

Федеральное государственное автономное образовательное учреждение высшего образования

«Южно-Уральский государственный университет»  
(национальный исследовательский университет)  
Высшая школа электроники и компьютерных наук  
Кафедра «Информационно-измерительная техника»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой, д.т.н., проф.  
\_\_\_\_\_/А.Л. Шестаков/  
«\_\_»\_\_\_\_\_2018 г.

Разработка программного обеспечения для датчиков физических величин

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ–12.03.01.2018.114-288.ВКР

Руководитель, доц. каф. ИНИТ  
\_\_\_\_\_/С.В. Колодий/  
«\_\_»\_\_\_\_\_2018 г.

Автор  
Студент группы КЭ-476  
\_\_\_\_\_/А.А. Сорокин/  
«\_\_»\_\_\_\_\_2018 г.

Нормоконтролёр, доц. каф. ИНИТ  
\_\_\_\_\_/А.С. Волосников/  
«\_\_»\_\_\_\_\_2018 г.

## АННОТАЦИЯ

Сорокин            А.А.            Разработка  
программного обеспечения датчиков  
физических величин – Челябинск:  
ЮУрГУ, КЭ-476; 2018; 75 с., 45 ил.,  
библиогр. список – 14 наим., 1 прил..

Роль датчиков в современной промышленности, как и в бытовой сфере в наши дни сложно переоценить. Для улучшения качества жизни в направлении нынешнего развития требуется собирать и обрабатывать огромные массивы информации о состоянии окружающей среды и физических объектов. Эту информацию получают с датчиков, поэтому необходимо уметь её извлекать.

В выпускной квалификационной работе были выполнены следующие задачи:

1. Изучение принципов действия датчика Холла и аналитический обзор.
2. Теоретический расчёт пределов измерения датчика.
3. Разработка структуры программы и программного обеспечения для связи датчика с отладочной платой.
4. Нахождение функции преобразования опытным путём.

Подп. и дата								
Взам. инв.								
Инв. №								
Подп. и дата								
Инв. №					<b>ЮУрГУ 12.03.01.2018.288 ВКР</b>			
Инв. №					Разработка программного обеспечения для датчиков физических величин	Лит	Лист	Листов
Разраб.							3	
Пров.						<b>ЮУрГУ</b>		
Т. контр.						<b>Кафедра ИНИТ</b>		
Н. контр.								
Утв.								

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	6
АНАЛИТИЧЕСКИЙ ОБЗОР.....	8
1.1 Датчик Холла .....	8
1.1.1 Понятие термина датчик. Его классификация.....	8
1.1.2 Эффект Холла и его применение в датчиках.....	11
1.1.3 Перспективные разработки .....	14
ЭКСПЕРИМЕНТАЛЬНОЕ ОПРЕДЕЛЕНИЕ ПРЕДЕЛОВ ИЗМЕРЕНИЯ ДАТЧИКА ХОЛЛА .....	26
2.1 Суть эксперимента с датчиком Холла .....	26
2.2 Описание элементов устройства .....	28
2.3 Теоретический расчёт пределов измерения .....	35
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДАТЧИКА .....	38
3.1 Общие сведения .....	38
3.2 Функциональное назначение .....	39
3.3 Требования к архитектуре программы .....	39
3.4 Общая структура программы .....	45
3.5 Архитектура класса ADC .....	46
3.6 Архитектура класса UART .....	47
3.7 Архитектура класса Button .....	47
3.8 Архитектура класса ButtonController .....	48
3.9 Архитектура класса CalculationCurrent.....	48
3.10 Архитектура класса UARTController .....	50
3.11 Описание логической структуры .....	51

ЭКСПЕРИМЕНТАЛЬНОЕ ОПРЕДЕЛЕНИЕ ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ДАТЧИКА .....	54
4.1    Определение функции преобразования для постоянного тока .....	54
ЗАКЛЮЧЕНИЕ .....	61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	62
ПРИЛОЖЕНИЕ А.....	64

## ВВЕДЕНИЕ

Самыми первыми датчиками, дающими человеку представление об окружающем мире, были и остаются его органы чувств. По мере изучения человеком природы окружающих его вещей он всё дальше углублялся в самую суть их происхождения, и на определённом этапе стала очевидной ограниченность его органов восприятия. Так появились первые датчики, с помощью которых человек смог получать новые знания об окружающем мире.

Современные датчики - это весьма сложные устройства, которые могут получать информацию одновременно о нескольких величинах, а также производить её обработку и воспроизводить в удобной для человека форме. Основными требованиями к датчикам являются: простота их конструкции, характеризующая надёжность датчика и ремонтпригодность, низкая стоимость производства, а также точность и своевременность информации о физической величине, которую он измеряет.

Тенденцией нынешнего развития информационно-измерительной сферы является построение сети для сбора информации об объекте на основе большого числа всевозможных датчиков, отправляющих информацию на контроллер, который обычно её обрабатывает и хранит, а также выводит в нужное время. Все эти датчики, при этом, могут иметь различные интерфейсы передачи данных, которые необходимо уметь настраивать для правильного функционирования сети.

Для того чтобы научиться взаимодействовать с различными датчиками, изучить принципы их работы и сферы применения ведущие производители электроники начали выпускать комплекты, включающие в себя отладочную плату, основой которой является микропроцессор, либо микроконтроллер, а также набор из различных датчиков и устройств ввода/вывода.

Целью данной работы является разработка лабораторной работы, которая в дальнейшем будет включена в методическое пособие по курсу «Программное обеспечение измерительных процессов». Для этого будут решены следующие задачи:

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- 1) Изучение принципов действия датчика Холла и аналитический обзор;
- 2) Теоретический расчёт пределов измерения датчика;
- 3) Разработка структуры программы и программного обеспечения для связи датчика с отладочной платой;
- 4) Нахождение функции преобразования опытным путём.

Лабораторная работа будет строиться вокруг определённого датчика Холла и задачей будет получить с его помощью информацию о величине тока. В работе рассматривается датчик Холла. Содержится необходимая теоретическая информация для понимания принципа его работы и физического эффекта, положенного в его основу. Такой датчик представляет собой микросхему из полупроводниковых элементов, размещённых на печатной плате. У него есть три контакта: питание, заземление и данные. Для получения информации с этого датчика будет использована отладочная плата XNUCLEO-F411RE, построенная на базе микроконтроллера STM32-F411ret6. Необходимо разработать структуру программы для получения данных с датчика и вывода её на монитор компьютера по интерфейсу USART. Написание непосредственно программы осуществляется на языке C++ в программе IAR Embedded Workbench. Заключительной частью работы станет определение функции преобразования датчика. Например, датчик Холла можно использовать для бесконтактного измерения переменного тока в сетях. В соответствии с этим будет собрана измерительная цепь и проведёт эксперимент. Полученная зависимость тока в проводнике от кода, получаемого с АЦП, к которому подведён информационный вывод датчика, будет использована в программе к этому датчику.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

# АНАЛИТИЧЕСКИЙ ОБЗОР

## 1.1 Датчик Холла

### 1.1.1 Понятие термина датчик. Его классификация

Стремительное развитие электроники и вычислительной техники оказалось предпосылкой для широкой автоматизации самых разнообразных процессов в промышленности, в научных исследованиях, в быту. Однако реализация этой предпосылки в значительной мере определяется возможностями устройств для получения информации о регулируемом параметре или процессе. Одним из таких устройств является датчик [1].

Датчик – средство измерений, предназначенное для выработки сигнала измерительной информации в форме, удобной для передачи, дальнейшего преобразования, обработки и (или) хранения, но не поддающейся непосредственному восприятию наблюдателем [2].

Другими словами, датчик – это преобразователь измеряемой (контролируемой) физической величины в величину, удобную для дальнейшего преобразования или измерения.

Датчик является обязательным элементом измерительных приборов, систем контроля и регулирования. Собственно, без датчиков невозможны ни измерение, ни контроль, ни регулирование. Принципы действия датчиков могут быть самыми разнообразными в зависимости, например, от физической природы измеряемой величины, её абсолютного значения, требуемой точности.

Существует значительное количество классификаций датчиков по различным классификационным признакам. Наиболее полное представление о всей разнообразии датчиков можно получить, приводя общую классификацию с учетом многих классификационных признаков (рисунок 1.1).

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

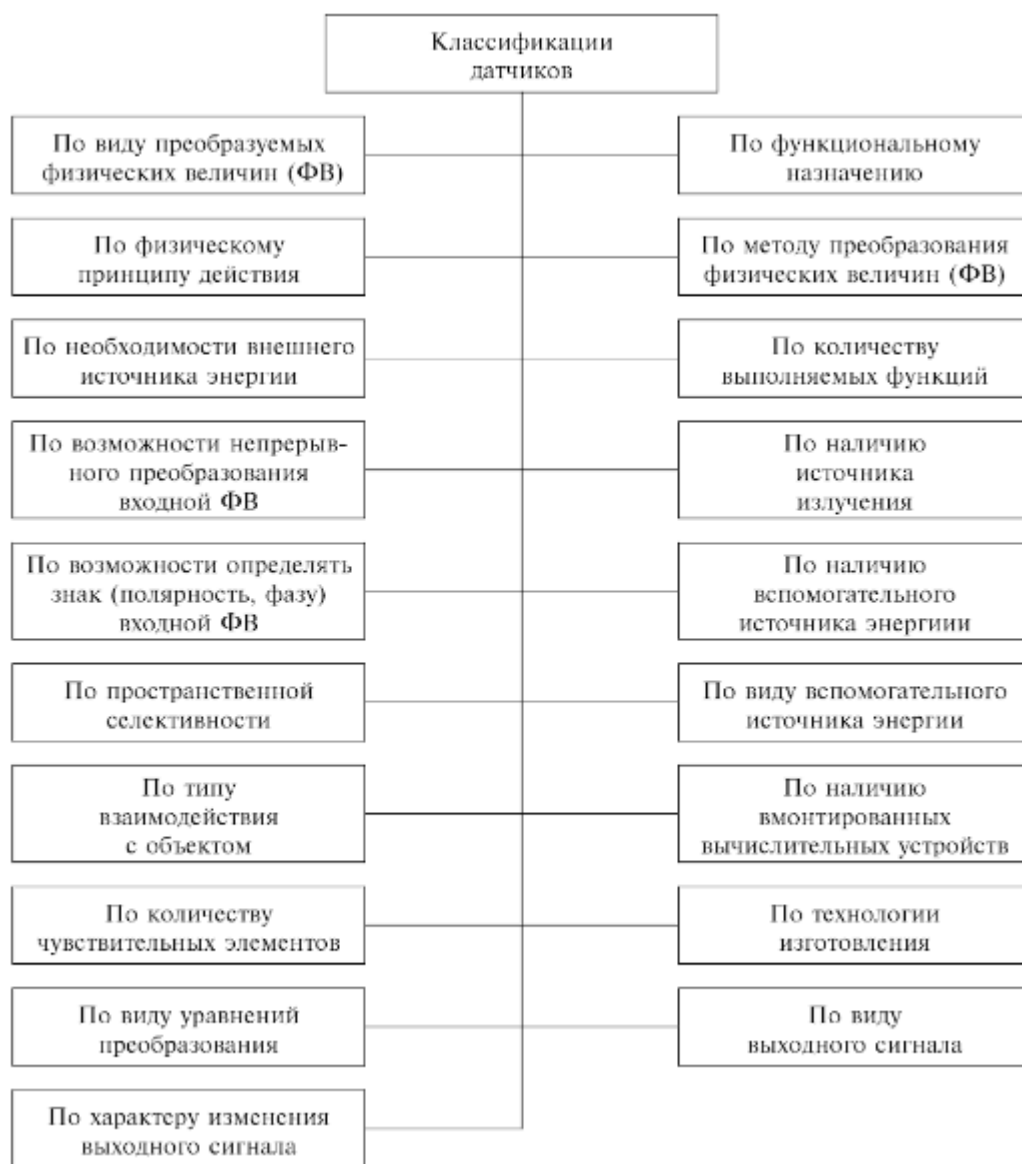


Рисунок 1.1 – Виды классификации датчиков

Большой интерес вызывает классификация измерительных преобразователей (датчиков) по виду входных и выходных величин (рисунок 1.2), которая состоит из четырех больших классов.



Рисунок 1.2 – Классификация преобразователей по виду  
ВХОДНЫХ И ВЫХОДНЫХ ВЕЛИЧИН

Изм.	Лист	№ докум.	Подпись	Дата



Важнейшим классификационным признаком для датчиков является физический принцип действия – принцип преобразования физических величин, который основывается на некотором физико-техническом эффекте (явлении) (рисунок 1.3).



Рисунок 1.3 – Классификация датчиков по принципу действия

Рассмотрим датчики, способные любую величину преобразовать в электрическую, поподробнее.

Назначение датчиков – реакция на определенное внешнее физическое воздействие и преобразование его в электрический сигнал, совместимый с измерительными схемами. Другими словами, можно сказать, что датчик – это преобразователь физической величины (часто неэлектрической) в электрический сигнал. Под термином электрический сигнал понимается сигнал, который может быть преобразован при помощи электронных устройств, например, усилен или передан по линии передач. Выходными сигналами датчиков могут быть напряжение, ток или заряд, которые описываются следующими характеристиками: амплитудой, частотой, фазой или цифровым кодом. Этот набор характеристик называется форматом выходного сигнала. Таким образом, каждый датчик характеризуется набором входных параметров (любой физической природы) и набором выходных электрических параметров.[4]

Техника конструирования и применения датчиков, или, как её можно кратко назвать, сенсорики, за последние годы развилась в самостоятельную ветвь

гкцизмерительной техники. С ростом автоматизации к датчикам физических параметров стали предъявляться все более высокие требования. При этом особое значение придается следующим показателям:

- миниатюрность (возможность встраивания);
- дешевизна (серийное производство);
- механическая прочность [1].

### 1.1.2 Эффект Холла и его применение в датчиках

Эдвин Холл открыл это физическое явление в 1879 году. Первоначально этот эффект применялся для изучения электропроводности металлов, полупроводников и других токопроводящих материалов. В настоящее время датчики Холла используются для обнаружения магнитных полей и определения положения и перемещения объектов.

Эффект Холла основан на взаимодействии между движущимися носителями заряда и внешним магнитным полем. В металлах носителями зарядов являются электроны. При движении электронов в магнитном поле на них действует отклоняющая сила (сила Лоренца):

$$\mathbf{F} = q\mathbf{v}\mathbf{B}, \quad (1.1)$$

где  $q = 1.6 \times 10^{-19}$  Кл – величина заряда электрона;

$v$  – его скорость;

$\mathbf{B}$  – магнитная индукция.

Выделенный шрифт указывает на то, что  $\mathbf{F}$  и  $\mathbf{B}$  являются векторами. Направление силы и ее величина зависят от пространственного расположения магнитного потока и направления движения электрона. Единицей измерения  $\mathbf{B}$  является тесла: 1 Тесла = 1 Ньютон / (ампер×метр) =  $10^4$  Гаусс.

Предположим, что электроны двигаются внутри электропроводной пластины, помещенной в магнитное поле  $\mathbf{B}$  (рисунок 1.4).

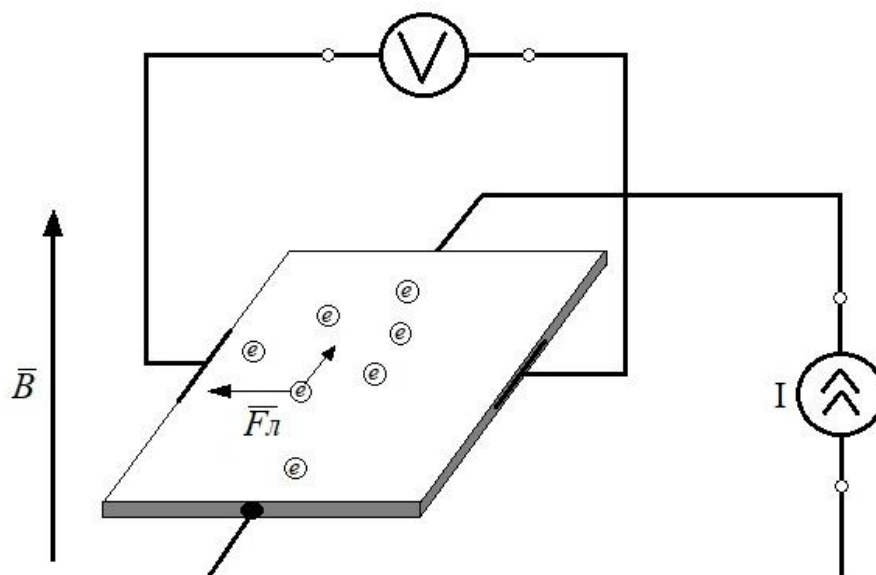


Рисунок 1.4 – Графическое представление эффекта Холла

На две стороны пластины нанесены дополнительные электроды, подключенные к вольтметру. Ещё два электрода расположены сверху и снизу пластины, они подсоединены к источнику электрического тока. Из-за действия внешнего магнитного поля возникает отклоняющая сила, смещающая электроны ближе к левому краю пластины, поэтому эта сторона становится более отрицательно заряженной, чем левая. Очевидно, что вследствие взаимодействия магнитного поля и электрического тока возникает поперечная напряженность потенциалов, получившая название напряженность Холла  $V_H$ . Знак и амплитуда этого напряжения зависят как от величины, так и направления магнитного и электрического полей. При фиксированной температуре оно определяется выражением:

$$V_H = hiB \sin \alpha, \quad (1.2)$$

где  $\alpha$  – угол между вектором магнитного поля и плоскостью пластины Холла (рисунок 1.5);

$h$  – полная чувствительность датчика, на значение которой влияют тип материала пластины, её геометрия (площадь активной зоны) и температура;

$i$  – выходной ток.

Полная чувствительность датчика Холла зависит от коэффициента Холла, который определяется градиентом поперечного электрического потенциала на единицу интенсивности магнитного поля и на единицу плотности тока. В соответствии с теорией свободных электронов в металлах, коэффициент Холла можно найти при помощи выражения:

$$H = \frac{1}{Ncq}, \quad (1.3)$$

где  $N$  – число свободных электронов в единице объема;

$c$  – скорость света.

В зависимости от кристаллической структуры материала заряды могут быть либо электронами (отрицательными), либо дырками (положительными). Поэтому и эффект Холла бывает либо положительным, либо отрицательным.

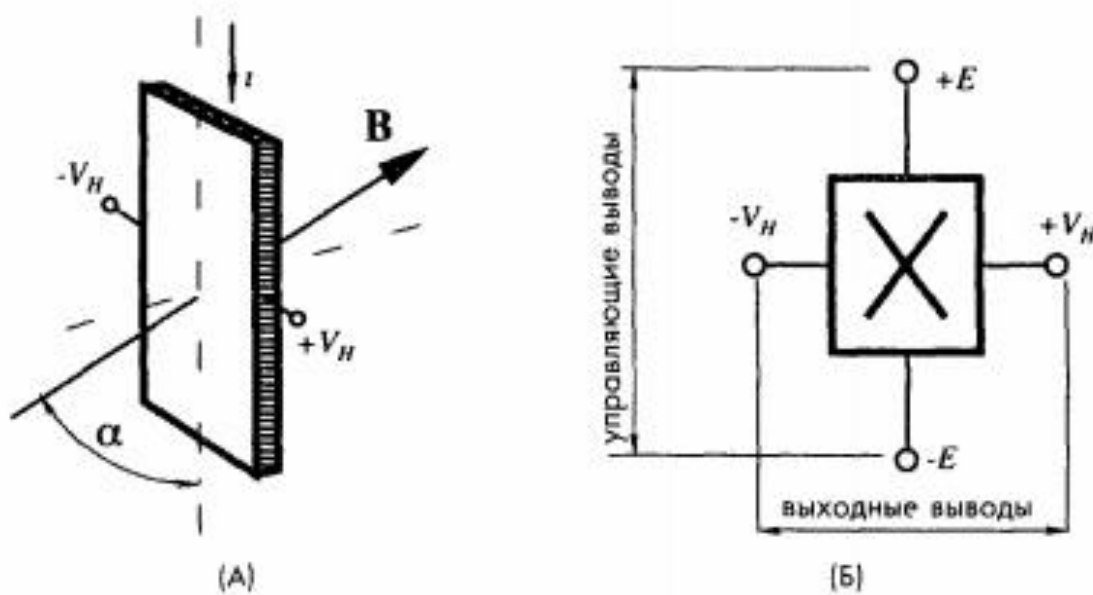


Рисунок 1.5 – Датчик Холла: а) выходной сигнал датчика Холла зависит от угла между вектором магнитного поля и плоскостью пластины; б) четыре вывода датчика Холла

Линейный датчик Холла обычно размещается в корпусе с четырьмя выводами. Два вывода для подключения тока управления называются управляющими выводами, а сопротивление между ними – сопротивление управляющей цепи  $R_i$ . Выводы для измерения выходного напряжения называются дифференциальными

выходами, а сопротивление между ними – выходным дифференциальным сопротивлением  $R_0$ . Эквивалентную схему датчика Холла (рисунок 1.6) можно представить в виде узлового соединения 4-х резисторов и двух источников напряжения, включенных последовательно с выходными выводами.

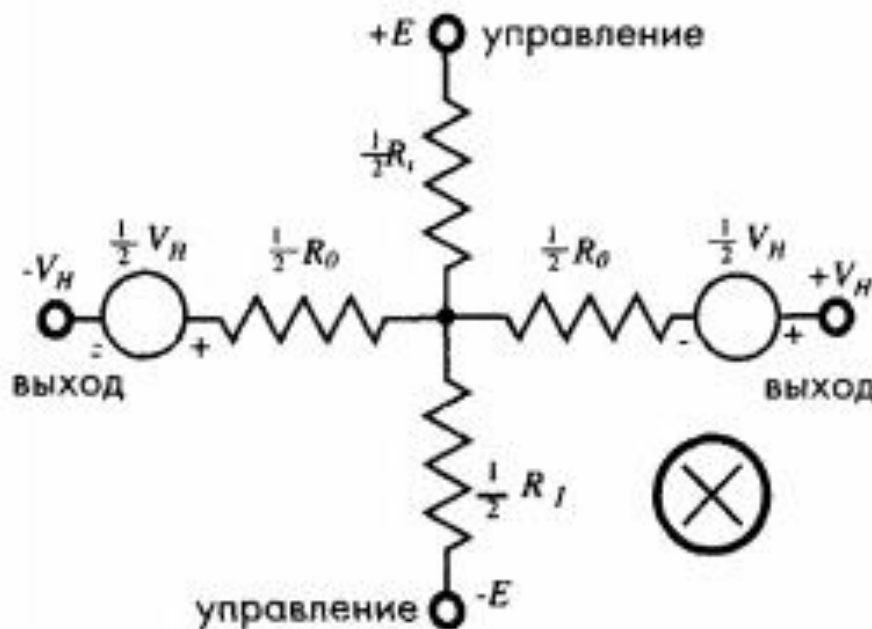


Рисунок 1.6 – Эквивалентная схема датчика Холла

Знак  $\otimes$  на рисунках 1.5 б) и 1.6 указывает на то, что вектор  $\mathbf{B}$  направлен от наблюдателя.

Датчик характеризуется следующими параметрами:

- сопротивлениями  $R_i$  и  $R_0$ ;
- напряжением смещения при отсутствии магнитного поля;
- чувствительностью;
- температурным коэффициентом чувствительности [4].

### 1.1.3 Перспективные разработки

В статье [5] в процессе выполнения исследований, связанных с изучением влияния магнитного поля со сложной динамикой его изменения на работу цифрового интеллектуального датчика температуры DS18B20, возникла необходимость непосредственной оценки параметров магнитного поля в области размещения датчика. Отсутствие в их распоряжении специализированного

магнитометра с возможностью точечной оценки векторных и количественных параметров поля, заставило провести дополнительное исследование по разработке приемлемого в такой ситуации метода решения возникшей проблемы.

Одним из доступных магниточувствительных элементов, позволяющих, в принципе, справиться с поставленной задачей, является датчик Холла. Однако, для решения измерительной задачи, с получением количественной оценки результатов, имеющейся по датчику Холла информации недостаточно. Необходимы тарировочные данные для конкретных типов датчика Холла, полученные в различных по характеру магнитных полях. Целью статьи является разработка методики тарировки (построения градуировочной характеристики) датчика Холла без использования труднодоступного и, как правило, дорогостоящего оборудования.

В качестве объекта для исследований был выбран датчик Холла типа SS495A (рисунок 1.7).

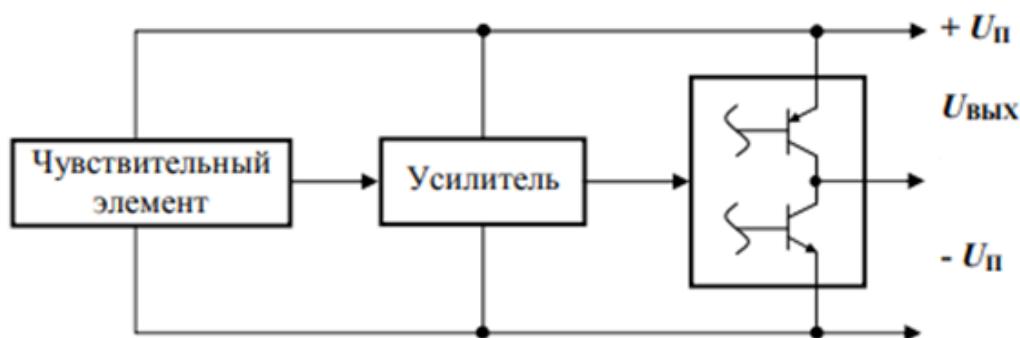


Рисунок 1.7 – Структурная схема датчика Холла типа SS495A

Для нормирования параметров датчика была измерительная катушка индуктивности на картонном каркасе, имеющая 1343 витков провода ПЭЛШО, диаметром 0,2 мм. Поскольку наиболее однородное и однозначно определяемое поле создается в геометрическом центре катушки, именно здесь был и размещен тарлируемый датчик Холла (рисунок 1.8).

Для геометрического центра соленоида с током, в котором размещен датчик Холла, имеет силу следующая формула напряженности магнитного поля:

$$H_{\max} = \frac{N}{L} I_{\text{кат}} \left[ 1 + \left( \frac{D}{L} \right)^2 \right]^{-\frac{1}{2}} = I_{\text{кат}} \cdot G, \quad \frac{A}{M},$$

(1.4)

где  $G = \left[ 1 + \left( \frac{D}{L} \right)^2 \right]^{-\frac{1}{2}};$

$D = 45$  мм – средний диаметр катушки;

$L = 85$  мм – длина намотки катушки;

$N$  – число витков катушки;

$I_{\text{кат}}$  – сила тока, протекающего через катушку.

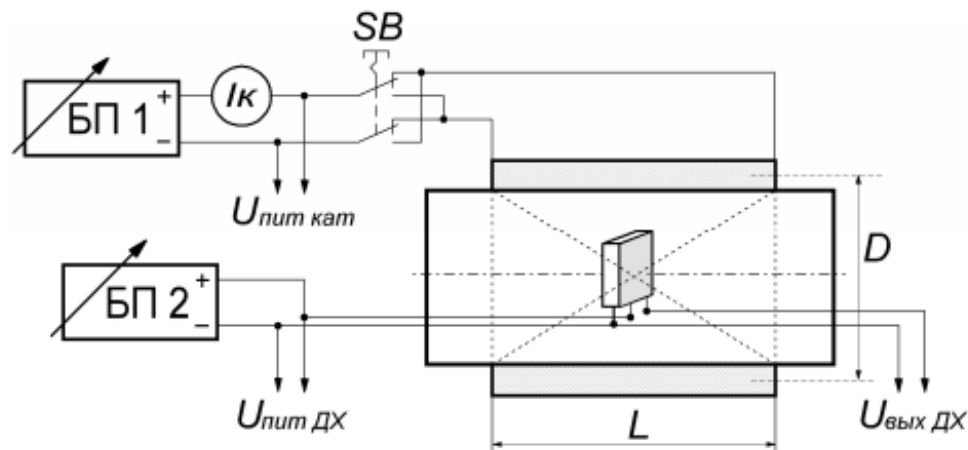


Рисунок 1.8 – Схема тарировки датчика Холла по магнитному полю катушки

Были получены результаты измерений выходного напряжения датчика Холла при различных токах, протекающих через катушку. Напряжение питания датчика Холла выбрано равным 9 В.

С помощью дальнейшей обработки результатов была проведена линейная аппроксимация имеющихся результатов, которая позволила получить эмпирическую формулу зависимости напряжения датчика Холла от тока катушки:

$$U_{\text{ВЫХ ДХ}} = 0,0105 \cdot I_{\text{кат}} + 4,6045, \quad (1.5)$$

Также для повышения чувствительности датчика Холла было решено установить выходной сигнал с помощью усилителя на операционном усилителе типа AD623 (рисунок 1.9).

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

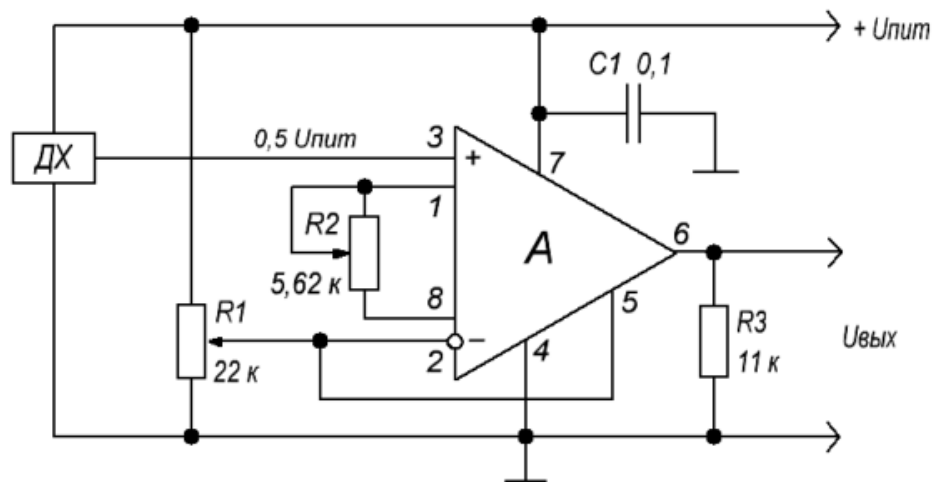


Рисунок 1.9 – Схема подключения датчика Холла к операционному усилителю AD623

Были получены результаты измерения выходного напряжения датчика Холла при различных напряжениях питания датчика и при фиксированных токах катушки. Аппроксимация экспериментальных данных в итоге позволила получить зависимость значения измеренной датчиком напряженности магнитного поля  $H$  от напряжения питания и выходного напряжения датчика Холла:

$$H = \frac{9123}{4,16 \cdot \ln(U_{\text{ПИТ ДХ}}) - 0,593 \cdot U_{\text{ПИТ ДХ}} - 3,14} \times \left( U_{\text{ВЫХ ДХ}} - \sqrt{0,308 \cdot U_{\text{ПИТ ДХ}}^2 - 2,7} \right) \quad (1.6)$$

Соотношение (1.6) позволяет оценивать параметры постоянного магнитного поля в произвольной точке пространства с помощью датчика Холла при любом доступном напряжении его источника питания, что даст возможность провести исследование поведения датчика температуры.

С целью решения поставленной выше задачи была проведена оценка работы датчика Холла в переменном магнитном поле. Исследование проводилось по схеме, аналогичной для постоянного магнитного поля. На катушку подавались дискретные значения переменного напряжения. При этом фиксировались токи катушки и выходное переменное напряжения датчика Холла. В результате аппроксимации получено:



$$-U_{ВЫХ ДХ} = 0,0105 \cdot (-I_{кат}) + 4,6045. \quad (1.7)$$

Сравнивая полученные эмпирические зависимости (1.5) и (1.7), обнаружена их полная идентичность, что позволяет сделать вывод о корректности измерений и расчетов и принципиальной возможности использования оттарированного датчика Холла для объектного исследования локальных характеристик постоянного и переменного магнитных полей.

Также были рассмотрены статьи в области магнитного поля с применением эффекта холла такие, как [6-9].

[6] рассматривает трехмерный магнитометр, который предназначен для измерения индуктивности постоянного магнитного поля в заданной точке пространства. При помощи трех датчиков можно определить вектор индукции постоянного магнитного поля. Таким образом, полученные показания будут точнее, чем при использовании одного датчика. Кроме того, полученный вектор будет указывать на максимум поля, который, как правило, находится там же, где и источник этого магнитного поля.

Был взят за основу магнитометр на основе трех датчиков Холла, расположенных в трех перпендикулярных плоскостях. Для устранения недостатков данного типа магнитометров, расположенных в трех перпендикулярных плоскостях, предложена система, состоящая из двух вертикальных и одного горизонтального датчика Холла, сформированных на одной кремниевой подложке с использованием стандартной 0,35 – мкм технологии КМОП (рисунок 1.10). Расположенные под прямым углом относительно друг друга вертикальные датчики Холла измеряют составляющие индукции магнитного поля, ориентированные в плоскости кремниевой подложки. Горизонтальный датчик Холла предназначен для измерения составляющей индукции поля, направленной по нормали к поверхности подложки.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

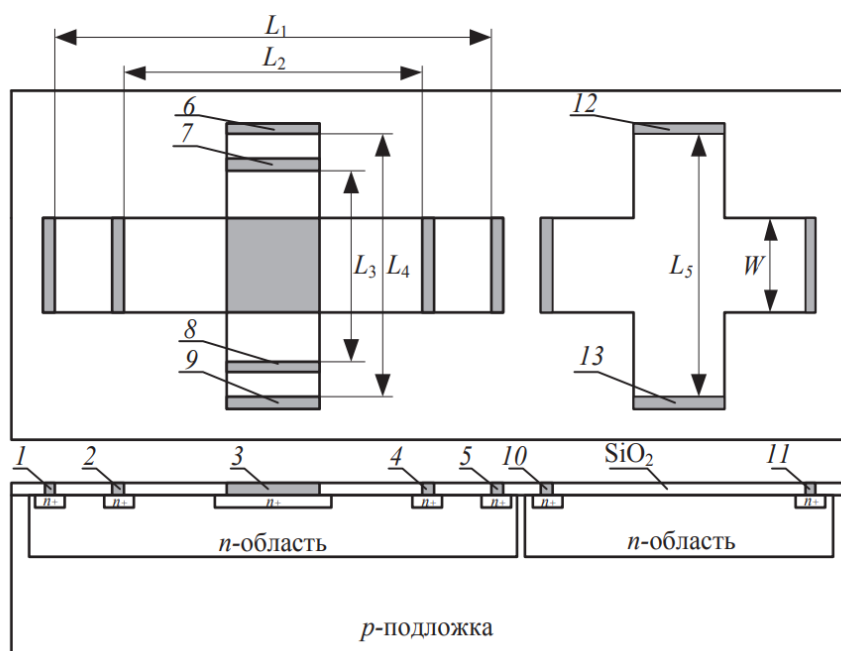


Рисунок 1.10 – Структура трехмерного магнитометра на основе двух вертикальных и одного горизонтального датчика Холла

В приповерхностном слое кремниевой подложки  $p$ -типа электропроводности с концентрацией примеси  $1 \cdot 10^{15} \text{ см}^{-3}$  создаются две  $n$ -области глубиной 1,4 мкм. В одной из этих областей формируются два перпендикулярных относительно друг друга вертикальных датчика Холла с геометрическими размерами  $L_1 = 25 \text{ мкм}$ ,  $L_2 = 20 \text{ мкм}$ ,  $L_3 = 6 \text{ мкм}$  и  $L_4 = 9 \text{ мкм}$ , а в другой – холловский крест из полосок длиной  $L = 9 \text{ мкм}$  и шириной  $W = 3 \text{ мкм}$ . Прилегающие к омическим контактам 1, 5, 6, 9 и 10  $n^+$  области легированы фосфором. Контакты 3 и 11 представляют собой токовые контакты для вертикальных датчиков Холла и холловского креста, соответственно. Холловские электроды 2, 4, 7 и 8 предназначены для измерения напряжения Холла при воздействии магнитного поля, ориентированного в плоскости подложки, в то время как электроды 12 и 13 используются для измерения напряжения Холла при воздействии поля, направленного по нормали к поверхности подложки.

Чувствительности кремневых датчиков Холла не всегда достаточно, для достижения более высокой чувствительности можно использовать в качестве подложек тонкие пленки  $\text{InSb}$  или  $\text{InAs}$ .

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Таким образом, при рассмотрении современных магнитометров, была выделена перспективная для коммерческого применения структура, основанная на трех датчиках Холла и позволяющая проводить трехмерное измерение.

В работах [7-8] объектом исследования являются датчики Холла для слабых магнитных полей. В [7] представлены результаты исследований структур, на основе которых изготовлены преобразователи Холла (ПХ), вольт-амперных характеристик полученных образцов и показана возможность оптимизации конструкции ПХ, обеспечивающей низкие остаточные напряжения.

В разработке ПХ на основе арсенида галлия использовались структуры  $n_i - n_{\sigma} - n - n^+$  - типа. Слои, легированные серой, выращены хлоридным методом газофазной эпитаксии. Концентрация носителей заряда в n-слое толщиной от 0,3 до 1,6 мкм изменялась от  $5 \cdot 10^{15}$  до  $10^{17} \text{ см}^{-3}$ . Толщина контактного  $n^+$  - слоя составляла  $0,3 \div 0,4$  мкм, концентрация  $\text{см}^{-3}$ . Подложка  $n_i$  была полуизолирующей, толщина буферного слоя ( $n_{\sigma}$ ) составляла 3 мкм. Кристаллы датчиков с планарной конструкцией активной n-области изготавливались по технологии полевых транзисторов на основе арсенида галлия. Омические контакты формировались путём напыления сплава (Au-Ge)+Ni с последующим вжиганием. Изоляция активных областей осуществлялась имплантацией водорода. Для разделения пластин на отдельные кристаллы использовали лазерное скрабирование. Кристаллы размещались на керамическом металлизированном основании и после разварки выводов заливались компаундом.

Датчик магнитного поля, разработанный в НИИПП на основе кристаллов арсенида галлия, имеет планарную конфигурацию. Простейшая эквивалентная схема преобразователя Холла в случае, когда кристалл полностью планарный (а), представляет собой мост сопротивлений (рисунок 1.11, а).

Если на вход моста подано управляющее напряжение  $U_1$ , обеспечивающее ток  $I_1$ , то напряжение на выходе в режиме питания от источника тока будет равно (рисунок 1.12, а)

										Лист
										20
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ 12.03.01. 2018.114-288 ВКР					

$$U_{\text{вых}} = \frac{R_1 \cdot R_4 - R_2 \cdot R_3}{R_1 + R_4 + R_2 + R_3} \quad (1.8)$$

Для баланса моста достаточно условия  $R_1 \cdot R_4 - R_2 \cdot R_3 = 0$ . В идеальном случае, если  $R_1 = R_2 = R_3 = R_4$ ,  $U_{\text{ост}}$  будет равно нулю. При отклонении от идеальности имеет место зависимость остаточного напряжения от управляющего напряжения или тока. Возможна компенсация  $U_{\text{ост}}$  при постоянной температуре схемой включения двух идентичных кристаллов на одной подложке или включением дополнительного компенсирующего сопротивления, однако особую сложность представляет температурная компенсация.

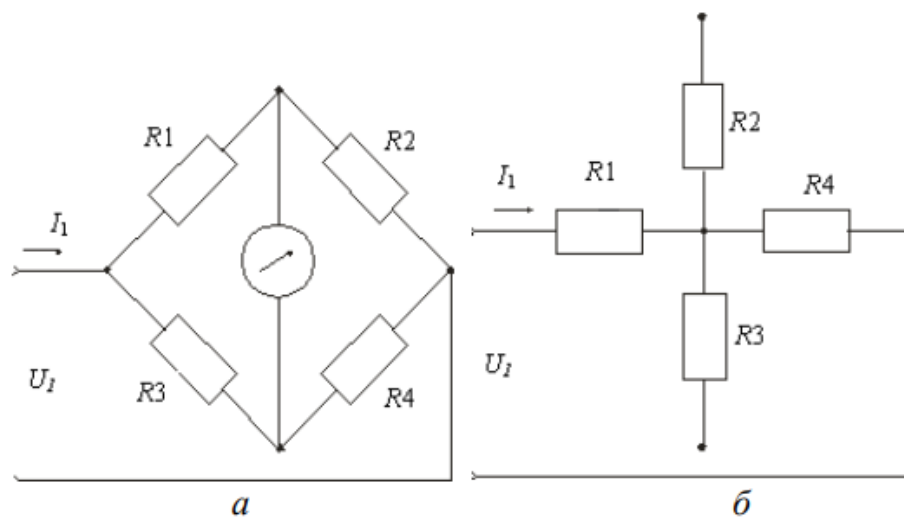


Рисунок 1.11 – Эквивалентная схема преобразователя Холла: а) планарного; б) с выставленными участками подложки

Альтернативным решением задачи является изменение конструкции преобразователя Холла. В измененной конструкции кристалла удалены участки структуры за пределами активной области. В этом случае эквивалентная схема преобразователя Холла имеет вид, представленный на рисунке 1.11, б, и основная причина появления  $U_{\text{ост}}$  тогда исключается. Необходимо устранить также операцию термокомпрессионной сварки входных и выходных выводов, которая приводит к увеличению остаточного напряжения. Необходимо также оптимизировать соединения к электрическим токовым и холловским выводам из осаждённого золота, которые привариваются к внешним выводам без нарушения кристалла. Кроме того, для уменьшения общей толщины преобразователей

Холла необходимо использовать подложку из фольгированного полиамида, толщина которого составляет 40–50 мкм. Конструкция преобразователя будет иметь в этом случае вид, представленный на рисунке 1.12. Сверху кристалл может заливаться компаундом или помещаться в стандартный корпус типа SOT-8. Общая толщина не превышает в этом случае при заливке 1 мм, а в корпусе 1,8 мм.

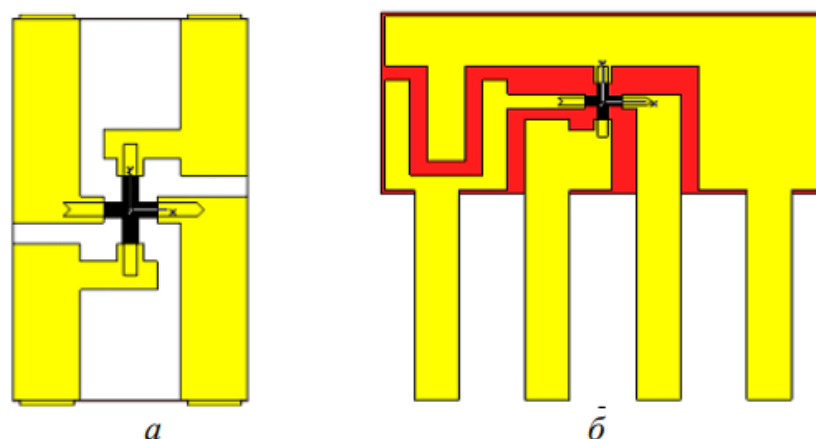


Рисунок 1.12 – Конструкция преобразователя Холла на полиамидной основе с выводами: а) в разные стороны; б) в одну сторону

Предлагаемая конструкция на полиамидной основе позволяет существенно повысить чувствительность преобразователя Холла.

В публикации [8] был проведен обзор различных вариантов датчиков слабых магнитных полей. Предложена перспективность применения в измерениях слабых магнитных полей датчиков Холла с концентраторами магнитного поля. Была рассчитана предельная чувствительность датчика Холла.

В статье [10] описан холловский градиометр, предназначенный для построения на его основе системы дистанционного магнитного мониторинга технического состояния трубопроводов различного назначения.

В основу разработки положен трёхкомпонентный прецизионный Холловский магнитометр. Блок-схема градиометра приведена на рисунке 1.13.

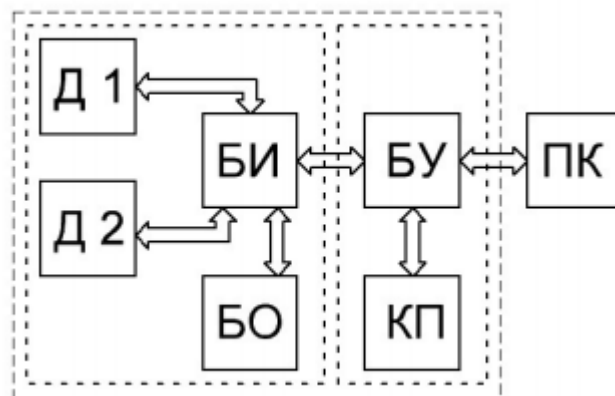


Рисунок 1.13 – Блок-схема холловского градиометра

Градиометр оснащён двумя трёхкомпонентными датчиками Д1 и Д2, каждый из которых состоит из трёх преобразователей Холла ортогонально расположенных относительно друг друга. Сигналы с Д1 и Д2 поступают на блок измерения (БИ), в котором они преобразуются в цифровую форму. Блок содержит однокристалльный микроконтроллер, который управляет процессом измерения и реализует алгоритм уменьшения систематической погрешности преобразователя Холла. В БИ также входит блок ориентации (БО), состоящий из трёхкомпонентного акселерометра и твердотельного гироскопа.

Управление градиометром и индикация результатов производится блоком управления (БУ). Блок реализован на одноплатном модуле TESTM32F439LCD35@TERRA, ядром которого является 32-разрядный микроконтроллер STM32F439. Модуль оснащён TFT-дисплеем 3,5 дюйма, контроллером сенсорного экрана и картой памяти (КП) формата microSD, позволяющей сохранять результаты измерений для последующей детальной обработки. Используемый микроконтроллер имеет встроенный контроллер USB 2.0 Full-speed, с помощью которого результаты измерений передаются на персональный компьютер (ПК) без извлечения накопителя информации. Такая возможность передачи данных удобна для первичной настройки и калибровки градиометра. Конструкция градиометра представлена на рисунке 1.14.

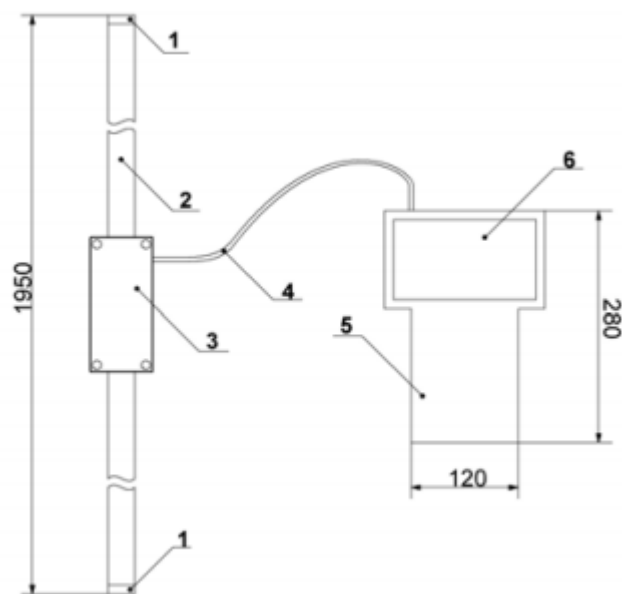


Рисунок 1.14 – Эскиз холловского градиометра: 1 – датчики градиометра, 2 – алюминиевый профиль, 3 – алюминиевый корпус в котором размещены БИ и БО, 4 – соединительный кабель, 5 – БУ с размещёнными внутри него элементами питания, 6 – сенсорный дисплей.

Базой прибора является алюминиевый профиль (2), квадратного сечения  $30 \times 30$  мм, в торцах которого расположены датчики (1), Д1 и Д2. В центре профиля закреплён литой алюминиевый корпус (3), размером  $35 \times 56 \times 140$ , в котором размещены блоки измерения и ориентации. Результаты измерения передаются на блок управления и индикации (5), с помощью 4-х жильного соединительного кабеля (4). Результаты измерения отображаются на ЖК индикаторе (6). Особенностью градиометра является то, что он производит измерение двух векторов магнитного поля и вычисляет градиент компонентов вектора. Система обработки позволяет отображать проекции и модули магнитного поля, координаты точек пространства, в которых расположены датчики, а так же определять градиент магнитного поля.

#### Технические характеристики

Градиометр является портативным аппаратно-программным комплексом и имеет следующие характеристики:

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

1) Уровень эквивалентного шума, приведенного к входу прибора:

- от пика до пика 400 нТл
- среднеквадратичное отклонение 67 нТл
- спектральный уровень шума 20 нТл, в полосе частот 0,1 – 10 Гц

2) Динамический диапазон  $\pm 30$  мТл

3) Батарейное питание от двух аккумуляторов 3,7 В, в течении 5 часов непрерывных измерений.

4) Сенсорный экран 320×240 точек.

5) Поддержка карт памяти до 32 Гб, для сохранения магнитограмм.

6) Интерфейс передачи данных – USB.

По результатам аналитического обзора можно сделать вывод о достаточно широком применении как непосредственно датчика холла, так и приборов на его основе. Также стоит отметить, что в большей части рассмотренных работ применены средства цифровой обработки информации, для взаимодействия с которыми необходимо специализированное программное обеспечение.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25



# ЭКСПЕРИМЕНТАЛЬНОЕ ОПРЕДЕЛЕНИЕ ПРЕДЕЛОВ ИЗМЕРЕНИЯ ДАТЧИКА ХОЛЛА

## 2.1 Суть эксперимента с датчиком Холла

В данной работе датчик Холла будет положен в основу прибора для измерения тока в проводнике бесконтактным способом. Чтобы понять принцип работы этого устройства, рассмотрим его упрощённую схему на рисунке 2.1.

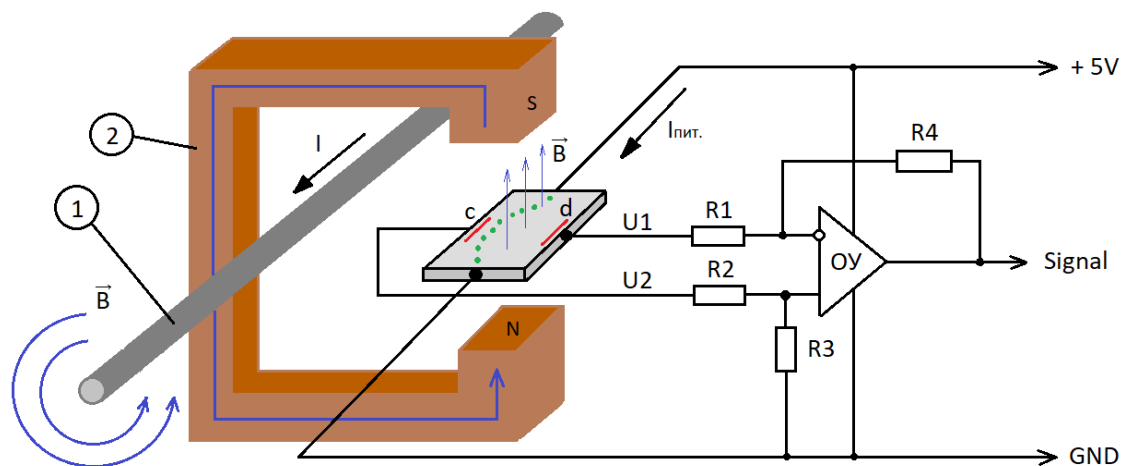


Рисунок 2.1 – Принципиальная электрическая схема измерительного преобразователя тока бесконтактным методом

Исследуемым сигналом в цепи является ток  $I$  электрического проводника 1. Этот проводник охвачен ещё одним проводником, так называемым магнитопроводом 2. Обычно он изготавливается из магнитных материалов – ферритов, имеющих индекс магнитной проницаемости  $\mu_n$  до  $5 \cdot 10^4$  и характеризующихся высоким удельным сопротивлением  $\rho \sim 10^{14} \text{ Ом} \cdot \text{м}$ . Ферриты – это соединения двойных окислов железа и одно-двухвалентных металлов (никеля, цинка, марганца, бария, лития и др.), обладающие свойствами ферромагнетизма.

Известно, что вокруг проводника с током существует магнитное поле  $\vec{B}$ , направление которого можно узнать по правилу буравчика, зная направление тока в проводнике. В магнитопроводе 2 наводится магнитное поле пропорциональное магнитному полю проводника с током, после чего этот магнитопровод можно с

некоторым приближением рассматривать как магнит с северным (N) и южным (S) полюсами между обкладок в его зазоре. В зазор помещён плоский проводник – чувствительный элемент, по которому течёт ток  $I_{\text{пит}}$  (в реальном эксперименте датчик, помещённый в зазор, уже содержит в себе усилитель и оконечный каскад). С противоположных концов пластинки прикреплены ещё 2 проводника в точках с и d, собирающие разноимённые носители заряда. При прохождении магнитного поля через пластинку с током носители зарядов с разным знаком отклонятся на противоположные стороны пластинки (одни к краю с, другие – к краю d) образуя разность потенциалов.

В реальности разность потенциалов на концах с и d пластинки очень мала и для её усиления используется обычный дифференциальный усилитель, обычно уже встроенный в головку преобразователя Холла. В общем случае:  $R_1 = R_2$ ,  $R_4 = R_3$ . Поэтому выходное напряжение можно найти как:

$$U_{\text{вых.}} = \frac{R_4}{R_1} (U_2 - U_1), \quad (2.1)$$

На картинке  $U_{\text{вых.}}$  обозначено как Signal. Из формулы (1) видно, что для увеличения выходного напряжения требуется увеличить номинал резистора ООС –  $R_4$ , либо уменьшить номинал резистора  $R_1$ . Обозначим  $\frac{R_4}{R_1} = \gamma$ , тогда:

$$U_{\text{вых.}} = \gamma (U_2 - U_1), \quad (2.2)$$

Следует помнить, что увеличивая  $\gamma$ , увеличится как мощность полезного сигнала, так и шумов от влияния температуры, внешних магнитных полей, неидеальности ОУ и т.д.

Полученное нами выходное напряжение  $U_{\text{вых.}}$  с некоторым коэффициентом пропорционально току в исследуемом проводнике, зная этот коэффициент, а он всегда известен, получают значение тока в исследуемом проводнике. Сигнал выходного напряжения имеет знак, поэтому можно, также, определить и направление тока в проводнике.

## 2.2 Описание элементов устройства

Внешний вид измерительной установки изображён на рисунке 2.2.

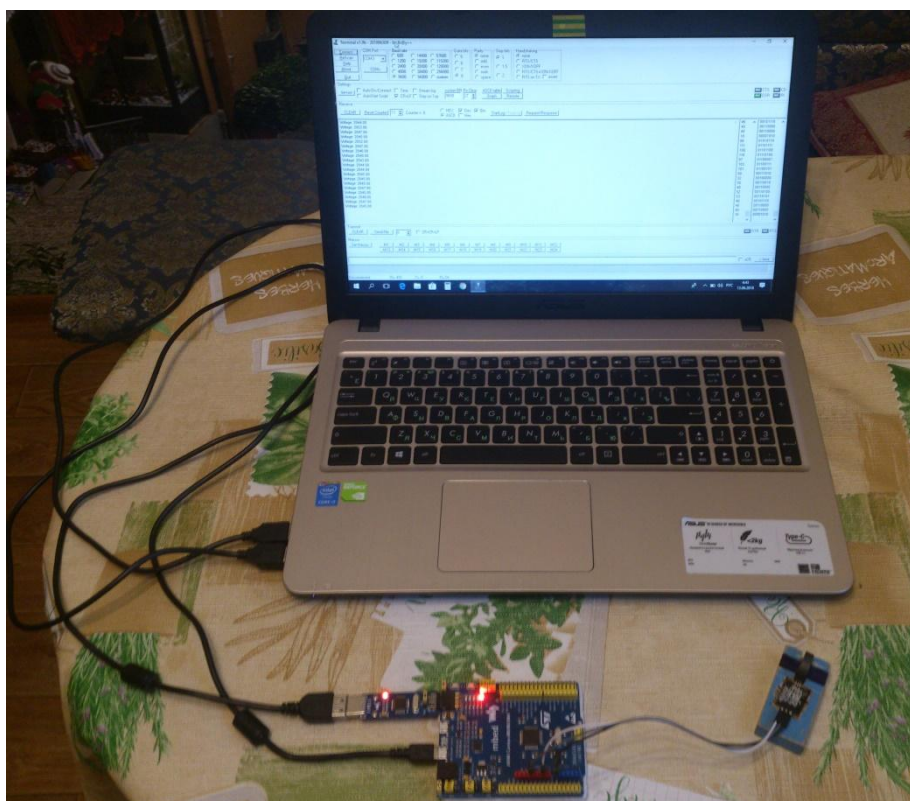


Рисунок 2.2 – Внешний вид измерительной установки  
где 1 – непосредственно датчик тока на основе датчика Холла,  
2 – отладочная плата XNUCLEO-F411RE,  
3 – персональный компьютер, используемый в качестве терминала.  
Для сборки датчика тока использовался кольцевой ферритовый сердечник  
M1000NM, 20x10x5, изображённый на рисунке 2.3.



Рисунок 2.3 – Ферритовый сердечник

Изм.	Лист	№ докум.	Подпись	Дата

Характеристики изделия: начальная магнитная проницаемость  $\mu_H = 1000$ , материал – марганец-цинковое НЧ, типоразмер – внешний диаметр  $D = 20$  мм, внутренний диаметр  $d = 10$  мм, высота кольца  $h = 5$  мм. Сердечник предназначен для работы в слабых и сильных магнитных полях на частотах до 30 МГц.

Следующим компонентом является сам датчик Холла. Конструктивное исполнение датчика и его вид на микросхеме приведены на рисунке 2.4.



Рисунок 2.4 – Датчик холла: а) внешний вид датчика; б) его исполнение на микросхеме

Характеристики датчика приведены в таблице 2.1.

Таблица 2.1 – Электрические характеристики датчика Холла

Параметр	Сим вол	Условия	Минимально	Обычно	Максимально	Единицы
Потребляемый ток	$I_{CC}$			3.5	4.5	мА
Выходное напряжение	$V_{OUT}$		1.45	1.6	1.75	В
Чувствительность выходного напряжения		В от 0 Гс, до $\pm 1000$ Гс	1.1	1.6	2.1	мВ/Гс
Выходное сопротивление	$R_O$			60	120	Ом
Диапазон измерения магнитной	В		$\pm 650$	$\pm 1000$		Гс

индукции						
Выходной шум		Частота флуктуаций магнитного поля от 10 Гц до 10 кГц		90		мкВ

Следуя рекомендациям по применению этого датчика с платой микроконтроллера STM32, напряжение питания должно соответствовать 3.3 В. При таком питании выходное напряжение датчика (между выводами GND и Signal) составит  $V_{OUT} = 1.6$  В.

Функция изменения выходного напряжения от индукции магнитного поля имеет линейный вид с зонами насыщения (см. рисунок 2.5).

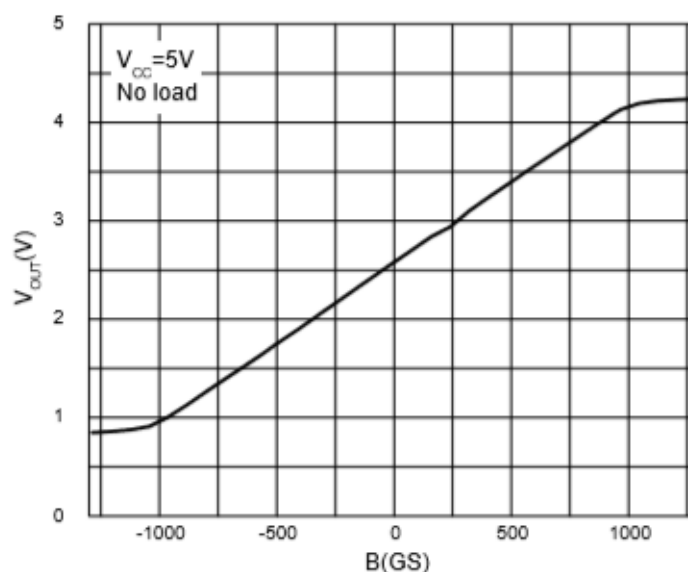


Рисунок 2.5 – График зависимости выходного напряжения от величины магнитной индукции

При построении зависимости использовалось питающее напряжение  $V_{CC} = 5$  В. Расположение чувствительного элемента в датчике показано на рисунке 2.6.

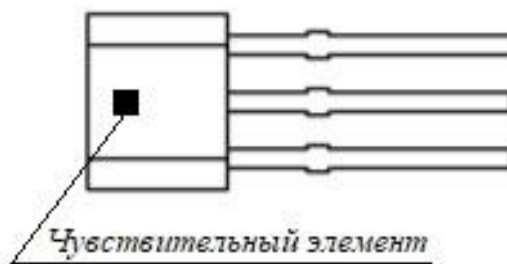


Рисунок 2.6 – Расположение чувствительного элемента в датчике

Платформой для разработки и отладки приложений для датчика послужила отладочная плата XNUCLEO-F411RE от производителя Waveshare. Её внешний вид представлен на рисунке 2.7.

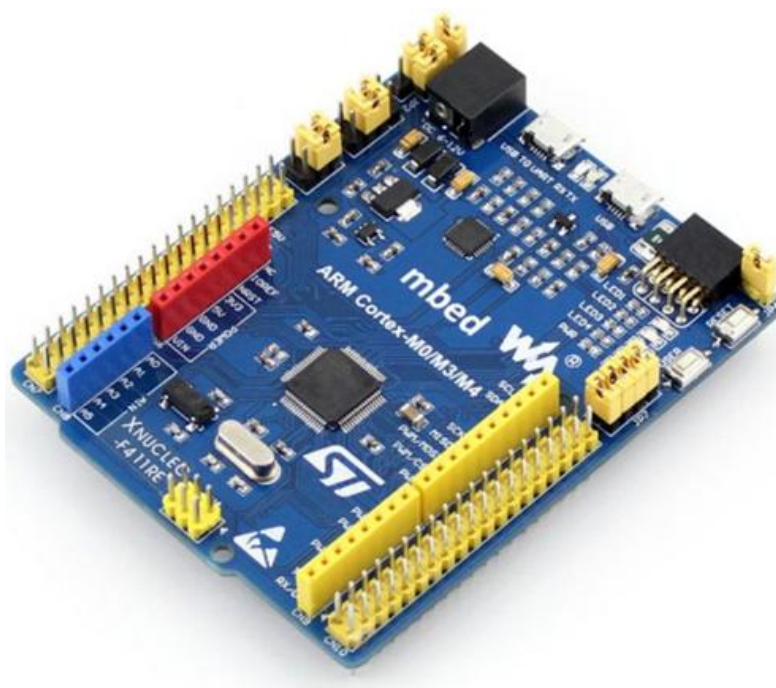


Рисунок 2.7 – Внешний вид отладочной платы

Основу данной платы составляет микроконтроллер STM32F411RET6. Он разработан на базе ядра Cortex-M4 с разрядностью машинного слова 32 бита и построенного по гарвардской архитектуре с разделением шины данных и шины команд, что повышает производительность благодаря распараллеливанию выборки инструкций и загрузки данных. Также архитектурой предусмотрен трёхступенчатый конвейер с выборкой инструкций, декодированием и выполнением инструкций. Во время выполнения одной инструкции, следующая - дешифрируется, а третья - считывается из памяти. Этот механизм отлично работает с линейным кодом, но, если требуется выполнить переход, то, прежде

чем продолжить выполнение кода программы, потребуется очистка и перезагрузка конвейера. Во избежание этого, трехступенчатый конвейер ЦПУ Cortex-M4 оснащён логикой предсказания переходов. Это означает, что при достижении инструкции условного перехода выполняется упреждающая выборка и, в результате, оба назначения инструкции условного перехода будут доступны для исполнения и снижение производительности не произойдет.

Также ядро поддерживает набор инструкций Thumb-2, который содержит как 32-битные, так и 16-битные инструкции для сокращения объёма кода за счёт оптимизации программы: 32-битные инструкции применяются, когда приоритетом является быстродействие программы, а 16-битные, когда более важна плотность кода. Структурная схема ЦПУ показана на рисунке 2.8.

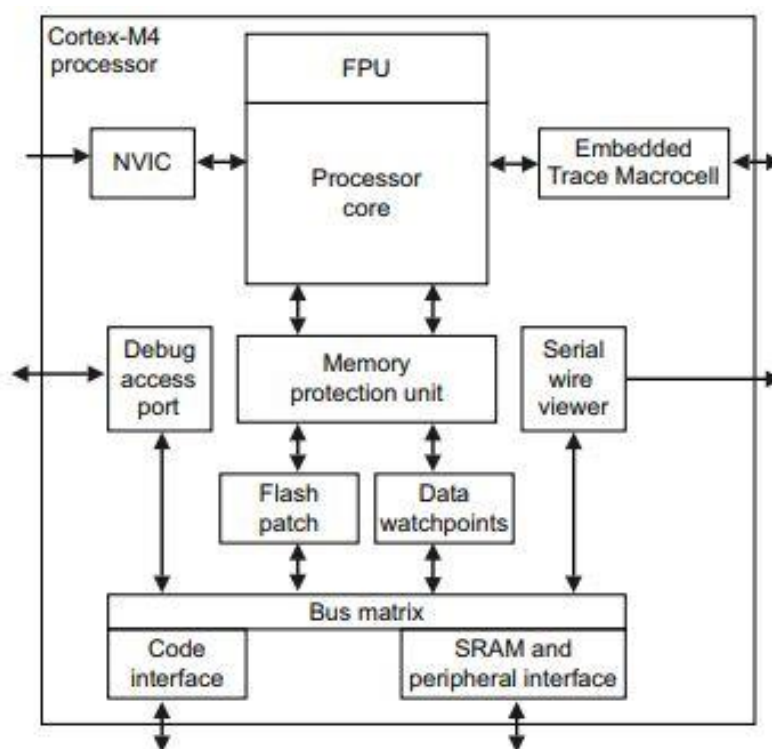


Рисунок 2.8 – Структурная схема ЦПУ Cortex-M4

На рисунке: FPU – блок, производящий операции над числами с плавающей точкой (математический сопроцессор), Processor core – ядро процессора, NVIC – модуль контроля прерываний, Embedded Trace Macrocell – встроенная трассировка макроячейки, Debug access port – порт отладки, Memory protection unit – блок защиты памяти, Serial wire viewer – средство просмотра последовательного канала, Flash patch - Флэш-патч, Data watchpoints – регистр для

Изм.	Лист	№ докум.	Подпись	Дата

подсчёта времени исполнения команд, Bus matrix – общая шина, Code interface – интерфейс кода, SRAM and peripheral interface – интерфейс для периферийных устройств и статической оперативной памяти.

Основными характеристиками микроконтроллера STM32F411RET6 являются: напряжение питания от 1.7 до 3.6 В, работа на частотах до 100 МГц, потребляемая мощность – 100 мкА/МГц, память программ – 512 кбайт, оперативная память – 128 кбайт, встроенный 12-битный 24-канальный АЦП последовательного приближения, контроллер прямого доступа к памяти на 16 каналов, использование алгоритма нахождения контрольной суммы для проверки целостности данных, 6 16-разрядных и 2 32-разрядных таймера, 81 порт ввода/вывода. Используемые интерфейсы приёма и передачи данных: USB 2.0, USART для 3х каналов, SPI/I2S для пяти каналов, I2C для 3х каналов, SDIO для SD/MMC/eMMC.

Далее подробнее рассмотрим систему тактирования микроконтроллера, порты общего назначения и АЦП.

Для формирования системной тактовой частоты процессора могут использоваться три основных источника:

- 1) HSI (high-speed internal) – внутренний высокочастотный RC-генератор;
- 2) HSE (high-speed external) – внешний высокочастотный генератор;
- 3) PLL – система фазовой автоподстройки частоты.

Также имеются два вторичных источника тактового сигнала:

- 4) LSI (low-speed internal) – низкочастотный внутренний RC-генератор, частотой 37 кГц;
- 5) LSE (low-speed external) – низкочастотный внешний источник, частотой 32,768 кГц.

У HSI сигнал формируется внутренним RC-генератором, настроенным на частоту 16 МГц. Точность этого генератора ниже, чем у HSE с кварцем, но из-за отсутствия внешних элементов обладает быстрым временем запуска.

Заводская калибровка производится с допустимой погрешностью в 1% при температуре 25°C. Калибровочное значение находится в битах HSICAL

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33



[7:0] регистра RCC\_ICSCR — Internal Clock Sources Calibration Register.  
Пользователю эти биты доступны только для чтения.

У HSE высокочастотный внешний сигнал может быть получен двумя способами:

- От внешнего источника тактового сигнала (от 4 до 26 МГц). Форма сигнала может быть не только прямоугольной, но и синусоидальной или треугольной. Коэффициент заполнения при этом должен быть равен 50%, то есть для прямоугольного импульса это "меандр".
- С использованием внешнего кварцевого/керамического резонатора от 4 МГц до 24 МГц.

PLL получает исходный сигнал от HSI или HSE. Он имеет два выхода:

1) Первый выход используется для генерации высоких частот – до 100 МГц, его выходная частота рассчитывается по формуле 2.3:

$$f = f(\text{PLL clock input}) \times (\text{PLLN} / \text{PLLM}) / \text{PLLP} \quad (2.3)$$

где PLLN – основной коэффициент умножения;

PLLM – коэффициент деления для основного входного тактового сигнала PLL;

PLLP – основной коэффициент деления.

2) Второй выход используется для подачи тактирования на интерфейс USB (48 МГц) и SDIO ( $\leq 50$  МГц). При этом частота высчитывается по формуле 2.4:

$$f(\text{USB}) = f(\text{PLL clock input}) \times (\text{PLLN} / \text{PLLM}) / \text{PLLQ} \quad (2.4)$$

где PLLQ – основной коэффициент деления PLL для USB.

На отладочной плате размещены 16 портов ввода/вывода. Каждый универсальный порт ввода/вывода имеет четыре 32-разрядных регистра конфигурации, два 32-разрядных регистра данных, 32-разрядный регистра установки/сброса, 32-разрядный регистр блокировки и два 32-разрядных регистра выбора альтернативных функций.

Каждому порту общего назначения можно программно назначить режим работы: цифровой вход, цифровой выход с подтяжкой к нулю или единице, аналоговый и альтернативный. Также у каждого порта можно установить скорость работы: 400 кГц, 2 МГц, 10 МГц и 40 МГц.

12-разрядный АЦП представляет собой аналого-цифровой преобразователь последовательного приближения (поразрядным уравниванием). Он имеет до 19 мультиплексированных каналов, что позволяет измерять сигналы от 16 внешних источников, двух внутренних источников и канала VBAT.

АЦ преобразование каналов может выполняться в разовом, непрерывном, сканирующем или прерывистом режиме. Результат преобразования сохраняется в 16-разрядном регистре данных с левым или правым выравниванием.

В основе работы такого класса АЦП лежит принцип дихотомии, т.е. последовательного сравнения измеряемой величины с половиной, четвертью, восьмой частью и т.д. от максимального значения опорного напряжения. Это позволяет для 12-разрядного АЦП последовательного приближения выполнить весь процесс преобразования за 12 последовательных шагов (итераций) вместо 23 при использовании последовательного счета и получить существенный выигрыш в быстродействии.

### 2.3 Теоретический расчёт пределов измерения

Расчёт основан на аналитическом методе подбора ферритового сердечника для датчика тока на основе эффекта Холла. Формулы взяты из американского журнала «Магнетизм» [14, с 4.12].

Начнём со второй формулы:

$$l_g = l_e \left( \frac{1}{\mu_e} - \frac{1}{\mu_i} \right) (0.3937), \quad (2.5)$$

где,  $l_g$  – длина зазора в ферритовом сердечнике;

$l_e$  – длина магнитного пути магнитопровода;

$\mu_e$  – эффективная проницаемость;

$\mu_i$  – начальная магнитная проницаемость;

0.3937 – коэффициент для перевода сантиметров в дюймы.

Из этой формулы необходимо выразить эффективную магнитную проницаемость:

$$\mu_e = \frac{l_e \cdot \mu_i \cdot (0.3937)}{l_g \cdot \mu_i + l_e \cdot (0.3937)} \quad (2.6)$$

Длину (высоту) зазора можно определить штангенциркулем, она составит  $l_g = 1.7$  мм. Для использования её значения в формуле (2.6) переведём  $l_g$  в дюймы:  $l_g = 0.067$  дюймов. Длину магнитного пути магнитопровода найдём по формуле 2.7:

$$l_e = 2\pi \cdot \left( \frac{R-r}{2} + r \right), \quad (2.7)$$

где, R – радиус внешнего кольца магнитопровода;

r – радиус внутреннего кольца магнитопровода.

Графически это выглядит так (рисунок 2.9):

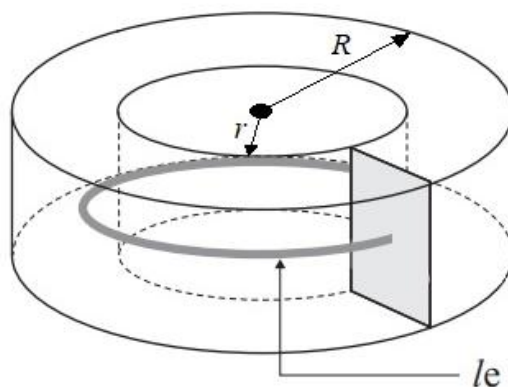


Рисунок 2.9 – Графическое пояснение формулы (2.7)

После расчёта длина магнитного пути составит  $l_e = 4.71$  см.

Начальная магнитная проницаемость находится из спецификации на магнитопровод и составляет  $\mu_i = 1000$ . Подставляя все найденные величины в формулу (2.6), получим:

$$\mu_e = \frac{4.71 \cdot 1000 \cdot 0.3937}{0.067 \cdot 1000 + 4.71 \cdot 0.3937} = 26.93$$

Имея значение  $\mu_e$ , можно использовать первую формулу:

$$\mu_e = \frac{B \cdot l_e}{4\pi \cdot N \cdot I'} \quad (2.8)$$

где, В – магнитная индукция;

N – количество витков исследуемого проводника вокруг катушки;

I – ток, протекающий по исследуемому проводнику.

Из этой формулы выразим ток, тогда:

$$I = \frac{B \cdot l_e}{4\pi \cdot N \cdot \mu_e} \quad (2.9)$$

N примем равной единице, так как через магнитопровод будет проходить только 1 проводник. Теперь, имея данные о пределе измерения датчиком величины магнитной индукции (таблица 2.1), определим пределы измерения тока:

Пусть магнитная индукция равна В = 1000 Гс, тогда:

$$I = \frac{1000 \cdot 4.71}{338.41} = 13.92 \text{ А.}$$

Пусть магнитная индукция равна В = 1 Гс, тогда:

$$I = \frac{1 \cdot 4.71}{338.41} = 0.0139 \text{ А.}$$

Таким образом, получили теоретические пределы измерения тока датчиком Холла.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

## РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДАТЧИКА

### 3.1 Общие сведения

Обозначение и наименование программы

Наименование: Программный компонент «Current Measuring». Версия 1.0.

Обозначение: «Current Measuring», версия 1.0

Программный компонент имеет следующие атрибуты:

- Наименование исполняемого файла – CurrentMeasuring.eww;
- Размер исполняемого файла – 161 байт;
- Версия файла 1.0;
- Версия продукта – 8.20.2
- Номер лицензии – 7117-595-415-8531;
- Внутреннее имя – CurrentMeasuring;
- Исходное имя файла – CurrentMeasuring.eww;
- Язык – English;

Основным необходимым требованием для функционирования программного компонента «CurrentMeasuring» является наличие пакета программ «IAR Embedded Workbench» (не ниже 8.20.2). Функционирование программного обеспечения тестировалось в следующих операционных системах: Windows 7 Service Pack 3, Windows 10.

Язык программирования

Программный компонент «CurrentMeasuring» реализован на языке программирования C++.

Язык C++ - это компилируемый, статически типизированный язык программирования общего назначения. Он поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык содержит обширную стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения,

									Лист
									38
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ 12.03.01. 2018.114-288 ВКР				

поддержку многопоточности и многие другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. Основной уклон языка сделан в сторону объектно-ориентированного и обобщённого программирования.

### 3.2 Функциональное назначение

Программный компонент «CurrentMeasuring» создан для преобразования напряжения, получаемого с датчика Холла, в ток.

#### Функциональные ограничения

Данный программный компонент используется только для датчика тока на основе датчика Холла с пределами измерения: 1...10 А постоянного тока и 1...10 А переменного тока.

### 3.3 Требования к архитектуре программы

Архитектура программы должна быть представлена в виде UML диаграмм в пакете StarUML.

StarUML – это проект с открытым кодом для разработки быстрых, гибких, расширяемых, функциональных и, главное, распространяемых бесплатно платформ UML/MDA для 32-разрядных систем Windows. Цель проекта StartUML – создание универсальной бесплатной платформы для моделирования, которая послужит аналогом для таких коммерческих проектов, как Rational Rose, Together и других.

StarUML имеет простую и мощную архитектуру с поддержкой плагинов, так что любой имеет возможность принять участие в расширении функций утилиты, разработав и подключив собственный модуль, используя СОМ-совместимые языки (C++, Delphi, C#, VB, ...). Это дает платформе много большие перспективы развития, нежели ее коммерческим аналогам.

Из данного требования был сделан вывод о том, что разрабатывать UML диаграмму классов нашего программного обеспечения мы будем в пакете StarUML. Ключевым понятием языка C++ является класс. Класс - это определяемый пользователем тип. Классы обеспечивают упрятывание данных, их

						ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата			39

инициализацию, неявное преобразование пользовательских типов, динамическое задание типов, контролируемое пользователем управление памятью и средства для перегрузки операций [C++].

Ниже опишем теорию разработки диаграммы классов.

UML Диаграмма классов – это диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними.

Существуют два вида диаграммы классов:

- Статический вид диаграммы рассматривает логические взаимосвязи классов между собой;
- Аналитический вид диаграммы рассматривает общий вид и взаимосвязи классов, входящих в систему.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

- Концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
- Точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
- Точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования);

В нашей работе мы будем основываться на статическом виде диаграмм класса, и будем использовать точку зрения реализации, так как мы разрабатываем программное обеспечение.

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

- В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы;
- Посередине располагаются поля (атрибуты) класса. Они выровнены по левому краю и начинаются с маленькой буквы;

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

- Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы.

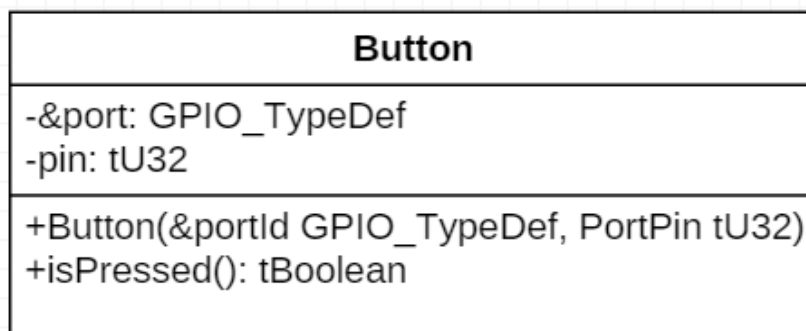


Рисунок 3.1 – Пример диаграммы класса, класс Button

Для задания видимости членов класса используются определенные символы, которые размещены перед именем данного члена класса:

Таблица 3.1 – Таблица видимости членов класса

+	Публичный (Public)
-	Приватный (Private)
#	Защищённый (Protected)
/	Производный (Derived) (может быть совмещён с
~	Пакет (Package)

Взаимосвязь – это особый тип логических отношений между сущностями, показанных на диаграмме классов и объектов.

Взаимосвязи объектов классов:

- Зависимость;
- Ассоциация;
- Агрегация;
- Композиция.

Теперь рассмотрим подробнее каждую взаимосвязь объекта класса.



Зависимость – обозначает такое отношение между классами, что изменение спецификации класса-поставщика может повлиять на работу зависимого класса, но не наоборот.

Ассоциация – показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому.

Агрегация – это разновидность ассоциации при отношении между целым и его частями. Как тип ассоциации агрегация может быть именованной. Одно отношение агрегации не может включать более двух классов (контейнер и содержимое).

Графически агрегация представляется пустым ромбом на блоке класса, и линией, идущей от этого ромба к содержащемуся классу.



Рисунок 3.2 – Диаграмма классов, показывающая агрегацию между двумя классами

Грубо говоря, общий класс «Group» содержит в себе класс «Student», т.е. играет роль контейнера для класса «Student». Агрегация является нестрогой разновидностью взаимоотношений между классами. При удалении класса «Group» класс «Student» продолжит функционировать.

Композиция – более строгий вариант агрегации.

Графически представляется, как и агрегация, но с закрашенным ромбиком. Здесь в качестве примера можно представить взаимосвязь классов «Квартира» и «Комната», где первый класс является контейнером. Видно, что при удалении контейнера, класс «Комната» не сможет существовать сам по себе и тоже удалится.

Взаимосвязи классов:

- Обобщение;
- Реализация.

Теперь рассмотрим поподробнее каждую взаимосвязь объекта класса.

Обобщение (наследование) показывает, что один из двух связанных классов (подтип) является частной формой другого (надтипа), который называется обобщением первого. На практике это означает, что любой экземпляр подтипа является также экземпляром надтипа.

Графически обобщение представляется линией с пустым треугольником у супертипа.

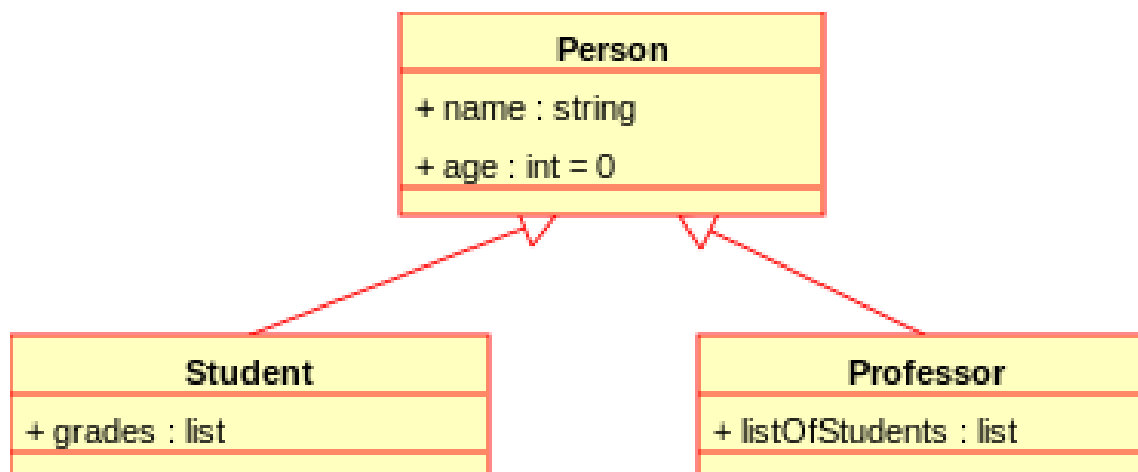


Рисунок 3.3 – Диаграмма классов, показывающая наследование двух подклассов от одного суперкласса

Реализация — отношение между двумя элементами модели, в котором один элемент (клиент) реализует поведение, заданное другим (поставщиком). Реализация — отношение целое-часть. Графически реализация представляется так же, как и наследование, но с пунктирной линией.

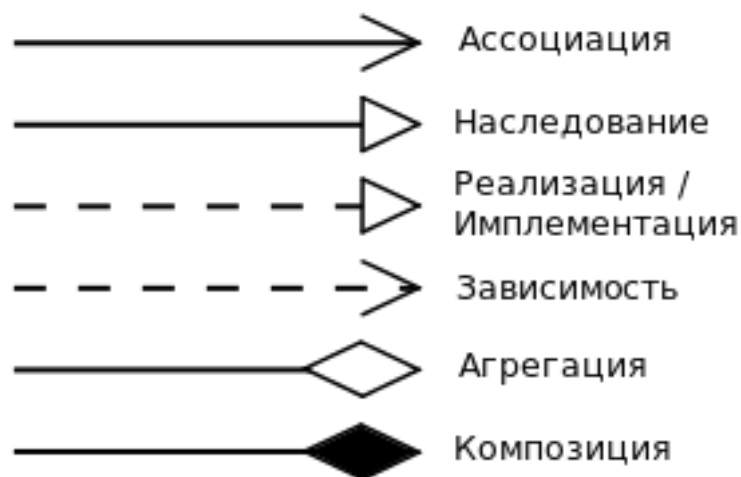


Рисунок 3.4 – Нотация UML для отображения взаимосвязей между классами на диаграммах

На рисунке 3.4 показаны все взаимосвязи между классами, которые используются на диаграммах. Данная информация будет использована в ходе выполнения работы, а в частности разработки архитектуры программного обеспечения.

При разработке должна использоваться Операционная Система Реального Времени FreeRTOS.

FreeRTOS — многозадачная операционная система реального времени (ОСРВ) для встраиваемых систем. Портитована на 35 микропроцессорных архитектур. Распространяется под модифицированной лицензией GPL с исключением, позволяющим разработчику присвоить модифицированный код операционной системы.

Операционная система реального времени, ОСРВ (англ. real-time operating system, RTOS) — тип операционной системы, основное назначение которой — предоставление необходимого и достаточного набора функций для работы систем реального времени на конкретном аппаратном оборудовании.

Основной задачей в таких системах является своевременность выполнения обработки данных и обеспечение реакции на определенное действие за отведенный квант времени.

### 3.4 Общая структура программы

Общая архитектура классов будет показывать только взаимосвязь между всеми классами, и давать понимания, какие классы как зависят от других, и как происходит работа программы.

Разработка архитектуры производилась в программе Star-UML, архитектура показана на рисунке 3.5.

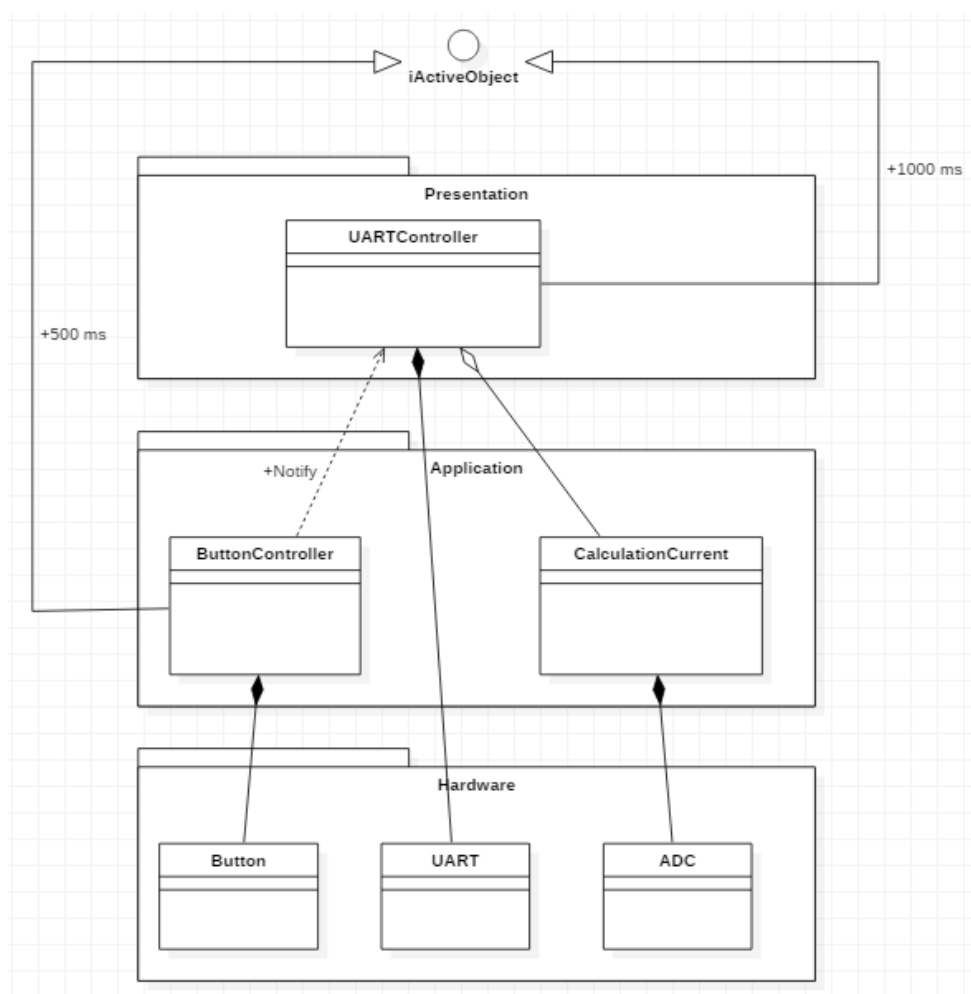


Рисунок 3.5 – Общая архитектура программы

Описание классов:

Класс ADC отвечает за управление АЦП – осуществляет включение, запуск на преобразование, получение преобразованного значения.

Класс UART отвечает за управление интерфейсом передачи данных – осуществляется включение и настройка интерфейса, реализована функция для передачи данных на периферию.

Класс Button отвечает за инициализацию кнопки. В классе задаются порт и пин кнопки, осуществляется проверка нажатия кнопки.

Класс CalculationCurrent содержит в себе функцию, реализованную в виде метода, для пересчёта данных полученных с АЦП в значения тока.

Класс ButtonController – активный класс, отвечающий за управление кнопкой. Он нотифицирует (ставит в известность) необходимые задачи о факте нажатия на кнопку.

Класс UARTController – активный класс. Использован для передачи значений тока из класса «CalculationCurrent» в класс «UART». По нажатию кнопки переключает режим измерения тока в классе «CalculationCurrent».

### 3.5 Архитектура класса ADC

На рисунке 3.6 показана архитектура класса ADC.

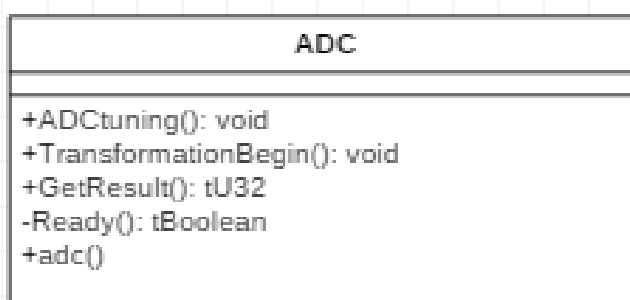


Рисунок 3.6 – Архитектура класса ADC

ADCtuning() – публичный метод, в котором происходит настройка АЦП: подача тактирования, режим работы, выбор канала, с которого будет считываться аналоговая величина, а также включение самого модуля АЦП.

TransformationBegin() – публичный метод, отвечающий за начало преобразования аналоговой величины. Как бы даёт команду «старт» для АЦП.

GetResult() – публичный метод, возвращающий оцифрованное значение аналоговой величины из буфера DR.

Ready() – приватный метод, используется методом GetResult() для получения информации о заполнении буфера DR.

adc() – конструктор.

### 3.6 Архитектура класса UART

На рисунке 3.7 показана архитектура класса UART.

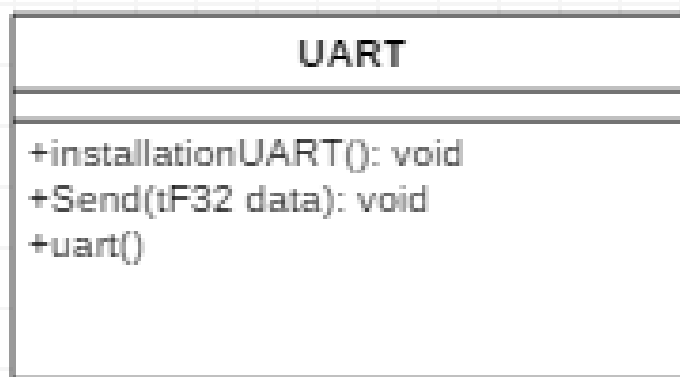


Рисунок 3.7 – Архитектура класса UART

installationUART() – публичный метод, в котором происходит настройка интерфейса UART: подаётся тактирование на модуль, устанавливается скорость передачи данных, включается модуль и устанавливается режим передачи данных.

Send(tF32 data) – публичный метод, отвечающий за отправку данных на периферию.

uart() – конструктор.

### 3.7 Архитектура класса Button

На рисунке 3.8 показана архитектура класса Button.

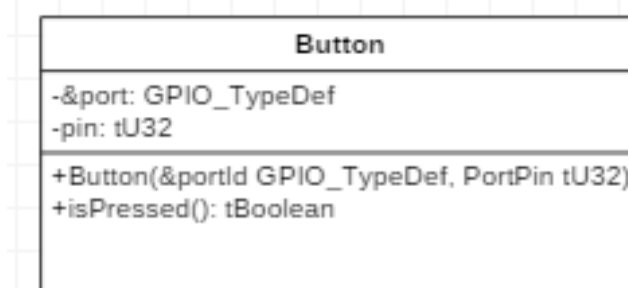


Рисунок 3.8 – Архитектура класса Button

`&port` – атрибут класса `Button`, отвечающий за назначение порта кнопки.

`pin` – атрибут класса `Button`, отвечающий за назначение пина кнопки.

`Button(&portId GPIO_TypeDef, PortPin tU32)` – конструктор, инициализирующий порт и пин кнопки.

`isPressed()` – метод, определяющий нажатие кнопки.

### 3.8 Архитектура класса `ButtonController`

На рисунке 3.9 показана архитектура класса `ButtonController`.

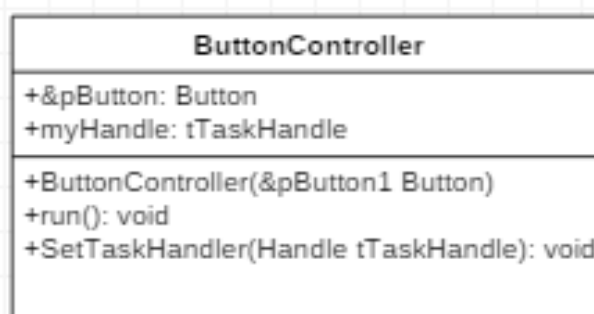


Рисунок 3.9 – Архитектура класса `ButtonController`

`&pButton` – объект класса `Button`.

`myHandle` – указатель на задачи, которые нужно оповестить.

`ButtonController(&pButton1 Button)` – конструктор, инициализирующий кнопку.

`run()` – метод, оповещающий задачи, если кнопка нажата.

`SetTaskHandler(Handle tTaskHandle)` – метод, создающий задачу.

### 3.9 Архитектура класса `CalculationCurrent`

На рисунке 3.10 показана архитектура класса `CalculationCurrent`.

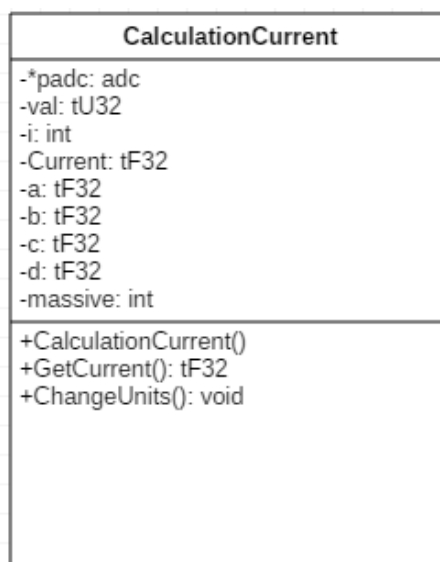


Рисунок 3.10 – Архитектура класса CalculationCurrent

\*padc – объект класса ADC.

val – целочисленная переменная, необходимая для хранения значения кода из АЦП.

i – переменная, которую можно использовать для счёта.

Current - дробная переменная, необходимая для хранения значения тока.

a – первый коэффициент функции преобразования кода в величину постоянного тока.

b – второй коэффициент функции преобразования кода в величину постоянного тока.

c – первый коэффициент функции преобразования кода в величину переменного тока.

d - второй коэффициент функции преобразования кода в величину переменного тока.

massive – массив из целых чисел, заполняется значениями из АЦП.

CalculationCurrent() – конструктор, в котором создаётся объект класса ADC и происходит настройка АЦП, также производится настройка таймера в режиме счётчика: подаётся тактирование, разрешается прерывание таймера по переполнению, устанавливается источник генерации по переполнению, задаётся предел счёта, устанавливается делитель частоты, счётчик включается.



GetCurrent() – метод, возвращающий значение тока. В зависимости от состояния кнопки возвращает значение постоянного либо переменного тока.

ChangeUnits() – метод, хранящий переменную *i*. В случае нажатия кнопки увеличивает значение переменной на 1. Если значение переменной становится равным 2, то она обнуляется.

### 3.10 Архитектура класса UARTController

На рисунке 3.11 показана архитектура класса UARTController.

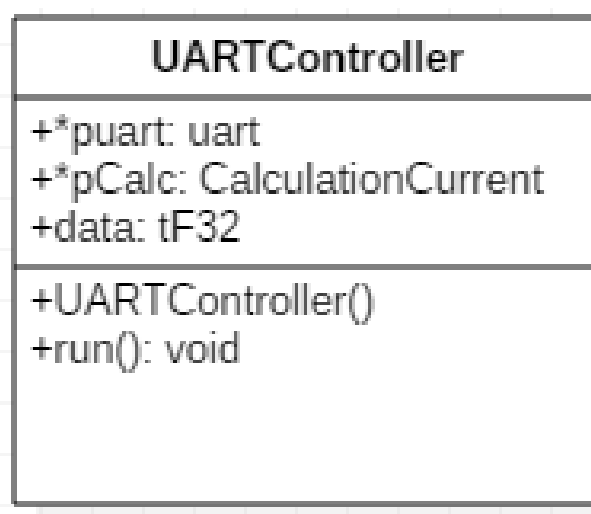


Рисунок 3.11 – Архитектура класса UARTController

\*puart – объект класса UART.

\*pCalc – объект класса CalculationCurrent.

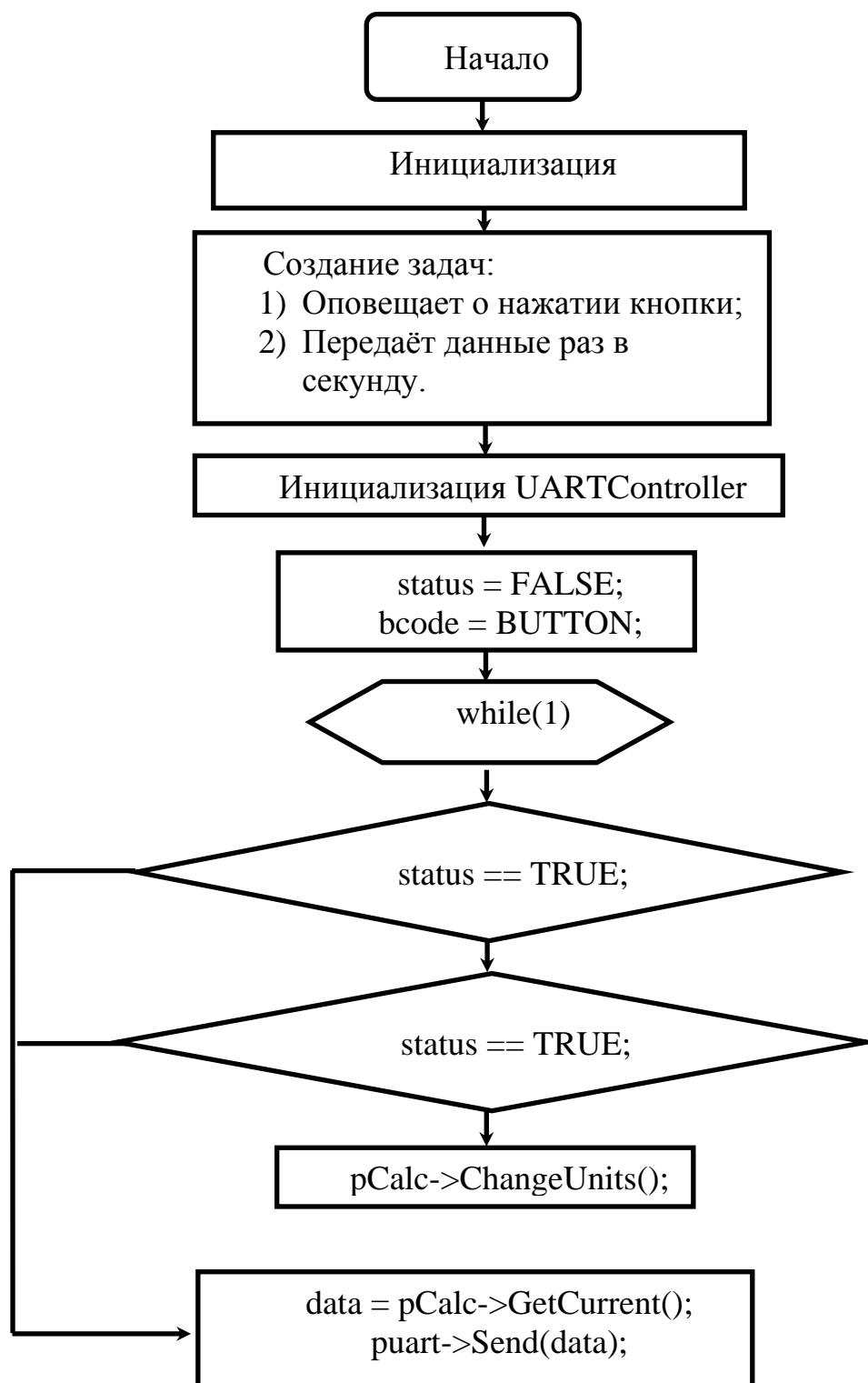
data – дробная переменная, созданная для хранения значения тока.

UARTController() – конструктор, в котором создаются объекты классов UART() и CalculationCurrent(), а также происходит настройка UART.

run() – метод, посылающий значения тока каждую секунду в интерфейс UART. Принимает нотификацию от класса ButtonController о нажатии кнопки, если кнопка нажата, меняется режим измерения тока с режима измерения постоянного тока на режим измерения переменного тока.

### 3.11 Описание логической структуры

Для того чтобы понять как работает данная программа достаточно изобразить блок-схему класса: UARTController. Блок-схема этого класса вместе с инициализацией основной программы изображена на рисунке 3.12.





Значения тока выводятся по интерфейсу UART. Для их отслеживания на компьютере размещена программа Terminal 1.9b. Данные выводятся в виде строки с помощью функции `printf`.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

# ЭКСПЕРИМЕНТАЛЬНОЕ ОПРЕДЕЛЕНИЕ ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ДАТЧИКА

## 4.1 Определение функции преобразования для постоянного тока

В этой главе будет построена математическая модель датчика. Обычно в качестве такой модели используется регрессионная модель в виде полинома приближения обратной функции преобразования (ОФП). Её коэффициенты определяются с помощью метода наименьших квадратов (МНК), у которого мерой точности является минимальная величина суммы квадратов отклонений полинома приближения от ОФП, т.е. мерой погрешности служит её среднеквадратическое значение.

Интегральный критерий, лежащий в основе МНК, не обеспечивает нахождение полинома, имеющего минимальную максимальную погрешность приближения. Поэтому необходимо использовать теорию равномерного приближения функций, основанную на критерии минимума максимальной погрешности приближения. При использовании равномерного приближения происходит интерполяция в определённых точках, называемых узлами интерполяции. Сложность такой математической модели обусловлена количеством узлов интерполяции.

Для нахождения оптимальной степени полинома приближения на фоне случайных сигналов необходимо использовать принцип внешнего дополнения. Этот принцип может быть реализован на основе критериев метода группового учёта аргументов (МГУА). Их суть заключается в создании внешнего дополнения путём разделения всех экспериментальных точек на две группы: А и Б. Группа А – обучающаяся (по данным этой группы находится полином приближения по МНК), группа Б – контрольная (по среднеквадратичному отклонению экспериментальных данных этой группы от полинома приближения определяется степень полинома приближения, соответствующая минимуму критерия).

Наиболее эффективными с точки зрения помехоустойчивости моделирования являются критерии  $d^2$ -симметричный критерий регулярности и  $s^2$ -симметричный

									Лист
									54
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ 12.03.01. 2018.114-288 ВКР				

критерий стабильности. Эти критерии и будут использоваться в дальнейшей работе.

Для определения функции преобразования был выбран диапазон токов от 1 А до 10 А. Этот диапазон был разбит на 10 равномерно распределённых точек. В каждой из этих точек было проведено 10 измерений.

Полученные данные приведены на рисунке 4.1.

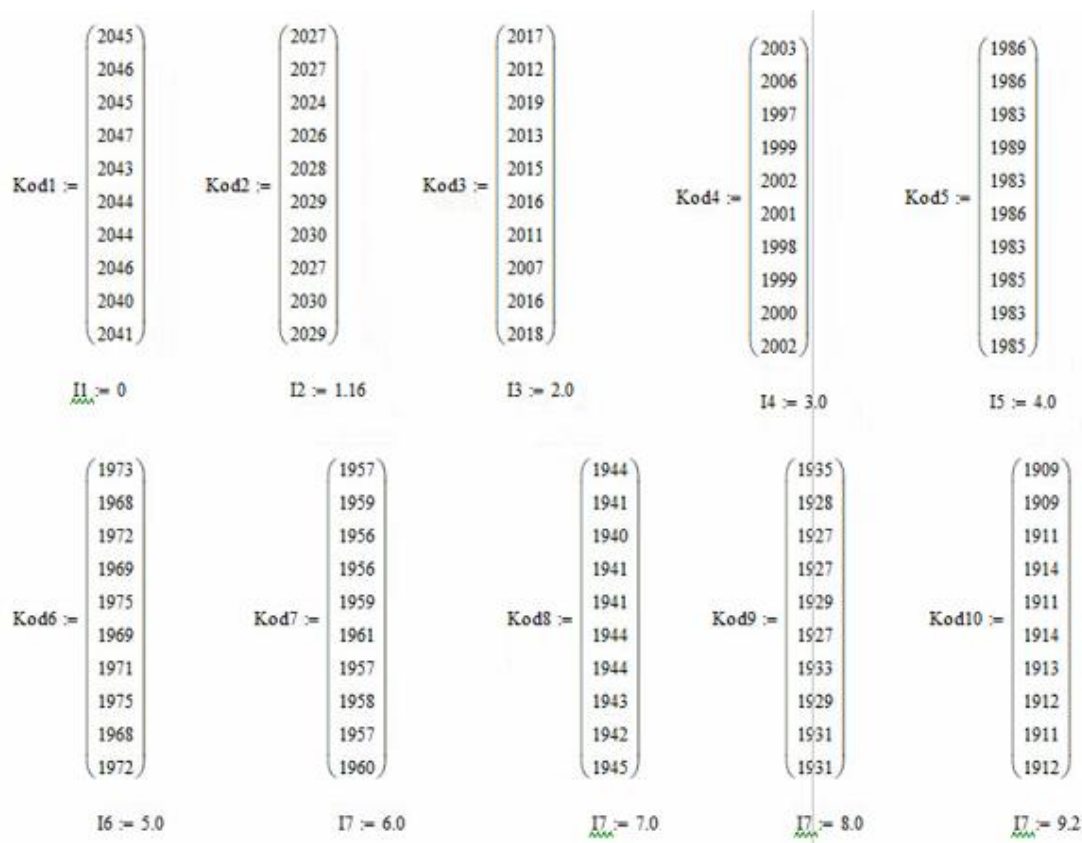


Рисунок 4.1 – Данные, полученные с датчика во время измерения постоянного тока

Для уменьшения погрешности в каждой точке было проведено 10 измерений. Под каждым вектором данных указано значение тока в амперах, при котором были получены эти значения.

Усредним значения кода в каждой точке и запишем соответствующие им значения тока в вектор (рисунок 4.2).

AverageKod :=	2044	Current :=	0
	2028		1.16
	2014		2.0
	2001		3.0
	1985		4.0
	1971		5.0
	1958		6.0
	1942		7.0
	1930		8.0
	1912		9.2

Рисунок 4.2 – Усреднённые значения кода и соответствующие им значения тока

Представим графически функцию преобразования АЦП, проведя через крайние точки диапазона измерений прямую, (рисунок 4.3).

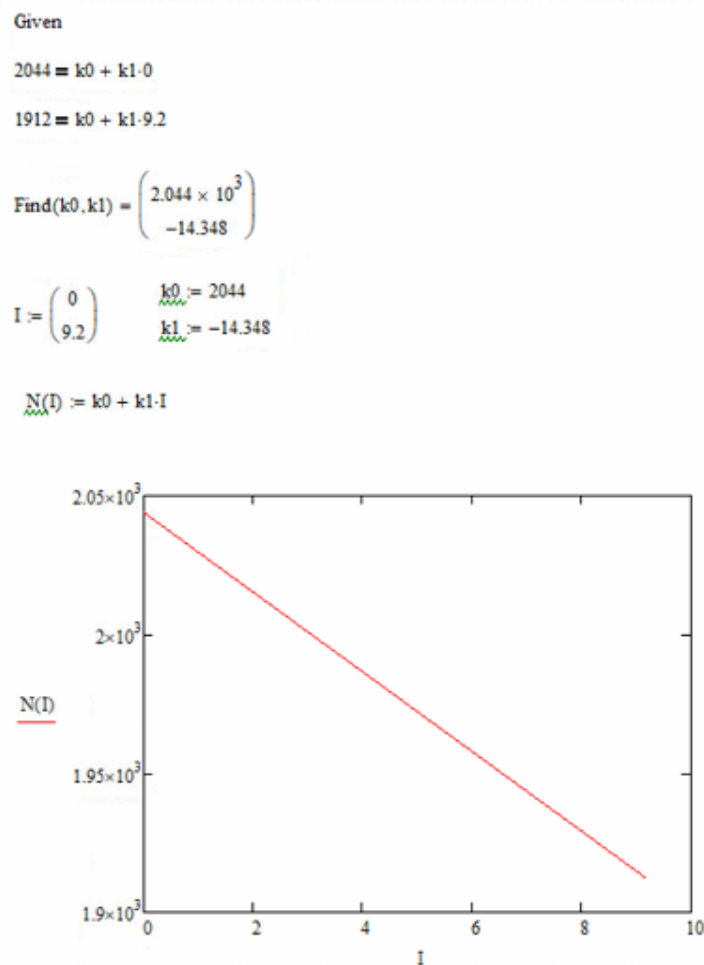


Рисунок 4.3 – Графическое представление функции преобразования

Приведём результаты работы функции преобразования (рисунок 4.4).

$N(0) = 2.044 \times 10^3$	$N(6.0) = 1.958 \times 10^3$
$N(1.16) = 2.027 \times 10^3$	$N(7.0) = 1.944 \times 10^3$
$N(2.0) = 2.015 \times 10^3$	$N(8.0) = 1.929 \times 10^3$
$N(3.0) = 2.001 \times 10^3$	$N(9.2) = 1.912 \times 10^3$
$N(4.0) = 1.987 \times 10^3$	
$N(5.0) = 1.972 \times 10^3$	

Рисунок 4.4 – Подстановка значений тока в функцию преобразования

Вычтем значения, полученные аналитической функцией преобразования, из экспериментальных значений тока и построим график погрешности линейности (рисунок 4.5).

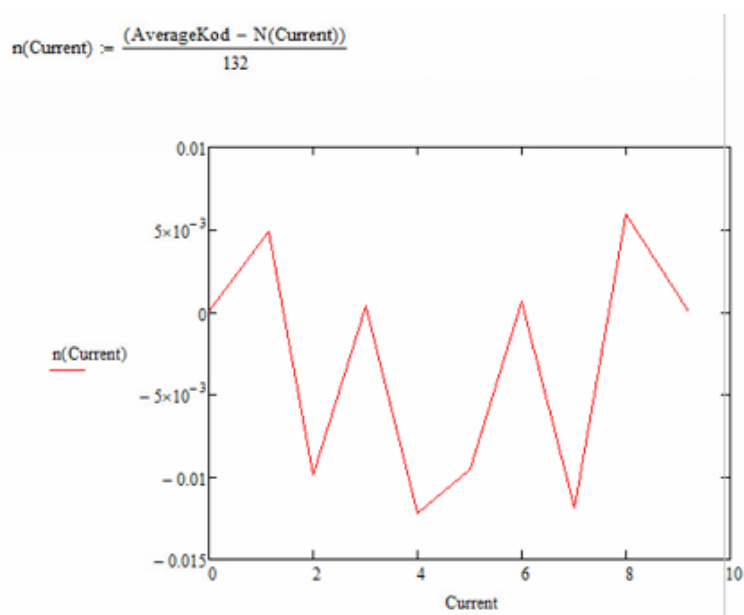


Рисунок 4.5 – График погрешности линейности

Можно заметить, что наибольшее значение погрешности получено ближе к середине диапазона.

Изм.	Лист	№ докум.	Подпись	Дата



Рассчитываем критерии  $d^2$  и  $s^2$  метода группового учета аргументов и определяем по ним максимальную степень полинома приближения ОФП (обратной функции преобразования) датчика (рисунок 4.6).

```

i := 0..4   j := 0..4   k := 0..9

y1_i := n(Current)_2,i   y2_j := n(Current)_2,j+1

x1_i := Current_i,2      x2_j := Current_j,2+1

p := 0..4

z1 := regress(x1,y1,4)
a1 := submatrix(z1,3,length(z1) - 1,0,0)

z2 := regress(x2,y2,4)
a2 := submatrix(z2,3,length(z2) - 1,0,0)

g5_j := sum_p [a1_p * (x2_j)^p]   g6_i := sum_p [a2_p * (x1_i)^p]

g7_k := sum_p [a1_p * (Current_k)^p]   g8_k := sum_p [a2_p * (Current_k)^p]

d2 := [sum_j (y2_j - g5_j)^2] + sum_i (y1_i - g6_i)^2   s2 := [sum_k (n(Current)_k - g7_k)^2] + sum_k (n(Current)_k - g8_k)^2

d2 = 1.499 x 10^-3   s2 = 1.499 x 10^-3

n0 := (1, 2, 3, 4, 5)^T   d20 := (7.315 x 10^-4, 1.402 x 10^-3, 1.267 x 10^-3, 1.499 x 10^-3, nn)^T   s2 := (1.075 x 10^-3, 1.469 x 10^-3, 1.281 x 10^-3, 1.499 x 10^-3, nn)^T

n1 := 1   d20 := 7.315 x 10^-4   s2 := 1.075 x 10^-3

```

Рисунок 4.6 – Применение методов  $d^2$  и  $s^2$  группового учета аргументов

В ходе расчётов были выбраны коэффициенты  $d^2 = 7,315e-4$  и  $s^2 = 1,075e-3$ , соответствующие первой степени полинома. Найдём коэффициенты ОФП (рисунок 4.7).

```
z := regress(AverageKod, Current, 1)
a := submatrix(z, 3, length(z) - 1, 0, 0)
```

$$a = \begin{pmatrix} 142.407973 \times 10^0 \\ -69.685101 \times 10^{-3} \end{pmatrix}$$

Рисунок 4.7 – Коэффициенты обратной функции преобразования  
 Проверим работу обратной функции преобразования датчика (рисунок 4.8).

$$\text{Cur}(x) := \sum_{i=0}^1 (a_i \cdot x^i)$$

$$\text{Cur}(\text{AverageKod}_0) = -0.028$$

	0
0	-0.028
1	1.087
2	2.062
3	2.968
Cur(AverageKod) =	4 4.083
	5 5.059
	6 5.965
	7 7.08
	8 7.916
	9 9.17

Рисунок 4.8 – Показания обратной функции преобразования датчика  
 Видно, что значения в столбце примерно совпадают с экспериментальными значениями тока в векторе «Current». Найдём относительную погрешность ОФП в этих точках (рисунок 4.9).

$$Dt := \left( \frac{\text{Current} - \text{Cur}(\text{AverageKod})}{92} \right) \cdot 100$$

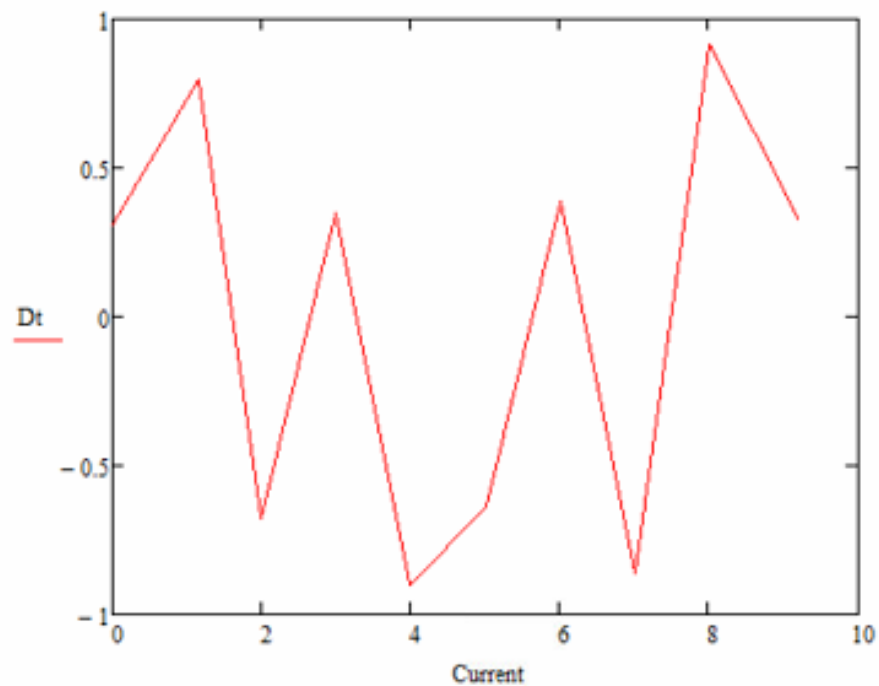


Рисунок 4.9 – График относительной погрешности ОФП

По графику можно сказать, что относительная погрешность  $\delta$  не превышает значения 1 %.

В качестве датчика сигнала постоянного тока в этом эксперименте использовался импульсный источник питания Mastech HY3030E (рисунок 4.10).



Рисунок 4.10 – Источник питания

В качестве нагрузки использовалась группа резисторов.

## ЗАКЛЮЧЕНИЕ

В ходе работы над выпускной квалификационной работой удалось решить следующие задачи:

1. Изучить принцип действия датчика Холла и провести аналитический обзор имеющихся измерительных приборов на его основе.
2. Произвести теоретический расчёт пределов измерения датчиком Холла.
3. Разработать программное обеспечения для получения данных с датчика, и их преобразования в значение силы тока.
4. Экспериментально установить функцию преобразования датчика.

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Адвоченко, Б.И. Датчик слабых магнитных полей на основе эффекта Холла / Б.И. Адвоченко, Г.Ф. Карлова, А.М. Цырендоржиева // Электронные средства и системы управления. Серия «Интеллектуальная силовая электроника и преобразовательная техника». – 2017. – Вып. 1(1). – С. 228-230.
2. Бакытбек, Е.Б. Датчики для измерения слабых магнитных полей / Е.Б. Бакытбек // Наука и образование сегодня. Серия «Технические науки». – 2016. – № 4(5). – С. 30-31.
3. Бардин, А.А. Холловский градиометр / А.А. Бардин, С.В. Перченко // Инженерный вестник Дона. Серия «Приборостроение». – 2014. – Вып. 31. № 4(1). – С. 110-116.
4. Быковский, Ю.М. Разработка метода оценки параметров магнитного поля с помощью датчика Холла / Ю.М. Быковский, В.В. Левченко // Водный транспорт. Серия «Судоходство, безопасность, технологии, подготовка». – 2012. – Вып. 1(13). – С. 75-82.
5. Виглеб, Г. Датчики / Герхард Виглеб; пер. с нем. М. Хацернова. – М.: Мир, 1989. – 196 с.
6. ГОСТ Р 51086-97. Датчики и преобразователи физических величин электронные. Термины и определения, – М.: Электрстандарт, 1998. – 13 с.
7. ГОСТ 8.009-84. Нормируемые метрологические характеристики средств измерений. – М.: Изд-во стандартов, 1986. – 27 с.
8. Грановский, В.А. Динамические измерения: Основы метрологического обеспечения / В.А. Грановский. – Л.: Энергоатомиздат, 1984. – 224 с.
9. Ковальчук, Т.В. Изучение эффекта Холла в полупроводниках / Т.В. Ковальчук, К.А. Царуш, С.М. Ожегова // Наука и производство урала. Серия «Совершенные технологии в науке и образовании». – 2017. Вып.13. – С. 108-111.
10. Крыжанов, Е.Э. Трехмерный датчик магнитного поля на основе эффекта Холла / Е.Э. Крыжанов, А.Н. Молчанов, Е.П. Поздеева // Проблемы внедрения

						ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата			62

результатов инновационных разработок: сб. науч. тр. – Уфа: АЭТЕРНА, 2017. – 199 с.

11. Сойка, А.К. Измерение сильных импульсных магнитных полей датчиками Холла / А.К. Сойка, И.О. Сологуб // Приборы и техника эксперимента. Серия «Общая экспериментальная техника». – 2010. – Вып. 1. – С. 132-134.

12. Датчики: справочное пособие / под общ. ред. В.М. Шарапова, Е.С. Полищука. – М.: Техносфера, 2012. – 624 с.

13. Фрайден, Дж. Современные датчики: справочник / Джекоб Фрайден; пер. с англ. Ю. Заболотной; под ред. Е. Свинцова. – М.: Техносфера, 2006. – 588 с.

14. [http://www.lodestonepacific.com/distrib/pdfs/Magnetics/Design\\_Application\\_Notes.pdf](http://www.lodestonepacific.com/distrib/pdfs/Magnetics/Design_Application_Notes.pdf)

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

## ПРИЛОЖЕНИЕ А

### Листинг программы

```
/*
*****
* FILENAME: main.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****
*****/

#include <stm32f411xe.h>
#include "frtoswrapper.h"
#include "application.hpp"
#include "ADC.h"
#include "UART.h"
uint32_t SystemCoreClock = 16'000'000U;

extern "C" {
int __low_level_init(void) {
//Switch on external 16 MHz oscillator
RCC->CR |= RCC_CR_HSION;
while ((RCC->CR & RCC_CR_HSIRDY) != RCC_CR_HSIRDY) {
}
//Switich system clock on extrenal oscilator
RCC->CFGR |= RCC_CFGR_SW_HSI;
while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_HSI) {
}

//GPIO's installation
RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN |
RCC_AHB1ENR_GPIOCEN; // GPIOA timing
GPIOA->MODER |= GPIO_MODER_MODE0; // output A0 is set to analog
mode
GPIOA->MODER |= GPIO_MODER_MODE2_1; // output A2 is set to
alternative mode
GPIOA->OTYPER &=~ GPIO_OTYPER_OT2; // push-pull mode
GPIOA->OSPEEDR |= GPIO_OSPEEDR_OSPEED2_1; // high speed
GPIOA->AFR[0] |= 0x07 << 8;

GPIOC->MODER &=~ (GPIO_MODER_MODE13_0);
```

					ИОУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

```

    NVIC_EnableIRQ(TIM2_IRQn);

    return 1;
}
}

```

```

cRTOS Rtos;
Application App(Rtos);

```

```

int main(void) {
    App.run();
}

```

### Класс Application

```

/*****
*****
* FILENAME: Application.h
*
* DESCRIPTION: Main Application for all classes
*
* Copyright (c) 2018 by South Ural State University
*****
*****/

#ifndef APPLICATION_H
#define APPLICATION_H
#include "ButtonController.h"
#include "UARTController.h"
#include "frtoswrapper.h"

class Application
{
public:
    Application(cRTOS &os): Rtos(os) {};
    void run();
private:
    UARTController Ua;
    Button button{*GPIOC, 13};
    ButtonController Bc{button};
    cRTOS &Rtos;
};
#endif

/*****

```



```

*****
* FILENAME: Application.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Sergey Kolody
*
*****
*****/
#include "Application.hpp"

constexpr tU16 TaskStackSize =
static_cast<tU16>(configMINIMAL_STACK_SIZE);
constexpr tU32 TaskPriority = 3U;

void Application::run()
{
    Rtos.taskCreate(&Ua, TaskStackSize, TaskPriority, "ButtonController");
    Bc.SetTaskHandler(Ua.taskHandle);
    Rtos.taskCreate(&Bc, TaskStackSize, TaskPriority, "UARTController");
    Rtos.startScheduler();
}

```

### Класс Button

```

/*****
*****
* FILENAME: Button.hpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****
*****/

#ifndef BUTTON_H
#define BUTTON_H
#include <stm32f411xe.h>
#include "susutypes.h"
#include "utils.hpp"

class Button
{
public:
    Button(GPIO_TypeDef &portId, tU32 portPin): port(portId), pin(portPin){}

```

```
tBoolean isPressed()
{
return!(utils::checkBit(port.IDR, pin));
}
```

```
private:
GPIO_TypeDef &port;
tU32 pin;
};
#endif
```

### Класс ADC

```

/*****
*****
* FILENAME: ADC.h
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****/
*****/
```

```
#ifndef ADC_H
#define ADC_H
#include <stm32f411xe.h>
#include "susutypes.h"
```

```
class adc
{
public:
void ADCTuning();
void TransformationBegin();
tU32 GetResult() const;
tBoolean Ready() const ;
adc();
};
#endif
```

```

/*****
*****/
```

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

```

*****
* FILENAME: Button.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****

*****/

#include "ADC.h"

adc::adc()
{
}

void adc::ADCtuning()
{
    RCC->APB2ENR |= RCC_APB2ENR_ADC1EN; // ADC timing
    ADC1->CR1 &=~ ADC_CR1_SCAN; // scanning mode OFF
    ADC1->CR2 |= ADC_CR2_CONT; // enabling continuous conversion
    ADC1->SMPR2 |= ADC_SMPR2_SMP0_2; // set channel sampling time
    ADC1->SQR1 &=~ ADC_SQR1_L; // one channel is scanned
    ADC1->SQR3 &=~ ADC_SQR3_SQ1; // channel A0 is scanned
    ADC1->CR2 |= ADC_CR2_ADON; // ADC ON
}

void adc::TransformationBegin()
{
    ADC1->CR2 |= ADC_CR2_SWSTART;
}

tU32 adc::GetResult() const
{
    if(Ready())
        return (tU32)ADC1->DR;
}

tBoolean adc::Ready() const
{
    return((ADC1->SR & ADC_SR_EOC) == ADC_SR_EOC);
}

```

Класс UART

						ИОУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата			68

```

/*****
*****
* FILENAME: UART.h
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****

*****/

#ifndef UART_H
#define UART_H
#include <stm32f411xe.h>
#include "susutypes.h"

class uart
{
public:
    void installationUART();
    void Send(tF32 data);
    uart();
};
#endif

/*****
*****
* FILENAME: UART.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****

*****/

#include "UART.h"
constexpr tU8 length = 17U;
uart::uart()
{
}
void uart::installationUART()
{
    RCC->APB1ENR |= RCC_APB1ENR_USART2EN; // USART timing
    USART2->BRR = 0x683; // BaudRate 9600

```

```

USART2->CR1 |= USART_CR1_UE; // USART ON
USART2->CR1 |= USART_CR1_TE; // Transmitter operation enabled
}

void uart::Send(tF32 data)
{
    char res [length];
    sprintf (res, "%s %6.2f\n", "Voltage:", data);
    for(int i = 0; i < length; i++)
    {
        while(!(USART2->SR & USART_SR_TC));
        USART2->DR = res[i];
        for(int i = 0; i < 16000; i++);
    }
    for(int i = 0; i < 160000; i++);
}

```

### Класс ButtonController

```

/*****
*****
* FILENAME: ButtonController.h
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****
*****/

#ifndef BUTTONCONTROLLER_H
#define BUTTONCONTROLLER_H
#include <stm32f411xe.h>
#include "Button.hpp"
#include "frtosWrapper.h"
#include "susutypes.h"

class ButtonController: public iActiveObject
{
public:
    ButtonController(Button& pButton1) : pButton(pButton1){};
    void SetTaskHandler(tTaskHandle Handle) {myHandle = Handle;};
    virtual void run(void);
private:
    tTaskHandle myHandle;

```

					ИОУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

```

    Button &pButton;

};
#endif

/*****
*****
* FILENAME: ButtonController.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****/
*****/
#include "ButtonController.h"
#include <stm32f411xe.h>
#include "frtosWrapper.h"

#define BUTTON_DELAY (tU32) (500/portTICK_PERIOD_MS)

void ButtonController::run()
{
    while(1)
    {
        if(pButton.isPressed()){
            Rtos.taskNotify(myHandle, 1U, eSetValueWithOverwrite);
        }
        Rtos.taskDelay(BUTTON_DELAY);
    }
}

```

### Класс CalculationCurrent

```

/*****
*****
* FILENAME: CalculationCurrent.h
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****/
*****/

#ifndef CALCULATIONCURRENT_H

```

					ИОУрГУ 12.03.01. 2018.114-288 ВКР <span style="float: right;">Лист 71</span>
Изм.	Лист	№ докум.	Подпись	Дата	







```

/*****
*****
* FILENAME: UARTController.h
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****
*****/

#ifndef UARTCONTROLLER_H
#define UARTCONTROLLER_H
#include <stm32f411xe.h>
#include "susutypes.h"
#include "rtosWrapper.h"
#include "UART.h"
#include "CalculationCurrent.h"

class UARTController : public iActiveObject
{
public:
    UARTController();
    void run() override;
private:
    uart* puart;
    CalculationCurrent* pCalc;
    tF32 data;
};
#endif

/*****
*****
* FILENAME: UARTController.cpp
*
* Copyright (c) 2018 by SUSU
* Author: Artyom Sorokin
*
*****
*****/

#include "UARTController.h"

#define BUTTON (tU32)0
#define MEASURE_DELAY (tU32)(1000/portTICK_PERIOD_MS)

```

					ИОУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		74

```

#define TASK_DELAY (tU32)(100/portTICK_PERIOD_MS)

UARTController::UARTController()
{
    this->puart = new uart();
    this->pCalc = new CalculationCurrent();
    puart->installationUART();
}

void UARTController::run()
{
    tBoolean status = FALSE;
    tU32 bcode = BUTTON;

    while(1)
    {
        status = (tBoolean)Rtos.taskNotifyWait((tU32)0, (tU8)ULONG_MAX, &bcode,
TASK_DELAY);
        if(status == TRUE)
        {
            if(bcode == 1U)
            {
                pCalc->ChangeUnits();
            }
        }
        data = pCalc->GetCurrent();
        puart->Send(data);
        Rtos.taskDelay(MEASURE_DELAY);
    }
}

```

					ЮУрГУ 12.03.01. 2018.114-288 ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75