Изменено	10/03/2023 11:45:32	Создано	10/03/2023 11:45:32	https://www.electronshik.ru/news/show/13304?from=terraelectronica

Знакомство с пакетом расширения X-CUBE-AI для реализации искусственного интеллекта. Часть 1

Перед вами перевод руководства с методическими указаниями по созданию законченного проекта с элементами искусственного интеллекта (ИИ) на базе микроконтроллеров семейства STM32. В процессе создания проекта производится автоматическое преобразование предварительно обученной нейронной сети и связывание сгенерированной библиотеки с основной программой. В руководстве описан пакет расширения X-CUBE-AI, полностью интегрированный в среду разработки STM32CubeMX. Также в данном руководстве описываются дополнительные тестовые приложения, предназначенные для оценки корректности работы (валидации) и определения производительности нейронной сети.

Основная часть данной публикации представляет собой практическое руководство по быстрому созданию проекта с ИИ для микроконтроллера (МК) STM32. В примерах используются отладочная плата и несколько общедоступных моделей для глубокого обучения (DL-модели). После внесения незначительных изменений данный проект также можно будет перенести на отладочную либо собственную плату на базе любого из микроконтроллеров семейств , , , , , или .

Также в этой статье подробно описано использование пакета X-CUBE-AI для создания дополнительных приложений, позволяющих оценить производительность ИИ и проверить его работоспособность. Уделено внимание внутренним особенностям пакета, таким как генерируемая им библиотека нейронной сети. Дополнительную информацию (использование утилиты командной строки, поддерживаемые фреймворки и слои, формируемые метрики) можно найти в месте установки пакета (папка «Documentation»).

Общие сведения

Пакет расширения X-CUBE-AI предназначен для создания проектов с поддержкой ИИ на базе микроконтроллеров STM32 с ядрами ARM® Cortex®-M.

Текущая версия руководства основана на:

X-CUBE-AIверсии 4.1.0;

клиентском API нейронной сети версии 1.1.0;
утилите командной строки версии 1.1.0.
Предварительно обученная DL-модель Keras, используемая в данном руководстве:

- распознавание человеческой активностиси с пользованием сверточной нейронной сети в Keras.

Что такое STM32Cube?

STM32Cube – это оригинальная программная платформа компании STMicroelectronics, обеспечивающая значительное увеличение производительности труда разработчика за счет уменьшения трудоемкости процесса разработки ПО, а также сокращения временных и финансовых затрат. Платформа STM32Cube поддерживает всю линейку микроконтроллеров STM32.

В состав платформы STM32Cube входят:

Набор удобных программных средств разработки, охватывающий все этапы проекта, начиная от разработки концепции и заканчивая его реализацией, в том числе:
STM32CubeMX - среда визуального конфигурирования, которая позволяет сконфигурировать микроконтроллер с использованием графического интерфейса и автоматически сгенерировать соответствующий код инициализации на языке Си;
STM32CubeIDE - интегрированная среда разработки (IDE) с возможностью конфигурирования периферийных устройств, генерации, компиляции и отладки кода;
STM32CubeProgrammer (STM32CubeProg) - программный пакет для программирования МК, поставляемый в двух вариантах: с графическим интерфейсом и с интерфейсом командной строки;
STM32CubeMonitor-Power (STM32CubeMonPwr) - программный пакет для измерения тока, помогающий оптимизировать ток потребления целевого МК.
Пакеты поддержки МК и МП для STM32Cube, комплекты встраиваемого ПО, специфичные для каждого семейства (например, STM32CubeF7 для семейства STM32F7), которые включают в себя:
STM32CubeHAL - функции уровня аппаратной абстракции, обеспечивающие максимальную переносимость между всеми моделями STM32;
STM32CubeLLAPI - API нижнего уровня, обеспечивающий максимальную производительность ПО и оптимизированный по размеру кода, а также предоставляющий пользователю практически полный контроль над аппаратными средствами;
Широкий набор программных компонентов промежуточного уровня, таких как OCPB, графические библиотеки, библиотеки поддержки USB, FAT и TCP/IP;
все необходимые для разработки утилиты, а также демонстрационные программы и многочисленные примеры по использованию периферийных модулей.
Пакеты расширения STM32Cube, которые содержат компоненты встраиваемого ПО, дополняющие функциональность пакетов поддержки МК и МП тем, что:
расширяют возможности стандартных программных компонентов промежуточного уровня;
содержат примеры ПО для определенных отладочных плат производства STMicroelectronics.

Чем X-CUBE-AI дополняет платформу STM32Cube?

Пакет X-CUBE-AI расширяет функциональные возможности среды STM32CubeMX, добавляя в нее автоматический генератор библиотеки нейронной сети. Этот генератор преобразует файлы моделей предварительно обученных нейронных сетей, формируемые различными фреймворками глубокого обучения (такими как Caffe, Keras, Lasagne, TensorFlow™ Lite и ConvNetJs), в оптимизированную по скорости выполнения и объему памяти (ОЗУ и Flash) библиотеку. Сгенерированная библиотека автоматически интегрируется с пользовательским приложением, в результате чего на выходе получается полностью настроенный проект, готовый к компиляции и исполнению на микроконтроллере STM32.

Кроме того, пакет X-CUBE-AI добавляет в среду STM32CubeMX новые фильтры, применяемые на этапе создания проекта. Эти фильтры позволяют отбирать только те микроконтроллеры, параметры которых отвечают определенным требованиям, таким как объем ОЗУ или Flash-памяти, предъявляемым используемой нейронной сетью.

Процессорное ядро X-CUBE-AI

Процессорное ядро X-CUBE-AI, показанное на рисунках 1 и 2, входит в состав пакета расширения X-CUBE-AI, который будет рассмотрен в разделе «Расширение STM32CubeMX». Это ядро содержит все необходимые инструменты, позволяющие из предварительно обученной нейронной сети (DL-модели) автоматически генерировать ее оптимизированную и надежную реализацию на языке Си для применения во встраиваемых системах с ограниченными аппаратными ресурсами. Сгенерированную библиотеку нейронной сети для МК STM32 (как специальные, так и общие компоненты) можно напрямую интегрировать в IDE-проект или подключить к системе сборки на основе Make. Помимо библиотеки, экспортируется четко определенный API нейронной сети (подробнее об этом – в разделе «Клиентский API нейронной сети»), позволяющий разрабатывать приложения с элементами ИИ. Поддерживаются различные фреймворки глубокого обучения (раздел «Поддерживаемые фреймворки глубокого обучения»).

Для использования всех возможностей ядра X-CUBE-AI в пакете имеется специальное консольное приложение, которое выполняет основные операции по анализу, валидации и генерации оптимизированной Си-библиотеки нейронной сети для устройств STM32 [7]. Кроме того, данная утилита может выполнять квантование модели Keras после обучения.

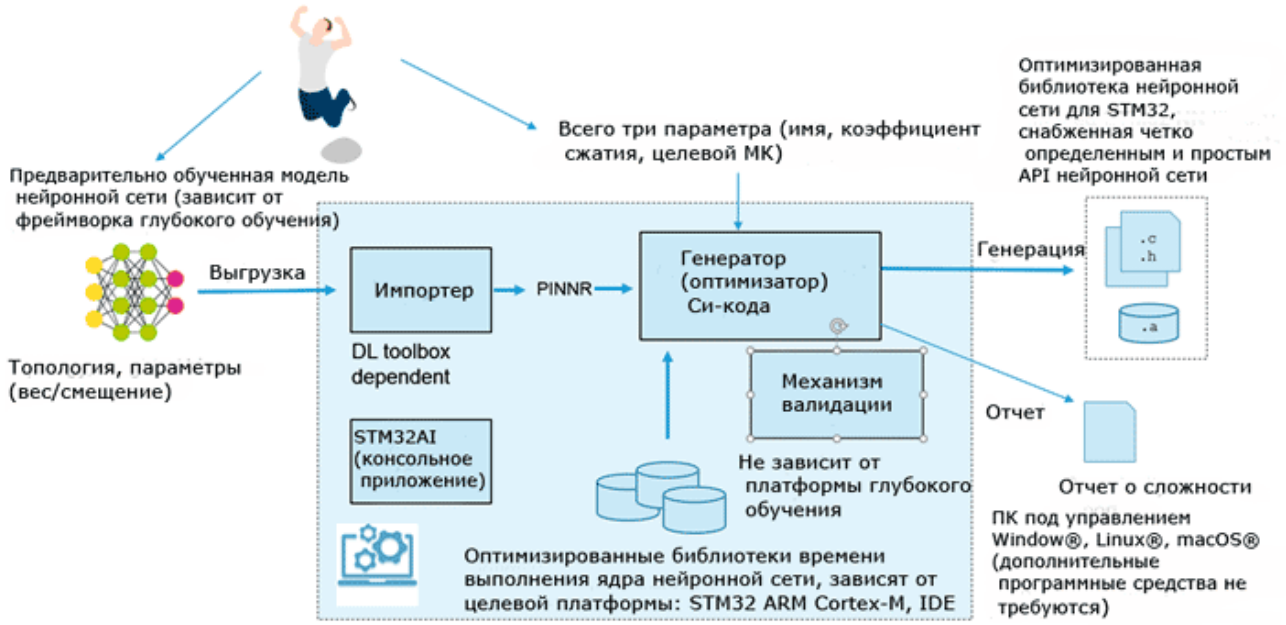

Рис. 1. Процесорное ядро X-CUBE-AI

Для настройки используется очень простой интерфейс. При наличии файла предварительно обученной DL-модели нейронной сети пользователю необходимо указать всего три параметра:

«имя» определяет имя сгенерированной модели на языке Си (значение по умолчанию “network”);
«степень сжатия» определяет коэффициент сжатия, которое применяется для уменьшения места, занимаемого параметрами веса/смещения в памяти (раздел «Оптимизатор потока выполнения и занимаемой памяти»);
«семейство STM32» используется для выбора конкретной оптимизированной библиотеки ядра нейронной сети.
На рис. 2 показаны основные поддерживаемые параметры загружаемых DL-моделей, а также модули времени выполнения целевой системы.

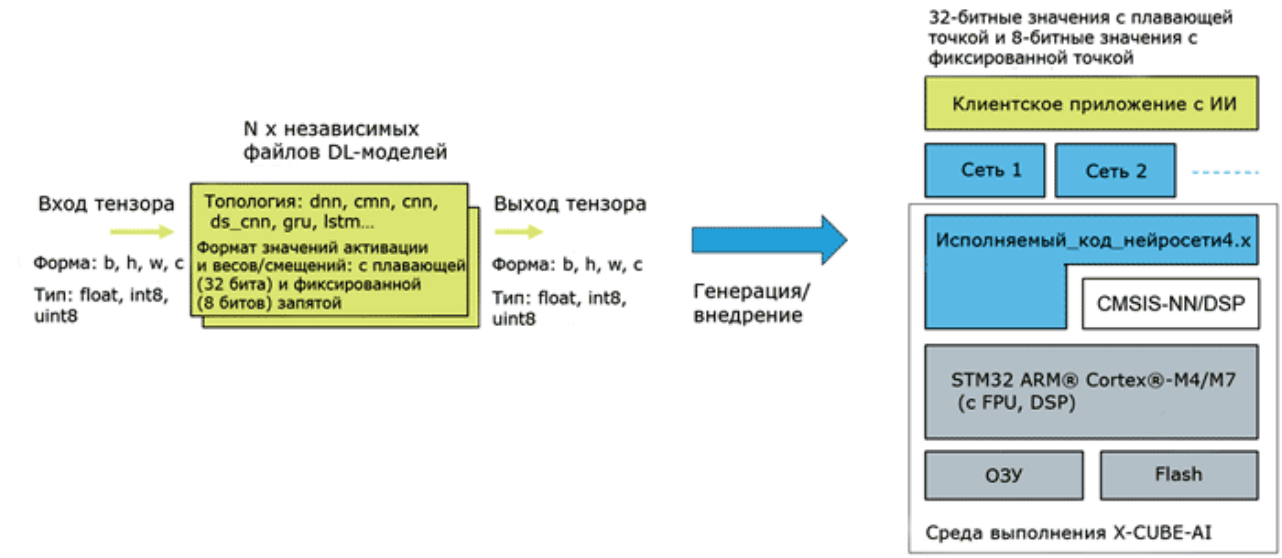

Рис. 2. Обзор пакета X-CUBE-AI

Поддерживаются только простые входные/выходные тензоры:

четыре измерения: пакет, высота, ширина, канал (в формате «канал-последний») [9];
типы с плавающей (32 бита) и фиксированной (8 битов) запятой.
Генерируемые модели на языке Си оптимизированы для процессорных ядер STM32 ARM® Cortex®-M4/M7 с поддержкой DSP-инструкций и блоком операций с плавающей запятой.
Генератор кода X-CUBE-AI может использоваться для генерации и внедрения предварительно квантованных моделей Keras, использующих 8-битные значения (целочисленные/с фиксированной запятой), и квантованных моделей TensorFlow™ Lite (рис. 3). Для применения моделей Keras необходим сам файл с преобразованной моделью (h5*) и специальный файл конфигурации формата тензора (json).

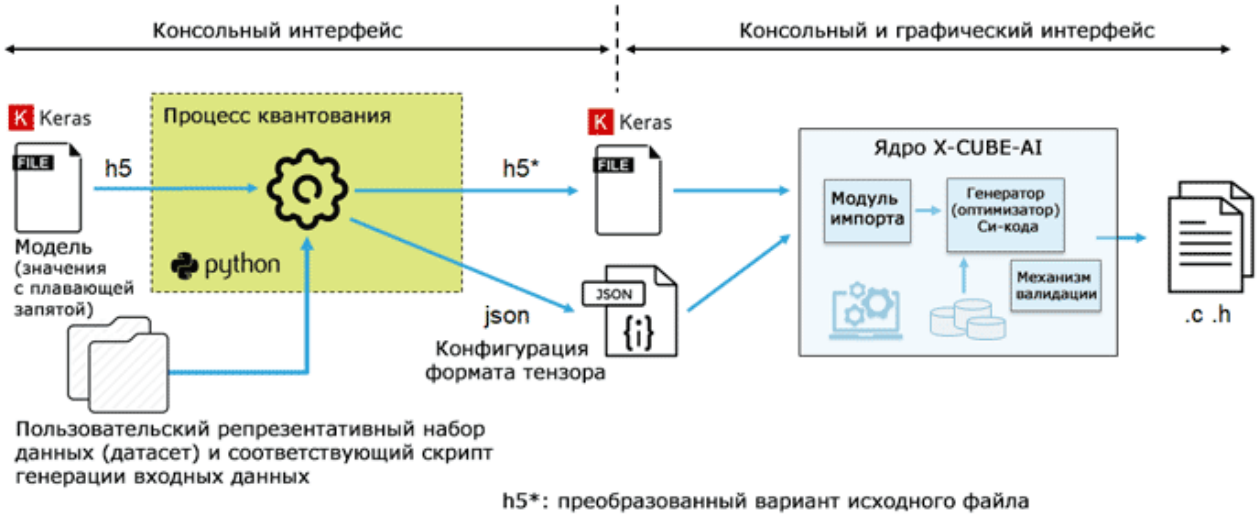

Рис. 3. Процесс квантования

Генератор кода преобразует (квантует) значения весов, смещений и соответствующие значения активации, представленные в формате с плавающей запятой, в 8-битный формат. Полученные значения применяются к оптимизированным и адаптированным Си-реализациям операторов поддерживаемых ядер [8]. При использовании модели TensorFlowLite возможно использование операторов, работающих с плавающей запятой, в этом случае операторы преобразования «float ? 8 бит» и «8 бит ? float» будут вставлены в модель автоматически. Цель этой операции – уменьшение размера модели с одновременным уменьшением времени, необходимого для формирования решения (последнее также положительно сказывается на энергопотреблении) без значительного ухудшения точности модели

Для генерации файла с преобразованной моделью и соответствующего ему файла конфигурации формата тензора из предварительно обученной модели Keras, использующей значения с плавающей запятой, применяется консольная утилита stm32ai, которая выполняет все необходимые операции по квантованию исходной обученной модели [7].

Расширение STM32CubeMX

STM32CubeMX представляет собой среду визуального конфигурирования микроконтроллеров STM32. Эта среда позволяет с помощью одного клика создать законченный IDE-проект для STM32, автоматически генерируя код инициализации микроконтроллера и целевой платформы (выводы, система синхронизации, периферийные модули и промежуточное ПО). При этом выбор и настройка параметров производятся при

помощи различных мастеров с графическим интерфейсом (мастер разрешения конфликтов конфигурации выводов, помощник настройки системы синхронизации и другие).

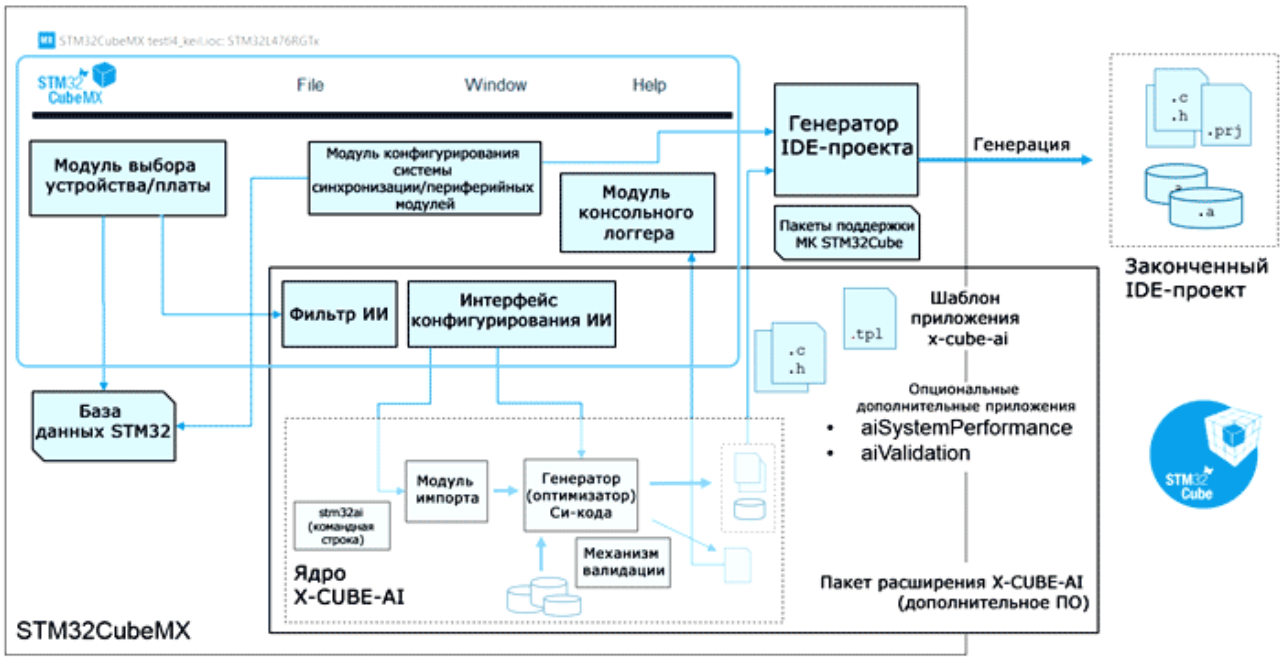

Рис. 4. Ядро X-CUBE-AI в составе STM32CubeMX

С точки зрения пользователя среды STM32CubeMX установленный пакет расширения X-CUBE-AI можно рассматривать как дополнительный периферийный модуль или дополнительную программную библиотеку. Верхний уровень ядра X-CUBE-AI (рисунок 4) реализует следующий функционал:

расширяет возможности фильтрации при выборе МК – в окне фильтра появляется дополнительный флажок «Artificial Intelligence», позволяющий исключить из рассмотрения устройства, не имеющие памяти требуемого объема. В частности, при выборе этого флажка сразу же отфильтровываются модели STM32 с процессорными ядрами, отличными от ARM® Cortex®-M4 или -M7;
предоставляет законченный мастер конфигурации подсистемы ИИ с графическим интерфейсом, позволяющий загружать несколько DL-моделей. Данный мастер позволяет выполнять валидацию сгенерированного Си-кода на настольном ПК и целевой плате;
расширяет возможности генератора IDE-проектов, обеспечивая генерацию оптимизированной для конкретного ядра STM32 ARM® Cortex®-M библиотеки нейронной сети и ее интеграцию в проект для выбранной среды разработки;
предоставляет опциональные надстройки, позволяющие генерировать законченный и полностью готовый к использованию проект приложения тестирования ИИ, использующий сгенерированные библиотеки ИИ. Пользователю достаточно импортировать этот проект в свою любимую среду разработки, скомпилировать его и загрузить полученную прошивку в целевой микроконтроллер. Не нужно ни добавлять свой код, ни вносить изменения в сгенерированный;
позволяет в один клик автоматически сгенерировать, запрограммировать и запустить на целевом устройстве программу валидации ИИ (с поддержкой внешней памяти).

Акронимы, аббревиатуры и определения

В таблице 1 приведены акронимы и аббревиатуры, встречающиеся в данном документе.

Таблица 1. Определение терминов, используемых в документе

ИИ	Искусственный интеллект, иногда называемый машинным интеллектом. Термином «ИИ» обычно обозначают устройства, которые выполняют свои задачи таким образом, что с человеческой точки зрения они кажутся «умными». То есть, речь идет о способность цифровых устройств выполнять задачи, считающиеся прерогативой интеллекта.
DL	Глубокое обучение (Deep Learning), также известное как глубокое структурное обучение или иерархическое обучение. DL-модели пытаются имитировать процессы обработки и передачи сигналов в нервных системах живых существ.
ML	Машинное обучение (Machine Learning) – класс методов ИИ, наделяющих систему способностью автоматически обучаться и совершенствоваться не за счет явного программирования, а за счет опыта, нарабатываемого в процессе применения решений множества сходных задач.
MACC	Сложность, выраженная числом операций умножения с накоплением. Эта единица измерения характеризует сложность DL-модели с точки зрения требуемыхвычислительных ресурсов.
PINNR	Платформенно-независимое представление нейронной сети – файл, формируемый модулем импорта ядра X-CUBE-AI, который содержит платформенно-независимое и переносимое внутреннее представление загруженной DL-модели, используемое на последующих этапах (оптимизация и генерация кода на языке Си)

Требуемое ПО

На компьютере должно быть установлено следующее программное обеспечение (раздел «Установка пакета X-CUBE-AI»):

STM32CubeMX версии 5.0.1 или выше;
пакеты расширения STM32CubeMX AI (X-CUBE-AI) 4.1.0;
STM32CubeProgrammer (STM32CubeProg) версии1.0 или выше. Этот пакет необходим для выполнения автоматической валидации ПО на целевой плате в том случае, если вы не используете среду STM32CubeIDE.
Также на компьютере должен быть установлен любой из указанных наборов инструментальных средств или интегрированных сред разработки (IDE) для STM32:

STM32CubeIDE версии0.1 или выше;
TrueSTUDIO® for STM32 версии0.1 или выше (atollic.com/truestudio);
IAR Embedded Workbench™ IDE - ARM v8.x илих (www.iar.com/iar-embedded-workbench);
?Vision® V5.25.2.0 - Keil® MDK-ARM Professional (www.keil.com);
System Workbench for STM32 (SW4STM32);
GNU ARM Embedded Toolchain (developer.arm.com/open-source/gnu-toolchain/gnu-rm).
Пакет X-CUBE-AI может использоваться на следующих операционных системах:

Windows® 10;

Ubuntu® 18.4 и Ubuntu® 16.4 (и производных от них);
macOS® (x64) (протестировано на OSX® ElCapitan и Sierra).

Лицензия

Пакет X-CUBE-AI предоставляется на условиях лицензионного соглашения Mix Ultimate Liberty + OSS + 3rd-party V1 (SLA0048).

Установка пакета X-CUBE-AI

Прежде чем устанавливать пакет расширения X-CUBE-AI, необходимо загрузить, установить и запустить среду STM32CubeMX (версия не ниже 5.0.1). Для установки пакета расширения выполните следующие действия:

Выберите в меню пункт «Help» ? «[Manage Embedded Software Packages» либо нажмите кнопку «INSTALL/REMOVE» (рис. 5).

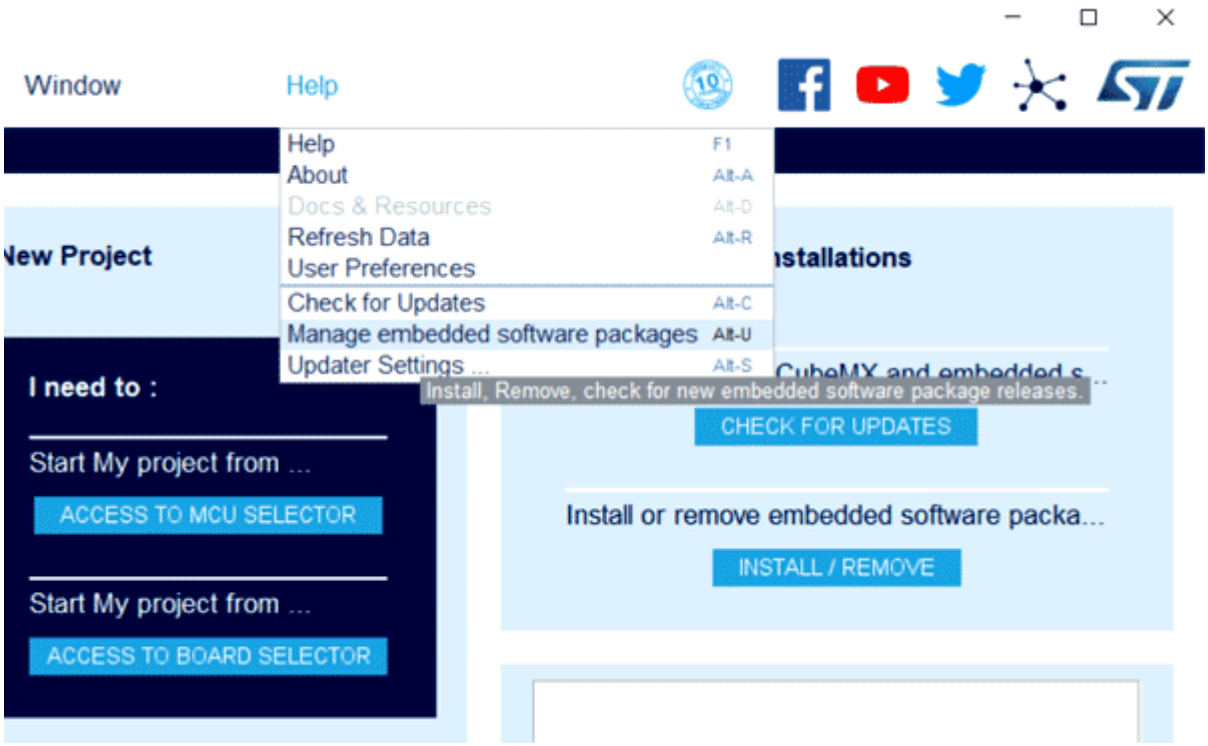

Рис. 5. Управление пакетами расширения в STM32CubeMX

В открывшемся диалоговом окне Embedded Software Packages Manager нажмите кнопку «Refresh», чтобы получить актуальный список доступных пакетов расширения. Перейдите на вкладку STMicroelectronics (рис. 6).

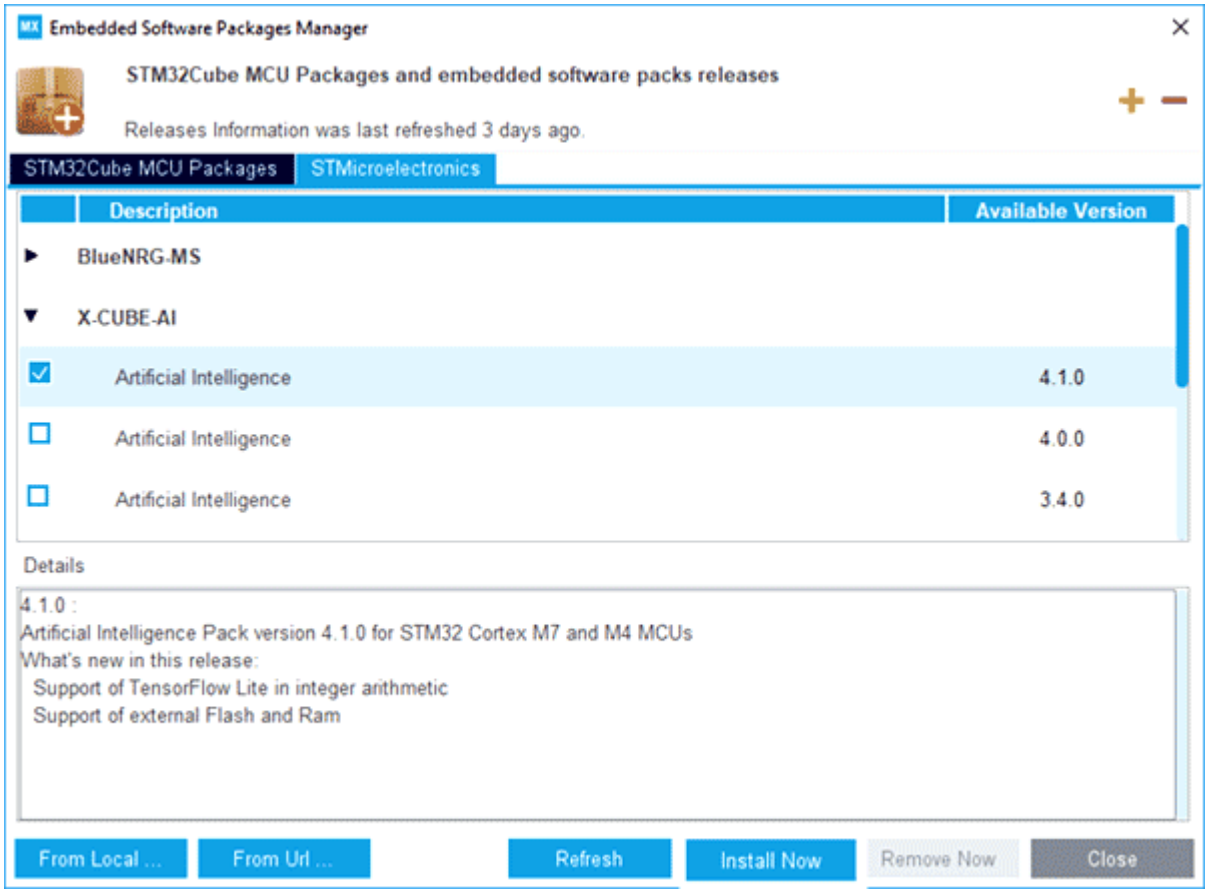

Рис. 6. Установка X-CUBE-AI в STM32CubeMX

Если пакет X-CUBE-AI уже установлен, перед повторной установкой рекомендуется его удалить.

Выберите требуемую версию пакета, поставив соответствующий флажок, и запустите установку, нажав кнопку «Install Now». После завершения установки отмеченный флажок станет зеленым и можно будет нажать кнопку «Close» (рис. 7).

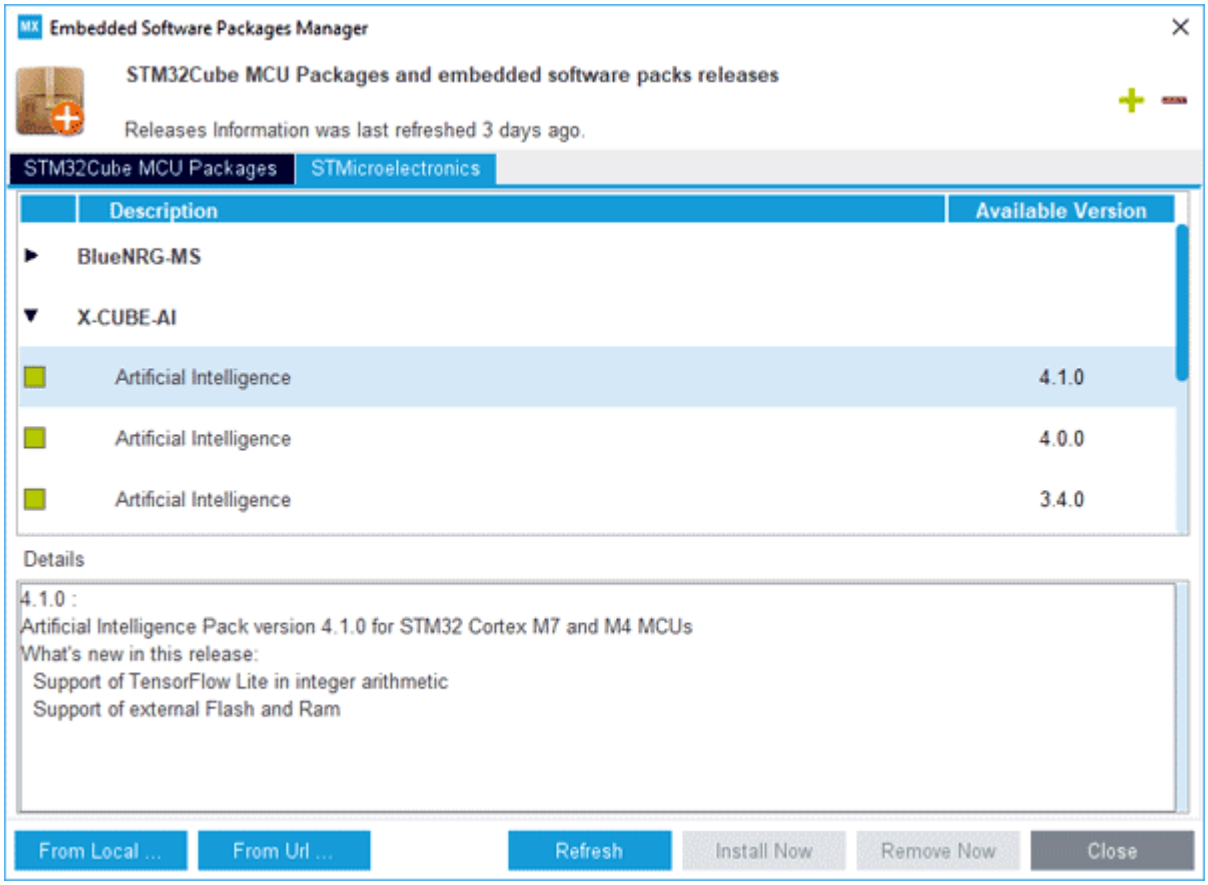

Рис. 7. Пакет X-CUBE-AI установлен в STM32CubeMX

Создание нового проекта STM32 AI

Выбор МК и отладочной платы

Запустите среду STM32CubeMX и нажмите кнопку «Access To Mcu Selector» или «Access To Board Selector». Также можно воспользоваться меню «File» ? «New Project...» или комбинацией горячих клавиш CTRL + N (рис. 8).

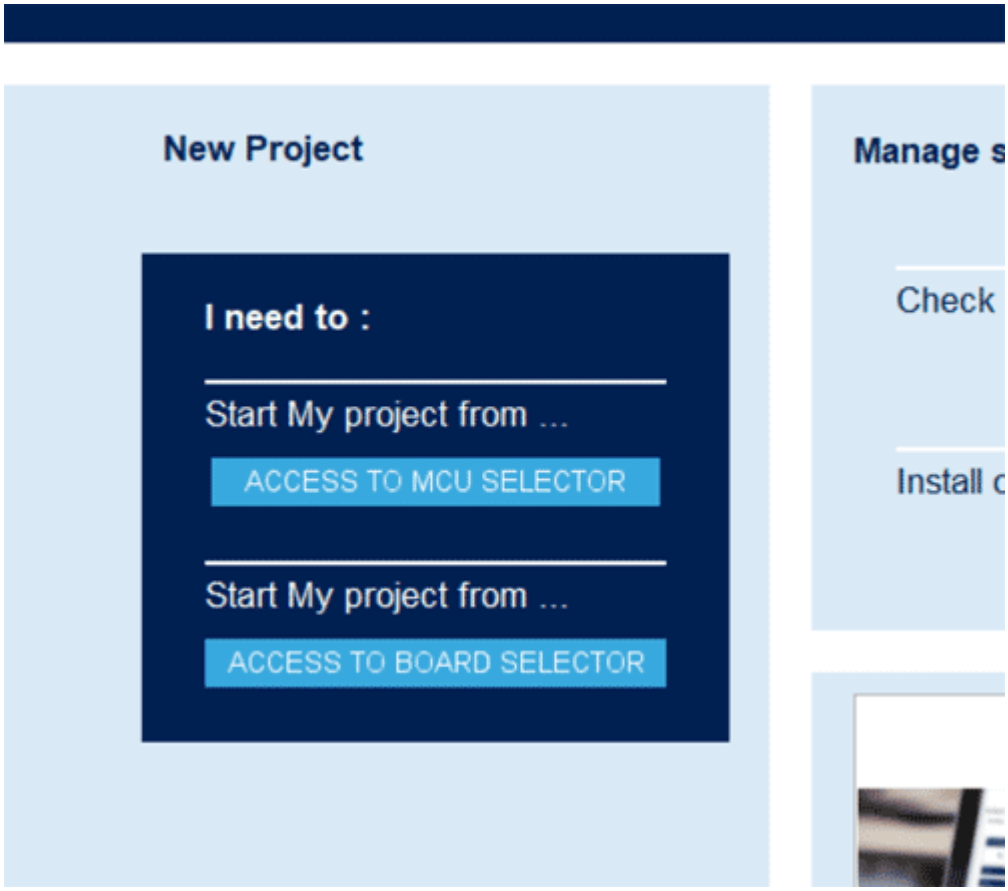

Рис. 8. Создание нового проекта

Теперь можно выбрать конкретную модель МК или отладочной платы, используя стандартные инструменты STM32CubeMX. Дополнительный фильтр «Artificial Intelligence», появившийся после установки пакета X-CUBE-AI, позволяет исключить из рассмотрения МК, у которых объема встроенной памяти (ОЗУ, Flash или обеих) недостаточно для хранения оптимизированной библиотеки нейронной сети. Данный фильтр показан на рис. 9.

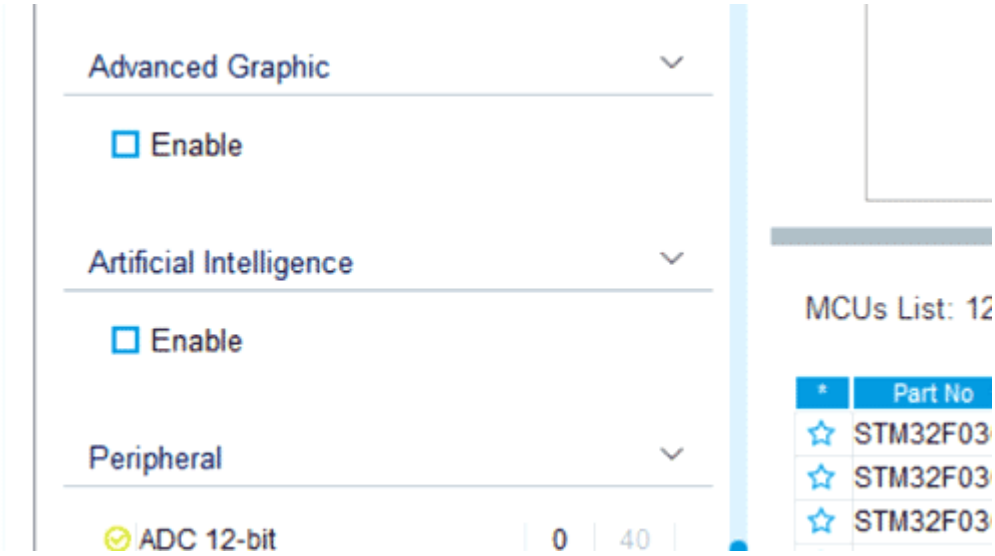

Рис. 9. Фильтр ИИ

Примечание. Данный фильтр доступен только при выборе МК (на вкладке MCU/MPUSelector), и его использование имеет смысл только в случае использования одной модели нейронной сети.

На рис. 10 показан результат работы фильтра в случае загрузки DL-модели и ее анализа с настройками по умолчанию. В примере используется общедоступная предварительно обученная модель нейронной сети (библиотека Keras): «Распознавание человеческой активности при помощи сверточной нейронной сети».

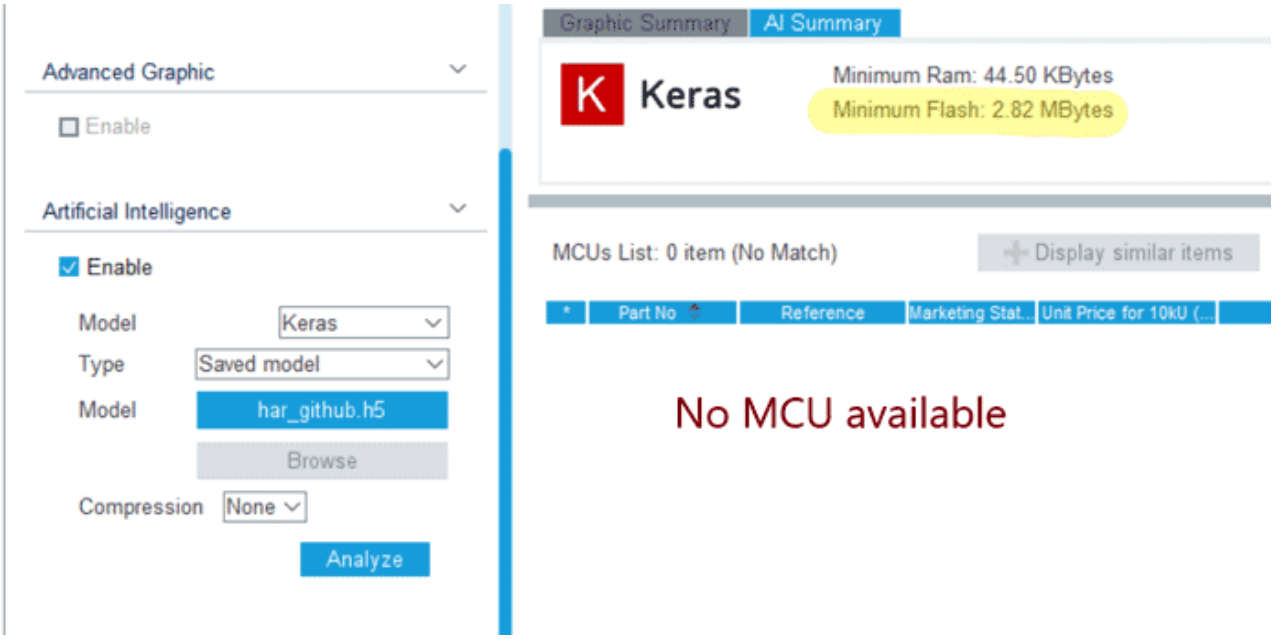

Рис. 10. Фильтр ИИ с настройками по умолчанию

На рис. 11 показан результат выбора коэффициента сжатия x4.

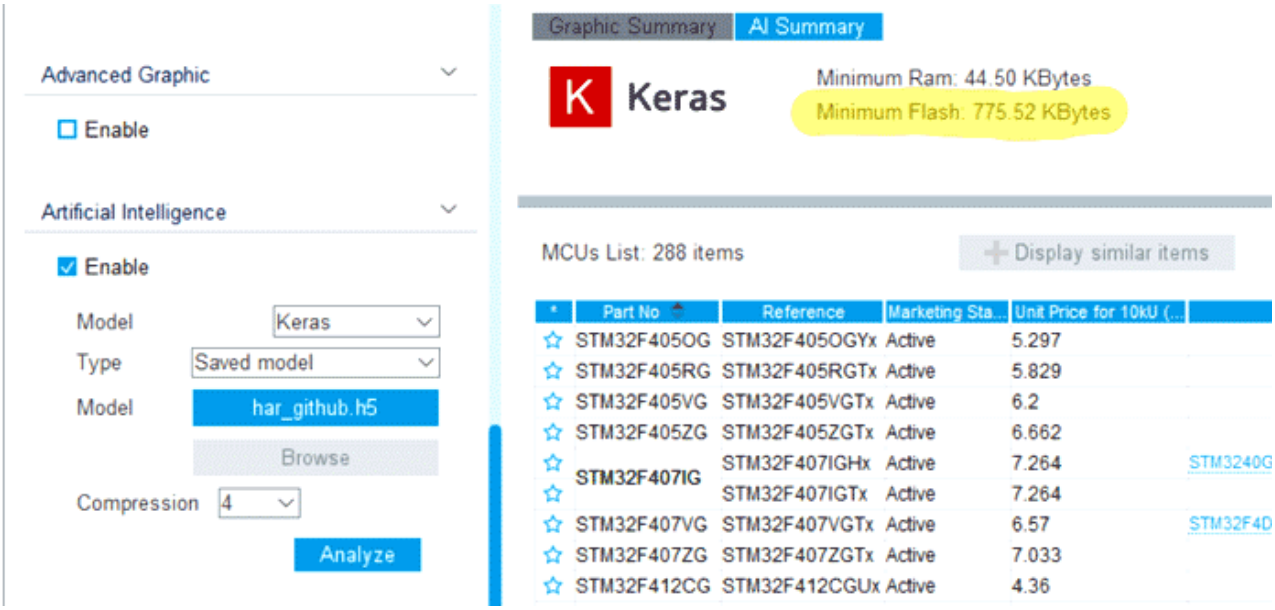

Рис. 11. Фильтр ИИ с коэффициентом сжатия x4

Примечание. Размер памяти также контролируется оптимизатором при генерации библиотеки нейронной сети, а в случае необходимости он извещает пользователя о нехватке ОЗУ и/или Flash-памяти в выбранном МК.

Для дальнейшей работы выберем отладочную плату NUCLEO-F746ZG, как показано на рис. 12.

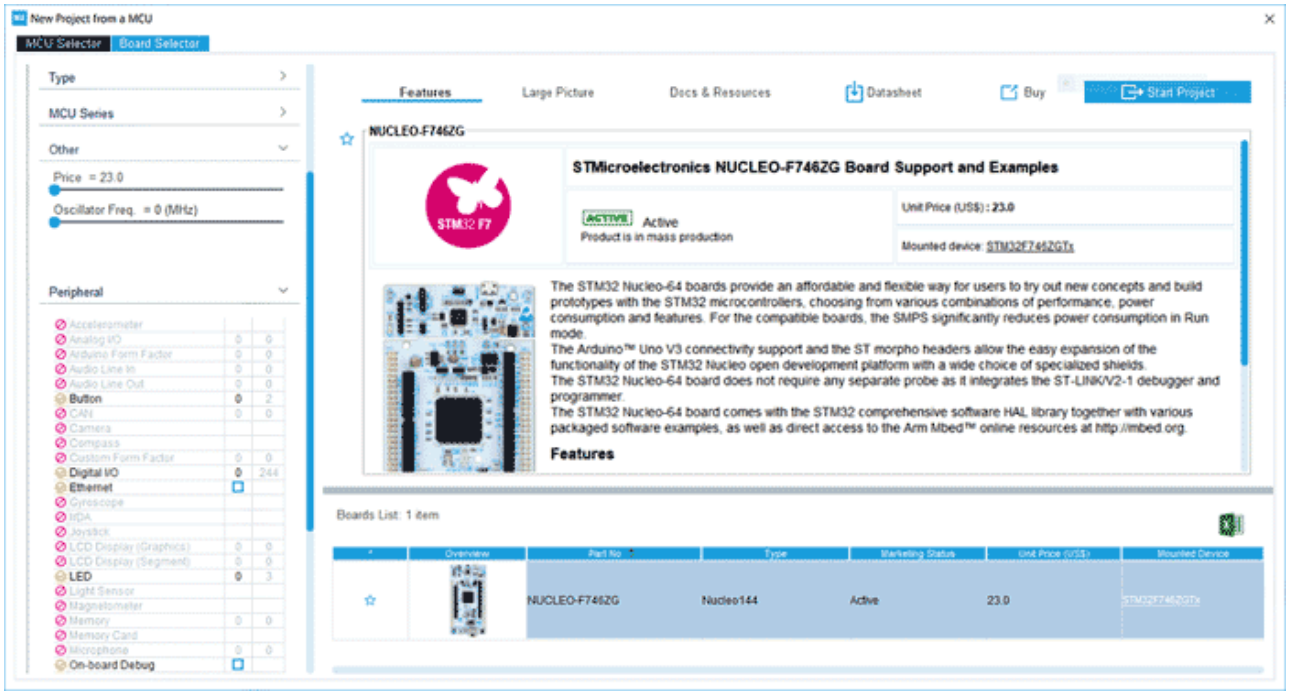

Рис. 12. Выбор платы NUCLEO-F746ZG

Для продолжения работы нажмите на кнопку «Start Project» и в появившемся диалоговом окне подтвердите инициализацию всех периферийных модулей микроконтроллера значениями по умолчанию (рис. 13).

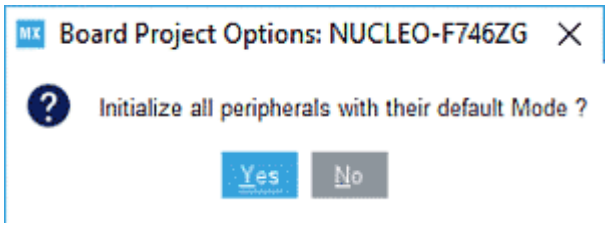

Рис. 13. Параметры инициализации всей периферии

Настройки аппаратной и программной платформ

После выбора МК или платы в основном окне STM32CubeMX появляется схематическое изображение выводов соответствующего микроконтроллера. В этом окне пользователь может настроить проект, добавляя в него дополнительные программные библиотеки и требуемые периферийные устройства, а также конфигурируя систему синхронизации МК.

Если выбрана сборка дополнительного приложения из состава пакета X-CUBE-AI (раздел «Добавление компонента X-CUBE-AI»), следует учесть, что это приложение для связи с хост-системой использует интерфейс UART. На отладочной плате NUCLEO-F746ZG выводы PD9 (TX) и PD8 (RX) микроконтроллера подключены к схеме отладчика ST-LINK для организации виртуального COM-порта (рис. 14).

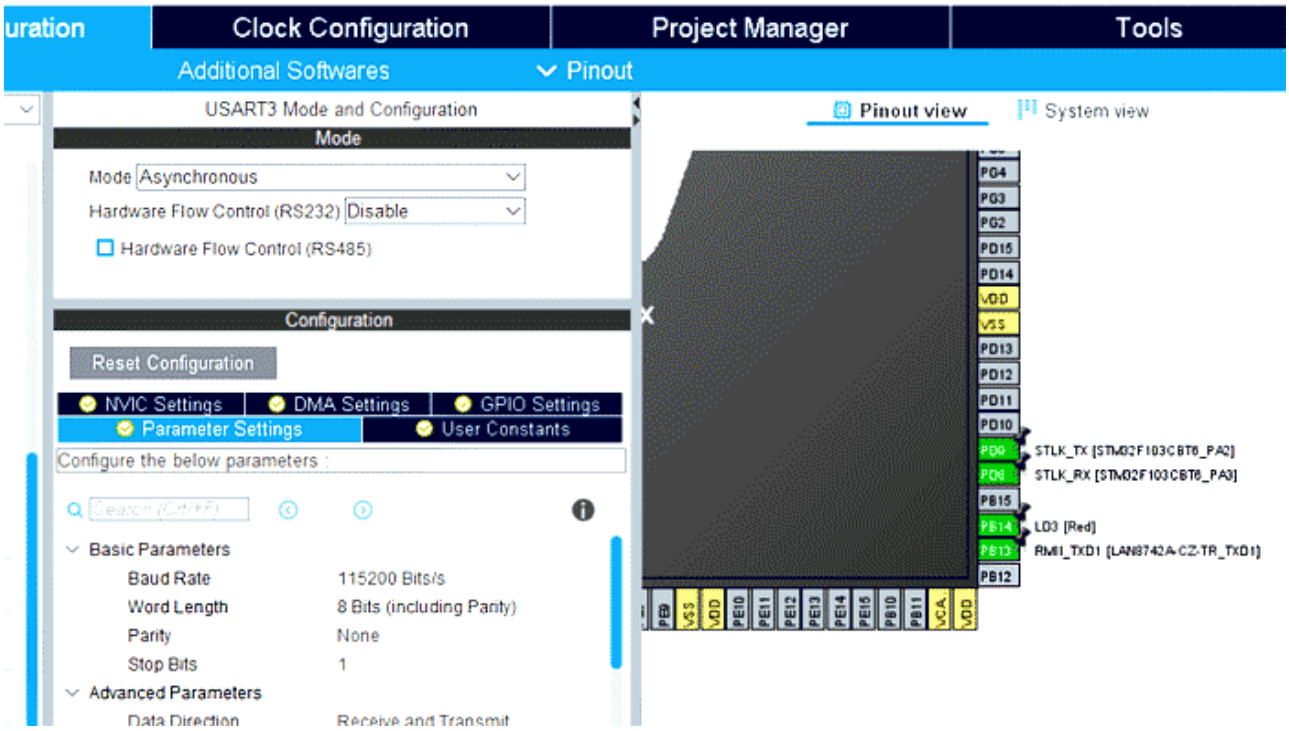

Рис. 14. Конфигурация модуля UART3

Чтобы обеспечить максимальную производительность МК, также рекомендуется сконфигурировать систему синхронизации и подсистему памяти.

Настройка частот тактового сигнала ЦПУ и системного тактового сигнала

Данная настройка осуществляется в несколько этапов:

перейдите на вкладку «Clock Configuration». По умолчанию частота системного тактового сигнала (SYSCLK, HCLK) устанавливается равной 72 МГц;
введите число 216 в поле «HCLK (MHz)», выделенное голубой рамкой (рис. 16), чтобы запустить мастер настройки системы синхронизации. Этот мастер автоматически сконфигурирует модули PLL и соответствующие тактовые сигналы. А при появлении диалогового окна, показанного на рис. 15, нажмите на кнопку «OK», чтобы продолжить работу.

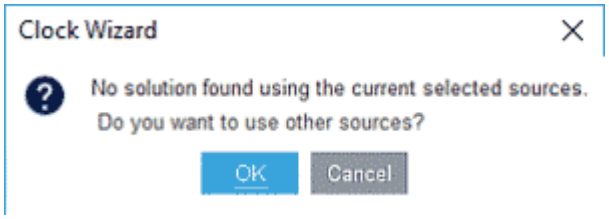

Рис. 15. Настройки системного тактового сигнала

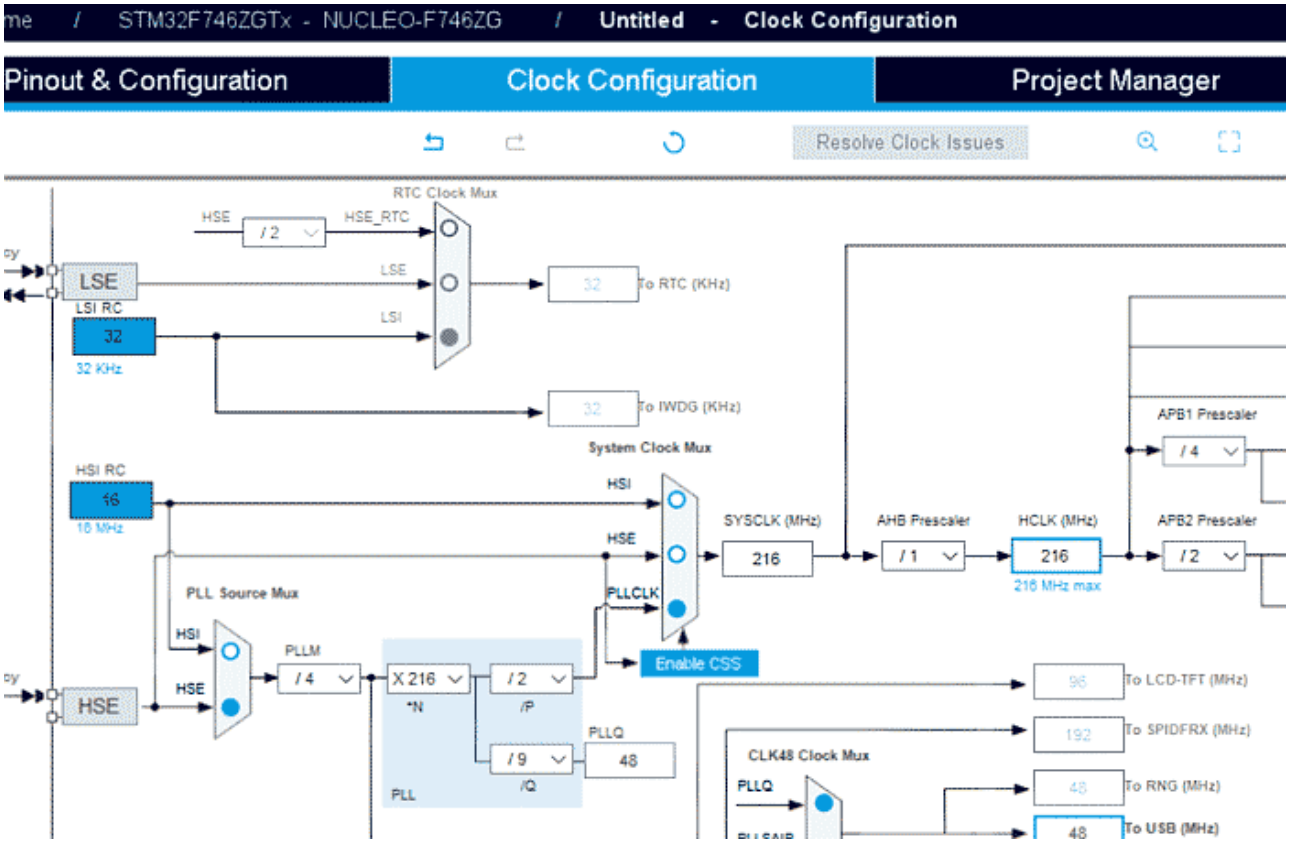

Рис. 16. Диалоговое окно мастера настройки системы синхронизации

Настройка подсистемы памяти МК

На вкладке «Pinout & Configuration» (рис. 17), выберите элемент «System Core» ? «CORTEX_M7», чтобы открыть мастер конфигурации Cortex®-M7.

Необходимо включить кэширование кода и данных, а также подсистему ускорителя ART.

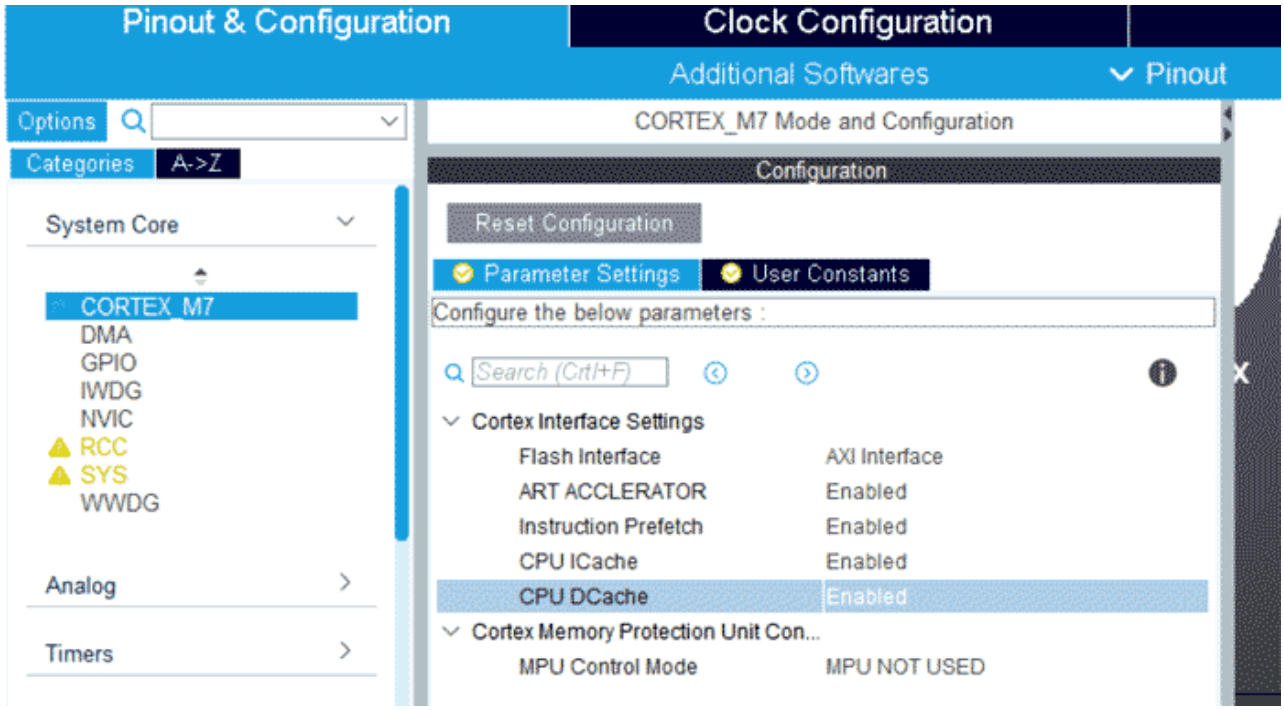

Рис. 17. Конфигурирование подсистемы памяти МК

Примечание. Не обязательно устанавливать максимально возможную частоту тактового сигнала МК. Она должна соответствовать рабочей конфигурации МК, используемой в конечном устройстве. Длительность циклов ожидания Flash-памяти будет автоматически скорректирована генератором кода STM32CubeMX.

Модуль CRC

Периферийный модуль CRC необходим для работы защищенного механизма времени исполнения библиотеки нейронной сети. Соответственно, он должен быть включен (рисунок 18).

Примечание. Данный модуль включается средой автоматически, поэтому вам не нужно делать этого вручную.

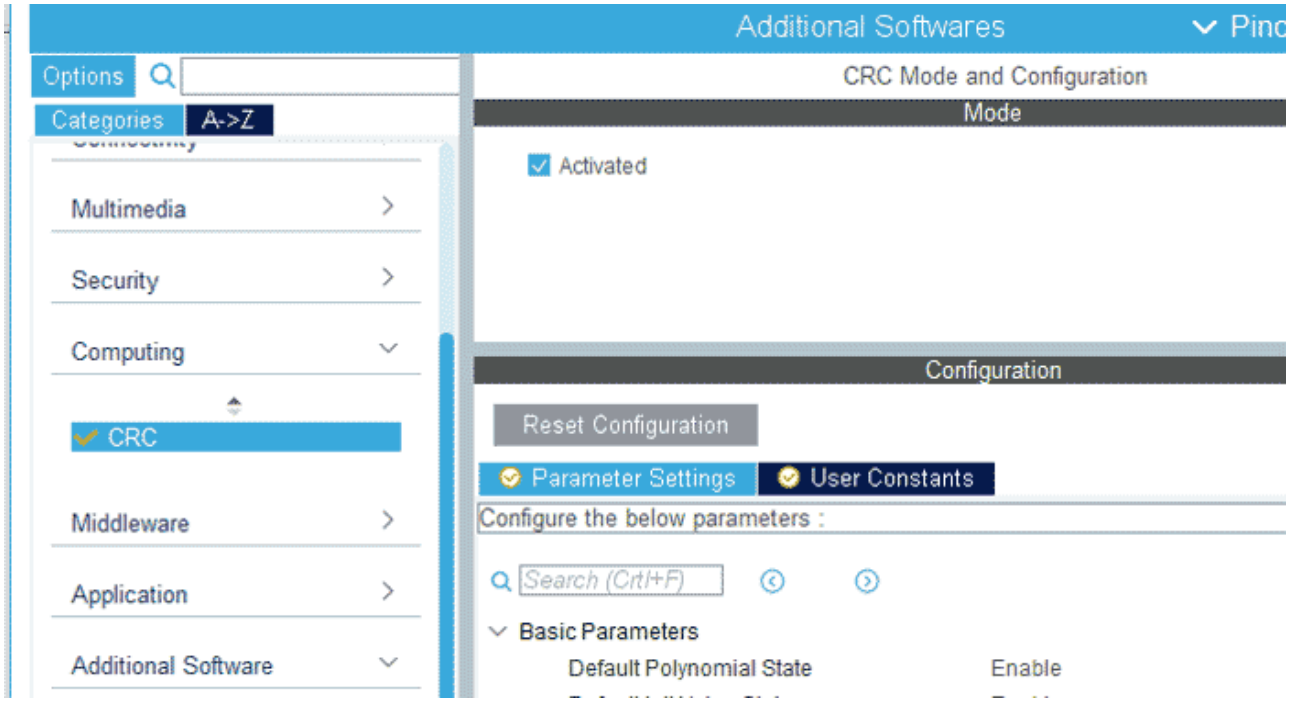

Рис. 18. Включение модуля CRC

Мастер конфигурации X-CUBE-AI

Добавляем компонент X-CUBE-AI

Добавление данного компонента происходит в несколько этапов:

Нажмите на кнопку «Additional Software», чтобы открыть окно подключения дополнительных программных модулей (рис. 19)

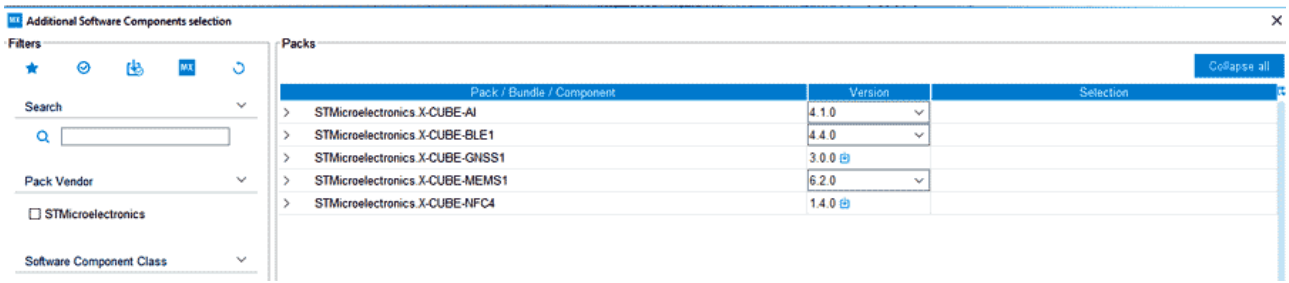

Рис. 19. Подключение дополнительных программных модулей

в открывшемся окне «Additional Software Component Selection» необходимо выставить флажок «X-CUBE-AI/Core» (рис. 20), чтобы иметь возможность загрузки моделей нейронной сети и генерации соответствующей библиотеки для STM Поскольку данная библиотека линкуется с проектом статически, пользователю необходимо реализовать только свое приложение или промежуточное ПО, используя четко определенный API нейронной сети, также генерируемый средой [6].

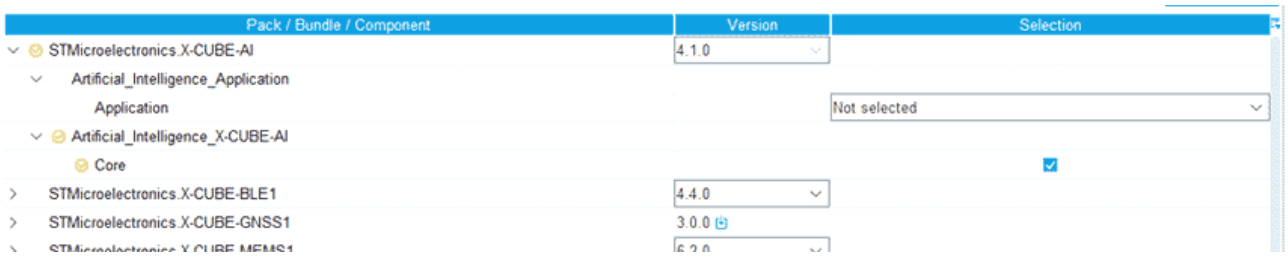

Рис. 20. Добавление ядра X-CUBE-AI

кроме того, в строке «X-CUBE-AI/Application» можно выбрать одно из дополнительных приложений X-CUBE-AI (рис. 21):

System - независимое тестовое приложение для определения производительности модели нейронной сети;

Validation - тестовое приложение для валидации нейронной сети;

Application - шаблон приложения с поддержкой ИИ.

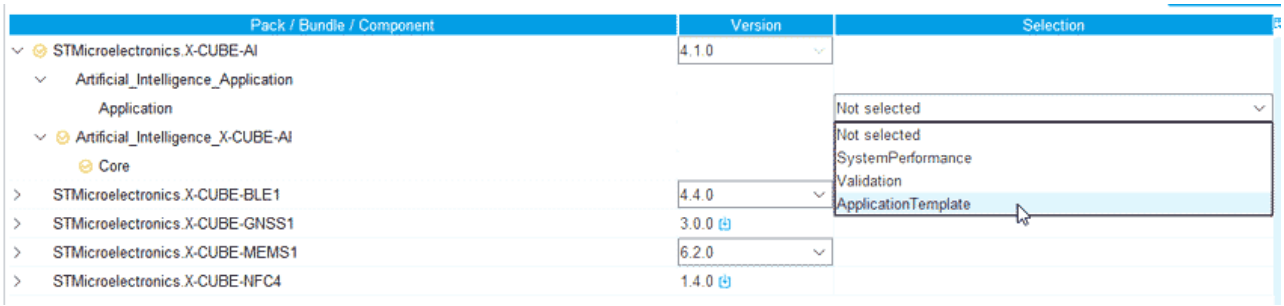

Рис. 21. Дополнительные приложения X-CUBE-AI

нажмите на кнопку «OK» для завершения выбора.

Включение компонента X-CUBE-AI

Для включения и конфигурирования компонента X-CUBE-AI необходимо выполнить следующие действия:

перейдите на вкладку «Pinout & Configuration» и раскройте элемент «Additional Software» в левой части панели, чтобы увидеть список установленных дополнительных программных пакетов. Выберите «STMicroelectronics X-CUBE-AI1.0», чтобы открыть окно начальной конфигурации подсистемы ИИ;
выставьте флажок «Artificial Intelligence Core», чтобы включить ядро X-CUBE-AI. Также выставьте флажок «Artificial Intelligence Application», если хотите включить в проект дополнительное приложение пакета ИИ. Какое именно из приложений будет включено в проект, мы указали на предыдущем шаге (рис. 22 и 23):

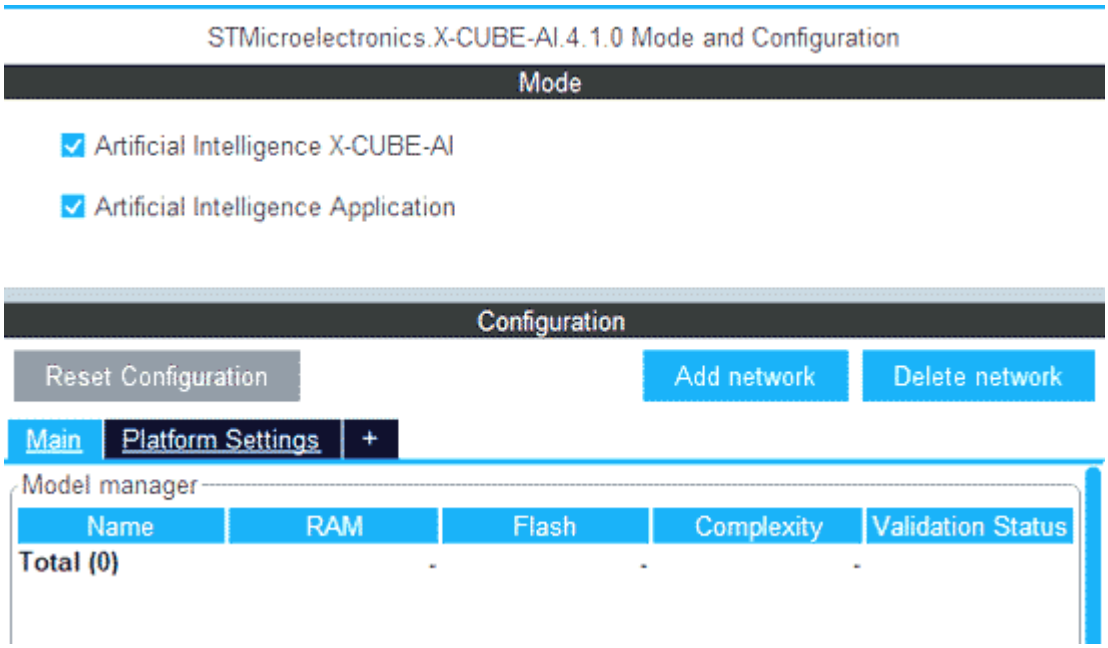

Рис. 22. Основная панель конфигурированияX-CUBE-AI

на вкладке «Main» отображается информация о загруженных нейронных сетях. Для добавления и удаления нейронной сети предназначены кнопки «Add model» и «Delete model» соответственно. Также для добавления модели можно нажать на корешок вкладки «+»;
на вкладке «Platform Settings» отображается информация о задействованном модуле USART, который используется для передачи отчета (приложение «AI System Performance») или для обмена данными с хостом (приложение «AI Validation»).

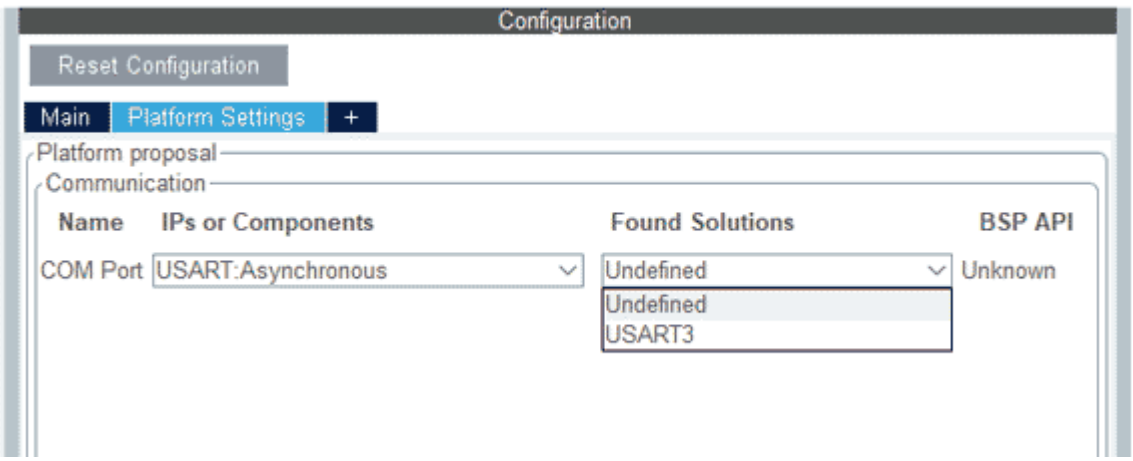

Рис. 23. Панель настроек платформы X-CUBE-AI

Загрузка файла предварительно обученной DL-модели

Нажмите на кнопку «Add model» в панели конфигурации или непосредственно на корешок вкладки «+», чтобы открыть новый мастер конфигурации модели нейронной сети. Если модель уже была загружена на этапе выбора модели МК, просто перейдите на вкладку «Network», чтобы увидеть параметры конфигурации нейронной сети (рис. 24).

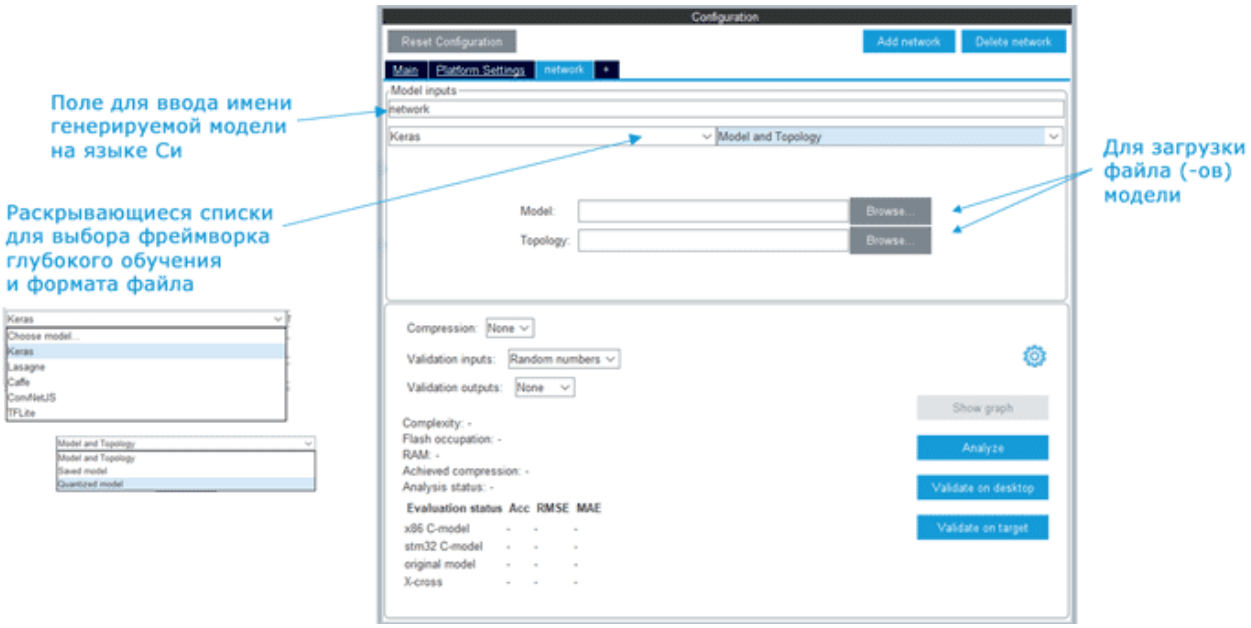

Рис. 24. Мастер конфигурации нейронной сети

Текстовое поле предназначено для задания имени сети (не более 32 символов). Эта строка используется в качестве составной части идентификаторов API нейронной сети [9]. Если в проекте будет только одна нейронная сеть, можно оставить значение этой строки, предлагаемое по умолчанию.

Раскрывающиеся списки предназначены для указания фреймворка глубокого обучения, использованного для экспорта файла DL-модели и соответствующих форматов файла (-ов), подробнее об этом – в разделе «Поддерживаемые фреймворки глубокого обучения»:

Нажмите кнопку «Browse…» для загрузки файлов DL-модели. В нашем практикуме мы будем использовать файл модели Keras (формат «Saved model»).

Нажмите на кнопку «Analyze…», чтобы запустить предварительный анализ нейронной сети для оценки возможности ее размещения в данной системе. Не забудьте перед этим выбрать коэффициент сжатия (поле «Compression»), равный 4, иначе вы увидите диалоговое окно с предупреждением, показанное на рисунке 25. Если появится диалоговое окно с сообщением «Invalid network», выберите в меню пункт «Window» ? «Outputs», чтобы открыть окно журнала и получить дополнительную информацию (раздел «Обработка ошибок»). После завершения анализа в панели мастера конфигурации будут отображены минимальный объем ОЗУ, занимаемый объем Flash-памяти и сложность исходной DL-модели, как показано на рисунке 26 (подробнее об этом будет сказано в разделе «Формирование отчета о параметрах модели»).

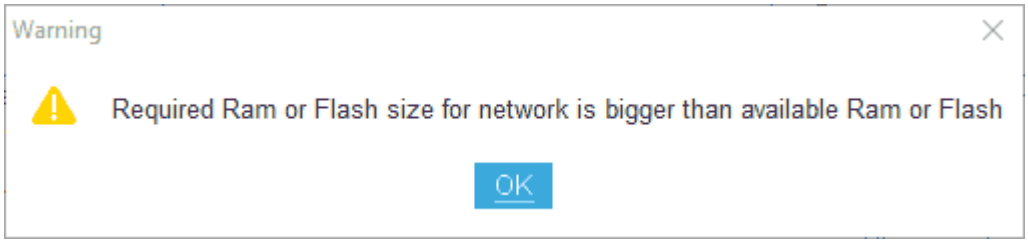

Рис. 25. Окно с сообщением о нехватке ОЗУ/Flash

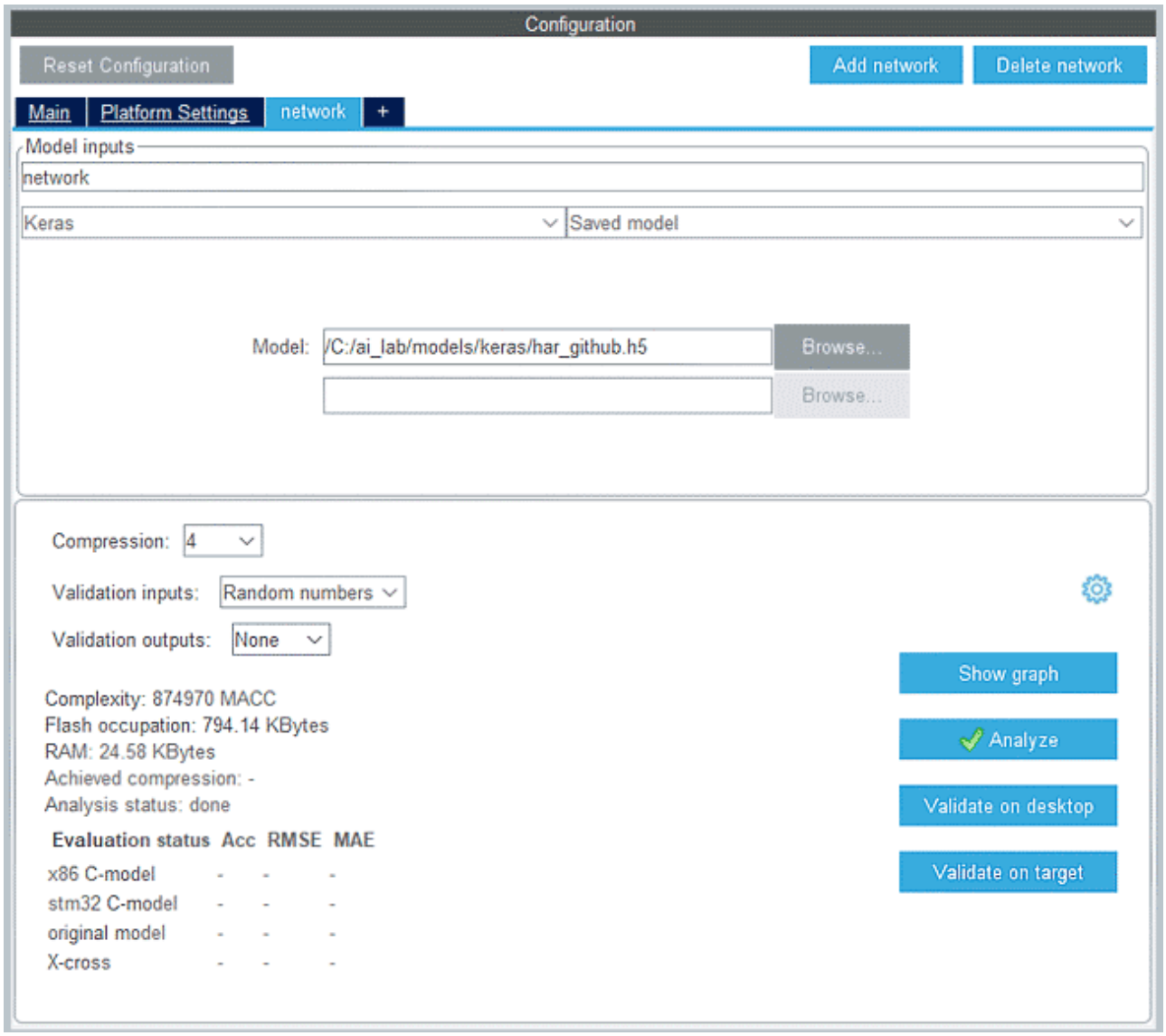

Рис. 26. Результат анализа загруженной DL-модели

Примечание. Дополнительную информацию (отладочные/информационные сообщения) можно найти в файле журнала C:\Users\<имя пользователя>\.stm32cubemx\STM32CubeMX.log или $HOME/.stm32cubemx/STM32CubeMX.log.

Нажав на кнопку расширенных настроек (пиктограмма в виде шестеренки), можно настроить использование внешней памяти (внешняя Flash-память может использоваться для хранения значений весов, а внешнее ОЗУ – для хранения значений весов и активации).

Если для проекта была выбрана отладочная плата STMicroelectronics с установленной внешней Flash-памятью/ОЗУ, то конфигурирование внешней Flash-памяти или ОЗУ

производится автоматически во время генерации кода. Для корректной инициализации Flash-памяти или ОЗУ используются функции BSP, предоставляемого пакетом поддержки МК STM32Cube (рис. 27). Внешняя Flash-память используется в режиме отображения на общее адресное пространство (memory-mapped mode).

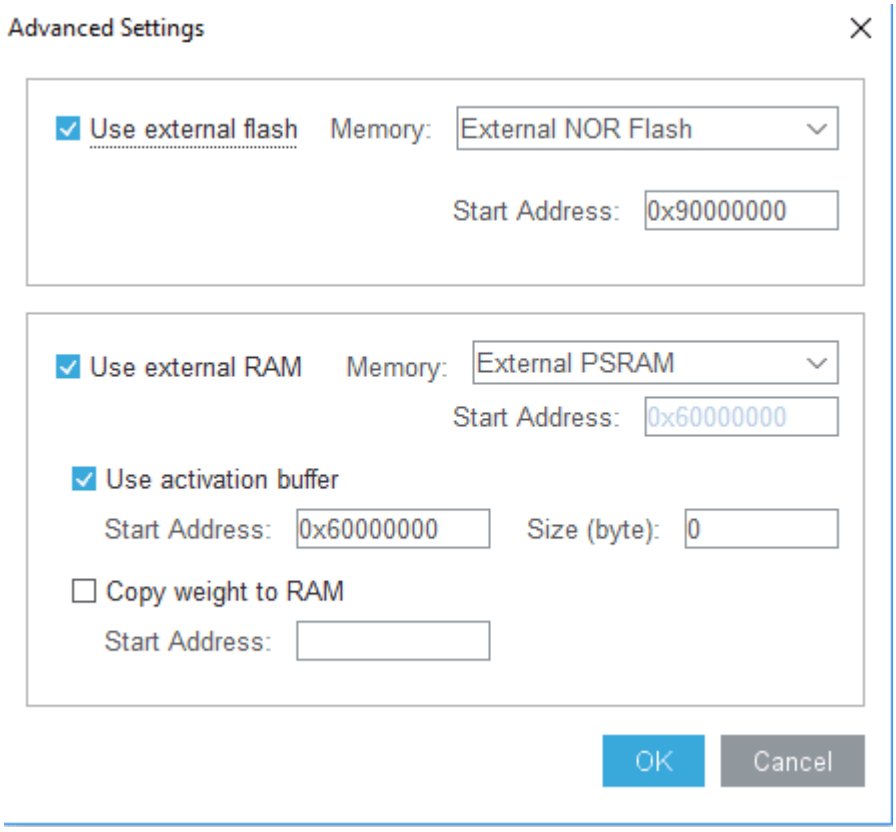

Рис. 27. Настройки для внешней памяти

Если поставлен флажок «Use external flash», то значения весов сохраняются в отдельный файл network_data.bin и генерируется код, указывающий на начальный адрес области отображения внешней Flash-памяти.

Файл network_data.bin необходимо вручную записать во внешнюю Flash-память, расположенную на плате, при помощи соответствующего инструментария, например, программатора STM32CubeProgrammer (STM32CubeProg).

Примечание. При включении автоматической валидации на целевой плате файл network_data.bin будет записан во внешнюю Flash-память автоматически.

Флажок «Use activation buffer» отвечает за размещение буферов значений активации во внешнем ОЗУ, начиная с адреса, указанного в поле «Start Address».

При желании можно настроить копирование значений весов во внешнюю память при старте прошивки (флажок «Copy weight to RAM»). В этом случае необходимо указать начальный адрес области памяти, куда будут скопированы эти значения.

При использовании внешней Flash-памяти или внешнего ОЗУ на микроконтроллерах STM32 с ядром ARM® Cortex®-M7 автоматически включается кэш команд (ICache) и данных (DCache). Также автоматически конфигурируется модуль защиты памяти для разрешения доступа к внешней памяти.

Формирование отчета о параметрах модели

После завершения анализа DL-модели пользователю становятся доступны значения параметров, указанных в таблице 2 и проиллюстрированных на рисунке 28.

Таблица 2. Параметры модели

Параметр	Описание
RAM	Показывает размер (в байтах) блока RW-памяти, используемого для хранения промежуточных значений в процессе формирования логического вывода (секция .data или .bss)
ROM/Flash	Показывает размер (в байтах) сгенерированного блока RO-памяти, содержащего значений весов/смещений после (возможного) сжатия (секция .rodata)
Complexity	Показывает функциональную сложность импортированной DL-модели, выраженную в количестве операций умножения с накоплением (MACC). В это значение также входит приблизительная оценка сложности функций активации, выражаемой в тех же единицах.

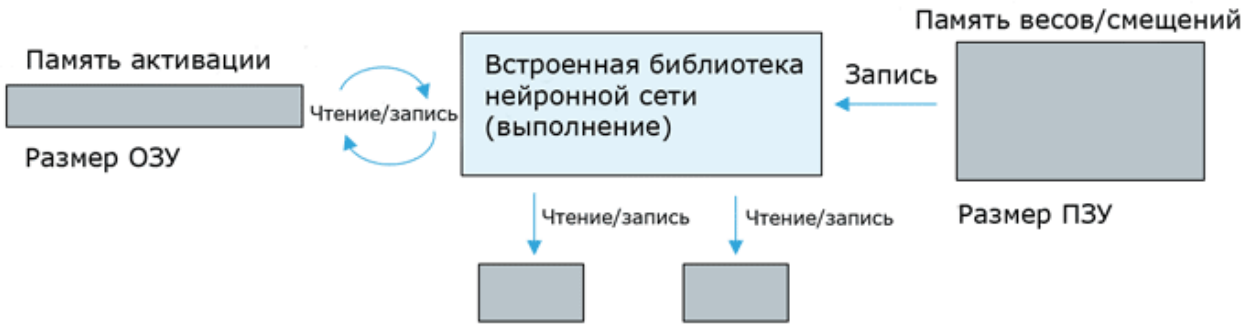

Рис. 28. Встроенная модель на языке Си (потоки данных)

Примечание. Минимальные размеры ОЗУ и Flash-памяти, указанные в отчете, не учитывают память, используемую пользовательским приложением (включая ОЗУ, необходимое для хранения входных и выходных тензоров). Данные параметры отражают только память, необходимую модели для хранения значений весов/смещений и активации. Также не учитывается память, необходимая для хранения кода, реализующего функции нейронной сети, и память под стек/кучу.

MACC – сколько это в тактах ЦПУ?

К сожалению, между сложностью модели, указанной в отчете, и реальной производительностью сгенерированной библиотеки нейронной сети нет прямой зависимости (такты ЦПУ/MACC). В связи с большим числом факторов, влияющих на процесс генерации библиотеки (в том числе используемый набор инструментальных средств, настройки МК и его подсистемы памяти, топология нейронной сети и ее слоев, примененные опции

оптимизации), довольно сложно заочно определить точную зависимость отношения «такты ЦПУ/MACC» от конкретных настроек STM32. Тем не менее, для грубой оценки можно использовать следующие значения (для модели, использующей 32-битные значения с плавающей запятой):

STM32 ARM® Cortex®-M4: ~ 9 тактов/MACC;
STM32 ARM® Cortex®-M7: ~ 6 тактов/MACC.
Для определения реальной производительности нейронной сети на целевой плате было подготовлено специальное тестовое приложение «AI System Performance» (более подробно об этом читайте в разделе «Приложение AI System Performance»).

Графическое представление сгенерированной модели на языке Си

Нажмите на кнопку «Showgraph»], чтобы увидеть основную информацию о структуре загруженной модели, которая была использована генератором Си-кода (рисунок 29). Данная структура показывает внутреннее представление загруженной DL-модели перед ее оптимизацией.

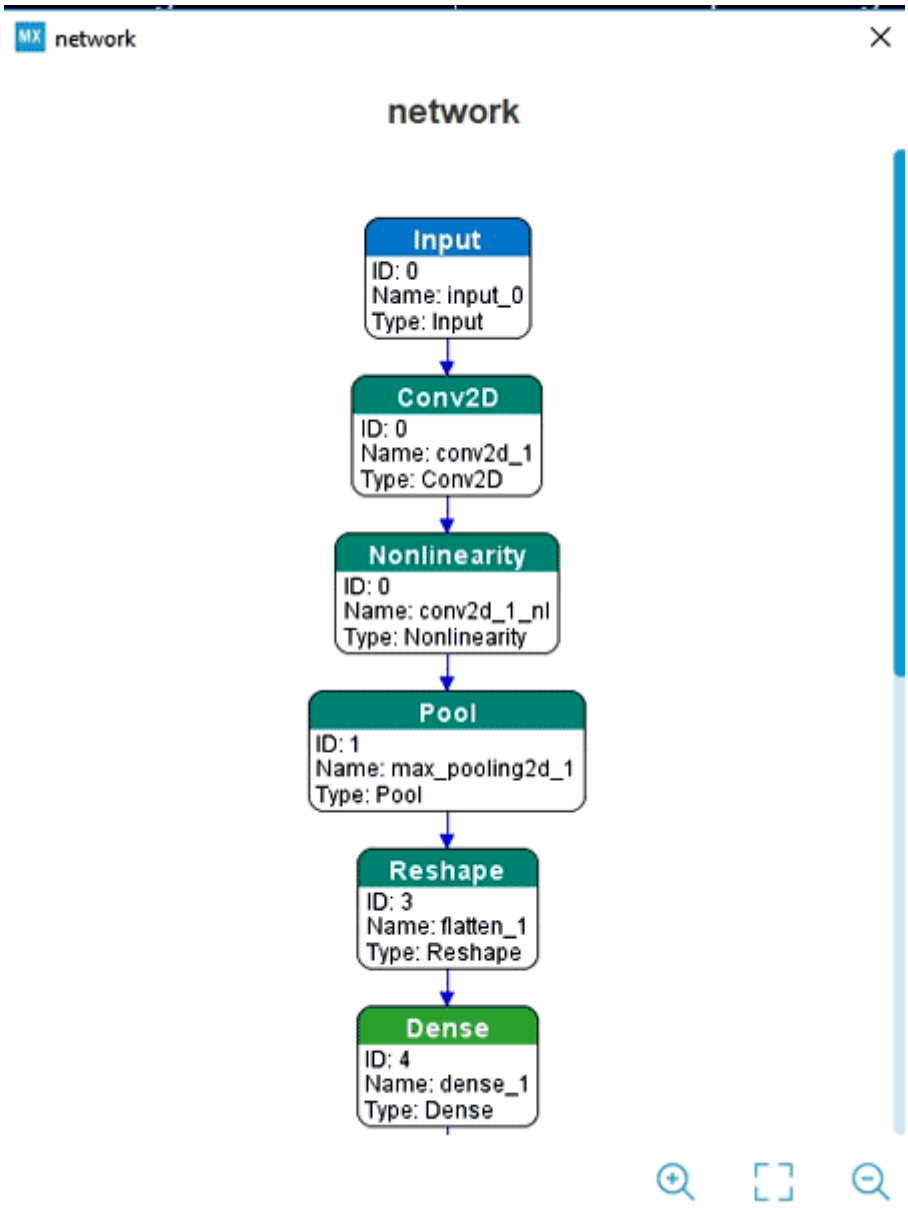

Рис. 29. Граф генерируемой модели на языке Си

Валидация сгенерированной модели на языке Си

Чтобы запустить процесс валидации сгенерированной модели на языке Си, нажмите кнопку «Validate on desktop» (рисунок 30). При отсутствии входных данных в качестве критерия точности работы модели в большинстве случаев используется значение относительной ошибки вычисления L2 (раздел «Механизм валидации (вычисление относительной ошибки L2)»). Обратите внимание, что выполнение данного этапа не обязательно, но крайне рекомендуется, особенно если при генерации модели было использовано сжатие (раздел «Оптимизатор потока выполнения и занимаемой памяти»).

Если же в вашем распоряжении имеются образцовые (эталонные) выходные значения с соответствующими им входными данными, то для вычисления метрик, указанных в таблице 3, используются уже сами расчетные значения [7].

Таблица 3. Метрики

Метрика	Описание
ACC	Точность классификации
RMSE	Корень среднеквадратичной ошибки
MAE	Средняя абсолютная ошибка

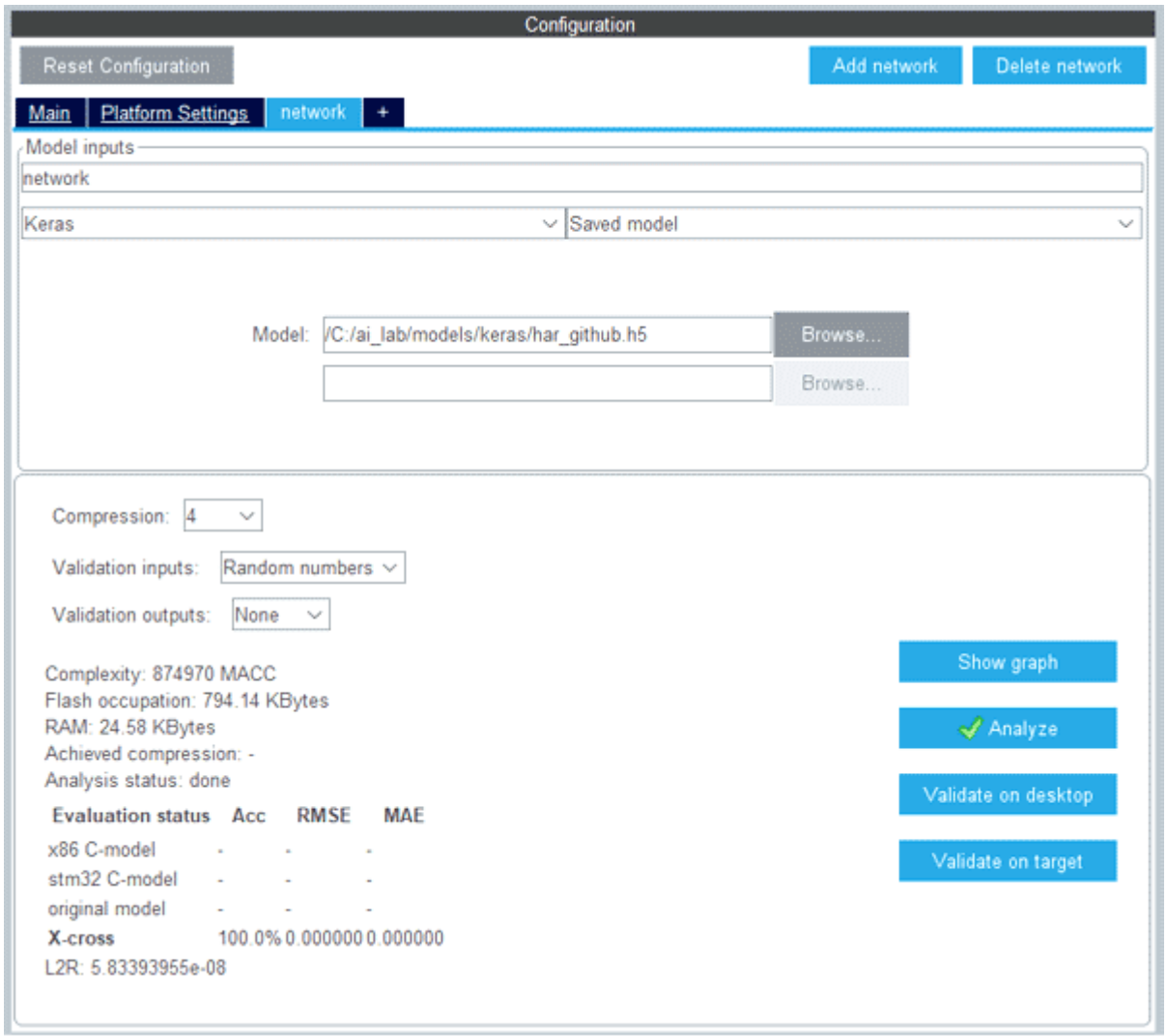

Рис. 30. Результат валидации модели

Более подробная информация выводится в виде текстового протокола в окне журнала, как показано на рис. 31. В этом журнале, в частности, для каждого сгенерированного слоя указывается ошибка L2R относительно оригинального слоя.

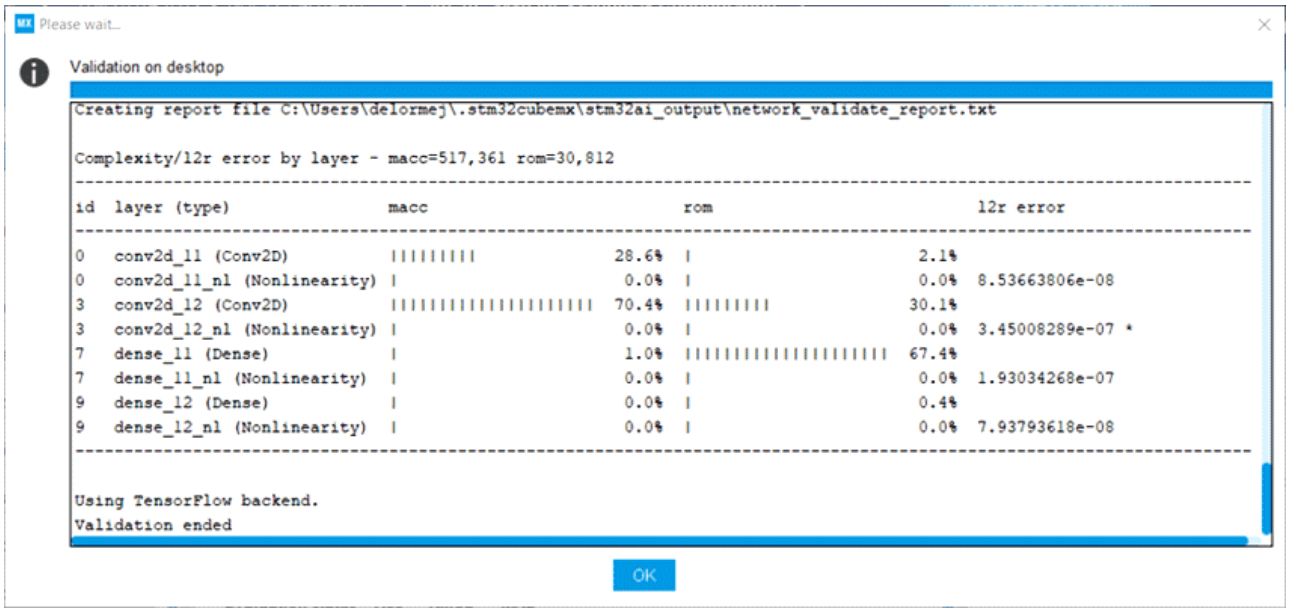

Рис. 31. Валидация на ПК (протокол)

Примечание. Начиная с этого момента, загруженную DL-модель можно интегрировать в генерируемый IDE-проект.

Валидацию на целевой платформе (кнопка «Validate on target») можно выполнить только после загрузки в целевой МК специального тестового приложения «AI Validation». Сборку этого приложения необходимо указывать при конфигурировании пакета X-CUBE-AI (раздел «Добавляем компонент X-CUBE-AI»). Информация, выводимая этим приложением, и его использование подробно описаны в разделе «Приложение AI Validation».

Кнопка «Validate on target» позволяет пользователю выполнить валидацию модели нейронной сети на целевой плате, а также (опционально) автоматически генерировать, компилировать, загружать и запускать временный проект, соответствующий текущей нейронной сети.

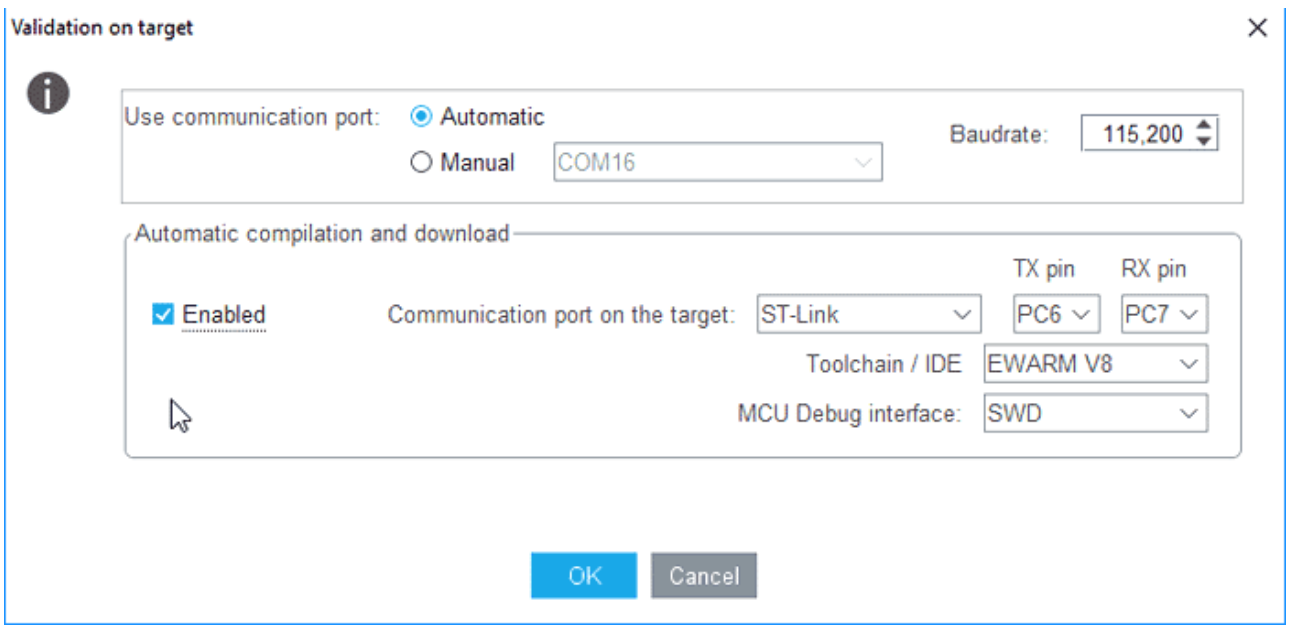

Рис. 32. Валидация на целевой платформе

Для того чтобы автоматически скомпилировать программу, загрузить ее в МК и запустить на выполнение, убедитесь, что используемый коммуникационный порт на целевой плате соответствует модулю USART (UART, LPUART), подключенному к отладчику ST-LINK для организации виртуального COM-порта.

При необходимости можно вручную задать используемый для связи периферийный модуль и указать выводы микроконтроллера, используемые для передачи и приема данных.

Значение поля «Toolchain/IDE» следует установить в соответствии с используемым набором инструментальных средств.

Если для программирования МК используется интерфейс JTAG, то необходимо изменить отладочный интерфейс, предлагаемый по умолчанию (поле «MCU Debug interface»).

При нажатии кнопки «OK» временный проект генерируется, загружается в микроконтроллер и запускается на исполнение. После этого выполняется обычная валидация нейронной сети.

Добавление новой DL-модели

Можно импортировать несколько DL-моделей. Мастер сам по себе никак не ограничивает их максимальное количество — оно определяется, главным образом, размерами ОЗУ и Flash-памяти выбранного микроконтроллера. Чтобы импортировать новую DL-модель, нажмите на корешок вкладки «+» и выполните все шаги, описанные выше. Суммарный объем ОЗУ и Flash-памяти, необходимый всем загруженным моделям, будет отображен на вкладке «Main» (рис. 33).

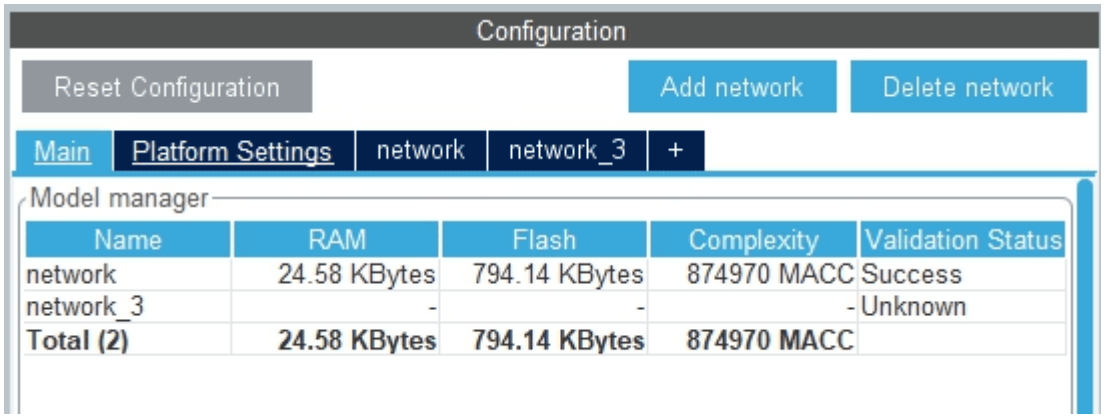

Рис. 33. Вкладка «Main» при загрузке нескольких моделей сетей

Источник:

Автор: ST Microelectronics Переводчик: Андрей Евстифеев (г. Королев)
Производители:

Разделы: , ,

Опубликовано: 03.12.2019