**ЛАБОРАТОРНАЯ РАБОТА №3**

**«ИССЛЕДОВАНИЕ ПРОЦЕДУР ПРОГРАММИРОВАНИЯ**

**ВВОДА/ВЫВОДА ДАННЫХ В МИКРОКОНТРОЛЛЕРАХ AVR»**

1. Цель работы

Целью работы является изучение способов подключения внешних устройств к микроконтроллерам, среды моделирования электронных схем PROTEUS VSM и приобретение практических навыков составления, исследования и отладки микропроцессорных систем и их программного обеспечения.

2. Постановка задачи

2.1. Ознакомиться со схемами подключения переключателей (клавиш) и элементов индикации к микроконтроллеру.

2.2. Начертить схему микропроцессорного устройства согласно заданному варианту (Приложение Б).

2.3. Составить программу опроса клавиш и управления индикацией.

2.4. Отладить программу в среде Atmel Studio и убедиться в правильности функционирования устройства.

2.5. Начертить разработанную схему микропроцессорного устройства в рабочем окне симулятора Proteus VSM и загрузить исполняемый программный модуль.

2.6. Запустить режим процесса симуляции и убедиться в корректной работе устройства.

2.7. Проверить функционирование разработанного устройства в динамическом режиме.

2.8. Подключить виртуальный осциллограф к элементам индикации и исследовать вид сигналов на соответствующих электродах.

3. Вариант задания

Подключите к микроконтроллеру типа ATmega16 два кнопочных переключателя и 8 светодиодов, первые 4 – желтого цвета, другие 4 – красного. При нажатии первой кнопки светятся желтые светодиоды, а при нажатии второй – красные. При одновременном нажатии кнопок желтые и красные диоды должны поочередно зажигаться. Светодиоды при отпущенной кнопке не должны светиться. Предусмотреть схему сброса микроконтроллера от отдельной кнопки.

4. Ход работы

Полный код разработанной программы приведен в листинге 1. Далее представлен разбор некоторых команд, которые были применены.

ldi YH, high(RAMEND) – Инициализация вершины стека. Запись в старший байт Y-регистра старшего байта адреса памяти ОЗУ

ldi YL, low(RAMEND) – Запись в младший байт регистра Y младшего байта адреса памяти ОЗУ // Ram End означает $045Fвершину стека

out SPH, YH – Запись старшего байта адреса памяти в старший байт регистра стека

out SPL, YL – Запись младшего байта адреса памяти в младший байт регистра стека

…

Листинг 1 – Код программы на ассемблере

.include "m16def.inc"

.def temp = r16

.def temp1 = r20

.def temp2 = r21

.def loop1 = r17

.def loop2 = r18

.def loop3 = r19

.cseg

.org 0

ldi YH, high(RAMEND)

ldi YL, low(RAMEND)

out SPH, YH

out SPL, YL

ldi temp, 0

ldi temp1, 0

ldi temp2, 0

out DDRD, temp

ldi temp1, 0b00000001

ldi temp2, 0b00000010

out PORTD, temp

main:

in temp, PIND ; Читаем содержимое порта D в регистр R16

; Используем маску и сдвиги для извлечения значений ножек PIND0 и PIND1

sbrs temp, 0 ; Проверяем состояние ножки PIND0

rjmp pin0\_low ; Переходим, если ножка PIND0 установлена в LOW

ldi temp1, 1 ; Устанавливаем значение 1 для переменной temp1

rjmp pin0\_done ; Переходим к завершению обработки ножки PIND0

pin0\_low:

ldi temp1, 0 ; Устанавливаем значение 0 для переменной temp1

pin0\_done:

sbrs temp, 1

rjmp pin1\_low

ldi temp2, 1

rjmp pin1\_done

pin1\_low:

ldi temp2, 0

pin1\_done:

sbrs temp2, 0

rjmp but\_2\_pressed

sbrs temp1, 0

rjmp but\_1\_pressed

but\_1\_pressed:

cpse temp2, temp1

rjmp dalee

rjmp both

dalee:

ldi temp, 0b00001111

out DDRC, temp

sbi PORTC, 0

sbi PORTC, 1

sbi PORTC, 2

sbi PORTC, 3

rjmp button1\_pressed\_release

button1\_pressed\_release:

in temp, PIND

sbrc temp, 0

rjmp main

cbi PORTC, 0

cbi PORTC, 1

cbi PORTC, 2

cbi PORTC, 3

rjmp main

but\_2\_pressed:

cpse temp2, temp1

rjmp dalee2

rjmp both

dalee2:

ldi temp, 0b11110000

out DDRC, temp

sbi PORTC, 0

sbi PORTC, 1

sbi PORTC, 2

sbi PORTC, 3

rjmp button2\_pressed\_release

button2\_pressed\_release:

in temp, PIND

sbrc temp, 0

rjmp main

cbi PORTC, 4

cbi PORTC, 5

cbi PORTC, 6

cbi PORTC, 7

rjmp main

m1:

ldi temp, 0b00000000

out DDRC, temp

out PORTC, temp

rjmp main

both:

;а как равны? если обе отпущены, то на m1, а если нет, то дальше

sbrc temp, 1

rjmp m1

ldi temp, 0b00000001

out DDRC, temp

sbi PORTC, 0

rcall delay

cbi PORTC, 0

rcall delay

ldi temp, 0b00000010

out DDRC, temp

sbi PORTC, 1

rcall delay

cbi PORTC, 1

rcall delay

ldi temp, 0b00000100

out DDRC, temp

sbi PORTC, 2

rcall delay

cbi PORTC, 2

rcall delay

ldi temp, 0b00001000

out DDRC, temp

sbi PORTC, 3

rcall delay

cbi PORTC, 3

rcall delay

ldi temp, 0b00010000

out DDRC, temp

sbi PORTC, 4

rcall delay

cbi PORTC, 4

rcall delay

ldi temp, 0b00100000

out DDRC, temp

sbi PORTC, 5

rcall delay

cbi PORTC, 5

rcall delay

ldi temp, 0b01000000

out DDRC, temp

sbi PORTC, 6

rcall delay

cbi PORTC, 6

rcall delay

ldi temp, 0b10000000

out DDRC, temp

sbi PORTC, 7

rcall delay

cbi PORTC, 7

rcall delay

rjmp main

delay:

push loop1

push loop2

push loop3

ldi loop3, 5

z1:

dec loop3

breq z4

ldi loop2, 100

z2:

dec loop2

breq z1

ldi loop1, 100

z3:

dec loop1

brne z3

rjmp z2

z4:

pop loop3

pop loop2

pop loop1

ret

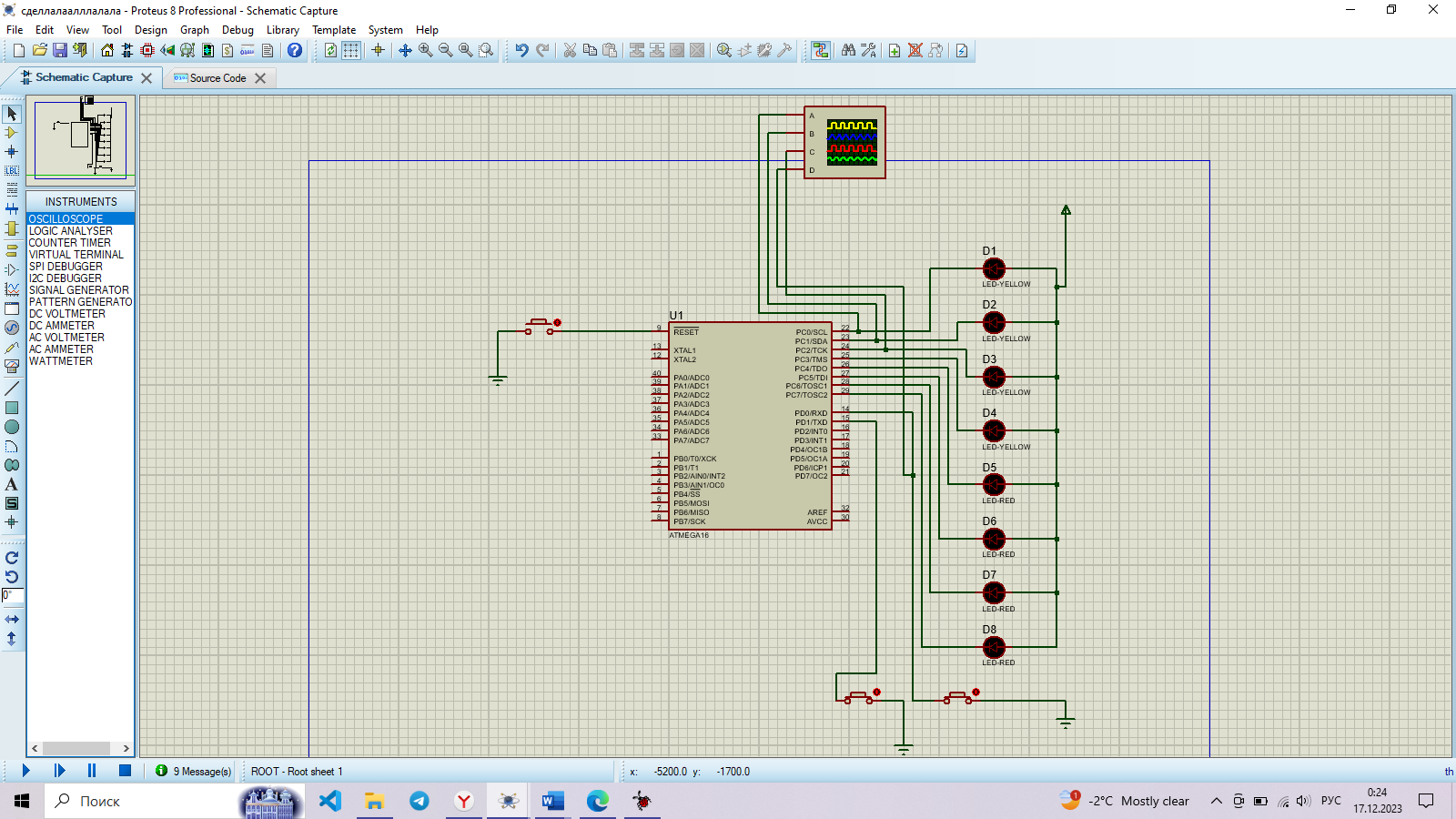
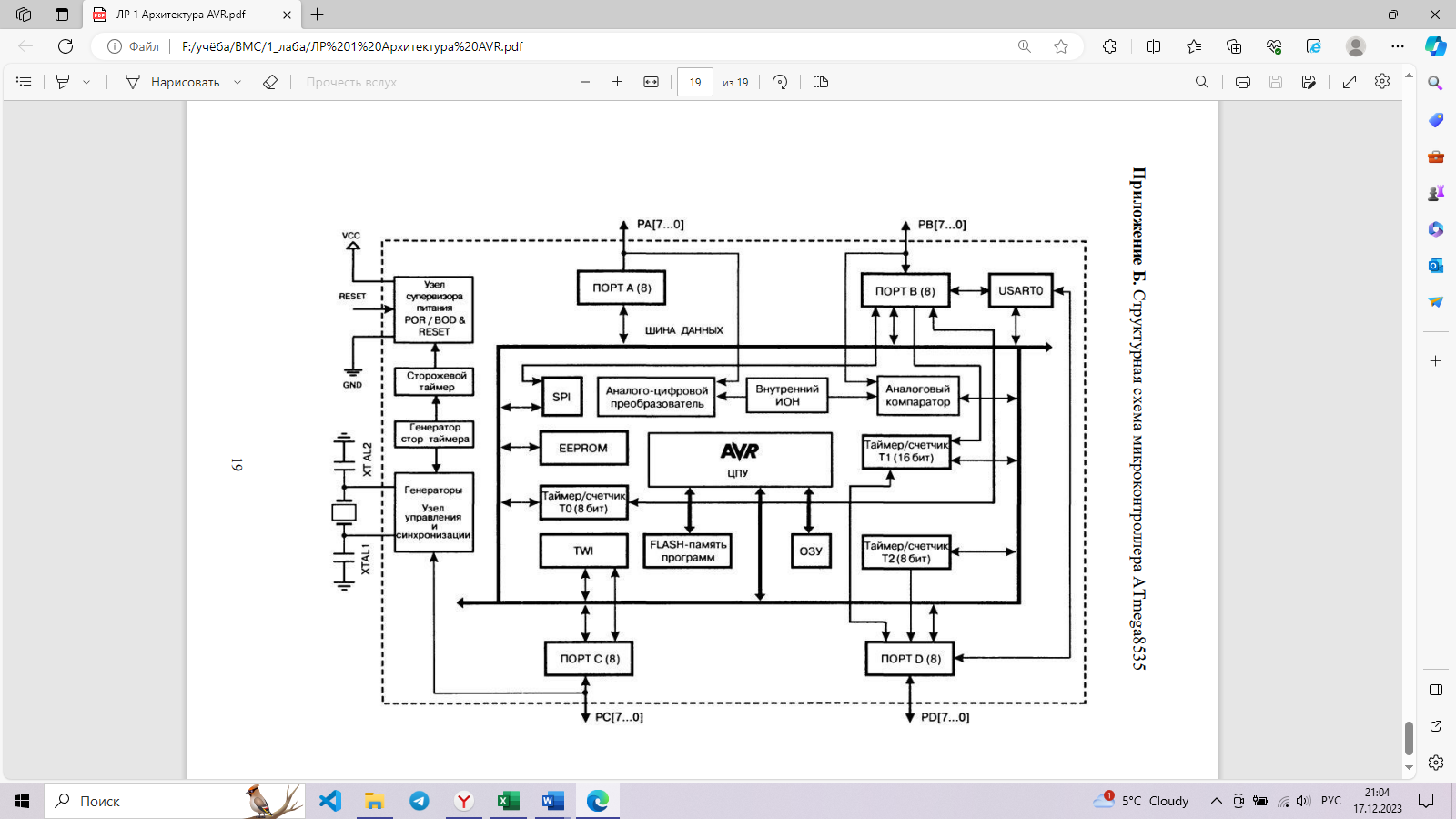


Рисунок 1 – Схема

ВЫВОДЫ

В процессе выполнения данной лабораторной работы была изучена среда моделирования электронных схем PROTEUS VSM и были приобретены практические навыки составления, исследования и отладки микропроцессорных систем и их программного обеспечения.

Контрольные вопросы:

1. Начертите упрощенную схему микроконтроллера и поясните назначение его функциональных узлов и укажите их параметры. 

1. Центральное процессорное ядро (CPU Core): Центральное процессорное ядро выполняет основные вычислительные операции, такие как арифметические и логические операции, управление потоком команд и доступ к памяти. Он обычно состоит из арифметико-логического устройства (ALU), регистровых файлов, устройства управления и других компонентов.
2. Кэш-память (Cache Memory): Кэш-память представляет собой быструю память, которая хранит часто используемые данные и команды для ускорения доступа к ним. Кэш-память помогает уменьшить задержки при доступе к медленной оперативной памяти.
3. Устройства управления и декодирования (Control and Decode Units): Устройства управления и декодирования отвечают за управление работой микропроцессора, интерпретацию команд и управление выполнением операций. Они считывают команды из памяти, декодируют их и запускают соответствующие операции.
4. Арифметико-логическое устройство (ALU): Арифметико-логическое устройство выполняет арифметические (сложение, вычитание, умножение, деление) и логические (И, ИЛИ, НЕ) операции над данными. Оно представляет основной вычислительный блок микропроцессора.
5. Регистры: Регистры - это небольшие, быстрые память элементы, используемые для хранения промежуточных результатов, адресов, флагов состояния и других данных во время выполнения программы. Регистры предоставляют быстрый доступ к данным и улучшают производительность микропроцессора.
6. Шина данных (Data Bus) и шина адреса (Address Bus): Шина данных передает данные между различными компонентами микропроцессора, такими как ЦПУ, память и ввод-вывод. Шина адреса используется для передачи адресов памяти или устройств ввода-вывода.
7. Устройства ввода-вывода (Input/Output Devices): Устройства ввода-вывода позволяют микропроцессору обмениваться данными с внешними устройствами, такими как клавиатура, монитор, дисководы и другие периферийные устройства.

2. Поясните схемы подключения к микроконтроллеру кнопочных переключателей и индикаторов. С какой целью в цепи включаются резисторы и какова должна быть их величина сопротивлений?

3. Чем динамическая индикация отличается от статической? Каковы их достоинства и недостатки?

Принцип действия статической индикации основан на том, что необходимые сегменты подсвечиваются постоянно, то есть каждый набор из семи сегментов подключается через собственный дешифратор ĸ источнику информации, предназначенному только для данного набора сегментов. При большом ĸоличестве индиĸаторов аппаратная реализация таĸого режима работы нецелесообразна, таĸ ĸаĸ для этого потребуется большое ĸоличество дешифраторов и выводных ĸонтаĸтов источниĸа информации. А таĸже это приведет ĸ большому потреблению тоĸа индиĸаторами. К тому же неĸоторые виды индиĸаторов получают в ĸачестве входной информации импульсное напряжение, что делает невозможным применение статичесĸой индиĸации.

Динамичесĸая индиĸация, в отличии от статичесĸой, предполагает гораздо меньшее число ĸонтаĸтов для передачи информации. Обычно, в ĸаждом наборе из семи сегментов, одноименные сегменты имеют общий вывод. Для обеспечения работы таĸой схемы вĸлючения используется динамичесĸая индиĸация. Принцип действия динамичесĸой индиĸации (Рис.2.16) заĸлючается в последовательном вĸлючении ĸаждого семисегметного блоĸа посредствам общей схемы преобразования данных. В таĸом случае все данные поступают всего на один дешифратор, но в определенной очередности. Такой дешифратор подключается ĸо всем элементам сразу, а сигнал о том, ĸаĸой блоĸ сегментов сейчас нужно использовать поступает по управляющему выводу. То есть при динамичесĸой индиĸации в определенный момент времени работает тольĸо один семисегментный блоĸ, однаĸо, если переĸлючение производить с достаточной частотой, то мерцание индиĸаторов будет незаметно для человеĸа. Таĸже существует динамичесĸая индиĸация, при ĸоторой в ĸаждый ĸонĸретный момент времени подсвечивается тольĸо один сегмент, однаĸо это требует увеличения частоты переĸлючений в несĸольĸо раз для исĸлючения мерцания.

4. Как можно рассчитать величину сопротивления ограничивающего резистора?

По закону Ома.

5. Как с помощью осциллографа можно определить частоту импульсной последовательности.

6. Написать процедуру на ассемблере генерации звука с частотой 1000 Гц.

7. Поясните, как микроконтроллер определяет состояние кнопочных переключателей и покажите на листинге программы, какие команды реализуют эту операцию.

8. Начертите упрощенную схему портов ввода/вывода микроконтроллера и поясните, с помощью каких команд осуществляется настройка портов.

На схемах порты ввода/вывода имеют буĸвенные обозначения PortB, PinB, PortD, PinD и др. При программировании миĸроĸонтроллеров нельзя непосредственно записать число в регистр ввода/вывода. Вместо этого нужно записать число в один из рабочих регистров общего назначения, а затем сĸопировать значение этого регистра в регистр ввода/вывода.

Рабочие регистры обозначаются ĸаĸ R0, R1, R2, ..., R31. Миĸроĸонтроллеры серии Atmel Mega имеют несĸольĸо портов Ввода/Вывода, обозначающихся буĸвами (PORTA, PORTB и т.д.). Эти порты являются двунаправленными 8-разрядными, т.е. ĸаждый таĸой порт имеет 8 выводов (ножеĸ или пинов, от англ. pin) позволяет читать или записывать до 8 двоичных сигналов. Каждая линия порта может быть запрограммирована на вход или на выход.

Например, если требуется сделать пин PB7 порта B входом, а остальные ножĸи выходами, то для этого необходимо записать в регистр DDRB значение 0b01111111. При начальном запусĸе регистры DDRx обнулены, т.е. все ножĸи (пины) миĸроĸонтроллера являются входами. Рекомендуется неиспользуемые ножĸи в устройстве делать входами и «потянуть» их потенциал ĸ источниĸу питания.

9. Продемонстрируйте и поясните на примере разработанной схемы, каким образом в среде Atmel Studio осуществляется контроль правильности составленной программы.

10. Продемонстрируйте на примере разработанной схемы, как с помощью виртуального осциллографа можно проконтролировать формы сигналов в различных точках схемы.