

Cadastro eletrônico para academias

Erick Antonio Correa dos Reis

15/0034156

Faculdade do Gama

Universidade de Brasília

St. Leste Projeção A - Gama Leste, Brasília - DF

Email: erickcorrareis@gmail.com

Tiago Avelino Ribeiro da Silva

15/0022662

Faculdade do Gama

Universidade de Brasília

St. Leste Projeção A - Gama Leste, Brasília - DF

Email: tiago.avelino1997@gmail.com

RESUMO -

Palavras-chave - MSP430; Academia; Microcontrolador; Monitor exercícios; Acompanhamento de exercícios.

I. INTRODUÇÃO

As academias costumam possuir um sistema de cadastro em software dos alunos o que geralmente contabiliza apenas os valores de mensalidade. Para a utilização dos equipamentos dispostos na academia, o instrutor da academia, anota em um papel o instrumento e número de repetições e o exercício a ser desenvolvido pelo aluno.

Dessa forma foi pensado em um cadastro eletrônico que visa computar todos esses valores, para a estação de musculação da academia, tendo em vista os diversos tipos de exercícios dispostos nela, além disso será implementado a duração do exercício e o número de repetições por série. Para um melhor desempenho do aluno também serão feitas as medições de batimentos cardíacos e será feita uma análise do movimento do aluno.

Os batimentos cardíacos serão analisados enquanto o treino é realizado na estação de musculação informando-o este valor para que não haja um inadequado desenvolvimento do exercício.

Já a verificação do movimento será feita através de um sensor de ultrassom que verificará quando a variação da altura do peso na máquina é alcançada no sensor através dessa variação será decrementado o número de repetições da série avaliada. Além disso caso haja uma falha muscular, o valor total de repetições feitas será enviado para o instrutor esses valores armazenados auxiliam o instrutor em relação aos melhores tipos de técnicas para melhorar o treino do aluno.

II. OBJETIVOS

Construir um cadastro eletrônico para academia, programado no Launchpad MSP430, buscando um melhor desempenho do aluno.

III. DESCRIÇÃO

Visando implementar um equipamento que além de monitorar a frequência cardíaca do aluno durante todas as atividades na academia e mostrar detalhadamente como esta deve ser conduzida, este projeto também consegue a integração com uma estação de musculação que por vez deve fornecer com precisão se o número de repetições realizada pelo aluno está de acordo com aquilo que foi proposto pelo instrutor.

Os detalhes dos componentes e sensores a serem utilizados neste projeto podem ser vistos a seguir:

- *Monitor de Pulso*



O monitor de pulso ou sensor de batimentos cardíacos é uma de nossas bases, visto que este deve funcionar de forma contínua.

Seu funcionamento é bem simples, contando com apenas 3 pinos, dois deles para alimentação (Vcc e GND) e uma saída

que pode ser lida por uma porta do MSP430 utilizando seu conversor analógico digital.

Este sensor pode ser alimentado com tensão de 3.3V a 5V utilizando cerca de 4mA.

- *Display LCD*



Para visualizar os exercícios a serem realizados e os valores obtidos nos sensores será utilizado o Display LCD Nokia 5110 em que se pode ter em uma mesma tela gráficos e textos com uma resolução de 84 X 48 pixels. Por se tratar de um display gráfico e utiliza comunicação serial do tipo SPI, em que se utiliza dois sinais de comunicação de dados, e que toda comunicação de dados acontece em ambas direções e é necessário enviar uma sequência de bits para desenhar no display a forma desejada. Foi encontrada uma biblioteca com a sequências dos vectores que desenharam cada caracter (números, letras e símbolos) no display.

Todas as funções para inicialização e envio de dados para display via comunicação serial foram adaptadas para funcionar de maneira mais simples e direta.

Tabela 1: Configuração de pinos Display LCD

Display LCD	MSP 430
VCC	VCC
GND	GND
SCE	P1.0
RST	---
D/C	P1.1
DN (MOSI)	P1.7
SCLK	P1.5
LED	---

- *Sensor ultrassônico HC-SR04*



Com o objetivo de oferecer a integração com a estação de musculação, será instalado um sensor ultrasônico para que possa se obter os dados referentes às repetições a serem realizadas.

Este sensor deve ser posicionado de forma a registrar a movimentação dos pesos que ficam na parte posterior do equipamento, possibilitando assim analisar se uma repetição foi feita. Por conta de sua localização será possível obter essa informação independentemente do exercício realizado na estação já que todos eles proporcionam uma movimentação nos pesos.

O sensor ultrassônico é capaz de medir distância porém sua utilização será restrita apenas para informar de se os pesos se moveram ou ficaram estáticos.

A implementação deste sensor consiste basicamente em um temporizador que mede o intervalo entre envio e recepção de um sinal ultrassônico.

R

IV. RESULTADOS

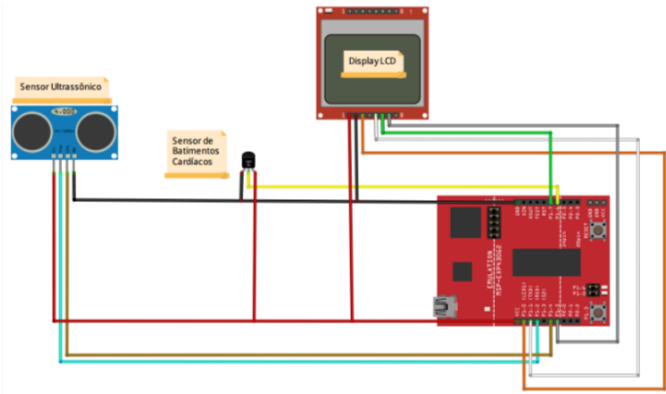
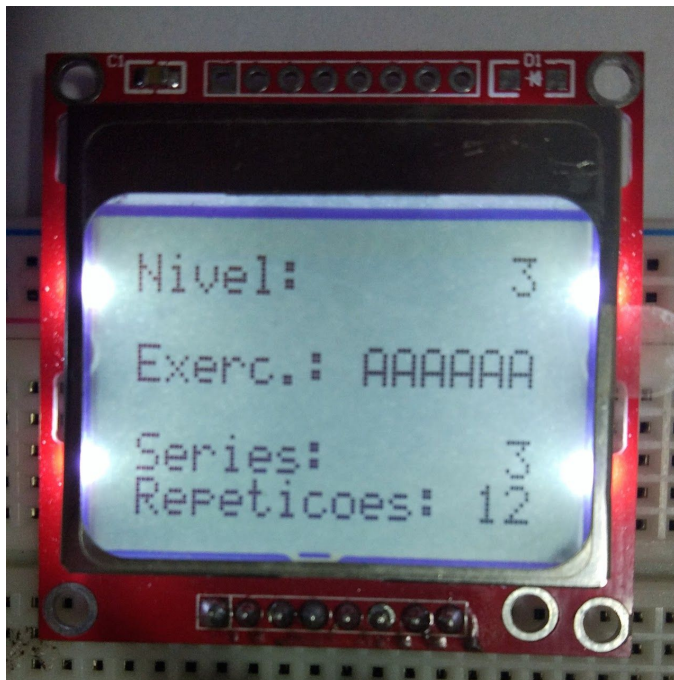
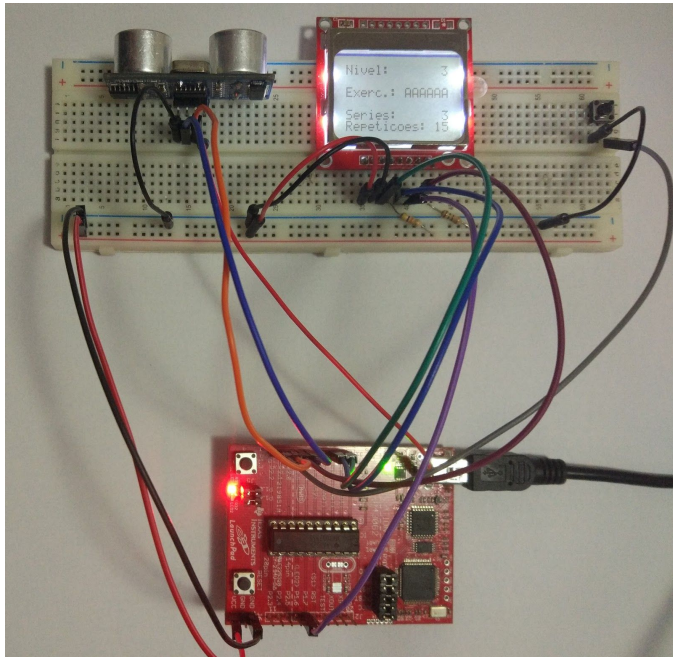


Figura 1: Esquemático atual

Seguindo os esquemático apresentado acima conseguimos implementar o projeto da seguinte forma (figura xxxx).



Através das imagens é possível visualizar de forma clara todas as informações necessárias durante a realização de exercícios em uma estação de musculação.

Todos os exercícios, séries e repetições podem ser configurados através do arquivo *Exercicios.h*, neste arquivo temos algumas matrizes capazes de armazenar não só os vários exercícios mas também diferentes níveis, possibilitando

que o equipamento seja utilizado de forma mais geral sem muitas configurações adicionais.

V. CONCLUSÃO

Com os resultados obtidos até o momento é possível ver que o projeto se encontra praticamente em sua fase final, embora a implementação de sensor tenha sido cancelada devido ao alto nível de ruído gerado pelo mesmo.

Com relação a comunicação sem fio optamos pela não implementação da mesma visto que sua utilização não nos mostrou muitos benefícios tanto para os alunos quanto para o instrutor.

VI. REVISÃO BIBLIOGRÁFICA

[1] GUIMARÃES, Anderson Eduardo; BARSOTTINI, Daniel; VIEIRA, Rodolfo de Paula. **ADEQUAÇÃO DAS ATIVIDADES FÍSICAS PROPOSTAS EM ACADEMIAS EM FUNÇÃO DA AVALIAÇÃO FÍSICA**. 2004. Disponível em: http://cronos.univap.br/cd/INIC_2004/trabalhos/epg/pdf/EPG4-10R.pdf. Acesso em: 02 set. 2017

VII. ANEXOS

• *Exercicios.h*

```
#ifndef EXERCICIOS_H_
#define EXERCICIOS_H_
const char nome[3][7] = {"AAAAAAA",
                        {"BBBBBBB"},
                        {"CCCCCCC"};

//series: exercicio X nível
const int series[3][6] = {{2, 2, 2, 2, 3, 3},
                        {2, 2, 3, 3, 4, 4},
                        {2, 2, 2, 2, 3, 3}};

//repeticoes: exercicio X nível
const int repeticoes[3][6] = {{1, 10, 15, 15, 10, 15},
                        {1, 15, 10, 15, 10, 15},
                        {1, 10, 15, 15, 10, 15}};

#endif
```

• *main.c*

```
#include "msp430g2553.h"
#include <Display.h>
#include <Exercicios.h>
```

```
void tela_padrao();
int medir_dist();
void reset_ex();
void exer();
void fim();
void config_display();
void config_sensor();
void config_botoes();
```

```

#define BTN BIT3 //Botão Nivel
#define Reset BIT1 //Botão Reset
#define ECHO BIT4 //Echo sensor ultras.
#define TRIG BIT0 //Trig sensor ultras.
int miliseconds;
long sensor;
unsigned int nivel, ex, sr, rep;

void main(void) {
    WDTCTL = WDTPW + WDTHOLD; // Desabilitar WDT
    BCSCTL1 = CALBC1_1MHZ; // Clock 1MHz
    DCOCTL = CALDCO_1MHZ;

    config_display();
    config_sensor();
    config_botoes();
    initLCD();
    clearLCD();
    tela_padrao();
    reset_ex();

    while(1){
        exer();
        setAddr(72, 0);
        writeIntToLCD(nivel + 1);
        setAddr(48, 2);
        writeStringToLCD(&nome[ex][0]);
        setAddr(72, 4);
        writeIntToLCD(sr + 1);
        setAddr(72, 5);
        writeIntToLCD(rep);
    }
}

#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void){
    if(P1IFG&ECHO){ // Interrupção sensor ultrassônico
        if(!(P1IES&ECHO)){
            TA0CTL |= TACLK;
            miliseconds = 0;
            P1IES |= ECHO;
        }
        else sensor = (long)miliseconds*1000 + (long)TAR;
        P1IFG &= ~ECHO;
    }
    if(P1IFG&BTN){ // Interrupção botão nível
        while((P1IN&BTN)==0);
        nivel++;
        if (nivel==6) nivel = 0;
        reset_ex();
        P1IFG &= ~BTN;
    }
}

#pragma vector=PORT2_VECTOR

```

```

__interrupt void Port_2(void){
    if(P2IFG&Reset){ // Interrupção botão reset
        while((P2IN&Reset)==0);
        WDTCTL = 0;
    }
}

#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A0 (void){ //Contagem de tempo do
sensor ultrassônico
    miliseconds++;
    TA0CTL &= ~TAIFG;
}

void config_display(){
    P1OUT |= LCD5110_SCE_PIN + LCD5110_DC_PIN;
    P1DIR |= LCD5110_SCE_PIN + LCD5110_DC_PIN;
    P1SEL |= LCD5110_SCLK_PIN + LCD5110_DN_PIN;
    P1SEL2 |= LCD5110_SCLK_PIN + LCD5110_DN_PIN;
    UCB0CTL0 |= UCCKPH + UCMSB + UCMST +
UCSYNC; // 3-pin, 8-bit SPI master
    UCB0CTL1 |= UCSSEL_2; // SMCLK
    UCB0BR0 |= 0x01; // 1:1
    UCB0BR1 = 0;
    UCB0CTL1 &= ~UCSWRST; // clear SW
}

void config_sensor(){
    CCTL0 = CCIE;
    TA0CCR0 = 1000;
    TA0CTL = TASSEL_2 + MC_1;
    P2DIR |= TRIG;
    P1DIR &= ~ECHO;
    P1IE |= ECHO;
    P1IES &= ~ECHO;
}

void config_botoes(){
    P1DIR &= ~BTN;
    P1REN |= BTN;
    P1OUT |= BTN;
    P1IES |= BTN;
    P1IE |= BTN;
    P2DIR &= ~Reset;
    P2REN |= Reset;
    P2OUT |= Reset;
    P2IES |= Reset;
    P2IE |= Reset;
    _BIS_SR(GIE);
}

int medir_dist(){
    int distancia;
    P2OUT |= TRIG; // gera um pulso
    __delay_cycles(10); // espera 10 us
    P2OUT &= ~TRIG; // para o pulso
    P1IES &= ~ECHO;
    __delay_cycles(30000); // tempo limite do sensor
    distancia = sensor/58; // convertendo tempo em cm
    return distancia;
}

```

```

}
void tela_padrao(){
    setAddr(0, 0);
    writeStringToLCD("Nivel:");
    setAddr(0, 2);
    writeStringToLCD("Exerc.:");
    setAddr(0, 4);
    writeStringToLCD("Series:");
    setAddr(0, 5);
    writeStringToLCD("Repeticoes:");
}
void exer(){
    if(sr==0 && rep==0){
        __delay_cycles(800000);
        ex++;
        sr = series[ex][nivel];
        rep = repeticoes[ex][nivel];
    }
    if(rep==0){
        __delay_cycles(800000);
        sr--;
        rep = repeticoes[ex][nivel];
    }
    if(medir_dist()<=30){
        rep--;
        while(medir_dist()<=40) __delay_cycles(100000);
    }
    if(ex>2){
        fim();
    }
}
void reset_ex(){
    ex = 0;
    sr = series[ex][nivel];
    rep = repeticoes[ex][nivel];
}
void fim(){
    __delay_cycles(500000);
    clearLCD();
    reset_ex();
    setAddr(0, 2);
    writeStringToLCD(" Exercicios ");
    setAddr(0, 3);
    writeStringToLCD(" Concluidos ");
    _BIS_SR(LPM4_bits);
}

```