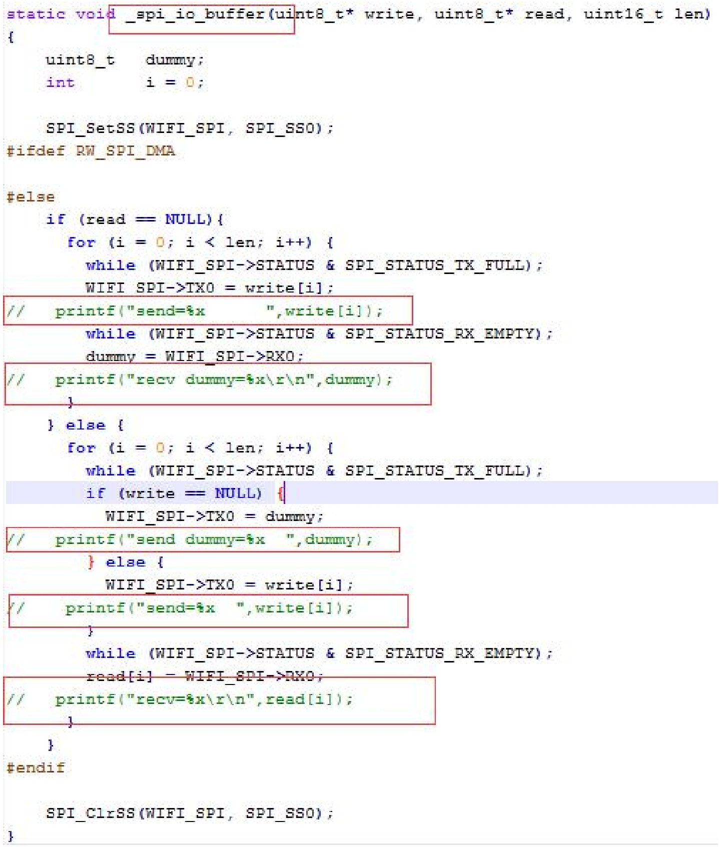
ETDX160127135

RAK439 SPI Interface Debug

1. View the SPI log

When RAK439 did not pass the initialize, you can open the SPI log output in SDK code and check what happened.
NOS modify _spi_io_buffer function at.\platform\ rw_lib_platform.c，
OS modify _spi_io_buffer function at .\platform\ rw_lib_platform_os.c.



```
static void _spi_io_buffer(uint8_t* write, uint8_t* read, uint16_t len)
{
    uint8_t   dummy;
    int       i = 0;

    SPI_SetSS(WIFI_SPI, SPI_SS0);
#ifdef RW_SPI_DMA

#else
    if (read == NULL){
      for (i = 0; i < len; i++) {
        while (WIFI_SPI->STATUS & SPI_STATUS_TX_FULL);
        WIFI_SPI->TX0 = write[i];
//    printf("send=%x       ",write[i]);
        while (WIFI_SPI->STATUS & SPI_STATUS_RX_EMPTY);
        dummy = WIFI_SPI->RX0;
//    printf("recv dummy=%x\r\n",dummy);
      }
    } else {
      for (i = 0; i < len; i++) {
        while (WIFI_SPI->STATUS & SPI_STATUS_TX_FULL);
        if (write == NULL) {
          WIFI_SPI->TX0 = dummy;
//    printf("send dummy=%x  ",dummy);
        } else {
          WIFI_SPI->TX0 = write[i];
//     printf("send=%x  ",write[i]);
        }
        while (WIFI_SPI->STATUS & SPI_STATUS_RX_EMPTY);
        read[i] = WIFI_SPI->RX0;
//    printf("recv=%x\r\n",read[i]);
      }
    }
#endif

    SPI_ClrSS(WIFI_SPI, SPI_SS0);
}
```

When call rw_sysDriverInit function, the start of the spi data：

```
send=44 recv=b4

send=0 recv=b4

send=0 recv=b4

send=80 recv=b4

send=c2 recv=b4

send=0 recv=b4

send=0 recv=c

send=0 recv=5b
```

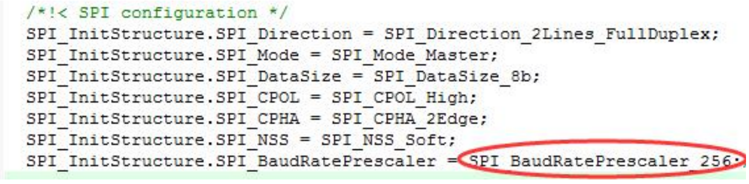
receive 5b presents spi interface is working properly, a poor contact of spi interface or power supply shortage will make the mcu can not receive 5b.

The front 7 bytes that different modules received may be different, but a same module received the 7 bytes is same every time.

2. View the waveform using oscilloscope

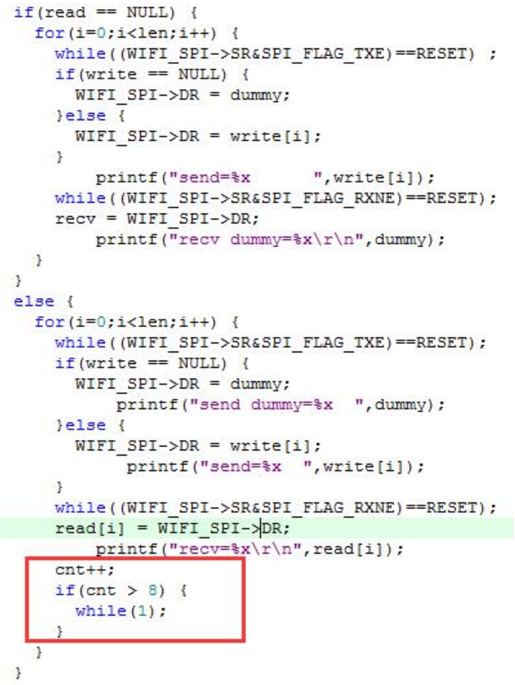
When call rw_sysDriverInit function, if the SPI data received not correct(the eighth is not 0x5b), may be the SPI configuration is not correct, you can view the waveform using a oscilloscope.

1. Modify code to reduce the SPI CLK, MCU is STM32F411(using SPI1, clk souce is APB2)
   APB2=SYSCLK(96MHz), SPI CLK=APB2/256=384KHz，easy to catch waveform.



```
/*!< SPI configuration */
SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStructure.SPI_CPOL = SPI_CPOL_High;
SPI_InitStructure.SPI_CPHA = SPI_CPHA_2Edge;
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_256;
```

2. When SPI has received 8 bytes, enter the while(1) loop, for easy to catch waveform.



```
if(read == NULL) {
  for(i=0;i<len;i++) {
    while((WIFI_SPI->SR&SPI_FLAG_TXE)==RESET) ;
    if(write == NULL) {
      WIFI_SPI->DR = dummy;
    }else {
      WIFI_SPI->DR = write[i];
    }
        printf("send=%x      ",write[i]);
    while((WIFI_SPI->SR&SPI_FLAG_RXNE)==RESET);
    recv = WIFI_SPI->DR;
        printf("recv dummy=%x\r\n",dummy);
  }
}
else {
  for(i=0;i<len;i++) {
    while((WIFI_SPI->SR&SPI_FLAG_TXE)==RESET);
    if(write == NULL) {
      WIFI_SPI->DR = dummy;
          printf("send dummy=%x  ",dummy);
    }else {
      WIFI_SPI->DR = write[i];
           printf("send=%x  ",write[i]);
    }
    while((WIFI_SPI->SR&SPI_FLAG_RXNE)==RESET);
    read[i] = WIFI_SPI->DR;
        printf("recv=%x\r\n",read[i]);
    cnt++;
    if(cnt > 8) {
      while(1);
    }
  }
}
```

3. UART print below when reset.

```
tcpudp_test.c:68  Host platform init...success
send=44  recv=d6
send=0  recv=d6
send=0  recv=d6
send=80  recv=d6
send=c2  recv=d6
send=0  recv=d6
send=0  recv=c
send=0  recv=5b
```

4. Oscilloscope setting: single catch，edge-triggered，trigger source select CLK，falling edge trigger.

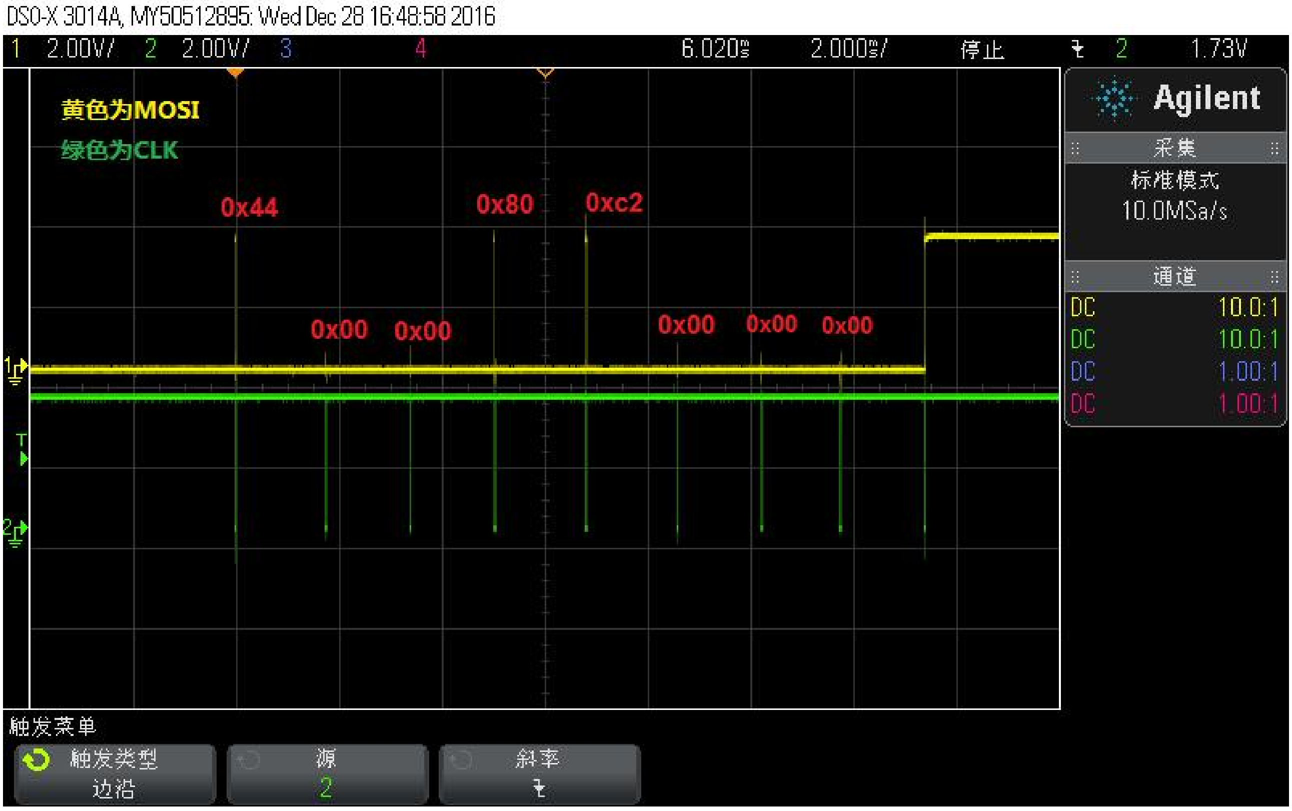


图 1

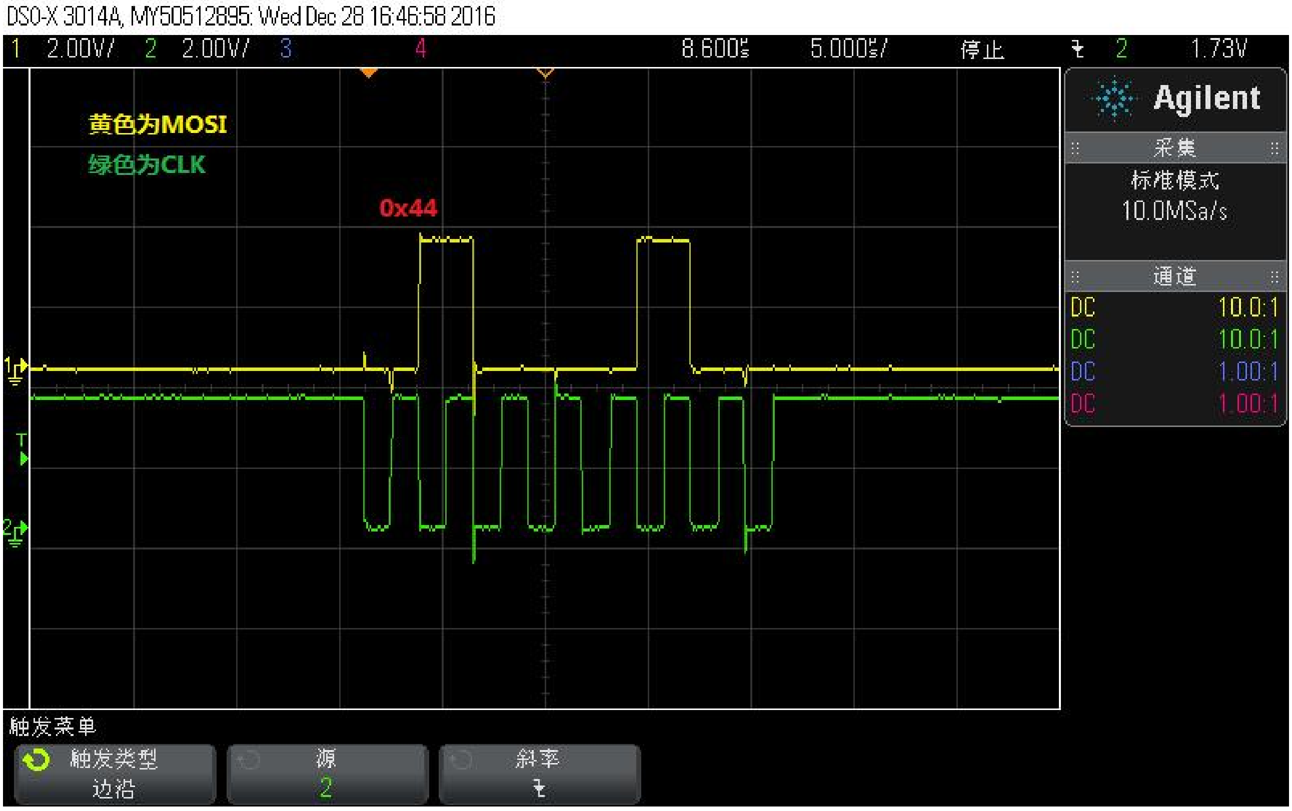


图 2

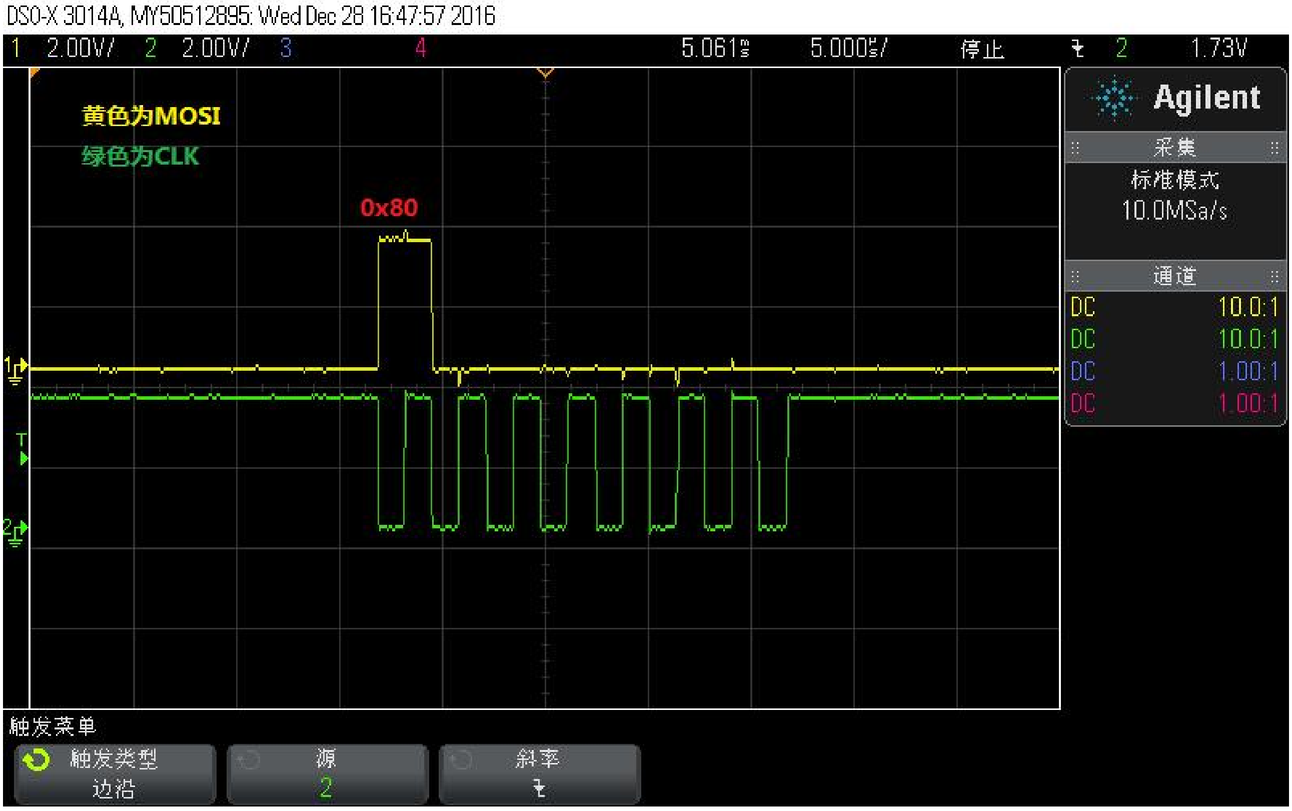

DSO-X 3014A, MY50512895: Wed Dec 28 16:47:57 2016
1 2.00V/ 2 2.00V/ 3 4 5.061us 5.000us/ 停止 ⮧ 2 1.73V
黄色为MOSI
绿色为CLK
0x80
Agilent
采集
标准模式
10.0MSa/s
通道
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1
触发菜单
触发类型
边沿
源
2
斜率

图 3

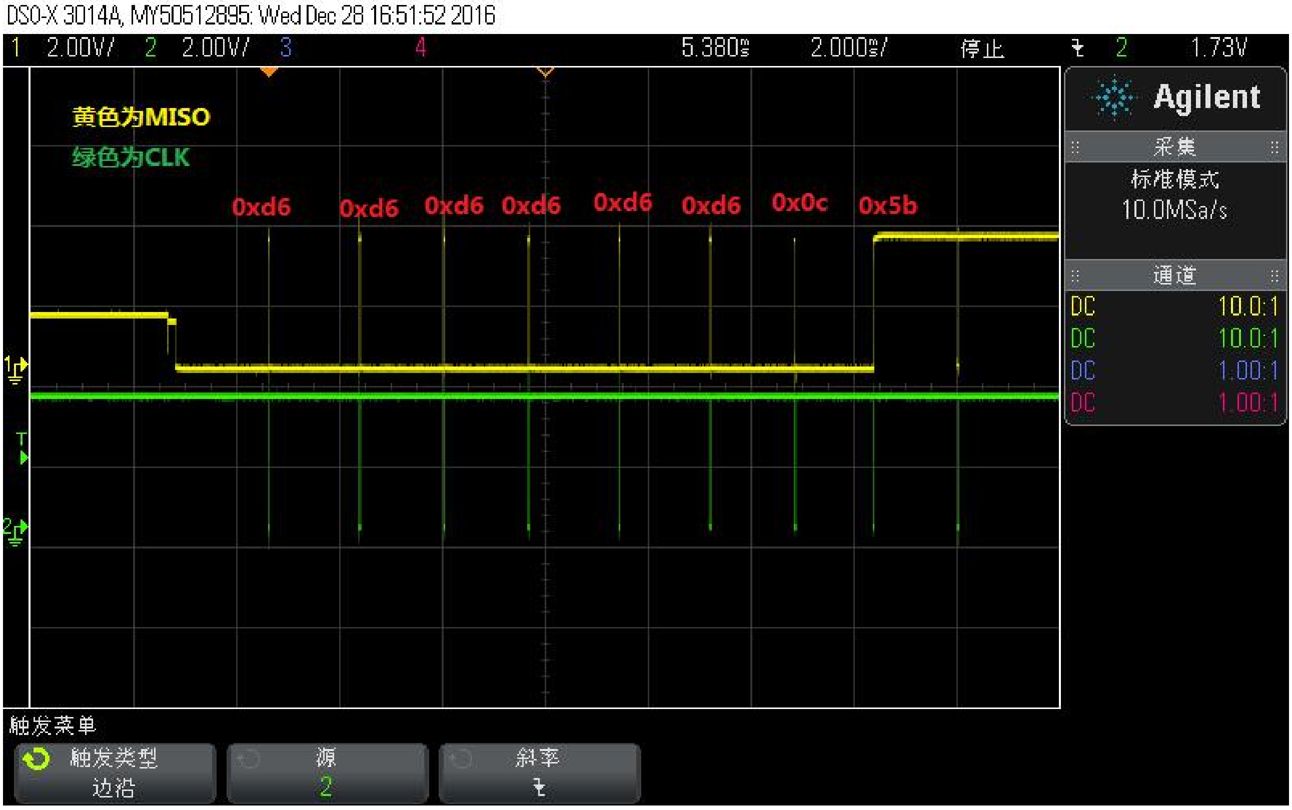

DSO-X 3014A, MY50512895: Wed Dec 28 16:51:52 2016
1 2.00V/ 2 2.00V/ 3 4 5.380us 2.000us/ 停止 ⮧ 2 1.73V
黄色为MISO
绿色为CLK
0xd6
0xd6
0xd6
0xd6
0xd6
0xd6
0x0c
0x5b
Agilent
采集
标准模式
10.0MSa/s
通道
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1
触发菜单
触发类型
边沿
源
2
斜率

图 4

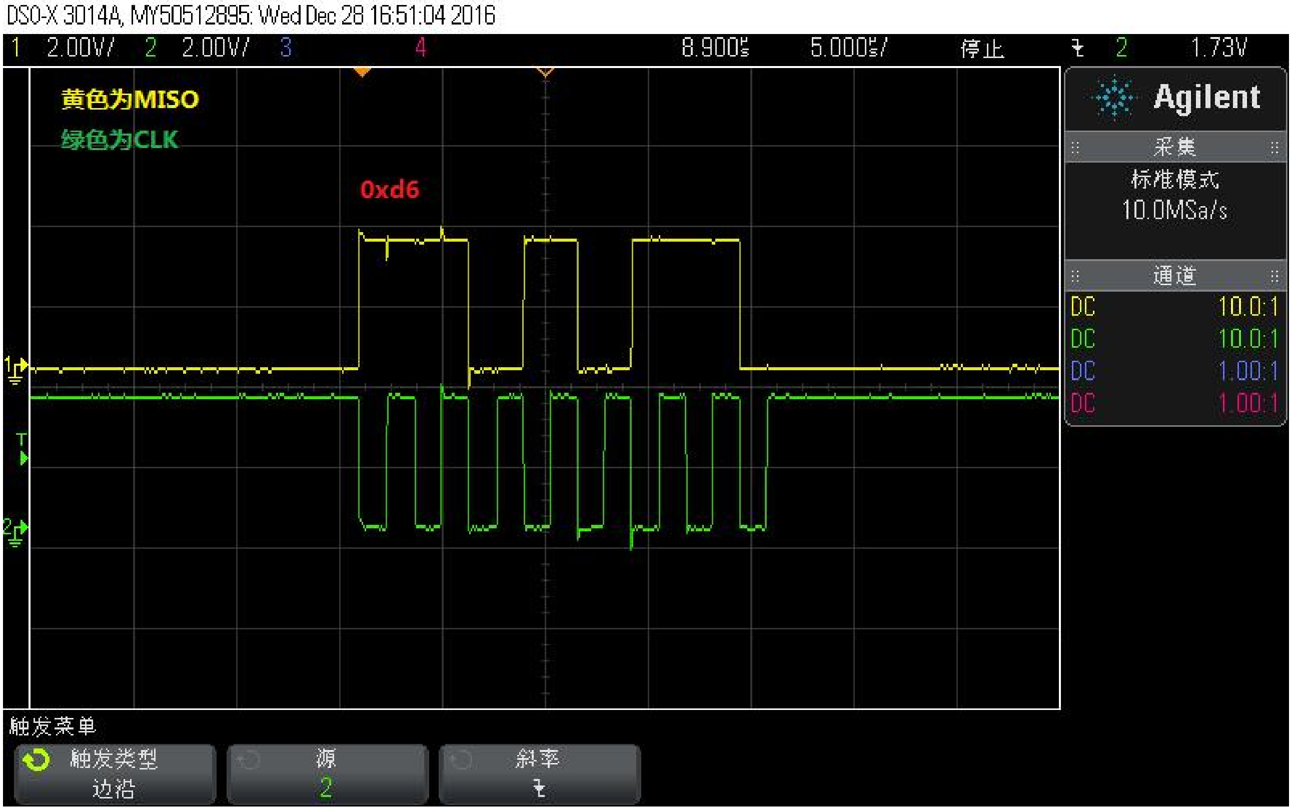

DSO-X 3014A, MY50512895: Wed Dec 28 16:51:04 2016
1 2.00V/ 2 2.00V/ 3 4 8.900us 5.000us/ 停止 ⧹ 2 1.73V
黄色为MISO
绿色为CLK
0xd6
Agilent
采集
标准模式
10.0MSa/s
通道
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1
触发菜单
触发类型
边沿
源
2
斜率

图 5

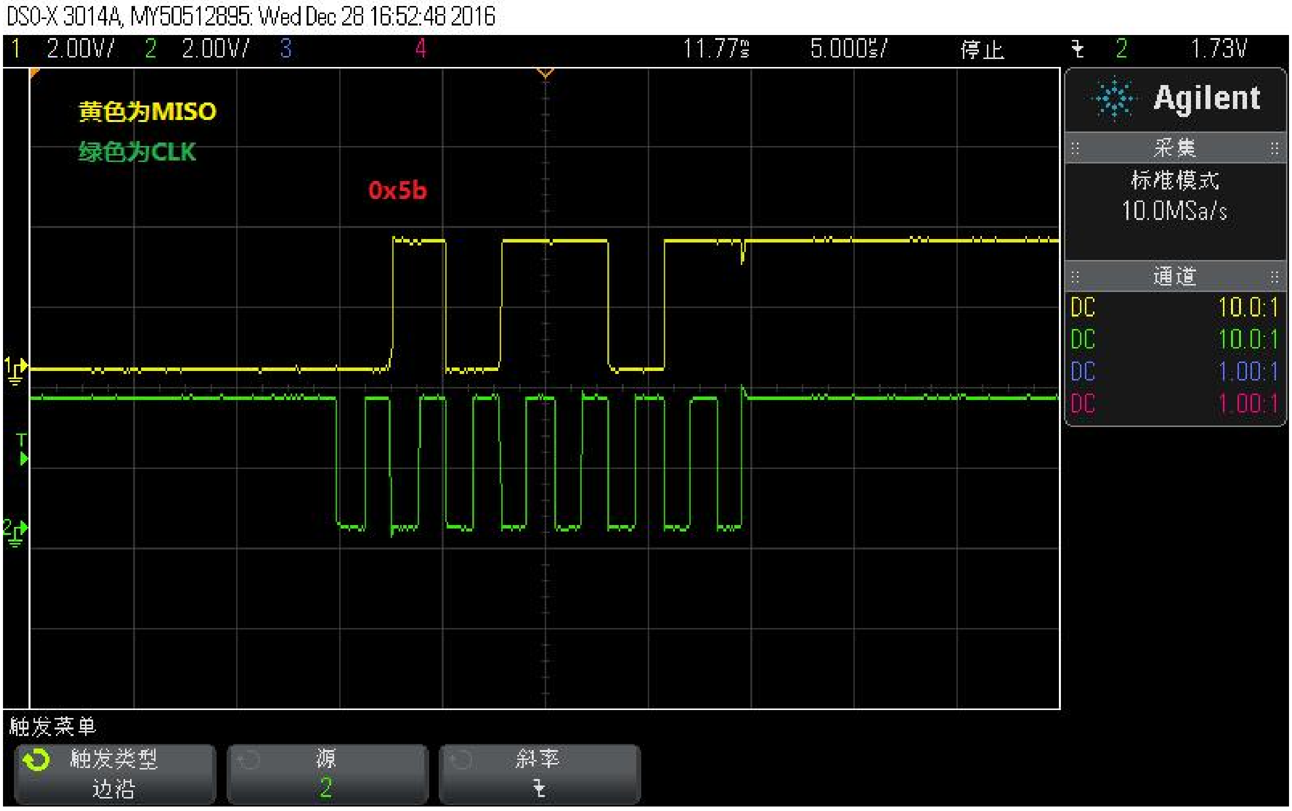

DSO-X 3014A, MY50512895: Wed Dec 28 16:52:48 2016
1 2.00V/ 2 2.00V/ 3 4 11.77us 5.000us/ 停止 ⧹ 2 1.73V
黄色为MISO
绿色为CLK
0x5b
Agilent
采集
标准模式
10.0MSa/s
通道
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1
触发菜单
触发类型
边沿
源
2
斜率

图 6

3. Change log

version	author	date	Change
V1.0	harry	2016/12/28	Create document