

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 23.06.2024 19:02:12
Уникальный программный ключ:
0b817ca911e6668abbf05d126d79eff1c11eabbf73e943df4a4851fdb56d089


МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра космического приборостроения и средств связи

УТВЕРЖДАЮ

Проректор по учебной работе


О.Г. Локтионова

« 11 » 06



ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ КОНТРОЛЛЕРОВ

Методические указания к лабораторным занятиям по дисциплине
«Промышленные контроллеры и встраиваемые микропроцессорные
системы»

УДК 621.382

Составители: О.Г. Бондарь

Рецензент

Кандидат технических наук, доцент кафедры космического приборостроения и систем связи *И.Г. Бабанин*

Программирование промышленных контроллеров: методические указания к лабораторным занятиям по дисциплине «Промышленные контроллеры и встраиваемые микропроцессорные системы» / Юго-Зап. гос. ун-т; сост.: О.Г. Бондарь. – Курск, 2024. – 64 с.

Содержатся базовые сведения о аппаратных средствах промышленных контроллеров, графических языках программирования стандарта МЭК 61131-3 LD (Ladder Diagrams), FBD (Function Block Diagram) и среде разработки FLProg. Приведен порядок выполнения лабораторных работ и задания.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям автоматике и электроники (УМО АЭ).

Предназначены для студентов направления подготовки бакалавров 11.03.03 «Конструирование и технология электронных средств». Могут быть полезны для бакалавров других направлений подготовки, изучающих применение микропроцессорных систем в промышленности.

Текст печатается в авторской редакции

Подписано в печать 11.06.24. Формат 60×84 1/16.
Усл. печ. л. 3,72. Уч.-изд. л. 3,37. Тираж 100 экз. Заказ 522.
Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	3
Введение	5
Планируемые результаты обучения	7
Оборудование и программное обеспечение	8
Правила техники безопасности при проведении лабораторных работ	9
Общие требования безопасности	9
Требования безопасности перед началом работ	10
Требования безопасности во время работы	10
Требования безопасности по окончании работы	10
1. Среда разработки FLProg и программирование ПЛК на языке релейных диаграмм LD	12
1.1 Цель работы	12
1.2 ПЛК на основе платформы ARDUINO	12
1.3 Устройство и назначение выводов контроллеров	13
1.4 Многофункциональная плата расширения	17
1.5 Программирование контроллеров на языке релейных диаграмм LD	19
1.6 Компиляция и запись проекта в контроллер	26
1.7 Подготовка к работе	27
1.8. Вопросы для самоконтроля	27
1.9 Порядок выполнения работы	28
1.10 Содержание отчёта	30
1.11 Контрольные вопросы	31
2 Программирование ПЛК на языке функциональных блоковых диаграмм....	33
2.1 Цель работы	33
2.2 Особенности языка функциональных блоковых диаграмм	33
2.2.1 Базовые элементы	34
2.2.2 Программирования контроллера на языке FBD	38
2.3 Компиляция и запись проекта в контроллер	41
2.4 Подготовка к работе	42
2.5 Вопросы для самоконтроля	42
2.6 Порядок выполнения работы	42
2.7 Содержание отчёта	44

2.8 Контрольные вопросы	44
3 Ввод аналоговых сигналов в МК AVR	46
3.1 Цель работы	46
3.2 Измерение температуры с помощью термометров сопротивления	46
3.2.1 Основные параметр ТС.....	48
3.3 Двухпроводное подключение ТС с подавлением влияния сопротивления соединительных проводов	52
3.4 Подготовка к работе.....	55
3.5 Вопросы для самоконтроля	55
3.6 Порядок выполнения работы	56
3.7 Содержание отчёта.....	57
3.8 Контрольные вопросы	58
Литература	59
Приложение 1	61

Введение

С появлением в 70-х годах прошлого столетия микропроцессорной техники начался новый этап в развитии электронной техники. Микроконтроллеры, используемые в различных устройствах, выполняют функции интерпретации данных, поступающих с клавиатуры пользователя или от датчиков, определяющих параметры окружающей среды, обеспечивают связь между различными устройствами системы и передают данные другим приборам. Разработка систем на базе микроконтроллеров – микропроцессорных систем (МПС) – это как аппаратная, так и программная задача.

Микропроцессорные системы, которые непосредственно взаимодействуют с объектом управления или контроля, принято называть встроенными. Встраиваемые вычислительные системы являются важнейшим элементом любой современной системы автоматизации, медицинского или бытового прибора, устройства связи. Во всех этих устройствах одним из основополагающих принципов внутренней организации и применения является принцип программируемости. Традиционное разделение вычислительных устройств на аппаратную и программную составляющие определило существование и активное развитие соответствующих технологий проектирования этих частей. Создание программного обеспечения встроенной системы существенно отличается от программирования информационных систем общего назначения. Здесь используются специальные технические решения и инструментальные средства на всех уровнях и этапах процесса проектирования. Следует отметить появление даже специального термина, определяющего данную область программирования – встроенное программное обеспечение.

Микроконтроллеры проникли и в сферу управления производственными процессами. Основной тенденцией развития современных систем управления является применение для автоматизации одинаковых базовых решений и стандартных компонентов. Уже на стадии проектирования отдельные элементы

комплексов ориентируются на совместное использование. При этом кроме базовых компонентов – программируемых логических контроллеров (**ПЛК**) в состав комплексов включаются панели операторов, устройства связи с объектами, компоненты промышленной сети, унифицированные преобразователи для подключения датчиков.

Процессоры ПЛК являются маломощными по сравнению с процессорами персональных компьютеров. Обратной стороной является отработанность схемотехнических и программных решений, обеспечивающая высокую надёжность промышленных контроллеров. Вместе с тем в рамках одного конструктива производится серия модулей разной вычислительной мощности, что позволяет оптимизировать стоимость системы в целом.

Унификация базовых компонентов коснулась и технологий программирования. Использование типовой аппаратной базы универсальных программируемых контроллеров избавляет проектировщика систем управления от необходимости их схемотехнического проектирования и изготовления. Поэтому можно сказать, что сегодня проекты с применением ПЛК на 90% являются программным продуктом.

При этом учитывается та особенность, что эксплуатацией оборудования систем управления занимаются инженерные работники, не являющиеся программистами. Это потребовало специфических подходов к самой процедуре программирования ПЛК, которая должна ориентироваться на специалистов в области автоматизации, технологов, системных интеграторов. В результате унификации стандарт МЭК 61131-3 предложил пять языков технологического проектирования.

- Sequential Function Chart (SFC) – язык последовательных функциональных блоков;
- Function Block Diagram (**FBD**) – язык функциональных блоковых диаграмм;

- Ladder Diagrams (**LD**) – язык релейных диаграмм;
- Statement List (**ST**) – язык структурированного текста;
- Instruction List (**IL**) – язык инструкций.

SFC, FBD. LD являются графическими языками, языки IL и ST — текстовые.

В цикле лабораторных работ рассмотрены элементы программирования на графических языках LD и FBD. Язык LD опирается на схемотехнику релейных схем, а для расширения его возможностей он дополнен функциональными блоками, в т.ч. позволяющими проводить арифметические вычисления. Язык функциональных блоковых диаграмм FBD также является графическим языком программирования, использующим аналогию с электрической схемой.

Разработка программного обеспечения для систем управления наиболее эффективно реализуется не средствами классического программирования, а применением готовых инструментальных проблемно-ориентированных средств. Такие инструменты существуют для создания программ рабочих и диспетчерских станций (SCADA-системы) и для массово применяемых программируемых контроллеров. Поскольку программное обеспечение привязано к конкретному набору аппаратных ресурсов, средств организации человеко-машинного интерфейса, средств связи, то говорят о создании проекта, а не программы.

Для знакомства с ПЛК, их программированием на языках LD и FBD используется среда разработки FLProg, ориентированная на использование МК на ядре AVR.

Планируемые результаты обучения

При выполнении лабораторного цикла формируются компетенции ПК-6 и ПК-7.

Обучающийся должен

знать:

- основные характеристики микропроцессоров / микроконтроллеров, микропроцессорных систем, программно-технических комплексов (ПТК);
- особенности построения комплексов сбора информации с датчиков, передачи информации по интерфейсам, способы обработки и хранения;
- УГО электронных компонентов; функции электронных компонентов; основы построения принципиальных и монтажных электрические схем

уметь:

- проводить расчеты основных параметров ПТК;
- проектировать узлы и модули ПТК малой и средней сложности;
- анализировать архитектуру ПТК малой и средней сложности;
- разрабатывать программное обеспечение для микроконтроллеров, в том числе с использованием специализированных систем;
- осуществлять монтаж и демонтаж ПТК малой и средней сложности.

владеть:

- методами выбора элементной базы для многоуровневых систем на базе микро-ЭВМ и персональных компьютеров; методами построения многоуровневых систем на базе микро-ЭВМ и персональных компьютеров;
- методами построения микропроцессорных систем, сопрягаемых с датчиками различных типов, в том числе в специализированных системах;
- навыками чтения и изображения электронных схем на основе современной элементной базы; навыками моделирования узлов ПТК с целью анализа и оптимизации их параметров с использованием традиционных средств исследования, включая стандартные пакеты прикладных программ

Оборудование и программное обеспечение

- ПК (Processor i5-2500, RAM DDR3 4 GB, HDD 320 GB, DVD RW, TFT-монитор 24” 1920x1080);

- Плата ARDUINO UNO/FREEDUINO с МК ATmega168/328;
- Многофункциональная плата расширения XD203;
- Среда разработки Portable FLProg 8.0.2;
- ARDUINO IDE-1.8.19.

Правила техники безопасности при проведении лабораторных работ

Общие требования безопасности

1. К работе с электроизмерительными приборами, электроустановками, ЭВМ под руководством преподавателя или ответственного за лабораторию допускаются лица, прошедшие инструктаж по охране труда, медицинский осмотр и не имеющие противопоказаний по состоянию здоровья.
2. При работе в лаборатории студенты должны соблюдать правила поведения, расписание учебных занятий, установленные режимы труда и отдыха.
3. При работе с электроизмерительными приборами возможно воздействие на работающих следующих опасных факторов: а) поражение электрическим током при прикосновении к оголенным проводам и при работе с приборами, находящимися под напряжением;
б) травмирование рук при использовании неисправного инструмента.
4. При несчастном случае пострадавший или очевидец несчастного случая обязан немедленно поставить в известность преподавателя или зав. лабораторией, который сообщает об этом администрации ПГТУ. При неисправности электроизмерительных приборов, инструмента следует прекратить работу и сообщить об этом преподавателю или зав. лабораторией.
5. Студенты, допустившие невыполнение или нарушение инструкции по охране труда, привлекаются к дисциплинарной ответственности в соответствии с правилами внутреннего трудового распорядка университета и подвергаются внеочередной проверке знаний правил техники безопасности.

Требования безопасности перед началом работ

1. Получив разрешение на проведение лабораторных работ, ПРОВЕРЬТЕ состояние и исправность приборов и кабелей, наличие и исправность защитного заземления.
2. Подготовьте необходимые для работы материалы, приспособления и разложите на свои места, уберите с рабочего стола все лишнее.

Требования безопасности во время работы

1. ПОМНИТЕ! Электрический ток величиной 0,1 А и напряжением свыше 42 В опасен для жизни человека.
2. Лабораторные работы студенты проводят только в присутствии преподавателя или ответственного за лабораторию.
3. Включение ПК осуществлять кнопкой включения на системном блоке.
4. При сборке электрической схемы использовать провода без видимых повреждений изоляции, избегать пересечений и скручиваний проводов, питание схемы подключать в последнюю очередь.
5. Все изменения в схеме проводить при отключённом от компьютера USB-кабеле.
6. Не допускать попадания влаги на элементы схемы и компьютера.
7. Наличие напряжения в электрической цепи проверять только приборами.
8. Не допускать предельных нагрузок измерительных приборов.
9. Не оставлять без надзора невыключенные электрические устройства и приборы.

Требования безопасности по окончании работы

1. Выключить ПК программно завершением работы.
2. Отключить электроизмерительные приборы и лабораторные установки от электросети / компьютера.
3. Привести в порядок рабочее место.

4. Сообщить преподавателю или ответственному за лабораторию об окончании работы и получить разрешение на уход из лаборатории.

1. Среда разработки FLProg и программирование ПЛК на языке релейных диаграмм LD

1.1 Цель работы

Ознакомление с аппаратными средствами стенда и изучение особенностей разработки проектов на языке релейных диаграмм в среде FLProg.

1.2 ПЛК на основе платформы ARDUINO

Ведущим производителем ПЛК считается крупнейшая немецкая инженеринговая компания Siemens (30-40% рынка). К другим крупным производителям относятся компании Rockwell Automation/ Allen Bradley и Honeywell Process - бизнес-подразделение Honeywell International (США), японские Mitsubishi Electric, Hitachi, Ltd, Omron, Koyo Electronics Industries, IDEC, Yokogawa Electric, французская Schneider Electric, Швейцарско-шведский промышленный гигант ABB, Тайваньская Delta, немецкая Phoenix Contact – относительно небольшая компания приносит доход около 2.3 миллиарда евро в год [1].

С зарубежными производителями в разных классах контроллерных средств конкурирует большое число российских предприятий: «Автоматика», ВЕЕА, «Волмаг», ДЭП, Завод электроники и механики, «ЗЭИМ Инжиниринг», «Интеравтоматика», «НВТ Автоматика», ОВЕН, ПИК «ЗЕБРА», РИУС, «Реалтайм», «Системотехника», «Трей», «Эмикон», «Импульс», «Инсист Автоматика», «Интеравтоматика», «Квантор», НИИтеплоприбор, «НВТ-Автоматика», ПИК «Прогресс», «Саргон», «Системотехника», ТЕКОН, «Электромеханика», ЭМИКОН и др. [2].

Используемые в ПЛК микроконтроллеры имеют разрядность от 8 до 32 и рассчитаны на широкий диапазон решаемых задач. Схожие черты ПЛК – это развитые средства ввода вывода аналоговых и цифровых сигналов, устойчивость к промышленным помехам, высокая надёжность, архитектура и технология программирования часто отвечающие стандарту МЭК 61131-3.

При этом ПЛК многих фирм имеют особенности, приводящие к неполной совместимости оборудования и технологий программирования. Вместе с тем всё чаще к ПЛК относят промышленные контроллеры поддерживающие стандартные языки программирования. Крупные производители поставляют ПЛК вместе с собственными системами разработки, при этом большинство их далеко небесплатны.

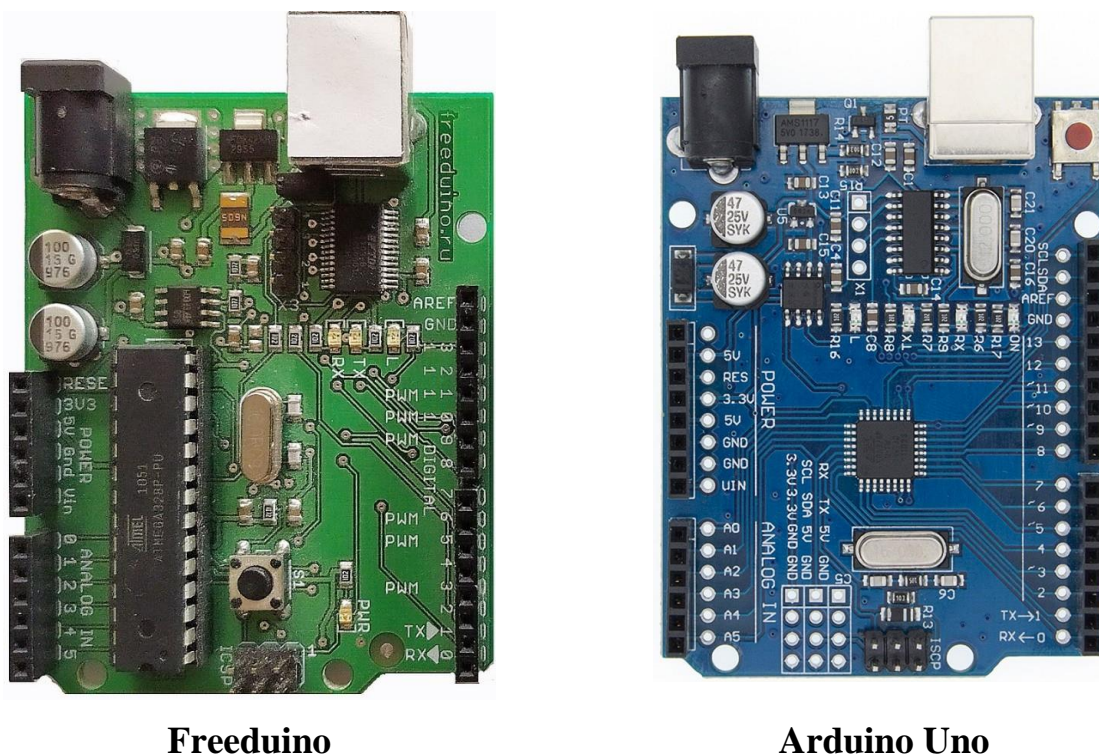
Компания CONELCOM GmbH (Австрия) производит ПЛК CONTROLLINO – ПЛК с открытым исходным кодом, полностью совместимый с платформой Arduino. Среди клиентов, которые уже используют Controllino, такие компании как Siemens, Sony, Bosch, Volkswagen, Dyson, Mattel, Mercedes-Benz, Массачусетский Технологический Институт, Virgin Galactic. Достоинством системы является её бесплатность, высокий уровень поддержки сообществом. При этом проектирование систем на основе данной платформы может осуществляться при поддержке нескольких систем программирования.

Следует учитывать, что обычные платы Arduino не поддерживают требования, предъявляемые к промышленным контроллерам по ряду параметров, но вполне подходят для освоения технологии построения управляющих систем на языках стандарта МЭК 61131-3. В этом контексте платформа Arduino подходит для обучения программированию ПЛК. В качестве среды проектирования выбрана программ FLProg, поддерживающая языки LD и FBD [3].

Лабораторные работы ориентированы на использование контроллеров Freeduino и модернизированной версии Arduino Uno (Китай), в которой вместо ATmega8U2 используется конвертор CH340G. Помимо них используются многофункциональные платы XD203.

1.3 Устройство и назначение выводов контроллеров

Внешний вид используемых контроллеров представлен на рисунке 1.



Freeduino

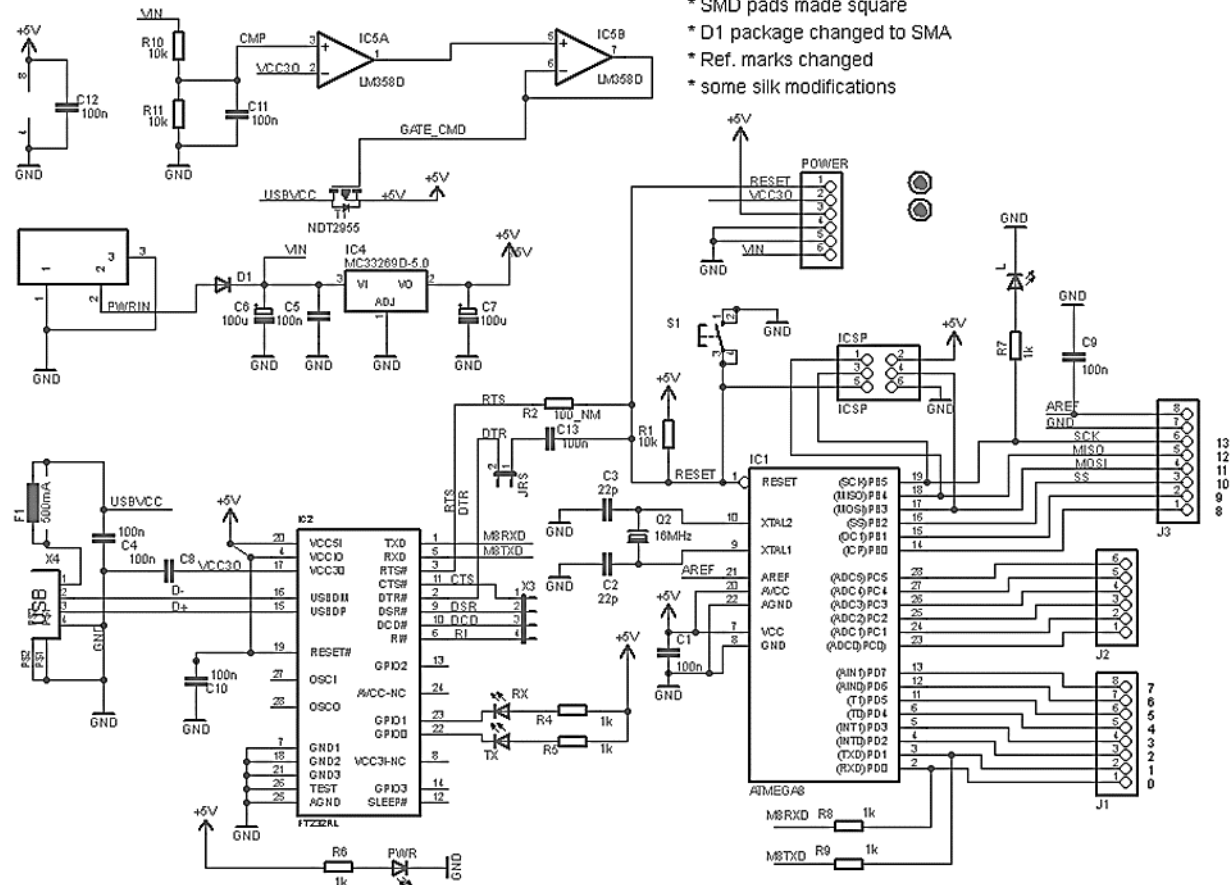
Arduino Uno

Рисунок 1 – Внешний вид контроллеров Arduino

Контроллеры отличаются несущественными деталями: типом корпуса установленного микроконтроллера ATmega328, конверторами UART-USB, местами размещения кнопок сброса. Каждая плата содержит микроконтроллер, кварцевый резонатор 16 МГц, стабилизатор напряжения с 5 В выходом, схему обнаружения внешнего напряжения питания 9-12 В, конвертор FTDI232 (Freeduino) или CH340G (Arduino Uno), разъём для подключения внешнего источника питания, разъём USB (тип B), краевые разъёмы для сопряжения с периферией, несколько светодиодов (индикации питания, сигналов последовательного интерфейса UART, и управляемый пользовательской программой). Интерфейс USB при подключении к ПК обеспечивает питание платы, загрузку программ и поддержку обмена данными с ПК.

На рисунке 2 приведена схема платы контроллера (Freeduino).

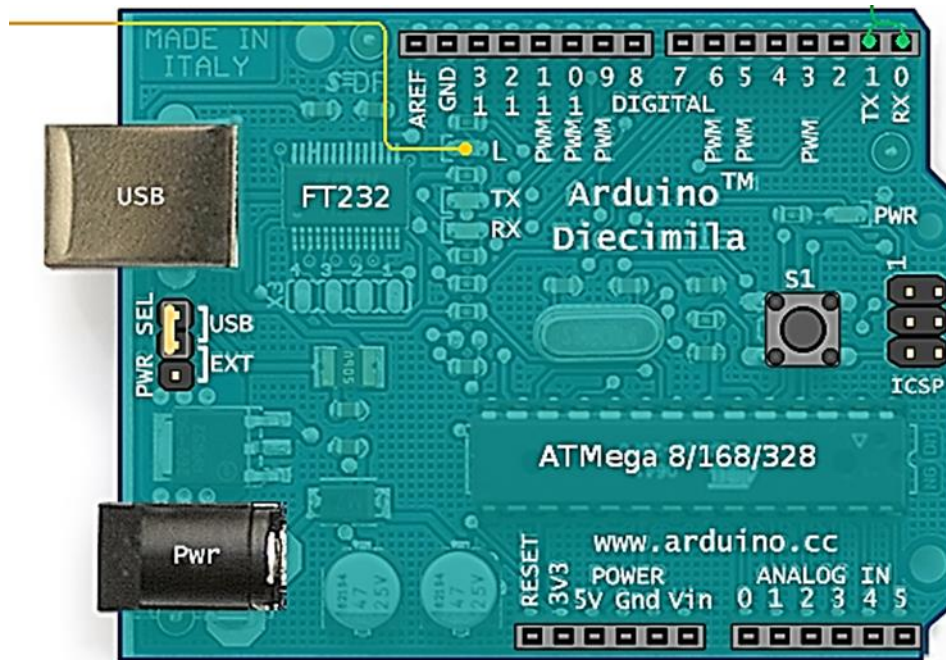
Freeduino 2009 (Arduino 2009 based). Small modifications by A. Nesterovsky:
 Distributed under a Creative Commons license Attribution-ShareAlike 2.5



- * RESET solder jumper replaced with ordinal jumper
- * SMD pads made square
- * D1 package changed to SMA
- * Ref. marks changed
- * some silk modifications

Рисунок 2 – Схема платы контроллера

Arduino – это развитая экосистема, включающая аппаратные и программные средства создания проектов, единообразность применения при варьировании типов микроконтроллеров в рамках ядра AVR [4]. Популярность её привела к тому, что на сегодня поддерживается множество микроконтроллеров с существенно различной архитектурой. Поскольку платформа открытая, то на рынке представлены похожие платы многих производителей. Так как Arduino является торговой маркой, то названия выпускаемых плат отличаются от оригинала (Freeduino, Japanino, Freetronics Eleven, Seeduino, CraftDuino, Diavolino). Прототипом для перечисленных плат послужила оригинальная плата стандартного размера Diecimila. Интерфейс стандартных плат приведен на рисунке 3.



Atmega168 Pin Mapping

Arduino function	Microcontroller Pin	Microcontroller Pin	Arduino function
reset	(PCINT14/RESET) PC6	1	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	GND
GND	GND	8	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/CP1) PB0	14	PB1 (OC1A/PCINT1) digital pin 9 (PWM)
		15	

Рисунок 3 – Назначение выводов стандартной платы

Стандартные платы характерны тем, что для них выпускается большое количество плат расширения, устанавливаемых поверх платы микроконтроллера (т.н. шильды).

Линейка оригинальных плат различается применяемым МК, формфактором, некоторыми отличиями в схемотехнике.

Цифровые выводы (digital pin) могут использоваться как цифровые входы или выходы (программируется), выводы последовательных интерфейсов, или

выходы широтно-модулированных импульсов. Аналоговые входы (analog input) используются как мультиплексируемые входы аналого-цифрового преобразователя. Альтернативные функции выводов представлены на рис. 3.

1.4 Многофункциональная плата расширения

Многофункциональная плата (шилด์) содержит три пользовательских кнопки, 4 светодиода, 4-х разрядный индикатор, биппер, потенциометр, позволяющий регулировать напряжение на аналоговом входе 0, цанговые разъёмы для подключения трёхвыводных аналоговых и цифровых датчиков. Внешний вид платы приведен на рисунке 4, а схема на рисунке 5.

Штыревые разъёмы на нижней стороне платы совместимы с разъёмами на плате Arduino.

Управление 4-х разрядным семисегментным светодиодным индикатором (U1) реализовано на двух сдвиговых регистров со встроенным буфером (U2, U3). Младшие 4 бита регистра U2 определяют активный (светящийся) разряд, а 8 бит регистра U3 определяют символ и разделитель.

Данные на вход 16 битового регистра подаются старшим разрядом вперёд с вывода 8 платы контроллера. При этом сначала подаётся код символа, затем унитарный код разряда с нулевой старшей тетрадой. Каждый бит сопровождается тактовым импульсом на выводе 7 платы контроллера. По завершении передачи всех 16 бит формируется импульс перезаписи нового кода в буферные регистры на выводе 4 платы контроллера.

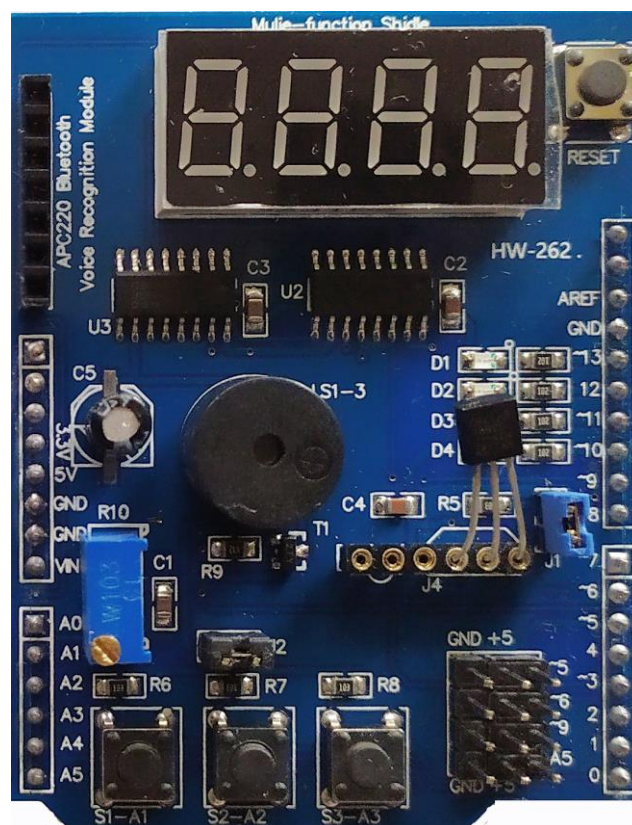


Рисунок 4 – Многофункциональная плата расширения

Такой способ управления исключает мигание разрядов при загрузке нового отображаемого кода.

Поскольку в каждый момент времени только один разряд индикатора является активным, то для индикации 4-х разрядных значений используется динамический режим управления (поочерёдный вывод символов и кода активного разряда с заданным временным интервалом).

Три пользовательские кнопки S1-S3 подключены к аналоговым входам A1-A3 платы контроллера. Нажатие на кнопку генерирует уровень логического нуля. Уровень логической единицы формируется подтягивающими резисторами R6-R8. Четыре одиночных светодиода D1-D4 активируются низким уровнем на выводах 10-13 платы контроллера.

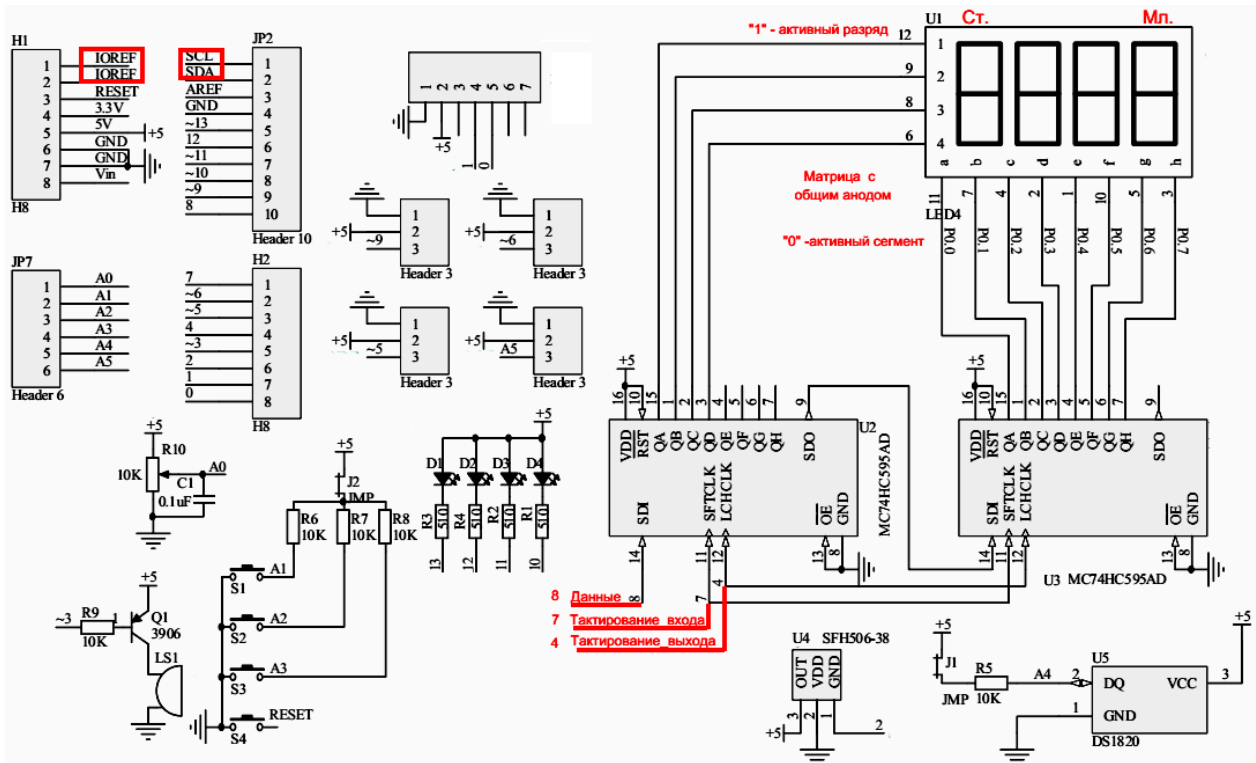


Рисунок 5 – Схема многофункциональной платы

Потенциометр задаёт напряжение на аналоговом входе A0. Аналоговый вход A4 может использоваться при поддержке интерфейса On-Wire при подключении цифрового датчика температуры DS1820, или как аналоговый вход, для чего необходимо снять перемычку J1.

1.5 Программирование контроллеров на языке релейных диаграмм LD

Свободно распространяемая программа поддержки программирования FLProg используется в портативной версии (номер текущей версии 8.0.2). Запускаемый файл FLProg.exe.

Основное окно программы стандартно. Главное меню содержит пункты: «Файл», «Вид», «Проект», «Инструменты», «Настройки», «Помощь». Для наиболее часто применяемых операций используются иконки. В меню файл можно открыть или создать новый проект.

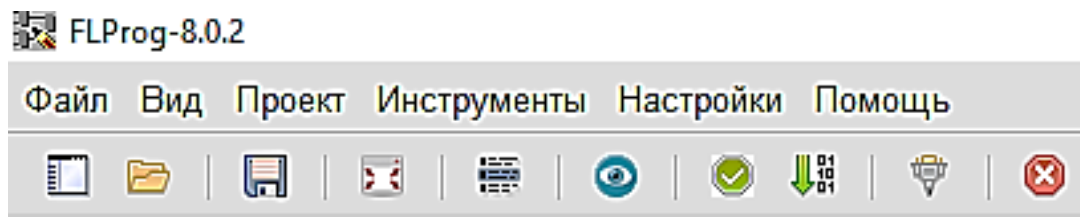


Рисунок 6 – Главное меню FLProg; слева направо иконки:

создать новый проект; открыть проект; сохранить проект; развернуть схему; показать список Undo; просмотр кода проекта; проверить проект; компилировать проект; открыть монитор коммуникационного порта; закрыть

При открытии нового проекта выбирается один из двух предлагаемых языков. В данной работе это язык релейных (лестничных) диаграмм – Ladder Diagram (LD, LAD, РКС). Синтаксис языка удобен для замены логических схем, выполненных на релейной технике. Язык ориентирован на специалистов по автоматизации, работающих на промышленных предприятиях. Обеспечивает наглядный интерфейс логики работы контроллера, облегчающий не только задачи собственно программирования и ввода в эксплуатацию, но и быстрый поиск неполадок в подключаемом к контроллеру оборудовании.

Программа на языке релейной логики имеет наглядный и интуитивно понятный инженерам-электрикам графический интерфейс, представляющий логические операции, как электрическую цепь с замкнутыми и разомкнутыми контактами. Протекание или отсутствие тока в этой цепи соответствует результату логической операции (истина – если ток течет; ложь – если ток не течет). Основными элементами языка являются контакты, которые можно образно уподобить паре контактов реле или кнопки. Пара контактов отождествляется с логической переменной, а состояние этой пары — со значением переменной. Различаются нормально замкнутые и нормально разомкнутые контактные элементы, которые можно сопоставить с нормально замкнутыми и нормально разомкнутыми кнопками в электрических цепях.

Дальнейшее знакомство с программой и языком будем осуществлять в ходе проектирования простейшей схемы включения электромагнитного пускателя. Подобный пускатель снабжается двумя кнопками «ПУСК» и «СТОП». При нажатии на кнопку «ПУСК» на обмотку пускателя подаётся ток, магнитное поле притягивает якорь, воздействующий на группу силовых и вспомогательных контактов. Силовые контакты подключают к сети мощную нагрузку. Вспомогательные используются в схемах автоматики для индикации и блокировки (рисунок 7). Логическая схема управления пускателем может быть реализована ПЛК. В примере аварийная ситуация будет имитироваться кнопкой «АВАРИЯ»

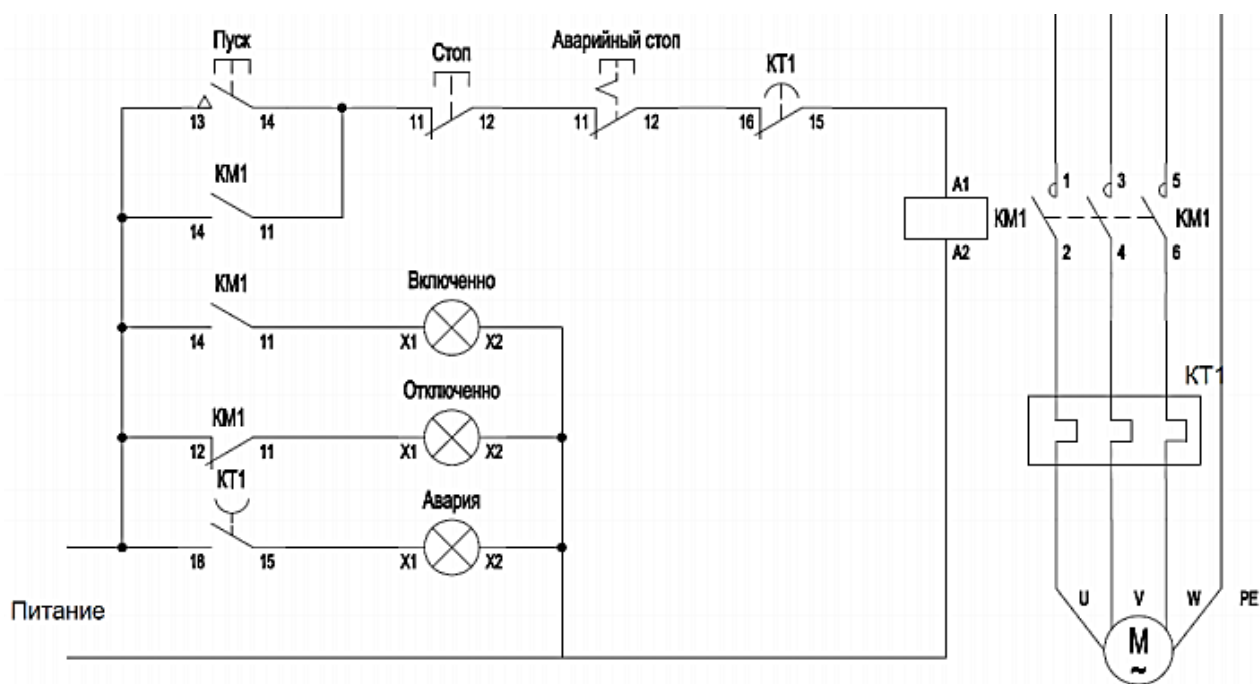


Рисунок 7 – Электромагнитный пускатель

Создадим новый проект на языке LD. Для этого в меню файл выберем «Новый проект для контроллера» и язык LD (LAD) и контроллер ARDUINO на МК ATmega328 – «Diecimila» или (Uno) в зависимости от используемой платы. Подтверждаем выбор в каждом открытом окне. При этом окно автоматически

закрывается. В левой части основного окна появляется дерево проекта, под основным меню – строка быстрых вызовов, в центре область схемы.

В правой части размещается библиотека блоков оборудования, которое можно применять в проекте. Над областью схемы размещается окно установленного оборудования. Его, левое и правое окно можно свернуть, или развернуть, а также изменить размеры. Подобные принципы построения свойственны большинству программ под операционной системой Windows и достаточно привычны.

Проект в FLProg представляет собой набор плат, на каждой из которых собран законченный модуль общей схемы. Для удобства работы каждая плата имеет наименование и комментарии. Так же каждую плату можно свернуть (для экономии места в рабочей зоне, когда работа над ней закончена) и развернуть. Красный индикатор в наименовании платы указывает на то, что в схеме платы есть ошибки. Платы следуют одна за одной сверху вниз. Каждая плата представлена шинами (слева она сплошная, а справа в FLProg представлена заземлением) и скоммутированным оборудованием между ними.

Проект заполняется начиная с первой платы. На первом шаге построения схемы в окне установленного оборудования выберем свободные цифровые входы-выходы контроллера. В проекте используется многофункциональная плата расширения, на которой кнопки подключены к выводам А1- А3. Назначаем функции кнопкам слева направо ПУСК (S1-A1), СТОП (S2-A2), АВАРИЯ (S3-A3). Находим в перечне свободных цифровых выводов указанные выводы и перетаскиваем их в область схемы (рисунок 8). Для выбранных выводов в таблице появятся дополнительные параметры - «Защита от дребезга» и «Подтягивающий резистор» (рисунок 9). В соответствии со схемой многофункциональной платы кнопки подключены к земле и установлен внешний подтягивающий резистор, поэтому во включении внутреннего подтягивающего резистора необходимости нет.

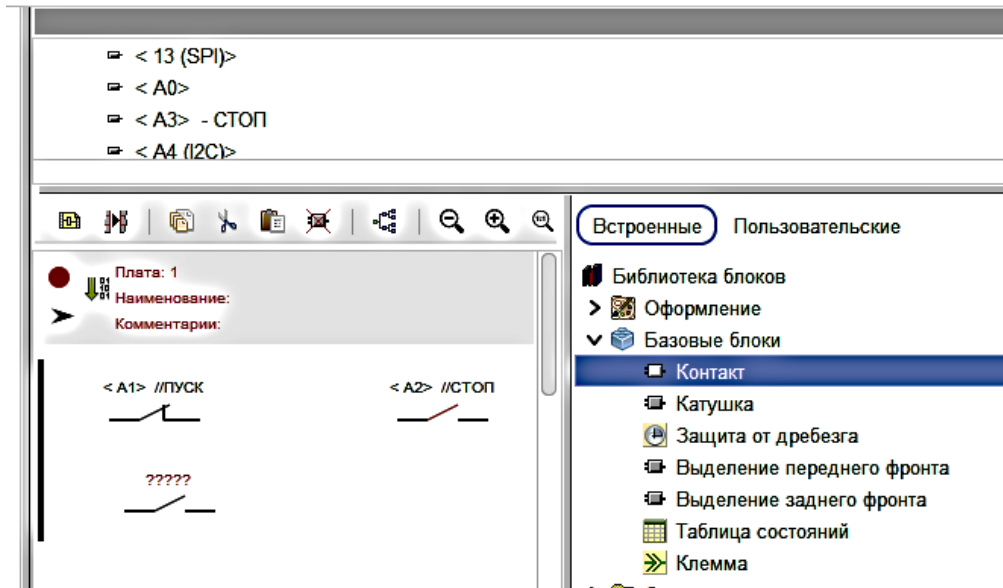


Рисунок 8 – Размещение на схеме контактов кнопок

Не нужна также и защита от дребезга. В графе «Комментарий» полезно указать название контакта, ассоциированное с выполняемой им функцией. Двойным щелчком по установленному оборудованию можно вызвать его свойства и изменить их.

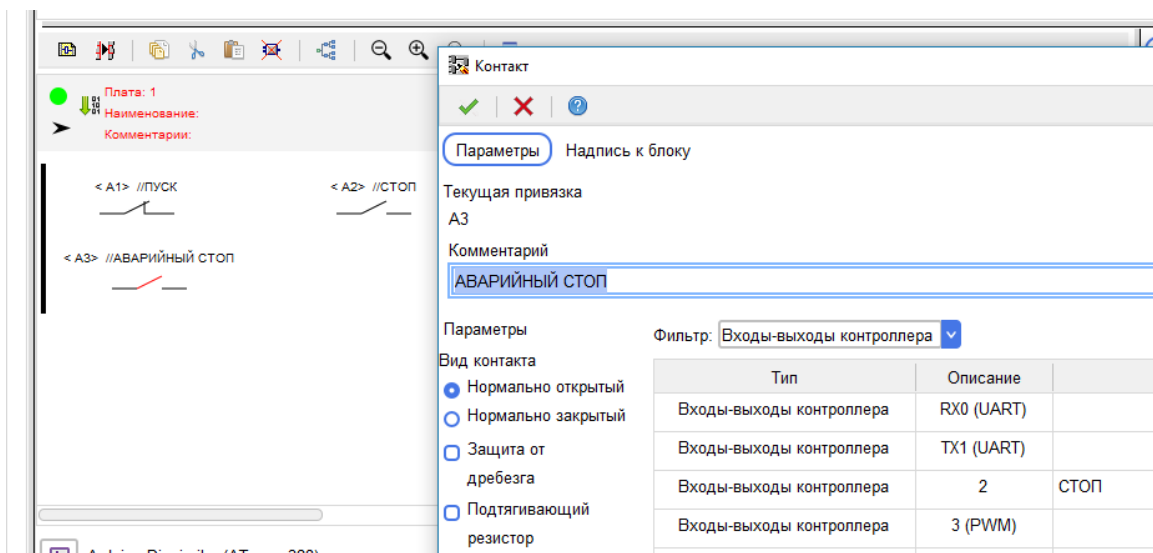


Рисунок 9 – Привязка контактов и выбор свойств

Далее из библиотеки блоков выбираем катушку. Двойным щелчком открываем её свойства и назначаем промежуточным реле К1.

На следующем шаге необходимо установить соответствие между уровнями на входе контроллера и состоянием контактов, поскольку в FLProg замкнутому контакту соответствует состояние 1, разомкнутому 0, а кнопки платы при замыкании генерируют 0. По этой причине следует инвертировать состояние контактов ПУСК, сделав контакты нормально замкнутыми. Контакты СТОП и АВАРИЯ нормально разомкнуты. Если кнопка СТОП не нажата, то на входе контроллера присутствует состояние 1, что эквивалентно замкнутой цепи промежуточного реле. Такая же ситуация имеет место и с кнопкой АВАРИЯ, т.е. при отсутствии аварийного состояния (кнопка не нажата) на соответствующем входе присутствует единица, и цепь замкнута. При нажатии кнопки авария на входе генерируется 0, а цепь питания промежуточного реле разрывается.

Соединяем элементы схемы наведением курсора на контакт и удерживая нажатой левую кнопку мыши перемещаем к следующему контакту.

Поскольку после нажатия на кнопку ПУСК она будет отпущена, то для удержания пускателя в положении «включено» необходимо использовать автоблокировку. Для этого в окне установленного оборудования находим «Промежуточное реле» и перетаскиваем на схему его контакты (рисунок 10).

Подключаем контакты кнопки «ПУСК» к левой шине платы, а контактами промежуточного реле шунтируем контакты кнопки ПУСК. Дадём название плате 1 «Управление». Для этого выполняем двойной щелчок на её заголовке и вводим название. Зелёный кружок в заголовке означает, что схема корректна и не использует ошибочных блоков.

Следующая плата (2) должна управлять выводами контроллера. Она создаётся нажатием на первую иконку в меню окна схемы «Добавить плату». На эту плату устанавливаем 2-е катушки и привязываем их к выводам платы контроллера.

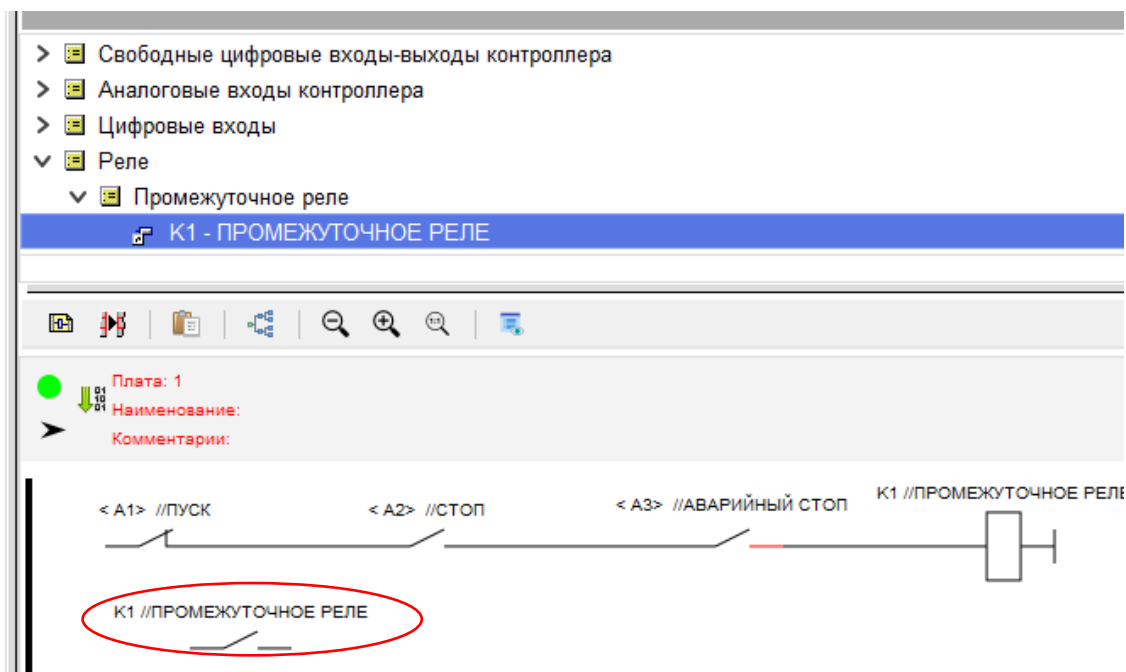


Рисунок 10 – Установка блокировочных контактов

При этом используем выводы 13-12, к которым подключены светодиоды, позволяющие индцировать состояния выводов. Назначим светодиод D1 индикатором включенного электромагнитного пускателя, а D2 –выключенного.

Для определения выводов, к которым подключены катушки двойным щелчком по катушке вызываем её свойства, выбираем вывод 13 двойным щелчком по нему и присваиваем название первой катушке (например, ВКЛ.). При включении катушки в проекте на выходе платы будет 5 В, при выключении – 0 В. Перетаскиваем из списка установленного оборудования две пары контактов промежуточного реле и каждой парой подаём напряжение с шины на соответствующую катушку. Следует помнить, что светодиоды многофункциональной платы светятся при низком уровне на выходах контроллера. Поэтому контакты реле, подключенные к катушке «ВКЛ.» должны быть нормально замкнутыми, а «ВЫКЛ» - нормально разомкнутыми. В этом случае при срабатывании промежуточного реле нормально замкнутые контакты разомкнутся, что эквивалентно обесточиванию цепи катушки «ВКЛ», т.е. установлению на выводе 13 логического нуля. Светодиод D1 подключенный к контакту 13 при этом будет светить.

Цепь катушки «ВЫКЛ» при сработавшем промежуточном реле будет замкнута, т.е. вывод 12 будет находится под напряжением высокого уровня, а светодиод D2 погаснет.

Схема проекта без аварийной сигнализации представлена на рисунке 11.

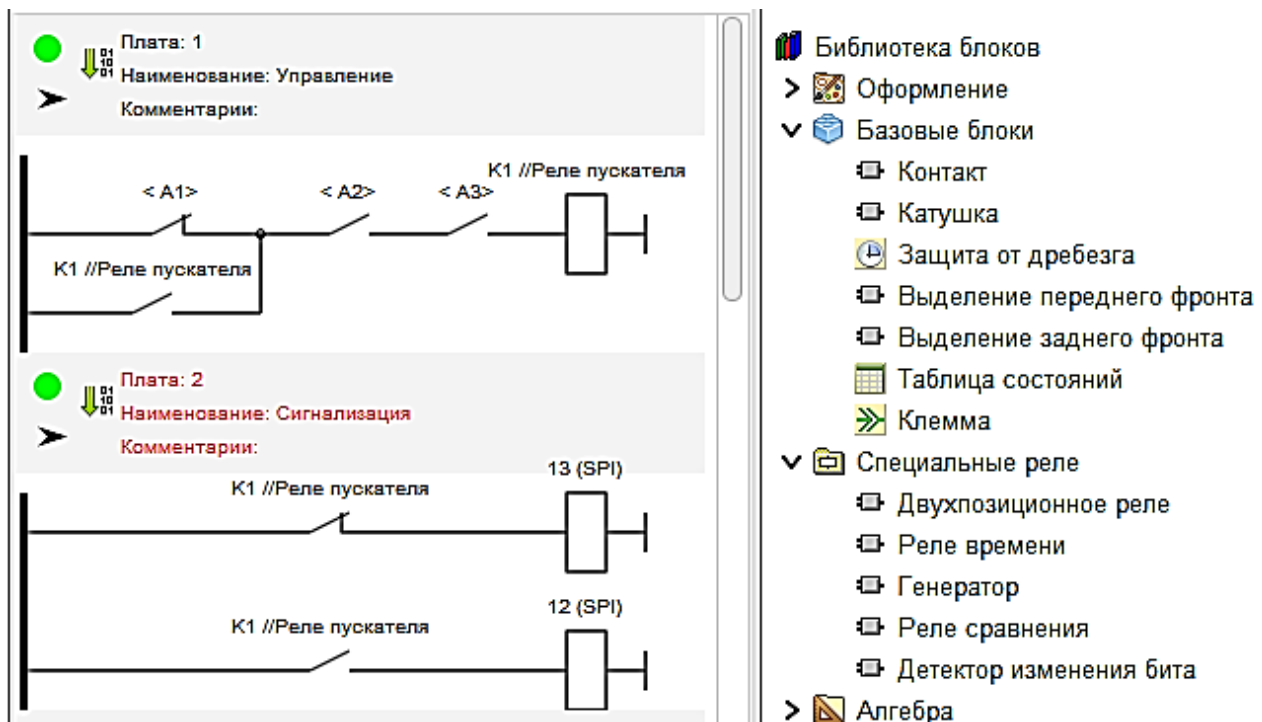


Рисунок 11 – Проект электромагнитного пускателя на ПЛК

Для экономии рабочего пространства и ускорения работы программы при больших проектах можно свернуть плату нажатием на стрелку в заголовке платы.

1.6 Компиляция и запись проекта в контроллер

После завершения проекта он компилируется в скетч ARDUINO через пункт главного меню программы или нажатием на иконку компиляции. При этом вызывается интегрированная среда разработки IDE ARDUINO. После загрузки среды со скетчем в главном меню оболочки «Инструменты» выбирается



коммуникационный порт и запускается компиляция и загрузка скетча в предварительно подключенную к порту USB персонального компьютера плату контроллера .

В меню «Инструменты» следует также убедиться в соответствии типа платы, указанного в FLProg и IDE ARDUINO.

1.7 Подготовка к работе

При домашней подготовке к работе следует тщательно изучить схему контроллера, возможности и устройство многофункциональной платы расширения и схемы (рис. 4-5) подключения на ней кнопок, светодиодов и узла звукового оповещения, схему электромагнитного пускателя (рис. 7) и схему логического устройства, реализующего базовые функции пускателя (рис. 11). Следует учитывать, что управление силовыми цепями (не показаны) осуществляется дополнительным контактом промежуточного реле, к которому подключаются катушка с назначенным выводом или используется вывод индикации «ВКЛ» или «ВЫКЛ», в зависимости от требуемого уровня напряжения для управления коммутатором силовой цепи. В качестве коммутатора силовой цепи могут использоваться твердотельные реле с встроенной гальванической развязкой, или упрощённые электромагнитные пускатели, не содержащие промежуточных реле и его контактной системы.

Изучить порядок и особенности создания схемы электромагнитного пускателя в среде разработки FLProg с использованием платы ARDUINO UNO и многофункциональной платы расширения.

Для самоконтроля письменно ответить на вопросы (п. 1.8).

1.8. Вопросы для самоконтроля

1. Каким образом объединяются в систему плата микроконтроллера и плата расширения?

2. Какие уровни формируются на выходе кнопок S1-S3 при нажатии?

3. Биппер на многофункциональной плате является активным устройством, содержащим встроенный генератор. При каком уровне на выводе МК он будет звучать?

4. Поясните по схеме включения светодиодов D1-D4 при каких уровнях на выходах МК они светятся.

5. Как в FLProg выбрать нужную плату микроконтроллера?

6. Как разместить на схеме платы контакты кнопки S1?

7. Для чего подключают цепи к левой шине и что эта шина символизирует?

8. Что является нагрузкой цепи, содержащей электрические контакты на схеме устройства?

9. С какой целью катушке реле на схеме ставится в соответствие контакт ПЛК?

10. Каким образом выполняется соединение элементов схемы?

1.9 Порядок выполнения работы

1. Включить компьютер запустить среду проектирования (файл FLProg.exe).

2. В соответствии со схемой проекта (рис. 11) последовательно собрать плату 1 «Управление». Убедиться в корректности выполненных соединений (зелёная метка у заголовка платы).

3. Добавить плату 2 «Сигнализация» и собрать её схему. Убедиться в корректности схемы.

4. Подключить многофункциональную плату расширения к плате микроконтроллера (при необходимости). Следить за правильностью ориентации плат относительно друг друга и отсутствием замыкания контактов платы расширения на разъём USB платы микроконтроллера (штыри не должны быть полностью погружены в гнезда).

5. Подключить контроллер интерфейсным кабелем к USB разъёму компьютера.
6. Компилировать проект.
7. После компиляции проекта и открытия программы в IDE ARDUINO в пункте меню «Инструменты» проверить соответствие типа платы используемой версии и в пункте «Порт» выбрать номер порта платы щелчком мыши.
8. Загрузить скетч в плату МК.
9. Выполнить проверку алгоритма функционирования, нажимая поочередно кнопки ПУСК – СТОП, ПУСК – АВАРИЯ, ПУСК – СТОП – АВАРИЯ, ПУСК – АВАРИЯ – СТОП. Записать состояние индикаторов для каждой последовательности нажатий. Отметить соответствие или несоответствие алгоритма функционирования спроектированного устройства алгоритму функционирования автомата.
10. Дополнить схему средствами звукового оповещения об аварийном отключении в соответствии со схемой рис. 12.

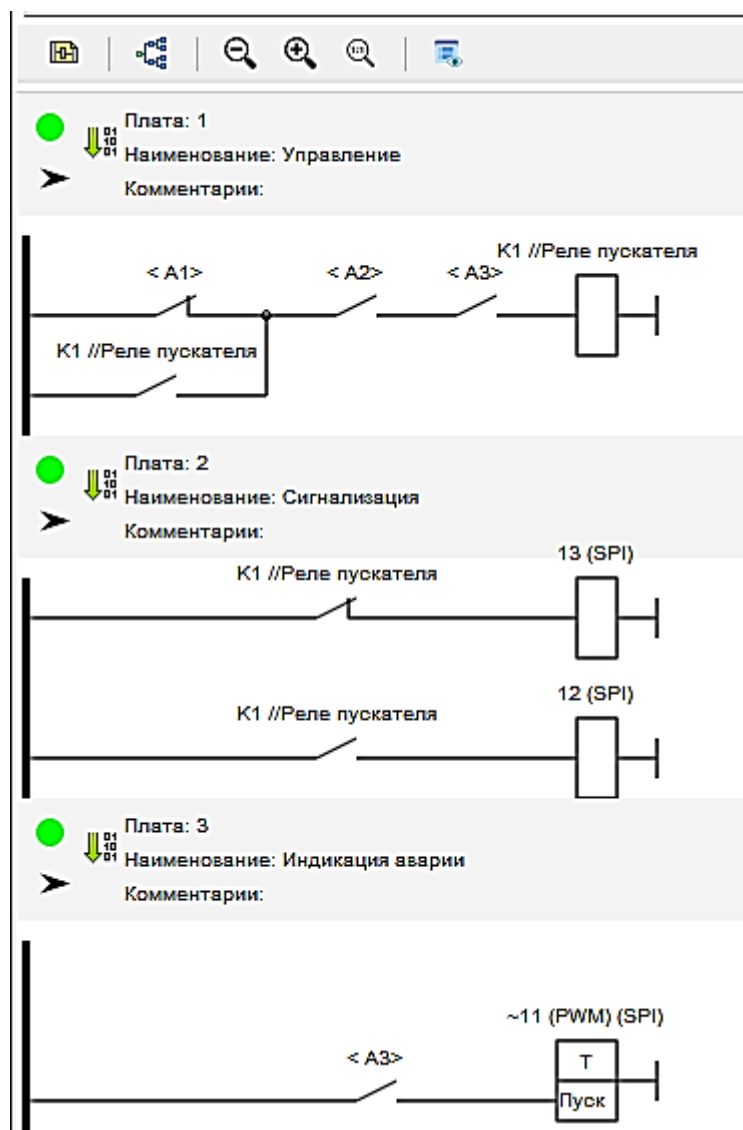


Рисунок 12 – Схема пускателя с аварийным автоматом

Для индикации аварийной ситуации следует использовать реле времени из раздела «Встроенные блоки/специальные реле». В свойствах выбирается контакт контроллера и тип задержки – на включение, что определяется тем, что затягивается время включения (низкий уровень выходного сигнала, при котором светится светодиод). Постоянная времени 2 с.

11. Исследовать работу пускателя в соответствии с п.9.

1.10 Содержание отчёта

Отчёт должен содержать:

1. Титульный лист;
2. Оглавление;
3. Наименование работы, цель исследований;
4. Электрическую схему электромагнитного пускателя и описание принципа функционирования;
5. Программы на языке LD пускателя без сигнализатора и с сигнализатором аварийного состояния;
6. Описание используемого оборудования ПЛК из библиотеки блоков и его параметров;
7. Ответы на вопросы для самоконтроля;
8. Описание поведения контроллера в двух вариантах реализации;
9. Ответы на контрольные вопросы;
10. Перечень литературы, использованной при подготовке и выполнении работы (ссылки на используемую техническую документацию).

1.11 Контрольные вопросы

1. Почему кнопке «ПУСК» ставится в соответствие нормально замкнутый контакт, а кнопкам «СТОП» и «АВАРИЯ» - нормально разомкнутые?
2. Какой логический уровень выходного сигнала ПЛК соответствует обмотке реле, находящейся под напряжением?
3. Укажите причину по которой бипер не подключен непосредственно к выводу 3 платы контроллера.
4. Оцените нагрузку по схеме принципиальной электрической многофункциональной платы величину нагрузки портов 10-13 МК.
5. Изучите схему платы контроллера и опишите, каким образом преодолевается конфликт внешнего источника питания с портом USB при одновременном их подключении.
6. Какой интерфейс микроконтроллера используется при загрузке в него программы работы ПЛК?

7. Какой интерфейс микроконтроллера используется для первоначальной загрузки программного обеспечения (в отсутствие прошитого загрузчика)? Для ответа на этот вопрос обратитесь к циклу лабораторных работ по дисциплине «Микропроцессорная техника».

2 Программирование ПЛК на языке функциональных блокковых диаграмм

2.1 Цель работы

Изучение особенностей разработки проектов на языке функциональных блокковых диаграмм в среде FLProg.

2.2 Особенности языка функциональных блокковых диаграмм

Язык функциональных блокковых диаграмм FBD (Function Block Diagram) – графический язык программирования стандарта МЭК 61131-3. Программа об­ра­зуется из списка цепей, выполняемых последовательно сверху вниз. При про­граммировании используются наборы библиотечных блоков. Блок (элемент) — это подпрограмма, функция или функциональный блок (И, ИЛИ, НЕ, триггеры, таймеры, счётчики, блоки обработки аналогового сигнала, математические опе­рации и др.). Каждая отдельная цепь представляет собой выражение, составлен­ное графически из отдельных элементов. К выходу блока подключается следу­ющий блок, образуя цепь. Внутри цепи блоки выполняются строго в порядке их соединения. Результат вычисления цепи записывается во внутреннюю перемен­ную либо подается на выход контроллера.

Язык программирования FBD или LAD выбирается при создании нового проекта для контроллера. После выбора языка FBD и микроконтроллера (ARDUINO/ATmega328/Diecimila, Duemilanove или Uno) откроется окно про­граммы с полями аналогичными полям при выборе языка LAD (рис.2.1).

Однако, здесь нет предварительно созданных входов. Их нужно создавать при необходимости. Для создания нового входа нужно нажать правую кнопку мыши (ПКМ) и в контекстном меню выбрать пункт «Вставить вход контроллера». Альтернативно, в поле тэгов выделить «Входы» и кликнуть по иконке «Добавить вход». Кроме того, имеются отличия в составе библиотеки блоков.

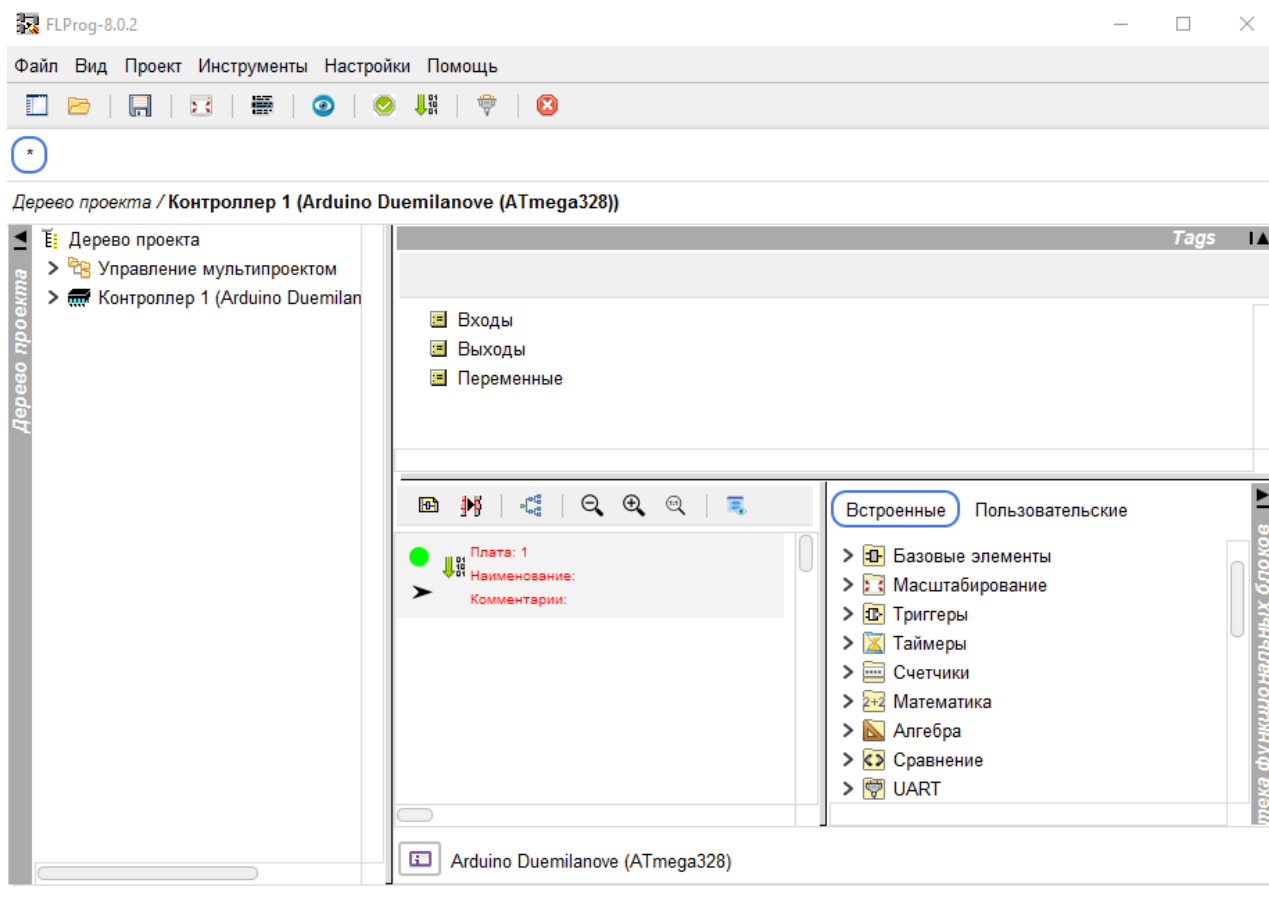


Рисунок 2.1 – Окно программы FLProg для программирования контроллера на языке FBD

Кратко рассмотрим эти блоки. Детально изучить их можно в видеоматериалах, размещённых на сервере кафедры.

2.2.1 Базовые элементы

Любой блок может быть перенесен на плату проекта перетаскиванием его из поля блоков или выбором с помощью контекстного меню при нажатии ПКМ.

Логические элементы ИЛИ, И, Исключающее ИЛИ (OR, AND, XOR).

Установив на плату логический элемент, щелчком по ПКМ можно вызвать его свойства и добавить вход. Так же с помощью ПКМ можно вставить вход или выход контроллера, инвертировать их сигналы. Для логического элемента XOR добавить вход невозможно.

Таблица состояний. Для реализации произвольных функций используется табличный метод их задания. С помощью одной таблицы можно реализовать одновременно несколько функций. Количество реализуемых функций определяется количеством выходов. Количество входов и выходов может быть изменено в свойствах таблицы, вызываемых двойным щелчком ЛКМ или в контекстном меню всплывающем при нажатии ПКМ (рис. 2.2, б). С помощью пункта «Добавить» вставляется строка значений входных переменных, для которой выходное значение равно единице. Выделив строку и вызвав пункт меню «Изменить», можно инвертировать любое значение любой переменной в строке щелчком ЛКМ по нему. Можно также удалить ненужную строку. Каждому выводу таблицы состояний соответствует своя таблица входов. Условное графическое обозначение для двух входов и трёх выводов приведено на рис. 2.2, а

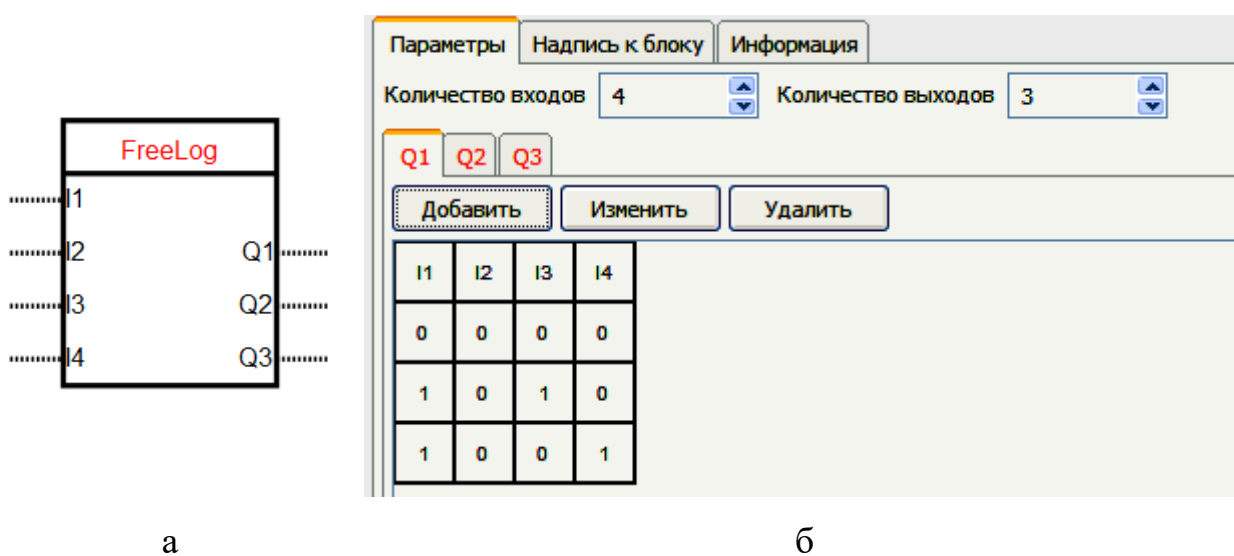


Рисунок 2.2 – Блок «таблица состояний» с 4 входами и 3 выходами: а – условное обозначение, б – свойства

Детектор изменения числа. Данный блок (2.3, а) реагирует на каждое изменение значения входной величины появлением импульса на выходе. В свойствах блока (рис. 2.3, б) можно задать верхнее и нижнее граничные значения приращения входной величины относительно предыдущего значения

(входного значения), вызвавшего срабатывание блока (гистерезис для прираще-ния). Если алгебраическая сумма приращений вверх и вниз не выходит за пре-делы гистерезиса, то выходные импульсы будут отсутствовать. При классиче-ском гистерезисе (задаются константы) срабатывание происходит при любом выходе за установленные граничные значения входной переменной, т.е. устрой-ство с гистерезисом реагирует на абсолютные значения переменной, а не её при-ращение.

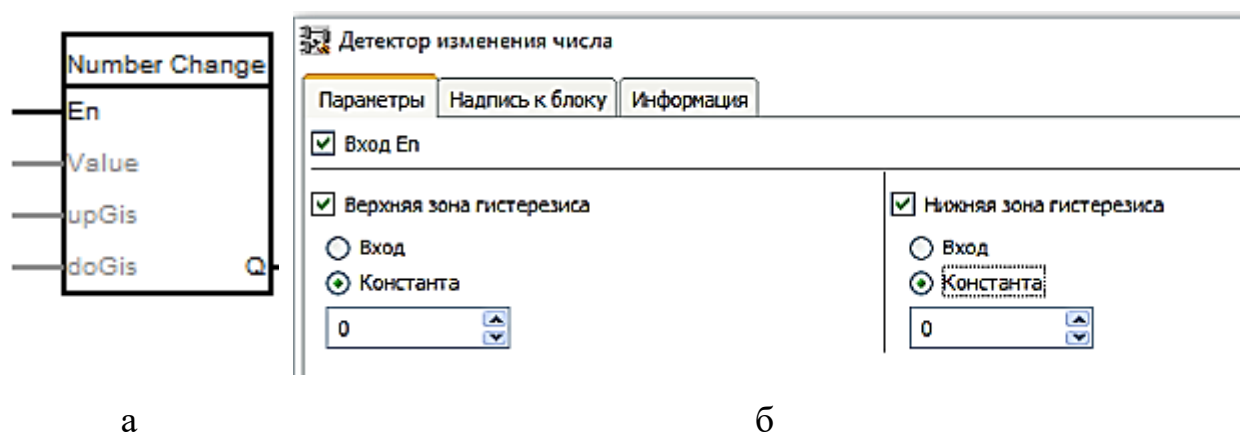


Рисунок 2.3 – Детектор изменения числа: а – условное обозначение,
б - свойства

Детекторы изменения строки и бита. Детектор изменения строки мо-жет быть настроен на изменение значения строки и/или изменение длины строки. При этом на его соответствующих выходах формируется импульс дли-тельностью в один цикл программы.

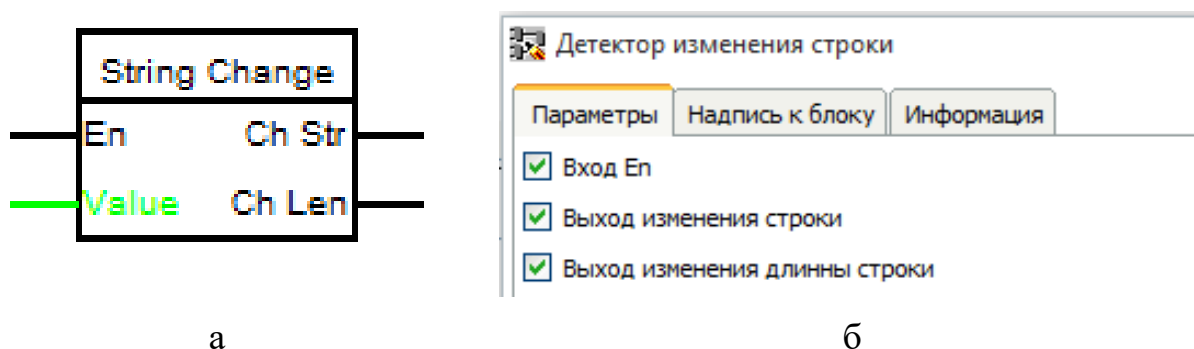


Рисунок 2.4 – Детектор изменения строки: а – условное обозначение,
б - свойства

Детектор изменения бита служит для обнаружения изменения состояния значения на входе с 1 на 0 и обратно. При изменении состояния на входе на выходе формируется импульс длительностью в один цикл исполнения программы (рис. 2.5)

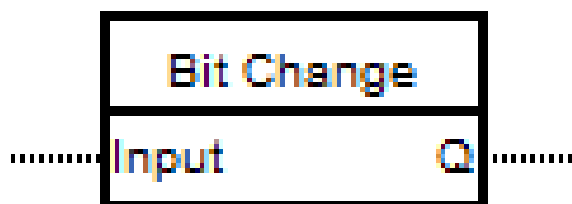


Рисунок 2.5 – Детектор изменения бита

Триггеры. Имеется несколько типов триггеров. Ни один из них не имеет параметров.

SR -триггер с приоритетом установки.

RS – триггер с приоритетом сброса.

Режим хранения этих двух триггеров $S = 0$, $R = 0$. Высокий уровень на входе S SR – триггера приводит к установке, а высокий уровень на входе R RS – триггера к сбросу, не зависимо от уровня альтернативного сигнала.

Rtrig – устройство выделения фронта. В момент перехода входного сигнала с низкого в высокий уровень на выходе Q устанавливается логическая единица на время одного цикла программы.

Ftrig – устройство выделения спада. В момент перехода входного сигнала с высокого в низкий уровень на выходе Q устанавливается логическая единица на время одного цикла программы.

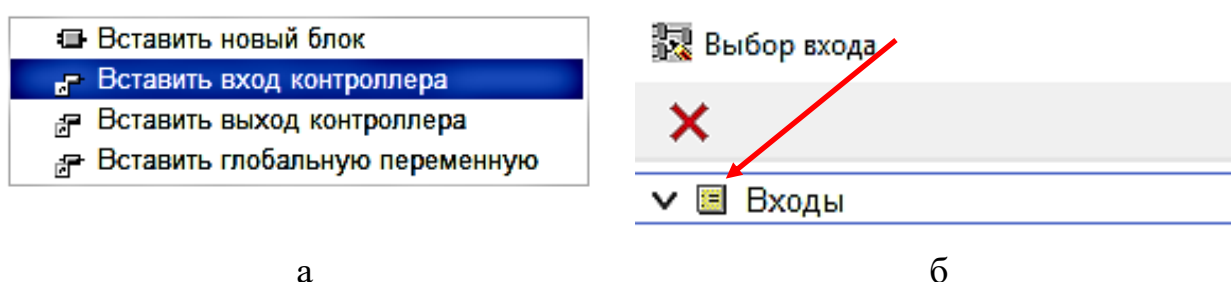
ТТ – счётный триггер, изменяющий своё состояние по фронту каждого импульса. Начальное состояние – 0.

Самый простой путь ознакомления с функциональными блоками – это перетащить выбранный блок на рабочее поле FLProg, открыть его свойства и выбрать справку (значок вопроса).

2.2.2 Программирование контроллера на языке FBD

Создадим схему управления электромагнитным пускателем аналогичную приведенной в работе №1. Первую плату назовём платой управления.

Определим назначение входов таким же образом, как и в проекте на языке LAD: ПУСК (S1-A1), СТОП (S2-A2), АВАРИЯ (S3-A3). Добавим вход «ПУСК» на плату 1. Для этого вызываем ПКМ контекстное меню, наводим на «Вставить вход контроллера» (рис. 2.6, а) и нажатием на ЛКМ открываем окно выбора входа. В нём выполняем щелчок на «Входы» (рис. 2.6, б).

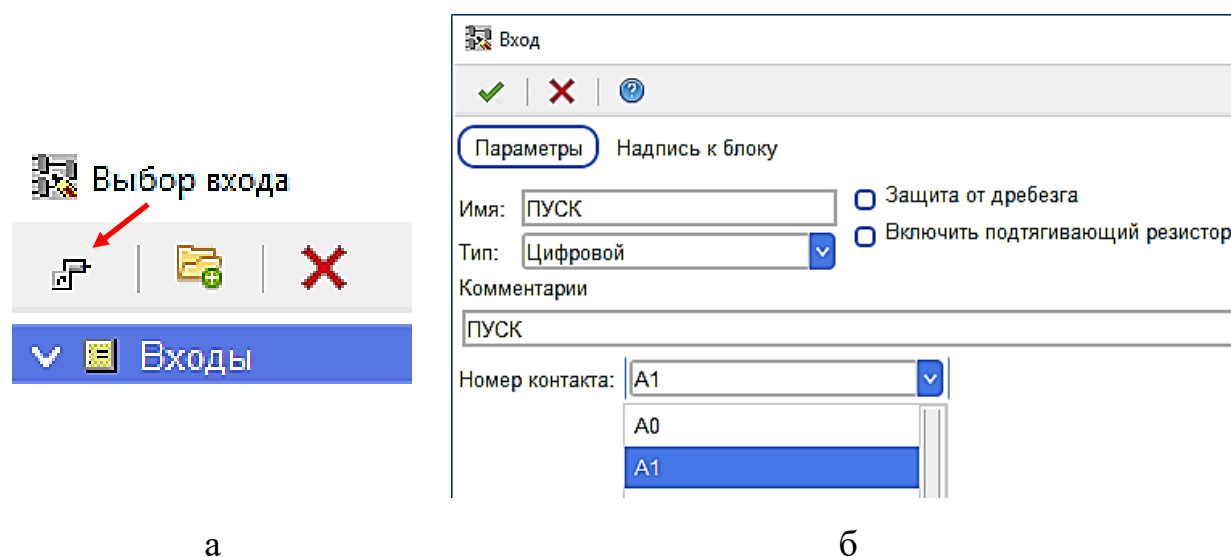


а

б

Рисунок 2.6 – Создание входов контроллера: а – всплывающее меню вставки, б – выбор входов

После этого открывается доступ к выбору входа (рис. 2.7,а), а при клике по иконке выбора и к параметрам входа.



а

б

Рисунок 2.7 – Назначение входов контроллера: а – доступ ко входам, б – выбор конкретного входа и его параметров

Состояние контактора идентифицируем состоянием RS – триггера. Выбор обусловлен тем, что триггер будет сброшен при любом состоянии входа установки S, если вход R = 1 и тем, что в качестве активного состояния выхода принимаем высокий уровень.

Выходной сигнал этого триггера используем в качестве переменной типа «булевой». Для этого в окне установленного оборудования кликом правой кнопки мыши по разделу «Переменные» выбираем «Добавить новую переменную». В окне параметров задаём имя, тип и начальное состояние.

Рисунок 2.8 – Параметры переменной

Нажав правую кнопку мыши на созданной переменной, перетаскиваем её на плату и соединяем выход триггера с её входом.

Состоянием триггера управляют кнопки. Включение пускателя осуществляется кнопкой «ПУСК». Предварительно щелчком ПКМ по входу S триггера выбираем пункт «инвертировать», т.к. кнопки при нажатии генерируют на выходе низкий уровень. Соединяем вход S триггера с кнопкой «ПУСК».

Перевод триггера в состояние «Сброс» осуществляется нажатием кнопки «СТОП» или по «АВАРИЯ». Обе кнопки генерируют 0 при нажатии. Наличие хотя бы одного нуля должно приводить к сбросу триггера. Нужная функция

реализуется логическим элементом «И-НЕ». Перетаскиваем блок «AND» на плату и соединяем его выход со входом R триггера. Вход R инвертируем. Кнопку «СТОП» и «Авария» соединяем со входами блока «AND». Схема платы управления представлена на рис. 2.9.

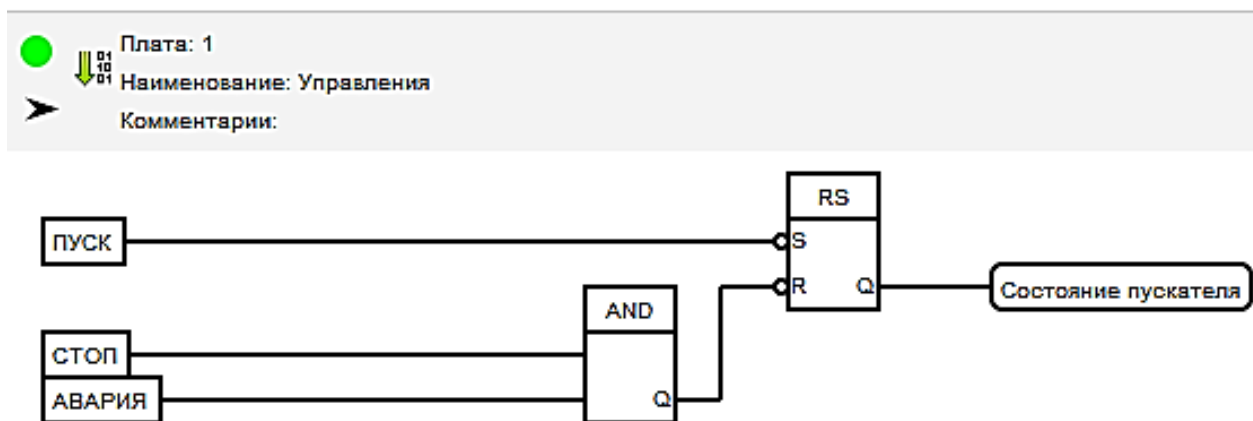


Рисунок 2.9 – Плата управления электромагнитного пускателя

Создаём плату 2 «Сигнализации» и добавляем на неё выходы контроллера 13, 12, 10, к которым подключены соответственно светодиоды D1 – «ВКЛЮЧЕНО», D2 – «ОТКЛЮЧЕНО», D4 – «АВАРИЯ» и 3 – звуковой сигнализатор аварийной ситуации.

Световой индикатор аварийной сигнализации включается при нажатии кнопки, имитирующей аварийную ситуацию, и выключается при нажатии кнопки «ПУСК». Сигнал для светодиода аварийной ситуации формируется RS-триггером. Сигнал установки триггера одновременно управляет звуковым сигналом и формируется с помощью таймеров и схемы «ИЛИ» из сигналов состояния пускателя и аварийной кнопки. Сигнал сброса поступает с кнопки «ПУСК». Поскольку активные уровни сигналов R и S триггера высокие, то сигналы с кнопки и схемы ИЛИ приходится инвертировать. Схема платы сигнализации представлена на рис. 2.10. Таймер ТОФ задерживает выключение сигнала. Интервал задержки следует выбирать в пределах 200 – 500 мс. Таймер ТОН задерживает включение (переход в высокий уровень) сигнала кнопки «АВАРИЯ».

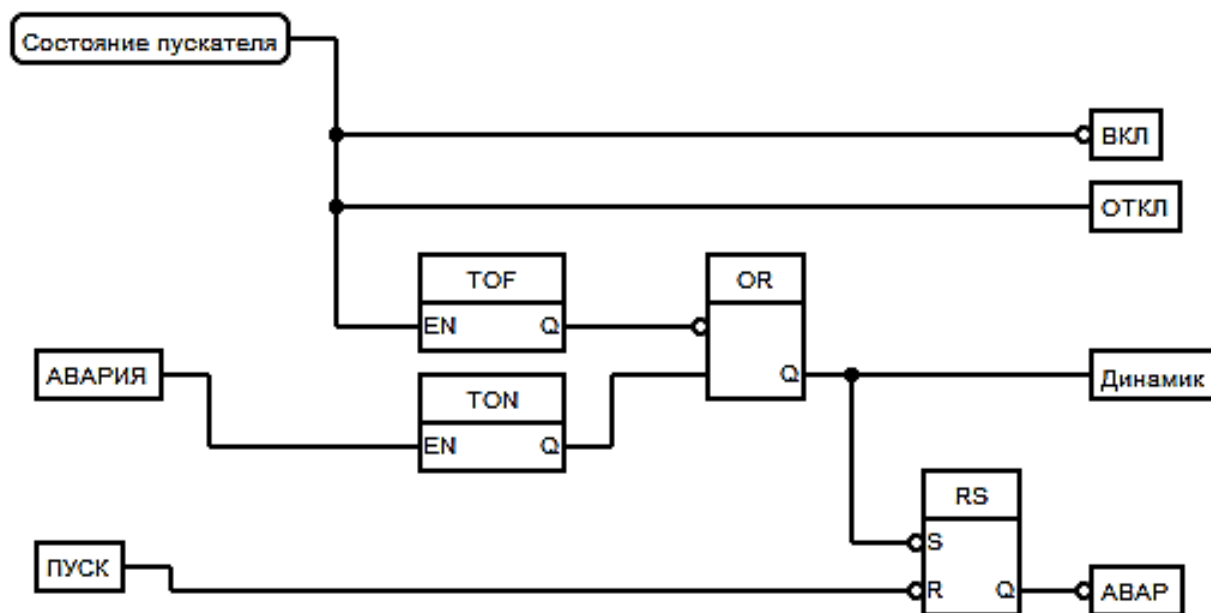



Рисунок 2.10 – Плата сигнализации электромагнитного пускателя

Интервал задержки определяет время звучания. При этом максимальное время звучания не может превысить интервал задержки таймера TOF.

Элементы индикации отображают режимы выключенного, включённого пускателя и аварийный режим. Срабатывание светового индикатора аварийного режима сопровождается коротким звуковым сигналом. Световой сигнал аварийного режима активен до повторного нажатия кнопки «ПУСК». Следует иметь в виду, что в реальной ситуации перед повторным включением пускателя необходимо устранить причину возникновения аварийной ситуации. Индикатор аварии при этом напоминает о необходимости устранения этой причины.

2.3 Компиляция и запись проекта в контроллер

После завершения проекта он компилируется в скетч ARDUINO через пункт главного меню программы или нажатием на иконку компиляции . При этом вызывается интегрированная среда разработки IDE ARDUINO. После загрузки среды со скетчем в главном меню оболочки «Инструменты» выбирается коммуникационный порт и запускается компиляция и загрузка скетча в

предварительно подключенную к порту USB персонального компьютера плату контроллера .

В меню «Инструменты» следует также убедиться в соответствии типа платы, указанной в FLProg и IDE ARDUINO.

2.4 Подготовка к работе

При домашней подготовке к работе изучить схему электромагнитного пускателя, собранную на ПЛК (рис. 2.9). Изучить функции и свойства функциональных блоков, из которых построен пускатель.

Привести на одном графике временные диаграммы сигналов при нажатии кнопки, имитирующей аварию на выходах: «Состояние пускателя», кнопки «АВАРИЯ» и кнопки «ПУСК», таймеров ТОF и TON, элемента «ИЛИ», триггера RS. При изображении временных диаграмм считать, что кнопка «ПУСК» нажата после завершения сигнала низкого уровня на выходе элемента «ИЛИ».

Ознакомиться с особенностями построения схемы на языке FBD в среде FLProg. Для самоконтроля письменно ответить на вопросы (п. 2.5).

2.5 Вопросы для самоконтроля

1. Как создать входы и выходы контроллера на плате?
2. Как создать переменную?
3. Какой тип следует установить переменной «Состояние пускателя» и какие значения может принимать эта переменная?
4. Чем определяется значение переменной «Состояние пускателя»?
5. С какой целью используется RS-триггер на плате управления?
6. Составьте таблицу состояний, связывающую сигналы кнопок «СТОП» и «АВАРИЯ» с значением сигнала R RS-триггера.

2.6 Порядок выполнения работы

1. Включить компьютер запустить среду проектирования (файл FLProg.exe).

2. Создать новый проект, выбрать язык FBD и контроллер в соответствии с используемой платой.
3. Собрать плату управления. Убедиться в корректности выполненных соединений (зелёная метка у заголовка платы).
4. Добавить плату сигнализации и собрать её схему. Убедиться в корректности схемы. Сохранить схему.
5. Подключить многофункциональную плату расширения к плате микроконтроллера (при необходимости). Следить за правильностью ориентации плат относительно друг друга и отсутствием замыкания контактов платы расширения на разъём USB платы микроконтроллера (штыри не должны быть полностью погружены в гнезда).
6. Подключить контроллер интерфейсным кабелем к USB разъёму компьютера. Компилировать проект.
7. После компиляции проекта и открытия программы в IDE ARDUINO в пункте меню «Инструменты» проверить соответствие типа платы используемой версии и в пункте «Порт» выбрать номер порта платы кликом ПКМ по нему.
8. Загрузить скетч в плату МК.
9. Выполнить проверку алгоритма функционирования, нажимая поочерёдно кнопки ПУСК – СТОП, ПУСК – АВАРИЯ, ПУСК – СТОП – АВАРИЯ, ПУСК – АВАРИЯ – СТОП. Записать состояние индикаторов для каждой последовательности нажатий. Отметить соответствие или несоответствие алгоритма функционирования спроектированного устройства алгоритму функционирования автомата. Зафиксировать результаты испытаний. При необходимости доработать схему, сохранить её и провести повторные испытания.
10. Модифицировать схему исключив таймеры в цепи управления динамиком. Сохранить схему.

11. Компилировать схему и провести её испытания в соответствии с п.9.
12. Результаты испытаний занести в отчёт и объяснить их.

2.7 Содержание отчёта

Отчёт должен содержать:

1. Титульный лист;
2. Оглавление;
3. Наименование работы, цель исследований;
4. Электрическую схему электромагнитного пускателя и описание принципа функционирования (из описания к лабораторной работе №1);
5. Схемы (все варианты) на языке FBD пускателя;
6. Описание используемого оборудования ПЛК из библиотеки функциональных блоков и его параметров;
7. Ответы на вопросы для самоконтроля;
8. Описание поведения контроллера в всех вариантах реализации и объяснения особенностей функционирования;
9. Ответы на контрольные вопросы;
10. Перечень литературы, использованной при подготовке и выполнении работы (ссылки на используемую техническую документацию).

2.8 Контрольные вопросы

1. Как обеспечивается сохранение состояния электромагнитного пускателя в схемах на языке LAD FBD?
2. Для чего используется задержка сигнала состояния пускателя в цепи динамика?
3. Какие изменения потребуются в схеме для индикации срабатывания аварийного оповещения световой вспышкой?
4. Как удалить вход контроллера из числа используемых в плате?

5. Что необходимо сделать для назначения номера ранее используемого и затем удалённого входа контроллера новому входу?
6. При назначении входа контроллера его тип может быть аналоговым или цифровым. Существуют ли ограничения при выборе входа контроллера в зависимости от его типа?

3 Ввод аналоговых сигналов в МК AVR

3.1 Цель работы

Изучение принципов построения модулей аналогового ввода на основе встроенного АЦП МК AVR.

3.2 Измерение температуры с помощью термометров сопротивления

Температура – один из важнейших параметров многих технологических процессов. Наиболее распространённые датчики для её измерения – это термопары (ТП), полупроводниковые термометры и термометры сопротивления (ТС). В таблице 3.1 приведены их обобщённые характеристики.

Таблица 3.1

Сравнительные характеристики типов датчиков температуры

Тип датчика температуры	Основные преимущества	Основные недостатки	Основные области применения
Термометры сопротивления	Высокая линейность номинальной статической характеристики Широкий диапазон рабочих температур Высокая стабильность основных параметров Устойчивость к воздействию агрессивных сред (ПТС) Относительно невысокая стоимость	Необходимость во внешней схеме для возбуждения	Широко используются как в относительно недорогих, так и в прецизионных системах измерения и контроля температур
Полупроводниковые термисторы	Дешевизна и доступность Высокий температурный коэффициент сопротивления	Необходимость во внешней схеме для возбуждения Высокая нелинейность номинальной статической характеристики Низкая стабильность основных параметров	Предназначены для применения в недорогих устройствах с низкими требованиями к точности измерений, в простых системах одно- и двухпорогового контроля температуры или для организации контроля температуры во второстепенных узлах сложной радиоэлектронной аппаратуры
Термопары	Самый широкий диапазон рабочих температур Высокая повторяемость	Низкое выходное напряжение Необходимость использования крупногабаритных конструкций для компенсации опорного	Широко используются в бюджетных устройствах с «электронной» компенсацией опорного спая с

	характеристик Высокое быстродействие	спая для достижения высокой точности измерений	невысокой точностью измерений. Используются в сверх- прецизионных измерительных системах 0,01...0,25 °С с компенсацией опор- ного спая с помощью со- уда Дьюара или специализирован- ных термостатов
--	---	---	--

В лабораторной работе рассмотрим только термометры сопротивления. ТС состоит из одного или нескольких термочувствительных элементов и внутренних соединительных проводов, помещенных в герметичный защитный корпус, а также внешних клемм и выводов, предназначенных для подключения к измерительному прибору. Чувствительный элемент (ЧЭ) термометра сопротивления представляет собой резистор, выполненный из металлической проволоки или плёнки, с выводами для крепления соединительных проводов, имеющий известную зависимость электрического сопротивления от температуры.

Под термином «термометр сопротивления» понимают как герметичный датчик в металлическом или керамическом корпусе с внешним разъёмом для подключения к измерительным приборам, так и сам чувствительный элемент, который может быть изготовлен в корпусе с проволочными выводами или в SMD-конструктиве для поверхностного монтажа.

Их основные преимущества – это высокая точность, широкий диапазон рабочих температур, малые размеры, устойчивость к вибрациям, линейность номинальной статической характеристики и относительно высокое значение температурного коэффициента сопротивления (ТКС). Основными материалами для изготовления ЧЭ ТС являются платина, медь, никель и их сплавы.

На практике чаще применяются платиновые термометры сопротивления с различной чистотой платины, которые обладают наивысшей стабильностью характеристик, устойчивостью к воздействию агрессивных сред и широким диапазоном рабочих температур.

По конструкции чувствительного элемента различают плёночные и проволочные термометры сопротивления. Медные и никелевые ТС чаще изготавливают из проволоки, а платиновые обоих типов. Плёночные имеют меньшую чувствительность к вибрациям, но и более узкий температурный диапазон.

3.2.1 Основные параметр ТС

Параметры ТС, в т.ч. классы допусков, схемы включения и др. нормируются ГОСТ 6651-2009.

Температурный коэффициент сопротивления

ТКС – это зависимость величины сопротивления ТС от температуры:

$$\alpha = (R_{100} - R_0) / (100 \cdot R_{100}),$$

где R_0 , R_{100} – сопротивление ТС при температуре 0 и 100 °С, округлённые до 5 знака после запятой.

В соответствие с ГОСТ 6651-2009 для платины $a = 0,00385$ 1/°С и $0,00391$ 1/ °С. ГОСТ определяет также номинальную статическую характеристику (НСХ) для двух диапазонов измеряемых температур минус 200 ... 0 °С и 0 ... 850 °С для каждого типа платиновых термометров сопротивления (ПТС) в зависимости от их ТКС.

Так для $a = 0,00385$ 1/°С в диапазоне температур минус 200 ... 0 °С НСХ описывается выражением:

$$R = R_0 [1 + At + Bt^2 + C(t-100)t^3].$$

Для диапазона 0 ... 850 °С:

$$R = R_0 (1 + At + Bt^2),$$

где $A = 3,9083 \times 10^{-3}$ °С, $B = -5,775 \times 10^{-7}$ °С, $C = -4,183 \times 10^{-12}$ °С.

Номинальное сопротивление

Номинальное сопротивление ТС – нормированное изготовителем сопротивление при °С, округлённое до целых и соответствующее ряду 10, 50, 100, 500,

1000 Ом. Однако используются ТС и 10 кОм. Это выгодно с точки зрения уменьшения влияния сопротивления проводов.

Диапазон измеряемых температур

Для большинства промышленных ТС установлены следующие диапазоны температур, °С:

платина -196 +660,

медь -50 +200,

никель -60 +180.

Схемы включения

Различают двух-, трёх- и четырёхпроводные схемы включения ТС. Двухпроводная схема включения не должна использоваться для классов допуска АА $\pm(0,1+0,0017|t|)$ °С и А $\pm(0,15+0,002|t|)$ °С. При использовании двухпроводной схемы следует вносить поправку на сопротивление проводов в измеренное значение сопротивления.

Трёхпроводная и четырёхпроводная схема позволяет практически исключить влияние сопротивления проводников при измерении сопротивления ТС.

При трёхпроводном включении измеряется сопротивление ТС, подключённого к линии, и дополнительно измеряется падение напряжения на проводнике, по которому протекает ток ТС (рис. 3.1).

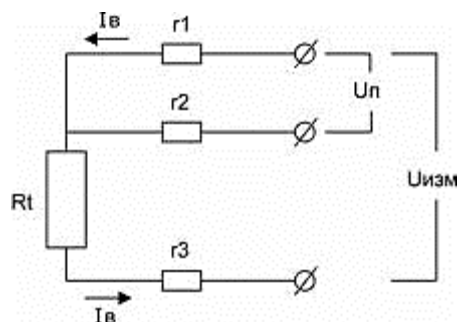


Рисунок 3.1 – Трёхпроводное подключение ТС

В этой схеме по проводнику с сопротивлением r_2 ток возбуждения не протекает, поэтому напряжение $U_{\text{п}}$ – является падением напряжения на проводнике. При

одинаковом сопротивлении проводов r_1 , r_3 напряжение на ТС (R_t) определяется как:

$$U_{TC} = U_{изм} - 2U_{п.} \quad (3.1)$$

В четырёхпроводной схеме напряжение на ТС измеряется с помощью двух дополнительных проводников, по которым ток не протекает, т.е. токовые и потенциальные зажимы разделены (рис. 3.2).

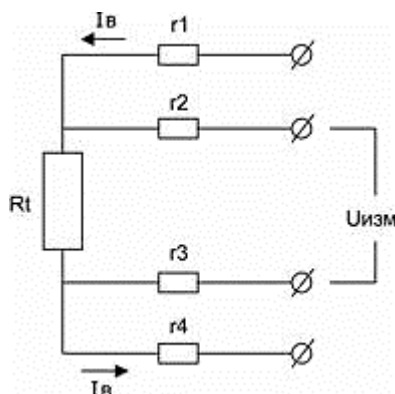


Рисунок 3.2 – Четырёхпроводная схема подключения ТС

Недостатками трёхпроводной и четырёхпроводной схем является сложность измерительной схемы и более высокая стоимость подключения в сравнении с двухпроводной схемой, особенно проявляющиеся в многоканальных системах измерения температуры. На рис. 3.3. для иллюстрации приведена схема устройства для измерения температуры при четырёхпроводном подключении ТС [5].

На опорном источнике МАХ6126А25, ОУ МАХ9617 и резисторе R2 реализован источник стабильного тока. Инструментальный ОУ МАХ4208 масштабирует напряжение на ТС. R1 – смещает напряжение в область рабочих синфазных напряжений масштабирующего ОУ. Измерительное устройство не является завершённым. Напряжение пропорциональное величине сопротивления ТС следует преобразовать в цифровой код, а цифровой код в соответствии номинальной статической характеристикой следует преобразовать в температуру. В

настоящее время для непосредственного преобразования напряжения ТС в цифровой код использую специализированные сигма-дельта АЦП.

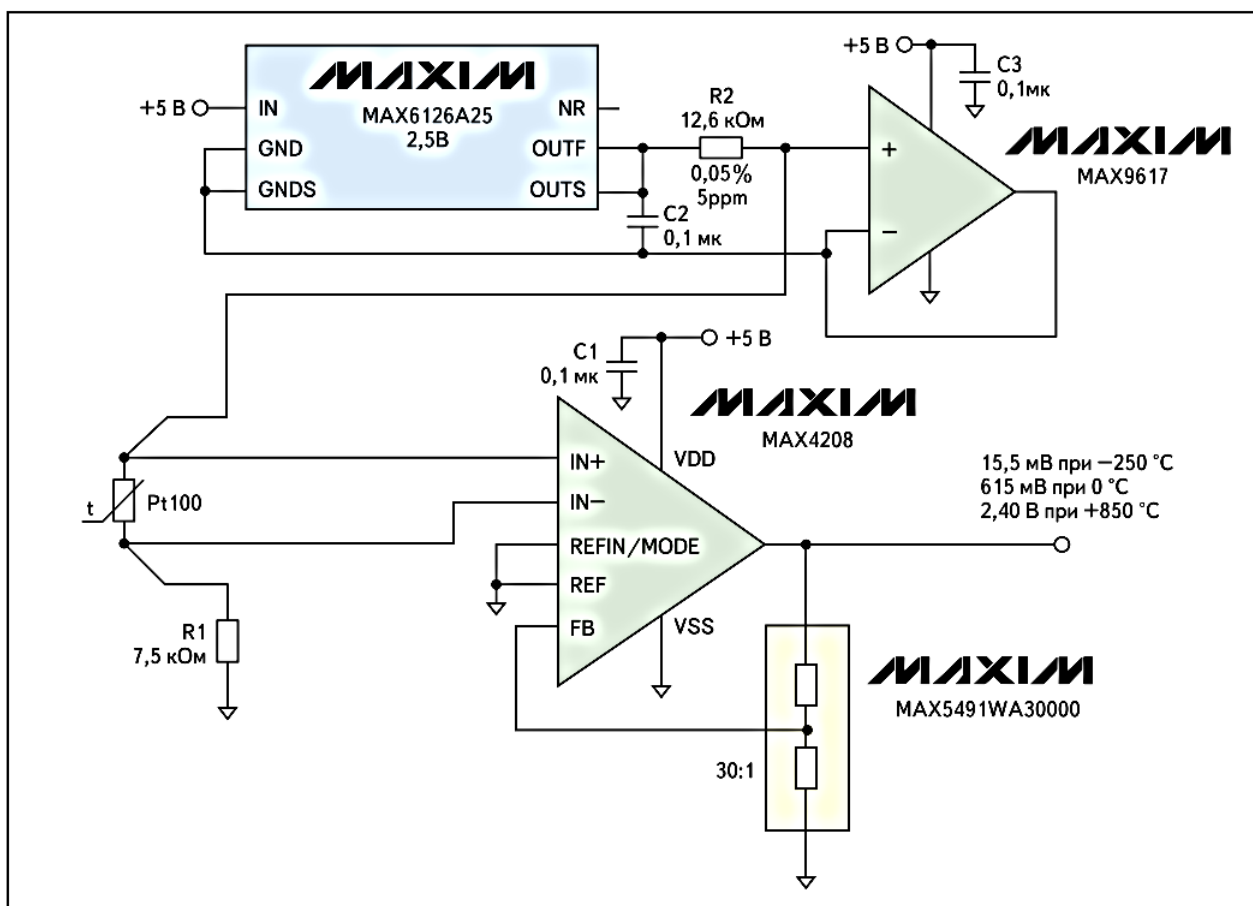


Рисунок 3.3 – Схема измерительного устройства при четырёхпроводном подключении ТС

Вместо источника тока для питания цепи ТС может использоваться источник напряжения. При этом питание ТС осуществляется через опорный резистор, а ток в цепи определяется измерением падения напряжения на опорном резисторе.

При невысоких требования к точности измерения и небольшой длине соединительных линий может использоваться двухпроводное подключение. В (2-5) рассмотрены способы реализации двухпроводного подключения ТС, позволяющие уменьшить влияние соединительных линий при измерении его сопротивления.

3.3 Двухпроводное подключение ТС с подавлением влияния сопротивления соединительных проводов

Модули ввода/вывода отечественных ПЛК обычно выполняются как универсальные, рассчитанные на измерение напряжений, подключение термопар или термометров сопротивления по схеме 4-х проводного подключения. Так в ПЛК MC-4A7D4R4O фирмы REALLAB, построенном на МК ATmega128 имеется 8 таких входов с допускаемой приведенной основной погрешностью $\pm 0,05\%$. Таким образом стандартная 4-х проводная схема накладывает существенные ограничения на количество обслуживаемых одним контроллером датчиков температуры. Применение двухпроводной схемы позволяет существенно расширить возможности ПЛК по количеству подключаемых датчиков [6].

Простейшая двухпроводная схема для измерения температуры может быть реализована на МК со встроенным АЦП. Рассматриваемый здесь способ иллюстрирует потенциальные возможности, т.к. встроенный АЦП МК AVR является лишь 10-разрядным и имеет допустимую погрешность $\pm 2 \text{ EMP}$.

Способ подавления влияния соединительных проводов при использовании двухпроводного подключения основан на измерении напряжения на ТС возбуждаемого напряжением шунтирующего его конденсатора (рис. 3.4). При этом ток по соединительным проводникам при измерении не протекает. Поскольку конденсатор C разряжается на термометр сопротивления (ТС), то напряжение необходимо измерять сразу после отключения цепи его заряда [2-3]. При подаче питания с вывода 4 контроллера ток, протекающий через опорный резистор R_0 , далее через сопротивление проводников линии подключения R_L и термометр сопротивления $R_{ТС}$, шунтированный конденсатором C , заряжает последний. Напряжение на опорном резисторе измеряется после установления переходного процесса. Поскольку цепь ТС представляет собой последовательное соединение, то напряжения на опорном резисторе и ТС

пропорциональны величинам их сопротивлений. Затем цепь питания ТС размыкается переводом вывод 1 МК в режим ввода (при этом его входное сопротивление достигает нескольких МОм).

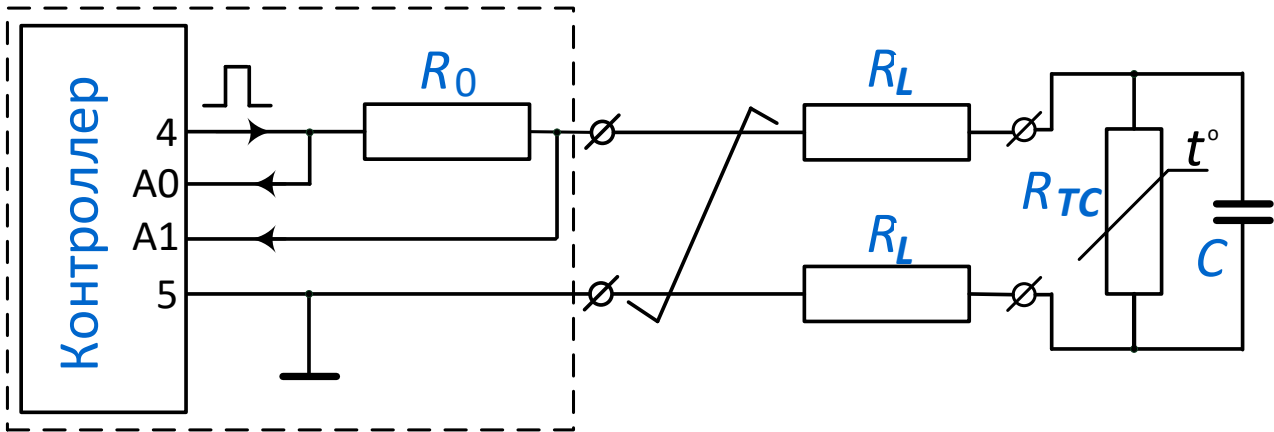


Рисунок 3.4 – Структурная схема измерения температуры при двухпроводном подключении термометра сопротивления

Переходный процесс в длинной линии при отключении возбуждающего ТС напряжения создаёт помехи при измерении напряжения на конденсаторе сразу после отключения. Поэтому для уменьшения погрешности измерения используются разные методы восстановления этого напряжения [7-10].

Представленный здесь метод базируется на восстановлении напряжения на конденсаторе U_{TC} в начальный момент времени разряда на основе результатов интегрирования переходного процесса разряда конденсатора, шунтирующего ТС.

При отключении цепи ТС на зажимах двухпроводной линии действует напряжение U , приложенное к выводам 3, 4 МК и определяемое сопротивлением R_{TC} , на которое разряжается конденсатор:

$$U = U_{TC} e^{-t/\tau}, \quad (1)$$

где t – время разряда конденсатора; $\tau = R_{TC}C$ – постоянная времени цепи разряда.

Интеграл напряжения на конденсаторе за временной интервал t от момента начала разряда:

$$S = \int_0^t U_{TC} e^{-t/\tau} dt = \tau U_{TC} (1 - e^{-t/\tau}). \quad (2)$$

Интегрируя на бесконечном интервале, а реально до практического завершения переходного процесса ($8 \dots 11 \tau$), получим:

$$S_M = \tau U_{TC}. \quad (3)$$

При интегрировании на ограниченном интервале t_1 :

$$S_1 = \tau U_{TC} (1 - e^{(-t_1/\tau)}) = S_M (1 - e^{(-t_1/\tau)}). \quad (4)$$

Из (4) определим постоянную времени:

$$\tau = -t_1 / \ln(1 - S_1 / S_M). \quad (5)$$

Из (3) и (5) определяем напряжение на ТС в конце переходного процесса заряда конденсатора, равное напряжению в начальный момент его разряда:

$$U_{TC} = -S_M \frac{\ln(1 - S_1 / S_M)}{t_1}. \quad (6)$$

При этом сопротивление термометра сопротивления:

$$R_{TC} = R_0 \frac{U_{TC}}{U_0}.$$

Измеряемая температура находится по градуировочной характеристике ТС.

Макетная плата для демонстрации метода измерения представлена на рис.3.5.

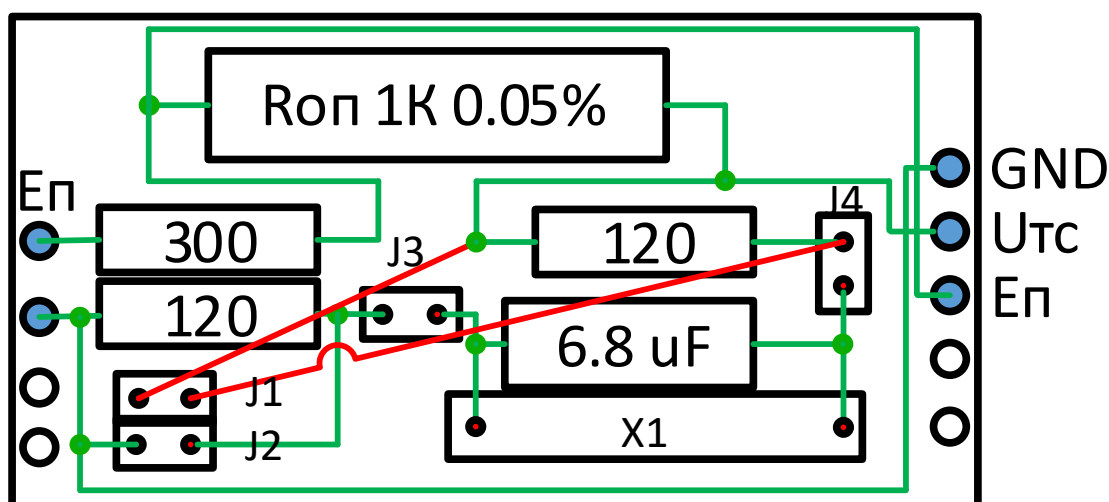


Рисунок 3.5 – Макетная плата для демонстрации двухпроводного метода подключения ТС

На плате предусмотрено два резистора по 120 Ом, имитирующих сопротивление проводов соединительной линии. Предусмотрена возможность шунтирования этих резисторов джамперами J1, J2. Разъём X1 для подключения имитирующего ТС резистора шунтирован конденсатором 6,8 мкФ. Джамперы J3, J4 позволяют разорвать измерительную цепь и подключить в разрыв присоединительную линию. Установка макетной платы на контроллер осуществляется в соответствии с рис. 3.6.

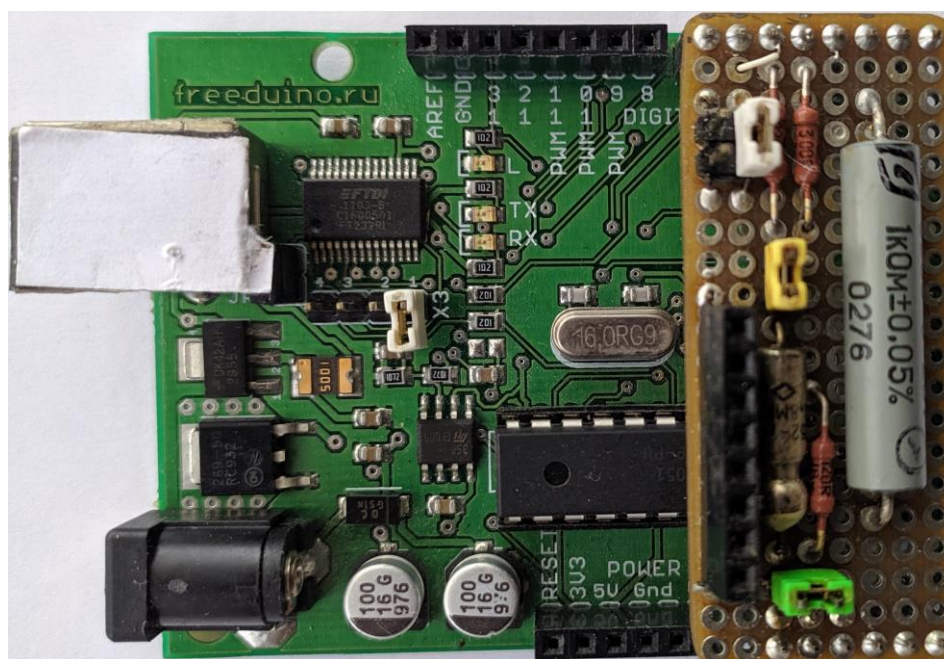


Рисунок 3.6 – Демонстрационный стенд

3.4 Подготовка к работе

Ознакомьтесь с характеристиками ТС и схемами их подключения. Изучить двухпроводную схему подключения ТС. Изучить программное обеспечение измерительного устройства (приложение). Ответить на вопросы для самоконтроля.

3.5 Вопросы для самоконтроля

1. К каким выводам подключена цепь возбуждения термометра сопротивления?

2. Какие выводы встроенного АЦП задействованы на плате для измерения напряжения на опорном резисторе.
3. Когда и как измеряется напряжение на опорном резисторе?
4. Какой вывод контроллера используется при измерении напряжения на ТС?
5. С какой целью используются задержки в программе?
6. Из каких соображений выбирается величина этих задержек?
7. Для чего проводятся повторные измерения напряжения на опорном резисторе?
8. Каково разрешение встроенного АЦП в измерительном устройстве?
9. Как скажется на результатах измерения задержка запуска АЦП после отключения цепи возбуждения?
10. Каким образом отключается цепь возбуждения?

3.6 Порядок выполнения работы

1. Установить (при необходимости) макетную плату на плату контроллера в соответствии с рис. 3.6.
2. Джемперы J1, J2 установить в положение – разомкнуто (или снять), J3, J4 – в положение замкнуто. Измеряемый резистор установить в разъем 1.
3. Подключить плату к USB разъёму компьютера.
4. Включить компьютер, загрузить скетч `Termo_Net_Integral` и `Terminal V1.9b`.
5. Установить в программе терминала битрейт 115200.
6. Установить в программе «`Termo_Net_Integral`» начальное значение времени интегрирования $t1 \rightarrow \text{int } TIO = 200$.
7. Загрузить скетч в контроллер.
8. Установить номер порта в терминальной программе, соответствующий порту контроллера и выполнить соединение нажав “Connect”.

9. Провести не менее 100 измерений и разорвать связь терминала с контроллером нажав “Disconnect”.
10. Сохранить результаты измерений в текстовой форме, для чего выделить текст в окне терминала скопировать Ctrl+C и вставить в документ «WORD» Ctrl+V.
11. Перенести рассчитанные значения в Excel, вычислить среднее значение и дисперсию результатов измерения. Определить отклонение от действительного значения измеряемого сопротивления.
12. Устанавливая последовательно значения времени интегрирования 300, 400, 500, 600, 700, 800 повторить п. п. 7-11 программы исследований.
13. По рассчитанным отклонениям среднего значения измеренного сопротивления от действительного значения определить отношение U_{ADC1}/U_{ADC2} при котором достигается минимальная величина этого отклонения.

3.7 Содержание отчёта

1. Титульный лист;
2. Оглавление;
3. Наименование работы, цель исследований;
4. Структурную схему стенда;
5. Текст программы измерения сопротивления ТС;
6. Ответы на вопросы для самоконтроля;
7. Результаты измерения и оценок средних значений и дисперсии в виде приложения Excel таблиц;
8. Графики поведения среднего значения, отклонения от действительного и дисперсии в зависимости от выбора времени измерения t_1 ;
9. Выводы об эффективности подавления влияния сопротивления линии и оптимальных условиях измерения;
10. Ответы на контрольные вопросы;

11. Перечень литературы, использованной при подготовке и выполнении работы (ссылки на используемую техническую документацию).

3.8 Контрольные вопросы

1. С какой целью используются трёхпроводные и четырёхпроводные схемы подключения термометров сопротивления?
2. Поясните, как функционирует трёхпроводная схема измерения сопротивления ТС.
3. Поясните, как функционирует четырёхпроводная схема измерения сопротивления ТС.
4. Каким образом удаётся исключить влияние сопротивления проводов в изучаемой в работе двухпроводной схеме подключения ТС?
5. Почему нельзя определить напряжение на термометре сопротивления измеряя его однократно на входе линии подключения ТС встроенным АЦП?
6. Как оценить время интегрирования переходного процесса разряда конденсатора шунтирующего ТС?
7. Как поведёт себя ошибка измерения опорного напряжения и напряжения на входе линии при изменении сопротивления ТС от номинального значения до максимального? Изобразить качественно график и дать пояснения.
8. По какой причине при интегрировании напряжения используется непрерывный режим работы АЦП вместо программного запуска преобразования?
9. Как достигается обесточивание линии питания термометра сопротивления?
10. Из каких соображений выбирается ёмкость конденсатора шунтирующего ТС?

Литература

1. 15 ведущих мировых производителей программируемых логических контроллеров (ПЛК) [Electronic resource]. – Режим доступа: <http://electrik.info/plc/1806-top-veduschih-proizvoditeley-plc.html>, свободный.
2. Шишов, О. В. Современные средства АСУ ТП : учебник / О. В. Шишов. – Москва ; Вологда : Инфра-Инженерия, 2021. – 532 с.
3. Что такое FLProg [Электронный ресурс]. URL: <https://flprog.ru/chto-takoe-flprog/> (дата обращения 16.06.2023).
4. Петин, В. А. Проекты с использованием контроллера Arduino. — СПб.: БХВ-Петербург, 2014. — 400 с.
5. Андрусевич, А. Термометры сопротивления: от теории к практике // Компоненты и технологии. 2011. №7. С. 76-81.
6. Пат. № 2775873. РФ, МПК G01K 7/16 (2006.01). Способ многоканального измерения температуры. [Текст] / Бондарь О. Г., Брежнева Е.О. – 2021116567, заявл. 08.06.2021; опубл. 11.07.2022 Бюл. № 20. - 7 с.
7. Бондарь О. Г., Брежнева Е. О., Родионов П. С. Многоканальный преобразователь температуры // Изв. вузов. Приборостроение. 2022. Т. 65, № 4. С. 254—261. <https://doi.org/10.17586/0021-3454-2022-65-4-254-261>
8. Пат № [2752132](https://patents.google.com/patent/RU2752132). Российская Федерация, МПК G01K 7/16; СПК G01K 7/16 (2021.05). Способ измерения температуры. [Текст] / Бондарь О. Г., Брежнева Е.О., Двойных Е.С. – [2020135234](https://patents.google.com/patent/RU2020135234), заявл. 27.10.2020; опубл. 23.07.2021, Бюл. № 21. - 9 с.
9. Бондарь О.Г., Брежнева Е.О., Калмыков А.И. Повышение точности измерения температуры: метод двухпроводного подключения термометра сопротивления //Измерительная техника 2022. №3. С. 53-58. <https://doi.org/10.32446/0368-1025it.2022-3-53-58>

10. Пат. № 2781754. РФ, МПК G01K 7/16 (2006.01). Способ измерения температуры / Бондарь О.Г., Брежнева Е.О., Поляков Н.В. – 2022103658, заявл. 14.02.2022; опубл. 17.10.2022 Бюл. № 29. – 10 с.

Приложение 1

Программа измерения температуры при двухпроводном подключении ТС.

```

/* Метод измерения температуры с ослаблением влияния её обесточиванием
 * напряжение на ТС оценивается методом интегрирования
 * напряжения (накоплением результата).
 * Накопление осуществляется в течении времени t1, задаваемого
 * количеством измерений и результат сохраняется.
 * Интегрирование продолжается до завершения разряда.
 * Время разряда не целесообразно брать более 11 постоянных времени
 * (при этом погрешность за счёт недоразряда станет < 0,002%)
 *
 * В данном алгоритме АЦП работает в режиме непрерывных измерений,
 * результат сохраняется по прерыванию. Количество измерений контролируется
 * в основном цикле.
 *
 * R0 = 1К, C=6.8uf, tau = 0.68
 * Полное время интегрирования должно быть >7t (погрешность <0,1%)
 *
 * Для представления результата используется вывод на монитор
 * Для контроля выводятся дополнительные параметры
 */

/*****
/* Отключение цепи возбуждения ТС осуществляется переводом в режим ввода
   При этом необходимо установить низкий уровень сигнала
   отключения подтягивающего резистора (цепь SW_REF).
*/

#define SW_REF 4 // вывод цепи импульсного питания
#define SW_GND 5 // вывод подключения к общему проводу (для многоканального измерения)
/*****

#define K 4000 // количество измерений напряжения на опорном резисторе

unsigned long U_ADC1; // интеграл за интервал TI
unsigned long U_ADC2; // полный интеграл за TIMax
unsigned long U_REF = 0; // накопленное напряжение на опорном резисторе

float K_TERM = 0; // относительное изменение термометра сопротивления

float U_ADC3 = 0;

int UDL = A1; // контакт для измерения напряжения на входе двухпроводной линии
int UPW = A0; // контакт для измерения напряжения питания
int UGND = A2; // контакт для измерения напряжения земли

float TERM = 0; // термосопротивление, кОм
float TERM1 = 0; // резервная переменная
int R_REF = 1; // Сопротивление опорного резистора 1 кОм
int TI0 = 700; // Время интегрирования t1 при начальном сопротивлении термометра
U_ADC1/U_ADC2=0,77 - оптимум
int TI = TI0; // начальное значение t1 (постоянная t=6,8 ms)
int TIMax = 0; // полное время интегрирования TIMax > 7*TI
volatile int REZULT = 0; // результат однократного измерения АЦП
volatile int Flag = 0; // флаг состояния интегрирования
int count = 0; // счётчик времен интегрирования

```

```

unsigned long m = 0;

//=====

void setup() {

    Serial.begin(115200);          // настройка скорости последовательного порта

    pinMode(SW_REF, OUTPUT); // цепи опорного источника
    pinMode(SW_GND, INPUT);  // и общего провода на вывод
    // digitalWrite(SW_GND, LOW); // подключить землю подключена переключкой постоянно
    sei(); // устанавливаем флаг разрешения прерывания

//=====

// Настройка АЦП - СПРАВКА

// DDRB = B01000000; // pin 12 в режиме OUTPUT при возбуждении и INPUT при отключении
// DIDR0 = 0x3F; // отключаем цифровые входы порта A для снижения шума

// Регистр ADMUX
// -----
// | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
// |-----|
// | ИОН | ИОН | Выравнивание | - | Мультиплексор |
// | 0 | 1 | результата | Резерв | 0 | 0 | 0 | 0 |
// |-----|
// | Питание | 0-вправо | вход A0
// | ADMUX = 0x40; // измеряем на ADC0, используем внутреннее опорное напр.= Епит
//
// Регистр ADCSRA
// Настройка делителя: 0,1-2; 2-4; 3-8; 4-16; 5-32; 6-64; 7-128
// -----
// | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
// |-----|
// | Разреш. | Старт | Режим непрер | Флаг | Разрешение | Делитель |
// | АЦП | АЦП | однократный | прерыв. | прерываний | |
// |-----|
// | 1 | 0/1 | 1/0 | | | 1 |
//
// Регистр ADCSRB
// -----
// | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
// |-----|
// | - | Компарат | - | - | - |
// | резерв | к MUX1 | резерв | резерв | резерв | Источник запуска |
// |-----|

}

//=====
void loop()
{
    U_ADC2 = 0;
    U_REF = 0;

    ADMUX = 0b01000000; // 0B0100000 опорное напряжение Епит. Выбран вход A0
    ADCSRA = 0b10010000 | 0x05; // разрешение АЦП, оптимально CLK/32. 26 мкс/измерение
    // Установка D5 автопреобразования блокирует измерения

```

```

// digitalWrite(SW_GND, LOW); // подключить землю
pinMode(SW_REF, OUTPUT); // цепь на вывод
digitalWrite(SW_REF, HIGH); // подключить цепь питания термометра сопротивления

delay(60); // зарядить конденсатор 8 постоянных времени

// цикл измерения среднего значения напряжения на опорном резисторе

for (int i=1; i<=K; i++)
{
    U_REF = U_REF + analogRead(UPW); // измерить напряжение Епит
//    U_ADC3 = analogRead(UDL); // холостое измерение для установления напряжения
    U_ADC2 = U_ADC2 + analogRead(UDL); // напряжение цепи датчика
}

U_REF = U_REF - U_ADC2; // интеграл напряжения на опорном резисторе
K_TERM = (float)U_ADC2/U_REF; // относительная величина сопротивления термосопротивления

TIMax = 10000; // > TI*7; // время полного интегрирования.

// =====

// интегрирование напряжения измерительной цепи в течении TIMax

ADMUX = 0x41; // Измерение на ADC1, используется опорное напр.= Епит
ADCSRA = 0b10111000 | 0x05; // Разрешить прерывания; (Делитель АЦП=32, Тизм=26 мкс)
ADCSRB = 0x40; // включаем АЦ каналы MUX в режим непрерывной выборки 40

count = 1; // начальное значение счётчика количества измерений.
Flag = 0;

U_ADC1 = 0;
U_ADC2 = 0;

m = micros(); // контроль времени измерения

pinMode(SW_REF, INPUT); // отключить опорное напряжение
digitalWrite(SW_REF, LOW); // отключить подтягивающий резистор

// Запускаем преобразование установкой бита 6 (ADSC) в ADCSRA
ADCSRA |= (1 << ADSC);

while (count <= TIMax)
{
    if (Flag==1)
    {
        U_ADC2 = U_ADC2 + RESULT; // интегрирование на интервале TIMax
        if (count <= TI)

        {
            U_ADC1 = U_ADC2;
        }

        count ++; // инкремент счётчика измерений
    }
}

```

```

    Flag = 0;
  }
}
ADCSRA &= ~(1 << ADSC); // стоп АЦП

m = micros()- m; // Измерение времени работы измерительного блока

// =====

// расчёт величины сопротивления термометра сопротивления

TERM1 = (float)U_ADC1/U_ADC2;

TERM1 = (float)(1-TERM1);
TERM1 = -log(TERM1);
TERM = (float)TERM1*(U_ADC2/VI)/(U_REF/K)*R_REF;

//*****

// Вывод данных в порт
U_ADC3 = (float)U_REF/K; // среднее значение напряжения на опорном резисторе (контроль)
Serial.print(" m = ");

Serial.print(m);

Serial.print(" U_REF := ");
Serial.print(U_ADC3, 2); // использована переменная плав. формата для контроля U_REF

Serial.print(" U_ADC1 := ");
Serial.print(U_ADC1);
Serial.print(" U_ADC2 := ");
Serial.print(U_ADC2);

U_ADC3 = (float)U_ADC1/U_ADC2 ;
Serial.print(" U_ADC1/U_ADC2= ");
Serial.print(U_ADC3, 5);

Serial.print(" t1= ");
Serial.print(TI);

Serial.print(" RT := ");
Serial.print(TERM, 4); // печать значения термосопротивления в кОм
Serial.println(" kOm");

delay(10);
}

// =====

//*** Процедура обработки прерывания АЦП ***/
ISR(ADC_vect)
{
  RESULT = ADCL; // сохраняем младший байт результата АЦП
  RESULT += ADCH << 8; // сохраняем старший байт АЦП
  Flag = 1;
}
// =====

```