**Автономная некоммерческая общеобразовательная организация "Физтех-лицей"
(АНОО «Физтех-лицей» им. П.Л. Капицы)**

XX научно-практическая конференция

«Старт в инновации»

Станция мониторинга окружающей среды

Выполнил
ученик 10 «И» класса
Больщиков Константин Андреевич

Руководитель
Больщиков Виталий Андреевич

Московская область, г. Долгопрудный

2021 г.

Оглавление

Введение

Сегодня нельзя представить повседневную жизнь без современных технологий, инноваций и автоматизации рутинной деятельности. Автоматизация процессов является одной из наиболее быстро развивающихся направлений в IT индустрии. Не последнюю роль в автоматизации и контролировании занимают встраиваемые системы, благодаря экспоненциальному росту развития электроники и микроэлектроники. При проектировании встраиваемых систем доступен широкий спектр технологий, который не только можно, но и нужно использовать. Однако, на практике, чаще всего используется подход «суперцикл» и машина состояний. Ранее данный подход был “золотым” стандартом при проектировании встраиваемых систем. При достаточно невысокой сложности проектируемой системы, архитектура “суперцикл” позволяла существенно экономить ресурсы микроконтроллера, а так же была проста для понимания. Тем не менее, по мере развития электроники, возрастали и задачи, возлагаемые на встраиваемые системы. Простота и лаконичность подхода “суперцикл” начала обрастать дополнительными условными операторами и громоздкими синтаксическими конструкциями, что привело к неминуемой потере гибкости при внесении изменений в архитектуру проектируемой системы. Решение возникшей проблемы было в использовании операционных систем реального времени (ОСРВ), так как современные микроконтроллеры имеют достаточный технический потенциал, для их запуска. В настоящее время встраиваемые системы, работающие под управлением ОСРВ, одновременно решают ряд необходимых задач и обеспечивают прдсказуемую во времени реакцию на происходящие в системе события. Способность гарантировать время реакции на внешнее событие - является отличительным признаком подобных операционных систем. Грамотное использование подобных ОСРВ позволяет существенно сократить время на проектирование, разработку и внесение архитектурных изменений в случае необходимости.

Цель проекта: Разработать портативную станцию мониторинга параметров окружающей среды с использованием операционных систем реального времени.

Задачи:

1. Ознакомиться с техническими возможностями современных микроконтроллеров на архитектуре ARM;
2. Ознакомиться с базовыми принципами работы операционных систем реального времени;
3. Ознакомиться с основными интерфейсами при передачи данных встраиваемых системах;
4. Разработать драйвер для датчика температуры;
5. Спроектировать и реализовать архитектуру системы мониторинга
6. Протестировать систему на отказоустойчивость.
7. Опубликовать проект на ресурсе github.com

Методы исследования:

1. Анализ и изучение технической литературы, спецификаций и листингов программ.

2. Реализация собственных исходных кодов программы.

Предполагаемый результат:

Презентация макета портативной станции мониторинга параметров окружающей среды, наглядно демонстрирующая успешность выполненной работы. Также, презентация блок-схемы внутренней архитектуры проекта, позволяющая оценить гибкость и масштабируемость проекта.

1. Разбор работы ОСРВ

Операционные системы реального времени предназначены для обеспечения интерфейса к ресурсам критических по времени систем реального времени. Основной задачей в таких системах является своевременность выполнения обработки данных. В качестве основного требования к ОСРВ выдвигается требование обеспечения предсказуемости или детерминированности поведения системы в наихудших внешних условиях, что резко отличается от требований к производительности и быстродействию универсальных ОС. Хорошая ОСРВ имеет предсказуемое поведение при всех сценариях системной загрузки (одновременные прерывания и выполнение задач), пример отличия в работе суперцикла от ОСРВ представлен на рисунке 1 и 2 соответственно (блок-схемы реализованы в программе PlantUML).

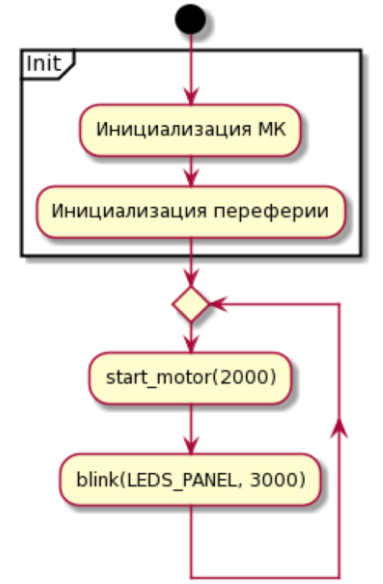

Рис 1. Блок-схема суперцикла

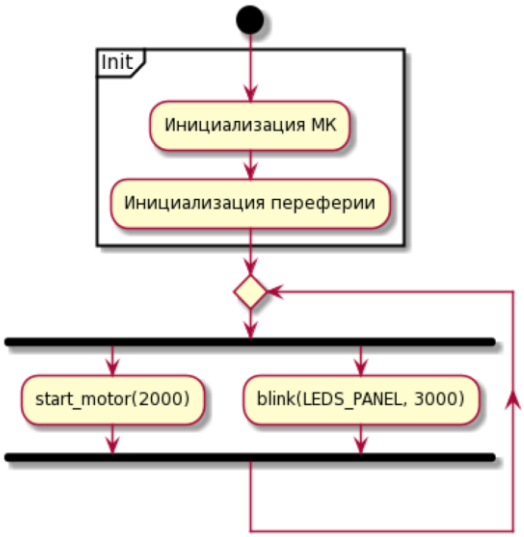

Рис.2 Блок-схема ОСРВ

Как видно из рисунка 1 и 2, подходы в проектировании разительно отличаются, тем что функции в ОСРВ выполняются параллельно. Таким образом ОСРВ обладает следующими принципиально критичными возможностями:

- Обеспечение многозадачности и допущение вытеснения
- Обеспечение приоритетности для задач
- Обеспечение предсказуемости механизмов синхронизации
- Обеспечение предсказуемым поведения, т.е. должно быть определено максимальное время отклика во всех сценариях рабочей нагрузки системы

В качестве ОСРВ в настоящем работе была выбрана FreeRTOS. Данная ОСРВ является бесплатной и портирована на многие МК, в том числе и STM32. STMicroelectronics официально поддерживает FreeRTOS, своевременно выпускает необходимые обновления при возникновении критических ошибок, а также

обеспечивает корректное взаимодействие между ядром ОСРВ и Hardware Abstraction Layer (HAL) - библиотекой для работы с периферией МК. FreeRTOS построена на основе микроядра (kernel), и обеспечивает планирование, диспетчеризацию задач, а также осуществляет их взаимодействие через примитивы синхронизации, такие как:

- Семафоры
- Мьютексы
- Критические секции

Как было отмечено ранее, FreeRTOS поддерживает работу в режиме многозадачности, что позволяет удобно взаимодействовать со многочисленными процессами внешнего мира (прерывания, периферийные интерфейсы). В этом режиме несколько процессов могут выполняться псевдопараллельно. т.е. задачи выполняются не одновременно, а поочередно - короткими временными отрезками, называемыми квантом времени. Пока выполняется одна задача, другие - ожидают получение возможности на выполнение, что отражено на рисунке 3.

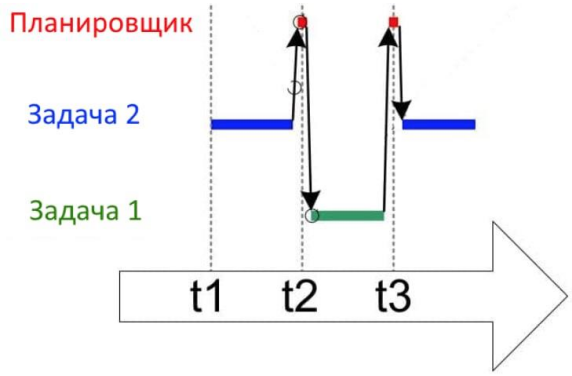

Рис. 3 Временная диаграмма выполнения задач

Как видно на рисунке 3, пока выполняется одна задача, другая ожидает переключения на выполнение. Переключение происходит благодаря работе планировщику задач в ОСРВ. Для организации ожидания используется механизм очереди. Данный тип многозадачности называется - вытесняющим. Каждой задаче выделяется короткий временной отрезок (квант времени), по истечении которого задача принудительно переключается («вытесняется»), сохраняя свой контекст, и управление передается задаче, находящейся первой в очереди готовых на исполнение задач. Данный режим работы не приводит к высокой реактивности системы, но обеспечивает гарантированный отклик на внешнее событие в пределах временной задержки. ОСРВ, работающая в условиях вытесняющей многозадачности, устойчива к ошибкам - «зависание» одной из задач приведет к потере квантов времени, выделенных этой задаче, но работоспособность остальных задач сохранится. В настоящей работе также используется вытесняющая многозадачность с квантом времени равным 1 мс.

2. Проектирование встраиваемой системы

Для демонстрации возможностей ОСРВ при проектировании встраиваемых систем, была выбрана задача мониторинга параметров окружающей среды и последующий вывод этих параметров в удобной для пользователя форме. Основными параметрами мониторинга были выбраны: атмосферное давление и температура. Данная система позволит пользователю автоматизировать контроль данных показателей в домашних условиях.

Большинство встраиваемых систем спроектированы на базе современных микроконтроллеров. Грамотный выбор МК открывает широкий спектр технических возможностей, поэтому чтобы выбрать нужный МК для данной системы была составлена таблица 1.

Таблица 1. Сравнительный анализ микроконтроллеров

Критерии	Ед. измерения	**esp8266**	**arduino nano**	**stm32f103**
КЛЮЧЕВЫЕ				
Технические характеристики				
Количество ядер	шт.	1	1	1
Частота процессора	MHz	80	16	72
Архитектура процессора (разрядность)	Bit	32	8	32
RAM	KB	160	2	20
Flash	MiB	1	32	64
Количество цифровых выходов	шт.	3	6	14
Количество аналоговых входов	шт.	5	8	10
Наличие Bluetooth (0 - нет; 1 - есть)	0/1	0	0	0
Цена	Р.	500	300	300
Интерфейсы				
Количество SPI	шт.	1	3	3
Количество I2C	шт.	1	2	2
Количество UART	шт.	2	1	5
Количество I2S	шт.	1	0	2

В таблице 1 приведены одни из самых популярных МК на рынке на данный момент. Из данной таблицы очевидно, что STM32F103 и esp8266 обладают наиболее высоким потенциалом с технической точки зрения, предлагая широкий спектр возможностей, однако stm32 более экономически выгоден, что также немаловажно. Так же, STMicroelectronics предлагает для своих контроллеров бесплатную и постоянно обновляемую среду для разработки STMCubeIDE. В данную среду встроен удобный визуальный механизм настройки выбранного процессора, и интегрированы дополнительные внешние библиотеки, такие как FreeRTOS и FatFs, что значительно ускоряет процесс разработки.

Также, необходимо определится с выбором датчика температуры и влажности, который позволит мониторить параметры окружающей среды. Сравнительный анализ датчиков приведен в таблице 2.

Таблица 2. Сравнительный анализ датчиков температуры и давления

Датчики температуры и давления				
Критерии	Ед. измерения	**DHT22**	**BMP180**	**BME280**
КЛЮЧЕВЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ				
Температура				
Мин. температура	°C	-40	-40	-40
Макс. температура	°C	80	85	85
Погрешность температуры	Δ °C	0.5	1.0	0.01
Макс. частота измерений	Гц (измер/сек)	0.5	120	157
Разрешение шкалы температуры	°C	0.1	0.1	0.01
Давление				
Мин. давление	гПа	-	300	300
Макс. давление	гПа	-	1100	1100
Погрешность	Δ гПа	-	2	1.3
Разрешение шкалы давления	гПа	-	1	0.16
Другое				

Цена	Р.	280	80	60
Напряжение питания	В	3-6	1.6-3.6	3.3
Протоколы подключения	-	Single wire	I2C	SPI, I2C

Из данной таблицы видно, что датчик атмосферного давления и температуры BMP280 является оптимальным для проектируемой системы, как с точки зрения технических возможностей, так и с точки зрения цены. BMP280 имеет неоспоримое преимущество по количеству возможных интерфейсов взаимодействия и точности измерений.

Для удобства пользователя вывод информации предполагается делать на экран смартфона посредством протокола беспроводной связи bluetooth. Сравнительный анализ модулей bluetooth представлен в таблице 3.

Таблица 3. Сравнительный анализ bluetooth модулей

Bluetooth модули			
Критерии	Ед. измерения	**HM-10**	**HC-06**
КЛЮЧЕВЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ			
Версия Bluetooth	-	4.0	2.0
Питание	V	5	3.6-6
Цена	Р.	250	340

В таблице 3 приведены одни из самых популярных bluetooth модулей на рынке на данный момент. Из данной таблицы видно, что данные модули практически идентичны с точки зрения физической реализации (оба модуля осуществляют обмен данными с МК через последовательный асинхронный интерфейс UART) и цены. Однако, неоспоримое преимущество HM-10 это поддержка BLE технологии, что позволяет сопрягать данный модуль не только со смартфонами под управлением OS Android, но и устройствами под управлением OS iOS.

3. Структурная схема проектируемой встраиваемой системы

Структурные схемы позволяют определить основные функциональные части проектируемой системы, их назначение и взаимосвязи, и служат для общего ознакомления. На структурной схеме раскрывается взаимодействие отдельных частей системы между собой. На рисунке 4 изображена структурная схема, проектируемой встраиваемой системы.

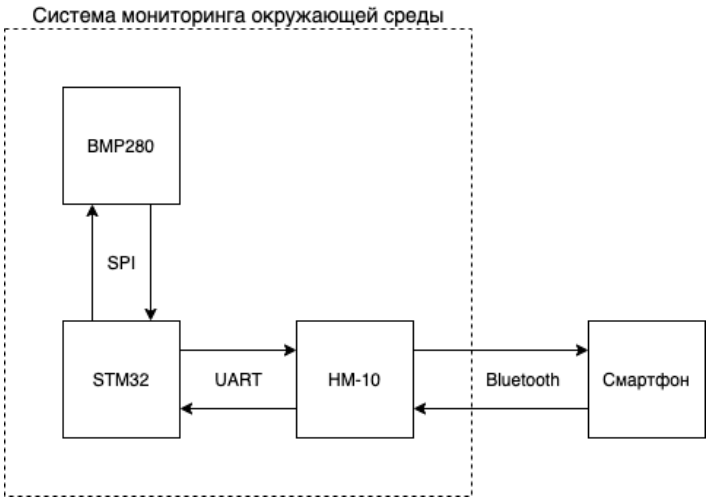

Рис. 4 Структурная схема системы мониторинга окружающей среды

На структурной схеме отлично видны составные части системы, что упрощает не только ознакомление с проектом, но и дальнейшую программную разработку системы.

4. Программная реализация

Как было отмечено ранее, ключевой элемент любой ОСРВ и FreeRTOS, в частности, это задача. В настоящей работе проект был разбит на 2 основные задачи, выполняющиеся параллельно, как показано на рисунке 5.

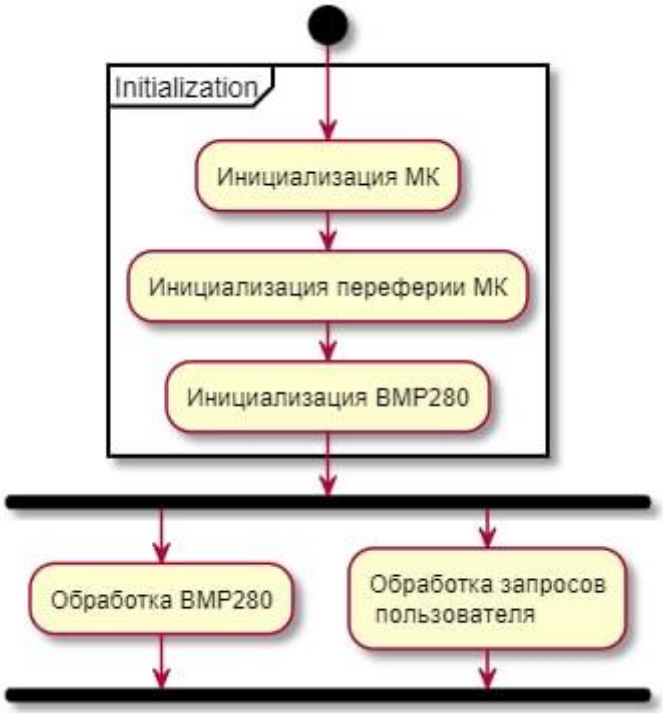

Рис. 5 Блок-схема общего вида программной реализации системы

На рисунке 5 видно, что прежде чем запустится планировщик FreeRTOS, выполняется блок инициализации. В первую очередь происходит инициализация МК, где устанавливается частота тактирования и инициализируется HAL библиотека для работы с периферией. Далее выполняется настройка цифровых портов (GPIO), интерфейсов (SPI, UART) и планировщика задач FreeRTOS. Последний этап

инициализации - это настройка BMP280. Датчик температуры и давления настраивается на работу по интерфейсу SPI т.к. BMP280 через SPI работает на более быстрых скоростях (10 МГц), чем I2C (3.4 МГц), хоть и требует для этого дополнительные цифровые порты.

Каждая из задач, абсолютно независимы и выполняются параллельно, что отображено на рисунках 6 и 7 соответственно.

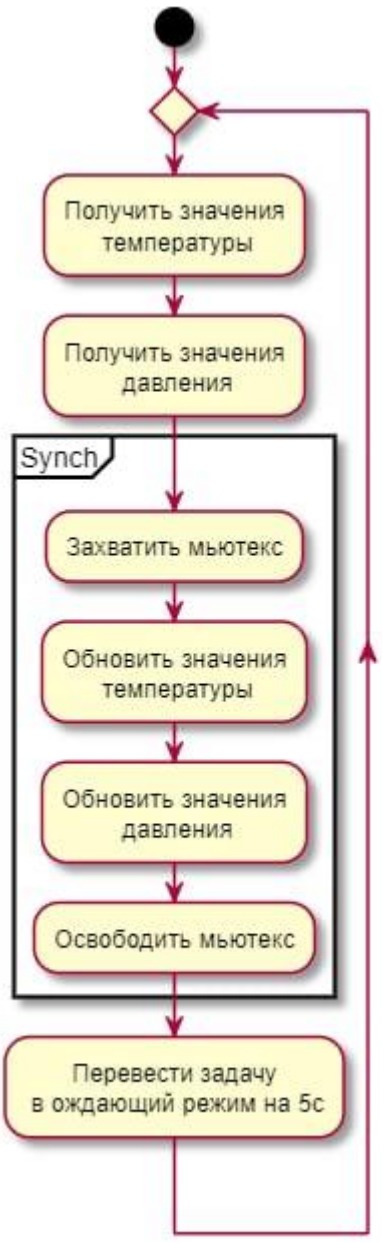

Рис. 6 Блок-схема задачи получения температуры

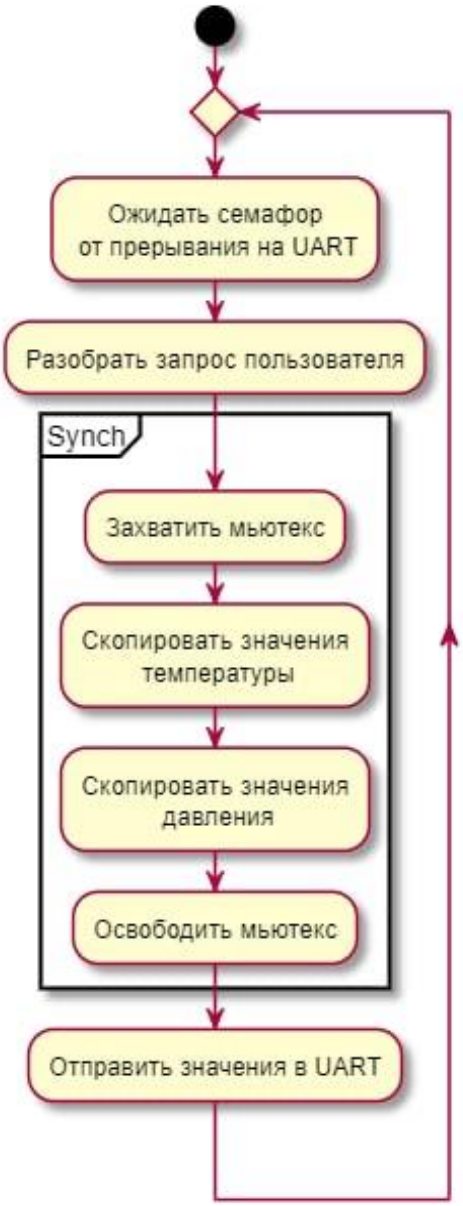

Рис. 7 Блок-схема задачи обработки запросов от пользователя

На рисунке 6 видно, что задача зациклена и выполняется бесконечно. Сначала выполняется запрос к BMP280 для получения значений температуры и давления. Затем необходимо обновить данные, однако это разделяемый ресурс между 2-мя задачами, который нуждается в синхронизации очередности доступа. Для защиты разделяемого ресурса был использован примитив синхронизации - мьютекс. После захвата мьютекса,

задача имеет право обновить значения температуры и давления. Затем необходимо освободить мьютекс, чтобы другая задача могла обратиться к разделяемому ресурсу. На последнем этапе задача переводится в ожидающий режим на 5с (время выбрано эмпирическим путем).

На рисунке 7 видно, что задача также зациклена и выполняется бесконечно. Данная задача находится в ждущем режиме бесконечно долго, т.к. ожидает захват примитива синхронизации - семафора. Семафор будет доступен только в случае возникновении прерывания на линии UART. Прерывание на линии UART может возникнуть только в случае, если пользователь отправил соответствующий запрос через смартфон на модуль HM-10. После сигнализации семафора, задача разбирает запрос пользователя, и если контрольная сумма совпадает, то происходит захват мьютекса для получения доступу к данным температуры и давления. После освобождения мьютекса, данные могут быть отправлены пользователю на смартфон.

В данном проекте не предполагается написание отдельного ПО на смартфон, однако, это может стать логичным продолжением настоящей работы.

В процессе реализации описываемой системы мониторинга окружающей среды были использованы среды разработки: STM32CubeIDE и Visual Studio Code (PlantUML). С листингом программы можно ознакомиться в приложении 1.

5. Физическая реализация

Для того чтобы собрать макет эксперементальной установки, необходимо сконфигурировать пины STM32. Удобнее всего данную процедуру производить в STM32CubeIDE, где с помощью интуитивно понятного интерфейса можно назначить на ножки контроллера соответствующие функции, что видно на рисунке 8.

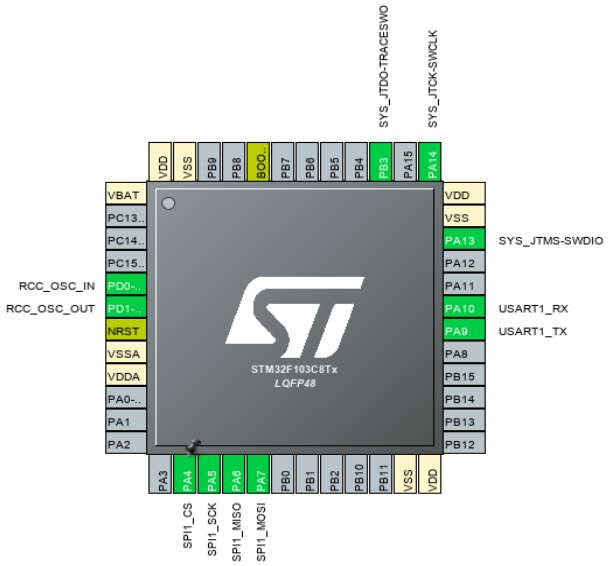

Рис. 8 Конфигурация STM32F103

На рисунке 8 видно, пины PA4-7 сконфигурированы для работы с SPI интерфейсом. Пины PA9-10 сконфигурированы для работы с UART интерфейсом. Остальные пины необходимы для тактирования контроллера от внешнего кварцевого резонатора и для подключения программатора.

Структурная схема на рисунке 4, дает лишь общее представление о виде реализуемой системы и не расскрывает схему сборки физической модели. Данный недостаток исправлен на рисунке 9.

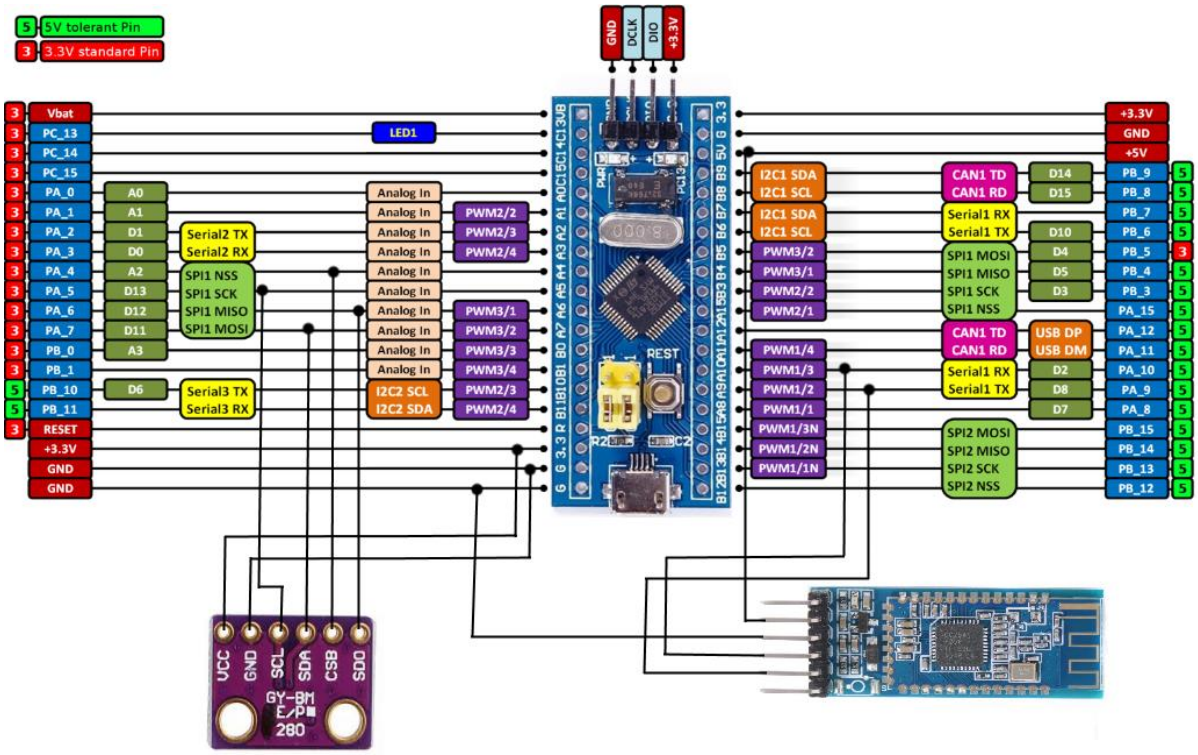

Рис. 8. Схема сборки системы мониторинга параметров окружающей среды

На рисунке 9 видно к каким пинам на макетной плате необходимо подсоеденить датчик bmp280 и Bluetooth-модуль, что без труда позволит собрать подобный макет для воспроизведения результатов эксперемента.

Заключение

ОС реального времени имеют ряд преимуществ перед ОС общего пользования. Она быстро переключается с задачи на задачу, запоминая свое состояние. При этом, никаких навыков для реализации таких функций не требуется, это встроено в систему. Так как мощности микроконтроллера ограничены, а время реакции на задачу критически важно, то ОСРВ подходят лучше ОСОП.

Станция мониторинга окружающей среды умеет измерять температуру и влажность. Полученные данные отправляются на смартфон.

В результате проделанной работы нами были выполнены все поставленные задачи:

- Была изучена литература для применения ОСРВ.
- Были изучены интерфейсы при передачи данных встраиваемых систем.
- Разработан драйвер для датчика температуры и давления.
- Применили многопоточное программирование.
- Протестировали систему на отказоустойчивость: отправляли неправильные команды в МК, проверяли подключенность датчиков.
- Код опубликован на https://github.com/kostyastrong/Environmental-monitoring-station-RTOS

Список использованных источников

1. КурницА. FreeRTOS — операционная система для микроконтроллеров [Электронный ресурс] // Компоненты и технологии. 2011. No2–11. Дата обращения 10.11.2020
2. Практическое руководство по программированию STMмикроконтроллеров [Электронный ресурс]: учебное пособие / С.Н. Торгаев [и др.]. — Электрон. текстовые данные. — Томск: Томский политехнический университет, 2015. — 111 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/55205.html Дата обращения 13.10.2020
3. Васильев А. С., Основы программирования микроконтроллеров. – Санкт – Петербург: Университет ИТМО, 2016. – 95 с.
4. Гусев В.Г. Электроника и микропроцессорная техника: учебник. – 6-е издание – Москва: КНОРУС, 2013. – 800 с.
5. Джозеф Ю., Ядро Cortex - M3 компании ARM. Полное руководство/ Джозеф Ю; пер. с англ. А. В. Евстифеева. – М.: Додэка-XXI, 2012. – 552с.: ил.
6. Клеменс Б. Язык С в XXI веке/пер. с английского А. А. Слинкина. – Москва.: ДМК Пресс, 2015. – 365 с.: ил.
7. Кузин, Александр Владимирович, Микропроцессорная техника: учебник: для студентов образовательных учреждений среднего профессионального образования, обучающихся по группам специальностей "Информатика и вычислительная техника", "Электротехника" / А. В. Кузин, М. А. Жаворонков. - 7- е изд., стер. - Москва: Академия, 2013. – 303 с.
8. Магда Ю. С., Программирование и отладка C/C++ приложений для микроконтроллеров ARM. – Москва.: ДМК Пресс, 2012. – 168 с.: ил
9. Огородников И.Н. Микропроцессорная техника: введение в CortexM3: учеб. пособие/ И.Н. Огородников. – Екатеринбург: изд-во Урал. Ун-та, 2015. – 116 с.
10. Страуструп Б. Язык программирования C++, Специальное издание, Пер. с англ. – М.: Издательство Бином, 2015 г. – 1136 с.: ил.
11. Техническое описание компетенции Инженер-проектировщик систем Интернета вещей [Электронный ресурс]. – URL: http://xn--h1aagpbh6b.xn--p1ai/upload/medialibrary/fbd/to_internet-veshchey.pdf / Дата обращения: 25.11.2020 г.
12. Бугаев В.И., Мусиенко М.П., Крайнык Я.М. Лабораторный практикум для изучения микроконтроллеров архитектуры ARM Cortex-M4 на базе отладочного модуля STM32F4 Discovery. — Москва-Николаев: МФТИ-ЧГУ, 2013.
13. J. Jiu. The Definitive guide to the ARM Cortext-M3. Newnes, 2010.
14. Datasheet STM32F103x8, STM32F103xB, DocID13587 Rev 17, 2015
15. Noviello C., Mastering STM32, a step-by-step guide to the most complete ARM Cortex – M platform, using a free and powerful development environment based on Eclipse and GCC, 2015 – 2016.

Приложение 1. Листинг PlantUml

```
@startuml
skinparam defaultTextAlignment center
start
repeat
:Ожидать семафор\n от прерывания на UART;
:Разобрать запрос пользователя;
partition Synch {
:Захватить мьютекс;
:Скопировать значения\n температуры;
:Скопировать значения\n давления;
:Освободить мьютекс;
}
:Отправить значения  в UART;
@enduml

@startuml
skinparam defaultTextAlignment center
start
repeat
:Получить значения\n температуры;
:Получить значения\n давления;
partition Synch {
:Захватить мьютекс;
:Обновить значения\n температуры;
:Обновить значения\n давления;
:Освободить мьютекс;
}
:Перевести задачу\n в ождающий режим на 5с;
@enduml

@startuml

start
partition Initialization {
:Инициализация МК;
:Инициализация переферии МК;
:Инициализация BMP280;
}

fork
    :Обработка BMP280;
fork again
    :Обработка запросов\n пользователя;
end fork

@enduml
```