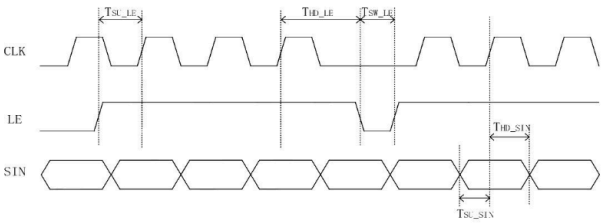
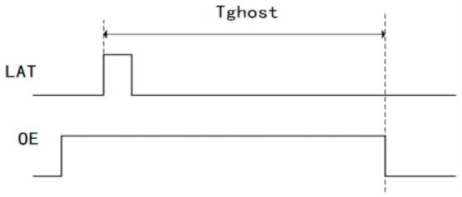
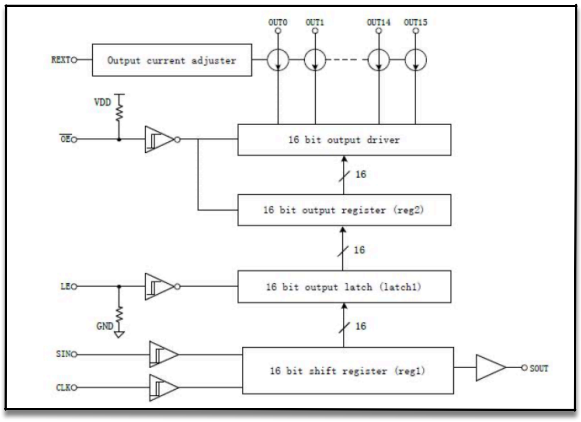
FM6126A Timing Information





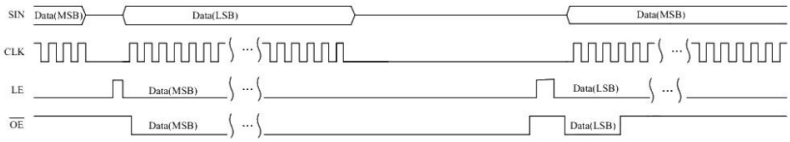


Note 1: When the LE pin is set to L, the latch circuit retains data; when the LE pin is H, the latch circuit does not retain data, and the data it ouput directly

When the OE (GCLK) pin is L, the OUT0-OUT15 pins will turn ON and OFF to respond to the data; set the OE (GCLK) pin to H, regardless of the data all output pins ..

The pin will be oFF when the OE(GCLK) pin i L, data can be transmitted and latched.

FM6126A double latch to imrpoved the principle of refresh rate

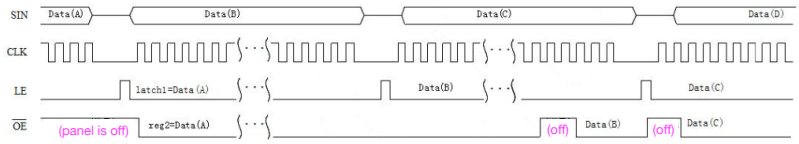


Reasons for low utilization rate of data transmission and data display of general constant current chip:

(1) When display a high-level data, the data display time may be much longer then the data transmission time, and the data cannot be counted during the extra time of the data display. According to transmission.

(2) When displaying low-level data, the data display time may be much shorter than the data transmission time, and the data cannot be counted during the extra time of data transmission. According to the show.

FM6126A data transmission and data display timing



FM6126A data transmission and data display timing is shown in the figure above, data(A) and data(C) are high-order data, data(B) and data(D) are high-low? data. Will show…

The high and low bits of the display data are combined according to time, so that the excess time fo the display hight data can be used for data transmission, or use the time of data transmission to …

For hight-level display, the perfect combination of data transmisssion and display can effectively improve the display refresh rate. The basic steaps are as follows:

(1) When the data(A) transfer is completed, a latch signal is generated on LE to latch data(A)

(2) After the data(A) is latched, OE changes from 1 to 0, to register data(a) and display data(A)

(3) While displaying data(A), transfer data(B)

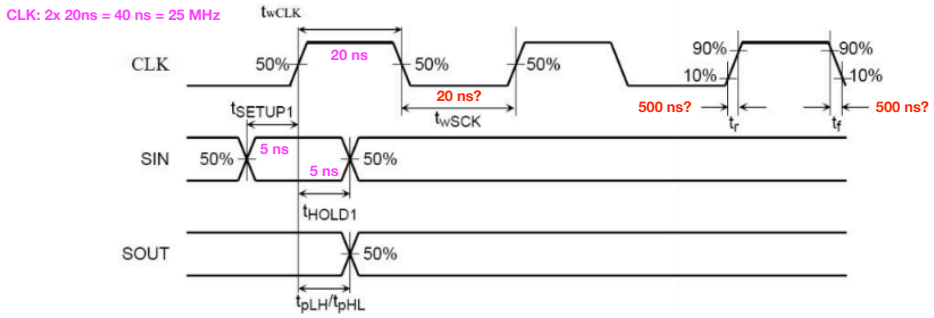
(4) After the data(B) transmission is completed, the latch signal LE is generated, data(B) is latched, and then data(C) is transmitted.

(5) After completing the display of data(A), register data(B) and display data(B)

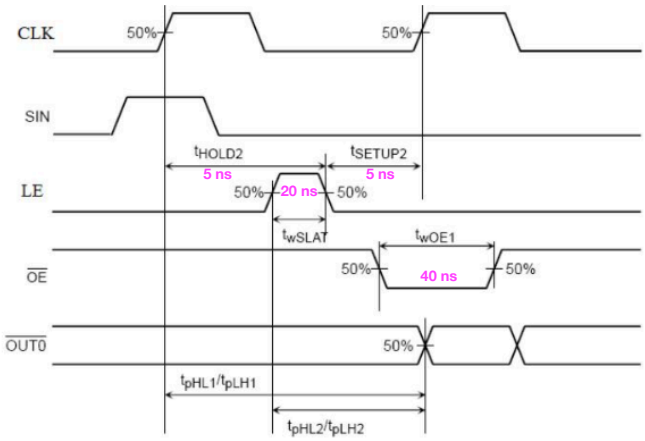
FM6126A - 16 channel double buffer constant current output LED driver chip

FM6126A Timing Information

1. CLK, SIN, SOUT



2. CLK, SIN, LE, $\overline{OE}$, $\overline{OUT0}$



Characteristic	Symbol	Test Circuit	Test Conditions	Minimum	Typical	Maximum	Unit
Serial Data Freq.	Fclk	6				30	MHz
Clock pulse width	TwCLK	6	SCK=H or L	20			ns
Latch pulse width	TwLE	6	LE=H	20			ns
Enable pulse width	TwOE	6	OE=H or L, Rext=890 Ohm	40			ns
Hold Time	THld1	6		5			ns
	THld2	6		5			ns
Setup Time	TSup1	6		5			ns
	TSup2	6		5			ns
Max clock rise time	Tr	6				500	ns
Max clock fall time	Tf	6				500	ns

FM6126A Configuration Information

```
' write the required init sequence to our panel drivers

' REF: https://www.gitmemory.com/issue/mrfaptastic/ESP32-RGB64x32MatrixPanel-I2S-DMA/23/665659409
' 64 & 128 do not appear to be different?? - 64: 1x panel, 128: 2x panels, etc.

' Control Register 1(12) - LE w/11 clocks
' Control Register 2(13) - LE w/12 clocks

' All FM6126A code is based on the excellent guesswork by shades66 in https://github.com/hzeller/rpi-rgb-led-matrix/issues/746

' Register 12 - brightness/gain settings, three 6bit values, aaaaaabbbbbbccccccc a= darkness?
'               seems to add red to the background when the leds are off, b=main brightness c=finer brightness
'               (i'm not sure if b & c are actually as 12 bit value but with b set to all 1's the value in c doesn't seem to make much difference)

' Register 13 - not sure what it's doing yet, just that 1 specific bit within seems to be an overall enable function.

' Now set all the values at the top to the same value for each of register 12/13 to get the same settings across the panel, the
'   current code loads different settings into each 32 columns.

' clocking in the register is simply clocking in the value (i've 2 panels so 128bits of data) and for the last 12/13 bits depending
'   on the register setting the latch to high. the final drop of latch to low clocks in the configuration. this is done by sending
'   the same value to r1/r2/g1/g2/b1/b2 at the same time to load the config into all the FM6126 chips

' Some necessary magic bit fields
' b12  - 1  adds red tinge
' b12  - 9/8/7/6/5  =  4 bit brightness
' b13  - 9   =1 screen on
' b13  - 6   =1 screen off

' NOTES from code:
'               0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
'               0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
'  int C12[16] = {0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}; // HIGH *NOTE* from https://github.com/2dom/PxMatrix/blob/master/PxMatrix.h#L372
'  int C12[16] = {0,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1}; // LOW
'  int C13[16] = {0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0};
```