

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет»
Факультет «Автотракторный»
Кафедра «Двигатели внутреннего сгорания и электронные системы автомобилей»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
_____ /В.Е. Лазарев/
_____ 20__ г.

Особенности программного обеспечения информационных управляющих
систем наземных транспортных средств

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ
РАБОТЕ

ЮУрГУ – 13.03.02.2017.040.00 ПЗ ВКР

Руководитель работы
_____ /А.С. Мартьянов/
_____ 20__ г.

Автор работы
студент группы П-410
_____ /Э.Р. Ишимов/
_____ 20__ г.

Нормоконтролер
_____ /Астафьев Д.В.
_____ 20__ г.

АННОТАЦИЯ

Э.Р.Ишимов. Особенности программного обеспечения информационных управляющих систем наземных транспортных средств – Челябинск: ЮУрГУ, АТ, П-410;2017, 68с.

В выпускной квалификационной работе представлена разработка микропроцессорной системы управления блоком управления отопителем, вентилятором и кондиционером автомобиля. Для достижения поставленной цели выполнен обзор литературы, рассмотрены различные системы управления, выполнено сравнение различных типов микроконтроллеров и микропроцессоров. Сформулированы задачи, определены требования к функциям, разработан алгоритм управления. Разработана модель системы управления климатической установкой автомобиля. На основе предложенного алгоритма создано программное обеспечение, реализующее необходимые функции управления. Проведены исследовательские эксперименты по проверке правильности функционирования, подтверждающие корректность предложенного решения.

					13.03.02.2017.038.00 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата						
Разраб.	Ишимов				Особенности программного обеспечения информационных управляющих систем наземных транспортных средств	Лит.	Лист	Листов		
Провер.	Мартьянов								5	68
Реценз.										
Н. Контр.	Астафьев					ЮУрГУ				
Утверд.	Лазарев									

Оглавление

1 ОБЗОР ЛИТЕРАТУРЫ. СРАВНЕНИЕ ТЕХНОЛОГИЙ	8
1.1 Чем микроконтроллер отличается от микропроцессора	8
1.2 Что такое микропроцессор.....	8
1.3Что такое микроконтроллер	11
1.4Сравниваем микроконтроллер и микропроцессор	13
2 РАЗРАБОТКА СИСТЕМЫ	14
2.1 Описание объекта управления.....	14
2.2 Требования к функциям	19
2.3 Алгоритм реализации функции	28
3 ИССЛЕДОВАТЕЛЬСКО – ЭКСПЕРИМЕНТАЛЬНЫЙ РАЗДЕЛ.....	29
3.1 Семейство микроконтроллеров.....	29
3.2 Основные достоинства и описание платформы Arduino	33
3.3 Язык Ардуино	35
3.4 Описание среды разработки	41
3.5 Разработка программы	46
3.6 Тестирование программы.	50
ЗАКЛЮЧЕНИЕ	60
СПИСОК ЛИТЕРАТУРЫ.....	61
ПРИЛОЖЕНИЕ. Исходный текст программы.....	62

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		6

ВВЕДЕНИЕ

Современные автомобили оснащаются значительным количеством электрических устройств, которые требуют качественного технического обслуживания и грамотной эксплуатации.

Цель настоящей работы состоит в разработке микропроцессорной системы управления блоком управления печкой, вентилятором и кондиционером легкового автомобиля.

Для достижения поставленной в работе цели необходимо решить следующие задачи:

1. Изучить научную и техническую литературу автомобильной и микропроцессорной сферы, касающуюся темы работы.
2. Сформулировать технические требования к разрабатываемой системе и определить необходимые инструментальные средства разработки.
3. Разработать и реализовать алгоритм управления блоком управления печкой, вентилятором и кондиционером автомобиля с помощью микропроцессорной системы управления
4. Провести экспериментальные исследования разработанной системы и сделать выводы.

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		7

ОБЗОР ЛИТЕРАТУРЫ. СРАВНЕНИЕ ТЕХНОЛОГИЙ

1.1 Чем микроконтроллер отличается от микропроцессора

Давайте разберемся, чем же на самом деле отличаются и в чем сходство этих двух типов цифровых радиоэлектронных устройств.

И микропроцессор и микроконтроллер предназначены для выполнения некоторых операций — они извлекают команды из памяти и выполняют эти инструкции (представляющие собой арифметические или логические операции) и результат используется для обслуживания выходных устройств. И микроконтроллер и микропроцессор способны непрерывно производить выборку команд из памяти и выполнять эти инструкции, пока на устройство подается питание. Инструкции представляют из себя наборы битов. Эти инструкции всегда извлекаются из места их хранения, которое называется памятью.

1.2 Что такое микропроцессор

«Микропроцессор (в англоязычной литературе MPU — MicroProcessorUnit) содержит функционал компьютерного центрального процессора, или ЦП (CPU — CentralProcessingUnit) на одном полупроводниковом кристалле (ИМС — интегральная микросхема или на западный манер — IntegratedCircuit).

По своей сути — это микрокомпьютер, который используется для выполнения арифметических и логических операций, управления системами, хранения данных и прочих.

Микропроцессор обрабатывает данные, поступающие с входных периферийных устройств и передает обработанные данные на выходные периферийные устройства.

Существует четыре основных типа процессоров, различающихся своей архитектурой»[1].

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		8

Формирование длинных «командных слов» стало возможным благодаря увеличению разрядности микропроцессорных устройств»[2].

В суперскалярных процессорах (SuperscalarProcessors) используются несколько декодеров команд, которые загружают работой множество исполнительных блоков. Планирование исполнения потока команд происходит динамически и осуществляется самим вычислительным ядром. Примером процессора с таким типом архитектуры является, например Cortex A8.

Отдельно хочу выделить микропроцессоры специального назначения (ASIC — Application Specific Integrated Circuit). Как следует из названия, предназначены для решения конкретной задачи. В отличие от микропроцессоров общего назначения, применяются в конкретном устройстве и выполняют определенные функции, характерные только для данного устройства. Специализация на выполнении узкого класса функций приводит к увеличению скорости работы устройства и, как правило, позволяет снизить стоимость такой интегральной схемы. Примерами таких микропроцессоров может быть микросхема, разработанная исключительно для управления мобильным телефоном, микросхемы аппаратного кодирования и декодирования аудио- и видеосигналов - так называемые цифровые сигнальные процессоры (DigitalSignalProcessing, DSP multiprocessors). Могут быть реализованы в виде ПЛИС (программируемая логическая интегральная схема). При разработке таких процессоров для описания их функциональности используют языки описания аппаратных устройств (HDL — Hardware Description Language), такие как Verilog и VHDL.

Системы на основе микропроцессоров строят примерно следующим образом.

									Лист
									10
Изм.	Лист	Недокум.	Подп.	Дата					



Рисунок 2 – Система основанная на микропроцессоре

Как видно, микропроцессор в этой системе имеет множество вспомогательных устройств, таких как постоянное запоминающее устройство, оперативная память, последовательный интерфейс, таймер, порты ввода/вывода и т.д. Все эти устройства обмениваются командами и данными с микропроцессором через системную шину. Все вспомогательные устройства в микропроцессорной системе являются внешними. Системная шина, в свою очередь, состоит из адресной шины, шины данных и шины управления.

Теперь, давайте, рассмотрим микроконтроллер.

1.3 Что такое микроконтроллер

«Ниже представлена блок-схема микроконтроллера. Какого же его основное отличие от микропроцессора? Все опорные устройства, такие как постоянное запоминающее устройство, оперативная память, таймер, последовательный интерфейс, порты ввода/вывода являются встроенными.

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		11

Программа, записанная в память микроконтроллера может быть защищена от возможности последующего чтения/записи, что обеспечивает защиту от ее несанкционированного использования.

1.4Сравниваем микроконтроллер и микропроцессор

Таблица 1 – Сравнение микропроцессора и микроконтроллера

-	Микропроцессор	Микроконтроллер
Использование	Компьютерные системы	Встраиваемые системы
Устройство	Содержит центральный процессор, регистры общего назначения, указатели стека, счетчики программы, таймер и цепи прерываний.	Содержит схему микропроцессора и имеет встроенные ПЗУ, ОЗУ, устройства ввода/вывода, таймеры и счетчики.
Память данных	Имеет много инструкций для перемещения данных между памятью и процессором.	Имеет одну-две инструкции для перемещения данных между памятью и процессором.
Электрические цепи	Высокая сложность	Достаточно простые
Затраты	Стоимость всей системы увеличивается.	Низкая стоимость системы
Число регистров	Имеет меньшее количество регистров, операции в основном производятся в памяти.	Имеет большее число регистров, поэтому проще писать программы
Запоминающее устройство	Основано на архитектуре фон Неймана. Программа и данные хранятся в том же модуле памяти.	Основано на Гарвардской архитектуре. Программы и данные хранятся в разных модулях памяти.
Время доступа	Время доступа к памяти и устройствам ввода/вывода больше.	Меньшее время доступа для встроенной памяти и устройств ввода/вывода.
Железо	Требует большее количество аппаратного обеспечения.	Требует меньшее количество

РАЗРАБОТКА СИСТЕМЫ

2.1 Описание объекта управления

Объектом управления является блок управления печкой, вентилятором и кондиционером на примере автомобиля HyundaiSolaris.

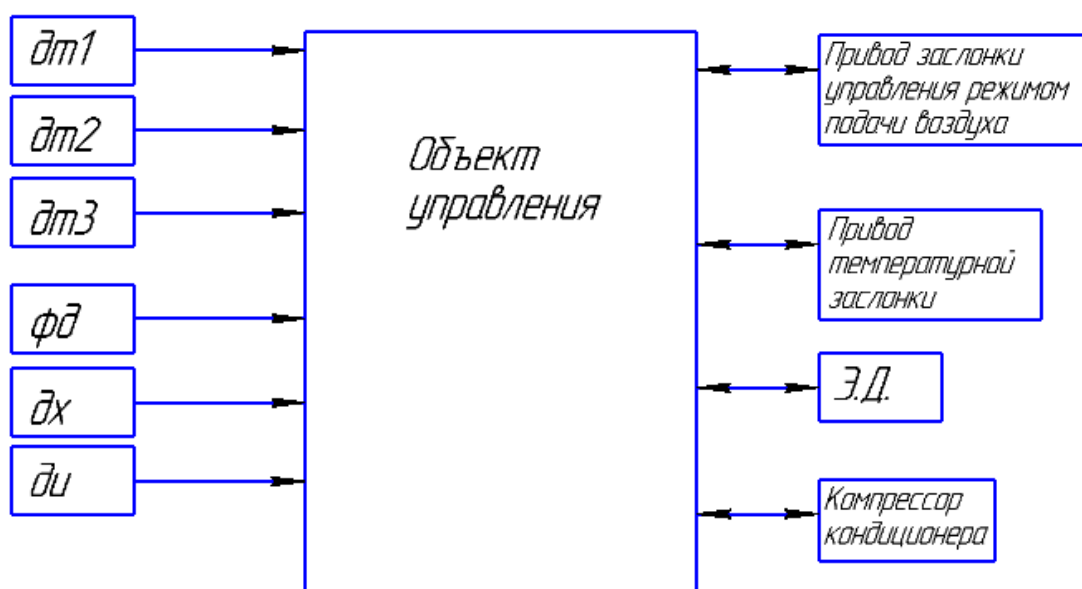


Рисунок 4 – Схема объекта управления

К входным сигналам блока управления относятся:

- датчик температуры салона dt1;
- датчик температуры окружающего воздуха dt2;
- датчик температуры охлаждающей жидкости dt3;
- фотодатчикfd;
- датчик давления хладагентаdx;
- датчик температуры испарителяди;
- обратная связь приводов.

К выходным сигналам блока управления относятся:

- управления электродвигателя вентилятора отопителя;
- управления привода температурной заслонки;
- привод заслонки управления режимом подачи воздуха;

- компрессор кондиционера.

Датчик температуры салона представляет собой термистор, который служит для измерения температуры воздуха внутри салона автомобиля. Сигнал, значение которого определяется сопротивлением датчика в соответствии с температурой внутреннего воздуха, поступает в блок управления отопителем. Согласно этому сигналу осуществляется регулирование температуры в салоне автомобиля в соответствии с заданным значением.

Таблица 2 – Зависимость сопротивления датчика температуры салона

Температура (°C/°F)	Сопротивление (Ом)
-20/-4	284
-10/-14	164,6
0/32	97.7
10/50	59.67
20/68	37.4
30/86	24.1
40/104	15.9
50/122	10,8

Датчик температуры окружающего воздуха. Датчик температуры наружного воздуха расположен перед конденсатором. Это термистор с негативным температурным коэффициентом, сопротивление которого при понижении температуры повышается (и наоборот). Выходной сигнал датчика используется при работе датчика температуры подаваемого воздуха, для управления положением заслонки регулирования температуры, частотой вращения электродвигателя вентилятора, режимом смешанной подачи воздуха и контроля влажности в салоне автомобиля.

Таблица 3 – Зависимость сопротивления датчика температуры окружающего воздуха

Температура окружающего воздуха [°C(°F)]	Сопротивление между клеммами 1 и 2 (кОм)
-30 (-22)	507
-20(-4)	284.5
-10 (14)	164.2
0 (32)	97,5
10	59,6
20 (68)	37.46
30(86)	24.18
40 (104)	16
50 (122)	10.83

Датчик температуры охлаждающей жидкости Датчик температуры охлаждающей жидкости двигателя (ЕСТ) служит для измерения температуры охлаждающей жидкости двигателя. Датчик ЕСТ находится рядом с корпусом термостата, на головке блока цилиндров. Датчик ЕСТ представляет собой термистор (резистор, напряжение которого меняется в соответствии с температурой окружающей среды), который подключен последовательно с резистором внутри блока управления двигателем (ЕСМ). От блока ЕСМ на датчик ЕСТ подается напряжение 5 В. Напряжение на выводах датчика ЕСТ, контролируемое блоком ЕСМ, преобразуется в показания температуры. При низкой температуре охлаждающей жидкости сопротивление датчика ЕСТ велико; при повышении температуры сопротивление датчика уменьшается. Поэтому при низкой температуре в блок ЕСМ поступает высокое напряжение сигнала, которое с ростом температуры падает. Сигнал от датчика ЕСТ используется при управлении инжекторами, определении момента зажигания, управлении оборотами холостого хода и управлении вентилятором.

Таблица 4 – Зависимость сопротивления датчика температуры охлаждающей жидкости

Температура (°C)	Сопротивление (кОм)
-40(-40)	48,14
-20(-4)	14,13~16,83
0 (32)	5,79
20 (68)	2,31~2,59
40 (104)	1,15
60 (140)	0,59
80 (176)	0,32

Фотодатчик. Датчик освещенности расположен по центру сопла подачи воздуха на остекление. В состав датчика освещенности HyundaiSolaris входит фотоэлектрический (чувствительный к дневному свету) элемент. Под действием солнечного излучения, попадающего на его приемную часть, в нем возникает электродвижущая сила, пропорциональная полученному количеству излучения. Она передается в блок управления температурой и используется для выполнения температурной компенсации солнечного излучения.



Рисунок 5 –Фотодатчик при попадании солнечного света под углом

при падении солнечных лучей под углом спереди, то есть прямо на водителя и переднего пассажира, им становится существенно теплее. Оптический элемент устроен таким образом, что значительная часть излучения в этом случае направляется на фотодиод. Хладопроизводительность климатической установки увеличивается, и нагревающее действие солнечного света компенсируется.



Рисунок 6 – Фотодатчик при попадании солнечного света вертикально при падении солнечных лучей вертикально оптический элемент направляет меньше света на светодиод. Хладопроизводительность уменьшается, поскольку люди в автомобиле не находятся под прямыми солнечными лучами.

Датчик давления хладагента Датчик давления в кондиционере, основанный на пьезоэлектрическом чувствительном элементе, используется для измерения давления хладагента. Пьезоэлектрический чувствительный элемент выгодно отличается от традиционного датчика релейного типа, поскольку его выходной сигнал связан с давлением линейной зависимостью. Благодаря этому удастся обеспечить оптимальное управление компрессором кондиционера и вентилятором, и, как следствие, снизить расход топлива. В функции датчика низкого давления кондиционера входит автоматическое отключение сплит-системы при недостаточном количестве в ней фреона. Иногда охлаждающая смесь может отсутствовать вовсе. Недостаток фреона грозит подсосом воздуха из-за очень низкого давления внутри. Такой процесс тоже крайне нежелателен. Датчик высокого давления, напротив, блокирует работу компрессора, если эта характеристика превышает допустимые значения. Избыточное давление может спровоцировать физическое разрушение системы. Кроме того, в функции датчика входит обязанность периодически включать и отключать вентилятор радиатора.

Датчик температуры испарителя Датчик температуры испарителя, расположенный на блоке отопителя HyundaiSolaris, служит для регистрации

						13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата			18

температуры сердечника испарителя. Это термистор с отрицательным температурным коэффициентом, сопротивление которого обратно пропорционально температуре. С помощью датчика температуры испарителя осуществляется преобразование значения температуры в электрический сигнал и передача его в ЭБУ кондиционера. Если температура сердечника опускается до определенного значения, компрессор кондиционера отключается по команде ЭБУ кондиционера, чтобы предотвратить замораживание испарителя.

Таблица 5 – Зависимость сопротивления датчика испарителя от температуры

Температура (°C/°F)	Сопротивление (Ом)
-10/-14	18
0,32	11
10/50	7,3
20/68	4,8
30/86	3,3
40/104	2~3

2.2 Требования к функциям

Таблица 6 – Внешние условия

-	$t_{\text{ОКР.ВОЗ}}$	$t_{\text{ОЖ}}$	$t_{\text{САЛ.ВОЗ}}$	ФД	Выбранный режим
1	От -40 до -20	<60	< зад	Да	Вниз
2	От -20 до 0	≥ 60	>зад	Нет	Вниз/Вперед
3	От 0 до 20	-	-	-	Вперед/Вверх
4	От 20 до 40	-	-	-	Вверх
-	4	2	2	2	4
Итого:	128				

Таблица 7 – Таблица состояний

Номер состояния	$t_{\text{ОКР.ВОЗ}}$ (От -40 до -20 От -20 до 0 От 0 до 20 От 20 до 40)	$t_{\text{ОЖ}}$ (<60 ≥ 60)	$t_{\text{САЛ.ВОЗ}}$ (< зад > зад)	ФД (Да/ Нет)	Выбранный режим Вниз Вниз/Вперед Вперед/Вверх Вверх
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	0	3
4	0	0	0	1	0
5	0	0	0	1	1
6	0	0	0	1	2
7	0	0	0	1	3
8	0	0	1	0	0
9	0	0	1	0	1
10	0	0	1	0	2
11	0	0	1	0	3
12	0	0	1	1	0
13	0	0	1	1	1
14	0	0	1	1	2
15	0	0	1	1	3
16	0	1	0	0	0
17	0	1	0	0	1
18	0	1	0	0	2
19	0	1	0	0	3
20	0	1	0	1	0
21	0	1	0	1	1
22	0	1	0	1	2
23	0	1	0	1	3
24	0	1	1	0	0
25	0	1	1	0	1
26	0	1	1	0	2
27	0	1	1	0	3
28	0	1	1	1	0
29	0	1	1	1	1
30	0	1	1	1	2
31	0	1	1	1	3
32	1	0	0	0	0

Продолжение таблицы 7

Номер состояния	t _{ОКР.ВОЗ} (От -40 до -20 От -20 до 0 От 0 до 20 От 20 до 40)	t _{ОЖ} (<60 ≥60)	t _{САЛ.ВОЗ} (< зад > зад)	ФД (Да/ Нет)	Выбранный режим Вниз Вниз/Вперед Вперед/Вверх Вверх
33	1	0	0	0	1
34	1	0	0	0	2
35	1	0	0	0	3
36	1	0	0	1	0
37	1	0	0	1	1
38	1	0	0	1	2
39	1	0	0	1	3
40	1	0	1	0	0
41	1	0	1	0	1
42	1	0	1	0	2
43	1	0	1	0	3
44	1	0	1	1	0
45	1	0	1	1	1
46	1	0	1	1	2
47	1	0	1	1	3
48	1	1	0	0	0
49	1	1	0	0	1
50	1	1	0	0	2
51	1	1	0	0	3
52	1	1	0	1	0
53	1	1	0	1	1
54	1	1	0	1	2
55	1	1	0	1	3
56	1	1	1	0	0
57	1	1	1	0	1
58	1	1	1	0	2
59	1	1	1	0	3
60	1	1	1	1	0
61	1	1	1	1	1
62	1	1	1	1	2
63	1	1	1	1	3
64	2	0	0	0	0
65	2	0	0	0	1
66	2	0	0	0	2

Изм.	Лист	Недокум.	Подп.	Дата
------	------	----------	-------	------

13.03.02.2017.038.00 ПЗ

Лист

21

Продолжение таблицы 7

Номер состояния	t _{ОКР.ВОЗ} (От -40 до -20 От -20 до 0 От 0 до 20 От 20 до 40)	t _{ОЖ} (<60 ≥60)	t _{САЛ.ВОЗ} (< зад > зад)	ФД (Да/ Нет)	Выбранный режим Вниз Вниз/Вперед Вперед/Вверх Вверх
67	2	0	0	0	3
68	2	0	0	1	0
69	2	0	0	1	1
70	2	0	0	1	2
71	2	0	0	1	3
72	2	0	1	0	0
73	2	0	1	0	1
74	2	0	1	0	2
75	2	0	1	0	3
76	2	0	1	1	0
77	2	0	1	1	1
78	2	0	1	1	2
79	2	0	1	1	3
80	2	1	0	0	0
81	2	1	0	0	1
82	2	1	0	0	2
83	2	1	0	0	3
84	2	1	0	1	0
85	2	1	0	1	1
86	2	1	0	1	2
87	2	1	0	1	3
88	2	1	1	0	0
89	2	1	1	0	1
90	2	1	1	0	2
91	2	1	1	0	3
92	2	1	1	1	0
93	2	1	1	1	1
94	2	1	1	1	2
95	2	1	1	1	3
96	3	0	0	0	0
97	3	0	0	0	1
98	3	0	0	0	2
99	3	0	0	0	3
100	3	0	0	1	0

Окончание таблицы 7

101	3	0	0	1	1
102	3	0	0	1	2
103	3	0	0	1	3
104	3	0	1	0	0
105	3	0	1	0	1
106	3	0	1	0	2
107	3	0	1	0	3
108	3	0	1	1	0
109	3	0	1	1	1
110	3	0	1	1	2
111	3	0	1	1	3
112	3	1	0	0	0
113	3	1	0	0	1
114	3	1	0	0	2
115	3	1	0	0	3
116	3	1	0	1	0
117	3	1	0	1	1
118	3	1	0	1	2
119	3	1	0	1	3
120	3	1	1	0	0
121	3	1	1	0	1
122	3	1	1	0	2
123	3	1	1	0	3
124	3	1	1	1	0
125	3	1	1	1	1
126	3	1	1	1	2
127	3	1	1	1	3

Таблица 8 –Управляющие воздействия

-	Привод 1(от 0 до 5, где 0 – закр., 5 – откp.)	Привод 2(от 0 до 5, где 0 – холод., 5 – тепло.)	Вентилятор (от 0 до 4, где 0 – мин., 4 – макс.)
1	0	5	0
2	1	5	0
3	2	5	0
4	3	5	0
5	0	5	0
6	1	5	0
7	2	5	0
8	3	5	0

Продолжение таблицы 8

-	Привод 1 (от 0 до 5, где 0 – закр., 5 – откр.)	Привод 2(от 0 до 5, где 0 – холод., 5 – тепло.)	Вентилятор (от 0 до 4, где 0 – мин., 4 – макс.)
9	0	5	0
10	1	5	0
11	2	5	0
12	3	5	0
13	0	5	0
14	1	5	0
15	2	5	0
16	3	5	0
17	0	5	2
18	1	5	2
19	2	5	2
20	3	5	2
21	0	5	2
22	1	5	2
23	2	5	2
24	3	5	2
25	0	4	1
26	1	4	1
27	2	4	1
28	3	4	1
29	0	3	1
30	1	3	1
31	2	3	1
32	3	3	1
33	0	5	0
34	1	5	0
35	2	5	0
36	3	5	0
37	0	5	0
38	1	5	0
39	2	5	0
40	3	5	0
41	0	5	1
42	1	5	1
43	2	5	1
44	3	5	1
45	0	5	1
46	1	5	1

Изм.	Лист	Недокум.	Подп.	Дата

13.03.02.2017.038.00 ПЗ

Лист

24

Продолжение таблицы 8

-	Привод 1 (от 0 до 5, где 0 – закр., 5 – откр.)	Привод 2(от 0 до 5, где 0 – холод., 5 – тепло.)	Вентилятор (от 0 до 4, где 0 – мин., 4 – макс.)
47	2	5	1
48	3	5	1
49	0	5	2
50	1	5	2
51	2	5	2
52	3	5	2
53	0	5	2
54	1	5	2
55	2	5	2
56	3	5	2
57	0	4	1
58	1	4	1
59	2	4	1
60	3	4	1
61	0	4	1
62	1	4	1
63	2	4	1
64	3	4	1
65	0	5	1
66	1	5	1
67	2	5	1
68	3	5	1
69	0	5	1
70	1	5	1
71	2	5	1
72	3	5	1
73	0	4	1
74	1	4	1
75	2	4	1
76	3	4	1
77	0	4	1
78	1	4	1
79	2	4	1
80	3	4	1
81	0	5	2
82	1	5	2
83	2	5	2
84	3	5	2

Изм.	Лист	Недокум.	Подп.	Дата

13.03.02.2017.038.00 ПЗ

Лист

25

Продолжение таблицы 8

-	Привод 1 (от 0 до 5, где 0 – закр., 5 – откр.)	Привод 2(от 0 до 5, где 0 – холод., 5 – тепло.)	Вентилятор (от 0 до 4, где 0 – мин., 4 – макс.)
85	0	4	2
86	1	4	2
87	2	4	2
88	3	4	2
89	0	3	1
90	1	3	1
91	2	3	1
92	3	3	1
93	0	2	1
94	1	2	1
95	2	2	1
96	3	2	1
97	0	4	1
98	1	4	1
99	2	4	1
100	3	4	1
101	0	3	1
102	1	3	1
103	2	3	1
104	3	3	1
105	0	2	1
106	1	2	1
107	2	2	1
108	3	2	1
109	0	1	1
110	1	1	1
111	2	1	1
112	3	1	1
113	0	1	1
114	1	1	1
115	2	1	1
116	3	1	1
117	0	0	1
118	1	0	1
119	2	0	1
120	3	0	1
121	0	0	1
122	1	0	1

Изм.	Лист	Недокум.	Подп.	Дата

13.03.02.2017.038.00 ПЗ

Лист

26

Окончание таблицы 8

-	Привод 1 (от 0 до 5, где 0 – закр., 5 – откр.)	Привод 2(от 0 до 5, где 0 – холод., 5 – тепло.)	Вентилятор (от 0 до 4, где 0 – мин., 4 – макс.)
123	2	0	1
124	3	0	1
125	0	0	2
126	1	0	2
127	2	0	2
128	3	0	2

2.3 Алгоритм реализации функции

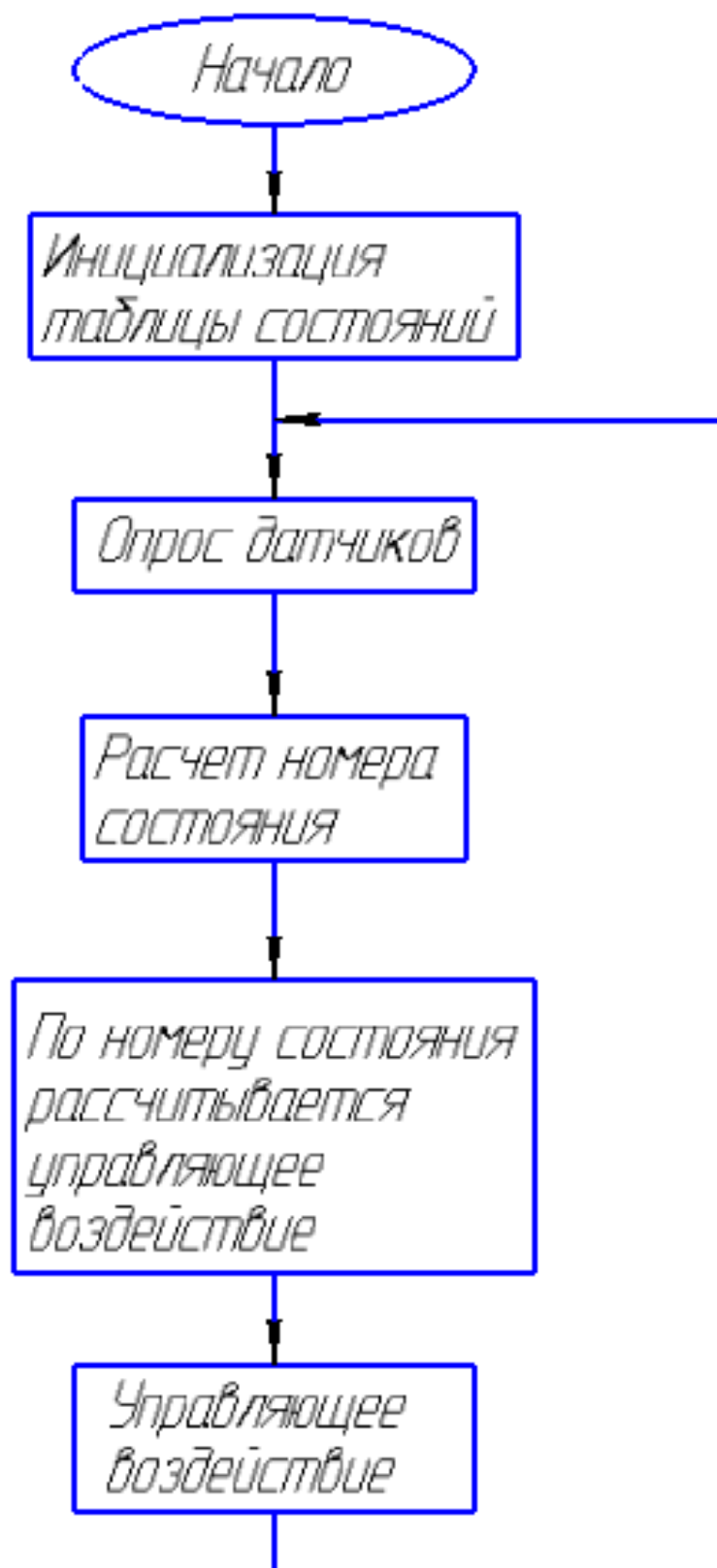


Рисунок 7 – Алгоритм реализации

ИССЛЕДОВАТЕЛЬСКО – ЭКСПЕРИМЕНТАЛЬНЫЙ РАЗДЕЛ

3.1 Семейство микроконтроллеров

В настоящее время существует множество микроконтроллеров и платформ для осуществления управления физическими процессами применительно к микропроцессорным комплексам.

На сегодняшний день наиболее известные семейства микроконтроллеров:

- MCS 51 (Intel)
- MSP430 (TI)
- ARM (ARM Limited)
- AVR (Atmel) (ATmega, ATtiny, XMega)
- PIC (Microchip).

AVR – семейство восьмибитных микроконтроллеров фирмы Atmel, впервые выпущенные в 1996 г. Они представляют собой мощный инструмент, универсальную основу для создания современных экономичных встраиваемых систем многоцелевого назначения. Идея разработки нового RISC-ядра принадлежит двум студентам Норвежского университета наук и технологий (г. Тронхейм) – Альфу Богену (Alf-EgilBogen) и ВегардуВоллену (VegardWollen). В 1995 году Боген и Воллен решили предложить американской корпорации Atmel выпускать новый 8-битный RISC-микроконтроллер и снабдить его Flash-памятью для программ на одном кристалле с вычислительным ядром. Идея была одобрена AtmelCorporation, и в конце 1996 года был выпущен опытный микроконтроллер AT90S1200, а во второй половине 1997 года корпорация Atmel приступила к серийному производству нового семейства микроконтроллеров. Новое ядро было запатентовано и получило название AVR. Существует несколько трактовок данной аббревиатуры. Кто-то утверждает, что это AdvancedVirtual RISC, другие полагают, что не обошлось здесь без инициалов разработчиков AlfEgilBogenVegardWollan RISC. Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		29

систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединенных в регистровый файл. В отличие от «идеального» RISC, регистры не абсолютно равноправны: – три «сдвоенных» 16-битных регистра-указателя X (r26:r27), Y (r28:r29) и Z (r30:r31); – некоторые команды работают только с регистрами r16...r31; – результат умножения (в тех моделях, в которых есть модуль умножения) всегда помещается в r0:r1.

На сегодняшний день доступны 3 линейки микроконтроллеров:

TinyAVR – имеет небольшой объем памяти, небольшие размеры, подходит для самых простых задач.

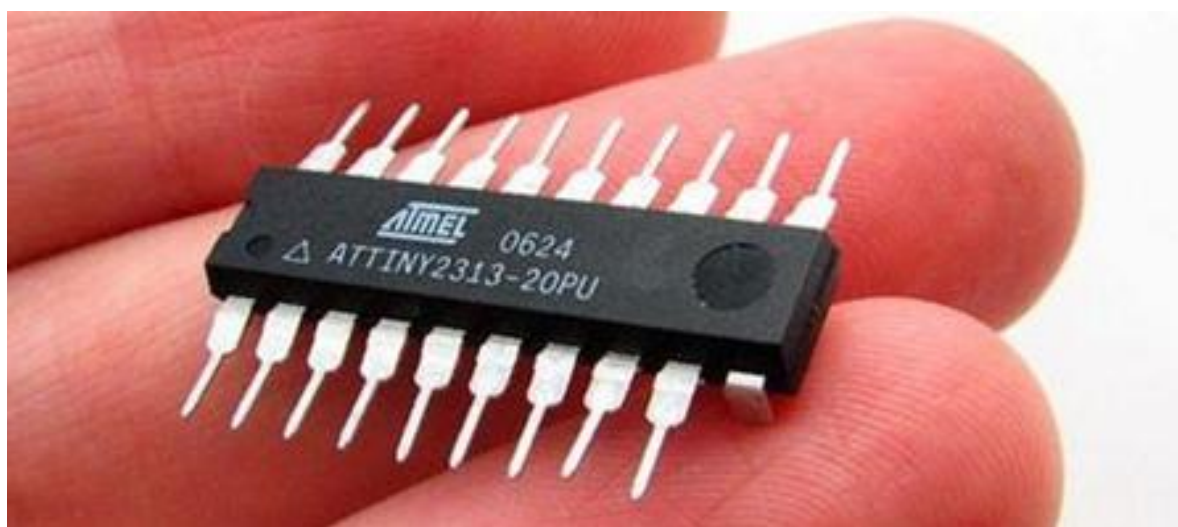


Рисунок 8 – Внешний вид микроконтроллера TinyAVR

MegaAVR – это наиболее распространенная линейка, имеющая большой объем встроенной памяти (до 256 КБ), множество дополнительных устройств и предназначенная для задач средней и высокой сложности.

платформа Arduino может стать основным элементом для исследования и решения задач в областях мехатроники и робототехники.

3.2 Основные достоинства и описание платформы Arduino

Arduino - это инструмент для проектирования электронных устройств, более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Это платформа, предназначенная для управления физическими процессами с использованием ЭВМ с открытым программным кодом, построенная на простой печатной плате с современной средой для написания программного обеспечения. Платы Arduino строятся на основе микроконтроллеров фирмы Atmel, а также элементов обвязки для программирования и интеграции с другими схемами (табл. 1). На платах присутствует линейный стабилизатор напряжения +5 В или +3,3 В. Тактирование осуществляется на частотах 8, 16 или 87 МГц кварцевым резонатором. В микроконтроллер предварительно прошивается загрузчик, поэтому внешний программатор не нужен. На концептуальном уровне все платы программируются через RS-232. Интегрированная среда разработки Arduino — это кроссплатформенное приложение на Java, включающее в себя редактор кода, компилятор и модуль передачи прошивки в плату. Среда разработки основана на языке программирования Processing и спроектирована для программирования новичками, не знакомыми близко с разработкой программного обеспечения. Строго говоря, это-язык C++, дополненный некоторыми библиотеками.

									Лист
									33
Изм.	Лист	Недокум.	Подп.	Дата					

Таблица 10 – Сравнительная характеристика плат Arduino

Arduino	Процессор Atmel		Флеш-память	SRAM	Двочные	ШИМ	Аналоговые	Размеры
	Вид	Тактовая частота, МГц	КБ	КБ	ВХОДЫ/ВЫХОДЫ	ВЫХОДЫ	ВХОДЫ	мм
Nano	ATmega168 or ATmega328	8	16/32	1	14	6	8	43 x 18
Diecimila	ATmega168	16	16	1	14	6	6	68,6 x 53.3
Uno	ATmega328P	16	32	2	14	6	6	68,6 x 53.3
Leonardo	Atmega32u4	16	32	2	14	6	12	68,6 x 53.3
Mega2560	ATmega2560	16	256	8	54	14	16	101,6 x 53.3
Due	SAM3X8E ARM Cortex-M3	87	512	256	54	12	12 + 2ЦАП	101,6 X 53,3

Программы обрабатываются с помощью препроцессора, а затем компилируется с помощью AVR-GCC.

Преимуществами плат семейства Arduino являются:

1. Большое количество доступных вариантов в линейке Arduino с возможностью выбора наиболее подходящего готового контроллера из большого списка устройств, имеющих в широких пределах варьируемые параметры.
2. Наличие плат расширения, предназначенных для увеличения функционала и выполнения конкретизированных технических задач без необходимости самостоятельного проектирования дополнительной периферии (платы для управления двигателями, датчиковые платы, беспроводные интерфейсы, дисплеи, устройства ввода) - несколько десятков видов, более 300 вариантов исполнения.
3. Полностью адаптированная для конечного пользователя среда программирования, подходящая для всей линейки плат Arduino и их клонов, включая ПО для программирования контроллеров для ОС Android.

4. Свободная бесплатная лицензия на устройства и ПО.
5. Существует полный русский перевод языка Arduino, предназначенный для преодоления языкового барьера при распространении платформы по России.

3.3 Язык Ардуино

Язык программирования Arduino является стандартным C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающими написание первой работающей программы. На данный момент Arduino — это, пожалуй, самый удобный способ программирования устройств на микроконтроллерах.

Язык Arduino можно разделить на три раздела:

- Операторы
- Данные
- Функции

К разделу операторы относятся две обязательные функции, которые программист должен написать для Arduino:

- `setup;`
- `loop.`

Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле.

К разделу операторы относятся следующие подразделы:

- Управляющие операторы
- Синтаксис
- Арифметические операторы
- Операторы сравнения
- Логические операторы
- Унарные операторы

К разделу данные относятся следующие подразделы:

- Константы

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		35

- Типы данных
- Преобразование типов данных
- Область видимости переменных и квалификаторы

К разделу функции относятся следующие подразделы:

- Цифровой ввод/вывод
- Аналоговый ввод/вывод
- Дополнительные функции ввода/вывода
- Работа со временем
- Математические функции
- Тригонометрические функции
- Генераторы случайных значений
- Внешние прерывания
- Функции передачи данных

Так же ардуино имеет личные библиотеки, а именно:

- Servo
- EEPROM
- SPI
- Stepper

Servo

«Эта библиотека функций для Arduino контроллера предоставляет набор функций для управления сервоприводами. Стандартные сервоприводы позволяют поворачивать привод на определенный угол от 0 до 180 градусов обычно. Некоторые сервоприводы позволяют совершать полные обороты на заданной скорости.

Библиотека Servo позволяет одновременно управлять 12-ю сервоприводами на большинстве плат Arduino и 48-ю на ArduinoMega. На контроллерах отличных от Mega использование библиотеки отключает

возможность использовать выходы 9 и 10 в режиме ШИМ даже если привод не подключен к этим выводам. На плате Mega могут быть использованы до двенадцати сервоприводов без потери функционала широтно-импульсной модуляции (ШИМ). При использовании Mega для управления от 12 до 23 сервоприводов нельзя будет использовать выходы 11 и 12 для ШИМ.

EEPROM

Микроконтроллеры ATmega имеют свою энергонезависимую память, то есть у пользователей Ардуино есть возможность сохранять данные в этой памяти и они могут быть использованы после выключения-включения или перезагрузки контроллера. Arduino библиотека EEPROM предоставляет удобный и простой интерфейс работы с этой памятью.

SPI

Библиотека SPI позволяет контроллеру Arduino взаимодействовать с устройствами поддерживающими SPI протокол. Arduino в данном случае выступает в качестве ведущего устройства.

Stepper

Библиотека Stepper предоставляет удобный интерфейс управления биполярными и униполярными шаговыми двигателями. Для управления шаговым двигателем, в зависимости от его типа (биполярный или униполярный) и выбранного способа подключения, понадобятся некоторые дополнительные электронные компоненты»[13].

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Версия докум.	Подп.	Дата		37

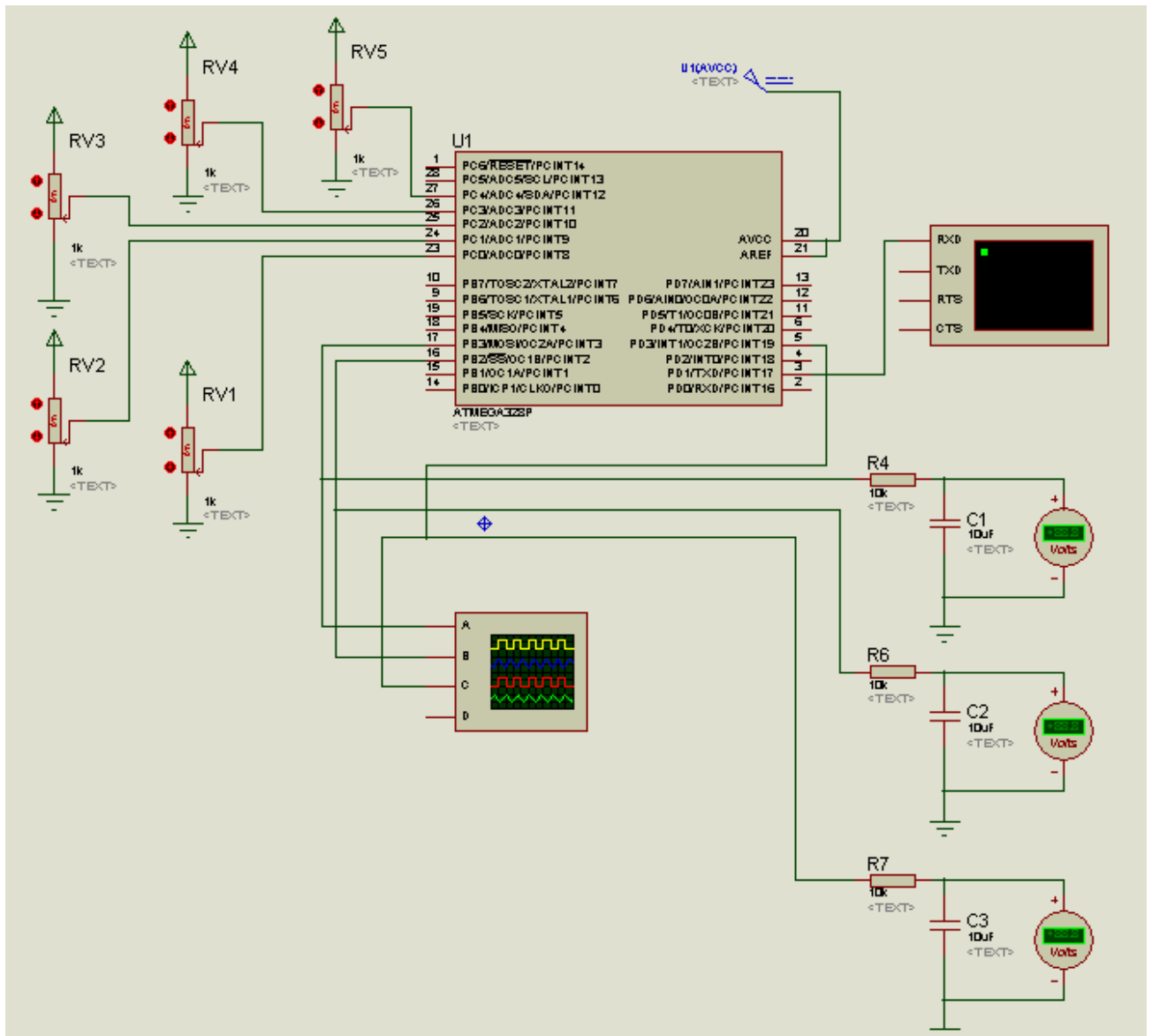


Рисунок 11 – Электрическая схема объекта управления

Так выглядит схема в общем виде. На ней представлены следующие элементы:

- источник питания U1(AVCC);
- микроконтроллер U1;
- резистор с переменным сопротивлениемRV – 5шт;
- вольтметр V – 3 шт;
- осциллограф OG;
- резисторы R – 3 шт;
- конденсаторы C – 3 шт.

Изм.	Лист	Недокум.	Подп.	Дата

Рассмотрим некоторые элементы в отдельности, какие они выполняют функции.

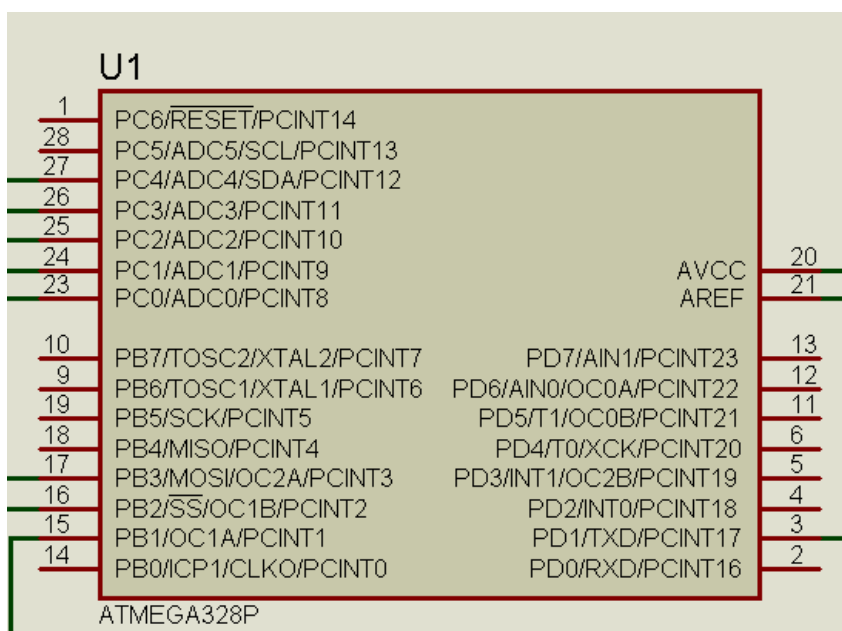


Рисунок 1 - Микроконтроллер АТМЕГА328Р

На рисунке 12 представлен микроконтроллер АТМЕГА328Р. Микроконтроллер является основой схемы, выполняет все основные функции, принимает и обрабатывает сигналы с датчиков, оказывает управляющее воздействие, а именно подаёт сигнал на форсунки и катушки зажигания в нужный момент, генерирует различные прерывания. Микроконтроллер имеет 28 выводов.

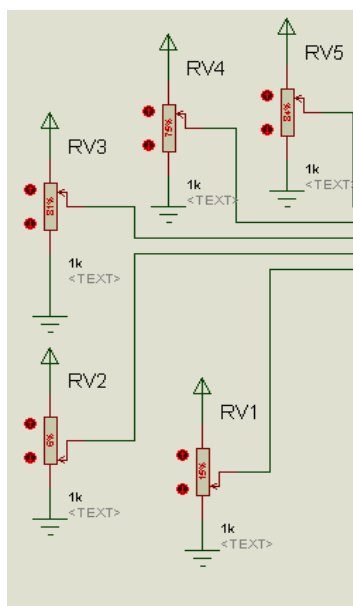


Рисунок 2 – Имитаторы датчиков

Изм.	Лист	Недокум.	Подп.	Дата
------	------	----------	-------	------

На рисунке 13 представлены устройства, имитирующие сигналы с датчиков.

3.4 Описание среды разработки

Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino.

Программа, написанная в среде Arduino, называется скетч. Скетч пишется в текстовом редакторе, имеющем инструменты вырезки/вставки, поиска/замены текста. Во время сохранения и экспорта проекта в области сообщений появляются пояснения, также могут отображаться возникшие ошибки. Окно вывода текста(консоль) показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины.

Дополнительные команды сгруппированы в пять меню: File, Edit, Sketch, Tools, Help. Доступность меню определяется работой, выполняемой в данный момент.

Блокнот (Sketchbook)

Средой Arduino используется принцип блокнота: стандартное место для хранения программ (скетчей). Скетчи из блокнота открываются через меню File-Sketchbook или кнопкой Open на панели инструментов. При первом запуске программы Arduino автоматически создается директория для блокнота. Расположение блокнота меняется через диалоговое окно Preferences.

Закладки, Файлы и Компиляция

Позволяют работать с несколькими файлами скетчей (каждый открывается в отдельной закладке). Файлы кода могут быть стандартными Arduino (без

										Лист
										41
Изм.	Лист	Недокум.	Подп.	Дата						

расширения), файлами C «расширение.c», файлами C++ «.cpp» или головными файлами (.h).

Загрузка скетча в Arduino

Перед загрузкой скетча требуется задать необходимые параметры в меню Tools-Board и Tools-SerialPort. Платформы описываются далее по тексту. В ОС Mac последовательный порт может обозначаться как dev/tty.usbserial-1B1 (для платы USB) или /dev/tty.USA19QW1b1P1.1 (для платы последовательной шины, подключенной через адаптер Keyspan USB-to-Serial). В ОС Windows порты могут обозначаться как COM1 или COM2 (для платы последовательной шины) или COM4, COM5, COM7 и выше (для платы USB). Определение порта USB производится в поле Последовательной шины USB Диспетчера устройств Windows. В ОС Linux порты могут обозначаться как /dev/ttyUSB0, /dev/ttyUSB1.

После выбора порта и платформы необходимо нажать кнопку загрузки на панели инструментов или выбрать пункт меню File-Upload to I/O Board. Современные платформы Arduino перезагружаются автоматически перед загрузкой. На старых платформах необходимо нажать кнопку перезагрузки. На большинстве плат во время процесса будут мигать светодиоды RX и TX. Среда разработки Arduino выведет сообщение об окончании загрузки или об ошибках.

При загрузке скетча используется Загрузчик (Bootloader) Arduino, небольшая программа, загружаемая в микроконтроллер на плате. Она позволяет загружать программный код без использования дополнительных аппаратных средств. Загрузчик (Bootloader) активен в течении нескольких секунд при перезагрузке платформы и при загрузке любого из скетчей в микроконтроллер. Работа Загрузчика (Bootloader) распознается по миганию светодиода (13 пин) (напр.: при перезагрузке платы).

Библиотеки

Библиотеки добавляют дополнительную функциональность скетчам, например, при работе с аппаратной частью или при обработке данных. Для использования библиотеки необходимо выбрать меню Sketch-ImportLibrary. Одна

										Лист
										42
Изм.	Лист	Недокум.	Подп.	Дата						

или несколько директив #include будут размещены в начале кода скетча с последующей компиляцией библиотек и вместе со скетчем. Загрузка библиотек требует дополнительного места в памяти Arduino. Неиспользуемые библиотеки можно удалить из скетча убрав директиву #include.

На Arduino.cc имеется список библиотек. Некоторые библиотеки включены в среду разработки Arduino. Другие могут быть загружены с различных ресурсов. Для установки скачанных библиотек необходимо создать директорию «libraries» в папке блокнота и затем распаковать архив. Например, для установки библиотеки DateTime ее файлы должны находиться в подпапке /libraries/DateTime папки блокнота.

Аппаратные средства других разработчиков

Поддерживаемые аппаратные средства других производителей добавляются в соответствующую подпапку папки блокнота. Устанавливаемые платформы могут включать собственные характеристики (в меню платформы), корневые библиотеки, загрузчик(Bootloader) и характеристики программатора. Для установки требуется распаковать архив в созданную папку. (Запрещено использовать наименование папки "arduino", так как могут быть перезаписаны встроенные данные платформы Arduino.) Для деинсталляции данных удаляется соответствующая директория.

Мониторинг последовательной шины (SerialMonitor)

Отображает данные посылаемые в платформу Arduino (плата USB или плата последовательной шины). Для отправки данных необходимо ввести текст и нажать кнопку Send или Enter. Затем выбирается скорость передачи из выпадающего списка, соответствующая значению Serial.begin в скетче. На ОС Mac или Linux платформа Arduino будет перезагружена (скетч начнется сначала) при подключении мониторинга последовательной шины.

Настройки

Некоторые настройки изменяются в окне Preferences (меню Arduino в ОС Mac или File в операционной системе Windows и Linux). Остальные настройки находятся в файле, месторасположение которого указано в окне Preferences.

Платформы

Выбор платформы влияет на: параметры (например: скорость центрального процессора и скорость передачи данных), используемые при компиляции и загрузке скетчей и на настройки записи загрузчика (Bootloader) микроконтроллера. Некоторые характеристики платформ различаются только по последнему параметру (загрузка Bootloader), таким образом, даже при удачной загрузке с соответствующим выбором может потребоваться проверка различия перед записью загрузчика (Bootloader).

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		44

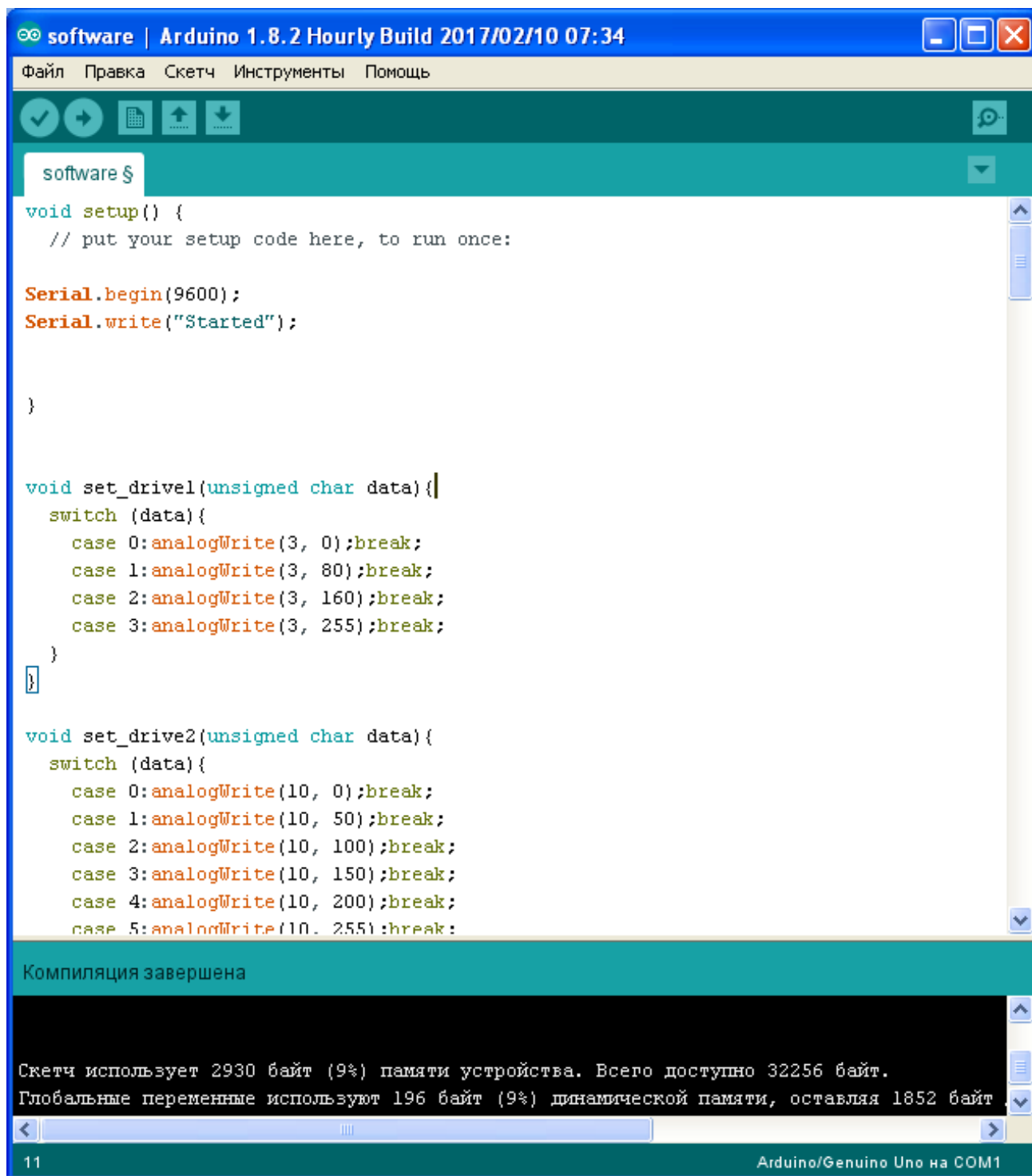


Рисунок 14 – Среда Arduino

Изм.	Лист	Недокум.	Подп.	Дата

13.03.02.2017.038.00 ПЗ

Лист

45

3.5 Разработка программы

Вызываем функция **setup**. Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек. Функция **setup** запускает только один раз, после каждой подачи питания или сброса платы Arduino.

```
void setup() {  
  // put your setup code here, to run once:
```

Устанавливаем режим работы заданного выхода(pin) как выхода.

```
  //pinMode(9, OUTPUT);  
  //pinMode(10, OUTPUT);  
  //pinMode(11, OUTPUT);
```

```
  Serial.begin(9600);  
  Serial.write("Started");  
}
```

```
void set_drive1(unsigned char data){
```

Оператор **switchcase** управляет процессом выполнения программы, позволяя задавать альтернативный код, который будет выполняться при разных условиях. В частности, оператор **switch** сравнивает значение переменной со значением, определенном в операторах **case**. Когда найден оператор **case**, значение которого равно значению переменной, выполняется программный код в этом операторе.

```
  switch (data){  
    case 0:analogWrite(9, 0);break;  
    case 1:analogWrite(9, 80);break;  
    case 2:analogWrite(9, 160);break;  
    case 3:analogWrite(9, 255);break;  
  }
```

```
void set_drive2(unsigned char data){  
  switch (data){  
    case 0:analogWrite(10, 0);break;
```

```

case 1:analogWrite(10, 80);break;
case 2:analogWrite(10, 160);break;
case 3:analogWrite(10, 255);break;
    }
}
voidset_fan(unsigned char data){
switch (data){
case 0:analogWrite(11, 0);break;
case 1:analogWrite(11, 80);break;
case 2:analogWrite(11, 160);break;
case 3:analogWrite(11, 255);break;
    }
}

```

После вызова функции `setup`, которая инициализирует и устанавливает первоначальные значения, функция `loop` делает точь-в-точь то, что означает её название, и крутится в цикле, позволяя программе совершать вычисления и реагировать на них.

```

voidloop() {
// put your main code here, to run repeatedly:
int t1,t2,t3,t4,t5;
int t1_state,t2_state,t3_state,t4_state,t5_state;
intstate;

```

Функция считывает значение с указанного аналогового входа. Напряжение поданное на аналоговый вход, обычно от 0 до 5 вольт будет преобразовано в значение от 0 до 1023.

```

t1 = analogRead (A0);
t2 = analogRead (A1);
t3 = analogRead (A2);
t4 = analogRead (A3);
t5 = analogRead (A4);

```

Данная функция передает данные через последовательный порт как ASCII текст. Эта функция может принимать различные типы данных. Так целые числа выводятся соответствующими им символами ASCII. Вещественные выводятся с помощью двух ASCII символов, для целой и дробной части. Байты передаются как символ с соответствующим номером. Символы и строки отсылаются как есть.

```
Serial.print(t1);  
Serial.print(" ");  
Serial.print(t2);  
Serial.print(" ");  
Serial.print(t3);  
Serial.print(" ");  
Serial.print(t4);  
Serial.print(" ");  
Serial.print(t5);  
Serial.print("\r\n");
```

Теперь задаемся условием входных параметров для регулирования климат-контролем.

```
if (t1 < 250)  
    t1_state = 0;  
if ((t1 > 251) && (t1 < 500))  
    t1_state = 1;  
if ((t1 > 501) && (t1 < 750))  
    t1_state = 2;  
if ((t1 > 751) && (t1 < 1024))  
    t1_state = 3;  
  
if (t2 < 600)  
    t2_state = 0;  
if ((t2 > 601) && (t2 < 1024))  
    t2_state = 1;  
  
if (t3 < 600)
```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		48

```

        t3_state = 0;
    if ((t3 > 601) && (t3 < 1024))
        t3_state = 1;

    if ((t4 > 0) && (t4 < 500))
        t4_state = 0;
    if ((t4 > 501) && (t4 < 1024))
        t4_state = 1;

    if (t5 < 250)
        t5_state = 0;
    if ((t5 > 251) && (t5 < 500))
        t5_state = 1;
    if ((t5 > 501) && (t5 < 750))
        t5_state = 2;
    if ((t5 > 751) && (t5 < 1024))
        t5_state = 3;

    state = (t1_state << 5) | (t2_state << 4) | (t3_state <<3) |
(t4_state << 2) | (t5_state << 0);

```

Выводим полученное состояние исходя из состояний каждого из датчиков.

```

Serial.print(state);
Serial.print("\r\n");

```

Исходя из полученного состояния по ниже приведенной таблице, программа определяет управляющее воздействие выходные узлов.

```

switch (state){
case 1 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 2 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 3 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 4 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 5 : set_drive1(0);set_drive2(5);set_fan(0);break;
.....
case 120 : set_drive1(3);set_drive2(0);set_fan(1);break;
case 121 : set_drive1(0);set_drive2(0);set_fan(1);break;
case 122 : set_drive1(1);set_drive2(0);set_fan(1);break;

```

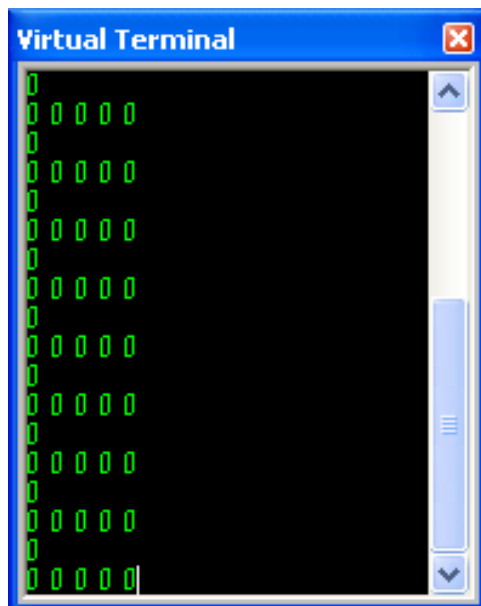



Рисунок 16 – Показания состояний первого результата

Данные состояний на терминале говорят о том, что: температура окружающей среды находится в пределах от -40 до -20; температура охлаждающей жидкости до 60 градусов; температура салона ниже заданной; на фотодатчик попадают солнечные лучи; выбранный режим подачи воздуха вниз.

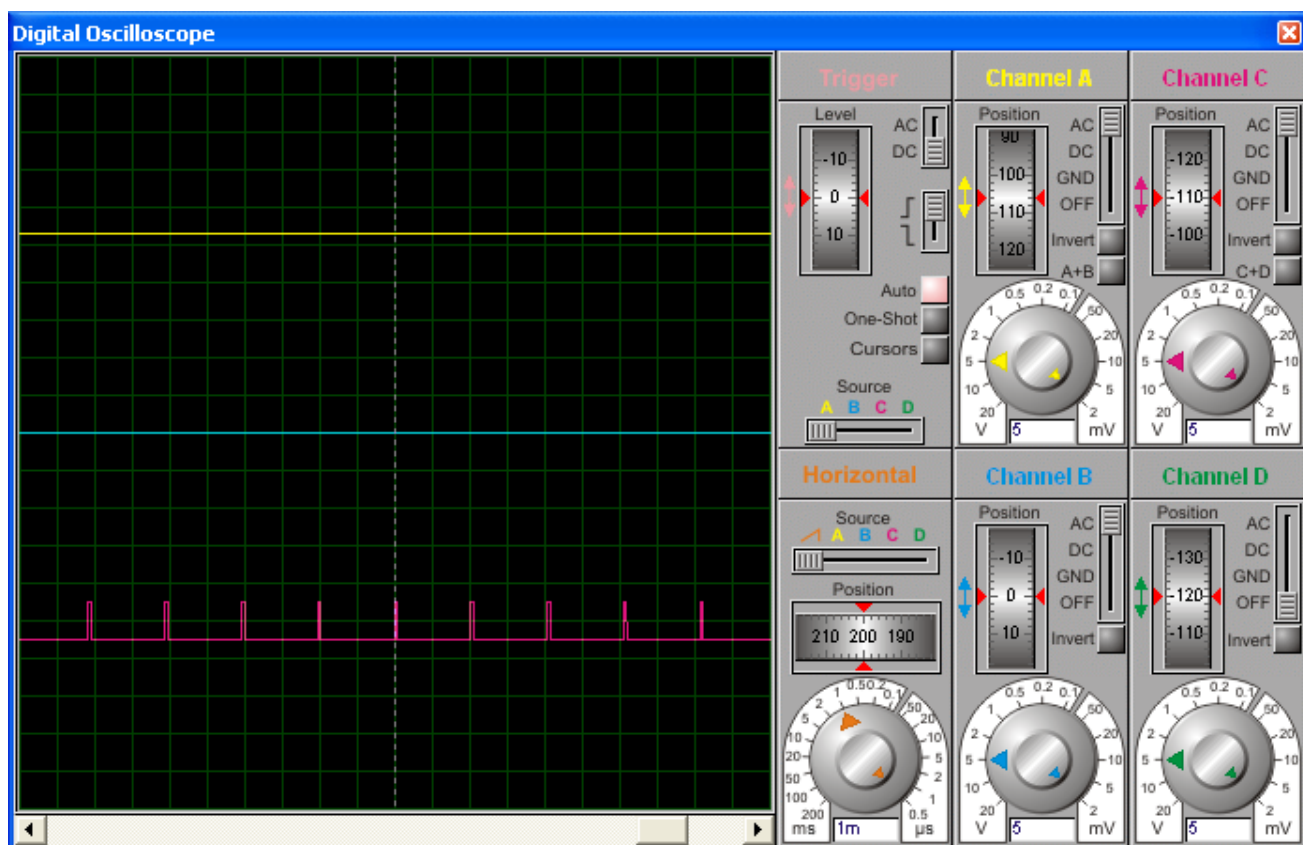


Рисунок 17 – Выходные импульсы первого результата

Изм.	Лист	Недокум.	Подп.	Дата

На рисунке 17 мы наблюдаем выходные импульсы при состояниях первого опыта. Первые две линии на осциллографе сплошные, это говорит о том, что на выходе у привода 1 и 2 соответственно, отсутствует напряжение. У вентилятора присутствует небольшое напряжение, так как осциллограф показывает выходной импульс малой ширины.

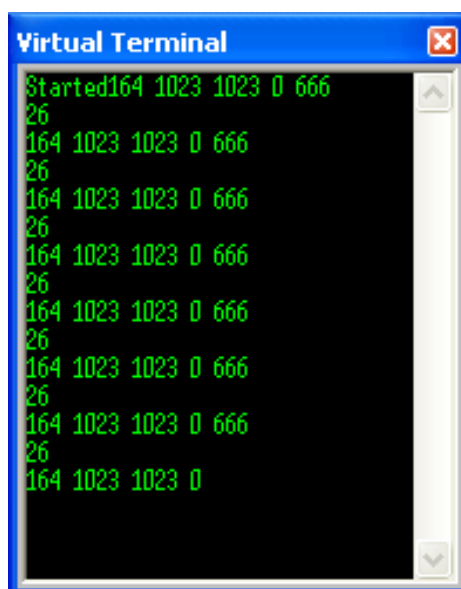


Рисунок 18 – Показания состояний второго результата

Данные состояний на терминале говорят о том, что: температура окружающей среды находится в пределах от -40 до -20; температура охлаждающей жидкости свыше 60 градусов; температура салона выше заданной; на фотодатчик попадают солнечные лучи; выбранный режим подачи воздуха вперед/вверх.

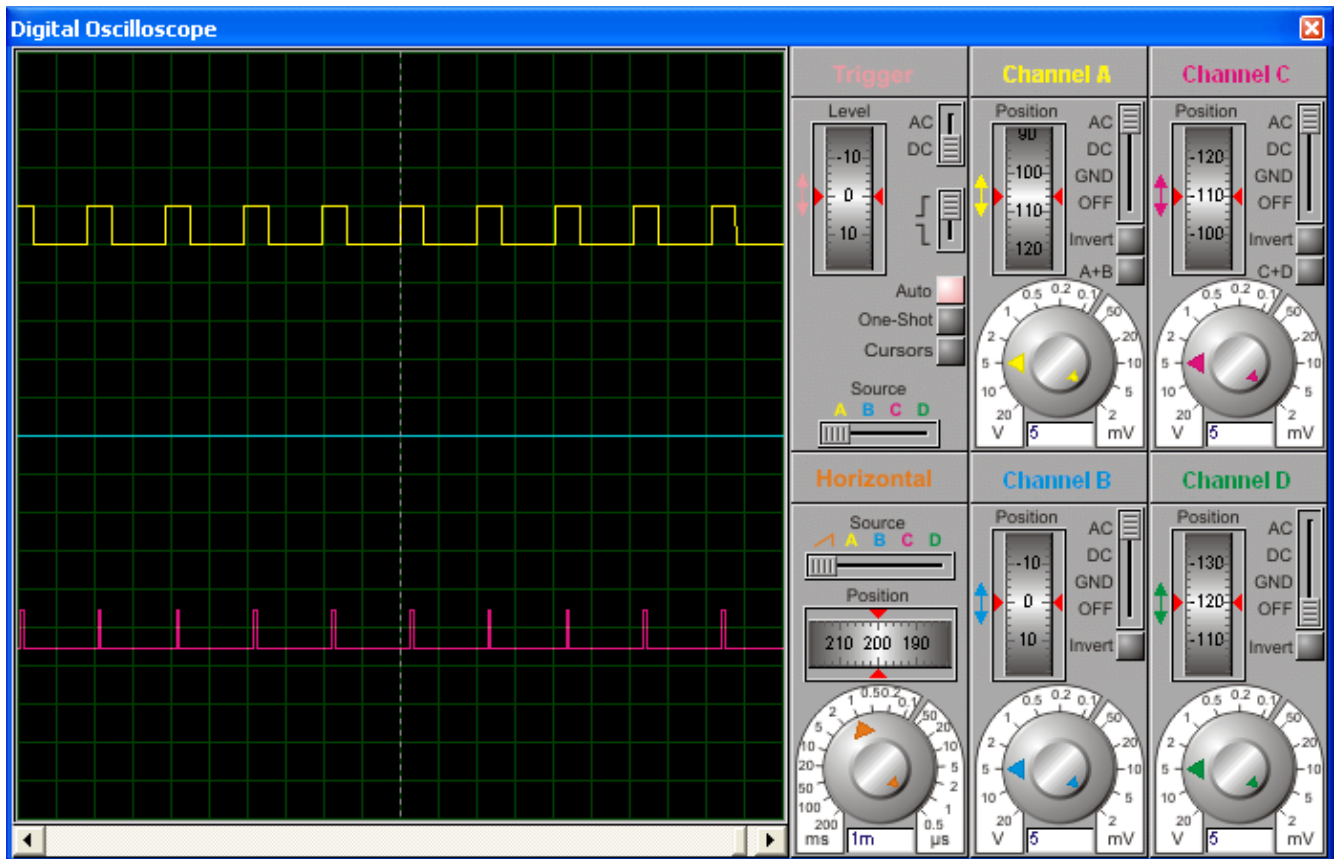


Рисунок 19 – Выходные импульсы второго результата

На рисунке 19 мы наблюдаем выходные импульсы при состояниях второго опыта. У привода один появились выходные импульсы, по сравнению с предыдущим опытом. У привода два выходные импульсы отсутствуют. У вентилятора присутствует небольшое напряжение.

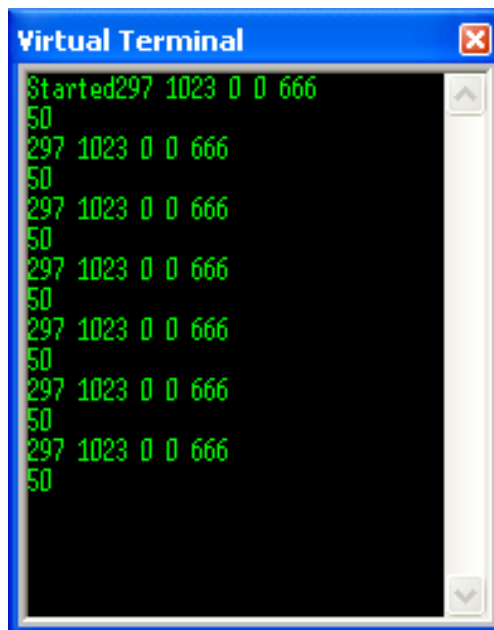


Рисунок 20 – Показания состояний третьего результата

Данные состояний на терминале говорят о том, что: температура окружающей среды находится в пределах от -20 до 0; температура охлаждающей жидкости свыше 60 градусов; температура салона ниже заданной; на фотодатчик попадают солнечные лучи; выбранный режим подачи воздуха вперед/вверх.

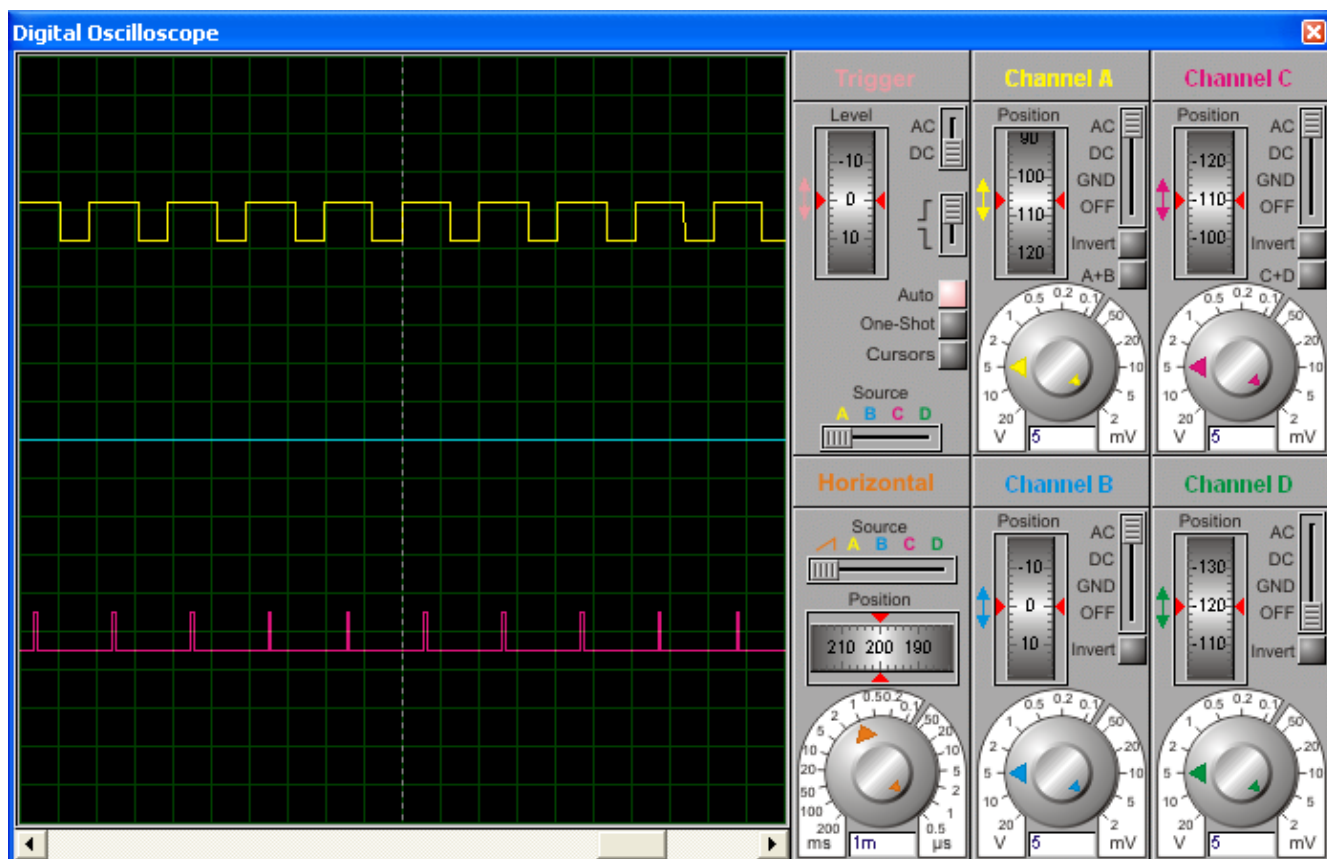


Рисунок 21 – Выходные импульсы третьего результата

На рисунке 21 мы наблюдаем выходные импульсы при состояниях третьего опыта. У привода один выходные импульсы снова увеличились по сравнению с предыдущим опытом, а это значит, что напряжение на выходе увеличивается. У привода два выходные импульсы по прежнему отсутствуют. У вентилятора присутствует небольшое напряжение.

Изм.	Лист	Недокум.	Подп.	Дата

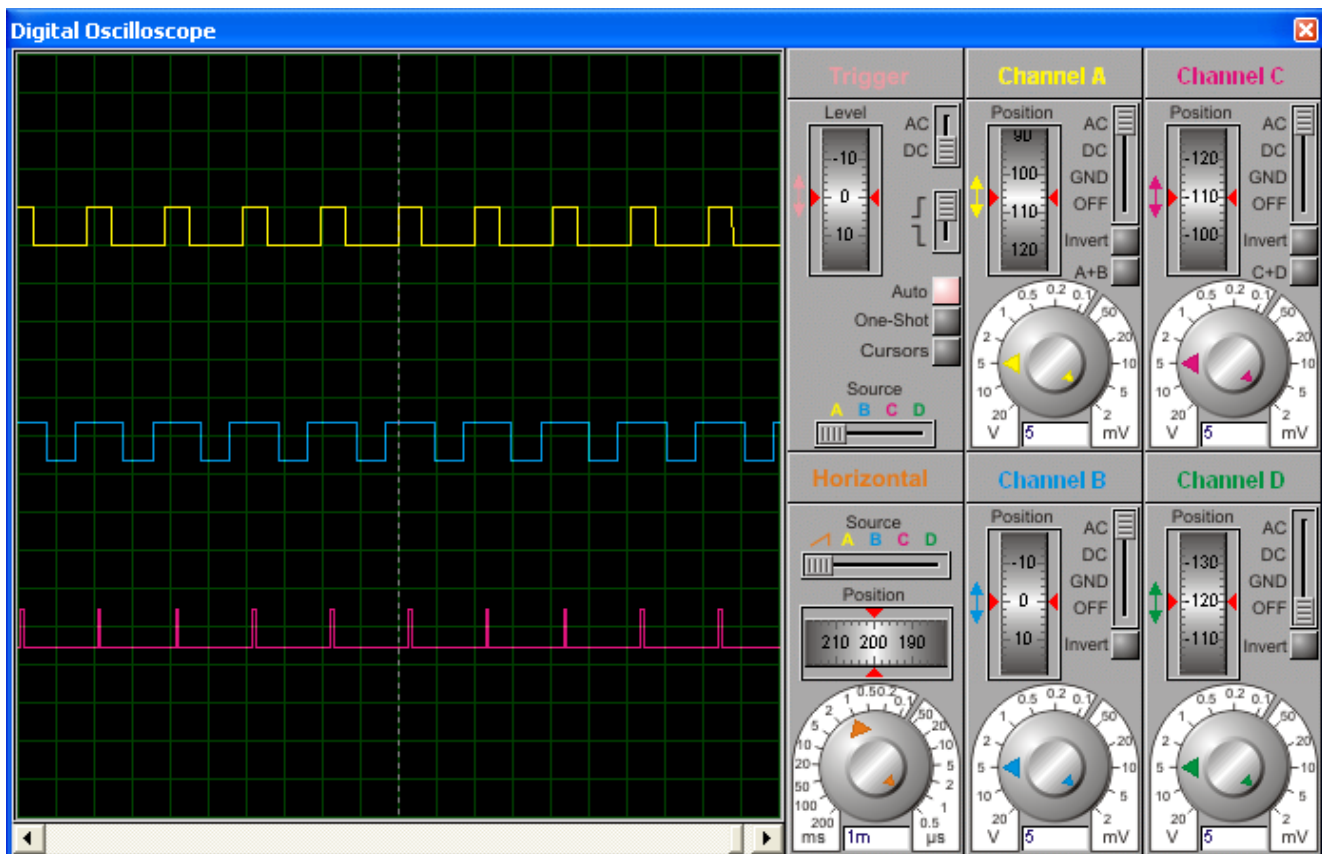


Рисунок 23 – Выходные импульсы четвертого результата

На рисунке 23 наблюдаем выходные импульсы при состояниях четвертого опыта. У привода один выходной импульс меньше чем у привода два, соответственно и напряжение подаваемое на привод два выше. У вентилятора присутствует небольшое напряжение.

```

1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0 0 1023 0
100
1023 0
  
```

Рисунок 24 – Показания состояний пятого результата

Данные состояний на терминале говорят о том, что: температура окружающей среды находится в пределах от 00 до 40; температура охлаждающей жидкости до 60 градусов; температура салона ниже заданной; на фотодатчик не попадают солнечные лучи; выбранный режим подачи воздуха вниз.

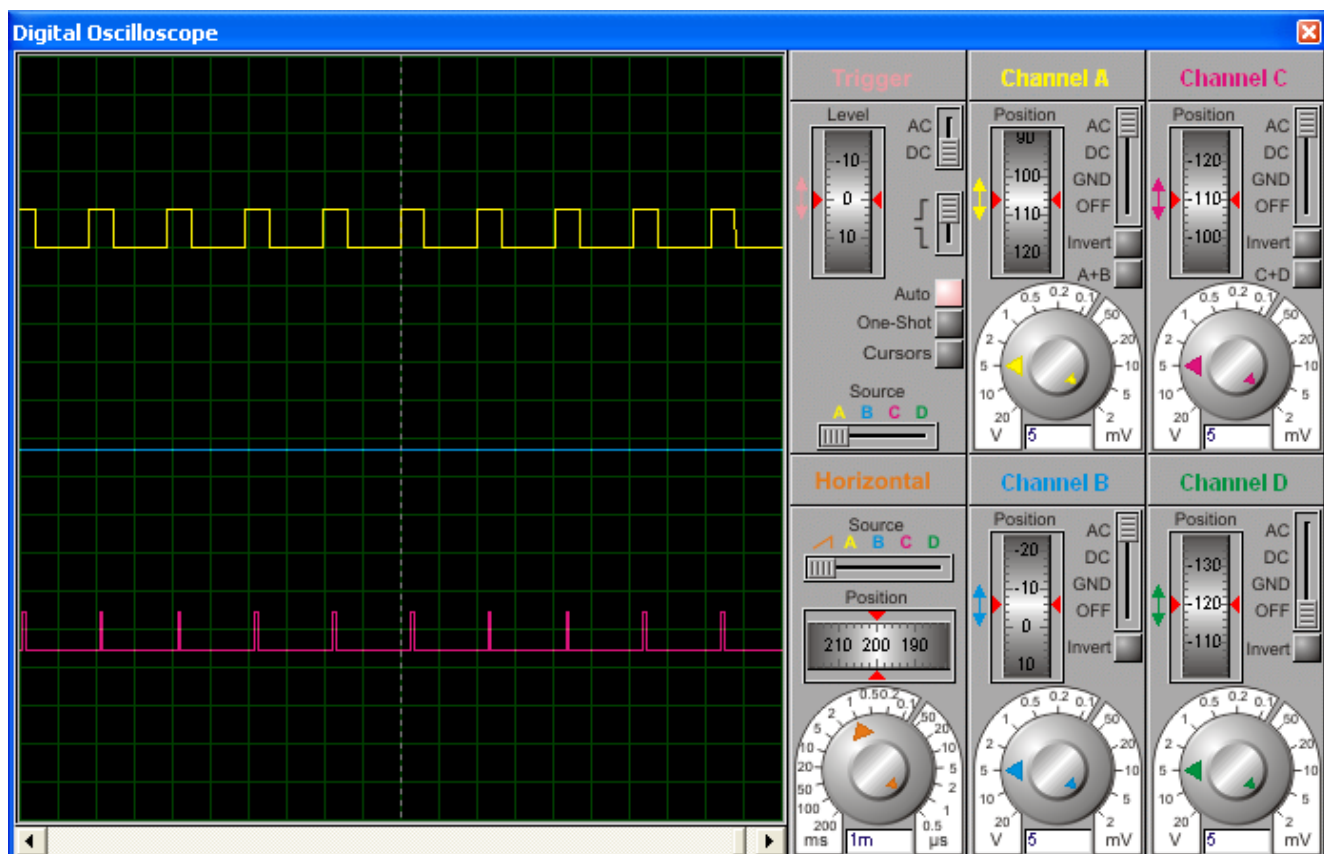


Рисунок 25 – Выходные импульсы пятого результата

На рисунке 25 наблюдаем выходные импульсы при состояниях пятого опыта. Привод один имеет выходной импульс средней ширины. Выходной импульс на втором приводе отсутствует. У вентилятора присутствует небольшое напряжение.

Изм.	Лист	Недокум.	Подп.	Дата

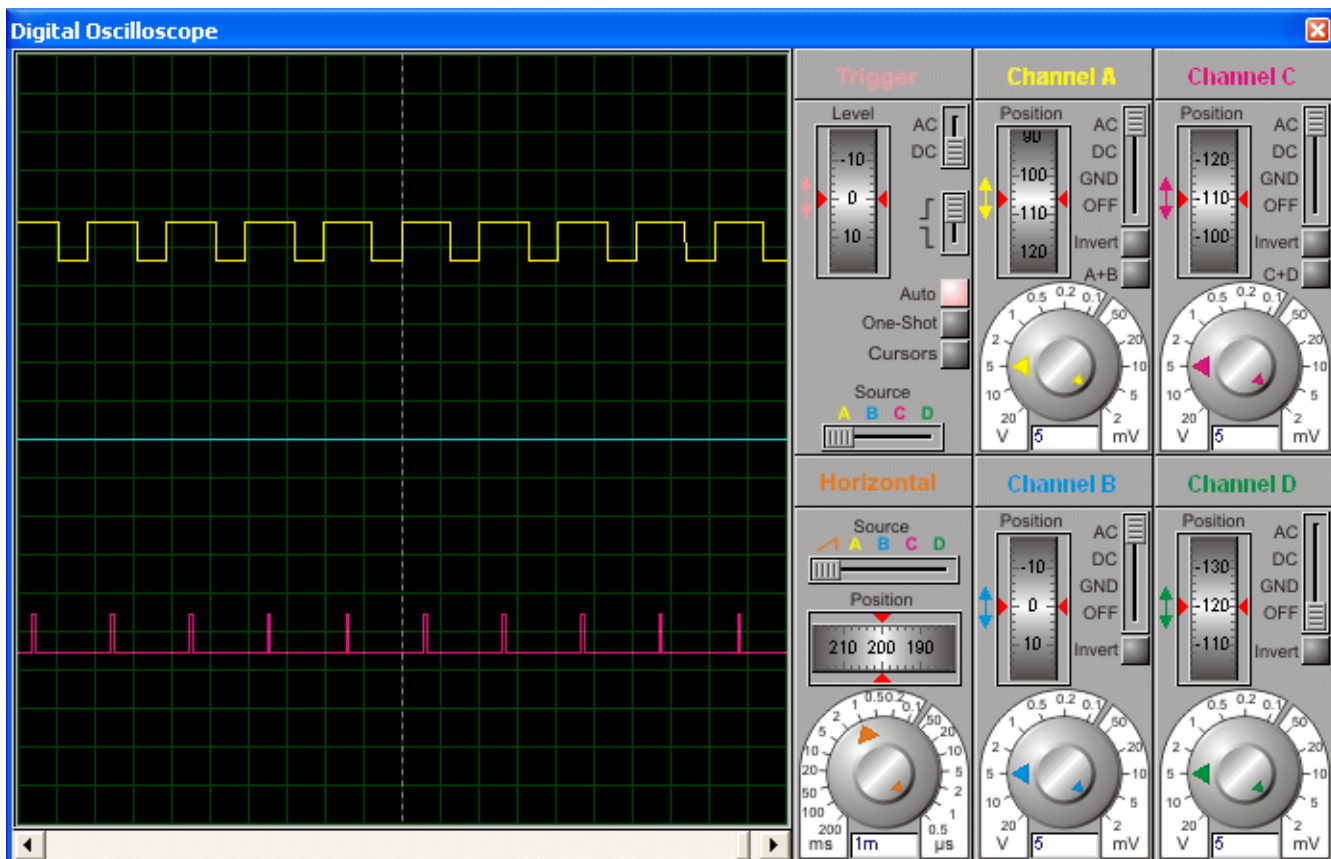


Рисунок 28 – Выходные импульсы шестого результата

На рисунке 28 мы наблюдаем выходные импульсы при состояниях шестого (последнего) опыта. Привод один имеет широкие выходные импульсы, это говорит о высоком выходном напряжении на приводе

Изм.	Лист	Недокум.	Подп.	Дата

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе была разработана система автоматического управления печкой, вентилятором и кондиционером на примере автомобиля HyundaiSolaris. Для достижения поставленной цели выполнен обзор научной и технической литературы, рассмотрены различные системы управления.

Сформулированы задачи, решаемые в процессе разработки, определены требования к функциям, разработан алгоритм управления климатической установкой автомобиля. Для реализации работы алгоритма выбраны следующие платформы для программирования и моделирования: пакеты программ ProteusDesignSuite и Arduino.

В среде ProteusDesignSuite разработана электрическая схема системы управления с соответствующими имитаторами датчиков, исполнительных устройств и контрольно-измерительных приборов. Основу схемы составила модель микропроцессора АТмега 328, позволяющая продемонстрировать работу разрабатываемой программы.

В среде программирования Arduino разработана программа для управления климатической установкой автомобиля.

Проведенные исследовательские эксперименты по проверке правильности функционирования системы управления, в серии экспериментов, подтвердили правильность функционирования разработанной системы управления.

Таким образом, по полученным результатам проделанной работы можно утверждать, что все поставленные в ходе работы задачи выполнены, и цель работы достигнута.

Уникальность выполненной работы составляет 65 %.

Руководитель Мартьянов А.С. _____

Студент Ишимов Э.Р. _____

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		60

СПИСОК ЛИТЕРАТУРЫ

- 1 Брей Б. Микропроцессоры INTEL. – СПб.: БХВ – Петербург, 2005. – 1028 с
 - 2 Вальпа О. Полезные схемы с применением микроконтроллеров и ПЛИС(+СД).- М.: Изд.дом «Додэка-XXI», 2006. – 416 с.
 - 3 Костров Б.В. Микропроцессорные системы и микроконтроллеры. – М.: «ТехБук», 2007. – 320 с.
 - 4 Тавернье К. PIC-микроконтроллеры. – М.: ДМК Пресс, 2003. – 272 с.
 - 5 Терехов В.М. Системы управления электроприводов. – М.: Изд. центр «Академия», 2006. – 304 с.
 - 6 Евстифеев А.В. «Микроконтроллеры AVR семейства Mega» – Москва – Издательский дом «Додэка - XXI», 2007.-595с.
 - 7 Вольфганг Трамперт «Измерение, управление и регулирование с помощью AVR-микроконтроллеров» (глава2 полностью посвящена динамической индикации, приведен расчет токоограничивающих резисторов).
 - 8 В. Н. Баранов «Применение микроконтроллеров AVR: схемы, алгоритмы, программы»(глава4 посвящена динамической индикации)
 - 9 Угрюмов Е. Цифровая схемотехника. - СПб.: БХВ - Петербург, 2002. - 528 с.
Костров Б.В. Микропроцессорные системы и микроконтроллеры. - М.: "ТехБук", 2007. - 320 с.
 - 10 Егоров Ф.А., Неугодииков А.П., Поспелов В.И. Микроконтроллер.—Москва, 2008
 - 11 Омельченко Е.Я., Танич В.О., Маклаков А.С. Краткий обзор и перспективы применения микропроцессорной платформы Arduino.// Электротехнические системы и комплексы.— Магнитогорск, 2013
- Электронные ресурсы:
- 12 <http://www.chipdip.ru/>
 - 13 <http://arduino.ru/>
 - 14 <http://compiro.ru/> - «Мир компьютерного железа»

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		61

ПРИЛОЖЕНИЕ. ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ

```
void setup() {
    // put your setup code here, to run once:

    //pinMode(9, OUTPUT);
    //pinMode(10, OUTPUT);
    //pinMode(11, OUTPUT);

    Serial.begin(9600);
    Serial.write("Started");
}
void set_drive1(unsigned char data){
    switch (data){
        case 0:analogWrite(9, 0);break;
        case 1:analogWrite(9, 80);break;
        case 2:analogWrite(9, 160);break;
        case 3:analogWrite(9, 255);break;
    }
}
void set_drive2(unsigned char data){
    switch (data){
        case 0:analogWrite(10, 0);break;
        case 1:analogWrite(10, 80);break;
        case 2:analogWrite(10, 160);break;
        case 3:analogWrite(10, 255);break;
    }
}
void set_fan(unsigned char data){
    switch (data){
        case 0:analogWrite(11, 0);break;
        case 1:analogWrite(11, 80);break;
        case 2:analogWrite(11, 160);break;
        case 3:analogWrite(11, 255);break;
    }
}
```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		62

```

void loop() {
    // put your main code here, to run repeatedly:
    int t1,t2,t3,t4,t5;
    int t1_state,t2_state,t3_state,t4_state,t5_state;
    int state;

    t1 = analogRead (A0);
    t2 = analogRead (A1);
    t3 = analogRead (A2);
    t4 = analogRead (A3);
    t5 = analogRead (A4);

    Serial.print(t1);
    Serial.print(" ");
    Serial.print(t2);
    Serial.print(" ");
    Serial.print(t3);
    Serial.print(" ");
    Serial.print(t4);
    Serial.print(" ");
    Serial.print(t5);
    Serial.print("\r\n");

    if (t1 < 250)
        t1_state = 0;
    if ((t1 > 251) && (t1 < 500))
        t1_state = 1;
    if ((t1 > 501) && (t1 < 750))
        t1_state = 2;
    if ((t1 > 751) && (t1 < 1024))
        t1_state = 3;

    if (t2 < 600)
        t2_state = 0;

```

```

if ((t2 > 601) && (t2 < 1024))
    t2_state = 1;

if (t3 < 600)
    t3_state = 0;
if ((t3 > 601) && (t3 < 1024))
    t3_state = 1;

if ((t4 > 0) && (t4 < 500))
    t4_state = 0;
if ((t4 > 501) && (t4 < 1024))
    t4_state = 1;

if (t5 < 250)
    t5_state = 0;
if ((t5 > 251) && (t5 < 500))
    t5_state = 1;
if ((t5 > 501) && (t5 < 750))
    t5_state = 2;
if ((t5 > 751) && (t5 < 1024))
    t5_state = 3;

state = (t1_state << 5) | (t2_state << 4) | (t3_state <<3) |
(t4_state << 2) | (t5_state << 0);

Serial.print(state);
Serial.print("\r\n");

switch (state){
case 1 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 2 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 3 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 4 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 5 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 6 : set_drive1(1);set_drive2(5);set_fan(0);break;

```

```

case 7 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 8 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 9 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 10 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 11 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 12 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 13 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 14 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 15 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 16 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 17 : set_drive1(0);set_drive2(5);set_fan(2);break;
case 18 : set_drive1(1);set_drive2(5);set_fan(2);break;
case 19 : set_drive1(2);set_drive2(5);set_fan(2);break;
case 20 : set_drive1(3);set_drive2(5);set_fan(2);break;
case 21 : set_drive1(0);set_drive2(5);set_fan(2);break;
case 22 : set_drive1(1);set_drive2(5);set_fan(2);break;
case 23 : set_drive1(2);set_drive2(5);set_fan(2);break;
case 24 : set_drive1(3);set_drive2(5);set_fan(2);break;
case 25 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 26 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 27 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 28 : set_drive1(3);set_drive2(4);set_fan(1);break;
case 29 : set_drive1(0);set_drive2(3);set_fan(1);break;
case 30 : set_drive1(1);set_drive2(3);set_fan(1);break;
case 31 : set_drive1(2);set_drive2(3);set_fan(1);break;
case 32 : set_drive1(3);set_drive2(3);set_fan(1);break;
case 33 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 34 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 35 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 36 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 37 : set_drive1(0);set_drive2(5);set_fan(0);break;
case 38 : set_drive1(1);set_drive2(5);set_fan(0);break;
case 39 : set_drive1(2);set_drive2(5);set_fan(0);break;
case 40 : set_drive1(3);set_drive2(5);set_fan(0);break;
case 41 : set_drive1(0);set_drive2(5);set_fan(1);break;

```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Докум.	Подп.	Дата		65

```

case 42 : set_drive1(1);set_drive2(5);set_fan(1);break;
case 43 : set_drive1(2);set_drive2(5);set_fan(1);break;
case 44 : set_drive1(3);set_drive2(5);set_fan(1);break;
case 45 : set_drive1(0);set_drive2(5);set_fan(1);break;
case 46 : set_drive1(1);set_drive2(5);set_fan(1);break;
case 47 : set_drive1(2);set_drive2(5);set_fan(1);break;
case 48 : set_drive1(3);set_drive2(5);set_fan(1);break;
case 49 : set_drive1(0);set_drive2(5);set_fan(2);break;
case 50 : set_drive1(1);set_drive2(5);set_fan(2);break;
case 51 : set_drive1(2);set_drive2(5);set_fan(2);break;
case 52 : set_drive1(3);set_drive2(5);set_fan(2);break;
case 53 : set_drive1(0);set_drive2(5);set_fan(2);break;
case 54 : set_drive1(1);set_drive2(5);set_fan(2);break;
case 55 : set_drive1(2);set_drive2(5);set_fan(2);break;
case 56 : set_drive1(3);set_drive2(5);set_fan(2);break;
case 57 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 58 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 59 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 60 : set_drive1(3);set_drive2(4);set_fan(1);break;
case 61 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 62 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 63 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 64 : set_drive1(3);set_drive2(4);set_fan(1);break;
case 65 : set_drive1(0);set_drive2(5);set_fan(1);break;
case 66 : set_drive1(1);set_drive2(5);set_fan(1);break;
case 67 : set_drive1(2);set_drive2(5);set_fan(1);break;
case 68 : set_drive1(3);set_drive2(5);set_fan(1);break;
case 69 : set_drive1(0);set_drive2(5);set_fan(1);break;
case 70 : set_drive1(1);set_drive2(5);set_fan(1);break;
case 71 : set_drive1(2);set_drive2(5);set_fan(1);break;
case 72 : set_drive1(3);set_drive2(5);set_fan(1);break;
case 73 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 74 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 75 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 76 : set_drive1(3);set_drive2(4);set_fan(1);break;

```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		66

```

case 77 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 78 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 79 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 80 : set_drive1(3);set_drive2(4);set_fan(1);break;
case 81 : set_drive1(0);set_drive2(5);set_fan(2);break;
case 82 : set_drive1(1);set_drive2(5);set_fan(2);break;
case 83 : set_drive1(2);set_drive2(5);set_fan(2);break;
case 84 : set_drive1(3);set_drive2(5);set_fan(2);break;
case 85 : set_drive1(0);set_drive2(4);set_fan(2);break;
case 86 : set_drive1(1);set_drive2(4);set_fan(2);break;
case 87 : set_drive1(2);set_drive2(4);set_fan(2);break;
case 88 : set_drive1(3);set_drive2(4);set_fan(2);break;
case 89 : set_drive1(0);set_drive2(3);set_fan(1);break;
case 90 : set_drive1(1);set_drive2(3);set_fan(1);break;
case 91 : set_drive1(2);set_drive2(3);set_fan(1);break;
case 92 : set_drive1(3);set_drive2(3);set_fan(1);break;
case 93 : set_drive1(0);set_drive2(2);set_fan(1);break;
case 94 : set_drive1(1);set_drive2(2);set_fan(1);break;
case 95 : set_drive1(2);set_drive2(2);set_fan(1);break;
case 96 : set_drive1(3);set_drive2(2);set_fan(1);break;
case 97 : set_drive1(0);set_drive2(4);set_fan(1);break;
case 98 : set_drive1(1);set_drive2(4);set_fan(1);break;
case 99 : set_drive1(2);set_drive2(4);set_fan(1);break;
case 100 : set_drive1(3);set_drive2(4);set_fan(1);break;
case 101 : set_drive1(0);set_drive2(3);set_fan(1);break;
case 102 : set_drive1(1);set_drive2(3);set_fan(1);break;
case 103 : set_drive1(2);set_drive2(3);set_fan(1);break;
case 104 : set_drive1(3);set_drive2(3);set_fan(1);break;
case 105 : set_drive1(0);set_drive2(2);set_fan(1);break;
case 106 : set_drive1(1);set_drive2(2);set_fan(1);break;
case 107 : set_drive1(2);set_drive2(2);set_fan(1);break;
case 108 : set_drive1(3);set_drive2(2);set_fan(1);break;
case 109 : set_drive1(0);set_drive2(1);set_fan(1);break;
case 110 : set_drive1(1);set_drive2(1);set_fan(1);break;
case 111 : set_drive1(2);set_drive2(1);set_fan(1);break;

```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		67


```
case 112 : set_drive1(3);set_drive2(1);set_fan(1);break;
case 113 : set_drive1(0);set_drive2(1);set_fan(1);break;
case 114 : set_drive1(1);set_drive2(1);set_fan(1);break;
case 115 : set_drive1(2);set_drive2(1);set_fan(1);break;
case 116 : set_drive1(3);set_drive2(1);set_fan(1);break;
case 117 : set_drive1(0);set_drive2(0);set_fan(1);break;
case 118 : set_drive1(1);set_drive2(0);set_fan(1);break;
case 119 : set_drive1(2);set_drive2(0);set_fan(1);break;
case 120 : set_drive1(3);set_drive2(0);set_fan(1);break;
case 121 : set_drive1(0);set_drive2(0);set_fan(1);break;
case 122 : set_drive1(1);set_drive2(0);set_fan(1);break;
case 123 : set_drive1(2);set_drive2(0);set_fan(1);break;
case 124 : set_drive1(3);set_drive2(0);set_fan(1);break;
case 125 : set_drive1(0);set_drive2(0);set_fan(2);break;
case 126 : set_drive1(1);set_drive2(0);set_fan(2);break;
case 127 : set_drive1(2);set_drive2(0);set_fan(2);break;
case 128 : set_drive1(3);set_drive2(0);set_fan(2);break;
}
}
```

					13.03.02.2017.038.00 ПЗ	Лист
Изм.	Лист	Недокум.	Подп.	Дата		68