

模拟电子线路自动测试程序 使用和再开发参考文档

华中科技大学电工电子科技创新中心 王贞炎
2022 年 4 月 25 日

目录

第 1 章 程序使用	4
1.1 程序简介.....	4
1.2 程序界面.....	4
1.3 程序使用例子.....	6
1.4 仪器设定.....	10
1.4.1 输入耦合设定.....	10
1.4.2 垂直设定.....	10
1.4.3 水平设定.....	11
1.4.4 触发设定.....	11
1.4.5 采样设定.....	12
1.4.6 自动设定.....	13
1.5 测量量设定.....	13
1.6 测量通道设定.....	15
1.7 分数计算公式.....	15
1.7.1 支持的函数.....	15
1.7.2 支持的运算符.....	16
1.7.3 支持的变量.....	17
1.7.4 分数计算公式示例.....	17
1.8 CSV 文件支持	18
1.8.1 通过 Excel 编辑文件	18
1.8.2 通过文本编辑器编辑文件.....	19
第 2 章 程序的再开发	21
2.1 程序开发环境.....	21
2.2 程序结构.....	21
2.2.1 FormMain 类字段.....	22
2.2.2 FormMain 类方法.....	23
2.2.3 GwScpi 类依赖.....	26
2.2.4 GwScpi 类字段.....	27
2.2.5 GwScpi 类方法.....	27
2.2.6 FFT 类.....	31

2.2.7 ExprAnalyse 类.....	31
2.2.8 EngineerFormatExt 方法扩展	31
2.3 再开发关键点.....	31
2.3.1 测量通道设定拓展.....	31
2.3.2 测量量拓展.....	32
2.3.3 分数计算公式拓展.....	32
2.4 关键代码.....	32
2.4.1 GwScpi.Command	32
2.4.2 GwScpi.AcquireMemory	33
2.4.3 GwScpi.ParseAndSetFromString	33
2.4.4 GwScpi.Measure	36
2.4.5 FormMain.DoMeasureRow.....	37
第 3 章 接口拓展板	39
3.1 硬件电路.....	39
3.1.1 MCU 部分	39
3.1.2 电源电流测量部分.....	40
3.1.3 通道扩展部分.....	41
3.1.4 UART-USB 接口部分.....	41
3.1.5 电源部分.....	42
3.2 与计算机的接口.....	42
3.2.1 接口和驱动.....	42
3.2.2 命令格式.....	43
3.3 MCU 程序设计	43

第1章 程序使用

1.1 程序简介

该程序可用于模拟电子线路实验和实作作品的自动测试。测试过程依赖一台带有以太网接口并支持 SCPI 指令接口的多功能示波器。需将作品和示波器连接，并将测试项目信息填写到程序界面之中，之后程序便可依据填写的测试项目逐条自动完成测试并根据填写的评分公式评分。如果待测试的端口多于示波器的模拟输入端口，还可另接程控集线板（通道拓展板）。

程序采用 C#语言和微软“.Net 6.0”框架开发，可运行于 Windows 7 或更新的 Windos 操作系统上，部分操作系统可能需要另行安装“.Net 6.0”运行时库，可通过微软官方网站下载（<https://dotnet.microsoft.com/en-us/download/dotnet/6.0>）。

1.2 程序界面

程序是一个单界面程序，如图 1 所示。界面中主要包含作品列表和测试项列表两个部分。



图 1 程序界面

作品列表列出待测试的作品，每行一个作品，各列为：

- 序号；
- 作品编号，也可以是学生学号或姓名等信息；
- 仪器 IP，与该作品连接的仪器的 IP 地址
- 仪器端口，与该作品连接的仪器的 IP 端口；

- 仪器 ID，主要用于指示仪器的连接状态；
- 总分算式，根据各个测试项目得分计算总分的公式，如有需要也可包含各个测试项目的测量值；
- 得分。

评分项目列表列出对每个作品做测试时需要测试的项目，每行一个评分项目，各列分别为：

- 序号；
- 测试项目描述；
- 提示信息，如果有提示信息，程序在进行此项测试之前会弹出对话框提示操作；
- 测试通道；
- 测试设定；
- 测量量；
- 分数算式，通过本项目的测试值计算项目得分的公式；
- 测量值；
- 得分，由测量值和得分计算公式计算得到的分数。

在作品列表上方有一些控件：

- 作品数量，用于设置作品数量，即作品列表的行数，增加时，作品列表将增加行，减少时，作品列表中最下方多出的行会被删除；
- 导出为文件按钮，将列表导出到外部 CSV 文件；
- 从文件导入按钮，从外部 CSV 文件导入列表；CSV 格式全称为逗号分隔的列表文件，可通过 MS Office Excel、LibreOffice Spreadsheet、Apple Numbers 或文本编辑器编辑。
- 复制公式到所有行按钮，将选中的一个分数算式单元格中的内容复制到其他所有分数算式单元格中；
- 测试仪器连接按钮，用于测试与仪器的以太网连接；
- 对选中的作品评分按钮，开始完整的一个或多个作品的测试流程。

在评分项目列表上方有一些控件：

- 评分项目数量，用于设置评分项目数量，即评分项目的行数，增加时，评分项目列表将增加行，减少时，评分项目列表最下方多出的行会被删除。
- 导出为文件按钮，将列表导出到外部 CSV 文件；

- 从文件导入按钮，从外部 CSV 文件导入列表；CSV 格式全称为逗号分隔的列表文件，可通过 MS Office Excel、LibreOffice Spreadsheet、Apple Numbers 或文本编辑器编辑；
- 外部集线板端口文本框，用于填写外部集线板的串行通信端口名；
- 测试选中的评分项按钮，使用作品列表中的第一个选中作品测试选中的评分项目，主要用来测试评分项目设置的正确性。

1.3 程序使用例子

程序的使用流程可分为以下几步：

- 编辑评分项目，根据作品测试所需的项目数量，设置评分项目列表的行数，然后在列表中编辑各个单元格，这一步也可直接从外部文件导入。
- 编辑作品列表，根据作品数量设置作品列表的行数，然后在列表中编辑各个单元格，这一步也可直接从外部文件导入。
- 测试仪器连接和测试评分项目，通过连接测试按钮和评分项目测试按钮验证仪器的以太网连接、仪器和作品的连接以及作品列表条目和评分项目条目的正确性。

这里以一个简单的例子说明程序的使用过程。这个例子将对一个偏置 2.5V、幅值 2.5V、频率 19k~21kHz 可调的方波进行以下几个项目的测试：

1. 由通道 1 输入的方波的波形质量；
2. 由通道 1 输入的方波的峰峰值；
3. 由通道 1 输入的方波的最小频率；
4. 由通道 1 输入的方波的最大频率。

被测作品有 1 个，与之连接的仪器 IP 地址为 “192.168.2.2”，端口为 3000。

首先，使用作品列表上方的作品数量数字框中调整作品数量为 1，大于 1 也可，然后在第一行的各列中：

- 序号列，将自动生成序号；
- 作品编号列，填入例如 “U202012345” 或 “#123” 之类的文本，作品编号列仅用于命名对应作品，程序运行时并不使用它；
- 仪器 IP 列，填入与作品连接，用于测试作品各项指标的仪器的 IP 地址，例如 “192.168.2.2”；
- 仪器端口列，填入与作品连接的仪器的端口，通常为 “3000”；

- 连接状态列，不可编辑，经过连接测试、指标测试或正式测试的作品所连接的仪器的状态将显示在此列，如果正常，将显示通过 SCPI “*IDN?” 指令查询到的仪器型号信息，如果不正常，将显示“连接失败”；
- 分数算式列，填入分数计算式，可以以评分项目中的分数为变量，例如填入“s1+s2+s3+s4”；
- 分数列，该列不可编辑，进行测试程序后将自动计算结果。

经过上述设置后的作品列表将如图 2 所示。

	序号	作品编号	仪器IP	仪器端口	仪器ID	总分算式	分数
	1	U00000001	192.168.2.2	3000		s1+s2+s3+s4	
▶	2	U00000002	192.168.2.3	3000			
	3	U00000003	192.168.2.4	3000			

图 2 作品列表

然后在评分项列表上方的评分项数量数字框中调整评分项数量为 4，大于 4 也可，然后在第一行的各列中：

- 序号列，将自动生成序号；
- 测试项目描述列，填入对此项测试的描述，例如：“方波信号的波形质量”，该项仅用于给用户提示，程序并不使用它；
- 提示信息列，填入此项的提示信息，因为在此项测试之前并不需要学生调整作品，不需要提示，这里留空即可；
- 测量通道列，填入被测方波连接到示波器的通道，这里填“1”，如果由外部集线板，可填“1,2”表示由外部集线板通道 2 连接到示波器通道 1；
- 仪器设定，填入对仪器进行设定的脚本字符串，这里填入“DC, 1V/div, -2.5Vpos, 100us/div”，表示将通道 1 设置为直流耦合输入，垂直档位为 1V 每格，垂直偏移为-2.5V，水平档位为 100μs/div；详细的仪器设定将在后文讲解；
- 测量量，填入“Rect”，可不区分大小写，表示测量方波的波形质量，其它支持的测量量将在后文讲解；
- 测量值，不可编辑，该项测试后，程序将自动填入；
- 得分算式，填入根据测量值计算得分的表达式，测量值以“x”替代，例如这里可以填入“2.5*x”，方波波形质量的测试结果为值域[0,1]的实数，“2.5*x”将使得得分结果在 0~2.5 分之间；
- 得分，不可编辑，该项测试后，程序将自动计算。

在第二行，各列：

- “2”，自动生成的序号；
- “方波信号的峰峰值”；
- “”，留空；
- “1”，使用通道 1；
- “2.5VtrigR”，设定上升沿触发，出发电平为 2.5V；
- “P2P”，表示测量峰峰值；
- “”，空测量值，不可编辑；
- “ $\text{sat}((x-4.5)*2.5/0.5, 0, 2.5)$ ”，表示满分为 2.5 分，最低为 0 分，测到 5Vp-p 时可得满分，如果小于等于 4.5Vp-p 则得 0 分，此间得分和峰峰值为线性关系；
- “”，空分数，不可编辑。

在第三行，各列：

- “3”，自动生成的序号；
- “方波信号的最小频率”；
- “调整信号频率到 19kHz 或更小”，进行此项测试之前，程序将弹出对话框，提示“调整信号频率到 19kHz 或更小”，在单击对话框中的“OK”按钮之后，测试才会继续；
- “1”，使用通道 1；
- “”，留空，经过前面的设定，这一项不需要另行设定；
- “Freq”，表示测量频率；
- “”，空测量值，不可编辑；
- “ $\text{sat}((19.5\text{e}3-x)*2.5/500, 0, 2.5)$ ”，表示满分为 2 分，最低分 0 分，测到 19000Hz 或以下时可得满分，如果大于等于 19500Hz 则得 0 分，此间得分和频率为线性关系；
- “”，空分数，不可编辑。

第四行，各列：

- “4”，自动生成的序号；
- “方波信号的最大频率”；
- “调整信号频率到 21kHz 或更大”，进行此项测试之前，程序将弹出对话框，提示“调整信号频率到 21kHz 或更大”，在单击对话框中的“OK”按钮之后，测试才会继续；
- “1”，使用通道 1；
- “”，在上一次测试后，不需要再调整仪器设定；
- “Freq”，表示测量频率；

- “”，空测量值，不可编辑；
- “ $\text{sat}((x-20.5\text{e}3)*2.5/500, 0, 2.5)$ ”，表示满分为 2 分，最低分 0 分，测到 21000Hz 或以上时可得满分，如果小于等于 21500Hz 则得 0 分，此间得分和频率为线性关系；
- “”，空分数，不可编辑。

经过上述设定后的评分项目列表如图 3 所示。

	序号	测量项目描述	提示信息	测量通道	仪器设定	测量量	分数算式	测量值	分数
	1	方波信号的占空比		1,1	DC, 1V/div, -2.5Vpos, 100us/div	Rect	$x*2.5$		
	2	方波信号的幅值		1,1	2.5VtrigR	P2P	$\text{sat}((x-2.2)*2.5, 0, 2.5)$		
	3	方波信号的最小频率	调整信号频率到19kHz或更小。	1,1		Freq	$\text{sat}((19.5\text{e}3-x)*2.5/500, 0, 2.5)$		
	4	方波信号的最大频率	调整信号频率到21kHz或更大。	1,1		Freq	$\text{sat}((x-20.5\text{e}3)*2.5/500, 0, 2.5)$		
	5								

图 3 评分项目列表

然后，可在作品列表左侧单击选中第一行整行，然后单击作品列表右上方的“开始测试评分”按钮，程序将对该作品进行上述 4 个评分项的测试和分数计算，期间会弹出两次对话框，如图 4 和图 5 所示分别是上述第 3 项和第 4 项评分项目中设定的提示信息。

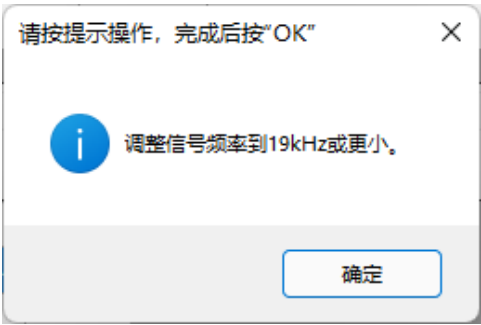


图 4 测量最小频率时的提示



图 5 测量最大频率时的提示

测试完成后，程序界面如图 6 所示。



图 6 完成实例测量过程后的界面

1.4 仪器设定

仪器设定字符串用于填写评分项中的“仪器设定”列，仪器设定字符串中可包含多个设定项，每个设定项之间以英文半角逗号（“,”）分割，设定项不分区大小写。

1.4.1 输入耦合设定

输入直流耦合，设定项格式为：DC。

输入交流耦合，设定项格式为：AC。

1.4.2 垂直设定

垂直档位，设定项为：<电压值>V/div。

其中<电压值>为以 V 为单位的电压值，不可为负值，可包含国际单位制词头：m、μ、n、p、f、a、k、M、G、T 等，其中“μ”因输入不便，也可用“u”替代。

例如：

- “1V/div”表示垂直档位 1 伏每格；
- “20mV/div”表示垂直档位为 20 毫伏每格；
- “50e-3V/div”都表示垂直档位为 50 毫伏每格。

注意，不能设置为示波器不支持的档位值。

垂直位置，设定项为：<电压值>Vpos。

其中<电压值>为以 V 为单位的电压值，可包含国际单位制词头：m、 μ 、n、p、f、a、k、M、G、T 等，其中“ μ ”因输入不便，也可用“u”替代，可包含负号，负号为英文半角连字符，不可用 Unicode 标准负号。

例如：

- “-500mVpos” 表示垂直位置为-500 毫伏。
- “4e-1Vpos” 表示垂直位置为 0.5 伏。

1.4.3 水平设定

水平档位，设定项为：<时间值>s/div。

其中<时间值>为以秒为单位的时间，可包含国际单位制词头：m、 μ 、n、p、f、a、k、M、G、T 等，其中“ μ ”因输入不便，也可用“u”替代。

例如：

- “2us/div” 表示水平档位为 2 微秒每格；
- “5e-2s/div” 表示水平档位为 50 毫秒每格。

水平位置，设定项为：<时间值>spos。

其中<时间值>为以秒为单位的时间，可包含国际单位制词头：m、 μ 、n、p、f、a、k、M、G、T 等，其中“ μ ”因输入不便，也可用“u”替代。

例如：

- “120uspos” 表示水平位置为 120 微秒；
- “5e-7spos” 表示水平档位为 500 纳秒。

1.4.4 触发设定

上升沿触发，设定项为：<电压值>VtrigR。

其中<电压值>为以 V 为单位的电压值，可包含国际单位制词头：m、 μ 、n、p、f、a、k、M、G、T 等，其中“ μ ”因输入不便，也可用“u”替代，可包含负号，负号为英文半角连字符，不可用 Unicode 标准负号。

例如：

- “0VtrigR” 表示上升沿触发，电平为 0V。
- “-10mVtrigR” 表示上升沿触发，电平为-10 毫伏。

下降沿触发，设定项为：<电压值>VtrigF。

其中<电压值>为以 V 为单位的电压值，可包含国际单位制词头：m、 μ 、n、p、f、a、k、M、G、T 等，其中“ μ ”因输入不便，也可用“u”替代，可包含负号，负号为英文半角连字符，不可用 Unicode 标准负号。

例如：

- “0VtrigF”表示下降沿触发，电平为 0V。
- “-10mVtrigF”表示下降沿触发，电平为-10 毫伏。

触发源，设定项为：TrigSrc:[CH1|CH2|CH3|CH4|D<num>]

其中 CH1~4 和 D? 可选其一。

例如：

- “TrigSrc:CH2”表示选择通道 2 为触发源；
- “TrigSrc:D7”表示选择数字通道 7（如有）为触发源。

触发源耦合，设定项为：TrigCoup:[AC|DC|HF|LF]

其中 AC（交流耦合）、DC（直流耦合）、HF（高频抑制）、LF（低频抑制）四者选其一。

例如：

- “TrigCoup:AC”表示选择触发源为交流耦合；
- “TrigCoup:HF”表示选择触发源为高频抑制。

触发模式，设定项为：TrigMode:[Aut|Norm]

其中 Auto（自动）和 Normal（常规）二者选其一。

例如：

- “TrigMode: Aut”表示选择触发模式为自动。

1.4.5 采样设定

采样平均，设定项为：<平均次数>Avg

其中<平均次数>为整数，并且必须是能被仪器所支持的。

如果平均次数为 0 或 1，则将切换至采样模式。

通常平均次数为 2、4、8、16、32、64、128、256 等等。

例如：

- “0Avg”、“1Avg”均表示不做采样平均；
- “16Avg”表示 16 帧平均。

采样深度，设定项为：<采样数>Smps

其中<采样数>为整数，可采用科学计数法或带有国际单位制词头，并且必须是能被仪器所支持的。

通常，采样数为 1k，10k，100k，1M 等等。

例如：

- “10kSmps”，表示采样深度为 1 万。

1.4.6 自动设定

自动设定，设定项为：Auto

使用该设定，将应用示波器的自动设置，如同在示波器上按下 AutoSet 按键。

1.5 测量量设定

测量量设定字符串用于填写评分项中的“测量量”列，不区分大小写，测量量设定字符串可以是：

- rect，表示测量方波的波形质量，测量值区间为[0,1]，1 表示质量很好，0 表示质量很差。方波的波形质量由方波的失真度和方波的上升下降时间对周期的占比来评估，具体评估指标如下：

代码 1 方波质量评估分段代码

```
if (distErr < 0.03 && rfDuty < 0.05)
    return 1.0;
else if (distErr < 0.05 && rfDuty < 0.08)
    return 0.9;
else if (distErr < 0.08 && rfDuty < 0.10)
    return 0.8;
else if (distErr < 0.10 && rfDuty < 0.12)
    return 0.7;
else if (distErr < 0.15 && rfDuty < 0.15)
    return 0.6;
else if (distErr < 0.2 && rfDuty < 0.2)
    return 0.5;
else
    return 0.0;
```

其中，distErr 为失真度偏差，它等于测得的失真度与标准方波失真度（48.3%）之差的绝对值；rfDuty 等于上升时间和下降时间之和与周期之比。

- **dist**, 表示测量失真度, 测量值为谐波总功率的平方根和基波总功率的平方根之比, 失真度是通过获取仪器采集到的原始数据, 去除直流成分后, 通过傅里叶分析和功率密度谱分析得到的, 分析长度为 8192, 采用系数为 10 的凯泽尔窗加窗处理。
- **amp**, 表示测量幅值, 测量值单位为 V。
- **p2p**, 表示测量峰峰值, 测量值单位为 V。
- **mean**, 表示测量平均值, 测量值单位为 V。
- **cmean**, 表示测量周期平均值, 测量值单位为 V。
- **rms**, 表示测量真有效值, 测量值单位为 V。
- **crms**, 表示测量周期真有效值, 测量值单位为 V。
- **freq**, 表示测量频率, 测量值单位为 Hz。
- **tfreq**, 表示测量触发频率, 测量值单位为 Hz。
- **period**, 表示测量周期, 测量值单位为 s。
- **pwidth**, 表示测量正脉冲宽度, 测量值单位为 s。
- **duty**, 表示测量占空比, 测量值单位为% (等于占空比乘以 100)。
- **pduty**, 与 **duty** 相同。
- **rovshoot**, 表示出测量上升沿后过冲, 测量值单位为% (等于过冲比例乘以 100)。
- **rpreshoot**, 表示测量上升沿前过冲, 测量值单位为% (等于过冲比例乘以 100)。
- **high**, 表示测量最大值, 测量值单位为 V。
- **low**, 表示测量最小值, 测量值单位为 V。
- **rrdly**, 表示测量上升沿到上升沿的延迟, 测量值单位为 s, 须指定参考通道。
- **rfdly**, 表示测量上升沿到下降沿的延迟, 测量值单位为 s, 须指定参考通道。
- **frdly**, 表示测量下降沿到上升沿的延迟, 测量值单位为 s, 须指定参考通道。
- **ffdly**, 表示测量下降沿到下降沿的延迟, 测量值单位为 s, 须指定参考通道。
- **phase**, 表示测量相位差, 测量值单位为° (等于弧度乘以 57.2958), 须指定参考通道。

1.6 测量通道设定

测量通道设定字符串格式：

<示波器通道>[,<外置通道>][[:<示波器参考通道>[,<外置参考通道>]]

例如：

- “1”，表示通过示波器通道 1 测量；
- “1,2” 表示通过示波器通道 1 外接的集线板通道 2 测量；
- “1:2” 表示通过示波器通道 1 测量，参考通道为示波器通道 2，此时如果进行相位测量，将测到通道 1 信号相对于通道 2 信号的相位。
- “1,2:2,4” 表示通过示波器通道 1 外接的集线板通道 2 测量，参考通道则为示波器通道 2 外接的集线板通道 4。
- “1,2:2” 表示通过示波器通道 1 外接的集线板通道 2 测量，参考通道则为示波器通道 2。
- “1:2,4” 表示通过示波器通道 1 测量，参考通道则为示波器通道 2 外接的集线板通道 4。

注意，参考通道中的示波器通道与测量通道中的示波器通道必须不相同。

1.7 分数计算公式

评分项中的分数算式和作品列表中的总分算式均可输入表达式。表达式为 c 语言风格表达式，程序在自动测试和评分过程中，会根据测量值计算项目得分，根据所有项目的得分和测量值计算总分。

1.7.1 支持的函数

函数均为小写。

表达式中支持的函数有：

- $\tanh(<x>)$ ，一元，双曲正切函数；
- $\tan(<x>)$ ，一元，正切函数；
- $\sqrt{<x>}$ ，一元，平方根函数；
- $\sinh(<x>)$ ，一元，双曲正弦函数；
- $\sin(<x>)$ ，一元，正弦函数；
- $\text{sign}(<x>)$ ，一元，符号函数，<x>为正值时返回 1，为负值时返回-1，为零时返回 0；
- $\text{round}(<x>)$ ，一元，四舍五入取整函数；

- `rem(<x>,<y>)`, 二元, 取余函数, 求 x/y 的正余数;
- `pow(<x>,<y>)`, 二元, 幂函数, 求 x^y ;
- `min(<x>,<y>)`, 二元, 返回 x 和 y 中较小的值;
- `max(<x>,<y>)`, 二元, 返回 x 和 y 中较大的值;
- `log10(<x>)`, 一元, 常用对数函数;
- `log(<x>,<y>)`, 二元, 对数函数, 求 $\log_y x$;
- `ln(<x>)`, 一元, 自然对数函数;
- `floor(<x>)`, 一元, 向下取整函数;
- `exp(<x>)`, 一元, 指数函数, 求 e^x ;
- `cosh(<x>)`, 一元, 正切余弦函数;
- `cos(<x>)`, 一元, 余弦函数;
- `ceil(<x>)`, 一元, 向上取整函数;
- `atan2(<x>,<y>)`, 二元, 返回 $\arg(x + jy)$;
- `atan(<x>)`, 一元, 反正切函数;
- `asin(<x>)`, 一元, 反正弦函数;
- `acos(<x>)`, 一元, 反余弦函数;
- `abs(<x>)`, 一元, 绝对值函数;
- `sat(<x>,<y>,<z>)`, 三元, 饱和函数, 将 x 饱和到 $[y, z]$ 区间, 如果 x 小于 y 返回 y , 如果 x 大于 z 返回 z , 否则返回 x 自身。
- `if(<x>,<y>,<z>)`, 三元, 条件函数, 如果 x 不等于 0, 返回 y , 否则返回 z 。

1.7.2 支持的运算符

表达式中支持的运算符有:

- 四则运算和求余: “+”、“-”、“*”、“/”、“%”。
- 关系运算符: “<”、“<=”、“>”、“>=”、“==”, 所有关系运算符, 如果为真, 返回值为 1, 否则返回值为 0。
- 按位操作运算符: “&” (与)、“|” (或)、“~” (非)、“^” (异或), 仅当操作数为整数时有效。

1.7.3 支持的变量

在评分项中的分数算式中，支持的变量仅有一个：“x”，即该项目的测量值。

在作品列表中的总分算式中，支持的变量数量为评分项目数量的两倍。分别为：

- $s_{<i>}$, $i = 1, 2, \dots, n$, n 为评分项数量，为第 n 个评分项的得分，如果由 10 个评分项，则它们分别是：“s1”、“s2”、“s3”、……、“s10”。
- $m_{<i>}$, $i = 1, 2, \dots, n$, n 为评分项数量，为第 n 个评分项的测量值，如果有 10 个评分项，则它们分别是：“m1”、“m2”、“m3”、……、“m10”。

1.7.4 分数计算公式示例

注意，以下例子中，公式中的空格都可以省略。

评分项中线性得分例子：

- 满分 5 分，1V 以上得满分，0.9V 以下得 0 分：
`sat((x - 0.9) / 0.1 * 5, 0, 5)`
- 满分 10 分，100mV 以下得满分，300mV 以上得 0 分：
`sat((0.3 - x) / 0.2 * 10, 0, 10)`
- 满分 5 分，与 1V 相差 0.1V 以内得满分，相差 0.3V 以上得 0 分：
`sat((0.3 - abs(x - 1)) / 0.2 * 5, 0, 5)`

评分项中分段得分例子：

- 频率小于 10MHz 得 0 分，频率大于等于 10MHz 得 1 分，大于等于 15MHz 得 2 分，大于等于 20MHz 得 3 分：
`if(x >= 20e6, 3, if(x >= 15e6, 2, if(x >= 10e6, 1, 0)))`
或者
`sat(floor((x - 5e6) / 5e6), 0, 3)`

总分算式中的例子：

- 总分等于所有 4 个项目得分之和：
`s1 + s2 + s3 + s4`
注意，不支持像 Excel 表格那样，使用类似“sum(s1:s4)”的表达式。
- 接上一个例子，但是如果第 3 项得分不为 0，第 4 项才得分：
`s1 + s2 + s3 + if(s3 > 0, s4, 0)`
或
`s1 + s2 + s3 + (s3 > 0) * s4`

- 如果第 3 项测量值与 10kHz 之差小于 1kHz，第 4 项才得分：

$$s1 + s2 + s3 + (\text{abs}(m3 - 10e3) < 1e3) * s4$$
- 如果第 3 项测量值与 10kHz 之差小于 1kHz，第 4 项加权 1，之差大于 3kHz，第 4 项不得分，其间线性加权：

$$s1 + s2 + s3 + \text{sat}((3e3 - \text{abs}(m3 - 10e3))/2e3, 0, 1) * s4$$

1.8 CSV 文件支持

在评分项较多，或者作品数量较多的情况下，推荐使用 Excel 或文本编辑器编辑好文件后导入到程序中使用。

1.8.1 通过 Excel 编辑文件

Excel 编辑文件时，需包含固定格式的列首。

对于评分项列表，列首如表 1 的第一行，表 1 也是一个简单的表格例子。

表 1 评分项目列表示例

测量项目描述	提示信息	测量通道	仪器设定	测量量	分数算式
方波信号的品质		1,1	DC, 1V/div, -2.5Vpos, 100us/div	Rect	$x * 2.5$
方波信号的幅值		1,1	2.5VtrigR	P2P	$\text{sat}((x - 4.5) * 2.5 / 0.5, 0, 2.5)$
方波信号的最小频率	调整信号频率到 19kHz 或更小。	1,1		Freq	$\text{sat}((19.5e3 - x) * 2.5 / 500, 0, 2.5)$
方波信号的最大频率	调整信号频率到 21kHz 或更大。	1,1		Freq	$\text{sat}((x - 20.5e3) * 2.5 / 500, 0, 2.5)$

对于作品列表，列首如表 2 的第一行，表 2 也是一个简单例子。

表 2 作品列表示例

作品编号	仪器 IP 地址	仪器端口	仪器 ID	分数算式	得分
U00000001	192.168.2.2	3000		$s1 + s2 + s3 + s4$	
U00000002	192.168.2.3	3000		$s1 + s2 + s3 + s4$	
U00000003	192.168.2.4	3000		$s1 + s2 + s3 + s4$	
U00000004	192.168.2.5	3000		$s1 + s2 + s3 + s4$	
U00000005	192.168.2.6	3000		$s1 + s2 + s3 + s4$	

注意：

1. 虽然上述表格中的单元格内出现了换行，实为本文档排版所需，实际表格中，每个单元格内都不能出现换行。
2. 保存时，务必选择文件类型为“CSV UTF-8（逗号分隔）(*.csv)”。

3. 如果不记得表格模式，可以使用程序导出功能，以导出的文件为模板进行修改。
4. 作品列表的导出/导入文件中可以包含仪器 ID 和得分，导入的仪器 ID 和得分在测试后都将被覆盖，导出的得分可用于使用 Excel 等应用进行后续统分工作。
5. 在导入操作之前，如果列表中已有内容，则内容的前 n 行都将被覆盖， n 为 CSV 文件中的行数。

1.8.2 通过文本编辑器编辑文件

CSV 文件本身是纯文本文件，它符合 RFC4180 规范，要点如下：

1. 单元格之间以英文半角逗号（“,”）分隔；
2. 分隔单元格的逗号前后不得有任何空格；
3. 行间以换行分隔，行换为 Windows 系统规格，包含回车符和换行符（“\r\n”），在 Windows 系统下的文本编辑器内，按回车新起一行即可，在 unix/linux/macos 系统中的，应将文本编辑器的换行模式切换为“Windows 习惯”；
4. 每行的最后一个单元个后，不用逗号；
5. 如果单元格内容包含有逗号，则用一对英文半角双引号将单元格内容包裹；
6. 如果单元格内容包含有英文半角双引号，则使用两个连续英文半角双引号代替。

表 3 是文本编辑器中的 CSV 文件示例，同样需要固定内容的行头。

表 3 文本编辑器中的 CSV 文件示例

测量项目描述,提示信息,测量通道,仪器设定,测量量,分数算式
方波信号的品质,,"1,1","DC, 1V/div, -2.5Vpos, 100us/div",Rect,x*2.5
方波信号的幅值,,"1,1",2.5VtrigR,P2P,"sat((x-4.5)*2.5/0.5, 0, 2.5)"
方波信号的最小频率,调整信号频率到 19kHz 或更小。,"1,1",,Freq,"sat((19.5e3-x)*2.5/500,0,2.5)"
方波信号的最大频率,调整信号频率到 21kHz 或更大。,"1,1",,Freq,"sat((x-20.5e3)*2.5/500,0,2.5)"

注意:

1. 保存时，务必选择编码（Encoding）为“UTF-8”。
2. 如果不记得表格模式，可以使用程序导出功能，以导出的文件为模板进行修改。
3. 作品列表的导出/导入文件中可以包含仪器 ID 和得分，导入的仪器 ID 和得分在测试后都将被覆盖，导出的得分可用于使用 Excel 等应用进行后续统分工作。
4. 在导入操作之前，如果列表中已有内容，则内容的前 n 行都将被覆盖，n 为 CSV 文件中的行数。

第2章 程序的再开发

虽然程序功能已相对完备，但目前仅测试了少量全国大学生电子设计竞赛综合测试题，若今后要推广使用，必然需要扩展功能，另外 SCPI 指令中还有很大一部分没有实现，在今后的使用中，不免会需要扩展这些指令以适应新的测试需求。

2.1 程序开发环境

- IDE: Visual Studio 2022 Community 版;
- 开发语言: C# 6.0
- 工程类型: .Net Window Form 窗体应用程序工程;
- SDK 依赖: .Net 6.0 SDK。具体依赖以下命名空间:
 - System.Windows, Windows 系统和窗体应用支持;
 - System.Windows.Forms, Windows 窗体控件支持;
 - System.Text, 文本和文本编码支持;
 - System.Text.RegularExpressions, 正则表达式支持;
 - System.Net, 网络支持;
 - System.Net.Socket, 以太网套接字支持;
 - System.Diagnostics, 调试功能支持;
 - System.Numerics, 数字和数学支持。

2.2 程序结构

程序由一个窗体类和多个功能类构成，包括：

- FormMain, 主窗体类，其中包括界面生成，和界面控件的用户事件响应代码，相关文件：FormMain.cs、FormMain.Designer.cs()。
- GwScpi, 针对固伟多功能示波器 MDO-2000ES 系列，实现 SCPI 指令接口，并扩展多种复杂测量功能的类，相关文件：GwScpi.cs;
虽然主要针对固伟 MDO-2000ES 系列多功能示波器，但因 SCPI 指令属于测试测量工业标准，所以，理应支持所有符合 SCPI 指令规范的多功能示波器。
- GwScpiEth: 以太网接口的 GwScpi 类封装，相关文件：GwScpi.cs;
- FFT.cs: 离散傅里叶变换、窗函数和功率密度谱分析，相关文件：FFT.cs。

- ExprAnalyse: 表达式解析, 相关文件: ExprAnalyse.cs。
- EngineeringFormatExt, 对 string 和 double 类型进行工程计数表达扩展, 使得它们支持带有国际单位制词头的数值表达, 相关文件: EngineeringFomatExt.cs

2.2.1 FormMain 类字段

FormMain 类包含以下字段:

- private Socket? soc
用于测试网络连接的套接字类实例。
- private System.ComponentModel.IContainer components
程序窗体中的控件集合。
- private DataGridView dgvScoreDetail
评分项列表视图。
- private NumericUpDown numScoreItemQuantity
评分项数量数值框。
- private GroupBox grpScoreSetting
容纳评分项数值框、“导出到文件”按钮、“从文件导入”按钮、“测试选中的评分项”按钮和评分项列表视图的分组框。
- private Label label1
“评分项数量:” 标签。
- private GroupBox grpEntries
容纳作品数量数值框、“导出到文件”按钮、“从文件导入”按钮、“复制公式到所有行”按钮、“测试仪器连接”按钮、“对选中的作品评分”按钮和作品列表视图的分组框。
- private Label label2
“作品数量:” 标签。
- private DataGridView dgvEntries
作品列表视图。
- private NumericUpDown numEntriesQuantity
作品数量数值框。
- private SplitContainer splitContainer1
用于分隔评分细则部分和待测作品部分, 实现拖拽分配两者空间分配的分隔容器。

- `private Button btnTestIns`
“测试仪器连接”按钮。
- `private Button btnTestMeasure`
“测试选中的评分项”按钮。
- `private Button btnEvalSelected`
“对选中的作品评分”按钮
- `private Button btnExportScoreDetail`
评分项中的“导出为文件”按钮。
- `private Button btnImportScoreDetail`
评分项中的“从文件导入”按钮。
- `private Button btnImportEntries`
作品列表中的“导出为文件”按钮。
- `private Button btnExportEntries`
作品列表中的“从文件导入”按钮。
- `private Button btnCopyExpr`
作品列表中的“复制公式到所有行”按钮。

2.2.2 FormMain 类方法

FormMain 类包含以下方法：

- `public FormMain()`
构造函数，用于初始化窗体类。
- `private void init_DgvScoreDetail()`
该方法用于初始化评分项列表视图，填入列首，根据 `numScoreItemQuantity.Value` 设定行数，并设定了一些示例。
- `private void init_GdvEntries()`
该方法用于初始化作品列表视图，填入列首，根据 `numEntriesQuantity.Value` 设定行数，并设定了一些示例。
- `private void gwScpi_Test()`
该方法用于在开发期间测试 `gwScpi` 类，已废弃。
- `private void FormMain_Load(object sender, EventArgs e)`
主窗体加载事件的处理方法，调用了 `init_DgvScoreDetail()` 和 `init_GdvEntries()`。

- `private void numScoreItemQuantity_ValueChanged(object sender, EventArgs e)`
`numScoreItemQuantity` 的数值变化事件的处理方法，将重新设定 `dgvScoreDetail` 的行数和序号。
- `private void numEntriesQuantity_ValueChanged(object sender, EventArgs e)`
`numEntriesQuantity` 的数值变化事件的处理方法，将重新设定 `dgvEntries` 的行数和序号。
- `private GwScpi? GetGwScpiFromIpAndEp(string ip, string port)`
 通过仪器的 IP 地址和端口号构造 `GwScpi` 实例的函数。
 参数：
 - `string ip`, ip 地址字符串，如 “192.168.2.2”；
 - `string port`, 端口字符串，如 “3002”；
 返回值：
 - `GwScpi?` 类型，如果构造成功，将返回一个 `GwScpi` 类实例，如果构造不成功，将返回 `null`，调用者可通过它是否为 `null` 来判断是否构造成功。IP 地址和端口无对应仪器，或网络连接不畅将导致构造不成功。
 异常：如果 `ip` 和 `port` 不是合乎格式的字符串，该方法将抛出 `FormatException` 或 `ArgumentException`。
- `private void btnTestIns_Click(object sender, EventArgs e)`
 处理 `btnTestIns` 的鼠标单击事件的方法。将根据作品列表中的选定行中的 IP 和端口信息尝试构造一个 `GwScpiEth` 类实例，并通过 SCPI 指令 “*idn?” 获取仪器 ID，填入作品列表中的仪器 ID 一列中。如果存在连接问题或 IP、端口格式问题，将填写 “连接失败” 到仪器 ID 列中。
- `private GwScpiEth? TryOpenInsFromEntryRow(DataGridViewRow row)`
 该方法根据作品列表中的行信息，尝试构造 `GwScpiEth` 类实例。
 参数：
 - `DataGridViewRow row`，必须是 `dgvEntries` 中的一行。
 返回值：

- GwScpiEth? 类型，如果构造成功，将返回一个 GwScpiEth 类实例，如果构造不成功，将返回 null，调用者可通过它是否为 null 来判断是否构造成功。row 中的 IP 地址和端口无对应仪器，或 IP 地址、端口格式不正确，或网络连接不畅将导致构造不成功。
- private void getIchAndEch(string str, out int ich, out int ech)

该方法解析测量通道（dgvScoreDetail 中的“测量通道”列内容中以冒号“:”分隔的两部分内容之一）字符串，获取仪器通道和外部通道编号。

参数：

 - string str，测量通道字符串，如“1”、“1,2”等。

返回值：

 - out int ich，如果 str 格式正确将返回从 1 开始的仪器通道编号，如果 str 格式有误，则返回 0。
 - out int ech，如果 str 格式正确将返回从 1 开始的外部通道编号，如果 str 格式有误，则返回 0。

异常：无
- private void DoMeasureRow(GwScpiEth gw, DataGridViewRow row)

该方法完成使用 gw 连接的仪器，对评分项目列表中的一行（row）的自动测量和评分。如果有提示信息，它将弹出提示对话框，等待确认和继续测试。如果测量通道、测量量或得分表达式填写不正确，它将给出提示并立刻返回，同时将得分列清空。

参数：

 - GwScpiEth gw，已连接到有效仪器的 GwScpiEth 类实例。
 - DataGridViewRow row，得分项目列表 dgvScoreDetail 中的一行。

返回值：无

异常：无
- private void btnTestMeasure_Click(object sender, EventArgs e)

btnTestMeasure 的鼠标单击事件处理方法，将检查 dgvEntries 和 dgvScoreDetail 中是否有选中的行，如无给出提示并返回，如有，使用 dgvEntries 中选中行指定的仪器，完成 dgvScoreDetail 中选中行指定的仪器设定、测量和评分。
- private void btnEvalSelected_Click(object sender, EventArgs e)

btnEvalSelected 的鼠标单击事件的处理方法，将对 dgvEntries 中所有选中的行逐个依据 dgvScoreDetail 中的所有评分项完成测量、各项评分和总分计算。如果 dgvEntries 中没有选中的行，则给出提示后退出。

- private void btnExportScoreDetail_Click(object sender, EventArgs e)
btnExportScoreDetail（评分项的“导出为文件”按钮）的鼠标单击事件的处理方法，将弹出保存文件对话框，用户选择好文件存储目录并填写文件名后，dgvScoreDetail（评分项列表）中的内容的将被导出到 UTF-8 编码的 CSV 文件。
- private void btnImportScoreDetail_Click(object sender, EventArgs e)
btnImportScoreDetail（评分项的“从文件导入”按钮）的鼠标单击事件的处理方法，将弹出打开文件对话框，用户选择好文件后，将尝试从被选择的 UTF-8 编码的 CSV 文件导入 dgvScoreDetail（评分项列表）内容。
- private void btnExportEntries_Click(object sender, EventArgs e)
btnExportEntries（作品列表的“导出为文件”按钮）的鼠标单击事件的处理方法，将弹出保存文件对话框，用户选择好文件存储目录并填写文件名后，dgvEntries（作品列表）中的内容的将被导出到 UTF-8 编码的 CSV 文件。
- private void btnImportEntries_Click(object sender, EventArgs e)
btnImportEntries（作品列表的“从文件导入”按钮）的鼠标单击事件的处理方法，将弹出打开文件对话框，用户选择好文件后，将尝试从被选择的 UTF-8 编码的 CSV 文件导入 dgvEntries（作品列表）内容。
- private void btnCopyExpr_Click(object sender, EventArgs e)
btnCopyExpr（作品列表的“复制公式到所有行”）的鼠标单击事件的处理方法，将判断是否有选中的“总分算式”单元格，如有，将其中内容复制到所有行的“总分算式”单元格。

2.2.3 GwScpi 类依赖

GwScpi 类中使用了方法定义 GwScpiRwFun，其定义为：

```
using GwScpiRwFun = Func<byte[], int, int, int>
```

它是与仪器 SCPI 接口通信的字符流读写函数的类型。字符流读写函数包含三个形参：

- 第 1 个参数，byte[] 类型，ASCII 编码的字符串缓冲区，对于写函数它存储待发出的字符串，对于读函数，接收到的字符串将被填充到其中。

- 第 2 个参数，int 类型，缓冲区在 byte[] 中的起始位置。
 - 第 3 个参数，int 类型，缓冲区的长度，对于写函数，它是待发出的字符数量，对于读函数，它是欲接收的字符数量。
- 它有一个返回值，int 类型，为实际写出/读入的字符数量。

2.2.4 GwScpi 类字段

GwScpi 类包含以下字段：

- private GwScpiRwFun wfun
GwScpiRwFun 类型的方法（代理），在构造函数中赋值，为与仪器 SCPI 接口通信的字符流的写函数。
- private GwScpiRwFun rfun
GwScpiRwFun 类型的方法（代理），在构造函数中赋值，为与仪器 SCPI 接口通信的字符流的写函数。
- private byte[] rxbuf
接受数据和处理数据的缓冲区，将在构造函数中实例化，默认大小为 64KiByte。
- public int AcqMemCh = 1
获取仪器原始数据时，使用的通道。须在调用 int AcquireMemory(hort[] rawData)方法前使用。

2.2.5 GwScpi 类方法

GwScpi 类包含以下方法：

- private void write(byte[] data, int len)
封装了 wfun，该函数会循环调用 wfun，直到写出由 len 指定的所需的数据量。
参数：
 - byte[] data，待写出的数据。
 - int len，待写出的数据长度，务必小于 data.Length。
 返回值：无
- private void read(byte[] data, int len)
封装了 rfun，该函数会循环调用 rfun，直到收到由 len 指定的所需的数据量。

参数:

- byte[] data, 存储待读入的数据。
- int len, 待读入的数据长度, 务必小于 data.Length。

返回值: 无

- public GwScpi(GwScpiRwFun write, GwScpiRwFun read, int rxBufSize = 65536)

构造方法。

参数:

- GwScpiRwFun write, 与仪器 SCPI 接口通信的字符流的写函数。
- GwScpiRwFun read, 与仪器 SCPI 接口通信的字符流的读函数。
- int rxBufSize = 65536, 接收数据和处理数据的缓冲区大小, 单位为 Byte, 默认值为 65536Byte (64KiB)。

- public string Command(string cmd, bool hasRtn = true)

符合 SCPI 规范的命令发送方法。

参数:

- string cmd, SCPI 命令字符串, 不包含换行符, 换行符将在方法内添加, 该方法不会检查 cmd 的合法性, 如果 cmd 不合法, 依据仪器对 SCPI 的具体实现, 可能返回 “?”, 可能不返回任何信息。
- bool hasRtn = true, 指定 cmd 指定的命令是否由返回信息。

返回值: 仪器返回的信息字符串, 如果 hasRtn 为 false, 则返回空字符串 (“”)。

- public int AcquireMemory(short[] rawData)

获取仪器采集到的原始数据。该方法会使用 “:ACQUIRE<ch>:MEMORY?” 指令, 获取由 AcqMemCh 字段指定的通道的原始数据, 该方法首先获取数据头, 然后解析数据头获得数据量, 再获取原始数据头, 最后做大小端转换和赋值到 rawData 中。

参数:

- short[] rawData, 用于存储获取到的原始数据。

返回值: 原始数据的长度。

- public void MeasureSource(int ch)

指定测量所用的通道。

参数:

- int ch, 从 1 开始的通道编号。

- public void MeasureSource(int ch1, int ch2)

指定测量所用的通道和参考通道。

参数：

- `int ch1`，从 1 开始的测量通道编号。
- `int ch2`，从 1 开始的参考通道编号。
- `public double MeasureAmplitude()`
通过发送 “:MEASure:AMPlitude?” 命令测量幅值。
返回值：以 V 为单位的幅值。
- `public double MeasurePeak2Peak()`
通过发送 “:MEASure:PK2Pk?” 命令测量峰峰值。
返回值：以 V 为单位的峰峰值。
- `public double MeasureRms()`
通过发送 “:MEASure:RMS?” 命令测量有效值。
返回值：以 V 为单位的有效值。
- `public double MeasurePeriod()`
通过发送 “:MEASure:PERiod?” 命令测量周期。
返回值：以 s 为单位的周期。
- `public double MeasureRise()`
通过发送 “:MEASure:RISe?” 命令测量上升时间。
返回值：以 s 为单位的上升时间。
- `public double MeasureFall()`
通过发送 “:MEASure:FALL?” 命令测量下降时间。
返回值：以 s 为单位的下降时间。
- `public void VerticalSetting(int ch, bool ac, double pos, double scale)`
对仪器的指定通道进行垂直设定。
参数：
 - `int ch`，从 1 开始的测量通道编号。
 - `bool ac`，`true`：设置为交流耦合；`false`：设置为直流耦合。
 - `double pos`，以 V 为单位的垂直位置。
 - `double scale`，以 V/div 为单位的垂直档位。
- `public void ParseAndSetFromString(int ch, string str)`
解析设定字符串，并对指定通道进行相应设置。
参数：
 - `int ch`：从 1 开始的测量通道编号。
 - `string str`：设定字符串，参考 1.6 节。

- `public double Measure(string str)`
测量由 `str` 指定的项目。
参数：
■ `string str`: 指定测量项目，参考 1.5 节。
返回值：测量值，具体单位，参考 1.5 节。
- `private void calcHarmPowers(double[] psd, double fbase, out double pBase, out double pHarm, out double pNoi)`
根据功率密度谱 `psd` 和基频，计算基频功率、谐波功率和噪声功率。
参数：
■ `double[] psd`: 功率密度谱，长度须为 8192。
■ `double fbase`: 基频频率。
返回值：
■ `out double pBase`: 基频功率。
■ `out double pHarm`: 谐波功率。
■ `out double pNoi`: 噪声功率。
- `private void acqMemAndCalcHarmPowers(out double pBase, out double pHarm, out double pNoi)`
使用 `int AcquireMemory(short[] rawData)` 方法获取原始数据，找出基频，并使用 `calcHarmPowers(double[] psd, double fbase, out double pBase, out double pHarm, out double pNoi)` 方法计算基频功率、谐波功率和噪声功率。
■ 参数：无。
返回值：
■ `out double pBase`: 基频功率。
■ `out double pHarm`: 谐波功率。
■ `out double pNoi`: 噪声功率。
- `private double Distortion()`
获取原始数据，并计算其失真度。
返回值：失真度，单位为 1（不是%）。
- `private double EvalRect()`
获取原始数据，通过分析其失真度并测量其上升、下降沿时间和周期，来评估如果其为方波，质量如何。
返回值：0~1 之间的质量指数，0 表示质量很差，1 表示质量很好。

2.2.6 FFT 类

<与再开发关系不大，暂略>

2.2.7 ExprAnalyse 类

<与再开发关系不大，暂略>

2.2.8 EngineerFormatExt 方法扩展

<与再开发关系不大，暂略>

2.3 再开发关键点

2.3.1 测量通道设定拓展

测量通道设定拓展可通过扩充 GwScpi 类中的 public void ParseAndSetFromString(int ch, string str)方法内容来达成。

通道设定的基本格式有以下几种：

- 无参数设定，如“AC”、“DC”
- 前置参数设定，如“1V/div”、“0VtrigR”、“64Avg”
- 后置参数设定，如“TrigMode:AUT”

对于无参数设定，可参考 ParseAndSetFromString 方法中的“AC”设定项的写法，在 if...else if...中添加类似代码 2 的代码段。

代码 2 添加“无参数设定”解析代码的示例

```
else if (word.Contains("AC", StringComparison.OrdinalIgnoreCase))
{
    this.Command(@":CHANNEL" + ch.ToString() + ":COUPLing AC", false);
}
```

对于前置参数设定，可参考 ParseAndSetFromString 方法中的“V/div”设定项的写法，在 if...else if...中添加类似代码 3 的代码段。

代码 3 添加“前置参数设定”解析代码的示例

```
else if ((k = word.IndexOf("V/div", StringComparison.OrdinalIgnoreCase)) > 0)
{
    string val = word.Substring(0, k);
    double d = val.ToDoubleFromEngineerFormat();
    d = Math.Round(d, 6);
    this.Command(@":CHANNEL" + ch.ToString() + ":SCALE " + d.ToString(), false);
}
```

对于后置参数，可参考 ParseAndSetFromString 方法中的 “TrigSrc:” 设定项的写法，在 if...else if...中添加类似代码 4 的代码段。

代码 4 添加“后置参数设定”解析代码的示例

```
else if ((k = word.IndexOf("TrigSrc:", StringComparison.OrdinalIgnoreCase)) > 0)
{
    string val = word.Substring(k + 8);
    this.Command("@":TRIGger:SOURce " + val.ToUpper(), false);
}
```

2.3.2 测量量拓展

测量量拓展可通过扩充 GwScpi 类中的 public double Measure(int ch, string str)方法内容来达成。

在 switch (str.ToLower())中增加新的 case 即可。

对于 SCPI 规范包含的测量指令，可采用类似 amp 的写法，如代码 5 所示。

代码 5 添加新的 SCPI 测量命令支持的代码段示例

```
case "amp":
    val = Command("@":MEASure:AMPlitude?");
    break;
```

对于 SCPI 规范不包含的测量指令，则须另行增加多个测量计算过程，增加额外的方法来完成，例如评估方波质量的 EvalRect()方法，如代码 6 所示。

代码 6 添加无 SCPI 测量命令支持的新测量量的代码段示例

```
case "rect":
    return EvalRect();
```

2.3.3 分数计算公式拓展

<略>

2.4 关键代码

2.4.1 GwScpi.Command

2.4.2 GwScpi.Command

代码 7 GwScpi.Command() 方法

```
public string Command(string cmd, bool hasRtn = true)
{
    byte[] txdata = Encoding.ASCII.GetBytes(cmd + "\n");
    write(txdata, txdata.Length);
    if (hasRtn)
    {
        int len = rfun(rxbuf, 0, 8192);
        string str = Encoding.ASCII.GetString(rxbuf, 0, len);
        if (str[str.Length - 1] == '\n')
            return str.Substring(0, str.Length - 1);
        else
            return str;
    }
    else
    {
        return "";
    }
}
```

2.4.3 GwScpi.AcquireMemory

代码 8 GwScpi.AcquireMemory() 方法

```
public int AcquireMemory(short[] rawData)
{
    string rtn = Command("@":ACquire"+AcqMemCh.ToString()+"@":MEMORY?");
    int sharp = rtn.IndexOf('#');
    int l = int.Parse(rtn.Substring(sharp + 1, 1));
    l = int.Parse(rtn.Substring(sharp + 2, 1));
    this.read(rxbuf, l + 1); // include '\n'
    //soc.Receive(rxbuf, 1, SocketFlags.None);
    for (int i = 0; i < l; i += 2)
    {
        Array.Reverse(rxbuf, i, 2);
    }
    Buffer.BlockCopy(rxbuf, 0, rawData, 0, l);
    //int l2 = rfun(rxbuf, 0, 8192);
    return l / 2;
}
```

2.4.4 GwScpi.ParseAndSetFromString

代码 9 GwScpi.ParseAndSetFromString() 方法

```
public void ParseAndSetFromString(int ch, string str)
{
    if (str == null)
        return;
    int len = str.Length;
    for (int i = 0, j = 0; i < len && j >= 0; i = j + 1)
    {
        int k;
        j = str.IndexOf(",", i);
        string word;
        if (j > 0)
        {
            word = str.Substring(i, j - i);
        }
    }
}
```

```

    }
    else
    {
        word = str.Substring(i, len - i);
    }
    if (word.Contains("DC", StringComparison.OrdinalIgnoreCase))
    {
        this.Command(@":CHANnel" + ch.ToString() + ":COUPling DC",
false);
    }
    else if (word.Contains("AC", StringComparison.OrdinalIgnoreCase))
    {
        this.Command(@":CHANnel" + ch.ToString() + ":COUPling AC",
false);
    }
    else if ((k = word.IndexOf("V/div",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        double d = val.ToDoubleFromEngineerFormat();
        d = Math.Round(d, 6);
        this.Command(@":CHANnel" + ch.ToString() + ":SCAle " +
d.ToString(), false);
    }
    else if ((k = word.IndexOf("Vpos",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        double d = val.ToDoubleFromEngineerFormat();
        d = Math.Round(d, 6);
        this.Command(@":CHANnel" + ch.ToString() + ":POSition " +
d.ToString(), false);
    }
    else if ((k = word.IndexOf("s/div",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        double d = val.ToDoubleFromEngineerFormat();
        d = Math.Round(d, 10);
        this.Command(@":TIMEbase:SCAle " + d.ToString(), false);
    }
    else if ((k = word.IndexOf("spos",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        double d = val.ToDoubleFromEngineerFormat();
        d = Math.Round(d, 10);
        this.Command(@":TIMEbase:POSition " + d.ToString(), false);
    }
    else if ((k = word.IndexOf("Vtrigr",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        this.Command(@":TRIGger:TYPe EDGE", false);
        this.Command(@":TRIGger:EDGE:SLOP RISE", false);
        this.Command(@":TRIGger:LEVel " +
val.ToDoubleFromEngineerFormat(), false);
    }
    else if ((k = word.IndexOf("Vtrigf",
StringComparison.OrdinalIgnoreCase)) > 0)

```

```

    {
        string val = word.Substring(0, k);
        this.Command(@":TRIGger:TYPE EDGE", false);
        this.Command(@":TRIGger:EDGE:SLOP FALL", false);
        this.Command(@":TRIGger:LEVel " +
val.ToDoubleFromEngineerFormat(), false);
    }
    else if ((k = word.IndexOf("TrigSrc:",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(k + 8);
        this.Command(@":TRIGger:SOURce " + val.ToUpper(), false);
    }
    else if ((k = word.IndexOf("TrigCoup:",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(k + 9);
        this.Command(@":TRIGger:COUPle " + val.ToUpper(), false);
    }
    else if ((k = word.IndexOf("TrigMode:",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(k + 9);
        this.Command(@":TRIG:MOD " + val.ToUpper(), false);
    }
    else if ((k = word.IndexOf("Avg",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        int d = int.Parse(val);
        if (d <= 1)
            this.Command(@":ACQuire:MODe SAMPlE");
        else
        {
            this.Command(@":ACQuire:MODe AVERAge");
            this.Command(@":ACQuire:AVERAge " + d.ToString(), false);
        }
    }
    else if ((k = word.IndexOf("Smps",
StringComparison.OrdinalIgnoreCase)) > 0)
    {
        string val = word.Substring(0, k);
        double d = val.ToDoubleFromEngineerFormat();
        int d2 = (int)Math.Round(d);
        this.Command(@":ACQuire:RECOrdlength " + d.ToString());
    }
    }
}

```

2.4.5 GwScpi.Measure

代码 10 GwScpi.Measure() 方法

```

public double Measure(string str)
{
    string val;
    switch (str.ToLower())
    {

```

```

case "rect":
    return EvalRect();
case "dist":
    return Distortion();
case "tfreq":
    val = Command("@:TRIGger:FREQuency?");
    break;
case "mean":
    val = Command("@:MEASure:MEAN?");
    break;
case "cmean":
    val = Command("@:MEASure:CMEan?");
    break;
case "amp":
    val = Command("@:MEASure:AMPlitude?");
    break;
case "rms":
    val = Command("@:MEASure:RMS?");
    break;
case "crms":
    val = Command("@:MEASure:CRMS?");
    break;
case "p2p":
    val = Command("@:MEASure:PK2Pk?");
    break;
case "freq":
    val = Command("@:MEASure:FREQuency?");
    break;
case "period":
    val = Command("@:MEASure:PERiod?");
    break;
case "pwidth":
    val = Command("@:MEASure:PWIDth?");
    break;
case "duty":
case "pduty":
    val = Command("@:MEASure:PDUTy?");
    break;
case "rovshoot":
    val = Command("@:MEASure:ROVShoot?");
    break;
case "rpreshoot":
    val = Command("@:MEASure:RPReshoot?");
    break;
case "high":
    val = Command("@:MEASure:HIGH?");
    break;
case "low":
    val = Command("@:MEASure:LOW?");
    break;
case "rrdly":
    val = Command("@:MEASure:FRRDeLay?");
    break;
case "rfdly":
    val = Command("@:MEASure:FRFDeLay?");
    break;
case "frdly":
    val = Command("@:MEASure:FFRDeLay?");
    break;
case "ffdly":

```

```

        val = Command("@":MEASure:FFFDeLay?");
        break;
    case "phase":
        val = Command("@":MEASure:PHase?");
        break;
    default: return double.NaN;
}
double rtn;
if(double.TryParse(val, out rtn))
    return rtn;
else
    return double.NaN;
}

```

2.4.6 FormMain.DoMeasureRow

代码 11 FormMain.DoMeasureRow() 方法

```

private void DoMeasureRow(GwScpiEth gw, DataGridViewRow row)
{
    string? promt = (string)row.Cells[2].Value;
    if (promt != null && promt.Length > 0)
    {
        MessageBox.Show((string)promt, "请按提示操作，完成后按“OK”",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    string ch = (string)row.Cells[3].Value;
    if (ch == null)
    {
        MessageBox.Show("测量通道填写不正确。", "警告", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        row.Cells[8].Value = string.Empty;
        return;
    }
    int colon = ch.IndexOf(':');
    string? chstr, rchstr;
    if (colon > 0)
    {
        chstr = ch.Substring(0, colon);
        rchstr = ch.Substring(colon + 1);
    }
    else
    {
        chstr = ch;
        rchstr = String.Empty;
    }
    int ich, ech, rich, rech;
    try
    {
        getIchAndEch(chstr, out ich, out ech);
        getIchAndEch(rchstr, out rich, out rech);
    }
    catch
    {
        MessageBox.Show("测量通道填写不正确。", "警告", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        row.Cells[8].Value = string.Empty;
    }
}

```

```

        return;
    }
    gw.Scpi.ParseAndSetFromString(ich, (string)row.Cells[4].Value);
    if (rich > 0)
        gw.Scpi.MeasureSource(ich, rich);
    else
        gw.Scpi.MeasureSource(ich);
    gw.Scpi.AcqMemCh = ich;
    Thread.Sleep(250);
    if (row.Cells[5].Value != null)
    {
        double val = gw.Scpi.Measure((string)row.Cells[5].Value);
        if (!double.IsNaN(val))
        {
            row.Cells[7].Value = val.ToEngineerFormatString(6);
            try
            {
                ExprAnalyse.Variable vars = new("x", val);
                ExprAnalyse ea = new
ExprAnalyse((string)row.Cells[6].Value, new ExprAnalyse.Variable[1]
{ vars });
                row.Cells[8].Value = ea.Value.ToString();
            }
            catch
            {
                MessageBox.Show("得分表达式填写不正确。", "警告",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                row.Cells[8].Value = string.Empty;
            }
        }
        else
        {
            MessageBox.Show("测量量填写不正确，或没有测到有效的值。", "警告",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            row.Cells[8].Value = string.Empty;
        }
    }
    else
    {
        MessageBox.Show("测量量填写不正确。", "警告", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        row.Cells[8].Value = string.Empty;
    }
}
}

```

第3章 接口拓展板

在待测端口较多，示波器本身通道数不够的情况下，虽然可通过自动测试软件中评分项中的提示信息提示学生连接探头到不同被测端口，但这样增加了测试复杂性，降低了测试效率。

如果使用外接集线板，则可通过分时复用的方式通道拓展示波器的测量通道，并由自动测试软件程控切换。

注意：对于需要参考通道的测量，被测通道和参考通道不能通过外接集线板连接到示波器的同一个通道，因为它们往往需要同时接入示波器。

已研制的外接集线板可将双通道示波器的两个通道各扩展为 6 个通道，两个通道中的任何一个都可以选择 6 个通道中的任何一个。此外，它还包含三个通道的电流测量能力。

3.1 硬件电路

外接集线板的硬件电路由以下几部分组成：

1. MCU 部分，主要为一片 ARM Cortex-M3 核心的 MCU。
2. 电源电流测量部分，主要由 TI 公司型号为 INA199A2 的电流测量仪表放大器 IC 构成。
3. 信号选择部分，主要由光耦继电器 ACQ210 构成。

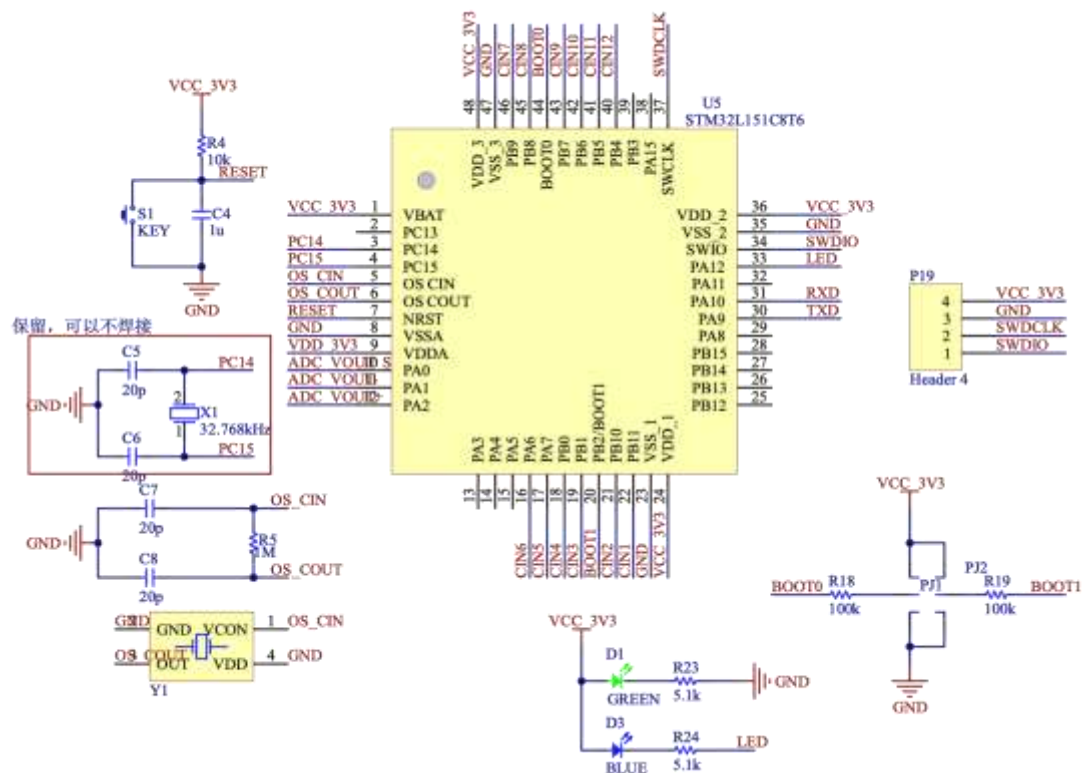
3.1.1 MCU 部分

这部分电路如图 7 所示。其中，MCU 型号为 STM32L151C8T6，由外部 32.768kHz 无源晶振配合内部震荡电路为其提供工作时钟源，期内有锁相环，实际内核工作频率为 32MHz。MCU 的复位部分使用简单的 RC 网络。

MCU 的大部分 IO 用来控制光耦继电器的通断。电源电流测量部分 INA199 输出的电压信号进入 MCU 的 ADC 输入通道。

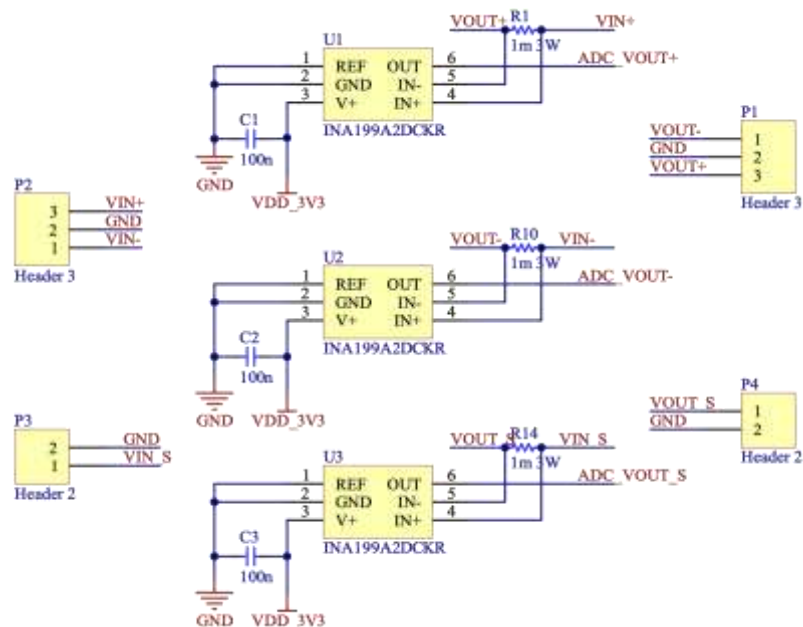
与计算机的连接则采用 UART 接口，由一片 USB-UART 桥接芯片完成协议转换。

调试接口则采用两线制的 SWD 接口。



3.1.2 电源电流测量部分

这部分电路如图 8 所示。



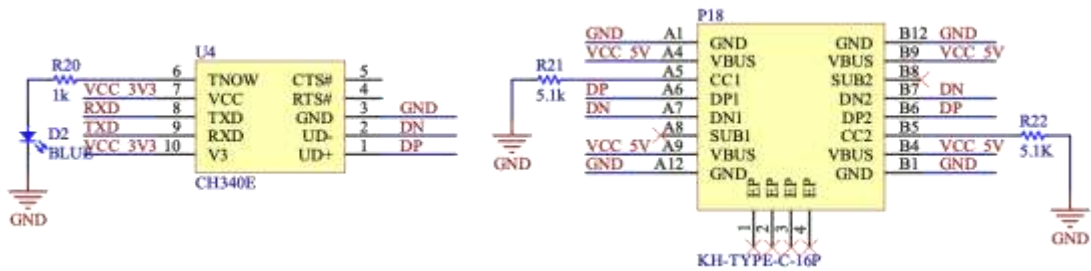


图 10 USB-UART 接口部分电路

3.1.5 电源部分

电源部分采用一片型号为 AMS1117-3.3 的低压差线性电压调整 IC，如图 11 所示。电路简单，不赘述。

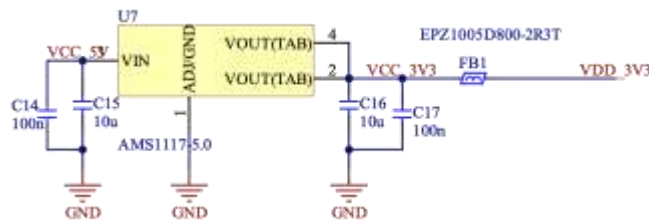


图 11 MCU 部分电路

3.2 与计算机的接口

3.2.1 接口和驱动

为与计算机连接，需安装 CH340 的设备驱动程序。通过 USB Type-C 电缆与计算机连接后，接口拓展板将表现为 USB 总线上的 UART 控制器。

在 MCU 中，实现了通过 UART 收发字符流的控制命令，命令完全基于可读的 ACSII 字符。

对于 Windows 系统，可使用 putty 等串行终端进行控制，C#和.Net 程序中可使用.Net 提供的 SerialPort 类库对其进行字符流读写操作，可用来对其发送命令和获取信息。

对于 Linux/MacOS 系统，系统会将其映射为字符文件，可使用 putty、minicom 等字符终端进行控制，程序中可使用 IO 文件读写 API 对其进行字符流读写操作，以对其发送命令和获取信息。

3.2.2 命令格式

所有命令和返回信息以回车换行符（“\r\n”）结束。

通道选择命令

格式：C<输出通道><输入通道>

其中，输出通道为一位数字，1 或 2；输入通道为一位数字，1、2、3、……、6。意为将指定的输出通道连接到指定的输入通道。如果该输入通道之前已连接至其它输入通道，则之前的连接将断开。

例如：发送“C25\r\n”，将会连接输入通道 5 到输出通道 2。

返回信息：“OK\r\n”

电流采集命令

格式：A<电流测量通道>

其中，电流测量通道为一位数字，1、2 或 3，分别对应双电源的正、负电源和单电源。

例如：发送“A2\r\n”，将使得接口拓展板采集负电源电流，并返回电流信息；

返回信息：“<电流值>\r\n”

其中，电流值为浮点或科学计数法表达的以 A 为单位的瞬时电流值。

例如，返回信息“0.123\r\n”，表示电流值为 0.123A。

3.3 MCU 程序设计

<略>