

浙江大学



本科实验报告

虚拟仪器技术

课程名称： 仪器系统设计

姓名：

学院： 生物医学工程与仪器科学学院

专业： 生物医学工程

学号：

指导老师： 周泓

2025 年 6 月 7 日

浙江大学实验报告

专业： 生物医学工程
姓名： _____
学号： _____
日期： 2025 年 6 月 7 日
地点： _____

课程名称： 仪器系统设计 指导老师： 周泓 成绩： _____
实验名称： 虚拟仪器技术 实验类型： _____ 同组学生姓名： _____

1. 描述基于虚拟仪器技术的虚拟示波器的功能描述和组成。

虚拟仪器技术概述：

虚拟仪器就是在通用的计算机平台上定义和设计仪器的功能，用户操作计算机的同时就是在使用一台专门的电子仪器。虚拟仪器以计算机为核心，充分利用计算机强大的图形界面和数据处理能力，提供对测量数据的分析和显示功能。与传统仪器相比，虚拟仪器具有仪器功能可由用户自己定义、速度和可靠性高、数据可以进行编辑存储和打印、基于软件的结构体系可大大节省开发费用等显著优点。



图 1: 虚拟仪器软件系统架构

功能描述：

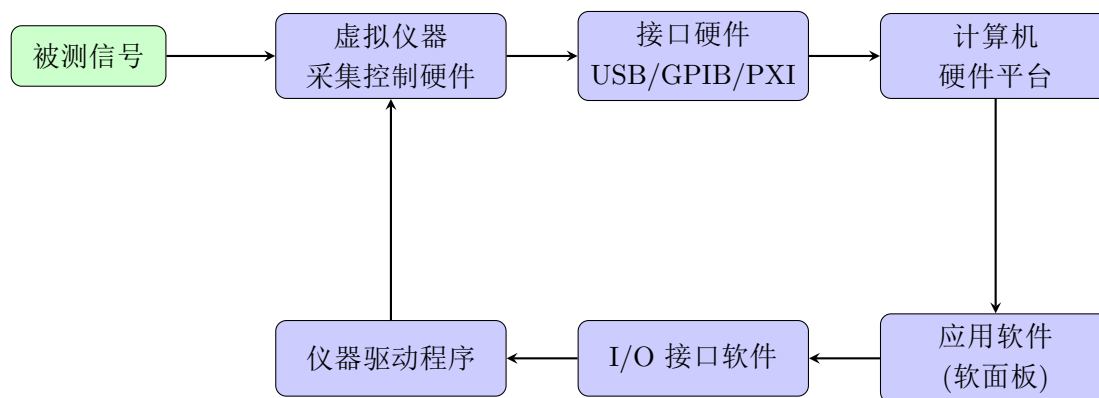
基于虚拟仪器技术的虚拟示波器，其核心是将传统示波器的信号分析和处理、结果的表达与输出功能转移给计算机来完成，利用计算机的高速计算能力和强大的显示能力更好地完成原有功能。虚拟示波器具备传统示波器的所有基本功能，同时具有更强的数据处理和分析能力。

具体功能包括：信号采集与显示功能可以实时采集模拟信号并将其数字化，支持最多 4 路 A/D 采样，分辨率达 12 位，采样信号频率最高可达 50KHz，输入范围为 0~3.3V，提供单通道单点、单通道连续与多通道连续采样模式。波形测量功能提供各种自动测量参数，如峰峰值、均方根值、频率、周期、占空比、上升时间、下降时间等。波形分析功能具备强大的信号处理能力，包括频谱分析（FFT）、数字滤波、数学运算等。触发功能提供多种触发模式确保波形稳定显示。数据存储与处理功能可将采集到的数据实时、直接地通过计算机总线传输到计算机内存或硬盘，存储为数据文件或数据库供后续分析使用。

组成：

虚拟示波器的结构包括虚拟仪器的硬件系统与软件系统两个部分。

硬件系统



软件系统

图 2: 虚拟示波器系统组成框图

硬件系统主要负责信号的实际采集和数字化，包括：被测信号作为待测量的模拟电信号源；虚拟仪器采集与控制硬件是核心部分，一般由微处理器系统或嵌入式系统构成，主要包括信号采集、信号调理、信号处理与信号通讯等功能，对于示波器需要高速、高精度的 A/D 转换器；接口硬件按照不同标准分为 GPIB、VXI、PXI、DAQ、PC 端口等多种接口总线，用于连接采集控制硬件与计算机平台；计算机硬件平台可以是各种类型的计算机，管理着虚拟仪器的软硬件资源，是虚拟仪器的硬件基础。

软件系统在计算机上运行，实现示波器的核心功能，从低层到顶层包括三个层次：I/O 接口软件驻留于计算机系统中，执行仪器总线的特殊功能，充当计算机与仪器之间的软件层连接；仪器驱动程序是完成对特定仪器控制与通信的软件程序集，是应用软件实现仪器控制的桥梁；应用软件（软面板）直接面向操作用户，提供直观友好的测控操作与显示界面、丰富的数据分析与处理功能。

2. 描述基于虚拟仪器技术的虚拟万用表的功能描述和组成。

功能描述：

基于虚拟仪器技术的虚拟万用表同样利用计算机平台来定义和实现传统万用表的功能。它充分发挥了虚拟仪器仪器功能可由用户自己定义的优势，当用户的测试要求变化时，可以方便地增减软件模块，或重新配置系统以满足测试要求，系统开放灵活。同时实现了软件的即插即用，用户通过友好的用户界面调用不同的软件模块来实现不同的测试功能，减少了人工干预，提高了测试的速度和可靠性。

基本测量功能包括：直流/交流电压测量可以精确测量直流电压和交流电压的有效值、峰值等，输入范围为 0~3.3V，分辨率达 12 位；直流/交流电流测量通过精密采样电阻实现电流到电压的转换；电阻测量采用恒流源法或分压法；通断测量和二极管测量等辅助功能。显示功能可以数字显示、模拟仪表盘显示或趋势图的形式实时显示测量结果。数据记录与存储功能体现了虚拟仪器数据可以进行编辑、存储和打印的优势，测试设备得到的数据实时、直接地通过计算机总线传输到计算机内存或硬盘，可以存储为数据文件或数据库，供以后分析使用。

组成：

虚拟万用表的组成同样遵循虚拟仪器的标准结构，分为硬件系统和软件系统两个主要部分。

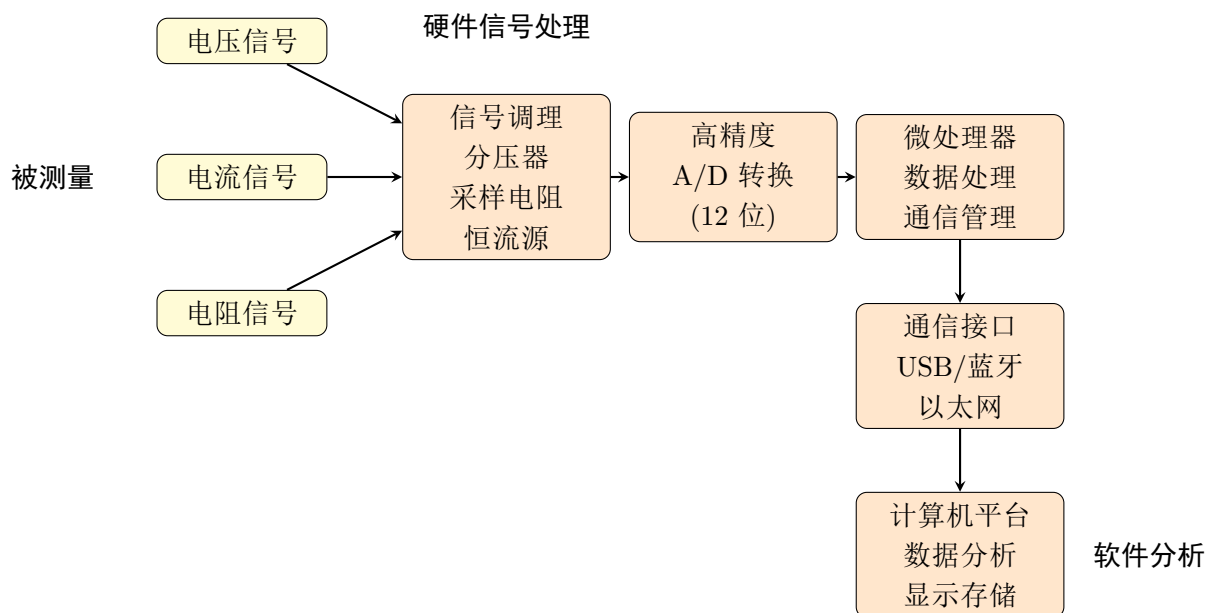


图 3: 虚拟万用表信号处理流程图

硬件系统主要负责将被测量的电学量转换为计算机可识别的数字信号：被测信号包括待测量的电压、电流、电阻等电学量；虚拟仪器采集与控制硬件包含高精度的模拟前端、信号调理电路和高分辨率的 A/D 转换器，同时需要微处理器或嵌入式系统控制采集过程和进行初步数据处理；接口硬件负责数据传输，根据虚拟仪器发展的两条主线，可选择适合高精度测试系统的 GPIB、VXI、PXI 总线方式，或适合普及廉价系统的 DAQ、并行接口、串行接口（特别是 USB 接口）；计算机硬件平台作为核心处理和显示平台，提供计算、存储和图形界面支持。

软件系统在计算机上运行，实现万用表的各项测量、显示和分析功能，同样采用三层结构：I/O 接口软件负责计算机与底层硬件之间的通信接口管理，为仪器驱动程序提供可调用的操作函数集；仪器驱动程序是万用表的专用驱动，负责控制硬件进行测量、读取数字化的测量值并传输给上层应用软件；应用软件（软面板）提供用户友好的图形用户界面，模拟传统万用表的显示屏和功能按钮，充分体现虚拟仪器软件即插即用的特点。

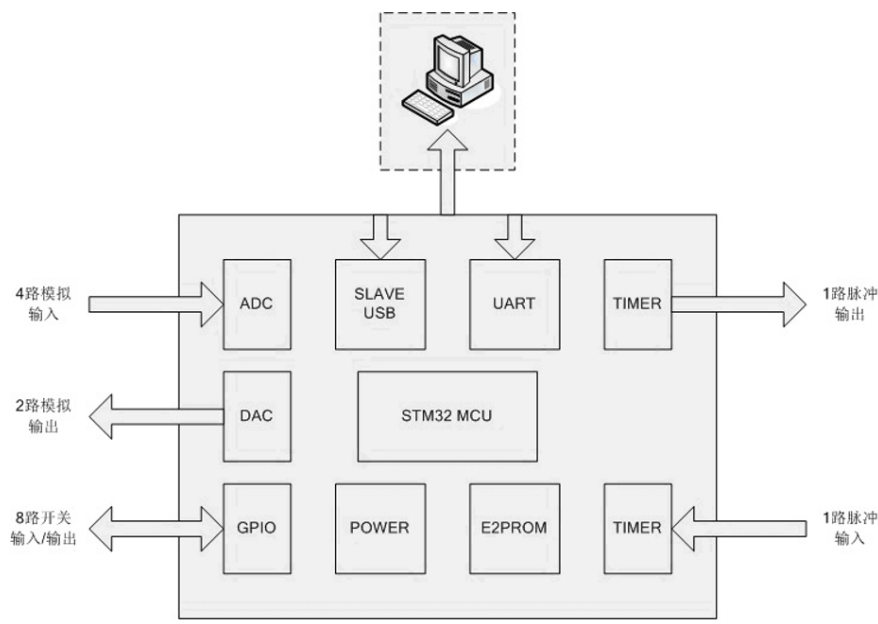


图 4: ZDVI 虚拟仪器硬件设计框图

3. 如果将示例虚拟仪器接口方式从 USB 通讯方式改成蓝牙方式, 该示例硬件需要进行哪些调整?

根据 ZDVI 虚拟仪器的硬件设计示例, 如果将接口方式从 USB 通讯改成蓝牙方式, 硬件需要进行以下调整:

首先需要移除 USB 接口相关芯片和电路。ZDVI 示例中包含两套 USB 接口电路: 从 USB 接口电路采用 STM32F103ZET6 微控制器 (MCU) 内部的 1 路 Slave USB 总线接口, 需要禁用或不使用该内部从 USB 接口功能; 主 USB 接口电路采用外置式主 USB 芯片 SL811, 并与 MCU 采用数据线和控制线进行通讯, 在改为蓝牙时需要移除 SL811 芯片及其相关的外围电路和 PCB 走线。

其次需要增加蓝牙通信模块及其外围电路。需要引入一个蓝牙通信模块, 例如蓝牙串口模块 (如 HC-05, HC-06 等) 或更高级的蓝牙 SoC (System-on-Chip)。这个模块将负责蓝牙协议栈的处理和无线数据传输。蓝牙模块需要与 STM32 MCU 建立通信连接, 通常蓝牙模块提供 UART 接口, 可以利用 STM32F103ZET6 内部的 5 路 UART 总线接口中的一路与蓝牙模块进行连接。根据 ZDVI 硬件设计, 串行接口电路采用 MCU 内部 UART 接口电路, 并通过 SP3232 芯片进行电平转换, 因此可能需要相应调整串行接口的分配和电平转换电路。

第三需要考虑电源电路的适配。ZDVI 系统电源电路采用开关电源芯片 LM1117-3.3 将 5V 电源转化为 3.3V, 支持双种电源供电方式 (总线供电和自供电)。大多数蓝牙模块工作电压为 3.3V, 与现有电源系统兼容, 但需要考虑蓝牙模块的功耗特性。蓝牙模块在发射时功耗较高, 可能需要评估电源电路的负载能力, 必要时增加电源滤波电容或调整电源管理策略。

第四需要考虑 PCB 布局和天线设计。蓝牙作为射频通信, 需要合适的天线设计和 PCB 布局。需要为蓝牙模块预留天线空间 (板载天线或外置天线接口), 确保射频信号的良好传输。同时需要考虑蓝牙模块与其他电路 (特别是时钟电路和模拟电路) 之间的电磁兼容性, 避免相互干扰影响系统性能。

总结来说, 主要调整是废弃或禁用原有的 USB 接口硬件和 MCU 内部 USB 功能, 并集成一个合适的蓝牙通信模块, 通过 UART 等接口与 MCU 连接, 实现无线通信。

4. 给出虚拟万用表的硬件设计方案

基于虚拟仪器技术的虚拟万用表的硬件设计方案，目标是实现电压、电流、电阻等电学量的精确测量，并将数据传输到计算机进行处理和显示。

基于 ZDVI 虚拟仪器的硬件设计经验，虚拟万用表的硬件组成部分及设计要点包括以下几个方面：

信号输入端需要提供安全、标准的输入插孔（如香蕉插孔），用于连接测试表笔。同时需要设置过压/过流保护电路，在输入端设置保险丝、压敏电阻、瞬态抑制二极管等，防止过高电压或电流损坏内部电路，确保人身安全和设备寿命。参考 ZDVI 设计中的开关量信号输入/输出电路，采用 8 路开关量输入/输出与 MCU 内部的 GPIO 模块连接，可以扩展为多路信号输入选择功能。

信号调理电路是核心模块，包括电压测量、电流测量和电阻测量等功能。电压测量需要分压器将高电压输入降至可测量范围，以及缓冲放大器确保高输入阻抗。电流测量需要精密采样电阻实现电流到电压的转换。电阻测量采用恒流源法或分压法。这些调理电路的输出需要与后续的 A/D 转换器匹配，确保信号在合适的电压范围内。

微处理器及数据转换模块方面，可以参考 ZDVI 中采用的 STM32F103ZET6 微控制器，该芯片具有优异的性能特点：采用 8MHz 晶振，通过内部锁相环倍频达到 72MHz 主频；内部集成 512KB Flash 和 64KB SRAM，足够存储复杂的算法和数据；具有 3 个 12 位 ADC，每个 ADC 共用多达 21 个外部通道，可以满足虚拟万用表多路信号采集的需求；2 路 12 位 DAC 输出可用于产生恒流源或校准信号；工作温度范围-40℃~+85℃，适应各种工作环境。

存储电路方面，参考 ZDVI 设计中的存储电路，采用 EEPROM 存储系统参数。可以选用 256 字节的 24LC02 或更大容量的存储器，与 MCU 采用 I2C 总线通讯。STM32F103ZET6 具有 2 路 I2C 总线接口，为存储器连接提供了便利。存储内容包括用户设置、校准数据、测量历史等。

通信接口模块负责将 MCU 处理后的数据传输到计算机。根据 ZDVI 的丰富接口资源，可以选择多种通信方式：5 路 UART 总线接口支持串行通信；1 路 Slave USB 总线接口实现 USB 通信；还可以通过外置芯片（如 ZDVI 中的 SL811）扩展更多接口功能。这种设计为虚拟万用表提供了灵活的选择。

调试与开发接口方面，参考 ZDVI 中的 JTAG 调试电路设计，采用 JTAG 接口电路与 MCU 相连，实现联机调试和程序下载，便于开发和维护。

电源电路设计方面，可以采用 ZDVI 中成熟的双电源供电方案：支持总线供电方式和自供电方式。开关电源芯片 LM1117-3.3 将 5V 电源转化为 3.3V，为 MCU 和其他数字电路供电。对于模拟电路部分，可能需要增加低噪声线性稳压器，确保模拟信号的高精度测量不受电源噪声影响。

报警与指示电路方面，参考 ZDVI 中的报警电路设计，采用 GPIO 与 MCU 相连，实现 BEEP 输出驱动。可以扩展为 LED 指示灯阵列，显示当前测量模式、连接状态、电池电量等信息，提升用户体验。

设计原则应注重高精度、高可靠性、灵活性和成本效益。高精度关键在于选择高分辨率 A/D 转换器和低噪声、高稳定性的模拟前端电路。高可靠性要求选用高质量的元器件，并进行适当的保护电路设计。灵活性需要考虑后续功能扩展的可能性，例如增加频率测量、电容测量等。成本效益要求在满足性能指标的前提下，选择性价比高的元器件和设计方案。

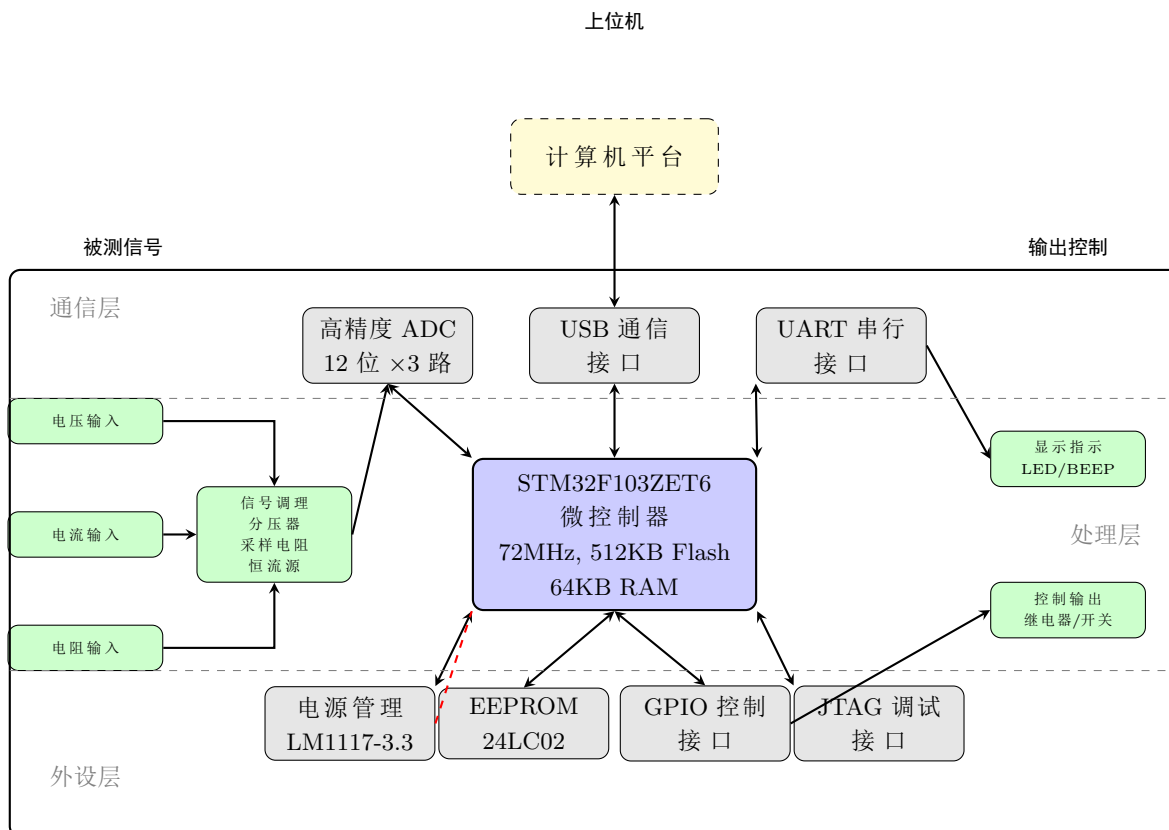


图 5: 虚拟万用表硬件设计架构图

5. 在本示例中，通讯协议不够可靠，如何进行改进以保证通讯的可靠性？

在 ZDVI 虚拟仪器的通讯协议设计中，已经定义了各种设备的基本命令格式，如蜂鸣器控制 (01H + Cmd)、EEPROM 读写 (02H + WR/RD + Addr + Data)、ADC 采集 (03H + ChanID)、DAC 输出 (04H + ChanID + DACValue)、PWM 输出 (05H + FreValue + DutyValue) 和 PWM 输入 (06H) 等。虽然这些协议定义了基本的功能和返回状态 (FFH 成功, 00H 命令出错)，但在实际应用中仍存在可靠性不足的问题。通讯的可靠性涉及数据传输过程中的完整性、准确性和时序保障。

针对 ZDVI 协议的具体特点，提出如表 1 的通讯协议可靠性改进方案。

通过引入这些机制，可以大大提高虚拟仪器通讯的鲁棒性和可靠性，确保数据传输的准确性和完整性，避免因通讯问题导致测量或控制的错误。

6. 如果将示例虚拟仪器接口方式从 USB 通讯方式改成蓝牙方式，该示例固件程序需要进行哪些调整？

根据 ZDVI 虚拟仪器固件的模块化设计架构，如果将接口方式从 USB 通讯改成蓝牙方式，固件程序（运行在 STM32 MCU 上）需要进行以下核心调整：

首先需要移除或禁用 USB 通信模块代码。根据 ZDVI 固件架构，需要移除或禁用以下 USB 相关

表 1: ZDVI 通讯协议可靠性改进方案

改进方案	原始协议示例	改进后协议	改进效果
增加校验机制	DAC 输出: 04 01 07 D0	04 01 07 D0 XX (XX 为 CRC/XOR 校验)	检测传输错误, 避免错误数据执行
帧头帧尾结构	蜂鸣器: 01 01	AA 55 01 01 XX 0D 0A (AA 55 为帧头, 0D 0A 为帧尾)	解决数据包边界识别, 防止粘包错位
数据包长度字段	PWM 输出: 05 27 10 32	AA 55 04 05 27 10 32 XX 0D 0A (04 表示载荷长度 4 字节)	确保数据包完整性, 检测截断数据
命令序列号	任意命令格式	AA 55 LEN SN CMD DATA XX 0D 0A (SN 为递增序列号)	检测丢失、重复、乱序命令
ACK/NACK 确认	返回: FFH/00H	成功: AA 55 02 SN FF 0D 0A 失败: AA 55 03 SN 00 Errorcode 0D 0A	确保命令执行确认, 支持超时重传
细化错误码	00H (命令出错)	01H-格式错误, 02H-参数超范围, 03H-设备忙, 04H-校验失败, 05H-硬件故障	精确诊断错误原因, 便于故障排除
读取命令改进	ADC 读取: 03 00→ 返回数据	命令: AA 55 02 SN 03 00 XX 0D 0A 应答: AA 55 04 SN ADCValueH ADC-ValueL XX 0D 0A	读取数据包含完整帧结构, 确保可验证性

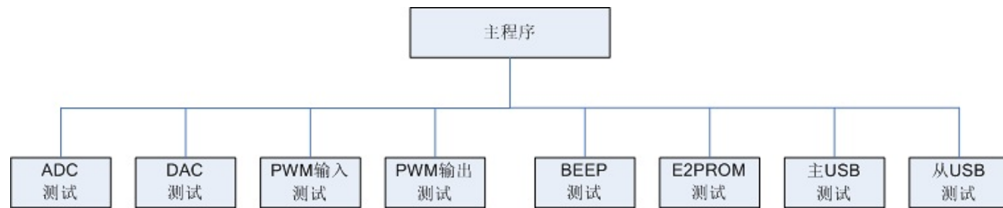


图 6: ZDVI 虚拟仪器固件程序架构

模块:

主 USB 接口模块需要移除的关键函数包括:

```

1 // 初始化SL811主控芯片的函数
2 void Init_SL811(void);
3
4 // 初始化U盘功能的函数
5 void Init_Udisk(void);
6
7 // 文件系统相关的U盘操作函数集
8 // (具体函数名根据实际固件而定)
  
```

从 USB 接口模块需要移除的关键函数包括:

```

1 // 初始化虚拟串口功能
2 void Init_VirtualCOM(void);
  
```



```

3
4 // 虚拟串口数据发送函数
5 void VirtualCOM_Write(uint8_t* pBufferPointer, uint32_t wBufferSize);
6
7 // 虚拟串口数据接收函数
8 uint32_t VirtualCOM_Read(uint8_t* pBufferPointer);

```

其次需要添加蓝牙通信模块代码。可以参考 ZDVI 现有的串口模块设计蓝牙通信模块，包含以下关键函数：

蓝牙模块初始化函数：

```

1 // 初始化蓝牙模块，配置UART接口参数
2 void Init_Bluetooth(void);
3
4 // 配置蓝牙模块工作模式，发送AT命令设置模块名称、配对密码、主从模式等
5 void Bluetooth_Config(void);

```

蓝牙数据传输函数：

```

1 // 通过蓝牙发送数据，替代原有的VirtualCOM_Write函数
2 void Bluetooth_SendData(uint8_t* pBufferPointer, uint32_t wBufferSize);
3
4 // 通过蓝牙接收数据，替代原有的VirtualCOM_Read函数
5 uint32_t Bluetooth_ReceiveData(uint8_t* pBufferPointer);

```

这些函数内部将调用现有串口模块的底层函数：

```

1 // UART数据发送函数
2 void USART_SendData(USART_TypeDef* USARTx, uint16_t Data);
3
4 // UART数据接收函数
5 uint16_t USART_ReceiveData(USART_TypeDef* USARTx);

```

通讯协议适配方面，ZDVI 固件中的各个功能模块（蜂鸣器、EEPROM、ADC、DAC、PWM 等）的上层逻辑可以保持不变，只需要在通信接口层进行替换。原来通过 USB 虚拟串口处理的命令解析和响应逻辑，现在改为通过蓝牙串口处理。具体来说：

```

1 // 蜂鸣器模块的宏定义保持不变
2 BEEP_ON();
3 BEEP_OFF();
4
5 // EEPROM模块的I2C函数保持不变
6 I2C_Write();
7 I2C_Read();
8
9 // ADC模块的采集函数保持不变
10 Get_ADCValue(uint8_t n);
11
12 // DAC模块的输出函数保持不变

```

```
13 DAC_Output(uint8_t n, float value);  
14  
15 // PWM模块的输入输出函数保持不变
```

主程序逻辑调整方面，需要在主程序的初始化部分将原有的 USB 虚拟串口初始化替换为蓝牙初始化，在数据收发处理部分进行相应的函数替换：

```
1 // 原来的USB虚拟串口初始化  
2 // Init_VirtualCOM();  
3  
4 // 替换为蓝牙初始化  
5 Init_Bluetooth();  
6  
7 // 原来的数据发送函数  
8 // VirtualCOM_Write(data, size);  
9  
10 // 替换为蓝牙数据发送函数  
11 Bluetooth_SendData(data, size);  
12  
13 // 原来的数据接收函数  
14 // size = VirtualCOM_Read(buffer);  
15  
16 // 替换为蓝牙数据接收函数  
17 size = Bluetooth_ReceiveData(buffer);
```

其他功能模块的调用逻辑保持不变，体现了 ZDVI 模块化设计的优势。

电源管理方面，需要考虑蓝牙模块的功耗特性。可以在现有的电源管理基础上增加蓝牙模块的电源控制功能，在低功耗模式下通过 GPIO 控制蓝牙模块的开关，以优化整体功耗。

总结而言，核心是剥离 USB 通信相关的所有代码和逻辑，并引入、初始化、配置蓝牙模块，通过 STM32 MCU 的 UART 接口实现数据的发送和接收，同时调整主程序中与通信相关的任务调度和数据流。

7. 如果虚拟仪器驱动程序中引出了全局变量，声明了大型数组结构和复杂的数据类型，会有什么影响？

根据 VPP 规范和虚拟仪器驱动程序设计原则，驱动程序应” 尽量避免引出全局变量，避免声明大型数组结构，避免声明复杂的数据类型（如类、结构、联合等），不采用键盘/屏幕标准输入/输出方式”。这些约束是为了确保驱动程序具有高可靠性、可移植性和性能。如果违反了这些原则，会产生以下不利影响：

这些不当设计会严重违反 VPP 规范的核心要求，大大降低虚拟仪器驱动程序的可靠性、稳定性、可维护性、可移植性和性能。驱动程序作为连接仪器与用户界面的桥梁，应追求简洁、高效和确定性的设计，确保” 仪器函数具有较强的内聚力” 和” 各函数之间提高独立性”，从而满足行业规范对虚拟仪器系统的要求。

表 2: 虚拟仪器驱动程序不当设计的影响分析

问题类型	具体影响	后果说明
引出全局变量	命名冲突和可维护性差	全局变量在整个程序范围内可见，容易与其他模块或应用程序中的变量发生命名冲突，使得代码难以理解和维护，违反 VPP 规范的模块化要求
	数据耦合度高	全局变量导致模块之间的耦合度增加，任何模块都可以修改全局变量的值，使得追踪数据流向和调试变得异常困难，违反“减少彼此的耦合度”原则
	线程安全问题	在多线程环境下，全局变量的访问需要额外的同步机制，否则可能导致数据竞争和系统崩溃
	可移植性差	过度依赖全局变量的驱动程序，其接口和内部实现紧密耦合，使得在不同操作系统、硬件平台下的移植变得困难
声明大型数组结构	内存消耗大	驱动程序运行在内核空间，内核内存有限且宝贵，大型数组会占用大量内存，可能导致内核内存不足和系统崩溃
	栈溢出风险	大型数组作为局部变量声明时存储在栈上，可能导致栈溢出，引发系统崩溃
	性能下降	访问大型数组可能导致缓存未命中率增加，降低数据访问效率，影响驱动程序性能
	实时性影响	在实时系统中，分配和操作大型内存块会引入不可预测的延迟，影响系统的实时响应能力
声明复杂数据类型	增加代码复杂性	在底层驱动程序中引入复杂的类和对象会增加代码复杂性，违反“一个仪器函数完成一个独立的仪器功能”原则
	调试难度增加	复杂的内存布局 and 对象生命周期管理使得调试变得困难，在内核模式下调试复杂数据结构更具挑战性
	兼容性问题	不同编译器或操作系统版本对复杂数据类型的内存布局和调用约定处理方式不同，导致兼容性问题
	资源管理挑战	类的构造和析构可能涉及动态内存分配，在内核模式下需要非常谨慎，容易引发内存泄漏或崩溃
	违反单一功能原则	复杂的类或结构可能封装过多功能，使得驱动程序的內聚力下降，耦合度提高

8. 如果将示例虚拟仪器接口方式从 USB 通讯方式改成蓝牙方式，该示例驱动程序需要进行哪些调整？

根据 VPP 规范，仪器供应厂家在提供虚拟仪器硬件模块的同时，必须提供相应的仪器驱动程序。ZDVI 虚拟仪器驱动程序基于虚拟串口的方式，采用串行的驱动函数设计。如果将接口方式从 USB 通讯改成蓝牙方式，需要在保持 VPP 规范要求的基础上进行以下调整：

1. 驱动程序安装方式调整

原 ZDVI 驱动程序安装过程包括：运行 VCP_V1.3.1_Setup.exe 程序安装虚拟串口驱动程序，在设备管理器中更新 STM32 串口驱动程序。改为蓝牙方式后，需要安装蓝牙驱动程序和相关的蓝牙开发库，可能涉及 Windows Bluetooth API 或第三方蓝牙通信库的集成。

2. 核心驱动函数接口重构

按照 VPP 规范“仪器驱动程序的设计和实现形式必须一致”的要求，保持函数接口的一致性，但底层实现需要调整：

```

1 // 蓝牙初始化函数（替代原串口初始化）
2 function ZDVI_Bluetooth_Init : integer;
3
4 // 保持原有的仪器功能函数接口不变
5 function ZDVI_BeepOn: integer;
6 function ZDVI_BeepOff: integer;
```

```

7 function ZDVI_E2PROM_Write(E2PROM_Addr : Char; E2PROM_Data : Char): integer;
8 function ZDVI_E2PROM_Read(E2PROM_Addr : Char; E2PROM_Data : PChar): integer;
9 function ZDVI_ADC(ChanID : integer; ADC_Value : PWord): integer;
10 function ZDVI_DAC(ChanID : integer; DAC_Value : Word): integer;
11 function ZDVI_PWMOUT(FreValue : Word; DutyValue : Byte): integer;
12 function ZDVI_PWM_Measure(FreValue : PWord; DutyValue : PByte): integer;

```

3. 遵循 VPP 规范的设计调整

根据 VPP 规范要求，调整后的驱动程序仍需保持以下特点：

符合返回状态要求：所有仪器函数必须包含一个返回状态值，蓝牙通信中可能增加连接状态、配对状态等新的错误码。

保持模块化结构：能够提供多级功能访问，底层蓝牙通信模块独立封装，上层仪器功能函数保持不变，体现“仪器驱动程序的结构一定要模块化”。

维持内聚力原则：遵循“一个仪器函数完成一个独立的仪器功能”，蓝牙连接管理与具体仪器功能分离。

避免复杂数据类型：在蓝牙通信实现中仍需“避免声明复杂的数据类型”，使用简单的数据结构进行蓝牙数据包的构建和解析。

4. 通信协议适配

虽然上层应用仍使用 ZDVI 定义的命令格式（如蜂鸣器 01H+Cmd、EEPROM 02H+WR/RD+Addr+Data 等），但底层传输需要适配蓝牙协议。保持数据耦合方式，避免功能耦合，符合 VPP 规范“函数之间必要的耦合采用数据耦合方式”的要求。

5. 保持高效和简洁性

遵循“仪器函数应避免包括高级的、复杂的数学分析与运算函数”原则，将蓝牙连接管理、协议处理等复杂逻辑封装在底层，上层仪器功能函数保持简洁高效。

通过这些调整，确保改进后的蓝牙驱动程序既满足功能需求，又符合 VPP 规范对虚拟仪器驱动程序的设计要求，保持良好的可维护性、可移植性和用户体验。

9. 结合实例，描述一下在 GUI 软件设计中同时支持生手和熟手的必要性。

在 GUI 软件设计中同时支持生手（新手用户）和熟手（经验丰富的用户）是至关重要的，因为这能大大提高软件的易用性、效率和用户满意度。不同的用户群体有不同的需求和使用习惯。

必要性体现在以下几个方面。提高易用性和降低学习成本（为生手），对于新手用户，一个直观、易于理解的界面能够降低他们的学习曲线，减少培训成本和客户支持成本。他们需要明确的指引、简单的操作流程和清晰的反馈。提高效率和生产力（为熟手），熟练用户通常追求效率和速度。他们熟悉软件的功能和操作，希望能够快速完成任务，避免不必要的步骤。为他们提供快捷方式、高级功能和自定义选项可以大大提高他们的工作效率。适应用户成长，用户从生手逐渐成长为熟手的过程中，软件应该能够适应这种变化，提供逐步深入的功能和更高效的操作方式。扩大用户基础，兼顾不同经验水平的用户可以吸引更多广泛的用户群体，从而提高产品的市场竞争力。

以 EDA 软件为实际案例，传统的 Cadence 软件在这方面存在明显不足。Cadence 界面复杂繁琐，功能按钮密密麻麻，新手用户面对如此复杂的界面往往不知所措，需要数月的专业培训才能掌握基本操作。而对于熟手用户，虽然功能强大，但操作步骤冗长，效率不高，缺乏有效的快捷操作方式。

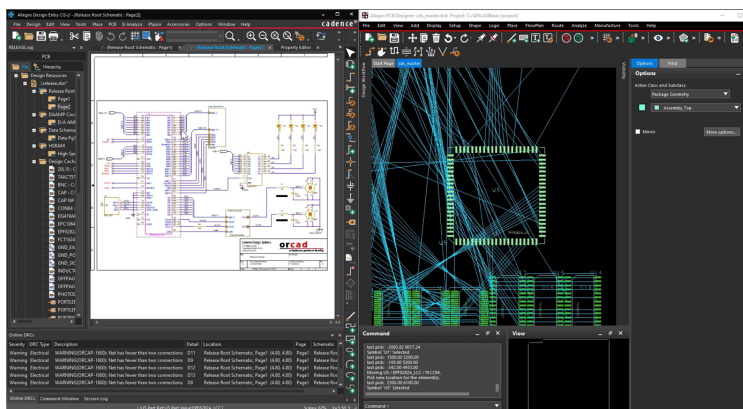


图 7: Cadence EDA 软件界面 - 复杂繁琐，不利于生手熟手的统一体验

相比之下，国产的嘉立创 EDA 软件很好地体现了同时支持生手和熟手的设计理念。对于新手用户（当年大二暑假电路综合短学期的我们），嘉立创提供了清晰的界面布局、直观的操作流程和丰富的帮助提示，降低了学习门槛；对于熟练用户，提供了快捷键操作、批量处理功能和高效的工作流程，大大提高了设计效率。这种设计使得嘉立创 EDA 能够服务于不同经验水平的工程师，真正实现了 GUI 设计的包容性和实用性。

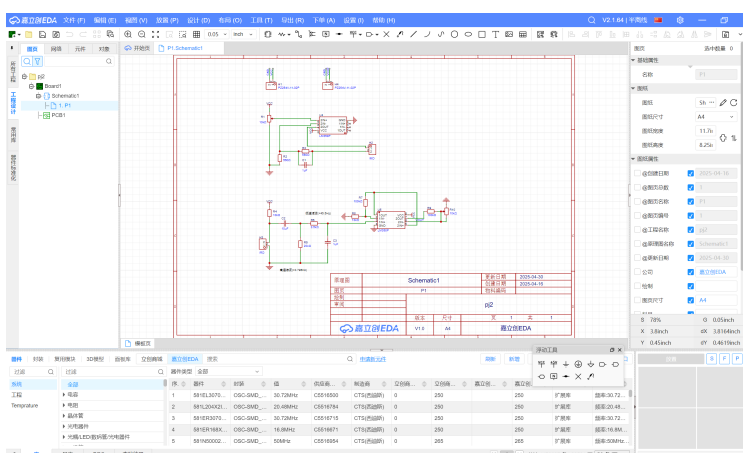


图 8: 嘉立创 EDA 软件界面 - 简洁高效，兼顾生手和熟手需求

10. 结合实例，描述一下在 GUI 软件设计中颜色与字体的设计要点。

在 GUI 软件设计中，颜色与字体是影响用户体验和信息传递效率的关键因素。以下是设计中应注意的事项，并以虚拟仪器软面板为例进行说明：

颜色设计要点包括以下几个方面：

一致性与标准方面，建立一套颜色标准并在整个应用中始终遵循。不同功能或状态的元素应使用一致的颜色编码。虚拟万用表软面板中，所有表示“测量结果”的数字显示框应使用统一的颜色（如绿色或蓝色）；所有表示“错误/警告”的信息框使用红色或橙色；“启动/停止”按钮使用不同的颜色以区分

状态。混乱的颜色使用会导致用户困惑，降低软件的专业性和可信度。

对比度与可读性方面，确保文本和背景之间有足够的对比度，以保证可读性。对于关键信息，对比度应更高。在虚拟示波器软面板中，波形线的颜色应与背景（如黑色或深灰色）形成鲜明对比，确保波形清晰可见。测量结果的数字显示（如“采样电压：3.20V”）应使用与背景色有高对比度的字体颜色。对比度不足会导致用户难以阅读信息，尤其是在光线不好的环境下或对于视力受损的用户。

避免过度使用颜色方面，颜色应有目的性，避免使用过多的颜色或过于鲜艳的色彩，这会分散用户注意力，造成视觉疲劳，并使界面显得杂乱无章。虚拟示波器软面板不应使用十几种不同的颜色来标记各种控件。核心信息区应保持简洁，突出数据本身。过度使用颜色会降低界面的专业性，使用户感到压抑或混乱。

颜色语义与文化背景方面，考虑颜色的普遍语义和文化背景。例如，红色通常表示停止、错误或危险；绿色表示启动、成功或安全。在虚拟仪器软面板中，用于“停止采样”的按钮可以使用红色，而“开始采样”的按钮可以使用绿色。当测量值超出安全范围时，用红色字体或背景警告用户。违背用户对颜色的普遍认知会造成误解和操作失误。

支持色盲用户方面，避免仅通过颜色来传递重要信息。对于色盲用户，某些颜色组合（如红绿色）难以区分。除了用颜色区分波形，还可以用不同的线型（实线、虚线、点线）或在图例中明确标注来区分不同通道的波形。错误信息除了红色，还可以配以感叹号图标。忽视色盲用户会导致部分用户无法正常使用软件，影响软件的无障碍性。

字体设计要点包括以下几个方面：

易读性方面，选择易于阅读的字体，通常是无衬线字体（如 Arial, Helvetica, Segoe UI, Noto Sans）在屏幕上表现更好。避免使用过于花哨或艺术性的字体。虚拟万用表显示测量结果的数字（如“0.000V”）应使用清晰、粗细适中的数字字体，确保即使远距离也能快速辨认。难读的字体会导致用户阅读疲劳，降低信息获取效率。

字号与行距方面，确保文字大小适中，既不过大占用空间，也不过小难以辨认。行距应足够，以避免文字堆叠，提高阅读舒适度。标题可以使用稍大的字号，而描述性文字和数据标签可以使用较小的字号，但都应保持在可读范围。不合适的字号和行距会影响信息呈现的美观性和阅读体验。

粗细与样式方面，通过字体粗细（Bold）、斜体（Italic）等样式来强调重要信息，但避免过度使用，以免分散注意力。在虚拟示波器中，当前活动的测量值或最重要的告警信息可以使用粗体字显示，以突出其重要性。过度使用样式会使得界面显得杂乱，降低视觉焦点。

一致性方面，在整个应用中保持字体类型和样式的统一性。例如，标题使用一种字体，正文使用另一种，并且这些字体在所有界面中保持一致。虚拟仪器软面板的所有菜单项、按钮文本、标签和数据输入框应使用一套预定义的字体家族和大小规则。字体的不一致会使得界面显得不专业、不协调。

国际化考虑方面，如果软件需要支持多语言，确保所选字体包含所有必要的字符集，并且在不同语言环境下显示效果良好。字体无法正确显示某些语言的字符会导致乱码或界面损坏。

通过精心设计颜色和字体，虚拟仪器软面板不仅能提供清晰、准确的信息，还能提升用户的操作舒适度和整体满意度，从而使得软件更具吸引力和实用性。

以 EDA 软件的颜色与字体设计为例，可以清楚地看到设计差距。传统的 Cadence 软件在颜色搭配上过于保守，大量使用灰色调，缺乏层次感，导致界面单调沉闷；字体选择上使用较小的系统默认字体，在高分辨率显示器上可读性差；颜色编码不够直观，用户难以快速识别不同功能区域；整体视觉效果给人以过时、不友好的印象，影响用户的使用体验和工作效率。

相比之下，嘉立创 EDA 软件在颜色与字体设计方面体现了现代化软件的优秀标准：采用了清新的配色方案，合理运用对比色突出重要功能；字体大小适中，层次分明，即使长时间使用也不会造成视觉疲劳；颜色语义明确，如绿色表示连接正常、红色提示错误警告，符合用户的直觉认知；整体设计简洁美观，提升了软件的专业形象和用户满意度。这种优秀的颜色与字体设计不仅提高了软件的易用性，更

重要的是体现了以用户为中心的设计理念，使得工程师能够在舒适的视觉环境中高效完成 PCB 设计工作。

11. 描述图形化编程软件平台与可视化编程工具的异同点。

图形化编程软件平台（如 LabVIEW、VEE）与可视化编程工具（如 Visual Basic、Delphi、C++ Builder、Visual C++、Java、Python 的 GUI 库）在软件开发领域都提供了可视化的开发环境，但它们在设计理念、应用领域和编程范式上存在显著差异。

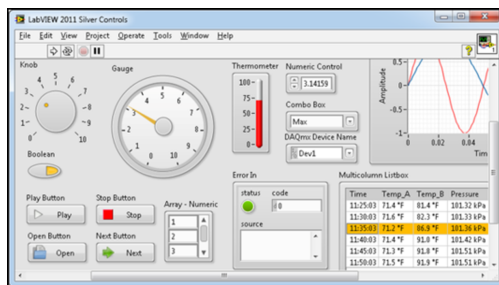


图 9: LabVIEW 图形化编程界面示例

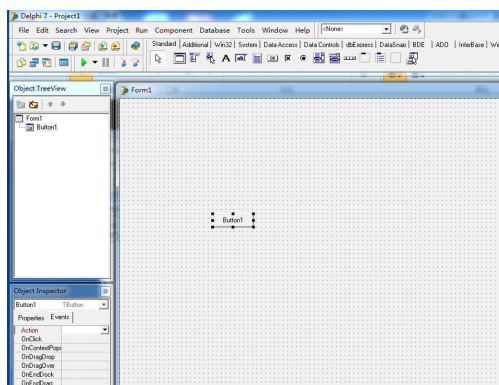


图 10: Delphi 可视化编程界面示例

相同点包括以下几个方面：

可视化界面设计方面，两者都提供了可视化的界面设计环境，允许开发者通过拖放、点击等方式布局用户界面元素（如按钮、文本框、图表等），而无需手动编写大量复杂的 GUI 代码。如图形化编程中的 Scratch 和 Blockly 都采用积木式拖拽编程。

简化编程方面，都旨在简化编程过程，提高开发效率。通过封装控件和自动化代码生成，减少了从头开始编写代码的工作量。

基于对象概念方面，它们都基于对象、属性、事件和方法的概念。在可视化编程中，对象主要分为窗体（Form）和控件（Control）；属性（Property）指对象所具有的特征；事件（Event）是发生在对象上的动作；方法（Method）是对象本身内含的函数或过程。开发者通过设置对象的属性来配置其外观和行为，并通过编写事件响应函数来处理用户交互或系统事件。

适合快速应用开发（RAD）方面，两者都支持快速应用程序开发模式，能够快速构建应用程序原型和最终产品，具有编程简单、自动生成程序代码、效率高的优点。

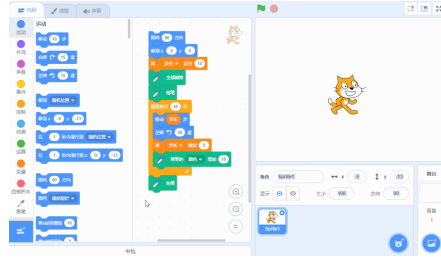


图 11: Scratch 积木式图形化编程界面

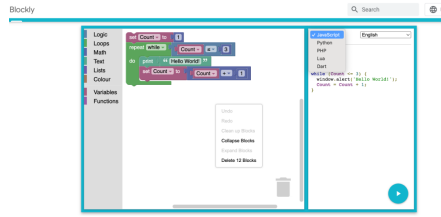


图 12: Blockly 图形化编程界面

不同点主要体现在表 3 中的几个方面：

表 3: 图形化编程软件平台与可视化编程工具的核心差异对比

对比维度	可视化编程工具	图形化编程软件平台
编程范式	命令式/面向对象编程范式，核心编程是文本驱动，通过指令控制程序流程	数据流式/图形化编程范式，源代码是图标与连线，数据驱动运行模式
主要使用者	软件开发者、程序员，需要较强编程基础	仪器工程师、科研人员，让其回归本职工作而非沉溺于编程细节
源码形式	核心代码为一行行文本代码，编程需要技巧	整个程序流程是用户可现场组态的数据流程图，用鼠标连线产生源代码
运行机制	控制流程型运行，程序从上到下执行	数据流驱动型运行，执行顺序由数据有效到达次序决定，支持并行处理
可运行性	可视化界面是图形化的，但源代码还是文本方式的	框图必须是可运行的，描述系统运行过程，每个部分都会执行操作
动态修改能力	不支持或仅支持部分结构修改，通常只能修改参数值	可对程序结构和运行流程进行动态修改，不只是参数设定
适用领域	通用应用程序开发，关注软件逻辑、算法实现	专用于测量控制、信号处理、数据分析、仪器集成领域
特殊功能	以通用数据类型为主	提供仪器专用数据类型（复数、频谱、时域数据等）、复杂数学分析能力、虚拟仪器接口

从实际应用角度来看，可视化编程工具如 Delphi 虽然提供了可视化界面设计，但仍然需要编写大量文本代码来实现业务逻辑，适合有编程基础的开发者。而图形化编程软件平台如 LabVIEW 则是为了让仪器工程师从繁重的编程任务中走出来，专注于真正的系统集成任务。它提供了仪器特有的数据类型自动识别与转换功能、复杂的数学分析能力、丰富的数据显示方式（数学字母显示、模拟仪表显示、极坐标显示、时域波形显示、频谱图形等）以及易于生成数据报表的功能。

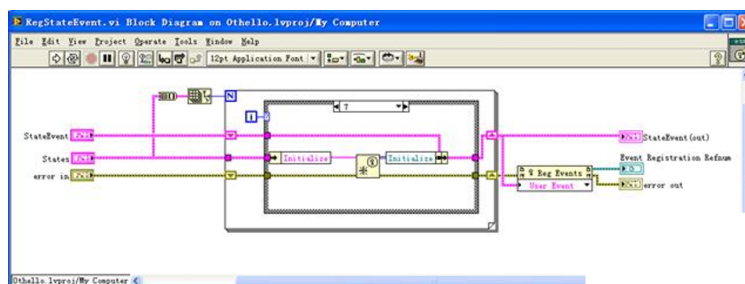


图 13: LabVIEW 数据流编程示例 - 体现图形化编程的数据流特色

12. 描述图形化编程软件平台与一般的绘图工具的异同点。

图形化编程软件平台（例如 LabVIEW、VEE）与一般的绘图工具在表面上都使用图形化界面，但它们的本质和用途存在根本差异。可视化技术是当前发展迅速并引人注目的技术之一，它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图像的形式表现出来。

相同点包括以下几个方面：

视觉化界面方面，两者都提供图形化界面，允许用户通过拖放、连接等方式创建和编辑图形元素。

符号表示方面，都使用图形符号（如方框、图标、线条）来表示概念、对象或流程。

直观性方面，相较于纯文本，图形化的方式通常更直观，易于理解和描述事物。

不同点主要体现在表 4 中的几个方面：

表 4: 图形化编程软件平台与一般绘图工具的根本差异

对比维度	图形化编程软件平台	一般绘图工具
设计目的	编程和控制实际硬件系统，创建可执行程序完成测量、控制、分析任务	创建静态图形、图表、示意图，主要用于可视化表达和文档制作
可运行性	框图必须可运行，描述系统运行过程，每个部分执行实际操作	创建的图形通常是静态的，仅用于展示说明，不具备可执行性
核心概念	基于数据流编程，连线承载实际数据传输，支持并行处理	连线仅为视觉元素，表示关系或流程，不承载实际数据
功能复杂性	提供专业编程功能：变量管理、条件判断、循环控制、函数调用、错误处理	主要关注图形绘制、编辑、格式化和布局，不涉及编程逻辑
硬件交互	专门与硬件设备交互：数据采集、测量仪器、控制器，处理实时信号	通常不与硬件直接交互，主要处理图形数据
专业应用	专用于测量控制、仪器自动化，提供信号处理、数据分析、仪器驱动函数库	适用于文档制作、演示文稿、网页设计、艺术创作等通用应用
动态修改	可对程序结构和运行流程进行动态修改，支持运行时改变程序逻辑	只能修改图形外观属性（颜色、大小、位置），不涉及程序逻辑
输出结果	控制信号、测量数据、分析结果等功能性输出，直接作用于物理世界	图像文件、图形对象或打印材料，主要用于展示和沟通
实时性	用于实时或准实时应用，需要快速响应外部事件和处理数据流	主要处理静态内容，对实时性无特殊要求

在现实生活中和科学研究过程中，人们常常以框图来描述事物。框图不仅可以描述事物的位置、大小，更为重要的是，它可以描述事物的运行过程。在计算机上以二维平台上的框图描述一个程序的结构和运行过程，就形成了图形化语言。与一些框图绘制系统不同的是，图形化语言需要满足两个规范：第一，图形化语言中的框图必须是可运行的，描述的是系统的运行过程，并且每个部分在运行中会执行一

定的操作；第二，框图可以动态修改，这里的修改不只是设定一些参数或变量值，框图编辑环境可以修改程序的结构和运行的流程。

总之，虽然两者都使用图形化界面，但图形化编程软件平台本质上是一个专业的编程环境，而一般的绘图工具是一个图形创作工具。前者的重点在于功能实现和系统控制，后者的重点在于视觉表达和信息展示。

13. 如果将示例虚拟仪器接口方式从 USB 通讯方式改成蓝牙方式，该示例软面板需要进行哪些调整？

根据软面板的功能特点，它作为虚拟仪器模块的控制接口，负责用户与仪器的交互界面，可以通过 VB、VC、Delphi 或 LabVIEW 等开发。



图 14: 虚拟仪器软面板主界面示例

如果将 ZDVI 虚拟仪器接口方式从 USB 通讯方式改成蓝牙方式，该示例软面板需要进行以下调整：

首先需要移除或禁用 USB 虚拟串口选择功能。ZDVI 示例软面板中可能包含选择串口号（如 COM8）和波特率（如 9600）的界面控件。由于通信方式改为蓝牙，这些基于虚拟串口的选项将不再适用。需要移除或禁用这些串口相关的下拉菜单、输入框和初始化按钮。

其次需要添加蓝牙设备选择与连接界面。软面板需要提供新的用户界面元素来管理蓝牙连接。这包括蓝牙扫描按钮，允许用户扫描附近的可用蓝牙设备；设备列表，显示扫描到的蓝牙设备列表，供用户选择目标设备；连接/断开按钮，用于建立或断开与选定蓝牙设备的连接；连接状态指示器，显示当前蓝牙连接的状态（如“未连接”、“正在连接”、“已连接”、“连接失败”等）；配对/绑定功能（如果需要），如果蓝牙设备需要配对才能通信，软面板可能需要引导用户完成配对过程（例如，输入 PIN 码）。

第三需要进行底层通信库/API 的替换。软面板的底层通信代码不再调用基于串口的 API（如 Windows API 中的 CreateFile、ReadFile、WriteFile 或 Delphi/VB 中与串口相关的组件），而是需要调用操

作系统的蓝牙通信 API。例如，在 Windows 平台上，可能需要使用 Windows 蓝牙 API（如 Bluetooth Low Energy GATT API 或 Windows.Devices.Bluetooth 命名空间）来发现设备、建立连接和进行数据读写。这意味着软面板应用程序所引用的底层通信库需要从串口库更换为蓝牙通信库。

第四需要更新错误处理和用户反馈。蓝牙通信会引入新的错误类型，例如“蓝牙设备未找到”、“连接超时”、“配对失败”、“信号干扰”等。软面板需要能够捕获这些错误，并以用户友好的方式向用户显示（例如，弹出错误消息框）。当蓝牙连接状态变化时（例如，从连接变为断开），软面板上相应的状态指示器需要及时更新，并可能伴随提示音或文字提示。

最后需要适配数据传输协议。虽然上层逻辑（如发送“01H + Cmd”控制蜂鸣器）可能保持不变，但底层传输机制需要适应蓝牙协议。软面板发送和接收的数据需要通过蓝牙接口进行封装和解封装。

总结来说，软面板的调整核心在于用户界面（GUI）方面移除旧的 USB 虚拟串口相关控件，添加新的蓝牙连接管理控件；底层代码方面将原有的串口通信代码替换为蓝牙通信 API 的调用；错误处理方面增加对蓝牙特有错误的识别和用户友好提示。这些调整将确保软面板能够通过蓝牙接口正确地与虚拟仪器硬件进行通信，并为用户提供直观的蓝牙连接管理功能。

14. 从软面板设计过程中，描述提高软面板操作过程中可靠性的方法。

从虚拟仪器软面板设计过程的角度，提高操作过程中的可靠性是确保用户能够稳定、准确、高效地使用仪器的关键。可靠性涉及减少误操作、提高系统稳定性以及及时反馈问题。以下是一些方法：

首先需要遵循 GUI 设计原则，特别是“一致性”和“阐明规则”。一致性要求界面元素（如按钮、文本框、图表）的布局、颜色、字体和行为应在整个软面板中保持一致。例如，所有输入框的格式都应相同，所有确认操作的按钮都应有统一的放置位置和颜色。阐明规则要求明确告知用户每个控件的功能和操作限制，例如，某个输入框只接受数字，或者某个功能只有在特定模式下才可用。这样可以减少用户学习成本和操作时的不确定性，降低误操作的可能性。当用户界面越直观，就越易用，越易用就越便宜，因为界面越好，培训用户就越容易，降低了培训成本；界面越出色，用户就越少求助，降低了客户支持成本。

其次需要提供清晰、及时的用户反馈。操作反馈要求用户执行任何操作（如点击按钮、拖动滑块）后，软面板应立即给出视觉或听觉反馈，表明操作已被接收。例如，按钮被点击时改变颜色，或者滑块拖动时实时显示数值变化（如 DA 输出面板中的旋钮和数字显示）。进度反馈要求对于耗时较长的操作（如数据采集、文件保存），显示进度条或状态信息，避免用户误以为软件卡死而重复操作。错误反馈要求当操作失败或发生错误时，提供清晰、具体且可操作的错误消息，指导用户如何解决问题，而不是模糊的错误代码。例如，显示“设备未连接，请检查 USB 线缆”而非“错误码：0x00A1”。及时、准确的反馈能让用户了解系统状态，增强信任感，减少不必要的重复操作或沮丧感。

第三需要进行输入验证和约束。实时验证要求对用户输入的数据进行实时校验。例如，在频率输入框中，如果用户输入的不是数字或超出了 20-50KHz 的范围，应立即给出提示。默认值和范围限制要求为输入控件提供合理的默认值，并强制限制输入范围。例如，在 DA 输出面板中，D/A 输出电压范围为 0-3.3V，应确保用户不能输入超出此范围的值。这样可以防止无效或错误的数据输入导致仪器工作异常，从源头上提高操作的正确性。

第四需要“灰掉”而不是“移除”不可用功能。当某些功能在当前系统状态下不可用时，应将其对应的控件设置为灰色（禁用），而不是从界面上消失。这样可以告知用户该功能存在但目前不可用，避免用户困惑或误操作，同时保持界面的稳定性。

第五可以提供撤销/恢复功能（如果适用）。对于某些可逆的配置或设置操作，提供“撤销”按钮，

允许用户回退到之前的状态，减少因误操作造成的损失。这样可以增加用户尝试和探索的信心，降低操作风险。

第六需要防止意外关闭或丢失数据。在用户尝试关闭未保存的配置或数据时，提供确认对话框，提醒用户保存。这样可以避免用户因疏忽而丢失重要数据或设置。

第七需要进行模块化设计和功能分配。将软面板的不同功能模块化，每个模块负责一个独立的仪器功能。例如，蜂鸣器面板、存储器面板、AD 采集面板等。这样可以使得界面结构清晰，用户更容易理解和操作，降低操作复杂性，同时也有利于维护和升级。

第八需要考虑操作焦点处打开窗口。当某个操作需要打开新窗口时，确保新窗口在用户的当前操作焦点附近弹出，减少用户的鼠标或视线移动，保持操作流畅性。这样可以优化用户体验，降低操作失误率。

最后需要遵循平台的人机交互习惯。如果软面板运行在 Windows 系统上，应遵循 Windows 的 GUI 设计规范（如按钮、菜单、快捷键的习惯用法）。这样用户会感到熟悉和舒适，从而减少学习成本和误操作。

通过上述方法的综合运用，可以显著提高虚拟仪器软面板在操作过程中的可靠性，确保用户能够高效、准确、安全地完成测量和控制任务。