53-作业报告

程序设计实习大作业 NULL 组

June 2025

1 程序功能介绍

本组大作业程序是基于 PySpice 库和 PyQT5 库实现的电路模拟器。

主要功能是在工作区将电路元件及其连接进行可视化，以及电路中各个节点的信号、直流电压的仿真模拟和可视化。

目前支持的元件类型：

基础元件：电阻 R（默认阻值 1000 欧），电压源 V（默认电压 5 伏），接地元件 GND

非线性元件：二极管 D（默认特征电流 Is 为 1e-14 安培）

交流元件：电容 C（默认电容值为 1 微法），电感 L（默认电感值为 1 毫亨），交流源（目前只支持正弦波形和方波波形）

辅助功能包括：

1. 电路文件的保存、打开、另存等操作；
2. 主窗口外观的自定义；
3. 元件移动，元件删除，添加元件等基础操作的 Undo 和 Redo。

2 项目代码结构及类设计细节

主要内容也更新在 README.md 中。

2.1 main.py

启用主窗口 CircuitSimulator。

非常简洁，只有四行代码。

2.2 stimulation.py

定义主窗口 CircuitSimulator(QMainWindow)。

主窗口类中的主要成员：

成员	代称	类名
元件列表区	self.component_dock	QDockWidget
参数编辑器	self.param_editor	ParameterEditorDock(QDockWidget)
工作区	self.scene	CircuitScene(QGraphicsScene)
输出终端	self.terminal	TerminalWidget
工具栏	self.menuBar()	自带成员
状态栏	self.statusBar()	自带成员
引脚电压显示	self.voltage_label	QLabel
快捷键管理	self.shortcut_manager	shortcutManager
命令管理器	self.command_manager	CommandManager

表 1: 主窗口 CircuitSimulator 的主要成员

2.3 components.py

实现引脚类 PinItem(QGraphicsEllipseItem)，以及可以基于鼠标互动实现引脚连接的工作区类 CircuitScene(QGraphicsScene)，以及连线类 WireItem(QGraphicsPathItem)

其中，连线类 WireItem 中，保留了 self.begin_pin 和 self.end_pin，维护连线的起始位置和结束位置的节点信息。

引脚类 PinItem 中，保留了：

self.pin_name 记录引脚的名称

self.parent_component 记录引脚所属的元件实例

self.connected_wires 用于存储以该引脚为端点的所有 WireItem

self.node_name 记录在仿真过程中，给该引脚赋予的节点名称

self.voltage 记录上一次仿真的直流电压值

self.ac_voltage 记录上一次仿真的波形

2.4 ComponentItem.py

实现视图元件类 GraphicComponentItem(QGraphicsPixmapItem)。

init 中补全了：

self.name 记录元件的编号名称

self.spice_type 记录元件的类别

self.pins 记录元件所有的引脚对象

self.param 记录元件所需要的参数等

2.5 spice_generator.py

定义两个函数：对给定的 CircuitScene 对象，给出符合 Spice 语法的 Spice 网表的函数 generate_spice_netlist(scene)；以及根据 CircuitScene 对象检查电路性质的函数 validate_connections(scene)。

2.6 basic.py,AC_source.py

在视图元件类 GraphicComponentItem 基础上派生了其他元件的元件类，规范了元件参数的格式。

包括：电阻元件类 ResistorItem，电压源元件类 VoltageSourceItem，电容元件类 CapacitorItem，电感元件类 InductorItem，接地元件类 GroundItem，二极管元件类 DiodeItem。

在视图元件类 GraphicComponentItem 基础上派生两个元件类 ACSourceItem 和 OscilloscopeItem，提供对交流电路分析的支持。

另外还定义了 OscilloscopeWindow(QMainWindow)，独立于主窗口来显示波形

2.7 shortcuts_manager.py,ai_manager.py

基于快捷键管理器 shortcutManager(QObject) 和快捷键设置器 shortcutSettingDialog(QDialog) 实现的管理快捷键功能。

基于 openai 库实现的基础 agent 对话功能。

2.8 files_manager.py

实现了对电路图文件的所有输入/输出（I/O）管理逻辑，封装了新建、打开、保存和另存为等功能，使用 QFileDialog 与用户交互选择文件路径。通过将画布上复杂的对象（元件、导线及其所有属性）序列化为结构化的

JSON 数据并写入文件来实现，以及在打开文件时读取 JSON 数据并反序列化，从而在画布上精确地恢复整个电路场景。

2.9 command_manager.py

定义命令管理器 CommandManager 用于统一管理模拟器的操作，操作包括：添加元件删除元件添加连线删除连线清空场景

命令的基类是 Command(ABC)，在基类上派生了如下子类：AddComponentCommand,RemoveComponentCommand,MoveComponentCommand，AddWireCommand，RemoveWireCommand，ClearSceneCommand，子类各自支持对应操作的撤回和重做。

2.10 parameter_editor.py, terminal.py

定义了一个编辑元件参数的侧面窗口 ParameterEditorDock(QDockWidget)

定义了一个用于输出所有必要的信息的下方窗口 TerminalWidget(QPlainTextEdit)

2.11

3 小组成员分工情况

见下表

姓名	分工内容
谢尚杰	1. 元件类、节点类、连线类的基本架构
	2. 连线功能
	3. 生成和检查电路网表以及电路模拟仿真功能
	4. 波形和电压可视化功能
王玺傲	1. 文件操作功能
	2. 撤回和重做功能
	3. 快捷键设置功能
刘子豪	1. 窗口外观优化
	2. 背景自定义功能
	3. 快捷键

表 2: 分工内容一览

4 项目总结与反思

4.1 开发流程

小组的选题在确认分组前后（4.20）完全确定。电路模拟器的想法主要是基于组员都对电路有一定的了解，并且元件和连接的动态与 QT 的功能非常契合。在此基础上我们找到了用于电路拟真的 python 库 PySpice，用于支持更复杂的电路计算。

此后小组进行了几次集中讨论。第一次讨论基于组长给出的主框架，解决了 github 的使用和环境的统一问题，并给出了下面的改进方向：

1.wires 外观的优化以及同元件一起移动的功能
2.UI 界面的美化
3. 实现电路连通性检查的函数
4. 实现更多的元件类型
5. 实现仿真结果的可视化（例如电压值显示等）
6.pins 的优化（例如接触提示和接触面积放大）
7. 实现网络的保存和加载
8. 运行 spice 与用户操作的分离
9. 撤回等操作
10. 调节元件参数

第三次集中讨论时，第一次集中讨论定下的 10 个目标已经实现 7 个，在此基础上我们重置了改进方向如下：

1.wires 外形的优化
2.UI 界面美化
3. 在终端接入 ai 模型
4. 运行 spice 与用户操作的分离
5. 撤回操作的完成
6. 示波器功能完善

第四次集中讨论是在路演前，确定了 version1.0 终稿，完成了演示视频的摄制。

4.2 分工模式

我们小组的分工模式比较自由，基本上是基于共同讨论下来的目标，每个人选择对应的一些作为阶段的任务。在此基础上互相帮助和监督。

大体上讲，组长谢尚杰主要负责的是电路连接仿真的算法以及可视化方法，即和 pyspice 库相关联的部分；组员王玺傲主要负责的是电路元件的保存读取和操作的管理，即应用的基础功能部分；组员刘子豪主要负责的是窗口的外观设计以及快捷键等功能，即主要的 UI 设计的部分。

4.3 项目反思

我们的项目虽然比较完整，但是在开发过程中我们也发现了一些不足，尤其是在进一步了解了一些商业化的电路模拟器之后，我们的模拟器更显得班门弄斧。

相比于成熟的电路模拟器，我们程序的明显短板有：

1. 支持的元件类型过少，适合的应用场景比较狭窄，比如中学物理教学等。如果可以完善数字电路，或者支持更丰富的非线性元件，模拟器的实用性可能会增加。因此可能需要除了 pyspice 之外其他库的支持。

2. 波形可视化的方法过于单调。一方面每次打开示波器窗口的波形显示都基于上一次运行 spice 仿真的结果，不能实时更新；另一方面我们将示波器设计成了一个元件，在检测信号的通道数上有限制，实际上一些模拟器将波形显示独立于元件来设计，会更灵活。

3. 没有与其他模拟器兼容的能力。我们设计的电路文件是一个类似于 json 的.circuit 文件格式。除此之外模拟器没有其他更多样化输出的功能，比如说绘制电路图，输出截图，输出波形信息等。 4.UI 设计还有很大的优化空间。