olikraus / **u8g2**

Work around for glitches on a 128x64 ST7920 display #367

New issue

Closed **martinmarcos** opened this issue on 21 Sep 2017 · 5 comments

martinmarcos commented on 21 Sep 2017

First of all, I would like to say I really like the library. You can notice immediately there is a lot work put into it.

I am trying to use it to control a 12864B V2.0 LCD display that apparently comes with an ST7920 controller. I am using it with an Arduino Mega2560. If a run the "Hello World" example through HW SPI on the display alone, it works perfectly. Unfortunately the application I am trying to develop also needs to use the hardware SPI bus to communicate with an SD card (for which I am using the SDFat library by the way). When using both, the LCD display and SD card a problem occurs. Again I am just using the "Hello World" example. It's a weird behavior, I get random lines and dots drawn across the bottom half of the screen, and only the bottom half. Its even weirder because the lines and dots don't stay drawn in the same place (its like they move around) but the "Hello World" does stay in the drawn the same place and doesn't get corrupt. The data been written to the SD doesn't get corrupted neither. For the SD card I using the default HW SS/CS line (Digital IO pin 53 on the Mega2560) and for the LCD I am using a different Digital IO (Pin 22 in this case). In my setup I have soldered the display's optional jumper 2 (JP2) middle pad to the "S" pad, which saves me having to ground the PSB in order to select Serial mode.

Looking around on the web for someone having a similar problem and I stumbled upon these 2 forum threads, How to properly connect LCD 128x64 via SPI? and Problem addressing two displays in U8GLIB. The former of the threads mentions that the problem is concerning the "E" pin on the LCD, which in Serial mode should function as clock signal (SCLK). It has been noted that in the v4.0 (I was using v3.0 and it didn't mention anything) of the ST7920's datasheet, more specifically on page 7, under the function explanation for the RS(CS) pin, you can read the following "When chip is disabled, SID and SCLK should be set as “H” or “L”. Transcient of SID and CLK is not allowed.". So it seems that it's not enough to use the normal SPI CS signal to get the display to be unresponsive to transients on the clock and data lines, you also have to ground (or set to Vcc) those lines, which totally defeats the purpose of the CS pin in the first place! In the thread, the solution that was given was to use an AND gate (a 74hct08 to be exact) between the CS line and both the clock and data lines in order to drive these lines low whenever the CS line is low.

I trying to figure out how to solve this. I don't have a AND gate lying around that I can use and it's not easy for me to get one (If it turns out I have no other way of solving it, I'll go the lengths to get one). My other options is to find a solution software wise. I could use the SW SPI. I've read around that it's slower that HW SPI, but exactly how much slower? How much time does it take to draw the complete 128x64 screen with the SW SPI? Does it depend on the amount of text or graphics getting drawn? Does the u8g2 SW SPI implementation disable interrupts and any moment? And if so, for how long? I am timing tasks by using the millis() function which uses interrupts and although I can bear some jitter I am concerned with getting timing messed up. The other solution would be to use parallel communication which I am sure is faster than SW SPI, right? If it's bit-banged, interrupts are going to be disable, so same story with timing, but I was thinking that if it's faster, it takes less time and so I've end up having less discrepancy, wouldn't I? Also, the ST7920 can work with 4 bit parallel communication, does the u8g2 library support that? And if so, how would you configure it.

Regards.
Martin.

olikraus commented on 22 Sep 2017 Owner

I collect measures here:
https://github.com/olikraus/u8g2/blob/master/sys/arduino/u8g2_page_buffer/FPS/FPS.ino#L154
They are not neccessarily up to date, but might give a feeling on the speed. Actually, SW SPI will be slow.

U8g2 does not support the 4-bit interface for the ST7920.

From the feedback of the last few years, I would say the ST7920 is the most strange and unreliable controller out there. If it is true, what you wrote about CS line, I would say, that makes this device even more unreliable.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

3 participants

martinmarcos commented on 22 Sep 2017

@olikraus Thanks for the timing info, not sure if I'm interpreting it correctly though. From these lines it would seem that there is practically no difference in speed between SW and HW SPI. That can't be right, can it? In a previous entry from August 2016 you can see what you've expect, HW SPI having a lot more FPS than SW SPI. So what changed? Also this entry shows that what I was theorizing about parallel being faster than SW SPI wouldn't be correct, apparently they are roughly the same speed.

By the way, I was able to get my hands on a quad 2-input AND gate (HD74LS08). I wired it like it was explained on the forum thread and I can confirm that the erroneous behavior (the random lines and dots) experienced before is completely gone. The only annoyance that I have now is a slight dimming of the display every time the Arduino writes to the SD card, which I think has nothing to do with u8g2. I'm using a chinese clone of the catalex microsd card adapter and whenever I disconnect it the dimming ceases. The adapter comes with an AMS1117 3.3v regulator and a LVC125A level converter. I not sure if it's due to a design error or if it justs draws enough current from the power lines during the write cycle to dim the display. I'll investigate.

I'm leaving the issue open to see if you have some input on my interpretation of the timing info, but other than that the issue is pretty much closed.

Thanks for this great library and for your comments.

olikraus commented on 22 Sep 2017 Owner

Your input on the CS line is really interesting. Thanks for the input.

SW SPI is 3 to 10 times slower (according to my measures)

olikraus commented on 30 Sep 2017 Owner

closing...

olikraus closed this on 30 Sep 2017

RobertFach referenced this issue on 9 Jan

128x64 ST7920 SPI Atmega328p pixel problem #377 Open

Polle0 commented on 16 Mar

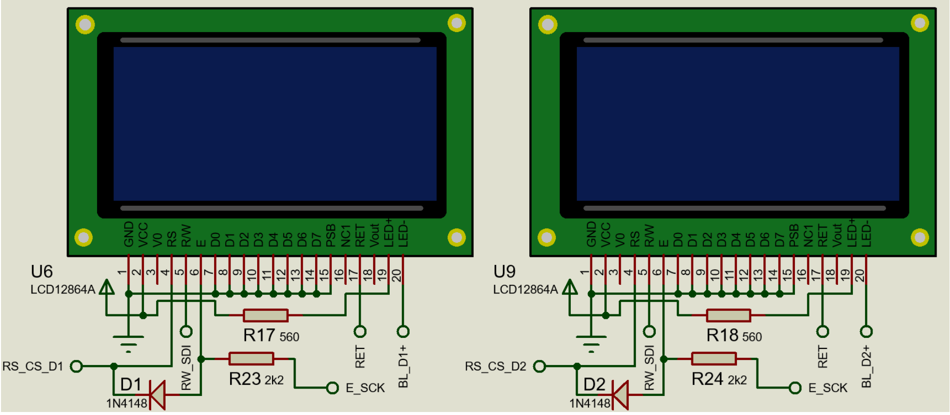
Hola @**martinmarcos** ...

yo tuve el mismo problema y buscando una solución encontré tu posteo.

Lo importante que tengo para contarte es que pude solucionarlo muy fácil con un par de diodos 1N4148 y un par de resistencias de 2k2.

Mi situación es la siguiente. Tengo dos de estos display alimentados a 5V y conectados a un PIC18F26K20 que se alimenta con 3.3V.

Particularmente en mi circuito lo solucione como indica la imagen siguiente...



Lo implemente en mi circuito y funciona OK con un SPI (BAUD = 530KHz).

La cuestión es que cuando el CS esta en alto, el diodo esta polarizado en inversa no conduce corriente y el display recibe el CLK. Cuando el CS esta en bajo, el dio se polariza en directa y gracias a la presencia de la resistencia el display deshabilitado no recibe el CLK pero si lo hace el que esta habilitado. ¡¡ Muy Fácil :D !!

Saludos y espero te sirva.

👍 2