ЛЕКЦИЯ № 6 · Проект MATLAB IoT

Задачи сопряжения мини-ПК Raspberry Pi с данными MATLAB



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

Цель лекции

Изучить архитектуру взаимодействия Raspberry Pi и MATLAB, способы подключения датчиков, организации передачи данных по файлам и сети, синхронизации измерений и обработки ошибок. Сформировать навыки построения воспроизводимого учебного IoT-прототипа с последующим анализом измерений в MATLAB Online.

План лекции

1. Роль Raspberry Pi и MATLAB в архитектуре IoT
2. Аппаратные интерфейсы Raspberry Pi и электрическая безопасность
3. Программная подготовка Raspberry Pi и MATLAB
4. Архитектуры сопряжения: локальная, файловая и сетевая
5. Передача измерений в CSV и чтение в MATLAB
6. Сетевой обмен: HTTP, MQTT и JSON
7. Взаимодействие с MATLAB через поддерживаемые пакеты
8. Временные метки, частота опроса и качество данных
9. Обработка ошибок и информационная безопасность
10. Сквозной пример: температурный мониторинг
11. Визуализация, контроль порогов и оценка точности
12. Порядок лабораторного внедрения и типичные неисправности

1 Роль Raspberry Pi и MATLAB в архитектуре IoT

Raspberry Pi — одноплатный компьютер, способный выполнять операционную систему, программы сбора данных и сетевые службы. MATLAB — вычислительная среда для обработки массивов, построения моделей, визуализации и анализа результатов. Их сопряжение позволяет перейти от отдельных датчиков к целостной системе мониторинга.

В типовой схеме датчик формирует электрический или цифровой сигнал; Raspberry Pi считывает показания, добавляет временную метку и сохраняет либо передаёт данные; MATLAB принимает массив, проверяет его корректность, рассчитывает показатели и строит графики. Такая архитектура разделяет сбор измерений и последующую аналитику.

Важно различать мониторинг и управление в реальном времени. Удалённый MATLAB-анализ может выполняться с сетевой задержкой и не гарантирует детерминированный период реакции. Поэтому защитные функции, критические к времени, необходимо реализовывать на локальном контроллере или специализированном устройстве.

1

Роль Raspberry Pi и MATLAB в архитектуре IoT (продолжение)

В рамках дисциплины рассматриваются три практически полезных варианта: обмен через CSV-файлы, сетевой обмен через API и прямое взаимодействие посредством совместимых пакетов MathWorks. Выбор зависит от наличия оборудования, лицензии, сетевой инфраструктуры и требований к частоте обновления.

2 Аппаратные интерфейсы и электрическая безопасность

Контакты GPIO предназначены для цифрового ввода-вывода. Типичный логический уровень GPIO Raspberry Pi составляет 3,3 В; подача 5 В на вход может повредить устройство. Нагрузки с большим током, двигатели и реле подключают через подходящие драйверы и схемы согласования, а не напрямую к контактам платы.

Шина I²C используется для подключения цифровых датчиков с адресацией; SPI — для высокоскоростного синхронного обмена; UART — для последовательного обмена байтами. Ethernet и Wi-Fi применяются для передачи данных в локальную сеть и облачные сервисы. Выбор интерфейса определяется характеристиками конкретного датчика.

Большинство плат Raspberry Pi не имеют встроенного универсального аналогового входа. Для аналогового сигнала, например напряжения 0–3,3 В, необходим внешний АЦП или модуль датчика с цифровым интерфейсом. Разрядность АЦП, опорное напряжение и электрическая схема влияют на точность преобразования.

2 Аппаратные интерфейсы и электрическая безопасность (продолжение)

До подключения необходимо изучить документацию конкретной модели Raspberry Pi и датчика, проверить питание, общую землю, распиновку и допустимые уровни. При монтаже выключают питание. Для промышленного применения также оценивают помехозащищённость, гальваническую развязку и условия эксплуатации.

Пример: цифровой температурный датчик по I^2C передаёт готовое числовое значение; аналоговый терморезистор требует измерительной схемы и АЦП. В обоих случаях MATLAB получает уже подготовленные числовые данные, но погрешности и процедура калибровки различаются.

3 Программная подготовка Raspberry Pi и MATLAB

На Raspberry Pi обычно устанавливают совместимую операционную систему, настраивают сеть, обновления и интерфейсы периферии. Для локального сбора данных можно использовать Python или другую поддерживаемую среду. Установка и включение интерфейсов зависят от версии операционной системы и модели платы.

В MATLAB Online преподаватель и студент создают папку проекта, сохраняют скрипты и тестовые CSV-файлы. Важно помнить, что MATLAB Online выполняется в облаке: браузерное окно само по себе не предоставляет прямого доступа к локальным GPIO или USB-портам.

Прямое соединение MATLAB с Raspberry Pi может быть доступно через соответствующие MathWorks support packages. Их установка, сетевые требования и совместимость зависят от версии MATLAB, лицензии, операционной системы и модели Raspberry Pi. Перед занятием следует проверить актуальную документацию.

Для воспроизводимости проекта рекомендуется зафиксировать модель Raspberry Pi, версию операционной системы, версию MATLAB, используемый интерфейс, период опроса, формат файла и список зависимостей. Это особенно важно, когда студенты работают на разных устройствах.

3 Программная подготовка Raspberry Pi и MATLAB (продолжение)

На первом этапе удобно использовать искусственно сформированные данные, чтобы проверить вычислительный тракт. После этого подключается реальный датчик; алгоритм анализа в MATLAB остаётся практически неизменным.

4

Архитектуры сопряжения: локальная, файловая и сетевая

Локальная архитектура предполагает, что вычисления и сбор данных выполняются на одном устройстве или в одной локальной сети. Она удобна при отладке, поскольку не зависит от внешнего облачного сервиса. При этом MATLAB может выполняться на отдельном компьютере, а Raspberry Pi — выступать источником измерений.

Файловая архитектура разделяет сбор и анализ по времени. Raspberry Pi сохраняет CSV-файл, который затем импортируется в MATLAB. Это простой и надёжный вариант для учебных экспериментов, но он не обеспечивает мгновенную визуализацию без дополнительного механизма передачи файлов.

Сетевая архитектура предполагает обмен по HTTP, MQTT или другому протоколу. Raspberry Pi может отправлять данные серверу, а MATLAB — читать их через API. Здесь возникают дополнительные задачи: аутентификация, задержки, повторная отправка, синхронизация часов и защита канала.

При выборе архитектуры следует учитывать частоту данных, допустимую задержку, объём измерений, доступность интернета, необходимость автономной работы и стоимость поддержки. Для низкочастотного мониторинга температуры CSV или HTTP могут быть достаточны; для высокочастотной вибродиагностики требуются иные решения по буферизации и пропускной способности.

4

Архитектуры сопряжения: локальная, файловая и сетевая (продолжение)

Сравнение вариантов показывает, что «прямое соединение» не всегда является лучшим: простая файловая схема может быть предпочтительнее для лабораторной работы, где главная цель — изучить математическую обработку и обеспечить повторяемость.

5 Передача измерений в CSV и чтение в MATLAB

CSV (Comma-Separated Values) — текстовый формат таблицы, в котором каждая строка содержит набор полей. Для IoT-измерений рекомендуется включать временную метку, значение, единицу измерения либо отдельное описание единиц и идентификатор датчика. Например: timestamp,temp_C,humidity_pct.

Если Raspberry Pi формирует файл, его можно перенести в MATLAB Online через загрузку в рабочую папку. В MATLAB функция readtable считывает таблицу с заголовками столбцов; затем можно обращаться к данным по именам переменных. Для обратной записи применяется writetable.

Важное условие — единый формат времени. Для простой учебной работы можно использовать время в секундах от начала эксперимента. Для реального распределённого проекта предпочтительны временные метки с часовым поясом или в UTC, чтобы исключить неоднозначность.

Нельзя считать отсутствие строки доказательством отсутствия физического события. Пропуск записи может быть вызван сбоем связи, отключением питания или ошибкой программного обеспечения. Поэтому полезно сохранять порядковый номер измерения.

При обработке файла сначала проверяют названия столбцов, типы значений, число строк, монотонность времени и наличие NaN/Inf. Только после этого следует выполнять расчёты и строить графики.

5

Передача измерений в CSV и чтение в MATLAB (продолжение)

```
% MATLAB: импорт учебного файла
data = readtable('sensor_log.csv');
t = data.time_s;
T = data.temp_C;
assert(numel(t)==numel(T),'Размеры не совпадают');
assert(all(diff(t)>0),'Нарушен порядок времени');
valid = isfinite(T);
fprintf('Корректных точек: %d\n',sum(valid));
plot(t(valid),T(valid),'-o'); grid on;
xlabel('Время, с'); ylabel('Температура, °C');
```

6 Сетевой обмен: HTTP, MQTT и JSON

HTTP — протокол запросов и ответов, широко используемый веб-сервисами. При типичном сценарии Raspberry Pi отправляет POST-запрос с измерениями на сервер, а MATLAB получает данные через GET-запрос или специализированный API. Для защищённой передачи применяется HTTPS.

JSON — текстовый формат представления структурированных данных. Запись может содержать поля device_id, timestamp и temperature_C. JSON удобен для сетевых API, однако имена полей, единицы и временные метки должны быть согласованы заранее.

MQTT — протокол обмена сообщениями по модели публикации и подписки. Устройство публикует измерение в определённую тему, а клиент-подписчик получает сообщения через брокер. Этот подход удобен при наличии нескольких датчиков и потребителей данных.

Наличие MQTT или HTTP не гарантирует доставку каждого сообщения ровно один раз. В проекте необходимо определить правила повторной отправки, обработку дубликатов и порядок сообщений. Полезно включать в сообщение уникальный идентификатор и порядковый номер.

6 Сетевой обмен: HTTP, MQTT и JSON (продолжение)

Для обмена с MATLAB используют поддерживаемые функции и интерфейсы конкретного сервиса. Нельзя переносить пример с одного API на другой без проверки адресов, структуры сообщений и требований авторизации.

```
% MATLAB: пример разбора JSON-строки
payload = '{"device_id":"pi01","time_s":30,"temp_C":26.4}';
sample = jsondecode(payload);
fprintf('Устройство %s: %.1f °C\n', ...
        sample.device_id,sample.temp_C);
% Пример формирования JSON для передачи
message = struct('device_id','pi01', ...
                 'time_s',40,'temp_C',26.7);
encoded = jsonencode(message);
disp(encoded);
```

7 Прямое взаимодействие через пакеты MathWorks

Для отдельных версий MATLAB существуют аппаратные пакеты поддержки Raspberry Pi. Они предоставляют функции подключения и доступа к поддерживаемым интерфейсам. Перед использованием необходимо проверить совместимость модели платы, версии программного обеспечения и доступной лицензии.

При подходящей конфигурации MATLAB может обращаться к Raspberry Pi через сетевое подключение. В документации соответствующего пакета приведены способы создания объекта соединения и взаимодействия с периферией. Конкретные команды зависят от версии и поддерживаемых функций.

Для лабораторного занятия рекомендуется сначала проверить сетевую доступность устройства, затем выполнить простую операцию чтения или записи на безопасном тестовом интерфейсе и лишь после этого подключать реальный датчик.

Если MATLAB Online не поддерживает нужный аппаратный сценарий, можно выполнять сбор данных на Raspberry Pi локальной программой и передавать результаты в MATLAB через CSV или API. Такой вариант сохраняет учебную цель без привязки к конкретному оборудованию.

При документировании необходимо отделять программный пример, проверенный на учебных данных, от аппаратного эксперимента. Реальный эксперимент требует указания модели датчика, схемы подключения, условий измерений и полученных результатов.

8 Временные метки, частота опроса и качество данных

Период дискретизации Δt определяет интервал между измерениями, а частота $f_s = 1/\Delta t$ — число отсчётов в секунду. При Δt=5 с частота равна 0,2 Гц. Для медленно изменяющейся температуры этого может быть достаточно, но для вибрации оборудования — обычно нет.

Если измеряемый сигнал содержит быстрые изменения, слишком редкий опрос может пропустить важные события. При выборе частоты учитывают динамику объекта, характеристики датчика, требования анализа и ограничения сети.

В распределённой системе время формирования показания, время отправки и время получения различаются. При анализе задержек полезно хранить хотя бы метку измерения и метку приёма, а также оценивать разность между ними.

Для обнаружения пропусков сравнивают фактические интервалы diff(t) с ожидаемым периодом. Однако неравномерность времени может быть нормальной особенностью системы, поэтому правило контроля необходимо выбирать с учётом допустимого джиттера.

8

Временные метки, частота опроса и качество данных (продолжение)

Перед вычислением среднего проверяют корректность данных. Для обнаружения NaN и Inf используется isfinite, для диапазона — сравнение с физически допустимыми границами, для пропусков — анализ временных меток.

```
t = [0 5 10 15 25 30]; % учебные секунды
T = [24.0 24.2 24.5 24.8 25.4 25.7];
dtExpected = 5;
intervals = diff(t);
missingIntervals = find(intervals > 1.5*dtExpected);
disp(intervals);
fprintf('Подозрительных интервалов: %d\n', ...
        numel(missingIntervals));
```

9 Обработка ошибок и информационная безопасность

Основные источники отказов — неправильное питание, нарушение соединения, недоступность датчика, потеря Wi-Fi, ошибка формата данных и недостаток памяти. Для каждой категории желательно предусмотреть диагностическое сообщение и безопасное поведение.

При кратковременной потере сети устройство может сохранять данные локально и передавать их позднее. Чтобы избежать дублирования, записи снабжают порядковыми номерами или уникальными идентификаторами. Буфер должен иметь ограниченный размер и понятную политику переполнения.

Для сетевых API следует применять защищённые соединения, разграничивать права чтения и записи, не включать ключи и пароли в открытые отчёты и не хранить их в публичных репозиториях. При доступе по SSH необходимо использовать надёжную аутентификацию и ограничивать сетевую поверхность устройства.

Программная обработка не должна автоматически трактовать пропущенное измерение как нулевое значение. Например, подстановка нуля вместо отсутствующей температуры может создать ложный сигнал об охлаждении.

9

Обработка ошибок и информационная безопасность (продолжение)

Если устройство участвует в управлении физическим процессом, при отказе сети и сервера оно должно переходить в заранее определённое безопасное состояние. Облачная аналитика не должна заменять локальную аварийную защиту.

10

Сквозной пример: температурный мониторинг

Постановка задачи: Raspberry Pi регистрирует температуру каждые 10 секунд и сохраняет файл sensor_log.csv. MATLAB импортирует данные, рассчитывает среднюю и максимальную температуру, отмечает превышения порога 28 °C и строит график. Для воспроизводимости сначала используется учебный набор, а не реальные показания.

Шаг 1. Формируется таблица из восьми отсчётов: время от 0 до 70 секунд, температура от 24,8 до 29,1 °C. В реальном проекте эти значения должны заменяться показаниями датчика. Шаг 2. Проверяются размеры и конечность чисел. Шаг 3. Выполняется вычисление показателей.

Шаг 4. Результаты визуализируются с подписью осей и горизонтальной линией порога. Шаг 5. Составляется краткий отчёт: число измерений, среднее, максимум, число превышений и описание возможных причин отклонения.

Шаг 6. Для перехода к Raspberry Pi создаётся отдельная программа сбора данных, которая записывает ту же структуру таблицы. Таким образом, аналитический MATLAB-скрипт не зависит от того, были ли данные введены вручную или получены аппаратно.

10

Сквозной пример: температурный мониторинг (продолжение)

Следует различать сигнал превышения и диагностическое заключение. Превышение заданного порога требует проверки, но не доказывает неисправность оборудования без учёта условий процесса и характеристик датчика.

```
% Полный воспроизводимый пример MATLAB
clear; clc; close all;
time_s = (0:10:70).';
temp_C = [24.8;25.1;25.6;26.0;26.7;27.5;28.4;29.1];
sensorData = table(time_s,temp_C);
writetable(sensorData,'sensor_log.csv');
data = readtable('sensor_log.csv');
t = data.time_s;
T = data.temp_C;
assert(numel(t)==numel(T),'Несовпадение длин');
assert(all(diff(t)>0),'Время не возрастает');
assert(all(isfinite(T)),'Некорректные измерения');
limit = 28;
```

10

Сквозной пример: температурный мониторинг (продолжение)

```
meanT = mean(T);
maxT = max(T);
alarms = T > limit;
fprintf('Средняя: %.2f °C\n',meanT);
fprintf('Максимум: %.2f °C\n',maxT);
fprintf('Превышений: %d\n',sum(alarms));
figure;
plot(t,T,'-o','LineWidth',1.5); hold on;
yline(limit,'--','Порог 28 °C');
grid on; xlabel('Время, с');
ylabel('Температура, °C');
title('Учебный мониторинг Raspberry Pi — MATLAB');
```

11 Визуализация, контроль порогов и оценка точности

График измерений должен включать название, единицы осей и обозначение порогов. При сравнении нескольких датчиков полезно использовать легенду и отдельные обозначения каналов. Слишком плотная визуализация может затруднить выявление отклонений.

Среднее значение $\bar{x}=(1/n)\Sigma x_i$ описывает центральный уровень ряда, а выборочное стандартное отклонение $s=\mathrm{sqrt}[\Sigma(x_i-\bar{x})^2/(n-1)]$ — разброс. Для оценки динамики используется скорость изменения $(x_{i+1}-x_i)/(t_{i+1}-t_i)$.

Погрешность датчика и погрешность вычислений различаются. Даже идеально выполненный MATLAB-расчёт не устраняет систематическую ошибку датчика. При наличии эталонных измерений можно вычислить MAE = mean(abs(y−yref)) и RMSE = sqrt(mean((y−yref).^2)).

Сглаживание временного ряда уменьшает случайные колебания, но может задерживать выявление коротких пиков. Поэтому защитный контроль нельзя строить только на сглаженных данных без отдельной оценки риска.

При сравнении результатов нужно указывать, являются ли данные измеренными, модельными или искусственно сформированными. Учебный пример демонстрирует алгоритм, но не подтверждает метрологические характеристики конкретного устройства.

12

Порядок лабораторного внедрения и типичные неисправности

Этап 1 — проверить питание, модель платы и документацию датчика. Этап 2 — проверить локальное считывание и вывести несколько значений. Этап 3 — сохранить таблицу с временными метками и единицами. Этап 4 — импортировать файл в MATLAB. Этап 5 — выполнить обработку и визуализацию.

Если датчик не отвечает, сначала проверяют питание и правильность интерфейса, затем адрес устройства, разрешение интерфейса и физические соединения. Если MATLAB не читает файл, проверяют текущую папку, имя файла, разделители, заголовки столбцов и типы данных.

Если в графике появляются разрывы, следует проверить NaN, неверные временные метки и пропущенные строки. Если расчёт даёт физически невозможные значения, необходимо проверить единицы измерения, коэффициенты калибровки и диапазон датчика.

Если сетевой обмен работает нестабильно, проверяют качество соединения, адрес сервера, тайм-ауты, аутентификацию и механизм повторной отправки. Диагностические сообщения следует сохранять отдельно от измерений.

Результатом лабораторного занятия должен стать воспроизводимый проект: схема подключения, описание оборудования, исходный код, файл данных, график, вычисленные показатели и вывод о работе системы.

3 Заключение

Рассмотрены аппаратные и программные основы сопряжения Raspberry Pi с MATLAB, файловый и сетевой обмен, особенности MATLAB Online, временная синхронизация, контроль качества измерений и обработка отказов. Сквозной пример показывает переход от структурированных показаний датчика к вычислению показателей и визуализации. Полученные знания создают основу для дальнейшего изучения IoT-платформ и организации удалённого мониторинга технологических объектов.

Контрольные вопросы (1/2)

1. Как распределяются функции между Raspberry Pi и MATLAB?

2. Чем мониторинг отличается от управления в реальном времени?

3. Какие аппаратные интерфейсы доступны на Raspberry Pi?

4. Почему опасно подавать 5 В на GPIO?

5. Как подключить аналоговый датчик к Raspberry Pi?

6. Какие параметры оборудования необходимо фиксировать в отчёте?

7. Чем отличается файловый обмен от сетевого?

8. Какие поля должен содержать журнал измерений?

9. Для чего в MATLAB используется readtable?

10. Как проверить монотонность временных меток?

11. Какова роль JSON в IoT?

12. В чём различие между HTTP и MQTT?

13. Что необходимо проверить перед применением пакета поддержки Raspberry Pi?

14. Почему MATLAB Online не всегда может напрямую обращаться к GPIO?

15. Как вычисляется частота дискретизации?

16. Почему важна синхронизация времени?

Контрольные вопросы (2/2)

17. Как обнаружить пропущенные измерения?

18. Какие действия следует выполнить при появлении NaN?

19. Как организовать буферизацию при потере сети?

20. Почему нельзя публиковать ключи API?

21. Как вычислить число превышений порога в MATLAB?

22. Чем отличается погрешность датчика от ошибки вычислений?

23. Для чего используются MAE и RMSE?

24. Какие этапы включает лабораторная проверка системы?

25. Какие материалы должны войти в отчёт по сопряжению Raspberry Pi и MATLAB?

Д Задания для самостоятельной работы

1. Составьте структурную схему системы «датчик — Raspberry Pi — CSV — MATLAB» и опишите функции каждого узла.

2. Создайте в MATLAB таблицу из 20 температурных отсчётов с периодом 5 с, сохраните её в CSV и повторно импортируйте.

3. Добавьте в учебный ряд пропуск NaN и разработайте корректную проверку данных без подстановки нуля.

4. Для двух датчиков температуры постройте общий график, рассчитайте средние и сравните отклонения от порога.

5. Разработайте формат JSON-сообщения для температуры, влажности, времени и идентификатора устройства.

6. Опишите алгоритм действий Raspberry Pi при потере сети на 2 минуты с учётом буферизации и последующей передачи данных.

Рекомендуемые источники

• Силлабус дисциплины «Проект MATLAB IoT», тема лекции №6 «Задачи сопряжения мини-ПК Raspberry Pi с данными MATLAB».

• MathWorks. MATLAB Documentation: https://www.mathworks.com/help/matlab/

• MathWorks. Raspberry Pi Support: https://www.mathworks.com/hardware-support/raspberry-pi-matlab.html

• Raspberry Pi Documentation: https://www.raspberrypi.com/documentation/

• MathWorks. ThingSpeak Documentation: https://www.mathworks.com/help/thingspeak/