LCD Module Non-blocking Driver

1 Description

The LCD module driver is implemented in a non-blocking way by relying on the `Timer0` interrupt functionality, such that where ever we need a delay we fire the interrupt and proceed to with the code until the interrupt raise a flag indicating the passage of the specified time.

2 LCD Module APIs

2.1 LCD Initialization

```c
/**
 * @brief Initializes the pins connected to the LCD display, clears the display,
 * and sets the cursor to the home position.
 */
void LCD_init(void);
```

2.2 LCD Send Command

```c
/**
 * @brief Sends a command to the LCD display.
 * @param[in] lcd_command The specified command to be sent to the LCD.
 */
void LCD_sendCommand(uint8 lcd_command);
```

2.3 LCD Display Character

```c
/**
 * @brief Displays the specified character on the LCD display.
 * @param[in] character The specified character to be displayed.
 */
void LCD_displayCharacter(uint8 character);
```

2.4 LCD Display String

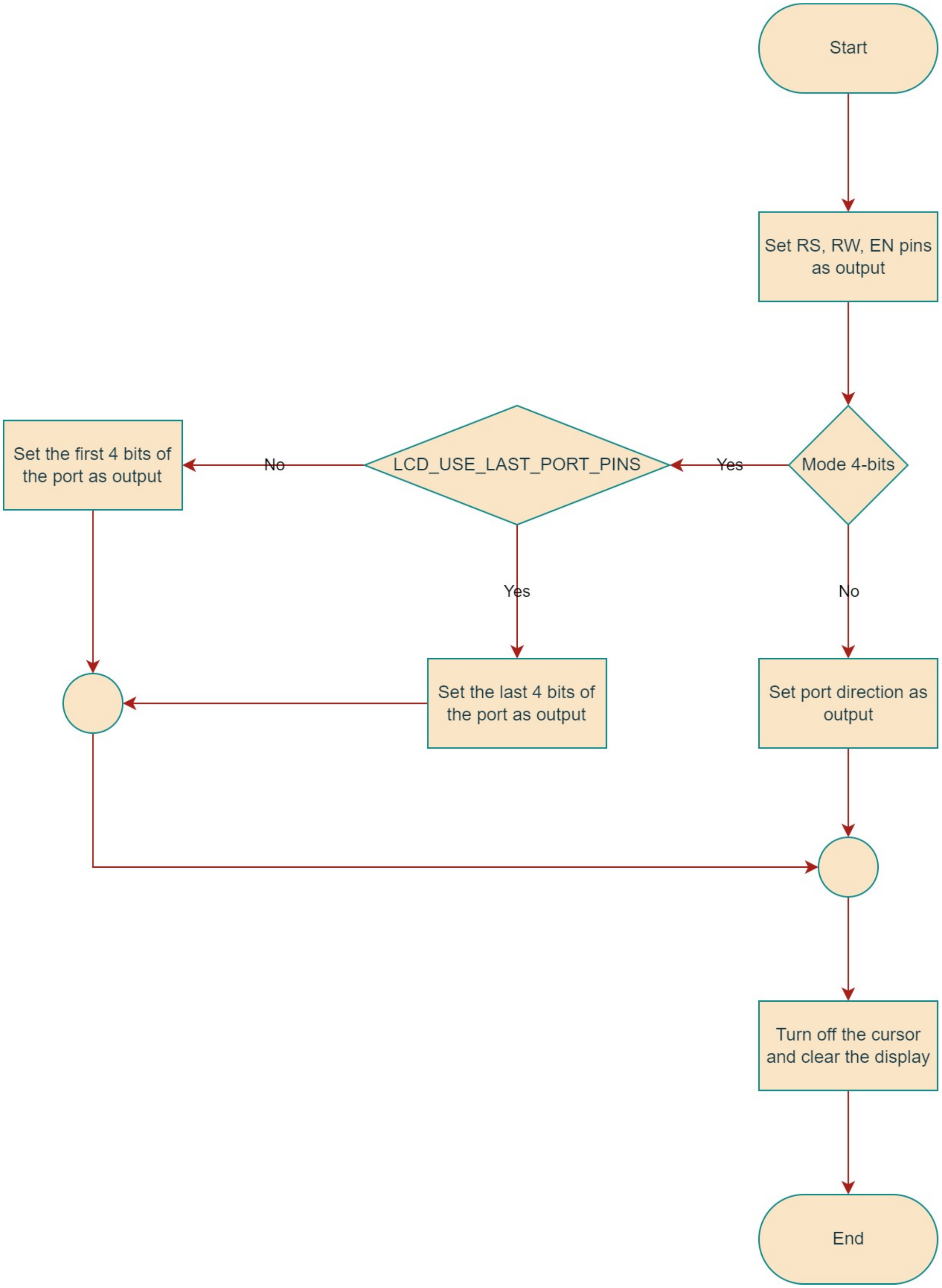
```
/**
 * @brief Displays a whole string on the LCD.
 * @param[in] p_string Address of the string to be displayed on the LCD.
 */
void LCD_displayString(uint8 *p_string);
```

2.5 LCD Move Cursor

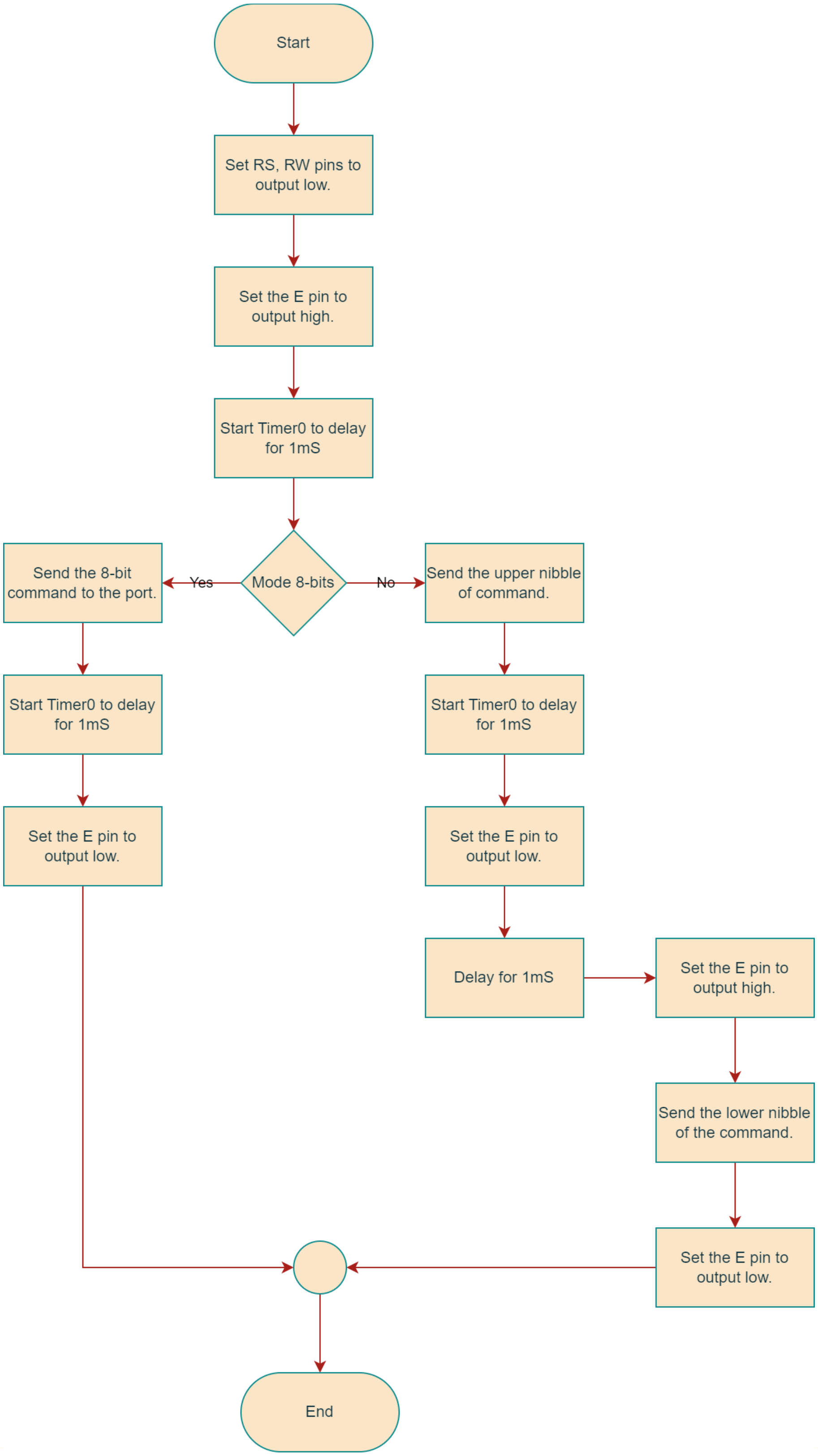
```
/**
 * @brief Moves the cursor to a specific row and column on the LCD display.
 * @param row The specified row, either row 0 or row 1.
 * @param col The specified column from 0 to 15.
 */
void LCD_moveCursor(uint8 row, uint8 col);
```

3 Low-level Design Flowcharts

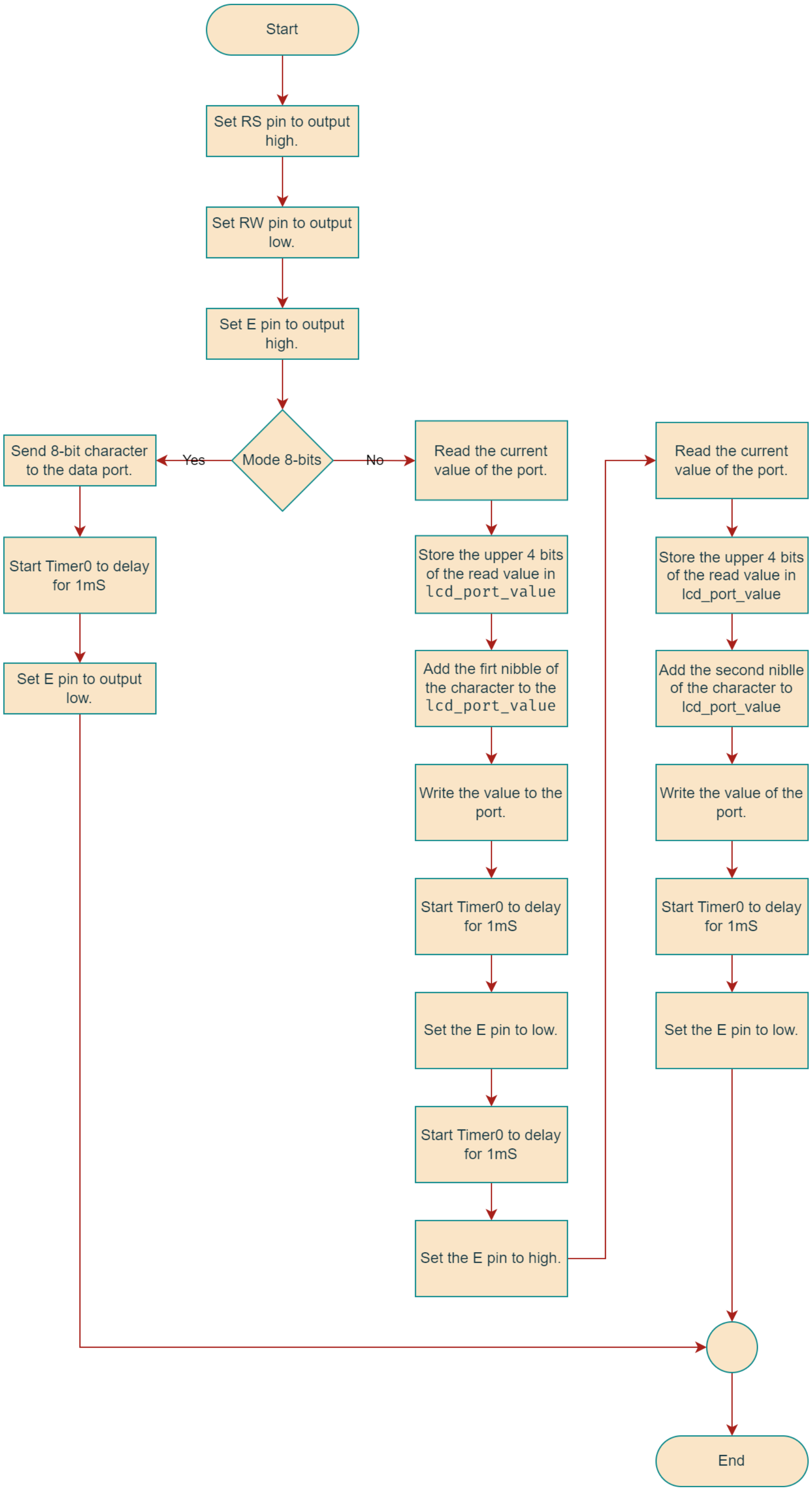
3.1 LCD Initialization Flowcharts


Start
Set RS, RW, EN pins as output
Mode 4-bits
Yes
LCD_USE_LAST_PORT_PINS
No
Set the first 4 bits of the port as output
Yes
Set the last 4 bits of the port as output
No
Set port direction as output
Turn off the cursor and clear the display
End

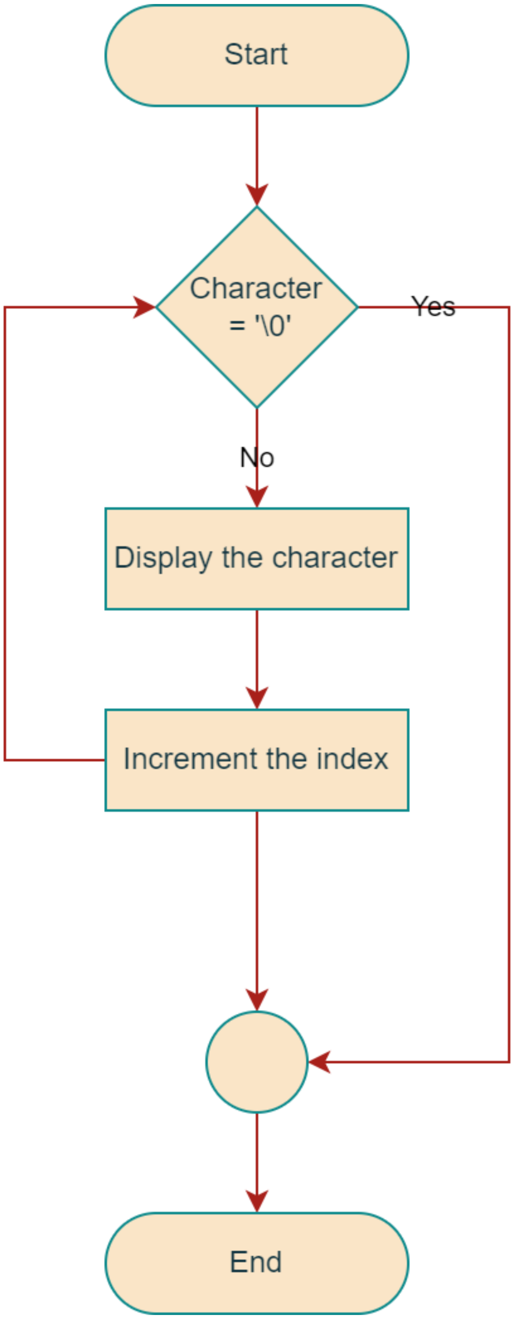
3.2 LCD Send Command Flowchart


Start
Set RS, RW pins to output low.
Set the E pin to output high.
Start Timer0 to delay for 1mS
Mode 8-bits
Yes
No
Send the 8-bit command to the port.
Start Timer0 to delay for 1mS
Set the E pin to output low.
Send the upper nibble of command.
Start Timer0 to delay for 1mS
Set the E pin to output low.
Delay for 1mS
Set the E pin to output high.
Send the lower nibble of the command.
Set the E pin to output low.
End

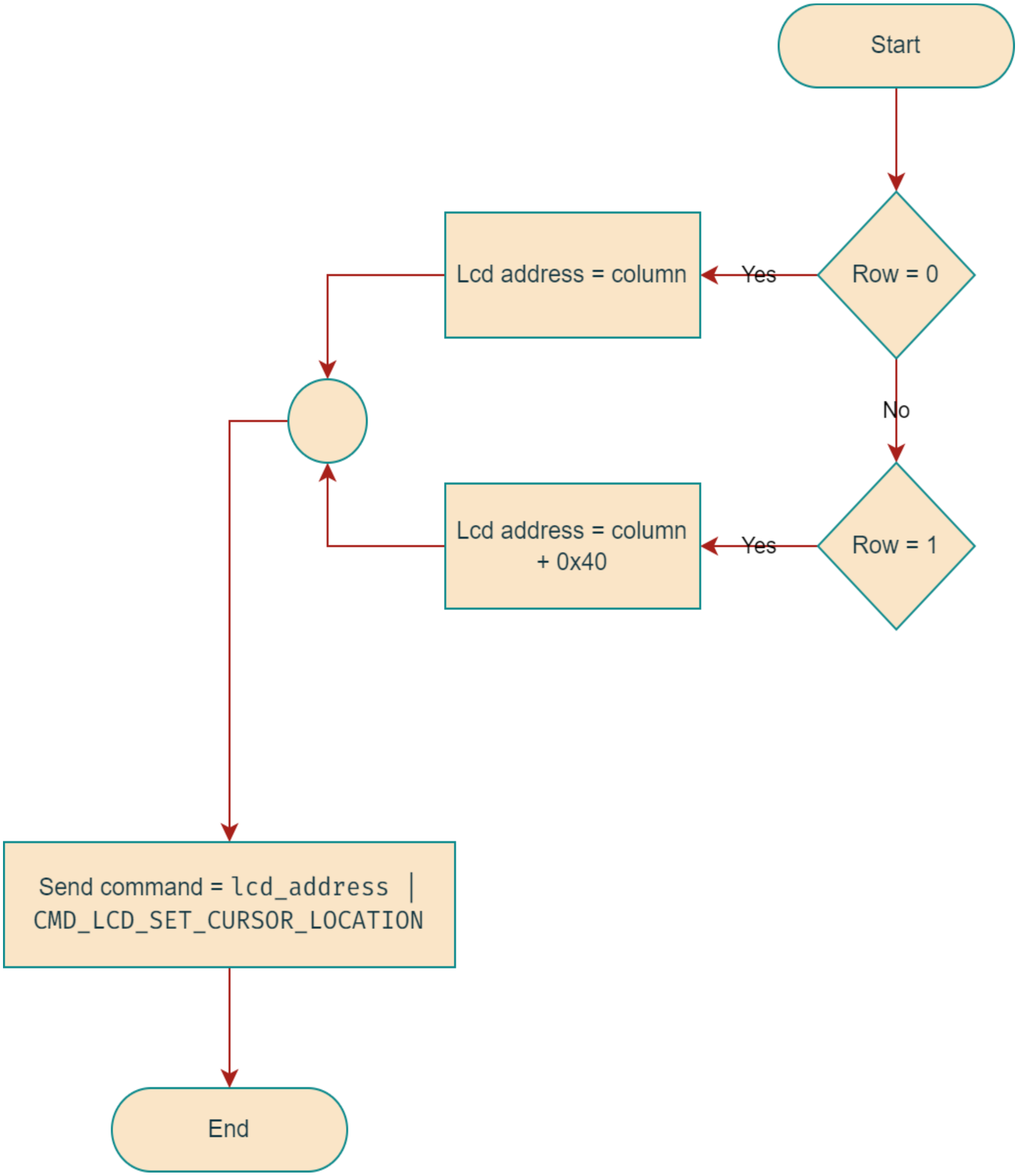
3.3 LCD Display Character


Start
Set RS pin to output high.
Set RW pin to output low.
Set E pin to output high.
Mode 8-bits
Yes
No
Send 8-bit character to the data port.
Start Timer0 to delay for 1mS
Set E pin to output low.
Read the current value of the port.
Store the upper 4 bits of the read value in lcd_port_value
Add the firt nibble of the character to the lcd_port_value
Write the value to the port.
Start Timer0 to delay for 1mS
Set the E pin to low.
Start Timer0 to delay for 1mS
Set the E pin to high.
Read the current value of the port.
Store the upper 4 bits of the read value in lcd_port_value
Add the second niblle of the character to lcd_port_value
Write the value of the port.
Start Timer0 to delay for 1mS
Set the E pin to low.
End

3.4 LCD Display String



3.5 LCD Move Cursor


Start
Row = 0
Yes
Lcd address = column
No
Row = 1
Yes
Lcd address = column + 0x40
Send command = lcd_address | CMD_LCD_SET_CURSOR_LOCATION
End