```
int DisplayRow; //Used tto manage the position on the row
int DisplayColumn;
unsigned long timer;



// Code for handling LED display.// MAX7219 SPI LED Driver
#define MAX7219_TEST        0x0f00 // Display in Test mode
#define MAX7219_BRIGHTNESS  0x0a00 // Set brightness of display
#define MAX7219_SCAN_LIMIT  0x0b00 // Set Scan limit
#define MAX7219_DECODE_MODE 0x0900 // Sets chip to accept bit patterns
#define MAX7219_SHUTDOWN    0x0C00 // Code for shutdown chip.

void SPI_16(int data) {
  SPI.beginTransaction(SPISettings( 8000000, MSBFIRST, SPI_MODE0 )) //begins
the spi transaction
  digitalWrite (12, LOW); //sets the cs to low
  SPI.transfer16( data ) //transfers the data
  digitalWrite (12, HIGH); //set the cs to high again
  SPI.endTransaction(); //ends the transaction
}

void setup() {
  // put your setup code here, to run once:
  // Set up display These need to be sent in this fashion in setup.
  SPI_16(MAX7219_TEST + 0x01); // Turn on all the LEDs.
  delay(100);                     // One time we can use a delay.
  SPI_16(MAX7219_TEST + 0x00); // all LEDS off.
  SPI_16(MAX7219_DECODE_MODE + 0x00); // Disable BCD mode.
  SPI_16(MAX7219_BRIGHTNESS + 0x03);  // Use lower intensity.
  SPI_16(MAX7219_SCAN_LIMIT + 0x0f);  // Scan all digits.
  SPI_16(MAX7219_SHUTDOWN + 0x01);    // Turn on chip.
  DisplayColumn = 0;// Not valid column, but first pass should increment to
1.
  DisplayRow = 0;// This will effect the pattern generated.


  SPI.beginTransaction(SPISettings( 8000000, MSBFIRST, SPI_MODE0 ) ) //begins
transaction with settings
  timer = millis(); //sets timer equal to the amount of milliseconds since
the program has began
  pinMode (12, OUTPUT); //sets pin 12 to an output pin
  pinMode (13, OUTPUT); //sets pin 13 to an output pin
  SPI.begin();
}

void loop() {
  if ((millis() - timer) > 500) { //checks if it has been 500 milliseconds
since the last interval
      DisplayColumn++; //increments the collumn
      if (DisplayColumn > 8) { //if the column is above 8 then it will reset
it back to 1
        DisplayColumn = 1;
      }
      if (DisplayRow &= 0x0080) { //if bit 7 is high
        DisplayRow<< 1; //shift left
      }
      else { //else
        DisplayRow << 1; //shift left
        DisplayRow ^= 0x0001; //force bit 1 to be high
      }
      SPI_16 ( ( DisplayColumn << 8 ) + DisplayRow ); //call the spi 16
function


  }
}
```