

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Политехнический институт: Заочный  
Кафедра «Системы автоматического управления»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

\_\_\_\_\_/ В.И. Ширяев

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Программное обеспечение визуализации и сбора данных для системы управления линией  
по производству жаренных семян подсолнечника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ – 09.03.01.2018.313.00 ПЗ ВКР

Руководитель работы

нач. отд. автомат. ИП Волков А.Е.

\_\_\_\_\_/ А.А. Привалов

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Автор работы

студент группы ПЗ-597

\_\_\_\_\_/ Р.К. Зинатуллин

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Нормоконтролер

доцент кафедры САУ, к.т.н.

\_\_\_\_\_/ В.О. Чернецкий

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Челябинск 2018

## АННОТАЦИЯ

Зинатуллин Р.К. Разработка программного обеспечения визуализации и сбора данных для системы управления линией по производству жаренных семян подсолнечника - Челябинск: ЮУрГУ, САУ, 2018, 61 с., 11 ил., 3 прил., библиографический список – 10 наим., 10 слайдов презентации.

В данной выпускной квалификационной работе рассматривается процесс создания программного обеспечения для визуализации и сбора данных системы управления линией по производству жаренных семян подсолнечника. Программное обеспечение представляет собой комплекс программ, состоящей из серверной и клиентской части, позволяющей взаимодействовать по каналам связи с микроконтроллером, для отображения состояния всех управляемых им узлов, сбора данных и хранения полученной информации в базе данных.

Цель работы – отслеживание и логирование всех событий на всех узлах производственной линии, с любого устройства, удовлетворяющего требованиям, предъявляемых к средствам отображения информации в условиях промышленной автоматизации. В дальнейшем планируется доработка программы, путем включения новых возможностей (отправка сообщений на почтовый адрес, смс уведомления).

Также выполнено описание комплекса программ, полученных в результате выполнения дипломного проекта.

В заключительной части сделаны выводы по эффективности разработки и полученных результатов от внедрения данного комплекса систем.

					<i>09.03.01.2018.313.00 ПЗ</i>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>			
<i>Разраб.</i>		<i>Зинатуллин Р.К.</i>			<i>Программное обеспечение визуализации и сбора данных для системы управления линией по производству жаренных семян подсолнечника</i>	<i>Лит.</i>	<i>Лист</i>
<i>Провер.</i>		<i>Привалов А.А.</i>				<i>Д</i>	<i>4</i>
							<i>60</i>
<i>Н. Контр.</i>		<i>Чернецкий В.О.</i>				<i>ЮУрГУ Кафедра САУ</i>	
<i>Утверд.</i>		<i>Ширяев В.И.</i>					

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ .....	9
1.1 Описание предметной области .....	9
1.2 Понятие и обзор современных представлений о микроконтроллерах .....	11
1.3 Преимущества разрабатываемого комплекса .....	18
1.4 Обзор аналогов .....	19
1.5 Выбор средств и технологий .....	22
1.6 Анализ необходимости использования на производстве .....	26
2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА .....	28
2.1 Теоретическое обоснование .....	28
2.2 Требования системы к рабочему месту оператора .....	30
2.3 Требования системы к серверной части .....	31
2.4 Описание протокола взаимодействия .....	32
2.5 Описание процесса подключения .....	33
2.6 Краткое описание технологического процесса.....	33
3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	39
3.1 Взаимодействие клиентской части системы с серверной.....	36
3.2 Состав и функционирование компонентов серверной части системы....	37
3.3 Взаимодействие микроконтроллера и серверной части системы.....	38
ЗАКЛЮЧЕНИЕ .....	39
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	40
ПРИЛОЖЕНИЕ А .....	41
ПРИЛОЖЕНИЕ Б .....	56
ПРИЛОЖЕНИЕ В .....	60

## ВВЕДЕНИЕ

Автоматизация производства всегда была одной из основных составных ускорения научно-технического прогресса в агропромышленном комплексе. В 70-80 годах она приобрела новые черты в связи с высокими темпами развития технических средств – микропроцессорной техники и персональных электронных вычислительных машин (ПЭВМ), функциональные возможности которых дают возможность использовать самые совершенные методы в рамках современных сложных систем управления.

Микропроцессорные устройства и ПЭВМ, связанные между собой вычислительными и управляющими сетями с использованием общих баз данных, позволяют внедрять компьютерные технологии в нетрадиционные сферы производства, что проявляется в интеграции производственных процессов и управлении ими. Таким образом, главным направлением автоматизации в агропромышленном комплексе на современном этапе является создание компьютерно-интегрированных производств. За границей эта концепция приобрела название CIM (Computer Integrated Manufacturing System).

Основой современных систем автоматизации стали функциональные возможности микропроцессорных систем управления, при создании которых решающую роль играют такие факторы, как использование принципов интеграции распределенного управления программных комплексов.

Следует добавить, что при автоматизации производства объектом является не отдельный технологический процесс или агрегат, а технологический комплекс со сложными взаимосвязями между его подсистемами.

Современные системы автоматизации на базе микропроцессорных устройств и ПЭВМ имеют широкие функциональные возможности и совершенные технологические характеристики, которые обеспечивают повышение надежности и живучести, производительность, оперативность управления, повышения количества входов-выходов, улучшения условий работы оператора.

Расширение функциональных возможностей современных микропроцессорных систем управления связано со значительным увеличением количества видов и систем отображения технологической информации: использование динамических мнемосхем, получение графиков изменения технологических параметров за любой отрезок времени, формирование предыстории развития процесса, архивирования с помощью таблиц, отчетных документов и т. д. все это дает возможность повышать оперативность управления, максимально учитывая производственную ситуацию.

					09.03.01.2018.313.00 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

Специалисты любого профиля работают в новом информационном пространстве, который требует от них знаний, необходимых при работе с автоматизированным оборудованием, автоматизированных технологических комплексах, использовании компьютерных технологий.

Применение МК можно разделить на два этапа: первый - программирование, когда пользователь разрабатывает программу и прошивает ее непосредственно в кристалл, и второй - согласование спроектированных исполнительных устройств с запрограммируемым МК. Значительно облегчают отладку программы на первом этапе - симулятор, который наглядно моделирует работу микропроцессора. На втором этапе для отладки используется внутрисхемный эмулятор, который является сложным и дорогим устройством, зачастую недоступным рядовому пользователю.

В тоже время в литературе мало уделено внимания вопросам обучения программированию некоторых недорогих МК, в сочетании с реальными исполнительными устройствами.

Разработка макета программатора отличающегося простотой, наглядностью и низкой себестоимостью, становится необходимой как для самого программирования кристаллов, так и для наглядного обучения широкого круга пользователей основам программирования МК.

С усложнением технологического процесса приходится применять современные компьютерные технологии в виде промышленных логических контроллеров и локальных сетей. Контроллеры Siemens Simatic S7-300 отвечают всем требованиям современности в области обеспечения контроля и автоматизации. Созданные на их основе локальные сети способны взять под контроль весь комплекс промышленных объектов, значительно облегчая персоналу в поиске неполадок и их дальнейшего устранения.

Для мониторинга и слежения за всеми узлами сети существуют панели оператора Simatic HMI, представляющие собой отдельно стоящие устройства с дисплеями. Большим минусом подобных устройств является их ограниченность и доступность только в пределах их зон нахождения.

Решение данной проблемы может быть получено при помощи разрабатываемой программной системы, к которой можно получить доступ, практически с любого персонального компьютера, или мобильного устройства, удовлетворяющего заданным требованиям. Разрабатываемая система позволяет вести опрос датчиков, используя заложенные алгоритмы, вести необходимое оповещение персонала и журнал сообщений, используя базу данных Microsoft SQL Server.

					09.03.01.2018.313.00 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

# 1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

## 1.1 Описание предметной области

На данный момент степень сложности линии промышленного производства на предприятии. зависит в первую очередь от производственной мощности самого завода, степени сложности изготавливаемого продукта, а также от методов производства. Применительно к нашему предприятию можно отметить тот факт, что технологические линии производства состоят в основе своей из конвейеров, двигателей различных мощностей постоянного и переменного токов, компрессорных подстанций, всевозможных вентиляционных каналов, а также шкафов управления этими линиями.

Анализ современного технического состояния нашего предприятия показал, что на данный момент для управления и контроля производством до сих пор используется, к сожалению, обычная коммутационная аппаратура либо морально устаревшие контроллеры, которые являются громоздкими и не эффективными в плане обслуживания. Они не способны в полной мере производить гибкий контроль и управление производственным объектом, а с усложнением линий создают порой серьезные трудности при ликвидации аварийных ситуаций. Выходом из сложившейся ситуации явилось, недавнее внедрение и применение современных промышленных контроллеров Siemens SIMATIC S7-300, способных в полной мере удовлетворить требованиями современного типа производства, а также обеспечить возможность их использования для управления из единых координационных центров с целью сбора оперативной информации и управления.

Независимо от характеристик начинки контроллера для управления его функциями обязательно предусмотрен и операторский пункт с соответствующим реле. Внешне такие устройства напоминают небольшой компьютер, обеспеченный устройствами ввода и вывода, датчиками технологических процессов и дисплеем. Самые простые контроллеры для автоматизации производства предусматривают возможность программирования через данную панель. Причем под программированием могут подразумеваться элементарные установки команд начального уровня. Наиболее сложные операторские панели также выполняют самодиагностику и само калибровку.

На нашем предприятии используются новые базовые панели Simatic HMI. Данные устройства предлагают высокое качество Simatic и – независимо от размеров дисплея – многочисленные стандартные функции программного обеспечения, например, систему сообщений, управление на основе рецептов, функции управления кривыми и переключение языков. Поэтому сотрудники

					09.03.01.2018.313.00 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

предприятия могут и в случае простых приложений получить выгоду от преимуществ визуализации за счет улучшения качества процесса.

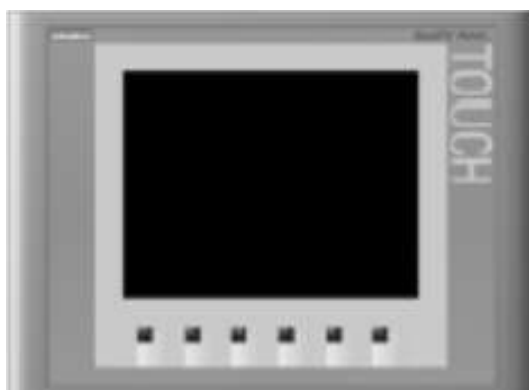


Рисунок 1 – Панель оператора КТР400 Basic

Данное устройство управления дает возможность графически отображать рабочие состояния, текущие данные процесса и неисправности подключенного ПЛК и удобно управлять и наблюдать за подлежащим контролю конвейером или установкой. Эти устройства облегчают управления конвейерами и контролируют их работу и дают реалистическое графическое представление о конвейере или установке, подлежащей контролю. В нашем случае они используются в упаковочной промышленности. Высокая степень защиты (IP65 спереди) и отказ от использования жестких дисков и вентиляторов обеспечивают пригодность этих устройств также для использования в тяжелых промышленных условиях, непосредственно на соответствующем конвейере. Эти устройства могут монтироваться в распределительных шкафах и пультах, в 19-дюймовых шкафах и стойках.

Многофункциональность поддерживается возможностями присоединения внешних периферийных устройств (клавиатуры, мыши, принтера), например через интерфейс USB, и возможностью использования плат CF и PC.

Данное устройство управления используется на предприятии для выполнения следующих функций:

- а) для наблюдения и управления процессом посредством системы меню. Мы можем, например, устанавливать заданные значения или управлять исполнительными элементами, вводя значения или активизируя спроектированные функциональные клавиши;
- б) для отображения процессов, машин и установок в виде полностью графических, динамических изображений;
- в) для отображения и редактирования сообщений и переменных процесса, например, в виде полей вывода, гистограмм, кривых, или индикаторов состояния;
- г) для непосредственного вмешательства в ход процесса через ввод.

					09.03.01.2018.313.00 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

График, тексты, функции, определенные сотрудником предприятия, и элементы управления и отображения, которые должны быть представлены на устройствах управления, сначала должны быть созданы на используемом для проектирования компьютере (ПК или УП) с помощью программного обеспечения для проектирования Simatic ProTool CS. Используемый для проектирования компьютер должен быть подключен к устройству управления, чтобы загрузить в него проект. Соединение может быть создано или последовательно, через сеть MPI/PROFIBUS-DP, через интерфейс USB или Ethernet, или через стандартную модемную линию.

После успешной загрузки проекта нужно подключить устройство управления к ПЛК. После этого устройство управления может обмениваться информацией с ПЛК и реагировать в соответствии с информацией, спроектированной для выполнения программы в ПЛК.

Ниже на рисунке 2 изображен этап проектирования:

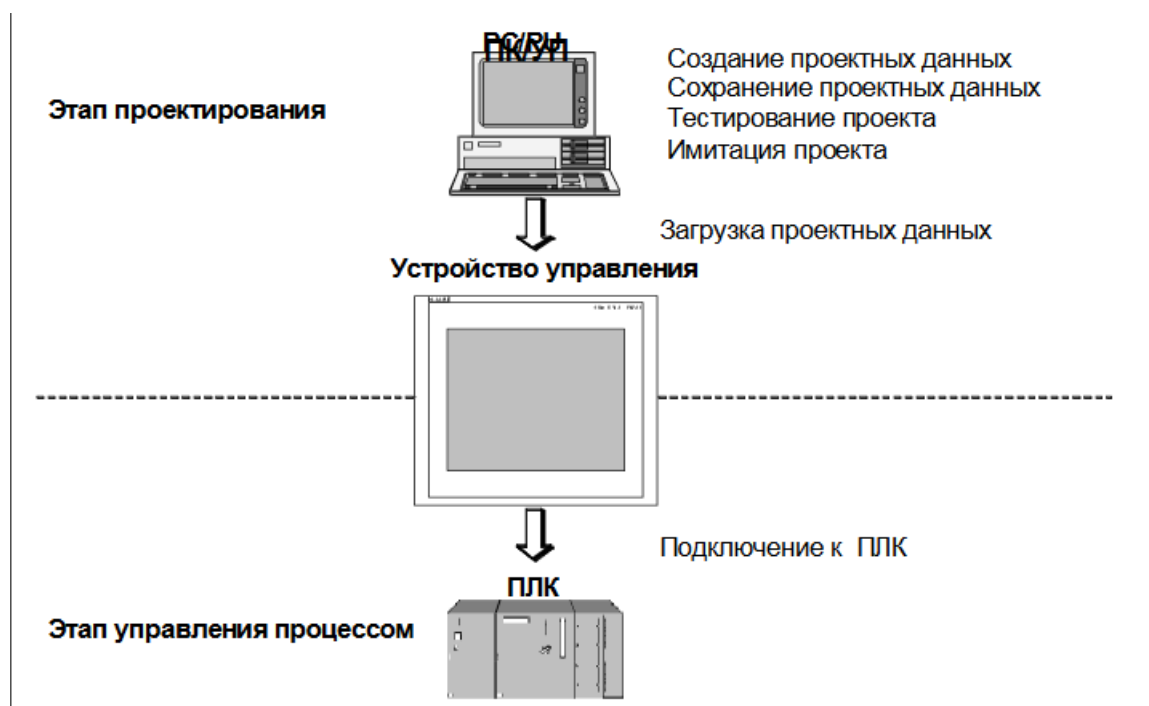


Рисунок 2 – Этапы проектирования и управления процессом

Разработанное мной программное обеспечение визуализации и сбора данных для системы управления линией, является аналогом вышеупомянутых базовых панелей, и работает в качестве Web-службы, что позволяет наблюдать за работой процессов, машин и установок в виде полностью графических изображений, с любого рабочего компьютера или ноутбука в пределах нахождения в локальной сети организации.

## 1.2 Понятие и обзор современных представлений о микроконтроллерах



Технологически контроллеры Siemens SIMATIC S7-300 имеют модульный тип исполнения, что позволяет разработчикам не ограничиваться определенным набором линий ввода-вывода, как это сделано в других контроллерах, а наращивать их число в зависимости от количества задействованных элементов. Максимальное число модулей варьируется в зависимости от модели контроллера [9].

На рисунке 3 изображены элементы контроллера Siemens SIMATIC S7-300:

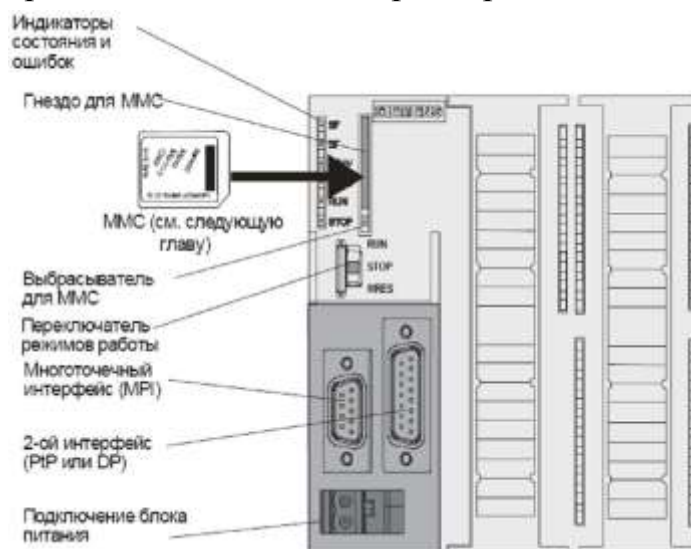


Рисунок 3 – Элементы и структуры контроллера Siemens Simatic S7-300

CPU и модули ввода-вывода, обычно, располагаются справа. CPU имеет удобную светодиодную индикацию, сигнализирующую о состоянии процессора (рисунок 4).

Индикаторы CPU:

SF	(красный)	Аппаратная или программная ошибка
BF	(красный)	Ошибка шины (только CPU с интерфейсом DP)
DC5V	(зеленый)	Питание 5 В для CPU и шины S7-300-в порядке.
FRCE	(желтый)	Активно задание на принудительную установку.
RUN	(зеленый)	CPU в состоянии RUN; светодиод мигает при запуске с частотой 2 Гц; при останове (HOLD) – с частотой 0,5 Гц
STOP	(желтый)	CPU в состоянии STOP, HOLD или в режиме запуска; светодиод мигает при запросе на общее стирание с частотой 0,5 Гц, во время стирания – с частотой 2 Гц

Рисунок 4 – Индикаторы состояния и ошибки CPU

С помощью одного из трех доступных языков программирования (LAD, STL и FBD) создается программа, которая после отлаживания записывается в CPU.

Для хранения программы используются обычные карты памяти MMC, что дает возможность при необходимости увеличить ее размер, тем самым загрузив более громоздкую программу. В общем случае, при загрузке проекта на карту на ней сохраняются блоки программы пользователя, архивы и рецепты, проектные данные, а также данные для сохранения и обновления операционной системы.

Являясь центральным узлом в модульной сборке всего контроллера, CPU обладает широкими коммуникативными возможностями. В зависимости от

конкретной модели частота счета CPU лежит в пределах от 10 до 60 кГц. Данного предела достаточно, так как промышленные процессы в большинстве своем медленные.

Для приема управляющих сигналов от всевозможных датчиков и кнопок CPU снабжается цифровыми и аналоговыми модулями ввода-вывода.

Цифровые модули ввода предназначены для приема сигналов от цифровых элементов управления или кнопок без необходимого преобразования уровней. Физически цифровой модуль представляет шину из определенного количества входов. Все входы потенциально развязаны, что позволяет использовать значительные входные напряжения, а защита от перенапряжений и индуктивная защита позволяет подключать на входах катушки реле и контакторы. На рисунке 5 изображен один из примеров организации цифрового модуля ввода.

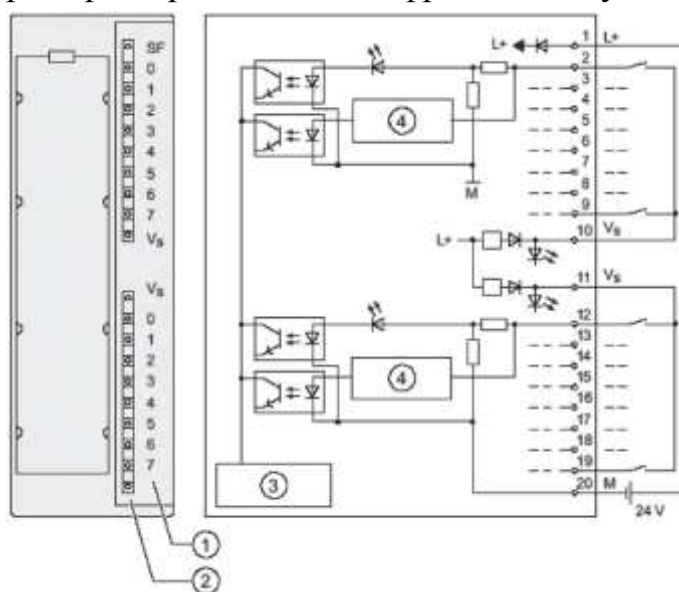


Рисунок 5 – Схема подключения цифрового модуля ввода

На нем: 1- номер канала; 2 - Индикаторы состояния (горят либо зеленым, либо красным); 3 - Подключение к задней шине; 4 - Распознавание обрыва провода.

Цифровые модули вывода по функциональному назначению предназначены для прямого управления исполнительными механизмами цифрового и аналогового типа через промежуточное реле, рассчитанное на напряжение срабатывания 24 вольта (выходное напряжение цифрового модуля).

Для работы с аналоговыми сигналами необходимо подключать аналоговые модули ввода и вывода. Среди их основных функций можно обозначить ввод аналоговых величин, их сравнение и измерение. В зависимости от модификации существуют 15, 13 и 12 разрядные аналоговые корпуса.

Как указывалось выше, CPU имеет широкие коммуникационные возможности. Функционирование систем управления отныне зависит не только от

программируемых логических контроллеров, но также в сильной степени определяется и окружением, в котором они находятся. И кроме средств визуализации, управления и контроля здесь также имеется в виду высокопроизводительная коммуникационная система. Семейство промышленных сетей SIMATIC имеет открытую гетерогенную коммуникационную систему, предназначенную для использования на всех уровнях иерархии систем автоматизированного управления в условиях промышленного производства.

Коммуникационные системы SIEMENS SIMATIC NET базируются на государственных и международных стандартах в соответствии с 7-уровневой моделью ISO/OSI (Open System Interconnection).[2] Основой таких коммуникационных систем служат локальные вычислительные сети (ЛВС), которые могут быть реализованы как:

- а) электрические
- в) беспроводные
- г) комбинация электрических/оптических/беспроводных
- д) электрические искробезопасные

В состав сетевого окружения SIMATIC входят следующие компоненты:

- 1) Коммуникационная сеть, состоящая из среды передачи, средств для подключения к сети и передачи данных, а также соответствующие технологии передачи;
- 2) Протоколы и службы, используемые для передачи данных между перечисленными выше устройствами;
- 3) Модули, предназначенные для подключения программируемого контроллера или компьютера к ЛВС (коммуникационные процессоры “СР” или “интерфейсные модули”).

Для объединения шкафов управления технологической линии производства в сеть используются сети Industrial Ethernet, Profibus-DP и АС-интерфейс. На рисунке 6 изображена иерархия сетей Siemens SIMATIC.



Рисунок 6 – Иерархия сетей Siemens Simatic

К особенностям Industrial Ethernet можно отнести стандарт Ethernet 10 BASE5/1, введенный еще в 1985 году. Стандарт базируется на использовании в качестве среды передачи коаксиального кабеля. Свойством таких сетей является не модулированная передача данных и метод доступа CSMA/CD. Среда распространения сигнала формирует единый канал связи, ресурсы которого должны использоваться одновременно всеми подключенными терминальными устройствами (ООД). Все подключенные терминальные устройства принимают передаваемую информацию одновременно. В любое время правом на передачу данных обладает лишь одно терминальное устройство. Если несколько терминальных устройств передают данные одновременно, в канале связи возникает коллизия. Сигналы терминальных устройств, пытающихся передавать данные одновременно, подавляют друг друга.

Терминальные устройства снабжены механизмом, который позволяет им обнаруживать такие коллизии. Все терминалы, оказавшиеся участниками коллизии, прекращают передачу и в течение некоторого времени, величина которого случайна и рассчитывается для каждого отдельного терминала по-разному, вновь предпринимают попытку передачи данных. Так повторяется до тех пор, пока один из терминалов не добьётся успешной передачи в отсутствии коллизий. Другие терминалы ожидают освобождения канала связи.

Техника доступа CSMA/CD функционирует без ошибок в сети Ethernet, протяжённость которой ограничена максимальным допустимым временем распространения пакета данных. Расстояние, в пределах которого протокол CSMA/CD работает корректно, называют доменом обнаружения коллизий или "коллизийным доменом". В классических сетях Ethernet (10 Мбит/с) домен обнаружения коллизий имеет протяжённость 4520 м.

В качестве среды передачи для построения сетей Industrial Ethernet используются витая пара и волоконно-оптический канал связи. Витая пара и оптический канал позволяют устанавливать соединения «точка-точка» между двумя электрически активными компонентами. Это означает, что между ООД и портом сетевого компонента всегда устанавливается прямая связь (прямой канал). Сетевой компонент осуществляет усиление принятых сигналов и их дальнейшую передачу через все свои выходные порты. В сетях Industrial Ethernet семейства SIMATIC NET эти задачи выполняются такими сетевыми компонентами, как OLM, ELM, OSM и ESM, а при использовании оптоволокну только компонентом OLM. Максимальная длина канала для витой пары, связывающего ООД и сетевой компонент (известная как длина сегмента.) не может превышать 100 м., трехпроводного коаксиального кабеля 50 м.

АС-интерфейс, необходимый для подключения датчиков и исполнительных механизмов, называемый сокращённо AS-i, является коммуникационной системой, предназначенной для использования на самом нижнем уровне иерархии промышленного автоматизированного комплекса – уровне управляемого процесса. Непременный атрибут этого уровня – развитая сеть соединительных кабелей, замещается одним единственным кабелем AS-интерфейса. С помощью AS-i кабеля и ведущего устройства AS-интерфейса простейшие бинарные датчики и исполнительные устройства могут подключаться к средствам управления на полевого уровня посредством модулей AS-интерфейса.

Отличительными чертами AS-интерфейса являются следующие основные характеристики:

- AS-интерфейс оптимален для подключения бинарных датчиков и исполнительных механизмов. Кабель AS-i используется как для обмена данными между датчиками/исполнительными механизмами (ведомыми устройствами AS-i) и ведущим устройством AS-i, так и для подачи напряжения питания на датчики/исполнительные механизмы;

- Более простой и экономичный монтаж соединений. Благодаря использованию метода прокалывания изоляции упрощается монтаж кабеля и достигается высокая гибкость, необходимая для построения древовидной топологии;

- Малое время реакции: ведущему устройству AS-i требуется не более 5 мс для циклического обмена данными с 31 узлом сети;

- В качестве узлов (AS-i ведомых) кабеля AS-интерфейса могут выступать либо датчики/исполнительные механизмы со встроенным AS-i интерфейсом, либо модули AS-i, к которым можно подключить до 4 обычных бинарных датчиков/исполнительных механизмов;

					09.03.01.2018.313.00 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

– При использовании стандартных AS-i модулей на кабеле AS-i может находиться до 124 исполнительных механизмов/датчиков;

– Если используются AS-i модули с расширенным режимом адресации, с одним ведущим устройством, с расширенным режимом адресации могут работать до 186 исполнительных механизмов и 248 датчиков.

Расширенные ведущие устройства AS-интерфейса обеспечивают простой доступ к аналоговым датчикам/исполнительным механизмам или модулям, функционирование которых соответствует профилю ведомых устройств AS-интерфейса 7.3/7.4.

AS-интерфейс обладает следующими характеристиками:

– Технология доступа "Ведущий - Ведомый" (Master - Slave);

– AS-интерфейс является системой с одним ведущим устройством. Это означает, что в сети AS-интерфейса присутствует одно единственное ведущее устройство, которое управляет обменом данными. Это устройство опрашивает поочередно все ведомые устройства AS-i одно за другим, ожидая от каждого ответ;

– Электронная установка адреса

Адрес ведомого устройства AS-i является его идентификатором. Присвоение адреса происходит в системе AS-интерфейса только один раз. Установку адреса можно выполнить либо с помощью специального модуля задания сетевых адресов, или с помощью ведущего устройства AS-i. Адрес постоянно хранится в ведомом устройстве AS-i. При изготовлении устройство по умолчанию всегда записывается адрес "0".

– Надёжность функционирования и гибкость

Используемая техника передачи (модуляция тока) гарантирует высокую эксплуатационную надёжность. Ведущее устройство контролирует напряжение на кабеле, а также передаваемые данные. Оно распознаёт ошибки передачи и выход из строя ведомых устройств и передаёт сообщение на PLC (programming logic controller). Пользователь имеет возможность среагировать на такое сообщение. Замена или добавление ведомых устройств в режиме нормальной работы не окажет влияние на обмен данными с другими ведомыми устройствами AS-i.

PROFIBUS — это коммуникационная сеть полевого уровня и уровня отдельных производственных участков, базирующаяся на стандарте EN 50170-1-2 и использующая гибридный метод доступа к шине (маркерное кольцо между активными узлами и "ведущий - ведомый" между активными и пассивными узлами). Средой передачи может являться витая пара, волоконно-оптический кабель или беспроводная среда. Архитектура протоколов PROFIBUS базируется

на модели OSI (Open System Interconnection). Модель ISO/OSI для коммуникационных стандартов, состоит из 7 уровней, подразделяющихся на два класса:

- ориентированных на пользователя с уровня 5 по уровень 7;
- ориентированных на сеть (уровни 1-4).

Уровни с 1 по 4 описывают пересылку передаваемых данных из одного пункта в другой, в то время как уровни с 5 по 7 предоставляют в распоряжение пользователя доступ к сети в соответствующей форме.

В данном типе сети реализованы первый, второй и седьмой уровни. Благодаря такой архитектуре достигается быстрая передача данных.

Данные передаются внутри 11- разрядного кадра в NRZ-коде (Non Return to Zero). Значения сигнала (биты) не изменяются во время передачи сигнала.

В то время, как передача бинарного значения “1” соответствует положительному значению на проводнике RxD/TxD-P (Receive/Transmit-Data-P), напротив, на проводнике RxD/TxD-N (Receive/Transmit-Data-N) присутствует “0”.

Состоянию покоя между отдельными телеграммами соответствует двоичный сигнал “1”. В литературе часто также оба проводника PROFIBUS обозначают как А-проводник и В-проводник. При этом А- проводник соответствует RxD/TxD-N, а В-проводник - RxD/TxD-P.

Физический уровень [уровень 1] в сетях PROFIBUS позволяет достигнуть электробезопасности и питания полевых приборов прямо через шину. Для передачи данных используется бит-синхронизированный, с манчестерским кодом протокол передачи без постоянной составляющей (обозначается также как H1). При передаче данных с помощью манчестерского кода бинарный “0” передается как смена фронта с 0 на 1, а бинарная “1” . как смена фронта с 1 на 0. используется витой экранированный или неэкранированный провод.

Согласно модели OSI на втором уровне реализуются функции управления доступом к шине, обеспечение безопасности данных, а также выполнение протокола передачи и формирование телеграмм. Уровень 2 обозначается в PROFIBUS как FDL-уровень (Fieldbus Data Link). Формат телеграммы уровня 2 способствует большей безопасности передачи. Вызывающая телеграмма имеет расстояние Хемминга HD (Hamming Distance) = 4. При HD = 4 может быть распознано до 3-х одновременных ошибок в фальсифицированных битах телеграммы данных. Это достигается благодаря применению особых стартового и завершающего знаков телеграммы, постоянно скользящей синхронизации, биту четности и контрольному байту.

Уровень 7 модели ISO/OSI представляет в распоряжение пользователя полезные коммуникационные службы. Этот пользовательский уровень состоит

					09.03.01.2018.313.00 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

PROFIBUS из FMS (Fieldbus Message Specification) и LLI (Lower Layer Interface) слоев.

Проведенный анализ особенностей линейки контроллеров Siemens SIMATIC S7-3xx как ключевого компонента в автоматизации производственных линий и процессов показал, что вариант развития управления промышленности с использованием данных контроллеров является выгодным в плане дальнейшего наращивания производственных мощностей, стабильности и самоорганизации.

Широкие функциональные возможности SIMATIC в плане организации локальных сетей различного уровня позволяют вести контроль и управление производственного процесса при практически любой степени сложности организации производства, достигая тем самым повышения эффективности и конкурентоспособности предприятия в целом. Стоит отметить, что способы автоматизации и дальнейшего управления, основанных на аппаратуре данного класса являются бурно развивающимися в настоящее время во всем мире и заслуживают пристального внимания. Анализ был выполнен с учетом современных достижений в области автоматизации и управления производственных процессов.

### 1.3 Преимущества разрабатываемого комплекса

Когда может потребоваться и зачем вообще нужна разработка программного обеспечения? Этот вопрос достаточно часто задают себе руководители предприятий, где эксплуатируется стандартизированное ПО, выпущенное фирмой разработчиком для удовлетворения запросов как можно большего числа сотрудников и технического процесса.

Однако зачастую возникает потребность сузить или расширить спектр выполняемых программой задач, чтобы она начала соответствовать индивидуальным потребностям конкретной компании или фирмы. В первую очередь это необходимо бизнесу, полная автоматизация задач для которого гарантирует не только удобство работы с ПО, но и повышение производительности труда, то есть существенное увеличение прибыли.

Способность программы максимально быстро и комфортно выполнять возложенную на нее задачу с учетом специфики эксплуатирующей ее организации и имеющейся инфраструктуры является ключевым преимуществом создания программного обеспечения.

Чтобы оценить актуальность и востребованность в разработке программного обеспечения, достаточно озвучить преимущества, получаемые предприятием после ввода решения в эксплуатацию:

а) отсутствие дорогостоящих лицензий на каждое рабочее место;

					09.03.01.2018.313.00 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		



- б) уникальность разрабатываемого решения, с точки зрения выполняющих им функций, так сказать «заточка» под нужды нашего предприятия;
- в) возможность взаимодействия разрабатываемого программного комплекса с типовыми программами, такими как, например SQL Server;
- г) улучшение производительности труда, за счет удобного и доступного средства мониторинга;
- д) доступность к интерфейсу программного средства, практически с любого рабочего места в сети предприятия;
- е) сбор и надежное хранение всех сообщений и событий, произошедших при работе технологического оборудования и их узлов, на долгосрочный период, если, например понадобится в будущем разобраться в произошедших событиях и инцидентах.

Первым делом, эксплуатация изготовленного программного решения избавляет руководство предприятия от существенных затрат на покупку лицензий для каждого компьютера, установленного в офисах компании.

Именно дороговизна типового ПО при наличии большого числа сотрудников, в нем нуждающихся, становится основным критерием для принятия решения о разработке собственного программного решения. Полученный продукт будет не только уникальным, но также учитывающим все нюансы производственного процесса на предприятии, максимально повышая работоспособность организации за счет упрощения и ускорения необходимых операций.

Взаимодействие выполняющих разные функции сотрудников предприятия за счет автоматизации сбора логов и сообщений значительно ускоряет выполнение работы, благотворно влияя на эффективность сотрудников.

#### 1.4 Обзор аналогов

Первым аналогом является Scada система визуализации добычного забоя на поверхности и рабочем месте горного диспетчера.

Система визуализации осуществляет сбор телеметрической информации о работе механизированного комплекса и забойных механизмов. Далее, посредством аппаратуры передачи данных, информация передается на Modbus OPC сервер, входящий в системы, который отвечает за обработку, архивацию и передачу данных для визуализации на SCADA системе, установленной на компьютере горного диспетчера.

Ключевые особенности системы визуализации:

1. Обеспечивает наглядную визуализацию текущего состояния крепи, системы управления крепью и забойных механизмов в режиме реального времени;

					09.03.01.2018.313.00 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

2. Обеспечивает наглядную визуализацию работы системы автоматизированного натяжения лавного конвейера;
3. Обеспечивает возможность просмотра мнемосхем через Web-клиенты;
4. Обеспечивает архивацию выбранной телеметрической информации;
5. В состав системы входит OPC сервер, поддерживающий спецификации OPC UA, OPC DA, OPC HDA;
6. Система осуществляет генерацию отчетов по результатам работы оборудования за выбранные временные интервалы (час, смена, сутки);
7. Система осуществляет рассылку сгенерированных отчетов по электронной почте;
8. Осуществляется архивация сгенерированных отчетов;
9. Имеется возможность сигнализации об авариях путем рассылки сообщений по электронной почте и (или) SMS;
10. Осуществляет построение графиков за выбранный временной интервал (в том числе можно использовать данные из архива);
11. Система способна задавать параметры управления автоматическим до распором секций;
12. Система способна осуществлять управление аппаратурой натяжения цепи лавного конвейера;
13. В состав системы по желанию заказчика включается сервер, осуществляющий «горячее» резервирование основного сервера визуализации.

Вторым аналогом является подсистема визуализации для системы управления технологическими процессами «OpenScada».

Данная подсистема визуализации предназначена для:

1. визуализации;
2. оперативной (текущей) информации: значений параметров технологического процесса (ТП), нарушений параметрами регламентных границ (алармов);
3. архивных данных: значений параметров; текстовых сообщений о нарушениях ведения ТП, действиях пользователя по управлению ТП, а также системных сообщений от всех подсистем OpenSCADA;
4. управления технологическим оборудованием и параметрами ведения ТП: изменения настроек и режимов работы контуров регулирования, изменения настроек системы ПАЗ, ввода числовых данных;

Применение данной подсистемы визуализации обеспечит:

1. оперативное представление информации обслуживающему персоналу о состоянии ТП и оборудования;
2. автоматизированный пуск и остановка технологического оборудования;

					09.03.01.2018.313.00 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

3. контроль действий операторов, управляющих ТП;
4. архивирование параметров ТП;
5. повышение безопасности ведения ТП за счет обеспечения работы технологического оборудования без постоянного присутствия в зоне его размещения эксплуатационного персонала.

Третьим аналогом является система InTouch.

Система InTouch предоставляет широкие возможности для графического представления, визуализации технологического/производственного процесса, что позволяет перевести на качественно новый уровень управление предприятием.

Основные преимущества данной системы:

Интеллектуальные графические возможности и независимые от разрешения экрана графические объекты для создания реалистичной и четкой графики

Разнообразный инструментарий создания сценариев для расширения и настройки приложений в соответствии со спецификацией системы. Система распределенных алармов, ведение истории в реальном времени для анализа. Встроенные тренды – как реального времени, так и исторические. Интеграция элементов управления Microsoft ActiveX и .NET. Обширная библиотека, включающая более 500 предварительно созданных и настраиваемых графических объектов и символов. Поддержка Microsoft Remote Desktop Services, аутентификации с помощью смарт-карт и технологии виртуализации Hyper-V.

InTouch - это приложение-генератор человеко-машинного интерфейса для систем SCADA и других систем автоматизации производства. InTouch дает возможность пользователям создавать операторские интерфейсы под Windows, которые тесно взаимодействуют с другими компонентами программного обеспечения фирмы Wonderware, например FactorySuite (интегрированный пакет программного обеспечения для полной автоматизации производства) и стандартными приложениями Microsoft Office. Это седьмое поколение программных продуктов фирмы Wonderware, лидирующей в своей отрасли и являющейся пионером в использовании Windows для автоматизации производства.

InTouch это программный пакет для быстрой и эффективной разработки и внедрения систем управления производственным процессом.

ООО «НТЦ «Комплексные системы» разрабатывает распределенные технологические информационные системы мониторинга, успешно работающие на ряде предприятий энергетики.

Предлагаемые решения предназначены для осуществления централизованного мониторинга технологических процессов крупных и средних

					09.03.01.2018.313.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

промышленных предприятий, в том числе предприятий с филиалами в разных городах.

Область применения:

- Промышленные предприятия с непрерывным циклом работы.;
- Промышленные предприятия, имеющие в своем составе удаленные объекты;
- Промышленные предприятия, осуществляющие функции диспетчерского контроля;

Возможности:

Система оперативного мониторинга технологических процессов обеспечивает следующие возможности:

- Объединение больших объемов информации, более 100000 точек измерения в одном узле оперативного мониторинга
- Сохранение архивов значений технологических параметров на неограниченном интервале времени.
- Создание распределенных систем. Объединение данных из узлов оперативного мониторинга, располагающихся на разных предприятиях в разных городах.

Особенности:

Создаваемые нами системы имеют своей основой программный комплекс «ДЕЛЬТА/8», разрабатываемый с 1999 года специально для создания масштабируемых информационных систем большой сложности и успешно работающий на крупнейших энергетических предприятиях нашей страны.

Основными достоинствами создаваемых систем являются:

- Гибкая архитектура, позволяющая интегрировать технологическую информацию из различных систем АСУТП, систем коммерческого учета тепла, газа и электроэнергии, телемеханики, диагностики, управления ОРЭ (МОДЭС), а также разнообразных расчетных технологических задач.
- Высокая масштабируемость. Система изначально спроектирована как распределенная, поэтому позволяет объединять на мнемосхемах и в отчетах информацию, которая обрабатывается и хранится на различных (в том числе – территориально удаленных) серверах.
- Полная поддержка открытых промышленных стандартов (OPC DA и OPC HDA, SQL, МЭК60870-5-104), позволяющая легко интегрировать в систему внешние модули и задачи, SCADA-системы и информационные системы других производителей (в том числе ERP и MES).
- Наличие в системе набора модулей для получения, обработки, отображения РДГ, ПБР, УДГ (задача исполнения диспетчерского графика). Данные модули

					09.03.01.2018.313.00 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

позволяют значительно улучшить качество исполнения диспетчерского графика в условиях работы на ОРЭ.

- Высокая эффективность программной реализации коммуникационного ядра системы. Имеются реально работающие внедрения с количеством параметров более 100000 и количеством подключенных АРМ более 100.

- Эффективная реализация архива позволяет хранить без усреднения историю изменения всех измеряемых каналов с глубиной, ограниченной только возможностями используемых аппаратных средств. Есть примеры работающих систем с количеством архивируемых параметров более 100000, интенсивностью записи в архив свыше 50 миллионов записей в сутки. Скорость выборки истории параметра – более 100000 точек в секунду.

- Мощный и производительный графический интерфейс системы предоставляет пользователю широкие возможности по просмотру и анализу информации в виде мнемосхем, графиков, таблиц и отчетов.

- Гибкая система формирования отчетов позволяет формировать качественную отчетную документацию и при необходимости добавлять и модифицировать отчетные формы силами эксплуатационного персонала.

- Максимальное использование стандартных технологий, интерфейсов и протоколов, таких как HTTP, ActiveX, HTML, ASP, JAVA, COM, OPC. Их применение дает возможность облегчить интеграцию отдельных компонентов системы между собой и с внешними системами, упростить эксплуатацию системы обслуживающим персоналом, уже имеющим навыки работы с этими технологиями.

### 1.5 Выбор средств и технологий

Для создания своего программного решения я выбрал среду разработки и визуализации Visual Studio 2017.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода.

					09.03.01.2018.313.00 ПЗ	Лист 23
Изм.	Лист	№ докум.	Подпись	Дата		

Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Редакции:

Visual Studio Express

Набор легковесных сред разработки, представляющих собой урезанную версию Visual Studio. В отличие от полной версии, каждая такая среда предназначена для какого-то одного языка программирования.

Она включает в себя небольшой набор инструментов, в отличие от полных версий: отсутствует дизайнер классов и многие другие инструменты, а также поддержка плагинов и удалённых баз данных в дизайнере данных.

Компиляторы в 64-битный код также недоступны в Express редакциях. Microsoft позиционирует эту линейку IDE для студентов и любителей. На настоящий момент существуют следующие Express редакции:

- Visual Basic Express;
- Visual C++ Express;
- Visual C# Express;
- Visual Web Developer Express;

Visual Studio Standard

Данная редакция предоставляет IDE для всех поддерживаемых продуктов и поддерживает полную версию MSDN Library. Поддерживается редактирование XML и XSLT, как и средства для тестирования объектов.

Однако, отсутствует обозреватель серверов и интеграция с Microsoft SQL Server.

Поддержка разработки под мобильные устройства изначально была включена в Visual Studio 2005 Standard, но в версии 2008 она доступна только в Professional редакции. [править] Visual Studio Professional

					09.03.01.2018.313.00 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

Редакция включает все возможности Standard Edition, расширяя их дополнительными, такими как интеграция с Microsoft SQL Server и поддержка удалённой отладки. [править] Visual Studio Tools for Office

SDK и расширение для Visual Studio, которое содержит утилиты для разработки под платформу Microsoft Office. Начиная с Visual Studio 2008 включено в версии Professional и выше.

#### Visual Studio Team System

Предоставляет набор инструментов для совместной разработки, подсчета метрик и создания отчетов, в дополнение к возможностям Professional редакции.

Имеются различные редакции VSTS, разделяемые по ролям, для которых продукт будет использоваться:

- Team Explorer (клиент для TFS);
- Architecture Edition;
- Database Edition;
- Development Edition;
- Test Edition.

В семействе продуктов Visual Studio используется единая интегрированная среда разработки (IDE), состоящая из нескольких элементов: строки меню, панели инструментов Стандартная, различных закрепленных или автоматически скрываемых окон инструментов в левой, нижней или правой областях, а также области редакторов. Набор доступных окон инструментов, меню и панелей инструментов зависит от типа проекта или файла, в котором выполняется разработка.

Ниже перечислены основные преимущества IDE-среды Visual Studio. Встроенный Web-сервер. Для обслуживания Web-приложения ASP.NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы.

Наличие в Visual Studio интегрированного Web-сервера позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовому Web-сайту с какого-нибудь внешнего компьютера, поскольку тестовый сервер может принимать соединения только с локального компьютера.

Поддержка множества языков при разработке. Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс (IDE). Более того, Visual Studio также еще позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение.

					09.03.01.2018.313.00 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

Единственным ограничением является то, что в каждой Web-странице можно использовать только какой-то один язык (очевидно, что в противном случае проблем при компиляции было бы просто не избежать). Меньше кода для написания. Для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP.NET тому не исключение. Например, добавление Web-элемента управления, присоединение обработчиков событий и корректировка форматирования требует установки в разметке страницы ряда деталей. В Visual Studio такие детали устанавливаются автоматически.

Интуитивный стиль кодирования. По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам.

Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует). Возможности отладки.

Предлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания загадочных ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти. Studio также имеет и множество других функций: возможность управления проектом; встроенная функция управления исходным кодом; возможность рефакторизации кода; мощная модель расширяемости.

Более того, в случае использования Visual Studio 2008 Team System разработчик получает расширенные возможности для модульного тестирования, совместной работы и управления версиями кода (что значительно больше того, что предлагается в более простых инструментах вроде Visual SourceSafe).

В качестве недостатка можно отметить невозможность отладчика (Microsoft Visual Studio Debugger) отслеживать в коде режима ядра. Отладка в Windows в режиме ядра в общем случае выполняется при использовании WinDbg, KD или SoftICE.

Для реализации идеи доступности программного продукта на разных рабочих станциях я выбрал технологию разработки ASP.NET.

					09.03.01.2018.313.00 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		



ASP.NET (Active Server Pages для .NET) — платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт. ASP.NET входит в состав платформы .NET Framework<sup>[2]</sup> и является развитием более старой технологии Microsoft ASP.

Позволяет в значительной степени облегчить процесс создание динамических страниц HTML.

Входит в состав платформы Microsoft .NET и получила свое развитие из более старой версии подобной технологии Microsoft ASP.

Благодаря тому, что внешнее сходство между ASP.NET и более старой версией технологии ASP очень велико, то перейти на работу с более современной технологией разработчику относительно просто.

Но, несмотря на внешнее сходство, внутреннее устройство новой технологии в значительной степени отличается от устаревшей ASP.

В ней содержится огромное количество уже готовых элементов управления, пользуясь которыми можно быстро создавать самые разнообразные интернет-сайты. Помимо этого, можно прибегнуть к всевозможным сторонним сервисам, которые предоставляют другие сайты, не нагружая тем самым лишними функциями свой проект.

Проще говоря, возможности ASP.Net практически безграничны. Использование данной технологии будет полезно как для опытного разработчика, так и для молодого программиста.

Существует огромное количество самых разнообразных курсов и сайтов, где происходит обучение работы с данной технологией. А если вы захотите узнать что-то конкретное, то справочная система MSDN, так же разработанная и созданная компанией Microsoft, в значительной степени облегчит ваши труды.

На данный момент последней версией данного продукта является версия ASP.NET 4.0, но в новой ОС от компании Microsoft присутствует версия 4.5.

Часть написанного мною кода реализована на HTML, это все, что касается визуализации, разметки, стилей, взаимодействие клиентской части и серверной, а также прочих сценариев.

Ядро, или программная логика, в том числе планировщик, службы, классы, взаимодействие серверной части и микроконтроллера, написано на языке C#.

Почему для разработки данной программы был выбран именно пакет Visual Studio?

Visual Studio является рекомендованной средой производителя Windows, поэтому под её компилятор заточены системные библиотеки Windows и далее, по индукции, большинство других библиотек. Поэтому для разработки чисто под

					09.03.01.2018.313.00 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

Windows, при прочих равных условиях, VS является наиболее разумным выбором.

## 1.6 Анализ необходимости использования на производстве

Разрабатываемый программный продукт позволит предприятию, значительно повысить производительность и безопасность, сократив количество потерь.

Данная система обеспечивает сокращение затрат на мониторинг и реагирование на инциденты, что в итоге дает эффективность и экономичность для данной специфики работ.

Программа использует клиент - серверную архитектуру, что подразумевает множество клиентских подключений. Данная технология позволит отслеживать работу всего технологического оборудования и их узлов с разных точек и мест предприятия, что в свою очередь повысит контроль за оборудованием и реагирование на внештатные ситуации. Соответственно выход из строя какого – либо клиентского рабочего места не повлияет на работу остальной части системы, затем в дальнейшем можно разворачивать дополнительные рабочие места, что улучшит масштабируемость и контроль за технологическим оборудованием предприятия.

Многие платные системы, аналогичного назначения позволяют вести запись логов и событий, и хранить их ограниченное время, что может воспрепятствовать в будущем разрешению критических ситуаций. Разрабатываемая мною программа позволяет не только отслеживать сообщения от всех узлов производственной линии, но и хранить эти события в базе данных SQL Server для долгосрочного использования.

Данное программное обеспечение показывает статус каждого узла на производственной линии (запущен, остановлен, недоступен и т.д.), а также показывает, когда этот статус возник, т.е. дату и время появления статуса узла.

Цель разрабатываемого программного обеспечения — сделать рабочий процесс понятнее для всех его участников. С помощью визуализации можно получить максимум информации за минимум времени.

Данный продукт рекомендуется для использования на производстве, потому что – это самый простой и эффективный способ мониторинга и сбора данных о работе технологических узлов, который сразу позволяет оценить состояние производственного процесса и его результативность.

					09.03.01.2018.313.00 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА

### 2.1 Теоретическое обоснование

Система мониторинга и оценки, в значении «уточнение того, что должно оцениваться, кем, как и когда», должна быть создана на этапе планирования или, самое позднее, в самом начале реализации. Основательный анализ проблемы и ее контекста должны быть выполнены в рамках разработки и планирования стратегии и могут служить отправной точкой для последующего мониторинга и оценки. Если такой анализ не проведен, важно провести его на более поздней стадии и внести необходимые изменения в запланированные мероприятия.

Система мониторинга — это способ направления и организации мониторинга таким образом, чтобы на его выполнение уходило меньше времени и усилий. Системы мониторинга различаются по уровню сложности от простой записи на листе бумаги, в нескольких блокнотах или файлах до электронной системы учета и баз данных. Но самое важное заключается не в уровне сложности системы, а в том, собрана ли информация, необходимая для принятия решений, проверена ли она и использовали ли ее для необходимых изменений и доработок.

Удачно спроектированная и организованная система сможет гарантировать сбор правильных данных в правильное время в течение технологического процесса и после того, а также то, что эти данные помогут управлять реализацией и стратегическим принятием решений. Она также будет гарантией того, что сотрудники и все задействованные стороны не будут перегружены количеством собранных данных, а денежные и временные затраты на сбор и анализ, сравнение данных и отчет о них останутся в разумных пределах.

Разрабатываемая программная система собирает всю информацию о работе всех узлов, непосредственно с первоисточника, т.е. с микроконтроллера, что может гарантировать сбор правильных данных в правильное время. Интерфейс программной системы построен таким образом, что все сотрудники и задействованные стороны не будут перегружены количеством собранных данных, а затраты на сравнение данных и отчетов останутся в разумных пределах.

Одним из недостатков, существующих на предприятии, является не своевременное выявление сбоев технологического оборудования, и соответственно запаздывание реагирования на такие действия.

Для сотрудников предприятия, которые занимаются контролем и мониторингом узлов технологической и производственной линии, новая программная система будет представлять собой приложение, которое считывает с базы данных информацию о неполадках и указывает конкретные проблемные узлы в производственной сети. Все работающее оборудование будет обозначено

					09.03.01.2018.313.00 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

зеленым цветом, остановленное – желтым, недоступное – серым, а находящееся в состоянии ошибки – красным.

## 2.2 Требования системы к рабочему месту оператора

Под рабочим местом оператора понимается зона его трудовой деятельности в системе "человек - машина", оснащенная техническими средствами и вспомогательным оборудованием, необходимым для осуществления функций контроля и управления производственным процессом.

Правильная организация условий труда предполагает:

- рациональную планировку оборудования и размещение постоянного рабочего места персонала с учетом психофизиологических характеристик и антропометрических данных;
- компоновка средств отображения информации и органов управления на стендах или столах;
- организация рабочей позы оператора;
- общее оформление помещения операторского пункта с учетом эстетических требований.

Все это обеспечит максимальную надежность и эффективность работы оператора.

При организации рабочего места должны быть соблюдены следующие условия:

- достаточное рабочее пространство для оператора;
- достаточные информационные связи между оператором и оборудованием, а также между операторами;
- оптимальное размещение рабочих мест в помещениях для оперативной работы;
- оптимальное размещение оборудования;
- необходимое естественное и искусственное освещение;
- допустимый уровень акустического шума, вибраций и других факторов производственной среды;
- наличие необходимых инструкций и предупредительных знаков;
- надежная индикация отказов электрического питания и аппаратуры.

При использовании данного программного обеспечения необходимо учитывать специфические требования к локальной сети. Необходимо понимать, что вся информация по сети передается пакетами, каждый пакет, кроме данных содержит служебные биты данных - адрес отправителя, адрес получателя, индексы доставки и целостности информации и многое другое, что заметно влияет на размер сетевого трафика и на общую загрузку сетевого оборудования.

					09.03.01.2018.313.00 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

Если для системы монтируется новая ЛВС, то при её проектировании должны учитываться следующие требования и рекомендации:

- Пропускная способность сети на участке от рабочего места до сервера - не менее 10 Мбит/сек;

Для использования данной программной системы необходим веб браузер. Поддерживаются следующие марки браузеров:

- а) Mozilla Firefox, версия не ниже 50.0;
- б) Google Chrome, версия не ниже 61.0;
- г) Яндекс браузер, версия не ниже 18.0.

Минимальные требования к аппаратной части рабочей станции:

- а) процессор - Intel Pentium или AMD K10;
- б) оперативная память – 1 Гб;
- в) Жесткий диск - 32 Гб;
- г) разрешение монитора – 1024 x 768;
- д) операционная система Windows 7,8.

Рекомендуемые требования к аппаратной части рабочей станции:

- а) процессор - Intel Core i5 или AMD Athlon X4;
- б) оперативная память – 4 Гб;
- в) Жесткий диск - 64 Гб;
- г) разрешение монитора – 1600 x 900;
- д) операционная система Windows 10.

Перед началом эксплуатации (открытия страницы в веб браузере) сайту нужно некоторое время (5-10 с.), чтобы прогрузиться, последующие открытия будут происходить быстрее, за счет кэширования статических объектов со страницы сайта.

### 2.3 Требования системы к серверной части

Требования к компьютеру, на который устанавливается серверная часть, зависят от количества одновременно работающих пользователей (количество зарегистрированных пользователей системы обычно в 2-3 раза больше).

Минимальные требования к аппаратной части сервера:

- а) процессор - Intel Xeon 3GHz или AMD Xeon E5-2609;
- б) оперативная память – 2 - 4 Гб;
- в) Жесткий диск - 128 Гб;
- г) отказоустойчивость – RAID 1;
- д) операционная система Windows Server 2016.
- е) кластер – не требуется.

Несмотря на то, что сервер в минимальной конфигурации обеспечивает полноценную работу системы, необходимо принимать во внимание, что данная архитектура не отвечает требованиям надежности в непрерывном режиме работы, что может привести к скрытым необратимым разрушениям данных в долгосрочной перспективе. Рекомендуем использовать варианты серверов, в рекомендуемой конфигурации.

Рекомендуемые требования к аппаратной части сервера:

- а) процессор - Intel 4xXeon 3GHz или AMD Opteron 6200;
- б) оперативная память – 8 - 16 Гб;
- в) Жесткий диск - 1024 Гб;
- г) отказоустойчивость – 2xRAID 1;
- д) операционная система Windows Server 2016.
- е) кластер – требуется.

Программные требования, предъявляемые к серверной части:

- а) операционная система – 64 разрядная Windows Server 2016;
- б) веб сервер – IIS 10, поддерживающий протоколы HTTP, HTTPS.
- в) сервер баз данных – Microsoft SQL Server 2016, а также для ее обслуживания SQL Server Management Studio 16 или 17.

При использовании данного программного обеспечения необходимо учитывать специфические требования к локальной сети. Необходимо понимать, что вся информация по сети передается пакетами, каждый пакет, кроме данных содержит служебные биты данных - адрес отправителя, адрес получателя, индексы доставки и целостности информации и многое другое, что заметно влияет на размер сетевого трафика и на общую загрузку сетевого оборудования.

Если для системы монтируется новая ЛВС, то при её проектировании должны учитываться следующие требования и рекомендации:

- Пропускная способность сети на участке от сервера до контроллера - не менее 100 Мбит/сек;
- Пропускная способность сети на участке от сервера до рабочей станции - не менее 10 Мбит/сек.

Данная аппаратная система должна размещаться в серверной комнате.

Помещение серверной комнаты не должно быть проходным. Желательно, чтобы оно не имело окон и не примыкало вплотную к внешним стенам здания, запрещается располагать серверную рядом с помещениями для хранения пожароопасных или агрессивных химических материалов. Не рекомендуется выделять помещение для серверной на верхних этажах здания, т. к. они наиболее подвержены повреждениям в случае пожара и могут заливаться при протечках крыши.

					09.03.01.2018.313.00 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

Не допускается размещение серверной под помещениями, связанными с потреблением воды (туалеты, душевые, столовые, буфеты и т. д.). При размещении серверной в подвале необходимы дополнительная гидроизоляция и тщательный выбор трасс прокладки трубопроводов.

Через серверную не должны прокладываться транзитом трубопроводы инженерных систем здания.

Запрещается располагать серверную в помещении, смежном с помещением производств с мокрыми технологическими процессами. Предпочтительно размещать серверную недалеко от грузовых или грузопассажирских лифтов, используемых для транспортировки тяжелого оборудования, например источников бесперебойного питания. В то же время следует избегать близкого размещения мощных источников электрических и магнитных полей, а также оборудования, которое может вызвать повышенную вибрацию.

#### 2.4 Описание протокола взаимодействия

Разрабатываемое программное обеспечение работает в режиме реального времени. Характерной особенностью является коммуникация именно в реальном времени или близкая к этому, что позволяет проводить мониторинг узлов в производственной линии, не теряя контроль, и избегая простоев оборудования.

Передача данных в режиме реального времени и есть главная особенность разрабатываемого программного обеспечения.

Выбор протокола транспортного уровня пал на протокол TCP/IP, по следующим причинам:

- TCP/IP сокет используется для реализации надежных поточных соединений между компьютерами в сети Internet.

- соединение с использованием TCP/IP сокетов является постоянным и определяется в двух направлениях. С помощью TCP/IP сокетов можно программировать подключение систем ввода/вывода к программам, расположенным на любом компьютере в сети.

- помимо этого TCP/IP сокет позволяет реализовать подключение и к локальной машине.

- зачастую на практике для соединения по TCP/IP сокетам на компьютерах открываются определенные порты, что позволяет расширить и разграничить канал подключения компьютера.

Для осуществления передачи данных между клиентом и сервером используется тот же протокол TCP/IP.

Взаимодействие клиента с сервером требует высокой степени надежности, поэтому необходима гарантированная доставка сообщений. Для этого

					09.03.01.2018.313.00 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

используется протокол, основанный на протоколе TCP. Рассмотрим его основные аспекты.

При подключении клиента устанавливается соединение с сервером. Сервер, получая запрос на соединение, выделяет поток для взаимодействия с пользователем и отправляет ему подтверждение.

После этого клиент отправляет серверу сообщение Authorization, которое сопровождает имя и пароль пользователя, пытающегося подключиться к серверу.

Сервер анализирует полученные данные - проверяет, есть ли пользователь с такими учетными данными в списке зарегистрированных пользователей, не находится ли пользователь с таким именем в сети, и отправляет клиенту сообщение AuthorizationSuccess или AuthorizationFailed в зависимости от результата проверки.

В случае регистрации пользователя, клиент отправляет сообщение Registration, сопровождая его учетными данными. Сервер проверяет, нет ли в списке зарегистрированных пользователей пользователя с таким именем, и отправляет клиенту сообщение RegistrationSuccess или RegistrationFailed в зависимости от результатов проверки.

Клиент, получив сообщение об успешной регистрации или авторизации, отправляет на сервер сообщение RequestSuccess для подтверждения.

Для взаимодействия сервера и микроконтроллера используется коммуникационный протокол DDE.

DDE (Dynamic Data Exchange - динамический обмен данными) представляет собой коммуникационный протокол, разработанный компанией Microsoft для обмена данными между различными приложениями и устройствами. Этот протокол реализует взаимосвязи типа клиент - сервер между двумя одновременно работающими устройствами.

В разрабатываемом программном обеспечении поддерживается также пакетированный DDE - обмен - FastDDE.

Применение последнего заметно повышает эффективность и производительность обмена данными благодаря уменьшению общего количества DDE - пакетов, которыми микроконтроллер и сервер обмениваются между собой.

Но принципиальные недостатки, связанные с надежностью и зависимостью от количества загруженных в текущий момент приложений Windows, остались.

Необходимость в появлении более совершенного технологичного протокола созрела! Но следует отметить, что отказ от DDE-механизма происходит не мгновенно хотя бы потому, что в мире наработано большое количество DDE - серверов.

					09.03.01.2018.313.00 ПЗ	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		



## 2.5 Описание процесса подключения

Подключение сотрудников предприятия к Web сайту осуществляется следующим способом:

- сотрудник должен быть членом домена Active Directory предприятия;
- нужно ввести гиперссылку <http://WebServer.local.ru/Page01.aspx> в поисковом окне браузера;
- сотрудник проходит сквозную авторизацию, с помощью логина и пароля, который он использовал для входа в систему.

## 2.6 Краткое описание технологического процесса

ГОСТ дает научно строгое, но сформулированное слишком сухим и наукообразным языком определение технологического процесса.

Если же говорить о понятии технологического процесса более понятным языком, то технологический процесс — это совокупность выстроенных в определенном порядке операций. Он направлен на превращение сырья и заготовок в конечные изделия.

Для этого с ними совершают определенные действия, обычно выполняемые механизмами. Технологический процесс не существует сам по себе, а является важнейшей частью более общего производственного процесса, включающего в себя в общем случае также процессы контрактации, закупки и логистики, продажи, управления финансами, административного управления и контроля качества.

На сегодняшний день оборудование для обработки семян подсолнечника продолжает расти, потому что пользуется наибольшим распространением вследствие непрерывности процесса, высокой производительности и возможности получения продукции высокого качества.

Технологи на предприятии занимают весьма важное положение.

Они являются своего рода посредниками между конструкторами, создающими идею изделия и выпускающими его чертежи, и производством, которому предстоит воплощать эти идеи и чертежи в металл, дерево, пластмассу и другие материалы.

При разработке техпроцесса технологи работают в тесном контакте не только с конструкторами и производством, но и с логистикой, закупками, финансами и службой контроля качества.

Не следует смешивать технологический процесс с технологией производства.

Под технологией производства нужно понимать не только последовательность выполняемых операций, но также приемы и способы выполнения этих операций.

					09.03.01.2018.313.00 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

Технология производства должна строиться на основе новейших достижений науки и техники, с учетом опыта работы новаторов и рационализаторов.

Место в производстве, на котором выполняется какая-либо производственная операция, называется рабочим местом. Установленные на рабочем месте станки, механизмы, стационарные приспособления, т. е. приспособления постоянные, укрепленные неподвижно, составляют оборудование рабочего места.

От того, как организовано рабочее место, от обеспеченности его инструментами и приспособлениями, от расположения материалов, инструментов и приспособлений относительно постоянного оборудования рабочего места и относительно самого рабочего, от подготовленности оборудования, инструмента и материалов к работе, от качества ухода за рабочим местом и оборудованием — от всего этого зависит производительность труда и качество продукции.

Именно техпроцесс и является той точкой, в которой сходятся требования всех этих подразделений и находится баланс между ними.

Далее описывается техпроцесс на нашем предприятии.

Не обработанное сырье (семена подсолнечника) поступают на ленточный конвейер, который занимается транспортировкой продукции от места выгрузки до ковшового конвейера.

Затем ковшовый конвейер перемещает сырье к участку сортировки.

На участке сортировки сырье попадает на весы – дозатор, который осуществляет контроль количества сырья.

Далее сырье попадает в сепаратор, который производит разделение сырья на фракции с разными характеристиками (отделение примеси).

После этого сырье поступает в камнеотборник, где подвергается очистке от камней, минеральных примесей и другой тяжелой фракции.

Очищенное сырье подается на ленточный конвейер, который в свою очередь транспортирует его до следующего конвейера.

Очередной конвейер транспортирует сырье до оптического сортировщика – сорттекс.

Сорттекс определяет дефектные компоненты, примеси, и удаляет их из потока продукции на основании цвета, формы или других свойств.

Ниже на рисунке 7 изображен процесс, в виде графической схемы:

					09.03.01.2018.313.00 ПЗ	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

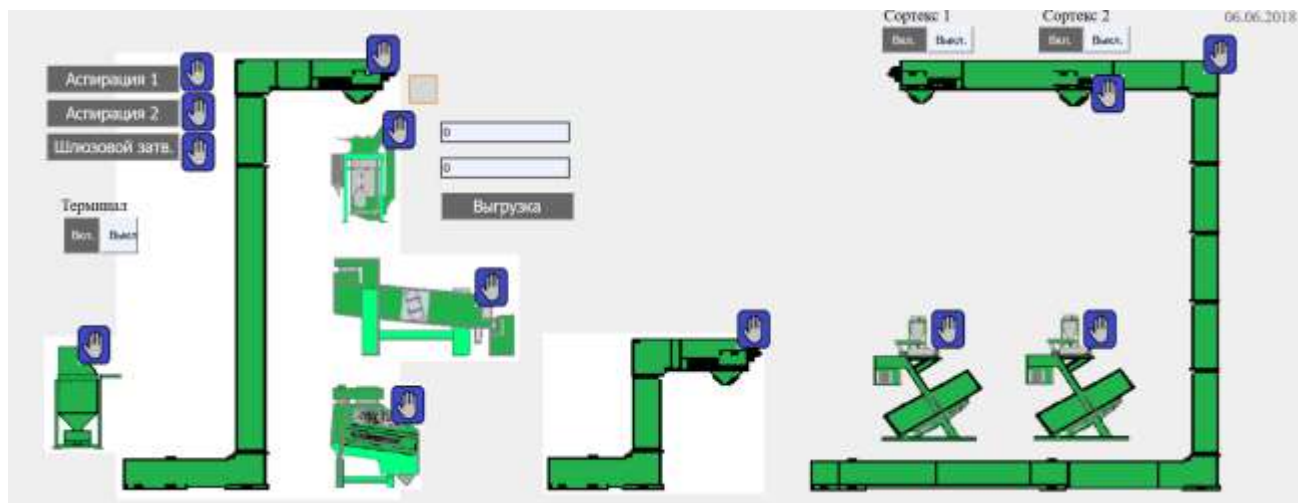


Рисунок 7 – графическое изображение технологического процесса

Сортекс не только отделяет очищенное сырье от примесей и брака (сырье с не соответствующей формой зерна), но и передает данный брак следующему тарельчатому конвейеру, который возвращает его обратно в сортекс, но уже в другой бункер.

Там уже соответственно сырье, либо проходит дальше, как годное для следующих процессов очистки, либо удаляется в бункер отработки.

Далее сырье попадает на следующий конвейер, который транспортирует его до устройства мойки сырья.

Затем очищенное сырье поступает на вибропитатель, который по своей специфике взаимодействует с обрабатываемой средой, обеспечивая более устойчивое истечение зерна через устье бункера, обезвоживает и препятствует зависанию и сводообразованию.

Просушенные зерна поступают на распределительный ленточный конвейер, откуда распределяется между двумя центрифугами.

Центрифуги применяются для отжима и очистки зерен от минеральных примесей.

Далее сырье поступает на следующий вибропитатель, где оно обрабатывается и перемещается ковшовым конвейером в бункер обжарочной машины, откуда готовая продукция попадает в упаковочный цех.

Ниже на рисунке 8 изображен очередной процесс, в виде графической схемы:

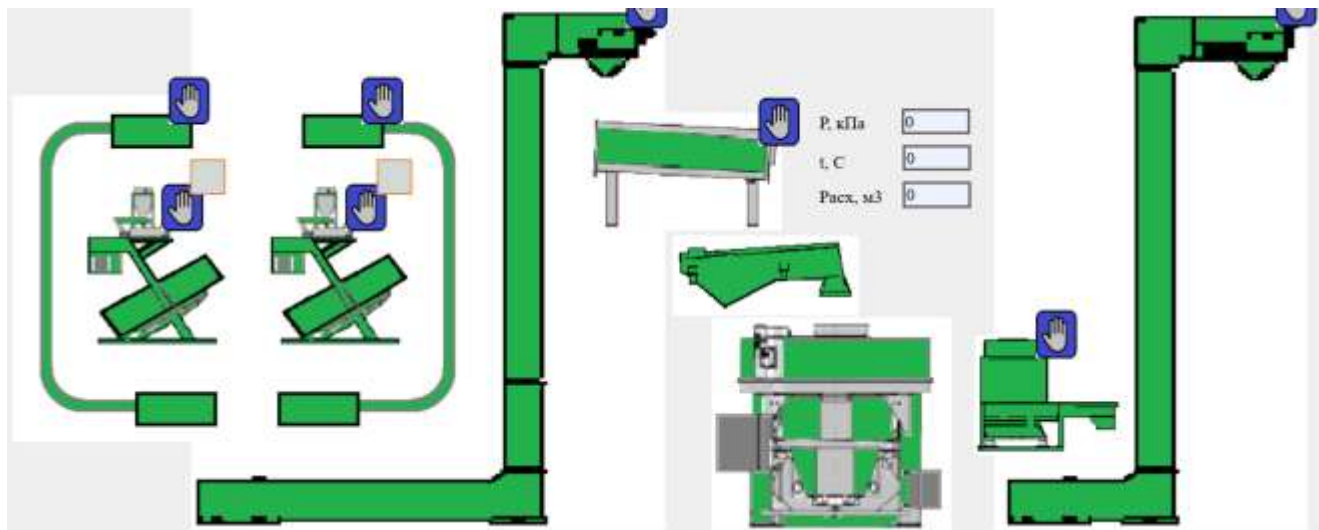


Рисунок 8 - графическое изображение технологического процесса

### 3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

#### 3.1 Взаимодействие клиентской части системы с серверной

После того как сотрудник организации перейдет по указанной ссылке на сайт (программное обеспечение визуализации и сбора данных для системы управления линией по производству жаренных семян подсолнечника), по умолчанию открывается самая первая страница (экран 1), показанная на рисунке 9:



Рисунок 9 – Первый экран программы

На данной странице отображаются узлы участвующие в производственном процессе на предприятии. Каждый узел имеет четыре статуса:

- Запущен, изображается зеленым цветом, узел находится в рабочем состоянии, проблемы отсутствуют;
- остановлен, изображается желтым цветом, узел остановлен или выключен (на данном рисунке как раз остановлен ковшовый конвейер);
- ошибка, изображается красным цветом, узел выдал сигнал ошибки, и не может продолжить свою работу;
- не доступен, изображается серым цветом, связь с узлом отсутствует.

Возле изображения каждого узла расположена метка времени и код состояния (в скобках). Метка времени показывает дату и время, когда узел был установлен в данное состояние. Код статуса визуализирует статус каждого узла в цифровом виде, это нужно для детализации статуса – ошибка, который может принимать множество значений.

На второй странице отображаются остальные узлы участвующие в производственном процессе на предприятии. Оставшиеся узлы имеют тоже количество статусов, с тем же описанием. Страница показана на рисунке 10:

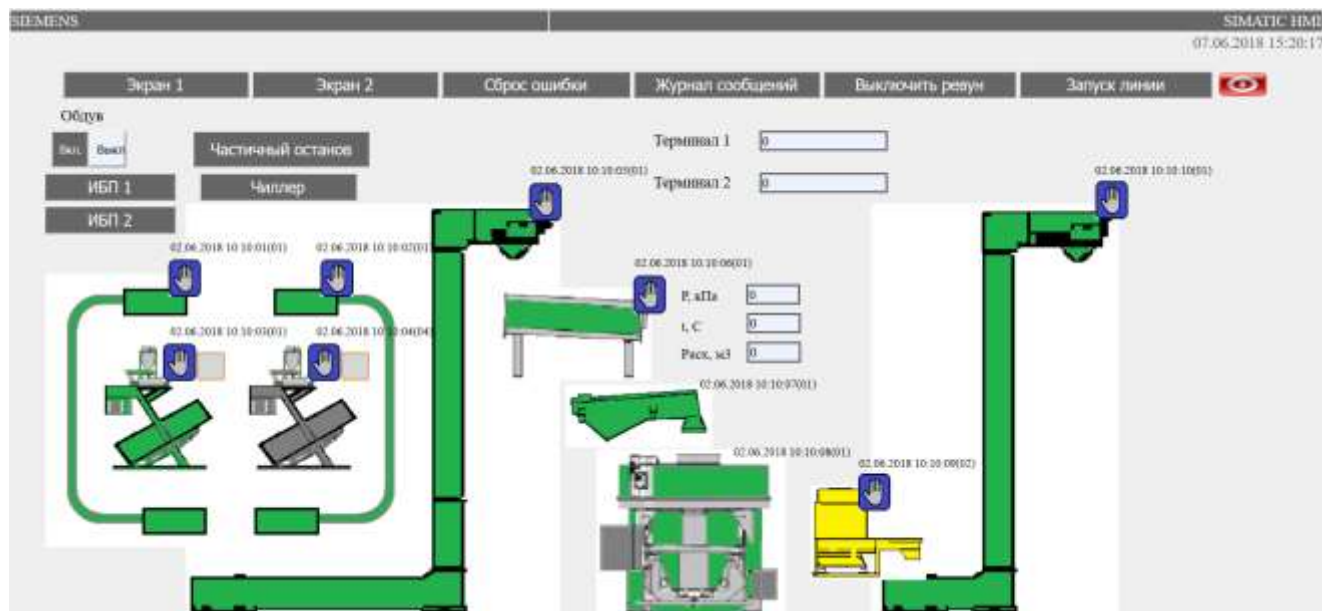


Рисунок 10 – Второй экран программы

Все сообщения, поступающие от узлов, фиксируются в журнале сообщений.

Ниже на рисунке 11 показан журнал сообщений. В него попадают все записи о статусах узлов, которые были изменены по любой причине. Журнал является архивом, который хранит информацию об изменении статусов узлов на протяжении двух месяцев. Очистка происходит автоматически раз в сутки.

Экран 1		Экран 2	Сброс ошибки	Журнал сообщений	Выключить реалун	Запуск линии
Дата и время	Узел	Статус узла	Идентификатор статуса			
06.06.2018 19:00:01	Конвейерный конвейер 1	Остановлен	02			
05.06.2018 13:13:13	Ленточный конвейер 1	Остановлен	02			
03.06.2018 9:09:09	Ленточный конвейер 1	Запущен	01			
02.06.2018 10:10:10	Конвейерный конвейер 5	Запущен	01			
02.06.2018 10:10:09	Выброшенный 2	Остановлен	02			
02.06.2018 10:10:06	Центрифуга 1	Запущен	01			
02.06.2018 10:10:07	Выброшенный 1	Запущен	01			
02.06.2018 10:10:06	Мельница 1	Запущен	01			
02.06.2018 10:10:05	Конвейерный конвейер 4	Запущен	01			
02.06.2018 10:10:04	Сортировщик 4	Налистует	04			
02.06.2018 10:10:03	Сортировщик 3	Запущен	01			
02.06.2018 10:10:02	Транспортерный конвейер 3	Запущен	01			
02.06.2018 10:10:01	Транспортерный конвейер 2	Запущен	01			
01.06.2018 19:00:01	Всего дозатор 1	Запущен	01			
01.06.2018 9:00:04	Сортировщик 1	Запущен	01			
01.06.2018 9:00:04	Сортировщик 2	Запущен	01			
01.06.2018 9:00:03	Конвейер 1	Запущен	01			
01.06.2018 9:00:03	Конвейерный конвейер 2	Запущен	01			
01.06.2018 9:00:02	Дозатор 1	Запущен	01			

Рисунок 11 – Журнал сообщений

Исходный код первой, второй страницы и журнала сообщений изображены в приложении А

### 3.2 Состав и функционирование компонентов серверной части системы

Серверная часть программы в своей работе использует базу данных Microsoft SQL Server 2016 Express. В ней созданы три таблицы: Nodes, Status и MessageLogs, которая является сводной. Отношения таблиц Nodes, Status со сводной являются один ко многим. В таблице Nodes хранится информация об

узлах: идентификатор и описание узла. В таблице Status хранится информация о статусах: идентификатор, описание и код статуса. Таблица MessageLogs содержит сводную информацию, которую сотрудник видит в журнале сообщений.

В программе используется класс ConnDb, который осуществляет подключение к базе данных и считывание информации об узлах и их статусах. В приложение Б приведен исходный код данного класса.

### 3.3 Взаимодействие микроконтроллера и серверной части системы

Взаимодействие сервера и микроконтроллера происходит для постоянного отслеживания статуса всех узлов. Для этого в программе используется служба Windows. Сервер использует клас ConnControllerToDb для создания подключения к микроконтроллеру, считывания информации об узлах и записи их в базу данных. Класс MonitorJob используется для создания простого планировщика, который будет опрашивать микроконтроллер через каждые 5с (запускать экземпляр класса ConnControllerToDb). Класс SmartWebService имеет метод OnStart(), который запускает экземпляр класса MonitorJob на выполнение.

Исходный код приведен в приложении В.

					09.03.01.2018.313.00 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

## ЗАКЛЮЧЕНИЕ

В работе выполнен анализ системы мониторинга и управления датчиками и устройствами с помощью панелей оператора Simatic HMI, находящихся под управлением микроконтроллеров фирмы Siemens Simatic S7 – 300. Выявлены основные недостатки при использовании существующей системы управления технологическими процессами, которые состояли в их ограниченности и доступности только в пределах их зон нахождения, затрудненной масштабируемости, в высоких трудовых и финансовых затратах на монтаж, в связи с необходимостью строительства специальных колонн и прочего оснащения.

Для устранения недостатков в существующую систему было внедрено программное обеспечение визуализации и сбора данных для системы управления линией по производству жаренных семян подсолнечника. Проведены испытания как при помощи виртуального оборудования, так и на реальном оборудовании.

Данная программная система обеспечила непрерывное слежение за узлами существующей системы технологического процесса, практически с любой рабочей станции или мобильного устройства находящегося в сети периметра предприятия. Непрерывная запись статуса узлов и их хранение в журнале сообщений, в течение длительного времени. Непрерывный мониторинг также открыл возможность, для раннего обнаружения неполадок в работе узлов, еще до того, как они приведут к серьезным последствиям и проблемам.

Данная программная система не предназначена для полной замены, или вытеснения вышеуказанных панелей оператора из работы управления существующего технологического процесса, но она гармонично сочетается с вышеуказанными технологиями, и дополняет их новыми возможностями, и обеспечивает высокий уровень гибкости и масштабируемости.

Предполагаемый эффект от внедрения работы – уменьшение количества сотрудников, которые должны осуществлять мониторинг производственного процесса, а т.е. сокращение числа панелей оператора. Уменьшение затрат составило 30%.

					09.03.01.2018.313.00 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СТО ЮУрГУ 04 –2008: Курсовое и дипломное проектирование: Общие требования к оформлению –Издательство «ЮУрГУ», 2008. –56 с.
2. Вуд, А., Микропроцессоры в вопросах и ответах / А. Вуд - Пер. с англ. под ред. Д.А. Поспелова. - М.: Энергоатомиздат. 1985. - 184 с.
3. Алексенко, А.И., Микросхемотехника / А.И. Алексеенко, И.В. Шагурин - М.: Радио и связь, 1990. - 496 с
4. Голощاپов, А.А., Microsoft Visual Studio 2017 / А.А. Голощاپов - БХВ-Петербург - Москва, 2017. - 544 с.
5. Понамарев, В.О., Программирование на C++/C# в Visual Studio .NET 2017 / В.О. Понамарев - БХВ-Петербург - Москва, 2017. - 352 с.
6. Жолобов, А.А., Технология автоматизированного производства / А.А. Жолобов – Издательство «Дизайн ПРО» - Москва, 2000. – 252 с.
7. Ключев, А.С., Проектирование систем автоматизации технологических процессов: Справочное пособие / А.С. Ключев, Б.В. Глазов - М.: Энергоатомиздат. 2006. - 184 с.
8. Шаумян, Г.А., Комплексная автоматизация производственных процессов / Г.А. Шаумян - Издательство «Машиностроение» - Москва, 2005. – 231 с.
9. Бродин, Б.В., Микроконтроллеры; Справочник / Б.В. Бродин - М.: Эком, 2016. - 395 с.
10. Хашемиан, Х.М., Датчики технологических процессов. Характеристики и методы повышения надежности / Х.М. Хашемиан - М.: Бином, 2014. - 336 с.

					09.03.01.2018.313.00 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

Исходный код первой страницы программы на HTML (визуализация)

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterForm.Master"
AutoEventWireup="true"
CodeBehind="wfPage01.aspx.cs" Inherits="SmartWeb.WebForm1" %>
<asp:Content ID="Content3" ContentPlaceHolderID=
"ContentHolder" runat="server">
<asp:UpdatePanel ID="UpdatePanel1"
runat="server" UpdateMode="Conditional">
<ContentTemplate>
<br />
<asp:ImageButton ID="imgBtnEz1Manual" s
tyle="position:fixed; top: 187px; left: 413px; z-index:1"
runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<br />
<asp:Button ID="btnScalesOverflow" runat="server"
BackColor="#D1DAD6" BorderColor="#F4863C"
BorderStyle="Solid" Height="32px" style="position:fixed; color:
#D1DAD6; top: 210px; left: 461px;" Width="32px" />
<br />
<asp:Label ID="lblSortex1Txt" style="position:fixed;
top: 154px; left: 995px;" runat="server"
Text="Соптекс 1" CssClass="auto-style3"></asp:Label>
<br />
<asp:Button ID="btnUnload" style="position:fixed;
top: 362px; left: 630px; width: 150px; color:
#FFFFFF; font-size: large;" runat="server"
Text="Выгрузка" BackColor="#666666" Height="32px" />
<asp:Button ID="btnChange2" style="position:fixed;
top: 390px; left: 71px; z-index:1; height: 42px; width: 42px;
color: #FFFFFF;" runat="server"
Text="Вкл." Height="42px" BackColor="#666666" BorderStyle="Outset" />
<asp:Label ID="lblSortex2Txt" runat="server"
style="position:fixed; top: 154px; left: 1174px;"
Text="Соптекс 2" CssClass="auto-style3"></asp:Label>
<asp:Button ID="btnSortex1Onn" style="position:fixed;
```

					09.03.01.2018.313.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

```

<br />
<asp:Label ID="lblTerminal" style="position:fixed;
top: 367px; left: 69px; z-index:1; font-size: large;
height: 22px; width: 85px;" runat="server"
Text="Терминал" Font-Size="Larger"></asp:Label>
<asp:Image ID="imgEz1" style="position:fixed;
top: 207px; left: 131px; width: 320px;"
runat="server" Height="500px" ImageUrl=
"~/Resources/Screen01/imgEz1_NoConnection.png" />
<asp:Button ID="btnChange1" style="position:fixed;
top: 390px; left: 114px; z-index:1; height: 42px;
width: 42px; text-align: center;" runat="server"
Text="Выкл." BackColor="#EDF3FE" Height="42px" />
<asp:Label ID="lblBc1" style="position:fixed;top: 495px;
left: 106px; z-index:3;" runat="server" Text="Загрузка...
" Font-Names="Times New Roman" Font-Size="Smaller"></asp:Label>
<br />
<br />
<asp:Image ID="imgScales1" style="position:fixed;
top: 280px; left: 370px; z-index:1;"
runat="server" Height="120px" ImageUrl=
"~/Resources/Screen01/imgScales1_NoConnection.png" />
<br />
<asp:Image ID="imgSeparator1" style="position:fixed;
top: 423px; left: 342px; z-index:1;" runat="server"
Height="120px"
ImageUrl="~/Resources/Screen01/imgSeparator1_NoConnection.png" />
<asp:Image ID="imgDestoner1"
style="position:fixed; top: 577px; left: 370px;
z-index:1;" runat="server"
Height="120px"
ImageUrl="~/Resources/Screen01/imgDestoner1_NoConnection.png" />
<br />
<br />
<asp:Image ID="imgBc1" style="position:fixed;
top: 525px; left: 50px;" runat="server"
Height="130px" ImageUrl=

```

```

"~/Resources/Screen01/imgBc1_NoConnection.png"
AlternateText="Не обнаружен файл рисунка" />
<br />
<br />
<br />
<asp:Image ID="imgEz2" style="position:fixed;
top: 521px; left: 611px;"
runat="server" ImageUrl="~/Resources/Screen01/imgEz2_NoConnection.png"
Height="180px" />
<br />
<br />
<asp:Image ID="imgEc3" style="position:fixed;
top: 197px; left: 903px;"
runat="server"
Height="500px" ImageUrl=
"~/Resources/Screen01/imgEc3_NoConnection.png" />
<br />
<asp:Label ID="lblEz1" style="position:fixed;
top: 169px; left: 430px; z-index:3;
right: 949px;" runat="server" Text="Загрузка..."
Font-Names="Times New Roman" Font-Size="Smaller"></asp:Label>
<br />
<asp:ImageButton ID="imgBtnEc3Manual"
style="position:fixed; z-index:1; top: 187px;
left: 1355px;" runat="server"
ImageUrl="~/Resources/Other/Manual1.png" />
<asp:Label ID="lblScales1" style="position:fixed;
z-index:3; top: 252px; left: 445px;" runat="server"
Text="Загрузка..." Font-Names=
"Times New Roman" Font-Size="Smaller"></asp:Label>
<asp:Label ID="lblEc3" style="position:fixed;
z-index:3; top: 239px; left: 1381px;"
runat="server" Text="Загрузка..." Font-Names="Times New Roman" Font-
Size="Smaller"></asp:Label>
<br />
<asp:Image ID="imgSortex1" style="position:fixed;
top: 497px; left: 976px;"

```

```

runat="server" ImageUrl="~/Resources/Screen01/imgSortex1_NoConnection.png"
Height="150px" />
<asp:Image ID="imgSortex2" style="position:fixed;
top: 495px; left: 1147px;"
runat="server" ImageUrl="~/Resources/Screen01/imgSortex2_NoConnection.png"
<asp:TextBox ID="txtbxUnload1"
style="position:fixed; top: 324px; left: 630px;
background-color: #EDF3FE;" runat="server"
BorderColor="#666666" BorderWidth="3px">0</asp:TextBox>
<asp:TextBox ID="txtbxUnload2"
style="position:fixed; top: 288px; left: 630px;
background-color: #EDF3FE;" runat="server"
BorderColor="#666666"
BorderWidth="3px" ViewStateMode="Enabled">0</asp:TextBox>
<asp:Button ID="btnSortex2On" style="position:fixed;
top: 177px; left: 1168px; color: #FFFFFF;"
runat="server" Text="Вкл." BackColor="#666666"
BorderStyle="Outset" Height="30px" Width="52px" />
<asp:ImageButton ID="imgBtnBc1Manual"
style="position:fixed; top: 517px; left: 92px;
z-index:1" runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnDestoner1Manual"
style="position:fixed; z-index:1; top: 580px;
left: 441px;" runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnSortex2Manual"
style="position:fixed;z-index:1;
top: 495px; left:1220px;"runat="server"ImageUrl=
"~/Resources/Other/Manual1.png" />
<asp:Label ID="lblValve" style="position:fixed;
top: 272px; left: 1183px; z-index:3"
runat="server" Text="Загрузка..."
Font-Names="Times New Roman" Font-Size="Smaller"></asp:Label>
<br />
<asp:ImageButton ID="imgBtnAsp2Manual"
runat="server" ImageUrl="~/Resources/Other/Manual1.png"
style="position:fixed; top: 252px; left: 205px; z-index:1" />
<asp:ImageButton ID="imgBtnGatewayManual"

```

```

style="position:fixed; top: 297px; left: 205px;
z-index:1" runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnSeparator1Manual"
style="position:fixed; z-index:1; top: 448px; left: 535px;"
runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgbtnEz2Manual"
style="position:fixed; z-index:1; top: 495px;
left: 830px;" runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnValveManual"
style="position:fixed; z-index:1;
top: 231px; left:1229px;"runat="server"ImageUrl=
"~/Resources/Other/Manual1.png" />
<asp:Button ID="btnSortex1Off"
style="position:fixed; top: 177px; left: 1044px;"
runat="server" Text="Выкл."
BackColor="#EDF3FE"Height="30px" Width="52px" />
<asp:ImageButton ID="imgBtnAsp1Manual"
style="position:fixed; top: 207px; left: 204px;
z-index:1" runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnScales1Manual"
left: 431px;"runat="server" ImageUrl="~/Resources/Other/Manual1.png" />
<asp:ImageButton ID="imgBtnSortex1Manual"
style="position:fixed; z-index:1;
top: 495px; left:1049px;"runat="server"ImageUrl=
"~/Resources/Other/Manual1.png" />
<asp:Label ID="lblSortex1" style="position:fixed;
z-index:3; top: 472px; left: 1048px;" runat="server"
Text="Запызка..." Font-Names="Times New Roman"
Font-Size="Smaller"></asp:Label>
<br />
<asp:Image ID="imgValve" style="position:fixed;
top: 230px; left: 1179px;"
runat="server" ImageUrl=
"~/Resources/Screen01/imgValve_NoConnection.png" />
<asp:Label ID="lblSeparator1" style="position:fixed;
z-index:3; top: 426px; left: 551px;" runat="server"
Text="Запызка..." Font-Names="Times New Roman"

```

```

Font-Size="Smaller"></asp:Label>
<asp:Label ID="lblDestoner1" style="position:fixed;
z-index:3; top: 561px; left: 456px;" runat="server"
Text="Зарплата..." Font-Names=
"Times New Roman" Font-Size="Smaller"></asp:Label>
<asp:Label ID="lblEz2" style="position:fixed; z-index:3;
top: 475px; left: 831px;" runat="server" Text=
"Зарплата..." Font-Names="Times New Roman"
Font-Size="Smaller"></asp:Label>
EventName="Tick" />
</Triggers>
</asp:UpdatePanel>
</asp:Content>
<asp:Content ID="Content4" runat="server" contentplaceholderid="head">
<style type="text/css">
    .auto-style1
    {
        color: #FFFFFF;
    }
    .auto-style3
    {
        font-size: large;
    }
</style>
</asp:Content>

```

Исходный код первой страницы программы на C#

```

using System;
using SmartWeb.AdminTasks;
namespace SmartWeb
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void timerPage01_Tick(object sender, EventArgs e)
        {

```

```

//Создание экземпляра класса ConnDb, отвечает за соединение с базой данных
ConnDb ConnectionDB = new ConnDb();

//Создание двумерного массива и инициализация его данными полученными
из базы данных
string[][] strSampleNodes = ConnectionDB.SqlConnection();

//Перебор в цикле всех узлов и присвоение изображения узла, в соответствии
с его статусом
for(int i = 0; i < strSampleNodes.Length; i++)
{
    //Строковая переменная, хранящая полный путь до изображения каждого узла
    string imgPath = "~/Resources/Screen01/" + strSampleNodes[i][1] + "_" +
strSampleNodes[i][4] + ".png";

    //Строковая переменная, хранящая метку времени и код статуса каждого узла
    string lblStatus = strSampleNodes[i][0] + "(" + strSampleNodes[i][2] + ")";

    //Присвоение каждому узлу изображения статуса и метки времени, в
соответствии с его статусом
    switch (strSampleNodes[i][1])
    {
        case "imgBc1":
            imgBc1.ImageUrl = imgPath;
            lblBc1.Text = lblStatus;
            break;
        case "imgEz1":
            imgEz1.ImageUrl = imgPath;
            lblEz1.Text = lblStatus;
            break;
        case "imgScales1":
            imgScales1.ImageUrl = imgPath;

            lblScales1.Text = lblStatus;
            break;
        case "imgSeparator1":
            imgSeparator1.ImageUrl = imgPath;
            lblSeparator1.Text = lblStatus;
            break;
        case "imgDestoner1":
            imgDestoner1.ImageUrl = imgPath;

```



```

lblDestoner1.Text = lblStatus;
break;
case "imgEz2":
    imgEz2.ImageUrl = imgPath;
    lblEz2.Text = lblStatus;
    break;
case "imgSortex1":
    imgSortex1.ImageUrl = imgPath;
    lblSortex1.Text = lblStatus;
    break;
case "imgSortex2":
    imgSortex2.ImageUrl = imgPath;
    lblSortex2.Text = lblStatus;
    break;
case "imgValve":
    imgValve.ImageUrl = imgPath;
    lblValve.Text = lblStatus;
    break;
case "imgEc3":
    imgEc3.ImageUrl = imgPath;
    lblEc3.Text = lblStatus;
    break;
}
}
} }

```

Исходный код второй страницы программы на HTML (визуализация)

```

<% @ Page Title="" Language="C#" MasterPageFile=
 "~/MasterForm.Master" AutoEventWireup="true" CodeBehind=
 "wfPage02.aspx.cs" Inherits="SmartWeb.WebForm2" %>
<asp:Content ID="Content3" ContentPlaceHolderID=
 "ContentHolder" runat="server">
<
<asp:Timer ID="tmrPage02" runat="server"
 Interval="2000" OnTick="timerPage02_Tick">
</asp:Timer>
</ContentTemplate>

```

```

</asp:UpdatePanel>
    &nbsp;
    <br />
</asp:Content>

```

Исходный код второй страницы программы на C#

```

using System;
using SmartWeb.AdminTasks;
namespace SmartWeb
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void timerPage02_Tick(object sender, EventArgs e)
        {
            //Создание экземпляра класса ConnDb, отвечает за соединение с базой данных
            ConnDb ConnectionDB = new ConnDb();

```

Продолжение приложения А

//Создание двумерного массива и инициализация его данными полученными из  
базы данных

```

        string[][] strSampleNodes = ConnectionDB.SqlConnection();
        //Перебор в цикле всех узлов и присвоение изображения сузла, в
        соответствии с его статусом
        for (int i = 0; i < strSampleNodes.Length; i++)
        {
            //Строковая переменная, хранящая полный путь до изображения каждого узла
            string imgPath = "~/Resources/Screen02/" + strSampleNodes[i][1] + "_"
            + strSampleNodes[i][4] + ".png";
            //Строковая переменная, хранящая метку времени и код статуса каждого узла
            string lblStatus=strSampleNodes[i][0]+ "(" + strSampleNodes[i][2] + ")";
            //Присвоение каждому узлу изображения статуса и метки времени, в
            соответствии с его статусом
            switch (strSampleNodes[i][1])
            {
                case "imgCableveyLeft":
                    imgCableveyLeft.ImageUrl = imgPath;

                    lblCableveyLeft.Text = lblStatus;

```

					09.03.01.2018.313.00 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

```

        break;
    case "imgCableveyRight":
        imgCableveyRight.ImageUrl = imgPath;
        lblCableveyRight.Text = lblStatus;
        break;
    case "imgSortex3":
        imgSortex3.ImageUrl = imgPath;
        lblSortex3.Text = lblStatus;
        break;
    case "imgSortex4":
        imgSortex4.ImageUrl = imgPath;
        lblSortex4.Text = lblStatus;
        break;
    case "imgEz4":
        imgEz4.ImageUrl = imgPath;
        lblEz4.Text = lblStatus;
        break;
    case "imgWasher1":
        imgWasher1.ImageUrl = imgPath;
        lblWasher1.Text = lblStatus;
        break;
    case "imgVf1":
        imgVf1.ImageUrl = imgPath;
        lblVf1.Text = lblStatus;
        break;
    case "imgCentrifuge1":
        imgCentrifuge1.ImageUrl = imgPath;
        lblCentrifuge1.Text = lblStatus;
        break;
    case "imgVf2":
        imgVf2.ImageUrl = imgPath;
        lblVf2.Text = lblStatus;
        break;
    case "imgEz5":
        imgEz5.ImageUrl = imgPath;
        lblEz5.Text = lblStatus;
        break; } } } } }

```

```

Исходный код класса ConnDb
using System.Data.SqlClient;
using System.Collections;
namespace SmartWeb.AdminTasks
{
/Класс для создания подключения к бд и считывания информации об узлах
class ConnDb
{
    private SqlConnection connSql;
    public ConnDb()
    {
        //Строка подключения к базе данных
        connSql = new
SqlConnection("Data Source=
WebServer.local.ru\\WEBSQL;Initial
Catalog=webdb;User ID=User;Password=Password");
    }
    public string[][] SqlConnection()
    {
        SqlCommand cmdSql;
        string lineSqlML;
        string lineSqlNodes;
        //Массив для хранения информации об узлах
        ArrayList strAllNodes = new ArrayList()
        connSql.Open();
        //Выборка всех узлов из базы данных
        lineSqlNodes = "SELECT Id FROM Nodes";
        cmdSql = new SqlCommand(lineSqlNodes, connSql);
        SqlDataReader readerSqlNodes = cmdSql.ExecuteReader()
        while (readerSqlNodes.Read())
        {
            strAllNodes.Add (readerSqlNodes["Id"].ToString());
        }
        string[][] strSampleNodes = new string [strAllNodes.Count][];
        readerSqlNodes.Close();
        for(int i =0; i < strAllNodes.Count; i++)
        {

```

//Запрос для выборки информации об узлах (дата, узел, статус, код статуса)

```

lineSqlML = "SELECT TOP 1 MessageLogs.DateTime,
Nodes.BindToObject AS NodesBindToObject,
Status.IdStatus, Status.Description, Status.BindToObject
AS StatusBindToObject" +
            " FROM MessageLogs INNER
JOIN Nodes ON MessageLogs.FkNodes = Nodes.Id INNER JOIN
Status ON MessageLogs.FkStatus" +
            " = Status.Id WHERE (MessageLogs.FkNodes = " +
strAllNodes[i] + ") ORDER BY MessageLogs.DateTime DESC";
cmdSql = new SqlCommand(lineSqlML, connSql);
SqlDataReader readerSql = cmdSql.ExecuteReader();
while (readerSql.Read())
{
    //Сохр. полученной инф. об узлах в массиве, для дальнейшей передачи
    strSampleNodes[i] = new string[5];
    strSampleNodes[i][0] = readerSql["DateTime"].ToString();
    strSampleNodes[i][1] = readerSql["NodesBindToObject"].ToString();
    strSampleNodes[i][2] = readerSql["IdStatus"].ToString();
    strSampleNodes[i][3] = readerSql["Description"].ToString();
    strSampleNodes[i][4] = readerSql["StatusBindToObject"].ToString();
}
readerSql.Close();}
connSql.Close();
return strSampleNodes;
}
}
}

```

```

Исходный код службы Windows: класс ConnControllerToDb
using System;
using System.Data.SqlClient;
using System.Collections;
using S7.Net;
namespace WebService.ScheduledTask
{
    class ConnControllerToDb
    {
        private SqlConnection connSql; //Строка подключения к БД на выборку
        private string lineSql; //SQL запрос к БД
        private SqlCommand cmdSql;
        private string sqlIdStatus = null; //Статус узла из БД
        private ArrayList arrId = new ArrayList(); //Массив со всеми Id узлов
        private ArrayList arrIdNodes = new ArrayList(); //Массив со всеми IdNodes узлов
        //Создание экземпляра класса, с настройками подкл. к микроконтр.
        private Plc plc = new Plc(CpuType.S7300, "Microcontroller.local.ruu", 0, 2);
        string microIdStatus = null; //Статус узла полученного от микроконтр.
        public ConnControllerToDb()
        {
            connSql = new SqlConnection("Data Source=WebServer\\WEBSQL;Initial
Catalog=webdb;User ID=User;Password=Password");
            lineSql = string.Format("SELECT Id, IdNodes FROM Nodes");
            cmdSql = new SqlCommand(lineSql, connSql);
        }
        public void SqlConnection()
        {
            string contrResponse; //Содержит статус коррект. подкл. к микроконтр.
            connSql.Open();
            //Выполняем запроса к БД и считываем записи об узлах
            SqlDataReader readerSqlNodes = cmdSql.ExecuteReader();
            while (readerSqlNodes.Read())
            {
                arrId.Add(readerSqlNodes["Id"].ToString());
                arrIdNodes.Add(readerSqlNodes["IdNodes"].ToString());
            }
            readerSqlNodes.Close();
        }
    }
}

```

```

contrResponse = plc.Open().ToString();
if (contrResponse == "NoError")
{
    GetStatusNodes();
}
else
{
    microIdStatus = "5";
    GetStatusNodes();
}
connSql.Close();
}
public void GetStatusNodes()
{
    for (int i = 0; i < arrId.Count; i++)
    {
        if (microIdStatus != "5")
            microIdStatus = plc.Read(arrIdNodes[i].ToString()).ToString();

        lineSql = "SELECT TOP 1 Status.IdStatus FROM MessageLogs " +
            "INNER JOIN Nodes ON MessageLogs.FkNodes = Nodes.Id " +
            "INNER JOIN Status ON MessageLogs.FkStatus = Status.Id " +
            "WHERE (MessageLogs.FkNodes = " + arrId[i].ToString() + ") " +
            "ORDER BY MessageLogs.DateTime DESC";
        cmdSql = new SqlCommand(lineSql, connSql);
        SqlDataReader readerSql = cmdSql.ExecuteReader();
        while (readerSql.Read())
        {
            //Сохранение полученной информации об узле
            sqlIdStatus = readerSql["IdStatus"].ToString();
        }
        readerSql.Close();

        if (sqlIdStatus != microIdStatus)
        {
            lineSql = "INSERT INTO MessageLogs (DateTime, FkNodes, FkStatus) " +
                "SELECT '" + DateTime.Now.ToString() + "', " +

```

```

        " '" + arrId[i].ToString() +
        "', Id FROM Status WHERE IdStatus = '" + microIdStatus + "'";
        cmdSql = new SqlCommand(lineSql, connSql);
        cmdSql.ExecuteNonQuery();
    }
}
plc.Close();
}
}
}

```

Исходный код службы Windows: класс MonitorJob

```

using System;
using System.Threading;
using System.Threading.Tasks;
namespace WebService.SheduledTask
{
    public class MonitorJob : IDisposable
    {
        private CancellationTokenseource m_cancel;
        private Task m_task;
        private TimeSpan m_interval;
        private bool m_running;

        public MonitorJob(TimeSpan interval)
        {
            m_interval = interval;
            m_running = true;
            m_cancel = new CancellationTokenseource();
            m_task = Task.Run(() => TaskLoop(), m_cancel.Token);
        }
        private void TaskLoop()
        {
            while (m_running)
            {
                ConnControllerToDb cctd = new ConnControllerToDb();
            }
        }
    }
}

```



```

cctd.SqlConnection();
        Thread.Sleep(m_interval);
    }
}
public void Dispose()
{
    m_running = false;

    if (m_cancel != null)
    {
        try
        {
            m_cancel.Cancel();
            m_cancel.Dispose();
        }
        catch
        {
        }
        finally
        {
        }

        m_cancel = null;
    }
}
}
}

```

Исходный код службы Windows: класс SmartWebService

```

using System.ServiceProcess;
using WebService.SheduledTask;
using System;
using System.Threading;
using System.Threading.Tasks;

```

```

namespace WebService
{
    public partial class SmartWebService : ServiceBase
    {

        public SmartWebService()
        {
            InitializeComponent();
        }

        private static MonitorJob m_job;
        protected override void OnStart(string[] args)
        {
            m_job = new MonitorJob(TimeSpan.FromSeconds(5));
        }

        protected override void OnStop()
        {
            m_job.Dispose();
        }
    }
}

```