

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 3

Применение мобильного приложения RaspController для ускоренного прототипирования IoT-систем удалённого видеонаблюдения и управления роботизированными устройствами

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент · Кафедра «Искусственный интеллект и big data»
«Технологии в интеллектуальных системах управления в IoT» · 4 курс, бакалавриат

1 Цель лекции

Сформировать системное представление об архитектуре и методах прототипирования IoT-систем на базе Raspberry Pi с мобильным интерфейсом RaspController, каналами удалённого доступа, видеонаблюдением и безопасным управлением исполнительными механизмами.

2 Задачи и результаты обучения

• Объяснить взаимодействие мобильного клиента, Raspberry Pi, сети, видеокамеры и исполнительных устройств.

• Различать локальное управление по SSH, передачу телеметрии и потоковую передачу видео.

• Разработать архитектуру прототипа дистанционного наблюдения и управления мобильным шасси.

• Оценить задержку, надёжность, защищённость и ограничения прототипа для интеллектуальных систем управления.

• Сформулировать программу эксперимента и критерии верификации.

3 План лекции

1. Роль мобильного интерфейса в архитектуре IoT
2. RaspController и особенности клиент-серверного взаимодействия
3. Подготовка Raspberry Pi и сетевого подключения
4. Видеонаблюдение и обработка видеопотока
5. Удалённое управление приводами мобильного робота
6. Математическая модель канала управления и показатели качества
7. Кибербезопасность, отказоустойчивость и исследовательская верификация

1 Мобильный интерфейс в интеллектуальной системе управления

Интернет вещей (IoT) объединяет физические устройства, датчики, вычислительные узлы и коммуникационные сервисы. Для интеллектуальной системы управления мобильный телефон выступает не только средством отображения данных, но и человеко-машинным интерфейсом (HMI): оператор получает изображения и телеметрию, задаёт режимы, инициирует команды и подтверждает тревоги. При этом вычисление управляющих воздействий, критичных к времени, следует размещать на локальном контроллере.

Типовая последовательность обмена: датчик/камера → Raspberry Pi → локальная сеть или защищённый удалённый канал → мобильное приложение; обратное направление: команда оператора → Raspberry Pi → проверка допустимости → драйвер привода → исполнительный механизм. Двусторонний обмен формирует контур наблюдения и управления, но его свойства зависят от задержек сети.

1.1. Уровни архитектуры

1 Мобильный интерфейс в интеллектуальной системе управления (продолжение)

Уровень	Состав	Основная функция
Физический	Камера, датчики, двигатели, драйверы	Измерение и воздействие
Периферийные вычисления	Raspberry Pi, службы Linux	Локальная обработка и защита
Коммуникационный	Wi-Fi/Ethernet, TCP/IP, SSH, HTTPS	Передача команд и данных
Прикладной	RaspController, интерфейс оператора	Мониторинг и ввод команд
Интеллектуальный	OpenCV, ML-модели, логика решений	Распознавание и рекомендации

1 Мобильный интерфейс в интеллектуальной системе управления (продолжение)

Важно разграничить управление через мобильное приложение и автономное интеллектуальное управление: наличие смартфона не делает систему интеллектуальной. Интеллектуальные функции возникают при анализе данных, прогнозировании состояния и адаптивном выборе воздействий.

4 Мобильное управление Raspberry Pi



Рис. 4. Цепочка передачи команды от оператора к устройству.

2 Назначение RaspController и принципы работы

RaspController — мобильный инструмент взаимодействия с Raspberry Pi, применяемый для быстрого доступа к ресурсам одноплатного компьютера. В учебном прототипе его целесообразно рассматривать как интерфейс администратора и оператора: подключение к Raspberry Pi, выполнение допустимых команд и просмотр доступной информации о системе. Конкретный набор экранов и функций зависит от версии приложения и настроек Raspberry Pi; перед лабораторной работой его необходимо проверить на установленной версии.

Наиболее универсальный способ связи — SSH: мобильный клиент аутентифицируется на Raspberry Pi, а команды выполняются в контексте разрешённой учётной записи. Поток видео при необходимости передаётся отдельным сервисом, поэтому нельзя считать, что любой выпуск RaspController автоматически предоставляет полноценный видеоплеер или драйвер камеры.

2.1. Преимущества и ограничения

Преимущества: быстрый запуск без разработки собственного мобильного приложения; доступность команд Linux; удобство диагностики; компактная аппаратная база. Ограничения: зависимость от сети, интерфейса приложения и разрешений пользователя; отсутствие гарантированного времени доставки; необходимость выделенного безопасного механизма для управления двигателями.

3 Подготовка аппаратно-программной платформы

Для учебного стенда необходимы Raspberry Pi с Raspberry Pi OS, карта памяти или SSD, блок питания достаточной мощности, камера (совместимая CSI/USB), локальная сеть, смартфон и при необходимости моторный драйвер. Пины GPIO не рассчитаны на прямое питание двигателя: требуется отдельный драйвер, согласование уровней и общая сигнальная земля там, где это предусмотрено схемой.

3.1. Первичная диагностика

```
hostnamectl
ip -br address
ping -c 4 192.168.1.1
```

Адрес шлюза 192.168.1.1 приведён только как пример: в конкретной сети используется фактический адрес маршрутизатора. Перед удалённым доступом следует включить SSH штатными средствами Raspberry Pi OS, создать отдельную учётную запись, установить обновления и настроить аутентификацию по ключу.

Подготовка аппаратно-программной платформы (продолжение)

3.2. Подключение мобильного клиента

Шаг 1. Подключить телефон и Raspberry Pi к доверенной локальной сети. Шаг 2. Определить IP-адрес Raspberry Pi командой hostname -I. Шаг 3. В мобильном приложении создать подключение с IP-адресом, портом SSH (обычно 22) и учётной записью. Шаг 4. Проверить соединение безопасной диагностической командой, например uptime. Шаг 5. Зафиксировать время ответа и случаи разрыва связи.

Для доступа через Интернет не рекомендуется открывать SSH непосредственно в публичную сеть. Предпочтительны VPN или иной контролируемый защищённый канал с ограничением доступа.

4 Организация мобильного видеонаблюдения

Система видеонаблюдения состоит из камеры, драйвера захвата, кодировщика, сетевого транспорта и средства просмотра. Камера создаёт последовательность кадров I(x,y,t). На Raspberry Pi возможны предварительная обработка, изменение разрешения, кодирование и публикация потока. На мобильном устройстве оператор наблюдает объект, а алгоритм компьютерного зрения может дополнительно обнаруживать события.

Современные выпуски Raspberry Pi OS используют стек libcamera и приложения семейства rpicam. Конкретные команды зависят от версии ОС и модели камеры. Для проверки совместимой камеры может применяться:

```
rpicam-hello --list-cameras
rpicam-still -o test.jpg
```

Для USB-камеры используют совместимые инструменты Video4Linux2. Сетевая трансляция требует отдельно настроенного сервера потокового видео (например, RTSP/WebRTC/MJPEG) и соответствующего клиента; выбор протокола определяется требованиями к задержке, совместимости и защите.

4 Организация мобильного видеонаблюдения (продолжение)

4.1. Компромисс качества и задержки

Увеличение разрешения и частоты кадров повышает информативность изображения, но увеличивает сетевую нагрузку и вычислительные затраты. При нестабильном Wi-Fi предпочтительнее снижать битрейт, обеспечивать адаптивную буферизацию и отделять видеотрафик от управляющих сообщений.

Ориентировочная пропускная способность видеопотока: $R \approx f \cdot S$, где f — частота кадров (кадр/с), S — средний объём закодированного кадра (бит). Для кодеков с переменным битрейтом выражение является оценочным.

5 Дистанционное управление мобильным роботизированным шасси

Рассмотрим двухколёсное дифференциальное шасси с двигателями, подключёнными к драйверу. Команды оператора («вперёд», «назад», «поворот», «стоп») поступают через мобильный интерфейс к сервису управления. Сервис преобразует их в ограниченные значения скорости и проверяет наличие действующего соединения.

Кинематическая модель дифференциального робота:

$$\dot{x} = v\cos\theta; \quad \dot{y} = v\sin\theta; \quad \theta = \omega; \quad v = r(\omega R + \omega L)/2; \quad \omega = r(\omega R - \omega L)/b,$$

где x,y — координаты, θ — ориентация, r — радиус колеса, b — расстояние между колёсами, ωR и ωL — угловые скорости колёс. Управляющая команда задаёт целевые скорости, но не гарантирует их точного достижения без обратной связи от энкодеров.

5.1. Принцип безопасного исполнительного контура

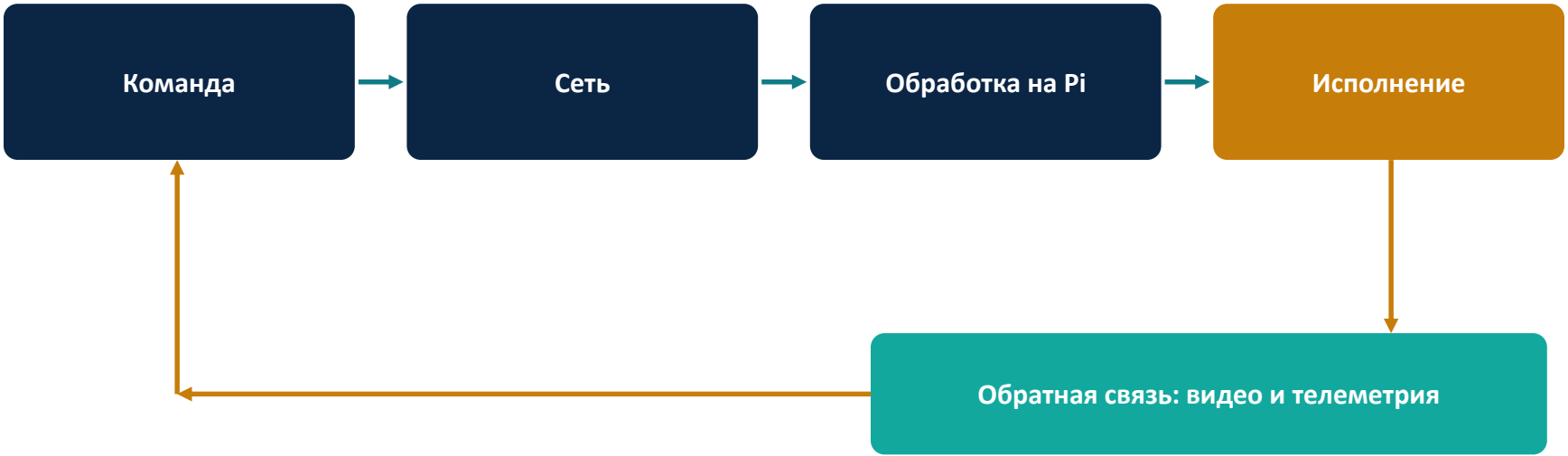
5 Дистанционное управление мобильным роботизированным шасси (продолжение)

Сетевые команды не должны непосредственно и бесконтрольно переключать GPIO. Рекомендуется отдельная служба, реализующая ограничение скорости, контроль срока действия команды, аварийную остановку и переход в безопасное состояние при потере связи. При разработке реального робота необходимы физический аварийный выключатель и независимые ограничения мощности.

```
Псевдокод локального контроллера:
если команда получена и прошла проверку:
    огран
ичить скорость и поворот
    обновить время последней команды
```

Этот псевдокод иллюстрирует архитектуру, а не готовую программу управления конкретным моторным драйвером. Подключение силовых цепей и программирование GPIO должны выполняться по документации выбранного оборудования.

8 Контур дистанционного управления



$$T = T(\text{сеть}) + T(\text{обработка}) + T(\text{исполнение})$$

Рис. 5. Суммарная задержка определяет допустимые режимы дистанционного управления.

6 Математические показатели качества удалённого управления

6.1. Полная задержка

$$T_total = T_capture + T_encode + T_network + T_decode + T_human + T_command + T_actuator.$$

Для автономного алгоритма вместо T_human учитывают время вычисления управляющего решения. В экспериментах следует отдельно измерять задержку изображения и задержку исполнительной команды: это разные каналы с различными требованиями.

6.2. Потери сообщений и доступность

$$P_loss = N_lost / N_sent \cdot 100\ \%; \quad A = T_available / T_observation \cdot 100\ \%.$$

Показатель P_loss отражает долю потерянных сообщений, A — доступность сервиса за период наблюдения. Для SSH/TCP потеря сетевых пакетов может проявляться как увеличение задержки из-за повторной передачи, поэтому дополнительно измеряют время ответа на уровне приложения.

Математические показатели качества удалённого управления (продолжение)

6.3. Джиттер и частота обновления

$$J = (1/(n-1))\,\Sigma|\Delta t_i - \Delta t_{i-1}|; \quad f_update = N_updates / T_observation.$$

Здесь Δt_i — последовательные интервалы между доставленными сообщениями. Сильные колебания интервалов ухудшают плавность дистанционного управления.

6.4. Пример расчёта

Пусть задержки захвата, кодирования, сети, декодирования и отображения составляют соответственно 35, 25, 60, 20 и 15 мс. Тогда задержка видеоканала равна 155 мс. Если обработка команды занимает 15 мс, её сетевая доставка — 60 мс, а реакция привода — 40 мс, то задержка командного канала составляет 115 мс. Значения условные и предназначены для демонстрации методики расчёта.

7 Информационная безопасность и отказоустойчивость

Риск дистанционного управления связан с компрометацией учётной записи, подменой команды, несанкционированным доступом к видео и отказом сети. Базовые меры: SSH-ключи, обновление системы, принцип минимальных привилегий, VPN, журналирование команд, ограничение доступа к камере, защита персональных данных и автоматическая остановка привода при потере связи.

При видеонаблюдении необходимо соблюдать применимые требования к приватности и обработке изображений людей. Использование распознавания лиц или иных биометрических функций требует отдельного правового и этического обоснования; для учебного прототипа достаточно обнаружения объектов без установления личности.

8 Методика экспериментального исследования прототипа

Исследовательский эксперимент рекомендуется проводить в трёх режимах: A — локальная сеть при устойчивом сигнале; B — сеть с ограниченной пропускной способностью; C — искусственно внесённая задержка и потери пакетов. Во всех режимах следует использовать одинаковую камеру, разрешение, алгоритм управления и маршрут движения.

8 Методика экспериментального исследования прототипа (продолжение)

Параметр	Метод измерения	Единица
Задержка видеопотока	Синхронизированные временные метки / тестовый таймер	мс
Задержка команды	От отправки до регистрации реакции контроллера	мс
Доступность	Доля времени исправной работы	%
Частота обновления	Число полученных кадров или сообщений за время	Гц
Отклонение траектории	Сравнение с эталонной траекторией	м
Частота аварийных остановок	Подсчёт срабатываний watchdog	раз

8 Методика экспериментального исследования прототипа (продолжение)

Для каждого режима проводят серию повторов, рассчитывают среднее, медиану, стандартное отклонение и доверительные интервалы. Необходимо описать условия Wi-Fi, расстояние, уровень сигнала и способ синхронизации времени. Сравнение режимов позволяет отделить влияние сети от влияния вычислительной нагрузки Raspberry Pi.

9 Практический пример: мобильный робот видеонаблюдения

Объект исследования — малогабаритное шасси с Raspberry Pi, камерой, драйвером двигателей и датчиком расстояния. Оператор использует RaspController для диагностики и выполнения разрешённых команд на Raspberry Pi; отдельный видеоклиент получает поток с камеры. Локальная служба управления ограничивает скорость, предотвращает движение при обнаружении препятствия и останавливает двигатели при обрыве связи.

Этапы: (1) подготовить ОС и сеть; (2) проверить камеру; (3) организовать защищённый доступ; (4) запустить службу управления; (5) проверить реакцию на команды без нагрузки; (6) провести движение на испытательном полигоне; (7) измерить показатели качества; (8) сформулировать ограничения и предложения по модернизации.

Переход к интеллектуальной системе возможен при добавлении распознавания препятствий OpenCV/нейросетью и автоматического ограничения скорости. Однако контур аварийной остановки должен оставаться независимым от вероятностного решения модели машинного обучения.

10 Заключение

Мобильное приложение ускоряет прототипирование интерфейса управления Raspberry Pi, однако его роль ограничена взаимодействием с вычислительным узлом. Надёжная IoT-система требует разделения интерфейса, видеоканала, локального контура управления и механизмов безопасности. Эффективность прототипа подтверждается не фактом подключения смартфона, а воспроизводимыми измерениями задержек, устойчивости, доступности и качества управления.

14 Контрольные вопросы

1. Какова роль мобильного клиента в многоуровневой IoT-архитектуре?
2. Чем удалённое управление отличается от автономного интеллектуального управления?
3. Какие функции RaspController необходимо проверить перед экспериментом?
4. Как работает SSH и почему предпочтительна аутентификация по ключу?
5. Какие узлы входят в тракт передачи видео?
6. Почему видео и команды целесообразно передавать раздельно?
7. Какие ограничения имеет управление приводами через сеть?
8. Почему двигатель нельзя подключать непосредственно к GPIO Raspberry Pi?
9. Как рассчитывается задержка командного канала?
10. Что такое джиттер и как он влияет на управление?
11. Как определяется доступность удалённого сервиса?

Контрольные вопросы (продолжение)

12. Какие условия должны вызывать аварийную остановку робота?
13. Как защитить удалённый доступ к Raspberry Pi?
14. Как организовать эксперимент по сравнению качества управления в разных сетевых условиях?
15. Какие интеллектуальные функции можно добавить к мобильному роботу видеонаблюдения?

15 Рекомендуемая литература и ресурсы

- **Ли П. Архитектура интернета вещей. М.:** ДМК Пресс, 2018. (Основная литература силлабуса).
- Матюшин А. О. Программирование микроконтроллеров: стратегия и тактика. М.: ДМК, 2016. (Основная литература силлабуса).
- **Raspberry Pi Documentation:** https://www.raspberrypi.com/documentation/
- **Raspberry Pi Camera software documentation:** https://www.raspberrypi.com/documentation/computers/camera_software.html
- **OpenSSH Manual Pages:** https://www.openssh.com/manual.html
- **OpenCV Documentation:** https://docs.opencv.org/

Примечание: тема и её направленность взяты из календарного плана загруженного силлабуса (неделя 3). Развёрнутая теория, формулы, пример эксперимента и рекомендации по безопасности являются учебно-методическим расширением темы.