ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.

**LCD** жидко-кристаллический экран.

**DD** графическое обозначение микроконтроллером.

**VCC** положительное напряжение источника питания и устройств.

**GND** общий провод источника питания и устройств.

**SDA** шина последовательных данных.

**SLC** шина тактирования.

**Реферат**

В современном учебном процессе большую роль играет обучение практическим навыкам во время лабораторных работ на лабораторных установках. В ходе которых студенты на практике изучают, закрепляют и подтверждают теоретические законы и формульные выражения, изученные в теоретических курсах, таких как:

1. Механика
2. Молекулярная физика
3. Электричество и магнетизм
4. Оптика

В данной курсовой работе был спроектирован 4-х разрядный секундомер с дистанционным управлением на микроконтроллере фирмы Atmel, модели Atmega328P.

Применение микроконтроллера позволило создать малогабаритное устройства отличающим низким электропотребление, а наличие большого количества свободно распространяемой литературы и программного обеспечения позволили произвести лёгкое адаптированное для выполняемых задач, указанных в курсовой работе.

Данный секундомер является универсальным устройством и его предполагается использовать в составе универсального измерительного блока для лабораторных установок лабораторий “Механики” кафедры общей физики в качестве измерителя временных интервалом с дистанционным управлением начала и окончанием счётом времени и управлением исполнительными механизмами: реле, электромагнит, электромагнитная муфта и т.д.

**Введение**

В настоящее время жизнь любого человека невозможно представить без электронных устройств. Благодаря тому, что внедрение электронных систем позволяет уменьшить человеческие риски, за счёт автоматизации человеческого труда, низкая стоимость конечных устройств и качество выполненной работы, зависящее только от выбранного оборудования.

Также не стоит забывать и про спектр использования данных устройств. Электронно-вычислительные устройства можно использовать во всех сферах жизни человека: топливной, химической, машиностроительной, деревообрабатывающей и т.д. Спектр применения ограничен лишь фантазией и сообразительность разработчиков электронных установок.

Кроме того, в течение последнего десятилетия в научно-технической сфере, все чаще стали применяться различные микроконтроллеры, которые позволяют ускорить и упросить достаточно большой спектр задач. В связи с этим, было принято решение о создании универсального блока управления на микроконтроллере.

Целью данной курсовой работы является разработка электронного секундомера с возможность управления через внешние устройства ввода, а также и через собственные органы управления, вывод измеренных результатов.

В ходе над курсовой работой были произведены:

* Анализ существующих схемных решений и подбор литературы
* Выбор из существующей компонентой базы микроконтроллера отвечающего на следующие требования: быстродействие, тактовая частота, ПЗУ, ОЗУ, электропотребление, возможность применения универсальных программ, низкое электропотребление, стоимость.
* Составление структурной, функциональной и принципиальной электрической схемы.
* Адаптация программного продукта из открытых источников, для управления микроконтроллером в режиме измерения времени.
* Сборка и наладка 4-х разрядного электронного секундомера с дистанционным управлением на микроконтроллере

**ГЛАВА 1. ВЫБОР МИКРОКОНТРОЛЛЕРА.**

Ввиду использования микроконтроллеров в данной курсовой работе, предстояло провести выбор из существующей базы вариантов оптимальный микроконтроллер. Основной особенность при выборе микроконтроллера являлось их широкая вариация моделей, что требовало с одной стороны сравнить много вариации моделей, но с другой стороны позволяло подобрать конкретно под нужную задачу устройство.

Поставленной в курсовой работе задаче соответствуют три типа микроконтроллеров:

1. ARM [3]
2. ARV [1]
3. PIC [2]

Сравнительные характеристики трёх групп микроконтроллеров приведены в Таблице 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  | PIC16F690 | Atmega 328P-PU | STM32F100C4T6B |
| Тип ядра | PIC | AVR | ARM |
| Тактовая частота | 8 МГц | 20 МГц | 24 МГц |
| ПЗУ | 8 Кб | 32 Кб | 16 Кб |
| ОЗУ | 1 КБ | 2 Кб | 4 Кб |
| Количество портов ввода/вывода | 18 | 23 | 37 |
| Разрядность шины данных | 8 бит | 8 бит | 32 бит |
| Цена | 4,6 рубля | 4,8 рубля | 4,6 рубля |

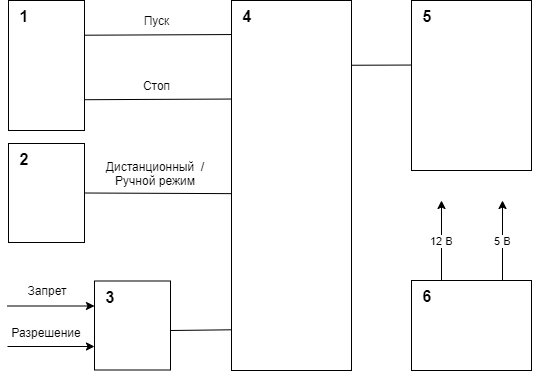
Так же были произведен сравнительные обзор микроконтроллеров семейства Intel MCS-51 с выше указанными микроконтроллерами, но семейство данных микроконтроллеров широко применяется в учебном процессе, так как имеется большое количество макетов, позволяющих программно-моделировать различные устройства, но данные микроконтроллеры отличаются низкими техническими характеристиками [4] использование их было признанно нецелесообразным.

После детального рассмотрения трех микроконтроллеров. Было решено остановится свой выбор для поставленной задачи в курсовой работе на самом распространённом и по большому наличию открытого программного продукта на микроконтроллере фирмы Atmega, а именно Atmega 328P-PU. Данное решение обосновывается тем фактом, что при разработки электронного секундомера с дистанционным управлением на микроконтроллере важное большое количество документации и вспомогательной литературы [5], что позволяет облегчить разработку и последующую поддержку конечного устройства в будущем.

**ГЛАВА 2. РАБОТА УСТРОЙСТВА ПО СТРУКТУРНОЙ, ФУНКЦИОНАЛЬНОЙ, ПРИНЦИПИАЛЬНОЙ СХЕМЕ.**

**2.1 Структурная схема электрическая разрабатываемой установки**

Важно сказать, что, начиная разработку электронного секундомера на базе микроконтроллера, была составлена структурная схема устройства. Структурная схема секундомера представлена на (Рис.1).



1 – схема управления секундомеров в ручном режиме, 2 – блок выбора режима работы секундомера, 3 – внешние сигналы управления счётом, 4 – блок микроконтроллера, 5 – блок индикации, 6 – блок питания.

Рисунок 1 - Структурная схема установки

Данная структурная схема определяет основные функциональные части изделия, их назначение и взаимосвязи между ними. Схема отображает принцип действия устройства в самом общем виде. Каждая часть схемы выполняет полностью законченную роль.

1. Схема управления секундомером в ручном режиме

Для упрощения использования и обслуживания реализацию данного блока решено было выполнить из двух кнопок: ПУСК и СТОП.

Кнопка ПУСК включает отсчёт времени, а кнопкой СТОП отсчёт времени останавливается в ручном режиме.

1. Блок выбора режима работы секундомера
2. Внешние сигналы управления счётом

Внешне управлять электронным секундомером на микроконтроллере только при активации дистанционного режима. В этом случае микроконтроллер ожидает разрешающую логическую единицу с блока внешнего управления для запуска отсчёта времени, а также запрещающую логическую единицу с внешнего управления для остановки секундомера.

1. Блок микроконтроллера

Является основным логическим и управляющем элементов в структурной схеме позволяет гибко управлять различными электронными и электрическими устройствами.

1. Блок индикации

Предназначен для вывода буквенно-цифровой информации поданной микроконтроллером и отображения с помощью жидкокристаллического дисплея (LCD) 1602.

1. Блок питания

Состоит из трех основных частей

1. Понижающий трансформатор с обычного входного переменного напряжения 220 В.
2. Выпрямитель в виде диодного моста.
3. Электролитический конденсатор большой емкости, который сглаживает полусинусоиды напряжения, делая их приближающимися к прямой линии

Данный блок имеет возможность управления в ручном или дистанционном режиме. Когда выбран ручной режим для включения электронного секундомера не доступно управление им дистанционно или с внешних устройств. Органом управления является кнопки. В дистанционном режиме управления микроконтроллер ожидает разрешающую логическую единицу с блока внешнего управления для запуска отсчёта времени, а также запрещающую логическую единицу с внешнего управления для остановки секундомера.

При получении разрешения на включение секундомера программное обеспечения запоминает время, прошедшее с запуска микроконтроллера в который пришла команда на включение секундомера.

После команды остановки секундомера запоминает значение времени, прошедшее с включения микроконтроллера в который пришёл сигнал остановки секундомера.

Расчёт временного интервала введется путем разницы между этими двумя значениями.

Точность измерения времени в данном режиме составляет одна 1 **миллисекунда** (мс).

Так же во время работы секундомера производится индикация временных значений. За счет разницы текущего времени и времени старта.

Данный способ является лучшим для измерения временных интервалов, при условии отсутствия дополнительных устройств. Так же плюсом данного принципа измерения времени является, то что программного обеспечения для доступа ко времени прошедшего с включения микроконтроллера встроено разработчиками микроконтроллера.

**2.2 Функциональные схемы разрабатываемой установки.**

**2.2.1. Функциональная схема управления секундомером в ручном режиме.**

Функциональная схема управления секундомером в ручном режиме состоит из двух управляющих кнопок, которые при нажатии отправляют управляющие сигналы на выводы микроконтроллера.



Разрешающий сигнал для начала отсчета времени отправляется на вывод DD6, а сигнал остановки счёта отправляется на вывод DD7.

2.2.2. Функциональная схема блока выбора режима работы секундомера.

Функциональная схема блока выбора режима работы секундомера может принимать два рабочих состояния: ручное или дистанционное управление счетчиком. В случае ручного управления подается к выводу DD5 микроконтроллера логическая единица, а в случае дистанционного управления подается логический ноль.



2.2.3. Функциональная схема внешних сигналов управления счётом.

Функциональная схема внешних сигналов управления счётом состоит из двух портов, подключенных напрямую к микроконтроллеру. При получении логической единицы на вывод DD6 микроконтроллера запускается секундомер.

При получении логической единицы на вывод DD6 микроконтроллера секундомер останавливается.

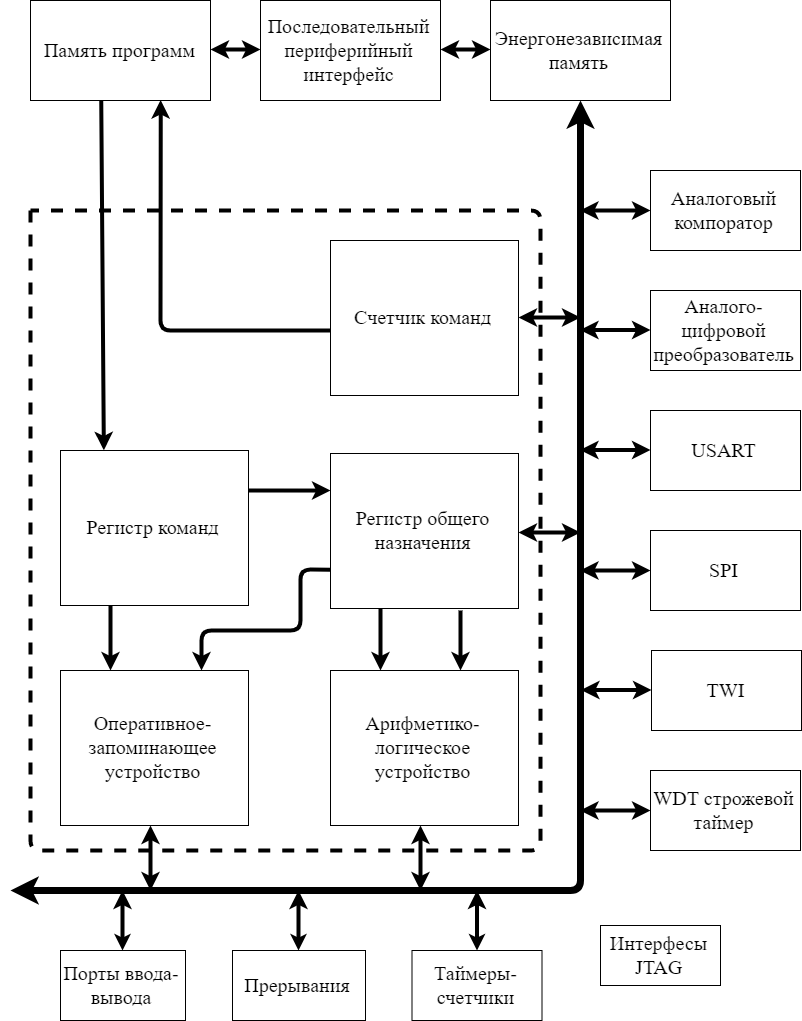


В обоих случаях управление логическим нулём не ведется. Реакция на логическую единицу происходит сразу после получения. Микроконтроллер не ожидает конца разрешающего импульса.

2.2.4. Функциональная схем блока микроконтроллера.

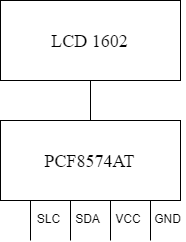
На (Рис.2) представлена функциональная схема микроконтроллера Atmega328, который является основный компонентом в разрабатываемом устройстве. Блок микроконтроллера состоит из:

1. Памяти программ - предназначена для хранения последовательности команд, управляющих функционированием микроконтроллера. Ее главное достоинство в том, что она построена на принципе электрической пере программируемости, т. е. допускает многократное стирание и запись информации. В процессе выполнения программа считывается из этой памяти, а блок управления обеспечивает ее декодирование и выполнение необходимых операций. Содержимое памяти программ не может меняться во время выполнения программы. Поэтому функциональное назначение микроконтроллера не может измениться, пока содержимое его памяти программ не будет перепрограммировано.
2. Последовательный-периферийный интерфейс - интерфейс для последовательного обмена данными. В том случае, если в системе присутствует внешний процессор, с помощью этого интерфейса может быть обеспечена межпроцессорная связь.
3. Энергонезависимая память – предназначена для долговременного хранения различной информации, которая может изменяться в процессе функционирования микроконтроллерной системы, используется энергонезависимая память (EEPROM-память). Данный микроконтроллер имеет блок энергонезависимой электрически перезаписываемой памяти данных размером 32 Кбайт. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, коэффициентов, серийных номеров, ключей и т.п.
4. Аналоговый компаратор - предназначен для сравнения непрерывно изменяющихся сигналов. Аналоговый компаратор имеет два входных аналоговых сигналов: анализируемый и опорный сигнал. После сравнения, на выходе получаем логический сигнал, содержащий 1 бит информации.
5. Аналого-цифровой преобразователь - устройство, выполняющее преобразования входной физической величины в ее числовое представление.
6. USART - удобный и простой последовательный интерфейс последовательного ввода-вывода, который может использоваться для работы с периферийными устройствами.
7. SPI - предназначен для организации обмена данными между двумя устройствами. С его помощью может осуществляться обмен данными между микроконтроллером и различными устройствами. Также с помощью этого интерфейса удобно производить обмен данными между несколькими микроконтроллерами AVR.
8. TWI - интерфейс позволяет объединить вместе до 128 различных устройств с помощью двунаправленной шины, состоящей из линии тактового сигнала (SCL) и линии данных (SDA).
9. WDT сторожевой таймер - предназначен для предотвращения катастрофических последствий от случайных сбоев программы. Он имеет свой собственный RC-генератор, работающий на частоте 1 МГц. Идея использования сторожевого таймеpа предельно проста и состоит в регулярном его сбрасывании под управлением программы или внешнего воздействия до того, как закончится его выдеpжка вpемени и не пpоизойдет сброс процессора.
10. Интерфейсы JTAG – создан для решения проблем тестирования электроники. Стандарт позволяет значительно упростить жизнь за счет встраивания специальной архитектуры в современные чипы, обеспечивающей доступ к выводам (точнее, к специальным блокам ввода-вывода) с помощью 4-х проводного последовательного интерфейса. Эта архитектура позволяет не только контролировать их состояние, но и управлять ими.
11. Таймеры-счётчики – нужен для измерения точных временных промежутков. Точность достигает одного машинного такта.
12. Блок прерывания – один из важнейших блоков в микроконтроллере. Прерывание прекращает нормальный ход программы для выполнения приоритетной задачи, определяемой внутренним или внешним событием.   Для каждого такого события разрабатывается отдельная программа, которую называют подпрограммой обработки запроса на прерывание (для краткости - подпрограммой прерывания), и размещается в памяти программ. При возникновении события, вызывающего прерывание, микроконтроллер сохраняет содержимое счетчика команд, прерывает выполнение центральным процессором текущей программы и переходит к выполнению подпрограммы обработки прерывания. После выполнения подпрограммы прерывания осуществляется восстановление предварительно сохраненного счетчика команд и процессор возвращается к выполнению прерванной программы. Для каждого события может быть установлен приоритет. Понятие приоритет означает, что выполняемая подпрограмма прерывания может быть прервана другим событием только при условии, что оно имеет более высокий приоритет, чем текущее. В противном случае центральный процессор перейдет к обработке нового события только после окончания обработки предыдущего.
13. Порты ввода-вывода служат для подключения внешних устройств.
14. Счетчик команд - это специализированный внутренний регистр микроконтроллера, в котором хранится адрес следующей команды.
15. Регистр команд - регистр, содержащий команду, исполняемую в данный момент процессором.
16. Регистр общего назначения - регистры, используемые для хранения промежуточных итогов вычислений. Имеют два значительных преимущества: обращение к регистрам выполняется намного быстрее, чем к памяти, при обращении к регистрам не нужно указывать адрес (как в случае с памятью).
17. Оперативное-запоминающее устройство - память, предназначенная для временного хранения данных и команд, необходимых процессору для выполнения им операций.
18. Арифметико-логическое устройство - служит для выполнения арифметических и логических преобразований над данными, называемыми в этом случае операндами.



2.2.5. Функциональная схема блока индикации.

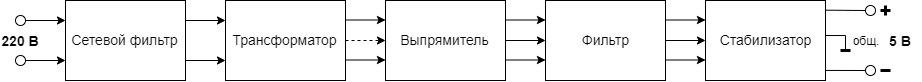
Функциональная схема блока индикации состоит из жидкокристаллического дисплея (Далее – LCD), микроcхемы PCF8574AT выступающей как расширитель ввода-вывода шины I2C обеспечивающего удаленное расширение портов вводов/выводов интерфейса I2C, что позволяет вместо 6 управляющих сигналов использовать 2.



В будущем данное сочетание может обеспечить возможность подключения к портам ввода/вывода микроконтроллера дополнительных устройств.

2.2.6. Функциональная схема блока питания.

Для питания устройства требуется блок питания, который должен иметь минимальные выходные характеристики по напряжению 5 В, а по току 40 мА.



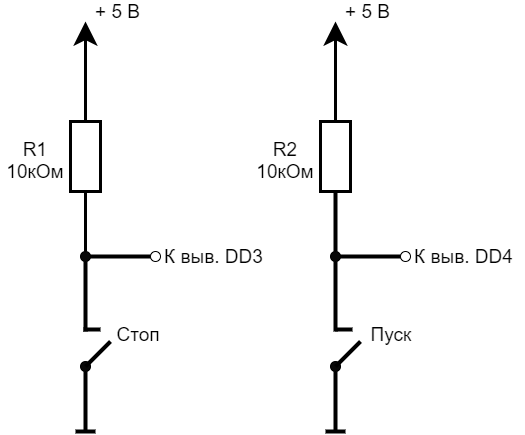
Расположение функциональных блоков расположено последовательно, то и работу устройства стоит рассматривать как последовательный набор действий:

1. Сетевой фильтр ослабляет помехи по сети.
2. Трансформатор преобразует напряжение переменного тока 220 В, в нужные нам 5 В.
3. Выпрямитель предназначен для превращения переменного напряжения в постоянное, путем отсечение отрицательных полуволн.
4. Фильтр сглаживает полуволны до получение неизменного значение напряжение во время работы блока питания.
5. Стабилизатор выполняет он работает так, что напряжение на выходе всегда одинаковое по напряжению.

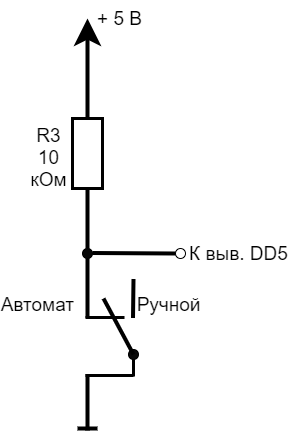
2.3. Принципиальные схемы разрабатываемой установки.

2.3.1. Принципиальные схема управления секундомером в ручном режиме

Во всех режимах работы блока управления защита от дребезга контактов выполнена программно, что дополнительно повышает надежность блока управления.



2.3.2. Принципиальная схема выбора режима работы секундомера.

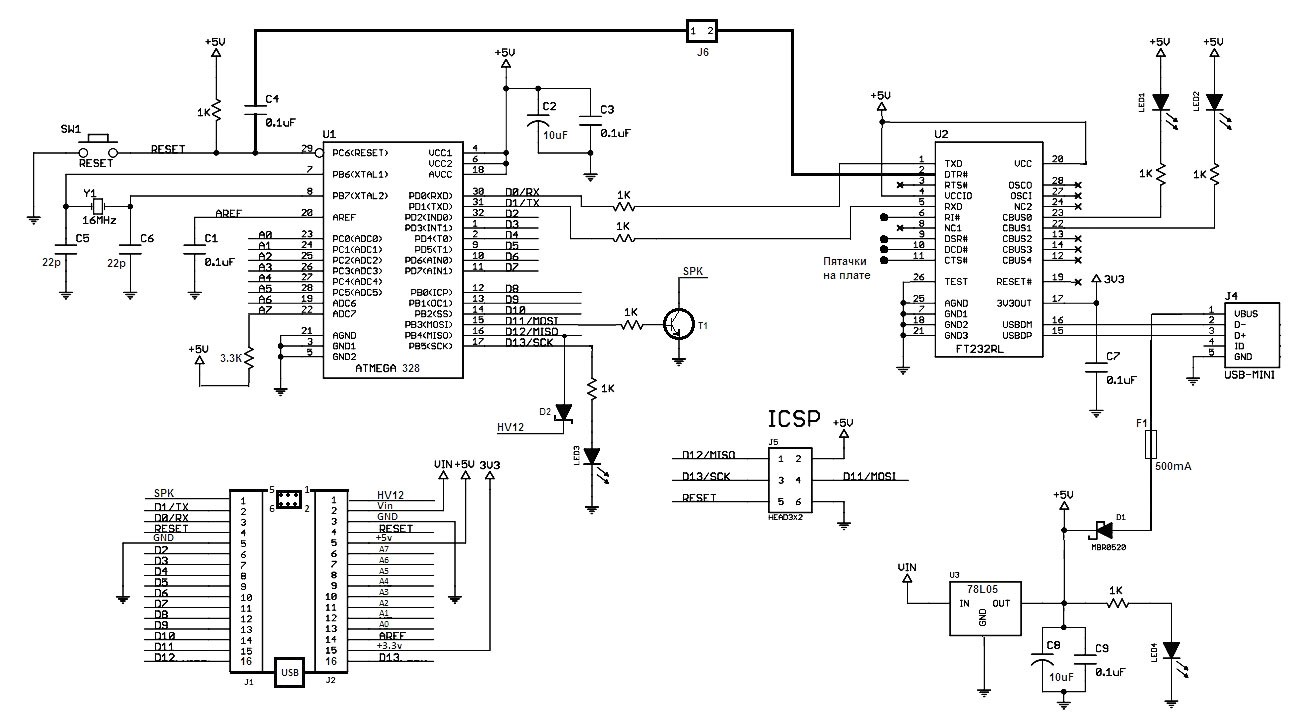


2.3.3. Принципиальная схема внешние сигналов управления счётом.



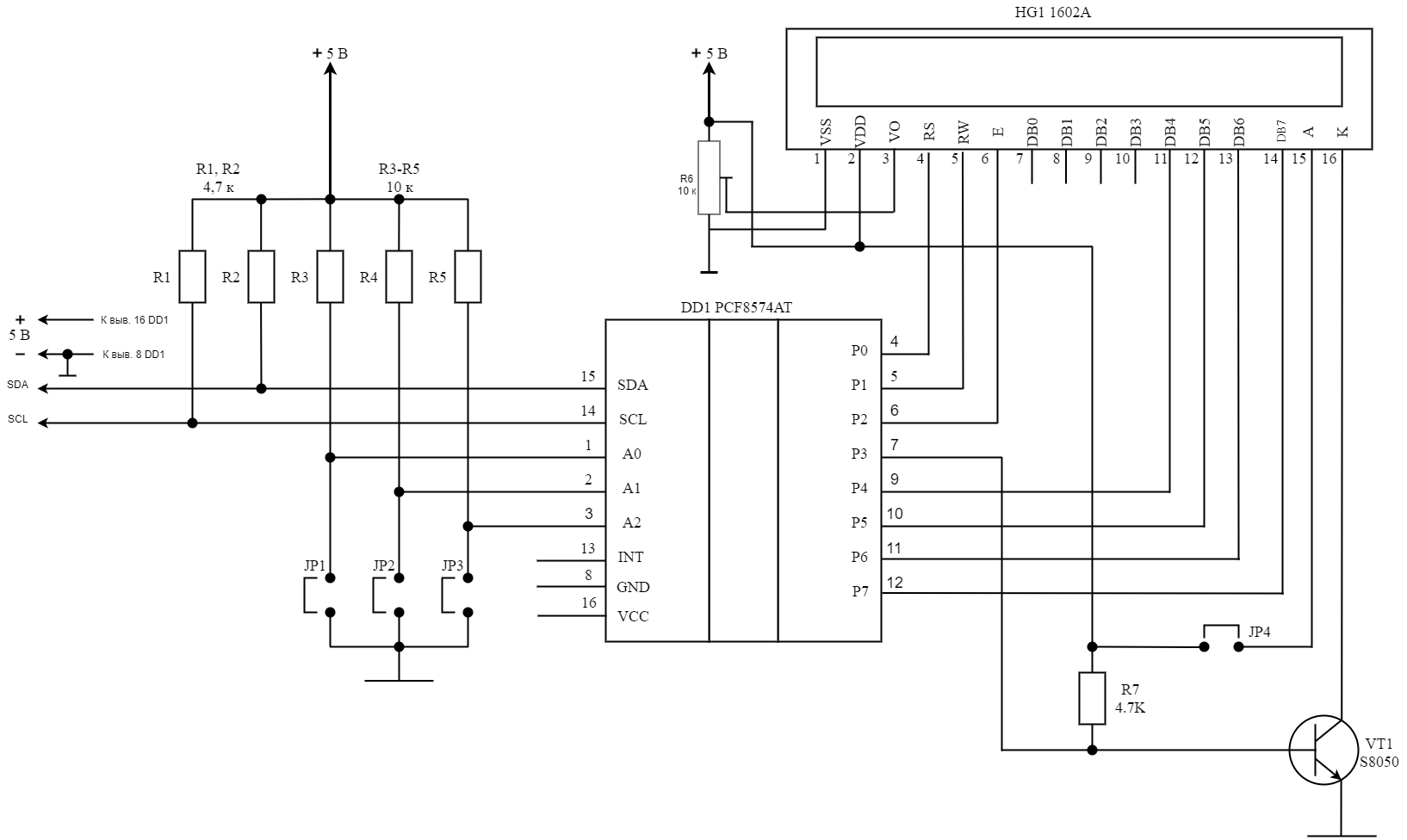
2.3.4. Принципиальная схема блока микроконтроллера.

Так как работу устройства по структурной и функциональной схеме было в достаточном объеме описано и ввиду ограничение в объеме данной работы. Рассмотрение принципиальной схемы приведено, но не рассматривается.



2.3.5. Принципиальная схема блока индикации.

Принципиальная схема представлена на рисунке :



**ГЛАВА 3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.**

Так как для выполнения задачи разработки секундомера с дистанционным управлением работе был выбран микроконтроллер Atmega328P, то следует сказать, что написание программного кода для данного микроконтроллера осуществляется с помощью языка С++. Отсчет времени получается благодаря встроенному таймеру-счетчики, который отсчитывает время после подачи питания.

В приложение 1 приведен программный код, под которым работает микроконтроллер. [Приложение 1]

Запуск исполняющей программы в микроконтроллере начинается с создания с функции *setup()* и *loop()*. В функции *setup()* задается первоначальная настройка работы микроконтроллера, таких как инициализация экрана, настройка яркости и прочие разовые действия нужные для запуска и настройки микроконтроллера. Функция *loop()* реализует основную логику программного обеспечения.

После входа исполняющего программного обеспечения в функцию *loop()* проверяется выбор режима работы: дистанционного или ручного. После проверки ожидается размещающий сигнал на включение секундомера. В нужным момент секундомер можно остановить путём подачи сигнала “Стоп”. После этого последнее значение измеренного времени фиксируется на экране и программа приходит в изначальное состояние.

ЗАКЛЮЧЕНИЕ

В данной курсовой работы были проанализированы основные модели микроконтроллеров, которые можно было использовать для решения поставленных задач в разработке конечного устройства. Следует отметить, что была спроектирована структурная, функциональная, принципиальная схема, которая бы полностью отвечала задачам, поставленным в данной курсовой работе.

Также, было разработано программное обеспечение, достоинством которого является использование встроенных программных решений. Что позволяет с легкость гарантировать бесперебойность устройства и стабильную работу.

Как результат, разработанное устройство является высокоточным блоком измерения времени.

Поскольку разработанное устройство является универсальным, благодаря нему можно в значительной степени упростить, ускорить, а кроме того увеличить точность большого количества исследовательских работ, связанных с измерением интервалов времени.

Также стоит отметь, что разработанный блок может быть использован и как дополнительный инструмент в работах где требуется измерение линейных и угловых передвижений тел, что позволит еще больше расширить диапазон выполняемых лабораторных работ на кафедре общей физики ГрГУ им. Янки Купалы. Примером таких работ может быть машина Атвуда или же работы, связанные с различного рода измерениями.

**СПИСОК ЛИТЕРАТУРЫ**

1. Atmel 8-bit Microcontroller with 4/8/16/32KBytes InSystem Programmable Flash [Электронный ресурс], 2013. – 31с. – Режим доступа: https://www.farnell.com/datasheets/1693866.pdf. – Дата доступа: 24.11.2019.
2. PIC16F631/677/685/687/689/690 Data Sheet [Электронный ресурс], 2006. – 294с. – Режим доступа: http://ww1.microchip.com/downloads/en/devicedoc/41262c.pdf. – Дата доступа:
3. Low & medium-density value line, advanced ARM®-based 32-bit MCU with 16 to 128 KB Flash, 12 timers, ADC, DAC & 8 comm interfaces [Электронный ресурс], 2016. – 96с. – Режим доступа: https://www.st.com/resource/en/datasheet/stm32f100cb.pdf. Дата доступа: 14.11.2019.
4. Герман, А.Е. Микроконтроллеры семейства Intel MCS-51: учебное пособие / А.Е. Герман- 2003. – С. 20-34.
5. Баранов, В.Н. Применение микрокотроллеров AVR : схемы, алгоритмы, программы: учебник / В.Н. Баранов – 2006. – С. 12-16.

ПРИЛОЖЕНИЕ 1. Программное обеспечение сконструированного устройства

// подключение библиотек для работы с ЖК экраном

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

// сигнал включения секундометра

int runStopwatchSignalPIN = 2;

// сигнал отключения секундометра

int breakStopwatchSignalPIN = 3;

// кнопка пуска секундометра

int runStopwatchButtonPIN = 10;

// кнопка остановки секундометра

int breakStopwatchButtonPIN = 11;

void setup() {

// Инициализация экрана

lcd.init();

// Включение подсветки дисплея

lcd.backlight();

// Первоначальное значение экрана

lcd.print("Здравствуйте!");

// Установка курсова в экрана на первоначальное значение

lcd.setCursor(8, 1);

// Устанавливается скорость передачи данных в битах в секунду

// Требуется для последовательной передачи данных.

Serial.begin(9600);

}

void loop() {

if(digitalRead(runStopwatchSignalPIN) == HIGH){

// Вход в режим секундомера

Serial.println("вход в секундометр");

startTimer\_whenSignalStart();

// Выход из режима секундомера

Serial.println("выход из секундометра");

}

// При наличие логической единицы - могу работать в ручном режиме

if(digitalRead(breakStopwatchSignalPIN) == HIGH){

// Вход в режим ожидания разрещающего сигнала

while(true){

// Ожидание разрещающего сигнала

if(digitalRead(runStopwatchButtonPIN) == HIGH){

// Запуск таймера

startTimer\_whenCaseButtonStart();

break;

}

}

}

}

// Секундомер в режиме дистанционного управления

void startTimer\_whenSignalStart(){

// Получение времени в который запустился секундомер

int tim = millis();

while(true){

// Получение времени работы секундомера

int c = millis() - tim;

// дистанционный сигналы остановки секундомера

if(digitalRead(runStopwatchSignalPIN) == LOW ){

// остановка секундомера

break;

}

// вывод времени работы секундомера

displayTimeDefault(c);

// ожидание 5 мс

delay(5);

}

}

// Секундомер в режиме ручного управления

void startTimer\_whenCaseButtonStart(){

// Получение времени в который запустился секундомер

int tim = millis();

while(true){

// Получение времени работы секундомера

int c = millis() - tim;

//ручной сигнал остановки секундомера

if(digitalRead(breakStopwatchButtonPIN) == HIGH ){

// остановка секундомера

break;

}

// вывод времени работы секундомера

displayTimeDefault(c);

// ожидание 5 мс

delay(5);

}

}

// функция отвечающая за вывод значения времени

void displayTimeDefault(int t){

// очистка предыдущего значения с экрана

lcd.clear();

// Инициализация временных переменных

int second = 0;

int milisecond = 0;

// Если время меньше секунды

if(t<1000){

second = 0;

// откидывание целого числа из времени

// от 600 мс остается 0.6с

milisecond = t % 1000/10;

}else{ // если время больше секунды

// округление до целого числа

second = t/1000;

// откидывание целого числа из времени

// от 600 мс остается 0.6с

milisecond = t % 1000/10;

}

// вывод на экран

lcd.print(second);

lcd.print(".");

lcd.print(milisecond);

}