8x8 LED Matrix Display Alphabets utilizing

Column scanning method

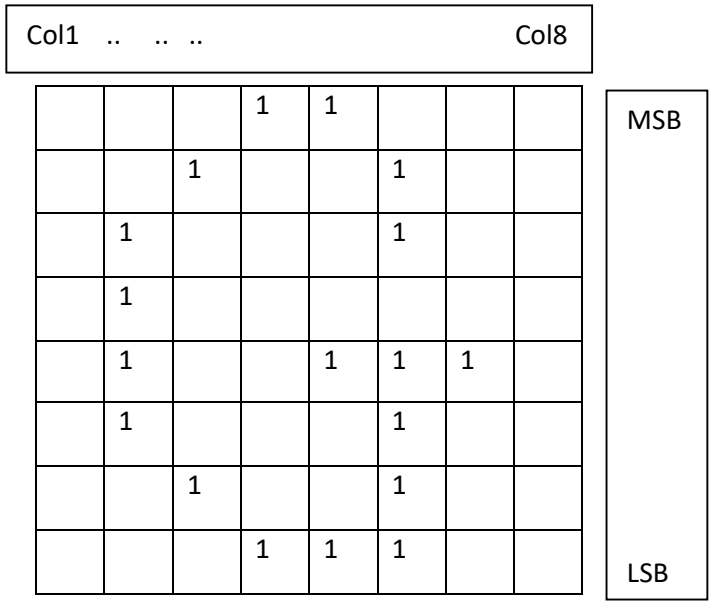
Originally written for Freenove Projects Kit with Arduino UNO
-Can be applied to any DIY kit using similar display method and circuitry that uses 74HC595 shift register ICs;

Details of the project can be found at:

https://youtu.be/h_GWgqs7esk

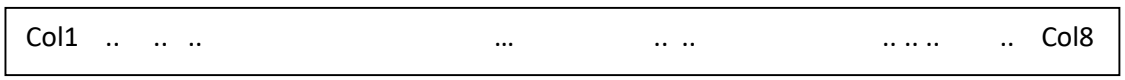
https://www.youtube.com/watch?v=NTRlhVGyMZo

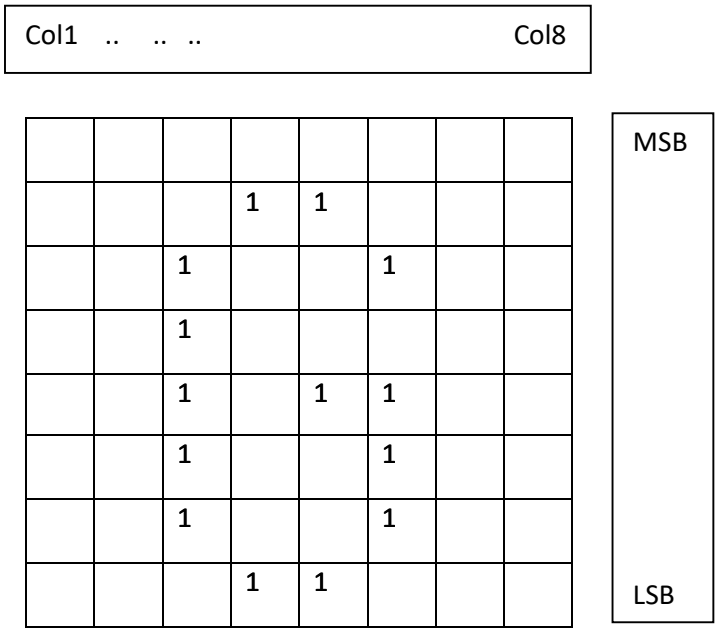
Github: https://github.com/sattar-rmit/Freenove-LED-matrix-scrolling-message



			1	1			
		1			1		
	1				1		
	1						
	1			1	1	1	
	1				1		
		1			1		
			1	1	1		

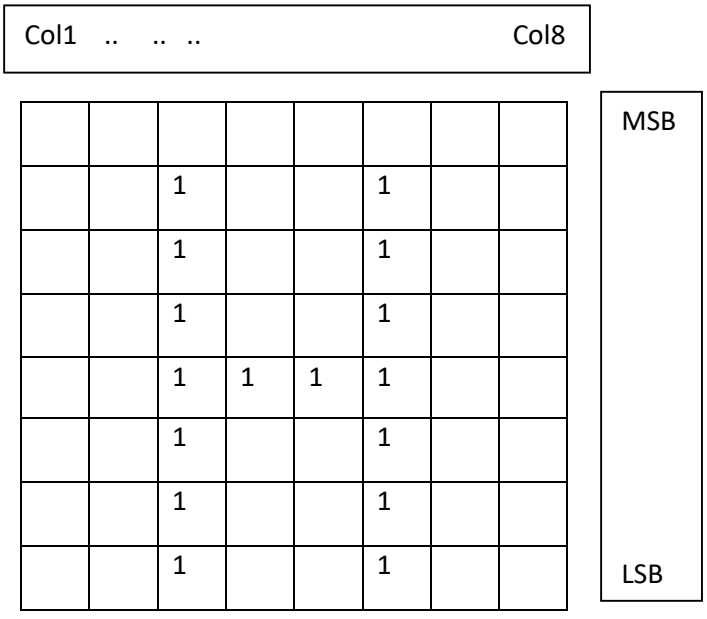
G = B00000000, B00111100, 01000010, 10000001, 10001001, 01101111, 00001000, 00000000





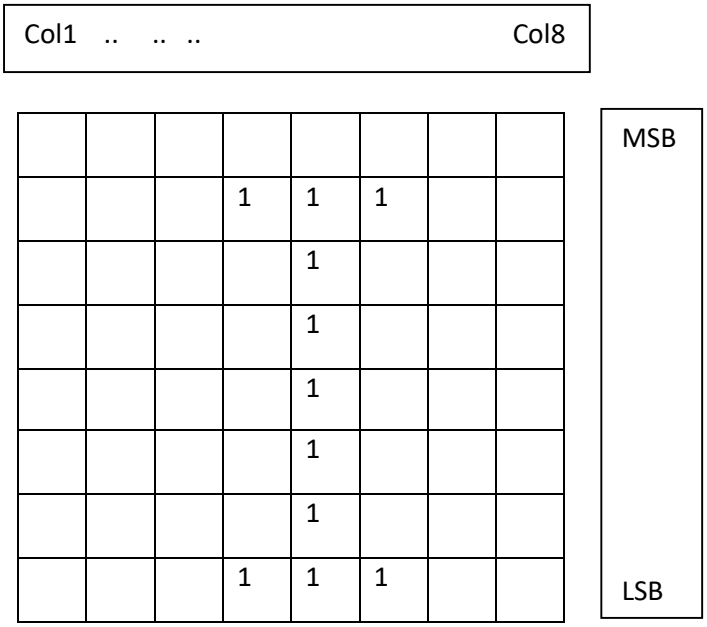
			1	1			
		1			1		
		1					
		1		1	1		
		1			1		
		1			1		
			1	1			

G = B00000000, B0000 0000, B00111110, B01000001, B01001001, B00101110, B00000000, B00000000

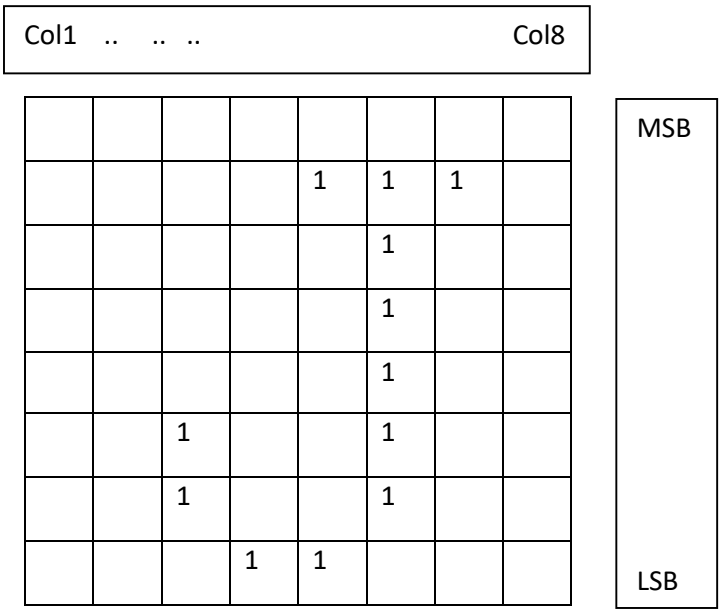


		1			1		
		1			1		
		1			1		
		1	1	1	1		
		1			1		
		1			1		
		1			1		

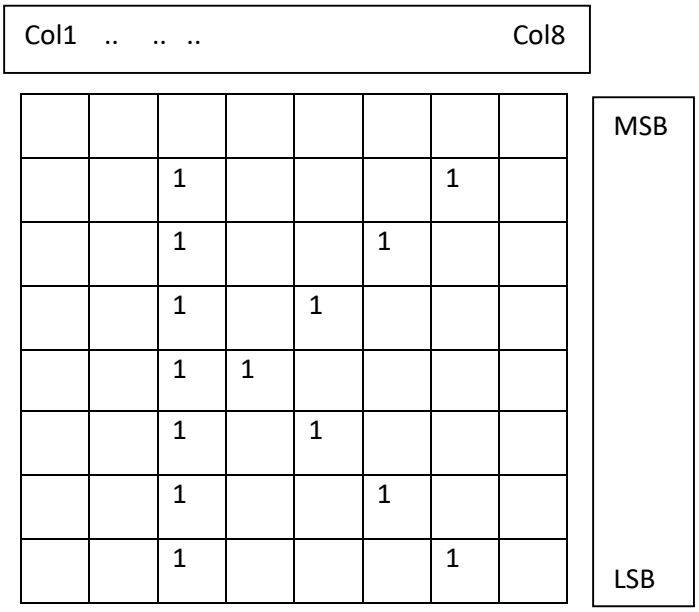
H = B00000000, B00000000, B01111111, B00001000, B00001000, B01111111, B00000000, B00000000



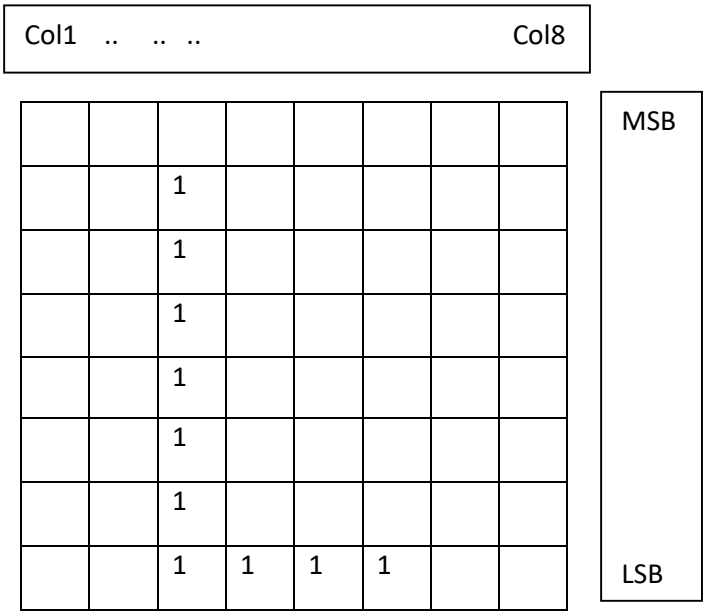
I = B00000000, B00000000, B00000000, B01000001, B01111111, B01000001, B00000000, B00000000



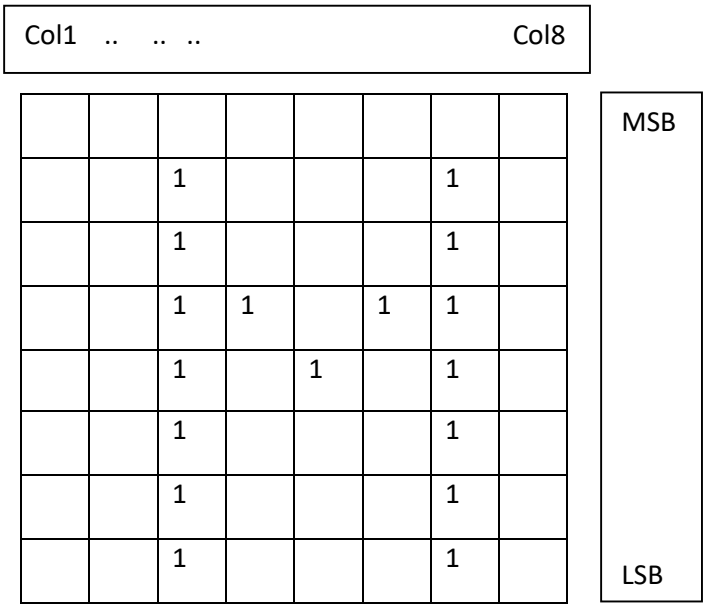
J = B00000000, B00000000, B00000110, B00000001, B01000001, B01111110, B01000000, B00000000



K= B00000000, B00000000, B01111111, B00001000, B00010100, B00100010, B01000001, B00000000

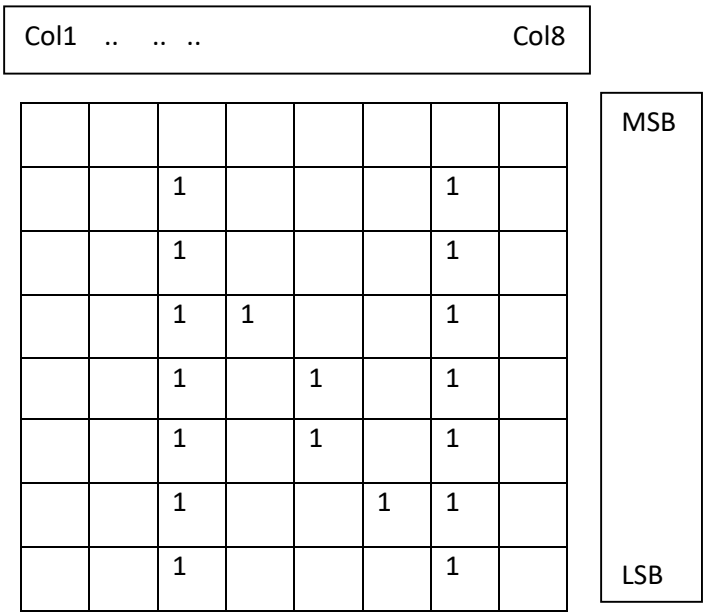


L = B00000000, B00000000, B01111111, B00000001, B00000001, B00000001, B00000000, B00000000



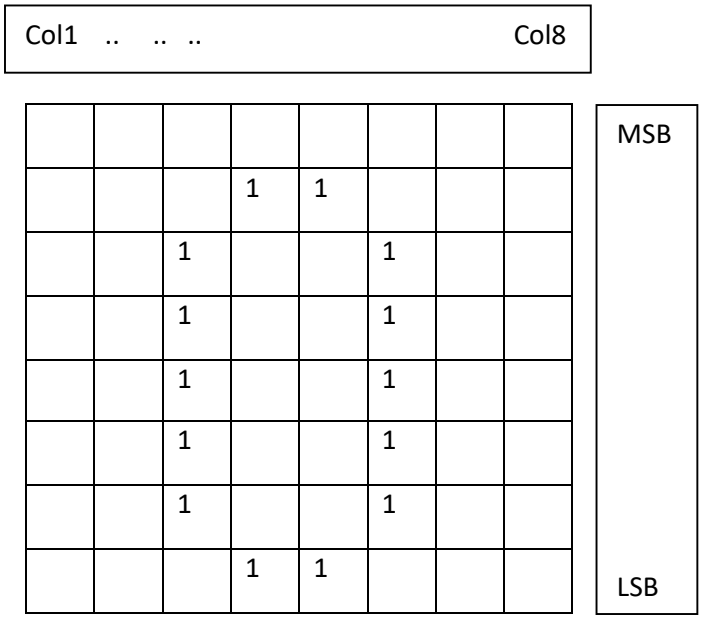
		1				1	
		1				1	
		1	1		1	1	
		1		1		1	
		1				1	
		1				1	
		1				1	

M = B00000000, B00000000, B01111111, B00010000, B00001000, B00010000, B01111111, B00000000



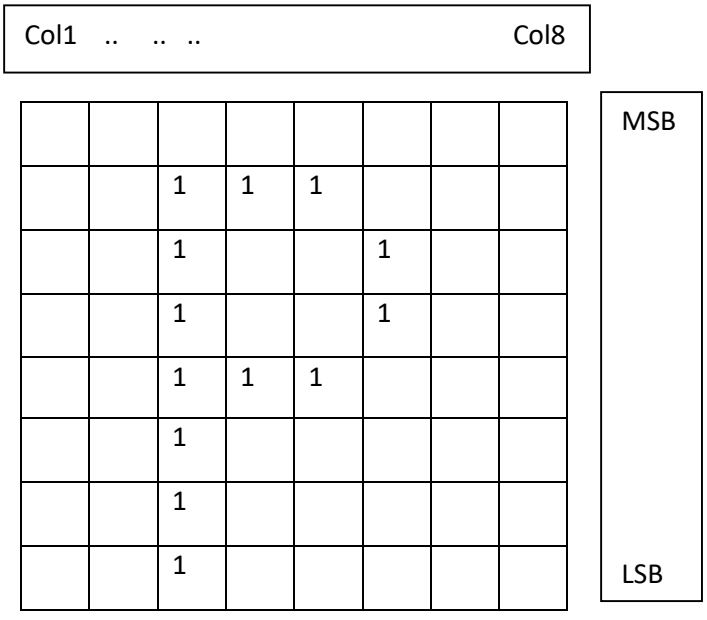
		1				1	
		1				1	
		1	1			1	
		1		1		1	
		1		1		1	
		1			1	1	
		1				1	

N = B00000000, B00000000, B01111111, B00010000, B00001100, B00000010, B01111111, B00000000



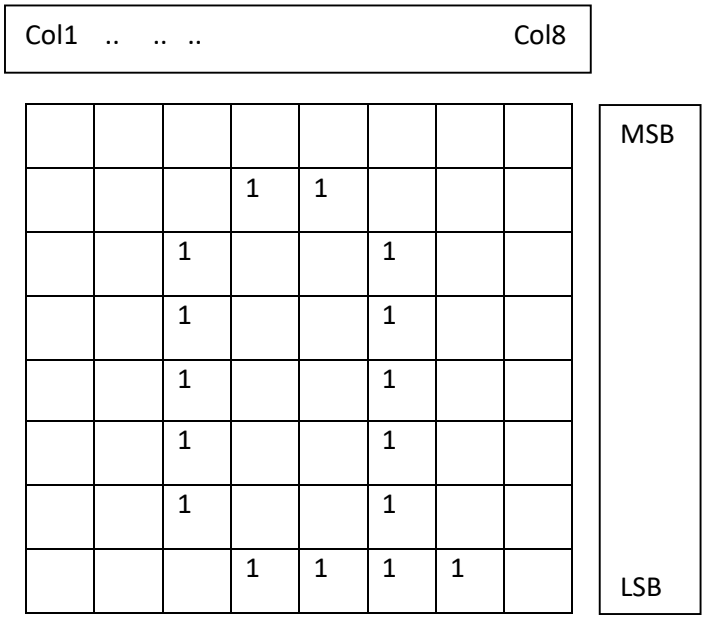
			1	1			
		1			1		
		1			1		
		1			1		
		1			1		
		1			1		
			1	1			

O = B00000000, B00000000, B00111110, B01000001, B01000001, B00111110, B00000000, B00000000



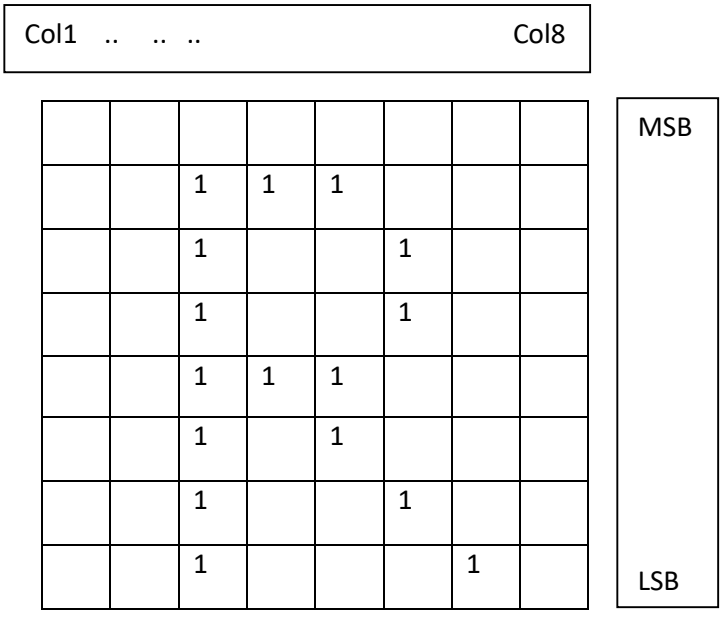
		1	1	1			
		1			1		
		1			1		
		1	1	1			
		1					
		1					
		1					

P = B00000000, B00000000, B01111111, B01001000, B01001000, B00110000, B00000000, B00000000



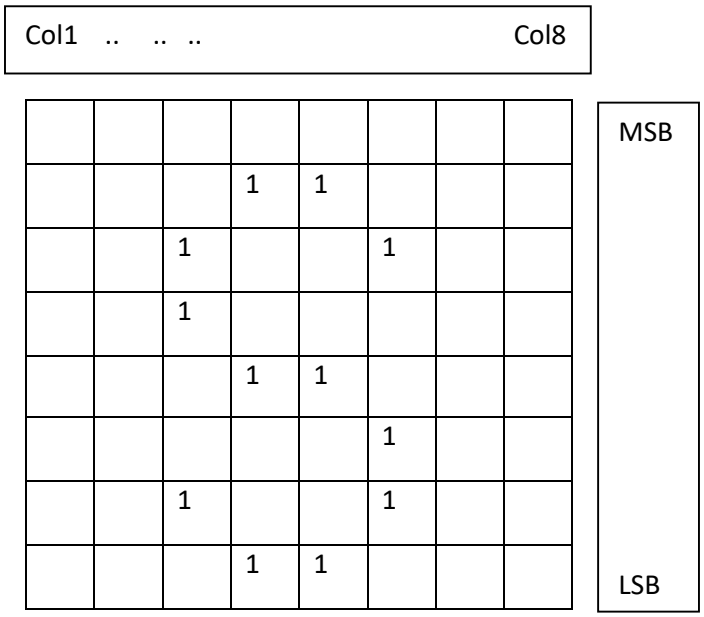
			1	1			
		1			1		
		1			1		
		1			1		
		1			1		
		1			1		
			1	1	1	1	

Q = B00000000, B00000000, B00111110, B01000001, B01000001, B00111111, B00000001, B00000000



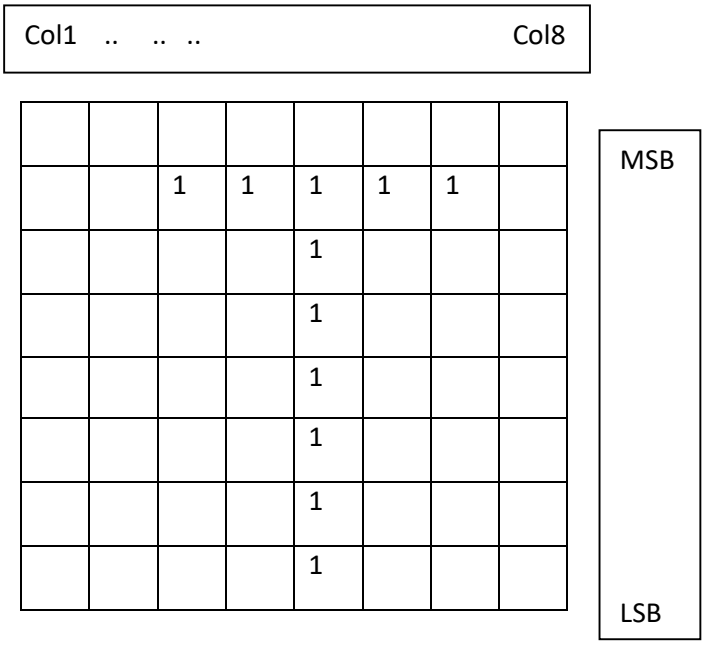
		1	1	1			
		1			1		
		1			1		
		1	1	1			
		1		1			
		1			1		
		1				1	

R = B00000000, B00000000, B01111111, B01001000, B01001100, B00110010, B00000001, B00000000



			1	1			
		1			1		
		1					
			1	1			
					1		
		1			1		
			1	1			

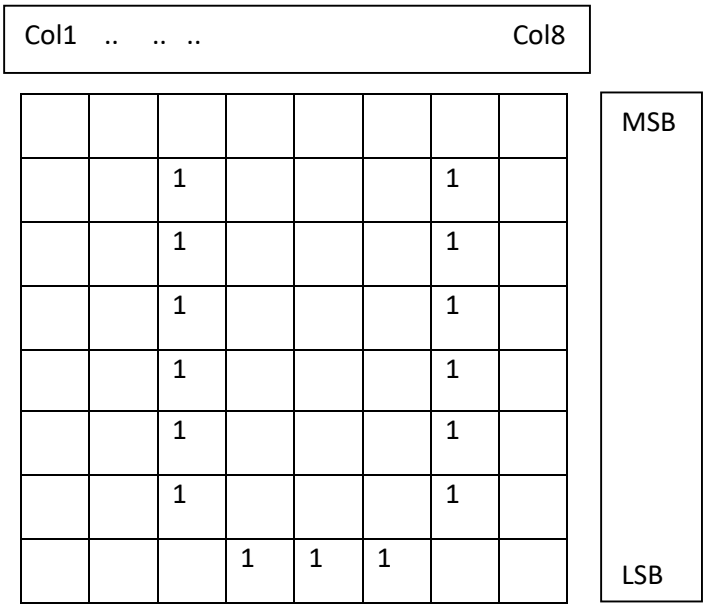
S = B00000000, B00000000, B00110010, B01001001, B01001001, B00100110, B00000000, B00000000



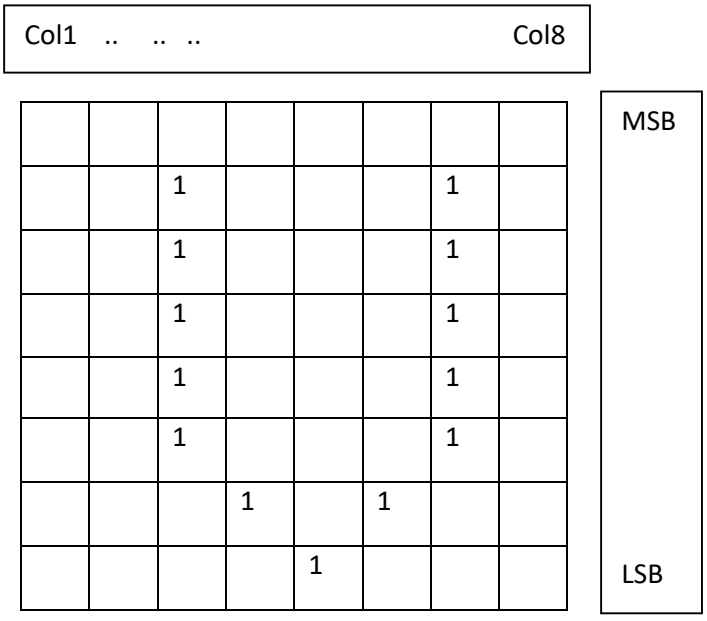
		1	1	1	1	1	
				1			
				1			
				1			
				1			
				1			
				1			

T = B00000000, B00000000, B01000000, B01000000, B01111111, B01000000, B01000000, B00000000

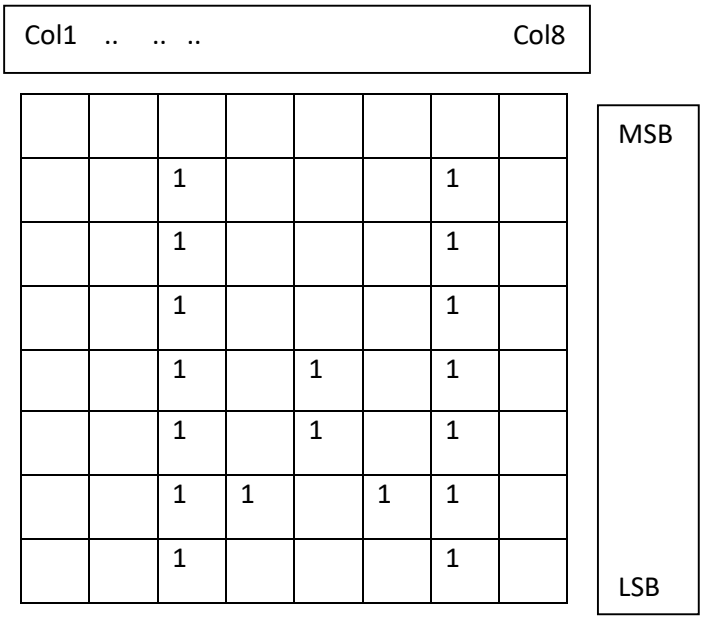
.



U = B00000000, B00000000, B01111110, B00000001, B00000001, B00000001, B01111110, B00000000

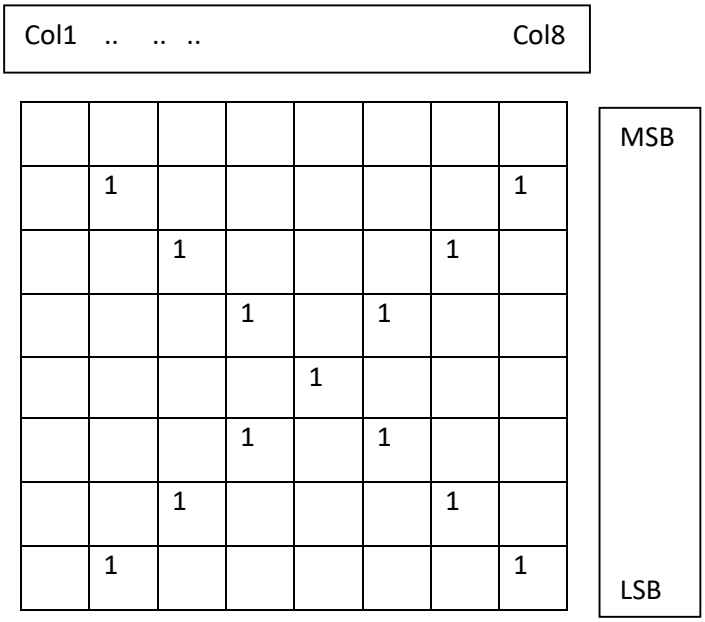


V = B00000000, B00000000, B01111100, B00000010, B00000001, B00000010, B01111100, B00000000



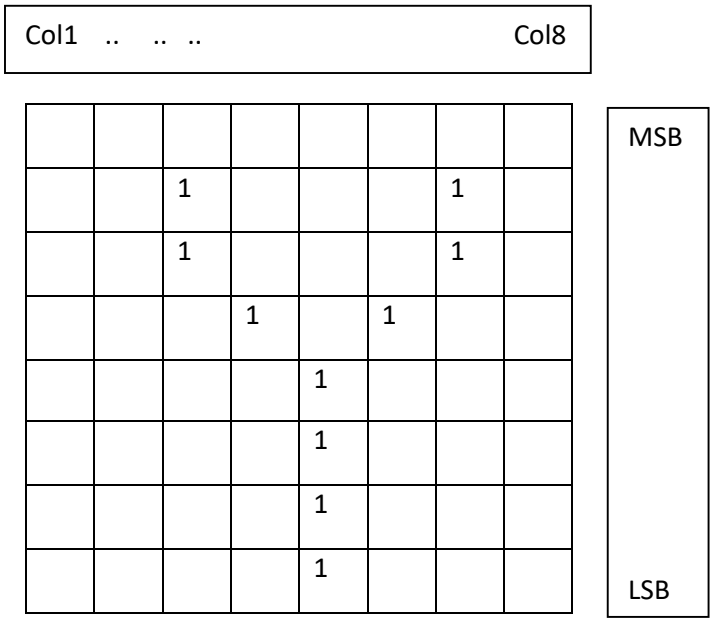
		1				1	
		1				1	
		1				1	
		1		1		1	
		1		1		1	
		1	1		1	1	
		1				1	

W = B00000000, B00000000, B01111111, B00000010, B00001100, B00000010, B01111111, B00000000



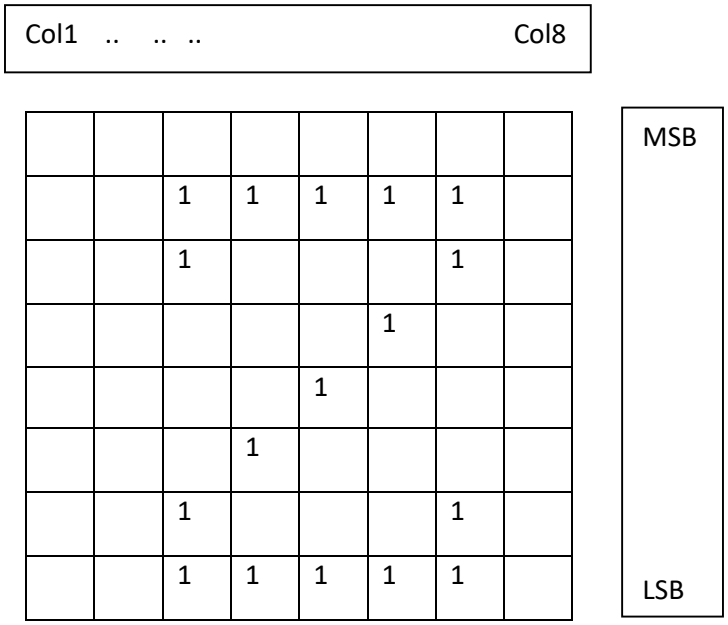
	1						1
		1				1	
			1		1		
				1			
			1		1		
		1				1	
	1						1

X = B00000000, B01000001, B00100010, B00010100, B00001000, B00010100, B00100010, B01000001



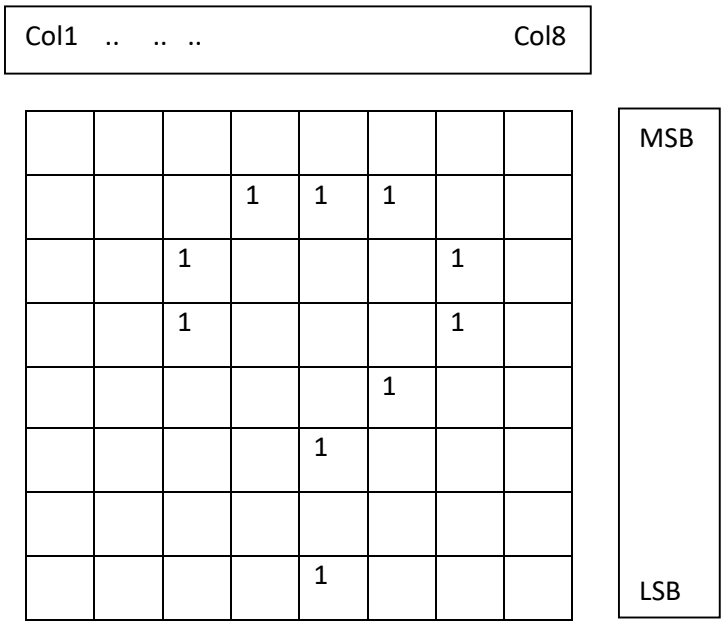
		1				1	
		1				1	
			1		1		
				1			
				1			
				1			
				1			

Y= B00000000, B00000000, B01100000, B00010000, B00001111, B00010000, B01100000, B00000000

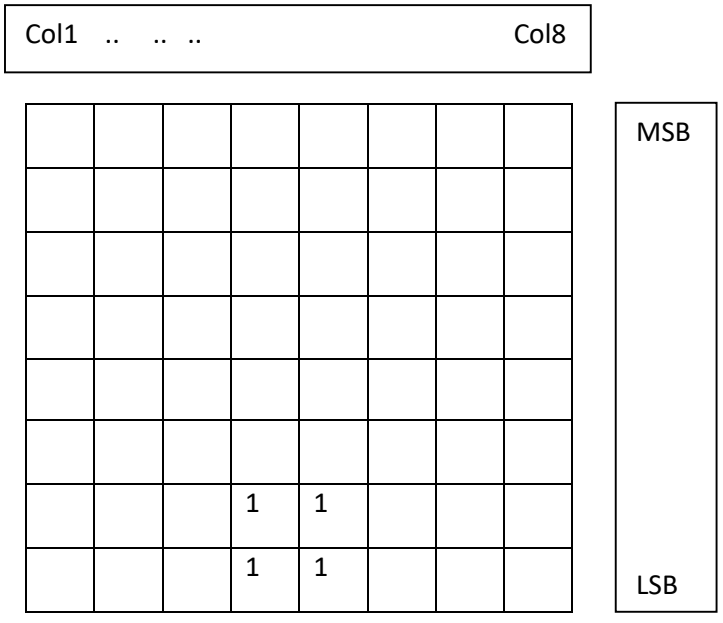


		1	1	1	1	1	
		1				1	
					1		
				1			
			1				
		1				1	
		1	1	1	1	1	

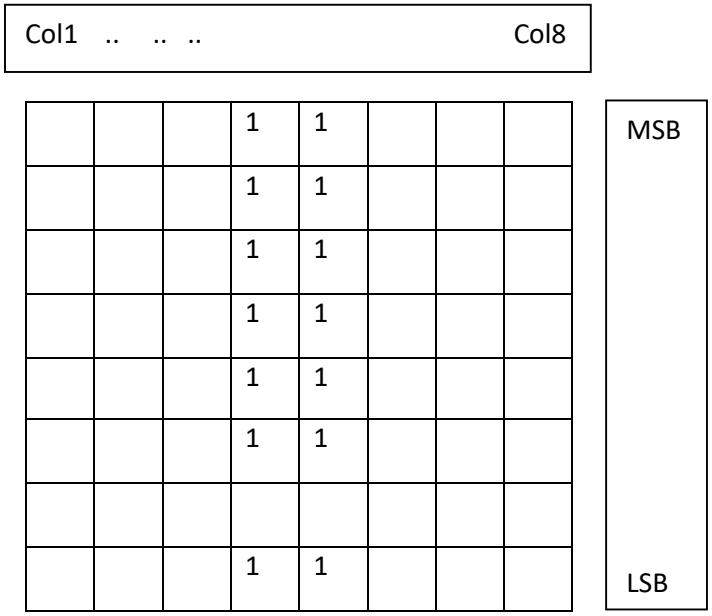
Z= B00000000, B00000000, B01100011, B01000101, B01001001, B01010001, B01100011, B00000000



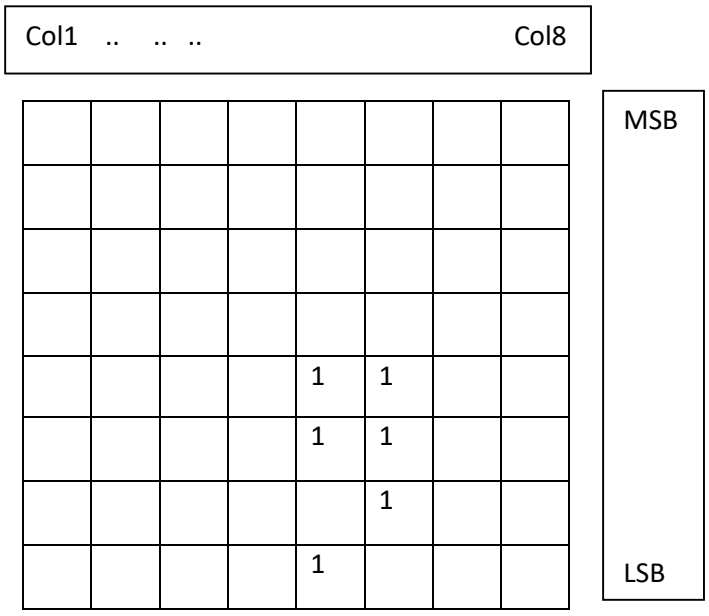
?= B00000000, B00000000, B00110000, B01000000, B01000101, B01001000, B00110000, B00000000



. = B00000000, B00000000, B00000000, B00000011, B00000011, B00000000, B00000000, B00000000



!= B00000000, B00000000, B00000000, B01111101, B01111101, B00000000, B00000000, B00000000



,= B00000000, B00000000, B00000000, B00000000, B00001101, B00001110, B00000000, B00000000

All characters:

```
// Define the data of numbers and letters, and save them in flash area
const int data[] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // " "
  0x00, 0x00, 0x21, 0x7F, 0x01, 0x00, 0x00, 0x00, // "1"
  0x00, 0x00, 0x23, 0x45, 0x49, 0x31, 0x00, 0x00, // "2"
  0x00, 0x00, 0x22, 0x49, 0x49, 0x36, 0x00, 0x00, // "3"
  0x00, 0x00, 0x0E, 0x32, 0x7F, 0x02, 0x00, 0x00, // "4"
  0x00, 0x00, 0x79, 0x49, 0x49, 0x46, 0x00, 0x00, // "5"
  0x00, 0x00, 0x3E, 0x49, 0x49, 0x26, 0x00, 0x00, // "6"
  0x00, 0x00, 0x60, 0x47, 0x48, 0x70, 0x00, 0x00, // "7"
  0x00, 0x00, 0x36, 0x49, 0x49, 0x36, 0x00, 0x00, // "8"
  0x00, 0x00, 0x32, 0x49, 0x49, 0x3E, 0x00, 0x00, // "9"
  0x00, 0x00, 0x3E, 0x41, 0x41, 0x3E, 0x00, 0x00, // "0"
  0x00, 0x00, 0x3F, 0x44, 0x44, 0x3F, 0x00, 0x00, // "A"
  0x00, 0x00, 0x7F, 0x49, 0x49, 0x36, 0x00, 0x00, // "B"
  0x00, 0x00, 0x3E, 0x41, 0x41, 0x22, 0x00, 0x00, // "C"
  0x00, 0x00, 0x7F, 0x41, 0x41, 0x3E, 0x00, 0x00, // "D"
  0x00, 0x00, 0x7F, 0x49, 0x49, 0x41, 0x00, 0x00, // "E"
  0x00, 0x00, 0x7F, 0x48, 0x48, 0x40, 0x00, 0x00,  // "F"
  B00000000, B00000000, B00111110, B01000001, B01001001, B00101110, B00000000, B00000000, //G
  B00000000, B00000000, B01111111, B00001000, B00001000, B01111111, B00000000, B00000000, //H
  B00000000, B00000000, B00000000, B01000001, B01111111, B01000001, B00000000, B00000000, //I
  B00000000, B00000000, B00000110, B00000001, B01000001, B01111110, B01000000, B00000000, //J
  B00000000, B00000000, B01111111, B00001000, B00010100, B00100010, B01000001, B00000000, //K
  B00000000, B00000000, B01111111, B00000001, B00000001, B00000001, B00000000, B00000000, //L
  B00000000, B00000000, B01111111, B00010000, B00001000, B00010000, B01111111, B00000000, //M
  B00000000, B00000000, B01111111, B00010000, B00001100, B00000010, B01111111, B00000000,//N
  B00000000, B00000000, B00111110, B01000001, B01000001, B00111110, B00000000, B00000000,//O
  B00000000, B00000000, B01111111, B01001000, B01001000, B00110000, B00000000, B00000000, //P
  B00000000, B00000000, B00111110, B01000001, B01000001, B00111111, B00000001, B00000000, //Q
  B00000000, B00000000, B01111111, B01001000, B01001100, B00110010, B00000001, B00000000, //R
```

```
  B00000000, B00000000, B00110010, B01001001, B01001001, B00100110, B00000000, B00000000, //S
  B00000000, B00000000, B01000000, B01000000, B01111111, B01000000, B01000000, B00000000, //T
  B00000000, B00000000, B01111110, B00000001, B00000001, B00000001, B01111110, B00000000, //U
  B00000000, B00000000, B01111100, B00000010, B00000001, B00000010, B01111100, B00000000, //V
  B00000000, B00000000, B01111111, B00000010, B00001100, B00000010, B01111111, B00000000, //W
  B00000000, B01000001, B00100010, B00010100, B00001000, B00010100, B00100010, B01000001, //X
  B00000000, B00000000, B01100000, B00010000, B00001111, B00010000, B01100000, B00000000, //Y
  B00000000, B00000000, B01100011, B01000101, B01001001, B01010001, B01100011, B00000000, //Z
  B00000000, B00000000, B00000000, B00000011, B00000011, B00000000, B00000000, B00000000,  // . OR PERIOD
  B00000000, B00000000, B00000000, B00000000, B00001101, B00001110, B00000000, B00000000,   // , COMMA
  B00000000, B00000000, B00000000, B01111101, B01111101, B00000000, B00000000, B00000000, // !
  B00000000, B00000000, B00110000, B01000000, B01000101, B01001000, B00110000, B00000000 // ?
};
```