

SLiCAP from import until creation of a circuit

```
from SLiCAP import *
    from SLiCAPinstruction import *
    from SLiCAPexecute import *
    from SLiCAPyacc import *
    from SLiCAPhtml import *
    from SLiCAPplots import *
    from SLiCAPpythonMaxima import *
    from SLiCAPmatrices import *
    from SLiCAPprotos import *
    from SLiCAPmath import *
    # Initialize DEVICES and MODELS:
    SLiCAPprotos.initAll()
    from SLiCAPlex import *
    from SLiCAPini import *
        from SLiCAPconfig import *      # The only module that may be modified by the user.
        # other Python modules.
        import docutils.core
        import docutils.writers.html5_polyglot
        import numpy as np
        import sympy as sp
        from scipy.signal import residue
        import ply.lex as lex
        from shutil import copy2 as cp
        from time import time
        from datetime import datetime
        import re
        import subprocess
        from threading import Timer
        import os
        import getpass
        import matplotlib._pylab_helpers as plotHelp
        from matplotlib import pyplot as plt

# Create an instance of the project class.
"""
This initializes the project paths, creates the directory
structure, a main index HTML page and compiles the
system libraries."""
my_prj = SLiCAP.initProject( < projectName > )
    Prj = SLiCAP.SLiCAPproject( < projectName > )
    SLiCAP.ini.upatePaths( < ProjectPath > )
    SLiCAP.html.startHTML(projectName)
    SLiCAPyacc.makeLibraries()
        # Create a circuit object
        """
        library models and sub circuits are stored in
        LIB.models and in LIB.circuits, respectively."""
        LIB = SLiCAPprotos.circuit()
        # tokenize the input file
        LIB.lexer = SLiCAPlex.tokenize( < libFileName > )
        # Create a 'flattened' circuit from the tokens
        LIB = SLiCAPyacc.makeCircuit(LIB)

# Create an instance of the instruction class
my_instr = SLiCAPinstruction.instruction
# Check a netlist and define the circuit from it for the instruction
SLiCAPinstruction.instruction.setCircuit( < fileName > )
    SLiCAPyacc.checkCircuit( < fileName > )
        # Create a circuit object:
        cir = SLiCAPprotos.circuit()
        # Tokenize the input file
        cir.lexer = SLiCAPlex.tokenize(fileName)
        # Create a 'flattened' circuit from the tokens
        cir = SLiCAPyacc.makeCircuit(cir)
        # Update the circuit data required for execution of instructions.
        SLiCAPyacc.updateCircuitData(cir)
    SLiCAPinstruction.instruction.circuit = cir
```