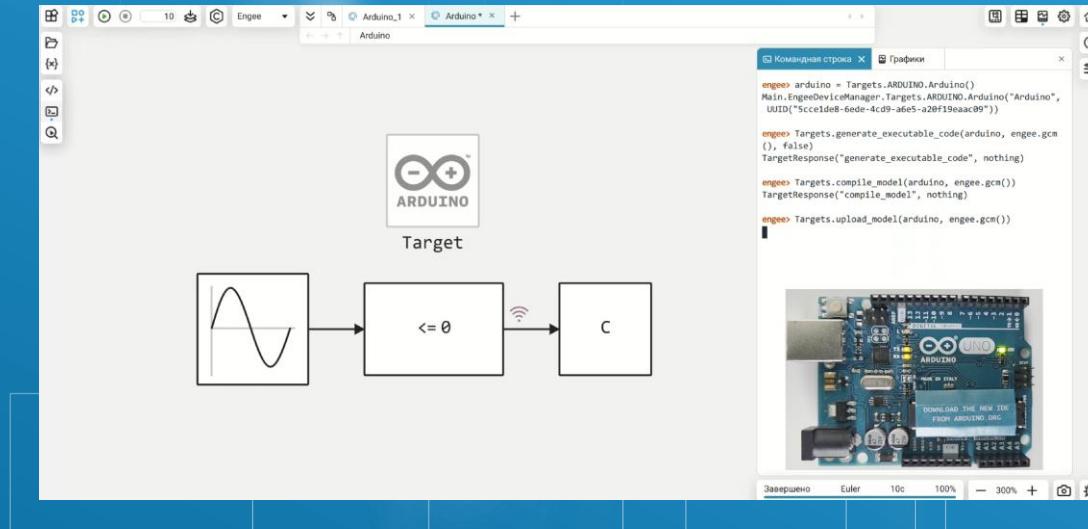




ЕНГЕЕ: Пакет поддержки встраиваемых систем

27 ноября 2025



ктн, инженер
Алексей Евсеев



План вебинара



1. Моделирование и встраиваемые системы



2. Таргетирование в Engee



3. Инструменты пакета поддержки



4. Таргетирование в примерах



5. Открытый механизм поддержки пользовательских устройств



6. Планы, перспективы



1. Моделирование и встраиваемые системы



Преимущества модельно-ориентированного проектирования (МОП):

- тестирование на каждом этапе,
- автоматизация тестирования,
- единый «источник правды» проекта - модель,
- автоматизация переходов между этапами.



Engee – российская платформа для разработки сложных

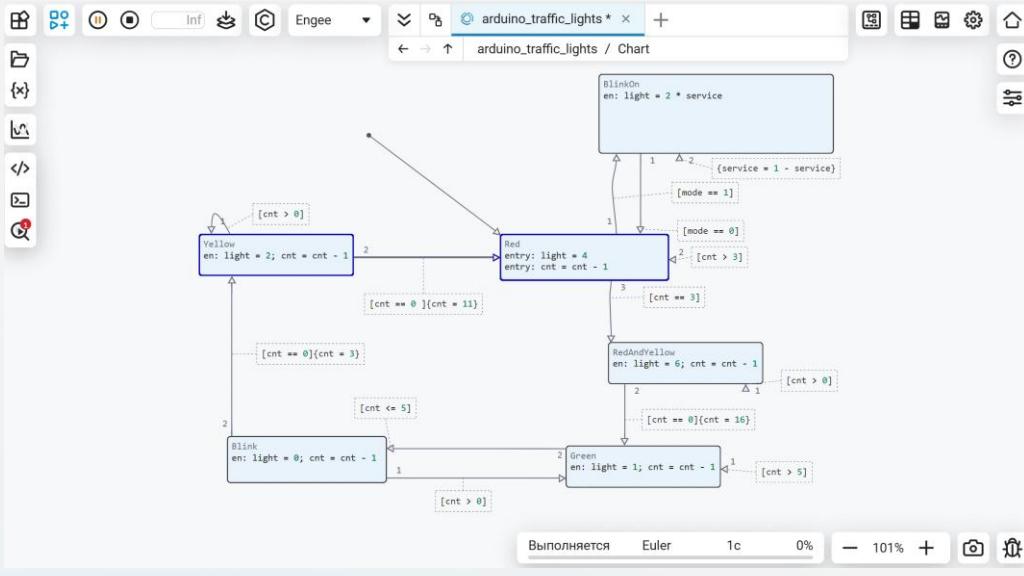
технических систем и алгоритмов с использованием МОП. Она
воплощает в себе:

- технические вычисления,
- динамическое (1-D, физическое, системное) моделирование,
- автоматическую генерацию кода,
- поддержку полунатурного моделирования.



V-цикл МОП

Модель алгоритма в Engee



Исходный код алгоритма на С

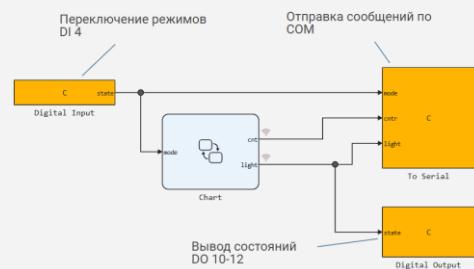
```

void arduino_traffic_lights_step() {
    int Chart_state = arduino_traffic_lights_S.Chart_state;
    int Chart_selector;
    if (Chart_state == 0) {
        if (Mode == 1) {
            ;
            Chart_selector = 6;
        } else if (Cnt > 3) {
            ;
            Chart_selector = 1;
        } else if (Cnt == 3) {
            ;
            Chart_selector = 2;
        } else {
            goto Chart_done;
        }
    } else if (Chart_state == 1) {
        if (Cnt > 0) {
            ;
            Chart_selector = 2;
        } else if (Cnt == 0) {
            Cnt = 16;
            Chart_selector = 3;
        }
    }
}

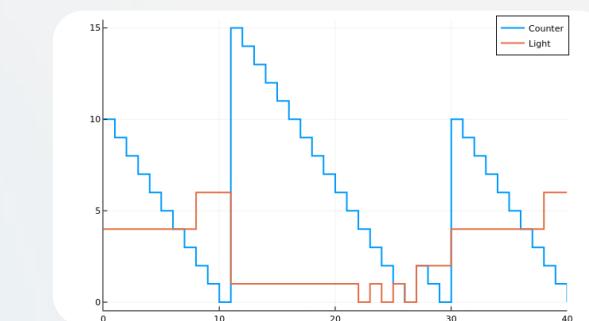
```



Алгоритм управления светофором



Наглядность при проектировании
Низкий порог входления в разработку



Гибкость в работе с переменными
Широкие возможности анализа данных

Загрузим и выполним описанную модель:

```

1 if "arduino_traffic_lights" in [m.name for m in engee.get_all_models()]
2   m = engee.open("arduino_traffic_lights");
3 else
4   m = engee.load("${@_DIR_}/arduino_traffic_lights.engee");
5 end
6
7 data = engee.run(m);

```

Сгенерируем код из модели:

```

1 engee.generate_code("${@_DIR_}/arduino_traffic_lights.engee",
2 "${@_DIR_}/arduino_traffic_lights_code")

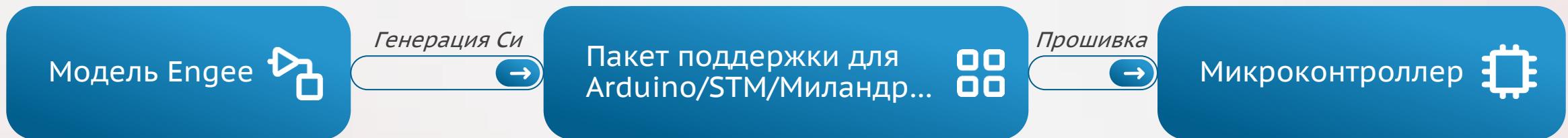
```

Автоматизация от разработки до
тестирования «в железе»

2. Таргетирование в Engee



Таргет или пакет поддержки для встраиваемых систем в Engee – это механизм перехода от модели к «железу». Он позволяет быстро и надёжно встроить алгоритмы управления в целевое устройство. Реализован с использованием платформы Engee.Интеграции.



Плюсы использования:

- экономия времени,
- снижение порога входа для алгоритмистов,
- проверка корректности работы кода на целевых устройствах.

Реализованные сценарии таргетирования:

- экспорт алгоритма,
- блоки периферии,
- независимое выполнение,
- интерактивное выполнение.



Сценарий 1: Экспорт алгоритма

The screenshot shows the Engee IDE interface. On the left, there's a file tree with a 'Файлы' (Files) tab open, showing files like 'chirp_lpf_code', 'chirp_lpf.c', 'chirp_lpf.h', 'main.c', 'arduino_chirp_lpf.ino', 'arduino_chirp_lpf.ngscript', and 'chirp_lpf_audio.wav'. In the center, there are two tabs: 'arduino_chirp_lpf.ino' and 'chirp_lpf.c'. The 'chirp_lpf.c' tab contains C code for a 'chirp_lpf_step' function. The right side of the interface shows a block diagram with a 'Float64' block connected to a 'Duty' block.

```
15 }
16 void chirp_lpf_step() {
17     double time;
18     double ToRad;
19     double FreqProd;
20     double LPF_in;
21     double OffAdd;
22     double AmpProd;
23
24     time = chirp_lpf_T.t;
25     ToRad = 314.15 * time;
26     FreqProd = ToRad * time;
27     LPF_in = sin(FreqProd);
28
29     OffAdd = LPF_out + (double)(lu);
30     AmpProd = 16u * OffAdd;
31     chirp_lpf_Y.Duty = AmpProd;
32
33     chirp_lpf_T.t = ((++chirp_lpf_T.clockTick0) * chirp_lpf_T.stepSize0;
34     chirp_lpf_T.clockTick1++;
35 }
```

Подходит для тех, кто:

- имеет готовый embedded-проект в сторонней IDE;
- готов каждую итерацию развития модели работать с кодом C/C++;
- использует Engee только для разработки алгоритма.

Основные шаги:

- Сборка модели из поддерживаемых для генерации кода блоков.
- Генерация кода.
- Software-in-the-Loop (SiL) тестирование (опционально).
- Подключение полученного кода в пользовательский проект по внешней IDE.
- Интеграция кода алгоритма и кода периферии вручную.

ПОГРУЖЕНИЕ В ENGEE:



ПОГРУЖЕНИЕ В КОД:



АВТОМАТИЗАЦИЯ:



ПОРОГ ВХОДА В ПРОЕКТ:





Сценарий 2: Блоки периферии

The screenshot shows the Engee IDE interface. On the left, there's a file browser with files like main.cpp, mik32_adc_lux_description.ngscript, mik32_adc_lux.engee, and platformio.ini. In the center, there are three main windows: 'EnableInterrupts' (containing C code for initializing interrupts), 'StartCode' (containing code for starting the block), and 'Настройки' (configuration panel). The 'Настройки' panel shows settings for a CFunction, including the number of input and output ports (both set to 1), and a calculation step of -1. At the bottom, there are standard IDE controls for file operations and zoom.

Подходит для тех, кто:

- использует свой toolchain встраивания кода С;
- не планирует на каждой итерации редактировать код периферии;
- хочет автоматизировать интеграцию кода алгоритма из Engee с кодом периферии.

Основные шаги:

- Сборка модели из поддерживаемых для генерации кода блоков.
- Перенос кода периферии в блоки С-функции, интеграция в модели.
- Генерация кода.
- Software-in-the-Loop (SiL) тестирование (опционально).
- Подключение полученного кода в пользовательский проект во внешней IDE.

ПОГРУЖЕНИЕ В ENGEE:



ПОГРУЖЕНИЕ В КОД:



АВТОМАТИЗАЦИЯ:

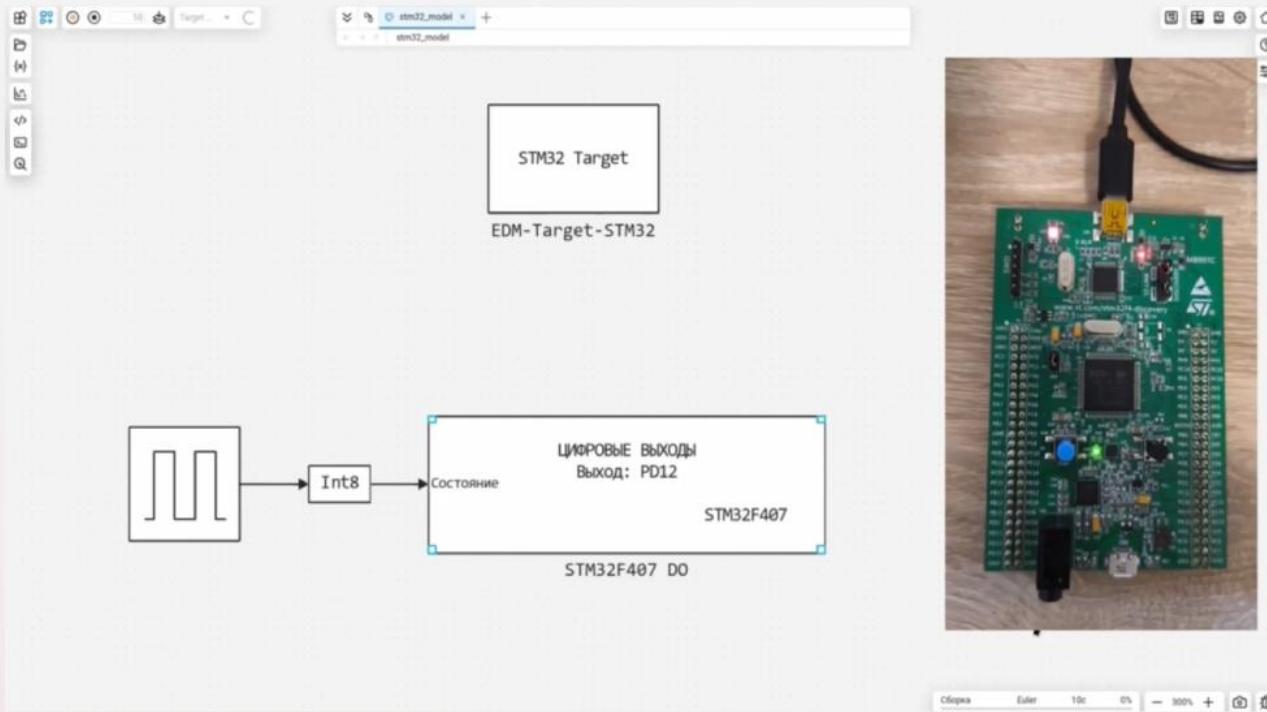


ПОРОГ ВХОДА В ПРОЕКТ:





Сценарий 3: Независимое выполнение



Подходит для тех, кто:

- не хочет/не может погружаться в код;
- готов полностью перенести разработку в Engee;
- завершил разработку модели и автоматизирует программирование целевых устройств.

Основные шаги:

1. Сборка модели из поддерживаемых для генерации кода блоков.
2. Подготовка платформы Engee.Интеграции.
3. Добавление блоков из пакета поддержки.
4. Переключение среды исполнения модели.
5. Прошивка контроллера по одной кнопке.

ПОГРУЖЕНИЕ В ENGEE:



ПОГРУЖЕНИЕ В КОД:



АВТОМАТИЗАЦИЯ:

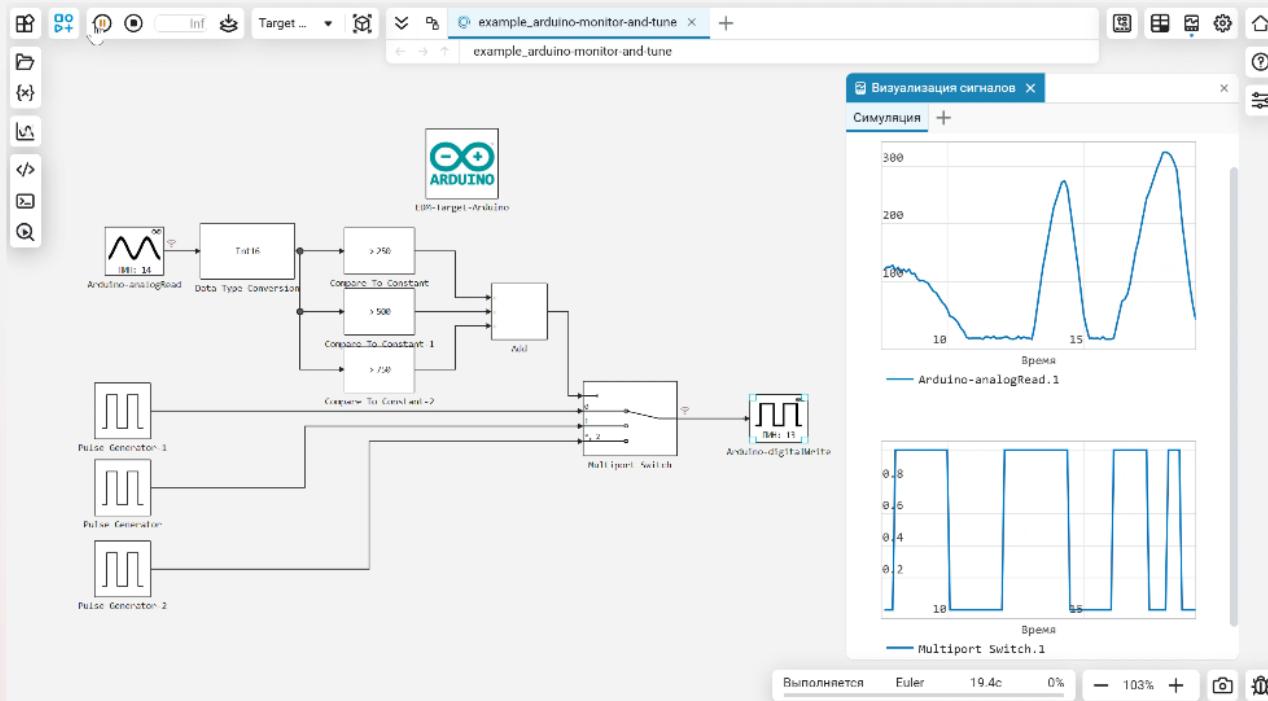


ПОРОГ ВХОДА В ПРОЕКТ:





Сценарий 4: Интерактивное выполнение



Подходит для тех, кто:

- не хочет/не может погружаться в код;
- готов полностью перенести разработку в Engee;
- хочет проверить работу алгоритма на целевом устройстве в реальном времени.

Основные шаги:

1. Сборка модели из поддерживаемых для генерации кода блоков.
2. Подготовка платформы Engee.Интеграции.
3. Добавление блоков из пакета поддержки.
4. Переключение среды исполнения модели.
5. Запуск модели на целевом устройстве по одной кнопке.

ПОГРУЖЕНИЕ В ENGEE:



ПОГРУЖЕНИЕ В КОД:



АВТОМАТИЗАЦИЯ:



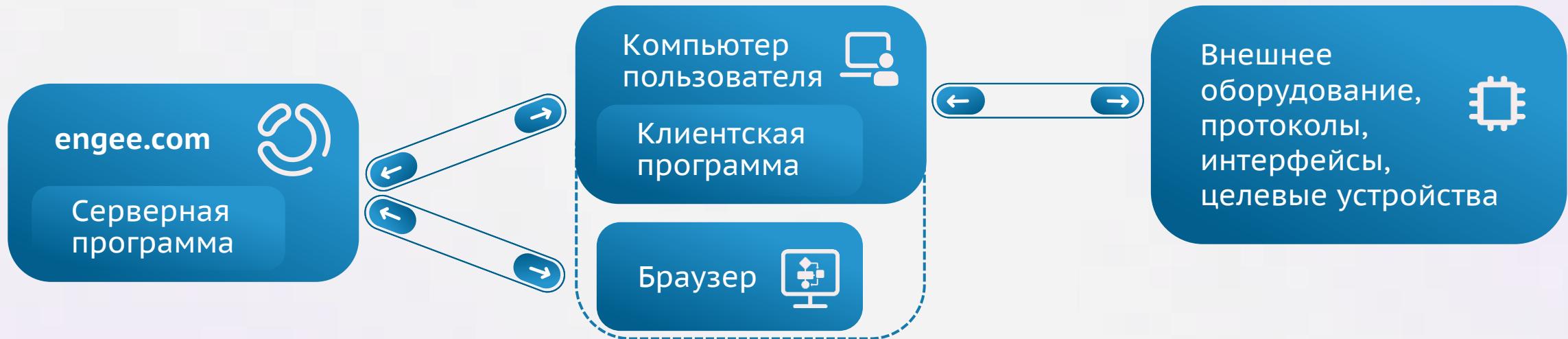
ПОРОГ ВХОДА В ПРОЕКТ:



3. Инструменты пакета поддержки



Платформа **Engee.Интеграции** – встроенная подсистема Engee, которая обеспечивает подключение Engee к внешнему оборудованию, в том числе, к целевым устройствам.



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





Платформа для подключения Engee к внешним устройствам, интерфейсам, протоколам, программным средам и встраиваемым системам через пакеты поддержки.

Engee.Интеграции

Встраиваемые системы

- Arduino
- ATmega328
- STM32
- X86
- КПМ РИТМ
- Artery
- МИК32
- Миландр

Аппаратное обеспечение

- КПМ РИТМ
- L-card
- CAN – модули
- COM порты
- Logitech G29
- Thrustmaster HOTAS

Протоколы обмена

- VISA
- UDP
- TCP/IP
- HTTP
- Modbus TCP/RTU
- Sockets

Программное обеспечение

- T-FLEX DOCs
- UNIGINE
- Универсальный механизм
- ЛОГОС
- Прагма Софт Студия

Библиотека блоков



Пакет поддержки встраиваемых устройств предоставляет новый раздел библиотеки блоков Engee:
Оборудование → Targets

The screenshot shows the Engee software interface with the following details:

- Block Library Panel:** On the left, a sidebar lists categories: vvvv, L-CARD, UNIGINE, PragmaSoft, UniversalMechanism, Targets, and Arduino. Under Targets, there are icons for EDM-Target-Arduino, Arduino-digitalRead, Arduino-digitalWrite, Arduino-analogRead, Arduino-analogWrite, Arduino-serialRead, Arduino-serialSetup, Arduino-serialWrite, EDM-Target-RIM, and EDM-Target-X86.
- Target Configuration Window:** A central window titled "newmodel_1 * newmodel1" displays target settings. It includes sections for "Targets" (set to "EDM-Target-Arduino") and "Arduino" (set to "ПИН: 13").
- Target Preview:** To the right, a preview area shows the "EDM-Target-Arduino" logo and a block labeled "Arduino-digitalWrite" with the pin number "ПИН: 13".
- Bottom Bar:** The bottom of the screen features a toolbar with icons for file operations, zoom, and other software functions.

Тестовые примеры



```
engee> engee.package.getdemos("Engee-Device-Manager")
```

The screenshot shows the Engee Device Manager software interface. On the left, there is a file browser window titled "Файлы" (Files) showing a list of examples. One example, "example_arduino-monitor-and-tune.ngscript", is selected. Below the file browser is a message: "Текущая папка не содержит git репозиторий. Можно инициализировать репозиторий здесь (кнопка Init) или". In the center, there is a flowchart titled "example_arduino-monitor-and-tune". The flowchart starts with an "analogRead" block (PIN: 6), followed by a "Data Type Conversion" block (Int16). This is followed by two "Compare To Constant" blocks: one for the value > 250 and another for > 500. The outputs of these two blocks are summed in an "Add" block. On the right side of the screen, there is a smaller diagram titled "Arduino-digitalWr" showing a "Multiport Switch" block connected to a "PIN: 13" output.

engee> engee.package.getdemos("Engee-Device-Manager")
"/user/Engee-Device-Manager-demos"
engee>

Готов Euler 10с 0% - 139% +

Блоки периферии - Arduino



EDM-Target-Arduino

Настройки

Main tab Информация Аннотация :

Тип: SubSystem

Посмотреть под маску

▼ Main group

Порт	COM4
Полное имя платы	<auto>
Путь к ArduinoCLI	D:\edm\arduino-cli.exe
Директория моделей	.

Настройки блока **EDM-Target-Arduino**
из пакета поддержки Arduino



Порт –

СОМ-порт целевого устройства :

`<auto> COM4 /dev/ttyUSB0`

Полное имя платы –

в формате VENDOR:ARCHITECTURE:BOARD_ID :

`<auto> arduino:avr:mega`

Путь к ArduinoCLI –

путь до исполняемого файла тулчайна :

`<auto> D:\edm\arduino-cli.exe`

Директория моделей –

путь хранения исходного кода моделей :

`. .\models`

Блоки периферии - Arduino



Arduino-digitalWrite

ПИН: 12

Arduino-digitalWrite

Number of input ports: 1

Number of output ports: 0

Function name: digitalWrite

Sample time: -1

Редактировать исходный код

Настройки

Основные Parameters Work Variables

Тип: CFunction

Посмотреть маску

Arduino-digitalWrite

OutputCode StartCode TerminateCode SharedCode

```
/* Этот код вызывается на каждом шаге расчета модели */
#ifndef Arduino
if (*(char *) Work->init == 0){
    pinMode(pin, OUTPUT);
    *Work->init = 1;
}
if (input1){
    digitalWrite(pin, HIGH);
} else {
    digitalWrite(pin, LOW);
}
#endif
```

Информация

Интерфейс функции void CFunction_step(double t, double input1, Work

Готов Euler 10c 0% 300% +

Блоки периферии – STM32



stm32 * x

stm32

Настройки X

Main tab Информация Аннотация

Тип: SubSystem

Посмотреть под маску

▼ Target STM32

Семейство: f3xx

Программатор: stlink

Путь к утилите для прошивки: cs/STM32Cube/STM32CubeProgrammer/bin/STM32_Programmer_CLI

Путь к проекту CubeMX: /home/art/STM32Cube/Projects/f407_example

Планировщик: freeRTOS

baremetal
✓ freeRTOS
/dev/ttyUSB0

▼ Настройка независимого режима

COM-порт: /dev/ttyUSB0

Резервируемый модуль USART: 2

Готов Euler 10c 0% — 150% +

stm32

EDM-Target-STM32

STM32 DI

Порт: GPIOC Вход: 8

STM32 DO

Порт: GPIOD Выход: 12

STM32 USART RX

Модуль: USART1

STM32 USART TX

Модуль: USART1

STM32 ADC

Модуль: ADC1



Документация Engee → Внешняя аппаратура и ПО → Интеграции со встраиваемыми системами

Документация Engee

Поиск

Глобус

Логин

Главная

- ▶ Об Engee
- ▶ Прикладные применения
- ▶ Среда вычислений
- ▶ Среда моделирования и симуляции
- ▶ Учебные курсы
- ▶ Генерация кода
- ▼ Внешняя аппаратура и ПО
 - ▶ Общее
 - ▶ Поддержка протоколов обмена
 - ▶ Интеграции с программным обеспечением
 - ▶ Интеграции с аппаратным обеспечением
 - ▼ Интеграции со встраиваемыми системами
 - Пакет поддержки Arduino
- ▶ Примеры
- ▶ Дополнительные материалы
- ▶ Что нового

Документация Engee / Внешняя аппаратура и ПО / Интеграции со встраиваемыми системами / Пакет поддержки Arduino

Режимы выполнения

Поведение режимов Target Hardware аналогично режимам выполнения моделей на КПМ «РИТМ» (подробнее см. в статье [Режимы выполнения модели](#)).

Режим «Запустить модель на железе»

Как запустить нажмите «Запустить модель на железе »:

   10 

Как управлять моделью можно запускать/останавливать/задавать время симуляции и компилировать модель.

Что происходит генерируется интерактивное приложение реального времени и запускается на целевом оборудовании. Отслеживать выполнение модели можно в Engee, в окне [визуализации сигналов](#)  и в [инспекторе данных](#) .

Что можно делать в режиме управлять моделью, менять параметры «на лету», записывать сигналы и просматривать их.

Режим «Запуск в независимом режиме»

Как запустить нажмите «Запуск в независимом режиме »:

 Target ... 

Как управлять моделью средствами конкретной платформы (веб-панель, утилиты и т.п.).

Что происходит генерируется автономное приложение реального времени и запускается на целевой платформе независимо от Engee. Выполнение модели видно на дисплее/веб-панели/журналах целевой платформы.

Что можно делать модель функционирует на целевой платформе вне Engee.

<https://engee.com/helpcenter/stable/ru/engee-hardware/arduino-support-package.html#режимы-выполнения>

4. Таргетирование в примерах

Независимый режим – Arduino Blink



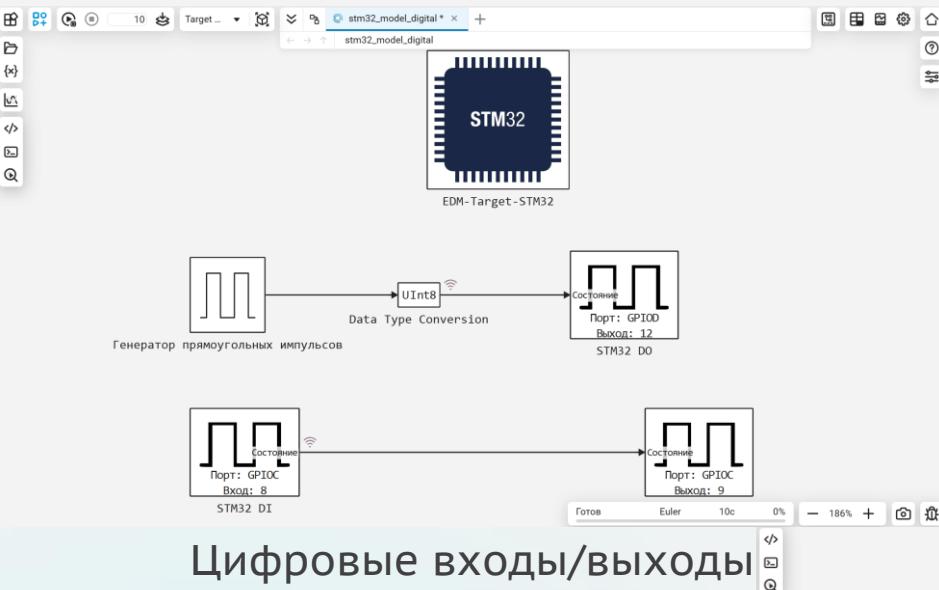
The image shows a web browser window with two tabs. The left tab is titled 'IP Webcam' and displays a control panel for a camera connected to an Arduino Uno. It includes buttons for 'Режим просмотра' (View mode), 'Полноэкранный режим' (Full screen mode), 'Режим прослушивания' (Listening mode), 'Нет аудио' (No audio), 'HTML5', 'Приём аудио' (Audio input), and a 'Снять фото' (Take photo) button. The right tab is titled 'Android Webcam Server' and shows a live video feed of an Arduino Uno microcontroller. The board is green with various components like chips, capacitors, and a USB port. A black USB cable is connected to the board. The video feed is displayed in a browser window with standard controls at the bottom.

Интерактивный режим – Arduino Blink

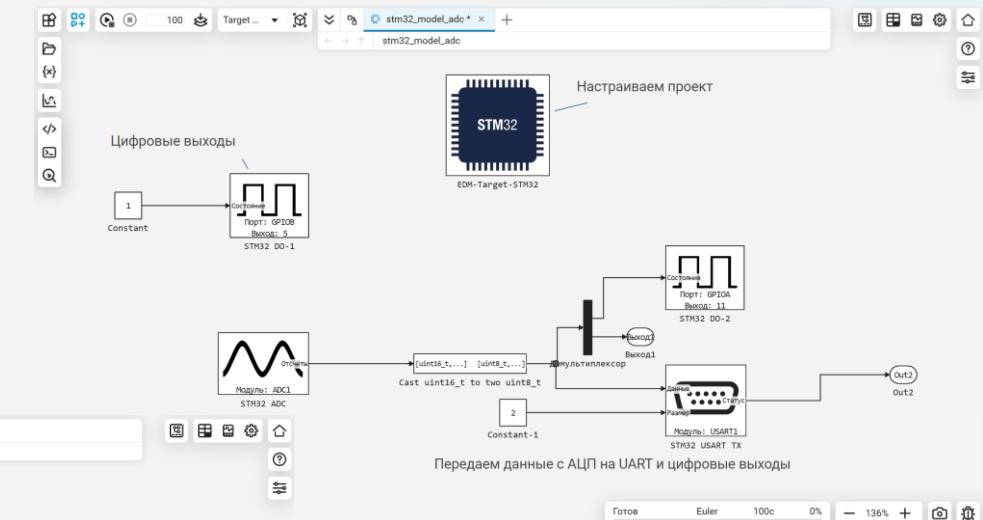


The screenshot shows the Engee 25.11.2 software interface. On the left, there is a signal visualization window titled "Визуализация сигналов" (Signal Visualization) showing a waveform labeled "Pulse Generator.1". The waveform has vertical ticks at 0.2, 0.4, 0.6, and 0.8, and horizontal ticks labeled 8, 9, and 10. Below the waveform is the text "Время" (Time). On the right, there is a schematic diagram showing a "Generator" block connected to an "Arduino-digitalWrite" block, which is labeled "ПИН: 13". The Arduino board in the video feed is an Arduino Uno, with its digital pins labeled 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, and 1. The Arduino's LED is visible, showing a blinking pattern. The video feed also shows the Arduino's power source, a 9V battery, and some jumper wires.

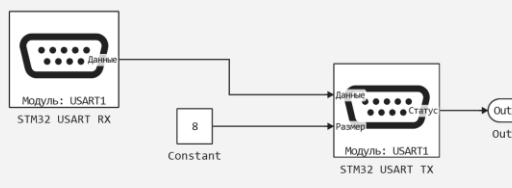
Тестовые примеры – STM32



Цифровые входы/выходы



Чтение аналогового сигнала

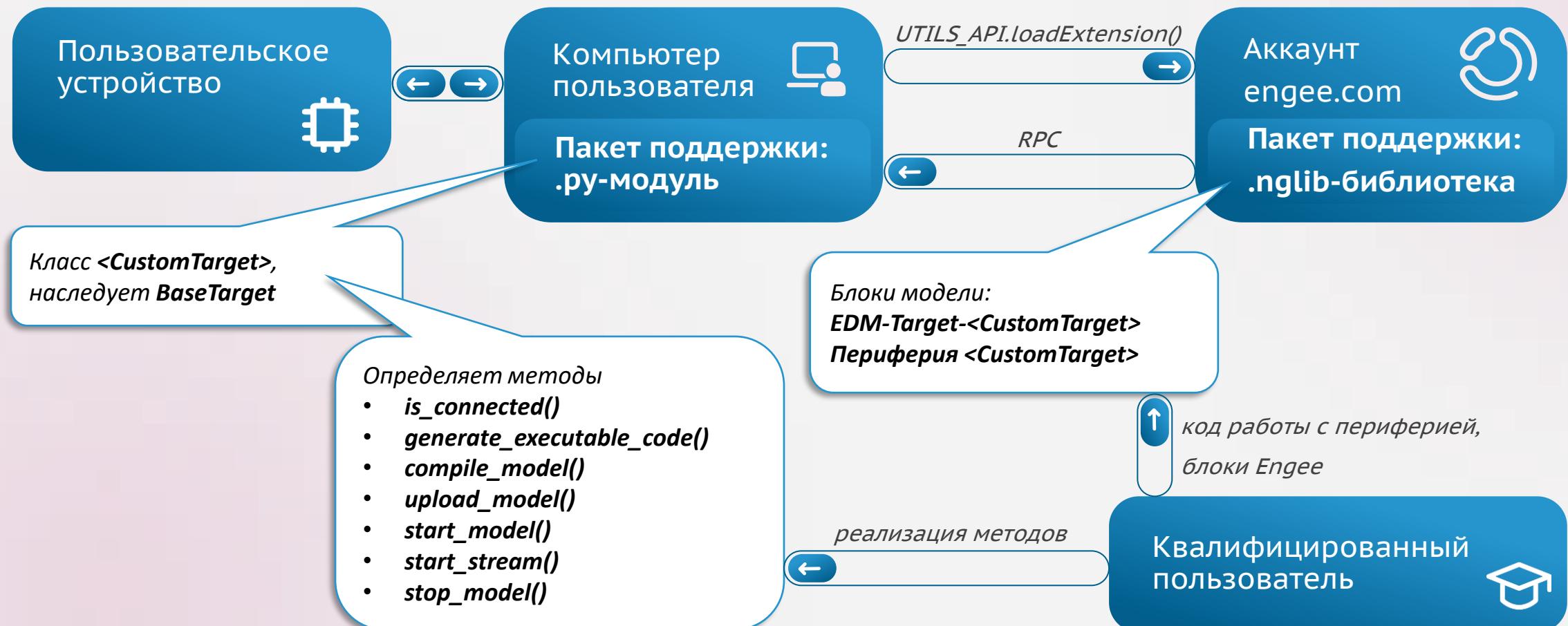


Последовательный порт:
чтение/запись

5. Открытый механизм поддержки пользовательских устройств



Скоро планируется представить **открытый «reference target»** - пакет поддержки на базе ATmega328, по аналогии с которым каждый пользователь сможет разработать свой собственный пакет поддержки.



6. Планы, перспективы



Новые пакеты поддержки:

- STM32 (F1xx – F4xx)
- ATmega328
- X86
- КПМ РИТМ
- Микрон МИК32
- Миландр 1986ВExx
- Artery
- Texas Instruments C2000

Расширение пакетов поддержки блоками:

- Arduino
- STM32
- ATmega328
- X86
- КПМ РИТМ

Расширение пакетов поддержки методами:

- STM32
- ATmega328
- X86
- КПМ РИТМ

Новые сценарии:

PIL (Processor-in-the-Loop) тестирование

Открытый «reference target»

ATmega328

Поддержка пользователей:

Наполнение документации
Покрытие примерами
Тренинги, курсы

Полезные материалы



[Пакет поддержки Arduino](#) ↗

[Курс по генерации кода в Engree](#) ↗

[Engree.Интеграции](#) ↗

[Методы UTILS_API](#) ↗

[Генератор кода Engree](#) ↗

[Пример пользовательского пакета](#) ↗

[Пользовательские пакеты поддержки](#) ↗

[Поддерживаемые блоки Engree](#) ↗



ЭКСПОНЕНТА
ЦЕНТР ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ
И МОДЕЛИРОВАНИЯ

Спасибо за внимание!

