ЛЕКЦИЯ № 9 · Проект MATLAB IoT

# ThingSpeak — открытая IoT-платформа с MATLAB-аналитикой




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

# Цель лекции

Изучить архитектуру платформы ThingSpeak, организацию каналов и полей данных, получение и передачу телеметрии, интеграцию с MATLAB Analysis и MATLAB Visualizations. Освоить построение воспроизводимого учебного проекта мониторинга с проверкой качества измерений, расчётом статистических характеристик и учётом ограничений облачного сервиса.

# План лекции

1. Назначение и архитектура ThingSpeak
2. Каналы, поля, записи и метаданные
3. Регистрация и создание канала
4. Запись данных: HTTP и функции MATLAB
5. Чтение данных и работа с временными метками
6. MATLAB Analysis: сценарии обработки
7. MATLAB Visualizations: создание графиков
8. Автоматизация анализа и событийные сценарии
9. Методы статистического анализа временных рядов
10. Безопасность, лимиты и надёжность
11. Практический проект мониторинга микроклимата
12. Проверка результатов и оформление отчёта

# 1 Назначение и архитектура ThingSpeak

**ThingSpeak —** облачный сервис для приёма, хранения, анализа и визуализации данных устройств Интернета вещей. Он позволяет объединять телеметрию с распределённых датчиков и выполнять обработку временных рядов средствами MATLAB. В учебной практике платформа удобна для демонстрации полного жизненного цикла данных: измерение, передача, сохранение, обработка и представление результата.

Типовая архитектура включает физический объект, датчик, устройство сбора данных (например, Raspberry Pi или Arduino с сетевым интерфейсом), канал передачи, облачную платформу и аналитическое приложение. Измерения поступают в канал ThingSpeak и далее используются для графиков, статистики или диагностических правил.

При проектировании важно разделять задачи устройства и облака. Raspberry Pi может выполнять локальную проверку измерений и буферизацию; ThingSpeak хранит временной ряд; MATLAB вычисляет производные показатели. Облачная аналитика не гарантирует детерминированного времени реакции, поэтому функции аварийной защиты следует оставлять на локальном оборудовании.

1

# Назначение и архитектура ThingSpeak (продолжение)

Открытость платформы в названии учебной темы означает доступность её интерфейсов и инструментов для создания IoT-проектов; это не означает, что все каналы публичны или что отсутствуют условия лицензирования. Доступность функций, квоты и ограничения определяются текущими правилами сервиса.

**Пример учебного применения —** мониторинг температуры и влажности лаборатории. Два параметра регистрируются с временными метками, публикуются в канал, а MATLAB рассчитывает средние значения, разброс и количество превышений установленных учебных порогов.

# 2 Каналы, поля, записи и метаданные

**Канал (channel) —** основная единица хранения данных ThingSpeak. Внутри канала значения организованы по полям (fields), каждое из которых соответствует измеряемому параметру. Например, Field 1 можно использовать для температуры в °C, Field 2 — для относительной влажности в процентах.

Одна запись (entry) обычно объединяет временную метку и значения одного или нескольких полей. В записи могут отсутствовать значения некоторых полей; при последующей обработке такие пропуски необходимо учитывать отдельно. Порядок записей и интервалы между ними имеют значение при расчёте производных и анализе динамики.

Канал имеет идентификатор (Channel ID), название, описание, настройки полей и параметры доступа. Полезно включать в описание единицы измерения, тип датчика и условия эксперимента, чтобы другой исследователь мог правильно интерпретировать данные.

Ключ записи (Write API Key) используется для отправки значений, ключ чтения (Read API Key) — для доступа к закрытым данным. Нельзя включать действующие ключи в публичные примеры кода или отчёты. Для публичных каналов чтение может быть доступно без секретного ключа.

# 2 Каналы, поля, записи и метаданные (продолжение)

В исследовательском проекте полезно отдельно хранить паспорт канала: обозначения полей, диапазоны значений, период опроса, версию программы и правила обработки ошибок. Это обеспечивает воспроизводимость расчётов.

# 3 Регистрация и создание учебного канала

Для работы с ThingSpeak пользователь создаёт учётную запись, доступную в рамках действующих условий сервиса, затем создаёт новый канал. В настройках указывают название, описание и активируют нужные поля. Для учебного мониторинга достаточно двух полей: Temperature_C и Humidity_pct.

После сохранения конфигурации следует записать Channel ID в документацию проекта и сохранить ключи доступа в защищённом месте. Проверку канала лучше начинать с одной тестовой записи, чтобы исключить ошибки настройки.

Далее необходимо убедиться, что графики отображают корректные единицы и что временные метки соответствуют ожидаемому часовому поясу. При использовании нескольких устройств следует заранее согласовать формат времени, желательно UTC.

Студентам рекомендуется присваивать каналам осмысленные названия, например Lab_Microclimate_Group01, и не публиковать личные данные или секретные ключи. При необходимости публичной демонстрации следует использовать отдельный учебный канал.

Условия доступа, максимальная частота публикации, объём хранения и доступные средства автоматизации могут изменяться. Перед лабораторной работой преподаватель должен проверить актуальные параметры аккаунта и документацию ThingSpeak.

# 4 Запись данных: HTTP API и MATLAB

Данные могут отправляться в ThingSpeak через HTTP-запросы с параметрами полей. В распространённой схеме устройство отправляет значения field1 и field2 на конечную точку обновления канала. Для защиты канала используется ключ записи.

После сетевого запроса необходимо проверить не только HTTP-статус, но и ответ сервиса, подтверждающий создание записи. Если запрос не принят, измерение следует сохранить в локальном буфере и повторить отправку позднее с соблюдением лимитов.

В MATLAB можно применять специализированную функцию thingSpeakWrite, если она доступна в используемой среде. Перед запуском необходимо заменить учебные обозначения собственным Channel ID и ключом записи. Команды с реальными ключами не следует публиковать.

Если данные собираются чаще разрешённого интервала обновления, следует отделить частоту измерения от частоты отправки. Возможны локальное накопление, агрегирование или пакетная загрузка через поддерживаемый API.

Учебный пример ниже показывает принцип передачи двух значений. Он требует существующего канала и действующих прав записи; значения не являются результатами физического измерения.

4

# Запись данных: HTTP API и MATLAB (продолжение)

```
% MATLAB: пример записи учебных данных
channelID = 123456; % заменить своим ID
writeKey = 'YOUR_WRITE_API_KEY'; % хранить безопасно
T = 25.4; H = 61.0;
thingSpeakWrite(channelID,[T H], ...
    'Fields',[1 2],'WriteKey',writeKey);
```

# 5 Чтение данных и временные метки

Для получения записей канала используется API чтения или функция MATLAB thingSpeakRead. В зависимости от параметров запроса можно получать определённое число последних точек либо данные за интервал времени. Для закрытого канала требуется соответствующее разрешение.

При обработке временных рядов следует проверять, что данные действительно получены: массив может быть пустым, а некоторые значения — отсутствовать. Вычисление mean без проверки пропусков способно привести к недостоверному результату.

Временная метка определяет момент записи или измерения в зависимости от выбранной архитектуры. При сетевых задержках момент публикации может отличаться от момента физического измерения. Поэтому для точного анализа динамики желательно хранить исходные временные метки.

Для равномерно дискретизированных рядов частота $f_s=1/\Delta t$. При неравномерном времени скорость изменения параметра вычисляется как $(x_{i+1}-x_i)/(t_{i+1}-t_i)$, а не через условный постоянный шаг.

# 5 Чтение данных и временные метки (продолжение)

При визуализации необходимо указывать единицы измерения и тип данных. Линия между соседними точками отражает способ графического представления, а не доказанное поведение процесса между измерениями.

```
% MATLAB: чтение последних 100 значений
channelID = 123456; % заменить своим ID
readKey = 'YOUR_READ_API_KEY';
[T,time] = thingSpeakRead(channelID, ...
    'Fields',1,'NumPoints',100,'ReadKey',readKey);
valid = isfinite(T);
if any(valid)
    plot(time(valid),T(valid),'-o'); grid on;
    xlabel('Время'); ylabel('Температура, °C');
else
    disp('Нет корректных измерений');
end
```

# 6 MATLAB Analysis: сценарии обработки

**MATLAB Analysis —** средство выполнения аналитических MATLAB-скриптов, связанных с данными ThingSpeak. С его помощью можно рассчитывать статистические характеристики, выявлять тенденции, сравнивать каналы и подготавливать производные показатели.

**Типовая последовательность анализа:** чтение данных → проверка качества → вычисление характеристик → формирование результата → при необходимости запись производного показателя в отдельное поле или канал. Для надёжности каждый этап должен иметь обработку пустых и некорректных данных.

Для температуры можно вычислять среднее $\bar{x}=(1/n)\Sigma x_i$, максимум, минимум и стандартное отклонение. Для определения превышений используется логическое условие $T>=T_{lim}$, а их количество равно sum(T>=T_lim).

При наличии пропусков NaN следует применять явную проверку isfinite или методы с пропуском отсутствующих значений, если это соответствует постановке задачи. Нельзя без обоснования заменять NaN нулём: это может создать искусственный спад температуры.

Аналитический скрипт должен быть воспроизводимым. В отчёте указывают входные поля, число анализируемых точек, критерии фильтрации, формулы, версию алгоритма и значения используемых порогов.

# 6 MATLAB Analysis: сценарии обработки (продолжение)

```
% MATLAB Analysis: пример вычислений по массиву
T = [24.8 25.0 25.5 NaN 26.4 28.1 29.0];
valid = isfinite(T);
Tv = T(valid);
if isempty(Tv)
    error('Нет достоверных измерений');
end
avgT = mean(Tv);
maxT = max(Tv);
numAlarm = sum(Tv >= 28);
fprintf('Средняя %.2f, максимум %.2f
',avgT,maxT);
fprintf('Превышений: %d
',numAlarm);
```

# 7 MATLAB Visualizations: построение графиков

MATLAB Visualizations предназначен для создания пользовательских графических представлений данных ThingSpeak. В отличие от стандартных графиков полей, аналитическая визуализация позволяет применять математические преобразования и отображать несколько вычисленных характеристик.

На графике технологического параметра рекомендуется показывать исходные измерения, оси с единицами, заголовок, временной диапазон и обозначение порога. Для сравнения нескольких рядов необходима легенда.

Например, можно отобразить температуру и её скользящее среднее movmean(T,w). Параметр w задаёт ширину окна в отсчётах. Чем шире окно, тем сильнее сглаживание и тем выше риск скрыть кратковременный пик.

Если данные содержат пропуски, при построении графика следует либо оставлять разрывы, либо явно указывать выбранный метод восстановления. Гладкая кривая не должна представляться как исходный измеренный сигнал.

# 7 MATLAB Visualizations: построение графиков (продолжение)

При публикации визуализации необходимо проверить, не раскрывает ли она конфиденциальные данные или идентификаторы оборудования. Доступность публичного просмотра определяется настройками канала и соответствующими разрешениями.

```
% MATLAB: визуализация учебных данных
t = (0:20:140).';
T = [24.5;24.9;25.3;26.0;26.8;27.7;28.1;28.5];
Ts = movmean(T,3);
plot(t,T,'o-',t,Ts,'-','LineWidth',1.4);
hold on; yline(28,'--','Учебный порог');
grid on; xlabel('Время, с');
ylabel('Температура, °C');
legend('Измерения','Скользящее среднее');
```

# 8 Автоматизация анализа и событийные сценарии

Для регулярной аналитики платформа может предоставлять механизмы запуска вычислений по расписанию и по событиям. В ThingSpeak используются такие возможности, как TimeControl и React, а также связанные сервисные действия в зависимости от конфигурации аккаунта.

Периодический сценарий подходит для расчёта среднего значения за заданный интервал, например каждые 30 минут. Событийный сценарий может использоваться для реакции на появление нового значения или на выполнение условия.

Однако периодическое выполнение в облаке не гарантирует точного момента срабатывания. Задержка сети, ограничения сервиса и доступность платформы влияют на время получения результата. Поэтому облачная автоматизация не должна быть единственным средством аварийного отключения.

При повторном запуске аналитики следует учитывать возможное повторное чтение тех же данных. Если скрипт создаёт новые записи, необходимо избегать бесконтрольного дублирования результатов.

В отчёте по автоматизации нужно фиксировать триггер, период или условие запуска, используемые поля, ожидаемое действие, ограничения и способ проверки успешного выполнения.

# 9 Статистический анализ временных рядов

Временные ряды IoT характеризуются изменением параметров во времени. Для их анализа применяют описательную статистику, сглаживание, оценку скорости изменения, поиск экстремумов и обнаружение отклонений от установленных границ.

Среднее значение характеризует общий уровень измерений, а стандартное отклонение — разброс. При наличии выбросов полезно дополнительно рассматривать медиану. Для сравнения нескольких датчиков необходимо убедиться, что измерения синхронизированы.

Скорость изменения температуры на интервале вычисляется как v_i=(T_{i+1}−T_i)/(t_{i+1}−t_i). При времени в секундах единицей будет °C/с. Большая скорость может указывать на изменение режима, но сама по себе не доказывает неисправность.

Корреляционный коэффициент между температурой и влажностью показывает статистическую взаимосвязь рядов, но не устанавливает причинность. При анализе реальных данных следует учитывать суточные циклы, внешние условия и различия датчиков.

9

# Статистический анализ временных рядов (продолжение)

При построении прогноза необходимо отличать интерполяцию внутри диапазона измерений от экстраполяции за его пределы. Последняя особенно чувствительна к изменению технологического режима и должна сопровождаться оценкой ограничений.

```
% MATLAB: статистика учебного ряда
t = (0:10:60).';
T = [22.0;22.7;23.4;24.0;24.9;25.5;26.1];
rate = diff(T)./diff(t);
fprintf('Средняя температура: %.2f °C
',mean(T));
fprintf('Станд. отклонение: %.2f °C
',std(T));
fprintf('Средняя скорость: %.4f °C/c
',mean(rate));
```

# 10 Безопасность, ограничения и надёжность

Ключи API необходимо хранить в защищённой конфигурации и предоставлять только тем компонентам, которым они действительно нужны. Ключ записи нельзя использовать как общедоступный идентификатор. Для передачи измерений следует использовать защищённое соединение HTTPS.

Платформа имеет условия использования и технические ограничения: частота обновления, квоты, объём хранения и доступность отдельных функций зависят от текущего типа аккаунта и лицензии. Эти значения необходимо проверять по актуальной документации перед запуском проекта.

При потере соединения устройство должно сохранять данные локально, ограничивать повторные запросы и восстанавливать передачу после появления сети. Для предотвращения дублирования полезны порядковые номера и временные метки измерений.

Система должна различать отсутствие измерения, некорректное значение и нормальное показание. Например, NaN>80 возвращает false, поэтому проверка порога без проверки достоверности может ошибочно обозначить аварийное состояние как норму.

Для производственных приложений облачная аналитика должна дополнять, а не заменять локальную систему безопасности. Важные команды необходимо проверять по допустимым диапазонам и правам доступа.

11

# Сквозной учебный проект: мониторинг микроклимата

**Постановка задачи:** сформировать учебный канал ThingSpeak с двумя полями — температурой и влажностью. Raspberry Pi или другой IoT-узел передаёт измерения, MATLAB анализирует историю и определяет превышения учебных порогов 30 °C и 75 % относительной влажности.

На этапе подготовки необходимо создать канал, указать единицы измерения, сохранить идентификатор и ключи, проверить отправку одной тестовой записи. Затем задают период измерения и отправки с учётом текущих ограничений сервиса.

Для воспроизводимого анализа сначала создадим искусственные данные в MATLAB. Таблица включает восемь измерений с шагом 20 секунд. Программа проверяет конечность значений, допустимость влажности и формирует логический массив предупреждений.

После проверки автономного кода учебные массивы можно заменить данными из ThingSpeak. Это позволяет отделить ошибки аналитики от ошибок связи и подключения датчика.

**Результат проекта:** таблица измерений, графики температуры и влажности, средние значения, количество превышений, описание алгоритма и анализ поведения при потере сети. В отчёте необходимо явно указать, какие данные являются учебными и какие получены физическим экспериментом.

11

# Сквозной учебный проект: мониторинг микроклимата (продолжение)

```
% MATLAB: полный автономный учебный пример
clear; clc; close all;
time_s = (0:20:140).';
temp_C = [24.5;25.0;25.8;26.7;28.0;29.4;30.2;29.6];
humidity_pct = [58;60;63;65;70;74;76;73];
valid = isfinite(temp_C) & isfinite(humidity_pct) ...
    & humidity_pct >= 0 & humidity_pct <= 100;
alarm = valid & (temp_C >= 30 | humidity_pct >= 75);
results = table(time_s,temp_C,humidity_pct,valid,alarm);
disp(results);
fprintf('Средняя T: %.2f °C
',mean(temp_C(valid)));
fprintf('Средняя H: %.2f %%
',mean(humidity_pct(valid)));
fprintf('Предупреждений: %d
',sum(alarm));
figure;
```

11

# Сквозной учебный проект: мониторинг микроклимата (продолжение)

```
plot(time_s,temp_C,'-o','LineWidth',1.4); hold on;
yline(30,'--','Порог 30 °C'); grid on;
xlabel('Время, с'); ylabel('Температура, °C');
title('Учебные данные микроклимата');
```

# 12 Проверка результатов и оформление отчёта

При проверке проекта сначала сравнивают количество отправленных и принятых записей, затем анализируют временные интервалы, пропуски и достоверность значений. Следует удостовериться, что каждому полю соответствует правильная физическая величина.

Для тестирования алгоритма необходимо предусмотреть нормальные значения, точное совпадение с порогом, превышение, NaN, отсутствие данных и потерю сетевого соединения. Это позволяет проверить корректность всех ветвей логики.

Графики должны иметь оси с единицами измерения, легенды при необходимости, понятные заголовки и указание учебных порогов. Не следует выдавать сглаженные или интерполированные точки за реальные измерения.

Отчёт рекомендуется структурировать следующим образом: цель, структурная схема, оборудование и программная среда, настройки канала без секретных ключей, алгоритм обмена, код MATLAB, таблица данных, графики, результаты проверки и выводы.

В заключении студент должен объяснить преимущества ThingSpeak для учебного IoT-мониторинга, ограничения облачного подхода и способы повышения надёжности при работе с реальными технологическими объектами.

# 3 Заключение

Изучены назначение ThingSpeak, структура каналов и полей, обмен телеметрией через API, использование MATLAB Analysis и MATLAB Visualizations, автоматизация обработки и методы анализа временных рядов. Практический пример показывает переход от измерений к графикам и диагностическим показателям. Для инженерного применения необходимо учитывать достоверность данных, права доступа, временные метки, ограничения сервиса и автономность локальных защитных функций.

# Контрольные вопросы (1/2)

1. Каково назначение ThingSpeak в архитектуре IoT?
2. Какие компоненты включает типовая система мониторинга?
3. Что называется каналом ThingSpeak?
4. Как организуются поля и записи канала?
5. Для чего нужен Channel ID?
6. Чем различаются ключи чтения и записи?
7. Какие единицы измерения следует фиксировать в метаданных?
8. Как создаётся учебный канал?
9. Какие параметры передаются при HTTP-записи?
10. Почему нужно проверять ответ API?
11. Для чего применяется thingSpeakWrite?
12. Как получить данные с помощью thingSpeakRead?
13. Почему важны временные метки?
14. Как проверить наличие NaN и Inf?
15. Что такое MATLAB Analysis?
16. Какие показатели можно рассчитать в MATLAB Analysis?
17. Что такое MATLAB Visualizations?

# Контрольные вопросы (2/2)

18. Чем скользящее среднее отличается от исходных измерений?

19. Какие сценарии автоматизации поддерживаются платформой?

20. Почему облачные сценарии не заменяют локальную аварийную защиту?

21. Как вычисляется скорость изменения параметра?

22. Почему корреляция не означает причинность?

23. Как учитывать ограничения частоты обновления?

24. Как организовать буферизацию при потере сети?

25. Какие разделы должен содержать отчёт по проекту?

# Д Задания для самостоятельной работы

1. Разработайте структуру канала ThingSpeak для трёх датчиков и опишите поля, единицы измерения и метаданные.

2. Создайте MATLAB-скрипт анализа 30 учебных температурных измерений: среднее, максимум, стандартное отклонение и превышения порога.

3. Сравните графики исходного ряда и скользящего среднего с окнами 3 и 5 отсчётов.

4. Разработайте схему периодической обработки данных ThingSpeak и укажите риски задержек.

5. Предложите алгоритм локального буферирования и повторной отправки измерений после отключения сети.

6. Подготовьте учебный отчёт с графиками температуры и влажности, проверкой ошибок и выводами о возможностях платформы.

# Рекомендуемые источники

- Силлабус дисциплины «Проект MATLAB IoT», тема лекции №9.
- MathWorks. ThingSpeak Documentation — https://www.mathworks.com/help/thingspeak/
- MathWorks. ThingSpeak REST API — https://www.mathworks.com/help/thingspeak/rest-api.html
- MathWorks. MATLAB Documentation — https://www.mathworks.com/help/matlab/
- Raspberry Pi Documentation — https://www.raspberrypi.com/documentation/