

Инструкция для пользователя
программного обеспечения
IASolutions QForecast

Содержание

1.	Установка программного обеспечения	3
2.	Краткое описание программного обеспечения	5
3.	Функциональное назначение	6
4.	Обеспечение и поддержание жизненного цикла программного обеспечения.....	7
4.1.	Техническая поддержка.....	7
4.2.	Сообщения об ошибках.....	7
4.3.	Пользователи программного обеспечения	7
4.4.	Скачать обновленную версию	7
5.	Описание реализации функций протокола DME.....	9
6.	Описание реализации функций протокола VMW	12
7.	Реализация протокола взаимодействия с лазерным сканером.	18
8.	Описание действий оператора измерительной системы.....	24
9.	Реализация модуля анализа трендов.	29

1. Установка программного обеспечения

Для установки программного обеспечения требуется на ЭВМ, соответствующей техническому заданию, открыть установочный файл IASolutionsQForecast.exe (прилагается на компакт диске).

Далее нужно выбрать язык (рисунок 1.1).



Рисунок 1.1. Выбор языка.

После выбора языка нужно выбрать директорию установки (рисунок 1.2).

Рисунок 1.2. Выбор директории установки.

Процесс установки будет отображаться в установочном окне (рисунок 1.3.).

Рисунок 1.3. Установочное окно.

После чего программа предложит завершить установку (рисунок 1.4.).

Рисунок 1.4. Завершение установки.

После окончания установки программа предложит выбрать дополнительные задачи (рисунок 1.5.).

2

Рисунок 1.5. Выбор дополнительных задач.

Установка ПО на персональный компьютер со следующей спецификацией:

- Microsoft Windows 10
- Процессор не менее x64 архитектуры, 8 ядер (16 потоков), тактовая частота ядра менее 3,7 ГГц.;

- Оперативная память 32 Гб;
- 1 SSD объемом 940 Гб, хранилище данных HDD 1 ТБ.
- Монитор;
- Клавиатура;
- Мышь.

2. Краткое описание программного обеспечения

Задача программного обеспечения – это автоматическое прогнозирование качества геометрии продукции механообработки (например, турбинные лопатки, шестерни и т.д.) на станках с ЧПУ.

Программное обеспечение встраивается в АСУ предприятия и подключается к системам мониторинга состояния оборудования. Результат – снижение брака на производстве, времени ремонта и простоя оборудования, что приведет к повышению рентабельности производства.

3. Функциональное назначение

Функциональное назначение продукта:

- 1) Сбор и хранение данных от производящего оборудования в виде последовательностей цифровых значений.
- 2) Сбор и хранение данных с систем контроля качества геометрии в виде отчетов с цифровыми значениями.
- 3) Выявление по собираемым данным характеристик работы оборудования, свидетельствующих о возможном появлении брака, износе и разбалансировке.
- 4) Сигнализация о необходимости переналадки оборудования.

4. Обеспечение и поддержание жизненного цикла программного обеспечения

4.1. Техническая поддержка

Техническая поддержка в отношении использования программного обеспечения осуществляется поставщиком в течение 12 (двенадцать) месяцев, с момента передачи права использования. Под технической поддержкой понимается предоставляемая по выделенной линии службы приема и разрешения технических запросов (телефон, e-mail указаны на сайте <https://iasolutions.ru/>) специалистами поставщика, консультационная помощь, включающая в себя: предоставление информации о новых версиях и исправлениях программного обеспечения, предоставление информации о базовых функциях продукта, консультации по проблемам с первичной инсталляцией и активацией программного обеспечения.

4.2. Сообщения об ошибках

В случае появления сообщения об ошибках требуется направить на телефон или e-mail (указаны на сайте <https://iasolutions.ru/>) скриншот сообщения об ошибках с описанием действий, приведших к ошибке. Также необходимо отправить файл журнала ошибок, который находится в папке logs, в которую установлено программное обеспечение.

4.3. Пользователи программного обеспечения

Пользователь программного обеспечения должен быть ознакомлен с инструкцией пользователя программного обеспечения.

4.4. Скачать обновленную версию

Уведомления о появлении новых версий программного обеспечения поступают на адрес электронной почты покупателя, указанный в договоре на приобретение лицензии. Для того, чтобы скачать обновленную версию программного обеспечения необходимо пройти по ссылке в сообщении. Также при необходимости можно обратиться по телефону или e-mail (указаны на сайте <https://iasolutions.ru/>) с указанием названия компании, приобретшей

лицензию на программное обеспечение. По данному запросу будет предоставлена ссылка для скачивания обновленной версии программного обеспечения.

5. Описание реализации функций протокола DME

Каждый объект системы обеспечивает программные возможности для поддержки следующих функций:

- GetProp(..), GetPropE(..)
- SetProperty(..)
- EnumProp(..), EnumAllProp(..)

Базовый набор свойств, общий для всех типов машин и инструментов, определен в спецификации протокола I++ DME Dimensional Measurement Equipment Interface Version 2.0. Дополнительные свойства оборудования могут быть добавлены и могут быть обработаны через указанные выше функции. Это увеличивает гибкость спецификации.

GoTo(..)

Клиент использует этот метод для многоосевого перемещения в целевую позицию с использованием активного инструмента. Сервер будет перемещать машину, так что все перечисленные оси начнут двигаться в одно и то же время. Движение контролируется блоком Tool.GoToPar (скорость, Ускорение), если это возможно. Состояние Готово будет отправлено клиенту, когда последняя ось достигнет цели должност.

Если какая-либо из осей X (), Y (), Z () не включена в качестве аргумента, ее значение пока и после выполнения GoTo () должно быть таким же, как и его значение непосредственно перед выполнением GoTo (). Если возможно, DME должен использовать текущее номинальное значение контроллера для предотвращения дрейфа. Необходимо обратить внимание: комбинированные движения линейных осей и оси вращения не будут синхронизированы (в зависимости от движения в пространстве и окончания движения) в некоторых комбинации (когда скорость оси не может контролироваться).

ClearAllErrors(..)

Клиент вызывает этот метод, чтобы сервер восстанавливался после ошибки.

Home(..)

Клиент использует этот метод для запуска машины. Сервер должен быть в состоянии Homed прежде, чем клиент может вызывать методы, которые перемещают машину. Когда выполняется эта команда, сервер переместит машину в исходное положение. Нулевое положение для каждой машины специфично для машины и зависит от реализации. Исходное положение для каждой машины фиксировано. Может возникнуть любой тип движения в радиальном направлении во время исполнения Home. Единственное требование состоит в том, чтобы конечная позиция была нулевой (исходной).

IsHomed(..)

Клиент использует этот метод для запроса, все ли оси машины находятся в нулевой позиции.

ReadAllTemperatures(..)

Клиент использует этот метод для считывания всех температурных датчиков, подключенных к DME. Эта команда предназначена для использования только для целей ведения журнала. Температура предоставляется в градусах Цельсия. Каждый датчик идентифицируется именем, которое является специфическим для аппаратного обеспечения DME и / или специфическим для реализации сервера.

GetErrStatusE(..)

Клиент использует этот метод для запроса состояния ошибки на сервере.

SaveActiveCoordSystem(..)

Клиент использует этот метод для сохранения активной системы координат детали.

SetCoordSystem(..)

Клиент использует этот метод для выбора системы координат, с которой он хочет работать.

GetCsyTransformation(..)

Клиент использует этот метод, чтобы получить преобразование координат с сервера.

CollisionVolume(..)

Клиент использует этот метод для запроса ограничений объема от коллизий выбранного инструмента. Возвращенные элементы содержат все возможные части, вызывающие коллизии. Положение этих элементов относится к выбранному инструменту и ориентации на текущую выбранную координату системы. С помощью оптических датчиков — это определенная точка диапазона измерения.

Tool(..)

Клиент использует этот метод для выбора указателя на фактический активный инструмент. Он также может использоваться как указатель на NoTool.

FindTool(..)

Клиент использует этот метод для получения указателя на инструмент с известным именем.

FoundTool(..)

Этот метод действует как указатель на инструмент с известным именем, выбранным FindTool. FoundTool () действителен только после вызова FindTool (), иначе в ответ будет получено «UnDefTool». Клиент использует этот метод для смены измерительного инструмента инструмента с помощью ProbeChanger или вручную.

ChangeTool(..)

Если во время выполнения ChangeTool () возникает ошибка, то нужно запросить сервер, какой инструмент активен.

OnMoveReportE(..)

Клиент использует этот метод для определения той информации, которую сервер должен отправить клиенту, пока машина движется. Команда определяет формат (последовательность) и содержимое.

Обратите внимание, должны выполняться все 3 условия:

- Текущий инструмент выровнен.
- Указанный целевой инструмент выровнен.

- Ток и указанный инструмент являются частью одного и того же узла механического инструмента.

`AlignTool(..)`

Клиент использует этот метод, чтобы ориентировать инструмент в соответствии с заданным вектором (-ами). Эта команда применима только для выравниваемых инструментов.

`EnumTools(..)`

Клиент использует этот метод для запроса имен доступных инструментов (вручную или автоматически). Данный метод возвращает список имен.

`SetTool(..)`

Клиент использует этот метод, чтобы установить данный инструмент активным.

`EnumToolCollection(..)`

Клиент использует этот метод для запроса `ToolCollection` системы. Данный метод возвращает имена и тип дочерних инструментов или коллекций.

`EnumAllToolCollections(..)`

Клиент использует этот метод для запроса инструментов системы. Он возвращает имена инструментов и имена наборов инструментов.

`GetChangeToolAction(..)`

Клиент использует этот метод для запроса текущего набора инструментов, чтобы перейти к определенному инструменту. Этот метод можно использовать для получения информации о том, какое действие необходимо выполнить и какое движение будет выполнено для перехода к данному инструменту.

6. Описание реализации функций протокола VMW

Для реализации модуля взаимодействия со станками нужно использовать следующие функции на базе команд, описанных в приложении к OPC серверу VMW.

CncMessage(..)

CncChannelType может генерировать события типа cncMessageType для предоставления информационных сообщений канала (например, запускаемых из программы обработки деталей с ЧПУ).

GeneralModelChangeEvent (..)

CncChannelType может генерировать события типа GeneralModelChangeEvent, определенные в МЭК 62541-3: 2015, чтобы информировать клиентов, когда объекты CncSpindle или CncAxis были добавлены или удалены из экземпляра CncChannelType.

NodeVersion(..)

Необязательное свойство NodeVersion должно присутствовать, если сервер генерирует GeneralModelChangeEvents для объекта типа CncChannelType. Свойство NodeVersion и отношение к GeneralModelChangeEvents определены в МЭК 62541-3: 2015.

CncAxis(..)

Объект оси ЧПУ.

CncSpindle(..)

Объект шпинделя ЧПУ.

ActFeedrate(..)

Фактическая скорость подачи.

ActJogIncrement (..)

Активное перемещение JOG, то есть изменение позиции, которое выполняется с каждой пользовательской командой JOG. Обычно есть заранее заданные приращения (1/10, 1/100, приращение положения 1/1000 в выбранной физической единице), но возможен также режим непрерывного перемещения и режим с переменным шагом. Режим непрерывного перемещения указывается со значением приращения -1, все остальные приращения представлены в соответствующих единицах.

ActGFunctions (..)

Массив активных G-функций; может содержать несколько активных G-функций одновременно (модальные и немодальные G-функции).

ActMainProgramFile (..)

Файл активной основной программы ЧПУ.

ActMainProgramFileOffset (..)

Текущее положение программы ЧПУ (соответствует строке программы).

ActMainProgramName (..)

Название основной программы ЧПУ.

ActMainProgramLine(..)

Номер строки основной программы ЧПУ (обычно определяется как N <Number>).

ActMFunctions(..)

Массив активных M-функций: одновременно могут быть активными несколько M-функций (модальные и немодальные M-функции).

ActModalOffsetFunction(..)

Функция смещения нуля (обычно G54, G55, G56 или G57; G53 отменяет смещения нуля).

ActOperationMode(..)

Активный режим работы канала.

ActProgramBlock(..)

Блок строк, содержащий предыдущие, фактические и последующие строки программы обработки с ЧПУ.

ActProgramFile(..)

Путь к активному файлу программы обработки детали ЧПУ (основной или подпрограммы).

ActProgramFileOffset(..)

Относительный путь к активному файлу программы обработки ЧПУ (основной или подпрограммы).

ActProgramLine(..)

Номер строки активной программы обработки с ЧПУ (основная или подпрограмма).

ActProgramName(..)

Имя активной программы обработки детали с ЧПУ (основная или подпрограмма).

ActProgramStatus(..)

Статус программы активного канала.

BlockMode(..)

Активный статус канала.

Состояние режима блокировки (true, если режим блокировки активен, иначе false). При активном режиме блокировки отдельные программные блоки обрабатываются один за другим. Каждый шаг должен быть запущен оператором.

CmdFeedrate(..)

Задание значения скорости подачи.

CmdOverride(..)

Переопределить значение уставки.

DryRunFeed(..)

Проверка значения скорости.

FeedHold (..)

Feed status (true in case of feed hold active, else false).

Id(..)

Уникальный числовой идентификатор канала.

PosTspVcsA(..)

Фактическое положение центральной точки инструмента (А координата вектора) в декартовой базовой системе координат станка. Центральная точка инструмента — это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspVcsB(..)

Фактическое положение центральной точки инструмента (В координата вектора) в декартовой базовой системе координат станка. Центральная точка инструмента — это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspVcsC(..)

Фактическое положение центральной точки инструмента (С координата вектора) в декартовой базовой системе координат станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspVcsX(..)

Фактическое положение центральной точки инструмента (Х координата) в декартовой базовой системе координат станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspVcsY(..)

Фактическое положение центральной точки инструмента (Y координата) в декартовой базовой системе координат станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspVcsZ(..)

Фактическое положение центральной точки инструмента (Z координата) в декартовой базовой системе координат станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается

функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента.

PosTspWcsA(..)

Фактическое положение центральной точки инструмента (А координата вектора) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

PosTspWcsB(..)

Фактическое положение центральной точки инструмента (В координата вектора) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

PosTspWcsC(..)

Фактическое положение центральной точки инструмента (С координата вектора) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

PosTspWcsX(..)

Фактическое положение центральной точки инструмента (Х координата) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она

рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

PosTcpWcsY(..)

Фактическое положение центральной точки инструмента (Y координата) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

PosTcpWcsZ(..)

Фактическое положение центральной точки инструмента (Z координата) в декартовой системе координат заготовки станка. Центральная точка инструмента - это точка отсчета на инструменте, как она рассматривается функцией коррекции инструмента ЧПУ. Если инструмента нет, центральная точка инструмента является нулевой точкой держателя инструмента. Следовательно, позиция соответствует позиции, запрограммированной в программе обработки детали с ЧПУ.

ToolId(..)

Идентификатор активного инструмента, который был выбран (например, вызов «T1» в программе обработки с ЧПУ). Возвращает пустую строку, если инструмент отсутствует.

7. Реализация протокола взаимодействия с лазерным сканером.

Основные параметры, настраиваемые в сканере:

Автоэкспозиция и время накопления. Интенсивность излучения, падающего на CMOS-матрицу, зависит от качества поверхности контролируемого объекта и расстояния до него. Поэтому время накопления

излучения матрицы автоматически подстраивается (изменяется) с целью достижения максимальной точности измерения (режим автоэкспозиции). Возможен также режим работы с постоянным временем накопления, заданным вручную. Данный режим предпочтительно использовать при контроле объектов, содержащих участки, резко отличающиеся по отражающей способности.

Уровень выходной мощности лазера. Изменение этого параметра позволяет настроить сканер на работу с минимальным предельным временем накопления (максимальное быстродействие) для конкретных поверхностей. Рабочее окно. Частота обновления результата во многом определяется диапазоном контроля, причем в большей степени по координате Z. Данный параметр позволяет выбрать размер рабочего окна в пределах всего диапазона как по X, так и по Z координате, тем самым позволяет настроить сканер на максимальное быстродействие.

Угол поворота системы координат. По умолчанию датчик передает координаты точек поверхности контролируемого объекта в собственной, базовой системе координат. Возможна также передача координат в любой произвольной системе, повернутой по отношению к базовой на некоторый угол, определяемый данным параметром.

Коэффициент деления — определяет коэффициент деления для входа внешней синхронизации.

Синхронизация измерения, результата и аналогового выхода. Параметры, определяющие режимы синхронизации сканера.

Задержка включения лазера. Данный параметр действителен только для сканеров с опцией ML. В таких сканерах лазер включается только на время экспозиции. Параметр задает время задержки включения лазера после возникновения сигнала внешней синхронизации. Задержка используется для устранения взаимного влияния сканеров друг на друга и позволяет последовательно включать сканеры, подключенные к одной линии внешней синхронизации.

Количество точек по координате X. Данный параметр определяет разрешающую способность сканера по координате X. Для сканеров с Ethernet-интерфейсом – всегда 1024 точки.

Время удержания точки. Данный параметр определяет время, в течение которого значения координат точки сохраняются на аналоговых выходах. Минимально допустимое значение для выхода 4...20 мА – 20 мкс и для выхода 0...10В - 1 мкс.

Режим работы. Сканер может работать в двух режимах – передача изображения и передача профиля. Режим работы определяется соответствующей командой запроса данных.

Для работы с этими параметрами была реализована библиотека с функциями для протоколов UDP и TCP/IP [4].

Функция RF_Search производит поиск в сети и в случае обнаружения устройств возвращает указатель на цепочку структур типа UDP_DevInfoBlock_PC.

UDP_DevInfoBlock_PC* RF_Search(USHORT uType, USHORT uTime)

uType - 0xFFFF – все устройства; 620 – RF620.

uTime - время поиска в миллисекундах

```
typedef struct UDP_DEVINFOBLOCK_PC {
```

```
unsigned short usDeviceType;
```

```
unsigned char ucIP[4];
```

```
unsigned char ucMAC[6];
```

```
unsigned char ucInfo[256];
```

```
UDP_DEVINFOBLOCK_PC *pNext;
```

```
} UDP_DevInfoBlock_PC, *lpUDP_DevInfoBlock_PC;
```

где usDeviceType – двухбайтная величина (байты [1-0]), содержащая тип устройства (для RF620 данная величина равна 620) (тип WORD);

ucIP[4] – 4 байтная структура (байты [5-2]), содержащая IP-адрес устройства (тип BYTE);

ucMAC[6] – 6 байтная структура (байты [11-6]), содержащая MAC-адрес устройства (тип BYTE);

ucInfo[256] – содержит информационные данные по устройству (байты [267- 12]) (тип BYTE);

pNext – указатель на следующую структуру UDP_DevInfoBlock_PC.

2) Функция RF_ClearList производит очистку и удаление цепочки структур типа UDP_DevInfoBlock_PC.

```
void RF_ClearList(UDP_DevInfoBlock_PC * lpDevBlock)
```

lpDevBlock – указатель на блок, описывающий выбранное устройство

Функция RF620_Connect создаёт подключение к устройству, описанному в структуре UDP_DevInfoBlock_PC из цепочки структур. В случае успеха возвращает SOCKET на устройство.

```
SOCKET RF620_Connect(UDP_DevInfoBlock_PC * lpDevBlock)
```

lpDevBlock – указатель на блок, описывающий выбранное устройство

Функция RF620_Disconnect закрывает SOCKET на устройство.

```
BOOL RF620_Disconnect(SOCKET sDevice)
```

Функция RF620_GetReult возвращает буфер, заполненный координатами точек.

```
BOOL RF620_GetReult(SOCKET sDevice, void * lpBuffer)
```

sDevice – сокет конкретного устройства

lpBuffer – указатель на данные, полученные от устройства (4100 байт).

Формат буфера:

байты [1 - 0] содержат циклический счётчик номера измерения;

байты [3 - 2] содержат циклический счётчик номера пакета;

байты [5 - 4] содержат координату Z 0-ой точки профиля;

байты [7 - 6] содержат координату X 0-ой точки профиля;

байты [9 - 8] содержат координату Z 1-ой точки;

байты [11 - 10] содержат координату X 1-ой точки;

.....

байты [4093 - 4092] содержат координату Z 1022-ой точки;

байты [4095 - 4094] содержат координату X 1022-ой точки;

байты [4097 - 4096] содержат координату Z 1023-ой точки;

байты [4099 - 4098] содержат координату X 1023-ой точки;

Счётчики - величины типа WORD.

Координата - величина типа SHORT.

Функция RF620_GetImage возвращает буфер с изображением шириной 640 и высотой 512 пикселей с градациями от 0 до 255.

BOOL RF620_GetImage(SOCKET sDevice, void * lpBuffer)

sDevice – сокет конкретного устройства

lpBuffer – указатель на данные, полученные от устройства (640*512 байт)

Функция RF620_ReadParams возвращает буфер с конфигурационным пространством

размером 512 байт.

BOOL RF620_ReadParams(SOCKET sDevice, void * lpBuffer)

sDevice – сокет конкретного устройства

lpBuffer – указатель на таблицу с параметрами (512 байт)

Функция RF620_WriteParams передаёт буфер с конфигурационным пространством

размером 512 байт.

BOOL RF620_WriteParams(SOCKET sDevice, void * lpBuffer)

sDevice – сокет конкретного устройства

lpBuffer – указатель на таблицу с параметрами (512 байт)

Функция RF620_FlushParams сохраняет текущее или восстанавливает заводское

конфигурационное пространство в энергонезависимой памяти сканера.

BOOL RF620_FlushParams(SOCKET sDevice, BOOL bDefault)

sDevice – сокет конкретного устройства

bDefault – восстановить заводские параметры если TRUE

Функция RF620_CalcAngle возвращает рассчитанное значение угла поворота.

BOOL RF620_CalcAngle(SOCKET sDevice, void * iAngle)

sDevice – сокет конкретного устройства

iAngle – указатель на переменную типа SHORT(2 байта).

Возвращает значение от -32767 до 32767, что соответствует углу [-60:60] градусов. Возвращаемое значение -32768 указывает на выход за диапазон определения угла поворота системы координат.

Все функции, имеющие тип BOOL, при успешном завершении возвращают TRUE иначе FALSE.

Массив ucInfo:

ucInfo[0]:

бит10: 00-Зарезервировано;

01-Зарезервировано;

10-Внутренняя синхронизация результата;

11-Возможна как внутренняя, так и внешняя синхронизация результата;

бит2: 0-внутренняя синхронизация измерения;

1-возможна как внутренняя, так и внешняя синхронизация измерения;

бит43: 00-датчик не имеет аналоговых выходов;

01- аналоговые выходы по напряжению;

10- аналоговые выходы по току;

бит5: 0-без модуляции лазерного излучения;

1-с модуляцией лазерного излучения.

ucInfo[3-1]:

Серийный номер устройства.

ucInfo[5-4]:

Базовое расстояние в мм.

ucInfo[7-6]:

Диапазон измерения по Z в мм.

ucInfo[9-8]:

Диапазон измерения по X(SMR) в мм.

ucInfo[11-10]:

Диапазон измерения по X(EMR) в мм.

ucInfo[13-12]:

Параметр приведения координат в мм.

Координаты, полученные от датчика, преобразуются по формулам:

$Z(\text{mm}) = (\text{Диапазон по } Z) * (Z) / (\text{Параметр приведения}); X(\text{mm}) = (\text{Диапазон по } X(\text{EMR})) * (X) / (\text{Параметр приведения});$

ucInfo[15-14]:

Недействительное значение. Если координата, полученная от датчика, равна

данному значению, то данные считаются недействительными.

ucInfo[19-16]:

Версия микропрограммы.

ucInfo[255-20]:

Зарезервированы.

Алгоритмы сбора данных и управления оборудованием отображены в блок-схемах 1-2 в приложении №1, а также в приложении №2 в п. 1-2 в виде исходного кода.

8. Описание действий оператора измерительной системы

Для авторизации оператора измерительной системы реализован интерфейс ввода логина и пароля (рисунок 2.1). Реализована поддержка базы данных операторов.

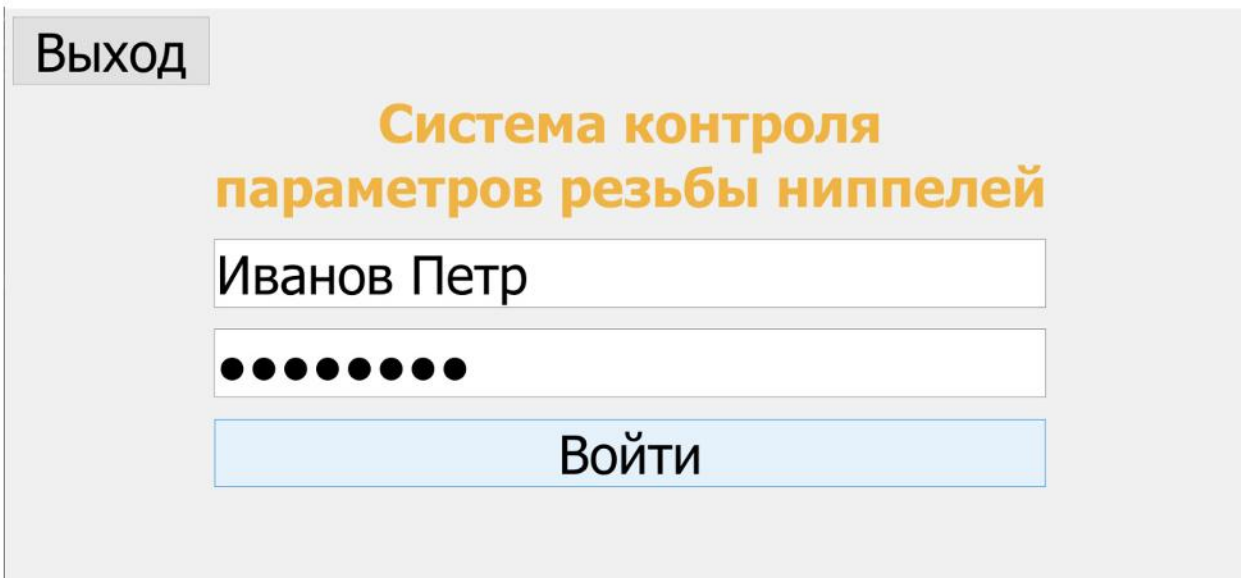


Рисунок 2.1. Интерфейс ввода логина и пароля.

После чего в соответствии с уровнем доступа появляется окно выбора действий (рисунок 2.2).



Рисунок 2.2. Окно выбора действий.

При входе оператора в раздел измерение резьбы появляется основное окно интерфейса, реализующее работу с отчётами об измерениях (рисунок 2.3). Для измерения ниппеля нужно ввести номер серии и делала и выбрать тип резьбы, если он не один. После измерения отображается протокол, а также эта информация сохраняется. Реализована функция сохранения протоколов на внешний носитель, кнопка «сохранить» отвечает за копирование отчета на Flash диск.

Назад	Иванов Петр	Измерить	База измерений		
Измерение резьбы нииппеля					
633-80 А 73	Серия / номер:	5	438	Дата:	
	Измерено Отклонение Нижний доп.				
Сохранить					

Рисунок 2.3. Основное окно интерфейса.

При входе в раздел «База измерений» для каждого оператора отображаются протоколы, которые были получены в результате его работы (рисунок 2.4). Протоколы можно выделять и экспортировать на внешние носители.

Назад

База данных протоколов измерений

Экспортировать Удалить

	#	Оператор	Дата	Тип	Серия	Номер
■	0023	Иванов Петр	19.04.2019 20:01	A 73	2	3
■	0022	Иванов Петр	17.04.2019 19:06	A 73	2	2
■	0021	Иванов Петр	23.03.2019 14:16	A 73	3	1

Рисунок 2.4. Отображение протоколов, которые были получены в результате измерений.

Для администратора доступны все протоколы. Также ему доступно удаление и редактирование протоколов (рисунок 2.5).

Назад Иванов Петр Измерить < >

Измерение резьбы ниппеля

633-80 A 73 Серия / номер: 2 3 Дата: 20.04.19 10:08

Экспортировать Удалить Редактировать Отчет: 0024

	Измерено	Номинал	Отклонение	Нижний доп.	Верхний доп.
Наружный диаметр	73.6307	73.0000	0.6307	-1.0000	1.0000
Наружный диаметр резьбы в плоскости торца	<input type="text" value="70.6277"/>	70.5060	0.1217	-0.1610	0.2140
Диаметр фаски в плоскости торца	74.8777	74.6000	0.2777	0.0000	1.0000
Угол наклона левой стороны профиля	30.3253	30.0000	0.3253	-1.0000	1.0000
Угол наклона правой стороны профиля	30.2457	30.0000	0.2457	-1.0000	1.0000
Радиус закругления вершины профиля r	0.4299	0.4320	-0.0021	0.0000	0.0450
Радиус закругления впадины профиля r1	0.3434	0.3560	-0.0126	-0.0450	0.0000
Высота профиля h1	1.4423	1.4120	0.0303	-0.1000	0.0500
Шаг резьбы средний	2.5429	2.5400	0.0029	-0.0120	0.0120

Сохранить

Рисунок 2.5. Редактирование протоколов администратором.

Реализована возможность работы с облаком точек, полученным в результате сканирования (рисунок 2.6).

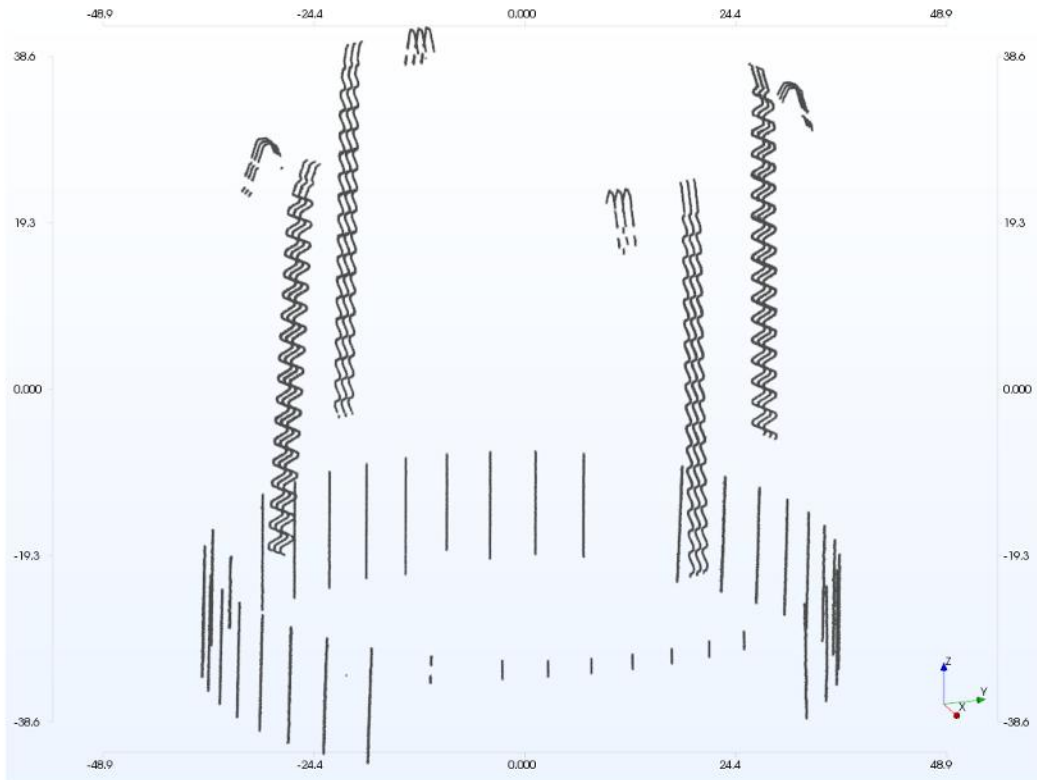


Рисунок 2.6. Работа с облаком точек

Также есть возможность анализа геометрических примитивов, созданных для измерений резьбы (рисунок 2.7).

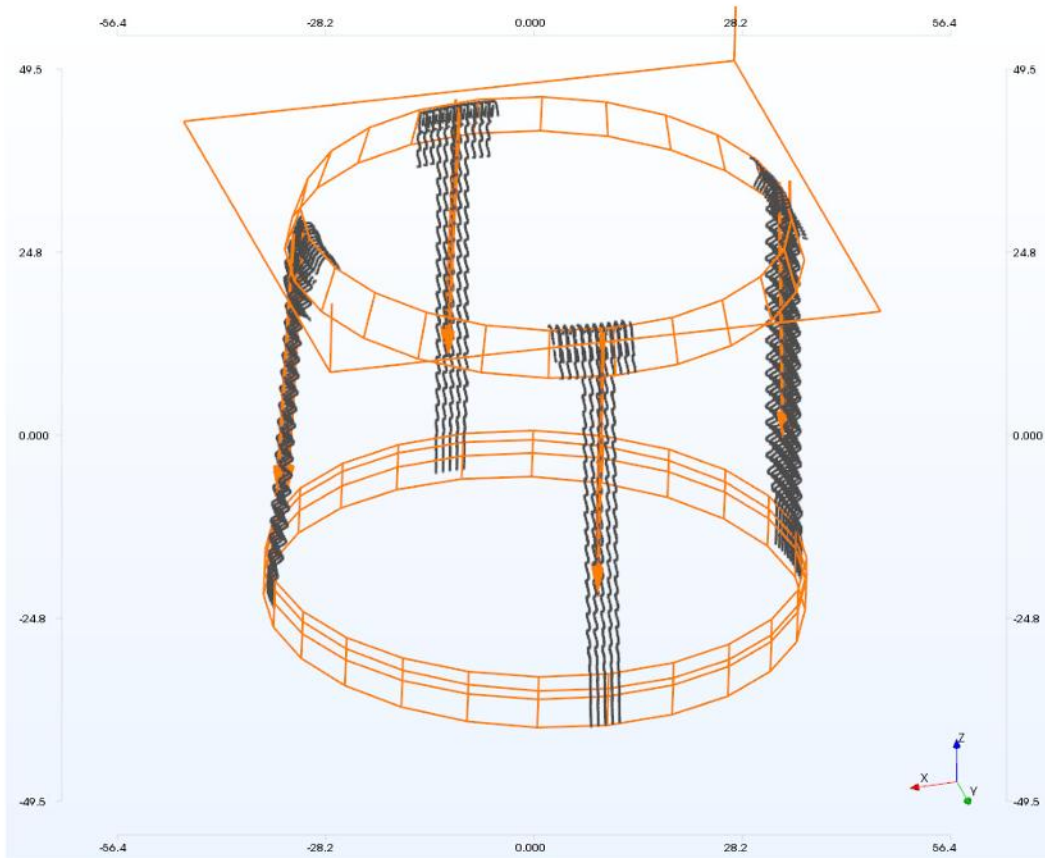


Рисунок 2.7. Анализ геометрических примитивов

При распечатывании протокол измерений выглядит как на рисунке 2.8.

Протокол № 0024
Карта контроля ниппеля А Ø 73
Номер партии и детали: 2 3

Параметр	Измеренное значение	Номинальное значение	Отклонение	Нижний допуск	Верхний допуск
Наружный диаметр	73.630 ₇	73.000 ₀	0.630 ₇	-1.000 ₀	1.000 ₀
Наружный диаметр резьбы в плоскости торца	70.627 ₇	70.506 ₀	0.121 ₇	-0.161 ₀	0.214 ₀
Угол наклона левой стороны профиля	30.325 ₃	30.000 ₀	0.325 ₃	-1.000 ₀	1.000 ₀
Угол наклона правой стороны профиля	30.245 ₇	30.000 ₀	0.245 ₇	-1.000 ₀	1.000 ₀
Радиус закругления вершины профиля r	0.429 ₀	0.432 ₀	-0.002 ₀	0.000 ₀	0.045 ₀
Радиус закругления впадины профиля r1	0.343 ₄	0.356 ₀	-0.012 ₆	-0.045 ₀	0.000 ₀
Высота профиля h1	1.442 ₃	1.412 ₀	0.030 ₃	-0.100 ₀	0.050 ₀
Шаг резьбы средний	2.542 ₀	2.540 ₀	0.002 ₀	-0.012 ₀	0.012 ₀
Шаг резьбы по 5ти виткам минимальный	2.535 ₈	2.540 ₀	-0.004 ₂	-0.037 ₅	0.037 ₅
Шаг резьбы по 5ти виткам максимальный	2.553 ₁	2.540 ₀	0.013 ₃	-0.037 ₅	0.037 ₅
Длина резьбы 10ти витков минимальная	25.432 ₇	25.400 ₀	0.032 ₇	-0.075 ₀	0.075 ₀
Длина резьбы 10ти витков максимальная	25.444 ₈	25.400 ₀	0.044 ₈	-0.075 ₀	0.075 ₀
Угол уклона	1.737 ₀	1.789 ₉	-0.052 ₉	-0.103 ₀	0.063 ₀
Уклон на 100 мм	6.065 ₂	6.250 ₀	-0.184 ₈	-0.360 ₀	0.220 ₀
Уклон на 25.4 мм	1.540 ₆	1.500 ₀	0.040 ₆	0.000 ₀	0.140 ₀

Рисунок 2.8. Анализ геометрических примитивов

9. Реализация модуля анализа трендов.

Графический объект для анализа трендов позволяет выводить данные со следующих источников:

- изменение значения переменных во времени;
- данные SIAD;
- данные индивидуальных архивов;
- исторические данные, полученные от серверов OPC и по прямым протоколам обмена.

Аргументы с типом данных с плавающей запятой, а также целочисленные, отображаются на графике (рисунок 3.1).

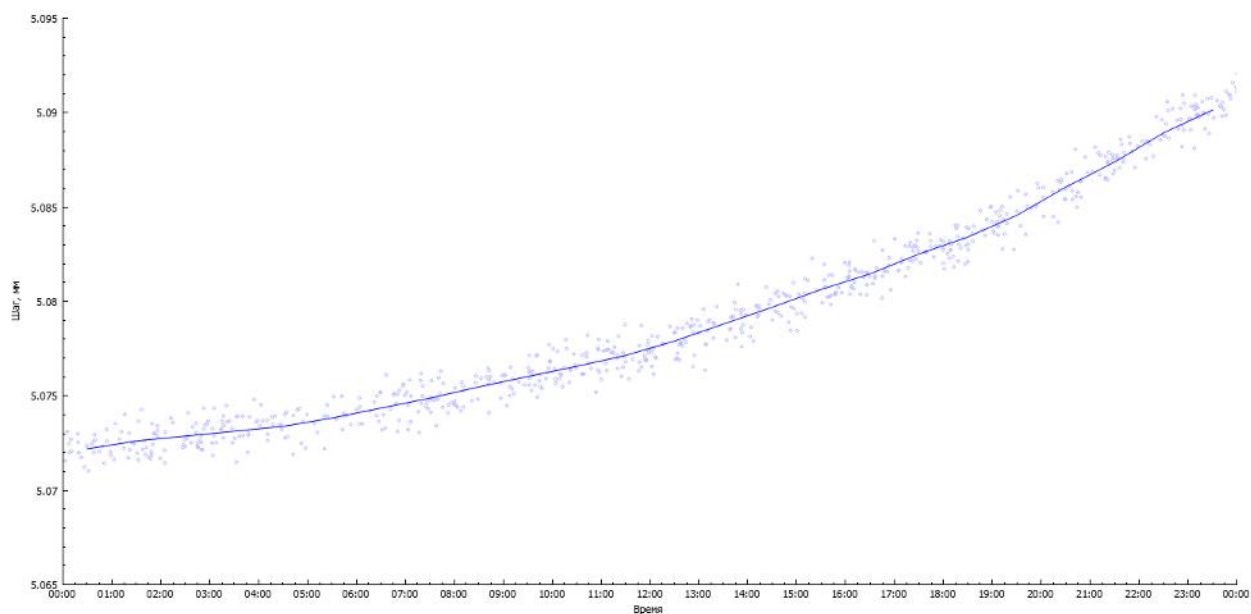


Рисунок 3.1. Графический объект с отображением прямой, соответствующей изменению шага зуба резьбы.

Помимо типовых свойств, графический объект Тренд имеет значительное количество специфических атрибутов.

Окно свойств объекта Тренд содержит вкладки – «Основные свойства» и «Кривые».

В Основных Свойствах конфигурируются следующие специфические атрибуты:

- Сетка – этот раздел содержит типовые инструменты задания параметров сетки тренда (рисунок 3.2);

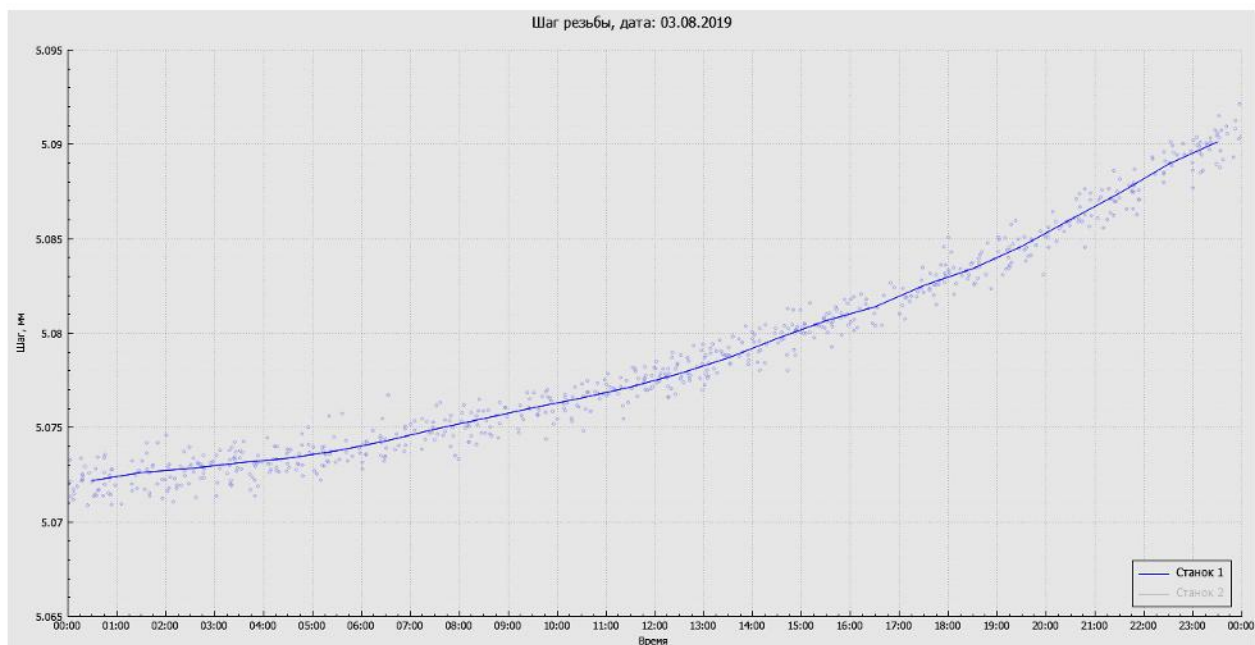


Рисунок 3.2. Графический объект с отображением сетки.

- Легенда – этот раздел содержит типовые инструменты задания параметров легенды, отображаемой части тренда. В легенду выводится имя кривой (имя аргумента, если для кривой не задан атрибут «Имя»); цвет и значение в точке, указанной визиром. Чтобы в реальном времени показать/скрыть кривую, надо нажатием ЛК установить/снять соответствующий флаг в столбце Кривые легенды (рисунок 3. 3);



Рисунок 3. 3. Графический объект с отображением двух кривых и легенды.

- Ось времени – этот раздел содержит настройки временной оси:
 - а. Показывать – если TRUE, на оси отображаются значения времени;
 - б. Разбиение – начальное количество делений видимой части оси;
 - в. Период подписи – период подписей на оси (в делениях). Линии сетки без подписи имеют цвет, заданный атрибутом «Доп. цвет» в разделе «Сетка»;
 - г. Диапазон – диапазон значений видимой части оси (0-100);

- д. Единицы – единицы измерения диапазона. Выбираются из меню: секунда, минута, час, день;
- е. Левая граница и Правая граница – аргументы, которые задают текущие значения временных границ тренда (рисунок 3. 4).

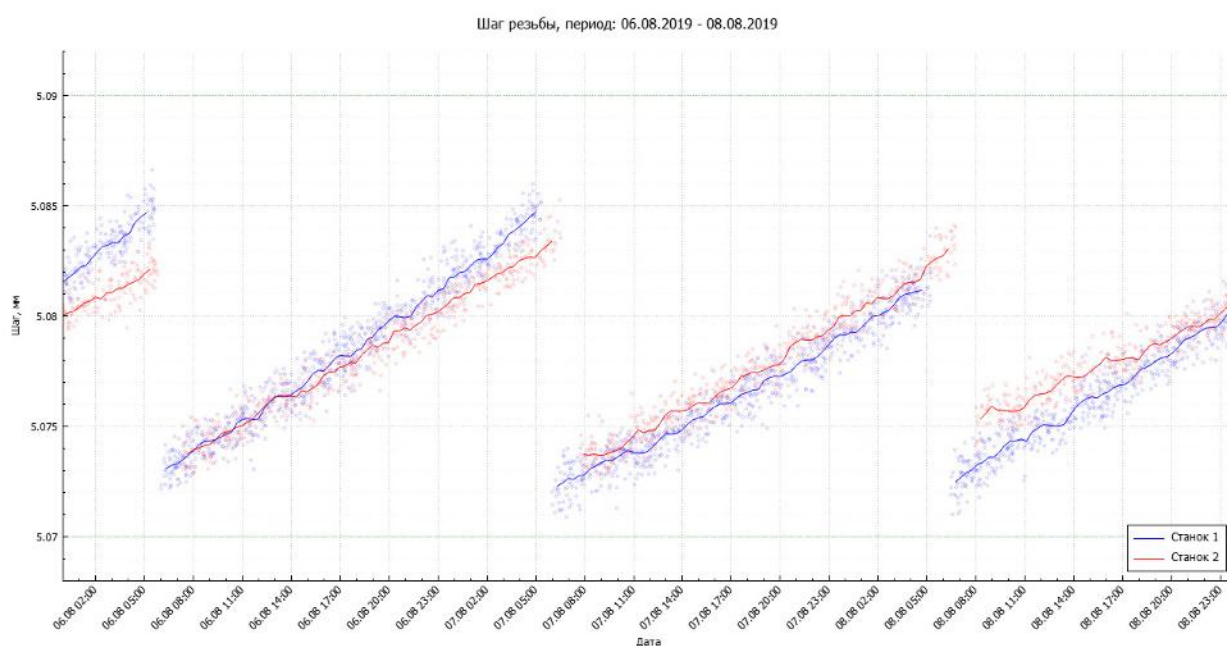


Рисунок 3. 4. Графический объект с отображением кривых на большом временном интервале.

- Ось значений – этот раздел содержит настройки оси значений:
 - а. Разбиение – начальное количество делений видимой части оси;
 - б. Период подписи - период подписей на оси (в делениях).
Линии сетки без подписи имеют цвет, заданный атрибутом «Доп. цвет» в разделе «Сетка».
 - в. Показывать – видимость всех значений:
 - Все оси – отображаются оси всех кривых;
 - Только активная – отображается ось кривой, выделенной в легенде.
- Буфер – количество хранимых в памяти значений каждой кривой для вывода в тренд (24-106, по умолчанию 500). Кроме того, этот параметр задает максимальное число значений, извлекаемых из

архива и отображаемых на тренде при переходе к заданной временной метке. Буфер можно сохранить в файл. Буфер сохраняется в память с именем <ID>.DRG, где ID – идентификатор канала, вызывающего шаблон экрана с трендом. Период сохранения буфера определяется параметром «Период сохранения доп. информации в настройках узла (вкладка Отчет тревог/Дамп/Параметры). При запуске проекта на тренде будут показаны значения последнего сохраненного буфера.

- Масштаб дискрет (%) – этот атрибут задает высоту поля, отводимого на панели для отображения одной кривой. Высота задается в процентах, относительно размера базового шрифта (100% значение по умолчанию).
- Цвета статусов – цвета для аргументов при их равенстве соответственно 0...7, отображаемых на панели. Для использования этих цветов атрибуту «Интерпретировать как» кривой должно быть установлено значение «Статус».

Чтобы получить на экране график, необходимо сконфигурировать хотя бы одну кривую на вкладке «Кривые»:

Для добавления кривой нужно выполнить команду «Добавить» из контекстного меню раздела «Кривые», для удаления/перемещения в списке – соответствующую команду из контекстного меню раздела «КриваяN» (рисунок 3. 5).

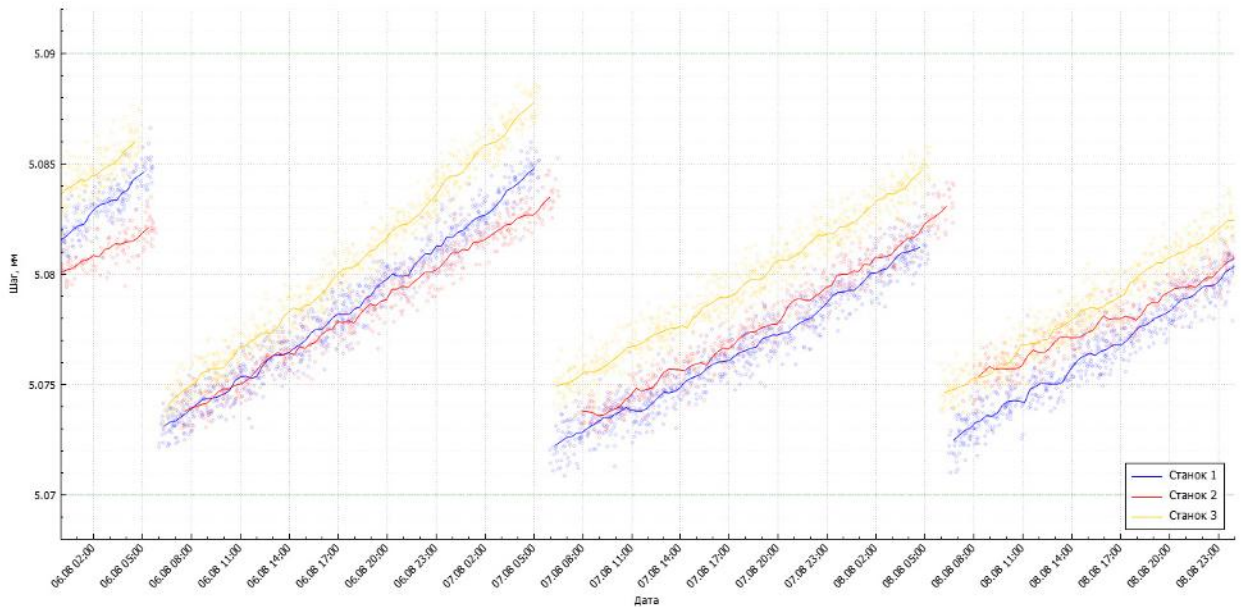


Рисунок 3. 5. Графический объект с отображением добавленной кривой.

Набор свойств кривой зависит от атрибута «Интерполировать как». Этот атрибут используется только при конфигурировании дискретной кривой. Он может принимать следующие значения:

- Значение – при равенстве аргумента 0 – тонкая линия, в противном случае – толстая линия; при изменении значения отображается метка, содержащая новое значение;
- Статус – толстая линия, образованная отрезками; цвет отрезка соответствует значению (задается атрибутом «Цвета статусов»); при изменении значения отображается метка, содержащая новое значение;
- Изменение (со значением) – толстая линия, образованная чередующимися отрезками двух различных тонов (тон меняется при изменении значения аргумента). При изменении значения отображается метка, содержащая новое значение;
- Изменение – метки изменения значения аргумента (рисунок 3. 6).

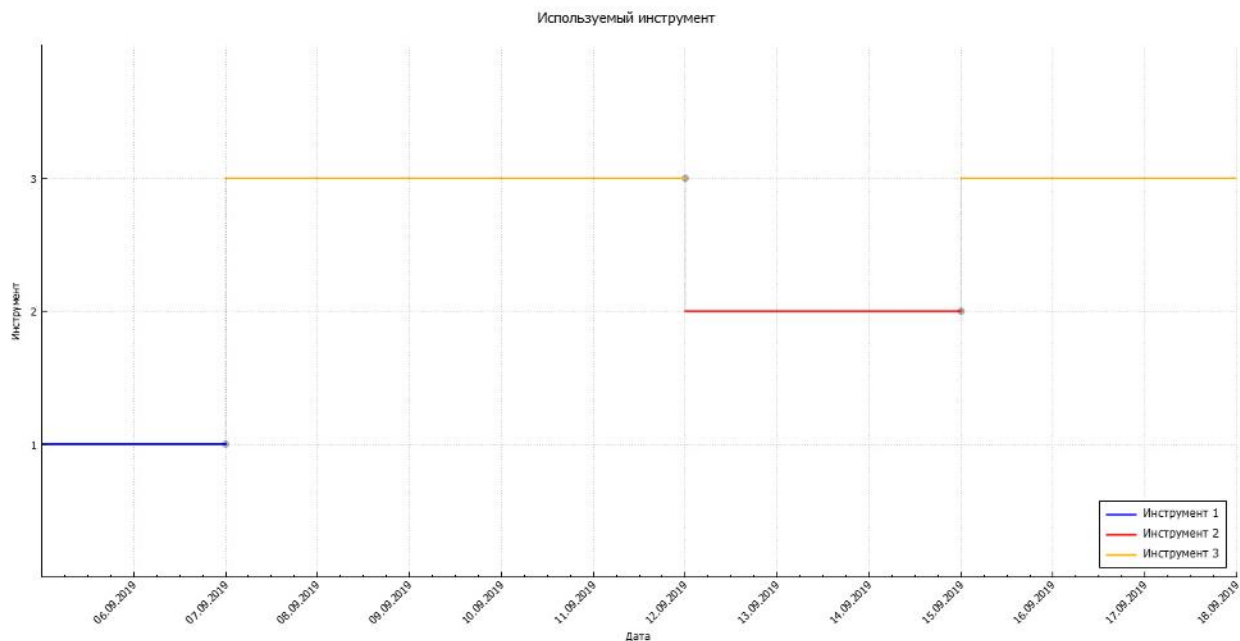


Рисунок 3. 6. Графический объект с отображением дискретной кривой.

Полный набор параметров кривой (для дискретной кривой актуальны только имя, привязка):

- Имя – имя кривой, при задании выводится в легенду, в противном случае в легенде отображает имя канала (0, если отображаемый аргумент привязан к аргументу или не имеет привязки).
- Привязка – привязка кривой к аргументу экрана. Для отображения значения атрибута канала (в том числе данных по этому атрибуту из SIAD) к аргументу должен быть привязан этот атрибут; для отображения индивидуального архива – атрибут 1,А соответствующего канала CALL.
- Цвет – цвет кривой.
- Стиль линии – стиль кривой.
- Толщина линии – толщина кривой в пикселях.
- Тип меток – выделение точки кривой с периодом изменения привязанного аргумента с помощью одной из следующих меток:
 - а. +
 - б. ×
 - в. •
 - г. ■

- Формат – формат значения кривой в легенде, должен соответствовать типу данных аргумента;
- Макс. значение – верхний предел оси значений.
- Мин. значение – нижний предел оси значений (рисунок 3. 7).

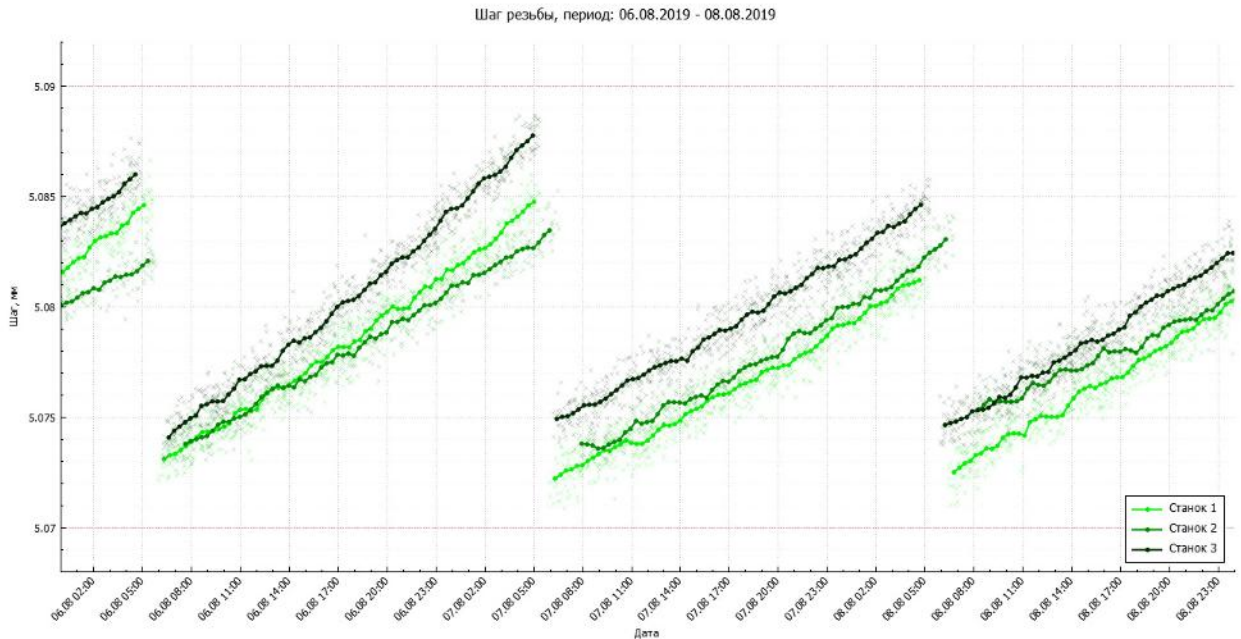


Рисунок 3. 7. Графический объект с установленными максимальными и минимальными значениями показателей.

- Интерполирование – этот атрибут может принимать следующие значения:
 - нет – нет интерполяции;
 - по периоду реального времени – интерполирование методом проведения прямой между текущим значением и его значением на предыдущем такте пересчета с удалением выбросов. Местные импульсные помехи значительно влияют на основные параметры распределения. Для удаления шумов в виде местных выбросов использованы сглаживающие фильтры, слабо влияющие на ровные участки. Для реализации предварительной фильтрации потребовалось восстановить информацию в невидимых точках, то есть интерполировать. В нашем случае невидимые точки приравнивались к среднему

арифметическому от значений координат правой и левой ближайших видимых точек, то есть применялась линейная интерполяция.

Если $y_n = -32768$, то $y_n = \frac{y_{n+k} + k \cdot y_{n-1}}{k + 1}$ и $x_n = \frac{x_{n+k} + k \cdot x_{n-1}}{k + 1}$, где $k \in \mathbb{N}$.

Данный метод может быть заменен на какой-либо другой, например, сплайн интерполяцию, что позволяет открытость программного кода и возможность использования Matlab (рисунок 3. 8).

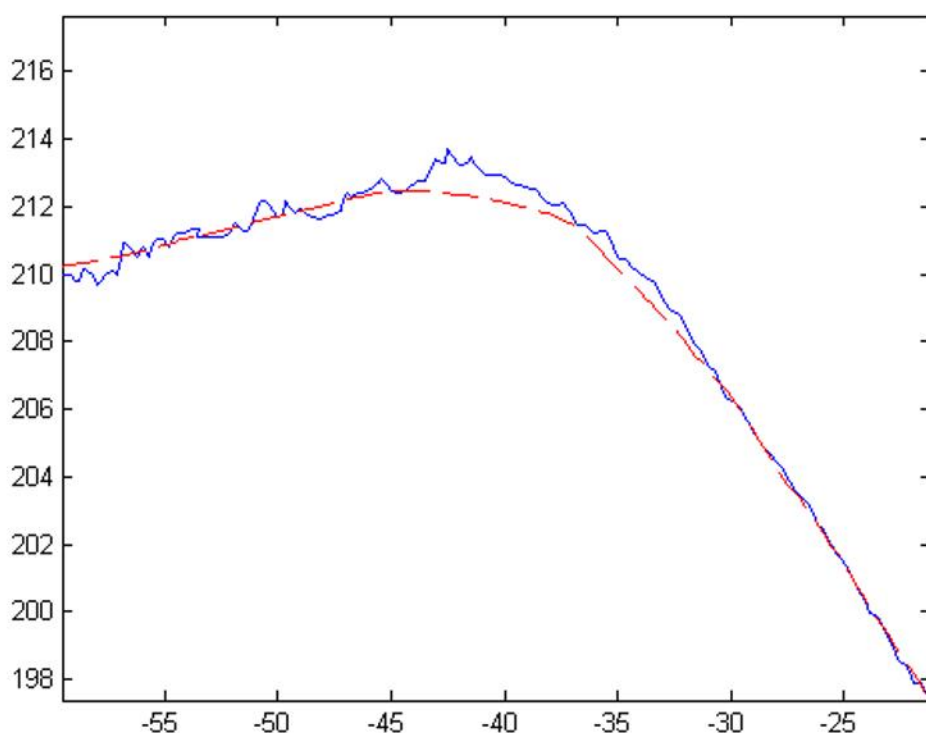


Рисунок 3. 8. Пример работы алгоритма интерполяции и фильтрации импульсов.

в. интерполяция с применением взвешенной локальной регрессии – интерполирование методом взвешенной локальной регрессии (рисунок 3. 9). При сглаживании при помощи взвешенной локальной регрессии для каждого

сглаживаемого значения данных, заданного в точке, выбирается набор из фиксированного числа рядом расположенных точек, каждой из которых назначается вес по следующей формуле [5]

$$w_i = \left(1 - \frac{|x_k - x_i|^3}{d(x_k)^3} \right),$$

где $d(x_k)$ - расстояние от x_k до наиболее удаленной точки из набора. То есть наибольший вес (равный единице) будет у y_k , а наименьший (равный нулю) у данных, расположенных на границах набора. Можно также воспользоваться разновидностью этого метода – робастной взвешенной локальной регрессией. Этот способ менее чувствителен по отношению к выбросам, поскольку выбор весов осуществляется за несколько итераций, на каждой из которых более удаленным от регрессионной кривой (найденной на предыдущей итерации) точкам назначаются меньшие веса, а если точка сильно удалена, то она совсем исключается при выполнении локальной линейной или квадратичной регрессии на следующей итерации.

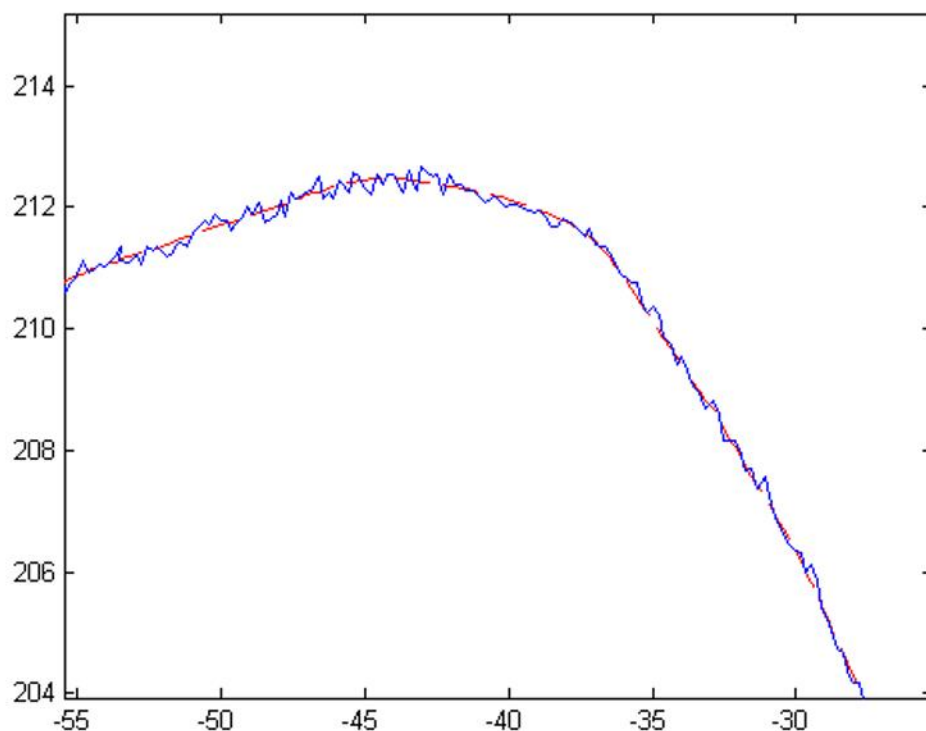


Рисунок 3. 9. Пример работы робастного алгоритма интерполяции и фильтрации импульсов.

Данные алгоритмы реализованы по умолчанию, но по опыту работы можно сказать, что для различных процессов требуются специальные алгоритмы, которые за счет гибкости открытого программного обеспечения возможно будет внедрить.

В реальном времени доступны следующие инструменты:

- Возврат к исходному масштабу тренда;
- Увеличение/уменьшение. Нажимая требуемые кнопки, можно приблизить или удалить требуемый участок тренда;
- вход в диалог редактирования кривых тренда, идентичный вкладке «Кривые» окна «Свойства объекта». Окно предназначено для добавления/удаления кривых и изменения их настроек в реальном времени (если на экране непосредственно (не в окне) размещен объект тренд, то данная функция тренда недоступна);
- переход к временной метке. При нажатии на эту кнопку появляется диалог задания даты времени (рисунок 3. 10):

После нажатия «ОК» на тренд выводится информация, начиная с указанного времени;

- Время визира – окно отображения информации о точке, указанной на графике визиром. Перемещая курсор вдоль кривой на графике в окне визира информация будет меняться в соответствии с выбранной точкой.

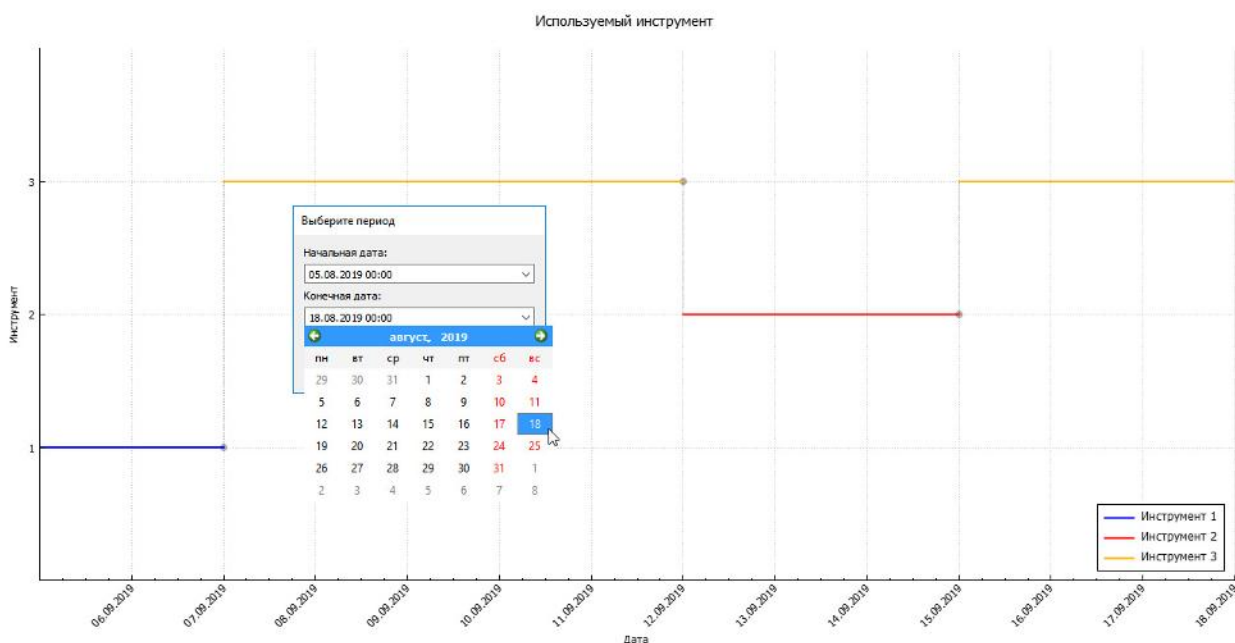


Рисунок 3. 10. Графический объект с диалоговым окном выбора даты.

- Для подключения функций Matlab к C++ Qt необходимо воспользоваться инструкцией доступной по адресу https://www.mathworks.com/help/matlab/matlab_external/compiling-engine-applications-in-an-ide.html