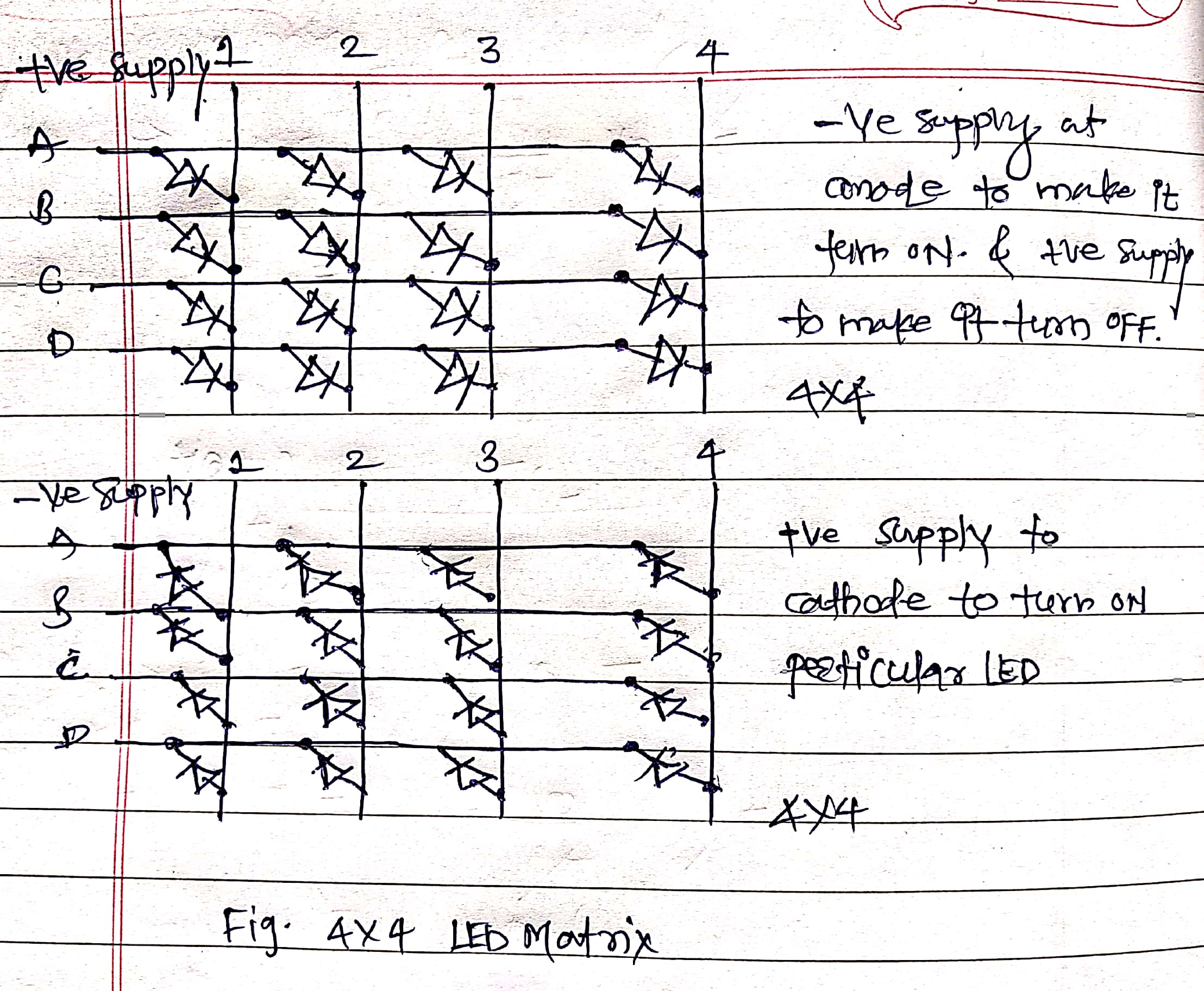
**Matrix of 1024 LED able to display text sent through the terminal design**

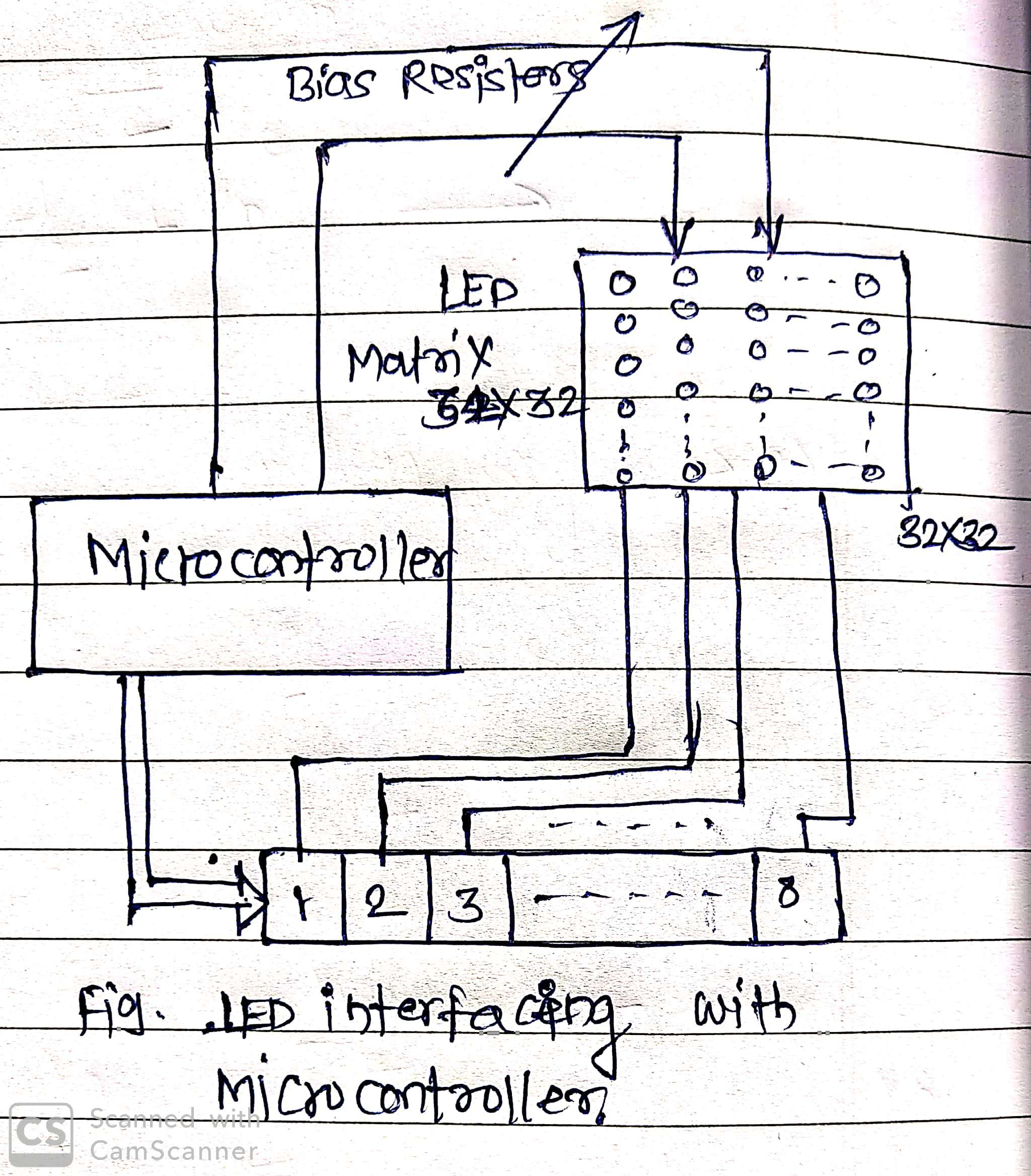
**Product we have designed has a display update rate is 20hz & also dissipate very less power**



In above figure I have shown the 4x4 led matrix module in which LED will be on based on the applied potential.

But we have to design 32x32 led matrix to display the different characters and numbers it has 1024 total number of LEDs. Most outdoor LED displays and some indoor LED displays are built around individually mounted LEDs. Presented here is a LED scrolling display that uses 64 LEDs to display alphabets and numbers. A cluster of red, green and blue diodes is driven together to form a full-colour display.

Each LED is addressed by its row and column number. In below figure we have shown the microcontroller interfacing with LED 32x32 matrix to display the text & characters.



Code:

#include <MaxMatrix.h>

**int** DIN = 7; // DIN pin of MAX7219 module

**int** CLK = 6; // CLK pin of MAX7219 module

**int** CS = 5; // CS pin of MAX7219 module

**int** maxInUse = 1;

MaxMatrix m(DIN, CS, CLK, maxInUse);

**char** A[] = {4, 8,

B01111110,

B00010001,

B00010001,

B01111110,

};

**char** B[] = {4, 8,

B01111111,

B01001001,

B01001001,

B00110110,

};

**char** smile01[] = {8, 8,

B00111100,

B01000010,

B10010101,

B10100001,

B10100001,

B10010101,

B01000010,

B00111100

};

**char** smile02[] = {8, 8,

B00111100,

B01000010,

B10010101,

B10010001,

B10010001,

B10010101,

B01000010,

B00111100

};

**char** smile03[] = {8, 8,

B00111100,

B01000010,

B10100101,

B10010001,

B10010001,

B10100101,

B01000010,

B00111100

};

**void** setup() {

m.init(); // MAX7219 initialization

m.setIntensity(8); // initial led matrix intensity, 0-15

}

**void** loop() {

// Seting the LEDs On or Off at x,y or row,column position

m.setDot(6,2,**true**);

delay(1000);

m.setDot(6,3,**true**);

delay(1000);

m.clear(); // Clears the display

**for** (**int** i=0; i<8; i++){

m.setDot(i,i,**true**);

delay(300);

}

m.clear();

// Displaying the character at x,y (upper left corner of the character)

m.writeSprite(2, 0, A);

delay(1000);

m.writeSprite(2, 0, B);

delay(1000);

m.writeSprite(0, 0, smile01);

delay(1000);

m. write Sprite(0, 0, smile02);

delay(1000);

m.write Sprite(0, 0, smile03);

delay(1000);

**for** (**int** i=0; i<8; i++){

m .shift Left (**false**, **false**);

delay(300);

}

m. clear();

}