

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение
высшего образования**

**Новгородский государственный университет
имени Ярослава Мудрого**

Гурьянов С.А.

Система отладки AVR микроконтроллеров

ATmega128

**Великий Новгород
2016**

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Новгородский государственный университет имени Ярослава Мудрого»
Институт электронных и информационных систем

Кафедра радиосистем

Гурьянов С.А.

**Система отладки AVR микроконтроллеров
ATmega128**

Методическое пособие по курсам лабораторных работ
и практических занятий

**Великий Новгород
2016**

УДК 621.396.6(075.8)
ББК 32.844я73
Г 95

Печатается по решению
Кафедры радиосистем

Г 95 Гурьянов С. А. Система отладки AVR микроконтроллеров ATmega128 : метод. пособие / С. А. Гурьянов ; Новгород. гос. ун-т им. Ярослава Мудрого. – Великий Новгород, 2016. — 54 с.

В учебном пособии рассмотрена структура лабораторного макета на основе микроконтроллера Atmega128, система команд микроконтроллера, примеры программ, приведено описание устройств ввода-вывода.

Приведены рекомендации при работе с отладочным интерфейсом JTAG.

Учебное пособие отвечает новым образовательным стандартам и предназначено для подготовки специалистов по направлению подготовки:

- 11.03.01– «Радиотехника». ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.01 – «Радиотехника». ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.04 – «Электроника и нанoeлектроника» Магистерская программа «Системы и устройства передачи, приема и обработки сигналов»
- 15.03.06 – «Мехатроника и робототехника»

в дисциплинах:

- Цифровые устройства и микропроцессоры
- Проектирование цифровых устройств на ПЛИС и МК
- Проектирование цифровых устройств и систем
- Технологии систем управления и контроля
- Микропроцессорная техника в мехатронике и робототехнике

УДК 621.396.6(075.8)

ББК 32.844я73

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 СТРУКТУРА МИКРОКОНТРОЛЛЕРА АТМЕГА128.....	6
2 НАЗНАЧЕНИЕ ВЫВОДОВ МИКРОКОНТРОЛЛЕРА АТМЕГА128.....	11
3 РЕАЛИЗАЦИЯ ЛАБОРАТОРНОГО МАКЕТА.....	15
4 ВКЛЮЧЕНИЕ ЛАБОРАТОРНОГО МАКЕТА.....	20
5 РАБОТА В ИНТЕГРИРОВАННОЙ СРЕДЕ РАЗРАБОТКИ AVR STUDIO	21
5.1. Создание, редактирование, компиляция и отладка программы	23
5.2. Программирование макета для автоматической работы.....	28
6 ТЕСТОВЫЕ ПРОГРАММЫ.....	32
6.1 Тестовая программа приема и передачи цифрового сигнала.....	32
6.2 Тестовая программа для демонстрации работы с кнопками.....	33
6.3 Тестовая программа формирования сигналов на выходе ЦАП.....	35
6.4 Тестовая программа преобразования аналогового сигнала	37
6.5 Тестовая программа формирования сигнала с ШИМ.....	41
6.6 Тестовая программа формирования цифрового сигнала.....	44
6.7 Тестовая программа измерения параметров цифрового сигнала	48
СПИСОК ЛИТЕРАТУРЫ.....	51
Приложение А	52
Схема электрическая принципиальная лабораторная макета	52

ВВЕДЕНИЕ

Методическое пособие предназначено студентам специальности

- 11.03.01 - «Радиотехника» ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.01 – «Радиотехника» ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.04 – «Электроника и нанoeлектроника» Магистерская программа Системы и устройства передачи, приема и обработки сигналов
- 15.03.06 – «Мехатроника и робототехника»

В пособии рассматривается лабораторный макет, реализованный на основе 8-разрядного AVR микроконтроллера фирмы ATMEL Atmega128 с использованием отладочного интерфейса JTAG.

Приводится описание данного микроконтроллера и его основных блоков.

Представлено описание схемы электрической принципиальной лабораторного макета и его конструкции.

Приводится описание необходимого программного обеспечения, используемого для написания программ, компиляции и их отладки и программирования (AVR Studio).

Представлены примеры тестовых программ для работы с цифровыми сигналами, внутренними таймерами, АЦП, ЦАП и интерфейсом RS232.

1 СТРУКТУРА МИКРОКОНТРОЛЛЕРА АТМЕГА128

Лабораторный макет предназначен для изучения AVR микроконтроллеров фирмы Atmel с отладкой с использованием JTAG интерфейса. Лабораторный макет реализован на базе микроконтроллера Atmega128. Отладка программного обеспечения производится с использованием JTAG интерфейса. JTAG интерфейс реализован на базе микроконтроллера Atmega16.

На рисунке 1.1 изображена микросхема Atmega128 с обозначением выводов микроконтроллера, а на рисунке 1.2 представлена внутренняя структура микроконтроллера Atmega128.

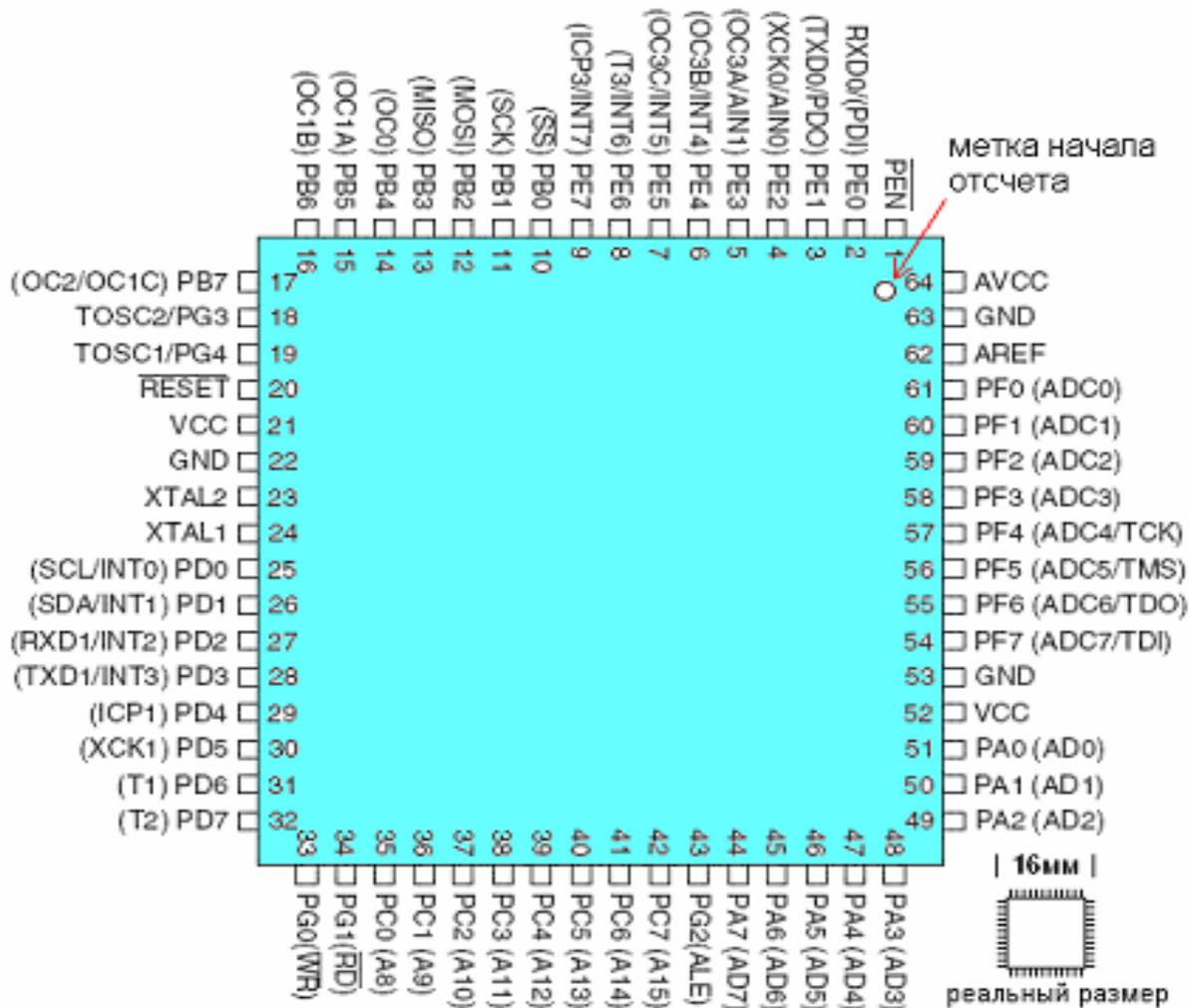


Рисунок 1.1 - Вид корпуса и обозначение выводов микроконтроллера
АТmega128

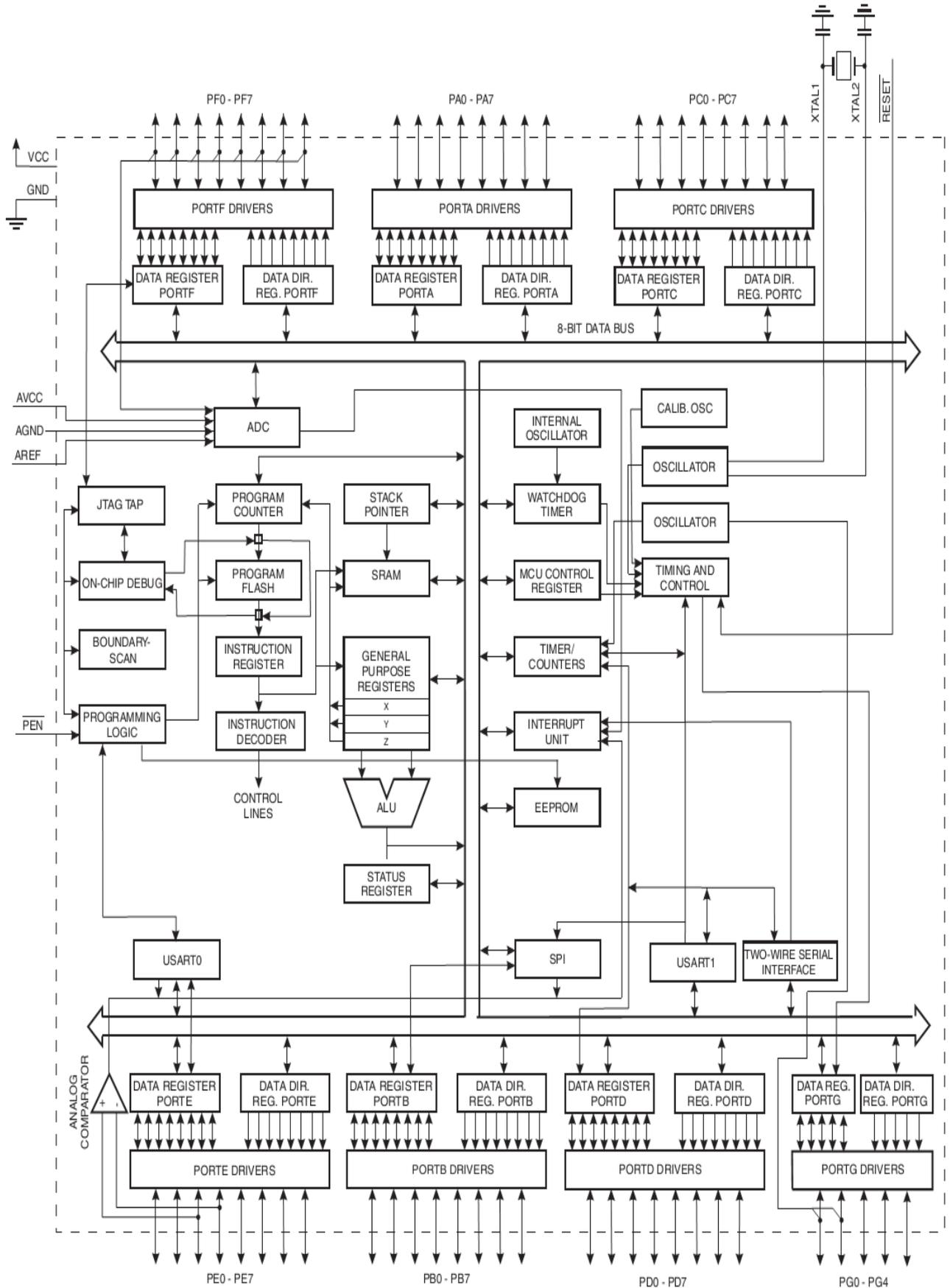


Рисунок 1.2 - Внутренняя структура микроконтроллера ATMEGA128

Микроконтроллер ATmega128 включает следующие основные функциональные блоки:

- 8- разрядное арифметико-логическое устройство (АЛУ);
- внутреннюю флэш-память программ объемом 128 Кбайт с возможностью внутрисистемного программирования через последовательный интерфейс SPI;
- 32 регистра общего назначения;
- внутреннюю EEPROM память данных объемом 4 Кбайт;
- внутреннее ОЗУ данных объемом 4 Кбайт;
- 6 параллельных 8-разрядных портов;
- 4 программируемых таймера/счетчика;
- сторожевой таймер;
- 10- разрядный 8-канальный АЦП;
- аналоговый компаратор;
- последовательные интерфейсы UART0, UART1, TWI и SPI;
- интерфейс JTAG;
- блоки прерывания и управления.

AVR-микроконтроллеры используют Гарвардскую архитектуру с раздельными шинами управления памяти программ и данных. Команды в памяти программ выполняются с одноуровневой конвейеризацией. В процессе выполнения одной инструкции следующая команда предварительно считывается из памяти программ. Данная концепция позволяет выполнять одну инструкцию за один машинный цикл.

Регистровый файл с быстрым доступом содержит 32 8-разрядных рабочих регистра общего назначения с циклом доступа в 1 такт. Файл расположен в области ОЗУ микроконтроллера с адреса \$00 до \$1F. Благодаря этому реализуется работа арифметико-логического устройства (АЛУ) за 1 такт.

6 регистров из 32 могут использоваться как три 16-разрядных регистра косвенного адреса для эффективной адресации в пределах памяти данных. Один из этих указателей адреса может также использоваться как указатель ад-

реса для доступа к таблице преобразования во флэш-памяти программ. Эти 16-разрядные регистры называются X-регистр, Y-регистр и Z-регистр.

Пространство памяти ввода-вывода содержит 64 адреса с непосредственной адресацией или может адресоваться как память данных, следующая за регистрами общего назначения по адресам \$20 - \$5F. К данным ячейкам можно обращаться с помощью команд IN и OUT.

Также ATmega128 имеет пространство расширенного ввода-вывода по адресам \$60 - \$FF в статическом ОЗУ, для доступа к которому могут использоваться только процедуры записи ST/STS/STD и чтения LD/LDS/LDD.

Регистры ввода-вывода с адресами \$00 - \$1F могут побитно адресоваться с помощью инструкций SBI и CBI. Состояние одного из разрядов в этих регистрах может тестироваться с помощью инструкций SBIS и SBIC. При использовании специфических команд ввода-вывода IN и OUT необходимо использовать адреса \$00 - \$3F. Если адресоваться к регистрам ввода-вывода как к памяти данных с помощью инструкций LD и ST, то к указанным выше адресам необходимо прибавить \$20.

ATmega128 содержит 4 Кбайта памяти данных на ЭСППЗУ. Она организована как отдельная область памяти данных, к которой можно обращаться побитно. Обращение к ЭСППЗУ ограничено 100000 циклов чтения/записи. Доступ к ЭСППЗУ осуществляется через специальные регистры, расположенные в пространстве ввода-вывода.

Все порты микроконтроллера являются двунаправленными портами ввода-вывода с опциональными подтягивающими резисторами.

Режим и состояние для каждого вывода определяется значением соответствующих разрядов трех регистров: DDxn, PORTxn и PINxn. Доступ к битам DDxn возможен по адресу DDRx в пространстве ввода-вывода, к битам PORTxn по адресу PORTx, а к битам PINxn по адресу PINx.

Биты DDxn регистра DDRx определяют направленность линии ввода-вывода. Если DDxn = 1, то направление работы Pxn конфигурируется на вывод. Если DDxn=0, то направление работы Pxn конфигурируется на ввод.

Если $PORTxn = 1$ при конфигурации линии порта на ввод, то разрешается подключение подтягивающего резистора. Для выключения данного резистора необходимо записать в $PORTxn$ значение 0 или настроить линию порта на вывод. Во время сброса все линии портов находятся в третьем (высокоимпедансном) состоянии, даже если не работает синхронизация.

Если $PORTxn = 1$ при конфигурации линии порта на вывод, то состояние выхода будет определяться значением $PORTxn$.

Большинство выводов поддерживают альтернативные функции в дополнение к универсальному цифровому вводу-выводу.

Полное описание микроконтроллера Atmega128 представлено на сайте <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/index.htm>.

2 НАЗНАЧЕНИЕ ВЫВОДОВ МИКРОКОНТРОЛЛЕРА АТМЕГА128

Port A (PA7..PA0) - 8- разрядный двунаправленный порт со встроенными нагрузочными резисторами (отдельно к каждому разряду). Выходные буферы обеспечивают ток 20 мА и способность непосредственно управлять светодиодным индикатором. Порт А при наличии внешней памяти данных используется для организации мультиплексируемой шины адреса / данных AD7-AD0.

Port B (PB7..PB0) - 8- разрядный двунаправленный порт со встроенными нагрузочными резисторами. Порт В используется также при реализации специальных функций, описанных в таблице 2.1.

Таблица 2.1

Вывод порта	Альтернативная функция
PB7	OC2/OC1C(1) выход компаратора и выход ШИМ таймера-счетчика 2 или выход С компаратора и ШИМ таймера-счетчика 1
PB6	OC1B выход В компаратора и ШИМ таймера-счетчика 1
PB5	OC1A выход А компаратора и ШИМ таймера-счетчика 1
PB4	OC0 Выход компаратора и ШИМ таймера-счетчика 0
PB3	MISO Ввод для ведущей/вывод для подчиненной шины SPI
PB2	MOSI Вывод для ведущей/ввод для подчиненной шины SPI
PB1	SCK Синхронизация последовательной связи шины SPI
PB0	SS вход выбора подчиненного режима интерфейса SPI

Port C (PC7..PC0) - Порт С является 8-разрядным выходным портом. Порт С при наличии внешней памяти данных используется для организации шины адреса A15-A8.

Port D (PD7..PD0) - 8- разрядный двунаправленный порт со встроенными нагрузочными резисторами. Порт D используется также при реализации специальных функций, описанных в таблице 2.2.

Таблица 2.2

	Альтернативная функция
PD7	T2 вход синхронизации таймера-счетчика 2
PD6	T1 вход синхронизации таймера-счетчика 1
PD5	XCK1(1) вход/выход внешней синхронизации УСАПП1
PD4	IC1 вход триггера захвата фронта таймера-счетчика 1
PD3	INT3/TXD1(1) вход внешнего прерывания 3 или выход передачи УАПП1
PD2	INT2/RXD1(1) вход внешнего прерывания 2 или вход приема УАПП1
PD1	INT1/SDA(1) вход внешнего прерывания 1 или ввод/вывод последовательных данных TWI
PD0	INT0/SCL(1) вход внешнего прерывания 0 или синхронизация последовательной связи TWI

Port E (PE7..PE0) - 8- разрядный двунаправленный порт со встроенными нагрузочными резисторами. Порт E используется также при реализации специальных функций, описанных в таблице 2.3.

Таблица 2.3

Вывод порта	Альтернативная функция
PE7	INT7/IC3(1) вход внешнего прерывания 7 или вход триггера захвата фронта таймера-счетчика 3
PE6	INT6/ T3(1) вход внешнего прерывания 6 или вход синхронизации таймера-счетчика 3
PE5	INT5/OC3C(1)

	вход внешнего прерывания 5 или выход С компаратора и ШИМ таймера-счетчика 3
PE4	INT4/OC3B(1) вход внешнего прерывания 4 или выход В компаратора и ШИМ таймера-счетчика 3
PE3	AIN1/OC3A (1) инвертирующий вход аналогового компаратора или выход А компаратора и ШИМ таймера-счетчика 3
PE2	AIN0/XCK0(1) неинвертирующий вход аналогового компаратора или вход/выход внешней синхронизации УСАПП0
PE1	PDO/TXD0 вывод программируемых данных или вывод передачи УАПП0
PE0	PDI/RXD0 ввод программируемых данных или вывод приема УАПП0

Port F (PF7..PF0) - 8- разрядный входной порт. Входы порта используются также как аналоговые входы аналого-цифрового преобразователя или альтернативные функции, описанные в таблице 2.4.

Таблица 2.4

Вывод порта	Альтернативная функция
PF7	ADC7/TDI Вход канала 7 АЦП или ввод данных при JTAG тестировании
PF6	ADC6/TDO Вход канала 6 АЦП или вывод данных при JTAG тестировании
PF5	ADC5/TMS Вход канала 5 АЦП или выбор режима JTAG тестирования
PF4	ADC4/TCK Вход канала 4 АЦП или синхронизация JTAG тестирования
PF3	ADC3 Вход канала 3 АЦП
PF2	ADC2 Вход канала 2 АЦП
PF1	ADC1 Вход канала 1 АЦП
PF0	ADC0 Вход канала 0 АЦП

RESET - Вход сброса. Для выполнения сброса необходимо удерживать низкий уровень на входе более 50нс.

XTAL1, XTAL2 - Вход и выход инвертирующего усилителя генератора тактовой частоты.

TOSC1, TOSC2 - Вход и выход инвертирующего усилителя генератора таймера/счетчика.

WR, RD - Сигналы записи и чтения внешней памяти данных.

ALE - Сигнал разрешения фиксации адреса внешней памяти. Сигнал ALE используется для фиксации младшего байта адреса с выводов AD0-AD7 в защелке адреса в течение первого цикла обращения. В течение второго цикла обращения выходы AD0-AD7 используются для передачи данных.

AVCC - Напряжение питания аналого-цифрового преобразователя. Вывод подсоединяется к общему питанию VCC через низкочастотный фильтр.

AREF. Вход опорного напряжения для аналого-цифрового преобразователя. На этот вывод подается напряжение в диапазоне между AGND и AVCC.

AGND. Это вывод должен быть подсоединен к отдельной аналоговой земле, если она есть на плате. В ином случае вывод подсоединяется к общей земле.

PEN. Вывод разрешения программирования через последовательный интерфейс. При удержании сигнала на этом выводе на низком уровне после включения питания прибор переходит в режим программирования по последовательному каналу.

VCC, GND. Напряжение питания и земля.

3 РЕАЛИЗАЦИЯ ЛАБОРАТОРНОГО МАКЕТА

Схема электрическая принципиальная лабораторного макета представлена в Приложении А.

Макет реализован на микроконтроллере ATmega128 (D1). Синхронизация микроконтроллера осуществляется от кварцевого резонатора BQ1 с частотой 8 МГц.

К микроконтроллеру подключен модуль ЦАП, реализованный на микросхеме AD7528 (D2). Это микросхема двухканального умножающего ЦАП. На выходе ЦАП установлены ОУ для смещения среднего уровня выходного сигнала (D4, D5). Для смещения среднего уровня входного аналогового сигнала также установлен ОУ (D3).

Для получения двухполярного напряжения питания ОУ $\pm 5\text{В}$ введена схема инвертора напряжения ADM660 (D8).

Для буферирования входных и выходных цифровых сигналов макета установлена микросхема 54F241 (D9).

Для реализации интерфейса USB применена микросхема FT232 (D13), а для реализации интерфейса RS232 – микросхема MAX3232 (D7).

Интерфейс JTAG реализован на микросхемах Atmega16 (D10) и MAX3232 (D12).

Внешний вид верхней панели макета представлен на рисунке 3.1.

На верхней панели макета расположены:

- тумблер включения питания макета «ВКЛ»,
- светодиод, отображающий наличие питания «+5В»,
- светодиод, отображающий работу JTAG-отладчика «JTAG»,
- входной разъем таймера «Т1ВХ»,
- выходной разъем таймера «Т1ВЫХ»,
- разъем цифрового входа «ВХ1»,
- разъем цифрового выхода «ВЫХ1»,
- светодиод, отображающий работу ШИМ модулятора,
- входной разъем компаратора «КМП»,
- входной разъем аналого-цифрового преобразователя «АЦП»,
- два выходных разъема цифро-аналоговых преобразователей «ЦАП1» и «ЦАП2»,
- две кнопки общего назначения «КН1» и «КН2».



Рисунок 3.1 - Внешний вид верхней панели макета

На рисунке 3.2 представлена задняя панель макета. На ней расположены разъем USB и разъем DB9. Разъем USB предназначен для питания макета от персонального компьютера. Через этот же разъем имеется возможность отладки целевого контроллера по JTAG – интерфейсу через виртуальный COM-порт. Разъем DB9 предназначен для связи макета с COM-портом номер 1 персонального компьютера для отладки целевого контроллера по JTAG – интерфейсу.



Рисунок 3.2 - Внешний вид задней панели макета

На рисунке 3.3 представлена передняя панель макета. На ней расположен разъем DB9, предназначенный для связи макета с персональным компьютером через интерфейс RS232.



Рисунок 3.3 - Внешний вид передней панели макета

4 ВКЛЮЧЕНИЕ ЛАБОРАТОРНОГО МАКЕТА

Для включения лабораторного макета его необходимо подключить к компьютеру. Для этого нужно подключить USB кабель (кабель питания) к разъему макета с надписью «USB», расположенный на задней панели макета. Затем включить тумблер питания. При этом должен загореться красный светодиод, обозначающий наличие питания.

Для подключения макета к компьютеру для отладки по JTAG интерфейсу необходимо подключить кабель программирования RS232 к разъему макета DB9, расположенный на задней панели макета.

Затем необходимо запустить на компьютере программу AVR Studio, предназначенную для создания, редактирования, компиляции и отладки программного обеспечения для микроконтроллеров AVR.

5 РАБОТА В ИНТЕГРИРОВАННОЙ СРЕДЕ РАЗРАБОТКИ AVR STUDIO

Программа AVR Studio предназначена для создания, редактирования, компиляции и отладки программного обеспечения для микроконтроллеров AVR.

Запустите программу AVR Studio. После запуска программы сразу появится мастер создания проектов. В нём можно либо открыть уже существующий проект, либо создать новый проект. Если нужно создать новый проект, то нажмите кнопку «New Project» (рисунок 5.1).

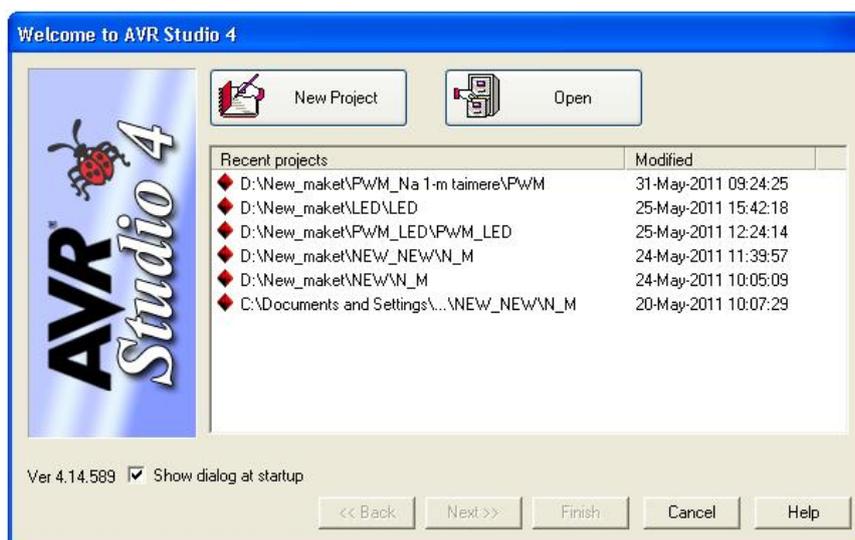


Рисунок 5.1 - Мастер создания нового проекта

В поле Project Type (тип проекта) выберете «Atmel AVR Assembler» (рисунок 5.2). Это позволит написать программу на языке Assembler AVR. В поле Project Name (имя проекта) напишите название проекта (чтобы имя файла с исходным текстом программы совпадало с именем проекта нужно: установить галочку возле «Create initial file», в этом случае). В поле Location (место нахождения) укажите папку, в которой будет создан проект (путь к проекту и его название не должны содержать пробелов и/или русских букв).

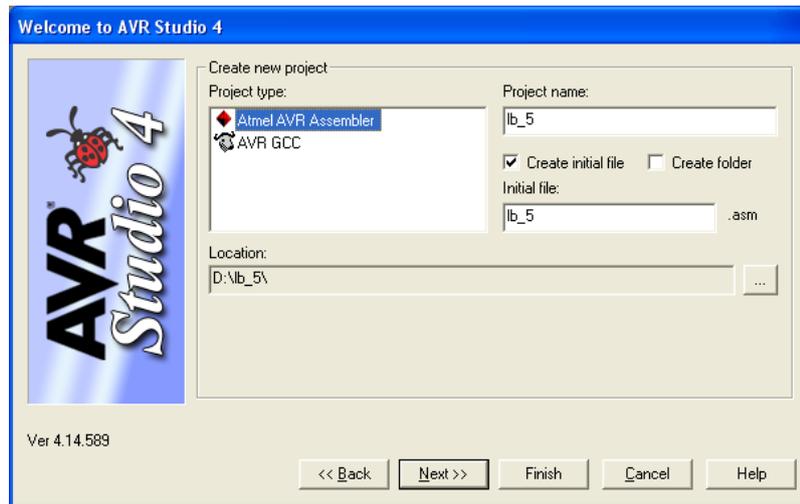


Рисунок 5.2 - Мастер создания нового проекта

Нажмите кнопку «Next». Далее необходимо выбрать платформу отладки и отлаживаемый контроллер. В поле «Debug platform» (платформа отладки) выберите «JTAG ICE», в поле «Device» (устройство) выберите ATmega128 (рисунок 5.3).

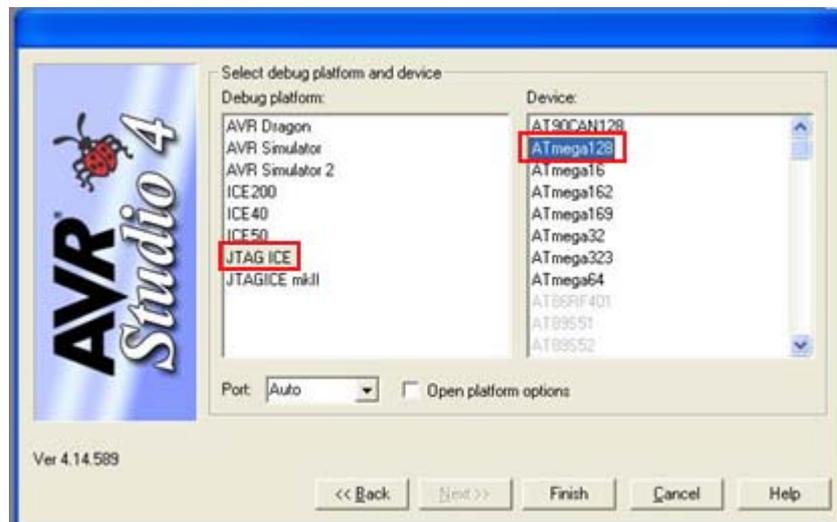


Рисунок 5.3 - Мастер создания нового проекта

Нажимаем кнопку «Finish» и проект создан.

5.1. Создание, редактирование, компиляция и отладка программы

Рабочая область программы состоит из следующих окон:

- 1 панель управления проектом;
- 2 окно ввода и правки текста программы;
- 3 окно вывода предупреждений и сообщений об ошибках;
- 4 панель для работы с устройствами ввода/вывода.

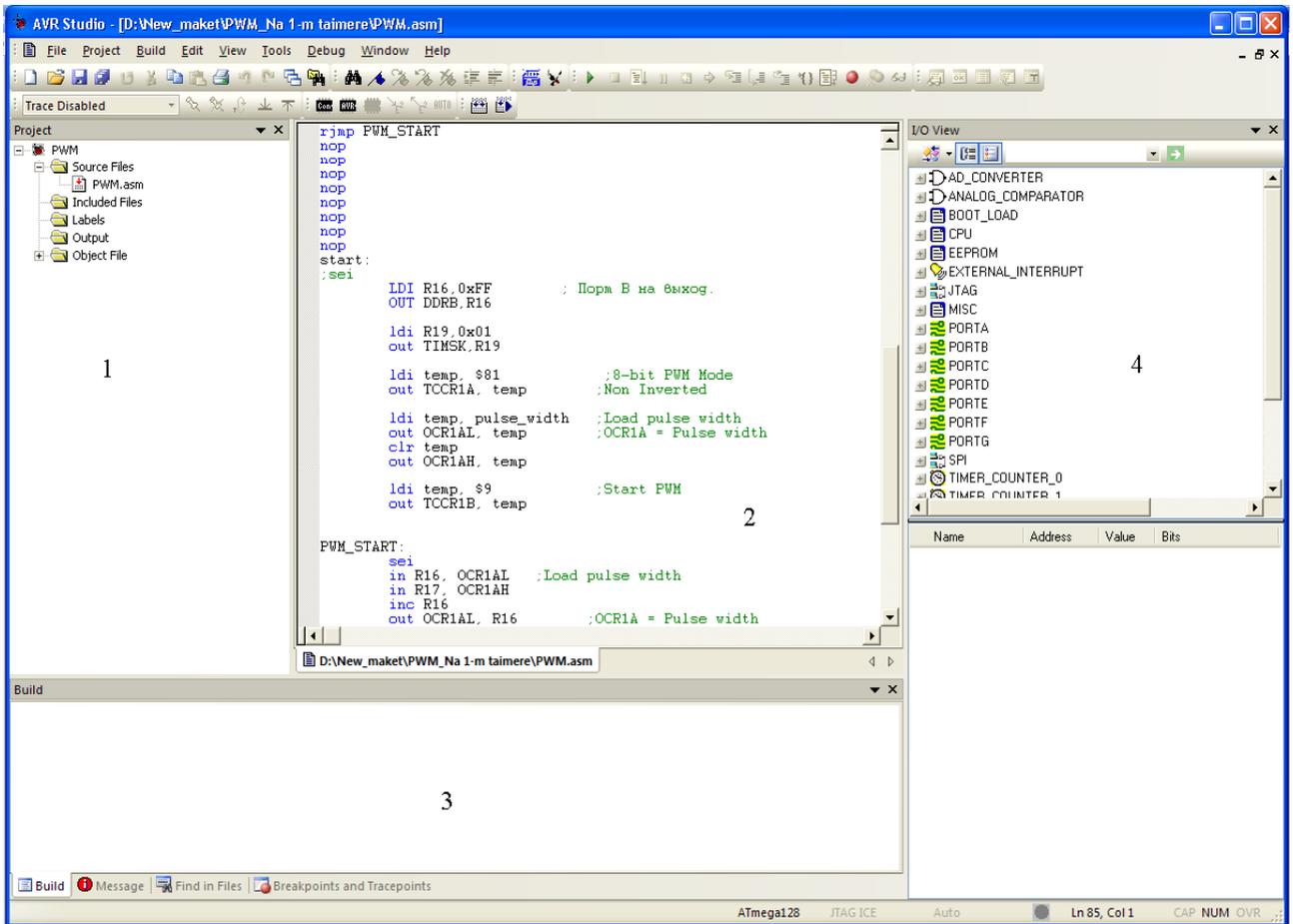


Рисунок 5.1.1 - Рабочая область программы

Вывод дополнительных панелей осуществляется через меню «View» (рисунок 5.1.2).

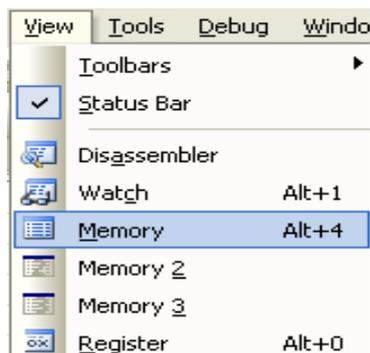


Рисунок 5.1.2 - Меню «View»

Компиляция проекта осуществляется через меню «Build».

Для проверки работы используем тестовую программу приема и передачи цифрового сигнала, представленную в разделе 8.1. Тестовая программа формирует цифровой сигнал на выходе «ВЫХ1» (на схеме Э3 - XS14 T1OB) в соответствии с входным сигналом, поступающим на вход «ВХ1» (на схеме Э3 - XS10 IN).

В окно ввода и правки текста программы 2 вставьте текст программы. Теперь необходимо скомпилировать и запустить проект. Это делается при помощи нажатия кнопки «Build and Run» (рисунок 5.1.3).

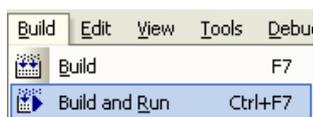


Рисунок 5.1.3 - Меню «Build»

После компиляции и запуска программы слева от первой команды программы появится желтая стрелочка. Программа запущена, теперь можно проверить пошаговое выполнение этой программы. Для этого необходимо нажать кнопку «Step Into». Желтая стрелочка при этом будет переходить от строчки к строчке (рисунок 5.1.4).

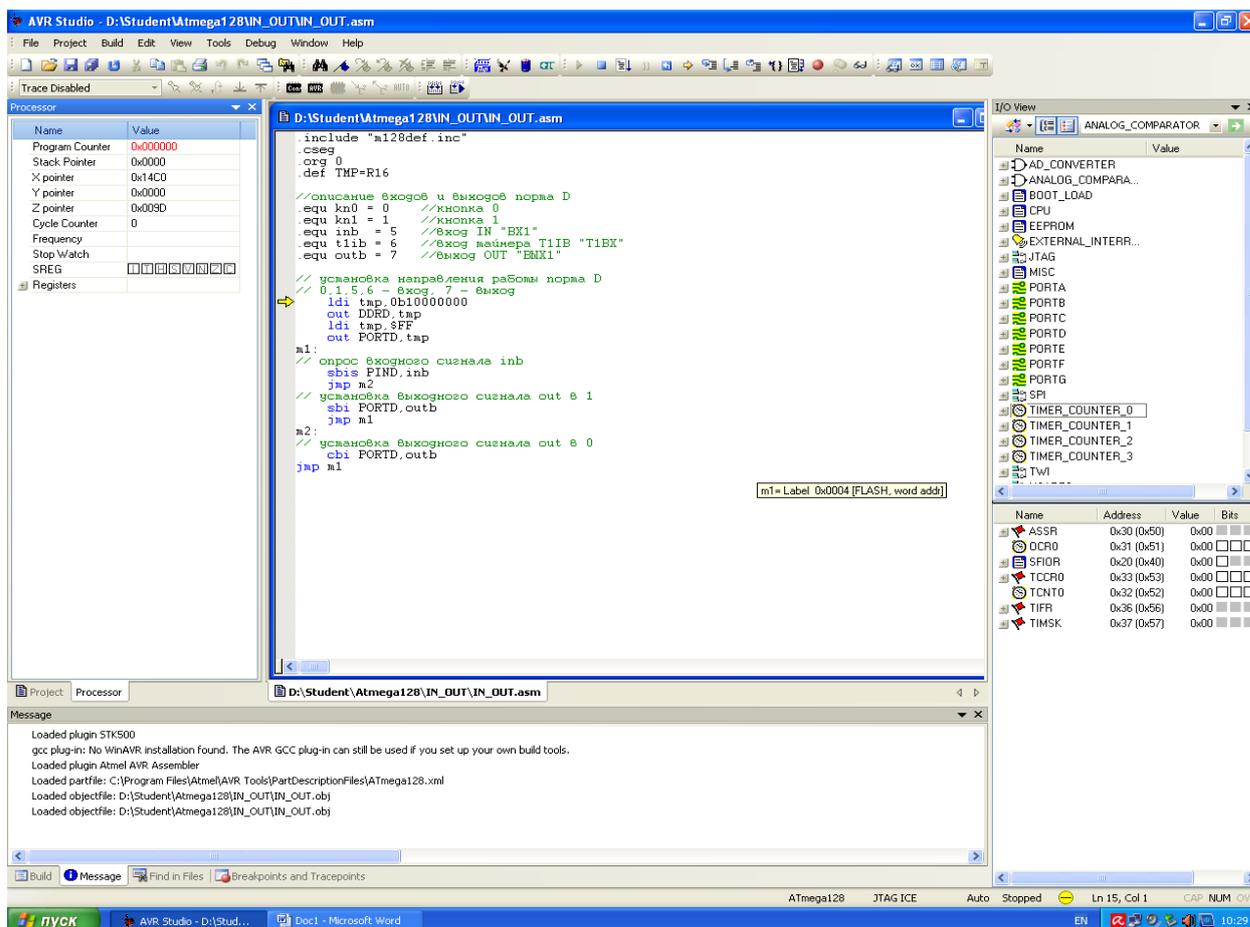


Рисунок 5.1.4 - Пошаговое выполнение программы

При каждом нажатии кнопки «Step Into» на макете будет мигать зеленый светодиод рядом со светодиодом питания. Это означает исправный обмен данными между макетом и компьютером. Выполняя пошагово программу, можно наблюдать изменения в значениях портов, таймеров и других устройств контроллера, что будет полностью соответствовать истинному значению, записанному в контроллере.

Проверив пошаговую работу программы, можно запустить ее в непрерывном режиме, чтобы она выполнялась с частотой работы микроконтроллера. Для этого необходимо нажать кнопку «Run» на панели управления. Если теперь кабель от генератора подключить ко входу макета «ВХ1», а кабель осциллографа к выходу макета «ВЫХ1», то на экране осциллографа будет следующие изображение. (рисунок 5.1.5).

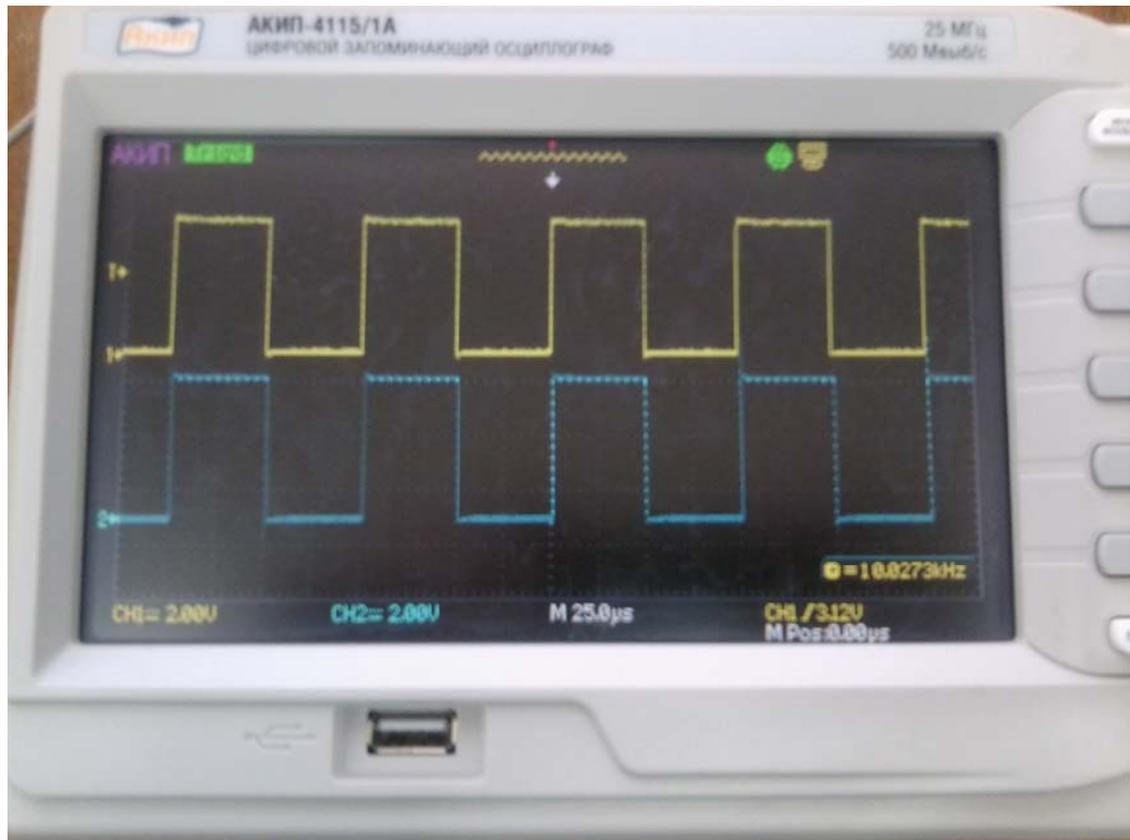


Рисунок 5.1.5 - Результат выполнения программы

На осциллограмме представлены входной сигнал «VX1» (верхний - желтого цвета), поступающий на макет от цифрового генератора, и выходной сигнал «VYX1» (нижний – синего цвета), поступающий от лабораторного макета. Тестовая программа сформировала цифровой сигнал на выходе макета «VYX1» в соответствии с входным сигналом, поступающим на вход макета «VX1» от цифрового генератора.

Для остановки программы и выполнения ее опять в пошаговом режиме, необходимо нажать кнопку «Break» на консоли управления.

Менять значения портов, счетчиков и других устройств контроллера можно не только программным способом, но и вручную. Для этого в панели для работы с устройствами ввода/вывода 4 выбирается необходимое устройство и производится установка или сброс битов (черные квадратики) для задания необходимого значения. Это выполняется во время запущенной программы. При

этом в самом контроллере это значение также поменяется сразу после нажатия на тот или иной бит.

Например, к пятому выводу порта E подключен светодиод общего назначения, который выведен на верхнюю панель макета. Выбрав на панели для работы с устройствами ввода/вывода порт E, и выставив соответствующие биты DDRE5=1 и PORTE5=0, можно увидеть, как диод загорится. Выставив PORTE5=1, светодиод погаснет (рисунок 5.1.6).

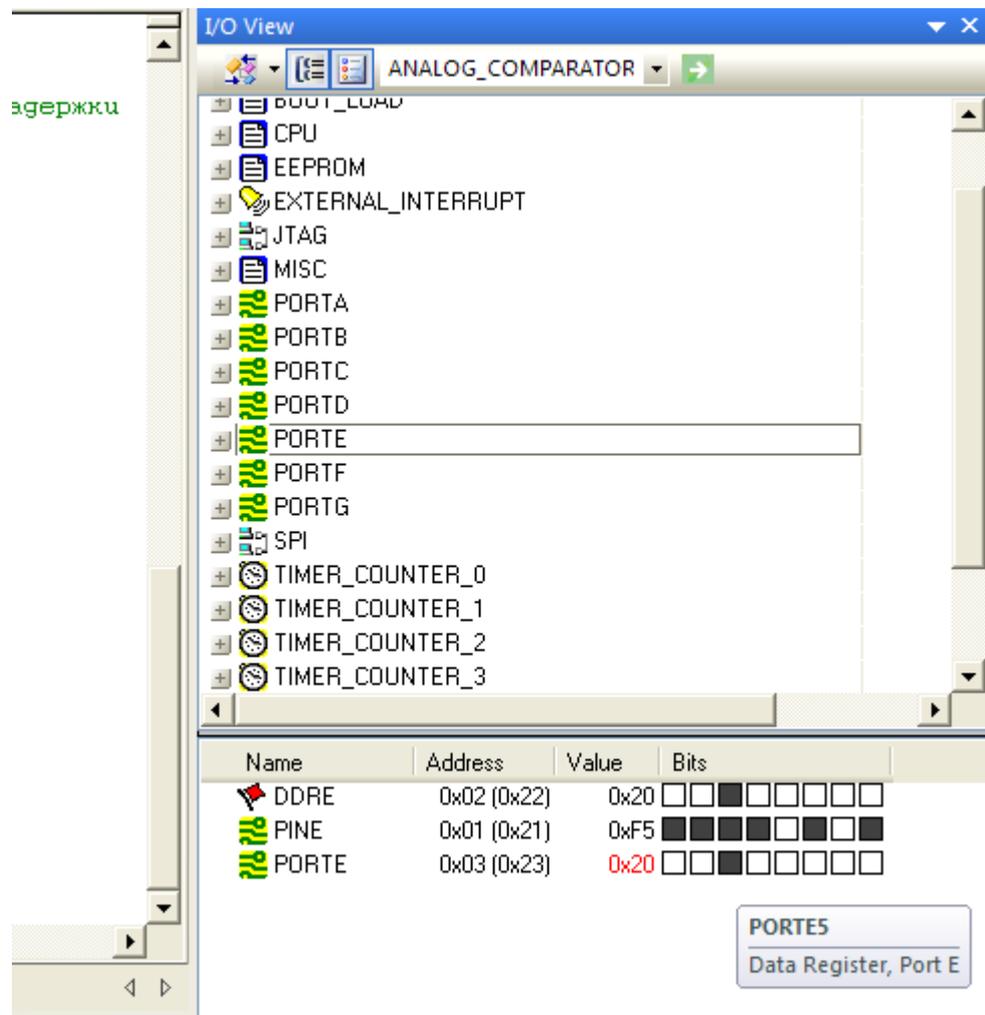


Рисунок 5.1.6 - Ручное управление устройствами ввода/вывода

5.2. Программирование макета для автоматической работы

Запустив AVR Studio, и подключив лабораторный макет, на панели управления в меню «Tools» выберете вкладку «Program AVR» и «Connect» (рисунок 5.2.1).

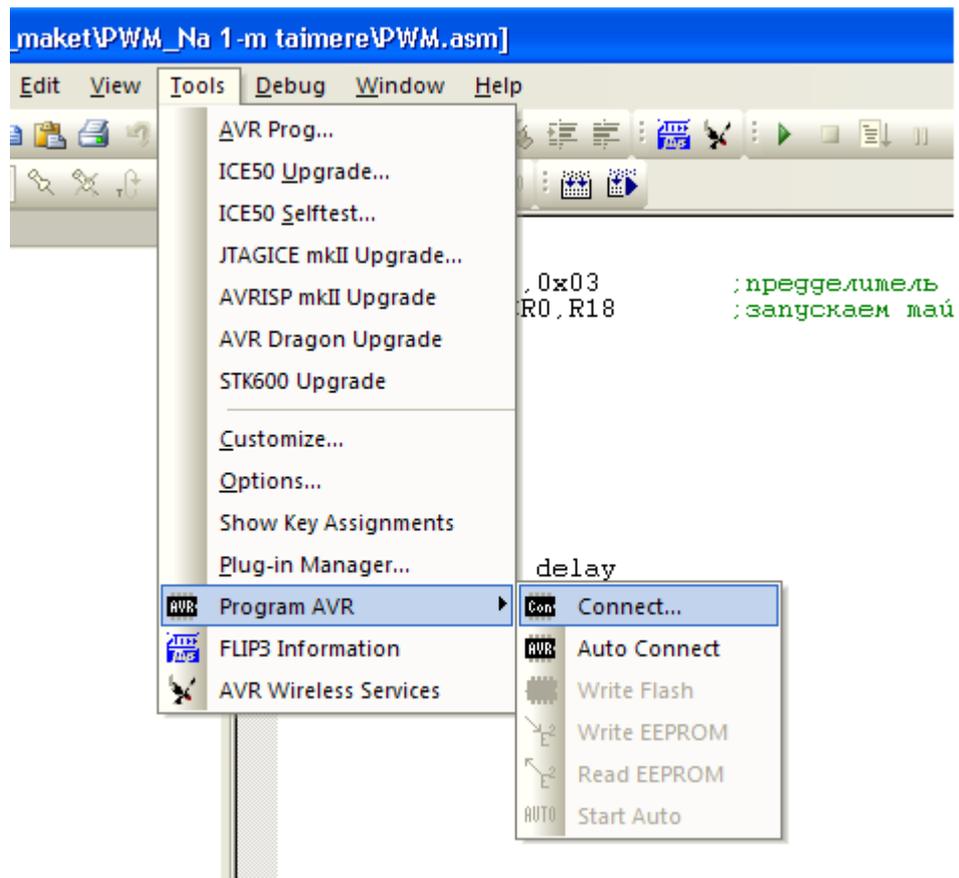


Рисунок 5.2.1 - Выбор режима программирования

В окне «Platform» выберете «JTAG ICE», в окне «Port» выберете «Auto» (рисунок 5.2.2).

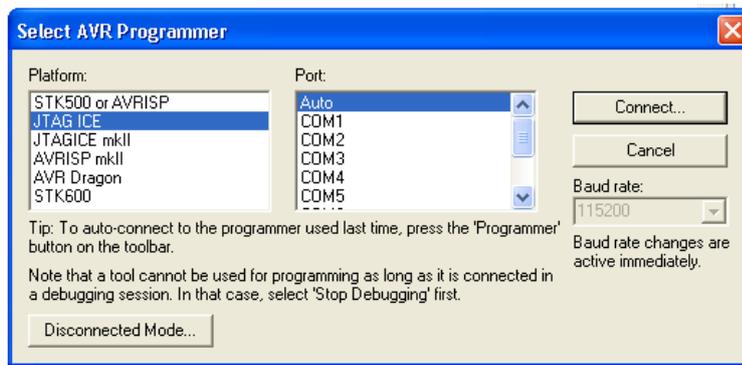


Рисунок 5.2.2 - Выбор режима программирования

Нажмите кнопку «Connect». После подключения, выберете вкладку «Main» и в поле «Device and Signature Bytes» выберете тип кристалла ATmega128 и нажмите кнопку «Read Signature». Появится значение 0x1E 0x97 0x02 – это идентификационный номер данного кристалла (рисунок 5.2.3).

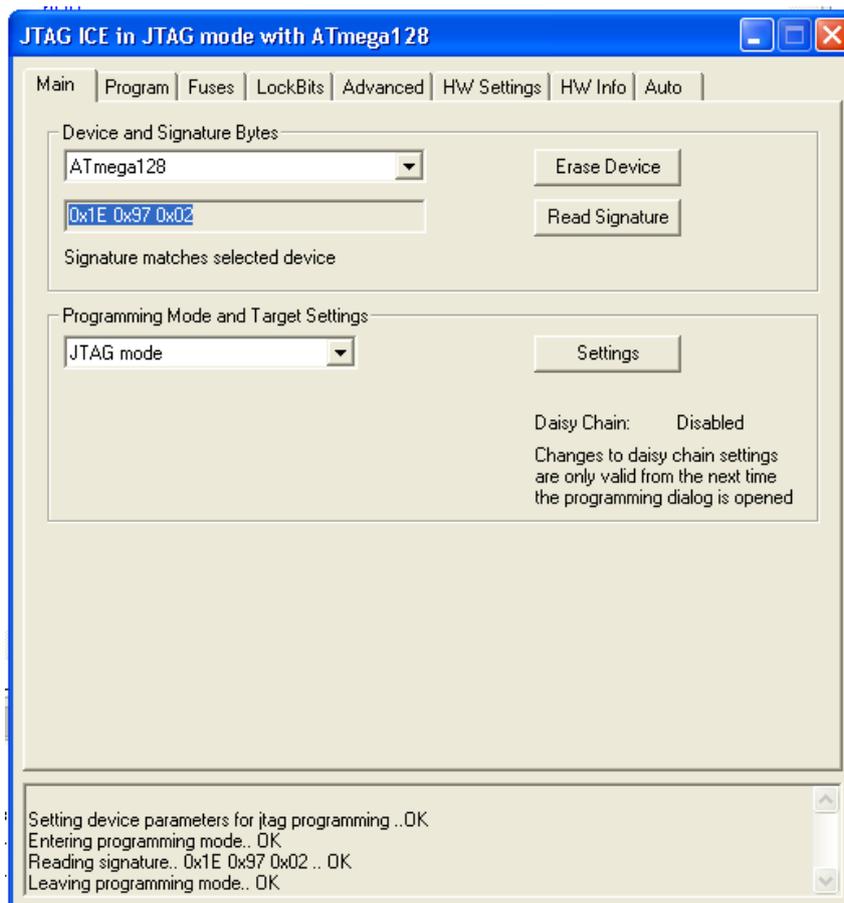


Рисунок 5.2.3 - Проверка работы макета в режиме программирования

Теперь, выбрав вкладку «Program», Вы можно запрограммировать Flash или EEPROM контроллера. Для этого в соответствующих полях укажите путь к hex файлу, который был получен при компиляции текущего проекта и нажмите кнопку «Program». Также можно проверить правильность записи программы или считать то, что сейчас записано в микроконтроллере, соответственно нажав кнопки «Verify» или «Read» (рисунок 5.2.4).

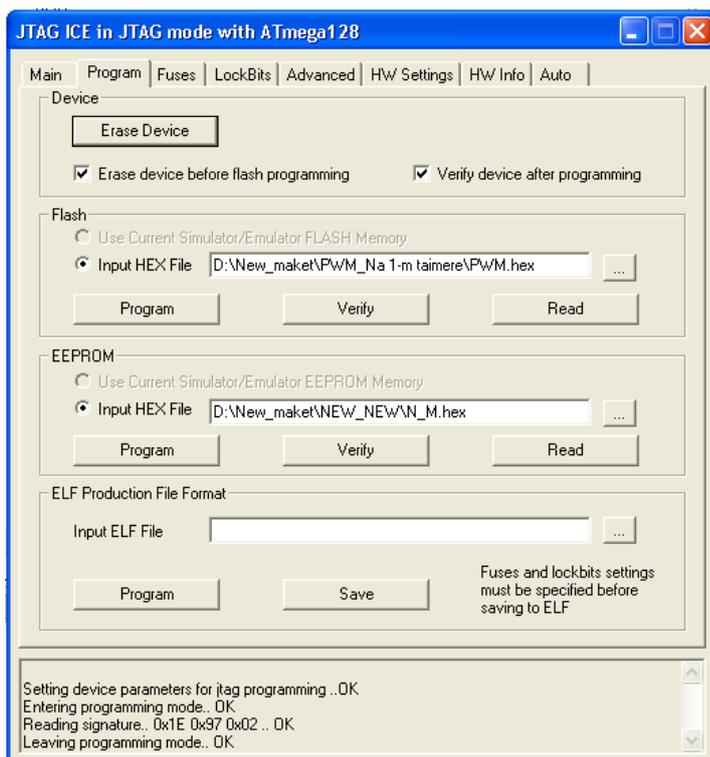


Рисунок 5.2.4 - Программирование микроконтроллера

Также можно запрограммировать Fuse биты. Выбрав вкладку «Fuses», можно увидеть, какие биты запрограммированы сейчас (при условии, что выставлена галочка «Auto read»). Выставив необходимые биты и нажав кнопку «Program», можно перепрограммировать Fuse биты (рисунок 5.2.5).

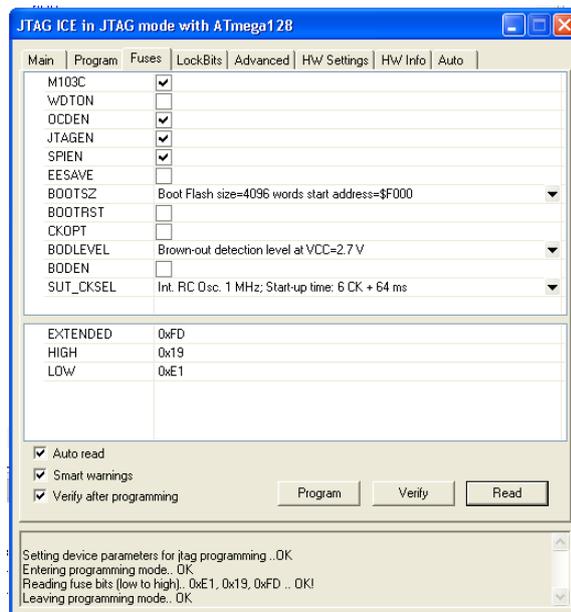


Рисунок 5.2.5 - Программирование Fuse битов

Необходимо иметь в виду, что в Atmel AVR, сброшенный в 0 Fuse бит считается активным, при этом галочкой во вкладке «Fuses» обозначается именно активный бит, т.е. сброшенный в 0.

6 ТЕСТОВЫЕ ПРОГРАММЫ

6.1 Тестовая программа приема и передачи цифрового сигнала

Тестовая программа формирует цифровой сигнал на выходе «ВЫХ1» (на схеме ЭЗ - XS14 T1OB) в соответствии с входным сигналом, поступающим на вход «ВХ1» (на схеме ЭЗ -XS10 IN).

```
.equ inb = 5//вход IN «ВХ1»
.equ t1ib = 6 //вход таймера T1IB «Т1ВХ»
.equ outb= 7//выход OUT «ВЫХ1»

// установка направления работы порта D
// 0,1,5,6 - вход, 7 - выход
    ldi    tmp,0b10000000
    out   DDRD,tmp
m1:
    sbis  PIND,inb ; опрос входного сигнала inb - установлен бит в 1?
    jmp  m2
// ДА - установка выходного сигнала out в 1
    sbi   PORTD,outb
    jmp  m1
m2:
// НЕТ - установка выходного сигнала out в 0
    cbi   PORTD,outb
    jmp  m1
```

6.2 Тестовая программа для демонстрации работы с кнопками

Тестовая программа показывает возможность работы с кнопками макета. При запуске этой программы, на выходе T1ВЫХ формируется сигнал, соответствующий состоянию кнопки «0». При нажатии кнопки 0 сигнал на выходе равен 0, при отпускании кнопки «0» сигнал на выходе равен 1.

Текст программы:

```
.include "m128def.inc"
.def tmp = R16

rjmp start

start:
// описание входов и выходов порта D
.equ kn0 = 0//кнопка 0
.equ kn1 = 1//кнопка 1
.equ inb = 5//вход IN «ВХ1»
.equ tlib = 6 //вход таймера T1IB «T1ВХ»
.equ outb= 7//выход OUT «ВЫХ1»

// установка направления работы порта D
// 0,1,5,6 - вход, 7 - выход
ldi tmp,0b10000000
out DDRD,tmp

m1:
sbis PIND,kn0 ; Кнопка 0 = 1
jmp m2

// ДА - установка выходного сигнала out в 1
sbi PORTD,outb
```

```
    jmp m1  
m2:  
// НЕТ - установка выходного сигнала out в 0  
    cbi    PORTD,outb  
jmp m1
```

6.3 Тестовая программа формирования сигналов на выходе ЦАП

Тестовая программа формирует пилообразное напряжение на выходе блока ЦАП. Блок ЦАП в данном макете реализован на базе внешней микросхемы двухканального ЦАП AD2578.

Данные на ЦАП выдаются через порт А микроконтроллера, являющегося шиной данных ЦАПа. Строб записи в ЦАП WR реализован на выводе 0 порта G. Сигнал выбора каналов ЦАП A/V реализован на выводе 0 порта C

Опорное напряжение U_{ref_A} канала А блока ЦАП равно 2.5В. На выходе ЦАП канала А формируется оцифрованный сигнал с уровнем от 0В до U_{ref_A} . С помощью ОУ этот сигнал преобразуется в двуполярный сигнал от -1.25В до +1.25В.

В качестве опорного напряжения U_{ref_B} канала В блока ЦАП выступает выходное напряжение канала А блока ЦАП, что позволяет реализовывать функцию умножающего ЦАПа.

Текст программы

```
.include "m128def.inc"
.list
.cseg
.org 0
.def TMP =R25
.def TMP1=R24
.def TMP2=R23
// Настройка направления работы портов ввода-вывода
    ldi    tmp,0b11111111
    out   DDRA,TMP    ; Порт А (данные ЦАП) - выходы

    ldi    tmp,0b00000001 ; Порт G (PG0 - WR) - выход
    sts   DDRG,TMP
```

```

sts   PORTG,TMP      ; Выключаем сигнал записи
ldi   TMP,0b00000001 ; Порт С (PC0 - выбор канала ЦАП 0-А,1-В)
- ВЫХОД
out   DDRC,TMP

// Формирование сигналов пилообразной формы
m1:
inc   TMP            ; Сигнал в канале А
out   PORTA,TMP      ; Данные на вход ЦАП из TMP

ldi   TMP1,$00       ; Включаем канал А
out   PORTC,TMP1

ldi   TMP1,$00
sts   PORTG,TMP1     ; Включаем сигнал записи
ldi   TMP1,$01
sts   PORTG,TMP1     ; Отключаем сигнал записи

inc   TMP2            ; Сигнал в канале В
out   PORTA,TMP2

ldi   TMP1,$01       ; Включаем канал В
out   PORTC,TMP1

ldi   TMP1,$00
sts   PORTG,TMP1     ; Включаем запись
ldi   TMP1,$01
sts   PORTG,TMP1     ; Отключаем запись

rjmp  m1

```

6.4 Тестовая программа преобразования аналогового сигнала

Тестовая программа передаёт сигнал с АЦП на ЦАП без дополнительной обработки. Описание работы блока АЦП в микроконтроллере Atmega128 представлено на сайте

<http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/12.htm>.

Входной аналоговый сигнал поступает на АЦП микроконтроллера на вывод 0 порта F. Напряжение преобразования U_{ref} в данном макете может программироваться как внутреннее, равное 2.56В, так и внешнее 2.5В. Поэтому входное измеряемое напряжение не должно превышать это значение. Входной двуполярный сигнал с уровнем от +1.25В до -1.25В преобразуется на входе АЦП с помощью ОУ в однополярный сигнал размахом 2.5В.

Данные, полученные с АЦП, выдаются на порт А микроконтроллера, являющегося шиной данных ЦАПа. Строб записи в ЦАП WR реализован на выводе 0 порта G. Сигнал выбора каналов ЦАП A/B реализован на выводе 0 порта C.

Опорное напряжение U_{ref_A} канала А блока ЦАП равно 2.5В, что соответствует напряжению преобразования АЦП. На выходе ЦАП канала А формируется оцифрованный сигнал с уровнем от 0В до U_{ref_A} . С помощью ОУ этот сигнал преобразуется в двуполярный сигнал, аналогичный по параметрам входному.

В качестве опорного напряжения U_{ref_B} канала В блока ЦАП выступает выходное напряжение канала А блока ЦАП, что позволяет реализовывать функцию умножающего ЦАПа.

Текст программы

```
.include "m128def.inc"  
.list  
.cseg
```

```

.org 0
.def TMP =R25
.def TMP1=R24

    rjmp RESET
.org ADCCaddr    ; Вектор прерывания АЦП
    rjmp ADC_in

RESET:
//Инициализация стека
//Устанавливаем стек в конец памяти RAM
    ldi    tmp,high(RAMEND)
    out    SPH,TMP
    ldi    TMP,low(RAMEND)
    out    SPL,TMP

//Настройка направления работы портов ввода-вывода
    ldi    tmp,$FF
    out    DDRA,TMP    ; Порт А (данные ЦАП) - выходы

    ldi    tmp,$01    ; Порт G (WR-PG0) - выход
    sts    DDRG,TMP
    sts    PORTG,TMP    ; Выключаем сигнал записи

    ldi    TMP,$01    ; Порт С (выбор канала ЦАП 0-А/1-В - PC0) - вы-
ход
    out    DDRC,TMP

//Настройка параметров АЦП
/* Регистр ADMUX

```

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
0	0	1	0	0	0	0	0

REFS1 REFS0 = 00 Опорное-AREF

ADLAR = 1 Выравнивание результата влево

MUX3-MUX0 = 0000 Вход ADC0

Результат преобразования - старшие 8 бит из АЦП содержатся в ADCH

*/

```
ldi   TMP,0b00100000
```

```
out   ADMUX,TMP
```

/* Регистр ADCSRA

ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
1	0	0	0	1	0	0	0

ADEN = 1 Разрешение работы АЦП

ADSC = 0 Запуск преобразования АЦП (осуществляется в основной программе)

ADFR = 0 Режим автоматического перезапуска / режим одиночного преобразования АЦП (1 / 0)

ADIF = 0 Флаг прерывания АЦП (устанавливается после завершения преобразования)

ADIE = 1 Разрешение прерывания от АЦП

ADPS2-ADPS0 = 000 Предделитель АЦП (2,2,4,8,16,32,64,128)

Время преобразования 000-5мкс 111-200мкс

*/

```
ldi   TMP,$88
```

```
out   ADCSRA,TMP
```

// Основная программа

```
sei
```

```

sbi   ADCSRA,ADSC           ; Запуск АЦП
// Первое одиночное преобразование выполняется 25 мксек
nop  nop  nop  nop  nop
m1:
// Ожидание прерывания от АЦП
    rjmp  m1

// Обработка прерывания АЦП
ADC_in:
    in    TMP1,ADCH         ; Сохраняем значение из АЦП в TMP1

    ldi   TMP,$00          ; Включение канала А
    out   PORTC,TMP
    out   PORTA,TMP1       ; Данные на вход ЦАП из TMP1
    ldi   tmp1,$00         ; Включение записи в ЦАП
    sts   PORTG,TMP1
    ldi   TMP1,$01         ; Отключение записи в ЦАП
    sts   PORTG,TMP1

sbi   ADCSRA,ADSC         ; Запуск преобразования АЦП

reti                               ; Выход из прерывания

```

6.5 Тестовая программа формирования сигнала с ШИМ

Тестовая программа формирует цифровой сигнал с широтно-импульсной модуляцией. Сигнал формируется на разъеме макета Т1ВЫХ (на схеме Т1ОВ). Описание работы модуля таймеров в микроконтроллере Atmega128 представлено на сайте <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/10.htm>.

Режим работы таймера определяет алгоритм счета и поведение выхода формирователя импульсов. Эти параметры определяются комбинацией бит, задающих режим работы таймера (WGMn3-0) и режим формирования выходного сигнала (COMnx1:0).

Текст программы:

```
include "m128def.inc"
.def tmp = R25
.org 0
rjmp start

.org OC1Caddr
    rjmp PWM_START

start:
    ldi    tmp,0b00100000 ; Бит номер 5 порта В на выход (Т1ВЫХ)
    out   DDRB,tmp

// Регистр разрешения прерываний TIMSK
//Разрешение прерывания при переполнении таймера-счетчика 1 – ТОIE1
    ldi    tmp, 0b00000100
    out   TIMSK,tmp

// начальное значение длительности = 0
    clr   tmp
    out   OCR1AL, tmp
```

```

clr    tmp
out    OCR1AH, tmp

```

```
/*Регистр TCCR1A
```

8-разр. быстрая ШИМ до 0x00FF, обновления и установка флага на вершине счета

COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10
1	0	0	0	0	0	0	1

COM1A1 COM1A0 = 1 0 Сброс ОСнА при совпадении, установка ОСнА
вершине счета

WGM11 WGM10 = 1 0 Режим 5

```
*/
```

```

ldi    tmp, 0b10000001
out    TCCR1A, tmp

```

```
/* Регистр TCCR1B
```

ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
0	0	0	0	1	0	0	1

ICNC1 ICSE1 режим захвата = не используются

WGM13 = 0, WGM12 = 1 Режим 5

CS12-CS10 = 001 Выбор тактового источника (останов,1,8,64,256,1024,
внешний)

```
*/
```

```

ldi    tmp, 0b00001001
out    TCCR1B, tmp

```

```
sei
```

```
main_loop:
```

```
    rjmp main_loop
```

```
// прерывание таймера 1 по сравнению OC1Caddr
```

```
PWM_START:
```

```
// чтение регистра сравнения
    in    R16, OCR1AL
    in    R17, OCR1AH

// изменение длительности импульса
    inc   R16

// запись младшего байта регистра сравнения
    out   OCR1AH, R17
    out   OCR1AL, R16
    reti
```

6.6 Тестовая программа формирования цифрового сигнала

Тестовая программа реализует формирование цифрового сигнала с заданными параметрами. Сигнал формируется на разъеме макета Т1ВЫХ (на схеме Т1ОВ). Описание работы модуля таймеров в микроконтроллере Atmega128 представлено на сайте <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arih128/10.htm>.

При формировании задаются длительности положительной и отрицательной полуволны сигнала. В программе применяется режим сброса таймера 1 при совпадении (СТС) СОМРА и СОМР. Длительности положительной и отрицательной полуволны сигнала записываются в регистры сравнения таймера. При достижении счетчиками таймера 1 заданных значений происходит прерывание и счет начинается снова.

Текст программы:

```
.include "m128def.inc"
.def tmp = r16

//входы и выходы порта D
.equ kn0 = 0      //кнопка 0
.equ kn1 = 1      //кнопка 1
.equ inb = 5//вход
.equ t1b = 6//вход таймера 1
.equ outb = 7     //выход

//коэффициент деления частоты
.equ PRC = 8

//частота таймера (МГц)
.equ CPU_FIO = 8

//параметры формируемого сигнала (мксек)
```

```

.equ T_H = 100
.equ T_L = 200
.equ Time_L = (T_L * CPU_FIO)/PRC
.equ Time_H = (T_H * CPU_FIO)/PRC

.cseg
.org 0
    rjmp RESET

.org OC1Aaddr      ; Прерывание таймера при совпадении COMPA
    rjmp TMR1_COMPA

.org OC1Baddr      ; Прерывание таймера при совпадении COMPB
    rjmp TMR1_COMPB

RESET:
// установка стека в конец памяти RAM
    ldi  tmp,high(RAMEND)
    out  SPH,tmp
    ldi  tmp,low (RAMEND)
    out  SPL,tmp

// установка направления работы порта D
// 0,1,5,6 - вход, 7 - выход
    ldi  tmp,0b10000000
    out  DDRD,tmp

//Загрузка регистра сравнения COMPA
    ldi  tmp,high(Time_H+Time_L)
    out  OCR1AH,tmp
    ldi  tmp,low(Time_H+Time_L)

```

```
out OCR1AL,tmp
```

```
//Загрузка регистра сравнения COMPB
```

```
ldi tmp,high(Time_H)
```

```
out OCR1BH,tmp
```

```
ldi tmp,low(Time_H)
```

```
out OCR1BL,tmp
```

```
/*Регистр TCCR1A
```

8-разр. быстрая ШИМ до 0x00FF, обновления и установка флага на вершине счета

COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10
0	0	0	0	0	0	0	0

COM1A1 COM1A0 = 0 0 Нормальная работа порта, сигналы ОСп отключены

WGM11 WGM10 = 0 0 Режим 4 (СТС) (WGM13 = 0, WGM12 = 1 в TCCR1B)

```
*/
```

```
ldi tmp,0
```

```
out TCCR1A,tmp
```

```
/* Регистр TCCR1B
```

ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
0	0	0	0	1	0	1	0

ICNC1 ICSE1 режим захвата = не используются

WGM13 = 0, WGM12 = 1 Режим 4 (СТС) (WGM11 WGM10 = 0 0 1 в TCCR1A)

CS12-CS10 = 010 Выбор тактового источника (останов,1,8,64,256,1024, внешний)

PRC = 8

```
*/
```

```
ldi tmp,0b00001010
```

```
out TCCR1B,tmp
```

```
//Занесение начального значения таймера = 0
    clr    tmp
    out    TCNT1H, tmp
    out    TCNT1L, tmp

//Разрешение прерывания по сравнению
    ldi    tmp, (1<<OCIE1B) | (1<<OCIE1A)
    out    TIMSK, tmp

// Разрешение прерываний
    sei

//Основной цикл
MainLoop:
    rjmp  MainLoop

//обработка прерывания таймера по сравнению COMPA
TMR1_COMPA:
//выключение сигнала
    sbi    PORTD,outb
    reti

//обработка прерывания таймера по сравнению COMPB
TMR1_COMPB:
//включение сигнала
    cbi    PORTD,outb
    reti
```

6.7 Тестовая программа измерения параметров цифрового сигнала

Тестовая программа измеряет длительность цифрового сигнала. Цифровой сигнал поступает от внешнего генератора на разъем макета Т1ВХ (на схеме Т11В). Описание работы модуля таймеров в микроконтроллере Atmega128 представлено на сайте <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/10.htm>.

Режим работы таймера определяет алгоритм счета и поведение выхода формирователя импульсов. Эти параметры определяются комбинацией бит, задающих режим работы таймера (WGMn3-0) и режим формирования выходного сигнала (COMnx1:0).

Текст программы:

```
include "m128def.inc"
.def tmp1 = R25
.def tmp2 = R24
.org 0
rjmp start

// описание входов и выходов порта D
.equ kn0 = 0//кнопка 0
.equ kn1 = 1//кнопка 1
.equ inb = 5//вход IN «ВХ1»
.equ tlib = 6 //вход таймера Т11В «Т1ВХ»
.equ outb= 7//выход OUT «ВЫХ1»

start:
// установка направления работы порта D
// 0,1,5,6 - вход, 7 - выход
    ldi    tmp1,0b10000000
    out   DDRD,tmp1
```

/*Регистр TCCR1A

Нормальный 16-разрядный режим счета

COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10
0	0	0	0	0	0	0	0

COM1A1 COM1A0 = 0 0 Нормальная работа порта

WGM11 WGM10 = 0 0 Режим 0

*/

```
ldi tmp1, 0
```

```
out TCCR1A, tmp1
```

/* Регистр TCCR1B

ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
0	0	0	0	0	0	1	0

ICNC1 ICSE1 режим захвата = не используются

WGM13 = 0, WGM12 = 0 Режим 0

CS12-CS10 = 010 Выбор тактового источника (останов,1,8,64,256,1024, внешний)

*/

```
ldi tmp1, 0b00000010
```

```
ldi tmp2, 0
```

m0:

```
sbic PIND,inb ; Входной сигнал равен 0
```

```
jmp m0 ; Ожидание окончания сигнала
```

m1:

```
sbis PIND,inb ; Входной сигнал равен 1
```

```
jmp m1 ; Ожидание начала сигнала
```

// ДА – запуск таймера

```
out TCCR1B, tmp1
```

m2:

```
sbic PIND,inb ; Входной сигнал равен 0
```

```
    jmp m2          ; Ожидание окончания сигнала
// ДА – выключение таймера
    out    TCCR1B, tmp2
m3:
// конец программы
    jmp m3
```

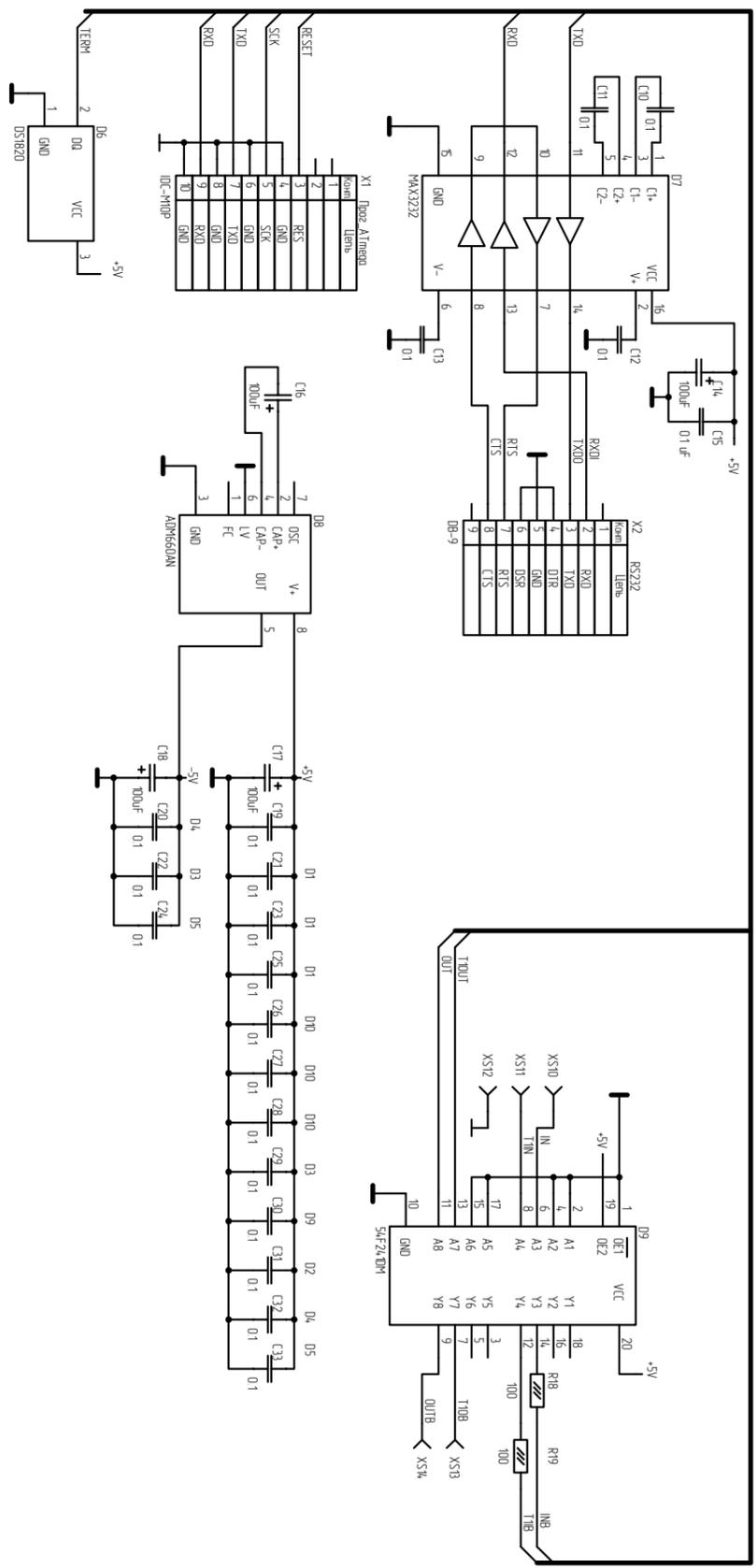
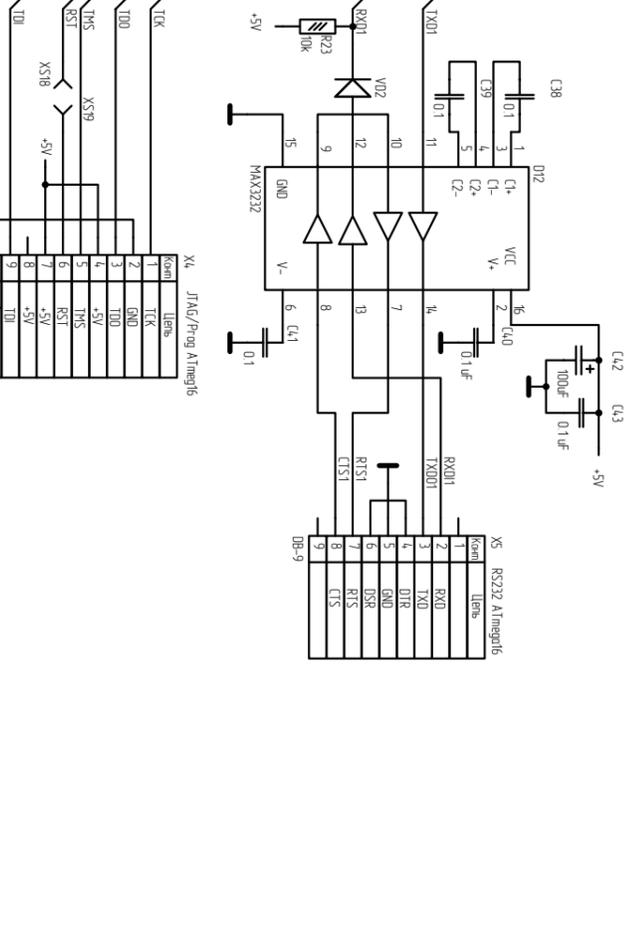
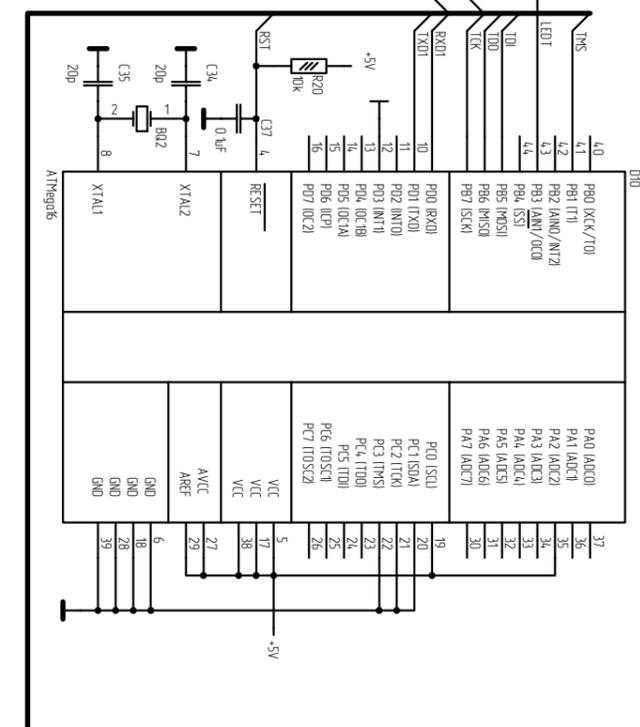
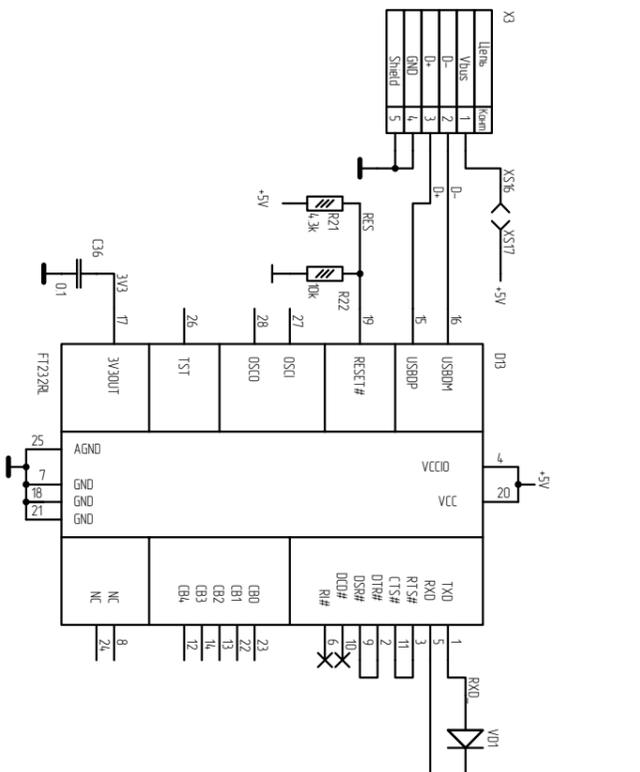
СПИСОК ЛИТЕРАТУРЫ

- 1 Микроконтроллеры семейства AVR [Электронный ресурс]. – Режим доступа: <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/index.htm>
- 2 ATmega128, ATmega128L. 8-разрядный AVR-микроконтроллер с внутренне программируемой флэш-памятью емкостью 128 кбайт [Электронный ресурс]. – Режим доступа: <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/index.htm>
- 3 Голубцов М. С. Микроконтроллеры AVR: от простого к сложному / М. С. Голубцов. – М. : СОЛОН-Пресс, 2003.
- 4 Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel / В. В. Гребнев. – М. : РадиоСофт, 2002.
- 5 Гурьянов С. А. AVR микроконтроллеры фирмы Atmel : курс лекций / С. А. Гурьянов : НовГУ им. Ярослава Мудрого. – Великий Новгород, 2005.
- 6 Евстифеев А. В. Микроконтроллеры AVR семейства Classic фирмы ATMEL / А. В. Евстифеев. – М. : Додэка-XXI, 2004.

Приложение А

Схема электрическая принципиальная лабораторная макета

Инв.№ подл.	Подп. и дата	Взаим.инв. №	Инв. № дубл.	Подп. и дата



Изм.	Лист	№ докум.	Подп.	Дата	Конструктор	Формат	Лист
							2