Gameboy

Daniela Oliveira e Julia Borges
16/0117089 e 17/0014355

1. Justificativa

Através da MSP430 pretende-se realizar o controle do jogo, com uma programação simples que faz referência ao jogo tetris. Que poderá ser utilizado juntamente com os profissionais de terapia ocupacional, pois é uma forma para ajudar no processo de recuperação de reação cognitiva junto com entretenimento.

2. Objetivos

Através desse projeto pretende-se compreender como é possível fazer o controle de uma matriz de leds através da MSP430, aplicando os conhecimentos passados na sala de aula para a prática. Portanto, o intuito é projetar um programa que responda a alguns comando pré-determinados.
O jogo deverá ter modo de pausa, fazendo com que os blocos continuem da forma que estava anteriormente. Deverá conseguir rotacionar as peças, deslocar as peças na horizontal, e quando uma linha de leds na horizontal for totalmente acesa deverá ser eliminada, fazendo com que o restante desça mantendo o mesmo padrão.

3. Requisitos

O projeto será o controle de uma matriz 8x8 de LEDs utilizando a MSP430 para a criação de um Tetris, em que peças com 5 formatos pré-determinados irão aparecer de maneira aleatória nas primeiras linhas da matriz para que o jogador as movimente com o objetivo de criar linhas com peças para diminuir o tamanho da pilha.

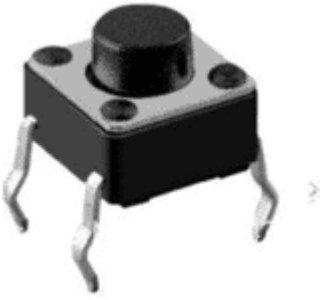
As peças poderão ser rotacionadas em 90º para a esquerda ou direita e serem movidas para a esquerda, direita ou para baixo. Além disso, o jogo pode ser iniciado, pausado ou resetado. Se a pilha tocar a primeira linha da matriz, a partida acaba e o jogo é resetado automaticamente.

Os materiais utilizados serão:

- MSP430F5529: microcontrolador para uso em aplicações de baixo consumo com 128KB de memória *flash* e 8KB de RAM e clock de 25Mhz. Sendo ideal, uma vez que se busca autonomia maior do sistema, que será alimentado por bateria, ao utilizar o modo de *low power*.

Figura 1 - MSP430F5529

- Botões: push buttons controle da peça, *start*, *reset* e pausa.

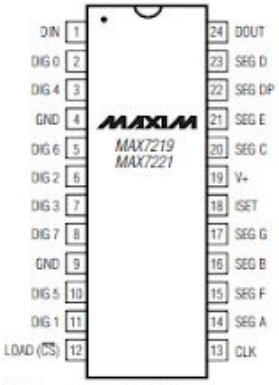


- Switch: para ligar e desligar a placa;

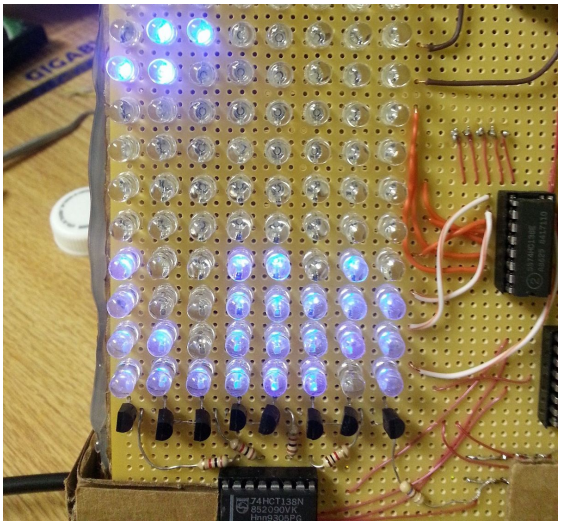
- LEDs vermelhos e placa de fenolite com furos: a matriz de LEDS será

fabricada para o projeto para melhorar a visibilidade dos LEDS e reduzir a quantidade de fios utilizando a placa.

- MAX7219: driver de display catodo comum serial input/output, utilizado para controle de display de 7 segmentos de até 8 dígitos,bar-graph display ou 64 LEDs individuais. O chip contém um decodificador BCD code-B, circuito de multiplexação, driver de dígito e segmento e uma RAM estática 8x8 que armazena cada dígito. Além disso, conta com modo low power com consumo de 150µA.



A figura abaixo exemplifica o jogo a ser construído, a partir de um projeto similar, que utiliza Arduino e controle de LEDS através de *shift register*.



4. Revisão bibliográfica

Outros projetos similares foram encontrados desenvolvidos em outros tipos de microcontroladores, o diferencial desse projeto proposto será o uso de interrupção para o modo pausa colocando em low power e também terá a opção de reset.

Um exemplo é o projeto de Arduino nano:*Arduino nano tetris game on homemade 16x8 matrix*[1], que não conta com pausa para o jogo, além de utilizador registradores de shift para a matriz ao invés do circuito integrado MAX7219.

5. Resultado

Projetos em arduino uno e nano já foram desenvolvidos e foi-se baseado neles o desenvolvimento dos códigos para funcionamento em MSP430. Para a primeira parte do projeto foi adquirido os componentes necessários para a produção do jogo. Para a primeira parte desse projeto adquiriu-se uma matriz de led pronta para ser testada e desenvolver o projeto. A matriz já possuía o MAX7219 para a multiplexação do display, facilitando o prosseguimento do projeto.

Dessa forma, para o teste da matriz e validação do projeto foi desenvolvido um código para formar as peças do jogo de forma aleatória. Acendendo apenas os leds correspondentes das peças base e usando o deslocamento por *shift* para o andar das peças.

Utilizou-se a ferramenta Energia IDE para a implementação no microcontrolador MSP30F5529 e validação do projeto. Nesta etapa não foi necessário o desenvolvimento de nenhuma biblioteca pois já existia uma biblioteca pronta para o controle dos leds na matriz.

6. Bibliografia

[1]Arduino Nano tetris game on homemade 16x8 matrix

https://www.hackster.io/mircemk/arduino-nano-tetris-game-on-homemade-16x8-matrix-dd6d60
[2]Arduino Tetris
https://github.com/marcosvalter/ArduinoTetris
[3] Arduino Based Bi-Color LED Matrix Tetris Game
https://www.instructables.com/id/Arduino-based-Bi-color-LED-Matrix-Tetris-Game/

7. Links

Trello:
https://trello.com/invite/b/LVK6CSho/591b732d2b6081f154f8c036ca0da700/projeto

Github:
https://github.com/JuliaBorgesSilva/projetofinal.git

8. Apêndice

Código desenvolvido para a formulação das peças do jogo:

```
#include <LedControl.h>

/**
* Led Matrix test
*/


#include <LedControl.h>

#define DATA_PIN     P1_3          // data for AS1106
#define CLOCK_PIN    P1_4          // clock for AS1106
#define LOAD_PIN     P1_5          // load CS for AS1106

// LedControl(int dataPin, int clkPin, int csPin, int numDevices=1);
LedControl          lc=LedControl(DATA_PIN, CLOCK_PIN, LOAD_PIN, 1);
unsigned long delaytime=100;


void setup() {

  pinMode(RED_LED,OUTPUT);
  pinMode(DATA_PIN,OUTPUT);
  pinMode(CLOCK_PIN,OUTPUT);
  pinMode(LOAD_PIN,OUTPUT);
  digitalWrite(LOAD_PIN,HIGH);
  digitalWrite(RED_LED, HIGH);
  delay(1000);
  digitalWrite(RED_LED, LOW);

  /*lc.init(); */
  lc.shutdown(0,false);
  lc.setIntensity(0,10);
  lc.clearDisplay(0);

}
byte piece[7][2] = {
  {B00111000,B00010000}, //t,
  {B00010000,B11110000}, //j
  {B00110000,B01100000}, //s
  {B01100000,B00110000}, //z
  {B01100000,B01100000}, //o
  {B11110000,B00010000}, //l
  {B00111100,B00000000} //i
  };
int blocktype;
void loop() {

  digitalWrite(RED_LED, HIGH);
  lc.clearDisplay(0);
   blocktype = random(7);

   for (int r=0; r<1; r++){
    for (int c=0; c<8; c++){
       lc.setColumn(0,c, piece[blocktype][0] );
       lc.setColumn(0,c+1,piece[blocktype][1]);
       if (c>0){
        lc.setColumn(0,c-1, B00000000);
       }

       else if(c==0){
       lc.setColumn(0,7, B00000000);
```

```
      }
      delay(250);

    }
   }
}
  /*digitalWrite(RED_LED, LOW);
  lc.clearDisplay(0);
  lc.setColumn(0, 1, B01010101);
      delay(1000);
  lc.setColumn(0, 3, B01010101);
      delay(1000);
  lc.setRow(0, 1, B01010101);
      delay(1000);
  lc.setRow(0, 3, B01010101);
      delay(1000);

  delay(10); */
```