    for(i=0;i<4;i++)
    {
        c[i]=w%10;
        w=w/10;
    }

    serialWrite(c[3]+48);
    serialWrite(c[2] +48);
    serialWrite('.');
    serialWrite(c[1] + 48);
    serialWrite(c[0] + 48);
    serialWrite(' ');
    serialWrite('\n');
}

void main(){
    mov.w #0x5480, &WDTCTL;
    mov.b &CALBC1_1MHZ, &BCSCTL1;
    mov.b &CALDCO_1MHZ, &DCOCTL;
    bis.b #0x8, &P1SEL;
    bis.b #0x40, &P1DIR;
    bic.b #0x40, &P1OUT;
    bic.b #0x41, &P1DIR;

    mov.w #0x3E7, &TACCR0;
    mov.w &TACCR0, &TACCR1;
    mov.w #0xE0, &TACCTL1;
    mov.w #0x212, &TACTL;

    ADC10AE0 = sensor;
    ADC10CTL0 = SREF_0 + ADC10SHT_0 + ADC10ON
+ ADC10IE;
    ADC10CTL1 = INCH_3 + SHS_1 + ADC10DIV_0 +
ADC10SSEL_3 + CONSEQ_2;
    _BIS_SR(LPM0_bits+GIE);
}

interrupt(TIMER0_A1_VECTOR) TA0_ISR(void){

    while(Pulso == 1){
        ADC10CTL0 |= ENC + ADC10SC;
        saida = ADC10MEM;

        if(saida > 525 && saida < 560){
            P1OUT |= LEDS;
            while(saida > 525 && saida < 560){
                ADC10CTL0 |= ENC + ADC10SC;
                saida = ADC10MEM;
                milisegundos ++;
            }
        }

        else{
            P1OUT &= ~LEDS;

```

```

while(saida > 525 && saida < 560){
    ADC10CTL0 |= ENC + ADC10SC;
    saida = ADC10MEM;
    milisegundos ++;
}
Pulso = 2;
}

while(Pulso == 2){
    ADC10CTL0 |= ENC + ADC10SC;
    saida = ADC10MEM;

    if(saida > 525 && saida < 560){
        P1OUT |= LEDS;
        while(saida > 525 && saida < 560){
            ADC10CTL0 |= ENC + ADC10SC;
            saida = ADC10MEM;
            milisegundos ++;
        }
    }

    else{
        P1OUT &= ~LEDS;
        while(saida > 525 && saida < 560){
            ADC10CTL0 |= ENC + ADC10SC;
            saida = ADC10MEM;
            milisegundos ++;
        }
    }
    Pulso = 0;
}

batimentos = 120000/milisegundos;
milisegundos = 0;

if(batimentos > 45 && batimentos < 125){
    if(media_A!=0 && media_B!=0 && media_C!=0
    && media_D!=0 && media_E!=0 && media_F!=0)
    {
        soma_media = media_A + media_B +
media_C + media_D + media_E + media_F!=0;
        bpm = soma_media/6;

        media_A = 0;
        media_B = 0;
        media_C = 0;
        media_D = 0;
        media_E = 0;
        media_F = 0;

        serialInit();
        serial_write_int(bpm);
        Pulso = 1;
    }
    else if(media_A == 0){
        media_A = batimentos;
    }
    else if(media_A!=0 && media_C==0 &&
media_D==0 && media_E==0 && media_F==0){
        media_B = batimentos;

```

```

    }
    else if(media_A!=0 && media_B!=0)
    {
        media_C = batimentos;
    }
    else if(media_A!=0 && media_B!=0 &&
media_C!=0)
    {
        media_D = batimentos;
    }
    else if(media_A!=0 && media_B!=0 &&
media_C!=0 && media_D!=0)
    {
        media_E = batimentos;
    }
    else if(media_A!=0 && media_B!=0 &&
media_C!=0 && media_D!=0 && media_E!=0)
    {
        media_F = batimentos;
    }
}
Pulso = 1;
}

#pragma vector = ADC10_VECTOR
__interrupt void ADC10_ISR(void)

{
    _bic_SR_register_on_exit(CPUOFF);
}

```

8. REFERÊNCIAS

[1] - PORTAL BRASIL. **Obesidade cresce 60% em dez anos no Brasil.** <<http://www.brasil.gov.br/saude/2017/04/obesidade-cresce-60-em-dez-anos-no-brasil>>

[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.

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