

## ПРИЛОЖЕНИЕ НА СВОБОДНИЯ СОФТУЕР С ОТВОРЕН КОД ПРИ МОДЕЛИРАНЕТО НА ЕЛЕКТРИЧЕСКИ ТРАНСПОРТНИ СРЕДСТВА

**Никола Македонски**  
[n.makedonski@tu-varna.bg](mailto:n.makedonski@tu-varna.bg)

*Технически университет – Варна  
Варна 9010, ул. "Студентска" № 1  
РЕПУБЛИКА БЪЛГАРИЯ*

***Ключови думи:** моделиране, електрически транспортни средства, свободен софтуер, Modelica*

***Резюме:** Моделирането в различните области на човешкото познание играе основна роля за изучаването, управлението и подобряването на естествените и изкуствените системи. Система, може да бъде всичко, чиито свойства някой иска да изучи. В този доклад се разглежда приложението на свободния софтуер с отворен код при моделирането на различни инженерни системи и в частност електрическите транспортни средства, като алтернатива на частния софтуер със затворен код. Характерно за свободния софтуер с отворен код е взаимодействието му с друг такъв, като се формира цялостна екосистема, която дава много големи възможности за научно-изследователска и образователна работа. Моделите, показани в този доклад са написани на езика Modelica. Modelica е език за моделиране на кибер-физически системи, поддържащ акаузална връзка на компоненти, управлявани от математически уравнения. Той предоставя обектно-ориентирани конструкции, които улесняват повторното използване на модели и могат да се използват удобно за моделиране на сложни системи, съдържащи например механични, електрически, електронни, магнитни, хидравлични, термични, контролни, електрически или ориентирани към процеса подкомпоненти. Тъй като езика Modelica е просто език за програмиране, то е необходим и софтуер имплементиращ езика. Тук е показано използването на софтуера OpenModelica [1]. OpenModelica е среда за моделиране и симулация с отворен код, базирана на Modelica, предназначена за индустриална и академична употреба.*

### **ВЪВЕДЕНИЕ**

Определението за свободен софтуер е:

“Свободен софтуер” (Free software) означава софтуер, който зачита **свободата и общността на потребителите**. Грубо казано, това означава, че потребителите имат свободата да **използват, копират, разпространяват, изучават, променят и подобряват** софтуера. По този начин свободният софтуер е **въпрос на свобода**, а не на цена. — Free Software Foundation [2].

Творецът, в частност ученият, изследователят, се нуждае от свобода за да твори.

Следователно е необходимо да се използва свободният софтуер. Използвайки свободен софтуер, човек може да си направи работна среда, такава, каквато най-добре подхожда на неговия индивидуален работен процес. Инсталирайки си някоя от многото свободни операционни системи по проекта GNU/Linux, човек получава подходящата среда за формирането на индивидуална работна среда. Тук е момента да се отбележи, че както свободата изисква действие, така и използването на свободен софтуер изисква знание и желание да се учи през целия живот. Съвременните персонални компютри и програмното обезпечение отдавна надхвърлиха чисто изчислителните задачи. В наше време това са инструменти, които, освен всичко друго, позволяват „материализирането“ на мислите ни.

Голямо предимство на свободния софтуер е необходимостта и желанието за изучаване на работата с него. Това създава общност от потребители от цял свят. По този начин възникват нови идеи за функционирането на софтуера, които се имплементират, като по този начин софтуера се подобрява и развива. Създаването на общности от хора с близки интереси е нещо уникално за свободния софтуер.

## РАБОТНА СРЕДА

Работата на преподавателя във висше учебно заведение може да се раздели на:

- учебна дейност;
- научно-изследователска дейност;
- административна дейност.

Учебната дейност, извън работата със студентите, е свързана най-вече с текстообработка. Това включва разработка на учебници, учебни помагала, презентационни материали и др. Голяма част от научно-изследователската дейност също включва текстообработка - публикационна дейност, научни трудове и др. Административната дейност се състои почти изцяло от обработка на текстови документи. Както е известно, най-добрата система за текстообработка и в частност за писане на научни документи е [LaTeX](#). Това е свободен софтуер. Освен това, съществува и свободен офис пакет - [LibreOffice](#). Освен всичко друго, офис пакета позволява редактирането и съвместимостта на документи създадени от несвободен софтуер.

Научно-изследователската работа е творчески процес, при който използването на свободен софтуер е от ключово значение, поради уникалното му свойство да формира цялостна екосистема за работа. Много често при този процес се налага "ръчната" разработка на софтуер, който цели специална обработка на данни. В областта на програмирането, свободните операционни системи са най-доброто, което съществува. Изборът на програмни езици обхваща областта от всички някога съществували до експериментални такива. Скриптовият език [Python](#) с общо предназначение е много подходящ за писане на програми за специфична обработка. Това, което го прави подходящ е наличието на многобройни библиотеки за научно-изследователска работа, като числени методи, алгоритми, обработка на времеви редове - [NumPy](#), [SciPy](#), символна математика - [SymPy](#), машинно обучение - [Scikit-learn](#), [TensorFlow](#), [PyTorch](#), визуализация на данни - [Matplotlib](#), [Plotly](#) и много други различни библиотеки.

Широко разпространено в научните среди в България, вероятно поради непознаване на възможностите и инерцията в мисленето, е използването на несвободния софтуер Matlab/Simulink.

Свободен софтуер с такива възможности, като [GNU Octave](#) и някои други, са разгледани в [3] [4]. За GNU Octave е казано, че [3] "It runs on operating systems like MS Windows, Linux, Mac OS X and others. It uses MATLAB files, i.e. MATLAB's Control System Toolbox functions, but a disadvantage is that in Octave there is no graphic editor for building block diagrams such as Simulink in MATLAB. This software can be used, for

example, as an introduction to MATLAB, instead of MATLAB, and each student can use it without any problems with licenses." Пак в тези източници [3] [4], е разгледан свободният софтуер [Scilab/Xcos](#) като графичен редактор за изграждането на блокови диаграми и симулации и използването му вместо Simulink.

## OPENMODELICA

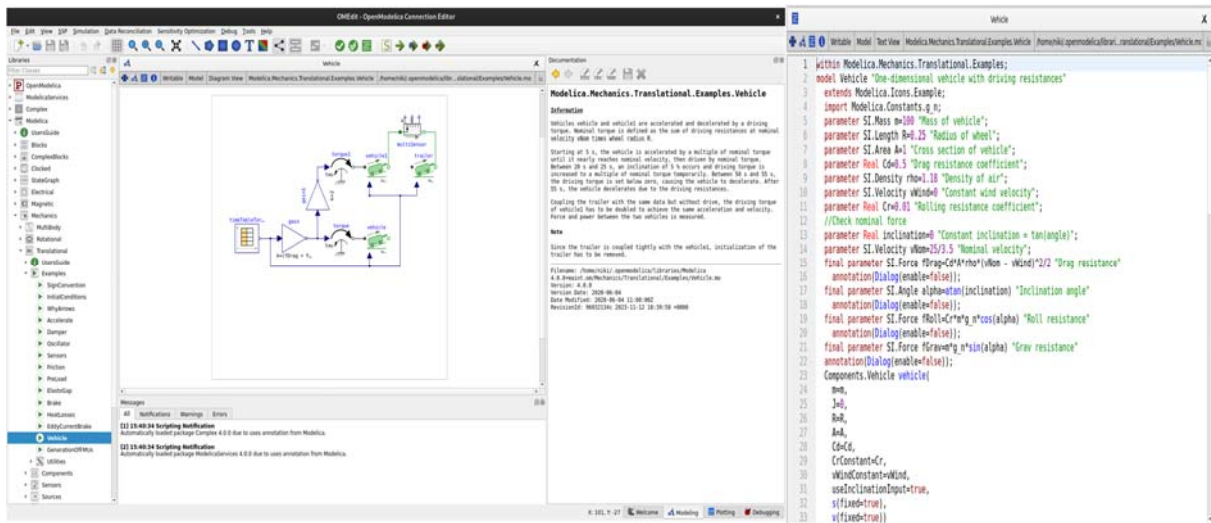
Едва в последните версии на Simulink е осъществено т.нар. физическо моделиране. От друга страна Modelica е език за моделиране на кибер-физически системи, поддържащ акаузална връзка на компоненти, управлявани от математически уравнения. Той предоставя обектно-ориентирани конструкции, които улесняват повторното използване на модели и могат да се използват удобно за моделиране на сложни системи, съдържащи например механични, електрически, електронни, магнитни, хидравлични, термични, контролни, електрически или ориентирани към процеса подкомпоненти. Тъй като езика Modelica е просто език за програмиране, то е необходим и софтуер имплементиращ езика. Тук е показано използването на софтуера OpenModelica [1].

OpenModelica е свободен софтуер, представляващ среда за моделиране и симулация, базирана на Modelica, предназначена за индустриална и академична употреба. Софтуерът е изграден от множество компоненти, като може би най-често използваният е OMEdit - OpenModelica Connection Editor е новият графичен потребителски интерфейс за редактиране на графични модели в OpenModelica. Той е реализиран чрез C++, използвайки библиотеката за графичен потребителски интерфейс Qt и поддържа стандартната библиотека Modelica, която е включена в OpenModelica.

OpenModelica имплементира стандартната библиотека Modelica, състояща се от следните главни библиотеки: **Analog** - аналогови електрически и електронни компоненти, като резистор, кондензатор, трансформатори, диоди, транзистори, преносни линии, ключове, източници и датчици; **Digital** - цифрови компоненти базирани на VHDL стандарта, като базови логически блокове; **Machines** - електрически асинхронни, синхронни и постоянно токови машини (двигатели и генератори), както и трифазни трансформатори; **FluxTubes** - базира се на концепцията за магнитни тръби. Използва се специално за моделиране на електромагнитни актуатори, неправилни форми и нелинейни магнитни характеристики на магнитопроводите, нелинейни сили, моделиране на разсейване. Съдържа таблични данни за магнитните характеристики на магнитопроводи от различни материали като стомана, електротехническа стомана, чисто желязо и магнитопроводи от различни сплави на желязото; **Translational** - механични преместващи се равнинно едномерни системи, като плъзгащи се маси, пружини и др.; **Rotational** - механични въртящи се едномерни системи, като инерция, зъбни и планетарни предавки, съединители, спирачки, лагери и др.; **MultiBody** - тримерни механични системи, състоящи се от връзки, тела, сили и измервателни елементи. Всеки компонент има анимация по подразбиране. Компонентите могат да се свързват помежду си произволно; **Fluid** - едномерен термо-флуиден поток в мрежа от съдове, тръби, флуидни машини, клапани и фитинги. Използват се различни физични среди, като несвиваеми и свиваеми флуиди, едно- или двуфазни среди и др.;

**Blocks** - входно/изходни блокове за моделиране на диаграми и логически мрежи - интегратори, ПИД регулатори, предавателни функции, линейни системи с модели на състоянието, дискретни предавателни функции, източници, таблици и др.; **StateGraph** - йерархични машини на състоянието. Modelica се използва като синхронен език на действието, като детерминистичното поведение е гарантирано.; **Math, Utilities** - съдържа функции работещи с вектори и матрици, като такива за решаване на линейни системи, собствени и сингулярни числа и др., а също и функции работещи с текстови

низове, потоци, файлове и др.



Фиг. 1. Пример за превозно средство от стандартната библиотека Modelica → Mechanics → Translational → Vehicle.

Както се вижда от този бегъл преглед, дори само стандартната библиотека на Modelica, може да служи както за обобщени модели на електрически транспортни средства, така и за детайлни модели на такива, както и за много подробното моделиране на отделните им възли, като задвижващите двигатели заедно с тяхното управление или управлението на електрическата батерия например.

На фиг. 1 е показан пример от стандартната библиотека на превозно средство. Отляво се вижда интерфейса на OMEdit. Може да се види стандартната библиотека и част от подбиблиотеките отляво, главният прозорец, в който потребителят изгражда модел по средата, а отдясно се вижда документация.

Отдясно се вижда синтаксиса на езика Modelica. Всеки модел всъщност представлява текстов файл съдържащ потребителски код на езика Modelica, който се анализира, оптимизира и компилира от компилатора на OpenModelica до код на езика C или C++, след което, чрез друг компилатор g++ се компилира до изпълним код. Това всъщност става много бързо и целият описан процес, плюс изпълнението на машинния код и изобразяването на различните променливи в модела, става, в повечето случаи, много по-бързо от аналогична симулация в Simulink.

Документацията на примера показан на фиг. 1, която е вградена в текстовият файл на самият модел, е следната. Превозните средства **vehicle** и **vehicle1** се ускоряват и забавят, чрез приложен въртящ момент. Номиналният въртящ момент е определен като сумата от съпротивлението при движение при номинална скорост  $v_{Nom}$  умножено по радиуса на колелата  $R$ . Стартирайки на петата секунда, превозното средство **vehicle** се ускорява от въртящ момент с големина няколко пъти номиналният въртящ момент докато почти достигне номиналната си скорост, след което задвижващият въртящ момент става равен на номиналният.

Между 20-тата и 25-тата секунда, превозното средство започва изкачване с наклон от 5% и задвижващия въртящ момент се увеличава до няколко пъти номиналният, за известно време. Между 50-тата и 55-тата секунда, задвижващият въртящ момент се задава да е отрицателен, което води до това, че превозното средство **vehicle**, забавя. След 55-тата секунда, превозното средство **vehicle** забавя поради съпротивлението при движение.

При купирането на ремаркетото **trailer** със същите данни, но без задвижване,

задвижващият въртящ момент на превозното средство **vehicle1** трябва да бъде удвоен за да се постигне същото ускорение и скорост. Силата и механичната мощност между превозното средство **vehicle1** и ремаркетото **trailer** се измерва.

### ЗАКЛЮЧЕНИЕ

В заключение може да се цитира отговора на въпроса „Каква е ролята на свободния софтуер за образованието?“ [5] „Свободата на софтуера играе основна роля в образованието. Образователните институции от всички нива трябва да използват и преподават безплатен софтуер, защото това е единственият софтуер, който им позволява да изпълнят основните си мисии: да разпространяват човешкото знание и да подготвят учениците да бъдат добри членове на своята общност. Изходният код и методите на свободния софтуер са част от човешкото познание. Напротив, несвободния софтуер е тайно, ограничено знание, което е обратното на мисията на образователните институции.

Свободният софтуер не е просто технически въпрос; това е етичен, социален и политически въпрос. Това е въпрос на човешки права, които потребителите на софтуер трябва да имат. Свободата и сътрудничеството са основни ценности на свободния софтуер. Системата GNU прилага тези ценности и принципа на споделяне, тъй като споделянето е добро и полезно за човешкия прогрес.“

### ЛИТЕРАТУРА:

- [1] P. Fritzon *et al.*, “The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development,” *Modeling, identification and control*, vol. 41, no. 4, pp. 241–295, 2020, doi: [10.4173/mic.2020.4.1](https://doi.org/10.4173/mic.2020.4.1).
- [2] Free Software Foundation, “What is free software?” Accessed: May 30, 2024. [Online]. Available: <https://www.gnu.org/philosophy/free-sw.html>
- [3] G. Janevska and M. Kostov, “Possibilities for open source software in mechatronic systems designbooktitleinternational scientific conference on information, communication and energy systems and technologies,” Sozopol, Bulgaria, 2018, pp. 343–346. Available: [https://www.researchgate.net/publication/326689388\\_Possibilities\\_for\\_Open\\_Source\\_Software\\_in\\_Mechatronic\\_Systems\\_Design](https://www.researchgate.net/publication/326689388_Possibilities_for_Open_Source_Software_in_Mechatronic_Systems_Design)
- [4] G. Janevska, “Примена на софтвер со отворен код за моделски базирано дизајнирање на мехатронички системи,” *Proceeding of papers, Faculty of Technical sciences - Bitola*, pp. 1–6, 2014, Available: [https://www.researchgate.net/publication/307151906\\_PRIMENA\\_NA\\_SOFTVER\\_SO\\_OTVOREN\\_KOD\\_ZA\\_MODELSKI\\_BAZIRANO\\_DIZAJNIRANE\\_NA\\_MEHATRONICKI\\_SISTEMI](https://www.researchgate.net/publication/307151906_PRIMENA_NA_SOFTVER_SO_OTVOREN_KOD_ZA_MODELSKI_BAZIRANO_DIZAJNIRANE_NA_MEHATRONICKI_SISTEMI)
- [5] Free Software Foundation, “How does free software relate to education?” Accessed: May 31, 2024. [Online]. Available: <https://www.gnu.org/education/education.html>

# APPLICATION OF FREE AND OPEN SOURCE SOFTWARE IN THE MODELING OF ELECTRIC VEHICLES.

**Nikola Makedonski**  
[n.makedonski@tu-varna.bg](mailto:n.makedonski@tu-varna.bg)

*Technical University of Varna*  
*Varna 9010, "Studentska" street No. 1*  
**REPUBLIC OF BULGARIA**

**Key words:** modeling, electrical vehicle, free software, Modelica

**Abstract:** Modeling in various fields of human knowledge plays a fundamental role in the study, management and improvement of natural and artificial systems. A system can be anything whose properties one wants to study. This report examines the application of open source free software in the modeling of various engineering systems, and in particular electric vehicles, as an alternative to proprietary closed source software. Characteristic of free and open source software is its interaction with other software, forming a complete ecosystem that provides great opportunities for scientific research and educational work. The models shown in this report are written in the Modelica language. Modelica is a modeling language for cyber-physical systems supporting acausal coupling of components governed by mathematical equations. It provides object-oriented constructs that facilitate model reuse and can be conveniently used to model complex systems containing, for example, mechanical, electrical, electronic, magnetic, hydraulic, thermal, control, electrical, or process-oriented subcomponents. Since the Modelica language is simply a programming language, software implementing the language is also needed. The use of OpenModelica software [1] is shown here. OpenModelica is an open source modeling and simulation environment based on Modelica designed for industrial and academic use.