DKAN0012A

Controlling the NKK 64x32 SmartSwitch and SmartDisplay

03 February 2010

Features

- IS01DBFRGB LCD SmartDisplay from NKK Switches
- Simple implementation featuring the ATmega88PA from Atmel
- Complete software solution

Introduction

The IS01DBFRGB SmartDisplay is used in every 64x32 LCD SmartSwitch. It features a 64x32 pixel monochrome LCD with red, green, and blue LED backlights. This application note describes the display's interface and provides an example implementation using a SPI peripheral. Complete C source code for Atmel's AVR ATmega88PA is provided.

Application

LCD

The IS01DBFRGB display has a high level of integration compared to the 36x24 pixel IS01BFRGB SmartDisplay, making this display much simpler to control. The display does not need to be refreshed continuously, and the LED backlights are controlled directly via SPI. Four commands are used to display data, set the color of the backlights, set the brightness of the backlights, and reset the display to the default state. The display has 6 pins: power (Vdd), ground (GND), and the rest for SPI communication, which consist of Serial ClocK (SCK), Slave Select (SS), Serial Data Out (SDO), and Serial Data In (SDI).

Commands and Data

The IS01DBFRGB display has four valid commands. The Display Data command (0x55) displays the next 256 bytes of data on the LCD. The Backlight color command (0x40) sets the color intensity of each LED. The Backlight Brightness command (0x41) sets the brightness of the entire display. Last, the Reset command (0x5E) returns the display to its initial status (determined at power activation). Table 1 summarizes these commands.

Table 1. Command Set

Command	Hex	Binary	Data	Remarks
Display Data	0x55	0b01010101	256 Bytes (64x32 = 2048 bits)	See Datasheet for details
Color Set	0x40	0b01000000	R R G G B B 1 1	For each RGB: 00 = off, 01 = 1/4, 10 = 1/2, 11 = Full
Color Brightness	0x41	0b01000001	X X X 1 1 1 1 1	For leading 3 bits: 000 = 1/20 (dark), 001 = 1/10, 010 = 1/7, 011 = 1/5, 100 = 1/3, 101 = 1/2, 110 = 2/3, 111 = 1 (bright)
Reset Display	0x5E	0b01011110	0 0 0 0 0 0 1 1	Return to initial status

Example Implementation

Code

Figure 1 illustrates the program flow chart.

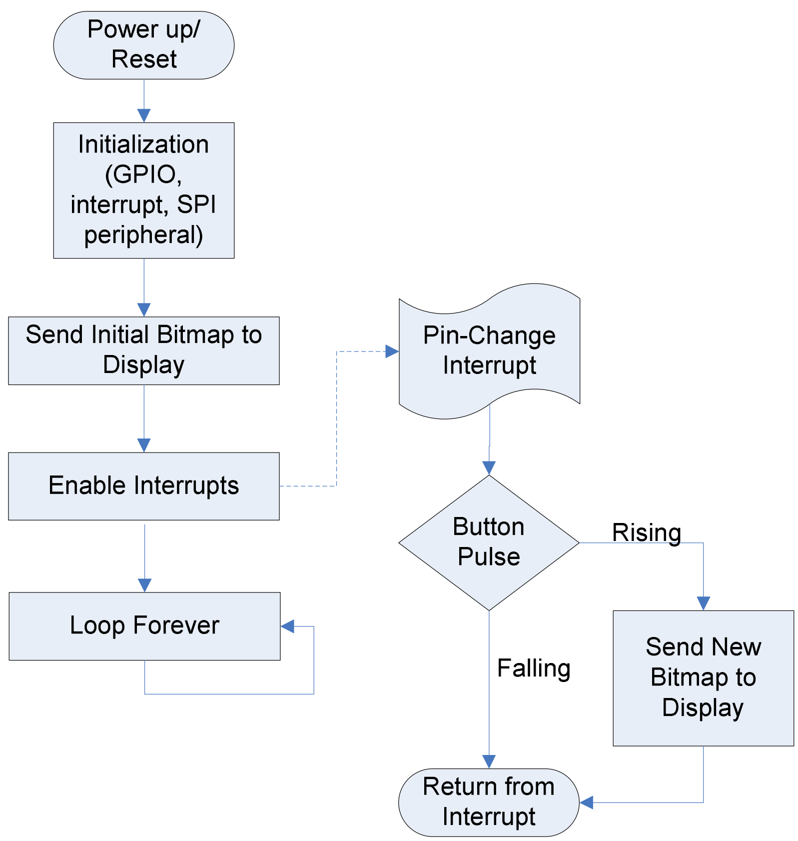

Figure 1. Program Flow Chart

Initialization

The program begins by initializing GPIO, interrupts, and the SPI peripheral. The SPI peripheral is configured as a SPI master using mode 2 for the communication parameters. Then, the display is configured with an initial backlight color, and brightness. Next, the initial bitmap data is sent to the display. Finally, a pin change interrupt is enabled on the GPIO pin connected to the switch, and the program waits for a pin change interrupt.

Pin Change Interrupt

An interrupt is generated on both a button press and a button release. After a button release, the interrupt service routine displays a new bitmap.

Bitmap

The bitmap is formatted to replicate the 64x32 display format. Each bit sets a pixel state (1=on, 0=off). The bitmaps are stored in the internal Flash, and each requires 256 bytes of memory. Figure 2 illustrates the format of the bitmap used in the example code.

```
unsigned char IS01DBFRGB_bitmap_3[32][8] PROGMEM=
{
  { 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111 },    //Line #1
  { 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111 },    //Line #2
  { 0b11111111, 0b11111111, 0b11111111, 0b11110000, 0b00001111, 0b11111111, 0b11111111, 0b11111111 },    //Line #3
  { 0b11111111, 0b11111111, 0b11111111, 0b00000000, 0b00000001, 0b11111111, 0b11111111, 0b11111111 },    //Line #4
  { 0b11111111, 0b11111111, 0b11111100, 0b00000000, 0b00000000, 0b01111111, 0b11111111, 0b11111111 },    //Line #5
  { 0b11111111, 0b11111111, 0b11111000, 0b00000000, 0b00000000, 0b00111111, 0b11111111, 0b11111111 },    //Line #6
  { 0b11111111, 0b11111111, 0b11110000, 0b00000000, 0b00000000, 0b00011111, 0b11111111, 0b11111111 },    //Line #7
  { 0b11111111, 0b11111111, 0b11100000, 0b00000000, 0b00000000, 0b00001111, 0b11111111, 0b11111111 },    //Line #8
  { 0b11111111, 0b11111111, 0b11000000, 0b00000000, 0b00000000, 0b00000111, 0b11111111, 0b11111111 },    //Line #9
  { 0b11111111, 0b11111111, 0b10000000, 0b01000000, 0b00000100, 0b00000011, 0b11111111, 0b11111111 },    //Line #10
  { 0b11111111, 0b11111111, 0b10000000, 0b11100000, 0b00001110, 0b00000011, 0b11111111, 0b11111111 },    //Line #11
  { 0b11111111, 0b11111111, 0b00000001, 0b11110000, 0b00011111, 0b00000001, 0b11111111, 0b11111111 },    //Line #12
  { 0b11111111, 0b11111111, 0b00000000, 0b11100000, 0b00001110, 0b00000001, 0b11111111, 0b11111111 },    //Line #13
  { 0b11111111, 0b11111110, 0b00000000, 0b01000000, 0b00000100, 0b00000000, 0b11111111, 0b11111111 },    //Line #14
  { 0b11111111, 0b11111110, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b11111111, 0b11111111 },    //Line #15
  { 0b11111111, 0b11111110, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b11111111, 0b11111111 },    //Line #16
  { 0b11111111, 0b11111110, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b11111111, 0b11111111 },    //Line #17
  { 0b11111111, 0b11111110, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b11111111, 0b11111111 },    //Line #18
  { 0b11111111, 0b11111111, 0b00000000, 0b00000000, 0b00000000, 0b00000001, 0b11111111, 0b11111111 },    //Line #19
  { 0b11111111, 0b11111111, 0b00000000, 0b00000000, 0b00000000, 0b00000001, 0b11111111, 0b11111111 },    //Line #20
  { 0b11111111, 0b11111111, 0b10000000, 0b00000000, 0b00000000, 0b00000011, 0b11111111, 0b11111111 },    //Line #21
  { 0b11111111, 0b11111111, 0b10000001, 0b10000000, 0b00000011, 0b00000011, 0b11111111, 0b11111111 },    //Line #22
  { 0b11111111, 0b11111111, 0b11000001, 0b11100000, 0b00001111, 0b00000111, 0b11111111, 0b11111111 },    //Line #23
  { 0b11111111, 0b11111111, 0b11100000, 0b11111111, 0b11111110, 0b00001111, 0b11111111, 0b11111111 },    //Line #24
  { 0b11111111, 0b11111111, 0b11110000, 0b00011111, 0b11110000, 0b00011111, 0b11111111, 0b11111111 },    //Line #25
  { 0b11111111, 0b11111111, 0b11111000, 0b00000000, 0b00000000, 0b00111111, 0b11111111, 0b11111111 },    //Line #26
  { 0b11111111, 0b11111111, 0b11111100, 0b00000000, 0b00000000, 0b01111111, 0b11111111, 0b11111111 },    //Line #27
  { 0b11111111, 0b11111111, 0b11111111, 0b00000000, 0b00000001, 0b11111111, 0b11111111, 0b11111111 },    //Line #28
  { 0b11111111, 0b11111111, 0b11111111, 0b11100000, 0b00001111, 0b11111111, 0b11111111, 0b11111111 },    //Line #29
  { 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111 },    //Line #30
  { 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111 },    //Line #31
  { 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111, 0b11111111 },    //Line #32
};
```

Figure 2. Bitmap Example

Considerations

The bitmaps are stored in the flash memory, so the number of possible bitmaps depends on the size of the microcontroller's Flash memory. Also, the SPI port is configured to use mode 2; other devices may need to use a different mode.

Conclusion

This application note presents a method of using the SPI interface to control the NKK 64x32 SmartDisplays and SmartSwitches. It provides a complete software solution utilizing an ATmega88PA, allowing a simple, effective implementation.

Disclaimer

This document is for informational use only and is subject to change without prior notice. Digi-Key makes no commitment to update or keep current the information contained herein. Digi-Key does not guarantee or warrant that any information provided is accurate, complete, or correct and disclaims any and all liability associated with the use of the information contained herein. The use of this information and Digi-Key's liability is subject to Digi-Key's standard Terms & Conditions which can be found at www.digi-key.com by clicking on the Terms & Conditions link at the bottom of the web page.

No license, whether express, implied, arising by estoppel or otherwise is granted under any intellectual property or other rights of Digi-Key or others.

Trademarks

DIGI-KEY® is a registered trademark of Digi-Key Corporation. All other trademarks, service marks and product names contained herein are the sole property of their respective owner and their use is for informational purposes only and does not imply any endorsement, recommendation, sponsorship or approval by the trademark owner of the contents.

Copyright