Nokia 3220 LCD Interface Tutorial

(Brief Introduction)

Version 1.1

Dated August 6th 2010

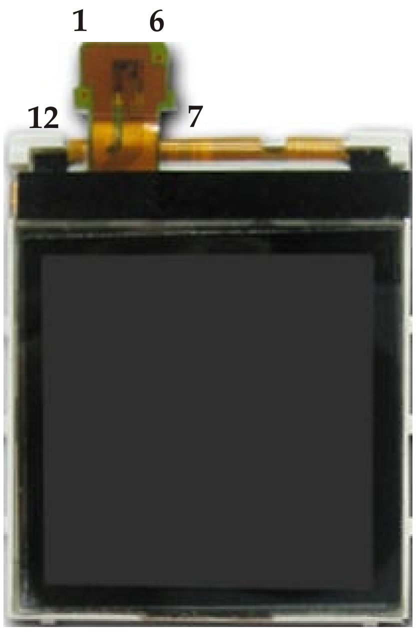
Author: Imran Naeem

Email: elect_design_eng@yahoo.com

NOKIA 3220 LCD
THIS IS ..
NOKIA 3220 LCD
FIRST TIME
HACKED BY..
IMRAN NAEEM
PAKISTAN
DATED.18-03-2010
* IMRAN NAEEM *

Introduction.

128x128xRGB Pixels like 6100 LCD , PCF8833 or compatible COG with 9-bit SPI like interface. See next page for SPI interface details ..

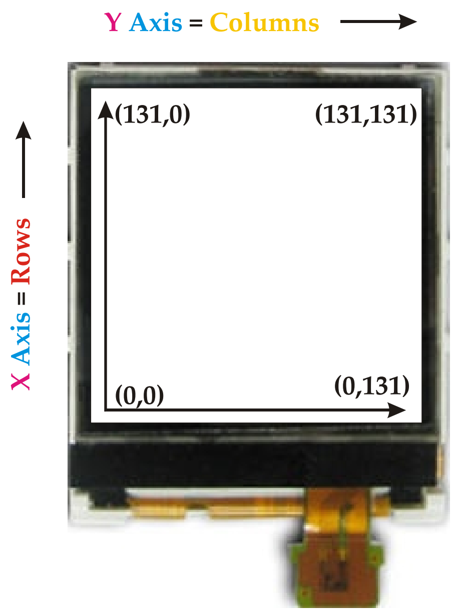


1	VLED-
2	1.8V
3	GND
4	SDA
5	CS
6	GND
7	N.C
8	RESET
9	SCLK
10	GND
11	2.8V
12	VLED+

Pins Assignments

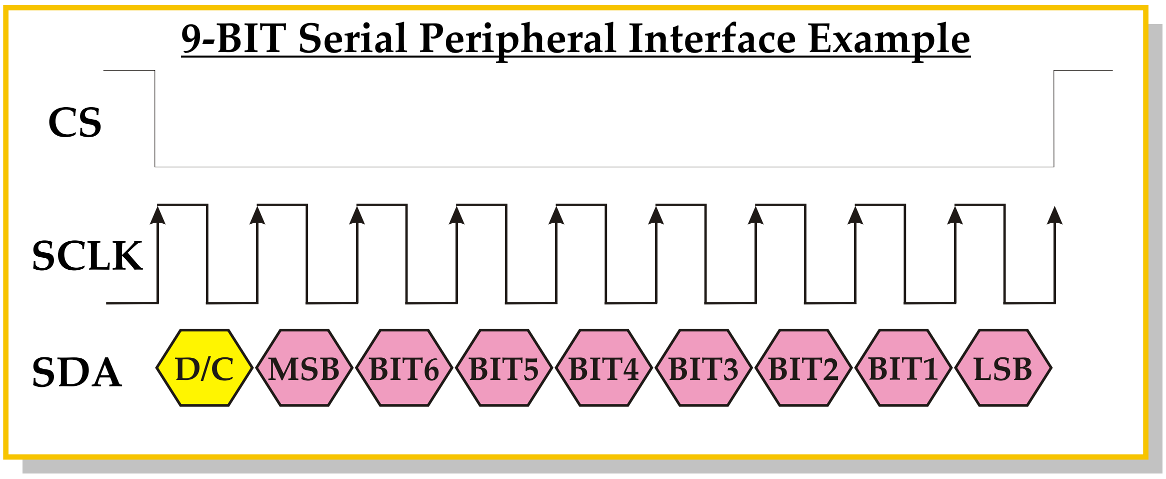
The orientation of display is based on the memory access control register settings ,, (0x36) MADCTL of COG.

Remember ,, X-Axis and Y-Axis are swapped in these type of LCDs ..

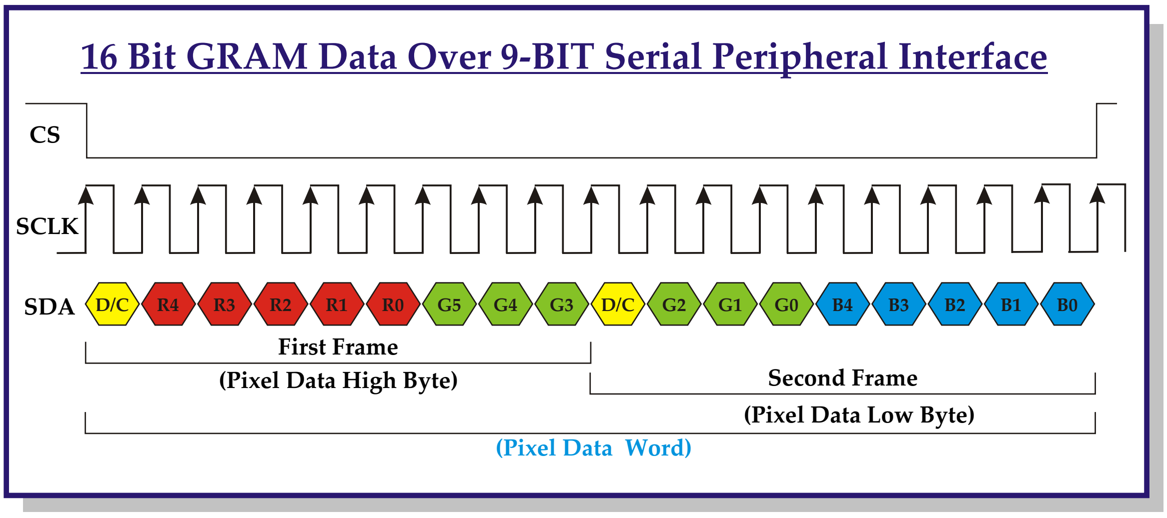


Default Orientation

Communication with the Display



16 BIT = 65K COLORS DISPLAY DATA TRANSMIT



For complete codes ,, I'll refer you James P. Lynch's Nokia 6100 interface tutorial ,, because I have partially used his codes after some modification for AVR type of MCUs.

I have included the fonts.h (Only small size) and fontcolors.h headers files in this tutorial ..

By the way ,, Till date ,, I have hacked the following Nokia Color LCDs with 100% controls. Nokia 6600/7610 , E51 , E66 , E71 , 2700 Classic , 6500 Slider , N73 .. Any many more on the ways ..

```
// *********************************************************************************
// Font tables for Nokia3220 LCD Display Driver
// FONT6x8 - SMALL font (mostly 5x7)
// FONT8x8 - MEDIUM font (8x8 characters, a bit thicker)
// FONT8x16 - LARGE font (8x16 characters, thicker)
// Author: Jim Parise, James P Lynch July 7, 2007
// *********************************************************************************

const unsigned char FONT6x8[] PROGMEM = {
  0x06,0x08,0x08,0x00,0x00,0x00,0x00,0x00, // columns, rows,
num_bytes_per_char
  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // space 0x20
  0x20,0x20,0x20,0x20,0x20,0x00,0x20,0x00, // !
  0x50,0x50,0x50,0x00,0x00,0x00,0x00,0x00, // "
  0x50,0x50,0xF8,0x50,0xF8,0x50,0x50,0x00, // #
  0x20,0x78,0xA0,0x70,0x28,0xF0,0x20,0x00, // $
  0xC0,0xC8,0x10,0x20,0x40,0x98,0x18,0x00, // %
  0x40,0xA0,0xA0,0x40,0xA8,0x90,0x68,0x00, // &
  0x30,0x30,0x20,0x40,0x00,0x00,0x00,0x00, // '
  0x10,0x20,0x40,0x40,0x40,0x20,0x10,0x00, // (
  0x40,0x20,0x10,0x10,0x10,0x20,0x40,0x00, // )
  0x00,0x20,0xA8,0x70,0x70,0xA8,0x20,0x00, // *
  0x00,0x20,0x20,0xF8,0x20,0x20,0x00,0x00, // +
  0x00,0x00,0x00,0x00,0x30,0x30,0x20,0x40, // ,
  0x00,0x00,0x00,0xF8,0x00,0x00,0x00,0x00, // -
  0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00, // .
  0x00,0x08,0x10,0x20,0x40,0x80,0x00,0x00, // / (forward slash)
  0x70,0x88,0x88,0xA8,0x88,0x88,0x70,0x00, // 0 0x30
  0x20,0x60,0x20,0x20,0x20,0x20,0x70,0x00, // 1
  0x70,0x88,0x08,0x70,0x80,0x80,0xF8,0x00, // 2
  0xF8,0x08,0x10,0x30,0x08,0x88,0x70,0x00, // 3
  0x10,0x30,0x50,0x90,0xF8,0x10,0x10,0x00, // 4
  0xF8,0x80,0xF0,0x08,0x08,0x88,0x70,0x00, // 5
  0x38,0x40,0x80,0xF0,0x88,0x88,0x70,0x00, // 6
  0xF8,0x08,0x08,0x10,0x20,0x40,0x80,0x00, // 7
  0x70,0x88,0x88,0x70,0x88,0x88,0x70,0x00, // 8
  0x70,0x88,0x88,0x78,0x08,0x10,0xE0,0x00, // 9
  0x00,0x00,0x20,0x00,0x20,0x00,0x00,0x00, // :
  0x00,0x00,0x20,0x00,0x20,0x20,0x40,0x00, // ;
  0x08,0x10,0x20,0x40,0x20,0x10,0x08,0x00, // <
  0x00,0x00,0xF8,0x00,0xF8,0x00,0x00,0x00, // =
  0x40,0x20,0x10,0x08,0x10,0x20,0x40,0x00, // >
  0x70,0x88,0x08,0x30,0x20,0x00,0x20,0x00, // ?
  0x70,0x88,0xA8,0xB8,0xB0,0x80,0x78,0x00, // @ 0x40
  0x20,0x50,0x88,0x88,0xF8,0x88,0x88,0x00, // A
  0xF0,0x88,0x88,0xF0,0x88,0x88,0xF0,0x00, // B
  0x70,0x88,0x80,0x80,0x80,0x88,0x70,0x00, // C
  0xF0,0x88,0x88,0x88,0x88,0x88,0xF0,0x00, // D
  0xF8,0x80,0x80,0xF0,0x80,0x80,0xF8,0x00, // E
  0xF8,0x80,0x80,0xF0,0x80,0x80,0x80,0x00, // F
  0x78,0x88,0x80,0x80,0x98,0x88,0x78,0x00, // G
  0x88,0x88,0x88,0xF8,0x88,0x88,0x88,0x00, // H
  0x70,0x20,0x20,0x20,0x20,0x20,0x70,0x00, // I
  0x38,0x10,0x10,0x10,0x10,0x90,0x60,0x00, // J
  0x88,0x90,0xA0,0xC0,0xA0,0x90,0x88,0x00, // K
  0x80,0x80,0x80,0x80,0x80,0x80,0xF8,0x00, // L
0x88,0xD8,0xA8,0xA8,0xA8,0x88,0x88,0x00, // M
0x88,0x88,0xC8,0xA8,0x98,0x88,0x88,0x00, // N
0x70,0x88,0x88,0x88,0x88,0x88,0x70,0x00, // O
0xF0,0x88,0x88,0xF0,0x80,0x80,0x80,0x00, // P 0x50
0x70,0x88,0x88,0x88,0xA8,0x90,0x68,0x00, // Q
0xF0,0x88,0x88,0xF0,0xA0,0x90,0x88,0x00, // R
0x70,0x88,0x80,0x70,0x08,0x88,0x70,0x00, // S
0xF8,0xA8,0x20,0x20,0x20,0x20,0x20,0x00, // T
0x88,0x88,0x88,0x88,0x88,0x88,0x70,0x00, // U
0x88,0x88,0x88,0x88,0x88,0x50,0x20,0x00, // V
0x88,0x88,0x88,0xA8,0xA8,0xA8,0x50,0x00, // W
0x88,0x88,0x50,0x20,0x50,0x88,0x88,0x00, // X
0x88,0x88,0x50,0x20,0x20,0x20,0x20,0x00, // Y
0xF8,0x08,0x10,0x70,0x40,0x80,0xF8,0x00, // Z
0x78,0x40,0x40,0x40,0x40,0x40,0x78,0x00, // [
0x00,0x80,0x40,0x20,0x10,0x08,0x00,0x00, // \ (back slash)
0x78,0x08,0x08,0x08,0x08,0x08,0x78,0x00, // ]
0x20,0x50,0x88,0x00,0x00,0x00,0x00,0x00, // ^
0x00,0x00,0x00,0x00,0x00,0x00,0xF8,0x00, // _
0x60,0x60,0x20,0x10,0x00,0x00,0x00,0x00, // ` 0x60
0x00,0x00,0x60,0x10,0x70,0x90,0x78,0x00, // a
0x80,0x80,0xB0,0xC8,0x88,0xC8,0xB0,0x00, // b
0x00,0x00,0x70,0x88,0x80,0x88,0x70,0x00, // c
0x08,0x08,0x68,0x98,0x88,0x98,0x68,0x00, // d
0x00,0x00,0x70,0x88,0xF8,0x80,0x70,0x00, // e
0x10,0x28,0x20,0x70,0x20,0x20,0x20,0x00, // f
0x00,0x00,0x70,0x98,0x98,0x68,0x08,0x70, // g
0x80,0x80,0xB0,0xC8,0x88,0x88,0x88,0x00, // h
0x20,0x00,0x60,0x20,0x20,0x20,0x70,0x00, // i
0x10,0x00,0x10,0x10,0x10,0x90,0x60,0x00, // j
0x80,0x80,0x90,0xA0,0xC0,0xA0,0x90,0x00, // k
0x60,0x20,0x20,0x20,0x20,0x20,0x70,0x00, // l
0x00,0x00,0xD0,0xA8,0xA8,0xA8,0xA8,0x00, // m
0x00,0x00,0xB0,0xC8,0x88,0x88,0x88,0x00, // n
0x00,0x00,0x70,0x88,0x88,0x88,0x70,0x00, // o
0x00,0x00,0xB0,0xC8,0xC8,0xB0,0x80,0x80, // p 0x70
0x00,0x00,0x68,0x98,0x98,0x68,0x08,0x08, // q
0x00,0x00,0xB0,0xC8,0x80,0x80,0x80,0x00, // r
0x00,0x00,0x78,0x80,0x70,0x08,0xF0,0x00, // s
0x20,0x20,0xF8,0x20,0x20,0x28,0x10,0x00, // t
0x00,0x00,0x88,0x88,0x88,0x98,0x68,0x00, // u
0x00,0x00,0x88,0x88,0x88,0x50,0x20,0x00, // v
0x00,0x00,0x88,0x88,0xA8,0xA8,0x50,0x00, // w
0x00,0x00,0x88,0x50,0x20,0x50,0x88,0x00, // x
0x00,0x00,0x88,0x88,0x78,0x08,0x88,0x70, // y
0x00,0x00,0xF8,0x10,0x20,0x40,0xF8,0x00, // z
0x10,0x20,0x20,0x40,0x20,0x20,0x10,0x00, // {
0x20,0x20,0x20,0x00,0x20,0x20,0x20,0x00, // |
0x40,0x20,0x20,0x10,0x20,0x20,0x40,0x00, // }
0x40,0xA8,0x10,0x00,0x00,0x00,0x00,0x00, // ~
0x70,0xD8,0xD8,0x70,0x00,0x00,0x00,0x00}; // DEL
```

```
#ifndef fontcolors_h
#define fontcolors_h

// Booleans
#define NOFILL 0
#define FILL 1

// 16-bit color definitions
#define         BLACK           0x0000
#define         BLUE            0x001F
#define         NBLUE           0x004E
#define         RED             0xF800
#define         GREEN           0x07E0
#define         CYAN            0x3DFF
#Define         MAGENTA         0xF81F
#define         PURPLE           0x7074
#define         YELLOW          0xEEC1
#define         WHITE           0xFFFF
#define         BROWN           0x92E1
#define         PINK            0xEB78
#define         ROSE            0xE84E
#define         ORANGE          0xEBA5
#define         GREY            0xDEFA
#define         DGREY           0x3187

// Font sizes
#define SMALL 0
#define MEDIUM 1
#define LARGE 2

// mask definitions
#define         BIT0    0x00000001
#define         BIT1    0x00000002
#define         BIT2    0x00000004
#define         BIT3    0x00000008
#define         BIT4    0x00000010
#define         BIT5    0x00000020
#define         BIT6    0x00000040
#define         BIT7    0x00000080
#define         BIT8    0x00000100
#define         BIT9    0x00000200
#define         BIT10   0x00000400
#define         BIT11   0x00000800
#define         BIT12   0x00001000
#define         BIT13   0x00002000
#define         BIT14   0x00004000
#define         BIT15   0x00008000
#define         BIT16   0x00010000
#define         BIT17   0x00020000
#define         BIT18   0x00040000
#define         BIT19   0x00080000
#define         BIT20   0x00100000
#define         BIT21   0x00200000
#define         BIT22   0x00400000
#define         BIT23   0x00800000
#define         BIT24   0x01000000
#define         BIT25   0x02000000
#define         BIT26   0x04000000
#define         BIT27   0x08000000
#define         BIT28   0x10000000
#define         BIT29   0x20000000
#define         BIT30   0x40000000
#define         BIT31   0x80000000
#endif // fontcolors_h
```