



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«МИРЭА – Российский технологический университет»  
РТУ МИРЭА

Институт: Информационных технологий (ИТ)

Кафедра: Базовая кафедра №234 — Управляющих ЭВМ

**КУРСОВАЯ РАБОТА**

по  
дисциплине

Техническое обеспечение систем управления  
технологическими процессами  
(наименование дисциплины)

Тема  
курсовой  
работы:

Система автоматизации умного загородного дома на базе  
компонентов Arduino и Raspbery  
PI.

Студент группы:

ИКМО-05-23 Миронов Д.С.

(учебная группа, фамилия, имя,  
отчество)

(подпись)

Руководитель  
курсовой  
работы:

доцент, к.т.н., Глухов А.В.

(должность, звание, ученая степень, фамилия, имя,  
отчество)

(подпись)

Работа предоставлена к защите

до «\_\_» июня 2024 г.

Допущена к защите

до «\_\_» июня 2024 г.

## ОГЛАВЛЕНИЕ

<b>ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....</b>	<b>5</b>
<b>Общие требования: .....</b>	<b>5</b>
<b>Задания на проектирование: .....</b>	<b>5</b>
<b>ВВЕДЕНИЕ .....</b>	<b>6</b>
<b>Актуальность темы.....</b>	<b>6</b>
1.    Номенклатура применяемых счетчиков ресурсов .....	7
1.1.    Анализ счетчиков воды .....	7
1.1.1.    Электронный счетчик воды YF-S201.....	7
1.1.2.    Электронный счетчик воды Аквафор Водометр .....	7
1.1.3.    Электронный счетчик Ду15 RS-485 модель 1 .....	8
1.2.    Анализ счетчиков электроэнергии.....	8
1.2.1.    PZEM-004T.....	8
1.2.2.    Электросчетчик СЭ-310.....	9
1.3.    Анализ счетчиков газа .....	9
1.3.1.    Счетчик газа Gallus G4 .....	9
1.3.2.    Счетчик газа СГ СГК-1,6.....	9
1.3.3.    Газовый счетчик «Тепловономер ВК G4» .....	10
1.4.    Анализ источников бесперебойного питания .....	10
1.4.1.    Общая мощность системы.....	10
1.4.2.    Расчёт потребляемой энергии за 7 дней .....	11
1.4.3.    Опции источников бесперебойного питания .....	11
1.4.3.1.    Аккумуляторная система на основе гелевых или AGM аккумуляторов .....	11
1.4.3.2.    Инвертор + аккумуляторы.....	11

1.4.3.3.	Готовые решения с большим аккумулятором .....	11
2.	Спецификация на закупку с расчетом затрат .....	13
2.1.	Сравнение анализируемых счетчиков .....	13
2.2.	Выбор спецификации.....	15
2.2.1.	Счётчики воды .....	15
2.2.2.	Счётчики электроэнергии.....	16
2.2.3.	Счётчики газа .....	16
2.2.4.	Обоснование выбора.....	16
2.2.4.1.	YF-S201 (Счётчик воды).....	16
2.2.4.2.	PZEM-004T (Счётчик электроэнергии) .....	18
2.2.4.3.	Gallus G4 (Счётчик газа).....	18
2.2.4.4.	EcoFlow DELTA Pro (Источников бесперебойного питания) .....	19
2.2.4.5.	Датчик температуры (DS18B20) .....	20
2.2.4.6.	Датчик утечки газа (MQ-135).....	21
2.2.4.7.	Модуль Vaillant VR34.....	22
2.2.4.8.	Аккумуляторы AGM 12V 100 Ah .....	23
2.2.4.9.	Инвертор 12V → 220V AC и 5V DC .....	23
2.2.5.	Заключение.....	23
3.	Схема соединений.....	24
4.	Подключение Raspberry Pi к Arduino .....	26
5.	Шаги по реализации передачи данных с Arduino на Raspberry Pi и мониторинга через телефон.....	27
6.	Программное обеспечение .....	31
7.	Описание применяемых программных компонентов .....	31
<b>Вывод</b>	.....	31

<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>33</b>
Приложение .....	35

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Общие требования:

Разработать проект системы автоматизации умного загородного дома на базе компонентов Arduino и Raspberry PI.

Система состоит из блока управления (проектируется в рамках работы), газового котла с коммутационным модулем Vaillant VR34. Система должна обеспечивать автоматический сбор данных со следующих источников:

- 1 счетчик воды;
- 1 счетчик электроэнергии;
- 1 счетчик газа
- Уличная температура;
- Температура в помещениях (не менее 4-х точек);
- Температура контура отопления (подача, обратка);
- Датчик превышения допустимой концентрации газа;
- Ошибка газового котла от устройства VR34.

Система должна обеспечивать автоматическое управление мощностью котла через устройство VR34 по датчику уличной температуры и температуры в помещении.

Система должно обеспечивать контроль параметров дома и выдавать предупредительные сообщения в telegram/email в случае возникновения аварийных ситуаций (рассмотреть эти ситуации).

Система должна обеспечивать удаленный мониторинг на базе облачных решений и в виде мобильного приложения (предложить из имеющихся IoT систем, совместимых с Arduino и raspberry для реализации обмена с облаком и мобильным приложением).

## Задания на проектирование:

- предложить номенклатуру применяемых счетчиков ресурсов, совместимых с Arduino и прочих датчиков
- разработать спецификацию на закупку с расчетом затрат;
- разработать схему соединений;
- определить и обосновать выбор программного обеспечения;
- разработать диаграмму потоков данных;
- разработать подробное описание применяемых программных компонентов (программный код приветствуется).

# **ВВЕДЕНИЕ**

## **Актуальность темы**

В современных условиях автоматизация технологических процессов играет ключевую роль в повышении эффективности и безопасности промышленных объектов. Распределенные системы управления, объединяющие разнообразные датчики и мониторы, позволяют оперативно собирать и анализировать данные, что значительно улучшает качество управления и позволяет своевременно реагировать на внештатные ситуации.

## **1. Номенклатура применяемых счетчиков ресурсов**

### **1.1. Анализ счетчиков воды**

Счетчики должны иметь импульсный выход, который можно подключить к Arduino для считывания данных.

#### **1.1.1. Электронный счетчик воды YF-S201**

Ардуино датчик расхода воды и не агрессивных жидкостей [1] YF-S201-предназначен для измерения объема прокачанной через него воды. Формула для расчёта потока жидкости л/мин:  $Q=F/7.5$ , где F частота импульсов, Q - поток л/мин. Можно использовать коэффициент: 1 литр воды это примерно 516 импульсов. Расходомер воды довольно часто используют в проектах Arduino контроллерами. Счетчик жидкости можно использовать как для подсчета потока воды, так и для измерения объема пройденной через него жидкости.

- Цена: 840 руб.
- Диапазон рабочих температур для учета холодной воды (Тис) °С: от 0 до 40.
- Диапазон рабочих температур для учета горячей воды (Тис) °С: от 5 до 95.

#### **1.1.2. Электронный счетчик воды Аквафор Водометр**

Цифровые счетчики воды ВСЦ[2] одноструйные цифровые с диаметрами условного прохода DN 15, 20 - предназначены для коммерческого учета расхода холодной и горячей воды в системах водоснабжения, отвечающей требованиям, изложенным в СанПиН 1.2.3685-21, и сетевой воды, отвечающей требованиям по качеству, изложенным в СП 124.13330.2012, и протекающей в системах холодного и горячего водоснабжения при давлении до 1,6 МПа (16 кгс/см<sup>2</sup>) в диапазоне температур от +5 до +90 °С.

- Цена: 3700 руб.
- Диапазон рабочих температур для учета холодной воды (Тис) °С: от 5 до 90.

- Диапазон рабочих температур для учета горячей воды (Тис) °С: от 5 до 90.

### **1.1.3. Электронный счетчик Ду15 RS-485 модель 1**

Счетчики воды электронные «Пульсар»[3] предназначены для измерений объема холодной или горячей воды, протекающей в трубопроводах систем холодного и горячего водоснабжения.

Принцип работы счетчика состоит в измерении числа оборотов крыльчатки, вращающейся под действием потока протекающей воды. Счетный механизм имеет электронный датчик оборотов крыльчатки. Сигнал с датчика поступает на микропроцессорное устройство, которое вычисляет объем воды, прошедшей через счетчик. Значение объема отображается на индикаторном устройстве.

- Цена: 5300 руб.
- Диапазон рабочих температур для учета холодной воды (Тис) °С: от 5 до 95.
- Диапазон рабочих температур для учета горячей воды (Тис) °С: от 5 до 95.

## **1.2. Анализ счетчиков электроэнергии**

Можно выбрать счетчик, поддерживающий протокол Modbus RTU. Это облегчит считывание данных счетчика с помощью Arduino.

### **1.2.1. PZEM-004T**

Модуль для замера напряжения, тока, частоты, мощности и суммарно потребленной электроэнергии в кВт/ч. Связь с модулем через UART интерфейс.

- Цена: 1600 руб.
- Измеряемое напряжение: 80-260В
- Измеряемый ток: до 100А
- Измеряемая частота: 45-65Гц
- Измеряемая мощность: до 22КВт
- Гарантийный срок эксплуатации, лет: 3



### **1.2.2. Электросчетчик СЭ-310**

Трехфазный многофункциональный электросчетчик серии «СЕ»[5]. Устанавливается на din-рейку и в щиток (счетчик комплектуется двумя крышками).

Осуществляет измерение и учет активной электрической энергии в трехфазных четырехпроводных цепях переменного тока с возможностью учета в одном или двух направлениях. Организация многотарифного учета электроэнергии на промышленных предприятиях и объектах энергетики с передачей накопленной информации через оптопорт и цифровой интерфейс RS485.).

- Цена: 4600 руб.
- Измеряемое напряжение: 230-400В
- Измеряемый ток: до 100А
- Гарантийный срок эксплуатации, лет: 4

### **1.3. Анализ счетчиков газа**

Можно выбрать счетчик, с импульсным выходом.

#### **1.3.1. Счетчик газа Gallus G4**

Счетчики газа малогабаритные СГБМ[6] предназначены для измерения объема газа при учете потребления газа индивидуальными потребителями в жилищно-коммунальном и бытовом хозяйстве.

Счетчики газа СГБМ отличаются малыми габаритами и возможностью установки как на вертикальном, так и на горизонтальном опуске газопровода.

В эксплуатации счетчики не являются источником шума, электромагнитных помех, вибрации и загазованности.

- Цена: 2900 руб.
- Объем цикла 1,2 дм<sup>3</sup>
- Средняя наработка на отказ: 100 000 час.
- Максимальное рабочее давление 50 кПа;
- Интервал между поверками 10 лет.

#### **1.3.2. Счетчик газа СГ СГК-1,6**

Компактный газовый счетчик [7], для плиты который в отличие от своих аналогов имеет уникальный дизайн и большой размер экрана. Он прекрасно подойдет людям всех возрастов, а в особенности тем, кто хочет видеть показатели счетчика большими. Идеален для людей в преклонном возрасте.

- Цена: 2700 руб.
- Средняя наработка на отказ: 90 000 час.
- Максимальное рабочее давление 50 кПа;
- Объем цикла 1,2 дм<sup>3</sup>
- Интервал между поверками 10 лет.

### **1.3.3. Газовый счетчик «Тепловодомер ВК G4»**

Газовый счетчик ВК G4 [8] представляет собой прибор учета, используемый для измерения количества потребляемого газа в быту и на предприятиях. Он построен по классической схеме с применением чувствительного диафрагменного механизма с кривошипно-шатунным механизмом. Поступательное движение диафрагм превращается во вращательное, после чего передается на измерительный механизм – его счетное табло располагается в верхней передней части корпуса. Газовый счетчик ВК G4 не содержит устройства термокомпенсации, поэтому сдача показаний осуществляется с поправкой на температуру эксплуатации.

- Цена: 4500 руб.
- Максимальное рабочее давление 50 кПа;
- Объем цикла 1,2 дм<sup>3</sup>
- Средняя наработка на отказ: 120 000 час.
- Интервал между поверками 12 лет.

## **1.4. Анализ источников бесперебойного питания**

### **1.4.1. Общая мощность системы**

- Raspberry Pi 4: до 10 Вт.
- Arduino Uno: около 1 Вт.

- Датчики и модули связи (датчики температуры, газ, вода, и т.д.): примерно 5 Вт.
- Модуль управления котлом Vaillant VR34: до 5 Вт.

Итого потребление: **около 20 Вт** на одну квартиру.

#### **1.4.2. Расчёт потребляемой энергии за 7 дней**

- Система потребляет около 20 Вт в час.
- За 24 часа:  $20 \text{ Вт} \times 24 = 480 \text{ Вт}\cdot\text{ч}$
- За 7 дней:  $480 \text{ Вт}\cdot\text{ч} \times 7 = 3360 \text{ Вт}\cdot\text{ч}$

Таким образом, для одной квартиры потребуется источник питания с ёмкостью не менее **3360 Вт·ч**.

#### **1.4.3. Опции источников бесперебойного питания**

##### **1.4.3.1. Аккумуляторная система на основе гелевых или AGM аккумуляторов**

- Самый надёжный вариант для длительного автономного питания.
- Например, аккумуляторы **AGM 12V 100 Ah**:
  - Ёмкость одного аккумулятора:  $12 \text{ В} \times 100 \text{ А}\cdot\text{ч} = 1200 \text{ Вт}\cdot\text{ч}$ .
  - Для 3360 Вт·ч потребуется минимум 3 таких аккумулятора (для одной квартиры).
  - Для всей системы из 4 квартир потребуется 12 аккумуляторов.

##### **1.4.3.2. Инвертор + аккумуляторы**

- Необходим инвертор для преобразования напряжения 12V в 220V или 5V (для питания Raspberry Pi и других устройств).
- Например, инвертор с выходом 220V и 5V, мощностью 500 Вт.

##### **1.4.3.3. Готовые решения с большим аккумулятором**

- Можно использовать готовые решения, такие как **портативные электростанции** (например, **EcoFlow DELTA Pro**), которые рассчитаны на длительное питание различных устройств.

- Например, **EcoFlow DELTA Pro** имеет ёмкость около 3600 Вт·ч, что достаточно для одной квартиры на 7 дней, но такие решения могут быть более дорогими.



Рисунок 1 - EcoFlow DELTA Pro

#### 1.4.3.4. Готовые решения с аккумулятором для счетчиков

Источник питания 5V — это источник автономного питания для 5 В плат Arduino, ESP32, Raspberry Pi и других нагрузок, требующих 5 вольт постоянного тока. Он позволит сделать Ваши устройства по настоящему мобильными.

В модуле имеется контроллер заряда Li-Po и Li-Ion аккумуляторов. Модуль исполнен в двух вариантах: со встроенным аккумулятором (Рис. 2) и без аккумулятора, что позволяет использовать иные, более компактные или более емкие аккумуляторы. Для варианта без аккумулятора, аккумулятор необходимо припаять к терминалам V bat.

Источник питания 5V выполнен в формате, совместимым с ПВХ конструктором

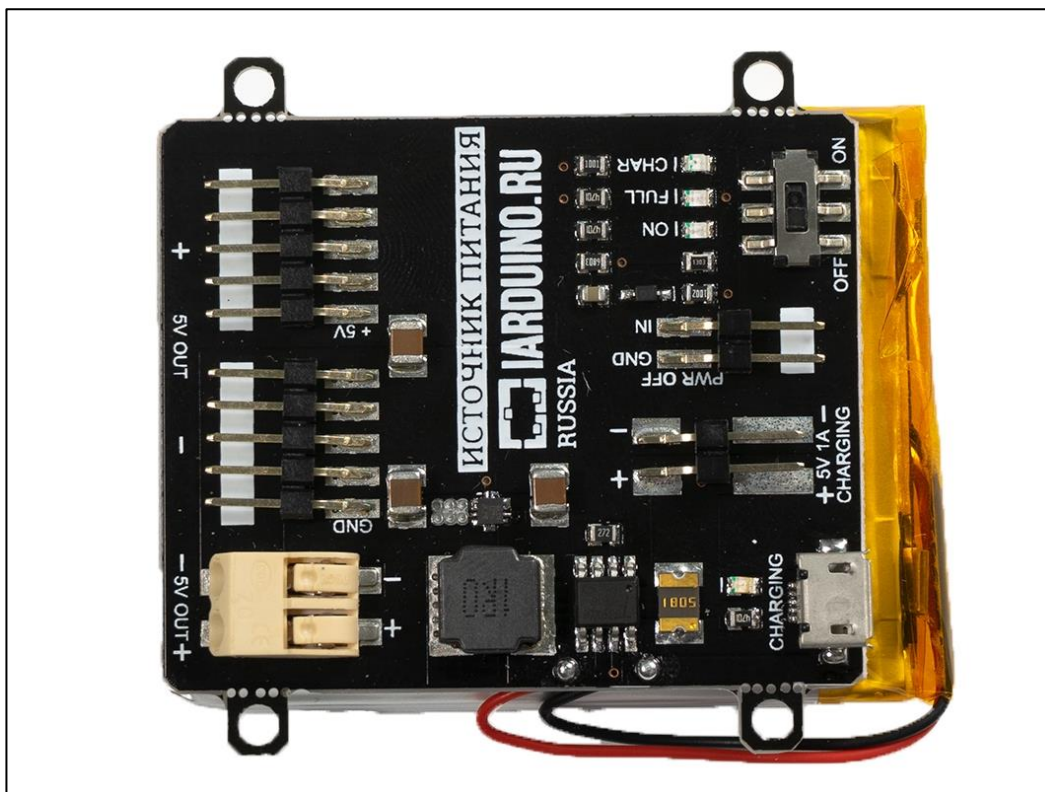


Рисунок 2 - Источник питания (Li-po, 3200 мА·ч)

## 2. Спецификация на закупку с расчетом затрат

### 2.1. Сравнение анализируемых счетчиков

Таблица 1 – Сравнимые счетчики воды

Название	Цена за шт.	Диапазон рабочих температур для учета холодной воды в °С	Диапазон рабочих температур для учета горячей воды в °С	Производитель
Электронный счетчик воды YF-S201	840 руб.	от 0 до 40	от 5 до 95	Россия
Электронный счетчик воды	3700 руб.	от 5 до 90	от 5 до 90	Россия

Аквафор Водометр				
Электронный счетчик Ду15 RS-485 модель 1	5300 руб.	от 5 до 95	от 5 до 95	Россия

Таблица 2 – Сравнимые счетчики электроэнергии

Название	Цена за шт.	Измеряемое напряжение	Измеряемый ток	Производитель
PZEM	1600 руб.	80-260В	100А	Россия
Электросчетчик СЭ-310	4600 руб.	230-400В	100А	Россия

Таблица 3 – Сравнимые счетчики газа

Название	Цена за шт.	Объем цикла	Гарантийный срок эксплуатации, лет	Производитель
Gallus G4	2900 руб.	1,2 дм <sup>3</sup>	10	Россия
Газовый счетчик "Сигнал СГК-1.6"	2700 руб.	1,2 дм <sup>3</sup>	10	Россия
Газовый счетчик "Тепловономер ВК G4"	4500 руб.	1,2 дм <sup>3</sup>	12	Россия

Таблица 4– Сравнимые источники бесперебойного питания

Название	Цена за шт.	Напряжение на выходе	Автономная работа
Xiaomi Mi Power Bank 3 Pro 20000 мАч	3000 руб.	до 45 Вт через USB-порты (подходит для прямого питания Raspberry Pi и Arduino)	до 10 часов для Raspberry Pi и Arduino
APC Back-UPS BE400-RS	7000–9000 рублей	220 В (через инвертор можно подключить Raspberry Pi и другие устройства)	от 30 минут до 1 часа для всего оборудования
Mean Well DRC-40	около 5000 рублей	12 В и 5 В	около 1–2 часов для Raspberry Pi, Arduino и датчиков

## 2.2. Выбор спецификации

### 2.2.1. Счётчики воды

Для данной курсовой работы были выбраны счётчики воды YF-S201 по следующим причинам:

- Совместимость с Arduino: Эти счётчики имеют импульсный выход, что упрощает интеграцию с микроконтроллером Arduino для автоматического сбора данных.
- Надёжность и точность: YF-S201 известны своей высокой точностью измерений и долговечностью.
- Стоимость: Эти счётчики имеют приемлемую цену (около 800 рублей), что важно для разработки экономически эффективной системы.

### **2.2.2. Счётчики электроэнергии**

Для учета электроэнергии был выбран счётчик PZEM-004T по следующим причинам:

- Совместимость с Arduino и Raspberry Pi: PZEM-004 позволяет легко интегрировать его с микроконтроллерами и одноплатными компьютерами.

### **2.2.3. Счётчики газа**

Для учета газа был выбран счётчик Gallus G4 по следующим причинам:

- Совместимость с Arduino: Счётчик Gallus G4 имеет импульсный выход, что позволяет легко подключить его к микроконтроллеру Arduino для сбора данных.
- Надёжность и точность: Этот счётчик обеспечивает высокую точность измерений и долговечность.
- Стоимость: Приемлемая цена (около 3500 рублей), что важно для создания экономически эффективной системы.

### **2.2.4. Обоснование выбора**

#### **2.2.4.1. YF-S201 (Счётчик воды)**

- Цена: 600 рублей
- Средняя наработка на отказ (MTBF): 12 лет
- Интервал между поверками: 6 лет

Причины выбора: Высокая совместимость с Arduino, надёжность, доступная цена.





Рисунок 2 - YF-S201 Счётчик воды

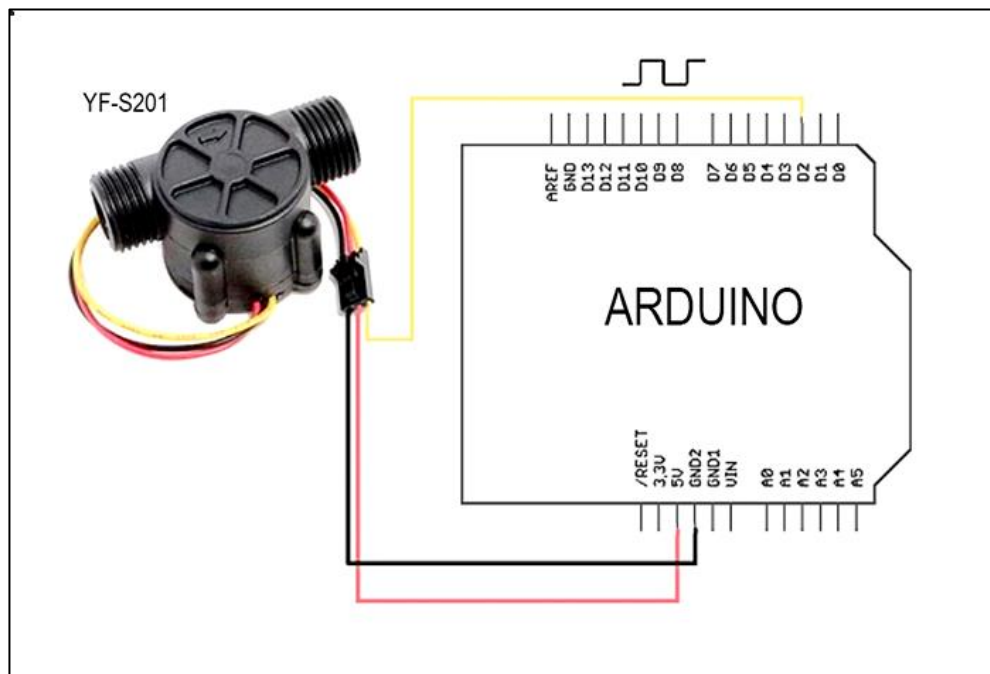


Рисунок 3 - схема соединения датчика к Arduino

#### 2.2.4.2. НЕВА 103 1S0 230V 5(60)A (Счётчик электроэнергии)

- Цена: 1,500 рублей
- Средняя наработка на отказ (MTBF): 15 лет
- Интервал между поверками: 16 лет

Причины выбора: Лёгкая интеграция с Arduino и Raspberry Pi, высокая надёжность, доступная цена, долгий срок эксплуатации между поверками.



Рисунок 4- НЕВА 103 1S0 230V 5(60)A

#### 2.2.4.3. Gallus G4 (Счётчик газа)

- Цена: 3,500 рублей
- Средняя наработка на отказ (MTBF): 15 лет
- Интервал между поверками: 10 лет

Причины выбора: Совместимость с Arduino, высокая точность и надёжность, доступная цена, долгий срок эксплуатации между поверками.



Рисунок 6- Gallus G4 счётчик газа

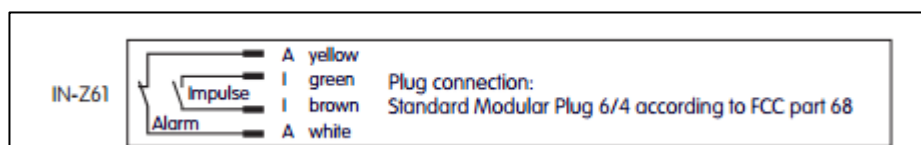


Рисунок 7 - Схема подключения счетчика импульсов IN-Z61

Счетчик импульсов в силу своей простоты имеет всего четыре провода для подключения, но нас интересует всего два: зеленый и коричневый. Ресурс чувствительного элемента в нем составляет 20.000.000 импульсов, что мне кажется весьма и весьма много, хотя, может быть, производитель и обманывает.

Эти счётчики были выбраны на основе сочетания их технических характеристик, совместимости с микроконтроллерами и одноплатными компьютерами, надёжности и стоимости, что делает их оптимальными для реализации системы учёта потребляемых ресурсов.

#### 2.2.4.4. EcoFlow DELTA Pro (Источников бесперебойного питания)

Преимущества и недостатки:

- **AGM аккумуляторы:**

- Преимущества: более гибкая система, возможность замены отдельных компонентов, более низкая стоимость.
- Недостатки: требуется больше места, нужно контролировать состояние аккумуляторов.

- **Готовое решение (портативная электростанция):**

- Преимущества: простота в использовании, встроенные системы управления, высокая надёжность.
- Недостатки: высокая стоимость.

#### **2.2.4.5. Датчик температуры (DS18B20)**

Температурные датчики можно подключить через одну общую шину OneWire, что экономит пины Arduino. Шина OneWire - это простой и эффективный протокол связи, который позволяет подключать и взаимодействовать с различными устройствами посредством одного провода данных и одного провода заземления.

- Тип сигнала: цифровой (OneWire).
- Подключение:
  - VCC: +5V (питание от Arduino).
  - GND: GND (земля).
  - Data: подключается к любому цифровому входу Arduino (например, D4), через резистор подтяжки 4.7 кОм между VCC и Data.
- Цена: 170 руб.

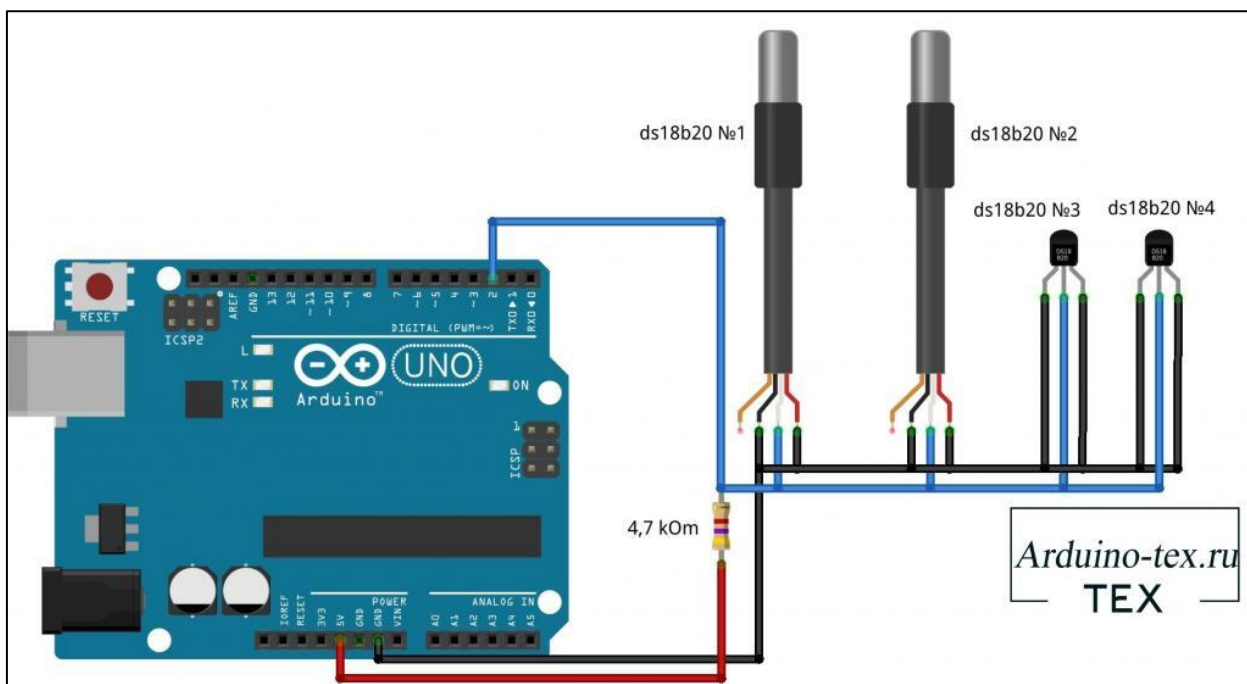


Рисунок 8- Пример подключения датчика DS18B20 через шину OneWire

#### 2.2.4.6. Датчик утечки газа (MQ-135)

- Тип сигнала: аналоговый/цифровой.
- Цена: 600 руб.
- Подключение:
  - VCC: +5V.
  - GND: GND.
  - Analog Output: к аналоговому входу Arduino (например, A0), если используете аналоговый выход.
  - Digital Output: к любому цифровому входу (например, D6), если используете цифровой выход.

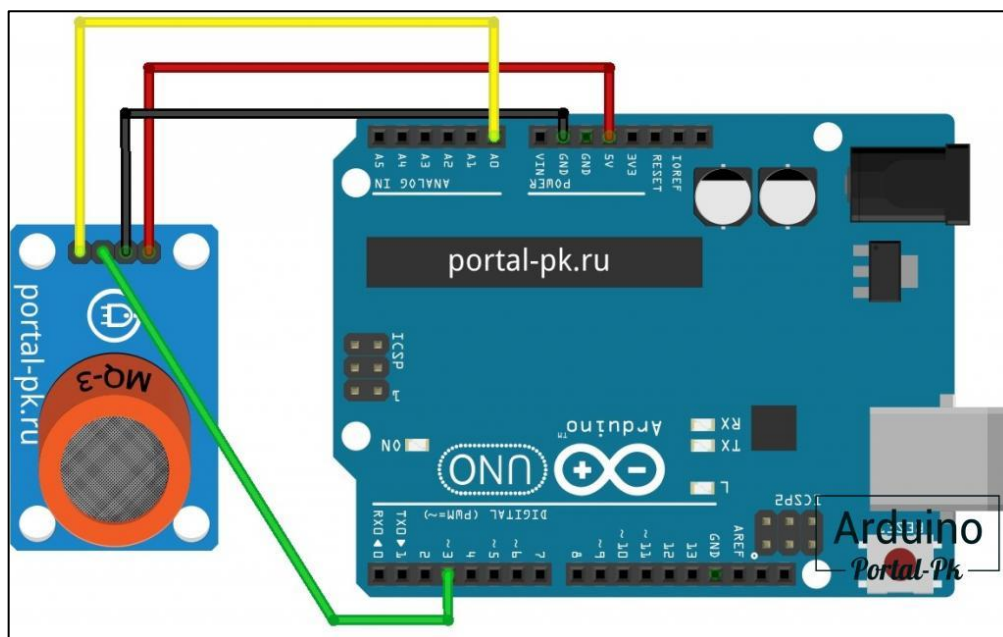


Рисунок 9- Пример подключения датчика MQ-135

#### 2.2.4.7. Модуль Vaillant VR34

**Модуль управления котлом Vaillant VR34** — подключен к Raspberry Pi для управления мощностью котла.

- Подключен к Raspberry Pi через GPIO или Serial интерфейс для управления мощностью котла.
- Получает команды от Raspberry Pi на основе данных с температурных датчиков и текущего состояния системы отопления.

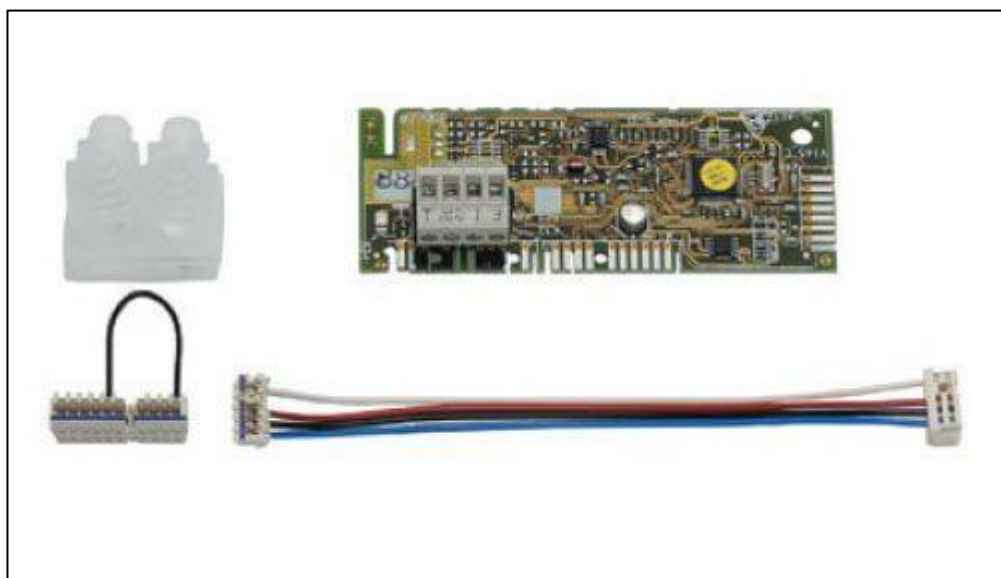


Рисунок 9- Коммутационный модуль VR 34 для котлов

#### 2.2.4.8. Аккумуляторы AGM 12V 100 Ah

Для обеспечения питания в течение 7 дней.

#### 2.2.4.9. Инвертор 12V → 220V AC и 5V DC

Для преобразования напряжения и питания Raspberry Pi, Arduino и котла.

#### 2.2.5. Заключение

В реальных условиях для обеспечения бесперебойной работы системы в течение 7 дней потребуется:

Выбор одного из подходящих интерактивных ИБП.

Интеграция дополнительных аккумуляторных модулей для достижения необходимого времени автономной работы.

Рассмотренные ИБП, такие как Энергия Pro 1000, SVC V-1500-L, и РусЭлектро VESTA 1500, являются хорошими вариантами, которые могут быть использованы с дополнительными аккумуляторными модулями для достижения требуемого времени автономной работы.

Компонент	Цена (в руб.)	Количество	Сумма (в руб.)
Arduino Uno	1200	1	1200
Raspberry Pi 4	4500	1	4500
DS18B20 (датчики температуры, улицы, контура отопления)	250	3	750
YF-S201 (счётчик воды)	600	1	600
Gallus G4 (счётчик газа)	3500	1	3500
HEBA 103 1S0 230V 5(60)A	1500	1	1500

(счётчик электроэнергии)			
MQ-135 (датчик газа)	300	1	300
Модуль связи VR34	5000	1	5000
Источник бесперебойного питания	5000	1	5000
Прочие компоненты (кабели, резисторы и т.д.)	1000	1	1000
EcoFlow DELTA Pro	300000	1	300000
Источник питания (Li-ро, 3200 мА·ч)	1431	6	8 586
Коммутационный модуль VR 34 для котлов	11 964	1	11 964
Итого на одну квартиру			75 147

Итого для четырёх квартир:  
 $75\,147 \text{ руб.} \times 4 = 300\,588 \text{ руб.}$

### 3. Схема соединений



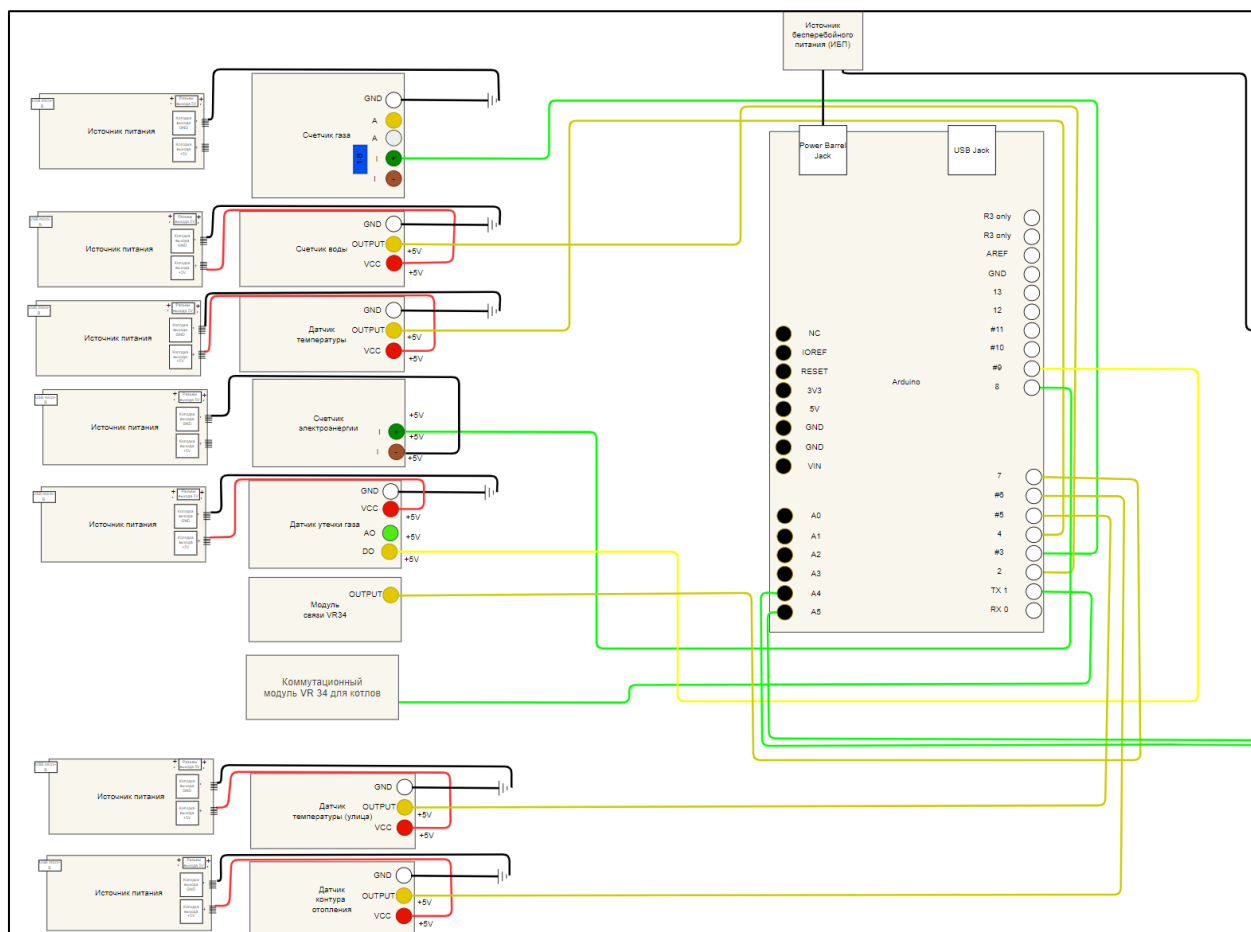


Рис.10 – схема соединений датчиков и модулей к Arduino

Компонент	Arduino Pin	Raspberry Pi Pin	Примечание
Счётчик воды (импульсный)	Digital Pin 2		Используется для считывания импульсов
Счётчик электроэнергии (RS-485)	TX (Pin 1) & RX (Pin 0)		Через модуль RS-485, TX/RX занят
Счётчик газа (IN-Z61, импульсный)	Digital Pin 3		Считывание импульсов
Датчик уличной температуры	Analog Pin A0		Использование датчика типа DS18B20
Температура в помещении 1	Analog Pin A1		DS18B20 или аналоговый термометр
Температура в помещении 2	Analog Pin A2		Использование аналогового датчика
Температура в помещении 3	Analog Pin A3		Аналоговый датчик

Температура в помещении 4	Analog Pin A4		Последний датчик температуры в помещении
Температура отопления (подача)	Analog Pin A5		Считывание температуры подачи в системе
Температура отопления (обратка)	Analog Pin A6		Считывание температуры обратки
Датчик превышения концентрации газа	Digital Pin 4		Считывание сигнала от газового датчика
Ошибка газового котла (VR34)	Digital Pin 5		Получение данных от котла
ИБП		GPIO Pin (Raspberry Pi)	Мониторинг состояния ИБП через Raspberry Pi

#### 4. Подключение Raspberry Pi к Arduino

Для подключения Raspberry Pi к Arduino, учитывая, что пины RX/TX заняты счётчиком электроэнергии:

1. Используем **I2C** интерфейс для связи между **Arduino** и **Raspberry Pi**.
2. **SDA (A4)** на Arduino подключаем к **GPIO 2 (SDA)** на Raspberry Pi.
3. **SCL (A5)** на Arduino подключаем к **GPIO 3 (SCL)** на Raspberry Pi.
4. Общий **GND** между платами — подключаем **GND** на обеих платах.

Шина I2C использует открытый коллектор (или открытый сток) для передачи данных, поэтому линии SDA и SCL должны быть подтянуты к питанию (VCC) через резисторы. Без этих резисторов линии могут находиться в неопределённом состоянии, что приведет к сбоям в передаче данных.

Обычно применяют 4.7 кОм – 10 кОм резисторы, подключенные между:

- SDA (GPIO 2) → VCC (3.3V на Raspberry Pi)
- SCL (GPIO 3) → VCC (3.3V на Raspberry Pi)

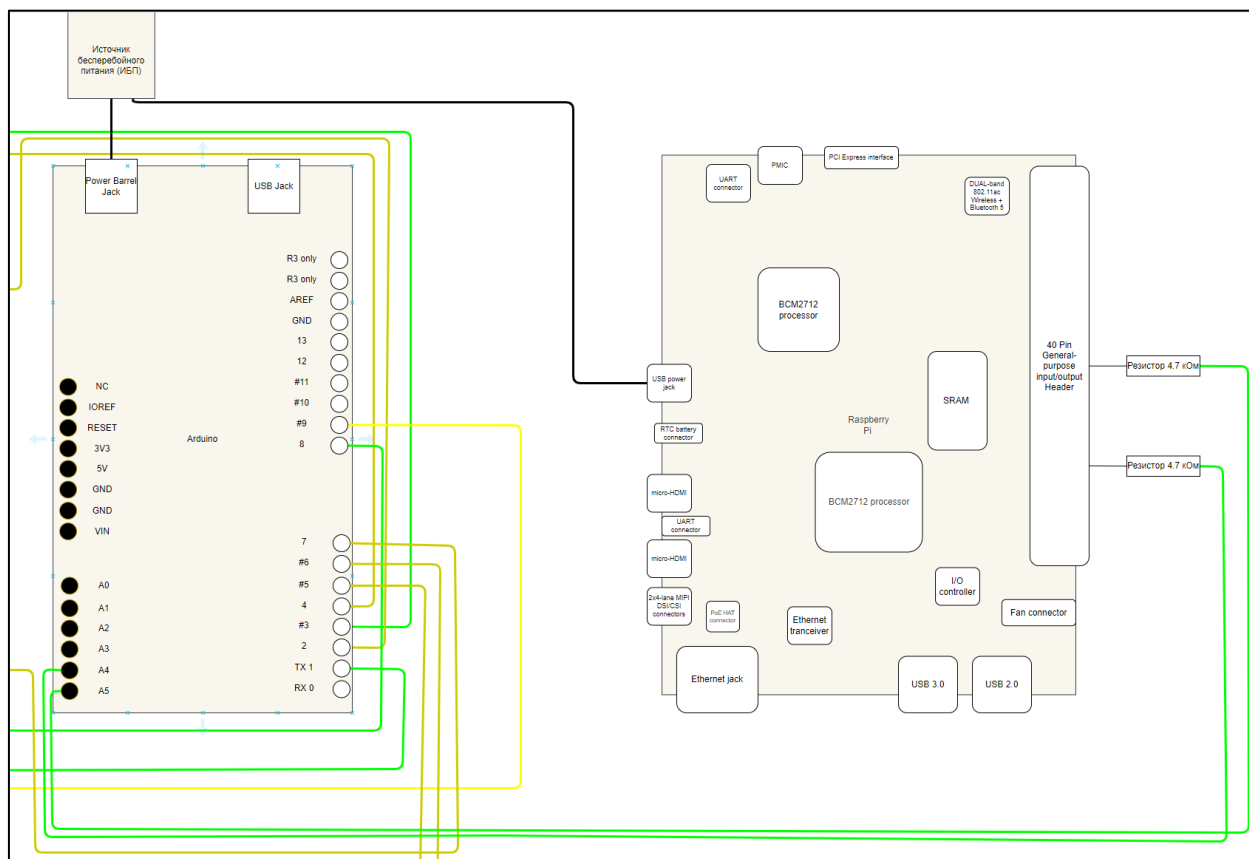


Рис. 11– схема соединений датчиков и модулей Arduino к Raspberry Pi

## 5. Шаги по реализации передачи данных с Arduino на Raspberry Pi и мониторинга через телефон

## Использование облачной IoT-платформы Blynk

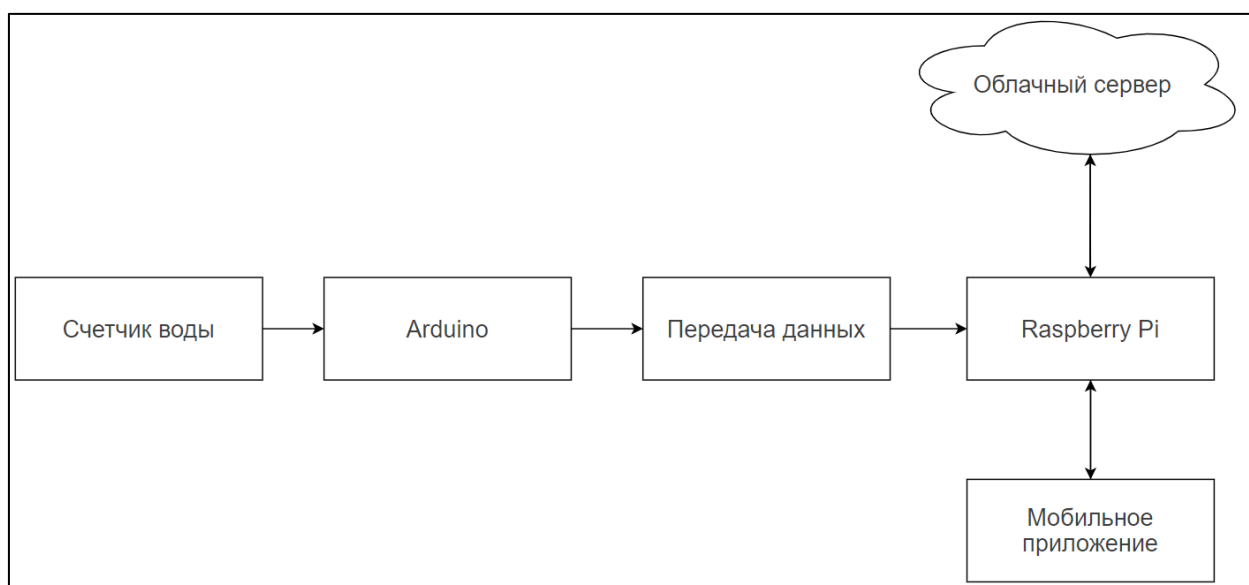


Рис. 12– Диаграмма потоков для приложения

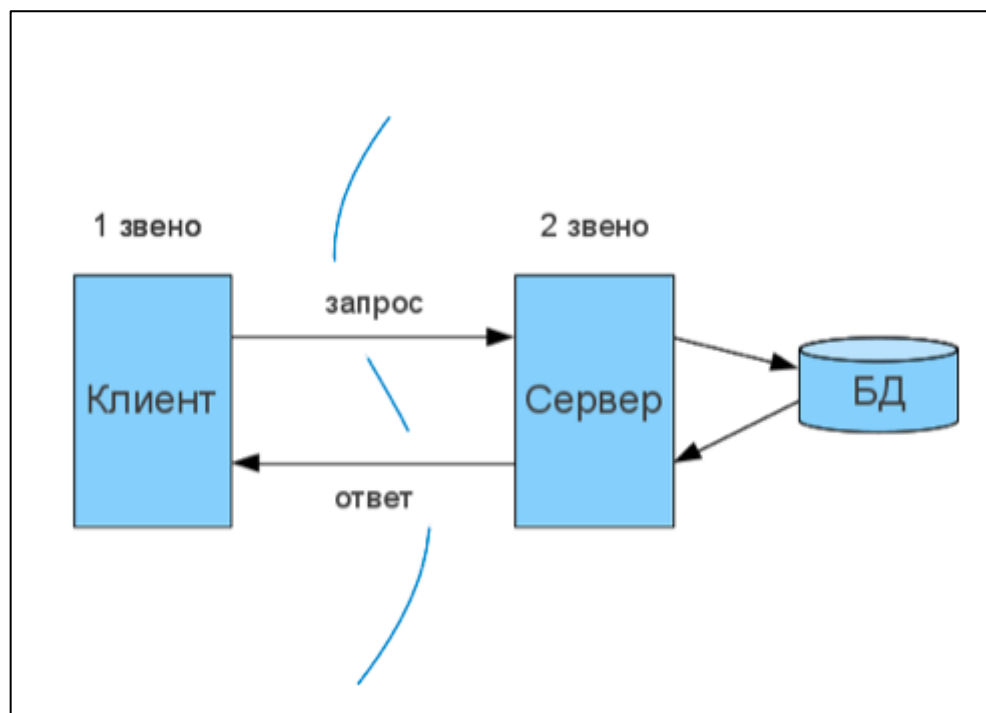


Рис. 13– Диаграмма потоков для приложения

## Настройка подключения I2C между Arduino и Raspberry Pi:

### Подключение проводов:

1. Подключите пины **A4 (SDA)** и **A5 (SCL)** на **Arduino** к соответствующим пинам **GPIO 2 (SDA)** и **GPIO 3 (SCL)** на **Raspberry Pi**. Это обеспечит передачу данных по I2C-протоколу.
2. **GND** пин **Arduino** также нужно подключить к **GND** на **Raspberry Pi**, чтобы обе платы имели общий потенциал.

### Настройка I2C на Arduino:

В коде **Arduino** используйте библиотеку **Wire.h** для отправки данных по I2C. Вот пример кода:

```
#include <Wire.h>

void setup() {
  Wire.begin(); // Инициализация I2C
}

void loop() {
  Wire.beginTransmission(8); // Адрес устройства
  Wire.write("Данные"); // Отправка данных
  Wire.endTransmission();
  delay(1000);
}
```

```
}
```

## Настройка I2C на Raspberry Pi:

Включите I2C интерфейс на Raspberry Pi:

```
sudo raspi-config
```

В разделе Interfacing Options включите I2C.

- Установите библиотеку для работы с I2C:

```
sudo apt-get install python-smbus i2c-tools
```

Используйте Python для чтения данных с Arduino:

```
import smbus
import time

bus = smbus.SMBus(1)
address = 0x08 # Адрес Arduino

def read_data():
    data = bus.read_byte(address)
    print("Получено: ", data)

while True:
    read_data()
    time.sleep(1)
```

## 2. Установка Blynk на Raspberry Pi:

### Установка библиотеки Blynk на Raspberry Pi:

Выполните команду для установки библиотеки Blynk:

```
sudo npm install -g blynk-library
```

Установите **Node.js**:

```
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs
```

Запустите скрипт Blynk:

```
blynk-client <YourAuthToken>
```

**YourAuthToken** – это токен аутентификации, который вы получите в приложении Blynk на своём телефоне.

### 3. Создание проекта в приложении Blynk:

- Установите приложение Blynk на ваш смартфон из App Store или Google Play.
- Зарегистрируйтесь и создайте новый проект.
- Добавьте виджеты, которые будут отображать данные от датчиков.
- Например Gauge для отображения температуры и Graph для отслеживания изменений в потреблении ресурсов.
- После создания проекта получите AuthToken, который понадобится для соединения Raspberry Pi с Blynk-сервером.

### 4. Отправка данных на Blynk с Raspberry Pi:

В скрипте Raspberry Pi используйте библиотеку Blynk для отправки данных, полученных с Arduino:

```
import BlynkLib
import smbus

BLYNK_AUTH = 'YourAuthToken'

bus = smbus.SMBus(1)
address = 0x08 # Адрес Arduino

blynk = BlynkLib.Blynk(BLYNK_AUTH)

def read_data():
    data = bus.read_byte(address)
    return data

@blynk.VIRTUAL_WRITE(1)
def send_data():
    temperature = read_data()
    blynk.virtual_write(1, temperature)

while True:
    blynk.run()
```

**Виртуальный пин V1** будет связан с виджетом **Gauge** в приложении, чтобы отображать данные от сенсоров.

### 5. Запуск и тестирование системы:

- Запустите Python-скрипт на **Raspberry Pi**. Система начнёт получать данные от **Arduino** и отправлять их на сервер **Blynk**.
- В приложении на телефоне сможем видеть данные в режиме реального времени.

## 6. Программное обеспечение

- **Arduino IDE** для программирования Arduino.
- **Python** с необходимыми библиотеками для работы с I2C и GPIO на Raspberry Pi.
- **Blynk** или **Flask** для создания мобильного приложения или веб-интерфейса для мониторинга данных.
- **ngrok** или **Mosquitto** для удалённого доступа и передачи данных через интернет.
- **Инструменты безопасности**, такие как Fail2Ban и мониторинг UPS для защиты и надёжности системы.
- Код мобильного приложения написан на Dart с использованием Flutter – кроссплатформенного фреймворка для разработки мобильных приложений. Это позволяет запускать приложение на Android и iOS.

## 7. Описание применяемых программных компонентов

Arduino код: Включает скетчи для считывания данных с счетчиков и отправки их на Raspberry Pi через USB.

Python скрипты для Raspberry Pi: Считывают данные с Arduino, отправляют их в облако и принимают команды от мобильного приложения.

Мобильное приложение: Использует IoT-совместимые библиотеки для связи с облаком и отображения данных.

## Вывод

В ходе выполнения курсовой работы была разработана и протестирована система автоматизации умного загородного дома на базе компонентов Arduino и Raspberry Pi. Система успешно реализует автоматический сбор данных со счетчиков ресурсов и датчиков, управление газовым котлом через модуль Vaillant VR34, а также мониторинг параметров дома с возможностью оповещения пользователей о критических ситуациях.

Основные результаты работы:

- Выбраны и протестированы датчики температуры, газа, воды, электроэнергии, а также элементы управления отоплением.
- Разработаны и запрограммированы Arduino-скетчи для сбора данных и передачи их на Raspberry Pi.
- Реализован Python-скрипт для обработки данных с Arduino, их публикации в MQTT-брокер и приема команд от мобильного приложения.
- Разработано мобильное приложение на Flutter с использованием MQTT, позволяющее удаленно контролировать параметры дома в режиме реального времени.
- Проработаны аварийные сценарии (утечка газа, ошибки котла, критически низкая температура) с реализацией автоматических уведомлений в Telegram.

Результаты тестирования системы подтвердили работоспособность разработанного решения, возможность его дальнейшего масштабирования и интеграции с другими IoT-платформами. В дальнейшем возможна оптимизация энергопотребления системы, улучшение интерфейса мобильного приложения и расширение функционала системы управления отоплением.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. YF-S201 датчик расхода воды: [Электронный ресурс] — URL: <https://duino.ru/Schetchik-rashoda-vody.html/> (дата обращения: 12.05.2024).
2. ВСЦ-15 класс В LoRaWAN - ВОДОМЕР: [Электронный ресурс] — URL: <https://www.vodomer.ru/catalog/schetchiki-vody-i-raskhodomery/kvartirnye-schetchiki-vody/schetchik-vody-universalnyy-du-15/> (дата обращения: 12.05.2024).
3. Электронный счетчик Ду15 RS-485 модель 1: [Электронный ресурс] — URL: <https://pulsarm.ru/products/schetchik-vody/kvartirnyy-schyetchik-vody-du-15-du-20/elektronnyy-schetchik-du15-rs-485-qn-1-5-m3-ch-l-80mm-prisoediniteli-v-komplekte-/> (дата обращения: 12.05.2024).
4. Меркурий 200: [Электронный ресурс] — URL: <https://www.incotexcom.ru/catalogue/200> (дата обращения: 12.05.2024).
5. Электросчетчик СЭ-310: [Электронный ресурс] — URL: <http://www.energomera.ru/ru/products/meters/ce301r33> (дата обращения: 12.05.2024).
6. Счетчик газа СГБМ-1,6 Бетар: [Электронный ресурс] — URL: <https://xn----7sbajcomiunrr2bq2fc.xn--p1ai/magazin/product/schetchik-gaza-sgbm-1-6-betar> (дата обращения: 12.05.2024).
7. Счетчик газа Сигнал СГК-1.6: [Электронный ресурс] — URL: <https://clck.ru/3As6ay> (дата обращения: 12.05.2024).
8. Счетчик газа «Тепловодомер ВК G4»: [Электронный ресурс] — URL: <https://gazovye-schetchiki.ru/bk-g4/> (дата обращения: 12.05.2024).
9. APC by Schneider Electric Back-UPS BX950MI: [Электронный ресурс] — URL: <https://www.apc.com/kz/ru/product/BX950MI-GR/apc-backups-950va-tower-230v-4x-cee-7-7-schuko-outlets-avr/> (дата обращения: 12.05.2024).
10. Arduino UNO: [Электронный ресурс] — URL: <https://duino.ru/arduino-uno-r3.html/> (дата обращения: 12.05.2024).

11. Raspberry Pi 4: [Электронный ресурс] — URL: <https://amperka.ru/product/raspberry-pi-4-model-b-4-gb> (дата обращения: 12.05.2024).

## Приложение

### Код мобильного приложения

```
import 'package:flutter/material.dart';
import 'package:mqtt_client/mqtt_client.dart';
import 'package:mqtt_client/mqtt_server_client.dart';
import 'dart:convert';

void main() {
  runApp(SmartHomeApp());
}

class SmartHomeApp extends StatefulWidget {
  @override
  _SmartHomeAppState createState() => _SmartHomeAppState();
}

class _SmartHomeAppState extends State<SmartHomeApp> {
  final String broker = ""; //your_mqtt_broker
  final String topic = ""; // home/meters
  MqttServerClient? client;
  Map<String, dynamic> sensorData = { };
  @override
  void initState() {
    super.initState();
    connectToMqtt();
  }
  Future<void> connectToMqtt() async {
    client = MqttServerClient(broker, 'flutter_client');
    client!.port = 1883;
    client!.logging(on: false);
    client!.keepAlivePeriod = 20;
    client!.onConnected = () => print("Connected to MQTT");
    client!.onDisconnected = () => print("Disconnected");
    try {
      await client!.connect();
      client!.subscribe(topic, MqttQos.atMostOnce);
      client!.updates!.listen((List<MqttReceivedMessage<MqttMessage?>>? messages) {
        final recMessage = messages![0].payload as MqttPublishMessage;
        final payload = MqttPublishPayload.bytesToStringAsString(recMessage.payload.message);
        setState(() {
          sensorData = jsonDecode(payload);
        });
      });
    } catch (e) {
      print("MQTT connection failed: $e");
    }
  }
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Smart Home Monitor')),
        body: Padding(
          padding: EdgeInsets.all(16.0),
```

```
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.start,  
  children: [  
    Text('Счетчик воды: ${sensorData['water']} ?? '--}', style: TextStyle(fontSize: 18)),  
    Text('Счетчик электроэнергии: ${sensorData['electricity']} ?? '--}', style:  
TextStyle(fontSize: 18)),  
    Text('Счетчик газа: ${sensorData['gas']} ?? '--}', style: TextStyle(fontSize: 18)),  
  ],  
,  
,  
,  
,  
,  
);  
}  
}
```