ЛЕКЦИЯ № 11 · Проект MATLAB IoT

Онлайн-визуализация метеостанции на базе Arduino в технологиях ThingSpeak



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

Цель лекции

Изучить принципы построения учебной IoT-метеостанции на базе Arduino, организацию сбора метеорологических параметров, передачу телеметрии в ThingSpeak, визуализацию временных рядов и анализ измерений средствами MATLAB. Сформировать навыки проверки качества данных, настройки сетевого обмена и оценки надёжности мониторинга.

План лекции

1. Назначение и архитектура метеостанции
2. Аппаратная платформа Arduino и сетевое подключение
3. Датчики температуры, влажности, давления и освещённости
4. Электрическое подключение и метрологические ограничения
5. Алгоритм измерений и локальная подготовка данных
6. Канал ThingSpeak и конфигурация полей
7. Передача показаний через HTTP API
8. Онлайн-графики и настройка отображения
9. MATLAB-анализ временных рядов
10. Потери данных, временные метки и устойчивость связи
11. Практический проект и проверка системы
12. Оценка результатов и направления расширения

1 Назначение и архитектура IoT-метеостанции

Метеостанция представляет собой измерительную систему, регистрирующую параметры окружающей среды во времени. В учебном проекте рассматриваются температура воздуха T, относительная влажность RH, атмосферное давление P и освещённость E. Дополнительными каналами могут стать скорость ветра, осадки и концентрация частиц, но их измерение требует отдельных датчиков и процедур калибровки.

Архитектура включает датчики, микроконтроллер Arduino, сетевой модуль, облачный канал ThingSpeak и MATLAB для последующего анализа. Arduino периодически считывает показания, проверяет их допустимость и отправляет через интернет. ThingSpeak сохраняет временные записи и предоставляет графики; MATLAB используется для вычисления статистических показателей и поиска отклонений.

Важно различать показания датчика и достоверные метеорологические данные. Даже корректно работающий датчик может показывать локально завышенную температуру из-за прямого солнечного нагрева корпуса. Поэтому качество установки, вентиляция и защита от осадков влияют на результат не меньше программного кода.

Учебная система предназначена прежде всего для мониторинга. Она не заменяет сертифицированную метеорологическую станцию без калибровки, аттестации и оценки неопределённости измерений. При подготовке отчёта необходимо явно указать назначение и ограничения макета.

2 Arduino и средства сетевого подключения

Arduino — семейство микроконтроллерных плат, предназначенных для взаимодействия с датчиками и исполнительными устройствами. В зависимости от модели доступны цифровые контакты, аналоговые входы, интерфейсы I²C, SPI и UART. В отличие от Raspberry Pi, типичная Arduino-плата выполняет программу без полноценной настольной операционной системы.

Для передачи данных в ThingSpeak требуется сетевое подключение. Оно может быть встроено в выбранную плату либо обеспечиваться внешним модулем Wi-Fi или Ethernet. Наличие сетевого интерфейса и совместимость библиотек необходимо проверять по конкретной модели платы.

Типовая структура программы Arduino состоит из setup(), где выполняется инициализация датчиков и сети, и loop(), где периодически выполняются измерение, проверка, передача и обработка ошибок. Программа должна исключать бесконтрольную частую отправку запросов.

При выборе платы учитывают напряжение логических уровней, объём памяти, число интерфейсов, энергопотребление и доступность библиотек. Для автономной установки дополнительно оценивают питание, устойчивость Wi-Fi и защиту корпуса от погодных воздействий.

3 Датчики метеорологических параметров

Температура измеряется терморезистивными, полупроводниковыми или цифровыми датчиками. В учебных проектах часто используются комбинированные датчики температуры и влажности. Важно изучить рабочий диапазон, заявленную точность, разрешение и время отклика конкретного изделия.

Относительная влажность RH показывает отношение текущего парциального давления водяного пара к давлению насыщенного пара при той же температуре, выраженное в процентах. Она зависит от температуры, поэтому два канала желательно измерять согласованно по времени.

Атмосферное давление измеряется в паскалях или гектопаскалях: 1 гПа = 100 Па. При анализе необходимо различать давление на высоте датчика и приведённое к уровню моря; без сведений о высоте и методике приведения эти величины нельзя напрямую отождествлять.

Освещённость обычно выражается в люксах, однако значение зависит от спектральной чувствительности датчика и геометрии установки. Освещённость не равна солнечной энергетической облучённости в Вт/м². Для соответствующих научных задач применяются специализированные приборы.

Перед объединением данных в таблицу все каналы должны иметь согласованные единицы и названия. Например: temp_C, humidity_pct, pressure_hPa, illuminance_lux.

4 Подключение датчиков и метрологические ограничения

Датчики с интерфейсом I^2C подключают к линиям SDA и SCL, питанию и общей земле согласно документации. Некоторые модули имеют подтягивающие резисторы, другие требуют их установки. Нельзя автоматически считать модуль безопасным для платы с иным логическим напряжением.

При работе с аналоговым датчиком Arduino преобразует входное напряжение в цифровой код встроенным АЦП, если он предусмотрен моделью. Связь между кодом n, опорным напряжением Vref и разрядностью N в идеализированной модели описывается формулой $V \approx n \cdot Vref/(2^N - 1)$. Реальные характеристики зависят от схемы и калибровки.

На точность влияют шум питания, нагрев корпуса, расположение сенсора, конденсация, старение и ошибки калибровки. Для температуры полезно проводить сравнение с эталонным термометром в нескольких точках, а не только в одном режиме.

Абсолютная ошибка отдельного измерения определяется как $e_i = x_i - xref,_i$. Средняя абсолютная ошибка $MAE = (1/n)\Sigma|e_i|$, а среднеквадратичная $RMSE = \sqrt{[(1/n)\Sigma e_i^2]}$. Без эталонных измерений вычислять MAE и RMSE как оценку точности датчика нельзя.

4

Подключение датчиков и метрологические ограничения (продолжение)

Размещение датчика в защитном экране с вентиляцией уменьшает влияние прямого солнечного излучения. В лабораторном отчёте указывают условия установки и возможные источники систематической погрешности.

5 Алгоритм сбора и локальной подготовки измерений

Последовательность работы микроконтроллера: инициализация → чтение датчиков → проверка успешности → проверка диапазонов → формирование сообщения → передача → регистрация результата → ожидание следующего периода. При ошибке одного датчика нельзя выдавать последнее значение за новое без специального признака устаревания.

Для периодического опроса задаётся интервал Δt. Частота измерения $f_s = 1/\Delta t$. Например, при $\Delta t = 30$ с частота равна $1/30 \approx 0{,}0333$ Гц. Для медленно меняющихся параметров окружающей среды такой интервал может быть достаточным для учебной демонстрации.

Если измерения получаются быстрее допустимой частоты записи в облако, следует разделить локальный период измерения и период публикации. Программа может агрегировать показания или сохранять их в локальный буфер; при этом важно не терять информацию о времени исходных измерений.

Проверка достоверности включает isfinite-эквивалент в используемом языке, диапазоны по спецификации датчика и контроль резких скачков. Скачок не всегда является ошибкой: его следует отмечать как подозрительный и проверять по контексту.

5 Алгоритм сбора и локальной подготовки измерений (продолжение)

Для устойчивой работы предпочтительно использовать неблокирующую периодическую логику на основе системного счётчика времени, а не длительные задержки, если программа одновременно должна обслуживать сеть и другие задачи.

6 Настройка канала ThingSpeak

В ThingSpeak создаётся канал, содержащий временные записи. Для учебной метеостанции удобно назначить Field 1 температуре, Field 2 относительной влажности, Field 3 атмосферному давлению и Field 4 освещённости. Названия и единицы измерения следует отразить в настройках.

Идентификатор канала используется для обращения к данным. Для записи требуется соответствующий Write API Key, а для чтения закрытого канала — Read API Key. Ключи доступа необходимо хранить вне общедоступного исходного кода и не помещать в отчёты.

Встроенные графики ThingSpeak позволяют наблюдать изменения каждого поля во времени. Канал не гарантирует достоверность измерений: он отображает то, что было корректно принято сервисом. Контроль физических диапазонов и качества данных остаётся задачей разработчика.

Действующие ограничения частоты обновления, объёма данных и доступных функций зависят от типа учётной записи и лицензии. Перед экспериментом необходимо проверить актуальные условия ThingSpeak, а не рассчитывать на произвольный интервал публикации.

Для демонстрации рекомендуется сначала отправить один тестовый набор значений, проверить его появление в канале и только затем включать регулярную публикацию.

7 Передача данных в ThingSpeak через HTTP

Один из способов загрузки — HTTP-запрос к интерфейсу обновления канала ThingSpeak. В запросе передаются ключ записи и значения полей. Для передачи по сети предпочтительно использовать HTTPS и проверять код ответа и содержимое сообщения сервиса.

При записи через API ThingSpeak может вернуть идентификатор созданной записи либо значение, сигнализирующее об отказе. Поэтому успешный сетевой запрос не следует автоматически считать успешным сохранением измерения.

Ниже приведён концептуальный пример для платы Arduino с совместимой библиотекой WiFi и HTTPClient. Он рассчитан на платы, где доступны соответствующие классы; для других моделей требуется адаптация сетевой библиотеки. Вместо учебных констант необходимо подключить код реального датчика.

Ключ записи должен быть подставлен через защищённую конфигурацию. В открытом учебном тексте оставлен заполнитель. Период публикации установлен 30 секунд как демонстрационный и должен соответствовать действующим ограничениям аккаунта.

7 Передача данных в ThingSpeak через HTTP (продолжение)

При потере сети измерение следует записать в локальный журнал, если оборудование и память это позволяют. Повторную передачу необходимо реализовать так, чтобы не возникали бесконечные запросы и не нарушались ограничения сервиса.

```
// Концептуальный пример: Arduino с WiFi и HTTPClient
#include <WiFi.h>
#include <HTTPClient.h>
const char* ssid = "YOUR_WIFI_NAME";
const char* password = "YOUR_WIFI_PASSWORD";
const char* writeKey = "YOUR_WRITE_API_KEY";
unsigned long lastSend = 0;
const unsigned long intervalMs = 30000;
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
}
void loop() {
```

7

Передача данных в ThingSpeak через HTTP (продолжение)

```
if (millis() - lastSend < intervalMs) return;
lastSend = millis();
float temperature = 25.4; // учебные данные
float humidity = 61.0;    // учебные данные
if (WiFi.status() != WL_CONNECTED) {
  Serial.println("Нет сети: сохранить в буфер");
  return;
}
HTTPClient http;
String url = "https://api.thingspeak.com/update";
String body = "api_key=" + String(writeKey)
    + "&field1=" + String(temperature, 2)
    + "&field2=" + String(humidity, 2);
http.begin(url);
http.addHeader("Content-Type",
               "application/x-www-form-urlencoded");
int status = http.POST(body);
```

7 Передача данных в ThingSpeak через HTTP (продолжение)

```
  String answer = status > 0 ? http.getString() : "";
  Serial.printf("HTTP: %d; ответ: %s\n",
                status, answer.c_str());
  http.end();
}
```

8 Онлайн-визуализация метеорологических данных

График температуры строится по оси времени, а вертикальная ось должна иметь обозначение °C. Для влажности используют %, для давления — гПа, для освещённости — лк. Смешивать эти величины на одной шкале без нормирования некорректно.

Встроенные графики ThingSpeak позволяют отслеживать тренды и отдельные выбросы. При визуальном анализе важно различать отсутствие измерений и реальное постоянство параметра. Если точки соединяются линией через длительный разрыв, пользователь может ошибочно принять этот интервал за непрерывное наблюдение.

При проектировании панели мониторинга следует показать текущее значение, время последней записи, график за выбранный интервал и статус актуальности данных. Одно только последнее значение без временной метки может быть устаревшим.

Пороговые линии полезны для учебных предупреждений, но их значения должны иметь объяснение. Например, температура 30 °C может быть условным порогом учебной задачи, но не универсальной границей опасности.

Качество визуализации оценивают по читаемости подписей, правильности единиц, разумному масштабу и возможности обнаруживать пропуски и аномалии.

9 MATLAB-анализ измерений ThingSpeak

В MATLAB данные ThingSpeak можно получать с помощью поддерживаемых функций чтения канала. Для закрытого канала требуется ключ чтения. Возвращённые значения и временные метки необходимо проверить до вычисления статистик.

Основные показатели: среднее $\bar{x}=(1/n)\Sigma x_i$, минимум, максимум, стандартное отклонение и доля корректных отсчётов. При наличии эталона дополнительно вычисляют MAE и RMSE. Для выявления тенденций применяют сглаживание, линейную регрессию или оценку скорости изменения.

Скользящее среднее уменьшает шум, но может скрывать короткие экстремумы. При поиске резких событий рекомендуется одновременно отображать исходные точки и сглаженный ряд.

Для временных меток типа datetime разность между соседними значениями даёт duration. Перевод в секунды позволяет вычислить фактические интервалы и выявить пропуски. При неравномерном шаге нельзя без проверки считать все измерения равноотстоящими.

Ниже приведён автономный пример на учебных данных, который выполняется без подключения к интернету. Для реального канала массивы заменяются результатами thingSpeakRead при наличии доступа и совместимой среды MATLAB.

9

MATLAB-анализ измерений ThingSpeak (продолжение)

```
% MATLAB: автономный анализ учебных измерений
clear; clc; close all;
time_min = (0:5:35).';
temp_C = [22.4;22.8;23.1;23.6;24.0;NaN;24.8;25.0];
humidity_pct = [62;61;60;59;58;57;56;55];
valid = isfinite(temp_C) & isfinite(humidity_pct);
T = temp_C(valid);
t = time_min(valid);
meanT = mean(T);
maxT = max(T);
smoothed = movmean(T,3);
fprintf('Средняя температура: %.2f °C\n',meanT);
fprintf('Максимальная температура: %.2f °C\n',maxT);
figure;
plot(t,T,'o-','LineWidth',1.3); hold on;
plot(t,smoothed,'-','LineWidth',1.5);
grid on; xlabel('Время, мин');
```

9

MATLAB-анализ измерений ThingSpeak (продолжение)

```
ylabel('Температура, °C');
legend('Исходные данные','Скользящее среднее');
title('Учебная IoT-метеостанция');
```

10 Потери данных, временные метки и надёжность связи

Метеостанция может терять записи из-за отсутствия Wi-Fi, перегрузки сервиса, перезапуска платы или ошибки датчика. Поэтому в инженерном проекте следует различать пропуск измерения, ошибку измерения и отказ передачи.

Если ожидается N_{exp} записей, а принято N_{ok}, долю доставки можно оценить как $R = N_{ok}/N_{exp} \cdot 100\ \%$. Эта величина характеризует полноту данных за выбранный интервал, но не доказывает правильность самих измерений.

Для выявления пропусков анализируют разность соседних временных меток $\Delta t_i = t_{i+1} - t_i$. Если ожидаемый шаг 30 секунд, интервал 90 секунд может свидетельствовать о двух отсутствующих промежуточных записях, но необходимо учитывать джиттер и возможное изменение расписания.

При буферизации желательно сохранять исходное время измерения, идентификатор устройства и номер записи. Если данные передаются после восстановления связи, время публикации не следует безусловно принимать за время измерения.

Устойчивость системы проверяют испытаниями: отключение Wi-Fi, повторное подключение, временная недоступность сервиса, ошибочный API-ключ и некорректное значение датчика. Результаты тестов должны быть отражены в отчёте.

11

Практический проект: Arduino — ThingSpeak — MATLAB

Цель практического проекта — создать работающий прототип онлайн-мониторинга микроклимата. На первом этапе студент выбирает совместимую плату Arduino и цифровой датчик температуры и влажности, изучает распиновку и собирает схему без подключения к опасным напряжениям.

На втором этапе выполняется локальное считывание данных и вывод значений в последовательный монитор. Проверяются диапазоны и единицы. На третьем этапе создаётся канал ThingSpeak с двумя полями, после чего отправляется тестовая запись.

На четвёртом этапе организуется периодическая передача с интервалом, разрешённым для используемого аккаунта. Для каждой попытки фиксируются время, результат и код ответа. На пятом этапе студент проверяет онлайн-графики и сопоставляет число локальных измерений с числом облачных записей.

На шестом этапе в MATLAB рассчитываются средняя температура, максимум, число пропусков и график временного ряда. Если доступ к ThingSpeak отсутствует, разрешается выполнить аналитическую часть на сохранённом CSV с обязательным указанием, что данные не были получены из облака.

Финальный отчёт включает структурную схему, таблицу оборудования, описание интерфейсов, исходный код, снимки экрана канала, графики MATLAB, расчёт полноты данных и обсуждение ограничений точности.

12 Оценка результатов и расширение метеостанции

Качество проекта оценивается не только по факту появления графика в ThingSpeak. Необходимо проверить корректность подключения, воспроизводимость передачи, согласованность временных меток, достоверность значений, понятность графиков и способность системы обнаруживать ошибки.

Расширение возможно за счёт барометрического датчика, датчика освещённости, измерения скорости ветра, локального дисплея и энергонезависимого журнала. Каждое расширение увеличивает требования к питанию, калибровке, синхронизации и обработке данных.

Для длительного мониторинга можно рассчитывать суточные и недельные показатели, сравнивать сезоны, анализировать корреляции и строить прогнозы. Однако прогнозирование требует достаточного объёма наблюдений и проверки модели на независимых данных.

При использовании метеоданных в управлении вентиляцией или поливом облачную аналитику следует отделять от локальной защитной логики. Команды, влияющие на оборудование, должны проверяться на допустимость и исполняться с учётом задержек связи.

Таким образом, учебная метеостанция является удобным примером полного цикла IoT: физическое измерение, цифровое представление, сетевой обмен, облачное хранение, визуализация и инженерная интерпретация.

3 Заключение

Изучены архитектура метеостанции на базе Arduino, выбор датчиков, правила электрического подключения, организация HTTP-передачи в ThingSpeak и визуализация метеорологических параметров. Показаны методы MATLAB-анализа, проверки достоверности, оценки пропусков и повышения устойчивости передачи. Реализация учебного прототипа формирует навыки проектирования измерительных IoT-систем с документированием точности и ограничений.

Контрольные вопросы (1/2)

1. Что такое IoT-метеостанция и какие задачи она решает?

2. Какие блоки входят в архитектуру Arduino — ThingSpeak — MATLAB?

3. Чем Arduino отличается от Raspberry Pi?

4. Какие параметры можно измерять в учебной метеостанции?

5. В каких единицах измеряется относительная влажность?

6. Чем давление на высоте датчика отличается от давления на уровне моря?

7. Почему освещённость в люксах не равна солнечной облучённости?

8. Какие интерфейсы используются для цифровых датчиков?

9. Какие риски возникают при неправильном подключении питания?

10. Как вычислить напряжение по коду АЦП в идеализированной модели?

11. Какие факторы влияют на погрешность температуры?

12. Что такое период измерения и частота дискретизации?

Контрольные вопросы (2/2)

13. Как устроены функции setup() и loop() Arduino?

14. Что такое канал и поле ThingSpeak?

15. Для чего нужен Write API Key?

16. Почему следует проверять ответ HTTP API?

17. Какие данные необходимо показывать на панели мониторинга?

18. Как обработать NaN в MATLAB?

19. Для чего применяется movmean?

20. Почему сглаживание может скрывать кратковременные пики?

21. Как определить долю успешно доставленных записей?

22. Почему важна исходная временная метка измерения?

23. Как организовать работу при потере Wi-Fi?

24. Какие проверки необходимы перед длительным экспериментом?

25. Что должно быть включено в итоговый отчёт по проекту?

Д Задания для самостоятельной работы

1. Разработайте структурную схему метеостанции с двумя датчиками и каналом ThingSpeak.

2. Создайте в MATLAB учебный ряд из 30 температурных отсчётов, добавьте три пропуска и определите полноту данных.

3. Подготовьте алгоритм Arduino для измерения температуры и влажности с проверкой диапазонов и регистрацией ошибок.

4. Опишите конфигурацию канала ThingSpeak с четырьмя полями, указав единицы и смысл каждого поля.

5. Постройте графики температуры и влажности за один условный час и сравните исходные и сглаженные значения.

6. Разработайте план испытаний метеостанции при потере сети, некорректном API-ключе и отказе датчика.

Рекомендуемые источники

- Силлабус дисциплины «Проект MATLAB IoT», тема лекции №11.
- MathWorks. ThingSpeak Documentation: https://www.mathworks.com/help/thingspeak/
- MathWorks. MATLAB Documentation: https://www.mathworks.com/help/matlab/
- Arduino Documentation: https://docs.arduino.cc/
- MathWorks. ThingSpeak REST API: https://www.mathworks.com/help/thingspeak/rest-api.html