

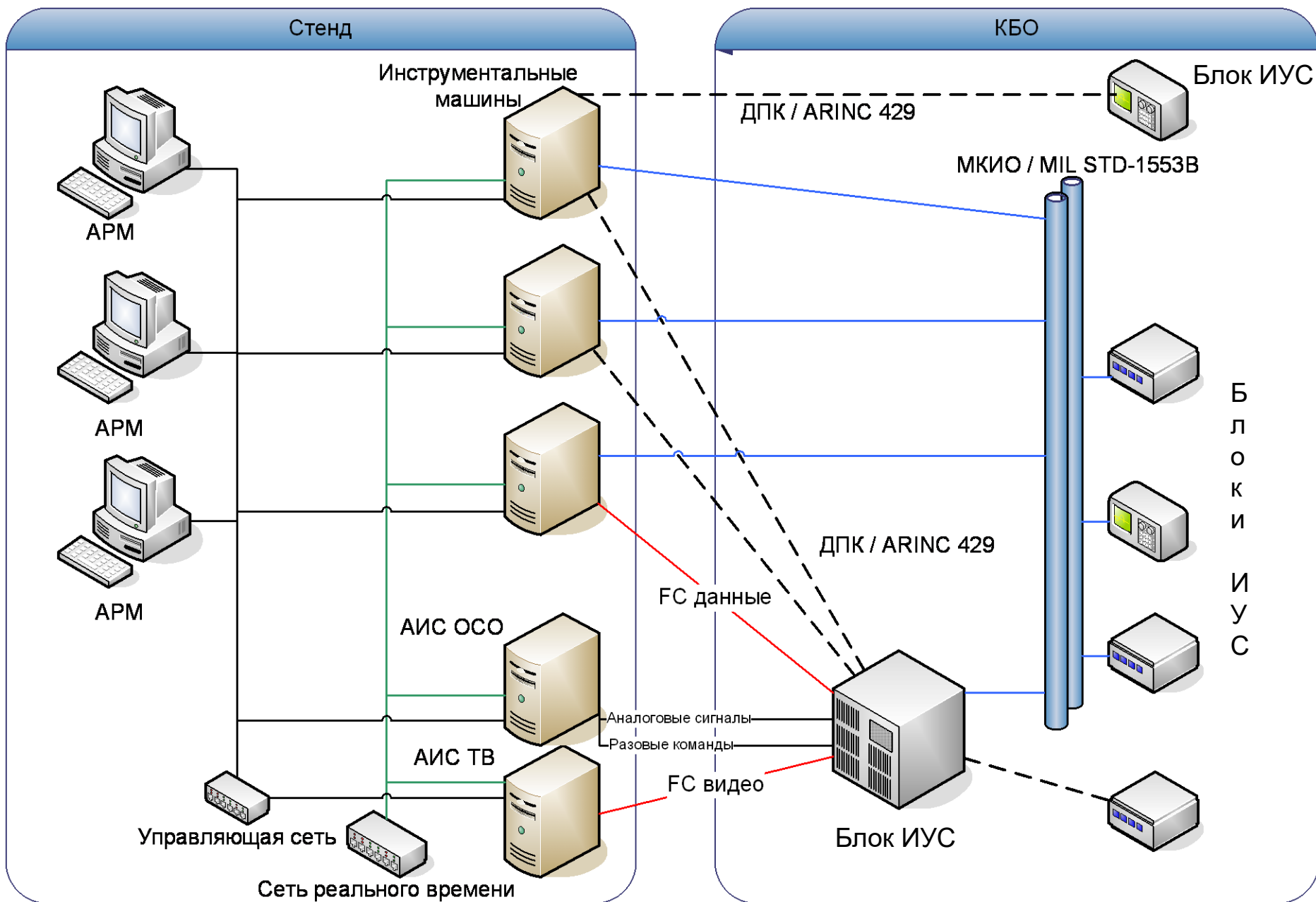
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

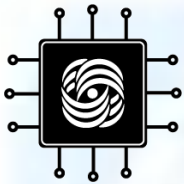
Лекция 11-Б:

Тестирование и интеграция ИУС РВ – средства функционального тестирования

Кафедра АСВК,
Лаборатория Вычислительных Комплексов
Балашов В.В.

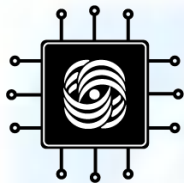
Архитектура стенда тестирования ИУС





Стенд отработки ПО БЦВМ

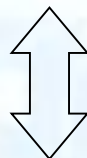




Структура ПО дискретно-событийной ИС

Средство оперативной визуализации и управления экспериментом

ARM



Компоненты моделирования:

- * вычислительная логика
- * упаковка/распаковка сообщений

Монитор событий

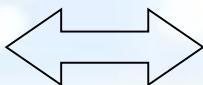
- * календарь событий

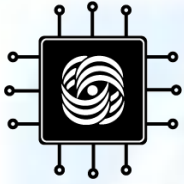
Асинхронная выдача в каналы

Драйверы адаптеров

Инструментальная
машина

ИУС





Программные средства функционального тестирования в составе стенда

Язык описания тестов (ЯОТ)

- Поддерживает тестирование ИУС через каналы бортовых интерфейсов (MIL-STD-1553B, ARINC 429, FC-AE-ASM, CAN, AFDX и т.п.) с использованием следующих основных элементов:
 - Сообщения и слова КБИ
 - Битовые поля сообщений/слов
- Предоставляет операторы для:
 - Определения структуры слов/сообщений (состав битовых полей, их имена и расположение)
 - Задания и обработки значений битовых полей
 - Автоматической проверки тестируемых условий
 - Взаимодействия с пользователем
 - Задания структуры тестового сценария

Основные понятия ЯОТ

- Тестовые компоненты (ТСК)
- Интерфейсы
 - Обеспечивают подключение к каналам БИ
- Сообщения бортовых интерфейсов
- Битовые поля сообщений
- Тестовые случаи (test case)

Тестовые компоненты

Тестовый компонент (ТСК) – это средство компоновки объектов ФТ в составе стенда. ТСК определяет состав и последовательность выполнения тестов.

С каждым ТСК связаны:

- *имя (определяет тип ТСК)*
- *описание сообщений КБИ с указанием их структуры*
- *описание тестов*

Структура ТСК

Заголовок ТСК определяет состав интерфейсов, состав и структуру сообщений КБИ, состав тестов и их привязку к требованиям

Тело ТСК определяет состав и логику выполнения тестовых случаев

Интерфейсы ТСК

Интерфейс – это объект средств ФТ, предоставляющий программный интерфейс для взаимодействия ТСК с натурными блоками ИУС, а также друг с другом

- Интерфейсы обеспечивают доступ ТСК к адаптерам КБИ
- Через интерфейсы обеспечивается передача и прием сообщений

Сообщение MIL STD-1553B

Сообщение MIL STD-1553B – это объект средств ФТ, соответствующий сообщению бортового интерфейса MIL STD-1553B.

С каждым сообщением связаны:

- *ИМЯ*
- *интерфейс MILS*
- *адрес*
- *подадрес*

С отдельными словами в составе сообщения можно работать как с элементами массива.

(сообщения для других типов интерфейсов устроены аналогично)

Сообщения MIL STD-1553B

RTS_OUTPUT_MESSAGE (<ИМЯ>,
<интерфейс MILS>,
<адрес>,
<подадрес>,
<кол-во слов в сообщении>,
<период в мс>)

RTS_INPUT_MESSAGE (<ИМЯ>,
<интерфейс MILS>,
<адрес>,
<подадрес>,
<кол-во слов в сообщении>,
<период в мс>)

RTS_OUTPUT_ / RTS_INPUT_TERM_MESSAGE (<ИМЯ>,
<интерфейс MILS>,
<подадрес>,
<кол-во слов в сообщении>)

Битовые поля и Биты

Битовое поле – это объект средств ФТ, соответствующий группе последовательных битов в составе слова сообщения.

Бит – это битовое поле, соответствующее 1 биту.

С каждым битовым полем связаны:

- *имя*
- *позиция в сообщении (номер первого слова, номер начального бита, кол-во битов)*
- *признак использования знакового или беззнакового поля*
- *коэффициент масштабирования (ЦМР)*

С битовым полем можно работать как с числовой переменной.

Битовые поля и биты

```
RTS_BITFIELD (<имя поля>,  
    <имя сообщения>[<номер слова>],  
    <начальный бит>,  
    <кол-во битов>  
)
```

```
RTS_BOOLEAN (<имя поля>,  
    <имя сообщения>[<номер слова>],  
    <позиция бита>)
```

Задание иерархического идентификатора теста

Идентификатор теста задается в формате:

<имя 1>.<имя 2>....<имя n>

Например: **«Су35.СОЛС.Наземный_контроль.17»**

RTS_TESTNAME (“<имя теста>”)

RTS_TESTNUM (<номер теста>)

RTS_TEST_SHOW ()

RTS_TEST_PASSED ()

RTS_TEST_FAILED ()

Задание соответствия требований идентификаторам тестов

```
RTS_DOCUMENT("<requirement specification document name>",  
    RTS_REQUIREMENT(<requirement identifier>, "<requirement  
name>"),  
    RTS_REQUIREMENT(<requirement identifier>, "<requirement  
name>"),  
    ...  
)
```

```
RTS_TESTSPEC ("<test name (or name template)>",  
    <requirement identifier>, <requirement identifier>, ...)
```


Задание соответствия требований идентификаторам тестов

Пример:

Заголовок

```
RTS_DOCUMENT( "Board computer system initialization",  
    RTS_REQUIREMENT(BCS.INIT.START, "Starting up MIL STD-1553B  
    subscribers"),  
    RTS_REQUIREMENT(BCS.INIT.WORK, "Starting BCS operation main  
    loop"),  
)  
  
RTS_TESTSPEC ("3.2.{1-4}", BCS.INIT.START)  
RTS_TESTSPEC ("3.2.{1-4}", BCS.INIT.WORK, BCS.SKV.WORK)
```

Тело

```
RTS_TESTNAME(3.2.2)
```

```
// логика тестового случая
```

Временная задержка, отправка данных

// Задержать выполнение теста на заданное время

RTS_DELAY (<duration (ms)>)

// Выдать сообщение в канал (для активных абонентов)

RTS_EXECUTE (<Message/word name>)

Взаимодействие с пользователем

```
// Отобразить значение объекта  
// и занести это значение в протокол  
RTS_DISPLAY (<object name>)
```

```
// Отобразить и записать в протокол текстовое сообщение  
RTS_COMMENT (<message text>)
```

```
// Приостановить выполнение теста  
// и ждать команды пользователя на продолжение  
RTS_PAUSE ()
```

```
// Запросить у пользователя заключение (тест пройден / не пройден)  
// и завершить тест в соответствии с полученным заключением  
RTS_CONFIRM ()
```

```
// Запросить у пользователя заключение и занести его в протокол  
RTS_REQUEST ()
```

```
// Запросить у пользователя текстовый комментарий и занести его в протокол  
RTS_EXPLAIN ()
```

Пользователь
выполняет все
действия через
интерфейс СОВУ

Тестовый случай

Тестовый случай – фрагмент теста, проверяющий элементарную функцию тестируемого устройства.

Тестовый случай может проверять правильность приема/передачи отдельного сообщения или слова, или отдельного битового поля в составе сообщения.

Типичная структура тестового случая:

- *Установка тестовых условий (входных значений)*
- *Ожидание результата*
- *Проверка ожидаемого результата (+ нумерация)*
 - Операторы **if** и **RTS_WHEN**

Тело:

```
RTS_TESTNAME ("<name>"),  
RTS_TESTNUM (<multi-level number>)           // автоинкремент
```

Оператор `RTS_WHEN`

```
RTS_WHEN ( "W46A185_13 == 0" , 15) { // ожидание 15 мс
    RTS_THEN():
        RTS_TEST_PASSED();
        break;
    RTS_OTHERWISE():
        RTS_COMMENT (
            "Ошибка: нарушено условие (W46A185_13 == 0)"
        );
        RTS_TEST_FAILED();
        break;
}
```

Пример тестового случая

```
// StateValue – битовое поле в RTS_OUTPUT_TERM_MESSAGE()
// ControlWord – битовое поле в RTS_INPUT_TERM_MESSAGE ()
// -----
RTS_TESTNAME( “Су35.СОЛС.Наземный_контроль.17”);

// Установка значения отправляемых данных
Control = 1;
// Ожидание выдачи слова
RTS_DELAY( 0.5 );
// Проверка ответа от тестируемого устройства
RTS_WHEN ( “State == 5”, 0.2) {
    //.....
}
```

Протокол тестирования

- Информация о смене (назначении) номеров тестов
- Информация о смене выполняемых тестов
- Результаты тестирования: «Тест прошел» / «Тест не прошел» с указанием номера теста
- Результаты всех команд взаимодействия с пользователем
- Комментарии с описаниями ожидаемого поведения теста и другая дополнительная информация, выдаваемая командой `RTS_COMMENT`
- Информация о текущих значениях объектов ФТ (в т.ч. битовых полей и битов), выдаваемая командой `RTS_DISPLAY`

Работа с протоколами тестирования

Заголовок

```
RTS_LOG_FILE(LOG1, "log1.log")  
// Файл «log1.log» будет помещен в подкаталог  
// «ПротоколыТестирования» каталога  
// результатов эксперимента
```

Тело

```
RTS_LOG(LOG1)  
RTS_CLEAR_LOG(LOG1)
```

Протоколы тестирования служат основой для формирования отчетов по тестированию.

Пример протокола тестирования

- * Протокол функционального тестирования LOG1
- * Собран компонентом tsk (Stand.tsk)
- * Тип компонента: TCK_Stand::TCK::TCK
- * Имя эксперимента: T1
- * Дата эксперимента: Втр Май 30 09:46:25 MSD 2006
- * Продолжительность эксперимента: 30с 000мс
- * Эксперимент запущен пользователем: mike

>>>> [0мс] от начала эксперимента

| Начало функционального тестирования БЦВМ из проекта DemoLang

Тест номер 3.2.1

>>>> [9с 000мс] от начала эксперимента

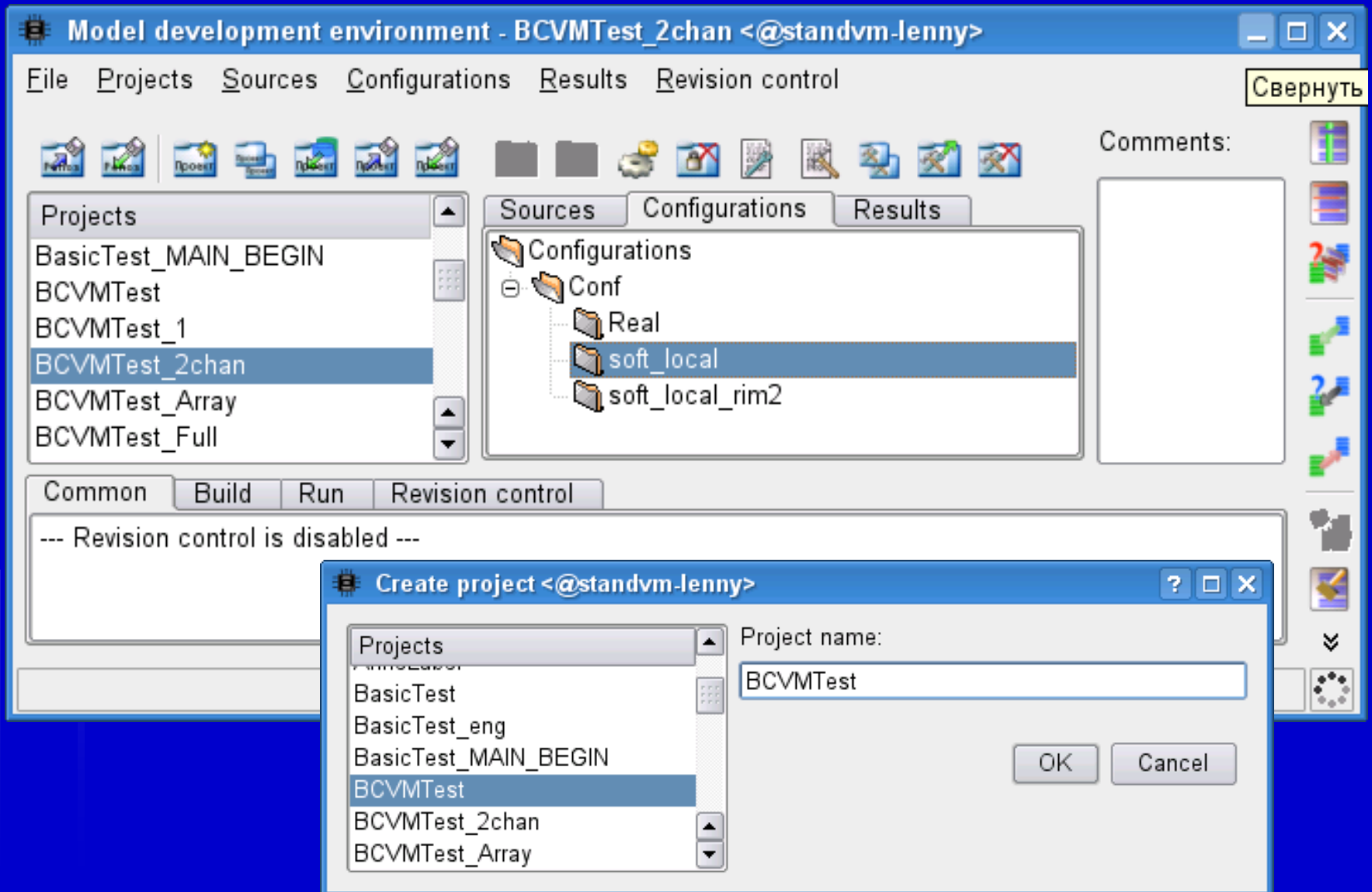
Тест номер 3.2.1 прошёл

Тест номер 3.2.2

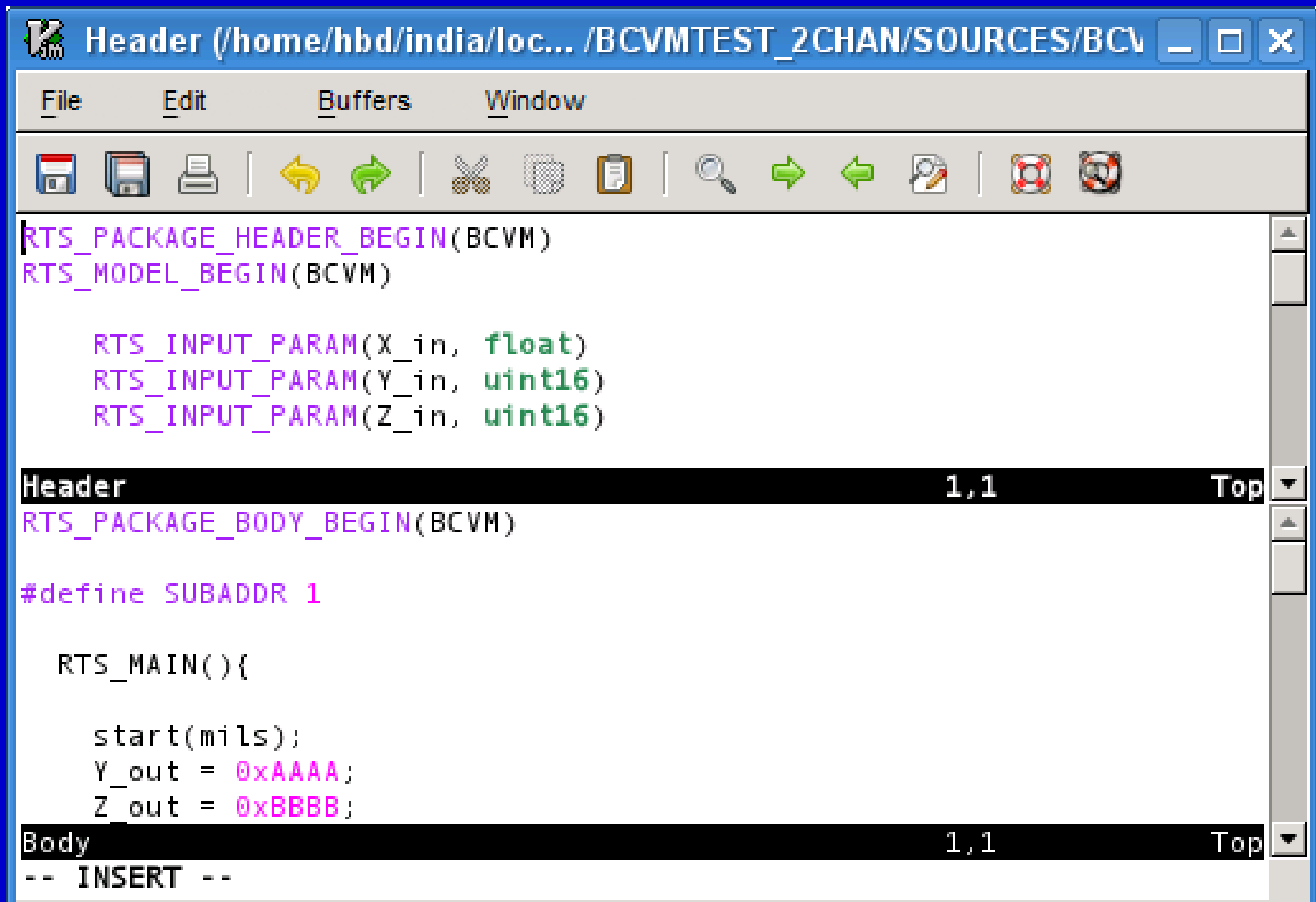
...

Подготовка и выполнение тестирования

Создание проекта



Редактирование кода ТСК



The image shows a screenshot of a code editor window titled "Header (/home/hbd/india/loc... /BCVMTEST_2CHAN/SOURCES/BCV)". The editor has a menu bar with "File", "Edit", "Buffers", and "Window". Below the menu bar is a toolbar with various icons for file operations and editing. The main text area contains C code for a package header and body. The code is color-coded: keywords in purple, identifiers in black, and literals in green or pink. The code is as follows:

```
RTS_PACKAGE_HEADER_BEGIN(BCVM)
RTS_MODEL_BEGIN(BCVM)

    RTS_INPUT_PARAM(X_in, float)
    RTS_INPUT_PARAM(Y_in, uint16)
    RTS_INPUT_PARAM(Z_in, uint16)

Header 1,1 Top
RTS_PACKAGE_BODY_BEGIN(BCVM)

#define SUBADDR 1

    RTS_MAIN(){

        start(mils);
        Y_out = 0xAAAA;
        Z_out = 0BBBB;

Body 1,1 Top
-- INSERT --
```

Привязка ТСК к инструментальным машинам

The screenshot shows a software window titled "/DemoLang_with_TCK_example/Configurations/TCK_en/Exp ~". The window contains a menu bar with "File" and "Filter", a toolbar with various icons, and a tabbed interface with tabs for "Models", "MIL STD-1553B Models", "Parameters", "Interfaces", and "FT Objects".

The main area displays a table of component models with columns for "Component models", "Type", "Binding", and "Tracing". The "tsk" component is selected, and its binding dropdown menu is open, showing options: "SIM PC", "Hardware", "Off", "SIM PC #0", and "SIM PC #1".

Component models	Type	Binding	Tracing
Stand		*	*
bcvm	BCS::Baget83::Baget83	SIM PC #0	All
tsk	TCK_Stand::TCK::TCK	SIM PC	All

The binding dropdown menu for the selected "tsk" component is open, showing the following options:

- SIM PC
- Hardware
- Off
- SIM PC #0
- SIM PC #1

Привязка интерфейсов к адаптерам КБИ

/DemoLang/Configurations/Conf_en/Local ~ Configuration settings editor. Version 1.0.0.0

File Filter

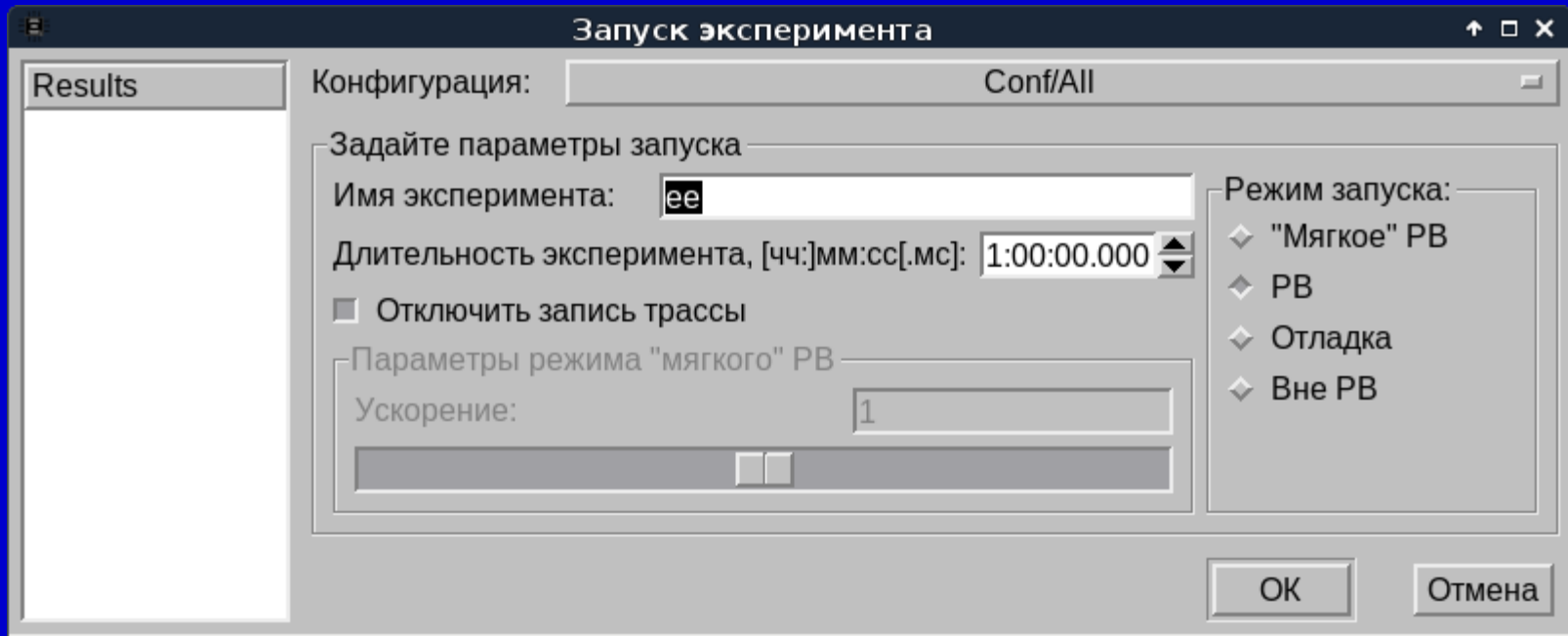
Models MIL STD-1553B Models ARINC 429 Models Parameters Interfaces DBDF

Inter-model MIL STD-1553B ARINC 429 interfaces

Models/Interfaces	Tracing	RT address	Controller	Binding	Adapter	Connects To	Alias	De
Stand	*			*				
bcs	*			*				
bcvm	*			*				
mils	All	0	True	With model	-- not set --	Stand.bcs.mils.io		
rv	*			*				
mils	All	18	False	With model	-- not set --	Stand.bcs.mils.io		
skv	*			*				
mils	All	17	False	With model	-- not set --	Stand.bcs.mils.io		

Stand:bcs::skv::milsAll

Запуск эксперимента



Оперативное управление экспериментом

Средство оперативного отображения и управления экспериментом

Окно Команды Справка

Дерево состояний

Имя	Состояние	Включен
Stand		
sender1	Фикти...	
sender1::out	Включен	<input checked="" type="checkbox"/>
sender2	Фикти...	
sender2::out	Включен	<input checked="" type="checkbox"/>
sender3	Фикти...	
sender3::out	Включен	<input checked="" type="checkbox"/>
sender4	Фикти...	
sender4::out	Включен	<input checked="" type="checkbox"/>
sender5	Фикти...	
sender5::out	Включен	<input checked="" type="checkbox"/>
sender6	Фикти...	
sender6::out	Включен	<input checked="" type="checkbox"/>
sender7	Фикти...	
sender7::out	Включен	<input checked="" type="checkbox"/>
sender8	Фикти...	

Параметры: По умолчанию

Модель	Сообщение	Имя	Значение
im1_term	IN	rxbf	0
im2_bcv	FromTerm	<Слова>	0:0x0000 1:0x00...
im2_bcv	FromTerm	FromTerm_rxbf	0
im2_bcv	ToTerm	<Слова>	0:0x0009 1:0x00...
im2_bcv	ToTerm	ToTerm_txbf	9
im2_term	OUT	<Слова>	0:0x0009 1:0x00...
im2_term	OUT	txbf	9
im2_term	IN	<Слова>	0:0x0000 1:0x00...
im2_term	IN	rxbf	0
im3_bcv	FromTerm	<Слова>	0:0x0000 1:0x00...
im3_bcv	FromTerm	FromTerm_rxbf	0
im3_bcv	ToTerm	<Слова>	0:0x0009 1:0x00...
im3_bcv	ToTerm	ToTerm_txbf	9

Состояние эксперимента: Ожидание соединения с ИМ ЧМ
Состояние эксперимента: Ожидание от пользователя команды начала эксперимента
Отправлена команда начать эксперимент. Длительность: 3600000 мс.
Состояние эксперимента: Эксперимент выполняется

Тестирование: fc_asm_im6_f3

Задержка (с): Внеочередной тест:

Тестовый компонент: Stand.fc_asm_im6_f3 Состояние: Тестирование остановлено

>>>> [0мс] от начала эксперимента
Тест "1"

Состояние эксперимента: Эксперимент остановлен пользователем | Модельное время: 0:00:09 | Расхождение: 0 мкс

Инструменты оперативного управления тестированием

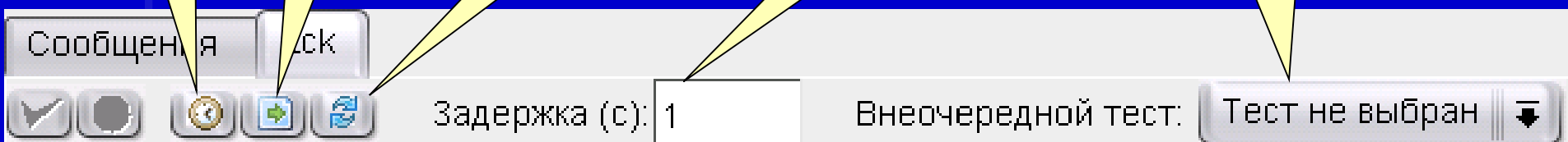
Ожидание ввода данных

Ожидание выбора теста

Циклическое выполнение теста

Задержка циклического выполнения

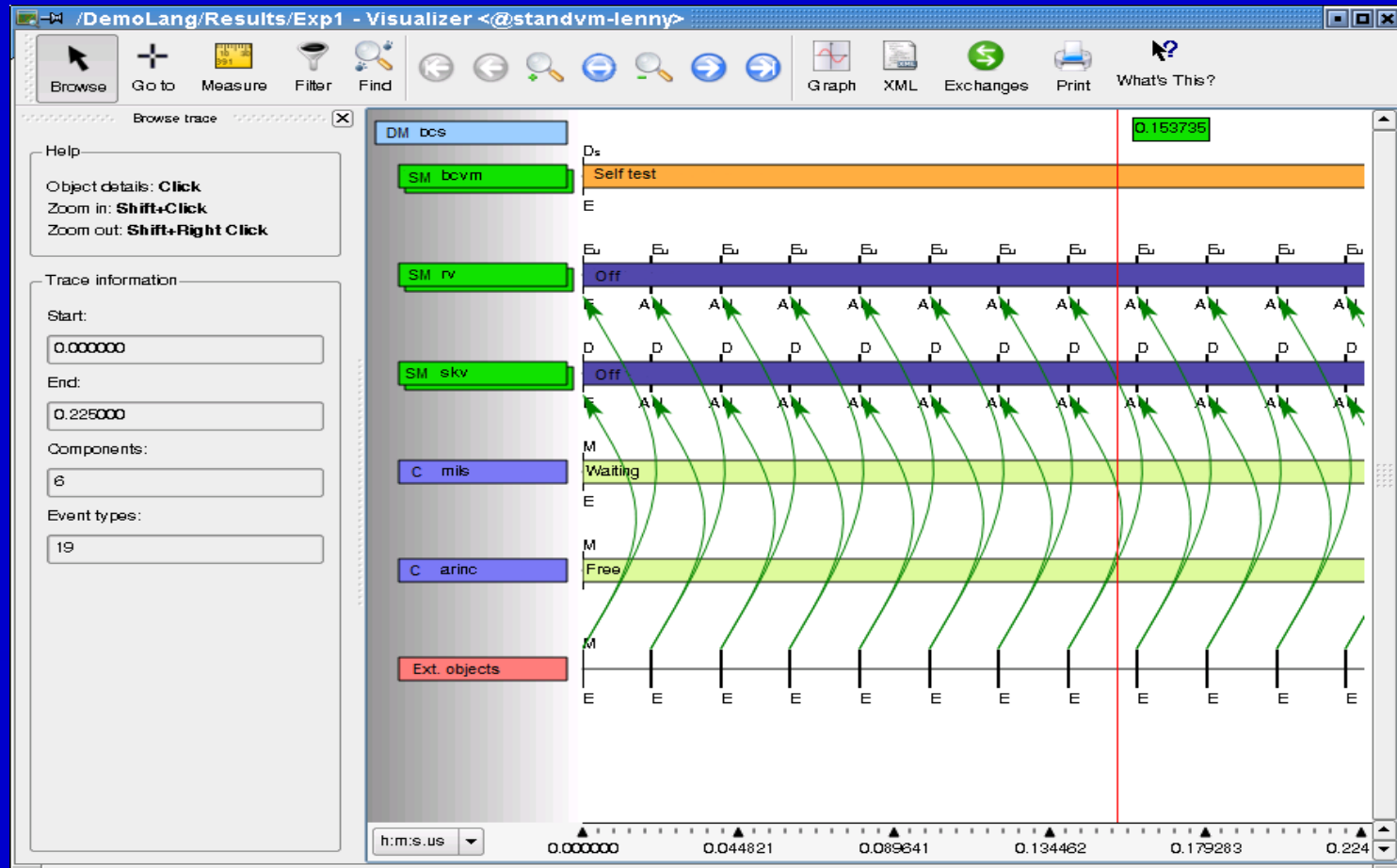
Выбор внеочередного теста



Тестовый компонент: Stand.tck Состояние: Тестирование остановлено

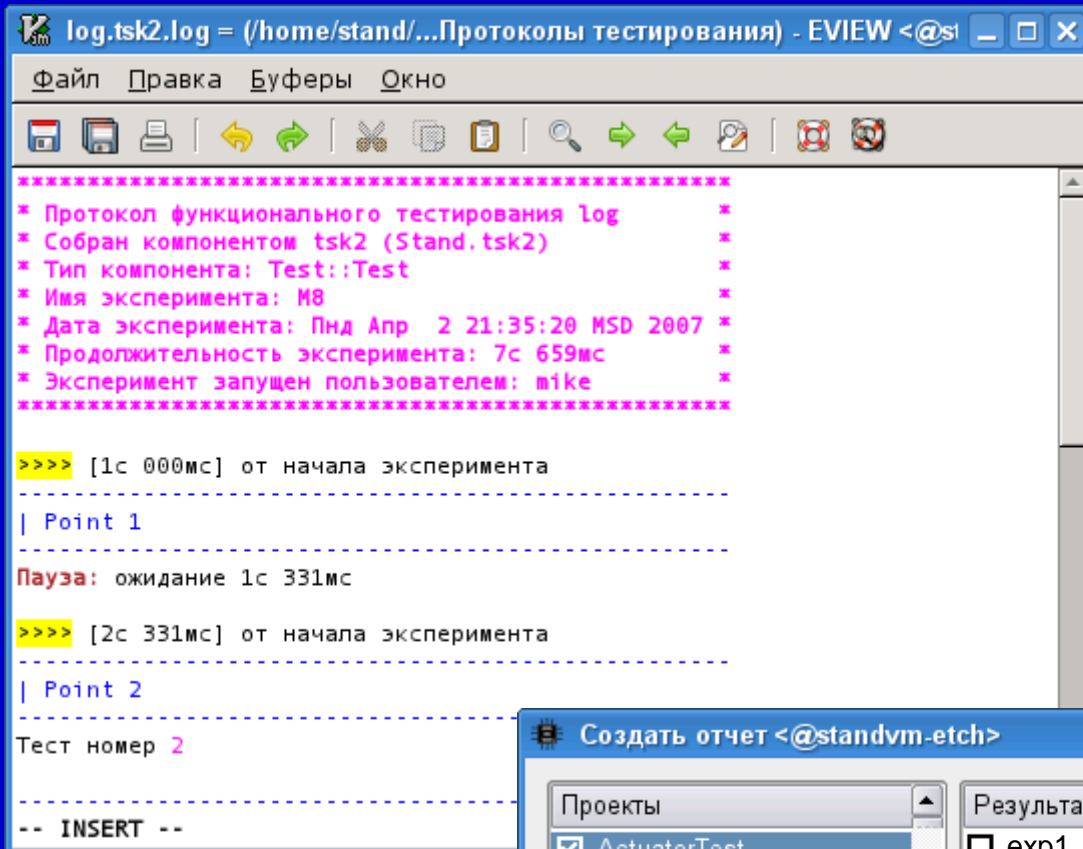
```
>>>> [0мс] от начала эксперимента
Тест "1"
Включён файл протокола тестирования LOG ("Log.log")
>>>> [3с 000мс] от начала эксперимента
Тест "ИУС.А.1"
>>>> [5с 000мс] от начала эксперимента
Тест "ИУС.А.1" прошёл
>>>> [6с 000мс] от начала эксперимента
Тест "ИУС.В.2"
```

Обработка результатов эксперимента



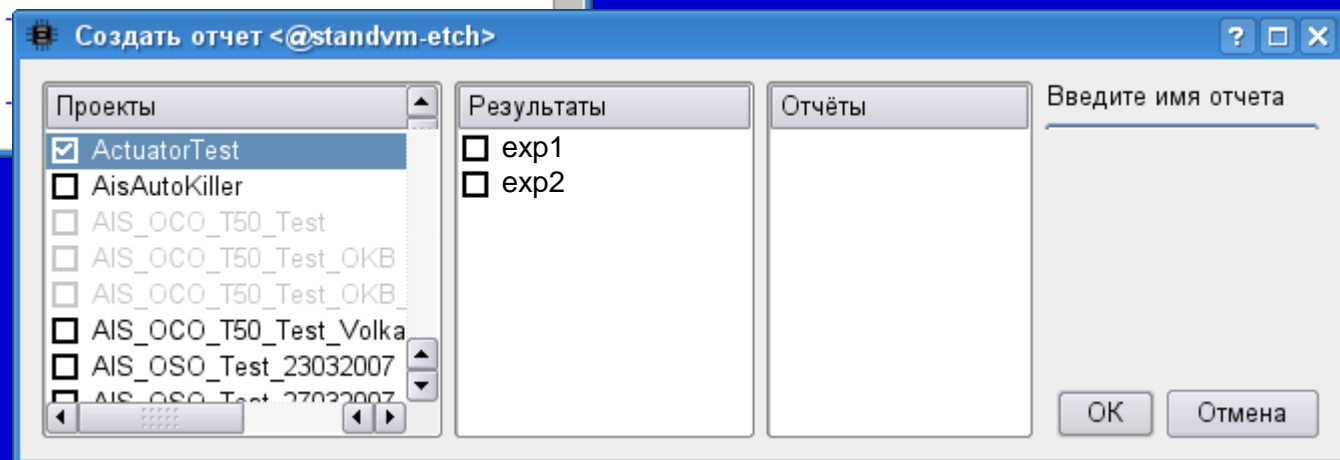
- Формирование отчётов по результатам тестирования
- Анализ временной диаграммы событий эксперимента

Отчет о тестировании



```
log.tsk2.log = (/home/stand/...Протоколы тестирования) - EVIEW <@si _ □ ×
Файл Правка Буферы Окно
*****
* Протокол функционального тестирования log *
* Собран компонентом tsk2 (Stand.tsk2) *
* Тип компонента: Test::Test *
* Имя эксперимента: M8 *
* Дата эксперимента: Пнд Apr  2 21:35:20 MSD 2007 *
* Продолжительность эксперимента: 7с 659мс *
* Эксперимент запущен пользователем: mike *
*****
>>>> [1с 000мс] от начала эксперимента
-----
| Point 1
-----
Пауза: ожидание 1с 331мс
>>>> [2с 331мс] от начала эксперимента
-----
| Point 2
-----
Тест номер 2
-----
-- INSERT --
```

- Отчет – таблица (.csv) с указанием статуса завершения тестов в ходе экспериментов
- Также формируется матрица прослеживаемости требований на тесты



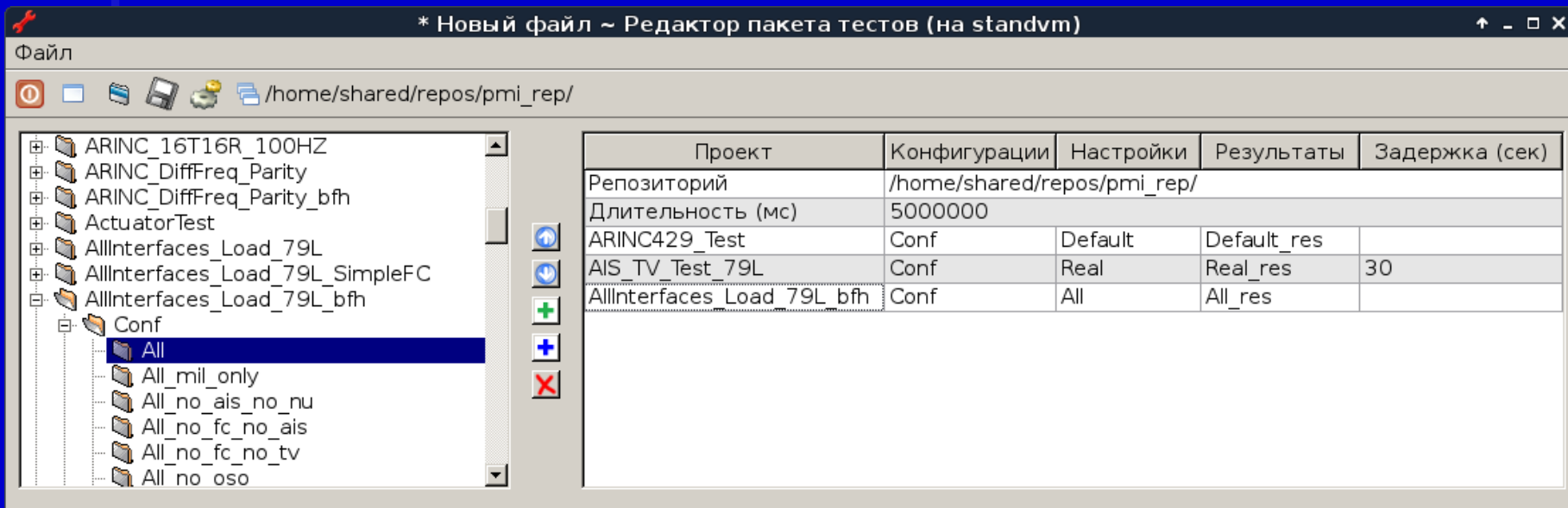
Создать отчет <@standvm-etich>

Проекты	Результаты	Отчёты	Введите имя отчета
<input checked="" type="checkbox"/> ActuatorTest	<input type="checkbox"/> exp1		
<input type="checkbox"/> AisAutoKiller	<input type="checkbox"/> exp2		
<input type="checkbox"/> AIS_OCO_T50_Test			
<input type="checkbox"/> AIS_OCO_T50_Test_OKB			
<input type="checkbox"/> AIS_OCO_T50_Test_OKB_			
<input type="checkbox"/> AIS_OCO_T50_Test_Volka			
<input type="checkbox"/> AIS_OSO_Test_23032007			
<input type="checkbox"/> AIS_OSO_Test_27032007			

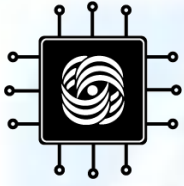
OK Отмена

Пакетный режим выполнения тестов

- Предназначен для прогона полностью автоматических тестов
- Пакет – набор записей, в каждой из которой задано:
 - имя проекта, конфигурации, набора настроек, набора результатов
 - задержка после завершения эксперимента (опционально)
- Результат пакетного запуска: сводный отчёт по всем экспериментам



Проект	Конфигурации	Настройки	Результаты	Задержка (сек)
Репозиторий	/home/shared/repos/pmi_rep/			
Длительность (мс)	5000000			
ARINC429_Test	Conf	Default	Default_res	
AIS_TV_Test_79L	Conf	Real	Real_res	30
AllInterfaces_Load_79L_bfh	Conf	All	All_res	



Спасибо за внимание!