Progress Report Examples

The following examples are drawn from past student reports, and illustrate how the general guidelines can be applied to a variety of design projects. The technical details have been removed in order to highlight the report structure and organization as opposed to the technical content or writing style. These examples have **not** been properly formatted with the appropriate use of tables, bullets, etc. It is expected that you do not use these as templates, but make the progress report suit your project.

Note that the examples follow an older reporting style, where there was no group document.

Example #1: Controller for Frequency Modulated Spectroscopy

Executive Summary

This project involves building a system that modulates light in the 0.1Hz to 50kHz range and detects the reflected light through the patient's tissue at multiple distances. This will be accomplished by three main modules: …

To date, the … modules have been built as planned and are now awaiting … for integration testing. Particularly, my main contributions include….

However, the X Module, which was originally outside the scope of our current project, was delayed due…. Without this setup, the … cannot be fully tested for noise tolerance and …

As a result, and I must now shift our focus completely to the …. Several key tasks include…. We plan to complete all these tasks by…. Despite the current difficulties, we will strive to complete the design of a single detection channel to demonstrate the potential of the instrument through the characterization of both the signals and noise in ….

1. Group Progress Summary

1.1 Summary of Project Goal

This project involves building a …A system block diagram of the proposed technical solution for this device is shown in Figure 1.

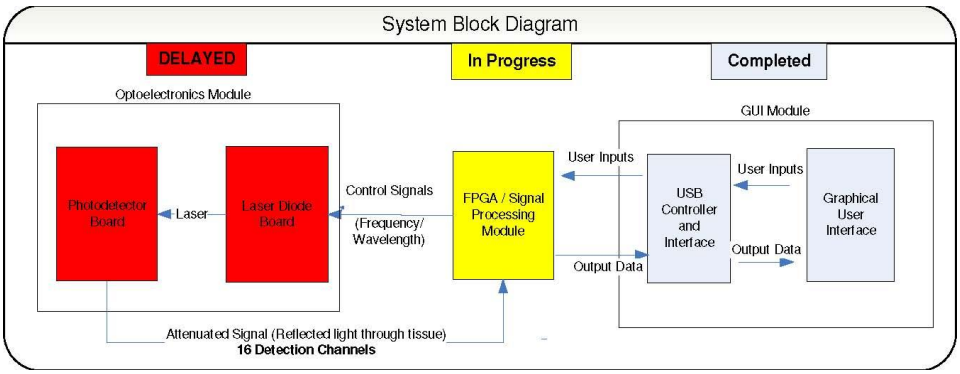


Figure 1: System block diagram of Frequency –Modulated Reflectance Spectroscopy Instrument

1.2 Summary of Group Progress

An overview of the progress was highlighted in Figure 1 (Blue: Completed, Yellow: In Progress, Red: Delayed). … the progress for each of the six stages is summarized below:

KEY STAGES	PROGRESS TO-DATE
STAGE 0: Prototype on DE2 Board	Completed
STAGE 1: Test Equipment Setup	Completed
STAGE 2: USB Microcontroller Setup	Completed
STAGE 3: FFT Model Refinement	In Progress
• Design peak extraction model	Completed implementation
• Simulate/Test with noise	Waiting for optoelectronics module
STAGE 4: GUI Design	Completed
STAGE 5: Custom PCB Design	In Progress
• Search Components and Spec.	Completed
• Component Library/Footprint Search	Completed
• Schematics Layout + Verification	Completed 3 revisions
• Simulate PCB+ Final Verification	Future Delays expected
• Integration Testing	N/A

Since the design review, the key responsibilities for _____ included: …

For _____, the key responsibilities included…

1.3 Summary of Changes

As a result of the departure of the research assistant, the optoelectronic module was not ready as planned and the PCB layout was also delayed. This has caused delays in Stage 3 … To resolve this issue, we have now taken on …

2. Individual Progress

2.1 Overview of Contributions

My main contributions to the project include…The following list provides an overview of my key contributions:

1) USB Microcontroller Setup and Testing (Appendix C)

2) Designed a refined FFT Model in DSP Builder (Appendix D)

3) Developed the GUI using Borland C++ Builder (Appendix E).

4) Verified PCB layout (Appendix F)

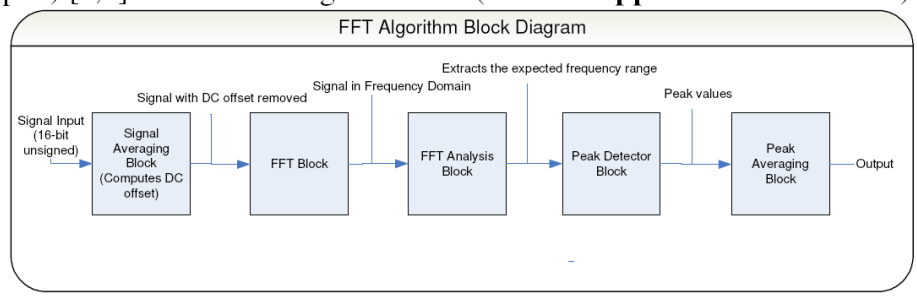
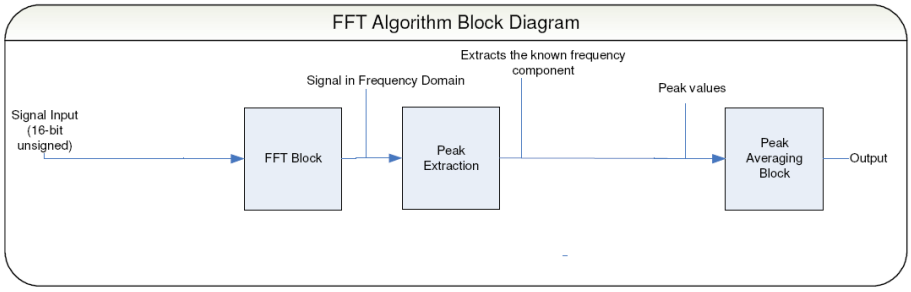
Table 2: Summary of Individual Milestones for the Current Reporting Period

Task #	Task Title	Category	Status	Old Completion Date	New Completion Date	Appendix
***STAGE 1: Test Equipment Setup**		Old	Completed	20-Nov-06	20-Nov-06	C
STAGE 2: USB Microcontroller Setup						
10	Implement USB protocol	Old	Completed	4-Dec-06	4-Dec-06	C
11	Design Basic GUI for testing	Old	Completed	11-Dec-06	11-Dec-06	C
STAGE 3: FFT Model Refinement						
12	Design peak extraction model	Old	Completed	4-Dec-06	4-Dec-06	D
STAGE 4: GUI Design						
14	Design Interface - Layout	Old	Completed	18-Dec-06	18-Dec-06	E
15	Link interface to USB layer	Old	Completed	1-Jan-07	1-Jan-07	E
STAGE 5: Custom PCB Design						
16	Search Components and Spec.	Old	Completed	18-Dec-06	18-Dec-06	F
18	Schematics Layout + Verification	Old	Delayed	29-Jan-07	1-Mar-07	F

2.2 Report on Individual Tasks

This section describes the details of the progress for each task listed previously….
(Discussion of Individual Tasks for Stage 2, 4 and 5 have been removed)

Milestone & Description	**STAGE 1: Test Equipment Setup**
Responsibility	
Status at start of reporting period	Started
Status at end of reporting period	Completed
Actions	In order to generate a 16-bit (digital) test signal, Kevin and I worked in parallel to setup the different components needed. To put this discussion in context, we used the Function Generator Card and 2 Data Acquisition Cards from National Instrument. An ADC and DAC are embedded inside the data acquisition cards. This was mainly s task. Meanwhile, I installed the *LabView* program on another computer to start designing the *LabView* test program. I searched for the documentation of the Data Acquisition Card and the *LabView 7* program. To learn how to setup the *LabView* Test Program (called a vi), I went through several examples for simple data acquisition and created an interface, which is later used to verify the setup.
Decisions	One data acquisition card only supports up to 8-bit parallel output or input, so from the start, we decided to use 2 data acquisition cards. However, the data acquisition speeds of these 2 cards are different and that caused a problem with the 16-bit output. In order to synchronize, the LabView test program had to include more complicated control blocks. Our final decision was to buy a new A/D Evaluation Board since the sine wave was unstable using the LabView setup. Therefore, the current setup does not use the National Instrument Data Acquisition cards, although we kept the Function Generator Card to generate the analog input. This current setup is fully operational.
Testing & Verification of Progress	**Appendix C**: To test the new A/D evaluation board (and function generator), sinusoidal waveforms (from the function generator) of varying amplitudes (0-2V) were sent in. Note that the USB microcontroller was integrated and tested together with the test equipment and only one waveform is shown to make the report more concise.

Milestone & Description	**STAGE 3: FFT Model Refinement**- *#12 Design peak extraction model*
Responsibility	
Status at start of reporting period	Not Started
Status at end of reporting period	Completed
Actions	The original FFT model has a problem with accuracy and it would fail if there was a huge peak (caused by noise) in the spectrum (FFT magnitude plot) [3,4]. Here is the original model (Refer to **Appendix D2** for details)  **Block diagram of original FFT-based algorithm.** The new FFT model extracts the known frequency component based on the input signal [5,6]. Here is the refined model (Refer to **Appendix D1** for details)  **Block diagram of refined FFT-based algorithm.**
Decisions	To refine the original model, the team had to decide among many alternatives. However, one alternative was especially attractive since it would simplify the design. This alternative relies on a key assumption– no frequency shift. That is, the original signal with known frequency will enter and exit the tissue with the same frequency despite light scattering and absorption. This assumption simplified the model tremendously.
Testing & Verification of Progress	**Appendix D3**: To test if this model works in simulation, a sine wave with a known frequency is set as the input signal. The output shows that the answer is almost the same as the expected amplitude. The slight discrepancy was found to be caused by the divider block inside the peak averaging block (which throws out the remainder and decimal place in the division). *****Note: The detailed simulation results will be reported by** 6

3. Progress Assessment

Our team has completed the GUI and signal processing modules on schedule…However, the departure of the research assistant has meant that we now need to shift our focus to the optoelectronics module… Future delays would be expected in the PCB layout due to this departure and the necessary shift in focus.

Despite the current difficulties, we are certain that we can complete the design of the single detection channel … This will allow us to demonstrate the potential of the instrument and understand the nature of both the biological signals and noise present in this system. If the PCB can be manufactured on time, we can also finish the 16-channel design to show the full potential of the instrument.

4. References

Appendix A:

Revised Gantt Chart

Activity #	Activity	Milestone?	Week: 11-Sep-06	18-Sep-06	25-Sep-06	2-Oct-06	9-Oct-06	16-Oct-06	23-Oct-06	30-Oct-06	6-Nov-06	13-Nov-06	20-Nov-06	27-Nov-06	4-Dec-06	11-Dec-06	18-Dec-06	25-Dec-06	1-Jan-07	8-Jan-07	15-Jan-07	22-Jan-07	29-Jan-07	5-Feb-07	12-Feb-07	19-Feb-07	26-Feb-07	5-Mar-07	12-Mar-07	19-Mar-07
	STAGE 0: Prototype on DE2 Board																												
1	Test Demos on DE2 Board		1	1	1																								
2	Setup DSP Development Software		2																										
3	Test FFT IP Megacore Library			2	2																								
4	Design RMS-Power Model					1	1	1	1																				
5	Design FFT Model					2	2	2	2																				
6a	Simulate and Compile FFT Model									1	1																		
6b	Simulate and Compile RMS Model									2	2																		
	STAGE 1: Test Equipment Setup																												
7	Setup Data Acquisition Card											2																	
8	Setup Function Generator Card												2																
9	Design Labview Test program (vi)											1	1																
	STAGE 2: USB Microcontroller Setup																												
10	Implement USB protocol													1	1														
11	Design Basic GUI for testing															1													
	STAGE 3: FFT Model Refinement																												
12	Design peak extraction model													2	2	2													
13	Simulate/Test with noise															2	2												
	STAGE 4: GUI Design																												
14	Design Interface - Layout															1	1												
15	Link interface to USB layer																1	1	1										
	STAGE 5: Custom PCB Design																												
16	Search Components and Spec.														1	1	1												
17	Component Library/Footprint Search																2	2	2										
18	Schematics Layout + Verification																							1,2	1,2				
19	Simulate PCB+ Final Verification																									1,2	1,2		
20	Integration Testing																											*1,2	1,2
	Stage 6: Optoelectronics Module																												
21	Review of previous work / future planning																			1,2	1,2	1,2							

Original Gantt Chart …

Appendix B – Project Goals and Requirements …

Appendix C: USB Microcontroller Setup

Using the GUI designed, the USB communication layer in the codes was tested. The correct setup of the USB microcontroller was verified by transmitting a sample sinusoidal input back to the PC. Note that this test also verifies the correct operation of the test equipment (function generator and the 16-bit A/D converter) and the interface hardware on the FPGA.

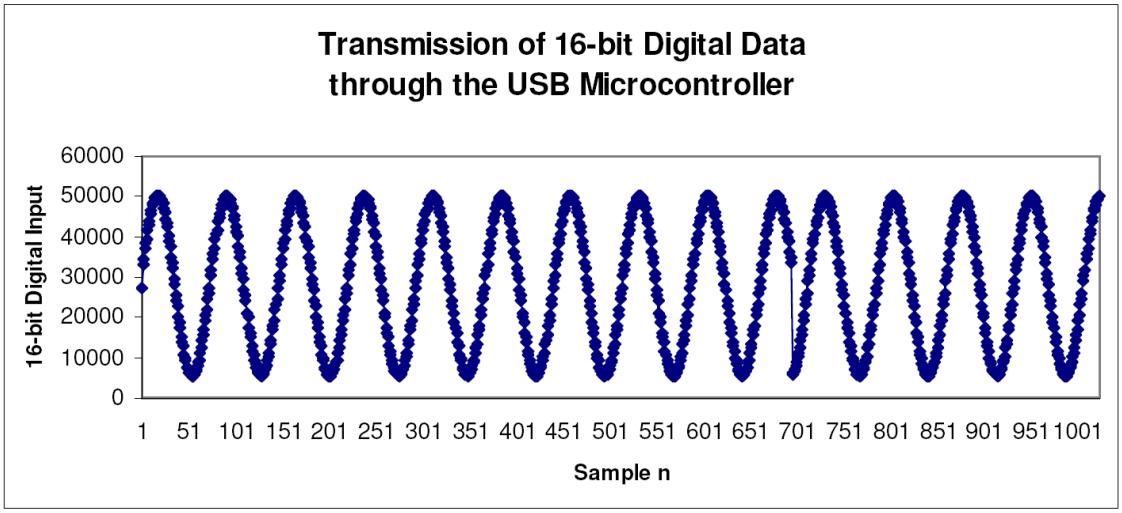


Figure C1: **USB Microcontroller Test Case.** This shows one thousand 16-bit samples received through the USB microcontroller taking 2000 transactions (1 byte/transaction). The digital data came from the analog signal generated by the National Instrument function generator card and the 16-bit ADC evaluation board.

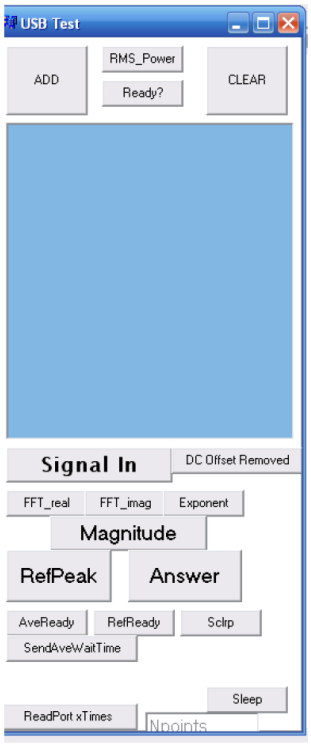


Figure C2: **Basic GUI for testing the USB microcontroller and the USB interface.**
The blue box in the center is where the digital data is displayed. The numbers are copied to an Excel application for plotting (Figure C1). Other buttons shown here are for probing the FFT model integrated with the USB hardware interface.

*The setup of the USB microcontroller actually involves a number of hardware layers, not shown here for clarity. Briefly, to get to this stage, not only does the test equipment have to work, but also all the hardware and software interfaces underneath.

For example, the Nios II processor is needed to communicate with the USB microcontroller through interrupt (Interrupt Service Routines). This Nios II processor requires the setup of both the hardware interface and an embedded program. On the GUI side, a specific command triggers an interrupt on the Nios II processor.

Appendix D: Refined FFT Model

Using DSP Builder, a refined FFT model was designed and implemented to extract the required frequency component. Note that the sub-blocks are not shown to simplify the diagram.

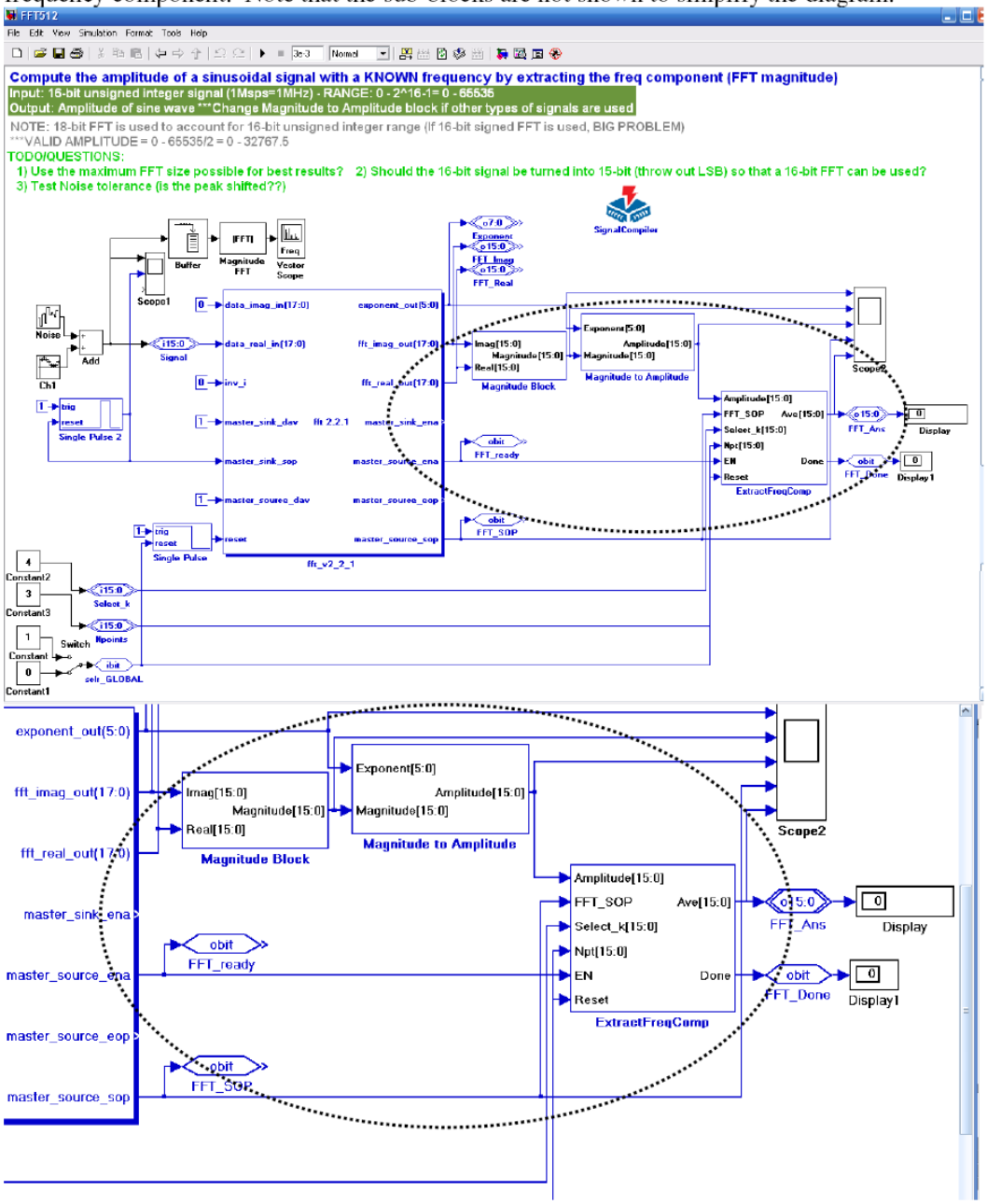


Figure D1: Refined FFT Model built using DSP Builder (integrated into Simulink).

The key to the refined model was the assumption that the signal will not change its frequency after passing through the patient's tissue. This allows the extraction of a specific frequency component without any complex signal processing schemes to locate the peaks in the FFT magnitude plot. The circle encloses the major custom designed blocks.

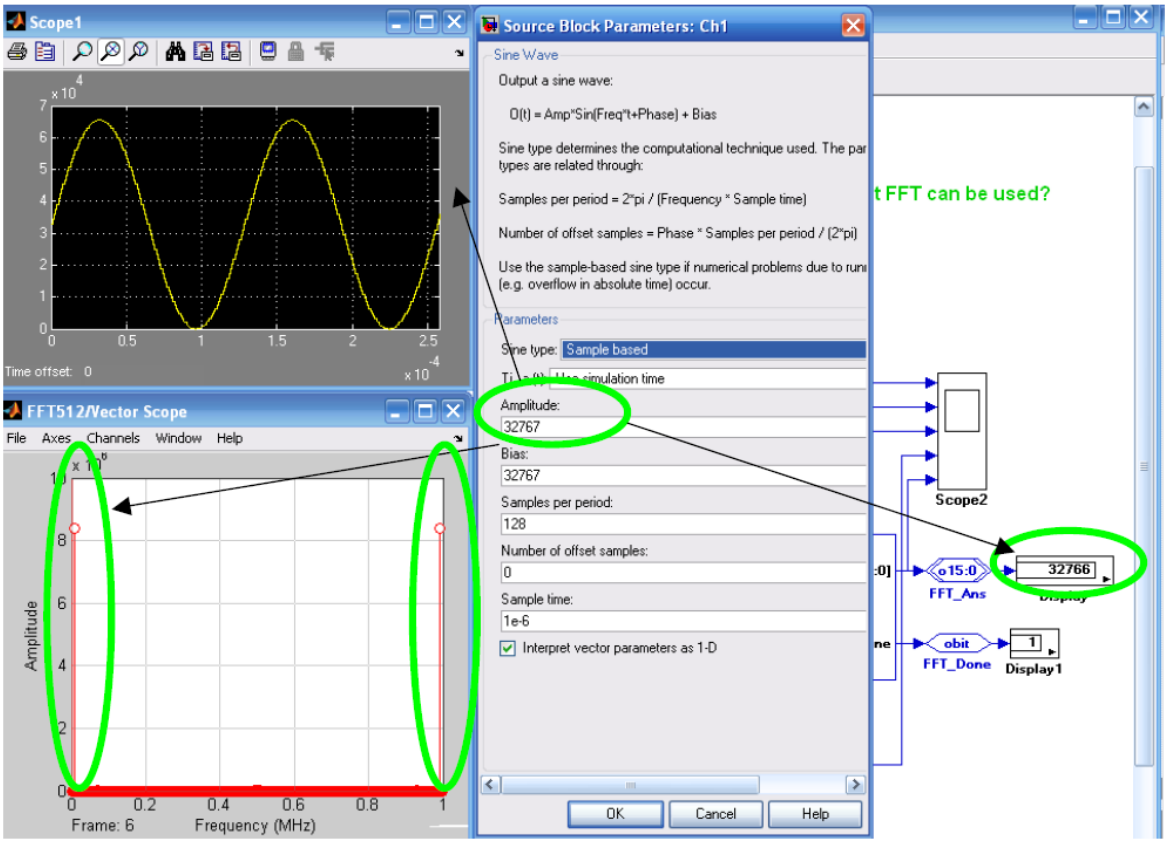


Figure D3: Demonstration of the operation of the New FFT Model in simulation.

The input to the model is a sine wave with an amplitude of 32767 (which is equal to 2^16-1 divided by 2). Note that this amplitude corresponds to the maximum swing in the 16-bit input from the 16-bit A/D converter. No negative values are expected since the analog signal corresponds to the light intensity, which is always positive. The output, called FFT_Ans, shows the amplitude extracted from the FFT magnitude plot. The value, 32766, is very close to the expected amplitude. The imprecision is the result of finite arithmetic precision in the digital signal processing blocks used.

Example #2: Secure watermark-based authentication system

Executive Summary

Significant progress has been made to date in realizing our final project goal. As a group we have accomplished the following:

- Designed overall user interface structure for …
- Made significant progress in integrating ….

I was solely responsible for a number of the tasks, all of which are now completed, or in progress. One such task was the design and implementation of … I decided that it would be best to display the PSNR values caused by… based on the assumption that our users will have a fair amount of technical knowledge.

I also conducted various tests of the signal processing code to …, and was able to verify... I decided to create …. This enabled me to perform various attacks on numerous images and see the results on one screen, which I found to be the most efficient way of testing the operation of the ….

One of the problems we encountered and are currently investigating is that we are only able to…. In addition, we have not yet determined how to enable …

1. Group Progress Summary

The goal of our project remains the same – that is, to design a user-friendly interface that makes use of our authentication and verification algorithms for a secure watermark-based digital authentication system. All of our requirements, detailed in Appendix B, have remained unchanged, and since the design review we have made significant progress in most of our requirements.

Some of our key accomplishments since the design review have been:

<bulleted list of accomplishments>

In order to accomplish the above tasks our key challenge was …. Another challenge was ….

Our group is currently on schedule. Some of the milestones (tasks #10, 11 and 12) had to be postponed, but … Below is a table outlining the key responsibilities of each member:

Student A	Student B
○ ***<bulleted list of accomplishments>*** ○	○ ○

2. Individual Progress, Work, and Contributions

My responsibilities thus far have been to design and integrate the signal processing code with the GUI's. … I have also verified that the signal processing code of the embedding and verification procedures is fully operational by creating a GUI that …

2.1 Milestone Summary

#	Task Title	Category	Status	Old Date	New Date
8	*1-line description*	Old	Completed	Nov. 13	Nov. 13
9	...	New	Completed		Nov. 27
10	*New description (previously " ")*	Modified	Completed	Jan. 1	Jan. 4
11	...	Old	Completed	Jan. 15	Jan. 30
15	*1-line description*	Old	In Progress	Feb. 5	Feb. 5

(Excerpts for tasks 8 and 15 only)

Task #8 Design and Coding of 'Image Search' Module

- Description: Design and code the portion of the interface which will be used to prompt the user to select his or her desired image for embedding or verification.
- Category: Old
- Design due week of November 13; code due week of November 27, 2006.

Responsibility : Student A (with assistance in compiling by Student B)

Status at start of reporting period

- Obtained source code from supervisor, including several images that are commonly used for authenticating purposes.

Status at end of reporting period: Completed

- Completed design and coding of Search GUI, including testing with several images

Actions

I learned how to design a GUI in Matlab by taking an online tutorial [1]. Code for obtaining an image from a particular file location was found on the Matlab website [2]. Code #1 in Appendix C is a portion of the code which I wrote in the *.m file that the GUI uses to implement the image search. Fig. 4 shows a sample display of a user-selected image.

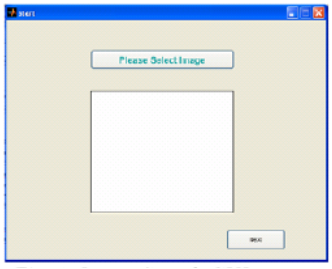


Fig. 4: Image Search GUI

Decisions

One of the decisions I had to make was which image format to use with our program. At first I was planning to only use *.bmp images, but upon testing *.gif and *.jpg formats, I found that these too worked with the program. I decided to include these as options for the user to select because…

Testing & Verification, Final Results

I tested several different image formats with the GUI I had designed, and found no problems.

Task #15 ***Code/Unit Test of 'Display Summary' Module***

Write the code to display PSNR values due to embedding distortion and give user the option to return to re-input the authentication parameters if unsatisfied with results

- **Category**: Old
- Original Due Date: week of January 15
- Completion Date: January 13

Responsibility : Student A

Status at start of reporting period: Completed design of interface, but had not started coding

Status at end of reporting period: Completed.

- Completed coding of the summary of the embedding results by January 13 (ran into one problem, detailed in 'Testing & Verification')

Actions

In order to determine how and where the PSNR values in the code were evaluated, I needed a complete understanding of the code. Due to the complexity of the code, I had to go over X's Ph.D. thesis [5] once more, and link the watermark embedding procedure to… Once I figured out which parameters were involved in evaluating…, I was able to display these values on the 'static text' object of the GUI (see Fig. 5). I was also able to integrate the embedding portion of the code with this interface so that…. Code #3 in Appendix C contains the portions of the code for calculating and displaying…

Actions (cont.)

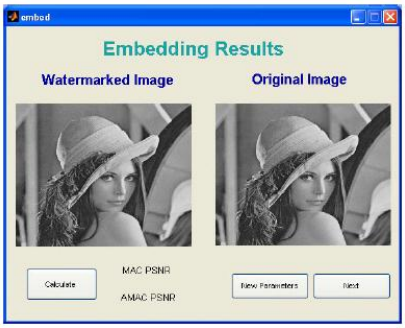


Fig. 5: Embedding Results

Decisions

In meeting this milestone, I had to decide on the best way to display the PSNR results. I chose to display them as in Fig. 6 (for entire display see Fig. 5). Since I had a limited amount of space to work with, by displaying the results over the 'Static Text' GUI object, I could relay the necessary information to the user, while efficiently making use of the space.

Fig. 6: PSNR results

Testing & Verification, Final Results

I performed 20 tests of this function by altering the authentication parameters …, to ensure that I was seeing the proper results. I found that only the value '2' for La…yielded the proper result, so I am still pursuing this problem by going through the code and the thesis [5].

3. Conclusion

Our project is currently on track with the exception of a couple of problems that were encountered. The first is that we have not yet figured out how to…. Implementation of this function will require further research of Matlab's Image Processing Toolbox, using [2], as well as looking into Matlab's GUIDE facility capabilities, using [3]. The second problem is determining the reason why we can only…. To troubleshoot this problem we must consult the thesis[5], and look over the code more thoroughly. We are still confident that we will be able to find solutions to these problems, so we expect not to have to compromise on our initial project goal.

4. References

[1] …

Appendix A: Gantt Chart (unchanged)

Activity #	Activity	Milestone?	Week: 11-Sep-06	18-Sep-06	25-Sep-06	2-Oct-06	9-Oct-06	16-Oct-06	23-Oct-06	30-Oct-06	6-Nov-06	13-Nov-06	20-Nov-06	27-Nov-06	4-Dec-06	11-Dec-06	18-Dec-06	25-Dec-06	1-Jan-07	8-Jan-07	15-Jan-07	22-Jan-07	29-Jan-07	5-Feb-07	12-Feb-07	19-Feb-07	26-Feb-07	5-Mar-07	12-Mar-07	19-Mar-07
1	Proposal	x	1	2																										
2	Requirement Specifications	x		1	1	2	2																							
3	Design Review	x						1	2	1																				
4	Design - UI										1	1	1																	
5	Code/Unit Test - UI													1	1															
6	Design - Save Image															1														
7	Code/Unit Test - Save Image															1														
8	Design - Search										2	2																		
9	Code/Unit Test - Search												2	2																
10	Design - Display Img														2	2														
11	Code/Unit Test - Display Img																2	2												
12	Design - Input																1	1	1											
13	Code/Unit Test - Input																			1	1	1								
14	Design - Summary																		2											
15	Code/Unit Test - Summary																			2	2									
16	Test Signal Processing																					2	2	2						
17	Integrate UI modules																						1	1						
18	Test UI																								1	1				
19	Integrate UI/SP Modules																								2	2				
20	System Test (Set A)																										1	1	1	1
21	System Test (Set B)																										2	2	2	2
	Key		done			started				after original schedule						before original schedule														

Appendix B – Project Goal and Requirements …
Appendix C - Sample code and GUI interface

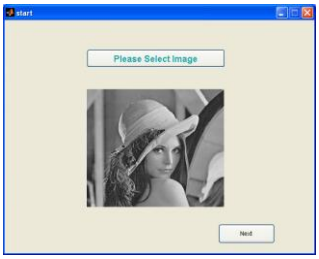


Fig. 3: Display of user-selected image for embedding/verification

Code #2: portion of code to implement the above image display

```
[I,map]=imread(imgname);
I = double(I); [M,N]=size(I);   % size of the image

handles.Image=I;
handles.M=M;
handles.N=N;
guidata(hObject, handles); ….
```