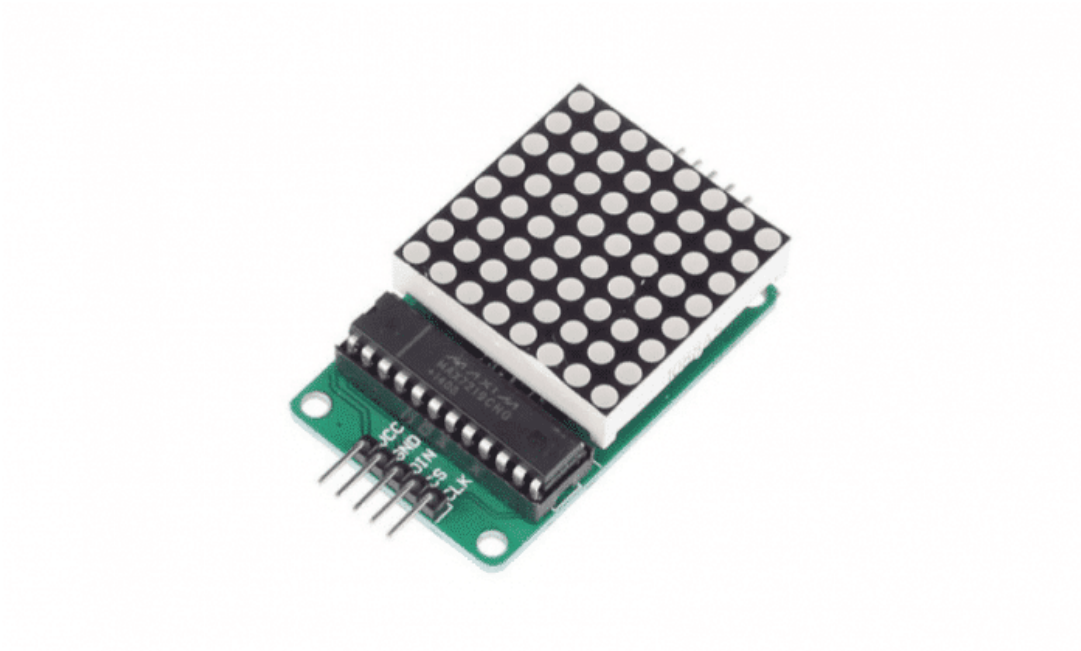
Módulo Matriz de LEDs com MAX7219

Por **Fábio Souza** - 27/04/2015



ÍNDICE DE CONTEÚDO

O uso de display de LEDs permite exibir diversas informações para o usuário. Com a combinação dos LEDs em forma de diplays de 7 segmentos, bargraph ou uma matriz de pontos é possível criar números, imagens ou até mesmos palavras. Henrique Puhlmannapresentou os displays de 7 segmentos e técnicas para multiplexação e acionamento usando microcontroladores.

Nesse artigo vamos abordar o uso de matriz de LEDs entendendo o seu funcionamento e utilizando o CI MAX7219 e um exemplo usando a placa Arduino UNO. Utilizaremos o módulo Matriz de LEDs 8X8 com MAX7219 , fornecido pela loja FILIPEFLOP .

Hardware para uso de Matriz de LEDs

Uma matriz de LEDs permite a criação de mensagens com maior quantidade de informação. Diferentemente dos display de 7 segmentos, é possível criar números, letras e desenhos de uma forma mais elegante. A figura 6 exibe uma matriz de LEDs 8X8:

Figura 1 - Matriz de LEDs 8X8

Nessa forma de montagem estão dispostos 64 LEDs para combinação e criação de imagens. Existem matrizes onde cada ponto é apenas um LED, ou seja, uma cor, e existem outras com 2 ou 3 LEDs permitindo a criação de imagens coloridas. Neste artigo vamos abordar apenas as matrizes de uma única cor.

Os LEDs na matriz são ligados de uma forma especial, para que se possa reduzir o número de pinos de acionamento, reduzindo assim o hardware de driver. Assim como os displays de 7 segmentos, as matrizes de LEDs também possuem montagens diferentes, que estão relacionadas à polaridade dos LEDs. A figura 2 exibe os dois tipos de montagens possíveis:

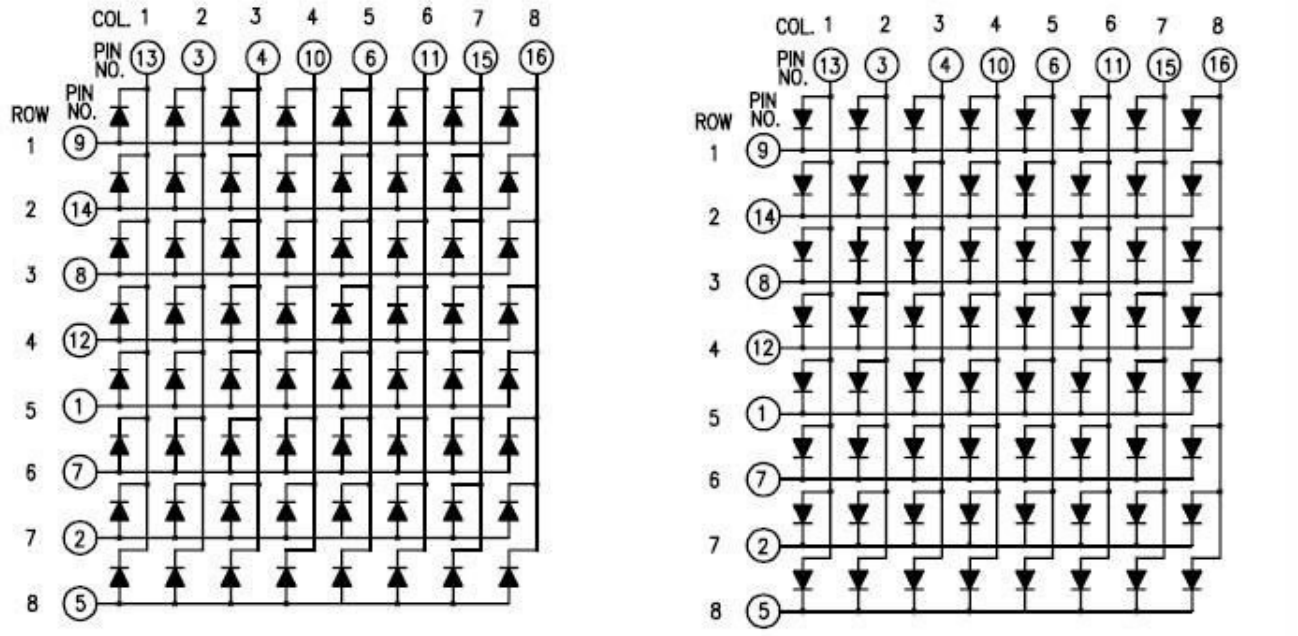


Figura 2 - Tipos de Montagens para Matriz 8X8

Note que a matriz está dividida em linhas e colunas, onde os LEDs são ligados em comuns. Essa forma de montagem reduz a quantidade de pinos para acionamento dos LEDs, 64 LEDs acionados por 16 pinos! Porém o acionamento requer técnica de multiplexação, para que se possa desenhar imagens utilizando todos os LEDs presentes na mesma. A multiplexação deverá acionar uma linha por vez e o LED desejado para cada coluna. Depois de um tempo essa linha deverá ser apagada e a próxima linha deverá ser acionada. Deve-se repetir esse ciclo continuamente para todas as linhas. A frequência de chaveamento entre linhas deverá ser feita acima de 50 Hz para que se aproveite do efeito de persistência de visão, onde uma imagem permanece por um tempo na retina. Dessa forma é possível exibir imagens usando todos os LEDs.

Para essa varredura pode-se utilizar um microcontrolador, porém serão necessários 16 pinos do mesmo para essa tarefa, e isso para apenas uma matriz. Para facilitar esse processo, pode-se utilizar circuitos digitais, como registradores de deslocamento, diminuindo a quantidade de pinos utilizados do microcontrolador.

Outra forma, mais interessante, é utilizar um CI dedicado para essa tarefa. O MAX7219 é dedicado para o acionamento de displays de LEDs. Com ele é possível acionar até 8 displays de 7 segmentos ou uma matriz de LEDs de 8X8. A entrada de dados é feita de forma serial, reduzindo a quantidade de pinos do microcontrolador. São utilizados apenas 3 pinos para interface com o microcontrolador, DIN, CLK e CS. O pino DIN é utilizado para inserir os bits (16 bits), que é sincronizado pelo pino CLK, o clock de sincronismo definido pelo microcontrolador. O pino CS é utilizado para transferir os 16 bits inseridos para o display. Os detalhes de funcionamento desse CI está disponível em seu datasheet .A figura 3 exibe a pinagem do MAX7219:

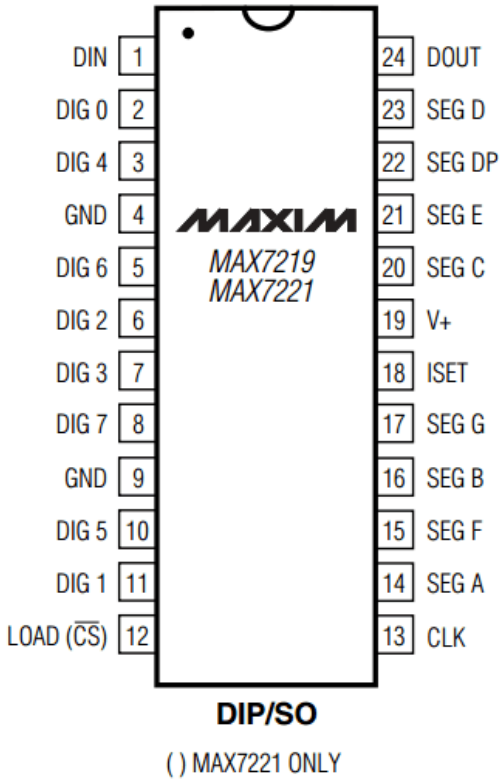


Figura 3 - Pinagem MAX7219

O diagrama de tempo para o MAX7219 é apresentado na figura 4:

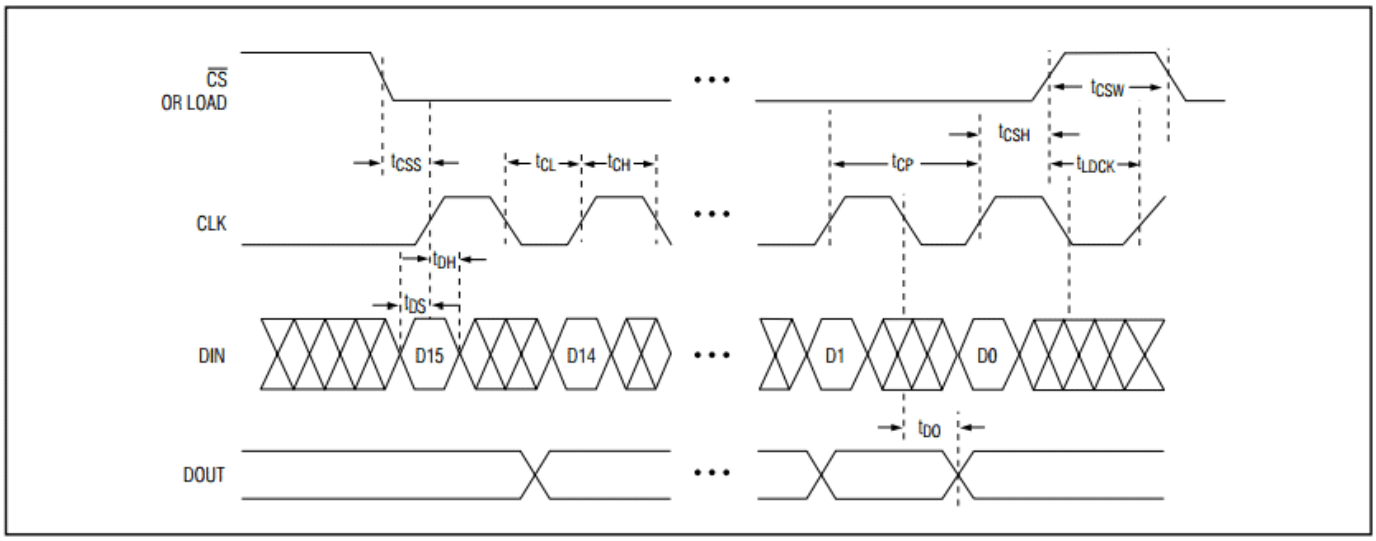


Figura 4 - Diagrama de tempo do MAX7219

A ligação do MAX7219 ao display necessita de atenção, caso seja feita em um protoboard, já que são necessárias muitas ligações. A figura 5 exibe o esquema de ligação do CI na Matriz de LEDs 8X8:

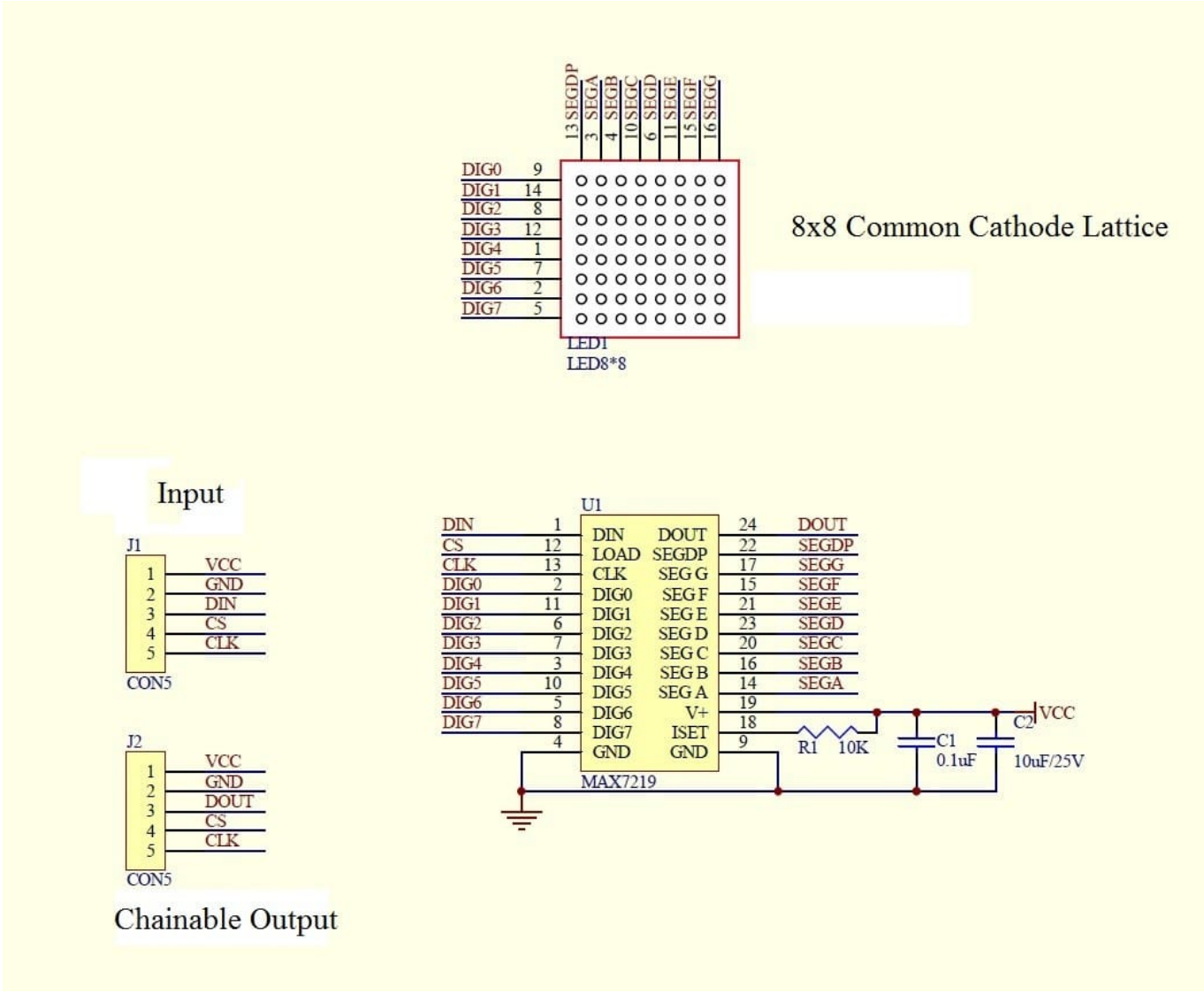


Figura 5 - Esquema de ligação

A figura 6 exibe um exemplo de ligação em uma protoboard. Note que são necessários muitos cabos:

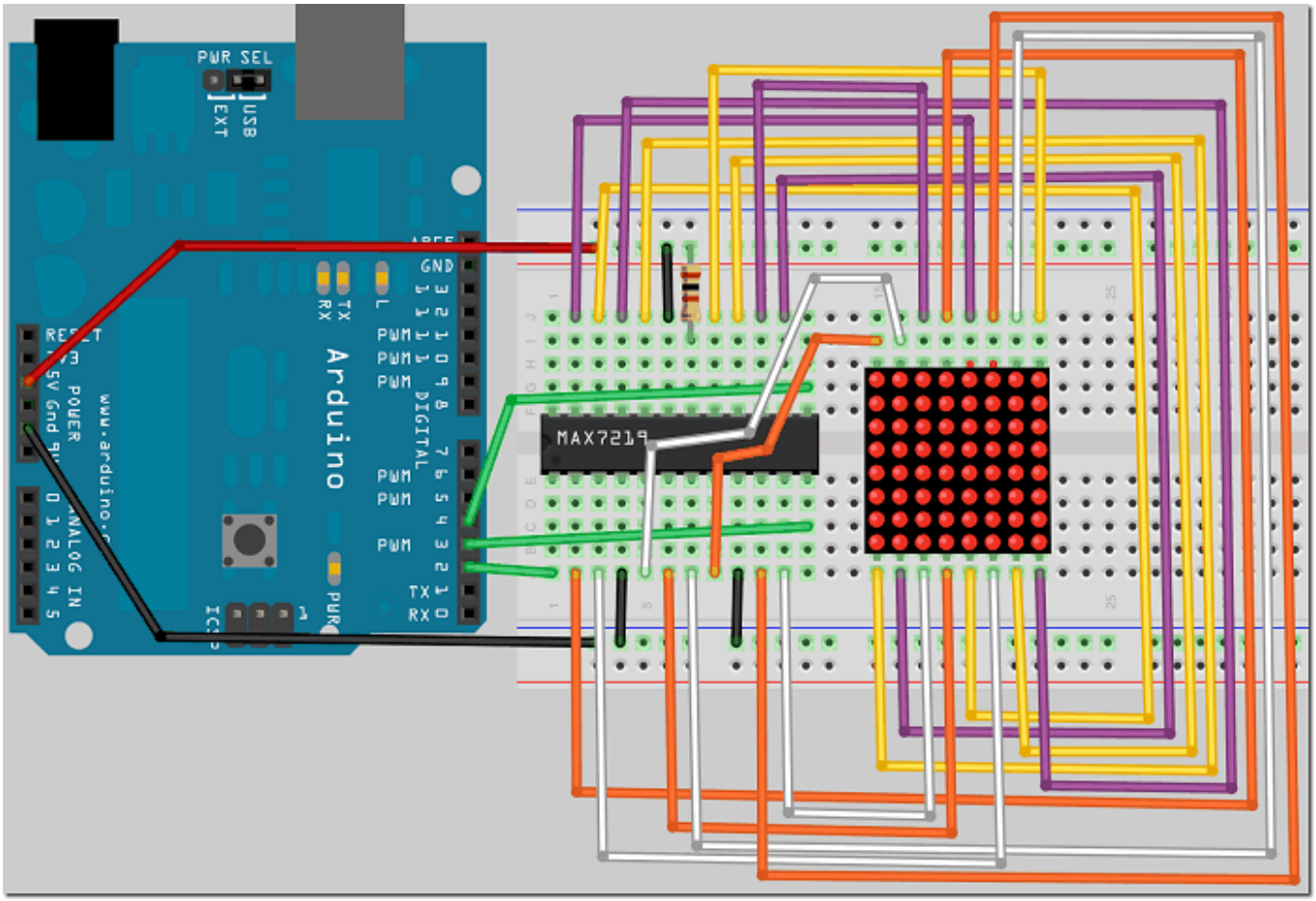


Figura 6 - Ligação em uma protoboard

Para facilitar a ligação do MAX7219 ao display, está disponível um módulo Matriz de LEDS 8X8 com MAX7219 na loja FILIPEFLOP. Esse módulo facilita a montagem de displays em Matriz de LEDs. A figura 7 exibe o módulo:

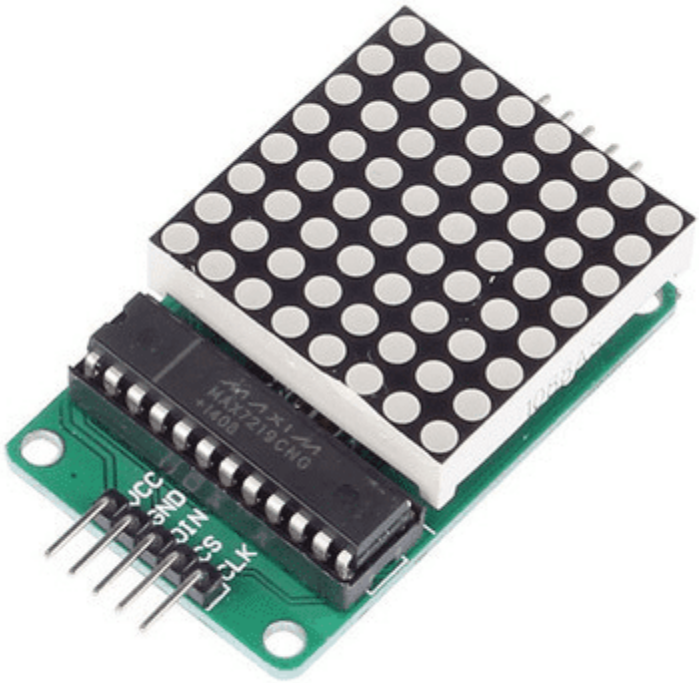


Figura 7 - Módulo Matriz de LED 8x8 com MAX7219

Para exemplificar o uso desse módulo, vamos criar uma aplicação utilizando a plataforma Arduino.

Exemplo de utilização de uma Matriz de LEDs

O exemplo a seguir tem como objetivo apresentar um jogo do tipo pong, utilizando a matriz de LEDs. Foi baseado no exemplo apresentado no livro Arduino Básico, da Novatec. Serão apresentados todos os detalhes da programação, para que você possa construir suas próprias aplicações utilizando a biblioteca LedControl.

Material necessário

Para esse exemplo você precisará dos seguintes componentes:

- Arduino UNO;
- Módulo Matriz de LEDs;
- Potenciômetro;
- Cabos.

Ligação do hardware

A ligação do Arduino ao módulo é bem simples, necessitando apenas de 5 cabos. A ligação deve ser feita conforme tabela 1 a seguir:

Arduino	Módulo
VCC	VCC
GND	GND
PINO 2	DIN
PINO 3	CS
PINO 4	CLK

Tabela 1: Ligação do Arduino ao módulo

O Potenciômetro deverá ser ligado à entrada analógica 5. Feitas as ligações necessárias, vamos passar para a programação.

Programação

Para o programa a seguir será necessário instalar a biblioteca LedControl. Após a instalação da mesma, compile o programa e faça o upload para a placa:

```
/*
   Exemplo PONG
   Livro Arduino Básico de Michael McRoberts
*/

//inclui biblioteca LedControl:http://playground.arduino.cc/Main/LedControl
#include "LedControl.h"

LedControl myMatrix = LedControl(12, 11, 10, 1); // cria uma instância de uma Matriz
                                                 //pino 12 - DIN
                                                 //pino 11 - CLK
                                                 //pino 10 - CS
int column = 1, row = random(8)+1;  // decide em que ponto a bola deve iniciar
int directionX = 1, directionY = 1; // certifica-se de que ela vai primeiro da esquerda para
int paddle1 = 5, paddle1Val;        // pino e valor do potenciômetro
int speed = 300;                    // velocidade
int counter = 0, mult = 10;         // variáveis auxiliares para controle de velocidade


void setup() {
  myMatrix.shutdown(0, false); // habilita o display
  myMatrix.setIntensity(0, 1); // define o brilho dos leds
  myMatrix.clearDisplay(0);    // limpa o display
  randomSeed(analogRead(0));   //
}


void loop() {

  paddle1Val = analogRead(paddle1);              //le valor do potenciometro
  paddle1Val = map(paddle1Val, 200, 1024, 1,6);  //mapeia valor para  entre 1 e 6

  column += directionX;      //atualiza valor da coluna
  row += directionY;         //atualiza valor da linha

  //verifica se boa atingiu a raquete
  if (column == 6 && directionX == 1 && (paddle1Val == row || paddle1Val+1 == row
|| paddle1Val+2 == row))
```

```
  {
    directionX = -1; //rebate bola
  }

  //verica se bola atingiu paredes: topo, base ou lateral esquerda
  if (column == 0 && directionX == -1 ) {directionX = 1;}
  if (row == 7 && directionY == 1 ) {directionY = -1;}
  if (row == 0 && directionY == -1 ) {directionY = 1;}

  //caso tenha atingido a coluna 7, significa que ultrapassou a raquete
  if (column == 7) { oops(); }  //chama a função oops, para finalizar o jogo


  myMatrix.clearDisplay(0); // limpa a tela para o próximo quadro de animação

  myMatrix.setLed(0, column, row, HIGH);            //desenha bola na linha e coluna atual

  //desenha raquete na coluna 7
  myMatrix.setLed(0, 7, paddle1Val, HIGH);
  myMatrix.setLed(0, 7, paddle1Val+1, HIGH);
  myMatrix.setLed(0, 7, paddle1Val+2, HIGH);

  if (!(counter % mult)) {speed -= 5; mult * mult;}
  delay(speed);
  counter++;
}

void oops()
{
  for (int x=0; x<3; x++)
  {
    myMatrix.clearDisplay(0);
    delay(250);
    for (int y=0; y<8; y++)
    {
      myMatrix.setRow(0, y, 255);
    }
    delay(250);
  }
  // reinicia todos os valores
  counter=0;
  speed=300;
  column=1;
  row = random(8)+1; // escolhe uma nova posição inicial
}
```

Inicialmente foi incluída ao projeto a biblioteca LedControl responsável pela manipulação do display através do MAX7912. Após a inclusão foi criada uma instância para a matriz:

```
#include "LedControl.h"

LedControl myMatrix = LedControl(12, 11, 10, 1); // cria uma instância de uma Matriz
                                                 //pino 12 - DIN
                                                 //pino 11 - CLK
                                                 //pino 10 - CS
```

Na sequência foram criadas as variáveis que serão utilizadas para controle do jogo. Inicialmente foram criadas as variáveis para controle de linha e coluna. O programa inicia na coluna 1 e a linha é iniciada aleatoriamente.

```
int column = 1, row = random(8)+1;    // decide em que ponto a bola deve iniciar
```

Depois foram criadas para controle de direção do deslocamento da bola É iniciado com valores para garantir o deslocamento da esquerda para a direita:

```
int directionX = 1, directionY = 1;  // certifica-se de que ela vai primeiro da esquerda para
```

Para leitura do potenciômetro, foi definido o pino e uma variável para leitura:

```
int paddle1 = 5, paddle1Val;         // pino e valor do potenciômetro
```

Por último foram criadas variáveis para o controle de velocidade de deslocamento. Foi iniciada a velocidade com 300 ms e um contador com zero e um multiplicador com 10:

```
int speed = 300;                     // velocidade
int counter = 0, mult = 10;          // variáveis auxiliares para controle de velocidade
```

Na função setup() foi habilitado o display, configurada a sua intensidade e também apagado o display. Por fim a função randomSeed() é inicializada com um valor aleatório, presente no pino A0:

```
void setup() {
  myMatrix.shutdown(0, false); // habilita o display
  myMatrix.setIntensity(0, 1); // define o brilho dos leds
  myMatrix.clearDisplay(0);    // limpa o display
  randomSeed(analogRead(0));   //
}
```

Na função loop() primeiramente é lido o valor do potenciômetro e mapeado para valores de 1 a 6.

```
paddle1Val = analogRead(paddle1);              //le valor do potenciometro
paddle1Val = map(paddle1Val, 200, 1024, 1,6);  //mapeia valor para  entre 1 e 6
```

Posteriormente é somado às variáveis linhas e colunas os valores presentes nas variáveis de direção:

```
column += directionX;      //atualiza valor da coluna
row += directionY;         //atualiza valor da linha
```

Próximo passo é verificar se a bola chegou à coluna 6 e atingiu a raquete. Caso tenha atingido a raquete é invertida a direção de deslocamento da bola:

```
if (column == 6 && directionX == 1 && (paddle1Val == row || paddle1Val+1 == row || paddle1Val+
{
  directionX = -1; //rebate bola
}
```

Na sequência é verificado se a bola atingiu as paredes, caso encontre algumas delas a direção é invertida:

```
if (column == 0 && directionX == -1 ) {directionX = 1;}
if (row == 7 && directionY == 1 ) {directionY = -1;}
if (row == 0 && directionY == -1 ) {directionY = 1;}
```

A ultima verificação é se a bola ultrapassou a raquete, caso chegue à coluna 7 o jogo é finalizado chamando a função oops():

```
if (column == 7) { oops(); }  //chama a função oops, para finalizar o jogo
```

Para finalizar o loop o display é atualizado. Primeiro ele é apagado, depois a bola é desenhada em sua nova posição e por ultimo é atualizada a posição da raquete:

```
myMatrix.clearDisplay(0); // limpa a tela para o próximo quadro de animação

myMatrix.setLed(0, column, row, HIGH);            //desenha bola na linha e coluna atual

//desenha raquete na coluna 7
myMatrix.setLed(0, 7, paddle1Val, HIGH);
myMatrix.setLed(0, 7, paddle1Val+1, HIGH);
myMatrix.setLed(0, 7, paddle1Val+2, HIGH);
```

Por último é atualizada a velocidade de deslocamento da bola:

```
if (!(counter % mult)) {speed -= 5; mult * mult;}
delay(speed);
counter++;
```

A função oops() apaga o display e depois acende todos os pontos, piscando 3 vezes indicando o fim do jogo. Tambéms são reiniciadas as variáveis e definida uma nova posição inicial:

```
void oops()
{
 for (int x=0; x<3; x++)
 {
   myMatrix.clearDisplay(0);
   delay(250);
   for (int y=0; y<8; y++)
   {
     myMatrix.setRow(0, y, 255);
   }
   delay(250);
 }

  // reinicia todos os valores
  counter=0;
  speed=300;
  column=1;
  row = random(8)+1; // escolhe uma nova posição inicial
}
```

Essa foi uma aplicação possível para um display Matriz de LEDs 8X8, porém existem diversas aplicações que podem ser feitas utilizando esse tipo de display. O uso do MAX7219 facilita a manipulação e desenvolvimento do hardware para criação de displays mais complexos. A biblioteca LedControl auxilia na programação de aplicações utilizando o MAX7219, vale a pena estudar a sua implementação e testar outras aplicações utilizando essa biblioteca. Confira o projeto Parola que apresenta um display de mensagem rolante bem interessante.

Referências

HENRIQUE PUHLMANN

Displays de LED de 7 segmentos

Sistemas Operacionais de Tempo Real - Displays de 7 segmentos

MAX7219

Arduino UNO

Módulo Matriz de LED 8x8 com MAX7219

Loja FILIPEFLOP

Livro Arduino Básico

Biblioteca LedControl

http://artofcircuits.com/product/max7219-8x8-led-matrix-module-un-assembled-kit

http://parola.codeplex.com/