

Ping Pong com Leds

Rafael Mendes Félix
Universidade de Brasília – Campus Gama
UnB
Gama, Brasil
rafaelmendesf@hotmail.com

Resumo — Este trabalho visa apresentar um projeto de um jogo do tipo ping pong que é microcontrolado por um MSP430.

Keywords — ping pong, msp

perfurada, e fazendo as ligações corretas, com o auxílio de solda, foi possível montar a matriz do projeto.

I. Introdução

Pong é o primeiro videogame lucrativo da história, dando origem a um novo setor da indústria. Não possuía gráficos espetaculares ou jogabilidade excelente, mas foi de importância fundamental na história do videogame. Foi criado por Nolan Bushnell e Ted Dabney na forma de um console ligado a um monitor, movido a moedas. *Pong* é um jogo eletrônico de esporte em duas dimensões que simula um tênis de mesa. O jogador controla uma paleta (barra vertical) no jogo movendo-a verticalmente no lado esquerdo da tela, e compete contra o computador ou outro jogador que controlam uma segunda raquete no lado oposto. Os jogadores usam suas paletas para acertar a esfera (bola) e mandá-la para o outro lado.

Deseja-se recriar este jogo, mas de uma forma mais lúdica, com leds tomando o lugar do monitor de fliperamas.

II. Justificativa

Foi escolhido recriar um clássico dos vídeos games, mas implementando de uma forma diferente. Em vez de usar monitores ou displays, foi decido implementar em uma matriz de leds 8x8 e acelerômetros como joystick.

III. Objetivos

Visamos implementar um sistema de mini game do tipo pong, em uma matriz de leds 8x8 e acelerômetros como controle.

IV. Hardware

O primeiro componente essencial para o projeto é uma matriz de leds. Ela servirá como se fosse a tela de um vídeo game, então é através dos leds que será visualizado as barras que se movimentam e a bola do jogo. Então foi confeccionado uma matriz própria.

As matrizes encontradas no mercado possuem dimensões pequenas, o que pode atrapalhar na jogabilidade do jogo, o deixando menos atrativo. Por isto, optou-se por fazer uma própria. Bastou comprar 64 leds e uma placa

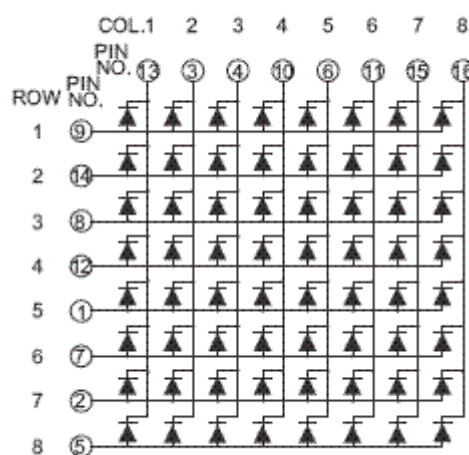


Figura1: Esquema de ligação dos leds



Figura2: Matriz de leds montada

Para controlar a matriz de leds, foi pensado em utilizar um shift register. Foi encontrado o CI 74595 que atende os requisitos necessários.

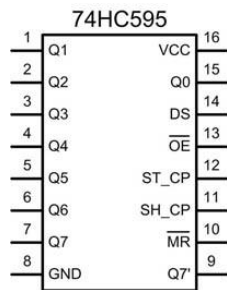


Figura3: CI 74595

Seu funcionamento é da seguinte maneira: ele recebe um sinal em série e após receber todo o sinal ele o envia de forma paralela. Suponha que temos um byte, equivalente a 8bits, por apenas um pino ele recebe esses 8 bits e após receber esses 8 bits ele os envia de forma paralela, os pinos responsáveis por enviar os bits de forma paralela são os pinos Q0 ao Q7, o pino que recebe o sinal em série é o pino 14, o DS.

Então, caso queiramos 8 leds, em vez de ligar cada led em um pino do MSP, o que faremos é usar esse CI. Ligaremos o DS no msp e manda os 8 sinais dos leds por ele de forma serial, e com cada led em um respectivo pino Qx (onde x varia de 0 a 7) do CI, quando acabar de mandar os 8 sinais dos LEDs pelo DS, o CI manda pela saída Q os 8 sinais diretos de forma paralela.

Podemos compreender o funcionamento do CI, bem como os seus pinos, a partir da seguinte tabela:

Latch	LOW	Prepara para a leitura dos bits na serial
Data	HIGH	Deixa data em nível alto (primeiro bit)
Clock	HIGH	Armazena o primeiro bit
Clock	LOW	Prepara para o armazenamento do segundo bit
Data	HIGH	Valor do segundo bit
Clock	HIGH	Armazena segundo bit
...
Data	LOW	Valor do oitavo bit
Clock	HIGH	Armazena o oitavo bit
Clock	LOW	Prepara clock em LOW
Latch	HIGH	Envia os oito bits na saída Q

Considerando que desejamos controlar uma matriz e que o shift register funciona bem controlando uma linha, utilizaremos dois shift register, um controlando as linhas e outro as colunas da matriz.

Para controlar as bases das barras, onde a bola quicará, foi pensado em utilizar acelerômetros.

O acelerômetro é um dispositivo que serve para medir a aceleração de um objeto em relação à gravidade. Acelerômetros são muito utilizados em celulares e videogames para executar algum tipo de ação dependendo da movimentação do aparelho.

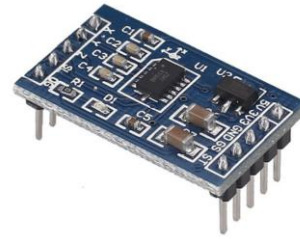


Figura4: Acelerômetro

A ideia é que de acordo com a movimentação do usuário, a barra se mova para a esquerda ou direita e com maior ou menor velocidade, dependendo da movimentação.

Por ser um módulo bastante comum e utilizado em projetos com micro controladores, foram encontradas referências de conexões para arduino que pode auxiliar no desenvolvimento do projeto.

Pino módulo	Função	Ligação ao Arduino
X	voltagem de saída eixo X	A0
Y	voltagem de saída eixo Y	A1
Z	voltagem de saída eixo Z	A2
3V3	Alimentação 3.3 Volts	Ligar ao pino 3.3 ou 5 volts do Arduino, dependendo da configuração desejada. Ligar ao 3.3 volts aumenta a precisão da leitura.
5 V	Alimentação 5 Volts	
ST	Self Test	Pino 12
GS	Seleção do modo de sensibilidade	Pino 10
OG	Deteção de queda livre linear	Pino 11
SL	Habilita o modo sleep	Pino 13
GND	Ground	GND

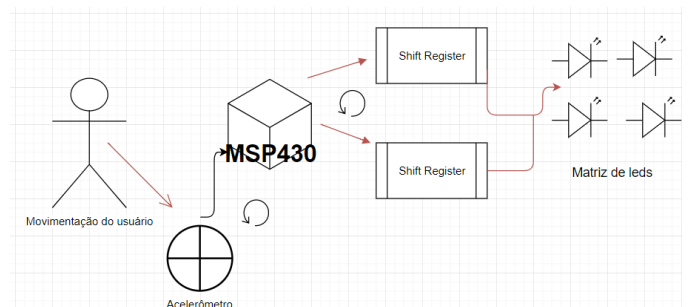


Figura5: esquemático básico do projeto

Conforme mostrado na figura a cima, devemos pensar em uma comunicação constante da movimentação do usuário com o msp e que como resposta comandará os registradores de deslocamento produzindo uma resposta visual, acendendo ou apagando os leds da matriz.

v. software

Para a utilização do acelerômetro, podemos utilizar como base o seguinte código:

```
#include <AcceleroMMA7361.h>
#include <LiquidCrystal.h>
```

```

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
AcceleroMMA7361 accelero;
int x;
int y;
int z;

void setup()
{
  lcd.begin(16, 2);
  Serial.begin(9600);
  accelero.begin(13, 12, 11, 10, A0, A1, A2);
  accelero.setAREfVoltage(3.3); //sets the AREF voltage to 3.3V
  accelero.setSensitivity(LOW); //sets the sensitivity to +/-6G
  accelero.calibrate();
  lcd.setCursor(0,0);
  lcd.print("X: ");
  lcd.setCursor(8,0);
  lcd.print("Y: ");
  lcd.setCursor(0,1);
  lcd.print("Z: ");
}

void loop()
{
  x = accelero.getXRaw();
  lcd.setCursor(3,0);
  lcd.print(x);
  y = accelero.getYRaw();
  lcd.setCursor(11,0);
  lcd.print(y);
  z = accelero.getZRaw();
  lcd.setCursor(3,1);
  lcd.print(z);
  Serial.print("nx: ");
  Serial.print(x);
  Serial.print("ty: ");
  Serial.print(y);
  Serial.print("tz: ");
  Serial.print(z);
  delay(500);
}

```

Este código lê as informações do acelerômetro e manda para um display LCD. Não é a intenção do projeto, mas a partir dele, percebemos que ele utiliza uma biblioteca própria. A `acceleromma7361`, caso utilizarmos o energia, podemos exportar essa biblioteca, o que facilitaria o projeto. Podemos perceber também que é necessário calibra o sensor no início do código. Percebemos também que é possível alterar a sensibilidade do sensor e para o projeto não necessitamos de uma sensibilidade muito alta.

Para controlar a matriz de leds, podemos observar o seguinte código:

```

int latch_c = 8;
int clock_c = 12;
int data_c = 11;

int latch_l = 2;
int clock_l = 3;
int data_l = 4;

```

```

byte c;
byte l;

void setup(){
  pinMode (latch_c,OUTPUT);
  pinMode (clock_c,OUTPUT);
  pinMode (data_c,OUTPUT);

  pinMode (latch_l,OUTPUT);
  pinMode (clock_l,OUTPUT);
  pinMode (data_l,OUTPUT);
}

void loop () {

  c = B00011000; //coluna ativado em 1
  l = B11100111; //linha ativado em 0

  mandar_c(c);
  mandar_l(l);
  escreve();
}

void mandar_c(byte c){
  digitalWrite(latch_c,LOW);
  for(int x=1;x<=128;x=x*2){
    digitalWrite(clock_c,LOW);
    if(c & x){
      digitalWrite(data_c,HIGH);
    }
    else{
      digitalWrite(data_c,LOW);
    }
    digitalWrite(clock_c,HIGH);
  }
}

void mandar_l(byte l){
  digitalWrite(latch_l,LOW);
  for(int x=1;x<=128;x=x*2){
    digitalWrite(clock_l,LOW);
    if(l & x){
      digitalWrite(data_l,HIGH);
    }
    else{
      digitalWrite(data_l,LOW);
    }
    digitalWrite(clock_l,HIGH);
  }
}

void escreve(){
  digitalWrite(latch_c,HIGH);
  digitalWrite(latch_l,HIGH);
  digitalWrite(latch_c,LOW);
  digitalWrite(latch_l,LOW);
}

```

```
}
```

Percebemos que este código opera uma matriz de leds através de dois shift register, conforme definido nas primeiras linhas. Observa-se que o intuito é acender os 4 leds centrais de uma matriz e que conforme está escrito, podemos identificar os seguintes comandos:

```
mandar_c(c); => manda o sinal da coluna para o CI e o armazena
mandar_l(l); => manda o sinal da linha para o CI e o armazena
escreve(); => manda o sinal da linha e da coluna para a matriz.
```

Ao iniciar a o jogo, foi pensado em escrever na matriz a palavra ping pong, de forma a ser uma tela de início do game.

```
void tela_de_inicio()
{
    int arr[8][8]=

    { {0x00,0x3c,0x24,0x24,0x3c,0x20,0x20,0x00}, //p
      {0x00,0x7c,0x10,0x10,0x10,0x10,0x7c,0x00}, //i
      {0x00,0x44,0x64,0x54,0x4c,0x44,0x44,0x00}, //n
      {0x00,0x3c,0x20,0x20,0x2c,0x24,0x3c,0x00}, //g
      {0x00,0x3c,0x24,0x24,0x24,0x24,0x3c,0x00} }; //o

    int seq[8]={0,1,2,3,0,4,2,3}; //PINGPONG

    int c;

    for(a=0;a<8;a++)
    {
        for(c=0;c<100;c++)
        {
            for(b=0;b<8;b++)
            {
                send(1<<b,arr[seq[a]][b]); //enviar ping pong
                delay(1);
            }
        }
    }
}
```

O game terá os seguintes botões de interrupção, que ainda devem ser refinados

```
P1REN |= BIT0 + BIT1 + BIT2 + BIT3; // P1.3 Enable
Pullup/Pulldown
P1OUT = BIT0 + BIT1 + BIT2 + BIT3; // P1.3
pullup
```

```
P1IE |= BIT0 + BIT1 + BIT2 + BIT3; // P1.3
interrupt enabled
P1IES |= BIT0 + BIT1 + BIT2 + BIT3; // P1.3 Hi/lo
falling edge
P1IFG &= ~(BIT0 + BIT1 + BIT2 + BIT3); // P1.3
IFG cleared just in case

P2REN |= BIT0 + BIT1 + BIT2; // P1.3 Enable
Pullup/Pulldown
P2OUT = BIT0 + BIT1 + BIT2; // P1.3 pullup
P2IE |= BIT0 + BIT1 + BIT2; // P1.3 interrupt
enabled
P2IES |= BIT0 + BIT1 + BIT2; // P1.3 Hi/lo falling
edge
P2IFG &= ~(BIT0 + BIT1 + BIT2); // P1.3 IFG
cleared just in case

_EINT();
```

VI. Conclusões

Apesar do projeto ter caminhado bem, com os primeiros testes na matriz funcionando, o comando de acelerômetro para fazer o controle de movimento de leds ainda deve ser aperfeiçoado. Mostrou-se relativamente fácil controlar alguns leds da matriz, porém quando pensado no conjunto inteiro, ainda há alguns problemas. Deve-se pensar em uma maneira de como direcionar a bola para que caso ela bata nas extremidades da barra ela não retorne perpendicularmente à barra, mas com algum ângulo. Mesmo com o pensamento básico do projeto ser simples, foram encontrados vários desafios para o cumprimento da proposta inicial.

VII. Bibliografias

<http://www.makeuseof.com/tag/how-to-recreate-the-classic-pong-game-using-arduino/>

<https://www.instructables.com/id/8-x-8-LED-Pong-with-Arduino/>

<https://circuits.io/circuits/2452966-fp-arduino-pong-8x8-led-matrix>

<http://mcuhq.com/28/msp430-3-axis-accelerometer-and-gyroscope-example-driver-using-the-lsm6ds0>