# 1 Анализ предметной области

**1.1. Особенности индикации состояния различных тел и сред.**

Термин индикация происходит от Немецкий термин indikator, который в свою очередь происходит от латинского слова indicare - «определять», внедрен в научный тезаурус в начале XIX в. для обозначения слежения за изменениями каких-либо параметров контролируемого тела, объекта или явления. В настоящее время под индикацией понимают комплекс методов и приемов, предназначенных для наблюдения, фиксации, контроля, характеристики и оценки состояния и стадий развития различных процессов, объектов и систем исследования для установления и контроля зависимостей от изменения условий во времени, статистики количественного и качественного порядка, сопоставления с нормой. Данный комплекс методов получил широкое распространение в сфере эмпирических исследований по химии, физике, биологии, медицине, электронике и т. д.

Одно из первых доказательств эффективности использования индикации в экспериментах обосновал французский иследователь Роберт Бойль. Для идентификации среды раствора ученый использовал простые бумажные полоски, пропитанные настоем лакмусового лишайника, изменявшие свой цвет, при погружении в жидкость, в зависимости от наличия ней определенных веществ.

Первым техническим индикатором, доказавшим эффективность индикации при исследовании окружающего мира и состояния человека, стало создание термометра, использующего ртуть и шкалу для измерения температур. Одним из главных признаков заболевания у живого организма является повышенная температура тела, однако далеко не всегда на ощупь можно было определить, насколько она отклонена от нормы. Благодаря термометру с достаточной точностью определяли, повышена ли у человека температура тела и констатировали наличие заболевания.

Таким образом, методы индикации прекрасно себя зарекомендовали в медицине. Использование термометра оказалось результативным и для химии: повышение температуры указывало на проходящие в ходе эксперимента процессы, что позволяло точнее контролировать ход исследования.

Методы индикации получали широчайшее распространение во всех практически значимых областях знаний. В биологии они показали свою эффективность при обнаружении вирусов различного вида на основе определенных признаков. В экологии - для определения наличия загрязнений окружающей среды: на основании полученных с помощью индикации данных можно обнаружить в различных мхах, встречающихся в городских парках, маркеры тяжелых металлов и иные следовые элементы даже в ультранизких концентрациях.

С развитием технического прогресса и переходом цивилизации на новую ступень развития, жизнь и здоровье человека стали подвергаться воздействию искусственно созданной им среды. Наличие электромагнитных полей в местах скопления людей настоятельно требуют их локализации и смягчения последствий воздействия. Нахождение в их зоне в течение длительного времени приводит к усталости, тошноте, головной боли. При значительных превышениях нормативов возможны повреждение сердца, мозга, центральной нервной системы. Излучение может влиять и на психику человека, провоцируя раздражительность, апатию, отсутствие эмоционального контроля и появление вспышек ярости. Возможно развитие трудно поддающихся лечению и диагностированию психоэмоциональных (депрессия, хроническая усталость), и онкологических заболевания. Медицинские исследования показали прямую средней силы корреляцию заболеваемости злокачественными заболеваниями головного мозга с максимальной нагрузкой от электромагнитного излучения, даже при использовании такого маломощного источника, как мобильные радиотелефоны. Очевидной становится необходимость в существенном изучении действии электромагнитного излучения на живые организмы. Для этого также эффективно используют методы индикации, например, замеры параметров полей, которые при повышенных отклонениях от нормы является маркером возможного возникновения проблем.

С дальнейшим развитием электроники индикаторы и методы индикации получили широкое распространение и стали неотъемлемой частью нашей повседневной жизни. Электронный индикатор помогает человеку быстро и наглядно оценить необходимые параметры, особенно те, которые человек непосредственно не может определить с помощью органов чувств. Если требуется высокая точность оценки, устанавливаются многоразрядные [цифровые индикаторы](https://ru.wikipedia.org/wiki/Цифровой_индикатор); в случаях, когда точность не требуется и необходимо увидеть лишь наличие или отсутствие сигнала, применяют единичные индикаторы.

Причисление тех или иных устройств к индикаторам определяется их применением. Так, например, обычная лампочка накаливания, созданная для освещения, при использовании в системах оповещения или пультах управления и контроля, может считаться индикатором. В то же время, [электронное табло](https://ru.wikipedia.org/wiki/Светодиодный_графический_экран), изготовленное из матричных светодиодных индикаторов и используемое для рекламы, уже индикатором не считается. Именно для этого необходимо классифицировать электронные индикаторы.

Существуют следующие группы индикаторов:

1. По принципу физической работы:

* накаливания;
* газоразрядные;
* вакуумные;
* вакуумно-люминисцентные;
* полупроводниковые;
* жидкокристаллические;
* электролюминафорные.

2. По виду отображаемой информации:

* точечные (единичные);
* цифровые;
* буквенно-цифровые;
* графические;
* матричные;
* мнемонические (изменение некоторого образа).

3. По виду излучаемой энергии:

* сегментные;
* матричные;
* экранные.

4. По числу знакомест:

* одноразрядные;
* многоразрядные;
* составные.

5. По виду излучаемой энергии:

* активные;
* пассивные.

6. По конструкции корпуса:

* цилиндрические;
* торцевые.

7. По материалу корпуса:

* стеклянные;
* стеклокерамические;
* металостеклянные;
* пластмассовые;
* пластмассо-керамические;
* металлические;
* металло-керамические.

8. По режиму управления:

* статический;
* мультиплексный – коды подаются на все индикаторы одновременно; конкретный индикатор выбирается управляющим сигналом.

9. По способу управления:

* встроенные;
* внешние.

10. По виду питающего напряжения:

* постоянные;
* переменные;
* импульсные.

11. По значению питающего напряжения:

* низковольтные (<5В);
* средневольтные (5..30В);
* высоковольтные (>30В).

Так или иначе, все они применяются для одной задачи – оповещение наблюдателя об изменении состояния системы и отклонений ее параметров от нормы.

## 1.2 Обзор сведений о существующих модулях индикации для системы сбора и обработки данных.

Обычно, в качестве самостоятельных технических устройств, индикаторы не используются. Основная область их применения состоит в том, чтобы при подключении к управляющей системе и получении от нее данных, реагировать на их отклонение от нормы. В качестве таких систем зачастую выступают системы сбора данных.

Система сбора данных  — комплекс средств, предназначенный для работы совместно с [персональным компьютером](https://ru.wikipedia.org/wiki/Персональный_компьютер), либо специализированной ЭВМ и осуществляющий автоматизированный сбор информации о значениях физических параметров в заданных точках объекта исследования с аналоговых и/или цифровых источников сигнала, а также первичную обработку, накопление и передачу данных.

Совместно с  [ПЭВМ](https://ru.wikipedia.org/wiki/Персональный_компьютер), оснащенной специализированным программным обеспечением, система сбора данных образует информационно-измерительную систему (ИИС). ИИС — это многоканальный измерительный прибор с широкими возможностями обработки и анализа данных. На основе ИИС могут быть построены различные [автоматизированные системы управления](https://ru.wikipedia.org/wiki/Автоматизированная_система_управления) (АСУ), среди которых: информационно-логические комплексы (их называют АСУ технологическими процессами — [АСУ ТП](https://ru.wikipedia.org/wiki/АСУ_ТП)), информационно-вычислительные комплексы (автоматизированная система научных исследований — [АСНИ](https://ru.wikipedia.org/wiki/АСНИ)), информационно-диагностические комплексы и информационно-контролирующие системы.

В систему сбора данных также включают и управляющие средства: линии цифрового ввода-вывода, [цифро-аналоговые преобразователи](http://www.lcard.ru/lexicon/dac).

Таким образом, система сбора данных охватывает сразу несколько [уровней программных и аппаратных средств](http://www.lcard.ru/lexicon/soft_hard_level).

Фундаментальный принцип построения систем сбора данных – это модульность, обеспечивающая гибкость при построении систем. Это могут быть как отдельные модули, так и модули, объединенные в блок ([крейт](http://www.lcard.ru/products/ltr)).

Системы сбора данных применяют как для автоматизации производства, так и для автоматизации лабораторных измерений.

Термин автоматизированная система сбора данных (АССД) также является устоявшимся. — Подразумеваются не только автономные системы сбора данных с удаленными интерфейсом, но и системы с ближним интерфейсом, собирающие данные в автоматическом режиме, с малой долей участия оператора.

В итоге можно сделать вывод, что системы сбора данных применяются во многих сферах деятельности человека. Так же эти системы активно используются в исследовательской сфере и при изучении определенных явлений. И чтобы следить за проходящим процессом, подобным системам необходимы модули, которые отвечают за индикацию состояния.

Модули индикации – электронные блоки, предназначенные для подключения к многофункциональным измерительным преобразователям и отображения принимаемых данных. Эти устройства могут быть оснащены светодиодными индикаторами или дисплеем, рассчитаны на монтаж в приборный щит или шкаф.

В данный момент существует большое множество модулей индикации. Зачастую они спроектированы специально для определенных систем и выполняют функцию индикации для параметров, обозначенных этими системами. Однако все они работают примерно по одному принципу и показывают информацию определенными способами.

Одними из самых распространенных индикаторов являются светодиоды. Светодиодные индикаторы приведены на рисунке 1.

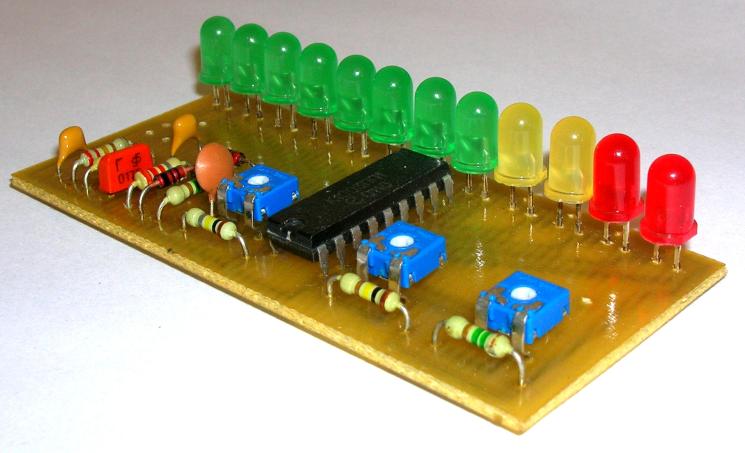


Рисунок 1 – Светодиодные индикаторы

Принцип работы таких индикаторов довольно прост – при подачи сигнала на эту систему, загорается один из светодиодов. На основе того, какой светодиод загорелся, можно судить о состоянии, в котором находится система в данный момент.

Также популярной системой индикации состояния является дисплей, на который выводятся данные. Дисплеи такого типа приведены на рисунке 2.



Рисунок 2 – Модуль индикации с текстовым дисплеем

Этот метод хорош, когда необходимо не только знать, что в системе что-то изменилось, но и когда нужно определить, что именно изменилось и как. Эта система наглядно показывает все необходимые параметры.

Еще более наглядным методом получения информации о состоянии системы являются модули индикации, показывающие данные графическим методом. Графический модуль индикации приведен на рисунке 3.

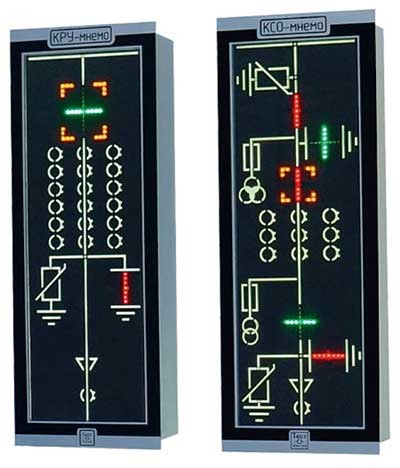


Рисунок 3 – Модуль индикации с графическим дисплеем

Благодаря такому модулю можно напрямую увидеть, в каком участке системы произошло событие и как оно влияет на всю систему в целом.

В некоторых случаях не удобно наблюдать за состоянием системы используя индикаторные модули, передающие информацию визуальным способом. В таких случаях применяют модули индикации, в которых информация об изменении параметров передается с помощью звука. Такой модуль приведен на рисунке 4.

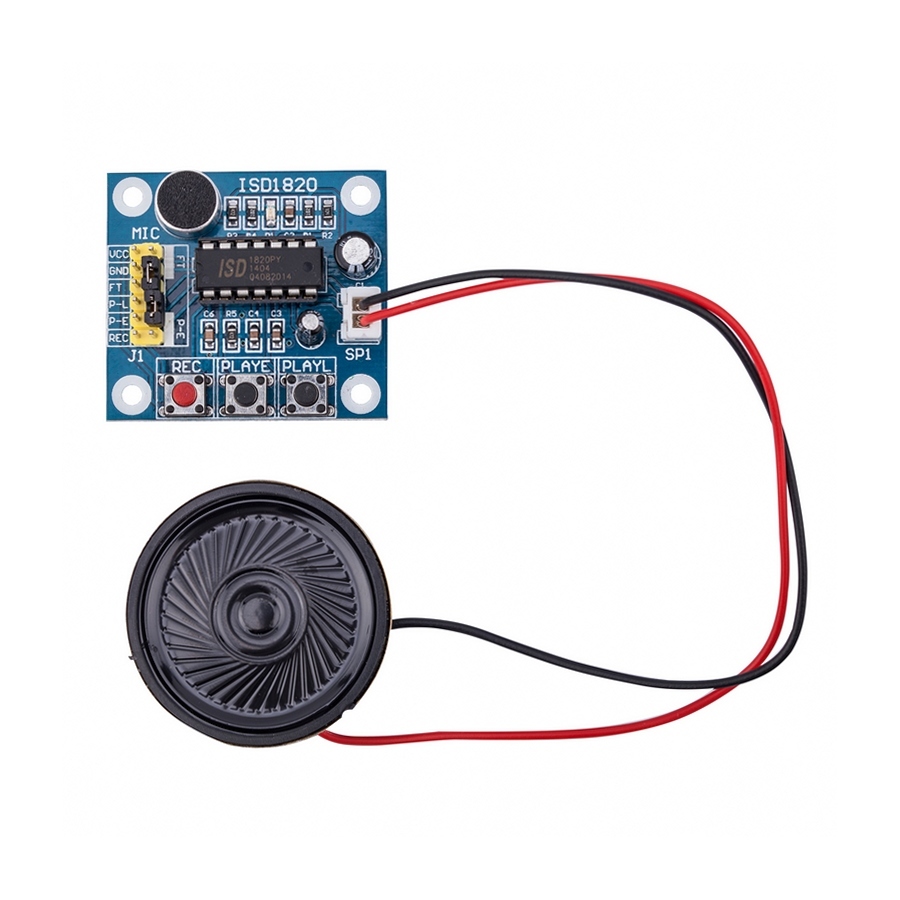


Рисунок 4 – Модуль индикации с динамиком

Модули такого типа используют динамик для создания звуков. Различные состояния системы могут передаваться различными звуками, что позволяет, не отвлекаясь от работы, реагировать на произошедшие изменения.

Развитием предыдущего модуля стал модуль, который оповещает пользователя не просто звуком, а фразой.

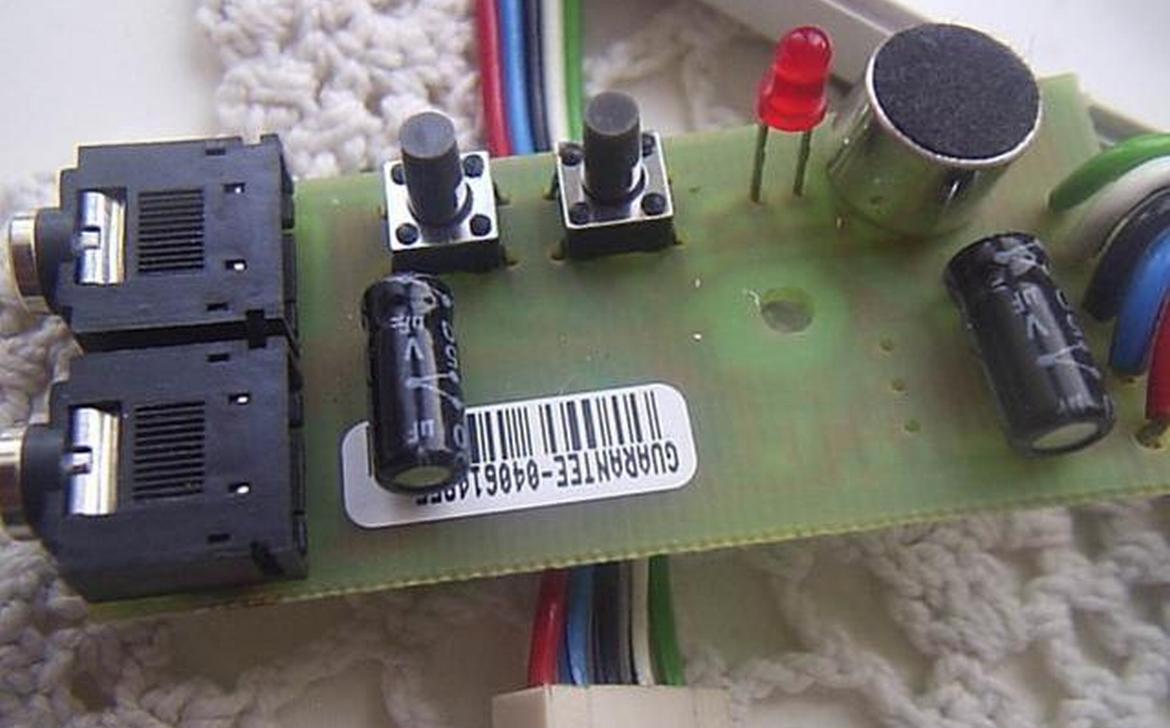


Рисунок 5 – Модуль речевого оповещения

Такие модули передают информацию о системе с помощью заранее записанных фраз. Благодаря такому методу, пользователь может получить более точную информацию о состоянии и предпринять более эффективные действия.

## 1.3 Постановка цели и задач работы

Исходя из описанного выше, можем сделать вывод, что в наше время имеется потребность в модулях индикации для систем сбора данных. На данный момент существует большое множество готовых решений, которые активно применяются в различных системах. Однако, зачастую модули индикации содержат в себе несколько вариантов оповещения пользователя, например, комбинируют сигнальный метод через светодиоды с текстовым, или сигнальный со звуковым. Но на данный момент не существует модуля, который совмещал бы в себе все варианты индикации и был бы универсален для системы любого вида.

Именно поэтому наша цель – разработать такой модуль, который будет отвечать заявленным выше критериям универсальности. Для этого необходимо спроектировать модуль, написать для него ПО, провести все необходимые тестирования и представить в виде электронного устройства, с дальнейшей реализацией в конкретной системе.

# Постановка задачи проектирования

* 1. **Системы автоматизированного проектирования**

Перед тем как преступить к созданию модулей индикации, необходимо создать схему будущих устройств и проверить их работоспособность. Для этого существуют системы автоматизированного проектирования (САПР).

САПР – программный пакет, предназначенный для создания чертежей, конструкторской и/или технологической документации и/или 3D моделей. Современные системы автоматизированного проектирования обычно используют совместно с системами автоматизации инженерных расчетов и анализа.

Однако, для решения поставленной задачи, не подойдет любая САПР. Необходима такая система, которая позволяет разработать схему устройства на базе микроконтроллера с возможностью загрузить в него тестовую прошивку. Таким образом, рассматриваться будут 3 системы автоматизированного проектирования:

* Proteus;
* SimulIDE;
* QUCS.
  + 1. **Proteus**

Proteus – это мощная система схемотехнического моделирования, сделанная на основе виртуальных моделей электронных элементов. Специфической чертой данного программного пакета является отличная возможность моделирования различной работы программируемых устройств: микропроцессоров, контроллеров, DSP и т.д.

Программный пакет Proteus позволяет собрать схему любого электронного устройства и симулировать его работу, выявляя ошибки, допущенные на стадии проектирования и трассировки. Программа состоит из двух модулей. **ISIS** – редактор электронных схем с последующей имитацией их работы. **ARES** – редактор печатных плат, оснащенный автотрассировщиком Electra, встроенным редактором библиотек и автоматической системой размещения компонентов на плате. Кроме этого ARES может создать трехмерную модель печатной платы.

Proteus включает в себя более 6000 электронных компонентов со всеми справочными данными, а также демонстрационные ознакомительные проекты. Программа имеет инструменты USBCONN и COMPIM, которые позволяют подключить виртуальное устройство к портам USB и COM компьютера. При подсоединении к этим портам любого внешнего прибора виртуальная схема будет работать с ним, как если бы она существовала в реальности. Proteus поддерживает следующие компиляторы: CodeVisionAVR и WinAVR (AVR), ICC (AVR, ARM7, Motorola), HiTECH (8051, PIC Microchip) и Keil (8051, ARM). Существует возможность экспорта моделей электронных компонентов из программы PSpice. Внешний вид программы Proteus приведен на рисунке 6.

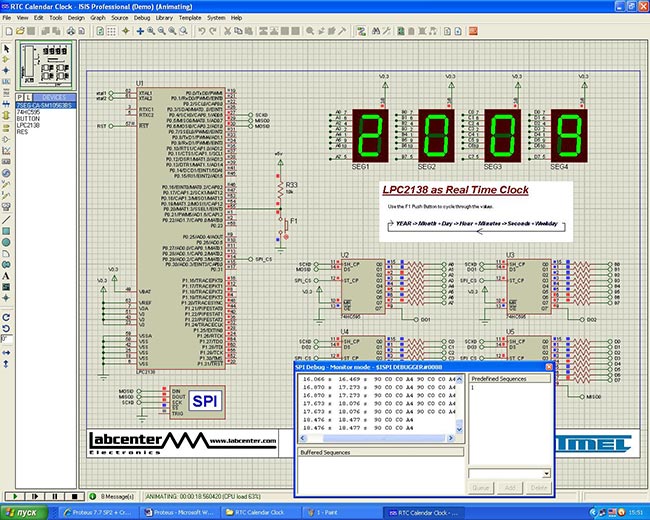


Рисунок 6 – Рабочее окно программы Proteus

Proteus является коммерческим продуктом. Есть бесплатная демонстрационная версия. Она обладает всеми функциями и возможностями платного пакета, но не позволяет сохранить или распечатать результат работы, создать свой собственный микроконтроллер.

Меню программы англоязычное. Полного русификатора для программы Proteus нет.

* + 1. **SimulIDE**

SimulIDE – симулятор электронных схем в реальном времени. Это простой инструмент, предназначенный для углубленного изучения. SimulIDE разработан, чтобы быть быстрым и простым в использовании, а так же предоставляет возможность моделирования с использованием микроконтроллеров.

SimulIDE предоставляет микроконтроллеры AVR, Arduino и PIC, к которым можно получить доступ, как и к другим компонентам. Благодаря gpsim и simavr возможно провести моделирование микроконтроллера с помощью загрузки прошивки. Также, при создании схемы с микроконтроллером, имеется возможность отслеживать работу с памятью для лучшего контроля процесса моделирования. Рабочее окно программы SimulIDE приведено на рисунке 7.

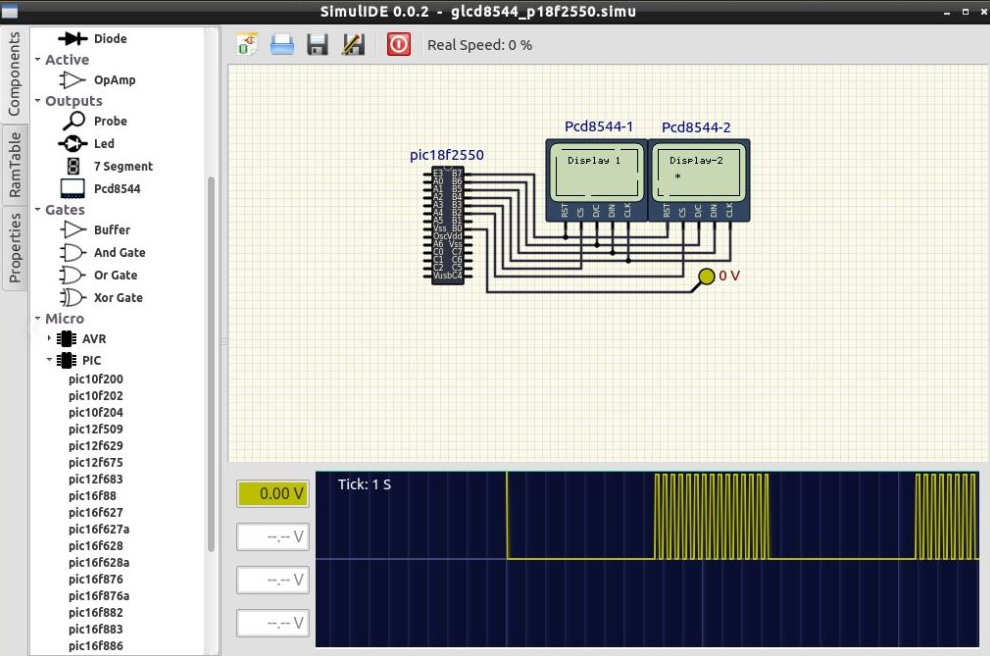


Рисунок 7 – Рабочее окно программы SimulIDE

Главными особенностями этой системы является бесплатный способ распространения, простой, не перегруженный интерфейс с поддержкой русского языка, общего быстродействия программы и наличия базовых компонентов, которые позволяют собрать большое количество схем для решения различных задач.

* + 1. **QUCS**

QUCS (Quite Universal Circuit Simulator) – это симулятор схем с удобным графическим интерфейсом, позволяющий конструировать и рассчитывать производительность электронных цепей и контуров различного уровня сложности. Внешний вид программы представлен на рисунке 8.

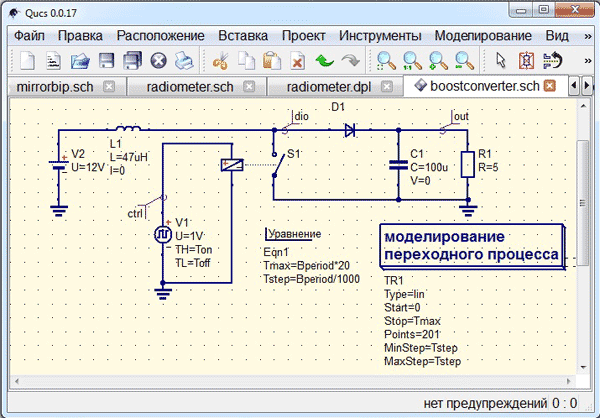


Рисунок 8 – Рабочее окно программы QUCS

Приложение Qucs основано на открытом исходном коде и, подобно аналогичным редакторам, обладает всеми необходимыми для модификации схем средствами. Для удобства имеется поддержка «горячих клавиш» и выпадающие меню. При работе со сложными схемами предлагается использование подсхем, позволяющих отделить часть основной схемы в виде блоков. Кроме того, программное обеспечение имеет собственный текстовый редактор, приложения для расчета фильтров и согласованных цепей, калькуляторы линий и синтеза аттенюаторов. Присутствует возможность оформления чертежа путем добавления рамки и штампа. К схемам можно подключать пользовательские уравнения, причем в формулах могут быть использованы все доступные функции данного софта или их сочетания.

Qucs имеет обширную базу современных компонентов, разбитых на группы: дискретные (резисторы, конденсаторы и т.д.), нелинейные (в основном транзисторы и диоды), цифровые (базовые цифровые устройства и логические вентили) и некоторые другие (источники, измерители, линии передачи данных). Особыми группами представлены виды моделирования, рисунки и диаграммы. Каждый элемент имеет собственное диалоговое окно свойств, которое может быть подвергнуто редактированию. Если в библиотеке нет необходимого компонента, то его можно добавить туда при наличии SPICE-модели.

После запуска симуляции Qucs многократно пересчитывает созданную электрическую цепь. Длительность процесса моделирования зависит от мощности компьютера и сложности цепи. Также имеется возможность выбирать вид полученных после симуляции данных, а именно табличную или декартовскую диаграммы, трехмерную или Смита, полярную или круговую. Диаграммы имеют свои собственные настраиваемые характеристики, которые зависят от применяемых видов моделирования. Для определения значений функций в определенные моменты времени в программе предусмотрен механизм маркеров.

* + 1. **Выбор системы автоматизированного проектирования**

После проведения анализа существующих САПР, была выбрана система SimulIDE, поскольку Proteus распространяется на платной основе, поэтому он не подходит как система для учебных целей. Также система QUCS была отклонена, поскольку она не имела необходимого функционала для реализации поставленной задачи.

**2.2 Встроенное ПО**

**2.2.1 Общая информация о языках программирования микроконтроллеров**

Языки программирования микроконтроллеров по своей структуре очень похожи на языки программирования для универсальных компьютеров. Микроконтроллер, аналогично компьютеру, получает на вход команды на машинном языке. Такой код слабо воспринимается человеком и вследствие этого плохо подходит для отладки программ. Сложившаяся ситуация послужила появлению специализированных языков, предназначенных для программирования микроконтроллеров.

**2.2.2 Классификация языков программирования микроконтроллеров**

Языки программирования микроконтроллеров могут отличаться по своему составу и применению, но главным отличием в данном случае будет уровень сложности использования программы специалистом. Наиболее распространена следующая классификация языков программирования микроконтроллеров:

* языки низкого уровня
* языки высокого уровня

**2.2.2.1 Ассемблер**

Языком низкого уровня является Ассемблер. Здесь каждому оператору соответствуют не более одной машинной команды. Такой язык программирования микроконтроллера достаточно громоздкий и трудно понимаемся для человека. Тем не менее, альтернативы ему нет, например, когда в процессе программирования микроконтроллера имеются ограниченные ресурсы, такие как 8-ми битные модели с малым объемом памяти. Также данный язык обеспечивает достаточно большое быстродействие и компактность программного кода, что зачастую является немаловажным фактором.

**2.2.2.2 PL/M**

К основным языкам высокого уровня относятся следующие языки программирования микроконтроллеров: PL/M.

Этот язык был разработан фирмой DigitalResearch для микропроцессоров Intel, не имел большой популярности, так как не мог составить конкуренцию таким языкам, как Pascal и Модула-2.

**2.2.2.3 Pascal**

Язык Pascal достаточно просто и удобен в применении, но в среде программирования микроконтроллеров не получил большого распространения. Его используют для простоты перехода с персональных компьютеров на контроллеры.

**2.2.2.4 Basic**

Этот язык программирования существует уже достаточно давно и в основном применялся для обучения программированию. Тем не менее, он используется и для программирования микроконтроллеров. Basic не рекомендуют начинающим специалистам, так как код его программы имеет плохую структурированность. Такой язык программирования микроконтроллеров можно применять при проектировании несложных электронных устройств.

**2.2.2.5 C/C++**

Данный язык программирования микроконтроллера является основным языком профессиональной работы с ним. Его популярность обоснована рядом причин. Стандартный язык имеет возможность преобразования исходного кода для нужного в работе микроконтроллера. Для этого необходимо учитывать архитектуру микроконтроллера выбранного типа и использовать компилятор. Также к преимуществам над другими языками программирования микроконтроллеров можно отнести факт наличия в C/C++ множества программных средств и библиотек, с которыми удобно работать начинающим специалистам.

При работе с языками программирования микроконтроллеров высокого уровня происходит увеличение производительности за счет замены одного оператора несколькими машинными командами. Языки программирования микроконтроллеров высокого уровня требуют больших затрат памяти, так как объем такой программы достаточно большой. Преимуществом их использования можно назвать возможность работы программы на различных микропроцессорах, при использовании программ-трансляторов.

**2.2.2.6 Выбор языка программирования**

За основу для достижения поставленной цели был выбран именно язык С, потому что он наиболее распространен и прост в освоении.

**2.3Компиляторы**

Для реализации поставленной задачи необходимо не только написать код для будущего устройства, но и собрать для него прошивку, чтобы микроконтроллер мог понимать то, что для него написано. В таких случаях применяют компиляторы.

Компилятор (от английского Compile – собирать, накапливать) – это ПО, которое создаётся для перевода программы, написанной на высокоуровневом языке программирования в машинный код, который в последствие будет исполняться процессором микроконтроллера.

В большинстве случаев компиляция программы происходит в полном объеме. Компилятор целиком считывает программу, проводит её пошаговый анализ (лексический, синтаксический, семантический), оптимизирует её, очищая от излишних конструкций, но сохраняя исходный смысл операций, и также целиком переводит её в машинный код.

Однако, иногда используется и построчная компиляция. В этом случае машинный код генерируется и исполняется для каждой полной грамматической конструкции исходного кода.

Для каждого языка программирования и практически для каждой операционной системы используется свой компилятор. Иногда для одного семейства операционных систем может использоваться один и тот же компилятор.

Для микроконтроллеров существует следующий набор компиляторов:

1. Известные компиляторы Си:

* GNU Compiler Collection – поддерживает ARM, AVR, MSP430 и многие другие архитектуры
* [Small Device C Compiller](https://ru.wikipedia.org/wiki/Small_Device_C_Compiler) — поддерживает множество архитектур
* [CodeVisionAVR](https://ru.wikipedia.org/wiki/CodeVisionAVR) (для AVR)
* [IAR](https://ru.wikipedia.org/w/index.php?title=IAR_Embedded_Workbench&action=edit&redlink=1) (для любых МК)
* [WinAVR](https://ru.wikipedia.org/wiki/WinAVR) (для AVR и AVR32)
* Keil (для архитектуры 8051 и ARM)
* HiTECH (для архитектуры 8051 и PIC от Microchip)

1. Известные компиляторы [бейсика](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B9%D1%81%D0%B8%D0%BA) для МК:

* [MikroBasic](https://ru.wikipedia.org/w/index.php?title=MikroBasic&action=edit&redlink=1) (архитектуры PIC, AVR, 8051 и ARM)
* [Bascom](https://ru.wikipedia.org/wiki/Bascom-avr) (архитектуры AVR и 8051)
* [FastAVR](https://ru.wikipedia.org/w/index.php?title=FastAVR&action=edit&redlink=1) (для архитектуры AVR)
* [PICBasic](https://ru.wikipedia.org/w/index.php?title=PICBasic&action=edit&redlink=1) (для архитектуры PIC)
* [Swordfish](http://www.sfcompiler.co.uk/) (для архитектуры PIC)

Поскольку за основу был выбран язык С, для решения задачи будет использоваться компилятор AVG gcc. Это один из самых популярных и доступных компиляторов для разрабатываемой в рамках этой работы системы.

* 1. **Среды разработки**

В данный момент существует большое количество сред разработки для различных языков программирования.

Среда разработки или IDE (сокр. англ. *Integrated Development Environment* – интегрированная среда разработки) – это специальный программный комплекс, предназначенный для полного цикла написания и тестирования программ на определенном языке. Пример среды разработки представлен на рисунке 9.

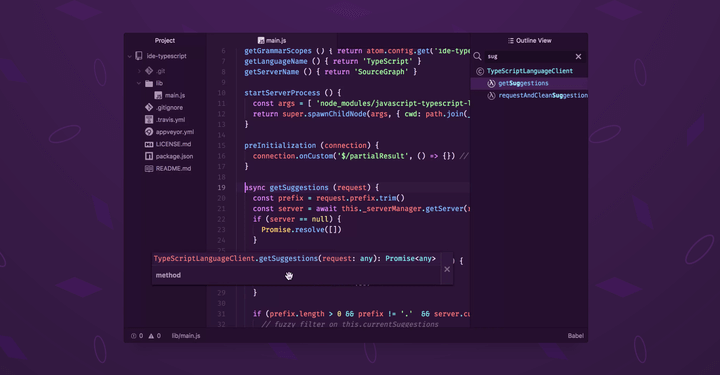


Рисунок 9 – Рабочее окно IDE

Типичная среда разработки содержит:

* Текстовый редактор с подсветкой синтаксиса языка, для которого разработана среда;
* Менеджер файлов и/или объектов;
* Компилятор или интерпретатор команд языка программирования;
* Средство обработки ошибок (дебагер);
* Инструменты для сборки проекта приложения в готовую программу.

Использование среды разработки для написания программ не является обязательным для небольших проектов (иногда достаточно и простого текстового редактора). Однако, при создании сложных программ и игр с большим количеством объектов IDE позволяет значительно лучше ориентироваться во всех материалах.

К достоинствам интегрированных сред разработки стоит отнести тот факт, что большинство из них имеют инструменты для визуализации процесса создания интерфейса программ.

Однако, у IDE есть и недостатки. Главный из них – довольно сложная структура интерфейса. Это требует от разработчика дополнительного изучения всех нюансов настройки и управления средой.

Поскольку разработка будет проводиться на языке СИ, будет использоваться QT Creator.

Qt Creator– [кроссплатформенная](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [IDE](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) для разработки на С, С++ и [QML](https://ru.wikipedia.org/wiki/QML).

Разработана Trolltech *(Digia)* для работы с фреймворком [Qt](https://ru.wikipedia.org/wiki/Qt" \o "Qt). Включает в себя графический интерфейс отладчика и визуальные средства разработки интерфейса как с использованием QtWidgets, так и QML. Внешний вид программы Qt Creator приведен на рисунке 10.

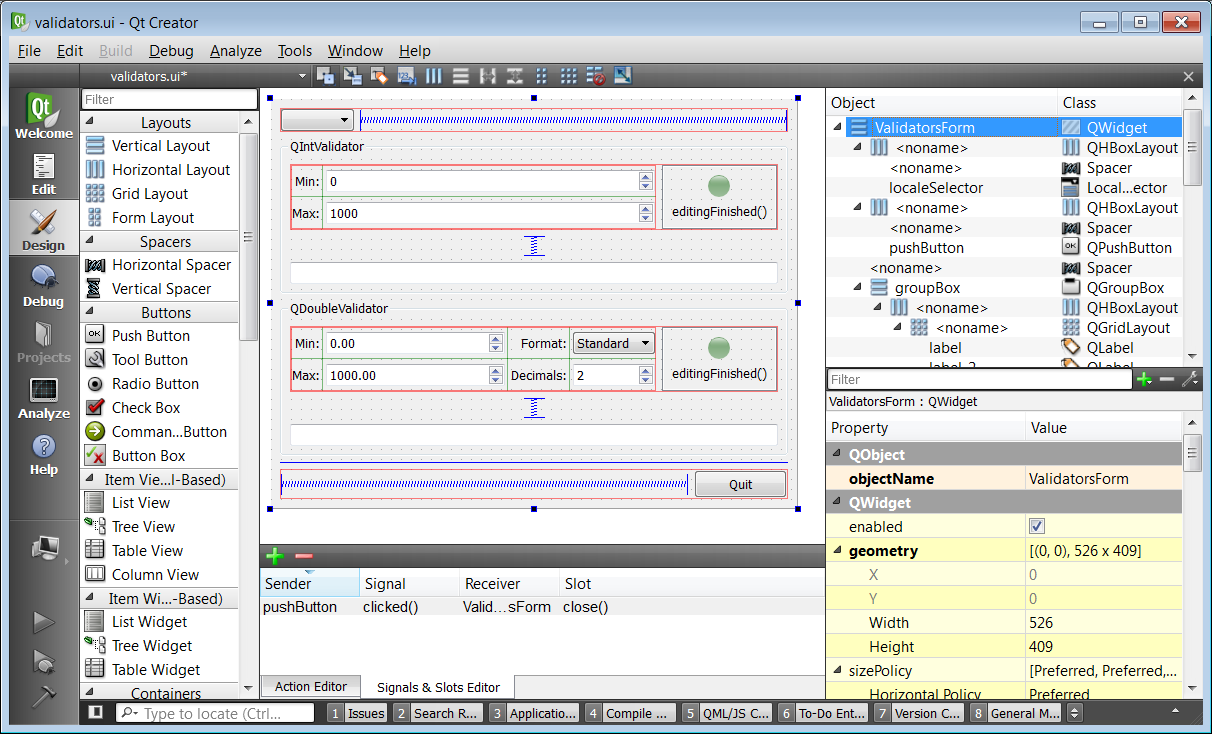


Рисунок 10 – Рабочее окно Qt Creator

Основная задача Qt Creator — упростить разработку приложения с помощью фреймворка Qt на разных платформах. Поэтому среди возможностей, присущих любой среде разработки, есть и специфичные, такие как отладка приложений на QML и отображение в отладчике данных из контейнеров Qt, встроенный дизайнер интерфейсов как на QML, так и на QtWidgets.

* 1. **Выбор микроконтроллеров**

Ключевым элементом в схеме устройств, которые будут разрабатываться в рамках этой работы, является микроконтроллер. В данный момент на рынке имеется большое количество различных компаний, которые занимаются разработкой микроконтроллеров и всего, что с ними связано.

* + 1. **Первый микроконтроллер**

   Первый микроконтроллер появился на свет в 1976 году, через 5 лет после создания первого микропроцессора. Это была микросхема фирмы Intel, получившая имя 8048.

   Помимо центрального процессора, на кристалле находились 1 КБайт памяти программ, 64 байта памяти данных, два восьмибитных таймера, генератор часов и 27 портов ввода/вывода.

   Микроконтроллеры семейства 8048 использовались в игровых консольных приставках Magnavox Odyssey, в клавиатурах первых IBM PC и в ряде других устройств.

* + 1. **8051**

   Следующий микроконтроллер Intel 8051, выпущенный в 1980 году, стал поистине классическим образцом устройств данного класса. Этот 8-битный чип положил начало целому семейству микроконтроллеров, которые господствовали на рынке вплоть до недавнего времени.

Большинство фирм производителей микроконтроллеров и сегодня выпускают устройства, основанные на этой архитектуре. Среди них Atmel, Maxim, Philips, Dallas, OKI, Siemens — можно перечислить более полутора десятков имен. Но 51-е семейство сдает свои позиции более молодым и совершенным микроконтроллерам.

* + 1. **Motorola и Zilog**

Другими яркими представителями восьмиразрядных микроконтроллеров явились изделия компаний Motorola (68HC05, 68HC08, 68HC11) и Zilog (Z8).

   Motorola длительное время не предоставляла средств, позволяющих дешево и быстро начать работать с ее контроллерами, что явно не способствовало их популярности у некорпоративных разработчиков. Однако стоит заметить, что за рубежом микроконтроллеры от Motorola занимают лидирующее положение на рынке. В нашей стране их популярность не очень высока, возможно, еще в силу отсутствия достаточного количества доступных учебных материалов и средств разработки.

Микроконтроллеры фирмы Zilog, основанной бывшими сотрудниками Intel, еще недавно казавшиеся столь многообещающими, не выдержали гонки в стремительно развивающемся секторе рынка, и сегодня система команд Z8 выглядит достаточно устаревшей.

* + 1. **PIC от Microchip**

Первые значительные перемены произошли с появлением PIC-контроллеров фирмы Microchip. Эти чипы предлагались по рекордно низким ценам, что позволило им в короткий срок захватить значительную часть рынка микроконтроллеров. К тому же кристаллы от Microchip оказались не уступающими, а нередко и превосходящими микроконтроллеры х51 по производительности и не требовали дорогостоящего программатора.

   Вместе с контроллерами появились дешевые комплекты PICSTART, содержащие все, что было нужно для того, чтобы, не имея ни средств, ни навыков работы с PIC-контроллерами, быстро создать и отладить на нем продукт.   
   Эти микроконтроллеры имели хорошие порты, но все остальное было сделано весьма неудобно. Архитектура оставляла желать лучшего, система команд была крайне ограничена. Тем не менее, PIC-контроллеры остаются популярными в тех случаях, когда требуется создать недорогую систему, не предъявляющую высоких требований по ее управлению. Микроконтроллер компании Microchip предоставлен на рисунке 11.



Рисунок 11 – Микроконтроллер компании Microchip

Чтобы перепрограммировать такой PIC-микроконтроллер, необходимо было посветить некоторое время в специальное окошечко кварцевой лампой. Сегодня новые микроконтроллеры от Microchip оснащаются электрически перепрограммируемой Flash-памятью программ.

**2.5.5 AVR от Atmel**

  Настоящая революция в мире микроконтроллеров произошла в 1996 году, когда корпорация Atmel представила свое семейство чипов на новом прогрессивном ядре AVR. Более продуманная архитектура AVR, быстродействие, превосходящее контроллеры Microchip, привлекательная ценовая политика способствовали оттоку симпатий многих разработчиков от недавних претендентов на звание контроллера номер 1. Микроконтроллер компании Atmel предоставлен на рисунке 12.



Рисунок 12 – Микроконтроллер компании Atmel

Микроконтроллеры AVR имеют более развитую систему команд, насчитывающую до 133 инструкций, производительность, приближающуюся к 1 MIPS/МГц, Flash ПЗУ программ с возможностью внутрисхемного перепрограммирования. AVR-архитектура оптимизирована под язык высокого уровня Си. Огромную роль сыграла доступность программного обеспечения и средств поддержки разработки. У Atmel много бесплатно распространяемых программных продуктов. Фирма Atmel уделяет этому вопросу большое внимание и выпускает чрезвычайно удачную и совершенно бесплатную среду разработки Atmel Studio, работающую под Windows.

Сторонние производители выпускают полный спектр компиляторов, программаторов, ассемблеров, отладчиков, разъемов и адаптеров.

 Для начинающих разработчиков немаловажным является и то, что для программирования AVR можно обойтись вовсе без аппаратного программатора. Самым популярным способом программирования этих микроконтроллеров долгое время являлись пять проводков, подсоединенных к параллельному порту ПК.  
   Можно считать, что AVR стали еще одним индустриальным стандартом среди микроконтроллеров общего назначения. Они легкодоступны в России и отличаются в среднем невысокой стоимостью, успешно конкурируя с микроконтроллерами PIC. Все это делает Atmel AVR одними из самых привлекательных для обучения.

* + 1. **STM от STMicroelectronics**

Восьмиразрядные микроконтроллеры STM8 были выпущены в 2008 году и позиционировались для задач, требующих высокой надежности при низком энергопотреблении. Они сразу обратили на себя внимание невысокой ценой и широким модельным рядом. Работа с микроконтроллерами STM8 не вызвала никаких сложностей как у тех, кто уже освоил работу с микроконтроллерами PIC или AVR, так и у начинающих, и они быстро обрели широкую популярность. STM8 имеют хороший набор периферии и развитые средства программирования.

Микроконтроллер компании SMT предоставлен на рисунке 13.



Рисунок 13 – Микроконтроллер компании STMicroelectronics

 Чуть раньше, в 2004 году, Acorn разработали 32-разрядное процессорное ядро ARM Cortex-M3 и компания STMicroelectronics стала одной из первых, кто вывел на рынок семейство микроконтроллеров на этом ядре, получивших название STM32. Сегодня ARM Cortex-M3 становится, возможно, самым популярным 32-разрядным процессорным ядром для встраиваемых систем, а микроконтроллеры на его базе имеют универсальную, хорошо сбалансированную архитектуру.

* + 1. **Выбор микроконтроллеров**

Из-за своей доступности и распространенности, для построения модулей индикации будут использоваться микроконтроллеры AVR от компании Atmel.

* 1. **Выбор компонентов**

Для того, чтобы решить поставленную задачу, необходимо разработать 5 модулей индикации:

* сигнальный;
* текстовый;
* графический;
* звуковой;
* речевой.
  + 1. **Сигнальный модуль**

Сигнальный модуль будет построен на основе светодиодов и микроконтроллера, который будет подавать на них сигнал. Светодиодов на рынке имеется огромное множество, которые практически не отличаются друг от друга, поэтому детально рассматриваться эти компоненты не будут.

* + 1. **Текстовый модуль**

Текстовый модуль основан на LED дисплее. Дисплеи этого вида могут одновременно отображать определенное количество символов, ограниченное геометрическими размерами. Внешний вид таких дисплеев представлен на рисунке 14.



Рисунок 14 – LED дисплеи

Маркируются они по такому образцу:

* 1602;
* 2002.

Где первые две цифры – количество символов в строке, а вторая пара – количество строк. Таким образом дисплей с названием 1602 может отображать одновременно 2 строки по 16 символов.

По типу ввода данных различают дисплеи:

* С параллельным вводом данных;
* С вводом данных по протоколу I2C.

Параллельный ввод данных предполагает передачу 8 или 4-битных слов по 10 или 6 выводам соответственно. Кроме данных на дисплей подаётся питание.

Одними из самых популярных дисплеев в данном сегменте являются дисплеи на основе HD44780 от фирмы HITACHI. Дисплеи именно такого типа будут использоваться в реализации этого модуля. Текстовый дисплей представлен на рисунке 15.

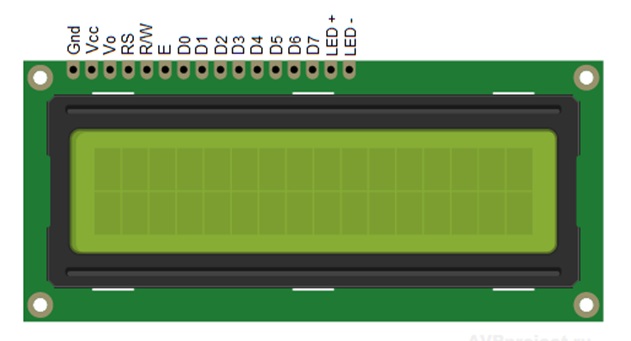


Рисунок 15 – LED дисплей HD44780

Среди отечественных производителей можно выделить фирму МЭЛТ у которой есть целый ряд различных дисплеев. Например, ниже изображен дисплей с маркировкой 20S4, по аналогии с предыдущей рассмотренной, это говорит нам о том, что он отображает 4 строки по 20 знаков. Внешний вид этого дисплея представлен на рисунке 16.



Рисунок 16 – LED дисплей фирмы МЭЛТ

**6.2.3 Графический модуль**

Графический модуль основывается на GLED дисплее. Эти дисплеи отличаются от рассмотренных выше преимущественно размерами (зачастую 128 x 64).

Основные компании, которые занимаются производством такими дисплеями, являются WinStar, intech и МЭЛТ. Внешний вид таких дисплеев представлен на рисунке 17.



Рисунок 17 – Графический LED дисплей

Помимо вышеперечисленных фирм и моделей графических дисплеев, нередко используют дисплеи от мобильных телефонов. Самым распространенным дисплеем такого типа является графический LCD дисплей от Nokia 5110 размерами 84 x 48. Этот дисплей представлен на рисунке 18.

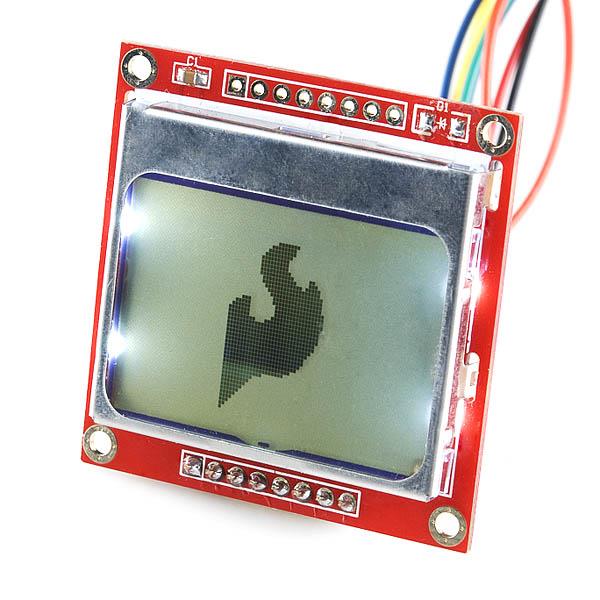


Рисунок 18 – Графический LED дисплей Nokia 5110

Из-за своей распространенности и простоты реализации, в этой работе будет использоваться дисплей именно такого типа.

**6.2.4 Звуковой и Речевой модуль**

Для создания звукового и речевого индикатора будут использоваться пьезодинамик. Современные микроконтроллеры имеют достаточно памяти и ресурсов, для того чтобы проигрывать файлы формата Wav. Для расширения функционала и проигрывания файлов в формате MP3 можно добавить модуль, который может работать с Micro-SD картами. Такой модуль представлен на рисунке 19.

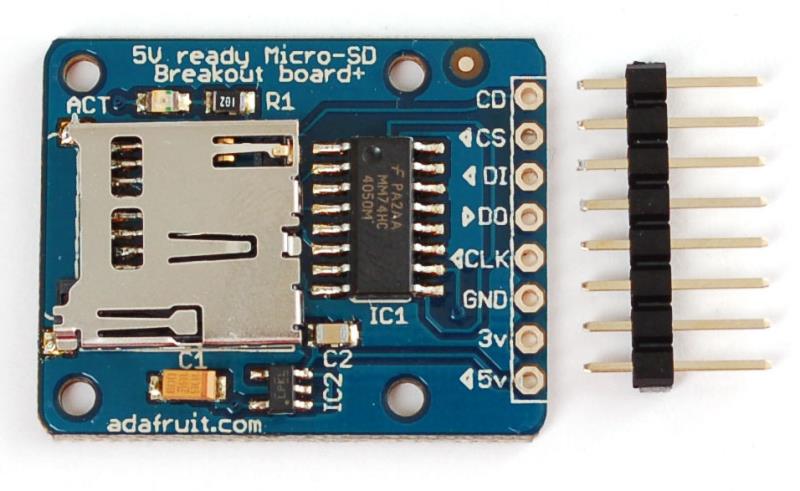


Рисунок 19 – Модуль чтения Micro-SD карт

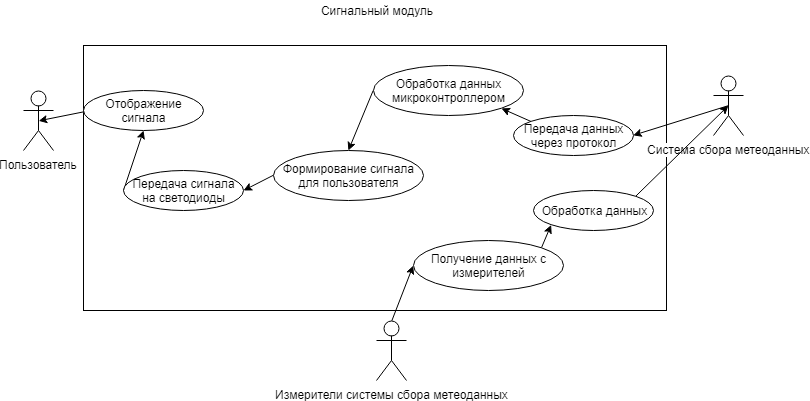
# Аспекты реализации

# 3.1 Разработка архитектуры модулей индикации

# 3.1.1 Диаграмма прецедентов

# Диаграмма прецедентов для сигнального модуля индикации представлена на рисунке 20.

# 



# Рисунок 20 – Диаграмма прецедентов для сигнального модуля

Для построения диаграммы следует выделить следующие субъекты в системе.

Таблица 1 – Описание субъектов системы

|  |  |
| --- | --- |
| Субъект | Описание |
| Пользователь | Данный субъект работает с системой сбора данных и на основе полученной информации от модулей индикации оценивает состояние системы. |
| Система сбора метеоданных | Субъект, который собирает данные от измерителей и обрабатывает их. |
| Модуль индикации | Данный субъект сообщает пользователю состояние системы, которое получает от системы сбора метеоданных. |

Для системы следует выделить следующие прецеденты.

Таблица 2 – Описание прецедентов

|  |  |
| --- | --- |
| Прецедент | Описание |
| Получение данных с измерителей | Данные от измерителей поступают в систему сбора метеоданных. |
| Обработка данных | Данные обрабатываются и готовятся к передаче на модуль индикации через протокол. |
| Передача данных через протокол | Подготовленные данные отправляются на модуль индикации. |
| Обработка данных микроконтроллером | Полученная от системы сбора метеоданных информация обрабатывается. |
| Формирование сигнала для пользователя | В зависимости от полученных данных, формируется сигнал для отображения состояния. |
| Передача данных на светодиоды | Данные в виде сигналов передаются на светодиоды, тем самым включая или отключая их. |
| Отображение сигнала | Пользователю представляется сигнал, на основе которого он может сделать вывод о состоянии системы. |

# Диаграмма прецедентов для текстового и графического модуля индикации представлена на рисунке 21.

# 

# Рисунок 21 – Диаграмма прецедентов для текстового/графического модуля

Для построения диаграммы следует выделить следующие субъекты в системе.

Таблица 3 – Описание субъектов системы

|  |  |
| --- | --- |
| Субъект | Описание |
| Пользователь | Данный субъект работает с системой сбора данных и на основе полученной информации от модулей индикации оценивает состояние системы. |
| Система сбора метеоданных | Субъект, который собирает данные от измерителей и обрабатывает их. |
| Модуль индикации | Данный субъект сообщает пользователю состояние системы, которое получает от системы сбора метеоданных. |

Для системы следует выделить следующие прецеденты.

Таблица 4 – Описание прецедентов

|  |  |
| --- | --- |
| Прецедент | Описание |
| Получение данных с измерителей | Данные от измерителей поступают в систему сбора метеоданных. |
| Обработка данных | Данные обрабатываются и готовятся к передаче на модуль индикации через протокол. |
| Передача данных через протокол | Подготовленные данные отправляются на модуль индикации. |
| Обработка данных микроконтроллером | Полученная от системы сбора метеоданных информация обрабатывается. |
| Формирование данных для пользователя | В зависимости от полученных данных, они подготавливаются к отображению на дисплее. |
| Передача данных на дисплей | Данные в виде текста или изображений передаются на дисплей, тем самым выводя необходимую информацию пользователю. |
| Отображение данных | Пользователю представляются данные, на основе которых он может сделать вывод о состоянии системы. |

# Диаграмма прецедентов для звукового и речевого модуля индикации представлена на рисунке 22.

# 

# Рисунок 22 – Диаграмма прецедентов для звукового/речевого модуля

Для построения диаграммы следует выделить следующие субъекты в системе.

Таблица 5 – Описание субъектов системы

|  |  |
| --- | --- |
| Субъект | Описание |
| Пользователь | Данный субъект работает с системой сбора данных и на основе полученной информации от модулей индикации оценивает состояние системы. |
| Система сбора метеоданных | Субъект, который собирает данные от измерителей и обрабатывает их. |
| Модуль индикации | Данный субъект сообщает пользователю состояние системы, которое получает от системы сбора метеоданных. |

Для системы следует выделить следующие прецеденты.

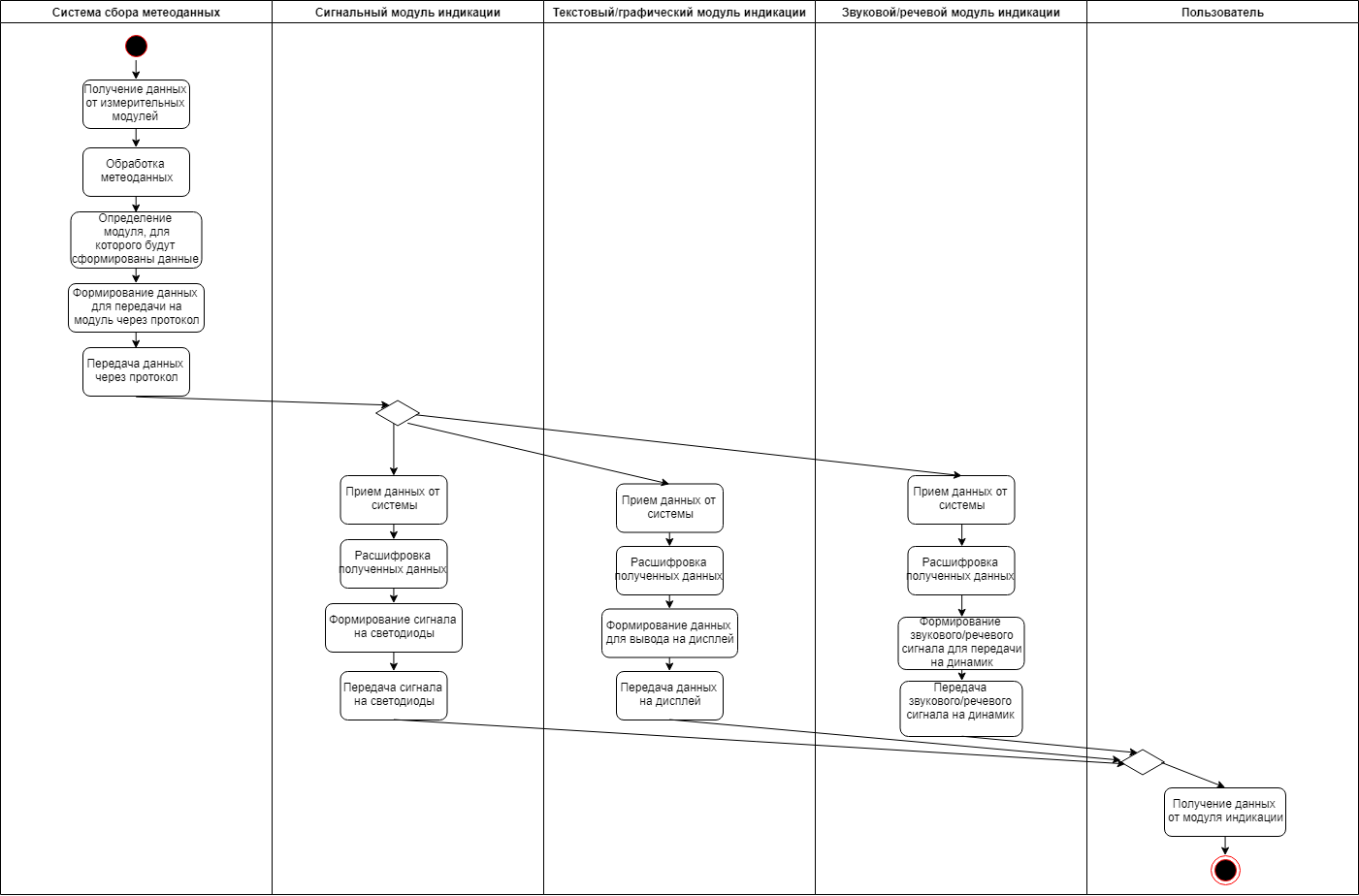
Таблица 6 – Описание прецедентов

|  |  |
| --- | --- |
| Прецедент | Описание |
| Получение данных с измерителей | Данные от измерителей поступают в систему сбора метеоданных. |
| Обработка данных | Данные обрабатываются и готовятся к передаче на модуль индикации через протокол. |
| Передача данных через протокол | Подготовленные данные отправляются на модуль индикации. |
| Обработка данных микроконтроллером | Полученная от системы сбора метеоданных информация обрабатывается. |
| Формирование звуковых данных для пользователя | В зависимости от полученных данных, формируется звуковой или речевой сигнал для отображения состояния. |
| Передача данных на динамик | Данные в виде звуковых сигналов передаются на динамик, тем самым воспроизводя определенные звуки или фразы. |
| Получение звукового/речевого оповещения | Пользователь получает информацию от системы, которая позволяет сделать вывод касательно ее состояния. |

# 3.1.2 Диаграмма деятельности

# На рисунке 23 представлена диаграмма деятельности для модулей индикации.

# 



# Рисунок 23 – Диаграмма деятельности для модулей индикации

Описание диаграммы деятельности приведено в таблице 7.

Таблица 7 – Описание диаграммы деятельности.

|  |  |  |
| --- | --- | --- |
| Дорожка | Состояние | Описание |
| Система сбора метеоданных | Получение данных от измерительных модулей. | Система сбора метеоданных получает от измерительных модулей информацию. |

Таблица 7 - продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| Дорожка | Состояние |  | Описание |
| Система сбора метеоданных | Обработка метеоданных. | | Полученные данные обрабатываются системой. |
| Определение модуля, для которого будут сформированы данные. | | В зависимости от полученных данных, система определяет модуль, который должен вывести информацию о состоянии системы пользователю. |
| Формирование данных для передачи на модуль через протокол. | | Данные подготавливаются для передачи на выбранный модуль индикации. |
| Передача данных через протокол. | | Передача данных через протокол на выбранный модуль индикации. |
| Сигнальный модуль индикации | Прием данных от системы. | | Микроконтроллер получает данные, отправленные системой. |
| Расшифровка полученных данных. | | Полученные данные от системы, микроконтроллер обрабатывает для формирования информации пользователю. |

Таблица 7 - продолжение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дорожка | Состояние |  | Описание | |
| Сигнальный модуль индикации | Формирование сигнала на светодиоды. | | В зависимости от информации, полученной от системы, формируется набор сигналов для отображения. | |
| Передача сигнала на светодиоды | | Светодиоды включаются согласно полученным данным от системы. | |
| Текстовый/графический модуль индикации | Прием данных от системы. | | | Микроконтроллер получает данные, отправленные системой. |
| Расшифровка полученных данных. | | | Полученные данные от системы, микроконтроллер обрабатывает для формирования информации пользователю. |
| Формирование данных для вывода на дисплей. | | | В зависимости от информации, полученной от системы, формируется текст или изображение. |

Таблица 7 - продолжение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дорожка | Состояние |  | Описание | |
| Текстовый/графический модуль индикации | Передача данных на дисплей. | | | На дисплее отображается необходимая для пользователя информация. |
| Звуковой/речевой модуль индикации | Прием данных от системы. | | | Микроконтроллер получает данные, отправленные системой. |
| Расшифровка полученных данных. | | | Полученные данные от системы, микроконтроллер обрабатывает для формирования информации пользователю. |
| Формирование звукового/речевого сигнала для передачи на динамик. | | | В зависимости от информации, полученной от системы, формируется звуковых или речевых сигналов. |
| Передача звукового/речевого сигнала на динамик. | | | Модуль через динамик воспроизводит информацию для пользователя. |
| Пользователь | Получение данных от модуля индикации | | | Пользователь через модуль индикации получает информацию о состоянии системы. |