

Fisioterapia ludica com MSP430

Uso de MSP430 para criação de uma fisioterapia ludica

Autor: Antônio Aldísio de Sousa Alves Fereira Filho
Universidade de Brasília
Microcontroladores e Microprocessadores
Gama- DF
aldisiofilho@gmail.com

Authors Name/s per 2nd Affiliation (*Author*)
line 1 (of *Affiliation*): dept. name of organization
line 2-name of organization, acronyms acceptable
line 3-City, Country
line 4-e-mail address if desired

Abstract—

Keywords—*component; formatting; style; styling; insert (key words)*

I. Introdução

A fisioterapia é uma das principais formas de reabilitação para uma pessoa vítima de doenças, traumas ou acidentes. Com ela, é possível obter a ajuda necessária para que essa pessoa volte a executar tarefas que, sem medidas auxiliares, seriam muitas vezes impossíveis e as cujas chances de recuperação seriam cada vez mais improváveis.

De acordo com o Conselho Nacional de Fisioterapia (COFFITO), podemos definir a fisioterapia: “*É uma ciência da saúde que estuda, previne e trata os distúrbios cinéticos funcionais intercorrentes em órgãos e sistemas do corpo humano, gerados por alterações genéticas, por traumas e por doenças adquiridas, na atenção básica, média complexidade e alta complexidade. Fundamenta suas ações em mecanismos terapêuticos próprios, sistematizados pelos estudos da biologia, das ciências morfológicas, das ciências fisiológicas, das patologias, da bioquímica, da biofísica, da biomecânica, da cinesia, da sinergia funcional, e da cinesia patológica de órgãos e sistemas do corpo humano e as disciplinas comportamentais e sociais.*”. A partir dessa definição, observa-se a vasta aplicação da fisioterapia no tratamento ou na prevenção de diversos tipos de doenças e traumas.

Dentre essas aplicações, destaca-se neste artigo aquela referente ao tratamento de traumas ou de doenças que impossibilitaram os movimentos dos membros superiores tais como ombros, cotovelos e pulsos. As soluções fisioterapêuticas atuais, nesses casos, têm como um dos principais obstáculos a falta de aparelhos ou métodos que incentivam pacientes ao tratamento ativo correto, especialmente ao público infantil e aos portadores de necessidades especiais, o que dificulta o tratamento das patologias e impõem necessidades de fisioterapeutas com especialidade em pedagogia até casos não muito complexos, o que em muitas ocasiões se torna desnecessário e exigem demanda desses profissionais, o que pode elevar os custos desses tratamentos.

A partir dessa problemática, o projeto em questão propõe uma solução lúdica que consiste em um aparelho eletrônico-visual em forma de jogo interativo controlado por

meio de movimentos específicos de acordo com o tratamento fisiológico dos membros superiores, de modo que a progressão e a evolução do tratamento avancem à medida que se completam os níveis e cada fase do jogo.

Dessa forma, espera-se obter uma solução eficiente no tratamento fisiológico de contusões ou de sequelas geradas a partir de alguma doença.

II. Justificativa

Como a fisioterapia é um trabalho de repetição de movimentos. Para crianças e idoso que realização alguma fisioterapia podem considerar que essa atividade como monótona e entediante.

A partir disso existe uma linha de trabalho na fisioterapia que desenvolve atividades lúdicas durante o atendimento medico assim o paciente não visualiza que estar realizando atividade. Segundo Rafael Dias, a atividade lúdica apresenta um elemento motivador dentro do processo de reabilitação do paciente. O paciente que desenvolve atividade lúdica não encara o tratamento tão seriamente como se fosse ao modelo tradicional.

Boa parte dos estudos realizados utiliza o console Wii U para desenvolvimento das atividades lúdicas. O valor monetário e acessibilidade do console dentro de um centro de fisioterapia ou hospital pode ficar inviável para o tratamento. O desenvolvimento de um sistema que utiliza o MSP430 e a possibilidade criação de mecanismo de baixo custo e sendo portátil. A diferença de portabilidade pode ser mostrada nas imagens abaixo:



Figura 1 – pessoas jogando wii U



Figura 2 – ideia de como será o sistema com MSP430

O projeto não visa construir um tablete para a fisioterapia, mas a ideia de portabilidade é comparada ao um tablet, pois esse motivo utilizamos uma foto com tablet para mostrar a ideia do projeto.

Porque não usar um tablet? Um sistema pronto e que é necessário apenas criar um software em vez de criar uma a partir do uso de MSP430. A utilização do msp430 irá ter um sistema único dentro do produto e podemos criar uma adaptação melhor para o usuário e o baixo custo que iremos trabalhar.

Por fim, iremos utilizar o MSP430 para desenvolver o sistema de fisioterapia lúdica para atividades nos membros do pulso e cotovelo. Esse sistema será um jogo (ainda não definido) onde será a base de uma matriz de *led* onde o paciente irá guiar o *led* a partir de movimentos rastreados via acelerômetro.

III. Objetivo

O desenvolvimento desse projeto tem como objetivo de construir um jogo que auxilie na fisioterapia de membros superiores, pulso, cotovelo e ombro. No qual o usuário não perceba que esteja realizando a fisioterapia.

Assim construindo um sistema que possa trabalhar com grupo de pessoa que tenha dificuldade com a fisioterapia tradicional.

Portanto, objetivo desse projeto é criar uma imersão para o paciente, onde ele possa trabalhar

movimentos fisioterápicos sem que perceba que esteja realizando a atividade de fisioterapia.

IV. Requisitos

Tendo em vista que o projeto visa implementar de forma eletrônica um tratamento lúdico fisioterapêutico para auxiliar o paciente que possui dificuldades de seguir o tratamento convencional, além de outras características, os requisitos são:

1. Conter um programa atrativo ao público alvo.
2. Conter sistema de sensor eletrônico que detecta os movimentos corretos do tratamento.
3. Acompanhar a evolução do paciente conforme o uso do equipamento.
4. O Produto deve ser constituído de um material leve e portátil.

V. Benefício

O benefício desse projeto é desenvolver um jogo de baixo custo onde possamos implementar em hospitais públicos para o tratamento de pacientes. Além do baixo custo temos a questão da mobilidade do equipamento que pode ser carregador igualmente ao um tablet.

Agora pensando no desenvolvimento da matéria iremos poder compreender melhor o uso do MSP430 na prática e trabalhando com algum sensor.

VI. Desenvolvimento

A metodologia principal utilizada para a elaboração dos códigos foi por meio de engenharia reversa. Em todos os códigos foi utilizada a ferramenta Energia para testar as funcionalidades do sistema antes de configurá-lo a nível de registradores por meio do software Code Composer Studio.

Nos códigos envolvendo o controle das matrizes de leds foram testados diversos códigos envolvendo o controle de matrizes de leds. Foram compilados trechos de códigos de diversos programas de exemplos base pela companhia desenvolvedora do Energia e, a partir da ferramenta Code Composer Studio, foram traduzidos para o nível de registradores com algumas adaptações nos códigos originais.

A. Materiais utilizados

O controle de 64 Leds, será feito a partir do CI MAX7219 com o envio serial de dados por meio de uma saída digital. A matriz bicolor utilizada foi a MXM-8X8-EG. O processamento será feito a partir da lauchpad contendo MSP430G2553. Aquisição de dados referentes aos membros superiores será feito a partir de do módulo MPU-6050.



Figura 3 - CI controlador da matriz de led (MAX7219).

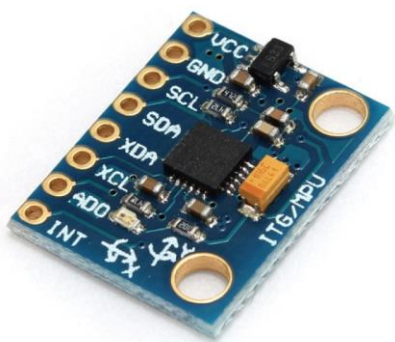


Figura 4 – Módulo giroscópio/acelerômetro (MPU6050).

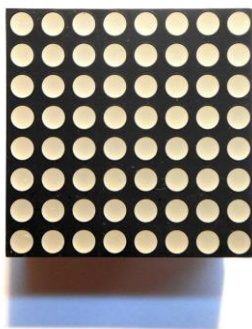


Figura 5 – Matriz de led bicolor (MXM-8X8-EG)

B. Métodos

Os componentes foram dispostos em protoboard e conectados por meio de “jumpers”. Na ligação do CI MAX7219 deve ser utilizada alimentação de 5V, bem como as entradas, cuja tensão mínima suportada é 4V. Para isso, foi conectado à saída do MSP transistor para amplificar a tensão das saídas do MSP430 e utilizamos *push button* para deslocamento da bolinha dentro da matriz de led.

Foi construído uma lógica apenas para uma matriz de led, onde acedemos todos os led dos cantos, porém um led

fica apagado indicando o buraco onde o jogador tem que encaminhar a bolinha que acende randomicamente no interior da matriz.

Não desenvolvimentos nada relacionando ao MPU-6050 no Code Composer, a comunicação de I2C está sendo estudada para conseguirmos criar uma lógica que o *software* entenda.

VII. Resultados e discussões

Em relação a montagem do circuito no protoboard podemos observar abaixo:

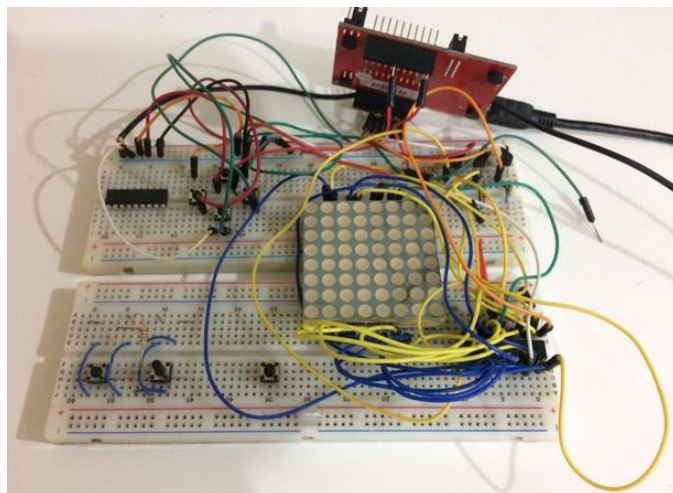


Figura 6 – Circuito montando

A maior parte que teve grande evolução foi a parte do código conseguimos integrar a MSP 430 com a criação da borda. No código precisamos definir algumas registradores no MAX 7219 para conseguir fazer a construção da borda toda acessa.

```
#define max7219_reg_digit0 0x01
#define max7219_reg_digit1 0x02
#define max7219_reg_digit2 0x03
#define max7219_reg_digit3 0x04
#define max7219_reg_digit4 0x05
#define max7219_reg_digit5 0x06
#define max7219_reg_digit6 0x07
#define max7219_reg_digit7 0x08
#define max7219_reg_decodeMode 0x09
#define max7219_reg_intensity 0x0a
#define max7219_reg_scanLimit 0x0b
#define max7219_reg_shutdown 0x0c
#define max7219_reg_displayTest 0x0f
```

Figura 7 –definição dos registradores no MAX 7219

Como necessitamos criar uma bolinha andando colocamos a função de *debounce* e uma função *delay* para jogar não reparar a troca de posição da bolinha.

O MSP 430 controla o MAX 7219, primeiramente ele varrer byte a byte dos registradores e seta uma máscara 0x01. Assim evitando algum lixo.

```
void scanByte(unsigned int sDATA) // Função para ler 1 byte de cada vez e enviar ao MAX7219.
{
    int i = 8;
    unsigned mask;
    while(i > 0) {
        mask = 0x01 << (i - 1); // fazendo a mascara para varrer "DATA".
        P1OUT &= ~CLK; // set clock to zero => XXXX X0XX
        if (sDATA & mask){ // verifica o bit a ser enviado para o Reg
            P1OUT |= DIN; // Envia 1 para o registrador através de DIN => XXX1 XXXX
        }else{
            P1OUT &= ~DIN; // Envia 0 ao DIN
        }
        P1OUT |= CLK; // Ao fim da leitura de DATA, setar clock para 1
        --i; // mover para próximo bit.
    }
} // end scanByte;
```

Figura 8 –varrendo byte a byte do MAX 7219

Após varrer cada registrador inicializamos o MAX 7219 que foi escrito no código abaixo:

```
void setReg(unsigned int reg, int col) // initialize all MAX7219's in the system
{
    int c = 0;
    P1OUT &= ~LOAD; // begin
    for ( c = 1; c <= maxInUse; c++) {
        scanByte(reg); // specify register
        scanByte(col); // (((data & 0x01) * 256) + data >> 1); // put data
    }
    P1OUT |= LOAD;
}
```

Figura 9 –Inicializando MAX 7219

Para o MSP saber onde está cada ponto acesso precisamos criar um mapa da matriz, sendo a função nomeada *changeMap* que ler ponto a ponto e salva os dados em um vetor chamado *limits*.

```
void changeMap()
{
    int i = 0;
    for(i = 0; i < 8; i++){
        if(i == portalIndex) // Se a coluna mover-se
        {
            if((portalIndex > 0) && (portalIndex < 7))
                limits[portalIndex] = portalX;
            else
                //if((portalY != 128) && (portalY != 1))
                limits[portalIndex] = ~portalY;
        }
        else
            limits[i] = limits0[i];
    }
}
```

Figura 10 –Função *changeMap*

Para construir a borda achamos o endereço nos registradores e deixamos sempre ligado assim para construir a borda ligada e como o *changeMap* identifica sempre a borda acessa e o único led que encontra-se apagado.

```
limits0[0] = 0b11111111;
limits0[1] = 0b10000001;
limits0[2] = 0b10000001;
limits0[3] = 0b10000001;
limits0[4] = 0b10000001;
limits0[5] = 0b10000001;
limits0[6] = 0b10000001;
limits0[7] = 0b11111111;
```

Figura 11 –Definindo borda

Criamos um laço infinito responsável para enviar as coordenada enviada pelos *push button* para conseguir fazer a movimentação da bolinha.

```
for(;;){

    // Laço responsável por enviar as coordenadas certas ao LED
    // Ele atualiza o valor enviado para o buffer[8] com as coordenadas atuais
    for(i = 0; i < 8; i++){
        if(i == coluna) // Se a coluna mover-se
            buffer[coluna] = limits[i] | linha; // Análise feita a partir da coluna, essa posição que receberá a nova posição
        // da linha ... isto é, será enviado o par (linha,coluna) para a matriz.
        else
            buffer[i] = limits[i]; // Nas demais colunas, a matriz será preenchida com as bordas do jogo.
    }
}
```

Figura 12 – Movimentação

Como é um jogo precisamos criar uma parte randômica para quando finalizar um jogo ele automaticamente define uma nova fase. Fizemos isso a partir de *timer A*.

```

#pragma vector=TIMER0_A1_VECTOR
__interrupt void Timer_A0(void){

    portalIndex++;
    if(portalIndex == 8)
    {
        portalIndex = 0;
    }

    portalY = (portalY << 1);
    if(portalY == 128)
        portalY = 0x02;
    else

    portalX ^= 0b10000001;
    TA0CTL &= ~TAIFG;
}

```

Figura 13 – Função randomicamente

Como ainda estamos estudando o protocolo de comunicação do MPU 6050 que é baseado em I2C. No qual o MSP é o mestre e acelerômetro é o escravo. Conseguimos trabalhar com ele no Energia, mas no Code Composer ainda não conseguimos trabalhar em cima.

VIII. Revisão Bibliográfica

- [1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)
- [2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
- [3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [4] K. Elissa, "Title of paper if known," unpublished.
- [5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.