

# Engee

Поддержка внешнего оборудования  
и программного обеспечения

Алексей Евсеев

инженер по встраиваемым системам и  
полунатурному моделированию



# Структура вебинара



## Начало работы, оборудование

- 1 Общие сведения, начало работы
- 2 COM-порт
- 3 Thrustmaster HOTAS
- 4 Logitech G29
- 5 CAN
- 6 Modbus
- 7 VISA
- 8 КПМ РИТМ

## Программное обеспечение

- 9 UDP
- 10 ПрагмаСофтСтудия
- 11 Unigine
- 12 T-Flex DOCs
- 13 Универсальность применения механизма

# Общие сведения



## СЦЕНАРИИ ПРИМЕНЕНИЯ

Обмен данными с оборудованием и внешними приложениями в процессе симуляции

Чтение/запись файлов

Автоматизированный сбор данных измерений

Управление устройствами

Конфигурирование устройств

Визуализация в сторонних приложениях

Косимуляция

! Внешнее оборудование, интерфейсы и протоколы подключены и работают на компьютере пользователя, но доступны непосредственно в Enggee.



Архитектура взаимодействия Enggee с низкоуровневыми интерфейсами

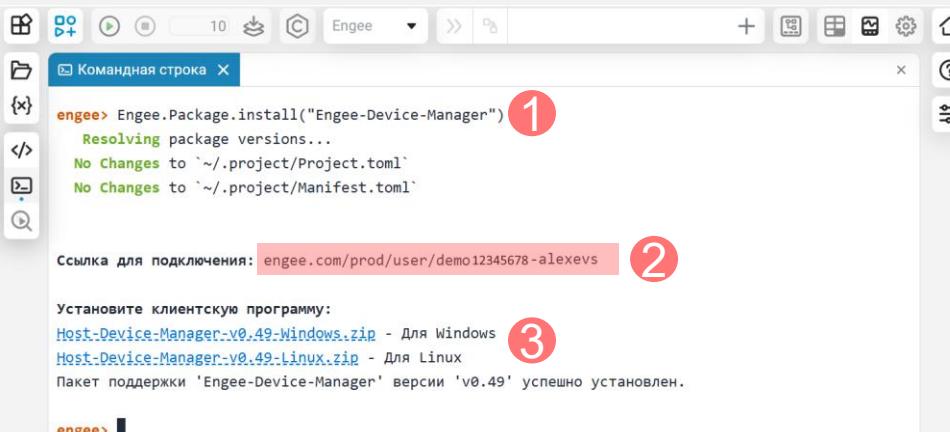


Развитие этапа разработки системной модели в парадигме МОП: источники и приёмники данных для системной модели  
- не инструменты Enggee, а внешние целевые среды.

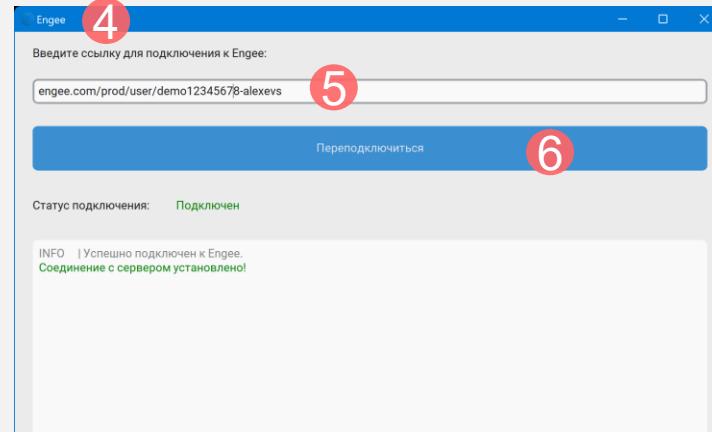
# Начало работы



engee> engee.package.install("Engee-Device-Manager")



Командная строка Engee



Клиентская программа

## ШАГИ ПОДКЛЮЧЕНИЯ:

1. Выполнить команду
2. Скопировать ссылку для подключения
3. Скачать клиентскую программу
4. Распаковать архив, запустить клиентскую программу
5. Вставить скопированную ссылку
6. Установить подключение

# Начало работы



Библиотека X

Ведите название блока

Оборудование

CAN

- CAN Настройка
- CAN Переход
- CAN Установка
- CAN RX
- CAN TX
- CAN Getstate

COM

- Несколькопорт.
- Несколькопорт.
- Несколькопорт.
- COM Setup
- COM TX
- COM RX

UDP

- UDP ПЕРЕДАЧА IP: 127.0.0.1 Порт: 1234 Порт: 1234
- UDP ПРИЕМКА IP: 0.0.0.0 Порт: 1234 Порт: 1234
- UDP TX
- UDP RX

Joystick

- Thrustmaster HOTAS Joystick
- Thrustmaster HOTAS Throttle

Wheel

- Logitech G29 Wheel

Библиотека X

Ведите название блока

Modbus

- Modbus TCP Setup
- Modbus Write Registers

UNIGINE

- UNIGINE TX

Pragmasoft

- PRAGMASOFT TX

UniversalMechanism

- UM Cosimulation

Обратная связь X

Тип обращения \*

Написать в техподдержку

Тема \*

Нужен новый блок для поддержки протокола Profibus DP

Теги \*

Выберите компонент Engee

Описание \*

Расскажите о своем опыте использования Engee или задайте вопрос команде технической поддержки

0/2000

Прикрепить файлы

Скриншоты, видеозаписи, скрипты или модели, которые помогут спроще обращение быстрее.

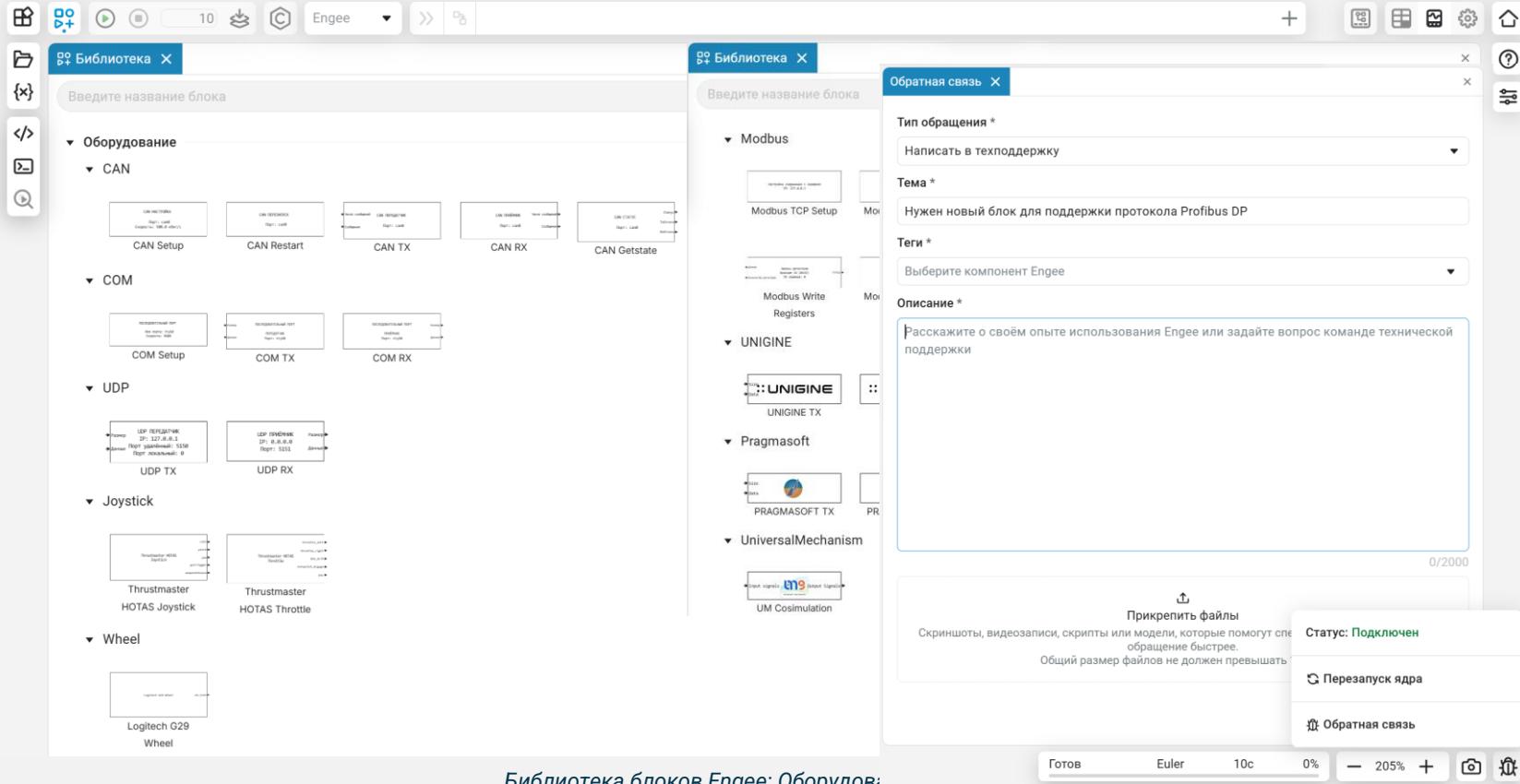
Общий размер файлов не должен превышать

Статус: Подключен

Перезапуск ядра

Обратная связь

Готов Euler 10с 0% - 205% +



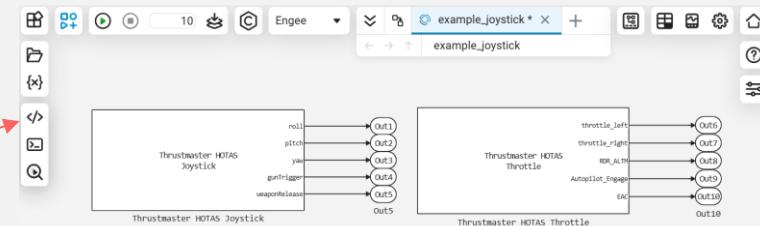
# Начало работы



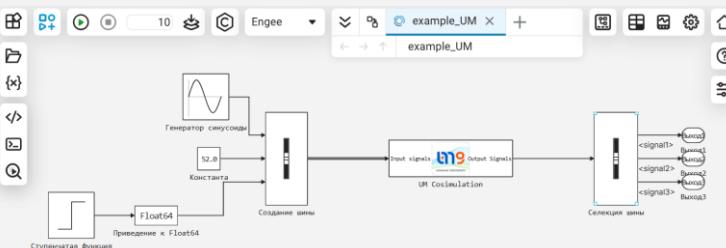
Файлы

Имя	Git	Размер	Тип	Дата
Engee-Device-Manager-v0.39-demos	--		dir	16.05.25 09:50
Engee-Device-Manager-v0.41-demos	--		dir	21.05.25 17:44
Engee-Device-Manager-v0.42-demos	--		dir	22.05.25 11:41
Engee-Device-Manager-v0.43-demos	--		dir	26.05.25 15:41
Engee-Device-Manager-v0.44-demos	--		dir	28.05.25 12:41
Engee-Device-Manager-v0.45-demos	--		dir	29.05.25 12:03
Engee-Device-Manager-v0.46-demos	--		dir	04.06.25 17:38
Engee-Device-Manager-v0.48-demos	--		dir	05.06.25 16:28
Engee-Device-Manager-v0.49-demos	--		dir	13.06.25 14:22
example_can.engee		18 КБ	engee	13.06.25 14:22
example_com.engee		6 КБ	engee	13.06.25 14:22
example_joystick.engee		6 КБ	engee	13.06.25 14:22
example_modbus_rtu.engee		9 КБ	engee	13.06.25 14:22
example_modbus_tcp.engee		8 КБ	engee	13.06.25 14:22
example_pragmasoft.engee		6 КБ	engee	13.06.25 14:22
example_ritm_ext.ngscript		3 КБ	ngscript	13.06.25 14:22
example_ritm_extmodel.engee		7 КБ	engee	13.06.25 14:22
example_ritm.ngscript		25 КБ	ngscript	13.06.25 14:22
example_udp.engee		39 КБ	engee	13.06.25 14:22
example_UM.engee		6 КБ	engee	13.06.25 14:22
example_unigine.engee		6 КБ	engee	13.06.25 14:22
example_visa.ngscript		28 КБ	ngscript	13.06.25 14:22
example_wheel.engee		11 КБ	engee	13.06.25 14:22

Файлы Engee: Примеры работы с внешним оборудованием и ПО

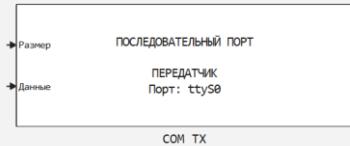
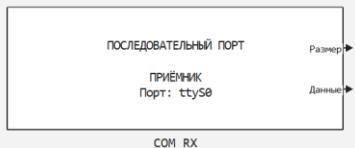
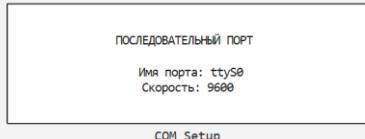


Тестовые модели из примеров



# СОМ-порт

Последовательный порт – интерфейс стандарта RS-232



Настройки: COM RX

Настройки X

Mask Информация Аннотация

Тип: EngeeFunction

Посмотреть под маску

▼ Main group

Порт: `"ttyS0"`

Размер буфера для приема: 8

Шаг расчёта, с: -1

Настройки: COM Setup

Настройки X

Mask Информация Аннотация

Тип: EngeeFunction

Посмотреть под маску

▼ Main group

Имя порта: `"ttyS0"`

Скорость передачи: 9600

Биты данных: 8

Стоповые биты: Один

Бит четности: Odd

Настройки: COM TX

Настройки X

Mask Информация Аннотация

Тип: EngeeFunction

Посмотреть под маску

▼ Main group

Порт: `"ttyS0"`

Размер буфера для передачи: 8

Шаг расчёта, с: -1

# СОМ-порт



## Содержимое блоков СОМ RX, СОМ TX

Настройки COM RX

EngeeFunctionCode

Common code

```
1 package_dir = "/internal_persistent_vol/support_packages/locations/Engee-Device-Manager/EngeeDeviceManager.j1"
2 include("${package_dir}/src/EngeeDeviceManager.j1")
3
4 using .EngeeDeviceManager
5 using .EngeeDeviceManager.Devices.COM
6
7
8 mutable struct Block <: AbstractCausalComponent
9   com::COM.Com
10  function Block()
11    com = COM.Com()
12    obj = new(com)
13    return obj
14  end
15 end
16
17
18
19 function (receiver::Block)(t::Real)
20   buffer_data = COM.receive(receiver.com, port, buf_size)
21
22  if buffer_data === nothing
23    return 0, zeros(UInt8, buf_size)
24  elseif length(buffer_data) == 0
25    return 0, zeros(UInt8, buf_size)
26  end
27
28  data_len = length(buffer_data)
29
30  return data_len, resize!(buffer_data, buf_size)
31 end
32
33 function terminate!(receiver::Block)
```

Настройки COM TX

EngeeFunctionCode

Common code

```
1 const Dimensions = NTuple{N, Int} where N
2 const SampleTime = NamedTuple{(:period, :offset, :mode), Tuple{Rational{Int64}, Rational{Int64}, Symbol}}
3
4
5
6
7
8 mutable struct Block <: AbstractCausalComponent
9   com::COM.Com
10  function Block()
11    com = COM.Com()
12    obj = new(com)
13    return obj
14  end
15 end
16
17
18
19 function (sender::Block)(t::Real, size, x)
20  if size == 0
21    return nothing
22  end
23
24  COM.transmit(sender.com, port, resize!(x, size))
25  return nothing
26 end
27
28 function terminate!(sender::Block)
29   COM.close(sender.com, port)
30   return nothing
31 end
```

# СОМ-порт



## Примеры из сообщества Engee

**COM SEND**



15 мая 2025 • alexevs • 19 • 0

**Внешнее оборудование - СОМ: передача сигнала в Arduino**

Начало работы с пакетом поддержки внешнего оборудования в Engee. Отправляем периодический сигнал из Engee в Arduino по СОМ, включая и выключая светодиод.

### Работа с внешним оборудованием в Engee

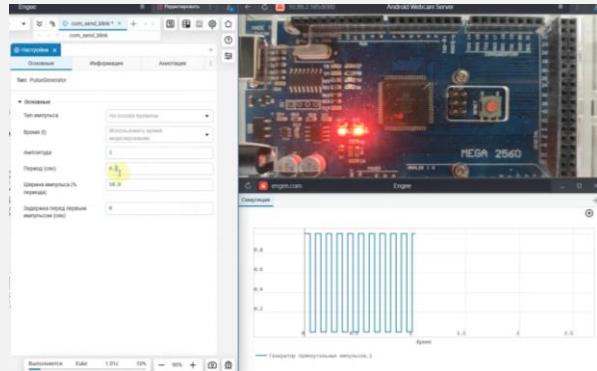
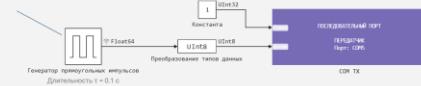
#### Отправка данных по СОМ: Arduino Blink

##### Шаги подготовки к выполнению модели:

1. Установить пакет поддержки внешнего оборудования: [здесь](#) (распаковать `Intall1117_Engee-Device-Handler`)
2. Установить подключение к СОМ, при необходимости сменить имена порта в настройках блоков СОМ
3. Запустить в устройстве программу обработки данных, передаваемых из Engee на СОМ



Значение можно менять "на лету", сохранив при этом их типы данных



**COM READ**



26 мая 2025 • alexevs • 12 • 0

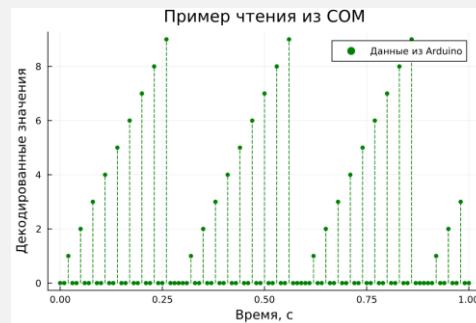
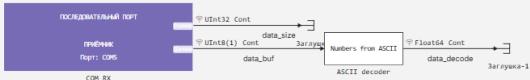
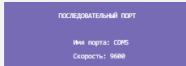
**Внешнее оборудование - СОМ: получение сигналов из Arduino**

Продолжаем работу с пакетом поддержки внешнего оборудования в Engee. Отправляем закодированные по ASCII сообщения из Arduino по СОМ и декодируем их в Engee.

### Работа с внешним оборудованием в Engee

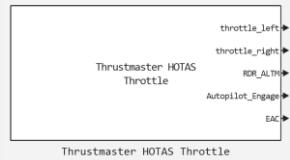
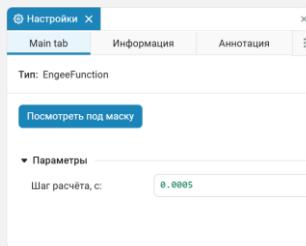
#### Получение данных по СОМ: Arduino

##### Справка



# Thrustmaster HOTAS

Игровые контроллеры управления самолётом, устройства HID (Human Interface Devices), подключаемые по USB



## Работа с внешним оборудованием в Engee

Thrustmaster HOTAS: стрельба по мишенями

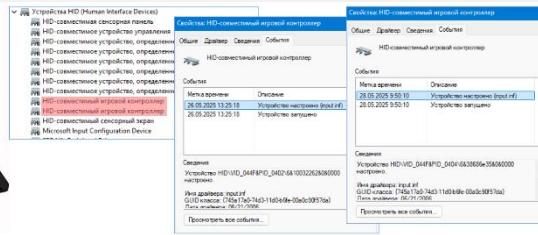
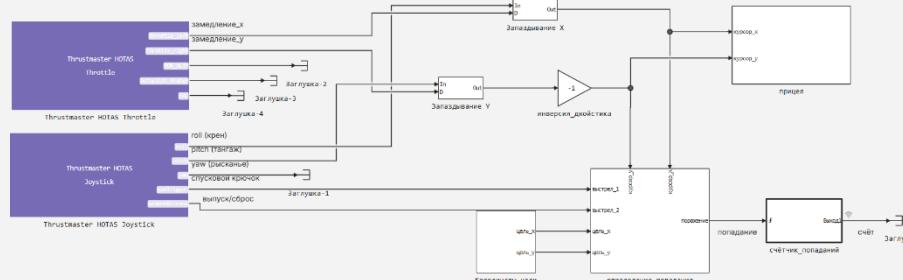
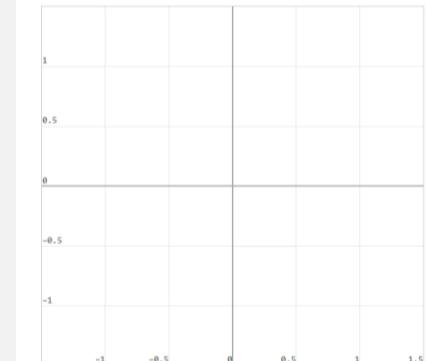


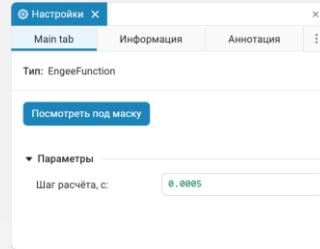
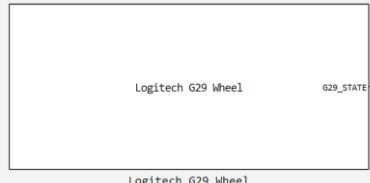
Таблица	
Сигнал	Значение
счёт	0.00



# Logitech G29



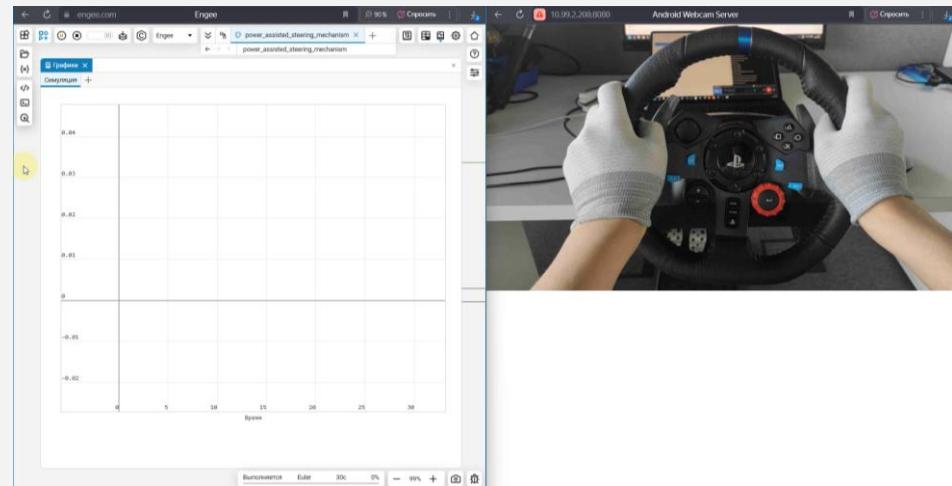
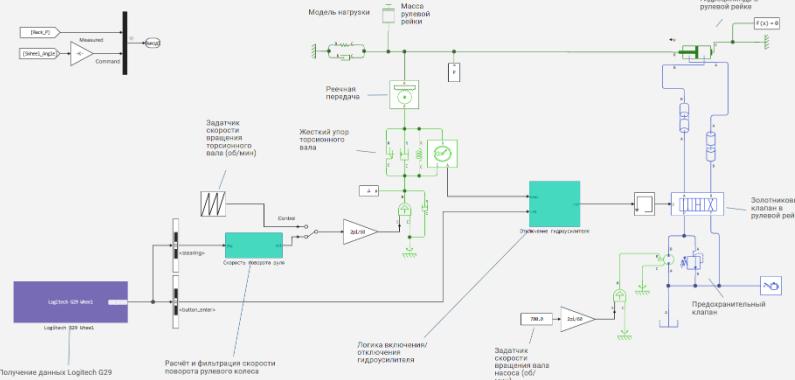
Игровые контроллеры управления гоночным автомобилем, устройства HID (Human Interface Devices), подключаемые по USB



Работа с внешним оборудованием в Enggee

Logitech G29: Рулевое управление с гидроусилителем

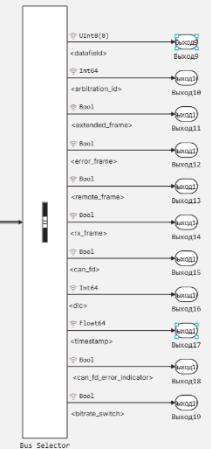
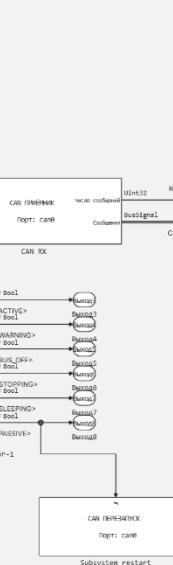
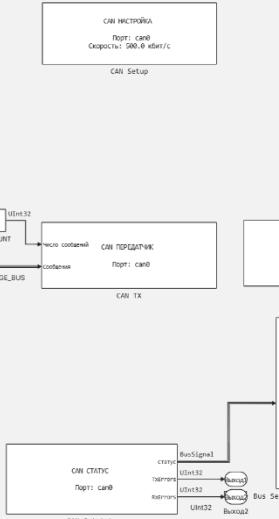
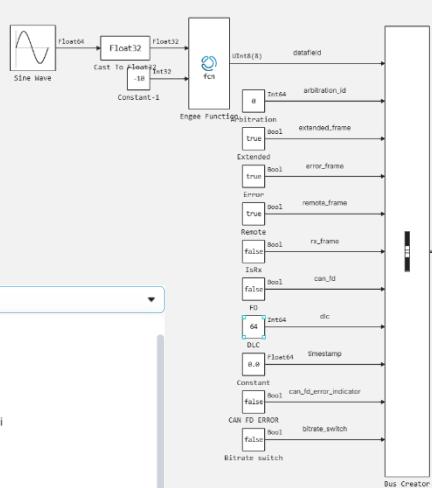
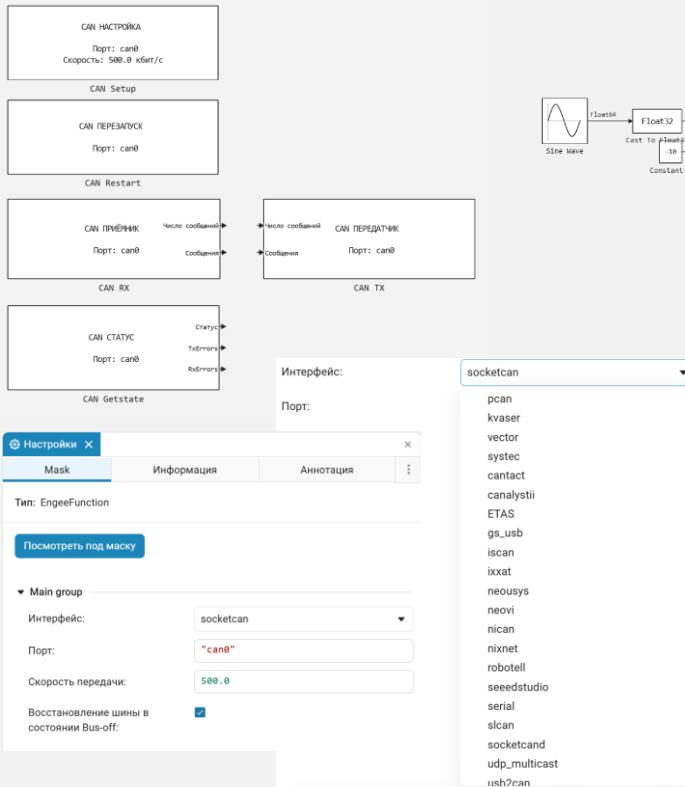
Основано на примере: [Модель гидроусилителя](#)



Получение данных Logitech G29



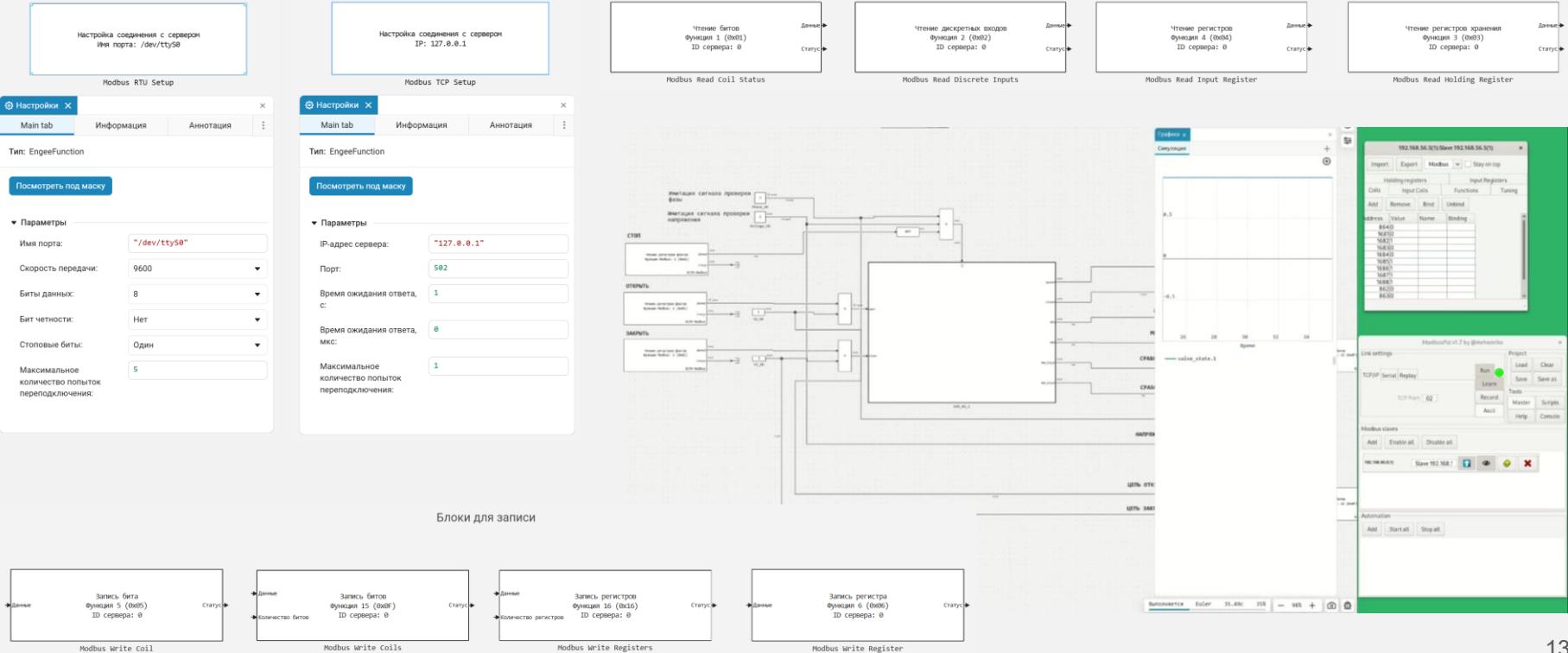
Промышленный сетевой стандарт последовательной пакетной шины. Наиболее востребован в автомобильной промышленности



# Modbus



## Промышленный протокол передачи данных по архитектуре master-slave.





## Virtual Instrument Software Architecture – спецификация, унифицирующая взаимодействие с измерительными приборами



### Методы VISA

- `EngeeDeviceManager.Devices.VISA.close_instrument`
- `EngeeDeviceManager.Devices.VISA.close_session`
- `EngeeDeviceManager.Devices.VISA.create_instrument`
- `EngeeDeviceManager.Devices.VISA.create_session`
- `EngeeDeviceManager.Devices.VISA.flush`
- `EngeeDeviceManager.Devices.VISA.get_ports`
- `EngeeDeviceManager.Devices.VISA.read`
- `EngeeDeviceManager.Devices.VISA.readbinblock`
- `EngeeDeviceManager.Devices.VISA.visatrigger`
- `EngeeDeviceManager.Devices.VISA.write`
- `EngeeDeviceManager.Devices.VISA.writebinblock`
- `EngeeDeviceManager.Devices.VISA.writeread`

### Пример конфигурирования устройства VISA при помощи SCPI-команд

```
using Main.EnggeeDeviceManager.Devices.VISA
```

```
visa = VISA.Visa()
```

```
VISA.create_session(visa, "@py")
```

```
port = last(VISA.get_ports(visa))
```

```
VISA.create_instrument(visa, port)
```

```
VISA.write(visa, port, ":CHAN1:BASE:WAV RAMP");
```

```
VISA.writeread(visa, port, ":CHAN1:BASE:WAV?", 1.2)
```



## Пример переключения форм генерируемых сигналов

```
FORMS = ["SIN", "SQU", "PULS", "RAMP", "ARB", "NOIS", "DC"];

for form in FORMS
    VISA.write(visa, port, ":CHAN1:BASE:WAV $form");
    sleep(1);
end

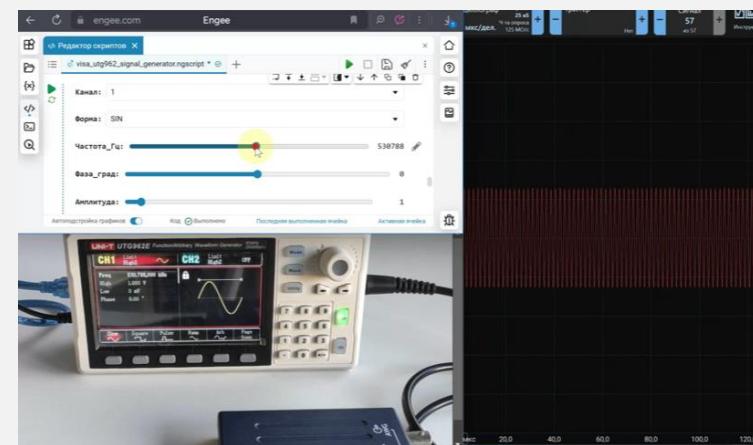
VISA.write(visa, port, "*RST");
```



## Пример настройки генерируемого сигнала

```
message = [
    ":CHANnel1$Канал:MODE Continue",
    ":CHANnel1$Канал:BASE:WAVE $Форма",
    ":CHANnel1$Канал:BASE:FREquency $Частота_Гц",
    ":CHANnel1$Канал:BASE:HIGH $Амплитуда",
    ":CHANnel1$Канал:BASE:LOW 0",
    ":CHANnel1$Канал:BASE:PHAsе $Фаза_град",
    ":CHANnel1$Канал:OUTPut ON"
]
```

```
UTG962_write(message)
```





## Комплекс полуцелочисленного моделирования (КПМ) РИТМ – вычислительная машина с ОС реального времени

### Управление РИТМом

#### Подключение модуля для взаимодействия с РИТМом

После установки и запуска EngeeDeviceManager достаточно выполнить:

```
1  using Main.EnggeeDeviceManager.Targets
2  using Main.EnggeeDeviceManager.Targets.RITM_API
```

### Создание объекта Ritm

Создадим объект РИТМ:

```
1  ritm = Targets.RITM.Ritm()
2  Targets.RITM_API.setUrl(ritm, "http://192.168.56.3:8000/")
```

1. Проверяем доступность машины РИТМ:

```
1  RITM_API.isConnected(ritm)
```



true

2. Собираем на РИТМе открытую на холсте standalone-модель:

```
1  model = Engee.engee.gcm()
2  Targets.upload_model(ritm, model)
3  Targets.generate_executable_code(ritm, model, False)
4  Targets.compile_model(ritm, model)
```



TargetResponse("Model compiled successfully", nothing)

3. Запускаем модель из Engee на машине РИТМ в режиме standalone:

```
1  Targets.start_model(ritm, model)
```



```
Targets.stop_model(ritm)
```

```
RITM_API.listFiles(ritm, "/home/ritm/")
```

```
RITM_API.readFile(ritm, "install_manifest.txt";
```

```
path="/home/ritm/build/newmodel_1/build/")
```

```
RITM_API.getFile(ritm, "install_manifest.txt";
```

```
from="/home/ritm/build/newmodel_1/build/", to="")
```

```
RITM_API.memInfo(ritm, "newmodel_1")
```

```
RITM_API.getLog(ritm, 10)
```

```
RITM_API.getScreenshot(ritm, "example.png")
```

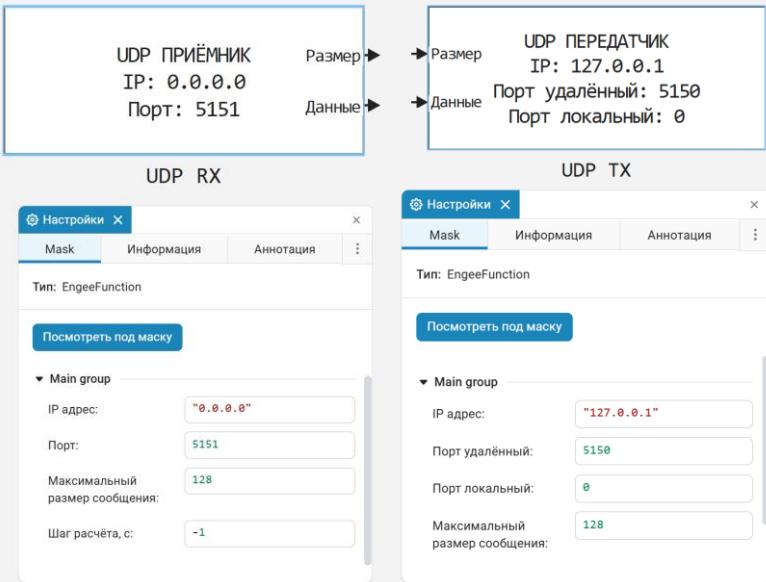
```
RITM_API.isRunning(ritm, "newmodel_1")
```

```
RITM_API.getData(ritm, "newmodel_1", "/user/", true)
```

UDP



## User Datagram Protocol – протокол пользовательских датаграмм.



## Работа с внешним оборудованием в Engee

## UDP: эхо-сервер в socat

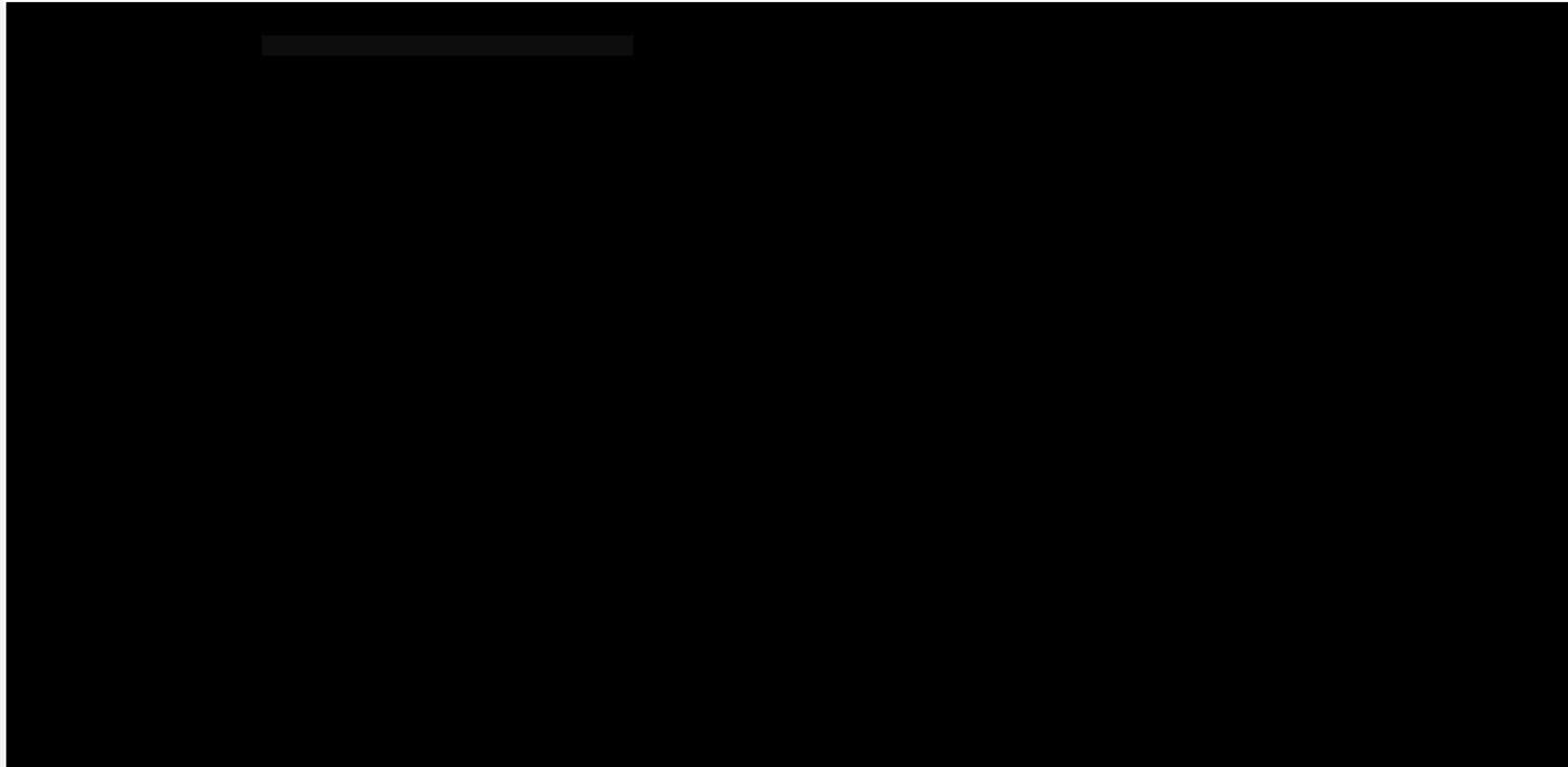
Для корректной работы демонстрационной модели нужен эхо-сервер, который будет получать пакеты с порта 5150 и отправлять на порт 5151. Сделать это можно следующим образом:

**Linux** socat UDP-RECV:5150 UDP-SEND:127.0.0.1:5151  
**Windows** (см. [ссылку](#))

The screenshot displays the Engee software interface. On the left, a 'Графики' (Graphs) panel shows a simulation graph with two curves: 'Engee Function.1.1' (blue) and 'Engee Function.2.1' (orange), plotted against 'Время' (Time) from 3 to 7. The graph shows a red curve (Engee Function.2.1) that dips below the blue curve (Engee Function.1.1) around time 4.5. On the right, a terminal window shows the command 'curl -X POST http://engee.com:8080/api/function'. Below the terminal, a message says 'Соединение с сервером установлено!' (Connection to the server is established!). The bottom part of the interface shows a 'Network' tab with a table of network traffic captures, including columns for No., Time, Source, Destination, Protocol, Length, and Info.



Пример косимуляции: Отправка данных из модели Engee в модель Simulink



# ПрагмаСофт Студия



САПР индикаторов и человека-машинных интерфейсов различных систем управления,  
в т.ч. авионики



PRAGMASOFT RX



PRAGMASOFT TX

Настройки ×

Main tab Информация Аннотация :

Тип: SubSystem

[Посмотреть под маску](#)

**Main group**

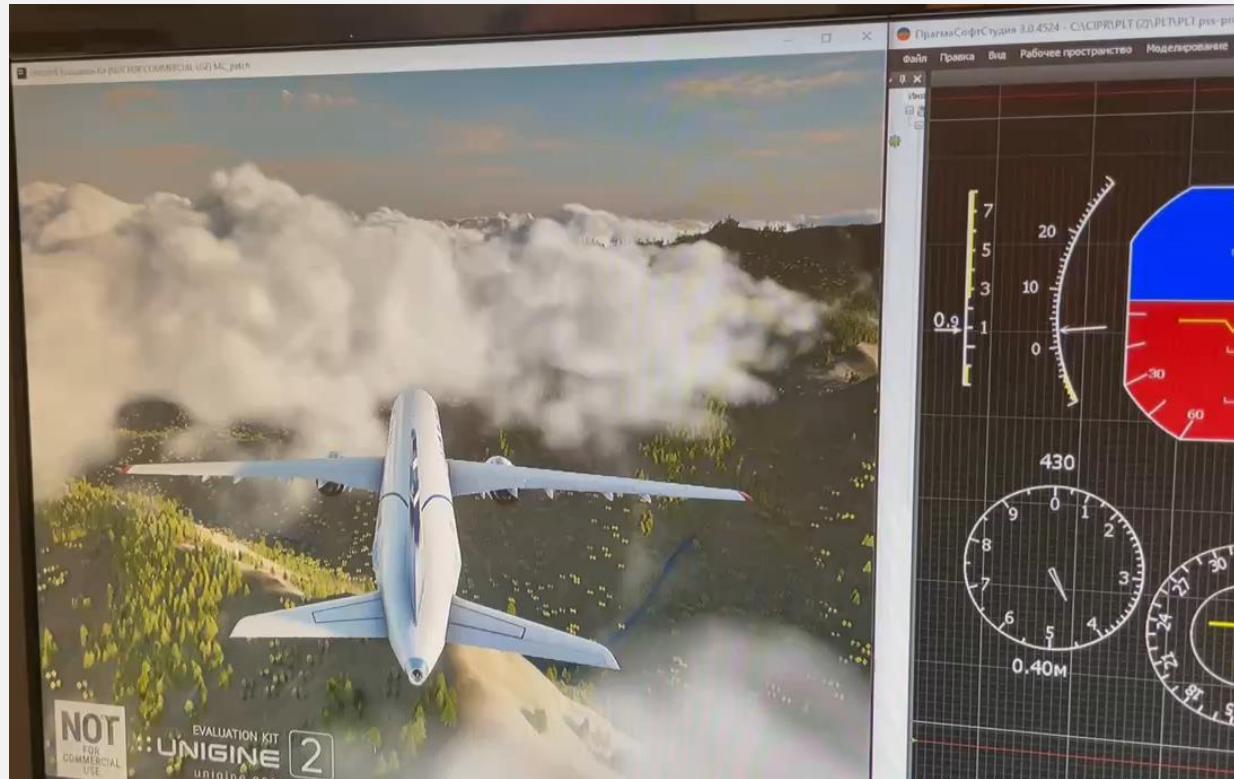
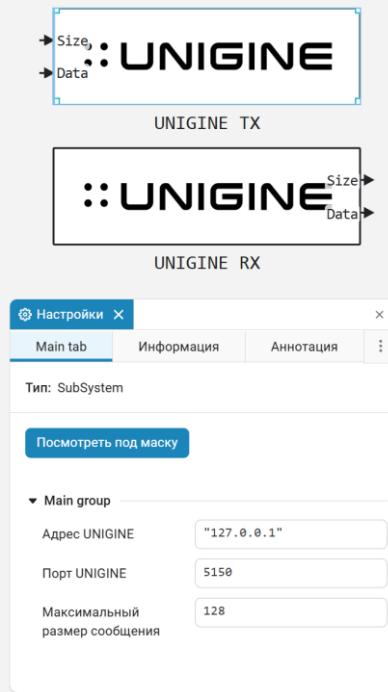
Адрес ПрагмаСофт:

Порт ПрагмаСофт:

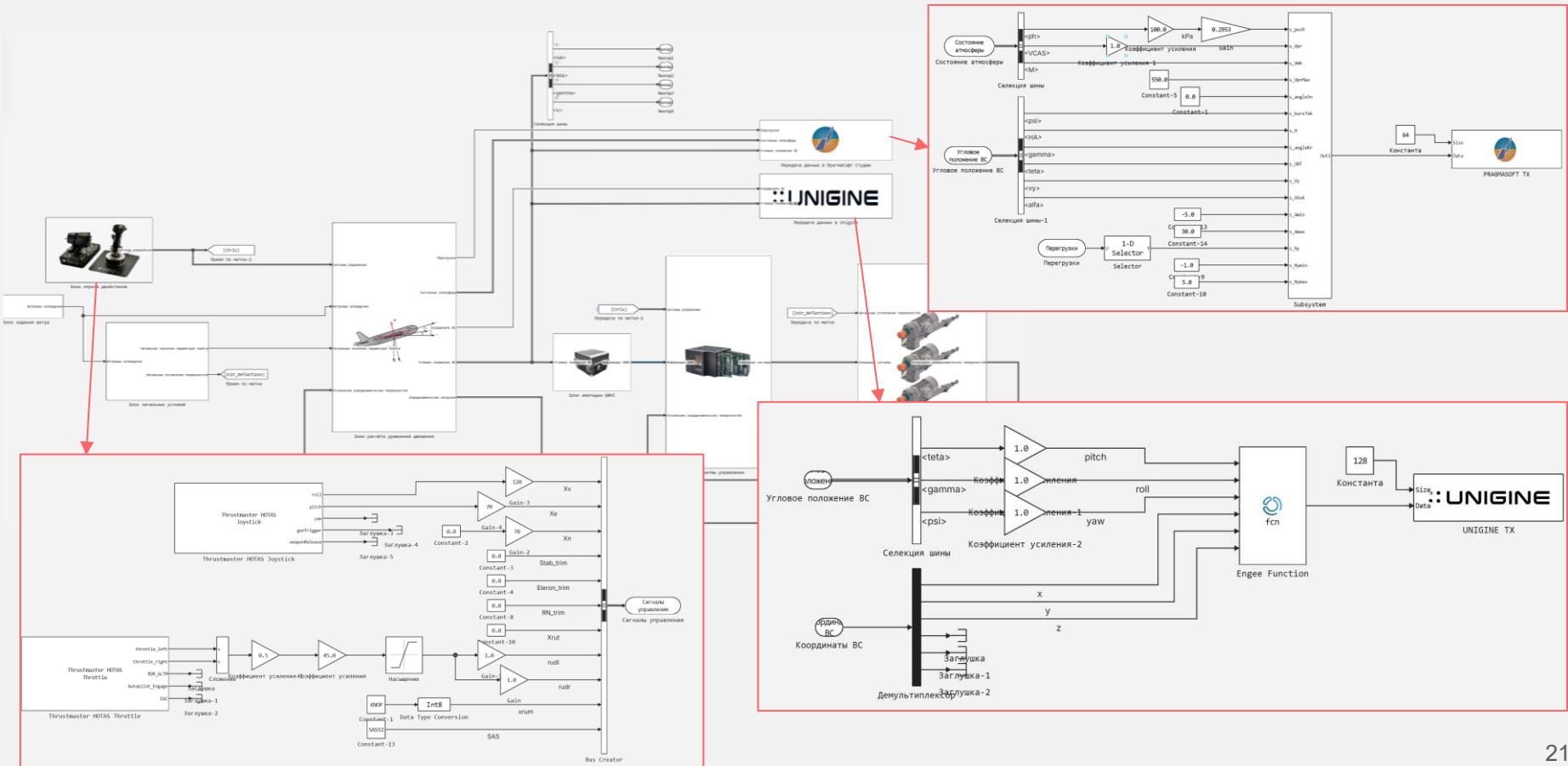
Максимальный размер сообщения:



## 3D-визуализация в реальном времени для симуляторов и тренажеров



# Системы управления МС-21, SSJ-100



# T-Flex DOCs



Комплексное управление инженерными данными предприятия, а также решение задач конструкторско-технологического и организационно-распорядительного документооборота

The screenshot shows the T-Flex DOCs software interface running on a Windows 10 system. The main window is titled "Windows 10 [Running] - Oracle VirtualBox".

The left sidebar displays a navigation tree with the following structure:

- localhost - T-FLEX DOCs Инструменты разработки
  - Документы
  - Импорт 17.04.2025 11:57:02
    - Расчёты
      - Геометрические данные
      - Данные для валидации
      - Инженерные расчёты
      - Исходные данные
      - Результаты расчёта
    - Классификаторы
    - Материалы
    - Системные справочники
    - Справочники CRM
    - Справочники PDM
    - Технологические справочники
    - Управление проектами
    - Управление требованиями
    - Файлы
    - Электронные компоненты

The main workspace shows a table titled "Инженерные расчёты" (Engineering Calculations) with the following data:

Назначение	Обозначение	Актуальная редакция
new_calc	default	<input checked="" type="checkbox"/>
new_calc	default	<input checked="" type="checkbox"/>

On the right, there is a code editor window titled "tflexdocs.ngscript" containing the following script:

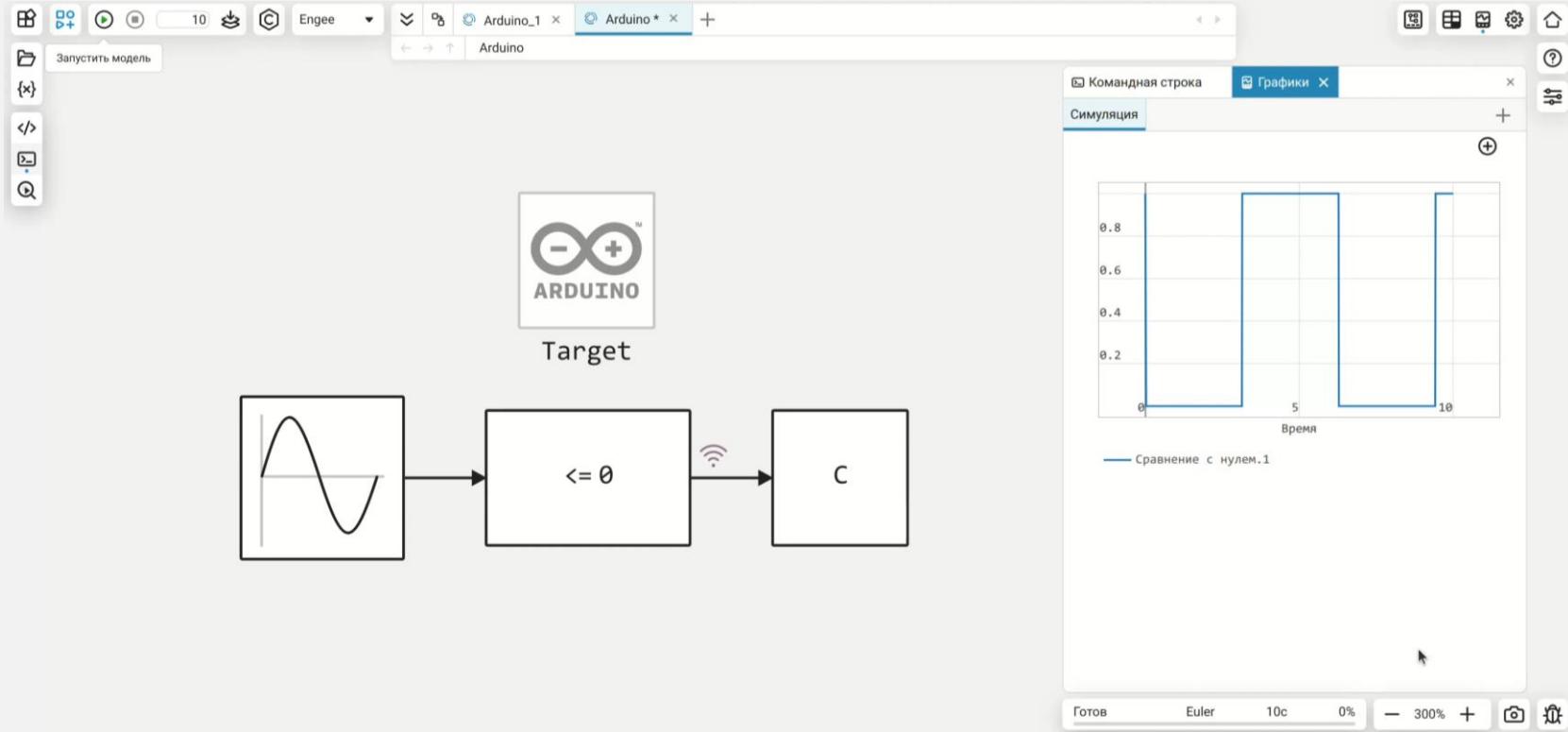
```
правим файлы в DOC's
DOCS.upload_files(tdocs, files, "new_calc")
```

Below the code editor, a status message reads: "upload files to TFlex DOCs".

# Универсальность применения механизма



Управление средствами сборки и загрузки кода, сгенерированного из модели Engree, в Arduino.



The screenshot shows the Engree software interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar shows "Engree" and "Arduino\_1" with an "Arduino" tab selected. On the left, there is a sidebar with icons for file operations and a "Запустить модель" (Run model) button. The main workspace contains a block diagram. On the right, there is a simulation window with a graph titled "Графики" (Graphs) and "Симуляция" (Simulation). The graph plots a signal over time, showing a square wave that is high for the first 5 units of time and low for the next 5 units, repeating this pattern. The x-axis is labeled "Время" (Time) and ranges from 0 to 10. The y-axis ranges from 0 to 0.8. A legend at the bottom of the graph says "Сравнение с нулем.1" (Comparison with zero.1). The Engree logo is visible in the center of the workspace.

Target

```
graph LR; A((Wavy Line)) --> B["<= 0"]; B --> C["C"]
```

Графики

Симуляция

Время

Сравнение с нулем.1

Готов Euler 10c 0% - 300% +

**Telegram-канал  
«Engee на старт»**

