上海格西信息科技有限公司

消息监听和篡改例子

版本 0.1

目录

1.	概述	3
2.	创建项目	4
	2.1 第1步添加串口设备 2.2 第2步添加变量	4 4
	2.3 第3步 添加序列	5
	2.4 第4步添加界面	8
3.	运行项目	9
	3.1 打开项目	9
	3.2 运行项目	9
	3.2.1 第 1 步 正常运行	9
	3.2.2 第 2 步 篡改指定 Type ID 的报文并转发	10

1. 概述

MASTER 设备和 SLAVE 设备通过串口连接,串口设置默认为波特率 57600,数据位 8,停止位 1,无 校验。设备间的通信协议结构和通信命令如下所示。



命令类型 Type-Identifier	名称	描述
80H	Link check	For checking link.
81H	Data Request	Request SLAVE for Data.
82H	Upload result	Ack. To the above Data Request Command.

命令 Link check 请求帧	命令 Link Check 响应帧
BLOCK 0	BLOCK 0
TI – 80H	TI – COH
RECORD LENGTH = 07H	RECORD LENGTH = 07H
BLOCK 1	BLOCK 1
{No DATA BYTE is available}	{No DATA BYTE is available}
BLOCK 2	BLOCK 2

命令 Data Request 请求帧	命令 Data Request 响应帧
BLOCK 0	BLOCK 0
TI – 81H	TI – C1H
RECORD LENGTH = 07H	RECORD LENGTH = N+07H
BLOCK 1	BLOCK 1

{No DATA BYTE is available}	(N Bytes)
	{Length is Variable}
BLOCK 2	BLOCK 2

命令 UpLoad Result 帧(无回复)

BLOCK 0
TI – 82H
RECORD LENGTH = $07H$
BLOCK 1
NO DATA BYTES
BLOCK 2

本演示例子, 演示如何监听和篡改 MASTER 设备和 SLAVE 设备之间的通信数据。



本例子文件位于: 〈软件安装目录〉\Examples\Solutions\ProtocolSimulation\MessageMonitor。

文件说明:

✓ MessageMonitor.gpj - 通信协议监听和篡改演示项目 - 中文 - 串口版

例子自带仿真器,可以脱离设备仿真运行。

串口版: 需要使用串口虚拟软件,如 VSPD 等,虚拟出两对串口(一对为 COM2 和 COM3,一对为 COM4 和 COM5)进行仿真运行。如果虚拟的串口号和例子预定义的串口号不同,可以修改例子串口号,也可以修改虚拟串口号。

2. 创建项目

2.1 第1步 添加串口设备

本演示项目需要和两台设备连接,分别是 MASTER 设备和 SLAVE 设备,对应"MASTERCOM"和 "SLAVECOM";另外,由于本项目仿真了 MASTER 设备和 SLAVE 设备,分别占用一个串口,对应 "MASTERCOM-SIM"和"SLAVECOM-SIM"。

"MASTERCOM"和"MASTERCOM-SIM"为VSPD 虚拟的一对串口 COM2 和 COM3, 虚拟交叉线连接, 即 COM2 和 COM3 可以互相通信; "SLAVECOM"和"SLAVECOM-SIM" 同理。

2.2 第2步 添加变量

建立变量组,保存通信过程的关键数据,作为转发或者篡改的中介桥梁。

*		格西测控大师						- • ×
「目・「首页」「视图」」	具 帮助						a 🚨 A	dministrator 🔹
 ② ○ ○	增切 貢制 撤销重做 上移下 编辑	移 网格线 常用						
▲ 项目管理器 平 ×	CommandVars ×				-	×	🕺 工具箱	ųх
捜索项目管理器 ▼	名称	数值类型	数值	数值格式	说明		搜索工具箱	- 🗄 -
🧰 通信协议监听和篡改 🔺 📤	👻 📃 CommandVars	Null	_			1	🗌 常用变量	~
✓ ■ 设备与接口	LENGTH	Int16	0					
T MASTERCOM	TYPE-ID	BitString	0x00			.		
SLAVECOM	SOURCE	BitString	0x00				83 BM	
MASTERCOM-SIM	DESTINATION	BitString	0x00				2337届1王	+ ^
SLAVECOM-SIM	PORT-NO	BitString	0x00				(Name)	Command
▼ 🔓 变量	SEQ-NO	Byte	0				Capacity	0
CommandVars	🧼 DATA	BitString	0x00				Descript	
▼ 🎦 序列								
> 🔓 监听和篡改								
> 🔓 仿真MASTER设备								
▶ 🔓 仿真SLAVE设备								
> 🍃 启动 🚽						-		
就绪							·	.::

2.3 第3步 添加序列

本演示项目建立四个序列来实现。

1) "启动"序列:通过脚本实现自动化配置;同时,监听的全局类 Monitor Master Global 也在这 里实现。

```
using System;
using Genesis;
using Genesis. Scripting;
using Genesis.Sequence;
using Genesis. Workbench;
using Genesis.Device;
public class Step C920C3896BD342DEB36C7B7ABA721631
{
  public ScriptContext Context { get; set; }
  11
  public Int32 BeginExecute(IStepContext context, IStep step)
  {
    MonitorMasterGlobal. Reset();
    11
    IDeviceSession dev = this.Context.GetDeviceSession("MASTERCOM");
    if (!dev. State)
    {
      dev. Open();
    }
```

```
dev = this.Context.GetDeviceSession("SLAVECOM");
    if (!dev. State)
    {
      dev. Open();
    dev = this.Context.GetDeviceSession("MASTERCOM-SIM");
    if (!dev. State)
    {
      dev. Open();
    dev = this.Context.GetDeviceSession("SLAVECOM-SIM");
    if (!dev. State)
    {
      dev. Open();
    // 启动监听和篡改序列和仿真 SLAVE 设备序列
    this. Context. StartStep("监听和篡改");
    this. Context. StartStep("仿真 SLAVE 设备");
    this. Context. OpenStepDataEditor();
    this. Context. OpenSchema("报文设置界面");
    return 0:
  }
  11
 public Int32 EndExecute(IStepContext context, IStep step)
   return 0;
  }
}
// 监听全局配置类, 用于 MASTER 转至 SALVE 的报文修改
public class MonitorMasterGlobal
 public static Int16 Length {get;set;}
  public static byte Source {get;set;}
 public static byte Destination {get;set;}
 public static byte PortNo {get;set;}
  public static byte SeqNo {get;set;}
 public static string Data {get;set;}
  public static bool CanModifyLength {get;set;}
  public static bool CanModifySource {get;set;}
 public static bool CanModifyDestination {get;set;}
  public static bool CanModifyPortNo {get;set;}
 public static bool CanModifySeqNo {get;set;}
 public static bool CanModifyData {get;set;}
```

```
public static void Reset()
{
    CanModifyLength = false;
    CanModifySource = false;
    CanModifyDestination = false;
    CanModifyPortNo = false;
    CanModifySeqNo = false;
    CanModifyData = false;
}
```

2) "监听和篡改"序列: 该序列是核心序列,实现监听、转发和篡改功能。本例子的通信协议只 有一个协议结构,故只需要两个并行分支,一个分支处理并转发来自 MASTER 设备的报文,一个分支处 理并转发来自 SLAVE 设备的报文。

转发和篡改是以变量为桥梁,通过脚本实现。

总的流程:收到报文->通过绑定协议字段的方式提取数据字段存入对应的变量->根据篡改配置情况设置转发帧对应的协议字段->转发。



3) "仿真 SLAVE 设备"序列: 该序列实现 SLAVE 设备的仿真,需要两个并行分支,一个分支处理 来自 MASTER 的 LinkCheck 命令,一个分支处理来自 MASTER 的 DataRequest 命令。

*	格西测控大师		X • -				
「目・「首页」 视图 コ	具帮助		a 🙎 Administrator 🝷				
● ●<	 ● ●						
保	编辑 □ 仿直SLΔVF设备 X	<u>亚</u> 示 寻山 ▼ X					
▲ · · · · · · · · · · · · · · · · · · ·	名称	说明					
💼 通信协议监听和篡改 🔷	✓ <a>B 仿真SLAVE设备	^	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□				
> 🔜 设备与接口	✓ ♣ Parallel						
▼	 Branch (CmdLinkCheck) 	并行分支:处理来自MAST					
> CommandVars	🗸 🆏 While						
┃ ▼ № 序列	CmdLinkCheck.request	来自MASTER	1237届1日 平 人				
	CmdLinkCheck.response		(Id) 46EEAC9				
	✓ In Branch (CmdDataRequest)	并行分支:处理来自MAST	(Name) 仿真SLAV				
	🗸 🌄 While		Active 🗹				
	CmdDataRequest.reques	来自MASTER	Categ				
	CmdDataRequest.respon		Descri				
	CmdUploadResult.ack.M	来自MASTER	Logga 🔽				
			Seque Random				
		-	Socket -1				

4) "仿真 MASTER 设备"序列: 该序列实现 MASTER 设备的仿真, MASTER 设备为主动设备, 每个 命令单独实现即可。执行的时候,可以单独执行一个命令,也可以连续执行多个命令。

*	格西测控大师						- Ο Σ	x
目・ 首页 视图 エ	具 帮助					a 🙎	Administrator	•
● 日 ◎ × 5 运行 添加 保存 粘贴 × 5	常切 夏制 散销 重做 上移 編辑	下移 脚本 协议	四格线 常用文	4				
▲ 项目管理器 平 ×	B 仿真MASTER设备 ×			*	×	% 工具箱	д р	×
搜索项目管理器 ▼	名称		说明			搜索工具箱	-	-
■ 通信协议监听和篡改 ▲	✓ 📴 仿真MASTER设备				•	<mark>告</mark> 常用步骤		
▶ ■ 设备与接口	CmdLinkCheck	: vock roguost	模仿MASTER设备2	司达LinkCh				
▼ 🛑 变量		eck.response						
> 📙 CommandVars	🖾 Wait					☺ 属性	д р	×
	🗸 🔓 CmdDataRequ	est	模仿MASTER设备	发送DataRe		(Id)	188CCC8	
> Ca 仿真MASTER设备	🖓 CmdDataRe	equest.request				(Name)	仿真MAS	
> 🔓 仿真SLAVE设备	🕼 CmdDataRe	equest.response				Active		
> 🎦 启动	🖓 CmdUpload	dResult.ack	Ack. To the above	Data Req		Categ		≣
▼ 🐻 画面	🔀 Wait					Descri	7	
📑 报文设置界面						Seque	Random	
						Socket	-1	
					•			•
就绪								.::

2.4 第4步 添加界面

用户画面主要用来配置篡改信息的。通过脚本把参数设置到全局类 MonitorMasterGlobal 中。

3. 运行项目

3.1 打开项目

从<软件安装目录>\Examples\Solutions\ProtocolSimulation\MessageMonitor目录中,打开 MessageMonitor.gpj串口版项目文件。



3.2 运行项目

3.2.1 第1步 正常运行

点击工具栏的"运行"按钮,进入运行模式,然后点击工具栏"序列数据"按钮,打开序列结果数据页面。

以上准备就绪,即可通过手工运行的方式,运行"仿真 MASTER 设备"序列。用鼠标右键弹出菜单,如下图。

♦ 格西测控大师 □ 回 XX								
■· 首页 视图 工具	帮助					a 🚨 Ad	ministrat	tor •
☆ ↓	3 日本			SData文件	常用文件			
	=			84	-			
	◎ 通信协议监听和第	业小 i改.Data II 报文设置	現面		4			• ×
搜索项目管理器	序号	名称	1	起始时间	执 (ms)	设备	状态	
፼ 通信协议监听和篡改 ▲	> 13	仿真SLAVE设备	1	20:19:23.923	0			
	✓ ¹ / ₂	监听和篡改	1	20:19:23.922	0			
MASTERCOM	~ 晶	Parallel	1	20:19:23.935	0			
SLAVECOM	✓ 1	Branch (MASTER request)		20:19:23.941	0			
MASTERCOM-SIM	~ %	While	1	20:19:23.941	0			
SLAVECOM-SIM	£67	CommandPacket.request.M 20:19:24.034		0	MASTERC			
▶ 盲 变量	> Þ	Branch (SLAVErespo	nse)	20:19:23.941	0			
CommandVars								
🗸 🖌 序列								
> 📴 监听和篡改 🛛 🏲								
✓ I 仿真MASTER设备	运行							
> 📴 CmdLinkCheck	循环法行							_
🔬 Wait ∾	法行之师	(7.5h		42. 444.42				
> 📴 CmdDataRequest 👒	运1]]秋	(100) 通け	3	叙 / 佃还	-			
		● 0 (0%) 進过 ● 0 (0%) 生附						
	切換激活	● 0 (0%) 火火						
 ✓ □ □ □ □ □ ✓ □ □ □ □ 		● 0 (0%) 完成						
□ 报文设置界面					-			-
就绪						第0行:	共16行	:

3.2.2 第2步 篡改指定 Type ID 的报文并转发

本演示例子只演示如何篡改从 MASTER 设备到 SLAVE 设备的指定报文,一次触发设置修改,只对下一条符合 Type ID 的报文进行修改设置,清空的参数不进行修改设置。

篡改参数设定后,下一次触发的数据,可以到"序列数据"页面查看相关命令的篡改情况,一般 篡改后,SLAVE 仿真设备不认得,故不会响应,从而导致 MASTER 侧的收不到响应帧而失败。

