

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ФИЗИКИ

КАФЕДРА РАДИОЭЛЕКТРОНИКИ

Направление: 03.04.03 Радиофизика
Профиль: «Радиофизические методы по областям применения
(радиофизические измерения)»

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
«ПРИМЕНЕНИЕ КОНЦЕПЦИИ ПРОМЫШЛЕННОГО ИНТЕРНЕТА
ВЕЩЕЙ (IIOT) ДЛЯ ПОСТРОЕНИЯ СИСТЕМ ТЕЛЕМЕХАНИКИ И
ТЕЛЕМЕТРИИ РАСПРЕДЕЛЕННЫХ ПРОМЫШЛЕННЫХ
ОБЪЕКТОВ НА ПРИМЕРЕ ТЕХНОЛОГИИ ДОБЫЧИ НЕФТИ».**

Студент 2 курса группы 06-829
«15» 06 2020 г.



(К.И. Булатов)

Научный руководитель
к.ф.-м.н., доцент
« » 2020 г.

(А.Г. Гаврилов)

Заведующий кафедрой
д.ф.-м.н., профессор
« » 2020 г.

(М.Н. Овчинников)

Казань-2020

Содержание

Введение.....	4
Глава 1. Основные сведения о нефтедобыче.....	5
1.1. Станок-качалка и принцип работы	5
1.2. Групповая замерная установка	8
Глава 2. Концепция Интернет Вещей (IoT) и промышленного Интернета Вещей (IIoT).....	13
2.1. Что такое интернет вещей (IoT)	13
2.2. Индустриальный интернет вещей (IIoT)	13
2.2.1. Как работает промышленный интернет вещей.....	14
2.3. Облачные технологии и Интернет Вещей.....	15
Глава 3. Состав программно-аппаратного комплекса.....	18
3.1. Контроллер «ADAM 3600»	18
3.2. Модуль «ADAM 3624»	20
Глава 4. Языки программирования, сервисы, протоколы и среды разработки	21
4.1. Node-RED – мощный инструмент для создания приложений Internet of Things.....	21
4.2. MQTT (Message Queue Telemetry Transport).....	27
4.3. Multiprog express 5.35 - программируемая система.....	28
4.4. Платформа Edgelinek.....	29
Глава 5. Описание прикладного программного обеспечения	32
5.1. Описание структурной схемы.....	32
5.2. Разработка ПО контроллера скважины СКН на языке ST для ADAM- 3600.....	33
5.3. Конфигурация контроллера «ADAM 3600» утилитой EdgeLink.....	39
5.4. Разработка клиентской части с помощью Node-RED.....	44
5.5. Комплексное тестирование программного обеспечения.....	46
Заключение	49
Список литературы	50

Приложение А	52
Приложение В.....	56

Введение

К основным особенностям добычи нефти и газа, определяющим рентабельность этой важной отрасли страны, относятся большая топологическая распределенность технологических объектов, сложные условия эксплуатации оборудования, большие капитальные и эксплуатационные затраты на добычу сырья, существенные потери эффективности разработок в случае отклонений режимов добычи от плановых и в аварийных ситуациях.

Одним из факторов повышения рентабельности нефтедобычи является разработка и внедрение новых методов и средств автоматизации контроля и управления технологическими режимами каждой производственной установки. Этому способствуют появление на рынке средств автоматизации различных автоматизированных информационно-управляющих комплексов.

Целью данной работы является внедрение технологий промышленного интернета вещей (IIoT) для построения систем телемеханики и телеметрии распределенных промышленных объектов на примере технологии добычи нефти. Реализация этой цели осуществлялась посредством решения следующих основных задач:

- Создать структурную схему системы
- Изучить практические аспекты в области технологии добычи нефти
- Освоить язык программирования Structured Text (ST)
- Освоить работу с облачным сервисом для обмена данными по протоколу MQTT
- Сконфигурировать контроллер «ADAM 3600» утилитой EdgeLink
- Разработать клиентское приложение на Node-RED
- Разработка ПО контроллера скважины СКН для ADAM-3600
- Провести комплексную отладку системы

Глава 1. Основные сведения о нефтедобыче

1.1. Станок-качалка и принцип работы

Станок-качалка - один из элементов эксплуатации нефтедобывающих скважин штанговым насосом. Операторы по добыче нефти и газа определяют это оборудование как: "Индивидуальный балансирный механический привод штангового насоса" [1].

Станок-качалка является важным видом нефтегазового оборудования и используется для механического привода к нефтяным скважинным штанговым (плунжерным) насосам. Конструкция станка-качалки представляет собой балансирный привод штанговых насосов, состоящий из редуктора и сдвоенного четырехзвенного шарнирного механизма [1].

Станок-качалка предназначен для индивидуального механического привода к нефтяным скважинным штанговым насосам. Станок-качалка конструктивно представляет собой индивидуальный балансирный привод штанговых насосов, состоящий из редуктора и сдвоенного четырехзвенного шарнирного механизма, с роторным и роторно-балансирным уравновешиванием, преобразующим вращательное движение кривошипов в вертикальное движение канатной подвески устьевого штока с прикрепленной к нему колонной насосных штанг [2].

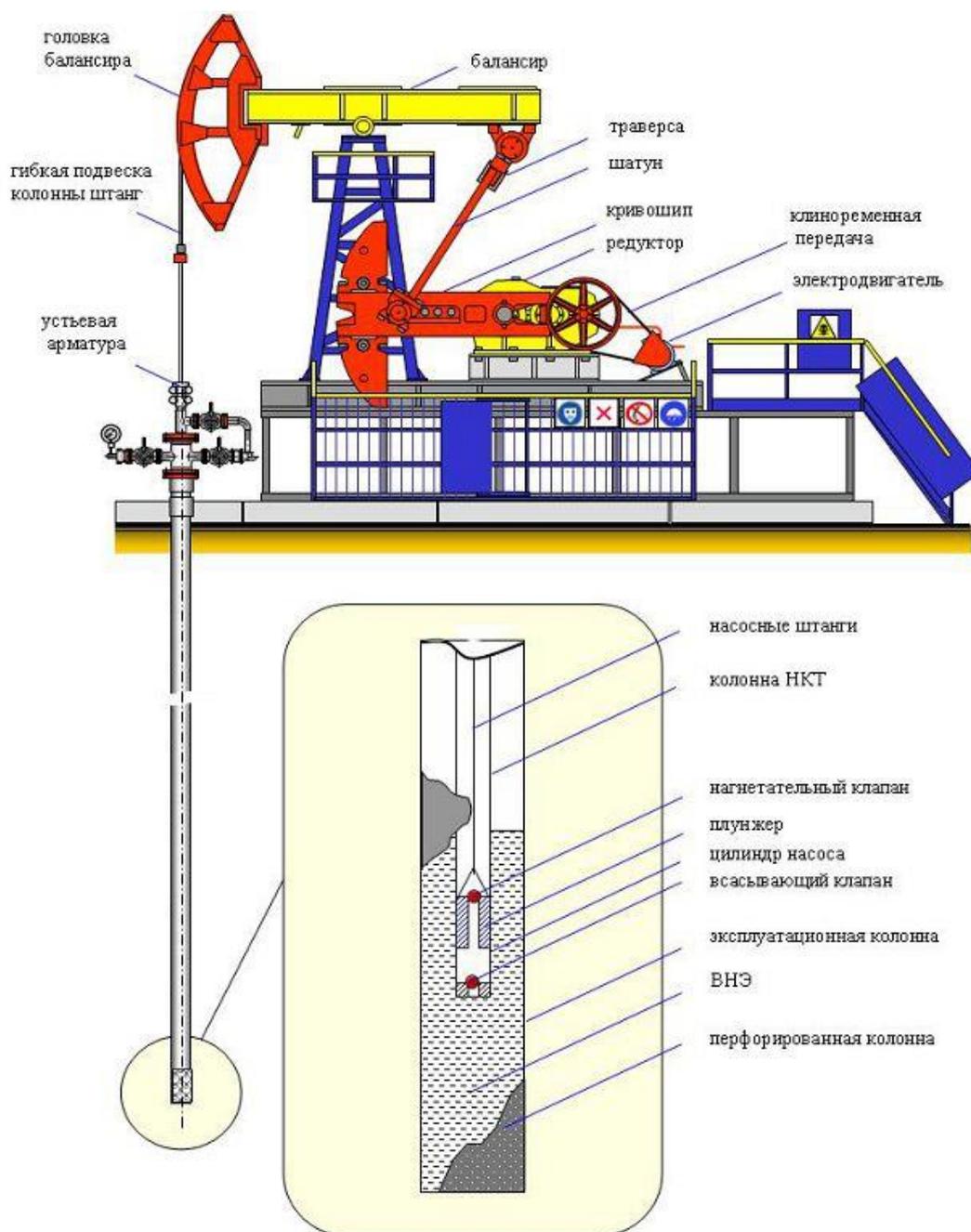


Рис. 1. Конструктивные элементы СК

Станки-качалки предназначены для передачи поступательного движения глубинному штанговому насосу, расположенному на дне скважины. Для уменьшения затрат на энергию оборудование должно обладать уникальной кинематической схемой. Дополнительным условием является применение современных комплектующих и компонентов [3].

Для анализа функциональности и особенности работы необходимо ознакомиться с конструкцией, которой обладает станок-качалка. Он состоит

из силовой установки, вращательное движение от которой поступает на ведущий вал редуктора. На нем расположен кривошип с системой противовесов. Для связи кривошипа с балансиром предусмотрены шатуны и траверсы. В свою очередь, балансир установлен на опорной стойке. Для уменьшения затраты энергии на торцевой части балансира расположена откидная головка [3].

Правильно установленный станок имеет следующие эксплуатационные качества:

- высокий показатель КПД. Обусловлен системой противовесов, которые позволят оптимизировать затраты энергии;
- надежность. Станок качалка способен работать продолжительное время. Главное — обеспечивать должный уровень смазки подвижных механизмов;
- сложность установки. Для нормальной эксплуатации станки-качалки необходимо устанавливать на обустроенные опорные платформы. Чаще всего их изготавливают методом заливки бетонной смесью.

Наряду с этой конструкцией в нефтедобывающей отрасли применяется безбалансирующее оборудование. Эти модели отличаются относительно небольшими размерами и массой, но при этом обладают более низким показателем КПД. Чаще всего устанавливаются в труднодоступных районах или местах, где обустройство полноценного фундамента затруднено [3].

Перечень основных технических характеристик:

- максимально допустимый показатель нагрузки на устьевом штоке. Он может варьироваться от 30 до 100 кН;
- длина хода штока. Обычно она составляет от 1,2 до 3 м;
- крутящий момент вала выходного редуктора. Он влияет на интенсивность движения штока и может быть равен от 6,3 до 56 кНм;
- число ходов балансира варьируется от 1,2 до 15 в минуту.

1.2. Групповая замерная установка

ГЗПУ (групповая замерная переключающаяся установка) – для производства замера дебита скважин и куста в целом и контроль за их работой. Состоит: корпус, трубная обвязка, гребенка, ПСМ, мерный газосепаратор, счетчик расхода TOP-1 (турбинный объемный расходомер), регулятор расхода, запорная арматура, вытяжка, обогреватели [4].

ПСМ (переключатель скважин многоходовой) – для автоматического и ручного перевода потока добываемой из отдельной скважины жидкости в газосепаратор. Состоит из: корпуса с входными патрубками, расположенными ассиметрично в горизонтальной плоскости корпуса, переключающей каретки, расположенной в корпусе с возможностью вращения относительно оси корпуса и соединенной через вал и зубчатую гребенку с поршневым гидроприводом, углового выходного патрубка с системой уплотнений, установленного в каретке так, что при вращении каретки он последовательно сообщается со всеми входными патрубками и соответственно, последовательно направляет на отводящий трубопровод поток жидкости от каждой подключенной к ПСМ скважине. Поток жидкости по трубопроводу направляется к двухкорпусному газосепаратору с поплавковым управляющим устройством. Разгазированная жидкость далее поступает на счетчик расхода TOP [4].

TOP-1 – для измерения объема жидкости выходящей из газосепаратора. Состоит из: углового подводящего патрубка и из цилиндрической проточной части с размещенной в ней крыльчаткой (турбиной), вал которой связан с понижающим шестеренчатым редуктором, вращающим магнитную муфту, которая в свою очередь за счет магнитных сил передает крутящий момент на внешний механический счетчик с указательной стрелкой и диском с двумя постоянными магнитами, которые при вращении диска замыкают контакты расположенного рядом с механическим счетчиком электромагнитного датчика и сигналы электромагнитного датчика регистрируются на блоке

местной автоматики, а замеряемая жидкость проходящая по проточной части через отверстие выполненное ниже турбинки поступает в отводящий патрубок расположенный соосно с входной частью подводящего патрубка. ТОР-1 устанавливается вертикально и работает следующим образом: жидкость через подводящий патрубок поступает в проточную часть и вращает находящуюся там турбинку, а затем через имеющиеся в проточной части окна поступает в отводящий патрубок. Замеренная на ТОРе жидкость проходит через регулятор расхода и далее соединяясь с газом в основной коллектор [4].

В системе сбора нефти и газа, АГЗУ устанавливается непосредственно на месторождении. К АГЗУ по выкидным линиям поступает продукция с нескольких добывающих скважин. К одной установке, в зависимости от её конструкции, может подключаться до 14 скважин [4].

При этом поочередно осуществляется замер дебита жидкости по каждой скважине. На выходе из АГЗУ продукция всех скважин поступает в один трубопровод — «сборный коллектор» и транспортируется на дожимную насосную станцию (ДНС) или непосредственно на объекты подготовки нефти и газа [4].

Рассмотрим маркировку АГЗУ на примере установки «Спутник АМ 40-8-400»:

- 40 — максимальное рабочее давление, в кгс/см².
- 8 — количество подключаемых скважин.
- 400 -максимальный измеряемый дебит скважины по жидкости в м³/сут.

АГЗУ состоит из двух отдельных блоков: технологического блока, аппаратного блока. Рассмотрим технологический блок (рис.2).

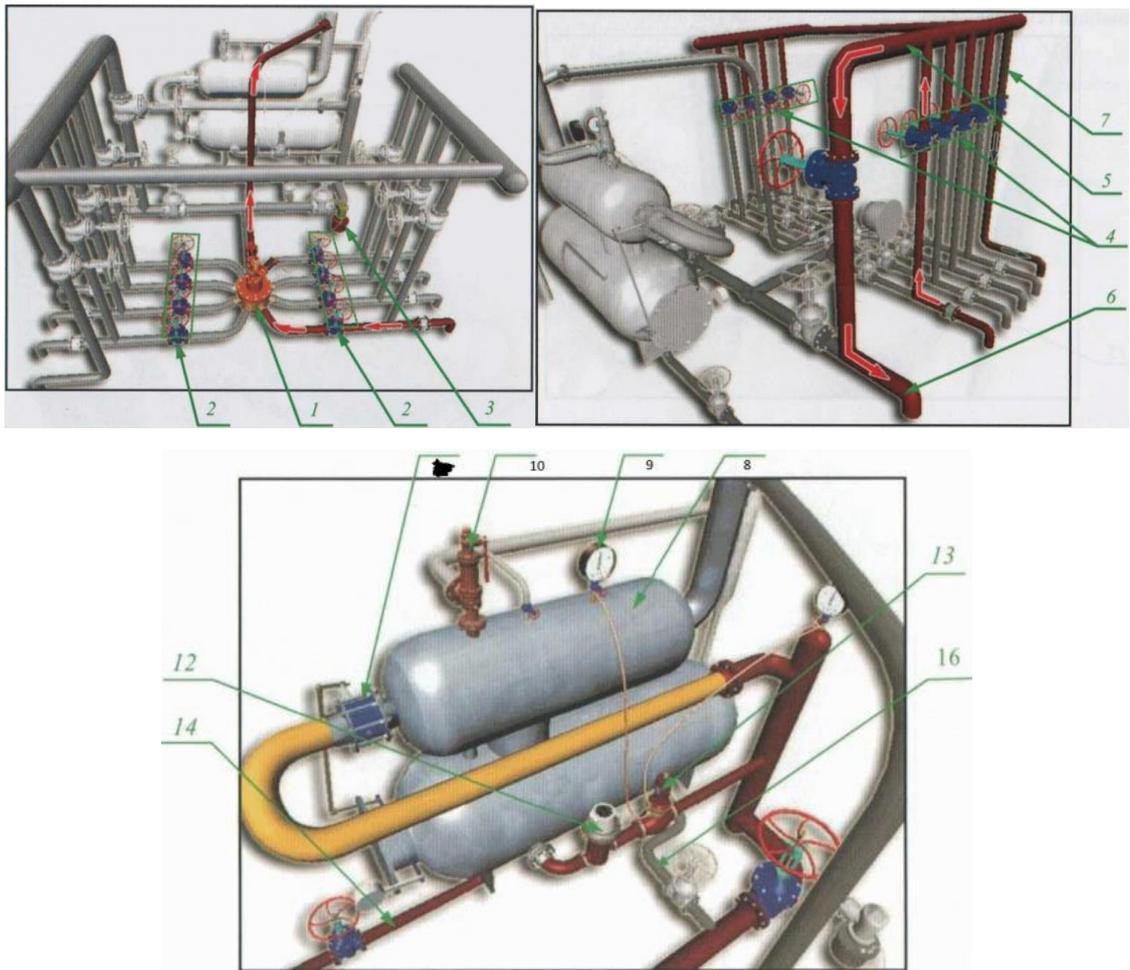


Рис. 2. Технологический блок

В технологическом блоке установлен переключатель скважин многоходовой (ПСМ) 1, к которому через нижний ряд задвижек 2 подводится продукция добывающих скважин. Автоматическое переключение ПСМ производится при помощи гидропривода 3. Система задвижек верхнего ряда 4 позволяет направлять продукцию скважин по байпасу 5 в сборный коллектор 6, минуя ПСМ, т.е. без замера. Для разрядки байпасной линии предусмотрена дренажная линия 7, выведенная в канализационный колодец либо в дренажную емкость. Основным элементом установки является емкость сепарационная 8, оснащенная контрольно-измерительными приборами 9 и пружинным предохранительным клапаном (СППК) 10. На выходе газа из ёмкости устанавливается газовая заслонка 11, а на трубопроводе выхода

жидкости — счетчик ТОР 12 и регулятор расхода 13. Для сброса грязи из емкости предусмотрена грязевая линия 16, а для слива жидкости — линия разрядки 14, выведенная в канализационный колодец, либо в дренажную емкость. Для аварийного сброса давления и разрядки ёмкости предусмотрена линия сброса 15, отводящая газ в атмосферу, а жидкость в дренажную линию.

Переключатель скважин многоходовой (ПСМ) предназначен для автоматической и ручной установки скважин на замер (рис.3).

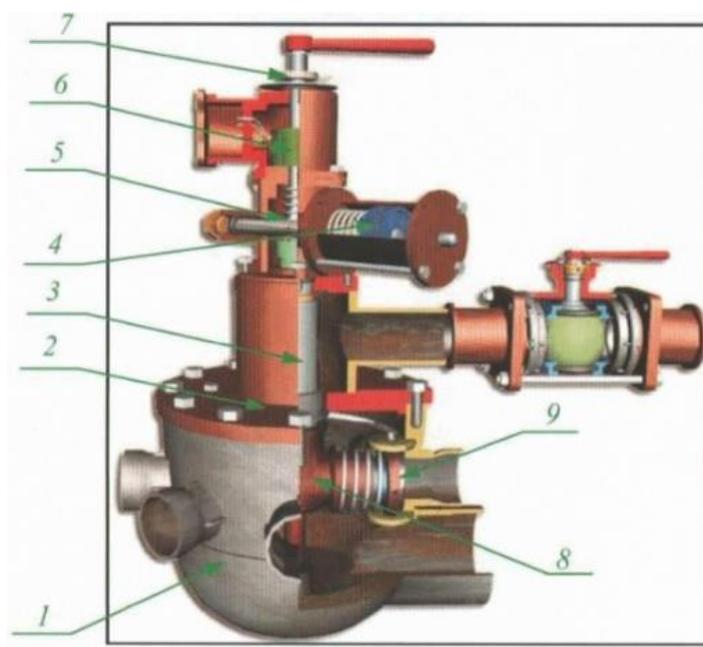


Рис. 3. ПСМ

ПСМ состоит из корпуса с патрубками 1, крышки 2 с измерительным патрубком, вала 3, поршневого привода 4 с зубчатой рейкой 5, датчика положения 6, указателя положения 7, угольника (поворотного патрубка) 8 и подвижной каретки 9. Корпус ПСМ на внутренней поверхности имеет две диаметральных канавки с выточками против каждого отверстия. По канавкам перемещаются ролики каретки. При перемещении роликов по канавкам, между резиновым уплотнением и корпусом ПСМ образуется зазор, а при попадании роликов в выточки уплотнение прижимается к корпусу пружиной, обеспечивая герметичность в замерном тракте. Жидкость из скважины, установленной на замер, проходит через каретку, угольник, патрубок с

отверстиями, установленный на валу ПСМ, и направляется на замер в ёмкость сепарационную. Жидкость с остальных скважин через выходной патрубок направляется в сборный коллектор.

Автоматическое переключение ПСМ осуществляется при помощи поршневого привода за счет давления масла, создаваемого гидроприводом.

Глава 2. Концепция Интернет Вещей (IoT) и промышленного Интернета Вещей (IIoT)

2.1. Что такое интернет вещей (IoT)

Интернет вещей (IoT) описывает сеть физических объектов - «вещей», которые встроены в датчики, программное обеспечение и другие технологии с целью соединения и обмена данными с другими устройствами и системами через Интернет. Эти устройства варьируются от обычных бытовых предметов до сложных промышленных инструментов [6].

2.2. Индустриальный интернет вещей (IIoT)

Промышленный IoT (IIoT) относится к применению технологии IoT в промышленных условиях, особенно в отношении контрольно-измерительных приборов и управления датчиками и устройствами, использующими облачные технологии. В последнее время отрасли используют межмашинную связь для обеспечения беспроводной автоматизации и управления. Но с появлением облачных и смежных технологий (таких как аналитика и машинное обучение) отрасли могут достичь нового уровня автоматизации и с его помощью создать новые модели доходов и бизнес-моделей. IIoT иногда называют четвертой волной промышленной революции или Индустрией 4.0 (рис.4) [6].



Рис.4. Индустриальный Интернет ПоТ (Industrial IoT)

2.2.1. Как работает промышленный интернет вещей

Промышленный Интернет вещей (IIoT) относится к использованию концепций IoT в промышленном производстве. В немецкоязычном мире промышленный Интернет вещей часто называют Индустрия 4.0. По сравнению с Интернетом вещей в частной сфере (IoT) основное внимание в промышленном секторе уделяется созданию сетей машин (машина-машина). Таким образом, ключевой концепцией IIoT является интеграция технологий машинного обучения и больших данных, что значительно повышает эффективность компаний. Однако сложность и требования намного выше в IIoT, чем в IoT [6].

Основными требованиями к архитектуре IIoT являются масштабируемость, возможности в реальном времени, совместимость, а также защита данных и безопасность. Датчики, исполнительные механизмы и интеллектуальные устройства, которые собирают данные (периферийные вычисления) и отправляют их на серверы (сеть). На уровне облачных вычислений данные обрабатываются с использованием интеллектуальных алгоритмов. Затем они формируют основу для автоматизированных

процессов. Крупные облачные провайдеры, такие как Microsoft, Amazon Web Services и Google, теперь предоставляют платформы IoT, которые облегчают разработку и администрирование приложений IIoT [6].

В IIoT комбинация датчиков и аналитики позволяет в режиме реального времени получать доступ к данным, которые ранее были недоступны. Результаты этих данных незамедлительно поступают в процессы по всей цепочке поставок [6]. Это делает возможным, например:

- Оптимизация процессов (например, через удаленный мониторинг)
- Большая гибкость производственных процессов
- Увеличение степени автоматизации
- Повышение эффективности работы и снижение частоты отказов
- Более быстрое выявление слабых сторон и проблем производительности
- Более точные прогнозы состояния машины и более эффективное обслуживание
- Экономия средств за счет избежания ненужного ремонта
- Лучшая доступность и меньше отказов компьютеров
- Улучшенный контроль качества и снижение уровня ошибок
- Улучшенная прозрачность благодаря всемирному доступу к машинным данным
- Улучшенное техническое обслуживание клиентов
- Разработка модных бизнес-направлений и моделей (например, дополнительные услуги, такие как дистанционное устранение неисправностей или профилактическое обслуживание)

2.3. Облачные технологии и Интернет Вещей

Среди наиболее полных определений облачных систем на сегодняшний день можно выделить следующие два:

- Облачные сервисы – это технология обработки данных, в которой программное обеспечение предоставляется пользователю как интернет-сервис, при котором от пользователя скрыта инфраструктура «облака» (облачной системы) и, поэтому, ему не требуются специальные знания и навыки для управления и использования данной «облачной» технологии [8].
- Облачные вычисления – концепция «вычислительного облака», согласно которой программы запускаются и выдают результаты работы в окно стандартного веб-браузера на локальном ПК, при этом все приложения и их данные, необходимые для работы, находятся на удаленном сервере в интернете [9].

Обычно сравнивают три модели облачной службы: программное обеспечение как услуга (SaaS), платформа как услуга (PaaS) и инфраструктура как услуга (IaaS) [7].

Программное обеспечение как услуга, также известное как облачные сервисы приложений, представляет собой наиболее часто используемый вариант для предприятий на облачном рынке. SaaS использует Интернет для доставки приложений, которыми управляет сторонний поставщик, своим пользователям. Большинство приложений SaaS запускаются непосредственно через ваш веб-браузер, а это значит, что они не требуют каких-либо загрузок или установок на стороне клиента [9]. SaaS предоставляет многочисленные преимущества сотрудникам и компаниям, значительно сокращая время и деньги, затрачиваемые на утомительные задачи, такие как установка, управление и обновление программного обеспечения. Это освобождает много времени для технического персонала, чтобы тратить на более насущные вопросы и проблемы в организации.

Сервисы облачной платформы, также известные как платформа как сервис (PaaS), предоставляют облачные компоненты для определенного программного обеспечения, в то время как используются в основном для приложений. PaaS предоставляет платформу для разработчиков, на которой

они могут опираться и использовать для создания индивидуальных приложений. Все серверы, хранилища и сети могут управляться предприятием или сторонним поставщиком, в то время как разработчики могут поддерживать управление приложениями. Независимо от размера вашей компании, использование PaaS дает множество преимуществ, в том числе: простая, экономически эффективная разработка и развертывание приложений, масштабируемость, высокая доступность, разработчики могут настраивать приложения без головной боли при обслуживании программного обеспечения, значительное сокращение объема необходимого кодирования [9].

Сервисы облачной инфраструктуры, известные как инфраструктура как услуга (IaaS), состоят из высоко масштабируемых и автоматизированных вычислительных ресурсов. IaaS полностью самообслуживается для доступа и мониторинга компьютеров, сетей, хранилищ и других служб. IaaS позволяет компаниям приобретать ресурсы по требованию и по мере необходимости, вместо того чтобы покупать оборудование напрямую. IaaS предлагает много преимуществ, в том числе: самая гибкая модель облачных вычислений, простота автоматизации развертывания хранилищ, сетей, серверов и вычислительной мощности, аппаратные покупки могут быть основаны на потреблении, клиенты сохраняют полный контроль над своей инфраструктурой, ресурсы можно приобрести по мере необходимости, высоко масштабируемый [9].

Глава 3. Состав программно-аппаратного комплекса

3.1. Контроллер «ADAM 3600»

Контроллер ADAM-3600 - универсальное устройство для сбора данных, автоматизации процессов, дистанционного управления промышленными объектами и мониторинга (рис.5). Одновременно может выступать как устройство сбора и передачи данных (УСПД), программируемый логический контроллер (ПЛК) и устройство связи с объектом (УСО). Имеет модули расширения портов ввода-вывода, а также модули беспроводных интерфейсов 4G/3G/GPRS, ZigBee, GPS, Wi-Fi. Устройство работает под управлением операционной системы RealTime Linux и процессора Cortex A8. Поддерживает открытые промышленные протоколы HTTP REST API, DNP3, ModBus, MQTT. Может быть интегрирован в любую SCADA-систему. Также поддерживает интеграцию в облачные системы IoT [10].

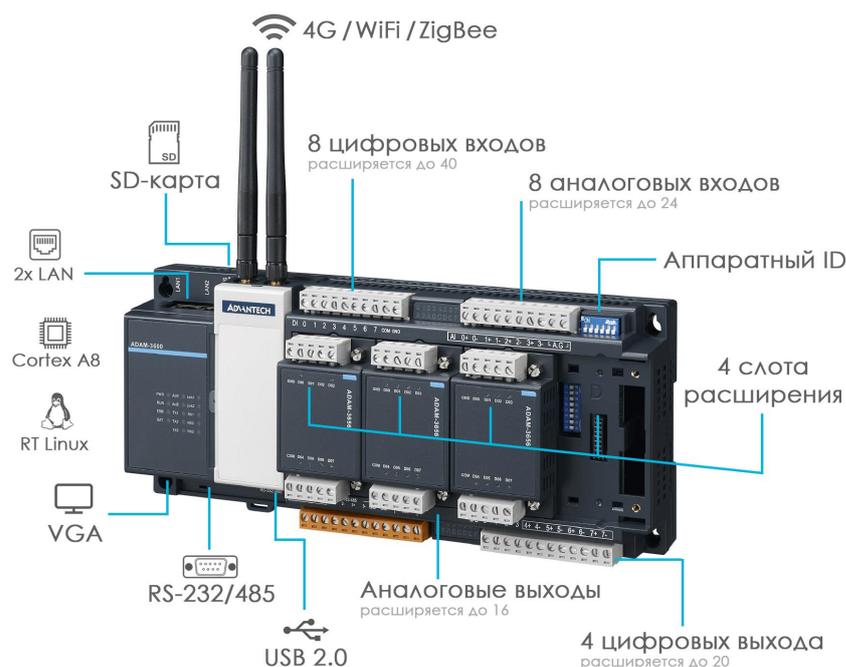


Рис.5. Логический контроллер «ADAM-3600»

Основные характеристики контроллера «ADAM3600»:

Программная часть:

- Контроллер работает под управлением ОС Linux RT 3.12;
- Поддержка открытых протоколов обмена данными: Modbus/RTU, Modbus/TCP, DNP3, МЭК 101/104.

Аппаратная часть:

- Процессор Cortex A8, память 256 МБ с пониженным энергопотреблением;
- Встроенные каналы ввода-вывода: 8xDI, 4xDO, 8xAI (дифференциальные +/- 10В, +/-2,5В, 0/4-20 мА);
- Модули расширения - до 4-х;
- Порты: 2 x Ethernet (независимые IP-адреса, IPv4/v6); COM1 RS-232/485; COM2/COM3 RS-485,USB, VGA.
- Поддержка карт памяти SD до 64 ГБ с возможностью ведения локального архива.
- Возможна установка платы расширения для беспроводных интерфейсов: miniPCI (2 слота, поддерживаемые функции беспроводной связи по GPRS, 3G, Wi-Fi, Zigbee.
- Имеется монтажное место для установки 2х антенн.
- Индикация LED: System / Serial/Ethernet/ DI/DO/Prog.

Количество выводов I/O может расширяться в зависимости от задач заказчика. В контроллер ADAM-3600 можно установить до 4 модулей расширения интерфейсов. Существует 6 типов модулей: цифровые входы/выходы, аналоговые входы/выходы, входы для термпары и резисторных датчиков (RTD) [10].

3.2. Модуль «ADAM 3624»

Модуль вывода, 4 канала аналогового вывода(ADAM-3624-AE) (рис.6).

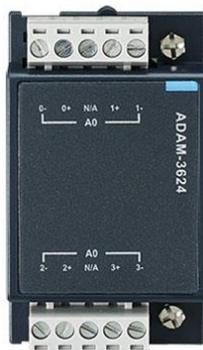


Рис.6. Модуль «ADAM-3624»

Табл.1. Технические характеристики Модуль «ADAM 3624»:

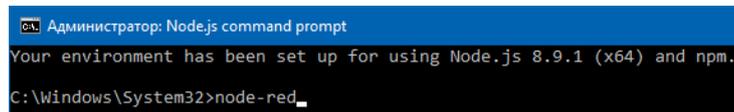
Каналов аналогового вывода	4
Диапазон выходного сигнала по напряжению	0...10 В
Диапазон выходного сигнала по току	0...20 мА, 4...20 мА
Разрядность ЦАП	12 Бит
Точность выходного сигнала	0.3 % (от измеряемого диапазона при 25°C)
Изоляция аналогового вывода	2000 В
Мощность потребления	1 Вт
Требования к температуре при работе	-40 ... 70 °С
Требования к температуре хранения	-40 ... 85 °С
Требования к влажности при хранении	5 ... 95 %

Глава 4. Языки программирования, сервисы, протоколы и среды разработки

4.1. Node-RED – мощный инструмент для создания приложений Internet of Things

Node-RED – это мощный инструмент для создания приложений IoT с уделением особого внимания упрощению соединения «проводки вместе» блоков кода для выполнения задач. Он использует подход визуального программирования, который позволяет разработчикам подключать predetermined кодовые блоки «узлы», вместе для выполнения задачи. Связанные узлы, как правило, комбинация входных узлов, узлов обработки и выходных узлов, когда они соединены вместе, называют «потoki» [12].

Пакет Node-RED устанавливается в виде модуля NPM в среду Node.js. Команда вызова из-под командной строки Node.js (рис.7).

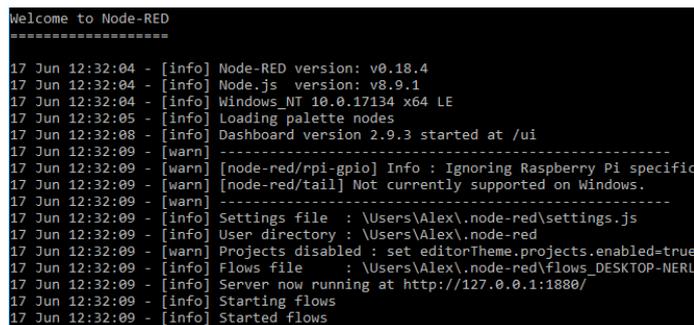


```

C:\Администратор: Node.js command prompt
Your environment has been set up for using Node.js 8.9.1 (x64) and npm.
C:\Windows\System32>node-red_
  
```

Рис.7. Командная строка

При запуске пакета, загружается в командной строке справочная информация по его состоянию (рис.8).



```

Welcome to Node-RED
*****
17 Jun 12:32:04 - [info] Node-RED version: v0.18.4
17 Jun 12:32:04 - [info] Node.js version: v8.9.1
17 Jun 12:32:04 - [info] Windows_NT 10.0.17134 x64 LE
17 Jun 12:32:05 - [info] Loading palette nodes
17 Jun 12:32:08 - [info] Dashboard version 2.9.3 started at /ui
17 Jun 12:32:09 - [warn] -----
17 Jun 12:32:09 - [warn] [node-red/rpi-gpio] Info : Ignoring Raspberry Pi specific
17 Jun 12:32:09 - [warn] [node-red/tail] Not currently supported on Windows.
17 Jun 12:32:09 - [warn] -----
17 Jun 12:32:09 - [info] Settings file   : \Users\Alex\.node-red\settings.js
17 Jun 12:32:09 - [info] User directory  : \Users\Alex\.node-red
17 Jun 12:32:09 - [warn] Projects disabled: set editorTheme.projects.enabled=true
17 Jun 12:32:09 - [info] Flows file      : \Users\Alex\.node-red\flows_DESKTOP-NERL
17 Jun 12:32:09 - [info] Server now running at http://127.0.0.1:1880/
17 Jun 12:32:09 - [info] Starting flows
17 Jun 12:32:09 - [info] Started flows
  
```

Рис. 8. Состояние сервера

Окно редактора запускается по адресу <http://127.0.0.1:1880/>. Рабочее окно Node-RED состоит из четырех главных компонентов: «шапка» (в ней находится кнопка «Deploy», кнопка главного меню), «палитра» (в ней содержатся ноды, которые можно использовать в редакторе), главная рабочая область (в ней создаются потоки) и боковая панель (рис.9).

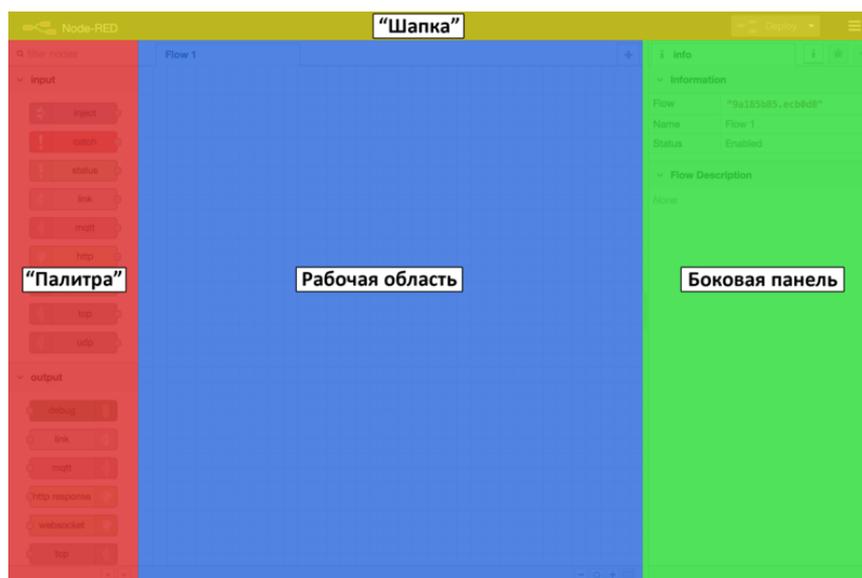


Рис. 9. Компоненты редактора

Палитра Node-RED включает набор узлов по умолчанию, которые являются основными строительными блоками для создания потоков. Все узлы включают документацию, которую вы можете видеть на вкладке «Информация» на боковой панели, после добавления узла в рабочую область. Существует три типа основных узлов: входные – формируют начальное сообщение, обработка узлов (функциональный) – позволяет каким-либо образом обрабатывать данные, выходные – заканчивают выполнение кода (рис.10).

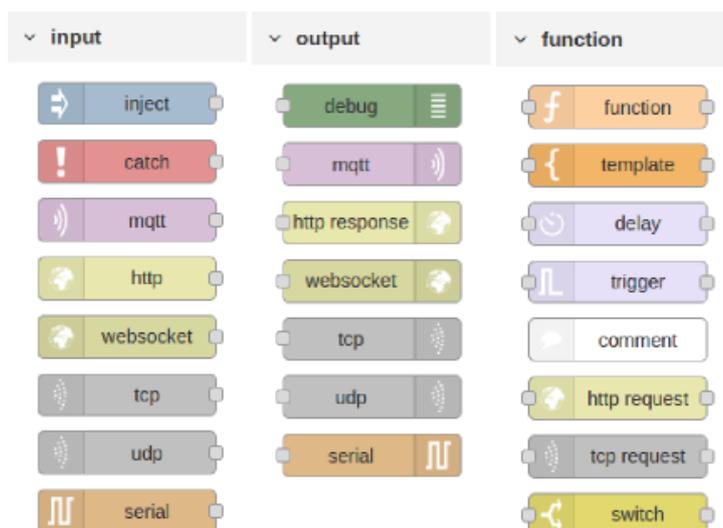


Рис. 10. Класс узлов

Основным входным узлом, является узел inject (рис.11). Сообщение, отправленное узлом Inject, имеет свойство payload.

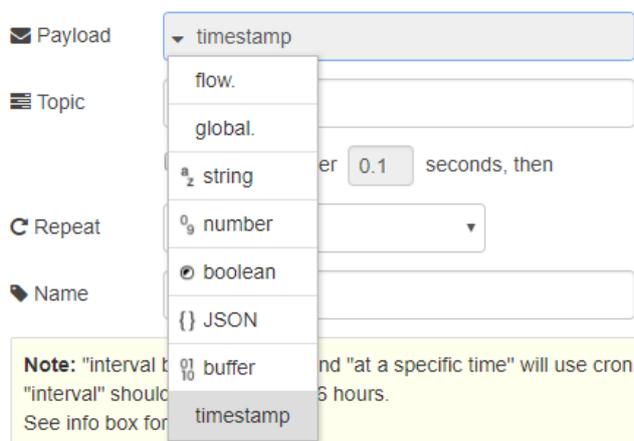
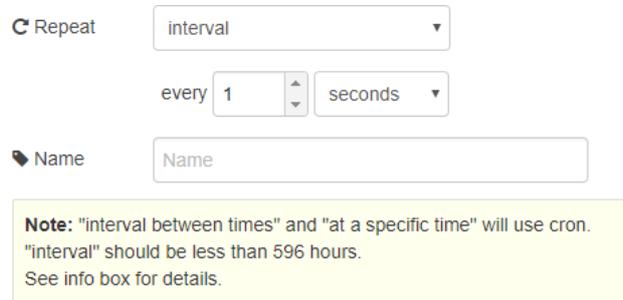


Рис. 11. Конфигурация узла inject

По умолчанию – это метка текущего времени в миллисекундах с 1 января 1970 года (UNIX-формат), также поддерживает инъекции строк, чисел, булевых объектов, объектов JavaScript или значений потока / глобального контекста. Если в потоке несколько входных узлов, в поле топик (topic), можно самому задать универсальное имя, для работы именно с ним. Узел Inject можно использовать для ручного запуска потока, нажав кнопку узла в

редакторе. Также можно использовать для автоматического запуска потока через равные промежутки времени (рис.12).



Repeat interval

every 1 seconds

Name

Note: "interval between times" and "at a specific time" will use cron. "interval" should be less than 596 hours. See info box for details.

Рис. 12. Настройка повторения

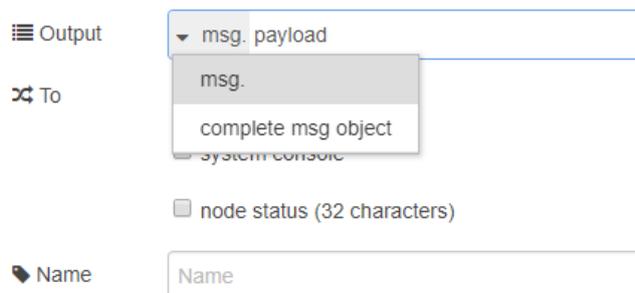
Основным функциональным узлом, является узел function. Позволяет писать скрипты на языке JavaScript (рис.13). Для вывода результата последней строкой по умолчанию возвращается объект (msg).



```
Function
1
2 return msg;
```

Рис. 13. Конфигурация узла function

Основным выходным узлом, является узел debug. Он работает в двух режимах, в первом он возвращает в окно отладки значение (payload) входного сообщения, а во втором все сообщение, т.е. объект (рис.14).



Output msg. payload

To msg.

complete msg object

system console

node status (32 characters)

Name

Рис. 14. Конфигурация узла debug

Так как Node-RED был разработан как IoT продукт, то по умолчанию в нем есть необходимый узлы для работы с протоколом MQTT, с параметрами работы (рис. 15).

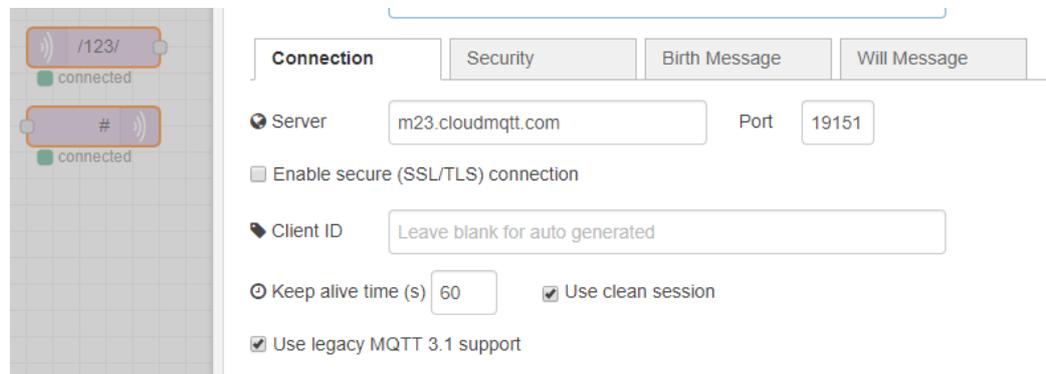


Рис. 15. Конфигурация узла MQTT

В Node-RED можно быть как издателем, так и подписчиком модели MQTT.

Являясь пакетом с открытым исходным кодом, возможно уже существует решение, которое вам нужно, созданное участниками сообщества Node-RED в виде библиотеки узлов на сайте Node-RED Library. Например, работа с облачными сервисами или СУБД (система управления базами данных), добавление библиотек осуществляется через командную строку Node.js или прямым образом из Node-RED в настройках, через вкладку установки (рис. 16). Во вкладке узлы можно отключить или удалить не нужные пакеты.

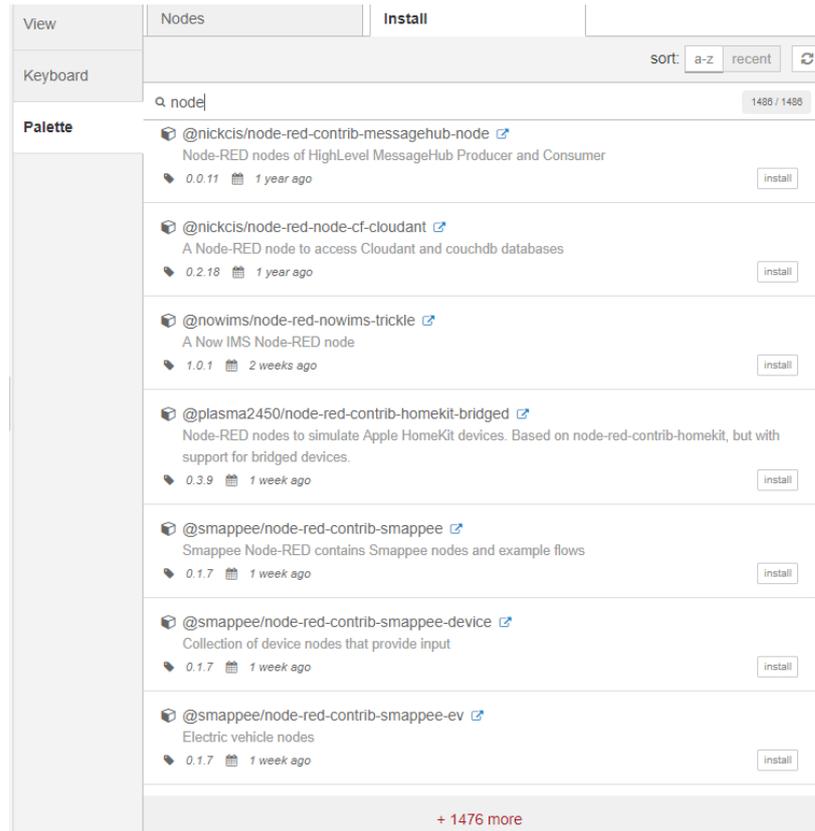


Рис. 16. Добавление сторонних узлов

Экспорт и импорт потоков. Выбирая готовые потоки их можно сохранять как библиотеки или как отдельный узел, а также экспортировать/импортировать их в JSON-формате (рис. 17).

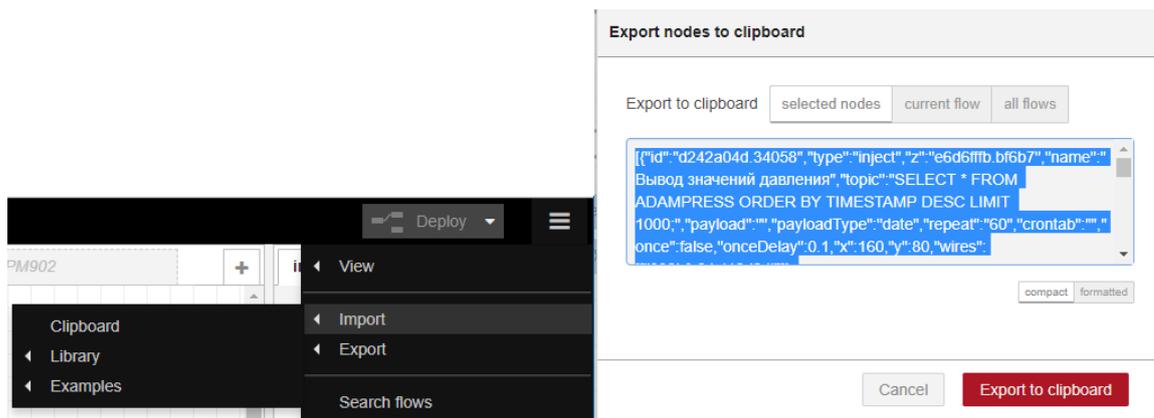


Рис. 17. Экспорт/импорт программ

4.2. MQTT (Message Queue Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) - это протокол для обмена сообщениями, который предоставляет сетевым клиентам с ограниченными ресурсами простой способ распространения телеметрической информации. Данный протокол использует шаблон связи «публикатор / подписчик» и используется для связи между устройствами.

Протокол MQTT охватывает два субъекта: клиент и брокер. Брокер MQTT - это сервер, а клиенты - подключенные устройства. Когда устройство или клиент хочет отправить данные на сервер - это называется публикацией.

Если соединение подписывающего клиента с брокером разрывается, то брокер будет буферизировать сообщения и отправлять их подписчику, когда он снова будет в сети. Если соединение клиента публикации с посредником отключено без уведомления, то посредник может закрыть соединение и отправить подписчикам кэшированное сообщение с инструкциями издателя.

MQTT является одним из наиболее часто используемых протоколов, касающихся IoT. MQTT позволяет устройствам IoT с ограниченными ресурсами отправлять или публиковать информацию по заданной теме на сервер, который выполняет функции посредника сообщений MQTT. Затем брокер передает информацию тем клиентам, которые ранее подписались на эту тему. Для человека тема выглядит как иерархический путь к файлу. Клиенты могут подписаться на определенный уровень иерархии темы или использовать подстановочный знак для подписки на несколько уровней [13].

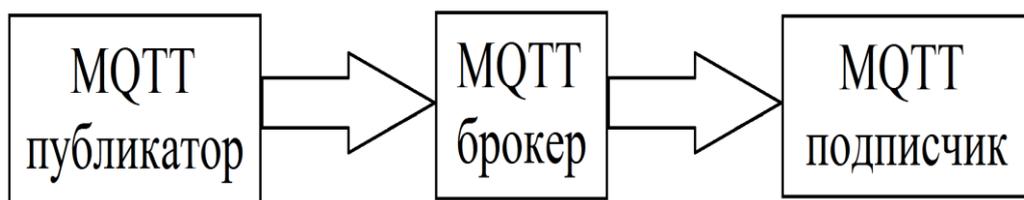


Рис.18. Структурная схема протокола MQTT

4.3. Multiprog express 5.35 - программируемая система

Advantech EdgeLink Studio — программа, работающая под операционными системами Windows XP и Windows 7, в которой реализован следующий функционал:

- Интерфейс настройки проекта без подключения к сети; удаленное развертывание конфигурации на основе настраиваемого идентификатора сети (NodeID)
- Простое сопоставление проектных тегов внешним входам/выходам, простое присвоение тегов службам Modbus и DNP3
- Пользовательская настройка каждого входного-выходного диапазона по отдельности; поддержка калибровки аналогового ввода для входов/выходов как самого изделия, так и расширительных модулей
- Способность EdgeLink Studio настраивать передачу данных через Ethernet, WiFi, 3G, GPRS
- Способность изделия ADAM-3600 работать сервером Modbus/RTU, Modbus/TCP и DNP 3.0;
- Пользовательский выбор протокола

Поддержка Advantech EdgeLink Studio удаленного мониторинга состояния обмена данными через последовательные и Ethernet-порты Advantech EdgeLink Studio можно загрузить на сайте службы поддержки Advantech.

“Multiprog Express” - высокоэффективный, инструмент для программирования контроллеров (ПЛК). Среда разработки программ позволяет обрабатывать несколько конфигураций и ресурсов в рамках одного проекта, включает в себя библиотеки и располагает системой отладки для симуляции ПЛК-приложений на компьютере. Система программирования позволяет запустить программу в ПЛК и настроить/изменить значение переменной (функция Force/Overwrite), предлагает функции для различных

этапов разработки приложения ПЛК. Система программирования обеспечивает удобное управление с использованием масштабирования, прокрутки, настраиваемых панелей инструментов, операций перетаскивания, менеджера ярлыков и закрепляемых окон.

Есть возможность применения протокола TCP/IP:

- запись приложения в ПЛК;
- дистанционное чтение/запись переменных при помощи OPC Сервера.

“Multiprog Express” разработан в соответствии со стандартом МЭК. Поддерживает программирование на всех языках стандарта IEC 61131, таких как текстовые языки ST (структурированный текст) и IL (список инструкций), а также графические языки FBD (диаграммы функциональных блоков), LD (лестничные диаграммы) и SFC (последовательное программирование). Он основан на стандарте IEC 1131-3 и включает в себя полный спектр функций IEC.

4.4. Платформа Edgelinek

Advantech EdgeLink Studio — программа, работающая под операционными системами Windows XP и Windows 7, в которой реализован следующий функционал:

- Интерфейс настройки проекта без подключения к сети; удаленное развертывание конфигурации на основе настраиваемого идентификатора сети (NodeID)
- Простое сопоставление проектных тегов внешним входам/выходам, простое присвоение тегов службам Modbus и DNP3
- Пользовательская настройка каждого входного-выходного диапазона по отдельности; поддержка калибровки аналогового ввода для входов/выходов как самого изделия, так и расширительных модулей

- Способность EdgeLink Studio настраивать передачу данных через Ethernet, WiFi, 3G, GPRS
- Способность изделия ADAM-3600 работать сервером Modbus/RTU, Modbus/TCP и DNP 3.0;
- Пользовательский выбор протокола
- Поддержка Advantech EdgeLink Studio удаленного мониторинга состояния обмена данными через последовательные и Ethernet-порты Advantech EdgeLink Studio можно загрузить на сайте службы поддержки Advantech.

Система осуществляет сбор, предварительную обработку и передачу данных, используя ресурсы модуля и интегрированной облачной платформы управления. Такой подход дает возможность внедрить сеть постоянного мониторинга, образованную многочисленными измерительными элементами, и выполнить высокопроизводительные операции с данными. Данные собираются по сети от различных сенсоров и передаются на ADAM-3600, затем суммируются, нормируются, калибруются (или пре-обрабатываются другим образом) через Advantech WISE-PaaS/EdgeLink (рис. 19). В конечном итоге данные передаются по протоколу MQTT в облачную платформу Azure Cloud Services с установленными интервалами [15].

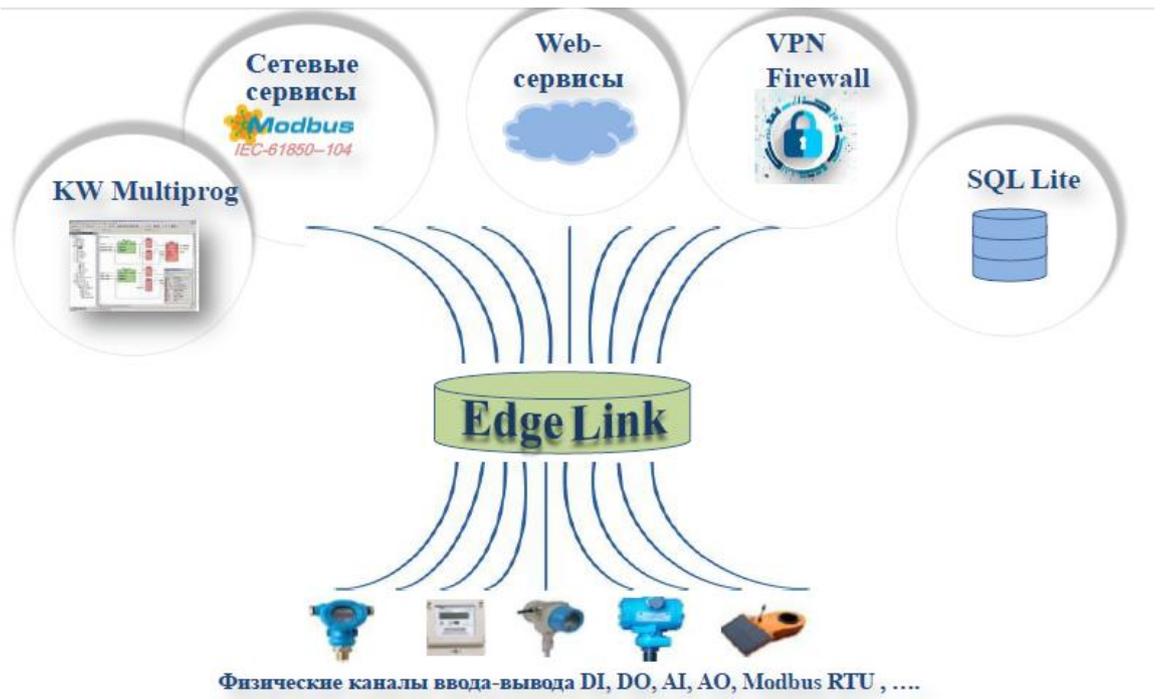


Рис.19. ПО Advantech EdgeLink - ресурс контроллера ADAM-3600

Глава 5. Описание прикладного программного обеспечения

5.1. Описание структурной схемы

Структурная схема состоит из имитационной модели поведения скважины, логического контроллера «ADAM 3600», облачного сервиса «CloudMQTT», компьютера (рис.20).

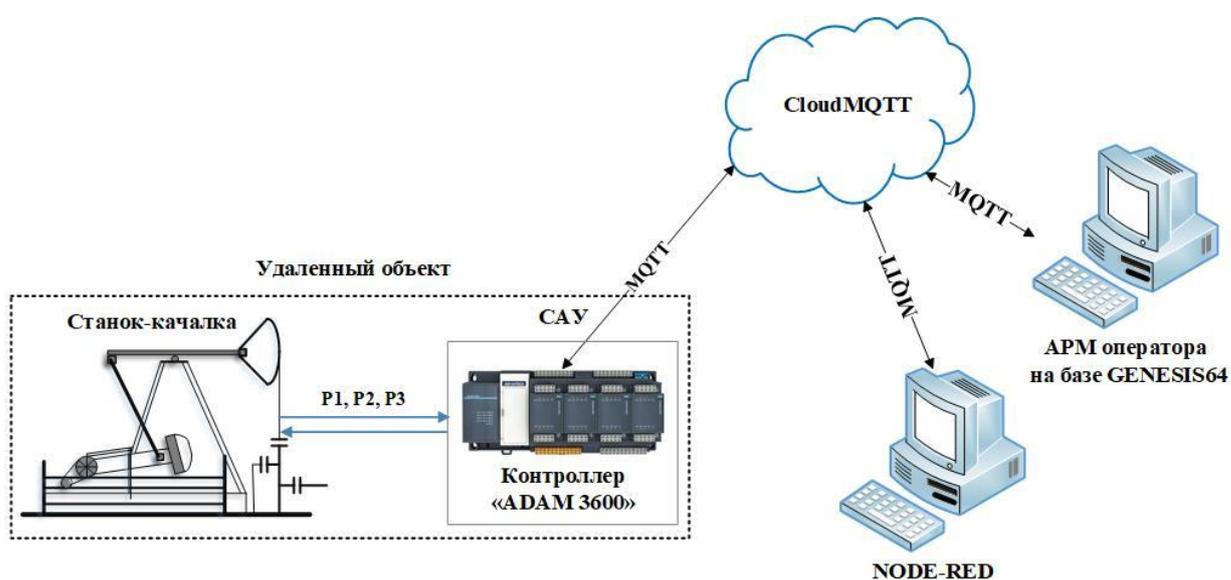


Рис.20. Структурная схема комплекса

В данной структурной схеме имитационную модель поведения скважины (СКН) выполняет модуль «ADAM 3624», который имеет 4 канала аналогового вывода. В нашем случае модуль запрограммирован так, чтобы передавал три значения тока 0-20mA, значения меняются синусоидально. Полученные значения, логический контроллер «ADAM 3600» при помощи математических формул преобразует в давление. Контроллер «ADAM 3600» поддерживает протокол mqtt и это позволяет обмениваться данными между контроллером и облачным сервисом (CloudMQTT). Далее разработанное клиентское приложение на Node-RED, позволяет запросить данные давления

для отслеживания состояния скважины в реальном времени и предназначено для дистанционного управления. Так же для комплексной отладки системы использовался АРМ оператора на базе GENESIS64.

5.2. Разработка ПО контроллера скважины СКН на языке ST для ADAM-3600.

Программное обеспечение было разработано на платформе Multiprog, который поддерживает 5 вариантов программирования.

Данное ПО состоит из трех частей написанные на языке ST: два функциональных блока (SAU, well) и основная программа (skn), позволяющая управлять и вызывать функциональные блоки (рис. 21).

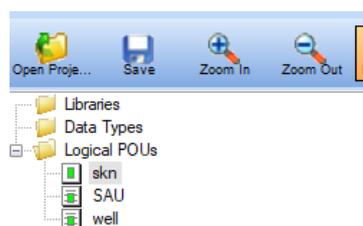


Рис.21. Дерево проекта

Рассмотрим функциональный блок well (рис. 22). Данная программа была написана для модуля «ADAM 3624», так как в данной работе имитационную модель поведения скважины выполняет именно этот модуль. Программа выполняет функцию передачи данных на логический котроллер, включение и выключение скважины, так как модуль имеет ограничение в передаче тока (0-20 mA), это было предусмотрено. На рисунке представлено часть программы, полная версия в приложение А.

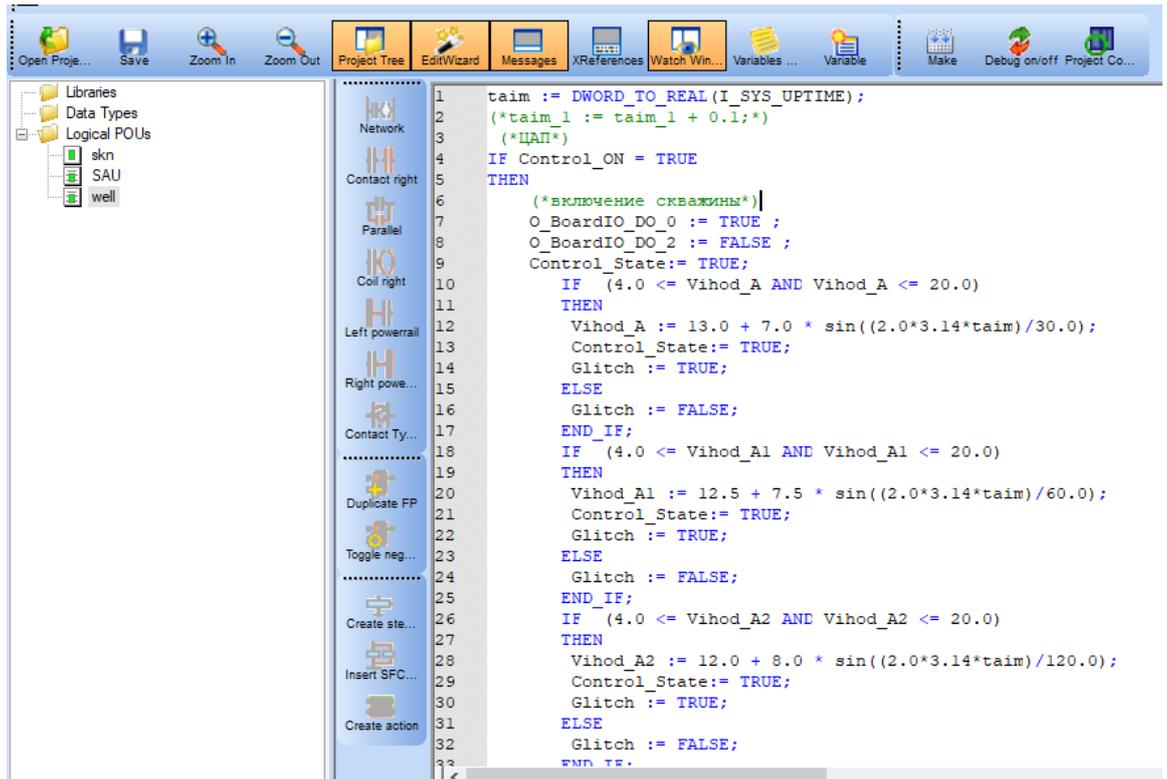


Рис.22. Функциональный блок well

Второй функциональный блок SAU (рис. 23), предназначен для преобразования полученных сигналов в сигналы давления (Атм). Данные передаются с модуля «ADAM 3624» на логический контроллер ADAM-3600. Для преобразования сигналов в давление используется полином третьего порядка.

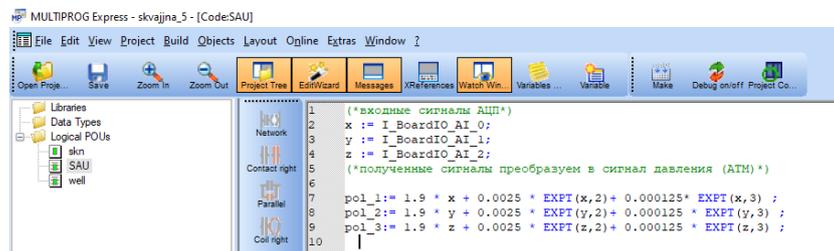


Рис.23. Второй функциональный блок

Рассмотрим программу skn (рис. 24). Она позволяет вызывать и управлять входными и выходными сигналами функциональных блоков. В

данной программе были созданные переменные (Tag), которые были использованы для дистанционного управления.

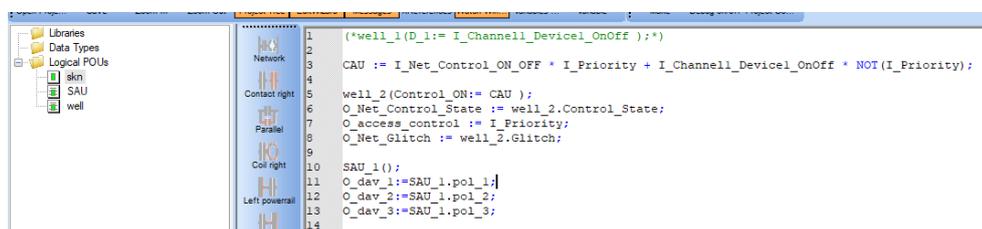


Рис.24. Программа skp

Выше была описана основная программа, которая использовалась в данной работе. Так как нам необходима связать написанную программу с контроллером ADAM 3600 произведем настройку и конфигурацию для ресурса TagLink:

1. Для настройки аппаратной части заходим в пункт Hardware и в настройках ресурсов выбираем тип коммуникации, в данном случае TCP/IP и вводим IP адрес логического контроллера (рис. 25), а в свойствах ресурсов “Resource properties” выбираем TagLink (рис. 26).

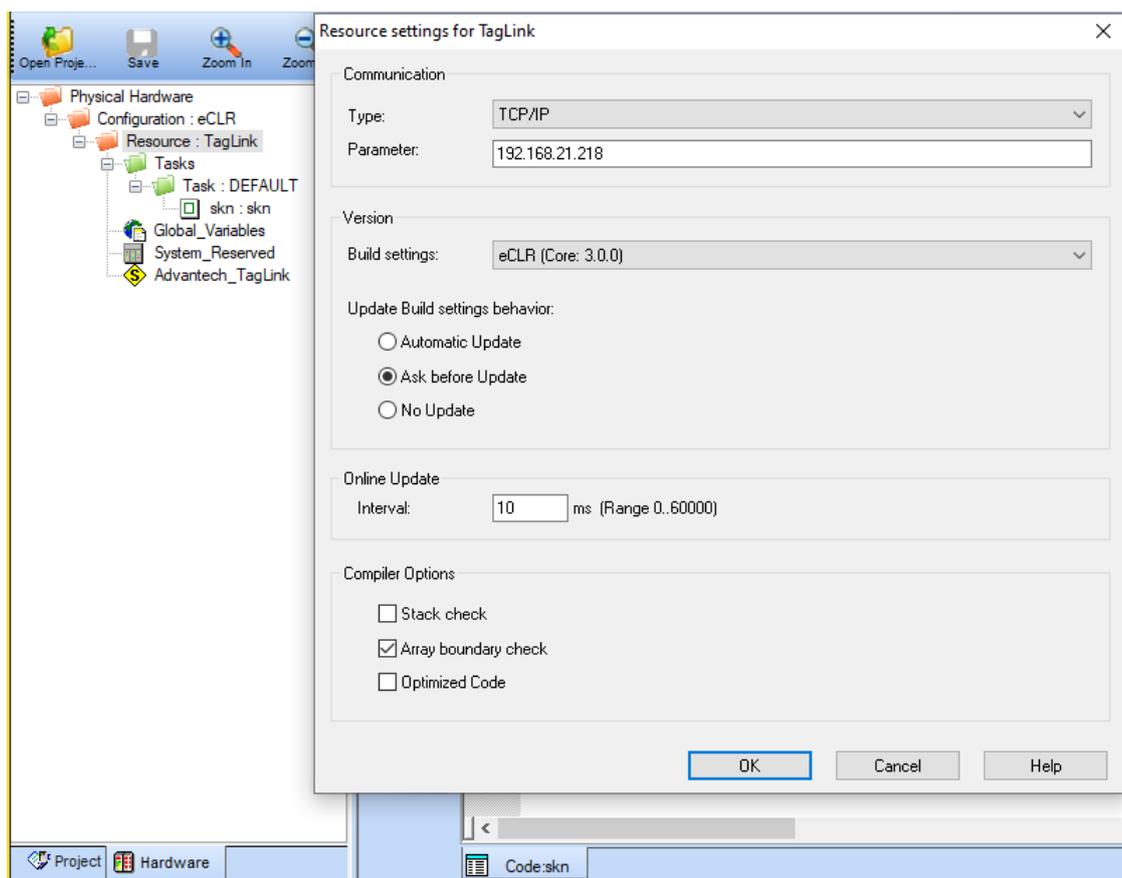


Рис.25. Настройки ресурсов “Resource settings”

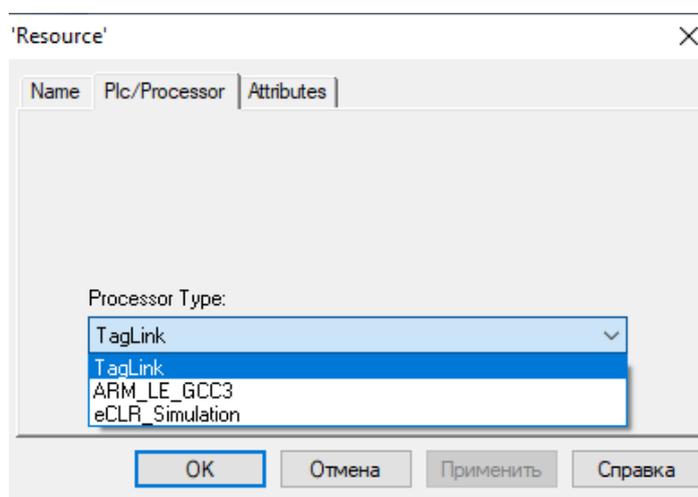


Рис.26. Свойства ресурсов “Resource properties”

2. Так как в нашем случае конфигурация портов контроллера «ADAM 3600» производится при помощи утилиты Advantech EdgeLink Studio, добавим глобальные переменные из этого проекта, которые описаны в следующем пункте (5.3). Для добавление глобальных переменных, в дереве проекта необходимо перейти во вкладку “Advantech_TagLink” и

найти сконфигурированный проект, который был создан в программе “Advantech EdgeLink Studio” (рис. 27).

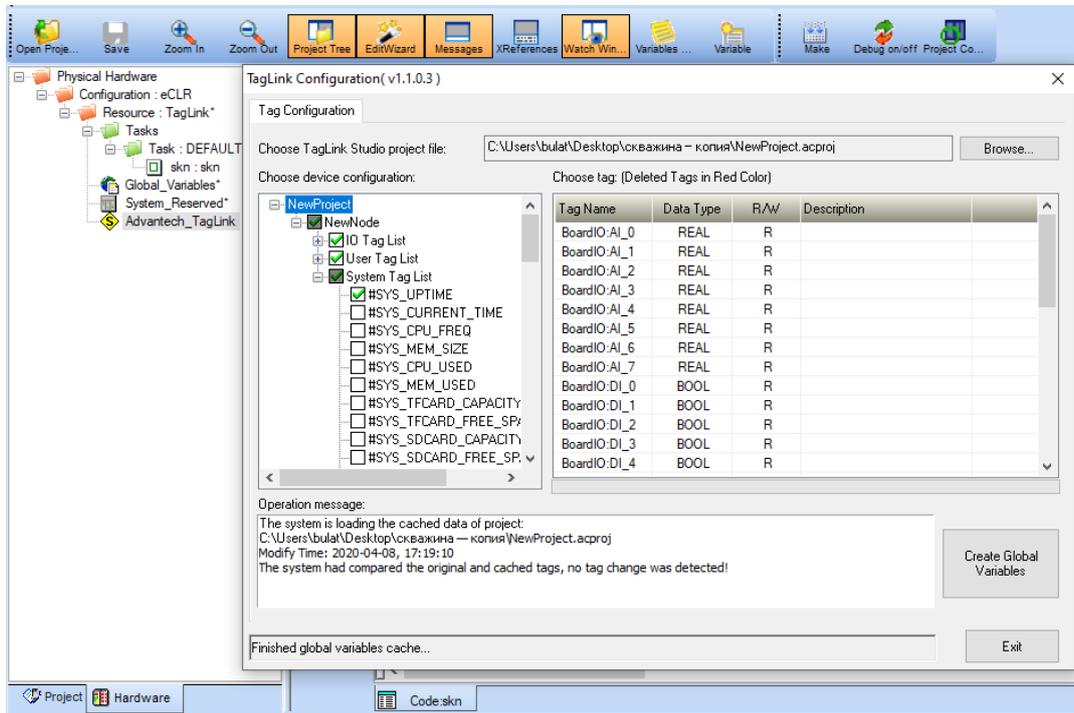


Рис.27. Импорт переменных из “EdgeLink” – конфигурация “TagLink”

Теперь выбираем переменные, которые нам необходимы для взаимодействия с контроллером, это:

- входные и выходные теги логического контроллера (рис. 28)

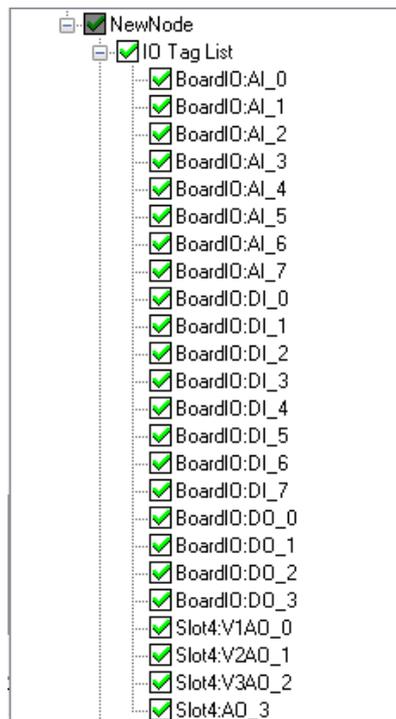


Рис.28. Список выбранных переменных “IO Tag” из “EdgeLink”

- пользовательские теги, для управления и передачи данных (рис. 29)

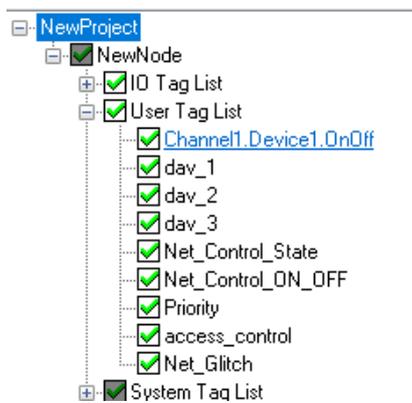


Рис.29. Список выбранных переменных “User Tag” из “EdgeLink”

- системные теги (рис. 30)

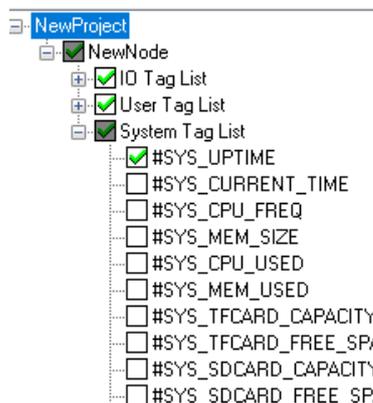


Рис.30. Список выбранных переменных “System Tag” из “EdgeLink”

Далее нажимаем на кнопку “Create Global Variables”, чтобы создать глобальные переменные.

5.3. Конфигурация контроллера «ADAM 3600» утилитой EdgeLink

Система EdgeLink позволяет осуществлять предварительную обработку, сбор и передачу данных, используя ресурсы модуля и интегрированной облачной платформы управления.

Конфигурация портов контроллера «ADAM 3600» производится при помощи утилиты Advantech EdgeLink Studio. Для конфигурации портов были выполнены следующие этапы:

1. Запускаем программу Advantech EdgeLink Studio и создаём новый проект (Create project). Далее появится диалоговое окно, где нужно ввести название проекта, его описание и выбрать папку для хранения (рис. 31).

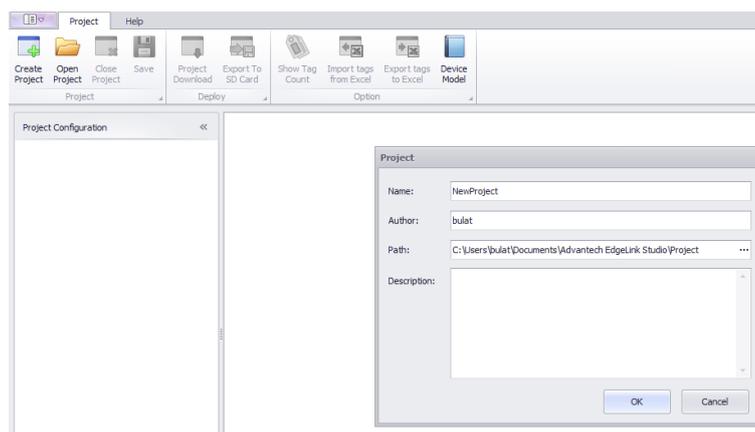


Рис.31. Создание нового проекта

2. Для добавления нового устройства, щелкаем правой клавишей мыши на название проекта и выберем пункт *Add Device*. Далее задаем имя устройства в проекте, модель контроллера, пароль для загрузки проекта, идентификацию устройства в сети для загрузки проекта: по ip-адресу, ip-адрес устройства (рис. 32). После добавления устройства, появятся основные разделы конфигурации (рис. 33).

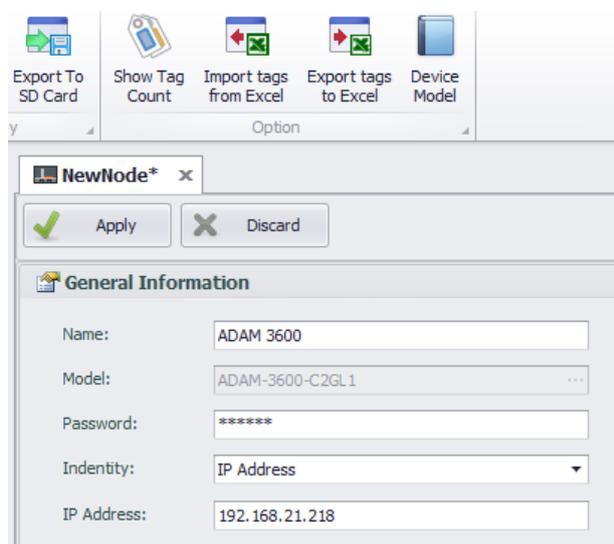


Рис.32. Добавление устройств и правка

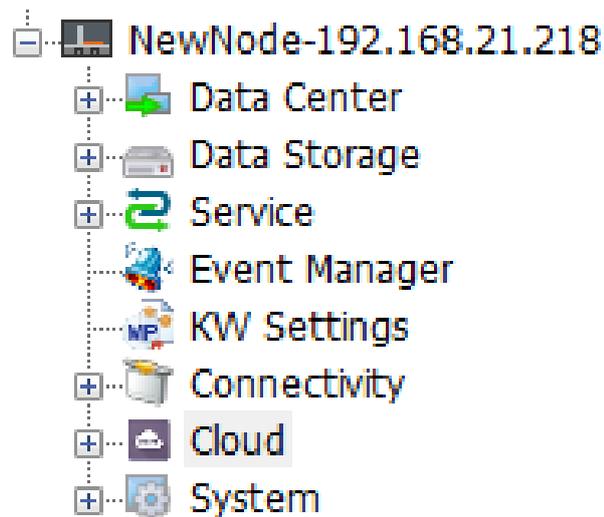


Рис.33. Разделы конфигурации

3. Как говорилось ранее, модель скважины выполняет модуль «ADAM 3624», который имеет 4 канала аналогового вывода. Для добавления этого модуля нажимаем на вкладку ADAM-3600-C2GL1A1E и в пункте Slot4 выбираем ADAM-3624. Далее задаем возможный диапазон изменения тока (0-20 mA) (рис. 34). Также во вкладке, относящейся к модулю 3624, занимающей место Slot4, создаются переменные для аналогового вывода (рис. 35).

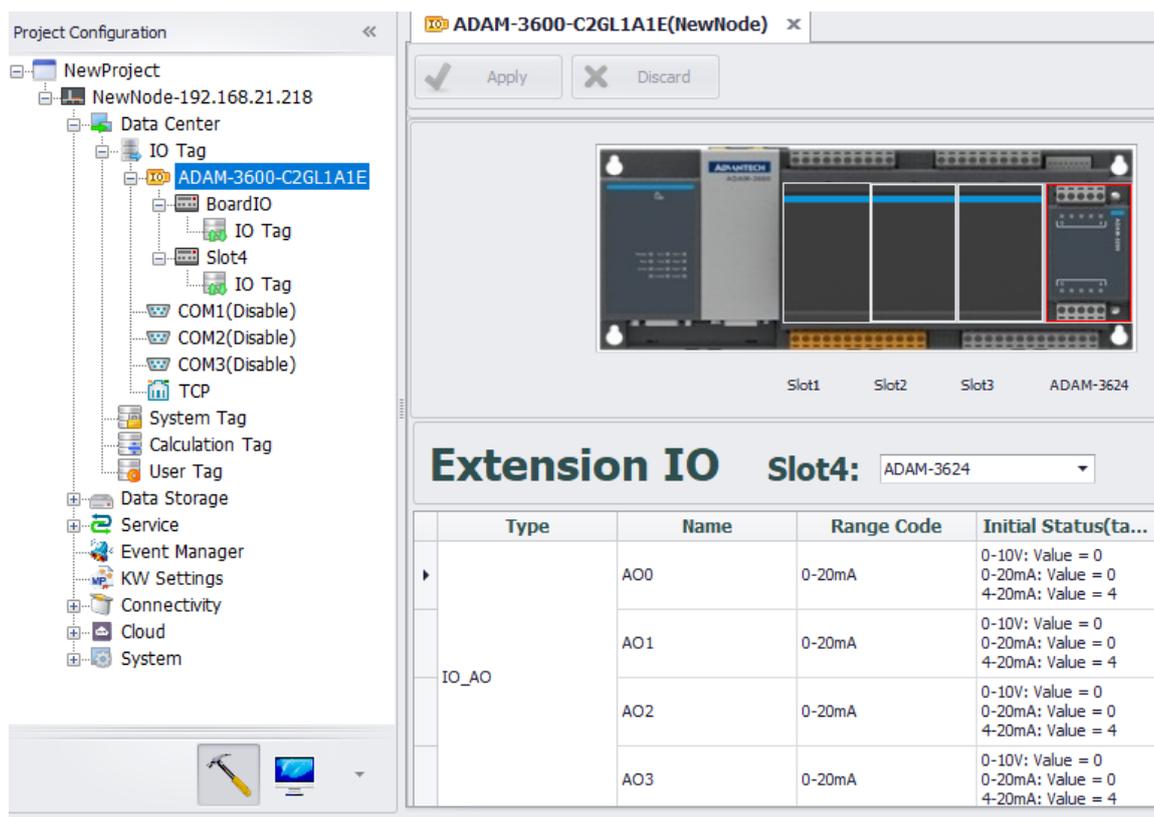


Рис.34. Добавление модуля «ADAM-3624»

Name	Data Type	Source	Initial Val...	Scan Rate	Address	Conversio...	Scale Type	Read Write
Slot4:V1AO_0	Analog		0	1	AO.0	N/A	No Scale	Read/Write
Slot4:V2AO_1	Analog		0	1	AO.1	N/A	No Scale	Read/Write
Slot4:V3AO_2	Analog		0	1	AO.2	N/A	No Scale	Read/Write
Slot4:AO_3	Analog		0	1	AO.3	N/A	No Scale	Read/Write

Рис.35. Создание переменных для аналогового вывода

4. Для управления и передачи данных были введены пользовательские теги: три давления (dav_1, dav_2, dav_3), включение и выключение скважины (Net_Control_NO_OFF), уведомление об аварии при повышении давления на скважине (Net_Glitch), режим скважины: вкл. или выкл. (Net_Control_State), ограничение доступа (Priority), выполнение команды ограничение доступа (access_control) (рис. 36).

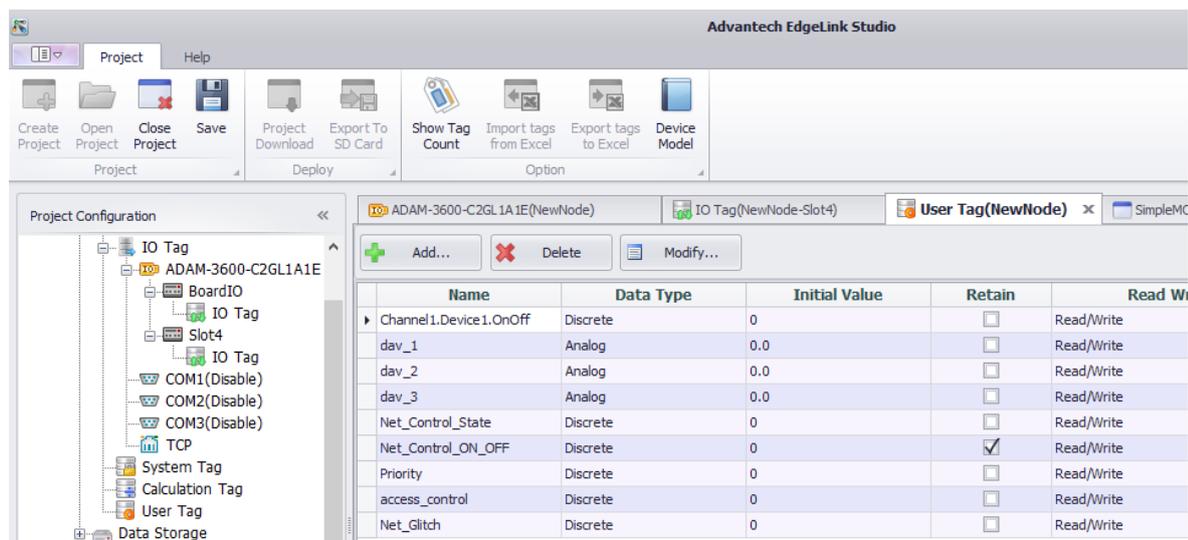


Рис.36. Настройка пользовательских тегов

5. В данной работе для дистанционного управления используется протокол mqtt. Поэтому во вкладке “Cloud” выбираем пункт SimpleMQTT и указываем: слева – данные брокера (домен, порт, логин, пароль), а справа те теги, которые нам необходимы (рис. 37).

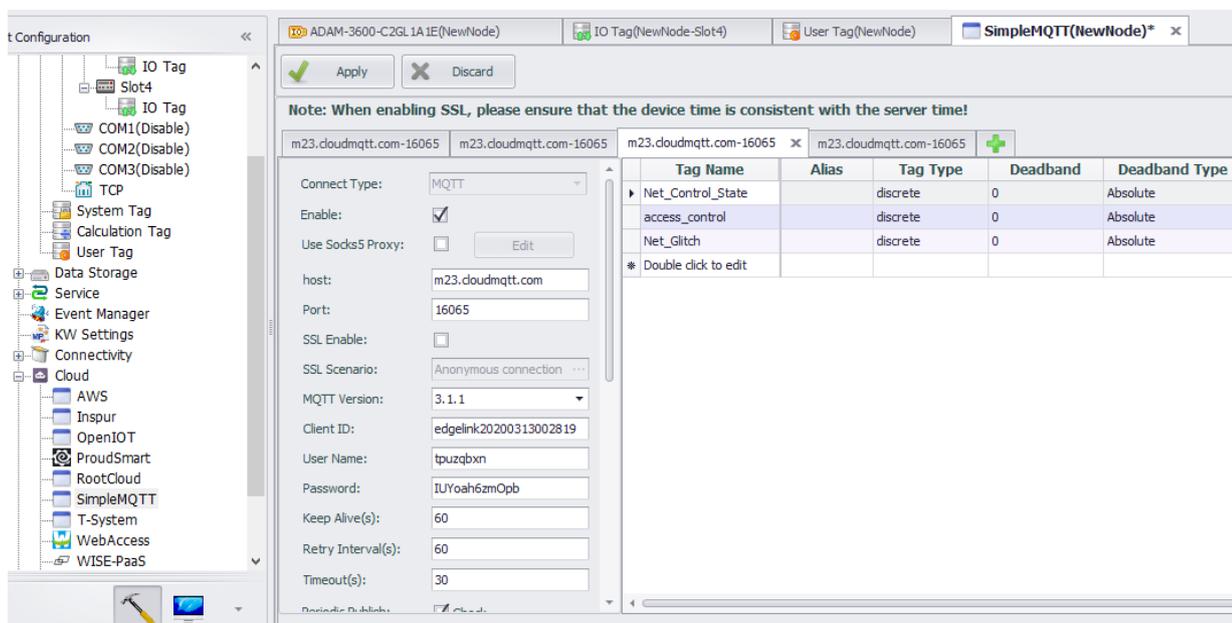


Рис.37. Конфигурация SimpleMQTT

6. Сохраняем и загружаем проект в контроллер ADAM 3600.

5.4. Разработка клиентской части с помощью Node-RED

Для разработки клиентской части, было использовано ПО Node-RED и базовые знания языка программирования JavaScript. Node-RED – это мощный инструмент для создания приложений IoT с уделением особого внимания упрощению соединения «проводки вместе» блоков кода для выполнения задач.

Создание клиентской части включала в себя следующие этапы:

1. Запускаем Node-RED, подробнее в теоретической части.
2. Добавляем необходимые узлы через палитру, поисковым запросом: `node-red-contrib-mqtt-broker` - сервер MQTT Broker на Node-RED, `node-red-dashboard` - для быстрого создания панели управления, позволяющей в реальном времени управлять и мониторить потоки данных (рис. 38).

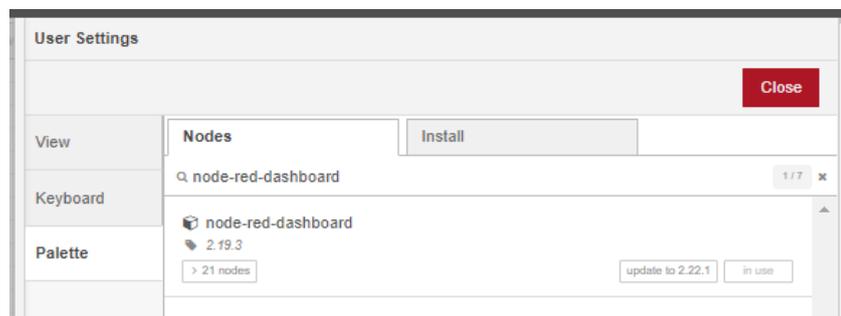


Рис.38. Палитра Node-RED

3. Рассмотрим первый поток (рис. 39). Первый нод отвечает за получение данных с контроллера по протоколу mqtt, для конфигурации этого нода нужно задать данные (домен, порт, логин, пароль). Полученные данные в JOSON формате обрабатываются, с помощью функционального узла. Настройка отображения осуществляется во вкладке dashboard, где узлы привязываются к нужной вкладке, задаются их размеры.

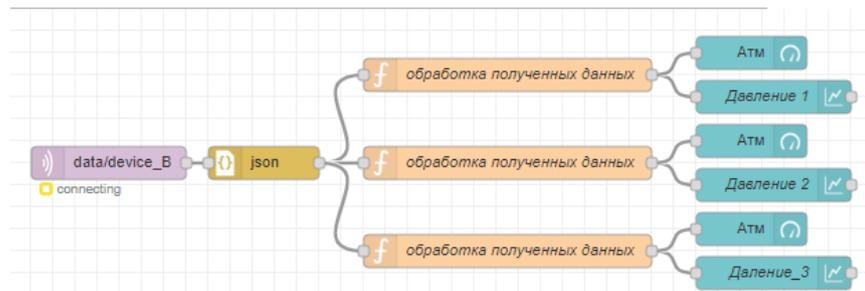


Рис.39. Первый поток

4. Второй поток отвечает за режим скважины (вкл, выкл) (рис. 40). Третий поток уведомляет об ограничении доступа к управлению (рис. 41), четвертый поток уведомляет об аварии на станции (рис. 42).

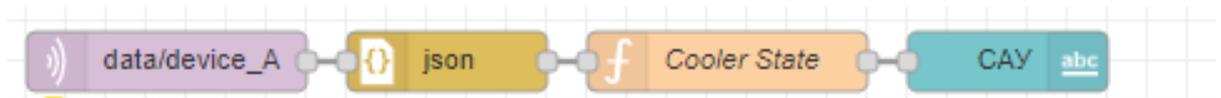


Рис.40. Второй поток



Рис.41. Третий поток

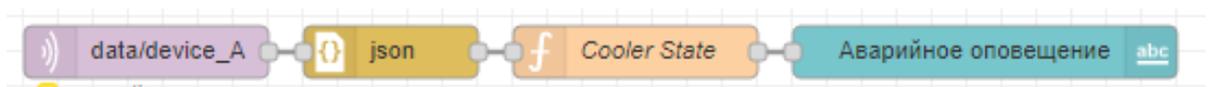


Рис.42. Четвертый поток

5. Пятый и шестой поток предназначен для управления скважиной (рис. 43).

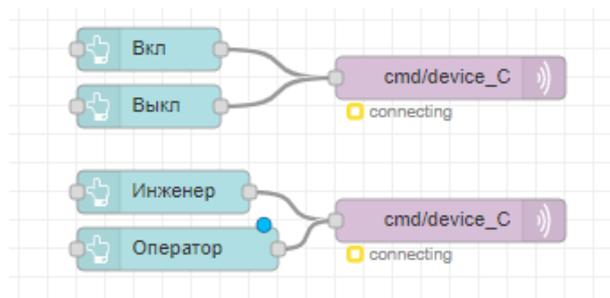


Рис.43. Пятый и шестой поток

6. Итоговая экранная форма для симулированных данных, отображаемых в реальном времени будет иметь вид, рисунок 44.

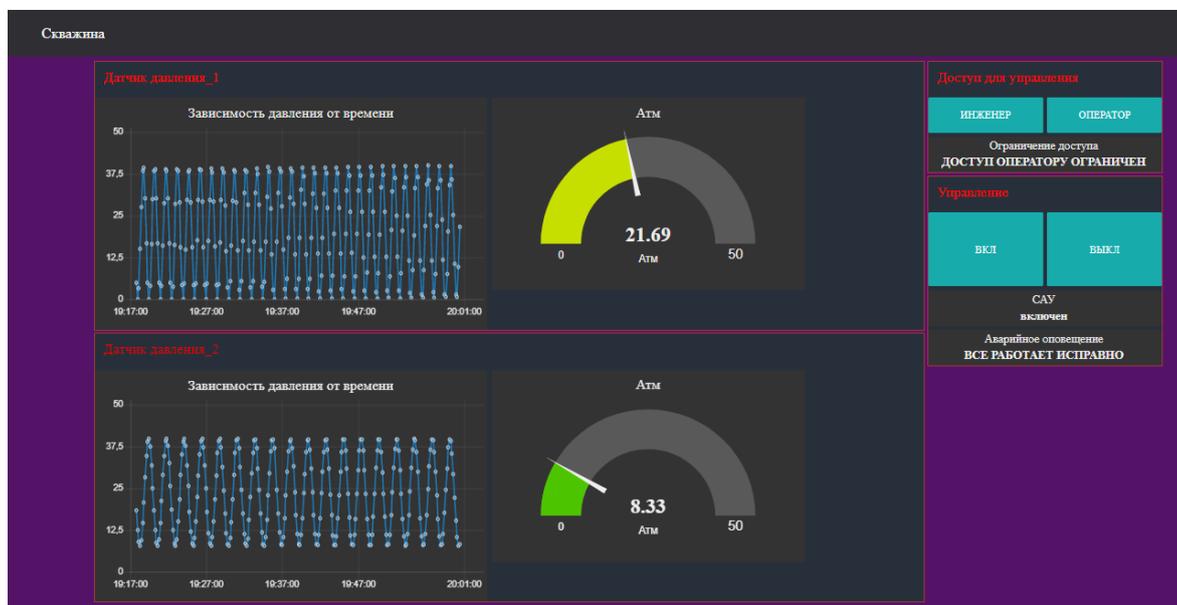


Рис.44. Экранная форма для симулированных данных

5.5. Комплексное тестирование программного обеспечения

После разработки и отладки ПО для АСК, необходимо было провести комплексное тестирование приложения на имитационной модели поведения СКН. Как говорилось ранее, модель поведения СКН была разработана на основе логического контроллера ADAM-3600.

Эксперимент предполагался как осуществление автоматизированного отслеживания текущих данных и управления СКН. Скважина СКН включает в себя буферное давление, которая меняется от 4 до 20 мА, затрубное давление, так же меняется от 4 до 20 мА, сигнальный мономер (защита по давлению). Была создана система автоматического управления скважины, которая отвечает за состояние скважины (вкл, выкл), аварийное отключение, управление (пуск/стоп), защиту по давлению. Все данные с контроллера скважины передаются в облачный сервис CloudMQTT по протоколу mqtt.

Разработанное клиентское приложение на Node-RED, позволяет запросить данные давления и вывести их на экран в реальном времени. Так же приложение предназначено для дистанционного управления и отслеживания состояния скважины в реальном времени.

Далее рассмотрим, более подробно, экранную форму на Node-RED, здесь мы можем наблюдать следующее (рис. 45):

- Три графика давления, меняющиеся в реальном времени.
- Три датчика давления.
- Доступ для управления, позволяет ограничить доступ к управлению оператору.
- состояние скважины (вкл, выкл).
- управление скважиной (пуск/стоп).
- аварийное оповещение и автоматическое отключении скважины.

Все три графика показывают зависимость давления от времени (Атм). На графиках можно увидеть резкие спады, это произошло из-за того, что скважина выключена. Для комплексной отладки системы использовался АРМ оператора на базе GENESIS64 (рис. 46).

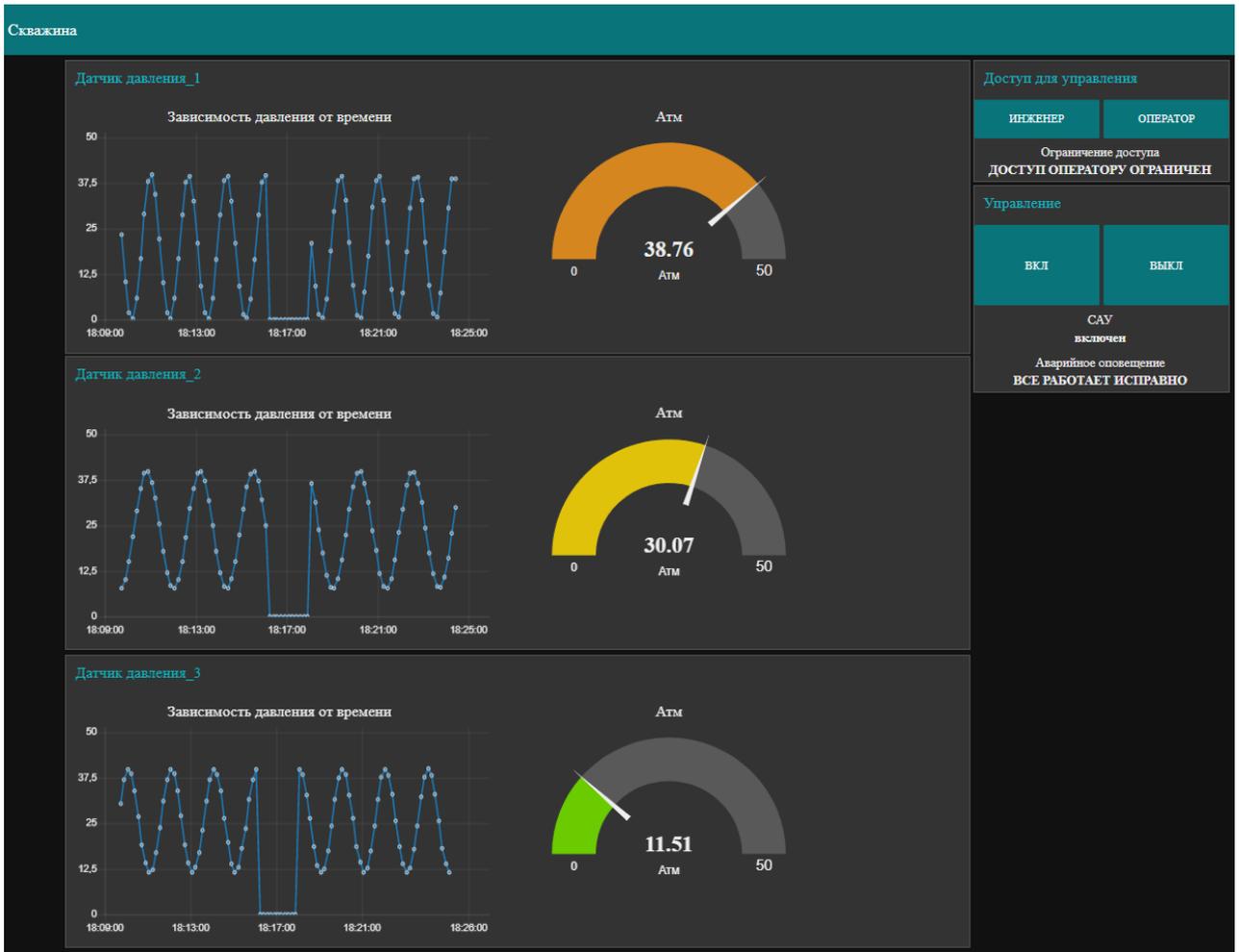


Рис.45. Экранная форма на Node-RED

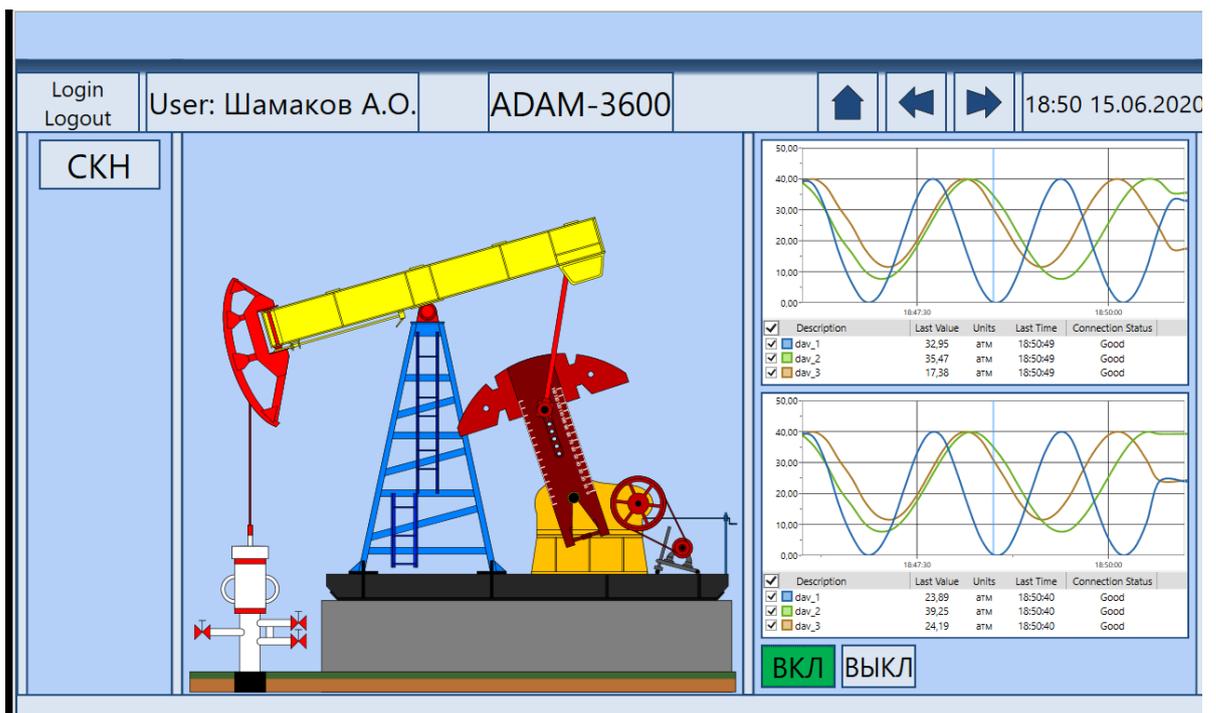


Рис.46. АРМ оператора на базе GENESIS64

Заключение

В результате проделанной работы были внедрены технологии промышленного интернета вещей (IIoT) для построения систем телемеханики и телеметрии распределенных промышленных объектов на примере технологии добычи нефти. Для разработки ПО контроллера скважины СКН для ADAM-3600, был освоен язык программирования Structured Text (ST). Была создана система автоматического управления скважины, которая отвечает за состояние скважины (вкл, выкл), аварийное отключение, управление (пуск/стоп), защиту по давлению. Все данные с контроллера скважины передаются в облачный сервис CloudMQTT по протоколу mqtt. Разработанное клиентское приложение на Node-RED, позволяет запросить данные давления и вывести их на экран в реальном времени. Так же приложение предназначено для дистанционного управления и отслеживания состояния скважины в реальном времени. Для комплексной отладки системы использовался APM оператора на базе GENESIS64. Исходя из поставленных требований, цель полностью достигнута.

Список литературы

1. Станок-качалка.
[Электронный ресурс] – Режим доступа:
<https://dic.academic.ru/dic.nsf/ruwiki/686885>
2. Принцип работы станка качалки.
[Электронный ресурс] – Режим доступа:
<https://vi-pole.ru/princip-raboty-stanka-kachalki.html>
3. Назначение, конструкция и технические характеристики станка-качалки.
[Электронный ресурс] – Режим доступа:
<http://stanokgid.ru/specializirovannyj/stanki-kachalki.html>
4. Устройство и принцип действия АГЗУ «Спутник».
[Электронный ресурс] – Режим доступа:
<https://rengm.ru/rengm/ustroystvo-i-princip-deystviya-agzu-sputnik.html>
5. Многоходовой переключатель скважин.
[Электронный ресурс] – Режим доступа:
<https://samaraburenje.ru/raznoe/mnogohodovoj-pereklyuchatel-skvazhin.html>
6. Интернет вещей Internet of Things.
[Электронный ресурс] – Режим доступа:
[http://www.tadviser.ru/index.php/Интернет_вещей_Internet_of_Things_\(IoT\)](http://www.tadviser.ru/index.php/Интернет_вещей_Internet_of_Things_(IoT))
7. Медведев А. Облачные технологии: тенденции развития, примеры исполнения // Современные технологии автоматизации, 2013. -№ 2-С. 6-9.
8. Облачные вычисления и сервисы: классификация, основные функции, преимущества и недостатки.

- [Электронный ресурс] – Режим доступа:
https://lvee.org/ru/reports/LVEE_2011_03
9. Облачные вычисления (Cloud computing).
[Электронный ресурс] – Режим доступа:
[http://www.tadviser.ru/index.php/Статья:Облочные_вычисления_\(Cloud_computing\)](http://www.tadviser.ru/index.php/Статья:Облочные_вычисления_(Cloud_computing))
10. ADAM-3600 — многофункциональный промышленный контроллер.
[Электронный ресурс] – Режим доступа:
<https://habr.com/ru/company/advantech/blog/446210/>
11. Деркач А.И., Маценко В.А. Контроллер ADAM-3600 в системах телеметрии распределенных промышленных объектов. /Казанский филиал ПРОСОФТ, С. 36.
12. Руководство по программированию NODE-RED.
[Электронный ресурс] – Режим доступа:
<http://noderedguide.com/>
13. MQTT и Modbus: сравнение протоколов, используемых в шлюзах для IoT.
[Электронный ресурс] – Режим доступа:
<https://habr.com/company/intel/blog/304228/>
14. Современная и мощная среда программирования MULTIPROG PRO.
[Электронный ресурс] – Режим доступа:
<https://rusautomation.ru/multiprog-omc-8000>
15. Интеллектуальный терминальный модуль ADAM-3600 с технологией TagLink от Advantech.
[Электронный ресурс] – Режим доступа:
https://nnz-ipc.ru/news/intellektualnyj_terminalnyj_modul_adam3600_s_tehnologiej_taglink_ot_advantech/

Приложение А

Основная программа (skn), позволяющая управлять и вызывать функциональные блоки:

(*переменная CAU отвечает за ограничения доступа при дистанционном управлении (инженер или оператор) *)

```
CAU := I_Net_Control_ON_OFF * I_Priority + I_Channel1_Device1_OnOff *
NOT(I_Priority);
```

(*вызывание функционального блока well*)

```
well_2(Control_ON:= CAU );
```

(*состояние скважины (вкл, выкл)*)

```
O_Net_Control_State := well_2.Control_State;
```

(*режим дистанционного управления (инженер, оператор)*)

```
O_access_control := I_Priority;
```

(*уведомление об аварии на станции при превышении давления на скважине*)

```
O_Net_Glitch := well_2.Glitch;
```

(*вызывание функционального блока SAU *)

```
SAU_1();
```

(*три выходных сигнала деления (АТМ) с функционального блока SAU *)

```
O_dav_1:=SAU_1.pol_1;
```

```
O_dav_2:=SAU_1.pol_2;
```

```
O_dav_3:=SAU_1.pol_3;
```

Функциональный блок SAU, предназначен для преобразования полученных сигналов в сигналы давления:

(*входные сигналы АЦП*)

```
x := I_BoardIO_AI_0;
```

```
y := I_BoardIO_AI_1;
```

```
z := I_BoardIO_AI_2;
```

(*полученные сигналы преобразуем в сигналы давления (АТМ)*)

pol_1:= 1.9 * x + 0.0025 * EXPТ(x,2)+ 0.000125* EXPТ(x,3) ;

pol_2:= 1.9 * y + 0.0025 * EXPТ(y,2)+ 0.000125 * EXPТ(y,3) ;

pol_3:= 1.9 * z + 0.0025 * EXPТ(z,2)+ 0.000125 * EXPТ(z,3) ;

Функциональный блок well. Программа выполняет функцию передачи данных на логический котроллер, включение и выключение скважины, так как модуль имеет ограничение в передаче тока (0-20 mA), это было предусмотрено:

(*реальное время контроллера*)

taim := DWORD_TO_REAL(I_SYS_UPTIME);

(*taim_1 := taim_1 + 0.1;*)

(*ЦАП*)

IF Control_ON = TRUE

THEN

(*включение скважины*)

(*светодиод горит красным при вкл-ой скважине *)

O_BoardIO_DO_0 := TRUE ;

O_BoardIO_DO_2 := FALSE ;

(*состояние скважины вкл*)

Control_State:= TRUE;

(*допустимый промежуток сигнала тока*)

IF (4.0 <= Vihod_A AND Vihod_A <= 20.0)

THEN

(*сигнал тока меняется по функции синуса с периодом 30с*)

Vihod_A := 13.0 + 7.0 * sin((2.0*3.14*taim)/30.0);

(*состояние скважины вкл*)

Control_State:= TRUE;

(*скважина работает в штатном режиме*)

```
Glitch := TRUE;

ELSE

(*уведомляет об аварии при превышении допустимого значения*)

Glitch := FALSE;

END_IF;

(*допустимый промежуток сигнала тока*)

IF (4.0 <= Vihod_A1 AND Vihod_A1 <= 20.0)

THEN

(*сигнал тока меняется по функции синуса с периодом 60с*)

Vihod_A1 := 12.5 + 7.5 * sin((2.0*3.14*taim)/60.0);

(*состояние скважины вкл*)

Control_State:= TRUE;

(*скважина работает в штатном режиме*)

Glitch := TRUE;

ELSE

(*уведомляет об аварии при превышении допустимого значения*)

Glitch := FALSE;

END_IF;

(*допустимый промежуток сигнала тока*)

IF (4.0 <= Vihod_A2 AND Vihod_A2 <= 20.0)

THEN

(*сигнал тока меняется по функции синуса с периодом 120с*)

Vihod_A2 := 12.0 + 8.0 * sin((2.0*3.14*taim)/120.0);

(*состояние скважины вкл*)

Control_State:= TRUE;

(*скважина работает в штатном режиме*)

Glitch := TRUE;

ELSE
```

(*уведомляет об аварии при превышении допустимого значения*)

Glitch := FALSE;

END_IF;

ELSE

(*если не выполняются условия представлены выше, то скважина отключается *)

(*светодиод горит синим при вык-ой скважине *)

O_BoardIO_DO_0 := FALSE ;

O_BoardIO_DO_2 := TRUE ;

Control_OFF := TRUE;

(*состояние скважины откл*)

Control_State:= FALSE;

Vihod_A := 0.0;

Vihod_A1 := 0.0;

Vihod_A2 := 0.0;

END_IF;

(*выходные сигналы с модуля *)

O_Slot4_V1AO_0 := Vihod_A;

O_Slot4_V2AO_1 := Vihod_A1;

O_Slot4_V3AO_2 := Vihod_A2;

Приложение В

Код клиентской части на ПО Node-RED:

```
[{"id":"d3357237.2187","type":"ui_button","z":"5a138742.9cc848","name":"","group":"266d2546.f5ac6a","order":1,"width":3,"height":2,"passthru":false,"label":"Вкл","tooltip":"","color":"","bgcolor":"","icon":"","payload":{"w":[{"tag":"Net_Control_ON_OFF","value":1}]},"payloadType":"str","topic":"","x":210,"y":500,"wires":["76c58d96.9cf474"]}, {"id":"323d4edd.2eb382","type":"ui_button","z":"5a138742.9cc848","name":"","group":"266d2546.f5ac6a","order":2,"width":3,"height":2,"passthru":false,"label":"Выкл","tooltip":"","color":"","bgcolor":"","icon":"","payload":{"w":[{"tag":"Net_Control_ON_OFF","value":0}]},"payloadType":"str","topic":"","x":210,"y":540,"wires":["76c58d96.9cf474"]}, {"id":"76c58d96.9cf474","type":"mqtt out","z":"5a138742.9cc848","name":"","topic":"cmd/device_C","qos":"","retain":"","broker":"89066621.c61978","x":420,"y":520,"wires":[]}, {"id":"3a59379d.3e67c8","type":"mqtt in","z":"5a138742.9cc848","name":"","topic":"data/device_B","qos":"2","datatype":"json","broker":"c58e06cf.04d878","x":150,"y":140,"wires":["cceb3b5.0af5d","f5b29960.cfc798","7cee0f60.f143c"]}, {"id":"cceb3b5.0af5d","type":"function","z":"5a138742.9cc848","name":"обработка полученных данных","func":"\n msg.payload=msg.payload.d[0].value;\nreturn msg;\n","outputs":1,"noerr":0,"x":450,"y":60,"wires":["1d2802c0.4d7e0d","8724ff5b.f670e"]}, {"id":"891eb503.9c0e68","type":"ui_chart","z":"5a138742.9cc848","name":"Давление 2","group":"3516689.24ba898","order":1,"width":10,"height":6,"label":"Зависимость давления от времени","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":true,"ymin":"0","ymax":"50","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#30ee20","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":690,"y":160,"wires":[]}, {"id":"f5b29960.cfc798","type":"function","z":"5a138742.9cc848","name":"обработка полученных данных","func":"\n msg.payload=msg.payload.d[1].value;\nreturn msg;\n","outputs":1,"noerr":0,"x":450,"y":140,"wires":["891eb503.9c0e68","5ee7a.48339186c"]}, {"id":"7cee0f60.f143c","type":"function","z":"5a138742.9cc848","name":"обработка полученных данных","func":"\n msg.payload=msg.payload.d[2].value;\nreturn msg;\n","outputs":1,"noerr":0,"x":450,"y":220,"wires":["6990a763.fcf218","bd0dbe78.49557"]}, {"id":"1d2802c0.4d7e0d","type":"ui_chart","z":"5a138742.9cc848","name":"Давление 1","group":"ce77be5.4d7964","order":1,"width":10,"height":6,"label":"Зависимость давления от времени","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":true,"ymin":"0","ymax":"50","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":690,"y":80,"wires":[]}, {"id":"90d00bfb.e56008","type":"function","z":"5a138742.9cc848","name":"Cooler State","func":"if (msg.payload.d[0].value) \n{ msg.payload = \" включен\" }\nelse\n{ msg.payload = \" выключен\" }\n\nreturn"}]
```

```

msg;,"outputs":1,"noerr":0,"x":370,"y":320,"wires":[["a1b793f1.8c39c"]],{"id":"a1b793f1.8c39c","type":"ui_text","z":"5a138742.9cc848","group":"266d2546.f5ac6a","order":3,"width":6,"height":1,"name":"","label":"CAУ","format":"{{msg.payload}}","layout":"col-center","x":510,"y":320,"wires":[]},{"id":"cd4013e1.6b388","type":"mqtt in","z":"5a138742.9cc848","name":"","topic":"data/device_A","qos":2,"datatype":"json","broker":"c58e06cf.04d878","x":170,"y":320,"wires":[["90d00bfb.e56008"]],{"id":"69b7b056.d51a2","type":"ui_button","z":"5a138742.9cc848","name":"","group":"b565e5ea.6d7a58","order":1,"width":3,"height":1,"passthru":false,"label":"Инженер","tooltip":"","color":"","bgcolor":"","icon":"","payload":"{ \"w\":[ { \"tag\":\"Priority\", \"value\":1}]}","payloadType":"str","topic":"","x":220,"y":600,"wires":[["2f180ee1.9ff672"]],{"id":"204ba86f.8d6e18","type":"ui_button","z":"5a138742.9cc848","name":"","group":"b565e5ea.6d7a58","order":2,"width":3,"height":1,"passthru":false,"label":"Оператор","tooltip":"","color":"","bgcolor":"","icon":"","payload":"{ \"w\":[ { \"tag\":\"Priority\", \"value\":0}]}","payloadType":"str","topic":"","x":230,"y":640,"wires":[["2f180ee1.9ff672"]],{"id":"2f180ee1.9ff672","type":"mqtt out","z":"5a138742.9cc848","name":"","topic":"cmd/device_C","qos":"","retain":"","broker":"89066621.c61978","x":420,"y":620,"wires":[]},{"id":"6990a763.fc218","type":"ui_chart","z":"5a138742.9cc848","name":"Даление_3","group":"2eb8a6f9.80730a","order":1,"width":10,"height":6,"label":"ЗАВИСИМОСТЬ ДАВЛЕНИЯ ОТ времени","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":true,"ymin":0,"ymax":50,"removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":690,"y":240,"wires":[]},{"id":"9a742a37.12daa8","type":"function","z":"5a138742.9cc848","name":"Cooler State","func":"msg.payload = msg.payload.d.find(item => item.tag == \"access_control\")\nif (msg.payload.value == 1)\n{\nmsg.payload = \"Доступ оператору ограничен\"\n}\nelse\n{\nmsg.payload = \" Инженер выкл. \"\n}\nreturn msg;","outputs":1,"noerr":0,"x":370,"y":380,"wires":[["9c6e905e.bd7fb"]],{"id":"9c6e905e.bd7fb","type":"ui_text","z":"5a138742.9cc848","group":"b565e5ea.6d7a58","order":3,"width":6,"height":1,"name":"","label":"Ограничение доступа","format":"{{msg.payload | uppercase}}","layout":"col-center","x":560,"y":380,"wires":[]},{"id":"afc623c5.4d379","type":"mqtt in","z":"5a138742.9cc848","name":"","topic":"data/device_A","qos":2,"datatype":"json","broker":"c58e06cf.04d878","x":170,"y":380,"wires":[["9a742a37.12daa8"]],{"id":"9207027a.02fda","type":"function","z":"5a138742.9cc848","name":"Cooler State","func":"msg.payload = msg.payload.d.find(item => item.tag == \"Net_Glitch\")\nif (msg.payload.value == 1)\n{\nmsg.payload = \"Все работает исправно\"\n}\nelse\n{\nmsg.payload = \" АВАРИЙНОЕ ОТКЛЮЧЕНИЕ \"\n}\nreturn msg;","outputs":1,"noerr":0,"x":370,"y":440,"wires":[["aeb186a2.dbbf48"]],{"id":"aeb186a2.dbbf48","type":"ui_text","z":"5a138742.9cc848","group":"266d2546.f5ac6a","order":4,"width":6,"height":1,"name":"","label":"Аварийное оповещение","format":"{{msg.payload | uppercase}}","layout":"col-center","x":570,"y":440,"wires":[]},{"id":"cf918a06.fa0f68","type":"mqtt in","z":"5a138742.9cc848","name":"","topic":"data/device_A","qos":2,"datatype":"json","broker":"c58e06cf.04d878","x":170,"y":440,"wires":[["9207027a.02fda"]],{"id":"bd0dbe78.49557",

```

```

"type":"ui_gauge","z":"5a138742.9cc848","name":"","group":"2eb8a6f9.80730a","order":2,"width":8,"height":5,"gtype":"gage","title":"АТМ","label":"АТМ","format":"{{value}}","min":0,"max":50,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":670,"y":200,"wires":[]},{ "id":"8724ff5b.f670e","type":"ui_gauge","z":"5a138742.9cc848","name":"","group":"ce77be5.4d7964","order":2,"width":8,"height":5,"gtype":"gage","title":"АТМ","label":"АТМ","format":"{{value}}","min":0,"max":50,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":670,"y":40,"wires":[]},{ "id":"5ee7a.48339186c","type":"ui_gauge","z":"5a138742.9cc848","name":"","group":"3516689.24ba898","order":1,"width":8,"height":5,"gtype":"gage","title":"АТМ","label":"АТМ","format":"{{value}}","min":0,"max":50,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":670,"y":120,"wires":[]},{ "id":"266d2546.f5ac6a","type":"ui_group","z":"","name":"Управление","tab":"64999325.2a5b9c","order":5,"disp":true,"width":6,"collapse":false},{ "id":"89066621.c61978","type":"mqtt-broker","z":"","name":"Mosca","broker":"m23.cloudmqtt.com","port":16065,"clientId":"","useTls":false,"compatmode":true,"keepalive":60,"cleansession":true,"birthTopic":"","birthQos":0,"birthPayload":"","closeTopic":"","closePayload":"","willTopic":"","willQos":0,"willPayload":"","id":"c58e06cf.04d878","type":"mqtt-broker","z":"","name":"Mosca","broker":"m23.cloudmqtt.com","port":16065,"clientId":"","useTls":false,"compatmode":true,"keepalive":10,"cleansession":true,"birthTopic":"","birthQos":0,"birthPayload":"","closeTopic":"","closePayload":"","willTopic":"","willQos":0,"willPayload":"","id":"3516689.24ba898","type":"ui_group","z":"","name":"Датчик давления_2","tab":"64999325.2a5b9c","order":2,"disp":true,"width":21,"collapse":false},{ "id":"ce77be5.4d7964","type":"ui_group","z":"","name":"Датчик давления_1","tab":"64999325.2a5b9c","order":1,"disp":true,"width":21,"collapse":false},{ "id":"b565e5ea.6d7a58","type":"ui_group","z":"","name":"Доступ для управления","tab":"64999325.2a5b9c","order":4,"disp":true,"width":6,"collapse":false},{ "id":"2eb8a6f9.80730a","type":"ui_group","z":"","name":"Датчик давления_3","tab":"64999325.2a5b9c","order":3,"disp":true,"width":21,"collapse":false},{ "id":"64999325.2a5b9c","type":"ui_tab","z":"","name":"Скважина","icon":"dashboard","disabled":false,"hidden":false}]

```