**СПЕЦИАЛЬНЫЕ ВОПРОСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ**

1. **Основные положения**

Одной из целей данной выпускной квалификационной работы является разработка алгоритмов тестирования ключевых узлов бортового синхронизирующего координатно-временного устройства (БСКВУ). БСКВУ предназначено для обеспечения комплекса управления космического аппарата (КА) и его целевой аппаратуры высокоточной навигационно-временной информацией и входит в состав малогабаритных КА. В связи с этим, очень важно предусмотреть систему тестов, охватывающую как можно большее число возможных неисправностей на этапе изготовления устройства.

Все разработанные алгоритмы реализованы програмно и, таким образом, было разработано соответствующее ПО для тестирования ­– консольное приложение, позволяющее поэтапно проверить требуемые узлы БСКВУ. Написание программы осуществлялось на персональном компьютере с использованием языка программирования С++ в кроссплатформенной интегрированной среде разработки (IDE) Qt Creator 5.7. Для обеспечения продуктивной работы, комфорта и безопасности необходимо, чтобы интерфейсы Qt Creator 5.7 и разработанного ПО соответствовали ГОСТ Р ИСО 9241.

Эргономика – научная дисциплина, изучающая взаимодействие человека с другими элементами системы [ссылка на ГОСТ Р ИСО 9241]. На сегодняшний день сложно представить себе жизнь без ПК. Каждый человек пользуется им не только на работе или учебе, но и дома. Поскольку взаимодействие с компьютером осуществляется через интерфейсы программ, важно поддержание их эргономичности, в целях обеспечения комфортной и безопасной среды для человека. В случае длительной и частой работы при несоблюдении требований ГОСТ Р ИСО 9241 у пользователя, помимо снижения продуктивности, могут возникать проблемы со здоровьем.

Поэтому, чтобы избежать любого риска безопасности человека, необходимо соответствие используемого и разрабатываемого ПО требованиям ГОСТ Р ИСО 9241.

1. **Эргономика интерфейса Qt Creator 5.7**

В данном разделе анализируется пригодность используемой IDE Qt Creator 5.7 с точки зрения эргономики.

Взаимодействие человека с ПО осуществляется через интерфейс программы. Пользовательский интерфейс (ПИ) – компоненты интерактивной системы, которые предоставляют пользователю информацию и возможность управления для выполнения задач [ссылка на ГОСТ Р ИСО 9241]. Таким образом, с помощью ПИ человек способен вести диалог с компьютером: управление его работой, выдача заданий, получение ответов на запросы и другой информации со стороны ПК.

ГОСТ Р ИСО 9241 определяет следующие принципы осуществления диалога между человеком и пользовательским интерфейсом разрабатываемого программного обеспечения:

* пригодность для выполнения задачи;
* информативность;
* соответствие ожиданиям пользователя;
* пригодность для обучения;
* управляемость;
* устойчивость к ошибкам;
* пригодность для индивидуализации.

Для оценки соответствия ПО Qt Creator требуемым принципам разработки диалога, далее будет подробно рассмотрено каждое из них, применительно к используемой IDE.

На рисунке 1 представлено стартовое окно Qt Creator, возникающее при запуске программы.

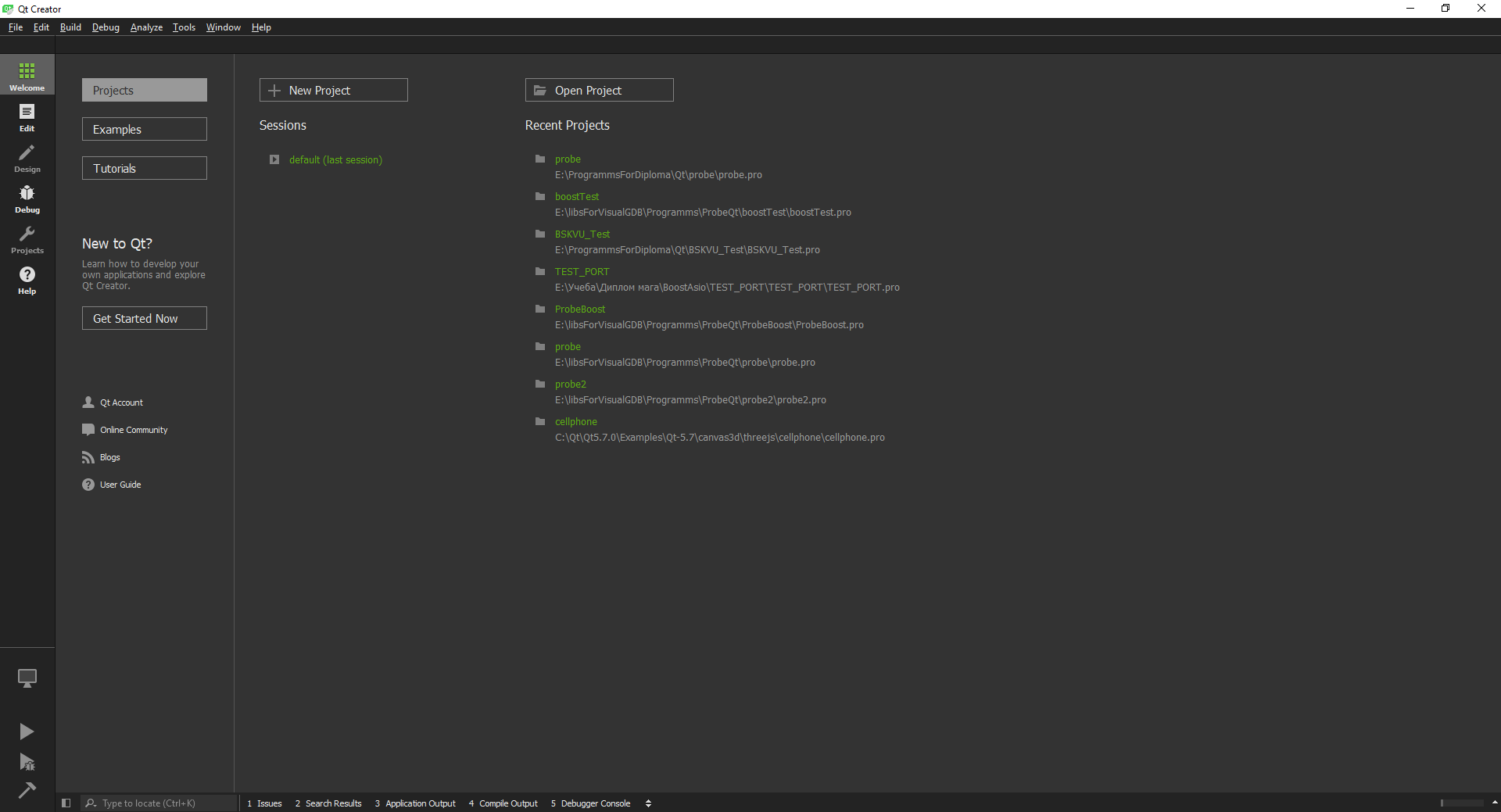


Рисунок 1 – Стартовое окно Qt Creator 5.7

Программное обеспечение располагает высоким уровнем информативности. В основном окне интерфейса, а также в левой и верхней частях присутствуют вкладки и кнопки, позволяющие быстро перейти к осуществлению требуемой операции: открыть имеющийся проект, создать новый, открыть пример и т.д. При создании нового проекта, всплывает дополнительное окно, представленное на рисунке 2.

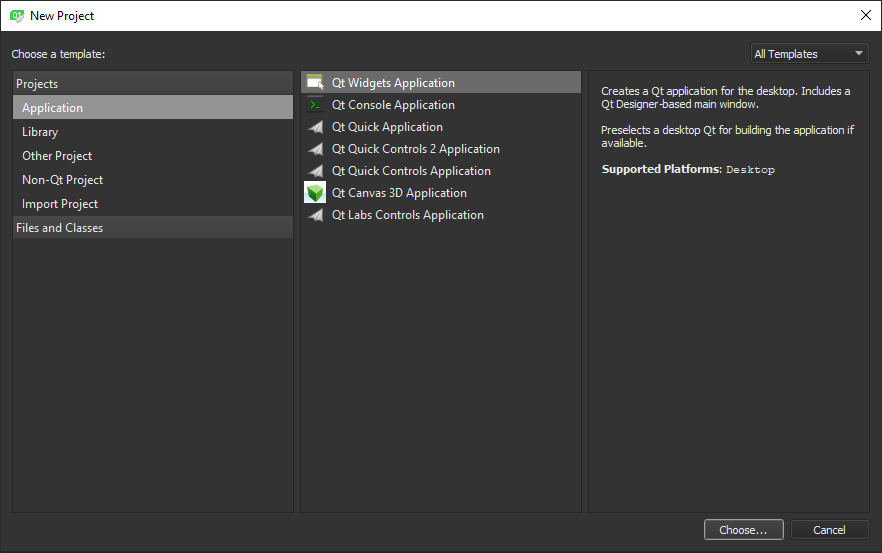


Рисунок 2 – Окно создания проекта

В левой части окна представлены разделенные по директориям шаблоны возможных проектов. В центральной части – наименование самих шаблонов. В правой части представлено описание содержания каждого шаблона: что он из себя представляет и какие возможности включает.

Qt Creator имеет возможности для обучения. Рассмотрим вновь стартовое окно программы (см. рисунок 1). В основной области окна представлены 2 вкладки: «Примеры» и «Уроки», содержащие в себе огромное количество всевозможных примеров и обучающих роликов, которые позволят неопытному пользователю в кратчайшие сроки начать работу в данном ПО. Так же в верхней части стартового окна присутствует вкладка «Помощь», при нажатии на которую появляется всплывающее окно с различной справочной информацией (см. рисунок 3).

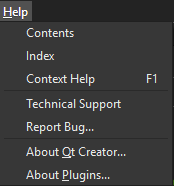


Рисунок 3 – Содержимое вкладки «Помощь»

IDE также имеет возможность индивидуализации пользовательского интерфейса, в целях обеспечения удобства каждого конкретного человека. В настройках среды можно задать цветовое оформление интерфейса (см. рисунок 4). Здесь присутствуют несколько готовых оформлений, однако в случае необходимости можно самостоятельно определить шрифт и размер текста, его цвет и т.д. Окно редактора кода можно разделить на 4 области, как это показано на рисунке 5. Размеры всех этих областей можно менять исходя из потребностей, а также добавлять новые и закрывать имеющиеся.

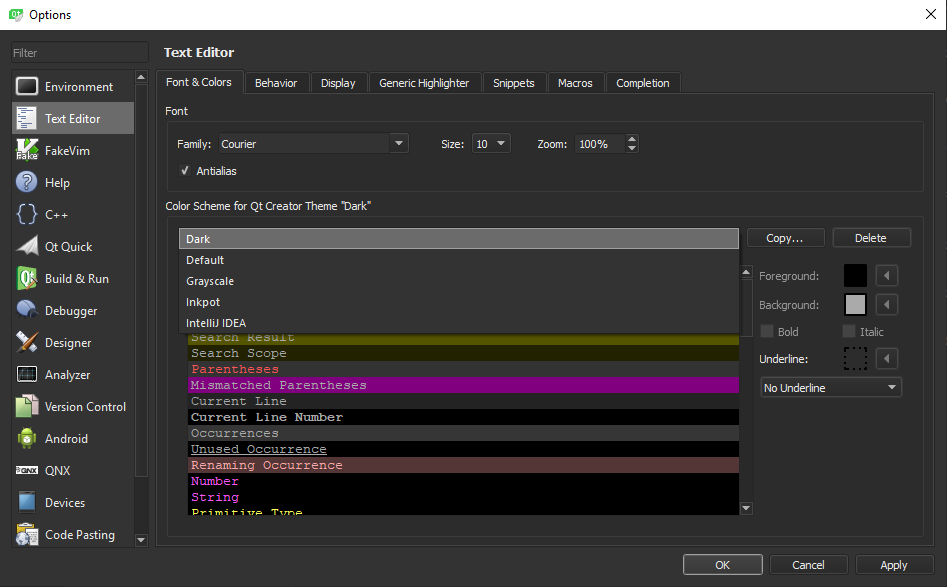


Рисунок 4 – Настройки интерфейса Qt Creator

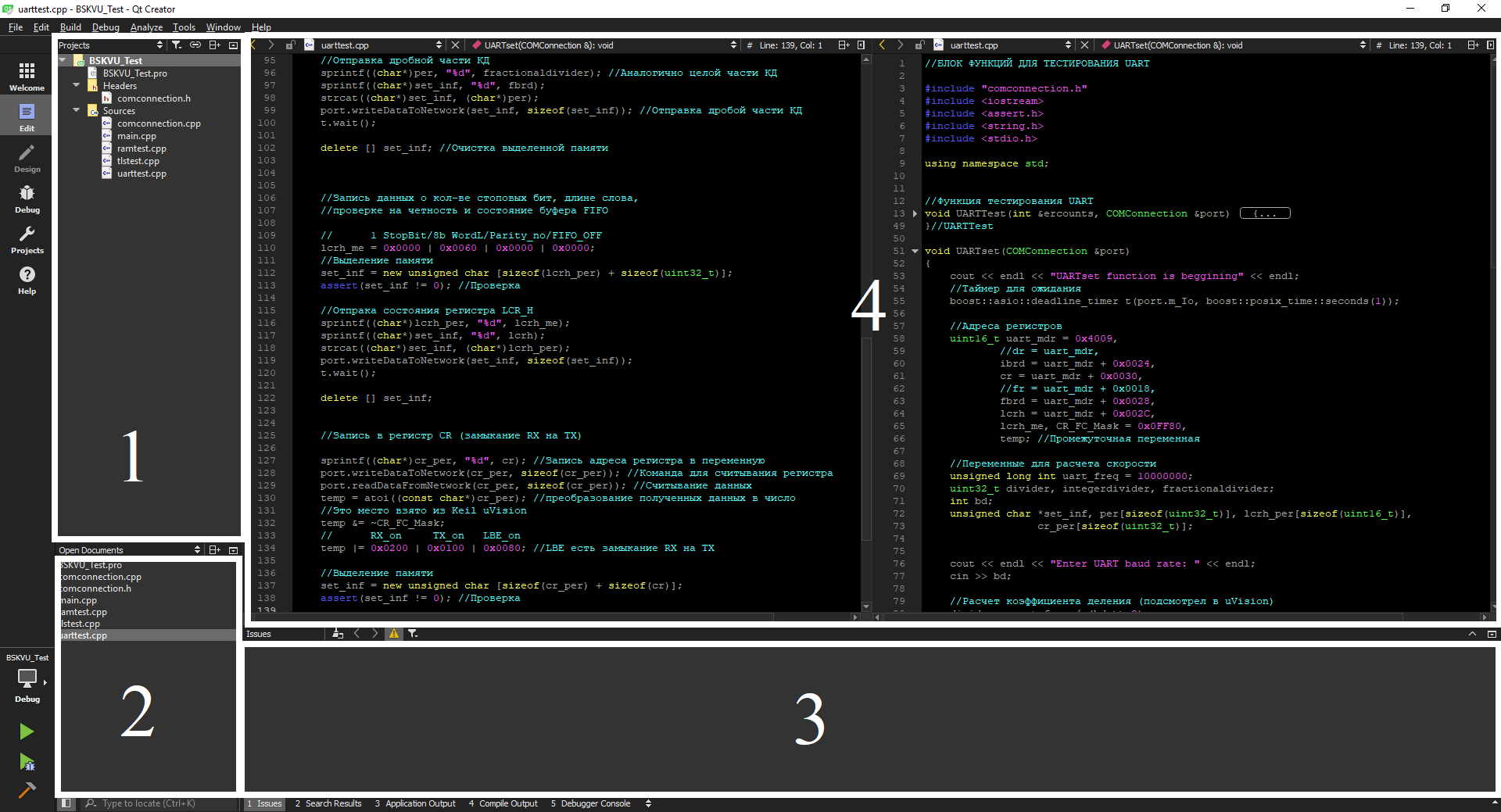


Рисунок 5 – Окно редактора кода

1 – открытые проекты, 2 – открытые документы, 3 – окно ошибок,

4 – область редактора кода

В случае, если в программном коде возникают ошибки во время написания, большая часть из них обнаруживается средой разработки еще до этапа компиляции, а строки с ошибками подчеркиваются, как это представлено на рисунке 6. При этом, если навести на ошибку, высвечивается пояснение.

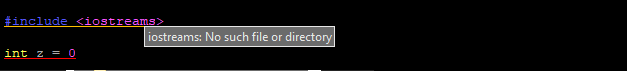


Рисунок 6 – Обнаружение ошибок в коде

В случае возникновения более серьезных нарушений, в процессе компиляции среда в отдельном окне выведет все ошибки с сопутствующей информацией, как это показано на рисунке 7.



Рисунок 7 – Окно вывода ошибок

Стандартные возможности управления запуском программы отмечены на рисунке 8. Здесь присутствуют кнопки для запуска компиляции и сборки проекта, запуска отладки и запуска самой программы на исполнение.



Рисунок 8 – Окно запуска программы

Свободная кроссплатформенная интегрированная среда разработки Qt Creator представляет собой инструмент с широкими возможностями для решения всевозможных задач разработки программного обеспечения любого назначения на языке C и C++. Помимо встроенных библиотек, она имеет возможность подключения сторонних. Таким образом, Qt Creator пригоден для работы во многих областях, в том числе для решения задач, поставленных в данной работе.

Исходя из всего вышесказанного, можно заключить, что интерфейс данной среды разработки эргономичен и соответствует требованиям и принципам ГОСТ Р ИСО 9241.

1. **Эргономика интерфейса разработанного ПО**

Помимо принципов разработки интерфейса программного обеспечения, упомянутых в предыдущем подразделе, оно также должно обладать результативностью, эффективностью, удовлетворенностью и доступностью [ссылка на ГОСТ Р ИСО 9241]. На рисунке 9 представлен начальный вид окна разработанной программы.

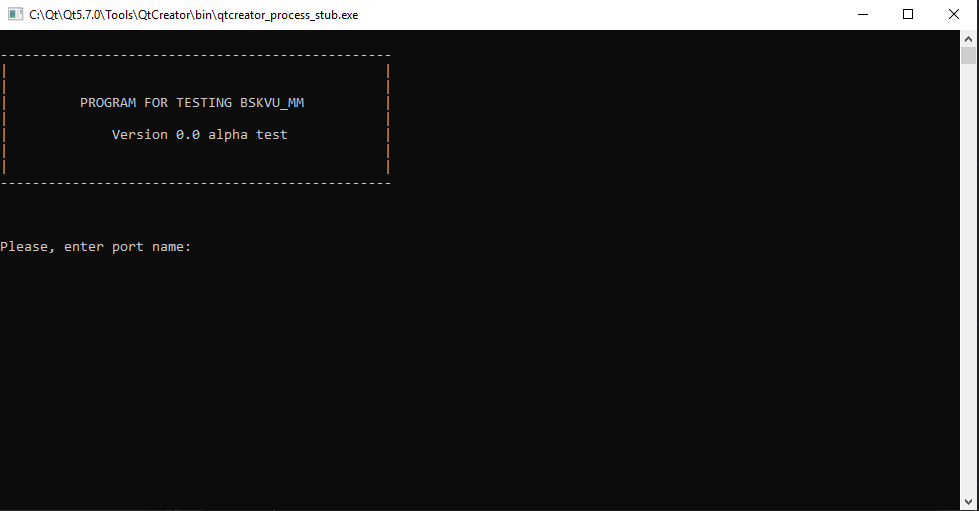


Рисунок 9 – Начальное окно программы.

Программа имеет объединенное окно ввода-вывода для осуществления диалога с пользователем. Сам же диалог в программе строится по принципу «вопрос-ответ». Для ввода информации используется клавиатура. Навигация по рабочему окну осуществляется с помощью мыши полосой прокрутки в правой части консольного окна программы.

Интерфейс программы не перегружен излишней информацией и поддерживает постоянный диалог с пользователем. Так, при вводе некорректного имени порта, программа выводит соответствующее сообщение об ошибке (см. рисунок 10).

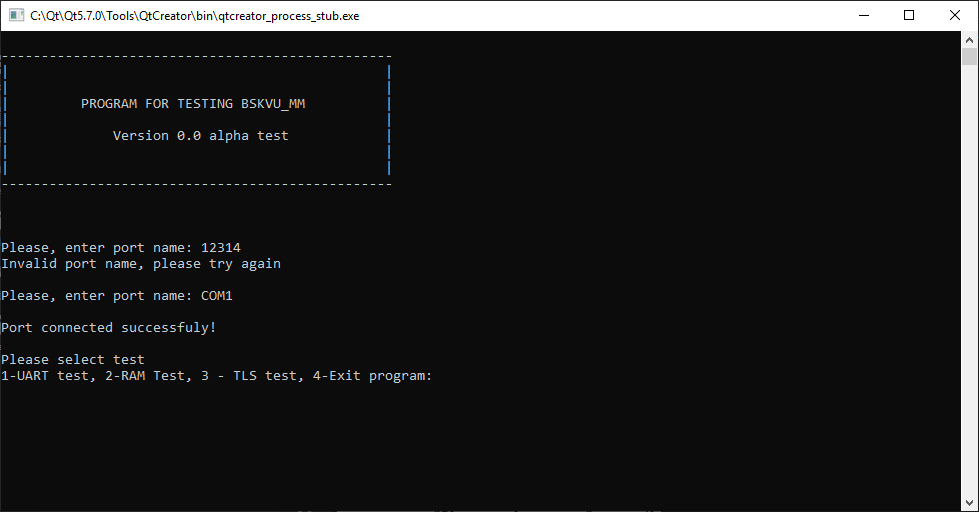


Рисунок 10 ­– Демонстрация наличия вывода ошибочного ввода

При наличии каких-либо ветвлений, подразумевающих выбор одной из опций программы, на экран выводится соответствующее сообщение с возможными вариантами дальнейших действий. После выбора теста, на экране появится сообщение о начале выполнение одной из подпрограмм, как это показано на рисунке 11.

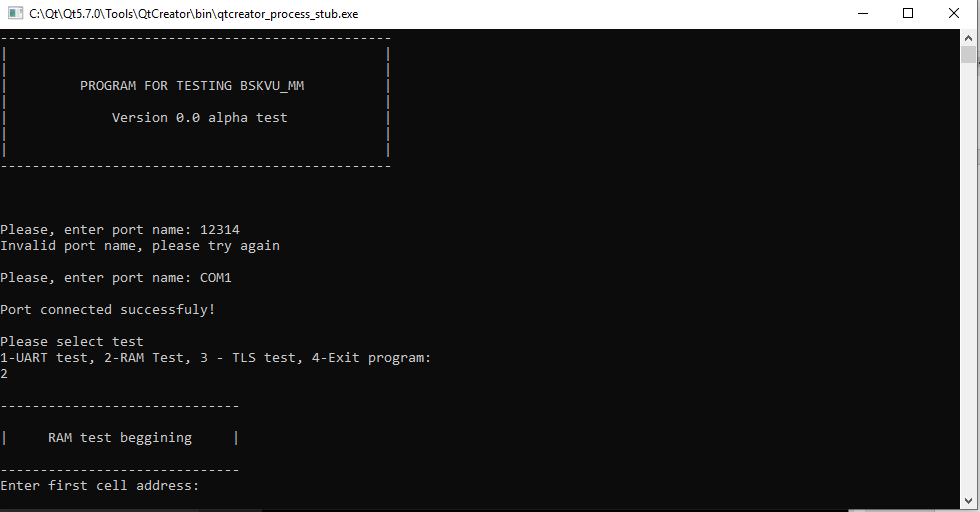


Рисунок 11 – Демонстрация выбора выполняемого теста

В процессе выполнения подпрограммы, на окне вывода будут появляться сообщения, информирующие пользователя о промежуточных результатах. По окончании работы подпрограммы выводится окончательный результат. Таким образом, все компоненты программного обеспечения имеют заголовки, подписи и короткие описания для удобства работы оператора.

Исходя из вышесказанного, можно сделать вывод, что пользовательский интерфейс разработанного ПО соответствует требованиям ГОСТ Р ИСО 9241 и пригоден для осуществления тестирования бортового синхронизирующего координатно-временного устройства.

1. **Аспекты функциональной безопасности**

ЭВМ является сложным устройством, способным решать широкий спектр потребительских задач и выполнять огромное количество функций. Однако правильность работы зависит не только от правильности функционирования его аппаратной части, но и программных средств. Надежность ПО обладает своей спецификой: зачастую, внося изменения в программу, программист не знает, как это изменение может повлиять на работоспособность ПО. Тем самым, любая модернизация программных средств может привести к новым ошибкам, которые сложно (а зачастую невозможно) определить на этапе тестирования, поскольку они могу проявляться только при определенных условиях.

Чтобы дать оценку рискам возникновения неполадок, необходимо составить перечень возможных отказов ПО, ранжировать по приоритету действия по ликвидации отказов и по их предупреждению. Это можно сделать с помощью FMECA анализа с вычислением значения приоритетности риска (англ. Risk Priority Number – RPN).

Для любого программного обеспечения можно выделить следующие причины нарушения работоспособности:

* ошибки, скрытые в программе;
* искажение входной информации;
* неверные действия пользователя;
* неисправность аппаратуры установки, на которой реализуется вычислительный процесс.

Каждый хороший программист перед выпуском своей программы подвергает ее процедуре отладки. Однако далеко не все ошибки можно обнаружить и исправить с помощью этого инструмента. В конечном счете, в коде программы могут оставаться скрытые ошибки, искажающие логику работы ПО. К ним можно отнести ошибки ввода-вывода. Они могут проявляться в неверном формировании команд для тестируемого устройства, в связи с чем, оно будет выполнять неправильные действия, что, несомненно, приведет к ошибочному результату. Данный риск можно оценить по рангу тяжести на 8 (очень высокая) – потеря основного функционала, по рангу появления – 3 (поскольку появление такой ошибки является случайностью), ранг обнаружения – 1 (такая ошибка будет обнаружена при первом же запуске программы).

Следующая причина проявляется при выходе входных данных из области допустимых значений. Это может произойти в результате сбоев в работе устройства передачи и приема информации, потери или изменении информации в накопителе вычислительной системы и т.д. Также сюда можно отнести ошибки в документации, которая используется при подготовке данных. Такой вид ошибок пагубно влияет на работоспособность программы (вплоть до потери функциональности). Таким образом ранг тяжести – 7, ранг появления – 1, вероятность обнаружения – 2.

Неверные действия пользователя в первую очередь связаны с неправильной интерпретацией сообщений от программы и с неправильными действиями в работе с ПО. В случае, если программное обеспечение эргономично и имеет качественную программную документацию, то риск возникновения ошибок минимален. Ранг тяжести – 5, ранг появления – 1, вероятность обнаружения – 1.

В результате неисправности аппаратуры может произойти искажение данных и текста самой программ в основной и внешне памяти. Подобная ошибка трудно диагностируема, поскольку не зависит от разработчика ПО и приводит в полной потере работоспособности программы. Ранг тяжести – 10, ранг появления – 1, вероятность обнаружения – 5.

Определив основные риски возникновения неполадок, необходимо рассчитать RPN и составить ранжированный список.

RPN рассчитывается по формуле (1) [ссылка на ГОСТ Р МЭК 61505 ]

 (1)

где S – ранг тяжести, О – ранг появления, D – ранг обнаружения.

В результате расчета RPN можно составить ранжированный список приоритета предотвращения рисков, 1 – элемент, которому необходимо уделить особое внимание, а о последнем можно беспокоится в последнюю очередь:

1. неисправность аппаратуры;
2. ошибки, скрытые в программе;
3. искажения входной информации;
4. неверные действия пользователя.

Следующим шагом является определение доли безопасных отказов (англ. Safe Failure Fraction, SSF), которая рассчитывается по формуле (2):

 (2)

где  – интенсивность безопасных отказов,  – интенсивность опасных отказов,  – суммарная интенсивность отказов. Из выше представленных причин неполадок опасными являются: неисправность аппаратуры, ошибки, скрытые в программе и искажение входной информации. Безопасным можно считать только неверные действия пользователя.

Таким образом, доля безопасных отказов равна 1. Это можно объяснить тем, что при расчете SSF принимаются во внимание только диагностируемые неисправности. В разработанном ПО маловероятно появление недиагностируемых ошибок. При детальном доля безопасных отказов может снизится, но вряд ли опуститься ниже 90%. Таким образом, руководствуясь ГОСТ Р МЭК 61508, разработанному программному обеспечению можно присвоить третий уровень полноты безопасности (УПБ 3).

1. **Выводы по разделу**

Программное обеспечения для тестирования бортового синхронизирующего координатно-временного устройства, разработанного в ходе выполнения выпускной квалификационной работы, обладает понятным, доступным и простым пользовательским интерфейсом, а также соответствует требованиям ГОСТ.

Так же в разделе были представлены возможные причины возникновения отказов разработанного программного обеспечения. В результате анализа диагностируемых факторов, которые могут повлиять на функциональные возможности ПО, ему был присвоен третий уровень полноты безопасноти.