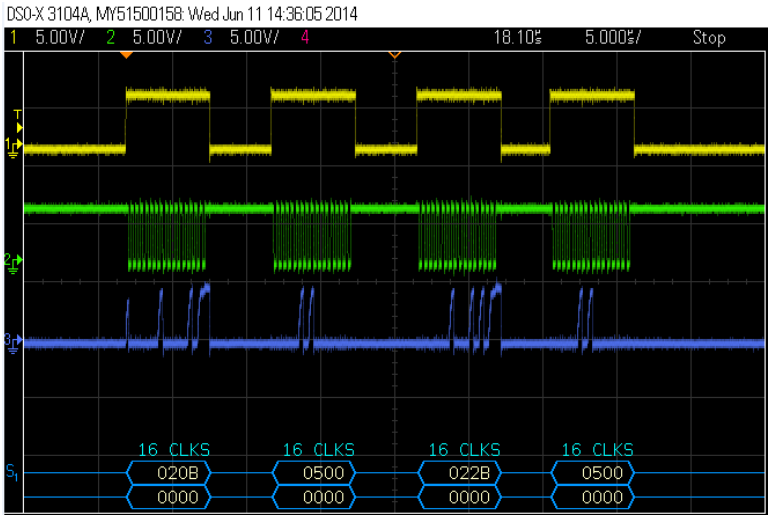
Code snippet.

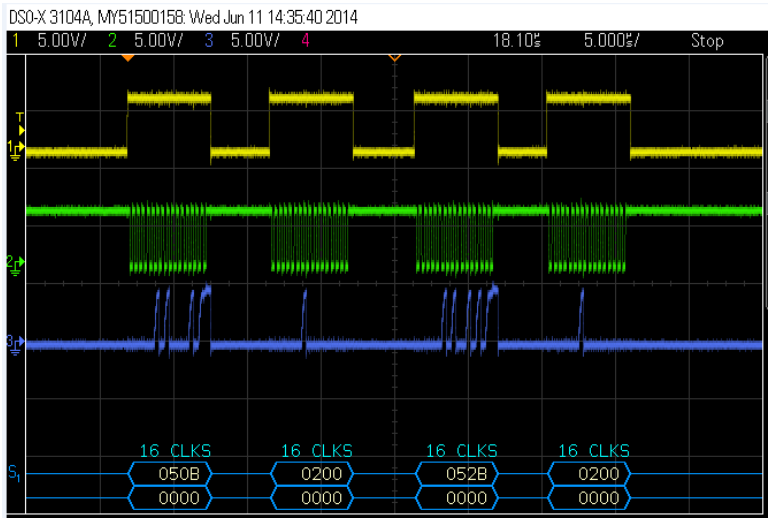
```
SPIM_2_ClearTxBuffer();
SPIM_2_ClearRxBuffer();
CyDelayUs(1u);
SPIM_2_TxEnable(); // also do this in ldriver routine
CyDelayUs(1u);
ldriver (0x020b, 0x0500); // CR2 to pg0, 0-05 W1HL write high order bits land
ldriver (0x022b, 0x0500); // CR2 to pg2, 2-05 W1LL write low order 2bits land don't really need right now
```

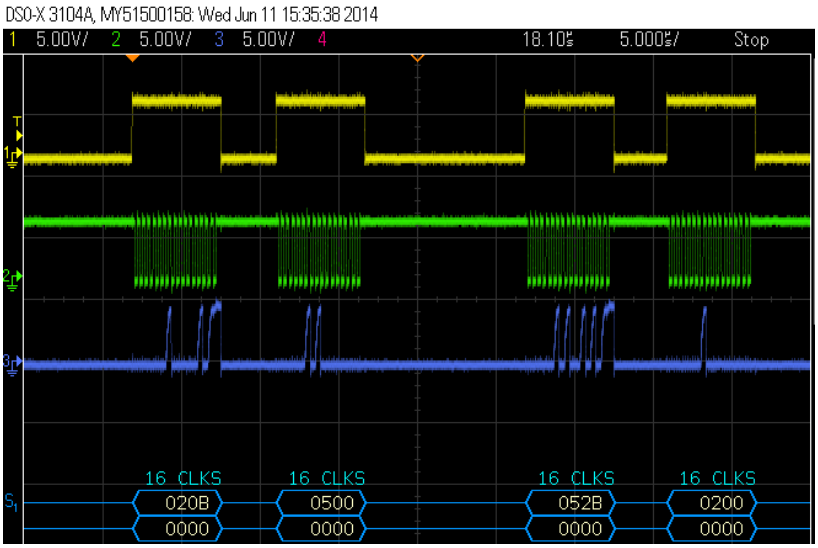
```
int ldriver (uint16 fword, uint16 sword)
    {
    SPIM_2_WriteTxData(fword); // goto pg#
    CyDelayUs(6u);  // need 2.7usec min
    SPIM_2_WriteTxData(sword); // set data
    CyDelayUs(6u);  // need 2.7usec min
    }
```

Correct zeroing of current.



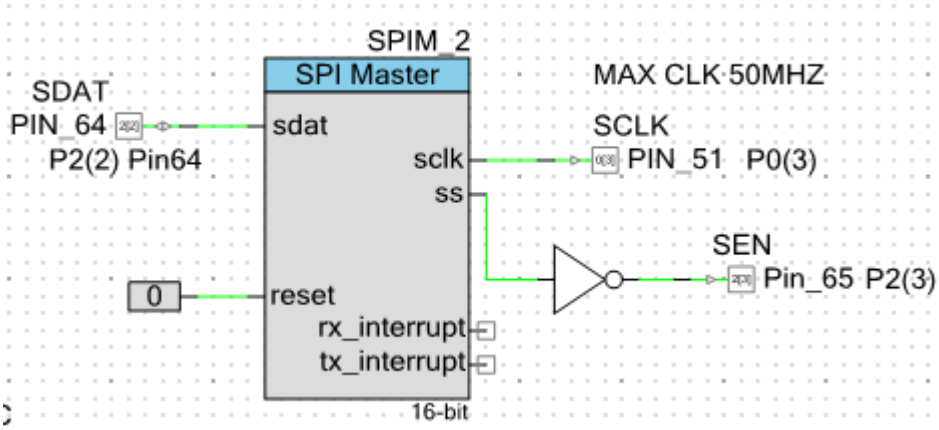
Bad display of zeroing current.





First case first nibble of 1st byte gets swapped with 1st nibble of 2nd byte, both sets.

Second case similar but in second set.



Top trace is SEN, followed by SCLK, then SDAT.

Thursday thought had fixed it by adding clear tx & rx buffer at end of txmt routine.

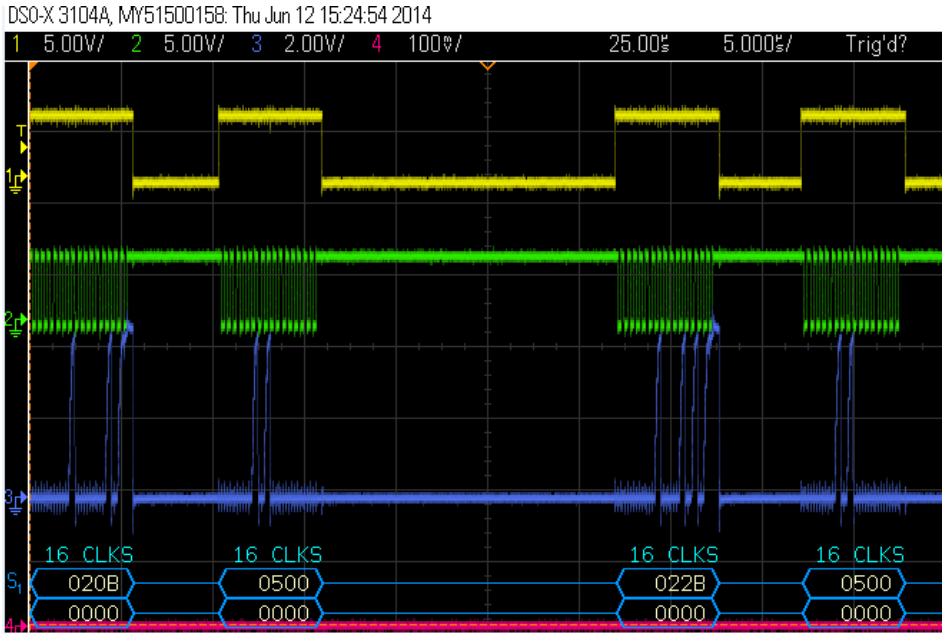
```
int ldriver (uint16 fword, uint16 sword)
        {
        SPIM_2_WriteTxData(fword); // goto pg#
        CyDelayUs(7u);  // need 2.7usec min
        SPIM_2_WriteTxData(sword); // set data
        CyDelayUs(7u);  // need 2.7usec min
        SPIM_2_ClearTxBuffer(); SPIM_2_ClearRxBuffer();
        }
```

Worked fine all morning. Added more routines, then started happening again.

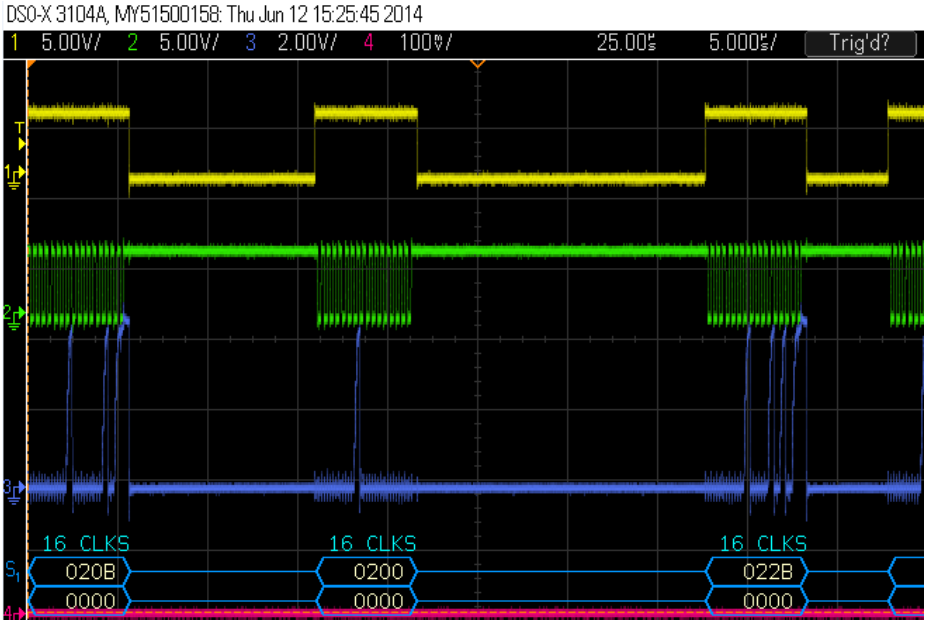
This time the command, instead of 020B, 0500, get 020B, 0200.

No longer swapping, but not updating a nibble.

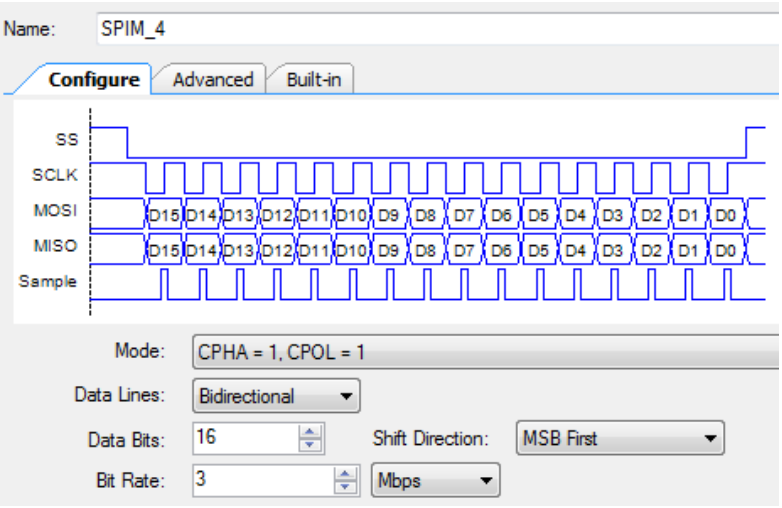
Below correct.

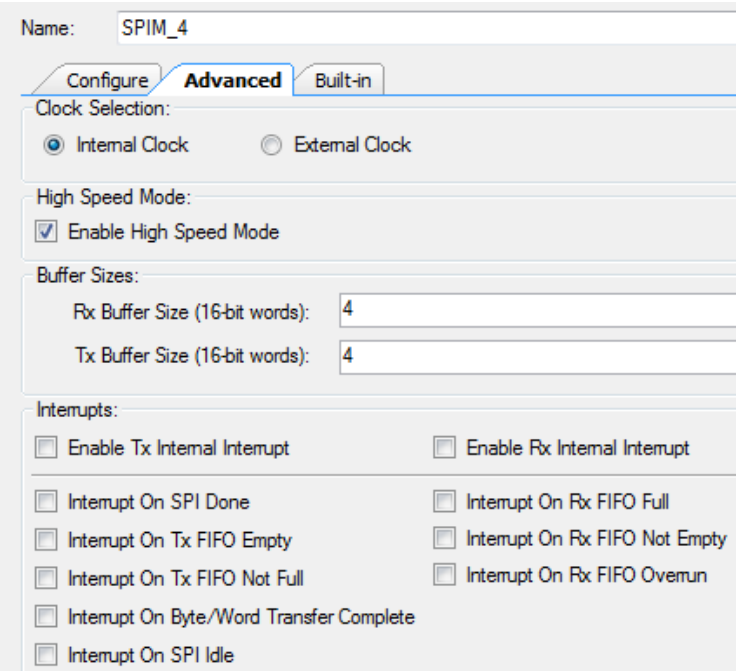


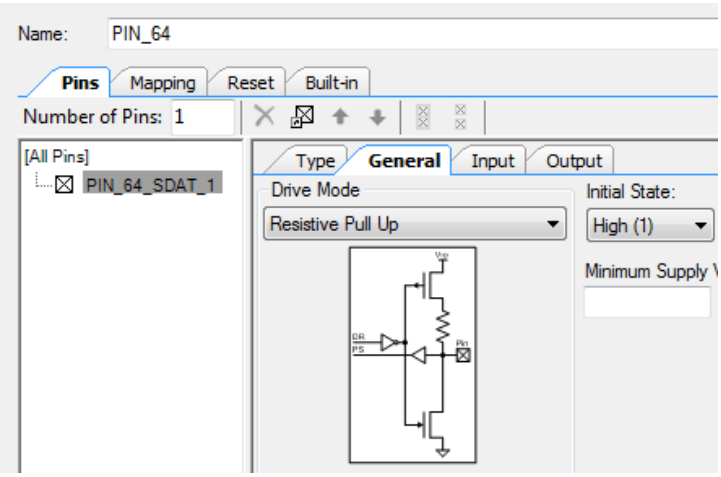
Then incorrect.



The bidirectional pin is set for resistive pull up.

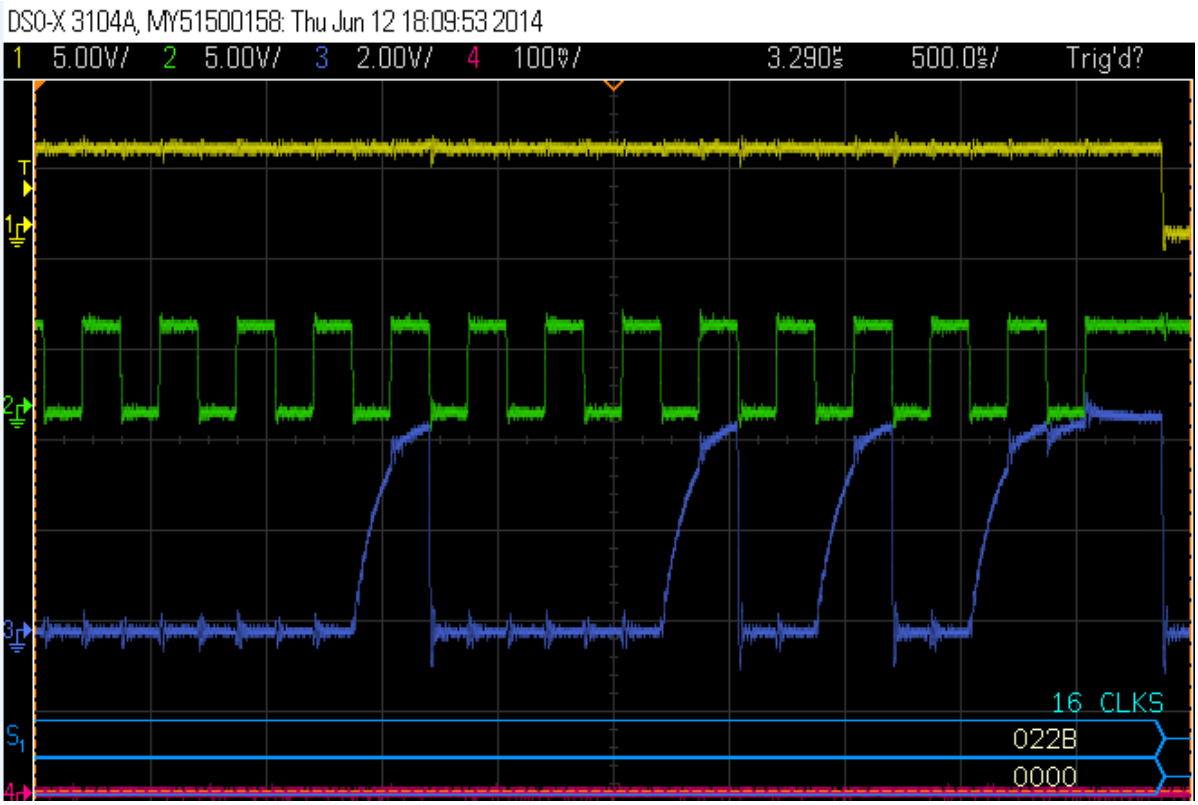






As far as rise time of SDAT

With slave connected:



With slave not connected, nothing else on SDAT except a 10k pullup. As below:

The slave datasheet says input currents of 15ua, and when outputs, up to 5ma.