# 颜色识别通信协议

## 硬件连接

==============================================================================================

FUNCTION LOGIC FPGA PIN NET/ARDUINO reserved

==============================================================================================

SPI\_\_ spi\_clk H13 FPGA\_AR\_SCK <- AR\_SCK

|\_ spi\_in M5 FPGA\_AR\_MOSI <- AR\_MOSI

\\_ spi\_out L5 FPGA\_AR\_MISO <- AR\_MISO

spi\_fss B2 AR\_D10 /SS

rst\_n A2 AR\_D9

int1 B3 AR\_D8

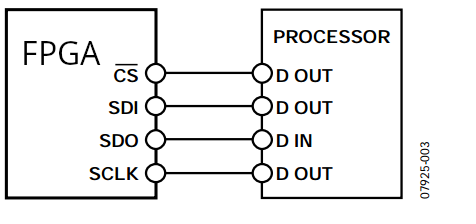
int1 需要可设置，支持电平或边沿中断

FPGA和Arduino 通过SPI进行通信，通信的硬件如上。

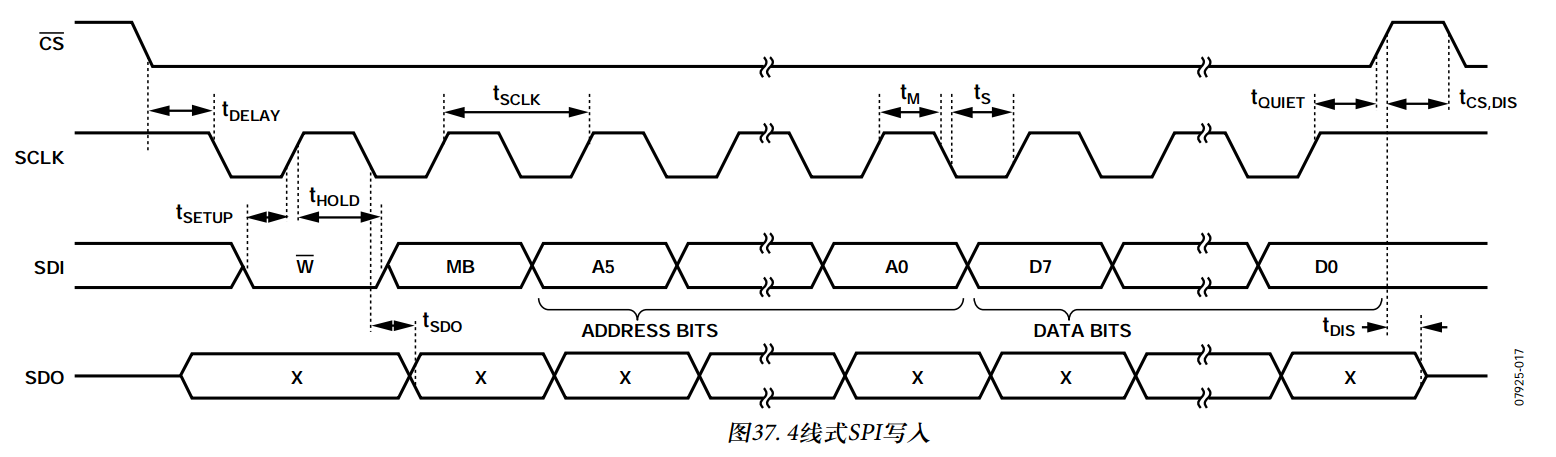
## SPI串行通信

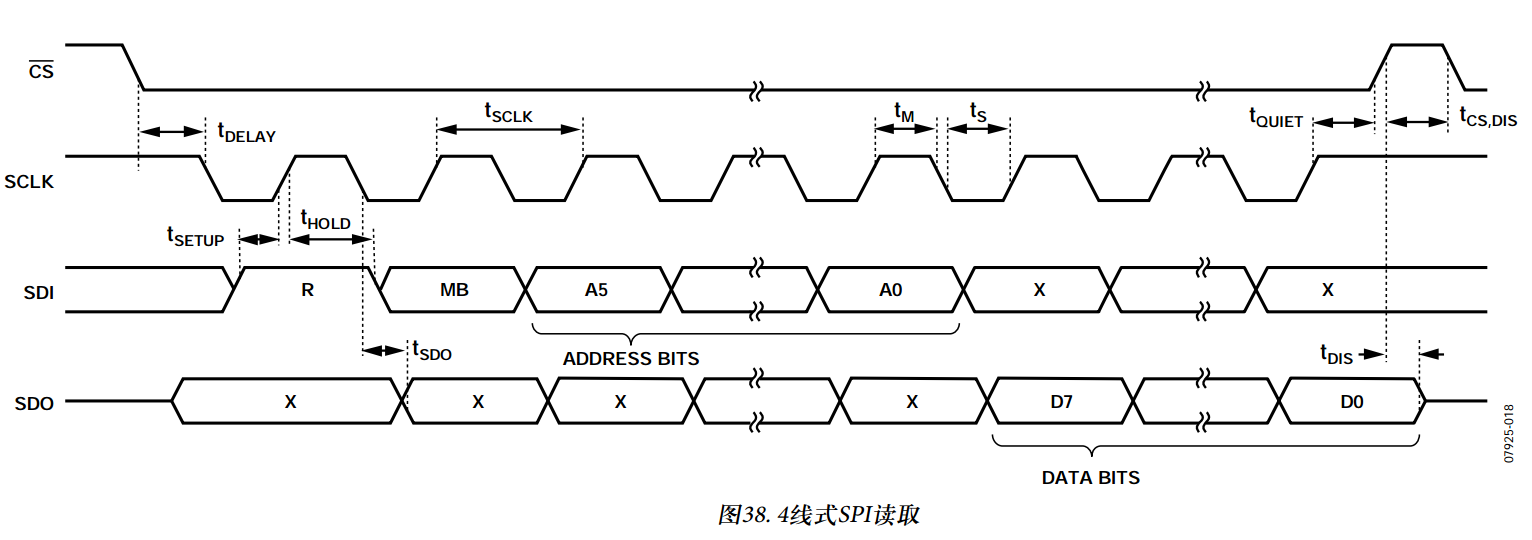
对于SPI，使用4线配置，下图所示。最大负载为100 pF时，最大SPI时钟速度为50 MHz，时序方案按照时钟极性(CPOL)= 1、时钟相位(CPHA)= 1执行。CS为串行端口使能线，由SPI主机控制。如图所示，此线必须在传输起点变为低电平，传输终点变为高电平。 SCLK

为串行端口时钟，由SPI主机提供。无传输期间， SCLK为空闲高电平状态。 SDI和SDO分别为串行数据输入和输出。 SCLK下降沿时数据更新， SCLK上升沿时进行采样。

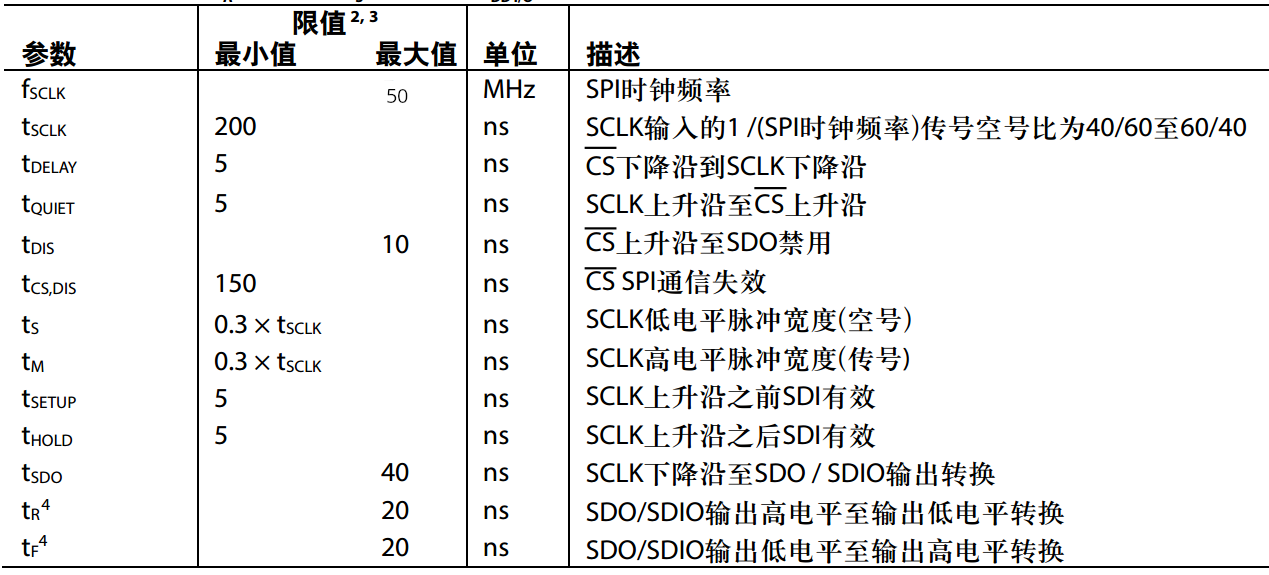


要在单次传输内读取或写入多个字节，必须设置位于第一个字节传输(MB，下图)R /W位后的多字节位。寄存器寻址和数据的第一个字节后，时钟脉冲的随后每次设置(8个时钟脉冲)导致FPGA指向下一个寄存器的读取/写入。时钟脉冲停止后，移位才随之中止， CS失效。要执行不同不连续寄存器的读取或写入，传输之间CS必须失效，新寄存器另行处理





SPI时序：



## 中断

FPGA为驱动中断提供输出引脚： INT1这个中断引脚都是推挽低阻抗引脚。中断引脚默认配置为高电平有效。设置SYS\_CON 寄存器(地址0x00)中的D6位，可以更改为低电平有效。所有功能都可以同时使用，但是，一些功能可能需要共享中断引脚。设置SYS\_CON寄存器(D2)的适当位，中断使能。

## Reset（可选）

FPGA为驱动提供Reset引脚，Reset设置为低电平有效。当MCU 复位FPGA时所有的设置信息恢复到默认值。

## 寄存器

所有的寄存器都为8位

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 地址 | 名称 | 类型 | 描述 |
| 0x00 | SYS\_CON | R/W | D0:系统使能，在标定颜色完毕后开始识别颜色  D1:Reset，所有寄存器恢复到默认值0  D2:中断使能，使能中断，中断引脚开始工作  D3:中断位1，每次坐标改变时，FPGA中断位置1，需要写0清除  D4:中断位2，每次颜色消失时，FPGA中断位置1，需要写0清除  D5:中断位3，每次颜色出现时，FPGA中断位置1，需要写0清除  D6: 高电平中断1，低电平中断0  D7: reserve |
| 0x01 | AREA\_SIZE | R/W | 识别颜色的面积（0-255），在标定颜色时指定标定颜色的面积，在识别颜色时反馈当前识别颜色面积的大小 |
| 0x02 | X\_DATA\_LOW | R/W | X轴坐标低8位，在标定颜色时指定x轴低8位的坐标处的颜色，在识别颜色时反馈当前识别x轴低8位的坐标 |
| 0x03 | X\_DATA\_HIGH | R/W | X轴坐标高8位，在标定颜色时指定x轴高8位的坐标处的颜色，在识别颜色时反馈当前识别x轴高8位的坐标 |
| 0x04 | Y\_DATA\_LOW | R/W | Y轴坐标低8位，在标定颜色时指定y轴低8位的坐标处的颜色，在识别颜色时反馈当前识别x轴低8位的坐标 |
| 0x05 | Y\_DATA\_HIGH | R/W | Y轴坐标高8位，在标定颜色时指定y轴高8位的坐标处的颜色，在识别颜色时反馈当前识别y轴高8位的坐标 |
| 0x06 | APPROX\_VAL | R/W | 识别颜色的近似度，在标定颜色时指定标定颜色的近似度，在识别颜色时反馈当前识别颜色的近似度，0为完全相近，255为不相近。 |

## 总体流程如下

Reset

开始识别

标定颜色

设置系统

# FPGA具体实现（以下说明都是从FPGA角度出发）

（注：暂时完成进度，有待后续更新）

以8位为交互信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 收到的命令 | 名称 | 类型 | 描述 |
| 0x00 | SYS\_CON | 输入 | D0:系统使能，为0时，开始识别颜色，并且绿灯灭；为1时，识别结束标定物体，可以随物体移动，并且绿灯亮。  D1:Reset，为0时处于工作状态，并且红灯灭；为1时，清除之前选定颜色（无法识别颜色），并且红灯亮。 |
| 0x01 | AREA\_SIZE | 输入 | 识别框的边长（识别框为正方形） |
| 0x82 | X\_DATA\_LOW | 输出 | X轴坐标低8位数据 |
| 0x83 | X\_DATA\_HIGH | 输出 | X轴坐标高8位数据 |
| 0x84 | Y\_DATA\_LOW | 输出 | Y轴坐标低8位数据 |
| 0x85 | Y\_DATA\_HIGH | 输出 | Y轴坐标高8位数据 |
| 0x86 | APPROX\_VAL | 输出 | 识别颜色的近似度，在标定颜色时指定标定颜色的近似度，在识别颜色时反馈当前识别颜色的近似度，0为完全相近，255为不相近。 |