

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.

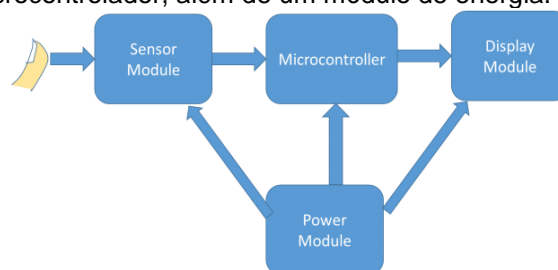


Figura 1 – Requisitos do projeto [4].

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 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.

5. ASPECTOS DE HARDWARE

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

Material	Qtd	Descrição/ Utilização
MSP-430G2553	1	Microcontrolador
HC-05	1	Módulo Bluetooth
Sensor de Pulso	1	Sinal do batimento cardíaco
Smartphone Android	1	Para 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: deseja-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.

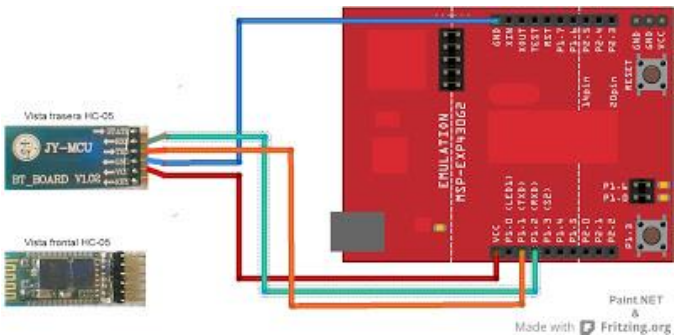


Figura 2 – Conexões hc-05e MSP430 [9].

Tradução das palavras em língua espanhola - *vista trasera*: 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 utiliza luz infravermelha, e que envia o sinal analógico para o microcontrolador. 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.

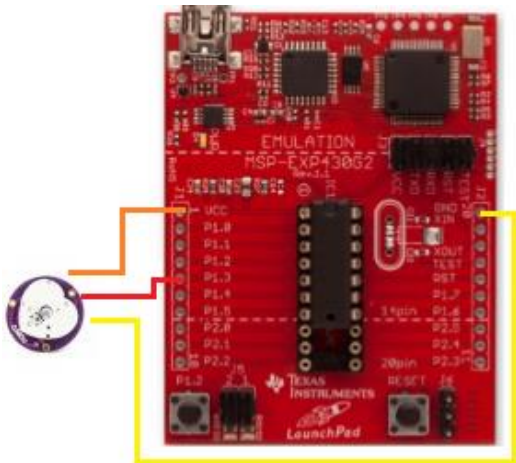


Figura 4 – Conexões Sensor de Pulso e MSP430.

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 “*Blue Therm*”, que se encontra de forma gratuita na Play Store para dispositivos Android.

A imagem abaixo ilustra parte dos nossos objetivos com a transmissão de dados para o *smartphone*, após de fazer todo o processamento das informações.

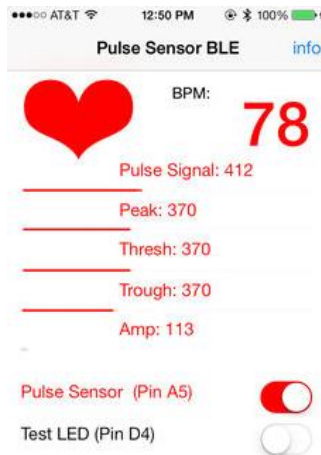


Figura 5 - Exemplo de exibição dos dados em um *Smartphone* [12]

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 dos nossos dados, e a configuração da plataforma, conforme o exemplo exibido.

• 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. [14]

7. ANEXOS

- Código utilizado para testar o módulo Bluetooth hc-05.[9]

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

```
int j,i,w =0;
```

```
unsigned int nadc=0;
unsigned char b[4], c[5];
unsigned char a=2;
```

```
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()
{
    WDTCTL = WDTPW + WDTHOLD;
    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    P1DIR = 0x41;
    P1OUT=0x00;
    serialInit();
    serialwriteString("hello !!");
    serialWrite("\n");
}
-----

    • Código utilizado para testar o sensor de pulso

-----

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

#define sensor BIT3;
#define LEDRED BIT6;

unsigned int bpm;

void convAD(void)
{
    TACCTL1 = OUTMOD_7;
    TACCR0 = 400-1;
    TACCR1 = TACCR0/2;
    TACTL = TASSEL_2 + ID_0 + MC_1;

    ADC10CTL0 = SREF_0 + ADC10SHT_0 + ADC10ON +
    ADC10IE;

    //ADC10CTL - Registrador 0; //SREF - Tensão de referência
    VCC e VSS;
    //ADC10SHT - Quantidade de periodos do clock para
    aproximações 4XADC10CLKs;
    //ADC10ON - Liga o ADC10 e habilita interrupções;
    //ADC10IE - Ativa Interrupção;

    ADC10CTL1 = INCH_3 + ADC10DIV_0 + ADC10SSEL_3
    + CONSEQ_2 + SHS_0;
    //INCH_3 - Seleção entrada analógica BIT 3;
    //ADC10DIV - Divisor de Clock /1;
    //ADC10SSEL - Seleção do klok para aproximações -
    SMCLK;
    //CONSEQ - Modo de operação - Repetir canal único;

```

//SHS - Seleção que dá inicio à conversão sucessiva
ADC10SC bit

```

ADC10AE0 |= sensor;
//Ativa conversão;
}

```

```

int main (void)
{
    WDTCTL = WDTPW + WDTHOLD;
    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    BCSCTL2&=~(DIVS_3);

```

```

    convAD();
    _BIS_SR(GIE);

```

```

while(1)
{
    ADC10CTL0 |= ENC + ADC10SC;
    // Inicio da amostragem e conversão
    bpm = ADC10MEM;

```

```

}
}
//ADC10 #pragma vector = ADC10_VECTOR
__interrupt void ADC10_ISR (void)
{
    __bic_SR_register_on_exit(CPUOFF); //Desabilita Master
    Clock
    //Retorna ao modo ativo
}
-----

```

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/>>