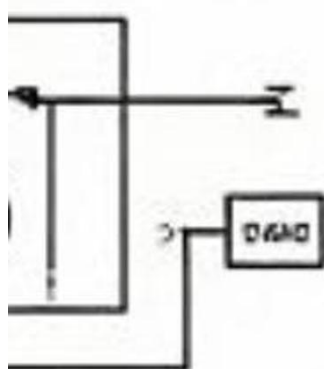


а составлен с учетом  
и не в соответствии с  
не в соответствии с



а в соответствии с  
с учетом - не  
и в соответствии с  
и в соответствии с

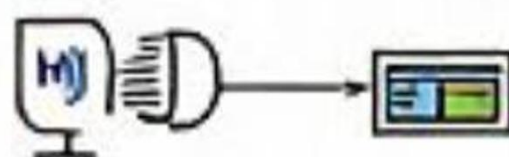


в соответствии с  
и в соответствии с  
и в соответствии с

и в соответствии с  
и в соответствии с  
и в соответствии с



Проектирование и  
разработка и  
разработка и



Проектирование и  
разработка и  
разработка и

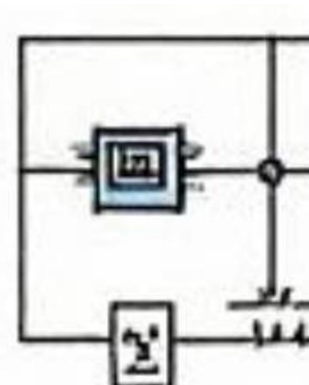
Проектирование и  
разработка и  
разработка и



Проектирование и  
разработка и  
разработка и

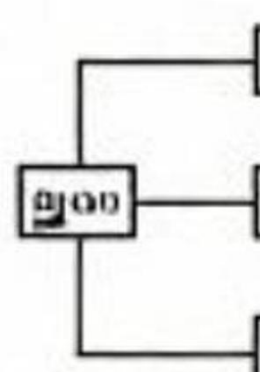
Проектирование и  
разработка и  
разработка и

Проектирование и  
разработка и  
разработка и



Проектирование и  
разработка и  
разработка и

Проектирование и  
разработка и  
разработка и



Проектирование и  
разработка и  
разработка и

Проектирование и  
разработка и  
разработка и

# Курс разработчика умного дома

Учитель Начальной

Учитель Начальной

**Курс разработчика умного дома**

«Автор»

2026

## **Начальной У. Ш.**

Курс разработчика умного дома / У. Ш. Начальной — «Автор», 2026

Практический курс для тех, кто хочет проектировать, программировать и запускать системы умного дома с нуля. Двадцать глав охватывают полный путь разработчика: от выбора железа (ESP32, Raspberry Pi) и протоколов (MQTT, Zigbee, Matter, Thread) до настройки Home Assistant, прошивки устройств на ESPHome, создания автоматизаций и обеспечения безопасности. В книге — рабочие примеры кода, схемы архитектуры, чек-листы и пошаговые проекты: защита от протечек, освещение по движению, климат-контроль, видеонаблюдение на Frigate, локальный голос. Особое внимание уделено локальному контролю без облачной зависимости, сетевой сегментации и масштабированию системы. Издание 2026 года учитывает актуальные стандарты Matter и требования к кибербезопасности. Подходит новичкам с базовыми знаниями программирования и инженерам, которые хотят собирать надёжные, приватные и полностью контролируемые решения «под ключ».

© Начальной У. Ш., 2026

© Автор, 2026

# Содержание

|                                                               |    |
|---------------------------------------------------------------|----|
| Глава 1. Введение в умный дом и профессию разработчика        | 5  |
| 1.1. Что такое умный дом                                      | 5  |
| 1.2. Эволюция умного дома                                     | 6  |
| 1.3. Зачем нужен разработчик умного дома                      | 7  |
| 1.4. Что вы изучите в курсе                                   | 8  |
| 1.5. Необходимые навыки и оборудование для старта             | 9  |
| Глава 2. Архитектура систем умного дома                       | 10 |
| 2.1. Классическая трёхуровневая модель                        | 10 |
| 2.2. Централизованная, распределённая и гибридная архитектура | 11 |
| 2.3. Локальный контроль как приоритет                         | 12 |
| 2.4. Основные роли компонентов                                | 13 |
| 2.5. Пример типичной архитектуры квартиры 2026                | 14 |
| Глава 3. Аппаратные платформы: Raspberry Pi, ESP32, Arduino   | 15 |
| 3.1. Raspberry Pi — мозг системы                              | 15 |
| 3.2. ESP32 — рабочая лошадка конечных устройств               | 16 |
| Конец ознакомительного фрагмента.                             | 17 |

# **Учитель Начальной**

## **Курс разработчика умного дома**

### **Глава 1. Введение в умный дом и профессию разработчика**

#### **1.1. Что такое умный дом**

Умный дом (Smart Home) — это жилое пространство, в котором устройства и системы объединены в единую сеть и способны автоматически или по команде управлять освещением, климатом, безопасностью, мультимедиа, энергопотреблением и другими функциями.

Современный умный дом — это не просто набор «умных» лампочек из магазина. Это полноценная IoT-система со следующими уровнями:

- Физический уровень — датчики, реле, клапаны, камеры, исполнительные механизмы
- Сетевой уровень — Wi-Fi, Zigbee, Thread, Ethernet, MQTT
- Уровень управления — контроллер (Home Assistant, OpenHAB, LoXone и др.)
- Уровень представления — мобильное приложение, дашборд, голос, физические панели

Ключевые свойства хорошей системы: локальность критических функций (работа без интернета), надёжность и предсказуемость, возможность ручного управления в любой момент, безопасность и приватность данных, масштабируемость и документированность.

## 1.2. Эволюция умного дома

Краткая хронология: 1970–90-е — X10 и проприетарные проводные системы; 2000-е — Z-Wave, Insteon; 2010-е — массовые экосистемы Nest, Philips Hue, Xiaomi; 2016+ — Home Assistant и ESPHome; 2022–2024 — Matter 1.0–1.2 и Thread; 2025–2026 — Matter 1.3–1.5 (камеры, энергия, EV), локальный AI, ужесточение требований безопасности (EU Cyber Resilience Act).

## 1.3. Зачем нужен разработчик умного дома

Рынок растёт, но готовые «коробочные» решения часто зависят от облака производителя, плохо стыкуются между собой, не дают тонкой настройки сценариев и собирают данные пользователя. Разработчик умеет спроектировать архитектуру под конкретный объект и бюджет, выбрать протоколы и железо, написать прошивки, настроить Home Assistant, обеспечить безопасность и сдать систему клиенту с документацией.

Востребованные роли: DIY-консультант и установщик для частных домов, инженер-проектировщик коммерческих объектов, разработчик прошивок и устройств, интегратор (KNX + HA, BACnet), техническая поддержка и обучение.

## **1.4. Что вы изучите в курсе**

- Объяснить архитектуру IoT-системы любому заказчику
- Выбрать и настроить Raspberry Pi / ESP32 под задачу
- Подключить датчики и реле, написать простую прошивку
- Развернуть Home Assistant, MQTT, Zigbee2MQTT
- Создать автоматизации на YAML, Node-RED и Python
- Настроить Matter/Thread и голосовое управление
- Сегментировать сеть и защитить систему
- Собрать полноценный проект «под ключ» и задокументировать его

## **1.5. Необходимые навыки и оборудование для старта**

Минимальные знания: базовый Python или C/C++, понимание IP-сетей, основы электроники (напряжение, ток, реле). Рекомендуемый стартовый набор: ESP32 DevKit (2–3 шт.), Raspberry Pi 4/5 + SSD, USB-координатор Zigbee, датчики DHT22/BME280, PIR, геркон, модуль реле, мультиметр.

## **Глава 2. Архитектура систем умного дома**

### **2.1. Классическая трёхуровневая модель**

Устройство → Шлюз/Хаб → Контроллер/Облако/Приложение. На нижнем уровне — конечные устройства (датчики, выключатели). Средний уровень — шлюз, который переводит протоколы (Zigbee → MQTT) и выполняет локальную логику. Верхний уровень — облако или локальный сервер с UI и голосовыми ассистентами.

## **2.2. Централизованная, распределённая и гибридная архитектура**

Централизованная (Home Assistant, Loxone): вся логика на одном контроллере — проще отлаживать, но при падении контроллера часть логики останавливается. Распределённая (edge): устройства сами принимают решения — надёжнее при потере связи. В 2026 году доминирует гибрид: локальный контроллер + edge-логика на ESP + опциональное облако.

## **2.3. Локальный контроль как приоритет**

Правило №1: свет, отопление, охрана и защита от протечек должны работать без интернета. Home Assistant + MQTT + ESPHome дают полную локальность. Matter и Thread тоже ориентированы на локальную работу.

## 2.4. Основные роли компонентов

| Роли компонентов в системе умного дома |                         |                         |
|----------------------------------------|-------------------------|-------------------------|
| Роль                                   | Примеры                 | Задачи                  |
| Контроллер                             | Home Assistant, OpenHAB | Логика, UI, интеграции  |
| Брокер                                 | Mosquitto, EMQX         | Обмен сообщениями MQTT  |
| Координатор                            | Zigbee2MQTT, ZHA        | Радиосеть Zigbee/Thread |
| Устройство                             | ESPHome, Tasmota        | Сбор данных, управление |
| Аналитика                              | InfluxDB, Grafana       | Долгосрочное хранение   |
| Голос                                  | Assist, Алиса, Alexa    | Естественный интерфейс  |
|                                        |                         |                         |

Рисунок: роли компонентов в системе умного дома

## 2.5. Пример типичной архитектуры квартиры 2026



Рисунок: типичная архитектура умного дома

## **Глава 3. Аппаратные платформы: Raspberry Pi, ESP32, Arduino**

### **3.1. Raspberry Pi — мозг системы**

Raspberry Pi 5 (4/8 ГБ) — лучший выбор по производительности. Pi 4 всё ещё отлично работает. Home Assistant Green/Yellow — готовые решения. Почему SSD вместо microSD: карты памяти деградируют от постоянной записи, SSD даёт скорость и надёжность.

## **3.2. ESP32 — рабочая лошадка конечных устройств**

ESP32 — двухъядерный MCU с Wi-Fi и Bluetooth. ESP32-C3/C6 — современные варианты с поддержкой Thread/Matter. ESP32-S3 — с AI-ускорителем. Питание 3.3 В, deep sleep — микроамперы. Для прошивки используйте ESPHome (YAML) или PlatformIO (C++).

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.