

# USB2SPI 适配器

## 用户手册

版本：2.0

发布日期：2024/08/10

文档编号：DM10001



深圳市安易博电子科技有限公司

# 目录

目录 .....	2
一、 简介 .....	4
二、 外观 .....	5
三、 安装 .....	7
3.1 安装 APP .....	7
3.2 安装驱动 .....	7
3.3 找到设备 .....	7
四、 使用 .....	9
4.1 第一次使用 .....	9
4.2 软件界面说明 .....	10
4.2.1 标题栏 .....	11
4.2.2 SPI 读区域 .....	12
4.2.2 SPI 写区域 .....	14
4.2.3 GPIO 配置和值设置区域 .....	15
4.2.4 Trig 配置区域 .....	16
4.2.5 SPI 适配器操作区域 .....	16
4.2.6 Logo 信息区域 .....	18
4.2.7 状态栏区域 .....	18
4.3 使用 SPI 功能 .....	20
4.3.1 SPI 作为主设备使用 .....	20
4.3.1.1 SPI 参数配置 .....	20
4.3.1.2 SPI Direct 和 buffer 读写方式的差别 .....	21
4.3.1.2 SPI 双工通讯和自检 .....	24

4.3.1.3 SPI 读操作 .....	26
4.3.1.4 SPI 写操作 .....	27
4.3.1.5 SPI 触发操作 .....	28
4.3.2 SPI 作为从设备使用 .....	29
4.3.2.1 进入从模式 .....	29
4.3.2.2 使用说明 .....	29
4.4 GPIO 使用 .....	29
4.4.1 使用说明 .....	29
4.4.2 操作步骤: .....	30
4.4.3 其他功能 .....	30
4.5 右键菜单操作 .....	30
4.5.1 内容框右键菜单操作 .....	30
4.5.1 log 信息框右键菜单操作 .....	32
4.6 固件更新 .....	32
4.7 软件版本信息和技术支持 .....	33
五、自主开发 .....	34
六、维护 .....	35

## 一、 简介

USB2SPI 适配器是一款 USB 转 SPI 转换板，附带 4 路可以配置 GPIO。USB 工作在高速模式，SPI 即可作为主设备，又可作为从设备，八种波特率可供选择，最高可达 30M。

### 系统特征：

- 1> 3.3V 输出电压。
- 2> 自恢复熔丝，防止过流。
- 3> USB2.0 高速通讯，平均数据传输率最快可达 28M BPS。
- 4> 提供库接口供第三方开发。
- 5> 唯一的设备序列号，可同时打开多个设备。
- 6> 支持 Hex 扩展名文件的解码，以及文本和 16 进制混合编辑模式。
- 7> 支持 win XP 以上系统(32bit,64bit)，支持 linux 系统。
- 8> 支持固件更新

### SPI 特征：

- 1. SPI 波特率最大可设置 30M，八种分频系数，分别是 2,4,8,16,32,64,128,256。
- 2. 极性和相位四种模式设置。
- 3. 支持 8bit、16bit 帧长度，LSB 和 MSB 发送顺序。
- 4. 两种读写模式，缓存模式和直接模式。
- 5. 三种读写方式，定时触发，IO 触发，手动执行。
- 6. 缓存模式单次最大读写长度 2M，直接模式 48K。
- 7. 工作在从模式时无间断接收。

### 电气特征：

- 1. 端口电压高电平在 2.7~3.6V，低电平在-0.3~0.3V。
- 2. 所有的数字端口最大驱动电流 25MA。
- 3. ESD 电压，HDM 2000V，CDM 500V。
- 4. 上拉下拉电阻大小 40K 左右。

## 二、 外观



端口说明见下表

端子号	功能说明	端子号	功能说明
GND	电源地	VDD	电源正，3.3V 输出，最大 500mA
IO1	可配置输入输出脚 1，上电前 IO1 拉低进入 DFU 模式	CSN	SPI 片选脚,主模式时，对于不需要片选的从设备，可以不接；从模式时，必须接主设备的 CS 脚
IO2	可配置输入输出脚 2，还作为 SPI 触发控制脚	SCLK	SPI 时钟脚
IO3	可配置输入输出脚 3	MOSI	SPI 主输出从输入脚
IO4	可配置输入输出脚 4	MISO	SPI 主输入从输出脚

## 三、 安装

下载软件包后，解压到指定文件夹

### 3.1 安装 app

运行安装目录下的文件 USB2SPI\_R3\_Setup.exe. 一路点击下一步到安装结束。

### 3.2 安装驱动

根据自己的电脑的操作系统选择 32bit 或 64bit 的驱动程序安装至结束。

32bit 系统，执行 M3F2xm\_DRV\_32bit\_R5\_Setup.exe 安装程序；

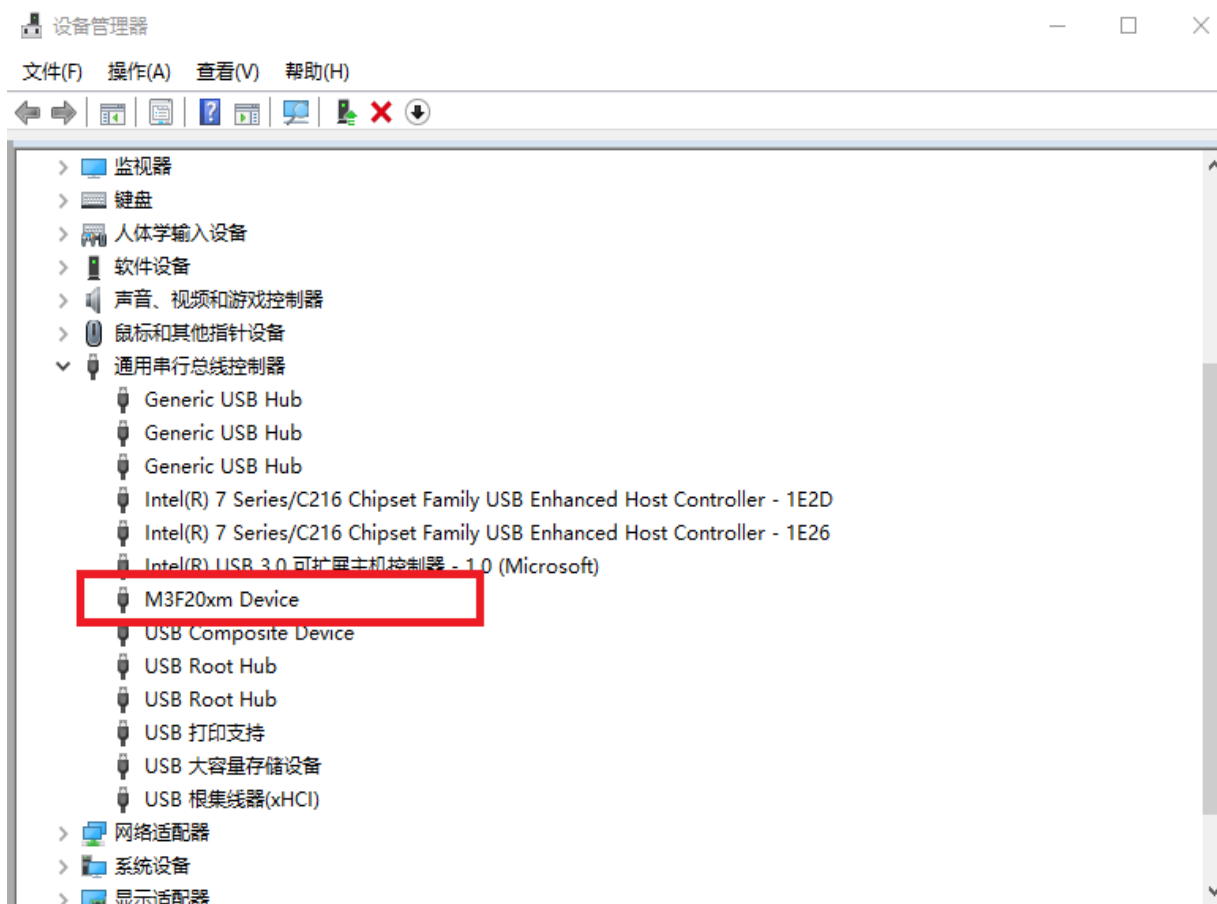
64bit 系统，执行 M3F2xm\_DRV\_64bit\_R5\_Setup.exe 安装程序；

注意一：安装驱动碰到数字签名认证时，一定要选择“信任”。

注意二：对于 win7 驱动程序安装，如果不是 win7 sp1，先更新到 SP1，如果已经是 SP1，则先安装 win7 补丁包，如果是 win7 32bit 系统，双击运行 Windows6.1-KB3033929-x86.msu，如果是 win7 64bit，双击运行 Windows6.1-KB3033929-x64.msu。

### 3.3 找到设备

插上 USB2SPI 适配器后，会自动寻找驱动，直到设备管理下通用串行总线控制器下出现一个 M3F20xm device 设备。如下图：



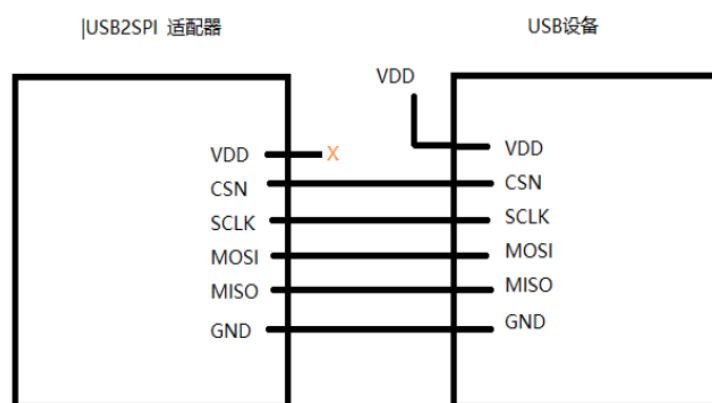


## 四、 使用

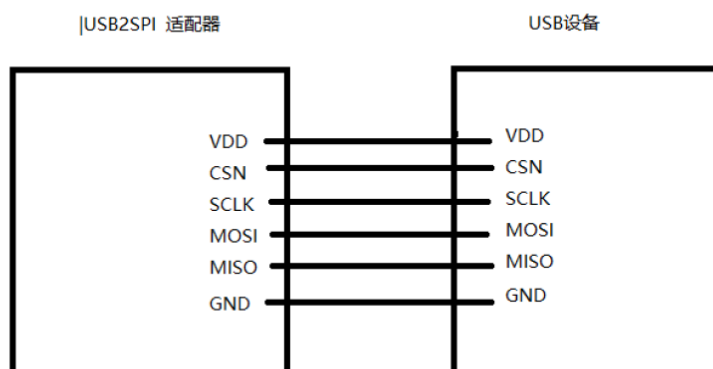
### 4.1 第一次使用

使用前，请先了解 SPI 的基本知识。接线如下图：

方式一：自供电

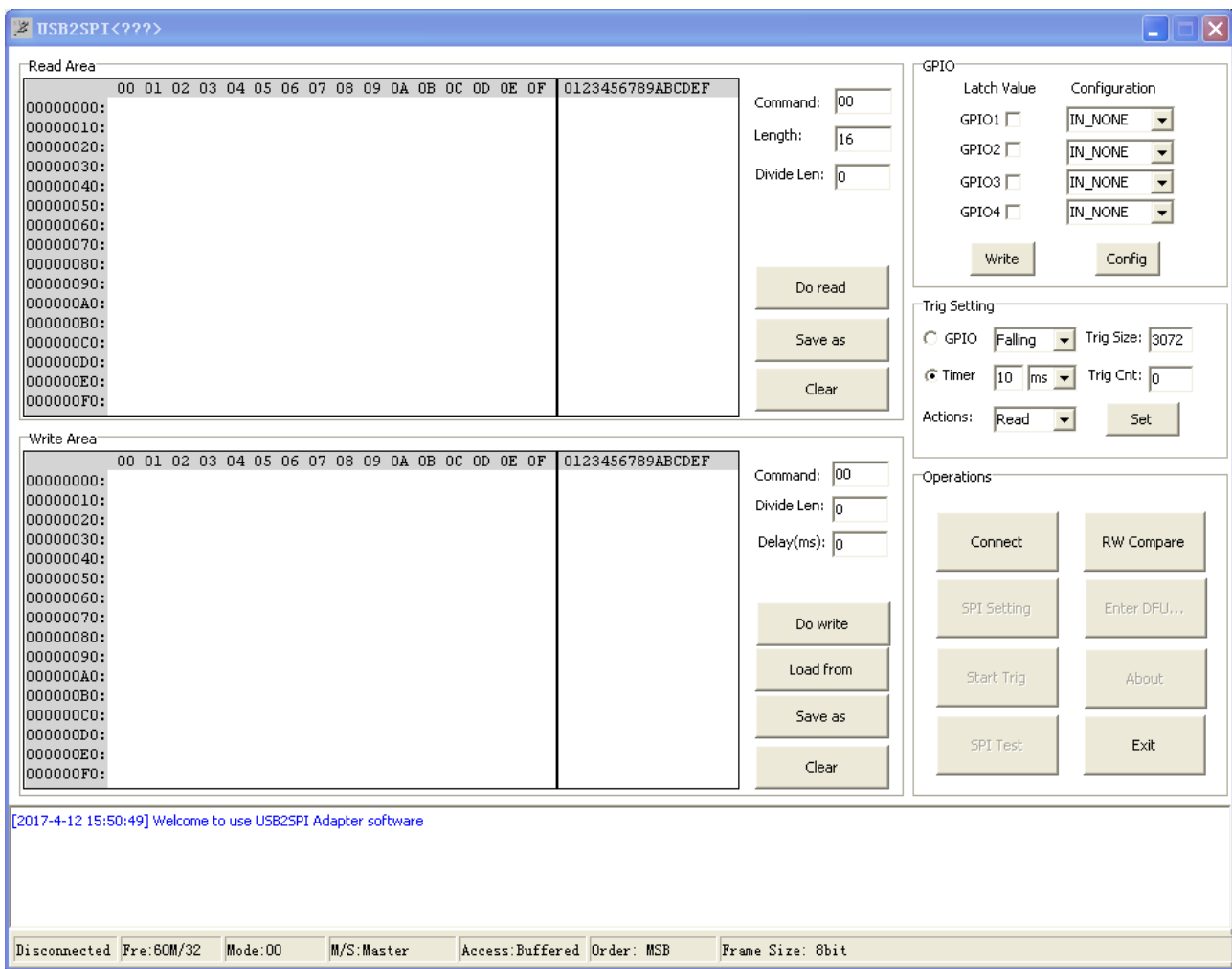


方式二：USB2SPI 适配器供电

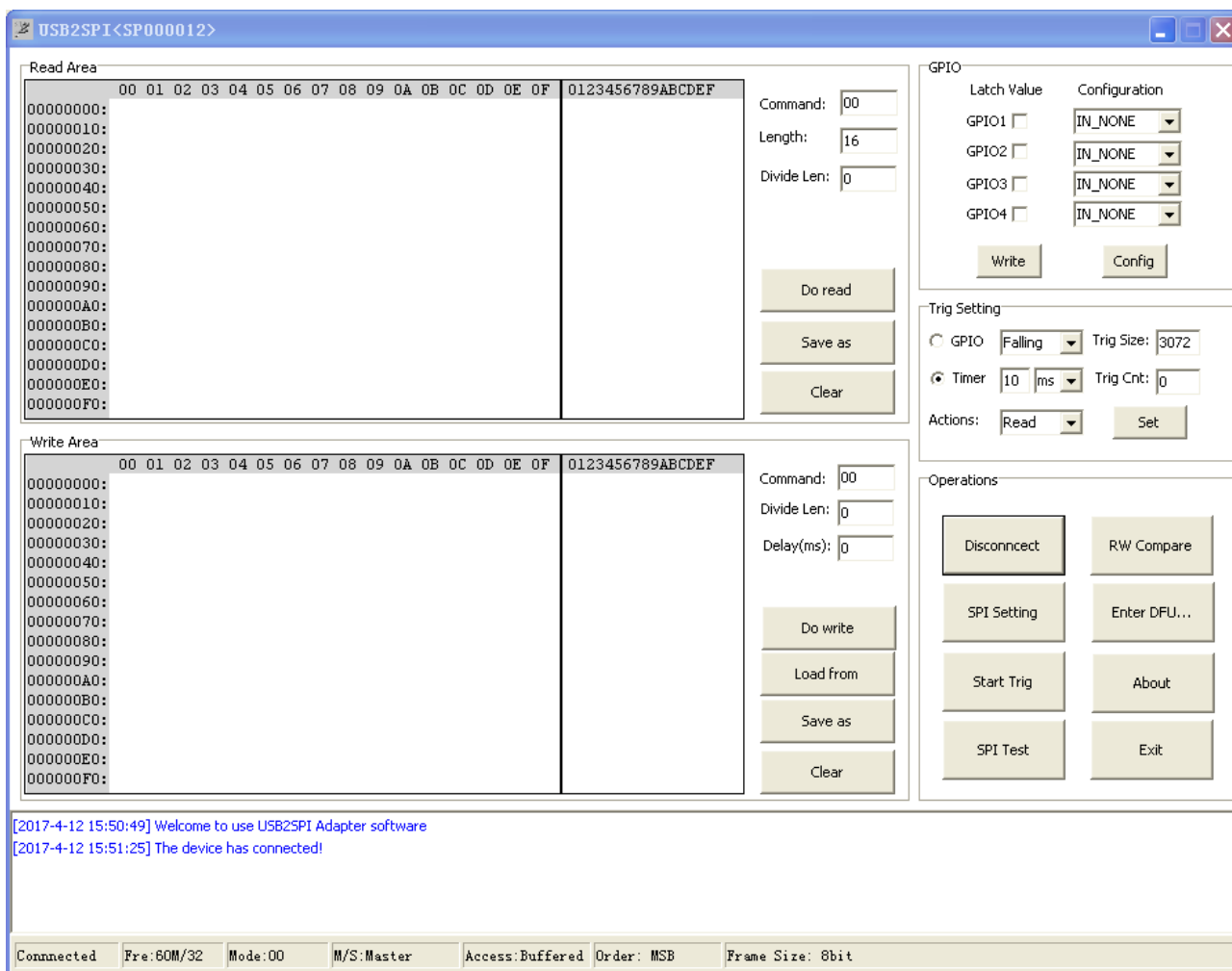


4.2 软件界面说明

在开始菜单中打开 USB2SPI，界面显示如下：

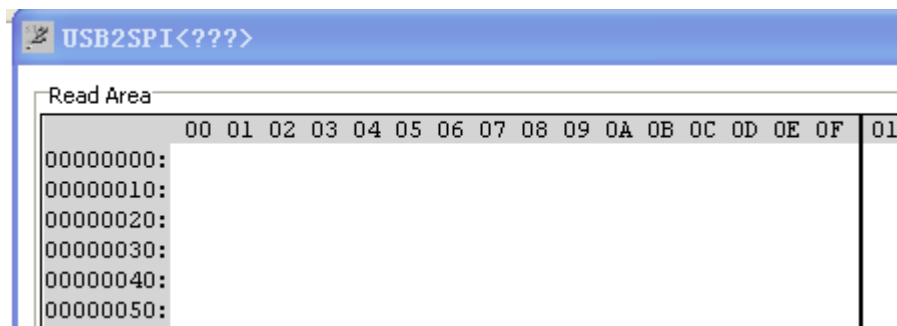


点击 “Connect” 按钮，连接成功后界面如下图

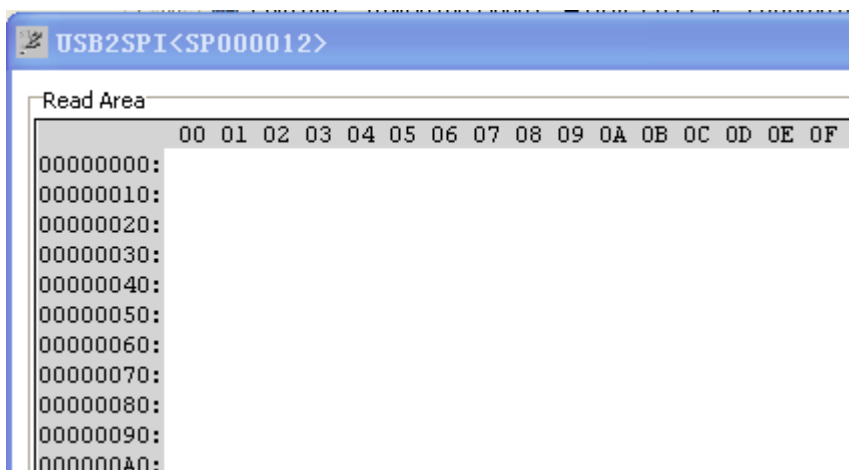


#### 4.2.1 标题栏

未连接时，标题栏显示为“USB2SPI<???”，如下图：

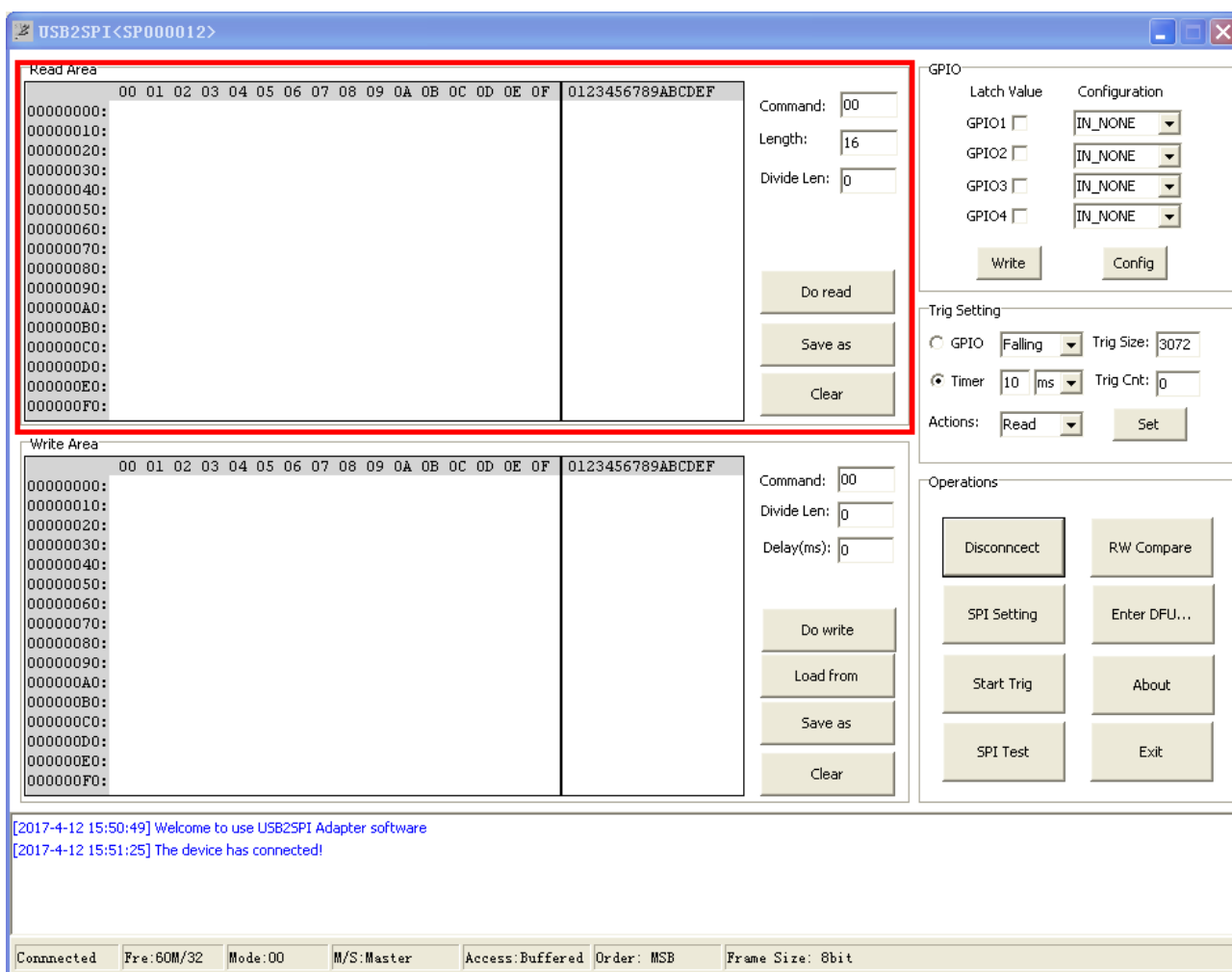


连接后，标题栏显示为“USB2SPI<序列号>”，如下图：

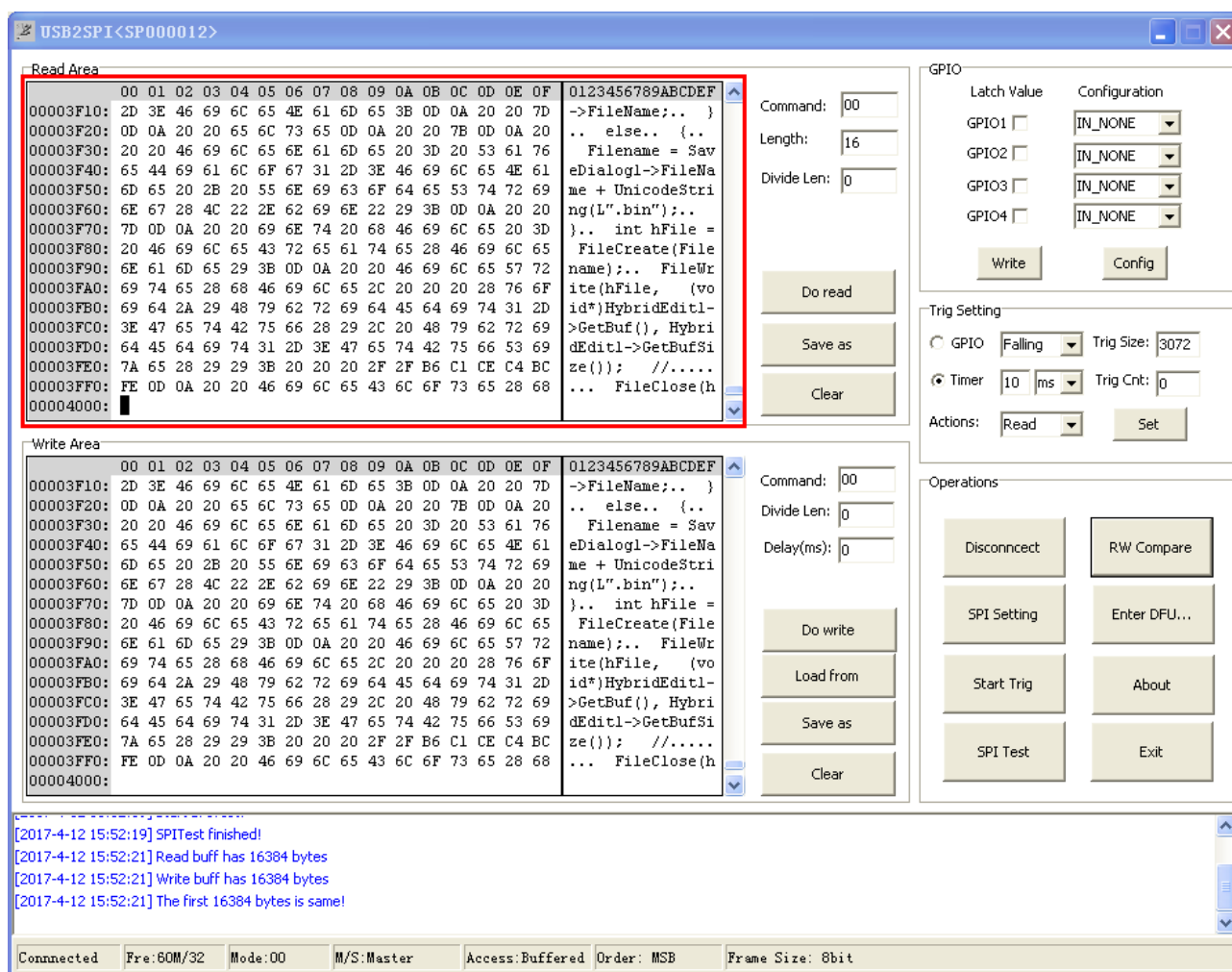


## 4.2.2 SPI 读区域

如下图红色部分，具体功能描述见 SPI 读操作



其中内容框见下图红色部分：



内容框用来显示从 SPI 端口读到的数据内容，分为二部分，左边是 16 进制显示，右边是对应的 ASC 字符显示，不能显示的字符一律以 '.' 替代。

**Command** 框用来填入命令参数或寄存器地址，必须是 16 进制格式输入。

**Length** 框用来填入要读的数据长度，单位是 byte，必须是十进制格式输入。

**Divide Len** 框用来填入分段读入的数据长度，单位是 byte，必须是十进制格式输入。

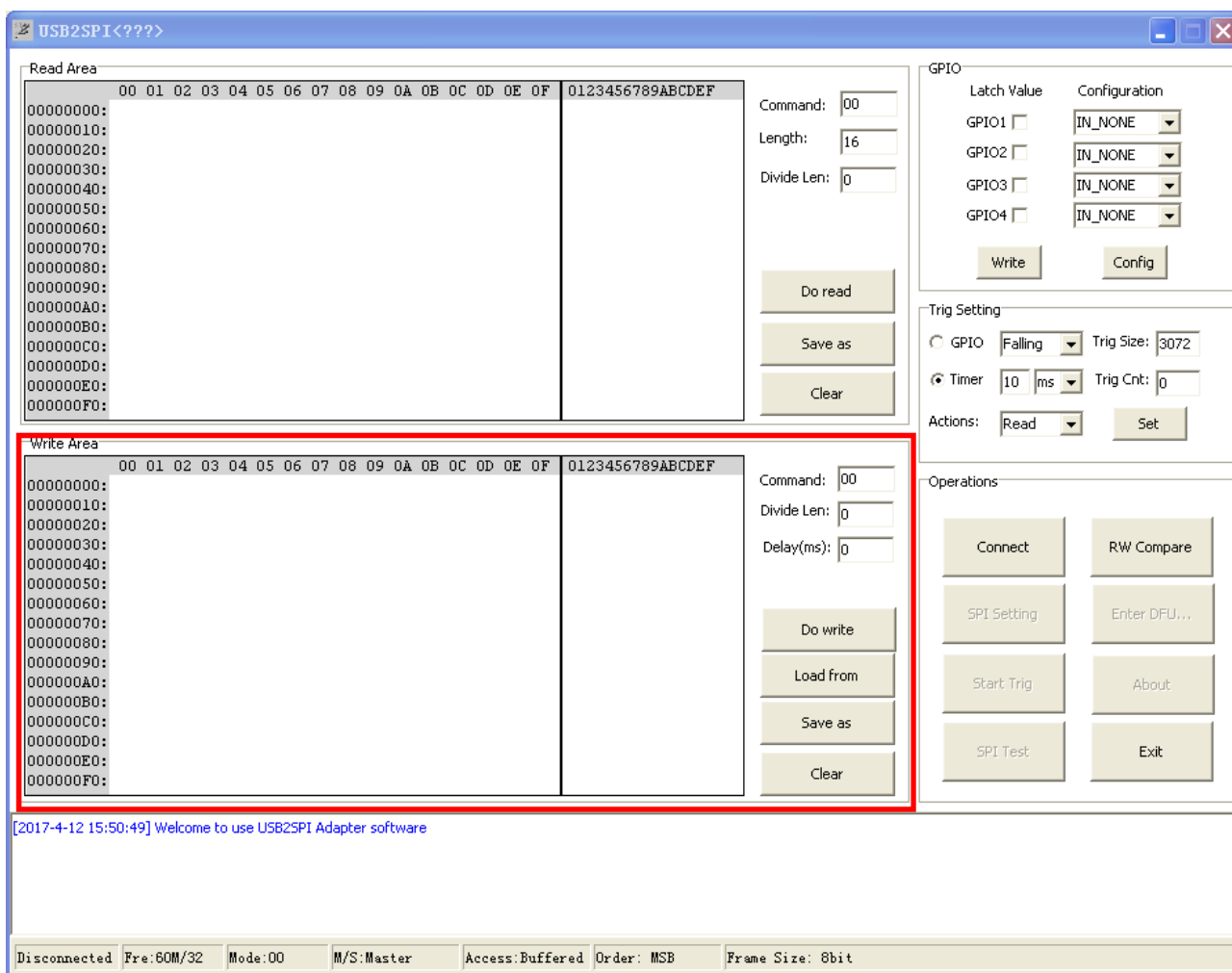
**Do Read** 按钮：点击后执行一次读操作。

**Save as** 按钮：点击后选择保存内容框的数据到文件。

**Clear** 按钮：点击后清除内容框里头的内容。

## 4.2.2 SPI 写区域

如下图红色部分，具体功能描述见 SPI 写操作



其中内容框显示功能跟 SPI 读区域一样，不同的是内容框里头的内容可以编辑。

**Command** 框用来填入命令参数或寄存器地址，必须是 16 进制格式输入。

**Divide Len** 框用来填入分段读入的数据长度，单位是 byte，必须是十进制格式输入。

**Delay** 框用来填入每次分段写完的延时，单位是毫秒，必须是十进制格式输入。

**Do Write** 按钮：点击后执行一次写操作。

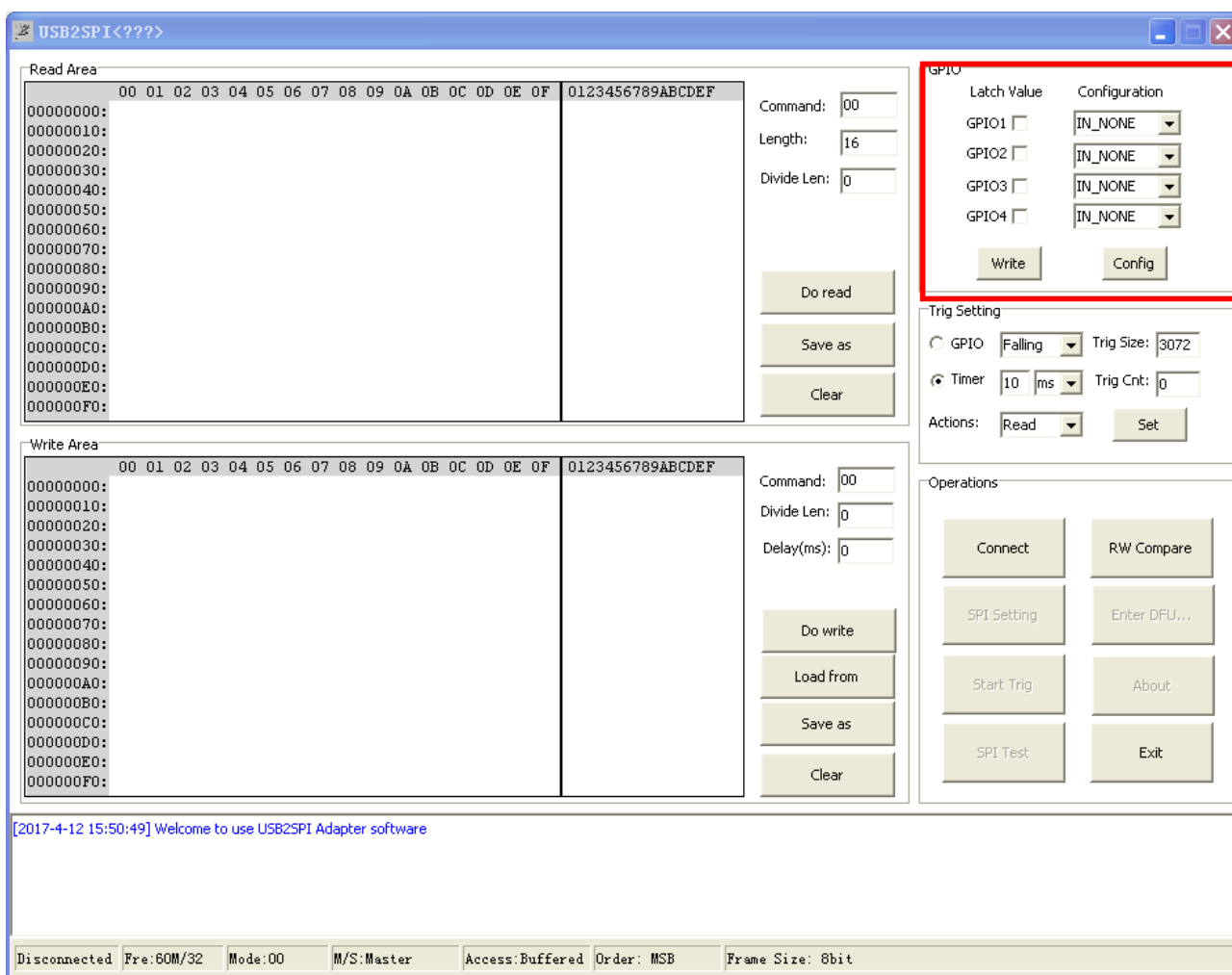
**Load from** 按钮：点击后载入文件到内容框，除了 hex 格式，其他文件一律按 bin 格式载入。

**Save as** 按钮：点击后选择保存内容框的数据到文件，文件按 bin 格式存入。

**Clear** 按钮：点击后清除内容框里头的内容。

### 4.2.3 GPIO 配置和值设置区域

如下图红色部分，具体功能描述见 GPIO 使用说明



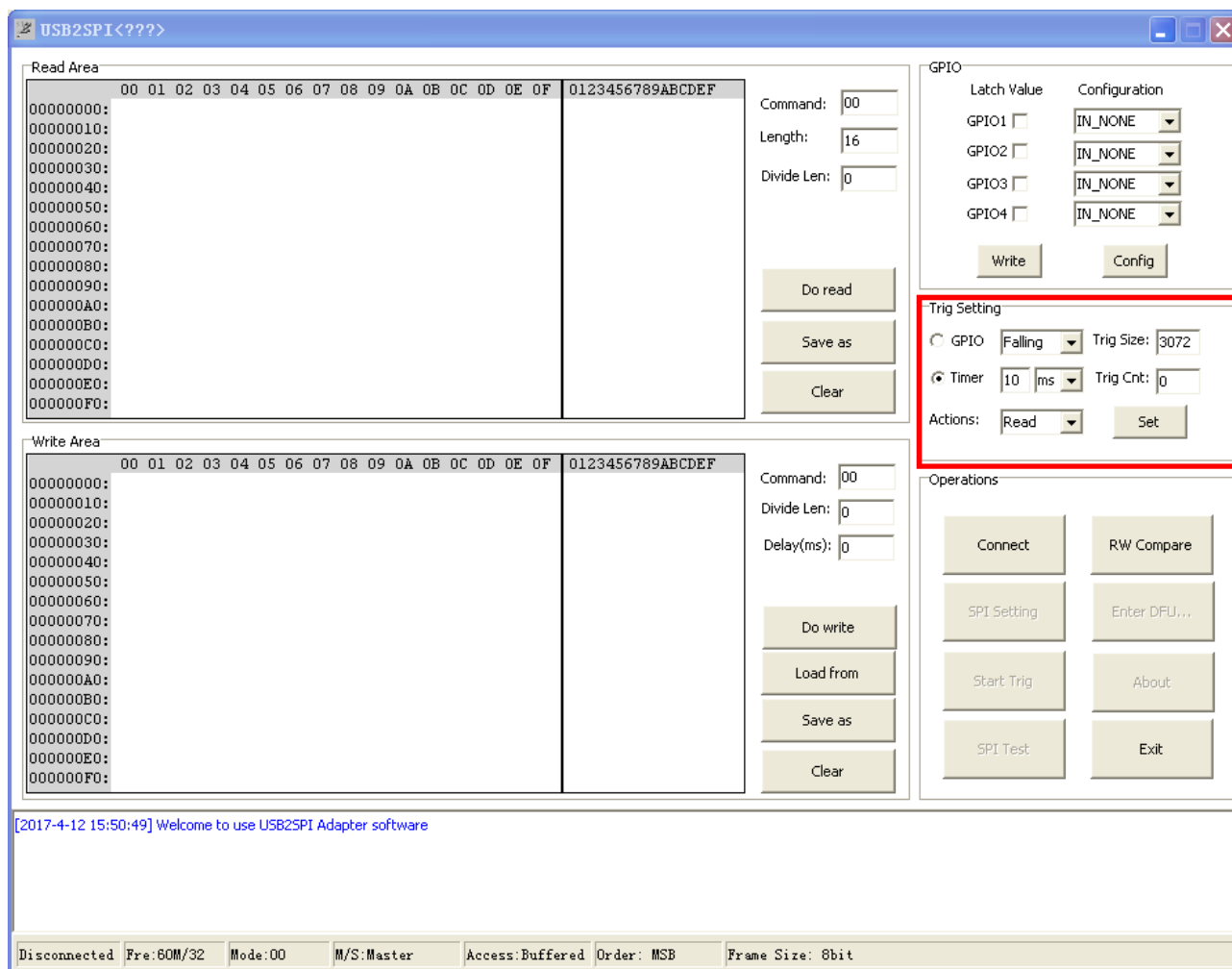
**Latch Value** 勾选时，对于输入，表示对应的 GPIO 电平高，不勾选是低；对于输出，勾选时表示待设置电平是高电平，不勾选时表示待设置电平是低电平。要点击 “Write” 按钮才发生作用。

**Write** 按钮：点击时用来设置 GPIO 电平。

**Config** 按钮：点击时用来配置 GPIO 功能。

#### 4.2.4 Trig 配置区域

如下图红色部分，具体功能描述见 SPI 触发操作



两种触发模式，GPIO 触发或者 Timer 触发。

**Trig Size:** 用来设置触发的缓存大小，最大 3072 byte。

**Trig Cnt:** 用来设置触发的次数。

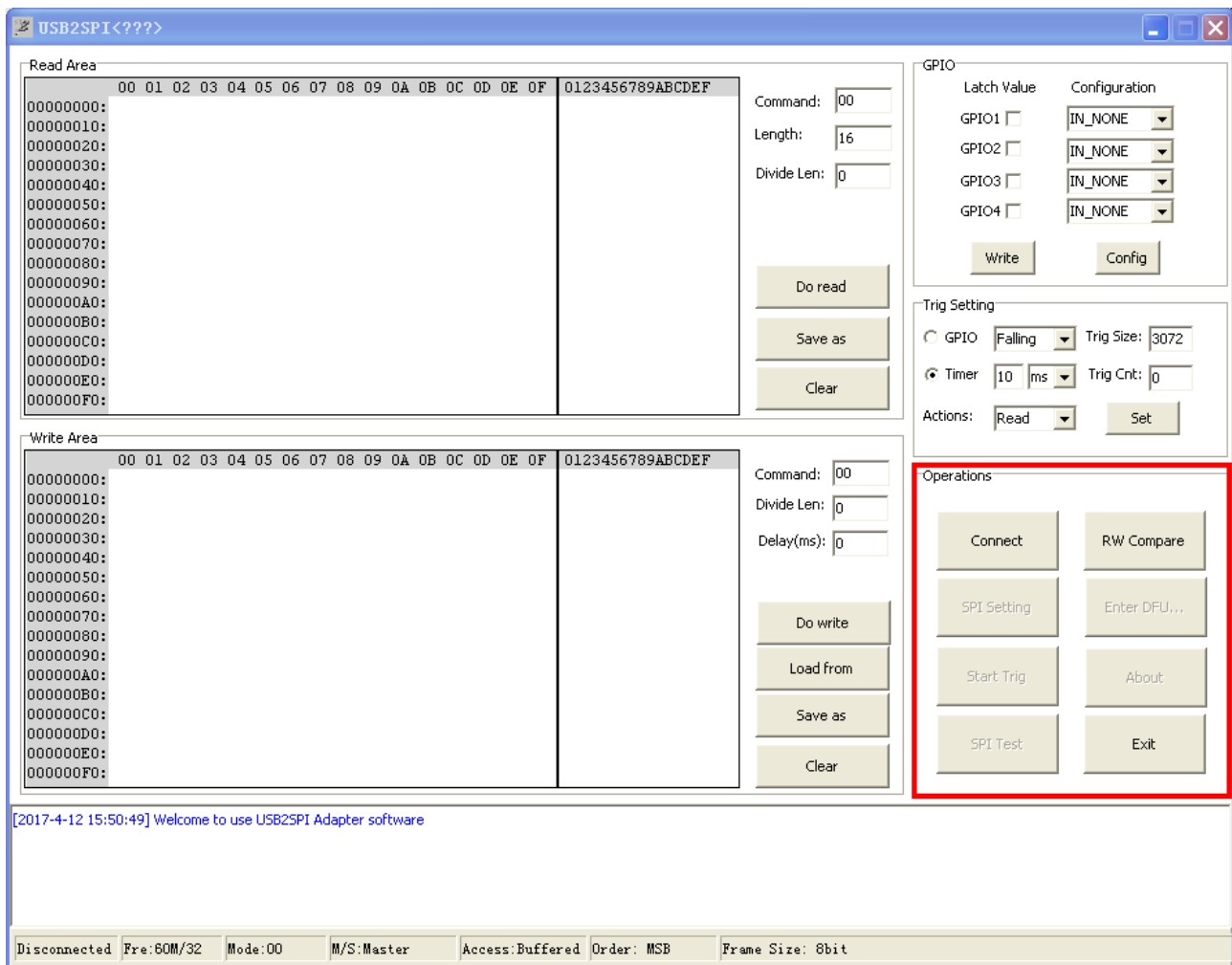
**Actions:** 用来设置触发产生后 SPI 执行的操作。

**Set 按钮:** 点击后执行触发配置。

#### 4.2.5 SPI 适配器操作区域

如下图红色部分





**Connect** 按钮：点击后连接或断开一个 USB2SPI 适配器设备。

**RW Compare** 按钮：点击后比较 SPI 读区域内容框和 SPI 写区域的内容框内容是否一致。

**SPI Setting** 按钮：点击后设置 SPI 参数。

**Enter DFU...**按钮：点击后 USB2SPI 进入 DFU 模式。

**Start Trig** 按钮：点击后 USB2SPI 进入触发模式。

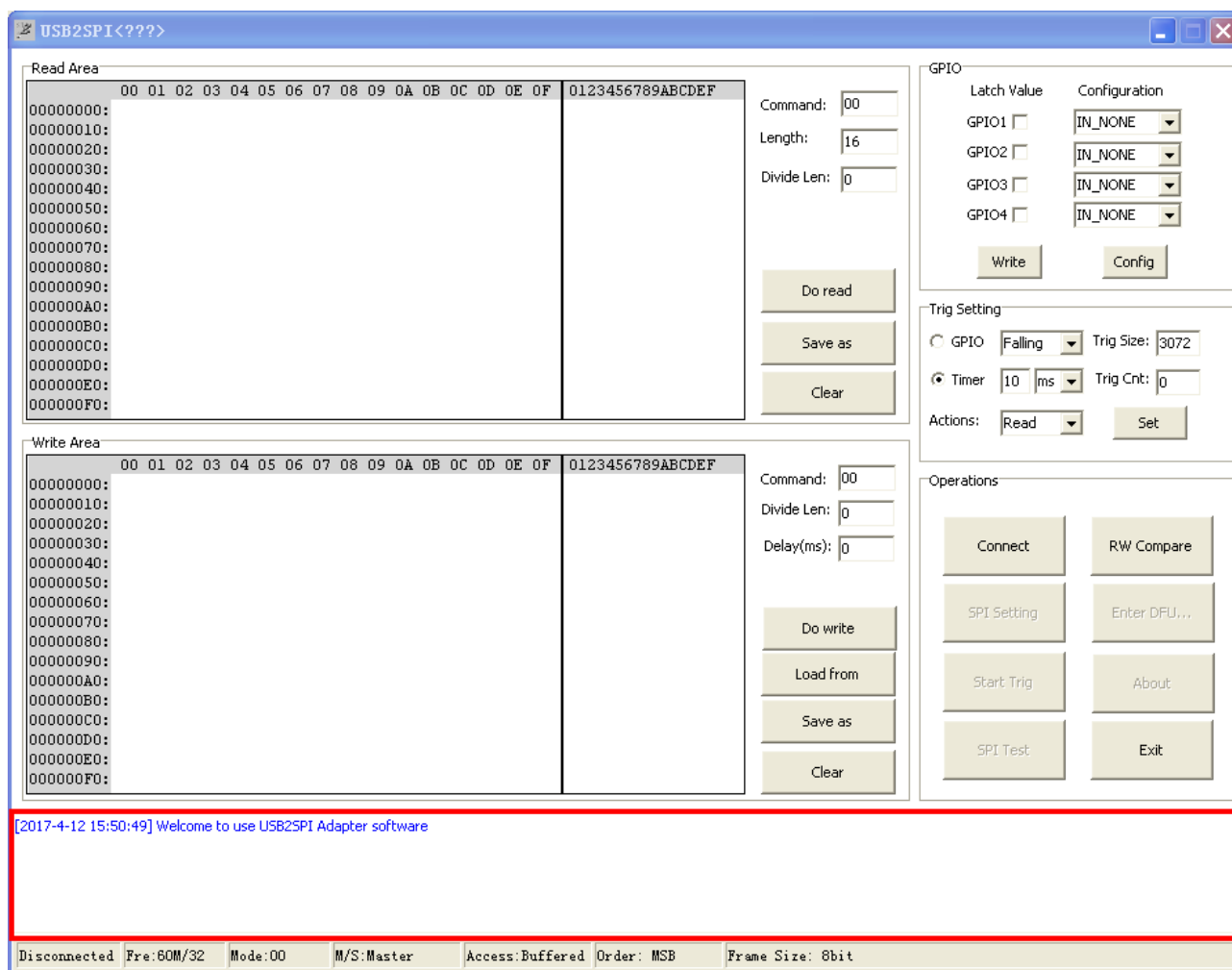
**About** 按钮：点击后显示 USB2SPI 产品软件信息和技术支持联系方式。

**SPI Test** 按钮：点击后执行一次 SPI 的双工通讯。

**Exit** 按钮：点击后关闭软件。

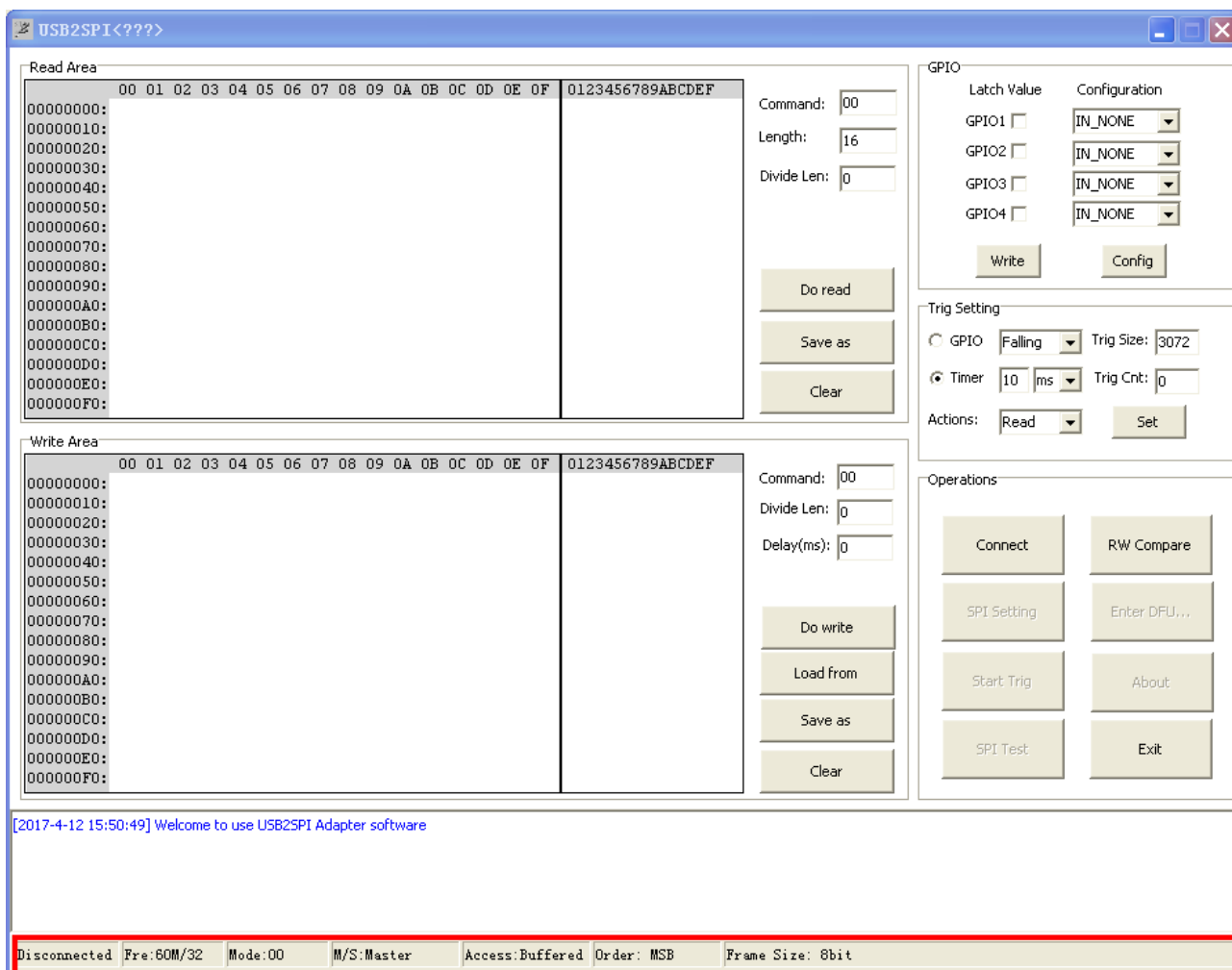
## 4.2.6 Logo 信息区域

如下图红色部分



## 4.2.7 状态栏区域

如下图红色部分



第一列显示 USB2SPI 适配器连接状态。

第二列显示 SPI 设置的频率。

第三列显示 SPI 的极性和相位。

第四列显示 SPI 是主模式还是从模式。

第五列显示 SPI 读写方式是 direct 还是 buffer。

第六列显示 SPI 数据发送顺序。

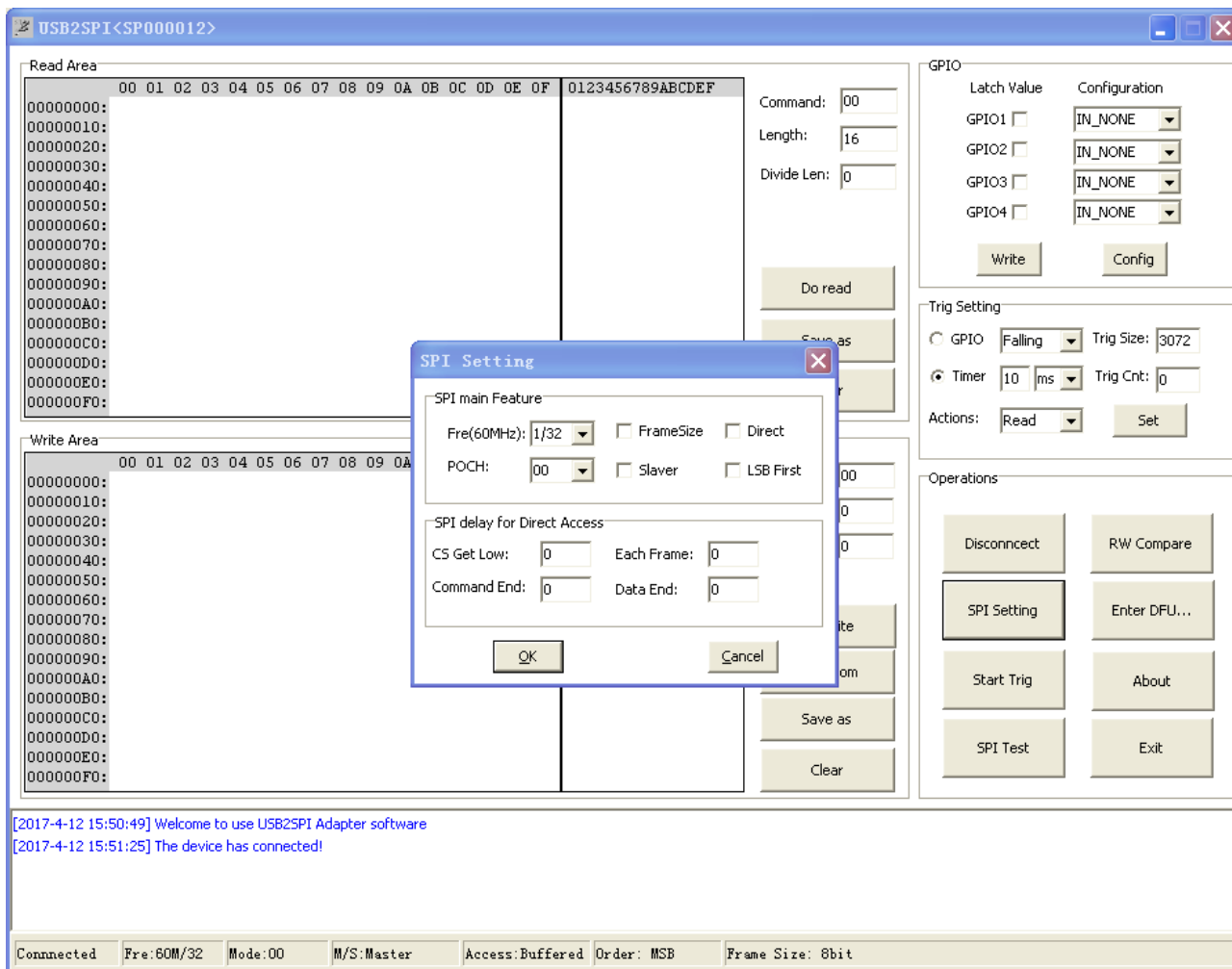
第七列显示 SPI 数据帧长度。

## 4.3 使用 SPI 功能

### 4.3.1 SPI 作为主设备使用

#### 4.3.1.1 SPI 参数配置

USB2SPI 适配器上电后 SPI 默认工作在主模式，连接成功软件界面最下端状态栏显示 SPI 默认的参数，如果需要更改，点击“SPI Setting”按钮，界面如下图：



各个参数说明如下：

**Fre:** SPI 主模式频率，实际频率是 60M 乘以分频系数(1/2,1/4,1/8,1/16,1/32,1/64,1/128,1/256).

**POCH:** SPI 工作模式，极性和相位。

- 00: CLK 空闲时低电平，第一个时钟沿采样；
- 01: CLK 空闲时低电平，第二个时钟沿采样；
- 10: CLK 空闲时高电平，第一个时钟沿采样；
- 11: CLK 空闲时高电平，第二个时钟沿采样；

**FrameSize:** SPI 数据寄存器长度，勾选时是 16bit 长，不勾选是 8bit 长。

**Slaver:** SPI 设备模式，勾选时工作为从设备，不勾选是为主设备。

**Direct:** SPI 读写方式，勾选时 Direct 读写方式，不勾选时 Buffer 读写方式。

**LSB First:** SPI 寄存器发送顺序，勾选时 LSB 优先，不勾选是 MSB 优先。

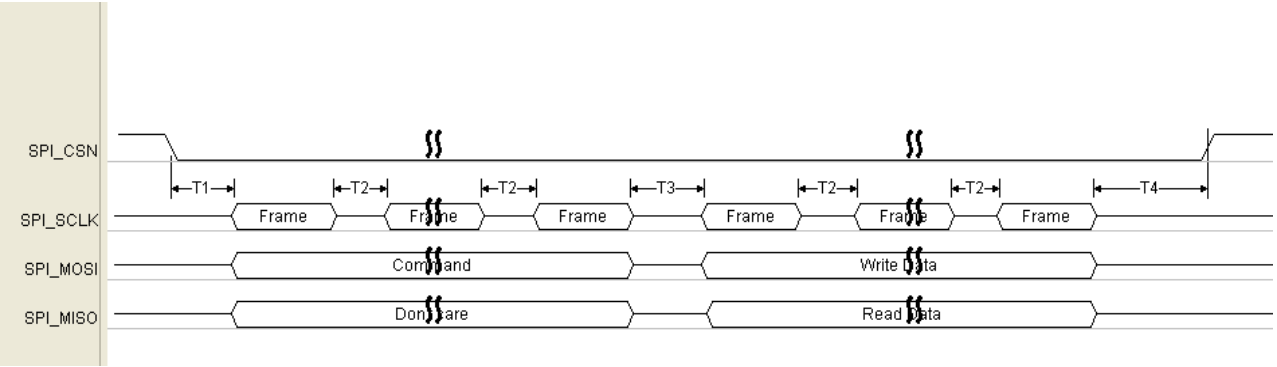
**Direct** 读写方式下的参数说明（这些参数只有在设置为 Direct 读写方式时才起作用）：

**CS Get Low(T1):** CS 变低后到开始发送数据的时间间隔，单位微秒。

**Command End(T3):** Command 发送后到数据传输时时间间隔，单位微秒。

**Each Frame(T2):** 每个帧之间的时间间隔，单位微秒。

**Data End(T4):** 数据传输结束后到 CS 变高的时间间隔，单位微秒。时序图示如下：



#### 4.3.1.2 SPI Direct 和 buffer 读写方式的差别

传输特点

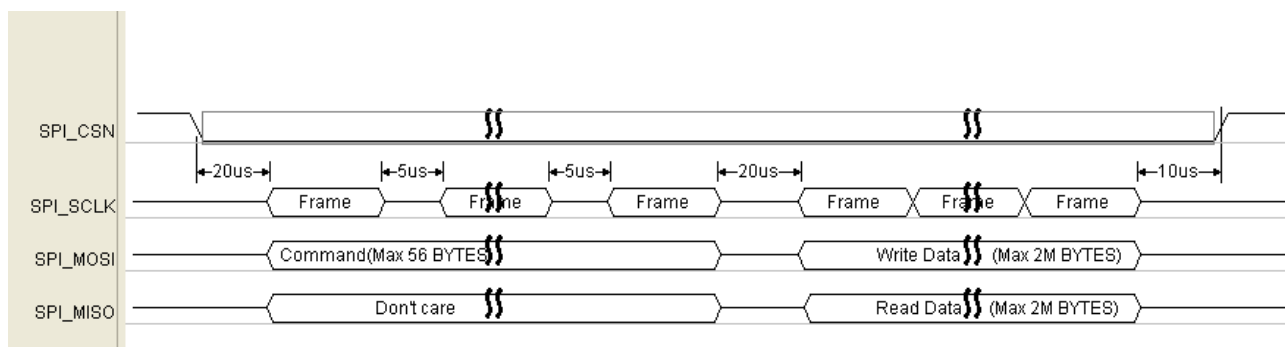
	Direct 方式	Buffer 方式

传输特点	<p>读: USB 适配器从 SPI 端口读完指定长度的数据后传给上位机;</p> <p>写: 上位机发送完要写入的数据后给 USB 适配器后, 再写入到 SPI 端口。</p>	USB 读写和 SPI 读写同时进行, SPI 读写以大小为 8192 byte 的块传输。块内传输帧间无延时。
最大可传输包大小	最大可传输大小是 48K	最大可传输大小是 2M
速率	传输速率慢, 时序可以控制	传输速度快
使用范围	数据包小, 速率要求不高, 时序可控制的场合	数据包大, 速率要求快, 需要大量数据传输的场合
备注:	<p>1&gt;触发模式时, SPI 传输只能是 direct 方式, 无需设置;</p> <p>2&gt;SPI 作为从设备, 这两种模式都不起作用;</p> <p>3&gt;<math>48K = 48 * 1024 = 49152</math>, <math>2M = 2 * 1024 * 1024 = 2097152</math></p>	

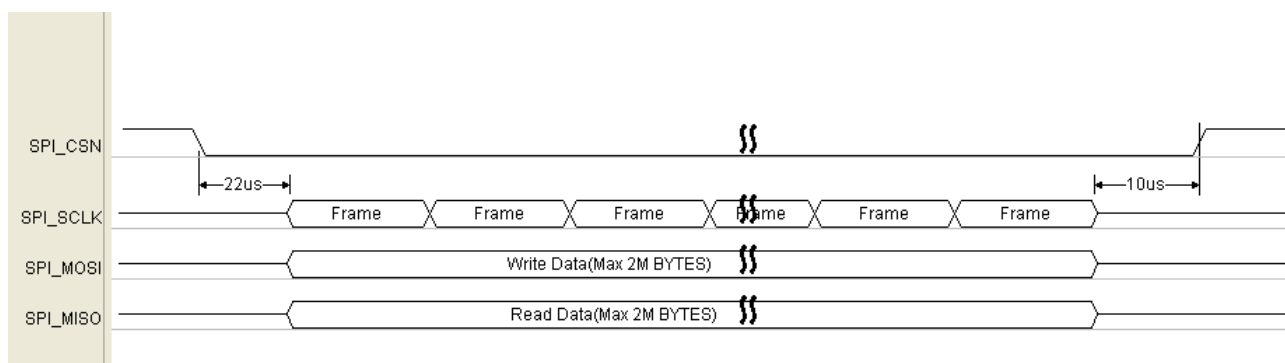
时序差别:

Buffer 方式读写:

时序图 1: (有命令参数)

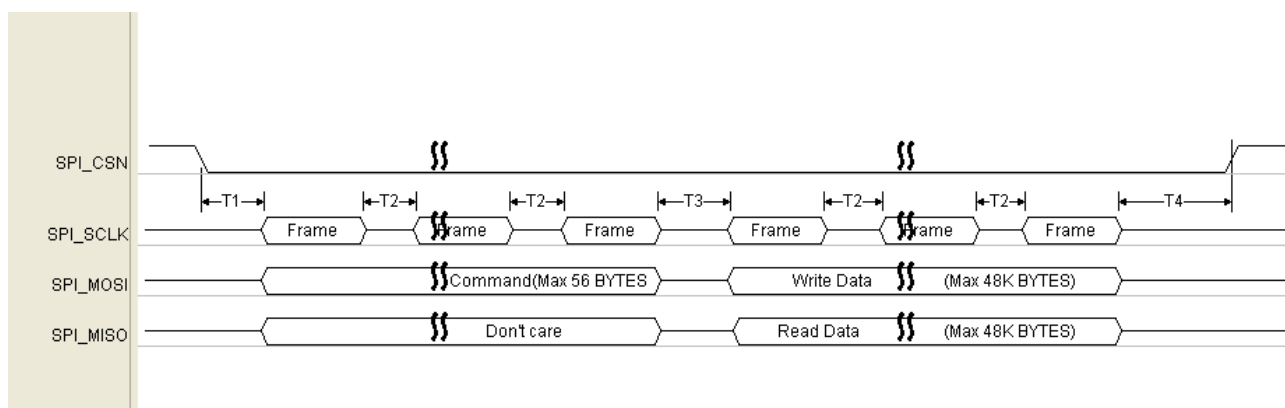


时序图 2: (无命令参数)

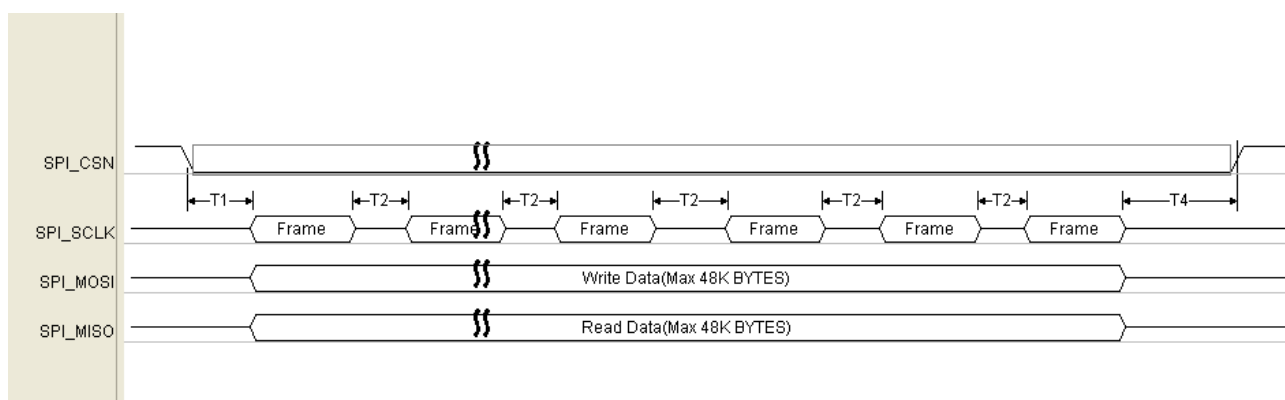


Direct 方式读写:

时序图 3: (有命令参数)



时序图 4: (无命令参数)



#### 4.3.1.2 SPI 双工通讯和自检

SPI 读写可以同时进行，即双工通讯。

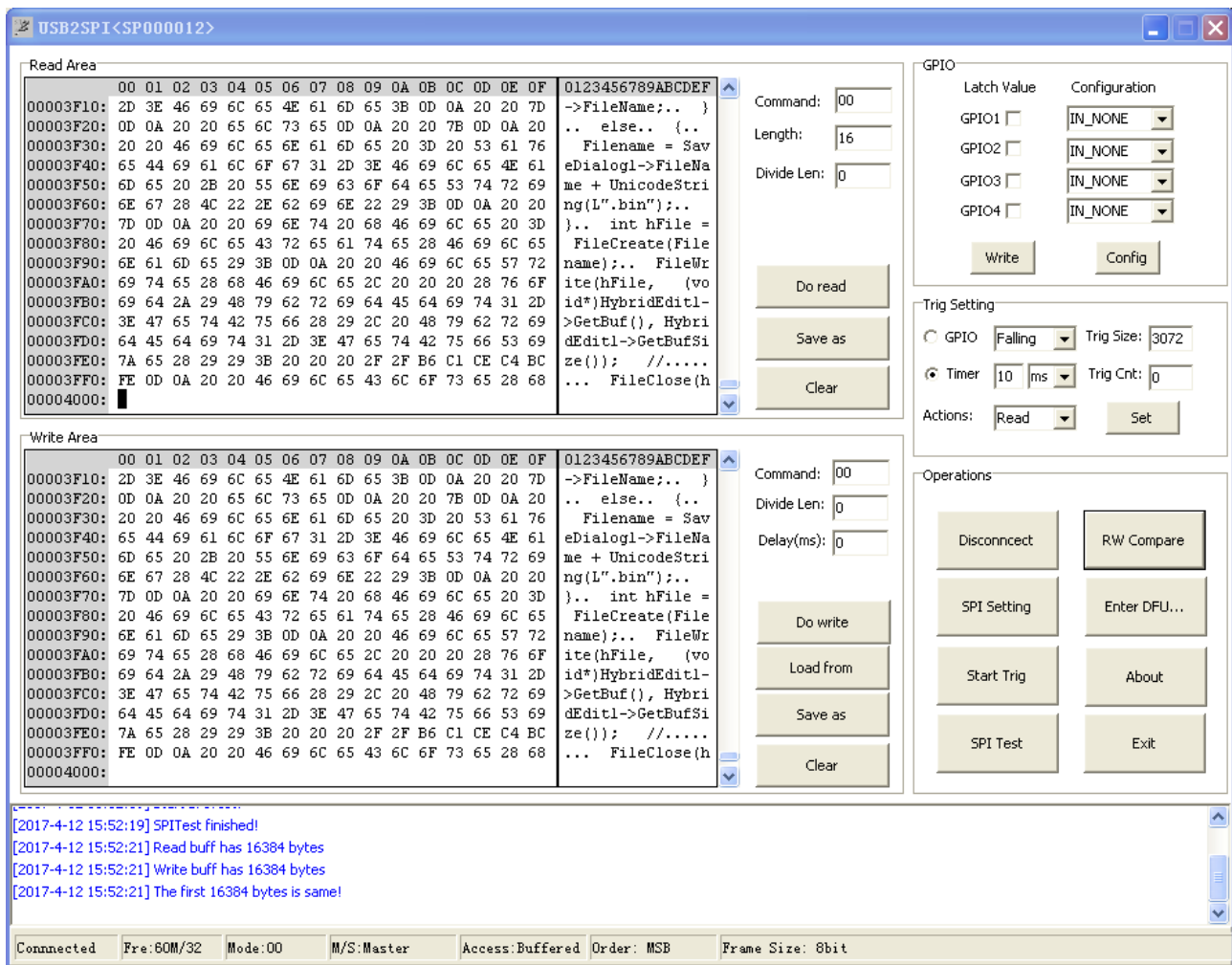
使用双工通讯时，SPI 必须工作为主设备，使用双工通讯时，先将要发送的数据添加或载入到写数据缓存区间，然后点击“SPI Test”按钮，SPI 在发送写区域数据时，顺便把收到的数据显示在读数据缓存区间。

双工通讯时命令参数区域的数据将被忽略。

双工通讯最大长度是 49152 byte，不论是 direct 还是 buffer 访问。

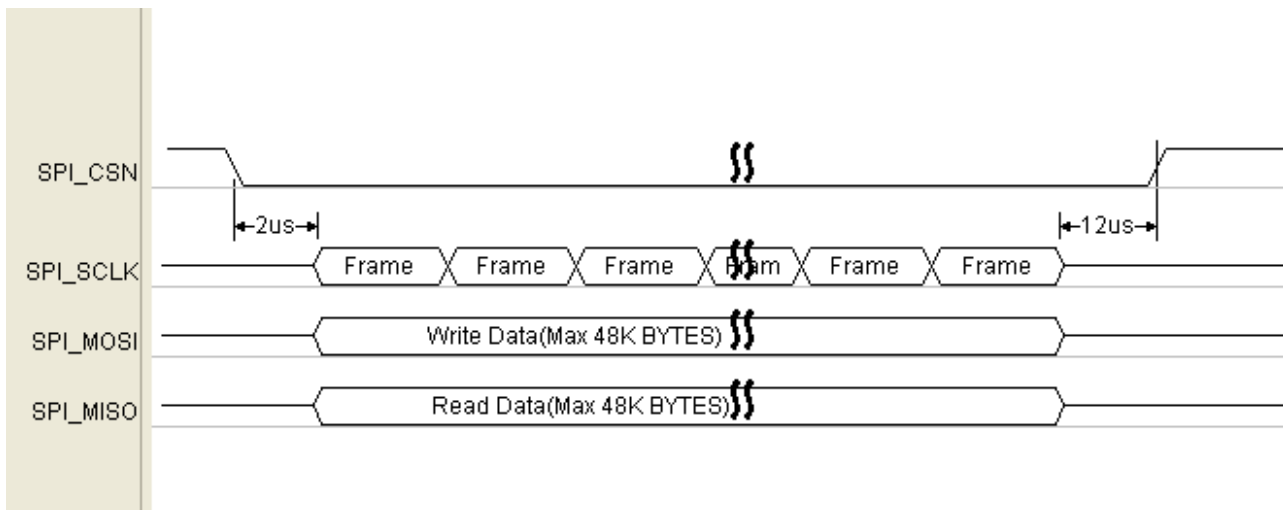
可以利用 SPI 的双工通讯实现 SPI 的自检，首先确保 SPI 工作为主设备，然后将 MSIO 和 MOSI 短接，连接成功后，在写数据缓存区域添加或载入要写的数据，点击“SPI Test”按钮，SPI 收到的数据将显示在读区域，再点击“RW Compare”按钮比较写入的数据和收到数据是否一致。演示结果如下图：



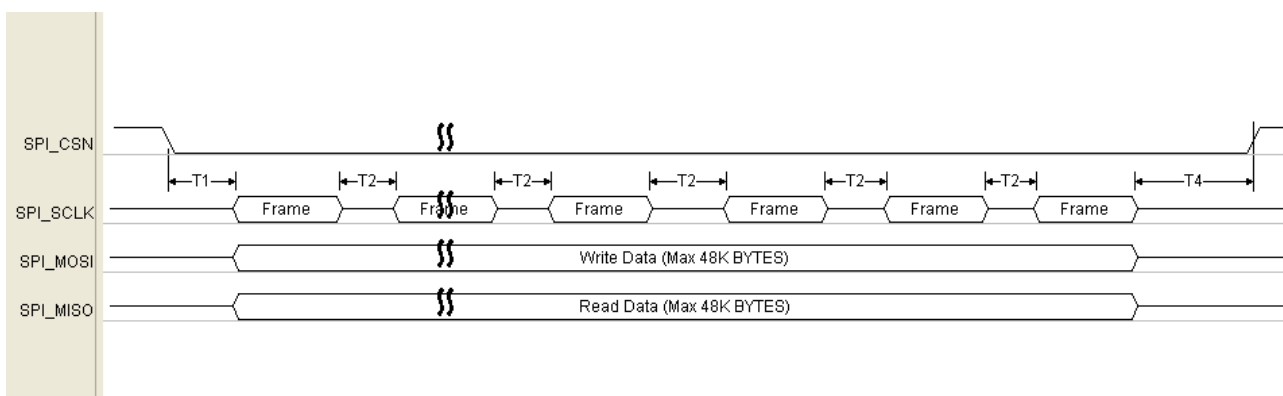


双工通讯时序图如下：

Buffer 读写方式



Direct 读写方式



#### 4.3.1.3 SPI 读操作

##### 操作步骤:

1>填入命令参数, 如果没有, 则什么都不填, 默认是 00。

2>填入读数据长度, 必填且必须大于 0, 默认是 16。

3>填入分段的长度, 默认的分段长度是 0, 即分段大小就是 SPI 一次读的最大包大小。如果 SPI 设置为 Direct 读方式, 最大包大小是 48K, 如果设置为 buffer 读方式, 最大包大小是 2M。分段长度设置值只能小于等于这个值。如果读数据长度大于分段大小, 则以分段设置值为单位读, 知道读完为止, 如果小于等于分段大小, 则一次性读入设置的数据包长度。

4>点击 do read 按钮执行读出的结果。

##### 操作分类:

1>无命令参数读操作，Command 框里头什么都不填，见 4.3.1.2 章节中的无命令参数的时序图 2 和 4。MOSI 线上的 write data 忽略。

2>有命令参数读操作，见 4.3.1.2 章节中的带命令参数的时序图 1 和 3。MOSI 线上的 writedata 忽略，MISO 线上的与 command 同步的数据忽略。

#### 4.3.1.4 SPI 写操作

##### 操作步骤：

1>填入命令参数，如果没有，则什么都不填，默认是 00，不同的设备会有不同的命令参数。

2>填入或载入要写入的数据内容到内容框，如果没有，则不填。

3>填入分段写长度，默认的分段长度是 0，即分段大小就是 SPI 一次写的最大包大小。如果 SPI 设置的是 Direc，则一次性最大写入的数据包大小是 48K，如果是 buffer 写方式，则最大数据包大小是 2M。分段长度设置值只能小于等于这个值。如果写数据长度大于分段大小，则以分段设置值为单位写，直到写完为止，如果小于等于分段大小，则一次性写入内容框里头的的数据。

4>填入分段写延时，单位毫秒，默认是 0，即不延时，一个分段写完后接着写下一个分段。如果不为 0，这每写入一段后，延时设置的时间后，再写入写一个段。

5>点击 do write 按钮执行写操作。

##### 操作分类：

1>无命令参数写操作，Command 框里头什么都不填，见 4.3.1.2 章节中的无命令参数的时序图 2 和 4。MISO 线上的所有数据忽略。

2>有命令参数写操作，见 4.3.1.2 章节中的带命令参数的时序图 1 和 3。MISO 线上的所有数据忽略。

#### 4.3.1.5 SPI 触发操作

##### 操作步骤:

1>选择触发模式，GPIO 或 Timer 触发。

2>设置触发条件，如果是 GPIO 触发，则需要选择 GPIO 触发条件，有 falling，raising，和 falling-raising 三种触发条件；只有 GPIO2 用于 IO 触发选择，其他 IO 不能用作 IO 触发。如果是 TIMER 触发，则需要设置定时时间，有 us，ms，s 三种定时时间间隔单位，设置的时间间隔必须保证能执行完所选的触发操作。

3>选择触发操作，有三种操作可供选择：

读操作：产生触发后，执行读操作，读操作的参数由 SPI 读区域的 command 和 Length 决定。

读后写：产生触发后，先执行读操作，后执行写操作。读操作的参数由 SPI 读区域的 command 和 Length 决定，写参数由 SPI 写区域的 command 决定。

写后读：产生触发后，先执行写操作，后执行读操作。读操作的参数由 SPI 读区域的 command 和 Length 决定，写参数由 SPI 写区域的 command 决定。

4>设置触发大小，此参数用于设置读缓存大小，当产生触发后执行读操作的数据字节数累计达到设置的值后，USB2SPI 适配器上报这些数据。默认值是 3072，也是最大可设置值。

5>设置触发次数，此参数用于设置触发的次数，当触发次数达到此设定值时，自动退出触发模式，停止产生触发事件。默认值为 0，表示触发次数无限大，触发模式不会自动停止。

6>点击“Set”按钮，触发配置生效。

7>点击“Start Trig”按钮，进入触发模式。在触发模式下，再次点击该按钮，则退出触发模式。

##### 注意事项:

1>在 SPI 触发模式，SPI 读写一律按 Direct 方式进行；

2>触发后产生的数据会显示在 SPI 读区域的内容框里头；

3>停止触发后，如果有剩下的未达到触发长度的数据会一次性上报到 SPI 读区域的内容框里头。

## 4.3.2 SPI 作为从设备使用

### 4.3.2.1 进入从模式

点击“SPI Setting”按钮，进入设置 SPI 参数对话框，勾选 Slaver。按“Ok”按钮进入从设备模式。

### 4.3.2.2 使用说明

- 1>在从设备模式，SPI 时钟频率，SPI FrameSize，SPI 读写方式,SPI Delay 参数不起作用。
- 2>在从设备模式，SPI 端口收到的数据会自动显示在 SPI 读区域的内容框。
- 3>在从设备模式，SPI 内部有一个 3072 的环形 buff，供 SPI 端口发送。每次发送都是以该环形 buff 的起始地址开始发送。超过 3072 后又回到起始地址。
- 4>在从设备模式，该环形 buff 可以读，在 SPI 读区域，command 参数填入环形 buff 的起始地址，再设置读入的长度，可以查看改环形 buff 的内容。
- 5>在从设备模式，该环形 buff 可以写，在 SPI 写区域，command 参数填入环形 buff 的起始地址，再填入或载入写的内容，可以更改该环形 buff 的内容。

## 4.4 GPIO 使用

### 4.4.1 使用说明

USB2SPI 适配器另外配置了 4 个 GPIO 供辅助使用，每个 GPIO 有 9 种模式，分别说明如下：

- 1>IN\_NONE：输入浮空模式，没有上拉和下拉；
- 2>IN\_PU：输入上拉模式；
- 3>IN\_PD：输入下拉模式；
- 4>PP\_NONE：推挽输出无上拉无下拉；
- 5>PP\_PU：推挽输出带上拉；
- 6>PP\_PD：推挽输出带下拉；
- 7>OD\_NONE：开漏输出无上拉无下拉；
- 8>OD\_PU：开漏输出带上拉；
- 9>OD\_PD：开漏输出带下拉；

#### 4.4.2 操作步骤:

1>选择 GPIO 的配置值

2>点击“Config”按钮，如果配置成功，相应的 GPIO Latch value 会刷新

3>对于配置为输出的 GPIO，先勾选或者取消勾选，然后点击“Write”按钮，可以改变 GPIO 对应的输出脚的电平，勾选为高电平，不勾选为低电平。

#### 4.4.3 其他功能

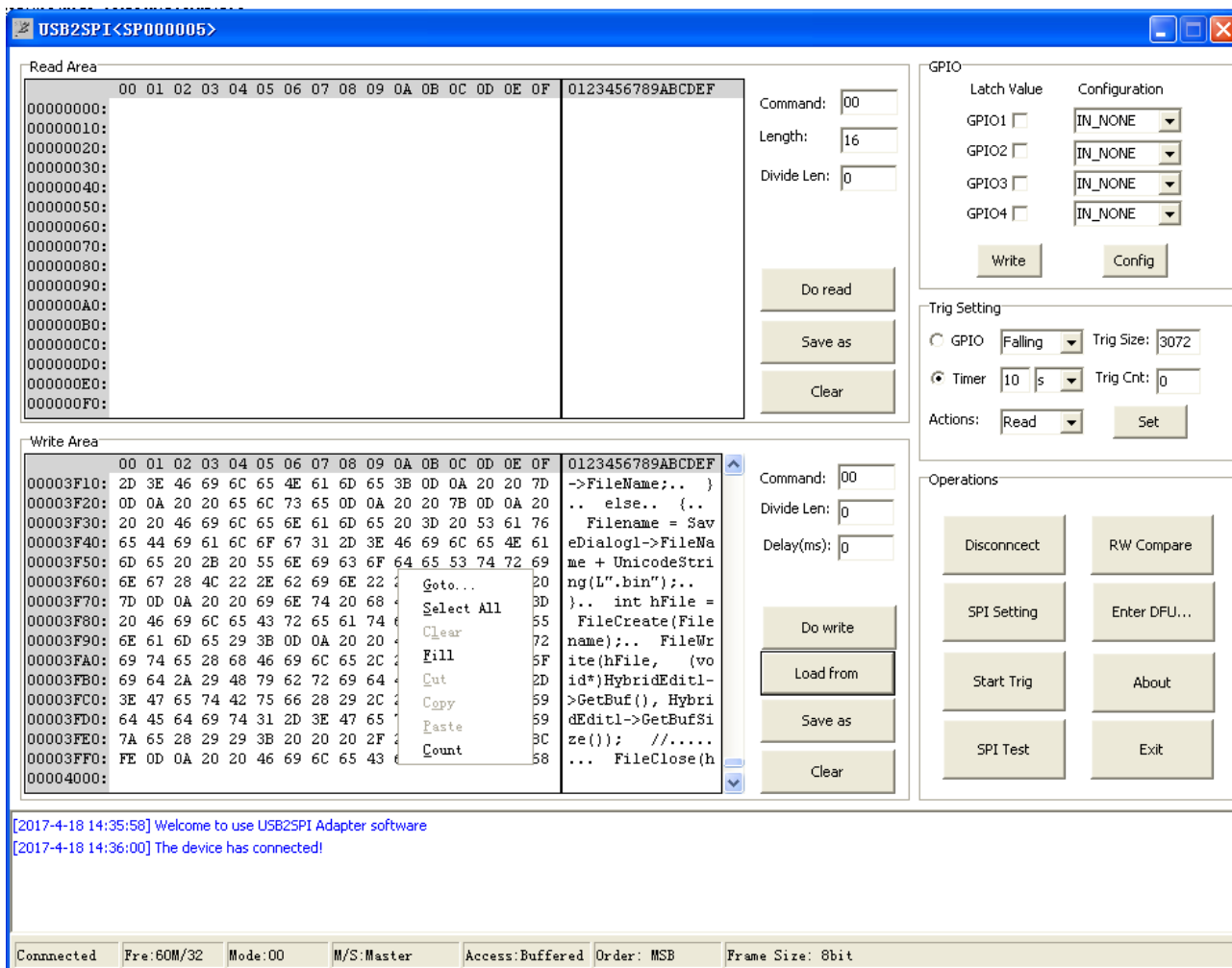
1>对于 GPIO1,上电时作为 DFU 模式检测脚，如果是低电平，则进入 DFU 模式；

2>对于 GPIO2，在进入 IO 触发模式后自动配置为输入脚，作为 IO 触发检测脚。

### 4.5 右键菜单操作

#### 4.5.1 内容框右键菜单操作

当鼠标移动到 SPI 读区域或者 SPI 写区域的内容框里头时，按右键将会弹出个右键，如下图：



菜单功能说明:

Goto...: 将编辑光标定位到所在偏移地址

Select All: 全选所有缓存内容

Clean: 将所选内容删除

Fill: 将某块区域内容全部改为相同的字节

Cut: 将所选内容删除, 并copy到粘贴板

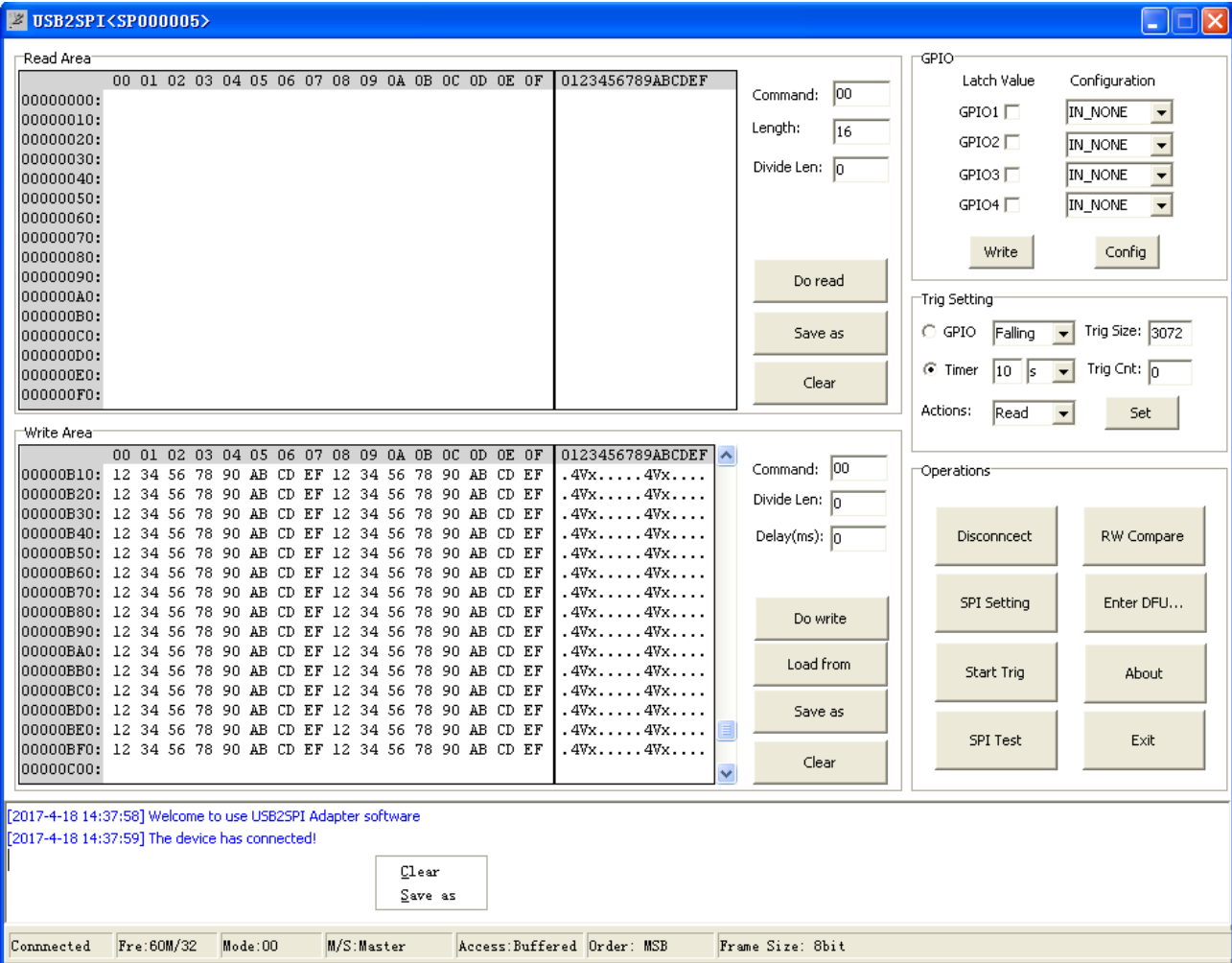
Copy: 将所选内容copy到粘贴板

Paste: 将粘贴板内容插入到编辑光标所在位置

Count: 统计内容框缓存的长度, 单位是 byte

4.5.1 log 信息框右键菜单操作

当鼠标移动到 Log 信息框里头时，按右键将会弹出个右键，如下图：



菜单功能说明：

Clear：清除 Log 信息。

Save as：以文本模式保存 Log 信息到文件。

4.6 固件更新

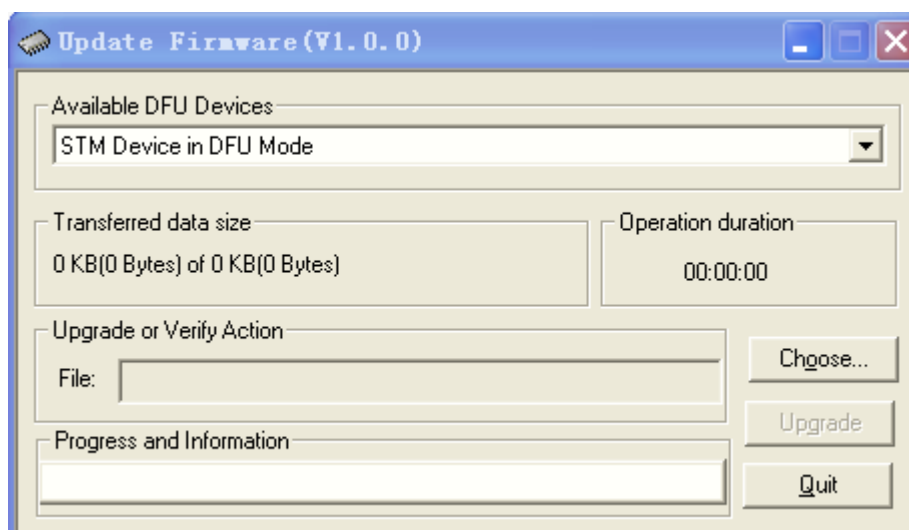
除非特别说明，USB2SPI 里头的固件无需更新。需要更新时，先进入 DFU 模式。有两种方式进入 DFU 模式。

一是将 IO1 和 GND 短接，然后上电，这样可以进入 DFU 模式；二是在应用软件界面，连接设备后，点击“Enter DFU ...”按钮进入 DFU 模式。

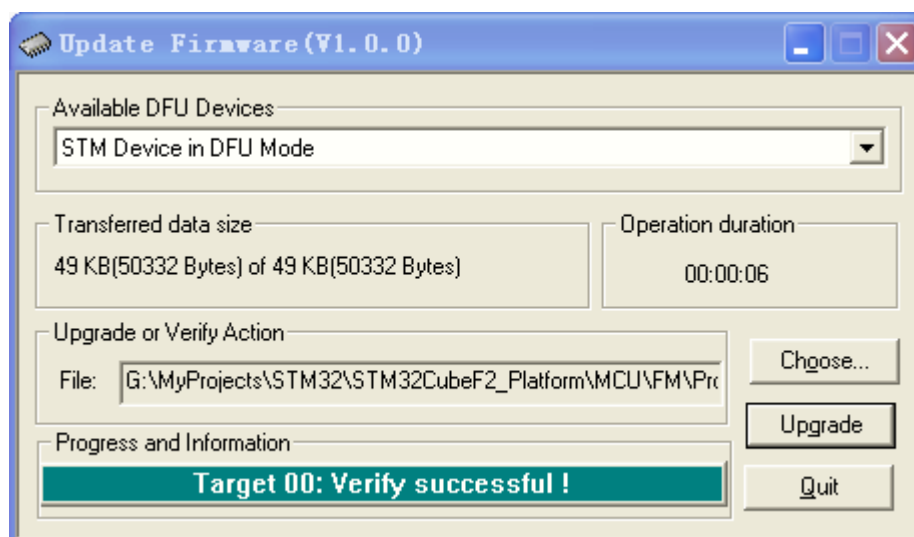


进入 DFU 模式后，Data 指示以 200ms 的间隔固定闪烁。

运行开始菜单的 DFU 软件，界面显示如下：



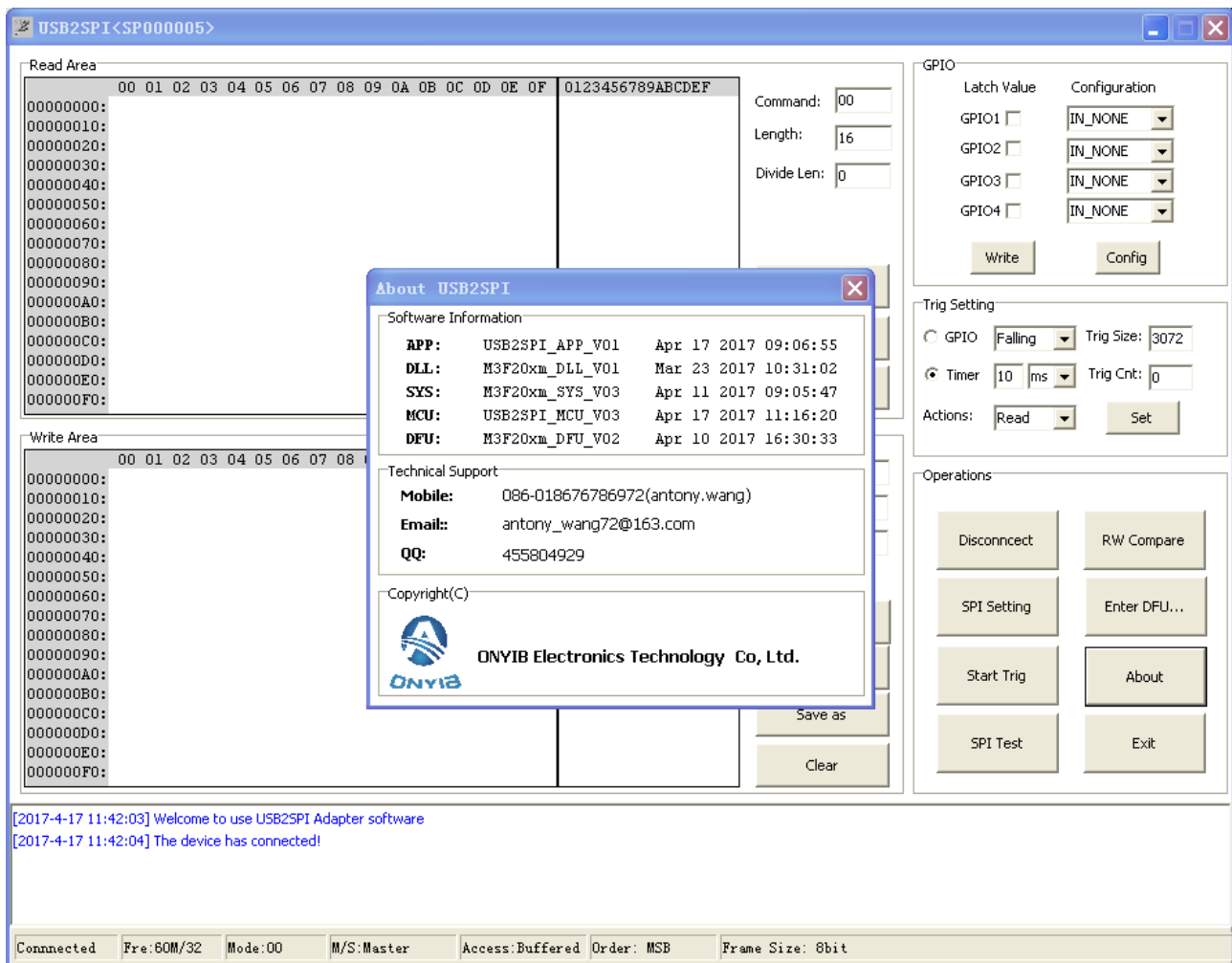
点击“choose”按钮，选择更新的文件，然后点击“Upgrade”按钮，更新成功后界面显示如下：



关闭 DFU 软件，重新拔插 USB2SPI 适配器，就可以使用新的固件了。

## 4.7 软件版本信息和技术支持

连接设备后，点击“about”，则弹出设备软件版本号，生成日期，以及技术支持联系方式，如下图：



## 五、自主开发

用户可以自己开发上位机软件，有关库函数调用请查看下载的软件包中的库函数文件夹里头的《USB2SPI 适配器库函数说明》。另外，下载的软件包里 DEMO 文件夹中，含有 VS2019 的 C#,VC,VB 三个 demo 项目，供开发参考。如果需要 linux 的库，请联系技术支持人员。

## 六、维护

- 1> 三个月的非人为硬件损坏，可以免费更换。
- 2> 软件终身免费更新和升级。
- 3> 特殊要求的可定制，价格面议。