

УДК 372.800.4

УРОК ИНФОРМАТИКИ В 10-М КЛАССЕ «МОБИЛЬНЫЕ УСТРОЙСТВА» (КАК ПРИМЕР ВНЕ- ДРЕНИЯ ЦИФРОВОГО КОНТЕНТА)

Аннотация. В статье представлен конспект урока по теме «Мобильные устройства» для учащихся 10-го класса. Материал охватывает изучение архитектуры мобильных систем, в частности систем на кристалле (SoC), и работу мобильных операционных систем. Принцип урока основан на использовании цифровых образовательных ресурсов в формате лонгрида, что способствует развитию навыков самостоятельной учебной деятельности. В практической части предлагаются интерактивные задания и кейсы, направленные на анализ аппаратного обеспечения и анализ рисков информационной безопасности. Урок спроектирован с учетом возможности использования как персональных компьютеров, так и личных мобильных устройств учащихся.

Ключевые слова: мобильные устройства, микропроцессорная подсистема, система на кристалле (SoC), информационная безопасность, системное программное обеспечение.

Введение. В условиях повсеместной цифровизации мобильные устройства перестали быть просто средством связи, став многофункциональными вычислительными системами. Для старшеклассников, стоящих на пороге выбора будущей профессии, важно разобраться в принципах функционирования современных



**Надежда Николаевна
Самылкина,**
*доктор педагогических наук,
доцент,
профессор кафедры тео-
рии и методики обучения
информатике
Института математики и
информатики,
Московский педагогиче-
ский государственный
университет,
г. Москва, Россия
E-mail: nsamylkina@yandex.ru*

Как цитировать статью: Самылкина Н. Н., Шерстякова Е. С. Урок информатики в 10-м классе «Мобильные устройства» (как пример внедрения цифрового контента) // Образ действия. 2026. Вып. 3 «Естественно-научное образование (лучшие практики)». С. 91–103.

систем. Однако многие из них не осознают всей сложности его внутренней структуры и потенциальных угроз безопасности.

Актуальность выбранной темы статьи обусловлена необходимостью перехода от потребительского подхода к технологиям к инженерно-аналитическому. В рамках школьного курса информатики изучение мобильных устройств позволяет наглядно продемонстрировать принципы конвергенции технологий и открытой архитектуры практически всех информационных систем.

Целью урока является формирование у учащихся системного представления об аппаратной и программной организации мобильных устройств, а также развитие навыков критического анализа информационных угроз.

Для достижения поставленной цели решаются следующие задачи:

- изучение архитектуры «система на кристалле» (SoC) и отличительных особенностей процессоров [2, с. 10–15];
- разбор уровней программного обеспечения [2, с. 18–20];
- практическое освоение методов оценки рисков при установке стороннего ПО [2, с. 45–47].

Методика проведения занятия базируется на использовании интерактивного сайта-лонгрида и решении практических кейсов, что позволяет визуализировать абстрактные понятия микроэлектроники.

Центральной темой урока является изучение структуры мобильных устройств. В отличие от классической архитектуры ПК, где компоненты разнесены по материнской плате, система на кристалле, используемая в мобильных устройствах, интегрирует в одном кристалле ключевые узлы, что критично для энергопотребления и компактности [2, с. 17].

1. Организационный этап

Изучение нового материала и выполне-



**Екатерина Сергеевна
Шерстякова,**

*Институт математики
и информатики,
Московский педагогиче-
ский государственный
университет,*

г. Москва, Россия

E-mail: eksherstyakova@mail.ru

ние заданий организовано в формате самостоятельной работы с образовательным сайтом (лонгридом), который доступен по ссылке: <https://lesson-mobile-devices.tilda.ws>. Реализация сайта размещена на рисунках 1–9. Ученики могут открыть этот сайт на стационарных компьютерах в классе либо с помощью личных мобильных телефонов. При выборе второго варианта учителю необходимо заранее согласовать использование смартфонов на уроке с администрацией школы.

Оценивание выстроено следующим образом:

Отметка «3» выставляется за выполнение базового задания «Программная иерархия» (рис. 10).

Отметка «4» выставляется за выполнение одной из следующих пар заданий: «Программная иерархия» и «Служба безопасности», либо «Программная иерархия» и «Конструктор инноваций», либо «Служба безопасности» и «Конструктор инноваций» (рис. 10–12).

Отметка «5» выставляется за успешное выполнение трех заданий: «Конструктор инноваций» и «Служба безопасности» и «Программная иерархия» (рис. 10–12).

Задание «Мобильные устройства внутри» вынесено как дополнительное для желающих получить экстрабаллы или углубить знания (рис. 13).

2. Мотивационный этап

Учитель актуализирует тему через создание проблемной ситуации: смартфоны есть у каждого, но большинство пользователей не представляет, какие сложные процессы происходят внутри гаджета каждую секунду. Цель урока — перейти от бытового восприятия устройства к пониманию его архитектуры и научиться осознанно защищать свои личные данные от киберугроз.

3. Этап изучения нового материала

На этом этапе учащиеся индивидуально и в собственном темпе работают с сайтом-лонгридом (рис. 1–13) [3]. Роль учителя трансформируется в позицию наставника-координатора: он направляет познавательную деятельность и помогает разобраться со сложными терминами. Изучение делится на три вектора:

- **Архитектурная интеграция (SoC).** С помощью интерактивных схем учащиеся анализируют устройство «системы на кристалле», изучая специализацию CPU, GPU, NPU (нейропроцессора для ИИ) и дополнительных модулей (ISP, DSP). Данный подход иллюстрирует принципы организации современных вычислительных систем и реализацию параллельных вычислений на мобильных платформах [2, с. 14–15].

- **Иерархия системного ПО.** Изучается программная вертикаль — от ядра (Kernel) до пользовательского интерфейса, а также формируется понимание операционной системы как супервизора, управляющего распределением ресурсов [2, с. 18–20]. Учащиеся знакомятся с классификацией

программного обеспечения и функциями мобильных ОС [1, с. 83–84].

– **Концепция Sandboxing.** Изучается механизм «песочницы», обеспечивающий программную изоляцию приложений, что закладывает основу информационной безопасности [2, с. 46–47].

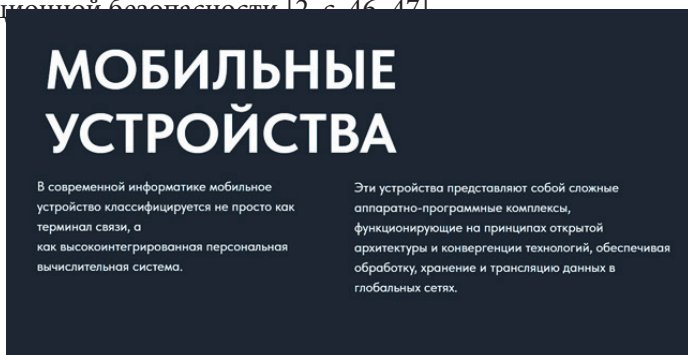
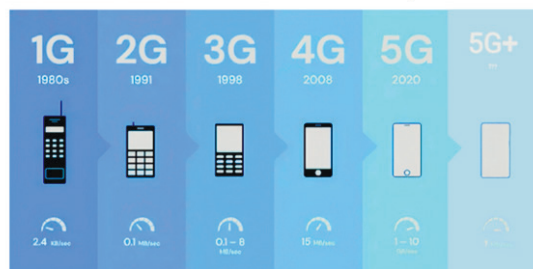


Рисунок 1. Раздел сайта «Мобильные устройства»

Ретроспективный анализ и технологическая конвергенция



Эволюция мобильных систем демонстрирует переход от узкоспециализированных устройств к универсальным вычислительным машинам (полным по Тьюрингу).

- **Аналоговая и ранняя цифровая эпохи (1G – 2G):** Мобильные телефоны функционировали преимущественно как радиотерминалы. Внедрение сетей 2G обеспечило переход к цифровой коммутации и пакетной передаче данных (SMS, ранний мобильный интернет).
- **Эпоха карманных персональных компьютеров (КПК):** Внедрение первых мобильных операционных систем (Windows Mobile, Symbian), позволивших исполнять сторонний скомпилированный код.
- **Эволюция пользовательских интерфейсов (post-2007):** Переход к емкостным сенсорным экранам и алгоритмам Multi-Touch. Взаимодействие стало базироваться на прямом манипулировании графическими объектами.
- **Современный этап:** Технологическая конвергенция достигла пика. Смартфон аккумулирует функции десятков автономных приборов, выступая ключевым узлом в концепции **Интернета вещей (IoT)** и периферийных вычислений (Edge Computing), управляя сенсорными сетями и системами «умного дома».

Что это такое?

Что внутри мобильных устройств?

Рисунок 2. Раздел сайта «Ретроспективный анализ и технологическая конвергенция»

Аппаратная архитектура (Hardware): уровень интеграции

В отличие от классической архитектуры фон Неймана, применяемой в настольных ПК, мобильные устройства строятся на базе сверхвысокой интеграции компонентов для минимизации энергопотребления и тепловыделения.

Микропроцессорная подсистема (SoC – System-on-a-Chip)

«Система на кристалле» представляет собой единый кремниевый кристалл, на котором размещены ключевые вычислительные блоки (чаще всего на базе архитектуры ARM)

«O SoC»

- **Центральный процессор (CPU):** Выполняет общие вычислительные задачи. Обычно используется кластерная архитектура (например, big.LITTLE), где мощные ядра активируются для выполнения ресурсоемких задач, а энергоэффективные ядра — для фоновых процессов.
- **Графический процессор (GPU):** Предназначен для рендеринга графического интерфейса и трехмерной графики.
- **Нейропроцессор (NPU):** Специализированный процессор для аппаратного ускорения алгоритмов машинного обучения, компьютерного зрения и обработки естественного языка.
- **Процессоры цифровой обработки сигналов (DSP) и изображений (ISP):** Обеспечивают обработку данных с камер и микрофонов в реальном времени.
- **Радиомодули (Baseband):** Реализуют аппаратную часть сетевых протоколов (Wi-Fi, Bluetooth, 4G/LTE, 5G).



Иерархия памяти

- **ОЗУ (RAM):** Энергозависимая память типа LPDDR (Low Power Double Data Rate). Хранит исполняемый код и данные запущенных процессов.
- **ПЗУ (Flash ROM):** Энергонезависимая твердотельная память (стандарты UFS/eMMC) для долговременного хранения файловой системы, ядра операционной системы и пользовательских данных.



Периферия и микроэлектромеханические системы (MEMS)

Современный смартфон оснащен массивом датчиков, созданных по технологии MEMS:

- **Акселерометр и гироскоп:** Измеряют линейное ускорение и угловую скорость, обеспечивая пространственную ориентацию устройства.
- **Оптические сенсоры:** CMOS-матрицы камер, датчики освещенности и LiDAR (лазерный дальномер для построения карты глубины).
- **Биометрические сканеры:** Тактикоскопические датчики и модули инфракрасного сканирования лица.

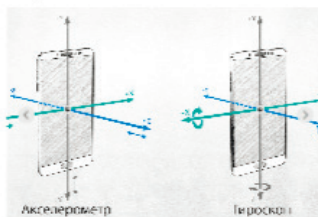


Рисунок 3. Раздел сайта «Аппаратная архитектура»

Системное программное обеспечение (Software)

Программное обеспечение мобильных устройств имеет строгую иерархическую структуру, которая изолирует аппаратные компоненты от приложений.

Операционная система и слой абстракций (HAL)

ОС выполняет роль супервизора, управляя распределением процессорного времени и памяти. Взаимодействие ядра ОС с аппаратными узлами осуществляется через слой аппаратных абстракций (HAL) и драйверы.

- **Android:** Открытая платформа на базе модифицированного ядра Linux. Использует виртуальные машины (ART) для компиляции и исполнения кода приложений.
- **iOS:** Проприетарная UNIX-подобная операционная система с закрытым исходным кодом, оптимизированная под конкретное аппаратное обеспечение Apple.
- **Аврора:** Отечественная мобильная ОС, разрабатываемая в рамках стратегии технологического суверенитета. Базируется на принципах доверенной среды загрузки и Мандатного управления доступом (MAC), что критично для корпоративного и государственного секторов.

Информационная безопасность и криптография

Мобильные ОС используют принцип «песочницы» (Sandboxing) — каждое приложение изолировано в собственном адресном пространстве памяти и не имеет прямого доступа к файлам других программ без явного разрешения пользователя.

- **Дистрибуция и верификация кода:** Загрузка прикладного ПО из официальных репозиториях (App Store, Google Play, RuStore) подразумевает предварительную проверку кода на вредоносные сигнатуры.
- **Несанкционированная установка:** Ручная установка пакетов (например, .apk архивов) нарушает цепочку доверия. Операционная система идентифицирует такие действия как угрозу и требует повышения привилегий доступа (административных прав) для продолжения установки.

"O Sandbox"

Рисунок 4. Раздел сайта «Системное программное обеспечение»

В конце сайта более подробно расписаны понятия «ретроспективный анализ», «технологическая конвергенция», «песочница» и «система на кристалле». Рисунки 5-8.

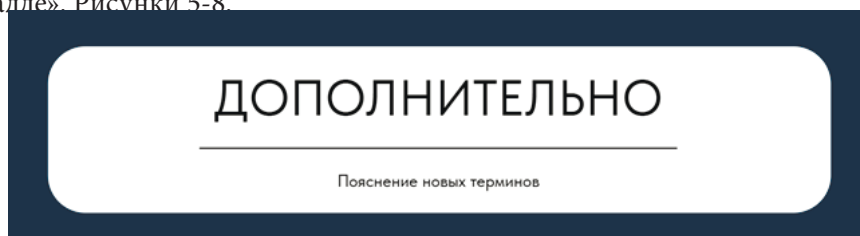


Рисунок 5. Раздел сайта «Дополнительно»

Что такое ретроспективный анализ и технологическая конвергенция

Ретроспективный анализ —

это рассмотрение эволюции мобильных систем, переход от узкоспециализированных устройств к универсальным вычислительным машинам. Он включает анализ разных этапов развития: от аналоговой и ранней цифровой эпох (1G–2G) до современных технологий.

Технологическая конвергенция —

это процесс, при котором смартфон аккумулирует функции десятков автономных приборов, выступая ключевым узлом в концепции интернета вещей (IoT) и периферийных вычислений (Edge Computing). В рамках урока подчеркивается, что современные мобильные устройства — это высокоинтегрированные системы, объединяющие различные технологии и функции.

Рисунок 6. Раздел сайта «Дополнительно. Что такое ретроспективный анализ и технологическая конвергенция»

Что такое SoC (система на кристалле)?

Устройство SoC (англ. System on a Chip) — в русской аббревиатуре СнК — система на кристалле — это автономный неразборный чип (электронная схема), выполняющий определённую функцию. Однокристальная Система может принимать и обрабатывать цифровые, аналоговые, радиосигналы или работать в аналого-цифровом режиме, объединив на кристалле процессор (или несколько), устройство ввода-вывода, блок оперативной и долговременной памяти. Особенность SoC — небольшой размер при возможностях, сравнимых с компьютером.

Двойственность архитектуры вычислительных систем.

Исторически развитие вычислительной техники характеризуется выраженным архитектурным и технологическим дуализмом: аппаратная и программная компоненты разрабатывались изолированно.

- **Аппаратное обеспечение** проектировалось как универсальная вычислительная платформа.
- **Программное обеспечение** разрабатывалось как функциональная надстройка, решающая конкретные прикладные задачи.
- **Кризис сложности встроенных систем:** Данная парадигма оказалась неэффективной в условиях экспоненциального роста сложности современных встраиваемых систем (embedded systems). Повышение требований к производительности привело к чрезмерному усложнению программного кода. Как следствие, возросли накладные расходы вычислительных ресурсов на поддержание внутренней иерархии управления, что повлекло за собой общее снижение надежности и отказоустойчивости систем.

Эволюция маршрутов проектирования

Традиционный (последовательный) маршрут Классический подход базируется на строгой последовательности этапов:

1. Разработка системной спецификации.
2. Трансляция спецификации в RTL-описание (Register Transfer Level) и логический синтез.
3. Физический синтез и прототипирование.
4. Программная имплементация (разработка ПО).

Критический недостаток: Отсутствие возможности программно-аппаратного ко-дизайна (совместного проектирования). Аппаратная часть создается до того, как программная часть может быть протестирована на ней. Выявление архитектурных ошибок на этапе написания ПО требует экономически нецелесообразного возврата к начальным стадиям (редизайна чипа).

Архитектурно-ориентированный маршрут на базе ESL

Современная научно-техническая база (внедряемая такими компаниями, как Cadence и Synopsys) опирается на **ESL (Electronic System Level — Системный уровень проектирования)**.

Рисунок 7.1. Раздел сайта «Дополнительно. Что такое SoC»

Концепция «Системы на кристалле» (СнК)

Для преодоления описанного кризиса была предложена концепция переноса части высокоуровневых функций управления и обработки данных из программной среды непосредственно в аппаратную реализацию. Технологическим базисом для этого послужило развитие программируемых логических интегральных схем (ПЛИС) и технологий создания заказных микросхем (ASIC).

Определение: Система на кристалле (System-on-a-Chip, SoC) — это не просто физическая микросхема, а комплексная технология проектирования проблемно-ориентированных вычислительных систем, стирающая технологические границы между аппаратными и программными компонентами.

Отличительные особенности СнК:

1. Основной проектирования служит аппаратная реализация сложных структур данных и алгоритмов.
2. Широко применяется концепция повторного использования (reuse) — интеграция готовых функциональных IP-блоков (Intellectual Property cores).
3. Обеспечивается гибкость программного уровня (Soft IP), что позволяет адаптировать аппаратную архитектуру под специфику конкретного алгоритма, а не наоборот.

Этот подход подразумевает параллельную разработку аппаратуры и ПО с их непрерывной ко-верификацией (совместным тестированием). Процесс проектирования на уровне ESL включает три ключевых этапа:

1. **Уровень спецификации (поведенческий уровень):** Создание исполняемой спецификации системы без привязки к конкретной архитектуре. Используются языки системного моделирования (например, UML, SystemC). Цель — формализовать реакцию системы на внешние воздействия.
2. **Макроархитектурный уровень:** Ключевой этап **программно-аппаратной декомпозиции** (Hardware/Software Partitioning). Инженеры распределяют вычислительную нагрузку между программными модулями и аппаратными сопроцессорами. Взаимодействие блоков моделируется на уровне передачи абстрактных сообщений.
3. **Микроархитектурный уровень:** Детальная проработка структурной организации устройства. Применяется TLM-моделирование (Transaction Level Modeling — моделирование на уровне транзакций), где описываются протоколы обмена данными, арбитраж шин и распределение памяти. После верификации на этом этапе генерируется код для производства чипа и финальные драйверы для программистов.

Методология проектирования «Систем на кристалле» представляет собой сдвиг парадигмы в микроэлектронике и информатике. Она требует перехода от изолированного программирования или схемотехники к комплексному аппаратно-программному ко-дизайну. Несмотря на то, что процессы макроархитектурной декомпозиции на данный момент формализованы и автоматизированы лишь частично, развитие ESL-инструментов является приоритетным направлением в индустрии вычислительной техники.

Рисунок 8.2. Раздел сайта «Дополнительно. Что такое SoC»

О "Sandbox"

это изолированная виртуальная среда для безопасного запуска и анализа программ, файлов или кода.

Как это работает

Песочница имитирует реальную компьютерную среду (операционную систему, ПО, аппаратные ресурсы), но при этом:

- жёстко ограничивает доступ к реальным ресурсам компьютера (файловой системе, реестру, сети);
- детально записывает все действия запущенного объекта;
- не позволяет изменениям выйти за пределы изолированной среды.

Основные типы песочниц

1. **Локальные (on-premise)** — развёртываются внутри организации на собственном оборудовании.
2. **Облачные (cloud)** — предоставляются как сервис по модели подписки.
3. **Встроенные в ПО** — например, песочницы в браузерах или антивирусах.

Где и зачем используют песочницы

В кибербезопасности:

- анализ подозрительных файлов и ссылок из писем или интернета;
- обнаружение новых, неизвестных угроз (в т.ч. APT-атак);
- поведенческий анализ вредоносного ПО (вирусов, троянов, шифровальщиков);
- изучение тактики злоумышленников (сопоставление с матрицей MITRE ATT&CK).

В разработке ПО:

- тестирование «сырого» кода без риска для основной системы;
- отладка программ в среде, имитирующей целевую платформу;
- проверка обновлений перед массовым развёртыванием;
- API-песочницы для тестирования взаимодействия приложений.

Ключевые возможности современных песочниц

- эмуляция разных ОС и конфигураций оборудования;
- поддержка множества типов объектов: файлы (PDF, офисные документы), архивы, ссылки, QR-коды;
- анализ сетевого трафика и попыток подключения к командным серверам;
- отслеживание изменений в файловой системе и реестре;
- интеграция с антивирусами и системами защиты (IDS/IPS, SIEM);
- применение методов машинного обучения для выявления аномалий;
- ретроспективный анализ (перепроверка файлов при появлении новых данных об угрозах).

Примеры реализации

Антивирусы и ИБ-решения:

- Kaspersky (ранее — «Безопасная среда»);
- Comodo Internet Security (Sandbox и Virtual Kiosk);
- Dr.Web vxCube;
- Check Point SandBlast Threat Emulation;
- Positive Technologies Sandbox.

Браузеры и приложения:

- Google Chrome (песочница для вкладок);
- Adobe Flash Player и Adobe Reader;
- macOS (встроенная песочница в ядре ОС).

Отдельные утилиты:

- Sandboxie;
- Firejail.

Рисунок 9. Раздел сайта «Дополнительно. О "Sandbox"»

4. Практическая работа

Практическая часть занятия направлена на активное применение полученных знаний, развитие системного мышления и формирование навыков обеспечения информационной безопасности. Учебная деятельность реализуется через выполнение заданий, размещенных на сайте (рис. 9) [3].



Рисунок 10. Раздел сайта «Закрепление материала»

Систематизация знаний (задание «Программная иерархия»)

Ученикам предлагается набор карточек с понятиями: «Приложение "Калькулятор"», «Игра с тяжелой 3D-графикой», «NFC-чип», «Драйвер камеры», «Нейропроцессор (NPU)», «Слой аппаратных абстракций (HAL)». Задача — распределить данные элементы по соответствующим уровням, что позволяет на практике усвоить знания о стеке технологий и взаимосвязи аппаратного и программного обеспечения.

Задание реализовано в онлайн-конструкторе genially.com по ссылке: <https://view.genially.com/69dce481c25a741a4712fbd3> (рис. 10).



Рисунок 11. Реализация задания «Программная иерархия»

Информационная безопасность и анализ угроз (задание «Служба безопасности»)

Один из ключевых этапов урока посвящен разбору практического кейса по кибербезопасности. Учащимся предлагается проанализировать риски, возникающие при установке программного обеспечения из несертифицированных источников (например, скачивание .apk-файлов с форумов). В ходе работы ученики:

- обосновывают причины уязвимости мобильных систем, оперируя терминами «верификация кода» и «нарушение цепочки доверия»;
- разбирают механизм защиты на основе принципа «песочницы» (Sandboxing), который изолирует работу приложений;
- оценивают эффективность «песочницы» в ситуациях, когда пользователь самостоятельно предоставляет вредоносному ПО системные разрешения;
- формулируют три-четыре профилактические меры для безопасной работы с мобильным устройством.

Задание реализовано через «Яндекс Формы» по ссылке: <https://forms.yandex.ru/u/69dc86c886d2d73b9dc06c006/> (рис. 11).

Служба безопасности

«Служба безопасности»

Представьте, что вы — специалисты отдела информационной безопасности в крупной компании. Сотрудники начали жаловаться, что на их корпоративные смартфоны (OS Android) проник неизвестный вирус, который крадет документы. Расследование показало, что все зараженные сотрудники скачивали модифицированные игры в формате .apk с игровых форумов. Составьте аналитическую записку для руководства, которая содержит:

1. Причины уязвимости: Объясните, почему скачивание .apk из форумов опаснее, чем установка из Google Play. Используйте термины из текста: верификация кода, нарушение цепочки доверия.
2. Механизм защиты: Опишите, как работает принцип «песочницы» (Sandboxing). Помог бы он в данной ситуации? Если приложение уже получило разрешения при установке, сможет ли «песочница» его остановить?
3. Профилактические меры: Что нужно запретить и что разрешить сотрудникам? (3-4 пункта).

* Причины уязвимости: Объясните, почему скачивание .apk из форумов опаснее, чем установка из Google Play. Используйте термины из текста: верификация кода, нарушение цепочки доверия.

Рисунок 12. Реализация задания «Служба безопасности»

Проектная деятельность (задание «Конструктор инноваций»)

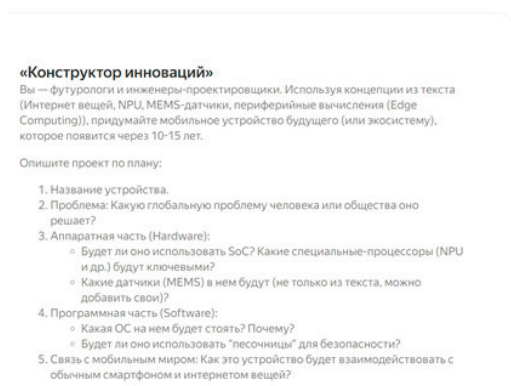
Завершается практический блок творческим заданием, где школьники примеряют на себя роли инженеров-проектировщиков и футурологов. Используя концепции интернета вещей (IoT), нейропроцессоров (NPU), MEMS-датчиков и периферийных вычислений (Edge Computing), учащиеся должны разработать концепт мобильного устройства будущего, которое

появится через 10–15 лет и будет решать конкретную социальную проблему.

Защита мини-проектов проходит по строгому алгоритму: описание проблемы, обоснование аппаратной части (SoC, датчики), выбор программной платформы с учетом безопасности и способов взаимодействия с глобальной сетью. Это задание наглядно демонстрирует учащимся потенциал мобильных технологий.

Задание реализовано через «Яндекс Формы» по ссылке: <https://forms.yandex.ru/u/69dce0d5493639b317b68938/> (рис. 12).

Конструктор инноваций



«Конструктор инноваций»

Вы — футурологи и инженеры-проектировщики. Используя концепции из текста (Интернет вещей, NPU, MEMS-датчики, периферийные вычисления (Edge Computing)), придумайте мобильное устройство будущего (или экосистему), которое появится через 10-15 лет.

Опишите проект по плану:

1. Название устройства.
2. Проблема: Какую глобальную проблему человека или общества оно решает?
3. Аппаратная часть (Hardware):
 - Будет ли оно использовать SoC? Какие специальные-процессоры (NPU и др.) будут ключевыми?
 - Какие датчики (MEMS) в нем будут (не только из текста, можно добавить свои)?
4. Программная часть (Software):
 - Какая ОС на нем будет стоять? Почему?
 - Будет ли оно использовать "песочницы" для безопасности?
5. Связь с мобильным миром: Как это устройство будет взаимодействовать с обычным смартфоном и интернетом вещей?

Рисунок 13. Реализация задания «Конструктор инноваций»

Дополнительное задание («Мобильные устройства внутри»)

Данное задание предлагается школьникам, досрочно завершившим работу на основных станциях.

В рамках этого задания учащиеся переходят от схем к виртуальному демонтажу смартфона. Им предлагается исследовать внутреннее строение реального устройства: определить местоположение материнской платы, батареи, системы охлаждения и шлейфов, а также соотнести теоретические знания о SoC с физическими микросхемами на плате.

Выполнение данного задания не только расширяет кругозор школьников, но и помогает развить инженерное мышление.

Задание реализовано в онлайн-конструкторе [genially.com](https://view.genially.com/69dba011fc0a5271b9ec2057) по ссылке: <https://view.genially.com/69dba011fc0a5271b9ec2057>.

Пример задания на рисунке 13.

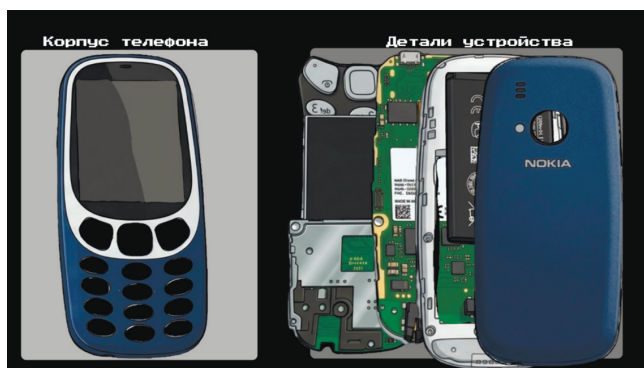


Рисунок 14. Реализация задания «Мобильные устройства внутри»

5. Рефлексия

В ходе фронтальной беседы учитель вместе с классом подводит итог урока, закрепляются понятия мобильного устройства как высокоинтегрированной системы и роли ОС. Совместно обсуждаются типичные затруднения и ошибки, возникшие при выполнении цифровых заданий.

Заключение. Предлагаемый конспект урока предназначен для углубленного уровня изучения информатики и содержит актуальный материал, расширяющий тему архитектуры компьютера, в соответствии с обновленными требованиями ФГОС СОО.

Список литературы

1. Босова Л. Л., Босова А. Ю. Информатика. 10 класс. Базовый уровень: учебник. М.: Просвещение, 2024. 288 с.
2. Кононов, Д. Ю. Информатика. 10-й класс. Углубленный уровень: учебник. М.: МФТИ. Физтех, 2024. 336 с.
3. Мобильные устройства: образовательный сайт-лонгрид [Электронный ресурс]. URL: <https://lesson-mobile-devices.tilda.ws/> (дата обращения: 16.05.2026).