СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc28248391)

[1 ВЫБОР МИКРОКОНТРОЛЛЕРА 6](#_Toc28248392)

[1.1 Основные ядра, используемые в микроконтроллерах 6](#_Toc28248393)

[1.2 Обзор серий микроконтроллеров 8](#_Toc28248394)

[1.3 Выбор модели микроконтроллера 10](#_Toc28248395)

[2 ПРИНЦЫП РАБОТЫ УСТРОЙСТВА 12](#_Toc28248396)

[2.1 Структурная схема электрическая 12](#_Toc28248397)

[2.2 Функциональная схема электрическая 15](#_Toc28248398)

[2.3 Принципиальная схема электрическая 18](#_Toc28248399)

[3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ 21](#_Toc28248400)

[3.1 Структура программного кода 21](#_Toc28248401)

[3.2 Описание режимов программного кода 25](#_Toc28248402)

[ЗАКЛЮЧЕНИЕ 27](#_Toc28248403)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 28](#_Toc28248404)

**ВВЕДЕНИЕ**

В современном мире электроника и различная вычислительная техника играет важную роль в различных областях человеческой деятельности. Это связано с тем, что прогресс не стоит на месте, все повсеместно модернизируется, улучшается, автоматизируется. Человек двигается и с каждым разом перед ним стоят более серьезные вопросы и задачи, которые без применения различных электронных устройств либо попросту невозможны, либо займут слишком много времени для решения той или иной задачи. Все эти, казалось бы, маленькие факты и нюансы достаточно серьезно влияют на скорость и точность получаемых результатов, фактически любых измерений. Кроме того, в последнее время в научно-технической сфере, все чаще стали применяться различные микроконтроллеры, которые позволяют ускорить и упросить достаточно большой спектр задач.

Поскольку практикум и выполнение лабораторных работ является важной частью при подготовке студентов инженерной специальности. А навыки, получаемые в физической лаборатории: понимание наблюдаемых процессов, пользование измерительными приборами, обработка полученных результатов, позволяют студентам применять их в своей будущей трудовой деятельности. Было принято решение о создании универсального блока управления на микроконтроллере. Данное устройство позволяет более детально изучить различные физические явления, связанные с линейными или угловыми перемещениями.[9]

Целью данной работы является разработка функционального блока для управления и обработки измерений линейных и угловых перемещений физических тел с использованием микроконтроллера.

Основные задачи, необходимые для раскрытия цели дипломной работы:

1. Выбор оптимального микроконтроллера, который бы отвечал следующим параметрам:
   1. быстродействие
   2. энергопотребление
   3. габариты
   4. объем памяти
   5. простота обслуживания
   6. стоимость
2. Проектирование структурной электрической схемы.
3. Проектирование функциональной электрической схемы.
4. Проектирование принципиальной электрической схемы.
5. Разработка и написание программного обеспечения.
6. Отладка программного кода.
7. Сборка и регулировка разработанного устройства.

# 1 ВЫБОР МИКРОКОНТРОЛЛЕРА

## 1.1 Основные ядра, используемые в микроконтроллерах

Так как для решение основной задачи дипломной было принято решение об использовании микроконтроллера, данную главу стоит начать с определения данного понятия.

Микроконтроллер – это специальная микросхема, предназначенная для управления различными электронными устройствами. Данное устройство объединяет в себе процессор, память, ПЗУ и периферию внутри одного корпуса, внешне похожего на обычную микросхему.[5]

Поскольку развитие микроэлектроники и её широкое применений в научно-технической сфере, промышленном производстве, в устройствах и системах управления самыми разнообразными объектами привело к тому, что рынок микроконтроллеров стал весьма разнообразным. Было принято решение провести анализ наиболее популярных в данный момент вариантов исполнения данных устройств.

Первым делом были выделены основные разновидности ядер микроконтроллеров, подходящих для выполнения основной задачи данной дипломной работы. Основные вариантами ядер микроконтроллеров являются ядра двух семейств AVR и ARM.

AVR - семейство восьми битных микроконтроллеров. Данные микроконтроллеры являются представителями гарвардской архитектуры, что подразумевает под собой разделение памяти программ и памяти данных. Также стоит отметить, что данное семейство имеет систему команд, близкую к идеологии RISC. Особенностью такого подхода, является упрощение системы команд, как результат конечный пользователь получает выигрыш в производительности и быстродействии. Процессор AVR имеет 32 8-битных регистра общего назначения, объединённых в регистровый файл. Кроме всего прочего, стоит отметить, что система команд микроконтроллеров AVR весьма развита и насчитывает в различных моделях от 90 до 133 различных инструкций. Большинство команд выполняется за 1 такт. Что крайне положительно сказывается на производительности.[10]

ARM — семейство лицензируемых 32-битных и 64-битных микропроцессорных ядер разработки компании ARM Limited. Также, как и семейство вышеупомянутых восьми битных микроконтроллеров AVR, представляет собой микроконтроллеры придерживающихся идеологии RISC для системных команд. Данное семейство микроконтроллеров является одним из лидеров на рынке микроконтроллеров за счет своей простоты и дешевизны. По приведенной статистике, на процессоры ARM приходилось порядка 90 % всех встроенных 32-разрядных процессоров. Поскольку данные процессоры имеют низкое энергопотребление, в связи с этим они находят широкое применение во встраиваемых системах и преобладают на рынке мобильных устройств, для которых данный фактор немаловажен[11].

## 1.2 Обзор серий микроконтроллеров

Разобравшись с видами ядер, применяемых в микроконтроллерах, стоит перейти к выбору конкретной серии модели, которая бы была оптимальным вариантом для выполнения данной дипломной работы.

После детального анализа была обнаружена еще одна популярная серия микроконтроллеров имеющая название PIC. Данная серия микроконтроллеров также, как и микроконтроллеры, построенные на ядрах ARM и AVR занимает достаточно серьезные позиции на рынке микроконтроллеров.

Начать стоит с рассмотрения микроконтроллеров серии PIC. Микроконтроллеры данной серии представляют гарвардскую архитектуру и производятся американской компанией Microchip Technology Inc. Стоит отметить, что под маркой PIC фирмой Microchip выпускаются 8-, 16- и 32-битовые микроконтроллеры и цифровые сигнальные контроллеры (DSC), отличительной особенностью которых является хорошая преемственность различных семейств: программная совместимость, совместимость по выводам, по периферии, по напряжениям питания. Номенклатура насчитывает более 500 различных контроллеров со всевозможными вариациями периферии, отличающимися объёмами памяти, количеством выводов, производительностью, диапазонами напряжений питания, рабочими температурами и др.

Следующим представителем для рассмотрения является серия микроконтроллеров STM8. Данная серия микроконтроллеров производится компанией STMicroelectronics более известной своими микроконтроллерами STM32, являющимися представителем микроконтроллеров построенных на основе 32-битного ядра ARM. Но в данный момент речь пойдет о другой серии данной компании, а именно о микроконтроллере STM8. Данная серия микроконтроллеров представляет собой микроконтроллеры, построенные на ядре собственной разработки компании STMicroelectronics, в основу которого легло ядро семейства ARM. Стоит отметить, что серия STM8 является представителем гарвардской архитектуры и имеют 3-ступенчатый конвейер, который минимизирует время выполнения команд. Максимальная тактовая частота — 24 МГц.[2]

Для представления микроконтроллеров на базе ядра AVR, была выбрана серия микроконтроллеров ATmega. Данная серия микроконтроллеров являются крайне популярным решением в данный момент, этот факт вызван тем, что именно данная серия обладает достаточно хорошими характеристика в сравнении со своими основными конкурентами, а именно серии 8-ми битных микроконтроллеров STM8 и серии PIC16.[1]

## 1.3 Выбор модели микроконтроллера

Таблица 1 - сравнительная характеристика наиболее популярных моделей серии: ATmega, STM8 и PIC16.

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры | Модель микроконтроллера | | |
| ATmega328P-AU | STM8L151C8T6 | PIC16F627A |
| Ядро | AVR | ST | PIC |
| Разрядность | 8 бит | 8 бита | 8 бита |
| Архитектура | RISC | RISC | RISC |
| Тактовая частота, МГц | 20 | 16 | 20 |
| Объем памяти программ, кбайт | 32 | 64 | 3,5 |
| Тип памяти программ | flash | flash | flash |
| Объем EEPROM, кбайт | 1 | 2 | 0.128 |
| Объем RAM, кбайт | 2 | 4 | 0,224 |
| I2C интерфейс | 1 | 1 | - |
| Напряжение питания | 1.8…5.5 в | 1.8…3.6 в | 3…5.5 в |
| Список команд | 90 | 32 | 35 |
| Стоимость, BYN | 4.40 | 4.80 | 3.40 |

Из таблицы 1 можно увидеть, что все микроконтроллеры, приведенные в таблице, являются представителя RISC архитектуры. Кроме того, стоит отметить, что все 3 модели имеют одинаковый тип памяти программ, но в тоже время модель из серии STM8, имеет в два раза больший объем данной памяти, в сравнении с моделью из серии Atmega и гораздо больший объем в сравнении с моделью из серии PIC16. Кроме того в пользу выбора модели из серии STM8, мог бы говорить тот факт, что у данной модели в два раза больше энергонезависимой памяти (EEPROM). Аналогичная ситуация обстоит и с оперативной памятью (RAM). Но стоит заметить, что модель из серии STM8 проигрывает двум оставшимся моделям в тактовой частоте, которая влияет на производительность микроконтроллера. Также немаловажным параметром является количество команд для управления микроконтроллером и в данном случае, бесспорным лидером является представитель серии Atmega. Так же немаловажным фактом при выборе микроконтроллера было наличие I2C интерфейса. В модели серии PIC16 данный интерфейс отсутствует, что является серьезным недостатком данного микроконтроллера.

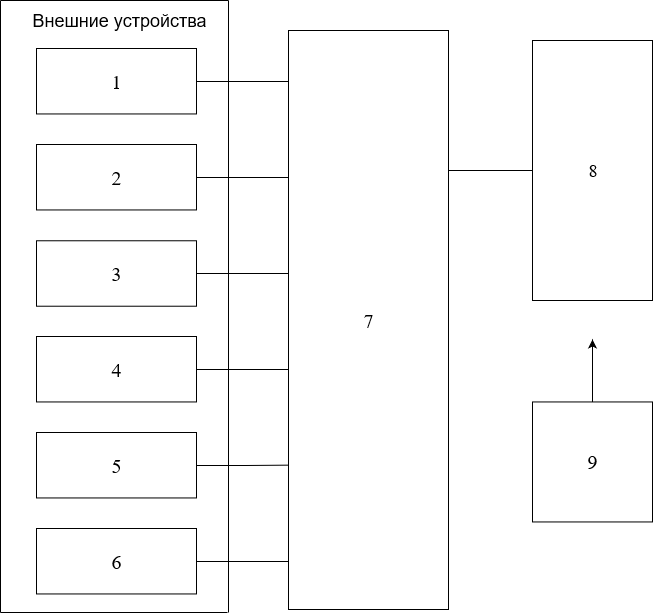
Сравнив 3 наиболее популярные модели, было решено выбирать модель микроконтроллера серии Atmega или же серии STM8. Поскольку модель микроконтроллера серии PIC16 пусть и дешевле, но проигрывает оставшимся двум вариантом почти по всем характеристикам и параметрам.

После более детально рассмотрения двух вышеупомянутых моделей, было принято решение отдать предпочтение модели серии Atmega. Данное решение обосновывается тем фактом, что данная модель выигрывает по быстродействию, а также имеет больший список различной технической литературы, что упрощает программную разработку. Так же немаловажным фактом является простота использования, а как результат, простата дальнейшей поддержки устройства в рабочем состоянии. Стоит отметить, что абсолютна вся документация для модели серии Atmega распространяется по свободной лицензии.

# 2 ПРИНЦЫП РАБОТЫ УСТРОЙСТВА

## 2.1 Структурная схема электрическая

Начиная разработку конечного устройства, которое полностью решало бы задачу, поставленную в данной дипломной работе, была составлена структурная схема готового устройства. Данная структурная схема представлена на рисунке 1.



1-датчик ИК, 2 - датчик подсчета периодов, 3 - датчик линейного перемещения, 4 - датчик углового перемещения влево, 5 - датчик углового перемещения вправо, 6 - блок выбора режима работы, 7 - блок измерения и управления, 8 - блок индикации, 9 - блок питания 5 В

Рисунок 1 - Структурная схема электрическая

В данной структурной схеме представлены основные функциональные компоненты разрабатываемого устройства. Кроме того, в ней представлены взаимосвязи между основными компонентами данной схема. Стоит отметить, что каждый компонент представленный в данной схеме реализует полностью законченное действие.

Поскольку блок измерения и управления 7 разрабатывается, как многофункциональное устройство, в данном блоке реализуется 4 режимом работы. Далее приведены основные режимы, реализованные на программном уровне:

1. Режим измерения линейного перемещения
2. Режим измерения углового перемещения
3. Режим подсчета периодов колебания
4. Режим выбора языка пользовательского интерфейса

Описание и рассмотрение разработанного и изготовленного блока управления стоит начать с описания работы режима измерения линейного перемещения.

При включении блока управления в сеть, устройство совершает аппаратный сброс. Далее с помощью Блока выбора режима 6 представленного в структурной схеме, происходит установка режима 1 в котором далее будет работать Блок измерения и управления 7.

Первым этапом выполнения данного режима является установка расстояния при помощи блока выбора режима 6, которое должно пройти тело. Задаваемое значение отображается на блоке индикации 8. В тоже время ожидается получение управляющего сигнала с ИК датчика 1 на блок управления и измерения 7. Данный управляющий сигнал является флагом для перехода блока 7 во второй этап выполнения управляющей программы.

Вторым этапом выполнения данного режима является получение и подсчет импульсов, получаемых от датчика линейного перемещения 3. После чего происходить обработка данных измерений на программном уровне. По окончанию линейного перемещения тела, устройство переходит в исходное положения и может быть использовано в любом из возможных режимов работы.

Второй режим реализует измерение и обработку импульсов, получаемых с датчиков углового перемещения 4 и 5, поступающих на соответствующие входы блока управления и измерения 7. Режим работы блока 7 в данном режиме аналогичен режиму работы данного элемента в режиме работы 1. Отличие состоит лишь в программной обработке получаемых значений.

Третий режим позволяет производить измерения периодов механических колебаний различных видом маятников (математический, маятник Обербека и др.). Данный режим реализуется с использованием датчика подсчета периодов 3. При перекрытии датчика 3 стержнем, он отправляет импульс на блок измерения и управления 7, который и производит обработку результатов.

Режим четыре предоставляет пользователю задать язык пользовательского интерфейса. Данный режим реализуется за счет задание нужного значения с помощью блока выбора режима 6. После чего происходит передача выбранного значения на блок 7, который в свою очередь реализуют данную функцию переводя всю текстовую информацию, выводимую на блок индикации 8 согласно выбранному значению языка.

## 2.2 Функциональная схема электрическая

На рисунке 2 приведена функциональная схема разработанного устройства. Схема содержит в себе функциональную схему блока внешних устройств, а также блока управления и измерения.

Функциональные и принципиальные схемы электрические внешних устройств построены по однотипным типовым схемам на операционном усилителе. Отличие заключается лишь в различных коэффициентах усиления, которые устанавливаются с помощью регулировочного резистора.

Принцип работы рассмотрим по функциональной схеме датчика линейного перемещения.

Функциональная схема датчика линейного перемещения, построена на оптопаре состоящей из пары фотодиодов FD1 AL106 (излучающий) и FD2 AL107 (принимающий). Сигнал от FD1 прошедший механический энкодер, представляющий собой диск с прямоугольными отверстиями и напечатанный на 3D принтере, через FD2 поступает на однокаскадный усилитель, собранный на ВТ1 и компаратор представляющий собой операционный усилитель AD1. С операционного усилителя сигналы ТТЛ уровня поступают на советующий вход блока управления и измерения.

**Функциональная схема блока управления**

Схема содержит устройства связи системы с внешними устройствами, а также устройство индикации. Внешнее управление становится возможным и при помощи системного разъема, но по другим правилам в отличие от порта ввода/вывода, что расширяет возможности контроллера, с его помощью может осуществляется связь с шинами базовой ЭВМ. Также этот разъем может быть использован при отладке работы микропроцессорной системы в целом.

Так же, стоит отметь наличие в системе трех шин, а именно шины управления, адреса и данных. Они представляют из себя проводные линии(проводники) соединяющие между собой различные составные части всей системы. Шины имеют различные разрядности. Также стоит отметить, что разрядность шины может зависеть от производительности системы. Большое количество информации, вызывает потребность в большей шине данных.

Далее будет рассмотрено назначение каждой из шин, применяемых в данном микроконтроллере.

Шина данных – шина основное назначение которой является передача различного рода данных. Данные могут быть получены как из памяти, так и от устройства ввода/вывода и переданы процессору. Поскольку обратное также верно, можно говорить о том, что данная шина является двунаправленной. Следует отметить, что используемый микроконтроллер имеет 8-ми разрядную шину данных, что говорит о том, что за одну передачу/прием максимум может быть передано 8 бит информации.

Шина адреса — главное назначение данного элемента является указание адреса ячеек памяти или портов ввода/вывода. Разрядность же имеет прямую зависимость от необходимого размера памяти.

Шина управления – основной целью данного компонента является передача информации устройству ввода/вывода или памяти о готовности микропроцессора выполнить пересылку данных.

Как результат, процесс работы имеет следующую структуру. По управляющей шине передаются сигналы, которые позволяют устройству ввода/вывода или же памяти обращаться к процессору с запросами. Разрядность данной шины большей степени зависит от типа используемого процессора и количества его управляющих или используемых сигналов.

Из приведенной выше схемы видно, что инициатором всех действий, выполняемых системой является микропроцессор. Данный микропроцессор вырабатывает сигналы управления и выдает их на шину управления. Далее по данной шине сигналы подают на устройства памяти, а также на устройства ввода/вывода. Стоит отметить, что именно процессором задается и адрес выбранной ячейки памяти. После чего выбранный адрес подает на шину адреса и на адресные входы микросхем памяти, в свою очередь по шине данных информация идет в процессор или из него.

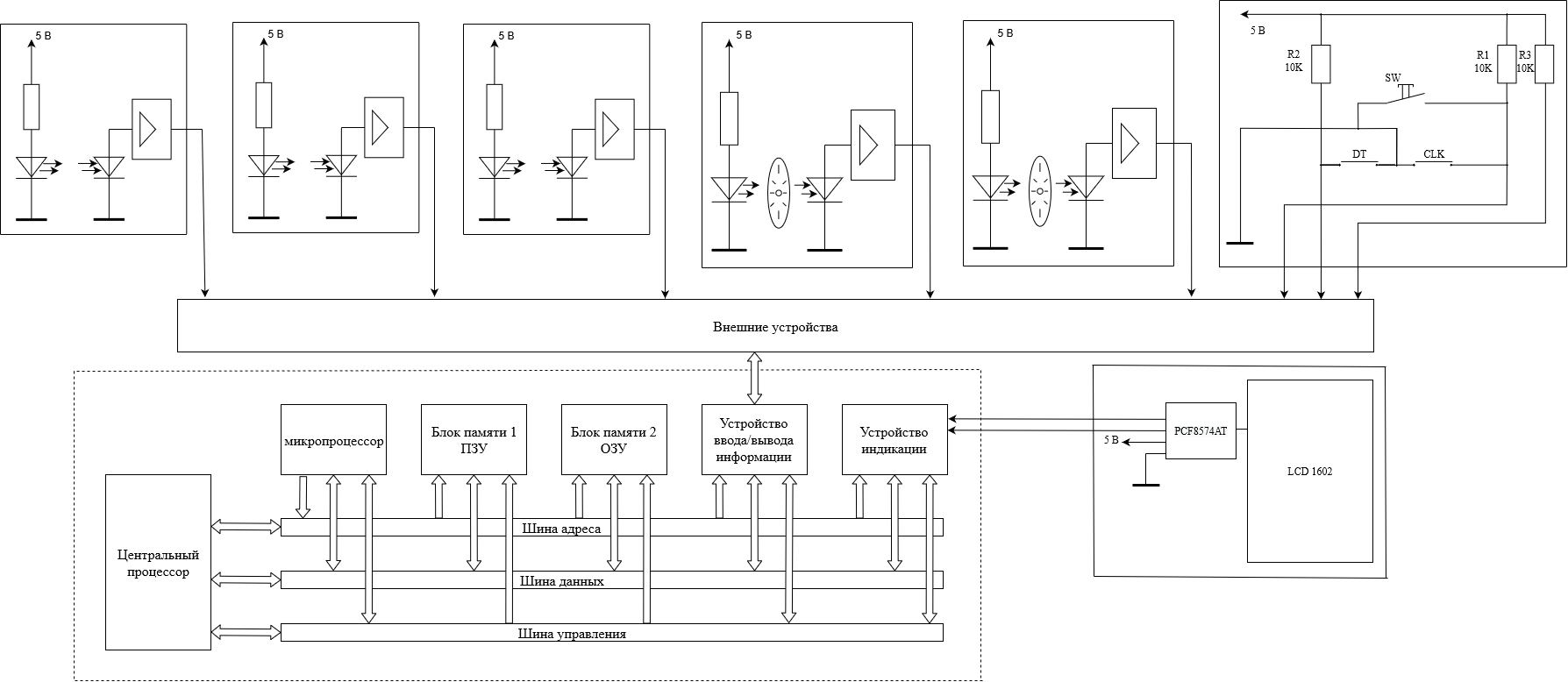


Рисунок 2 – функциональная схема разработанного устройства

## 2.3 Принципиальная схема электрическая

На рисунках 3а, 3б, 3в представлена принципиальная схема электрическая разработанного устройства.

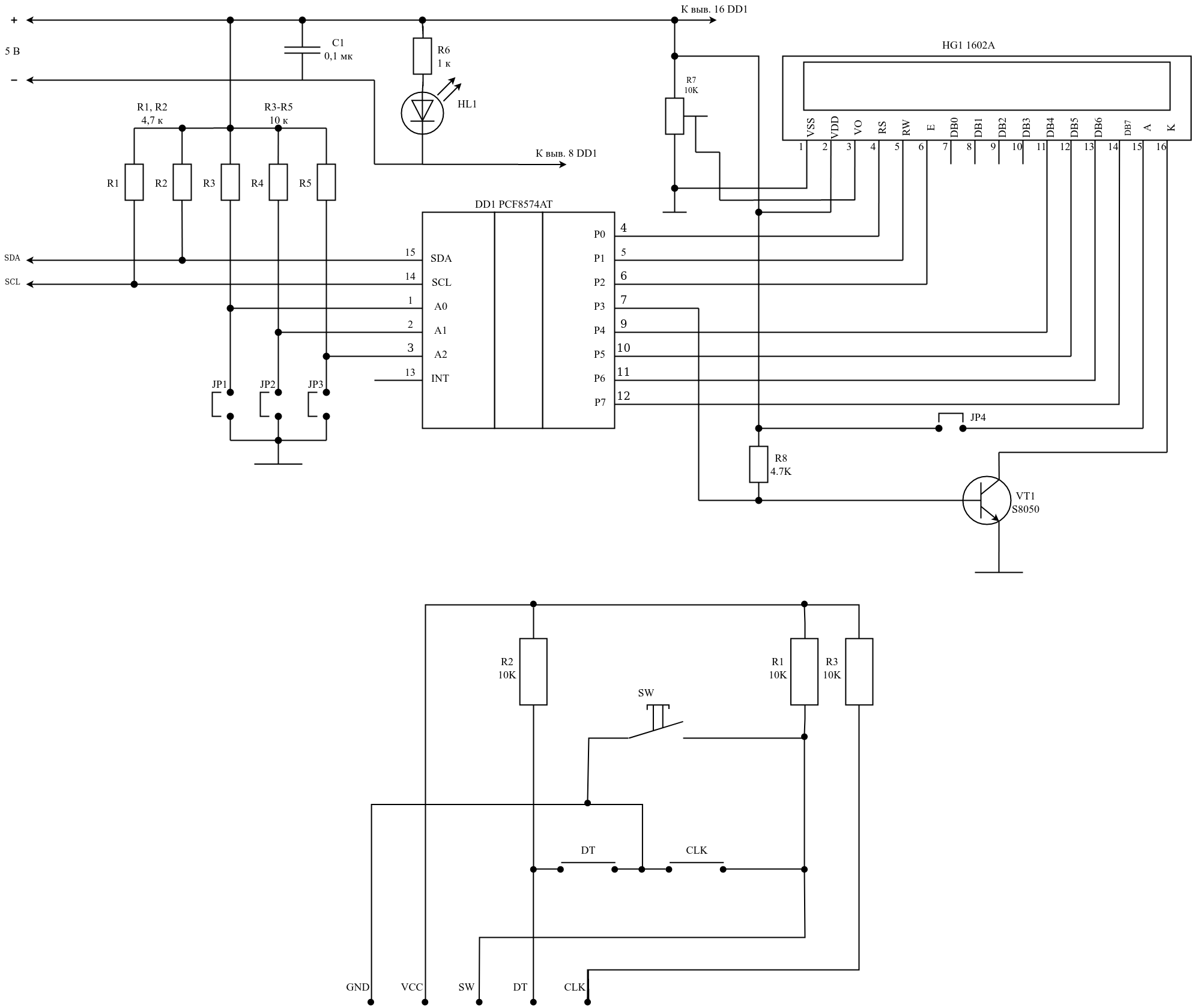


Рисунок 3а — принципиальная схема электрическая блока индикации, а также блока выбора режима.

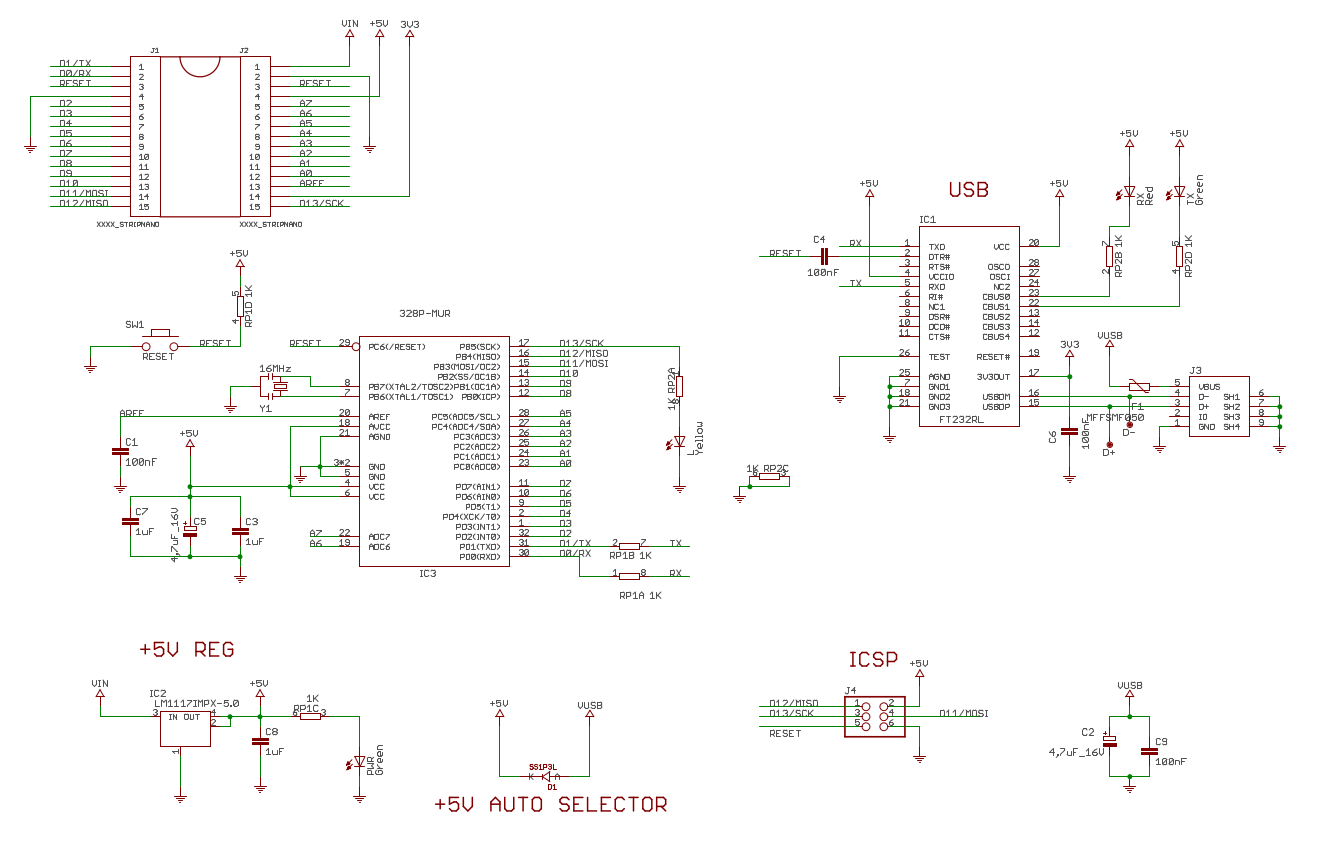


Рисунок 3б — принципиальная схема электрическая блока управления и измерения.

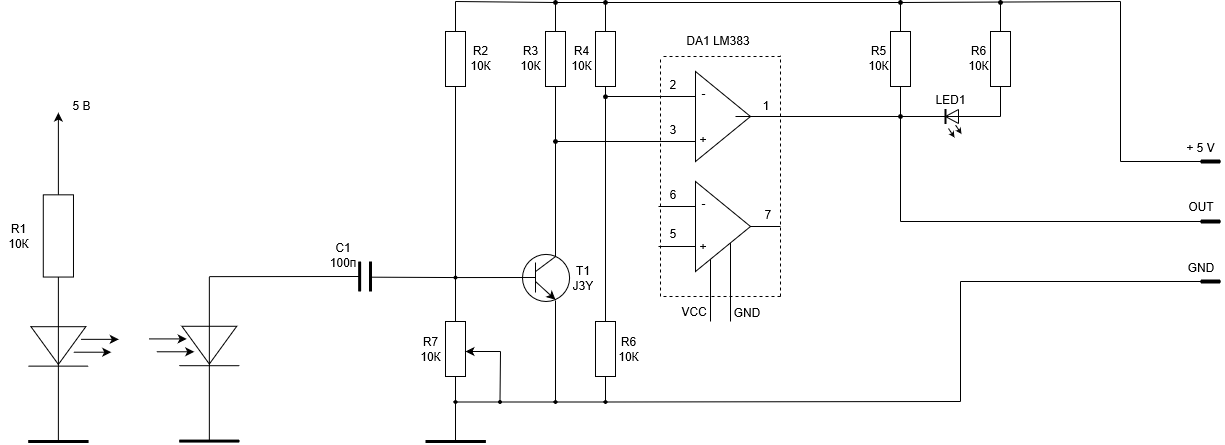


Рисунок 3в — принципиальная схема электрическая усилителя датчиков внешних устройств.

Поскольку объем данной дипломной работы ограничен, а схема структурная электрическая и схема функциональная электрическая рассмотрены достаточно подробно. Принципиальная схема в данной работе не рассматривается.

# 3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

## 3.1 Структура программного кода

Поскольку для реализации задачи поставленной в данной дипломной работе был выбран микроконтроллер Atmega328P, то следует сказать, что написание программного кода для данного микроконтроллера осуществляется с помощью языка С++.

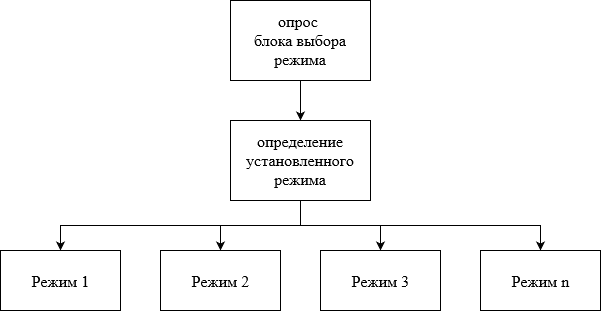


Рисунок 4 – структура программного кода

Как видно из рисунка 4 разработанный блок является многофункциональным. Это достигается за счет возможности выбора режима и написания большого количества программных модулей, реализующих нужный функционал для определенных измерений.

Программирование микроконтроллера начинается с создания двух функций с именами *setup()* и *loop()*. Функция *setup()* выполняется лишь один раз и реализует первоначальная настройка основных компонентов используемых в дальнейшем в программном коде. По завершению, функция *setup()* передает управление микроконтроллером функции *loop()*, которая реализует основную логику выполняемую данным микроконтроллером [3].

void *setup()* {

/\* энкодер \*/

pinMode(pin\_DT, INPUT);

pinMode(pin\_CLK, INPUT);

/\*кнопка энкодера\*/

pinMode(pin\_Btn, INPUT\_PULLUP);

/\*стартовые значения \*/

counter = 5 // параметр для задания расстояния

stapForEncoder = 5; //величина шага для задания расстояния

}

void *loop()* {

/\*функция выбора режима\*/

*choiseOfMode();*

}

Следующим этапом является вызов функции *choiseOfMode()*.Данная функцияотвечает за вызов одного из 4 режимов.

void *choiseOfMode()* {

/\*вызов функциия для определения выборанного режима\*/

switch ( *encoderMoveSetting()* ) {

case 1:{

*firstMove();* /\*вызов первого режима\*/

break;

}

case 2:{

*secondMove();* /\*вызов второго режима\*/

break;

}

case 2:{

*ThirdMove();* /\*вызов третьего режима\*/

break;

}

case 2:{

*fourthMove();* /\*вызов четвертого режима\*/

break;

}

}

}

Решение о том, какой режим будет вызван определяется с помощью вызова функции *encoderMoveSetting()*.Данная функция выполняет задание значения, соответствующего одному из возможных режимов

int *encoderMoveSetting()* {

/\*переменные для реализации функции\*/

int choisedMode = 1;

int lastChoisedMode = 1;

while( true ){

/\*определение положения энкодера\*/

switch ( *GetEncoderState()* ) {

case eNone:{

break;

}

case eRight:{ //поворот в правую сторону

choisedMode++; //увеличение значения выбранного режима на 1

break;

}

case eLeft:{ //поворот в левую сторону

choisedMode--; //уменьшение значения выбранного режима на 1

break;

}

case eButton: { //нажатие кнопки - выбор режима

return choisedMode;

}

}

}

}

Для связи функции *encoderMoveSetting()* с энкодером, используется еще одна функция, *GetEncoderState()*. Даннаяфункцияопределяет положение энкодера с помощью опроса энкодера, при этом возвращая одно из 4 возможных значений.

1. eNone – энкодер не изменил своего положения
2. eLeft – энкодер совершил поворот влево
3. eRight – энкодер совершил поворот вправо
4. eButton – кнопка энкодера нажата

eEncoderState *GetEncoderState()* {

eEncoderState Result = eNone; // Считываем состояние энкодера

CurrentTime = millis();

if (CurrentTime - LastTime >= 5) {

// Считываем не чаще 1 раза в 5 мс для уменьшения ложных срабатываний

LastTime = CurrentTime;

if (digitalRead(pin\_Btn) == LOW ) {

if (ButtonPrev) {

Result = eButton; // Нажатие кнопка

ButtonPrev = 0;

}

} else {

ButtonPrev = 1;

EncoderA = digitalRead(pin\_DT);

EncoderB = digitalRead(pin\_CLK);

if ((!EncoderA) && (EncoderAPrev)) { // Сигнал A изменился с 1 на 0

if (EncoderB) Result = eRight; // B=1 => энкодер вращается по часовой

else Result = eLeft; // B=0 => энкодер вращается против часовой

}

EncoderAPrev = EncoderA; // запомним текущее состояние

}

}

return Result;

}

## 3.2 Описание режимов программного кода

Ввиду ограничения на объема, данную главу рассмотрим на примере режима 1.

После выбора первого режима работы, который является режимом измерения линейного перемещения тела, функция *choiseOfMode()* предает управлениемикроконтроллером первому модулю.

void *firstMove()* {

/\*задание расстояния для линейного перемещения\*/

*encoderDistanceSetting();*

/\*измерение пройденного расстояния\*/

*counterDown();*

}

Сначала функция *firstMove()* передает управление функции *encoderDistanceSetting()*. В свою очередь данная функция реализует задание линейного расстояния необходимого для прохождения физического тела, на основе значений получаемых функцией *GetEncoderState()*.

void *encoderDistanceSetting()* {

/\*определение положения энкодера\*/

switch ( *GetEncoderState()* ) {

case eNone: return;

case eLeft: { // Энкодер вращается влево

if (counter == 0){ //блокировка на задание отрицательных значений

return;

}

counter = counter - stapForEncoder;

break;

}

case eRight: { // Энкодер вращается вправо

counter = counter + stapForEncoder;

break;

}

case eButton:{

buttonStateEncoder = true ;

return;

}

}

После того, как было задано расстояние, которое должно пройти тело. Микроконтроллер переходит в режим подсчета импульсов. Этот режим реализуется с помощью функции void counterDown()

void counterDown(){

while (counter != 0 ){

stateMinus = digitalRead(pinMinus);

if (stateMinus != HIGH){ //ожидаем импульса от датчика линейного перемещения

counter--;

break;

}

}

}

После того, как тело завершило свое передвижение, устройство переходит первоначальное состояние. После чего, снова возможен выбор режима с помощью функции *choiseOfMode().*

# ЗАКЛЮЧЕНИЕ

В результате выполнения данной дипломной работы, были проанализированы основные варианты микроконтроллеров, которые могли бы быть использованы в разработке конечного устройства. Также следует отметить, что был проделан большой объем работ как в проектировании структурной и функциональной схемах, так и в разработке принципиальной схемы, которая бы полностью отвечала задачам поставленным в данной дипломной работе.

Кроме того, было разработано программное обеспечение, достоинством которого является использование так называемых модулей. Что позволяет с легкость добавлять новый функционал.

Как результат, разработанное устройство является многофункциональным блоком управления измерений линейных, а также угловых перемещений.

Поскольку разработанное устройство является универсальным, благодаря нему можно в значительной степени упростить, ускорить, а кроме того увеличить точность большого количества исследовательских, а также различных лабораторных работ при выполнении которых, требуется измерение линейного или же углового перемещения.

Кроме того, стоит отметь, что данный блок также может быть использован и в паре с каким-либо секундомером/таймером, что позволит еще больше расширить диапазон выполняемых лабораторных работ. Примером таких работ может быть машина Атвуда или же работы, связанные с различного рода маятников.[4]

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel: ИП РадиоСофт, 2002. - 170c.
2. Голубцов М.С. Микроконтроллеры ARM. От простого к сложному: СОЛОН-Пресс, 2003. – 288c.
3. Оптика и атомная физика: Лабораторный практикум по физике/ Под рел.Р.И.Солоухина. Новосибирск: Наука. Сиб. ото-пне, 1983. 384 с.
4. Физический практикум/ Под ред.В.И.Ивероновой1 В 2 ч. М.: Физматгиз, 1967 - 1968. Ч.1-2.
5. ЭВМ в курсе общей физики.Изд-во Моск.ун-та, 1982. 230 с.
6. Методы физических измерений: Лабораторный практикум по физике/Под ред.Р.И.Солоухяна.
7. <https://www.alldatasheet.com/datasheet-pdf/pdf/506518/STMICROELECTRONICS/STM8L151C8T6.html> Дата доступа 11.12.2019
8. <https://web.mit.edu/6.115/www/document/16f628.pdf> Дата доступа 11.12.2019
9. <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf> Дата доступа 13.12.2019
10. https://ru.wikipedia.org/wiki/AVR Дата доступа 9.12.2019
11. https://ru.wikipedia.org/wiki/ARM Дата доступа 25.11.2019