ЛЕКЦИЯ № 10 · Проект MATLAB IoT

Сравнительный анализ особенностей технологии ThingSpeak с другими интернет-технологиями визуализации технологических данных



Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент
Кафедра «Искусственный интеллект и big data»
ОП 6B07113 «Интеллектуальные системы управления»

Цель лекции

Сформировать умение обоснованно выбирать интернет-платформу визуализации и анализа технологических данных. Изучить функциональные особенности ThingSpeak, Grafana, Node-RED Dashboard и промышленных SCADA/HMI-решений, сопоставить архитектуры, задержки, хранение, интеграцию, безопасность и стоимость владения. Закрепить сравнение на воспроизводимом MATLAB-примере.

План лекции

1. Задачи визуализации технологических данных
2. Архитектура IoT-мониторинга и поток данных
3. ThingSpeak: функции, преимущества и ограничения
4. Grafana и базы временных рядов
5. Node-RED и браузерные панели мониторинга
6. SCADA/HMI как промышленная альтернатива
7. Критерии и методика сравнительного анализа
8. Частота измерений, задержка и полнота данных
9. Аналитика MATLAB и интеграционные сценарии
10. Кибербезопасность, надёжность и эксплуатация
11. Практический пример сравнения визуализации
12. Выбор платформы для инженерного проекта

1 Задачи визуализации технологических данных

Технологические данные — измерения и события, характеризующие состояние объекта: температура, давление, расход, уровень, вибрация, ток, положение исполнительных механизмов и коды аварий. Их визуализация необходима для наблюдения динамики, выявления отклонений, анализа режимов и подготовки решений.

Инженерный график должен сохранять физический смысл показателей. Необходимо указывать единицы измерения, временную шкалу, источник данных, пределы допустимых значений и качество отсчётов. Красивый интерфейс не компенсирует отсутствие достоверных измерений.

Для оператора важны актуальные значения и предупреждения; для исследователя — история, экспорт, воспроизводимость и математический анализ; для инженера по эксплуатации — журналы событий, доступность системы и диагностика потери связи. Поэтому одна платформа не всегда оптимальна для всех задач.

1

Задачи визуализации технологических данных (продолжение)

Следует различать визуализацию, аналитику и управление. Визуализация отображает данные, аналитика вычисляет новые показатели, а управление формирует воздействие на объект. Для задач с требованиями безопасности и жёсткого реального времени необходим локальный проверенный контур управления.

В рамках лекции платформы сравниваются как типовые архитектурные варианты. Их возможности, лицензии, тарифы и ограничения могут изменяться; перед внедрением необходимо сверять актуальную документацию конкретных версий.

2 Архитектура системы интернет-мониторинга

Обобщённая архитектура включает датчики, контроллер или шлюз, транспортный протокол, сервер приёма, хранилище временных рядов, аналитический модуль и интерфейс пользователя. Например, Raspberry Pi собирает показания, публикует их по HTTPS или MQTT, а сервер сохраняет историю и формирует панель.

Данные могут передаваться по модели push, когда устройство само отправляет новые измерения, либо pull, когда сервер периодически опрашивает источник. Выбор влияет на сетевую нагрузку, работу через NAT, задержку и реакцию на сбой связи.

Временная метка измерения и время приёма — разные величины. Для корректного анализа следует хранить исходное время, идентификатор источника и при необходимости номер сообщения. Сетевые задержки не должны интерпретироваться как изменения физического процесса.

Систему характеризуют частота опроса $f_s=1/\Delta t$, пропускная способность, доля доставленных отсчётов, задержка отображения, доступность и глубина хранения. Для низкочастотной температуры и высокочастотной вибрации требования принципиально различаются.

2

Архитектура системы интернет-мониторинга (продолжение)

При сравнении важно рассматривать полный тракт от датчика до экрана, а не только интерфейс веб-панели. Узким местом может стать устройство, сеть, брокер, база данных, вычислительный сервер или браузер.

3 ThingSpeak: облачная платформа с MATLAB-аналитикой

ThingSpeak — облачный сервис для сбора, хранения, визуализации и анализа IoT-данных с интеграцией MATLAB. Основной объект — канал с временными записями и полями измерений. Доступ организуется через идентификаторы каналов и API-ключи с разграничением прав чтения и записи.

Преимущество ThingSpeak в учебной среде — небольшой объём начальной настройки, готовая веб-визуализация, REST API и возможность использовать функции MATLAB для расчёта статистик и построения графиков. Это особенно удобно при небольшом количестве датчиков и сравнительно редких измерениях.

Для более сложных систем ограничения могут проявляться в частоте обновления, числе каналов и объёме данных, возможностях оформления панелей и интеграции с локальными промышленными протоколами. Конкретные лимиты определяются условиями аккаунта и лицензии; их нельзя считать неизменными.

Типовой учебный сценарий: Raspberry Pi передаёт температуру и влажность в Field 1 и Field 2; ThingSpeak сохраняет историю; MATLAB Analysis вычисляет средние значения, обнаруживает превышения и формирует результаты для дальнейшего отображения.

При использовании облака нужно учитывать доступность интернета, защиту API-ключей, правила хранения данных и возможность экспорта. ThingSpeak целесообразно рассматривать прежде всего как платформу мониторинга и аналитики, а не как единственный контур критического управления.

4 Grafana и хранилища временных рядов

Grafana — платформа визуализации и наблюдаемости, которая подключается к различным источникам данных. Она не заменяет автоматически все компоненты сбора и хранения: часто используется совместно с базами временных рядов, SQL-хранилищами или системами метрик.

Её сильная сторона — гибкие панели, фильтры по времени и устройствам, отображение нескольких метрик, совместное использование дашбордов и механизмы оповещений в поддерживаемых конфигурациях. Это полезно для мониторинга множества установок и сравнительного анализа каналов.

При самостоятельном развёртывании Grafana инженер должен организовать поступление данных, настроить источник, права доступа, резервное копирование и обслуживание инфраструктуры. Поэтому функциональная гибкость сопровождается большей сложностью внедрения.

Для технологических сигналов часто применяется связка: Raspberry Pi → MQTT/HTTP → сервер обработки → база временных рядов → Grafana. Каждый компонент должен быть согласован по временным меткам, частоте и единицам измерения.

При выборе Grafana следует оценивать не только внешний вид панели, но и характеристики хранилища: срок хранения, агрегацию, точность запросов, восстановление после сбоев и стоимость эксплуатации.

5 Node-RED и браузерные панели

Node-RED — среда потокового программирования, в которой логика обработки строится из соединённых узлов. Она удобна для приёма сообщений MQTT, HTTP-запросов, преобразования форматов, маршрутизации и интеграции устройств.

Визуализация в экосистеме Node-RED реализуется с помощью подходящих dashboard-компонентов или внешних веб-интерфейсов. Доступность конкретных компонентов зависит от установленных пакетов и их версий; нельзя предполагать наличие одинаковой панели в каждой установке.

Преимущество Node-RED — быстрое прототипирование интеграционных сценариев без написания большого объёма кода. Например, поток может получать JSON-сообщение от датчика, проверять диапазон, сохранять запись и отправлять её на панель.

Для серьёзной эксплуатации необходимо учитывать управление версиями потоков, обработку исключений, права доступа, устойчивость после перезапуска и внешнее хранилище истории. Без дополнительных компонентов Node-RED не обязательно обеспечивает долгосрочное хранение технологических рядов.

Node-RED особенно полезен как шлюз между оборудованием и другими платформами, а не только как самостоятельный инструмент построения графиков.

6 Промышленные SCADA/HMI-системы

SCADA (Supervisory Control and Data Acquisition) — класс систем диспетчерского контроля и сбора данных. HMI (Human–Machine Interface) — человеко-машинный интерфейс для отображения состояний и взаимодействия оператора с оборудованием. Конкретные возможности зависят от продукта и конфигурации.

Промышленные решения обычно ориентированы на технологические объекты, работу с контроллерами, тревогами, журналами событий, разграничением прав и эксплуатационными процедурами. Они могут интегрироваться с промышленными протоколами через встроенные драйверы или шлюзы.

В отличие от учебной облачной платформы, SCADA-проект требует детальной проработки тегов, сигнализации, прав операторов, резервирования и тестирования. При этом локальная архитектура может быть предпочтительнее там, где интернет недоступен или действуют ограничения на передачу производственных данных.

Не следует отождествлять SCADA с безопасным автоматическим управлением. Критические функции защиты обычно должны быть реализованы в подходящих контроллерах и независимых защитных системах; интерфейс оператора не заменяет эти средства.

6 Промышленные SCADA/HMI-системы (продолжение)

Для исследовательского стенда ThingSpeak может быть достаточно, тогда как для промышленной линии часто требуется решение с проверенной интеграцией, архивированием, тревогами и поддержкой жизненного цикла.

7 Критерии и методика сравнения

Обоснованный выбор платформы начинается с технического задания. Необходимо определить количество датчиков N, частоту измерений f_s, требуемую задержку отображения Lmax, срок хранения R, число пользователей, требования к безопасности и необходимость аналитики MATLAB.

Далее формируется матрица критериев. Качественные оценки могут включать удобство внедрения, гибкость интерфейса, возможности интеграции, эксплуатационную сложность и доступность документации. Количественные показатели — задержка, потери сообщений, объём хранения, стоимость и трудозатраты.

Для демонстрации можно использовать взвешенную сумму $S=\Sigma w_i r_i$, где $w_i \geq 0$, $\Sigma w_i = 1$, а r_i — нормированная оценка критерия. Однако итог зависит от весов, поэтому нельзя объявлять универсального победителя без описания требований.

Веса для учебного проекта могут отдавать приоритет простоте и MATLAB-аналитике; для промышленного объекта — надёжности, безопасности и интеграции с контроллерами. Рекомендуется проводить анализ чувствительности, изменяя веса и проверяя устойчивость выбора.

7 Критерии и методика сравнения (продолжение)

Оценки ниже являются качественным ориентиром, а не результатом независимого стендового тестирования. При закупке или внедрении их следует заменить проверенными измерениями и актуальными условиями лицензирования.

Примечание: сравнение отражает типовые архитектуры. Функциональность конкретного продукта и лицензии необходимо проверять по актуальной документации.

7 Критерии и методика сравнения — таблица

Критерий	ThingSpeak	Grafana + хранилище	Node-RED + dashboard
Начальный запуск	Обычно простой	Требует источника данных	Удобен для прототипа
MATLAB-аналитика	Встроенная интеграция	Через отдельную интеграцию	Через отдельную интеграцию
Гибкость панелей	Ориентирована на каналы	Высокая при настройке	Зависит от компонентов
Хранение истории	В рамках сервиса и тарифа	Определяется базой данных	Требует настройки хранилища
Интеграция протоколов	Через API/шлюзы	Через источники и шлюзы	Широкие возможности потоков
Эксплуатация	Облачная зависимость	Обслуживание стека	Обслуживание потоков и узлов

8 Частота измерений, задержка и полнота данных

Для анализа пригодности платформы следует измерять задержку L=t_display−t_measure. Эта величина складывается из времени опроса, локальной обработки, передачи, записи, запроса и отрисовки. Если часы устройств не синхронизированы, прямое вычисление L может быть некорректным.

Полнота данных определяется отношением полученных уникальных отсчётов к ожидаемому числу: C=Nreceived/Nexpected·100 %. При этом необходимо исключать дубликаты и учитывать плановые остановки. Если датчик измеряет раз в 10 секунд, за 10 минут при включённых обеих границах ожидается 61 отсчёт.

Для оценки стабильности можно рассчитать медиану задержки, 95-й процентиль и долю превышений допустимого времени. Среднее значение само по себе скрывает редкие, но значительные задержки.

Частота отображения на графике не равна частоте измерения. Например, интерфейс может обновляться раз в 30 секунд, хотя локальный контроллер записывает значения каждую секунду. В отчёте эти частоты должны быть указаны отдельно.

8

Частота измерений, задержка и полнота данных (продолжение)

Для высокочастотных сигналов важно рассмотреть буферизацию и агрегирование: среднее, минимум, максимум и стандартное отклонение по окнам. Агрегация уменьшает объём данных, но может скрывать кратковременные события, если сохраняется только среднее.

```
% MATLAB: показатели полноты и интервалов
expectedTime = (0:10:100).';
receivedTime = [0 10 20 30 50 60 70 90 100].';
completeness = 100*numel(unique(receivedTime)) / ...
    numel(expectedTime);
intervals = diff(receivedTime);
fprintf('Полнота: %.1f %%\n',completeness);
fprintf('Максимальный интервал: %.1f c\n',max(intervals));
```

9 MATLAB-аналитика и интеграционные сценарии

В ThingSpeak аналитика MATLAB может быть частью облачного рабочего процесса. В архитектуре Grafana расчёты часто выполняются до записи в хранилище или внешним аналитическим сервисом; Grafana отображает подготовленные данные. В Node-RED часть предварительной логики реализуется потоками, а сложные модели могут вызываться отдельно.

Пример аналитической задачи: по температурному ряду вычислить скользящее среднее, скорость изменения и число превышений порога. Для измерений T_i с равномерным шагом Δt скорость $r_i=(T_i-T_{i-1})/\Delta t$. Если шаг неравномерный, используют фактические разности времени.

При выборе аналитического инструмента нужно учитывать доступность исходных данных, точность временных меток, возможность повторного запуска вычислений и документирование версии алгоритма. Результат, полученный из разных источников, должен быть воспроизводим при одинаковых входных данных.

Математические модели можно выполнять локально в MATLAB и публиковать только итоговые показатели. Такой подход уменьшает сетевую нагрузку, но требует контроля версий алгоритма и явного указания, какие данные были агрегированы.

9 MATLAB-аналитика и интеграционные сценарии (продолжение)

Для учебной работы целесообразно сначала выполнить расчёт на локальном массиве, а затем подключить реальный источник ThingSpeak. Это позволяет отделить ошибки алгоритма от проблем сети и авторизации.

```
% MATLAB: вычисление показателей
T = [24.1 24.3 24.7 25.2 25.5 26.0 26.8];
dt = 20; % секунд
smoothT = movmean(T,3);
rate = diff(T)/dt;
limit = 26;
fprintf('Среднее: %.2f °C\n',mean(T));
fprintf('Превышений: %d\n',sum(T>limit));
figure;
plot((0:numel(T)-1)*dt,T,'o-'); hold on;
plot((0:numel(T)-1)*dt,smoothT,'-');
```

9 MATLAB-аналитика и интеграционные сценарии (продолжение)

```
yline(limit,'--','Порог'); grid on;
xlabel('Время, с'); ylabel('Температура, °C');
legend('Измерения','Скользящее среднее','Location','best');
```

10 Кибербезопасность, надёжность и эксплуатация

Облачная платформа требует защиты учётных данных и ключей API. При самостоятельном развёртывании дополнительно необходимо защищать серверы, брокеры сообщений, базы данных и панели мониторинга. Во всех вариантах важны разграничение прав, защищённые соединения и журналирование действий.

Надёжность оценивается не только доступностью интерфейса. Следует проверить поведение при отключении интернета, перезапуске Raspberry Pi, отказе хранилища и задержке записи. Желательны локальный буфер и механизм восстановления передачи без потери последовательности.

При хранении производственных данных учитывают правила доступа, срок хранения, резервное копирование и требования организации. Облачное размещение может быть ограничено внутренней политикой или требованиями к данным.

Общая стоимость владения включает не только цену лицензии, но и время настройки, серверные ресурсы, резервное копирование, обновления, обучение персонала и техническую поддержку. Бесплатный стартовый вариант не обязательно дешевле при масштабировании.

Для систем с исполнительными механизмами интернет-панель должна рассматриваться как вспомогательный уровень. Критические блокировки и аварийная защита должны функционировать независимо от облачного сервиса.

11 Практический пример сравнения платформ

Рассмотрим учебную установку с двумя датчиками: температуры и влажности. Raspberry Pi формирует измерения каждые 20 секунд, а пользователь должен видеть текущие значения, историю за сутки и предупреждения при превышении заданных границ. Требуется возможность математического анализа в MATLAB.

Вариант A — ThingSpeak: создать канал с двумя полями, передавать записи через API, использовать встроенные графики и MATLAB для расчёта статистик. Достоинство — удобная учебная интеграция; ограничения частоты и доступа необходимо проверить для конкретного аккаунта.

Вариант B — Grafana: передавать данные в отдельное хранилище временных рядов и создать панель с графиками, фильтрами и индикаторами. Достоинство — гибкая визуализация; дополнительная задача — развёртывание и обслуживание инфраструктуры.

Вариант C — Node-RED: получать сообщения, преобразовывать JSON, направлять данные в хранилище и показывать их через доступный dashboard. Достоинство — гибкая интеграционная логика; аналитика MATLAB и долговременный архив потребуют дополнительного решения.

11

Практический пример сравнения платформ (продолжение)

Для честного сравнения все три варианта должны использовать одинаковый набор исходных измерений, одинаковые пороги и одинаковый период обновления. Иначе различия на графиках могут быть вызваны входными данными, а не свойствами платформ.

Результаты испытаний следует оформить таблицей: время настройки, число доставленных отсчётов, задержка отображения, наличие экспорта, доступность MATLAB-аналитики и поведение при потере сети. Без измерений нельзя утверждать, что конкретная платформа объективно быстрее или надёжнее.

12

Алгоритм выбора платформы для инженерного проекта

Первый этап — формулировка цели. Для учебного стенда важны простота, доступность и демонстрация MATLAB-аналитики. Для диспетчеризации производственной линии важнее надёжность, интеграция с контроллерами, управление тревогами и эксплуатационная поддержка.

Второй этап — описание данных: число источников, типы сигналов, частота, объём, сроки хранения и требования к временным меткам. Третий этап — требования к пользователям, правам доступа, сети и резервированию.

Четвёртый этап — предварительный отбор решений. ThingSpeak подходит для компактных облачных IoT-экспериментов с MATLAB; Grafana — для развитых панелей на основе существующих источников; Node-RED — для интеграционных потоков; SCADA/HMI — для специализированной промышленной эксплуатации.

Пятый этап — пилотное испытание на одинаковых данных. Необходимо измерить задержки, потери, трудозатраты и корректность восстановления после отказов. Шестой этап — анализ полной стоимости и рисков, а не только возможностей демонстрационной версии.

Итоговый выбор должен быть оформлен как инженерное обоснование: требования → варианты → критерии → измерения → ограничения → решение. Такой подход позволяет защитить выбор перед преподавателем, техническим заказчиком или проектной комиссией.

3 Заключение

ThingSpeak, Grafana, Node-RED и SCADA/HMI решают пересекающиеся, но не тождественные задачи. ThingSpeak удобен для учебного облачного мониторинга и MATLAB-аналитики; Grafana ориентирована на гибкую визуализацию данных из подключённых источников; Node-RED упрощает интеграционные потоки; SCADA/HMI применяется в промышленной диспетчеризации. Универсального лидера нет: выбор определяется частотой данных, задержкой, безопасностью, инфраструктурой и требованиями эксплуатации.

Контрольные вопросы (1/2)

1. Какие задачи решает визуализация технологических данных?

2. Чем визуализация отличается от аналитики и управления?

3. Какие уровни содержит типовая архитектура IoT-мониторинга?

4. Что такое модель push и модель pull?

5. Какие параметры характеризуют качество временного ряда?

6. Что представляет собой канал ThingSpeak?

7. Какие преимущества даёт интеграция ThingSpeak с MATLAB?

8. Почему ограничения тарифа необходимо проверять перед внедрением?

9. Для чего предназначена Grafana?

10. Почему Grafana часто используется вместе с отдельной базой данных?

11. Какова роль Node-RED в IoT-проекте?

12. От чего зависят возможности dashboard в Node-RED?

13. Чем SCADA отличается от простой веб-панели?

14. Почему HMI не заменяет аварийную защиту?

Контрольные вопросы (2/2)

15. Какие критерии используются для выбора платформы?

16. Что означает взвешенная оценка альтернатив?

17. Почему веса критериев влияют на результат?

18. Как вычисляется полнота данных?

19. Как определяется задержка отображения?

20. Почему важно синхронизировать часы устройств?

21. Чем частота измерений отличается от частоты обновления графика?

22. Как MATLAB применяется в сравнительном проекте?

23. Какие риски возникают при хранении данных в облаке?

24. Из каких компонентов складывается стоимость владения?

25. Как организовать корректное пилотное сравнение платформ?

Д Задания для самостоятельной работы

1. Составьте сравнительную таблицу ThingSpeak, Grafana, Node-RED и SCADA по восьми выбранным критериям. Обоснуйте каждый критерий.

2. Спроектируйте архитектуру мониторинга температуры и влажности на Raspberry Pi с использованием ThingSpeak и MATLAB.

3. Для 100 ожидаемых и 94 полученных уникальных измерений вычислите полноту данных; объясните, как учитывать дубликаты.

4. Создайте в MATLAB график температуры, скользящее среднее и отметьте превышения учебного порога.

5. Предложите матрицу весов для учебного проекта и для промышленной линии. Сравните результаты ранжирования решений.

6. Разработайте план испытаний трёх вариантов визуализации на одинаковом наборе данных с оценкой задержек, потерь и отказоустойчивости.

Рекомендуемые источники

- Силлабус дисциплины «Проект MATLAB IoT», тема лекции №10.
- MathWorks. ThingSpeak Documentation: https://www.mathworks.com/help/thingspeak/
- Grafana Documentation: https://grafana.com/docs/
- Node-RED Documentation: https://nodered.org/docs/
- MathWorks. MATLAB Documentation: https://www.mathworks.com/help/matlab/