

# Инструкция по использованию библиотекой OpenVSM v0.2

Библиотека была написана мною изначально для сугубо личных целей — собирать модели в ОС Linux. Так как использовался компилятор g++, то возникли проблемы с тем, что у VC++, которым собран Proteus, иные соглашения по вызовам нежели у g++. Среди всех способов обойти проблемы, был выбран способ, точно работоспособный на огромном количестве компиляторов — переписать код на C, поскольку расхождений тут быть не может.

В процессе работы стало быстро понятно, что отлаживать компилируемый код не слишком удобно. Было бы удобней использовать хотя бы на время отладки что-то вроде скриптов. Сказано — сделано. Был выбран язык [Lua](#), как крайне гибкий и что немаловажно — быстрый.

## Принцип работы

Все моделируемые устройства используют одну и ту же библиотеку, лишь скрипты должны быть уникальными. Разумеется, можно использовать разные версии библиотеки, но это лишь увеличивает занятое место на диске. При создании модели нужно лишь создать следующие параметры в текстовом скрипте модели:

```
{MODDLL="MY_DEVICE",HIDDEN STRING}
{LUA="LUA",STRING}
{MODDLL=openvsm.dll}
{LUA=my_script.lua}
```

В данном случае, Lua файл модели называется my\_script.lua.

Сам скрипт должен располагаться в директории, указанной в переменной окружения LUAVSM.

В самом общем случае, необходимо:

1. Библиотеку openvsm.dll положить в директорию с моделями Proteus, либо рядом с файлом DSN (Для Proteus 8+).
2. Создать директорию для скриптов, либо использовать любую имеющуюся
3. Создать переменную окружения с путём к этой директории (выполняется в cmd.exe):

```
setx LUAVSM "C:\script"
```

4. Создать файл скрипта и начать работу.

# Сборка библиотеки

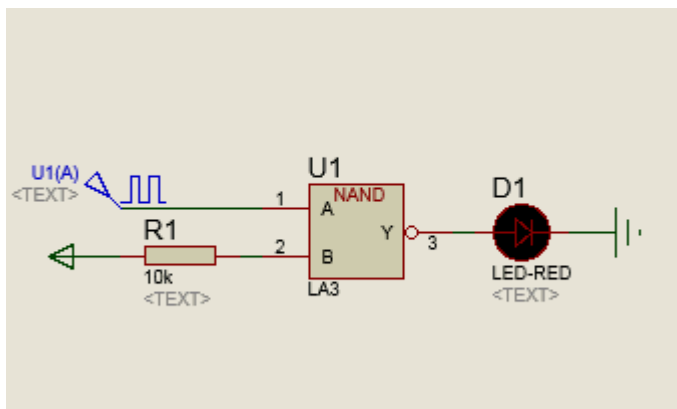
Сборка библиотеки нужна крайне редко. Либо если вы хотите исправить/улучшить код, добавить возможностей, если вы хотите написать новый модуль к движку. Ну или просто из любви к искусству.

## Linux

## Windows

## Моделирование на Lua

Создадим простейшую модель и на примере её разберем основу работы (Все примеры доступны в разделе Sample репозитория).



Текстовый скрипт для модели:

```
{*DEVICE}
NAME=LA3
{PREFIX=U}
{*PROPDEFS}
{PRIMITIVE="Primitive Type",HIDDEN STRING}
{MODDLL="NAND Gate",HIDDEN STRING}
{PACKAGE="PCB Package",HIDDEN PACKAGE}
{LUA="LUA",STRING}
{*INDEX}
{CAT=TTL 155 series}
{SUBCAT=Gates & Inverters}
```

```

{ITFMOD=}
{MFR=USSR}
{DESC=NAND}
{*COMPONENT}
{PRIMITIVE=DIGITAL}
{LUA=nand.lua}
{MODDLL=openvsm.dll}
{PACKAGE=DIL16}

```

И собственно скрипт, моделирующий вентиль И-НЕ

```
--NAND gate
```

```
SAFE_MODE=true
LOGIC_TYPE=TTL
```

```
device_pins =
{
  {is_digital=true, name = "A"},
  {is_digital=true, name = "B"},
  {is_digital=true, name = "Y"},
}

```

```
function device_simulate ()
  Y:set(1-(A:get() * B:get()))
end

```

Вот и всё, запускаем симуляцию

