

1553 系列板卡

用户手册

版 本： 13.1.0

发布日期： 2018-7-1

目 录

1553 系列板卡.....	0
1. 概述.....	7
1.1. 功能.....	2
1.2. 基本工作环境.....	3
1.3. 板卡功能结构图.....	4
2. 硬件使用说明.....	5
2.1. 通讯接口定义.....	5
2.2. 硬件安装.....	6
3. 驱动接口说明.....	7
3.1. 驱动运行环境.....	7
3.2. 驱动安装说明.....	7
3.3. 驱动程序使用说明.....	7
3.3.1. 驱动程序引用文件.....	7
3.3.2. 驱动程序引用的结构.....	7
3.3.2.1. BC 发送消息附加功能结构.....	7
3.3.2.2. BC 发送消息分支跳转结构.....	8
3.3.2.3. BC 发送消息结构.....	9
3.3.2.4. BC 发送微帧附加功能结构.....	11
3.3.2.5. BC 发送微帧结构.....	12
3.3.2.6. BC 发送微帧参数结构.....	13
3.3.2.7. BC 发送微帧簇结构.....	14
3.3.2.8. 接收消息结构.....	14
3.3.2.9. BC 发送消息自动重试结构.....	18
3.3.2.10. BC 发送消息/微帧停止结构.....	19
3.3.2.11. RT 状态字置位结构.....	19
3.3.2.12. RT 非法命令表结构.....	20
3.3.2.13. RT 子地址发送模式结构.....	21
3.3.2.14. MT 命令字过滤表结构.....	21
3.3.2.15. 外部触发输出结构.....	23
3.3.2.16. 外部触发输入结构.....	23
3.3.3. 驱动程序函数接口说明.....	24
3.3.3.1. MF1553_Open.....	24
3.3.3.2. MF1553_Close.....	25
3.3.3.3. MF1553_Reset.....	25
3.3.3.4. MF1553_ResetChannel.....	25
3.3.3.5. MF1553_SpeedSel.....	25
3.3.3.6. MF1553_AddTimeTag.....	25
3.3.3.7. MF1553_SetTimeTag.....	25
3.3.3.8. MF1553_GetTimeTag.....	26
3.3.3.9. MF1553_SetResponseTimeout.....	26
3.3.3.10. MF1553_BC_Init.....	26

3.3.3.11.	MF1553_BC_CfgWord.....	26
3.3.3.12.	MF1553_BC_WriteMsg.....	27
3.3.3.13.	MF1553_BC_WriteMsgs.....	27
3.3.3.14.	MF1553_BC_WriteFrmPar.....	27
3.3.3.15.	MF1553_BC_WriteFrm.....	27
3.3.3.16.	MF1553_BC_UpdateFrm.....	28
3.3.3.17.	MF1553_BC_AddEndOfFrmList.....	28
3.3.3.18.	MF1553_BC_GetEndOfFrmList.....	28
3.3.3.19.	MF1553_BC_FrmRunCtl.....	28
3.3.3.20.	MF1553_BC_GetMsgInfo.....	29
3.3.3.21.	MF1553_BC_GetMsgsInfo.....	29
3.3.3.22.	MF1553_BC_GetFrmParInfo.....	29
3.3.3.23.	MF1553_BC_GetFrmInfo.....	30
3.3.3.24.	MF1553_BC_WriteMsgDataBlock.....	30
3.3.3.25.	MF1553_BC_Start.....	30
3.3.3.26.	MF1553_BC_Stop.....	31
3.3.3.27.	MF1553_BC_MsgBufferClear.....	31
3.3.3.28.	MF1553_BC_GetLastMsgId.....	31
3.3.3.29.	MF1553_BC_GetLastFrmId.....	31
3.3.3.30.	MF1553_BC_GetIntMsgId.....	31
3.3.3.31.	MF1553_BC_GetIntFrmId.....	32
3.3.3.32.	MF1553_BC_GetMsgCntTotal.....	32
3.3.3.33.	MF1553_BC_GetMsgNumNewly.....	32
3.3.3.34.	MF1553_BC_ReadMsg.....	32
3.3.3.35.	MF1553_BC_ReadMsgs.....	33
3.3.3.36.	MF1553_BC_ReadLastMsg.....	33
3.3.3.37.	MF1553_BC_GetMsgRefresh.....	33
3.3.3.38.	MF1553_BC_IsRunning.....	34
3.3.3.39.	MF1553_BC_TxTriggerMode.....	34
3.3.3.40.	MF1553_BC_TxTriggerCmd.....	34
3.3.3.41.	MF1553_BC_RxTriggerMode.....	34
3.3.3.42.	MF1553_RT_Init.....	35
3.3.3.43.	MF1553_RT_RTAddr.....	35
3.3.3.44.	MF1553_RT_TxMode.....	36
3.3.3.45.	MF1553_RT_SetTimeTagOnSync.....	36
3.3.3.46.	MF1553_RT_SetIllCmdTable.....	36
3.3.3.47.	MF1553_RT_SetRTStatusWord.....	37
3.3.3.48.	MF1553_RT_VectorWordAutoClrEn.....	37
3.3.3.49.	MF1553_RT_SetVectorWord.....	38
3.3.3.50.	MF1553_RT_GetVectorWord.....	38
3.3.3.51.	MF1553_RT_SetBitWord.....	38
3.3.3.52.	MF1553_RT_WriteMsgDataBlock.....	38
3.3.3.53.	MF1553_RT_SendLoopBackCmd.....	39

3.3.3.54.	MF1553_RT_SetSaDepth.....	39
3.3.3.55.	MF1553_RT_MsgBufferClear.....	39
3.3.3.56.	MF1553_RT_GetMsgNumRxNewly.....	40
3.3.3.57.	MF1553_RT_GetMsgNumTxNewly.....	40
3.3.3.58.	MF1553_RT_ReadMsgRx.....	40
3.3.3.59.	MF1553_RT_ReadMsgsRx.....	40
3.3.3.60.	MF1553_RT_ReadMsgTx.....	41
3.3.3.61.	MF1553_RT_ReadMsgsTx.....	41
3.3.3.62.	MF1553_MT_Init.....	41
3.3.3.63.	MF1553_MT_SetCmdFilterTable.....	41
3.3.3.64.	MF1553_MT_Start.....	42
3.3.3.65.	MF1553_MT_Stop.....	42
3.3.3.66.	MF1553_MT_MsgBufferClear.....	42
3.3.3.67.	MF1553_MT_GetMsgNumNewly.....	42
3.3.3.68.	MF1553_MT_ReadMsg.....	42
3.3.3.69.	MF1553_MT_ReadMsgs.....	43
3.3.3.70.	MF1553_INT_Thread.....	43
3.3.3.71.	MF1553_INT_Init.....	43
3.3.3.72.	MF1553_INT_EnableGlobal.....	43
3.3.3.73.	MF1553_INT_EnableBit.....	44
3.3.3.74.	MF1553_INT_StatusClear.....	45
3.3.3.75.	MF1553_INT_Status.....	46
3.4.	驱动函数调用步骤.....	47
3.4.1.	驱动函数调用步骤（查询方式）.....	47
4.	Demo 应用程序说明.....	49
4.1.	Demo 运行环境.....	49
4.1.1.	硬件环境.....	49
4.1.2.	软件环境.....	49
4.1.3.	开发工具.....	49
4.2.	Demo 简介.....	50
4.2.1.	板卡选择窗口.....	50
4.2.2.	主窗口.....	50
4.2.3.	Demo 菜单功能.....	50
4.2.3.1.	工作模式菜单命令.....	50
4.2.3.2.	总线控制器（BC）菜单命令.....	51
4.2.3.3.	远程终端（RT）菜单命令.....	51
4.2.3.4.	总线监控器（BM）菜单命令.....	52
4.2.3.5.	查看菜单命令.....	52
4.2.3.6.	窗口菜单命令.....	52
4.2.3.7.	帮助菜单命令.....	52
4.2.4.	Demo 操作建议.....	52
4.2.4.1.	开启应用程序.....	52
4.2.4.2.	板卡号选择.....	53

4.2.4.3.	设置时间标签与应答超时.....	53
4.2.4.4.	新建工作模式.....	53
4.2.4.5.	总线控制器（BC）操作.....	53
4.2.4.6.	远程终端（RT）操作.....	53
4.2.4.7.	总线监控器（BM）操作.....	53
4.3.	Demo 使用详细说明.....	54
4.3.1.	板卡号选择.....	54
4.3.2.	程序运行模式说明.....	55
4.3.3.	设置时间标签与应答超时.....	55
4.3.4.	新建工作模式.....	56
4.3.5.	总线控制器（BC）操作.....	57
4.3.5.1.	设置 BC.....	57
4.3.5.2.	配置消息.....	59
4.3.5.2.1.	消息的配置与显示.....	59
4.3.5.2.2.	创建消息.....	61
4.3.5.2.3.	编辑消息.....	62
4.3.5.2.4.	编辑消息附加功能.....	64
4.3.5.3.	配置微帧.....	66
4.3.5.3.1.	微帧的配置与显示.....	66
4.3.5.3.2.	编辑微帧.....	68
4.3.5.4.	配置主帧.....	70
4.3.5.5.	停止 BC.....	71
4.3.5.6.	启动 BC.....	71
4.3.5.7.	开启/停止接收消息.....	71
4.3.5.8.	清空接收数据.....	72
4.3.5.9.	显示数据.....	72
4.3.6.	远程终端（RT）操作.....	73
4.3.6.1.	设置 RT.....	73
4.3.6.2.	配置数据.....	75
4.3.6.3.	开启/停止接收消息.....	76
4.3.6.4.	清空接收数据.....	76
4.3.6.5.	显示数据.....	76
4.3.7.	总线监控器（BM）操作.....	77
4.3.7.1.	设置过滤器.....	77
4.3.7.2.	开启/停止数据接收消息.....	78
4.3.7.3.	清空接收数据.....	78
4.3.7.4.	显示数据.....	78
4.3.8.	工具条命令.....	78
4.3.8.1.	MF1553 主窗口工具条命令.....	78
4.3.8.2.	总线控制器（BC）工具条命令.....	79
4.3.8.3.	远程终端（RT）工具条命令.....	80
4.3.8.4.	总线监控器（BM）工具条命令.....	80
4.3.9.	帮助.....	80

4.3.10.	退出程序.....	80
4.4.	配置信息的保存和加载.....	81

手册内容简介

本手册内容共分为四个章节，分别为“概述”、“硬件使用说明”、“驱动接口说明”和“应用程序说明”。

第一章“概述”，是对硬件功能、工作环境的整体概述，更多的细节在第二章“硬件使用说明”、第三章“驱动接口说明”和第四章“应用程序说明”中进行详细描述。

第二章“硬件使用说明”，描述了硬件的硬件接口定义、及安装说明。

第三章“驱动接口说明”，该部分内容是本手册的重点内容，它详细描述了接口函数的功能及使用方法，通过这一章节的了解，用户可以进行产品的应用程序开发。

第四章“应用程序说明”，介绍了提供给用户的应用程序的使用方法，通过这个应用程序，用户可以不需编程而直接进行一些基本的通讯操作，达到快速应用的目的。另外，在该章中，也提供了驱动接口使用例程，为用户开发适合个人要求的应用程序的提供参考。

1. 概述

此说明书适用于下表中我公司的各种型号板卡：

速率	接口	板卡型号	说明
1M 速率	PCI 接口	ZXB-1553-PCI-1S-1M	PCI 接口单通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-1F-1M	PCI 接口单通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-2S-1M	PCI 接口双通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-2F-1M	PCI 接口双通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-3S-1M	PCI 接口 3 通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-3F-1M	PCI 接口 3 通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-4S-1M	PCI 接口 4 通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-PCI-4F-1M	PCI 接口 4 通道多功能 1M 标准速率 1553B 板卡
	CPCI/PXI 接口	ZXB-1553-CPCI/PXI-1S-1M	CPCI/PXI 接口单通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-1F-1M	CPCI/PXI 接口单通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-2S-1M	CPCI/PXI 接口双通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-2F-1M	CPCI/PXI 接口双通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-3S-1M	CPCI/PXI 接口 3 通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-3F-1M	CPCI/PXI 接口 3 通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-4S-1M	CPCI/PXI 接口 4 通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-4F-1M	CPCI/PXI 接口 4 通道多功能 1M 标准速率 1553B 板卡
	USB 接口	ZXB-1553-USB-1S-1M	USB 接口单通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-USB-1F-1M	USB 接口单通道多功能 1M 标准速率 1553B 板卡
		ZXB-1553-USB-2S-1M	USB 接口双通道单功能 1M 标准速率 1553B 板卡
		ZXB-1553-USB-2F-1M	USB 接口双通道多功能 1M 标准速率 1553B 板卡
1/4M 速率	PCI 接口	ZXB-1553-PCI-1S-1/4M	PCI 接口单通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-1F-1/4M	PCI 接口单通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-2S-1/4M	PCI 接口双通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-2F-1/4M	PCI 接口双通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-3S-1/4M	PCI 接口 3 通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-3F-1/4M	PCI 接口 3 通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-4S-1/4M	PCI 接口 4 通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-PCI-4F-1/4M	PCI 接口 4 通道多功能 1/4M 标准速率 1553B 板卡
	CPCI/PXI 接口	ZXB-1553-CPCI/PXI-1S-1/4M	CPCI/PXI 接口单通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-1F-1/4M	CPCI/PXI 接口单通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-2S-1/4M	CPCI/PXI 接口双通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-2F-1/4M	CPCI/PXI 接口双通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-3S-1/4M	CPCI/PXI 接口 3 通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-3F-1/4M	CPCI/PXI 接口 3 通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-4S-1/4M	CPCI/PXI 接口 4 通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-CPCI/PXI-4F-1/4M	CPCI/PXI 接口 4 通道多功能 1/4M 标准速率 1553B 板卡
	USB 接口	ZXB-1553-USB-1S-1/4M	USB 接口单通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-USB-1F-1/4M	USB 接口单通道多功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-USB-2S-1/4M	USB 接口双通道单功能 1/4M 标准速率 1553B 板卡
		ZXB-1553-USB-2F-1/4M	USB 接口双通道多功能 1/4M 标准速率 1553B 板卡

1.1. 功能

■ 基本特性:

- 1553B 总线板卡, 标准双冗余设计
- 根据需求, 可选择 1M 标准速率板卡, 或 1M/4M 兼容速率板卡
- 根据需求, 可选择 1~4 通道 (USB 等部分接口卡只可选 1~2 通道)
- 根据需求, 可选择单功能, 或多功能板卡
- 单功能板卡, 即每个通道可做 1 个 BC (总线控制器), 或 0~31 个 RT (远程终端), 或 1 个 MT (总线监视器)
- 多功能板卡, 即每个通道可做 1 个 BC (总线控制器), 0~31 个 RT (远程终端), 1 个 MT (总线监视器), 即 BC、RT、MT 可同时存在
- 64 位硬件时标 (分辨率 1us)
- 软件可设置应答超时 (分辨率 0.5us, 取值 $0 \sim 2^{32}-1$)
- 硬件上为每个通道配有大容量缓存: $16M \times 16bit$

■ 详细特性:

BC 模式

- BC 按主帧发送消息, 主帧中包含 1~15 个微帧, 且每个微帧包含 1~2047 条消息
- 主帧特点如下:
 - ◆ 主帧由微帧组成, 且包含 1~15 个微帧
 - ◆ 可动态插入、删除、替换微帧
 - ◆ 可动态切换微帧的执行状态
- 微帧特点如下:
 - ◆ 微帧由消息组成, 且包含 1~2047 条消息
 - ◆ 可设置微帧的执行周期 (分辨率 1us, 取值 $0 \sim 2^{32}-1$)。设定值为 0 表示为事件微帧, 否则为周期微帧
 - ◆ 在周期微帧情况下, 可设置微帧重发次数 (取值 $0 \sim 2^{32}-1$)。设定值为 0 表示无限次, 否则为有限次
 - ◆ 可设置微帧间隔时间 (分辨率 1us, 取值 $0 \sim 2^{32}-1$)。如设定值小于实际执行时间, 则设置无效, 发送完此微帧后, 可立刻发送后续微帧。如设定值大于实际执行时间, 发送完此微帧后, 必须等待间隔时间到达后, 才可以发送后续微帧
 - ◆ 可设置微帧执行开始引发脉冲输出。脉冲电平可选择 LVTTTL/TTL、或 RS422
 - ◆ 可设置微帧执行结束引发脉冲输出。脉冲电平可选择 LVTTTL/TTL、或 RS422
 - ◆ 可设置微帧执行开始引发中断 (只用在支持中断的板卡)
 - ◆ 可设置微帧执行结束引发中断 (只用在支持中断的板卡)
 - ◆ 可设置触发输入上升沿/下降沿引发微帧执行。这里的有效沿可以有 3 种选择: 外部触发输入原始信号的有效边沿、或外部触发输入分频后的有效边沿、或外部触发输入倍频后的有效边沿。外部触发输入电平有 2 种选择: LVTTTL/TTL、或 RS422
- 消息特点如下:
 - ◆ 可设置消息自动重试, 重试通道可选, 重试条件可设, 重试次数可设 1~2 次
 - ◆ 可设置消息间隔时间 (分辨率 1us, 取值 $0 \sim 2^{32}-1$)。如设定值小于实际执行时间, 则设

置无效，发送完此消息后，可立刻发送后续消息。如设定值大于实际执行时间，发送完此消息后，必须等待间隔时间到达后，才可以发送后续消息

◆可设置分支跳转功能。表示此消息执行后在满足设置的条件下，自动跳转至指定微帧处执行。注意，微帧中任何消息都可以设置分支跳转功能，但微帧中的消息执行顺序不会被跳转打断。至少等到执行完此微帧中剩余消息后，才会执行跳转帧

◆可设置消息执行开始引发脉冲输出。脉冲电平可选择 LVTTL/TTL、或 RS422

◆可设置消息执行结束引发脉冲输出。脉冲电平可选择 LVTTL/TTL、或 RS422

◆可设置消息执行开始引发中断（只用在支持中断的板卡）

◆可设置消息执行结束引发中断（只用在支持中断的板卡）

◆可设置消息执行正确时不写入缓存

◆可设置消息执行错误时不写入缓存

■ 在发送过程中，可动态更新 BC→RT 类型消息的数据区

■ 可缓存 23831 条已处理完成的消息

■ 可支持查询方式接收消息

■ 可支持中断方式接收消息（USB 等部分接口卡除外）

RT 模式

■ 可设置 0~31 个 RT

■ 可设置非法命令表功能

■ 可设置单缓冲和循环缓冲数据发送方式

■ 循环缓冲数据发送方式下，子地址数据区大小为 4096 Words

■ 循环缓冲数据发送方式下，子地址数据区指针回卷深度可设 1~4096 间的任意值

■ 可设置矢量字自动清除功能

■ 可缓存 23831 条最新接收到的发送数据的消息

■ 可缓存 23831 条最新接收到的接收数据的消息

■ 可支持查询方式接收消息

■ 可支持中断方式接收消息（USB 等部分接口卡除外）

MT 模式

■ 可设置命令字过滤功能

■ 可缓存 23831 条最近监控到的消息

■ 可支持查询方式接收消息

■ 可支持中断方式接收消息（USB 等部分接口卡除外）

1.2. 基本工作环境

操作系统：WinXP/Win7

工作温度：-40 ~ +85℃

相对湿度：0 ~ 95%

1.3. 板卡功能结构图

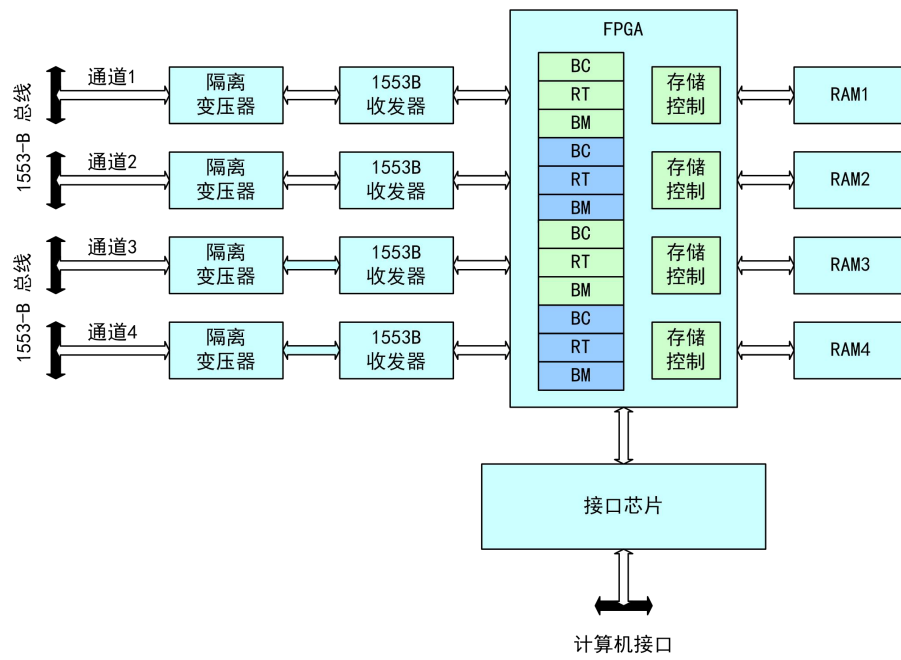


图 1-1 板卡功能结构图

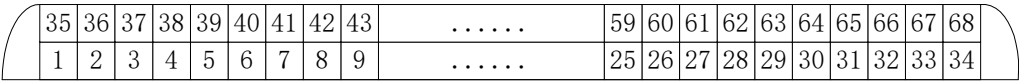
2. 硬件使用说明

2.1. 通讯接口定义

PCI 接口卡、CPCI/PXI 接口卡。此类板卡接插件为 SCSI-68 连接器，且板卡侧为母座，电缆侧为公头。

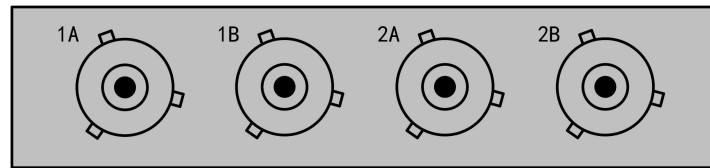
表 2-1 标准 SCSI-68 连接器（J_SCSI68）接头信号表

管脚	名称	说明	管脚	名称	说明
1	CH1A+	1 通道 A 冗余 1553 总线+	35	CH1B+	1 通道 B 冗余 1553 总线+
2	CH1A-	1 通道 A 冗余 1553 总线-	36	CH1B-	1 通道 B 冗余 1553 总线-
3	CH2A+	2 通道 A 冗余 1553 总线+	37	CH2B+	2 通道 B 冗余 1553 总线+
4	CH2A-	2 通道 A 冗余 1553 总线-	38	CH2B-	2 通道 B 冗余 1553 总线-
5	CH3A+	3 通道 A 冗余 1553 总线+	39	CH3B+	3 通道 B 冗余 1553 总线+
6	CH3A-	3 通道 A 冗余 1553 总线-	40	CH3B-	3 通道 B 冗余 1553 总线-
7	CH4A+	4 通道 A 冗余 1553 总线+	41	CH4B+	4 通道 B 冗余 1553 总线+
8	CH4A-	4 通道 A 冗余 1553 总线-	42	CH4B-	4 通道 B 冗余 1553 总线-
9			43		
10			44		
11			45		
12			46		
13			47		
14			48		
15			49	GND	系统地
16			50	GND	系统地
17			51	GND	系统地
18			52	GND	系统地
19	DOUT7	TTL 数字量输出 7	53	TX0-	RS422 电平数字量输出 0-
20	DOUT6	TTL 数字量输出 6	54	TX0+	RS422 电平数字量输出 0+
21	DOUT5	TTL 数字量输出 5	55	RX0+	RS422 电平数字量输入 0+
22	DOUT4	TTL 数字量输出 4	56	RX0-	RS422 电平数字量输入 0-
23	DOUT3	TTL 数字量输出 3	57	TX1-	RS422 电平数字量输出 1-
24	DOUT2	TTL 数字量输出 2	58	TX1+	RS422 电平数字量输出 1+
25	DOUT1	TTL 数字量输出 1	59	RX1+	RS422 电平数字量输入 1+
26	DOUT0	TTL 数字量输出 0	60	RX1-	RS422 电平数字量输入 1-
27	DIN7	TTL 数字量输入 7	61	TX2-	RS422 电平数字量输出 2-
28	DIN6	TTL 数字量输入 6	62	TX2+	RS422 电平数字量输出 2+
29	DIN5	TTL 数字量输入 5	63	RX2+	RS422 电平数字量输入 2+
30	DIN4	TTL 数字量输入 4	64	RX2-	RS422 电平数字量输入 2-
31	DIN3	TTL 数字量输入 3	65	TX3-	RS422 电平数字量输出 3-
32	DIN2	TTL 数字量输入 2	66	TX3+	RS422 电平数字量输出 3+
33	DIN1	TTL 数字量输入 1	67	RX3+	RS422 电平数字量输入 3+
34	DIN0	TTL 数字量输入 0	68	RX3-	RS422 电平数字量输入 3-



板卡上 SCSI 68 接头点号示意图

USB 接口卡。板卡的 1553 信号通过 BJ77 连接器引出，如下图，可按照挡板上的信号说明连接到网络中。注意，在板卡为单通道的情况下，信号 2A 和 2B 并不引出。



2.2. 硬件安装

a) 板卡安装：

- ① PCI 接口卡。将板卡插入主机箱的 PCI 插槽，然后用螺丝将板卡挡板和机箱固定在一起；
- ② CPCI 或 PXI 接口卡。将板卡沿导轨槽插入主机箱的插槽，直至板卡完全插入，并用螺丝固定板卡和机箱；
- ③ USB 接口卡。使用 USB（支持 USB2.0 以上）线缆将板卡连接到主机上。

b) 用板卡配套线缆将板卡接入 1553B 网络，确认接触良好后，到此硬件安装结束；

c) 开机后，提示“发现新硬件”，驱动程序安装见第三章驱动接口说明。

3. 驱动接口说明

3.1. 驱动运行环境

WinXP/Win7 操作系统

3.2. 驱动安装说明

- a) 硬件安装完毕；
- b) 计算机开机后提示“发现新硬件”，点击硬件安装的下一步，根据设备的操作系统，选择请指向光盘中“驱动安装文件\...”中相应系统的目录；
- c) 点击下一步，直到驱动程序安装成功；
- d) 安装完毕，在设备管理器会看到”1553-xxxx”设备。

3.3. 驱动程序使用说明

3.3.1. 驱动程序引用文件

库文件：MF1553.dll 和 MF1553.lib

函数库头文件：MF1553_dll.h 和 defines.h

3.3.2. 驱动程序引用的结构

3.3.2.1. BC 发送消息附加功能结构

typedef struct

```
{
    WORD   SMsgModeEndTrgOutEn;      // 消息执行结束引发脉冲输出使能
    WORD   SMsgModeStartTrgOutEn;    // 消息执行开始引发脉冲输出使能
    WORD   SMsgModeEndTrgIntEn;      // 消息执行结束引发中断使能
    WORD   SMsgModeStartTrgIntEn;    // 消息执行开始引发中断使能
    WORD   SMsgModeOkMsgRcvXEn;      // 消息执行正确时不写入缓存使能
    WORD   SMsgModeErrMsgRcvXEn;     // 消息执行错误时不写入缓存使能
}STRUCT_MF1553_BC_SMSG_MODE;
```

结构参数说明：

SMsgModeEndTrgOutEn：消息执行结束引发脉冲输出使能。最低位有效，0:不使能；1:使能。

脉冲输出必须满足以下条件：

- ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数；
- ② 消息执行结束引发脉冲输出使能。

SMsgModeStartTrgOutEn：消息执行开始引发脉冲输出使能。最低位有效，0:不使能；1:使能。

脉冲输出必须满足以下条件：

- ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数；
- ② 消息执行开始引发脉冲输出使能。

SMsgModeEndTrgIntEn: 消息执行结束引发中断使能。最低位有效，0:不使能；1:使能。

中断生成必须满足以下条件：

- ① 总中断使能；
- ② “消息执行引发中断”的分中断使能；
- ③ 消息执行结束引发中断使能。

SMsgModeStartTrgIntEn: 消息执行开始引发中断使能。最低位有效，0:不使能；1:使能。

中断生成必须满足以下条件：

- ① 总中断使能；
- ② “消息执行引发中断”的分中断使能；
- ③ 消息执行开始引发中断使能。

SMsgModeOkMsgRcvXEn: 消息执行正确时不写入缓存使能。最低位有效，0:不使能；1:使能。

不写入缓存的消息，软件是接收不到执行后的此条消息。如在 BC 发送大量 BC→RT 类型的消息时，成功执行的消息并不需要软件回读，则这里可设置使能。

SMsgModeErrMsgRcvXEn: 消息执行错误时不写入缓存使能。最低位有效，0:不使能；1:使能。

不写入缓存的消息，软件是接收不到执行后的此条消息。如在 BC 和多个 RT 通讯时，BC 与网络中不存在的 RT 通讯时，执行的结果一定是错误消息（超时错误）。为屏蔽这类消息，此参数可设置使能，软件将接收不到这些错误消息。

3.3.2.2. BC 发送消息分支跳转结构

typedef struct

```
{
    WORD    SMsgBranchEn;           // 分支跳转使能
    WORD    SMsgBranchToFrmId;      // 跳转至目标微帧的 Id
    WORD    SMsgBranchDataLocation; // 检测字在消息中的位置
    WORD    SMsgBranchDataMask;     // 检测字掩码
    WORD    SMsgBranchDataExpectation; // 检测结果期望值
    WORD    Reserve;                // 保留
}STRUCT_MF1553_BC_SMSG_BRANCH;
```

结构参数说明：

SMsgBranchEn: 分支跳转使能。最低位有效，1: 使能； 0: 不使能。

在分支跳转不使能的情况下，后续的 4 个参数无意义。

SMsgBranchToFrmId: 目标微帧的 Id，取值 0~14。

在此消息执行结果满足设定条件后，需要执行的微帧的 Id（通常为事件微帧）。微帧 Id，表示微帧在发送主帧中存放的位置，取值 0~14。

SMsgBranchDataLocation: 检测字在消息中的位置。取值 0~35。

检测字可以是接收消息块中的任何一个字，此参数表示检测字在“BC 接收消息块”中的存放位置。“BC 接收消息块”的存放内容，参考表 3-6 《BC 接收消息块结构》。取值 0~35，始终的，0 表示 BC 发送的第 1 个命令字。

SMsgBranchDataMask: 检测字掩码。

如校验“检测字(16bit)”全部 Bit 位, 那么掩码设为 0xffff。

如校验“检测字(16bit)”部分 Bit 位(如检测 RT 状态字中是否有服务请求位 RT_Status[8]), 在掩码中, 将需要检测的 Bit 位设为 1, 不需要检测的 Bit 位设为 0。

检测过程中, 需要对检测字中的这些指定位置的 Bit 位一一检测, 全部检测成功, 才表示分支条件满足。

SMsgBranchDataExpectation: 检测结果期望值。

预先设定的检测期望值。检测字用指定掩码位的检测结果, 和期望值比较, 如相同, 则表示跳转条件满足, 否则, 不满足。公式表达: 检测“检测字”位与“检测字的掩码”的结果和“检测的期望值”是否相等。

Reserve: 保留参数, 暂未使用。

举例:

BC 发送周期性的查询消息 (BC 从 RT1 取矢量字), 在返回的 RT 状态字中的“服务请求位”置位的情况下, 发送 Id=3 的微帧。那么分支跳转参数设置如下:

```
SMsgBranchEn = 1;           // 分支跳转使能
SMsgBranchToFrmId = 3;      // 跳转至目标微帧的 Id
SMsgBranchDataLocation = 1; // 检测字在消息中的位置
SMsgBranchDataMask = 0x0100; // 检测字掩码
SMsgBranchDataExpectation = 0x0100; // 检测结果期望值
```

3.3.2.3. BC 发送消息结构

typedef struct

```
{
    DWORD SMsgChannel;           // 消息通道选择
    DWORD SMsgFormat;           // 消息消息格式
    DWORD SMsgRetryEn;          // 消息重试使能
    DWORD SMsgGapTime;          // 消息间隔时间
    WORD SMsgBlock[36];         // 发送消息块
    STRUCT_MF1553_BC_SMSG_MODE SMsgMode; // 消息附加功能结构
    STRUCT_MF1553_BC_SMSG_BRANCH SMsgBranch; // 消息分支跳转结构
}STRUCT_MF1553_BC_SMSG;
```

结构参数说明:

SMsgChannel: 消息发送的通道选择。最低位有效, 1: Channel A; 0: Channel B。

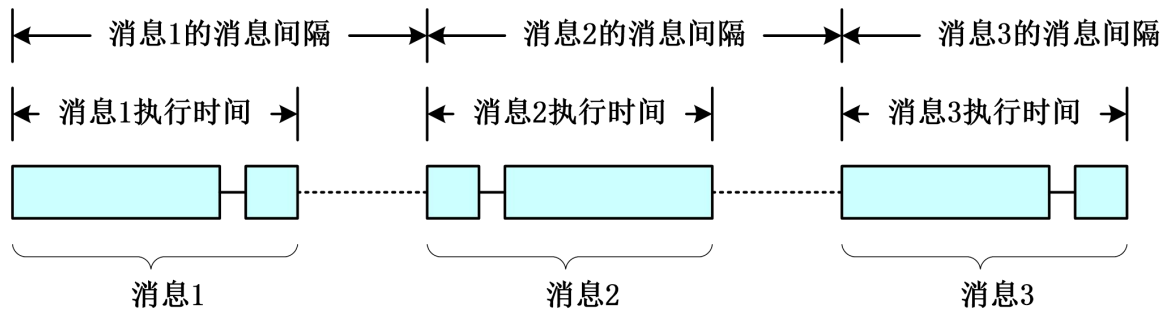
SMsgFormat: 设置消息的格式。低 8 位有效, 消息的格式设置见下表 3-1 《BC 发送消息格式》。

SMsgRetryEn: 消息重试允许位。最低位有效, 1: 允许重试; 0: 不允许重试。

SMsgGapTime: 设置消息间隔时间。分辨率 1us, 取值 0~2³²-1。

消息间隔是指当前消息的起始位置到后一条消息起始位置的时间, 如下图所示。消息类型和数据量的多少决定每条消息的实际执行时间不同 (1M 总线速率下, 经计算, 最长一条消息 (RT→RT 发送 32 个数据字) 发送时间为 730us 左右)。如设定值小于实际执行时间, 则设置无效, 发送完此消息后, 可立刻发送后续消息。如设定值大于实际执行时间, 发送完此消息后,

必须等待间隔时间到达后，才可以发送后续消息。



SMsgBlock: 存放待发送的消息，消息的存放格式见表 3-2 《BC 发送消息块结构》。

说明：最长消息的情况为“RT→RT 发送 32 个数据”（1 个接收命令字、1 个发送命令字、1 个发送 RT 状态字、1 个接收 RT 的状态字），共占用 36 个字的空间，所以硬件为每条消息分配 36 个字空间大小。不足 36 字的消息，从起始位置依次存放。

SMsgMode: “BC 发送消息附加功能结构”指针，存放消息的附加功能。

SMsgBranch: “BC 发送消息分支跳转结构”指针，存放消息的分支跳转功能。

表 3-1 BC 发送消息格式

SMsgFormat [7:0]	格式说明
0x00	BC-to-RT（如果 $T/\overline{R}=0$ ）或者 RT-to-BC (如果 $T/\overline{R}=1$)
0x01	RT-to-RT
0x02	Broadcast
0x03	RT-to-RTs （Broadcast）
0x04	Mode Code
0x05	保留
0x06	Broadcast Mode Code
0x07	保留

表 3-2 BC 发送消息块结构

BC-to-RT传输	RT-to-BC传输	RT-to-RT传输
接收命令字	发送命令字	接收命令字
数据字#1		发送命令字
数据字#2		
...	...	...
数据字#n		
...		
不带数据的模式代码	发送带数据的模式代码	接收带数据的模式代码
模式代码命令字	发送模式代码命令字	接收模式代码命令字
		数据字
...	...	...
广播	RT-to-RTs (广播) 传输	不带数据的广播模式代码
接收命令字 (广播)	接收命令字 (广播)	模式代码命令字 (广播)
数据#1	发送命令字	...
数据#2		带数据的广播模式代码
...	...	模式代码命令字 (广播)
数据字#n		数据字
...		...

3.3.2.4. BC 发送微帧附加功能结构

typedef struct

```
{
    DWORD SFrmModeEndTrgOutEn;           // 微帧执行结束引发脉冲输出使能
    DWORD SFrmModeStartTrgOutEn;         // 微帧执行开始引发脉冲输出使能
    DWORD SFrmModeEndTrgIntEn;           // 微帧执行结束引发中断使能
    DWORD SFrmModeStartTrgIntEn;         // 微帧执行开始引发中断使能
    DWORD SFrmModeRxTriggerDivCnt;       // 外部输入触发微帧执行的分频次数
}STRUCT_MF1553_BC_SFRM_MODE;
```

结构参数说明：

SFrmModeEndTrgOutEn: 微帧执行结束引发脉冲输出使能。最低位有效，1: 使能；0: 不使能。

脉冲输出必须满足以下条件：

- ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数；
- ② 微帧执行结束引发脉冲输出使能。

SFrmModeStartTrgOutEn: 微帧执行开始引发脉冲输出使能。最低位有效，1: 使能；0: 不使能。

脉冲输出必须满足以下条件：

- ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数；
- ② 微帧执行开始引发脉冲输出使能。

SFrmModeEndTrgIntEn: 微帧执行结束引发中断使能。最低位有效，1: 使能；0: 不使能。

中断生成必须满足以下条件：

- ① 总中断使能；

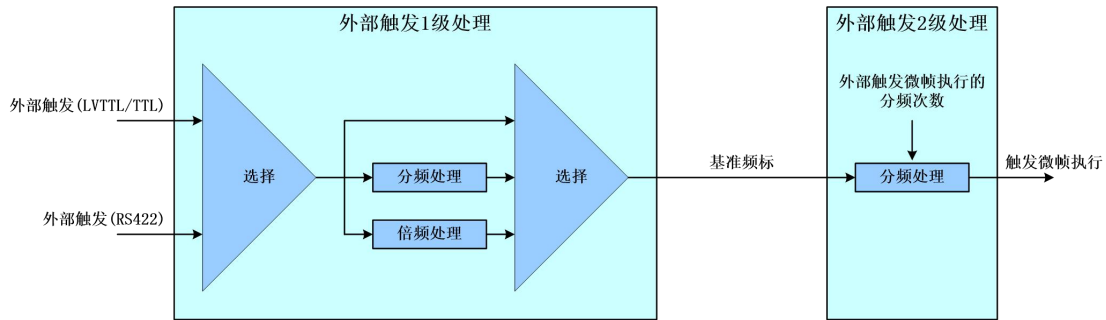
- ② “微帧执行引发中断”的分中断使能
- ③ 微帧执行结束引发中断使能。

SFrmModeStartTrgIntEn: 微帧执行开始引发中断使能。最低位有效，1: 使能；0: 不使能。

中断生成必须满足以下条件：

- ① 总中断使能；
- ② “微帧执行引发中断”的分中断使能
- ③ 微帧执行开始引发中断使能。

SFrmModeRxTriggerDivCnt: 外部输入触发微帧执行的分频次数，此参数对应下图中的右侧部分。



0 值表示外部输入不会触发微帧执行。

非 0 值表示“1 级处理后的基准频标”经过 2 级分频后，触发此微帧执行。

触发成功必须满足以下条件：

- ① 使用 MF1553_BC_RxTriggerMode 函数设置合理的外部触发输入 1 级处理参数；
- ② 设置合理的外部输入 2 级分频次数；
- ③ 外部输入信号有效触发。

举例：外部触发输入为 5ms 微周期的方波信号，上升沿有效，作为 1553B 发送消息的同步信号使用，且要求触发 Id=2 的微帧以 1ms 的周期发送，触发 Id=3 的微帧以 50ms 的周期发送。那么“外部触发 1 级处理”可对原始信号进行 5 倍频（10 倍频也可以），生成周期为 1ms 的基准频标信号。Id=2 的微帧的 2 级分频次数设为 1，Id=3 微帧的 2 级分频次数设为 50。

3.3.2.5. BC 发送微帧结构

typedef struct

```
{
    DWORD SFrmRun;                // 微帧的执行状态
    DWORD SFrmPeriod;             // 微帧发送周期
    DWORD SFrmRepeatCnt;          // 微帧重发次数
    DWORD SFrmGapTime;            // 微帧间隔时间
    DWORD SFrmSMsmsgCnt;          // 微帧中消息数量
    STRUCT_MF1553_BC_SFRM_MODE SFrmMode; // 微帧附加功能结构
    STRUCT_MF1553_BC_SMSMSG SFrmSMsmsg[2047]; // 微帧中消息结构数组
}STRUCT_MF1553_BC_SFRM;
```

结构参数说明：

SFrmRun: 微帧的执行状态。最低位有效，1: 执行；0: 不执行。

参数为 1 时，在运行 BC_Start 函数后，BC 立即执行此微帧；

参数为 0 时，在运行 BC_Start 函数后，BC 跳过此微帧，直到用户调用函数 MF1553_BC_FrmRunCtl 来启动该微帧。

SFrmPeriod: 微帧发送周期。分辨率 1us，取值 $0 \sim 2^{32}-1$ 。

参数为 0 时，为事件微帧；

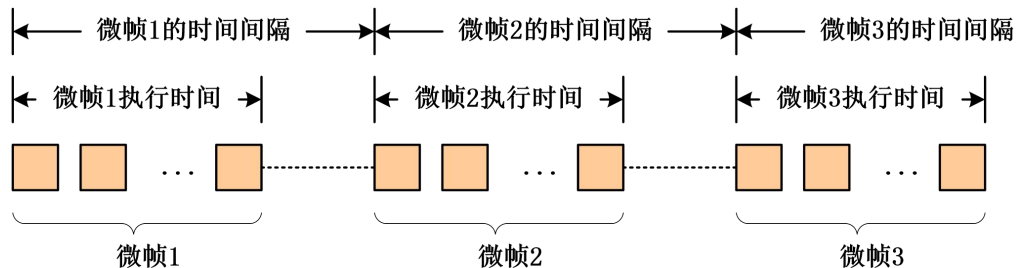
参数为非 0 时，为周期微帧。

SFrmRepeatCnt: 微帧重发次数。取值 $0 \sim 2^{32}-1$ ，且此参数只在周期微帧下有效。

注意，参数为 0 时表示无限次。

SFrmGapTime: 微帧间隔时间。分辨率 1us，取值 $0 \sim 2^{32}-1$ 。

微帧间隔是指当前微帧的起始位置到后一个微帧起始位置的时间，如下图所示。如设定值小于实际执行时间，则设置无效，发送完此微帧后，可立刻发送后续微帧。如设定值大于实际执行时间，发送完此微帧后，必须等待间隔时间到达后，才可以发送后续微帧。



SFrmMsgCnt: 微帧中消息数量。取值 $1 \sim 2047$ 。

SFrmMode: “BC 发送微帧附加功能结构”指针，存放微帧的附加功能。

SFrmSMsg[2047]: “BC 发送消息结构”指针数组，存放微帧中的所有消息。

3.3.2.6. BC 发送微帧参数结构

typedef struct

```
{
    DWORD SFrmRun;                // 微帧的执行状态
    DWORD SFrmPeriod;             // 微帧发送周期
    DWORD SFrmRepeatCnt;          // 微帧重发次数
    DWORD SFrmGapTime;            // 微帧间隔时间
    DWORD SFrmSMsgCnt;            // 微帧中消息数量
    STRUCT_MF1553_BC_SFRM_MODE SFrmMode; // 微帧附加功能结构
}STRUCT_MF1553_BC_SFRMPAR;
```

结构参数说明：

SFrmRun: 微帧的执行状态。最低位有效，1: 执行；0: 不执行。

参数为 1 时，在运行 BC_Start 函数后，BC 立即执行此微帧；

参数为 0 时，在运行 BC_Start 函数后，BC 跳过此微帧，直到用户调用函数 MF1553_BC_FrmRunCtl 来启动该微帧。

SFrmPeriod: 微帧发送周期。分辨率 1us，取值 $0 \sim 2^{32}-1$ 。

参数为 0 时，为事件微帧；

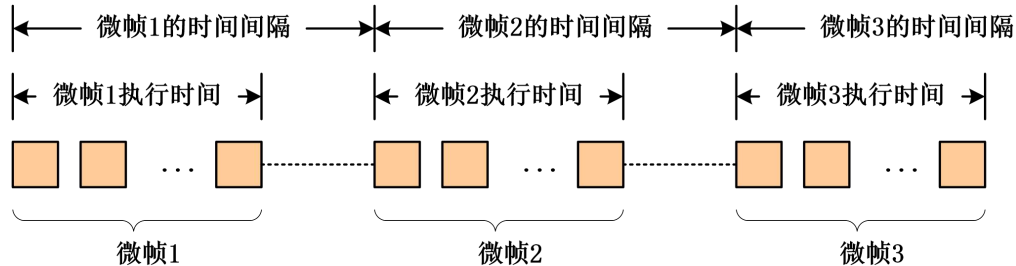
参数为非 0 时，为周期微帧。

SFrmRepeatCnt: 微帧重发次数。取值 $0 \sim 2^{32}-1$ ，且此参数只在周期微帧下有效。

注意，参数为 0 时表示无限次。

SFrmGapTime: 微帧间隔时间。分辨率 1us，取值 $0 \sim 2^{32}-1$ 。

微帧间隔是指当前微帧的起始位置到后一个微帧起始位置的时间，如下图所示。如设定值小于实际执行时间，则设置无效，发送完此微帧后，可立刻发送后续微帧。如设定值大于实际执行时间，发送完此微帧后，必须等待间隔时间到达后，才可以发送后续微帧。



SFrmMsgCnt: 微帧中消息数量。取值 1~2047。

SFrmMode: “BC 发送微帧附加功能结构” 指针，存放微帧的附加功能。

3.3.2.7. BC 发送微帧簇结构

typedef struct

```
{
    DWORD SFrmCnt;           // 微帧数量
    STRUCT_MF1553_BC_SFRM SFrm[15]; // 微帧数组
}STRUCT_MF1553_BC_SMFRM;
```

结构参数说明：

SFrmCnt: 微帧数量。取值 1~15。

SFrm: “BC 发送微帧结构” 指针数组，存放待发送的所有微帧。

3.3.2.8. 接收消息结构

typedef struct

```
{
    DWORD RMsgBlockStatus; // 接收消息块状态描述字
    DWORD RMsgResposeTime0; // 第 1 个 RT 响应时间
    DWORD RMsgResposeTime1; // 第 2 个 RT 响应时间
    DWORD RMsgTimeTagLow;   // 时标低 32 位
    DWORD RMsgTimeTagHigh;  // 时标高 32 位
    WORD RMsgBlock[36];     // 接收消息块
}STRUCT_MF1553_RMSG;
```

结构参数说明：

RMsgBlockStatus: 接收消息块状态描述字。

BC 接收消息块状态描述字格式见表 3-3 《BC 消息块状态描述字》；

RT 接收消息块状态描述字格式见表 3-4 《RT 消息块状态描述字》；

MT 接收消息块状态描述字格式见表 3-5 《MT 消息块状态描述字》。

RMsgResponseTime0: 第 1 个 RT 响应时间。分辨率 0.5us。

RT 响应时间，只在 BC、MT 的接收消息结构中有意义，在 RT 接收消息结构中无意义。

只有在消息执行成功，且消息类型为需要 RT 响应的消息类型，此参数才有意义。因为在部分消息类型中，是不需要 RT 返回状态字。

RMsgResponseTime1: 第 2 个 RT 响应时间。分辨率 0.5us。

RT 响应时间，只在 BC、MT 的接收消息结构中有意义，在 RT 接收消息结构中无意义。

只有在消息执行成功，且消息类型为 RT-RT 消息，此参数才有意义，且表示了接收方 RT 响应时间。因为在其他消息类型中，最多只有 1 个 RT 返回状态字。

RMsgTimeTagLow: 时标低 32 位。分辨率 1us。

RMsgTimeTagHigh: 时标高 32 位。分辨率 1us。

RMsgBlock: 存放接收到的消息。

BC 消息的存放结构见表 3-6 《BC 接收消息块结构》；

RT 消息的存放结构见表 3-7 《RT 接收消息块结构》；

MT 消息的存放结构见表 3-8 《MT 接收消息块结构》。

表 3-3 BC 消息块状态描述字

BIT	名称	说明
15	EOM	BC 消息结束标志，1 有效
14	SOM	BC 消息开始标志，1 有效
13	CHANNEL B/A	消息发送通道指示 0：消息从 A 通道发出 1：消息从 B 通道发出
12	ERROR FLAG	消息出错指示，当出现 8、9、10 位中的任何一种情况时，1 有效
11	STATUS SET	状态置位。状态字低 11 位存在非 0 的位，则置位。1 有效
10	FORMAT ERROR	帧格式错误，1 有效
9	NO RESPONSE TIMEOUT	应答超时，1 有效
8	CMD_ERR	发送的命令有误，1 有效
7	MASKED STATUS SET	屏蔽的状态置位。状态字低 11 位存在非 0 的位，且 BC 控制字的相应的状态屏蔽位为 0，则置位。1 有效
6	RETRY COUNT 1	重试次数 0：0 次 1：1 次 3：2 次
5	RETRY COUNT 0	
4	GOOD DATA BLOCK TRANSFER	发送类型消息传送正常指示，1 有效 注：在 BC 接收 RT 返回的数据传输成功时，此位置位。在 BC 发送给 RT 的数据传输成功时，此位不置位。
3	AD_ERR	RT 状态字地址有误，1 有效
2	LEN_ERR	字长有误，1 有效
1	INCORRECT SYNC TYPE	同步字头错误，1 有效
0	INVALID WORD	同步字、曼彻斯特编码、校验或位长度有误，1 有效

表 3-4 RT 消息块状态描述字

BIT	名称	说明
-----	----	----

BIT	名称	说明
15	保留	
14	保留	
13	CHANNEL B/A	消息发送通道指示， 0：消息从 A 通道发出；1：消息从 B 通道发出
12	ERROR FLAG	消息出错指示，当出现 8、9、10 位中的任何一种情况时，1 有效
11	RT-RT FORMAT	RT-to-RT 消息指示，1 有效
10	FORMAT ERROR	帧格式错误，1 有效
9	NO RESPONSE TIMEOUT	应答超时，1 有效
8	保留	
7	ROB	循环缓冲模式下缓冲区溢出标志
6	CMDILL	非法指令指示
5	WL_ERR	数据长度错误标志
4	SYN_ERR	同步字头错误标志
3	WD_ERR	数据错误标志
2	AD_ERR	状态字地址错误标志
1	CMD2_ERR	RT-RT 传输中第二个命令字错误标志
0	CMD_ERR	命令字错误标志

表 3-5 MT 消息块状态描述字

BIT	名称	说明
15	EOM	消息结束标志，1 有效
14	SOM	消息开始标志，1 有效
13	CHANNEL B/A	消息发送通道指示 0：消息从 A 通道发出 1：消息从 B 通道发出
12	ERROR FLAG	消息出错指示，当出现 8、9、10 位中的任何一种情况时，高有效
11	RT-RT FORMAT	RT-to-RT 消息指示，1 有效
10	FORMAT ERROR	帧格式错误，1 有效
9	NO RESPONSE TIMEOUT	应答超时，1 有效
8	保留	
7	GOOD	消息传送正常指示，1 有效 注：消息（包括各种类型消息）传输成功时，此位置位。
6	ROB	循环缓冲模式下缓冲区溢出标志
5	LEN_ERR	数据长度错误标志
4	SYN_ERR	同步字头错误标志
3	WD_ERR	数据错误标志
2	AD_ERR	状态字地址错误标志
1	CMD2_ERR	RT-to-RT 传输中第二个命令字错误标志
0	CMD_ERR	命令字错误标志

表 3-6 BC 接收消息块结构

BC-to-RT传输		
接收命令字	发送命令字	接收命令字
数据字#1	RT状态字	发送命令字
数据字#2	数据字#1	发送方RT状态字
...	数据字#2	数据字#1
数据字#n	...	数据字#2
RT状态字	数据字#n	...
		数据字#n
		接收方RT状态字
不带数据的模式代码		
模式代码命令字	发送带数据的模式代码	接收带数据的模式代码
RT状态字	发送模式代码命令字	接收模式代码命令字
	RT状态字	数据字
	数据字	RT状态字
广播		
接收命令字 (广播)	RT-to-RTs (广播) 传输	不带数据的广播模式代码
数据#1	接收命令字 (广播)	模式代码命令字 (广播)
数据#2	发送命令字	
...	RT状态字	带数据的广播模式代码
数据字#n	数据字#1	模式代码命令字 (广播)
	数据字#2	数据字
	...	
	数据字#n	

☐ BC发送的内容

☒ RT发送的内容

表 3-7 RT 接收消息块结构

BC-to-RT传输 / 广播		
接收命令字	RT-to-BC传输	其它模式代码
数据字#1	发送命令字	命令字
数据字#2	数据字#1	
...	数据字#2	
数据字#n	...	
...	数据字#n	
	...	

☐ BC发送的内容

☒ RT发送的内容

表 3-8 MT 接收消息块结构

BC-to-RT传输	RT-to-BC传输	RT-to-RT传输
接收命令字	发送命令字	接收命令字
数据字#1	RT状态字	发送命令字
数据字#2	数据字#1	发送方RT状态字
...	数据字#2	数据字#1
数据字#n	...	数据字#2
RT状态字	数据字#n	...
		数据字#n
		接收方RT状态字
不带数据的模式代码	发送带数据的模式代码	接收带数据的模式代码
模式代码命令字	发送模式代码命令字	接收模式代码命令字
RT状态字	RT状态字	数据字
	数据字	RT状态字
广播	RT-to-RTs (广播) 传输	不带数据的广播模式代码
广播命令字	接收命令字 (广播)	模式代码命令字 (广播)
数据#1	发送命令字	
数据#2	RT状态字	
...	数据字#1	
数据字#n	数据字#2	
	...	
	数据字#n	带数据的广播模式代码
		模式代码命令字 (广播)
		数据字

☐ BC发送的内容
☒ RT发送的内容

3.3.2.9. BC 发送消息自动重试结构

```

typedef struct
{
    DWORD RetryEn;           // 消息自动重试使能
    DWORD RetryNum;          // 消息自动重试次数
    DWORD RetryIfMsgError;   // 消息在错误时自动重试使能
    DWORD RetryIfStatusSet;  // 消息在 RT 状态字置位时自动重试使能
    DWORD RetryAlterChannelOnBus1; // 消息第一次自动重试是否改变通道
    DWORD RetryAlterChannelOnBus2; // 消息第二次自动重试是否改变通道
}STRUCT_MF1553_BC_RETRY_CASE;
  
```

结构参数说明:

RetryEn: 消息自动重试使能。最低位有效, 1: 使能; 0: 不使能。

RetryNum: 消息自动重试次数。最低位有效, 1: 重试 2 次; 0: 重试 1 次。

RetryIfMsgError: 消息错误时消息自动重试使能。最低位有效, 1: 使能; 0: 不使能。

此位设为 1 时, 在出现应答超时、格式错误 (状态字地址错误, 数据格式错误, 同步字错误) 情况下, 消息会自动重试。

RetryIfStatusSet: RT 状态字置位时消息自动重试使能。最低位有效, 1: 使能; 0: 不使能。

此位设为 1 时, 在返回的 RT 状态字中, 各状态位 (RtStatus[10]:消息错误位; RtStatus[8]:

服务请求位； RtStatus[7:5]:保留位； RtStatus[4]:广播位； RtStatus[3]:忙位； RtStatus[2]:子系统标志位； RtStatus[0]:终端标志位）存在至少 1 位置位的情况下，消息会自动重试。

RetryAlterChannelOnBus1: 第一次自动重试改变通道。最低位有效，1: 改变通道；0: 不改变通道。

此位设为 1 时，如果消息第一次传输是在通道 A，那么第一次自动重试改为 B 通道传输；
此位设为 0 时，第一次自动重试不改变通道。

RetryAlterChannelOnBus2: 第二次自动重试改变通道。最低位有效，1: 改变通道；0: 不改变通道。

此位设为 1 时，如果消息第一次传输是在通道 A，那么第二次自动重试改为 B 通道传输；
此位设为 0 时，第二次自动重试不改变通道。

3.3.2.10. BC 发送消息/微帧停止结构

typedef struct

```
{
    DWORD MsgStopOnStatusSet;        // 状态置位时停止消息使能
    DWORD MsgStopOnMsgError;         // 消息错误时停止消息使能
    DWORD FrmStopOnStatusSet;        // 状态置位时停止当前微帧使能
    DWORD FrmStopOnMsgError;         // 消息错误时停止当前微帧使能
}STRUCT_MF1553_BC_STOP_CASE;
```

结构参数说明：

MsgStopOnStatusSet: 状态置位时停止消息使能。最低位有效，1:使能；0: 不使能。

此位设为 1 时，如果 RT 状态字中 STATUS_SET 置位(RtStatus[10]:消息错误位； RtStatus[8]:服务请求位； RtStatus[7:5]:保留位； RtStatus[4]:广播位； RtStatus[3]:忙位； RtStatus[2]:子系统标志位； RtStatus[0]:终端标志位)，在执行完本条消息后将停止执行后续消息；但如果消息自动重试使能，那么先自动重试，重试仍有状态置位再停止。

MsgStopOnMsgError: 消息错误时停止消息使能。最低位有效，1:使能；0: 不使能。

此位设为 1 时，如果消息出错(包括字错误、帧格式错误、超时错误)，在执行完本条消息后将停止执行后续消息。但如果消息自动重试使能，那么先自动重试，重试仍有错误再停止。

FrmStopOnStatusSet: 状态置位时停止当前微帧使能。最低位有效，1:使能；0: 不使能。

此位设为 1 时，如果 RT 状态字中 STATUS_SET 置位(RtStatus[10]:消息错误位； RtStatus[8]:服务请求位； RtStatus[7:5]:保留位； RtStatus[4]:广播位； RtStatus[3]:忙位； RtStatus[2]:子系统标志位； RtStatus[0]:终端标志位)，在执行完本帧所有消息后将停止执行后续帧。

FrmStopOnMsgError: 消息错误时停止当前微帧使能。最低位有效，1:使能；0: 不使能。

此位设为 1 时，在“简单消息模式”下，且在帧自动重发方式下，消息出错(包括字错误、帧格式错误、超时错误)时，在执行完本帧所有消息后将停止执行后续帧。

3.3.2.11. RT 状态字置位结构

typedef struct

```
{
    DWORD TerminalFlag;              // 终端标志位
    DWORD SubSystemFlag;            // 子系统标志位
    DWORD ServiceRequest;            // 服务请求位
}
```

```

        DWORD Busy;                // 忙位
        DWORD DynamicBusControl;    // 动态总线控制位
    }STRUCT_MF1553_RT_STATUS_WORD;

```

结构参数说明:

TerminalFlag: 终端标志位。最低位有效，1:置位；0: 未置位。

此位设为 1 时，RT 的状态字中的“Terminal Flag”位将会置 1。

SubSystemFlag: 子系统标志位。最低位有效，1:置位；0: 未置位。

此位设为 1 时，RT 的状态字中的“Subsystem Flag”位将会置 1。

ServiceRequest: 服务请求位。最低位有效，1:置位；0: 未置位。

此位设为 1 时，RT 的状态字中的“Service Request”位将会置 1。

Busy: 忙位。最低位有效，1:置位；0: 未置位。

此位设为 1 时，RT 的状态字中的“Busy”位将会置 1。

DynamicbusControl: 动态总线控制位。最低位有效，1:置位；0: 未置位。

此位设为 1 时，RT 的状态字中的“Dynamic Bus Control Acceptance”位将会置 1。

3.3.2.12. RT 非法命令表结构

```

typedef struct
{
    DWORD CmdTable[32][2][32]; // RT 非法命令表
}STRUCT_MF1553_RT_ILLEGAL_CMD_TABLE;

```

结构参数说明:

CmdTable[I][J][K]: 一个三维的 RT 非法命令表，一维坐标 I 代表 RT 的地址，二位坐标 J 代表发送或接收位 (J=0, 代表接收; J=1, 代表发送)，三维坐标 K 代表 RT 的子地址。数组的值的定义如表 3-9 《RT 非法命令表单元内容》:

表 3-9 RT 非法命令表单元内容

BIT	说明
31	1: 数据量为 31 的命令字非法 0: 合法
30	1: 数据量为 30 的命令字非法 0: 合法
29	1: 数据量为 29 的命令字非法 0: 合法
28	1: 数据量为 28 的命令字非法 0: 合法
27	1: 数据量为 27 的命令字非法 0: 合法
26	1: 数据量为 26 的命令字非法 0: 合法
25	1: 数据量为 25 的命令字非法 0: 合法
24	1: 数据量为 24 的命令字非法 0: 合法
23	1: 数据量为 23 的命令字非法 0: 合法
22	1: 数据量为 22 的命令字非法 0: 合法
21	1: 数据量为 21 的命令字非法; 0: 合法
20	1: 数据量为 20 的命令字非法; 0: 合法
19	1: 数据量为 19 的命令字非法; 0: 合法

BIT	说明
18	1: 数据量为 18 的命令字非法; 0: 合法
17	1: 数据量为 17 的命令字非法; 0: 合法
16	1: 数据量为 16 的命令字非法; 0: 合法
15	1: 数据量为 15 的命令字非法; 0: 合法
14	1: 数据量为 14 的命令字非法; 0: 合法
13	1: 数据量为 13 的命令字非法; 0: 合法
12	1: 数据量为 12 的命令字非法; 0: 合法
11	1: 数据量为 11 的命令字非法; 0: 合法
10	1: 数据量为 10 的命令字非法; 0: 合法
9	1: 数据量为 9 的命令字非法; 0: 合法
8	1: 数据量为 8 的命令字非法; 0: 合法
7	1: 数据量为 7 的命令字非法; 0: 合法
6	1: 数据量为 6 的命令字非法; 0: 合法
5	1: 数据量为 5 的命令字非法; 0: 合法
4	1: 数据量为 4 的命令字非法; 0: 合法
3	1: 数据量为 3 的命令字非法; 0: 合法
2	1: 数据量为 2 的命令字非法; 0: 合法
1	1: 数据量为 1 的命令字非法; 0: 合法
0	1: 数据量为 32 的命令字非法; 0: 合法

举例: CmdTable[3][1][20] = 0x00000001 表示 RT 地址为 3, RT 的子地址为 20, 发送 32 个数据的命令为非法的。

3.3.2.13. RT 子地址发送模式结构

```
typedef struct
{
    DWORD TxMode[32][32];    // RT 子地址发送模式表
}STRUCT_MF1553_RT_TX_MODE;
```

结构参数说明:

TxMode[I][J]: 一个二维的发送模式设置数组, 数组的行坐标 I 代表 RT 的地址, 数组的列坐标 J 代表 RT 的子地址, 数组的值为模式设置位, 取值 0 为单缓冲模式, 此种模式下 RT 发送数据都从发送数据区起始点读取数据发送; 取值 1 为循环缓冲模式, 此种模式下 RT 顺序从发送区读取数据发送, 直到发送数据区指针到达设定的边界时, 发送指针自动回到起始位置继续读数用于发送。

举例: TxMode[1][6] = 1 表示 RT 地址为 1, 子地址为 6 的远程终端的发送采用循环缓冲的模式。

3.3.2.14. MT 命令字过滤表结构

```
typedef struct
```

```
{
    DWORD Filter[32][2];    // MT 命令字过滤表
}STRUCT_MF1553_MT_CMD_FILTER_TABLE;
```

结构参数说明:

Filter[I][J]: 一个二维的命令字过滤表，数组的行坐标 I 代表待监测的远程终端地址，数组的列坐标 J 代表发送或接收位（J=0：接收 J=1：发送），数组的内容值定义如表 3-10:

表 3-10 RT 非法命令表单元内容

BIT	说明	
31	1: 方式代码被监测;	0: 不被监测, 消息丢掉
30	1: 子地址为 30 的远程终端被监测;	0: 不被监测, 消息丢掉
29	1: 子地址为 29 的远程终端被监测;	0: 不被监测, 消息丢掉
28	1: 子地址为 28 的远程终端被监测;	0: 不被监测, 消息丢掉
27	1: 子地址为 27 的远程终端被监测;	0: 不被监测, 消息丢掉
26	1: 子地址为 26 的远程终端被监测;	0: 不被监测, 消息丢掉
25	1: 子地址为 25 的远程终端被监测;	0: 不被监测, 消息丢掉
24	1: 子地址为 24 的远程终端被监测;	0: 不被监测, 消息丢掉
23	1: 子地址为 23 的远程终端被监测;	0: 不被监测, 消息丢掉
22	1: 子地址为 22 的远程终端被监测;	0: 不被监测, 消息丢掉
21	1: 子地址为 21 的远程终端被监测;	0: 不被监测, 消息丢掉
20	1: 子地址为 20 的远程终端被监测;	0: 不被监测, 消息丢掉
19	1: 子地址为 19 的远程终端被监测;	0: 不被监测, 消息丢掉
18	1: 子地址为 18 的远程终端被监测;	0: 不被监测, 消息丢掉
17	1: 子地址为 17 的远程终端被监测;	0: 不被监测, 消息丢掉
16	1: 子地址为 16 的远程终端被监测;	0: 不被监测, 消息丢掉
15	1: 子地址为 15 的远程终端被监测;	0: 不被监测, 消息丢掉
14	1: 子地址为 14 的远程终端被监测;	0: 不被监测, 消息丢掉
13	1: 子地址为 13 的远程终端被监测;	0: 不被监测, 消息丢掉
12	1: 子地址为 12 的远程终端被监测;	0: 不被监测, 消息丢掉
11	1: 子地址为 11 的远程终端被监测;	0: 不被监测, 消息丢掉
10	1: 子地址为 10 的远程终端被监测;	0: 不被监测, 消息丢掉
9	1: 子地址为 9 的远程终端被监测;	0: 不被监测, 消息丢掉
8	1: 子地址为 8 的远程终端被监测;	0: 不被监测, 消息丢掉
7	1: 子地址为 7 的远程终端被监测;	0: 不被监测, 消息丢掉
6	1: 子地址为 6 的远程终端被监测;	0: 不被监测, 消息丢掉
5	1: 子地址为 5 的远程终端被监测;	0: 不被监测, 消息丢掉
4	1: 子地址为 4 的远程终端被监测;	0: 不被监测, 消息丢掉
3	1: 子地址为 3 的远程终端被监测;	0: 不被监测, 消息丢掉
2	1: 子地址为 2 的远程终端被监测;	0: 不被监测, 消息丢掉
1	1: 子地址为 1 的远程终端被监测;	0: 不被监测, 消息丢掉
0	1: 方式代码被监测;	0: 不被监测, 消息丢掉

3.3.2.15. 外部触发输出结构

```
typedef struct
{
    DWORD PulsePhysicalTypeSel; // 触发输出脉冲电平类型选择
    DWORD PulsePolaritySel;     // 触发输出脉冲极性选择
    DWORD PulseWide;            // 触发输出脉宽
}STRUCT_MF1553_TRGTX_MODE;
```

结构参数说明:

PulsePhysicalTypeSel: 触发输出脉冲电平类型选择。最低位有效, 1: RS422; 0: LVTTL/TTL。

PulsePolaritySel: 触发输出脉冲极性选择。最低位有效, 0:正脉冲; 1:负脉冲。

PulseWide: 触发输出脉宽。分辨率 1us, 取值 $0 \sim 2^{32}-1$ 。

3.3.2.16. 外部触发输入结构

```
typedef struct
{
    DWORD PulsePhysicalTypeSel; // 触发输入电平类型选择
    DWORD PulseEffectEdgeSel;   // 触发输入有效沿选择
    DWORD PulseRawFxSel;        // 选择触发输入原始信号作为基准频标
    DWORD PulseDivFxSel;        // 选择触发输入分频处理后的信号作为基准频标
    DWORD PulseMulFxSel;        // 选择触发输入倍频处理后的信号作为基准频标
    DWORD PulseFxCnt;           // 分频倍数/倍频倍数
    DWORD PulseMulFxClkCnt;     // 触发输入倍频处理时, 单周期内系统时钟计数量
}STRUCT_MF1553_TRGRX_MODE;
```

结构参数说明:

PulsePhysicalTypeSel: 触发输入电平类型选择。最低位有效, 1: RS422; 0: LVTTL/TTL。

PulseEffectEdgeSel: 触发输入有效沿选择。最低位有效, 1: 下降沿; 0: 上升沿。

PulseRawFxSel: 选择触发输入原始信号作为基准频标。最低位有效, 1: 使能; 0: 不使能。

此位设为 1 时, 将选择原始信号作为基准频标。

PulseDivFxSel: 选择触发输入分频处理后的信号作为基准频标。最低位有效, 1: 使能; 0: 不使能。

此位设为 1 时, 将选择分频处理后的信号作为基准频标。

PulseMulFxSel: 选择触发输入倍频处理后的信号作为基准频标。最低位有效, 1: 使能; 0: 不使能。

此位设为 1 时, 将选择倍频处理后的信号作为基准频标。

PulseFxCnt: 分频倍数/倍频倍数。取值 $1 \sim 2^{32}-1$ 。

注意: 只有在 PulseDivFxSel=1, 或 PulseMulFxSel=1 时, 此参数才有意义。

PulseMulFxClkCnt: 触发输入倍频处理时, 单周期内系统时钟计数量。取值 $1 \sim 2^{32}-1$ 。

注意: 只有在 PulseMulFxSel=1 时, 此参数才有意义。

检测原始输入信号有效边沿(可选上升沿或下降沿), 生成第 1 个触发脉冲。以此为起始, 计数系统时钟 PulseMulFxClkCnt 次, 生成第 2 个触发脉冲, 然后计数器重新计数 MulFxClkCnt 次, 生成第 3 个触发脉冲, 重复此操作, 直到生成的脉冲数量在原始信号一个周期内(原始信号

两个有效跳变沿之间)达到倍频值 `PulseFxCnt`。这样，在原始信号一个周期内，就生成了 `FxCnt` 个触发脉冲，达到的倍频的目的。

倍频时，只要参数 `PulseFxCnt`、`PulseMulFxCnt` 设置合理，可将误差控制得很小。误差只在原始信号 1 个周期内累计，周期更替时误差清 0。

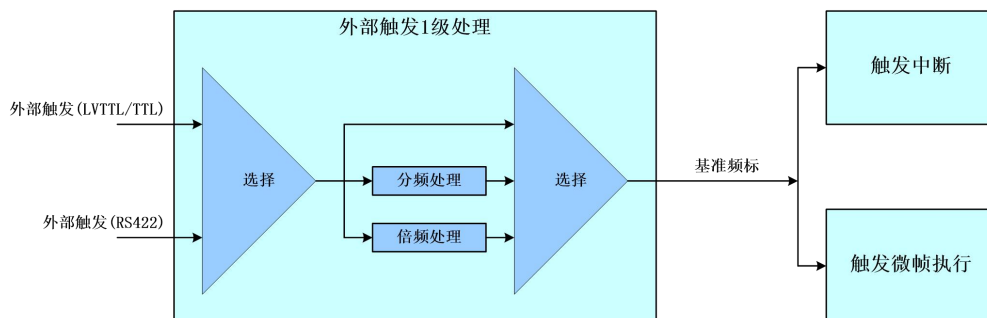
这里提到的基准频标有 3 个源可供选择：

- ① 触发输入的原始信号；
- ② 触发输入分频处理后的信号；
- ③ 触发输入倍频处理后的信号。

只能选择其中 1 种，不可以同时选择。

基准频标有两种用途：

- ① 引发中断，通知上位机；
- ② 用于触发微帧的执行。



举例：需求在外部触发信号是 5ms 为周期的信号，需要倍频 5 次，每次以 1ms 触发执行 1 次指定微帧。由于系统时钟 48Mhz，1ms 需要系统时钟计数 $48000=0x0000BB80$ 次，参数需要设置如下：
`PulseFxCnt=0x00000005`、`PulseMulFxCnt=0x0000BB80`。

3.3.3. 驱动程序函数接口说明

3.3.3.1. MF1553_Open

函数原型： `int MF1553_Open (DevHandle *phMF1553, DWORD CardId);`

函数功能： 打开板卡，并分配板卡资源。

参数说明： `phMF1553`：板卡句柄的指针。

`CardId`：板卡编号，取值 0~255。若 PC 机中同时插 256 块同型号板卡，板卡的编号按板卡所在的插槽离 CPU 的距离由近到远以次编号 0、1...255。若 PC 机中同时只插一块板，板卡编号为 0。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.2. MF1553_Close

函数原型: int MF1553_Close (DevHandle hMF1553);

函数功能: 关闭板卡, 释放板卡资源。

参数说明: hMF1553: 板卡的句柄。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.3. MF1553_Reset

函数原型: int MF1553_Reset (DevHandle hMF1553);

函数功能: 板卡复位函数, 全局复位整个板卡。

参数说明: hMF1553: 板卡的句柄。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.4. MF1553_ResetChannel

函数原型: int MF1553_ResetChannel (DevHandle hMF1553, DWORD ch);

函数功能: 复位指定通道函数。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.5. MF1553_SpeedSel

函数原型: int MF1553_SpeedSel (DevHandle hMF1553, DWORD ch, DWORD Speed4MX1M)

函数功能: 1553B 总线速率选择。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Speed4MX1M: 总线速率选择。1: 4Mbps 速率; 0: 1Mbps 速率。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.6. MF1553_AddTimeTag

函数原型: int MF1553_AddTimeTag (DevHandle hMF1553, DWORD ch, DWORD Enable);

函数功能: 启动或停止时标模式。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Enable: 时标使能位。1: 使能; 0: 不使能。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.7. MF1553_SetTimeTag

函数原型: int MF1553_SetTimeTag (DevHandle hMF1553, DWORD ch, DWORD TimeTagLow,

DWORD TimeTagHigh);

函数功能: 设置 64 位时标起始值, 分辨率 1us。只在时标模式使能的情况下, 设置才有效。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

TimeTagLow: 时标起始值低 32 位。

TimeTagHigh: 时标起始值高 32 位。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.8. MF1553_GetTimeTag

函数原型: int MF1553_GetTimeTag (DevHandle hMF1553, DWORD ch, DWORD *TimeTagLow, DWORD *TimeTagHigh);

函数功能: 获取 64 位当前时标值, 分辨率 1us。时标模式不使能时, 获取的时标始终为常值 0。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

TimeTagLow: 当前时标值低 32 位。

TimeTagHigh: 当前时标值高 32 位。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.9. MF1553_SetResponseTimeout

函数原型: int MF1553_SetResponseTimeout (DevHandle hMF1553, DWORD ch, DWORD Timeout);

函数功能: 设置 RT 应答超时时间。硬件启动默认值为 20, 表示 10us。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Timeout: 超时时间, 分辨率 0.5us。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.10. MF1553_BC_Init

函数原型: int MF1553_BC_Init (DevHandle hMF1553, DWORD ch);

函数功能: 初始化总线控制器。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.11. MF1553_BC_CfgWord

函数原型: int MF1553_BC_CfgWord (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_BC_RETRY_CASE *RetryCase, STRUCT_MF1553_BC_STOP_CASE *StopCase);

函数功能: 配置 BC 消息自动重试机制, BC 消息停止机制, BC 微帧停止机制。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RetryCase: “BC 发送消息自动重试结构” 指针，内容参考相应结构体说明。

StopCase: “BC 发送消息/微帧停止结构” 指针，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.12. MF1553_BC_WriteMsg

函数原型: int MF1553_BC_WriteMsg (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, STRUCT_MF1553_BC_SMSG *Msg);

函数功能: 写入 1 条消息到硬件缓存。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息 Id。取值 0~2046。

Msg: “BC 发送消息结构” 指针，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.13. MF1553_BC_WriteMsgs

函数原型: int MF1553_BC_WriteMsgs (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, DWORD MsgCnt, STRUCT_MF1553_BC_SMSG *Msgs);

函数功能: 写入消息组到硬件缓存。注意，写消息组时，不能跨微帧边界。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息组中起始消息的 Id。取值 0~2046。

MsgCnt: 写入消息数量。取值 1~2047。

Msgs: “BC 发送消息结构” 指针，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.14. MF1553_BC_WriteFrmPar

函数原型: int MF1553_BC_WriteFrmPar (DevHandle hMF1553, DWORD ch, DWORD FrmId, STRUCT_MF1553_BC_SFRMPAR *FrmPar);

函数功能: 写入微帧参数（不包含微帧中消息内容）到硬件缓存。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

FrmPar: “BC 发送微帧参数结构” 指针，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.15. MF1553_BC_WriteFrm

函数原型: int MF1553_BC_WriteFrm (DevHandle hMF1553, DWORD ch, DWORD FrmId,

STRUCT_MF1553_BC_SFRM *Frm);

函数功能：写入 1 个微帧到硬件缓存。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

Frm: “BC 发送微帧结构”指针，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.16. MF1553_BC_UpdateFrm

函数原型： int MF1553_BC_UpdateFrm (DevHandle hMF1553, DWORD ch, DWORD FrmId, STRUCT_MF1553_BC_SFRM *Frm);

函数功能：更新 1 个微帧到硬件缓存。此函数可以让新更新的微帧与原微帧无缝衔接，实际上在硬件内部完成了微帧消息组 A、B 两区域间无缝转换。无缝更新，保证了动态更新消息时，发送微帧的完整性。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

Frm: “BC 发送微帧结构”指针，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.17. MF1553_BC_AddEndOfFrmList

函数原型： int MF1553_BC_AddEndOfFrmList (DevHandle hMF1553, DWORD ch, DWORD FrmCnt);

函数功能：添加微帧链表结束符。在 BC 启动前，必须设置合理的链表结束符。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmCnt: 微帧数量，也可以理解为微帧链表结束符位置。取值 1~15。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.18. MF1553_BC_GetEndOfFrmList

函数原型： int MF1553_BC_GetCntOfFrmList (DevHandle hMF1553, DWORD ch, DWORD *FrmCnt);

函数功能：获取微帧链表结束符。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmCnt: 微帧数量，也可以理解为微帧链表结束符位置。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.19. MF1553_BC_FrmRunCtl

函数原型： int MF1553_BC_FrmRunCtl (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD Run);

函数功能：控制 BC 微帧的执行状态。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0 ~ 14。

Run: 微帧执行状态位。最低位有效, 1: 执行启动; 0: 执行停止。

在 BC 正在运行的情况下 (BC 的运行由函数 MF1553_BC_Start 启动), 运行此函数且 Run 参数为 1, 则会使编号为 FrmId 的微帧开始执行。如果此微帧是事件微帧 (周期为 0), 则运行此函数 (Run 参数为 1) 后会将此微帧立即执行 1 次。如果此微帧是周期微帧 (周期不为 0), 则运行此函数 (Run 参数为 1) 后会启动此微帧的周期运行。

在 BC 正在运行的情况下 (BC 的运行由函数 MF1553_BC_Start 启动), 运行此函数且 Run 参数为 0, 则会使编号为 FrmId 的微帧停止执行。如果此微帧是事件微帧 (周期为 0), 则运行此函数 (Run 参数为 0) 不起任何作用。如果此微帧是周期微帧 (周期不为 0), 则运行此函数 (Run 参数为 0) 后会停止此微帧的周期运行。

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.20. MF1553_BC_GetMsgInfo

函数原型：int MF1553_BC_GetMsgInfo (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, STRUCT_MF1553_BC_SMSG *Msg);

函数功能：获取硬件缓存中的 1 条消息。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息 Id。取值 0~2046。

Msg: “BC 发送消息结构” 指针, 内容参考相应结构体说明。

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.21. MF1553_BC_GetMsgsInfo

函数原型：MF1553_BC_GetMsgsInfo (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, DWORD MsgCnt, STRUCT_MF1553_BC_SMSG *Msgs);

函数功能：获取硬件缓存中的消息组。注意, 读消息组时, 不能跨微帧边界。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息组中起始消息的 Id。取值 0~2046。

MsgCnt: 读出消息数量。取值 1~2047。

Msgs: “BC 发送消息结构” 指针, 内容参考相应结构体说明。

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.22. MF1553_BC_GetFrmParInfo

函数原型：MF1553_BC_GetFrmParInfo (DevHandle hMF1553, DWORD ch, DWORD FrmId,

STRUCT_MF1553_BC_SFRMPAR *FrmPar);

函数功能：获取硬件缓存中的指定微帧参数（不包含微帧中消息内容）。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

FrmPar: “BC 发送微帧参数结构”指针，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.23. MF1553_BC_GetFrmInfo

函数原型：MF1553_BC_GetFrmInfo (DevHandle hMF1553, DWORD ch, DWORD FrmId, STRUCT_MF1553_BC_SFRM *Frm);

函数功能：获取硬件缓存中的指定微帧。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

Frm: “BC 发送微帧结构”指针，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.24. MF1553_BC_WriteMsgDataBlock

函数原型：MF1553_BC_WriteMsgDataBlock (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, DWORD Offset, DWORD WordCnt, WORD *DataBuf);

函数功能：修改 BC 发送类型消息中的数据区。注意参数需合理设置，不能超过数据区边界，否则操作失败。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息 Id。取值 0~2046。

Offset: 数据区偏移量，取值 0~31。

WordCnt: 写入的数据字的数量，取值 1~32。

DataBuf: 存放待写入的数据。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.25. MF1553_BC_Start

函数原型：int MF1553_BC_Start (DevHandle hMF1553, DWORD ch);

函数功能：BC 启动运行。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.26. MF1553_BC_Stop

函数原型: int MF1553_BC_Stop (DevHandle hMF1553, DWORD ch);

函数功能: BC 停止运行。执行此函数，使 BC 由运行状态进入停止状态，立即停止处理所有消息。如执行此函数时，BC 正在处理某条消息，则等待处理完当前的消息，然后停止。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.27. MF1553_BC_MsgBufferClear

函数原型: int MF1553_BC_MsgBufferClear (DevHandle hMF1553, DWORD ch);

函数功能: 清空硬件上 BC 接收消息缓存。在接收消息之前执行此操作，以避免读出硬件缓存中之前残留的消息。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.28. MF1553_BC_GetLastMsgId

函数原型: int MF1553_BC_GetLastMsgId (DevHandle hMF1553, DWORD ch, DWORD *FrmId, DWORD *MsgId);

函数功能: 获取最近一条 BC 执行完毕的消息的 Id，包括此消息所在微帧的 Id。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

MsgId: 消息 Id。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.29. MF1553_BC_GetLastFrmId

函数原型: int MF1553_BC_GetLastFrmId (DevHandle hMF1553, DWORD ch, DWORD *FrmId);

函数功能: 获取最近一个 BC 执行完毕的微帧的 Id。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.30. MF1553_BC_GetIntMsgId

函数原型: int MF1553_BC_GetIntMsgId (DevHandle hMF1553, DWORD ch, DWORD *FrmId, DWORD *MsgId);

函数功能: 获取最近一条生成中断的消息的 Id，包括此消息所在微帧的 Id。

在多条消息都可能产生中断的情况下，在中断服务程序中执行此函数，读取产生中断的消息的 Id，以判断是哪一条消息生成的中断。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

MsgId: 消息 Id。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.31. MF1553_BC_GetIntFrmId

函数原型：int MF1553_BC_GetIntFrmId (DevHandle hMF1553, DWORD ch, DWORD *FrmId);

函数功能：获取最近一个生成中断的微帧的 Id。

在多个微帧都可能产生中断的情况下，在中断服务程序中执行此函数，读取产生中断的微帧的 Id，以判断是哪一个微帧生成的中断。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.32. MF1553_BC_GetMsgCntTotal

函数原型：int MF1553_BC_GetMsgCntTotal (DevHandle hMF1553, DWORD ch, DWORD *MsgCntTotal);

函数功能：获取从板卡复位开始到目前 BC 总的执行消息数量。包括用户禁止接收的消息，未来及读取的消息等。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgCntTotal: 消息数量。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.33. MF1553_BC_GetMsgNumNewly

函数原型：int MF1553_BC_GetMsgNumNewly (DevHandle hMF1553, DWORD ch, DWORD *MsgNum);

函数功能：获取 BC 新消息数量。即硬件处理完成、软件未来得及读取的消息数量。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.34. MF1553_BC_ReadMsg

函数原型：int MF1553_BC_ReadMsg (DevHandle hMF1553, DWORD ch, DWORD *FrmId, DWORD

*MsgId, STRUCT_MF1553_RMSG *Msg);

函数功能：单个读取 BC 处理完的消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读出消息也是按 FIFO 机制读取。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

MsgId: 消息 Id。

Msg: “接收消息结构”指针，内容参考相应结构体说明。

返回值：读出消息，返回值为 1；未读出消息，返回值为 2；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.35. MF1553_BC_ReadMsgs

函数原型：int MF1553_BC_ReadMsgs (DevHandle hMF1553, DWORD ch, DWORD MsgNum, DWORD *FrmIds, DWORD *MsgIds, STRUCT_MF1553_RMSG *Msgs);

函数功能：批量读取 BC 处理完的消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读出消息也是按 FIFO 机制读取。执行此函数之前，通常先执行函数 MF1553_BC_GetMsgNumNewly 以获取消息数量。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

FrmIds: 微帧 Id 组。

MsgIds: 消息 Id 组。

Msgs: “接收消息结构”指针组，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.36. MF1553_BC_ReadLastMsg

函数原型：int MF1553_BC_ReadLastMsg (DevHandle hMF1553, DWORD ch, DWORD *FrmId, DWORD *MsgId, STRUCT_MF1553_RMSG *Msg);

函数功能：获取 BC 最近处理的那条消息

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。

MsgId: 消息 Id。

Msg: “消息接收结构”指针，内容参考相应结构体说明。

返回值：操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.37. MF1553_BC_GetMsgRefresh

函数原型：int MF1553_BC_GetMsgRefresh (DevHandle hMF1553, DWORD ch, DWORD FrmId, DWORD MsgId, STRUCT_MF1553_RMSG *Msg);

函数功能：用户指定消息 Id 来读取 BC 消息。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FrmId: 微帧 Id。取值 0~14。

MsgId: 消息 Id。取值 0~2046。

Msg: “消息接收结构”指针, 内容参考相应结构体说明。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.38. MF1553_BC_IsRunning

函数原型: int MF1553_BC_IsRunning (DevHandle hMF1553, DWORD ch);

函数功能: 获取 BC 运行状态

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: BC 正在运行, 返回值为 1; BC 停止运行, 返回值为 2; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.39. MF1553_BC_TxTriggerMode

函数原型: int MF1553_BC_TxTriggerMode (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_TRGTX_MODE *Mode);

函数功能: 外部触发输出模式设置。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Mode: “外部触发输出结构”指针, 内容参考相应结构体说明。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.40. MF1553_BC_TxTriggerCmd

函数原型: int MF1553_BC_TxTriggerCmd (DevHandle hMF1553, DWORD ch);

函数功能: 强制外部触发输出命令。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.41. MF1553_BC_RxTriggerMode

函数原型: int MF1553_BC_RxTriggerMode (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_TRGRX_MODE *Mode);

函数功能: 外部触发输入模式设置。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Mode: “外部触发输入结构”指针, 内容参考相应结构体说明。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.42. MF1553_RT_Init

函数原型: int MF1553_RT_Init (DevHandle hMF1553, DWORD ch);

函数功能: 初始化远程终端。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.43. MF1553_RT_RTAddr

函数原型: int MF1553_RT_RTAddr (DevHandle hMF1553, DWORD ch, DWORD AddrVector);

函数功能: RT 地址使能。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

AddrVector: RT 地址使能数组。位定义见表 3-11 《RT 地址使能表》。

返回值: 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

表 3-11 RT 地址使能表

BIT	说明
31	保留
30	1: 地址为 30 的远程终端使能; 0: 不使能
29	1: 地址为 29 的远程终端使能; 0: 不使能
28	1: 地址为 28 的远程终端使能; 0: 不使能
27	1: 地址为 27 的远程终端使能; 0: 不使能
26	1: 地址为 26 的远程终端使能; 0: 不使能
25	1: 地址为 25 的远程终端使能; 0: 不使能
24	1: 地址为 24 的远程终端使能; 0: 不使能
23	1: 地址为 23 的远程终端使能; 0: 不使能
22	1: 地址为 22 的远程终端使能; 0: 不使能
21	1: 地址为 21 的远程终端使能; 0: 不使能
20	1: 地址为 20 的远程终端使能; 0: 不使能
19	1: 地址为 19 的远程终端使能; 0: 不使能
18	1: 地址为 18 的远程终端使能; 0: 不使能
17	1: 地址为 17 的远程终端使能; 0: 不使能
16	1: 地址为 16 的远程终端使能; 0: 不使能
15	1: 地址为 15 的远程终端使能; 0: 不使能
14	1: 地址为 14 的远程终端使能; 0: 不使能
13	1: 地址为 13 的远程终端使能; 0: 不使能
12	1: 地址为 12 的远程终端使能; 0: 不使能
11	1: 地址为 11 的远程终端使能; 0: 不使能
10	1: 地址为 10 的远程终端使能; 0: 不使能
9	1: 地址为 9 的远程终端使能; 0: 不使能

BIT	说明
8	1: 地址为 8 的远程终端使能; 0: 不使能
7	1: 地址为 7 的远程终端使能; 0: 不使能
6	1: 地址为 6 的远程终端使能; 0: 不使能
5	1: 地址为 5 的远程终端使能; 0: 不使能
4	1: 地址为 4 的远程终端使能; 0: 不使能
3	1: 地址为 3 的远程终端使能; 0: 不使能
2	1: 地址为 2 的远程终端使能; 0: 不使能
1	1: 地址为 1 的远程终端使能; 0: 不使能
0	1: 地址为 0 的远程终端使能; 0: 不使能

3.3.3.44. MF1553_RT_TxMode

函数原型： int MF1553_RT_TxMode (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_RT_TX_MODE *TxMode);

函数功能： 设置 RT 子地址数据发送模式

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

TxMode: “RT 子地址发送模式结构”指针, 内容参考相应结构体说明。

返回值： 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.45. MF1553_RT_SetTimeTagOnSync

函数原型： int MF1553_RT_SetTimeTagOnSync (DevHandle hMF1553, DWORD ch, DWORD ClearTimeTagOnSyncEn, DWORD LoadTimeTagOnSyncEn);

函数功能： 时标在接收到同步方式指令时, 处理方式设置。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

ClearTimeTagOnSyncEn: 在接收到不带数据的同步指令时, 时标清 0 使能。

参数设为 1 时, 时标在接收到不带数据的同步方式指令后, 时标清 0。

参数设为 0 时, 时标在接收到不带数据的同步方式指令后, 时标不清 0。

LoadTimeTagOnSyncEn: 在接收到带数据的同步指令时, 加载时标使能。

参数设为 1 时, 时标在接收到带数据的同步方式指令后, 将命令中的数据加载到时标中作为初始值。

参数设为 0 时, 时标在接收到带数据的同步方式指令后, 不加载。

返回值： 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.46. MF1553_RT_SetIllCmdTable

函数原型： int MF1553_RT_SetIllCmdTable (DevHandle hMF1553, DWORD ch, DWORD IllCmdTableEn, DWORD IllCmdRcvDataEn, STRUCT_MF1553_RT_ILLEGAL_CMD_TABLE *IllCmdTable);

函数功能：RT 非法命令表设置。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

IllCmdTableEn: 非法命令表使能。1: 使能; 0: 不使能。

IllCmdRcvDataEn: 接收到非法命令时, 数据接收使能。此参数在非法指令表不使能下无意义。

参数设为 1 时, 接收到非法命令, 对于接收数据类型的命令, RT 将数据写入接收数据缓存, 并将状态字中的消息差错位置位; 对于发送数据类型的命令, RT 只发送状态字 (如果需要发送), 且将状态字中的消息差错位置位, 不发送数据 (如果需要发送)。

参数设为 0 时, 接收到非法命令, 对于接收数据类型的命令, RT 不会将数据写入接收数据缓存, 并将状态字中的消息差错位置位; 对于发送数据类型的命令, RT 只发送状态字 (如果需要发送), 且将状态字中的消息差错位置位, 不发送数据 (如果需要发送)。

IllCmdTable: “RT 非法命令表结构” 指针, 内容参考相应结构体说明。

返回值：空

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.47. MF1553_RT_SetRTStatusWord

函数原型：int MF1553_RT_SetRTStatusWord (DevHandle hMF1553, DWORD ch, DWORD RTAddr, STRUCT_MF1553_RT_STATUS_WORD *RTStatusWord);

函数功能：RT 的状态字置位。

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 的地址。取值 0~30。

RTStatusWord: “RT 状态字置位结构” 指针, 内容参考相应结构体说明。

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.48. MF1553_RT_VectorWordAutoClrEn

函数原型：int MF1553_RT_VectorWordAutoClrEn (DevHandle hMF1553, DWORD ch, DWORD Enable);

函数功能：硬件自动清除 RT 矢量字使能。在此位使能的情况下, 矢量字的 D0~D15 位分别代表 RT 的子地址 1~16 中是否有已准备好的发送数据。RT 子地址单/双缓冲的情况下, RT 只需处理完 1 条发送消息, 硬件就会自动将矢量字对应的位清 0; RT 子地址循环缓冲的情况下, 待发送消息中的数据指针位置到达回卷深度后, 硬件才会自动将矢量字对应的位清 0。在硬件自动清 RT 矢量字不使能的情况下, 硬件不会自动改变矢量字, 完全由软件设置

参数说明：hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Enable: 自动清除 RT 矢量字使能位。最低位有效。1: 使能; 0: 不使能。

返回值：操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.49. MF1553_RT_SetVectorWord

函数原型： int MF1553_RT_SetVectorWord (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD VectorWord);

函数功能： 设置 RT 矢量字。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

VectorWord: RT 矢量字。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.50. MF1553_RT_GetVectorWord

函数原型： int MF1553_RT_GetVectorWord (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD *VectorWord);

函数功能： 获取 RT 矢量字。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

VectorWord: RT 矢量字。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.51. MF1553_RT_SetBitWord

函数原型： int MF1553_RT_SetBitWord (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD BitWord);

函数功能： 设置 RT 自检字。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

BitWord: RT 自检字。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.52. MF1553_RT_WriteMsgDataBlock

函数原型： int MF1553_RT_WriteMsgDataBlock (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD SubAddr, DWORD WordCnt, WORD *DataBuf);

函数功能： 设置 RT 子地址发送数据区。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

SubAddr: RT 子地址。取值 1~30。

WordCnt: 数据量。

子地址发送为单/双缓冲模式下，此参数取值 1~32。

子地址发送为循环缓冲模式下，此参数取值 1~4096。

DataBuf: 数据缓存。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.53. MF1553_RT_SendLoopBackCmd

函数原型: int MF1553_RT_SendSaLoopBackCmd (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD SubAddr);

函数功能: 强制 RT 子地址循环发送数据区指针，回到此数据区起始位置。只在相应的发送子地址为“循环缓冲方式”下有效。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

SubAddr: RT 子地址。取值 1~30。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.54. MF1553_RT_SetSaDepth

函数原型: int MF1553_RT_SetSaDepth (DevHandle hMF1553, DWORD ch, DWORD RTAddr, DWORD SubAddr, DWORD Depth);

函数功能: 设置 RT 子地址循环发送数据区的指针回卷深度。只在发送子地址为“循环缓冲方式”下有效。

例如，在数据缓冲区大小 Size 为 4096 Word 情况下，当指针回卷深度不设置（默认为 0）或设置为 0，数据区发送到 4096 个数据后，指针回卷，从数据区起始位置开始继续发送；当指针回卷深度设置为 200，数据区发送到 200 个数据后，指针回卷，从数据区起始位置开始继续发送，而不是从 201 的位置开始发送。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

RTAddr: RT 地址。取值 0~30。

SubAddr: RT 子地址。取值 1~30。

Depth: 循环发送缓冲区指针回卷深度。取值 0~4095。参数设为 0 时，表示回卷深度为满量程 4096 个字。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.55. MF1553_RT_MsgBufferClear

函数原型: int MF1553_RT_MsgBufferClear (DevHandle hMF1553, DWORD ch);

函数功能: 清空硬件上 RT 接收消息缓存。在接收消息之前执行此操作，以避免读出硬件缓存中之前残留的消息。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.56. MF1553_RT_GetMsgNumRxNewly

函数原型： int MF1553_RT_GetMsgNumRxNewly (DevHandle hMF1553, DWORD ch, DWORD *MsgNum);

函数功能： 获取 RT 接收类型新消息数量。即硬件处理完成、软件未来得及读取的消息数量。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

返回值： 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.57. MF1553_RT_GetMsgNumTxNewly

函数原型： int MF1553_RT_GetMsgNumTxNewly (DevHandle hMF1553, DWORD ch, DWORD *MsgNum);

函数功能： 获取 RT 发送类型新消息数量。即硬件处理完成、软件未来得及读取的消息数量。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

返回值： 操作成功, 返回值为 0; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.58. MF1553_RT_ReadMsgRx

函数原型: int MF1553_RT_ReadMsgRx (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_RMSG *Msg);

函数功能： 单个读取 RT 处理完的接收类型消息。硬件将处理完的消息按 FIFO 机制写入缓存, 所以软件读出消息也是按 FIFO 机制读取。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Msg: “接收消息结构” 指针, 内容参考相应结构体说明。

返回值： 读出消息, 返回值为 1; 未读出消息, 返回值为 2; 操作失败, 返回值为-1; 参数错误, 返回值为-2。

3.3.3.59. MF1553_RT_ReadMsgsRx

函数原型： int MF1553_RT_ReadMsgsRx (DevHandle hMF1553, DWORD ch, DWORD MsgNum, STRUCT_MF1553_RMSG *Msgs);

函数功能： 批量读取 RT 处理完的接收类型消息。硬件将处理完的消息按 FIFO 机制写入缓存, 所以软件读出消息也是按 FIFO 机制读取。执行此函数之前, 通常先执行函数 MF1553_RT_GetMsgNumRxNewly 以获取消息数量。

参数说明： hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

Msgs: “接收消息结构”指针组，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.60. MF1553_RT_ReadMsgTx

函数原型: int MF1553_RT_ReadMsgTx (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_RMSG *Msg);

函数功能: 单个读取 RT 处理完的发送类型消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读出消息也是按 FIFO 机制读取。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Msg: “接收消息结构”指针，内容参考相应结构体说明。

返回值: 读出消息，返回值为 1；未读出消息，返回值为 2；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.61. MF1553_RT_ReadMsgsTx

函数原型: int MF1553_RT_ReadMsgsTx (DevHandle hMF1553, DWORD ch, DWORD MsgNum, STRUCT_MF1553_RMSG *Msgs);

函数功能: 批量读取 RT 处理完的发送类型消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读出消息也是按 FIFO 机制读取。执行此函数之前，通常先执行函数 MF1553_RT_GetMsgNumTxNewly 以获取消息数量。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

Msgs: “接收消息结构”指针组，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.62. MF1553_MT_Init

函数原型: int MF1553_MT_Init (DevHandle hMF1553, DWORD ch);

函数功能: 初始化总线监视器。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.63. MF1553_MT_SetCmdFilterTable

函数原型: int MF1553_MT_SetCmdFilterTable (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_MT_CMD_FILTER_TABLE *FilterTable);

函数功能: 设置 MT 命令字过滤表。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

FilterTable: “MT 命令字过滤表结构” 指针组，内容参考相应结构体说明。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.64. MF1553_MT_Start

函数原型: int MF1553_MT_Start (DevHandle hMF1553, DWORD ch);

函数功能: MT 启动监听。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.65. MF1553_MT_Stop

函数原型: int MF1553_MT_Stop (DevHandle hMF1553, DWORD ch);

函数功能: MT 停止监听。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.66. MF1553_MT_MsgBufferClear

函数原型: int MF1553_MT_MsgBufferClear (DevHandle hMF1553, DWORD ch);

函数功能: 清空硬件上 MT 接收消息缓存。在接收消息之前执行此操作，以避免读出硬件缓存中之前残留的消息。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.67. MF1553_MT_GetMsgNumNewly

函数原型: int MF1553_MT_GetMsgNumNewly (DevHandle hMF1553, DWORD ch, DWORD *MsgNum);

函数功能: 获取 MT 新消息数量。即硬件处理完成、软件未来得及读取的消息数量。

参数说明: hMF1553: 板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

返回值: 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.68. MF1553_MT_ReadMsg

函数原型: int MF1553_MT_ReadMsg (DevHandle hMF1553, DWORD ch, STRUCT_MF1553_RMSG *Msg);

函数功能: 单个读取 MT 处理完的消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读

出消息也是按 FIFO 机制读取。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

Msg: “接收消息结构”指针，内容参考相应结构体说明。

返回值： 读出消息，返回值为 1；未读出消息，返回值为 2；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.69. MF1553_MT_ReadMsgs

函数原型： int MF1553_MT_ReadMsgs (DevHandle hMF1553, DWORD ch, DWORD MsgNum, STRUCT_MF1553_RMSG *Msgs);

函数功能： 批量读取 MT 处理完的消息。硬件将处理完的消息按 FIFO 机制写入缓存，所以软件读出消息也是按 FIFO 机制读取。执行此函数之前，通常先执行函数 MF1553_MT_GetMsgNumNewly 以获取消息数量。

参数说明：hMF1553：板卡的句柄。

ch: 0: 通道 1; 1: 通道 2; 2: 通道 3; 3: 通道 4。

MsgNum: 消息数量。

Msgs: “接收消息结构”指针组，内容参考相应结构体说明。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.70. MF1553_INT_Thread

函数原型： int MF1553_INT_Thread (DevHandle hMF1553);

函数功能： 创建中断线程。

参数说明：hMF1553：板卡的句柄。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.71. MF1553_INT_Init

函数原型： int MF1553_INT_Init (DevHandle hMF1553, INT_HANDLER IntFunc);

函数功能： 中断初始化。初始化的过程中，使能了总中断。

参数说明：hMF1553：板卡的句柄。

INT_Func: 中断回调函数。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.72. MF1553_INT_EnableGlobal

函数原型： int MF1553_INT_EnableGlobal (DevHandle hMF1553, DWORD Enable);

函数功能： 使能总中断。

参数说明：hMF1553：板卡的句柄。

Enable: 总中断使能位。最低位有效。1: 开启总中断；0: 关闭总中断。

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.73. MF1553_INT_EnableBit

函数原型: int MF1553_INT_EnableBit (DevHandle hMF1553, DWORD EnableBitVector);

函数功能: 使能分中断。

参数说明: hMF1553: 板卡的句柄。

EnableBitVector: 分中断使能矢量。分中断使能位定义: 1: 中断使能; 0: 不使能。位定义如下表 3-12 《分中断位使能表》。

表 3-12 分中断位使能表

BIT	说明
31	4 通道外部触发生成中断使能
30	4 通道 BC 指定微帧生成中断使能
29	4 通道 BC 指定消息生成中断使能
28	4 通道 MT 消息结束中断使能
27	4 通道 RT 发送消息结束中断使能
26	4 通道 RT 接收消息结束中断使能
25	4 通道 BC 帧结束中断使能
24	4 通道 BC 消息结束中断使能
23	3 通道外部触发生成中断使能
22	3 通道 BC 指定微帧生成中断使能
21	3 通道 BC 指定消息生成中断使能
20	3 通道 MT 消息结束中断使能
19	3 通道 RT 发送消息结束中断使能
18	3 通道 RT 接收消息结束中断使能
17	3 通道 BC 帧结束中断使能
16	3 通道 BC 消息结束中断使能
15	2 通道外部触发生成中断使能
14	2 通道 BC 指定微帧生成中断使能
13	2 通道 BC 指定消息生成中断使能
12	2 通道 MT 消息结束中断使能
11	2 通道 RT 发送消息结束中断使能
10	2 通道 RT 接收消息结束中断使能
9	2 通道 BC 帧结束中断使能
8	2 通道 BC 消息结束中断使能
7	1 通道外部触发生成中断使能
6	1 通道 BC 指定微帧生成中断使能
5	1 通道 BC 指定消息生成中断使能
4	1 通道 MT 消息结束中断使能
3	1 通道 RT 发送消息结束中断使能
2	1 通道 RT 接收消息结束中断使能
1	1 通道 BC 帧结束中断使能
0	1 通道 BC 消息结束中断使能

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.74. MF1553_INT_StatusClear

函数原型： int MF1553_INT_StatusClear (DevHandle hMF1553, DWORD StatusClearVector);

函数功能： 清中断，用来清除相应中断状态位。

参数说明： hMF1553：板卡的句柄。

StatusClearVector： 分中断清 0 矢量。分中断清 0 位定义：1：清此中断状态位；0：不清。

位定义如下表 3-13 《清分中断状态位》

表 3-13 清分中断状态位

BIT	说明
31	清 4 通道外部触发生成中断状态位
30	清 4 通道 BC 指定微帧生成中断状态位
29	清 4 通道 BC 指定消息生成中断状态位
28	清 4 通道 MT 消息结束中断状态位
27	清 4 通道 RT 发送消息结束中断状态位
26	清 4 通道 RT 接收消息结束中断状态位
25	清 4 通道 BC 帧结束中断状态位
24	清 4 通道 BC 消息结束中断状态位
23	清 3 通道外部触发生成中断状态位
22	清 3 通道 BC 指定微帧生成中断状态位
21	清 3 通道 BC 指定消息生成中断状态位
20	清 3 通道 MT 消息结束中断状态位
19	清 3 通道 RT 发送消息结束中断状态位
18	清 3 通道 RT 接收消息结束中断状态位
17	清 3 通道 BC 帧结束中断状态位
16	清 3 通道 BC 消息结束中断状态位
15	清 2 通道外部触发生成中断状态位
14	清 2 通道 BC 指定微帧生成中断状态位
13	清 2 通道 BC 指定消息生成中断状态位
12	清 2 通道 MT 消息结束中断状态位
11	清 2 通道 RT 发送消息结束中断状态位
10	清 2 通道 RT 接收消息结束中断状态位
9	清 2 通道 BC 帧结束中断状态位
8	清 2 通道 BC 消息结束中断状态位
7	清 1 通道外部触发生成中断状态位
6	清 1 通道 BC 指定微帧生成中断状态位
5	清 1 通道 BC 指定消息生成中断状态位
4	清 1 通道 MT 消息结束中断状态位
3	清 1 通道 RT 发送消息结束中断状态位
2	清 1 通道 RT 接收消息结束中断状态位
1	清 1 通道 BC 帧结束中断状态位

BIT	说明
0	清 1 通道 BC 消息结束中断状态位

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.3.3.75. MF1553_INT_Status

函数原型： int MF1553_INT_Status (DevHandle hMF1553, DWORD *Status);

函数功能： 读取中断状态，以确定如何响应中断

参数说明： hMF1553：板卡的句柄。

Status： 双字指针，指向中断状态寄存器空间。每位定义：1: 中断状态有效；0: 无效。
此32位字的每一位定义如下表3-14《中断状态位》。

表 3-14 中断状态位

BIT	说明
31	4 通道外部触发生成中断状态
30	4 通道 BC 指定微帧生成中断状态
29	4 通道 BC 指定消息生成中断状态
28	4 通道 MT 消息结束中断状态
27	4 通道 RT 发送消息结束中断状态
26	4 通道 RT 接收消息结束中断状态
25	4 通道 BC 帧结束中断状态
24	4 通道 BC 消息结束中断状态
23	3 通道外部触发生成中断状态
22	3 通道 BC 指定微帧生成中断状态
21	3 通道 BC 指定消息生成中断状态
20	3 通道 MT 消息结束中断状态
19	3 通道 RT 发送消息结束中断状态
18	3 通道 RT 接收消息结束中断状态
17	3 通道 BC 帧结束中断状态
16	3 通道 BC 消息结束中断状态
15	2 通道外部触发生成中断状态
14	2 通道 BC 指定微帧生成中断状态
13	2 通道 BC 指定消息生成中断状态
12	2 通道 MT 消息结束中断状态
11	2 通道 RT 发送消息结束中断状态
10	2 通道 RT 接收消息结束中断状态
9	2 通道 BC 帧结束中断状态
8	2 通道 BC 消息结束中断状态
7	1 通道外部触发生成中断状态
6	1 通道 BC 指定微帧生成中断状态
5	1 通道 BC 指定消息生成中断状态
4	1 通道 MT 消息结束中断状态

BIT	说明
3	1 通道 RT 发送消息结束中断状态
2	1 通道 RT 接收消息结束中断状态
1	1 通道 BC 帧结束中断状态
0	1 通道 BC 消息结束中断状态

返回值： 操作成功，返回值为 0；操作失败，返回值为-1；参数错误，返回值为-2。

3.4. 驱动函数调用步骤

3.4.1. 驱动函数调用步骤（查询方式）

1. 打开板卡
 - 打开板卡（MF1553_Open）
2. 复位板卡
 - （1）全局复位板卡（MF1553_Reset）
 - （2）复位板卡通道（MF1553_ResetChannel）
3. 初始化板卡
 - （1）1553B 总线速率选择（MF1553_SpeedSel）
 - （2）启动或停止时间标签模式（MF1553_AddTimeTag）
 - （3）设置应答超时（MF1553_SetResponseTimeout）
4. BC 模式（指定通道）
 - （1）BC 初始化设置
 - ① BC 初始化（MF1553_BC_Init）
 - ② BC 配置（MF1553_BC_CfgWord）
 - （2）BC 消息发送
 - ① 写 BC 微帧链表，写微帧的 Id 依次增加，0，1，2...（MF1553_BC_WriteFrm）
 - ② 添加微帧链表结束符（MF1553_BC_AddEndOfFrmList）
 - ③ 启动 BC（MF1553_BC_Start）
 - ④ BC 运行过程中，可执行以下操作：
 - 可调用函数 MF1553_BC_FrmRunCtl 动态切换微帧的执行状态；
 - 可调用函数 MF1553_BC_WriteFrm 动态插入、更新微帧
 - 可调用函数 MF1553_BC_WriteMsg 动态插入、更新消息
 - 可调用函数 MF1553_BC_WriteMsgs 动态插入、更新消息组
 - 可调用函数 MF1553_BC_WriteMsgDataBlock 动态更新发送类型消息中的数据
 - ⑤ 停止 BC（MF1553_BC_Stop）
 - （3）BC 消息接收
 - ① 清空接收缓存（MF1553_BC_MsgBufferClear）
 - ② 读取 BC 消息数量（MF1553_BC_GetMsgNumNewly）
 - ③ 根据上一步读取的数量，批量读取消息（MF1553_BC_ReadMsgs）
 - ④ 循环重复②③步

5. RT 模式（指定通道）

- (1) RT 初始化 (MF1553_RT_Init)
- (2) 设置 RT 地址使能 (MF1553_RT_RTAddr)
- (3) 设置 RT 子地址数据发送模式 (MF1553_RT_TxMode)
- (4) 设置清置时标功能 (MF1553_RT_SetTimeTagOnSync)
- (5) 设置 RT 非法命令表 (MF1553_RT_SetIllCmdTable)
- (6) 设置 RT 状态字 (MF1553_RT_SetRTStatusWord)
- (7) 设置 RT 矢量字 (MF1553_RT_SetVectorWord)
- (8) 设置 RT 硬件自动清除矢量字使能 (MF1553_RT_VectorWordAutoClrEn)
- (9) 设置 RT 自检字 (MF1553_RT_SetBitWord)
- (10) 设置 RT 子地址发送数据 (MF1553_RT_WriteMsgDataBlock)
- (11) 设置 RT 子地址循环缓冲区指针回卷深度 (MF1553_RT_SetSaDepth)
- (12) RT 消息接收
 - ① 清空接收缓存 (MF1553_RT_MsgBufferClear)
 - ② 读取 RT 接收类型消息数量 (MF1553_RT_GetMsgNumRxNewly)
 - ③ 根据上一步读取的数量，批量读取接收类型消息 (MF1553_RT_ReadMsgsRx)
 - ④ 读取 RT 发送类型消息数量 (MF1553_RT_GetMsgNumTxNewly)
 - ⑤ 根据上一步读取的数量，批量读取发送类型消息 (MF1553_RT_ReadMsgsTx)
 - ⑥ 循环重复②~⑤步

6. MT 模式（指定通道）

- (1) MT 初始化 (MF1553_MT_Init)
- (2) 设置 MT 监控消息过滤表 (MF1553_MT_SetCmdFilterTable)
- (3) MT 启动监听 (MF1553_MT_Start)
- (4) MT 消息接收
 - ① 清空接收缓存 (MF1553_RT_MsgBufferClear)
 - ② 读取 MT 消息数量 (MF1553_MT_GetMsgNumNewly)
 - ③ 根据上一步读取的数量，批量读取消息 (MF1553_MT_ReadMsgs)
 - ④ 循环重复②③步
- (5) MT 停止监听 (MF1553_MT_Stop)

7. 关闭板卡

- (1) 全局复位板卡 (MF1553_Reset)
- (2) 关闭板卡 (MF1553_Close)

4. Demo 应用程序说明

MF1553 应用程序是本公司针对 1553 系列板卡产品开发的程序，提供 BC/RT/BM（总线控制器/远程终端/总线监视器）的创建、配置、通讯、关闭，以及消息的保存。

在演示程序中使用了 microsoft FlexGrid Control, 如果机器上未安装该控件,程序将不能正常运行. 在附带光盘“应用程序\extra”目录下含有该控件, 先用 regsvr32.exe 注册此控件 (msflxgrd.ocx), 再将.reg 文件导入注册表。

4.1. Demo 运行环境

4.1.1. 硬件环境

至少 128M 内存
显示器分辨率至少可设为 800*600
1553-xxxx 通讯板卡

4.1.2. 软件环境

WinXP/Win7 操作系统
1553-xxxx 硬件驱动程序

4.1.3. 开发工具

Visual Studio 2010

4.2. Demo 简介

4.2.1. 板卡选择窗口



图 4-1 板卡号选择窗口

4.2.2. 主窗口

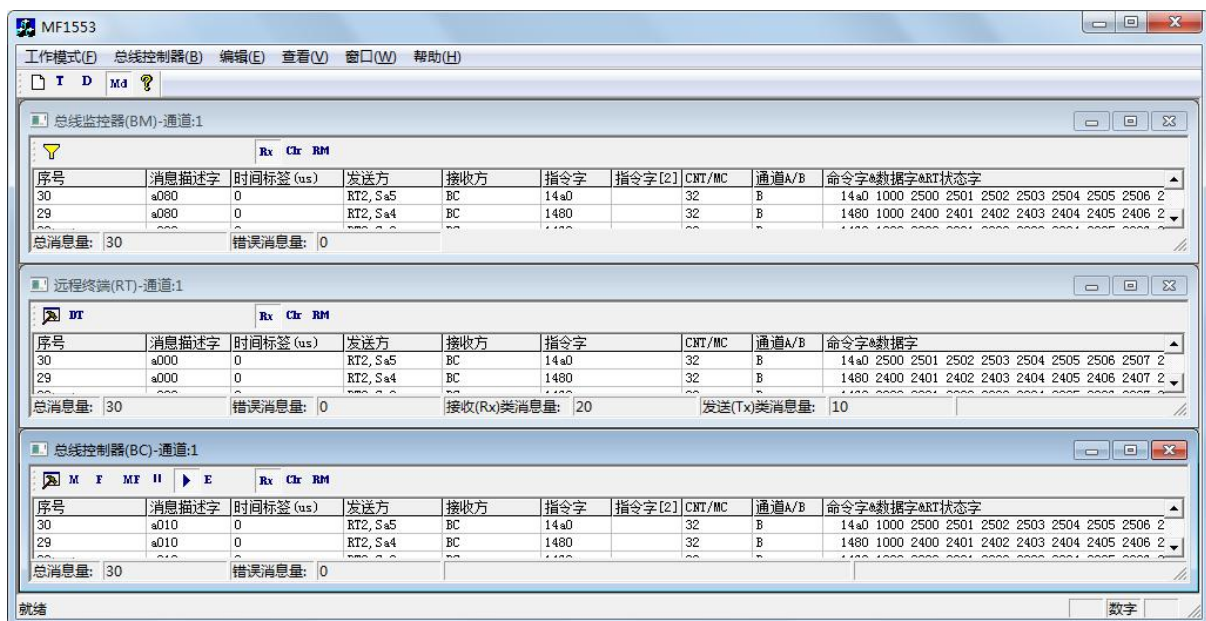


图 4-2 MF1553 应用程序主窗口

4.2.3. Demo 菜单功能

4.2.3.1. 工作模式菜单命令

工作模式菜单提供了以下命令：

新建模式	创建 BC、RT、BM 操作(工作模式)。
设置时标与超时	设置时间标签使能/不使能, 和超时时间设置。
数字量 I/O	打开数字量 IO 操作窗口。
接收消息方式	选择接收消息方式: ① 后台文件接收 (默认); ② 实时显示接收。

注:

BC -- 总线控制器: Bus Controller

RT -- 远程终端: Remote Terminal

BM -- 总线监控器: Bus Monitor

4.2.3.2. 总线控制器 (BC) 菜单命令

总线控制器 (BC): Bus Controller。

定义: 总线控制器是总线系统中组织信息传输的终端。

总线控制器命令如下:

设置	显示设置窗口。
消息	显示消息配置窗口。
创建微帧	显示微帧配置窗口。
创建主帧	显示主帧配置窗口。
停止 BC	停止 BC, 退出运行状态。
启动 BC	将用户配置好的数据帧写入硬件, 且启动 BC 运行。
运行控制	在 BC 运行状态下, 执行删除、插入、更新微帧操作等。
开启接收/停止接收	开启接收/停止接收操作。
清空接收数据	清空接收数据显示列表。
消息显示	打开已保存文件的消息。

注:

操作菜单中的“设置”、“消息”、“创建微帧”、“创建主帧”, 需要处于“停止 BC”状态。

操作菜单中的“运行控制”, 需要处于“启动 BC”状态。

操作菜单中的“清空接收数据”、“消息显示”, 需要处于“停止接收”状态。

4.2.3.3. 远程终端 (RT) 菜单命令

远程终端 (RT): Remote Terminal。

定义: 远程终端是总线系统中不作为总线控制器或总线监控器的所有终端。

远程终端菜单命令如下:

设置	显示设置窗口。
数据配置	显示数据配置窗口。
开启接收/停止接收	开启接收/停止接收操作。
清空接收数据	清空接收数据显示列表。
消息显示	打开已保存文件的消息。

注:

操作菜单中的“清空接收数据”、“消息显示”，需要处于“停止接收”状态。

4.2.3.4. 总线监控器（BM）菜单命令

总线监控器（BM）：Bus Monitor。

定义：总线监控器是总线系统中指定作接收且记录总线上传输的信息并有选择地提取信息以备后用的终端。

总线监控器菜单命令如下：

过滤器	显示过滤设置窗口。
开启接收/停止接收	开启接收/停止接收操作。
清空接收数据	清空接收数据显示列表。
消息显示	打开已保存文件的消息。

注：

操作菜单中的“清空接收数据”、“消息显示”，需要处于“停止接收”状态。

4.2.3.5. 查看菜单命令

查看菜单提供了以下命令：

工具栏	显示或隐藏工具栏。
状态栏	显示或隐藏状态栏。

4.2.3.6. 窗口菜单命令

窗口菜单提供了以下命令。这些命令使用户能在应用程序窗口中安排多个文档的多个视图：

层叠	按重叠方式安排窗口。
平铺	按互不重叠平铺方式安排窗口。
活动窗口 1, 2, ...	转到指定的窗口。

4.2.3.7. 帮助菜单命令

帮助菜单提供以下的命令，为您提供使用这个应用程序的帮助：

帮助主题	提供您可从其得到帮助的主题索引。
关于	显示这个应用程序的版本号、公司信息。

4.2.4. Demo 操作建议

4.2.4.1. 开启应用程序

从当前路径中找到“...\应用程序\source_poll\Debug”或“...\应用程序\source_poll\Release”中的 MF1553.exe，直接双击运行程序。

4.2.4.2. 板卡号选择

当在同一设备中插入多块同样的板卡时,应用程序要求指定板卡号,板卡号选择范围为0到255,默认为0号,并且需要选择和硬件设备相对应的“通道数”和“功能”。

4.2.4.3. 设置时间标签与应答超时

启动或停止时间标签,并设置应答超时时间(硬件上电默认超时时间为10us)。

4.2.4.4. 新建工作模式

执行菜单中的“新建模式”命令,从“新建工作模式窗口”中选择需要创建的工作模式。

4.2.4.5. 总线控制器(BC)操作

设置总线控制器(BC);
配置消息;
创建微帧;
创建主帧;
开启接收消息;
启动BC;
停止BC;
停止接收消息;
打开文件查看接收消息。

4.2.4.6. 远程终端(RT)操作

设置远程终端(RT);
配置RT的数据;
开启接收消息;
停止接收消息;
打开文件查看接收消息。

4.2.4.7. 总线监控器(BM)操作

消息过滤设置;
开启监视消息;
停止监视消息;
打开文件查看接收消息。

4.3. Demo 使用详细说明

MF1553 应用程序大致包括三类操作(工作模式):

- ◆ 总线控制器 (BC)
- ◆ 远程终端 (RT)
- ◆ 总线监控器 (BM)

如图所示，MF1553 应用程序窗口为多文本窗口形式，允许同时存在多个**总线控制器 (BC)**、**远程终端 (RT)** 和**总线监控器 (BM)** 操作（工作模式）。但是，只允许用户在同一时刻操作一个窗口，这个窗口的标题栏会变为激活状态，如上图中的“BC 总线控制器”窗口就为激活状态，称这类窗口为活动窗口，本帮助中有时也称其为“活动工作模式窗口”。

4.3.1. 板卡号选择



图 4-3 MF1553 应用程序主窗口

“**板卡号**”范围为 0 到 255。当用户在同一设备中插入 1 块或多块相同系列板卡时，用户可通过板卡号的选择，对指定板卡进行操作。板卡号与对应板卡的顺序选择关系由如下规则决定：

PCI 系列接口卡：

板卡号范围为 0 到 255。板卡号与对应板卡的顺序选择关系由如下规则决定：默认距离设备主板上 CPU 最近的为 0 号，最远的为 N 号， $255 \geq N > 0$ 。即自距离 CPU 插槽由近到远的顺序，板卡号便会由小变大。

USB 系列接口卡：

只插入 1 块板卡 Card_A 时，板卡的板卡号为 0；再插入 1 块板卡 Card_B 时，板卡 Card_B 的板卡号为 0，板卡 Card_A 的板卡号升级为 1；再插入 1 块板卡 Card_C 时，板卡 Card_C 的板卡号为 0，板卡 Card_B 的板卡号升级为 1，板卡 Card_A 的板卡号升级为 2；注意，由于 USB 接口卡插入后，已经挂接在设备上的同型号板卡对应的板卡号会改变，在打开板卡的情况下，不要在操作板卡的过程中插入新的相同系列板卡。

“**通道数**”和“**功能**”选择由待打开的板卡型号决定。如型号为“1553-xxx-2F”板卡为双通

道多功能板卡，所以“通道数”选择 2，“功能”选择“多功能”。

“总线速率”选择 1553B 板卡的总线速率。我公司 1553B 板卡系列，一部分板卡兼容 1Mbps 和 4Mbps 两种速率，另一部分板卡只支持单一速率。选择之前，先确定此板卡支持的总线速率，否则选择无效。

当用户完成板卡号的选择后，执行“确定”命令便可提交设置给应用程序，程序会自动通过该信息查找板卡，如果找到板卡软件会进入到 MF1553 应用程序的主窗口，等待用户执行新的操作。执行“关闭”或“取消”命令，程序会立刻退出。

4.3.2. 程序运行模式说明

程序运行时有两种模式，一种是“演示模式”，另一种是“应用模式”。当应用程序在提交了板卡号设置后，没有找到指定板卡时，会提示错误信息，如下图所示：

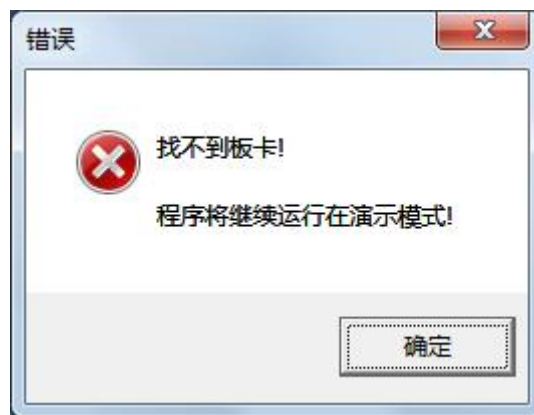


图 4-4 板卡打开错误信息窗口

建议用户在出现该提示后，及时退出应用程序，确认板卡的驱动程序被正确安装，且设备中插入了我公司的与应用程序配套的 1553 板卡。

此时程序不会立刻退出，会运行在“演示模式”，在该模式下，程序只会提供基本设置项的演示，不提供数据收发功能的演示。

若程序在提交板卡号后没有出现该错误提示，直接进入主窗口中，则程序工作在“应用模式”下，在该模式下，程序可对板卡提供的大部分功能进行操作与应用，也可以进行数据通讯操作。以下的说明，主要针对“应用模式”进行介绍。

4.3.3. 设置时间标签与应答超时



图 4-5 时标使能与超时时间设置窗口

该窗口提供“启动/停止时标功能”与“应答超时时间设置”两个功能设置。

“启动/停止时标功能”选项	选中时为“启动时标功能”，未选中时为“停止时标功能”。
“应答超时”编辑窗口	应答超时的设置。取值范围 $0 \sim 2^{32}-1$ ，分辨率 0.5us，实际应答超时时间计算公式如下： 实际应答超时时间 = 输入值 * 0.5us。

4.3.4. 新建工作模式

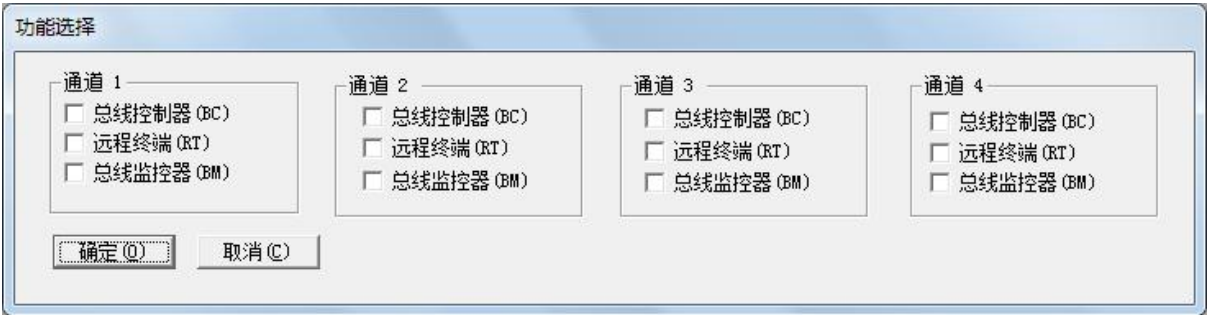


图 4-6 工作模式选择窗口

MF1553 应用程序的每个通道都有“总线控制器（BC）”、“远程终端（RT）”和“总线监控器（BM）”三种工作模式。

如在“板卡选择界面”中选择的是单功能板卡，那么在当前窗口的一个通道中只能选中 1 种工作模式；如在“板卡选择界面”中选择的是多功能板卡，那么在当前窗口的一个通道中能选择多种工作模式。

执行“取消”命令，该窗口不做任何操作；执行“确定”命令，应用程序会创建选中的有效工作模式，当没有任何有效的工作模式被选中时，不做任何操作。

4.3.5. 总线控制器（BC）操作

4.3.5.1. 设置 BC

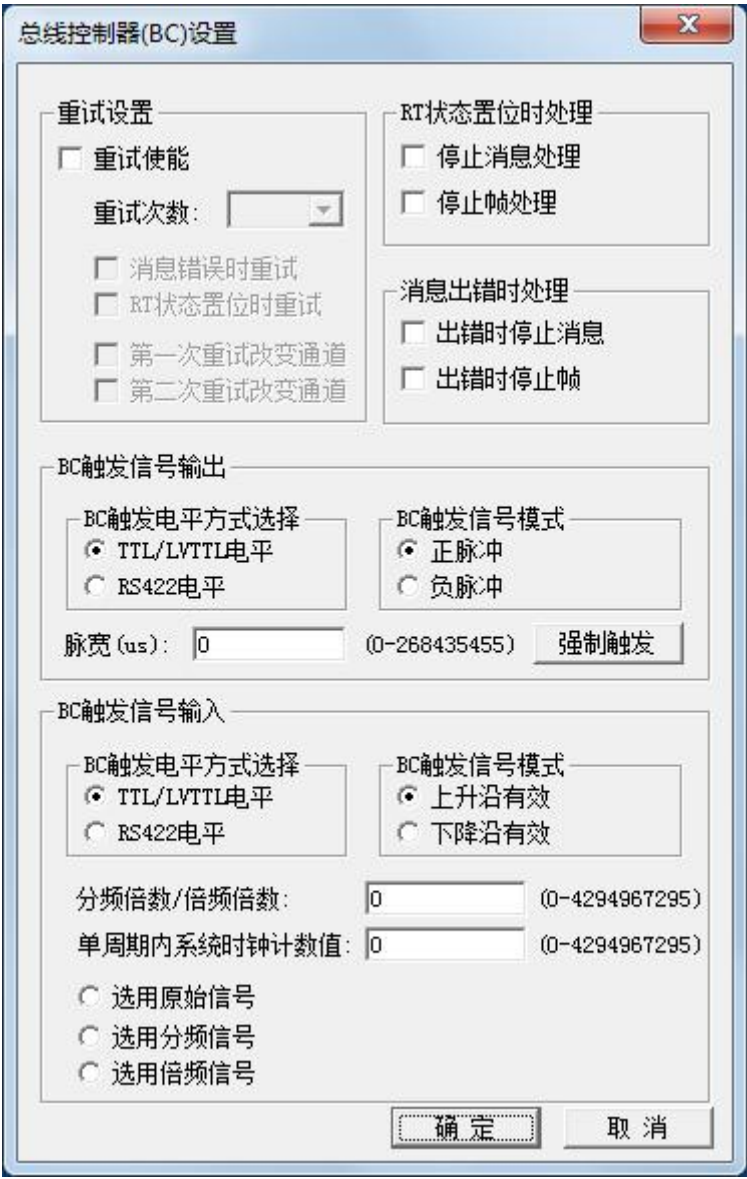


图 4-7 BC 功能设置窗口

“重试设置”栏，设置了消息重试是否使能，重试的条件，以及重试的方式。

“重试使能”选项	选中，表示下列的重试条件，和重试方式可操作。 未选中，关于重试的设置不可操作。
“重试次数”选项	从下拉菜单中选择“一次”、或“二次”。选择“一次”，表示消息失败后，只重试一次，无论重试成功或失败；选择“二次”，表示消息失败后，如果第一次重试成功，则不再重试；如果第一次重试也失败，进行第二次重试。
“消息错误时重试”选项	选中后，在消息错误时(包括字错误、帧格式错误、超时错误)，重试。未选中，即使消息错误，也不重试。

“RT 状态置位时重试”选项	选中后,在 RT 状态置位时(RtStatus[10]:消息错误位; RtStatus[8]: 服务请求位; RtStatus[7:5]: 保留位; RtStatus[4]:广播位; RtStatus[3]:忙位; RtStatus[2]:子系统标志位; RtStatus[0]:终端标志位),重试。未选中,即使 RT 状态置位,也不重试。
“第一次重试改变通道”选项	选中后,第一次重试从另一通道(如刚完成的消息从 A 通道,那么重试从 B 通道;反之亦然)。未选中,第一次重试仍从原通道。
“第一次重试改变通道”选项	选中后,第二次重试从另一通道(如刚完成的消息从 A 通道,那么第二次重试从 B 通道;反之亦然)。未选中,第二次重试仍从原通道。

“RT 状态置位时处理”栏,设置 RT 状态置位时,消息停止,或帧停止。

“停止消息处理”选项	选中,表示在 RT 状态置位时,停止执行后续消息。未选中,RT 状态是否置位,不影响后续消息的执行。
“停止帧处理”选项	选中,表示在 RT 状态置位时,停止执行后续帧。未选中,RT 状态是否置位,不影响后续帧的执行。

“消息错处时处理”栏,设置消息出错时,消息停止,或帧停止。

“出错时停止消息”选项	选中,表示在消息出错(如有重试,重试也失败)时,停止执行后续消息。未选中,消息是否出错,不影响后续消息的执行。
“出错时停止帧”选项	选中,表示在消息出错(如有重试,重试也失败)时,停止执行后续帧。未选中,消息是否出错,不影响后续帧的执行。

“BC 触发信号输出”栏,设置 BC 触发信号输出模式。

“TTL/LVTTL 电平”选项	选中,表示外部触发输出以 TTL/LVTTL 电平方式引出。 此项选中之前,必须相应的,将板卡上 TTL/LVTTL 输出信号引出到外部。板卡使用前,根据外部触发输出的电平高低的要求(TTL 或 LVTTL),合理地设置板卡上跳线帽位置。
“RS422 电平”选项	选中,表示外部触发输出以 RS422 电平方式引出。 此项选中之前,必须相应的,将板卡上 RS422 输出信号引出到外部。
“正脉冲”选项	选中,表示选择触发输出的有效信号是正脉冲。
“负脉冲”选项	选中,表示选择触发输出的有效信号是负脉冲。
“脉宽(us)”编辑窗口	触发输出脉冲宽度,分辨率 1us。
“强制触发”按钮	手动点击 1 次,按照当前设定参数,脉冲输出 1 次。

“BC 触发信号输入”栏,设置 BC 外部触发信号输入模式。

“TTL/LVTTL 电平”选项	选中,表示外部触发输入以 TTL/LVTTL 电平方式引入。 此项选中之前,必须相应的,将外部 TTL/LVTTL 输入信号引入到板卡上。板卡使用前,根据外部触发输入
------------------	--

	的电平高低的要求（TTL 或 LVTTTL），合理地设置板卡上跳线帽位置。
“RS422 电平”选项	选中，表示外部触发输入以 RS422 电平方式引入。 此项选中之前，必须相应的，将外部 RS422 输入信号引入到板卡上。
“上升沿有效”选项	选中，表示选择触发输入的有效边沿为上升沿。
“下降沿有效”选项	选中，表示选择触发输入的有效边沿为下降沿。
“分频倍数/倍频倍数”编辑窗口	当选中“选用分频信号”选项时，这里填写的是分频倍数；当选中“选用倍频信号”选项时，这里填写的是倍频倍数；
“单周期内系统时钟计数值”编辑窗口	当选中“选用倍频信号”选项时，这里填写的是倍频时，1 个周期内系统时钟计数的数量；
“选用原始信号”选项	选中，表示外部触发输入原始信号作为触发信号。
“选用分频信号”选项	选中，表示外部触发输入分频信号作为触发信号。
“选用倍频信号”选项	选中，表示外部触发输入倍频信号作为触发信号。

执行“确定”命令，程序会将设置提交给硬件，执行“取消”命令关闭该窗口。窗口中的复选项当为“选中”状态时为有效，当为“不选中”状态时为无效。

4.3.5.2. 配置消息

消息的配置是 1553 数据通讯中必不可少的环节，关于消息的类型、格式及其详细说明，请参考 1553 相关的标准手册。

4.3.5.2.1. 消息的配置与显示

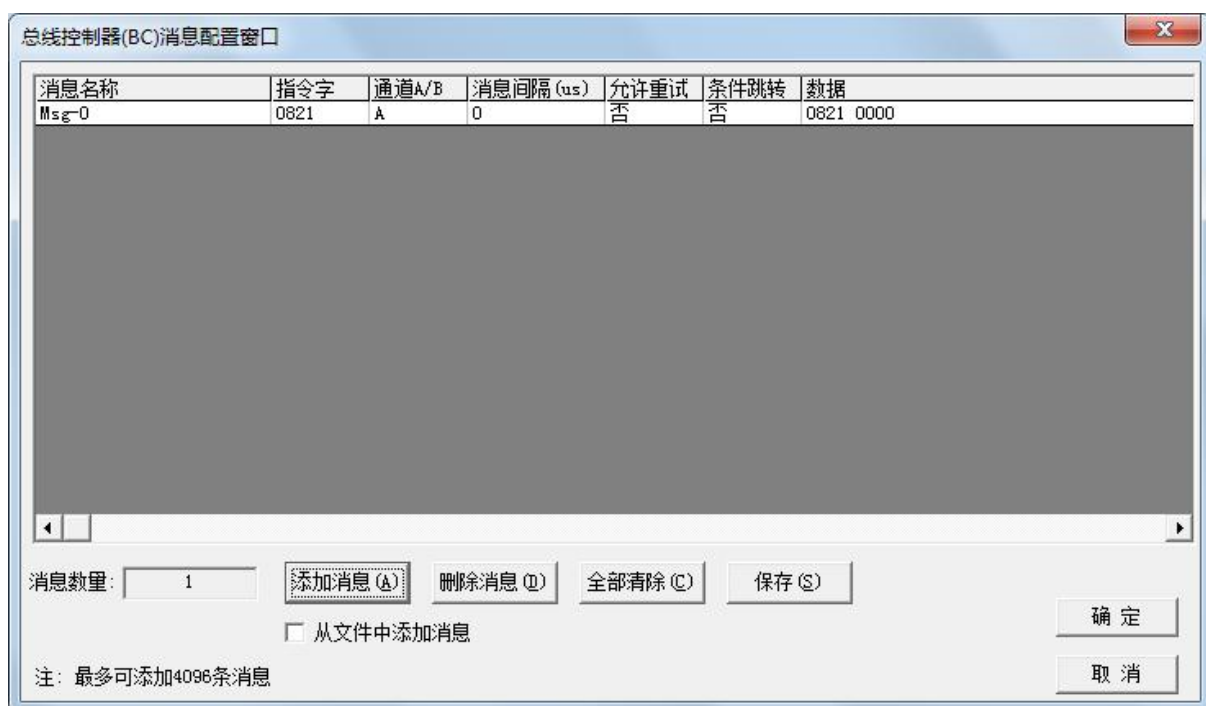


图 4-8 BC 消息配置窗口

上图中所示的窗口为总线控制器（BC）的“消息配置窗口”，用户可通过操作该窗口的功能，实现消息配置。窗口中的列表提供了对已配置消息的显示，程序要求用户一次最多只能配置 4096 条消息。“消息数量”窗口显示的是当前列表中，已经配置了的消息的总数量。

对消息的操作包括如下操作：

操 作	说 明
添加消息	执行“添加消息”命令进行消息添加，程序会按顺序弹出相关的对话框，提示用户进行消息的创建与编辑操作。 如果选中了“从文件中添加消息”，则程序会提示用户指定一个之前已经保存了的消息文件（.msg），确认后，程序会将文件中的消息读入到窗口的列表中，从文件中读取的消息在添加到列表中时，会遵循如下规则： - 从现有消息的最后一条消息开始添加； - 列表中不允许出现同名的消息，在添加时当消息重名，程序提示是否更新同名消息，用户此时可做出选择； - 文件中的消息总数量加上现有列表窗口中的消息总量，如果超过了程序允许的最多添加的消息数量（4096 条），则文件中排列靠后的消息不会被添加到列表窗口中。 添加消息需要“创建消息窗口”与“消息编辑窗口”组合来使用，才会使消息的添加与编辑变得简便，“创建消息窗口”与“消息编辑窗口”的使用请参考 4.3.5.2.2 和 4.3.5.2.3 的内容。 当消息列表中没有消息时，则用户可通过用鼠标双击消息列表的空白处，也可以实现添加消息操作。
删除消息	执行“删除消息”命令可将当前选中位置的消息从列表中删除，如果当前没有被选中的消息位置，则程序会自动从最后一条消息开始删除。
全部清除	执行“全部清除”命令，程序会将列表窗口中的所有消息全部清除掉。
保存	点击“保存”按钮，程序会将列表窗口中现有的消息保存到用户指定的消息文件（.msg）中。
编辑消息	用户可在消息列表中所选的消息上，用鼠标进行双击操作，程序会弹出“消息编辑窗口”供用户对该消息进行编辑。

消息列表会将消息内容分段显示，也就是将消息信息进行分类显示，列表提供了如下字段进行消息主要信息的显示：

字段名称	说明
消息名称	显示用户为该消息定义的字称。
指令字	消息中的指令字。
通道 A/B	当前消息通讯时选择的通道，A 或 B。
消息间隔(us)	即该消息发送完成+等待延时的时间。分辨率 1us，取值 0~2 ³² -1。
允许重试	当消息出错或状态置位时，是否允许消息重试。
条件跳转	是否设置了有效的消息分支跳转结构。
数据	显示消息缓冲区中的数据，包括命令字与数据部分，第一个字是指令字，之后的均为数据，当消息为 RT 到 RT 类型时，则前两个字为指令字。

执行“确定”操作，程序会将当前消息列表窗口中的消息全部加入消息配置列表，等待下次弹

出该窗口时，这些加入消息配置列表的消息，会再次显示在消息列表窗口中。执行“取消”或“关闭”操作，程序不会对消息的配置进行任何修改，当再次打开消息配置窗口时，消息列表中显示的是上次配置的消息。

4.3.5.2.2. 创建消息

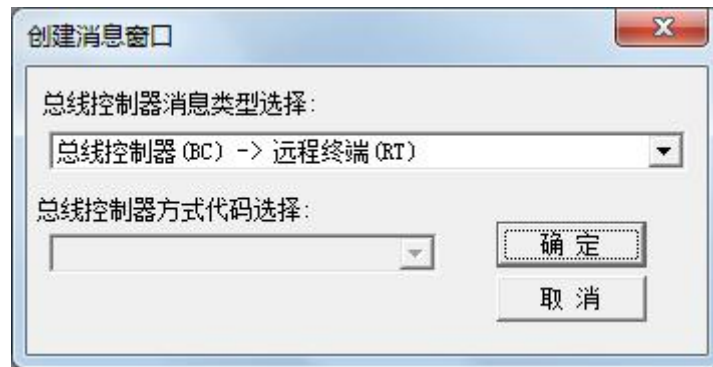


图 4-9 BC 创建消息窗口

进行消息添加操作时，程序最先弹出的是“创建消息窗口”，如上图所示。该窗口提供了对 1553 消息的分类，用户可以通过该分类，先确定创建消息的类型。通过对“总线控制器消息类型选择”和“总线控制器方式代码选择”的操作进行消息分类。

在该窗口中，执行“确定”进入到“消息编辑窗口”，当执行“确定”命令后，消息编辑窗口中会将用户所选的消息类型显示在编辑面板上，以便用户进行消息的详细编辑。执行“取消”退出到消息配置窗口。这个默认的消息类型为“总线控制器消息类型选择”的第一个类型。

4.3.5.2.3. 编辑消息



图 4-10 BC 消息编辑窗口

当用户在“消息配置窗口”双击指定消息，或“创建消息窗口”点击确定后，程序会弹出该窗口让用户对所选消息进行详细编辑。

“消息名称”为此消息编辑一个名称，为 18 位字母、数字和符号组合。注意在“消息配置窗口”中，不得有重名消息。

“消息格式”栏，在“创建消息窗口”选中消息类型后，这里的 3 个选项（“方式”选项、“广播”选项、“远程终端/远程终端”选项）已自动生成。当然用户可以根据要求修改其中选项，来修改当前消息格式，但不建议修改。因为修改这 3 个选项后，生成的消息类型，已经不是在“创建消息窗口”选中消息类型。

“常规参数设置”栏，设置了消息的标准参数。

“通道 A/B”选项	在下拉列表中，选择发送通道：A 通道、B 通道。
“消息时间间隔(us)”编辑窗口	设置消息间的间隔，分辨率 1us，取值 0~2 ³² -1。 消息间隔是指当前消息的起始位置到后一条消息起始位置的时间。如设定值小于实际执行时间，则设置无效，发送完此消息后，可立刻发送后续消息。如设定值大于实际执行时间，发送完此消息后，必须等待间隔时间到达后，才可以发送后续消息。
“允许重试”选项	选中后，允许消息重试；未选中，不允许消息重试。 注意，当选中重试之前，重试方式应该在“设置 BC”窗口中提前设置，请参考 4.3.5.2.1 的内容。

“附加功能设置”按钮，击中后，进入到“消息附加功能”窗口。在“消息附加功能”窗口中，对消息分支跳转、消息引发脉冲输出、消息引发中断等附加功能进行设置。

“指令字配置”栏，配置消息指令字。选择合适的地址、T/R 位、子地址、和数据量/方式代码，最终的指令字会以十六进制的形式显示在“指令字配置”区的“指令字”项中。注意，“消息格式”、“指令字配置”与“数据字配置”是三个相关的操作，改变其中的一个区，另外两个区会做相应的变化。

- ① D15-D11 是 RT 地址，从下拉列表中选择 1~31 的任一地址。如在“创建消息窗口”中选择广播类型的消息，在此处自动生成 RT 地址的默认值为 31，不需修改。
- ② D10 是 T/R 位，也是在“创建消息窗口”中选择消息类型后，此处自动生成，不需修改。
- ③ D9-D5 是子地址，选择 1~30。如在“创建消息窗口”中选择方式代码类型的消息，在此处自动生成 RT 子地址的默认值为 31，不需修改。
- ④ D4-D0 是消息数量、或方式代码。在消息有发送数据时，可选择消息数量为 0~31，其中 0 代表 32 个数；在“创建消息窗口”中选择方式代码类型的消息，软件会根据选择的方式代码的类型，在此处自动生成方式代码值，不需修改。

“数据字配置”栏，配置消息中发送的数据。界面上提供了“生成随机数”和“生成顺序数”的能力，使数据配置变得更为简单。执行“生成随机数”命令，程序会按照消息的信息，在“数据字配置区”生成随机数据。执行“生成顺序数”命令前，需要用户指定一个基数值，该命令在生成数据时，会按基数进行累加填入数据。

“远程终端/远程终端”当选定时，程序会在“指令字配置”区显示出第二个指令字的配置，如下图所示：



消息名称: Msg_0

消息格式: ☐ 方式 ☐ 广播 ☒ 远程终端/远程终端

常规参数设置: 通道A/B: 通道[A] 消息时间间隔(us): 0 (Dec.) ☐ 允许重试

附加功能设置

指令字配置

指令字	远程终端地址	发/收	子地址/方式	数据字计数/方式代码
D15-D0	D15-D11	D10	D9-D5	D4-D0
0821	1	接收	1	1
1421	2	发送	1	1

注: 消息名称最多可输入18个字母、数字和符号组合。消息时间间隔取值范围为0到4294967295。

数据字配置 (Hex.)

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0
9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0
17	18	19	20	21	22	23	24
0	0	0	0	0	0	0	0
25	26	27	28	29	30	31	32
0	0	0	0	0	0	0	0

生成随机数 生成顺序数 基数: 0 (Hex.)

确定 取消

图 4-11 BC 消息编辑窗口

执行“确定”命令，程序会将消息编辑面板上的消息信息配置到消息队列中，并会关闭该窗口，最后“消息配置窗口”中会显示该消息。

如果用户执行的是“添加消息”命令，点击确定后，会检查现有消息队列中是否有同名的消息，如果有同名消息存在，程序会提示是否更改消息，如果没有同名的消息则会将该条新消息添加到“配置消息窗口”中消息列表的最末尾处。

当用户在“消息配置窗口”双击指定消息，该窗口会显示用户所选消息的内容，用户可在原有内容上进行修改，确认后，程序也会提示是否将原有消息更新。如果用户在编辑消息时，更改了消息的名称，且该消息名称在消息队列中不存在，则程序会对现有的消息进行“添加”操作，窗口关闭后，消息便会出现在“配置消息窗口”中消息列表的最末尾。

用户执行“取消”或“关闭”命令，程序不会对消息列表做任何改动。

4.3.5.2.4. 编辑消息附加功能

当用户在“消息编辑窗口”击中“附加功能设置”按钮，程序会弹出该窗口让用户对所选消息的附加功能进行详细编辑。

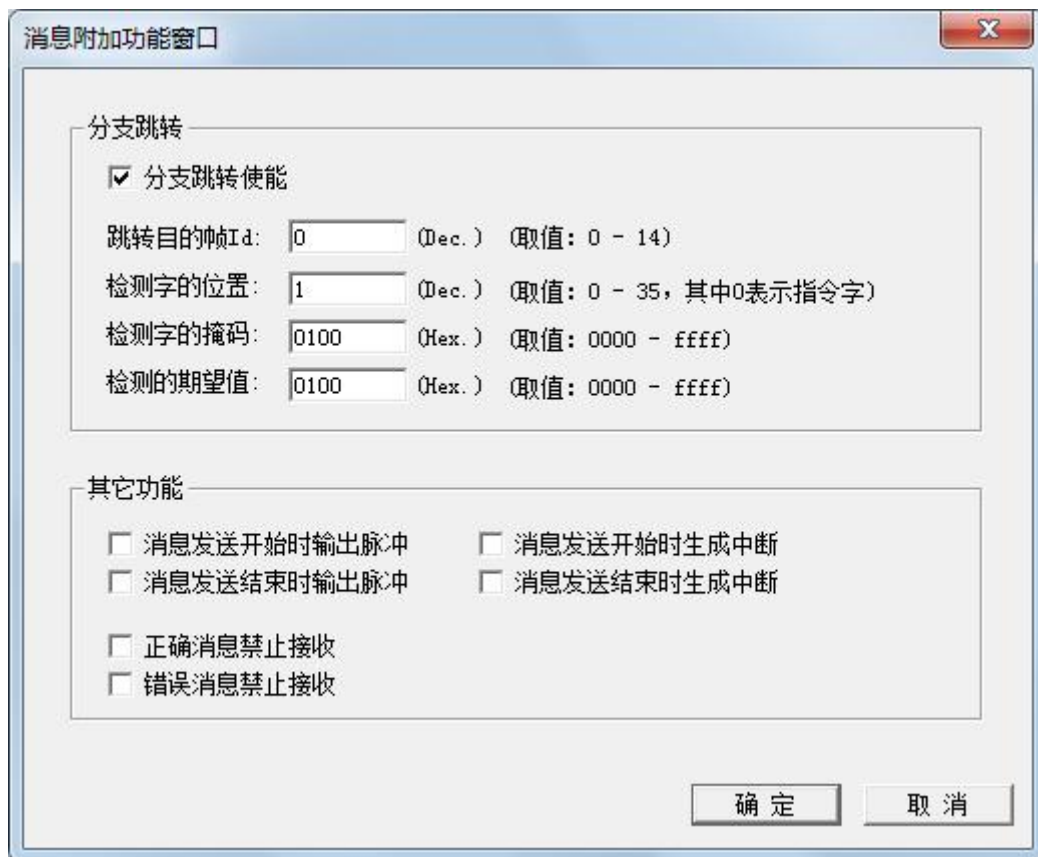


图 4-12 BC 消息附加功能窗口

“条件跳转”栏，设置 BC 消息分支跳转功能。

“分支跳转使能”选项	选中，消息启动分支跳转功能。表示此消息执行结果满足设定条件后，跳转至“跳转目的帧 Id”处执行。未选中，表示不启动分支跳转功能，这种情况下，后续的 4 个跳转参数无意义。通常跳转目的帧为事件微帧，而这种事件微帧的特点是：在某条消息执行结果满足设定条件后，就执行一次。
“条件目的帧 Id”编辑窗口	在此消息执行结果满足设定条件后，需要执行的微帧的 Id（通常为事件微帧）。微帧 Id，表示微帧在发送主帧中存放的位置，取值 0~14。
“检测字的位置”编辑窗口	检测字可以是接收消息块中的任何一个字，此参数

	表示检测字在“BC 接收消息块”中的存放位置。“BC 接收消息块”的存放内容，参考表 3-6 《BC 接收消息块结构》。取值 0~35，始终的，0 表示 BC 发送的第 1 个命令字。
“检测字的掩码”编辑窗口	如校验“检测字（16bit）”全部 Bit 位，那么掩码设为 0xffff。 如校验“检测字（16bit）”部分 Bit 位（如检测 RT 状态字中是否有服务请求位 RT_Status[8]），在掩码中，将需要检测的 Bit 位设为 1，不需要检测的 Bit 位设为 0。 检测过程中，需要对检测字中的这些指定位置的 Bit 位一一检测，全部检测成功，才表示分支条件满足。
“检测的期望值”编辑窗口	预先设定的检测期望值。检测字用指定掩码位的检测结果，和期望值比较，如相同，则表示跳转条件满足，否则，不满足。公式表达：检测“检测字”位与“检测字的掩码”的结果和“检测的期望值”是否相等。

举例：

BC 发送周期性的查询消息（BC 从 RT1 取矢量字），在返回的 RT 状态字中的“服务请求位”置位的情况下，发送 Id=3 的微帧。那么分支跳转参数设置如下：

```

SMMsgBranchEn = 1;           // 分支跳转使能
SMMsgBranchToFrmId = 3;      // 跳转至目标微帧的 Id
SMMsgBranchDataLocation = 1; // 检测字在消息中的位置
SMMsgBranchDataMask = 0x0100; // 检测字掩码
SMMsgBranchDataExpectation = 0x0100; // 检测结果期望值

```

“其它功能”栏，设置 BC 消息的其他附加功能。

“消息发送开始时输出脉冲”选项	选中，表示消息执行开始引发脉冲输出使能。未选中，表示不使能。 脉冲输出必须满足以下条件： ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数； ② 消息执行开始引发脉冲输出使能。
“消息发送结束时输出脉冲”选项	选中，表示消息执行结束引发脉冲输出使能。未选中，表示不使能。 脉冲输出必须满足以下条件： ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数； ② 消息执行结束引发脉冲输出使能。
“消息发送开始时生成中断”选项	选中，表示消息执行开始引发中断使能。未选中，表示不使能。 中断生成必须满足以下条件： ① 总中断使能； ② “消息执行引发中断”的分中断使能； ③ 消息执行开始引发中断使能。

“消息发送结束时生成中断”选项	选中，表示消息执行结束引发中断使能。未选中，表示不使能。 中断生成必须满足以下条件： ① 总中断使能； ② “消息执行引发中断”的分中断使能； ③ 消息执行结束引发中断使能。
“正确消息禁止接收”选项	选中，表示消息执行正确时不写入缓存使能。未选中，表示不使能（执行正确时写入缓存）。 不写入缓存的消息，软件是接收不到执行后的此条消息。如在 BC 发送大量 BC→RT 类型的消息时，成功执行的消息并不需要软件回读，则这里可设置使能。
“错误消息禁止接收”选项	选中，表示消息执行错误时不写入缓存使能。未选中，表示不使能（执行错误时写入缓存）。 不写入缓存的消息，软件是接收不到执行后的此条消息。如在 BC 和多个 RT 通讯时，BC 与网络中不存在的 RT 通讯时，执行的结果一定是错误消息（超时错误）。为屏蔽这类消息，此参数可设置使能，软件将接收不到这些错误消息。

4.3.5.3. 配置微帧

消息配置完成后，需要用户将消息组成消息帧（微帧）。

4.3.5.3.1. 微帧的配置与显示



图 4-13 BC 微帧配置窗口

上图中所示的窗口为总线控制器（BC）的“微帧配置窗口”，用户可通过操作该窗口的功能，

实现微帧配置。窗口中的列表提供了对已配置微帧的显示，程序要求用户一次最多只能配置 15 个微帧。“微帧数量”窗口显示的是当前列表中，已经配置了的微帧的总数量。

对微帧的操作包括如下操作：

操 作	说 明
添加帧	执行“添加帧”命令进行微帧添加，程序会按顺序弹出相关的对话框，提示用户进行微帧的创建与编辑操作。 如果选中了“从文件中添加帧”，则程序会提示用户指定一个之前已经保存了的消息文件（.frm），确认后，程序会将文件中的消息读入到窗口的列表中，从文件中读取的微帧在添加到列表中时，会遵循如下规则： - 从现有微帧的最后一个微帧开始添加； - 列表中不允许出现同名的微帧，在添加时当微帧重名，程序提示是否更新同名微帧，用户此时可做出选择； - 文件中的微帧总数量加上现有列表窗口中的微帧总量，如果超过了程序允许的最多添加的微帧数量（15 个），则文件中排列靠后的微帧不会被添加到列表窗口中。 添加微帧需要通过“微帧编辑窗口”来执行，“微帧编辑窗口”的使用请参考 4.3.5.3.2。 当微帧列表中没有微帧时，则用户可通过用鼠标双击微帧列表的空白处，也可以实现添加微帧操作。
删除帧	执行“删除帧”命令可将当前选中位置的微帧从列表中删除，如果当前没有被选中的微帧位置，则程序会自动从最后一条微帧开始删除。
全部清除	执行“全部清除”命令，程序会将列表窗口中的所有微帧全部清除掉。
保存	点击“保存”按钮，程序会将列表窗口中现有的微帧保存到用户指定的微帧文件（.frm）中。
编辑微帧	用户可在消息列表中所选的微帧上，用鼠标进行双击操作，程序会弹出“微帧编辑窗口”供用户对该消息进行编辑。

微帧列表会将微帧内容分段显示，也就是将微帧信息进行分类显示，列表提供了如下字段进行微帧主要信息的显示：

字段名称	说明
微帧名称	显示用户为该微帧定义的字称。
立即执行	微帧立即执行状态。
周期(us)	微帧执行周期。分辨率 1us，0 表示事件微帧，非 0 表示周期微帧。
重发次数	微帧重发次数。取值 0~2 ³² -1，0 表示无限次。只在周期微帧下有效。
微帧间隔(us)	即该微帧发送完成+等待延时的时间。分辨率 1us。
消息数量	微帧中消息数量。

执行“确定”操作，程序会将当前微帧列表窗口中的微帧全部加入微帧配置列表，等待下次弹出该窗口时，这些加入微帧配置列表的微帧，会再次显示在微帧列表窗口中。执行“取消”或“关闭”操作，程序不会对微帧的配置进行任何修改，当再次打开微帧配置窗口时，微帧列表中显示的是上次配置的微帧。

4.3.5.3.2. 编辑微帧

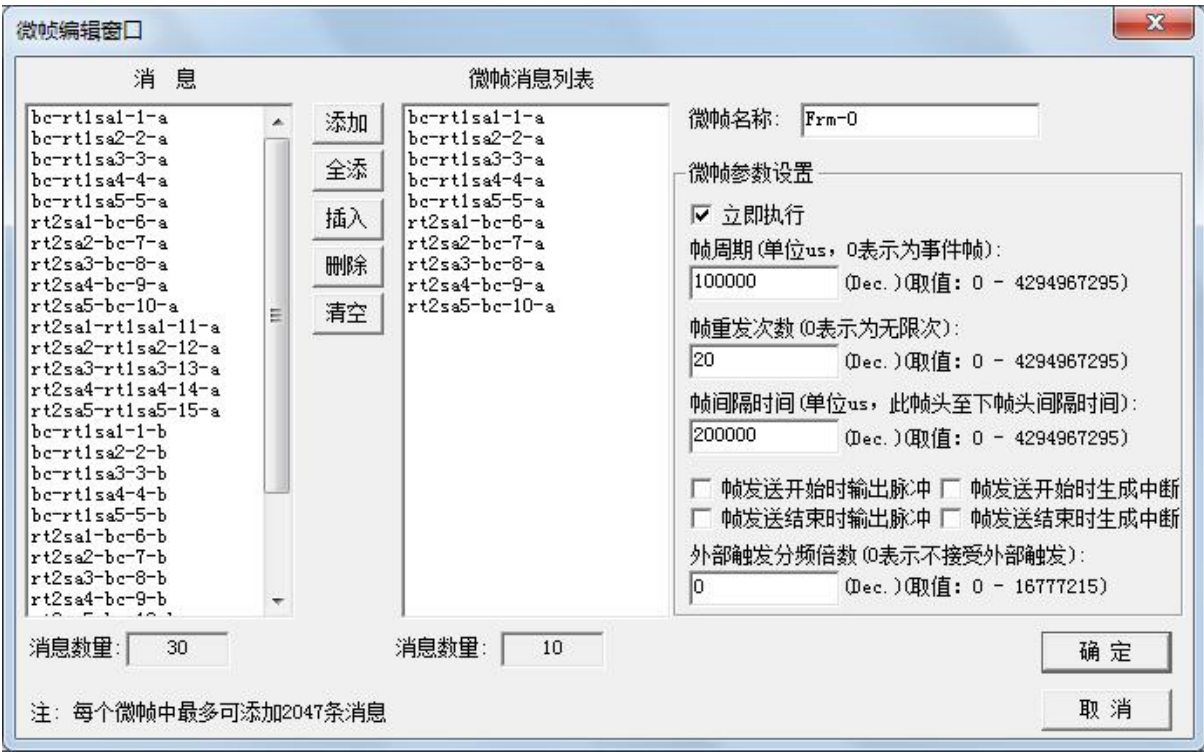


图 4-14 BC 微帧编辑窗口

当用户在“微帧配置窗口”单击“添加帧”按钮，或双击指定微帧，程序会弹出该窗口让用户对新添加的微帧，或所选微帧进行详细编辑。

“微帧名称”为此微帧编辑一个名称，为 18 位字母、数字和符号组合。注意在“微帧配置窗口”中，不得有重名微帧。

“微帧参数设置”栏，设置了微帧的标准参数、附加功能参数。

“立即执行”选项	选中，在运行 BC_Start 函数后，BC 立即执行此微帧； 未选中，在运行 BC_Start 函数后，BC 跳过此微帧，直到用户调用函数 MF1553_BC_FrmRunCtl 来启动该微帧
“帧周期”编辑窗口	微帧发送周期。分辨率 1us，取值 0~2 ³² -1。参数为 0 时，为事件微帧；参数为非 0 时，为周期微帧。
“帧重发次数”编辑窗口	微帧重发次数。取值 0~2 ³² -1，且此参数只在周期微帧下有效。注意，参数为 0 时表示无限次。
“帧间隔时间”编辑窗口	微帧间隔时间。分辨率 1us，取值 0~2 ³² -1。微帧间隔是指当前微帧的起始位置到后一个微帧起始位置的时间。如设定值大于实际执行时间，发送完此微帧后，必须等待间隔时间到达后，才可以发送后续微帧。
“帧触发开始时输出脉冲”选项	微帧执行开始引发脉冲输出使能。 脉冲输出必须满足以下条件： ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数； ② 微帧执行开始引发脉冲输出使能。
“帧触发结束时输出脉冲”选项	微帧执行结束引发脉冲输出使能。

项	脉冲输出必须满足以下条件： ① 使用 MF1553_BC_TxTriggerMode 函数设置合理的脉冲输出参数； ② 微帧执行结束引发脉冲输出使能。
“帧触发开始时生成中断”选项	微帧执行开始引发中断使能。 中断生成必须满足以下条件： ① 总中断使能； ② “微帧执行引发中断”的分中断使能 微帧执行开始引发中断使能。
“帧触发结束时生成中断”选项	微帧执行结束引发中断使能。 中断生成必须满足以下条件： ① 总中断使能； ② “微帧执行引发中断”的分中断使能 微帧执行结束引发中断使能。
“外部触发分频倍数”编辑窗口	外部输入触发微帧执行的分频次数。详细设置参考 3.3.2.4 节“BC 发送微帧附加功能结构”。

“微帧消息列表”中的消息，为当前微帧中的消息，需要从左侧“消息列表”中加入。“消息列表”中的消息为用户在“消息配置窗口”中配置好的消息。要求用户配置好至少 1 条消息的情况下，才能操作该窗口。

操 作	说 明
添加	用户需要提前在“消息列表”中选择一条消息，然后将其添加到“微帧消息列表”的结尾处。
全添	将“消息列表”中的消息，按照顺序依次添加到“微帧消息列表”的结尾处。添加的消息和“微帧消息列表”中消息总和不能超过最大值 2047。
插入	用户需要提前在“消息列表”中选择一条消息，然后执行“插入”操作。如果用户在“微帧消息列表”中没有选择上任何消息时，程序会将“消息列表”中选择的消息插入到“微帧消息列表”中的结尾处；如果用户在“微帧消息列表”窗口中选择了一条消息，则程序会将“消息列表”中选择的消息插入到“微帧消息列表”窗口中选择的那条消息的前一位置。
删除	如果用户在“微帧消息列表”中没有选择上任何消息时，会删除“微帧消息列表”中结尾处的那条消息；如果用户在“微帧消息列表”窗口中选择了一条消息，则程序会删除“微帧消息列表”中选择的那条消息。
清空	清空“微帧消息列表”中所有消息。

执行“确定”命令，程序会将消息编辑面板上的微帧信息配置到消息队列中，并会关闭该窗口，最后“微帧配置窗口”中会显示该微帧。

如果用户执行的是“添加帧”命令，点击确定后，会检查现有微帧队列中是否有同名的微帧，如果有同名微帧存在，程序会提示是否更改微帧，如果没有同名的微帧则会将该条新微帧添加到“配置微帧窗口”中微帧列表的结尾处。

当用户在“微帧配置窗口”双击指定微帧，该窗口会显示用户所选微帧的内容，用户可在原有内容上进行修改，确认后，程序也会提示是否将原有微帧更新。如果用户在编辑微帧时，更改了微

帧的名称，且该微帧名称在微帧队列中不存在，则程序会对现有的微帧进行“添加”操作，窗口关闭后，消息便会出现现在“微帧配置窗口”中微帧列表的最末尾。

用户执行“取消”或“关闭”命令，程序不会对微帧列表做任何改动。

4.3.5.4. 配置主帧

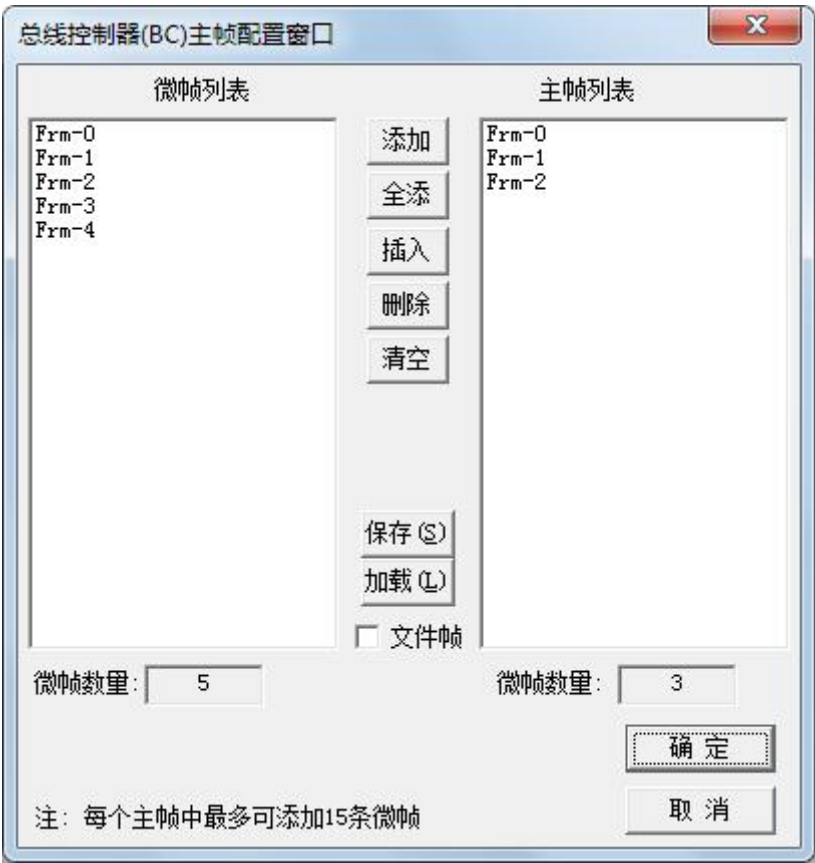


图 4-15 BC 主帧配置窗口

消息配置、微帧配置完成后，再将微帧打包成主帧的形式，进行传输。上图为 BC 的“主帧配置窗口”。组主帧有两种方式：

- ① 从“微帧列表”中加入。要求用户先配置好至少 1 个微帧的情况下，才能操作该窗口。窗口左侧的“微帧列表”为用户在“微帧配置窗口”中配置好的微帧，右侧的“主帧列表”为用户将要打包到主帧中的微帧。用户打包到主帧中的微帧数量最多为 15 个。

操 作	说 明
添加	用户需要提前在“微帧列表”中选择一个微帧，然后将其添加到“主帧列表”的结尾处。
全添	将“微帧列表”中的微帧，按照顺序依次添加到“主帧列表”的结尾处。添加的微帧和“主帧列表”中微帧总和不能超过最大值 15。
插入	用户需要提前在“微帧列表”中选择一个微帧，然后执行“插入”操作。如果用户在“主帧列表”中没有选择上任何微帧时，程序会将“微帧列表”中选择的微帧插入到“主帧列表”中的结尾处；如果用户在“主帧列表”窗口中选择了微帧，则程序会将“微帧列表”中选择的微帧插入到“主帧列表”窗口中选择的那条微帧的前一位置。

删除	如果用户在“主帧列表”中没有选择上任何消息时，会删除“主帧列表”中结尾处的那个微帧；如果用户在“主帧列表”窗口中选择了微帧，则程序会删除“主帧列表”中选择的那个微帧。
清空	清空“主帧列表”中所有微帧。
保存(S)	点击“保存(S)”按钮，程序会将“主帧列表”中所有微帧作为一文件保存到用户指定的主帧文件（.mfrm）中。

② 从文件中加载主帧。则程序会提示用户指定一个之前已经保存了的主帧文件（.mfrm），确认后，程序会将文件中的主帧读入到“主帧列表”中。“主帧列表”加载的消息的数量最多 15 个。这种情况下，加载文件后，微帧的名称显示在“主帧列表”中，但无法查看各微帧的内容。

操 作	说 明
加载(L)	点击“加载(L)”按钮，程序会提示用户指定一个之前已经保存了的主帧文件（.mfrm），确认后，程序会将文件中的主帧中内容读入到“主帧列表”中。

注：

1、从微帧列表中组主帧方式下，当用户需要重新改变微帧时，建议用户先到“微帧配置窗口”和“微帧编辑窗口”将需要修改的微帧进行修改，然后再到该窗口中重新将主帧配置一下（这里指的是执行一下“确定”按钮，即再次提交一次主帧设置），再进行后面的数据通讯操作。

2、从微帧列表中组主帧方式下，当用户需要重新改变消息时，建议用户先到“消息配置窗口”将需要修改的消息进行修改，再到“微帧配置窗口”和“微帧编辑窗口”将和此消息相关的微帧进行修改，最后再到该窗口中重新将主帧配置一下（这里指的是执行一下“确定”按钮，即再次提交一次主帧设置），再进行后面的数据通讯操作。

4.3.5.5. 停止 BC

用户可从工具栏与菜单中找到该命令，执行此命令，所有 BC 消息立即终止发送任何消息。

4.3.5.6. 启动 BC

用户可从工具栏与菜单中找到该命令，但要求在发送主帧前，请确认已经对主帧进行了配置操作，保证至少放置 1 个微帧。

4.3.5.7. 开启/停止接收消息



序号	消息描述字	时间标签(us)	发送方	接收方	指令字	指令字[2]	CNT/MC	通道A/B	命令字#数据字#RT状态字
30	a010	0	RT2, Sa5	BC	14a0		32	B	14a0 1000 2500 2501 2502 2503 2504 2505
29	a010	0	RT2, Sa4	BC	1480		32	B	1480 1000 2400 2401 2402 2403 2404 2405
28	a010	0	RT2, Sa3	BC	1460		32	B	1460 1000 2300 2301 2302 2303 2304 2305
27	a010	0	RT2, Sa2	BC	1440		32	B	1440 1000 2200 2201 2202 2203 2204 2205
26	a010	0	RT2, Sa1	BC	1420		32	B	1420 1000 2100 2101 2102 2103 2104 2105

图 4-16 BC 开启/停止接收消息

用户可通过执行菜单中或工具栏里的“开启接收/停止接收”  按钮，来控制程序是否接收已

处理的消息。按钮按下，表示“开启接收消息”，按钮“弹起”，表示不接收消息。

一旦处于“开启接收消息”，程序有两种接收消息方式。Demo 主菜单上“接收消息方式”按钮，决定接收消息方式：①后台文件接收（弹起）；②实时显示接收（按下）。

后台文件接收方式	当开启接收后，首先清空硬件缓冲区之前接收的消息，避免读出在此之前硬件中残留的消息。然后程序会及时地将硬件 BC 缓冲区中的消息数据读取出来，并将其按顺序保存在当前软件执行目录中新建的.txt 文件中。程序中 RT 与 BM 的模块对缓冲区的操作与上面描述的相同。 注意：后台文件接收方式，不需实时显示消息，保存消息耗时很少，可用于高速率消息的接收。
实时显示接收方式	当开启接收后，首先清空硬件缓冲区之前接收的消息，避免读出在此之前硬件中残留的消息。然后程序会及时地将硬件 BC 缓冲区中的消息数据读取出来，并将消息信息依次显示在接收窗口中。程序中 RT 与 BM 的模块对缓冲区的操作与上面描述的相同。 注意：在实时显示接收消息方式下，方便用户实时观测刚接收的消息。但由于 Windows 界面刷新显示消息比较耗时，在消息速率很大的情况下，会阻塞后续消息的接收。硬件可缓存 23831 条已完成的消息，如长时间阻塞，造成缓存溢出，丢消息。

建议用户在进行数据通讯操作前，先将接收开启，否则会丢失“开启接收”这一刻之前，硬件接收的所有消息。

4.3.5.8. 清空接收数据

该操作会将文件中接收的所有数据全部清除掉（后台文件接收方式），或将接收数据显示窗口中的所有数据全部清除掉（实时显示接收方式）。该操作在“开启接收”状态时无效。

4.3.5.9. 显示数据

程序将接收到的消息数据分类，以字段形式显示，如下所示：

字段名称	说 明
序号	按程序接收到消息的先后顺序编号，最新的消息会显示到接收窗口的最上方，起始编号为 1。
消息描述字	双击该项可显示当前消息的具体描述信息。（只有在停止接收状态下有效）。(暂不支持)
时间标签(us)	显示消息通讯时的相对时间，单位为 1 微秒。
发送方	指消息数据的发送方。
接收方	指消息数据的接收方。
指令字	消息中的指令字。

指令字 (2)	当消息为 RT 到 RT 的类型时，该位置显示消息中的第二个指令字。
CNT/MC	显示消息指令字中的“数据量/方式代码”数值。
通道 A/B	显示消息通讯时选用的通道。
命令字 & 数据字 & RT 状态字	命令字、数据与 RT 状态字的合集。
备注	RT 到 RT 消息与消息出错两种提示，即该栏会显示两种信息，当用户看到消息出错信息时，可以通过查看“消息描述字”获取更多的信息。

4.3.6. 远程终端（RT）操作

4.3.6.1. 设置 RT



图 4-17 RT 设置窗口

该窗口主要是用于对远程终端（RT）与其子地址（SA）功能特性的配置，用户对 RT 与 SA 进行选择时，其相应的功能特性项会进行自动保存与显示。RT 与 SA 取值均为 0 到 31，当 RT 为 31 时，则表明为广播式通讯。SA 为 0 或 31 时，则表明为方式代码传输。具体内容请参考 1553 标准手册。

“远程终端（RT）地址选择”栏，选择参与通讯的远程终端。用户可选择 31 个 RT 中任意个地址有效，参与数据通讯。在地址下拉列表中，选中 RT 地址后，必须选中“当前（RT）有效”选项，否则为此地址不参与通讯。在地址下拉列表中切换地址时，前一地址设置保存，显示新地址的设置。

“所有 RT 有效”选项，选中后，软件使能所有 0~30 的地址。

“所有 RT 无效”选项，选中后，软件取消使能所有 0~30 的地址。

“远程端子地址(SA)选择”选项，为了配合选择此子地址“数据发送模式”。

“数据发送模式”下拉列表中，可选择“单缓冲”或“循环缓冲”。“数据发送模式”的选择，是当前窗口中显示的 RT 和 SA 的数据发送模式。改变 RT 或 SA，前一“数据发送模式”设置保存，显示新的 RT、SA 的“数据发送模式”的设置。

“RT 状态字”栏，设置的是，当前显示 RT 地址的“RT 状态字”。改变 RT 地址，前一地址的“RT 状态字”设置保存，显示新 RT 的“RT 状态字”的设置。

子系统标志	影响 RtStatus[2]。选中后，RT 状态字中“子系统标志位”为 1；未选中为 0。
服务请求	影响 RtStatus[8]。选中后，RT 状态字中“服务请求位”为 1；未选中为 0。
动态总线控制接受	此项设置暂时无效，RtStatus[1]始终为 0。因为 RT 暂不支持“动态总线控制”功能。选中与否，“动态总线控制接受位”始终为 0。
终端标志	影响 RtStatus[0]。选中后，RT 状态字中“终端标志位”为 1；未选中为 0。
忙	影响 RtStatus[3]。选中后，RT 状态字中“忙位”为 1；未选中为 0。

“RT 矢量字”编辑窗口，设置当前显示 RT 地址的“RT 矢量字”。改变 RT 地址，前一地址的“RT 矢量字”设置保存，显示新 RT 的“RT 矢量字”的设置。

“RT 自检字”编辑窗口，设置当前显示 RT 地址的“RT 自检字”。改变 RT 地址，前一地址的“RT 自检字”设置保存，显示新 RT 的“RT 自检字”的设置。

其它选项设置：

同步时清除时标	设置，RT 接收到“同步”方式代码消息后，是否清零硬件 32 位时标。 选中后，RT 接收到“同步”指令，立即清零时标；未选中，RT 接收到“同步”指令，不清零时标。
同步时加载时标	设置，RT 接收到“带数据字的同步”方式代码消息后，是否清零硬件 32 位时标。 选中后，RT 接收到“带数据字的同步”指令，立即用消息所带唯一数据字(16 位)加载时标；未选中，RT 接收到“带数据字的同步”指令，不加载时标。
(RT)非法指令表使能	设置下方的“非法指令表”是否使能。 选中后，指令表使能；未选中，指令表不使能。
(RT)非法指令数据不接收	设置 RT 收到“非法消息”后，是否接收此“非法消息”中的数据。这项设置，必须在非法指令表使能的情况下有效。 选中后，非法消息的数据不接收；未选中，非法消息的数据接收。

非法指令表在默认情况下是不使能的。用户可以设置自己的非法指令表，来对 RT 的数据通讯工作进行控制，具体信息请参考“驱动程序使用说明”部分。这里的选择项针对的是数据量合法性。如果用户要让不同的 RT 与 SA 有不同的非法指令表设置，需要先将“所有 (RT) 均采用相同设置”与“所有 (SA) 均采用相同设置”置为“不选中”状态。如果这两项被选中，则程序会立即将所有 RT 或 SA 非法指令表项均设为相同设置，用户在更改其中一组 RT 与 SA 非法指令表设置时，也会将所有的 RT 与 SA 非法指令表设为相同设置。

用户可将 RT 设置保存成文件，方便下次直接加载：

保存(S)	点击“保存(S)”按钮，程序会将当前 RT 设置保存到用户指定的文件 (.rts) 中。
加载(L)	点击“加载(L)”按钮，程序会提示用户指定一个之前已经保存了的 RT 设置文件 (.rts)，确认后，程序会将文件中的设置信息读入到设置窗口中。

执行“应用”命令，程序会提交设置给硬件，并在软件中保存本次设置信息，以供下次开启该窗口时参考，如果执行“取消”或“关闭”命令，程序不会做任何操作。

4.3.6.2. 配置数据

图 4-18 RT 配置数据窗口

用户可通过操作“数据配置窗口”对远程终端的数据进行配置，配置的内容主要有数据量与数据内容。RT 提供 0 到 31 的地址，每个 RT 下都会有 1 到 30 的 SA 地址。

用户在选择 RT 与 SA 的同时，程序会自动地将设置保存到软件，也会将新的配置信息及时地显示在窗口中。

建议用户在操作时，先选择好 RT 与 SA 的地址，然后对数据量进行选择，数据量为 0 时，表明该 RT 下的 SA 没有数据，Demo 界面上，数据字最多可设置 256 个(实际每个子地址数据区大小为 4096 个字)。活用“生成随机数”与“生成顺序数”会为数据输入提供便利。

指定的 RT 子地址循环发送数据区的指针回卷深度（只在相应的发送子地址为“循环缓冲方式”下有效）。取值范围 0~4095。设置为 0 表示回卷深度为满 Size 量程 4096。当指针回卷深度不设置（默

认为 0) 或设置为 0，数据区发送到 4096 个数据后，指针回卷，从数据区起始位置开始继续发送；当指针回卷深度设置为 200，数据区发送到 200 个数据后，指针回卷，从数据区起始位置开始继续发送，而不是从 201 的位置开始发送。

用户可将 RT 发送数据保存成文件，方便下次直接加载：

保存(S)	点击“保存(S)”按钮，程序会将当前 RT 数据保存到用户指定的文件 (.rdt) 中。
加载(L)	点击“加载(L)”按钮，程序会提示用户指定一个之前已经保存了的 RT 数据文件 (.rdt)，确认后，程序会将文件中的数据读入到窗口中。

执行“应用”命令，程序会提交设置数据给硬件，执行“取消”或“关闭”命令，程序不会做任何操作。

4.3.6.3. 开启/停止接收消息



序号	消息描述字	时间标签(us)	发送方	接收方	指令字	CNT/MC	通道A/B	命令字&数据字
30	a000	0	RT2, Sa5	BC	14a0	32	B	14a0 2500 2501 2502 2503 2504 2505 2506
29	a000	0	RT2, Sa4	BC	1480	32	B	1480 2400 2401 2402 2403 2404 2405 2406
28	a000	0	RT2, Sa3	BC	1460	32	B	1460 2300 2301 2302 2303 2304 2305 2306
27	a000	0	RT2, Sa2	BC	1440	32	B	1440 2200 2201 2202 2203 2204 2205 2206
26	a000	0	RT2, Sa1	BC	1420	32	B	1420 2100 2101 2102 2103 2104 2105 2106

图 4-19 RT 开启/停止接收消息

操作与 4.3.6.3 相似。

4.3.6.4. 清空接收数据

该操作会将文件中接收的所有数据全部清除掉（后台文件接收方式），或将接收数据显示窗口中的所有数据全部清除掉（实时显示接收方式）。该操作在“开启接收”状态时无效。

4.3.6.5. 显示数据

程序将接收到的消息数据分类，以字段形式显示，如下所示：

字段名称	说 明
序号	按程序接收到消息的先后顺序编号，最新的消息会显示到接收窗口的最上方，起始编号为 1。
消息描述字	双击该项可显示当前消息的具体描述信息。（只有在停止接收状态下有效）。
时间标签(us)	显示消息通讯时的相对时间，单位为 1 微秒。
发送方	指消息数据的发送方。
接收方	指消息数据的接收方。
指令字	消息中的指令字。
CNT/MC	显示消息指令字中的“数据量/方式代码”数值。
通道 A/B	显示消息通讯时选用的通道。
命令字&数据字	命令字与数据字的合集。

备注	RT 到 RT 消息与消息出错提示，即该栏会显示两种信息，当用户看到消息出错信息时，可以通过查看“消息描述字”获取更多的信息。
----	---

4.3.7. 总线监控器（BM）操作

4.3.7.1. 设置过滤器



图 4-20 BM 过滤器窗口

提供总线监控器（BM）方式下的监视过滤信息，BM 只会监视设置了过滤信息类型的消息，过滤时依据的条件主要有 RT 地址与“发/收”操作，更多的信息请参考驱动程序使用说明部分。

用户可将 MT 过滤表保存成文件，方便下次直接加载：

保存(S)	点击“保存(S)”按钮，程序会将当前 MT 过滤表保存到用户指定的文件（.mtf）中。
加载(L)	点击“加载(L)”按钮，程序会提示用户指定一个之前已经保存了的 MT 过滤表文件（.mtf），确认后，程序会将文件中的设置读入到窗口中。

执行“应用”命令，程序会提交设置数据给硬件，执行“取消”或“关闭”命令，程序不会做任何操作。

4.3.7.2. 开启/停止数据接收消息

总线监控器(BM)-通道:1									
Rx Cx RM									
序号	消息描述字	时间标签(us)	发送方	接收方	指令字	指令字[2]	CNT/MC	通道A/B	命令字&数据字&RT状态字
30	a080	0	RT2, Sa5	BC	14a0		32	B	14a0 1000 2500 2501 2502 2503 2504 2505
29	a080	0	RT2, Sa4	BC	1480		32	B	1480 1000 2400 2401 2402 2403 2404 2405
28	a080	0	RT2, Sa3	BC	1460		32	B	1460 1000 2300 2301 2302 2303 2304 2305
27	a080	0	RT2, Sa2	BC	1440		32	B	1440 1000 2200 2201 2202 2203 2204 2205
26	a080	0	RT2, Sa1	BC	1420		32	B	1420 1000 2100 2101 2102 2103 2104 2105

图 4-21 BM 开启/停止数据接收消息

操作与 4.3.6.3 相似。

4.3.7.3. 清空接收数据

该操作会将文件中接收的所有数据全部清除掉（后台文件接收方式），或将接收数据显示窗口中的所有数据全部清除掉（实时显示接收方式）。该操作在“开启接收”状态时无效。

4.3.7.4. 显示数据

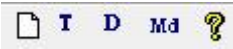
程序会将及时接收到的消息数据信息分类显示，以字段形式显示，如下所示：

字段名称	说 明
序号	按程序接收到消息的先后顺序编号，最新的消息会显示到接收窗口的最上方，起始编号为 1。
消息描述字	双击该项可显示当前消息的具体描述信息。（只有在停止接收状态下有效）。
时间标签(us)	显示消息通讯时的相对时间，单位为 1 微秒。
发送方	指消息数据的发送方。
接收方	指消息数据的接收方。
指令字	消息中的指令字。
指令字（2）	当消息为 RT 到 RT 的类型时，该位置显示消息中的第二个指令字。
CNT/MC	显示消息指令字中的“数据量/方式代码”数值。
通道 A/B	显示消息通讯时选用的通道。
命令字 & 数据字 & RT 状态字	命令字、数据与 RT 状态字的合集。
备注	RT 到 RT 消息与消息出错提示，即该栏会显示两种信息，当用户看到消息出错信息时，可以通过查看“消息描述字”获取更多的信息。

4.3.8. 工具条命令

4.3.8.1. MF1553 主窗口工具条命令

主工具栏：



功能：

	新建工作模式	创建 BC、RT、BM 操作（工作模式）。
	设置时标与超时	显示“启动/停止时标与应答超时设置窗口”。
	数字量 I/O	显示数字量 I/O 窗口。
	接收消息方式	选择接收消息方式为以下两种选择： ① 后台文件接收（默认）； ② 实时显示接收。
	帮助主题	显示产品型号信息、版本信息、公司信息。

4.3.8.2. 总线控制器（BC）工具条命令

总线控制器工具栏：



总线控制器工具栏中的操作与总线控制器菜单命令功能相对应。

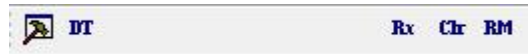
	设置	显示设置窗口。
	消息	显示消息配置窗口。
	组微帧	显示微帧配置窗口。
	组主帧	显示主帧配置窗口。
	停止 BC	停止 BC 操作。
	运行 BC	启动 BC 操作。
	动态控制	显示动态控制窗口。
	接收/停止接收	开启接收/停止接收操作。
	清空接收数据	清空接收数据显示列表及状态栏。
	打开接收文件	打开接收消息的文件。（只在文件接收消息模式下使用）

注：

操作菜单中的“设置”、“消息”、“组微帧”与“组主帧”时，需要处于“停止 BC”状态。
操作菜单中的“清空接收数据”与“打开接收文件”时，需要处于“停止接收”状态。

4.3.8.3. 远程终端（RT）工具条命令

远程终端工具栏：



远程终端工具栏中的操作与远程终端菜单命令功能相对应。

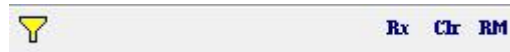
	设置	显示设置窗口。
	数据配置	显示数据配置窗口。
	接收/停止接收	开启接收/停止接收操作。
	清空接收数据	清空接收数据显示列表及状态栏。
	打开接收文件	打开接收消息的文件。（只在文件接收消息模式下使用）

注：

操作菜单中的“清空接收数据”与“打开接收文件”时，需要处于“停止接收”状态。

4.3.8.4. 总线监控器（BM）工具条命令

总线监控器工具栏：



总线监控器工具栏中的操作与总线监控器菜单命令功能相对应。

	过滤器	显示过滤设置窗口。
	接收/停止接收	开启接收/停止接收操作。
	清空接收数据	清空接收数据显示列表及状态栏。
	打开接收文件	打开接收消息的文件。（只在文件接收消息模式下使用）

注：

操作菜单中的“清空接收数据”与“打开接收文件”时，需要处于“停止接收”状态。

4.3.9. 帮助

软件提供软件的版本和公司信息，如遇到技术等问题，可与我公司人员联系。

4.3.10. 退出程序

建议用户停止并关闭当前所有的工作模式操作，然后执行“工作模式”菜单中的“退出”命令，

或点击主窗口右上角的“关闭按钮”，可以退出程序。

4.4. 配置信息的保存和加载

软件共包括 BC、RT、MT 三种工作模式，每种工作模式的设置窗口都有“保存”和“加载”数据的功能，包括以下界面：

- ① BC 消息配置窗口；
- ② BC 微帧配置窗口；
- ③ BC 主帧配置窗口；
- ④ RT 设置窗口；
- ⑤ RT 数据配置窗口；
- ⑥ BM 过滤设置窗口。

有效地使用配置数据文件操作，可以提高硬件功能配置的效率，也会减少用户重复地进行复杂的配置工作的次数。