

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «КАМЧАТСКИЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО
«КамчатГТУ»)

Факультет информационных технологий

Кафедра «Системы управления»

ОТЧЕТ
о научно-исследовательской работе

Студента(ки) 2 курса направления подготовки
27.03.04 «Управление в технических системах»

группы 19УСБ Скосырева Никиты Александровича

Место прохождения практики: ФГБОУ ВО «КамчатГТУ»

Сроки прохождения практики: с «05» 07 2021 г. по «17» 07 2021 г.

<p>Руководитель практики от университета: <u>Мищенко Михаил Александрович</u> <i>(Ф.И.О. руководителя от университета)</i> <u>доцент кафедры «Системы управления», к.ф.-м.н.</u> <i>(должность руководителя практики)</i></p> <p>Оценка: _____ «<u>17</u>» <u>07</u> 2021 г.</p>	
<p>_____</p> <p>(подпись)</p>	

Петропавловск-Камчатский
2021 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «КАМЧАТСКИЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО
«КамчатГТУ»)

Факультет информационных технологий

Кафедра «Системы управления»

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
НА ПРАКТИКУ**

Обучающийся: Скосырев Никита Александрович

Тип практики: практика по получению первичных профессиональных умений и навыков

Направление подготовки/специальность:

27.03.04 «Управление в технических системах»

Направленность (профиль): «Управление и информатика в технических системах»

Группа: 19 УСб

№ п/п	Наименование разделов (этапов)	Наименование и содержание работы (мероприятий)	Сроки выполнения
1	1	Разработать устройство на основе МК ATmega8, пяти светодиодов и кнопки. При нажатии кнопки на светодиоды с задержкой в 1 секунду выводится бинарный код каждой буквы из вашей последовательности ФИО.	05.07 – 17.07

Руководитель практики
от университета

(подпись)

Мищенко М.А.
(ФИО)

Задание принял

(подпись)

Скосырев Н.А.
(ФИО студента)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «КАМЧАТСКИЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО
«КамчатГТУ»)

Факультет информационных технологий
Кафедра «Системы управления»

ДНЕВНИК
прохождения практики

Фамилия (И.О.) обучающегося: Скосырев Н.А.
Группа: 19УСБ

Место прохождения практики: ФГБОУ ВО «КамчатГТУ»

Руководитель от университета: доцент кафедры «Системы управления», к.ф.-
м.н. Мищенко М.А.

Дата	Подразделение предприятия	Краткое описание работы	Отметки и подпись руководителя практики
05.07	Каф. СУ	Инструктаж по технике безопасности	
06.07	Каф. СУ	Изучение семейства Микроконтролеров AVR	
07.07	Каф. СУ	Изучение архитектуры	
08.07	Каф. СУ	Характеристики и возможности МК ATmega8	
09.07	Каф. СУ	Используемые программы для разработки	
010.07	Каф. СУ	Знакомство с Proteus	
11.07	Каф. СУ	Знакомство с Atmel Studio	
12.07	Каф. СУ	Разработка кода	
13.07	Каф. СУ	Разработка кода	
14.07	Каф. СУ	Разработка кода	
15.07	Каф. СУ	Разработка кода	
16.07	Каф. СУ	Результаты работы	
17.07	Каф. СУ	Подготовка материалов для отчета	

Подпись практиканта _____

Подпись руководителя практики от университета _____

Содержание

Введение.....	6
1. Описание семейства Микроконтролеров AVR.....	7
1.1 Описание архитектуры.....	10
1.2 Характеристики и возможности МК ATmega8.....	11
2. Используемые программы для разработки.....	14
2.1 Proteus.....	14
2.2 Atmel Studio.....	15
3. Разработка кода.....	16
4 Результаты работы.....	17
Заключение.....	18
Список использованных источников.....	19
Приложение 1.....	20

Введение

Целью организации и проведения практики по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности (учебной практики) является закрепление знаний, полученных студентами при изучении естественнонаучных и профессиональных дисциплин, подготовка их к изучению последующих дисциплин и прохождению производственной практики.

Задачами прохождения практики являются:

- предоставление студентам объективного и полного представления об избранной ими специальности, ее сферах и направлениях;
- выравнивание навыков работы студентов на персональных компьютерах, углубленное освоение ими программных систем для последующего использования в учебном процессе;
- закрепление полученных в процессе обучения знаний на практике для реализации общей образовательной программы.

1 Описание семейства Микроконтролеров AVR.

AVR – это название популярного семейства микроконтроллеров, которое выпускает компания Atmel. Кроме AVR под этим брендом выпускаются микроконтроллеры и других архитектур, например, ARM и i8051.

Существует три вида микроконтроллеров:

- AVR 8-bit.
- AVR 32-bit.
- AVR xMega

Самым популярным уже более десятка лет является именно 8-битное семейство микроконтроллеров. Многие радиолюбители начинали изучать микроконтроллеры с него. Почти все они познавали мир программируемых контроллеров делая свои простые поделки, вроде светодиодных мигалок, термометров, часов, а также простой автоматики, типа управления освещением и нагревательными приборами.

Микроконтроллеры AVR 8-bit в свою очередь делятся на два семейства:

- Attiny – из названия видно, что младшее (tiny – юный, молодой, младший), в основном имеют от 8 пинов и более. Объем их памяти и функционал обычно скромнее, чем в следующем;
- Atmega – более продвинутые микроконтроллеры, имеют большее количество памяти, выводов и различных функциональных узлов.

Самым мощным подсемейством микроконтроллеров является xMega – эти микроконтроллеры выпускаются в корпусах с огромным количеством пинов, от 44 до 100. Столько необходимо для проектов с большим количеством датчиков и исполнительных механизмов. Кроме того, увеличенный объем памяти и скорость работы позволяют получить высокое быстродействие.

Микроконтроллеры применяются почти везде. Практически каждое устройство в 21 веке работает на микроконтроллере: измерительные приборы, инструменты, бытовая техника, часы, игрушки, музыкальные шкатулки и

открытки, а также многое другое; одно лишь перечисление займет несколько страниц текста.

Разработчик может использовать аналоговый сигнал подводя его на вход микроконтроллера и манипулировать с данными о его значении. Эту работу выполняет аналогово-цифровой преобразователь (АЦП). Данная функция позволяет общаться пользователю с микроконтроллером, а также воспринимать различные параметры окружающего мира с помощью датчиков.

В распространенных AVR-микроконтроллерах, например, Atmega328, который на 2017 году является сердцем многих плат Arduino, но о них позже. Используется 8 канальный АЦП, с разрядностью 10 бит. Это значит вы сможете считать значение с 8 аналоговых датчиков. А к цифровым выводам подключаются цифровые датчики, что может быть очевидным. Однако цифровой сигнал может являться только 1 (единицей) или 0 (нулем), в то время как аналоговый может принимать бесконечное множество значений.

В общем структура AVR микроконтроллера изображена на схеме:

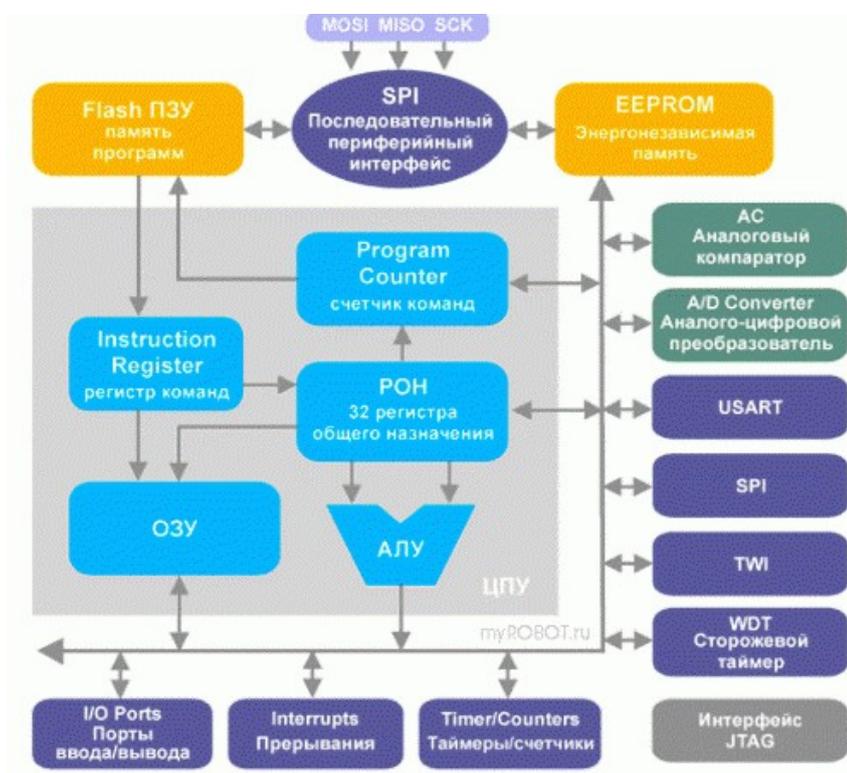


Рисунок – 1. Структура AVR микроконтроллера

Все узлы подписаны, но всё же некоторые названия могут быть не столь очевидными. Давайте рассмотрим их обозначения.

АЛУ – арифметико-логическое устройство. Нужно для выполнения вычисления.

Регистры общего назначения (РОН) – регистры которые могут принимать данные и хранить их в то время пока микроконтроллер подключен к питанию, после перезагрузки стираются. Служат как временные ячейки для операций.

Прерывания – что-то вроде события которое возникает по внутренним или внешним воздействиям на микроконтроллер – переполнение таймера, внешнее прерывание с пина МК и т.д.

JTAG – интерфейс для внутрисхемного программирования без снятия микроконтроллера с платы.

Flash, ОЗУ, EEPROM – виды памяти – программ, временных рабочих данных, долгосрочного хранения независимая от подачи питания к микроконтроллеру соответственно порядку в названиях.

Таймеры и счетчики – важнейшие узлы в микроконтроллере, в некоторых моделях их количество может быть до десятка. Нужны для того, чтобы отчитывать количество тактов, соответственно временные отрезки, а счетчики увеличивают свое значение по какому-либо из событий. Их работа и её режим зависят от программы, однако выполняются эти действия аппаратно, т.е. параллельно основному тексту программы, могут вызвать прерывание (по переполнению таймера, как вариант) на любом этапе выполнения кода, на любой его строке.

A/D (Analog/Digital) – АЦП, его назначение мы уже описали ранее.

WatchDogTime (Сторожевой таймер) – независимый от микроконтроллера и даже его тактового генератора RC-генератор, который отсчитывает определенный промежуток времени и формирует сигнал сброса МК, если тот работал, и пробуждения – если тот был в режиме сна (энергосбережения). Его работу можно запретить, установив бит WDTE в 0.

Выходы микроконтроллера довольно слабые, имеется в виду то, что ток через них обычно до 20-40 миллиампер, чего хватит для розжига светодиода и LED-индикаторов. Для более мощной нагрузки – необходимы усилители тока или напряжения, например, те же транзисторы.

1.1 Описание архитектуры

Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединённых в регистровый файл. В отличие от «идеального» RISC, регистры не абсолютно ортогональны:

Некоторые команды работают только с регистрами r16...r31. К ним относятся команды, работающие с непосредственным операндом: ANDI/CBR, ORI/SBR, CPI, LDI, LDS (16-бит), STS (16-бит), SUBI, SBCI, а также SER.

Команды, увеличивающие и уменьшающие 16-битное значение (в тех моделях, где они доступны) с непосредственным операндом (ADIW, SBIW), работают только с одной из пар r25: r24, r27: r26 (X), r29: r28 (Y), или r31: r30 (Z);

Команда копирования пары регистров (в тех моделях, где доступна) работает только с соседними регистрами, начинающимися с нечётного (r1: r0, r3: r2, ..., r31: r30);

Результат умножения (в тех моделях, в которых есть модуль умножения) всегда помещается в r1: r0. Также только эта пара используется в качестве операндов для команды самопрограммирования (где доступна);

Некоторые варианты команд умножения принимают в качестве аргументов только регистры из диапазона r16...r23 (FMUL, FMULS, FMULSU, MULSU).

1.2 Характеристики и возможности МК ATmega8

Все приборы одного семейства AVR совместимы по исходным кодам и тактированию. Семейство обеспечено комплектом программ и системами отладки, включающими: макро-ассемблеры, отладчики/симуляторы программ, внутрисхемные эмуляторы, и отладочные устройства.

Микроконтроллеры семейства AVR поставляются в очищенном состоянии – содержимое и Flash-памяти программ и ЭСППЗУ данных находится в состоянии FF и готово к программированию.

Характеристики ядра контроллера: Основными характеристиками центрального процессора микроконтроллеров рассматриваемого семейства Classic являются:

- полностью статическая архитектура;
- минимальная тактовая частота равна нулю;
- АЛУ подключено непосредственно к регистрам общего назначения;
- большинство команд выполняется за один машинный цикл;
- многоуровневая система прерываний; поддержка очереди прерываний;
- от 3 до 16 источников прерываний (из них до 2 внешних);
- наличие программного стека.

Характеристики подсистемы ввода/вывода. Основными характеристиками подсистемы ввода/вывода являются:

- программное конфигурирование и выбор портов ввода/вывода;
- каждый вывод может быть запрограммирован как входной или как выходной независимо от других;
- входные буферы с триггером Шмитта на всех выводах;
- возможность подключения ко всем входам внутренних подтягивающих резисторов (сопротивление резисторов составляет 35...120 кОм);
- нагрузочная способность всех выводов составляет до 20 мА, что позволяет непосредственно управлять светодиодными индикаторами.

Периферийные устройства. Микроконтроллеры семейства Classic обладают достаточно развитой периферией. Набор периферийных устройств, имеющихся в составе того или иного микроконтроллера, зависит от конкретной модели. Перечислим все периферийные устройства, так или иначе встречающиеся в микроконтроллерах семейства:

- 8-разрядный таймер/счетчик с предделителем (таймер T0);
- 16-разрядный таймер/счетчик с предделителем (таймер T1);
- 8-разрядный таймер/счетчик с возможностью работы в асинхронном режиме (таймер T2);
- сторожевой таймер (WDT);
- одно- или двухканальный 8... 10-разрядный генератор сигнала с широтно-импульсной модуляцией (ШИМ);
- одноканальный 8-разрядный генератор сигнала с ШИМ;
- аналоговый компаратор;
- 10-разрядный АЦП (6 или 8 каналов);
- полнодуплексный универсальный асинхронный приемопередатчик (UART);
- последовательный синхронный интерфейс SPI.

Архитектура ядра. Ядро является «сердцем» микроконтроллеров AVR. Оно выполнено по усовершенствованной RISC (enhanced RISC) архитектуре, в которой используется ряд решений, направленных на повышение быстродействия микроконтроллеров. Схематическое упрощенное изображение ядра показано на рис. 1.

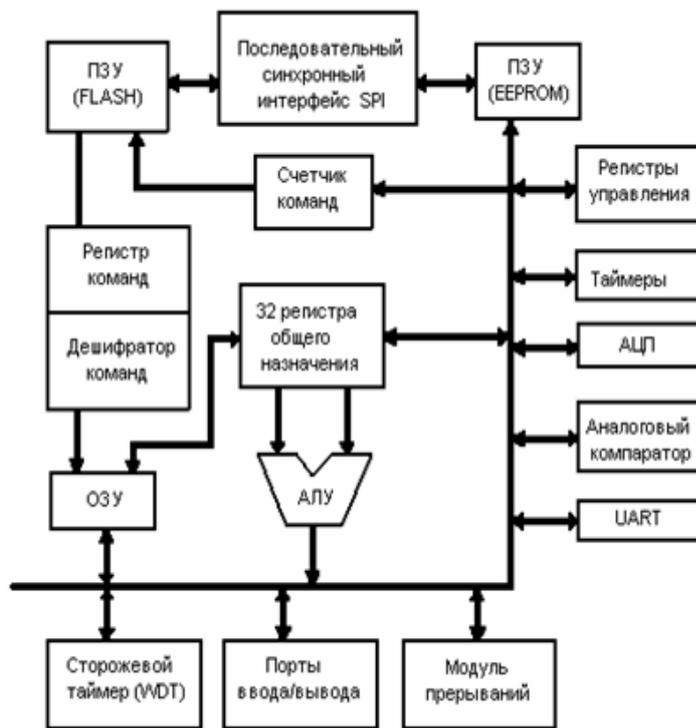


Рисунок – 2. Архитектура ядра микроконтроллера

Арифметико-логическое устройства (АЛУ) выполняет все вычисления и непосредственно подключено к 32 рабочим регистрам, объединенным в регистровый файл. Благодаря этому АЛУ выполняет одну операцию (чтение содержимого регистров, выполнение операции и запись результата обратно в регистровый файл) за один машинный цикл.

В микроконтроллерах AVR практически все команды (за исключением команд, у которых одним из операндов является 16-разрядный адрес) занимают одни ячейку памяти программ.

2 Используемые программы для разработки

2.1 Proteus

Мощнейшая система автоматизированного проектирования, позволяющая виртуально смоделировать работу огромного количества аналоговых и цифровых устройств.

Программный пакет Proteus VSM позволяет собрать схему любого электронного устройства и симулировать его работу, выявляя ошибки, допущенные на стадии проектирования и трассировки. Программа состоит из двух модулей. ISIS – редактор электронных схем с последующей имитацией их работы. ARES – редактор печатных плат, оснащенный автотрассировщиком Electra, встроенным редактором библиотек и автоматической системой размещения компонентов на плате. Кроме этого ARES может создать трехмерную модель печатной платы.

Proteus VSM включает в себя более 6000 электронных компонентов со всеми справочными данными, а также демонстрационные ознакомительные проекты. Программа имеет инструменты USBCONN и COMPIM, которые позволяют подключить виртуальное устройство к портам USB и COM компьютера. При подсоединении к этим портам любого внешнего прибора виртуальная схема будет работать с ним, как если бы она существовала в реальности. Proteus VSM поддерживает следующие компиляторы: CodeVisionAVR и WinAVR (AVR), ICC (AVR, ARM7, Motorola), HiTECH (8051, PIC Microchip) и Keil (8051, ARM). Существует возможность экспорта моделей электронных компонентов из программы PSpice.

Несмотря на то, что программа работает с устройствами, состоящими из нескольких микроконтроллеров и даже с чипами от разных производителей в одном устройстве, необходимо четко понимать, что симуляция повторяет

работу реальной схемы не абсолютно точно! Чтобы избежать ошибок, нужно ясно представлять конечный результат.

Proteus VSM является коммерческим продуктом. Есть бесплатная демонстрационная версия. Она обладает всеми функциями и возможностями платного пакета, но не позволяет сохранить или распечатать результат работы, создать свой собственный микроконтроллер.

Меню программы англоязычное. Полного русификатора для Proteus VSM нет. Устанавливать программу необходимо в папку без кириллических символов в названии.

2.2 Atmel Studio

Atmel Studio – интегрированная среда разработки (IDE) от компании Atmel для разработки приложений под микроконтроллеры ARM Cortex-M и AVR.

Программный пакет AVR Studio разрабатывается с 2004 года. Начиная с версии 6.0, программа сменила название на Atmel Studio. Программа позволяет работать как на ассемблере, так и на C/C++. Содержит в себе мастер проектов, виртуальный симулятор, редактор исходного кода, модуль внутрисхемной отладки и интерфейс командной строки. Поддерживает компилятор GCC и плагин AVR RTOS (операционной системы реального времени). Пользователи могут выбрать наиболее оптимальные для их проекта способы кодирования. Визуальные инструменты позволяют ускорить написание программы. Благодаря связке программных пакетов Atmel Studio и Proteus от фирмы Labcenter Electronics возможно программирование микроконтроллеров без наличия какой-либо материальной базы. Atmel Studio по праву считается лучшей средой создания приложений для контроллеров AVR.

3 Разработка кода

Для разработки кода были использованы данные букв ФИО, которые соответствуют бинарному коду.(См. приложение 1)

Таблица 1 – Шифр ФИО

с	к	о	с	ы	р	е	в	н	и	к	и	т	а	а	л	е	к	с	а	н	д	р	о	в	и	ч
1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	
0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1
0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1	1	0	1	0	1	0	0	0	0	1	1	0	1	0	0	1	1	0	0
0	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	0	1	0

4 Результаты работы

В результате работы был создан проект «представить каждую букву вашего ФИО в виде двоичного кода». В нем реализован проект на МК ATmega8 в котором дорожка из LED лампочек последовательно зажигается в соответствии с бинарным кодом ФИО.

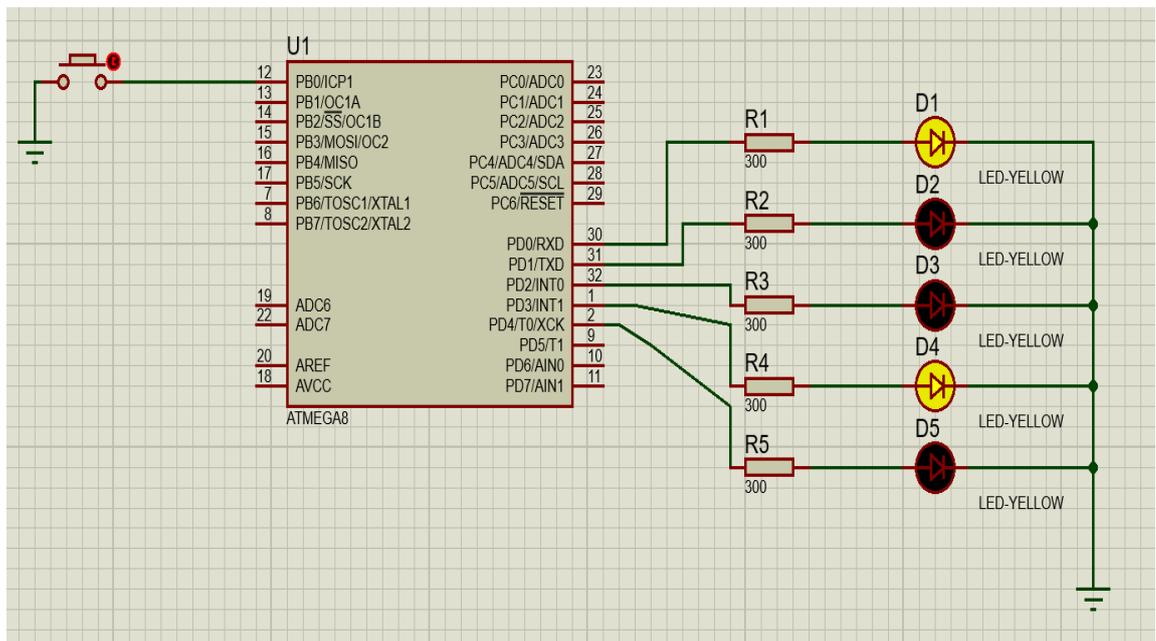


Рисунок – 3. Схема Проекта

Заключение

В результате работы были изучены основы работы с МК AVR, а в частности с ATmega8. Были изучены материалы по среде программирования микроконтроллеров и печатных плат. Бали использованы Proteus – пакет программ для автоматизированного проектирования (САПР) электронных схем так как он является более удобным и понятным. Он содержит все необходимые библиотеки и компоненты, которые могут понадобиться для первичного знакомства с ATmega8. И Atmel Studio – основанная на Visual Studio бесплатная проприетарная интегрированная среда разработки (IDE) для разработки приложений для 8- и 32-битных микроконтроллеров семейства AVR и 32-битных микроконтроллеров семейства ARM от компании Atmel

По средствам всего изученного материала был реализован проект «представить каждую букву вашего ФИО в виде двоичного кода». Который представлен в пункте 5 «результаты работы». Там представлен рабочий экземпляр. И в пункте 4 «разработка кода» был представлен код, который был написан в Atmel Studio и удачно экспортирован в Proteus.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сайт all-Arduino [Электронный ресурс] URL: <https://all-AVR.ru/> (Дата обращения: 6.07.2021).
2. Сайт wikipedia [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/AVR> (Дата обращения: 6.07.2021).
3. Сайт vse-o-mikrokontrollerakh [Электронный ресурс] URL: https://elektrovesti.net/tools/483_vse-o-mikrokontrollerakh-avr (Дата обращения: 6.07.2021).
4. Сайт schem.net [Электронный ресурс] URL: <https://schem.net/beginner/beginner73.php> (Дата обращения: 6.07.2021).
5. Сайт ustroystvo-mikrokontrollerov-avr [Электронный ресурс] URL: <http://elektrik.info/main/praktika/1333-ustroystvo-mikrokontrollerov-avr.html> (Дата обращения: 6.07.2021).

Приложение 1

```
#define F_CPU 1000000UL
#include <avr/io.h> //библиотеки
#include <util/delay.h> //библиотеки
#define K PORTB // присвоение имени порту
#define LED PORTD // присвоение имени порту
#define Z 300 //транзистор
int main(void)
{
    DDRB = 0b00000000; //начальное состояние портов
    DDRD = 0b11111111; //начальное состояние портов
    LED = 0b00000000; //начальное состояние портов
    K = 0b11111111; //начальное состояние портов
    while (1)
    { if (PINB == 0b11111110) //условие кнопки
        {
            PORTD = 0b00001001; // С
                _delay_ms(1000);
            PORTD = 0b00011010; // К
                _delay_ms(1000);
            PORTD = 0b00011110; // О
                _delay_ms(1000);
            PORTD = 0b00001001; // С
                _delay_ms(1000);
            PORTD = 0b00011011; // Ы
                _delay_ms(1000);
            PORTD = 0b00010001; // Р
```

```
_delay_ms(1000);  
PORTD = 0b00010100; // E  
_delay_ms(1000);  
PORTD = 0b00001000; // B  
_delay_ms(1000);  
PORTD = 0b00001110; // H  
_delay_ms(1000);  
PORTD = 0b00010010; // И  
_delay_ms(1000);  
PORTD = 0b00011010; // К  
_delay_ms(1000);  
PORTD = 0b00010010; // И  
_delay_ms(1000);  
PORTD = 0b00011001; // Т  
_delay_ms(1000);  
PORTD = 0b00000000; // А  
_delay_ms(1000);  
PORTD = 0b00000000; // А  
_delay_ms(1000);  
PORTD = 0b00000110; // Л  
_delay_ms(1000);  
PORTD = 0b00010100; // E  
_delay_ms(1000);  
PORTD = 0b00011010; // К  
_delay_ms(1000);  
PORTD = 0b00001001; // С  
_delay_ms(1000);  
PORTD = 0b00000000; // А  
_delay_ms(1000);  
PORTD = 0b00001110; // H
```

```

    _delay_ms(1000);
    PORTD = 0b00000100; // Д
    _delay_ms(1000);
    PORTD = 0b00010001; // Р
    _delay_ms(1000);
    PORTD = 0b00011110; // О
    _delay_ms(1000);
    PORTD = 0b00001000; // В
    _delay_ms(1000);
    PORTD = 0b00010010; // И
    _delay_ms(1000);
    PORTD = 0b00000011; // Ч
    _delay_ms(1000);
}
else
{
    LED = 0b00000000; //условие кнопки
}
}
}
}

```