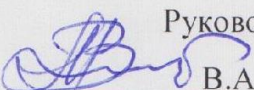


Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Радиофизический факультет
Кафедра радиоэлектроники

УДК 621.317.39.084.2

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

 Руководитель ООП
В.А. Мещеряков

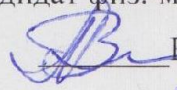
«15» февраля 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА

**РАЗРАБОТКА УСТРОЙСТВА УПРАВЛЕНИЯ СВЧ ДАТЧИКОМ-
ДАЛЬНОМЕРОМ НА ПЛАТФОРМЕ STM32 F4 С МИКРОКОНТРОЛЛЕРОМ
CORTEX M4**

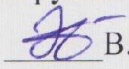
по основной образовательной программе подготовки специалиста
по специальности 11.05.01 «Радиоэлектронные системы и комплексы»

Руководитель ВКР
кандидат физ.-мат. наук, доцент

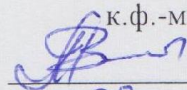
 В.А. Мещеряков

«14» февраля 2019

Автор отчета
студент группы №738

 В.В. Ни

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет
Кафедра радиоэлектроники

УТВЕРЖДАЮ
Руководитель ООП
к.ф.-м.н., доцент
 В.А. Мещеряков
« 28 » декабря 2018г.

ЗАДАНИЕ

на подготовку выпускной квалификационной работы (ВКР)
студенту 738 группы Ни Владиславу Витальевичу

1. Тема ВКР: Разработка устройства управления СВЧ датчиком-дальномером на платформе STM32 F4 с микроконтроллером Cortex M4.
2. Срок сдачи студентом выполненной ВКР:
 - а) на кафедре 10.02.2019,
 - б) в ГЭК 14.02.2019
3. Краткое содержание работы:

Задание ориентировано на написание выпускной квалификационной работы по теме «Разработка устройства управления СВЧ датчиком-дальномером на платформе STM32 F4 с микроконтроллером Cortex M4».

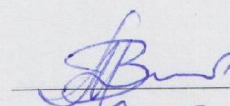
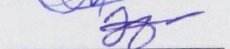
4. Календарный график выполнения ВКР:

а) оформление литературного обзора	15.01.2019–22.01-2019
б) оформление материалов по устройству управления датчиком	15.01.2019–22.01-2019
в) оформление конструкторско-технологической документации	22.01.2019–29.01-2019
г) оформление выводов по работе	29.01.2019–05.02-2019
д) техническое оформление ВКР в соответствии с требованиями	29.01.2019–05.02-2019
е) оформление документов на цитируемую чистоту работы	05.02.2019–12.02-2019
ж) оформление документов для сдачи работы в библиотеку	05.02.2019–12.02-2019

5. Дата выдачи задания « 28 » декабря 2018 г.

Руководитель ВКР –
кандидат физ.-мат. наук,
доцент кафедры радиоэлектроники

Задание принял к исполнению

 Мещеряков В.А.
 Ни В. В.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

МК – микроконтроллер;

АЦП – аналогово-цифровой преобразователь;

ЦАП – цифро-аналоговый преобразователь;

ШИМ – широтно-импульсная модуляция;

ОУ – операционный усилитель;

ФНЧ – фильтр низких частот;

РЕФЕРАТ

Выпускная квалификационная работа 33 с., 3 гл., 25 рис., 12 источников.

МИКРОКОНТРОЛЛЕР, АЦП, ГЕНЕРАТОР, ДАТЧИК-ДАЛЬНОМЕР, ЦАП

Объектом исследования является отладочная платформа STM32F429I-DISCO со встроенным МК STM32F429ZIT6U.

Цель работы – создание действующего макета устройства управления датчиком-дальномером.

В процессе работы производился поиск программного обеспечения, осваивалось программное обеспечение для программирования МК, рассматривались примеры программ, изучался способ контроля АЦП и ЦАП, разрабатывался алгоритм, позволяющий найти нули функции.

В результате работы:

- 1) Приведен литературный обзор;
- 2) Освоены базовые навыки программирования МК
- 3) Составлен алгоритм, позволяющий найти максимумы функции
- 4) Реализована программа, позволяющая снимать сигнал с датчика и находить максимумы функции.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 Система сбора и обработки информации. Обзор литературы	5
1.1 Система сбора и обработки информации	5
1.2 Семейство микроконтроллеров STM32	7
1.3 Короткое описание платы STM32F429I-DISCO	7
1.4 Система ввода/вывода	8
1.5 Вывод аналоговых сигналов. ЦАП	9
1.6 Ввод аналоговых сигналов. АЦП	10
1.7 Выбор среды разработки	10
1.8 Программные средства для программирования платы	12
2 Алгоритм определения расстояния до объекта	15
2.1 Алгоритм нахождения нулей функции	15
3 Экспериментальная часть	20
3.1 Разработка устройства генерации пилообразного напряжения	20
3.2 Разработка устройства обработки сигнала с датчика	20
3.3 Операционный усилитель	23
3.4 Устранения помех	24
3.5 Разработка корпуса	26
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	29
ПРИЛОЖЕНИЕ А (справочное) Фрагмент программы обработки сигнала с датчика и генерации пилообразного сигнала	31
ПРИЛОЖЕНИЕ Б (справочное) Схема электрическая принципиальная ОУ с ФНЧ	33

ВВЕДЕНИЕ

Существует проблема создания микроконтроллерной системы сбора и обработки информации с датчика. Эту проблему многие стремятся решить.

Данные датчики позволяют определять расстояние до измеряемого объекта. Они могут использоваться для различных целей.

В настоящее время существуют различные датчики: ультразвуковые, лазерные, СВЧ.

Ультразвуковые имеют ряд недостатков: объект, до которого измеряется расстояние, должен быть достаточно крупным, чтобы от него отразился конусообразный пучок звуковых волн, высокая степень погрешности в звуконезащищенном помещении, дальность не более 35 метров. Однако они самые доступные на рынке

Лазерные тоже имеют ряд недостатков: на улице при ярком солнечном свете лазерный луч теряет яркость, что затрудняет работу прибора на больших расстояниях, при дождливой погоде дальномер также работает не вполне корректно из-за дождевых капель

В данной работе описывается модель СВЧ дальномера с однопроводной линией. Такой датчик лишен таких недостатков.

СВЧ дальномеры изобретены различными компаниями в разных странах. Однако они использовали в основном 8 и 16 битные МК, за редким исключением – 24 битные.

В данной работе используется 32-битный МК, что означает, что все операции, а так же обработка информации будет производиться быстрее остальных МК, и генератор, разработанный доцентом кафедры радиоэлектроники РФФ ТГУ Новиковым С.С., с однопроводной линией, которая обеспечивает наименьшие потери, относительно других способов распространения.

МК был выбран STM32F429I-DISCO. Данный МК на данный момент один из самых распространенных и доступных 32-битных МК на рынке электроники.

1 Система сбора и обработки информации. Обзор литературы

1.1 Система сбора и обработки информации

Цель работы – создать система сбора и обработки информации с датчика. Датчик представляет собой линию, на конце генератор, излучающий электромагнитную волну, которая распространяется вдоль проволоки. Далее волна отражается от препятствия, возвращается в генератор и меняет характеристики генератора на измерение. Детектируя сигнал, можно извлечь информацию о измерении расстояния от конца линии до препятствия. Для этого необходимо обработать полученную информацию и получить значение расстояния в готовом виде.

Генератором выступает генератор синусоидальных колебаний СВЧ диапазона, частотой которого можно управлять через варикап. Генератор собран Новиковым С.С.. Структурная схема генератора изображена на рисунке 1. Изначально генератор посылает волну одной частоты в определенном диапазоне.

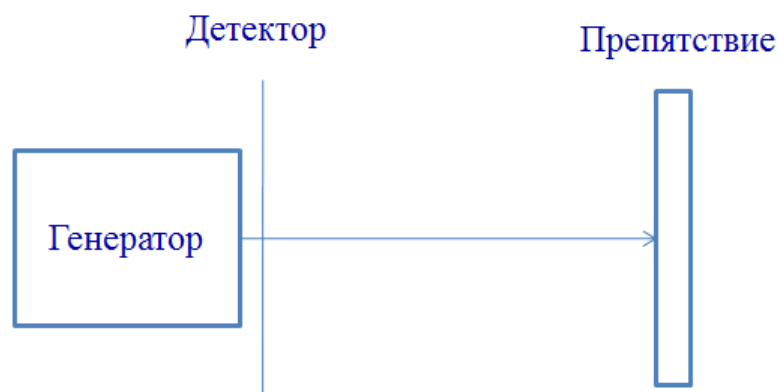


Рисунок 1 – Структурная схема генератора

Есть так же однопроводная линия, вдоль которой распространяется это колебание. В качестве линии выступает медный провод, вдоль которого распространяется электромагнитная волна до препятствия. Отразившись от препятствия, волна возвращается в генератор. В результате интерференции этих двух волн образуется стоячая волна. В поле этой стоячей волны помещён детектор, сигнал с которого пропорционален напряжению, которым в этой точке обладает эта волна. При перестройке частоты генератора картина всей стоячей волны изменяется и сигнал на детекторе будет соответствовать другому напряжению, которое будет у стоячей волны в этой точке. Частота будет меняться в заданном диапазоне. Структурная схема генератора изображена на рисунке 2. Т.к. картина стоячей волны изменялась, соответственно и детектор в процессе перестройки будет то в пучности то в узле. Значит, на выходе будет

периодический во времени сигнал, показанный на рисунке 2. Оценивая тем или иным способом количество периодов этого сигнала можно судить о расстоянии.

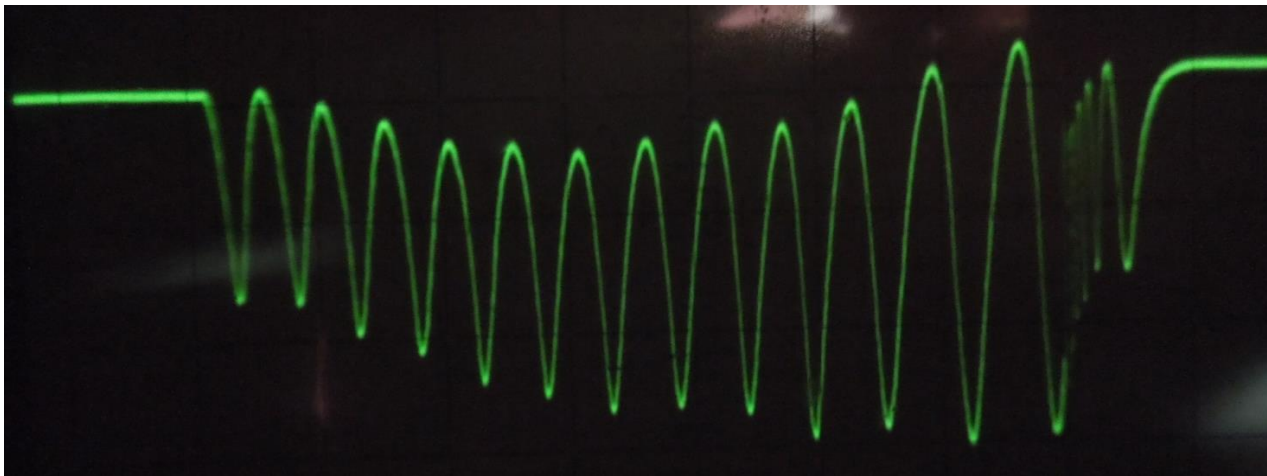


Рисунок 2 – Сигнал, получаемый генератором

Система сбора и обработки информации состоит из двух частей: система измерения полувольт стоячей волны (рисунок 3) и система генерирования пилообразного напряжения (рисунок 4).

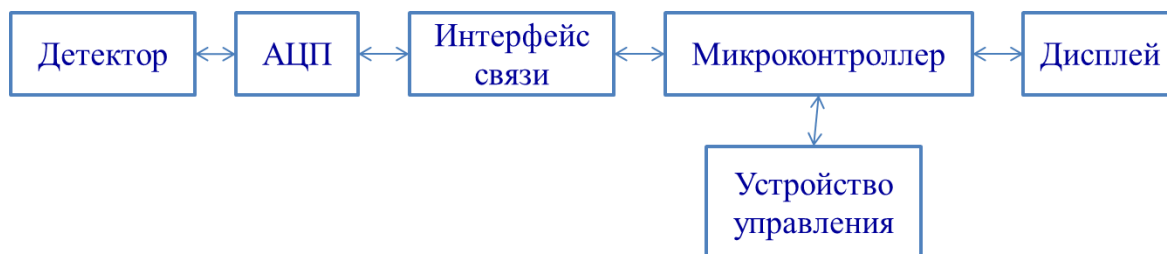


Рисунок 3 – Система измерения полувольт стоячей волны

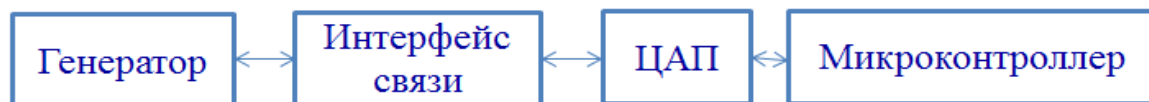


Рисунок 4 – Система генерирования пилообразного сигнала

Интерфейсом связи выступает отладочная плата STM32F429I-DISCO с МК STM32F429ZIT6U со встроенным дисплеем и кнопкой управления. МК необходим для обработки информации и вывода на дисплей информации в готовом виде.

1.2 Микроконтроллеры STM32

МК STM32 построены с использованием 32-разрядного ядра Cortex различных версий (в МК, установленном на плате используется ядро Cortex-M4). В таблице 1 приведены основные характеристики МК

Таблица 1 – Основные характеристики ядра МК STM32

Характеристика	Значение
Ширина слов для данных, разряд	32
Архитектура	Гарвард
Конвейер	3-ступенчатый
Набор инструкции	RISC
Организация памяти программ, разряд	32
Буфер предвыборки, разряд	2x64
Средний размер инструкции, байт	2
Тип прерываний	Векторизированные
Задержка реагирования на прерывания	12 циклов
Режимы управления энергопотреблением	Сон, сон по выходу, глубокий сон
Отладочный интерфейс	ST-LINK, JTAG

МК данного типа построены на гарвардской архитектуре и имеют 3-ступенчатый конвейер, который минимизирует время выполнения команд. Они разработаны для построения систем с максимальной энергоэффективностью и имеют несколько режимов управления энергопотреблением. В них используются внутренние интерфейсы памяти шириной больше, чем средняя длина инструкции. Это минимизирует число доступов к шине памяти, а, следовательно, и потребление электроэнергии, связанное с операциями по шине и чтением энергонезависимой памяти. Технология непрерывной обработки прерываний с исключением внутренних операций над стеком (tail chaining) сокращает время реакции на прерывания и исключает лишние операции со стеком.

1.3 Короткое описание платы STM32F429I-DISCO

Плата STM32F429I-DISCO (рисунок 5) предназначена для ознакомления с возможностями 32-битного МК на основе ARM-архитектуры, а также для реализации собственных устройств и приложений с использованием аппаратного обеспечения платы.

Плата STM32F429I-DISCO оснащена:

- а) МК STM32F429ZIT6 с ядром Cortex-M4F тактовой частотой 168 МГц, 2 Мб Flash-памяти, 256 кб RAM в корпусе LQFP144;
- б) отладчиком ST-Link/V2 для отладки и программирования МК;

- в) питанием платы через USB или от внешнего источника питания 3 В или 5 В;
- г) 2,4 – дюймовым QVGA TFT сенсорным дисплеем;
- д) двумя кнопками: синяя для программирования пользователем, черная для перезапуска.

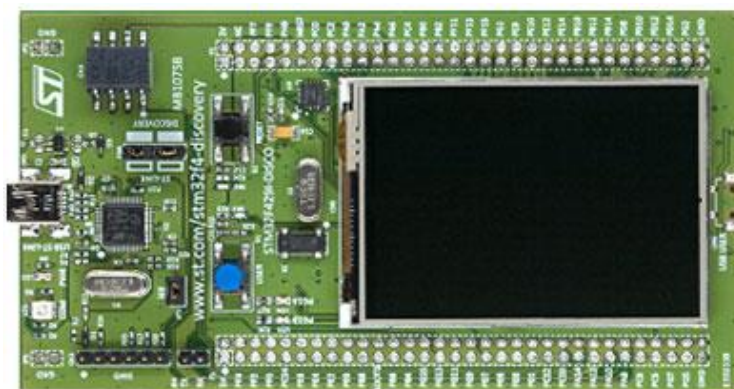


Рисунок 5 – Внешний вид платы STM32F429I-DISCO

Таким образом, отладочная плата оснащена большим количеством периферии, что позволяет сразу же реализовывать на ней примеры различной сложности.

1.4 Система ввода/вывода

МК содержит в себе семь 16-разрядных портов ввода/вывода общего назначения. Они обозначены как GPIOx, где x может принимать значения A, B, C, D, E, F, G. Каждый порт GPIO имеет восемь 32-битных регистров. Из них: четыре регистра конфигурации и два регистра данных.

В регистры конфигурации входят:

- а) GPIOx_MODER – регистр, который задает направление ввода/вывода каждой ножки. Направление может быть: вход, выход, аналоговый, альтернативной функции. По умолчанию направление всех ножек устанавливается на вход, за исключением портов A и B;

- б) GPIOx_TYPER – регистр, задающий тип выхода. Всего можно задать два типа вывода: двухтактный или открытый сток;

- в) GPIOx_SPEEDR – регистр, с помощью которого задается частота тактирования каждого вывода. Частота описывается как: низкая, средняя, повышенная и высокая;

- г) GPIOx_PUPDR – регистр, который управляет подтяжкой каждой ножки. Всего три вида подтяжек: без подтяжки, подтяжка к питанию, подтяжка к земле. По умолчанию подтяжки нет, кроме портов A и B;

Регистры данных представляют из себя настройки ножки на ввод и на вывод:

а) GPIOx_IDR – регистр ввода. Если вывод МК настроен на вход, то любой, поданный на ножку сигнал пройдет через триггер направления и отразится в этом регистре в виде соответствующего 0 или 1;

б) GPIOx_ODR – регистр вывода. В этом регистре то, что программист, запишет в этот регистр, то и попадет на ножки МК в виде логического уровня цифрового сигнала.

1.5 Вывод аналоговых сигналов. ЦАП

Цифро-аналоговый преобразователь – устройство, преобразующее цифровой код в аналоговый сигнал по величине, пропорциональной значению кода. В данном МК ЦАП взвешивающего типа. Схема ЦАП показана на рисунке 7.

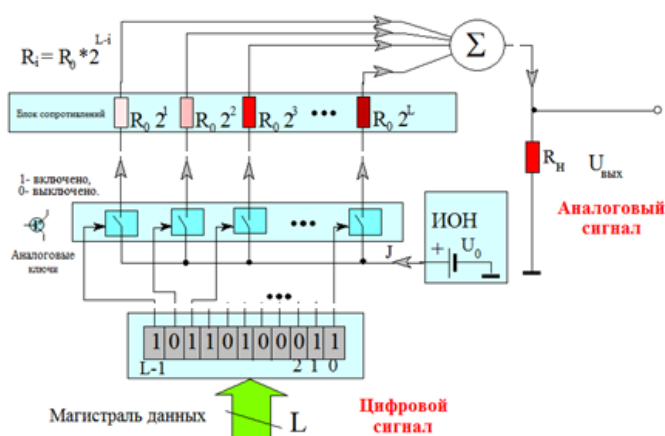


Рисунок 7 – Схема ЦАП

Некоторые характеристики и особенности:

- а) разрешение 12 бит;
- б) 2 канала вывода.

Принцип преобразования: поступает двоичное число. Каждая единица в двоичном коде числа включает сопротивление, а ноль выключает. Сопротивления имеют разные. Каждое значение тока пропорционально позиции символа в числе. Потом токи втекают в одну общую точку и, по первому закону Кирхгофа, суммируются, и получается общий суммарный ток, который идет через сопротивление и на нем создается падение напряжения, пропорциональное значению изначальной цифры.

1.6 Ввод аналоговых сигналов. АЦП

В МК STM32F429 есть мощный модуль АЦП. Схема АЦП приведена на рисунке 6.



Рисунок 6 – Схема АЦП

Некоторые характеристики и особенности:

- а) 18 каналов ввода: 16 внешних и 2 внутренних;
- б) разрешение 12 бит;
- в) генерирование всевозможных прерываний и сигналов для DMA;
- г) скорость оцифровки – до 0,9 MSPS с программируемым временем захвата и преобразования;
- д) автокалибровка.

Необходимость в этом модуле возникает часто, потому что природа вокруг нас не дискретна, а непрерывна, и всевозможные датчики обычно выдают именно аналоговый сигнал.

Принцип оцифровки: входное напряжение сравнивается с опорными напряжениями V_{REF-} и V_{REF+} : V_{REF-} подключается к земле, после чего V_{REF+} по желанию: либо к питанию процессора (оно плавающее и шумное, годится только для неточных измерений), либо к внешнему источнику опорного питания.

Однако имеется возможность программно настроить ноги V_{REF-} и V_{REF+} на прямое соединение с землёй и питанием.

Входное напряжение V_{in} будет измерено относительно V_{REF-} и V_{REF+} , и результат преобразования сложен в выходной регистр в такой пропорции:

$$V_{in}/(V_{REF} + (-V_{REF})) \times 4096$$

Например, 1,2 В при питании АЦП от 3,3 В преобразуются в 1490

1.7 Выбор среды разработки

Для разработки программного обеспечения под 32-разрядные процессоры ARM (в том числе и для рассматриваемой платы) рекомендуются следующие программные продукты:

- а) Atollic TrueSTUDIO;
- б) IAR Embedded Workbench;
- в) Keil μ Vision с инструментами MDK-ARM;
- г) Coocox CoIDE.

Рассмотрим каждый по отдельности.

Atollic TrueSTUDIO является коммерческим ПО и, следовательно, стоит денег. Однако разработчики дают возможность ознакомиться с их продуктом, представив бесплатную версию. Предложенный вариант среды разработки, TrueSTUDIO Lite, не имеет ограничений по времени пользования и размеру создаваемого кода. Демоверсия включает откомпилированные динамические библиотеки и допускает разработку полноценных приложений. Однако в ней отсутствует поддержка C++ и расширенного программного инструментария, нет возможности для создания коллективной работы, в частности интеграции с системами контроля версий и управления ошибками/задачами. Доступ к полным возможностям приложения можно получить путем покупки версий TrueSTUDIO Pro и TrueSTUDIO Premium.

IAR Embedded Workbench является коммерческим продуктом, его стоимость составляет около 3000 долларов за одну пользовательскую лицензию. Однако, в качестве дополнения к полнофункциональной версии, существует бесплатный вариант среды программирования с единственным ограничением на размер выходного кода до 4 или 8 КБ в зависимости от модели контроллера

Программа Keil μ Vision является платной и стоит очень больших денег. По ссылке ниже, после заполнения анкеты можно скачать демонстрационную версию. Разумеется, в ней присутствует ряд ограничений и основное из них – 32 Кб на размер программы.

Основной отличительной чертой всех перечисленных сред есть то, что они являются коммерческими. Бесплатное их использование возможно лишь с ограничениями по размеру бинарного кода программы, неполным набором функций или по срокам использования.

Наиболее известные бесплатные альтернативы – Coocox IDE и использование Eclipse вместе с плагином для разработки для 32-разрядных процессоров. Второй вариант требует множество дополнительных настроек и достаточно сложен для тех, кто только начинает разработку под МК. Среда разработки Coocox CoIDE 1.7 – бесплатный инструмент, который ориентирован на разработку для 32-разрядных процессоров ARM разных производителей: Atmel, Energy Micro, Freescale, Holtek, NXP, Nuvoton. Она построена на основе Eclipse, однако не имеет таких же широких возможностей для расширения. Несмотря на это, данная среда требует минимальных настроек для начала

программирования и отладки, поэтому представляется идеальным вариантом для начала работы с платой STM32F429I-DISCO. Для начала разработки, кроме самой CoIDE, потребуется установка средств для построения проекта GNU Toolchain for ARM Embedded Processors. Разработка с использованием CoIDE связана с концепцией репозитория: все доступные компоненты и библиотеки устанавливаются с помощью мастера, исходя из того, для какого именно процессора создан проект. Для загрузки библиотек необходимо подключение к сети Internet. Интерфейс Coosox IDE приведен на рисунке 8.

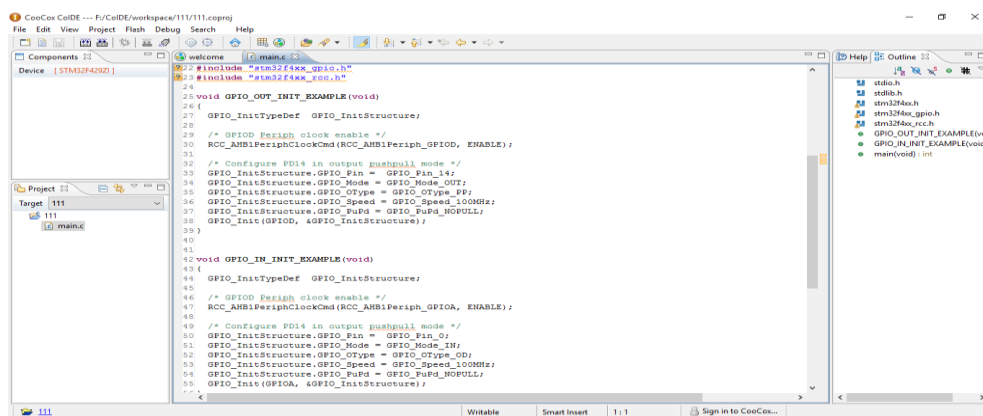


Рисунок 8 – Интерфейс программы Coosox IDE

Однако при дальнейшем использовании выясняется, что данная среда оказывается малоиспользуемой большинством программистов, из-за чего программировать МК становится проблематично, потому что очень мало наработанного материала в данной среде разработки.

Решением проблемы является STM32CubeMx. STM32CubeMx — новый программный продукт от компании STMicroelectronics, позволяющий при помощи понятного графического интерфейса произвести настройку любой имеющейся периферии МК.

Принцип работы в данной среде прост — создается проект, выбирается МК и пользователю сразу же предлагается большая схема со всеми выводами выбранного МК.

На рисунке 9 показан интерфейс программы.

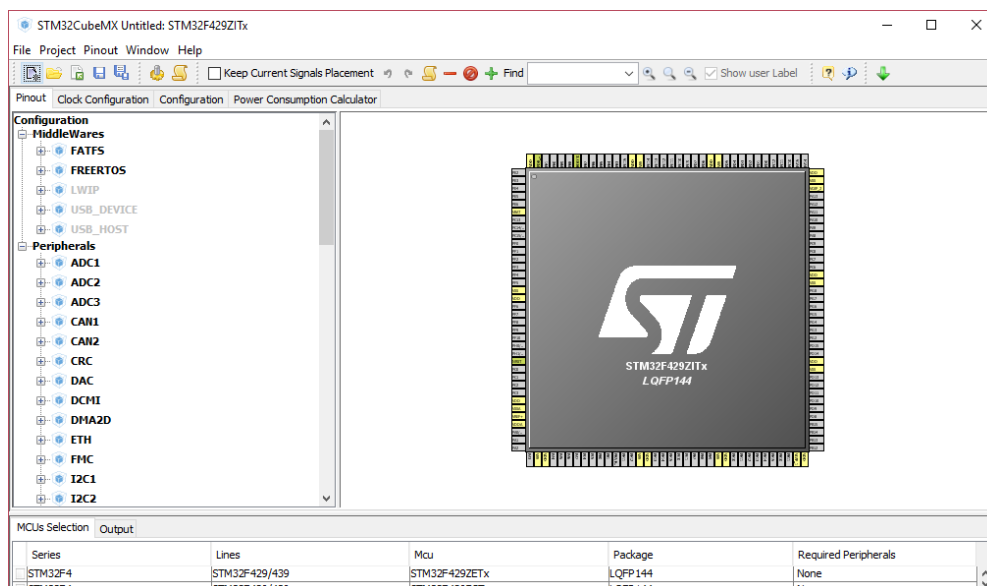


Рисунок 9 – Интерфейс программы STM32CubeMx

Нажимая на выводы и заходя в разнообразные меню, легко настраивается как периферия, так и режимы работы каждого конкретного вывода. Благодаря этому можно наглядно увидеть, какие выводы уже заняты, а какие еще свободны.

После выбора всех нужных выводов и настроек периферии (например, частоты тактирования, режимы работы таймеров...), и STM32CubeMx дает возможность сгенерировать соответствующий код. То есть в итоге получается готовый проект, с уже прописанными настройками, с подключенными библиотеками и с готовой инициализацией всей выбранной периферии. Базой для всего этого являются библиотеки CMSIS и HAL.

Библиотека STM32 HAL Driver пришла на смену хорошо известной, но устаревшей библиотеке Standard Peripheral Library (SPL). Структура и функции этих библиотек очень схожи, но HAL лучше работает по принципу «вызвать функцию и сразу заработало». Это позволяет минимизировать низкоуровневые настройки периферии.

Данная среда программирования позволяет генерировать код для дальнейшего использования для следующих сред программирования:

- а) Atollic TrueSTUDIO;
- б) IAR Embedded Workbench;
- в) Keil μ Vision с инструментами MDK-ARM;

Исходя из перечисленных выше описаний данных программ, остановимся на Keil μ Vision, т.к. она условно-бесплатна, а ограничение кода в 32 Кбайт не ограничивает нужное нам пространство памяти. Интерфейс программы показан на рисунке 10.

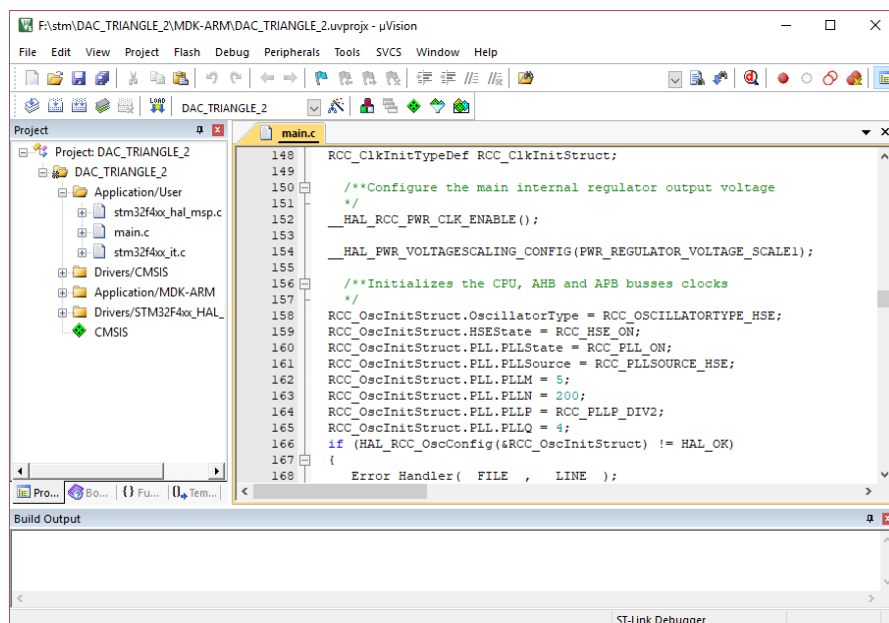


Рисунок 10 – Интерфейс программы Keil uVision

1.8 Программные средства для программирования платы

Несмотря на то, что среды разработки интегрируют в себе функциональность по перепрограммированию МК, полезно будет узнать, что существуют и отдельные программы, которые предназначены специально для работы с памятью (программирование, очистка, проверка) устройств. Кроме того, при работе с режимами пониженного энергопотребления именно эти программы позволяют выполнить программирование независимо от текущего режима работы, с чем могут возникнуть проблемы при выполнении данных действий из среды разработки. В данном разделе рассмотрим два программных продукта, которые реализуют функциональность для выполнения перепрограммирования без использования среды разработки: STM32 ST-LINK Utility и ST Visual Programmer. Программа STM32 ST-LINK Utility предназначена для работы с 32-разрядными контроллерами через интерфейс ST-LINK (в том числе и с его второй версией). Ее интерфейс представлен на рисунке 11. Он организован максимально просто и понятно. Основная рабочая область организована в виде двух вкладок, в которых отображается память устройства, а также представлен двоичный файл для записи. Наиболее важные команды сосредоточены в меню Target. Если в момент открытия программы устройство не было подключено к ПК, то выполнить подключение следует командой Target\Connect. После этого в панели сообщений должно появиться сообщение об успешном подключении и информация о подключенном устройстве. Выполнить программирование устройства можно командой Target\Program.... При этом

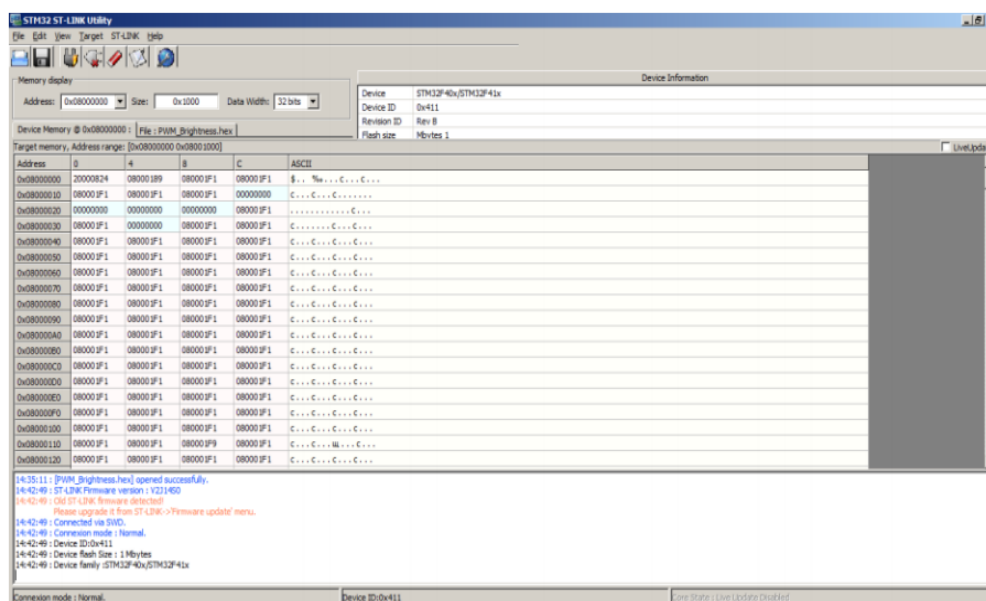


Рисунок 11 – Интерфейс программы STM32 ST-LINK

Таким образом, был изучен генератор изменяющейся частоты, разработана система управления и измерения датчиком, выбрано оптимальное программное обеспечение для программирования МК.

2 Алгоритм определения расстояния до объекта

2.1 Алгоритм поиска максимумов функции

Составим алгоритм на примере обыкновенного гармонического сигнала. Разделим сигнал на N равных интервалов (шагов) (рисунок 12).

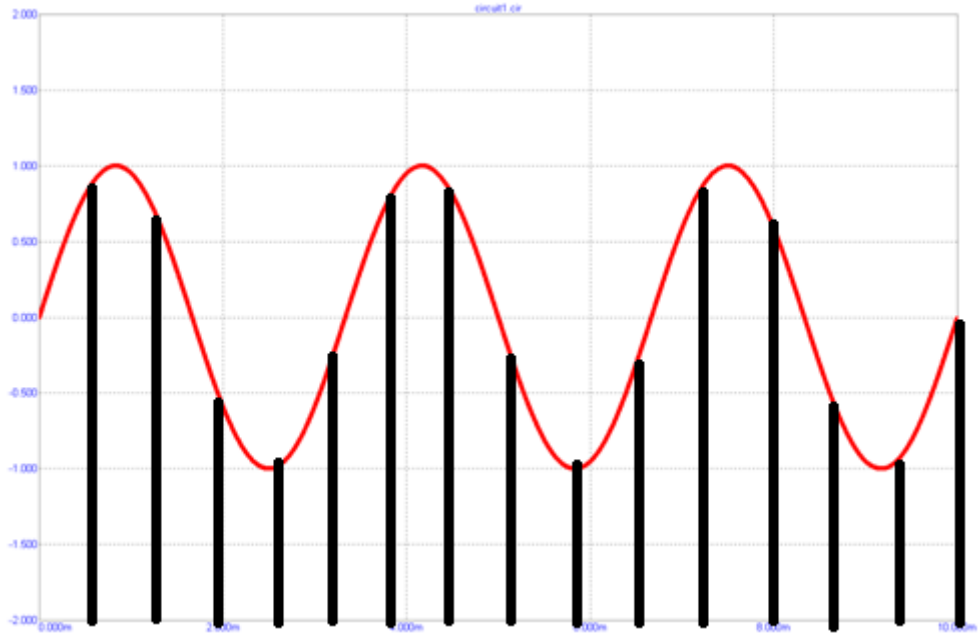


Рисунок 12 – Разбиение гармонического сигнала на шаги дискретизации

Шаг дискретизации возьмем равным единице. Введем счетчик элементов массива j , счетчик шагов i , число нулей функции k .

На каждом шаге дискретизации, записываем значение функции на данном шаге и запоминаем его в массиве $U[j]$. Этот массив нам будет необходим, для определения количества нулей функции.

После составления массива, необходимо найти значения, которое удовлетворяет следующему условию:

$$U[j-1] > U[j] < U[j+1] \quad (1)$$

Это условие показывает, что в интервале находится нуль функции. Необходимо посчитать, сколько будет таких интервалов.

Начнем рассматривать массив с самого начала, т.е. при $j=0$. Счетчик числа нулей приравняем к нулю $k=0$. Во время обработки массива, важно не выйти за пределы массива.

а) проверяем условие $j < N-2$. Если оно истинно переходим в пункт б), иначе переходим в пункт г);

б) проверяем условие (1). Если оно истинно, счетчик нулей увеличиваем на единицу и переходим в пункт в), иначе переходим в пункт в);

в) j увеличиваем на единицу, т.е. переходим к следующему значению массива, после чего переходим в пункт а);

г) вывод k . Конец алгоритма.

Блок-схема алгоритма представлена на рисунке 13.

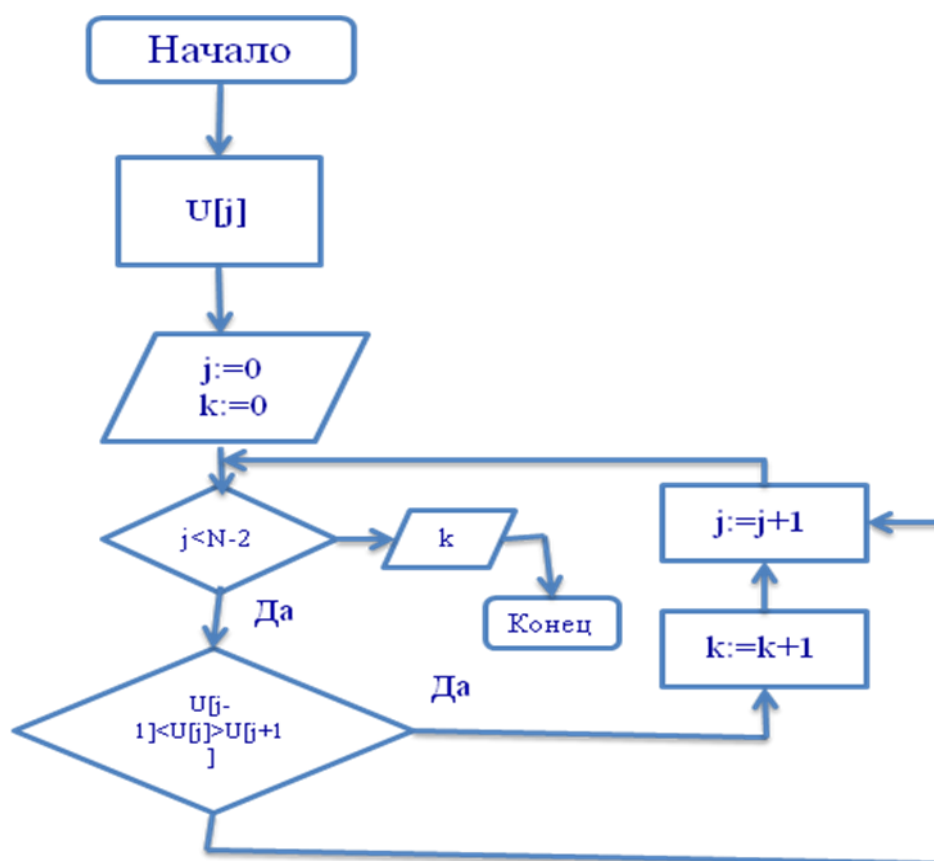


Рисунок 13 – Блок-схема алгоритма поиска нулей функции

Таким образом, разработан рабочий алгоритм поиска нулей функции.

2.2 Проверка алгоритма в среде Delphi

Было решено предварительно проверить данный алгоритм в среде Delphi следующим образом:

- 1) Сгенерировать несколько сигналов синусоидальной формы уменьшающихся по экспоненте;
- 2) Найти среднее значение в каждой точке сигнала, для нахождения средней точки;
- 3) Через все средние точки провести линию, для получения сглаженной кривой;

4) Посчитать число полувольт.

Скриншот программы Delphi, где смоделирован эксперимент, показан на рисунке 14. Суть его заключается в следующем: построить 20 функций, в каждый момент времени сложить значения функций, разделить на их количество для получения среднего, построить кривую по полученным точкам, найти максимумы построенной функции.

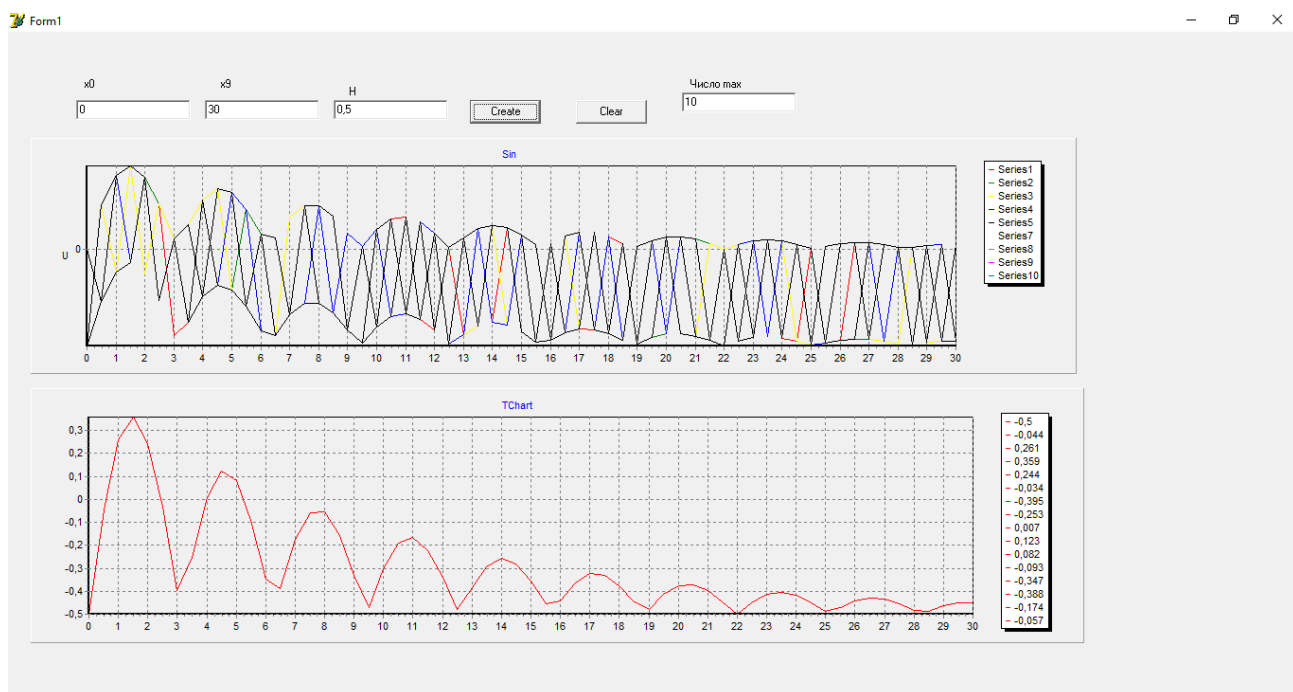


Рисунок 14 – Результат эксперимента проверки алгоритма по поиску числа полувольт

Как видим, алгоритм работает достаточно точно, поэтому можно этот алгоритм внедрить в МК.

3 Экспериментальная часть

3.1 Разработка программы генератора пилообразного напряжения

Далее разработал программу, которая позволила использовать в МК ЦАП как генератор пилообразного напряжения. Фрагмент программы, отвечающий за генерирование сигнала, представлен в приложении А.

С помощью этой программы, МК запрограммирован так, чтобы генератор был ЦАП, а ЦАП генерировал пилообразный сигнал и передавал его на СВЧ-генератор, чтобы это напряжение попадало на варикап генератора и заставляло качаться его частоту. Таким образом, ЦАП является источником напряжения для варикапа СВЧ-генератора. Сигнал генерируется из массива чисел. Последовательность чисел растет, а после максимального значения, сбрасывается до нуля. Данные числа, переводятся ЦАП МК в уровень выходного напряжения от 0 до 3,3 В. На рисунке 15, приведено изображение на осциллографе данного сигнала.



Рисунок 15 – Сигнал, подаваемый с ЦАП

Исходя из осциллограммы, можно убедиться, что устройство генерации пилообразного напряжения работает исправно.

3.2 Разработка устройства обработки сигнала с датчика

Для того, чтобы МК обрабатывал сигнал, необходимо настроить АЦП. Фрагмент программы приведен в приложении А.

Данный фрагмент программы позволяет АЦП обрабатывать сигнал с датчика, который принял отраженную волну. На данный момент реализована программа,

позволяющая оцифровать сигнал с амплитудой до 3,3 В, поиском количества максимумов и выводом значения на дисплей (см. рисунок 16).

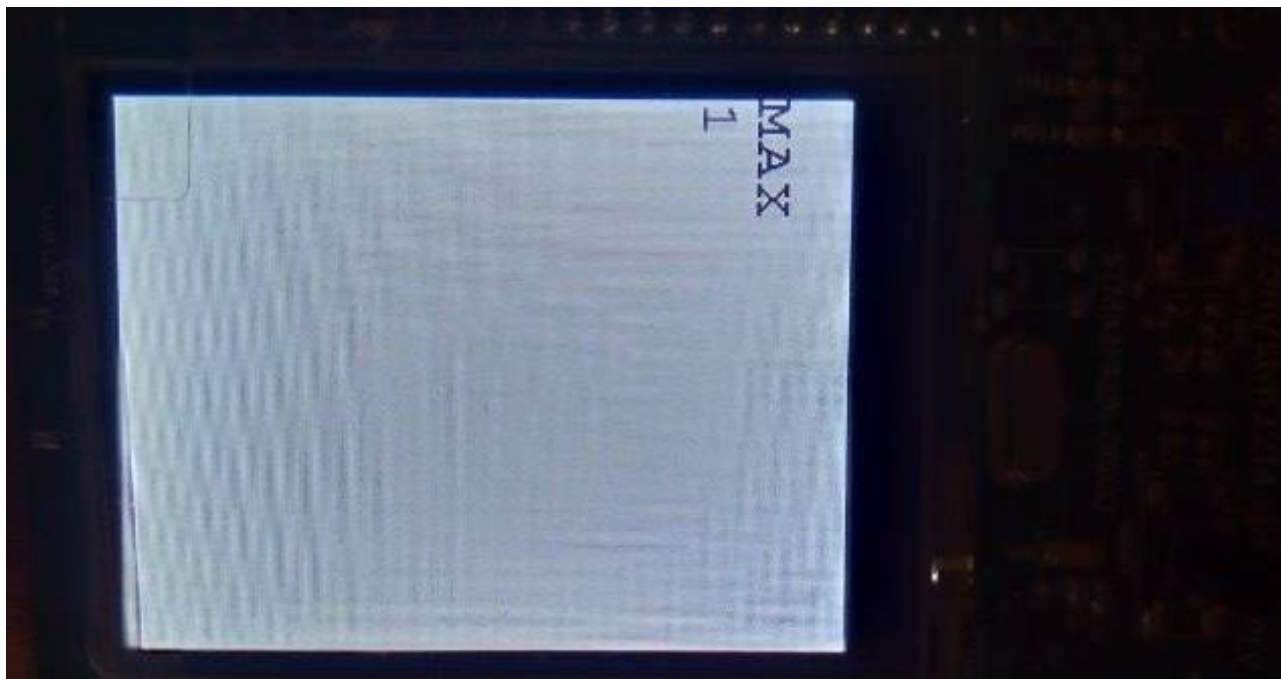


Рисунок 16 – Вывод числа максимумов на дисплей

Однако в дальнейшем появилась проблема: сигнал, который принимает датчик, содержит шумы. Из-за этого программа по поиску максимумов работает некорректно, поэтому необходимо модернизировать программу.

При разработке программы было решено провести следующий эксперимент: задать гармоническую функцию с определенным количеством периодов и добавлением случайного числа, чтобы было возможно достоверно подсчитать количество полувольт при наличии шума.

Смоделированный сигнал описывается следующей функцией

$$v = 1500 \times (\sin(0,98 \times m) + 1,0 + v1), \quad (2)$$

где v – значение амплитуды сигнала, m – число, принимающее значение от 0 до 127, $v1$ – случайное число, принимающее значение от -10 до 10.

Далее построим данную функцию 2000 раз, после чего запустим процесс обработки. Складываем все значения в каждый момент времени и делим на 2000. Получаем отличный результат обработки, показанный на рисунке 17. Т.к. значения постоянно растут, это говорит о том, что процедура сглаживания линии прошла успешно. Однако, следует отметить, что данный результат МК вывел на дисплей в течение десяти секунд, что является очень большим промежутком времени. Было решено сократить число

построений до 1000 раз. Как видно, и в этом случае получился отличный результат, но время быстрогодействия МК было сокращено только до пяти секунд, что так же было очень большим промежутком времени. В итоге принято решение остановиться на 50, т.к. это было минимальное количество снятия кривой, при котором результат оказывался удовлетворительным.

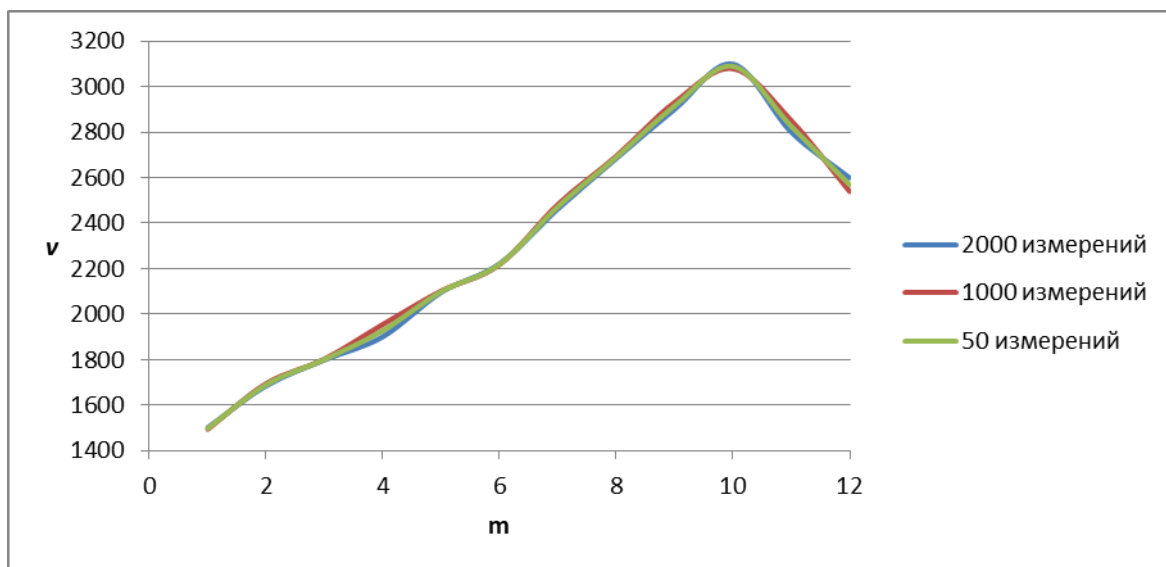


Рисунок 17 – Результаты поиска оптимального количества измерений

В итоге было найдено оптимально количество снятий сигнала с датчика, однако возникла новая проблема: АЦП не позволял работать ЦАП. Соответственно ЦАП было необходимо заменить другим периферийным устройством. В связи с этим, решено было использовать ШИМ, с помощью которого можно организовать перестройку частоты генератора. В отличие от ЦАП, ШИМ не конфликтует с АЦП. Осциллограмма ШИМ показана на рисунке 18.

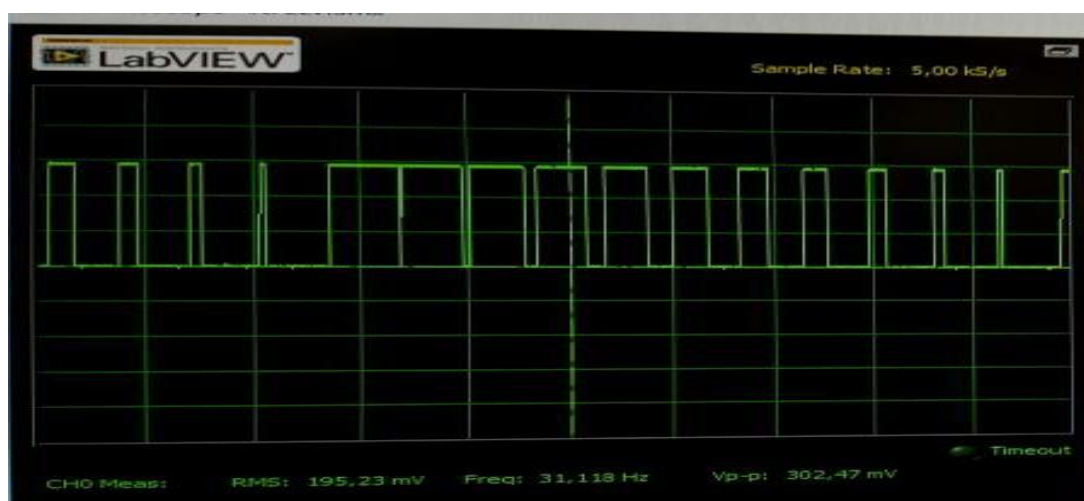


Рисунок 18 – Осциллограмма ШИМ

Из рисунка 18 следует, что скважность импульсов уменьшается. Это означает, что частота генератора будет перестраиваться с высокой частоты на низкую частоту, что не повлияет на работу прибора при измерении расстояния. Таким образом, устройство по сбору и обработке информации сможет выполнить поставленную задачу.

3.3 Операционный усилитель

Частотный диапазон генератора в устройстве измерения расстояния составляет 3200 – 3800 МГц. Элементом, задающим частоту генератора, является варикап. Для того, чтобы обеспечить перестройку частоты генератора в его частотном диапазоне не варикап подается напряжение равное 1 – 20 В.

Так как МК может вырабатывать напряжение максимальной амплитуды 3,3 В, то для того, чтобы перестраивать частоту генератора, сигнал необходимо усилить с 3,3 до 20 В, для чего необходимо было сконструировать усилитель.

Было решено сконструировать усилитель на основе ОУ LM32N. Напряжение внешнего источника было выбрано 24 В, амплитуду входного источника усиливаемого напряжения было решено выбрать в 3,3 В, так как это позволяет снизить ступенчатость усиленного сигнала, который подавался на варикап СВЧ-генератора. В среде Multisim было проведено моделирование усилителя (рисунок 19), в качестве аналога LM324N в среде схемотехнического моделирования выступил операционный усилитель 3288RT. С помощью виртуального осциллографа были сняты временные диаграммы (рисунок 2), где сигнал 3,3 В – синяя линия, сигнал 20 В – красная линия.

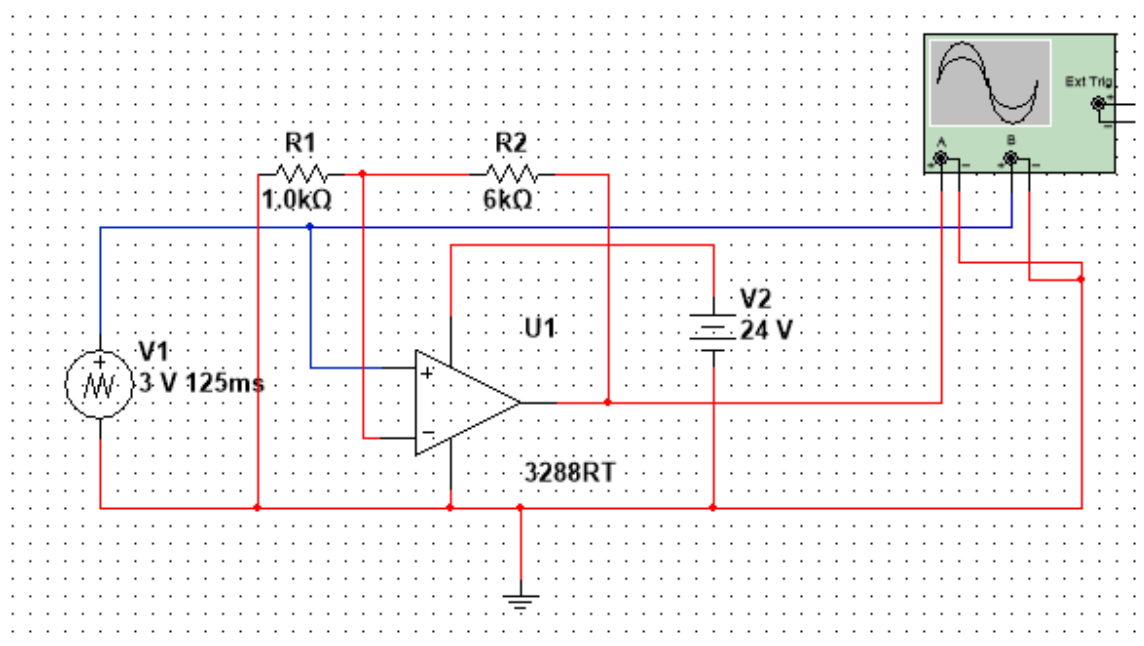


Рисунок 19 – Усилитель на основе ОУ в среде Multisim

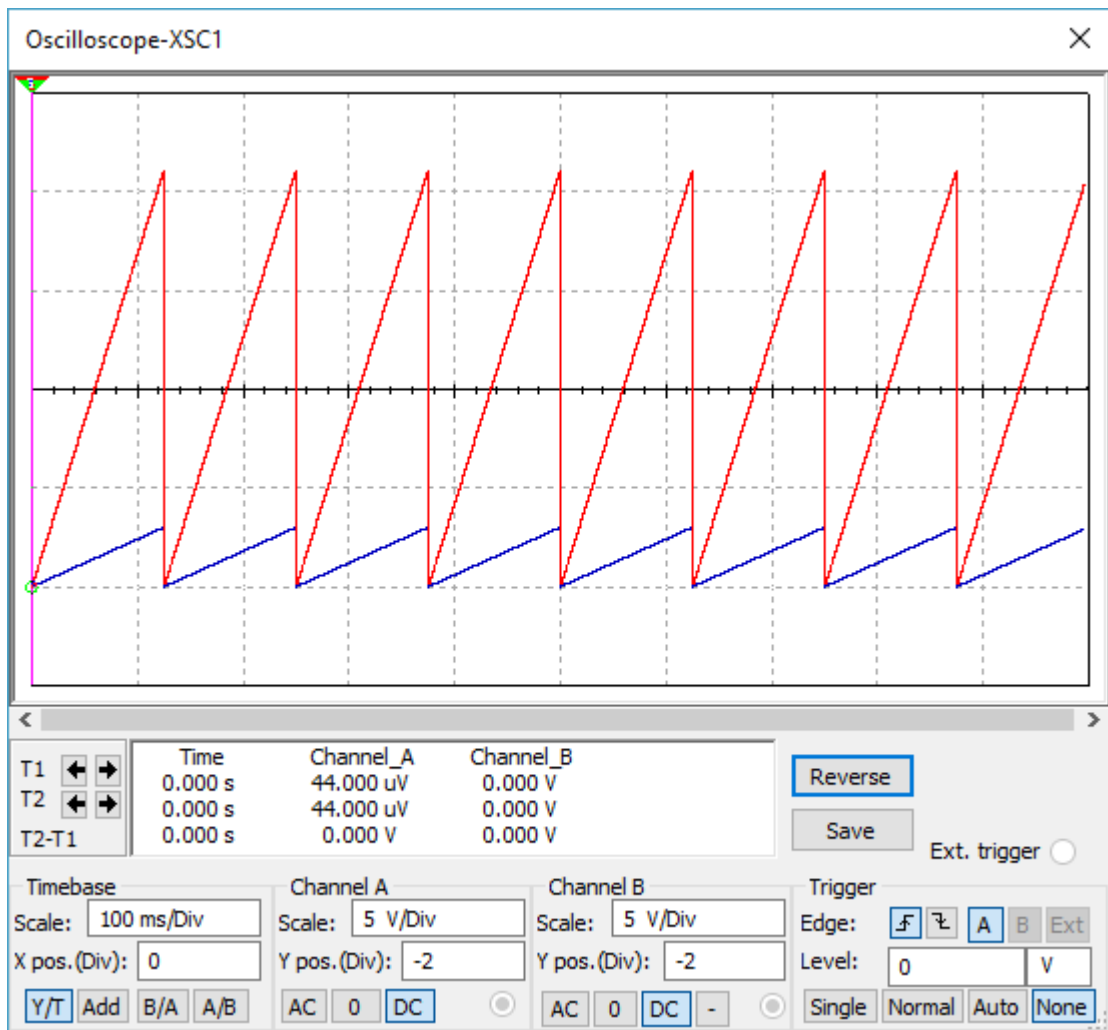


Рисунок 20 – Временная диаграмма с усилителя на основе ОУ

Таким образом был разработан усилитель.

3.4 Устранение помех

На сегодняшний день проблема влияния высокочастотных помех на радиоэлектронные приборы до сих пор актуальна. Для того чтобы избавиться от высокочастотных помех, практически в каждом радиоэлектронном устройстве устанавливается ФНЧ. Для сглаживания сигнала с детектора в работе использовался RC-фильтр низкой частоты. Информационный сигнал лежит в полосе частот: 0 – 2500 Гц. Это означает, что частота среза фильтра должна равняться $f_c = 2,5$ кГц. Так как сопротивление нагрузки в схеме равно 5 кОм, то сопротивление резистора $R1$ выбираем порядка 100 Ом. Для случая ФНЧ, реактивное сопротивление конденсатора $X_C = R2$. Сопротивление второго резистора находим по формуле делителя (2):

$$R2 = \frac{U_{\text{вых}} \times R_{\text{общ}}}{U_{\text{вх}}} = \frac{U_{\text{вых}} \times (R1 + R2)}{U_{\text{вх}}} = \frac{0,07 \times (100 + R2)}{0,1} = 233 \text{ Ом} \quad (2)$$

Известно, что емкость конденсатора можно вычислить по следующей формуле:

$$C = \frac{1}{2\pi \times f_c \times R2} = \frac{1}{6,28 \times 2500 \times 233} = 273 \text{ нФ} \quad (3)$$

Основываясь на расчетах, в программе Multisim был смоделирован необходимый ФНЧ (рисунок 21). На рисунке 22 изображены временные диаграммы двух сигналов: синяя линия – сигнал с выхода генератора, красная линия – сигнал после ФНЧ.

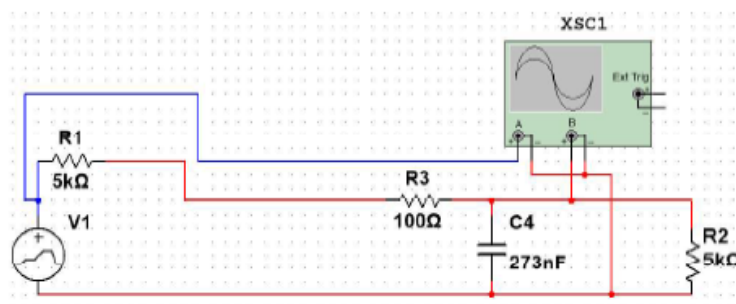


Рисунок 21 – ФНЧ в среде Multisim

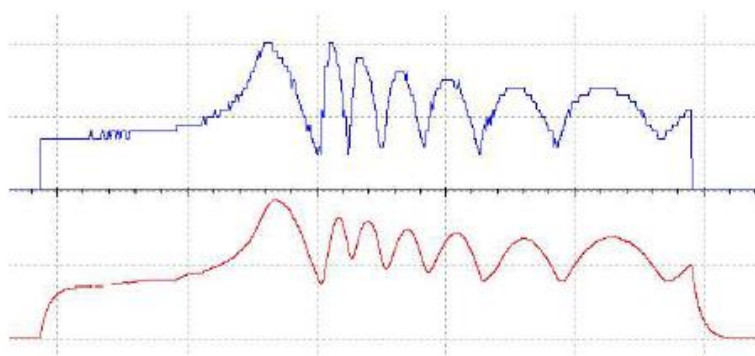


Рисунок 22 – Временная диаграмма с ФНЧ

Затем, на основании моделирования усилителя в подразделе 3.3 и расчетов фильтра была разработана принципиальная электрическая схема усилителя и ФНЧ (рисунок 23)

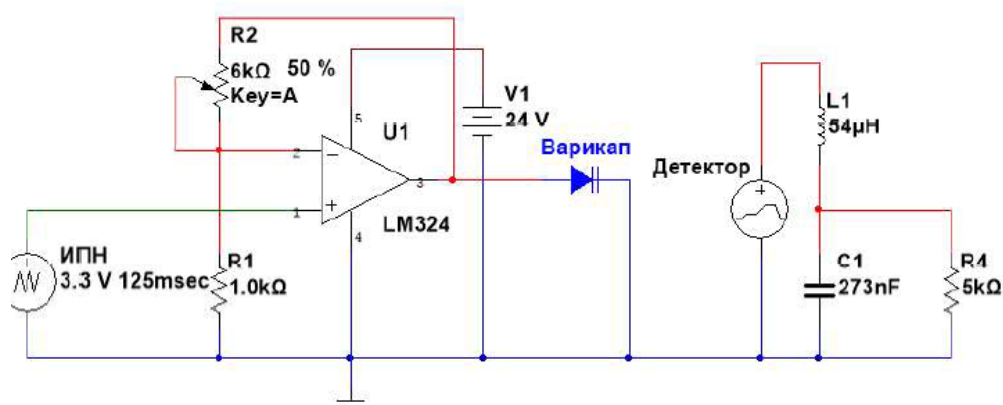


Рисунок 23 – Принципиальная электрическая схема ФНЧ и ОУ

С помощью потенциометра R2, задается диапазон выходного напряжения ОУ. Он был выбран от 0 до 20 В. Сопротивление резистора при этом составило порядка 6 кОм.

На основе разработанной схемы было собрано устройство, показанное на рисунке 24:

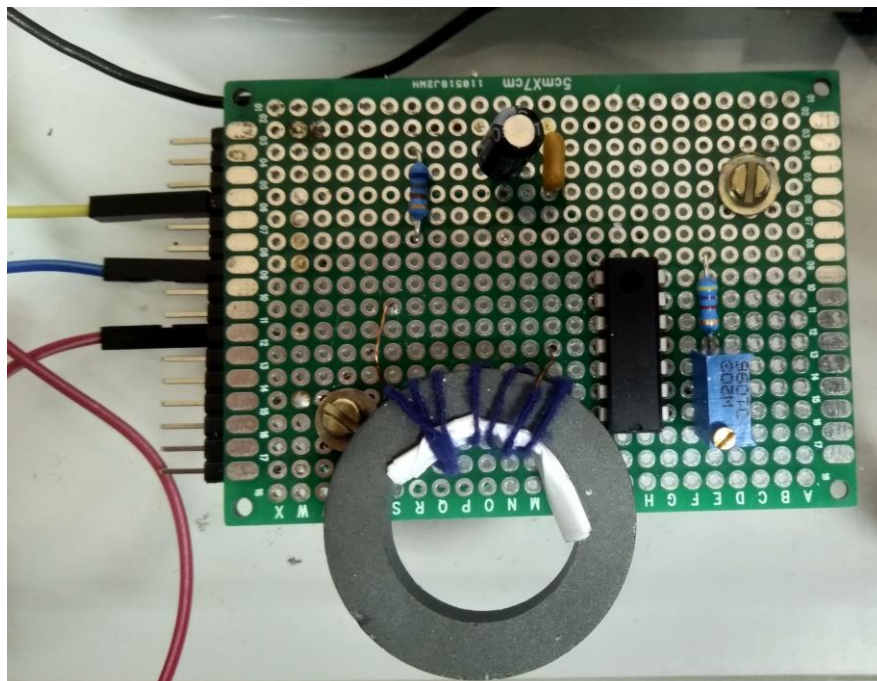


Рисунок 24 – Внешний вид устройства

В результате работы удалось создать компактное устройство, выполняющее роль согласования усилителя и варикапа СВЧ датчика, а также реализовать на этой же схеме ФНЧ, для уменьшения влияния помех на исследуемый сигнал с детектора.

3.5 Разработка корпуса устройства

Все компоненты устройства сконструированы и проведены испытания, после чего было начато изготовление корпуса. Корпус было решено сделать из органического стекла, так как это позволяет изучать устройство, не разбирая его. Корпус будет состоять из двух пластин органического стекла. На одной пластине будут располагаться отладочная плата с МК, разъемы и ОУ с ФНЧ. Отладочная палата крепится к пластине посредством специальных креплений, позволяющих без усилий извлекать плату и перепрограммировать ее или заменить. ОУ с ФНЧ прикрепляются посредством винтов, разъемы приклеиваются на термоклей. Верхняя крышка пластина устройства имеет отверстие для кнопки RESET. Чертеж корпуса приведен в приложении Б.

На рисунке 25 изображен внешний вид устройства со всеми его компонентами внутри.

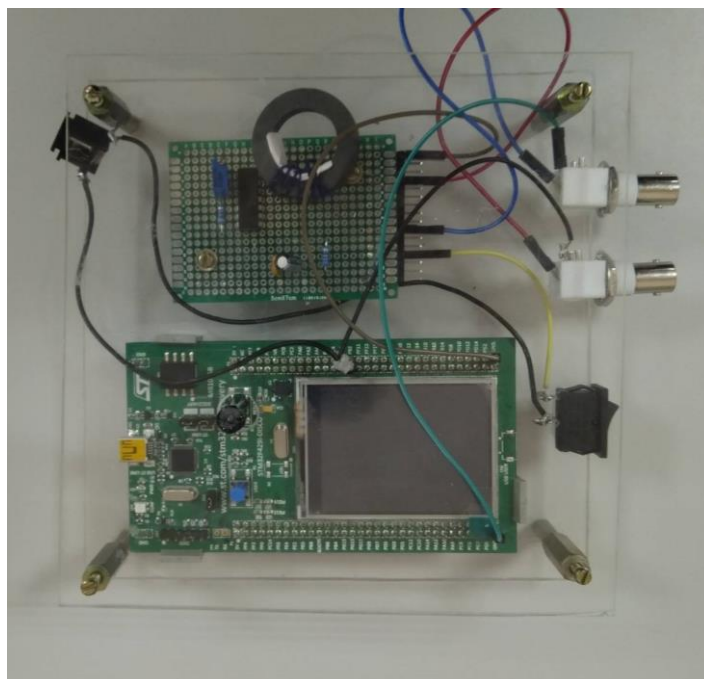


Рисунок 25 – Внешний вид устройства управления СВЧ датчиком

В итоге удалось разработать эргономичный корпус для разрабатываемого устройства. Габаритные размеры устройства составили: 150×150×50 мм. В ближайшее время планируется завершить финальные испытания с конечным вариантом устройства, внести корректировки и написать отчет.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы были получены следующие основные результаты:

- 1) было выбрано оптимальное программное обеспечение для программирования МК;
- 2) освоены базовые навыки программирования МК;
- 3) освоены режимы работы портов ввода/вывода, принцип работы АЦП и ЦАП;
- 3) составлена блок-схема алгоритма поиска нулей;
- 4) разработано устройство, позволяющее генерировать пилообразный сигнал и обрабатывать сигнал с датчика;
- 5) разработаны и собраны ОУ и ФНЧ;
- 6) разработан и изготовлен корпус для устройства.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ARM Limited. Procedure call standard for the ARM® architecture, November 2015. – URL: http://infocenter.arm.com/help/topic/com.arm.doc.ih0042f/IHI0042F_aapcs.pdf (Дата обращения: 26.09.2018).
2. STMicroelectronics. How to get the best ADC accuracy in the STM32F10xxx devices, November 2008. – URL: <http://www.bdtic.com/Download/ST/AN2834.pdf> (Дата обращения: 27.10.2018).
3. STMicroelectronics. User manual STM32 value line discovery, 2010. – URL: http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/CD00267113.pdf (Дата обращения: 20.09.2018).
4. STM8 и STM32 – объединенное пространство 8- и 32-разрядных микроконтроллеров / Бородулин А. / Компоненты и технологии №10. – 2009. – с. 55-59. – URL: http://ecworld.ru/media/bip/pdfs/borodulin_ct1009.pdf (Дата обращения: 09.10.2018).
5. STM32F4DISCOVERY, Отладочный комплект на базе STM32F407VGT6 ARM CortexM4-F [Электронный ресурс] – Режим доступа : URL: <http://www.chipdip.ru/product/stm32f4discovery/>. – Загол. с экрана. (Дата обращения: 14.10.2018).
6. Программирование STM32F4. USART. Пример программы. [Электронный ресурс] – Режим доступа: URL: <http://microtechnics.ru/programmirovanie-stm32f4-uart-primer-programmy/>. – Загол. с экрана. (Дата обращения: 17.10.2018).
7. Микроконтроллеры AVR. UART. Использование прерываний. [Электронный ресурс] – Режим доступа : URL: <http://microtechnics.ru/mikrokontrollery-avr-uart-ispolzovanie-preryvanij/>. – Загол. с экрана. (Дата обращения: 21.10.2018).
8. STM 32 и ввод аналоговых сигналов АЦП/ADC. - [Электронный ресурс] – Режим доступа: URL: [http:// catethysis.ru/stm32-adc/](http://catethysis.ru/stm32-adc/). – Загол. с экрана. (Дата обращения: 10.11.2018).
9. Atollic TrueSTUDIO. - [Электронный ресурс] – Режим доступа: URL: http://cxem.net/software/atollic_truestudio.php/. – Загол. с экрана. (Дата обращения: 18.09.2018).
10. Keil uVision. - [Электронный ресурс] – Режим доступа: URL: <http://cxem.net/software/keil.php/>. – Загол. с экрана. (Дата обращения: 18.09.2018).

11. IAR Embedded Workbench. - [Электронный ресурс] – Режим доступа: URL: <http://cxem.net/software/iar.php/>. – Загол. с экрана. (Дата обращения: 18.03.2016).
12. Coocox CoIDE. - [Электронный ресурс] – Режим доступа: URL: <http://cxem.net/software/coide.php/>. – Загол. с экрана. (Дата обращения: 18.03.2016).

ПРИЛОЖЕНИЕ А

(Справочное)

Фрагмент программы обработки сигнала с датчика и генерации пилообразного сигнала

```
for (n=0;n<=3;n++)
{
    for (m=0;m<=127;m++)
    {
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 100);
        x[n][m]=(uint16_t) HAL_ADC_GetValue(&hadc1);*/

        /*    sprintf(st, "%i", x[1][m]);
                BSP_LCD_DisplayStringAtLine(i+1,    (uint8_t*)st);        i++;
HAL_ADC_Stop(&hadc1);*/*    }    }*/

/*    for (n=0;n<=9;n++)
{
    for (m=0;m<=127;m++)
    {
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 100);
        double y=(uint16_t) HAL_ADC_GetValue(&hadc1);
        x[m]=x[m]+y;
    }*/

    for (m=0;m<=127;m++)
    {
        x[m]=0;

    }

    p=0;

    for (n=0;n<=49;n++)
    {
        for (m=0;m<=127;m++)
        {
```

```

        p=m*512;
//    HAL_TIM_PWM_Start(&htim3,TIM_CHANNEL_1);

    HAL_TIM_PWM_Start_IT(&htim3,TIM_CHANNEL_1);

    TIM3->CCR1=p;

    for(d=0;d<300;d++)
    {

    }

    /*    double v1=0.1*(-10+rand()%21);
        v=1500*(sin(0.98*m)+1.0+v1);*/

        /*x[m]=x[m]+v;*/    }
//HAL_TIM_PWM_Stop(&htim10,TIM_CHANNEL_1);

/* for (n=0;n<=127;n++) {x[n]=x[n]/50;}

i=1;

    for (m=0;m<125;m++)
    {
    if (x[m] < x[m+1] & x[m+1] > x[m+2])
        u=u+1;
    }

    sprintf(str, "%i", u);
    BSP_LCD_DisplayStringAtLine(5, (uint8_t*)str);*/

}

```

ПРИЛОЖЕНИЕ Б

(Справочное)

Схема электрическая принципиальная ОУ и ФНЧ

					Схема электрическая принципиальная						
Изм.	Лист	№ докум.	Подпись	Дата	Операционный усилитель с фильтром нижних частот	Лит.			Масса	Масштаб	
Разраб.		Ни В.В.							1	1 : 1	
Провер.											
Т. Контр.						Лис 1			Листов	14	
Реценз.						33					РФФ 738
Н. Контр.											
Утверд.											

Введите текст:

...или загрузите файл:

Файл не выбран...

Выбрать файл...

Укажите год публикации:

2019

Выберите коллекции

- | | | |
|---|--|--|
| ☒ Все | ☒ Википедия | ☒ Российские журналы |
| ☒ Рефераты | ☒ Российские конференции | ☒ Энциклопедии |
| ☒ Авторефераты | ☒ Иностранные журналы | ☒ Англоязычная википедия |
| ☒ Иностранные конференции | | |
| ☒ PubMed | | |

Анализировать

[Проверить по расширенному списку коллекций системы Руконтекст](#)Обработан файл:
Диплом. Альфа.doc.

Год публикации: 2019.

Оценка оригинальности документа - 100.0%

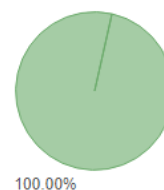
Процент условно корректных заимствований - 0.0%

Процент некорректных заимствований - 0.0%

[Просмотр заимствований в документе](#)

Время выполнения: 20 с.

Заимствования отсутствуют

[Дополнительно](#)