Touch-based Adaptive Predictive Screens

Design Report

5/6/2015
College of Engineering, University of Idaho

Agbecha, Irene
Email: agbe0821@vandals.uidaho.edu

David Barry
Email: barr5257@vandals.uidaho.edu

Jonathan Simmons
Email: simm4031@vandals.uidaho.edu

TABLE OF CONTENTS

Executive Summary

Touch screen devices are used in almost every area of life today and are often in unstable environments. Changing environmental factors introduce user problems like false touches (the activation of unintended functionality on a touch screen). The problem is software is not available to compensate adequately for the changes in these unstable environments. An example of where this problem is encountered is on air planes where the software systems are operated using touch screens. The touch screens are usually small enough that when turbulence (introducing vibration) is experienced, the chance of false touches is increased. Another scenario is a touch screen in a police car in motion. Our project is aimed at creating a finished product for Esterline (www.esterline.com) that will make touch screen based systems more adaptable to changes in both light intensity and vibration, and ultimately greatly improve the touch screen user experience.

Background

Esterline touchscreen products make their way into hospital equipment, airplanes, jets, cars, etc. This indicates that they are exposed to a wide variety of environmental factors such as light, therefore inspiring the need to make them adaptive to these different environments. In some of these environments such as airplanes, environmental factors pose a more serious problem. An example would be turbulence experienced on an airplane, introducing vibration. In a situation like this, the problem of 'false touches' could be encountered. A false touch is the accidental trigger of unintended functionality. The dangers and problems of false touches are apparent—ranging from the possibly catastrophic effects of triggering unintended functionality to the frustration of users. Having touchscreens that are

able to adapt dynamically to these factors that they are exposed to, will not only improve the user experience, but make touchscreen systems safer and less prone to error.

Problem Definition

The main goal of completing this project is to develop enhancements for touchscreen products that will enable them adapt dynamically and efficiently to changing environmental factors; specifically to changes in light intensity and the introduction of vibration to the system. These are the two environmental factors of focus in this project.

One of the sub-goals of the project was to make the system able to compensate for the presence of vibration and for changes in light intensity. A very important sub-goal of the project was to incorporate a configurable compensation switch. This is so that tests can be performed with and without compensation to enable an accurate comparison of performance with and without compensation for each of the factors. A final and equally important sub-goal was to design and conduct experiments that adequately test the system functionality. The purpose of this is to provide a way assess the effects of the compensation solutions that will be implemented. A complete needs table is available in appendix A (Table2).

Deliverables for the project include a system design, software algorithms, experimental data to prove system functionality, system documentation, and a prototype.

With regards to specifications, the two most important metrics to stress are the ‘percentage of false touches’ and the ‘discernibility of on screen characters.’ The

‘percentage of false touches’ metric measures the effectiveness of the vibration solution. This measurement is obtained by keeping track of the number of false touches encountered during the usage of the touchscreen system with and without vibration compensation. The ‘discernibility of on screen characters’ metric asses the efficacy of the light compensation solution by keeping track of the touchscreen’s visibility under varying lighting conditions. A complete specifications table is available in appendix A (Table1).

This project had a few constraints. The most notable ones include the software-based solution requirement, hardware limitations, and testing limitations. The solution were required to be software based solutions. Some mechanical solutions were proposed to the client (Esterline) but rejected. Testing constraints seemed to pose a problem as it was difficult to find suitable testing equipment; a variable light source for light intensity testing and a vibration table/chamber for vibration testing.

Project Plan

The development process model used for this project is the waterfall model. A complete diagram is provided in appendix B (figure1). Also, a complete schedule is provided in appendix B (figure2). The team members had different responsibilities that were assigned based on member skills and interest. Assignments were as described in the table below.

Irene Agbecha	Client primary contact
Jonathan Simmons	Keeping team documents and purchasing
David Barry	Meeting minutes and agendas

Concepts Considered

The need for sensors to perceive source light intensity (light intensity at the surface if the touchscreen) and vibration were apparent, however, the choices were difficult to make. All the approaches we considered attempt to compensate for vibration only in two axes; X and Y.

Vibration Compensation

After considering many vibration sensors, we decided to use an accelerometer for vibration sensing since that gives greater versatility in terms of how that data can be used. In addition to sensing vibration, accelerometer data can be used to obtain positional information.

One of the main ideas we originally considered for vibration compensation was enlarging portions of the touchscreen based on anticipations using a proximity sensor. The criterion for this was that vibration makes it difficult to see and access mainly small screen components. Therefore, enlarging these components would at the very least, majorly reduce the effects of vibration on the touchscreen GUI, making it easier to access GUI components. After more consideration of this approach, we concluded that we would need a proximity sensor to keep track of the user's finger movement on the touch screen. This is what would provide information about which sections of the screen to enlarge. The screen was to be divided into four quadrants and based on which quadrant the user's finger is in, that quadrant would be enlarged and displayed. For proximity sensing, we first considered using three small area digital sensors positioned strategically to help determine the user's finger position on the screen. Two were to be placed at the top of the touchscreen on both sides and one was to be placed down at the bottom. A

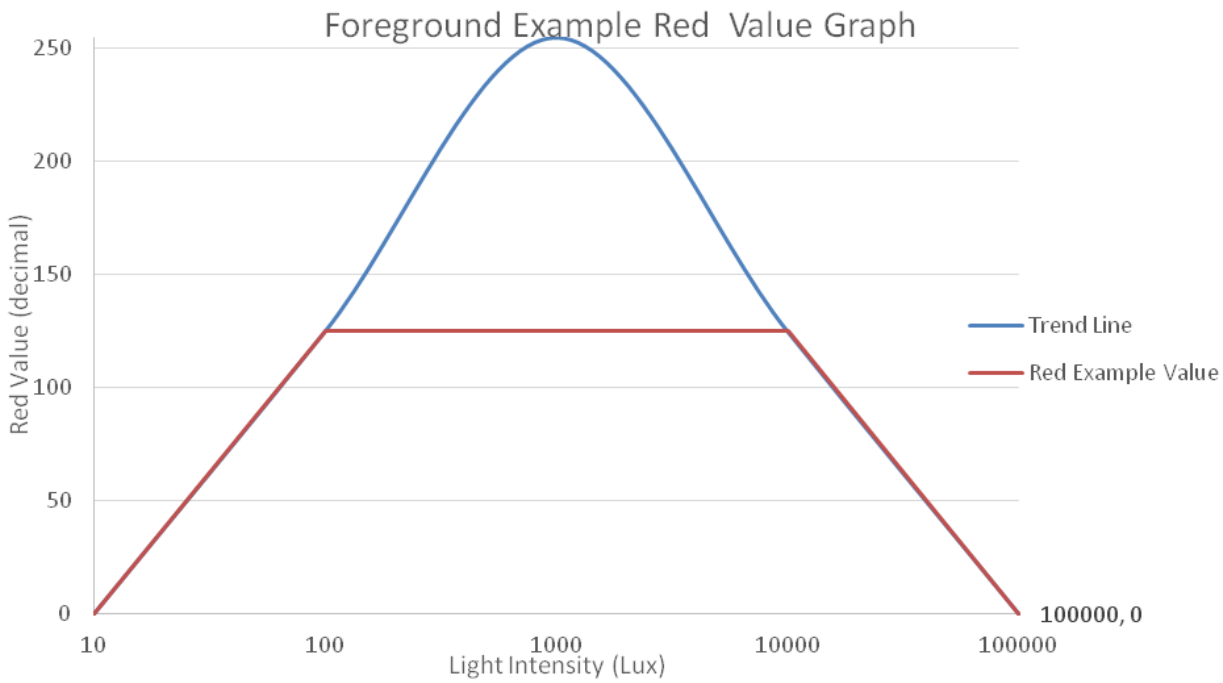
comparison between the top and bottom sensors would tell whether the user's finger is in a top or bottom quadrant and comparison between the sensor on the right and that on the left would tell if the user is in a left or right quadrant. This way a specific quadrant could be determined for enlargement. We called this solution "Proportional Magnification."

The second solution which we called, "Steady State" is a very different approach. This approach involves moving the GUI components frequently to counteract vibration experienced by the touchscreen system. It might be apparent that this solution requires a system that has the speed needed to keep track of and respond to every single pulse or rhythmic movement of the touch screen frame. Here, the GUI components are moved in the direction opposite the movement of the entire touchscreen system. This happens as frequently as the touchscreen vibrates. The overall effect of this is that when the user views the touchscreen GUI during vibration compensation, the GUI appears to be stable or unmoving, even in the presence of vibration.

Light Compensation

We initially considered changing screen display light intensity as a way of compensating for changing source light intensity. However, after learning that we could not obtain the drivers necessary to manipulate the touchscreen hardware to do this, we disregarded this as an approach. Esterline's Hardware Engineer, Mitch Butzer, suggested a contrast ratio manipulation approach. This approach essentially involves modifying the contrast ratio of GUI components components—mainly between foreground and background components—in order to achieve better visibility under different source light intensities. This idea was further developed

by creating a correlation between source light intensity and the expected contrast between the foreground and the background GUI components. A graph illustrating this is below.



Contrast ratio is modifications are achieved by modifying the RGB (Red green Blue) color values of the GUI components.

Testing

One of our requirements for this project was to develop an experimental system that would test the functionality of the finished touchscreen compensation system. We came up with test program ideas for both the vibration compensation solution and the light compensation solution.

For vibration compensation testing, we considered incorporating a false touch counter that would keep track of the number of false touches encountered by the test user with and without vibration compensation.

On the other hand, for light testing, we implemented "discernibility checkers". Discernibility checkers consist of displayed text and text fields for entering the displayed text. Based on text entered the number of discernible characters is determined by comparing the displayed text with the user entered text. This was our idea for determining user visibility under varying lighting conditions. The GUI storyboard in appendix B figure3-figure5) portrays very carefully, the initial idea for the test program.

Concepts Selection

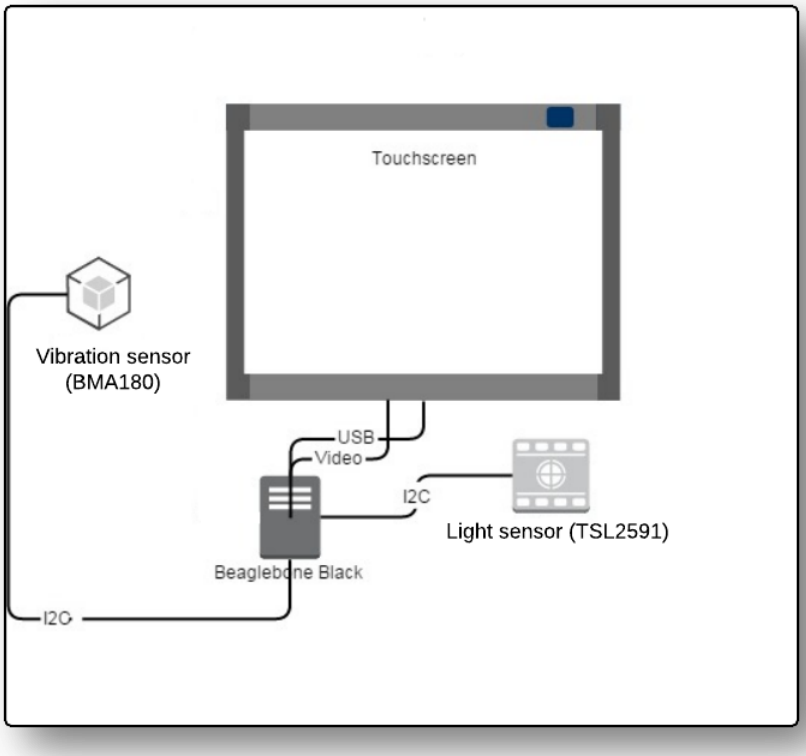
The initially considered concepts for light compensation and for testing were developed and implanted. For vibration compensation however, a choice had to be made between the two solution considered as the resources for implementing both were not available. A careful comparison was done and a final decision was made based on pros and cons of each solution, as expressed in the table below

	Proportional magnification	**Steady state**
Pros	• Can tolerate larger magnitudes of vibration • Better for rhythmic vibration	• Non-intrusive • Does not need proximity sensor • Better for pulse vibration but also good for rhythmic vibrations
Cons	• Requires a proximity sensor • Slightly intrusive to new users • Terrible for pulse vibrations	• Shrinking screen/on-screen components • System speed requirement

From analyzing the table above, we concluded that the steady state solution has more pros, is a more elegant solution to the problem, and is more suitable for the two kinds of vibrations that we expect the system to encounter—pulse vibration(sudden and non-recurrent) and rhythmic vibration (recurrent). The last rationale was the most instigating.

System Architecture

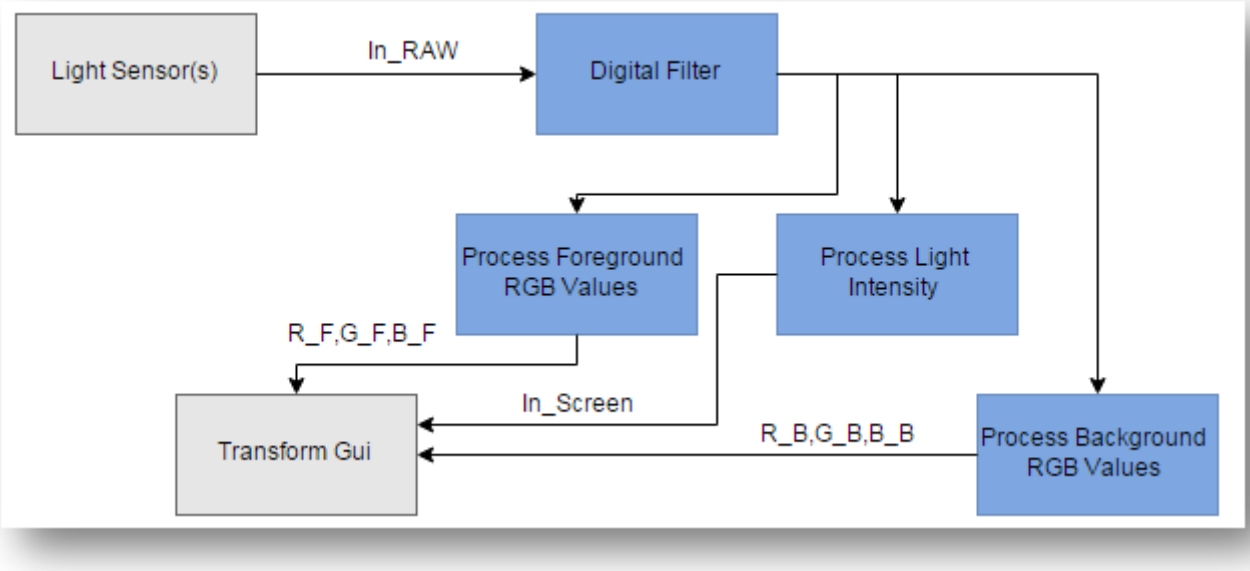
On completion of the conceptual design creation, we created a physical model of the system as well as a software model. A diagram of the physical model is below.



Physical Model

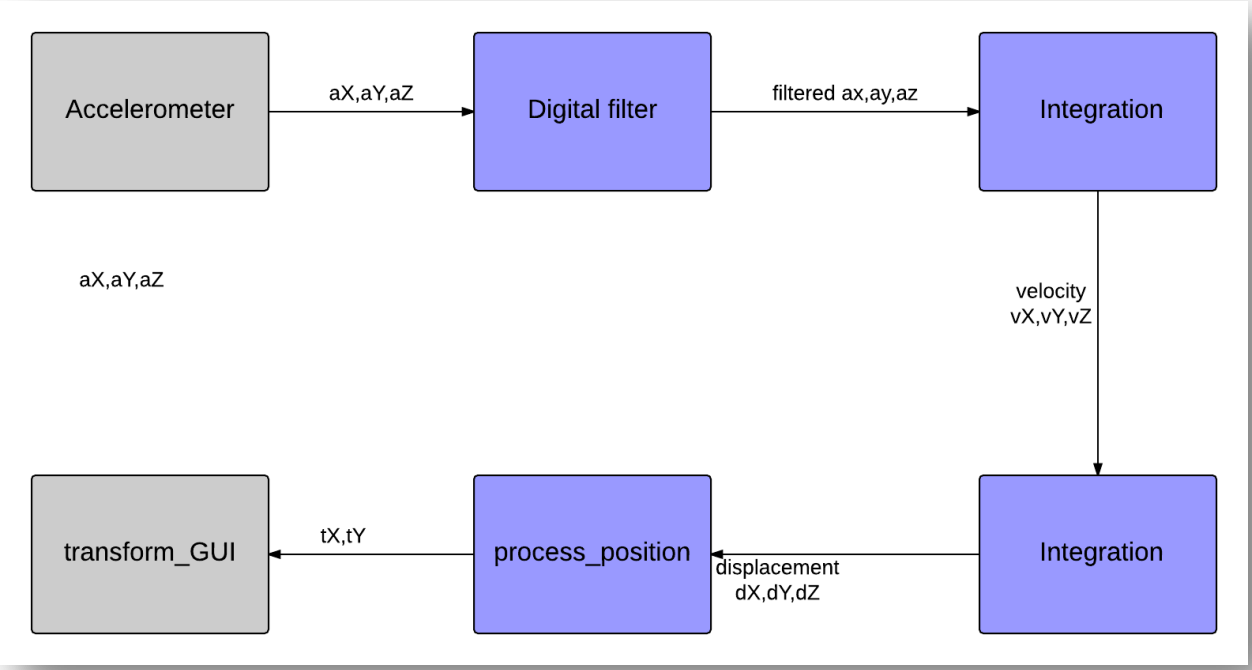
This physical model consists of a light sensor for source light intensity, and accelerometer for vibration sensing (vibration sensor), a controller—the beagle bone black in this case—and a touchscreen. Both sensors are connected to and communicate with the controller through an I2C interface. The controller has a processor that runs at 1GHz and processing information from both sensors in addition to supporting the software application framework and the application itself. The framework used for GUI application development is QT and the language of development is C++.

The software model on the other hand consists of algorithms for data processing sensor data, compensation implementation, and the underlying QT framework.



Light compensation algorithm

This algorithm this they key component of the light compensation. It processes the data from the light sensor and provides output that is finally used to compensate, in the GUI, for light intensity changes.



Vibration compensation algorithm

The algorithm is uses data from the vibration sensor to provide GUI transformation information which is used to compensate for the presence of the vibration in the system.

Testing and Results

The GUI created as illustrated in the storyboard in appendix B (figure3-figure5), incorporates testing capabilities in addition to implementation. As described in the 'concepts considered' section of this report, a false touch counter was implemented for testing vibration compensation efficiency and a 'discernible checker' was also implemented for testing light compensation efficiency. A vibration chamber was used to simulate vibration for testing vibration compensation. For light testing, the enhanced touchscreen system was exposed to varying lighting conditions including an intense high energy light lamp. After some data was collected, our analysis

based on the requirements showed a 27.7% increase in discernibility of on screen characters with light compensation. This was as expected. For vibration compensation however, no significant decrease in the number of false touches was observed. After more testing, observation, and analysis, we came to the conclusion that vibration compensation did not produce results for mainly two reasons: test equipment limitation and hardware limitations. The vibration chamber used for vibration testing did not provide type of vibration in the X and Y axis that we had anticipated. The hardware limitation we experienced was in the processor's inability to handle sensor sample rates beyond 200Hz per sensor. Every attempt to sample higher led to 100% CPU usage and the eventual slowing down and crashing of the entire touchscreen system.

Future Work

An attempt at the proportional magnification solution discussed earlier in the concepts section would be a good task to undertake for continued development of this system.

Also, the implantation of compensation for shadows cast on the screen with respect to light compensation would be another good sub-goal to add to the list of goals and sub-task to tackle. This would be very complex to implement and would require going through the entire life cycle of the product again—from requirements gathering all the way up to testing and validation. This is because this new requirement would have to be clearly understood from the Client's standpoint before proceeding.

Budget Summary

Date	Supplier	Cost	
4/24/2015	*SF Cable*	$ 20.45	Fabrication
4/29/2015	*Amazon*	$ 51.04	Accelerometer
4/24/2015	*Amazon*	$ 20.50	Fabrication
4/20/2015	*Amazon*	$ 9.00	Fabrication
4/20/2015	*Amazon*	$ 10.04	Fabrication
4/20/2015	*Amazon*	$ 8.04	Fabrication
11/12/2015	*Amazon*	$ 13.02	Light Sensor
11/14/2015	*Amazon*	$ 53.50	Connectors
10/31/2015	*Digi-Key*	$ 60.05	Acclerometer & Light Sensor
2/13/2015	*Adafruit*	$ 20.85	Light Sensors
3/27/2015	*Amazon*	$ 220.00	Vibration Plate
N/A	*N/A*	$ 300.00	Trips to Esterline
Feb-15	*N/A*	$ 50.00	Acclerometer
Feb-15	*RadioShack*	$ 35.00	Power Inverter & Outlet
Apr-15	*N/A*	$ 130.00	Poster Board
	Total	$ 1,001.49	

Appendices

Appendix A

Metrics	Units	Unit Alias	Marginally Acceptable	Ideal
Percentage of false touches	Number of false touches per 1,000 touches	FTT	20% decrease with compensation	80% decrease with compensation
Discernibility of On-screen Characters	Number of discernible characters per page	CPP	50% increase in CPP with compensation	90% increase with compensation
Response Time	Microseconds	µs	Maintains at most the current response time	Reduces the current response time
Accuracy	Millimeters	mm	20% increase with compensation	50% increase with compensation
Aesthetically pleasing	Subjective	N/A	Presentable	Very aesthetically pleasing and trendy
Cost of product prototype	Dollars	$	Within budget	Within budget
System stability with and without peripherals	Binary	N/A	Yes	Yes: with all peripherals
Number of interfaces	Interfaces	Int	USB	USB
System Controller	Hardware Model	N/A	Any capable processor	ARM Cortex A8
Togglable Compensation	Binary	N/A	Yes	Yes

Table1: Specifications Table

Needs	Importance
Compensates for Vibration	5
Compensates for changes in Light intensity	5
Responds Quickly	5
Responds Accurately	5
Compensates for Finger Jitter	4
GUI is aesthetically pleasing	3
Hardware aesthetically pleasing	2
Is Cost Effective	3
Has modular components	3
Is Portable/platform independent	2
Has easily Useable(to client) Hardware	1
Has Compensation Switch	5
Is able to Collect Experimental Data	5

Table2: Needs Table

Appendix B

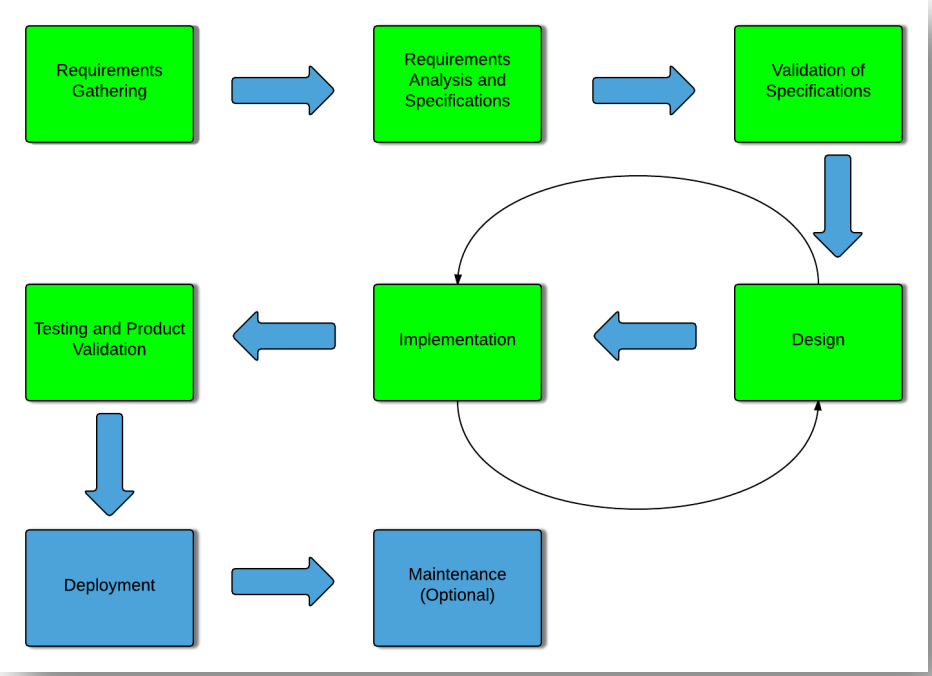


Figure1: Process Model

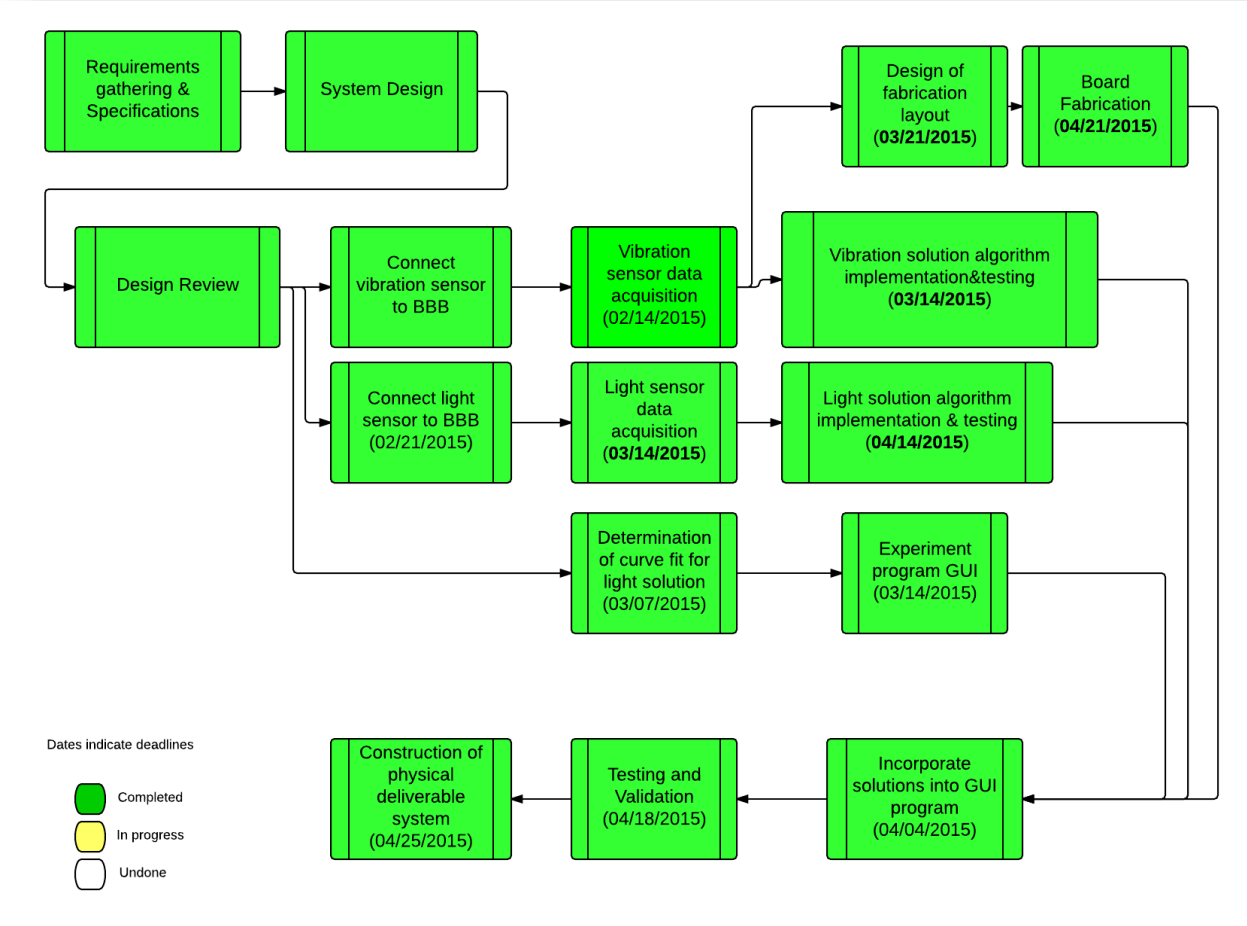


Figure2: Project Timeline

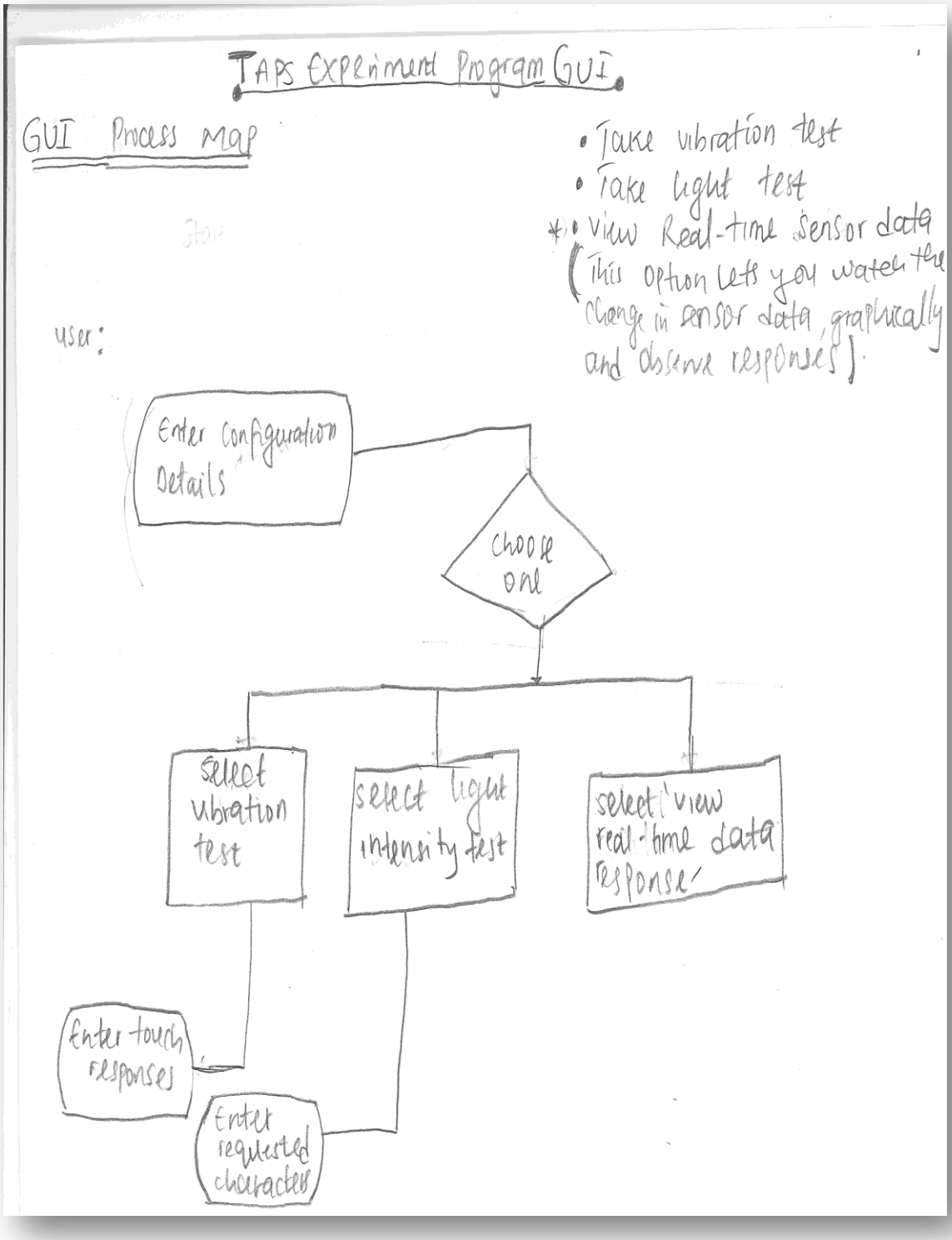


Figure3: storyboard

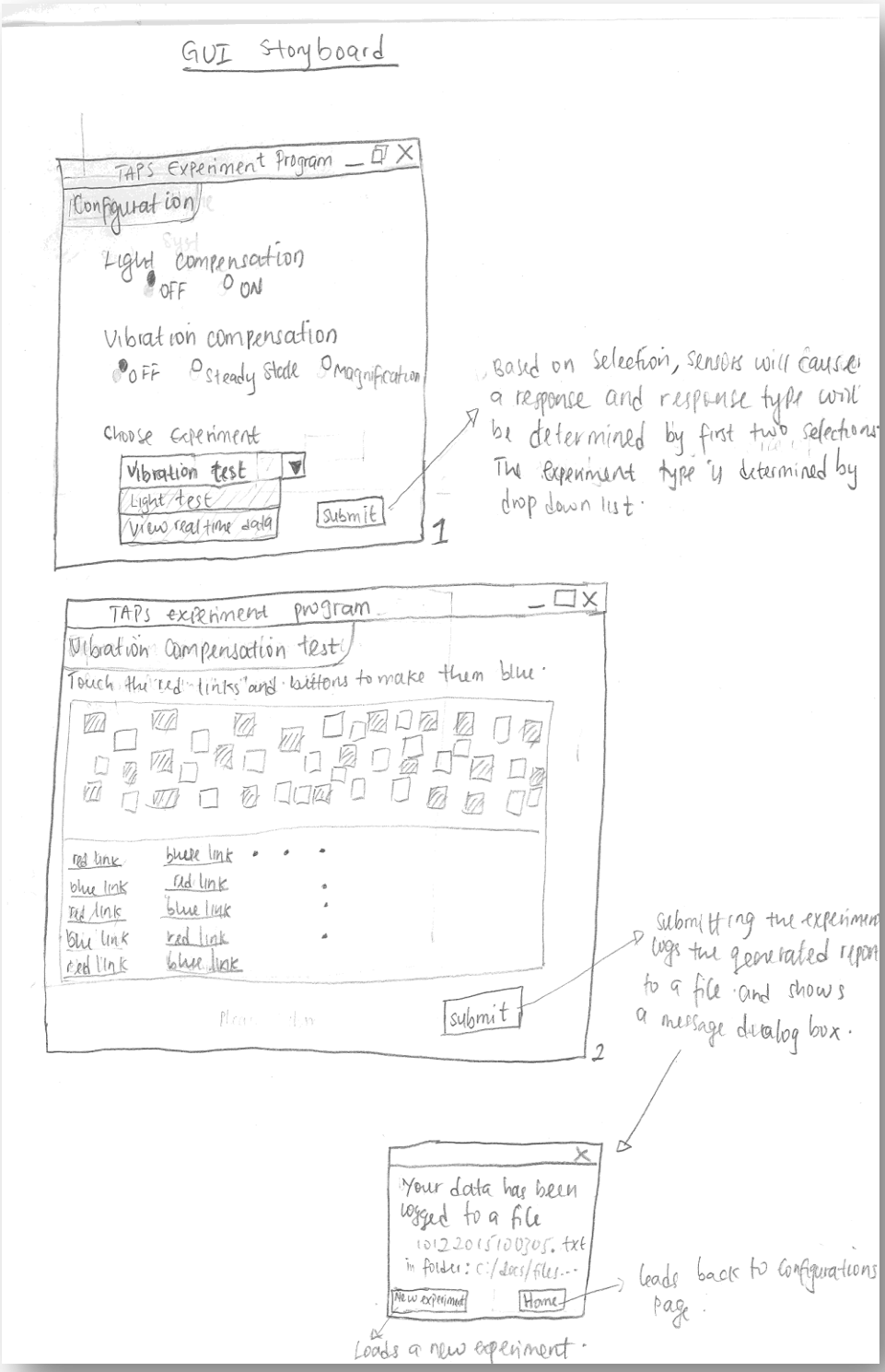


Figure4: storyboard

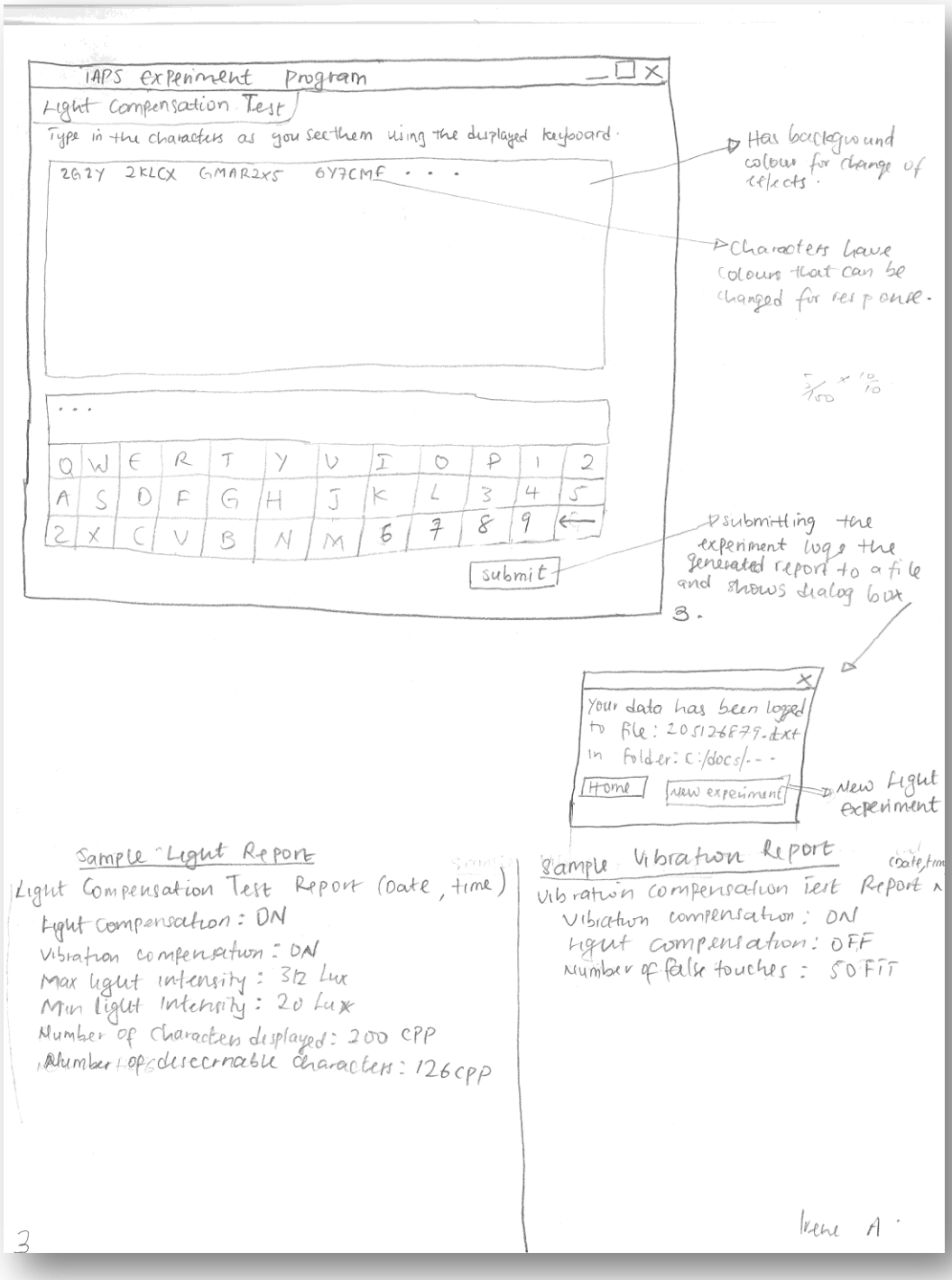


Figure5: storyboard

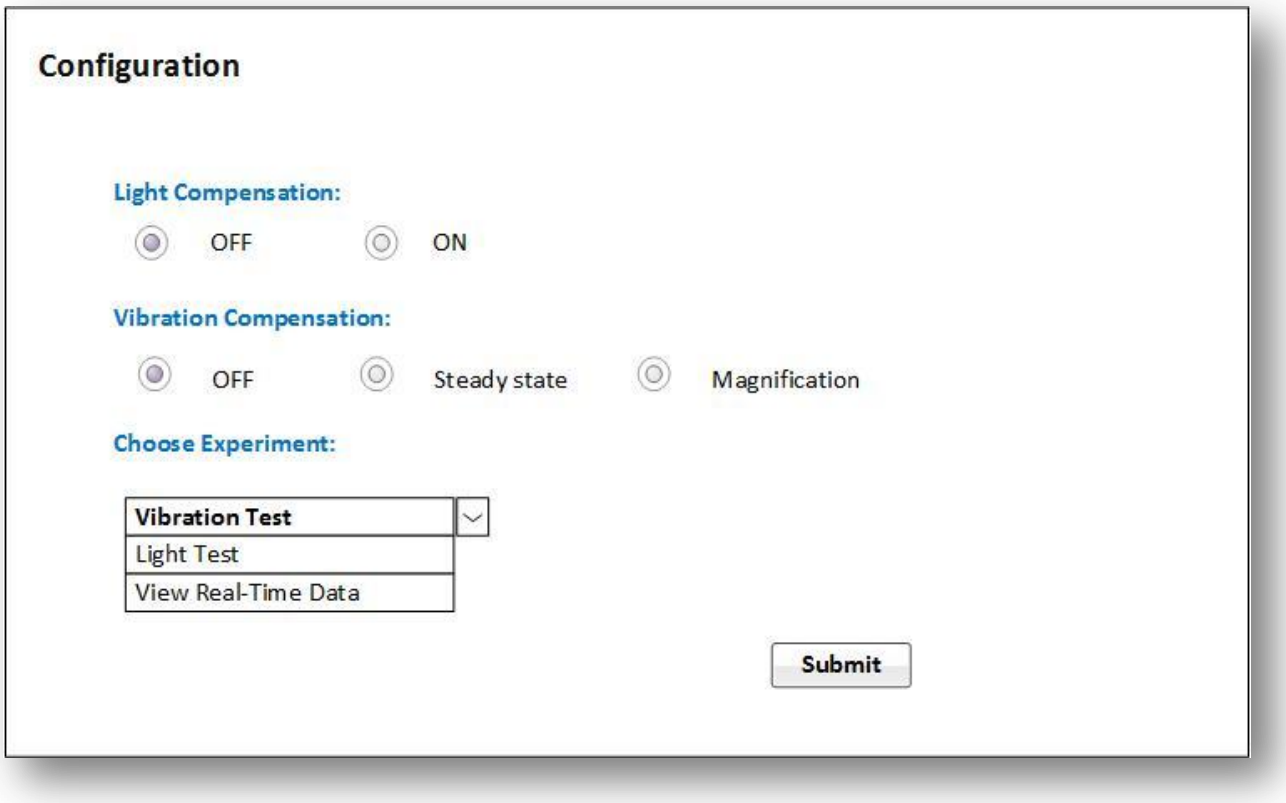


Figure6: GUI wireframe

Vendor Datasheets

BMA180 Accelerometer: http://irtfweb.ifa.hawaii.edu/~tcs3/jumpman/jumppc/1107-BMA180/BMA180-DataSheet-v2.5.pdf

TSL2591 Light Sensor: https://learn.adafruit.com/downloads/pdf/adafruit-tsl2591.pdf

Beagle Bone Black controller: http://beagleboard.org/BLACK