**ЗАДАНИЕ**

Общая характеристика объекта управления*.* Агрегат содержит один резервуар (Р), который оснащен датчиками и имеет входные трубы и выпускную трубу. Дополнительно резервуары могут содержать электронагреватель (TEN), миксер, приводимый во вращение электродвигателем (ED). Задана последовательность операций, которая обеспечивает химико-технологический процесс приготовления продута.

Датчик нижнего уровня (DL) сигнализирует о том, что резервуар пуст, если сигнал DL=0. Датчики верхнего уровня (DH) сигнализируют о том, что резервуар заполнен, если сигнал DH=0. На каждой трубе имеется кран, приводимый в движение электромагнитом. Кран будет открыт, если на электромагнит подать сигнал управления низкого уровня (Kn=0, n–номер крана). Если сработает датчик нижнего уровня, то соответствующий выпускной кран должен быть закрыт.

Датчик температуры (DT) использует термосопротивление, величина которого зависит от температуры. Датчик проградуирован так, что при температуре 10 С0 он выдает 0,1 В, а при температуре 200 С0 – 2,0В.

После включения нагревателя устройство должно поддерживать температуру в диапазоне от Тмин до Тмакс. В исходном состоянии резервуары пуст, все краны закрыты, нагреватель и двигатель выключены.

Технологический цикл должен начинаться после нажатия кнопки «Пуск». По завершении цикла подать звуковой сигнал частотой 1 – 2 кГц длительностью 2 с.

*Описание логики работы устройства*

*Вариант 2.*

1. Открыть кран К1 на 30 с.
2. Открыть кран К2 до заполнения резервуара. Р
3. Запустить электродвигатель на 5 мин. Одновременно включить нагреватель.
4. Выключить нагреватель. Слить жидкость из Р.
5. Открыть кран К3 до заполнения Р водой.
6. Включить двигатель и перемешивать в течение 3 мин.
7. Слить воду.

Цикл завершен.

Даны значения Тмин=85, Тмакс=90. Режим работы светодиодных цифровых индикаторов – динамический.

Студент

Преподаватель

**Содержание**

Задание

Введение

1 Аппаратная часть

1.1 Структура устройства

1.2 Описание принципиальной электрической схемы

1.3 Выбор и расчет элементов схемы

1.4 Конструкция устройства

2 Программное обеспечение

2.1 Алгоритм функционирования

2.2 Описание программы

2.3 Технология подготовки и отладки ПО

2.4 Результаты тестирования и отладки ПО

Заключение

Список использованных литературных источников

Приложение А

**Введение**

Область применения микроконтроллеров - это различные контроллеры устройств автоматики, пластиковые карты, контроллеры периферийных устройств.

Развитие микроэлектроники и её широкое применений изделий в промышленном производстве, в устройствах и системах управления самыми разнообразными объектами и процессами является в настоящее время одним из основных направлений научно-технического прогресса.

Использование микроконтроллеров в изделиях не только приводит к повышению технико-экономических показателей (стоимости, надежности, потребляемой мощности, габаритных размеров), но и позволяет сократить время разработки изделий и делает их модифицируемыми, адаптивными. Использование микроконтроллеров в системах управления обеспечивает достижение высоких показателей эффективности при низкой стоимости.

Микроконтроллеры представляют собой эффективное средство автоматизации разнообразных объектов и процессов.

AVR фирмы Atmel 8-битные RISC-микроконтроллеры для встраиваемых приложений являются, пожалуй, наиболее интересным направлением, развиваемым фирмой. Они представляют собой мощный инструмент, прекрасную основу для создания современных высокопроизводительных и экономичных встраиваемых контроллеров многоцелевого назначения.

Несмотря на то, что микроконтроллеры AVR появились на рынке около 10 лет назад, их популярность до сих пор очень высока. С каждым годом они захватывают все новые и новые ниши на рынке. Не последнюю роль в этом играет соотношение показателей цена/быстродействие/энергопотребление, до сих пор являющееся едва ли не лучшим на рынке 8-битных микроконтроллеров. Кроме того, постоянно растет число выпускаемых сторонними производителями разнообразных программных и аппаратных средств поддержки разработок устройств на их основе. Все это позволяет говорить о микроконтроллерах AVR как об индустриальном стандарте среди 8-битных микроконтроллеров.

В настоящее время в рамках единой базовой архитектуры микроконтроллеры AVR подразделяются на несколько семейств:

- Tiny AVR;

- Mega AVR;

- Mega AVR для специальных применений;

- ASIC/FPGA AVR.

Далее будет более подробно рассмотрено семейство Mega. Микроконтроллеры этого семейства имеют наиболее развитую периферию, наибольшие среди всех микроконтроллеров AVR объемы памяти программ и данных. Они предназначены для использования в мобильных телефонах, в контроллерах различного периферийного оборудования (такого как принтеры, сканеры, современные дисковые накопители, приводы CD-ROM/DVD-ROM и т. п.), в сложной офисной технике и т. д.

Микроконтроллеры семейства Mega поддерживают несколько режимов пониженного энергопотребления, имеют блок прерываний, сторожевой таймер и допускают программирование непосредственно в готовом устройстве.

**1 Аппаратная часть**

* 1. **Структура устройства**



Рисунок1. Структурная схема устройства

Используемые сокращения:

1)DL – Датчик нижнего уровня;

2) DH - Датчики верхнего уровня ;

3) DT - Датчик температуры;

4)K1, K2, K3 – электромагнитные краны ;

5)TEN – электрический нагревательный элемент;

6) Arduino Duemilanove – платформа Arduino Duemilanove.

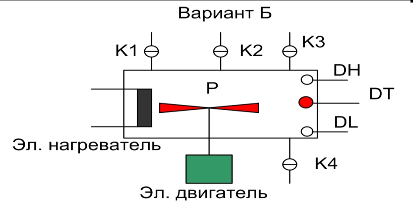


Рисунок 1.1 схема установки

**1.2 Описание принципиальной электрической схемы**

Arduino Duemilanove

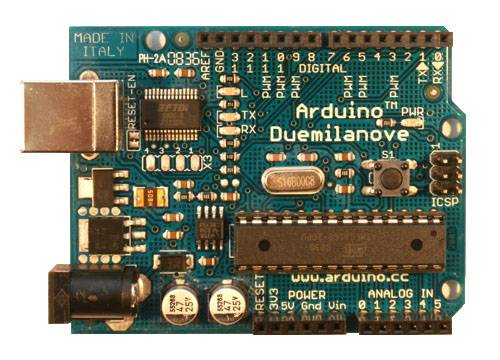


Рисунок 2. Arduino Duemilanove

**Общие сведения**

Arduino Duemilanove («2009») построена на одном из микроконтроллеров: ATmega168 (техническое описание) или ATmega328 (техническое описание). Платформа содержит 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или батареи.

Duemilanove (в переводе с итальянского – 2009) была названа в честь года своего выпуска – 2009 год. Данная платформа является последней из серии Arduino с USB. Для сравнения с предыдущими версиями необходимо обратиться к полному списку плат Arduino.

Характеристики

Микроконтроллер ATmega168

Рабочее напряжение 5 В

Входное напряжение (рекомендуемое) 7-12 В

Входное напряжение (предельное) 6-20 В

Цифровые Входы/Выходы 14 (6 из которых могут использоваться как выходы ШИМ)

Аналоговые входы 6

Постоянный ток через вход/выход 40 мА

Постоянный ток для вывода 3.3 В 50 мА

Флеш-память 16 Кб (ATmega168) или 32 Кб (ATmega328) при этом 2 Кб используются для загрузчика

ОЗУ 1 Кб (ATmega168) или 2 Кб (ATmega328)

EEPROM 512 байт (ATmega168) или 1 Кб (ATmega328)

Тактовая частота 16 МГц

Схема и исходные данные

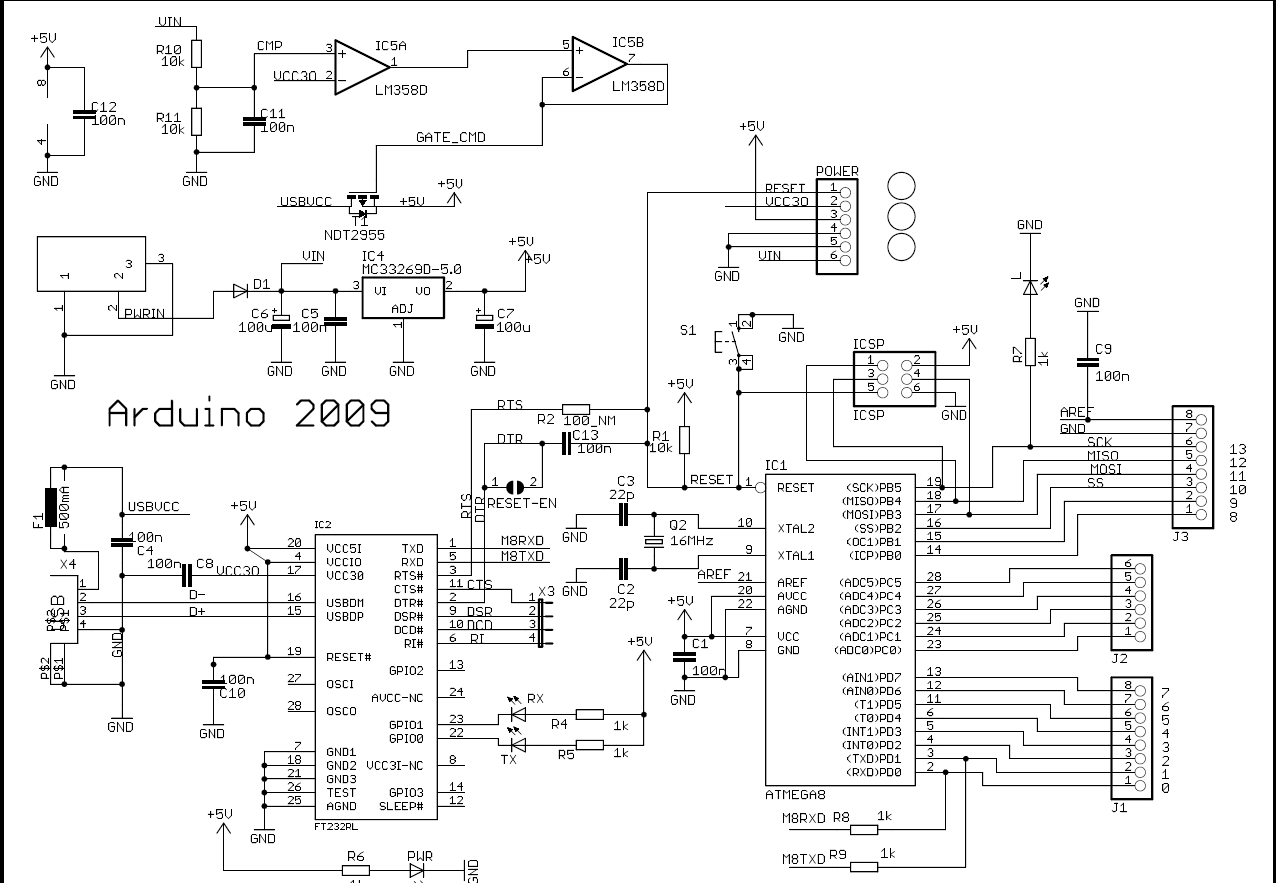


Рисунок 3. Принципиальная схема Arduino Duemilanove

**Питание**

Arduino Duemilanove может получать питание через подключение USB или от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с центральным положительным полюсом. Провода от батареи подключаются к выводам Gnd и Vin разъема питания.

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Выводы питания:

VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод.

5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.

3V3. Напряжение на выводе 3.3 В генерируемое микросхемой FTDI на платформе. Максимальное потребление тока 50 мА.

GND. Выводы заземления.

Память

Микроконтроллер ATmega168 имеет 16 кБ флеш-памяти для хранения кода программы, а микроконтроллер ATmega328, в свою очередь, имеет 32 кБ (в обоих случаях 2 кБ используется для хранения загрузчика). ATmega168 имеет 1 кБ ОЗУ и 512 байт EEPROM (которая читается и записывается с помощью библиотеки EEPROM), а ATmega328 – 2 кБ ОЗУ и 1 Кб EEPROM.

Входы и Выходы

Каждый из 14 цифровых выводов Duemilanove, используя функции pinMode(), digitalWrite(), и digitalRead(), может настраиваться как вход или выход. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:

Последовательная шина: 0 (RX) и 1 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины FTDI USB-to-TTL.

Внешнее прерывание: 2 и 3. Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt().

ШИМ: 3, 5, 6, 9, 10, и 11. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции analogWrite().

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Посредством данных выводов осуществляется связь SPI, которая, хотя и поддерживается аппаратной частью, не включена в язык Arduino.

LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

На платформе Duemilanove установлены 6 аналоговых входов, каждый разрешением 10 бит (т.е. может принимать 1024 различных значения). Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством вывода AREF и функции analogReference(). Некоторые выводы имеют дополнительные функции:

I2C: 4 (SDA) и 5 (SCL). Посредством выводов осуществляется связь I2C (TWI), для создания которой используется библиотека Wire.

Дополнительная пара выводов платформы:

AREF. Опорное напряжение для аналоговых входов. Используется с функцией analogReference().

Reset. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки с целью блокировки микроконтроллера.

Обратите внимание на соединение между выводами Arduino и портами ATmega168.

**Связь**

На платформе Arduino Duemilanove установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega168 и ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема FTDI FT232RL направляет данный интерфейс через USB, а драйверы FTDI (включены в программу Arduino) предоставляют виртуальный COM порт программе на компьютере. Мониторинг последовательной шины (Serial Monitor) программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Светодиоды RX и TX на платформе будут мигать при передаче данных через микросхему FTDI или USB подключение (но не при использовании последовательной передачи через выводы 0 и 1).

Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Duemilanove.

ATmega168 и ATmega328 поддерживают интерфейсы I2C (TWI) и SPI. В Arduino включена библиотека Wire для удобства использования шины I2C. Более подробная информация находится в документации. Для использования интерфейса SPI обратитесь к техническим данным микроконтроллеров ATmega168 и ATmega328.

Программирование

Платформа программируется посредством ПО Arduino. Из меню Tools > Board выбирается «Arduino Diecimila или Duemilanove w/ ATmega168» или «Arduino Duemilanove w/ ATmega328» (согласно установленному микроконтроллеру). Подробная информация находится в справочнике и инструкциях.

Микроконтроллеры ATmega168 и ATmega328 поставляются с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через выводы ICSP (внутрисхемное программирование). Подробная информация находится в данной инструкции.

Автоматическая (программная) перезагрузка

Duemilanove разработана таким образом, чтобы перед записью нового кода перезагрузка осуществлялась самой программой, а не нажатием кнопки на платформе. Одна из линий FT232RL, управляющих потоком данных (DTR), подключена к выводу перезагрузки микроконтроллеров ATmega168 или ATmega328 через резистор 100 нФ. Активация данной линии, т.е. подача сигнала низкого уровня, перезагружает микроконтроллер. Программа Arduino, используя данную функцию, загружает код одним нажатием кнопки Upload в самой среде программирования. Подача сигнала низкого уровня по линии DTR скоординирована с началом записи кода, что сокращает таймаут загрузчика.

Функция имеет еще одно применение. Перезагрузка Duemilanove происходит каждый раз при подключении к программе Arduino на компьютере с ОС Mac X или Linux (через USB). Следующие полсекунды после перезагрузки работает загрузчик. Во время программирования происходит задержка нескольких первых байтов кода во избежание получения платформой некорректных данных (всех, кроме кода новой программы). Если производится разовая отладка скетча, записанного в платформу, или ввод каких-либо других данных при первом запуске, необходимо убедиться, что программа на компьютере ожидает в течение секунды перед передачей данных.

На Duemilanove имеется возможность отключить линию автоматической перезагрузки разрывом соответствующей линии. Контакты микросхем с обоих концов линии могут быть соединены с целью восстановления. Линия маркирована «RESET-EN». Отключить автоматическую перезагрузку также возможно подключив резистор 110 Ом между источником 5 В и данной линией. Подробная информация находится в соответствующей ветке форума.

Токовая защита разъема USB

В Arduino Duemilanove встроена перезагружаемая плавкая вставка (предохранитель), защищающая порт USB компьютера от токов короткого замыкания и сверхтоков. Хотя практически все компьютеры имеют подобную защиту, тем не менее, данный предохранитель обеспечивает дополнительный барьер. Предохранитель автоматически прерывает обмен данных при прохождении тока более 500 мА через USB порт.

Физические характеристики

Длинна и ширина печатной платы Duemilanove составляют 6.9 и 5.3 см соответственно. Разъем USB и силовой разъем выходят за границы данных размеров. Три отверстия в плате позволяют закрепить ее на поверхности. Расстояние между цифровыми выводами 7 и 8 равняется 0,4 см, хотя между другими выводами оно составляет 0,25 см.

**1.2 Описание принципиальной электрической схемы**

Схема электрическая принципиальная представлена на рисунке 1.2

Рисунок 4



**1.3 Выбор и расчет элементов схемы**

При разработке устройства были использованы следующие электронные компоненты:

– термодатчик DS 18B20;

- платформа Arduino

– датчики PL – нижнего уровня и датчик PH – высокого уровня

- электрический нагревательный элемент

Для включения электронагревателя или электродвигателя микроконтроллер должен выдать сигнал низкого уровня на соответствующую линию порта вывода.

После включения нагревателя устройство должно поддерживать температуру в диапазоне от Тмин до Тмакс. Текущее значение температуры отображается на 3-значном цифровом индикаторе.

В исходном состоянии резервуары пуст, все краны закрыты, нагреватель и двигатель выключены.

Технологический цикл должен начинаться после нажатия кнопки «Пуск». По завершении цикла подать звуковой сигнал частотой 1 – 2 кГц длительностью 2 с.

Датчик температуры (DT) использует термосопротивление, величина которого зависит от температуры. Датчик проградуирован так, что при температуре 10 С0 он выдает 0,1 В, а при температуре 200 С0 – 2,0В.

Даны значения Тмин=85, Тмакс=90. Следовательно Тмин =174 Тмакс =184

**1.4 Конструкция устройства**

Общая характеристика объекта управления*.* Агрегат содержит один резервуар (Р), который оснащен датчиками и имеет входные трубы и выпускную трубу. Дополнительно резервуары могут содержать электронагреватель (TEN), миксер, приводимый во вращение электродвигателем (ED). Задана последовательность операций, которая обеспечивает химико-технологический процесс приготовления продута.

Датчик нижнего уровня (DL) сигнализирует о том, что резервуар пуст, если сигнал DL=0. Датчики верхнего уровня (DH) сигнализируют о том, что резервуар заполнен, если сигнал DH=0. На каждой трубе имеется кран, приводимый в движение электромагнитом. Кран будет открыт, если на электромагнит подать сигнал управления низкого уровня (Kn=0, n–номер крана). Если сработает датчик нижнего уровня, то соответствующий выпускной кран должен быть закрыт.

Датчик температуры (DT) использует термосопротивление, величина которого зависит от температуры. Датчик проградуирован так, что при температуре 10 С0 он выдает 0,1 В, а при температуре 200 С0 – 2,0В.

После включения нагревателя устройство должно поддерживать температуру в диапазоне от Тмин до Тмакс. В исходном состоянии резервуары пуст, все краны закрыты, нагреватель и двигатель выключены.

Технологический цикл должен начинаться после нажатия кнопки «Пуск». По завершении цикла подать звуковой сигнал частотой 1 – 2 кГц длительностью 2 с.

**2 Программное обеспечение**

**2.1 Алгоритм функционирования**

Открыть кран К1 на 30 с.

Открыть кран К2 до заполнения резервуара. Р

Запустить электродвигатель на 5 мин. Одновременно включить нагреватель.

Выключить нагреватель. Слить жидкость из Р.

Открыть кран К3 до заполнения Р водой.

Включить двигатель и перемешивать в течение 3 мин.

Слить воду.

Цикл завершен.

Даны значения Тмин=85, Тмакс=90. Режим работы светодиодных цифровых индикаторов – динамический.174-184

Алгоритм функционирования робота представлен на Рисунке 5.

Описание алгоритма:



Рисунок 5. Блок схема алгоритма

**2.2 Описание программы**

Функция setup () главная функция.

Функция loop() обрабатывает сообщение старта и запускает цикл.

Инициализируем порты

pinMode(5, INPUT); //dh – на ввод. Датчик высокого уровня

pinMode(6, INPUT); //dl - на ввод. Датчик низкого уровня

pinMode(7, INPUT); //start на ввод кнопка «START»

pinMode(8, OUTPUT); //k1 на вывод, кран К1

pinMode(9, OUTPUT); //k2 на вывод, кран К2

pinMode(10, OUTPUT); //k3 на вывод, кран К3

pinMode(12, OUTPUT); //ten на вывод для нагревательного элемента

pinMode(13, OUTPUT); //dvig на вывод для управления двигателем

pinMode(3, OUTPUT); //tone на вывод спикера

Открыть кран К1 на 30 с.

Открыть кран К2 до заполнения резервуара. Р

Запустить электродвигатель на 5 мин. Одновременно включить нагреватель.

Выключить нагреватель. Слить жидкость из Р.

Открыть кран К3 до заполнения Р водой.

Включить двигатель и перемешивать в течение 3 мин.

Слить воду.

Цикл завершен.

Даны значения Тмин=85, Тмакс=90. Режим работы светодиодных цифровых индикаторов – динамический.174-184

**2.3 Технология подготовки и отладки ПО**

Разработка программы для микроконтроллера производилась в среде Среда разработки Arduino

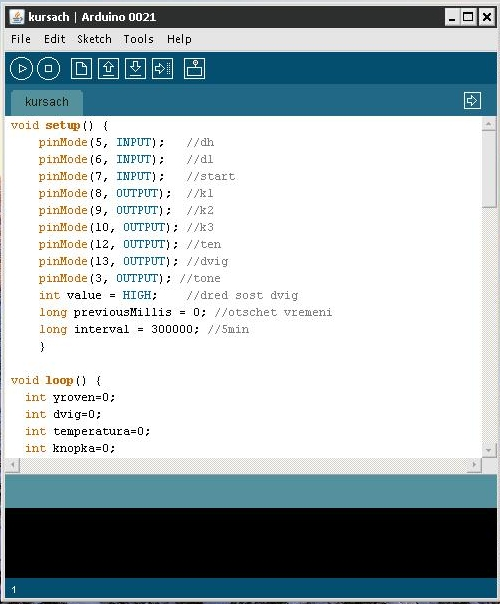


Рисунок 6 . среда разработки Arduino 0021

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:

Verify/Compile

Проверка программного кода на ошибки, компиляция.

Stop

Остановка мониторинга последовательной шины(Serial monitor) или затемнение других кнопок.

New

Создание нового скетча.

Open

Открытие меню доступа ко всем скетчам в блокноте. Открывается нажатием в текущем окне.

Примечание: из-за наличия ошибки в Java данное меню не может прокручиваться; при необходимости открыть скетч из этого списка проследуйте в меню File | Sketchbook.

Save

Сохранение скетча.

Upload to I/O Board

Компилирует программный код и загружает его в устройство Arduino. Описание загрузки приведено ниже.

Serial Monitor

Открытие мониторинга последовательной шины (Serial monitor).

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

Edit

Copy for Discourse

Копирует в буфер обмена подходящий для размещения на форуме код скетча с выделением синтаксиса.

Copy as HTML

Копирует код скетча в буфер обмена как HTML код, для размещения на веб-страницах.

Sketch

Verify/Compile

Проверка скетча на ошибки.

Import Library

Добавляет библиотеку в текущий скетч, вставляя директиву #include в код скетча. Подробная информация в описании библиотек ниже (Libraries).

Show Sketch Folder

Открывает папку, содержащую файл скетча, на рабочем столе.

Add File...

Добавляет файл в скетч (файл будет скопирован из текущего места расположения). Новый файл появляется в новой закладке в окне скетча. Файл может быть удален из скетча при помощи меню закладок.

Tools

Auto Format

Данная опция оптимизирует код, например, выстраивает в одну линию по вертикали открывающую и закрывающую скобки и помещает между ними утверждение.

Board

Выбор используемой платформы. Список с описанием платформ приводится ниже.

Serial Port

Меню содержит список последовательных устройств передачи данных (реальных и виртуальных) на компьютере. Список обновляется автоматически каждый раз при открытии меню Tools.

Burn Bootloader

Пункты данного меню позволяют записать Загрузчик (Bootloader) в микроконтроллер на платформе Arduino. Данное действие не требуется в текущей работе с Arduino, но пригодится, если имеется новый ATmega (без загрузчика). Перед записью рекомендуется проверить правильность выбора платформы из меню. При использовании AVR ISP необходимо выбрать соответствующий программатору порт из меню Serial Port.

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 расширения), файлами С (расширение \*.с), файлами С++ (\*.срр) или головными файлами (.h).

Загрузка скетча в Arduino

Перед загрузкой скетча требуется задать необходимые параметры в меню Tools > Board и Tools > Serial Port. 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 > Import Library. Одна 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 находится в подпапке /libraries/DateTime папки блокнота.

Смотрите данную инструкцию для написания собственной библиотеки.

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", т.к. могут быть перезаписаны встроенные данные платформы Arduino.) Для деинсталляции данных удаляется соответствующая директория.

Подробная информация по созданию сборок описаний аппаратных средств других производителей находится на страницах сайта Google Code.

Мониторинг последовательной шины (Serial Monitor)

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.

Имеется возможность обмена информацией с платформой через программы Processing, Flash, MaxMSP и т.д. (см. подробности на странице описаний интерфейсов).

**Заключение**

При выполнении данного курсового проекта была изучена архитектура микроконтроллера ATmega132 фирмы AVR и построена на его основе платформа Arduino. Приобретены навыки программирования микропроцессорных систем и построения на их основе систем управления химико-технологическим процессом.

**Список использованных литературных источников**

1. Евстифеев А.В. «Микроконтроллеры AVR семейства Mega» – Москва – Издательский дом «Додэка - ХХI», 2007.-595с.
2. Б.Ф. Бессарабов, В.Д. Федюк, Д.В. Федюк Справочник "Диоды, тиристоры, транзисторы и микросхемы широкого применения"- Изд. «Воронеж», 1994-320с.

**Приложение А**

void setup() {

pinMode(5, INPUT); //dh

pinMode(6, INPUT); //dl

pinMode(7, INPUT); //start

pinMode(8, OUTPUT); //k1

pinMode(9, OUTPUT); //k2

pinMode(10, OUTPUT); //k3

pinMode(12, OUTPUT); //ten

pinMode(13, OUTPUT); //dvig

pinMode(3, OUTPUT); //tone

int value = HIGH; //dred sost dvig

long previousMillis = 0; //otschet vremeni

long interval = 300000; //5min

}

void loop() {

int yroven=0;

int dvig=0;

int temperatura=0;

int knopka=0;

knopka = digitalRead(7); // узнаём состояние кнопки

if(knopka==HIGH)

{

//1

digitalWrite(8, LOW);

delay(30000);

digitalWrite(8, HIGH);

//2

yroven = digitaRead(5);

while ( yroven = 1)

{digitalWrite(9, LOW);

yroven = digitaRead(5);

}

digitalWrite(9, HIGH);

//3

digitalWrite(13, LOW); //vkl dvig

dvig = 1;

while (dvig = 1)

{

temperatura = analogRead(2);

if (temperatura > 184)

digitalWrite(12, HIGH); //vikl nagrev

if (temperatura < 174)

digitalWrite(12, LOW); //vkl nagrev

if (millis() - previousMillis > interval) {

previousMillis = millis();

dvig = 0;

}

}

digitalWrite(13, HIGH); //vikl dvig

//4

digitalWrite(12, HIGH); //vikl nagrev

yroven = digitaRead(6);

while (yroven = 1){

digitalWrite(8, LOW);

yroven = digitaRead(6);

}

digitalWrite(8, HIGH);

//5

yroven = digitaRead(5);

while (yroven = 1){

digitalWrite(10, LOW);

yroven = digitaRead(5);

}

digitalWrite(10, HIGH);

//6

digitalWrite(13, LOW); //vkl dvig

delay(18000);

digitalWrite(13, HIGH); //vikl dvig

//7

yroven = digitaRead(6);

while (yroven = 1){

digitalWrite(8, LOW); //otkr k1

yroven = digitaRead(6);

}

digitalWrite(8, HIGH); //zakr k1

digitalWrite(3, HIGH);

delay(2000);

digitalWrite(3, LOW);

}

}