

LED's Memory Game

Felipe Chermont
Engenharia de Eletrônica
Universidade de Brasília, FGA
Gama, Brasil
chernox27@gmail.com

Guilherme Simões
Engenharia de Eletrônica
Universidade de Brasília, FGA
Gama, Brasil
g.simoesdias@gmail.com

I. INTRODUÇÃO

A. Revisão Bibliográfica

[1] O Transtorno de Déficit de Atenção com Hiperatividade (TDAH) é um transtorno neurológico que aparece geralmente na infância e se não acompanhado e trabalhado pode atrapalhar o âmbito pessoal e profissional durante a vida adulta. [2] Segundo os dados da Organização Mundial de Saúde, cerca de 4% da população adulta tem o TDAH, equivalente a aproximadamente 2 milhões de brasileiros. Já na população jovem, o TDAH afeta 6% das crianças e destas apenas 69% concluem os estudos.

Segundo o site NeuroSaber [3] o jogo da memória está entre as 10 melhores brincadeiras simples para acalmar crianças hiperativas. Sendo bastante usada pelos professores e pais de crianças que sofrem com este transtorno, por estimular habilidades como o pensamento, a memorização a identificação de cores e sons estabelecendo conceitos de igualdade e diferença, entre outros.

Pensando nisso, decidimos desenvolver um jogo da memória, utilizando a msp430, LEDs, botões e um buzzer, onde a pessoa deverá ver uma sequência pseudoaleatória de LEDs, com diferentes sons e após um sinal, repetir a sequência que foi mostrada anteriormente.

O projeto será baseado nos trabalhos [5] “msp430launchpad-examples” para inicializar os leds e como fazer o *debouncing* dos botões e no trabalho [6] “Catch the LED” onde veremos como criar uma sequência pseudoaleatória de LED's

B. Justificativa

As capacidades cognitivas como memória e atenção são de extrema importância na vida pessoal de uma pessoa e o desenvolvimento incompleto dessas habilidades causa um

impacto enorme na vida pessoal de um adulto. [4] Como destacado pela Organização Brasileira de Déficit de Atenção, os sintomas na vida adulta podem ocasionar prejuízos no trabalho, nas relações amorosas, problemas na condução de veículos dentre outros.

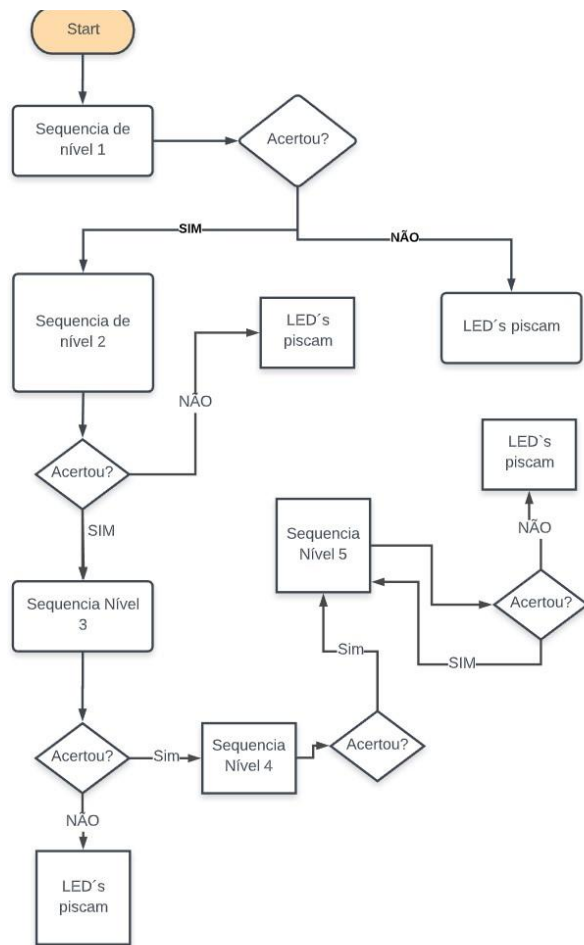
Este problema pode ser atenuado se algumas regiões do cérebro forem estimuladas desde a infância, para realizar este estímulo um método que vem se provando eficaz é a aplicação de jogos da memória. No entanto, para se manter a atenção de uma criança com TDAH é necessário fazer uso recursos chamativos como luzes e sons. Logo, decidimos fazer uso no jogo da memória de LEDs e buzzer.

Agora, para realizar a execução jogo foi necessário escolher um microcontrolador, dentre as opções estavam a MSP430 e a Armel AVR (Presente no Arduino) por conta do preço e acessibilidade, a escolha pelo MSP, além de ser a placa utilizada na matéria, oferece vários modos de LPM (Low Power Mode), contém uma arquitetura de 16 bits e a MSP 430 permite a mudança do clock, algo que o Arduino não permite.

C. Objetivos

O objetivo do projeto é, através da msp430, desenvolver um jogo da memória que ajude pessoas com TDAH no desenvolvimento da memória e da concentração, por meio da memorização e repetição das sequencias geradas, de forma divertida fazendo uso de estímulos visuais e sonoros.

Conforme o Diagrama a seguir:



D. Requisitos

O projeto deve atender aos requisitos:

1. acender os LEDs numa sequência pseudoaleatória ao se iniciar o jogo;
2. gerar sons diferentes através do buzzer; correspondentes a cada LED.
3. enviar um sinal de cada botão que corresponde ao LED para verificar se a pessoa acertou a sequência.
4. acionar outro nível, aumentando a sequência sempre que a pessoa acertar a sequência, limitado até 5 vezes.
5. Piscar todos os LEDs 3 vezes caso a pessoa perca o jogo e entrar em modo de espera até ser inicializado novamente.

Componentes usados:

- 4 LEDs Difusos (5mm) nas cores azul, verde, amarelo e vermelho;
- 5 Botões sem trava;
- 1 Protoboard;
- 1 Buzzer de 5V;
- 1 MSP430G

***Imagem de cada componente em anexo**

E. Benefícios

O projeto visa beneficiar pessoas com dificuldade de concentração devido ao transtorno. Proporcionando uma forma de treinar a concentração e à memória de forma divertida e intuitiva, além de desafiadora.

Link Para o GitHub e Trello:

GitHub Disponível em:

<<https://github.com/chermont04/Eletronica-Embarcada>>

Trello Disponível em:

<<https://trello.com/b/VZx9T9tQ/eletronicaembarcada>>



Figura 01: Led's



Figura 06: MSP 430G

V. Resultados

Na primeira parte do experimento, foram comprados e testado os materiais, sendo eles:

- Os quatro leds de 10mm aplicando uma voltagem de 3.3V, vista pelo datasheet deles;

- Os quatro botões de dois pinos, testados verificando a continuidade ao serem pressionados;
- O buzzer, similar ao teste dos Leds, foi aplicado uma voltagem de 5V;

Após o teste dos novos materiais, foi implementado o Código, conforme o Apêndice, no microcontrolador MSP430G, onde com isso foi possível começar a aplicação do projeto. Criando-se a sequência pseudoaleatória fazendo uso de amostragens de dois clocks diferentes que foram em seguida somados e em sequencia multiplicados por uma constante. Com a sequencia gerada, os leds piscam seguindo o vetor, sendo que a primeira repetição acende o led uma vez e conforme uma variável "round" é incrementada, simulando que o jogador está acertando a sequência, até o round alcançar 9 o numero de vezes que o led pisca aumenta de mesma equivalente.

REFERENCES

- [1] VINOCUR – Evelyn – TDAH:o que é, sintomas e tratamento – Disponível em: < <https://www.minhavidacom.br/saude/temas/tdah>>. Acesso em: 13 set. 2019.
- [2] JOSÉ - Fernando - Aumenta o número de pessoas com TDAH e o diagnóstico adequado é o maior desafio - 16 abril 2019 - Disponível em: <<http://www.osul.com.br/aumenta-o-numero-de-pessoas-com-tdah-eo-diagnostico-adequado-e-o-maior-desafio/>>. Acesso em: 13 set. 2019.
- [3] 10 brincadeiras simples para desacelerar as crianças com TDAH - 05 ago. 2018 - Disponível em: <<https://neurosaber.com.br/10-brincadeiras-simples-para-desacelerar-as-criancas-com-tdah/>>. Acesso em: 13 set. 2019.
- [4] GHIGIARELLI - Denise - O TDAH NO ADULTO E O PROCESSAMENTO DAS EMOÇÕES - 27 abr. 2016. Disponível em: <<https://www.normaseregras.com/normas-abnt/referencias/>>. Acesso em: 13 set. 2019.
- [5] Alf7 – MSP430 – Launchpad-Examples, Disponível em: <<https://github.com/alf7/MSP430-Launchpad-Examples>>
- [6] AdityaWadhwa – CatchTheLED, Disponível em: <<https://github.com/AdityaWadhwa/CatchTheLED>>

<Apêndice>

main.c

```
1#include <msp430G2553.h>
2#include <msp430.h>
3
4#define RED_LED    BIT1           // Red LED    (P1.1)  BIT1
5#define GREEN_LED  BIT2           // Green LED  (P1.2)  BIT2
6#define BLUE_LED   BIT4           // Blue LED   (P1.4)  BIT4
7#define YELLOW_LED BIT5           // Yellow LED  (P1.5)  BIT5
8
9#define TEN_MS    1
10#define QUART_SEC 25
11#define ONE_SEC   100
12#define BLINK     50
13#define PAUSE     50
14
15#define ENABLE_PINS 0xFFFE
16#define ACLK        0x0100
17#define SMCLK       0x0200
18#define CONTINUOUS  0x0020
19#define UP          0x0010
20
21void Reset(void); 22void
Wait(int t);
23
24int GetFirstNumber(void);
25int GetSecondNumber(void);
26void MakeSequence(int sequence[16], int first_number, int second_number); 27
28void BlinkLeds(int sequence[16], int round);
29
30void main(void)
31{
32    WDTCTL = WDTPW | WDTHOLD;
33
34    TA0CTL |= ACLK | UP;
35    TA0CCR0 = 400;
36
37    TA0CTL |= ACLK | CONTINUOUS;
38    TA1CTL |= SMCLK | CONTINUOUS;
39
40    P1DIR |= BLUE_LED | YELLOW_LED | RED_LED | GREEN_LED;
41    P1OUT &= (~RED_LED);
42    P1OUT &= (~BLUE_LED);
43    P1OUT &= (~GREEN_LED);
44    P1OUT &= (~YELLOW_LED);
45
46    while(1)
47    {
48        int first_number = 0;
49        first_number = GetFirstNumber();
50        Wait(QUART_SEC);
51        int second_number = 0;
52        second_number = GetSecondNumber();
53        int sequence[8] = {0.0};
54        MakeSequence(sequence, first_number, second_number);55
56        int game_state = 1;
57        int round = 0;
58        while(game_state == 1)
59        {
60            Wait(ONE_SEC);
```

```

61         BlinkLeds(sequence, round);
62         Wait(TEN_MS);
63         if (game_state == 1)
64         {
65             if (round == 9)
66             {
67                 round = 0;
68             }
69             else {round++;}
70         }
71         Wait(TEN_MS);
72     }
73 }
74
75 }
76
77 void Wait(int t)
78 {
79     int i = 0;
80     // While the count set has not been reached
81     while (i <= t)
82     { // When another 10 milliseconds have expired
83         if (TA0CTL & TAIFG)
84         {
85             // Increase the count and start another 10-millisecond round
86             i++;
87             TA0CTL &= (~TAIFG);
88         }
89     }
90 }
91
92 int GetFirstNumber(void)
93 {
94     int first_num = 0;
95     first_num = TA0R;
96     return first_num;
97 }
98
99 int GetSecondNumber(void)
100 {
101     int second_num = 0;
102     second_num = TA1R;
103     return second_num;
104 }
105
106
107 void MakeSequence(int sequence[8], int first_number, int second_number)
108 {
109     int i;
110     int first_array[16] = {0.0};
111     int second_array[16] = {0.0};
112     for (i = 0; i < 16; i++)
113     {
114         first_array[(15 - i)] = ((first_number >> i) & 0x01);
115         second_array[(15 - i)] = ((second_number >> i) & 0x01);
116     }
117     for (i = 0; i < 3; i++)
118     {
119         sequence[i] = (first_array[i]) + (second_array[i] * 4);
120     }
121     for (i = 3; i < 6; i++)

```

```

122     {
123         sequence[i] = (first_array[i]) + (second_array[i] * 1); 124     }
125     for (i = 6; i < 8; i++)
126     {
127         sequence[i] = (first_array[i]) + (second_array[i] * 3);
128     }
129 }
130
131 void BlinkLeds(int sequence[16], int round)
132 {
133     int i = 0;
134     do
135     {
136         switch (sequence[i])
137         {
138             case (0): P1OUT |= RED_LED;
139                     Wait(BLINK);
140                     P1OUT &= (~RED_LED);
141                     Wait(PAUSE);
142                     break;
143
144             case (1): P1OUT |= GREEN_LED;
145                     Wait(BLINK);
146                     P1OUT &= (~GREEN_LED);
147                     Wait(PAUSE);
148                     break;
149
150             case (3): P1OUT |= BLUE_LED;
151                     Wait(BLINK);
152                     P1OUT &= (~BLUE_LED);
153                     Wait(PAUSE);
154                     break;
155
156             case (4): P1OUT |= YELLOW_LED;
157                     Wait(BLINK);
158                     P1OUT &= (~YELLOW_LED);
159                     Wait(PAUSE);
160                     break;
161         }
162
163         i = i + 1;
164
165     }
166     while (i <= round);
167 }
168

```