MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT of COMPUTER ENGINEERING

CENG 492-COMPUTER ENGINEERING DESIGN 2

'BluePost'

TEST SPECIFICATIONS REPORT

by

Duygu CEYLAN – e1394782 **Seda ÇAKIROĞLU - e1394816**

Ertay KAYA – e1356948 **Hüseyin ÖĞÜNÇLÜ - e1395318**

Gözde ÖZBAL – e1395326

TABLE OF CONTENTS:

1. INTRODUCTION

1.1. Goals and Objectives

The 'BluePost' project has several modules including File Uploading, Format Conversion,Sending Data to the Board via Bluetooth, Register Process, Retrieving Data from User via Bluetooth Evaluation Kit, Sending Data to User via Bluetooth Evaluation Kit, and VGA Process. As AlınYazılım members, we are assured that processing and duties of each module should be verified so that all of these modules work in harmony and correctly when they are integrated. As a result, we give great importance to testing our project in order to obtain a bug-free, and logically correct product with high performance.

1.2. Scope of Document

The purpose of this document is to describe the testing process of the project 'BluePost'. In fact, while developing our project up to now, we did testing for each module, so we will explain about the testing process that took place since the beginning of the project and we will also state the future plans about testing process.

1.3. Statement of Testing Plan Scope

Testing process of the project “BluePost” includes unit testing, integration testing, validation testing, performance testing, stress testing, alpha and beta testing.

Unit testing: For the modules File Uploading, Format Conversion, Sending Data to the Board via Bluetooth Module, Serial To Parellel Conversion, and VGA Process Module

Integration testing: We are assured that this is the most important part because there are a lot of modules modules in our project and their harmony is very critical.

Validation testing: In two categories: requirement validation and design validation.

Performance testing: Includes testing of methods used to improve the speed and the quality of the project.

Stress testing: In order to test the control of each data for our project

Alpha testing: Testing of BluePost by someone other than the group members, working on the same project.

at the environment that the game is developed with all the required conditions for the game(development environment).

Beta testing: Testing of BluePost by many people(customer side) at different environments(customer environment).

1.4. Major Constraints

1.4.1. Time

Especially testing the parts of our project written with VHDL is time consuming and we have to form and then test the modules written in VHDL in order to finish our project. And we have nearly one month to complete the project BluePost, including removing the bugs found and improving the project. As a result of this, time is the greatest constraint for testing of the project “InterDimension”.

1.4.2. Usefulness of Obtained Test Results/Cost Ratio

The amount of usefulness of test results per the amount of time it takes and per the amount of human resource requirement is one of the most important factors limiting the testing of the project. Because giving too much importance to test some modules can be unnecessary when we consider the test results/cost ratio since working on testing process requires both time and labor.

1.4.3. Hardware

Especially in beta testing part, we should provide each tester the necessary hardware to test our product completely. And since some parts of the hardware are unique, we are not able to make the beta testers test our project at the same time.

1.4.4. Staff

There are five members in AlınYazılım and since the group will also be working on debugging and development at the same time, the staff is also a major constraint for the testing phase.

1.4.5. Platform dependency

Our project is platform dependent, in fact works in only Window operating system because of the fact that our project uses the stack initialization process of Windows. We are assured that this will be also a major constraint for the testers who do not use Windows operating system.

2. TESTING PLANS AND STRATEGIES

2.1. UNIT TESTING

While developing our project by proceeding module by module, we are making sure that every module is processing correctly so that when these modules are combined, we can be sure that there are no problems in the basement. We believe that the modules can work in harmony more easily by proceeding in this bottom up manner. All the modules are tested in a black box manner, by observing the input that is provided to the module and the output or the effect produced as a result. If a problem occurs during black box testing and it can be handled, we are also planning to make use of white box testing.

2.1.1. Testing File Uploading Module

In order to test the File Uploading Module we developed some test scenarios according to the normal process flow of the module as follows. These test scenarios cover all the possible flows. If any one of these scenarios does not terminate successfully, we will debug our code by placing break points in the functions of the related Java class and correct the errors accordingly.

Scenario 1

1) ADD image tab will be selected.
2) Attach button will be clicked
3) An image file will be selected.
4) OK button will be clicked on the file browser window.
5) Slide number will be selected from the combobox.
6) Time duration will be entered.
7) ADD button will be clicked.
8) “The image was successfully added” message is expected if an image for the slide number chosen has not been specified previously. If specified, “Do you want to change the image for slide X” message is expected. If CHANGE button is clicked, the image for the specified slide is changed and ”Image is successfully changed” mesaage is expected.
9) If “CANCEL” button is clicked, the operation is cancelled.

Scenario 2

1) SAVE button will be clicked either from the toolbar or the menubar.
2) A window showing the current BluePost projects in a tree view manner will appear.
3) User will provide a name for the project.
4) If a project with the same name does not exist, the current folder will be renamed to the specified name.
5) If a project with the same name exists, user will be asked whether (s)he wants to change the project with the specified name.
6) If the answer is YES, the project with the specified name is changed with the current project.
7) If the answer is NO, the user is prompted to enter a new name.

Scenario 3

1) OPEN button will be clicked either from the menubar or the toolbar.
2) If changes to the current project that have not been saved yet exist, the user will be asked whether he wants to save the current project.
3) If YES button is clicked, the current project will be saved.And the scenario 2 will be executed as if the SAVE button was clicked.
4) If NO button is clicked, the changes to the current project are discarded.
5) A file browser window showing the BluePost projects, appears.
6) A project is selected from the tree view.
7) OK button is clicked.
8) The current directory changes to the directory of the selected project.

Scenario 4

1) Images tab is selected.
2) Images selected from the current project are shown as thumbnails.
3) The radiobutton for one of the images is selected and the properties(slide number and time duration) of that image are seen.
4) Necessary changes are made to the properties of the image.
5) SAVE button is clicked.

6) If the new slide number provided does not exist in the project, the changes are saved. Otherwise, the user is asked whether he wants to change the image for the slide number or not.
7) If YES button is clicked, the image for the slide number is changed.
8) If NO button is clicked, the changes are discarded.

Scenario 5

1) SEND button is selected from the toolbar or the menubar.
2) The password is entered. START SENDING button is clicked.
3) The progress of the operation is observed by the progress bar.
4) When the hex file is formed, bluetooth operations begin.
5) If an error occurs during the bluetooth operations(no device or service found) an error message is displayed.
6) If no error occurs, a message indicating the situation is viewed.

Scenario 6

1) CHANGE PASSWORD button is selected from the toolbar or the menubar.
2) First the old password, then the new password is entered.
3) If the old password matches with the current password, the two new passwords entered are the same, and the new password contains at least six characters, “The password is successfully changed” message is viewed.
4) If one of the above conditions is missing, an appropriate error message is viewed and the change password window reappears.

Scenario 7

1) The CLOSE button is clicked.
2) If there are changes to the current project that have not been saved yet, scenario 2 will be executed as if the SAVE button was clicked.
3) The program exits.

2.1.2. Testing Sending Data to the Board via Bluetooth Module

While implementing this module we used a java archive package bluecove.jar that does the all the necessary operations (stack initialization, service discovery etc.) for this module. Thus we did not develop any test scenario for this module. We just did some tests about the integration and useage of this archive by other components while testing the File Uploading Module.

2.1.3. Testing Format Conversion Module

We developed the following scenario for testing this module that checkes the correctness of the newly formed image. If the scenario fails we will debug the module code and detect the errors accordingly.

Scenario 1

1) The image which will be resized, is displayed on a panel before this module operates.
2) After the operation is completed, the same image is displayed with the new dimensions.
3) The properties of the image checked.
4) When this module is integrated with the file upload module, and the images selected are formatted accordingly, the images are displayed on the monitor without any loss of pixels.

2.1.4. Testing Register Process Module

While implementing this module, we used a VHDL code written before by XESS designers. When we ran this code it worked without any problems. Thus, we did not develop any test scenarios for this module.

2.1.5. Testing VGA Process Module

VGA Process Module is tested according to the following scenario. After applying the scenario we will check the display images whether the pixels are located correctly and duration of each image is the same as set by file upload module. If the test fails we will debug the VHDL code of the module and detect errors by using ModelSim.

Scenario 1

1) Images are selected by the file upload module
2) The .hex file formed by the format conversion module is transformed to the XSA board.
3) Images are displayed according to the order and time constraints specified by the user.

2.1.6. Testing Serial-Parallel Converter Module

Our Serial-Parallel Converter Module mainly consists of two parts: A board with serial, parallel ports and a microchip that makes the conversion in both ways and the software loaded onto the microchip. Testing strategies for the board are described in the “Hardware Testing” section. In order to test the software that is loaded on the microchip we use the simulation tool *Proteus ISIS.* The test procedure is conducted as follows: Using the ISIS’ hardware simulators we simulate the complete hardware that we designed. In addition we use an LCD monitor simulator that is not a part of the hardware we designed to see the testing results. After simulating the hardware we use this simulation and load the software onto it. We first give an 8 bit data to the serial port input and observe the output as an ascii character in the LCD connected to the parallel port output. The same procedure is followed while sending data from parallel port to serial port. If the character output on the LCD monitors are different from the characters we expect to see then we check the contents of the microchip registers and ports to observe at which step of execution the error occured. We first check some determined key registers whose values are the results of some main code segments. After finding the erronous key register, we debug the code segments that is related to this key register by checking the intermediate values stored on the other registers that used only for storing these intermediate values. We repeat the testing operation after correcting the errors untill we make sure that the conversion procedures are implemented without any error.

2.2. INTEGRATION TESTING

As all the project members, we are assured, the fact that all the modules work correctly and in harmony is really important for our project. So, as we proceed by completing some modules, we immediately combine them and start integration testing process. In this manner, that is by using bottom up modeling and being sure that the component modules are correct, we believe we won't have to deal with the errors in the basement while proceeding.

1	File Uploading Module	Format Conversion Module	Format Conversion uses the data formed by the file uploading module.
2	File Uploading Module	Sending Data to the Board via Bluetooth Module	Second module uses the data of the first module to send.
3	Format Conversion Module	Sending Data to the Board via Bluetooth Module	Second module uses the data of the first module to send.
4	Retrieving Data from User via Bluetooth Evaluation Kit Module	Sending Data to the Board via Bluetooth Module	Both modules use the same harware, so they are are in interaction with each other.
5	Register Process Module	Retrieving Data from User via Bluetooth Evaluation Kit Module	First module takes the information to be kept in SDRAM from the second module
6	VGA Process Module	Register Process Module	The data which was saved to the SDRAM by the Register Process Module is used by the VGA Process Module
7	Sending Data to the Board via Bluetooth Module	Retrieving Data from User via Bluetooth Evaluation Kit Module	The data prepared by the first module is sent by means of the second module from the user to the board

The integration testing method can be white box testing or black box testing according to the specific module and its functionalities. The tester is free to choose between these two methods by considering the module properties.

Scenario 1

1) A project consisting of the images selected by the user is formed by means of the File Uploading Module.
2) The user clicks the SEND button.
3) The formats of the images are converted to the bit streams that contains the RGB values of the pixels.
4) The new formatted images and the text message are sent to the board by means of the Sending Data to the Board via Bluetooth Module and Retrieving Data from User via Bluetooth Evaluation Kit Module.
5) The received data is stored in SDRAM with the help of the Register Process Module.

6) VGA Process Module reads the data stored in SDRAM.
7) The text messages stored in SDRAM are sent to the user via Bluetooth Evaluation Kit Module.

2.3. VALIDATION TESTING

Validation testing is a system test that validates the conformance of the implementation with the requirements analysis and design. In order to understand this conformance after the implementation phase, we will use black box testing and see any incorformance occurs. We have divided validation testing process into two groups:

2.3.1. Requirements Validation

The tests for validation with respect to the requirements will be black box tests. While doing these tests we consult our requirement analysis report that we have generated according to the customers' expectations and requirements. In our report, we determined the requirements of the project as follows:

- A user will be able to upload upto 10 images in .jpeg and .gif formats.
- A user will be able to assign any time length for each image seperately.
- A user will be able to save and update the images in a slide show.
- A user will be able to send event information as a .txt file to the bluetooth device users.

During the requirements validation, we will be testing whether all of the above functionalities are provided by 'BluePost'.

2.3.2. Design Validation

During the implementation of our project, we try to satisfy the design issues stated in the detailed design report in terms of the class and hardware circuit designs. By this way, we aim to provide the conformance of them with the implemented components. We observed that there are some changes to these components due to the implementation constraints while applying the design validation tests. We use white box testing method during these processes because of the fact that we want to observe and compare all the implemented classes and circuits with the ones we stated in the detailed design report.

2.4. HARDWARE TESTING

‘BluePost’ Project contains 3 main hardware boards(XSA 3S1000, BlueRadios Evaluation Kit and Serial-Parallel Converter Board) one of which (Serial-Parallel Converter Board) is designed by the project team. Since XSA 3S1000 and BlueRadios Evaluation Kit are commercial products that are used widely throughout the world we did not need to develop any strategy to test the working of these boards.

We developed a simple testing strategy for the Serial-Parallel Converter Board that we designed and it is as follows: We will do the testing by checking the communication between serial and parallel port pins one by one. For example, to check the 4th data bit of the parallel port interface we will send an 8 bit data from serial port whose 4th bit is 1 and all the rest is 0. Then we will check if the output from parallel port is the same as the input of serial port, that is, we check if the 4th data pin of parallel port conducts the bits correctly. While doing these tests we will use a PC that has both parallel and serial ports and will install 2 programs, Portmon and CCS Serial Input/Output Monitor that monitors the data that comes to parallel and serial ports respectively.

For the rest of the Serial-Parallel Converter Board we did not develop any testing strategy due to the insufficieny of our knowledge of electricity. If an error occurs while testing and/or running the board we will get help of experts in order to fix the problem.

2.5. HIGHER ORDER TESTING

Since ‘BluePost’ is a commercial project that will be used by many people in a wide area Higher Order Testing has also great importance during the test phases. Thus, after making sure that all the tests described above have been passed successfully, we will also do some higher order tests for a complete success of our project as described below:

2.5.1. Performance Testing

In our project, time limitations should be considered as a main issue. However, most of these limitations are caused by the hardware used in the project. The data rate of BlueRadios Kit and parallel port of XSA3S1000 board are constant so we have nothing to do with them. Thus, what we consider is the time required to create the .hex file that contains image data and time

limitations of the parallel-serial converter board we designed. To test the performance of the .hex file formation, we will use our software and upload 10 different high resolution images and observe the time required for converting these image files to 800*600 pixel*pixel images and writing these images to a .hex file. In addition we will upload this big .hex file to the SDRAM of the XSA 3S1000 board by using our Serial-Parallel converter board and be able to observe the maximum data rate of our board.

2.5.2. Stress Testing

As stress testing, we will test the performance of the BlueRadios Evaluation Kit by locating as much bluetooth device as we can find around it and see what happens while Kit tries to connect and send event information to all these devices.

2.5.3. Alpha and Beta Testing

After completing the previous tests we will also do some alpha tests. First we will do some black box tests with the project group I2TECH and check all the components of the project. If any problem occurs during black box tests, we will also do some white box tests and find the errors that cause the problem. After the alpha tests are completed successfully, we will also do Beta Tests according to the following scenario: After a lecture in our department, we will use the projectors in the classrooms and run our project. Then we will make a student upload the images and send the corresponding event info to the bluetooth devices owned by the other students in the classroom. Afterwards, we will get feedback from the students and check whether any problem occured during the test process.

3. TEST RECORD KEEPING AND LOG

In ‘BluePost’ we develop both software and hardware designs in various environmets and languages. In order to be able to have a common record logging for these designs we use a table whose rows correspond to tests done and columns give information about the test results as follows:

Column 1; *Tested Module:* In this column the name of the tested module is recorded for each test.

Column 2; *Test Date:* In this column the date at which the test done is recorded for each test.

Column 3; *Test Result:* Either passed or failed. If the test is passed successfully the remaining parts are not filled.
Column 4; Problematic *class/entity/hardware component:* If the found bug is in a module that is implemented using Java then it's class is recorded. If it is in a module implemented by VHDL, it's entity is recorded and if there is a problem in a hardware component then its name is recorded.
Column 5; *Description of the Problem*: In this column, the found bug is described properly for each test.
Column 6; *Fix Date*: In this column, the date at which the found bug is fixed is recorded for each test.

4. TESTING TOOLS AND ENVIRONMENTS

- ✓ *Eclipse SDK*: For Java code developing and debugging
- ✓ *ISE Webpack*: For VHDL circuit design and simulation.
- ✓ *ModelSim*: For VHDL circuit design simulation and debugging.
- ✓ *Proteus ISIS*: For serial-parallel converter board simulation.
- ✓ Portmon: For parallel port data sending and monitoring.
- ✓ CSS: As C compiler and for getting .hex file from C code to be loaded to the parallel-serial converter board.
- ✓ CCS Serial Input/Output Monitor: For serial port data sending and monitoring.
- ✓ *Microsoft HyperTerminal*: For monitoring the data transfer via bluetooth.

5. TESTING RESOURCES AND STAFFING

The only resource required for testing is a PC in which the development environments, simulators and monitoring tools described in the other sections are installed. The AlınYazılım team is divided into 2 subteams for unit testing purposes as follows:

- ✓ *Duygu, Seda*: File Uploading, Format Conversion and VGA Process Modules
- ✓ *Ertay, Gözde, Hüseyin*: Other units.

Integration testing, validation testing and Higher Order testing will be done with contribution of the all team members.

6. TESTING SCHEDULE

The following table shows the schedule of testing steps for ‘BluePost’:

Test Plan Delivery: (Deadline) 06.05.2007

Unit Test and Integration Tests: (Start) 01.03.2007 - (Deadline) 20.05.2007

Validation Tests: (Start) 10.05.2007 - (Deadline) 25.05.2007

Performance and Stress Tests: (Start) 15.05.2007 - (Deadline) 30.05.2007

Beta and Alpha Tests: (Start) 20.05.2007 - (Deadline) 05.06.2007

Results (i.e. Bugs) Tracing and Correction: (Deadline) 08.06.2007